AC –24/11/2023
Item No. – 6.2 (N)

As Per NEP 2020

University of Mumbai



Title of the program

A- U.G. Certificate in Information Technology
B- U.G. Diploma in Information Technology
C- B.Sc. (Information Technology)
D- B.Sc. (Honours) in Information Technology
E- B.Sc. (Honours with Research) in Information Technology

Syllabus for Semester –

Sem I & II

Ref: GR dated 20th April, 2023 for Credit Structure of UG

(With effect from the academic year 2024-25 Progressively)

University of Mumbai



Syllabus for Approval

(As per NEP 2020)

Sr. No.	Heading	Particulars	
1	**Title of program** **O. _____A**	A	Title of the program U.G. Certificate in Information Technology
	O. _____B	B	U.G. Diploma in Information Technology
	O. _____C	C	B.Sc. (Information Technology)
	O. _____D	D	B.Sc. (Honours) in Information Technology
	O. _____E	E	B.Sc. (Honours with Research) in Information Technology
2	**Eligibility** **O. _____A**	A	10+2 (A learner must have completed HSC or equivalent with 45% of aggregate for open category and 40% of aggregate in case of reserved candidates in one attempt with Mathematics and/or Statistics as one of the subjects **(OR)** Passed Equivalent Academic Level 4.0 with CGPA equivalent to 45% for open category and 40% in case of reserved candidates with Mathematics and/or Statistics as one of the subjects
	O. _____B	B	Under Graduate Certificate in Information Technology Academic Level 4.5
	O. _____C	C	Under Graduate Diploma in Information Technology Academic Level 5.0
	O. _____D	D	Bachelors of Science in Information Technology with minimum CGPA of 7.5 Academic Level 5.5
	O. _____E	E	Bachelors of Science in Information Technology with minimum CGPA of 7.5 Academic Level 5.5
3	**Duration of program** **R. _____**	A	One Year
		B	Two Years
		C	Three years
		D	Four years

		E	Four years
4	**Intake Capacity** **R. _____**		
5	**Scheme of Examination** **R. _____**	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination	
6	**Standards of Passing** **R. _____**	40% in each component	
7	**Sem. I & II Credit Structure** **R: _____A** **R: _____B** **Sem. III & IV Credit Structure** **R: _____C** **R: _____D** **Sem. V & VI Credit Structure** **R: _____E** **R: _____F**	Attached herewith	
8	**Semesters**	A	Sem I & II
		B	Sem I, II, III& IV
		C	Sem I, II, III, IV, V & VI
		D	Sem I, II, III, IV, V, VI, VII & VIII
		E	Sem I, II, III, IV, V, VI, VII & VIII
9	**Program Academic Level**	A	4.5
		B	5.0
		C	5.5
		D	6.0
		E	6.0
10	**Pattern**	Semester	
11	**Status**	New	
12	**To be implemented from Academic Year Progressively**	From Academic Year: 2023-24	

Sign of Chairperson
Dr. Mrs. R. Srivaramangai
Ad-hoc BoS (IT)

Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sign of Offg. Dean,
Prof. Shivram S. Garje
Faculty of Science & Technology

Preamble

1) Introduction

Information technology (IT) continues to be a dynamic and rapidly evolving field with high demand for skilled professionals. The demand for IT workers is driven by various factors, and the landscape may have evolved over a period of time. NEP envisages the multidisciplinary approach thus making IT much more applicable in all fields of life. This facilitates multi-institutional mobility of the students within India as well as abroad thus making the students attain different proficiency levels right from certificate to B.Sc Honours with Research. This new syllabus under NEP will thus enables the students for higher education, research and career in the field of IT

2) Aims and Objectives

The aims and objectives of a Bachelor of Science (B.Sc) program in Information Technology (IT) generally revolve around providing students with a comprehensive understanding of the principles, technologies, and applications within the field of information technology. The entire program collectively aim to produce graduates who are well-rounded IT professionals, capable of contributing to the design, development, and management of information technology systems in various industries. The specific details of the curriculum may vary among institutions offering B.Sc in Information Technology programs.

3) Learning Outcomes

The B. Sc. (Information Technology) Programme shall prepare and enable the graduates to:

- ✓ Demonstrate proficiency in programming languages, Data structures, Design and implement software solutions with their technical competence
- ✓ Analyze user requirements and design effective IT systems or applications.
- ✓ Apply system analysis and design methodologies to address complex business challenges.
- ✓ Acquire the skills of Database Management, Networking and Security, Web Technologies
- ✓ Plan, execute, monitor, and control IT projects.
- ✓ Analyze and solve complex IT problems using critical thinking skills.
- ✓ Apply concepts of artificial intelligence, machine learning, cloud computing, and IoT
- ✓ Effectively communicate technical information both orally and in writing.

4) Any other point (if any)

PROGRAMME SPECIFIC OUTCOMES (PSO)

On completing the B. Sc.(Information Technology) at the University of Mumbai, the graduates shall be able to

- Technical Proficiency:
  - Demonstrate a comprehensive understanding of fundamental concepts, principles, and technologies in information technology.
  - Apply programming and software development skills to design and implement IT solutions.
- System Thinking and Analysis:
  - Apply system analysis and design methodologies to analyze and address

complex problems.
    - Design and develop IT systems that meet user requirements and organizational needs.
- Database Management:
    - Design, implement, and manage relational databases to store and retrieve information effectively.
    - Demonstrate proficiency in using database management systems and querying languages.
- Networking and Security:
    - Understand and implement computer networks, protocols, and security measures.
    - Evaluate and implement security solutions to protect information systems.
- Web Technologies:
    - Develop web applications using a variety of technologies and programming languages.
    - Design and create user interfaces that adhere to web design principles.
- Project Management:
    - Apply project management principles to plan, execute, and deliver IT projects.
    - Demonstrate the ability to work effectively within project teams.
- Emerging Technologies:
    - Stay informed about and adapt to emerging technologies in the IT field.
    - Apply concepts of artificial intelligence, machine learning, cloud computing, and IoT to solve real-world problems.
- Critical Thinking and Problem-Solving:
    - Analyze and solve complex IT problems using critical thinking skills.
    - Apply problem-solving strategies to troubleshoot and resolve technical issues.
- Communication Skills:
    - Effectively communicate technical information to diverse audiences, both orally and in writing.
    - Collaborate with team members and stakeholders to achieve common goals.
- Ethics and Professionalism:
    - Demonstrate ethical behavior and professionalism in all aspects of the IT profession.
    - Adhere to ethical standards and legal considerations related to information technology.

5) Credit Structure of the Program (Sem I, II, III, IV, V and VI)
Under Graduate Certificate in Information Technology

(Credit Struture Sem I & II)

	R: _____A									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC, RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	I	**6**		**-**	**2+2**	**VSC:2, SEC:2**	**AEC:2, VEC:2, IKS:2**	**CC:2**	**22**	**UG Certificate 44**
		• Programming with C - 02 • Database Management Systems - 02 • Practical I - 02				**VSC :** Combinational and Sequential Design- 02 **SEC – 02** Office Tools for Data Management **OR** Fundamentals of Telecommunication Systems				
	R: _____B									
	II	**6**		**2**	**2+2**	**VSC:2, SEC:2**	**AEC:2,VEC:2**	**CC:2**	**22**	
		• OOPs with C++ - 02 • Web Designing - 02 • Practical II - 02				• VSC : Assembly Language Programming – 02 • SEC: 02 • Web Programming **OR** • PL/SQL				
	Cum Cr.	**12**	**-**	**2**	**8**	**8**	**10**	**4**	**44**	
Exit option: Award of UG Certificate in Major with 40-44 credits and an additional 4 credits core NSQF course/ Internship OR Continue with Majorand Minor										

Under Graduate Diploma in Information Technology

Credit Structure (Sem. III & IV)

	R: _____C									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC, RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	III	**8**		**4**	**2**	**VSC:2**	**AEC:2,**	**FP :2 CC:2**	**22**	
		• Python Programming -02 • Python Programming Practical-02 • Data Structures-02 • Data Structures Practical-02				VSC : Operating Systems-02				**UG Diploma 88**
	R: _____D									
	IV	**6**		**4**	**2**	**SEC:2**	**AEC: 2**	**CEP : 2 CC: 2**	**22**	
		• Core Java - 02 • Core Java Practical-02 • Software Engineering-02 • Software Engineering Practical-02				• Computer Graphics and Animation -02 **OR** • Mojo-02 **OR** • Mobile Programming-02				
	Cum Cr.	28		10	12	12	14	12	88	
Exit option; Award of UG Diploma in Major and Minor with 80-88 credits and an additional 4 credits core NSQF course/ Internship OR Continuewith Major and Minor										

B.Sc. (Information Technology)

Credit Structure (Sem. V & VI)

	R: _____E									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC, RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
	V	**10**	**4**	**4**		**VSC: 2**		**FP/CEP:2**	**22**	
		• Advanced Web Programming-02 • Advanced Web Programming Practical-02 • Business Intelligence-02 • Business Intelligence Practical-02 • Software Project Management-02	• Linux Administration-02 • Linux Administration Practical-02 OR • EARN-02 • EARN Practical-02 OR • Enterprise Java-02 Enterprise Java Practical-02			• Advanced Mobile Programming-02		FP: Project Dissertation-02	22	**UG Degree 132**
	R: _____E									
	VI	**10**	**4**	**4**				**OJT:4**	**22**	
		• Security in Computing -02 • Security in Computing Practical-02 • AI and ML-02 • AI and ML Practical-02 • Software Quality Assurance-02	• Enterprise Networking-02 • Enterprise Networking Practical-02 OR • Principles of GIS-02 • Principles of GIS Practical-02					• OJT: Project Implementation-04		
	Cum Cr.	**48**	**8**	**18**	**12**	**14**	**14**	**18**	**132**	
Exit option: Award of UG Degree in Major with 132 credits OR Continue with Major and Minor										

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

SEMESTER I

Syllabus
B.Sc. (Information Technology)
(Sem.- I)

Major Courses

Name of the Course: Programming with C

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course allows the students to understand the fundamental concepts of programming which will allow them to program applications in C.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To understand the concepts of computer programming. CO 2. To understand syntax and semantics of the C language CO 3. To understand loops and decision making in programming. CO 4. To understand the use of arrays, structures, union and pointers. CO 5. To understand functions for modular code and handle errors.	
8	**Course Outcomes (OC):** OC 1. Students can build flowcharts, pseudocode for C programs. OC 2. Students can use C language syntax and semantics in their programs. OC 3. Students can implement loops and decision making. OC 4. Students can use different types of data structures in their programs. OC 5. Students can write well-structured, readable, and maintainable C code and debug programs if there are any errors.	
9	**Modules:-** **Module 1:**	
	1. Introduction: Algorithms, History of C, Structure of C Program. Program Characteristics, Compiler, Linker and preprocessor, pseudo code statements and flowchart symbols, Desirable program characteristics. Program structure. Compilation and Execution of a Program, C Character Set, identifiers and keywords, data types and sizes, constants and its types, variables, Character and character strings, typedef, typecasting **2. Type of operators**: Arithmetic operators, relational and logical operators, Increment and Decrement operators, assignment operators, the conditional operator, Assignment operators and expression, Precedence and order of Evaluation Block Structure, Initialization, C Preprocessor	**15 Hrs**
	Module 2:	

	1. Control Flow: Statements and Blocks, If-Else, Else-If, Switch, Loops- While and For Loops Do-while, Break and Continue, Goto and Labels **2.** Basics of functions. User defined and Library functions **3.** Pointer and Addresses, Pointer and Function Arguments, Pointer and Arrays. **4.** User-defined data types- structure and union	**15 Hrs**
10	**Books and References:** 1. C Programming Language, Brian W. Kernighan, Dennis M. Ritchie , 2017 2. Let Us C, Yashvant Kanetkar, BPB Publications,2008. 3. Mastering in C, K. R. Venugopal and Sudeep R. Prasad, Tata McGraw-Hill Publications. 4. A Computer Science –Structure Programming Approaches using C, Behrouz Forouzan, Cengage Learning. 5.. Schaum's outlines Programming with C, Byron S. Gottfried, Tata McGraw- Hill Publications. 6. Basics of Computer Science, by Behrouz Forouzan, Cengage Learning. 7. Programming Techniques through C, by M. G. Venkateshmurthy, Pearson Publication.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Database Management System

Sr.No.	Heading	Particulars	
1	**Description the course : Including but Not limited to:**	The objective of the course is to present an introduction to fundamentals of database management systems, with an emphasis on how to organize, maintain and retrieve - efficiently, and effectively -information from a DBMS.	
2	**Vertical :**	Major	
3	**Type :**	Theory	
4	**Credits:**	2 credits (1 credit = 15 Hours for Theory)	
5	**Hours Allotted :**	30 Hours	
6	**Marks Allotted:**	50 Marks	
7	**Course Objectives(CO):** CO 1. To make students aware fundamentals of database system. CO 2. To give idea how ERD components helpful in database design and implementation. CO 3. To experience the students working with database using MySQL. CO 4. To familiarize the student with normalization, database protection and different DDL, DML, DQL, DCL Statements CO 5. To make students aware about importance of protecting data from unauthorized users.		
8	**Course Outcomes (OC):** OC 1. Define and describe the fundamental elements of relational database management system. OC 2. To relate the basic concepts of relational data model, entity-relationship model, relational database OC 3. Design ER-models to represent simple database application scenarios. OC 4. Understand the normalization and its role in the database design process OC 5. Transform the ER-model to relational tables, populate relational database and formulate SQL OC 6. Understand basic database storage structures and access techniques: file and page organizations, indexing methods and hashing.		
9	**Modules:-** **Module 1:**		
	1. Introduction to Databases and transactions What is database system, purpose of database system, view of data, relational databases, database architecture, transaction management **2. Data Models** The importance of data models, Basic building blocks, Business rules, The evolution of data models, Degrees of data abstraction **3. Database Design, ER-Diagram** Database design and ER Model: overview, ER-Model, Constraints, ER-Diagrams, ERD Issues, Codd's rules, Relational Schemas **4. Relational database model** Logical view of data, keys, integrity rules		**15 Hrs**

	Module 2:	
	1. Database Design theory and normalization: Basics of functional dependencies and normalization for relational databases. Relational database design and further dependencies. **2. SQL, Indexing:** Introduction to SQL, Complex queries, triggers, views, joining database tables and schema modification. Query Processing and optimization. File structure, hashing and indexing **3. Transaction management and concurrency control and recovery:** Introduction to transaction processing concepts and theory. Concurrency control technique. Database recovery technique	. **15 Hrs**
10	**Text Books** 1. "Fundamentals of Database System", Elmasri Ramez, Navathe Shamkant, Pearson Education, Seventh edition, 2017 2. Database Management Systems", Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, 2014 3. Database Systems: Design implementation and management by Carlos Coronel, Steven Morris, Peter Rob	
11	**Reference Books** 1. "Database System Concepts", Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw Hill, 2017 2. "MySQL: The Complete Reference", Vikram Vaswani , McGraw Hill, 2017 3. "Learn SQL with MySQL: Retrieve and Manipulate Data Using SQL Commands with Ease", Ashwin Pajankar, BPB Publications, 2020	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Major Practical 1

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Programming with C -practical This course is stepping stone to learn other languages. This course provides students hands on experiences of coding exercises and projects. . Database Management System's practical approach is useful to gain the knowledge for software backend development. It benefits to user by providing data definition, data access, reduced data redundancy, data integrity, data sharing, data organizing, data consistency, data accuracy, and security
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (60 Hours of Practical work in a semester)
5	**Hours Allotted :**	30 Hours (C Programming Practical) + 30 Hours(DBMS - Practical)
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To provide exposure in developing algorithm, flowchart and to write efficient code. CO 2. To understand loops and decision making in programming. CO 3. To understand the arrays, structures, union. CO 4. To understand the use of function and pointers. CO 5. To Identify entities and its relationship with relational model structure. CO 6. To understand relational database using SQL and constraints implementation using create table queries. CO 7. To Understand DML operations and backing of database CO 8. To understand how to retrieve data from database and learn how to retrieve single value after performing calculations on group of values CO 9. To understand built-in functions to perform operations on data CO 10. To understand how to fetch data from two or more tables, which is joined to appear as single set of data CO 11. To understand nested and larger query as advanced fetching of data to understand concept of virtual table. CO 12. To understand how to control user access in a database.	

8	**Course Outcomes (OC):** OC 1. Students can demonstrate the concepts of datatypes, variables and operators in C. OC 2. Students can implement the concept of control statements and looping in C program. OC 3. Students can demonstrate the use of arrays, strings and structures in C OC 4. Students can implement modular C program using functions and pointers. OC 5. Students can demonstrate the use of arrays, strings and structures in C. OC 6. Students able to perform various operations such as insert, update delete and retrieve data from database using SQL queries. OC 7. Students able to perform alteration in tables and can restore and take backup of the database. OC 8. Students able to perform operations using simple SQL Queries to fetch data and learns various aggregate functions to get single value. OC 9. Students able to perform SQL Queries using JOIN keyword for joining two or more tables. OC 10. Students able to perform nested queries using in, exists operators. OC 11. Students able to create new table by joining one or more tables and learn how to hide attribute from end user. OC 12. Students able to restrict the user from accessing data in database. OC 13. Students should be able to create, manipulate the database management system to evaluate the business information problem.

	9	**Module 1:- Programming with C** **1. Practical 1**:- a. To calculate simple interest taking principal, rate of interest and number of years as input from user. Write algorithm & draw flowchart for the same. b. Write a program to find greatest of three numbers using conditional operator. Write algorithm & draw flowchart for the same. c. Write a program to check if the year entered is leap year or not. Write algorithm & draw flowchart for the same. **2. Practical 2:-** a. Write a program to calculate roots of a quadratic equation. b. Write a menu driven program using switch case to perform add / subtract / multiply / divide based on the users choice. c. Write a program to print the pattern of asterisks. **3. Practical 3** a. Write a program using while loop to reverse the digits of a number. b. Write a program to calculate the factorial of a given number. c. Write a program to print the Fibonacci series. **4. Practical 4** a. Write a program to print area of square using function. b. Write a program using recursive function. c. Write a program to square root, abs() value using function. d. Write a program using goto statement . **5. Practical 5** a. Write a program to print rollno and names of 10 students using array. b. Write a program to sort the elements of array in ascending or descending order **6. Practical 6** a. Write a program to extract the portion of a character string and print the extracted part. b. Write a program to find the given string is palindrome or not. c. Write a program to using strlen(), strcmp() function . **7. Practical 7** Write a program to swap two numbers using a function. Pass the values to be swapped to this function using call-by-value method and call-by-reference method. **8. Practical 8** a. Write a program to read a matrix of size m*n. b. Write a program to multiply two matrices using a function. **9. Practical 9** Write a program to print the structure using Title Author Subject Book ID Print the details of two students. **10. Practical 10** Create a mini project on “Bank management system”. The program should be menu driven.	**30 Hrs**

		Module 2 1. Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.) 2. Perform the following: • Viewing all databases • Creating a Database • Viewing all Tables in a Database • Creating Tables (With and Without Constraints) • Inserting/Updating/Deleting Records in a Table 3. Perform the following: • Altering a Table • Dropping/Truncating/Renaming Tables • Backing up / Restoring a Database 4. Perform the following: • Simple Queries • Simple Queries with Aggregate functions 5. Queries involving • Date Functions • String Functions • Math Functions 6. Join Queries • Inner Join • Outer Join 7. Subqueries • With IN clause • With EXISTS clause 8. Converting ER Model to Relational Model and apply Normalization on database. (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys and normalization up to 3rd Normal Form). 9. Views • Creating Views (with and without check option) • Dropping views • Selecting from a view 10. DCL statements • Granting and revoking permissions • Saving (Commit) and Undoing (rollback)	**30 Hrs**
10	**Text Books:** 1.“Fundamentals of Database System”, Elmasri Ramez, Navathe Shamkant, Pearson Education, Seventh edition, 2017 . 2.Database Management Systems”, Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, 2014		
11	**Reference Books:** 1. MASTERING C, K. R. Venugopal and Sudeep R. Prasad, Tata McGraw-Hill Publications. 2. “A Computer Science –Structure Programming Approaches using C”, Behrouz		

	Forouzan, Cengage Learning. 3. Schaum's outlines "Programming with C", Byron S. Gottfried, Tata McGraw-Hill Publications. 4. "Basics of Computer Science", Behrouz Forouzan , Cengage Learning. 5. "Programming Techniques through C", M. G. Venkateshmurthy, Pearson Publication. 6. "Programming in ANSI C", E. Balaguruswamy, Tata McGraw-Hill Education. 7. "MySQL: The Complete Reference", Vikram Vaswani , McGraw Hill, 2017. 8. "Learn SQL with MySQL: Retrieve and Manipulate Data Using SQL Commands with Ease", Ashwin Pajankar, BPB Publications, 2020.	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Vocational Skill Course (VSC)

Name of the course: Combinational and Sequential Design

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Combinational and Sequential Design is a course that focuses on digital electronics and the design of circuits that combine multiple digital components. The course covers the theoretical and practical aspects of both combinational and sequential circuit design, as well as their applications. Digital circuits are used in many electronic devices, including computers, smartphones, and communication systems. The design of these circuits is critical to the performance and functionality of these devices. Understanding the basics of combinational and sequential design is essential for anyone interested in pursuing a career in the field of digital electronics. The course will cover the various techniques and tools used in digital circuit design, including Boolean algebra and K-map simplification. The course is highly relevant in today's technological landscape, as all modern electronics devices are based on digital circuits. The skills learned in the course are highly useful in various fields, such as computer and electronics engineering, telecommunications, and robotics. The application of combinational and sequential design is quite broad, and the skills acquired from the course can be applied in various areas. Students will be able to design digital circuits, troubleshoot and repair digital circuits, and optimize circuit performance. The course is highly interesting and engaging, providing students with the opportunity to explore and analyze complex digital circuitry. It is also connected to other courses such as Digital Logic Design, Computer Organization, and Microcontrollers. The demand for professionals with digital circuit design skills is high in various industries such as electronics, semiconductors, telecommunications, and computing. There is an increasing demand for professionals with these skills,

		and job prospects are promising for those with a solid background in digital circuit design. In summary, Combinational and Sequential Design is a course that offers students a comprehensive understanding of digital circuits' design principles and techniques. The knowledge and skills gained from this course are highly useful and applicable in various industries, with promising career prospects.
2	**Vertical :**	Vocational Skill Course(VSC)
3	**Type :**	Practical
4	**Credits :**	2 credits (60 hours in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To provide students with a comprehensive understanding of combinational and sequential circuit design principles and techniques. CO 2. To enable students to apply Boolean algebra, K-map simplification, and other design techniques to create optimized digital circuits. CO 3. To equip students with the necessary tools and skills to implement arithmetic circuits, data path circuits, and memory circuits. CO 4. To enable students to analyze and troubleshoot digital circuits to ensure optimal performance. CO 5. To provide students with hands-on practical experience in designing and implementing digital circuits using simulation software and real-world hardware.	
8	**Course Outcomes (OC):** OC 1. Students can explain the differences between combinational and sequential circuits, and identify their different applications. OC 2. Students can define the concept of Boolean algebra and its importance in digital circuit design. OC 3. Students can explain and apply the principles of K-map simplification and other design techniques. OC 4. Students can design and construct combinational circuits using Boolean algebra and K-maps. OC 5. Students can design and implement arithmetic circuits such as adders, subtractors, and multipliers. OC 6. Students can design and implement data path circuits such as registers, multiplexers, and decoders. OC 7. Students can implement digital circuits using breadboards, logic probes, and oscilloscopes. OC 8. Students can troubleshoot and verify the correctness of digital circuits using real-world hardware and measure their performance using various metrics.	
9	**Modules:-** **Module 1:**	

