BAGALKOT UNIVERSITY, JAMAKHANDI

PROGRAM /COURSE STRUCTURE AND SYLLABUS
as per the Choice Based Credit System (CBCS)
designed in accordance with
Learning Outcomes-Based Curriculum Framework (LOCF)
of State Education Policy (SEP) 2025
for
BACHELOR OF COMPUTER APPLICATIONS (BCA)

w.e.f.

Academic Year 2025-26 and onwards

Board of Studies (UG) Committee
(Computer Science)

Bachelor of Computer Applications(BCA) Programme
2025-2026

1	Dr.DAYANAND G SAVAKAR PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,BAGALKOT UNIVERSITY,JAMAKHANDI	CHAIRMAN
2	Dr.AZIZ MAKANDAR PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,KARNATAKA STATE AKKAMAHADEVI WOMEN UNIVERSITY,VIJAYAPUR	MEMBER(EXTERNAL)
3	Dr.RAMESH B MAGURI ASSISTANT PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,JANATA SHIKSHNA SANGHA SHREE TAMMANNAPPA CHIKODI ARTS AND COMMERCE COLLEGE,BANHATTI	MEMBER
4	Shri.PRAVEENKUMAR S AKKIMARADI ASSISTANT PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,BASAVESHWAR SCIENCE COLLEGE,BAGALKOT	MEMBER

Dr.DAYANAND G SAVAKAR
CHAIRMAN BOS(UG)
PROFESSOR,DEPARTMENT OF COMPUTER SCIENCE,
BAGALKOT UNIVERSITY,JAMAKHANDI

PREAMBLE

Computer Application (CA) has been evolving as an important branch of science and technology in last two decade 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. The ever- evolving discipline of computer application has strong connections to other disciplines. Many problems in science, engineering, health care, business, and other areas can be solved effectively with computers and its applications, but finding a solution requires both computer science expertise and knowledge of the particular application domain. 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.

Universities and other HEIs introduced programmes of computer application. Information Technology is growing rapidly. Increasing applications of computers in almost all areas of human endeavour has led to vibrant industries with concurrent rapid change in technology. Unlike other basic disciplines, developing core competency in this discipline that can be reasonably stable becomes a challenge. In India, it was initially introduced at the Master (postgraduate) level as MCA and M.Tech. Later on, engineering programmes such as B.Tech and B.E in Computer Science & Engineering and in Information Technology were introduced in various engineering College/Institutions to cater to the growing demand for trained engineering manpower in IT industries. Parallelly , BCA, BSc and MSc programmes with specialisation in Computer Science were introduced to train manpower in this highly demanding area.

BCA and BCA (Hons) are aimed at undergraduate level training facilitating multiple career paths. Students so graduated, can take up postgraduate programmes in CS or MCA leading to research as well as R&D, can be employable at IT industries, or can

pursue a teaching profession or can adopt a business management career.BCA and BCA (Hons) aims at laying a strong foundation of computer application at an early stage of the career. There are several employment opportunities and after successful completion of BCA, graduating students can fetch employment directly in companiesas programmer, Web Developer, Software Engineer, Network Administrator, Data Scientist, or AI/ML personnel.

The Program outcomes in BCA are aimed at allowing flexibility and innovation indesign 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 outcome-based curriculum framework, help students learn solving problems, accomplishing IT tasks, and expressing creativity, both individually and collaboratively. The proposed framework will help Students learn programming techniques and the syntax of one or more programming languages.

All students must, therefore, have access to a computer with a modern programming language installed. The computer science framework does not prescribe a specific language. The teacher and students will decide which modern programming languages students will learn. More importantly, students will learn to adapt to changes in programming languages and learn new languages as they are developed. The present Curriculum Framework for BCA degrees is intended to facilitate the students to achieve the following.

- To develop an understanding and knowledge of the basic theory of Computer Science and Information Technology with good foundation on theory, systems and applications such as algorithms, data structures, data handling, data communication and computation
- To develop the ability to use this knowledge to analyse new situations in the application domain

- To acquire necessary and state-of-the-art skills to take up industry challenges. The objectives and outcomes are carefully designed to suit to the above- mentioned purpose.
- The ability to synthesize the acquired knowledge, understanding and experiencefor a better and improved comprehension of the real-life problems
- To learn skills and tools like mathematics, statistics and electronics to find the solution, interpret the results and make predictions for the future developments
- To formulate, to model, to design solutions, procedure and to use softwaretools to solve real world problems and evaluate

PROGRAMOUTCOMES:

By the end of the program the following outcomes will be achieved by the students:

1. **Discipline knowledge:** Acquiring knowledge on basics of Computer Science and ability to apply to design principles in the development of solutions for problems of varying complexity
2. **Problem Solving:** Improved reasoning with strong mathematical ability to Identify, formulate and analyse problems related to computer science and exhibiting a sound knowledge on data structures and algorithms.
3. **Design and Development of Solutions:** Ability to design and development of algorithmic solutions to real world problems and acquiring a minimum knowledge on statistics and optimization problems. Establishing excellent skills in applying various design strategies for solving complex problems.
4. **Programming a Computer**: Exhibiting strong skills required to program a computer for various issues and problems of day-to-day applications with thorough knowledge on programming languages of various levels.
5. **Application Systems Knowledge:** Possessing a sound knowledge on computer application software and ability to design and develop app for applicativeproblems.
6. **Modern Tool Usage:** Identify, select and use a modern scientific and IT tool or technique for modelling, prediction, data analysis and solving problems in the area of Computer Science and making them mobile based application software.
7. **Communication:** Must have a reasonably good communication knowledgeboth in oral and writing.
8. **Project Management:** Practicing of existing projects and becoming independent to launch own project by identifying a gap in solutions.
9. **Ethics on Profession, Environment and Society:** Exhibiting professional ethics to maintain the integrity in a working environment and also have concern on societal impacts due to computer-based solutions for problems.

10. **Lifelong Learning:** Should become an independent learner. So, learn to learnability.
11. **Motivation to take up Higher Studies:** Inspiration to continue educationstowards advanced studies on Computer Science.

By the end of the program the students will be able to:

The Bachelor of Computer Application (BCA (Hons)) program enables students to attain following additional attributes besides the afore-mentioned attributes, by the time of graduation:

1. Apply standard Software Engineering practices and strategies in real -time software project development
2. Design and develop computer programs/computer -based systems in the areas related to AI, algorithms, networking, web design, cloud computing, IoT and data analytics.
3. Acquaint with the contemporary trends in industrial/research settings and thereby innovate novel solutions to existing problems
4. The ability to apply the knowledge and understanding noted above to the analysis of a given information handling problem.
5. The ability to work independently on a substantial software project and as an effective team member.

