

ETERNAL UNIVERSITY

(ESTABLISHED UNDER HIMACHAL PRADESH GOVERNMENT ACT NO.3 OF 2009)

BARU SAHIB HIMACHAL PRADESH



WORLD PEACE THROUGH VALUE BASED EDUCATION

AKAL COLLEGE OF ENGINEERING & TECHNOLOGY

B.C.A. (HONS. WITH RESEARCH) (BATCH
2024 -28) CURRICULUM (SEMESTER III & IV)

APPROVED VIDE ANNEXURE 4.5.4 OF 87TH
ACADEMIC COUNCIL MEETING HELD ON
25TH JULY, 2025

TO BE IMPLEMENTED FROM THE ACADEMIC
SESSION 2025-26

TSB Singh

Dean
Academic Affairs
Eternal University
Baru Sahib (H.P.) 173101

Simran Singh

Registrar (Officiating)
Eternal University
Baru Sahib (H.P.) 173101

ETERNAL UNIVERSITY

(Established Under Himachal Pradesh State Act No.3 of 2009)

STUDY SCHEME AND SYLLABUS

FOR

4 Year – B.C.A. (Hons. with Research)

2024-2028



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

AKAL COLLEGE OF ENGINEERING & TECHNOLOGY

ETERNAL UNIVERSITY

BARU SAHIB, SIRMAUR, H.P.

To be implemented from Academic Session 2024-2025

ABOUT THE BCA

Computer Application has been evolving as an important branch of science and technology in the last few years and it has carved out a space for itself like computer science and engineering. Computer application spans theory and more application and it requires thinking both in abstract terms and in concrete terms. Computer science has a wide range of specialties. These include computer architecture, software systems, Graphics, Artificial Intelligence, Mathematical and Statistical Analysis, Data science, Computational Science and Software Engineering.

SALIENT FEATURES OF THE DEPARTMENT

- Research oriented curriculum designed to enable students to acquire all the skills needed to collect and analyze the data.
- The Institute draws upon its strength of highly qualified well-trained faculty, state of art infrastructure and innovative teaching methodology.
- Elective courses that bridge the gap between industry requirements and academia.
- Hands-on experience in most of the courses of computer applications so as to impart practical knowledge in the relevant field.
- To keep the students at par with the emerging technologies prevailing in the market, the institute is furnished with various specialized research labs and software labs.

BCA (BACHELOR OF COMPUTER APPLICATION)

The Program outcomes in BCA are aimed at allowing flexibility and innovation in design and development of course content, in method of imparting training, in teaching learning process and in assessment procedures of the learning outcomes. The emphasis in BCA courses, in the outcome-based curriculum framework, help students learn how to solve problems, accomplish IT tasks, and express creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages. After graduating with a 4 years degree, the students are eligible for 1 year MCA (Master in Computer Application) Programme.

VISION

To prepare technically proficient and skillful computer professionals thereby contributing towards building a strong and developed nation.

MISSION

To provide innovative and quality knowledge to students for global competence and excellence. Also, to prepare high quality Professionals for catering the needs of industry.

ELIGIBILITY CRITERIA

10+2 or its equivalent examination in any stream conducted by a recognized Board/ University/Council

DURATION

4 Years

PROGRAM OUTCOMES

PO1: Core Computing Competency and Analytical Skills: Graduates will build a strong foundation in computer science and programming, enabling them to analyze complex problems, determine computational requirements, and design reliable software solutions.

PO2: Applied Project Development Skills: Graduates will gain hands-on experience to design, develop, and deploy innovative, scalable software systems that address real-world challenges.

PO3: Research Mindset and Technological Innovation: Graduates will demonstrate curiosity and initiative in exploring and applying emerging technologies, advancing research, and fostering creativity.

PO4: Professional Communication and Collaborative Leadership: Graduates will excel in effective communication, teamwork, and leadership in multidisciplinary project environments.

PO5: Adaptability and Continuous Learning: Graduates will embrace lifelong learning, adapting proactively to evolving technologies and methodologies for ongoing self-improvement.

PO6: Ethical, Social, and Environmental Responsibility: Graduates will act with integrity and responsibility, considering the ethical, societal, and environmental impacts of their technology decisions.

PO7: Entrepreneurial and Industry Readiness: Graduates will possess technical expertise, business acumen, and project management skills, preparing them for successful careers or entrepreneurship in the IT sector.

PROGRAMME SPECIFIC OUTCOMES (PSO)

PSO1: Gain proficiency in programming languages (C, Java, PHP), databases, software engineering, networking, and theoretical concepts, and apply them practically to develop software applications and manage projects using ICT tools.

PSO2: Build strong analytical, logical thinking, and problem-solving abilities to design efficient computer-based solutions for real-world challenges.

PSO3: Develop research skills and utilize emerging technologies to innovate, contribute new knowledge in computer science, and prepare for research publication.

PSO4: Acquire industry-relevant skills, entrepreneurial abilities, and awareness of ethical and social responsibilities to effectively pursue professional careers.

PSO5: Communicate scientific and technical information clearly, work effectively both individually and in teams, and engage in lifelong learning to stay updated in the fast-evolving IT industry.

Semester 1						
Nature of the Course	Course Code	Course Title	L	T	P	C(C= L+T+(p/2))
DSC	BCA 101	Foundations of IT	3	0	2	4
DSC	BCA 102	Mathematics for Computer Science	3	1	0	4
DSC	BCA 103	Database Management System	3	1	2	5
DSE	BCA 104	Problem Solving and Programming Fundamentals 1	3	1	2	5
AEC	BCA 105	Environmental Studies for Technology	2	0	0	2
VAC	BCA 106	New Age Life Skills 1	1	0	2	2
		Total	15	3	8	22

Semester 2						
Nature of the Course	Course Code	Course Title	L	T	P	C(C= L+T+(p/2))
DSC	BCA 107	Problem Solving and Programming Fundamentals 2	3	1	2	5
DSC	BCA 108	Agile Software Development Life Cycle	3	1	0	4
DSC	BCA 109	Web Development Frontend 1-HTML-CSS	2	1	4	5
VAC	BCA 110	New Age Life Skills 2	1	1	2	3
SEC	BCA 111	Introduction to Generative AI	4	1	0	5
		Total	13	5	8	22

Semester 3						
Nature of the Course	Course Code	Course Title	L	T	P	C(C= L+T+(p/2))
DSE	BCA 201	Web Development Frontend 2 -JS React	2	1	4	5
DSC	BCA 202	Operating System Fundamentals	2	1	2	4
DSC	BCA 203	Computer Network	2	1	0	3
DSE	BCA 204	Data Structures and Algorithms (Problem Solving 3)	2	1	4	5
AEC	BCA 205	New Age Life Skills 3	1	0	2	2
VAC	BCA 206	Introduction to ChatGPT	2	1	0	3
		Total	11	5	12	22

Semester 4						
Nature of the Course	Course Code	Course Title	L	T	P	C(C= L+T+(P/2))
DSC	BCA 207	Web Development Backend 1	3	0	4	5
DSC	BCA 208	Web Development Backend (Deployment, Security, and Scalability of Web Apps)	3	1	4	6
DSE	BCA 209	System Design (Low level)	3	1	0	4
AEC	BCA 210	New Age Life Skills 4	1	0	2	2
SEC	BCA 211	Introduction to Cloud Computing	3	1	2	5
		Total	13	3	12	22

Semester 5						
Nature of the Course	Course Code	Course Title	L	T	P	C(C= L+T+(p/2))
DSC	BCA 301	Artificial Intelligence Fundamentals (Self-Paced)	0	0	6	3
DSC	BCA 302	Introduction to Competitive Programming	0	0	6	3
DSE	BCA 303	Introduction to Prompt Engineering (Self-Paced)	0	0	6	3
IAPC-1	PROJ 301	Major Project/ Internship	0	0	26	13
		Total	0	0	44	22

Semester 6						
Nature of the Course	Course Code	Course Title	L	T	P	C(C= L+T+(p/2))
DSE	BCA 304	Python for Data Science & AI	3	0	2	4
VAC	GP 301	NCC/NSS and Evaluation of Overall Extracurricular Activities	0	0	4	2
IAPC-2	PROJ 302	Major Project/internship	0	0	32	16
		Total	3	0	38	22

SEMESTER-7

S.No.	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC	BCA 401	Advanced Machine Learning	3	0	2	4
2	DSE	BCA 402/BCA 403/BCA 404	Advanced R/ Computer Vision/ Quantitative and Statistical Concepts	3	0	2	4
3	DSE	BCA 405/BCA 406/BCA 407	Research Methodology/ML Ops/DevOps	3	0	2	4
4	DSE	BCA 408/BCA 409/BCA 410	Applied Cryptography/ Introduction to Cloud Computing/ Advanced Generative AI	3	0	2	4
5	IAPC-3	BCA 411	Dissertation on Major Or Dissertation on Minor Or Academic Project/Entrepreneurship	0	0	12	6
Total Credits				12	0	20	22

SEMESTER-8

S.No .	Course Type	Course Code	Course Title	Hrs./Week			Credits
				L	T	P	
1	DSC	BCA 412	Time Series Analysis and Forecasting	3	0	2	4
2	DSE	BCA 413/BCA 414/BCA 415	Pattern Recognition/Deep Learning/Network Simulation	3	0	2	4
3	DSE	BCA 416/BCA 417/BCA 418	Entrepreneurship Practices/ Mobile Computing/Distributed Computing Systems	3	1	0	4
4	DSE	BCA 419/BCA 420/BCA 421	Natural Language Processing/ Artificial Neural Networks/Soft Computing	3	0	2	4
5	IAPC-4	BCA 422	Dissertation on Major Or Dissertation on Minor Or Academic Project/Entrepreneurship	0	0	12	6
Total Credits				12	1	18	22

OVERALL CREDIT DISTRIBUTION TABLE

SEMESTER	Hrs. / Week			Credits
	L	T	P	
SEMESTER-I	15	3	8	22
SEMESTER-II	13	5	8	22
SEMESTER-III	11	5	12	22
SEMESTER-IV	13	3	12	22
SEMESTER-V	0	0	44	22
SEMESTER-VI	3	0	38	22
SEMESTER-VII	12	0	20	22
SEMESTER-VIII	12	1	18	22

Note – L: Lecture Hour, T: Tutorial Hour, P: Practical Hour, TC: Total Credits, IAE: Internal Assessment Examination, ESE: End Semester Examination.

Definition of Credit

1 Hr. Lecture (L) per week	1 Credit
1 Hr. Tutorial (T) per week	1 Credit
1 Hr. Practical (P) per week	0.5 Credit
2 Hrs. Practical (P) per week	1 Credit

Semester 1						
Nature of the Course	Course Code	Course Title	L	T	P	C(C= L+T+(p/2))
DSC	BCA 101	Foundations of IT	3	0	2	4
DSC	BCA 102	Mathematics for Computer Science	3	1	0	4
DSC	BCA 103	Database Management System	3	1	2	5
DSE	BCA 104	Problem Solving and Programming Fundamentals 1	3	1	2	5
AEC	BCA 105	Environmental Studies for Technology	2	0	0	2
VAC	BCA 106	New Age Life Skills 1	1	0	2	2
		Total	15	3	8	22