		1. Study of Logic gates and their ICs and universal gates: a. Study of AND, OR, NOT, XOR, XNOR, NAND and NOR gates b. Study of IC 7400, 7402, 7404, 7408, 7432, 7486, 74266 c. Implement AND, OR, NOT, XOR, XNOR using NAND gates. d. Implement AND, OR, NOT, XOR, XNOR using NOR gates. **2. Implement the given Boolean expressions using minimum number of gates.** a. Verifying De Morgan's laws. b. Implement other given expressions using minimum number of gates. c. Implement other given expressions using minimum number of ICs. **3. Implement combinational circuits.** a. Design and implement combinational circuit based on the problem given and minimizing using K-maps. (Various Equations, SOP, POS forms can be given) **4. Implement code converters.** a. Design and implement Binary – to – Gray code converter. b. Design and implement Gray – to – Binary code converter. c. Design and implement Binary – to – BCD code converter. d. Design and implement Binary – to – XS-3 code converter. **5. Implement Adder and Subtractor Arithmetic circuits.** a. Design and implement Half adder and Full adder. b. Design and implement BCD adder. c. Design and implement XS – 3 adder. d. Design and implement binary subtractor. e. Design and implement BCD subtractor. b. Design and implement XS – 3 subtractor.	**30 Hrs**
	Module 2:		
		6. Implement Arithmetic circuits. a. Design and implement a 2-bit by 2-bitultiplier. b. Design and implement a 2-bit comparator. **7. Implement Encode and Decoder and Multiplexer and Demultiplexers.** a. Design and implement 8:3 encoder. b. Design and implement 3:8 decoder. c. Design and implement 4:1 multiplexer. Study of IC 74153, 74157 d. Design and implement 1:4 demultiplexer. Study of IC 74139 e. Implement the given expression using IC 74151 8:1 multiplexer. f. Implement the given expression using IC 74138 3:8 decoder. **8. Study of flip-flops and counters.** a. Study of flip-flops and counters. b. Study of IC 7473. c. Study of IC 7474. d. Study of IC 7476. e. Conversion of Flip-flops.	**30 Hrs**

	f. Design of 3-bit synchronous counter using 7473 and required gates. g. Design of 3-bit ripple counter using IC 7473. **9. Study of counter ICs and designing Mod-N counters.** a. Study of IC 7490, 7492, 7493 and designing mod-n counters using these. b. Designing mod-n counters using IC 7473 and 7400 (NAND gates) **10. Design of shift registers and shift register counters.** a. Design serial – in serial – out, serial – in parallel – out, parallel – in serial – out, parallel – in parallel – out and bidirectional shift registers using IC 7474. b. Study of ID 7495. c. Implementation of digits using seven segment displays.	
10	**Text Books** 1. Digital Electronics and Logic Design, N. G. Palan, Technova	
11	**Reference Books** 1. Digital Principles and Applications, Malvino and Leach, Tata McGrawHill 2. Modern Digital Electronics, R. P. Jain, Tata McGrawHill 3. Digital Design, M. Morris R. Mano, Michael D. Ciletti, Pearson Education, 2012	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Skill Enhancement Course (SEC)

Name of the Course: Office Tools for Data Management

Sr.No.	Heading	Particulars
1	**Description the course:**	• **Introduction:** The MS Access course offers a comprehensive understanding of Microsoft's relational database management system, making it a valuable skill in today's data-driven environment. This course is designed to empower individuals with the tools needed to efficiently organize, manage, and analyse data. • **Relevance and Usefulness:** It provides practical insights into leveraging a relational database system for enhanced efficiency and organization. The MS Access course is useful for individuals seeking to enhance their data management skills. • **Applications:** With applications in various sectors, from business to research and project management, MS Access is versatile. It facilitates the creation of databases for tasks ranging from contact management to complex systems for inventory and financial analysis. • **Interest and Connection with Other Courses:** Its practical applications and user-friendly interface make it attractive to individuals looking to boost their data management skills. The MS Access course establishes a practical link with other data-related courses, offering foundational knowledge in database management. It complements courses in data analysis, business intelligence, and information systems. • **Demand in the Industry:** As businesses increasingly rely on data for decision-making, there is a growing demand for professionals skilled in database management. Proficiency in MS Access is particularly sought after in roles involving data organization, analysis, and reporting. • **Job Prospects:** Professionals completing the MS Access course are well-positioned for roles requiring efficient data management and analysis. Job prospects include positions in database administration, data analysis, and business intelligence, where MS Access proficiency is a valuable asset.
2	**Vertical :**	Skill Enhancement Course(SEC)
3	**Type :**	Practical
4	**Credits :**	2 credits
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):**	

	CO 1. Participants will grasp essential database concepts, including tables, relationships, and normalization principles. CO 2. Participants will design and construct well-organized databases in MS Access, showcasing proficiency in table design and data validation. CO 3. Participants will master the creation of complex queries in MS Access, enabling them to extract specific information efficiently. CO 4. Participants will develop expertise in crafting user-friendly forms and interfaces in MS Access, optimizing data entry processes. CO 5. Participants will generate comprehensive reports in MS Access, demonstrating skills in grouping, sorting, and presenting data for meaningful analysis.	
8	**Course Outcomes (OC):** OC 1. Participants can explain normalization importance, identify table relationships, and justify database design choices. OC 2. Participants create well-structured MS Access databases with proper relationships, data types, and normalization. OC 3. Participants execute advanced queries in MS Access, retrieving specific information based on diverse criteria. OC 4. Participants design intuitive MS Access forms, incorporating controls for an efficient and user-friendly data entry experience. OC 5. Participants produce insightful MS Access reports, organizing and presenting data effectively for analysis.	
9	**Modules:- All Practicals are based on MS Access** **Module 1:**	
	Practical 1: A. Getting familiar with MS Access Ribbon options. B. With the help of access wizard Create Database. Add 2 Tables. In each table add 5 columns of different data types. Add 10-10 entries in each table. Add necessary integrity constraints. C. Use the Table Wizard to create a table. Add and delete fields in an existing table. Establish an input mask and validation rule for fields within a table. Switch between the Design and Datasheet views of a table. **Practical 2:** A. Create and use an Input Mask to enter the data in sample table. B. Adding records in table by using Datasheet View, using a Form and using SQL. C. Create the Employee Database with necessary table and data and then implement the following transitions: • Delete the record for Kelly Marder. • Change Pamela Milgrom's salary to $59,500. • Use the Replace command to change all occurrences of "Manager" to "Supervisor". **Practical 3:** A. Create the Bookstore database with necessary table and data and modify the database to accommodate the following:	**30 Hrs**

		i. Add the book Exploring Microsoft Office 2000 Vol II (ISBN: 013-011100-7) by Grauer/Barber, published in 1999 by Prentice Hall, selling for $45.00. ii. Change the price of Memory Management for All of Us to $29.95. iii. Delete The Presentation Design Book. B. Create a table employ with (idno, name, job, age, salary). Insert 10 records. Create a query to display the information of all managers. Create a query to display the names of employs who"s salary is >15000. C. Use the Form Wizard to create a form, Move and size controls within a form. Use the completed form to enter data into the associated table. **Practical 4:** A. Add fields to an existing table. Use the Lookup Wizard to create a combo box. Add controls to an existing form to demonstrate inheritance. Add command buttons to a form. B. Generate and use various the queries using Query Wizards. C. Generate and use various the Query with User Input. D. Demonstrate use of Expression Builder. **Practical 5:** A. Use the report wizard to create a new report. Modify an existing report by adding, deleting, and/or modifying its controls. B. Create a query containing a calculated control. Then, create report based on that query. Use the Sorting and Grouping command to add a group header and group footer to a report. C. Use action queries to modify a database. Create a crosstab query to display summarized values from a table.	
		Module 2:	
		Practical 6: A. Create and Open a database with multiple tables; Identify the one-to-many relationships within the database and to produce reports based on those relationships. B. Create and Open a database with multiple tables; Identify the one-to-one relationships within the database and to produce reports based on those relationships. C. Create and Open a database with multiple tables; Identify the Many-to-Many relationships within the database and to produce reports based on those relationships. **Practical 7:** A. Demonstrate use of look up tables. B. Use the Report Wizard to create a report having the following requirements: i. Select the LastName field from the Author table. ii. Select the Title and Price fields from the Book table. iii. Select the PubName field from the Publisher table.	**30 Hrs**

	iv. View the data by Publisher. v. Add a grouping level using LastName. vi. Sort the report by the Title field in ascending order. vii. Choose Stepped layout and Portrait orientation. viii. Type Book List as the report's title. C. Define the relationship between two tables and add a subform to a form. **Practical 8:** A. Import an Access table from an Excel workbook. Create a one-to-many relationship between tables in a database. Create a multiple-table query. B. Import external data from the Excel spreadsheet file Bookstore. i. Make sure Import the source data into a new table in the current database is selected. ii. Select the Author worksheet. iii. Make sure that First Row Contains Column Headings is selected. iv. For the AuthorID field, set the Data Type option to Long Integer and set the Indexed option to Yes (No Duplicates). v. Select Choose my own primary key and make sure the AuthorID field is selected. vi. Save the table with the name Author. C. Export data from access to various formats. **Practical 9:** A. Relationships: Create and Use Author and Book Table. i. Create a relationship between the AuthorID field in the Author table and the AuthorCode field in the Book table. Put a checkmark in the box labeled Enforce Referential Integrity. ii. Create a relationship between the PubID field in the Publisher table and the PubID field in the Book table. Put a checkmark in the box labeled Enforce Referential Integrity. B. Create a switchboard; Use the Link Tables command to associate tables in one database with the objects in a different database. C. Create an AutoExec and a Close Database macro and demonstrate the use. **Practical 10:** A. Create the College Library database find out the following: - i. Total no. of copies of books subject wise. ii. A report displays all books group by Publisher. iii. A report displays all books group by Book Title. iv. A report displays all books group by Book Edition B. Demonstrate the use of Database Splitter Wizard by splitting database. C. Make Access database as an executable-only	
10	**Online reference/Text Books** 1. https://www.quackit.com/microsoft_access/tutorial/	

	2. https://www.tutorialspoint.com/ms_access/index.htm 3. Access 2016 in easy steps, by Mike McGrath, In Easy Steps, 1st Edition, 2017 4. Relational Databases and Microsoft Access, by Ron McFadyen, 1st Edition	
11	**Reference Books** 1. MICROSOFT ACCESS 2019 by David Murray, Kendall Hunt Publishing, 1st Edition, 2020. 2. Step by Step Microsoft Access 2013, by Joyce Cox and Joan Lambert, 1st Edition, Microsoft Press, 2013 3. Access 2019 Bible, by Michael Alexander, Richard Kusleika, Wiley, 1st Edition, 2018 4. Access 2019 For Dummies, by Laurie A. Ulrich, Ken Cook, Wiley, 1st Edition, 2018	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Name of the Course: Fundamentals of Telecommunication Systems

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	The course on Fundamentals of Telecommunication Systems aims to provide an in-depth understanding of the basic concepts and theories of signals and systems, as well as their applications in the field of telecommunication engineering. The course also focuses on the latest trends in 5G technology, providing students with insights into the driver, pillars, and challenges of 5G networks. Relevance and Usefulness: The course is highly relevant to students pursuing degrees in electronics and communication engineering, as well as those interested in telecommunications engineering. By focusing on key concepts and terminologies, such as sets, mappings, functions, and systems operators, the course provides a foundation for understanding both the theoretical and

		practical aspects of signals and systems. Additionally, the course helps students understand the role of 5G technology in enabling advanced wireless communication and the internet of things (IoT), which can be useful for developing innovative applications and services. Application and Interest: By completing the course, students will be equipped to apply their knowledge and skills in a range of industries and sectors, including telecommunication, internet of things, and wireless communication. The course is also highly engaging, as it covers several fascinating topics, including wireless communication, 5G technology, and IoT, among others. Connections with Other Courses: The course has links with other courses in electronics and communication engineering, including digital signal processing, telecommunication theory and practice, mobile communication, Information Technology and internet of things. Demand in the Industry and Job Prospects: Graduates with a background in signals and systems and 5G technology are in high demand in the telecommunication industry, as there is an increasing need for professionals who can design, implement, and oversee advanced communication networks. Specializations in 5G technology and signals and systems can open up a range of job prospects, including roles such as telecommunications engineer, network architect, systems engineer, and wireless communication developer, among others. In conclusion, the course in signals and systems and 5G technology is highly relevant and useful for students pursuing degrees in electronics and communication engineering and Information Technology. The course is engaging and provides a solid foundation in key concepts and technologies, enabling students to pursue a range of job prospects within the telecommunication industry.
2	**Vertical :**	Skill Enhancement Course(SEC)
3	**Type :**	Theory
4	**Credits :**	2 credits (30 hours in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. Identify the fundamental concepts and terminologies of signals and systems theory through an introduction to sets, mappings, functions, and systems operators.	

	CO 2. Demonstrate knowledge of the properties of continuous-time signals and systems, transformations of the independent variable, properties of functions, and representation of arbitrary functions. CO 3. Demonstrate knowledge of the properties of discrete-time signals and systems, transformations of the independent variable, properties of sequences, and representation of arbitrary sequences. CO 4. Analyze the drivers for 5G technology, identify the 10 pillars of 5G, and describe the evolution of wireless communication from LTE technology to beyond 4G. CO 5. Discuss the 5G internet of things (IoT), explain networking reconfiguration and virtualization support, and identify the mobility and quality of service control in 5G networks. CO 6. Evaluate the challenges of small cells in 5G mobile networks and identify the capacity limits and achievable gains with densification.	
8	**Course Outcomes (OC):** OC 1. Identify the fundamental concepts and terminologies of signals and systems theory through an introduction to sets, mappings, functions, and systems operators. OC 2. Demonstrate knowledge of the properties of continuous-time signals and systems, transformations of the independent variable, properties of functions, and representation of arbitrary functions. OC 3. Demonstrate knowledge of the properties of discrete-time signals and systems, transformations of the independent variable, properties of sequences, and representation of arbitrary sequences. OC 4. Analyze the drivers for 5G technology, identify the 10 pillars of 5G, and describe the evolution of wireless communication from LTE technology to beyond 4G. OC 5. Discuss the 5G internet of things (IoT), explain networking reconfiguration and virtualization support, and identify the mobility and quality of service control in 5G networks. OC 6. Evaluate the challenges of small cells in 5G mobile networks and identify the capacity limits and achievable gains with densification.	
9	**Modules:-** **Module 1: Signals and Systems:** 1. **Signals and Systems:** Introduction, Signals, Systems, Why Signals and Systems? Preliminaries, Overviews, Sets, Mappings, Functions, Sequences, Abuse of notations, System operators, Basic Signal Properties. 2. **Continuous-Time Signals and Systems:** Overview, Transformations of the Independent Variable, Transformations and the Dependent Variable, Properties of functions, Elementary functions, Representation of Arbitrary Functions using elementary functions, Continuous -time systems, Properties of systems, 3. **Discrete-Time Signals and Systems:** Overview, Transformations of the independent variable, Properties of Sequences, Elementary Sequences, Representing Arbitrary Sequences Using Elementary Sequences, Discrete-Time Systems, Properties of Systems	**15 Hrs**
	Module 2: Fundamentals of 5G Networks	
	4. **Drivers for 5G:** Introduction, Historical trend of Wireless Communication, Evolution of LTE technology to beyond 4G, 5G	**15 Hrs**

	Roadmap, 10 pillars of 5G, 5G in Europe, 5G in Asia, 5G in Asia, 5G Architecture 5. **The 5G Internet:** Introduction, Internet of Things and Context-Awareness, Networking Reconfiguration and Virtualisation Support, Mobility, Quality of Service Control, Emerging Approach for Resource Over-Provisioning 6. **Small Cells for 5G Mobile Networks:** Introduction, What are small cells? Capacity Limits and Achievable Gains with Densification, Mobile Data Demand, Demand vs Capacity, Small-Cell Challenges, Conclusions and future directions	
10	**Text Books:** 1. Signals and Systems, Michael Adams, University of Victoria, 3rd Edition, 2012 2. Fundamentals of 5G Mobile Networks, Edited by Jonathan Rodriguez, Wiley Publications, 2015	
11	**Reference Books** 1. Signals and Systems, Michael Adams, University of Victoria, 3rd Edition, 2012 2. Fundamentals of 5G Mobile Networks, Edited by Jonathan Rodriguez, Wiley Publications, 2015	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

SEMESTER II

Syllabus
B.Sc. (Information Technology)
(Sem.- II)

Major Courses

Name of the Course: Object Oriented Programming using C++

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course provides students knowledge and skills to understand and implement the object oriented skills. It will help them to implement OOP solutions to real-world problems.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To explain the difference between object oriented programming and procedural programming. CO 2. To understand OOP principles to create modular, reusable, and maintainable code. CO 3. To understand the concept of polymorphism ,virtual functions,inheritance and exception handling. CO 4. To understand file handling concepts using C++.	
8	**Course Outcomes (OC):** OC 1. Students can explain the key concept of OOP and their application in software development. OC 2. Students can Design and implement classes and objects to model real-world entities. OC 3. Students can apply the concepts of polymorphism, virtual functions, inheritance and exception handling in program. OC 4. Students can apply operator overloading, runtime polymorphism, generic Programming OC 5. Students can implement file handling concepts in program	
9	**Modules:-** **Module 1:**	
	1. Object Oriented Methodology: Introduction, Advantages and Disadvantages of Procedure Oriented Languages, Application of OOPS, Principles of OOPS: Objects, Classes, Data Abstraction and Data Encapsulation, Inheritance, Polymorphism, Dynamic Binding, Message Passing. 2. Classes and Objects: Simple classes (Class specification, class members accessing), Defining member functions, passing object as an argument, Returning object from functions, friend classes, friend function.	**15 Hrs**

	3. Constructors and Destructors: Introduction, Default Constructor, Parameterized Constructor and examples, Destructors. 4. Program development using Inheritance: Introduction, Advantages provided by inheritance, choosing the access specifier, Derived class declaration, derived class constructors, class hierarchies, multiple inheritance, multilevel inheritance, hybrid inheritance.	
	Module 2:	
	5. Polymorphism: Concept of function overloading, overloaded operators, overloading unary and binary operators. 6. Virtual Functions: Introduction and need, Pure Virtual Functions, this Pointer, abstract classes, virtual destructors. 7. Exception Handling: Introduction, Exception Handling Mechanism, Concept of throw & catch with example. 8. Working with Files: Introduction, File Operations, Various File Modes, File Pointer and their Manipulation.	**15 Hrs**
10	**Text Books** 1. Object-oriented Programming C++, Hari Mohan Pandey , Laxmi Publications 2. C++ Programming: An Object-Oriented Approach, Behrouz A. Forouzan, Richard F. Gilberg , McGraw-Hill Education 3. C++ How to Program , Paul Deitel, Harvey Deitel	
11	**Reference Books** 1. Object Oriented Programming in C++ , E Balagurusamy 2. Object-Oriented Programming in C++ , Robert Lafore, Pearson Education. 3. Programming with ANSI C++ , Bhushan Trivedi 4. Demystified Object- Oriented Programming with C++, Dorothy R. Kirk	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper:** **External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Web Designing

Sr.No	Heading	Particulars
1	**Description the course : Including but Not limited to:**	The objective of Web Designing course is to provide instructions on creating and maintaining a web page for publishing on the Internet. Students will be able to use HTML editor to author pages that include text and graphics..
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To understand the fundamentals of Internet, and the principles of web design CO 2. To develop basic websites using HTML and Style Sheets. CO 3. To understand different style sheets used in web designing. CO 4. To implement JavaScript as a tool to add dynamism to static HTML pages.	
8	**Course Outcomes (OC):** OC 1. Learners will be able to use the HTML programming language OC 2. Learners will be able to execute web pages designed using HTML OC 3. Describe the concepts of World Wide Web, and the requirements of effective web design OC 4**.** List various tags in html and **use** these to **create** web page OC 5 : Gain necessary skills for designing and developing web applications	
9	**Modules:-** **Module 1:**	
	1. Introduction to HTML 5: What Is HTML? Understanding HTML Tags, Setting Up the Document Structure: Specifying the Document Type, Creating the HTML, Specifying a Page Title. Formatting Text by Using Tags: Creating Headings, Applying Bold and Italic Formatting, Applying Superscript and Subscript Formatting, Using Monospace and Preformatted Text. Using Lists and Backgrounds: Creating Bulleted and Numbered Lists, Creating Definition Lists, Inserting Special Characters, Inserting Horizontal Lines, Choosing Background and Foreground Colors. Creating Hyperlinks and Anchors- Hyperlinking to a Web Page, Creating Hyperlinking to an E-Mail Address, Hyperlinking to Other Content. **Style Sheets and Graphics**: Introduction to Style Sheets: Understanding Styles, Constructing Style Rules, Creating Styles for Nested Tags, Applying Styles to Hyperlinks, Creating and Linking to External Style Sheets.	**15 Hrs**

		Formatting Text by Using Style Sheets: Specifying a Font Family, Specifying a Font Size and Color, Applying Bold and Italics, Applying Strikethrough and Underlining, Creating Inline Spans, Adjusting Spacing Between Letters. Formatting Paragraphs by Using Style Sheets: Indenting Paragraphs, Applying a Border to a Paragraph, Specifying the Horizontal Alignment of a Paragraph, **Displaying Graphics** Selecting a Graphics Format, Preparing Graphics for Web Use, Inserting Graphics, Arranging Elements on the Page, Controlling Image Size and Padding, Hyperlinking from Graphics, Using Thumbnail Graphics, Including Alternate Text for Graphics, Adding Figure Captions 2. Page Layout and Navigation- Creating Navigational Aids , Creating a Text-Based and Graphical Navigation Bar, Creating an Image Map, Creating Tables, Specifying the Size of a Table, Specifying the Width of a Column, Merging Table Cells. Formatting Tables-Applying Table Borders, Applying Borders by Using Attributes, Applying Borders by Using Styles, Changing Cell Padding, Spacing, and Alignment. Setting Horizontal and Vertical Alignment **Creating User Forms**- Creating a Basic Form- Creating a Text Box,Special Field types for E-Mail and Web Addresses, Creating a Text Area, Creating a Submit or Clear Button, Creating Check Boxes and Option Buttons, Additional Input Types in HTML5 Incorporating Sound and Video- What's New with Audio and Video in HTML5?,Embedding Video Clips- Introducing the <video> Tag, The <embed> Tag: Your Fallback Plan, Placing a Video Clip on a Web Page. Incorporating Audio on a Web Page- Playing Audio with the <audio> Tag, Placing an Audio Clip on a Web Page	
	Module 2:		
		1. JavaScript: Introduction to JavaScript: Variable, statements, Operators, Comments, constructs, Functions, expressions, JavaScript console, Scope, Events, Strings, String Methods, Numbers, Number Methods, Dates, Date Formats, Date,Methods,Arrays, Array Methods, Booleans, Comparisons, Control Structures: Conditions, Switch, Loop For, Loop While, Break. **Operators**: Arithmetic Operators, Assignment Operators, Comparison Operators, Logical Operators, Bitwise Operators Statements: Conditional Statements – if else, switch, Loops – while, do while, for, for in, for of, Loop Control – break, continue, labels JavaScript Objects: User-defined Objects, with Keyword, Native Objects – Array, String, Date, Math, Number, RegExp, Cookies Events and Event Handlers: HTML Events, DOM Events, DOM Event Listener,onAbort, onBlur, onChange, onClick, onDblClick, onError, onFocus, onKeyDown,onKeyPress, onKeyUp, onLoad, onMouseDown, onMouseMove, onMouseOut,onMouseOver, onMouseUp, onReset, onResize, onSelect, onSubmit, onUnload 2. Basics of JQuery, JQuery selection and events, JQuery Effects, JQuery traversal and manipulation, Data attributes and templates, jQuery Plugins.	**15 Hrs**

	3. JSON – JSON: Introduction, JSON Grammar, JSON Values, JSON Tokens, Syntax, JSON vs. XML, Data Types, Objects, Arrays, Creating JSON, JSON Object, Parsing JSON, Persisting JSON, Data Interchange, JSON HTML, JSONP	
10	**Text Books** 1. Step by Step HTML5 by Faithe Wempen, Microsoft Press,2011 2. The Complete Reference HTML & CSS, Thomas A. Powell. McGrawHill, 5 th Edition,2010 3. The Complete Reference JavaScript Thomas A. Powell &Fritz Schneider McGrawHill 3rd 2012 4. Web Technologies: HTML, JAVASCRIPT, PHP, JAVA, JSP, XML and AJAX, Black Book Kindle Edition,by Kogent Learning Solutions Inc 5. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed Kindle Edition,by DT Editorial Services 6. JSON at work ,Tom MArrs,O'REILLY,First edition,2017	
11	**Reference Books** 1. Learning Web Design A Beginner's Guide to Html, CSS, JavaScript, And Web Graphics, Jennifer Niederst Robbins, O'Reilly, 5th Edition,2018. 2. Ivan Bayross, "Web Enabled Commercial Applications Development using HTML, DHTML, Javascript, Perl CGI", BPB, 2004 3. HTML 5 for Web Designers (By: Jeremy Keith) – http:// freepdf-books.com 4. Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON kindle Edition by Lindsay Bassett, O'REILLY	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Class test of 1 of 15 marks Class test of 2 of 15 marks Average of the two: 15 marks Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks	**Format of Question Paper: External Examination (30 Marks)– 1 hr duration**
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any two (out of four) from Module 1 (15 marks) Q2: Attempt any two (out of four) from Module 2 (15 marks)	

Name of the Course: Major Practical II

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	**Object Oriented Programming usng C++ Practical** OOP encourages modular objects for reusable code, ensures well-organized and maintainable code via encapsulation, inheritance, and polymorphism, allowing flexibility and easy updates. Additionally, OOP models real-world scenarios, enhancing system understanding. **Web Designing Practical** Applying basic programming principles to the construction of websites

2	**Vertical :**	Major Practical
3	**Type :**	Practical
4	**Credits :**	2 credits (Total 60 hrs; 1 credit = 15 Hours for Theory or 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To explain the important characteristics of the C++ programming language. CO 2. To combine components of the C++ programming language to develop structured program. CO 3. To demonstrate the skills essential to compile, debug, and test C++ programs correctly. CO 4. To understand how to effectively implement HTML. CO 5. To develop the concept of basic and advanced text formatting. CO 6. To understand Hyper linking, Designing of webpage.	
8	**Course Outcomes (OC):** OC 1. Utilize C++ characteristics in software design and development. OC 2. Explain object-oriented techniques and explain how C++ supports them. OC 3. Employ C++ to demonstrate practical skill developing object-oriented solutions. OC 4. Examine a problem statements and design and develop object-oriented software using good coding practices and procedures. OC 5. Design static web pages using Hyper Text Markup Language (HTML). OC 6. Use their learned skills, knowledge and abilities to develop web sites OC 7. Collect information from the user with HTML Forms. OC 8. Enhance the look of web pages by implementing audio and video	
9	**Module I**	
	1. a. Write a C++ program to create a simple calculator. b. Write a C++ program to convert seconds into hours, minutes and seconds. c. Write a C++ program to find the volume of a square, cone, and rectangle. 2. a. Write a C++ program to find the greatest of three numbers. b. Write a C++ program to find the sum of even and odd n natural numbers c. Write a C++ program to generate all the prime numbers between 1 and n, where n is a value supplied by the user 3. a. Write a C++ program using classes and object Student to print name of the student, roll_no. Display the same. b. Write a C++ program for Structure bank employee to print name of the employee, account_no. & balance. Display the same also display the balance after withdraw and deposit	**30 Hrs**

c. Design the class Demo which will contain the following methods: readNo(), factorial() for calculating the factorial of a number, reverseNo() will reverse the given number, isPalindrome() will check the given number is palindrome, isArmstrong() which will calculate the given number is armStrong or not. WherereadNo() will be private method.

d. Write a program to demonstrate function definition outside class and accessing class members in function definition.

