DATABASE MANAGEMENT SYSTEM			
Course Code	**1MMC102**	Scheme	2026
Type of Course	**PCC**	Semester	I
Teaching Hours/Week (L:T:P)	3:0:0	CIE Marks	50
Total Hours of Pedagogy L /T/P/SL&TW	42:0:0:48	SEE Marks	50
Credits	03	Total Marks	100
Examination type (SEE)	Theory	Exam Hours	03

Course Outcomes

At the end of the course, the student will be able to:

CO1: Apply the basic concepts of Database Management
CO2: Write SQL, PL/SQL queries for a given problem scenario
CO3: Apply NoSQL data modeling concepts
CO4: Perform CRUD operations using MongoDB.
CO5: Build a database for any given use-case.

Module-1

Introduction to DBMS: Characteristics of database approach, Advantages and applications of DBMS, Three-schema architecture and data independence

Entity Relationship Model: Entity types and entity sets, Attributes and keys, Relationship types and constraints, Weak entities, ER diagrams for real-world applications, ER-to-relational mapping.

TextBook-1: Chapter 1, 2, 3, 9 Number of Hours: 09

Module-2

SQL Fundamentals: SQL data types and constraints, DDL, DML, DCL, TCL commands, Simple and nested queries, Joins and set operations, Aggregate functions and grouping, Views.

Database Design: Functional dependencies, Normalization (1NF, 2NF, 3NF, BCNF)

Transactions Processing Concepts: Transaction concepts, ACID properties.

TextBook-1: Chapter 4, 5, 6, 15, 20 Number of Hours: 08

Module-3

Advanced SQL and PL/SQL: Stored procedures, functions and triggers, Cursors and exception handling

NoSQL: Introduction to NoSQL and need for NoSQL databases, Characteristics of NoSQL systems, Types of NoSQL databases – Key-value stores, Document databases, Column-family databases and Graph databases, Storage architecture basics.

TextBook-1: Chapter 6, **TextBook-2:** Chapter 1, 2, 3 Number of Hours: 08

Module-4

MongoDB: Introduction to MongoDB, CRUD Operations, Querying, Aggregation, Indexing, Replication, Sharding.

TextBook-3: Chapter 1, 2, 3, 4, 5, 6, 8, 9 Number of Hours: 08

Module-5

Database Security and Recovery: Introduction to Database Security, Discretionary Access Control, Mandatory Access Control and Role-Based Access Control, SQL Injection, Database Backup and Recovery Concepts, Types of Database Failures, Recovery Techniques based on Deferred and Immediate Update, Shadow Paging.

TextBook-1: Chapter 21, 23 Number of Hours: 09

Textbooks:

1. Ramez Elmasri, Shamkant B. Navathe, *Fundamentals of Database Systems*, Pearson, 7th Edition, 2016
2. Shashank Tiwari, *Professional NoSQL*, Wiley India/Wrox Publication, 1st Edition, 2011.
3. Kristina Chodorow, *MongoDB: The Definitive Guide*, O'Reilly Media, 1st Edition.

Reference books / Manuals:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts*, McGraw Hill, 7th Edition.
2. Raghu Ramakrishnan and Johannes Gehrke, "*Database Management Systems*", 3rd Edition, McGraw Hill, 2018.
3. Pramod J. Sadalage and Martin Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Addison-Wesley Professional.

Web links and Video Lectures (e-Resources):

- MySQL Documentation: https://dev.mysql.com
- https://w3schools.com/sql
- MongoDB Documentation: https://www.mongodb.com/resources/products/fundamentals/mongodb-tutorials
- Interactive SQL: https://sqlbolt.com/

Assessment Structure (both CIE and SEE)

The weightage assigned to Continuous Internal Evaluation (CIE) and Semester End Examination (SEE) shall be **50% each**. A student shall secure a minimum of 50% of the maximum marks prescribed for CIE and a minimum of **40%** of the maximum marks prescribed for SEE to be eligible for passing.

A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to a subject/course only if he/she secures not less than **50% of the aggregate marks** obtained in CIE and SEE taken together.

Continuous Internal Evaluation:

Continuous Comprehensive Assessments (CCA)+ Internal Assessment Test (IA) = Continuous Internal Evaluation CIE [25+25=50 marks]

A minimum of **two Internal Assessment Tests (IA)** shall be conducted. Each test shall be conducted for a maximum of **25 marks**. The average of the two test marks shall be the marks secured by a student in IA.

In addition, **Continuous Comprehensive Assessment (CCA)** shall be conducted for a total of **25 marks**.

Introduction to AI and Applications			
Course Code	**1MMC103**	Scheme	2026
Type of Course	PCC	Semester	I
Teaching Hours/Week (L:T:P)	3:0:0	CIE Marks	50
Total Hours of Pedagogy L /T/P/SL&TW	42: 0 : 0 : 48	SEE Marks	50
Credits	3	Total Marks	100
Examination type (SEE)	Theory	Exam Hours	3

Course outcome (Course Skill Set)

At the end of the course, the student will be able to:

CO1: Explain the fundamental concepts of artificial intelligence, intelligent agents, agent architectures, and the characteristics of problem-solving environments.

CO2: Apply search strategies, optimization techniques, and knowledge representation methods to solve artificial intelligence problems.

CO3: Analyse adversarial search, logical inference, and machine learning approaches for intelligent decision-making in diverse application domains.

CO4: Apply explainability, bias mitigation, fairness, ethics, responsible AI principles, and performance measures in the development and assessment of artificial intelligence solutions.

CO5: Analyse and explain the use of Artificial Intelligence techniques in solving real-world problems.

Module-1

Introduction and Intelligent Agents: Introduction; What is Artificial Intelligence? The Foundations of Artificial Intelligence; The History of Artificial Intelligence; The State of the Art; Intelligent Agents; Agents and Environments; The Nature of Environments; The Structure of Agents.

Textbook 1: Chapters 1.1 to 1.4, 2.1 to 2.3 Number of Hours: 8

Module-2

Solving Problems by Searching: Problem-Solving Agents; Example Problems; Searching for Solutions; Uninformed Search Strategies; Informed (Heuristic) Search Strategies; Local Search Algorithms and Optimization Problems; Search in Complex Environments; Online Search Agents and Unknown Environments.

Textbook 1: Chapters 3.1 to 3.5 Number of Hours:8

Module-3

Adversarial Search and Knowledge-Based Systems: Adversarial Search and Games; Optimal Decisions in Games; Alpha–Beta Pruning; Knowledge-Based Agents; Propositional Logic; First-Order Logic; Inference in First-Order Logic.

Textbook 1: Chapters 5.1 to 5.3,7.1 to 7.2 and 9.1 to 9.3 Number of Hours:8

Module-4

Machine Learning Concepts and Responsible AI: Introduction to Machine Learning, Relationship between Artificial Intelligence, Machine Learning and Deep Learning, Types of Machine Learning (Supervised, Unsupervised and Reinforcement Learning), AI Model Development Lifecycle, Model Training, Validation and Testing, Challenges in Machine Learning, Explainable AI (XAI), Responsible AI, Bias, Fairness, Transparency, Ethics and Risks of Artificial Intelligence.

