PUNYASHLOK AHILYADEVI HOLKAR SOLAPUR UNIVERSITY, SOLAPUR

Name of the Faculty: **Science and Technology**

NEP 2020

Syllabus: **Entire Computer Science**

Name of the Programme: **B.Sc.[ECS]-III (Sem.– V and VI)**

With Effect From: **June 2026**

B.Sc.[ECS] Part- III Sem.- V										
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		Dot NET Core	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-7	0	0	4	50	30	20	2
	DSC1-8		Theory of Computation	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		Advanced Python for Web and Data Science	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-1	0	0	2	25	15	10	1
	OR									
	DSE1-2		ReactJS	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), IKS, VEC	IKS2		IKS Applications in Emerging Computing Domains	2	0	0	50	30	20	2
Total				**13**	**0**	**18**	**550**	**330**	**220**	**22**

B.Sc.[ECS] Part- III Sem.- VI										
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		ASP.Net Core MVC	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-10	0	0	4	50	30	20	2
	DSC1-11		Advanced Java Programming	3	0	0	75	45	30	3
	Practical		Practical	0	0	4	50	30	20	2

			based on DSC1-11							
	DSC1-12		Data Warehousing and Data Mining	3	0	0	75	45	30	3
	Practical		Practical based on DSC1-12	0	0	4	50	30	20	2
	DSE1-3		Mobile Application Devlopment	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-3	0	0	2	25	15	10	1
	OR									
	DSE1-4		NoSQL Database (MongoDB)	2	0	0	50	30	20	2
	Practical		Practical based on DSE1-4	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 /	FP2/CEP2/OJT1			0	0	4	50	30	20	2

Community Engagement & Services										
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**				

Type: DSC1-7

B.Sc(ECS)-III (Semester V)

Course Title: Dot NET Core

(Paper Code:)

Course Objectives:

I. 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#.

II. To understand the importance of Classes and objects along with constructors, Arrays and Vectors in C#.

III. Discuss the principles of inheritance, interface and demonstrate though problem analysis assignments how they relate to the design of methods, abstract classes and interfaces and packages in C#.

IV. To understand importance of Multi-threading and different exception handling mechanisms in C#.

V. To understand basic idea about how to design GUI base windows application using C#.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: 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.

CO2: Appreciate the significance of classes, objects, constructors, arrays, and vectors within the C# framework.

CO3: 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#.

CO4: Recognize the importance of multi-threading and various exception handling mechanisms in C# programming.

CO5: Acquire a foundational understanding of designing GUI-based Windows applications using C#.

Unit	Content
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.

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-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

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:

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

Type: Practical **B.Sc(ECS)-III (Semester V)** **Course Title: Practical based on DSC1-7** **(Paper Code:)**	
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.	Implement encapsulation, inheritance, and polymorphism concepts in C#.
15.	Use constructors, properties, and methods to define the behavior of objects.
16.	Implement try-catch blocks to handle exceptions and prevent application crashes.
17.	Throw custom exceptions to handle specific error conditions.
18.	Use finally blocks to execute cleanup code regardless of whether an exception is thrown.

19.	Read from and write to text files using StreamReader and StreamWriter classes.
20.	Implement file input/output operations such as reading, writing, appending, and deleting files.
21.	Handle file exceptions and ensure proper resource management using IDisposable interface.
22.	Use LINQ queries to perform filtering, sorting, grouping, and aggregation operations on collections.
23.	Create and manage multiple threads using the Thread class or ThreadPool.

Type: DSC1-8

B.Sc(ECS)-III (Semester V)

Course Title: Theory of Computation

(Paper Code:)

Course Objectives:

I. Understand the fundamental concepts of Formal Languages and Automata Theory.
II. Analyze and design Deterministic and Non-Deterministic Finite Automata and their equivalence.
III. Apply Regular Expressions and study properties and closure properties of Regular Sets.
IV. Understand Context Free Grammars (CFGs), derivations, ambiguity, simplification, and normal forms.
V. Design and analyze Pushdown Automata (PDA) and understand their applications.
VI. Understand the concept of Turing Machines, their computational power, and language acceptance capability.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Construct and analyze Deterministic and Non-Deterministic Finite Automata, including ε-moves, and prove their equivalence.

CO2: Develop and simplify Regular Expressions, verify equivalence with finite automata, and apply closure properties of regular sets.

CO3: Design and analyze Context Free Grammars, construct derivation trees, remove ambiguity, and convert CFGs into normal forms.

CO4: Design Pushdown Automata (PDA) for context-free languages and explain the equivalence between acceptance by final state and empty stack.

CO5: Explain the model of Turing Machine, design simple Turing Machines, and analyze the languages accepted by them.