4.

a. Write a friend function for adding the two complex numbers, using a single class

b. Write a friend function for adding the two different distances and display its sum, using two classes.

c. Write a friend function for adding the two matrix from two different classes and display itssum

d. Write a Program to find Maximum out of Two Numbers using friend function.

Note: Here one number is a member of one class and the other number is member of some other class.

5.

a. Design a class Complex for adding the two complex numbers and also show the use of constructor.

b. Design a class Geometry containing the methods area() and volume() and also overload the area()function

c. Design a class StaticDemo to show the implementation of static variable and staticfunction

d. Write a C++ program to overload new/delete operators in a class.

e. Write a C++ Program to generate Fibonacci Series by using Constructor to initialize the Data Members.

6.

a. Overload the operator unary(-) for demonstrating operator overloading

b. Overload the operator + for adding the timings of two clocks, And also pass objects as an argument.

c. Overload the + for concatenating the two strings. For e.g “Py” +“thon” =Python

7.

a. Implement the concept of method overriding.

b. Show the use of virtual function

c. Show the implementation of abstract class.

8.

a. Write a C++ Program that illustrate single inheritance.

b. Write a C++ Program that illustrate multiple inheritance.

c. Write a C++ Program that illustrate multi-level inheritance.

d. Write a C++ Program that illustrate Hierarchical inheritance.

9.

a. Show the implementation of exception handling

b. Show the implementation for exception handling for strings
c. Show the implementation of exception handling for using the pointers.

10.

a. Design a class FileDemo open a file in read mode and display the total number of words and lines in the file.
b. Design a class to handle multiple files and file operations
c. Design a editor for appending and editing the files

Module II

1 Use of Basic and Advanced Tags, Lists and Backgrounds

a. Understanding elements, Tags and basic structure of HTML files
b. Design a web page using basic and advanced text formatting tags.
c. Design a web page using ordered, unordered list and description list.
d. Design a web page by choosing Background and Foreground Colors
e. Design a web page using Nested list and special characters.
f. Write an HTML code to display your CV on a web page.

2 Creating Hyperlinks, Anchors and style sheets

a. Design a web page with links to different pages and allow navigation between web pages.
b. Design a web page that automatically redirects the user to Other Content
c. Creating Hyperlinking to an E-Mail Address
d. Design a web page for creating Styles for Nested Tags
e. Design a web page by applying Styles to Hyperlinks
f. Design a web page by Creating and Linking to External Style Sheets.

3 Formatting Text and Paragraph by Using Style Sheets and displaying graphics

a. Design a web page by using text formatting tags
b. Design a web page using Indenting Paragraphs, Applying Border to a Paragraph and Specifying Horizontal Alignment of a Paragraph
c Implement a web page by creating inline spans and adjusting space between lines
d. Implement a web page by inserting a image and controlling the image size and padding
e. Design a web page by making image as a hyperlink
f. Develop a web page by using thumbnail graphics and also implement text for graphics

4 Tables , Page Layout and Navigation

a. Display a time table and display it in tabular format,
b. Write an html program to get the following output

NAME	SUBJECT	MARKS

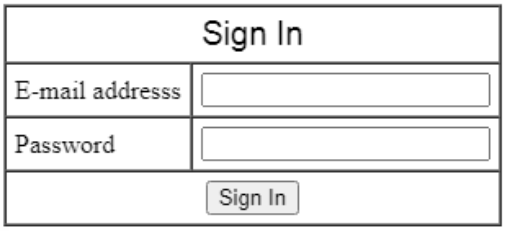
30 Hrs

Hillary	Advanced Web	75
	Operating System	60
Lary	Advanced Web	80
	Operating System	75
	Total Average: 72.5	

c. Design a table by merging the table cells.
d. Design a web page by Creating a Text-Based Navigation Bar
e . Design a web page by Creating a Graphical Navigation Bar
f. Design a web page with Image Map

5. Forms and Introducing video and audio tags

a. Design a web page with a form that uses all types of controls.
b. Design an admission form for any course in your college with text, pass word fields, check boxes, radio button and reset button.
c. Write a program to get the following output



d. Design a web page by placing a Video Clip on a Web Page
e. Design a web page by placing an Audio Clip on a Web Page
f. Design a web page embedding image, audio and video.

6 .Basics of java script
a. Using JavaScript, design a web page to accept a number from the user and print its Factorial.
b. Using JavaScript, a web page that prints Fibonacci series/any given series.
c. Write a JavaScript program to display all the prime numbers between 1 and 100.
d. Write a JavaScript program to accept a number from the user and display the sum of its digits.

7. Java Script: Validating User fields
a. Demonstrate the use of Document object methods.
b. Using java script, demonstrate validating Text Input Fields, Drop-down Lists and Checkboxes
c. Using java script, demonstrate validating Radio buttons and Validating Multi-Select Boxes
d. Write a Java script to prompt for users name and display it on the screen.

	8. Java Script : Handling the events a.Using java script, demonstrate the use of onAbort, onBlur, onChange, onClick, onDblClick events b. Using java script, demonstrate the use of onDragDrop, onError, onFocus events c. Using java script, demonstrate the use of onKeyDown, onKeyPress, onKeyUp, onLoad, onReset, onResize, onSelect, onSubmit, onUnload events d. Using java script, demonstrate the use of onMouseDown, onMouseMove, onMouseOut, onMouseOver, onMouseUp, onMove events. e. Using java script, demonstrate the use of onKeyDown, onKeyPress, onKeyUp, onLoad, onReset, onResize, onSelect, onSubmit, onUnload events 9. JQuery a. use JQuery effect in page b. Write a jQuery Code to find the data passed with the on() method for each element. c. Use JQuery Events d. JQuery traversal and manipulation 10. JSON Basics and Working with JSON a. Creating JSON b. Parsing JSON c. Persisting JSON d. Demonstrate use of JSON objects in array, print array on web page using document object e. Read data from json file and convert it into a JavaScript object and display the data in web page using document object	
10	Text Books 1. Object-oriented Programming C++, Hari Mohan Pandey 2. C++ Programming: An Object-Oriented Approach, Behrouz A. Forouzan, Richard F. Gilberg 3. C++ How to Program, Paul Deitel, Harvey Deitel 4. Step by Step HTML5, Faithe Wempen, Microsoft Press,2011 5. The Complete Reference HTML & CSS, Thomas A. Powell. McGraw Hill, 5th Edition,2010	
11	**Reference Books** 1. Object Oriented Programming in C++, E Balagurusamy 2. Object-Oriented Programming in C++ by Robert Lafore 3. Programming with ANSI C++, Bhushan Trivedi 4. Demystified Object- Oriented Programming with C++, Dorothy R. Kirk 5. Learning Web Design A Beginner's Guide to Html, CSS, JavaScript, And Web Graphics, Jennifer Niederst Robbins, O'Reilly, 5th Edition,2018. 6. "Web Enabled Commercial Applications Development using HTML, DHTML, Javascript, Perl CGI", Ivan Bayross, BPB, 2004 7. HTML 5 for Web Designers (By: Jeremy Keith) – http:// freepdf-books.com	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**

13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Vocational Skill Courses (VSC)

Name of the Course: Assembly Language Programming

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Introduction: The 8085 Assembly Language Programming course covers the principles and practices of writing low-level software that controls the 8085 microprocessor. This course provides an in-depth understanding of the 8085 microprocessor architecture and its instruction set, as well as how to write, debug, and optimize assembly language programs for this microprocessor. Relevance and Usefulness: The course is relevant to computer science/engineering students interested in learning about microprocessors and embedded systems programming. The course provides the fundamental knowledge and skills required to design and implement computer systems with low-level software control. Assembly language programming is the foundation of modern computer technology, which makes the course relevant to anyone interested in computer systems and programming. Application and Interest: The course is essential for students aspiring to work in the field of embedded systems, microcontroller/microprocessor programming, or any programming role that involves low-level software development. By the end of the course, students will be able to write efficient and optimized assembly language programs that control the functionality of a microprocessor. Connection with Other Courses: 8085 Assembly Language Programming is a fundamental course that provides an understanding of how computer systems work at the lowest level. It connects with several other computer science courses, such as Computer Organization and Architecture, Operating Systems, Compiler Design, and Embedded Systems Design. Demand in the Industry and Job Prospects: There is a high demand in the industry for programmers who possess knowledge of low-level software development, such as programming microprocessors with assembly language. Many industries, including aerospace, automotive, healthcare, and consumer electronics, require low-level software development skills in their employees. Job prospects for graduates with expertise in 8085 Assembly language

		programming are abundant in these sectors. Job roles may include embedded software engineer, hardware engineer, firmware developer, software developer, and testing/validation engineer.
2	**Vertical :**	Vocational Skill Course(VSC)
3	**Type :**	Practical
4	**Credits :**	2 credits (60 hours in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To gain a thorough understanding of the 8085 microprocessor architecture and its associated instruction set. CO 2. To develop the ability to write and debug assembly language programs for the 8085 microprocessor. CO 3. To learn the principles of computer organization and how they relate to the 8085 microprocessor. CO 4. To become proficient in the use of 8085 assembly language programming tools, simulators, and debuggers. CO 5. To learn how to interface different input/output devices with the 8085 microprocessor. CO 6. To understand the concept of interrupts and how they are used in 8085 assembly language programming.	
8	**Course Outcomes(CO):** OC 1. Explain the architecture of the 8085 microprocessor and its associated instruction set. OC 2. Identify the different types of registers and their functions in the microprocessor. OC 3. Describe the memory organization and addressing modes of the 8085 microprocessor. OC 4. Write assembly language programs for the 8085 microprocessor using various instructions and addressing modes. OC 5. Debug and troubleshoot assembly language programs for the 8085 microprocessor using simulators and debuggers. OC 6. Implement conditional branching and looping constructs in assembly language programs. OC 7. Use 8085 assembly language programming tools, such as editors, assemblers, and emulators for developing and testing programs. OC 8. Simulate microprocessor operations using emulators and debuggers. OC 9. Connect input/output devices, such as LEDs, switches, and displays, to the 8085 microprocessor. OC 10.	
9	**Modules:-** **Module 1:**	
	1. Perform the following Operations related to memory locations. a. Store the data byte 32H into memory location 4000H. b. Exchange the contents of memory locations 2000H and 4000H **2. Simple assembly language programs.**	**30 Hrs**

a. Subtract the contents of memory location 4001H from the memory location 2000H and place the result in memory location 4002H.
b. Subtract two 8-bit numbers.
c. Add the 16-bit number in memory locations 4000H and 4001H to the 16-bit number in memory locations 4002H and 4003H. The most significant eight bits of the two numbers to be added are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.
d. Add the contents of memory locations 40001H and 4001H and place the result in the memory locations 4002Hand 4003H.
e. Subtract the 16-bit number in memory locations 4002H and 4003H from the 16-bit number in memory locations 4000H and 4001H. The most significant eight bits of the two numbers are in memory locations 4001H and 4003H. Store the result in memory locations 4004H and 4005H with the most significant byte in memory location 4005H.
f. Find the l's complement of the number stored at memory location 4400H and store the complemented number at memory location 4300H.
g. Find the 2's complement of the number stored at memory location 4200H and store the complemented number at memory location 4300H.

3. Packing and unpacking operations.

a. Pack the two unpacked BCD numbers stored in memory locations 4200H and 4201H and store result in memory location 4300H. Assume the least significant digit is stored at 4200H.
b. Two digit BCD number is stored in memory location 4200H. Unpack the BCD number and store the two digits in memory locations 4300H and 4301H such that memory location 4300H will have lower BCD digit.

4. Register Operations

a. Write a program to shift an eight bit data four bits right. Assume that data is in register C.
b. Program to shift a 16-bit data 1 bit left. Assume data is in the HL register pair
c. Write a set of instructions to alter the contents of flag register in 8085.
d. Write a program to count number of I's in the contents of D register and store the count in the B register.

5. Multiple memory locations.

a. Calculate the sum of series of numbers. The length of the series is in memory location 4200H and the series begins from memory location 4201H. a. Consider the sum to be 8 bit number. So, ignore carries. Store the sum at memory location 4300H. b. Consider the sum to be 16 bit number. Store the sum at memory locations 4300H and 4301H
b. Multiply two 8-bit numbers stored in memory locations 2200H and 2201H by repetitive addition and store the result in memory locations 2300H and 2301H.
c. Divide 16 bit number stored in memory locations 2200H and 2201H by the 8 bit number stored at memory location 2202H. Store the quotient in memory locations 2300H and 2301H and remainder in memory locations 2302H and 2303H.

	d. Find the number of negative elements (most significant bit 1) in a block of data. The length of the block is in memory location 2200H and the block itself begins in memory location 2201H. Store the number of negative elements in memory location 2300H e. Find the largest number in a block of data. The length of the block is in memory location 2200H and the block itself starts from memory location 2201H. Store the maximum number in memory location 2300H. Assume that the numbers in the block are all 8 bit unsigned binary numbers.	
	Module 2:	
	1. Calculations with respect to memory locations. a. Write a program to sort given 10 numbers from memory location 2200H in the ascending order. b. Calculate the sum of series of even numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 8 bit number so you can ignore carries and store the sum at memory location 2Sample problem: c. Calculate the sum of series of odd numbers from the list of numbers. The length of the list is in memory location 2200H and the series itself begins from memory location 2201H. Assume the sum to be 16-bit. Store the sum at memory locations 2300H and 2301H. d. Find the square of the given numbers from memory location 6100H and store the result from memory location 7000H e. Search the given byte in the list of 50 numbers stored in the consecutive memory locations and store the address of memory location in the memory locations 2200H and 2201H. Assume byte is in the C register and starting address of the list is 2000H. If byte is not found store 00 at 2200H and 2201H f. Two decimal numbers six digits each, are stored in BCD package form. Each number occupies a sequence of byte in the memory. The starting address of first number is 6000H Write an assembly language program that adds these two numbers and stores the sum in the same format starting from memory location 6200H g. Add 2 arrays having ten 8-bit numbers each and generate a third array of result. It is necessary to add the first element of array 1 with the first element of array-2 and so on. The starting addresses of array I, array2 and array3 are 2200H, 2300H and 2400H, respectively **2. Assembly programs on memory locations.** a. Write an assembly language program to separate even numbers from the given list of 50 numbers and store them in the another list starting from 2300H. Assume starting address of 50 number list is 2200H b. Write assembly language program with proper comments for the following: c. A block of data consisting of 256 bytes is stored in memory starting at 3000H. This block is to be shifted (relocated) in memory from 3050H onwards. Do not shift the block or part of the block anywhere else in the memory.	**30 Hrs**

	d. Add even parity to a string of 7-bit ASCII characters. The length of the string is in memory location 2040H and the string itself begins in memory location 2041H. Place even parity in the most significant bit of each character. e. A list of 50 numbers is stored in memory, starting at 6000H. Find number of negative, zero and positive numbers from this list and store these results in memory locations 7000H, 7001H, and 7002H respectively f. Write an assembly language program to generate Fibonacci number. g. Program to calculate the factorial of a number between 0 to 8. **3. String operations in assembly programs.** a. Write an 8085 assembly language program to insert a string of four characters from the tenth location in the given array of 50 characters b. Write an 8085 assembly language program to delete a string of 4 characters from the tenth location in the given array of 50 characters. c. Multiply the 8-bit unsigned number in memory location 2200H by the 8-bit unsigned number in memory location 2201H. Store the 8 least significant bits of the result in memory location 2300H and the 8 most significant bits in memory location 2301H. d. Divide the 16-bit unsigned number in memory locations 2200H and 2201H (most significant bits in 2201H) by the B-bit unsigned number in memory location 2300H store the quotient in memory location 2400H and remainder in 2401H e. DAA instruction is not present. Write a sub routine which will perform the same task as DAA. **4. Calculations on memory locations.** a. To test RAM by writing '1' and reading it back and later writing '0' (zero) and reading it back. RAM addresses to be checked are 40FFH to 40FFH. In case of any error, it is indicated by writing 01H at port 10 b. Arrange an array of 8 bit unsigned no in descending order c. Transfer ten bytes of data from one memory to another memory block. Source memory block starts from memory location 2200H where as destination memory block starts from memory location 2300H d. Write a program to find the Square Root of an 8 bit binary number. The binary number is stored in memory location 4200H and store the square root in 4201H. e. Write a simple program to Split a HEX data into two nibbles and store it in memory **5. Operations on BCD numbers.** a. Add two 4 digit BCD numbers in HL and DE register pairs and store result in memory locations, 2300H and 2301H. Ignore carry after 16 bit. b. Subtract the BCD number stored in E register from the number stored in the D register c. Write an assembly language program to multiply 2 BCD numbers	
10	**Text Books** 1. 8080A/8085 Assembly Language Programming, Lance A. Leventhel, Osborne, 1978	

11	**Reference Books** 1. Microprocessors Architecture, Programming and Applications with the 8085, Fifth Edition, Penram Publications, 2012	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Skill Enhancement Courses (SEC)

Name of the course : Web Programming

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course covers a range of topics aimed at equipping students with the skills and knowledge needed to create visually appealing, functional, and user-friendly websites. The course provides an insight into emerging technologies to design and develop state of the art web applications using client-side scripting, server-side scripting, and database connectivity. website development includes all related development tasks, such as client-side scripting, server-side scripting, server and network security configuration, eCommerce development, and content management system (CMS) development. Website design is a combination of different elements that work together to create an effective and user-friendly experience. These include the use of typography, layout, color theory, grid systems, motion graphics, and responsive designs.
2	**Vertical :**	Skill Enhancement Course(SEC)
3	**Type :**	Practical
4	**Credits:**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO)** **CO1:** To understand how to use Java script objects and XML. **CO2:** To create well organized, styled web pages **CO3:** To add versatility to a web page using jQuery **CO4:** To deploy a local web server and run a simple web application. **CO5:** To read and process data in MySQL using PHP. **CO6:** To understand usage of Bootstrap	
8	**Course Outcomes (OC)** OC1: Knowledge in different java script objects. OC2: How to use XML with CSS and XSL OC3: validate a form using jQuery OC4: handle asynchronous requests OC4: Write and deploy PHP with database and to simplify web development. OC5: Create a responsive layout using the Bootstrap	

	9	**Modules:** **Module 1:** **1. Write JavaScript code for** a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document c. Storing and Retrieving Cookies **2. Create a XML file with Internal / External DTD and display it using** a. CSS b. XSL **3. Write PHP scripts for- Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions** **4. Write PHP scripts for** a. Retrieving data from HTML forms b. Working with Arrays c. Working with Files (Reading / Writing) **5. Advanced PHP** a. Write a PHP program to demonstrate use of sessions and cookies. b. Write a PHP program to demonstrate use of filters.	**30 Hrs**
		Module 2 **6. PHP and MySQL** a. Write a PHP program to create: Create a database College b. Create a table Department (Dname, Dno, Number_of_faculty) c. Write a PHP program to create a database named “College”. Create a table named “Student” with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 to 75 in a tabular format. **7. Write a PHP program** a. Update rows in a table b. Delete rows from a table **8. Design a PHP page for authenticating a user** **9. Write PHP scripts for** a. Storing and Retrieving Cookies b. Storing and Retrieving Sessions **10. Perform the following using Bootstrap:** a. Create a responsive layout using the Bootstrap grid system b. Create a simple Bootstrap navbar with dropdown menus c. Create a basic Bootstrap form with validation	**30 Hrs**
	10	**Text Books** • HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press, 2016 • Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India, 2018	

	• PHP: A Beginners Guide, Vikram Vaswani, TMH	
11	**Reference Books** • HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY, 2011 • Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O'Reilly, 2018 • PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley, 2018	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

Name of the Course: PLSQL Practical

Sr.No.	Heading	Particulars	
1	**Description the course : Including but Not limited to:**	PL/SQL ,Oracle's procedural extension language for SQL, allows developers to include procedural language components such as loops, conditional statements and functions. The course enables students with practical experience in using PL/SQL for effective database programming and development.	
2	**Vertical :**	Skill Enhancement Course(SEC)	
3	**Type :**	Practical	
4	**Credits :**	2 credits	
5	**Hours Allotted :**	60 Hours	
6	**Marks Allotted:**	50 Marks	
7	**Course Objectives(CO):** CO 1. Comprehend the basics of PL/SQL and gain knowledge about control and conditional statement in PL/SQL. CO 2. Understand working with cursors,collections and composite data types in PL/SQL. CO 3. Develop expertise in creating stored procedures and functions. CO 4. Explore the use of triggers to automate responses to events within the database. CO 5. Understand the concept of Exception handling. CO 6. Design modular applications using packages.		
8	**Course Outcomes (OC):** OC 1. Use PL/SQL variables ,data types, control and conditional statement. OC 2. Apply sequences and cursor in PL/SQL. OC 3. Work with Collection and Composite Data Types. OC 4. Develop PL/SQL structures like functions, procedures and triggers for database applications. OC 5. Handle errors and exceptions in PL/SQL programs. OC 6. Develop PL/SQL packages.		
9	**Modules:-** **Module 1:**		
	1. **PL/SQL Basics**- Use of variables, Write executable statement, Interacting with Oracle Server, Create anonymous PL/SQL block,Sequences 2. **Control Structure in PL/SQL**- Using while loop, Do loop, For loop, Use of GOTO statement 3. **Create conditional statement using PL/SQL**- Using if statement, Using if else statement, Using elsif ladder, Using case expression. 4. **Create cursor in PL/SQL**- Implicit cursor, Explicit cursor, Parameterized cursor **5. Collection and Composite Data Types** - Working with Collections,Working with Composite Data Types		**30 Hrs**

	Module 2:	
	1. Creation of Procedures in PL/SQL **2. Functions in PL/SQL** 3. **Creation of Trigger –** Create Row level trigger, Create Statement level trigger, Create instead of trigger 4. **Handling exceptions**- Creation of user defined exception, Creation of system defined exception 5. **Creation of Package in PL/SQL**	**30 Hrs**
10	**Text Books** 1. Programming with PL/SQL for Beginners , H. Dand, R. Patil and T. Sambare, X –Team 2. Oracle pl/sql Programming ,Feuerstein, S., & Pribyl, B. ," O'Reilly Media, Inc.".	
11	**Reference Books** 1. Oracle Database PL/SQL Language Reference, 12c Release 1 (12.1) E50727-04 , Alpern, D., Belden, E., Agrawal, S., Baer, H., Castledine, S., Chang, T., & Yang, M. 2. Oracle PL/SQL for dummies , Rosenblum, M., & Dorsey, P. (2006), John Wiley & Sons. 3. PL/SQL Programming ,Ivan Bayross, BPB	
12	**Internal Continuous Assessment: 40%**	**Semester End Examination: 60%**
13	**Continuous Evaluation through:** Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks	

QUESTION PAPER PATTERN

(External and Internal)

<table>
<tr><td rowspan="4">I</td><td colspan="2">A Theory of 2 credits is evaluated for a total of 50 Marks</td></tr>
<tr><td>Internal Continuous Assessment:</td><td>40%[20 Marks]</td></tr>
<tr><td colspan="2">Continuous Evaluation through:
Class test of 1 of 15 marks
Class test of 2 of 15 marks
Average of the two: 15 marks

Quizzes/ Presentations/ Assignments: 5 marks
Total: 20 marks</td></tr>
<tr><td colspan="2">External Semester End Examination: 60%[30 Marks]</td></tr>
<tr><td colspan="3">Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)
Q1: Attempt any two (out of four) from Module 1 (15 marks)
Q2: Attempt any two (out of four) from Module 2 (15 marks)</td></tr>
<tr><td rowspan="4">II</td><td colspan="2">A Practical of 2 credits is evaluated for a total of 50 Marks</td></tr>
<tr><td>Internal Continuous Assessment:</td><td>40%[20 Mrks]</td></tr>
<tr><td colspan="2">Continuous Evaluation through:
Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.</td></tr>
<tr><td colspan="2">Semester End Examination: 60%[30 Marks]</td></tr>
<tr><td colspan="3">Format of Question Paper: Duration 2 hours. Certified copy of Journal is compulsory to appear for the practical examination(30 Marks)
Practical Slip:
Q1. From Module 1 13 marks
Q2. From Module 2 12marks
Q3. Journal and Viva 05 marks</td></tr>
</table>

Examination and Standard of Passing:

Regulations regarding the scheme of exams, number of credits and standard of passing will be as prescribed by the University of Mumbai.