SEMESTER-3										
Category	Course code	Title of the Paper	Marks			Teaching hours/week			Credit	Duration of exams (Hrs)
			IA	SEE	Total	L	T	P		
L-5		Language-1	20	80	100	4	0	0	3	3
L-6		Language-2	20	80	100	4	0	0	3	3
DSC7	2E3XXXM07T	Database Management System	20	80	100	4	0	0	4	3
	2E3XXXM07L	DBMS Lab	10	40	50	0	0	4	2	3
DSC8	2E3XXXM08T	C# and .Net Framework	20	80	100	4	0	0	4	3
	2E3XXXM08L	C# and .Net Framework Lab	10	40	50	0	0	4	2	3
DSC9	2E3XXXM09T	Computer Communication and Networks	20	80	100	4	0	0	4	3
CE1	2E3XXXE01T	Multimedia & Animation	20	80	100	3	0	0	3	3
	2E3XXXE02T	Open Source Tools								
	2E3XXXE03T	Digital Marketing								
Total Marks					700	Semester Credits			25	

SEMESTER-4										
Category	**Course code**	**Title of the Paper**	**Marks**			**Teaching hours/week**			**Credit**	**Duration of exams (Hrs)**
			IA	**SEE**	**Total**	**L**	**T**	**P**		
L-7		**Language-1**	**20**	**80**	**100**	**4**	**0**	**0**	**3**	**3**
L-8		**Language-2**	**20**	**80**	**100**	**4**	**0**	**0**	**3**	**3**
DSC10	**2E4XXXM010T**	**Python Programming**	**20**	**80**	**100**	**4**	**0**	**0**	**4**	**3**
	2E4XXXM010L	**Python Programming Lab**	**10**	**40**	**50**	**0**	**0**	**4**	**2**	**3**
DSC11	**2E4XXXM011T**	**Web Technology**	**20**	**80**	**100**	**4**	**0**	**0**	**4**	**3**
	2E4XXXM011L	**Web Technology Lab**	**10**	**40**	**50**	**0**	**0**	**4**	**2**	**3**
DSC12	**2E4XXXM012T**	**Operating System Concepts**	**20**	**80**	**100**	**4**	**0**	**0**	**4**	**3**
CE2	**2E3XXXE04T**	**Machine Learning**	**20**	**80**	**100**	**3**	**0**	**0**	**3**	**3**
	2E3XXXE05T	**Internet of Things**								
	2E3XXXE06T	**Unix Operating System**								
COM3		**Skill/Practical based Learning**	**10**	**40**	**50**	**1**	**0**	**2**	**2**	**2**
Total Marks					**750**	**Semester Credits**			**27**	

Concept Note, Abbreviation Explanation and Coding:

Concept Note:

1. CBCS is a mode of learning in higher education which facilitates a student to have some freedom in selecting his/her ownchoices, across various disciplines for completing a UG/PG program.
2. A credit is a unit of study of a fixed duration. For the purpose of computation of workload as per UGC norms the following is mechanism be adopted in the university:
   One credit (01) = One Theory Lecture (L) period of one (1) hour.One
   credit (01) = One Tutorial (T) period of one (1) hour.
   One credit (01) = One practical (P) period of two (2) hours.
3. Wherever there is a practical there will be no tutorial and vice-versa
4. Vocational course is a course that enables individual to acquire skills set that are required for a particular job.
5. Internship is a designated activity that carries some credits involving more than 25 days of working in an organization (either in same organization or outside) under the guidance of an identified mentor. Internship shall be an integral part of the curriculum.

Abbreviation Explanations:

1. DSC: Discipline Specific Core Course.
2. L1: Language One
3. L= Lecture; T= Tutorial; P=Practical.
4. MIL= Modern Indian Language; English or Hindi or Telugu or Sanskrit or Urdu

Program Coding:

1. Code 25: Year of Implementation

2. Code BCA: BCA Program under the faculty of Applied Science of the University
3. Code 1: First Semester of the Program, (2 to 6 represent higher semesters)
4. Code LK: Language Kannada, FK for Functional Kannada, similarly Language English, Language Hindi, Language Sanskrit, &Language Urdu
5. Code 1: Course in that semester.

BCA III SEMESTER

Year	II	**Course Code:2E3XXXM07T** **Course Title: Database Management System**	**Credits**	04
Sem.	III		**Hours**	52
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA:03 hours.**	
Course Out comes		1. Explain the various database concepts and the need for database systems. 2. Identify and define database objects, enforce integrity constraints on a database using DBMS. 3. Demonstrate a Data model and Schemas in RDBMS. 4. Identify entities and relationships and draw ER diagram for a given real-world problem. 5. Convert an ER diagram to a database schema and deduce it to the desired normal form. 6. Formulate queries in Relational Algebra, Structured Query Language (SQL) for database manipulation. 7. Explain the transaction processing and concurrency control techniques & NOSQL MongoDB Database System.		
Unit No.		**Course Content**		**Hours**
Unit I		**Database Architecture:** Introduction to Database system applications. Characteristics and Purpose of database approach. People associated with Database system. Data models. Database schema. Database architecture. Data independence. Database languages, interfaces, and classification of DBMS.		**13**
Unit II		**E-R Model:** Entity-Relationship modeling: E – R Model Concepts: Entity, Entity types, Entity sets, Attributes, Types of attributes, key attribute, and domain of an attribute. Relationships between the entities. Relationship types, roles and structural constraints, degree and cardinality ratio of a relationship. Weak entity types, E -R diagram.		**13**
Unit III		**Relational Data Model:** Relational model concepts. Characteristics of relations. Relational model constraints: Domain constrains, key constraints, primary & foreign key constraints, integrity constraints and null values. Relational Algebra: Basic Relational Algebra operations. Set theoretical operations on relations. JOIN operations Aggregate Functions and Grouping. Nested Sub Queries-Views. Introduction to PL/SQL & programming of above operations in PL/SQL.		**13**

Unit IV	**Data Normalization:** Anomalies in relational database design. Decomposition. Functional dependencies. Normalization. Normal Forms: 1NF 2NF 3NF & BCNF. Query Processing Transaction Management: Introduction Transaction Processing. Transactions: read & write operations. Need of concurrency control: The lost update, & Dirty read problem. Types of failures. Transaction states. Desirable properties (ACID properties) of Transactions. **NOSQL &MongoDB:** Introduction, Relational vs. NoSQL, Document Model), Mongosh Shell, Databases, Collections, Documents (BSON), _id, Insert (insertOne, insertMany), Read (find, Query, Project), Update (updateOne, $set), Delete (deleteOne).	**13**

References:

1. Fundamentals of Database Systems, Ramez Elamassri, Shankant B. Navathe, 7th Edition, Pearson, 2015
2. An Introduction to Database Systems, Bipin Desai, Galgotia Publications, 2010.
3. Introduction to Database System, C J Date, Pearson, 1999.
4. Database Systems Concepts, Abraham Silberschatz, Henry Korth, S.Sudarshan, 6th Edition, McGraw Hill, 2010.
5. Database Management Systems, Raghu Rama Krishnan and Johannes Gehrke, 3rd Edition, McGraw Hill, 2002