Unit	Content
I	**Finite Automata and Regular Expression** **20 Lectures**

Finite Automata: Introduction, Deterministic Finite Automata, Non Deterministic Finite Automata, The Equivalence of DFAs and NFAs, Finite Automata with ε- Moves, Equivalence of NFA with ε-Transitions and NFA without Transitions.

Regular Expression: Operations on set of strings, Regular Expression, Regular Sets, Pumping Lemma, Equivalence of finite automata and regular expression, Properties of Regular Sets, Closure properties.

II	**Context Free Grammar, Push-Down Automata and Turing Machine** **25 Lectures**

Context Free Grammar (CFG) Derivation and Language generated by grammar, Derivation Trees, Ambiguity of CFG, Simplification of CFG, Normal forms of CFG, Closure properties of CFG.

Pushdown Automata: Introduction, Definitions, Equivalence of acceptance by final state and empty stack, Definition of DPDA and NPDA, their correlation and examples of NPDA, Application of PDA.

Introduction of Turing Machine: Turing Machine model and definition of TM, language accepted by TM, Design of TM and examples.

List of Reference Books:

I.	J.P. Hopcroft, Rajeev Motwani, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, II Edition, Pearson Education, 2001.
II.	John Martin, Introduction to Languages and Theory of Computation, TataMcGrawHill, 2003.
III.	Daniel I.A., Cohen, Introduction to Computer Theory, 2nd Edition, John Wiley and Sons, Inc, 2000.

Type: Practical

B.Sc(ECS)-III (Semester V)

Course Title: Practical based on DSC1-8

(Paper Code:)

Tool: JFLAP

1.	Design a DFA which accept string is start with a over ∑ = {a, b}.
2.	Design a DFA which accept string starting with a and ending with b over ∑ = {a, b}.
3.	Design a DFA which accept string either ending with ab or bc over ∑ = {a, b, c}.
4.	Design a DFA which accept string does not having abc as substring over ∑ = {a, b, c}.
5.	Design DFA which accept strings length is odd over ∑ = {0,1}.
6.	Draw a DFA over ∑ = {a, b} which containing string start and end with different alphabets.
7.	Draw a DFA ∑ = {0,1} which accept string either ending with 00 or 11.
8.	Draw a DFA ∑ = {a, b} having both 'ab' and 'ba' as a substring .
9.	Design an DFA that reads string made up of letters in the word 'CHRIOT' and recognize those string that contain the word 'CAT 'as a substring.
10.	Draw a DFA which accept string with starting with 'a' and 'abc' as substring in it ∑ = {a, b, c}.
11.	Draw a DFA which accept string starting with a but not having abc as substring in it over ∑ = {a, b, c}.
12.	Draw a DFA which containing string starting with 0 and ending with 2 with 11 as a substring ∑ = {0, 1, 2}.
13.	Design a DFA which contains at most length five over ∑ = {0,1}.
14.	Design a DFA which accept number is divisible by 3.
15.	Draw a DFA over ∑ = {a, b,c } which containing string if it starts with a then it contain abc in it .if it start with b the it ends with c.

16. Draw a DFA which accept string containing third number from right end is 1 ∑ = {0, 1}.

17. Draw a DFA which accept string exactly having two consecutive ones ∑ = {0, 1}.

18. Draw a DFA string ∑ = {0, 1} where even position in string is occupied by 0 and odd position by 1.

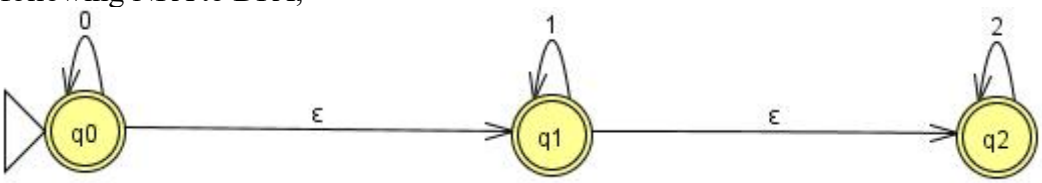
19. Design an NFA which accepts string having abc as a substring ∑ = {a, b, c}.