A student is said to have passed if he/she secures 40% of marks allotted in each head of passing. External evaluation of 30 marks and Internal evaluation of 20 marks are treated as separate heads of passing.

The Ten Point Grading System prescribed by the University of Mumbai will be as follows:

Letter Grades and Grade Points

Semester GPA/ Program CGPA Semester/ Program	**% of Marks**	**Alpha-Sign / Letter GradeResult**	**Grade Points**
9.00-10.00	90.0-100	O (Outstanding)	10
8.00-<9.00	80.0-<90.0	A+ (Excellent)	9
7.00-<8.00	70.0-<80.0	A (Very Good)	8
6.00-<7.00	60.0-<70.0	B+ (Good)	7
5.50-<6.00	55.0-<60.0	B (Above Average)	6
5.00-<5.50	50.0-<55.0	C (Average)	5
4.00-<5.00	40.0-<50.0	P (Pass)	4
Below 4.00	Below 40	F (Fail)	0
Ab (Absent)	-	Absent	0

This syllabus is applicable to IDOL students as well, w.e.f. 2025-26

Appendix B

Justification for B.Sc. (Information Technology)

1.	Necessity for starting the course:	**A large amount of The demand for IT professionals is consistently high, and individuals with a B.Sc in IT can find opportunities in various sectors, including technology companies, healthcare, finance, government, and more.**
2.	Whether the UGC has recommended the course:	**Yes**
3.	Whether all the courses have commenced from the academic year 2024-2025	**To be implemented from 2024-2025 onwards**
4.	The courses started by the University are self-financed, whether adequate number of eligible permanent faculties are available?:	**Self-financed** **Yes. Some experts are called as visiting faculties**
5.	To give details regarding the duration of the Course and is it possible to compress the course?:	**4 years. Not possible to compress the program**
6.	The intake capacity of each course and no. of admissions given in the current academic year:	**60 seats for one division. Admissions will be held from 2024-2025 onwards**
7.	Opportunities of Employability / Employment available after undertaking these courses:	**B.Sc in Information Technology can open up a wide range of opportunities and employment prospects across various industries. Additionally, as technology continues to advance, new roles and specialties within the IT field are continually emerging, providing diverse career paths for IT graduates.**

Sign of Chairperson
Dr. Mrs. R. Srivaramangai
Ad-hoc BoS (IT)

Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sign of Offg. Dean,
Prof. Shivram S. Garje
Faculty of Science & Technology

University of Mumbai

Website – mu.ac.in
Email id - dr.aams@fort.mu.ac.in
aams3@mu.ac.in

Academic Authorities,
Meetings &Services (AAMS)
Room No. 128, M. G. Road, Fort,
Mumbai – 400 032.
Tel.022-68320033

Re- accredited with A ++ Grade (CGPA 3.65) by NAAC
Category- I University Status awarded by UGC

No.2026 May AAMS C-106/44024 Date: 14th May, 2026

CIRCULAR:-

Attention of all the Principals of the Affiliated Colleges & Autonomous Colleges, all Academic Heads of University Departments, the Directors of the Recognized Institutions/Centers, Director, Centre for Distance & Online Education, Director, Dharmaveer Anand Dighe Thane Sub-Campus, Co-ordinator, School of Engineering and Applied Sciences, Kalyan Sub-Campus, Director, Chitrakar Padmabhushan Dr.Dhananjay Keer Ratnagiri Sub-Campus, Director, Sindhudurg Sub-Campus, Principal, Vishwabhushan Bharatratna Dr. B. A. Ambedkar College, Ambadave, Ratnagiri and Principal, V.V. Dalvie College, Talere, Sindhudurg is invited to this office Circular No. AAMS/ICD/2025-26/ 37 of dated 27 May, 2025 relating to the NEP UG & PG Syllabus.

They are hereby informed that the recommendations made by the **Board of Studies in Information Technology** at its meeting held on 17th February, 2026 vide item No.1 and subsequently passed by the Board of Deans at its meeting held on 17th March, 2026 vide item No. 6.16 (N) have been accepted by the Academic Council at its meeting held on 25th March, 2026 **vide** item No. 6.34 (N). In accordance therewith syllabus of **B.Sc. (Information Technology) (Scheme - I) (Sem V & VI)** (NEP 2020) is introduced as per appendix with effect from the academic year 2026-27.

(The said circular is available on the University's website www.mu.ac.in).

(Dr. Prasad Karande)
REGISTRAR

MUMBAI – 400 032
14th May, 2026

To

All the Principals of the Affiliated Colleges & Autonomous Colleges, all Academic Heads of Departments, the Directors of the Recognized Institutions/Centers, Director, Centre for Distance & Online Education, Director, Dharmaveer Anand Dighe Thane Sub-Campus, Co-ordinator, School of Engineering and Applied Sciences, Kalyan Sub-Campus, Director, Chitrakar Padmabhushan Dr.Dhananjay Keer Ratnagiri Sub-Campus, Director, Sindhudurg Sub-Campus, Principal, Vishwabhushan Bharatratna Dr. B. A. Ambedkar College, Ambadave, Ratnagiri and Principal, V.V. Dalvie College, Talere, Sindhudurg

AC./6.34 (N)/25/3/2026

Copy forwarded with Compliments for information to:-

1) The Chairman, Board of Deans,
2) The Dean, Faculty of Science,
3) The Chairman, **Board of Studies in Information Technology**
4) The Director, Board of Examinations and Evaluation,
5) The Director, Department of Students Development,
6) The Director, Department of Information & Communication Technology,
7) The Director, Centre for Distance and Online Education (CDOE)Vidyanagari,
8) The Deputy Registrar, Admission, Enrolment, Eligibility & Migration Department (AEM),

Copy forwarded for information and necessary action to :-	
1	The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Dept)(AEM), dr@eligi.mu.ac.in
2	The Deputy Registrar, Result unit, Vidyanagari drresults@exam.mu.ac.in
3	The Deputy Registrar, Marks and Certificate Unit,. Vidyanagari dr.verification@mu.ac.in
4	The Deputy Registrar, Appointment Unit, Vidyanagari dr.appointment@exam.mu.ac.in
5	The Deputy Registrar, CAP Unit, Vidyanagari cap.exam@mu.ac.in
6	The Deputy Registrar, College Affiliations & Development Department (CAD), deputyregistrar.uni@gmail.com
7	The Deputy Registrar, PRO, Fort, (Publication Section), Pro@mu.ac.in
8	The Deputy Registrar, Executive Authorities Section (EA) eau120@fort.mu.ac.in He is requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to the above circular.
9	The Deputy Registrar, Research Administration & Promotion Cell (RAPC), rapc@mu.ac.in
10	The Deputy Registrar, Academic Appointments & Quality Assurance (AAQA) dy.registrar.tau.fort.mu.ac.in ar.tau@fort.mu.ac.in
11	The Deputy Registrar, College Teachers Approval Unit (CTA), concolsection@gmail.com
12	The Deputy Registrars, Finance & Accounts Section, fort draccounts@fort.mu.ac.in
13	The Deputy Registrar, Election Section, Fort drelection@election.mu.ac.in
14	The Assistant Registrar, Administrative Sub-Campus Thane, thanesubcampus@mu.ac.in
15	The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan, ar.seask@mu.ac.in
16	The Assistant Registrar, Ratnagiri Sub-centre, Ratnagiri, ratnagirisubcentar@gmail.com
17	The Director, Centre for Distance and Online Education (CDOE), Vidyanagari, director@idol.mu.ac.in
18	Director, Innovation, Incubation and Linkages, Dr. Sachin Laddha pinkumanno@gmail.com
19	Director, Department of Lifelong Learning and Extension (DLLE), dlleuniversityofmumbai@gmail.com

Copy for information :-	
1	P.A to Hon'ble Vice-Chancellor, vice-chancellor@mu.ac.in
2	P.A to Pro-Vice-Chancellor pvc@fort.mu.ac.in
3	P.A to Registrar, registrar@fort.mu.ac.in
4	P.A to all Deans of all Faculties
5	P.A to Finance & Account Officers, (F & A.O), camu@accounts.mu.ac.in

To,

1	The Chairman, Board of Deans pvc@fort.mu.ac.in
2	**Faculty of Humanities,** **Offg. Dean** 1. Prof.Anil Singh Dranilsingh129@gmail.com **Offg. Associate Dean** 2. Prof.Manisha Karne mkarne@economics.mu.ac.in 3. Dr.Suchitra Naik Naiksuchitra27@gmail.com
	Faculty of Commerce & Management, **Offg. Dean,** 1 Prin.Ravindra Bambardekar principal@model-college.edu.in **Offg. Associate Dean** 2. Dr.Kavita Laghate kavitalaghate@jbims.mu.ac.in 3. Prin.Kishori Bhagat kishoribhagat@rediffmail.com

	Faculty of Science & Technology **Offg. Dean** 1. Prof. Shivram Garje ssgarje@chem.mu.ac.in **Offg. Associate Dean** 2. Dr. Madhav R. Rajwade Madhavr64@gmail.com 3. Prin. Deven Shah sir.deven@gmail.com
	Faculty of Inter-Disciplinary Studies, **Offg. Dean** 1.Dr. Anil K. Singh aksingh@trcl.org.in **Offg. Associate Dean** 2.Prin.Chadrashekhar Ashok Chakradeo cachakradeo@gmail.com 3. Dr. Kunal Ingle drkunalingle@gmail.com
3	Chairman, Board of Studies,
4	The Director, Board of Examinations and Evaluation, dboee@exam.mu.ac.in
5	The Director, Board of Students Development, dsd@mu.ac.in DSW direcotr@dsw.mu.ac.in
6	The Director, Department of Information & Communication Technology, director.dict@mu.ac.in

AC – 25/03/2026
Item No. 6.34 (N)

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4 & 6 (Scheme – I)		
Faulty of Science & Technology		
Board of Studies in Information Technology		
Name of the Programme – B.Sc. (Information Technology)		
U.G. Third Year Programme	**Exit Degree**	**B.Sc. (Information Technology)**
Semester		**V & VI**
From the Academic Year		**2026-27**

University of Mumbai

(As per NEP 2020)

Sr.No.	Heading	Particulars
1	**Title of program** **O:_____**	**B.Sc. (Information Technology)**
2	**Exit Degree**	**B.Sc. (Information Technology)**
3	**Scheme of Examination** **R: _____**	**NEP** **40% Internal** **60% External, Semester End Examination** **Individual Passing in Internal and External Examination**
4	**Standards of Passing** **R: _____**	**40%**
5	**Credit Structure** **R: SU-510 E(I) R: SU-510 F(I)**	**Attached herewith**
6	**Semesters**	**Sem. V & VI**
7	**Program Academic Level**	**5.5**
8	**Pattern**	**Semester**
9	**Status**	**New**
10	**To be implemented from Academic Year**	**2026-27**

Sd/-
Sign of the BOS
Dr. R. Srivaramangai
Chairman/Co-ordinator
BOS/Ad-hoc BOS in Information Technology

Sd/-
Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sd/-
Sign of the Offg. Dean
Prof. Shivram S. Garje
Faculty of Science & Technology

B.Sc. (Information Technology)

Credit Structure (Sem. V & VI)

(B. Sc.)- Major & Minor

	R: SU-510 E (I)								
Vertical		**1**		**2**	**3**	**4**	**5**	**6**		
Level	**Semester**	**Major (M1)**		**Minor (M2)**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC,RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		Mandatory	**Electives**							
5.5	V	10 Dot Net Core and Progressive Web Application (2) Dot Net Core and Progressive Web Application(Practical) (2) Artificial Intelligence and Application Development (2) Artificial Intelligence Application Development and Jira (Practical) (2) Indian Knowledge Systems in Information Technology (2)	4 Enterprise Java Practical (2) **OR** Android Practical (2) Full Stack development MERN Practical (2) **OR** Linux Administration Practical (2)	4 (2+2)	----	VSC: 2 Internet of Things Practical (2)	----	CEP:2	22	**UG Degree 132**

	R: SU-510 F (I)								
	VI	10 Project Management (2) Computer Security (2) Data Analytics and Business Intelligence (2) Data Analytics and Business Intelligence and Computer Security Practical (2) Cloud Computing and Testing Tools Practical (2)	4 Geographical Information System Practical (2) **OR** Enterprise Networking Practical (2) Cyber law and Digital Policy (2) **OR** IT Service Management (2)	4 (2+2)	----	----	----	OJT :4	22	
	Cum Cr.	48	8	18	12	8+6	8+4+2	8+2+2 +2+4	132	
Exit option: Award of UG Degree in Major and Minor with 132 credits OR Continue with Major and Minor									

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Community Engagement Project, CC – Co-Curricular, RP – Research Project]

[* 2 Credit IKS Major paper (Core Subject Specific Theory may be included in Sem III or V)]

Program Structure for Sem. V & VI
(NEP 2020)

Vertical No		Paper Title		Credits
Sem. V				
1.	**Mandatory**	1. Dot Net Core and Progressive Web Application		2
		2. Dot Net Core and Progressive Web Application(Practical)		2
		3. Artificial Intelligence and Application Development		2
		4. Artificial Intelligence Application Development and Jira (Practical)		2
		5. Indian Knowledge Systems in Information Technology		2
	Electives	1	1. Enterprise Java Practical Or 2. Android Practical	2
		2	1. Full Stack development MERN Practical Or 2. Linux Administration Practical	2
2.	**Minor** **(To be selected from other programme)**			4 (2+2)
4.	**VSC (Practical)**	1. Internet of Things Practical		2
6.	CEP (To be selected from CEP topic list)			2
			Credits	**22**
Sem. VI				
1.	**Mandatory**	1. Project Management		2
		2. Computer Security		2
		3. Data Analytics and Business Intelligence		2
		4. Data Analytics and Business Intelligence and Computer Security Practical		2
		5. Cloud Computing and Testing Tools Practical		2
	Electives	1.	1. Geographical Information System Practical Or 2. Enterprise Networking Practical	2
		2	1. Cyber law and Digital Policy Or 2. IT Service Management.	2
2.	**Minor** (To be selected from other discipline)			4 (2+2)
6.	**VI**	**OJT**		4
			Credits	**22**
			Total Credits	**44**

Sem. - V

Sem. - V
Vertical – 1
Major
Mandatory
(2+2+2+2+2)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Dot .Net Core and Progressive Web Application

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	.NET Core and PWA is a major theory course It bridges two of the most important domains in contemporary full-stack and cross-platform web development: server-side application development on the modern Microsoft .NET platform using C# and ASP.NET Core, and client-side Progressive Web Application (PWA) engineering using open web standards. Together, they equip students with an end-to-end skill set for building enterprise-grade, cross-device, and offline-capable web applications — a profile that is in exceptionally high demand across Indian IT industry, product companies, and global MNCs. language fundamentals (data types, control flow, methods, error handling) through object-oriented programming (classes, inheritance, polymorphism, interfaces, exception handling) — before progressing to ASP.NET Core architecture: the request pipeline, middleware, routing, the Model-View-Controller (MVC) pattern, Razor Pages, and RESTful Web API development. The second half pivots to Progressive Web Applications — exploring the PWA philosophy, Service Workers and their lifecycle, the Web App Manifest, caching strategies, offline functionality, push notifications, responsive/mobile-first design, and the critical integration of PWA front-ends with ASP.NET Core back-ends, rounded off by security fundamentals (CORS, JWT overview) and performance optimisation using industry tools such as Lighthouse.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):**	

	CO 1: To provide students with fundamental and advanced knowledge of C# programming and the modern .NET platform. CO 2: To enable students to design and develop web applications using ASP.NET Core architecture and MVC concepts. CO 3: To familiarize students with restful API development and integration with front-end applications. CO 4: To introduce the concepts, architecture, and implementation of Progressive Web Applications (PWA). CO 5: To develop skills in building responsive, secure, and performance-optimized web applications using modern web standards.
8	**Course Outcomes (OC):** OC 1: Develop applications using C# and implement object-oriented programming concepts in the .NET environment. OC 2: Design and deploy dynamic web applications using ASP.NET Core, routing, middleware, and MVC architecture. OC 3: Build and consume restful APIs and integrate backend services with client-side applications. OC 4: Create Progressive Web Applications with features such as service workers, web app manifest, and offline caching. OC 5: Apply modern web development practices including responsive design, security mechanisms, and performance optimization in real-world applications.
9	**Modules:-**
	Module 1: Introduction to C#.NET
	1. Overview of .NET Ecosystem:.NET Framework vs. .NET Core / .NET 5+ / .NET 7+, Common Language Runtime (CLR), Assemblies, CLI tools 2. Basics of C# Language: Data types, Variables, Operators, Expressions, Control Structures: If, Switch, Loops, Methods, Parameters, Error Handling 3. Object Oriented Programming in C#: Classes & Objects, inheritance, Polymorphism, Interfaces, Namespaces and Exception Handling 4. ASP.NET Core Fundamentals: Web Server Basics, Middleware, Request Lifecycle, Routing, Controllers, Views (MVC),Razor pages and REST API endpoints, Web Forms
	Module 2: PWA 1. Introduction to PWA: Concept and evolution of Progressive Web Applications, key characteristics (reliable, fast, engaging), comparison with native and hybrid apps, requirement of HTTPS. 2. Core PWA Components: Web App Manifest (properties and configuration), Service Workers (registration and lifecycle), caching strategies, offline functionality, and basic push notifications. 3. Responsive Web Design: Mobile-first approach, media queries, flexible layouts, and UI/UX principles for cross-device compatibility.

	4. Integration with .NET Backend: Consuming ASP.NET Core Web APIs, deployment basics, performance testing using tools like Lighthouse. Security and Optimization: CORS basics, authentication concepts (JWT overview), performance and accessibility best practices.
10	**Text Books** 1. Beginning ASP.NET 4.5 in C#, Matthew MacDonald,Apress 2. C# 2015, Anne Bohem and Joel Murach,Murach,Third 3. Progressive Web App,Dean Alan Hume
11	**Reference Books** 1. C# 4.0: The Complete Reference Paperback – 27 April 2010 by Herbert Schildt (Author) 2. C#: A Beginner's Guide, McGraw-Hill Professional 3. Progressive Web Applications by Jason Grigsby
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks
13	**Semester End Examination: 60%**
	Format of Question Paper: External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any three (out of five/Six) from Module 1 (15 marks) Q2: Attempt any three (out of five/Six) from Module 2 (15 marks)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: (Dot) .Net Core and Progressive Web Application (Practical)

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	.Net Core and PWA practical course focuses on developing hands-on skills in C# and .NET technologies along with modern web development. Students work on basic programming tasks like arithmetic operations, sorting, and patterns, and gradually learn advanced concepts such as inheritance, interfaces, delegates, and exception handling. It also includes building web applications using ASP.NET, working with databases through Entity Framework Core, and performing CRUD operations. In addition, students are introduced to Progressive Web Applications (PWAs), where they learn to create web apps with offline support and real-world features.
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** **CO1.** To provide fundamental knowledge of C# programming and object-oriented concepts. **CO2.** To develop problem-solving skills using advanced C# features such as delegates, events, interfaces, and exception handling. **CO3.** To enable students to design and develop web applications using ASP.NET Web Forms and ASP.NET Core. **CO4.** To introduce database connectivity and ORM concepts using SQL Server and Entity Framework Core. **CO5.** To familiarize students with Web API development and modern web technologies such as Progressive Web Applications (PWA).	
8	**Course Outcomes (OC):** OC1. Develop C# programs using basic programming constructs, arrays, strings, and pattern generation. OC2. Apply Object-Oriented Programming concepts such as inheritance, polymorphism, interfaces, constructors, delegates, and exception handling.	

	OC3. Design and develop ASP.NET Web Forms and ASP.NET Core applications using server controls and validation techniques. OC4. Implement database connectivity using SQL Server and Entity Framework Core to perform CRUD operations. OC5. Develop Web APIs and Progressive Web Applications (PWA) with offline support, service workers, and manifest configuration.
9	**Modules:-**
	Module 1:
	1.Write the program for the following: a. Create an MVC application to perform basic **Arithmetic Operations**. b. Create an application to print Floyd's triangle till n rows in C#. c. Write C# code to arrange the name of cities in sorted order. Accept name of 10 cities from the user **2. Write the program with different features of C#** a. Function Overloading b. Inheritance (all types) c. Constructor overloading d. Interfaces e. Using Delegates and events f. Exception handling 3. **Write the program for the following**: a. Create methods add(). multiply(), substract() ,divide() with suitable parameters and call these methods using concept of C# delegate. b. Write a program using multicast delegate. c. Create a class **BankAccount** with AccountNumber and Balance. Implement property for Balance with validation (Balance cannot be negative). **4. Create an ASP.NET Web Forms application using simple server controls.** a. Design a Student Registration Form using the following ASP.NET controls: i. Label and TextBox to accept Student Name ii. RadioButton to select Gender (Male/Female) iii. DropDownList to select Course (BSc IT, BSc CS, BCA) iv. CheckBox for accepting Terms & Conditions v. Calendar control to select Date vi. Button control to submit the form vii. Label control to display the output b. Create a Registration form to demonstrate use of various Validation controls. c. Create Web Form to demonstrate use of Adrotator Control. **5. Create the following application:** a. Create a web application to demonstrate the use of different types of Cookies. b. Create a **Product Table** in SQL Server and perform **Insert Operation** using Entity Framework Core.

	c. Develop a web application to perform **CRUD Operations (Create, Read, Update, Delete)** on Employee table using EF Core. d. Authentication and Authorization in ASP.NET Core.
	Module 2:
	1. a. Create a simple website for a College Information Portal and convert it into a Progressive Web Application. b. Create a PWA that displays a custom offline page when the user is not connected to the internet. 2. a. Develop a PWA that works in offline mode using Service Worker. b. Create a Student Record Management PWA. 3. a. Create an HTML page that would be the starting point of the application. This HTML will contain a link to the file named manifest.json. This is an important file that would be created in the next step. b. Create a simple Web API that returns Student data in JSON format. 4. a. Develop CRUD operations using ASP.NET Core Web API and Entity Framework Core. b. Create an API endpoint to calculate Factorial of a Number and test it using Postman. 5. a. Create a PWA with a Web App Manifest file including: App name Icons Start URL Display mode Implement Push Notifications in a PWA. b. Develop an E-Commerce PWA with product listing and offline cart functionality.
10	**Text Books** 1. Programming in C# – E. Balagurusamy, McGraw Hill Education. 2. ASP.NET Core in Action – Andrew Lock, Manning Publications. 3. Pro ASP.NET Core MVC – Adam Freeman, Apress. 4. Entity Framework Core in Action – Jon P Smith, Manning Publication
11	**Reference Books** 1. Professional ASP.NET Core – Jon Galloway et al., Wrox Publication. 2. Learning Progressive Web Apps – John Wargo, Addison-Wesley.
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and write-up submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and write-up submission totalling to 50 marks and can be converted to 20 marks.

