Mahatma Education Society's

Pillai College of Arts, Commerce & Science (Autonomous)

Affiliated to University of Mumbai

'NAAC Accredited 'A' grade (3 cycles)'
'Best College Award' by University of Mumbai
ISO 9001:2015 Certified

SYLLABUS

Program: Bachelors of Science (B. Sc.) in Information Technology

S.Y.B.Sc.Information Technology

PCACS/BSCIT/SYL/2024-25/SY

As per National Education Policy
Choice Based Credit & Grading System

Academic Year 2024-25

Mahatma Education Society's

Pillai College of Arts, Commerce & Science

(Autonomous)

Affiliated to University of Mumbai

NAAC Accredited 'A' grade (3 cycles)

Best College Award by University of Mumbai

ISO 9001:2015 Certified

5,April 2024

Attendance for BOS Meeting

Board of Studies in the Department of Information Technology

Sr. No.	Name of the	Details	Sign
1	Prof. Deepika Sharma	Chairperson (Head of Department of Information Technology & Computer Science), Vice Principal	
2	Dr. Gajanan Wader	Principal	
3	Mrs.Munawira Kotyad Founder and CEO Wonderwheel Enterprises, Pillai, Director Pillai Center for Innovation & Research	Faculty Specialization	Absent
4	Dr. Amiya Kumar Tripathy Director Center for GeoAI & ML, Professor, Computer Engineering, Don Bosco Institute of Technology, Mumbai	Subject Expert From Outside Parent University	
5	Dr.Mrs. Anjali Kulkarni CKT College, New Panvel	Vice Chancellor Nominee, University of Mumbai	
6	Mr. Tito Idicula, Director, Programming Hub	Alumni representative	
7	Mr. Anant Baddi, Security Solution Architect, cloud Google Google	Industry Representative (Industry/Corporate/Allied Sector)	Absent
8	Mr. Bhupendra Kesariya Professor,N. M. .College, Vile Parle	Subject Expert in Mathematics From Outside Parent University	

9	Mrs. Anju Somani	Faculty Specialization	Somani
10	Mrs. Shubhangi Pawar	Faculty Specialization	Pawar
11	Dr. Kumudini Das	Faculty Specialization	K Das
12	Mrs. Soly Zachariah	Faculty Specialization	
13	Mrs. Ramya S. Kumar	Faculty Specialization	
14	Mrs. Sujata Shahabade	Faculty Specialization	Sujata
15	Mrs. Sreevidya T.V.	Faculty Specialization	
16	Mr. Omkar Sherkhane	Faculty Specialization	
17	Mr. Abhijeet Salvi	Faculty Specialization	

1. INTRODUCTION TO BACHELORS IN INFORMATION TECHNOLOGY (I.T.) PROGRAM

B.Sc. in Information Technology is a three years undergraduate programme that has been designed meticulously to meet the requirements of dynamic I.T. industry. This programme aims at fostering concepts of Information technology in students and equip them with the required technical, logical, problem solving and soft skills, which prepare them for the corporate world. It also focuses on inculcating effective communication skills which a software professional must have.No education is complete without incorporating social and moral values. This programme takes care of this aspect as well. The core courses of the program are supplemented by electives so that students can tailor the program according to their interest. State of art computer laboratories, in the campus, help students to practically implement the concepts learned. Qualified and experienced faculty members guide students in their project work. As we all know degrees in I.T. lead to rewarding and lucrative careers, excellent placement and incubation assistance is provided.

2. Program outcomes

Sr No	PO Title	POs in brief
PO1	**Core Knowledge**	Develop a strong foundation in the core principles and theories of their chosen field of study to pursue a profession of choice by understanding fundamental concepts, methodologies, and key terminologies
PO2	**Research Skills**	Trigger the research aptitude by developing basic research skills, including the ability to conduct literature reviews, design experiments, collect and analyze data, and draw meaningful conclusions.
PO3	**Communication Skills**	Communicate scientific concepts, experimental results and analytical arguments clearly and concisely, both verbally and in writing and also ability to present their work through written, oral, and visual presentations, including an original research proposal in a clear and understandable manner to both technical and non-technical audiences.
PO4	**Ethical and Professional Behavior**	Understand and adhere to ethical standards by recognizing the importance of integrity, honesty and ethical responsibility in scientific research and professional practice.
PO5	**Teamwork and Collaboration**	Ability to work cohesively to achieve common goals, solve problems and contribute to the success of a project or task paving way to individual and collective growth.
PO6	**Adaptability and Lifelong Learning**	Engage themselves in lifelong learning to keep up with the pace of changing technology and interdisciplinary approach to provide better solutions and new ideas for the sustainable developments
PO7	**Technical Skills**	Acquisition of specialized technical skills and expertise relevant to the specific field of study i.e advanced laboratory techniques, computational skills, or other specialized methodologies.
PO8	**Critical Thinking and Problem-Solving Skills**	Graduates would be equipped with the ability to analyze information critically, think logically, and solve complex problems. Applying scientific methods, mathematical reasoning, and logical approaches to real-world situations.

3. Program Specific Outcomes

Sr No	PSOs in brief
PSO1	Possess skill sets in information management, networking, web designing, mobile app development, Database management, programming and testing.
PSO2	Effectively integrate I.T. based solution in the users domain after properly analyzing the requirements and the constraints.
PSO3	Develop an ability to use appropriate techniques, skills and tools required for computing problems
PSO4	Ability to comprehend and write effective project reports in a multidisciplinary environment in the context of changing technologies.

Course structure

Semester III						
Course Code	**Course Type**	**Course Title**	**Theory/ Practical**	**Marks**	**Credits**	**Lectures/ Week**
PUSIT301	**MAJ**	Operation Research	Theory	100	2	4
PUSIT302	**MAJ**	Advanced Java	Theory	100	2	4
PUSIT303	**MAJ**	Advanced Database System	Theory	100	2	4
PUSIT304D/ PUSIT304U/ PUSIT304U	**DISCMIN**	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3 Full Stack Development paper-1	Theory/ Practical	100	3	6
PUSIT305	**SEC**	Mini Project	Theory/ Practical	100	2	3
PUSIT 306P	**MAJ**	Practicals(PUSIT301)	Practical	50	2	2
PUSIT 307P	**MAJ**	Practicals(PUSIT302)	Practical	50	2	2
PUSIT 308P	**MAJ**	Practicals(PUSIT303)	Practical	50	2	2
PUAEC30	**AEC**	Languages (To be taken from pool)	Practical	100	2	3(online)
PUIDC30	**IDC**	To be taken from pool	Theory/ Practical	100	3	6
Total				850	22	27
All Subjects having Field Project as part of Continuous Assessment-2						

Abbreviations:

IDC: Interdisciplinary Course

AEC : Ability Enhancement Course

SEC : Skill Enhancement Course

Course Structure

Semester- IV

Course Code	Course Type	Course Title	Theory/ Practical	Marks	Credits	Lectures/ Week
PUSIT401	**MAJ**	Advanced Web Programming	Theory	100	2	4
PUSIT402	**MAJ**	Operating System	Theory	100	2	4
PUSIT403	**MAJ**	Data Structure using Python	Theory	100	2	4
PUSIT404	**DISCMIN**	Track1-Data Science Introduction to Data Science Track2-UI-UX UI-UX Design Track-3 Full Stack Development paper-2	Theory/ Practical	100	3	6
PUSIT405	**SEC**	Swayam course (NPTEL)	Theory	100	2	online
PUSIT 406P	**MAJ**	Practicals(PUSIT401)	Practical	50	2	2
PUSIT 407P	**MAJ**	Practicals(PUSIT402)	Practical	50	2	2
PUSIT 408P	**MAJ**	Practicals(PUSIT403)	Practical	50	2	2
PUAEC40	**AEC**	To be taken from pool	Theory	100	2	3
PUIDC40	**IDC**	To be taken from pool	Theory/ Practical	100	3	6
Total				850	22	27
All Subjects having Field Project as part of Continuous Assessment-2						

Abbreviations:

IDC: Interdisciplinary Course

AEC : Ability Enhancement Course

SEC : Skill Enhancement Course

Evaluation Pattern

Marking Code	Marking Scheme
A	60 Marks Final Exam, 20 Marks Internal Exam, 20 Marks Project.
B	60 Marks Final Exam, 40 Marks Internal Exam.
C	100 marks distributed within report /case study/ project/ presentation etc.
D	50 Marks Practical Examination.(10 marks viva,10 marks Journal,30 marks Program Executions)

SEMESTER III

Course Code	Course Type	Course Title	Evaluation Pattern
PUSIT301	**MAJ**	Operation Research	A
PUSIT302	**MAJ**	Advanced Java	A
PUSIT303	**MAJ**	Advanced Database System	A
PUSIT304D/ PUSIT304U/ PUSIT304F	**DISCMIN**	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3 Full Stack Development paper-1	C
PUSIT305	**SEC**	Mini Project	C
PUSIT 306P	**MAJ**	Practicals(PUSIT301)	D
PUSIT 307P	**MAJ**	Practicals(PUSIT302)	D
PUSIT 308P	**MAJ**	Practicals(PUSIT303)	D
PUAEC30	**AEC**	Languages (To be taken from pool)	C
PUIDC30	**IDC**	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3	C

		Full Stack Development paper-1l	

SEMESTER IV

Course Code	Course Type	Course Title	Evaluation Pattern
PUSIT401	**MAJ**	Advanced Web Programming	A
PUSIT402	**MAJ**	Operating System	A
PUSIT403	**MAJ**	Data Structure using Python	A
PUSIT404	**DISCMIN**	Track1-Data Science Introduction to Data Science Track2-UI-UX UI-UX Design Track-3 Full Stack Development paper-2	C
PUSIT405	**SEC**	Swayam course (NPTEL)	C
PUSIT 406P	**MAJ**	Practicals(PUSIT401)	D
PUSIT 407P	**MAJ**	Practicals(PUSIT402)	D
PUSIT 408P	**MAJ**	Practicals(PUSIT403)	D
PUAEC40	**AEC**	To be taken from pool	C
PUIDC40	**IDC**	Track1-Data Science Introduction to Data Science Track2-UI-UX UI-UX Design Track-3 Full Stack Development paper-2	C

SEMESTER III

BOS	Mathematics and Statistics
Class	S.Y. B.Sc. I.T.
Semester	III
Course Name	Operation Research
Course Code	PUSIT301
Course Type	Major
Level of the Course	Medium
Credits	2 Theory + 2 Practical

Course Objectives:

1. Knowledge of formulating mathematical models for quantitative analysis of managerial problems in industry.
2. Ability to understand and analyse managerial problems in industry so that they are able to use resources (capitals, materials, staffing, and machines) more effectively.

