

**PUNYASHLOK AHILYADEVI HOLKAR
SOLAPUR UNIVERSITY, SOLAPUR**



Name of the Faculty: Science and Technology

CHOICE BASED CREDIT SYSTEM

Syllabus: Computer Application

Name of the Course: BCA - III(Semester –V and VI)

(Syllabus to be implemented w.e.f. June 2024)

Punyashlok Ahilyadevi Holkar Solapur University,
Solapur
Faculty of Science and Technology
Choice Based Credit System
(CBCS) (w.e.f.2024-25) Revised
Structure for BCA - III

Subject/ Core Course	Name and Type of the Paper		No. of Papers/ Practica l	Hrs./ Week			Total Mark sper Paper	UA	CA	Credits
	Type	Name		L	T	P				
Class:	BCA - III Semester – V									
Ability Enhancement Course	(AECC)	English (Business English)	Paper II Part A	4	--	--	50	40	10	2.0
Core	DSC 1E	Data Communication and Networking	Paper IX	4	--	--	100	80	20	4.0
	DSC 1F	Data Warehousing and Data Mining	Paper X	4	--	--	100	80	20	4.0
	DSC 1G	.NET Core	Paper XI	4	--	--	100	80	20	4.0
	DSE 1	A-Data Science using Python or B-Kotlin Programming or C-ReactJS	Paper XII	4	--	--	100	80	20	4.0
Skill Enhancement Course	SEC 2	Advanced Java		4	--	--	100	80	20	4.0
Total Theory Semester-V				24			550	440	110	22
Class:	BCA - III Semester – VI									
Ability Enhancement Course	(AECC)	English (Business English)	Paper II Part B	4	--	--	50	40	10	2.0
Core	DSC 2E	Network Security	Paper XII	4	--	--	100	80	20	4.0
	DSC 2F	Data Visualization using PowerBI	Paper XIII	4	--	--	100	80	20	4.0
	DSC 2G	ASP.Net Core	Paper XIV	4	--	--	100	80	20	4.0

	DSE 2	A- Linux and Shell Programming or B- Mobile Application Development using Kotlin or C- Internet of Things	Paper XV	4	--	--	100	80	20	4.0
Total Theory Semester-VI				20			450	360	90	18
Practical's on	Project			--	--	5	100	80	20	4.0
	DSC 1F and 2F			--	---	5	100	80	20	4.0
	DSC 1G and 2G			--	--	5	100	80	20	4.0
	DSE 1A/B/C and 2A/B/C			--	--	5	100	80	20	4.0
Total (practical's)						20	400	320	80	16
Grand Total				42	--	20	1400	1120	280	56

% Theory paper: (Core Computer Science Course)

\$The students can choose MOOCs/ NPTEL/SWAYAM/Pathshala/Add-on / Skill based courses of university/college initiated courses of same credits.

\$ These courses are not compulsory, but after completion of these courses students get additional credits on their

Marklist.

\$SEC Courses initiated by colleges

Abbreviations:

L: Lectures T: Tutorials P: Practicals

UA : University Assessment CA : College Assessment CC: Core Course

AEC : Ability Enhancement Course DSE : Discipline Specific Elective Paper SEC : Skill

Enhancement Course

BCA-III (CBCS)							
Sem:		V					
Paper Type:		DSC 1E		Paper No:		Paper IX	
Paper Name:		Data Communication and Networking					
Credit:		04		Theory:		4 Hrs./week	
Marks:	UA:	80	CA:	20	Total:	100	

Course Objectives:

1. To understand the structure of Data Communications System and its components.
2. Be familiarize with different network terminologies.

Course Outcomes: Upon successful completion of this course, students will be able to-

1. Familiarize with contemporary issues in network technologies.
2. Know the layered model approach explained in OSI and TCP/IP network models
3. Identify different types of network devices and their functions within a network.
4. Know the Basic routing mechanisms, IP addressing scheme and internet working concepts.
5. Familiarize with IP and TCP Internet protocols.
6. Understand major concepts involved in design of WAN, LAN and wireless networks.
7. Know the basics of network configuration and maintenance.
8. Know the fundamentals of network security issues.

Unit 1: Introduction to Computer Networks

[20]

Network Definition, Network Topologies, Network Classifications, Network Protocol, Layered Network Architecture, Overview of ISO-OSI Reference Model, Overview of TCP/IP Protocol Suite.

Data Communication Fundamentals and Techniques:

Signals-Analog and Digital Signal, Data-Rate Limits, Digital to Digital Line Encoding Schemes, Pulse Code Modulation, Parallel and Serial Transmission, Digital to Analog Modulation, Multiplexing Techniques- FDM, TDM, Transmission Media, Switching: Circuit Switching, Message Switching, Packet Switching

Unit 2: Data Link Layer Functions and Protocols**[10]**

Design issues, Error Detection and Error Correction Techniques, Data-Link Control- Framing and Flow Control, Error Recovery Protocols-Stop and Wait ARQ, Go-Back-N ARQ, Point to Point Protocol on Internet.

Unit 3: Multiple Access Protocol and Network Layer**[10]**

Design issues, CSMA/CD Protocols, Ethernet LANS; Connecting LAN and Back -Bone Networks- Repeaters, Hubs, Switches, Bridges, Router and Gateways, Networks Layer Functions and Protocols, Routing, Routing Algorithms, Network Layer Protocol of Internet - IP Protocol, Internet Control Protocols.

Unit 4: Transport, Session, Presentation and Application Layer Protocol**[20]**

Transport Services- Error and Flow Control, Connection Establishment and Connection Release, Flow Control & Buffering, TCP/IP protocol suite, Concept of- TCP, UDP, IP, FTP, DNS, Telnet, SMTP, POP, HTTP, WWW, ARP, RARP.

Reference Books:

- B. A. Forouzan: Data Communications and Networking, Fourth edition, THM Publishing Company Ltd 2007.
- S. Tanenbaum: Computer Networks, Fourth edition, PHI Pvt. Ltd 2002

BCA-III (CBCS)							
Sem:	V						
Paper Type:	DSC 1F			Paper No:	Paper XV		
Paper Name:	Data Warehousing and Data Mining						
Credit:	04			Theory:	4 Hrs./week		
Marks:	UA:	80	CA:	20	Total:	100	

Course Objectives:

1. To understand the principles of Data warehousing and Data Mining.
2. To understand the Architecture of a Data Mining system.
3. To perform classification, association, and prediction of data.

Course Outcomes: The students should be able to:

1. Identify data mining problems and implement the data warehouse.
2. Write association rules for a given data pattern.
3. Choose classification and clustering solutions.

Unit 1: Introduction to Data Warehouse and Data Mining [15]

Differences between Operational Database Systems and Data Warehouses, Data Warehouse Architecture, Data Warehouse Components, A Multidimensional Data Model, Schemas, Data Warehouse Implementation, Data cube Technology, OLAP operations, Data mining query language, Data Mining:- What is data mining, Evolution, KDD, What kind of data, Architecture, data mining views, Data Mining Functionalities, Issues in Data Mining.

Unit 2: Data Preprocessing and Association Rule mining [15]

Data Preprocessing: An Overview, Extract, Transform, Load (ETL) Processes, Data Cleaning, Data Integration, Data Transformation and Data Discretization, Data Reduction, Frequent Patterns,

Associations, and Correlations: Market Basket Analysis, Frequent Itemsets, Closed Itemsets, and Association Rules, Frequent Itemset Mining Methods-Apriori Algorithm: Finding Frequent Itemsets, Generating Association Rules from Frequent Itemsets, Improving the Efficiency of Apriori, A Pattern-Growth Approach for Mining Frequent Itemsets, Mining Multilevel and multidimensional Association Rules, Constraint-Based Frequent Pattern Mining

Unit 3: Supervised Learning Technique [15]

supervised and unsupervised learning, What Is Classification? What is regression, difference

between classification and regressing, General Approach to Classification, Issues regarding Classification and Predication, Binary and Multiclass Classification, Types of classifications, Classification by Decision tree induction, Bayesian Classification, Classification by Back propagation, Logistic regression, k-Nearest-Neighbor Classifiers, SVM, Introducing Ensemble Methods-Bagging, Boosting, AdaBoost, Random Forests, Other classification methods, Prediction: regression. Model Evaluation and Selection-Metrics for Evaluating Classifier Performance, Cross-Validation, underfitting and overfittin.

Unit 4: Unsupervised Learning Technique and Applications [15]

Clustering: What is Cluster Analysis, Types of data in Cluster Analysis, A Categorization of Major Clustering Methods., Partitioning Methods, Hierarchical Methods, Density-Based Methods, Model-Based Clustering Methods: Statistical Approach, Neural Network Approach, Outlier Analysis, Applications and Trends in Data Mining: Data Mining Applications, Data Mining for Financial Data Analysis, Data Mining for Retail and Telecommunication Industries, Data Mining in Science and Engineering, Data Mining for Intrusion Detection and Prevention, Data Mining and Recommender Systems, Spatial Data Mining. Text Data Mining, Multimedia Data Mining, Web Data Mining, Privacy, Security, and Social Impacts of Data Mining, Data Mining and Intelligent Query Answering, Trends in Data Mining.

Reference Books:

- Data Mining – Concepts and Techniques – Jiawei Han, Micheline Kamber, Morgan Kaufmann Publishers, Elsevier, 2nd Edition, 2006.
- Introduction to Data Mining, Pang – Ning Tan, Vipin Kumar, Michael Steinbach, Pearson Education.
- Data Warehouse Fundamentals, Pualraj Ponnaiah, Wiley Student Edition.