Textbook 2: Chapters 2.1 to 2.5, Chapter 8.1 to 8.5 Number of Hours: 9

,

Artificial Intelligence Applications: Introduction to Natural Language Processing and Large Language Models, Computer Vision, Robotics and Intelligent Autonomous Systems, Multi-Agent Systems, AI Applications in Healthcare, Finance, Agriculture, Smart Cities and Transportation, Responsible and Ethical AI.

Textbook 1: Chapter 23.1 to 23.4, 25.1 to 25.3, 26.1 to 26.3, 27.1 to 27.3

Textbook 2: Chapter 4.1 to 4.3, 7.1 to 7.4, 8.1 to 8.4 Number of Hours: 9

Textbooks:

1. Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig, Pearson Education, 4th Global Edition, 2021.
2. Tom Taulli, Artificial Intelligence Basics: A Non-Technical Introduction, 2nd Edition, A press, 2023.

Reference books / Manuals:

1. Artificial Intelligence: Foundations of Computational Agents, David L. Poole and Alan K. Mackworth, Cambridge University Press, 2017.
2. Deep Learning, Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016.
3. Machine Learning, Tom M. Mitchell, McGraw-Hill Education, 1997.
4. Artificial Intelligence, McGraw-Hill Education, 3rd Edition.

Web links and Video Lectures (e-Resources):

- NPTEL course Artificial Intelligence: Concepts and Techniques
  https://onlinecourses.nptel.ac.in/noc25_cs159/preview
- Course AI for Everyone (Deep earning.ai)
  AI for Everyone
- Geeks for Geeks Introduction to AI and Its Application
  https://www.geeksforgeeks.org/artificial-intelligence/artificial-intelligence/

Assessment Structure (both CIE and SEE)

The weightage assigned to Continuous Internal Evaluation (CIE) and Semester End Examination (SEE) shall be **50% each**. A student shall secure a minimum of 50% of the maximum marks prescribed for CIE and a minimum of **40%** of the maximum marks prescribed for SEE to be eligible for passing.

A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to a subject/course only if he/she secures not less than **50% of the aggregate marks** obtained in CIE and SEE taken together.

Continuous Internal Evaluation:

Continuous Comprehensive Assessments (CCA)+ Internal Assessment Test (IA) = Continuous Internal Evaluation CIE [25+25=50 marks]

A minimum of **two Internal Assessment Tests (IA)** shall be conducted. Each test shall be conducted for a maximum of **25 marks**. The average of the two test marks shall be the marks secured by a student in IA.
In addition, **Continuous Comprehensive Assessment (CCA)** shall be conducted for a total of **25 marks**.

Data Structures and Algorithms			
Course Code	**1MMC104**	Scheme	2026
Type of Course	**PCC**	Semester	I
Teaching Hours/Week (L:T:P)	3: 0 : 0	CIE Marks	50
Total Hours of Pedagogy L /T/P/SL&TW	42:0: 0:48	SEE Marks	50
Credits	**3**	Total Marks	100
Examination type (SEE)	Theory	Exam Hours	3

Course outcome (Course Skill Set)

At the end of the course, the student will be able to:

CO1: Demonstrate fundamental concepts of different data structures and basic algorithmic analysis.
CO2: Use linear structures like arrays, linked lists, stacks, and queues to manage sequential data.
CO3: Develop functions using non-linear structures like trees and graphs alongside their searching and traversal protocols.
CO4: Apply suitable data structures and algorithm techniques to effectively solve real-world computational or engineering problems.

Module-1

Foundations of Data Structures & Algorithmic Analysis:

Data Structures Introduction: Fundamental Data Structures.
Linear Array: Arrays, Dynamically Allocated Array,
Linked Memory: Singly linked lists and chains, Doubly linked lists.
Algorithms Analysis Introduction: What is an Algorithm?, Fundamentals of Algorithmic Problem Solving, The Analysis Framework, Asymptotic Notations and Basic Efficiency Classes.

Textbook1: 2.1, 2.2, 4.1, 4.8
Textbook2: 1.4, 1.1, 1.2, 2.1, 2.2 Number of Hours:8

Module-2

Stacks, Queues, and Mathematical Analysis of Algorithms:

Stacks and Queues: Stacks, Stacks Using Dynamic Arrays, Queues, Circular queues Using Dynamic Arrays, Evaluation of expression.
Mathematical Analysis: Mathematical Analysis of Non-recursive Algorithms, Mathematical Analysis of Recursive Algorithms.

Textbook1: 3.1, 3.2, 3.3, 3.4, 3.6
Textbook2: 2.3, 2.4 Number of Hours:9

Module-3

Tree Structures, Searching & Hashing:

Trees: Introduction, Binary Trees, Binary Tree Traversal, Binary Search Trees (BST).
Searching Algorithms: Sequential Search and Brute-Force String Matching, and Decrease and Conquer - Binary Search.
Hashing: Introduction and Static Hashing.

Textbook1: 5.1, 5.2, 5.3, 5.7, 8.1, 8.2
Textbook2: 3.2, 4.4 (only Binary Search) Number of Hours:9

Module-4

Graph Algorithms & Sorting Paradigms:

Graphs: The Graph Abstract Data Type.
Graph Traversals: Depth-First Search (DFS) and Breadth-First Search (BFS).
Brute-Force Sorting: Selection Sort and Bubble Sort.
Divide-and-Conquer Sorting: Merge Sort and Quick Sort.

Textbook1: 6.1
Textbook2: 3.1, 3.5, 5.1, 5.2 Number of Hours:8

Module-5

Greedy, Transform-and-Conquer, and Balanced Trees:

The Greedy Method: Minimum Spanning Trees via Prim's Algorithm. Single-Source Shortest Path via Dijkstra's Algorithm.
Transform-and-Conquer Structures: Heaps: Heaps and Heap Sort, Balanced Search Trees: AVL Trees, 2-3 Trees.

Textbook2: 6.3, 6.4, 9.1, 9.3 Number of Hours:8

Suggested Learning Resources: (Text Book/ Reference Book/ Manuals):

Text books:

1. Ellis Horowitz and Sartaj Sahni, *Fundamentals of Data Structures in C*, 2nd Edition (2023), Universities Press(India) Private limited.
2. Anany Levitin, *Introduction to the Design and Analysis of Algorithms*, 3rd Edition (2024), Pearson Education, Limited.

Reference books / Manuals:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, *Introduction to Algorithms*, 4th Edition (2022), MIT Press.
2. Seymour Lipschutz, *Data Structures (Schaum's Outlines)*, Revised 1st Edition (2014), McGraw Hill.
3. Reema Thareja, *Data Structures using C*, 3rd Edition (2023), Oxford University Press.
4. Robert Sedgewick and Kevin Wayne, *Algorithms*, 4th Edition (2011), Addison-Wesley.
5. Adnan Aziz, Tsung-Min Kuo, and Amit Prakash, *Elements of Programming Interviews in C*, 2nd Edition (2014), CreateSpace.

