Mahatma Education Society's

Pillai HOC College of Arts, Science & Commerce (Autonomous)

Rasayani

Affiliated to University of Mumbai
NAAC Accredited with "A+" Grade in cycle II
ISO 9001:2015 Certified

SYLLABUS

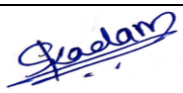
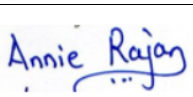
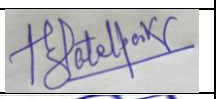
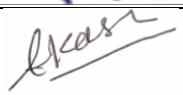
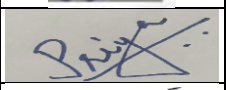
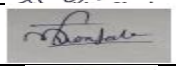
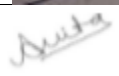
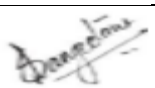
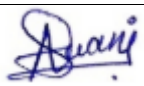
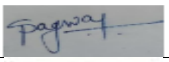
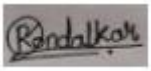
Bachelors of Science (B. Sc.) in Information Technology

S. Y. B. Sc. Information Technology

As per National Education Policy 2020

Academic Year 2026-27

Mahatma Education Society's
College Code: 870
PILLAI HOC COLLEGE OF ARTS, SCIENCE & COMMERCE
Pillai HOCL Educational Campus, HOC Colony, Rasayani, Via. Panvel, Dist. Raigad. Pin 410207
Tel: 02192 - 669000 / 01 / 02 / 03 / 04 / 05 / 06 / 07 / 08 / 09 Website : www.phcasc.ac.in Email : phcasc@mes.ac.in
(NAAC Accredited 'A+' Grade , CGPA – 3.26 in Cycle 2 & ISO 9001:2015 Certified)
Affiliated to the University of Mumbai, Approved by Government of Maharashtra
(AUTONOMOUS COLLEGE)

Sr. No	Name	Designation	Signature
1	Dr. Swapna Kadam	Vice Chancellor Nominee	
2	Dr. Annie Rajan,	Subject Expert	
3	Dr. Homraj Patelpaik	Subject Expert	
4	Mr. Swapnil H. Patil	Industry Representative	
5	Mr. Akash Ghadge	Alumni Representative	
6	Dr. Rinkoo Shantnu	Principal	
7	Ms. Priyanka Sorte	Member	
8	Ms. Priya Prakash	Member	
9	Ms. Harshita Singh	Member	
10	Ms. Mrunal Wanjale	Member	
11	Ms. Arpita Kante	Member	
12	Ms. Anita Mhatre	Member	
13	Ms. Sangeeta Menon	Member	
14	Ms. Aarti Wani	Member	
15	Ms. Rutuja Madane	Member	
18	Ms. Sonali Dagwar	Member	
19	Ms. Rutuja Kondalkar	Member	

1. Introduction

A **B.Sc. in Information Technology** is a three-year undergraduate program designed to bridge the gap between theoretical knowledge and practical application. This degree is designed to equip students with the essential knowledge and skills needed to manage, process, secure, and communicate information using computer systems and applications. Throughout the course, students develop key competencies:
Throughout the course, students develop key competencies:

- **Focus:** B.Sc. IT programs are generally **application-oriented**, emphasizing the practical implementation and management of IT systems, rather than the deep theoretical underpinnings. The curriculum revolves primarily around **databases, software, and networking**.
- **Duration:** The program typically lasts **three years** and is divided into six semesters.
- **Goal:** The primary goal is to produce graduates capable of designing, developing, implementing, and maintaining various IT solutions to solve real-world problems for businesses and organizations.

This interdisciplinary training not only prepares students for technical employment and advanced studies but also encourages them to launch their own startups or venture into new career paths, with the benefit of excellent placement and incubation assistance. The curriculum ensures exposure to modern advancements and new sub-fields within the field.

2. Programme Outcomes (POs)

PO. No.	PO Title	PO's in Brief
PO1	**Fundamental Knowledge Acquisition**	Graduates will demonstrate a comprehensive and foundational knowledge of their chosen discipline along with an awareness of interdisciplinary connections.
PO2	**Critical Thinking and Analytical Reasoning**	Graduates will be able to analyse complex problems, synthesize data from multiple sources (qualitative and quantitative), and employ logical reasoning to formulate well-supported conclusions and arguments.
PO3	**Effective Communication**	Graduates will exhibit proficiency in both written and oral communication, articulating ideas clearly, persuasively, and ethically to diverse audiences
PO4	**Problem Solving**	Graduates will possess the ability to identify, formulate, and design solutions for real-world problems in their professional or social contexts, applying relevant theoretical knowledge and practical skills.