UnitNo.	Name of Unit	Topic No.	Name of the Topic	Hours
1	Linear Programming Problem	1.1	Introduction To Operations Research	10
		1.2	Linear Programming Problems: Introduction and Formulation - Decision Variables, Objective Function, Constraints, Non Negativity Constraints	
		1.3	Linear Programming Problems: Graphical Method Maximization & Minimization Type Problems - Two Decision Variables and Maximum Three Constraints Problem.	
		1.4	Programming: Linear Programming Problems: Simplex Method - Only Maximization Type Problems - Two or Three Decision Variables and Maximum Three Constraints Problem.	
		2.1	Assignment Problem – Hungarian Method Maximization & Minimization Type Problems. Balanced and Unbalanced Problems	
		2.2	Prohibited Assignment Problems, Unique or Multiple Optimal Solutions. Maximum 5 x 5 Matrix. Up to Maximum Two Iterations after Row and Column Minimization.	
		2.3	Transportation Problems- Maximization & Minimization Type Problems. Balanced and Unbalanced problems. Prohibited	

2	Transportation and Assignment		Transportation Problems, Unique or Multiple Optimal Solutions.	10
			Initial Feasible Solution (IFS) by: a. North West Corner Rule (NWCR) b. Least Cost Method (LCM) c. Vogel's Approximation Method (VAM). Maximum 5 x 5 Transportation Matrix. Finding Optimal Solution by Modified Distribution (MODI) Method. (u, v and Δ). Maximum Two Iterations (i.e. Maximum Two Loops) after IFS)	
3	PERT and CPM	3.1	Critical Path Method (CPM). Concepts: Activity, Event, Network Diagram, Merge Event, Burst Event, Concurrent and Burst Activity, Construction of a Network Diagram. Node Relationship and Precedence Relationship. Principles of Constructing Network Diagram. Use of Dummy Activity Numerical Consisting of Maximum Ten (10) Activities.	10
		3.2	Critical Path, Sub-critical Path, Critical and Non-critical Activities, Project Completion Time. Forward Pass and Backward Pass Methods. Calculation of EST, EFT, LST, LFT, Head Event Slack, Tail Event Slack, Total Float, Free Float, Independent Float and Interfering Float	
		3.3	Program Evaluation and Review Technique (PERT) Three Time Estimates of PERT: Optimistic Time (a), Most Likely Time (m) and Pessimistic Time (b). Expected Time (te) of an Activity Using Three Time Estimates. Difference between CPM and PERT. Numerical Consisting of Maximum Ten (10) Activities. Construction of PERT Network using te values of all Activities. Mean (Expected) Project Completion Time. Standard Deviation and Variance of Activities. Project Variance and Project Standard Deviation. 'Prob. Z' Formula. Standard Normal Probability Table. Calculation of Probability from the Probability Table using 'Z' Value and Simple Questions related to PERT Technique	
		4.1	Job Sequencing Problem. Processing Maximum 9 Jobs through Two Machines only. Calculations of Idle Time, Elapsed Time etc.	

4	Job Sequencing and Theory of Games	4.2	Processing Maximum 6 Jobs through Three Machines only. Calculations of Idle Time, Elapsed Time etc.	10
		4.3	Theory of Games: Introduction Terminology of Game Theory: Players, Strategies, Play, Payoff, Payoff matrix, Maximin, Maximax, Saddle Point.	
		4.4	Numerical based on: Two Person Zero Sum Games - Pure Strategy Games (Saddle Point available)	
			Total number of Lectures	**40**

Course Outcomes: By the end of the course the student will be able to:

1. Be able to understand the application of OR and frame a LP Problem with solution – graphical.
2. Be able to solve LPP by applying SIMPLEX ALGORITHM.
3. Be able to build and solve Transportation Problem problems using appropriate method.
4. Be able to build and solve Assignment PROBLEM problems using appropriate method.
5. Be able to design and solve simple models of CPM and PERT to improve decision making and develop critical thinking and objective analysis of decision problems.
6. Enables to take best course of action out of several alternative courses for the purpose of achieving objectives by applying game theory and sequencing models.

References:

1. Operations Research Techniques for Management, Kapoor V.KSultan Chand & Sons, 7th.
2. Operations Research, Kantiswarup, Gupta P.K. &Manmohan, Sultan Chand & Sons, 9th.
3. Operations Research, Sharma S. D, Kedarnath, Ramnath & Company, 8th.
4. Quantitative Techniques in Management, Vora N. D. Tata McGraw Hill co, 3rd.
5. Operations Research, P. K. Gupta D. S. Hira Sultan Chand & Sons.

Practicals

S.N.	Topic	CO
1	Basic inbuilt commands in R	**CO1**
2	R program on Matrix	**CO1**
3	R program on lpp graphical solution	**CO2**
4	R program on lpp simplex	**CO2**
5	R program on assignment Minimization,Maximization	**CO3**
6	R program on Assignment Balanced and Unbalanced	**CO3**
7	R program on TransportationIFS	**CO3**
8	R program on CPM	**CO4**
9	R program on PERT	**CO5**
10	R and Python program on Game theory	**CO6**
	Total number of Lectures	**20**

CASE STUDY

CASE STUDY	
1	The data for a PERT network is given in the following table. Answer the following questions based on the data. (time in weeks)

Activity	Optimistic time (a)	Most likely time (m)	Pessimistic time (b)
1-2	4	5	6
1-3	4	6	8
2-3	3	4	5
2-4	5	6	7
3-5	1	3	5
4-6	1	2	3
5-6	2	4	6
6-7	1	2	3

2	A company produced two products A,B. Profit per unit of A is Rs.30 AND B is Rs.50.There Resources M1,M2,M3 are utilized. Capacities of M1,M2,M3 are 4,6,12 Hours Respectively. Following feasible solution has been obtained by the simplex method. Based on this answer the following questions with Justification.

Cj			30	50	0	0	0
c	x	b	x1	x2	S1	S2	S3
0	S1	4	1	0	1	0	0
0	S2	0	-3/2	0	0	1	-1/2
50	X2	6	3/2	1	0	0	1/2
Zj			75	50	0	0	25

BOS	Computer Science
Class	S.Y.I.T
Semester	III
Subject Name	Advanced Java
Subject code	PUSIT303
Course Type	Major
Level of the Subject	Medium
Credit	2 Theory + 2 Practical

Course Objectives:

1.Design and develop GUI applications using Abstract Windowing Toolkit (AWT), Swing, Event Handling
and Multithreading for designing Simulation and Game based applications.
2.Enhance knowledge to manipulate and store data. Designing applications using pre-built frameworks.

Unit No.	Name of Unit	Topic No.	Name of Topic	Hours
1	File handling Multithreading	1.1	**File Handling** : Introduction to File Handling, Differentiate between input streams and output streams. FileInputStream and FileOutputStream. File Operations,Create File,Get File data,Write into File,Read from File,Delete File, FilePermission Class,FileWriterClass, FileReaderClass ,RandomAccessFile Class	**10**
		1.2	**Multithreading**:Threads and Multithreading, The Lifecycle of a thread, Creating and running threads, Thread-safety Synchronizing threads, Communication between threads, Thread Class vs. Runnable Interface.	
2	Networking & Event Handling	2.1	**Networking:** what is java Networking ?Networking Terminology, Socket Programming ,URL Class, URLConnection Class.	**10**
		2.2	**The Delegation Event model**: Event sources and event listeners **Event Classes**:ActionEvent class,MouseEvent class,ItemEvent class,KeyEvent class,TextEvent class,Windowevent class ,Adapter classes **Event Listener Interface:**ActionListenerInterface ,ItemListenerInterface,KeyListenerInterface,MouseListener Interface and MouseMotionListener Interface,WindowListenerInterface **Layouts Manager:**BorderLayout,FlowLayout,GridLayout**,**GridBagLayout	
3	Swing Components	3.1	**Swing Components** – I: Introduction to JFC and Swing, difference between awt and swing. Swing API	**10**

			Components, JComponent Class, Windows, Dialog Boxes, and Panels, Labels, combobox, Buttons,radio button , Check Boxes, Menus, JScrollPane, list, Text-Entry Components.	
		3.2	**Advanced Swing Component**:tabbed pane,Tables and Trees,Toolbars, tooltips,progress bar, Colors and File Choosers.	
4	JDBC Connectivity	**4.1**	JDBC Introduction, JDBC Architecture, Types of JDBC Drivers, The Connectivity Model, The java.sql package, Navigating the ResultSet object's contents.	**10**
		4.2	Manipulating records of a ResultSet object through User Interface , The JDBC Exception classes, Database Connectivity, Data Manipulation (using Prepared Statements. and CallableStatement)	
	Total No of Lectures			**40**

Course Outcomes:

1.Analyze the change in file handling, multithreading and networking
2.Evaluate the role of socket programming and event handling.
3. Create a full set of UI widgets and other components, including windows, menus, buttons, checkboxes, text fields, scrollbars and scrolling lists, using Swings.
4. To learn and create advanced standalone GUI concepts
5.Learn skills to develop real time applications
6.Students learn to access database through Java programs, using Java Database Connectivity (JDBC)

References

1.Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD
2.Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR
3. Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill Additional Reference
4.The Java Tutorials of Sun Microsystems Inc.
5.https://www.simplilearn.com/java-networking-article

CASE STUDY	
1	You have been assigned to develop an Online Bookstore Management System (OBMS) using Java with JDBC connectivity to a relational database. The system aims to facilitate the management of books, customers, orders, and inventory for an online bookstore. The OBMS should include the following functionalities:Book Management,Customer management, inventory management,user interface,database connectivity.
2	You are tasked with developing a Student Record Management System (SRMS)using Java Swing components. The system aims to provide administrators with tools to manage student records efficiently, including adding, editing, deleting, and searching for student information. The SRMS should include the following functionalities:Student Information Management,Course, Enrollment Management and user interface.