13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination. (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hours)** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Title of Paper <u>Artificial Intelligence and Application Development</u>

Sr. No.	Heading	Particulars
1	Description of the course : Including but Not limited to : 	This course introduces the fundamental concepts and practical methodologies of Artificial Intelligence with a strong emphasis on real-world application development. It covers intelligent agents, problem-solving through search, probabilistic reasoning, machine learning, and deep learning techniques using standard AI frameworks and development tools. Students learn to design, implement, evaluate, and deploy AI models as part of complete application systems. The course integrates theoretical foundations with hands-on development, enabling learners to build intelligent solutions such as predictive analytics systems, recommendation engines, chatbots, and computer vision applications.
2	Vertical :	Major
3	Type :	Theory
4	Credit:	2 credits (1 credit = 15 Hours for Theory)
5	Hours Allotted :	30 Hours
6	Marks Allotted:	50 Marks
7	Course Objectives(CO): CO 1. To understand and apply core AI concepts including intelligent agents, search techniques, probabilistic reasoning, and fundamental machine learning models. CO 2.To implement supervised and introductory deep learning algorithms with appropriate evaluation and performance analysis. CO3. To design end-to-end AI pipelines integrating data preprocessing, model development, validation, and inference within application architectures. CO4. To develop and deploy AI-based applications while ensuring scalability, robustness, and ethical considerations.	
8	**Course Outcomes (OC):** OC1. Apply intelligent agent models, search algorithms, probabilistic reasoning, and fundamental machine learning techniques to solve structured AI problems. OC2. Implement and evaluate supervised and basic deep learning models using appropriate performance metrics.	

	OC3. Design and construct end-to-end AI pipelines including data preprocessing, model training, validation, and inference integration within applications. OC4. Develop and deploy AI-based applications ensuring scalability, robustness, and adherence to ethical and responsible AI practices.
9	**Modules**:- Per credit One module can be created
	Module 1: Foundations and theoretical approach
	1. **Introduction**: What Is AI? The Foundations of Artificial Intelligence, The State the Art, Risks and Benefits of AI. 2. **Intelligent Agents**: Agents and Environments, Good Behaviour: The Concept of Rationality, The Nature of Environments, The Structure of Agents 3. **Solving Problems by Searching**: Problem-Solving Agents, Example Problems, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions 4. **Quantifying Uncertainty**: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use, Naive Bayes Models, The Wumpus World Revisited **5. Probabilistic Reasoning**: Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Exact Inference in Bayesian Networks, Approximate Inference for Bayesian Networks, Causal Networks
	Module 2: AI Application Development and Ethics
	1. Learning from Examples: Forms of Learning, Supervised Learning, Learning Decision Trees, Model Selection and Optimization, The Theory of Learning, Linear Regression and Classification, Nonparametric Models, Ensemble Learning, Developing Machine Learning Systems 2. **The Machine Learning Landscape** :What Is Machine Learning?, Why Use Machine Learning?, Types of Machine Learning Systems, Supervised and Unsupervised Learning, Batch and Online Learning, Instance-Based Versus Model-Based Learning, Main Challenges of Machine Learning, Insufficient Quantity of Training Data, Nonrepresentative Training Data, Poor-Quality Data, Irrelevant Features, Overfitting the Training Data, Underfitting the Training Data, Stepping Back, Testing and Validating, Exercises 3. **Classification**: MNIST, training a Binary Classifier, Performance Measures, Measuring Accuracy Using Cross-Validation, Confusion Matrix, Precision and Recall, Precision/Recall Tradeoff, The ROC Curve, Multiclass Classification, Error Analysis, Multilabel Classification, Multioutput Classification 4. **Decision Trees:** Training and Visualizing a Decision Tree, Making Predictions, Estimating Class Probabilities, The CART Training Algorithm, Computational Complexity, Gini Impurity or Entropy? Regularization Hyperparameters, Regression, Instability, Exercises

	5. **Philosophy, Ethics, and Safety of AI:** The Limits of AI, Can Machines Really Think?, The Ethics of AI
10	**Text Books** 1. Artificial Intelligence: A Modern Approach – Stuart Russell and Peter Norvig – Pearson Education, 2020 2. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow – Aurélien Géron,O'Reilly Media, 2022 (Third Edition)
11	**Reference Books:** 1. The Hundred-Page Machine Learning Book – Andriy Burkov – Andriy Burkov, 2020 (First Edition) 2. Machine Learning for Absolute Beginners – Oliver Theobald – Independently published, 2020 (First Edition) 3. Artificial Intelligence: Foundations of Computational Agents – David L. Poole and Alan K. Mackworth– Cambridge University Press, 2023 (Third Edition) 4. Online Resources: https://artint.info/3e/slides/index.html https://soclibrary.futa.edu.ng/books/ Machine%20Learning%20Engineering%20(Andriy%20Burkov)%20(Z-Library).pdf
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks
13	**Semester End Examination: 60%**
	External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Artificial Intelligence Application Development and Jira (Practical)

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This practical component provides hands-on experience in implementing Artificial Intelligence and machine learning algorithms using modern tools. It covers search techniques, probabilistic modelling, classification, regression, and model evaluation. Students work with datasets to build, validate, and analyse AI systems**.** This course also provides hands-on training in Agile Project Management using Jira Cloud. Students will learn how to create and manage Scrum and Kanban projects, configure workflows, manage issues, generate reports, implement automation, and apply security controls. The course emphasizes practical implementation aligned with industry standards to enhance employability and project management skills.
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1. Implement intelligent agent models and search algorithms to solve AI problems. CO2. Develop probabilistic models and perform inference using appropriate tools. CO3. Build and train machine learning models for classification and regression tasks. CO4. Evaluate AI models using validation techniques and performance metrics. CO5. Develop and deploy a complete AI application using datasets. CO6. Assess bias, fairness, and ethical considerations during AI model development. CO7. To understand Agile project management concepts. CO8. To configure and manage projects using Jira Cloud. CO9. To implement Scrum and Kanban frameworks.	

	CO10. To customize workflows, permissions, dashboards, and automation.
8	Course Outcomes (OC): **OC1.** Implement intelligent agent models and classical search algorithms to solve structured AI problems. OC2. Construct probabilistic models and perform inference using Bayes' rule, Naive Bayes, and Bayesian networks. OC3. Develop and train machine learning models for classification and regression using real-world datasets. OC4. Evaluate and compare AI models using appropriate performance metrics such as accuracy, precision, recall, F1-score, and ROC curves. OC5. Apply cross-validation and basic hyperparameter tuning techniques to improve model performance. OC6. Design and deploy a basic AI application while analysing fairness, bias, and ethical considerations. OC7. Create and configure Jira projects. OC8. Manage issues and workflows effectively. OC9. Implement Scrum and Kanban practices. OC10. Use JQL, reports, and dashboards for project tracking. OC11. Configure permissions, security, and automation rules
9	Modules:-
	Module 1: **Introduction to AI and Intelligent Agents** **1.** Design a Simple Rational Agent by defining PEAS for Vacuum Cleaner Environment and Autonomous Taxi. Classify environments (fully/partially observable**,** deterministic/stochastic, episodic/sequential). Implement a simple table-driven agent in Python. **Problem Solving by Searching (Uninformed Search)** **2.** Given an initial configuration of the 8-puzzle and a goal configuration, write a Python program to find the shortest sequence of moves to reach the goal state using Breadth-First Search (BFS). **3.** Given two water jugs of 4 litres and 3 litres capacity, write a Python program to obtain exactly 2 litres in one jug using Depth-First Search (DFS). [Vary capacity of jugs] Given a weighted graph representing cities and distances between them, write a Python program to find the least-cost path from Arad to Bucharest using Uniform Cost Search (UCS). [Provide any other weighted graph for applying UCS] **Problem Solving by Searching (Informed Search)** **1.** Given an initial configuration of the 8-puzzle and a goal configuration, write a Python program to find the solution using Greedy Best-First Search with the Manhattan Distance heuristic. **2.** Given an initial configuration of the 8-puzzle and a goal configuration, write a Python program to find the shortest path using A* search with the Manhattan Distance heuristic.

	3. Given a weighted graph representing cities and distances between them, write a Python program to find the shortest path from Arad to Bucharest using **A*** search with a heuristic function (straight-line distance to destination). **Bayes' Rule Application** 1. **Scenario:** A medical test is used to detect a particular disease. The probability that a randomly selected person has the disease is 1%. If a person has the disease, the test returns positive with probability 99%. If a person does not have the disease, the test still returns positive with probability 5% (false positive rate). A person takes the test and receives a positive result. **Problem Statement:** Write a Python program to compute the probability that the person actually has the disease given that the test result is positive. **Naïve Bayes' Classification** 1. Using a small dataset (e.g., weather or spam dataset), implement a Naive Bayes classifier using scikit-learn**.** **Probability Reasoning (Bayesian Networks & Inference)** 1. Using a simple example (Burglary–Alarm), create a Bayesian Network using pgmpy**.** 2. Write a program to compute Posterior Probability using Bayesian Networks(Burglar|Alarm = True) **Machine Learning** 1. Using any small dataset (e.g., Iris or a CSV dataset), write a Python program to demonstrate the basic machine learning workflow. a. Load dataset using Pandas. b. Perform basic preprocessing (handling missing values or scaling). c. Split dataset into training and testing sets. d. Train a simple classifier. e. Display training and testing accuracy. 2. Using a suitable regression dataset, implement Linear Regression using scikit-learn. 3. Using the MNIST dataset (digit vs not digit), implement a binary classifier. 4. Implement k-fold cross-validation on a classification model. 5. Implement a Decision Tree classifier using the Iris or any relevant dataset. 6. Implement Decision Tree regression on a regression dataset. 7. Implement k-Nearest Neighbors (kNN) classifier. 8. Implement multiclass classification using Logistic Regression or Decision Tree. **Ethics and Bias Analysis** Select any publicly available dataset (for example, Iris, Adult Income, or a classification dataset of your choice) and analyze it for possible bias and class imbalance**.**
	Module 2: Jira Setup & Project Configuration, Advanced Configuration, Reporting & Governance
	Practical 1: Jira Setup & Project Creation

	A. Create a Jira Cloud account and explore the dashboard. B. Create a Company-managed Scrum/Kanban Project. C. Configure project details **Practical 2: Team Creation & Role Assignment** A. Add Team Members B. Assign Roles **Practical 3: Issue Types & Configuration** A. Create and configure Issue Types (Epic, Story, Task, Bug) B. Create Components and Versions. **Practical 4: Issue Management** A. Create issues under an Epic. B. Assign issues to users and set Priority & Due Date. Practical 5: Workflow Customization A. Design a custom workflow B. Add transitions and conditions. C. Publish and associate workflow with a project. Practical 6: Scrum Board & Sprint Management A. Create and configure Scrum Board. B. Create and start a Sprint. C. Complete Sprint and generate Sprint Report **Practical 7: Kanban Board Management** A. Create a Company-managed Kanban Project. B. Configure Kanban Board C. Track issue progress and generate Control Chart **Practical 8: JQL & Filters** A. Write basic JQL queries (Status, Assignee, Priority). B. Create and save filters. **Practical 9: Reports & Dashboards** A. Generate Burndown Chart and Velocity Chart. B. Create Pie Chart / Issue Statistics Report. C. Create a Custom Dashboard with gadgets. **Practical 10: Permission & Security Management** A. View and modify Project Permission Scheme. B. Configure Project Roles and assign permissions. **Practical 11: Jira Automation** A. Create an Automation Rule (Trigger → Condition → Action). B. Configure rule (e.g., When issue is moved to Done → send email / auto-assign). C. Test and enable automation rule. **Practical 12: Introduction to Jira Mobile App** A. Install and configure the Jira Mobile App. B. View and update issues from mobile (Status, Comment, Assign). Receive notifications and manage real-time updates

10	**Text Books** 1. Artificial Intelligence: A Modern Approach – Stuart Russell and Peter Norvig – Pearson Education, 2020 2. Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow – Aurélien Géron,O'Reilly Media, 2022 (Third Edition) 3. Schwaber, Ken & Sutherland, Jeff. The Scrum Guide. **4.** Official Documentation of Jira Software by Atlassian.
11	**Reference Books** 1. The Hundred-Page Machine Learning Book – Andriy Burkov – Andriy Burkov, 2020 (First Edition) 2. Machine Learning for Absolute Beginners – Oliver Theobald – Independently published, 2020 (First Edition) 3. Artificial Intelligence: Foundations of Computational Agents – David L. Poole and Alan K. Mackworth – Cambridge University Press, 2023 (Third Edition) 4. Online Resources: a. https://artint.info/3e/slides/index.html b. https://soclibrary.futa.edu.ng/books/Machine%20Learning%20Engineering%20(Andriy%20Burkov)%20(Z-Library).pdf 5. Highsmith, Jim. *Agile Project Management: Creating Innovative Products.* Addison-Wesley. 6. Cohn, Mike. *Succeeding with Agile: Software Development Using Scrum.* Addison-Wesley. 7. Rubin, Kenneth S. *Essential Scrum: A Practical Guide to the Most Popular Agile Process.* Addison-Wesley. 8. https://www.atlassian.com/software/jira/guides 9. JIRA 4.3 Documentation User documentation for JIRA Software Server 7.1
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and write-up submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and write-up submission totalling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination. (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hours)** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

IKS
(2)

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Indian Knowledge Systems in Information Technology

Sr.No.	Heading	Particulars
1	**Description the course: Including but Not limited to:**	This course explores Indian Knowledge Systems through philosophy, mathematics, logic, linguistics, and science, highlighting their relevance to information technology. It examines epistemology, algorithms, number systems, logic, Sanskrit grammar, artificial intelligence, mechanical computation, and modern digital preservation. The syllabus connects ancient Indian thought with contemporary computing, interdisciplinary research, ethical technology, and future innovations across education, governance, culture, and emerging global technologies.
2	**Vertical :**	IKS
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives (CO):** CO 1. To Analyze the core methods of knowledge generation in Indian Knowledge System CO 2. To Explore selected contributions from Nyāya, Vaiśeṣika, Vedic mathematics, and ancient computational traditions in modern data science CO 3. To Integrate IKS-informed perspectives into data ethics and responsible AI CO 4. To Leverage IKS domains for domain-specific data science innovation CO 5. To Develop culturally rooted, multidisciplinary problem-solving competence	
8	**Course Outcomes (OC):** OC 1. Apply classical Indian logical, mathematical, and algorithmic traditions OC 2. Compare and contrast epistemological frameworks for reliable knowledge generation in data-intensive environments. OC 3. Integrate ethical and value-based principles from Indian Knowledge Systems OC 4. Analyze and adapt domain-specific insights from traditional Indian knowledge repositories OC 5. Design multidisciplinary, culturally grounded data science solutions for pressing Indian societal challenges.	
9	**Modules: -**	
	Module 1:	

	Unit 1: Introduction to Indian Knowledge Systems and Their Relevance to Information Technology **Philosophical foundations of knowledge in Indian traditions:** Concepts of *Pramana* (means of knowledge), *Pratyaksha* (perception), *Anumana* (inference), *Upamana* (comparison), and *Shabda* (verbal testimony); epistemological frameworks in Nyaya, Vedanta, and Mimamsa schools. **Relationship between Indian Knowledge Systems and Information Technology:** Understanding how structured knowledge representation, systematic reasoning, and rule-based frameworks in Indian traditions relate to modern computational models and information systems. **Relevance of IKS to modern computing and interdisciplinary research:** Role of traditional knowledge in algorithmic thinking, data representation, knowledge management, and artificial intelligence research. **Development of the decimal place value system :** Evolution of the base-10 positional number system in ancient India; transmission of the system to other civilizations; importance of place value in digital computation. **Concept of Zero and its computational significance :** Philosophical and mathematical interpretations of *Shunya*; Brahmagupta's rules for arithmetic operations involving zero; role of zero in modern computing and binary logic. **Binary systems and combinatorics in Pingala's Chandas Shastra :** Study of binary representation of poetic meters; patterns of short and long syllables; combinatorial methods resembling binary enumeration and algorithmic procedures. **Unit 2: Mathematical Foundations and Indian Logic & Its Applications in Ancient India and Their Relevance to Information Technology** **Aryabhata's computational algorithms :** Methods for calculating square roots, cube roots, and trigonometric values; use of iterative procedures and mathematical tables in astronomical calculations. Algorithmic thinking in ancient Indian mathematics : Use of step-by-step computational procedures, rule-based calculations, recursion-like structures, and systematic numerical methods. **Fundamentals of Nyaya logic :** Structure of logical reasoning in the Nyaya school; classification of knowledge, fallacies, and logical arguments. **Indian syllogism and inference structures :** Five-part syllogism consisting of proposition, reason, example, application, and conclusion; role of inference in knowledge validation. **Comparison of Indian and Western logical systems :** Differences between Nyaya logic and Aristotelian logic; multi-step reasoning and contextual inference in Indian philosophy. **Inference systems and reasoning models :** Types of inference such as *Purvavat*, *Sheshavat*, and *Samanyatodrishta*; relevance of these inference mechanisms in logical reasoning. **Kautilya's Arthashastra and strategic decision-making :** Analytical thinking, statecraft strategies, problem-solving frameworks, and structured planning models described in the Arthashastra. **Applications in modern computing systems :** Relevance of logical frameworks in programming languages, knowledge-based systems, rule-based expert systems, and artificial intelligence reasoning mechanisms. **Unit 3: Sanskrit Grammar and Computational Linguistics**

	Panini's Ashtadhyayi and rule-based linguistic systems : Structure of Panini's grammar consisting of nearly four thousand rules; meta-rules and rule hierarchy; systematic description of phonetics and morphology. **Panini's grammar and formal language theory :** Similarity between Panini's generative rules and modern formal grammars; production rules and symbol manipulation. **Generative grammar and rule-based systems : Processes** of derivation and transformation of words and sentences; parallels with modern programming structures and algorithmic rules. **Sanskrit and programming language development :** Precision and unambiguity of Sanskrit syntax; influence on formal language design and structured programming concepts. **Sanskrit grammar and compiler design :** Lexical analysis, syntax rules, and grammar parsing concepts comparable with modern compiler design principles. **Natural Language Processing and Sanskrit :** Application of Sanskrit grammatical structures in machine translation, semantic representation, computational linguistics, and AI-based language processing.
	Module 2:
	Unit 4: Artificial Intelligence and Cognitive Models in Ancient Indian Thought **Concepts of cognition and consciousness in Indian philosophy :** Understanding perception, cognition, memory, and reasoning in Vedanta, Nyaya, and Yoga traditions. **Knowledge representation in classical Indian systems :** Categorization of knowledge, classification of reality, and systematic organization of concepts. **Learning models and mental processes** : Processes of observation, inference, and experiential learning described in philosophical texts. **Artificial entities and automata in Indian mythology :** Descriptions of mechanical beings and artificial constructs in classical narratives and epics. **Predictive astronomy and computational models :** Astronomical calculations in Aryabhatiya and later mathematical texts; prediction of planetary movements using mathematical formulas. **Pattern recognition and classification in ancient Indian texts :** Use of classification systems in linguistics, medicine, and philosophy to organize knowledge systematically. **Unit 5: Mechanical Devices and Early Computational Concepts** **Early counting devices and numerical tools :** Use of counting boards, abacus-like tools, and other calculation aids in trade and astronomy. **Chaturanga and combinatorial reasoning :** Strategic planning and move combinations in the ancient Indian game Chaturanga; connection with modern game theory and computational strategy. **Combinatorics and recursion in Indian mathematics :** Permutation and combination techniques used in poetic meters, mathematics, and astronomy. **Cryptography and coded communication in ancient India :** Use of symbolic codes, secret writing, and encrypted communication in governance and literature. **Indian astronomy and early computational models :** Mathematical models used to compute planetary positions, eclipses, and calendrical calculations. **Comparison with early global computational devices :** Comparative understanding of Indian computational techniques with inventions such as the Antikythera mechanism. **Unit 6: Present Innovations and Future Directions**

	Digitization of traditional knowledge systems : Efforts to digitize manuscripts, preserve classical texts, and create digital repositories. **Knowledge management and digital archives :** Use of databases, metadata standards, and information systems to organize traditional knowledge. **Application of cloud computing and AI in preserving IKS :** Using modern computing technologies to analyze, classify, and preserve ancient texts and data. **Integration of Indian Knowledge Systems with emerging technologies :** Interdisciplinary research connecting IKS with artificial intelligence, data science, and computational linguistics. **Ethical and philosophical perspectives for future technologies :** Applying principles from Indian philosophy to guide ethical AI development and responsible technological innovation.
10	**Text Books** 1. Bhanu Murthy, T. S. A Modern Introduction to Ancient Indian Mathematics. New Age International, 1992. 2. Kak, Subhash and Rao, T. R. N. Computing Science in Ancient India. Center for Advanced Computer Studies, University of Southwestern Louisiana, 1998. 3. Mahadevan, B., Bhat, Vinayak Rajat, and Nagendra Pavana R. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning, 2022. 4. Kapoor, Kapil and Singh, Avadesh Kumar. Indian Knowledge Systems. D.K. Printworld. 5. Plofker, Kim. Mathematics in India. Princeton University Press. 6. Matilal, B. K. The Character of Logic in India. SUNY Press. 7. Cardona, George. Panini: A Survey of Research. Motilal Banarsidass. 8. Radhakrishnan, S. Indian Philosophy (Vol. 1 & 2). Oxford University Press. 9. Briggs, Rick. Knowledge Representation in Sanskrit and Artificial Intelligence. AI Magazine, 1985. 10. https://www.researchgate.net/publication/383541219_Sanskrit's_Role_in_Advancing_AI_A_Comprehensive_Study
11	**Reference Books** • Mahadevan, B., Bhat, Vinayak Rajat, and Nagendra Pavana R. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning, 2022. • Kapil Kapoor and Avadesh Kumar Singh. Indian Knowledge Systems. D.K. Printworld. • Kak, Subhash and Rao, T. R. N. Computing Science in Ancient India. Center for Advanced Computer Studies, University of Southwestern Louisiana, 1998. • Bhanu Murthy, T. S. A Modern Introduction to Ancient Indian Mathematics. New Age International, 1992. • Kim Plofker. Mathematics in India. Princeton University Press, 2009. • C. K. Raju. Cultural Foundations of Mathematics. Pearson Education. • Briggs, Rick. Knowledge Representation in Sanskrit and Artificial Intelligence. AI Magazine, 1985.
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks

	3. Quizzes/ Presentations/ Assignments: 5 marks **4.** Total: 20 marks
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration: 1hr)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

SEM V
Vertical – 1
Electives
(2+2)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Enterprise Java Practical

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This course equips learners with the knowledge and practical skills required to design, develop, and deploy Enterprise Java applications through extensive hands-on laboratory sessions. It covers core and advanced technologies including Servlets, JDBC, Cookies and Session management, JSP, EJB, Hibernate, and the Spring Framework. Emphasis is placed on building scalable, secure, and database-driven web applications. By the end of the course, learners will be able to integrate these technologies effectively to develop robust, real-world enterprise solutions aligned with industry standards.
2	**Vertical :**	Electives
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1:To understand the concept of and develop applications using servlets and database connectivity. CO2:To develop applications that can handle cookies, sessions and file operations. CO3:To understand the concept of and design applications using Java server pages. CO4:To understand the concept of and design applications using Enterprise Java Beans. CO5:To understand the concepts of persistence, Hibernate and develop JPA Applications, Hibernate applications. CO6: To understand Spring Framework	
8	**Course Outcomes (OC):** OC1. Proficiently understand and apply servlets and database connectivity concepts to develop dynamic web applications. OC2. Demonstrate the ability to develop applications capable of managing cookies, sessions, and performing file operations effectively.	