BCA-III (CBCS)							
Sem:		V					
Paper Type:		DSC 1G		Paper No:		Paper XI	
Paper Name:		Dot NET Core					
Credit:		04		Theory:		4 Hrs./week	
Marks:	UA:	80	CA:	20	Total:	100	

Course Objectives:

1. To understand how to design, implement, test, debug, and document programs that use basic data types and computation, simple I/O, conditional and control structures, string handling and functions in C#.
2. To understand the importance of Classes and objects along with constructors, Arrays and Vectors in C#.
3. Discuss the principles of inheritance, interface and demonstrate through problem analysis assignments how they relate to the design of methods, abstract classes and interfaces and packages in C#.
4. To understand importance of Multi-threading and different exception handling mechanisms in C#.
5. To understand basic idea about how to design GUI base windows application using C#.

Course Outcomes: Students will be able to:

1. Gain proficiency in designing, implementing, testing, debugging, and documenting programs utilizing fundamental data types, computation, basic I/O, conditional and control structures, string manipulation, and functions within the C# programming language.
2. Appreciate the significance of classes, objects, constructors, arrays, and vectors within the C# framework.
3. Explore the principles of inheritance and interfaces, demonstrating their application through problem analysis assignments and their relevance to method design, abstract classes, interfaces, and packages in C#.
4. Recognize the importance of multi-threading and various exception handling mechanisms in C# programming.
5. Acquire a foundational understanding of designing GUI-based Windows applications using C#.

Unit 1: Introduction to C#

[20]

Understanding .NET- The .NET Framework, .NET Core, Download and install C# Development Environments - Visual Studio, Visual Studio Code, building console apps using Visual Studio 2022 and Building console apps using Visual Studio Code, C# Basics- Variables and Data Types, Reference and Value Types-Nullable types, Elvis operator, Null coalescing operator, Boxing and unboxing, Keywords, Initialization, Type Inference, Console Input and Output., Operators, Operator precedence, Type conversion, C# statements- Branching, Jumping, Looping, Complex data types- Enums, Arrays, Tuples.

Unit 2: Object Oriented Programming

[20]

Classes and object-Declaration, Access modifiers, Data, Methods, Method parameters, Constructors, Deconstruct, Method overloading, Properties, Local and global variable and methods, Static classes, methods and members, nested classes, Indexers, Partial types and methods, Structs and Records, Inheritance- Base and derived classes, advantages, Types. Constructors in inheritance. Abstract classes, sealed class, Interfaces - Defining and implementing, Default interface methods, Interface inheritance, .NET interfaces, Polymorphism- Virtual methods. Method overriding, operator overloading, Abstract methods, Sealed types.

Unit 3: Threading, Exception and Resource Management

[12]

Exception- about exception, Exceptions Hierarchy, Throwing and Catching Exception, The try-finally Construct, IDisposable and the "using" Statement, Advantages of Using Exceptions, inbuilt exception, custom exception, Threading-about threading, Thread Name, Thread Priority, and Thread State, Foreground and background threads in C#, Multithreading - An Overview, The Thread Class, ThreadPool Threads, Collections- Generic collections, Concurrent collections, Specialized collections, Performance considerations, Resource Management- Finalizers, Garbage Collection, IDisposable, The using statement, Serialization-Attributes, JSON serialization, Binary serialization, XML serialization

Unit 4: Delegate, Event and LINQ

[8] Delegates-Multicast delegates, generic delegates,

Action<T>, Predicate<T>, Func<T>, Lambdas-Expression and statement lambdas, Parameters, Return type, Captures, Events- Defining, Raising, Standard and custom events, LINQ- Enabling features, LINQ expression, LINQ pattern, Joins, Aggregations, Basic of Windows application.

Reference Books:

- Nagel, Karli Watson, Jay Glynn, Morgan Skinner, Bill Evjen.
- Inside C# - Microsoft Press by Tom Archer, Andrew Whitechapel.
- Programming Microsoft Visual C# 2005 - The Language (Microsoft Press) by Donis Marshal

BCA-III (CBCS)							
Sem:	V						
Paper Type:	DSE 1A			Paper No:	Paper XI		
Paper Name:	Data Science using Python						
Credit:	04			Theory:	4 Hrs./week		
Marks:	UA:	80	CA:	20	Total:	100	

Course Outcomes:

1. Know basic data types in Python.
2. Know operators, and how to clean and merge datasets.
3. Know pandas library, the main methods for DataFrames.
4. Know how to import data in Python.
5. Know how to work in Spyder.

Unit-1: Introduction to Data Science: Evolution of Data Science – Data Science life cycle, Data Science Roles – Stages in a Data Science Project – Applications of Data Science in various fields – Data Security Issues.

Data Collection Strategies – Introduction to Collection of Data, Primary and Secondary Data, Methods of Collecting Primary Data, Methods of Secondary Data, Statistical Errors, Rounding off Data.

Data Pre-Processing Overview – Data Cleaning – Data Integration and Transformation – Data Reduction – Data Discretization, Outlier analysis, Testing and Training

Model design and Development: Model Evaluation using Visualization – Residual Plot – Distribution Plot –, Measures for In-sample Evaluation – Prediction and Decision Making. Generalization Error – Out-of-Sample Evaluation Metrics – Cross Validation – Over fitting – Under Fitting and Model Selection – Prediction by using Ridge Regression – Testing Multiple Parameters by using Grid Search.

Unit-2: Introduction to NumPy: Arrays and Vectorized Computation- The NumPy ndarray- Creating ndarrays- Data Types for ndarrays- Arithmetic with NumPy Arrays- Basic Indexing and Slicing - Boolean Indexing-Transposing Arrays and Swapping Axes.

Universal Functions: Fast Element-Wise Array Functions- Mathematical and Statistical

Methods-Sorting Unique and Other Set Logic.

Introduction to pandas: Series, DataFrame, Creating Data Frame from an Excel, .csv file, python Dictionary, python List and Tuples, Operations on Data Frames: Dropping Entries Indexing, Selection, and Filtering- Function Application and Mapping- Sorting and Ranking. Summarizing and Computing Descriptive Statistics- Unique Values, Value Counts, and Membership. Reading and Writing Data in Text Format.

Data visualization using Matplotlib and Seaborn: Basic plot-Line, Bar, scatter, subplot, Statistics plots-Box, Histogram, errorbar and pie, 3D plots-scatter, surface, triangular, Heatmap, Density Plots, CatPlot (Categorical Plot), Joint Distribution Plot.

Books Recommended:

1. Python Data Analytics, Fabio Nelli, Apress
2. Wes McKinney, “Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython”, O’Reilly, 2nd Edition, 2018. Python for data science for dummies 2nd Edition, John Paul Mueller, Luca Massaron, and Wiley
3. Jake VanderPlas, “Python Data Science Handbook: Essential Tools for Working with Data”, O’Reilly, 2017.
4. Programming through Python, M. T. Savaliya, R. K. Maurya, G. M. Magar, STAREDU Solutions
5. Introducing Data Science: Big Data, Machine Learning, and More, Using Python Tools, Davy Cielen, Arno D.B. Meysman, et al., Mining

BCA-III (CBCS)							
Sem:	V						
Paper Type:	DSE-1 B			Paper No:	Paper XI		
Paper Name:	Kotlin Programming						
Credit:	04			Theory:	4 Hrs./week		
Marks:	UA:	80	CA:	20	Total:	100	

Course Objective: -

Students will try to learn:

1. The Kotlin programming language.
2. Familiarize with Kotlin syntax, features, and basic programming concepts.
3. Explore language features such as type inference, null safety, and extension functions, and understand their importance in Kotlin programming.
4. Provide a solid foundation in object-oriented programming (OOP) concepts within the context of Kotlin which covers classes, objects, inheritance, interfaces, and other OOP principles, and demonstrate their implementation in Kotlin.
5. To introduce functional programming concepts in Kotlin, including higher-order functions, lambda expressions, and functional manipulation of collections and understand how functional programming enhances code readability, maintainability, and reusability.
6. Equip the skills to write Kotlin code for simple applications.
7. Lay the foundation for further exploration and application of Kotlin in software development.

Course Outcomes: -

Students will be able to:

8. Gain a comprehensive understanding of the Kotlin programming language.
9. Acquire familiarity with Kotlin syntax, features, and fundamental programming concepts.
10. Explore and grasp the significance of language features such as type inference, null safety, and extension functions in Kotlin programming.
11. Develop a strong foundation in object-oriented programming (OOP) principles within the Kotlin context, covering classes, objects, inheritance, interfaces, and other related concepts.

12. Introduce functional programming concepts in Kotlin, including higher-order functions, lambda expressions, and functional manipulation of collections, while understanding their impact on code readability, maintainability, and reusability.
13. Develop the skills necessary to write Kotlin code for simple applications.
14. Establish a groundwork for further exploration and utilization of Kotlin in software development endeavors.