Year	II	**Course Code: 2E3XXXM07L** **Course Title:** Database Management System LAB	**Credits**	02
Sem.	III		**Hours**	52
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 10		**Summative Assessment Marks:40**	**Duration of ESA: 03 hours.**	

Section	Course Content	Hours

PART-A

I. Execute DDL (Data Definition Language) Commands

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Create table with appropriate constraints (PRIMARY KEY)
2. Alter table to add a new column (email)
3. Drop email column
4. Rename a table student to student1
5. Drop a table student1

II. Execute DML (Data Manipulation Language) Commands

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Insert multiple records using INSERT
2. Update percentage for specific students
3. Delete students with percentage < 40

III. Execute DCL and TCL Commands

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Context: Admin role simulation

Tasks:

1. Grant SELECT, INSERT, UPDATE privileges to a user
2. Revoke privileges
3. Demonstrate use of SAVEPOINT, ROLLBACK, and COMMIT

IV. Execute Single-Line Queries and Group Functions

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Select students by specific course or percentage
2. Use functions like COUNT(), SUM(), MAX(), MIN(), AVG() on percentage column
3. Use GROUP BY and HAVING to calculate average % by course

V. Create Views for a Particular Table

Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

Tasks:

1. Create a view of students from BCA course

2. Create a view with selected columns (rollno, name, percentage)
3. Insert into and update data through view (check if updatable)
4. Drop the view

PART-B

VI. Implement Join Operations in SQL

Tables:

STUDENT(student_id, name, department)
COURSE(course_id, course_name, department)
ENROLLMENT(enroll_id, student_id, course_id, enroll_date)

Tasks:

1. Inner Join: List student names with their course names
2. Left Join: List all students with course details (if any)
3. Join + GROUP BY: Number of students enrolled in each course

VII. Implement Nested Queries

Tables:

STUDENT(rollno, name, department)
COURSE(course_id, course_name, department)
ENROLLMENT(enroll_id, student_id, course_id, enroll_date)

Tasks:

1. Find the student(s) with the highest percentage
2. Use subqueries to fetch students enrolled in a specific department's course

VIII. Implement lock Table operation with Data Insertion and commit for
Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

IX. Implement PLSQL Stored procedure add_student to insert row into table
Table: STUDENT(rollno:int, name:String, gender:String, course:String, percentage:number)

X. Implement following exercise using NOSQL Database MongoDB

Tasks

1. Create a database named myFirstDB.
2. Create a collection named items.
3. Insert the following documents into the items collection:
   i. { name: "Apple", color: "red", price: 1 }
   ii. { name: "Banana", color: "yellow", price: 0.5 }
   iii. { name: "Orange", color: "orange", price: 1.2 }
4. Retrieve all documents from the items collection.
5. Find all items that have a price less than 1.1.
6. Find the item with the name "Banana".
7. Update the price of the "Apple" to 1.1.
8. Delete the item with the name "Orange".

Year	II	**Course Code: 2E3XXXM08T** **Course Title:** .NET using C#	**Credits**	04
Sem.	III		**Hours**	52
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA:03 hours.**	
Course Out comes		1. Understand the foundational concepts of the .NET platform and C# language structure. 2. Apply Object-Oriented Programming (OOP) principles using C#. 3. Utilize interfaces, collections, delegates, and events in C# applications. 4. Develop GUI-based applications using Windows Forms and interact with databases. 5. Manage file I/O operations and understand .NET assemblies.		

Unit No.	Course Content	Hours
Unit I	**Introduction to C# .NET :** Introducing the Building Blocks of the .NET Platform (CLR, CTS, and CLS), Anatomy of C# program, The System. Environment Class, The System. Console Class, Understanding Value Types and Reference Types. NET Array Types. OOP's Concepts: Defining Classes and Creating objects, Pillars of OOP, C# constructors, Understanding C# Partial types, Understanding Boxing and Unboxing Operations.	**13**
Unit II	**Interfaces**: Defining Interfaces in C#, Implementing an Interface in C#, Contrasting Interfaces to Abstract Base Classes. **Exception handling**: The Role of .NET Exception Handing, throwing generic exceptions, catching exceptions, Configuring the state of an exception, System – Level Exception, Application-Level Exception, Processing Multiple Exception, Generic catch statements, Inner exceptions, Finally Block. **Delegates, Events**: .**NET Delegate** type, defining a Delegate in C#, Delegate Base Classes, Delegate examples, Understanding C# Events	**13**
Unit III	**GUI using Windows Forms and Database Programming**: Controls- TextBox, label, Button, checkbox, radiobutton, listbox, comboxbox ,GridView, Datetime picker, Common properties, methods and events , menus, context menus, Menustrip, Graphics and GDI, SDI and MDI, Dialog boxes; Database Programming - Understanding the Role of Managed Provider and ADO.NET Objects , Connecting to Database, Performing CRUD operations.	**13**
Unit IV	**LINQ: LINQ to Objects** LINQ-Specific Programming Constructs, Understanding the Role of LINQ, Applying LINQ Queries to Primitive Arrays, Returning the Result of a LINQ Query, and Investigating the C #LINQ Query Operators. Introduction to Entity Framework and ORM concepts; Setting up Entity Framework 1.0 in	**13**

	.NET Framework 3.5 with Visual Studio; Creating an ADO.NET Entity Data Model (EDMX) using Database First approach; Understanding the EDMX structure including Conceptual, Storage, and Mapping layers; Performing CRUD operations: inserting new records, retrieving records with LINQ to Entities, updating existing records, and deleting records; Overview of basic LINQ to Entities queries for data access	

Reference Books:

1. Andrew Troelsen: Pro C# with .NET 3.0, Special Edition aPress, India,2007.
2. E. Balagurusamy: Programming in C#,, 5th Reprint, Tata McGraw Hill,2004.
3. Herbert Schildt: The Complete Reference C#, Tata McGraw Hill,2004
4. C# 2008 programming cogent learning solutions Inc. DreamtechPress.
5. C#2008 Programming covers .net 3.5 Black Book – Beginners Edition Kogent learning solutions Inc. DreamtechPress.

Year	II	**Course Code: 2E3XXXM08L** **Course Title: .**NET using C# LAB	**Credits**	02
Sem.	III		**Hours**	52
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 10		**Summative Assessment Marks:40**	**Duration of ESA:03 hours.**	

Section	Course Content	Hours
Part A	1. Write a C# program to show the machine details like machine name, Operating System, Version, Physical Memory and calculate the time since the Last Boot Up.(Hint: Use System. Environment Class). 2. Write a program in C# to count a total number of alphabets, digits and special characters in a string. 3. Demonstrate operator overloading to add two complex numbers. 4. Write a program to implement multilevel inheritance. 5. Write a C# program to demonstrate ATM software using switch statement 6. Write a program to demonstrate System exception and Application exception. 7. Write a program that inputs the coordinates of three mouse clicks from the user and then draws a triangle in the output window using those three points.	
Part B	1. Create currency converter application. 2. Create an application to simulate the working of Font Dialog box using list boxes, label and button controls. 3. Create a GUI application to implement login page operations. 4. Create GUI application to insert the data into the database having fields such as Roll No, Name, Age and ContactNo using Execute-Non-Query. 5. Create GUI application to implement retrieve and update the data in the database. 6. Create a Windows Form using Entity Framework to Insert a Book and Display all Books with LINQ to Entities in a DataGridView. 7. Create a Windows Form using Entity Framework to Update and Delete a Book by selecting a record with LINQ queries from the DataGridView.	