20. Draw an NFA for string starting with a and ending with b over Σ = {a, b}.

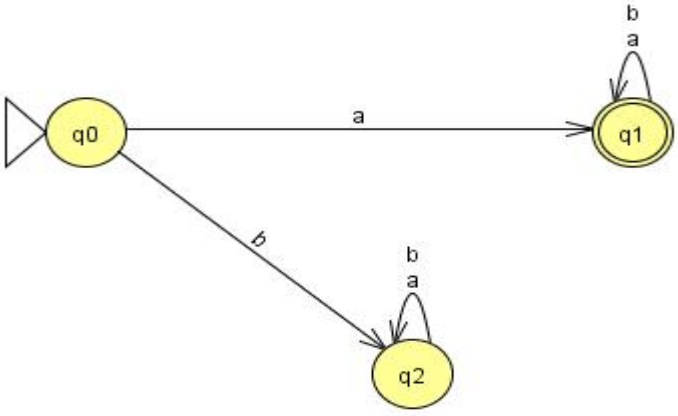
21. Convert following NFA to DFA;

Q / Σ	**0**	**1**
p	{p,q}	{p}
q	{r}	{r}
r	{s}	ϕ
*s	{s}	{s}

22. Convert following NFA to DFA;



23. Find out the Regular Expression for



24. Removal of unit production of following grammar;

I)	II)
A → BB A →a B →b	S→ A S → b A → B A → b

		B → a B → S
25.	Removal ε-production of following grammar;	
	I) S → aSa S → bSb S→ ε	II) S → a S → b S →Ya X → Y X → ε Y→ b Y → ε
26.	Convert following Context Free Grammar(CFG) to Chomsky Normal Form (CNF);	
	I) S→ABAB A→ Aa A→a B→ b	II) S → BA A → aA A → ε B→ bB B → ε
27.	Design a PDA for accepting palindrome string over ∑= {0, 1}.	
28.	Design a PDA to check whether a given string over {a,b} ends in abb.	
29.	Design TM for L = $\{a^n b^n \mid n > 1\}$.	
30.	Construct Turing machine for copy string over ∑ = {a, b}.	
31.	Construct Turing Machine that recognize the language: L = {x ϵ {0, 1} *| x ends in 00}.	

Type: DSC1-9

B.Sc(ECS)-III (Semester V)

Course Title: Artificial Intelligence

(Paper Code:)

Course Objectives:

I. Introduce students to the fundamental concepts of Artificial Intelligence, its applications, and ethical considerations.
II. To understand Problem Solving using various peculiar search strategies for AI.
III. To introduce students to adversarial search strategies used in game-playing AI and familiarize them with constraint satisfaction techniques essential for solving combinatorial problems in A.I
IV. To acquaint with the fundamentals of knowledge and reasoning.
V. To develop a mind to solve real world problems unconventionally with optimality.

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	Content
I	**Introduction to Artificial Intelligence and Problem Solving** **25 Lectures**

Introduction to the fundamentals of AI, history, applications, and challenges, Applications and Impact of AI in Various Domains, Types of AI: Narrow AI, General AI, and Super AI, Intelligent Agents, Agents and Environments, Concept of Rationality, Nature of Environments, Structure of Agents. AI in Problem Solving: State Space Representation, AI Ethics and Challenges Solving Problems by Searching, Problem-Solving Agents, Example Problems, Search Algorithms, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions, Search in Complex Environments, Local Search and

Optimization Problem Game Theory, Optimal Decisions in Games, Heuristic Minimax algorithm, Alpha–Beta Tree Search, Monte Carlo Tree Search, Stochastic Games, Partially Observable Games, Limitations of Game Search Algorithms, Constraint Satisfaction Problems (CSP), Constraint Propagation: Inference in CSPs, Backtracking Search for CSPs.

II	**Knowledge Representation**	**20 Lectures**

Logical Agents, Knowledge-Based Agents, The Wumpus World problem, Propositional Logic: A Very Simple Logic, Propositional Theorem Proving, Effective Propositional Model Checking, Agents Based on Propositional Logic, First-Order Logic, Representation Revisited, Syntax and Semantics of FirstOrder Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic. Inference in First-Order Logic, Propositional vs. First-Order Inference, Unification and First-Order Inference, Forward Chaining, Backward Chaining, Resolution, Knowledge Representation.

Automated and Classical Planning, Hierarchical Planning, Planning under Uncertainty, Analysis of Planning Approaches, Limits of AI, Ethics of AI, Future of AI, AI Components, Introduction of Generative AI, Explainable AI (XAI), Federated Learning, Edge AI.

List of Reference Books:

I.	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
II.	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
III.	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. Dr.Nilakshi Jain,“Artificial Intelligence,As per AICTE: Making a System Intelligent”,Wiley publication, ISBN: 9788126579945
IV.	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. Dr.Nilakshi Jain,“Artificial Intelligence,As per AICTE: Making a System Intelligent”,Wiley publication, ISBN: 9788126579945
V.	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.