Unit 1: - Introduction to Kotlin

(15)

Introduction to Kotlin, Download and Install JDK, Download and Install IntelliJ Idea, Creating New Kotlin Project, Creating Kotlin File, main Function, Run Kotlin Application, Single Line and MultiLine Comment, Data Type- basic data types (Int, String, Boolean, etc.), Variable using var and val, type inference, Operators and expressions: Arithmetical Operators, Comparison Operators ,Equality and Inequality Operators ,Logical Operators ,Increment and Decrement Operators ,Augmented Assignments etc., Control flow: if Expression, if else Expression ,when Expression, for Loop ,while Loop ,do while Loop ,Break and Continue, Getting User Input using readLine, Getting User Input using Scanner, String ,String Concatenation ,String Literals - Escaped String ,String Literals - Raw String ,String Templates

Unit 2: - Functions and Lambdas

(15)

Defining functions: function syntax, parameters, return types., Function without Parameters, Function with Parameters, Function with Default Argument, Function with Named Argument, Higher-order functions: passing functions as arguments, returning functions from functions. Lambda expressions: syntax, usage, capturing variables., Anonymous Function, Null Safety- Nullable and non-nullable types in Kotlin., Safe calls (?), Elvis operator (?:), and the !! operator., Handling nullability: safe casts (as?), let, run, apply, also functions

Unit 3: - Collections

(10)

Introduction to Kotlin collections: array lists, sets, maps or Dictionary, Basic operations on collections: iterating, adding, removing elements., Functional programming with collections: map, filter, reduce.

Unit 4: - Object-Oriented Programming

(20)

Classes and objects: class declaration, properties, methods., Constructors: primary constructors, secondary constructors., Getter and Setter, Inheritance: superclass, subclass, Inheritance with Constructor, overriding methods., Properties and Function, Super, Visibility Modifiers, Abstract Class and Method, Interfaces: defining interfaces, implementing interfaces in classes., Data Class, Object Destructuring, Exception Handling: try-catch block, throw keyword, finally block., Kotlin's approach to checked and unchecked exceptions., File I/O:-Reading from and writing to files., Working with text files: reading lines, writing to files., Handling file exceptions.

References:

1. "Kotlin Programming: The Big Nerd Ranch Guide" by Josh Skeen and David Greenhalgh
2. "Kotlin in Action" by Dmitry Jemerov and Svetlana Isakova Online Kotlin documentation and tutorials.

BCA-III (CBCS)							
Sem:	V						
Paper Type:	DSE-1 C			Paper No:	Paper XI		
Paper Name:	ReactJS						
Credit:	04			Theory:	4 Hrs./week		
Marks:	UA:	80		CA:	20		Total: 100

Course Objectives:

1. Understand Single Page React application is different than traditional web development frameworks.
2. Develop an application from scratch.
3. Understand the benefits of unidirectional data flow.

Course Outcomes:

1. To understand basic fundamentals of Front-end Technologies
2. To apply the advanced concepts of ReactJS and WebAPI
3. To develop ReactJS applications.

Unit 1: Introduction to ReactJS

[20]

Introduction, Workflow, Scope, Pros and Cons, Difference between JS and JSX, React Components overview, Child Components, JSX expressions,

Building Blocks of ReactJS: JSX, Components, State and Props, Conditional Rendering, Why JSX, Advantages of JSX, Expressions in JSX, Implementation of JSX, Creating a react component with jsx

Environment Setups: Node setup, How to use NPM, Npm and Setting Environment for ReactJS projects, How to create package.json and purpose, IDE for ReactJS, ReactJS browser plugins overview.

Components: Types of components, Functional component vs Class Component, Converting Functional Components to Class Components, Component Life Cycles and its different methods.

Unit 2: Conditional Rendering and List

[10]

if-else Statement, logical andand operator, operators, Preventing Component from Rendering, Switch case operator

List and Keys: react key prop, map function to iterate the List, References, use Refs, Create

Refs, access Refs, Event Binding types: Bind () method, Arrow function

Props and State: What is a state, use and role of the state, what are props, Props validation, Passing data between multiple components, Managing Component State

Unit 3: Handling Events and Forms [10]

Lists of Form components, Setup Controlled and Uncontrolled form components, Control Input elements, Form Submission and Validation, how to set default values on all formats of Input elements, Form validations, writing Styles, Animations overview, Event, Event Binding, Event Handlers, Common React Events, Key Events, Event Pooling, Synthetic Event.

Unit 4: Routing and State Management [20]

Introduction to React Router, History of Router, Single Page Application Overview, configure React Router, Load the router library, Navigating between Routes, Route Parameters and Nested Routes, Dynamic Routing, Nesting Routes, Invalid URL, Handle Conditional statement in JSX

State Management: Local State vs. Global State, State Lift-Up, Context API for Global State

Redux: Introduction to Redux, Redux Architecture- Actions, Reducers, and Store, Provider Component, Dispatchers, View Controllers, Connecting React with Redux

Hooks: Introduction to Hooks, The useState hook, useEffect hook, Custom hook, useRef hook, useMemo hook, The useContext hook, The useReducer hook, Another Hooks.

Reference Books:

- Introduction to React By Cory Gackenheimer, Apress
- React and React Native: A complete hands-on guide to modern web and mobile development with React.js By Adam Boduch, Roy Derks
- React 16 Essentials: A fast-paced, hands-on guide to designing and building scalable and maintainable web apps with React 16 By Artemij Fedosejev, Ada

BCA-III (CBCS)							
Sem:		V					
Paper Type:		SEC-2			Paper No:		Paper XI
Paper Name:		Advanced Java					
Credit:		04			Theory:		4 Hrs./week
Marks:	UA:	80	CA:	20	Total:	100	

Course Objectives:

1. To learn GUI programming using swing Technology
2. To study database programming using Java.
3. To study web development concept using Servlet and JSP
4. To learn socket programming concept using Networking.

Course Outcomes: Upon successful completion of this course, students will be able to-

1. Design, create, build, and debug Java applications.
2. Explore Integrated Development Environment (IDE).

Unit 1: Networking and Database

[15]

Networking: Basics, networking classes and interfaces, using java.net package, doing **Working with databases:** Steps for Connecting to databases, Types of Drivers, Handling Exceptions, Creating and Using Statement Objects, Using Statements to Insert, Update, Delete Data into a Database, Using the ResultSet Class, Data navigation, Prepared Statements, Callable Statements.

Unit 2: Introducing Servlet

[18]

Advantages of Servlet over CGI, Features of Servlet, Introducing Servlet API, Javax.servlet package, Javax.servlet.http package, Introducing Servlet, Advantages of Servlet over CGI, Features of Servlet, Servlet life Cycle, Init(), Service(), Destroy(), Working with GenericServlet and HttpServlet, RequestDispatcher interface, Include() and forward() ,Use of RequestDispatcher, Session in Servlet, Introducing session, Session tracking mechanism, Cookies, Advantages and disadvantages, use of cookies, Hidden form field, Advantages and

disadvantages, use of Hidden form field, URL rewritten, disadvantages, use of URL rewritten, HttpSession, Advantages and disadvantages, use of URL HttpSession

Unit 3: Java Server Pages

[12]

Introduction, Advantages of JSP over Servlet, JSP architecture , JSP life cycle, Implicit objects in JSP- request, response, out, page, page Context, application, session, config, exception, JSP tag elements- Declarative, Declaration, scriptlet, expression, action. Java Bean- Advantages and Disadvantages, Use Bean tag- setProperty and getProperty, Bean In Jsp, JSTL core tag: General purpose tag, conditional tag, networking tag, JSTL SQL tags, JSTL formatting tags, JSTL xml tags, Custom tag: empty tag, body content tag, iteration tag, simple tag , Introducing internationalization and Java: local class, ResourceBundle class, Handling different Error and Exceptions.

Unit 4: Hibernate and Spring

[15]

Introduction Hibernate (HB), Architecture of HB, Application of HB, HB with annotation, HB web application, Inheritance mapping: Table per Hierarchy, (TPH), TPH using annotation, Table Per, Concrete (TPC), TPC using annotation, Table Per Subclass (TPS), TPS using annotation, Collection mapping: Mapping list, one to many by list, one to many by bag, one to many by set, one to many by map.

Spring: Introduction to spring, Spring modules, Spring application, Dependency injection: constructor Injection (CI), CI dependant object, CI with collection, CI with map, CI inheriting bean, Spring JDBC: JDBC template, PreparedStatement, ResultSetExtractor, RowMapper, NamedParameter, Simple JDBC template, Spring with Hibernate.

Reference books:

- Java the complete Reference by Herbert Schildt
- Java Servlet Programming by Jasan Hunter
- Java Server Programming by Bayross and Shah
- JDBC, Servlet and JSP Black Book- Santosh Kumar K.
- Hibernate- Sharanam Shah and Vaishali Shah
- Spring Persistence with Hibernate- Paul Tepper Fisher, Brian D Murphy

BCA-III (CBCS)							
Sem:	VI						
Paper Type:	DSC 2E			Paper No:	Paper XII		
Paper Name:	Network Security						
Credit:	04			Theory:	4 Hrs./week		
Marks:	UA:	80		CA:	20		Total: 100

Course Objective: -

Students will try to learn:

1. Introduce students to the fundamental concepts of security, including its necessity, approaches, principles, and various types of security attacks.
2. Provide an overview of cryptography, including plaintext and ciphertext, substitution and transposition techniques, encryption and decryption processes.
3. Introduce symmetric and asymmetric key cryptography algorithms, steganography, key range, key size, and potential types of cryptographic attacks.
4. Investigate Internet security protocols such as Secure Socket Layer/Transport Layer Security (SSL/TLS), Secure Electronic Transaction (SET), and email security protocols like PGP and S/MIME.
5. Cover user authentication basics, including passwords, smart cards, and biometrics, as well as authentication protocols like Kerberos as well as discuss network security principles, firewalls, types of firewalls, and IP security mechanisms to safeguard networks against unauthorized access and malicious activities.

Unit 1: - Fundamental Security Concepts

(10)

Security Concepts: Introduction, The need for security, Security approaches, Principles of security, Types of Security attacks – Active and Passive, Security services, Security Mechanisms, A model for Network Security, Access Control Mechanisms: Access Matrix, ACL and capabilities, Access Control Models,.