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 101
Course Title	Foundations of IT
Semester	I
Type of Course	DSC
Credits	3+0+1
Course Prerequisites	–
Course Objective(s)	1. Understand the basics of computer fundamentals. 2. Familiarize with the working principles of a computer 3. Familiar with the various devices and interfaces 4. Compare and contrast different computer components and their functionalities. 5. Evaluate the implications of technology choices on user experience and security.
Course Outcome (CO)	CO1: Describe key computer components and differentiate between hardware and software. CO2: Explain how data is processed within a computer and the role of the operating system. CO3: Identify common computer peripherals and input/output interfaces. CO4: Assess the importance of ergonomics and security considerations in technology usage protocols. CO5: Demonstrate basic skills and perform routine maintenance tasks to ensure optimal computer performance.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	1	1	1	1	2	1	3	2	1	1	1
CO2	3	2	1	1	2	2	1	3	3	2	2	1
CO3	3	2	1	1	1	2	1	3	2	1	1	1
CO4	3	2	1	1	2	2	1	3	3	1	1	1
CO5	2	1	1	2	3	3	1	2	2	1	2	2
Avg.	2.8	1.6	1.0	1.2	1.8	2.2	1.0	2.8	2.4	1.2	1.4	1.2
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Computers: Computer Definition, Purpose of Computers, Evolution of Computing Machines. Explore the History of Computers, Early Mechanical Computers, Emergence of Digital Computers, Pioneering Computer Innovations, Modern Computing Era, Discuss the Various Uses of Computers in Modern Society, Communication and Networking.	11
UNIT II	Differentiate Between Computer Hardware and Software, Hardware vs. Software, Examples of Computer Hardware, Types of Software (System and Application). Explore the Key Components of a Computer System, (CPU), Memory (RAM and ROM), (HDD, SSD), Motherboard, Input Devices (Keyboard, Mouse, Touchscreen), Output Devices (Monitor, Printer), Storage Devices (USB Drives, Optical Discs),	11
UNIT III	How computers process information: Binary Representation, Data Encoding, Character Encoding (ASCII, Unicode), Data Types and Formats. Number System, Binary Numbers, and Conversions, Number Systems Overview (Decimal, Binary, Hexadecimal), Binary Arithmetic (Addition, Subtraction), Binary to Decimal Conversion, Decimal to Binary Conversion, Hexadecimal Representation. Logic Gates, Logic Gates Overview, AND, OR, NOT Gates, NAND, NOR, XOR, XNOR Gates, Logic Gate Combinations. Culmination.	12
UNIT IV	Computer Networks and Internet: Definition of Computer Networks, Networking Purpose and Importance, Network Protocols (TCP/IP, HTTP, FTP). Discuss Web Browsers and Online Services, Web Browsers and Their Role, Online Services	11

	(Email, Social Media, Cloud), Internet Safety and Security, Browsing and Search Engines.	
	Practical	30
	Total Theory + Practical	75
LIST OF PRACTICALS 1. Dismantling and assembling a computer. 2. Identifying motherboard, CPU, RAM, HDD, SSD. 3. Installing operating systems (e.g., Windows, Linux). 4. Creating and formatting partitions. 5. Installing and uninstalling software. 6. Configuring BIOS/UEFI settings. 7. Understanding input/output devices. 8. Basic network setup using LAN. 9. Constructing and analyzing basic logic gate circuits. 10. Binary to decimal and vice versa conversions. 11. Setting up printers and scanners. 12. Practicing safe shutdown and reboot procedures. 13. Exploring file systems and file extensions. 14. Introduction to MS-DOS and basic commands. 15. Troubleshooting common hardware issues.		

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	Yes
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Fundamentals of Computers	V. Rajaraman
2	Computer Fundamentals	P.K. Sinha & Priti Sinha
3	Introduction to Information Technology	ITL Education Solutions
4	Essentials of Computer Science	E. Balagurusamy

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 102
Course Title	Mathematics for Computer Science
Semester	I
Type of Course	DSC
Credits	3+1+0
Course Prerequisites	–
Course Objective(s)	1. Develop proficiency in fundamental mathematical concepts 2. Apply mathematical techniques to solve real-world problems. 3. Enhance critical thinking and problem-solving skills.
Course Outcome (CO)	CO1: Effectively work with algebraic expressions and equations to address real-world challenges. CO2: Demonstrate precision in long division for polynomials and numbers, simplifying complex mathematical problems. CO3: Proficiently solve linear equations and systems, applying these skills in practical scenarios. CO4: Efficiently find HCF, LCM, and work with exponents and logarithms, applying them to various mathematical applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	1	2	2	1	1	3	3	1	2	2
CO2	3	2	1	1	2	1	1	3	3	1	1	2
CO3	3	3	1	2	2	1	2	3	3	1	2	2
CO4	3	2	1	1	2	1	1	3	2	1	1	1
Avg.	3.0	2.1	1.3	1.3	2.0	1.3	1.6	3.0	2.9	1.3	1.7	1.7
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Arithmetic : Fundamentals of arithmetic, beginning with exploration of basic operations, encompassing addition, subtraction, multiplication, and division, importance of	11

	understanding order of operations, denoted by PEMDAS, applying order of operations to solve arithmetic problems, practical problem-solving in arithmetic, real-world applications, word problems, and multi-step problem-solving.	
UNIT II	Understanding Algebra, Algebraic Expressions, Variables and Constants, Terms and Coefficients. Solving Simple Equations, Linear Equations, Solving Equations with One Variable, Equations and Solutions, Using Variables and Constants, The Role of Variables, Constants in Algebra, Expressing Relationships, Algebraic Notation, Expressions and Formulas, Mathematical Expressions, Evaluating Expressions, Algebraic Formulas, Introduction to Polynomials, Adding and Subtracting Polynomials, Multiplying and Factoring Polynomials, Quadratic Equations. Matrix operations - Addition and subtraction, Scalar multiplication, Multiplication, Transpose Matrix properties - Square matrices, Symmetric and skew-symmetric matrices, Diagonal matrices, Identity matrix, Zero matrix, Determinant, Inverse, Linear equations Types of matrices - Square matrices, Symmetric and skew-symmetric matrices, Diagonal matrices, Identity matrix, Zero matrix	14
UNIT III	Exponents: Understanding Exponents, Exponent Terminology, Writing Exponents, Examples of Exponent Notation. Laws of Exponents, Multiplying Exponents, Dividing Exponents, Raising Exponents to Exponents, Negative Exponents, Exponent Properties with Parentheses. Scientific Notation, Introduction to Scientific Notation, Converting to Scientific Notation, Exponent Properties with Parentheses, Exponents and powers, Intro to rational exponents, Unit-fraction exponents, Fractional exponents, Evaluating fractional exponents.	10
UNIT IV	Logarithms : Understanding Logarithms, Logarithmic Terminology, Writing Logarithms, Examples of Logarithmic Notation, Basic Logarithmic Properties, Logarithms of Different Bases, Logarithmic Rules, Equations with Logarithmic Terms, Isolating and Solving for Logarithms, Logarithmic Equations with Exponents, Logarithms in Real-World Scenarios.	10
	Tutorial	15
	Total Theory + Tutorial	60

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorial	15
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	60

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Discrete Mathematics and Its Applications	Kenneth H. Rosen
2	Mathematics for Computer Science	Eric Lehman, F. Thomson Leighton (MIT OpenCourseWare)
3	Higher Engineering Mathematics	B.S. Grewal

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 103
Course Title	Database Management System
Semester	I
Type of Course	DSC
Credits	3+1+1
Course Prerequisites	–
Course Objective(s)	1. Learning higher-level programming concepts 2. Applying different programming algorithms for various problem statements. 3. Write optimized, efficient, and quality code
Course Outcome (CO)	CO1: You will gain a deeper and practical real time understanding of SQL and CO2: MySQL as you create and optimize database schemas, CO3: Project provides valuable hands-on experience in designing, developing, and implementing database-driven applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	2	1	2	2	2	3	2	2	2	1
CO2	3	3	2	1	2	2	3	3	2	2	2	1
CO3	3	3	3	2	2	2	3	3	3	2	3	2
Avg.	3.0	2.7	2.3	1.3	2.0	2.0	2.7	3.0	2.3	2.0	2.3	1.3
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction and Basics of DBMS: Introduction of DBMS (Database Management System) , History of DBMS, Advantages of Database Management System, Disadvantages of DBMS, Application and Need for DBMS, DBMS Architecture, 1-level, 2-Level, 3-Level, Difference between File System and DBMS, Entity Relationship Model	11

UNIT II	SQL PART 1 (Structured Query Language): Intro, RDBMS, SQL, SELECT, WHERE, AND, OR, NOT, ORDER BY, INSERT INTO, NULL Values, UPDATE, DELETE, LIMIT, MIN and MAX, COUNT, AVG, SUM, LIKE, Wildcards, IN, BETWEEN. Aliases, Joins, INNER JOIN, LEFT JOIN, RIGHT JOIN, CROSS JOIN, Self Join.	11
UNIT III	SQL PART 2 (Structured Query Language): UNION, GROUP BY, HAVING, EXISTS, ANY, ALL, INSERT SELECT, CASE, Null Functions, Database, Create DB, Drop DB, Create Table, Drop Table, Alter Table, Constraints, Not Null, Unique, Primary Key, Foreign Key, Check, Default, Create Index, Auto Increment, Dates, Views.	12
UNIT IV	Relational algebra: introduction, Selection and projection, set operations, renaming, Joins, Division, syntax, semantics. Operators, grouping and ungrouping, relational comparison. Transaction management: ACID properties, Normalization (1NF, 2NF, 3NF, BCNF),	11
	Tutorial	15
	Practical	30
	Total Theory + Tutorial + Practical	90

LIST OF PRACTICALS

1. Create a database CollegeDB and a table Students with fields: student_id, name, age, course, marks, and city.
2. Insert at least 5 records into the Students table.
3. Update a student's marks and city using UPDATE with WHERE clause.
4. Delete a student record based on their student_id.
5. Query all students using SELECT with WHERE, AND, and OR conditions.
6. Create a Courses table with fields: course_id, course_name, and establish a foreign key relationship with the Students table.
7. Perform INNER JOIN and LEFT JOIN between Students and Courses to display student names with course names.
8. Use GROUP BY and HAVING to find average marks per course and filter groups with average > 70.
9. Write a nested subquery to list students whose marks are above the overall class average.
10. Create a view TopStudents to show students with marks above 75 from a specific city.
11. Apply PRIMARY KEY and FOREIGN KEY constraints in both tables appropriately.
12. Create new database users and assign privileges (one with read-only, one with full access).
13. Write a stored procedure InsertStudent to insert new records into the Students table.
14. Perform backup and restore of the CollegeDB database using .sql dump.

15. Use LIKE, IN, BETWEEN, and aggregate functions (AVG, COUNT, SUM) to perform data filtering and analysis.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorial	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	90

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Books	Author
1	Database System Concepts	Abraham Silberschatz, Henry F. Korth, S. Sudarshan
2	Learning SQL	Alan Beaulieu
3	MySQL Cookbook	Paul DuBois

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 104
Course Title	Problem Solving and Programming Fundamentals 1
Semester	I
Type of Course	DSE
Credits	3+1+1
Course Prerequisites	–
Course Objective(s)	1. Understand programming concepts. 2. Develop algorithmic thinking. 3. Cultivate problem-solving skills. 4. Learn debugging and testing techniques. 5. To apply concepts and techniques for implementation
Course Outcome (CO)	CO1: Understanding of algorithms and problem-solving with Language agnostic or block coding. CO2: Proficiency in working with variables in a block coding environment. CO3: A solid foundation in problem-solving CO4: Learn to write optimal codes with respect to time and space CO5: Implement sorting and searching algorithms and compare their runtime complexity

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	2	2	2	2	2	3	3	2	2	2
CO2	3	2	1	2	2	1	2	3	3	1	2	2
CO3	3	2	1	2	3	2	2	3	3	2	2	2
CO4	3	2	2	1	3	2	2	3	3	2	2	1
CO5	3	3	2	1	2	2	2	3	3	2	2	1
Avg.	3.0	2.2	1.6	1.6	2.4	1.8	2.0	3.0	3.0	1.8	2.0	1.6
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Programming : variables and operators (+, -, *, /, %, !,), Logical Operators, and (&&), or (), Condition Operator, Looping, Flowcharts, Flowchart Definition, Symbols and Shapes in Flowcharts, How to Create a Flowchart, Use Cases of Flowcharts, Advantages of Flowcharts.	11
UNIT II	Programming with block coding : Methods and functions, Writing Pseudocode for Algorithms, Pseudocode Conventions and Syntax, Transition from Pseudocode to Code, Block Coding, Block Coding Overview, Visual Programming with Blocks, Number System and Practice Session,	12
UNIT III	Algorithms, Definition, Importance of Algorithms, Designing and Implementing Algorithms, Algorithm Efficiency and Analysis, Real-World Applications of Algorithms, Pseudocode, Pseudocode Definition, Purpose of Pseudocode.	11
UNIT IV	Implement Data Structure Library - Function - What are functions? Parameters, Return Values and type, call by value, call by reference, and Arrays. What is 1d Array, 2d Array? Memory Management of Array. Introduction to String - Usages, Memory Management of Strings, Operations on String.	11
	Tutorial	15
	Practical	30
	Total Theory +Tutorial + Practical	90

LIST OF PRACTICALS

1. Design a flowchart to perform basic arithmetic operations (addition, subtraction, multiplication, division) between two numbers.
2. Write pseudocode for a calculator that performs arithmetic operations based on user input.
3. Implement a command-line calculator that takes two numbers and an operator as input and prints the result.
4. Create a BMI calculator that accepts height and weight as input and displays BMI and health category.
5. Develop a number guessing game where the program randomly picks a number and the user tries to guess it using hints.
6. Write a program to generate a multiplication table for a number entered by the user.
7. Create a program to find the maximum and minimum value from a list of user-entered numbers.
8. Implement linear search to find whether a given number exists in a list or not.