PO5	**Information and Digital Literacy**	Graduates will demonstrate the capability to locate, evaluate, and effectively use information from various sources, and utilize modern tools and Information and Communication Technology (ICT) for professional and academic tasks.
PO6	**Research Skills and Scientific Temperament**	Graduates will develop a sense of inquiry and research methodology, including the ability to design experiments (where applicable), collect and analyse data, and interpret results while maintaining scientific rigor and intellectual honesty.
PO7	**Ethical Reasoning and Professional Integrity**	Graduates will recognize ethical dilemmas, commit to professional and academic ethics, and demonstrate an understanding of moral and social responsibilities in their personal and professional conduct.
PO8	**Employability and Professional Skills**	Graduates will acquire the necessary job-ready skills, managerial competencies, and professional values to secure gainful employment or pursue advanced education in their respective fields.
PO9	**Environmental and Sustainability Consciousness**	Graduates will understand the importance of environmental conservation and sustainable development, displaying responsibility toward ecological challenges and advocating for healthy environmental practices.
PO10	**Life-Long Learning**	Graduates will develop the capacity for independent and self-directed learning to continuously upgrade their knowledge and skills, enabling them to adapt to rapid technological and societal changes.
PO11	**Civic and Social Responsibility**	Graduates will act as responsible citizens with an informed awareness of constitutional values, engaging proactively in community development and addressing social needs.
PO12	**Empathy and Social Intelligence**	Graduates will be able to cultivate and demonstrate affective, interpersonal, social and emotional intelligence.

3.Programme Specific Outcomes (PSOs)

PSOs. No.	PSO Title	PSOs in brief
PSO1	**Software Development Fundamental**	Develop understanding to design and build functional software using programming, data structures, databases, and web technologies.
PSO2	**Advanced Technical Problem Solving**	Apply the skill to solve tough IT problems using critical thinking and applying concepts like AI, Machine Learning, Cloud, and IoT.
PSO3	**Project Management and Research Communication**	Develop a research mindset to plan and manage IT projects, and clearly communicate technical ideas to others.
PSO4	**System Analysis and Secure Design**	Understand what users need and design secure, working IT systems using proper analysis methods.

4. Evaluation Pattern

Marking Code	Marking Scheme
A	50 Marks Semester End Exam, 50 Marks Continuous Assessment (distributed within 15 Marks Class Test, 15 Marks Presentation & Assignment, 10 Marks Online Quiz, 10 Marks Attendance & Class Participation)
B	50 Marks Semester End Exam
C	100 marks Continuous Assessment (distributed within 30 Marks Class Test, 30 Marks Presentation & Assignment, 30 Marks Online Quiz, 10 Attendance & Class Participation)
D	50 Marks of Continuous Assessment (distributed within 15 Marks Class Test, 15 Marks Presentation & Assignment, 10 Marks Online Quiz, 10 Marks Attendance & Class Participation)
E	50 Marks Practical Examination (distributed within 30 Marks Practical Module 1 & 2, 10 Marks Journal, 10 Marks Viva)

Course Structure

Semester - III							
Course Code	**Course Type**	**Course Title**	**Theory/ Practical**	**Marks**	**Credits**	**Lectures /Week**	**Evaluation Pattern**
HUSIT208	Major	CODE WITH PYTHON	Theory	100	2	2	A
HUSIT208P	Major-Practical	PRACTICAL(HUSIT208)	Practical	50	1	2	E
HUSIT209	Major	PRINCIPLES OF OPERATING SYSTEM	Theory	100	2	2	A
HUSIT209P	Major-Practical	PRACTICAL(HUSIT209)	Practical	50	1	2	E
HUSIT210	Major	DATA STRUCTURES	Theory	100	2	2	A
HUSIT210P	Major-Practical	PRACTICAL(HUSIT210)	Practical	50	1	2	E
HUSIT211	Minor	R PROGRAMMING FOR DATA SCIENCE	Theory	100	2	2	A
HUSIT211P	Minor-Practical	PRACTICAL(HUSIT211)	Practical	50	1	2	E
HUSIT212	SEC	MEAN STACK DEVELOPMENT	Theory	100	2	2	A
HUSIT212P	SEC Practical	PRACTICAL(HUSIT212)	Practical	50	1	2	E
	AEC	हिंदी भाषा एवं साहित्य संवर्धन	Theory	50	2	2	D
	OE	RESEARCH METHODOLOGY	Theory	100	3		C
	CC	EXTENSION/NSS	Theory	50	2		D
				950	**22**		

Abbreviations:

SEC: Skill Enhancement Course
IKS: Indian Knowledge System
VAC: Value Added Course
VEC: Value Education Course
AEC: Ability Enhancement Course
OE: Open Elective

BOS	**Mathematics, Statistics and Computer Application**				
Course	**Code With Python**				
Course Code	**HUSIT208**	**Level**	**5.0**		
			Theory	**Practical**	**Total**
Semester	**III**	**Credit**	02	02	04
Type	**Major**	**No of Teaching hours**	**30**	**30**	**60**

Evaluation/ Assessment	**Total Marks**	**Semester End**	**Continuous Evaluation**	**Practical Examination**
	150	**50**	**50**	**50**

Learning Objectives	
1	Master the core features of Python, including its execution model and a wide range of data types.
2	Develop proficiency in control flow by working with conditional statements, loops and other control structures.
3	Work efficiently with arrays, strings, and complex data structures, leveraging Python's capabilities for data manipulation.
4	Apply functions, modules, and string operations to solve real-world programming problems with flexibility and ease.
5	Manage file operations, utilize regular expressions, and handle date and time functions for comprehensive Python programming tasks.