Unit 2: - Cryptography Concepts and Techniques

(20)

Introduction to Cryptography, plain text and cipher text, substitution techniques, transposition techniques, encryption and decryption, symmetric and asymmetric key cryptography, steganography, key range and key size, possible types of attacks, Symmetric Key Cryptographic Algorithms: Algorithm Types and Modes, An overview of Symmetric Key Cryptography, DES, International Data Encryption Algorithm (IDEA), RC5, Blowfish, AES, Asymmetric Key Cryptography: Brief History of Asymmetric Key

Cryptography, An overview of Asymmetric Key Cryptography, The RSA Algorithm, Symmetric and Asymmetric Key Cryptography Together

Unit 3: - Digital Signatures and Internet Security Protocols (20)

Digital Signatures: Introduction, Message digests, MD5, SHA-512, MAC, HMAC, Knapsack Algorithm, Elliptic curve Technology, ELGamal Algorithm. , Internet Security Protocols: Secure Socket Layer/TLS, Secure Electronic Transaction, SSL versus SET, E-mail Security- PGP, S/MIME.,

Unit 4: - User Authentication and Network Security (10)

User Authentication and Kerberos: Authentication basics, Passwords, use of smart cards, Biometrics, Kerberos., Network Security: Firewalls, types of firewalls, IP Security, Intrusion : Intruders, Audit Records, Intrusion Detection, honey pots.

Reference Books:

1. Cryptography and Network Security by Atul Kahate, Tata McGraw-Hill
2. Cryptography and Network Security by Behrouz A. Forouzan, Debdeep Mukhopadhyay, Special Indian Edition, Tata McGraw-Hill.
3. Network Security Essentials: Applications and Standards by William Stallings, Pearson Education.
4. Fundamentals of Computer Security Technology: Edward Amoroso, Prentice-Hall.
5. Cryptography and Data Security: Dorothy E. Denning, Addison-Wesley.
6. Cryptography -Theory and Practice: Douglas R. Stinson, CRC Press.
7. Building Internet Firewalls: D. Brent Chapman and Elizabeth D. Zwicky, O'Reilly and Associates.

BCA-III (CBCS)							
Sem:	VI						
Paper Type:	DSC 2F			Paper No:	Paper XII		
Paper Name:	Data Visualization using PowerBI						
Credit:	04			Theory:	4 Hrs./week		
Marks:	UA:	80		CA:	20		Total: 100

Course Objective: -

Students will try to learn:

1. To gain proficiency in navigating the Power BI interface, understanding its components, and utilizing its basic features for data visualization.
2. To learn how to import, clean, and transform data from various sources into formats suitable for visualization in Power BI.
3. To understand the importance of data modeling techniques such as creating relationships, measures, and calculated columns for effective analysis.
4. To master the creation of various types of visualizations, including bar charts, line charts, scatter plots, and maps, using Power BI's intuitive tools.
5. To explore customization options to enhance visual appeal and effectively convey insights to stakeholders.
6. To learn to design interactive dashboards by combining multiple visualizations, slicers, filters, and drill-down functionalities.
7. To understand principles of dashboard layout, organization, and storytelling to create engaging and actionable presentations of data.
8. To understand the process of sharing Power BI reports and dashboards securely with colleagues and stakeholders.

Unit 1: - Introduction to Power BI

(10)

Overview of Business Intelligence, BI Uses and Users, Various BI Tools, Why Power BI, Introduction to Power BI, Features of Power BI, Power BI Components, Building Blocks of Power BI, Architecture of Power BI, Power BI Desktop Installation, Loading and Transforming dataset: Data Sources-File Sources, Databases, Azure, Other Sources, Loading Data-Web Pages, CSV Files, Text Files, XML Files, Excel, Microsoft Access Databases, SQL Server and other databases, Refreshing Data.

Unit 2: - Creating a Data Model

(10)

Data Modeling in the Power BI Desktop Environment-The Power BI Desktop Data View, Data Models, Managing Power BI Desktop Data- Manipulating Tables, Manipulating Columns, Power BI Desktop Data Types, Formatting Power BI Desktop, Data Currency Formats, Preparing Data for Dashboards, Categorize Data, Apply a Summarization, Define Sort by Columns, Sorting Data in Power BI Desktop Tables, Adding Hierarchies, Designing a Power BI Desktop Data Model- Data View and Relationship View, Creating and Deleting Relationships Manually and Automatically

Unit 3: - Transforming Datasets

(20)

Editing Data After a Data Load, Transforming Data Before Loading, Dataset Shaping- Renaming Columns, Reordering Columns, Removing Columns, Merging Columns, Duplicating Columns, Splitting Columns, Removing Records, Removing Duplicate Records, Sorting Data, Reversing the Row Order, Filtering Data-Selecting Specific Values, Finding Elements in the Filter List, Filtering Text Ranges, Filtering Numeric Ranges, Filtering Date and Time Ranges, Data Cleansing: Viewing a Full Record, Changing Data Type , Detecting Data Types, Replacing Values, Transforming Column Contents, Filling Down, Using the First Row As Headers, Grouping Records, Extending Data, Appending Data, Merging Data- Adding Data, Aggregating Data During a Merge Operation, Extending the Data Model with Calculated Columns, Creating Custom Columns, Index Columns, Types of Join- Joining on Multiple Columns, Preparing Datasets for Joins, Correct and Incorrect Joins, Examining Joined Data, The Expand and Aggregate Buttons.

Unit 4: - Power Query Editor and Data Visualizations

(20)

Power Query Editor - What is DAX, Different type of DAX functions-Aggregate functions, Date functions, Logical functions, Math functions, String functions, Trigonometric functions and other functions. Adding Measures to the Data Model Basic Aggregations in Measures, Using Multiple Measures, Cross-Table Measures, More Advanced Aggregations, Filtering Data in Measures, Analyzing Data over Time. Data Visualizations Charts in Power BI-Types of charts, Maps in Power BI, Table and Matrix in Power BI, Subtotal and Total in Matrix, Cards and Filters in Power BI, Conditional Formatting, Slicers in Power BI- slicers, adding a Slicer, Applying Slicers, clearing a Slicer, deleting a Slicer, modifying a Slicer, Formatting Slicers-Slicer Orientation, Modifying the Outline, Adjusting Selection Controls, Setting the Exact Size and X and Y coordinates of a Slicer, Slicer Header, Slicer Items Designing Power BI Dashboards and Reports Dashboards, reports, Dashboards versus reports, Dashboard design-What is KPI, Whento use KPI, Requirements for KPI, KPI Visualizations, Visual selection, Layout, Navigation pane, Full screen mode, Supporting tiles, Custom date filters, Single- dashboard, Multiple-dashboards, Organizational dashboards, Multiple datasets Dashboard tiles- Tile details and custom links, Images and text boxes, SQL Server Reporting Services. Deploying the Power BI Report Server Live

Dashboard pages, Live report pages, Mobile-optimized dashboards Case study – Superstore, IPL Analysis, Product Sales Data Analysis, Marketing Campaign Insights Analysis, Financial Performance Analysis, Loan Application Analysis

Reference Books: -

1. Pro Power BI Desktop-Free interactive data analysis with Microsoft Power BI by AdamAspin, Apress
2. Introducing Microsoft Power BI by Alberto Ferrari and Marco Russo, Microsoft Press
3. Mastering Microsoft Power BI by Brett Powell, Packt BIRMINGHAM– MUMBAI
4. Microsoft Power BI Complete Reference by Devin Knight, Brian Knight, Mitchell Pearson, Manuel Quintana, Brett Powell, Packt
5. Learn Power BI by Greg Deckler, Packt

BCA-III (CBCS)							
Sem:	VI						
Paper Type:	DSC 2G			Paper No:	Paper XII		
Paper Name:	ASP.Net Core						
Credit:	04			Theory:	4 Hrs./week		
Marks:	UA:	80		CA:	20		Total: 100

Course Objective: -

Students will try to learn:

1. Demonstrate the creation of ASP.NET Core MVC Web Applications using .NET 8, covering project file structure, main method, hosting options, and configuration files.
2. Cover Models, Controllers, Views, and Dependency Injection in ASP.NET Core MVC applications.
3. Introduce Entity Framework Core and guide the installation process also explain DbContext in Entity Framework Core and database connection string configuration and database operations.
4. Discuss Transactions, Migration, and Database Seeding in Entity Framework Core.
5. Introduce Partial Views, View Components, and Razor View Engine.
6. Cover Action Results, Routing, Model Binding, HTML Helpers and Tag Helpers in ASP.NET Core MVC.
7. Explain Data Annotations and Model Validations, including custom validations and remote validation.
8. Discuss different methods of State Management, including Cookies and Sessions.