Practicals

Practical.No	Practicals
1	a. Write a Java program to read a text file (input.txt) line by line and display its content on the console. b. Develop a Java program that reads integers from a file and calculates their sum.
2.	c. Write a Java program that appends a new line of text to an existing file (output.txt). d. bCreate a Java program that writes a list of strings to a file, with each string on a new line.
3	a. Write a Java program that creates two threads and prints "Hello, World!" from each thread. b. Implement a Java program that demonstrates the use of Runnable interface to create threads.
4	a. Design a chatting application by using Socket programming.
5	a. Write a Java program to create a custom event class (e.g., CustomEvent) and an event listener interface (e.g., CustomEventListener) with methods to handle the event.
6	a. Design a swing program to print the factorial for an input value . b. Design a swing application that contains the interface to add student information and display the same.
7.	Design a calculator based on swing application
8.	a. Create a Swing JFrame with multiple JRadioButton components. b. Write a Java program with a JFrame containing a JCheckBox component.
9	a. Create a login page. If the username and password are correct then it says message “Hello <username>” else a message “login failed”
10	a. Create a registration page in Java using JDBC. Accept the details such as Username, Password, Email, and Country from the user using HTML Form and store the registration details in the database.
	Total No.of lectures : 40

BOS	Computer Science
Class	S.Y.B.Sc.I.T
Semester	III
Subject Name	Advanced Database Management Systems
Subject code	PUSIT302
Course Type	Major
Level of the Subject	Medium
Credits	2 Theory + 2 Practical

Course Objectives:

1. To implement basic data management functions using SQL.
2. To learn advanced database systems concepts for implementation and usage.

Unit No.	Name of Unit	Topic No.	Name of Topic	Hours
1	Introduction to PL/SQL and Composite Data Types	1.1	**PL/SQL**: Overview of PL/SQL and its significance in Oracle databases,Basic syntax and structure of PL/SQL blocks,Data Types in PL/SQL– Variable Declaration and initialization	10
		1.2	**Composite Data Types:** PL/SQL Records, The %ROWTYPE Attribute, Insert and Update with PL/SQL Records	
		1.3	**Control Structures:** Conditional Structures, Iterative Control Structures, Sequential Control Structures	
2	Stored Procedures and Functions	2.1	**Stored Procedures:**Understand Procedure**,**Creating a Procedure,Executing a procedure,Deleting Procedure, Parameters-IN,OUT,IN OUT,	10
		2.2	**Functions:** Creating a function,Calling a Function, PL/SQL Recursive FunctionsParameters-IN,OUT,IN OUT	
		2.3	**SQL Transaction Control Statements:** Defining a Transaction, Committing a Transaction,Rolling Back Transactions,Savepoints,Commands	
3	Cursors and Exception Handling	3.1	**Cursors**: Concept of a cursor, types of cursors: implicit cursors; explicit cursor, Cursor for loops, Cursor variables, Parameterized cursors	10
		3.2	**Exception Handling** :Understand Exceptions, Handle Exceptions with PL/SQL, Trap Predefined Oracle Server Errors,Trap User-Defined Exceptions, Propagate Exceptions	
		3.3	**Triggers:** Triggers, Uses of Triggers, Parameters, Levels of Triggers	

4	Packages and MongoDB	4.1	**Packages:** Advantages of Packages, Components of a Package, Develop a Package, enable visibility of a Package's Components, Create the Package Specification and Body, Invoke the Package Constructs, Overloading Subprograms in PL/SQL	10
		4.2	**Overview of MongoDB:** A MongoDB Document document-oriented data model, MongoDB Query API and its Uses	
		4.3	**Basic Operations on MongoDB:** Create Database, Create collection,Insert Documents in Database, Find Documents in Database, Update Database,Delete Database	
			Total No. of Lectures	**40**

Course Outcomes: By the end of the course the student will be able to:
1: Understanding PL/SQL Fundamentals,Data Types,Composite Data Types
2: Implement the Control Structures,Procedures,Functions .
3: Describe the use of transactions and cursors in PL/SQL
4: Examine how to handle Exceptions by making use of Advanced SQL
5: Interpret the role of Packages and Triggers in PL/SQL
6: Demonstrate basic operations in MongoDB through practical exercises.

References:

1. Oracle SQL and Pl/SQL, Joel Murach .
2.PL/SQL Language Reference 11g, , Sheila Moore, E. Belden,Introduction to Database System C.J.Date Pearson First 2003
3. Ivan Bayross, “SQL,PL/SQL -The Programming language of Oracle”, B.P.B. Publications
4.Ramakrishnam, Gehrke, Database Management Systems, Bayross, McGraw-Hill,3rd Edition
5.Abraham Silberschatz, Henry F. Korth,S. Sudarshan , Database System Concepts, 6th Edition
6."Learning MongoDB: A beginner's guide to building scalable and flexible databases" -Alok Singh:
7."MongoDB: The Definitive Guide" - Shannon Bradshaw, Eoin Brazil, Kristina Chodorow

CASE STUDY	
1	**Opti Mart Supermarkets Database Improvement** Opti Mart Supermarkets is a popular retail chain with stores all over the country. They're facing some big problems with their database – it's just not keeping up with all the stuff they need to do! Their database stores lots of important information, like what's in stock, who's buying what, how sales are doing, and how they're working with suppliers. The bosses at Opti Mart are worried because these problems are slowing everything down. It's making it harder for them to keep customers happy. So, they've decided it's time to fix things up and make their database work better. As a database expert, you've been asked to come up with a plan to make Opti Mart's database awesome.
2	**Student Course Registration System** ABC University is implementing a new Student Course Registration System to streamline the process of enrolling students in courses each semester. The system needs to handle tasks such as student registration, course enrollment, generating class schedules, and maintaining student records.As a database specialist, you have been tasked with designing and implementing the database for this system. Your goal is to create a database that efficiently manages student information, course offerings, and enrollment data while ensuring data integrity and security.

Practicals

Practical No	Details	Hours
1	PL/SQL Basic a.Basic Program a.Working with Composite Data Types	2
2	Writing Control Structures a.Conditional Structures b.Iterative Control Structures c.Sequential Control Structures	2
3	Cursors a. Writing Implicit Cursor Programs b.Writing Explicit Cursors programs	2
4	Exceptions. a. System Defined Exceptions b.User Defined Exceptions	2
5	Procedures a. Creating Procedures	2
6	Functions a.Creating functions. b.Recursive functions.	2
7	Creating Database Triggers a.Statement Level trigger b.Row level trigger	2
8	Packages a. Creating Packages	2
9	MongoDB a.Create a database using MongoDB b.Create Collection	2
10	Basic Operations a.Insert a document into the database b.Find and Select document from collection c.Update Operation	2
	Total no of Lectures	**20**

BOS	Computer Science
Class	S.Y.B.Sc. I.T.
Semester	III
Course Name	Data Visualization using Python
Course Code	PUSIT304D
Type of course	Discipline Minor
Level of the Subject	Medium
Credit points	3

Course Objectives:

1. To expose students to visual representation methods and techniques that increase the understanding of complex data
2. To introduce students to Python packages that will allow them to create easy to read and understand graphs, charts and other visual representations of data using Python.

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Introduction to Data Visualization and Python libraries for Data Visualization	1.1	Introduction: Data Visualization and Its Importance, Need of Data Visualization in Businesses, Future of Data Visualization, Use of Data Visualization in Business Decision Making	10
		1.2	Data Visualization Techniques: Loading libraries, Popular libraries for data visualization in python, introduction to plots in python, Types of Data required for plot, Installing python libraries.	
		1.3	Defining plot types : bar, line and stacked charts, Drawing a simple sine and cosine plot, Defining axis lengths and limits, Defining plot line styles, properties and format strings	
2	Drawing Plots & Customizing them	2.1	Customizing plots: Setting ticks, labels, and grids, Adding a legend and annotations, Moving spines to the center, Setting the transparency and size of axis labels.	10

		2.2	Making bar charts with error bars, Making pie charts count, Plotting with filled areas, Drawing scatter plots with colored markers	
		2.3	Advanced Customization : Adding a shadow to the chart line, Adding a data table to the figure, Using subplots, Customizing grids, Creating contour plots, Timelines	
3	Matplotlib , Seaborn Plotting and Plotly Plotting.	3.1	Matplotlib: Line Plot, Bar Plot, Scatter plot, Histogram plot, Stack Plot, Pie chart	10
		3.2	Seaborn Plotting: Strip plot, Box Plot, Swarm plot, Joint plot,relational plot, HeatMap, Violin Plot, Facet_grid	
		3.3	Plotly Plotting: Gnatt Chart , Waterfall Chart , Funnel Chart	
4	Making 3D Visualizations and Animations., Plotting Charts with Images and Maps	4.1	3D Visualization and Animations: Creating 3D bars, Creating 3D histograms, Animating in matplotlib	10
		4.2	Plotting with images and maps: Plotting Data on a map using Basemap, Plotting data on a map using Google Map API, Generating Captchas	
		4.3	Animation with Plotly : Bubble Chart, Bar Charts, Adding Control Buttons to Animations, Race Bar Chart	
Total No. of Lectures				**40**

Course Outcomes:

1. Explain the need of Data Visualization and the use of Python .
2. Apply different customization techniques to the graphs to make data more meaningful .
3. Create 3D Visualizations , animations and generate Captchas .
4. Analyze data and use appropriate graphs and charts.
5. Plotting the Google map data with various maps.
6. Compare different plotting techniques with Matplotlib, Plotly and seaborn Library.

References:

1. Dr. Ossama Embarak, “Data Analysis and Visualization using Python”, APress
2. Igor Milovanović , Dimitry Foures , Giuseppe Vettigli, “Python Data Visualization Cookbook”, Packt Publishing
3. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.
4. VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with

Data. O'Reilly Media.

5. Healy, K. (2019). Data Visualization: A Practical Introduction. Princeton University Press.
6. Kandel, S., Heer, J., Plaisant, C., & Kennedy, J. (2012). Designing Interactive Visualization Tools to Support Creativity and Insight. ACM.

CASE STUDY

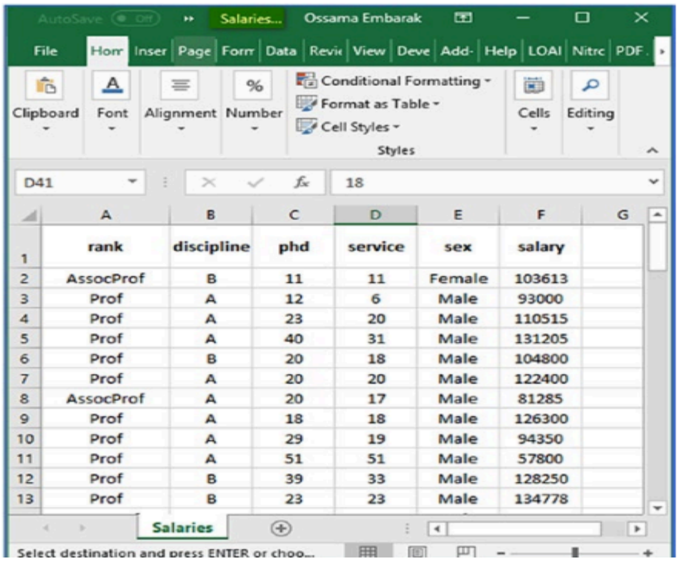
1

2013-2018, the Sales of the company were too good in terms of quantity, profit, and unit sales. But after five years slowly the sales drastically decreased due to the involvement of competitors in the market. The company has 4 branches North, South, east, and Center. The Sales manager wants to analyze the Sales data of the company so that he can find the reasons for the loss.
Consider the following columns for the dataset.

