# INDIRA COLLEGE OF ENGINEERING AND MANAGEMENT, PUNE

(An Autonomous Institute Affiliated to Savitribai Phule Pune University Pune)

Parandwadi, Pune – 410506, Ph. 02114 661500, www.indiraicem.ac.in

Approved by AICTE & Government of Maharashtra, Accredited by NAAC

## Department of Master of Computer Applications

FY & SY MCA (Batch 2025-27)

National Education Policy (NEP)-2020 Compliant Curriculum

Structure and Syllabus

Prepared by: - Board of Studies in MCA

Approved by: - Academic Council, ICEM, Pune

(Pattern 2025)

# MCA Curriculum (2025 Pattern)

# Batch of 2025 - 2027

## INSTITUTE VISION-MISSION/QUALITY POLICY

### Our Vision

The institute envisions to develop itself into a center of academic excellence in the field of Engineering and Management education in order to develop future technocrats and managers having right knowledge, skill and attitude to serve the society and industries to fulfill their ever changing requirements.

### Our Mission

- To train our students to become best Engineering Entrepreneurs today, who will lead the organizations successfully into the future; locally, nationally and globally.
- To provide an environment which fosters continuous improvement & innovation with related technical support & facilities to enhance student and faculty effectiveness.
- To provide programs focusing on the holistic development of the individual with the emphasis on personality grooming, physical fitness and a strong sense of social and environmental responsibility.
- To improve logic & scientific reasoning and to develop global mindset amongst the students and prepare them to work in heterogeneous environment.

### Quality policy

We are committed to quality engineering / management education and continual quality enrichment by establishing and applying mechanisms for satisfaction of our stakeholders.

| | | |
|---|---|---|
| Dr. Darshana Desai | Dr. Saurabh Gupta | Dr. Nilesh Uke |
| HOD-MCA, BoS Chairman | Dean-Academics | Principal, ICEM |

# MCA DEPARTMENT

## VISION-MISSION

### Vision

- To offer a well-balanced program of instructions, practical experience and to provide opportunities for overall development, groom the students in excellent professionals, knowledge seekers and good human being.

### Mission

- To prepare learner-sensitive educators with the knowledge, skills, and dispositions to contribute to a better society.
- To ensure quality learning & teaching take place in the classroom every day.
- To provide equitable access to quality education for the students.
- Teacher facilitate learning & constantly nature every learner.
- Our Mission is to foster the success of our students and their communities through innovative, flexible learning opportunities for people of all ages, backgrounds, and aspirations resulting in self-fulfillment and competitiveness in an increasingly global society.
- To strive for excellence in development and deployment of computer applications by empowering students.
- To impart quality and value based education to raise satisfaction level of all stakeholders and enhancing sense of social responsibility.

| | | |
|---|---|---|
| Dr. Darshana Desai | Dr. Saurabh Gupta | Dr. Nilesh Uke |
| HOD-MCA, BoS Chairman | Dean-Academics | Principal, ICEM |

**MCA Programme Outcomes (POs):**

At the end of the MCA programme the learner will possess the following Program Outcome:

| | |
|---|---|
| PO1 | **Computational Knowledge:** Relate & apply fundamental knowledge of computing technology appropriate to the discipline |
| PO2 | **Problem Analysis:** Ability to Analyze, identify and formulate tangible products/services/solutions/applications with computing requirements to solution. |
| PO3 | **Design/development of solutions:** Ability to design, implement and evaluate a computer-based system, process, component or program to meet desired needs based solution with appropriate consideration for public health and safety, cultural, societal, and environmental considerations. |
| PO4 | **Conduct & investigate complex computing problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide valid conclusions. |
| PO5 | **Modern Tool usage:** Create, select, adapt and apply appropriate techniques, resources, and modern computing tools to complex computing activities, with an understanding of the limitations. |
| PO6 | **Professional Development Ethics:** Understand and commit to professional ethics and cyber regulations, responsibilities, and norms of professional computing practice. |
| PO7 | **Lifelong learning:** Recognize the need, and have ability to engage in independent learning for continual development as a Computing professional. |
| PO8 | **Project management and finance:** Demonstrate knowledge and understanding of computing and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| PO9 | **Communicate Effectively:** Communicate effectively with the computing community, and with society at large, about complex computing activities by being able to comprehend and write effective reports, design documentation, make effective presentations, and give and understand clear instructions. |
| PO10 | **Social and environmental concern(S):** Understand and assess societal, environmental, health, safety, legal, and cultural issues within local and global contexts, and the consequential responsibilities relevant to professional computing practice. |

Dr. Darshana Desai
HOD-MCA, BoS Chairman

Dr. Saurabh Gupta
Dean-Academics

Dr. Nilesh Uke
Principal, ICEM

| | |
|---|---|
| PO 11 | **Team Work and Leadership:** Function effectively as an individual and as a member or leader in diverse teams and in multidisciplinary environments. |
| PO 12 | **Innovation and Entrepreneurship:** Identify a timely opportunity and using innovation to pursue that opportunity to create value and wealth for the betterment of the individual and society at large. |

Following are the session details per credit for each of L-P-T model

1) Every ONE-hour session per week of Lecture(L) amounts to 1 credit per semester,
2) Minimum of TWO hours per week of Practical(P) amounts to 1 credit per semester,
3) Minimum of ONE hours per week of Tutorial(T) amounts to 1 credit per semester

**Lecture-Practical/Project-Tutorial (L-P-T)**

A course shall have either or all the three components, i.e. a course may have only lecture component, or only practical/project component or a combination of any two/three components

- **Lecture (L):** Classroom sessions delivered by faculty in an interactive mode. It should be conducted as per the scheme of lectures indicated in the respective course.
- **Practical/Project(P):** Practical / Project Work consisting of Hands-on experience /Field Studies / Case studies that equip students to acquire the much-required skill component. Besides separate Practical/Project course, three courses in each semester include few practical assignments and it will be evaluated under internal evaluation
- **Tutorial(T):** Session consisting of participatory discussion/ self-study/ desk work/ brief seminar presentations by students and such other novel methods that make a student to absorb and assimilate more effectively the contents delivered in the Lecture sessions

# MCA Syllabus Structure

## 2025-27

| Semester-I | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Teaching Scheme | | | | | Evaluation Scheme | | |
| Course Code | Course Type | Course | L | P | T | TOTAL HRS | TOTAL Credits | CIA | ESE | Total |
| MCA101 | CM | JAVA Programming | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA102 | CM | Data Structure & Algorithms | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA103 | CM | Object Oriented Software Engineering | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA104 | CM | Operating Systems and Shell Programming | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA105 | CM | Mathematics and Business Statistics | 2 | 0 | 0 | 2 | 2 | 50 | 50 | 100 |
| MCA101L | SEC | JAVA Programming Lab | 0 | 4 | 0 | 4 | 2 | 50 | 0 | 50 |
| MCA102L | SEC | Data Structure & Algorithms Lab | 0 | 4 | 0 | 4 | 2 | 50 | 0 | 50 |
| MCA106 | ME | (Major Elective 1)<br>Elective 1.1: Fundamentals of Cloud Computing & Networking<br>Elective 1.2: Web Technologies<br>Elective 1.3: Data Science | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA106L | ME | Elective1 Subject Lab | 0 | 2 | 0 | 2 | 1 | 50 | 0 | 50 |
| MCA107 | PROJ | Capstone Project | 0 | 4 | 0 | 4 | 2 | 50 | 0 | 50 |
| MCA108 | CM | Soft Skills & Business Communication | 1 | 0 | 0 | 1 | 1 | 25 | 0 | 25 |
| MCA109 | IKS | Yoga | 1 | 0 | 0 | 1 | 1 | 25 | 0 | 25 |
| | | Noncredit course (Bridge course) | | | | | | | | |
| | | | **19** | **14** | **0** | **33** | **26** | **550** | **300** | **850** |

Dr. Santosh Deshpande
BoS Member

Dr. Darshana Desai
HOD-MCA, BoS Chairman

Dr. Saurabh Gupta
Dean-Academics

Dr. Nilesh Uke
Principal, ICEM

INDIRA COLLEGE OF ENGINEERING & MANAGEMENT PUNE-411048
Principal Office

| Semester II | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Teaching Scheme | | | | | Evaluation Scheme | | |
| Course Code | Course Type | Course | L | P | T | TOTAL HRS | TOTAL Credits | CIA | ESE | Total |
| MCA201 | CM | Python Programming | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA202 | CM | Advanced Database Management System | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA203 | CM | Software Testing & Tools | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA204 | RM | Research Methodology | 3 | 0 | 1 | 4 | 4 | 50 | 50 | 100 |
| MCA201L | CM | Python Programming Lab | 0 | 4 | 0 | 4 | 2 | 50 | 0 | 50 |
| MCA205 | PROJ | Mini Project | 0 | 4 | 0 | 4 | 2 | 50 | 0 | 50 |
| MCA206 | ME | (Major Elective 2)<br>Elective 2.1: Cloud Computing Management and Network Security<br>Elective 2.2: Advance Web Technologies<br>Elective 2.3: Artificial Intelligence & Machine Learning | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA206L | ME | Elective2 Subject Lab | 0 | 2 | 0 | 2 | 1 | 50 | 0 | 50 |
| MCA207 | ME | (Major Elective 3)<br>Elective 3.1: IOT<br>Elective 3.2: Blockchain<br>Elective 3.3: Power BI | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA207L | ME | Elective3 Subject Lab | 0 | 2 | 0 | 2 | 1 | 50 | 0 | 50 |
| MCA208 | IKS | Vedic Mathematics | 1 | 0 | 0 | 1 | 1 | 50 | 0 | 50 |
| | | | | | | | | | | |
| | | | | | | | | | | |
| | | | **19** | **12** | **1** | **32** | **26** | **550** | **300** | **850** |

Dr. Santosh Deshpande
BoS Member

Dr. Darshana Desai
HOD-MCA, BoS Chairman

Dr. Saurabh Gupta
Dean-Academics

Dr. Nilesh Uke
Principal, ICEM

INDIRA COLLEGE OF ENGINEERING & MANAGEMENT, PUNE-410506
Principal Office

| Semester III | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Teaching Scheme | | | | | Evaluation Scheme | | |
| Course Code | Course Type | Course | L | P | T | TOTAL HRS | TOTAL Credits | CIA | ESE | Total |
| MCA301 | CM | Mobile Application Development | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA302 | CM | Software Project Management & DevOps | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA301L | CM | Mobile Application Development Lab | 0 | 4 | 0 | 4 | 2 | 50 | 0 | 50 |
| MCA303 | ME | (Major Elective 4)<br>Elective 4.1 Cloud Migration and Management<br>Elective 4.2 MERN Stack Development<br>Elective 4.3 Deep Learning | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA303L | ME | Elective4 Subject Lab | 0 | 2 | 0 | 2 | 1 | 50 | 0 | 50 |
| MCA304 | ME | (Major Electives 5)<br>Elective 5.1 Cloud API & Services<br>Elective 5.2 UI-UX Design<br>Elective 5.3 Tableau | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA304L | ME | Elective5 Subject Lab | 0 | 2 | 0 | 2 | 1 | 50 | 0 | 50 |
| MCA305 | ME | (Major Electives 6)<br>Elective 6.1 Cyber Security and Ethical Hacking<br>Elective 6.2 E commerce & Digital Marketing<br>Elective 6.3 Generative AI Prompt Engineering | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA305L | ME | Elective6 Subject Lab | 0 | 2 | 0 | 2 | 1 | 50 | 0 | 50 |
| MCA306 | RP | Research Project | 0 | 12 | 0 | 12 | 6 | 200 | 0 | 200 |
| | | | | | | | | | | |
| | | | **15** | **22** | **0** | **37** | **26** | **650** | **250** | **900** |

Dr. Santosh Deshpande
BoS Member

Dr. Darshana Desai
HOD-MCA, BoS Chairman

Dr. Saurabh Gupta
Dean-Academics

Dr. Nilesh Uke
Principal, ICEM

INDIRA COLLEGE OF ENGINEERING & MANAGEMENT, PUNE-411006
Principal Office

| Semester IV | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | Teaching Scheme | | | | | Evaluation Scheme | | |
| Course Code | Course Type | Course | L | P | T | TOTAL HRS | TOTAL Credits | CIA | ESE | Total |
| MCA401 | MOOC | Self-Learning Course/ MOOC-1 | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA402 | MOOC | Self-Learning Course/ MOOC-2 | 3 | 0 | 0 | 3 | 3 | 50 | 50 | 100 |
| MCA403 | MOOC | Self-Learning Course/ MOOC-3 | 2 | 0 | 0 | 2 | 2 | 0 | 50 | 50 |
| MCA404 | OJT | Industrial Internship/On Job Training | 0 | 24 | 0 | 24 | 12 | 250 | 200 | 450 |
| | | | **8** | **24** | **0** | **32** | **20** | **350** | **350** | **700** |
| | | Total: | **61** | **72** | **1** | **134** | **98** | **2100** | **1200** | **3300** |

| Abbreviations | Course Full Name |
|---|---|
| CM | Major/Core Mandatory |
| ME | Major Elective |
| RM | Research Methodology |
| OJT | Industrial Internship/On Job Training |
| RP | Research Project |
| MOOC | MOOC Course |
| CIA | Continuous Internal Assessment/Evaluation |

Dr. Santosh Deshpande
BoS Member

Dr. Darshana Desai
HOD-MCA, BoS Chairman

Dr. Saurabh Gupta
Dean-Academics

Dr. Nilesh Uke
Principal, ICEM

INDIRA COLLEGE OF ENGINEERING & MANAGEMENT, PUNE-411048
Principal Office

| No. | MOOC-1 (3 Credits) | Platform | MOOC-2 (3 Credits) | Platform | MOOC-3 (2 Credits) | Platform |
|---|---|---|---|---|---|---|
| 1 | IBM AI Engineering Professional Certificate | Coursera | Certified Kubernetes Security Specialist (CKS) | Linux Foundation | AWS Certified Developer - Associate | AWS Training |
| 2 | Google Cloud Security Engineer | Google Cloud | Certified Blockchain Developer (Hyperledger) | Linux Foundation | Microsoft Certified: Azure Security Engineer Associate | Microsoft |
| 3 | UiPath RPA Developer Foundation | UiPath Academy | Automation Anywhere RPA Certification | Automation Anywhere University | Blue Prism RPA Developer Certification | Blue Prism University |
| 4 | Microsoft Power Platform Developer Associate (PL-400) | Microsoft | IBM Data Science Professional Certificate | Coursera | Salesforce Administrator Certification | Salesforce |
| 5 | JIRA Software Fundamentals | Atlassian University | Agile with JIRA | Coursera (Atlassian) | JIRA Administration Certification | Atlassian |
| 6 | Certified DevSecOps Professional (CDP) | Practical DevSecOps | AWS Certified Solutions Architect - Associate | AWS Training | Google Associate Cloud Engineer | Google Cloud |
| 7 | Software Testing with Selenium & JUnit | Udemy | ISTQB Certified Tester (CTFL) | ISTQB | Appium Mobile Automation Testing | Udemy |
| 8 | AWS Certified Advanced Networking - Specialty | AWS Training | Microsoft Certified: DevOps Engineer Expert (AZ-400) | Microsoft | Certified Ethical Hacker (CEH) | EC-Council |
| 9 | LoadRunner Performance Testing | Micro Focus | JMeter Performance Testing | Udemy | Postman API Testing | Postman Academy |
| 10 | AI for IT Operations (AIOps) Certification | Coursera (IBM) | DevOps with GitHub Actions & Docker | Udemy | Python for Automation Testing | Udemy |

Note: These are tentative MOOC Courses, Student can change with the approval from authorities.

Dr. Santosh Deshpande
BoS Member

Dr. Darshana Desai
HOD-MCA, BoS Chairman

Dr. Saurabh Gupta
Dean-Academics

Dr. Nilesh Uke
Principal, ICEM

Principal Office

| COURSE TITLE | JAVA PROGRAMMING | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA101 | COURSE CATEGORY | CM | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | 07-2024 | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This Advanced Java Programming course is designed to provide MCA students with a comprehensive understanding of advanced Java concepts, focusing on Object-Oriented Programming (OOP) features, multithreading, graphical user interface development with AWT and Swing, web technologies including Servlets and JSP, database connectivity using JDBC, and an introduction to the Spring and Hibernate frameworks. |
| **Course Objective** | 1. To deepen the understanding of advanced OOP & exception handling concepts in Java.<br>2. To enhance skills in multithreading and concurrency in Java.<br>3. To develop proficiency in building graphical user interfaces using AWT and Swing.<br>4. To understand and implement web applications using Servlets and JSP.<br>5. To establish a foundation in database connectivity with JDBC.<br>6. To introduce the core concepts of the Spring and Hibernate frameworks. |
| **Course Outcome** | At the end of the course, students will be able to:<br>**CO1**: Demonstrate the application of advanced Object-Oriented Programming (OOP) features in Java. **(Apply)**<br>**CO2**: Implement multithreading and manage concurrency in Java applications. **(Apply)**<br>**CO3**: Design and develop graphical user interfaces using AWT and Swing. **(Create)**<br>**CO4**: Perform database operations using Java Database Connectivity (JDBC). **(Apply)**<br>**CO5**: Develop dynamic web applications using Servlets and Java Server Pages (JSP). **(Create)**<br>**CO6**: Understand and utilize basic concepts of the Spring and Hibernate frameworks. **(Understand)** |

**Prerequisite :**

The prerequisite for an Advanced Java Programming subject is a foundational understanding of Java programming, including basic concepts of object-oriented programming (OOP), data structures, exception handling, and multithreading. Familiarity with core Java libraries and basic GUI development is also recommended.

| MODULE | sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Overview of OOP Features & Exception Handling** | | | |
| 1.1 Inheritances and Polymorphism 1.2Abstract Classes and Interfaces 1.3Inner Classes and Anonymous Classes 1.4 Exceptions & Errors 1.5 Types of Exception 1.6 Use of try, catch, finally, throw, throws in Exception Handling 1.7 In-built and User Defined Exceptions Checked and Un-Checked Exceptions 1.8 Java Collection Framework | 7 | 15% | **CO-1** |

| | | | |
|---|---|---|---|
| **MODULE 2: Multithreading and Concurrency** | | | |
| 2.1 Understanding Multi-Threaded Programming 2.2Thread Lifecycle 2.3 Thread Priorities2.4 Synchronizing Threads 2.5 Inter Communication of Threads 2.6 Deadlock | 4 | 10% | **CO-2** |
| **MODULE 3: AWT , Swing & JDBC (Java Database Connectivity)** | | | |
| 3.1 Introduction to Abstract Window Toolkit (AWT) 3.2Event Handling in AWT 3.3Introduction to Swing and Swing Components 3.4 Layout Managers 3.5 Building GUI Applications with AWT and Swing 3.6Introduction to JDBC, JDBC Architecture and API, 3.7 Connecting to Databases 3.8 Executing SQL Queries Prepared Statements and Callable Statements 3.9 ResultSet and Metadata | 7 | 15% | **CO-3** |
| **MODULE 4: Java Web Technologies (Servlets and JSP)** | | | |
| 4.1 Introduction to Servlets Servlet Lifecycle 4.2 Request and Response Handling 4.3 Session Management<br>4.4 Introduction to Java Server Pages (JSP), JSP Tags, Scriptlets, and Expression Language 4.5 MVC Architecture in Java Web Applications | 5 | 15% | **CO-4** |
| **MODULE 5: Spring** | | | |
| 5.1 Overview of Spring Framework 5.2 Spring Core Concepts 5.3 Dependency Injection (DI) and Inversion of Control (IoC) 5.4 Spring Beans and Bean Lifecycle 5.5Introduction to Spring MVC | 10 | 20% | **CO-5** |
| **MODULE 6: Hibernate framework** | | | |
| 6.1 Overview of Hibernate, Object-Relational Mapping (ORM) Concepts 6.2 Hibernate Architecture 6.3 Hibernate Configuration and Mapping,<br>6.4 CRUD Operations with Hibernate 6.5 Hibernate Query Language (HQL) | 12 | 25% | **CO-6** |

| | |
|---|---|
| **TEXTBOOKS** | |
| 1 | "Modern Java in Action" by Raoul-Gabriel Urma, Mario Fusco, and Alan Mycroft |
| 2 | "Spring in Action" by Craig Walls<br>”Java Persistence with Hibernate" by Christian Bauer, Gavin King, and Gary Gregory |
| **REFERENCE BOOKS** | |
| 1 | Java: The Complete Reference" by Herbert Schildt |
| 2 | Java 8 Programming, BlackBook, DreamTech Press, Edition 2015 |
| **E BOOKS** | |
| 1 | Website: https://www.oracle.com/java/technologies/javase/jdk13-archive-downloads.html |
| 2 | "Head First Java" by Kathy Sierra and Bert Bates Website: |
| 3 | https://www.oreilly.com/library/view/head-first-java/9780596009205/ |

| | |
|---|---|
| | **Online tutorials and resources:** |
| | Oracle Java Tutorials: https://docs.oracle.com/javase/tutorial/ |
| 1 | JavaTPoint: https://www.javatpoint.com/ |
| 2 | Baeldung: https://www.baeldung.com/ |
| 3 | |
| **MOOC** | |
| 1 | edX: Course Title: "Advanced Java Programming" Provider: Hong Kong University of Science and Technology (HKUST) |
| 2 | Udacity: Course Title: "Java Programming Basics" |

| COURSE TITLE | | JAVA PROGRAMMING LAB | | | CREDITS | 2 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA101L | COURSE CATEGORY | SEC | L-P-T | 0-4-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Attendance | Internal Exam | ESE |
| 40% | 20% | 40% | -- |

**Practical Assignment Questions**

1. Write a Java program that takes two integers as input from the user and performs division, handling division by zero and invalid input types.

2. Create a Java program that reads from a user-specified file, implementing exception handling for file not found and I/O errors.

**3.** Create a class hierarchy for animals. Design a base class Animal with properties like name and age. Then, create two subclasses: Dog and Cat. Each subclass should have a method sound() that returns the sound the animal makes.

4. Design a class hierarchy for bank accounts. Create a base class BankAccount with properties like accountNumber and balance. Then, create two subclasses: SavingsAccount and CurrentAccount. Implement methods to deposit and withdraw money, and override a method to display account details specific to each account type.

5. Develop a class hierarchy for geometric shapes. Create a base class Shape with a method area(). Then, implement two subclasses: Circle and Rectangle. Each subclass should have a constructor to initialize its dimensions and override the area() method to calculate the area of the shape.

**6.** Implement a Java program demonstrating the use of abstract classes and interfaces in a banking application scenario. Define classes Account (abstract class), SavingsAccount, and CurrentAccount implementing different interfaces for operations like deposit, withdraw, and calculateInterest.

7. Implement a Java program to demonstrate multithreading using the Runnable interface for printing numbers 1 to 10 using two threads.

**8.** Write a Java program that creates two threads. The first thread should print numbers from 1 to 10 with a delay of 500 milliseconds between each number. The second thread should print the letters from 'A' to 'J' with a delay of 700 milliseconds between each letter. Use the Thread class to create the threads.

**9.** Create a Java program that uses multiple threads to increment a shared counter. Implement a class Counter with a synchronized method increment() that increases the counter by 1. Create three threads that each increment the counter 1000 times. After all threads finish, print the final value of the counter to ensure it is correct.

**10.** Design a simple GUI application using Swing components that includes a JFrame with a JLabel, a JTextField, and a JButton. When the button is clicked, the text entered in the text field should be displayed in the label

- Create a JFrame.
- Add a JLabel to display instructions.
- Add a JTextField for user input.
- Add a JButton to trigger the action.
- Implement an ActionListener for the button to update the label with the text from the text field.

**11.** Experiment with different layout managers in Java to understand their behavior. Create a JFrame with multiple JButtons arranged using different layout managers such as BorderLayout, FlowLayout, GridLayout, and BoxLayout.

a. Create a JFrame.
b. Add multiple JButtons with different labels.
c. Use different layout managers for each button to observe their arrangement.

**12.** Develop a menu-driven GUI application using Swing components. The application should include a menu bar with options for File (with sub-options New, Open, Save, Save As, Exit) and Edit (with sub-options Cut, Copy, Paste). Implement basic functionalities for each menu option.

- Create a JFrame.
- Add a JMenuBar.
- Add JMenu items for File and Edit.
- Add JMenuItems for the sub-options under each menu.
- Implement ActionListeners for each menu item to perform the respective actions (e.g., display a dialog for New/Open, save a file for Save, exit the application for Exit, etc.).

13. Develop a Java program that demonstrates basic event handling using buttons. Create a JFrame with two buttons labeled "Button 1" and "Button 2". When "Button 1" is clicked, display a message saying "Button 1 clicked!" and when "Button 2" is clicked, display a message saying "Button 2 clicked!”

- Create a JFrame.
- Add two JButtons with labels "Button 1" and "Button 2".
- Implement ActionListeners for each button to handle the click events.
- Display appropriate messages when each button is clicked.

14. Develop a Java program that demonstrates custom events and listeners. Create a scenario where an alarm system is triggered when a button is pressed. Implement custom event classes and listeners to handle the alarm eventCreate a JFrame.

- Add a JButton labeled "Trigger Alarm".

| | |
|---|---|
| | • Define a custom event class (e.g., AlarmEvent) and a corresponding listener interface (e.g., AlarmListener).<br>• Implement the AlarmListener interface in a class responsible for handling the alarm event.<br>• Trigger the custom event when the "Trigger Alarm" button is pressed.<br>• Display a message or perform an action when the alarm event is triggered.<br>15. Develop a Java application to perform CRUD operations on a student database using JDBC.<br>• Create a database schema for a student table with fields like student_id, name, age, and grade.<br>• Establish a JDBC connection to the database.<br>• Write SQL queries to create the student table, insert sample data, update records, and delete records.<br>• Implement exception handling to manage SQL exceptions.<br>• Execute the Java program to demonstrate CRUD operations.<br>16.Create a Java program to demonstrate transaction management and rollbacks using JDBC.<br>• Establish a connection to a database that supports transactions.<br>• Write Java code to perform multiple SQL operations within a transaction, such as transferring funds between bank accounts.<br>• Implement commit and rollback operations based on specific conditions (e.g., if a transaction fails).<br>• Use SQL exceptions to handle errors and ensure data integrity.<br>• Execute the program and observe the effect of commit and rollback operations on the database.<br>17.Create a database schema named "University" with tables for storing student records.<br>d. Create a stored procedure named "getStudentById" that accepts a student ID as input and returns the corresponding student details.<br>e. Populate the student table with sample data.<br>f. Establish a JDBC connection to the "University" database.<br>g. Write a Java method to invoke the "getStudentById" stored procedure using CallableStatement.<br>h. Prompt the user to input a student ID.<br>i. Pass the input student ID to the CallableStatement as a parameter.<br>j. Execute the CallableStatement to retrieve the student details.<br>k. Display the retrieved student details (e.g., ID, name, age, etc.) to the user.<br>18. Develop a servlet that handles form submission from a web page. The servlet should extract form parameters (such as name, email, etc.), process them, and display the submitted data back to the user.<br>• Create a servlet class that extends HttpServlet.<br>• Implement the necessary methods (e.g., doGet or doPost) to handle HTTP requests.<br>• Read form parameters using the request object.<br>• Process the form data (e.g., validate inputs, perform calculations).<br>• Generate an HTML response to display the submitted data back to the user.<br>19. Develop a web application that includes user authentication using servlets and JavaServer Pages (JSP). Users should be able to log in with a username and password, and upon successful authentication, they should be redirected to a welcome pageCreate a servlet to handle user authentication. |

- Implement a login form using JSP.
- Use session management to keep track of authenticated users.
- Validate user credentials against a predefined set (e.g., in-memory storage or database).
- Upon successful authentication, redirect the user to a welcome page using JSP.

**20.** Create a dynamic web application for performing CRUD (Create, Read, Update, Delete) operations using servlets and JSP. The application should allow users to interact with a database to manipulate data records.

- Design a database schema for storing data records (e.g., user information, product details).
- Implement servlets to handle CRUD operations (e.g., adding new records, retrieving records, updating records, deleting records).
- Develop JSP pages to interact with users (e.g., display data, input forms for adding/updating records).
- Use JDBC (Java Database Connectivity) to connect to the database and perform database operations.
- Implement error handling and validation for user inputs.

21. Develop a simple Java application to demonstrate the usage of Spring IOC container and Dependency Injection (DI) features.

- Configure a Spring IOC container using XML-based configuration.
- Define two POJO classes: Employee and Address, with appropriate attributes and methods.
- Implement Dependency Injection using Setter Injection to inject Address object into the Employee class.
- Write a Java program to retrieve an Employee object from the Spring IOC container and display its details along with the associated Address.
- Test the application to ensure proper DI and object creation.

22. Implement a simple Java application using Spring Framework that demonstrates Dependency Injection (DI) using constructor injection.

**Instructions:**

1. Create an interface MessageService with a method sendMessage().
2. Create a class EmailService implementing MessageService that prints "Email message sent".
3. Create a class SMSService implementing MessageService that prints "SMS message sent".
4. Create a class MessageProcessor that depends on MessageService for sending messages.
5. Configure Spring to inject EmailService into MessageProcessor using constructor injection.
6. Test the application by creating an instance of MessageProcessor in main method and invoking sendMessage().

23. Create a Java application using Hibernate to perform CRUD operations on a Student entity.

| | |
|---|---|
| | **Instructions:**<br><br>1. Define a Student entity with fields id, name, email, and age.<br>2. Configure Hibernate to connect to a database (MySQL or H2).<br>3. Implement methods to perform CRUD operations:<br>o createStudent(Student student)<br>o readStudent(int studentId)<br>o updateStudent(Student student)<br>o deleteStudent(int studentId)<br>4. Test the CRUD operations by creating instances of Student and invoking these methods.<br><br>24. Develop a Spring MVC application to handle a simple "Hello World" request-response.<br><br>**Instructions:**<br><br>1. Create a controller HelloController with a method sayHello() mapped to URL /hello.<br>2. Configure Spring MVC to handle this request and respond with a view displaying "Hello, World!".<br>3. Implement a simple JSP view hello.jsp that displays the greeting message.<br>4. Test the application by accessing http://localhost:8080/hello in web browser.<br><br>25. Create a Java application using Hibernate to perform a CRUD operation using Hibernate Query Language (HQL).<br><br>**Instructions:**<br><br>1. Define a Product entity with fields id, name, price, and quantity.<br>2. Implement methods to:<br>o Insert new Product objects into the database.<br>o Retrieve all Product objects using HQL.<br>o Update a Product object.<br>o Delete a Product object by ID.<br>3. Test the CRUD operations by creating instances of Product and invoking these methods. |

Note:- Additional lab experiments will be also given time to time according to topic as per the syllabus

| COURSE TITLE | DATA STRUCTURES & ALGORITHMS | | CREDITS | 3 |
|---|---|---|---|---|
| COURSE CODE | MCA102 | COURSE CATEGORY / CM | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | 07-2024 | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides an in-depth understanding of data structures and algorithms using JavaScript. Students will learn to apply various algorithmic approaches, including greedy algorithms, dynamic programming, and divide and conquer, to solve real-world problems. By the end of the course, students will be proficient in implementing and analyzing data structures and algorithms, ensuring efficient optimized performance in JavaScript. |
| **Course Objective** | • To understand the core principles of data structures and their importance in software development.<br>• To explore and implement various data structures (arrays, linked lists, stacks, queues, trees, graphs) using JavaScript.<br>• To develop an ability to solve problems using algorithmic approaches such as greedy algorithms, dynamic programming, and divide and conquer.<br>• To enhance proficiency in memory management and understand the performance trade-offs of different data structures.<br>• To prepare students to write efficient code for competitive programming and real-time applications in JavaScript. |
| **Course Outcome** | After end of this course student will be able:<br>• **CO1**: Understand and implement fundamental data structures such as arrays, linked lists, stacks, and queues in JavaScript.<br>• **CO2**: Apply sorting and searching algorithms (e.g., binary search, quick sort) and evaluate their performance using time and space complexity.<br>• **CO3**: Design and analyze advanced data structures like trees, graphs, heaps, and hash tables for solving complex problems.<br>• **CO4**: Develop efficient algorithms using different paradigms (greedy algorithms, dynamic programming, etc.) for problem-solving in JavaScript.<br>• **CO5**: Optimize solutions for real-world applications by understanding memory management and algorithmic efficiency in JavaScript. |

**Prerequisites:**

- Basic knowledge of programming languages, preferably C++.
- Understanding of fundamental data structures like arrays, linked lists, stacks, and queues.
- Familiarity with recursion and basic sorting/searching algorithms.

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Arrays** | | | |
| **Topic:** 1.1 Introduction & Definition of an Array, 1.2 Memory Allocation & Indexing in JavaScript, 1.3 Operations on 1-D & 2-D Arrays (Array of Arrays), 1.4 Dynamic Arrays using JavaScript (Array Object methods like push(), pop(), etc.), 1.5 Memory | 8 | 10% | CO-1 |

| | | | |
|---|---|---|---|
| management and Garbage Collection in JavaScript | | | |
| **MODULE 2: Linked List** | | | |
| **Topic:** 2.1 Introduction & Definition of a Linked List, 2.2 Memory Allocation in JavaScript, 2.3 Types of Linked Lists (Singly, Doubly, Circular), 2.4 Operations on Singly Linked Lists (Insert, Delete, Search, Traverse), 2.5 Circular & Doubly Linked Lists (with JavaScript implementation) | **6** | **15%** | **CO-2** |
| **MODULE 3: Stacks and Queues** | | | |
| **Topic:** 3.1 Introduction and Definition of a Stack, 3.2 Implementation of Stacks using Arrays and Linked Lists (JavaScript code), 3.3 Applications of Stacks (Expression Conversion, String Reversal using JavaScript), 3.4 Introduction and Definition of a Queue, 3.5 Implementation of Queues using Arrays and Linked Lists (JavaScript code), 3.6 Advanced Queues (Priority Queue, Deque), 3.7 Stack and Queue implementations using JavaScript (Classes, Functions) | **6** | **15%** | **CO-3** |
| **MODULE 4: Trees and Graphs** | | | |
| **Topic:** 4.1 Tree Definition and Representation, 4.2 Binary Search Tree (BST) and its operations (Insertion, Deletion, Search), 4.3 Tree Traversal (Inorder, Preorder, Postorder), 4.4 AVL Tree and its Rotations (with code examples in JavaScript), 4.5 Directed and Undirected Graphs, 4.6 Graph Representations (Adjacency Matrix, List), 4.7 Graph Traversals (BFS, DFS with JavaScript), 4.8 Advanced Trees (Red-Black Trees, B-Trees), 4.9 Graph Algorithms (Dijkstra, Prim, Kruskal in JavaScript) | **10** | **25%** | **CO-3 and CO-4** |
| **MODULE 5: Searching and Sorting** | | | |
| **Topic:** 5.1 Linear Search (with examples in JavaScript), 5.2 Binary Search (with examples in JavaScript), 5.3 Interpolation Search (JavaScript implementation), 5.4 Merge Sort, 5.5 Quick Sort, 5.6 Bubble Sort (All in JavaScript), 5.7 Heaps (Min and Max), 5.8 Hash Tables (Implementation of Hash Maps in JavaScript using Objects and `Map`), 5.9 Hash Functions (Designing custom hash functions in JavaScript) | **10** | **25%** | **CO-4 and CO-5** |
| **MODULE 6: Advanced Algorithms and Optimization** | | | |
| **Topic:** 6.1 Greedy Algorithms, 6.2 Dynamic Programming (JavaScript-based solutions), 6.3 Divide and Conquer (JavaScript examples), 6.4 NP-Complete Problems, 6.5 Approximation Algorithms, 6.6 Optimization Techniques for Competitive Programming (focusing on JavaScript) | **5** | **10%** | **CO-5** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | **"Data Structures and Algorithms with JavaScript" by Michael McMillan** https://www.amazon.com/Data-Structures-Algorithms-Michael-McMillan/dp/1449364934 |
| 2 | **"JavaScript: The Good Parts" by Douglas Crockford** https://www.amazon.com/JavaScript-Good-Parts-Douglas-Crockford/dp/0596517742 |
| **REFERENCE BOOKS** | |
| 1. | **"Eloquent JavaScript" by Marijn Haverbeke** https://eloquentjavascript.net |

| | |
|---|---|
| 2. | **"Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein**<br>https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844 |
| **E BOOKS** | |
| 1. | **"JavaScript Data Structures and Algorithms" by Sammie Bae**<br>**https://opendatastructures.org/ods-cpp.pdf** |
| 2 | **"Open Data Structures (in pseudocode)" by Pat Morin**:<br>https://opendatastructures.org/ods-python.pdf |
| **MOOC** | |
| 1. | **"Algorithms and Data Structures" on edX**<br>https://www.edx.org/course/algorithm-and-data-structures |
| 2. | **NPTEL Data Structures and Algorithms**<br>https://archive.nptel.ac.in/courses/106/102/106102064/ |

| COURSE TITLE | | DATA STRUCTURES & ALGORITHMS LAB | | | CREDITS | 2 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA102L | COURSE CATEGORY | SEC | L-P-T | 0-4-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Arrays**

**1: Dynamic Inventory Management System:** Design an inventory management system for a warehouse using a **1-D array**. The array should store product details (ID, name, quantity, and price). Implement operations for adding new products, updating inventory levels, and calculating the total inventory value dynamically.