Year	II	**Course Code: 2E3XXXM09T** **Course Title: Computer Communication and Networks**	**Credits**	04
Sem.	III		**Hours**	52
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA: 03 hours.**	
Course Out comes		1. Explain the transmission technique of digital data between two or more computers and a computer network that allows computers to exchange data. 2. Apply the basics of data communication and various types of computer networks in real world applications. 3. Compare the different layers of protocols. 4. Compare the key networking protocols and their hierarchical relationship in the conceptual model like TCP/IP and OSI.		

Unit No.	Course Content	Hours
Unit I	**Introduction:** Data Communication, Data Communication Model, Introduction to computer networks, evolution of networking (ARPANET, NSFNET, INTERNET), Definition, Importance, and Applications of Computer Network, Network Topologies - Star, Ring, Mesh, Bus, Hybrid, Network Types-PAN, LAN, MAN, WAN, Network Models: Protocol Layering, the OSI Model, TCP/IP Protocol Suite. **Physical Layer:** Transmission Media – Twisted Pair, Coaxial Cable, Fibre Optics, Radio Transmission, Microwaves and Infrared Transmission, Switching, Multiplexing.	**13**
Unit II	**Data-Link- Layer::** Data Link Layer design issues, Error detection – Single parity checking, Checksum, polynomial codes – CRC, Error correction-Hamming code, Elementary data link protocols, sliding window protocols. Media Access Control (MAC): ALOHA-Pure, slotted, CSMA, CSMA/CD, CSMA/CA, Reservation, Polling, Token Passing.	**13**
Unit III	**Network Layer:** Introduction, Network layer design issues, Network-Layer Services, Network-Layer Performance, IPV4 Addresses, Network Layer Protocols: Internet Protocol (IP), ICMPv4, Mobile IP, Routing algorithms – Flooding, Distance vector routing, Hierarchical routing, Link state routing, Congestion, control algorithms – Leaky bucket, token bucket algorithm, admission control, Hop by Hop choke packets , Next Generation IP: IPv6 Addressing.	**13**

Unit IV	**Transport Layer:** Introduction, Transport-Layer Protocols, and Transport-Layer Protocols: User Datagram Protocol (UDP), Transmission Control Protocol: TCP Services, TCP Features, Segment, A TCP Connection. **Application Layer:** WWW, E-MAIL, Domain Name System (DNS), Quality of Service: Flow Control To Improves QoS, Integrated Services Wireless Technologies - Wi-Fi, Bluetooth, Cellular Networks (4G, 5G), Network Configuration -IP Configuration, Ping, Basic Network Security -Firewalls, VPNs, Basic Threats and Counter measures.	

Reference Books:

1. Behrouz A. Forouzan, "Data Communications and Networking with TCPIP Protocol Suite| 6th Edition", McGraw HillEducation,2022.
2. Andrew S. Tanenbaum, Nick Feamster, David J. Wetherall, Computer Networks, 6th Edition, Prentice Hall, 2022.
3. Larry L. Peterson and Bruce S. Davie, "Computer Networks A System Approach", 5th Edition,MKP,2012.
4. James F. Kurose , Keith W. Ross, " Computer Networking, A Top-Down Approach", 5thEdition, Pearson, 2012.

Web Resources:

1. https://www.geeksforgeeks.org/computer-network-tutorials/
2. https://codescracker.com/networking/

Year	**II**	**Course Code: 2E3XXXE01T**	**Credits**	**03**
Sem.	**III**	**Course Title: Multimedia & Animation**	**Hours**	**42**
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA: 03 hours.**	
Course Outcomes		1. **Write a well-designed, interactive Web site with respect to current standards and practices.** 2. **Demonstrate in-depth knowledge of an industry-standard multimedia development tool and its associated scripting language.** 3. **Determine the appropriate use of interactive versus standalone Web applications.**		

Unit No.	Course Content	Hours
Unit I	**Web Design:** Origins and evolution of HTML, Basic syntax, Basic text markup, Images, Lists, Tables, Forms, Frame, Overview and features of HTML5. CSS: Introduction, Levels of style sheets, Style specification formats, Selector forms, Property value forms, Font properties, List properties, Color, Alignment of text, The and tags; Overview and features of CSS3. JavaScript: Object orientation and JavaScript; General syntactic characteristics; Primitives, operations, and expressions; Scree output and keyboard input.	**10 hours**
Unit II	**Animation:** What is an Animation? The Start and End States, Interpolation Animations in HTML. All About CSS Animations, Creating a Simple Animation, Detailed Look at the CSS Animation Property, Keyframes Declaring Multiple Animations, Wrap-up. All About CSS Transitions Adding a Transition, Looking at Transitions in Detail, The Longhand Properties, Longhand Properties vs. Shorthand Properties, Working with Multiple Transitions.	**10 hours**
Unit III	**Bootstrap 5:** Introduction to Bootstrap 5 (What is Bootstrap, Advantages, CDN vs Local Setup, Folder Structure), Bootstrap Grid System (Containers, Rows, Columns, Breakpoints, Responsive Layouts), Typography and Utilities (Headings, Display classes, Text utilities, Spacing, Colors, Borders, Shadows), Bootstrap Components (Buttons, Alerts, Badges, Breadcrumbs, Cards, Dropdowns, Modals, Navs and Tabs, Navbar, Pagination, Progress bars, Spinners, Toasts), Forms and Inputs (Form controls, Form layout, Form validation, Input	**10 hours**

	groups, Floating labels), Images and Media (Responsive images, Figures, Embeds, Ratios), Tables and List Groups (Table classes, Striped, Hover, Bordered, Responsive tables, List groups with content), Utilities and Helpers (Flex utilities, Position, Display, Sizing, Visibility, Overflow, Z-index).	
Unit IV	**HTML5 – CANVAS:** The Rendering Context, Browser Support, HTML5 Canvas Examples, Canvas - Drawing Rectangles, Canvas - Drawing Paths Canvas - Drawing Lines, Canvas - Drawing Bezier Curves, Canvas – Drawing Quadratic Curves, Canvas - Using Images, Canvas - Create Gradients, **HTML5 - Styles and Colors, Canvas** - Text and Fonts, Canvas - Pattern and Shadow, Canvas - Save and Restore States, Canvas - Translation, Canvas - Rotation, Canvas - Scaling, Canvas - Transforms, HTML5 Canvas - Composition, Canvas – Animations	**12 hours**
Reference Resources for Learning		
References Books: **1.** The Complete Reference HTML and CSS, 5th Edition, Thomas A Powell, 2017. **2.** Animation in HTML, CSS, and JavaScript, Kirupa Chinnathambi, Createspace Independent Pub, 2013. 1st Edition **3.** https://www.w3.org/Style/CSS/current-work#CSS3 **4.** http://bedford-computing.co.uk/learning/cascading-style-sheets-css/		