Order id, Cust_id, Cust_name, Order_date, Month, Year, City, Region, product, category, unit_price, quantity, discount, total price, profit_loss

2

1) ABC college of Arts,Science and commerce was established in 1978 by Dehradun Education Society. It is ideally located in the heart of Uttarakhand G. A. Marg is served by a number of BEST bus routes.As the first degree college , it was started with the aim to cater higher education needs of students in neighboring areas. The college is affiliated to the University of Uttarakhand. We are blessed with dedicated, experienced and well qualified teachers. The below give chart shows salaries data set.Answer the following questions given below:



D41 | 18

	A	B	C	D	E	F	G
1	rank	discipline	phd	service	sex	salary	
2	AssocProf	B	11	11	Female	103613	
3	Prof	A	12	6	Male	93000	
4	Prof	A	23	20	Male	110515	
5	Prof	A	40	31	Male	131205	
6	Prof	B	20	18	Male	104800	
7	Prof	A	20	20	Male	122400	
8	AssocProf	A	20	17	Male	81285	
9	Prof	A	18	18	Male	126300	
10	Prof	A	29	19	Male	94350	
11	Prof	A	51	51	Male	57800	
12	Prof	B	39	33	Male	128250	
13	Prof	B	23	23	Male	134778	

Select destination and press ENTER or choo...

Practicals

Practical No.	Details
1	Plot a Simple histogram and bar plot and apply various customization techniques.
2	Create a simple plot and add ticks, labels, axes
3	Plot Strip plot, Box Plot, Swarm plot, Joint plot, on Tips dataset.
4	Plot relational plot, HeatMap, Violin Plot, Facet_grid on Tips dataset.
5	To add legends and annotations to the graph
6	Create an exploded pie chart and stack plot.
7	Create a TimeLine on Date time column from Sample Superstore dataset.
8	Create a 3D bar for a sample data
9	Demonstrate some matplotlib and Plotly animations with Bar Race Chart and Bubble chart.
10	To add an annotation to a chart using images and text
11	● To plot data on a map using GoogleMap API ● To create a simple Captcha Generator
12	Use the Plotly Library to show the use of Waterfall and Gantt charts.
	Total No. of Lectures :20

BOS	Computer Science
Class	S.Y.B.Sc.I.T
Semester	III
Subject Name	Interaction Design
Course Type	Discipline Minor
Subject code	PUSIT304 (U)
Level of The Subject	Intermediate
Credits	3

Course Objectives:

1. To demonstrate a deep understanding of interaction design principles, methodologies, and tools.
2. Equipped to analyze user needs, design intuitive interfaces, and develop interactive systems that prioritize user satisfaction, efficiency, and accessibility.

Unit No.	**Name of Unit**	**Topic No.**	**Name of Topic**	**Hours**
I	Introduction to Interaction Design	1.1	Introduction,Good and Poor Design,What Is Interaction Design?,The User Experience,Understanding Users,Accessibility and Inclusiveness,Usability and User Experience Goals	10
		1.2	The process of Interaction Design Introduction,What Is Involved in Interaction Design?,Some Practical Issues	
II	Conceptualizing & Cognitive Aspects	2.1	Introduction,Conceptualizing Interaction,Conceptual Models, Interface Metaphors, Interaction Types, Paradigms, Visions, Theories, Models, and Frameworks	10
		2.2	Cognitive Aspects,Introduction,What Is Cognition?,Cognitive Frameworks	
III	Emotional Interaction & Interfaces	3.1	Introduction,Emotions and the User Experience,Expressive Interfaces and Emotional Design,Annoying Interfaces,Affective Computing and Emotional AI, Persuasive Technologies and Behavioral Change,Anthropomorphism	10
		3.2	Introduction, Interface Types,Natural User Interfaces and Beyond,Which Interface?	
IV	Data gathering & Interaction Design in Practice	4.1	Introduction,Five Key Issues,DataRecording,Interviews,Questionnaires,Observation,Choosing and Combining Techniques	
		4.2	Introduction,AgileUX,Design Patterns,Open Source Resources,Tools for Interaction Design	10
	Total Lectures			**40**

Course Outcomes:

1. Understand the principles and theories of interaction design.
2. Analyze user needs and behaviors to inform design decisions.
3. Apply user-centered design methodologies in the creation of interactive experiences.
4. Develop proficiency in prototyping tools and techniques.
5. Conduct usability testing to evaluate and refine design solutions.

6. Gain practical experience in designing interfaces for web and mobile applications.

References:

1) Interaction Design: beyond human-computer interaction, Fifth Edition Published by John Wiley & Sons, Inc.
2) The Design of Everyday Things" by Don Norman "About Face: The Essentials of Interaction Design" by Alan Cooper, Robert Reimann, and David Cronin
3) "Interaction Design: Beyond Human-Computer Interaction" by Jenny Preece, Helen Sharp, and Yvonne Rogers"
4) The Elements of User Experience" by Jesse James Garrett
5) Cooper, A., Reimann, R., & Cronin, D. (2014). About Face: The Essentials of Interaction Design (4th ed.). Wiley.

CASE STUDY	
1	Redesigning a Mobile Banking App A leading bank wants to enhance its mobile banking app to provide a more intuitive and user-friendly experience for its customers. The current app has received complaints about its complex navigation, limited features, and inconsistent visual design. The bank aims to improve user satisfaction, increase engagement, and attract new customers through a comprehensive redesign of the mobile app.
2	Improving E-commerce Checkout Process An e-commerce company noticed a high rate of cart abandonment during the checkout process on their website. Users were abandoning their carts primarily due to a lengthy and confusing checkout process, leading to a loss in sales and revenue. The company aimed to improve the checkout experience to reduce cart abandonment and increase conversions.

Practicals

Practical No	**Details**
1.	Introduction to Interaction Design.
2.	Create surveys to gather quantitative data on user preferences and behaviors.
3.	Perform contextual inquiries by observing users in their natural environment.
4.	Design responsive layouts for a website, considering different screen sizes and orientations.
5.	Prototype a mobile-first approach by designing the mobile version of an interface first and then scaling up for larger screens.
6.	Develop scenarios or user stories to illustrate how each persona would interact with the product or service.
7.	Create user flow diagrams to visualize the paths users take through an app or website.

8.	Use Figma or Adobe XD's diagramming tools to map out the user journey from entry point to conversion.
9.	Design and prototype microinteractions such as button presses, menu toggles, and scroll animations to enhance user engagement.
10.	Work collaboratively on a design project with team members, utilizing Figma's real-time collaboration features such as shared editing and commenting.
	Total No of Lectures:20

BOS	Information Technology
Class	S.Y.B.Sc.I.T
Semester	III
Course Name	Full stack Development Paper-I
Course Code	PUSIT304(F)
Type of course	Discipline Minor
Level of the Subject	Medium
Credit points	3

Course Objectives:

1. Provide an overview of React.js and its key features, including virtual DOM, component-based architecture, and declarative syntax.
2. Understanding JSX: Introduce JSX (JavaScript XML), the syntax extension used in React.js for defining the structure and layout of components.

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Getting Started with React.js	1.1	**Introduction:** Introduction to React and its benefits, Environment setup for React development, Refresher on ES6 concepts, Create React App, Folder Structure.	10
		1.2	**Templating using JSX:** Understanding component architecture and its significance, Introduction to components and their types,Working with React.createElement to create elements,Using expressions, logical operators, attributes, and children in JSX.	
		1.3	**Working with Props and State :** Understanding the concept of state and its significance in React, Setting and reading component states, Working with props to pass data between components, Validating props using propTypes, Using default props to supply default	

			values, Rendering lists using the React key prop and the map function.	
2	Understanding Component Lifecycle:	2.1	Stateful Function Components with Hooks: Using hooks, specifically useState, useEffect, useContext, and useReducer,Creating custom hooks,	10
		2.2	Overview of the lifecycle of a React component,Handling side effects and managing cleanup,Working with events and error management,	
		2.3	Exploring the React event	
3	Working with Forms	3.1	Managing controlled and uncontrolled components. Utilizing the defaultValue prop. Accessing the DOM element using the React ref prop. Building a currency converter project.	10
		3.2	Understanding useContext and useReducer hooks, Creating custom hooks.	
		3.3	Routing with React Router: Setting up React Router for navigation in single-page applications,Configuring routes with BrowserRouter and HashRouter. Making routes dynamic using route params,Working with nested routes and navigation using Link and NavLink components.	
4	State Management and Redux:	4.1	Introduction to Redux and its principles. Installing and setting up Redux. Creating actions, reducers, and the store.	10
		4.2	Setting up Redux for state management. Implementing actions and reducers for the catalog and cart.	
		4.3	Using the connect() higher-order function to connect components. Utilizing Redux Hooks for state management. Implementing middleware and persistence in Redux.	
			Total No. of Lectures	**40**

Course Outcomes:

1. Gain a solid understanding of React.js and its core concepts, including virtual DOM, component-based architecture, and JSX syntax.
2. Develop the ability to create reusable and modular components in React.js, allowing for efficient and scalable front-end development.

3. Effectively manage state and props in React.js components, enabling dynamic rendering and interaction within applications.
4. How to bind event handlers, handle form submissions, and update component state based on user interactions.
5. React Router, a popular routing library for React.js, and demonstrates how to implement client-side routing in a React application.
6. To apply the concepts learned throughout the course and build real-world React.js applications.