**2: Real-time Seating Arrangement**: Create a **2-D array** to represent seating arrangements for a movie theater. The system should handle seat booking and cancellation requests dynamically. Add constraints for group bookings, ensuring they are seated together.

**3: Traffic Data Analysis**: Collect real-time traffic data (e.g., car counts, speed, and entry/exit times) on multiple road lanes using **a dynamic array**. Analyze traffic flow patterns for peak hours using array operations like sorting, searching, and aggregation.

**4: Multi-Dimensional Data Representation**: Design a **multi-dimensional array** to represent geographical data (e.g., latitude, longitude, altitude) for a drone-based mapping system. Implement functions to extract and manipulate data for different regions dynamically.

**MODULE 2: Linked List**

**1: Task Scheduler Simulation:** Implement a **priority-based task scheduler** using a doubly linked list. Each task has a priority and duration. When new tasks are added, they should be inserted at the correct position based on priority. Simulate task execution and removal once completed.

**2: Train Route Simulation (Circular Linked List):** Simulate a train system where each station is represented as a node in a **circular linked list**. The train moves around the stations in a loop. Add operations to dynamically insert new stations or remove old ones, with proper memory management.

**3: Real-Time Collaborative Editor:** Build a **version control system** for a real-time collaborative text editor using a doubly linked list. Each edit is a node, allowing users to navigate forwards and backwards through changes. Include an undo/redo feature, tracking the position of multiple collaborators.

**4: Browser Tabs Management:** Implement browser tab management where each open tab is a node in a **singly linked list**. Users can open new tabs, close specific ones, and switch between them. Optimize the solution for handling thousands of open tabs efficiently.

**5: Music Streaming Queue:** Develop a **dynamic playlist system** where songs are represented as nodes in a **circular doubly linked list**. The user can queue songs, skip, replay, and move to the next song seamlessly. Ensure the playlist can handle thousands of songs with minimal memory overhead.

**6: Memory Block Allocation (Garbage Collection)**: Simulate **dynamic memory block allocation and deallocation** using a **singly linked list**, where each node represents a memory block. Implement garbage collection to identify and release unreferenced blocks periodically

## MODULE 3: Stacks and Queues

**1: Expression Evaluation (Infix to Postfix Conversion):** Implement a **calculator** that converts **infix expressions to postfix notation** using stacks. Evaluate the postfix expression to return the result. Handle complex expressions with parentheses and operator precedence efficiently.

**2: Online Ticketing System (Priority Queue)**: Design an **online ticketing system** using a **priority queue** where VIP customers are served first. Regular customers are served based on their order of arrival. Simulate ticket booking, cancellation, and serve operations, ensuring the system works under heavy traffic conditions.

**3: Undo-Redo Functionality for a Code Editor**: Create an **undo-redo feature** using two stacks to track changes made in a code editor. As the user performs actions (e.g., writing, deleting text), track each action and allow them to undo or redo changes.

**4: Job Queue System**: Simulate a **job processing system** where jobs (like printing documents) are queued. Implement the queue with the ability to dynamically prioritize certain jobs (e.g., emergency print requests) using a **priority queue**.

**5: Stock Span Problem**: Solve the **Stock Span Problem** using a **stack**, where for each day's stock price, you calculate the number of consecutive days the price was less than or equal to today's price.

**6: Bank ATM Queue Simulation**: Implement a **bank ATM queue** where customers are queued for transactions. Simulate different types of transactions (deposit, withdrawal, balance check) with varying processing times. Use a **deque** (double-ended queue) to allow priority transactions at either end

## MODULE 4: Trees and Graphs

**1: Organizational Hierarchy Management System**: Implement an **organization's hierarchy** using a **tree structure** where each node represents an employee. Simulate promotions, new hires, and removals dynamically, ensuring the tree stays balanced.

**2: E-Commerce Recommendation System (Binary Search Tree)**: Build an **e-commerce recommendation system** where products are stored in a **binary search tree (BST)** based on customer ratings. Implement operations to find products within a specific rating range and suggest similar products.

**3: Social Network Friend Recommendation (Graph)**: Use a **graph** to represent connections between users in a social network. Implement a **BFS algorithm** to suggest friend recommendations based on mutual connections.

**4: Shortest Path in a City (Graph)**: Given a city represented as a **graph** with road networks (nodes for intersections, edges for roads), use **Dijkstra's algorithm** to find the shortest path between any two intersections.

**5: File System Management (Tree)**: Simulate a **file system** where directories and files are stored in a **tree structure**. Implement operations like creating new files, deleting files, and listing files in different traversal orders (pre-order, post-order, in-order).

**6: AVL Tree for Stock Price Management**: Use an **AVL tree** to maintain stock prices. Ensure that after each insertion, the tree remains balanced by performing rotations.

**7: Graph Coloring Problem (Greedy)**: Solve the **graph coloring problem** using a **greedy algorithm** to minimize the number of colors needed to color a graph such that no two adjacent nodes share the same color.

**8: Minimum Spanning Tree for a Power Grid**: Implement **Kruskal's algorithm** to find the **minimum spanning tree (MST)** for a power grid system connecting cities. Each city is a node, and each connection between cities has a cost.

**9: Red-Black Tree for Dynamic Leaderboard**: Implement a **red-black tree** to manage a dynamic gaming leaderboard. As players gain points, their rank in the tree adjusts in real time.

**10: Cycle Detection in Graph**: Implement a graph traversal algorithm (DFS) to detect cycles in a directed and undirected graph, simulating dependencies between software modules.

## MODULE 5: Searching and Sorting

**1: E-commerce Product Search with Binary Search**: Implement a **binary search** algorithm to search for products in a sorted product catalog. Compare its performance against **linear search**.

**2: Contact List Sorting (Merge Sort)**: Sort a large list of phone contacts using **merge sort** and compare the time complexity with **quick sort** when applied to smaller lists.

**3: Event Ranking System (Heap Sort)**: Implement **heap sort** to rank participants in a large-scale competition based on their scores. Test your solution with large datasets.

**4: Efficient Storage using Hash Tables**: Design a **hash table** to store and retrieve employee records based on employee IDs. Implement different **hash functions** and collision handling techniques (chaining, open addressing).

**5: Searching in a Rotated Sorted Array**: Solve the problem of searching for a specific element in a **rotated sorted array** using a modified **binary search** algorithm.

**6: Sorting a Music Library (Quick Sort)**: Implement **quick sort** to arrange songs in a music library by different parameters (duration, artist, genre). Optimize the algorithm for large datasets.

**7: Caching using LRU Cache**: Implement an **LRU (Least Recently Used) Cache** system using a combination of **hash maps** and **doubly linked lists** to store frequently accessed data efficiently.

**8: Dictionary Implementation with Hashing**: Create a **dictionary** where words are stored using a **hash table**. Implement efficient lookup, insertion, and deletion operations using custom **hash functions**.

**9: Inventory Search using Interpolation Search**: Implement an **interpolation search** algorithm for finding items in an inventory management system where the data distribution is uniform. Compare its performance with binary and linear search algorithms.

**10: Sorting Patient Data in a Hospital**: Design an algorithm to sort patient data based on emergency levels using **heap sort**. Ensure that the sorting happens in real-time for critical situations in an emergency room.

## MODULE 6: Advanced Algorithms and Optimization

**1: Delivery Route Optimization (Greedy Algorithm)**: Solve the **delivery route optimization** problem for a delivery service using a **greedy algorithm**. Minimize the total distance traveled by the delivery driver to deliver packages to multiple destinations.

**2: Knapsack Problem (Dynamic Programming)**: Solve the **0/1 Knapsack Problem** using **dynamic programming**, where you are given a set of items, each with a weight and value, and must determine the most valuable combination that can fit within a weight limit.

**3: Divide and Conquer Approach for Matrix Multiplication**: Implement a **divide and conquer** algorithm (Strassen's algorithm) for **matrix multiplication**. Compare its performance with the standard matrix multiplication algorithm for large matrices.

**4: Approximation Algorithms for NP-Complete Problems**: Implement an **approximation algorithm** for solving the **traveling salesman problem**. Analyze how close the solution is to the optimal path and discuss the complexity of the algorithm

| COURSE TITLE | OBJECT ORIENTED SOFTWARE ENGINEERING | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA103 | COURSE CATEGORY | CM | L P T | 3-0-0 |
| Version 1.0 | Approval Details | 01-07-2024 | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments /Case Study Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides an in-depth understanding of object-oriented software engineering principles, methodologies, and tools. It focuses on the complete software development lifecycle with an emphasis on object-oriented techniques and their practical application in real-world scenarios. |
| **Course Objective** | • Understand the fundamental concepts of object-oriented software engineering.<br>• Learn and apply object-oriented analysis and design methodologies.<br>• Utilize UML for modeling software systems.<br>• Enhance problem-solving skills using object-oriented approaches.<br>• Gain practical experience through case studies and project work. |
| **Course Outcome** | After end of this course student will be able:<br>• CO1: Demonstrate understanding of object-oriented concepts and principles.<br>• CO2: Apply object-oriented analysis and design methodologies to software development.<br>• CO3: Utilize UML for modeling various aspects of software systems.<br>• CO4: Design user interfaces and user experiences using modern techniques.<br>• CO5: Implement agile methodologies in software projects |

**Prerequisites: Students** should have a basic understanding of programming concepts and data structures. Prior knowledge of object-oriented programming languages such as Java or C++ would be beneficial.

| Module, Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to OOSE** | | | |
| 1.1 Overview of Software Engineering 1.2 Evolution of Software Engineering 1.3 Object-Oriented Concepts 1.4 Benefits of Object-Oriented Approach 1.5 Software Development Lifecycle and Types | **5** | **10%** | **CO-1** |
| **MODULE 2: Requirements Analysis and Design** | | | |

| | | | |
|---|---|---|---|
| 2.1 Requirements elicitation and analysis 2.2 Requirement Engineering 2.3 Types of Requirements: Functional and Nonfunctional 2.4 Four Phases of Requirement Engineering 2.5 Software requirement Specification (SRS) Structure and contents of SRS 2.6 IEEE standard format SRS | **9** | **20%** | **CO-2** |
| **MODULE 3:** Use-case Driven Object-Oriented Analysis | | | |
| 3.1 Introduction to oops concepts 3.2 Class and object - Abstraction and encapsulation -Method and messages - Interface, Inheritance and polymorphism Use case modeling 3.3 Object-oriented design principles 3.4 Structural Diagram - Class Diagram, Associations and links, Aggregation, Composition and containment- Inheritance, Sub Types and IS-A hierarchy 3.5 Behavioral Diagram - Use case Diagram, Sequence diagram 3.6 State modeling diagram - Activity Diagram | **12** | **30%** | **CO-2 & CO-3** |
| **MODULE 4:** SOFTWARE DESIGN Pattern | | | |
| 4.1 Software Design process 4.2 Design concepts Coupling & Cohesion 4.3 Functional independence 4.4 Design patterns 4.5Creational Patterns (Factory, Singleton) 4.6 Structural Patterns (Adapter, Decorator) 4.7 Behavioral Patterns (Observer, Strategy) | **9** | **20%** | **CO-3, CO-4** |
| **MODULE 5:** User Interface UI/UX Design | | | |
| 5.1 Elements of good design 5.2 Eight golden rules for design 5.3 User interface design-Case Study UI/U X5.4 Interactive UI design 5.5 Introduction to UI and UX- Principles of User Interface Design 5.6 Interaction Design 5.7 Usability and User Experience 5.8 Wireframing and Prototyping - Evaluation and Testing of UI/UX | **6** | **10%** | **CO-4** |
| **MODULE 6:** Agile Methodology | | | |
| 6.1 Introduction to Agile Methodologies- Agile Manifesto and Principles 6.2 Scrum Framework - Roles: Product Owner, Scrum Maste 6.3 Development Team 6.4 Agile Planning: Release and Sprint Planning 6.5 Agile Metrics: Burndown Charts, Velocity 6.6 Agile Development Practices: TDD 6.7 Continuous Integration | **4** | **10%** | **CO-5** |

**TEXT BOOKS**

| | |
|---|---|
| 1. | "Object-Oriented Software Engineering: An Agile Unified Methodology" by David Kung |
| 2. | "Software Engineering: A Practitioner's Approach" by Roger S. Pressman |
| 3. | "Object-Oriented Analysis and Design with Applications" by Grady Booch, Robert A. Maksimchuk, Michael W. Engle, Bobbi J. Young, Jim Conallen, and Kelli A. Houston |

**REFERENCE BOOKS**

| | |
|---|---|
| 1. | "UML Distilled: A Brief Guide to the Standard Object Modeling Language" by Martin Fowler |
| 2. | "Clean Code: A Handbook of Agile Software Craftsmanship" by Robert C. Martin |
| 3. | "Design Patterns: Elements of Reusable Object-Oriented Software" by Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides |

| E BOOKS | |
|---|---|
| 1. | https://www.geeksforgeeks.org/software-design-patterns/ |
| 2 | Design Patterns: Elements of Reusable Object-Oriented Software by Erich Gamma et al.: [https://example.com/design-patterns] |
| 3. | "The UX Book: Process and Guidelines for Ensuring a Quality User Experience" by Rex Hartson and Pardha Pyla (E-book) |
| **MOOC** | |
| 1. | Coursera - Object-Oriented Software Engineering: https://example.com/coursera-oose |
| 2. | edX - Software Engineering Essentials: https://example.com/edx-soft-eng |
| 3. | "Introduction to User Experience Design" by Georgia Institute of Technology on Coursera |
| 4. | "Software Development Processes and Methodologies" by the University of Minnesota on Coursera |

| COURSE TITLE | Operating System & Shell Programming | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA104 | COURSE CATEGORY | CM | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | 07-2024 | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz / Activity Based | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course will introduce you to modern operating systems. We will focus on UNIX-based operating systems, though we will also learn about alternative operating systems, including Windows. The course will begin with an overview of the structure of modern operating systems. Over the course of the subsequent units, we will discuss the history of modern computers, analyze in detail each of the major components of an operating system (from processes to threads), and explore more advanced topics in the field, including memory management and file input/output. The class will conclude with a discussion of various system-related security issues. |
| **Course Objective** | 1. To learn the fundamentals of Operating Systems and handle processes and threads and their communication<br>2. To learn the mechanisms involved in memory management in contemporary OS<br>3. To know the functionality of Multiprocessor OS and Mobile OS.<br>4. To gain knowledge of distributed operating system concepts.<br>5. To learn about the Basics of Linux.<br>6. To learn programmatically to implement Linux OS mechanisms. |
| **Course Outcome** | CO1: **Understand** structure of OS, process management and synchronization.<br>CO2: **Understand** multicore and multiprocessing OS.<br>CO3: **Analyze** Real-time and embedded OS<br>CO4: **Apply** Windows and Linux OS fundamentals and administration<br>CO5: **Apply** to solve shell scripting problems |

**Prerequisites:** Basics of Operating system

| Module, Topic | Sessions | Weightage (%) | Co-Mapping |
|---|---|---|---|
| **MODULE 1: Overview of OS** | | | |
| 1.1 Overview of operating systems, 1,2 Functionalities and Characteristics of OS , 1.3 Hardware concepts related to OS ,1.4 CPU states, I/O channels, 1.5 Memory Management, Memory Management Techniques, 1.6 Contiguous & Non-Contiguous allocation, Logical & Physical Memory - Conversion of Logical to Physical address, 1.7 Paging, 1.8 Demand Paging Page Replacement Concept, 1.9 Segmentation - Segment with paging 1.10 Virtual Memory Concept, Thrashing | 7 | 15% | **CO-1** |
| **MODULE 2: Process Management** | | | |

| | | | |
|---|---|---|---|
| 2.1 Process Management and Synchronization, 2.2 PCB, 2.3 Job and processor scheduling. Scheduling Concept, 2.4 Process hierarchies, 2.5 Problems of concurrent processes, 2.6 Critical sections, Mutual exclusion Synchronization, 2.7 Deadlock, 2.8 Device and File Management, 2.9 Overview Techniques, File Systems | 8 | 17% | **CO-2** |
| **MODULE 3: Processor Management** | | | |
| 3. Multiprocessor and Multicore Operating Systems, 3.1 Introduction, Advantages and Disadvantages, Multicore System Vs. Multiprocessor System, 3.2 Types of Multiprocessors , Symmetric Multiprocessors, Asymmetric Multiprocessors 3.3 Basic Multicore Concepts: Memory Sharing Styles, Uniform Memory Access (UMA), Non-Uniform Memory Access (NUMA), No Remote Memory Access, (NORMA),3.4 Cache Coherence, Inter-Process and intercore Communication: Shared Memory, Message Passing, 3.5 Mobile Operating Systems, Concept Need and Features, Types of Mobile OS Overview of Android OS, Applications of Mobile OS 3.6 Distributed Operating Systems, Concept Need and Features, Examples of Distributed OS with brief introduction, Applications of Distributed OS | 8 | 17% | **CO2, CO-3** |
| **MODULE 4: Real Time OS** | | | |
| 4. Real Time OS,4.1 Introduction and use of RTOS, 4.2 Components of RTOS, Types of RTOS, Features of RTOS, Factors for selecting in RTOS, 4.3 Applications of RTOS, Disadvantages of RTOS, 4.4 Embedded OS, 4.5 Concept Need and Features of embedded OS, 4.6 Examples of embedded OS with brief introduction, 4.7Applications of embedded OS | 4 | 10% | **CO-3** |
| **MODULE 5: Types of OS** | | | |
| 5.Windows OS and Windows Server, Architecture,5.1 Windows OS , Introduction, Windows OS Installation, Process Management, Control Panel Overview, Users, Security and Privacy Settings, Identify Accessibility Settings5.2 Service Management, Syncing Devices and File Sharing5.3 Windows Utilities (Accessories, Disk, Management, Resource Monitor, Backup, and Recovery), Basic Troubleshooting (Networking, Security, Device Driver). Introduction to Ubuntu, Introduction, Overview of Kernel,5.3 Installation of Ubuntu File system, Basic Commands of Linux, Managing Processes in Linux, 5.4 Installing and deleting software packages, User Management, File and Device Management, Backup and recovery,5.5 Introduction to Graphical Environment (GNOME), Ubuntu Utilities, (Virtual Box, Evolution, Gimp, Bleach Bit, Unity Tweak Tool etc.), SAMBA Overview | 12 | 25% | **CO-4** |
| **Module 6: Shell Scripting** | | | |
| 6. Linux Shell Scripting,6.1 Introduction,6.2 Variables, Flow Controls, Loops,6.3 Functions, 6.4 Lists,6.5 Manipulating Strings 6.6 Reading and Writing Files,6.7 Positional Parameters 6.8 Case statement,6.9 Real time scripts for different system administration activities | 6 | 16% | **CO-5** |
| **TEXTBOOKS** | | | |

| | |
|---|---|
| 1 | Silberschatz, A., Galvin, P.B. and Gagne, G., Operating System Concepts (10 ed.), John Wiley, 2018. ISBN 978-1-119-32091-3 |
| 2 | Stallings William, Operating Systems Internals and Design Principles (9 ed.), Prentice Hall, 2021. ISBN 978-0134670959 |
| 3 | Abraham Silberschatz, Peter Baer Galvin, Greg Gagne (2006), Operating System Principles, 7th edition, Wiley India Private Limited, New Delhi. |
| **REFERENCE BOOKS** | |
| 1 | Andrew S Tanenbaum and Herbert Bos, Modern Operating Systems (1 ed.), Pearson, 2021. ISBN 9789332575776. |
| 2 | Stallings (2006), Operating Systems, Internals and Design Principles, 5th edition, Pearson |
| 3 | Education, India. |
| 4 | Andrew S. Tanenbaum (2007), Modern Operating Systems, 2nd edition, Prentice Hall of India, |
| 5 | "Operating Systems: Internals and Design Principles" by William Stallings. |
| | P.C.P. Bhatt: Introduction to Operating Systems Concepts and Practice, 3rd Edition, PHI, 2010 |
| **E BOOKS** | |
| 1 | Introduction to operating system https://nptel.ac.in/courses/106/106/106106144/ |
| 2 | https://www.freebookcentre.net/ComputerScience-Books-Download/Operating-System-Notes.html |
| 3 | https://www.freebookcentre.net/ComputerScience-Books-Download/Introduction-to-Operating-Systems-Lectures.html |
| **MOOC/Coursera** | |
| 1 | https://www.coursera.org/learn/os-power-user |
| 2 | https://www.coursera.org/specializations/codio-introduction-operating-systems |
| 3 | https://www.coursera.org/learn/comparch |

| COURSE TITLE | MATHEMATICS AND BUSINESS STATISTICS | | | CREDITS | 2 |
|---|---|---|---|---|---|
| COURSE CODE | MCA105 | COURSE CATEGORY | CM | L P T | 2-0-0 |
| Version 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10 % | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides an integrated introduction to key mathematical and statistical concepts essential for data analysis and problem-solving in the field of computer applications. Designed for MCA students, it combines foundational mathematical techniques with statistical methods commonly used in business and technology contexts. Students will develop skills to analyze data, apply mathematical models, and use statistical tools for decision-making and forecasting. |
| **Course Objective** | 1. Develop a strong understanding of fundamental mathematical concepts and their applications.<br>2. Gain proficiency in descriptive and inferential statistics relevant to business and technology.<br>3. Apply mathematical and statistical methods to real-world problems in computer science and business. |
| **Course Outcome** | • **CO1: Apply** fundamental mathematical techniques such as algebra and functions to solve complex problems related to computer applications and data analysis.<br>• **CO2: Demonstrate** proficiency in descriptive statistics by effectively summarizing and visualizing data using appropriate statistical methods and tools.<br>• **CO3: Utilize** probability theory and distributions to model and analyze random processes, assess risks, and make informed decisions based on probabilistic data.<br>• **CO4: Perform** inferential statistical analyses, including hypothesis testing and confidence interval estimation, to draw valid conclusions and make data-driven decisions.<br>• **CO5: Implement** regression analysis to explore and quantify relationships between variables, applying these techniques in practical business scenarios to support decision-making and forecasting. |

**Prerequisites:** Fundamental knowledge of algebra, basic statistics, introductory probability, and basic calculus.

| **MODULE 1: Fundamentals of Mathematics** | | | |
|---|---|---|---|
| 1.1 Algebraic expressions 1.2 solving linear equations, matrix operations (addition, subtraction, multiplication, inversion), determinants, 1.3 systems of linear equations.<br>1.4 Types of functions (linear, quadratic, exponential, logarithmic), 1.5 graphical representation and interpretation, 1.6 transformations of functions, optimization problems. | **6** | **20%** | **CO-1** |
| **MODULE 2: Descriptive Statistics** | | | |

| | | | |
|---|---|---|---|
| 2.1 Methods of data collection (surveys, experiments, observational studies) 2.2 types of data (qualitative, quantitative) 2.3 data organization (frequency tables, cross-tabulations). 2.4 Calculating and interpreting mean, median, mode, variance, standard deviation, range, Coefficient of Variance 2.5 Constructing and interpreting histograms, bar charts, pie charts, box plots, scatter plots. | **6** | **20%** | **CO-2** |
| **MODULE 3: Probability and Distributions** | | | |
| 3.1 Basic probability rules (addition and multiplication rules), conditional probability, Bayes' theorem 3.2 probability of combined events (independent and dependent events).<br>3.3 Discrete distributions (Binomial distribution), continuous distributions (Normal distribution) 3.4 properties and applications of these distributions, Skewness and Kurtosis.<br>3.5 Definition of random variables, expected value, variance, covariance, probability mass and density functions. | **6** | **20%** | **CO-3** |
| **MODULE 4: Inferential Statistics** | | | |
| 4.1 Formulating null and alternative hypotheses 4.2 Type I and Type II errors, significance levels, p-values 4.3 hypothesis tests for means (z-test, t-test), and proportions.<br>4.4 Constructing confidence intervals for population means and proportions 4.5 interpretation of intervals, margin of error. | **6** | **20%** | **CO-4** |
| **MODULE 5: Regression Analysis** | | | |
| 5.1 Model formulation, least squares estimation, interpretation of regression coefficient 5.2 goodness-of-fit measures (R-squared, adjusted R-squared).<br>5.3 Using regression analysis for forecasting, trend analysis, and decision-making 5.4 practical case studies and examples. | **6** | **20%** | **CO-5** |

| **TEXT & REFERENCE BOOKS** | |
|---|---|
| 1. | **"Discrete Mathematics and Its Applications"** by Kenneth H. Rosen |
| 2. | **"Business Statistics: A First Course"** by David M. Levine, Kathryn A. Szabat, and David F. Stephan |
| 3. | **"Calculus: Early Transcendentals"** by James Stewart |
| **E BOOKS** | |
| 1. | **"Discrete Mathematics and Its Applications"** Kenneth H. Rosen |
| 2. | **"Business Statistics: A First Course"** David M. Levine, Kathryn A. Szabat, David F. Stephan |
| **MOOC** | |
| 1. | **Introduction to Statistics** Platform: Coursera |
| 2. | **Mathematics for Data Science**Platform: edX |
| 3. | **Data Science and Machine Learning Bootcamp with R** Udemy |

| COURSE TITLE | Web Technologies | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA106 | COURSE CATEGORY | ME 1.2 | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | 18-4-2025 | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz/ Activity Based | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course is designed to build your skills in web development and design, no matter how little experience or technical knowledge you currently have. You probably visit several websites every day, whether for business, entertainment or education. But have you ever wondered how these websites actually work? How are they built? How do browsers, computers, and mobile devices interact with the web? What skills are necessary to build a website? With a billion websites now on the internet, the answers to these questions |
| **Course Objective** | 1. To impart the design, development and implementation of Dynamic Web Pages.<br>2. To implement the Latest properties of CSS3<br>3. To implement the Concept of Web servers<br>4. To develop programs for Web Javascript<br>5. To design and implement dynamic websites with good sense of designing and latest technical aspects. |
| **Course Outcome** | **CO1: Understand** Outline the basic concepts of Advance Internet Technologies **CO2: Apply** Design appropriate user interfaces and implements webpage based on given problem Statement<br>**CO3: Apply** Implement concepts and methods of using web servers<br>**CO4: Analyze** Implement concepts and methods of JS and Ajax<br>**CO5: Apply** Build Dynamic web pages using Bootstrap, HTML and CSS and PHP |

**Prerequisites:** Student must have basic working knowledge of HTML, CSS, JavaScript, web development tool

| **MODULE 1: Introduction to HTML** | **No of Session** | **Weightage%** | |
|---|---|---|---|
| 1.1 Introduction to HTML5 , 1.2 tags and attributes, 1.5 different types of tags, Grouping using Div Span, Lists, Images, Hypelink and Anchors, Tables, iframe | 5 | 10% | **CO-1** |
| **MODULE 2: CSS** | | | |
| 2.1 Introduction to CSS3, 2.2 Architecture of CSS, 2.3 CSS Modules, 2.4 CSS Framework, Selectors and Pseudo Classes, 2.5 Fonts and Text Effects, Colors, Background Images, and Masks | 5 | 10% | **CO-2** |
| **MODULE 3: JavaScript** | | | |
| 3.1 Concept of script, Types of Scripts, 3.2 Introduction to JavaScript, Variables, identifiers constants in JavaScript and examples of each. 3.3 Operators in JavaScript's, various types of JavaScript operator, Examples on JavaScript operators, 3.4 Control and looping structure, examples on control and looping structures (if, if…else, for, while, do | 10 | 20% | **CO2, CO-3** |

| | | | |
|---|---|---|---|
| while, switch) 3.5 Concept of array, how to use it in JavaScript, types of an array, examples , Methods of an array, examples on it. 3.6 Event handling in JavaScript with examples, Math and date object and examples on it. 3.7 String object and examples on it, and some predefined functions, 3.8 DOM concept in JavaScript, DOM objects, 3.9 Validations in JavaScript, examples on it | | | |
| **MODULE 4:Web Servers** | | | |
| 4.1 Web server architecture, approaches , 4.2 Working with web browser, types of web servers, 4.3 Features of web servers, Benefits of web servers, 4.4 Uses of web servers, when to use web servers | 5 | 10% | **CO-5** |
| **MODULE 5:Ajax** | | | |
| 5.1 Introduction to AJAX: Exploring different web technologies, 5.2 Creating a simple AJAX application, 5.3 Interacting with the Web Server Using the XMLHttpRequest Object, Create an XMLHttpRequest Object, Interact with the Web Server. 5.4 Differentiating AJAX and Non-AJAX application.5.5 Working with PHP and AJAX: Introduction, 5.6 Process Client Requests, Accessing Files Using PHP, 5.7 Implementing Security and Accessibility in AJAX applications: 5.8 Introduction, Secure AJAX Applications, and Accessible Rich Internet Applications | 10 | 25% | **CO-4** |
| **Module 6: PHP** | | | |
| 6.1 Installation of Apache Tomcat (Xampp/Lampp/MySQL) 6.2 Installing and Configuring PHP, Introduction, PHP and the Web Server Architecture, PHP Capabilities, 6.3 PHP and HTTP Environment 6.4 Variables , Variables, Constants, Data Types, Operators, Working with Arrays , Decision Making, Flow Control and Loops , Introduction to Laravel , 6.5 Creating a Dynamic HTML Form with PHP, Database Connectivity with MySQL, Performing basic database operations (CRUD) , 6.6 Using GET, POST, REQUEST, SESSION, and COOKIE Variables | 10 | 25% | **CO-5** |

| **TEXT BOOKS** | |
|---|---|
| 1 | Complete reference HTML, TMH 2. HTML5 & CSS3, Castro Elizabeth 7th Edition |
| 2 | Beginning Node.js by Basarat Ali Syed 100 45 |
| 3 | Angular: Up and Running- Learning Angular, Step by Step by Shyam Seshadri |
| 4 | Beginning PHP, Apache, MySQL web development |
| **REFERENCE BOOKS** | |
| 1 | Introducing HTML5 - Bruce Lawson, Remy Sharp |
| 2 | Node.js in Action, 2ed by Alex Young, Bradley |
| 3 | Meck Mastering Node.js by Pasquali Sandro |
| 4 | Angular Essentials by Kumar Dhananjay |
| 5 | Complete Ref. PHP |
| **E BOOKS** | |
| 1. | https://nodejs.org/en/docs/guides/ |
| 2 | https://www.coursera.org/learn/web-development |
| **MOOC** | |
| 1. | https://www.coursera.org/learn/angular |
| 2 | https://www.coursera.org/specializations/web-applications |
| 3 | https://www.coursera.org/learn/database-applications-php |
| 4 | https://www.coursera.org/learn/server-side-nodejs |

| COURSE TITLE | | FUNDAMENTALS OF CLOUD COMPUTING & NETWORKING LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA106L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**MODULE 1: Introduction to Cloud Computing**

**Task 1.**Setting Up a Basic Web Server Using Docker (Introduction to Docker & Container)

**Task 2.** Comparative Table and Network Simulation Using Cisco Packet Tracer (Basic Networking + Cloud vs Cluster vs Grid)

**MODULE 2: Cloud Models**

**Task 1.** Exploring Cloud Service Models (IAAS, PAAS, SAAS) Using Real-Time

**Task 2**.Simulating XAAS and Cloud Storage Types Using Google Cloud Free Tier

**MODULE 3: Virtualization**

**Task 1.** Creating and Running a Virtual Machine Using Oracle VirtualBox (Full Virtualization)

**Task 2**. Exploring Open Source Virtualization Manager – Proxmox VE (Demo/Simulation)

**MODULE 4: Cloud Architecture & Management**

**Task 1**. Consuming a Public REST API Using Postman (RESTful Web Services)

**Task 2.** Create a Cloud Computing Architecture Diagram (Theoretical + Visual Activity)

**MODULE 5: Cloud Storage, Computing, and Networking & Emerging trends in cloud computing**

**Task 1**. Introduction to Omni-Cloud with Real-World Use Cases

**Task 2.** Exploring Blockchain Types through Case Studies

**MODULE 6 :Moving Applications to the Cloud**

**Task 1.** Case Study on Cloud Migration Strategies

**Task 2**. Demonstrate Data Migration to Cloud Using Google Drive / OneDrive

**Task 3**. Design a Cloud Bursting Scenario

| COURSE TITLE | DATA SCIENCE LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|
| COURSE CODE | MCA106L | COURSE CATEGORY | ME3.3 | L-P-T | 0-2-0 |
| Version 1.0 | Approval Details | | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | - |

**MODULE 1: Introduction To Data Science and Python**

**1.1** Explore and summarize real-world Data Science applications using case studies
**1.2** Write a Python script to demonstrate basic syntax, data types, and control flow
**1.3** Load and manipulate a dataset using NumPy arrays
**1.4** Perform basic data analysis using Pandas (e.g., loading, slicing, filtering)
**1.5** Compare simple machine learning models using scikit- learn on a toy dataset (e.g., Iris dataset)

**MODULE 2: Data Handling & Preprocessing**

**2.1** Collect and load a dataset from an open data source (e.g., Kaggle, UCI)
**2.2** Identify and handle missing values and outliers in a dataset
**2.3** Perform data wrangling: merging, reshaping, and renaming columns
**2.4** Apply normalization and standardization techniques on numerical data
**2.5** Encode categorical variables using one-hot encoding and label encoding

**MODULE 3: Exploratory Data Analysis (EDA)**

**3.1** Calculate and interpret central tendency and dispersion measures for a dataset
**3.2** Visualize data distributions using histograms and box plots
**3.3** Create scatter plots to identify relationships between variables
**3.4** Compute and visualize correlation matrices
**3.5** Perform basic statistical inference (e.g., confidence intervals, t-test)

**MODULE 4: Introduction to Machine Learning**

**4.1** Implement linear regression on a real-world dataset (e.g., house price prediction)
**4.2** Build a logistic regression model for binary classification (e.g., diabetes prediction)
**4.3** Apply K-means clustering on a dataset and visualize the clusters
**4.4** Demonstrate overfitting and underfitting using polynomial regression
**4.5** Evaluate model performance using confusion matrix and metrics (Precision, Recall, F1-score)

**MODULE 5: Data Visualization**

**5.1** Create line, bar, and pie charts using Matplotlib
**5.2** Generate comparative visualizations using Seaborn (e.g., bar plots, box plots)
**5.3** Plot scatter plots with regression lines using Seaborn's regplot()
**5.4** Build a dashboard-style layout with multiple plots for a dataset
**5.5** Apply best practices in visualization (titles, labels, legends, color use)

**MODULE 6: Tools & Techniques in Data Science**

**6.1** Build an end-to-end data pipeline (from data ingestion to model prediction)

**6.2** Process a large dataset using PySpark or Hadoop (basic level)

**6.3** Analyze a case study on bias and fairness in AI models

**6.4** Develop a mini-project: customer churn prediction, sentiment analysis, etc.

**6.5** Present a report on data science project lifecycle and its industrial relevance

| COURSE TITLE | | Fundamentals of Cloud Computing & Networking | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA106 | COURSE CATEGORY | ME1.1 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides an introduction to the principles and concepts of cloud computing, focusing on its role in modern IT infrastructure. It covers the architecture, deployment models, services, and security aspects of cloud computing, equipping students with the skills to design and deploy cloud-based solutions. |
| **Course Objective** | 1. Understand the core concepts and architecture of cloud computing.<br>2. Explore various cloud service models and deployment strategies.<br>3. Learn about cloud storage, computing, and networking services.<br>4. Address security, privacy, and compliance in cloud environments.<br>5. Gain practical experience with leading cloud platforms. |
| **Course Outcome** | CO1: **Understand** the foundational concepts and architecture of cloud computing.<br>CO2: **Analyze** different cloud service and deployment models.<br>CO3: **Utilize** cloud storage, computing, and networking services.<br>CO4: **Identify** and implement security measures in cloud environments.<br>CO5: **Develop** and deploy applications on cloud platforms. |