	OC3. Proficient in understanding and designing applications using Java Server Pages (JSP), enabling dynamic and interactive web content creation OC4. Adept at comprehending and designing applications utilizing Enterprise Java Beans (EJB), facilitating the development of scalable and distributed enterprise-level applications OC5. Possess a thorough understanding of persistence concepts, Hibernate framework, and the ability to develop Java Persistence API (JPA) and OC6. Hibernate applications proficiently. OC7. Apply Spring Framework, dependency injection with spring
9	**Modules:-**
	Module 1: 1. Implement the following Simple Servlet applications. a. Create a simple calculator application using servlet. b. Create a servlet for a login page. If the username and password are correct then it says message “Hello <username>” else a message “login failed” c. Create a registration servlet in Java using JDBC. Accept the details such as Username, Password, Email, and Country from the user using HTML Form and store the registration details in the database. 2. Implement the following Servlet applications with Cookies and Sessions. a. Using Request Dispatcher Interface create a Servlet which will validate the password entered by the user, if the user has entered "Servlet" as password, then he will be forwarded to Welcome Servlet else the user will stay on the index.html page and an error message will be displayed. b. Create a servlet that uses Cookies to store the number of times a user has visited servlet. c. Create a servlet demonstrating the use of session creation and destruction. Also check whether the user has visited this page first time or has visited earlier also using sessions. 3. Implement the Servlet IO and File applications. a. Create a Servlet application to upload and download a file. b. Develop Simple Servlet Question Answer Application using Database. c. Create simple Servlet application to demonstrate Non-Blocking Read Operation. 4. Implement the following JSP applications a. Develop a simple JSP application to display values obtained from the use of intrinsic objects of various types. b. Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt, gender-radio button).

	c. Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC 5. Implement the Servlet IO and File applications. a. Create a Servlet application to upload and download a file. b. Develop Simple Servlet Question Answer Application using Database. c. Create simple Servlet application to demonstrate Non-Blocking Read Operation. 6. Implement the following JSP JSTL and EL Applications. a. Create an html page with fields, eno, name, age, designation, salary. Now on submit this data to a JSP page which will update the employee table of database with matching eno. b. Create a JSP page to demonstrate the use of Expression language. c. Create a JSP application to demonstrate the use of JSTL.
	Module 2: 7. Implement the following EJB Applications a. Create a Currency Converter application using EJB b. Develop a Simple Room Reservation System Application Using EJB c. Develop simple shopping cart application using EJB [Stateful Session Bean]. 8. Implement the following EJB applications with different types of Beans a. Develop simple EJB application to demonstrate Servlet Hit count using Singleton Session Beans. b. Develop simple visitor Statistics application using Message Driven Bean [Stateless Session Bean]. c. Develop simple Marks Entry Application to demonstrate accessing Database using EJB. 9. Implement the following JPA applications. a. Develop a simple Inventory Application Using JPA. b. Develop a Guestbook Application Using JPA. c. Create simple JPA application to store and retrieve Book details. 10. Implement the following JPA applications with ORM and Hibernate. a. Develop a JPA Application to demonstrate use of ORM associations. b. Develop a Hibernate application to store Feedback of Website Visitor in MySQL Database c. Develop a Hibernate application to store and retrieve employee details in MySQL Database. 11. Implement the following Hibernate a. Develop an application to demonstrate Hibernate One- To -One Mapping Using Annotation.

	b. Develop Hibernate application to enter and retrieve course details with ORM Mapping c. Develop a five page web application site using any two or three Java EE Technologies. 12. Implement the following using Spring Concept a. Build web application in Java with spring boot3 b. Develop application using Spring Framework, Lightweight c. Containers and Dependency Injection with Spring
10	**Text Books** 1. Java EE 8 Application, David R Heffelfinger, Packt, First, 2017 2. Java EE Essentials, Arun Gupta, O'reilly, First, 2013
11	**Reference Books** 1. Java EE 7 For Beginners, Sharanam Shah & Vaishali Shah, SPD First, 2017 2. Java EE cookbook, Elder Moraes, Packt, First, 2018 3. Advanced Java Programming, Uttam Kumar Roy, Oxford Press , 2015
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 Hrs)** **Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Android Practical

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course introduces students to Android application development using Android Studio. It covers user interface design, Android components, layouts, intents, services, database integration, networking, background processing, media APIs, and security features to develop complete mobile applications. The course emphasizes hands-on practical implementation to build real-world, user-friendly, and data-driven Android applications.
2	**Vertical :**	Electives
3	**Type :**	Practical
4	**Credits :**	2 credits
5	**Hours Allotted :**	60
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To understand the fundamentals of Android application development including project structure, components, and lifecycle management. CO 2. To design and implement user interfaces using various Android layouts and UI controls. CO 3. To develop applications using fragments, intents, events, listeners, and adapters for effective user interaction. CO 4. To implement background processing using services, threads, handlers, and broadcast receivers. CO 5. To develop data-driven applications using SQLite database and JSON-based network communication. CO 6. To implement advanced Android features such as media APIs, notifications, security, and runtime permission handling.	
8	**Course Outcomes (OC):** OC 1: Design and develop basic Android applications using Android Studio and implement Activity lifecycle methods. OC 2: Build responsive user interfaces using various Android layouts, UI controls, and resource files. OC 3: Implement fragments, intents, events, listeners, and adapters to enable effective user interaction and navigation. OC4: Develop applications incorporating services, broadcast receivers, notifications, and background processing mechanisms. OC 5: Create data-driven Android applications using SQLite database and JSON-based network communication. OC 6: Apply advanced Android features including media APIs, threading, security mechanisms, and runtime permission handling in real-world applications.	

9	**Modules**
	Module I: Android Fundamentals & UI Design
	Android Basics and Resources 1. Develop a simple "Hello World" Android application demonstrating project creation, Android Studio interface, Android components, AVD creation, and application execution. 2. Develop an Android application demonstrating the use of Programming Resources (colors.xml, themes.xml, strings.xml, dimens.xml, drawable resources) and apply them in UI components. 3. Develop an Android application demonstrating complete Activity Lifecycle methods and analyze execution using Logcat. **Fragments and Activity Interaction** 4. Develop an Android application to create and integrate a Fragment and demonstrate Fragment lifecycle methods. 5. Develop an Android application implementing Fragment-based UI navigation between multiple screens. 6. Develop an Android application demonstrating communication between Activity and Fragment. **Basic Layout Management** 7. Develop an Android application using LinearLayout (Horizontal and Vertical) to design structured UI. 8. Develop an Android application using RelativeLayout to align UI components dynamically. 9. Develop an Android application using TableLayout to design a structured Login/Registration Form. **Advanced Layout and View Handling** 10. Develop an Android application using ScrollView to display long scrollable content. 11. Develop an Android application using AbsoluteLayout demonstrating fixed positioning. 12. Develop an Android application using FrameLayout to overlay UI elements. **Adapter Based Views** 13. Develop an Android application using ListView with ArrayAdapter and implement item click handling. 14. Develop an Android application using GridView with Custom Adapter and image selection handling. 15. Develop an Android application implementing RecyclerView as a modern alternative to ListView.
	Module 2: Android System Components & Data Handling
	Menus, Dialogs and UI Interaction 1. Develop an Android application implementing AppBar (Toolbar) and Options Menu. 2. Develop an Android application demonstrating AlertDialog and custom DialogFragment. 3. Develop an Android application implementing contextual menus and popup menus. **Intents, Events and Listeners**

	4. Develop an Android application implementing Explicit and Implicit Intents. 5. Develop an Android application demonstrating Event Handling using Listeners (OnClick, OnLongClick, etc.). 6. Develop an Android application using Adapters to populate Spinner or ListView dynamically. **Services, Notifications and Broadcast Receivers** 7. Develop an Android application implementing Started and Bound Services. 8. Develop an Android application implementing Notifications using Notification Manager. 9. Develop an Android application implementing Static and Dynamic Broadcast Receivers. **Database and Networking** 10. Develop an Android application implementing SQLite database CRUD operations. 11. Develop an Android application integrating SQLite with ListView or RecyclerView. 12. Develop an Android application to fetch and parse JSON data from a web service and display it in UI components. **Concurrency, Media and Security** 13. Develop an Android application demonstrating Threads and Handlers for background processing. 14. Develop an Android application implementing Media APIs and Telephone APIs. 15. Develop an Android application implementing Runtime Permissions and security handling in AndroidManifest.xml.
10	**Text Books** 1. Phillips, Bill; Stewart, Chris; Marsicano, Kristin Hardy, Android Programming: The Big Nerd Ranch Guide, Big Nerd Ranch Guides, 5th Edition, 2022. 2. Meier, Reto, Professional Android, Wrox Publishing, 4th Edition, 2018. 3. Griffiths, Dawn; and Griffiths, David, Head First Android Development, O'Reilly Media, 2nd Edition, 2017. 4. Haseman, Chris; and Sutton, Kevin, Learning Android Application Development, Addison-Wesley Professional, 1st Edition, 2020. 5. Darwin, Ian F., Android Cookbook: Problems and Solutions for Android Developers, O'Reilly Media, 2nd Edition, 2017.

11	**Reference Books** 1. Subramanian, S.; and Bali, Rachit, Android Application Development: Basics to Advanced, McGraw-Hill Education, 1st Edition, 2021. 2. Murphy, Shaun, Android Studio 4 Development Essentials – Java Edition, Payload Media, 10th Edition, 2021. 3. Sayed Hashimi, and Satya Komatineni, Mastering Android Application Development, Wiley, 2nd Edition, 2018. 4. Kotlinlang.org, Kotlin for Android Developers (Official Guide), JetBrains Documentation, 2024 (Online Reference). 5. Google Developers Documentation Team, Android Developers Official Guide, Google Inc., 2024 (Online Reference).
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Practical Examination (2 hours duration)
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hrs)** **Certified copy of Journal is compulsory to appear for the practical examination** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Full Stack development MERN Practical

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This practical course provides structured training in full-stack web application development using the MERN stack (MongoDB, Express, React, and Node.js). The course is designed to develop the fundamental skills required to build dynamic, data-driven web applications by integrating frontend and backend technologies in a systematic manner. The course begins with environment setup and core JavaScript programming concepts, forming the foundation for frontend and backend development. Learners progressively build user interfaces using React, implement components with props and state, manage routing, and handle data from external APIs. In the backend module, students develop server-side applications using Node.js and Express, create RESTful APIs, and perform database operations using MongoDB. Emphasis is placed on hands-on implementation of CRUD operations, API creation and integration, data handling, and validation techniques. Learners will gain practical exposure to connecting the frontend with backend services and executing a complete data flow from the user interface to the database. Installation and configuration of development tools, JavaScript fundamentals and ES6 features, React components and state management, routing and API handling, Express server development, REST API implementation, MongoDB CRUD operations, frontend-backend integration, and validation and execution of a complete MERN-based web application.
2	**Vertical :**	Practical
3	**Type :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
4	**Credits :**	60 Hours
5	**Hours Allotted :**	50 Marks
6	**Marks Allotted:**	Practical

7	**Course Objectives(CO):** **CO1:** To install and configure the MERN development environment and execute basic JavaScript programs using Node.js. **CO2:** To apply fundamental JavaScript concepts such as operators, functions, arrays, objects and control structures in program development. **CO3:** To develop basic user interfaces using React and understand component-based architecture. **CO4:** To implement React components using props, state and event handling for interactive applications. **CO5:** To implement routing and handle data from external APIs in React applications. **CO6:** To develop simple server-side applications and APIs using Node.js and Express. **CO7:** To perform database creation and CRUD operations using MongoDB and connect it with Node.js. **CO8:** To integrate frontend and backend for sending, receiving and displaying data. **CO9:** To implement data management operations such as insert, update and delete using the MERN stack. **CO10:** To execute, test and validate the complete workflow of a full stack web application.
8	**Course Outcomes (OC):** **OC1:** Learners will be able to install required development tools and run basic JavaScript programs using Node.js. **OC2:** Learners will be able to develop JavaScript programs using essential programming constructs and functions. **OC3:** Learners will be able to create and execute basic React applications using functional components. **OC4:** Learners will be able to design interactive React components using props, state and events. **OC5:** Learners will be able to develop multi-page React applications and display data retrieved from APIs. **OC6:** Learners will be able to create simple backend servers and APIs using Express and Node.js. **OC7:** Learners will be able to create databases and perform CRUD operations using MongoDB. **OC8:** Learners will be able to connect React frontend with backend APIs and display dynamic data. **OC9:** Learners will be able to perform CRUD operations through integrated frontend and backend applications. **OC10:** Learners will be able to test and execute a complete MERN-based application ensuring proper data flow and validation.
9	**Modules:-**
	Module 1: Frontend Development using React

	1.**Environment Setup and JavaScript Basics** a. Install Node.js and Visual Studio Code, and verify installation. b. Create a simple Node.js file and run it using the terminal. c. Write a Node.js script demonstrating basic JavaScript commands (console log, variables, and arithmetic operations). d. Create and run a simple program using variables and functions. e. Demonstrate use of operators and control statements in JavaScript f. Write programs using loops and template literals. 2. **Core JavaScript Concepts for MERN** a. Create functions and call them with parameters. b. Work with arrays and objects in JavaScript. c. Demonstrate use of arrow functions. d. Write a program using array methods (map, filter, reduce). e. Create a program that manipulates objects and displays output. f. Implement a small JavaScript program combining functions, arrays, and objects. 3.**React Introduction and JSX** a. Create a React application using Vite/CRA and display "Welcome to MERN Stack Development". b. Display dynamic data using JSX. c. Declare variables and write a function to calculate the sum of two numbers and display the result. d. Apply basic CSS styling to a React component e. Create a simple functional component and render it in App.js. f. Create a React component that displays the current date and time dynamically using JavaScript. 4.**React Components, Props, and State** a. Create multiple components and render them on a single page. b. Pass data between components using props c. Use the useState hook to manage and update data. d. Create a form component to accept user input and display it dynamically. e. Display a list of items stored in the state using map(). f. Build a small interactive component (e.g., counter or to-do list). 5.**React Routing and Data Handling** a. Install and configure React Router in the React application. b. Create multiple pages and implement navigation between them. c. Use Link and Routes to move between pages. d. Fetch sample data from a public API using Axios. a. Display fetched data in table or list format. b. Implement search or filter functionality on displayed data.
	Module 2: Backend Development Using Node.js, Express, and MongoDB

	6.**Node.js and Express Fundamentals** a. Install Express and create a simple server. b. Display a message on the browser using the server response. c. Create different routes in Express. d. Send a JSON response from the server. e. Test routes using browser or Postman. f. Run the server and observe the output in the console. 7.**MongoDB Basics** a. Install and start MongoDB. b. Create a database and collection. c. Insert records in MongoDB. d. View records from the collection. e. Update and delete records. f. Connect MongoDB with Node.js. 8.**API Creation and Integration** a. Create a simple REST API using Express b. Implement a GET API to retrieve data. c. Fetch data from the API in a React application. d. Send data from React to the server using a POST request. e. Store data in MongoDB. f. Display stored data in React page. 9.**User Data Management** a. Create an API to add user details using Express and MongoDB. b. Design a React form to send user data to the backend. c. Retrieve and display user data from MongoDB. d. Implement an update operation for existing records. e. Implement the delete operation for selected records. f. Display updated records dynamically in React. 10.**Full Stack Execution and Validation** a. Integrate React frontend with Node and Express backend. b. Perform insert, update, and delete operations through UI. c. Display database records dynamically in React. d. Implement frontend form validation. e. Test all APIs using Postman. f. Execute and verify the complete working of the frontend, backend, and database together.
10	**Text Books**

11	**Reference Books** 1. Web Development with Node and Express. Ethan Brown. O'Reilly Media 2nd Edition, 201 4. Learning React Alex Banks & Eve Porcello. O'Reilly Media 2nd Edition, 2020 5. MongoDB: The Definitive Guide Shannon Bradshaw, Eoin Brazil & Kristina Chodorow O'Reilly Media 3rd Edition, 2019 6. Ultimate Full-Stack Web Development with MERN Nabendu Biswas BPB Publications 1st Edition, 2022 7. Full Stack Development with MongoDB, Express, Angular and Node Colin J. Ihrig & Adam Bretz McGraw Hill (TMH) 2nd Edition, 2018
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 hours)** **Certified copy of Journal is compulsory to appear for the practical examination** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Linux Administration Practical

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This course introduces learners to the fundamental and applied concepts of Linux system administration through extensive hands-on laboratory practice. Students will install, configure, secure, and manage a Linux operating system environment using virtualization platforms. The course focuses on developing practical abilities required for entry-level roles such as System Administrator The laboratory-oriented pedagogy enables learners to work in real-world computing
2	**Vertical :**	Elective Major
3	**Type :**	Practical
4	**Credits :**	2 credits
5	**Hours Allotted :**	60 Hrs
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1. Understand the architecture, features, and working environment of the Linux operating system and perform installation using virtualization tools. CO2. Use Linux command-line interface effectively to manage files, directories, text processing, permissions, and software packages. CO3. Administer users, groups, file systems, and storage devices while applying appropriate access control and security practices. CO4. Configure essential networking services including IP addressing, SSH remote access, web server, and network file sharing. CO5. Monitor system performance, manage processes, implement firewall rules, and troubleshoot common system issues. CO6. Automate administrative tasks using shell scripting and scheduling tools such as cron and at.	
8	**Course Outcomes (OC):** OC 1. Install and configure a Linux operating system and work in graphical and command-line environments. OC 2. Perform file handling, text processing, and package management using standard Linux utilities and commands. OC 3. Create and manage users, groups, permissions, and access control lists to secure system resources. OC 4. Configure networking, remote login (SSH), and basic server services such as Apache web server and network storage. OC 5. Analyse and manage processes, partitions, logical volumes, and system performance.	

	OC 6. Write basic shell scripts and schedule automated tasks to manage and monitor system operations.
9	**Modules:-**
	Module 1:
	1. a. Installing Red Hat Enterprise Linux (Virtual Box/ VmWare) b. Linux Graphical Environment i. Display/Login Manager ii. Desktop Environment iii. Linux Directory Structure and File Systems 2. Basic System Commands a. Use input-output redirection (>, >>, |, 2>, etc) b. Use grep and regular expressions to analyse text c. Log in and switch users in multi-user targets d. Working with Text Files cat,cut,less,hesd,tail,sort,wc,grep, awk 3. Essential File Management Tools a. Managing Files ,Using Links b. Working with Archives and Compressed Files c. Create, delete, copy, and move files and directories d. Archive, compress, unpack, and uncompress files using tar, star, gzip, and bzip2 e. Create hard and soft links f. Managing File Ownership, Managing Basic Permissions g. Managing Advanced Permissions, Managing ACLs h. Setting Default Permissions with umask i. Working with User-Extended Attributes j. List, set, and change standard ugo/rwx permissions k. Create and configure set-GID directories for collaboration l. Create and manage access control lists 4. Working with Users, Groups, and Permissions a. Creating and Managing Users b. Creating and Managing Groups c. Using Permissions and Advanced Permissions d. Change passwords and adjust password aging for local user accounts e. Configure superuser access -> su,sudo,root 5. Managing Software a. Managing Software Packages with YUM b. Using yum c. Managing Software Packages with RPM d. Install and update software packages repository from the local file system
	Module 2:
	6. Managing Processes a. Introduction to Process Management b. Managing Shell Jobs

	c. Using Common Command-Line Tools for Process Management d. Using top to Manage Processes e. Using tuned to Optimize Performance f. Identify CPU/memory-intensive processes and kill processes 7. Managing Storage a. Understanding MBR and GPT Partitions b. Managing Partitions and File Systems c. Mounting File Systems d. List, create, delete partitions on MBR and GPT disks e. Configure systems to mount file systems at boot by universally unique ID (UUID) or label f. Add new partitions and logical volumes, and swap to a system non-destructively g. Create, mount, unmount, and use vfat, ext4, and xfs file systems 8. Configuring Networking a. Managing Network Addresses and Interfaces b. Validating Network Configuration c. Managing Network Configuration with nmtui and nmcli d. Configure IPv4 and IPv6 addresses e. Configure DNS-hostname resolution 9. Configuring SSH a. Using SSH and Related Utilities b. Access remote systems using SSH c. Configuring Key-Based Authentication with Passphrases d. Configure key-based authentication for SSH 10. Configuring web server a. Configuring a Basic Apache Server b. Understanding Apache Configuration Files c. Creating Apache Virtual Host 11. Configure Firewall a. Linux Firewall with iptables b. Working with Firewalld 12. Accessing Network Storage a. Using NFS Services b. Using CIFS Services c. Mounting Remote File Systems Through fstab d. Using Automount to Mount Remote File Systems 13. Creating Shell scripts a. automate or monitor tasks/processes b. Getting and setting system information 14. Scheduling Tasks a. Configuring Cron to Automate Recurring Tasks b. Configuring At to Schedule Future Tasks c. Schedule tasks using at and cron
10	Text Books
11	Reference Books 4. Red Hat RHCA 8 Cert Guide by Sander van Vugt

	5. RHCSA Red Hat Enterprise Linux 8 Training and Exam Preparation Guide (EX200) by Asghar Ghori 6. Red Hat Enterprise Linux 8 Administration - Master Linux Administration Skills and Prepare for the RHCSA Certification Exam (Packt, 2021)
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2Hr)** **Certified copy of Journal is compulsory to appear for the practical examination Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Vertical - 4

VSC
(2)
(Practical)

Syllabus
B.Sc. (Information Technology)
(Sem.- V)

Name of the Course: Internet of Things Practical

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This IoT practical course uses Raspberry Pi to build real-world projects with sensors, displays, security modules, GPS, ADCs, relays, and cameras. Students learn Python-based interfacing, automation, data acquisition, and MQTT communication through hands-on exercises, gaining strong foundational skills for developing functional IoT systems and smart applications.
2	**Vertical :**	Electives
3	**Type :**	Theory / Practical
4	**Credits :**	2 credits (1 credit = 1 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To introduce students to the fundamentals of Internet of Things (IoT) through hands-on practical experiments using Raspberry Pi. CO 2. To develop skills in interfacing digital, analog, and communication modules such as sensors, RFID, fingerprint readers, GPS, relays, and displays. CO 3. To enable students to write Python programs for device control, sensor data processing, automation, and real-time applications. CO 4. To familiarize learners with IoT communication protocols such as MQTT for data exchange and remote monitoring. CO 5. To build confidence in assembling complete IoT systems combining hardware, software, and networking components. CO 6. To cultivate problem-solving and prototyping skills essential for designing IoT-based smart solutions.	
8	OC 6. Course Outcomes (OC): OC 7. After completing this course, students will have: OC 8. Ability to Interface and Control Digital & Analog Hardware Components. OC 9. Competence in Implementing Real-World IoT Communication and Automation. OC 10. Skills in Reading, Processing, and Displaying Sensor Data OC 11. Ability to Implement Identity & Security Mechanisms OC 12. Capability to Design and Deploy an Integrated IoT-Based Smart System	

	OC 13. . Ability to Integrate and Program Multiple Hardware Interfaces for Security and Automation OC 14. Proficiency in Designing Multi-Layer Authentication and Access Control Systems OC 15. Competence in IoT Communication and Web-Based Control of Embedded Devices OC 16. Skills in Implementing Intelligent Monitoring Systems Using Raspberry Pi OC 17. Capability to Deploy, Test, and Evaluate Complete Embedded Solutions on Raspberry Pi
9	**Modules:-**
	Module 1:
	1. Create different blinking patterns with LEDs by connecting to GPIO pints of Raspberry Pi. 2. Use Telegram to control Raspberry Pi GPIO. 3. Interface TM1637 with Raspberry Pi and Display current time. 4. Interface GPS module with Raspberry Pi. 5. Interface ADS 1115 with Raspberry Pi to act as oscilloscope. 6. Interface 16x2 LCD with Raspberry Pi using I2C module to display text. 7. Interface RFID module with Raspberry Pi to read and write to RFID tags/cards. 8. Interface Fingerprint Sensor with Raspberry Pi. 9. Interface Pi Camera with Raspberry Pi to capture image and video. 10. Design IoT based home automation system using Raspberry Pi

	Module 2: 11. Interface Fingerprint Sensor and 16X2 LCD with Raspberry and display message RFID UID matched or unmatched. 12. Simulate door lock by interfacing fingerprint sensor and relay moule with raspberry Pi. 13. Implement basic Publisher – Subscriber using MQTT. Publish messages using paho mqtt. Subscribe and control LED. 14. Design a web dashboard using Flask. Control relay module via web. 15. Create two factor authentication system by interfacing fingerprint sensor, RFID and relay with Raspberry Pi. 16. Create a motion detector by interfacing camera and 16X2 display with Raspberry Pi. Print the appropriate message on the display. 17. Design and implement a Node-RED flow on Raspberry Pi that integrates a physical push-button and an LED with a web dashboard. The flow must allow users to toggle the LED from the dashboard and via the physical button. A button press should force the LED ON for 5 seconds and then automatically restore the LED to the dashboard's last state. 18. Write a program to read the CPU%, Memory% and CPU temperature of Raspberry Pi. 19. Design and implement an embedded access-control prototype that grants entry only when either a whitelisted RFID tag is detected or a legitimate fingerprint is verified, provides real-time visual feedback on an LCD, indicates status via an LED, and pulses a relay (active-LOW) to unlock for a fixed interval. The system must log all events (time, method, result) for auditability. 20. Install Windows IoT 10 core on Raspberry pi and run the demo program.

