PUNYASHLOK AHILYADEVI HOLKAR SOLAPUR UNIVERSITY, SOLAPUR

Name of the Faculty: Science and Technology

NEP 2020

Syllabus: - Bachelor of Computer Application

Name of the Programme: BCA- III (Sem.– V and VI)

With Effect from: **June 2026**

BCA Part-III Sem.-V (NEP-2020)										
Subject/Core Course	Name and Type of the Paper			Hrs. / Week			Total Marks	UA	CA	Credits
	Type	Code	Name	L	T	P				
Major	DSC1-7		Data Communication and Networking	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-7	0	0	4	50	30	20	2
	DSC1-8		.Net Core	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-8	0	0	4	50	30	20	2
	DSC1-9		Artificial Intelligence	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-9	0	0	4	50	30	20	2
	DSE1-1		Data Science using Python	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-1	0	0	2	25	15	10	1
	OR									
	DSE1-2		NOSQL Database (MongoDB)	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-2	0	0	2	25	15	10	1
	OR									
	DSE1-3		Cyber Security	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-3	0	0	2	25	15	10	1
Vocational and Skill Enhancement Courses	VSC3		Hands on Training related to DSE	0	0	4	50	30	20	2
Ability Enhancement Course (AEC)	IKS2		Professional Ethics	2	0	0	50	30	20	2
Total				**13**	**0**	**18**	**550**	**330**	**220**	**22**

BCA Part-III Sem.-VI (NEP-2020)										
Subject/Core Course	Name and Type of the Paper			Hrs. / Week			Total Marks	UA	CA	Credits
	Type	Code	Name	L	T	P				
Major	DSC1-10		Data Warehousing and Data Mining	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-10	0	0	4	50	30	20	2
	DSC1-11		ASP.Net Core	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-11	0	0	4	50	30	20	2
	DSC1-12		Data Visualization using PowerBI	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-12	0	0	4	50	30	20	2
	DSE1-4		Linux and Shell Programming	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-4	0	0	2	25	15	10	1
	OR									
	DSE1-5		React JS	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-5	0	0	2	25	15	10	1
Vocational and Skill Enhancement Courses	VSC4		Hands on Training related to DSE	0	0	4	50	30	20	2

Field Project/ RP/CC/Internship/Apprenticeship/ Community Engagement and Services	FP2 / CEP2 / OJT1		Related to major subject	0	0	4	50	30	20	2
Total				**11**	**0**	**22**	**550**	**330**	**220**	**22**

Abbreviations:				
L: Lectures	**T**: Tutorials	**P**: Practical	**UA**: University Assessment	**CA**: College Assessment
Decipilne Specific Course: **DSC**		Decipilne Specific Elective: **DSE**		
Ability Enhancement Courses: **AES**		Indian Knowledge System: **IKS**		
Value Education Courses: **VEC**		Vocational Skill and Skill Enhancement Courses: **VSEC**		
Field Project/ RP/CC/Internship/ Apprenticeship/ Community Engagement and Services: **FP / CEP**				

B.C.A – III Semester-V						
Paper Category:	DSC1-7 **(Major)**					
Paper Name:	Data Communication and Networking					
Theory:	3 Hrs./Week		**Credit:**	3		
Marks:	**UA:**	45	**CA:**	30	**Total:**	75

Course Objectives:

1. To understand the fundamental concepts of data communication, networking components, protocols, and network topologies.
2. To study and compare network models such as OSI and TCP/IP and their layered architecture.
3. Analyze physical layer transmission and switching techniques.
4. Study data link layer error and flow control mechanisms.
5. Learn network layer routing and TCP/IP protocol operations.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain the basic concepts of data communication, network models, and topologies.
CO2: Analyze and differentiate analog/digital signals, transmission media, and switching techniques.
CO3: Apply error detection and correction methods and understand data link layer protocols.
CO4: Understand and evaluate routing algorithms and working of network devices.
CO5: Describe and use TCP/IP protocols and network services in practical scenarios.

Unit: I	Introduction and Physical Layer	20 Lectures

Introduction: Data communication Components, Data Flow, Protocols & Standards, Network Topologies, Connection oriented and connection less services.
Models: ISO-OSI reference model and the TCP/IP reference model.
Physical Layer:

- **Signal Analysis:** Analog & Digital signals, Period, Frequency, Phase, Amplitude, Bandwidth, Bit Rate, Bit Length.
- **Transmission Impairment:** Attenuation, Distortion, Noise.
- **Media Types:**
  - **Guided Media:** Twisted Pair, Coaxial Cable, and Fiber Optic Cable.
  - **Unguided (Wireless):** Radio Waves, Microwaves, Infrared, and Satellite Communication.
- **Digital & Analog Processing:** Manchester/Differential Manchester Coding, Pulse Code Modulation (PCM), and Modulation techniques (AM, FM, PM).
- **Transmission & Efficiency:** Parallel/Serial and Synchronous/Asynchronous modes; Multiplexing (FDM, TDM, WDM).

Switching: Circuit, Message, and Packet Switching.

Unit: II	Data Link Layer and Upper Layers	25 Lectures

Data Link Layer: Introduction

- **Error Management:** Types of errors, Hamming Distance, Parity Check, Cyclic Redundancy Check (CRC), Checksum, and Hamming Code.
- **Control Mechanisms:** Framing, Flow Control, and Error Control.
- **Protocols:** Stop and Wait (including ARQ), Go Back N ARQ, Selective Repeat ARQ, HDLC, and Point-to-Point Protocol.

Network Layer: Design issues and Routing Algorithms (Shortest Path, Distance Vector, and Link State).

- **Network Devices:** Hubs, Switches, Repeaters, Bridges, Routers, Gateways

TCP/IP Protocol Suite: Comprehensive study of UDP, TCP, IP, RTP, FTP, DNS, TELNET, SMTP, POP, HTTP, WWW, ARP, and RARP.

List of Reference Books:

- Computer Networking by Tanenbaum.
- Data Communication and Networking by William Stallings.
- Data Communication and Networking by B. A. Forouzan
- Data Communication and Networking by Jain

B.C.A – III Semester-V						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSC1-7					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1	Execute networking commands: arp, ipconfig, hostname, netstat, nslookup, pathping, ping, route, tracert.
2	Study different types of network cables.
3	Prepare and test straight-through and cross-over cables using a crimping tool.
4	Study IP address configuration: classification of IP addresses, static and dynamic addressing.
5	Study IPv4 and IPv6 addressing, subnetting, and supernetting.
6	Study network devices: switch, router, and bridge.
7	Configure a LAN, share a folder on one system, and access it from another system using IP address.
8	Block a website using Windows Defender Firewall.
9	Write the steps for printer installation in Windows.
10	Share a printer on a network and print from another computer.
11	Write a program to display the IP address of the local host.
12	Write a program to display the host name and IP address of a given website.
13	Write a program to demonstrate client–server communication using TCP sockets.
14	Write a TCP client program that sends a message to the server.
15	Write a TCP server program that receives and displays the client message.
16	Write a program to demonstrate UDP client–server communication.
17	Write a client–server program where the client sends two numbers and the server returns their sum.
18	Write a client–server program where the client sends a string and the server returns its length.
19	Write a client–server program where the client sends a string and the server converts it to uppercase.
20	Write a program to implement an echo server (server returns the same message sent by the client).