Course Outcomes	
CO1	Demonstrate mastery of Python features to tackle a wide range of programming challenges & Utilize control flow statements to ensure accurate and logical program execution
CO2	Efficiently manipulate arrays, strings, and data structures to enhance. data handling and problem-solving.
CO3	Design modular, efficient programs by leveraging functions, modules, and string operations.
CO4	Manage file operations, employ regular expressions, and manipulate date and time data to improve program functionality and performance.

Modules At Glance

Module No.	**Content**	**No. of Hours**	**Mapping with CO**
1	Introduction to Python Programming	15	CO1,CO2,CO3
2	Exploring Lists, Tuples and Dictionaries	15	CO3,CO4
		30	

Syllabus

Module No.	Content	No. of Hours
1	**Basic Elements of Python Programming:** Features of Python, Execution of a Python Program, Python Interpreter, Comments, IDLE, Data types, Dictionary, Sets, Mapping, Basic Elements of Python, Variables, Input Function, Output Statements, Command Line Arguments. Operators, Precedence of Operators, Associativity of Operators **Control Statements:** The if statement, The if ... else Statement, The if ... elif ... else Statement, Loop Statement- while loop, for loop, Infinite loop, Nested loop, The else suite, break statement, continue statement, pass statement, assert statement, return statement. **Arrays:** Creating Arrays, Indexing and Slicing of Arrays, Basic Array Operations, Arrays Processing, Mathematical Operations on Array, Aliasing Arrays, Slicing and Indexing in NumPy Arrays, Basic slicing, Advanced Indexing, Dimensions and Attributes of an Array **Functions:** Function definition and call, Returning Results, Returning Multiple Values from a Function,Pass Value by Object Reference, Parameters and Arguments, Recursive Functions. Modules in Python. Strings: Creating Strings, Working with Strings, Formatting Strings	15
2	**List:** Exploring List, Tuples and Dictionaries: Lists, List Functions and Methods, List Operations, List Slices, Nested Lists, Tuples, Functions in Tuple. **Dictionaries:** Creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries **Files in Python:** Opening and Closing a File, Working with Text Files, Working with Binary Files, The 'with' statement, The seek() and tell() Methods, Random Accessing of Binary Files, Zipping and Unzipping Files, **Regular Expressions:** Introduction, Sequence Characters in Regular Expressions, Special Characters in Regular Expressions, Using Regular Expression on Files. **Date And Time in Python:** Time, Date, Date and Time Now, combining date and times, formatting date and time, Finding and comparing dates, Sorting dates, Knowing the Time taken by a Program, Working with Calendar module	15
Case Studies		
M1	Student Result Management System using Python A college wants to develop a Student Result Management System using Python to manage student details, marks, and results. The system should allow the user to enter student data, process marks, calculate results, and generate reports efficiently	

M2	Library File-Based Information Management System A college library wants to develop a Python-based system to manage book records, member details, and borrowing history. The system stores data in files, processes records using collections, extracts information using regular expressions, and tracks borrowing dates and times.

Reference Books:

1. Learning Python, Fourth Edition by Mark Lutz, O'Reilly Media, 4th Edition, 2009
2. Python Basics: A Practical Introduction to Python 3 (Revised and Updated) by David Amos, Dan Bader, Joanna Jablonski, Fletcher Heisler, 4th Edition
3. Let Us Python by Yashwant B. Kanetkar, BPB Publication, 3rd Edition, 2019
4. Python: The Complete Reference by Martin C. Brown, McGraw Hill, 3rd Edition, 2018
5. Beginning Python: From Novice to Professional by Magnus Lie Hetland, Apress, 2017

Semester End Evaluation (50 Marks) **Time : 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks : 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Syllabus

List of Practical		CO Mapping	Hours
1	Write programs for the following: a. Write a program that asks the user to enter their name and their age. Print out a message addressed to them that tells them the year that they will turn 100 years old. b. Write a program to accept a number from the user and depending on whether the number is even or odd, print out an appropriate message to the user.	CO1, CO2	3 HRS
2	Write programs for the following: a.Write a program to generate the Fibonacci series. b.Write a program to accept a number from the user display sum of its digits.	CO2	3 HRS
3	Write programs for the following: a. Write a program to perform basic operations, indexing and slicing on arrays. b. Write a program to implement mathematical functions on arrays.	CO2	3 HRS
4	Write programs for the following: a. Write a program to perform slicing, basic and advanced indexing on NumPy arrays. b .Write a program to analyze dimensions and attributes of arrays	CO2	3 HRS