Unit 1: - Introduction to ASP.Net Core MVC

(10)

Overview of Microsoft Web Technologies, Introduction to ASP.NET Core Framework. NET Core Environment Setup, Download and Install Visual Studio 2022, Download and Install .NET Core SDK, Download and Install SQL Server 2022, Download and Install SSMS, Creating ASP.NET Core Web Application using .NET 8, NET Core Project File Structure, NET Core Main Method, NET Core InProcess Hosting, OutOfProcess Hosting, LaunchSettings.json File, AppSettings.json file, Middleware Components, Web Root (wwwroot) Folder, Static Files Middleware, Configuring Default Page, Developer Exception Page Middleware Command Line Interface, Project Templates in ASP.NET Core Application, Introduction to ASP.NET Core MVC Framework, Set up MVC in ASP.NET Core, Models, Controllers and Views in ASP.NET Core MVC, ASP.NET Core Dependency Injection, Creating ASP.NET Core

Application using MVC

Unit 2: - Entity Framework Core (15)

Introduction to Entity Framework Core, How to Install Entity Framework Core, DbContext in Entity Framework Core, Database Connection String in Entity Framework Core, CRUD Operations in Entity Framework Core, Entity States in Entity Framework Core, Data Annotation Attributes in Entity Framework Core- Table Attributes, Column Attributes, Key Attribute, ForeignKey Attribute, Index Attribute, InverseProperty Attribute, NotMapped Attribute, Required Attribute, MaxLength and MinLength Attribute, Database Generated Attribute, TimeStamp Attribute, ConcurrencyCheck Attribute, Relationships in Entity Framework Core- One-to-One Relationships, One-to-Many Relationships, Many-to-Many Relationships, Self-Referencing Relationship, Asynchronous Programming with Entity Framework Core, Disconnected Entities in Entity Framework Core, Stored Procedures in Entity Framework Core, Transactions in Entity Framework Core, Migration in Entity Framework Core, Database Seedd in Entity Framework Core, Entity Framework Core Database First Approach

Unit 3: Model, View, Controller and Routing (20)

ViewData, ViewBag, Strongly Typed View, ViewModel, TempData, Post-Redirect-Get (PRG) Pattern Example, Layout View, Sections in Layout View, ViewStart, ViewImports, Partial Views, Different Ways to Render Partial View, View Components, Razor View Engine and Razor Syntax, How to Install and use Bootstrap in ASP.NET Core MVC, Action Results in ASP.NET Core MVC- Action Results, View Result, Partial View Result, JSON Result, Content Result, File Result, Redirect Results, Status Results, Object Result, EmptyResult , Routing in ASP.NET Core MVC, Custom Routing, Custom Route Constraints in Web Application, Attribute Routing, Attribute Routing using Tokens, Attribute Routing vs Conventional Routing, Model Binding in ASP.NET Core MVC, Model Binding using- FromForm, FromQuery, FromRoute, FromHeader, FromBody, Complex Type, Custom Model Binding in ASP.NET Core MVC

Unit 4: - HTML, Tag Helper, Data Annotation Validation and State management

(15)

HTML Helpers for-TextBox, TextArea, DropDownList, RadioButton, Check Box, ListBox, Password, Hidden, Custom HTML Helper in ASP.NET Core MVC, Creating Form Using HTML Helpers, Different Ways to Generate Links in ASP.NET Core MVC, Tag Helpers for- Image Tag , Environment Tag, Navigation Menus, Form Tag, Partial Tag, Creating Custom Tag Helper, View Component Tag Helper, Cache Tag Helper, Data Annotations, Model Validations, Data Annotation Attributes- Custom Data Annotation, Remote Validation, Blacklist and Whitelist Checks using Data Annotation, Displaying and Formatting Attributes, Real-Time Examples of Data Annotations in ASP.NET Core MVC, Cookies, Encrypt Cookies, Persistent vs Non-Persistent Cookies, Sessions, In-Memory vs Distributed Sessions, Differences Between Cookies and Sessions, Upload File, Restrict Uploaded File Size, Restrict Uploaded File Type, Save Uploaded file to Database, Display Images, Delete Images, Upload Multiple Files, Export

Data to Excel File, Import Excel Data to Database, Generate PDF, Generate Password Protected PDF, Convert HTML to PDF, Send Email with Attachment

Reference Books:

1. Pro ASP.NET Core MVC by ADAM FREEMAN, Apress
2. ASP.NET MVC with Entity Framework and CSS by Lee Naylor, Apress
3. ASP.NET MVC Core 2.0 Cookbook by Engin Polat & Stephanen Belkheraz, Packt Publishing;
4. ASP.NET Core in Action by Andrew Lock
5. ASP.NET Core Application Development: Building an application in four sprints (Developer Reference) by James Chambers, David Paquette & Simon Timms, Microsoft Press
6. Pro ASP.NET Core MVC by ADAM FREEMAN, Springer Nature
7. Professional ASPNET MVC 5 by Galloway Jon. Matson David. Wilson Brad. Allen K Scott, wiley

BCA-III (CBCS)					
Sem:	VI				
Paper Type:	DSE 2-A		Paper No:	Paper XII	
Paper Name:	Linux and Shell Programming				
Credit:	04		Theory:	4 Hrs./week	
Marks:	UA:	80	CA:	20	Total: 100

Course Objective: -

Students will try to learn:

1. To introduce Basic Linux general purpose Commands
2. To learn different editor
3. To learn shell script concepts.
4. To learn file management and permission advance commands.
5. To learn awk, grap, perl scripts.

Unit 1: - Introduction of Linux

(10)

History of Linux, Architecture of Linux system & features, Kernel, Shell & its type, Difference between Windows and Linux. Linux Distributions, Working environments: KDE, GNOME, Xface4, Hardware requirement, Installation procedure of Linux, Create partitions, Configuration of X system Users & Groups Management:- Create Users, Create groups, Special groups, Assigning permissions to users and Groups, File and Directory permissions- chmod, chown, chgrp., Linux File System:-Hierarchy of File system, File System parts- Boot Block, Super Block, Inode, Block, Data Block, File types, Devices and Drives in Linux, Mounting devices (CD/DVD, usb, hard drive partition), file system

Unit 2: - Linux Command

(15)

Linux commands File and directory Management Commands:-mkdir, rmdir, cd and pwd, file, ls, cat, more, less, File and Directory Operations: find, cp, mv, rm, ln etc, Printing the files - lpr, lpq, lprm etc., Filter Commands & Editor:- Filters: head, tail , pr, cut, paste, sort, uniq, tr, grep, egrep, fgrep, sed., Communication commands:- mesg, talk, write, wall, mail., Text Editors- vi, vim, Archive and File compression commands,

Unit 3: - Linux System Management and Administration

(15)

Process Management: Shell process, Parent and children, Process status, System process, Multiple jobs in background and foreground, Changing process priority with nice. Listing processes, ps, kill, premature termination of process., Disk management and System Administration:-Disk Partitioning- RAID, LVM

etc., disk related Management Tools- Fdisk, Parted etc. , Boot Loaders- GRUB, LILO, Custom Loaders, System administration – Role of system administrator, identifying administrative tasks & files, Configuration and log files, Chkconfig, Security Enhanced Linux, Installing and removing packages with rpm command, Understanding various Servers:- DHCP, DNS, Squid, Apache, Telnet, FTP, Samba.

Unit 4: - Shell Programming

(20)

Introduction to shell scripting, Writing and executing simple shell scripts, Variables, data types, and operators in shell scripting, Meta characters, Control structure, Loop structure and case statement, Writing and using shell functions, Passing arguments to shell scripts and functions, Returning values from functions, I/O and Redirection, Piping,

Course Outcomes: -

Students will be able to:

1. Identify the basic Linux general purpose commands.
2. Apply and change the ownership and file permissions using advance Linux commands.
3. Use the awk, grep, perl scripts.
4. Implement shell scripts.
5. Apply basic of administrative task.

Reference Books :

1. Official Red Hat Linux Users guide by Redhat, Wiley Dreamtech India
2. Beginning Linux Programming by Neil Mathew & Richard Stones, Wiley Dreamtech India
3. Red Hat Linux Bible by Cristopher Negus, Wiley Dreamtech India
4. UNIX Shell Programming by Yeswant Kanethkar, BPB
5. Shell Scripting: Expert Recipes for Linux, Bash, and More" by Steve Parker
6. Classic Shell Scripting" by Arnold Robbins and Nelson H.F. Beebe
7. Learning the bash Shell: Unix Shell Programming" by Cameron Newham and Bill Rosenblatt

BCA-III (CBCS)							
Sem:		VI					
Paper Type:		DSE 2 B		Paper No:		Paper XVI	
Paper Name:		Mobile Application Development using Kotlin					
Credit:		04		Theory:		4 Hrs./week	
Marks:	UA:	80	CA:	20	Total:	100	

Course Objective: -

Students will try to learn:

1. Student will understand Set up of Android Studio and understand a development environment for Android app development using Kotlin.
2. Students will able to Create interactive and user-friendly UIs by designing layouts, using XML, and incorporating UI elements such as buttons, text fields, and lists.
3. Students will able to understand user interaction by handling touch events, button clicks, and using dialogs to provide feedback and information.
4. Students will able to Develop multi-screen apps by understanding the concepts of activities, intents, and fragments for effective navigation
5. Student understands data passing between screens using intents, bundles, and fragment communication.
6. Students will learn about data storage options in Android, including SharedPreferences for simple data storage and SQLite databases for more complex data management.
7. Student will Understand the process of preparing and deploying an Android app to the Google Play Store.

Course Outcomes

Students will be able to:

8. Understand the setup process of Android Studio and establish a development environment for Android app development using Kotlin.
9. Create interactive and user-friendly user interfaces (UIs) by designing layouts using XM and incorporate various UI elements such as buttons, text fields, and lists to enhance the user experience.
10. Implement responsive UI elements that enhance user engagement and interaction.
11. Implement effective navigation between screens to provide seamless user experiences within the app.
12. Understand and implement data passing between screens using intents, bundles, and fragment communication.

13. Implement data storage solutions that suit the requirements of the app and ensure data integrity and security.
14. Learn the steps involved in deploying the app to the Google Play Store and ensure compliance with store guidelines and policies.