9. Write a bubble sort program to sort numbers in ascending order and count the number of comparisons made.
10. Demonstrate use of 1D and 2D arrays through sample programs like matrix addition or student score lists.
11. Create a simple login simulation that accepts a username and password and checks for correctness using conditionals.
12. Write and use custom functions for tasks like checking if a number is prime or calculating factorial.
13. Perform string manipulations such as reversing a string, checking for a palindrome, and counting vowels.
14. Translate a basic pseudocode algorithm into block coding using tools like Scratch or Blockly.
15. Write a program to convert a decimal number into binary using loops or recursion.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorial	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	90

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	

Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Python Programming for the Absolute Beginner	Michael Dawson
2	Think Like a Programmer	V. Anton Spraul
3	Coding for Beginners in Easy Steps	Mike McGrat

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 105
Course Title	Environmental Studies for Technology
Semester	I
Type of Course	AEC
Credits	2+0+0
Course Prerequisites	–
Course Objective(s)	1. Implement sustainable computing practices for reduced environmental impact. 2. Utilize data-driven approaches to support biodiversity conservation and ecosystem management. 3. Promote clean energy solutions and recycling practices for sustainable resource management. 4. Harness technology to address environmental challenges and promote awareness. 5. Foster interdisciplinary collaboration to integrate environmental considerations into various fields and promote responsible decision-making.
Course Outcome (CO)	CO1: Understanding Environmental Impacts of Technology CO2: Sustainable Technology Design CO3: Policy and Regulation Knowledge CO4: Resource Management and Conservation CO5: Environmental Ethics and Social Responsibility

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	2	2	3	1	2	3	2	2	2	3	2	2
CO2	2	2	3	1	2	3	3	2	2	3	3	2
CO3	1	1	2	1	2	3	3	1	1	3	3	1
CO4	2	1	2	1	2	3	2	1	1	2	2	1
CO5	1	1	2	2	2	3	2	1	1	2	3	2
Avg.	1.6	1.4	2.4	1.2	2.0	3.0	2.4	1.4	1.4	2.6	2.6	1.6

1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Green Computing: Energy-efficient coding practices, Sustainable data centers, Green software development tools, Cloud computing for sustainability, Virtualization techniques for energy savings, Optimization of server cooling systems, Power management software for devices, Renewable energy integration in computing facilities.	7
UNIT II	Data for Nature: Environmental sensor networks, Geospatial analysis software, AI and machine learning in environmental monitoring, Open data initiatives, Remote sensing technologies, Wildlife tracking and conservation databases, Climate modeling and prediction algorithms, Community science platforms for data collection.	7
UNIT III	Clean Energy and Recycling: Software solutions for renewable energy management, Electronic waste tracking systems, Circular economy platforms, Energy-efficient algorithms, Smart grid technologies for energy distribution, Waste-to-energy conversion software, Eco-friendly product design tools, Life Cycle assessment software for products and materials.	8
UNIT IV	Tech for a Greener Future: Blockchain for sustainable supply chains, Virtual reality for environmental education, Cybersecurity for environmental protection, Sustainable software startups, Smart home automation for energy efficiency, Environmental impact assessment tools, Sustainable transportation planning software, Eco-friendly packaging design software.	8
	Total Theory	30

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	30
Practical	
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	

Revision	
Others If any:	
Total Number of Contact Hours	30

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Environmental Studies	Benny Joseph
2	Textbook of Environmental Studies for Undergraduate Courses	Erach Bharucha
3	Green Computing Tools and Techniques	Bud E. Smith

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 106
Course Title	New Age Life Skills 1
Semester	I
Type of Course	VAC
Credits	1+0+1
Course Prerequisites	–
Course Objective(s)	1. Identify common communication problems that may be holding learners back 2. Perceive what the non-verbal messages are communicating to others 3. Understand the role of communication in the teaching-learning process 4. Learn to communicate through the digital media 5. Understand the importance of empathetic listening 6. Explore communication beyond language
Course Outcome (CO)	CO1: Get a clear understanding of good communication skills. CO2: Know what they can do to improve their communication skills. CO3: Demonstrate the effectiveness of digital marketing for business and using the tools to reach a global audience. CO4: Use nonverbal communication effectively in communication as an aid.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	1	1	1	3	2	1	2	1	1	1	1	3
CO2	1	1	1	3	3	1	2	1	2	1	1	3
CO3	1	3	2	2	2	2	3	3	2	2	3	2
CO4	1	1	1	3	2	1	1	1	1	1	1	3
Avg.	1	1.5	1.25	2.75	2.25	1.25	2	1.5	1.5	1.25	1.5	2.75
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Building Vocabulary: vowel words, words with silent alphabets, Sorting out words, identifying themes/categories, synonyms and antonyms, prefixes and suffixes, multiple-meaning words, context clues, building spellings, idioms, Foundation of Reading: Would this happen in real life, what is the picture about, analysis and inferences (identify the word for the pictures, what will happen next, what/who am I, use actions/words to describe the character), story-reading, describing the characters, beginner level comprehension reading and practice, introduction to informational text.	3
UNIT II	Grammar: Punctuation, Sentence structure, types of sentences, parts of speech: noun, pronoun, verb, adjective, and adverb, articles, prepositions, determinants, conjunctions, subject-verb agreement, contractions, editing & omitting. Listening/Watching: English songs, movies, and web series. Speaking: Greeting each other, Asking basic questions about others, Giving or asking for directions, Individual introduction (Where am I from, my family, relatives, educational background), Making requests for individual needs and necessities.	4
UNIT III	Goal setting, how to dream: Understanding the Importance of Goals, Setting SMART Goals, Creating a Goal Action Plan, Overcoming Challenges and Obstacles, Periodic Review and Adjustments, Unlocking Imagination, Visualization Techniques, Turning Dreams into Goals, Staying Motivated, Sharing Dreams and Goals. Cyber security: Introduction to Cybersecurity, Data Privacy Basics, Common Cyber Threats, Protecting Personal Devices, Online Behavior and Social Media Safety. Group discussion skills: Active Listening, Effective Communication, Respecting Diverse Perspectives, Conflict Resolution, Leadership in Group Discussions	4
UNIT IV	Office Tools: Types of OS, Files and Directories, Windows Operating Environment, Control Panel, Taskbar, Desktop Icons. Working with MS word -Opening & Saving files, Editing text documents, Inserting, Deleting, Cut, Copy, Paste, Undo, Redo, Find, Search, Replace, Formatting page & setting Margins, Converting files to different formats, Importing & Exporting documents, Sending files to others, Using Toolbars, Ruler, Using Icons, using help, Formatting Documents - Setting Font styles, Font selection- style, size, color etc, Typeface - Bold, Italic, Underline, Case settings, Highlighting,	4

	Special symbols, Setting Paragraph style, Alignments, Indents, Line Space, Margins, Bullets & Numbering.	
	Practical	30
	Total Theory + Practical	45

LIST OF PRACTICALS

1. Play a vocabulary game where you group words and find their synonyms or antonyms.
2. Join a spelling bee that includes tricky words with silent letters and homophones.
3. Read a short story and then retell it in your own words, focusing on characters and what happened.
4. Make your own mini-dictionary with new idioms and words that have more than one meaning.
5. Do a roleplay where you practice giving and asking for directions clearly.
6. Prepare and speak about yourself in a short self-introduction activity.
7. Practice gestures and body language in a mirror activity to understand non-verbal communication.
8. Watch a short English video and discuss what it means with your classmates.
9. Take part in a mock interview with a friend and give each other feedback.
10. Write and send a formal email to a teacher or imaginary employer using correct format.
11. Open a document in Google Docs or LibreOffice Writer, type something, save it, and format the text (bold, italic, headings).
12. Insert a picture or table in your Google Doc or LibreOffice file and format it neatly.
13. Try an editing challenge: use undo/redo, find and replace, and apply different text styles.
14. Share a Google Doc with a classmate or use LibreOffice on a shared computer and work on it together.
15. Work in a group to create a flyer or newsletter in Google Docs or LibreOffice for an upcoming event or awareness campaign.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	

Total Number of Contact Hours	45
-------------------------------	----

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Soft Skills: Enhancing Employability	M. S. Rao
2	Personality Development and Soft Skills	Barun K. Mitra
3	The 7 Habits of Highly Effective People	Stephen R. Covey

SEMESTER-2

Course Type	Course Code	Course Title
DSC	BCA 107	Problem Solving and Programming Fundamentals 2
DSC	BCA 108	Agile Software Development Life Cycle
DSC	BCA 109	Web Development Frontend 1- HTML-CSS
VAC	BCA 110	New Age Life Skills 2
SEC	BCA 111	Introduction to Generative AI

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 107
Course Title	Problem Solving and Programming Fundamentals 2
Semester	II
Type of Course	DSC
Credits	3+1+1
Course Prerequisites	–
Course Objective(s)	1. Learning Higher level programming concepts 2. Applying different programming algorithmics for various problem statements. 3. Write optimized, efficient, and quality code
Course Outcome (CO)	CO1: Apply programming paradigms like recursion, Greedy algo CO2: Implement Data Structure Library of Map and OOPS concepts CO3: Analyze the time and space complexity of algorithms to evaluate their efficiency in solving computational problems. CO4: Design and implement advanced data structures like trees, graphs, and heaps for optimized problem-solving in real-world applications.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	2	1	2	1	1	3	3	2	2	2
CO2	3	3	2	1	2	1	1	3	3	2	2	2
CO3	3	2	2	1	2	1	1	3	3	2	2	2
CO4	3	3	2	2	2	1	2	3	3	2	2	2
Avg.	3	2.5	2	1.25	2	1	1.25	3	3	2	2	2
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Dynamic Data Structure: ArrayList, Operations on ArrayList: Add: Insert an element at the end of the list, Remove: Remove an element by its index or the element itself, Access: Get the element at a specific index (position) in the list. Problem-Solving Techniques: Recursion - Definition, Base Case, Recursive Case, Examples. Backtracking - Definition, Example: Finding all paths through a maze. Applying Problem-Solving Techniques: Recursion in Arrays, Get Subsequence, Target Sum Subsets.	12
UNIT II	Basics of time and space complexity: Asymptotic notation (big O, big omega, big theta), Common algorithms and data structures (e.g., sorting, searching), Implement sorting and searching algorithms and compare their runtime complexity and algorithm analysis (e.g., time complexity, space complexity).	11
UNIT III	Method for handling data structures: LIFO (Last In, First Out) and FIFO (First In, First Out). Principle analogy of a stack, Principle analogy of a queue. Intermediate level data structures: Linked Lists and their types: Singly linked lists, Doubly linked lists, Circular linked lists; Generic Tree, Binary Tree, Binary Search Tree.	11
UNIT IV	Object-oriented programming system and Key Value Pair : Introduction to OOPS, Introduction to Map, Highest Frequency Character, Highest Frequency Character, Heap, object-oriented programming system.	11
	Tutorials	15

	Practical	30
	Total Theory + Tutorial + Practical	90

LIST OF PRACTICALS

1. Write a program using recursion to calculate the factorial of a number and generate the Fibonacci sequence.
2. Solve a problem using backtracking, such as the subset sum or finding a path through a maze.
3. Create a program using ArrayList and Map to store and display student records or product details.
4. Implement binary search to find an element in a sorted list.
5. Compare binary search and linear search by measuring their performance on large lists.
6. Implement sorting algorithms like selection sort and quick sort to arrange numbers in ascending order.
7. Create a linked list with insert and delete operations, and display the updated list after each operation.
8. Implement a stack using arrays or a linked list and perform push, pop, and peek operations.
9. Create a queue and a circular queue, and demonstrate enqueue, dequeue, and display functions.
10. Build a binary tree and write functions for inorder, preorder, and postorder traversal.
11. Create a graph using either an adjacency list or matrix and print its connections.
12. Use a heap structure to find the k-largest or k-smallest numbers in a list.
13. Implement object-oriented programming concepts like class, object, and inheritance in a sample program.
14. Build a mini project that combines OOP concepts with data structures (e.g., library system, student portal).
15. Write code examples to analyze time complexity and understand Big O notation through practice.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorials	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	