5	Write programs for the following: a. Write a function to check the input value is Armstrong and also write the function for Palindrome. b. Write a recursive function to print the factorial for a given number.	CO2	3 HRS
6	Write programs for the following: a. Write a program to compute number of characters and words in a string. b. Create a file geometry.py to calculate base areas for shapes square and circle. In another file, write a function pointyShapeVolume(x, y, squareBase) that calculates the volume of a square pyramid if squareBase is True and of a right circular cone if squareBase is False. X is the length of an edge on a square if squareBase is True and the radius of a circle when squareBase is False. y is the height of the object. First use squareBase to distinguish the cases. Use the circleArea and squareArea from the geometry module to calculate the base areas.	CO3	3 HRS
7	Write programs for the following: a. Write a program that takes two lists and returns True if they have at least one common member. b. Write a Python script to sort (ascending and descending) a dictionary by value.	CO3	3 HRS
8	Write programs for the following: a. Write a program to accept and pass radius to a function that returns area and circumference (using tuple). b. Write a program to perform basic file operations on text files and binary files.	CO4	3 HRS
9	Write programs for the following: a. a.Write a program to count the occurrences of a specific word in a file using regular expressions. b. b.Write a program to extract all hyperlinks (<a href="...">) from an HTML file.	CO4	3 HRS
10	Write programs for the following: a. Write a program that compares two dates (in DD/MM/YYYY format) and prints which one is earlier. b. Write a program to measure program execution time. c. Write a program using the calendar module to print the weekday of the first day of a given month and year.	CO4	3 HRS

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10

BOS	**Mathematics, Statistics and Computer Application**				
Course	**Principles of Operating System**				
Course Code	**HUSIT209**	**Level**	**5.0**		
			Theory	**Practical**	**Total**
Semester	**III**	**Credit**	02	01	03
Type	**Major**	**No of Teaching hours**	**30**	**30**	**60**

Evaluation/ Assessment	**Total Marks**	**Semester End**	**Continuous Evaluation**	**Practical Examination**
	150	**50**	**50**	**50**

Learning Objectives	
1	To understand the fundamentals and structure of modern operating systems.
2	To analyze process management and CPU scheduling techniques.
3	To understand multithreading and concurrency control mechanisms.
4	To evaluate memory management and virtual memory techniques.
5	To understand deadlock handling and system reliability.
6	To familiarize students with modern file systems used in industry.

Course Outcomes	
After successful completion of this course, students would be able to: -	
CO1	Explain the fundamentals of operating systems, including architecture, system calls, and protection mechanisms.
CO2	Analyze process and thread management, including CPU scheduling algorithms and multithreading.
CO3	Apply concurrency and synchronization techniques to manage race conditions and process coordination.
CO4	Describe memory management, deadlock handling, and file system concepts used in modern operating systems.

Modules at Glance

Module No.	**Content**	**No. of Hours**	**CO Mapping**
1	Modern Operating Systems & Process Management	15	CO1,CO2
2	Memory, Reliability & Storage Systems	15	CO3, CO4

Syllabus

Module No.	Content	No. of Lectures
1	**Operating System Fundamentals:** Role and functions of operating systems, types of operating systems (including modern systems), operating system services and utilities, user mode and kernel mode, system calls, basics of protection and security. **Process & Thread Management:** Process concept, process states, process control block, Inter-process Communication, scheduling algorithms (FCFS, SJF, Priority, Round Robin), Threads and multithreading in modern operating systems. **Concurrency & Synchronization:** Concurrency in operating systems, Race conditions and critical section problem, semaphores and mutex locks, classical synchronization problems.	15
2	**Memory Management:** Main memory organization, swapping, paging, Contiguous Memory Allocation, Structure of page table, virtual memory, demand paging, page replacement algorithms (FIFO, LRU, Optimal, Clock), frame allocation, thrashing. **Deadlock Handling:** Deadlock concept, Deadlock conditions, deadlock prevention, deadlock avoidance (Banker's algorithm),deadlock detection & recovery from deadlock, deadlocks in real-world systems. **File Systems & Storage Management:** File concept, file attributes, file access methods, directory structures, file system architecture, disk allocation methods, overview of modern file systems (EXT4, NTFS, FAT).	15
M1	An operating system must handle multiple user processes where response time is critical and all processes should get equal CPU access. Which CPU scheduling algorithm should be used and why?	
M2	A system uses demand paging with FIFO page replacement. Given a reference string and number of frames, calculate the number of page faults.	

Reference Books:

1. Silberschatz, A., Galvin, P. B., & Gagne, G. (2018). Operating system concepts (10th ed.). Wiley.
2. Tanenbaum, A. S., & Bos, H. (2015). Modern operating systems (4th ed.). Pearson Education.
3. Stallings, W. (2018). Operating systems: Internals and design principles (9th ed.). Pearson Education.
4. Dhamdhere, D. M. (2007). Operating systems: A concept-based approach (2nd ed.). McGraw-Hill Education.
5. Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2018). Operating systems: Three easy pieces. Arpaci-Dusseau Books.