Web links and Video Lectures (e-Resources):

- W1**:** National Programme on Technology Enhanced Learning (NPTEL), *Programming, Data Structures and Algorithms in C*, Indian Institute of Technology Madras.
  https://nptel.ac.in/courses/106106127

- W2: National Programme on Technology Enhanced Learning (NPTEL), *Design and Analysis of Algorithms*, Indian Institute of Technology Madras.
  https://nptel.ac.in/courses/106106131

- W3: Massachusetts Institute of Technology (MIT) OpenCourseWare, *Introduction to Algorithms (6.006)*, Department of Electrical Engineering and Computer Science.
  https://ocw.mit.edu/courses/6-006-introduction-to-algorithms-spring-2020/

- W4: Visualgo Data Structure Visualizations, *Interactive animations of sorting, search trees, and graphs*.
  https://visualgo.net/en

Assessment Structure (both CIE and SEE)

The weightage assigned to Continuous Internal Evaluation (CIE) and Semester End Examination (SEE) shall be **50% each**. A student shall secure a minimum of 50% of the maximum marks prescribed for CIE and a minimum of **40%** of the maximum marks prescribed for SEE to be eligible for passing.

A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to a subject/course only if he/she secures not less than **50% of the aggregate marks** obtained in CIE and SEE taken together.

Continuous Internal Evaluation:

Continuous Comprehensive Assessments (CCA)+ Internal Assessment Test (IA) = Continuous Internal Evaluation CIE [25+25=50 marks]

A minimum of **two Internal Assessment Tests (IA)** shall be conducted. Each test shall be conducted for a maximum of **25 marks**. The average of the two test marks shall be the marks secured by a student in IA.

In addition, **Continuous Comprehensive Assessment (CCA)** shall be conducted for a total of **25 marks**.

	Operating System and Shell Scripting			
	Course Code	**1MMC105**	Scheme	2026
	Type of course	**IPCC**	Semester	I
	Teaching Hours/Week (L:T:P)	42: 0: 28	CIE Marks	50
	Total Hours of Pedagogy L/T/P/SL&TW	42:00: 28: 50	SEE Marks	50
	Credits	04	Total Marks	100
	Type of Examination (IPCC):	Theory- CIE +SEE, and Lab-SEE.	Exam Hours	03

Course outcome (Course Skill Set)

At the end of the course, the student will be able to:

CO1: Explain the basic concepts of operating systems, including processes, memory, file systems, and resource management

CO2: Analyze CPU scheduling, deadlock handling, memory allocation, and page replacement techniques for solving operating system problems.

CO3: Use Linux commands for file management, process management, file permissions, filters, pipes, and input/output redirection.

CO4: Develop shell scripts using variables, operators, decision-making statements, loops, functions, arrays, and command-line arguments.

CO5: Apply shell scripting to automate file management, backups, log analysis, and basic system monitoring tasks.

Module-1

Operating System Fundamentals

Introduction to Operating Systems: Evolution, Types of Operating Systems, Batch Systems, Multiprogramming, Multitasking, Time-sharing Systems and Real-time Systems. Operating System Services, System Calls, Kernel and User Mode. Operating System Structures and Architectures. Process Concept, Process States, Process Control Block (PCB) and Context Switching.

Text Book 1: Chapter 1 (Sections 1.1–1.12, pp. 4–57), Chapter 2 (Sections 2.1–2.11, pp. 59–102), Chapter 3 (Sections 3.1–3.3, pp. 106–123). Number of Hours:9

Module-2

Process, Memory and File Management

Threads and Multithreading. CPU Scheduling: FCFS, SJF, Priority and Round Robin Scheduling Algorithms. Memory Management: Contiguous Memory Allocation, Paging, Segmentation and Virtual Memory. Page Replacement Algorithms: FIFO, LRU and Optimal. Deadlocks: Prevention, Avoidance, Detection and Recovery. File Systems and Directory Structures.

Text Book 1: Chapter 4 (Sections 4.1–4.7), Chapter 5 (Sections 5.1–5.7), Chapter 8 (Sections 8.1–8.8), Chapter 9 (Sections 9.1–9.7), Chapter 10 (Sections 10.1–10.6), Chapter 13 (Sections 13.1–13.7). Number of Hours:9

Module-3

Linux Environment and Command Utilities

Linux architecture and file system hierarchy. Linux shell environment and Bash shell. Linux file and directory commands. File permissions and ownership. Process

	management commands. Shell expansion and environment variables. Standard input, standard output and standard error. Pipes and input/output redirection. Filter Commands: cat, tee, head, tail, wc, sort, uniq, cut, paste, tr, grep and comm.Regular Expressions and Text Processing: Basic and extended regular expressions, pattern matching using grep, stream editing using sed, field processing and report generation using awk. **Text Book 3:** Chapters 1–7, 9–11, 19 and 20. **Text Book 2:** Chapter 8 (The Shell), Chapter 10 (Customizing the Environment), Chapter 12 (Simple Filters), Chapter 13 (Filters Using Regular Expressions – grep and sed), and Chapter 18 (awk – An Advanced Filter). Number of Hours:8
	Module-4
	Shell Scripting Fundamentals Introduction to Shell Scripting; script creation, execution and debugging; shell variables and environment variables; quoting and command substitution; positional parameters and command-line arguments; arithmetic operations and expressions; conditional statements using if, if-else and case; looping constructs using for, while and until; shell functions; arrays; string manipulation; reading user input; script testing, debugging and basic shell script development. **Text Book 2:** Chapter 8 (*The Shell*), Chapter 10 (*Customizing the Environment*), Chapter **14** (*Essential Shell Programming*), and Chapter **21** (*Advanced Shell Programming*). **Text Book 3:** Chapter **24** (*Writing Your First Script*), Chapter **25** (*Starting a Project*), Chapter **26** (*Top-Down Design*), Chapter **27** (*Flow Control – Branching with if*), Chapter **28** (*Reading Keyboard Input*), Chapter **29** (*Flow Control – Looping with while and until*), Chapter **31** (*Flow Control – Branching with case*), Chapter **32** (*Positional Parameters*), Chapter **33** (*Flow Control – Looping with for*), Chapter **34** (*Strings and Numbers*), and Chapter **35** (*Arrays*). Number of Hours:8
	Module-5
	Shell Scripting and Automation File handling using shell scripts. Text processing using `grep`, `sed` and `awk` in shell scripts. Regular expressions in shell scripting. Log file analysis. CSV data processing. Backup and archival automation. Task scheduling using `cron`. System monitoring scripts. User and process administration scripts. Error handling, signals and traps. Introduction to DevOps automation concepts using Git, Docker and Jenkins through official documentation and instructor-guided demonstrations. Development of mini automation projects integrating Linux commands and shell scripting. **Text Book 2:** Chapter 12 (Simple Filters), Chapter 13 (Filters Using Regular Expressions – grep and sed), Chapter 14 (Essential Shell Programming), Chapter 15 (Essential System

Administration), Chapter 18 (awk – An Advanced Filter), Chapter 21 (Advanced Shell Programming), and Chapter 25 (Advanced System Administration).