Year	**II**	**Course Code: 2E3XXXE02T**	**Credits**	**03**
Sem.	**III**	**Course Title: Open Source Tools**	**Hours**	**42**
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA: 03 hours.**	
Course Outcomes		1. **Write a well-designed, interactive Web site with respect to current standards and practices.** 2. **Demonstrate in-depth knowledge of an industry-standard multimedia development tool and its associated scripting language.** 3. **Determine the appropriate use of interactive versus standalone Web applications.**		

Unit No.	Course Content	Hours
Unit I	**Open Source Software:** Introduction to Open sources, Need of Open Sources, Open Source –Principles, Standard Requirements, Advantages of Open Sources –Free Software – FOSS. Licenses – GPL, LGPL, Copyrights, Patents, Contracts & Licenses and Related Issues Application of Open Sources. Open Source Operating Systems : FEDORA, UBUNTU	**10 hours**
Unit II	**Programming Tools & Techniques**. Usage of design Tools like Argo UML, Basics of Git and GitHub (version control concepts, git init, add, commit, push, pull, clone, simple branching); Introduction to Linux Command Line (basic shell commands for developers); Open Source IDE - Visual Studio Code (installation, extensions, basic usage); Package Managers in Linux- Introduction to apt &dpkg (for ubuntu os) and rpm for fedora; Code Formatting Tools	**10 hours**
Unit III	**Introduction to Docker:** Understanding Docker and containerization, benefits of containers over virtual machines; Overview of Docker images and containers, their use in development and production; Basic Docker commands (docker run, docker ps, docker stop, docker rm); Introduction to Docker files for creating custom images; Using Docker Hub for accessing public images; Overview of Docker Compose for managing multi-container applications; Introduction to Docker networking and volumes for data persistence.	**10 hours**
Unit IV	Introduction to Open Source Movement; Case studies of Apache HTTP Server, Postman Tool: Introduction to Postman;	**12 hours**

	Setting up Postman and creating workspaces; Sending GET, POST, PUT, DELETE requests; Using Postman Collections to organize requests; BSD Unix, Mozilla Firefox, Wikipedia (MediaWiki), Joomla CMS, GNU Compiler Collection (GCC), and Libre-Office; Comparative analysis of success factors, community governance, licensing, and future trends in open source.	

Reference Resources for Learning

References Books:

1. “Open Source Licensing: Software Freedom and Intellectual Property Law" by Lawrence Rosen
2. "Producing Open Source Software" by Karl Fogel
3. "Pro Git" by Scott Chacon and Ben Straub
4. "Linux Command Line and Shell Scripting Bible" by Richard Blum and Christine Bresnahan
5. "Visual Studio Code: End-to-End Editing and Debugging Tools for Web Developers" by Bruce Johnson
6. "Docker: Up & Running" by Karl Matthias and Sean P. Kane
7. "The Docker Book: Containerization is the New Virtualization" by James Turnbull
8. "The Art of Community" by Jono Bacon
9. "Postman Essentials" by Arvind M. and Krishna Sankar

Year	**II**	**Course Code: 2E3XXXE03T**	**Credits**	**03**
Sem.	**III**	**Course Title: Digital Marketing**	**Hours**	**42**
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA: 03 hours.**	
Course Outcomes		1. Define and implement digital marketing strategies using key channels like SEO, SEM, social media, content, and email marketing. 2. Apply SEO and SEM techniques to optimize websites and manage paid search campaigns. 3. Design and execute social media and content marketing campaigns to enhance engagement and conversions. 4. Create effective email marketing campaigns and utilize affiliate marketing and digital tools for business growth.		
Unit No.		**Course Content**		**Hours**
Unit I		**Introduction to Digital Marketing:** Definition, importance, and scope of digital marketing. Differences between digital and traditional marketing. Digital marketing channels: SEO, SEM, social media marketing, content marketing, and email marketing. Creating digital marketing strategies, identifying target audiences, setting KPIs. Website fundamentals and the importance of online presence.		**10 hours**
Unit II		**SEO and SEM (Search Engine Optimization and Search Engine Marketing)** SEO principles: keyword research, on-page SEO (title tags, meta descriptions), off-page SEO (backlinking). Introduction to SEM: paid search advertising, Google Ads, campaign targeting. Analyzing performance using Google Analytics and Google Search Console. Improving website rankings and driving traffic.		**10 hours**
Unit III		**Social Media Marketing and Content Marketing:** Creating social media strategies for Facebook, Instagram, LinkedIn, Twitter. Planning posts, audience engagement, and social media management tools (Buffer, Hootsuite). Content marketing: blogging, video marketing, podcasting. Creating content calendars, distribution, and optimization. Measuring success through engagement, reach, and conversions.		**10 hours**
Unit IV		**Email Marketing, Affiliate Marketing, and Tools** Email marketing basics: creating campaigns, managing mailing lists, designing newsletters. Key performance metrics: open		**12 hours**

	rates, click-through rates. Affiliate marketing basics. Digital marketing tools: HubSpot, MailChimp, Canva, Google Analytics. Career opportunities in digital marketing and professional networking strategies.	

Reference Resources for Learning

References Books:

1. "Digital Marketing: A Practical Approach" by R. K. Suri and Rajeev Kumar
2. "Digital Marketing for Beginners" by Shubham Agarwal
3. "The Art of Digital Marketing" by Pradeep Gohil
4. "Internet Marketing: A Practical Approach" by S. R. S. Sharma and Sumati Reddy

BCA 4th semester

Year	II	**Course Code: 2E4XXXM010T**	**Credits**	04
Sem.	IV	**Course Title: Python Programming**	**Hours**	52
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA:03hrs.**	
Course Outcomes		**1. Explain the basic concepts of Python Programming.** **2. Identify the methods to create and manipulate lists, tuples, and dictionaries.** **3. Discover the Data analysis with python using numpy, pandas and matplotlib** **4. Understand Web Development with Flask Framework and Database Integration**		
Unit No.		**Course Content**		**Hours**
Unit I		**Python Basics:** Identifiers; Keywords; Statements and Expressions; Variables; Operators; Precedence and Association; Data Types; Indentation; Comments; Built- in Functions-Console Input and Console Output, Type Conversions; Python Libraries; Importing Libraries with Examples. Control Flow: Types of Control Flow; Control Flow Statements- if-else, elif, while loop, break, continue statements, for loop Statement; range () and exit () functions. **Exception Handling:** Types of Errors; Exceptions; Exception Handling using try, except and finally.		**13 hours**
Unit II		**Functions, Strings** Functions: Types of Functions; Function Definition- Syntax Function Call, Passing Parameters/arguments, return statement: Default Parameters; Key Word Arguments Recursive Functions; Strings: Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings-Concatenation, Comparison, Slicing and Joining, Traversing; Python String Methods. **Lists, Dictionary:** Lists: Creating Lists; Operations on Lists; Built-in Functions on Lists Implementation of Stacks using Lists; Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries.		**13 hours**