**Prerequisites: Basic knowledge of Networking**

| Module Name | No of lectures | Weightage | CO-PO mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Cloud Computing** | | | |
| 1.1. Introduction to Cloud Computing ,Basic Networking Concepts<br>1.2. Cloud Computing vs. Cluster Computing vs. Grid Computing<br>1.3. Characteristics, Pros and Cons of Cloud<br>1.4. Introduction to Dockers<br>1.5. Introduction to Container | 5 | 10% | **CO-1** |
| **MODULE 2: Cloud Models** | | | |
| 2.1 Cloud Service Models - IAAS, PAAS, SAAS & difference<br>2.2 Cloud Deployment Models-Public, Private, Hybrid, Community<br>2.3. XAAS- Anything as a Service – Storage as a service, Network as a Service, Database as a Service etc<br>2.4. Cloud Storage Types : Block, File, Object Storage<br>2.5 Cloud Platforms | 8 | 20% | **CO-2** |
| **MODULE 3: Virtualization** | | | |
| 3.1. Introduction to Virtualization concept & Hypervisors<br>3.2. Pros and Cons of Virtualization<br>3.3. Machine Image, Virtual Machine(VM).<br>3.4. Xen: Para virtualization, VMware: Full Virtualization<br>3.5. Microsoft Hyper-V<br>3.6. Open Source Virtualization Manager | 8 | 20% | **CO-3** |

| | | | |
|---|---|---|---|
| **MODULE 4: Cloud Architecture & Management** | | | |
| 4.1. Introduction to Service Oriented Architecture , Web Services: SOAP and REST<br>4.2. Relating SOA and Cloud Computing.<br>4.3. Service Level Agreement (SLA), Billing, Pricing, and Support.<br>4.4. Cloud Computing Architecture.<br>4.5. Multi Cloud Environment<br>4.6. Edge Computing Concepts<br>4.7. Cloud Bursting | 8 | 20% | **CO-4** |
| **MODULE 5: Cloud Storage, Computing, and Networking & Emerging trends in cloud computing** | | | |
| 5.1 Storage services, Computing services, Networking services,<br>5.2 Cloud service providers (AWS, Azure, Google Cloud),<br>5.3 Pricing and performance consideration<br>5.4 Omni Cloud.<br>5.5 Blockchain Technology.<br>5.6 Types of Blockchain technology.<br>5.7 Cloud AI | 8 | 15% | **CO-3** |
| **MODULE 6 :Moving Applications to the Cloud** | | | |
| 6.1. Cloud Migration Strategies and Process<br>6.2. Issues in Inter Cloud<br>6.3. Applications in the Clouds<br>6.4. Cloud Service Attributes<br>6.5. Cloud Bursting<br>6.6. Data Migration in Cloud | 8 | 15% | **CO-5** |
| **TEXT BOOKS** | | | |
| 1. | 1. "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl<br>2. "Architecting the Cloud" by Michael J. Kavis | | |
| **REFERENCE BOOKS** | | | |
| 1. | 1. "Mastering Cloud Computing" by Rajkumar Buyya<br>2. "Cloud Computing Bible" by Barrie Sosinsky | | |
| **E BOOKS** | | | |
| 1. | "Cloud Computing Explained" by John Rhoton (computingclouds.files.wordpress.com/2012/05/cloud-computing-explained.pdf) | | |
| 2 | "NIST Cloud Computing Standards Roadmap" (nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291.pdf) | | |
| **MOOC** | | | |
| 1. | "Cloud Computing Specialization" by the University of Illinois on Coursera (coursera.org/specializations/cloud-computing) | | |
| 2. | "Introduction to Cloud Computing" by IBM on Coursera (coursera.org/learn/introduction-to-cloud) | | |

| COURSE TITLE | DATA SCIENCE | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA106 | COURSE CATEGORY | ME1.3 | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course aims to provide MCA students with a fundamental understanding of Data Science, including data manipulation, statistical analysis, machine learning, and data visualization. The curriculum is designed to introduce students to the tools and techniques used in data science and their practical applications. |
| **Course Objective** | To introduce the basic concepts and techniques of data science.<br>To develop skills in data manipulation and analysis.<br>To provide an understanding of statistical methods for data analysis.<br>To introduce machine learning algorithms and their applications.<br>To develop skills in data visualization and communication of results. |
| **Course Outcome** | By the end of the course, students will be able to:<br><br>CO1: Understand the fundamental concepts of data science, its applications, and lifecycle.<br>CO2: Learn to handle, clean, and preprocess data for analysis.<br>CO3: Apply exploratory data analysis and statistical techniques to derive insights.<br>CO4: Understand the basics of machine learning algorithms and implement simple models.<br>CO5: Analyze with data visualization techniques to communicate results.<br>CO6: Evaluate real-world data science case studies and apply learned techniques. |

| MODULE | No. of lectures | Weightage (%) | CO Mapped |
|---|---|---|---|
| **MODULE 1: Introduction To Data Science and Python** | | | |
| 1.1 Definition of Data Science, 1.2 Applications of Data Science 1.3 Data Science Lifecycle, 1.4 Data-Driven Decision Making, 1.5 Libraries: Data Manipulation Libraries (Pandas, NumPy), 1.6 Machine Learning Frameworks (scikit-learn, TensorFlow, Pytorch) | 5 | 15% | **CO1, CO2** |
| **MODULE 2: Data Handling & Preprocessing** | | | |
| 2.1 Data Collection Methods, 2.2 Types of Data: Structured, Unstructured, Semi-structured, 2.3 Data Cleaning: Handling missing values, outliers, Data Wrangling 2.4 Data Transformation: Normalization, Standardization, 2.5 Feature Engineering: Encoding categorical variables, Feature scaling | 6 | 15% | **CO2, CO3** |
| **MODULE 3: Exploratory Data Analysis (EDA)** | | | |

| | | | |
|---|---|---|---|
| 3.1 Understanding Data Distributions: Mean, Median, Mode, Variance, Standard deviation, 3.2 Visualizing Distributions: Histograms, Box plots, Scatter plots, 3.3 Identifying Patter ns and Relationships in Data, 3.4 Correlation and Covariance, 3.5 Introduction to Statistical Inference | 10 | 20% | **CO2, CO3** |
| **MODULE 4: Introduction to Machine Learning** | | | |
| 4.1 Introduction to Machine Learning and its Types (Supervised, Unsupervised, Reinforcement Learning), 4.2 Supervised Learning: Regression (Linear, Logistic), Classification, 4.3 Unsupervised Learning: Clustering (K-means, Hierarchical), 4.4 Overfitting and Underfitting, 4.5 Model Evaluation Metrics (Accuracy, Precision, Recall, F1-score) | 8 | 20% | **CO4** |
| **MODULE 5: Data Visualization** | | | |
| 5.1 Importance of Data Visualization (Seaborn, Matplotlib), 5.2 Types of Plots: Line plot, Bar plot, Pie chart, Scatter plot, 5.3 Visualization Tools: Matplotlib, Seaborn, 5.4 Best Practices in Visualization | 6 | 15% | **CO5** |
| **MODULE 6: Tools & Techniques in Data Science** | | | |
| 6.1 Implementing Data Science Pipelines, 6.2 Introduction to Big Data & Tools (Hadoop, Spark), 6.3 Ethical Issues in Data Science, 6.4 Real world Data Science Projects and Applications in Industry | 10 | 15% | **CO6** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1 | Python for Data Analysis by Wes McKinney |
| 2 | Data Science from Scratch by Joel Grus |
| 3 | Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido |
| **REFERENCE BOOKS** | |
| 1 | Jure Leskovek, Anand Rajaraman and Jeffrey Ullman. Mining of Massive Datasets. v2.1, Cambridge University Press. |
| **E BOOKS** | |
| 1 | "Data Science from Scratch: First Principles with Python" by Joel Grus |
| 2 | "Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking" by Foster Provost and Tom Fawcett |
| **MOOC** | |
| 1 | Coursera: "Introduction to Data Science" by University of Washington |
| 2 | edX: "Data Science and Machine Learning Essentials" by Microsoft<br>Coursera: "Data Science Specialization" by Johns Hopkins University<br>edX: "Introduction to Data Science" |

| COURSE TITLE | | CAPSTONE PROJECT | | | CREDITS | 2 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA107 | COURSE CATEGORY | PROJ | L-P-T | 0-4-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |
| ASSESSMENT SCHEME | | | | | | |
| Review 1 | Review 2 | Review 3 | Final Review | | Attendance | ESE |
| 10% | 10% | 20% | 40% | | 20% | -- |

# Capstone Project Schedule

| Sr.No | Reviews | Topic | Documents Required |
|---|---|---|---|
| 1 | | Project Title Submission | **Project Title** |
| 2 | Review 1 | Project Synopsis | **INTRODUCTION**<br>• Existing System and Need for System<br>• Scope of Work<br>• Operating Environment - Hardware and Software Detail Description of Technology Used<br>**PROPOSED SYSTEM**<br>• Proposed System<br>• Objectives of System<br>• User Requirements |
| 3. | Review 2 | Design | **ANALYSIS & DESIGN**<br>• Module Hierarchy Diagram<br>• Use Case Diagrams<br>• Class Diagram<br>• E-R Diagram<br>• Activity Diagram<br>• Sequence Diagram<br>• Web Site Map Diagram (in case of Web Site) |
| 5 | Review 3 | Database Connectivity | • User Interface Design (Screens etc.)<br>• Database Table Structure<br>• Database Connectivity<br>• Form Validation, Reports |
| 6 | Review 4 | Final Submission | Final Project Execution, checking Complete Documentation of Project with all Diagrams and PPT, Viva and submission of Spiral Binding documentation. |

# Capstone Project Guidelines

1. Students can work individually or in pairs (maximum of 2) for the Capstone Project. Projects must relate to both laboratory subjects being taught in the current semester. The student may take up the project individually or in group. If project is done in group, each student must be given a responsibility for distinct modules.
2. Few hours per week is dedicated to project work. During this time, students must report to their assigned guides for support and discuss project progress. Attendance is mandatory for all review sessions and weekly project hours; project diaries must be presented at each review. (check class time table for more details)
3. Students must submit a clear and self-descriptive project title by the end of the first week of the semester. Following guide allotment, a project synopsis must be submitted by the second week, including sections on existing systems, need for the project, scope of work, objectives, and user requirements.
4. There will be a total of 4 reviews throughout the semester. Specific deliverables are required at each stage, including design documents, implementation details, and testing strategies. Students are expected to create a PowerPoint presentation for each review, summarizing their progress.
5. Documentation must adhere to specified standards: use Times New Roman, size 12 pt, with 1.5line spacing and justified alignment. Margins should be set to 1" (left and right), 2" (top), and 1.5" (bottom). All documentation must be spiral-bound and submitted by the final deadline.
6. The final evaluation will consist of a project presentation (PowerPoint) and a viva voce conducted by faculty members. Students must submit a comprehensive project report, including all relevant diagrams and design documents, on the final submission date.
7. All project work must be original and must adhere to the ICEM's academic integrity policies. Plagiarism will not be tolerated and may result in academic penalties.
8. Students are encouraged to utilize resources such as textbooks, online articles, and tutorials to support their project development. A list of recommended resources should be submitted along with the final project report.
9. Feedback from guides must be incorporated into subsequent project phases. Students should schedule regular meetings (in Capstone Project slot) with their respective guides to ensure alignment and address any challenges faced during project development.
10. Any changes to the project scope or objectives must be discussed with and approved by the guide before implementation. Documentation of such changes should be maintained in the project diary.
11. All project-related communications, including submission links and updates, will be shared via email and designated online platforms. Students should regularly check these platforms for important announcements.

| **MOOC** | |
|---|---|
| 1. | **Coursera - "Project Management Principles and Practices" by Google**<br>https://www.coursera.org/specializations/project-management |
| 2. | **edX - "Capstone Project: Data Science for Everyone" by IBM**<br>https://www.edx.org/professional-certificate/ibm-data-science |

| COURSE TITLE | SOFT SKILLS & BUSINESS COMMUNICATION | | | CREDITS | 1 |
|---|---|---|---|---|---|
| COURSE CODE | MCA108 | COURSE CATEGORY | CM | L-P-T | 1-0-0 |
| Version 1.0 | Approval Details | 07-2024 | | | |

**ASSESSMENT SCHEME**

| Seminar | Surprise Test / Quiz | Internal Exam | Attendance | ESE |
|---|---|---|---|---|
| 20% | 20% | 20% | 40% | -- |

| | |
|---|---|
| **Course Description** | This course focuses on developing essential soft skills and effective business communication techniques required for professional growth. It emphasizes verbal and non-verbal communication, presentation skills, team collaboration, and leadership, preparing students for the corporate environment. |
| **Course Objective** | • To enhance students' communication skills in professional settings.<br>• To develop interpersonal and team-building skills.<br>• To prepare students for effective public speaking and presentations.<br>• To improve time management and problem-solving abilities.<br>• To build confidence and leadership qualities in a business context. |
| **Course Outcome** | After end of this course student will be able:<br>• **CO1**: Demonstrate effective verbal and written communication in a business setting.<br>• **CO2**: Exhibit interpersonal skills and teamwork in professional interactions.<br>• **CO3**: Apply presentation and public speaking skills in business scenarios.<br>• **CO4**: Develop leadership and problem-solving skills.<br>• **CO5**: Manage time and stress efficiently in a workplace environment. |

**Prerequisites:**

- Basic knowledge of the English language.
- Familiarity with email and digital communication tools.

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Soft Skills** | | | |
| **Topic:** 1.1 Definition and importance of soft skills, 1.2 Overview of communication skills, 1.3 Importance of emotional intelligence, 1.4 Understanding self-awareness and self-regulation, 1.5 Soft skills in the workplace. | **2** | **10%** | **CO-1** |
| **MODULE 2: Verbal and Non-Verbal Communication** | | | |
| **Topic:** 2.1 Verbal communication techniques, 2.2 Importance of tone, pitch, and clarity, 2.3 Non-verbal cues: body language, facial expressions, and gestures, 2.4 Barriers to effective communication, 2.5 Active listening skills and their significance, 2.6 Techniques for overcoming communication barriers. | **3** | **20%** | **CO-2** |
| **MODULE 3: Business Communication Tools** | | | |

| | | | |
|---|---|---|---|
| **Topic:** 3.1 Writing professional emails: structure and etiquette, 3.2 Creating effective business reports and proposals, 3.3 Conducting meetings: agenda setting and minutes writing, 3.4 Presentation tools and techniques (e.g., PowerPoint, Prezi), 3.5 Utilizing digital communication platforms (e.g., Slack, Zoom), 3.6 Visual communication: charts, graphs, and infographics. | **2** | **15%** | **CO-3** |
| **MODULE 4: Teamwork and Collaboration** | | | |
| **Topic:** 4.1 Dynamics of team collaboration and roles, 4.2 Building trust and rapport within a team, 4.3 Conflict resolution strategies: negotiation and mediation, 4.4 Leadership styles and their impact on teamwork, 4.5 Strategies for fostering inclusive team environments, 4.6 Tools for effective team collaboration (e.g., Trello, Asana). | **3** | **20%** | **CO-3 and CO-4** |
| **MODULE 5: Time and Stress Management** | | | |
| **Topic:** 5.1 Understanding the importance of time management, 5.2 Prioritization techniques: Eisenhower Matrix and ABCD method, 5.3 Goal setting and action planning, 5.4 Managing workplace stress: techniques and tools, 5.5 Work-life balance strategies, 5.6 The role of mindfulness in stress management. | **2** | **15%** | **CO-4 and CO-5** |
| **MODULE 6: Public Speaking and Presentation Skills** | | | |
| **Topic:** 6.1 Structuring an effective presentation: introduction, body, conclusion, 6.2 Techniques for engaging your audience, 6.3 Overcoming stage fear: tips and tricks, 6.4 Using visual aids and technology effectively, 6.5 Rehearsal techniques for effective delivery, 6.6 Handling Q&A sessions and audience feedback. | **3** | **20%** | **CO-5** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | **Soft Skills: Enhancing Employability by M.S. Rao, I.K. International Publishing House.** |
| 2 | **Business Communication by Meenakshi Raman & Sangeeta Sharma, Oxford University Press.** |
| **REFERENCE BOOKS** | |
| **1.** | **Developing Soft Skills by Dr. K. Alex, S. Chand Publishing.** |
| 2. | **Effective Business Communication by Asha Kaul, Prentice Hall India.** |
| **E BOOKS** | |
| 1. | **Soft Skills Development by Kul Bhushan** https://www.pdfdrive.com/soft-skills-development-e53627497.html |
| 2 | **Effective Communication Skills by MTD Training** https://www.pdfdrive.com/effective-communication-skills-e15874210.html |
| | |
| 1. | **Effective Communication in the Workplace** https://www.edx.org/course/effective-communication-in-the-globalized-workplace |
| 2. | **NPTEL Data Structures and Algorithms** https://archive.nptel.ac.in/courses/106/102/106102064/ |

| COURSE TITLE | PYTHON PROGRAMMING | | CREDITS | 3 |
|---|---|---|---|---|
| COURSE CODE | MCA201 | COURSE CATEGORY | CM | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 07-2024 | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces students to the fundamental concepts and practical applications of Python programming. It covers basic programming constructs, data structures, object-oriented programming, and advanced topics like libraries and real-world applications. The course is designed to enhance problem-solving skills and prepare students for data analysis and software development tasks using Python. |
| **Course Objective** | • To introduce basic programming concepts using Python.<br>• To develop problem-solving skills through Python.<br>• To implement algorithms and data structures in Python.<br>• To understand file handling, modules, and libraries.<br>• To develop knowledge of object-oriented programming in Python. |
| **Course Outcome** | By the end of the course, students will be able to:<br><br>• CO1: Understand Python syntax and core programming constructs.<br>• CO2: Apply Python for problem-solving and algorithm implementation.<br>• CO3: Utilize Python libraries and modules for data manipulation and analysis.<br>• CO4: Develop object-oriented programs in Python.<br>• CO5: Design and implement real-world applications using Python. |

| MODULE | No. of lectures | Weightage (%) | CO Mapped |
|---|---|---|---|
| **MODULE 1: Fundamentals of Python** | | | |
| 1.1 Overview of Python's history and features, 1.2 Installing Python and IDEs, 1.3 Syntax-variables- data types: 1.4 Handling user inputs and displaying outputs, 1.5 Operators: Arithmetic-logical and comparison operators, 1.6 Conditional statements: if-else-nested if, 1.7 Loops: for-while-nested loops | 5 | 10% | **CO1** |
| **MODULE 2 : Data Structures and Libraries** | | | |
| 2.1 Lists-tuples-sets-dictionaries: Usage and operations on built-in data structures, 2.2 String operations: String methods and formatting, 2.3 Numpy basics: Introduction to Numpy arrays and operations, 2.4 File handling: Reading from and writing to files | 6 | 12% | **CO2** |
| **MODULE 3: Functions- Modules & Packages- Exceptional Handling** | | | |

| | | | |
|---|---|---|---|
| 3.1 Functions: Defining and calling functions- recursion-lambda functions, 3.2 Scope and global variables: Understanding local vs global scope and variable lifetime, 3.3 Python built-in modules – math-random-datetime, 3.4 Package: import basics, 3.5 Python namespace packages, 3.6 User defined modules and packages, 3.7 Exception Handling Programming using Exception handling | 10 | 18% | **CO2,CO3** |
| **MODULE 4: Object-Oriented Programming** | | | |
| 4.1 Concept of class-object, 4.2 Constructor-destructors, 4.3 Inheritance- super class-method overriding, 4.4 Overloading operators, 4.5 Static and Class methods, 4.6 Python Regular Expression, 4.7 Pattern matching and searching using regex in python, 4.8 Multithreading, 4.9 Understanding threads, 4.10 Synchronizing the threads, 4.11 Programming using multithreading, 4.12 Understanding threads, 4.13 Synchronizing the threads, 4.14 Programming using multithreading | 8 | 20% | **CO4, CO5** |
| **MODULE 5: Python database interaction using MongoDB** | | | |
| 5.1 Introduction to NoSQL database, 5.2 Types of NoSQL, 5.3 Document Based: MongoDB, 5.4 Key-Value Database – Couchbase, 5.5 Wide-column Databases: Cassandra, 5.6 Graph/node Databases: Neo4j, 5.7 SQL Vs NoSQL, 5.8 Introduction to MongoDB with python, 5.9 Installing MongoDB on Windows, 5.10 Exploring Collections and Documents, 5.11 Performing CRUD Operations, 5.12 Commit-Rollback and Cursor operation | 6 | 15% | **CO4, CO5** |
| **MODULE 6: Web Development using Django** | | | |
| 6.1 Introduction to Web Development and Django, 6.2 Django Project Structure and Django Models, 6.3 Django Views and Django Templates, 6.4 Django URLs and Django Forms, 6.5 Django Authentication and Advanced Django Features, 6.6 Django Rest Framework (DRF) and Testing in Django,6.7 Deployment and Performance Optimization | 10 | 25% | **CO5** |
| **TEXT BOOKS** | | | |
| 1 | "Python Programming: An Introduction to Computer Science" by John Zelle. | | |
| 2 | "Learning Python" by Mark Lutz. | | |
| 3 | "Python for Data Analysis" by Wes McKinney. | | |
| **REFERENCE BOOKS** | | | |
| 1 | "Automate the Boring Stuff with Python" by Al Sweigart. | | |
| 2 | "Fluent Python" by Luciano Ramalho. | | |
| **E BOOKS** | | | |
| 1 | "Data Science from Scratch: First Principles with Python" by Joel Grus | | |
| 2 | "Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking" by Foster Provost and Tom Fawcett | | |
| **MOOC** | | | |

| | |
|---|---|
| 1<br>2 | https://www.udemy.com/course/complete-python-bootcamp/<br>https://www.edx.org/certificates/professional-certificate/the-georgia-institute-of-technology-introduction-to-python-programming |

| COURSE TITLE | | PYTHON PROGRAMMING LAB | | | CREDITS | 2 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA201L | COURSE CATEGORY | SEC | L-P-T | 0-4-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | - |

| | |
|---|---|
| **Course Description** | This course introduces the fundamentals of Python programming. It covers basic syntax, control structures, data types, functions, modules, file handling, object-oriented programming, and various Python libraries. The course aims to provide students with a solid foundation in Python, enabling them to develop efficient and scalable applications. |
| **Course Objective** | 1. Comprehend the basic syntax and semantics of Python programming.<br>2. Implement control structures and data types in Python.<br>3. Develop functions and modules for modular programming.<br>4. Utilize object-oriented programming concepts in Python.<br>5. Handle files and exceptions in Python applications.<br>6. Use Python libraries for data analysis and visualization. |
| **Course Outcome** | After end of this course student will be able:<br>CO1: To understand the basic concepts of Python programming.<br>CO2: To apply Python programming constructs to solve computational problems.<br>CO3: To develop applications using object-oriented programming techniques in Python.<br>CO4: To utilize Python libraries for data processing and visualization.<br>CO5: To implement file handling and exception handling in Python programs. |

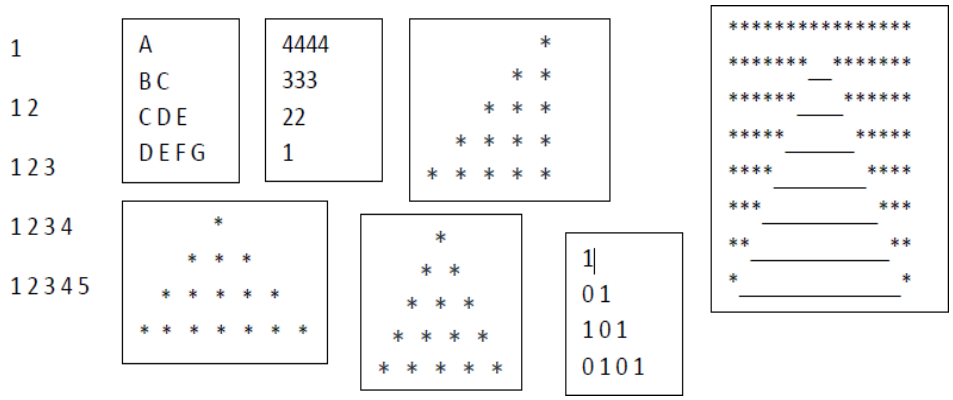
**Prerequisites:**

- Fundamental knowledge of basic data structures (e.g., arrays, linked lists, stacks, queues).
- Understanding of basic algorithms and their complexity (e.g., searching, sorting).
- Proficiency in a programming language (preferably C, C++, or Java).

## MODULE 1: FUNDAMENTALS OF PYTHON

1. Write a Python program that calculates the area of a circle based on the radius entered by the user.
    a. Sample Output : r = 1.1 Area = 3.8013271108436504
2. Write a Python program that accepts the user's first and last name and prints them in reverse order with a space between them.
3. Write a Python program that accepts a sequence of comma-separated numbers from the user and generates a list and a tuple of those numbers.
    a. Sample data : 3, 5, 7, 23 Output : List : ['3', ' 5', ' 7', ' 23'] Tuple : ('3', ' 5', ' 7', ' 23') 4.
4. Write a Python program that determines whether a given number (accepted from the user) is even or odd, and prints an appropriate message to the user.
5. Write a Python program to concatenate N strings.
6. Write a Python program to do arithmetical operations addition and division.
7. Write a Python program to find the area of a triangle.
8. Write a Python program to swap two variables.
9. Write a Python program to generate a random number.
10. Write a Python program to convert kilometers to miles.

11. Write a Python program to display calendar.
12. Write a Python program to swap two variables without temp variable.
13. Write a Python Program to Check if a Number is Positive, Negative or Zero.
14. Write a Python Program to Check Leap Year.
15. Write a Python Program to Check if a Number is Odd or Even.
16. Write a program that asks the user to enter a number and then prints all the even numbers from 0 to that number.
17. Write a program that asks the user to enter a string and then prints each character in the string on a new line.
18. Write a program to print following pattern




19. Write a Python program to find ASCII value of a character.
20. Write a Python program to make a simple calculator with 4 basic mathematical operations.
21. Write a Python program to find a largest element in an array.
22. Write a Python program to add two matrices.

## MODULE 2: DATA STRUCTURES AND LIBRARIES

1. Create a dictionary of your favourite books and their authors and print it.
2. Add a new book to the dictionary and print the updated dictionary.
3. Remove a book from the dictionary and print the updated dictionary.
4. Use the keys() method to print a list of the book titles in the dictionary.
5. Use the values() method to print a list of the author names in the dictionary.
6. Create a set of your favourite colors and print it.
7. Add a new color to the set and print the updated set.
8. Remove a color from the set and print the updated set.
9. Create a new set that contains only the colors that start with the letter "B" and print it.
10. Use the len() function to find the number of colors in the set and print it.
11. Program to Find the GCD of Two Positive Numbers.
12. Write Python Program to Find the Sum of Digits in a Number.
13. Write a program that prints the first 10 multiples of 3.
14. Create a list of your favorite movies and print the third movie in the list.
15. Add a new movie to the list and print the updated list.
16. Remove the second movie from the list and print the updated list.
17. Sort the list in alphabetical order and print the sorted list.
18. Create a new list that contains only the first and last movie in the original list and print it.
19. Create a tuple of your favorite foods and print the second food in the tuple.
20. Try to change the second food in the tuple and see what happens.
21. Create a new tuple that contains only the first and last foods in the original tuple and print it. Use the **len()** function to find the number of foods in the tuple and print it.

## MODULE 3: FUNCTIONS, MODULES & PACKAGES, EXCEPTION HANDLING

1. Write a Program to Find the Sum of All Odd and Even Numbers up to a Number Specified by theUser using functions
2. Write Python Code to Determine Whether the Given String Is a Palindrome or Not Using Slicing – Use functions.
3. Write Python function to Count the Total Number of Vowels, Consonants and Blanks in a String.
4. Write a function to Print the Characters Which Are Common in Two Strings.
5. Write a Program to Read marks from user and Find the percentage of marks of student.
6. Write a function to Display the Fibonacci Sequences up to nth Term Where n is Provided by the user.
7. Write a Python function That Accepts a Sentence as Input and Removes All Duplicate Words. Print the Sorted Words.
8. Write Python function to Implement Stack Operations using *args.
9. Create a package named library and implement few functions of library in python.
10. Create a module in python to perform simple calculator operations.

**MODULE 4: OBJECT-ORIENTED PROGRAMMING**

1. Write a program to create point class with x,y,z coordinate and methods increment point, decrementpoint, add points , less than , greater than , equal to , check in which quadrant it lies,check whether the point is collinear and print point.
2. Create class watch with hr,min,sec,alarm,type and methods setalarm, stopalarm,showtime.
3. Write Python Program to Simulate a Bank Account with Support for depositMoney, withdrawMoney and showBalance Operations.
4. Create class vehicle with attributes(color,capacity,enginpower,tyre ) and behaviour (start,stop)
5. Create class car which inherit vehicle class with attributes(airbags,gear,speed,fuel,) and methods(accelerate ,fillfuel,playmusic(),onAC())
6. Create class electric car with attribute(battery) and behaviour(charging(),battery level().

**MODULE 5: PYTHON DATABASE INTERACTION USING MongoDB**

1. Insert Single Document – A program to insert one document into a MongoDB collection.
2. Insert Multiple Documents – Inserts a list of documents into a collection at once.
3. Find All Documents – Retrieves and prints all documents from a specified collection.
4. Find Document by Field – Searches for documents that match a specific field value (e.g., name = "Alice").
5. Update a Single Document – Updates the first document that matches a given condition.
6. Update Multiple Documents – Updates all documents that meet a certain filter.
7. Delete a Single Document – Removes one document based on a condition.
8. Delete Multiple Documents – Deletes all documents that satisfy a condition.
9. Find Documents with Projection – Retrieves documents but only returns specific fields.
10. Sort Documents – Finds and sorts documents based on one or more fields.
11. Limit Query Results – Retrieves a limited number of documents from a query.
12. Skip Documents in Query – Skips a specified number of documents and returns the rest.
13. Create an Index – Creates an index on one or more fields to speed up queries.
14. Drop an Index – Removes an index from a collection.
15. Aggregate with $group – Groups documents by a field and performs aggregations like sum or average.
16. Aggregate with $match and $project – Filters and reshapes documents in an aggregation pipeline.
17. Check if Collection Exists – Checks whether a collection exists in the database.
18. Count Documents – Returns the count of documents that match a query.
19. Paginate Results – Implements pagination to retrieve documents in chunks or pages.
20. Connect to MongoDB Atlas – Connects to a remote MongoDB cluster hosted on MongoDB Atlas.

**MODULE 6: WEB DEVELOPMENT USING Django**

1. Create a simple Django project and run the development server.
2. Build a basic Django model and perform migrations.
3. Implement a simple Django form and handle user input.
4. Create a basic user authentication system.
5. Develop a REST API using Django Rest Framework (DRF).

Note: Subject teacher can give additional assignments for practice apart from the mentioned assignment list.

| COURSE TITLE | Advanced DBMS | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA204 | COURSE CATEGORY | C M | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | 01/07/2024 | | LEARNING LEVEL | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments /Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course delivers advanced concepts of database management systems (DBMS) focusing on modern database technologies, query optimization, transaction management |
| **Course Objective** | The primary objective is to equip students with advanced knowledge and skills in designing, implementing, and managing complex database systems to handle large volumes of data efficiently. |
| **Course Outcome** | By the end of this course, students will be able to:<br>1. Design and implement advanced database schemas.<br>2. Design relational database using E-R model and normalization<br>3. Optimize database queries and transactions for performance.<br>4. Explain concepts of Parallel, Distributed and their applications<br>5.Evaluate and select appropriate database technologies for specific applications. |

**Prerequisites:** Basic knowledge of database concepts

| Module Name | Sessions | Weightage(%) | CO-PO Mapping |
|---|---|---|---|
| **MODULE 1: Database Design & SQL Query Processing** | | | |
| 1.1 Database and Need for DBMS, Characteristics of DBMS<br>1.2 Database 3-tier schema (ANSI/SPARC) and system architecture of DBMS<br>1.3 Views of data- Schemas and instances<br>1.4 Data Modeling using Entity-Relationship (ER) Diagram : Representation of Entities, Attributes, Relationships and their Types, Cardinality, Generalization, Specialization, Aggregation.<br>1.5 Normalization (1NF,2NF,3NF,BCNF)<br>1.6 Introduction to SQL Query Processing (DDL, DML, Aggregate Functions and Joins ) | 4 | 10% | **CO-1 & CO-2** |
| **MODULE 2: Transaction Management & Concurrency Control** | | | |
| 2.1 Concept of Transaction and Transaction processing ACID properties of transactions, States of transaction<br>2.2 Concurrency Control and Problems in Concurrency Control<br>2.3 Concurrency Control Protocols: Lock-Based Protocol and Time Stamp-based ordering protocols<br>2.4 Scheduling of Transactions | 7 | 15% | **CO-3** |

| | | | |
|---|---|---|---|
| 2.5 Recovery techniques: Undo logging, Redo logging<br>2.6 Deadlock , Deadlock handling Methods | | | |
| **MODULE 3: Parallel Databases** | | | |
| 4.1. Introduction to Parallel Databases<br>4.2. Parallel Database Architectures<br>4.3. I/O parallelism<br>4.4. Inter-query and Intra-query parallelism<br>4.5. Inter-operational and Intra-operational parallelism<br>4.6. Key elements of parallel database processing: Speed-up, Scale-up Synchronization and Locking | 8 | 20% | **CO-4** |
| **MODULE 4: Distributed Databases** | | | |
| 4.1. Introduction to Distributed Database System<br>4.2. Homogeneous and Heterogeneous Databases<br>4.3. Distributed data storage (Fragmentation and Replication)<br>4.4. Distributed transactions<br>4.5. Concurrency control schemes in DDBMS<br>4.6. Commit protocols 2 phase and 3 Phase | 8 | 15% | **CO-4** |
| **MODULE 5: Database Recovery and Security Techniques** | | | |
| 5.1 Database Recovery and Security Techniques<br>5.2 Failure Classification<br>5.3 Recovery and Atomicity<br>5.4 Log-Based Recovery<br>5.5 Check Points, Shadow Paging<br>5.6 Introduction to Database backup, Types of backups<br>5.7 Database Security in DBMS, Importance of Database Security, Security Threats, Challenges in Database Security<br>5.8 Discretionary access control based on grant & revoking Privilege<br>5.9 Encryption- its types & Public & Private key Infrastructures | 8 | 20% | **CO-5** |
| **Module 6 : NOSQL Database** | | | |
| 6.1 Introduction to NOSQL Database: Overview, History of NoSQL Databases, The Definition of the Four Types of NoSQL Databases.<br>6.2 Introduction to MongoDB Database, JSON and JSON Structure, NoSQL Key/Value databases, Graph NoSQL Databases using Neo4J<br>6.3 NoSQL database development tools and programming languages, Future Trends in NoSQL Databases<br>6.4 Introduction to FireBase, Firebase Console, Understanding firebase project structure and settings<br>6.5 Security Rules, Understanding and implementing App security measures in Firebase. | 10 | 20% | **CO-5** |

| **TEXT BOOKS** | |
|---|---|
| 1. | "Database System Concepts" by Abraham Silberschatz et al. |
| 2. | "Database Management Systems" by Raghu Ramakrishnan et al. |

| **REFERENCE BOOKS** | |
|---|---|
| 1. | "Database Systems: The Complete Book" by Hector Garcia-Molina et al. |
| 2. | Database System Concepts" by Abraham Silberschatz, Henry F. Korth, and S. Sudarshan |

| E BOOKS | |
|---|---|
| 1. | Database System Concepts by Abraham Silberschatz et al. |
| 2. | Database Management Systems by Raghu Ramakrishnan et al. |
| **MOOC** | |
| 1. | Coursera - Advanced Database Management |

| COURSE TITLE | SOFTWARE TESTING AND TOOLS | | CREDITS | 3 |
|---|---|---|---|---|
| COURSE CODE | MCA203 | COURSE CATEGORY CM | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | 07-2024 | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test/ Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces students to the fundamental concepts, techniques, and tools used in software testing. It covers various types of testing, including unit testing, integration testing, system testing, and acceptance testing. Students also learn about automated testing tools and techniques for ensuring software quality and reliability. |
| **Course Objective** | 1. To understand fundamental concepts, definitions, and objectives of software testing.<br>2. To Identify and apply various types of testing methods, including static and dynamic techniques.<br>3. To Design and execute effective test cases using appropriate test design techniques.<br>4. To Manage the software testing process through comprehensive test planning and monitoring.<br>5. To Utilize various software testing tools effectively to enhance testing efficiency and effectiveness. |
| **Course Outcome** | After the end of course student will be able to<br>**CO1** Understand the fundamental concepts and objectives of software testing.<br>**CO2** Analyze different types of testing and their role in the software development lifecycle.<br>**CO3** Apply static and dynamic testing techniques to improve software quality.<br>**CO4** Manage testing processes, including planning, monitoring, and reporting.<br>**CO5** Evaluate and utilize various testing tools to enhance testing efficiency. |

| MODULE | No of lectures | Weightage | CO Mapped |
|---|---|---|---|
| **MODULE 1: Introduction to Software Testing** | | | |
| 1.1. Definition & Objectives of Testing ,1.2. Role of testing and its effect on quality, 1.3. Causes of software failure: Definition of –Error Bug Fault Defect and Failure, 1.4. Economics of Testing, 1.5. Seven Testing Principles ,1.6. | 4 | 8% | **CO-1, CO-2** |