Text Book 3: Chapter 19 (Regular Expressions), Chapter 20 (Text Processing), Chapter 24 (Writing Your First Script), Chapter 30 (Troubleshooting), Chapter 32 (Positional Parameters), Chapter 34 (Strings and Numbers), Chapter 35 (Arrays), and Chapter 36 (Exotica).

Number of Hours:8

PRACTICAL COMPONENTS OF IPCC

List of Experiments

Operating Systems Programs (1–4)

Experiment 1: CPU Scheduling Simulator

Develop a program to implement and compare FCFS and SJF Scheduling. The program should calculate Waiting Time, Turnaround Time, Average Waiting Time and Average Turnaround Time.

Experiment 2: Page Replacement Algorithm Simulator

Develop a program to implement FIFO and LRU Page Replacement algorithms. The program should track page hits and page faults, display frame contents after each reference and compare alg12 ptorithm efficiency.

Experiment 3: Banker's Algorithm for Deadlock Avoidance

Develop a program to accept Allocation, Maximum and Available matrices, compute Need Matrix, determine safe state and display safe sequence.

Experiment 4: Inter Process Communication using Pipes

Develop a program to create parent and child processes, communicate using pipes, pass data from parent to child and perform string transformation in child process.

Shell Scripting Programs (5–10)

Experiment 5: Basic File, Directory and Permission Audit Tool

Develop a shell script that accepts a directory path and generates a report containing total files and directories, largest files, empty files and directories, recently modified files and world-writable files.

Experiment 6: Student Record Processing and Grade Analysis

Develop a shell script to accept marks of multiple students, calculate total and average, assign grades, display topper and show pass/fail statistics.

Experiment7: Basic Log File Analyzer using Filters
Develop a shell script to count total log entries, count ERROR/WARNING/INFO messages, extract IP addresses and identify repeated IPs.

Experiment 8: Simple Backup Automation Script

Develop a shell script to backup a directory, compress using tar, add timestamp and log backup status.

	Experiment 9: Basic System Health Monitoring Script Develop a shell script to monitor CPU, memory and disk usage, check threshold values, generate alerts and save reports. **Experiment 10: Basic Deployment Validation Script** Develop a shell script to check installed packages, running services, validate disk and memory availability and generate final PASS/FAIL report.

Text Books:

1. Abraham Silberschatz, Peter B. Galvin and Greg Gagne, *Operating System Concepts*, 10th Edition, Wiley, 2018.
   ISBN: 978-1119456339
2. Sumitabha Das, *UNIX: Concepts and Applications*, 4th Edition, McGraw Hill Education, 2017.
   ISBN: 978-0070635463
3. William E. Shotts Jr., *The Linux Command Line: A Complete Introduction*, 2nd Edition, No Starch Press, 2019.
   ISBN: 978-1593279523

Reference Books/Manuals:

1. Andrew S. Tanenbaum and Herbert Bos, *Modern Operating Systems*, 5th Edition, Pearson, 2022.
   ISBN: 978-0137618880
2. Daniel J. Barrett, *Linux Pocket Guide*, 4th Edition, O'Reilly Media, 2024.
   ISBN: 978-1098156268
3. Scott Chacon and Ben Straub, *Pro Git*, 2nd Edition, Apress, 2019.
   ISBN: 978-1484200773
4. Karl Matthias and Sean P. Kane, *Docker: Up & Running*, 3rd Edition, O'Reilly Media, 2023.
   ISBN: 978-1098131821
5. Nicolas Modrzyk, *Jenkins: The Definitive Guide*, Packt Publishing, 2024. *(or the latest available edition)*

Web links and Video Lectures (e-Resources):

Web Links (e-Resources)

1. Operating System Concepts Companion Website
2. GNU Bash Reference Manual
3. GNU Bash Project Home
4. The Linux Documentation Project (TLDP)
5. Advanced Bash-Scripting Guide (TLDP)
6. Linux Manual Pages Project (man7.org)
7. Linux Foundation Training Resources

8. Git Documentation
9. Docker Documentation
10. Jenkins Documentation

Video Resources

1. NPTEL – Operating Systems
2. NPTEL – Programming in C and Unix
3. MIT OpenCourseWare – Operating System Engineering (6.828)
4. Linux Foundation – Introduction to Linux (LFS101)
5. freeCodeCamp – Linux Command Line for Beginners (YouTube)
6. freeCodeCamp – Bash Scripting Full Course (YouTube)

Assessment Structure (both CIE and SEE)

The weightage assigned to Continuous Internal Evaluation (CIE) and Semester End Examination (SEE) shall be 50% each. A student shall secure a minimum of 50% of the maximum marks prescribed for CIE and a minimum of 40% of the maximum marks prescribed for SEE to be eligible for a pass in the course.

A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to a subject /course only if he/she secures not less than 50% of the aggregate marks obtained in CIE and SEE taken together.

Continuous Internal Evaluation:

(i)Theory component

The CIE marks of 25 shall be earmarked for two tests.

The first test shall be conducted after completing two modules of the syllabus and the second one after completing the rest three modules.

Each test shall be conducted for 25 marks. The average of the two tests scaled down to 25 marks shall constitute the test marks for a maximum of 25 marks.

(ii) Laboratory component

Out of 25 marks, 15 marks shall be assigned for assessment of regular laboratory classwork as per the rubrics listed in Table - ICs/IPCCs1.

The remaining 10 marks shall be based on the practical test conducted by two Internal examiners appointed by the HoD. The allotment of marks shall be as per the rubrics defined for the practical test as indicated in the Table – ICs/IPCCs 1.

To qualify and become eligible to appear for the SEE of the IC/IPCC, a student shall secure at least 13 marks out of 25 in the CIE theory component, and at least 12 marks in the CIE Practical component.

Table – ICs/IPCCs 1 Distribution of marks corresponding to laboratory component

Component	Description	Marks
Allotment of marks for regular classwork		
Ability in Conducting Experiments	Initiative, skill, safety practices, teamwork, and independent handling of equipment during regular lab sessions.	5 marks
Laboratory Record / Journal	Regular and neat maintenance of lab record, accuracy of results, and timely submission.	10 marks
	Subtotal	**15**
Allotment of marks for CIE practical test		
Laboratory Test / Experiment Performance	Ability to set up apparatus, follow procedure, take observations, and obtain correct results in the test experiment.	5 marks
Viva-Voce	Understanding of theory, concepts, and experimental procedure; ability to explain results and answer related questions.	5 marks
	Subtotal	**10**
	Total	**25**

*****

JAVA PROGRAMMING			
Course Code	**1MMC106**	Scheme	2026
Type of course	**IPCC**	Semester	I
Teaching Hours/Week (L:T:P)	3:0:2	CIE Marks	50
Total Hours of Pedagogy L/T/P/SL&TW	42:0:28:50	SEE Marks	50
Credits	04	Total Marks	100
Type of Examination (IPCC):	Theory- CIE +SEE, and Lab- CIE only.	Exam Hours	03

Course Outcomes (Course Skill Set)

At the end of the course, the student will be able to:

CO1: Demonstrate object-oriented programming concepts.
CO2: Apply Java constructs to obtain solutions.
CO3: Develop reusable and efficient Java code.
CO4: Build a Java based client-server application.
CO5: Implement Java application using Spring framework.