Revision	
Others If any:	
Total Number of Contact Hours	90

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	Yes
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Think Python: How to Think Like a Computer Scientist	Allen B. Downey
2	Automate the Boring Stuff with Python	Al Sweigart
3	Python Programming: An Introduction to Computer Science	John Zelle

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 108
Course Title	Agile Software Development Life cycle
Semester	II
Type of Course	DSC
Credits	3+1+0
Course Prerequisites	–
Course Objective(s)	1. Modern Skills: Agile is the industry standard for software development, preparing students for real-world practices. 2. Teamwork Prowess: Agile emphasizes collaboration, a vital skill for successful software projects. 3. Adaptable to Change: Software requirements often evolve - Agile teaches students to be flexible and adjust. 4. Lifelong Learning: Agile promotes continuous improvement, a valuable mindset for any software engineer.
Course Outcome (CO)	CO1: Agile Methodology Fluency: Students gain a solid understanding of various Agile methodologies like Scrum and Kanban, allowing them to effectively participate in Agile projects. CO2: Project Management: They learn essential Agile project management skills, including user story estimation, backlog management, and tracking progress through tools and techniques. CO3: Collaboration and Communication: Agile practices emphasize teamwork and communication, equipping students with the ability to collaborate effectively with diverse stakeholders. CO4: Adaptability: Agile embraces change, and through this learning, students acquire the skills to adapt to evolving requirements and unexpected challenges throughout the development process.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	2	2	2	1	2	1	2	2	1	2	2	2
CO2	1	3	2	2	2	1	3	2	2	2	3	2
CO3	1	2	2	3	2	2	2	2	2	2	2	3
CO4	2	2	2	2	3	2	2	2	3	2	2	2
Avg.	1.5	2.25	2	2	2.25	1.5	2.25	2	2	2	2.25	2.25

1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Agile and Traditional SDLC: Waterfall model, Agile Manifesto, values (individuals & interactions, working software, collaboration & response to change), principles, benefits (faster feedback, adaptability, motivation).	12
UNIT II	Agile Methodologies and Practices: Scrum, Kanban, Extreme Programming (XP), user stories, sprint planning, daily stand-ups, sprint reviews, retrospectives, Product Owner, Scrum Master, Development Team, roles, responsibilities, and interactions.	12
UNIT III	Implementing Agile in Practice: User story estimation, backlog management, velocity tracking, project management, testing, quality assurance, continuous integration (CI), continuous delivery (CD), Agile tools, project boards, communication platforms, code repositories.	12
UNIT IV	Case studies: Real-life examples of companies using Agile successfully	9
	Tutorial	15
	Total Theory + Tutorial	60

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorials	15
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	60

ASSESSMENT METHODS

Formative	Summative
-----------	-----------

Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	Yes
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Agile Software Development: Principles, Patterns, and Practices	Robert C. Martin
2	Scrum: The Art of Doing Twice the Work in Half the Time	Jeff Sutherland
3	The Agile Samurai	Jonathan Rasmusson

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 109
Course Title	Web Development Frontend 1 - HTML CSS
Semester	II
Type of Course	DSC
Credits	2+1+2
Course Prerequisites	–
Course Objective(s)	1. Foundational Building Blocks: Introduce the fundamental building blocks of web development, including HTML for structuring content and CSS for styling the visual presentation. 2. Interactive Elements: Equip students with the ability to create interactive web pages using basic HTML elements like forms, buttons, and links. 3. Visual Design and Layout: Enable students to understand and apply CSS properties to control the layout, appearance, and responsiveness of web pages across different devices. 4. Version Control and Collaboration: Introduce the concept of version control using Git and GitHub, allowing students to effectively manage code changes, collaborate with others, and track project history.
Course Outcome (CO)	CO1: Solid understanding of web development fundamentals: Students gain a firm grasp of the core concepts of building web pages, including HTML structure and CSS styling. CO2: Ability to create basic interactive web experiences: They can build functional websites with basic user interactions, forms, and navigation elements. CO3: Effective visual design skills: Students can apply CSS to control the layout, appearance, and responsiveness of their web pages, ensuring a visually appealing user experience. CO4: Proficiency in version control and collaboration: They can effectively utilize Git and GitHub to manage code versions, collaborate with others on projects, and track code history, laying the foundation for future development endeavors.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	1	1	2	1	1	3	2	1	1	2
CO2	3	3	1	2	2	1	1	3	3	1	2	2
CO3	2	2	1	2	2	2	1	2	2	1	1	2
CO4	2	3	1	3	2	1	2	2	2	1	2	3
Avg.	25	2.5	1	2	2	1.25	1.25	2.5	2.25	1	1.5	2.25
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Fundamentals of Web Development and Basic HTML: Introduction to HTML and Web Development, Understanding the Internet, Browser, and the World Wide Web, The Role of Web Developers, Web Development Tools and Environments, HTML (Hypertext Markup Language), Structure of HTML Documents, HTML Elements and Tags, Creating Links and Anchors, Working with Forms and Input Elements	7
UNIT II	Cascading Style Sheets (CSS) Basics: Introduction to CSS, CSS Selectors and Properties, Styling Text (Font properties, Text alignment, Text decoration, Text transformation), Styling Colors and Backgrounds (Color property, Background color, Background image, Background repeat, Background position), Box Model Basics (Margin, Border, Padding, Width, Height), Introduction to Layout (Block-level vs Inline elements, Display property, Floats), Positioning Basics (Static, Relative, Absolute, Fixed), Introduction to Responsive Design (Viewport meta tag, Media queries, Fluid layouts).	7
UNIT III	Advanced CSS Concepts: Flexbox Layout, Advanced Selectors and Pseudo-classes, Transitions and Animations, Responsive Images and Media Queries, Typography and Web Fonts, CSS Preprocessors (e.g., Sass, Less), Cross-Browser Compatibility and Vendor Prefixes, Performance Optimization Techniques.	8
UNIT IV	Git & Github: Git basics (repositories, commits, branches, merges, staging), GitHub features (repositories, issues, pull requests, collaboration tools), working with remote repositories, branching/merging, collaborative development on GitHub	8

	(forking, pull requests, code reviews), GitHub workflows, and best practices/tips for effective Git/GitHub usage (commit messages, branching strategies, repository organization, pitfalls).	
	Tutorials	15
	Practical	60
	Total Theory + Tutorials + Practical	105

LIST OF PRACTICALS

1. Create your personal profile web page using HTML with sections like introduction, hobbies, and goals.
2. Build a simple landing page for a restaurant that includes the name, menu, images, and contact details.
3. Design a portfolio or resume website with sections for education, projects, and contact info.
4. Add a contact form to your site using different input types like text, email, and radio buttons.
5. Style a navigation bar using CSS, including horizontal links and hover effects.
6. Use Flexbox to create a responsive layout with a header, sidebar, and main content area.
7. Add media queries to make your webpage look good on mobile, tablet, and desktop.
8. Create a pricing table using CSS Grid with columns for Basic, Standard, and Premium plans.
9. Add smooth transitions and hover effects to buttons and images using CSS.
10. Use Google Fonts and external CSS stylesheets to improve typography and design consistency.
11. Create a styled HTML table for showing data like schedules, scores, or bookings using CSS.
12. Design a homepage for a Learning Management System (LMS) with sections like dashboard, courses, and announcements.
13. Create an image gallery with thumbnails, hover effects, and lightbox-style layout.
14. Practice version control by pushing your website code to a GitHub repository.
15. Collaborate with classmates on GitHub using branches, pull requests, and merge approvals.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	30
Tutorial	15
Practical	60
Seminar/Journal Club	
Small group discussion (SGD)	

Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	105

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Books	Author
1	HTML and CSS: Design and Build Websites	Jon Duckett
2	Learning Web Design	Jennifer Robbins
3	CSS: The Definitive Guide	Eric A. Meyer

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 110
Course Title	New Age Life Skills 2
Semester	II
Type of Course	VAC
Credits	1+1+1
Course Prerequisites	–
Course Objective(s)	1. Identify common communication problems that may be holding learners back 2. Perceive what the non-verbal messages are communicating to others 3. Understand the role of communication in the teaching-learning process 4. Learn to communicate through the digital media 5. Understand the importance of empathetic listening 6. Explore communication beyond language
Course Outcome (CO)	CO1: Understand the fundamentals of effective communication, including verbal and non-verbal techniques. CO2: Identify areas for self-improvement and apply strategies to enhance communication skills such as active listening and confident speaking. CO3: Apply nonverbal communication effectively to support and enhance verbal interactions. CO4: Demonstrate the use and impact of digital marketing tools in business communication to reach a global audience.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	1	1	1	3	2	2	1	1	1	1	1	3
CO2	1	1	2	3	3	2	2	1	2	2	2	3
CO3	1	1	2	3	2	2	2	1	2	1	2	3
CO4	2	2	2	3	3	2	3	2	2	3	3	3
Avg.	1.25	1.25	1.75	3	2.5	2	2	1.25	1.75	1.75	2	3
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Reading Strategies: Main theme of the text/passage, Author's purpose of the text, text structure, point of view text, inferences from the text, introduction to the fictional, literary, and social text, Character sketch, intermediate/advanced comprehension passage practice. Grammar: Tenses, Direct and indirect Speech, Active and passive, editing and omitting. Listening: Audio-based assignments, and audio-passages, podcasts, talk shows with assignments, exercises. Watching: Documentaries. Writing: Developing coherent paragraphs, writing short essays on a topic, and descriptive writing. Speaking: Delivering speech and presentations, beginner/intermediate level interactions in English with an individual, Making enquiries related to a topic/need/situation, expressing opinions, agreements, and disagreements.	3
UNIT II	Critical thinking and design thinking: Critical Evaluation and Problem Solving, Creative Solutions, Analyzing information objectively, Identifying key factors and variables, Defining the problem clearly, Generating alternative solutions, Thinking outside the box, Encouraging innovation in problem-solving. Gender: Unlearning Gender Roles and Stereotypes, Understanding Choice and Agency, Consent and LGBTQIA+, Recognizing and challenging societal expectations, Promoting equality and inclusivity, Empowering individuals to make autonomous decisions, Recognizing the impact of personal choices, Respecting boundaries and consent, Creating an inclusive environment for LGBTQIA+ individuals.	4
UNIT III	Food and Nutrition: Informed and Healthier Food Choices, Positive Eating Behaviors, Overall Well-being through Nutrition, Providing information on nutritional values, Encouraging mindful eating, Promoting a balanced and varied diet, Addressing emotional eating habits, Connecting nutrition to physical and mental well-being, Creating sustainable and healthy eating habits Financial literacy: Different Components of Financial Literacy, Managing Finances, Healthy Relationship with Money, Basics of budgeting, saving, and investing, Credit and debt management, Creating a personal budget, Setting financial goals and priorities, Fostering a positive mindset towards finances, Balancing spending and saving habits	4

UNIT IV	Office Tools: Setting Page style -Formatting Page, Page tab, Margins, Layout settings, Paper tray, Border & Shading, Columns, Header & footer, Setting Footnotes & endnotes – Shortcut Keys; Inserting manual page break, Column break and line break, Creating sections & frames, Anchoring & Wrapping, Setting Document styles, Table of Contents, Index, PageNumbering, date & Time, Author etc., Creating Master Documents, Web page. Creating Tables- Table settings, Borders, Alignments, Insertion, deletion, Merging, Splitting, Sorting, and Formula, Drawing - Inserting ClipArts, Pictures/Files etc., Tools – Word Completion, Spell Checks, Mail merge, Templates, Creating contents for books, Creating Letter/Faxes. Page layout & view.	4
	Tutorials	15
	Practical	30
	Total Theory + Practicals	60

LIST OF PRACTICALS

1. Listen to a podcast or talk show and write a short summary highlighting the main points and your reflections.
2. Write a paragraph on a given topic or prompt such as your hobby, your favorite place, or a life lesson.
3. Write a short essay on a topic like your favorite movie, an inspiring person, or a current event.
4. Prepare and deliver a 2-minute presentation using visual aids like slides, posters, or charts.
5. Participate in an enquiry roleplay where you practice making and responding to polite requests.
6. Join a classroom debate and take a clear position on a topic, supporting it with reasons and examples.
7. Take part in a gender awareness activity by discussing real-life scenarios and how to respond respectfully.
8. Join a 1-hour design thinking challenge in teams to solve a creative or real-world problem.
9. Track your meals for one week in a food diary and write a reflection about your eating habits.
10. Create a monthly budget using a spreadsheet tool to track your income, expenses, and savings.
11. Participate in a mock investment challenge by selecting virtual stocks or mutual funds and tracking their progress.
12. Format a digital document by adding section breaks, headers, footers, and a table of contents.
13. Use mail merge to generate personalized certificates, letters, or event invites from a data source.