Unit 1: - Introduction to Android App Development (20)

Overview of Android app development and Android Architecture, Setting up Android Studio., Creating a simple "Hello World" app., Understanding the Android project structure., User Interface (UI) Design and Layouts, UI Design Principles-Material Design guidelines, UI best practices, Layouts and Views and view groups-Introduction to Android layouts, Creating layouts in XML, Designing Interactive views-Buttons, text fields, labels, list, checkbox, spinner, radio and radio group, Handling user interaction, Building Complex UIs-Lists and RecyclerView, Custom UI components

Unit 2: - User Interaction and Navigation (15)

Handling User Input-Responding to button clicks and touch events, Toast messages and dialogs, android components activity, service, content provider, broadcast receiver, Navigating Between components, Creating multiple screens in your app, Fragment-Based UI, Introduction to Fragments, Building flexible UIs with fragments, Passing Data Between Screens, Sending data with Intents, Fragment communication

Unit 3: - Data Storage and Retrieval (15)

Introduction to Data Storage, Overview of data storage options in Android, Choosing the right storage method, Shared Preferences, Using Shared Preferences for simple data storage, Saving and retrieving data, SQLite Database, Introduction to SQLite databases in Android, Performing database operations, Integrating Data Storage, Incorporating data storage into your app, Best practices for data handling

Unit 4: - App Deployment and Finalization (10)

App Deployment, Preparing the app for the Google Play Store, Generating a signed APK, App Finalization, App optimization and performance, Gathering user feedback and making improvements, App Presentation, Final app project presentations, Sharing experiences and challenges, Creating SMS sending application, email application, google map application

Reference Books: -

1. "Android App Development" by Reto Meier
2. "Kotlin for Android App Development" by Peter Sommerhoff

3. "Android User Interface Design: Turning Ideas and Sketches into Beautifully Designed Apps" by Ian G. Clifton
4. "Android Programming for Beginners" by John Horton
5. "Android Fragments" by Dave MacLean
6. "Android Programming: The Big Nerd Ranch Guide" by Bill Phillips and Chris Stewart
7. "Android SQLite Essentials" by Sunny Kumar Aditya

BCA-III (CBCS)							
Sem:		VI					
Paper Type:		DSE 2C			Paper No:		Paper XII
Paper Name:		Internet of Things					
Credit:		04			Theory:		4 Hrs./week
Marks:	UA:	80	CA:	20	Total:	100	

Course Objectives:

1. To understand the concept of IoT and its applications.
2. To reduce import dependency on IoT components and promote indigenization.
3. To Understand the the fundamentals of IoT with its architecture, and protocols.
4. It also covers the overview and programming of two widely used IoT platforms Arduino and Raspberry Pi.

Course Outcome: Students should be able to

1. Demonstrate the architecture and functioning of IoT systems including the sensors and microcontrollers with their interfacing and software needs considering application areas.
2. Diagnose the various IoT protocols with detailing of their elements and overall functioning within IoT systems for efficient communication.
3. Design an IoT system to take the benefit of the Clouds for computing and storage considering security issues.
4. Leverage the benefits of IoT technologies for automating the various real-life challenges in various application areas.
5. Develop the software components of the IoT system using Arduino/Raspberry Pi Programming.

Unit 1: Introduction

[20]

What is IoT, why important, Elements of an IoT ecosystem, Technology drivers, Business drivers, Trends and implications, Application areas of IoT, Characteristics of IoT, Things in IoT, IoT stack, Enabling technologies, IoT challenges, IoT levels, IoT and cyber-physical system, IoT and WSN, Overview of Governance, Privacy and Security Issues.

IOT ARCHITECTURE: IoT Open source architecture (OIC)- OIC Architecture and Design principles- IoT Devices and deployment models- IoTivity : An Open source IoT stack - Overview-

IoTivity stack architecture- Resource model and Abstraction.

Sensors, Microcontrollers, and Their Interfacing: Sensor interfacing, Types of sensors, Controlling sensors, Microcontrollers, ARM.

Unit 2: Protocols **[10]**

Protocol Standardization for IoT – Efforts – M2M and WSN Protocols – SCADA and RFID Protocols – Issues with IoT Standardization – Unified Data Standards – Protocols – IEEE802.15.4–BACNet Protocol– Modbus – KNX – Zigbee– Network layer – APS layer – Security.

Unit 3: Cloud and Applications **[10]**

IoT and cloud, Fog computing, Security in cloud, Case study.

Application Building with IoT: Various applications of IoT: Food, Healthcare, Lavatory maintenance, Water quality, Warehouse, Retail, Driver Assistance, and Collision impact, IoT applications for industry: Future Factory Concepts, Brownfield IoT, Smart Objects, Smart Applications. Study of existing IoT platforms /middleware, IoT- A, Hydra etc.

Unit 7: Arduino and Raspberry Pi **[20]**

Arduino: Architecture, Programming and Application Raspberry Pi : Architecture, Programming and Application.

IoT Security: Various security issues and needs, architecture, requirements, challenges and algorithms.

Reference Books:

- Internet of Things, Vasudevan, Nagrajan and Sundaram, Wiley India
- IoT Fundamentals, David Hince et al, Cisco Press
- 21 IoT Experiments, Yashavant Kanetkar, Shrirang Korde, BPB
- IoT Based Projects, Rajesh Singh et al, BPB
- Internet of Things with ARDUINO and BOLT, Ashwin Pajankar, BPB

Credits: Practical's – (4)

Total Lectures: 60 Hrs.

Contact Hrs. (Pr):5

University Evaluation: 80 Marks Internal Evaluation: 20 Marks

Project (Practical-IV)

Instructions:

1. Team size for major project not exceed than two students.
2. Real time and live project followed by Presentation and Viva-Voce.

Credits: Practical's – (4)

Total Lectures: 60 Hrs.

Contact Hrs. (Pr):5

University Evaluation: 80 Marks

Internal Evaluation: 20 Marks

DSC1F (Tools: Weka or Python)

Practical based on DSC 1E (Data warehousing and Mining)

1. What options are available on main panel? Describe and create the arff and csv file format.
2. Load iris dataset. How many instances are this dataset have? How many attributes? What is the range of possible values of the attribute petallength?
3. Explore the dataset using various visualization tools available in Weka.
4. Analyze basic statistics of the dataset such as mean, median, standard deviation, etc.
5. Identify any missing values and outliers in the dataset.
6. Preprocess the dataset by handling missing values, outliers, and noise.
7. Perform attribute selection to identify the most relevant features.
8. Normalize, discretize or standardize numerical attributes.
9. Load the weather.nominal dataset. Use the following filters in weka.
 - a. unsupervised.instance.RemoveWithValues to remove all instances in which the humidity attribute has the value high.
 - b. Convert numeric value to nominal
 - c. Convert nominal to string
 - d. Discretizes data
10. Load the iris dataset. Use the following filter of weka.
 - a. Add noise to last column (i.e. Class).
 - b. Randomize the data
 - c. Normalize the data
 - d. Reorder the data
11. Implement and evaluate different classification algorithms available in Weka (e.g., Decision Trees, Naive Bayes, k-Nearest Neighbors).
12. Train the models using the training set and evaluate their performance on the testing set using various metrics like accuracy, precision, recall, and F1-score.
13. Compare the performance of different classifiers and identify the most suitable one for the dataset.
14. Apply regression algorithms available in Weka (e.g., Linear Regression, Polynomial Regression).
15. Train the regression models using the training set and evaluate their performance on the testing set using metrics like Mean Squared Error (MSE) and R-squared.
16. Implement and evaluate clustering algorithms available in Weka (e.g., k-Means, Hierarchical Clustering).
17. Explore different clustering techniques and their impact on clustering quality.

18. Visualize the clusters using Weka's visualization tools and analyze their characteristics.
19. Apply association rule mining algorithms (e.g., Apriori) to identify interesting patterns in the dataset.
20. Adjust parameters such as minimum support and confidence to control the quality of the discovered rules.
21. Interpret and analyze the discovered rules to gain insights into the dataset.
22. Implement feature selection techniques to identify the most informative features in the dataset.
23. Evaluate the impact of feature selection and dimensionality reduction on the performance of classification or clustering algorithms.
24. Implement ensemble learning techniques (e.g., Random Forest, AdaBoost) using Weka's ensemble classifiers.
25. Compare the performance of ensemble methods with individual classifiers and analyze the benefits of ensemble learning.

Practical based on DSC 1G (Dot NET Core)

1. Create a simple console application that prints "Hello, World!" to the console.
2. Compile and run the application using the .NET Core CLI or Visual Studio.
3. Perform basic arithmetic operations (addition, subtraction, multiplication, division) on numeric variables.
4. Use string interpolation or concatenation to display variable values.
5. Implement conditional statements (if-else, switch-case) to control program flow based on different conditions.
6. Use loops (for, while, do-while) to iterate over arrays, collections, or sequences of data.
7. Create nested loops and conditional statements for more complex control flow logic.
8. Declare and initialize arrays of different data types.
9. Access array elements using index notation and perform array manipulation operations (sorting, searching, etc.).
10. Define and call methods with different access modifiers (public, private, protected).
11. Pass parameters to methods and return values from methods.
12. Overload methods with different parameter types and number of parameters.
13. Create classes and objects to represent real-world entities.
14. Create class library for checking Odd number, even number, prime number, Armstrong number etc.
15. Implement encapsulation, inheritance, and polymorphism concepts in C#.
16. Use constructors, properties, and methods to define the behavior of objects.
17. Implement try-catch blocks to handle exceptions and prevent application crashes.
18. Throw custom exceptions to handle specific error conditions.
19. Create custom exception and use them.
20. Use finally blocks to execute cleanup code regardless of whether an exception is thrown.
21. Read from and write to text files using StreamReader and StreamWriter classes.