B.C.A – III Semester-V						
Paper Category:	DSC1-8 **(Major)**					
Paper Name:	**.**Net Core					
Theory:	3 Hrs./Week		**Credit:**	3		
Marks:	**UA:**	45	**CA:**	30	**Total:**	75

Course Objectives:

1. To understand how to design, implement, test, debug, and document programs using basic data types, input/output operations, conditional statements, control structures, string handling, and functions in C#.
2. To understand the importance of classes and objects along with constructors, arrays, and vectors (collections) in C#.
3. To study the principles of inheritance and interfaces, and demonstrate through problem-solving assignments how they relate to the design of methods, abstract classes, interfaces, and packages in C#.
4. To understand the importance of multithreading and different exception handling mechanisms in .NET applications.
5. To understand the basic concepts required to design GUI-based Windows applications using .NET and Windows Forms or related technologies.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Gain proficiency in designing, implementing, testing, debugging, and documenting programs using fundamental data types, computations, basic input/output operations, conditional and control structures, string manipulation, and functions in C#.

CO2: Demonstrate understanding of object-oriented programming concepts including classes, objects, constructors, arrays, and vectors (collections) within the .NET environment.

CO3: Apply the principles of inheritance and interfaces in program design and problem solving, and relate them to method design, abstract classes, interfaces, and package organization in C#.

CO4: Recognize and implement multithreading and exception handling mechanisms to develop robust and reliable applications using .NET.

CO5: Design and develop basic GUI-based Windows applications using Windows Forms or similar technologies in the .NET platform.

Unit: I	**Introduction to C# and Object Oriented Programming**	**20 Lectures**

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.

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: II	**Threading, Exception and Resource Management, Delegate, Event and LINQ**	**25 Lectures**

Exception: about exception, Exceptions Hierarchy, Throwing and Catching Exception, The try-finally Construct, IDisposable and the "using" Statement, Advantages of Using Exceptions, inbuild exception, custom exception,

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
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.

List of 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 Marshall

B.C.A – III Semester-V						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSC1-8					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1	Create a simple console application that prints **"Hello, World!"**
2	Compile and run the application using .NET CLI and Visual Studio
3	Perform basic arithmetic operations on numeric variables
4	Use string interpolation and concatenation to display variable values
5	Implement conditional statements (if-else, switch-case)
6	Use loops (for, while, do-while) to iterate over data
7	Implement nested loops and conditional logic
8	Declare and initialize arrays of different data types
9	Perform array operations such as sorting and searching
10	Define and call methods with different access modifiers
11	Pass parameters to methods and return values
12	Implement method overloading
13	Create classes and objects representing real-world entities
14	Implement encapsulation, inheritance, and polymorphism
15	Use constructors, properties, and methods to define object behavior
16	Implement try-catch blocks to handle exceptions
17	Create and use custom exceptions
18	Use finally blocks for cleanup operations
19	Read from and write to text files using StreamReader and StreamWriter
20	Implement file operations: read, write, append, delete
21	Demonstrate resource management using IDisposable and using
22	Use LINQ queries for filtering, sorting, grouping, and aggregation
23	Create and manage threads using the Thread class
24	Implement background and foreground threads
25	Use ThreadPool to execute parallel tasks

B.C.A – III Semester-V						
Paper Category:	DSC1-9					
Paper Name:	Artificial Intelligence					
Theory:	3 Hrs./Week		**Credit:**	3		
Marks:	**UA:**	45	**CA:**	30	**Total:**	75

Course Objectives:

1: To introduce students to the fundamental concepts of Artificial Intelligence, its real-world applications, and associated ethical considerations.

2: To develop an understanding of problem solving in AI using various search strategies and state-space representations.

3: To familiarize students with adversarial search strategies used in game-playing AI and with Constraint Satisfaction Problem techniques for solving combinatorial problems.

4: To acquaint students with the fundamentals of knowledge representation and reasoning mechanisms used in intelligent systems.

5: To cultivate analytical and creative thinking skills for solving real-world problems using intelligent, efficient, and optimal AI techniques

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain the fundamentals of AI and its various applications in real-world scenarios.

CO2: Build smart system using different informed search / uninformed search or heuristic approaches

CO3: Design AI systems using adversarial search algorithms and solve problems using constraint satisfaction techniques for structured environments.

CO4: Apply knowledge and reasoning algorithms for real-world problem-solving.

CO5: Represent complex problems with expressive yet carefully constrained language of representation.

Unit: I	**Introduction to Artificial Intelligence and Problem Solving**	**25 Lectures**

History and Applications of AI, type of AI, Intelligent Agents- Agents and environments, Rational agents, Types of environments, Structure of Agents, Problem-solving agents, Problem formulation, Overview of AI domains (ML, NLP, Robotics)

Searching Algorithms: Uninformed Search- Breadth First Search (BFS), Depth First Search (DFS), Informed Search- Heuristic functions, Hill Climbing, A*, AO* Algorithm

Game Playing: Adversarial search, Mini-Max Algorithm, Alpha-Beta Pruning, Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs.

Knowledge Representation & Reasoning: Knowledge-Based Agents, Propositional Logic- First Order Logic, Forward and Backward Chaining, Rule-Based Systems, Reasoning under uncertainty, basic probability & Bayes theorem

Applied Reasoning in AI Systems: Inference mechanisms, Rule engines in enterprise applications, Introduction to symbolic AI

Unit: II	**AI techniques**	**20 Lectures**

Expert Systems: Introduction & structure, Knowledge engineering, Knowledge acquisition methods, Types of expert systems, Model-based & case-based reasoning, Inference with uncertainty.

Basics of NLP: Challenges in NLP, Applications of NLP in Industry,

Text preprocessing- Tokenization, stop word removal, Stemming, Lemmatization

Language Models: N-grams, Part-of-Speech Tagging

Syntax & Semantics: Context-Free Grammar, Word Sense Disambiguation

NLP Resources: WordNet, Penn Treebank, Porter Stemmer

Introduction of Generative AI, Explainable AI (XAI), Federated Learning, Edge AI.

List of Reference Books:

- Nilsson Nils J , “Artificial Intelligence: A new Synthesis”, Morgan Kaufmann Publishers Inc. San Francisco, CA, ISBN:978-1-55-860467-4 2. Patrick Henry Winston, “Artificial Intelligence”, Addison-Wesley Publishing Company, ISBN: 0- 201-53377-4
- Natural Language Processing with Python, Steven Bird, Ewan Klein, Edward Loper, O’Reilly
- Andries P. Engelbrecht-Computational Intelligence: An Introduction, 2nd Edition-Wiley

IndiaISBN:978-0-470-51250-0 4. Dr. Lavika Goel, “Artificial Intelligence: Concepts and Applications”, Wiley publication, ISBN:978812655.