Type: Practical **B.Sc(ECS)-III (Semester V)** **Course Title: Practical Based on DSC1-9** **(Paper Code:)**	
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.	Case Studies : Google DeepMind – AI for Energy Efficiency in Data Centers and Game Playing (e.g., AlphaGo)
20.	Case Studies :BBC & Amazon Alexa – AI-Driven Interactive Media with Logic-Based Chatbots
21.	Case Studies: OpenAI's GPT-4 & ChatGPT – Impacts of Generative AI and Prompt Engineering.

Type: DSE1-1

B.Sc(ECS)-III (Semester V)

Course Title: Advanced Python for Web and Data Science

(Paper Code:)

Course Objectives:

I. To learn Django web application devlopment.
II. To learn Django object relational mapping.
III. To study database connectivity of Django web application.
IV. To study Django web framework.
V. To learn basic libraries used in data Science.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Design, Create, Build, and Debug Python web applications.

CO2: Explore Integrated Development Environment (IDE) such as PyCharm, VSCode.

CO3: To create RestAPI using DRF.

CO4: To make Data Analysis and visualization.

Unit	Content
I	**Web development using Django** **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 and 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.

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, Stages in a Data Science Project.

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:

I.	Python Web Development with Django, Jeff Forcier, Paul Bissex, Wesley J Chun.
II.	Django: Web Development with Python, Samuel Dauzon, Aidas Bendoraitis, Arun Ravindran
III.	Web Development with Django, Ben Shaw, Saurabh Badhwar, Chris Guest, Bharath Chandra KS
IV.	The Definitive Guide to Django, Jacob Kaplan-Moss, Adrian Holovaty.
V.	An Introduction to Data Science, Jeffrey S. Saltz, Jeffrey M. Stanton
VI.	Data Science for Beginners: Introduction to Data Science, Prof John Smith
VII.	Introduction to Data Science A Python Approach to Concepts, Techniques and Applications, Laura Igual, Santi Segui, Jordi Vitria.

Type: Practical B.Sc(ECS)-III (Semester V) Course Title: Practical based on DSE1-1 (Paper Code:)	
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

Type: DSE1-2

B.Sc(ECS)-III (Semester V)

Course Title: ReactJS

(Paper Code:)

Course Objectives:

I. To introduce the fundamentals of ReactJS including JSX, components, props, and state.
II. To understand React component architecture, lifecycle methods, and rendering techniques.
III. To configure and manage the React development environment using Node and NPM.
IV. To implement event handling, forms, routing, and navigation in React applications.
V. To understand state management techniques including Context API and Redux.
VI. 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 implement lifecycle methods.

CO3: Configure React development environment using Node, NPM, and required 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 architecture, and React Hooks such as useState, useEffect, useReducer, and useContext.

Unit	Content
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.

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:

I.	Introduction to React By Cory Gackenheimer, Apress
II.	React and React Native: A complete hands-on guide to modern web and mobile development with React.js By Adam Boduch, Roy Derks
III.	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

Type: Practical **B.Sc(ECS)-III (Semester V)** **Course Title: Practical Based on DSE1-2** **(Paper Code:)**	
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?
26.	How to send state/props to another component in React with onClick?
27.	How to set an object key inside a state object in React Hooks?

Type: DSE1-3

B.Sc(ECS)-III (Semester V)

Course Title: Cyber Security

(Paper Code:)

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

.

Type: VSC3 **B.Sc(ECS)-III (Semester V)** **Course Title: Hands on Training related to DSE1-1** **(Paper Code:)**	
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.

Type: VSC3 B.Sc(ECS)-III (Semester V) Course Title: Hands on Training related to DSE1-2 (Paper Code:)	
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?

Type: VSC3

B.Sc(ECS)-III (Semester V)

Course Title: Hands on Training related to DSE1-3

(Paper Code:)

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

Type: IKS2

B.Sc(ECS)-III (Semester V)

Course Title: IKS Applications in Emerging Computing Domains

(Paper Code:)

Course Objectives:

I. To provide an in-depth understanding of Indian Knowledge Systems (IKS)
II. To explore theoretical and practical connections between IKS and computing
III. To analyze how ancient Indian knowledge inspires modern and emerging technologies
IV. To promote ethical, sustainable, and indigenous computing perspectives

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Understand foundational concepts of Indian Knowledge Systems