Unit III	**Tuples and Sets:** Creating Tuples; Operations on Tuples; Built-in Functions on Tuples; Tuple Methods; Creating Sets; Operations on Sets; Set Methods. **Data Analysis with Python:** NumPy: Creating and manipulating 1D and 2D arrays, indexing, slicing, reshaping, Pandas: Working with Series and DataFrame objects, loading data from CSV and Excel files, indexing, selecting, filtering, handling missing values (dropna(), fillna()), merging and concatenating datasets, performing groupby operations with aggregation functions, value_counts()). Matplotlib: Creating line plots, bar charts, histograms, and scatter plots	**13 hours**
Unit IV	**Database Interaction with MYSQL, Python Flask** Database Interaction with MySQL: MYSQL and Python: Connecting to MYSQL databases, cursor object, execute(), insert, update and delete, fetchall(), fetchone() select command **Python Flask**: Overview of the Flask framework, including setting up basic applications, its features, Defining routes, view functions, request handling, retrieving data from requests using GET and POST, and returning HTTP string responses and json response using jsonify() to build Rest API, rendering HTML templates with Flask, Using HTML forms, and template syntax and control structures and session and flash message.	**13 hours**
Reference Resources for Learning		
References Books: **1.** Introduction to Python Programming 2019, Gowrishankar S. and Veena A. **2.** Flask made simple - A Beginner-Friendly Guide to Python Web Development with Flask Somnath Mukherjee		

<table>
<tr><td>Year</td><td>II</td><td rowspan="2">Course Code: 2E4XXXM010L
Course Title: Python Programming Lab</td><td>Credits</td><td>02</td></tr>
<tr><td>Sem.</td><td>IV</td><td>Hours</td><td>52</td></tr>
<tr><td colspan="2">Course Pre-requisites, if any:</td><td colspan="3">Knowledge of Programming</td></tr>
<tr><td colspan="2">Formative Assessment Marks:10</td><td colspan="2">Summative Assessment Marks:40</td><td>Duration of ESA: 03hrs.</td></tr>
<tr><td colspan="2"></td><td colspan="3">Part-A
1. Write a python program to find roots of a quadratic equation
2. Write a python program to find sum of n natural numbers
3. Write a python program to print multiplication table
4. Write a python program to check a number is prime or not
5. Write a python program to find factorial of a number using recursive function
6. Write Program to find Square of Number use Exception Handling until the number is valid
7. Write python program to read sentence and find the number of vowels
8. Write a python program uses lists to store book titles allowing users to add books, search for books, and display all books.
9. Write a python to read a sentence and count of frequency of words
10. Write a python program to implement stack operations
Part-B
11. Write a python to read five subject names of BCA and five subject names of BSC and find a) all subject names of Both b) common subjects c) subjects only for BCA using set operations
12. Write a python program Create a 2D NumPy array for customer sales data, compute total spending per customer, find the most purchased product category, and identify the highest spender.
13. Write a python program to Load an employee dataset (CSV), filter employees who joined after 2020, calculate the average salary per department, and find the highest and lowest paying departments.
14. Write a Python program to plot a line chart for monthly temperature data labelling axes, adding a title, customizing colors.
15. Write a Python program to load a dataset, extract relevant columns, and create a bar chart to compare the total revenue of different companies from the dataset.
16. Write a python program to read product details such as product id,</td></tr>
</table>

	name and price and insert into MYSQL Database table product and display all products where price greater than 100 17. Write a python Flask App to create ten static student details list and create an HTML template form which takes input rollno and display the particular student data using GET method using HTML template rendering 18. Write a python Flask App to create customer HTML register form template and store data using POST method into MYSQL table customer. 19. Write a python Flask App to create Login HTML Form template check the credentials in MYSQL Database Users table if exist store in a session and redirect it to another home page route otherwise display error message 20. Write a python Flask App to fetch MySQL products table and display it in an html table products.html file include columns such as pid, name, image, price. Where image column should contain path of image file

Year	II	Course Code: 2E4XXXM011T	Credits	04
Sem.	IV	Course Title: Web Technology	Hours	52
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA: 03 hours.	
Course Outcomes		1. Students will be able to create structured web pages using HTML. 2. Understand and apply CSS for responsive design. 3. Write JavaScript programs to enhance interactivity. 4. Work with the DOM for dynamic content manipulation. 5. Build basic React applications and work with APIs.		

Unit No.	Course Content	Hours
Unit I	**Introduction to HTML and CSS** HTML head and body sections. Essential HTML Tags – Headings, paragraphs, links, images, tables, lists, forms, and input fields. Introduction to CSS – Syntax, selectors. Box Model – Margin, padding, border, and content layout. CSS Layout Techniques – Flexbox & Grid. CSS Media Queries– Making websites responsive for different screen sizes. Styling Essentials – Colors, fonts, background properties, and opacity. Navigation Bar & Card Design – Key properties: display, flex, justify-content, align-items, gap, border-radius, box- shadow. Typography – Font styles, sizes, line-height, letter-spacing, text alignment. Styling Forms & Tables – Input styling button styles form validation with CSS	13 hours
Unit II	**Javascript Basics** <script> tag, output using console, document.write() and window alert() object, Input using prompt(), keywords, constants, variables using var and let, datatypes, string templates, operators, conditional control statements, looping statements modules import and export Functions: Defining functions with parameters, returning values, anonymous functions, arrow functions, callback functions, Arrays: creation, accessing array elements, spread operator, Array methods and map(), filter(), reduce(),Object literals and array of objects, object de-structuring and array de-structuring, string methods.	13 hours

Unit III	**Document Object Model (DOM)** Introduction to DOM: Understanding the Document Object Model; Selecting Elements through getElementByID(), getElementByClassName(), querySelector(). Manipulating Elements – Changing text, HTML content innerHTML, and CSS styles dynamically. Event Handling – Click, hover, keypress, and other events Form Validation – Handling user input validation with JavaScript using regular expressions, JSON basics, JSON.stringfy() and parse(), Promises Async and Await Fetch() Function to fetch API, localStorage, sessionStorage	**13 hours**
Unit IV	**Single Page Application and React** Introduction to Single Page Applications (SPA) – Concept of SPAs and why React is used. Introduction to React – Understanding React components, JSX, and Virtual DOM. React State & Props – Managing data in React applications. React Hooks useState() & useEffect() – Handling component state and side effects. Fetching APIs in React – Making API calls using Fetch API, handling JSON data and displaying dynamic content.	**13 hours**
Reference Resources for Learning		
References Books: **1.** "HTML5, JavaScript, and jQuery 24-Hour Trainer" – Steven W. Disbrow. **2.** "Eloquent JavaScript" (3rd Edition) – Marijn Haverbeke. **3.** "Web Design with HTML, CSS, JavaScript, and jQuery" – Jon Duckett. **4.** "Web Technology: A Developer's Perspective" – N.P. Gopalan & J. Akilandeswari.		