- Artificial Intelligence: A Modern Approach, Stuart Russell & Peter Norvig, Pearson / Prentice Hall
- Knowledge Representation and Reasoning, Ronald Brachman & Hector Levesque, Morgan Kaufmann
- Building Expert Systems: Principles, Procedures, and Applications, Frederick Hayes-Roth, Donald A. Waterman, Douglas B. Lenat, Addison-Wesley
- Speech and Language Processing, Daniel Jurafsky & James H. Martin, Pearson / Prentice Hall

B.C.A – III Semester-V						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical Based on DSC1-9					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1.	Write a program to implement Breadth-First Search (BFS)
2.	Write a program to implement Depth-First Search (DFS)
3.	Write a program to implement the Tic-Tac-Toe game.
4.	Write a program to implement the 8-Puzzle problem.
5.	Write a program to implement the Water Jug problem.
6.	Write a program to implement the Travelling Salesman Problem (TSP)
7.	Write a program to implement the A* Algorithm.
8.	Write a program to implement the Minmax algorithm.
9.	Write a program to implement the Monkey Banana problem.
10.	Write a program to implement Alpha-Beta Pruning
11.	Write a program to represent Propositional Logic statements and evaluate logical expressions (AND, OR, NOT, IMPLIES).
12.	Write a program to implement Predicate Logic representation with simple facts and rules.
13.	Write a program to demonstrate different Types of Knowledge (Declarative, Procedural, Heuristic, Meta-knowledge) with suitable examples.
14.	Write a program to implement a Rule-Based System using IF–THEN rules.
15.	Write a program to implement Forward Chaining Algorithm.
16.	Write a program to implement Backward Chaining Algorithm.
17.	Write a program to represent knowledge using a Semantic Network.
18.	Write a program to represent knowledge using Frames.
19.	Simple rule-based inference program
20.	Basic Bayesian probability example
21.	Implement Decision Tree
22.	Mini Expert System design
23.	Text preprocessing using NLTK
24.	Build a simple N-gram model

B.C.A – III Semester-V						
Paper Category:	DSE1-1 **(Major)**					
Paper Name:	Data Science using Python					
Theory:	2 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. To learn the fundamentals of web application development using the Django framework.
2. To understand and implement Object-Relational Mapping in Django.
3. To study database connectivity and data manipulation in Django-based web applications.
4. To understand the architecture and components of the Django web framework.
5. To learn the basic libraries used in Data Science such as NumPy, pandas, and Matplotlib.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Design, develop, build, and debug Python-based web applications using Django.
CO2: Use modern IDEs such as **PyCharm** and **Visual Studio Code** for Django development.
CO3: Create RESTful APIs using **Django REST Framework**.
CO4: Perform data analysis and visualization using Python libraries.

Unit: I	**Django Web Framework**	**15 Lectures**

Introduction to Web framework, Introduction to Django, History of Django, Advantages of Django, Django's MVT design Architecture, Environment Setup- Install Python, Create Virtual Environment, Install a Django. Django Commands Overview, Creating a Project, Project Structure, Creating App, Difference between Project and App, Creating Views, URL Routing, Django Models-Datatypes, Basic Django ORM Queries, Create Table (Model), Setting a Database, Migrate, Manipulating Data (CRUD) Operations, Django Template System- Render function, Django Template Language (DTL), Filters, Tags, Comments. Adding static and css files, Django Forms, Django Sessions and Cookies, Django Mixins, Django Middleware, Authentication and Authorization, Django Admin- Interface, Creating Super User, Including Models, Customise the Django Admin, Setting fields to Display, Add, Update, Delete an Object.

Django Rest Framework (DRF) - Introduction, What is REST API?, Requirements and Tools, Installation of DRF, Building REST API in Django- Create a Django Project, Create a Model, Make a Serializer, Create the Views, Set Up URLs and Test it. DRF Permissions- Built-in Permissions, Custom Permissions.

Unit: II	**Introduction to Data Science**	**15 Lectures**

Evolution of Data Science, Components of Data Science, Applications of Data Science, Data Science Life cycle, Roles in Data Science

Introduction to Numpy- Definition, Creating Numpy Arrays, Array Indexing, Array Slicing, Numpy Datatypes, Difference Between Copy and View, Operation on Numpy Array- Shaping, Reshaping, Iterating, Joining, Splitting, Searching, Sorting, and Filtering.

Introduction to pandas- Definition, Use of pandas, installation, Series, DataFrame- Creating Data Frame from an Excel, .csv file, Python Dictionary, Python List and Tuples, Operations on Data Frames, Indexing, Selection, Filtering, 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.

List of Reference Books:

- Python Web Development with Django, Jeff Forcier, ,Paul Bissex.Wesley J Chun
- Django: Web Development with Python, Samuel Dauzon, ,Aidas BendoraitisArun Ravindran
- Web Development with Django, Ben Shaw, Saurabh Badhwar, Chris Guest, Bharath Chandra KS
- The Definitive Guide to Django, Jacob Kaplan-Moss, .Adrian Holovaty
- An Introduction to Data Science, Jeffrey S. Saltz, Jeffrey M. Stanton

B.C.A – III Semester-V						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSE1-1					
Practical:	2 Hrs./Week		**Credit:**	1		
Marks :	**UA:**	15	**CA:**	10	**Total:**	25

1.	Django Installation and Project Setup.
2.	Demonstrate the steps to create simple Django project.
3.	Demonstrate the steps to create simple Django app.
4.	Create Django project Student Management System which contains Add Student, View Student List, Update Student and Delete Student views.
5.	Create URL patterns, views, and templates to render dynamic HTML pages using Django's templating engine
6.	Implement data filtering, searching, and sorting in Django using query parameters.
7.	Create an application to add static CSS files in a Django project and apply styling to web pages.
8.	Create an application to add static Image files in a Django project and apply to web pages.
9.	Create an application to add static JavaScript files in a Django project and apply to web pages.
10.	Create and validate a form using Django Forms and display submitted data.
11.	Create an application to implement session management in Django.
12.	Create an application to demonstrate create, retrieve, and delete cookies in Django.
13.	Create an application to implement Mixins in Django Class-Based Views (CBVs).
14.	Create Custom Middleware in Django for processing requests and responses.
15.	Create User Authentication System to Implement login system contains User Registration, Login, Logout, Dashboard (Login required), Display username after login.
16.	Install DRF and Create REST API-Create Student API: GET, POST, PUT, PATCH and DELETE, Test using Postman/swagger.
17.	Create a 1D, 2D, and 3D NumPy array
18.	Convert a Python list into a NumPy array.

19.	Generate an array of 10 random integers between 1 and 100.
20.	Perform addition, subtraction, multiplication, and division on two NumPy arrays.
21.	Compute the mean, median, standard deviation, and variance of a given array.
22.	Create a Pandas Series from a list, NumPy array, and dictionary.
23.	Create a DataFrame from a dictionary of lists.
24.	Create a pandas dataframe and add a new column to a DataFrame.
25.	Write a program to demonstrate Line Plot using Matplotlib.
26.	Write a program to demonstrate Bar Chart using Matplotlib
27.	Write a program to demonstrate Scatter Plot using Matplotlib
28.	Write a program to demonstrate Histogram using Seaborn
29.	Write a program to demonstrate Box Plot using Seaborn
30.	Write a program to demonstrate Heatmap using Seaborn

B.C.A – III Semester-V						
Paper Category:	DSE1-2 **(Major)**					
Paper Name:	NOSQL Database (MongoDB)					
Theory:	2 Hrs./Week		**Credit:**	2		
Marks :	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. To understand the advantages of MongoDB over traditional Relational Database Management System.
2. To learn MongoDB data types, installation procedures, and data modeling techniques.
3. To perform database and collection operations using MongoDB commands.
4. To implement CRUD operations and advanced querying mechanisms.
5. To apply the Aggregation Framework and indexing techniques for query optimization.
6. To understand MongoDB security features such as authentication and authorization.
7. To explore replication, sharing, and scalability concepts in MongoDB.
8. To perform bulk operations and execute advanced administrative commands.
9. To integrate MongoDB with programming languages such as **Java**, **PHP**, and **Python**.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Understand MongoDB data types, installation, and data modeling strategies.
CO2: Apply query, projection, and update operators to manipulate MongoDB documents.
CO3: Perform database and collection operations such as create, drop, and list.
CO4: Utilize advanced MongoDB features including text search, upserts, multi-update, and aggregation pipelines.
CO5: Develop applications by integrating MongoDB with programming languages such as Java, PHP, and Python.