Module-1

Java Programming Fundamentals: An overview of Java, Data Types and Operators, Program Control Statements, Introducing Classes, Objects and Methods. Methods, Classes, Constructors, Access control, Pass Objects to Methods, How Arguments are passed, Returning Objects, Method Overloading, Overloading Constructors.

TextBook-1: Chapter – 1, 2, 5, 6, 8 Number of Hours: 09

Module-2

Inheritance, Interfaces and Packages: Inheritance, using super to Call Superclass constructors, using super to Access Superclass Members, creating a Multilevel Hierarchy, Superclass References and Subclass Objects, Method Overriding, Using Abstract Classes. Interface Fundamentals, Creating an Interface, implementing an Interface, Using Interface References, Implementing Multiple Interfaces, Constants in Interfaces, Interfaces can be extended, Nested Interfaces, Package Fundamentals, Packages and Member Access, Importing Packages.

TextBook-1: Chapter – 9, 10 Number of Hours:08

Module-3

Exception Handling and Multi-Threading Programming: The Exception Hierarchy, Exception Handling Fundamentals, the Consequences of an Uncaught Exception, using Multiple catch clauses, Catching subclass Exceptions, Throwing an Exception, Throwable, using finally, using throws; Multithreading: The Thread Class and Runnable Interface, Creating Thread, Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities, Synchronization, using Synchronization Methods, The Synchronized Statement, Thread Communication using notify(), wait() and notify All(), Suspending, Resuming and stopping Threads.

TextBook-1: Chapter – 11, 12 Number of Hours: 08

Module-4

Servlet Programming: Servlet Structure, Packaging, Lifecycle, HTTP Request and response, Handling client request, Form data, HTTP status request headers, HTTP Status codes, HTTP response headers, Handling cookies, Session tracking.

JSP Programming: Need of JSP, Basic syntax, Scripting elements, Limiting Java code in JSP, JSP expression, JSP directives, JSP attributes.

TextBook-3: Chapter – 2, 3, 4, 5, 8, 9 Number of Hours: 08

Module-5

Spring Framework: Introducing the Spring Framework, Problems with the Traditional Approach to J2EE, Lightweight Frameworks, Spring's Values, Spring in Context, Inversion of Control and Dependency Injection, Architecting Applications with Spring.

TextBook-2: Chapter – 1, 2, 3 Number of Hours: 09

PRACTICAL COMPONENTS OF IPCC

List of experiments

1. Develop a Student Management System. Create a Student class with attributes (USN, Name, Semester, CGPA) and methods to read, display, and update student details. Create multiple objects and demonstrate object creation and method invocation.

2. Create a Student class with overloaded constructors to initialize student details in different ways: Default constructor, Constructor with USN and Name, Constructor with USN, Name, Semester, and CGPA. Overload the calculateResult() method to perform different operations: Calculate total marks for 3 subjects, Calculate total marks for 5 subjects, Calculate percentage using obtained marks and maximum marks, Calculate SGPA using grade points of different courses.

3. Develop an Employee Payroll System. Create an abstract class Employee with an abstract method calculateSalary(). Derive classes PermanentEmployee and ContractEmployee and override the salary calculation method. Display salary details of different employees.

4. Develop a Banking Application. Create an interface Transaction with methods deposit(), withdraw(), and displayBalance(). Implement the interface in SavingsAccount and CurrentAccount classes. Organize classes into appropriate packages and import them into the main program.

5. Develop an ATM Simulation. Create user-defined exceptions such as InsufficientBalanceException and InvalidAmountException. Handle invalid withdrawals, negative deposits, division by zero, and invalid menu selections using appropriate exception handling mechanisms.

6. Develop a Railway Ticket Booking System. Create multiple passenger threads attempting to book tickets simultaneously. Use synchronization to avoid overbooking and assign different thread priorities to VIP and regular passengers.

7. Develop a Student Registration Web Application. Design an HTML registration form, process user input using a Servlet, store data in request/session objects, and display the registration details using a JSP page.

8. Develop an Online Course Registration System. Create login, course selection, and confirmation pages. Use Servlets for processing requests, JSP for displaying dynamic content, and session management to maintain user login information.

9. Develop a Student Database Management System using JDBC. Connect to a MySQL database and implement CRUD (Create, Read, Update, Delete) operations on the Student table using prepared statements and proper exception handling.

10. Develop an Employee Management System using Spring Boot. Implement entity mapping, repository, service, and controller layers. Perform CRUD operations on employee records and persist data in a MySQL database.

Textbooks:

1. Herbert Schildt, *Java: The Complete Reference*, McGraw-Hill, 12th Edition, ISBN: ISBN13-9788195096770, 2022.
2. Rod Johnson, Juergen Hoeller, Alef Arendsen, Thomas Risberg, Colin Sampaleanu, *Professional Java™ Development with the Spring Framework*, July 2005, Wrox, ISBN: 9780764574832.
3. Jason Brittain and Brian D. Eubanks, *Java Servlet and JSP Programming,* O'Reilly Media, 1st Edition, 2001

Reference books:

1. Kathy Sierra and Bert Bates, *Head First Design Patterns*, O'Reilly Media.
2. Joseph B. Ottinger Jeff Linwood Dave Minte, *Beginning Hibernate 6 Java Persistence from Beginner to Pro*, 5th Edition, Apress.

Web links and Video Lectures (e-Resources):

1. Java Programming Fundamentals: www.java.com
2. Java Coding: https://www.programiz.com/java-programming
3. JSP Fundamentals: https://jakarta.ee/specifications/pages/
4. Spring Framework: https://spring.io/projects/spring-framework

Assessment Structure (both CIE and SEE)

The weightage assigned to Continuous Internal Evaluation (CIE) and Semester End Examination (SEE) shall be 50% each. A student shall secure a minimum of 50% of the maximum marks prescribed for CIE and a minimum of 40% of the maximum marks prescribed for SEE to be eligible for a pass in the course.

A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to a subject /course only if he/she secures not less than 50% of the aggregate marks obtained in CIE and SEE taken together.

Continuous Internal Evaluation:

(i)Theory component

The CIE marks of 25 shall be earmarked for two tests.

The first test shall be conducted after completing two modules of the syllabus and the second one after completing the rest three modules.

Each test shall be conducted for 25 marks. The average of the two tests scaled down to 25 marks shall constitute the test marks for a maximum of 25 marks.