CO2: Explain mathematical and logical traditions in IKS

CO3: Analyze IKS applications in AI and emerging computing domains

CO4: Apply IKS concepts to real-world technological problems

CO5: Appreciate ethical and sustainable computing practices

Unit	Content
I	**IKS and Artificial Intelligence** **15 Lectures**
Indian perspectives on intelligence, cognition, and consciousness, Mind (Manas), intellect (Buddhi), memory (Smriti) concepts, Knowledge representation in IKS, Rule-based reasoning and expert systems inspired by IKS, Paninian grammar and its influence on NLP, Learning concepts in Indian philosophy and ML analogy, Ethical AI from Indian philosophical traditions.	
II	**IKS in Emerging Computing Domains** **15 Lectures**
KS-inspired ideas in Machine Learning and pattern recognition, Swarm intelligence and collective behavior in Indian texts, Yoga, meditation, and computational neuroscience (conceptual), IKS in Human–Computer Interaction (HCI), Ancient cryptography and secure communication systems, IKS perspectives in Quantum Computing, Indigenous technological innovations, Applications in healthcare, education, agriculture, and sustainability.	

List of Reference Books:	
I.	UGC – Indian Knowledge Systems (IKS) Documents
II.	Subhash Kak – Computing the Mind
III.	Fritj of Capra – Systems Thinking

Type: DSC1-10

B.Sc(ECS)-III (Semester VI)

Course Title: ASP .Net Core MVC

(Paper Code:)

Course Objectives:

I. Demonstrate the creation of ASP.NET Core MVC Web Applications using .NET 8, covering project file structure, main method, hosting options, and configuration files.
II. Cover Models, Controllers, Views, and Dependency Injection in ASP.NET Core MVC applications.
III. Introduce Entity Framework Core and guide the installation process, also explain DbContext in Entity Framework Core and database connection string configuration and database operations.
IV. Discuss Transactions, Migration, and Database Seeding in Entity Framework Core.
V. Introduce Partial Views, View Components, and Razor View Engine.
VI. Cover Action Results, Routing, Model Binding, HTML Helpers and Tag Helpers in ASP.NET Core MVC.

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 MVC applications.

CO7: Data Annotations and Model Validations, including custom validations and remote validation.

CO8: Discuss different methods of State Management, including Cookies and Sessions.

Unit	Content
I	**Introduction to ASP.Net Core MVC and Entity Framework Core** **20 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.

II	**Model, View, Controller and Routing, HTML, Tag Helper, Data Annotation Validation and State management to Django** **25 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:

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

Type: Practical **B.Sc(ECS)-III (Semester VI)** **Course Title: Practical Based on DSC1-10** **(Paper Code:)**	
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.

Type: DSC1-11

B.Sc(ECS)-III (Semester VI)

Course Title: Advanced Java Programming

(Paper Code:)

Course Objectives:

I. Understand database connectivity using JDBC and perform CRUD operations.
II. Develop dynamic web applications using Servlets and manage sessions.
III. Design server-side applications using Java Server Pages (JSP) and JSTL.
IV. Understand and implement the Hibernate ORM framework for database interaction.
V. Apply Spring Framework concepts, including dependency injection and JDBC template integration.
VI. Integrate database and web technologies to develop enterprise-level Java applications.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Establish database connectivity using JDBC, handle exceptions, and perform Insert, Update, Delete, and Retrieve operations using Statement, PreparedStatement, and CallableStatement.

CO2: Develop dynamic web applications using Servlets, explain servlet lifecycle, manage sessions using cookies, URL rewriting, and hidden form fields.

CO3: Create JSP-based web pages using implicit objects, scripting elements, JSTL tags, JavaBeans, and implement internationalization and exception handling.

CO4: Develop database-driven applications using Hibernate, including annotation-based configuration, inheritance mapping, and collection mapping.

CO5: Implement enterprise applications using Spring Framework, apply dependency injection, integrate Spring with JDBC and Hibernate.

Unit	Content	
I	**Database Connectivity, Networking and Servlet Programming**	**20 Lectures**

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.

Networking: Basics of Networking, Socket Programming in Java Networking, using java.net package ServerSocket Class Constructosr and Methods, Socket Class Constructors and Methods, InetAddress Class Methods, Implementing Socket Client and Server Programming.

Servlet: Introduction to Servlet, Features of Servlet, javax.servlet package, javax.servlet.http package, Servlet life Cycle, Working with Generic Servlet and HttpServlet, Request Dispatcher interface, Include() and forward() methods, Use of Request Dispatcher, Servlet Session and Cookies, Hidden form fields, URL rewritten.

II	**Java Server Pages, Hibernate and Spring**	**25 Lectures**

JSP: Introduction to JSP, Advantages of JSP over Servlet, JSP architecture, JSP life cycle, Implicit objects in JSP, JSP tag elements- Declarative, Declaration, Scriplet, Expression and Action. Use-Bean tag- setProperty and getProperty, Advantages of Java Beans, JSTL core tag- General purpose tag, conditional tag, networking tag, JSTL SQL tags, JSTL formatting tags and JSTL XML tags. Custom tags in JSP- empty tag, body content tag, iteration tag, simple tag. Introducing internationalization and Java-local class, ResourceBundle class, Handling different errors and Exceptions.