Unit: I	**MongoDB Fundamentals and Core Operations**	**15 Lectures**

Introduction: Introduction to NoSQL Databases, Advantages over RDBMS, Introduction to MongoDB, Architecture of MongoDB, MongoDB Data Types, Installation, Data Modeling.

Operators and Commands: Query & Projection Operators, Update Operators, Aggregation Pipeline Stages, limit(), sort(), Aggregation Commands, Geospatial Commands, Query & Write Commands, Query Plan Cache Commands, Authentication Commands, User Management, Role Management, Replication Commands, Sharding Commands, Session Commands.

Database and Collection Operations: Create Database, Drop Database, Create Collection, Drop Collection

CRUD and Advanced Operations: Insert, Update, Delete, Query Documents, SQL to MongoDB Mapping, Text Search, Partial Updates, Document Limits, Multi-Update and Upsert operations, Wire Protocol, Bulk Operations, Common administrative commands

Unit: II	**Advanced MongoDB, Commands and Connectivity**	**15 Lectures**

MongoDB Shell and Methods: MongoDB shell usage, Collection methods and cursor methods, Database commands and query plan cache methods, Security and Replication, User and role management Authentication mechanisms, Replication configuration and methods, Connectivity and Integration Connecting MongoDB with: Java, PHP,Python, Basic driver usage and CRUD operations from applications

List of Reference Books:

- Kristina Chodorow – *MongoDB: The Definitive Guide*
- Shannon Bradshaw – *MongoDB Applied Design Patterns*
- Dan Sullivan – *NoSQL for Mere Mortals*
- Guy Harrison – *Next Generation Databases: NoSQL, Big Data, and Cloud*
- Pramod J. Sadalage and Martin Fowler – *NoSQL Distilled*

B.C.A – III Semester-V						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSE1-2					
Practical:	2 Hrs./Week		**Credit:**	1		
Marks :	**UA:**	15	**CA:**	10	**Total:**	25

1	Installation and configuration of MongoDB Community Edition.
2	Installation and use of MongoDB Compass (GUI Tool).
3	Working with MongoDB Shell (mongosh).
4	Creating and dropping databases.
5	Creating and dropping collections.
6	Inserting single and multiple documents.
7	Performing find() queries with conditions.
8	Using comparison operators ($eq, $gt, $lt, $gte, $lte).
9	Using logical operators ($and, $or, $not, $nor).
10	Projection of specific fields in query results.
11	Sorting query results (ascending and descending).
12	Using limit() and skip() methods.
13	Updating documents using $set, $unset, $inc.
14	Replacing entire documents using replaceOne().
15	Deleting documents using deleteOne() and deleteMany().
16	Working with embedded documents.
17	Working with arrays and array operators ($push, $pull, $addToSet).
18	Querying arrays using $elemMatch.
19	Creating Single Field Index.
20	Creating Compound Index.
21	Creating Multikey Index on array fields.
22	Creating Text Index and performing text search.
23	Viewing and managing indexes using getIndexes().
24	Perform the following operations: Create database, Create collection, List databases and collections, Drop database and collection
25	Insert single and multiple documents into a collection using: insertOne(),insertMany()
26	Retrieve documents using:find()
27	Perform update operations using: updateOne(),updateMany()
28	Retrieve specific fields using projection and sort results in ascending and descending order.
29	Create a text index and perform text search queries on a collection.
30	Perform aggregation operations using: $match,$group,$project,$sort
31	Create single-field and compound indexes and analyze query performance.
32	Design a MongoDB schema for: Student database,E-commerce application
33	Perform bulk write operations using: bulkWrite()
34	Create users, assign roles, and authenticate using MongoDB security features.
35	Demonstrate basic steps for configuring: Replica set, Sharded cluster (conceptual or simulated)
36	Connect MongoDB with Python using PyMongo and perform CRUD operations.
37	Write a Java program to connect to MongoDB and perform insert and query operations.

B.C.A – III Semester-V						
Paper Category:	DSE1-3 **(Major)**					
Paper Name:	Cyber Security					
Theory:	2 Hrs./Week		**Credit:**	2		
Marks :	**UA:**	30	**CA:**	20	**Total:**	50

The syllabus shall be prescribed by Punyashlok Ahilyadevi Holkar Solapur University, Solapur.

B.C.A – III Semester-V						
Paper Category:	VSC3					
Paper Name:	Hands on Training related to DSE1-1					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks :	**UA:**	30	**CA:**	20	**Total:**	50

1.	Develop a CRUD (Create, Read, Update, Delete) application with Django for managing a simple database-backed model.
2.	Create Django Forms and Validation with Registration Form contains Name, Email, Password, Confirm Password and Validation Required fields, Email validation, Password match validation.
3.	Create views and templates for listing, adding, editing, and deleting records from the database.
4.	Integrate user authentication and authorization into a Django application using Django's built-in authentication system. Create user registration and login forms, and implement password reset functionality. Restrict access to certain views or functionalities based on user roles and permissions.
5.	Customize the Django Admin Interface by modifying model display, filters, search, ordering, and form layout.
6.	Define models with relationships (one-to-one, one-to-many, many-to-many) to represent complex data structures. Implement CRUD operations for related models and navigate relationships using Django's ORM (Object-Relational Mapper).
7.	Use Django's admin interface to manage database records and relationships.
8.	Develop a Web Application Online Library Management System includes the Modules: Admin Login, Add Book, Issue Book, Return Book and Fine Calculation.
9.	Create a RESTful API using Django REST Framework with CRUD operations and implement permissions.
10.	Extract specific elements from a given NumPy array
11.	Reverse a NumPy array using slicing.
12.	Find the maximum and minimum values along a specific axis.
13.	Convert a 1D array into a 2D matrix of shape (3,3).
14.	Flatten a multi-dimensional array into a 1D array.