(ii) Laboratory component

Out of 25 marks, 15 marks shall be assigned for assessment of regular laboratory classwork as per the rubrics listed in Table - ICs/IPCCs1.

The remaining 10 marks shall be based on the practical test conducted by two Internal examiners appointed by the HoD. The allotment of marks shall be as per the rubrics defined for the practical test as indicated in the Table – ICs/IPCCs 1.

To qualify and become eligible to appear for the SEE of the IC/IPCC, a student shall secure at least 13 marks out of 25 in the CIE theory component, and at least 12 marks in the CIE Practical component.

Table – ICs/IPCCs 1 Distribution of marks corresponding to laboratory component

Component	Description	Marks
Allotment of marks for regular classwork		
Ability in Conducting Experiments	Initiative, skill, safety practices, teamwork, and independent handling of equipment during regular lab sessions.	5 marks
Laboratory Record / Journal	Regular and neat maintenance of lab record, accuracy of results, and timely submission.	10 marks
	Subtotal	**15**
Allotment of marks for CIE practical test		
Laboratory Test / Experiment Performance	Ability to set up apparatus, follow procedure, take observations, and obtain correct results in the test experiment.	5 marks
Viva-Voce	Understanding of theory, concepts, and experimental procedure; ability to explain results and answer related questions.	5 marks
	Subtotal	**10**
	Total	**25**

*****

Data Structures and Algorithms Laboratory			
Course Code	**1MMCL107**	Scheme	2026
Type of course	**Professional Core Course Lab**	Semester	I
Teaching Hours/Week (L:T:P)	0: 0: 2	CIE Marks	50
Total Hours of Pedagogy L:T:P:TWSL	0: 0: 28: 2	SEE Marks	50
Credits	1	Total Marks	100
Type of Examination	**Practical**	Exam Hours	3

Course outcome (Course Skill Set)

At the end of the course, the student will be able to:

CO1: Make use of appropriate data structures and algorithmic techniques to solve complex, real-world computational problems.

CO2: Develop code to perform required operations and validate results.

CO3: Make use of appropriate programming tools and development environments for implementing, testing, and debugging data structure and algorithm applications.

CO4: Build a technical document consisting of experiment setup, design, implementation, and results with inferences.

Note:

1. The laboratory syllabus consists of PART-A and PART-B. While PART-A has 6-8 conventional experiments, PART-B has 6 typical open-ended experiments. The maximum marks for the laboratory course are 100.
2. Both PART-A and PART-B are considered for CIE and SEE.
3. Students have answer 1(one) question from PART-A and 1(one) question from PART-B.
   a. The questions set for SEE shall be from among the experiments under PART-A. It is evaluated for **80 marks** out of the maximum 100 marks.
   b. The open-ended question set for SEE shall be any other open-ended question and not selected from the experiments under PART-A. It shall be evaluated for **20 marks.**

For Continuous Internal Evaluation (CIE) during the semester, classwork shall typically include open-ended questions from **Part-B**, along with other similar questions designed to enhance the students' skills.

PART – A
List of experiments

1. Implement a Program in C for converting an Infix Expression to Postfix Expression.
2. Design, develop, and execute a program in C to simulate the working of a Queue of integers using an array. Provide the following operations: a. Insert b. Delete c. Display.

3. Develop a C program to simulate the working of a singly linked list providing the following operations: a. Display & Insert b. Delete from the beginning/end c. Delete a given element.
4. Develop a C program to Implement the following searching techniques: a. Linear Search b. Binary Search.
5. Develop a C program to implement the following sorting algorithms using user defined functions: a. Bubble sort (Ascending order) b. Selection sort (Descending order).
6. Develop a C Program to find Minimum Cost Spanning Tree of a given connected undirected graph using Prim's algorithm.

PART – B
OPEN ENDED EXPERIMENTS

Open-ended experiments are a type of laboratory activity where the outcome is not predetermined and students are given the freedom to explore, design, and conduct the experiment based on the problem statements as per the concepts defined by the course coordinator. It encourages creativity, critical thinking, and inquiry-based learning.

1. **Concept 1: Stack Applications**
   Design and develop an application using stack operations for solving a suitable computational problem.
2. **Concept 2: Queue Applications**
   Develop an application using queue-based data structures to perform efficient data management operations.
3. **Concept 3: Doubly Linked List Applications**
   Design and implement an application using doubly linked list concepts for dynamic data storage and manipulation.
4. **Concept 4: Sorting Techniques and Performance Analysis**
   Implement suitable sorting techniques for a given problem and analyze the performance based on input size and execution time generator.
5. **Concept 5: Graph Algorithms Applications**
   Apply graph-based algorithms to solve problems related to shortest path.
6. **Concept 6: Recursive / Non recursive Techniques**
   Develop an application using recursive or non-recursive techniques to solve computational problems.

Text books:

1. Ellis Horowitz and Sartaj Sahni, Fundamentals of Data Structures in C, 2nd Edition (2023), Universities Press(India) Private limited.
2. Anany Levitin, Introduction to the Design and Analysis of Algorithms, 3rd Edition (2024), Pearson Education, Limited.

Reference books/Manuals:

1. Richard F. Gilberg and Behrouz A. Forouzan, Data Structures: A Pseudocode Approach with C, 2nd Edition, (2007), Cengage Learning.
2. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C*, 2nd Edition, (2002), Pearson Education.
3. Reema Thareja, *Data Structures Using C*, 2nd Edition, (2014), Oxford University Press.

Web links and Video Lectures (e-Resources):

- NPTEL – Programming, Data Structures and Algorithms using C: https://nptel.ac.in/courses/106105171
- NPTEL – Data Structures and Algorithms: https://nptel.ac.in/courses/106102064
- VisuAlgo – Data Structure Visualizations: https://visualgo.net/en
- GeeksforGeeks – Data Structures in C: https://www.geeksforgeeks.org/data-structures/
- GeeksforGeeks – C Programming: https://www.geeksforgeeks.org/c-programming-language/
- Programiz – C Programming and DSA: https://www.programiz.com/c-programming https://www.programiz.com/dsa
- OnlineGDB – Online C Compiler: https://www.onlinegdb.com/online_c_compiler

Assessment Structure (both CIE and SEE)

The weightage assigned to Continuous Internal Evaluation (CIE) and Semester End Examination (SEE) shall be 50% each. A student shall secure a minimum of 50% of the maximum marks prescribed for CIE and a minimum of 40% of the maximum marks prescribed for SEE to be eligible for a pass in the course.

A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to a subject /course only if he/she secures not less than 50% of the aggregate marks obtained in CIE and SEE taken together.

Continuous Internal Evaluation (CIE):

For the Professional Core Course Laboratory, the maximum marks prescribed are 100. Out of these, 50 marks are allotted for Continuous Internal Evaluation (CIE) and 100 marks for the Semester End Examination (SEE). The SEE marks obtained out of 100 shall be proportionately scaled down to 50 marks. The final result shall be declared based on the aggregate of the CIE marks (50) and the scaled-down SEE marks (50), totalling 100 marks."