| | | | |
|---|---|---|---|
| Software Testing Life cycle, 1.7. Validation & Verification Concepts - V Model and W Model | | | |
| **MODULE 2:Software Testing Fundamentals** | | | |
| 2.1. Agile Testing- Test Driven Software Development, 2.2. Levels of Testing,2.2.1. Unit (Component) Testing, 2.2.2. Integration Testing ,2.2.3. System Testing, 2.2.4. User Acceptance Testing (UAT), 2.3. Test Types, 2.3.1. Functional testing (Black-box), 2.3.2. Non-functional testing (Testing of software product characteristics), 2.3.3. Structural testing (White-box) ,2.3.4. Testing related to changes - Confirmation (Re-testing) and Regression Testing ,2.4. Non-Functional Testing Types – 2.4.1. Performance (Load & Stress) ,2.4.2. Usability, 2.4.3. Maintainability ,2.4.4. Portability, 2.4.5. Security, 2.4.6. Localization & Internationalization ,2.5. Concept of Smoke testing and Sanity Testing | 5 | 12% | **CO-1, CO-2** |
| **MODULE 3: Static Testing** | | | |
| 3.1. Static Techniques – Review 3.1.1. Review Process (Informal & Formal) ,3.1.2. Desk Checking, 3.1.3. Technical or Peer Review, 3.1.4. Walkthrough, 3 3.1.5. Inspection, 3.2. Static Techniques – Static Analysis, 3.2.1. Data flow analysis, 3.2.2. Control flow analysis, 3.2.3. Static Analysis by Tools (Automated Static Analysis) | 10 | 22% | **CO-3** |
| **MODULE 4: Dynamic Testing** | | | |
| 4.1. Test Design Techniques-Black Box Testing Techniques: 4.1.1. Equivalence Partitioning ,4.1.2. Boundary Value Analysis, 4.1.3. Decision Table Testing ,4.2. Test Design Techniques -White Box Testing ,Techniques (coverage based and fault-based) 4.2.1. Statement coverage, 4.2.2. Branch & Decision coverage ,4.2.3. Path coverage, 4.2.4. McCabe's Cyclomatic Complexity Metric (Computation of Cyclomatic Complexity to be covered) ,4.2.5. Data Flow based Testing, 4.2.6. Mutation Testing, 4.3. Test Design Techniques -Experience based techniques: 4.3.1. Error Guessing, 4.3.2. Exploratory Testing. | 10 | 22% | **CO-3** |
| **MODULE 5: Test Management** | | | |
| 5.1. Test Organization- Roles & Skills of Tester Test Lead Test Manager ,5.2. Test Planning- Test Plan as per IEEE 829 STANDARD TEST PLAN TEMPLATE , 5.3.2. Reporting Test Status (IEEE 829: TEST SUMMARY REPORT TEMPLATE to be discussed) ,5.3.3. Test Control, 5.4. Requirement Traceability Matrix (Horizontal & Vertical), Test Scenario Test Suite Test Cases (both Positive & Negative Test Cases, as per IEEE 829: TEST CASE SPECIFICATION TEMPLATE) ,5.5. Configuration Management- Configuration Management support for Testing, 5.6. Risk and Testing- Project Risk & Product Risk, 5.7. Incident/ Defect Management ,5.7.1. Defect Life Cycle ,5.7.2. Defect/ Incident Report (IEEE 829: TEST INCIDENT REPORT TEMPLATE to be discussed) | 5 | 12% | **CO-4** |

| MODULE 6: Testing Tools | | | |
|---|---|---|---|
| 6.1. Types of Test tools –CAST (only type & their purpose & Benefits and Risks should be covered) ,6.2. Introduction of a tool into an organization, 6.3. Testing tools ,6.3.1. Selenium -WebDriver and Test NG ,6.3.2. JMeter, 6.3.3. Postman ,6.3.4. ETL Testing Tool, 6.4. JIRA (Project Management) | 11 | 24% | **CO5** |

| TEXT BOOKS | |
|---|---|
| 1 | Foundations of Software Testing by Rex black, Erik Van Veenendaal, Dorothy Graham |
| 2 | Software Engineering by Sommerville-Pearson,8thEdition Daniel Galin, "Software Quality Assurance: From Theory to Implementation", Pearson Addison-Wesley |
| 3 | Effective Methods for Software Testing by William Perry- Wiley Pub, 3rd Edition. |

| REFERENCE BOOKS | |
|---|---|
| 1 | Roger S. Pressman, "Software Engineering-A Practitioner's Approach", McGraw Hill pub.2010 |
| 2 | Software Testing in Real World Edward Kit- Pearson Pub |
| 3 | Software Testing Techniques by Boris Beizer-DreamTech Pub,2nd Edition |
| 4 | Software Testing by Ron Patton, TechMedia Pub. |
| 5 | Introducing Software by Testing Louise Tamres |
| 6 | Fundamentals of Software Engineering –Rajib Mall, 3rd Edition |

| E BOOKS | |
|---|---|
| 1 | "Selenium Testing Tools Cookbook" by Unmesh Gundecha |
| 2 | "Practical Software Testing: A Process-Oriented Approach" by Ilene Burnstein |
| 3 | "Software Testing: Principles and Practices" by Srinivasan Desikan and Gopalaswamy Ramesh |
| 4 | "Testing Computer Software" by Cem Kaner, Jack Falk, and Hung Q. Nguyen |

| MOOC | |
|---|---|
| 1 | Software Testing Fundamentals on Udemy: https://www.udemy.com/course/software-testing-fundamentals-b/?srsltid=AfmBOorJBGnMb1gLUQqFdVFtVrT_NFEwctfh6NHR0ktvBh6orzz_81sH |
| 2 | Software Testing and Automation Specialization on Coursera https://www.coursera.org/specializations/software-testing-automation |
| 3 | Introduction to Software Testing on edX: https://www.edx.org/learn/software-engineering |
| 4 | Testing Automation with Python on Coursera: https://www.coursera.org/specializations/packt-selenium-python-automation-testing-from-scratch-and-frameworks |

| COURSE TITLE | Research Methodology | | | CREDITS | 4 |
|---|---|---|---|---|---|
| COURSE CODE | MCA204 | COURSE CATEGORY | CM | L P T | 3 1 0 |
| Version 1.0 | Approval Details | DDMMYYYY | | LEARNING LEVEL | |

| ASSESSMENT SCHEME | | | | |
|---|---|---|---|---|
| First Periodical Assessment /Seminar | Second Periodical Assessment/ Presentation | Research Paper Writing | Attendance | ESE |
| 10% | 10% | 20% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This detailed syllabus ensures a thorough understanding of research methodology and data science, equipping students with the necessary skills to conduct and evaluate research effectively using modern data science tools and techniques. |
| **Course Objective** | 1. To introduce the fundamental concepts and principles of research.<br>2. To develop an understanding of various research methodologies and techniques, with an emphasis on data science.<br>3. To enhance skills in data collection, analysis, and interpretation using data science tools.<br>4. To foster the ability to critically evaluate research work and methodologies in data science.<br>5. To prepare students for undertaking research projects and writing research reports integrating data science methods. |
| **Course Outcome** | 1. CO1: Understand the basic concepts and principles of research.<br>2. CO2: Apply appropriate research methodologies and data science techniques for different types of research.<br>3. CO3: Demonstrate skills in data collection, analysis, and interpretation using data science tools.<br>4. CO4: Critically evaluate research work and methodologies in the field of data science.<br>5. CO5: Develop a comprehensive research proposal, research paper and write a research report incorporating data science methods. |

**Prerequisites:** Basic understanding of Mathematics and Statistics

| **MODULE 1:** Introduction to Research and Research Process | | | |
|---|---|---|---|
| Definition, Characteristics, Objectives, Types (Basic, Applied, Descriptive, Analytical, Quantitative, Qualitative), Research Process (Problem Identification, Literature Review, Hypothesis Formulation, Research Design, Data Collection, Data Analysis, Interpretation, Reporting), Ethical Considerations in Research | **7** | **11%** | **CO1** |
| **Module 2: Literature Review and Research Design** | | | |
| Importance and Sources of Literature Review, conducting a Systematic Review, Writing Literature Reviews (Structure, Critical Analysis, Synthesis), Research Design (Exploratory, Descriptive, | **8** | **12%** | **CO2** |

| | | | |
|---|---|---|---|
| Experimental), Experimental and Non-Experimental Research, Role of Data Science in Research Design | | | |
| **MODULE 3: Sampling Techniques and Data Collection Methods** | | | |
| Importance of Sampling, Probability & Non-Probability Sampling Techniques, Determining Sample Size, Data Types, Data Collection Methods (Surveys, Interviews, Observations, Secondary Data), Data Collection in Data Science (APIs, Web Scraping, Sensor Data), Data Quality & Cleaning Techniques | **15** | **30%** | **CO3** |
| **MODULE 4: Data Analysis Techniques and Hypothesis Testing** | | | |
| Quantitative Data Analysis (Descriptive Statistics, Inferential Statistics), Qualitative Data Analysis (Thematic & Content Analysis), Introduction to Data Science Techniques (Machine Learning, Data Mining, Big Data Analytics), Statistical Tools & Software (Python, R, SPSS), Hypothesis Formulation & Types (Null, Alternative), Hypothesis Testing Methods (t-test, Chi-square, ANOVA) | **10** | **15%** | **CO4** |
| **MODULE 5: Measurement, Scaling, and Data Interpretation** | | | |
| Types of Measurement Scales (Nominal, Ordinal, Interval, Ratio), Issues of Validity & Reliability, Measurement & Scaling in Data Science (Metrics, KPIs), Data Interpretation Techniques (Graphical, Tabular), Advanced Data Science Techniques (Predictive Analytics, Data Visualization), Tools for Data Visualization (Tableau, Power BI, Matplotlib) | **12** | **20%** | **CO4** |
| **MODULE 6: Report Writing and scholarly publishing** | | | |
| Structure of a Research Report (Introduction, Literature Review, Methodology, Results, Discussion, Conclusion, References), Citation Styles (APA, MLA, Chicago), Components of a Research Proposal, Proposal Writing Techniques, Research Ethical Issues (Plagiarism, Informed Consent, Confidentiality), Ethical Considerations in Data Science (Bias, Fairness, Transparency), Obtaining Ethical Approvals<br>AI Tools in Research, Medley/Zotero, Open-source tools , Turnitin Plaiarism | **8** | **12%** | **CO5** |
| **TEXTBOOKS** | | | |
| 1. | Creswell, J. W., & Creswell, J. D. (2017). *Research design: Qualitative, quantitative, and mixed methods approaches* (5th ed.). Thousand Oaks, CA: Sage Publications. | | |
| 2 | Kumar, R. (2019). *Research methodology: A stepbystep guide for beginners* (5th ed.). Los Angeles, CA: Sage Publications. | | |
| **REFERENCE BOOKS** | | | |
| 1. | Leedy, P. D., & Ormrod, J. E. (2014). Practical research: Planning and design (10th ed.). Boston, MA: Pearson. | | |
| 2. | American Psychological Association. (2020). Publication manual of the American Psychological Association (7th ed.). Washington, DC: Author. | | |
| **E BOOKS** | | | |
| 1. | Dawson, C. (2017). Introduction to research methods: A practical guide for anyone undertaking a research project (6th ed.). Retrieved from eBook Central. | | |
| 2 | Saunders, M., Lewis, P., & Thornhill, A. (2019). Research methods for business students (8th ed.). Retrieved from VitalSource. | | |
| **MOOC** | | | |
| 1. | Coursera: Research Methods: A Practical Guide | | |
| 2. | edX: Introduction to Research for Essay Writing | | |
| 3. | MOOC from University of London: Understanding Research Methods | | |

| COURSE TITLE | | Mini PROJECT | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA205 | COURSE CATEGORY | Mini Project | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |
| ASSESSMENT SCHEME | | | | | | |
| Review 1 | Review 2 | Review 3 | Final Review | Attendance | ESE | |
| 10% | 10% | 20% | 40% | 20% | -- | |

# Mini Project Schedule

| Sr.No | Reviews | Topic | Documents Required |
|---|---|---|---|
| 1 | | Project Title Submission | **Project Title** |
| 2 | Review 1 | Project Synopsis | **INTRODUCTION**<br>• Existing System and Need for System<br>• Scope of Work<br>• Operating Environment - Hardware and Software Detail Description of Technology Used<br>**PROPOSED SYSTEM**<br>• Proposed System<br>• Objectives of System<br>• User Requirements |
| 3. | Review 2 | Design | **ANALYSIS & DESIGN**<br>• Module Hierarchy Diagram<br>• Use Case Diagrams<br>• Class Diagram<br>• E-R Diagram<br>• Activity Diagram<br>• Sequence Diagram<br>• Web Site Map Diagram (in case of Web Site) |
| 5 | Review 3 | Database Connectivity | • User Interface Design (Screens etc.)<br>• Database Table Structure<br>• Database Connectivity<br>• Form Validation, Reports |

| | | | |
|---|---|---|---|
| 6 | Review 4 | Final Submission | Final Project Execution, checking Complete Documentation of Project with all Diagrams and PPT, Viva and submission of Spiral Binding documentation. |

# Mini Project Guidelines

1. Students can work individually or in pairs (maximum of 2) for the Mini Project. Projects must relate to both laboratory subjects being taught in the current semester. The student may take up the project individually or in group. If project is done in group, each student must be given a responsibility for distinct modules.
2. Few hours per week is dedicated to project work. During this time, students must report to their assigned guides for support and discuss project progress. Attendance is mandatory for all review sessions and weekly project hours; project diaries must be presented at each review. (check class time table for more details)
3. Students must submit a clear and self-descriptive project title by the end of the first week of the semester. Following guide allotment, a project synopsis must be submitted by the second week, including sections on existing systems, need for the project, scope of work, objectives, and user requirements.
4. There will be a total of 4 reviews throughout the semester. Specific deliverables are required at each stage, including design documents, implementation details, and testing strategies. Students are expected to create a PowerPoint presentation for each review, summarizing their progress.
5. Documentation must adhere to specified standards: use Times New Roman, size 12 pt, with 1.5line spacing and justified alignment. Margins should be set to 1" (left and right), 2" (top), and 1.5" (bottom). All documentation must be spiral-bound and submitted by the final deadline.
6. The final evaluation will consist of a project presentation (PowerPoint) and a viva voce conducted by faculty members. Students must submit a comprehensive project report, including all relevant diagrams and design documents, on the final submission date.
7. All project work must be original and must adhere to the ICEM's academic integrity policies. Plagiarism will not be tolerated and may result in academic penalties.
8. Students are encouraged to utilize resources such as textbooks, online articles, and tutorials to support their project development. A list of recommended resources should be submitted along with the final project report.
9. Feedback from guides must be incorporated into subsequent project phases. Students should schedule regular meetings (in Mini Project slot) with their respective guides to ensure alignment and address any challenges faced during project development.
10. Any changes to the project scope or objectives must be discussed with and approved by the guide before implementation. Documentation of such changes should be maintained in the project diary.
11. All project-related communications, including submission links and updates, will be shared via email and designated online platforms. Students should regularly check these platforms for important announcements.

| **MOOC** | |
|---|---|
| 1. | **Coursera - "Project Management Principles and Practices" by Google**<br>https://www.coursera.org/specializations/project-management |

| | |
|---|---|
| 2. | **edX - "Mini  Project: Data Science for Everyone" by IBM**<br>https://www.edx.org/professional-certificate/ibm-data-science |

| COURSE TITLE | | Cloud Computing Management and Security LAB | | | CREDITS | | 1 |
|---|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206L | COURSE CATEGORY | SEC | L-P-T | | 0-1-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | | |
| ASSESSMENT SCHEME | | | | | | | |
| Assignments | | Internal Exam | | | Attendance | | ESE |
| 40% | | 40% | | | 20% | | -- |
| MODULE: Topic | | | | | Sessions | Weightage (%) | CO Mapping |
| **MODULE 1: Introduction to Cloud Computing Management** | | | | | | | |
| **Task 1.** Deploy a Web App on a Cloud Platform (Cloud Management Principles & Infrastructure Management)<br>**Task 2.** Estimate Cloud Costs Using Cost Calculators (Cloud Cost Management)<br>**Task 3**. Set Up Basic Monitoring & Alerts (Cloud Monitoring and Optimization) | | | | | 4 | 10% | CO-1 |
| **MODULE 2: Cloud Security** | | | | | | | |
| **Task 1.** Simple Data Migration to the Cloud<br>**Task 2**. Identify the Best Cloud Migration Strategy<br>**Task 3.** Enable Basic Security Settings on a Cloud Resource | | | | | 4 | 10% | CO-2 |
| **MODULE 3: Cloud Database and File System** | | | | | | | |
| **Task 1.** Create a Simple Data Warehouse Schema in Google BigQuery<br>**Task 2.** Upload and Organize Data in a Cloud-Native File System<br>**Task 3.** Explore General-Purpose Cloud Storage<br>**Task 4.** Hands-On with Different Cloud Storage Types | | | | | 6 | 20% | CO-3 |
| **MODULE 4: Managing Cloud Infrastructure** | | | | | | | |
| **Task 1**. Create and Manage Virtual Machines (Virtualization)<br>**Task 2.** Implement Cloud Storage Management with AWS S3 or Google Cloud Storage<br>**Task 3**. Configure and Monitor Cloud Networking (Cloud Network Management) | | | | | 6 | 20% | CO-4 |
| **MODULE 5: Cloud Security Practices and Solutions** | | | | | | | |
| .**Task 1.** Implement server-side encryption for files uploaded to a cloud storage service (e.g., AWS S3 or Google Cloud Storage).<br>**Task 2**. Set up SSL/TLS encryption for a website hosted in the cloud to ensure secure communication between the user and the server. | | | | | 5 | 20% | CO-5 |
| **MODULE 6: Cloud Security & Implementation of Cloud** | | | | | | | |

| | | | |
|---|---|---|---|
| **Task 1.** Configure a basic security group on a cloud provider (e.g., AWS, Azure) to restrict access to a VM or service.<br><br>**Task 2**. Design a secure cloud architecture for a hypothetical company using best practices for identity management, data protection, and access control.<br><br>**Task 3**. Implement data encryption at rest and in transit for a cloud-hosted database or application. | **5** | **20%** | **CO-5** |

| COURSE TITLE | | Cloud Computing Management & Security | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206 | COURSE CATEGORY | ME | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 07- 2024 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course explores the principles and practices of cloud computing management and security. It covers the deployment, management, and security of cloud infrastructure and services, with an emphasis on practical skills and real-world applications. |
| **Course Objective** | 1. Understand the management and operational aspects of cloud computing.<br>2. Explore the security challenges and solutions in cloud environments.<br>3. Learn about cloud service models and deployment strategies.<br>4. Develop skills for managing cloud infrastructure and services.<br>5. Gain hands-on experience with leading cloud platforms. |
| **Course Outcome** | CO1: **Understand** cloud computing management principles and best practices.<br>CO2: **Analyze** security challenges and implement security measures in cloud environments.<br>CO3: **Utilize** cloud service models and deployment strategies effectively.<br>CO4: **Manage** and optimize cloud infrastructure and services.<br>CO5: **Develop** and deploy secure cloud-based applications. |

**Prerequisites: Basic knowledge of fundamentals of Cloud**

| Module Name | No of lectures | Weightage | CO-PO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Cloud Computing Management** | | | |
| 1.1 Cloud management principles<br>1.2 Cloud infrastructure management,<br>1.3 Cloud service management,<br>1.4 Cloud cost management,<br>1.5 Cloud monitoring and optimization | 5 | 10% | **CO-1** |
| **MODULE 2: Cloud Security** | | | |
| 2.1 Data Migration in Cloud.<br>2.2 Cloud Migration Strategies and Process(Six R for Cloud Migration)<br>2.3 Cloud Security Fundamentals<br>2.4 Cloud Computing Security Challenges<br>2.5 Privacy and Security in the Cloud<br>2.6 Quality of Services in Cloud Computing (QoS). | 8 | 15% | **CO-2** |
| **MODULE 3: Cloud Database and File System** | | | |
| 3.1 Core concepts of data warehousing<br>3.2 Primary components and architectures of data warehousing<br>3.3 Cloud Native file system<br>3.4 General Purpose Cloud Storages<br>3.5 Storage Types | 8 | 20% | **CO-3** |

| | | | |
|---|---|---|---|
| Hands on practical based on this module. | | | |
| **MODULE 4: Managing Cloud Infrastructure** | | | |
| 4.1 Virtualization<br>4.2 Cloud storage management<br>4.3 Cloud network management<br>4.4 High availability and disaster recovery<br>4.5 Cloud performance optimization<br><br>Hands on practical based on this module. | 8 | 20% | **CO-4** |
| **MODULE 5: Cloud Security Practices and Solutions** | | | |
| 5.1 Basic security concepts: encryption,<br>5.2 IAM (Identity and Access Management), network security<br>5.3 AWS security services overview<br>5.4 Managing access with AWS IAM<br>5.5 Case Study on Cloud Security.<br><br>Hands on practical based on this module. | 8 | 20% | **CO-5** |
| **MODULE 6: Cloud Security & Implementation of Cloud** | | | |
| 6.1. Cloud Security Fundamentals<br>6.2. Cloud Security Architecture<br>6.3. Cloud Computing Security Challenges<br>6.4. Privacy and Security in Cloud<br>6.5. Identity Management and Access control<br>6.6. Demonstrate the commercial cloud computing Infrastructures<br>6.7. Introduction to Dockers Container<br>6.8. Case Study's based on Cloud Computing Concepts.<br><br>Hands on practical based on this module. | 8 | 15% | **CO-5** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | 1. "Cloud Computing: Concepts, Technology & Architecture" by Thomas Erl<br>2. "Cloud Security and Privacy" by Tim Mather |
| **REFERENCE BOOKS** | |
| 1. | 1. "Mastering Cloud Computing" by Rajkumar Buyya<br>2. "Architecting the Cloud" by Michael J. Kavis |
| **E BOOKS** | |
| 1. | "Cloud Computing Explained" by John Rhoton (computingclouds.files.wordpress.com/2012/05/cloud-computing-explained.pdf) |
| 2 | "NIST Cloud Computing Standards Roadmap" (nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-291.pdf) |
| **MOOC** | |
| 1. | "Google Cloud Platform Fundamentals: Core Infrastructure" by Google Cloud on Coursera (coursera.org/learn/gcp-fundamentals) |
| 2. | "Introduction to Cloud Computing" by IBM on Coursera (coursera.org/learn/introduction-to-cloud) |

| COURSE TITLE | | ADVANCE WEB TECHNOLOGIES LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: React JS**

1. **Dynamic E-commerce Product Page**: Develop a product display component using **PROPS**, manage availability status with **useState**, and implement a pop-up on **click events**.
2. **Multi-Step Form for Job Application**: Create a **React form** with input validation using **useState and useEffect**, enable navigation with **Redux**, and allow users to save progress.
3. **Ride-Sharing App with React Router**: Implement **React Router** for pages like Home, Book Ride, and Ride History, use **Router Links**, and manage ride data with **Redux**.
4. **Movie Recommendation System with Custom Hooks**: Build a recommendation system using a **Custom Hook** for API fetching, update suggestions dynamically with **useEffect**, and track selected movies using **useState**.
5. **Social Media Dashboard with Real-Time Updates**: Create a post feed using **useState** for dynamic updates, fetch posts with **useEffect**, and manage likes/comments globally with **Redux**.
6. **Event Booking System with State Management**: Design an event listing component with **PROPS**, handle bookings dynamically with **useState**, and manage global event availability using **Redux**.

**MODULE 2: Node JS**

1. **Real-Time Chat Application**: Develop a real-time chat app using **Node.js** and **WebSockets**, manage events with the **EventEmitter module**, and handle multiple user connections dynamically.
2. **CLI-Based Task Manager**: Create a **command-line task manager** using **Node.js REPL**, allowing users to add, delete, and list tasks, and integrate **NPM packages** for file storage.
3. **E-commerce Backend with NPM Modules**: Build an **e-commerce backend** using **Node.js**, manage dependencies with **NPM**, and structure the project using **custom modules** for handling users, products, and orders.
4. **Web Server for Blog Platform**: Develop a simple **web server** using **Node.js HTTP module**, serve HTML pages dynamically, and implement **routing** for different blog sections.
5. **Event-Driven File Logger**: Implement a **file logging system** using **Node.js events**, where log messages are stored in a file when triggered by different system actions.
6. **API Server for IoT Data Processing**: Create a **Node.js API** to receive real-time IoT sensor data, process it using **event-driven architecture**, and store logs using **file system modules**.

**MODULE 3: Angular JS**

1. **E-Learning Platform with Dynamic Content**: Develop an **Angular SPA** for an e-learning platform, using **modules and components** to display courses, and implement **routing** for navigating between lessons.
2. **Real-Time Weather Dashboard**: Build a **weather dashboard** using **Angular services** to fetch live data, apply **dependency injection** for API calls, and display results dynamically with **data binding**.
3. **Task Management App with Two-Way Binding**: Create a **task manager** where users can add, edit, and delete tasks, utilizing **two-way data binding** for instant UI updates and **event binding** for task actions.

4. **E-commerce Product Filter System**: Implement an **Angular product catalog** with **directives and filters** to allow users to search and filter products dynamically based on categories and price range.
5. **Finance Dashboard with String Interpolation**: Develop a **finance dashboard** that displays **real-time stock market data** using **string interpolation** to update UI dynamically with fetched data.
6. **Online Booking System with Angular Routing**: Build an **online booking system** for hotels, implementing **Angular routing** for seamless page transitions and **SPA architecture** for a smooth user experience.

**MODULE 4: Mongo DB**

1. **User Management System with Authentication**: Develop a **user registration and login system** using **MongoDB and Mongoose**, implement **authentication and authorization** to restrict access to certain pages.
2. **E-Commerce Inventory Management**: Create an **inventory management system** with **CRUD operations using Mongoose**, ensuring proper **data validation and schema relationships** for products, categories, and suppliers.
3. **Blog Platform with Profanity Filtering**: Build a **blogging platform** where users can post articles, implement **profanity filtering and post moderation** to automatically review and flag inappropriate content.
4. **Real-Time Analytics Dashboard with Caching**: Design a **real-time analytics dashboard** that stores user interactions in **MongoDB**, optimize performance with **caching techniques**, and prevent duplicate API calls.
5. **Secure API for Financial Transactions**: Implement a **secure REST API** for processing transactions, integrate **authentication and authorization** with JWT, and protect sensitive endpoints from unauthorized access.
6. **Online Learning Platform with Role-Based Access**: Develop an **e-learning platform** where users have different roles (admin, instructor, student), using **MongoDB relationships** to manage course enrollments and **protect routes** based on roles.

**Module 5: Express JS**

1. **Task Management API with Express**: Develop a **RESTful API** using **Express.js** for a task management system, implementing **CRUD operations** and handling requests with **middleware**.
2. **E-Commerce Product Catalog API**: Create an **Express.js API** to serve product details, implement **routing** for different product categories, and use **middleware** for logging and authentication.
3. **Movie Review Platform with Templating**: Build a **movie review website** using **Express.js and a templating engine (EJS/Pug)** to render dynamic content and implement **route-based navigation**.
4. **Authentication System with Middleware**: Design an **Express-based login system** using **JWT authentication**, protect routes with **custom middleware**, and handle role-based access control.
5. **News Aggregator with API Integration**: Create a **news aggregator platform** using **Express.js**, fetch news from external APIs, and use **middleware** for request validation and rate limiting.
6. **Food Delivery App Backend**: Develop a **food delivery backend API** using **Express.js**, implement **nested routing** for users, restaurants, and orders, and optimize request handling with **middleware functions**.

| COURSE TITLE | | ADVANCE WEB TECHNOLOGIES | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206 | COURSE CATEGORY | ME2.2 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments /Project | Practical Assessment/ Test | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | Advance Web Technologies Web development typically refers to the coding and programming side of website production. When you learn web development, you might start out writing a simple page of HTML text and build up to creating more complex, feature-rich applications designed to be accessed from various Internet-connected devices. |
| **Course Objective** | 1. To impart the design, development and implementation of Dynamic Web Pages.<br>2. To implement the Latest properties of Node, Angular, Express<br>3. To implement the Concept of NodeJS.<br>4. To develop programs for the Web using Angular and SPA.<br>5. To design and implement dynamic websites with a good sense of design and the latest technical aspects. |
| **Course Outcome** | **CO1: Understand** Outline the basic concepts of Advance Internet Technologies<br>**CO2: Apply** Design appropriate user interfaces and implements webpage based on given problem Statement<br>**CO3: Apply** Implement concepts and methods of NodeJS<br>**CO4: Apply** Implement concepts and methods of Angular<br>**CO5: Analyze** Build Dynamic web pages using server-side programming with Database Connectivity |
| **Prerequisites:** Student must have hands-on working knowledge of HTML, CSS, JavaScript | |

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: React JS** | | | |
| 1. 1 Introduction and Setup to React ,1.2 Component-based Architecture 1.3 Handling Events and Forms, 1.4 React Router, 1.5 State Management with Redux, 1.6 React Hooks App Component and JSX Functional, Component, Adding CSS, 1.7 Click Events, 1.8 PROPS, Forms, Use State, Use Effects, Custom Hook, 1.9 Router Links, State Management | **10** | **20%** | **CO-1** |
| **MODULE 2: Node JS** | | | |
| 2.1 Node JS, 2.2 introduction and how it works, 2.3 installation of node js, 2.4 REPL, 2.5 NPM, 2.6 How modules work, Webserver Creation, 2.7 Events | **8** | **20%** | **CO2** |
| **MODULE 3: Angular JS** | | | |
| 3.1 Angular (Latest Stable Version), 3.2 Introduction (Features and Advantage),3.3 Type Script, Modules, Components, 3.4 Directives, Expression, Filters, Dependency Injection, 3.5 Services, 3.6 Routing, SPA (Single Page Application), Data binding, property binding, Event Binding, Two-way data binding, String Interpolation. | **10** | **20%** | **CO-3** |
| **MODULE 4: Mongo DB** | | | |

| | | | |
|---|---|---|---|
| 4.1 SQL and NoSQL Concepts, 4.2 Database Integration – Mongo DB, 4.3 CRUD with Mongoose & Mongo DB I, 4.4 CRUD with Mongoose & Mongo DB II, 4.5 Mongo Data Validation, Modelling Relationships, 4.6 Authentication, 4.7 Profanity Filtering and Post Moderation, 4.8 Caching & Performance, 4.9 Protecting express App | **10** | **20%** | **CO-4** |
| **Module 5: Express JS** | | | |
| 5.1 Introduction to Express ,5.2 Building Restful APIs using express, 5.3 Express Middleware's, 5.4 Rendering/Template engines, Routing<br><br>Hands on practice on topics learned in this module as listed below: | **4** | **10%** | **CO-5** |
| **Module 6: Hands on Application Development** | | | |
| **Front-end:** Creating interactive user interfaces with React/Angular.<br>**Back-end:** Developing APIs with Node.js and Express.js.<br>**Database:** Designing and interacting with a MongoDB database.<br><br>The emphasis is on practical experience. need to develop code, debug, test, and potentially work in a team.<br><br>Essentially, this section aims to equip you with the skills to build real-world web applications using industry-standard technologies. | **3** | **10%** | **ALL CO** |

| | |
|---|---|
| **TEXT BOOKS** | |
| **1.** | **"Data Structures and Algorithms with JavaScript" by Michael McMillan**<br>**https://www.amazon.com/Data-Structures-Algorithms-Michael-McMillan/dp/1449364934** |
| **2** | **"JavaScript: The Good Parts" by Douglas Crockford**<br>**https://www.amazon.com/JavaScript-Good-Parts-Douglas-Crockford/dp/0596517742** |
| **REFERENCE BOOKS** | |
| **1.** | **"Eloquent JavaScript" by Marijn Haverbeke**<br>**https://eloquentjavascript.net** |
| **2.** | **"Introduction to Algorithms" by Cormen, Leiserson, Rivest, and Stein**<br>**https://www.amazon.com/Introduction-Algorithms-3rd-MIT-Press/dp/0262033844** |
| **E BOOKS** | |
| **1.** | **"JavaScript Data Structures and Algorithms" by Sammie Bae**<br>**https://opendatastructures.org/ods-cpp.pdf** |
| **2** | **"Open Data Structures (in pseudocode)" by Pat Morin:**<br>**https://opendatastructures.org/ods-python.pdf** |
| **MOOC** | |
| **1.** | **"Algorithms and Data Structures" on edX**<br>**https://www.edx.org/course/algorithm-and-data-structures** |
| **2.** | **NPTEL Data Structures and Algorithms**<br>**https://archive.nptel.ac.in/courses/106/102/106102064/** |

| COURSE TITLE | | Artificial Intelligence & Machine Learning Lab | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206L | COURSE CATEGORY | ME(Elective 3) | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 10% | - |

**MODULE 1: Introduction to Artificial Intelligence**

1. AI-Powered Chatbot for Customer Support
2. Predictive Analytics for Sales Forecasting
3. AI-Based Image Recognition for Object Detection

**MODULE 2 : Search and Optimization Techniques**

1. Develop a program to find a path between two points in a maze using Breadth-First Search (BFS) and Depth-First Search (DFS).
2. Implement A* and Greedy Best-First Search to solve the classic 8-puzzle problem (a sliding puzzle with 3x3 grid).
3. Solve a Sudoku puzzle using Constraint Satisfaction Problem (CSP) techniques and Genetic Algorithms.

**MODULE 3: Machine Learning for AI**

1. Write a Python program demonstrating how a simple ML model (e.g., Linear Regression) contributes to an AI system.
2. Analyze and demonstrate bias issues in a simulated AI recruitment model trained on historical hiring data.
3. Implement a smart surveillance system that detects and tracks faces in real-time for security purposes.

**MODULE 4: Introduction to Machine Learning**

1. Use Supervised Learning (Regression) to predict house prices based on features like area, number of rooms, and location.
2. Use Unsupervised Learning to segment customers based on purchasing behavior.
3. Classify emails as spam or not spam using Decision Trees, Naïve Bayes, and SVM.)
4. Design an AI system that reads user movie reviews and automatically classifies them as positive or negative.

**MODULE 5: Machine Learning Workflow and Tools**

1. Define a machine learning problem to predict whether a customer will churn based on historical service data (e.g., tenure, usage, complaints).
2. Collect and clean a dataset for housing prices (e.g., size,

location, number of rooms). Handle missing values, outliers, and convert categorical variables.

3. Select and evaluate models (Logistic Regression, Decision Tree, SVM) to predict credit score levels (Good/Bad). Use metrics like Accuracy, Precision, Recall, F1-score.
4. Write a Python script to load a student dataset, preprocess it, and build a regression model to predict final exam scores.
5. Use NumPy, Pandas, and Scikit-learn to build and evaluate a classification model on the Pima Indians Diabetes dataset.