Semester End Evaluation (50 Marks) **Time: 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks: 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Practical Syllabus

Sr. No	List of Practical	No. of. Lectures	CO Mapping
1	Write a program to demonstrate process creation using fork() system call and illustrate parent–child process execution.	3	CO1
2	Write a program to implement Inter-Process Communication using pipes between two processes.	3	CO1
3	Write a program to demonstrate multithreading using POSIX threads (pthreads) for concurrent execution of tasks.	3	CO 2
4	Write a multithreaded program to generate the Fibonacci sequence using thread libraries.	3	CO2
5	Write a program to implement the Bounded Buffer (Producer–Consumer) problem using semaphores or mutex locks.	3	CO3
6	Write a program to simulate the Readers–Writers synchronization problem.	3	CO3
7	Write a program to implement CPU scheduling algorithms (FCFS and Round Robin) and calculate waiting time and turnaround time.	3	CO2
8	Write a program to implement Banker's Algorithm for deadlock avoidance and determine safe sequence.	3	CO4
9	Write a program to simulate Page Replacement Algorithms (FIFO and LRU) and analyze page faults.	3	CO4
10	Write a program demonstrating basic file system operations such as file creation, reading, writing, and deletion using system calls.	3	CO4

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10

BOS	**Mathematics, Statistics and Computer Application**				
Course	**Data Structures**				
Course Code	**HUSIT210**	**Level**	**5.0**		
			Theory	**Practical**	**Total**
Semester	**III**	**Credit**	02	01	03
Type	**Major**	**No of Teaching hours**	**30**	**30**	**60**

Evaluation/ Assessment	**Total Marks**	**Semester End**	**Continuous Evaluation**	**Practical Examination**
	150	**50**	**50**	**50**

Learning Objectives	
1	To introduce fundamental concepts of data structures, abstract data types, and algorithms.
2	To develop understanding of linear data structures such as arrays, linked lists, stacks, and queues.
3	To provide knowledge of non-linear data structures including trees and graphs.
4	To enable students to learn sorting, searching, and hashing techniques for efficient data handling.
5	To develop problem-solving skills and ability to analyze algorithm efficiency using Big-O notation.

Course Outcomes	
After successful completion of this course, students would be able to: -	
CO1	Understand fundamentals of data structures, abstract data types, and analyze algorithms using Big-O notation.
CO2	Implement linear data structures such as Arrays, Linked Lists, Stacks, and Queues and perform their basic operations.
CO3	Implement Binary Trees and Graphs and perform traversals like DFS and BFS.
CO4	Apply sorting, searching, and hashing techniques for efficient data management.

Modules at Glance

Module No.	**Content**	**No. of Hours**	**CO Mapping**
1	Basic Concepts and Linear Data Structures	15	CO1,CO2,CO3
2	Non-Linear Data Structures, Sorting, Searching, Graphs and Hashing	15	CO4,CO5

Syllabus

Module No.	Content	No. of Lectures
1	**Introduction to Data Structures & Analysis of Algorithms:** Introduction to the Theory of Data Structures, Advantages of Data Structures, Operations of Data Structures, Abstract Data Types, Classification of Data Types: Primitive Data Types, Non - Primitive Data Types. Algorithms, Characteristics of an algorithm, Factors of an algorithm, Complexity, Big O notation, Algorithm Analysis. **Arrays & Linked Lists:** Introduction to Linear and Non- Linear Data Structures, One-Dimensional Arrays, Array Operations, Two- Dimensional arrays. Introduction to Lists and Linked Lists, Dynamic Memory Allocation, Basic Linked List Operations, Doubly Linked List, Circular Linked List. **Stacks & Queues:** Introduction to Stacks, Representation of Stacks through Arrays, Representation of Stacks through Linked Lists, Applications of Stacks (infix to postfix conversion). Introduction, Representation of Queues, Circular Queues, Double Ended Queues- Deques, Priority Queues, Application of Queues.	15
2	**Binary Trees:** Introduction to Non- Linear Data Structures, Introduction Binary Trees, Types of Trees, Basic Definition of Binary Trees, Properties of Binary Trees, Representation of Binary Trees, Operations on a Binary Search Tree, Binary Tree Traversal, Applications of Binary Tree **Sorting and Searching:** Sorting – An Introduction, Bubble Sort, Insertion Sort, Merge Sort, Searching – An Introduction, Linear or Sequential Search, Binary Search. **Graphs:** Introduction to Graphs, Terms Associated with Graphs, Sequential Representation of Graphs, Linked Representation of Graphs, Traversal of Graphs: DFS & BFS, Spanning Trees, Application of Graphs. **Hashing:** Hash functions and hash tables, Collision handling: separate chaining and linear probing.	15
Case Study Scenario		
M1	A railway ticket counter serves passengers in the order they arrive. During peak hours, many passengers stand in line to book tickets. The system must ensure that the first person in line gets served first, and new passengers join at the end of the line. Sometimes, a separate fast-track line is opened for senior citizens.	
M2	A city metro network connects multiple stations through different routes. Passengers want to find all possible paths between two stations and choose the shortest or most convenient one. The system must represent stations and connections efficiently and allow traversal of routes.	