14. Create a master document by combining smaller documents or chapters into one formatted file.
15. Design a professional report or newsletter using formatting tools, visuals, and proper layout structure.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	15
Tutorials	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	60

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	Yes
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Emotional Intelligence	Daniel Goleman
2	Critical Thinking	Richard Paul & Linda Elder
3	The Design Thinking Playbook	Michael Lewrick, Patrick Link, Larry Leifer

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 111
Course Title	Introduction to Generative AI
Semester	II
Type of Course	SEC
Credits	4+1+0
Course Prerequisites	–
Course Objective(s)	1. Understand the fundamentals of AI technologies and their applications in various industries. 2. Learn to identify opportunities to leverage AI solutions to solve problems within your organization. 3. Gain insights into the capabilities and limitations of current AI technologies. 4. Explore the societal impact of AI and learn strategies to navigate through technological changes. 5. Develop a sustainable AI strategy tailored to your business needs, whether you're a non-technical professional or a machine learning expert.
Course Outcome (CO)	CO1: Gain a comprehensive understanding of AI technologies and their practical applications across various industries. CO2: Identify opportunities within your organization to leverage AI solutions effectively to address challenges and optimize processes. CO3: Develop a nuanced understanding of the capabilities and limitations of contemporary AI, enabling informed decision-making. CO4: Recognize the societal implications of AI and acquire strategies to navigate through the impacts of technological advancements.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	3	2	2	2	3	3	2	3	2	2
CO1	2	3	2	2	3	2	2	2	3	2	3	2
CO1	3	2	3	2	2	2	3	3	3	3	2	2
CO1	2	2	3	3	3	3	2	2	2	2	2	3
Avg.	2.5	2.25	2.75	2.25	2.5	2.25	2.5	2.5	2.5	2.5	2.25	2.25
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	What is AI: Introduction, Machine Learning, What is data, The terminology of AI, What makes an AI company?, What Machine Learning can and cannot do, Intuitive explanation of deep learning	15
UNIT II	Building AI Projects: Workflow of a Machine Learning project, Workflow of a Data Science project, Every job function needs to learn to use data, How to choose an AI project, Working with an AI team, Technical tools for AI teams.	15
UNIT III	AI in Your Company: Case study: Smart speaker, Case study: Self-driving car, Example roles of an AI team, AI Transformation Playbook, AI pitfalls to avoid, Taking your first step in AI, Survey of major AI applications, Survey of major AI techniques.	15
UNIT IV	AI and Society: A realistic view of AI, Discrimination / Bias, Adversarial attacks, Adverse uses, AI and developing nations, AI and jobs, Conclusion.	15
	Tutorial	15
	Total Theory + Tutorial	75

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	60
Tutorial	15
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	Yes
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Deep Learning with Python	François Chollet
2	Artificial Intelligence: A Modern Approach	Stuart Russell & Peter Norvig (for foundational understanding)
3	AI for Everyone (Coursera companion)	Andrew Ng (Online + Notes)

SEMESTER-3

Course Type	Course Code	Course Title
DSE	BCA 201	Web Development Frontend 2 -JS React
DSC	BCA 202	Operating System Fundamentals
DSC	BCA 203	Computer Network
DSE	BCA 204	Data Structures and Algorithms (Problem Solving 3)
AEC	BCA 205	New Age Life Skills 3
SEC	BCA 206	Introduction to ChatGPT

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 201
Course Title	Web Development Frontend 2 - JS React
Semester	III
Type of Course	DSE
Credits	2+1+2
Course Prerequisites	–
Course Objective(s)	1. To establish a strong foundation in modern JavaScript (ES6+) and its application in manipulating the DOM. 2. To master the core principles of React, including component-based architecture, state management with Hooks, and JSX. 3. To gain practical experience in building and deploying complete, interactive single-page applications using the React ecosystem.
Course Outcome (CO)	CO1: Build dynamic web features using modern JavaScript (ES6+) and direct DOM manipulation. CO2: Develop modular user interfaces using React's component-based architecture and manage application state effectively with Hooks. CO3: Construct and deploy complete single-page applications featuring client-side routing, API integration, and modern styling. CO4: Apply performance optimization techniques and best practices in React applications to enhance user experience and ensure scalability.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	1	1	2	1	2	3	2	1	2	2
CO2	3	3	1	2	2	1	2	3	2	1	2	2
CO3	3	3	2	2	2	2	3	3	2	2	3	2
CO4	2	2	2	2	2	1	2	2	2	2	2	2
Avg.	2.75	2.5	1.5	1.75	2	1.25	2.25	2.75	2	1.5	2.25	2
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to JavaScript: JavaScript Fundamentals: Introduction to JavaScript, variables and scope (var, let, const), data types (primitive and complex), operators (arithmetic, comparison, logical), control structures (if/else, switch, loops), functions (declarations, expressions), arrays, and objects, ES6+ Features: Arrow functions, template literals for string interpolation, destructuring for arrays and objects, Asynchronous JavaScript: Understanding the event loop, callbacks, Promises, and async/await, Advanced Array Methods: In-depth use of map(), filter(), and reduce().	7
UNIT II	The Document Object Model (DOM): The DOM and its Structure: What is the DOM?, understanding the hierarchical tree structure of a web page, Accessing DOM Elements: Methods like getElementById, getElementsByTagName, and modern selectors querySelector and querySelectorAll, DOM Manipulation: Changing inner HTML and text content, modifying element attributes and styles, creating, adding, and removing elements from the DOM, Event Handling: Attaching event listeners, the event object, common events (click, mouseover, keydown), understanding event bubbling and capturing, Debugging: Using browser developer tools to inspect elements, debug JavaScript, and analyze console errors.	7
UNIT III	Introduction to React: Getting Started with React: What is a React?, core philosophy, key features (component-based architecture, the virtual DOM, JSX), Setting Up a React Environment: Installing Node.js and npm, initializing a new project with Vite, Creating React Components: Writing Functional Components, understanding JSX syntax and its rules	8

	(e.g., single root element, camelCase properties), State and Props: Managing local state with the useState Hook, passing data down with props, understanding unidirectional data flow, Component Lifecycle and Rendering: Handling user events, implementing conditional rendering, and an introduction to component lifecycle events with the useEffect Hook for basic side effects.	
UNIT IV	Building Interactive Web Applications with React: Advanced State Management: Using the useContext Hook for global state to avoid "prop drilling", Client-Side Routing: Setting up routes with React Router, using BrowserRouter, Route, Link, and useNavigate for navigation, Handling Forms: Creating controlled components for predictable form behavior, managing form state, and performing user input validation, API Integration: Fetching data from external APIs using the fetch API with async/await, managing loading and error states in the UI, Styling in React: In-depth look at CSS Modules for scoped styles and using a utility-first framework like Tailwind CSS, Optimization and Deployment: Introduction to component optimization with React.memo, building the application for production, and deploying to a cloud service like Vercel or Netlify.	8
	Tutorial	15
	Practical	60
	Total Theory + Tutorial + Practical	105

LIST OF PRACTICALS

1. Create a To-Do List feature using useState to add, update, and delete tasks.
2. Integrate a Quiz Module to test users with multiple-choice questions using dynamic state handling.
3. Design the app as a Single Page Application (SPA) using React Router for navigation (Home, Quiz, Tasks, Profile, etc.).
4. Add a Blog Section that displays dummy blog posts using reusable components.
5. Build a Resume Builder page with form inputs and preview sections using props and reusable components.
6. Fetch external data (e.g., user profile info) using useEffect and display it on the dashboard.
7. Style your app using CSS Modules or Tailwind CSS for consistency and responsiveness.
8. Implement form validation for all forms using controlled components with real-time error feedback.
9. Add a Dark Mode Toggle and apply theme-based styling using state.
10. Use the Context API to manage global state for user data and theme preferences.
11. Create a Multi-Step Form for collecting user information across pages or sections.

12. Deploy the complete app to Netlify or Vercel and ensure it runs live.
13. Optimize performance by using React.memo to avoid unnecessary re-renders.
14. Use map(), filter(), and reduce() functions to dynamically render and manage data (tasks, blog, quiz, etc.).
15. Build and use a Custom Hook for reusable logic (e.g., input handling, theme toggling, or API fetching).

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	30
Tutorial	15
Practical	60
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL)	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	105

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination

End Term Semester Examination	University Examination
-------------------------------	------------------------

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Learning React: Modern Patterns for Developing React Apps	Alex Banks & Eve Porcello
2	Fullstack React: The Complete Guide to ReactJS and Friends	Anthony Accomazzo, Nathaniel Murray, Ari Lerner
3	React Up & Running	Stoyan Stefanov

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 202
Course Title	Operating System Fundamentals
Semester	III
Type of Course	DSC
Credits	2+1+1
Course Prerequisites	–
Course Objective(s)	1. To understand the architecture and core components of the Linux operating system. 2. To learn how Linux manages processes, memory, and the file system. 3. To develop proficiency in the Linux command line and shell scripting for system interaction.
Course Outcome (CO)	CO1: Navigate the Linux environment and manage files and permissions proficiently from the command line. CO2: Write shell scripts to automate system administration tasks. CO3: Apply knowledge of process, memory, and file system management to perform basic Linux system administration and troubleshooting. CO4: Configure and manage essential Linux services such as networking, user authentication, and software installation to maintain system functionality and security.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	1	2	2	2	1	3	2	1	2	2
CO2	3	3	2	2	2	2	1	3	3	2	2	2
CO3	3	3	2	2	2	3	2	3	3	2	2	2
CO4	3	3	2	3	2	3	2	3	2	2	3	2
Avg.	3	2.75	1.75	2.25	2	2.5	1.5	3	2.5	1.75	2.25	2
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction: What is an Operating System, Core Functions and Services, Types of Operating Systems (Batch, Time-Sharing, Distributed, Real-Time), OS Architecture: Monolithic vs Microkernel, System Structures: The role of the Kernel and Shell, System Calls as the OS interface, Overview of Popular Operating Systems: Brief introduction to Microsoft Windows, Apple's macOS and iOS, and Google's Android, Focus on Linux as a case study for all OS concepts, Linux Fundamentals: History and Philosophy of Unix/Linux, The Filesystem Hierarchy Standard (FHS), Navigating the Filesystem (cd, ls, pwd, find, etc.)	8
UNIT II	The Shell (bash syntax, command structure, I/O redirection, piping) and Scripting Basics (variables, control structures), File and Directory Management (cp, mv, rm, mkdir), User and Permissions: The file permission model (chmod, chown), Introduction to System Configuration: Role of key files in /etc, Introduction to Security: Principles of user privilege and access control, Introduction to Package Management: Understanding repositories, installing, updating, and removing software using apt (for Debian/Ubuntu)	7
UNIT III	Process Concepts: The Process in Linux, Process States, Process Control Block, Process Creation: fork() and exec() system calls, Process Scheduling: Summary of classic algorithms, The Completely Fair Scheduler (CFS), Inter-Process Communication: Pipes and Semaphores, Process Control and Monitoring: Using ps and top to inspect processes, kill to terminate, and nice to manage priority, Memory Concepts: Physical and Virtual Memory, Memory Management in Linux: Paging and Demand Paging, Swap Space: Role and Management, Virtual Memory Management: Monitoring memory usage with free and /proc/meminfo, Page Replacement in Linux: The LRU-like approach	8
UNIT IV	The Virtual File System (VFS), Linux File Systems: ext4 journaling and structure, File Allocation and Management: Inodes, Directory Structure, Free Space Management, Disk Scheduling and Management: The I/O scheduler in Linux, Mounting and Unmounting File Systems (mount, umount), analyzing disk usage with df and du, Practical Service Management with systemd: Using systemctl to manage services and journalctl to view logs	7
	Tutorial	15

	Practical	30
	Total Theory +Tutorial + Practical	75

LIST OF PRACTICALS

1. Navigate directories using Linux commands (ls, cd, pwd).
2. File and directory operations (touch, mkdir, cp, mv, rm).
3. Change file permissions and ownership (chmod, chown).
4. Use text editors like nano or vim to create scripts.
5. Write shell scripts to automate file backup.
6. Create a script using control structures (if/else, loops).
7. Monitor processes using ps, top, kill.
8. Use grep, find, and awk for searching text.
9. Practice inter-process communication with pipes.
10. Install and manage packages using apt.
11. Write a script to check disk usage and memory.
12. Use cron to schedule jobs.
13. Create a user and manage permissions.
14. Study process management with fork(), exec().
15. Explore virtual memory and page file analysis.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	30
Tutorial	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL)	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	75