References:

1. Functional Web Development with React and Redux by Alex Banks and Eve Porcello:
2. React Up and Running: Building Web Applications by Stoyan Stefanov:
3. Pro React by Cassio de Sousa Antonio:
4. React Cookbook: Create Dynamic Web Apps with React using Redux, Webpack, Node.js, and GraphQL by Carlos Santana Roldan:
5. Fullstack React: The Complete Guide to ReactJS and Friends by Anthony Accomazzo, Nate Murray, and Ari Lerner:

CASE STUDY	
1	Traveldock is the biggest Polish marketplace that connects individual adventure trip organizers with their customers. They offer a wide range of activities, from trekking, biking and sailing in Poland to exploring Chernobyl or diving in Sri Lanka. Challenge: Travel Ducks goal was to consolidate the market of boutique trips offered by individual providers on a very fragmented market. The end goal was to create an easy-to-use marketplace where tour organizers can showcase and sell their adventure trips.
2	SnowShow is the most prominent Polish winter tour provider, and also one of the biggest in Europe. They target the Alps in terms of destination and students and grads as their audience. Their uniqueness lies in the combination of sport (skiing and snowboarding) with music. SnowShow organizes over 60 trips yearly. The biggest one, called "Music Fest", attracts over 2000 adventurous people and the biggest music stars. Challenge: We started our journey with SnowShow in the fall of 2015. Their system already existed and was developed by a freelancer. SnowSnow needed a more robust solution with a trustworthy and scalable team. The booking system we took over was complex and included multiple implicit business logic. Making it simple, readable and self-descriptive was the main business and technical challenge for us at the beginning of the project.

Practicals

Sr. No.	Practicals
1.	Create a new React project using Create React App.
2.	Review ES6 features like arrow functions, template literals, destructuring, spread syntax,
3.	**Templating using JSX:** 1. Create a simple React component using JSX. 2. Use expressions, logical operators, attributes, and children in JSX.
4.	**Understanding Components and their types:** 1. Explore the concept of components in React. 2. Understand the difference between functional components and class components
5.	**Working with Props and State:** 1. Create a component that accepts and uses props. 2. Set and read component states using useState. 3. Pass data between components using props.
6.	**Stateful Function Components with Hooks:** 1. Create stateful function components using useState. 2. Utilise useEffect for handling side effects and cleanup.
7.	**Lifecycle Methods and Event Handling:** 1. Understand the lifecycle of a React component. 2. Handle events using the React event system.
8.	**Accessing the DOM Element using the React ref prop:** Use the ref prop to access and manipulate DOM elements in React.
9.	**Routing with React Router:** 1. Install and set up React Router for navigation in single-page applications. 2. Create dynamic routes using route parameters.
10.	**Installing and Setting up Redux:** 1. Install the Redux library and set up the Redux store. 2. Add middleware to handle asynchronous actions.
	Total No of lectures:20

SEMESTER IV

BOS	Information Technology
Class	SY.B.Sc. I.T.
Semester	IV
Course Name	Advanced Web Programming
Course Code	PUSIT401
Type of course	Major
Level of the Subject	Advanced
Credit Points	2 Theory + 2 Practical

Course Objectives:

1. To provide insight into .NET technologies for web programming and enable them to design and develop interactive and responsive web applications.
2. To learn Connectivity with Database using ADO.Net.

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Introduction to .NET Framework	1.1	The .NET Framework**:** .NET Languages, Common Language Runtime, .NET Class Library	10
		1.2	The C# Language: C# Language Basics, Variables and Data Types, Variable Operations, Object-Based Manipulation, Conditional Logic, Loops, Methods, Exception Handling.	
		1.3	Types, Objects, and Namespaces: The Basics About Classes, building a Basic Class, Value Types and Reference Types, Understanding Namespaces and Assemblies, Advanced Class Programming.	
2	ASP. NET Web Controls	2.1	Web Form Fundamentals: Introducing Server Controls, Using the Page Class, Using Application Events,	10
		2.2	Form Controls: Stepping Up to Web Controls, Web Control Classes, List Controls, Table Controls, Web Control Events and AutoPostBack, Validation, Understanding Validation, Using the Validation Controls,	

3	Rich Controls and Master Pages	3.1	Rich Controls, The Calendar, The Ad Rotator, Pages with Multiple Views, User Controls, Website Navigation: Site Maps, The Sitemap Path Control, The Tree View Control, The Menu Control.	10
		3.2	Styles, Themes, and Master Pages: Styles, Themes, Master Page Basics, Advanced Master Pages	
	State management ADO.NET Data Controls	4.1	State Management: Understanding the Problem of State, Using View State, Transferring Information Between Pages, Using Cookies, Managing Session State, Configuring Session State, Using Application State, Comparing State Management Options	10
4		4.2	ADO.NET Fundamentals: Understanding Databases, Congaing our Database, Understanding SQL Basics, Using Direct Data Access, Using Disconnected Data Access. The Data Controls: The Grid View, Formatting the Grid View, selecting a GridView Row, Editing with the GridView, Sorting and Paging the GridView, The DetailsView and FormView	
			Total No. of Lectures	**40**

Course Outcomes:

1. Identify the features of .net framework along with the feature of c#.
2.Implement various controls and navigation techniques for integrating web pages within the site.
3.Apply master pages and state management techniques for web pages.
4. Develop crud applications using sql server database.
5 Understand the fundamentals of Angular Forms and its controls
6.Using ADO.net

References:

1. Beginning ASP.NET 4.5 in C#Matthew MacDonald, Apress(2012)
2. The Complete Reference ASP .NET, MacDonald, Tata McGraw Hill
3. Building Web Solutions with ASP.NET and ADO.NET, Wintellect
4. ASP.NET 4.0 Programming, J.Kanjilal, Tata McGraw Hill(2011)
5.https://www.w3schools.com/angular/

CASE STUDY	
1	Building an Employee Management System ABC Corporation needed an Employee Management System (EMS) using ASP.NET with C#. Their requirements included user authentication, a master page for layout consistency, employee data entry with validation controls, web server controls for data input, ADO.NET for database connectivity, role-based security, and CSS styling for aesthetics. By following these guidelines, ABC Corporation successfully developed an EMS that tracks employee information securely and provides an appealing user interface.
2	XYZ Corporation plans to develop a web-based Employee Management System (EMS) utilizing web server controls like TextBoxes, Labels, and Buttons for data input, display, and actions. Rich controls such as GridView for employee lists and a DropDownList for department selection will be employed. Validation controls like RequiredFieldValidator and CustomValidator will ensure data accuracy, while a Menu and SiteMapDataSource will enhance navigation. The EMS aims to streamline employee record management, offer a user-friendly interface, and improve HR department efficiency and data reliability at XYZ Corporation.

Practicals

Practical No.	**Details**
1.	Working with basic C# and ASP .NET . Create an application that obtains four int values from the user and displays the product. b. Create an application to demonstrate string operations c. Programs to create and use DLL
2	Create an application to demonstrate following operations i. Generate Fibonacci series. ii. Test for prime numbers. iii. Test for vowels. iv. Use of foreach loop with arrays v. Reverse a number and find the sum of digits of a number
3.	Create a simple application to demonstrate use of following concepts i. Function Overloading ii. Inheritance (all types) iii. Constructor overloading iv. Interfaces

	v. Using Delegates and events vi. Exception handling
4	. Create a simple web page with various server controls to demonstrate setting and use of their properties. (Example: AutoPostBack) b. Demonstrate the use of Calendar control to perform following operations. i. Display messages in a calendar control ii. Display vacation in a calendar control iii. Selected day in a calendar control using style
5	Working with Form Controls a. Create a Registration form to demonstrate use of various Validation controls. b. Create Web Form to demonstrate use of Ad rotator Control c. Create Web Form to demonstrate use of User Controls.
6.	Working with Navigation, Beautification and Master page. Create Web Form to demonstrate use of Website Navigation controls and Site Map. Create a web application to demonstrate the use of Master Page with applying Styles and Themes for page beautification. c. Create a web application to demonstrate various states of ASP.NET Pages.
7.	Working with Database (Connected Data Access) Write a web application to perform CRUD operation.
8	Create a web application to display Using Disconnected Data Access and Data Binding using Grid View.
9	Working with GridView control Create a web application to demonstrate use of GridView button column and GridView events. b. Create a web application to demonstrate GridView paging and Create your own table format using GridView.
	Total no of lectures:20

BOS	Computer Science
Class	S.Y.B.Sc.I. T.
Semester	IV
Course Name	Operating System
Course Code	PUSIT402
Type of course	Major
Level of the Subject	Moderate
Credit points	2 Theory + 2 Practical

Course Objectives:

1. To understand the basic concepts of Operating Systems concepts.
2. To understand linux and its different set of basic commands.

Unit No	Name of Unit	Topic No	Contents	Hours
1	Introduction and Concepts & process	1.1	Introduction: Definition of Operating system, Architecture, Structures of O.S, Basic functions, System calls, multiprogramming. time sharing, parallel, distributed & real time O.S.	10
		1.2	Process Management: Process Concept, General structure of a typical process, Process states, Process control Block. Process Scheduling	
		1.3	Uni-processor Scheduling: Types of scheduling: Pre-emptive, Non-preemptive, Scheduling algorithms: FCFS, SJF, RR, Priority	
2	Memory Management and file management	2.1	Deadlock: Principles of deadlock, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection	10
		2.2	Memory Management: requirements, Memory partitioning: Fixed and Variable Partitioning, Fragmentation, Swapping, Memory Management techniques Paging. Segmentation,	

		2.3	Demand/virtual Memory: Concepts, management of VM, Page Replacement Policies (FIFO, LRU, Optimal,	
3		3.1	File-System Interface: File Concept, attributes, permission, Access Methods, File System Mounting	10
		3.2	A brief history of LINUX, architecture of LINUX, features of LINUX, introduction to vi editor. Linux commands- PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, man, echo, bc,	
		3.3	Linux file system, File permissions Basic file commands in linux: cd, mkdir, rmdir, cp, mv, rm, cat, more, Basic file attributes: ls, chmod, directory permissions	
	4	4.1	The shell: introduction, shell interpretive commands, wildcard characters, ,pipes, shell script, shell variables	10
		4.2	Process: process, shell process, process commands (ps and its options), utilities: ps command, disk utilities: unlink, du, df, mount, umount, find, unmask	
		4.3	Processing utilities and backup utilities, tail, head, Advance Commands: grep and sed.	
	Total number of Lectures			**40**

Course Outcomes:

1. Understand the objectives, functions and structure of OS
2. Analyze the concept of process management and evaluate performance of process scheduling algorithms.
3. Ability to understand file concepts and file File structure in linux
4. Ability to use various Linux commands that are used to manipulate system operations at admin level and a prerequisite to pursue a job as a Network administrator.
5. Ability to write Shell Programming using Linux commands.
6. Understand use of linux advanced commands .

References:

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley,8th Edition
2. Sunithabha Das,UNIX concepts and applications, 4th edition, Tata McGrawHill publication
3. Achyut S. Godbole, Atul Kahate, Operating Systems, Tata McGraw Hill 3.
Naresh Chauhan, Principles of Operating Systems, Oxford Press

4. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016.