Reference Books:

1. Data Structures Using C — Reema Thareja
2. Fundamentals of Data Structures — Ellis Horowitz & Sartaj Sahni
3. Data Structures and Algorithm Analysis in C — Mark Allen Weiss
4. Schaum's Outline of Data Structures — Seymour Lipschutz
5. Data Structures Using C and C++ — Langsam, Augenstein & Tenenbaum

Semester End Evaluation (50 Marks) **Time : 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks : 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Practical Syllabus

Sr. No	List of Practical	No. of. Lectures	CO Mapping
1	Write a program to: a. Insert an element at a given position b. Delete an element from a given position c. Search an element using linear search	3	CO1,CO2
2	Write a program to: a. Insert a node at beginning, end, and given position b. Delete a node from a given position	3	CO1,CO2
3	Write a program to: a. Perform push, pop, and peek operations on stack.	3	CO1,CO2
4	Write a program to: a. Implement a queue using array b. Perform enqueue and dequeue operations c. Implement a circular queue	3	CO1,CO2
5	Write a program to: a. Insert nodes into BST b. Search for a node in BST	3	CO4.CO5
6	Write a program to: a. Perform pre-order traversal b. Perform in-order traversal c. Perform post-order traversal	3	CO4.CO5
7	Write a program to implement: a. Bubble sort b. Insertion sort c. Merge sort	3	CO4.CO5
8	Write a program to implement: a. Linear search b. Binary search (on sorted array)	3	CO4.CO5
9	Write a program to: a. Perform Depth First Search (DFS) b. Perform Breadth First Search (BFS)	3	CO4.CO5

10	Write a program to: a. Handle collisions using separate chaining or linear probing	3	CO4.CO5

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10

BOS	**Mathematics, Statistics and Computer Application**				
Course	**R Programming for Data Science**				
Course Code	**HUSIT211**	**Level**	**5.0**		
			Theory	**Practical**	**Total**
Semester	**III**	**Credit**	02	01	03
Type	**Minor**	**No of Teaching hours**	30	30	60

Evaluation/ Assessment	**Total Marks**	**Semester End**	**Continuous Evaluation**	**Practical Examination**
	150	**50**	**50**	**50**

Learning Objectives	
1	To provide students with a strong foundation in R programming basics, including its user interface, objects, functions, and scripts.
2	To help students master the use of R packages and help pages to carry out data analysis tasks more efficiently.
3	To enable students to work with R objects such as atomic vectors, matrices, arrays, lists, data frames, and understand loading and saving data for effective analysis, with a focus on efficient data manipulation using the dplyr package.
4	To teach students how to create informative graphs and plots using the ggplot package.

Course Outcomes	
CO1	Students will have the ability to write their own functions, manage arguments and understand scripts in the R programming language.
CO2	Students will proficiently use packages and help pages in R programming for the better resolution of the challenges of data analysis
CO3	Students will gain a strong foundation in R objects, different data frames, working with loading data, saving data and will be able to efficiently manipulate data frames using dplyr.
CO4	Students will be able to work efficiently on informative and visually appealing plots using the ggplot2

Modules At Glance

Module No.	Content	No. of Hours	Mapping with CO
1	Basics Of R Programming	15	1,2
2	Working with R	15	3,4,5
		30	

Syllabus

Module No.	Content	No. of Hours
1	R Basics: The R User interface, objects, functions, sample with replacement writing own functions arguments scripts. Packages and Help pages: Packages, Help packages. R Objects: Atomic Vectors, Attributes, Matrices, Arrays, Class, Coercion, Lists, data frames, loading data, saving data. Modifying Values: Changing values in place, logical subsetting, missing information. Environments: Introduction, working with environments, scoping rules, assignment, evaluation, closures.	15
2	Programs: Strategy, if statements, else statements, lookup tables, code comments. Loops: Expected values, expand.grid, for loops, while loops, repeat loops. Speed: Vectorized code, how to vectorize? writing fast loops, vectorized code in practice. R for data science: Data Manipulation with dplyr (Filtering, selecting, mutating, arranging, summarizing, grouping data frames). Data Visualization with ggplot2: Introduction to ggplot2, creating scatter plots, line plots, bar charts, histograms, box plots, customizing plots.	15
Case Study Scenario		
M1	**Academic Performance and Behavioral Analysis** **Scenario:**A university wants to understand the relationship between student attendance, assignment submission patterns, and final exam scores across different departments.	
M2	**Retail Sales and Regional Trend Analysis** **Scenario:** A retail chain needs to analyze its sales performance across different regions (North, South, East, West) and product categories to prepare for an end-of-year board meeting.	

Reference Books:

1. Hands-On Programming with R, Garrett Grolemund, O'Reilly, 2014
2. R Programming: A Step-by-Step Guide for Absolute Beginners. Daniel Bell, Guzzler Media, 2020
3. R in Action (3rd Edition) by Robert Kabacoff.
4. The Art of R Programming by Norman Matloff
5. Advanced R (2nd Edition) by Hadley Wickham.