10 and 11	**Reference Books:** 1. Programming the Raspberry Pi: Getting Started with Python — Simon Monk, McGraw-Hill, 3rd ed. 2. Raspberry Pi Cookbook — Simon Monk, O'Reilly, 2nd ed. 3. Exploring Raspberry Pi — Derek Molloy, Wiley. 4. Getting Started with Python on Raspberry Pi — (Raspberry Pi Press / Foundation) 5. Exploring Arduino — Jeremy Blum, 2nd ed., Apress. 6. Practical Electronics for Inventors — Paul Scherz & Simon Monk, McGraw-Hill, 4th ed. 7. **RFID Handbook: Fundamentals and Applications** — Klaus Finkenzeller, Wiley, 3rd ed. 8. **Near Field Communication (NFC): From Theory to Practice** — Vedat Coskun et al., Wiley, 2nd ed. 9. **Handbook of Fingerprint Recognition** — Davide Maltoni et al., Springer, 2nd ed. 10. **Understanding GPS/GNSS: Principles and Applications** — Elliott D. Kaplan & Christopher Hegarty, Artech House 11. **GPS for Land Surveyors** — Jan Van Sickle, CRC Press. 12. **Programming the Raspberry Pi: Picamera2 and libcamera Guides** — (Foundation/Pi Press articles, community books) 13. **MQTT Essentials – A Lightweight IoT Protocol** — Gastón C. Hillar (Packt) 14. **Node-RED: A visual tool for wiring the Internet of Things** — Nick O'Leary & Dave Conway-Jones 15. **Flask Web Development** — Miguel Grinberg, O'Reilly, 2nd ed. *16.* **Designing Connected Products** — Claire Rowland et al., O'Reilly.
12	**Internal Continuous Assessment: 40%**
	Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks..
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration: 2 hours)** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Sem. – VI

Sem. - VI
Vertical – 1
Major Mandatory
(2+2+2+2+2)

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Project Management

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course introduces the fundamental concepts and practices of project management in the context of IT projects. It covers project planning, scheduling, cost estimation, risk management, and quality control using standard tools and techniques. Students learn to manage the complete project life cycle, from initiation to closure, with effective teamwork and communication. The course equips learners with practical skills to successfully plan and execute real-world IT projects
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To introduce fundamental concepts and principles of software project management. CO 2. To understand project planning, estimation, scheduling, and risk management techniques. CO 3. To familiarize students with modern software process models and project evaluation methods. CO4. To develop managerial, leadership, and quality-oriented skills required for successful software projects.	
8	**Course Outcomes (OC):** **After completion of the course, students will be able to:** OC1. Explain core concepts, processes, and challenges of software project management. OC2. Apply project planning, estimation, scheduling, and risk management techniques to software projects. OC3. Compare and select appropriate software process models and project approaches. OC4. Analyze project performance, manage teams and quality, and execute effective project closure.	
9	**Modules:-**	
	Module I: Software Project Management, Evaluation & Planning	
	1. **Introduction to Software Project Management:** Introduction, Importance of Software Project Management, What is a Project? Software Projects versus Other Types of Projects, Contract Management and Technical Project Management,	

	Activities Covered by Software Project Management, Plans, Methods and Methodologies, Categorization of Software Projects. 2. **Project Initiation and Management Concepts**: Project Charter, Stakeholders, Setting Objectives, Business Case, Project Success and Failure, What is Management? Management Control, Project Management Life Cycle, Traditional versus Modern Project Management Practices. 3. **Project Evaluation and Programme Management**: Project Portfolio Management, Evaluation of Individual Projects, Cost–Benefit Evaluation Techniques, Risk Evaluation, Programme Management, Strategic Programme Management, Benefits Management, Reservations about Programme Management. 4. **Project Planning**: Stepwise Project Planning—Project Selection, Scope and Objectives, Infrastructure, Project Characteristics, Product and Activity Identification, Effort Estimation, Risk Identification, Resource Allocation, Plan Review and Execution. 5. **Selection of Project Approach**: Build vs Buy, Choosing Methodologies and Technologies, Software Processes, Choice of Process Models, Structure versus Speed of Delivery.
	Module 2: Estimation, Risk, Quality & Project Closure
	1. **Software Effort Estimation:** Estimation Issues, Over- and Under-Estimation Problems, Estimation Basis, Bottom-up and Top-down Approaches, Parametric Models, Expert Judgment, Estimation by Analogy, Function Point Analysis (Albrecht, Mark II, COSMIC), COCOMO II, Cost Estimation, Staffing Patterns, Schedule Compression, Capers Jones Estimation Rules. 2. **Software Process Models:** Waterfall Model, Spiral Model, Prototyping, Incremental Delivery, RAD, Agile Methods, Extreme Programming (XP), Scrum, Lean Software Development, Managing Iterative Processes, Selection of Appropriate Process Models. 3. **Activity Planning & Scheduling:** Project Scheduling, Network Planning Models, Critical Path Method, Activity Float, Project Duration Reduction, Activity-on-Arrow Networks. 4. **Risk Management & Resource Allocation**: Risk Categories, Risk Identification, Assessment, Planning, Boehm's Top 10 Risks, PERT Technique, Monte Carlo Simulation, Critical Chain Concepts, Resource Identification and Scheduling, Cost Schedules. 5. **Monitoring, Quality & Closure:** Earned Value Analysis, Change Control, Software Configuration Management (SCM), Managing People and Teams, Leadership and Communication, Software Quality Concepts, Quality Models (ISO 9126), Quality Planning, Testing, Software Reliability, Project Closure Process, Financial Closure, Project Closeout Report.
10	**Text Books** 1. Software Project Management – Dr. Pallawi Bulakh – *Nirali Prakashan*, 2022 2. Software Project Management– Neelima Padmawar, Deepak Kapase, Pritam Ahire, Vandana G. Dixit & Tulshiram Sul – *TechKnowledge Publications*, 2023 3. Software Project Management – *Er. Rishabh Anand (Reprint editions in circulation– S.K. Kataria & Sons*, 3rd Edition (reprint 2025)

11	Reference Books ● Software Project Management ,Bob Hughes, Mike Cotterell, Rajib Mall TMH 6th 2018 ● Project Management and Tools & Technologies – An overview Shailesh Mehta ,SPD, 1st 2017 ● Software Project Management, Walker Royce ,Pearson, 2005
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks Total: 20 marks
13	**Semester End Examination: 60%**
	Format of Question Paper: External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:1 hour)** Q1: Attempt any three (out of five/Six) from Module 1 (15 marks) Q2: Attempt any three (out of five/Six) from Module 2 (15 marks)

Name of the Course: Computer Security

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Security course introduces the fundamental concepts and practices of information security, including risk analysis, secure system design, authentication, encryption, database security, and network protection. It covers security technologies such as firewalls, intrusion detection systems, and operating system security models. The course emphasizes protecting information assets from cyber threats and unauthorized access. With the increasing demand for cybersecurity in industries like banking, IT, healthcare, and e-commerce, this course prepares students for careers in cybersecurity and information protection.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1. Understand fundamental concepts, evolution, and importance of information security. CO2. Learn risk analysis, threat identification, and secure design principles. CO3. Understand authentication, authorization, encryption, and database security mechanisms. CO4. Study network device security, firewall technologies, and network protection strategies. CO5. Understand intrusion detection, prevention systems, and security monitoring tools. CO6. Learn operating system security models and trusted computing standards.	
8	**Course Outcomes (OC):** OC1. Explain the importance, evolution, and methodology of information security. OC2. Analyze risks, threats, attacks, and apply secure design principles. OC3. Implement authentication, authorization, encryption, and database security techniques. OC4. Apply network security controls including firewalls and network hardening. OC5. Understand intrusion detection systems and security monitoring tools. OC6. Evaluate operating system security models and apply trusted computing concepts.	
9	**Modules:-**	

	Module 1:
	1. **Information Security Overview:** The Importance of Information Protection, The Evolution of Information Security, Justifying Security Investment, Security Methodology, how to Build a Security Program, The Impossible Job, The Weakest Link, Strategy and Tactics, Business Processes vs. Technical Controls. Risk Analysis: Threat Definition, Types of Attacks, Risk Analysis, Secure Design Principles: The CIA Triad and Other Models, Defense Models, Zones of Trust, Best 2. **Authentication and Authorization:** Authentication, Authorization Encryption: A Brief History of Encryption, Symmetric-Key Cryptography, Public Key Cryptography, Public Key Infrastructure. Database Security: General Database Security Concepts, Understanding Database Security Layers, Understanding Database- Level Security, Using Application Security, Database Backup and Recovery, Keeping Your Servers Up to Date, Database Auditing and Monitoring. 3. **Network Device Security**: Switch and Router Basics, **Network Hardening Firewalls**: Overview, The Evolution of Firewalls, Core Firewall Functions, Additional Firewall Capabilities, Firewall Design.
	Module 2: 1. **Intrusion Detection and Prevention Systems:** IDS Concepts, IDS Types and Detection Models, IDS Features, IDS Deployment Considerations, Security Information and Event Management (SIEM). Voice over IP (VoIP) and PBX Security: Background, VoIP Components, VoIP Vulnerabilities and Countermeasures, PBX, TEM: Telecom Expense Management. **Operating System Security Models**: Operating System Models, Classic Security Models, Reference Monitor, Trustworthy Computing, International Standards for Operating System Security
	2. **Virtual Machines and Cloud Computing**: Virtual Machines, Cloud Computing. **Secure Application Design**: Secure Development Lifecycle, Application Security Practices, Web Application Security, Client Application Security, Remote Administration Security. **Physical Security:** Classification of Assets, Physical Vulnerability Assessment, Choosing Site Location for Security, **Securing Assets:** Locks and Entry Controls, Physical Intrusion Detection
10	**Text Books** 1. William Stallings, Cryptography and Network Security: Principles and Practice, 8th Edition, Pearson, 2023. 2. Principles of Information Security by Ph.D. Whitman, Michael E. Herbert J. Mattord , 7th Edition, Cengage Learning, 2021 3. Cloud Security: A Comprehensive Guide to Secure Cloud Computing 1st Edition, by Ronald L. Krutz
11	**Reference Books**

	1. The Complete Reference: Information Security Mark Rhodes- Ousley McGraw-Hill Second 2013 2. Essential Cybersecurity Science Josiah Dykstra O'Reill Fifth 2017 3. Principles of Computer Security: CompTIA Security+ and Beyond Wm.Arthur Conklin,Greg White McGraw Hill Second 2010
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks **4.** Total: 20 marks
13	**Semester End Examination: 60%**
	Format of Question Paper: External Examination(30 Marks)– 1 hr duration
14	**Format of Question Paper:** (**Semester End Examination: 30 Marks. Duration:)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Data Analytics and Business Intelligence

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	Business Analytics equips students with the conceptual foundations, quantitative techniques, and practical tool-based skills required to extract meaningful, decision-relevant insights from business data. The course occupies a critical position in the curriculum — it synthesises students' prior knowledge of programming, databases, and statistics, and channels it towards real-world organisational problem-solving across domains such as marketing, finance, human resources, supply chain, and e-commerce. Foundations of Business Analytics & Data Analysis — builds the conceptual and technical groundwork: the four analytics types (Descriptive, Diagnostic, Predictive, Prescriptive), data collection from structured and unstructured sources, data cleaning and transformation (handling missing values, normalisation, standardisation), Exploratory Data Analysis (EDA) covering measures of central tendency and dispersion, correlation, covariance, and data visualisation, followed by practical mastery of industry analytics tools — Microsoft Excel (statistical functions, PivotTables, charts), Power BI, and Tableau. Predictive Analytics & Business Applications — advances into predictive modelling: Simple Linear Regression and Time Series basics, Classification (Decision Trees), Clustering (K-Means), Market Basket Analysis (Association Rules: Support, Confidence, Lift), and concludes with a business decision-making capstone covering interpretation of results, industry case studies, and the ethical dimensions of data analytics.
2	**Vertical :**	Major
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory)

5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. To introduce students to the fundamental concepts and significance of Business Analytics in modern organizations. CO 2. To develop an understanding of data collection methods, pre-processing techniques, and exploratory data analysis. CO 3. To familiarize students with statistical tools and analytical techniques used in business decision-making. CO 4. To provide knowledge of predictive models such as regression, classification, and clustering. CO 5. To enable students to interpret analytical results and apply them to real-world business scenarios.	
8	**Course Outcomes (OC):** OC 1. Explain the key concepts and types of Business Analytics. OC 2. Perform data cleaning, transformation, and exploratory data analysis using analytical tools. OC 3. Apply statistical and predictive techniques to analyze business data. OC 4. Identify patterns and relationships using clustering and association rule methods. OC 5. Interpret and communicate analytical findings to support effective business decision-making.	
9	**Modules:-**	
	Module 1: Foundations of Business Analytics & Data Analysis	
	5. Introduction to Business Analytics: Meaning and importance of Business Analytics, Types of Analytics: Descriptive, Diagnostic, Predictive, Prescriptive, Applications in Marketing, Finance, HR, Operations 6. Data Collection and Pre-processing: Sources of Business Data (Structured & Unstructured), Data Cleaning and Transformation Handling Missing Values, Data Normalization and Standardization 7. Exploratory Data Analysis (EDA): Measures of Central Tendency (Mean, Median, Mode), Measures of Dispersion (Variance, Standard Deviation), Correlation and Covariance, Data Visualization (Bar chart, Pie chart, Histogram, Line graph) 8. Tools for Business Analytics: Introduction to Excel for Analytics Basic Statistical Functions, Pivot Tables and Charts, Power BI, Tabulae	
	Module 2: Predictive Analytics & Business Applications 5. Introduction to Predictive Analytics: Concept of Prediction, Regression Analysis (Simple Linear Regression), Time Series Basics 6. Classification and Clustering: Concept of Classification, Decision Trees (Basic idea), K-Means Clustering (Basic concept and steps)	

	7. Market Basket Analysis: Concept of Association Rules, Support, Confidence, Lift, Business Use Cases (Retail, E-commerce) 8. Business Decision Making: Interpreting Analytical Results, Case Studies in Business Analytics, Ethical Issues in Data Analytics
10	**Text Books** 1. Camm, Cochran, Fry, Ohlmann, Anderson, Sweeney, Williams - Essentials of Business Analytics, Cengage Learning. 2. James Evans, Business Analytics, Pearson. 3. Albright Winston, Business Analytics - Data Analysis - Data Analysis and Decision Making, Cengage Learning, Reprint 4. Sahil Raj, Business Analytics, Cengage Learning. 5. Ratnoo & Ratnoo, Essentials of R for Data Analytics, Wiley
11	**Reference Books** 1. Seema Acharya & D. R. Khanna – Fundamentals of Business Analytics 2. Thomas W. Miller – Modeling Techniques in Predictive Analytics 3. Ramesh Sharda, Dursun Delen, Efraim Turban – Business Intelligence: A Managerial Perspective on Analytics
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks **4.** Total: 20 marks
13	**Semester End Examination: 60%**
	Format of Question Paper: External Examination (30 Marks)– 1 hr duration
14	**Format of Question Paper: (Semester End Examination: 30 Marks. Duration:1 hour)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Data Analytics and Business Intelligence and Computer Security in Practical

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	The subject gives hands-on, laboratory counterpart to the Business Analytics Theory course, offered as a Major Practical in the sixth and final semester of the T.Y.I.T. programme under NEP 2020. Where the theory course builds conceptual and quantitative foundations, this practical course develops the applied, tool-based proficiency that employers directly evaluate in analytics roles. Students graduate having worked on real-world business datasets using three industry-standard platforms: Microsoft Excel, Python/R, and Microsoft Power BI — the exact tool trifecta encountered in Business Analyst, Data Analyst, MIS Executive, and BI Developer job descriptions. The course spans the full analytics pipeline from data warehousing to predictive modelling: importing and analysing data warehouse data in Excel using PivotTables and PivotCharts; performing OLAP-style cube analysis; applying What-If Analysis tools (Goal Seek, Scenario Manager, Data Tables) for business scenario modelling; implementing classification algorithms (Decision Trees, k-Nearest Neighbours) and clustering algorithms (K-Means) using Python (scikit-learn) or R (caret/kmeans); and building Simple and Logistic Regression models on data warehouse datasets. It focuses on lookup-based data retrieval, dimensional modelling (ROLAP, MOLAP, HOLAP cube design with fact and dimension tables), and professional Power BI dashboard development on real-world datasets including Sales data and the Netflix content dataset. Security in Computing Practical builds practical skills in threat detection, encryption, and secure system design through lab simulations. This lab-based course equips students with tools to secure computing environments, covering vulnerabilities, access controls, and ethical hacking basics. It familiarizes students with common cyber threats

		Enable hands-on configuration of firewalls, IDS/IPS, and secure protocols Develop skills in vulnerability scanning and basic penetration testing in controlled environments.
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1. To understand the fundamentals of Data Warehousing including data warehouse architecture, ETL process, OLAP operations, and cube models (ROLAP, MOLAP, HOLAP) CO2. To develop analytical skills using Microsoft Excel by applying Pivot Tables, Pivot Charts, Lookups (VLOOKUP, HLOOKUP), What-If Analysis, and data visualization techniques. CO3. To implement predictive analytics techniques such as Linear Regression and Logistic Regression for real-world datasets. CO4. To design interactive dashboards using Power BI for business decision-making. CO5. To configure and secure network devices using industry-standard protocols. CO6. To design and implement Access Control Lists (ACLs) CO7. To implement Layer 2 security mechanisms on Cisco switches. CO8. To configure IOS Intrusion Prevention Systems (IPS), site-to-site IPsec VPNs and Cisco ASA basic firewall settings using the CLI CO9. To develop host-based security investigation skills using the Windows a. SysInternals Suite and to perform network reconnaissance and attack surface b. mapping using the Nmap port scanner. **CO10.** To capture, filter, and analyse live network traffic using Wireshark, interpreting protocol behaviour across TCP (3-way handshake	

8	**Course Outcomes (OC):** OC 1. Apply data warehousing concepts and perform OLAP analysis using Excel and cube models. OC 2. Perform data analysis and visualization using Pivot tables, What-If analysis, and dashboards. OC 3. Implement data mining techniques such as classification and clustering using R/Python. OC 4. Apply regression techniques (Linear and Logistic) for predictive analysis OC 5. Design dimensional models and create BI dashboards using Power BI. OC 6. Configure routers for Syslog event logging OC 7. Configure, apply, and verify Extended Numbered IPv4 ACLs and IPv6 ACLs OC 8. Configure a Zone-Based Policy Firewall (ZPF) on a Cisco router OC 9. Implement Layer 2 security on Cisco switches OC 10. Configure IOS IPS using the CLI — enabling IPS OC 11. Use the Windows SysInternals Suite to investigate system security OC 12. Capture and analyse live network traffic using Wireshark
9	**Modules:-**
	Module 1:
	1. Perform the analysis for the following: a. Import the data warehouse data in Microsoft Excel and create the Pivot table and Pivot Chart. b. Import the cube in Microsoft Excel and create the Pivot table and Pivot Chart to perform data analysis. 2. Apply the what – if Analysis for data visualization. Design and generate necessary reports based on the data warehouse data. Use Excel. 3. Perform the data classification using classification algorithm using R/Python. 4. Perform the data clustering using clustering algorithm using R/Python. 5. Perform the Linear regression on the given data warehouse data using R/Python. 6. 6. Perform the logistic regression on the given data warehouse data using R/Python. 7. Perform lookups using Microsoft Excel a. Perform vlookup on any products data. b. Perform hlookup on any products data. 8. Create the cube with suitable dimension and fact tables based on ROLAP, MOLAP and HOLAP model. 9. Create dashboard in POWER BI of Sales data. Use scattered chart and cards. 10. Create the dashboard in POWER BI of Netflix dataset.
	Module 2:
	21. Configure Routers for a. Syslog, NTP, and b. SSH Operations

	c. OSPF MD5 authentication. 22. Configure AAA Authentication a. Configure a local user account on Router and configure authenticate on the console and vty lines using local AAA b. Verify local AAA authentication from the Router console and the PC-A client 23. Configuring Extended ACLs a. Configure, Apply and Verify an Extended Numbered ACL 24. **Configuring IPv6 ACLs** 25. Configure IP ACLs to Mitigate Attacks a. Verify connectivity among devices before firewall configuration b. Use ACLs to ensure remote access to the routers is available only from management station PC-C. c. Configure ACLs on to mitigate attacks. 26. Configuring a Zone-Based Policy Firewall (ZPF) a. Verify connectivity among devices before firewall configuration. b. Configure a zone-based policy (ZPF) firewall on R3. c. Verify ZPF firewall functionality using ping, SSH, and a web browser 27. DHCP Snooping and ARP Inspection 28. Configure IOS Intrusion Prevention System (IPS)Using the CLI a. Enable IOS IPS. b. Configure logging. c. Modify an IPS signature. d. Verify IPS 29. Implementing Layer 2 security a. Assign the Central switch as the root bridge. b. Secure spanning-tree parameters to prevent STP manipulation attacks. c. Enable port security to prevent CAM table overflow attacks 30. Implementing Layer 2 VLAN Security a. Enable trunking and configure security on the new trunk link 31. Configure and Verify a Site-to-Site IPsec VPN Using CLI 32. Configuring ASA Basic Settings and Firewall Using CLI 33. Exploring Processes, Threads, Handles, and Windows Registry (SysInternals Suite) 34. Use a Port Scanner to Detect Open Ports a. Nmap, a port scanner and network mapping, tool to detect open ports 35. Using Wireshark a. to Observe the TCP 3-Way Handshake b. to Examine a UDP DNS Capture 36. to Examine FTP and TFTP Captures
10	**Text Books**

	5. **Microsoft Excel 365 Bible** – Michael Alexander & Richard Kusleika Publisher: Wiley *(Covers Pivot Tables, What-If Analysis, Lookups, Data Analysis Tools.)* 6. **Python for Data Analysis** – Wes McKinney Publisher: O'Reilly *(Hands-on implementation of regression and data analysis using Python.)* 7. **R for Data Science** – Hadley Wickham & Garrett Grolemund Publisher: O'Reilly(*Practical guide for clustering, classification and visualization using R.)* 8. **Practical Business Intelligence with Power BI** – Brett Powell *Publisher: Apress (Step-by-step dashboard development.)*
11	**Reference Books** 1. **Business Intelligence Guidebook** – Rick Sherman *Practical BI implementation strategies.* 2. **Applied Predictive Modeling** – Max Kuhn & Kjell Johnson *Regression and classification practical modeling.* 3. 3. **Power BI Cookbook** – Brett Powell *Advanced dashboard techniques and case studies.* 4. CCNA Security 2.0 Lab Manual
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 Hours)** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Cloud Computing and Testing Tools Practical

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	Cloud Computing Practical is the hands-on laboratory companion. This lab-based course bridges theoretical cloud concepts with applied, tool-based experience, equipping students with the skills to deploy, configure, and simulate virtualised cloud environments — the exact competencies demanded in roles such as Cloud Engineer, System Administrator, DevOps Associate, and Cloud Solutions Architect. the virtualisation and simulation foundation: installing and configuring VMware ESXi Server to manage bare-metal hypervisor deployments; implementing virtual machines on the open-source Xen Server (Type-1 paravirtualisation); exploring native virtualisation using Microsoft Hyper-V (hardware-assisted Type-1 hypervisor); followed by setting up the CloudSim simulation toolkit (Java JDK + Eclipse IDE), creating and simulating a cloud data centre with configurable host resources, and provisioning virtual machines within the CloudSim environment Software Testing Tools Practical is a hands-on laboratory course that provides students with direct, tool-based experience across the complete spectrum of modern software testing — from web UI automation and unit testing to load testing, behavior-driven development, and API testing. The course covers ten industry-standard tools and frameworks — Selenium IDE, Selenium RC, Selenium WebDriver, JUnit, TestNG, NUnit, AutoIT, Apache JMeter, Cucumber, and SoapUI/Postman — equipping students for roles such as QA Engineer, Automation Test Engineer, SDET (Software Developer in Test), and Performance Test Engineer.
2	**Vertical :**	Major
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):**	

	CO1. To understand and implement Types of hypervisor-based virtualization platforms — VMware ESXi, Xen, and Microsoft Hyper-V **CO2.** To install and configure the CloudSim simulation toolkit (Java JDK + Eclipse IDE), understand its layered architecture, and successfully run basic simulation programs **CO3.** To create and simulate cloud data centers and virtual machines in CloudSim. **CO4.** To simulate task scheduling in a cloud environment by creating multiple cloudlets, assigning them to virtual machines using scheduling algorithms, and to model multiple distributed data centers CO5. To introduce students to web UI test automation using Selenium IDE, Selenium RC, and Selenium WebDriver, covering record-and-playback, scripted test creation, assertion-based verification, and screenshot capture. CO6. To develop proficiency in writing and executing unit test cases using JUnit (Java) and NUnit (.NET), including assertion methods, test lifecycle annotations, and coverage of both positive and negative test scenarios. CO7. To enable students to organise and manage automated test suites using TestNG lifecycle annotations, and to handle native desktop dialogs within browser tests by integrating AutoIT scripts with Selenium WebDriver. CO8. To equip students with skills to perform performance and load testing using Apache JMeter, including designing test plans, simulating concurrent user traffic, and interpreting throughput, response time, and error rate metrics. CO9. To introduce Behavior-Driven Development (BDD) using Cucumber with Gherkin syntax for writing human-readable acceptance tests, and to API testing using SoapUI and Postman for verifying RESTful web service correctness, payload integrity, and scalability under load.
8	**Course Outcomes (OC):** OC1. Install and configure Type-1 hypervisors (VMware ESXi, Xen, and Microsoft Hyper-V) on bare-metal or lab environments; create, configure, and manage virtual machines OC2. Install the CloudSim toolkit (JDK + Eclipse IDE + CloudSim library), import the project, and execute sample simulation programs successfully OC3. Create and simulate a cloud data center in CloudSim by programmatically defining Data Center objects and host configurations OC4. Create cloudlets with defined length, file size, and output size OC5. Simulate task scheduling by creating multiple cloudlets, assigning them to VMs via scheduling algorithms, and model multiple distributed data centers with independent host OC6. Execute cloud simulations and analyse performance metrics (task execution time, finish time, actual CPU time, and resource utilization OC7. Record and replay browser-based test scenarios using Selenium IDE with element assertions; write and execute Selenium RC scripts and Selenium WebDriver tests OC8. Write JUnit test cases in Eclipse to verify web application search functionality; OC9. Configure and run a TestNG test suite with @Test, @BeforeMethod,