**MODULE 6: Case Studies**

1. Build a regression model to predict daily or monthly sales for a retail store based on historical sales data, promotions, holidays, store location, and footfall.
2. Predict weekly revenue for a restaurant using features like customer reviews, location, cuisine type, and past performance.
3. Develop a regression model to estimate a player's salary based on performance metrics such as goals scored, matches played, assists, and market value.
4. Build a classification model to predict whether a patient is likely to develop diabetes based on features such as BMI, age, glucose level, and blood pressure.
5. Detect fraudulent transactions using machine learning models trained on transaction history, location, and spending patterns.

| COURSE TITLE | Artificial Intelligence & Machine Learning | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA206 | COURSE CATEGORY | ME(Elective 3) | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces the fundamentals of Artificial Intelligence (AI) and Machine Learning (ML), including intelligent agents, search techniques, knowledge representation, supervised and unsupervised learning, model evaluation, and real-world case studies. Students will gain both theoretical knowledge and hands-on experience using Python and popular libraries such as NumPy, Pandas, and Scikit-learn, preparing them to apply AI/ML techniques to solve practical problems ethically and efficiently. |
| **Course Objective** | To provide foundational knowledge of AI, its history, applications, and ethical aspects.<br>To develop problem-solving skills using search strategies and optimization techniques.<br>To introduce core machine learning concepts including supervised, unsupervised, and reinforcement learning.<br>To equip students with the skills to preprocess data, select appropriate models, and evaluate performance.<br>To familiarize students with modern ML tools and libraries for implementation.<br>To apply AI/ML concepts through case studies in various domains like healthcare, retail, and finance. |
| **Course Outcome** | By the end of the course, students will be able to:<br><br>CO1: Remember the history, applications, and terminology of AI and ML.<br>CO2: Understand intelligent agents, search techniques, and ML types.<br>CO3: Apply AI search algorithms and ML models to solve real-world problems<br>CO4: Analyze model performance using metrics like accuracy, precision, and recall.<br>CO5: Evaluate different algorithms for suitability in specific domains.<br>CO6: Create end-to-end AI/ML solutions using appropriate tools and libraries. |

| MODULE | No. of lectures | Weightage (%) | CO Mapped |
|---|---|---|---|
| **MODULE 1: Introduction to Artificial Intelligence** | | | |
| 1.1 Definition and History of AI,<br>1.2 Applications of AI<br>1.3 Intelligent Agents and Environments<br>1.4 Problem-Solving using AI<br>1.5 AI Ethics and Social Implications | 5 | 10% | CO1,CO2 |
| **MODULE 2 : Search and Optimization Techniques** | | | |
| 2.1 Uninformed Search Strategies (BFS, DFS), 2.2 Informed Search (A*, Greedy Search), 2.3 Constraint Satisfaction Problems (CSP), 2.4 Genetic Algorithms and Optimization | 6 | 12% | CO3,CO4 |
| **MODULE 3: Machine Learning for AI** | | | |

| | | | |
|---|---|---|---|
| 3.1 Introduction to Relationship between AI and ML, 3.2 Applications of ML in AI Domains 3.3 ML Techniques Powering AI Systems, ML Models Used in AI, Challenges and Future Scope | 10 | 18% | CO5,CO6 |
| **MODULE 4: Introduction to Machine Learning** | | | |
| 1.1 History and evolution of Machine Learning 1.2 Applications in industries: Healthcare, Finance, Marketing, 1.3 Types of Machine Learning : Supervised, Semi supervised, Unsupervised | 8 | 20% | CO1,CO2 |
| **MODULE 5: Machine Learning Workflow and Tools** | | | |
| 2.1 Problem formulation, 2.2 Data collection and pre-processing, 2.3 Model selection and evaluation, 2.4 Overview of Python for ML, 2.5 Introduction to ML libraries: NumPy, Pandas, Scikit-learn | 6 | 15% | CO3,CO4 |
| **MODULE 6: Case Studies** | | | |
| 6.1 REGRESSION Case Studies 6.2 Retail Store Sales Prediction 6.3 Restaurant Sales Prediction 6.4 Sport Player Salary Prediction 6.5 CLASSIFICATION Case Studies 6.6 Diabetes Prediction for Preventive Care 6.7 Credit Card Fraud Detection 6.8 Heart Diseases Prediction for Preventive Care | 10 | 25% | CO5,CO6 |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1 | "Artificial Intelligence: A Modern Approach" *by Stuart Russell and Peter Norvig (4th Edition, Pearson, 2020)* |
| 2 | “Machine Learning: A Probabilistic Perspective" *by Kevin P. Murphy (MIT Press, 2012)* |
| 3 | "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" *by Aurélien Géron (2nd Edition, O'Reilly Media, 2019)* |
| **REFERENCE BOOKS** | |
| 1 | "Pattern Recognition and Machine Learning" *by Christopher M. Bishop* |
| **E BOOKS** | |
| 1 | "Deep Learning" *by Ian Goodfellow, Yoshua Bengio, and Aaron Courville* |
| 2 | "Python Machine Learning" *by Sebastian Raschka and Vahid Mirjalili* |
| **MOOC** | |
| 1 | Coursera: "Introduction to Data Science" by University of Washington |
| 2 | AI For Everyone *by Andrew Ng on Coursera*<br><br>Machine Learning *by Andrew Ng on Coursera* |

| COURSE TITLE | | Blockchain LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA207L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Introduction to Blockchain**

1. Simulate a centralized vs decentralized system using basic Node.js or any flow diagramming tool to compare their control and failure points.
2. Identify 3 different real-world systems and categorize them into Public, Private, or Consortium blockchain types with justification.
3. Create a JSON-based ledger and manually simulate how changes in a block can invalidate a chain without hash integrity.
4. Use a visual tool (e.g., IBM Blockchain 101 simulator) to simulate how blockchain transactions are linked.
5. Compare a traditional relational database table with a blockchain block structure and document key differences.
6. Research and prepare a flowchart showing the evolution of blockchain from Bitcoin to modern decentralized applications (DApps).

**MODULE 2: Blockchain Architecture & Cryptographic Foundations**

1. Write a program to compute SHA-256 hash of a given text (use any language like JavaScript, Python, or Java).
2. Create a Merkle tree manually for four transactions and calculate the root hash using a hash function.
3. Simulate digital signature verification using a public-private key pair (using Node.js crypto library or OpenSSL).
4. Build a prototype JSON block containing index, timestamp, data, previousHash, and hash.
5. Demonstrate how a small change in input results in a completely different SHA-256 hash (Avalanche Effect).
6. Illustrate and document the structure of a blockchain node (peer, full node, miner) using diagrams and explanation.

**MODULE 3: Consensus Mechanisms**

1. Simulate Proof of Work in code: find a hash value with a specific number of leading zeros.
2. Create a game-based activity to simulate PoW and PoS among students acting as miners and validators.
3. Compare time and energy consumption between a simple PoW simulation and a pseudo PoS algorithm.
4. Build a simulation to show how PBFT consensus works with 4 nodes, including handling 1 faulty node.
5. Conduct a scenario analysis showing how consensus prevents double-spending in Bitcoin.
6. Create a flowchart to compare consensus mechanisms (PoW, PoS, DPoS) based on security, scalability, and decentralization.

**MODULE 4: Cryptocurrency and Bitcoin Blockchain**

1. Simulate a Bitcoin transaction from one wallet to another using a Bitcoin testnet and document the process.
2. Demonstrate how to generate a Bitcoin address using command-line or a library like `bitcoinjs-lib`.
3. Trace a real Bitcoin transaction on the blockchain explorer and interpret its fields (inputs, outputs, fees).
4. Set up a simple blockchain in JavaScript and simulate a mining process by adjusting difficulty levels.
5. Create a mini ledger of transactions and demonstrate how miners pick and validate them into a block.
6. Document a Bitcoin fork (e.g., Bitcoin Cash) and simulate what happens when two chains temporarily exist.

**Module 5: Ethereum and Smart Contracts**

1. Write and deploy a smart contract using Solidity on Remix that stores and retrieves student information.
2. Create a token using ERC-20 standard on the Ethereum testnet using MetaMask and Remix.
3. Write a Solidity contract for a basic voting system and simulate voting with 3 candidates.
4. Demonstrate how gas is calculated by deploying a loop-based smart contract and measuring gas usage.
5. Create a DApp frontend using HTML and Web3.js to interact with a smart contract deployed on Remix.
6. Develop a smart contract for a simple escrow payment system between buyer and seller.

| COURSE TITLE | | IOT LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA207L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Introduction to IoT**

1. Simulate a real-life IoT use case (e.g., smart home) and identify its components from the IoT ecosystem.
2. Create a comparative chart of IoT vs M2M vs CPS using any digital tool (Google Sheets, Excel, etc.).
3. Identify and list IoT applications around your college campus with justification for each.
4. Create a presentation or infographic that explains the evolution of IoT from RFID to AIoT.
5. Given 3 use cases (e.g., smart agriculture, home automation, and traffic monitoring), classify the IoT functional blocks used in each.
6. Develop a block diagram for any IoT system and annotate its components from the ecosystem.

**MODULE 2: IoT Architecture and Protocols**

1. Design a 3-layer architecture diagram of an IoT system and label each layer with components.
2. Use diagrams or a simulator to show the working of HTTP vs MQTT in an IoT data transmission scenario.
3. Simulate an IoT communication scenario using Node-RED or a simple MQTT broker (e.g., Mosquitto).
4. Create a mapping of functional blocks (sensing, network, processing) to a real-life IoT application (e.g., smart parking).
5. Prepare a comparative table highlighting features of HTTP, MQTT, CoAP, and AMQP.
6. Configure a basic MQTT publisher and subscriber model using online MQTT tools or IDE.

**MODULE 3: Sensors, Actuators & Embedded Systems**

1. Connect a temperature sensor (e.g., LM35) to Arduino and display output on the serial monitor.
2. Interface an IR sensor with Arduino and simulate obstacle detection.
3. Simulate an actuator (LED/Buzzer) response when sensor threshold is crossed.
4. Develop a program using Arduino IDE that reads analog input and displays converted digital value.
5. Identify five different sensors and actuators, and simulate their use in respective IoT applications using Tinkercad or Proteus.
6. Design a simple embedded IoT system (e.g., room light automation) using NodeMCU or Raspberry Pi.

**MODULE 4: Communication Models and IoT Protocol Stack**

1. Create a layered protocol stack diagram for a chosen IoT device and explain each layer's function.
2. Simulate data transfer using BLE in any IoT mobile app (e.g., nRF Connect or Blynk with Arduino).
3. Implement RESTful API in a microcontroller (NodeMCU) to send sensor data to a cloud platform.
4. Create a chart comparing BLE, Zigbee, Wi-Fi, LoRa, and RFID for speed, range, and use case.
5. Demonstrate WebSocket vs HTTP in data exchange using online simulators.
6. Build a mock setup where sensor data is collected and pushed to cloud (simulate edge communication).

**Module 5: IoT Applications in Real World**

1. Design a solution architecture for a smart agriculture system (e.g., auto irrigation using soil moisture).
2. Create a mock dashboard (use Google Sheets or ThingSpeak) for smart city air quality monitoring.
3. Simulate a health monitoring system using pulse sensor on Arduino and plot the graph on serial monitor.
4. Identify 3 industrial IoT systems and draft a comparison report on technology stack used.
5. Choose one IoT-based startup and present how they apply IoT in their solution stack.
6. Build a virtual prototype of a smart streetlight system using sensors and relay module in Tinkercad.

| COURSE TITLE | | IOT: INTERNET OF THINGS | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA207 | COURSE CATEGORY | ME3.1 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 04/2025 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10 % | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces the fundamental concepts of the Internet of Things (IoT), exploring its applications, technologies, and the methodologies used for designing IoT solutions. Students will gain knowledge about IoT architectures, protocols, sensors, and the integration of IoT with cloud computing, big data, and machine learning. |
| **Course Objective** | • To understand the core concepts and technologies of IoT.<br>• To learn about the architecture and protocols used in IoT systems.<br>• To explore the various IoT applications in different domains.<br>• To design and implement IoT solutions using hardware and software tools.<br>• To integrate IoT with emerging technologies like cloud computing and big data. |
| **Course Outcome** | • CO1: Explain the fundamental concepts and architecture of IoT.<br>• CO2: Analyze various IoT protocols and communication models.<br>• CO3: Design and develop IoT solutions using appropriate hardware and software.<br>• CO4: Implement IoT solutions in different application domains.<br>• CO5: Integrate IoT systems with cloud services and big data analytics.<br>• CO6: Evaluate security and privacy issues in IoT deployments. |

**Prerequisites:**

- Basic knowledge of computer networks
- Understanding of programming languages (Python/C/C++)

| MODULE 1: INTRODUCTION TO IOT | | | |
|---|---|---|---|
| **Topic:**<br>Definition and evolution of IoT, architectures and models<br>Key components and technologies,<br>enabling technologies: RFID, sensor networks and cloud computing, IoT system design methodologies, Case studies of successful IoT implementations, IoT market trends and future directions<br><br>**Suggested Readings:**<br>Internet of Things: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti | **9** | **15** | **CO-1** |
| **MODULE 2: IOT PROTOCOLS AND COMMUNICATION** | | | |
| **Topic:** IoT communication models, Protocols: MQTT, CoAP, AMQP, and HTTP, Wireless communication technologies: Wi-Fi, Bluetooth, Zigbee, LoRa, and NB-IoT<br><br>**Suggested Readings:** | **12** | **20** | **CO-2** |

| | | | |
|---|---|---|---|
| The Internet of Things: Key Applications and Protocols by Olivier Hersent, David Boswarthick, and Omar Elloumi | | | |
| **MODULE 3: IOT HARDWARE PLATFORMS AND DEVICES** | | | |
| **Topic:** Sensors, actuators, and microcontrollers, Raspberry Pi, Arduino, and other IoT development boards, Interfacing sensors and actuators with microcontrollers<br><br>**Suggested Readings:** Reference Book: Raspberry Pi Cookbook by Simon Monk | **12** | **20** | **CO-3** |
| **MODULE 4: IOT APPLICATION DEVELOPMENT** | | | |
| **Topic:** IoT application domains: Smart Home, Healthcare, Agriculture, Industrial IoT, and Smart Cities, Case studies and project ideas<br><br>**Suggested Readings:** E-Book: Designing the Internet of Things by Adrian McEwen and Hakim Cassimally (Link: https://www.ebook3000.com/Designing-the-Internet-of-Things_256002.html) | **9** | **15** | **CO-4** |
| **MODULE 5: IOT WITH CLOUD COMPUTING AND BIG DATA** | | | |
| **Topic**: Integration of IoT with cloud services, IoT data management and analytics, Role of big data in IoT<br><br>**Suggested Readings:** Textbook: Cloud Computing and the Internet of Things by Amita Kapoor | **9** | **15** | **CO-5** |
| **MODULE 6: IOT SECURITY AND PRIVACY** | | | |
| **Topic:** Security challenges in IoT, Privacy concerns and data protection, IoT security frameworks and best practices<br><br>**Suggested Readings:** Reference Book: Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations by Fei Hu | **9** | **15** | **CO-6** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | • Internet of Things: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti<br>• The Internet of Things: Key Applications and Protocols by Olivier Hersent, David Boswarthick, and Omar Elloumi<br>• Cloud Computing and the Internet of Things by Amita Kapoor |
| **REFERENCE BOOKS** | |
| 1. | Raspberry Pi Cookbook by Simon Monk |
| 2. | Security and Privacy in Internet of Things (IoTs): Models, Algorithms, and Implementations by Fei Hu |
| **E BOOKS** | |
| 1. | Designing the Internet of Things by Adrian McEwen and Hakim Cassimally<br><br>Link: https://www.ebook3000.com/Designing-the-Internet-of-Things_256002.html |
| 2 | Internet of Things: Principles and Paradigms by Rajkumar Buyya and Amir Vahid Dastjerdi |

| | |
|---|---|
| | Link: https://www.pdfdrive.com/internet-of-things-principles-and-paradigms-e158579198.html |
| **MOOC** | |
| 1. | Coursera: IoT (Internet of Things) Specialization by University of California, Irvine<br>Link: https://www.coursera.org/specializations/internet-of-things |
| 2. | edX: The Internet of Things (IoT) by Curtin University<br>Link: https://www.edx.org/course/the-internet-of-things-iot |

| COURSE TITLE | | BLOCKCHAIN | | | CREDITS | 4 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA207 | COURSE CATEGORY | ME3 | L P T | 2-2-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides a comprehensive understanding of blockchain technology, its architecture, and its applications. Students will explore the principles behind blockchain, its cryptographic foundations, consensus mechanisms, and how it enables secure, transparent, and decentralized systems. |
| **Course Objective** | • To introduce the fundamental concepts of blockchain technology.<br>• To understand the architecture and workings of blockchain and its applications.<br>• To explore cryptographic principles and consensus mechanisms used in blockchain.<br>• To provide hands-on experience with blockchain development and smart contracts.<br>• To analyze and evaluate real-world blockchain use cases. |
| **Course Outcome** | • CO1: Understand the foundational principles of blockchain technology.<br>• CO2: Explain the architecture and components of blockchain.<br>• CO3: Analyze the cryptographic techniques used in blockchain.<br>• CO4: Evaluate different consensus mechanisms.<br>• CO5: Develop and deploy smart contracts on a blockchain platform.<br>• CO6: Assess the potential applications and implications of blockchain technology. |

**Prerequisites:**

- Basic understanding of computer networks and cryptography.
- Familiarity with programming concepts and data structures.

| MODULE 1: INTRODUCTION TO BLOCKCHAIN | | | |
|---|---|---|---|
| **Topic:** History of blockchain, Basic concepts, Blockchain vs. traditional databases, Types of blockchains (public, private, consortium), Key features of blockchain technology. | **9** | **15** | **CO-1** |
| **MODULE 2: BLOCKCHAIN ARCHITECTURE** | | | |
| **Topic:** Blockchain structure, Blocks and transactions, Nodes and network, Distributed ledger technology, Blockchain lifecycle, Forks, Wallets. | **12** | **20** | **CO-2** |
| **MODULE 3: CRYPTOGRAPHY IN BLOCKCHAIN** | | | |
| **Topic:** Cryptographic hash functions, Public key cryptography, Digital signatures, Merkle trees, Zero-knowledge proofs. | **9** | **15** | **CO-3** |
| **MODULE 4: CONSENSUS MECHANISMS** | | | |

| | | | |
|---|---|---|---|
| **Topic:** Proof of Work (PoW), Proof of Stake (PoS), Delegated Proof of Stake (DPoS), Practical Byzantine Fault Tolerance (PBFT), Consensus in permissioned and permissionless blockchains. | **12** | **20** | **CO-4** |
| **MODULE 5: SMART CONTRACTS AND DEVELOPMENT** | | | |
| **Topic**: Introduction to smart contracts, Ethereum and Solidity, Writing and deploying smart contracts, Tools and frameworks, Decentralized applications (DApps). | **12** | **20** | **CO-5** |
| **MODULE 6: BLOCKCHAIN APPLICATIONS AND FUTURE TRENDS** | | | |
| **Topic:** Blockchain in finance, supply chain, healthcare, government, Emerging trends in blockchain technology, Challenges and future of blockchain. | **6** | **10** | **CO-6** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | • "Blockchain Basics" by Daniel Drescher<br>• "Mastering Blockchain" by Imran Bashir<br>• "Applied Cryptography" by Bruce Schneier |
| **REFERENCE BOOKS** | |
| 1. | "Blockchain: Blueprint for a New Economy" by Melanie Swan |
| 2. | "Ethereum and Solidity: The Complete Developer's Guide" by Ritesh Modi |
| **E BOOKS** | |
| 1. | "Mastering Bitcoin" by Andreas M. Antonopoulos<br>https://github.com/bitcoinbook/bitcoinbook |
| 2 | "Bitcoin and Cryptocurrency Technologies" by Arvind Narayanan et al.<br>https://press.princeton.edu/books/hardcover/9780691171692/bitcoin-and-cryptocurrency-technologies |
| **MOOC** | |
| 1. | Coursera: "Blockchain Revolution" by INSEAD<br>Link: https://www.coursera.org/specializations/internet-of-things |
| 2. | Coursera: "Bitcoin and Cryptocurrency Technologies" by Princeton University<br>Link: https://www.coursera.org/learn/cryptocurrency |

| COURSE TITLE | | LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA207L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Introduction to Business Intelligence and Power BI**

1. **Retail Sales Analysis Dashboard**: Develop a **Power BI dashboard** for a retail chain, integrating **multiple data sources** to analyze sales trends, customer demographics, and product performance.
2. **Healthcare Patient Monitoring System**: Create a **BI solution** for hospitals using **Power BI Desktop and Service** to track patient admissions, treatment outcomes, and doctor performance.
3. **Financial Performance Reporting for Enterprises**: Implement a **self-service BI system** using **Power BI Ecosystem components** to generate interactive reports on revenue, expenses, and profit margins.
4. **E-Commerce Customer Insights Dashboard**: Design a **Power BI mobile-friendly dashboard** to visualize customer purchase behavior, abandoned carts, and personalized recommendations.
5. **Manufacturing Quality Control Monitoring**: Use **Power BI data sources** to integrate IoT sensor data from factory machines, analyze defect rates, and optimize production processes.
6. **Banking Fraud Detection System**: Develop a **BI solution** for fraud detection using **Power BI Service**, aggregating real-time transaction data and identifying suspicious patterns.

**MODULE 2: Data Acquisition and Transformation**

1. **Sales Data Consolidation for Multi-Branch Retail**: Connect **Power BI** to **Excel and SQL Server** to consolidate sales data from multiple branches, use **merging and appending** to create a unified dataset.
2. **Customer Segmentation for Marketing Campaigns**: Utilize **Power Query Editor** to clean and transform customer data, create **custom columns** for segmentation based on purchase history, and perform **data profiling** to check data quality.
3. **Hospital Patient Records Management**: Connect **Power BI** to a hospital database, filter and sort patient data, merge records from multiple sources, and ensure **data accuracy with profiling techniques**.
4. **Financial Transactions Fraud Detection**: Use **Power Query** to extract transactional data, create **calculated columns** for anomaly detection, and apply **data transformation techniques** to identify suspicious activities.
5. **E-commerce Product Performance Analysis**: Import product sales data from **multiple sources**, clean the dataset by handling missing values, create **calculated metrics** for product profitability, and generate insights for decision-making.
6. **Employee Performance Dashboard for HR**: Extract HR data from **SQL Server**, apply **data quality checks**, merge records from attendance and performance databases, and create **custom columns** for ranking employees based on key performance indicators.

**MODULE 3: Data Modeling**

1. **Sales and Inventory Data Model for Retail Business**: Design a **star schema** for a retail business, linking sales data, inventory, and product tables with **relationships** to enable dynamic reporting on product performance and stock levels.
2. **Customer Insights Dashboard with Snowflake Schema**: Implement a **snowflake schema** for customer data analysis, relating customer details, purchase history, and feedback tables, optimizing the model for **performance** and complex queries.

3. **Financial Reporting System with Optimized Data Model**: Create a **data model** for financial reports, ensuring **relationships** between accounts, transactions, and departments are set correctly and implement **data type optimizations** for faster performance.
4. **Human Resources Analytics**: Develop an HR analytics model using **best practices** to track employee performance, compensation, and department metrics, implementing **relationship management** between personnel and payroll tables.
5. **Real Estate Market Analysis**: Design a **data model** for a real estate market analysis system using a **star schema**, with tables for properties, sales transactions, and customer details, ensuring optimal performance with **data types** and indexing.
6. **IoT Data Processing for Smart Homes**: Build a **data model** for a smart home system, connecting device logs, user data, and energy consumption tables, optimizing the model for **real-time performance** and large data sets.

**MODULE 4: Data Analysis with DAX**

1. **Sales Performance Analysis with DAX**: Use **DAX functions** like **SUM** and **AVERAGE** to calculate total sales, average sales per product, and use **FILTER** to segment data by region or time period.
2. **Employee Salary and Bonus Calculation**: Implement **calculated columns** and **measures** with **DAX** to compute annual salary, performance bonuses, and apply **conditional formatting** based on performance thresholds.
3. **Customer Retention Dashboard**: Apply **time intelligence functions** like **DATESBETWEEN** to calculate customer retention rates over specific time periods and create custom metrics to track churn.
4. **Financial Year Analysis**: Use **DAX** to calculate yearly financial metrics by applying **DATEADD** to compare monthly sales growth and expenses over multiple years for trend analysis.
5. **Inventory Turnover Ratio Calculation**: Create a **measure** with **DAX** to calculate the inventory turnover ratio and apply **conditional formatting** to highlight low-turnover products.
6. **Project Performance Tracking**: Track project progress by using **DAX** to create calculated columns for project completion percentage and apply **time intelligence** to compare the current month's progress against historical performance.

**Module 5: Data Visualization and Report Creation**

1. **Sales Insights Dashboard with Interactive Visualizations**: Create a **Power BI dashboard** using **bar charts, pie charts, and tables** to visualize sales by product category, and add **interactive slicers** for filtering by region and time period.
2. **Geospatial Analysis of Customer Locations**: Use **maps** and **scatter plots** to visualize customer locations on a geographical map, allowing users to drill through for more details about customers in specific regions.
3. **Financial Overview Report with Customized Visualizations**: Design a **financial report** with **line charts and KPI indicators**, customize **tooltips and colors** to highlight key financial figures, and add **filters** for users to analyze quarterly performance.
4. **Employee Performance and Productivity Dashboard**: Implement a **dashboard** with **tables and bar graphs** to analyze employee performance, apply **conditional formatting** for ranking, and enable **drill-through** to view individual employee details.
5. **Inventory and Stock Level Monitoring**: Build an **interactive dashboard** using **stacked bar charts** and **maps** to visualize inventory levels across multiple stores, with interactive **filters** to drill into stock details by location and product type.
6. **E-Commerce Conversion Funnel Visualization**: Create a **conversion funnel report** using **Power BI charts** to track user journey stages (website visits, product views, purchases), and add **interactive slicers** to segment data by customer demographics or time period.

| COURSE TITLE | | POWER BI | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA207 | COURSE CATEGORY | ME 3.3 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 18-4-2025 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments /Project | Practical Assessment/ Test | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides a comprehensive introduction to Microsoft Power BI, a leading business intelligence and data visualization tool. Students will learn to connect to diverse data sources, transform and clean data, build interactive reports and dashboards, and share insights with stakeholders. The course emphasizes practical skills and real-world applications of Power BI in data analysis and decision-making. |
| **Course Objective** | 1. To introduce students to the fundamentals of data visualization and business intelligence.<br>2. To enable students to proficiently use Power BI for data analysis, reporting, and dashboard creation.<br>3. To provide hands-on experience in data transformation, modeling, and advanced DAX calculations.<br>4. To prepare students to apply Power BI skills in real-world business scenarios.<br>5. To equip students with the knowledge required for Microsoft Power BI certification exams.. |
| **Course Outcome** | CO1: Understand the fundamental concepts of Business Intelligence and the role of Power BI.<br>CO2: Acquire data from various sources and perform data cleaning and transformation using Power Query.<br>CO3: Design and implement data models in Power BI, including creating relationships and optimizing data for performance.<br>CO4: Utilize DAX language to create calculated columns, measures, and perform complex data analysis.<br>CO5: Create interactive reports and dashboards using a variety of visualizations and interactive elements.<br>CO6: Publish and share reports, collaborate with others, and understand the security and administration aspects of Power BI. |
| **Prerequisites:** MS Excel, SQL and DBMS | |

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Business Intelligence and Power BI** | | | |
| 1.1 Business Intelligence (BI) Concepts, 1.2 Evolution of BI and Self-Service BI, 1.3 Introduction to Power BI Desktop, Service, and Mobile, 1.4 Power BI Ecosystem and Components, 1.5 Data Sources in Power BI, 1.6 Benefits and Applications of Power BI | **9** | **20%** | **CO-1** |
| **MODULE 2: Data Acquisition and Transformation** | | | |
| 2.1 Connecting to Data Sources (Excel, SQL Server, Databases, etc.), 2.2 Power Query Editor Interface and Basic Transformations, 2.3 Data Cleaning and Transformation Techniques (Filtering, Sorting, Merging, Appending), 2.4 Creating Custom Columns and Calculated Columns, 2.5 Data Profiling and Quality Checks | **10** | **20%** | **CO2** |
| **MODULE 3: Data Modeling** | | | |

| | | | |
|---|---|---|---|
| 3.1 Data Modeling Concepts and Best Practices, 3.2 Creating and Managing Relationships between Tables, 3.3 Star and Snowflake Schema Design, 3.4 Data Types and their Impact on Performance, 3.5 Data Model Optimization Techniques | **10** | **20%** | **CO-3** |
| **MODULE 4: Data Analysis with DAX** | | | |
| 4.1 Introduction to DAX Language and Syntax, 4.2 Creating Calculated Columns and Measures, 4.3 Common DAX Functions (SUM, AVERAGE, FILTER, etc.), 4.4 Time Intelligence Functions (DATEADD, DATESBETWEEN, etc.), 4.5 Conditional Formatting and Calculations | **10** | **20%** | **CO-4** |
| **Module 5: Data Visualization and Report Creation** | | | |
| 5.1 Selecting Appropriate Visualizations, 5.2 Creating Charts, Graphs, Tables, Maps, and other Visualizations, 5.3 Customizing Visualizations (Formatting, Colors, Tooltips), 5.4 Adding Interactive Elements (Filters, Slicers, Drill-through), 5.5 Designing Effective Report Layouts | **4** | **10%** | **CO-5** |
| **Module 6: Hands on Application Development** | | | |
| Design Principles for Effective Dashboards- Creating Interactive Reports and Dashboards in Power BI- Customizing Visualizations and Using Custom Visuals- Publishing and Sharing Reports in Power BI Service | **2** | **10%** | **ALL CO** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | Larson, M., & Green, B. (2017). *Analyzing Data with Power BI and Power Pivot for Excel*. Retrieved from O'Reilly |
| 2. | Alexander, C., & White, T. (2018). *Microsoft Power BI Quick Start Guide*. Birmingham, UK: Packt Publishing. |
| 3. | Ferrari, M., & Russo, A. (2017). *The Definitive Guide to DAX: Business Intelligence with Microsoft Excel, SQL Server Analysis Services, and Power BI*. Redmond, WA: Microsoft Press. |
| **E BOOKS** | |
| 1. | Sarkar, R. (2018). *Mastering Microsoft Power BI*. Retrieved from Packt |
| 2. | Larson, M., & Green, B. (2017). *Analyzing Data with Power BI and Power Pivot for Excel*. Retrieved from O'Reilly |
| **MOOC** | |
| **1.** | Coursera: Data Analysis and Visualization with Power BI |
| **2.** | LinkedIn Learning: Power BI Essential Training |

| COURSE TITLE | | MOBILE APPLICATION DEVELOPEMNT | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA 301 | COURSE CATEGORY | CM | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test/ Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | Mobile application development is one of the rising and growing trend in the industry of mobile. This course examines the principles of mobile application design and covers the necessary concepts which are required to understand mobile based applications and develop Mobile based Applications in particular. After completing this course, students will design and build a variety of real-time Apps using Mobile application development. |
| **Course Objective** | The aim of this course is to help the student to attain the following industry identified competency through various teaching learning experiences: |
| **Course Outcome** | The theory, practical experiences and relevant soft skills associated with this course are to be taught and implemented, so that the student demonstrates the following ***industry oriented*** COs associated with the above mentioned competency:<br><br>**CO1 Interprete** features of Mobile application development operating system.<br>**CO2 Configure** Mobile application environment and development tools.<br>**CO3 Develop** rich user Interfaces by using layouts and controls.<br>**CO4 Use** User Interface components for mobile application development.<br>**CO5 Create** Mobile application using database.<br>**CO6: Demonstrate** Hybrid Mobile App Framework. |

| MODULE | No of lectures | Weightage | CO Mapped |
|---|---|---|---|
| **MODULE 1: Mobile application and its tools** | | | |
| 1.1 Introduction to Mobile application, Mobile application Ecosystem. 1.2 Need of Mobile application, Features of Mobile application,<br>1.3 Tools and software required for developing an Mobile application, 1.4 Mobile application Architecture | 5 | 12% | CO1 |
| **MODULE 2: Installation and configuration of Mobile application** | | | |

| | | | |
|---|---|---|---|
| 2.1 Operating System, Java JDK, Mobile application SDK, 2.2 Mobile application Development Tools (ADT), Mobile application Virtual Devices(AVDs), Emulators, 2.3 Dalvik Virtual Machine, Difference between JVM and DVM 2.4 Steps to install and configure Mobile application Studio and SDK | 8 | 18% | CO1,CO2 |
| **MODULE 3: UI Components and Layouts** | | | |
| 3.1 Control Flow, Directory Structure,<br>3.2 components of a screen, 3.3 Fundamental UI Design Linear Layout; Absolute Layout; Frame Layout; Table Layout; Relative Layout | 6 | 15% | CO3 |
| **MODULE 4: Designing User Interface With View** | | | |
| 4.1 Text View, Edit Text; Button, Image Button; Toggle Button; Radio Button And Radio Group; Checkbox; Progress Bar , 4.2 List View; Grid View; Image View; Scroll View; 4.3 Custom Toast Alert<br>4.4 Time And Date Picker | 6 | 15% | CO4 |
| **MODULE 5: Activity, Multimedia and Databases** | | | |
| 5.1 Intent, Intent Filter , 5.2 Activity Lifecycle; 5.3 Broadcast Lifecycle ,5.4 Content Provider; Fragments Service: Features Of service, Mobile application platform service, 5.5 Defining new service, Service Lifecycle, Permission, example of service<br>5.6 Multimedia framework, Play Audio and Video, Text to speech, 5.7 Audio Capture, Camera Bluetooth, Animation , 5.8 SQLite Database, necessity of SQLite, 5.9 Creation and connection of the database | 14 | 25% | CO4, CO5 |
| **MODULE 6 : Introduction to Flutter** | | | |
| 6.1. Overview of Flutter 6.2. Installation of Flutter 6.3. Architecture of Flutter 6.4. Introduction to Dart Programming 6.5. Demonstration of Simple application | 6 | 15% | CO6 |
| **TEXT BOOKS** | | | |
| 1 | Mobile application Dixit, Prasanna Kumar Vikas Publications, New Delhi 2014, ISBN: | | |
| 2 | 9789325977884<br>Pro Mobile application 5 Maclean David, Komatineni Satya, Allen Grant press Publications, 2015, ISBN: 978-1-4302-4680-0 | | |
| 3 | Mobile application Programming for Beginners Hortan, John Packet Publication, 2015, ISBN: 978-1-78588-326-2 | | |
| **REFERENCE BOOKS** | | | |
| 1 | **"Android Programming: The Big Nerd Ranch Guide"** by Bill Phillips and Chris Stewart | | |
| 2 | **"iOS Programming: The Big Nerd Ranch Guide"** by Christian Keur and Aaron Hillegass | | |
| 3 | **Mobile App Development with Ionic: Cross-Platform Apps with Ionic, Angular, and Cordova"** by Chris Griffith | | |
| **E BOOKS** | | | |

| | |
|---|---|
| 1 | Mobile application Programming: The Big Nerd Ranch Guide" by Bill Phillips, Chris Stewart, and Kristin Marsicano |
| 2 | "Learning iOS Development: A Hands-On Guide to Building iPhone and iPad Apps" by Alok Mehta |
| **MOOC** | |
| 1 | https://www.coursera.org/specializations/mobile application-app-development |
| 2 | https://www.udemy.com/courses/development/mobile-apps/?srsltid=AfmBOoqUogz5JGf2gfC2GA6En6SsR91bpviwqsaAhd6PVs6qoIbbwScx |
| 3 | https://www.edx.org/learn/mobile-development |

| COURSE TITLE | | MOBILE APPLICATION DEVELOPMENT LAB | | | CREDITS | 2 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA301L | COURSE CATEGORY | SEC | L-P-T | 0-4-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Mobile application and its tools**

1. Create "Hello World" application to display "Hello World" in the middle of the screen in red colour with white background.

**MODULE 2: Installation and configuration of Mobile application**

1.Install /configure java development kit (JDK), android studio and android SDK.

2. Configure android development tools (ADT) plug-in and create android virtual device.

**MODULE 3: UI Components and Layouts**

1.Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.

2.Implement Linear and Absolute Layout
Scenario: You're designing an app where the login form needs to display username and password fields stacked vertically. You use LinearLayout for this simple vertical arrangement. Additionally, a separate section of the app requires precise control over the placement of some icons, so you use AbsoluteLayout for that section.

3.Implement Frame, Table, and Relative Layouts
Scenario: For a weather app, you need to create a layout with a header at the top, a content section with multiple elements arranged in a table (e.g., hourly forecast), and a floating button. You use FrameLayout to stack elements like the floating button over the content, TableLayout to arrange forecast data in rows, and RelativeLayout to position the header relative to the content.

4.Create a screen that has input boxes for User Name, Password, Address, Gender(radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout)

**MODULE 4: Designing User Interface With View**

1.Create sample application with login module. (Check username and password), validate it for login screen or alert the user with a Toast.

2. Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a

horizontal line between the job title and the phone number.

3. Develop an Android application using controls like Button, TextView, EditText for designing a calculator

4. Write an android code to make phone call using Intent

5.Write an android code to turn ON/OFF Bluetooth

6.Write an android code to turn ON /OFF the Wi-Fi

**MODULE 5: Activity And Multimedia with databases**

1.Create a SIGN Upton activity with Username and Password. Validation of password should happen

2. Create a SIGN Up activity with Username and Password. Validation of password should happen based on the following rules: - Password should contain uppercase and lowercase letters. - Password should contain letters and numbers. - Password should contain special characters. - Minimum length of the password (the default value is 8). On successful SIGN UP proceed to the next Login activity. Here the user should SIGN IN using the Username and Password created during signup activity. If the Username and Password are matched then navigate to the next activity which displays a message saying "Successful Login" or else display a toast message saying "Login Failed". The user is given only two attempts and after that display a toast message saying "Failed Login Attempts" and disable the SIGN IN button. Use Bundle to transfer information from one activity to another.

3. Create an application that will create database to store username and password.

4. Create an application to change screen colour as per the user choice from a menu.

5. Develop a simple application with one Edit Text so that the user can write some text in it.
Create a button called "Convert Text to Speech" that converts the user input text into voice.