Hibernate: Introduction to Hibernate, Architecture of HB, HB application, HB application using annotation, inheritance mapping, collection mapping.

Spring: Introduction to Spring, Spring module, application of Spring, dependency injection- constructor injection and setter injection, simple spring JDBC template, HB with spring.

List of Reference Books:

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

Type: Practical B.Sc(ECS)-III (Semester VI) Course Title: Practical based on DSC1-11 (Paper Code:)	
1.	Write a JDBC program to implement CRUD operations
2.	Write a JDBC program to implement Prepared statement.
3.	Write a JDBC program to implement Callable statement.
4.	Write a Network program to demonstrate InetAddress Class.
5.	Write a Network Program to Create Socket for Client Server Communication.
6.	Write a program which demonstrates life cycle of Servlet
7.	Write a program by using GenericServlet
8.	Write a program by using HttpServlet
9.	Write a Servlet program to send request to another page
10.	Write a Servlet program to track the user by using (Cookies, URL rewriting, Hidden form field and HttpSession)
11.	Write Jsp program which will display its life cycle
12.	Write a Jsp program by using its implicit objects like request, response, out, page, pageContext,application, session, config, exception
13.	Write a Jsp program which will use scriplet, expression and declarative tag.
14.	Write a Jsp program which will create bean and calculate simple interest
15.	Write a Jsp program to create bean to check account balance(from database)
16.	Write a Jsp program to insert, update and delete data into database.
17.	Write a Jsp program which will use JSTL core tag, JSTL SQL tags, JSTL formatting tags, JSTL xml tags, Customtag: empty tag, body content tag,iteration tag, simple tag
18.	Write a program to display a message in different languages (use java internationalization)

19.	Write a simple Hibernate program.
20.	Write a HB with annotation.
21.	Write a HB web application.
22.	Write a HB Inheritance mapping: Table per Hierarchy(TPH), TPH using annotation,
23.	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.
24.	Write a Spring program to show Dependency injection: constructor Injection (CI),CI dependant object, CI with collection, CI with map, CI inheriting bean
25.	Write a Spring Spring JDBC program using : JDBC template, Result set Exactor, Named Parameter, Simple JDBC template. Spring with Hibernate

Type: DSC1-12

B.Sc(ECS)-III (Semester VI)

Course Title: Data Warehouseing and Data Mining

(Paper Code:)

Course Objectives:

I. To understand concepts and architecture of Data Warehousing
II. To study data preprocessing and OLAP operations
III. To learn core data mining techniques and algorithms
IV. To apply data mining for knowledge discovery and decision making

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain concepts and architecture of data warehousing

CO2: Design dimensional models and perform OLAP operations

CO3: Apply data preprocessing techniques

CO4: Implement data mining algorithms for classification and clustering

CO5: Analyze real-world data mining applications

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

Introduction: 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.

II	**Data Preprocessing and Data Mining Techniques**	**25 Lectures**

Data Preprocessing: An Overview, 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

Classification concepts, Decision Tree classification, Naive Bayes classifier, K-Nearest Neighbor (KNN), Clustering concepts, k-Means clustering algorithm, Hierarchical clustering, Outlier analysis, Data mining applications.

List of Reference Books:

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

Type: Practical

B.Sc(ECS)-III (Semester VI)

Course Title: Practical Based on DSC1-12

(Paper Code:)

Tools: Weka / Python

1. How to 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)).

Table: Class-Labeled Training Tuples from the All Electronics Customer Database

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(Name it as house_selling_rate.arff).

House size (square 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	????

	Calculate the selling price of last house by using regression model.
5.	Load the soybean.arff file in weka, and use the classification method (use J48), analyse the confusion matrix, visualize the decision tree.
6.	Practical's on Mining Associations Rule
7.	Practical's on Classifications and predictions
8.	Practical's on Clustering
9.	Load the weather.nominal dataset. Use the following filters in weka. I. unsupervised.instance.RemoveWithValues to 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
10.	Load the iris dataset. Use the following filter of weka. I. Add noise to last column (i.e. Class). II. Randomize the data III. Normalize the data IV. 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.	Implement and evaluate clustering algorithms available in Weka (e.g., K-Means, Hierarchical Clustering).
15.	Explore different clustering techniques and their impact on clustering quality.
16.	Visualize the clusters using Weka's visualization tools and analyze their characteristics.