15.	Resize an existing NumPy array without changing its elements.
16.	Extract all even numbers from a NumPy array.
17.	Replace all negative values in an array with zero.
18.	Load a CSV file into a Pandas DataFrame and display its first 5 rows.
19.	Sort a DataFrame based on a column in ascending and descending order
20.	Create a DataFrame with a categorical column (e.g., "Department") and find all unique departments.
21.	Find the frequency of different categories in a "Product Category" column.
22.	Line Plot using Matplotlib I. Create a dataset representing the monthly sales of a company for one year. II. Plot a line chart using Matplotlib. III. Label the x-axis as "Month" and the y-axis as "Sales". **IV.** Add a title and grid to the plot.
23.	Bar Chart using Matplotlib I. Create a dataset with product categories and their respective sales. II. Plot a bar chart showing the sales of each category. **III.** Add appropriate labels and colors.
24.	Scatter Plot using Matplotlib I. Create a dataset with advertising expenses vs. sales. II. Plot a scatter plot showing the relationship. **III.** Add labels and a title.
25.	Histogram using Seaborn I. Create a dataset with the ages of customers. II. Plot a histogram showing the distribution of ages. **III.** Add labels and a title.
26.	Box Plot using Seaborn I. Create a dataset with sales data for different product categories. **II.** Plot a box plot to analyse sales distribution.
27.	Heatmap using Seaborn I. Create a dataset with correlation values between different features (e.g., sales, advertising, profit). **II.** Plot a heatmap to visualize correlations.

B.C.A – III Semester-V						
Paper Category:	VSC3					
Paper Name:	Hands on Training related to DSE1-2					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks :	**UA:**	30	**CA:**	20	**Total:**	50

1.	Create and explore a NoSQL document structure: Insert sample JSON documents to demonstrate flexible schemas.
2.	Compare RDBMS vs MongoDB with a practical schema: Model the same data (e.g., user accounts) in SQL and MongoDB.
3.	Explore MongoDB data types: Insert and query documents using types like String, NumberInt, Boolean, Array, Date, etc.
4.	Basic MongoDB data modeling example: Design embedded vs referenced models for blog posts and comments.
5.	Use query and projection operators: Demonstrate $eq, $gt, $lt, $in, $and, $or, $exists, and projection { field: 1 }.
6.	Create and drop a database: Use use dbName, db.dropDatabase().
7.	Create and drop collections: Use db.createCollection("students"), db.students.drop().
8.	Explore collection indexes and options: Create indexes and check using db.collection.getIndexes().
9.	Set up schema validation rules: Use JSON schema validation to restrict document structure.
10.	CRUD operations: Insert, query with filters, update fields, and delete documents.
11.	Use of db.runCommand() and server information: Run db.runCommand({ serverStatus: 1 }) and db.isMaster().
12.	Bulk operations and upsert example: Demonstrate bulkWrite() with mixed inserts and updates.
13.	Check collection statistics and perform partial updates: Use db.collection.stats() and $set for field-level updates.
14.	Using MongoDB shell: collection and cursor methods: Demonstrate .find(), .countDocuments(), .forEach(), .toArray().
15.	Query plan cache and role management: Use db.collection.getPlanCache().clear() and db.createRole().
16.	Python MongoDB CRUD application using PyMongo: Connect to MongoDB Atlas/local and perform CRUD using Python.
17.	Java MongoDB connection example: Use MongoDB Java Driver to connect and perform basic operations.
18.	Backup and Restore using mongodump and mongorestore.

B.C.A – III Semester-V						
Paper Category:	VSC3					
Paper Name:	Hands on Training related to DSE1-3					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks :	**UA:**	30	**CA:**	20	**Total:**	50

The practical list shall be prescribed by Punyashlok Ahilyadevi Holkar Solapur University, Solapur.

B.C.A – III Semester-V						
Paper Category:	IKS2					
Paper Name:	Professional Ethics					
Theory:	2 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1: To introduce students to the fundamental concepts, principles, and importance of professional ethics in the field of computer science.

2: To familiarize students with major ethical theories such as Utilitarianism, Deontology, Virtue Ethics and their relevance to computing professions.

3: To make students aware of professional codes of conduct established by organizations such as

- Association for Computing Machinery and
- Institute of Electrical and Electronics Engineers.

4: To develop the ability to recognize ethical, legal, and social issues related to privacy, intellectual property, cybersecurity, and professional responsibility.

5: To enable students to analyze and resolve ethical dilemmas using structured ethical decision-making approaches.

6: To promote responsible computing practices and encourage ethical behavior in software development, data handling, and digital communication.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Understand the fundamental concepts, principles, and importance of professional ethics in computer science.

CO2: Apply ethical theories and decision-making models to analyze real-world professional situations.

CO3: Identify, evaluate, and resolve ethical dilemmas encountered in computing and IT professions.

CO4: Demonstrate ethical responsibility and integrate professional ethics into daily professional and organizational practices.

Unit: I	**Introduction to Professional Ethics in Computer Science**	**15 Lectures**

Definition and Importance of Professional Ethics, Ethical Theories: Utilitarianism, Deontology, Virtue Ethics, Code of Ethics in Computer Science (ACM, IEEE), Ethical Responsibilities of Computer Professionals Overview of Ethical Issues in Technology Privacy and Confidentiality, Understanding Privacy in the Digital Age, Ethical Implications of Data Collection and Usage Confidentiality Issues in Software Development

Unit: II	**Intellectual Property and Copyright**	**15 Lectures**

Intellectual Property and Copyright, Overview of Intellectual Property Rights, Copyright, Trademarks, and Patents in Software, Ethical Use of Open-Source Software Plagiarism in Programming and Software Development Security, Cybercrime, and Professional Accountability, Ethical Considerations in Cybersecurity, Responsibility for Security in Software Development Ethical Hacking vs. Malicious Hacking

List of Reference Books:

- Reynolds, G. (2018). Ethics in Information Technology (6th ed.). Cengage Learning.
- Spinello, R. A. (2007). Computer Ethics: A Global Perspective. Jones and Bartlett Publishers.
- Quinn, M. J. (2024). Ethics for the Information Age (9th ed.). Pearson Education.
- Perkins, C. B. (2001). Computers and Society. Pearson Education.
- Spinello, R. A. (2003). The Ethics of Information Technology and Business. Wadsworth Publishing.
- Cooper, T. L. (2006). The Responsible Administrator: An Approach to the Ethics of the Profession (3rd ed.). Jossey-Bass.

B.C.A – III Semester-VI						
Paper Category:	DSC1-10 **(Major)**					
Paper Name:	Data Warehousing and Data Mining					
Theory:	3 Hrs./Week		**Credit:**	3		
Marks :	**UA:**	45	**CA:**	30	**Total:**	75

Course Objectives:

1. To introduce the concepts, architecture, and components of Data Warehousing systems.
2. To understand multidimensional data models and perform OLAP operations
3. To study data preprocessing techniques and the ETL process used in preparing data for analysis.
4. To learn fundamental Data Mining techniques such as association rule mining, classification, regression, and clustering.
5. To apply data mining algorithms to discover knowledge and patterns from large datasets for decision making.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain the architecture, components, and design principles of data warehousing systems.

CO2: Design multidimensional schemas and perform OLAP operations such as roll-up, drill-down, slice, and dice.

CO3: Apply data preprocessing techniques including data cleaning, integration, transformation, and reduction.

CO4: Implement data mining algorithms such as Apriori, decision trees, regression models, and clustering techniques.

CO5: Evaluate and apply appropriate data mining methods to analyze real-world datasets and support decision-making processes.

Unit: I	**Introduction to Data Warehouse and Data Mining**	**20 Lectures**

Introduction to Data Warehouse: 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,

Introduction to Data Mining: Evolution, KDD, What kind of data, Architecture, data mining views, Data Mining Functionalities, Issues in Data Mining.