6.Write an android application using SQLite to create table and perform CRUD operations (Example. COURSE table (ID, Name, Duration, Description), perform ADD, UPDATE, DELETE and READ operations)

**MODULE 6: Security and Application Deployment**

**1.Deploy a Map-Based Application**
**Scenario:** You're developing a food delivery app. You use **Google Maps** to allow users to track the delivery in real-time, view nearby restaurants, and get directions to the restaurant or the delivery destination. Deploy & publish this app on google play store

2. Develop a simple app for **bus ticket reservation** that allows users to select a bus route, choose the number of tickets, and complete the booking process. Deploy & publish this app on google play store

Note:- Additional lab experiments will be also given time to time according to topic as per the syllabus

| COURSE TITLE | Software Project Management & DevOps | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA302 | COURSE CATEGORY | CM | L- P- T | 3- 0-0 |
| Version 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | | |
|---|---|---|---|---|
| First Periodical Assessment /Seminar | Second Periodical Assessment | Assignments | Attendance | ESE |
| 10% | 10% | 20% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides students with a deep understanding of Software Project Management (SPM) principles and the DevOps methodology. It covers project planning, execution, risk management, agile development, CI/CD pipelines, containerization, and automation. Students will gain hands-on experience in modern tools such as Git, Docker, Kubernetes, and Jenkins. The course prepares students to manage and deliver high-quality software efficiently. |
| **Course Objective** | 1. To understand the fundamentals of Software Project Management, including planning, execution, monitoring, and risk assessment.<br>2. To explore Agile, Scrum, and DevOps methodologies for software development and continuous integration.<br>3. To learn modern DevOps tools such as Git, Docker, Kubernetes, and Jenkins for automation and deployment.<br>4. To analyze and manage software project risks, resource allocation, and cost estimation techniques.<br>5. To develop skills in collaboration, automation, and monitoring in DevOps environments.. |
| **Course Outcome** | CO1 Apply Software Project Management principles to real-world projects, including planning, scheduling, and risk management.<br>CO2 Utilize Agile, Scrum, and DevOps methodologies to improve software development efficiency and delivery.<br>CO3 Implement CI/CD pipelines using DevOps tools like Git, Docker, and Jenkins for automation and deployment.<br>CO4 Evaluate and manage risks, quality assurance, and cost estimation in software projects.<br>CO5 Integrate monitoring, security, and best DevOps practices in software projects for continuous improvement. |

**Prerequisites:** Basic understanding of Software Development Life Cycle

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1:** Introduction to Software Project Management | | | |
| Definition, Scope, and Importance of SPM, Project Lifecycle & Phases, Role of a Project Manager, Stakeholder Management & Communication- Stakeholder Management & Communication | 7 | 15% | CO1 |

| | | | |
|---|---|---|---|
| **Module 2:** Project Planning and Risk Management | | | |
| Project Initiation & Feasibility Study, Work Breakdown Structure (WBS), Estimation Techniques (COCOMO, Function Point Analysis), Risk Identification, Assessment, and Mitigation, Quality Management and Project Monitoring (KPIs) | **7** | **15%** | **CO2** |
| **MODULE 3: Agile & Scrum Methodology** | | | |
| Introduction to Agile Development, Scrum Framework: Roles, Artifacts, Events, Sprint Planning, Execution & Retrospective, Kanban, Extreme Programming (XP), SAFe, Agile Metrics and Performance Tracking | **7** | **15%** | **CO3** |
| **MODULE 4: DevOps Principles and Continuous Integration/Delivery (CI/CD)** | | | |
| Introduction to DevOps and its Culture, CI/CD Concepts and Best Practices, Version Control with Git & GitHub, Jenkins, GitLab CI/CD, Travis CI, Automated Testing and Deployment Strategies | **9** | **20%** | **CO2, CO3** |
| **MODULE 5: Containerization, Orchestration, and Cloud Deployment** | | | |
| Introduction to Docker & Containerization, Building & Managing Containers, Kubernetes: Architecture & Components, Cloud DevOps: AWS, Azure, GCP CI/CD, Infrastructure as Code (IaC) with Terraform & Ansible | **9** | **20%** | **CO3, CO5** |
| **MODULE 6: Monitoring, Security, and DevOps Best Practices** | | | |
| Application & Infrastructure Monitoring (ELK, Prometheus, Grafana), Security in DevOps (DevSecOps), Threat Modeling & Vulnerability Assessment, DevOps Case Studies & Industry Trends, Future Trends in SPM & DevOps | **6** | **15%** | **CO4, CO5** |
| **TEXTBOOKS** | | | |
| 1. | **Software Project Management** – Bob Hughes, Mike Cotterell, Rajib Mal | | |
| 2 | **The DevOps Handbook** – Gene Kim, Patrick Debois, John Willis, Jez Humble | | |
| **REFERENCE BOOKS** | | | |
| 1. | **Accelerate: The Science of Lean Software and DevOps** – https://www.amazon.com/Accelerate-Science-Lean-Software-DevOps/dp/1942788339 | | |
| 2. | **Google SRE Handbook** – https://sre.google/books/ | | |
| **E BOOKS** | | | |
| 1. | **Accelerate: The Science of Lean Software and DevOps** – https://www.amazon.com/Accelerate-Science-Lean-Software-DevOps/dp/1942788339 | | |
| 2 | **Google SRE Handbook** – https://sre.google/books/ | | |
| 3 | **Infrastructure as Code** – https://www.oreilly.com/library/view/infrastructure-as-code/9781098114671/ | | |
| 4 | **Free DevOps Course by Microsoft** – https://learn.microsoft.com/en-us/devops/ **Scrum Guide** – https://scrumguides.org/ | | |
| **MOOC** | | | |
| 1. | **Coursera: DevOps, Cloud, and Agile Specialization** – https://www.coursera.org/specializations/devops-cloud-agile | | |
| 2. | **edX: DevOps Professional Certificate** – https://www.edx.org/professional-certificate/linuxfoundationx-introduction-to-devops | | |

| | |
|---|---|
| 3. | **Udemy: DevOps CI/CD with Jenkins, Kubernetes, Docker & AWS** – https://www.udemy.com/course/devops-ci-cd-with-jenkins-kubernetes-docker-aws-git/ |

| COURSE TITLE | | CLOUD MIGRATION AND MANAGEMENT LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| | | | | | | |
| COURSE CODE | | MCA303L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

| MODULE: Topic |
|---|
| **MODULE 1: Introduction to Cloud Migration**<br>**Task 1.**Identify Business and Technical Drivers for Cloud Migration<br><br>**Task 2.** Simulate a Rehost ("Lift and Shift") Migration using AWS EC2<br><br>**Task 3**. Compare Migration Strategies with a Sample App |
| **MODULE 2: Cloud Migration Planning and Assessment**<br><br>**Task 1.** Choosing the Right Cloud Service Model (IaaS, PaaS, SaaS)<br><br>**Task 2**. Selecting Cloud Providers: AWS, Azure, Google Cloud |
| **MODULE 3: Cloud Migration Methodologies & Tools**<br>**Task 1**. Lift and Shift (Rehosting) Migration to the Cloud<br><br>**Task 2**. Replatforming an Application to the Cloud |
| **MODULE 4: Security, Compliance, and Risk Management**<br><br>**Task 1**. Deploy a sample application (or VM) to a cloud environment.<br><br>**Task 2.** Use secure communication (SSL/TLS).<br><br>**Task 3**. Ensure data encryption at rest and in transit. |
| **MODULE 5: Cloud Management and Optimization**<br>**Task 1.** Cloud Cost Optimization and Auto-Scaling Setup<br><br>**Task 2**. Performance Monitoring & Cloud DevOps Integration for Continuous Monitoring |
| **MODULE 6 : Automation and Future Trends in Cloud Migration**<br>**Task 1.** Using Infrastructure as Code (IaC) for Cloud Migration with Terraform<br><br>**Task 2**. AI and Machine Learning for Cloud Optimization |

| COURSE TITLE | | Cloud Migration and Management | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA303 | COURSE CATEGORY | ME | L P T | 3-0-0 |
| Version | 1.0 | Approval Details | | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides insights into cloud migration strategies, tools, and security best practices. Students will learn to analyze risks, optimize cloud resources, and implement automation using DevOps and Infrastructure as Code. By the end, they will be able to develop efficient cloud-based solutions for modern IT environments. |
| **Course Objective** | 1.Understand the fundamental concepts, strategies, and challenges of cloud migration.<br>2. Identify various cloud migration approaches, tools, and best practices.<br>3.Analyze cloud security, compliance, and risk management in migration processes.<br>4. Apply cloud management techniques for resource optimization and cost efficiency.<br>5. Develop solutions for monitoring, automation, and performance tuning in cloud environments. |
| **Course Outcome** | CO1: **Understand** the principles, benefits, and risks of cloud migration strategies.<br>CO2: **Identify** suitable cloud migration methodologies and select appropriate tools.<br>CO3: **Analyze** security, compliance, and governance challenges in cloud migration.<br>CO4: **Apply** cloud resource management techniques to optimize performance and cost.<br>CO5: **Develop** automated solutions for cloud monitoring and performance optimization. |

**Prerequisites: Basic knowledge of Cloud Computing Concepts**

| Module Name | No of lectures | Weightage | CO-PO mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Cloud Migration** | | | |
| 1.1 Fundamentals of Cloud Migration<br>1.2 Business and Technical Drivers for Cloud Adoption<br>1.3 Migration Frameworks and Strategies (Rehost, Refactor, Rearchitect, Rebuild) | 5 | 10% | **CO-1** |
| **MODULE 2: Cloud Migration Planning and Assessment** | | | |
| 2.1 Pre-Migration Considerations (Workload Analysis, Cost Estimation)<br>2.2 Cloud Readiness Assessment<br>2.3 Choosing the Right Cloud Service Model (IaaS, PaaS, SaaS)<br>2.4 Selecting Cloud Providers: AWS, Azure, Google Cloud | 6 | 15% | **CO-1 & CO-2** |
| **MODULE 3: Cloud Migration Methodologies & Tools** | | | |

| | | | |
|---|---|---|---|
| 3.1 Lift and Shift (Rehosting)<br>3.2 Replatforming and Refactoring Approaches<br>3.3 Cloud-Native Migration Strategies<br>3.4 Migration Tools (AWS Migration Hub, Azure Migrate, Google Migrate for Compute Engine)<br>3.5 Downtime Minimization and Data Consistency<br><br>Hands on Practical based on this unit. | 9 | 20% | **CO-2** |
| **MODULE 4: Security, Compliance, and Risk Management** | | | |
| 4.1 Cloud Security Best Practices in Migration<br>4.2 Compliance and Legal Aspects (GDPR, HIPAA, ISO 27001)<br>4.3 Identity and Access Management (IAM) in Cloud<br>4.4 Disaster Recovery and Backup Strategies<br><br>Hands on Practical based on this unit. | 9 | 20% | **CO-3** |
| **MODULE 5: Cloud Management and Optimization** | | | |
| 5.1 Cloud Cost Optimization Strategies (Right-Sizing, Reserved Instances)<br>5.2 Performance Monitoring & Analytics (CloudWatch, Azure Monitor, Stackdriver)<br>5.3 Auto-Scaling and Load Balancing<br>5.4 Cloud Governance & Policy Management<br><br>Hands on Practical based on this unit. | 8 | 20% | **CO-4** |
| **MODULE 6 : Automation and Future Trends in Cloud Migration** | | | |
| 6.1 Infrastructure as Code (IaC) for Cloud Migration (Terraform, AWS CloudFormation)<br>6.2 AI and Machine Learning in Cloud Optimization<br>6.3 Multi-Cloud and Hybrid Cloud Strategies<br>6.4 Edge Computing and Serverless Technologies<br><br>Hands on Practical based on this unit. | 8 | 15% | **CO-5** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | 1. **"Cloud Computing: Concepts, Technology & Architecture"** – Thomas Erl, Ricardo Puttini, Zaigham Mahmood<br>2. **"Architecting the Cloud: Design Decisions for Cloud Computing Service Models (SaaS, PaaS, and IaaS)"** – Michael J. Kavis |
| **REFERENCE BOOKS** | |
| 1. | 1. **"Cloud Migration Handbook: A Step-by-Step Guide to Efficiently Move Your IT Infrastructure to the Cloud"** – José Antonio Hernández<br>2. **"Cloud Computing Patterns: Fundamentals to Design, Build, and Manage Cloud Applications"** – Christoph Fehling, Frank Leymann, Ralph Mietzner |
| **E BOOKS** | |
| 1. | **"AWS Migration Whitepaper"** – Amazon Web Services |
| 2. | **"Microsoft Cloud Adoption Framework for Azure"** |
| **MOOC** | |
| 1. | **"Google Cloud's Architecting with Google Kubernetes Engine"** – Coursera (Offered by Google Cloud) |
| 2. | **"AWS Cloud Migration Essentials"** – Udemy |

| COURSE TITLE | | MERN STACK DEVELOPMENT LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA303L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Introduction to MERN Stack**

1. **MERN Stack Folder Setup for Blogging App:** Create the base folder structure for a full-stack blogging platform with separate folders for client and server. Initialize both using `npm` and `create-react-app`.
2. **Architecture Blueprint Documentation:** Draw and document the MERN Stack architecture for an e-commerce project showing the flow of data and components (frontend/backend/database).
3. **Simple React Homepage with Bootstrap Styling:** Create a React homepage using JSX and Bootstrap. Add a header, navigation bar, and footer.
4. **Express Server Basic Setup:** Set up a basic Express.js server with a root route that returns "MERN Stack API Working" on the browser.
5. **MongoDB Atlas Setup and Connection Test:** Create a MongoDB cluster on Atlas and connect it to your local Express app using Mongoose.
6. **Terminal-based CRUD Planning:** Create a command-line interactive menu using Node.js that simulates Create, Read, Update, and Delete operations (mock database with arrays).

**MODULE 2: MongoDB & Mongoose**

1. **Mongoose Schema for Inventory App:** Design and implement a product schema with fields like name, quantity, price, and category.
2. **CRUD API for Library System:** Develop Express.js routes using Mongoose to handle CRUD operations for books (title, author, genre).
3. **Student Record Filter using Aggregation:** Use Mongoose aggregation to get the number of students enrolled per course.
4. **MongoDB Query for Filtering and Sorting Products:** Write and test MongoDB queries to filter products by price range and sort them by name.
5. **Reference vs Embedded Document Example:** Implement both referenced and embedded schema relationships for a blog and its comments.
6. **Data Validation with Mongoose:** Add validation rules in Mongoose schema for user registration (e.g., email format, password length).

**MODULE 3: Node.js & Express.js**

1. **REST API for Task Manager:** Create a Node.js and Express API to manage tasks (create, update, delete). Use HTTP methods appropriately.
2. **Custom Middleware Logging:** Write middleware that logs request type, URL, and timestamp for every incoming request.
3. **User Authentication System with JWT:** Build a user login system with JSON Web Token authentication and protected routes.
4. **Express Router for Modular Routes:** Separate user, task, and admin routes using Express Router and link them in your main server file.
5. **Error Handling Middleware:** Implement centralized error handling in Express and test with invalid routes and exceptions.

6. **Environment Variable Configuration:** Securely store sensitive data (e.g., DB URI, JWT secret) in `.env` and access it in your server.

**MODULE 4: React.js & Frontend Development**

1. **User Profile Component with Props:** Create a reusable React component to display a user profile using props (name, email, avatar).
2. **Login Form with Validation:** Build a login form using useState and show error messages dynamically if fields are empty.
3. **Navigation with React Router:** Create a multi-page application with React Router including Home, About, and Contact pages.
4. **Redux-Based Counter App:** Implement a counter using Redux with actions for increment, decrement, and reset.
5. **Todo List with State and Props:** Create a Todo app where new tasks can be added, marked completed, or deleted. Use state and props.
6. **Search Filter with useState and useEffect:** Build a product list with a search bar that filters items as the user types.

**Module 5: Integration & API Communication**

1. **Blog Fetch and Display:** Fetch blog data from an Express API using Axios and display it in a React frontend component.
2. **Login Authentication with Token:** Implement login from React, send credentials to backend, store JWT token, and use it for accessing protected routes.
3. **Protected Route for Dashboard:** Create a protected route in React that only allows access if a valid token is present.
4. **Global Error Handling Component:** Show user-friendly error messages when backend APIs fail using a reusable error component.
5. **Loading Spinner for Async Calls:** Display a loading spinner during API requests and hide it once data is loaded.
6. **Memoization in React for Performance:** Use `React.memo` and `useMemo` in a list rendering component to optimize performance.

**Module 6: Deployment & Cloud Hosting**

1. **Deploy React Frontend on Vercel:** Deploy a React application on Vercel and configure a custom domain.
2. **Deploy Node Backend on Render:** Deploy an Express backend with MongoDB on Render and connect it to your frontend.
3. **Environment Variables on Production:** Use `.env` for production secrets like MongoDB URI, JWT_SECRET, and access them securely.
4. **Auto Deployment with GitHub Actions:** Set up a GitHub Actions workflow to auto-deploy the backend when code is pushed to the main branch.
5. **End-to-End MERN App Deployment:** Deploy a complete MERN app (frontend + backend + DB) with proper routing and API connection.
6. **Audit and Optimize Production Build:** Run a performance audit (Lighthouse) on your deployed frontend, analyze, and optimize.

| COURSE TITLE | | MERN Stack Development | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA 303 | COURSE CATEGORY | ME 4.2 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 18-4-2025 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces students to the **MERN (MongoDB, Express.js, React.js, and Node.js) stack**, enabling them to develop **full-stack web applications**. It covers **database management, backend development, frontend development, and API integration**, equipping students with practical knowledge for modern web application development. |
| **Course Objective** | • Understand **MERN Stack architecture** and its components.<br>• Develop and manage **MongoDB databases** for storing application data.<br>• Implement **RESTful APIs** using **Node.js and Express.js**.<br>• Design and develop **responsive frontend applications** using **React.js**.<br>• Integrate **frontend and backend** using APIs.<br>• Deploy MERN stack applications on cloud platforms. |
| **Course Outcome** | CO1 Understand the architecture and components of the MERN stack.<br>CO2 Implement database operations using MongoDB and Mongoose.<br>CO3 Develop RESTful APIs with Express.js and Node.js.<br>CO4 Build interactive and responsive user interfaces using React.js.<br>CO5 Integrate frontend and backend using API calls.<br>CO6 Deploy MERN stack applications on cloud platforms. |

**Prerequisites:** Basic knowledge of HTML, CSS, and JavaScript; understanding of client-server architecture; familiarity with basic programming concepts.

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to MERN Stack** | | | |
| 1.1 Overview of Web Development Technologies, 1.2 Introduction to Full-Stack Development, 1.3 Understanding MERN Stack and Its Architecture, 1.4 Setup and Installation of MERN Stack | 6 | 10% | **CO-1** |
| **MODULE 2: MongoDB & Mongoose** | | | |
| 2.1 Introduction to MongoDB, 2.2 Installation and Setup of MongoDB, 2.3 MongoDB Basics: CRUD Operations, 2.4 Introduction to Mongoose, 2.5 Defining a Mongoose Schema and Model, 2.6 Mongoose CRUD Operations, 2.7 Schema Validation and Middleware | 10 | 20% | **CO-2** |
| **MODULE 3: Node.js & Express.js** | | | |
| 3.1 Introduction to Node.js, 3.2 Setting up an Express.js Server, 3.3 Middleware and Routing, 3.4 RESTful APIs and HTTP Methods, 3.5 Authentication and Authorization (JWT) | 8 | 20% | **CO-3** |
| **MODULE 4: React.js & Frontend Development** | | | |

| | | | |
|---|---|---|---|
| 4.1 Introduction to React.js and JSX, 4.2 React Components, Props, and State, 4.3 Handling Events and Forms, 4.4 React Router and Navigation, 4.5 State Management using Redux | 8 | 20% | **CO-4** |
| **MODULE 5: Integration & API Communication** | | | |
| 5.1 Connecting Frontend with Backend using Fetch/Axios, 5.2 Handling Asynchronous Requests, 5.3 Authentication and Protected Routes, 5.4 Error Handling and Debugging, 5.5 Performance Optimization | 8 | 20% | **CO-5** |
| **MODULE 6: Deployment & Cloud Hosting** | | | |
| 6.1 Introduction to Deployment Strategies, 6.2 Hosting MERN Apps on **Vercel, Netlify, Heroku**, 6.3 Managing Environment Variables, 6.4 Continuous Deployment (CD) using GitHub Actions, 6.5 Best Practices for Production Deployment | 5 | 10% | **CO-5** |

| **TEXT BOOKS** | |
|---|---|
| 1. | **Shama Hoque**, *Full-Stack React, TypeScript, and Node*, Packt Publishing. |
| 2. | **Ethan Brown**, *Web Development with Node and Express*, O'Reilly Media. |
| 3. | **Adam Boduch**, *React and React Native*, Packt Publishing. |
| **REFERENCE BOOKS** | |
| 1. | **Brad Traversy**, *MERN Stack Front To Back*, Independently Published. |
| 2. | **Kyle Simpson**, *You Don't Know JS* Series, O'Reilly Media. |
| 3. | **Robin Wieruch**, *The Road to React*, Leanpub. |
| **E BOOKS** | |
| 1. | MongoDB Basics - https://www.mongodb.com/lp/book/mongodb-basics |
| 2 | Node.js Handbook - https://thevalleyofcode.com/nodejs/ |
| **MOOC** | |
| 1. | **MongoDB University** – Free Course on MongoDB Basics https://university.mongodb.com/courses |
| 2. | **Full-Stack MERN Course** – FreeCodeCamp https://www.freecodecamp.org/news/mern-stack-full-tutorial/ |

| COURSE TITLE | Deep Learning | | | CREDITS | 1 |
|---|---|---|---|---|---|
| COURSE CODE | MCA303L | COURSE CATEGORY | ME4.3 | L-P-T | 0-2-0 |
| Version 1.0 | Approval Details | | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces students to the fundamental concepts of Artificial Intelligence (AI) and Deep Learning (DL). Fundamental topics include intelligent agents, search algorithms, machine learning techniques, neural networks, and deep learning models for practical applications.<br><br>**Prerequisites**: Basic understanding of Machine Learning and Python programming. |
| **Course Objective** | • To introduce core concepts and architecture of deep learning systems.<br>• To explore key algorithms such as CNNs, RNNs, GANs, and Transformers.<br>• To implement DL models using frameworks like TensorFlow and PyTorch.<br>• To evaluate and optimize models using real-world datasets.<br>• To discuss deployment strategies and ethical concerns in deep learning. |
| **Course Outcome** | By the end of the course, students will be able to:<br>CO1: Understand core deep learning concepts, architectures, and workflows.<br>CO2: Apply machine learning principles to build neural network models.<br>CO3: Design and implement deep learning models for real-world tasks.<br>CO4: Deploy and evaluate DL systems using tools such as TensorFlow & PyTorch.<br>CO5: Examine ethical and social implications of DL technologies. |

**MODULE**

**MODULE 1: Introduction To Machine Learning**

1. AI-Powered Chatbot for Customer Support
2. Predictive Analytics for Sales Forecasting
3. AI-Based Image Recognition for Object Detection

**MODULE 2 : Search and Optimization Techniques**

1. Develop a program to find a path between two points in a maze using **Breadth-First Search (BFS)** and **Depth-First Search (DFS)**.
2. Implement **A*** and **Greedy Best-First Search** to solve the classic **8-puzzle problem** (a sliding puzzle with 3x3 grid).
3. Solve a Sudoku puzzle using **Constraint Satisfaction Problem (CSP) techniques** and **Genetic Algorithms**.

**MODULE 3: Machine Learning for AI**

1. Use **Supervised Learning** (Regression) to predict house prices based on features like area, number of rooms, and location.
2. Use **Unsupervised Learning** to segment customers based on purchasing behavior.
3. Classify emails as **spam or not spam** using Decision Trees, Naïve Bayes, and SVM.

**MODULE 4: Fundamentals of Neural Networks**

| |
|---|
| 1. Develop a perceptron model to classify points in a 2D space (e.g., classifying red and blue dots).<br>2. Building a Multi-Layer Perceptron (MLP) for Handwritten Digit Recognition.<br>3. Compare different **activation functions** (ReLU, Sigmoid, Tanh) and observe their effect on learning. |
| **MODULE 5: Deep Learning Architectures** |
| 1. Train a CNN to classify images from a dataset like **CIFAR-10** or **Fashion-MNIST.**<br>2. Train an **LSTM-based** model to generate text, such as poetry or news headlines.<br>3. Use a **GAN** to generate realistic images from random noise. |
| **MODULE 6: Applications and Deployment of Deep Learning** |
| 1. Train a deep learning model to predict diseases (e.g., diabetes or heart disease) based on patient data.<br>2. Deploy a trained **finance fraud detection model** as a web application.<br>3. Investigate AI bias in a model trained on gender-based salary prediction. |

| COURSE TITLE | | Deep Learning | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA303 | COURSE CATEGORY | ME4.3 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces students to the fundamental concepts of Artificial Intelligence (AI) and Deep Learning (DL). Fundamental topics include intelligent agents, search algorithms, machine learning techniques, neural networks, and deep learning models for practical applications. |
| **Course Objective** | • To understand the fundamental concepts of Artificial Intelligence and Deep Learning.<br>• To learn problem-solving strategies using search and optimization techniques.<br>• To explore machine learning and deep learning approaches for AI-driven solutions.<br>• To implement deep learning architectures for real-world applications.<br>• To evaluate AI models using appropriate metrics. |
| **Course Outcome** | By the end of the course, students will be able to:<br>CO1: Understand AI principles, problem-solving methods, and search techniques.<br>CO2: Apply machine learning concepts for intelligent decision-making.<br>CO3: Develop deep learning models using neural networks.<br>CO4: Implement AI and DL techniques for real-world applications.<br>CO5: Evaluate and optimize AI models for improved performance. |

| MODULE | No. of lectures | Weightage (%) | CO Mapped |
|---|---|---|---|
| **MODULE 1: Introduction to Artificial Intelligence** | | | |
| 1.1 Definition and History of AI<br>1.2 Applications of AI<br>1.3 Intelligent Agents and Environments<br>1.4 Problem-Solving using AI<br>1.5 AI Ethics and Social Implications<br><br>Hands on practice on topics learned in this module as listed below: | 5 | 10% | **CO1** |
| **MODULE 2 : Search and Optimization Techniques** | | | |
| 2.1 Uninformed Search Strategies (BFS, DFS)<br>2.2 Informed Search (A*, Greedy Search)<br>2.3 Constraint Satisfaction Problems (CSP)<br>2.4 Genetic Algorithms and Optimization<br><br>Hands on practice on topics learned in this module as listed below: | 6 | 12% | **CO1,CO2** |

| | | | |
|---|---|---|---|
| **MODULE 3: Machine Learning for AI** | | | |
| 3.1 Introduction to Machine Learning<br>3.2 Supervised, Unsupervised, and Reinforcement Learning<br>3.3 Decision Trees, Naïve Bayes, and SVM<br>3.4 Evaluation Metrics for AI Models<br><br>Hands on practice on topics learned in this module as listed below: | 10 | 18% | CO2 |
| **MODULE 4: Fundamentals of Neural Networks** | | | |
| 4.1 Introduction to Neural Networks<br>4.2 Perceptron and Multi-Layer Perceptron (MLP)<br>4.3 Activation Functions and Backpropagation<br>4.4 Optimization Techniques (SGD, Adam)<br><br>Hands on practice on topics learned in this module as listed below: | 8 | 20% | CO3 |
| **MODULE 5: Deep Learning Architectures** | | | |
| 5.1 Convolutional Neural Networks (CNN)<br>5.2 Recurrent Neural Networks (RNN) and LSTMs<br>5.3 Generative Adversarial Networks (GANs)<br>5.4 Attention Mechanisms and Transformers<br><br>Hands on practice on topics learned in this module as listed below: | 6 | 15% | CO4, CO5 |
| **MODULE 6: Applications and Deployment of Deep Learning** | | | |
| 6.1 AI in Healthcare, Finance, and Robotics<br>6.2 Model Deployment using TensorFlow and PyTorch<br>6.3 AI Ethics and Bias in DL Models<br>6.4 Future Trends in AI and DL<br><br>Hands on practice on topics learned in this module as listed below: | 10 | 25% | CO4, CO5 |

| **TEXT BOOKS** | |
|---|---|
| 1 | "Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig. |
| 2 | "Deep Learning" by Ian Goodfellow, Yoshua Bengio, and Aaron Courville. |
| 3 | "Pattern Recognition and Machine Learning" by Christopher M. Bishop. |
| **REFERENCE BOOKS** | |
| 1 | "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron. |
| 2 | "Neural Networks and Deep Learning" by Michael Nielsen. |
| **E BOOKS** | |
| 1 | "Deep Learning with Python" by François Chollet. |
| 2 | "AI Superpowers" by Kai-Fu Lee. |
| **MOOC** | |
| 1 | https://www.coursera.org/specializations/deep-learning/ |
| 2 | https://www.udacity.com/course/ai-artificial-intelligence/ |

| COURSE TITLE | | CLOUD COMPUTING MANAGEMENT AND SECURITY LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA304L | COURSE CATEGORY | SEC | L-P-T | 0-1-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

| MODULE: Topic |
|---|
| **MODULE 1: Introduction to Cloud Computing Management**<br>**Task 1.** Deploy a Web App on a Cloud Platform (Cloud Management Principles & Infrastructure Management)<br><br>**Task 2.** Estimate Cloud Costs Using Cost Calculators (Cloud Cost Management)<br><br>**Task 3**. Set Up Basic Monitoring & Alerts (Cloud Monitoring and Optimization) |
| **MODULE 2: Cloud Security**<br>**Task 1.** Simple Data Migration to the Cloud<br><br>**Task 2**. Identify the Best Cloud Migration Strategy<br><br>**Task 3.** Enable Basic Security Settings on a Cloud Resource |
| **MODULE 3: Cloud Database and File System**<br><br>**Task 1.** Create a Simple Data Warehouse Schema in Google BigQuery **Task 2.** Upload and Organize<br><br>Data in a Cloud-Native File System **Task 3.** Explore General-Purpose Cloud Storage<br><br>**Task 4.** Hands-On with Different Cloud Storage Types |
| **MODULE 4: Managing Cloud Infrastructure**<br>**Task 1**. Create and Manage Virtual Machines (Virtualization)<br><br>**Task 2.** Implement Cloud Storage Management with AWS S3 or Google Cloud Storage<br><br>**Task 3**. Configure and Monitor Cloud Networking (Cloud Network Management) |
| **MODULE 5: Cloud Security Practices and Solutions**<br>**Task 1.** Implement server-side encryption for files uploaded to a cloud storage service (e.g., AWS S3 or Google Cloud Storage).<br><br>**Task 2**. Set up SSL/TLS encryption for a website hosted in the cloud to ensure secure communication between the user and the server. |

**MODULE 6: Cloud Security & Implementation of Cloud**

**Task 1.** Configure a basic security group on a cloud provider (e.g., AWS, Azure) to restrict access to a VM or service.

**Task 2**. Design a secure cloud architecture for a hypothetical company using best practices for identity management, data protection, and access control.

**Task 3**. Implement data encryption at rest and in transit for a cloud-hosted database or application.

| COURSE TITLE | CLOUD API & SERVICES | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA304 | COURSE CATEGORY | ME | L P T | 3-0-0 |
| Version 1.0 | Approval Details | | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | Students will learn cloud security best practices, including encryption, IAM, and threat mitigation. The course emphasizes hands-on learning through real-world case studies and practical implementations. |
| **Course Objective** | **1.Understand** the fundamentals, architecture, and significance of Cloud APIs in cloud computing.<br>2. **Identify** different cloud service models and APIs provided by cloud platforms.<br>3. **Analyze** security, compliance, and governance factors in cloud APIs.<br>4. **Utilize** cloud APIs for integrating storage, networking, and compute services.<br>5.**Create** cloud-based applications and automation workflows using APIs and DevOps tools. |
| **Course Outcome** | CO1: **Understand** cloud API fundamentals, architecture, and benefits.<br>CO2: **Identify** cloud service models, API types, and cloud providers.<br>CO3: **Analyze** security, compliance, and governance factors in cloud APIs.<br>CO4: **Utilize** cloud APIs for integrating storage, networking, and compute services.<br>CO5: **Create** cloud-based applications using APIs, DevOps, and automation techniques. |

**Prerequisites: Basic knowledge of Cloud Computing**

| Module Name | No of lectures | Weightage | CO-PO mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Cloud APIs** | | | |
| 1.1 Overview of Cloud Computing and APIs<br>1.2 Role and Importance of APIs in Cloud Services<br>1.3 RESTful vs. SOAP APIs in Cloud Computing<br>1.4 API Authentication & Authorization (OAuth, API Keys, JWT)<br>1.5 Cloud API Development & Challenges<br><br>Hands on practical based on this unit. | 5 | 10% | **CO-1** |
| **MODULE 2: Cloud Service Models and API Types** | | | |
| 2.1 Infrastructure as a Service (IaaS) APIs: AWS EC2, Google Compute Engine, Azure Virtual Machines<br>2.2 Platform as a Service (PaaS) APIs: AWS Lambda, Google App Engine, Azure Functions<br>2.3 Software as a Service (SaaS) APIs and Integration Techniques<br>2.4 API Management: Gateways, Rate Limiting, and Versioning<br>2.5 Cloud API Performance Optimization & Monitoring<br><br>Hands on practical based on this unit. | 8 | 15% | **CO-1 & CO-2** |

| | | | |
|---|---|---|---|
| **MODULE 3: Cloud Storage and Database APIs** | | | |
| 3.1 Cloud Storage APIs: AWS S3, Google Cloud Storage, Azure Blob Storage<br>3.2 Database APIs: AWS DynamoDB, Google Firestore, Azure Cosmos DB<br>3.3 Data Streaming and Messaging APIs: Kafka, AWS Kinesis, Google Pub/Sub<br>3.4 Cloud Backup, Disaster Recovery, and Migration APIs<br>3.5 Use Cases and Best Practices in Cloud Storage & Database APIs<br><br>Hands on practical based on this unit. | 8 | 20% | **CO-2** |
| **MODULE 4: Security and Governance in Cloud APIs** | | | |
| 4.1 Security Challenges in Cloud APIs (Data Leakage, DDoS Attacks)<br>4.2 API Security Measures: Encryption, Authentication, and Firewalls<br>4.3 Compliance and Legal Considerations (GDPR, HIPAA, ISO 27001)<br>4.4 Identity & Access Management (IAM) APIs for Secure Authentication<br>4.5 API Monitoring, Logging, and Risk Management<br><br>Hands on practical based on this unit. | 8 | 20% | **CO-3** |
| **MODULE 5: Cloud Networking and Compute APIs** | | | |
| 5.1 Networking APIs: AWS VPC, Azure Virtual Network, Google Cloud VPC<br>5.2 Compute APIs: AWS EC2, Azure Compute, Google Compute Engine<br>5.3 Load Balancing & Auto-Scaling APIs<br>5.4 DNS Management, Traffic Routing, and API Orchestration<br>5.5 API-Driven Infrastructure and Cloud Automation | 8 | 20% | **CO-4** |
| **MODULE 6 : Cloud API Development, Automation, and Future Trends** | | | |
| 6.1 API Development Lifecycle for Cloud Services<br>6.2 Automating Cloud Operations with APIs (Terraform, CloudFormation)<br>6.3 DevOps Integration and CI/CD Pipelines using Cloud APIs<br>6.4 Multi-Cloud and Hybrid Cloud API Strategies<br>6.5 Future Trends in Cloud API Management and AI-Driven Automation<br><br>Hands on practical based on this unit. | 8 | 15% | **CO-5** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | **1. "Designing Web APIs: Building APIs That Developers Love"** – Brenda Jin, Saurabh Sahni, Amir Shevat<br>2. **"Cloud Computing: Concepts, Technology & Architecture"** – Thomas Erl, Zaigham Mahmood, Ricardo Puttini |
| **REFERENCE BOOKS** | |
| 1. | 1. **"API Design Patterns"** – JJ Geewax<br>2. **"Cloud Computing: Theory and Practice"** – Dan C. Marinescu |
| **E BOOKS** | |
| 1. | **"REST API Design Handbook"** – Mark Masse |
| 2. | **"Google Cloud APIs: The Complete Guide"** – Google Cloud Documentation |

| MOOC | |
|---|---|
| 1. | **"Developing APIs with Google Cloud's Apigee API Platform"** – Coursera (Offered by Google Cloud) |
| 2. | "Introduction to Cloud Computing" by IBM on Coursera (coursera.org/learn/introduction-to-cloud) |

| COURSE TITLE | | Tableau LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Introduction to Tableau**

1. Open Tableau and explore the interface. Identify and label key components: Data pane, Sheet, Columns/Rows shelves, and Marks card.
2. Load a sample Excel file and classify the fields into dimensions and measures. Change the data type of fields where required.
3. Create a table showing Sales by Region. Use formatting to bold headers and apply color to high-performing regions.
4. Use the “Show Me” feature to visualize Profit by Sub-Category using different chart types and discuss when each is appropriate.
5. Create a visualization comparing sales and quantity using dual-axis and synchronize axes.
6. Perform a basic field operation: Create a calculated field to show "Profit Ratio = Profit/Sales" and display it in a chart.