22. Implement file input/output operations such as reading, writing, appending, and deleting files.
23. Handle file exceptions and ensure proper resource management using `IDisposable` interface.
24. Use LINQ queries to perform filtering, sorting, grouping, and aggregation operations on collections.
25. Create and manage multiple threads using the `Thread` class or `ThreadPool`.

Practical based on DSE 1 B (Kotlin Programming)

1. Compile and run the program using the Kotlin compiler or an Integrated Development Environment (IDE) like IntelliJ IDEA.
2. Declare variables of different data types (`Int`, `Float`, `Double`, `String`, `Boolean`).
3. Perform basic arithmetic operations (addition, subtraction, multiplication, division) on numeric variables.
4. Implement conditional statements (`if-else`, `when`) to control program flow based on different conditions.
5. Use loop structures (`for`, `while`, `do-while`) to iterate over arrays, lists, or ranges of values.
6. Create nested loops and conditional statements for more complex control flow logic.
7. Define and call functions with different parameters and return types.
8. Implement higher-order functions that take other functions as parameters or return functions as results.
9. Use lambda expressions to define anonymous functions and pass them as arguments to other functions.
10. Create and manipulate collections such as lists, sets, maps, and arrays in Kotlin.
11. Perform common operations on collections, such as iteration, filtering, mapping, and folding.
12. Use Kotlin's standard library functions (e.g., `filter`, `map`, `reduce`) to process collections efficiently.
13. Define classes and objects to represent real-world entities in Kotlin.
14. Implement inheritance, polymorphism, and encapsulation principles in Kotlin classes.
15. Use data classes and object declarations to create simple data structures and singleton objects.
16. Understand Kotlin's null safety features and how to handle nullable types using safe calls (`?.`), null checks, and the elvis operator (`?:`).
17. Use the safe cast operator (`as?`) and smart casts to safely cast nullable types to non-nullable types.
18. Create extension functions and extension properties to add new functionality to existing classes without modifying their source code.
19. Overload operators such as plus (`+`), minus (`-`), times (`*`), and equals (`==`) to define custom behavior for user-defined types.
20. Read from and write to files using Kotlin's standard library functions for file I/O.
21. Handle exceptions and errors using `try-catch` blocks and the `throw` keyword to gracefully handle unexpected conditions in Kotlin programs.
22. WAP which will demonstrate use of abstract class.
23. WAP to demonstrate interface and Multiple interfaces in kotlin.
24. WAP to implement following data classes-

- a. equals (), b. Hash Code () c. To String () d. Copy ().
25. WAP to which demonstrate all exception handler

Practical Assignments on DSE 1C: (ReactJS)

1. How useEffect works in ReactJS?
2. How to concatenate unicode and variable in ReactJS?
3. How to pass data from one component to other component in ReactJS?
4. How to set input box to be a floating number in ReactJS?
5. How to use useState in arrow function instead of hook?
6. How to add theme to your React App?
7. How to add a function in JSX?
8. How to fetch data from APIs using Asynchronous await in ReactJS?
9. How to get cell value on React-Table?
10. How to access nested object in ReactJS?
11. How to set default value in select using ReactJS?
12. How to change the position of the element dynamically in ReactJS?
13. How to solve too many re-renders error in ReactJS?
14. How to publish a ReactJS component to NPM?
15. How to change the navbar color when you scroll in ReactJS?
16. How React Native is different from ReactJS?
17. How to Create a Countdown Timer Using ReactJS?
18. How to Create a Navigation Bar with Material-UI?
19. How to locally manage component's state in ReactJS?
20. How to add a CSS class whenever the component is updated in ReactJS?
21. How to Create a Toggle Switch in React as a Reusable Component?
22. How to create a custom progress bar component in React.js?
23. How to pass data from one component to other component in ReactJS?
24. How to convert functional component to class component in ReactJS?
25. How to put ReactJS component inside HTML string?
26. How to use CssBaseLine Component in ReactJS?
27. How to use ToggleButtonGroup Component in ReactJS?
28. How to use componentWillMount() in React Hooks?
29. How to use Link Component in ReactJS?
30. How to use BottomNavigation Component in ReactJS?
31. How to use Breadcrumbs Component in ReactJS?
32. How to use TextField Component in ReactJS?

33. How to use Portal Component in ReactJS?
34. How to use ScopedCssBaseline Component in ReactJS?
35. How to use Popper Component in ReactJS?
36. How to use Slide Component in ReactJS?
37. How to use Grow Component in ReactJS?
38. How to use Collapse Component in ReactJS?
39. How to use Fade Component in ReactJS?
40. How to use Zoom Component in ReactJS?
41. How to use Popover Component in ReactJS?
42. How to apply validation on Props in ReactJS?
43. What is prop drilling and how to avoid it?
44. How to create new elements with ReactJS mapping props?
45. How to pass multiple props in a single event handler in ReactJS?
46. How to send state/props to another component in React with onClick?
47. How to create a translucent text input in ReactJS?
48. How to set an object key inside a state object in React Hooks?

DSC 2F

Practical based on DSE 2 C (Data Visualization using Power BI)

1. Import data from different sources such as Excel, CSV, databases, and web services into Power BI.
2. Perform data cleaning tasks such as removing duplicates, handling missing values, and formatting data types.
3. Apply data transformation operations such as splitting columns, merging queries, and creating calculated columns.
4. Create relationships between different tables in the dataset.
5. Define hierarchies and drill-down paths to facilitate data exploration.
6. Implement calculated measures and calculated columns using DAX (Data Analysis Expressions).
7. Design interactive reports and dashboards using a variety of visualization types (e.g., bar charts, line charts, pie charts, maps, tables).
8. Customize the appearance of visualizations by adjusting formatting options such as colors, fonts, and labels.
9. Demonstrate at list 5 Maps
10. Demonstrate Table and Matrix with total and subtotal.
11. Demonstrate drill operations on any visuals
12. Incorporate slicers, filters, and bookmarks to enable users to interactively explore data.
13. Implement advanced calculations using DAX functions (e.g., CALCULATE, FILTER, RELATED, SUMMARIZE).

14. Perform time intelligence calculations such as year-over-year growth, moving averages, and cumulative totals.
15. Utilize AI-powered features such as Q&A (natural language querying) and Quick Insights to uncover hidden patterns and trends in data.
16. Create interactive tooltips and drill-through pages to provide additional context and detail to visualizations.
17. Demonstrate at list 5 Maps
18. Demonstrate any 10 visuals with formatting.
19. Demonstrate how to insert different objects and adding action to them.
20. Demonstrate different levels of filters.
21. Design sample report and dashboard.
22. Implement dynamic filtering and highlighting to focus attention on relevant data points.
23. Publish reports and dashboards to the Power BI Service for sharing with colleagues and stakeholders.
24. Configure row-level security to restrict access to sensitive data based on user roles and permissions.
25. Explore and utilize custom visuals available in the Power BI marketplace to enhance data visualizations.

Practical based on DSC 2G (ASP.NET Core)

1. Create a new ASP.NET Core MVC project using Visual Studio or the .NET CLI.
2. Explore the project structure and understand the role of important files such as Startup.cs, Program.cs, and the Views folder.
3. Define model classes representing entities in the application domain.
4. Generate scaffolded controllers and views using Entity Framework Core for CRUD operations on the model classes.
5. Customize the generated views and controllers to meet specific requirements.
6. Define custom routes using attribute routing and convention-based routing.
7. Implement route constraints to restrict the format of URL parameters.
8. Demonstrate how routing works and how URLs map to controller actions.
9. Create HTML forms for user input and data submission.
10. Implement form validation using data annotations and ModelState.IsValid.
11. Bind form data to model properties using model binding techniques.
12. Set up a database context and configure entity classes for use with Entity Framework Core.
13. Perform database migrations to create or update the database schema based on changes to the model
14. Demonstrate HTML helper tag.
15. Demonstrate Tag helper.
16. Data Annotation Attributes and Relationships in Entity Framework Core
17. Perform CRUD operations using Stored Procedures in Entity Framework Core
18. Demonstrate migration concept and add sample record using seeding technique

19. Demonstrate routing concept in ASP.Net core MVC
20. Demonstrate different ActionResult concept.
21. Implement CRUD operations (Create, Read, Update, Delete) using Entity Framework Core methods.
22. Create RESTful API endpoints for accessing application data using ASP.NET Core MVC controllers.
23. Implement HTTP methods (GET, POST, PUT, DELETE) to perform CRUD operations on resources.
24. Use attribute routing and model binding to define API routes and handle incoming requests.
25. Validate form input and handle form submissions before sending requests to the server.

Practical based on DSE 2 A (Linux and Shell Programming)

1. Navigate the file system using commands such as ls, cd, pwd, mkdir, and rmdir.
2. View file contents using commands such as cat, less, and head.
3. Manipulate files and directories using commands such as cp, mv, rm, and touch.
4. Use commands such as chmod and chown to change file permissions and ownership.
5. Understand the concept of file permissions (read, write, execute) for users, groups, and others.
6. View running processes and their resource usage using commands such as ps, top, and htop.
7. Manage processes using commands such as kill, killall, pkill, and pgrep.
8. Monitor system performance using commands such as df, du, and free.
9. Configure network settings using commands such as ifconfig, ip, and netstat.
10. Manage users and groups using commands such as useradd, usermod, groupadd, and passwd.
11. Shell Programming:
12. Write and execute a simple shell script using a text editor and the bash interpreter.
13. Use variables to store and manipulate data within a shell script.
14. Write comments to document the purpose and functionality of shell scripts.
15. Implement conditional statements in shell.
16. Use loop constructs (for, while) to iterate over lists of items or perform repetitive tasks.
17. Define and call functions within shell scripts to encapsulate reusable code blocks.
18. Pass arguments to functions and return values using the return statement.
19. Organize shell scripts into modular components for better code organization and maintainability.
20. Read input from files and process data using commands such as cat, grep, awk, and sed.
21. Use redirection operators (>, >>, <) to redirect input and output streams between files and commands.
22. Implement error handling mechanisms in shell scripts to gracefully handle unexpected conditions.
23. Use exit codes and error messages to communicate errors to users and other scripts.
24. Write Linux script for checking given number is prime, Armstrong and palindrome.
25. Write Linux script to display Fibonacci sequence up to n numbers.