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

Association Rule Mining: Market Basket Analysis, Frequent Itemsets, Closed Itemsets, Association Rules, Apriori Algorithm, Frequent Pattern-Growth algorithm, Mining Multilevel and multidimensional Association Rules, Constraint-Based Frequent Pattern Mining

Unit: II	**Classification, Regression, and Clustering Techniques**	**25 Lectures**

Supervised and Unsupervised learning: Classification, regression, difference between classification and regressing, Issues regarding Classification and Predication, Binary and Multiclass Classification.

Types of classifications algorithm: Decision tree induction, Bayesian Classification, Back propagation, Logistic regression, k-Nearest-Neighbor, SVM, Ensemble Methods-Bagging, Boosting, AdaBoost, Random Forests, Model Evaluation for Classification

Types of regression algorithm: simple linear, multiple linear and polynomial regression, Model Evaluation for regression, Cross-Validation, underfitting and overfitting

What is Cluster Analysis, Difference between classification and clustering,

Clustering Methods: Partitioning Methods, Hierarchical Methods, Density-Based Methods, Model-Based Clustering Methods, Applications and Trends in Data Mining.

List of Reference Books:

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

B.C.A – III Semester-VI						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical based on DSC1-10					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1) Create and load data set in Weka. Apply pre-processing operations on given attributes.
(Note: Use following Excel data file (Name it as weather.arff) for doing above activity).

Outlook	Temp	Humidity	Windy	Class
Sunny	75	70	yes	Play
Sunny	80	90	yes	Dontplay
Sunny	85	85	no	Dontplay
Sunny	72	95	no	Dontplay
Sunny	69	70	no	Play
Overcast	72	90	yes	Play
Overcast	83	78	no	Play
Overcast	64	65	yes	Play
Overcast	81	75	no	Play
Rain	71	80	yes	Dontplay
Rain	65	70	yes	Dontplay
Rain	75	80	no	Play
Rain	68	80	no	Play
Rain	70	96	no	Play

2) Create and load following data set in Weka. Perform various pre-processing operations on given attributes. (**Note :** Use following Excel data set (Name it as All_Electronics.arff)).

RID	Age	income	student	credit-rating	class
1	Youth	high	no	fair	no
2	Youth	high	no	excellent	no
3	middle-age	high	no	fair	yes
4	Senior	medium	no	fair	yes
5	Senior	low	yes	fair	yes
6	Senior	low	yes	excellent	no
7	middle-age	low	yes	excellent	yes
8	Youth	medium	no	fair	no
9	Youth	low	yes	fair	yes
10	Senior	medium	yes	fair	yes
11	Youth	medium	yes	excellent	yes
12	middle-age	medium	no	excellent	yes
13	middle-age	high	yes	fair	yes
14	Senior	medium	no	excellent	no

3) Use ID3(Iterative Dichotomiser 3) algorithm to classify weather data from the "weather.arff" file. Perform initial pre-processing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data.

4) Create the appropriate Regression Model with Weka by using following data set. Calculate the selling price of last house by using regression model.

House size (sq.feet)	Lot size	Bedrooms	Granite	Upgraded bathroom?	Selling price
3529	9191	6	0	0	$205,000
3247	10061	5	1	1	$224,900
4032	10150	5	0	1	$197,900
2397	14156	4	1	0	$189,900
2200	9600	4	0	1	$195,000
3536	19994	6	1	1	$325,000

2983	9365	5	0	1	$230,000
3198	9669	5	1	1	????

5) Demonstrate data reduction techniques (sampling, feature selection)

6) Design star schema and snowflake schema for a sample data warehouse

7) Perform OLAP operations: roll-up, drill-down, slice, and dice on sample data

8) Generate frequent itemsets and association rules using Apriori in **Weka**

9) Implement FP-Growth algorithm and compare results with Apriori

10) Load the weather.nominal dataset. Use the following filters in weka.

I. Remove all instances in which the humidity attribute has the value high.
II. Convert numeric value to nominal
III. Convert nominal to string
IV. Discretizes data

11) Comparing Classification Algorithms. Implement and evaluate multiple classification algorithms in Weka such as: Decision Tree, Naïve Bayes, k-Nearest Neighbor

- Train models using cross-validation.
- Compare their performance using: Accuracy,Precision,Recall,F1-score
- Identify the best performing classifier.

12) Apply clustering algorithms on a dataset using Weka.

- Implement K-means clustering.
- Apply hierarchical clustering.
- Visualize cluster assignments.
- Analyze cluster quality.
- Feature Selection

13) Perform simple linear regression on a dataset

14) Perform multiple linear regression and analyze coefficients

15) Evaluate regression models using MSE and R^2

16) Create the dataset house_selling_rate.arff using the given housing data. Use Weka to build a regression model and predict the missing selling price.

- Load the dataset in Weka.
- Apply Linear Regression algorithm.
- Predict the missing value.
- Interpret the regression model output.

17) Load the soybean.arff dataset and apply J48 classifier.

- Train the classifier.
- Analyze the confusion matrix.
- Visualize and interpret the decision tree.
- Association Rule Mining

18) Apply Apriori algorithm on any dataset to discover frequent itemsets and association rules.

- Set minimum support and confidence values.
- Generate association rules.
- Interpret at least three interesting rules.
- Data Filtering and Transformation

19) Load the iris.arff dataset and perform the following preprocessing operations:

- Add noise to the class attribute.
- Randomize the dataset.
- Normalize the numeric attributes.
- Reorder the attributes.

20) Use Weka's attribute selection methods to identify the most informative features.

- Apply InfoGainAttributeEval.
- Apply subset selection methods.
- Compare classification performance before and after feature selection.

B.C.A – III Semester-VI						
Paper Category:	DSC1-11					
Paper Name:	ASP.Net Core					
Theory:	3 Hrs./Week		**Credit:**	3		
Marks:	**UA:**	45	**CA:**	30	**Total:**	75

Course Objectives:

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.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Demonstrate the creation of ASP.NET Core MVC Web Applications using .NET 8.
CO2: Understand project file structure and implement Models, Controllers, Views, and Dependency Injection in ASP.NET Core MVC applications.
CO3: Utilize Entity Framework Core for Data Access and perform database operations using Entity Framework Core.
CO4: Discuss transactions, migration, and database seeding in Entity Framework Core.
CO5: Implement Reusability in Views using Partial Views, View Components, and the Razor View Engine for efficient view management.
CO6: Cover Action Results, Routing, Model Binding, HTML Helpers, and Tag Helpers in ASP.NET Core.
CO7: Data Annotations and Model Validations, including custom validations and remote validation.
CO8: Discuss different methods of State Management, including Cookies and Sessions.

Unit: I	**Introduction to ASP.Net Core and Entity Framework Core**	**25 Lectures**

Overview of Microsoft Web Technologies, Introduction to ASP.NET Core Framework. NET Core Environment Setup, Install .NET Core SDK, SQL Server 2022, SSMS, Creating ASP.NET Core Web Application using .NET 8, NET Core Project File Structure, NET Core Main Method, NET Core In Process Hosting, out of Process Hosting, Launch Settings.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.

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: II	**Model, View, Controller and Routing, HTML, Tag Helper, Data Annotation Validation and State management**	**20 Lectures**

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.
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.