**MODULE 2: Data Connection and Preparation**

1. Connect Tableau to a live MySQL database and load the sales data. Switch the connection to Extract mode and compare performance.
2. Connect Excel and CSV datasets and join them on the common field “Product ID.” Display joined fields in a table.
3. Clean imported data by renaming fields, removing null values, and changing data types appropriately.
4. Apply filters to show only the top 10 profitable products in the last quarter.
5. Create a sorted view of categories based on descending order of sales. Use manual sorting and field-based sorting.
6. Use Data Interpreter to clean a raw Excel sheet and describe the before/after structure.

**MODULE 3: Building Data Visualizations**

1. Create a bar chart showing total sales by category. Add data labels and color by profit.
2. Build a line chart to show the sales trend over the last 12 months. Use continuous date fields.
3. Use the sample Superstore dataset to create a pie chart showing market share by region.
4. Design a map showing sales by state in India. Use filled map visualization and add tooltips.
5. Add interactive filter controls to show data by Region and Sub-Category.
6. Create a parameter to switch between Sales and Profit in a bar chart and update title dynamically using calculated fields.

**MODULE 4: Interactive Dashboards and Storytelling**

1. Create a dashboard combining three different sheets (Sales by Region, Profit by Category, and Top 10 Customers).
2. Add a filter to control all sheets in the dashboard simultaneously using a global filter.
3. Implement dashboard actions: Create a filter action where clicking a region updates customer data below.
4. Use a parameter action to allow the user to switch views between bar and pie chart in the dashboard.
5. Optimize dashboard performance by removing unused fields and using extracts.

6. Create a story using at least three sheets to present an analysis of product performance across different years.

**MODULE 5: Advanced Tableau Features**

1. Create a calculated field to show Discount Category: “Low,” “Medium,” “High” based on discount % thresholds.
2. Use a table calculation to compute Running Total of Sales by Month.
3. Implement a FIXED LOD expression to show average profit per customer regardless of filter applied.
4. Build a forecast for next 6 months' sales using built-in forecasting. Customize confidence intervals.
5. Create a trend line and explain the correlation between Discount and Profit visually.
6. Optimize performance by creating an extract and using aggregated data for large datasets

**MODULE 6: Case Studies and Industry Applications**

1. Create a sales analysis dashboard for a retail company showing sales trends, profit heatmaps, and region-wise performance.
2. Design a financial KPI dashboard showing Revenue, Expenses, and Net Profit using bullet graphs and gauges.
3. Create a healthcare data dashboard that highlights patient counts by department and monthly admission trends.
4. Analyze a real-world dataset (e.g., stock data or COVID-19 cases) and build an insightful visualization.
5. Redesign a cluttered chart using best practices (color theory, grid alignment, interactivity).
6. Evaluate an existing public Tableau dashboard (from Tableau Public) and suggest improvements based on data storytelling principles.

| COURSE TITLE | | U/UX DESIGN LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA305L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**Module 1: Introduction to UI/UX Design**

1. Setup Figma/Adobe XD and Create Your First Artboard
2. Identify and Compare Good vs Bad UI Designs
3. Apply Nielsen's 10 Usability Heuristics to an Existing Website
4. Create a Design Thinking Canvas for a Selected Problem
5. Conduct a Competitor Analysis of Two Similar Apps

***Module 2: User Research & Information Architecture***

6. Design a User Survey Form for a Target Application
7. Conduct a User Interview and Document Insights
8. Create User Personas Based on Research
9. Build an Empathy Map for a Selected User Persona
10. Develop a Customer Journey Map
11. Perform a Card Sorting Exercise
12. Create a Site Map and Define Navigation for a Website

***Module 3: Visual & Interaction Design Principles***

13. Design a Color Palette Based on Product Emotions
14. Choose and Apply Typography for Web and Mobile
15. Create a UI Component Library (Buttons, Inputs, Cards)
16. Design a Homepage Using Visual Hierarchy Principles
17. Add Microinteractions (Hover, Click, Tap)
18. Apply Gamification Techniques to a User Flow

***Module 4: Wireframing & Prototyping***

19. Create Low-Fidelity Wireframes (Paper/Digital)

20. Convert Low-Fidelity Wireframes into High-Fidelity Screens
21. Design a Multi-Screen App Flow (3–5 Screens)
22. Build an Interactive Prototype with Transitions
23. Perform Developer Handoff and Export Design Specs

***Module 5: Usability Testing & Accessibility***

24. Conduct a Heuristic Evaluation of a Peer Project
25. Perform A/B Testing on Two Versions of a Design
26. Conduct a Web Accessibility Audit (WCAG Guidelines)
27. Redesign a Screen for Accessibility (Visually Impaired Users)
28. Create Responsive Layouts for Mobile, Tablet, and Desktop

***Module 6: UI/UX Trends & Emerging Technologies***

29. Design a Dark Mode Version of an Existing App
30. Create a Conceptual Mockup for Voice UI or AR-based UI

| COURSE TITLE | | E COMMERCE & DIGITAL MARKETING LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Introduction to E-Commerce**

1. **Explore and compare** the user interface, pricing, and features of Shopify, WooCommerce, and Magento. Create a short presentation highlighting their strengths for different business types.
2. **Design a mock homepage** for an e-commerce website using free tools like Canva or Figma, showcasing key components: navigation, product categories, cart, login.
3. **Create a timeline infographic** using any online tool (e.g., Canva) that visually shows the evolution of e-commerce from 1990s to today.
4. **Prepare a comparison chart** between Traditional Commerce and E-Commerce in terms of setup cost, reach, transaction handling, customer interaction, and scalability.
5. **Research and present** the latest trends in Indian e-commerce (e.g., ONDC, voice search, live commerce) using Google Trends and Statista.
6. **Simulate creating an online store** on Shopify (free trial version) or WooCommerce and upload at least 3 sample products with pricing and description.

**MODULE 2: E-Commerce Business Models & Payment Systems**

1. **Classify 6 real-world businesses** (e.g., Amazon, OLX, Etsy, Fiverr, BigBasket, Swiggy) into B2B, B2C, C2C, or C2B models and justify the classification.
2. **Create a visual flowchart** of a Dropshipping model and compare it with a traditional inventory-based model using draw.io or Lucidchart.
3. **Simulate a payment transaction** using Razorpay's Payment Gateway demo. Analyze the checkout experience and prepare a report.
4. **Compare at least 3 mobile wallet apps** (e.g., Google Pay, PhonePe, Paytm) based on features, UI, security, and ease of use.
5. **Prepare a list of security protocols** and tools used to secure online transactions (e.g., SSL, PCI-DSS). Briefly describe their role.
6. **Develop a basic webpage** using HTML/CSS that simulates a checkout page with dummy cart items and a "Pay Now" button.

**MODULE 3: Fundamentals of Digital Marketing**

1. **Conduct a live SEO audit** of any website using tools like Ubersuggest or SEOptimer and list on-page/off-page SEO recommendations.
2. **Design a sample Google Ads campaign** using Google Ads interface (without launching) for a fictional product – define target audience, budget, keywords.
3. **Create a newsletter email template** using Mailchimp or any free HTML email template generator, for promoting a new e-commerce product.
4. **List and compare** 3 influencer marketing campaigns (India-based) with platform used, campaign goal, and engagement results.
5. **Build a landing page wireframe** in Canva or Figma for a digital product launch with focus on SEO elements like title, meta, content, CTA.

6. **Research and summarize trends** in digital marketing post-2020 using Google Trends and industry reports.

**MODULE 4: Social Media & Content Marketing**

1. **Create a one-week social media content calendar** for a fictional e-commerce fashion brand targeting Instagram and LinkedIn.
2. **Design a short YouTube video script or storyboard** for a product unboxing or tutorial using Canva or video tools like InVideo.
3. **Create a blog outline** (title, intro, subheads, keywords) for a topic like "Top 5 gadgets to buy in 2025" using ChatGPT or similar tools.
4. **Register for an affiliate program** like Amazon Associates or Flipkart and simulate creating affiliate links for 3 products.
5. **Analyze a viral social media campaign** (e.g., Zomato ads, Amul creatives) and present how content and platform were used strategically.
6. **Create a sponsored post template** in Canva including branding, CTA, hashtags, and product visuals for any product category.

**Module 5: Analytics & Conversion Optimization**

1. **Use Google Analytics demo account** to analyze website traffic and present a report on top pages, bounce rate, and user location.
2. **Draw a conversion funnel diagram** for an online store using Lucidchart or Canva, indicating key touchpoints (landing → product → cart → payment).
3. **Create an A/B test plan** comparing two versions of a landing page headline or CTA button using tools like Google Optimize.
4. **Simulate retargeting ads** using Meta Ads Manager walkthrough (or research-based mock) to define a campaign for abandoned cart users.
5. **Design a lead capture form** using Google Forms or Typeform to collect user emails for newsletter subscription.
6. **Present a heatmap analysis case study** from Hotjar or CrazyEgg and summarize user behavior insights and optimizations.

**Module 6: Legal, Security & Ethical Aspects**

1. **Draft a sample privacy policy** for an e-commerce site using a template generator and tailor it for GDPR compliance.
2. **List top 5 cybersecurity threats** for e-commerce businesses and suggest prevention techniques for each (can be in tabular format).
3. **Analyze a real online fraud case** (Indian context if possible) and report how it happened and what prevention could have worked.
4. **Simulate an AI-powered chatbot demo** (like Tidio or Intercom) and explain how it improves security or user interaction.
5. **Create a checklist** of ethical digital marketing practices vs. unethical ones with examples (e.g., clickbait, misleading ads).
6. **Design a quiz** for consumers with 5 questions to test their awareness about online scams and fraud prevention tips.

| COURSE TITLE | | GENERATIVE AI PROMPT ENGINEERING LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206L | COURSE CATEGORY | SEC | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**MODULE 1: Fundamentals of Prompt Engineering**

1. **Compare AI, RPA, and Generative AI** using a mind-mapping or diagram tool (e.g., Lucidchart, Miro). Create a digital map showing their differences, similarities, and overlapping use cases.
2. **Research and prepare a timeline** of automation evolution from rule-based systems to modern AI and RPA. Present this using a visual infographic.
3. **Explore any AI-based tool** (e.g., ChatGPT, Google Bard) and summarize how it can solve real-world problems in education, marketing, or healthcare.
4. **Interview an AI tool** (like ChatGPT) to list current trends in AI and automation. Ask at least 5 relevant prompts and analyze its responses.
5. **Create a short video or screencast** summarizing the importance and scope of emerging technologies, highlighting one practical example from each.
6. **Build a simple web page or poster** using Canva or HTML summarizing what you learned about AI, RPA, and Generative AI with industry use case snippets.

**MODULE 2: Fundamentals of Prompt Engineering**

1. **Experiment with prompt formats** (declarative, interrogative, instructive) in ChatGPT. Evaluate which format generates the most accurate answer for a given task (e.g., recipe creation, explanation of a topic).
2. **Design prompts** for 3 different tasks: a creative story, a business email, and a Python code snippet. Evaluate how well the output meets expectations.
3. **Tune a single prompt iteratively** (change tone, add constraints) and compare results. Document changes and resulting outputs in a table.
4. **Create prompts optimized for a customer service bot** and a creative writing assistant. Highlight the differences in style and detail.
5. **Use a prompt to summarize a news article** and another to write a poem on the same topic. Reflect on how prompt framing affects AI output.
6. **Compare responses from different AI models** (e.g., GPT-3.5 vs Claude or Gemini, if available) using the same prompt. Analyze their response styles.

**MODULE 3: Robotic Process Automation (RPA)**

1. **Design a simple RPA workflow** using UiPath or Power Automate that automatically downloads an email attachment and saves it to a folder.
2. **Create a bot to automate a repetitive task**, such as extracting data from an Excel sheet and sending it via email.
3. **Simulate an attendance marking bot** that records student names and time stamps in a spreadsheet.
4. **Use UiPath/Blue Prism/Automation Anywhere trial version** to build a bot that fills out an online form based on input from an Excel file.

5. **Explore RPA in real industries:** Pick one domain (e.g., banking) and create a flowchart showing how RPA is used to streamline operations.
6. **Record and document the steps** to create a basic task automation using any open-source RPA tool. Submit a step-by-step guide or video.

**MODULE 4: Introduction to Generative AI**

1. **Generate an image using Midjourney or DALL·E** based on a creative prompt. Write the prompt and reflect on the quality of the output.
2. **Generate a short poem, blog post, and code snippet** using ChatGPT. Compare how each content type is handled differently by the model.
3. **Use OpenAI or similar tool to translate text** between English and Hindi, then generate a poem in Hindi. Reflect on linguistic fluency and style.
4. **Compare a GAN-based image output** and a transformer-based text output. Identify the role of the underlying model in content generation.
5. **Use ChatGPT to simulate a coding tutor** – give it a prompt to explain recursion with JavaScript examples. Evaluate its teaching ability.
6. **Create a comic strip idea using AI tools**: generate the story using GPT and visuals using Midjourney. Assemble in Canva or PowerPoint.

**Module 5: Applications and Case Studies**

1. **Design a use case of Generative AI in Marketing**: Create social media content (text + image) using AI tools for a fictional brand.
2. **Choose a creative field** (e.g., design, video editing, writing) and showcase how AI tools can enhance productivity. Provide before/after results.
3. **Simulate a business scenario** where RPA is used to automate invoice processing. Explain steps and tools used.
4. **Develop a prompt for a fashion design assistant AI** and show how the AI generates a seasonal design concept.
5. **Identify an industry problem** (e.g., customer query response delays). Use prompt engineering to simulate a chatbot solution with examples.
6. **Document 3 real-life AI/RPA applications** from recent news or case studies and present their impact on business operations.

**Module 6: Ethical Considerations & Future Trends**

1. **Analyze a biased AI response** using ChatGPT. Try a few prompts and note any unintended stereotypes or biases. Suggest how to mitigate them.
2. **Simulate an ethical dilemma** using AI – ask ChatGPT to solve a real-world conflict and analyze the fairness of its solution.
3. **Compare privacy policies** of AI platforms (OpenAI, Bard, etc.). Highlight how they handle user data and potential risks.
4. **Design a responsible AI charter** for your classroom/lab use. Mention rules on data privacy, fairness, and ethical prompts.
5. **Create a poster or visual report** on AI risks and mitigation – focus on security, hallucinations, misuse, and privacy breaches.
6. **Predict AI's role in future jobs**: Use ChatGPT to simulate 3 career profiles of 2035 and write a blog post or video script on them.

| COURSE TITLE | | TABLEAU | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA305 | COURSE CATEGORY | ME 5.3 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 18-4-2025 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides an in-depth introduction to **Tableau**, a leading data visualization tool used for business intelligence and data analytics. The course covers **data connections, visualization techniques, dashboard creation, and advanced analytics**, helping students develop skills to analyze and present data effectively. By the end of the course, students will be able to build **interactive dashboards and reports** to support decision-making processes. |
| **Course Objective** | • Understand the fundamentals of Tableau and its role in data visualization.<br>• Connect to various data sources and perform data preparation using Tableau.<br>• Create and customize visualizations using different chart types.<br>• Design **interactive dashboards** and **stories** for data analysis.<br>• Utilize advanced Tableau features such as calculated fields, parameters, and LOD expressions.<br>• Implement data-driven decision-making using Tableau. |
| **Course Outcome** | CO1: Understand Tableau's interface, data visualization principles, and data preparation techniques.<br>CO2: Connect Tableau with various data sources and perform data transformation operations.<br>CO3: Build interactive and customized data visualizations for effective analysis.<br>CO4: Design and optimize dashboards with advanced interactivity features.<br>CO5: Apply advanced analytical techniques and performance optimization strategies.<br>CO6: Analyze case studies and implement industry best practices in Tableau. |
| **Prerequisites:** Basic knowledge of Data Analytics & Visualization, understanding of SQL and Databases, and familiarity with Spreadsheets (Excel/Google Sheets). | |

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Tableau** | | | |
| 1.1 Introduction to Data Visualization, 1.2 Importance of Tableau in Business Intelligence, 1.3 Tableau Interface and Components, 1.4 Data Types and Field Operations | 8 | 10% | **CO-1** |
| **MODULE 2: Data Connection and Preparation** | | | |
| 2.1 Connecting to Different Data Sources (Excel, SQL, Cloud, etc.), 2.2 Data Extraction and Live Connection, 2.3 Data Cleaning, Filtering, and Sorting, 2.4 Merging and Joining Data | 7 | 15% | **CO-2** |
| **MODULE 3: Building Data Visualizations** | | | |
| 3.1 Creating Basic Charts (Bar, Line, Pie, Scatter), 3.2 Creating Maps and Geospatial Visualizations, 3.3 Using Filters and Parameters in Charts, 3.4 Customizing Charts (Colors, Labels, Tooltips) | 8 | 20% | **CO-3** |

| | | | |
|---|---|---|---|
| **MODULE 4: Interactive Dashboards and Storytelling** | | | |
| 4.1 Creating Dashboards in Tableau, 4.2 Adding Filters, Actions, and Navigation, 4.3 Designing for Performance and Usability, 4.4 Creating a Story in Tableau | 10 | 25% | **CO-4** |
| **MODULE 5: Advanced Tableau Features** | | | |
| 5.1 Creating Calculated Fields and Using Table Calculations, 5.2 Using LOD (Level of Detail) Expressions, 5.3 Implementing Forecasting and Trend Analysis, 5.4 Performance Optimization in Tableau | 8 | 20% | **CO-5** |
| **MODULE 6: Case Studies and Industry Applications** | | | |
| 6.1 Real-world Applications of Tableau, 6.2 Case Studies from Finance, Healthcare, and Retail, 6.3 Best Practices in Data Visualization | 4 | 10% | **CO-6** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | Joshua N. Milligan, "Learning Tableau 2022," Packt Publishing. |
| 2. | Daniel G. Murray, "Tableau Your Data!: Fast and Easy Visual Analysis with Tableau Software," Wiley. |
| **REFERENCE BOOKS** | |
| 1. | Fundamentals of Data Visualization: **https://serialmentor.com/dataviz/** |
| 2. | Tableau Public - Learning Resources: **https://public.tableau.com/en-us/s/resources** |
| **MOOC** | |
| 1. | Tableau for Beginners - Coursera: **https://www.coursera.org/learn/data-visualization-tableau** |
| 2. | Data Visualization with Tableau - Udemy: **https://www.udemy.com/course/tableau-for-beginners/** |
| 3. | Tableau Training - EdX: **https://www.edx.org/course/data-visualization-with-tableau** |

| COURSE TITLE | | UI UX DESIGN | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA204 | COURSE CATEGORY | ME5.2 | L- P- T | 3-0-0 |
| Version | 1.0 | Approval Details | 07/2024 | | | |
| ASSESSMENT SCHEME | | | | | | |
| First Periodical Assessment | | Second Periodical Assessment | Assignment/ Portfolio Design | Quiz/ Surprise Test | Attendance | ESE |
| 10% | | 10% | 10% | 10% | 10% | 50% |
| Course Description | | This course explores the fundamentals of **User Interface (UI) and User Experience (UX) Design** with a focus on **human-centered design, usability principles, visual aesthetics, prototyping, and modern UX research methodologies**. Students will gain hands-on experience in **wireframing, usability testing, interaction design, and the latest UI/UX trends** using **industry-standard tools like Figma, Adobe XD, Sketch, and usability testing frameworks**. | | | | |
| Course Objective | | 1. To introduce students to **UI/UX design principles** and their role in product development.<br>2. To equip students with **user research techniques**, including **persona creation, journey mapping, and information architecture**.<br>3. To teach the **principles of visual and interaction design** for effective UI/UX solutions.<br>4. To provide practical experience in **wire framing, prototyping, and usability testing**.<br>5. To familiarize students with **modern UI/UX trends, accessibility standards, and emerging technologies** | | | | |
| Course Outcome | | 1. **CO1 Understand core UI/UX design principles, usability heuristics, and design thinking methodologies.**<br>2. **CO2 Conduct user research, persona development, journey mapping, and create structured information architecture.**<br>3. **CO3 Apply principles of visual design, interaction design, accessibility, and user behavior psychology.**<br>4. **CO4 Develop wireframes, prototypes, and perform usability testing to validate user experience.**<br>5. **CO5 Integrate responsive design, accessibility compliance, and emerging UI/UX trends in real-world applications.** | | | | |
| **Prerequisites:** Basic understanding of Software Engineering process | | | | | | |
| **MODULE 1: Introduction to UI/UX Design** | | | | | | |
| Introduction to UI & UX, Differences Between UI and UX, Importance of User-Centered Design, Usability Heuristics (Nielsen's 10 Principles), UI/UX in Product Development, Design Thinking Process | | | | **6** | **10%** | **CO1** |
| **Module 2: User Research & Information Architecture** | | | | | | |
| Understanding Users & Behavior, Conducting User Research, Types of Research (Qualitative & Quantitative), Surveys & | | | | **6** | **15%** | **CO2** |

| | | | |
|---|---|---|---|
| Interviews, Persona Creation, Empathy Mapping, Customer Journey Mapping, Card Sorting, Information Architecture, Site Maps & Navigation Systems | | | |
| **MODULE 3: Visual & Interaction Design Principles** | | | |
| Fundamentals of Visual Design, Color Psychology in UI, Typography & Readability, UI Components & Design Systems, Interaction Design, Motion UI & Micro interactions, Affordances & Signifiers in UI, Gamification & Persuasive Design | **7** | **20%** | **CO3** |
| **MODULE 4: Wireframing & Prototyping** | | | |
| Wireframing Techniques (Low/High Fidelity), Paper Prototyping, Digital Wireframing (Figma, Adobe XD, Sketch), Storyboarding & Scenario Mapping, Interactive Prototypes, Design Handoff to Developers | **10** | **20%** | **CO4** |
| **MODULE 5: Usability Testing & Accessibility** | | | |
| Usability Testing Methods (A/B Testing, Heuristic Evaluation, Eye-Tracking), Cognitive Load & UX, Web Content Accessibility Guidelines (WCAG), Designing for Different User Groups (Elderly, Disabled, Children), Responsive UI Design, Adaptive vs. Responsive Layouts | **10** | **20%** | **CO5** |
| **MODULE 6: UI/UX Trends & Emerging Technologies** | | | |
| Dark Mode UI, Neumorphism, Material Design & Fluent Design, Conversational UI (Chatbots, Voice UI), Augmented Reality (AR) & Virtual Reality (VR) in UX, Ethical Considerations in UX Design | **6** | **15%** | **CO5** |
| **TEXTBOOKS** | | | |
| 1. | The Design of Everyday Things – Don Norman | | |
| 2 | Don't Make Me Think: A Common Sense Approach to Web Usability – Steve Krug | | |
| **REFERENCE BOOKS** | | | |
| 1. | About Face: The Essentials of Interaction Design – Alan Cooper | | |
| 2. | Laws of UX: Using Psychology to Design Better Products – Jon Yablonski | | |
| **E BOOKS** | | | |
| 1. | Universal Principles of Design – William Lidwell https://www.amazon.com/dp/1592535879 | | |
| 2 | Google Material Design Guidelines – Google https://material.io/design/ | | |
| 3 | UX Design Guide – Nielsen Norman Group https://www.nngroup.com/articles/ux-guide/ | | |
| 4 | Web Content Accessibility Guidelines (WCAG) – W3C https://www.w3.org/WAI/standards-guidelines/wcag/ | | |
| **MOOC** | | | |
| 1. | UI/UX Design Specialization – Coursera https://www.coursera.org/specializations/ui-ux-design | | |
| 2. | Human-Computer Interaction for UX Design – edX https://www.edx.org/course/human-computer-interaction-ux-design | | |
| 3. | Ultimate Figma UI/UX Design Course – Udemy https://www.udemy.com/course/figma-ux-ui-design/ | | |
| 4. | Google UX Design Course Playlist – YouTube https://www.youtube.com/results?search_query=google+ux+design | | |

| COURSE TITLE | | Artificial Intelligence & Machine Learning Lab | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA206L | COURSE CATEGORY | ME(Elective 3) | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | | | | |

**ASSESSMENT SCHEME**

| Assignments | Internal Exam | Attendance | ESE |
|---|---|---|---|
| 40% | 40% | 10% | - |

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Artificial Intelligence** | | | |
| 1. AI-Powered Chatbot for Customer Support<br>2. Predictive Analytics for Sales Forecasting<br>3. AI-Based Image Recognition for Object Detection | 3 | 15% | CO1 |
| **MODULE 2 : Search and Optimization Techniques** | | | |
| **1.** Develop a program to find a path between two points in a maze using Breadth-First Search (BFS) and Depth-First Search (DFS)**.**<br>2. Implement A* and Greedy Best-First Search to solve the classic 8-puzzle problem (a sliding puzzle with 3x3 grid).<br>3. Solve a Sudoku puzzle using Constraint Satisfaction Problem (CSP) techniques and Genetic Algorithms**.** | 3 | 15% | CO 2,CO3 |
| **MODULE 3: Machine Learning for AI** | | | |
| 1. Write a Python program demonstrating how a simple ML model (e.g., Linear Regression) contributes to an AI system.<br>2. Analyze and demonstrate bias issues in a simulated AI recruitment model trained on historical hiring data.<br>3. Implement a smart surveillance system that detects and tracks faces in real-time for security purposes. | 8 | 20% | |
| **MODULE 4: Introduction to Machine Learning** | | | |
| 1. Use Supervised Learning (Regression) to predict house prices based on features like area, number of rooms, and location.<br>2. Use Unsupervised Learning to segment customers based on purchasing behavior.<br>3. Classify emails as spam or not spam using Decision Trees, Naïve Bayes, and SVM.)<br>4. Design an AI system that reads user movie reviews and automatically classifies them as positive or negative. | 8 | 20% | CO 1 |
| **MODULE 5: Machine Learning Workflow and Tools** | | | |

| | | | |
|---|---|---|---|
| 1. Define a machine learning problem to predict whether a customer will churn based on historical service data (e.g., tenure, usage, complaints).<br>2. Collect and clean a dataset for housing prices (e.g., size, | 4 | 15% | CO 2,C O3, |
| location, number of rooms). Handle missing values, outliers, and convert categorical variables.<br>3. Select and evaluate models (Logistic Regression, Decision Tree, SVM) to predict credit score levels (Good/Bad). Use metrics like Accuracy, Precision, Recall, F1-score.<br>4. Write a Python script to load a student dataset, preprocess it, and build a regression model to predict final exam scores.<br>5. Use NumPy, Pandas, and Scikit-learn to build and evaluate a classification model on the Pima Indians Diabetes dataset. | | | CO4 |
| **MODULE 6: Case Studies** | | | |
| 1. Build a regression model to predict daily or monthly sales for a retail store based on historical sales data, promotions, holidays, store location, and footfall.<br>2. Predict weekly revenue for a restaurant using features like customer reviews, location, cuisine type, and past performance.<br>3. Develop a regression model to estimate a player's salary based on performance metrics such as goals scored, matches played, assists, and market value.<br>4. Build a classification model to predict whether a patient is likely to develop diabetes based on features such as BMI, age, glucose level, and blood pressure.<br>5. Detect fraudulent transactions using machine learning models trained on transaction history, location, and spending patterns. | 4 | 15% | CO 5,C O6 |

| COURSE TITLE | Artificial Intelligence & Machine Learning | | | CREDITS | 3 |
|---|---|---|---|---|---|
| COURSE CODE | MCA206 | COURSE CATEGORY | ME(Elective 3) | L-P-T | 3-0-0 |
| Version 1.0 | Approval Details | | | | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces the fundamentals of Artificial Intelligence (AI) and Machine Learning (ML), including intelligent agents, search techniques, knowledge representation, supervised and unsupervised learning, model evaluation, and real-world case studies. Students will gain both theoretical knowledge and hands-on experience using Python and popular libraries such as NumPy, Pandas, and Scikit-learn, preparing them to apply AI/ML techniques to solve practical problems ethically and efficiently. |
| **Course Objective** | To provide foundational knowledge of AI, its history, applications, and ethical aspects.<br>To develop problem-solving skills using search strategies and optimization techniques.<br>To introduce core machine learning concepts including supervised, unsupervised, and reinforcement learning.<br>To equip students with the skills to preprocess data, select appropriate models, and evaluate performance.<br>To familiarize students with modern ML tools and libraries for implementation.<br>To apply AI/ML concepts through case studies in various domains like healthcare, retail, and finance. |
| **Course Outcome** | By the end of the course, students will be able to:<br><br>CO1: Remember the history, applications, and terminology of AI and ML.<br>CO2: Understand intelligent agents, search techniques, and ML types.<br>CO3: Apply AI search algorithms and ML models to solve real-world problems<br>CO4: Analyze model performance using metrics like accuracy, precision, and recall.<br>CO5: Evaluate different algorithms for suitability in specific domains.<br>CO6: Create end-to-end AI/ML solutions using appropriate tools and libraries. |

| MODULE | No. of lectures | Weightage (%) | CO Mapped |
|---|---|---|---|
| **MODULE 1: Introduction to Artificial Intelligence** | | | |
| 1.1 Definition and History of AI,<br>1.2 Applications of AI<br>1.3 Intelligent Agents and Environments<br>1.4 Problem-Solving using AI<br>1.5 AI Ethics and Social Implications | 5 | 10% | CO1,CO2 |
| **MODULE 2 : Search and Optimization Techniques** | | | |
| 2.1 Uninformed Search Strategies (BFS, DFS), 2.2 Informed Search (A*, Greedy Search), 2.3 Constraint Satisfaction Problems (CSP), 2.4 Genetic Algorithms and Optimization | 6 | 12% | CO3,CO4 |
| **MODULE 3: Machine Learning for AI** | | | |

| | | | |
|---|---|---|---|
| 3.1 Introduction to Relationship between AI and ML, 3.2 Applications of ML in AI Domains 3.3 ML Techniques Powering AI Systems, ML Models Used in AI, Challenges and Future Scope | 10 | 18% | CO5,CO6 |
| **MODULE 4: Introduction to Machine Learning** | | | |
| 1.1 History and evolution of Machine Learning 1.2 Applications in industries: Healthcare, Finance, Marketing, 1.3 Types of Machine Learning : Supervised, Semi supervised, Unsupervised | 8 | 20% | CO1,CO2 |
| **MODULE 5: Machine Learning Workflow and Tools** | | | |
| 2.1 Problem formulation, 2.2 Data collection and pre-processing, 2.3 Model selection and evaluation, 2.4 Overview of Python for ML, 2.5 Introduction to ML libraries: NumPy, Pandas, Scikit-learn | 6 | 15% | CO3,CO4 |
| **MODULE 6: Case Studies** | | | |
| 6.1 REGRESSION Case Studies 6.2 Retail Store Sales Prediction 6.3 Restaurant Sales Prediction 6.4 Sport Player Salary Prediction 6.5 CLASSIFICATION Case Studies 6.6 Diabetes Prediction for Preventive Care 6.7 Credit Card Fraud Detection 6.8 Heart Diseases Prediction for Preventive Care | 10 | 25% | CO5,CO6 |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1 | "Artificial Intelligence: A Modern Approach" *by Stuart Russell and Peter Norvig (4th Edition, Pearson, 2020)* |
| 2 | “Machine Learning: A Probabilistic Perspective" *by Kevin P. Murphy (MIT Press, 2012)* |
| 3 | "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" *by Aurélien Géron (2nd Edition, O'Reilly Media, 2019)* |
| **REFERENCE BOOKS** | |
| 1 | "Pattern Recognition and Machine Learning" *by Christopher M. Bishop* |
| **E BOOKS** | |
| 1 | "Deep Learning" *by Ian Goodfellow, Yoshua Bengio, and Aaron Courville* |
| 2 | "Python Machine Learning" *by Sebastian Raschka and Vahid Mirjalili* |
| **MOOC** | |
| 1 | Coursera: "Introduction to Data Science" by University of Washington |
| 2 | AI For Everyone *by Andrew Ng on Coursera*<br><br>Machine Learning *by Andrew Ng on Coursera* |

| COURSE TITLE | CYBER SECUIRTY & ETHICAL HACKING | | CREDITS |
|---|---|---|---|
| COURSE CODE | MCA305_6.1 | COURSE CATEGORY | ME | L-P-T |
| Version | 1.0 | Approval Details | 07-2024 | |

**ASSESSMENT SCHEME**

| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test/ Quiz | Attendance | |
|---|---|---|---|---|---|
| 10% | 10% | 10% | 10% | 10% | |

| | |
|---|---|
| **Course Description** | This course provides an in-depth exploration of cyber security principles, threats, and defenses in digital landscape. Students will examine foundational concepts such as the CIA Triad, cyber threat vulnerabilities, along with the implications of cyber crimes and digital forensics. The course covers and solutions for securing mobile and wireless devices, as well as addressing organizational secur Additionally, students will be introduced to ethical hacking practices and security testing to identi vulnerabilities. By the end of the course, students will be equipped with the knowledge to assess cyber security risks in diverse environments. |
| **Course Objective** | • To Understand the core principles of cyber security.<br>• To Evaluate legal and forensic dimensions in cyber security.<br>• To Identify and mitigate security risks in mobile and wireless environments.<br>• To Assess and address organizational cyber security threats.<br>• To Apply ethical hacking techniques for security testing. |
| **Course Outcome** | After the end of course student will be able to<br><br>**CO1**. Understand Cyber Security Fundamentals<br>**CO2.** Analyze Legal and Forensic Aspects in Cyber Security<br>**CO3.** Assess Security Risks in Mobile and Wireless Devices<br>**CO4**. Examine Organizational Cyber Security Issues<br>**CO5**. Apply Ethical Hacking and Security Testing Techniques |

| MODULE | No of lectures | Weightage | CO |
|---|---|---|---|
| **MODULE 1: Introduction to Cyber Security** | | | |
| 1.1 Basic Cyber Security Concepts,1.2 layers of security, Vulnerability, threat, Harmful acts, 1.3 Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, 1.4 Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, 1.5 Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy. | 5 | 10% | C |
| **MODULE 2: Cyberspace and the Law & Cyber Forensics** | | | |
| 2.1 Introduction, Cyber Security Regulations, Roles of International Law. 2.2 The INDIAN Cyberspace, National Cyber Security Policy. 2.3 Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, 2.4 Cyber Forensics and Digital evidence, Forensics Analysis of Email, 2.5 Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics | 6 | 16% | |
| **MODULE 3: Cybercrime: Mobile and Wireless Devices** | | | |

| | | | |
|---|---|---|---|
| 3.1 Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, 3.2 Credit card Frauds in Mobile and Wireless Computing Era, 3.3 Security Challenges Posed by Mobile Devices, 3.4 Registry Settings for Mobile Devices, Authentication service Security, 3.5 Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops. | 6 | 16% | |
| **MODULE 4 Cyber Security: Organizational Implications**: | | | |
| 4.1 Introduction, cost of cybercrimes and IPR issues, web threats for organizations, 4.2 security and privacy implications, 4.3 social media marketing: security risks and perils for organizations, 4.4 social computing and the associated challenges for organizations | 8 | 18% | |
| **MODULE 5: Ethical Hacking Foundations** | | | |
| 5.1 An Introduction to Ethical Hacking : Security Fundamental, Security testing, Hacker and Cracker,5.2 Test Plans-keeping It Legal, Ethical and Legality 5.3 The Technical Foundations of Hacking: The Attacker's Process, The Ethical Hacker's Process, Security and the Stack | 10 | 20% | |
| **MODULE 6: Footprinting and Scanning** | | | |
| 6.1 Information Gathering, Determining the Network Range, 6.2 Identifying Active Machines, 6.3 Finding Open Ports and Access Points, 6.4 OS Fingerprinting Services,6.5 Mapping the Network Attack Surface | 10 | 20% | |

**TEXT BOOKS**

| | |
|---|---|
| 1 | Nina Godbole and SunitBelpure, Cyber Security Understanding Cyber Crimes,Computer Forensics and Legal Perspect |
| | B.B.Gupta,D.P.Agrawal,HaoxiangWang,ComputerandCyberSecurity:Principle s, Algorithm, Applications, and Perspect |
| | ISBN 9780815371335,2018. |
| 2 | Hacking the Hacker, Roger Grimes, Wiley |
| | Learn Ethical Hacking from Scratch: Your stepping stone to penetration testing Zaid Sabih by Packt Publishi |
| 3 | Learning Nessus for Penetration TestingBy Himanshu Kumar, PACKT publication |
| 4 | |

**REFERENCE BOOKS**

| | |
|---|---|
| 1 | Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRCPress. |
| 2 | Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&FGroup. |
| 3 | Ethical Hacking and Penetration Testing Guide Rafay Baloch by CRC Press |

**E BOOKS**

| | |
|---|---|
| 1 | "The Basics of Cyber Safety: Computer and Mobile Device Safety Made Easy" by John Sammons |
| 2 | "Cybersecurity and Cyberwar: What Everyone Needs to Know" by P.W. Singer and Allan Friedman |
| 3 | "Hacking: The Art of Exploitation" by Jon Erickson |

**MOOC**

| | |
|---|---|
| 1 | https://nptel.ac.in/courses/106/105/106105217/ |
| | https://www.edx.org/learn/cybersecurity/rochester-institute-of-technology-cybersecurity-fundamentals |

| | |
|---|---|
| 2<br>3 | https://www.udemy.com/course/the-complete-internet-security-privacy-course-volume-1/?srsltid=AfmBOoqProuYCdjxqX_KBWXg6mmaTlhnwNuUo00_PCeB2YZwSyjEeyOE&couponCode=S |

| COURSE TITLE | | CYBER SECURITY & ETHICAL HACKING LAB | | | CREDITS | 1 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA305 L | COURSE CATEGORY | CM | L-P-T | 0-2-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | |
|---|---|---|---|
| Assignments | Internal Exam | Attendance | ESE |
| 40% | 40% | 20% | -- |

**Practical Assignment Questions**

**Hands on Practice Assignments**

**MODULE 1:**

1. Write simple Python scripts for tasks like string manipulation, reading from and writing to files, basic network communication.
2. Implement basic encryption and decryption algorithms in Python Caesar cipher, AES, DES

**MODULE 2:**

1. Write a program in python to generate and verify hashes (MD5, SHA-256) for files and messages.
2. Build a simple Python Client-Server application, understanding socketS
3. Write a python script to capture and analyze network packets(using libraries like Scapy or PySpark

**MODULE 3:**

1. Create a web scraper in Python to gather data from websites(using BeautifulSoup, Selenium)
2. Perform a Simple penetration testing task using Python (Eg: port scanning, vulnerability scanning with tools like Nmap in Python.