<table>
<tr><td>Year</td><td>II</td><td rowspan="2">Course Code: 2E4XXXM011L
Course Title: Web Technology Lab</td><td>Credits</td><td>02</td></tr>
<tr><td>Sem.</td><td>IV</td><td>Hours</td><td>52</td></tr>
<tr><td colspan="2">Course Pre-requisites, if any:</td><td colspan="3">Knowledge of Programming</td></tr>
<tr><td colspan="2">Formative Assessment Marks:10</td><td colspan="2">Summative Assessment Marks:40</td><td>Duration of ESA: 03hrs.</td></tr>
<tr><td colspan="2"></td><td colspan="3">Part-A
1. Write HTML – Create a personal portfolio webpage with a bio, profile image, and social media links.
2. Write HTML – Design a student registration form with input fields for name, email, gender, and course selection.
3. Write HTML – Develop a timetable using tables and include rowspan and colspan for better structuring.
4. Write HTML and CSS – Build a responsive navigation bar using Flexbox with hover effects.
5. Write HTML and CSS – Design a modern login form with styled input fields, buttons, and shadows.
6. Write HTML and CSS – Create a responsive card layout using CSS Grid and Flexbox for a product showcase.
7. Write JavaScript – Implement a simple interest calculator that takes input and displays the result dynamically.
8. Write JavaScript – Create a to-do list application where users can add, remove, and mark tasks as completed.
9. Write JavaScript – Filter and search a list of students dynamically using filter() and map().
10. Write JavaScript – Build a temperature converter that switches between Celsius and Fahrenheit.

Part-B
11. Write JavaScript – Implement a shopping cart where adding/removing items updates the total price dynamically.
12. Write JavaScript – Change background color dynamically on a button click using JavaScript.
13. Write JavaScript – Validate a registration form for correct username format, password strength, and empty fields.
14. Write JavaScript – Display a real-time digital clock using setInterval().
15. Write JavaScript – Implement a countdown timer that stops at zero using setTimeout().</td></tr>
</table>

	16. Experiment: Counter App Create a React app that uses useState() to store a count value. Add buttons to increment and decrement the count. Pass the count as props to a child component and display it there. 17. Experiment: User Card with Props Create a React app that stores user information such as name, email, and a loggedIn flag in the parent component. Pass this data as props to a UserCard component. Conditionally render a greeting like "Welcome, Guest!" or display the user's name based on the loggedIn status. 18. Experiment: To-Do List Create a React app that uses useState() to manage a list of to-do items. Allow users to add and remove tasks. Use conditional rendering to display a message like "No tasks available" when the list is empty. 19. Experiment: Fake User Registration App Create a React app with a form that includes fields like name, email, and password. On form submission, send the data using a POST request to a fake online API that simulates user registration. Display a success message with the submitted data, or show error messages if something goes wrong. (API: **https://jsonplaceholder.typicode.com/users)** 20. Experiment: Country Info Finder Create a React app that allows users to enter a country name in an input field. When the form is submitted, use fetch() and useEffect() to call a public API and retrieve real-time information like capital, population, region, and flag. Pass the data to a child component and display it. Use conditional rendering to show loading and error messages. Use Api: **https://restcountries.com/v3.1/name/india**

Year	II	**Course Code: 2E4XXXM012T**	**Credits**	04
Sem.	IV	**Course Title: Operating System**	**Hours**	52
Course Pre-requisites, if any		**NA**		
Formative Assessment Marks: 20		**Summative Assessment Marks:80**	**Duration of ESA:03hrs.**	
Course Outcomes		**1. To analyze the memory management and its allocation policies** **2. To understand synchronization and deadlock conditions and their possible solutions** **3. To discuss the storage management policies with respect to different storage**		

Unit No.	Course Content	Hours
Unit I	**Introduction:** Computer System Organization, Architecture, Structure, Operations, Process Management, Memory Management, Storage Management. Operating System Structures: Services, System Calls, Types, Operating System Structure, System Boot. Processes: Process Concept, Scheduling, Operations, Inter-process Communication. Multithreaded Programming: Multithreading Models	**13 hours**
Unit II	**Process Synchronization:** The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization Examples. Process Scheduling: Criteria, Scheduling Algorithms, Multi-Processor Scheduling, Real-time CPU Scheduling. Deadlocks: System model, Characterization, Methods for handling deadlocks, Deadlock Prevention, Avoidance, Detection and Recovery from deadlock.	**13 hours**
Unit III	**Memory Management Strategies:** Background, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory Management: Demand Paging; Copy-on-Write, Page Replacement; Allocation of Frames,	**13 hours**
Unit IV	**File System:** File Concept, Access Methods, Directory and Disk Structure, Protection. File-system Implementation: Structure, File- System and Directory Implementation, Allocation Methods, Free Space Management. Mass-Storage Structure: Overview, Disk Scheduling, Disk Management.	**13 hours**

Reference Resources for Learning
References Books: **1.** Abraham Silberschatz, Peter Baer Galvin, Greg Gagne; “Operating Systems Concepts”, 9th Edition, 201 6 India, Wiley. **2.** William Stallings, "Operating Systems and Design Principles", Pearson, thEdition,2018

Year	II	Course Code: 2E3XXXE04T	Credits	03
Sem.	IV	Course Title: Machine Learning	Hours	42
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA:03hrs.	
Course Outcomes		1. Understand ML concepts and data preprocessing techniques. 2. Implement and evaluate regression and classification models. 3. Apply clustering and dimensionality reduction techniques.		

Unit No.	Course Content	Hours
Unit I	**Introduction to Machine Learning:** Fundamental concepts of Machine Learning (ML), applications, supervised, unsupervised, and reinforcement learning. Real-world ML applications in finance, healthcare, and recommendation systems. Data preprocessing techniques, handling missing values, feature scaling (normalization and standardization), encoding categorical variables. Python ecosystem for ML, including NumPy, Pandas, and Matplotlib. Hands-on exercises in data manipulation, visualization, exploratory data analysis (EDA) using Pandas and Matplotlib	10 hours
Unit II	**Regression Analysis & Evaluation Metrics** Regression models, simple linear regression, multiple linear regression, polynomial regression. Mathematical intuition and implementation using Scikit-Learn. Evaluation metrics including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R-squared. Practical exercises in implementing regression models on real datasets such as predicting house prices or stock prices. Fine tuning hyper parameters and understanding model performance	10 hours
Unit III	**Classification Algorithms & Model Optimization** Classification algorithms, Logistic Regression, k-Nearest Neighbors (k-NN), Decision Trees, Support Vector Machines (SVM). Training and testing classification models, performance evaluation using confusion matrices, precision-recall curves, and F1-scores. Practical	10 hours

	implementation of binary and multiclass classification tasks, such as spam email detection or digit classification using the MNIST dataset.	
Unit IV	**Clustering & Dimensionality Reduction** Unsupervised learning techniques, K-Means Clustering, Hierarchical Clustering. Implementing clustering on datasets to identify patterns and groupings without predefined labels. Principal Component Analysis (PCA), t-SNE for dimensionality reduction while retaining essential features. Hands-on exercises in customer segmentation and visualizing high-dimensional data. Limitations of clustering algorithms and effective cluster interpretation	**12 hours**
Reference Resources for Learning		