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Operating System Concept	Abraham Silberschatz, Peter B. Galvin, Greg Gagne
2	Modern Operating Systems	Andrew S. Tanenbaum
3	Operating Systems: Internals and Design Principles	William Stallings

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 203
Course Title	Computer Network
Semester	III
Type of Course	DSC
Credits	2+1+0
Course Prerequisites	–
Course Objective(s)	1. To understand the layered architecture of network models like TCP/IP. 2. To learn the functions and protocols of the Physical, Data Link, Network, Transport, and Application layers. 3. To gain hands-on experience in network configuration, analysis, and troubleshooting using modern tools.
Course Outcome (CO)	CO1: Explain the function of each layer in the OSI and TCP/IP models and identify key protocols at each layer. CO2: Analyze network traffic and packet structures using tools like Wireshark. CO3: Design and configure simple LANs, perform IP subnetting, and implement static routing protocols

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	2	1	2	1	1	3	2	2	1	1
CO2	3	3	2	2	3	1	1	3	3	2	2	1
CO3	3	3	3	3	3	1	1	3	3	3	2	1
Avg.	3.0	2.7	2.3	2.0	2.7	1.0	1.0	3.0	2.7	2.3	1.7	1.0
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to Computer Networks: Understanding what a network is, Knowing types like LAN and Wi-Fi, Learning about cables and devices like routers and switches, Basics of internet speed and delay.	7
UNIT II	Data Link Layer Functions: How data is sent in small pieces, Simple ways to check and fix errors, How devices share a connection, Basics of Ethernet and Wi-Fi, multiple access protocols.	8
UNIT III	Network Layer Functions: What IP addresses are, Basics of how data finds its way on the internet, Simple idea of routers and common network tools like DHCP, IPV4 and IPV6. Transport Layer Functions: How apps send and receive data, Difference between TCP (safe) and UDP (fast), Basics of how data flow is managed.	8
UNIT IV	Application Layer Protocols: Common things like websites (HTTP/HTTPS), email, and DNS, Basics of keeping data safe with firewalls, VPN, and SSL, Practice making a small network, Watching how data moves, and building simple tools like chat apps or file sharing.	7
	Tutorial	15
	Total Theory + Tutorial	45

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	30
Tutorial	15
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL)	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	

Total Number of Contact Hours	
-------------------------------	--

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Computer Networking: A Top-Down Approach	James F. Kurose, Keith W. Ross
2	Data Communications and Networking	Behrouz A. Forouzan
3	Computer Networks	Andrew S. Tanenbaum

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 204
Course Title	Data Structures and Algorithms (Problem Solving 3)
Semester	III
Type of Course	DSE
Credits	2+1+2
Course Prerequisites	–
Course Objective(s)	1. To master advanced algorithmic paradigms including Recursion, Backtracking, Dynamic Programming, and Greedy algorithms. 2. To design and implement solutions for complex problems using advanced data structures. 3. To develop the ability to analyze and optimize code for time and space complexity, preparing for competitive programming and technical interviews.
Course Outcome (CO)	CO1: Solve complex computational problems by identifying and applying the appropriate algorithmic paradigm (Greedy, Backtracking). CO2: Implement and utilize advanced data structures like Trees, Heaps, and Graphs to build efficient solutions. CO3: Analyze the time and space complexity of algorithms and write optimized, production-quality code.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	3	2	2	2	1	1	3	3	2	2	1
CO2	3	3	3	2	3	1	1	3	3	2	2	1
CO3	3	3	3	3	3	2	1	3	3	3	2	2
Avg.	3.0	3.0	2.7	2.3	2.7	1.3	1.0	3.0	3.0	2.3	2.0	1.3
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Strings, and Searching & Sorting: String Algorithms, Advanced Pattern Matching, String Processing Applications, Implementation and Algorithms, Searching Algorithms, Advanced Sorting Algorithms, Algorithm Comparisons.	7
UNIT II	Hashing and Trees: Hash Table Fundamentals, Collision Resolution Techniques, Dictionaries, Advanced Hashing Concepts, Tree Fundamentals, Tree Traversals (Inorder, Preorder, Postorder), Specialized Trees (AVL, B-Trees), Heap Structures and Applications.	8
UNIT III	Graphs and Traversal Techniques: Graph Representations (Adjacency Matrix/List), Graph Traversals (BFS, DFS), Path Algorithms (Dijkstra, Floyd-Warshall), Minimum Spanning Tree Algorithms, Advanced Graph Concepts, Topological Sorting, Applications of Graphs.	7
UNIT IV	Algorithmic Paradigms and Advanced Algorithms: Divide and Conquer, Greedy Algorithms, Dynamic Programming, Backtracking, String Matching Techniques, Numerical Algorithms, Geometric Algorithms, Introduction to NP-Complete Problems and Approximation Techniques.	8
	Tutorial	15
	Practical	60
	Total Theory + Tutorial + Practical	105

LIST OF PRACTICALS

1. Implement a program to search a student ID using linear and binary search and compare their performance on large datasets.
2. Sort student scores using bubble, selection, insertion, merge, and quick sort and compare their efficiency.
3. Simulate an undo operation using a stack implemented with arrays and linked lists.
4. Create a banking queue system using normal and circular queues to manage customers.
5. Manage student records using singly and doubly linked lists with insert, delete, search, and reverse operations.
6. Convert a postfix expression into a binary tree and perform inorder, preorder, and postorder traversals.
7. Create a phonebook using a binary search tree with insert, delete, and search functionalities.
8. Maintain a leaderboard using min-heap and max-heap for inserting and retrieving top scores.
9. Represent a map as a graph and use BFS and DFS to find paths between locations.

10. Solve the N-Queens puzzle using backtracking and display valid board configurations.
11. Find the shortest path between two cities using Dijkstra's algorithm.
12. Design a login system using a hash table to store and retrieve usernames with collision handling.
13. Compare time and space complexity of sorting algorithms using large datasets.
14. Find the closest pair of points in a 2D plane using divide and conquer.
15. Build a friend suggestion system using graph traversal to find mutual connections.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	30
Tutorial	15
Practical	60
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL)	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	105

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes

End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Data Structures and Algorithms Made Easy	Narasimha Karumanchi
2	Introduction to Algorithms (CLRS)	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
3	Grokking Algorithms	Aditya Bhargava

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 205
Course Title	New Age Life Skills 3
Semester	III
Type of Course	AEC
Credits	1+0+1
Course Prerequisites	–
Course Objective(s)	1. Identify common communication problems that may be holding learners back 2. Perceive what the non-verbal messages are communicating to others 3. Understand the role of communication in the teaching-learning process 4. Learn to communicate through the digital media 5. Understand the importance of empathetic listening 6. Explore communication beyond language
Course Outcome (CO)	CO1: Get a clear understanding of good communication skills. CO2: Know what they can do to improve their communication skills. CO3: Utilize active listening in communication CO4: Speak confidently.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	2	1	2	1	2	3	2	2	1	1	1	1
CO2	2	2	2	1	2	3	2	2	2	1	1	1
CO3	2	2	3	2	2	3	2	2	2	2	1	1
CO4	2	2	3	2	3	3	2	2	2	2	2	1
Avg.	2.0	1.8	2.5	1.5	2.3	3.0	2.0	2.0	1.8	1.5	1.3	1.0
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Advance Reading: Author's purpose and tone, text structure, Writing: Organizing writing, Developing texts/essays on a topic, framing arguments (logical and factual). Grammar: Editing & Omitting. Listening: Advanced audio-based assignments, and audio-passages with exercises. Watching: Documentary-based assignments/presentations. Speaking: Debates, understanding of appropriacy for conversations, confident interactions in English with individuals.	4
UNIT II	Sex Education and Sexual Health: Basic knowledge of sexual health and reproductive processes, Understanding sexual well-being and healthy behaviors, Promoting positive attitudes towards relationships, Learning the importance of consent and how to prevent harm in intimate situations.	3
UNIT III	Basic First Aid and Emergency Care: Awareness of risks related to injuries and illnesses, Simple ways to prevent accidents and health issues, Basic emergency response skills and first aid procedures.	3
UNIT IV	Personal Branding Basics: Creating presentations to communicate ideas clearly, Using templates, layouts, and formatting to design impactful slides, Adding visuals like photos, icons, and charts to support your story, Applying simple animation and transitions for better engagement, Practicing how to present confidently using your slides, Learning how to design a professional CV, Writing an effective cover letter, and Preparing simple visual portfolios or handouts to showcase skills.	5
	Practical	30
	Total Theory + Practical	45

LIST OF PRACTICALS

1. Participate in a mock group discussion on a social or tech topic, focusing on respectful communication and leadership.
2. Practice an elevator pitch by introducing yourself and your work in under 60 seconds.
3. Conduct a cyber hygiene audit by reviewing your social media, passwords, and online safety habits.
4. Join a classroom debate on ethical tech use, such as AI in education or privacy vs surveillance.
5. Create a digital resume or LinkedIn profile and get feedback from a peer.
6. Research a topic online and present your findings in a 3–5 minute talk.
7. Make a short explainer video or infographic using tools like Canva or PowerPoint.

8. Take part in a peer evaluation activity and give/receive structured feedback.
9. Write and send a formal email, such as requesting feedback or an internship opportunity.
10. Design a personal budget sheet to plan your monthly student expenses.
11. Plan and present a tech-awareness campaign on topics like cyberbullying or phishing.
12. Work in teams for a hackathon-style challenge to create a “Digital Etiquette Handbook.”
13. Track your activities for 3 days, reflect on time management, and identify improvements.
14. Perform a roleplay on handling workplace conflict or asking for help professionally.
15. Collaborate using tools like Google Docs or MS Teams to co-create a shared document or project plan.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	45

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination

End Term Semester Examination	University Examination
-------------------------------	------------------------

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Thinking, Fast and Slow	Daniel Kahneman
2	The Power of Now: A Guide to Spiritual Enlightenment	Eckhart Tolle
3	The 7 Habits of Highly Effective People	Stephen R. Covey

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 206
Course Title	Introduction to ChatGPT
Semester	III
Type of Course	SEC
Credits	2+1+0
Course Prerequisites	–
Course Objective(s)	1. To master the principles of effective prompt engineering for Large Language Models (LLMs). 2. To leverage AI for improving code quality, testing, documentation, and dependency management.
Course Outcome (CO)	CO1: Craft precise and effective prompts to elicit desired responses from generative AI models for various development tasks. CO2: Utilize AI tools to accelerate the software design, prototyping, and coding process. CO3: Employ AI to enhance software quality through automated test generation, documentation, and code optimization.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	2	1	2	2	3	3	2	2	2	2
CO2	3	3	3	2	3	2	2	3	3	2	3	2
CO3	3	3	3	3	3	2	2	3	3	3	3	2
Avg.	3.0	2.7	2.7	2.0	2.7	2.0	2.3	3.0	2.7	2.3	2.7	2.0
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Foundations of Prompt Engineering : Large Language Models (LLMs), Randomness in AI outputs, Setting up ChatGPT, Crafting first prompts, Understanding prompts, Prompt intuition, Introducing new information, Reusable and scalable prompt structures.	8
UNIT II	Prompt Patterns and Design Techniques: Chain-of-Thought, Question Refinement, Cognitive Verifier, Audience Persona, Flipped Interaction.	7
UNIT III	Few-Shot Prompting and Advanced Patterns: Few-shot prompting techniques, Chain-of-Thought prompting, ReAct prompting, Gameplay pattern, Template pattern, Meta-Language, Recipe, Alternative Approaches.	8
UNIT IV	Refinement, Real-World Use, and Capstone Application – Prompt refinement strategies, Evaluating prompt effectiveness, Real-world applications, Capstone integration, using co - pilot.	7
	Tutorial	15
	Total Theory + Tutorial	45

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	30
Tutorial	15
Practical	
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL)	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	45

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	The Art of Prompt Engineering with ChatGPT	Nathan Hunter
2	AI 2041: Ten Visions for Our Future	Kai-Fu Lee & Chen Qiufan
3	Hands-On ChatGPT for Beginners: Mastering Prompt Engineering and Real-World Applications	Tanmoy Roy

SEMESTER-4

Course Type	Course Code	Course Title
DSC	BCA 207	Web Development Backend 1
DSC	BCA 208	Web Development Backend (Deployment, Security, and Scalability of Web Apps)
DSE	BCA 209	System Design (Low level)
AEC	BCA 210	New Age Life Skills 4
SEC	BCA 211	Introduction to Cloud Computing