**MODULE 4:**

1. Use python to interact with security-related APIs (eg. VirusTotal, Shodan)
2. Write python scripts for basic static malware analysis (file signature analysis, string extraction).

**MODULE 5:**

1. Use Google and Whois for Reconnaisasance.
2. Use Nmap scanner to perform port scanning of various forms –ACK, SYN, FIN, NULL, XMAS.

**MODULE 6:**

1. Use WireShark sniffer to capture network traffic and analyze.
2. Create a simple keylogger using python
3. Use Metasploit to exploit

| COURSE TITLE | | E commerce & Digital Marketing | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA305 | COURSE CATEGORY | ME 6.2 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10% | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course provides an in-depth understanding of e-commerce fundamentals, digital marketing strategies, and the role of emerging technologies in online business. Students will explore various business models, payment systems, SEO, social media marketing, and analytics to enhance digital presence and business growth. |
| **Course Objective** | The objectives of this course are to:<br>• Provide a foundational understanding of e-commerce concepts, platforms, and business models.<br>• Introduce digital marketing techniques, including SEO, SEM, and content marketing.<br>• Explore social media marketing strategies and paid advertising campaigns.<br>• Discuss the significance of analytics, customer engagement, and conversion optimization.<br>• Understand cybersecurity measures, legal aspects, and ethical considerations in e-commerce. |
| **Course Outcome** | **CO 1** Explain fundamental concepts of e-commerce and digital marketing.<br>**CO 2** Analyze different e-commerce business models and online payment systems.<br>**CO 3** Apply digital marketing techniques such as SEO, SEM, and social media strategies.<br>**CO 4** Utilize analytics tools for performance tracking and conversion rate optimization.<br>**CO 5** Understand legal, security, and ethical concerns in e-commerce. |
| **Prerequisites:** Basic knowledge of the internet, web technologies, fundamental business concepts, and familiarity with social media and online platforms. | |

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to E-Commerce** | | | |

| | | | |
|---|---|---|---|
| 1.1 Definition, Evolution, and Benefits, 1.2 Difference Between Traditional & Digital Commerce, 1.3 E-Commerce Platforms (Shopify, WooCommerce, Magento), 1.4 Components of E-Commerce, 1.5 Current Trends in E-Commerce | 8 | 15% | **CO-1 & CO-2** |
| **MODULE 2: E-Commerce Business Models & Payment Systems** | | | |
| 2.1 B2B, B2C, C2C, C2B Models, 2.2 Dropshipping & Direct-to-Consumer Models, 2.3 Online Payment Gateways (PayPal, Stripe, Razorpay), 2.4 Mobile Wallets & Cryptocurrencies, 2.5 Security Concerns in Online Transactions | 8 | 15% | **CO-2** |
| **MODULE 3: Fundamentals of Digital Marketing** | | | |
| 3.1 Overview of Digital Marketing & Trends, 3.2 Search Engine Optimization (SEO) – On-page & Off-page, 3.3 Search Engine Marketing (SEM) – Google Ads, PPC, 3.4 Email Marketing & Automation, 3.5 Influencer Marketing | 10 | 20% | **CO-3** |
| **MODULE 4: Social media & Content Marketing** | | | |
| 4.1 Social Media Strategies (Facebook, Instagram, LinkedIn), 4.2 Video Marketing (YouTube, TikTok, Reels), 4.3 Blogging & Content Strategy, 4.4 Affiliate Marketing & Sponsored Content, 4.5 Case Studies of Successful Digital Marketing Campaigns | 10 | 20% | **CO-3** |
| **MODULE 5: Analytics & Conversion Optimization** | | | |
| 5.1 Website Traffic Analysis (Google Analytics), 5.2 Conversion Funnel & User Behavior, 5.3 A/B Testing & Landing Page Optimization, 5.4 Customer Retargeting & Lead Nurturing | 5 | 15% | **CO-4** |
| **MODULE 6: Legal, Security & Ethical Aspects** | | | |
| 6.1 Data Privacy & GDPR Compliance, 6.2 Cybersecurity Threats in E-Commerce, 6.3 Ethical Issues in Digital Marketing, 6.4 Consumer Rights & Online Fraud Prevention, 6.5 Role of AI & Machine Learning in Security | 4 | 15% | **CO-5** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | **E-Commerce: Business, Technology, Society** – Kenneth C. Laudon & Carol Traver |
| 2. | **Digital Marketing for Dummies** – Ryan Deiss & Russ Henneberry |
| **REFERENCE BOOKS** | |
| 1. | **Marketing 4.0: Moving from Traditional to Digital** – Philip Kotler |
| 2. | **The Art of SEO** – Eric Enge, Stephan Spencer, & Jessie Stricchiola |
| 3. | **Google Analytics Demystified** – Joel Davis |
| **E BOOKS** | |
| 1. | **E-Commerce Concepts & Technologies** – https://www.researchgate.net |
| 2 | **The Beginner's Guide to SEO** – https://moz.com/beginners-guide-to-seo |
| 3 | **The Digital Marketing Playbook** – https://openlibrary.org |
| **MOOC** | |
| 1. | **Google Digital Garage - Fundamentals of Digital Marketing** (Free Certification) – https://learndigital.withgoogle.com |
| 2. | **Coursera – E-Commerce & Digital Marketing Specialization** – https://www.coursera.org |

| COURSE TITLE | | RECENT TECHNOLOGIES (PROMPT ENGINEERING, RPA & GENERATIVE AI) | | | CREDITS | 3 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA305 | COURSE CATEGORY | MC6.3 | L-P-T | 3-0-0 |
| Version | 1.0 | Approval Details | 18-04-2025 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| First Periodical Assessment | Second Periodical Assessment | Seminar/ Assignments/ Project | Surprise Test / Quiz | Attendance | ESE |
| 10% | 10% | 10% | 10 % | 10% | 50% |

| | |
|---|---|
| **Course Description** | This course introduces students to cutting-edge technologies such as Prompt Engineering, Robotic Process Automation (RPA), and Generative AI. The curriculum covers theoretical concepts and practical applications to equip students with skills for modern AI-driven automation. The course also emphasizes ethical considerations and industry use cases to provide a comprehensive understanding of these emerging technologies. |
| **Course Objective** | By the end of this course, students will be able to:<br>1. Understand the fundamentals of Prompt Engineering, RPA, and Generative AI.<br>2. Develop and optimize AI-generated prompts for automation and content generation.<br>3. Implement RPA solutions for workflow automation.<br>4. Utilize Generative AI for creative and business applications.<br>5. Evaluate ethical considerations and industry applications of these technologies. |
| **Course Outcome** | CO 1 Understand Prompt Engineering, RPA, and Generative AI concepts.<br>CO 2 Apply Prompt Engineering techniques for AI interactions.<br>CO 3 Develop basic RPA workflows and automation solutions.<br>CO 4 Implement Generative AI models for content creation and problem-solving.<br>CO 5 Analyze ethical concerns and real-world applications of these technologies. |
| **Prerequisites:** Basic programming, AI fundamentals, automation basics, data handling, problem-solving, cloud & web technologies (optional). | |

| MODULE: Topic | Sessions | Weightage (%) | CO Mapping |
|---|---|---|---|
| **MODULE 1: Introduction to Recent Technologies** | | | |
| 1.1 Overview of AI, RPA, and Generative AI, 1.2 Evolution of automation and AI-driven solutions, 1.3 Importance and scope of emerging technologies | 6 | 10% | **CO-1** |

| | | | |
|---|---|---|---|
| **MODULE 2: Fundamentals of Prompt Engineering** | | | |
| 2.1 Concept and importance of Prompt Engineering, 2.2 Types of prompts and their effectiveness, 2.3 Optimizing prompts for different AI models (e.g., GPT, LLMs), 2.4 Hands-on exercises on AI prompt tuning | 8 | 20% | **CO-1 & CO-2** |
| **MODULE 3: Robotic Process Automation (RPA)** | | | |
| 3.1 Introduction to RPA and its significance, 3.2 Tools and platforms for RPA (UiPath, Blue Prism, Automation Anywhere), 3.3 Creating simple RPA bots, 3.4 Case studies on industry applications | 8 | 20% | **CO-1 & CO-3** |
| **MODULE 4: Introduction to Generative AI** | | | |
| 4.1 Basics of Generative AI and its evolution, 4.2 Working of GANs, Transformers, and LLMs, 4.3 Applications in text, image, and code generation, 4.4 Hands-on experiment using OpenAI, Midjourney, or other models | 8 | 20% | **CO-1 & CO-4** |
| **MODULE 5: Applications and Case Studies** | | | |
| 5.1 Real-world applications in business and creative industries, 5.2 Generative AI in marketing, design, and automation, 5.3 RPA use cases in banking, healthcare, and logistics, 5.4 Industry-specific prompt engineering examples | 9 | 20% | **CO-2 to CO-5** |
| **MODULE 6: Ethical Considerations & Future Trends** | | | |
| 6.1 Ethical concerns in AI and automation, 6.2 Bias, fairness, and security challenges, 6.3 Future trends in AI, RPA, and automation technologies, 6.4 Responsible AI development and governance | 6 | 10% | **CO-5** |

| | |
|---|---|
| **TEXT BOOKS** | |
| 1. | **"Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow"** – Aurélien Géron |
| 2. | **"Artificial Intelligence: A Guide for Thinking Humans"** – Melanie Mitchel |
| 3. | **"Introduction to Robotic Process Automation"** – Alok Mani Tripathi |
| **REFERENCE BOOKS** | |
| 1. | **"Generative Deep Learning"** – David Foster |
| 2. | **"Robotic Process Automation: Guide to Building Software Robots"** – Tom Taulli |
| 3. | **"The Age of AI: And Our Human Future"** – Henry Kissinger, Eric Schmidt |
| **E BOOKS** | |
| 1. | **"Artificial Intelligence: Foundations of Computational Agents"** – David Poole, Alan Mackworth https://artint.info/2e/html/ArtInt2e.html |
| 2 | **"Deep Learning with Python"** – François Chollet https://github.com/fchollet/deep-learning-with-python-notebooks |
| 3. | **"AI & Machine Learning for Coders"** – Laurence Moroney https://github.com/lmoroney/mlday |
| **MOOC** | |
| 1. | **Coursera - Prompt Engineering for ChatGPT** https://www.coursera.org/learn/prompt-engineering |

| | |
|---|---|
| 2. | **Udacity - Introduction to Generative AI**<br>https://www.udacity.com/course/introduction-to-generative-ai--ud304 |

| COURSE TITLE | Research Project | | | CREDITS | 6 |
|---|---|---|---|---|---|
| COURSE CODE | MCA303 | COURSE CATEGORY | RP | L- P- T | 0-12-0 |
| Version 1.0 | Approval Details | 07-2024 | | | |

**ASSESSMENT SCHEME**

| First Periodical Review | Second Periodical Review | Third Periodical Review | Research Paper Publication Review | Final Research Report | ESE |
|---|---|---|---|---|---|
| 10% | 20% | 20% | 20% | 30% | - |

**Prerequisites:**

Understanding of Research Methodology, Writing and drafting reports in MS-WORD/ LaTeX

**Course Objectives:**

- To Gain a comprehensive understanding of research, its process, and different types.
- To develop the ability to identify and define research problems effectively.
- To Explore various research strategies and apply appropriate methodologies to solve Research problems.

**Course Outcomes:**

On completion of the course, learners should be able to

| CO# | Cognitive Domain | Course Outcomes |
|---|---|---|
| CO1 | Understand | Demonstrate a clear understanding of research concepts, processes, and methodologies, including literature review and research proposal development. |
| CO2 | Analyze | Compare and contrast quantitative and qualitative research approaches, identify a research interest area, and apply suitable research design. |
| CO3 | Apply | Develop strong academic writing and presentation skills for effectively communicating research findings |

**Guidelines :**

- The research project duration should be a minimum of 3 to 4 months.
- The project can be undertaken individually or in groups (maximum 3 students) based on the scope and complexity of the research.
- The group should have maximum 3 students depending on level or size of the research project.
- The project should be working research which falls under one or more of the following research categories: Fundamental, Applied, Exploratory, Descriptive, Qualitative, Quantitative, Empirical, Theoretical, Survey-based, or Design & Creation.
- Data collection can be primary and/or secondary, with sources including Kaggle,

GitHub, Google Dataset Search, IEEE DataPort, and government data repositories. Data should be latest and updated, preferably post-2020.

- Students/groups must present or publish their research in reputed journals or conferences indexed in Scopus, Web of Science, Google Scholar, or other recognized platforms, such as National/International Conferences, Proceedings, Double-Blind Reviewed Journals, ISSN/ISBN-numbered publications, etc.
- The literature survey must include references from at least 25 research publications from reputed sources. The total references cited should exceed 50.
- For final evaluation, students/groups must submit a detailed research report similar to a thesis, including:
    - Introduction to the Proposal
    - Theoretical Background
    - Literature Survey
    - Proposed Work
    - Results & Analysis
    - Discussion & Comparison with Previous Work
    - References
- The submitted research project documentation/report should follow the UGC/AICTE rules and regulations about the plagiarism.
- The plagiarism check should follow UGC/AICTE guidelines, and should be conducted using platforms like Turnitin, Drillbit, or similar software. The plagiarism limit is ≤10% for text-based research and ≤15% for coding-based research.
- The research project report format should follow these guidelines:
    - **Font**: Times New Roman
    - **Font Size**: 12 pt (body text), **14 pt (titles & headings)**
    - **Spacing**: 1.5 line spacing
    - **Justified text alignment**
    - **Figure/Table names** should be in *Italics*
- References & Citations must follow APA style.
- Institute will organize research conferences on a rotational basis to accommodate student presentations and discussions.

**Evaluation Parameters :**

- Originality of Proposed Work – 10%
- literature survey – 10%
- Proposed Work – 10%
- Results Obtained – 10%
- Detailed Report – 30%
- Work Presented/Published By Student/Group Of Students – 20%
- Final Presentation. – 10%

**Important Links:**

For plagiarism check - https://www.turnitin.com/login_page.asp?lang=en_us
https://www.drillbitplagiarism.com/

**Tools :**

For coding - SPSS, R, Python, MATLAB, SAS

For report writing - LaTex, MS-Word

For Citation Management - Zotero/Mendeley

Coding - Jupyter Notebook or similar IDEs

**Reference Books :**


1. **Research Methodology: A Handbook of Methods and Techniques** by S. R. K. Sharma (2011), Sage Publications India.
2. **Research Methodology in Social Sciences** by K. R. Sharma (2013), Ramesh Book Depot.
3. **Statistical Methods for Research** by S. P. Gupta (2017), Sultan Chand & Sons.
4. **Research Methodology: An Introduction** by R. P. Srivastava (2012), Kitab Mahal.
5. **Fundamentals of Research Methodology in Social Sciences** by K. L. Sharma (2015), Atlantic Publishers.
6. **Practical Research: Planning and Design** by P. D. Leedy & J. E. Ormrod (2019), Pearson Education.
7. **The SAGE Handbook of Qualitative Research** by N. K. Denzin & Y. S. Lincoln (2017), SAGE Publications.
8. **Research Methods in Education** by L. Cohen, L. Manion, & K. Morrison (2018), Routledge.
9. **Methods in Social Research** by W. J. Goode & P. K. Hatt (1952), McGraw-Hill.
10. **Action Research: A Guide for the Teacher Researcher** by G. E. Mills (2017), Pearson Education.

**Date:**

# CERTIFICATE

**This is to certify that Mr/Ms. ……..……., has successfully / Partially completely his research project work entitled "……...…….” in partial fulfilment of MCA – II SEM –III MCA306RP Research Project for the year 2025-2026. He/She has worked under our guidance and direction.**

**<Project Guide Name>**
**Project Guide**

**<Director Name>**
**Director,**

**Date:**
**Place**

**A**

**RESEARCH PROJECT REPORT**

**ON**

**TITLE OF RESEARCH PROJECT**

**IN PARTIAL FULFILLMENT OF**

**MASTER OF COMPUTER APPLICATION**

**BY**

**NAME OF STUDENT**

**MCA –II SEM – III**

**(2024-2025)**

**UNDER THE**

**GUIDANCE OF**

**FAUCLTY NAME**

**SUBMITTED TO**

**SAVITRIBAI PHULE PUNE UNIVERSITY**

**<NAME OF THE COLLEGE>**

Cover Page

Completion certificate of Institute

Presenters Certificate

Publishers Certificate

Self-declaration

Originality Report/Plagiarism Report

Acknowledgement

INDEX

| COURSE TITLE | | INDUSTRIAL INTERNSHIP / ON JOB TRAINING | | | CREDITS | 12 |
|---|---|---|---|---|---|---|
| COURSE CODE | | MCA404 | COURSE CATEGORY | OJT | L-P-T | 0-24-0 |
| Version | 1.0 | Approval Details | 07-2024 | | | |

| ASSESSMENT SCHEME | | | | | |
|---|---|---|---|---|---|
| Review 1 | Review 2 | Review 3 | Final review | Attendance | ESE |
| 10% | 10% | 10% | 10% | 15% | 45% |

The project is an outcome of technical skills and domain knowledge acquired by the students during the program. Students demonstrate problem solving skills, analytical ability, logical thinking, communication skills and team work during the course of the project. The project can be implementation of a research work published in any reputed journal.

The project work shall be based on the knowledge acquired by the student during the course and has to be done compulsorily as industrial internship. The project aims to provide an opportunity of designing and building complete system or subsystems based on area where the student likes to acquire specialized skills.

Project work is to be carried out either individually or in a group. Each group or individual will be assigned a guide. At the end of the term the student should submit at least two copies of the project report in a prescribed format. Examination will be carried out by a pair of examiners for each group with one internal and one external examiner appointed.

Evaluation Details:
- Individual student, one project per student, must carry out major project.
- Student must submit a Detailed Project Report (60 to 80 Pages) in a format as specified by the department.
- Internal guides will evaluate the performance (Continuous Internal Evaluation) for 250 Marks.
- The Report will be evaluated for 450 marks by both internal and external evaluators.
- Internal and external examiners will evaluate final viva-voce for 200 marks which includes demonstration and presentation of project work jointly.

1. The project may be done individually or in groups. However, if project is done in groups, each student must be given a responsibility for distinct modules.
2. Selected project/module must have relevant scope as per the marks assigned and can be carried out in the Institute or outside with prior permission of the Institute.

3. Internal guide should monitor and evaluate the progress of the project on individual basis through handwritten workbook maintained by students containing various project milestones with learnings and remarks from internal guide for concurrent evaluation.
4. The Semester IV project should be having sufficient scope for 400 marks. The project work will carry 300 marks for internal assessment and 250 marks for external assessment.
5. Students are expected to show working demo of the project during final evaluation in semester IV.
6. The project report should be prepared as per the University prescribed format with all the chapters mentioned in project guidelines. And it should be printed on back-to-back pages (one copy) which should be signed by the internal guide and the Director of the Institute. A client (colleges, Non IT organization, and IT organization) certificate should be attached to prove the authenticity of the project work done.
7. The project will be assessed internally as well as externally by the examiners appointed by the institutions and University.

**Type of Projects**

**1. Application Development**

The students are advised to choose a project that involves window-based development, web-based development, mobile-based development, projects based on machine learning. Analysis and interpretation of any company specific data is not permitted.

**2. Embedded Systems / IoT**

A project should be developed and implemented for application specific system after thorough investigation of the latest development in the field of electronics or communication to facilitate their efficient operation. The Real Time Operating System (RTOS) or open source platform can be used to develop embedded applications such as Robotics, Microcontroller / Microprocessor based projects etc. An IOT project can be used to design products for reliability and security using simple electronics concepts and integrating with a cloud platform to get the data real-time and make some operational analysis. It has to use efficient algorithms for strong authentication and security protocols and disable non-essential services.

Few examples of IoT applications
Smart home, Health care applications, Smart waste management, Activity Tracker etc.

**3. ETL Projects**

Extract, transform, load (ETL) is the process of integrating the data from one or more sources. It is expected from the student that he should demonstrate the entire ETL process with reference to any domain like finance, banking, insurance, retail etc.
Data extraction consists of extracting the data from homogeneous or heterogeneous sources and transforming it into a proper format using data cleansing. The data can be finally loaded into a final target database such as operational data base, a data mart or data warehouse. This data can be further used for the purpose of querying and analyzing.

## 4. Research Projects

The research project will be able to demonstrate the skills of working scientifically, and through the project the students will able to understand how to do a literature review, and how to appraise the literature to address questions. To explore an area of interest (develop some expertise and a deeper understanding of a topic). Understand the tools to critically and thoughtfully appraise problems which are faced every day; to learn communicate scientific research in verbal presentations and written form. As an example, the students can identify any problem, by observation or through survey to understand the problem in depth and propose the solution by applying the research methodology.

**Project Guidelines:**

## 1. Application Development Project

| Chapter No | | Details |
|---|---|---|
| **1** | | **Introduction** |
| | 1.1 | Company Profile / Institute Profile / Client Profile |
| | 1.2 | Abstract |
| | 1.3 | Existing System and Need for System |
| | 1.4 | Scope of System |
| | 1.5 | Operating Environment - Hardware and Software |
| | 1.6 | Brief Description of Technology Used<br>1.6.1 Operating systems used (Windows or Unix)<br>1.6.2 RDBMS/No Sql used to build database (mysql/ oracle, Teradata, etc.) |
| **2** | | **Proposed System** |
| | 2.1 | Study of Similar Systems ( If required research paper can be included) |
| | 2.2 | Feasibility Study |
| | 2.3 | Objectives of Proposed System |
| | 2.4 | Users of System |
| **3** | | **Analysis and Design** |
| | 3.1 | System Requirements (Functional and Non-Functional requirements) |
| | 3.2 | Entity Relationship Diagram (ERD) |
| | 3.3 | Table Structure |
| | 3.4 | Use Case Diagrams |
| | 3.5 | Class Diagram |
| | 3.6 | Activity Diagram |
| | 3.7 | Deployment Diagram |
| | 3.8 | Module Hierarchy Diagram |
| | 3.9 | Sample Input and Output Screens (Screens must have valid data. All<br>reports must have at-least 5 valid records.) |
| **4** | | **Coding** |
| | 4.1 | Algorithms |
| | 4.2 | Code snippets |
| **5** | | **Testing** |
| | 5.1 | Test Strategy |
| | 5.2 | Unit Test Plan |
| | 5.3 | Acceptance Test Plan |
| | 5.4 | Test Case / Test Script |

| | | |
|---|---|---|
| | 5.5 | Defect report / Test Log |
| 6 | | **Limitations of Proposed System** |
| **7** | | **Proposed Enhancements** |
| **8** | | **Conclusion** |

| | | |
|---|---|---|
| **9** | | **Bibliography** |
| **10** | | **Publication / Competition certificates** |
| **11** | | **Appendix – Cost sheet , Data sheet** |
| **12** | | **User Manual** (All screens with proper description/purpose Details<br>about validations related to data to be entered.) |

**2. Embedded Systems / IoT Project**

| Chapter No | | Details |
|---|---|---|
| **1** | | **Introduction** |
| | 1.1 | Company Profile / Institute Profile / Client Profile |
| | 1.2 | Abstract |
| | 1.3 | Existing System and Need for System |
| | 1.4 | Scope of System |
| | 1.5 | Operating Environment - Hardware and Software |
| | 1.6 | Brief Description of Technology Used<br>1.6.1 Operating systems used (Windows or Unix)<br>1.6.2 Database (if applicable) |
| **2** | | **Proposed System** |
| | 2.1 | Study of Similar Systems ( If required research paper can be included) |
| | 2.2 | Feasibility Study |
| | 2.3 | Objectives of Proposed System |
| | 2.4 | Users of System |
| **3** | | **Analysis and Design** |
| | 3.1 | Technical requirements – H/W , S/W |
| | 3.2 | System Architecture / Block Diagram |
| | 3.3 | System Hardware Details |
| | 3.4 | Pin Diagrams |
| | 3.5 | Interface diagrams |
| | 3.6 | Design Sequence |
| | 3.7 | System Software Details |
| | 3.8 | Process / System Flow chart |
| **4** | | **Coding** |
| | 4.1 | Algorithms |
| | 4.2 | Code snippets (if applicable) |

| | | |
|---|---|---|
| **5** | | **Testing** |
| | 5.1 | Results & reports |
| | 5.2 | Test cases |
| | 5.3 | Acceptance Testing |
| | 5.4 | Test reports in IEEE format |
| **6** | | **Limitations of Proposed System** |
| **7** | | **Proposed Enhancements** |

| | | |
|---|---|---|
| **8** | | **Conclusion** |
| **9** | | **Bibliography** |
| **10** | | **Publication / Competition certificates** |
| **11** | | **Appendix – Cost sheet , Data sheet** |
| **12** | | **User Manual** (All screens with proper description/purpose Details<br>about validations related to data to be entered.) |

**3. ETL Projects**

| Chapter No | | Details |
|---|---|---|
| **1** | | **Introduction** |
| | 1.1 | Company Profile / Institute Profile / Client Profile |
| | 1.2 | Existing System functionality (Source System for which the ANALYTICS is<br>being developed) |
| | 1.3 | Business process understanding and specifications<br>1.3.1 Business Requirement Specifications:<br>1.3.1.1 The o/p from BR Analysis are BRS Business Requirement Specifications (Business specific Rules to be mentioned here from analysis point of view)<br>1.3.1.2 Identify the dimensions, required attributes, measures, filter conditions, adjustments for KPIs going to be used in the Target system and its availability in the Source System. If any gaps suggest remediation of gaps<br>1.3.2 Business Rules Collection<br>1.3.3 Identify the Key Performance Indicator (specified by<br>1.3.4 Establish the User Acceptance Criteria<br>client) |
| | 1.4 | Scope of the project |
| | 1.5 | Operating Environment - Hardware & Software, Description of Tools / Technology to be used in the Target system<br>1.5.1.1 Operating systems used (Windows or Unix)<br>1.5.1.2 RDBMS/NoSql used to build database (mysql/ oracle, Teradata, etc.)<br>1.5.1.3 ETL tools used (Talend/Informatica, Datastage etc)<br>1.5.1.4 OLAP/ Data mining/ machine learning/ analytics tools used (Python/ Cognos, BO, etc.)<br>1.5.1.5 Data visualization tools (power BI / Tableau) |
| **2** | | **Proposed System** |
| | 2.1 | Creating multiple ETL strategies - Specifying metadata details, identifying heterogeneous architectures, processes for I/O only for ETL, scrapping , identifying the volatilities in the channels , designing<br>strategies in the context of the business and existing ERP |
| | 2.2 | Comparing them in the context of selected business system (as per the business requirements) |
| | 2.3 | Suggesting optimum solution (process) |
| **3** | | **Analysis and Design** |

| | | |
|---|---|---|
| | 3.1 | Use Case Diagram |
| | 3.2 | Activity diagram to demonstrate Process flow (execution of ETL process) |
| | 3.3 | Design of Target system (Elaborate the tiers of DW architecture in the Target System) |
| | 3.4 | Database schema / Table specifications of Target system |
| | 3.5 | Details of Source & Targets of mapping in the database |
| | 3.6 | Details of Load (Full/Incremental etc.) |
| | 3.7 | Design of ETL schema/strategy |
| **4** | **4.1** | **Design of strategy for Visualization**<br>4.1.1 Visualizations in support of comparison of performance of various ETL strategies<br>4.1.2 Data visualization using different techniques (if any) |
| **5** | | **Drawbacks and Limitations Proposed Enhancements** |
| **6** | | **Conclusion** |

## 4. Research Projects

Research projects especially are designed to gain knowledge about some specified area and the deliverable is that knowledge gained, usually encapsulated in some form of report.
Students are expected to contribute something new to academic or practical knowledge in their research area—something original that is more than the accepted knowledge.

Completing a Research Project as part of your coursework is an opportunity to:

- learn to read and interpret other people's research critically by doing your own. This gives you an insight into the effects of practical difficulties and theoretical debates on published research
- develop and apply the knowledge that you have learnt in 4 semesters of your curriculum.
- submit a paper for peer-reviewed publication. (If successful, this will give a boost to your c.v.) If you wish to enroll in a research degree such as PhD, a research project as part of your coursework will assist the committee evaluating your application in assessing whether you are ready to do independent research.

**Research Index**

1. Title page

2. Acknowledgements

You should acknowledge the assistance given to you by your supervisors, and any other person or organization that has helped you in the planning, conduct, analysis or reporting of your project.

3. Abstract

This is a synopsis of your study question, aims and objectives, background literature, methods, results, key conclusions and recommendations. This should be 250–300 words long and should be very clear and easy to follow.

4. Introduction

In this section of your report you introduce the subject, provide the background to the topic or problem, outline the study question (or problem or study hypothesis), and outline the aims and objectives of your study.

## 5. Literature review

This is a review of the literature on the topic or problem you are studying. It should include a review of any other studies or projects similar or relevant to yours, and perhaps a review of the literature on the method you have chosen if your project tests a new method of research or analysis.

## 6. Methods

This section includes the methodology of your research. It will cover such issues as: In case of Computer Management Research :

- Study design
- Study population, sampling frame and numbers, sampling method
- survey design
- survey or data collection instruments
- protocol for obtaining data
- ethical issues and how they are addressed
- information letters, consent forms
- data management and analysis methods
- statistical analysis and tests
- In case of Computer Science Research:
- Study design
- System Architecture
- Implementation
    - Experimental Implementation
    - Simulation
- Data management and analysis methods
- Analysis and testing

## 7. Results

In this section you present the results of your research. Tables, figures and graphs are an excellent means of presenting this sort of information. All tables, figures and graphs, should be numbered consecutively throughout the whole report, and labelled with a clear and concise descriptive title.

## 8. Discussion

In this section you interpret your results and discuss their implications, with reference to other published research. Any limitations in your research methodology should also be referred to here. Examiners expect you to acknowledge these limitations as an integral part of your evaluation of your project.

## 9. Conclusion

This section summarizes the key results and the conclusions that you can draw from these results. It also needs to reflect what your initial project aims and objectives were.

## 10. Recommendations

It is good research practice to make recommendations or to suggest directions for further research or actions as a result of your project findings.

## 11. References

This is a list of all the references and sources you used in your literature review, methodology and discussion. This includes books, journal articles, abstracts, conference and symposium papers, media articles, and any form of published literature or comment.

## 12. Appendices

This section may contain copies of any questionnaires if any or evaluation instruments used covering letters, participant information and ethics approvals, or additional explanations.

# MOOC Guideline For MCA

**Preamble:**

The Master of Computer Applications (MCA) program is designed to empower students with advanced knowledge and expertise in cutting-edge fields such as Artificial Intelligence, Machine Learning, Cloud Computing, Cybersecurity, Full Stack Development, and Data Science. To complement classroom learning and provide exposure to global industry trends, the program incorporates Massive Open Online Courses (MOOCs) as a mandatory component in Semester 4.

Objective:

- To gain proficiency in emerging technologies and stay updated with industry advancements.
- To develop specialized skills in a chosen domain of interest.
- To strengthen problem-solving abilities and practical implementation expertise.
- To enhance employability through globally recognized certifications.

Introduction:
As a prerequisite for the award of the MCA degree, students are required to complete MOOC certifications. This initiative aims to cultivate a culture of lifelong learning, encourage specialization in high-demand technologies, and boost the employability of graduates. By leveraging renowned online learning platforms, students will acquire in-depth knowledge and hands-on skills in areas such as Data Science, Cloud Computing, Cybersecurity, Full Stack Development, and other relevant domains, ensuring they are well-prepared for the dynamic IT industry.

**Comprehensive MOOC Certificate Guidelines:**

1. **Mandatory Completion:**

   Students are required to successfully complete one MOOC course from the choices from the each designated as "MOOC 1" and "MOOC 2, MOOC 3" each carrying 3 and 2 credits respectively, to be eligible for the award of the MCA degree.

2. **Course Duration:**

Only MOOC courses with a duration of 12 weeks or more will be considered for the award of 3 credits. Students must register exclusively for courses meeting this duration requirement.

3. **Registration Timeline:**

   Students can begin registering for MOOC courses from the commencement of Semester 4.

4. **Submission Deadline:**

   Completed MOOC certificates must be submitted by the end of Semester 4. Failure to submit the required MOOC certificates by this deadline will result in the student not being entitled to receive the MCA degree.

5. **Platform Reputability:**

   Certificates from reputable platforms like NPTEL, SWAYAM, Coursera, edX, and Udacity are preferred due to their industry recognition.

6. **Curriculum Relevance:**

   MOOC courses should directly complement or expand upon the MCA curriculum, enhancing core knowledge and practical skills.

7. **Specialization Alignment:**

   Students should prioritize MOOCs that align with their chosen specialization (e.g., Data Science, Cloud Computing, Cybersecurity, Full Stack).

8. **Practical Skill Focus:**

   MOOCs focusing on hands-on projects, coding assignments, and real-world case studies are highly recommended.

9. **Advanced Topics and Latest Technologies:**

   Encourage students to pursue MOOCs covering advanced topics and the latest technologies not extensively covered in the core curriculum.

10. **Certification Verification:**

    Students are responsible for ensuring the authenticity of their submitted MOOC certificates.

11. **Institutional Approval:**

    The institution reserves the right to approve or reject MOOC courses that do not align with the academic framework.

12. **Faculty Consultation:**

    Students are encouraged to consult faculty advisors before enrolling in MOOC courses.

13. **Certification Discrepancies:**

    Any discrepancies in certification will be subject to review by the academic committee.

14. **Internship/Project Alignment:**

    Students may choose MOOC courses in the field or technology related to their internship or project activities, provided they meet all other criteria.

| No. | MOOC-1 (3 Credits) | Platform | MOOC-2 (3 Credits) | Platform | MOOC-3 (2 Credits) | Platform |
|---|---|---|---|---|---|---|
| 1 | IBM AI Engineering Professional Certificate | Coursera | Certified Kubernetes Security Specialist (CKS) | Linux Foundation | AWS Certified Developer - Associate | AWS Training |
| 2 | Google Cloud Security Engineer | Google Cloud | Certified Blockchain Developer (Hyperledger) | Linux Foundation | Microsoft Certified: Azure Security Engineer Associate | Microsoft |
| 3 | UiPath RPA Developer Foundation | UiPath Academy | Automation Anywhere RPA Certification | Automation Anywhere University | Blue Prism RPA Developer Certification | Blue Prism University |
| 4 | Microsoft Power Platform Developer Associate (PL-400) | Microsoft | IBM Data Science Professional Certificate | Coursera | Salesforce Administrator Certification | Salesforce |
| 5 | JIRA Software Fundamentals | Atlassian University | Agile with JIRA | Coursera (Atlassian) | JIRA Administration | Atlassian |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | | Certification | |
| 6 | Certified DevSecOps Professional (CDP) | Practical DevSecOps | AWS Certified Solutions Architect - Associate | AWS Training | Google Associate Cloud Engineer | Google Cloud |
| 7 | Software Testing with Selenium & JUnit | Udemy | ISTQB Certified Tester (CTFL) | ISTQB | Appium Mobile Automation Testing | Udemy |
| 8 | AWS Certified Advanced Networking - Specialty | AWS Training | Microsoft Certified: DevOps Engineer Expert (AZ-400) | Microsoft | Certified Ethical Hacker (CEH) | EC-Council |
| 9 | LoadRunner Performance Testing | Micro Focus | JMeter Performance Testing | Udemy | Postman API Testing | Postman Academy |
| 10 | AI for IT Operations (AIOps) Certification | Coursera (IBM) | DevOps with GitHub Actions & Docker | Udemy | Python for Automation Testing | Udemy |