Semester End Evaluation (50 Marks) **Time : 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks : 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Syllabus

Sr. No	List of Practical	No. of. Lectures	CO Mapping
1	**R Basics and Data Types** a) Write R code to create a vector of student names and another vector of their corresponding scores in a class test. Combine these into a data frame and display the first 5 rows. b) Create a list containing a student's name, ID, and a vector of their marks in three different subjects. Access and print the student's name and their marks in the second subject.	3	CO3
2	**R Functions** a) Write an R function to calculate the factorial of a given number. Demonstrate its use by calculating the factorial of 5 and 7. b) Explore the use of default argument values in R functions. Create a function with some arguments having default values and demonstrate how to call the function with and without specifying those arguments.	3	CO1
3	**Working with Matrices and Arrays** a) Create two matrices, A and B, of size 3x3 with random numbers. Perform matrix addition, subtraction, and multiplication. Calculate the transpose of matrix A. b) Create a 3x3x3 array and demonstrate how to access specific elements and slices of the array.	3	CO3
4	**Control Structures - If/Else and Loops** a) Write an R program to determine if a number is prime or not. b) Use a for loop to iterate through a vector of numbers and print whether each number is even or odd.	3	CO1
5	**Working with Data Frames** a) Load a CSV file into an R data frame. Display the structure of the data frame (number of rows, columns, data types). b) From the loaded data frame, extract all rows where a specific column (e.g., "Salary") is above a certain threshold (e.g., 50000) and save the result to a new CSV file.	3	CO3
6	**Working with dplyr** Using the dplyr package, perform the following operations on the student data frame: 1. department. 2. Select the Name, Department, and MathScore columns.	3	CO3

	3. Create a new column called AverageScore that calculates the average of MathScore, ScienceScore, and EnglishScore for each student. 4. Arrange the data frame in descending order of AverageScore. 5. Group the data by Department and calculate the average MathScore for each department.		
7	**List Manipulation** a) Create a nested list representing a hierarchical structure (e.g., a company organization). Access and modify elements at different levels of the list. b) Write a function that takes a list of numeric vectors as input and returns a new list containing the mean of each vector.	3	CO3
8	**R for Data Science Basics** a) Given a vector of numerical data, calculate the mean, median, and standard deviation using R functions. b) Create a simple scatter plot in R using two numerical vectors of the same length. Label the axes appropriately.	3	CO2, CO4
9	**Text Manipulation** Performing Text Manipulation using str_sub(),str_trim() & str_to_lower(),str_detect()str_replace_all()str_extract()str_pad()str_split_fixed()str_c() or str_glue() and str_split() in R	3	CO2
10	**Data Aggregation** a) Using a sales data frame with columns like "Region", "Product", and "Sales", calculate the total sales for each region. b) From the same sales data frame, calculate the average sales for each product within each region.	3	CO3

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10

BOS	**Mathematics, Statistics and Computer Application**				
Course	**MEAN Stack Development**				
Course Code	**HUSIT212**	**Level**	**5.0**		
			Theory	**Practical**	**Total**
Semester	**III**	**Credit**	02	02	04
Type	**SEC**	**No of Teaching hours**	**30**	**30**	**60**

Evaluation/ Assessment	**Total Marks**	**Semester End**	**Continuous evaluation**	**Practical Examination**
	150	**50**	**50**	**50**

Learning Objectives	
1	To understand the fundamentals of NoSQL databases and develop skills in designing and managing data using MongoDB with Mongoose.
2	To learn backend application development using Express.js including RESTful API design, middleware usage, and error handling.
3	To understand server-side programming concepts using Node.js, including asynchronous programming and request handling.
4	To develop dynamic and interactive frontend applications using Angular, including routing, forms, validation, and integration with backend services.

Course Outcomes	
CO1	Design and implement NoSQL databases by creating collections, schemas, and performing CRUD operations using MongoDB and Mongoose.
CO2	Develop backend applications and RESTful APIs using Express.js, incorporating middleware, routing, and error handling techniques.
CO3	Build server-side applications using Node.js by applying asynchronous programming, modularization, and HTTP server concepts.
CO4	Create responsive frontend applications using Angular and integrate them with backend APIs to develop complete full-stack applications.