Practical based on DSE 2 B (Mobile Application Development using Kotlin)

1. Set up a new Android project in Android Studio with Kotlin support.
2. Create a simple "Hello World" app that displays a greeting message on the screen. Run the app on an emulator or physical device to verify functionality.
3. Design and implement a user interface with various UI components such as TextViews, EditTexts, Buttons, and ImageViews.
4. Create an application that takes the name from a text box and shows hello message along with the name entered in text box, when the user clicks the OK button
5. Use layout files (XML) to define the structure and appearance of the app's screens. Apply styling and theming to enhance the visual appeal of the app.
6. Create a screen that has input boxes for User Name, Password, Address, Gender(radio buttons for male and female), Age (numeric), Date of Birth (Date Picket), State (Spinner) and a Submit button. On clicking the submit button, print all the data below the Submit Button (use any layout)
7. Implement event listeners to handle user interactions with UI elements (e.g., button clicks, text input).
8. Validate user input and provide feedback to users for incorrect or incomplete input.
9. Use input controls such as EditTexts and Spinners to capture user data.
10. Implement navigation between different screens or fragments within the app.
11. Use activities, fragments, and intents to navigate between screens and pass data between them.
12. Create a multi-screen app with a navigation drawer, tab layout, or bottom navigation bar for seamless navigation.
13. Integrate data storage solutions such as SharedPreferences, SQLite databases, or Room Persistence Library to store and retrieve app data.
14. Implement CRUD (Create, Read, Update, Delete) operations for managing data records within the app.
15. Fetch and display remote data from RESTful APIs using libraries like Retrofit or Volley.
16. Handling Device Features:
17. Design an android application Send SMS using Intent
18. Design an android application for menu.
19. Access and utilize device features such as camera, location, sensors, and permissions.
20. Implement functionality to capture photos or videos using the device's camera.
21. Retrieve the user's current location and display it on a map using Google Maps API.
22. Prepare the app for deployment by optimizing performance, reducing APK size, and ensuring compliance with platform guidelines.
23. Create a user registration application that stores the user details in a database table.
24. Generate signed APKs or app bundles for release and upload them to Google Play Store or other app distribution platforms.
25. Manage app releases, updates, and user feedback through the app store console and monitoring tools.

DSE 2-C

Practical Assignments on DSE 2B: (Internet of Things)

1. Write a program to sense the available networks using Arduino.
2. Write a program to measure the distance using ultrasonic sensor and make LED blink using Arduino.
3. Write a program to detect the vibration of an object with sensor using Arduino.
4. Write a program to connect with the available Wi-Fi using Arduino.
5. Write a program to sense a finger when it is placed on the board Arduino.
6. Write a program to get temperature notification using Arduino.
7. Write a program for LDR to vary the light intensity of LED using Arduino.
8. Write a program to install MySQL database in Raspberry pi.
9. Write a program to work with basic MySQL queries by fetching data from database in Raspberry pi.
10. Write a program to switch light on when the input is 1 and switch the light off when the input is 0 using Raspberry pi.

UA
Punyashlok Ahilyadevi Holkar Solapur University, Solapur.
Faculty of Science and Technology.
Nature of Question Paper
BCA- III w.e.f. AY 2024-25

Time: 3 Hrs

Total Marks: 80

Instructions

- 1) All Questions are compulsory 2) Figure to right indicate full marks.

Q.1 A) Choose correct alternative. (MCQ)

10 Marks

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.
- 7.
- 8.
- 9.
- 10.

Q. 1) B) Fill in the blank/Definition/One sentence answer/ One word answer
/ Give the name /Predict the product etc.

06 Marks

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

Q. 2) Solve any Eight of the following.

16 Marks

- a)
- b)
- c)
- d)
- e)
- f)
- g)
- h)
- i)
- j)

Q. 3) A) Attempt any Two of the following.

10 Marks

- a)
- b)
- c)

- Q. 3) B) Short note/Solve** **06 Marks**
- Q. 4 A) Attempt any Two of the following.** **08 Marks**
- a)
b)
c)
- Q. 4) B) Describe/Explain/Solve** **08 Marks**
- Q. 5) Attempt any Two of the following.** **16 Marks**
- a)
b)
c)

CA

Punyashlok Ahilyadevi Holkar Solapur University, Solapur
Faculty of Science and Technology. Nature of Question Paper for CBCS Pattern
BCA- III w.e.f. AY 2024-25

Time:

Internal Evaluation: 20 Marks

Choose any two of the following Home Assignment / Unit Test / Tutorial /Seminar

Pattern of Examination

External Evaluation + Internal Evaluation

80 Marks + 20 Marks = 100 Marks

Passing Criteria –

- 1. Written Exam – 32 out of 80**
- 2. Continuous Assessment (CA) – 08 out of 20**

Format for Title Page and For Embossing

PROJECT REPORT
ON
TITLE OF THE SYSTEM

Submitted in partial fulfillment for the award of degree of

BACHELOR IN COMPUTER APPLICATION
IN
COMPUTER APPLICATION

To the

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Submitted By

NAME OF STUDENTS

Name of College

Department of Computer Science

Under The Guide

Name of Guide

20__ - 20__

The Guidelines regarding the documentation and scope of project are mentioned here below:

For Application Level Projects

Project Report should be submitted in following format for Commercial Application Projects viz. Payroll, Sales, Purchase, Inventory, Book Shop, Examination system etc.

3.

- Blank Pages at beginning
- Title Page
- Certificate Page
- Certificate from Guide and Head of Department
- Declaration
- Acknowledgement Page
- Index with printed Page with Numbers

1. CHAPTER 1: INTRODUCTION

- 1.1. Existing System and Need for System
- 1.2. Scope of Work
- 1.3. Operating Environment – Hardware and Software
- 1.4. Detail Description of Technology Used

2. CHAPTER 2: PROPOSED SYSTEM

- 2.1. Proposed System
- 2.2. Objectives of System
- 2.3. User Requirements

3. CHAPTER 3 : ANALYSIS & DESIGN

- 3.1. Data Flow Diagram (DFD)
- 3.2. Entity Relationship Diagram (ERD)
- 3.3. Data Dictionary
- 3.4. Table Design
- 3.5. Code Design
- 3.6. Menu Tree
- 3.7. Input Screens
- 3.8. Report Formats
- 3.9. Test Procedures and Implementation

4. CHAPTER 4: USER MANUAL

- 4.1. User Manual
- 4.2. Operations Manual / Menu Explanation
- 4.3. Forms and Report Specifications

- 4.4. Drawbacks and Limitations
- 4.5. Proposed Enhancements
- 5. CHAPTER 5: Conclusions
- 6. Bibliography
- 7. ANNEXURES :
 - 7.1. ANNEXURE 1 : INPUT FORMS WITH DATA
 - 7.2. ANNEXURE 2 : OUTPUT REPORTS WITH DATA
 - 7.3. ANNEXURE 3 : SAMPLE CODE
- Blank Pages at the end.

For System-Level Projects

Project Report should be submitted in following format for Commercial Application Projects viz. Payroll, Sales, Purchase, Inventory, Book Shop, Examination system etc.

- 2 Blank Pages at beginning
- Title Page
- Certificate from Company
- Certificate from Guide and Head of the Department
- Declaration
- Acknowledgement
- Index with printed Page Numbers
- 1. CHAPTER 1: INTRODUCTION
 - 1.1. Existing System and Need for System
 - 1.2. Scope of Work
 - 1.3. Operating Environment – Hardware and Software
 - 1.4. Detail Description of Technology Used
- 2. CHAPTER 2: PROPOSED SYSTEM
 - 2.1. Proposed System
 - 2.2. Objectives of System
 - 2.3. User Requirements
- 3. CHAPTER 3: ANALYSIS & DESIGN
 - 3.1. Object Diagram
 - 3.2. Class Diagram
 - 3.3. Use Case Diagrams
 - 3.4. Module Hierarchy Diagram

- 3.5. Component Diagram
- 3.6. Module Specifications
- 3.7. Interface Diagram (in case of WAP and Embedded Systems)
- 3.8. User Interface Design (Screens etc.)
- 3.9. Table specifications (in case back end is a database)
- 3.10. Test Procedures and Implementation
- 4. CHAPTER 4: USER MANUAL
 - 4.1. User Manual
 - 4.2. Operations Manual / Menu Explanation
 - 4.3. Program Specifications / Flow Charts
 - 4.4. Drawbacks and Limitations
 - 4.5. Proposed Enhancements
- 5. CHAPTER 5: Conclusions
- 6. Bibliography
- 7. ANNEXURES :
 - 7.1. ANNEXURE 1 : USER INTERFACE SCREENS
 - 7.2. ANNEXURE 2 : OUTPUT REPORTS WITH DATA (if any)
 - 7.3. ANNEXURE 3 : SAMPLE PROGRAM CODE (which will prove sufficient