17.	Apply association rule mining algorithms (e.g., Apriori) to identify interesting patterns in the dataset.
18.	Adjust parameters such as minimum support and confidence to control the quality of the discovered rules.
19.	Interpret and analyze the discovered rules to gain insights into the dataset.
20.	Implement feature selection techniques to identify the most informative features in the dataset.
21.	Evaluate the impact of feature selection and dimensionality reduction on the performance of classification or clustering algorithms.

Type: DSE1-3

B.Sc(ECS)-III (Semester VI)

Course Title: Mobile Application Devlopment

(Paper Code:)

Course Objectives:

I. To introduce the fundamentals of Android, its versions, features, devices, and application ecosystem.
II. To understand Android architecture including Linux Kernel, Android Runtime (Dalvik VM), libraries, and application framework.
III. To set up the Android development environment using Java, Android SDK, IDE, and AVD.
IV. To develop Android applications using core components such as Activities, Services, Broadcast Receivers, Content Providers, and Intents.
V. To implement data storage techniques including files, SQLite databases, and content providers.
VI. To design user-friendly Android interfaces and integrate multimedia, messaging, and location-based services into applications.

Course Outcomes (COs):

After completing this course, students will be able to:

CO1: Explain Android architecture, versions, features, and the Android development environment.

CO2: Configure Android SDK, IDE, and create Android Virtual Devices (AVDs) for application testing.

CO3: Develop Android applications using core components such as Activities, Services, Broadcast Receivers, Content Providers, and Intents.

CO4: Design interactive user interfaces using Views, View Groups, Layout Managers, XML resources, and screen adaptation techniques.

CO5: Implement data management using file handling, SQLite databases, and content providers.

CO6: Create Android applications integrating SMS, email, multimedia (audio/video/camera), animations, and location-based services.

Unit	Content
I	**Introduction Android Software Development Platform and Framework 15 Lectures**

What is Android, Android Versions and its Feature Set, Various Android Devices on the Market, Android Market Application Store, Android Development Environment, System Requirements, Android SDK, Installing Java, and ADT bundle - Android Studio, Integrated Development Environment (IDE), Creating Android Virtual Devices (AVDs).

Android Architecture Overview and Application: Android Architecture, Android Runtime - Dalvik Virtual Machine, Core Libraries, Dalvik VM Specific Libraries, Java Interoperability Libraries, Application Framework. Creating a New Android Project, Defining the Project Name and SDK Settings, Project Configuration Settings, Configuring the Launcher Icon, Creating an Activity, Running the Application on AVD, Reviewing the Layout and Resource Files.

Understanding Java SE and the Dalvik Virtual Machine, The Directory Structure of an Android Project, Common Default Resources Folders, The Values Folder, Screen Sizes , Launching Mobile Application: The AndroidManifest.xml File, Android Application Components- Android Activities: Defining the UI, Android Services, Broadcast Receivers, Content Providers, Android Intent Objects: Messaging for Components.

II	**Understanding Android User Interfaces, Views and Layouts, Databases, Intents, Location-based Services and Development 15 Lectures**

Designing for Different Android Devices, Views and View Groups, Android Layout Managers, The View Hierarchy, Designing an Android User Interface using the Graphical Layout Tool, Displaying Text with Text View, Retrieving Data from Users, Using Buttons, Check Boxes and Radio Groups, Getting Dates and Times from Users, Using Indicators to Display Data to Users, Adjusting Progress with Seek Bar, Working with Menus using views, Gallery, Image Switcher, Grid View, and Image View views to display images, Creating Animation.

Saving and Loading Files in SQLite Databases, Android Database Design, Exposing Access to a Data Source through a Content Provider, Content Provider Registration, Native Content Providers Intents and Intent Filters: Intent Overview, Implicit Intents, Creating the Implicit Intent Example Project, Explicit Intents, Creating the Explicit Intent Example Application, Intents with Activities, Intents with Broadcast Receivers.

Application Development: Sending SMS Messages Programmatically, Getting Feedback after Sending the Message, Sending SMS Messages Using Intent Receiving, Sending email, Introduction to location-based service, Configuring the Android Emulator for Location -Based Services, Geocoding and Map-

Based Activities Multimedia: Audio, Video, Camera: Playing Audio and Video, Recording Audio and Video, Using the Camera to Take and Process Pictures.	
List of Reference Books:	
I.	Android Programming Unleashed (1st Edition) by Harwani.
II.	Beginning Mobile Application Development in the Cloud (2011), Richard Rodger.
III.	Learn Android App Development by Wallace Jackson.
IV.	Professional Android Application Development by Reto Meier.