Modules At Glance

Module No.	**Content**	**No. of Hours**	**Mapping with CO**
1	MongoDB & Mongoose (Introduction to NoSQL, installation using MongoDB Compass, collections, documents, CRUD operations, schema design, validation, indexing), Express.js Framework (Introduction to Express.js, middleware, RESTful APIs, routing, error handling, environment configuration)	15	CO1, CO2
2	Node.js Fundamentals (Introduction to Node.js, npm, modules, event loop, HTTP server, routing), Angular Framework (Introduction to Angular, components, data binding, directives, services, routing, forms, validation, HTTP client, observables, CRUD integration)	15	CO3, CO4

		30	

Syllabus

Module No.	Content	No. of Hours
1	**MongoDB & Mongoose (Database Layer)** Introduction to NoSQL and MongoDB, installation and setup using MongoDB Compass, understanding collections and documents, performing CRUD operations using Mongo Shell and Compass, introduction to Mongoose, defining schemas and models, data validation techniques, indexing basics **Express.js Framework (Application Layer)** Introduction to Express.js, creating Express applications, understanding middleware concepts including built-in, custom and third-party middleware, RESTful API design using GET, POST, PUT and DELETE methods, routing and use of router modules, error handling mechanisms, environment configuration using dotenv	15
2	**Angular Framework (Frontend Development)** Introduction to Angular and Angular CLI, understanding Angular project structure, components, templates and styling, one-way and two-way data binding, structural directives including ngIf and ngFor, services and dependency injection, Angular routing including routerLink, routerOutlet and lazy loading, template-driven and reactive forms, form validation and error handling, HTTP client for API integration, observables and async pipe, basic CRUD operations with backend integration **Node.js Fundamentals (Server Runtime)** Introduction and installation of Node.js, understanding npm and package.json, working with node modules and module exports, event loop and asynchronous programming concepts, creating basic HTTP server, handling requests and routing, introduction to backend integration concepts	15
Case Study Scenario		
M1	A college library wants to develop a backend system to manage book records, members, and issue/return transactions. The system should allow the librarian to add new books, update book details, delete records of lost or outdated books, and view all available books. It should also manage member details and track which books are issued and returned. The backend of the system is to be developed using Express.js, while data will be stored in MongoDB using Mongoose for schema design and validation. Using Module 1 concepts, explain how you would design and implement the system including creating collections for books, members, and transactions, defining schemas and models using Mongoose, implementing CRUD operations through RESTful APIs (GET, POST, PUT, DELETE), using middleware for request processing and validation, handling errors effectively in Express, organizing routes using router modules, and configuring environment variables using dotenv for database connection and server settings.	

M2	A college wants to develop an Online Feedback System where students can submit feedback for courses and faculty members. The system should provide a user-friendly interface for submitting feedback and allow administrators to view and analyze responses. The frontend will be developed using Angular, and the server-side runtime will use Node.js to handle requests and communicate with backend APIs. Using Module 2 concepts, describe how you would design and implement the system including creating Angular components for feedback forms and dashboards, setting up Angular routing for navigation between pages, implementing template-driven or reactive forms with proper validation, using services and HttpClient to send and retrieve data from backend APIs, handling asynchronous operations using observables and async pipe, creating a basic Node.js server to handle requests, managing API integration, handling errors in HTTP communication, and organizing the application for scalability and maintainability.

Reference Books:

1. Web Development with MongoDB and Node.js by Mike Wilson, published by O'Reilly Media, 2nd Edition, 2020.
2. Express in Action by Evan Hahn, published by Manning Publications, 1st Edition, 2016.
3. Learning Angular by Aristeidis Bampakos and Pablo Deeleman, published by O'Reilly Media, 3rd Edition, 2022.
4. MongoDB: The Definitive Guide by Shannon Bradshaw, Eoin Brazil and Kristina Chodorow, published by O'Reilly Media, 3rd Edition, 2019.
5. Pro Git by Scott Chacon and Ben Straub, published by Apress, 2nd Edition, 2014.

Semester End Evaluation (50 Marks) **Time : 2 Hours**

Paper Pattern

Question No.	Questions	Total Marks : 50
Q1	Attempt 3 out of 5	15
Q2	Attempt 3 out of 5	15
Q3	Attempt 3 out of 5	15
Q4	Case Study	05

Practical Syllabus

Sr. No	List of Practical	No. of. Lectures	CO Mapping
1	Write a program to connect to MongoDB using Mongoose and create a schema and model for a collection (e.g., Student/Book)	3	CO1
2	Develop a program to perform CRUD operations (Create, Read, Update, Delete) using Mongoose	3	CO1
3	Implement data validation and indexing in MongoDB using schema validation rules in Mongoose	3	CO1

4	Write a program to create a basic HTTP server using Node.js and handle different routes (/, /about, /contact)	3	CO3
5	Develop a program demonstrating asynchronous programming in Node.js using callbacks, Promises, and async/await	3	CO3
6	Create a basic application using Express.js with routing for multiple endpoints and JSON responses	3	CO2
7	Design and implement RESTful API endpoints (GET, POST, PUT, DELETE) using Express.js integrated with MongoDB	3	CO2
8	Implement middleware (custom logger/authentication) and error handling in an Express.js application	3	CO2
9	Create an application using Angular with components, data binding, and directives (*ngIf, *ngFor) to display dynamic data	3	CO4
10	Develop a complete CRUD frontend using Angular with reactive forms, validation, and integration with backend APIs using HttpClient	3	CO4

Semester End Practical Evaluation **Time: 2 Hours**

Question No.	Questions	Total Marks
Q.1	Program	30
Q.2	Journal	10
Q.3	Viva & Attendance	10