Out of 50 marks, 40 marks shall be assigned for assessment of regular laboratory PART -A classwork, and 10 marks for Part – B. (Table 1)

Out of 40 marks, 25 marks shall be based on the regular classwork and allotment of marks shall be as per rubrics listed in Table 1.

The remaining 15 marks shall be based on the practical test conducted by two Internal examiners appointed by the HoD. The allotment of marks shall be as per the rubrics defined for the practical test as indicated in the Table 1.

The open-ended question set for CIE shall be any other open-ended question and not selected from the experiments under PART-A. It shall be evaluated for 10 marks. The allotment of marks shall be as per rubrics listed in Table 2

Table – 1 Distribution of marks corresponding to laboratory component

Component	Description	Marks
Allotment of marks for regular classwork		
Ability in Conducting Experiments	Initiative, skill, safety practices, teamwork, and independent handling of equipment during regular lab sessions.	10 marks
Laboratory Record / Journal	Regular and neat maintenance of lab record, accuracy of results, and timely submission.	15 marks
	Subtotal	**25**
Allotment of marks for CIE practical test		
Laboratory Test / Experiment Performance	Ability to set up apparatus, follow procedure, take observations, and obtain correct results in the test experiment.	5 marks
Viva-Voce	Understanding of theory, concepts, and experimental procedure; ability to explain results and answer related questions.	10marks
	Subtotal	**15**
	Total	**40**

Table 2 Distribution of marks corresponding to PART -B of the laboratory component				
Evaluation Rubrics for continuous internal evaluation CIE). Maximum marks:10				
Criteria and Guiding factors	**Students' Level and Rubrics**			
	Excellent: E, Good: G, Fair: F Needs Improvements: NI			
	E	**G**	**F**	**NI**
(1) Problem Understanding **Criteria:** Is the problem understood correctly and framed appropriately? (i)Clarity in interpreting the given problem. (ii)Identification of relevant concepts and constraints. (iii) Ability to rephrase or model the problem logically.	2.0	1.5	1.0	0.5
(2) Conceptual Knowledge **Criteria**: Are the chosen methods or concepts valid for this type of problem? (i)Application of appropriate theories, principles, or laws. (ii)Integration of multiple concepts where necessary. (iii)Accuracy in stating relevant equations or models.	2.0	1.5	1.0	0.5
(3) Approach and Strategy **Criteria**: Is the approach well thought out and methodical? (i)Logical structure and progression. (ii)Use of diagrams, sketches, or flowcharts if applicable. (iii)Identification of assumptions and boundary conditions.	2.0	1.5	1.0	0.5
(4) Execution and Accuracy **Criteria**: Are the calculations correct? Is the final solution reasonable? (i) Correct application of formulas/methods. (ii)Accuracy of calculations or simulations (if any). (iii)Unit consistency and result verification.	2.0	1.5	1.0	0.5
(5) Innovation or Depth **Criteria**: Has the student shown original thinking or alternative methods? (i)Novel or efficient solutions. (ii)Consideration of alternate approaches. (iii)Interdisciplinary integration or practical extensions.	1	0.75	0.5	0.25
(6) Presentation and Clarity **Criteria**: Is the answer easy to follow and professionally written? (i)Organization and neatness. (ii)Clarity of explanation, labelling, and steps. (iii) Use of proper technical terminology.	1	0.75	0.5	0.25

Total	10	7.5	5.0	2.5

Note:

(i)The marks shown for Excellent, Good, Fair, Needs Improvement are indicative. The evaluator may also assign intermediate scores depending on the quality of work.

(ii)If the total score results in a fraction, it should be rounded off to the next higher integer.

(iii)The listed criteria are guidelines; evaluators may also consider other authentic dimensions (e.g., depth of insight, critical thinking, and practical awareness, recognition of limitations, or justified compromises.

(iv)To award the full10 marks in the excellent category, the student must demonstrate reasonably good performance across all six criteria; otherwise, proportionally lower marks may be given.

****

WEB PROGRAMMING LAB			
Course Code	**1MMCL108**	Scheme	2026
Type of course	**Professional Core Course Lab**	Semester	I
Teaching Hours/Week (L:T:P)	0:0:2	CIE Marks	50
Total Hours of Pedagogy L:T:P:SL:TW	0:0:28:2	SEE Marks	50
Credits	01	Total Marks	100
Type of Examination	**Practical**	Exam Hours	03

Course outcomes

At the end of the course, the student will be able to:

- CO1: Design responsive web pages using HTML5, CSS3, Bootstrap, and JavaScript.
- CO2: Implement client-side and server-side scripting techniques for form processing.
- CO3: Develop database-driven web applications by performing CRUD operations using SQL.
- CO4: Build web applications that interact with NoSQL databases such as MongoDB.

Note:

1. The laboratory syllabus consists of PART-A and PART-B. While PART-A has 6-8 conventional experiments, PART-B has 6 typical open-ended experiments. The maximum marks for the laboratory course are 100.
2. Both PART-A and PART-B are considered for CIE and SEE.
3. Students have answer 1(one) question from PART-A and 1(one) question from PART-B.
   a. The questions set for SEE shall be from among the experiments under PART-A. It is evaluated for **80 marks** out of the maximum 100 marks.
   b. The open-ended question set for SEE shall be any other open-ended question and not selected from the experiments under PART-A. It shall be evaluated for **20 marks.**

For Continuous Internal Evaluation (CIE) during the semester, classwork shall typically include open-ended questions from **Part-B**, along with other similar questions designed to enhance the students' skills.

PART – A
List of experiments

1. Design and develop a responsive multi-page website for an Online Book Store. The website should include Home, About, Events/Services, Gallery, Contact, and Registration pages. Use HTML5 semantic tags, CSS3 styling, Flexbox/Grid, Bootstrap components such as Navbar, Cards, Carousel, Forms, Buttons, and Footer.

2. Develop an Online Student Registration Form containing fields such as Name, USN, Email, Mobile Number, Password, Confirm Password, Date of Birth, Gender, Course, Address, and Terms & Conditions. Implement client-side validation using JavaScript for mandatory fields, password strength, email format, phone number, age verification, password matching, and error messages without page refresh.

3. Develop a Student Result Management Dashboard. Allow users to add, edit, delete, and search student records dynamically using DOM manipulation. Calculate total marks, percentage, grade, and display results in an HTML table without reloading the page. Include sorting and filtering features.

4. Create a Weather Information Application (or Employee Directory/Product Catalog). Fetch JSON data from a public REST API (or local JSON file) using Fetch API. Display data dynamically as cards or tables. Implement loading indicators, error handling, search functionality, filtering, and asynchronous updates without page refresh.

5. Develop a User Login and Shopping Cart Simulation. Store username and user preferences using cookies. Maintain login status and shopping cart contents using session storage (or server-side sessions if PHP is used). Provide Login, Logout, Add to Cart, Remove from Cart, and View Cart functionalities. Display personalized welcome messages based on stored session information.