CASE STUDY	
1	Samsan Company is a leading manufacturer of household appliances, including washing machines. They are aiming to enhance the functionality of their washing machines by integrating an operating system that can automate and control all machine operations efficiently. Objective: The objective is to design an operating system specifically tailored for Samsan Company's washing machines. The operating system should be compact in size, portable, and capable of managing various aspects of the machine's operations. Automation: The operating system should automate key functions such as water inlet and outlet, drum rotation, temperature control, and detergent dispensing. User Interface: A user-friendly interface should be developed to allow users to set wash cycles, select water temperature, adjust spin speed, and monitor the progress of the wash cycle. Fault Detection: The operating system should include algorithms for detecting faults such as water leaks, imbalance in load distribution, motor malfunctions, and sensor failures. Energy Efficiency: Implement energy-saving features such as automatic shutdown after completing a cycle, standby mode when not in use, and optimized power usage during operation. Security: Ensure secure communication protocols between the operating system and external devices (e.g., mobile apps for remote control) to prevent unauthorized access and potential cyber threats. Portability: The operating system should be lightweight and adaptable to different hardware configurations to ensure compatibility across various models of Samsan washing machines. Designing a tailored operating system for Samsan Company's washing machines aligns with their goal of delivering innovative and high-performance appliances to customers. By focusing on automation, user experience, efficiency, and security, the operating system project aims to elevate the functionality and value proposition of Samsan washing machines in the competitive market landscape
2	VACC is a small company specializing in data management services for their customers. Recognizing the need for efficient data handling within their own operations, they have decided to set up a computer lab with a Local Area Network (LAN) in their office premises. Objective: The objective is to establish a client-server architecture within the computer lab, enabling centralized data storage on the server machine managed by the IT manager. This setup will allow other employees to access and work with the stored data seamlessly. Requirements: Server Configuration: Procure and configure a dedicated server machine

	capable of handling the company's data storage needs and running essential server software. Client Machines: Set up multiple client machines in the computer lab, each equipped with the necessary networking capabilities to connect to the LAN and access the server. Networking Infrastructure: Install networking equipment such as switches, routers, and cables to establish a reliable LAN connection within the office premises. Client-Server Software: Deploy client-server software that facilitates secure data transfer and communication between client machines and the server, ensuring data integrity and access control. Data Backup: Implement a robust data backup strategy to safeguard against data loss and ensure data availability in case of hardware failures or system errors. Security Measures: Configure firewall settings, encryption protocols, and access permissions to protect the server and data from unauthorized access and cyber threats. Benefits: Centralized Data Management: The client-server setup allows for centralized storage and management of company data, improving data organization and accessibility for employees. Collaborative Work: Employees can collaborate more effectively by accessing shared data and resources stored on the server, enhancing productivity and teamwork. Data Security: Implementing security measures ensures data confidentiality, integrity, and availability, reducing the risk of data breaches or unauthorized access. Scalability: The LAN-based client-server model is scalable, allowing VACC to expand its computer lab infrastructure and accommodate future growth and technological advancements. Efficient IT Management: Centralized server management by the IT manager streamlines data maintenance, backup, and software updates, optimizing IT operations within the company. By establishing a computer lab with LAN and implementing a client-server architecture, VACC enhances its data management capabilities, promotes collaboration among employees, and ensures the security and reliability of its data infrastructure, contributing to overall operational efficiency and business continuity.

Practicals

Practical No	Details
1	Implement FCFS scheduling algorithm in Java. Implement SJF (with no preemption) scheduling algorithm in Java
2	Implement RR scheduling algorithm in Java Implement RR scheduling algorithm in Java
3	Write a Java program that implements the page-replacement algorithm. FIFO , optimal
4	General purpose utilities in linux PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, man, echo, bc,
5	Working with Directories and files : a. ls, mkdir, rmdir, b. file, touch, rm, cp. mv, file, touch, rm, cp. mv, rename
6	Process : process, shell process,process commands (ps and its options), utilities: ps command
7	disk utilities: unlink, du, df, mount, umount, find, unmask
8	practical on Advance Commands: a. grep and sed. b. Processing utilities and backup utilities , tail, head
9	Write Shell Script for followings 1. Program to print hello message 2. program to ad two numbers 3. Program to 1) find how many users logged in 2) count how many files and directories
10	Write Shell Script for followings 1. To find the global complete path for any file. 2. To broadcast a message to a specified user or a group of users logged on any terminal. 3. To copy the file system from two directories to a new directory in such a way that only the latest file is copied in case there are common files in both the directories.
	Total number of Lectures 20

BOS	Computer Science
Class	S.Y. B.Sc. IT
Semester	IV
Subject Name	Data Structures Using Python
Course Type	Major
Subject Code	PUSIT403
Level of the Subject	Basic
Credits	2Theory + 2 Practical

Course Objectives:
1 To Understand data structures and different techniques to manage data.
2 Introduce the concept of data structures through ADT including List, Stack, Queues and develop applications using data structure algorithms.

Unit No.	Name of Unit	Topic No.	Name of Topic	Hours
1	Abstract data types and Algorithm Analysis	1.1	**Abstract Data Types and Arrays:** Introduction, The Date Abstract Data Type, Bags, Iterators, Application. Array Structure, Python List, Two Dimensional Arrays, Matrix Abstract Data Type, Application	10
		1.2	**Sets and Maps:** Sets-Set ADT, Selecting Data Structure, List based Implementation, Maps-Map ADT, List Based Implementation, Multi-Dimensional Arrays-Multi-Array ADT, Implementing Multi Arrays .	
		1.3	**Algorithm Analysis:** Complexity Analysis-Big-O Notation, Evaluating Python Code,Evaluating Python List, Application	
2	Linked structure and sorting	2.1	**Searching and Sorting:** Searching-Linear Search, Binary Search, Sorting-Bubble, Selection and Insertion Sort, Working with Sorted Lists-Maintaining Sorted List	10
		2.2	**Linked Structures**: Introduction, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, Bag ADT-Linked List Implementation. Linked List Iterators, Applications-Polynomials.	
		2.3	**Advanced Linked List**: Doubly Linked Lists-Organization and Operation,Circular Linked List-Organization and Operation	
3	Stack and Queue	3.1	**Stacks and Queue**: Stack ADT, Implementing Stacks-Using Python List, Using Linked List, Stack Applications-Balanced Delimiters, Evaluating Postfix Expressions	10

		3.2	**Recursion:** Recursive Functions, Properties of Recursion, Its working, Recursive Applications.	
		3.3	**Hash Table and Collision:** Introduction, Hashing-Linear Probing, Clustering, Rehashing, Separate Chaining, Hash Functions, Collision.	
4	Tree and Graph and Heap	4.1	**Binary Trees:** Tree Structure, Binary Tree-Properties, Implementation and Traversals ,Expression Trees, Binary Search Tree, Operations on Binary Search Tree	10
		4.2	**Heaps:**Heaps, types and Heap Sort.	
		4.3	**Graph:** Introduction, Graph, Graph Terminology, Memory Representation of Graph, Adjacency Matrix Representation of Graph, Adjacency List or Linked Representation of Graph, Graph Traversal, Shortest Path problem using Dijkstra's algorithm, Applications of the Graph.	
			Total No. of Lectures	**40**

Course Outcomes:

1. Use the primitive data types and abstract data types using Python programming language.
2. Apply the basic concepts of Data structure using Python
3. Analyze complexity of data using algorithms
4. Understanding the concept of stack, link list,queue
5. Evaluate different techniques to search and sort data.
6. Create Tree and Graph and Heap techniques to store data properly in memory.

References:

1.Data Structure and Algorithms in Python,Goodrich, tamassia,Goldwasser.
2. Data structure and Algorithms using Python-rance D. Necaise, College of William and Mary,2016,J.Wiley.
3.Data Structure and Algorithmic Thinking with Python-Narasimha karumanchi,2015,Careermonk publication.
4.Fundamentals of Python:Data Structure,Kenneth lambert,Delmar Cengage Learning.
5:Python :The Complete Reference by Martin C. Brown

CASE STUDY	
1	**Inventory Management System** In an inventory management system, we can utilize abstract data types such as bags to represent different categories of items. Each category can be implemented as a bag ADT, allowing efficient storage and retrieval of items. For example, a bag can represent the category of electronics, another for clothing, and so on. Iterators can be employed to traverse through these bags, facilitating operations like adding new items, removing items, and updating quantities. To manage the inventory efficiently, we can implement a

	linked list to store the details of each item within a category. Each node of the linked list can hold information such as item name, quantity, price, etc. Using recursion, we can perform operations like searching for a specific item, updating its quantity, or removing it from the inventory. For instance, when a customer purchases an item, a recursive function can be employed to update the quantity of the item in the inventory recursively across all categories if needed, ensuring accurate stock management.
2	**Social Network Analysis Tool** In a social network analysis tool, sets can be employed to represent connections between users. Each user can have a set containing their friends or followers. Maps can be used to store additional information about each user, such as their profile details, interests, etc. For instance, a map can associate each user ID with a profile containing their name, age, location, etc. Multi-dimensional arrays can be utilized to store metrics such as the number of likes, comments, or shares each user's posts receive. Hash tables can be utilized for efficient storage and retrieval of user profiles and their corresponding data. Collision handling techniques such as separate chaining can ensure that each user's data is stored correctly and can be accessed quickly. Graphs can be employed to analyze the network structure, identifying communities, influential users, or potential connections. Algorithms like Dijkstra's algorithm can be applied to find the shortest path between users, facilitating efficient communication within the network.

Practicals

Practical No.	Details
1	Create Array ADT and perform 2D array addition operation.
2	Create a Singly Linked list ADT and perform traverse operation.
3	Implement Linear Search to find an item in a list.
4	Implement binary search to find an item in an ordered list.
5	Implement Sorting Algorithms for Bubble Sort.
6	Implement Sorting Algorithms for Insertion Sort.
7	Implement use of Sets and various operations on Sets.
a	Implement working of Stacks. (pop method to take the last item added off the stack and a push method to add an item to the stack)
8	Implement a queue as a list which you add and delete items from.
10	Implement Binary Tree and its traversals.
11	Recursive implementation of
a	Factorial
b	Fibonacci
c	Tower of Hanoi
12	Write a program to display the adjacency matrix for a given Graph.
	Total no of lectures:20

BOS	Computer Science
Class	S.Y.B.Sc.I.T.
Semester	IV
Subject Name	Introduction to Data Science
Course Type	Discipline Minor
Subject Code	PUSIT404(D)
Level of the Subject	Medium
Credits	3

Course Objectives:

1. Understanding basic Data Science concepts.
2. Learning to detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization.