List of Reference Books:

- Professional ASP.NET– Wrox Publication by Bill Evjen, Scott Hanselman, Farhan Muhammed, Sirnivasa Sivakumar, Devin Rader.
- Microsoft ASP.NET Step by Step - Microsoft Press by George Shepherd.

B.C.A – III Semester-VI						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical Based on DSC1-11					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

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.	Deminstrate 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.	Working with Entity Framework Core: I. Set up a database context and configure entity classes for use with Entity Framework Core. II. Perform database migrations to create or update the database schema based on changes to the model classes.
13.	Implement CRUD operations (Create, Read, Update, Delete) using Entity Framework Core methods.
14.	Create RESTful API endpoints for accessing application data using ASP.NET Core MVC controllers.
15.	Implement HTTP methods (GET, POST, PUT, DELETE) to perform CRUD operations on resources.
16.	Use attribute routing and model binding to define API routes and handle incoming requests.
17.	Validate form input and handle form submissions using JavaScript before sending requests to the server.

B.C.A – III Semester-VI						
Paper Category:	DSC1-12					
Paper Name:	Data Visualization using PowerBI					
Theory:	3 Hrs./Week		**Credit:**	3		
Marks:	**UA:**	45	**CA:**	30	**Total:**	75

Course Objectives:

1: To introduce students to the concepts of Business Intelligence and the role of Microsoft Power BI in modern analytics.

2: To develop proficiency in importing, cleaning, and transforming data from multiple sources using Power BI tools.

3: To enable students to design and implement effective data models by creating relationships, calculated columns, and measures.

4: To equip students with the skills required to create meaningful visualizations, dashboards, and interactive reports for business insights.

5: To familiarize students with dashboard storytelling, deployment, and secure sharing of Power BI reports

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain the architecture, components, and applications of Power BI in business intelligence solutions.

CO2: Perform data extraction, transformation, and loading (ETL) using Power Query Editor.

CO3: Develop efficient data models and implement calculations using **Data Analysis Expressions**.

CO4: Design interactive dashboards and reports using charts, maps, slicers, and KPIs.

CO5: Publish, share, and maintain Power BI dashboards securely for organizational use.

Unit: I	**Introduction to Power BI**	**20 Lectures**

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.

Creating a Data Model

Editing Data After a Data Load, Transforming Data Before Loading 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: II	**Transforming Datasets , Power Query Editor and Data Visualizations**	**25 Lectures**

Transforming Datasets

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.

Power Query Editor and Data Visualizations

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, When to use KPI, Requirements for KPI, KPI Visualizations, Visual selection, Layout, Navigation pane, Full screen mode, Supporting tiles, Custom date filters, Single- dashboard, Multiple dashboard, 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

List of Reference Books:

- Pro Power BI Desktop-Free interactive data analysis with Microsoft Power BI by Adam Aspin, A press
- Introducing Microsoft Power BI by Alberto Ferrari and Marco Russo, Microsoft Press
- Mastering Microsoft Power BI by Brett Powell, Packt BIRMINGHAM– MUMBAI
- Microsoft Power BI Complete Reference by Devin Knight, Brian Knight, Mitchell
- Pearson, Manuel Quintana, Brett Powell, Packet
- Learn Power BI by Greg Deckler, PacktI (CBCS).

B.C.A – III Semester-VI						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical Based on DSC1-12					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

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

B.C.A – III Semester-VI						
Paper Category:	DSE 1-4					
Paper Name:	Linux and Shell Programming					
Theory:	2 Hrs./Week		**Credit:**	2		
Marks :	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. To introduce basic Linux operating system concepts and general-purpose commands.
2. To familiarize students with text editors such as vi and vim.
3. To develop understanding of shell scripting concepts and programming.
4. To provide knowledge of file management, permissions, and advanced Linux commands.
5. To introduce scripting tools such as awk, grep for text processing.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain basic Linux concepts, architecture, and general-purpose commands

CO2: Apply file management techniques and modify file permissions and ownership

CO3: Utilize text processing tools such as awk, grep, and basic scripting

CO4: Develop and execute shell scripts using variables, control structures, and functions

CO5: Perform basic system administration tasks including user management, process handling etc.

Unit: I	**Introduction of Linux and Linux Commands**	**15 Lectures**

Introduction of Linux: 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 & 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

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., I/O and Redirection, Piping

Linux Commands: Filter: head, tail, pr, cut, paste, sort, uniq, tr, grep, egrep, fgrep, sed. Communication commands: mesg, talk, write, wall, mail., Archive and File compression commands Text Editors- vi, vim

Unit: II	**Linux System Management and Shell Programming**	**15 Lectures**

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.

Shell Programming: 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

List of Reference Books:

- Official Red Hat Linux Users guide by Redhat, Wiley Dreamtech India
- Beginning Linux Programming by Neil Mathew & Richard Stones, Wiley Dreamtech India
- Red Hat Linux Bible by Cristopher Negus, Wiley Dreamtech India
- UNIX Shell Programming by Yeswant Kanethkar, BPB

B.C.A – III Semester-VI						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical Based on DSE 1-4					
Practical:	2 Hrs./Week		**Credit:**	1		
Marks :	**UA:**	15	**CA:**	10	**Total:**	25

1	Demonstrate basic Linux commands: pwd, ls, date, who, cal and explain their outputs
2	Create, navigate, and remove directories using mkdir, cd, rmdir
3	Create and display files using cat, more, less
4	Perform file operations using cp, mv, rm with suitable examples
5	Search files using find command with multiple conditions
6	Demonstrate input-output redirection and piping with practical examples
7	Explain Linux file system hierarchy and display disk usage using df and du
8	Demonstrate file types and directory listing using ls -l
9	Mount and unmount storage devices (USB/CD)
10	Archive and compress files using tar, gzip, and zip
11	Print files using lpr, lpq, lprm commands
12	Demonstrate use of head, tail, cut, paste
13	Sort and remove duplicate entries using sort and uniq
14	Search patterns using grep, egrep, fgrep
15	Perform text substitution using sed
16	Combine multiple filters using pipes to process a text file
17	Create users and groups using useradd, groupadd
18	Modify file permissions using chmod (symbolic and numeric modes)
19	Change ownership using chown and chgrp
20	Demonstrate special permissions (SUID, SGID, Sticky bit) with examples
21	Display and analyze running processes using ps
22	Manage processes using kill and demonstrate process termination
23	Execute background and foreground jobs using &, jobs, fg
24	Change process priority using nice and explain its effect
25	Demonstrate disk partitioning using fdisk (theoretical or simulation)
26	Explain boot loaders (GRUB/LILO) and system configuration basics
27	Write a shell script to display system information (date, users, disk usage)
28	Write a shell script using loops (for, while)
29	Write a shell script using conditional statements (if, case)
30	Write a shell script using functions and command-line arguments

B.C.A – III Semester-VI

Paper Category:	DSE 1-5					
Paper Name:	ReactJS					
Theory:	2 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

Course Objectives:

1. To introduce the fundamentals of ReactJS including JSX, components, props, and state.
2. To understand React component architecture, lifecycle methods, and rendering techniques.
3. To configure and manage the React development environment using Node.js and npm.
4. To implement event handling, forms, routing, and navigation in React applications.
5. To understand state management techniques using Context API and Redux.
6. To develop modern React applications using Hooks and advanced component patterns.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain ReactJS fundamentals including JSX, components, props, state, and expressions
CO2: Develop React components using functional and class-based approaches and apply lifecycle methods
CO3: Configure and manage React development environment using Node.js, npm, and supporting tools
CO4: Design dynamic user interfaces using conditional rendering, lists, keys, forms, and event handling
CO5: Implement routing and navigation using React Router in Single Page Applications (SPA)
CO6: Manage application state using Context API, Redux, and React Hooks

Unit: I	**Introduction to ReactJS, Conditional Rendering and List and Keys**	**15 Lectures**

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., Conditional Rendering Components: if-else Statement, logical operator and Ternary operator, 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, Lists of Form components, Setup Controlled and Uncontrolled form components, Control Input elements.