6. Develop a Student Information REST API. Implement CRUD operations (Create, Read, Update, Delete) for student records using Node.js with Express (or PHP). Create REST endpoints such as GET, POST, PUT, and DELETE. Test the APIs using Postman or browser requests. Return responses in JSON format and validate request data before processing.

PART – B

OPEN ENDED EXPERIMENTS

Open-ended experiments are a type of laboratory activity where the outcome is not predetermined and students are given the freedom to explore, design, and conduct the experiment based on the problem statements as per the concepts defined by the course coordinator. It encourages creativity, critical thinking, and inquiry-based learning.

1. Develop a web-based College ERP module for managing student information which will include the following:
   - Design responsive web pages using Bootstrap.
   - Implement Login page.
   - Create Student Registration form with JavaScript validation.
   - Display student details using DOM manipulation.
   - Develop REST APIs for student CRUD operations.
   - Maintain user login using session/cookies.
   - Fetch and display student data from the server asynchronously using Fetch API.

2. Develop a web application to manage campus placement activities which will include the following:
   - Responsive dashboard for students and placement officer.
   - Student profile registration.
   - Company registration.
   - Job posting.
   - Student application form with validation.
   - Display eligible students dynamically.
   - Store login information using sessions.
   - Create REST APIs for students and companies.

3. Develop a Hospital Appointment System which will include the following features:
    - Create responsive patient registration and login pages.
    - Appointment booking form with validation.
    - Display available doctors dynamically.
    - Book and cancel appointments.
    - Store user session after login.
    - Develop REST APIs for doctors and appointments.

4. Develop a web application for managing Inventory along with the following features:
    - Product registration form.
    - Stock entry and update.
    - Search products.
    - Display inventory dynamically.
    - Maintain login session.
    - REST APIs for inventory operations.

5. Develop a basic Online Examination Portal along with the following features:
    - Student login.
    - Display questions fetched from JSON.
    - Timer using JavaScript.
    - Auto evaluation.
    - Display score dynamically.
    - Store exam session.
    - REST APIs for questions and results.

6. Develop a Hostel Room allocation System along with the following features:
    - Student registration.
    - Hostel room booking.
    - Room availability display.
    - Student search.
    - Session management.
    - CRUD APIs for rooms and students.

Textbooks:

1. Ramez Elmasri, Shamkant B. Navathe, *Fundamentals of Database Systems*, Pearson, 7th Edition, 2016.
2. Shashank Tiwari, *Professional NoSQL*, Wiley India/Wrox Publication, 1st Edition, 2011.
3. Kristina Chodorow, *MongoDB: The Definitive Guide*, O'Reilly Media, 1st Edition.
4. Robin Nixon, *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*, O'Reilly Media, 5th Edition.

Reference books/Manuals:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan, *Database System Concepts*, McGraw Hill, 7th Edition.
2. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", 3rd Edition, McGraw Hill, 2018.
3. Pramod J. Sadalage and Martin Fowler, *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*, Addison-Wesley Professional
4. Ethan Brown, *Web Development with Node and Express*, O'Reilly Media.

Web links and Video Lectures (e-Resources):

- MySQL Documentation: https://dev.mysql.com
- https://w3schools.com/sql
- MongoDB Documentation: https://www.mongodb.com/resources/products/fundamentals/mongodb-tutorials

Interactive SQL: https://sqlbolt.com/

Assessment Structure (both CIE and SEE)

The weightage assigned to Continuous Internal Evaluation (CIE) and Semester End Examination (SEE) shall be 50% each. A student shall secure a minimum of 50% of the maximum marks prescribed for CIE and a minimum of 40% of the maximum marks prescribed for SEE to be eligible for a pass in the course.

A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to a subject /course only if he/she secures not less than 50% of the aggregate marks obtained in CIE and SEE taken together.

Table – 1 Distribution of marks corresponding to laboratory component

Component	Description	Marks
Allotment of marks for regular classwork		
Ability in Conducting Experiments	Initiative, skill, safety practices, teamwork, and independent handling of equipment during regular lab sessions.	10 marks
Laboratory Record / Journal	Regular and neat maintenance of lab record, accuracy of results, and timely submission.	15 marks
	Subtotal	**25**
Allotment of marks for CIE practical test		
Laboratory Test / Experiment Performance	Ability to set up apparatus, follow procedure, take observations, and obtain correct results in the test experiment.	5 marks
Viva-Voce	Understanding of theory, concepts, and experimental procedure; ability to explain results and answer related questions.	10marks
	Subtotal	**15**

Total	**40**

Table 2 Distribution of marks corresponding to PART -B of the laboratory component				
Evaluation Rubrics for continuous internal evaluation CIE). Maximum marks:10				
Criteria and Guiding factors	**Students' Level and Rubrics**			
	Excellent: E, Good: G, Fair: F Needs Improvements: NI			
	E	**G**	**F**	**NI**
(1) Problem Understanding **Criteria:** Is the problem understood correctly and framed appropriately? (i)Clarity in interpreting the given problem. (ii)Identification of relevant concepts and constraints. (iii) Ability to rephrase or model the problem logically.	2.0	1.5	1.0	0.5
(2) Conceptual Knowledge **Criteria**: Are the chosen methods or concepts valid for this type of problem? (i)Application of appropriate theories, principles, or laws. (ii)Integration of multiple concepts where necessary. (iii)Accuracy in stating relevant equations or models.	2.0	1.5	1.0	0.5
(3) Approach and Strategy **Criteria**: Is the approach well thought out and methodical? (i)Logical structure and progression. (ii)Use of diagrams, sketches, or flowcharts if applicable. (iii)Identification of assumptions and boundary conditions.	2.0	1.5	1.0	0.5
(4) Execution and Accuracy **Criteria**: Are the calculations correct? Is the final solution reasonable? (i) Correct application of formulas/methods. (ii)Accuracy of calculations or simulations (if any). (iii)Unit consistency and result verification.	2.0	1.5	1.0	0.5
(5) Innovation or Depth **Criteria**: Has the student shown original thinking or alternative methods? (i)Novel or efficient solutions. (ii)Consideration of alternate approaches. (iii)Interdisciplinary integration or practical extensions.	1	0.75	0.5	0.25

(6) Presentation and Clarity **Criteria**: Is the answer easy to follow and professionally written? (i)Organization and neatness. (ii)Clarity of explanation, labelling, and steps. (iii) Use of proper technical terminology.	1	0.75	0.5	0.25
Total	10	7.5	5.0	2.5

Note:

(i)The marks shown for Excellent, Good, Fair, Needs Improvement are indicative. The evaluator may also assign intermediate scores depending on the quality of work.

(ii)If the total score results in a fraction, it should be rounded off to the next higher integer.

(iii)The listed criteria are guidelines; evaluators may also consider other authentic dimensions (e.g., depth of insight, critical thinking, and practical awareness, recognition of limitations, or justified compromises.

(iv)To award the full10 marks in the excellent category, the student must demonstrate reasonably good performance across all six criteria; otherwise, proportionally lower marks may be given.

****