Type: Practical **B.Sc(ECS)-III (Semester VI)** **Course Title: Practical Based on DSE1-3** **(Paper Code:)**	
1.	Develop an application that uses GUI components, Font and Colours.
2.	Develop an application that uses Layout Managers and event listeners.
3.	Develop a native calculator application.
4.	Write an application that draws basic graphical primitives on the screen.
5.	Develop an application that makes use of database.
6.	Develop an application that makes use of RSS Feed.
7.	Implement an application that implements Multi-threading
8.	Develop a native application that uses GPS location information.
9.	Implement an application that writes data to the SD card.
10.	Implement an application that creates an alert upon receiving a message.
11.	Write a mobile application that creates alarm clock.

Type: DSE1-4

B.Sc(ECS)-III (Semester V)

Course Title: NoSQL Database (MongoDB)

(Paper Code:)

Course Objectives:

I. To introduce the fundamental concepts of NoSQL databases and MongoDB.
II. To understand the advantages of MongoDB over traditional RDBMS.
III. To learn MongoDB data types, installation process, and data modeling techniques.
IV. To perform database and collection operations in MongoDB.
V. To implement CRUD operations and advanced query mechanisms.
VI. To apply aggregation framework and indexing techniques for query optimization.
VII. To understand MongoDB security features including authentication and authorization.
VIII. To explore replication, sharding, and scalability concepts in MongoDB.
IX. To perform bulk operations and advanced command executions.
X. 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, data modeling, and installation procedures.

CO2: Apply various query, projection, and update operators to interact with MongoDB data.

CO3: Perform operations such as create, drop, and list collections and databases.

CO4: Use advanced features such as text search, upserts, multi-update, and wire protocol methods.

CO5: Connect MongoDB with programming languages like Java, PHP, and Python to implement real-world integration scenarios.

Unit	Content
I	**MongoDB Fundamentals and Core Operations** **20 Lectures**

Introduction: Inrodution 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, Upsert, Wire Protocol, Bulk Operations, Common Commands

II	**Advanced MongoDB, Commands and Connectivity**	**10 Lectures**

MongoDB Shell and Connectivity: MongoDB Shell, Collection Methods, Cursor Methods, Database Commands, Query Plan Cache Methods, User & Role Management, Replication Methods, Connectivity with Java, PHP, Python

List of Reference Books:

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

Type: Practical B.Sc(ECS)-III (Semester VI) Course Title: Practical based on DSE1-4 (Paper Code:)	
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.	Setting up Replica Set (Replication concept demo).

Type: VSC4 B.Sc(ECS)-III (Semester VI) Course Title: Hands on Training related to DSE1-3 (Paper Code:)	
1.	Installation of Java (JDK), Android SDK, and IDE.
2.	Creating and configuring Android Virtual Devices (AVDs).
3.	Creating a new Android project and understanding project structure.
4.	Developing and running a Hello World application.
5.	Create an Android applications to demonstrate activity life cycle.
6.	Working with AndroidManifest.xml file.
7.	Creating an application using Activities and XML layouts.
8.	Working with TextView, EditText, and Button controls.
9.	Implementing CheckBox, RadioButton, Date and Time Picker.
10.	Designing layouts using LinearLayout and RelativeLayout.
11.	Implementing Explicit and Implicit Intents.
12.	Creating an Android application to demonstrate Android Services.
13.	Creating an Android application to demonstrate Broadcast Receivers.
14.	Creating an Android application to demonstrate Content Provider.
15.	Creating and managing SQLite database using CRUD operations.
16.	Sending SMS programmatically and receiving feedback.
17.	Sending Email using Intent.
18.	Developing a Location-Based application using Maps and Geocoding.
19.	Creating an Android application to Play Audio and Video files.

20.	Working with Menus, GridView, Gallery, and ImageView.
21.	Creating an Android application to demonstrate animations in Android.

Type: VSC4 B.Sc(ECS)-III (Semester VI) Course Title: Hands on Training related to DSE1-4 (Paper Code:)	
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.

Punyashlok Ahilyadevi Holkar Solapur University, Solapur

Faculty of Science and Technology

EQUIVALENT SUBJECT FOR OLD SYLLABUS

Name of the Programme: B.Sc. (ECS) - 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	No Equivalence
3.	Dot NET Core	Dot NET Core
4.	Advanced Python Programming	No Equivalence
5.	Theory of Computation	Theory of Computation
6.	Mobile Application and Development	Mobile Application and Development (Sem-VI)
7.	Artificial Intelligence	Artificial Intelligence
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 Warehousing and Data Mining	Data Warehousing and Data Mining
4.	ASP.Net Core MVC	ASP.Net Core MVC
5.	React JS	React JS (Sem.-V)
6.	Compiler Construction	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.