Unit: II	**Handling Events and Forms, Routing and State Management**	**15 Lectures**

Events and Forms: Form Submission and Validation, how to set default values on all formats of Input elements, writing Styles, Animations overview, Event, Event Binding, Event Handlers, Common React Events, Key Events, Event Pooling, Synthetic Event. **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.

List of 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

B.C.A – III Semester-VI						
Paper Category:	Practical **(Major)**					
Paper Name:	Practical Based on DSE 1-5					
Practical:	2 Hrs./Week		**Credit:**	1		
Marks:	**UA:**	15	**CA:**	10	**Total:**	25

1	How use Effect 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 a function in JSX?
7	How to access nested object in ReactJS?
8	How to set default value in select using ReactJS?
9	How to solve too many re-renders error in ReactJS?
10	How React Native is different from ReactJS?
11	How to locally manage component's state in ReactJS?
12	How to add a CSS class whenever the component is updated in ReactJS?
13	How to pass data from one component to other component in ReactJS?
14	How to convert functional component to class component in ReactJS?
15	How to put ReactJS component inside HTML string?
16	How to use componentWillMount() in React Hooks?
17	How to use Link Component in ReactJS?
18	How to use BottomNavigation Component in ReactJS?
19	How to use TextField Component in ReactJS?
20	How to use Portal Component in ReactJS?
21	How to use ScopedCssBaseline Component in ReactJS?
22	How to use Popper Component in ReactJS?
23	How to use Slide Component in ReactJS?
24	How to use Collapse Component in ReactJS?
25	How to use Zoom Component in ReactJS?

B.C.A – III Semester-VI						
Paper Category:	VSC4					
Paper Name:	Hands on Training related to DSE1-4					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1.	Study and execute basic Linux commands (pwd, ls, cd)
2.	Create and remove directories using mkdir and rmdir
3.	Perform file operations using cp, mv, and rm
4.	Display file contents using cat, more, and less
5.	Search files and text using find and grep
6.	Change file permissions using chmod, chown, and chgrp
7.	Create and extract archive files using tar and gzip
8.	Display and manage processes using ps and kill
9.	Check disk usage using df and du
10.	Use filter commands (head, tail, sort, uniq, cut, paste)
11.	Redirect input/output and use piping (>, >>, <, `
12.	Create and view hidden files
13.	Use ln to create hard and soft links
14.	Use wc to count lines, words, and characters
15.	Use history and command recall features
16.	Create users and groups
17.	Modify user passwords and group settings
18.	Assign file permissions to users and groups
19.	Mount and unmount storage devices
20.	List running processes and change priority using nice
21.	Write a shell script to print “Hello World”
22.	Write a shell script to perform addition of two numbers
23.	Write a shell script to perform all arithmetic operations
24.	Write a shell script to check whether a number is even or odd
25.	Write a shell script to find the largest of three numbers
26.	Write a shell script to calculate factorial of a number
27.	Write a shell script to check whether a file exists
28.	Write a shell script to count lines, words, and characters in a file
29.	Write a shell script using loop to print numbers 1 to 10
30.	Write a shell script to print multiplication table of a number
31.	Write a shell script to reverse a number
32.	Write a shell script to check palindrome number
33.	Write a shell script to sum digits of a number
34.	Write a shell script to generate Fibonacci series
35.	Write a shell script using case statement (menu-driven program)
36.	Write a shell script using functions for arithmetic operations
37.	Write a shell script to display system information (date, user, uptime)
38.	Write a shell script to list all files in a directory
39.	Write a shell script to monitor disk usage and alert if full
40.	Write a shell script to backup files into a compressed archive

B.C.A – III Semester-VI						
Paper Category:	VSC4					
Paper Name:	Hands on Training related to DSE1-5					
Practical:	4 Hrs./Week		**Credit:**	2		
Marks:	**UA:**	30	**CA:**	20	**Total:**	50

1.	How to add theme to your React App?
2.	How to fetch data from APIs using Asynchronous await in ReactJS?
3.	How to get cell value on React-Table?
4.	How to change the position of the element dynamically in ReactJS?
5.	How to publish a ReactJS component to NPM?
6.	How to change the navbar color when you scroll in ReactJS?
7.	How to Create a Countdown Timer Using ReactJS?
8.	How to Create a Navigation Bar with Material-UI?
9.	How to Create a Toggle Switch in React as a Reusable Component?
10.	How to create a custom progress bar component in React.js?
11.	How to use CssBaseLine Component in ReactJS?
12.	How to use ToggleButtonGroup Component in ReactJS?
13.	How to use Breadcrumbs Component in ReactJS?
14.	How to use Grow Component in ReactJS?
15.	How to use Fade Component in ReactJS?
16.	How to use Popover Component in ReactJS?
17.	How to apply validation on Props in ReactJS?
18.	What is prop drilling and how to avoid it?
19.	How to create new elements with ReactJS mapping props?
20.	How to pass multiple props in a single event handler in ReactJS?
21.	How to create a translucent text input in ReactJS?

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology

EQUIVALENT SUBJECT FOR OLD SYLLABUS

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

Semester-V		
Sr. No.	**Name of the CBCS Paper (w.e.f. 2024 - 25)**	**Name of the New Paper as per NEP2020 (w.e.f. 2026-2027)**
1.	English (Business English)	No Equivalence
2.	Data Communication and Networking	Data Communication and Networking
3.	Dot NET Core	Dot NET Core
4.	Data Warehousing and Data Mining	Data Warehousing and Data Mining (Sem.VI)
5.	Data Science using Python	Data Science using Python
6.	Advanced Java	No Equivalence
7.	ReactJS	ReactJS
Semester-VI		
Sr. No.	**Name of the CBCS Paper (w.e.f. 2024 - 25)**	**Name of the New Paper as per NEP2020 (w.e.f. 2026-2027)**
1.	English (Business English)	No Equivalence
2.	Network Security	No Equivalence
3.	Data Visualization using PowerBI	Data Visualization using PowerBI
4.	ASP.Net Core	ASP.Net Core
5.	Linux and Shell Programming	Linux and Shell Programming
6.	Mobile Application Development using Kotlin	No Equivalence
7.	Internet of things	No Equivalence

For Science faculty: CA- Continuous Assessment (Internal Examinations) of Total Marks: (40%)
The pattern / Examination nature may be as follows:

- One internal examination of 40% marks or two examinations of 20% marks each.
- Open book examination / Home Assignment / Classroom test / Seminar / Field Work report / Project Report etc.