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Introduction to Data & Data Analysis	1.1	What is Data? Different kinds of data, Data Sources, Different types of data sources,	10
		1.2	Data Science lifecycle, Data Collection	
		1.3	Data Extraction, Data Analysis & Modeling	
		1.4	Exploratory Data Analysis (EDA); Univariate , Bivariate , Multivariate , Graphical, Non- Graphical.	
2	Python Libraries for Data Science , Numpy and Pandas	2.1	The World of arrays with Numpy: creating an array, Mathematical operations, Indexing and slicing, Shape manipulation, Sorting and Searching	10
		2.2	Empowering Data analysis with pandas: the data structure of pandas, Inserting and exporting data,	
		2.3	Data Cleansing**:** checking missing data, filling missing data, merging operations	
		2.4	Data Operations**:**Data transformations Dimension reduction, Feature extraction, Smoothing and aggregating, Aggregation operations, Joins	
3	PySpark Architecture, RDD, Data Frames for	3.1	Introduction : Who uses PySpark Features , Advantages, PySpark Architecture , Modules and Packages	10
		3.2	PySpark RDD : RDD creation, RDD	

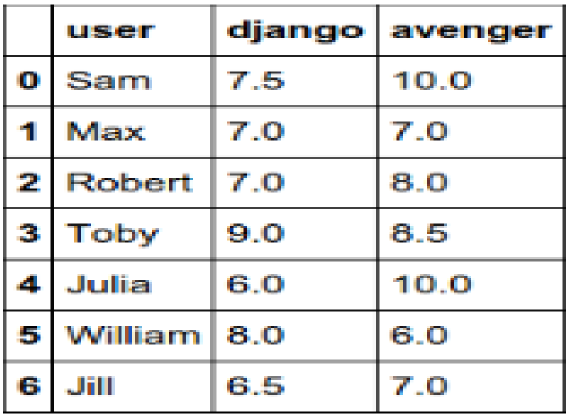
	Bigdata Analytics.		operations PySpark DataFrame: Difference between Pandas Dataframe and Pyspark dataframe, DataFrame creation, DataFrame Operations, PySpark SQL , Function.	
4	Machine Learning basics & Generating Recommendation systems	4.1	Introduction to Machine learning: Different types of Machine Learning, Linear Regression, Logistic Regression, K-means Clustering, Hierarchical Clustering	10
		4.2	Generating Recommendations Systems: User Based collaborative filtering, Item Based collaborative filtering, Context Based filtering	
		4.3	Case Study Theory: Analyzing Unstructured Data using Text mining techniques. (Case Study Practical Implementation to be performed in lab as part of Practical's)	
			Total Lectures	**40**

Course Outcomes:

1 . Enumerate the Various types of Data and Data Analysis Techniques.
2. Illustrate the Various Data transformation techniques.
3 . Apply the various functions on data for data cleaning and exporting.
4 . Illustrating the various data handling techniques using PySpark.
5. Analyzing the data with various techniques.
6. Predict the data using Machine Learning Techniques.s

References :

1. Mastering Python for Data Science, Explore the world of data science through Python and learn how to make sense of data, Samir Madhavan Packt Publishing.
2. Python Data Science Handbook: Essential Tools for Working with Data, Jake VanderPlas O'Reily
3. Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. O'Reilly Media.
4. Grolemund, G., & Wickham, H. (2017). R for Data Science: Import, Tidy, Transform, Visualize, and Model Data. O'Reilly Media.
5. McKinney, W. (2017). Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython (2nd ed.). O'Reilly Media.

<table>
<tr><th>CASE STUDY</th><th></th></tr>
<tr><td>1</td><td>The provided dataset captures information about transactions at a restaurant, including details such as the total bill amount, tip given, gender of the customer, smoking status, day of the week, time of the day, and the size of the party. Each row in the dataset represents a specific dining event with associated attributes. The key features include:-
total_bill: The total amount of the bill for each dining event.
tip: The amount of tip given during each transaction.
sex: The gender of the customer (Male or Female).
smoker: Whether the customer is a smoker or non-smoker (Yes or No).
day: The day of the week when the dining event occurred.
time: The time of the day for the dining event (e.gDinner).
size: The size of the party (number of individuals).
The dataset seems to focus on a specific day ('Sun') and time ('Dinner').
The goal could be to analyze and understand patterns or relationships within the provided data, such as the impact of total bill amount on tips, differences in tipping behavior between genders, or variations in total bill amounts across different party sizes.</td></tr>
<tr><td>2</td><td>An OOT App Manager wants to give suggestions to their customers based on their previous web series ratings. The manager uses the euclidean distance formula to calculate the similarity score to check the similar user.

<table>
<tr><th></th><th>user</th><th>django</th><th>avenger</th></tr>
<tr><td>0</td><td>Sam</td><td>7.5</td><td>10.0</td></tr>
<tr><td>1</td><td>Max</td><td>7.0</td><td>7.0</td></tr>
<tr><td>2</td><td>Robert</td><td>7.0</td><td>8.0</td></tr>
<tr><td>3</td><td>Toby</td><td>9.0</td><td>8.5</td></tr>
<tr><td>4</td><td>Julia</td><td>6.0</td><td>10.0</td></tr>
<tr><td>5</td><td>William</td><td>8.0</td><td>6.0</td></tr>
<tr><td>6</td><td>Jill</td><td>6.5</td><td>7.0</td></tr>
</table></td></tr>
</table>

Practicals

Practical No	Details
1.	Write a NumPy program to swap rows and columns of a given array in reverse order.
2.	A. Write a NumPy program to sort an given array by the nth column. B. Write a NumPy program to partition a given array in a specified position and move all the smaller elements values to the left of the partition, and the remaining values to the right, in arbitrary order (based on random choice). C. Write a NumPy program to count the number of dimensions, number of elements and number of bytes for each element in a given array. D. Write a NumPy program to get a copy of a matrix with the elements below the k-th diagonal zeroed
3.	A. Write a Pandas program to split the following dataframe by Unique_column and get mean, min, and max value of age for respective dataset. B. Write a Pandas program to split a given data frame into groups with multiple aggregations.
4.	Perform various data transformation techniques on numerical and date time data.
5.	Perform Univariate , Bivariate and Multivariate graphical and non graphical analysis on Sample Superstore dataset.
6.	Practical to implement Data Manipulation using Pandas Techniques.(Handling of missing value , Renaming a column, Data Extraction, Categorical encoding)
7.	Setup Github Account, loading data from different source files formats (csv, excel) and summarizing data with statistics.
8.	Perform the Data Recommendation technique of some random data.
9.	Create a dataframe using PySpark and perform basic operations.
10.	Practical to implement case study on Analyzing Unstructured Data using Text Mining.
	Total No of Lectures:20

BOS	Computer Science
Class	S.Y.B.Sc.I.T
Semester	IV
Subject Name	UX/UI Design
Course Type	Discipline Minor
Subject Code	PUSIT404(U)
Level of the Subject	Basic
Credits	3

Course Objectives:

1. To understand the Designing Platform for User Interface and User Experience .
2. To understand user needs and make it more effective using visual design processes, Wireframes & Prototyping

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Introduction to UI/UX	1.1	Introduction to what is interaction, Human Computer Interaction (HCI). What is UX and UX design, UX Design versus UI Design, Importance of UX, UX design process, Full stack design, Understanding what is full stack designer And its objectives.	10
		1.2	UX Design Process, Discovery and planning, UX strategy, UX research: The discover stage, The explore stage, The test stage, UX analysis and Design	
2	Overview of UI User Behaviour, Persona & Design Behaviour	2.1	User Behaviour Basics and User Research, The Gestalt theory : The Proximity law, The Similarity law, The Closure law, The Figure-Ground law, The Common Region law	10
		2.2	User Personas, Understanding the user personas, Four different perspectives on personas, Benefits of personas	
		2.3	Designing Behaviour, Five factors/preconditions for users to take actions, Models of behavior change	
3	Overview of Visual Design Principles, Processes,	3.1	Visual Design Principles and Processes: Introducing visual design principles and processes, Basics of visual design, using lines, Using shapes, Types of shapes, Shape	10

	Wireframes & Prototyping		usage in visual design, Using colors, Textures, Forms, Design principles, Visual design tools.	
		3.2	Wireframes and Prototyping, what is a wireframe, how to create wireframes, Types of wireframes. Wireframing tools: Sketch wireframes, Stencilling and paper cut-outs, Wireframing software, creating wireframes using graphic design software	
		3.3	What is prototyping, Prototyping methods, the process of creating prototypes, Prototyping tools	
4	UI Design Implementation and Post Launching Activities	4.1	UI Design and Implementation, User interface design, UI design tools, Creating the Design System in Sketch: Creating a structure for files and folders, Following the proper naming convention,	10
		4.2	Choosing the colors and creating the palettes, choosing fonts/typefaces, Creating and configuring the grid, Designing the UI components	
		4.3	Frontend UI Implementation and Process, UI Design handover Using a handover design tool: Handing-off UI design using Zeplin, handing over design using Sympli, Frontend development/UI development, CSS layouts, CSS pre-processor's, CSS postprocessors, CSS methodologies, CSS frameworks.	
		4.4	Post-launching UX Activities, Collecting the correct user feedback, User accessibility testing (UI testing), A/B testing, Tracking and recording user UI sessions, Creating and analyzing conversion funnels,	
	Total Lectures			**40**

Course Outcomes:

1. Understand basics of UI/ UX Design
2. Illustrate different types of User behaviour, persona & Design Behaviour.
3. Identify different aspects of Visual design, Principles, Processes , Wireframes & Prototyping.
4. UI Design and Implementation.
5. Understanding the UI Design Handover Process using tools.
6. Learning the Post- Launch UX Activities

References :

1. Hands-on UX Design for Developers, Design, prototype and implement compelling user experience from scratch, By Elvis Canziba, Packt Publication , 2018.
2. Sarrah Vesselov, Taurie Davis - Building Design Systems_ Unify User Experiences through a Shared Design Language-Apress (2019).
3. Marsden, P. (2015). Mobile User Experience: Patterns to Make Sense of it All. Morgan Kaufmann.
4. Norman, D.A. (2013). The Design of Everyday Things: Revised and Expanded Edition. Basic Books.
5. Gopinath, S., & Bindal, A. (2015). Designing UIs for Cross-Platform Applications: A Practical Guide. Apress.