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 207
Course Title	Web Development Backend 1
Semester	IV
Type of Course	DSC
Credits	3+0+2
Course Prerequisites	–
Course Objective(s)	1. Gain a comprehensive understanding of the MERN stack (MongoDB, Express.js, Node.js) and its applications in building modern web applications. 2. Set up a development environment using Node.js and Express.js for MERN stack development. 3. Design and implement RESTful APIs for data interaction and management. 4. Create and manage database schemas using Mongoose, an ODM library for MongoDB. 5. Implement user authentication and access control mechanisms in a MERN stack application.
Course Outcome (CO)	CO1: Understand and use the MERN stack: Learn to build complete web applications using MongoDB, Express, React, and Node.js. CO2: Set up your development tools: Get your computer ready to code MERN stack applications quickly. CO3: Build secure APIs: Create APIs that send and receive data safely and efficiently. CO4: Manage user access: Secure your application with user logins and permission controls.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	3	2	2	2	2	3	3	3	2	3	2
CO2	2	2	1	1	2	1	2	2	2	1	2	1
CO3	3	3	2	1	2	3	3	3	2	2	3	2
CO4	3	2	2	1	2	3	3	3	2	2	3	2
Avg.	2.8	2.5	1.8	1.3	2.0	2.3	2.8	2.8	2.3	1.8	2.8	1.8
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to MERN Stack architecture: MongoDB, Express.js, React, and Node.js, Setting up the development environment (Node, npm, VSCode, MongoDB Atlas), Introduction to JavaScript runtime with Node.js, Understanding core modules, package management, and project structure, First backend server with Express.js, Creating a sample project integrating MongoDB and Express.js.	11
UNIT II	Understanding REST architecture, Creating APIs using Express.js (GET, POST, PUT, PATCH, DELETE), Handling routes, query parameters, and middleware, Sending JSON responses and managing status codes, Error handling and logging practices, Introduction to Postman and API testing tools.	11
UNIT III	Introduction to NoSQL and MongoDB concepts and CRUD Operations: Create: save(), create(), Read: find(), findById(), lean(), filtering, sorting, pagination, Update: findOneAndUpdate(), updateOne(), updateMany(), \$set, \$inc, Delete: deleteOne(), deleteMany(), findByIdAndDelete(), Specifying data validation rules (e.g., required fields, minimum length). Creating custom methods for specific data manipulation within the schema. CRUD Operations with Mongoose, Mongoose ODM: schema creation, validation, and virtuals, Relational data with references and population, Project: Build a complete API with database integration.	11
UNIT IV	API Development, Authentication, and Access Control: Building RESTful APIs with Mongoose: Integrating Mongoose models into Express routes for data access, Folder Structure for Large Projects: MVC pattern, service layers, Authentication & Authorization: Sessions vs Tokens, JWT: Generating, verifying,	12

	and using tokens, Secure routes with middleware, Logging and Error Tracking: Usingmorgan, winston, Creating and exporting Postman Collections for API testing, Documenting and Hosting APIs using Swagger for interactive API Documentation , Hands-on: Create a secure Login-Register system with protected routes and role access.	
	Practical	60
	Total Theory + Practical	105

LIST OF PRACTICALS

1. Set up a backend server using Node.js and Express for a simple project like a task tracker.
2. Create RESTful APIs to perform CRUD operations for a blog, product list, or student records.
3. Test your API endpoints using Postman by sending different HTTP requests (GET, POST, PUT, DELETE).
4. Connect your Express app to MongoDB or MySQL to store and retrieve data for your project.
5. Implement user authentication using JWT or session-based login in your app (e.g., login/signup).
6. Define custom routes and middlewares to separate logic and handle repeated operations like logging.
7. Add validation and error-handling middleware to ensure data integrity and give proper user feedback.
8. Build a basic blog backend with login, post creation, comment section, and user roles.
9. Enable file uploads using multer for uploading profile pictures or documents.
10. Deploy your backend project on Render or Heroku and test it live using an API client.
11. Use dotenv to manage environment variables like database URLs and secret keys securely.
12. Create a feedback form API where users submit name, message, and rating—store responses in the database.
13. Organize your project using MVC structure for better code maintainability and scalability.
14. Add rate limiting and sanitize input to prevent spam and protect your backend APIs.
15. Use CORS properly and explore basic API security practices like token checks and input filtering.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Practical	60
Seminar/Journal Club	

Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	105

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	Yes
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Node.js Web Development	David Herron
2	Express in Action: Writing, Building, and Testing Node.js Applications	Evan Hahn
3	Learning MongoDB	Satyajit Singh

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 208
Course Title	Web Development Backend (Deployment, Security, and Scalability of Web Apps)
Semester	IV
Type of Course	DSC
Credits	3+1+2
Course Prerequisites	–
Course Objective(s)	1. Prepare backend code for production-ready deployment. 2. Optimize backend performance through monitoring and refactoring. 3. Learn security practices to protect user data and prevent vulnerabilities. 4. Understand proper code practices 5. Learn to deploy the code to cloud
Course Outcome (CO)	CO1: Secure your backend with middleware and encryption. CO2: Manage app state and handle errors effectively. CO3: Test and debug your backend code for smooth operation. CO4: Deploy your backend to cloud platforms like Heroku or AWS. CO5: Manage environment variables and deploy updates seamlessly. CO6: Scale your application to handle increasing user traffic. CO7: Optimize infrastructure for better performance and responsiveness.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	3	1	2	3	2	3	2	2	2	1
CO2	3	2	2	1	2	2	2	3	2	2	2	1
CO3	3	2	2	1	2	2	2	3	2	2	2	1
CO4	3	3	2	2	2	2	3	3	2	2	3	2
CO5	3	2	2	2	2	2	3	3	2	2	3	2
CO6	3	3	3	2	2	2	3	3	3	2	3	2
CO7	3	3	3	2	2	2	3	3	3	2	3	2

Avg.	3.0	2.4	2.4	1.6	2.0	2.1	2.6	3.0	2.3	2.0	2.6	1.6
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Introduction to middleware in Express.js, State management and validation, Error handling and logging (using tools like Winston, Morgan), Backend testing using Postman/Newman and writing unit tests (Jest), Code refactoring and modularization, Performance profiling tools (Node.js Profiler, Chrome DevTools).	12
UNIT II	Bottlenecks and Monitoring: Using tool like Database Indexing and Aggregation Framework in MongoDB, Caching Strategies: In-memory caching using node-cache, memory-cache, Redis Introduction: Why & How to use, Lazy loading and efficient response management, Hands-on: Implement server-side caching and query optimization in an API.	13
UNIT III	Introduction to Cloud Platforms: Heroku, Render, Vercel, Swagger, AWS (EC2, S3, IAM basics), Deployment Process: Preparing codebase for deployment, Environment variables on cloud, Handling static assets.	10
UNIT IV	CI/CD with GitHub Actions: Automatic testing and deployment pipelines, Docker Basics (Optional Advanced): Containers, Dockerizing Node.js apps, Hands-on: Deploy a MERN app on Render/Heroku with environment setup.	10
	Tutorials	15
	Practical	60
	Total Theory +Tutorials + Practical	105

LIST OF PRACTICAL

1. Feedback Tracker
2. Learning management system
3. Event
Scalable Blog App
4. Deploy your backend project to a cloud platform like Render, Railway, or Heroku.
5. Configure CORS policies to control which frontend apps can access your backend.
6. Enable HTTPS locally using self-signed certificates for secure development.
7. Use Helmet middleware to add essential security headers to your Express app.
8. Protect private API routes using JWT authentication for user authorization.
9. Apply rate limiting with express-rate-limit to prevent abuse of API endpoints.

Locator

10. Sanitize user input to prevent XSS attacks and NoSQL injection vulnerabilities.
11. Log and monitor API performance using Morgan or Winston middleware.
12. Add email verification during signup using NodeMailer or SendGrid sandbox.
13. Connect your backend to a remote cloud database like MongoDB Atlas or MySQL Cloud.
14. Set up a basic CI/CD pipeline using GitHub Actions to automate deployment.
15. Integrate Stripe or Razorpay sandbox for simulating online payments.
16. Use Redis or in-memory caching to store and retrieve frequently accessed data quickly.
17. Implement error logging for debugging and tracking issues across your backend.
18. Perform load testing using Apache Benchmark or Postman Runner to assess API performance.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorials	15
Practical	60
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	105

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	Yes
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	
Journal Club	

Professional Activity	Yes
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Web Security for Developers: Real Threats, Practical Defense	Malcolm McDonald
2	The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security	Gene Kim, Jez Humble, Patrick Debois, John Willis
3	Designing Scalable Web Applications	Kenneth C. Goldman

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 209
Course Title	System Design (Low level)
Semester	IV
Type of Course	DSE
Credits	3+1+0
Course Prerequisites	–
Course Objective(s)	1. Understand the key principles of system design, both high-level and low-level, with a focus on web application architecture. 2. Develop the skill to capture, analyze, and translate functional and non-functional requirements into concrete design decisions. 3. Learn to model, visualize, and communicate complex systems using industry-standard methodologies and notations. 4. Prepare for practical design challenges including security, performance, and scalability.
Course Outcome (CO)	CO1: Understand and describe the process of moving from user requirements to detailed software design. CO2: Identify and justify design choices balancing functional needs and performance/security trade-offs. CO3: Demonstrate ability to create HLD, LLD, ER diagrams, and system architecture documentation. CO4: Apply modeling notations (UML, CRC) for capturing and translating requirements into design. CO5: Address real-world challenges such as scalability, security, and maintainability in system design. CO6: Develop a portfolio-quality design project ready for real-world application or interview discussion.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	3	2	2	2	2	2	3	2	2	2	2
CO2	3	3	2	2	2	3	3	3	3	2	2	2
CO3	3	3	2	2	2	2	2	3	3	2	2	2
CO4	3	2	2	2	2	2	2	3	3	2	2	2
CO5	3	3	3	2	2	3	3	3	3	3	3	2

CO6	3	3	2	3	3	2	3	3	3	2	3	3
Avg.	3.0	2.8	2.2	2.2	2.2	2.3	2.5	3.0	2.8	2.2	2.3	2.2
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	HLD vs LLD: definitions and purposes, Identifying core services and modules in web applications such as Authentication, User Management, Content or Catalog, Payments, Activity Logs, Capturing Functional Requirements (e.g., 'User can enroll in a course'), Capturing Non-Functional Requirements (e.g., 'API latency < 200 ms', '99.9% uptime'), Mapping requirements to design decisions including how functional requirements suggest modules and how non-functional requirements drive design choices such as caching and SLAs	11
UNIT II	CRC cards: identifying Class, Responsibility, Collaborator, UML Class Diagrams: classes, attributes, methods, relationships, UML Sequence Diagrams: mapping API call flows, Converting wireframes and user stories into design artifacts Lab Work: CRC-card workshop for one selected feature producing four to five cards, Draw a UML class diagram and a sequence diagram for the main interaction of the selected feature	10
UNIT III	Architecture styles: Monolith, Modular Monolith, Microservices, Service boundaries including cohesion and coupling principles, Database Schema Design including ER diagrams, normalization, relationships (1:1, 1:N, M:N), indexing strategies, Designing RESTful APIs including resource naming, HTTP methods, versioning, Configuration patterns such as environment-driven versus hard-coded settings, Authentication versus Authorization, JWT and session token mechanisms, Role-Based Access Control: defining roles, permissions, resources, Enforcement points including API middleware or guards and front-end checks, Basics of data security including hashing, encryption, and input validation	12
UNIT IV	Key Non-Functional Requirements: performance, scalability, availability, maintainability, Techniques for optimization such as caching using Redis, load balancing, database indexing, read replicas, Trade-off frameworks including serverless versus dedicated servers, monolith versus microservices, ORMs versus raw SQL, Observability with logging, metrics, and alerting design, Case studies of complete web application architectures	12

	such as LMS, dashboards, or trackers, How front-end development patterns influence API and data-flow design, integrating all aspects from requirements to HLD, LLD, RBAC, and trade-offs into a cohesive design	
	Tutorial	15
	Total Theory + Tutorial	60

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorial	15
Practical	
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL)	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	60

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	
Quiz	
Seminars/ Presentation	Yes
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination

End Term Semester Examination	University Examination
-------------------------------	------------------------