References Books:

1. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow – Aurélien Géron, 3rd Edition, O'Reilly Media.
2. Python Machine Learning – Sebastian Raschka, Vahid Mirjalili, 3rd Edition, Packt Publishing.
3. Machine Learning for Absolute Beginners – Oliver Theobald, 2nd Edition, Scatterplot Press.
4. Introduction to Machine Learning with Python – Andreas C. Müller, Sarah Guido, 1st Edition, O'Reilly Media.
5. Applied Machine Learning with Python – Andrea Giussani, 1st Edition, Apress.
6. Machine Learning Yearning – Andrew Ng, Self-Published.

Year	II	Course Code: 2E3XXXE05T	Credits	03
Sem.	IV	Course Title: Internet of Things	Hours	42
Course Pre-requisites, if any		NA		
Formative Assessment Marks: 20		Summative Assessment Marks:80	Duration of ESA: 03 hours.	
Course Outcomes		1. Understand IoT architecture and hardware components. 2. Program microcontroller devices using Python. 3. Integrate devices with cloud platforms and visualize data. 4. Address IoT security concerns and manage data.		

Unit No.	Course Content	Hours
Unit I	**Introduction to IoT and Hardware Basics** Definition and characteristics of IoT, evolution and applications of IoT in smart homes, health, agriculture, and industries. IoT architecture: perception, network, and application layers. Overview of microcontrollers and microprocessors. Introduction to sensors and actuators. Basic setup of Arduino and Raspberry Pi. Interfacing sensors (temperature, light, motion) with Arduino.	10 hours
Unit II	**Programming IoT Devices using Python** Installing Python on Raspberry Pi, basic Python programming review. GPIO pin configuration. Reading data from sensors using Python. Controlling devices (LED, buzzer, relay) using Python scripts. Simple Python projects like smart light and motion alarm	10 hours
Unit III	**IoT Communication Protocols and Cloud Integration** IoT communication protocols: HTTP, MQTT, Bluetooth, Zigbee, Wi-Fi. Implementing MQTT protocol using Python (paho-mqtt). Connecting devices to open IoT platforms like ThingSpeak or Adafruit IO. Sending sensor data to cloud. Visualizing data using Python and ThingSpeak.	10 hours
Unit IV	**IoT Security and Data Management** Security issues in IoT devices. Authentication, encryption basics. Secure communication protocols (TLS, HTTPS). Introduction to JSON and data serialization. Storing and retrieving IoT data. Managing data using SQLite with Python.	10 hours

Reference Resources for Learning

References Books:

1. "Internet of Things with Python" by Gaston C. Hillar, Packt Publishing.
2. "Mastering IoT with Python" by Colin Dow, Packt Publishing.
3. "Getting Started with Raspberry Pi" by Matt Richardson and Shawn Wallace, Make:
4. "Python Programming for Arduino" by Pratik Desai, Packt Publishing.
5. "Hands-On Internet of Things with MQTT" by Tim Pulver, Packt Publishing.

Year	II	Course Code: 2E3XXXE06T	Credits	03
Sem	IV	Course Title: UNIX AND SHELL PROGRAMMING	Hours	42
Formative Assessment Marks: 20	Summative Assessment Marks:80			Duration of ESA: 03 hours
Course Outcomes	1. **Understands shell concept in Linux environment and practice basic commands of UNIX/Linux.** **2. Study of advance commands and filters of UNIX/Linux.** **3. Using vi editor for writing shell scripts.**			
Unit	**Course Content**			**Hours**
Unit I	**Introduction and Basic Commands**: Brief history, Salient features of UNIX system, The UNIX architecture, Internal and External Commands along with options: cal, bc, date, echo, cat, who, cp, mv, rm, tty, sty, uname, pwd, passwd., man, Script commands Flexibility of CommandUsage. **File System and File Handling Commands:** more, lp, wc, spell, ispell, directory commands: mkdir, cd,rmdir, ls command & its options:(-x,-f,-a,-d,-l), cmp, comm, diff: Compressing files, gzip, gunzip, and unzip commands, changing file permissions: chmod, changing group permissions: chgrp, changing file ownership: chown, umask, locating files: find, path.			**10 Hours**
Unit II	**Standard I/O, Redirection Pipes & Filters:** Standard I/O, Redirection, pipe & pipeline, filter, tee command, terminal & trash files, pr, head, tail, cut, paste, sort, uniq, tr, grep, **The Process:** Process Basics, ps: Process Status& its options, mechanism of Process Creation, Internal and External Commands, Running jobs in Background, nice: Job Execution with Low Priority, Killing Processes with Signals, Job Control, fg and bg commands at and batch: Execute. Later, cron: Running Jobs Periodically, time: Timing Processes.			**10 Hours**
Unit III	**The vi Editor**: vi Basic, Input Mode - Entering and Replacing Text, Saving Text and Quitting - The ex-Mode, Navigation, Editing Text, Undoing Last Editing Instructions, (u and U) Repeating the Last Command (.), Searching for a Pattern (I and ?), Substitution Search and Replace (:s) **Shell:** The Shell's Interpretive Cycle, Pattern Matching - The Wild - cards, Escaping and quoting, redirection: The three Standard Files, /dev/null and /dev/tty: Two Special Files, Command Substitution.			**10 Hours**
Unit IV	**Shell programming**: Shell scripts, shell variables, read, using command line arguments, exit &exit status of command,			**12 Hours**

	arithmetic operators, relational operators, increment & decrement operators, assignment operators, the logical operators && and ||,if conditional, test and [ ] to evaluate expressions, case conditional, expr: computation & string handling, while looping, for looping, break & continue statements.	

Reference Resources for Learning

Reference Books

1. Sumitabha Das, Unix Concepts and Applications, McGraw Hill publications.
2. Venkateshmurthy, Introduction to UNIX and Shell Programming, Pearson Education.
3. Glass, UNIX for Programmers and Users, 3/e Pearson Education.
4. Kernighan, The UNIX Programming Environment.
5. Kochan, UNIX Shell Programming Pearson.

Evaluation Scheme for Lab Examination

Assessment Criteria		**Marks**
Activity–1 from Part A	Write up on the activity/task	07
	Demonstration of the activity/ task	08
Activity-2 from Part B	Write up on the activity/task	07
	Demonstration of the activity/ task	08
Viva based on Lab Activities		10
Total		**40**