CASE STUDY	
1	UX design of a simple mobile application on cooking, recipes and food shopping.The creative team wanted to step aside from the traditional recipe app where users just save the directory of the favorite meals, taken from the app database, or add their own recipes. We had a goal to create a bit more universal food app for users who love cooking. It includes the recipe database which is constantly updated. Also, the application has a supplies manager. To make UX more extended, it allowed users to find the recipes by the supplies they currently had at home or create a shopping list to buy ingredients that were missing.The app design included the comprehensive and diverse functionality which had to be presented to users in a simple and clear way. The designers had to analyze and prioritize all the points, as there was a high risk of overloading the screen. By research and testing, the user scenarios were created to determine which information about the meal in the recipe is found the most important.As the recipe app is aimed at daily basic operations and quite a diverse target audience, the user interface has to be super easy and accessible for users with different levels of tech-literacy and all types of mobile devices. The application layout is structured around intuitive navigation, high readability, light background, and eye-catching visuals. Clear and solid typography based on san-serif fonts makes the information scannable and legible on the screens of different sizes. Color contrast is used for amplifying quick navigation: bright color accents attract users' attention to interactive zones and active states of the layout elements. The search field is easily found on the top of the screen: its functionality is clarified for users with both text prompt and search icon.
2	One of India's most successful startups, Ola suffers from a common problem— clutter. An app used by millions on the daily provides so many options for customization that the user is often overwhelmed.Going back to the roots , Ola solves a fundamental problem — that of servicing taxi rides with the help of our smartphones.After talking to frequent users of the app spanning from different age groups (18–49), I inferred that users sometimes found booking cabs on Ola harder as compared to rival platforms — which I connected to some inconsistent flows. This combined with the constant call to actions (CTAs) and tiny text placement might've been the reason.As Indians, most of us like to compare the various ride types at a glance, knowing the price difference between a Mini and a Prime ride on a less-busy day can lead to a much more comfortable ride.For the more tech-savvy users, it was

	found that the app at many places, displayed the same type of information. A major example of this could be found in the ‘Know Your Ride’ section, which redirects the user to a ‘WebView’, whereas long-pressing the ride type also leads to a short description of the car specs and ‘Know More’ leads to the same ‘WebView’. But quite possibly, the biggest issue almost everyone complained about was the display of singular prices at the confirm ride screen. As Indians, most of us like to compare the various ride types at a glance, knowing the price difference between a Mini and a Prime ride on a less-busy day can lead to a much more comfortable ride.Identifying weaknesses,Inconsistent user flows,Too many options in Menu,Promos / CTAs (Call to action) densely populated across the experience,Laggy animations,Task While observing the current situation, I realized that all of these weaknesses could be solved by simplifying the user flow for some basic actions and then adding secondary CTAs & promos on top of it.But quite possibly, the biggest issue almost everyone complained about was the display of singular prices at the confirm ride screen. As Indians, most of us like to compare the various ride types at a glance, knowing the price difference between a Mini and a Prime ride on a less-busy day can lead to a much more comfortable ride.Identifying weaknesses,Inconsistent user flows,Too many options in Menu,Promos / CTAs (Call to action) densely populated across the experience,Laggy animations,Task While observing the current situation, I realized that all of these weaknesses could be solved by simplifying the user flow for some basic actions and then adding secondary CTAs & promos on top of it.

Practicals

Practical No	Details
1.	Introduction to Figma Tool
2.	Design a UX prototype to demonstrate Font, Color, Typography.
3.	Create UX design for Login Page.
4.	Perform Boolean Operations and Design Logo.
5.	a) Perform User Research (for online shopping /any domain specific) b) System Concept Statement (prepare short summary report on what approach used)
6.	User Requirement Analysis (Draw Flow Model).
7.	Create User Personas and User Scenario.
8.	Create a Site map, Wireframe, Screens, Widgets.
9.	Setting Properties, Screen transitions , Header and Footer.
10.	Perform Usability Testing .
	Total No. Of Lectures:20

BOS	Information Technology
Class	S.Y.B.Sc.I.T
Semester	IV
Course Name	Full stack Development Paper-II
Course Code	PUSIT404(F)
Type of course	Discipline Minor
Level of the Subject	Moderate
Credit points	4 Theory + 1 Practical

Course Objectives:

1.Understand the JavaScript and technical concepts behind Node JS.

2.Connect to a SQL or Mongo database in Node Understand how the MEAN stack works

Unit No.	Name of Unit	Topic No	Cont ent	Hours
1	Getting Started with Node.js	1.1	Introduction : Introduction, What is Node JS?, Advantages of Node JS, Traditional Web Server Model, Node.js Process Model. Difference : Node.js vs AngularJS, Node.js vs Python, Node.js vs PHP, Node.js vs Java Setup Dev Environment : Install Node.js on Windows, Installing on Linux.,	10
		1.2	Node JS Console: get started with console, examples, Working in REPL: REPL Environment,How to start REPL, Node.js Simple expressions, REPL Commands Node.js Package Manager: Installing Modules using npm, Uninstalling a Module, Global vs Local Installation,Searching a Module,	
		1.3	Node.js Basic: Command Line Options & Node.js Global Objects, Node.js Buffer: Creating Buffers, Writing to buffers,Reading from buffers. Node.js Streams: What are Streams?, Reading from a Stream, Writing to a Stream,Piping the Streams,Chaining the Streams.	

2	Understanding Component Lifecycle:	2.1	File System: Fs.read, File Writing a File, Writing a file asynchronously, Opening a file, Deleting a file, Other IO Operations Node.js OS: what is OS, its methods with examples. Node.js Errors : Errors, types, examples.	10
		2.2	Node.js Timer : Set timer functions with examples, Clear timer functions with examples. Node.js DNS: What is DNS, its methods with examples. Node.js Net: What is socket programming, implements client -server application. Node.js Path : What is path, Node.js Path Methods, examples. Node.js StringDecoder: What is String Decoder, methods, examples.	
		2.3	Node.js Crypto : What is Hash, What is HMAC, Encryption Example using Hash and HMAC, Encryption example using Cipher, Decryption example using Decipher Node.js TLS/SSL: What is TLS/SSL, What is public-key cryptography, Node.js TLS client example. Node.js Debugger: Syntax , example.	
3	Callbacks & Events	3.1	Node.js Process: Node.js Process Properties with examples, Node.js Process Functions with examples Callbacks: Blocking Code Example, Non Block Code Example Events : Event Driven Programming, Difference between Events and Callbacks, EventEmitter class, Returning event emitter, Inhering events.	10
		3.2	Creating Web server: What is Web Server, Web Application Architecture, Creating Web Server using Node.js	

4	Node.js with Express	4.1	What is Express.js? Features and advantages of Express.js Installing Node.js with NPM Creating Routes and Handling Requests	10
		4.2	Handling different HTTP request methods (GET, POST, PUT, DELETE) Accessing route parameters and query parameters Middleware in Express.js	
		4.3	Creating a MongoDB database connection Node.js, Performing CRUD operations us database (Create, Read, Update, Delete),Query data using MongoDB, filters and projections	
	Total No. of Lectures			**40**

Course Outcomes:

1. Understand the fundamental concepts and architecture of Node.js.
2. Compare and contrast Node.js with other technologies such as AngularJS, Python, PHP, and Java.
3. Gain proficiency in working with Node.js basic commands, global objects, buffers, streams, file system operations, and the OS module.
4. Learn to write asynchronous code using callbacks and implement event-driven programming using EventEmitter.
5. Create web servers using Node.js and understand web application architecture.
6. Gain practical knowledge of integrating Node.js with databases like MySQL and MongoDB, including performing CRUD operations.

References

1. "Modern Full-Stack Development with Node.js" by F. Leitner et al. (2017).
2. "Microservices in Practice: Insights from Full-Stack Development" by M. Keller et al. (2016)
3. "Towards a Unified Full-Stack Development Framework for IoT Applications" by S. Hu et al. (2020)
4. "Full-Stack Development in Cloud Environments: Challenges and Opportunities" by N. Sharma et al. (2018)
5. "Agile Full-Stack Development: A Case Study" by P. Johnson et al. (2019)

CASE STUDY	
1	Developing Real-Time Chat-bots As you may be aware, real-time conversations are now commonly employed on almost every website on the internet, as you may have seen. To the point that they've practically become a must, particularly for commercial websites and digital items, The good news is that Node.js comes pre-loaded with essential features for developing real-time chat applications. The Node.js Event API enables developers to create

	server-side events and push notifications, commonly used in real-time chat applications like Skype and Facebook Messenger.
2	Developing Single-Page Applications Single-page apps have grown more popular in recent years. In SPAs, the whole program is virtually contained on a single page, allowing users to experience something similar to a desktop application. Node.js is an excellent choice for developing SPAs since it can effectively handle asynchronous calls and intensive I/O operations. Furthermore, Node.js is well-suited for data-driven single-page apps, in which the server serves as the back-end, transmitting data to the client, while the client-side handles all of the HTML renderings.

Practicals

Practical No.	Details
1	Node.js Console: 1. Demonstrate different operators using console.log() 2. Print all even or odd numbers between 1 and 10 3. Find the maximum number from an array of numbers 4. Calculate the factorial of a given number
2	Working in REPL: 1. Calculate the sum of two numbers entered by the user 2. Calculate the Area of a Circle 3. Convert Temperature from Celsius to Fahrenheit 4. Generate a Random Number 5. Check if a Number is Even or Odd
3	Node.js Buffer and Node.js Streams: 1. Create a buffer with a capacity of 10 bytes, write "Hello!" to it, and read its contents 2. Read data from a text file using a readable stream and write it to a new file using a writable stream
4.	File System: 1. Synchronous and asynchronous file reading, writing, and appending 2. Synchronous and asynchronous file deletion 3. Read the contents of a text file, convert the text to uppercase, and write it to a new file 4. Creating a directory

5.	Node.js modules OS: 1. Print the name and version of the operating system using the Node.js os module 2. Resolve the IP address of a domain name using the Node.js dns module (DNS module) 3. Implement a simple TCP server that responds with a message when a client connects (Net module)
6	Node.js Errors: 1. Throw and catch an error with a custom message Node.js Timer: 1. Use setTimeout to display a "Hello, World!" message after a delay of 3 seconds
7	Node.js Crypto: 1. Generate a hash of a string using the Node.js crypto module
8	Node.js TLS/SSL: 1. Connect to a secure HTTPS website and print the response using the Node.js tls module
9	Node.js Process and Debugger: 1. Print the current process ID, title, and command-line arguments using the Node.js process object Identify and fix a syntax error using the Node.js debugger
10	Performing CRUD operations using database (Create, Read, Update, Delete),
	Total No of lectures:20

****