	@AfterMethod, @BeforeClass, and @AfterClass lifecycle annotations; write and integrate an AutoIT OC10. Implement NUnit test cases for a .NET authentication system covering positive and negative scenarios; design Apache JMeter test plans OC11. Write Cucumber .feature files in Gherkin (Given–When–Then) and implement Java step definition classes linking each step to Selenium WebDriver code
9	**Modules:-**
	Module 1:
	1. Implement VMWAre ESXi Server. 2. Implementing VMs on Xen Server 3. Native Virtualization using Hyper-V 4. Installation of CloudSim Environment 5. To create and simulate a cloud data center using CloudSim. 6. To create virtual machines in CloudSim environment. 7. Virtual Machine Scheduling in cloud sim 8. Cloudlet Creation and Execution 9. To simulate task scheduling using CloudSim 10. Simulation of Multiple Data Centers 11. Performance Analysis of Cloud Simulation 12. Study of CloudSim Architecture
	Module 2:
	13. Selenium IDE: a. Install Selenium IDE in your browser of choice. b. Record a simple test scenario such as logging into the web application. c. Add assertions to verify elements on the page. d. Export the test case to a programming language of your choice (e.g., Java). 14. Selenium RC Tools: a. Install Selenium RC and set up a test environment. b. Write a test script using Selenium RC to automate a registration form on the web application. c. Execute the test script and verify the results. 15. Creating JUnit Tests using Eclipse: a. Set up a Java project in Eclipse for automation testing. b. Write a JUnit test case to verify the search functionality of the web application. c. Run the JUnit test case and analyze the test results. 16. Web Testing using WebDriver Tool: a. Set up WebDriver in your preferred programming language (e.g., Java). b. Write a WebDriver test script to automate the checkout process of the web application. c. Implement synchronization techniques to handle dynamic elements on the page. d. Execute the WebDriver test script and capture screenshots for each step. 17. Automation using AutoIT Tool:

	a. Install AutoIT and familiarize yourself with its scripting language. b. Write an AutoIT script to automate a file upload process on the web application. c. Integrate the AutoIT script with your Selenium WebDriver test script. d. Execute the combined test script and verify the file upload functionality. 18. Setting Up TestNG Testing Framework: a. Set up a TestNG testing framework in your preferred IDE (e.g., IntelliJ IDEA, Eclipse). Create a test class with multiple test methods to validate different functionalities of a sample web application. Utilize TestNG annotations such as @Test, @BeforeMethod, @AfterMethod, etc., to organize and execute your test cases effectively. 19. Unit Testing with Junit a. Develop a Java class representing a simple calculator with methods for addition, subtraction, multiplication, and division. Write Junit test cases to verify the correctness of each operation. Utilize Junit assertions to validate expected outcomes and ensure that the calculator functions as expected under various scenarios. 20. Unit Testing with Nunit and Load Testing with Jmeter: a. Implement a .NET class representing a user authentication system with methods for login and logout. Create Nunit test cases to validate the authentication functionality, including positive and negative test scenarios. Use Nunit's Assert methods to verify the behavior of the authentication system under different conditions. b. Set up a load testing environment using Apache Jmeter to assess the performance of a web application under heavy user load. Create Jmeter test plans to simulate concurrent user traffic accessing different pages of the application. Analyze Jmeter test results to identify performance bottlenecks and optimize system performance. 21. Behavior-Driven Development (BDD) with Cucumber/Cypress: a. Implement behavior-driven development (BDD) practices using Cucumber to define and execute acceptance tests for a web application. Write feature files in Gherkin syntax to describe various application features and scenarios. Develop step definitions in your preferred programming language to automate the execution of Cucumber scenarios. 22. API Testing with SoapUI/POSTMAN: Use SoapUI to perform functional and performance testing of a RESTful web service. Create SoapUI test suites to validate different API endpoints and verify the correctness of request and response payloads. Configure SoapUI load tests to simulate concurrent user interactions and measure the service's scalability and responsiveness.
10 **11**	**Text Books** **Reference Books** 5. **GitHub** 6. **Vmware VCA lab manual** *7.* https://clouds.cis.unimelb.edu.au/cloudsim/
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through:

	Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totalling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks practical exam of 2 hours duration
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2 Hours)** **Practical Slip:** Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

SEM VI
Vertical – 1
Electives
(2+2)

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Geographical Information System Practical

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course provides practical training in Geographic Information Systems (GIS) using QGIS. Students learn to manage, analyze, and visualize spatial data including vector and raster datasets. The course covers map creation, spatial analysis, georeferencing, projections, and automation using processing tools.
2	**Vertical :**	Elective
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** **CO1:** Understand the fundamentals of GIS concepts and spatial data types. **CO2:** Develop skills in managing, editing, and analyzing vector and raster datasets. **CO3:** Apply spatial analysis techniques to solve real-world geographic problems. **CO4:** Create raster data analysis on real applications.	
8	**Course Outcomes (OC):** **OC1** Demonstrate the ability to load, manage, and visualize spatial datasets using QGIS. **OC2:** Perform vector and raster analysis operations including buffering, overlay, and terrain analysis. **OC3:** Apply georeferencing, projection management, and spatial querying techniques effectively. **OC4:** Design and present GIS-based solutions through map layouts and analytical reports.	
9	**Modules:-**	
	Module 1: Getting Started with QGIS 1. QGIS Installation & Interface Overview • Install QGIS 3.40. • Explore panels: Browser, Layers, Map Canvas. • Load sample vector and raster data. (Training Manual: “An Overview of the Interface”, “Adding your first layers”)	

	2. Navigating Map Canvas & Basic Tools Pan, zoom, identify features. Use selection tools and measure tools. Explore project properties**.** (Based on "Navigating the Map Canvas") **3. Symbology & Layer Styling** Change symbology for point, line, polygon. Apply color ramps and symbols. Save layer styles. (Based on "Symbology") **4. Working with Vector Attributes** Open attribute table. Sort, filter, calculate field values. Basic statistics. (Based on "Vector Attribute Data") **5. Labeling Vector Data** **Use labeling tool.** **Customize label appearance**. **6. Vector Data Classification** Classify using unique values and graduated styles. Explore classification methods (Natural Breaks, Quantile). (Based on "Classification") **7. Creating Print Layout** Insert map in layout. Add title, legend, scale bar, north arrow. (Based on "Using Print Layout") **8. Dynamic Print Layout and Export** Create multiple map views. Export as PDF and image 9. Thematic Mapping Create choropleth maps using attribute data **10. Creating New Vector Layers** Create point, line, and polygon layers. Add new features manually Spatial Operations **11. Feature Topology and Editing** Enable topology rules. Snap to feature **12. Vector Analysis** Reprojecting and Transforming data Vector Analysis Network Analysis

	Spatial Statistics
	Module 2: Raster Data and Analysis
	13. Loading and Styling Raster Data Add raster layers (e.g., DEM, satellite). Change raster symbology **14. Terrain Analysis** Slope, aspect, hillshade analysis **15. Raster Operations** Clip and merge rasters using processing tools **16. Raster to Vector Conversion** **17. Installing and Using Plugins** Install essential plugins (e.g., QuickOSM, Georeferencer). Use plugin feature 18. Create a base map, analyze the map and print the final map. 19. Forestry Application with GIS
10	**Text Books**
11	**Reference Books/links for online resources** 1. https://docs.qgis.org/3.40/en/docs/training_manual/index.html 2. https://libguides.brown.edu/gis_data_tutorials/intro_qgis
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 2Hr)** **Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Enterprise Network Practical

Sr.No.	Heading	Particulars
1	**Description the course :** **Including but Not limited to:**	This practical course focuses on enterprise-level networking skills, enabling students to configure, troubleshoot, and manage networks in real-world business environments. It aligns with emphasis on skill-based, multidisciplinary learning through lab work and experiential methods. The course totals 50 marks, with 2 credits dedicated to practical sessions. Develop proficiency in setting up LAN/WAN topologies, protocols, and enterprise hardware like routers and switches. Apply troubleshooting techniques for network issues, including security and performance optimization. Foster industry-relevant skills such as simulation using tools like Cisco Packet Tracer or GNS3
2	**Vertical :**	Elective
3	**Type :**	Practical
4	**Credits :**	2 credits (1 credit = 30 Hours of Practical work in a semester)
5	**Hours Allotted :**	60 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. Impart hands-on skills in configuring and managing enterprise network devices like switches and routers. CO2: Enable students to troubleshoot common network faults and optimize performance in simulated environments. CO3: Familiarize with enterprise protocols (e.g., OSPF, BGP) and security features for scalable networks	
8	**Course Outcomes (OC):** OC1. Apply networking fundamentals to solve enterprise problems. OC2. Identify and troubleshoot complex network issues systematically. OC3. Develop scalable LAN/WAN architectures with security protocols. OC4. Use simulators like Packet Tracer for enterprise configurations.	
9	**Modules:-**	
	Module 1:	
	1. Campus network architecture a. Static VLANS, Trunking, and VTP b. EtherChannel 2. Spanning Tree Impimentation	

	a. Implement Spanning Tree Protocols b. Multiple Spanning Tree 3. Inter VLAN and DHCP implementation a. Inter-VLAN Routing b. DHCP V4 and DHCP V6 4. Redundancy implementation a. First Hop Redundancy Protocols – HSRP and VRRP b. Hot Standby Router Protocol for IPV6 c. Gateway Load Balancing Protocol (GLBP) 5. Switching Features and Technologies a. IP Service Level Agreements and Remote SPAN in a Campus Environment b. Securing Layer 2 Switches c. Securing VLANs
	Module 2: 6. Basic Network and Routing Concepts a. Basic RIPng and Default Gateway Configuration 7. OSPF implementation a. OSPF Virtual Links b. Multi-Area OSPFv2 and OSPFv3 with Stub Area c. OSPFv3 Address Families 8. Manipulating Routing Updates a. Redistribution Between RIP and OSPF b. Redistribution Between RIPng for IPv6 and OSPFv3 c. Controlling Routing Updates 9. Path Control Implementation a. Configure and Verify Path Control Using PBR b. Configure IP SLA Tracking and Path Control 10. Enterprise Internet Connectivity a. Configure NAT Services 11. Implementation of BGP a. Configuring BGP with Default Routing b. Using the AS_PATH Attribute c. Configuring IBGP and EBGP Sessions, Local Preference, and MED d. IBGP, Next Hop and Synchronization
10	**Text Books**
11	**Reference Books** 4. CCNP Routing and Switching ROUTE 300-101 Official Cert Guide 5. CCNP Routing and Switching SWITCH 300-115 Official Cert Guide 6. CCNP and CCIE Enterprise Core ENCOR 350-401 (Official Cert Guide), 2nd Edition **7.** CCNP Enterprise Advanced Routing ENARSI 300-410 Official Cert Guide
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through:

	Students are expected to attend each practical and submit the written practical of the previous session. Performing Practical and writeup submission will be continuous internal evaluation. 2.5 marks can be awarded for each practical performance and writeup submission totaling to 50 marks and can be converted to 20 marks.
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration:2Hr)** **Certified copy of Journal is compulsory to appear for the practical examination** Practical Slip: Q1. From Module 1 13 marks Q2. From Module 2 12 Marks Q3. Journal and Viva 05 marks

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: Cyber Law and Digital Policy

Sr.No.	Heading	Particulars
1	**Description the course : Including but Not limited to:**	This course introduces students to the legal and policy framework governing cyberspace in India. It provides a comprehensive understanding of the evolution of cyber law, particularly under the Information Technology Act, 2000, and examines the regulatory mechanisms addressing digital rights, intermediary liability, cybercrimes, and electronic evidence. The course further explores emerging challenges in digital governance, including data protection under the Digital Personal Data Protection Act, 2023, privacy rights, domain name disputes, and Intellectual Property Rights (IPR) in the era of Artificial Intelligence. By integrating legal theory with practical case analysis, the course equips students with analytical skills to understand cyber frauds, digital forensics, AI-related IPR concerns, and contemporary digital policy developments in India.
2	**Vertical :**	Elective
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO1 To introduce the foundational components and statutory evolution of Cyber Law in India. CO2 To understand the legal complexities of online freedom and the liability of intermediaries. CO3 To analyse the mechanisms of cyber frauds and the evidentiary value of digital forensics. CO4 To evaluate the legal impact of AI on Intellectual Property Rights and Data Privacy.	

8	**Course Outcomes (OC):** OC1 Describe the legal history and current components of the IT Act. OC2 Assess the obligations of digital platforms under Intermediary Liability rules. OC3 Apply the provisions of the DPDP Act, 2023 to data protection challenges. OC4 Identify legal risks and protections related to AI-generated content and digital IPR.
9	**Modules:-**
	Module 1: Statutory Framework & Data Policy
	1.1 Legal Foundations: Cyber Law & Components of Cyber Law in India; Overview of the Information Technology Act, 2000. **1.2 Statutory Evolution:** 25 Years of the IT Act, 2000 (Milestones and Amendments). **1.3 Digital Rights & Liability:** Online Freedom of Speech and Expression; Liability of Intermediaries (Safe Harbour Framework). **1.4 Privacy Policy:** Privacy of Online Data and Information; Digital Personal Data Protection Act, 2023; Right to be Forgotten.
	Module 2: Cyber Crimes, Forensics & Digital IPR
	2.1 Offence Analysis: Cyber Crimes and Cyber Contraventions; Investigation of Cyber Frauds & Various Cyber Scams (Phishing, Identity Theft). **2.2 Digital Evidence:** E-evidence and Computer Forensics; Legal standards for admissibility of electronic records in India. **2.3 Digital IPR Framework:** IPR Issues in Digital Media (International and Indian Scenario); Copyright and Patent Issues in the Digital Medium. **2.4 Emerging Policy Issues:** Domain Name Dispute & Resolution; Trademark Issues in Digital Medium; IPR Issues in the Era of Artificial Intelligence (AI).
10	**Text Books** 11. **Cyber Laws, Information Technology & Artificial Intelligence** - Dr. Jyoti Rattan (11th Edition).
11	**Reference Books** 8. **Cyber Law of India** – Dr. S.R. Myneni 9. **Information Technology Law and Practice** – Vakul Sharma 10. **Cyber Laws** – Farooq Ahmad 11. **The Indian Cyber Law** - Suresh T. Viswanathan

12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through: 1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks **4.** Total: 20 marks
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 1 hr)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

Syllabus
B.Sc. (Information Technology)
(Sem.- VI)

Name of the Course: IT Service Management

Sr.No.	Heading	Particulars
1	**Description the course: Including but Not limited to:**	IT Service Management (ITSM) is a strategic approach to designing, delivering, managing, and improving the way IT services are provided to meet the needs of an organization and its customers, with a strong focus on aligning IT capabilities with business objectives. The course typically introduces key frameworks such as ITIL (IT Infrastructure Library), covering core processes like service strategy, service design, service transition, service operation, continual service improvement, and modern practices including ITIL 4's guiding principles, value streams, and the four dimensions of service management.
2	**Vertical :**	Elective
3	**Type :**	Theory
4	**Credits :**	2 credits (1 credit = 15 Hours for Theory work in a semester)
5	**Hours Allotted :**	30 Hours
6	**Marks Allotted:**	50 Marks
7	**Course Objectives(CO):** CO 1. Understand the fundamentals of IT service management and the ITIL 4 framework. CO 2. Explore the various management practices in ITIL, including general management, service management, and technical management. CO 3. Develop an understanding of driving stakeholder value and creating value for stakeholders. CO 4. Gain knowledge of creating, delivering, and supporting services within the ITIL service value system. CO 5. Explore the concept of High-Velocity IT and the importance of continual improvement.	
8	**Course Outcomes (OC):** **CO 1:** Apply the principles and concepts of ITIL 4 to enhance IT service management practices. **CO 2:** Demonstrate proficiency in implementing ITIL management practices. **CO 3:** Successfully drive stakeholder value and engage in value co-creation. **CO 4:** Develop the skills to create, deliver, and support services within the ITIL service value system. **CO 5:** Understand the principles and techniques of High-Velocity IT and foster a culture of continual improvement.	
9	**Modules:-**	
	Module 1:	

	Introduction: IT service management in the modern world, About ITIL 4, The structure and benefits of the ITIL 4 framework. **Key concepts of service managemen**t : Value and value co-creation, Organizations, service providers, service consumers, and other stakeholders, Products and services, Service relationships, Value: outcomes, costs, and risks. **The four dimensions of service management**: Organizations and people, Information and technology, Partners and suppliers, Value streams and processes, External factors. **The ITIL service value system :** Service value system overview, Opportunity, demand, and value, The ITIL guiding principles, Governance, Service value chain, Continual improvement. **ITIL management practices:** General management practices, Service management practices, Technical management practices.
	Module 2:
	Drive Stakeholder Value: Introduction, The customer journey, Step 1: Explore, Step 2: Engage, Step 3: Offer, Step 4: Agree, Step 5: Onboard, Step 6: Co-create, Step 7: Realize. **Create, Delivery and Support (CDS):** Service value system key concepts and challenges, Using a shift-left approach, Plan and manage resources in the service value system, The use and value of technology across the service value system. **Value streams for new services**: Reviewing service value chains and service value streams, ITIL practices and value streams for new services, Change enablement, Service design and software development and management, Service validation and testing, Release management and deployment management **Value streams for user support**: ITIL practices and value streams for user support, Service desk, Incident management, Problem management, Knowledge management, Service level management and Monitoring and event management. **How to create, deliver and support services** **High-Velocity IT:** Introduction, Key concepts, culture, techniques, Continual improvement.
10	**Text Books** 1. ITIL Foundation 4 Edition Roman Jouravlev, Akashay Anand, et. al TSO, AXELOS 2nd Edition 2019 2. ITIL 4 High-Velocity IT Akashay Anand, Dan Ashby, et. al TSO, AXELOS 1st Edition 2020 3. ITIL 4 Create, Delivery and Support (CDS) Claire Agutter AXELOS 1st Edition 2021
11	**Reference Books** 1. ITIL 4 Drive Stakeholder Value Roman Jouravlev, Pavel Demin, et. al, TSO, AXELOS 1st Edition 2020 2. ITIL 4 Direct, Plan and Improve, Akashay Anand, Mauricio Corona, et. al. TSO, AXELOS 1st Edition 2020
12	**Internal Continuous Assessment: 40%**
	Continuous Evaluation through:

	1. Class test of 1 of 15 marks 2. Class test of 2 of 15 marks Average of the two: 15 marks 3. Quizzes/ Presentations/ Assignments: 5 marks **4.** Total: 20 marks
13	**Semester End Examination: 60%**
	30 marks Semester End Examination
14	**Format of Question Paper:** (**Semester End Examination : 30 Marks. Duration: 1 hr)** Q1: Attempt any three (out of five/six) from Module 1 (15 marks) Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

QUESTION PAPER PATTERN
(External and Internal)
(Mentioned along with each course)

Format of Question Paper: (Semester End Examination : 30 Marks. Duration: 1 hr)

Q1: Attempt any three (out of five/six) from Module 1 (15 marks)

Q2: Attempt any three (out of five/six) from Module 2 (15 marks)

Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	% of Marks	Alpha-Sign/ Letter Grade Result	Grading Point
9.00 - 10.00	90.0 – 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	-	Ab (Absent)	0

Vertical-6

As Per NEP 2020

University of Mumbai



Syllabus for Community Engagement Project (CEP) Vertical-6	
Faulty of Science & Technology	
Board of Studies in Information Technology	
Name of the Programme – B.Sc. (Information Technology)	
Semester	**V**
Credit	**2**
Duration	**30 hrs (Field Work+ Survey) + 15hrs (Discussion + Report Writing) : Total - 45 hrs**
From the Academic Year	**2026-27**

Name of Faculty:- Science & Technology
Name of Programme :- B.Sc. (Information Technology)
Duration :- 30 hrs (Field Work+ Survey) + 15hrs (Discussion + Report Writing) : Total - 45 hrs

Indicative Topics for CEP

Sr. No.	Name of the Topic
1.	Digital Literacy Training for Senior Citizens
2.	Basic Computer Skills for Rural School Students
3.	Smartphone Usage & Online Safety for Women Self-Help Groups
4.	Digital Awareness Program for Persons with Disabilities
5.	Teaching Internet Basics to First-Generation Learners
6.	Bridging the Digital Divide in Urban Slum Communities
7.	Multilingual Digital Literacy Content Development
8.	Helping Small Vendors Go Digital (UPI, QR, Apps)
9.	Cyber Awareness Campaign for Rural Communities
10.	Digital Skills Bootcamp for Unemployed Youth
11.	Cybercrime Awareness and Prevention Workshops
12.	Social Media Safety & Digital Footprint Awareness
13.	Phishing, Scam & Fraud Detection Awareness Program
14.	Password Hygiene and Personal Data Protection Campaign
15.	Cyber Ethics and Responsible Internet Usage Program
16.	Online Safety for Children and Adolescents
17.	Digital Privacy Awareness for Community Organizations
18.	Secure Online Banking Awareness Programme
19.	Community Helpdesk for Cyber Safety Queries
20.	Fake News Detection and Media Literacy Programme
21.	Assisting Citizens in Using Government Digital Portals
22.	Awareness Program on Digital India Initiatives
23.	Helping Communities Access Online Healthcare Services
24.	Training on DigiLocker, Aadhaar & PAN Services
25.	Online Grievance Redressal Systems Awareness
26.	Digital Tax Filing Support for Small Businesses

27.	Voter Awareness Using Digital Platforms
28.	Digital Skill Support for Local Government Offices
29.	Smart Village / Smart City IT Enablement Project
30.	Creating Digital Guides for Public Welfare Schemes
31.	Developing E-Learning Content for Government Schools
32.	Setting Up Virtual Classrooms for Rural Schools
33.	Coding Awareness Program for School Children
34.	STEM Education Support Using Free Digital Tools
35.	Educational App Development for Local Language Learning
36.	Digital Library Creation for Community Centers
37.	Online Career Guidance Portal for Underprivileged Students
38.	Learning Management System (LMS) Setup for NGOs
39.	Assistive Learning Tools for Differently-Abled Students
40.	Digital Assessment Tools for Community Schools
41.	Health Awareness Campaign Using Mobile Applications
42.	Telemedicine Awareness and IT Support Program
43.	Environmental Monitoring Using Sensors & Data Analytics
44.	Waste Management Tracking System for Local Bodies
45.	Air & Water Quality Awareness Using Data Visualization
46.	IT-Based Disaster Preparedness and Alert Systems
47.	Nutrition & Health Tracking App for Community Use
48.	Digital Record System for Community Health Workers
49.	Smart Farming Advisory System for Farmers
50.	Climate Change Awareness Using Interactive Dashboards
51.	Website & Digital Presence Development for NGOs
52.	IT Support for Self-Help Groups and Micro-Entrepreneurs
53.	Digital Marketing Training for Small Businesses
54.	Inventory & Billing Software for Local Shops
55.	Data Management Solutions for Non-Profit Organizations
56.	Crowdfunding Platform Development for Social Causes
57.	Social Media Strategy for Community-Based Organizations
58.	Technology Support for Rural Startups
59.	Building Open-Source Solutions for Social Problems
60.	Community-Based Hackathon for Social Innovation

The topics are indicative and the faculty members should allot Community Engagement Project that are relevant and important as per core Subject. The Community Engagement Project may be taken individual or in a group up to 4 students with proper guidance from Faculty.

Evaluation Pattern:-

Evaluation during CEP Program involves two key components :-

External Evaluation 60%

Internal Evaluation 40%

Evaluation Chart

(i) External Evaluation (Marks 30)

Criteria	Marks
Objectives, Literature Review , Methodology, Data Analysis, Conclusion and Recommendations	15
Overall Project Report Structure and Style	5
Presentation Skills & Communication	10
Total	30

(ii) Internal Evaluation by Guide (Marks 20)

Criteria	Marks
Attendance, Community interactions completion and interaction with Supervisor	10
Overall Report quality	10
Total	20

*** Please see the Guidelines for Community Engagement Project for UG Students, as per NEP 2020**

Sd/-
Sign of the BOS
Dr. R. Srivaramangai
Chairman/Co-ordinator
BOS/Ad-hoc BOS in Information Technology

Sd/-
Sign of the
Offg. Associate Dean
Dr. Madhav R. Rajwade
Faculty of Science & Technology

Sd/-
Sign of the Offg. Dean
Prof. Shivram S. Garje
Faculty of Science & Technology