FEEDBACK PROCESS

Feedback Process	Student's Feedback
-------------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	System Design Interview – An Insider's Guide	Alex Xu
2	Designing Data-Intensive Applications	Martin Kleppmann
3	The System Design Primer	Donne Martin

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 210
Course Title	New Age Life Skills 4
Semester	IV
Type of Course	VAC
Credits	1+0+1
Course Prerequisites	–
Course Objective(s)	1. Identify suitable job opportunities and tailor job application materials accordingly. 2. Develop effective interview skills and articulate strengths and experiences confidently. 3. Navigate the onboarding process and adapt to workplace culture seamlessly. 4. Set performance expectations, seek professional development, and maintain work-life balance. 5. Explore career advancement opportunities, build a professional network, and handle workplace challenges effectively.
Course Outcome (CO)	CO1: Proficiency in identifying and applying for suitable job opportunities. CO2: Confidence in interview settings and effective communication of strengths and experiences. CO3: Seamless adaptation to workplace culture and integration into job roles. CO4: Achievement of performance goals, continuous professional development, and balanced work-life management. CO5: Exploration of career growth opportunities, expanded professional network, and effective resolution of workplace challenges.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	1	2	1	2	2	2	3	2	2	1	3	2
CO2	1	1	1	3	2	2	3	1	1	1	3	3
CO3	1	1	1	3	3	2	3	1	2	1	3	3
CO4	1	1	1	2	3	2	3	1	2	1	3	3
CO5	1	1	2	2	3	2	3	1	2	2	3	3
Avg.	1.0	1.2	1.2	2.4	2.6	2.0	3.0	1.2	1.8	1.2	3.0	2.8
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Placement Prep: Job Title Identification, Responsibilities Decoding, Qualifications Assessment, Company Expectations and Culture, Adaptation to Job Descriptions, Highlighting Relevant Skills, Showcasing Achievements, Keywords and Phrases Integration, Formatting and Structure Considerations, Cover Letter Alignment.	3
UNIT II	Placement Prep: Answering common interview questions: Introduction and Background, "Why Are You Interested in This Position?", "Can You Tell Me About Yourself?", "What Are Your Strengths and Weaknesses?", "Describe a Challenging Situation You Faced at Work and How You Handled It.", "Where Do You See Yourself in Five Years?", "Why Should We Hire You?", "Tell Me About a Successful Team Project You've Worked On.", "How Do You Handle Stressful Situations?", "What Do You Know About Our Company?", "Do You Have Any Questions for Us?"	4
UNIT III	Post placement: Onboarding Process involves orientation, paperwork, and HR processes. Adapting to the Workplace Culture includes understanding company values, norms, and building relationships. Job Role Transition encompasses clarifying expectations, responsibilities, and acquiring necessary skills. Performance Expectations involve setting goals, regular reviews, and feedback sessions. Professional Development Opportunities identify skill enhancement and participation in training programs.	4

UNIT IV	Post placement: Work-Life Balance is managed by handling workload and setting boundaries. Navigating Company Policies requires familiarity and compliance with workplace regulations. Career Advancement involves exploring paths, seeking mentorship, and guidance. Building a Professional Network includes engaging with colleagues and participating in networking events. Handling Workplace Challenges requires strategies for conflict resolution and seeking support when needed.	4
	Practical	30
	Total Theory + Practical	45

LIST OF PRACTICALS

1. Write and format an academic report with headings, subheadings, and references.
2. Create a mail merge letter to send personalized event invites to multiple recipients.
3. Design a certificate in Word with editable fields for names and achievements.
4. Schedule and manage calendar invites for meetings or events using Google Calendar or Outlook.
5. Draft a formal email inquiring about a job or internship opportunity.
6. Collaborate on a shared document and track changes, suggestions, and comments.
7. Create a structured meeting agenda and minutes (MOM) for a group discussion or project meeting.
8. Write a tailored cover letter for a specific internship or job application.
9. Prepare a project proposal using a formal template and present it to peers or mentors.
10. Record a mock interview and evaluate communication, clarity, and confidence.
11. Reflect on your performance in a recent team project through a written self-assessment.
12. Create and analyze a personal monthly budget using a spreadsheet tool.
13. Deliver a 3-minute pitch on a chosen social cause, supported by facts or visuals.
14. Prepare a newsletter using MS Word with formatting tools like columns, headers, and images.
15. Write a summary report that includes headings, footnotes, and a conclusion section.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL) / Tutorial	
Problem Based Learning (PBL)	

Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	45

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination
End Term Semester Examination	University Examination

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Crucial Conversations: Tools for Talking When Stakes Are High	Kerry Patterson, Joseph Grenny, Ron McMillan, Al Switzler
2	Emotional Intelligence 2.0	Travis Bradberry & Jean Greaves
3	How to Talk to Anyone	Leil Lowndes

Name of the College	Akal College of Engineering and Technology
Name of the Program	BCA (Hons. with Research)
Course Code	BCA 211
Course Title	Introduction to Cloud Computing
Semester	IV
Type of Course	SEC
Credits	3+1+1
Course Prerequisites	–
Course Objective(s)	1. Gain a solid understanding of cloud computing fundamentals and service models, aligning with AWS terminology. 2. Explore and compare different deployment models (public, private, hybrid) and their applications within the AWS ecosystem. 3. Analyze the benefits, challenges, and security considerations of cloud adoption, drawing insights from AWS best practices. 4. Learn about key cloud computing services (IaaS, PaaS, SaaS) and their use cases, focusing on relevant AWS services like Amazon EC2, Amazon S3, and Amazon Elastic Beanstalk. 5. Identify and explore popular cloud platforms, including a strong focus on AWS and its offerings. 6. Utilize resources from Amazon Cloud Academy to complement classroom learning and gain practical experience through hands-on labs.
Course Outcome (CO)	CO1: Understand cloud computing fundamentals and AWS terminology. CO2: Compare deployment models (public, private, hybrid) within the AWS ecosystem. CO3: Analyze benefits, challenges, and security considerations of cloud adoption, guided by AWS best practices. CO4: Learn key cloud computing services and their AWS counterparts (e.g., EC2, S3, Elastic Beanstalk). CO5: Explore AWS offerings and gain practical experience through hands-on labs and resources from Amazon Cloud Academy.

Mapping of Course Outcomes (COs) to Program Outcomes (POs) & Program Specific Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO 1	PSO 2	PSO 3	PSO 4	PSO 5
CO1	3	2	2	1	2	2	2	3	2	2	2	1
CO2	3	2	2	1	2	2	2	3	2	2	2	1
CO3	2	2	2	1	2	3	2	2	2	2	2	1
CO4	3	3	2	1	2	2	3	3	2	2	2	2
CO5	3	3	2	2	2	2	3	3	3	2	3	2
Avg.	2.8	2.4	2.0	1.2	2.0	2.2	2.4	2.8	2.2	2.0	2.2	1.4
1: Weak Correlation, 2: Moderate Correlation; 3: Strong Correlation												

SYLLABUS

UNIT(S)	COURSE CONTENT	LECTURES
UNIT I	Definitions: Cloud Computing. NIST Definition of Cloud Computing. Operating System Linux: Ubuntu/Debian , Scripting (Bash, Powershell) , Cloud Deployment Models: Public, Private, Hybrid, Community Cloud Service Models: IaaS (e.g. AWS EC2, DigitalOcean Droplets), PaaS(e.g. Heroku, Firebase Hosting), SaaS , (FaaS, BaaS – emerging models), Databases: Firebase RealtimeDB(NOSQL), Firestore, AWS DynamoDB RDS(SQL), Differences Between Services and Deployment Model Cloud Ecosystem: Global infrastructure regions (Regions, Availability Zones, AWS, GCP).	11
UNIT II	IaaS (Infrastructure as a Service): AWS EC2, Instance Types, Storage/Volumes, Keypairs, Elastic IP, Cloud SQL, Firebase, Firebase Cloud Functions, CI/CD integration, Cloud Providers, Serverless (AWS Lambda, Cloudflare, Vercel, Azure, GCP Functions, Netlify), Configuration Management (Ansible, Chef, Puppet), Provisioning (AWS CDK, Cloud Formation, Pulumi, Terraform), Secret Management, Docket, LXC SaaS (Software as a Service): Google Workspace, Characteristics of SaaS products. Hands-on Labs: Deploy a static website on S3, on EC2 Setup a NoSQL database with Firebase or DynamoDB.	11
UNIT III	Cloud Identity & Access Management (IAM): Users, Groups, Roles, Policies, Principle of Least Privilege, AWS IAM, GCP IAM comparison.	11

	Networking & Protocol, FTP/FSTP, HTTP, HTTPS, SSL/TLS, SSH, OSI Model, IP4, IP6, TCP/UDP Network Security: UDP, DNS, DHCP, Wireless Security: WPA, WPA2, WPA3, Introduction to Encryption basics in the cloud, Basic networking in CIDR Block, AWS: VPC, subnets (public subnet, private subnets, security groups overview., Infrastructure Monitoring, Web Servers (Nginx, Apache), RDS: DB Instance , Storage Type, Backup/Restore, CDN: Cloudfront	
UNIT IV	Introduction to Containerization with Docker (basic concepts). Kubernetes, Containerization and virtualization. Serverless Computing: AWS Lambda, Creating/ Invoking Functions, Layers, Custom runtimes, Versioning/aliases Azure Functions, GCP Cloud Functions, Event-Driven Architectures, Use cases: Real-time image processing, APIs. Infrastructure as Code (IaC): Introduction to Terraform, AWS CloudFormation, Deployment pipelines using CI/CD: GitHub Actions, AWS CodePipeline, DevOps in Cloud: CI/CD Pipelines, Monitoring, GitOps, Observability tools, Future Trends: Cloud-Native, Multi-cloud, Edge Computing, Quantum Cloud, Green Cloud, Generative AI & ML Models on the Cloud (AWS Bedrock, GCP Vertex AI), Cloud Design Pattern.	12
	Tutorial	15
	Practical	30
	Total Theory + Tutorial + Practical	90
LIST OF PRACTICALS 1. Create a free cloud account on AWS, Azure, or Google Cloud Platform (GCP). 2. Launch a virtual machine (e.g., EC2 on AWS or Compute Engine on GCP) and access it via SSH. 3. Host a static website using Amazon S3, Firebase Hosting, or similar free-tier cloud service. 4. Deploy a backend project to Heroku, Render, or Vercel with live URL access. 5. Set up a cloud database using Firebase, Firestore, or MongoDB Atlas and connect it to your app. 6. Create a cloud storage bucket and upload/download files like images or PDFs. 7. Implement a CI/CD pipeline using GitHub Actions to automate your app deployment. 8. Experiment with Docker by containerizing a simple app and deploying it to the cloud. 9. Configure auto-scaling and load balancing using AWS or GCP free-tier tools or simulators. 10. Apply basic IAM policies to control access for different users or services in your cloud project. 11. Monitor your cloud app by viewing logs and usage stats with CloudWatch or GCP Logs.		

12. Set up environment variables securely in your cloud-hosted project for API keys and configs.
13. Deploy a simple REST API using serverless/cloud functions (e.g., Firebase Functions or AWS Lambda).
14. Create a cloud cost comparison sheet to estimate expenses across AWS, Azure, and GCP.

LEARNING STRATEGIES AND CONTACT HOURS

Learning Strategies	Contact Hours
Lecture	45
Tutorial	15
Practical	30
Seminar/Journal Club	
Small group discussion (SGD)	
Self-directed learning (SDL)	
Problem Based Learning (PBL)	
Case/Project Based Learning (CBL)	
Revision	
Others If any:	
Total Number of Contact Hours	90

ASSESSMENT METHODS

Formative	Summative
Multiple Choice Questions (MCQ)	
Viva-voce	Yes
Quiz	
Seminars/ Presentation	
Problem Based Learning (PBL)	
Journal Club	
Professional Activity	
Assignment	Yes
End Term Practical Examination	University Examination

End Term Semester Examination	University Examination
-------------------------------	------------------------

FEEDBACK PROCESS

Feedback Process	Student's Feedback
------------------	--------------------

RECOMMENDED BOOKS

Sr. No.	Book Title	Author(s)
1	Cloud Computing: Concepts, Technology & Architecture	Thomas Erl
2	Cloud Computing: Principles and Paradigms	Rajkumar Buyya, James Broberg, Andrzej Goscinski
3	AWS Certified Solutions Architect Official Study Guide	Joe Baron, Hisham Baz, Tim Bixler, Biff Gaut, Kevin E. Kelly, Sean Senior, John Stamper