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: Bachelor of Computer Applications

S.Y. Bachelor of Computer Applications

PCACS/BCA/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)

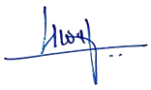
Affiliated to University of Mumbai

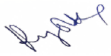
NAAC Accredited 'A' grade (3 cycles)

Best College Award by University of Mumbai

ISO 9001:2015 Certified

Board of Studies in the Department of Computer Science

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 Pillai, Director Pillai Center for Innovation & Research	Management Representative	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.. 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	
10	Mrs. Shubhangi Pawar	Faculty Specialization	
11	Mrs. Soly Zachariah	Faculty Specialization	
12	Mrs. Ramya S. Kumar	Faculty Specialization	
13	Mrs. Sujata Shahabade	Faculty Specialization	
14	Mrs. Sreevidya T.V.	Faculty Specialization	
15	Mr. Omkar Sherkhane	Faculty Specialization	
16	Mr. Abhijeet Salvi	Faculty Specialization	

Introduction to Bachelor of Computer Applications

Bachelor of Computer Application 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 and business 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 Computer Application lead to rewarding and lucrative careers, excellent placement and incubation assistance is provided.

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

Program Specific Outcomes

Sr No	PSOs in brief
PSO1	Understand the fundamentals and applications of programming, Data Structures, Databases, Networking, Internet of Things,Mobile Computing, Information security and Data Analytics.
PSO2	Effectively integrate I.T. based solution in the users domain after properly analyzing the requirements and the constraints.
PSO3	Unique Knowledge of Technology in Business Applications and Computational tools for simulation.
PSO4	Ability to comprehend and write effective project reports in a multidisciplinary environment in the context of changing technologies.

Course Structure

S.Y.B.C.A

Semester III						
Course Code	**Course Type**	**Cours Title**	**Theory/ Practical**	**Marks**	**Credits**	**Lectures/ Week**
PUSCA301	MAJOR	Database Management System	Theory	100	2	4
PUSCA302	MAJOR	Data Structure using Python	Theory	100	2	4
PUSCA303	MAJOR	Business Accounting	Theory	100	2	4
PUDSE304D/ PUDSE304U/ PUDSE304F	DISCIPLINARY MINOR	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3 Full Stack Development paper-1	Theory/Practical	150	2+2	4+2
PUSCA305	SEC	Mini Project	Theory/ Practical	50	2	2
PUSCA306P	MAJOR	Database Management System Practical	Practical	50	2	2
PUSCA307P	MAJOR	Data Structure using Python Practical	Practical	50	2	2
PUSCA308P	MAJOR	Business Accounting Practical	Practical	50	2	2
PUAEC	AEC	Languages (To be taken from pool)	Theory	100	2	3
PUIDC	IDC	To be taken from pool	Theory	100	2	4
Total				850	22	33
All Subjects having Field Project as part of Continuous Assessment-2						

Abbreviations:

IDC: Interdisciplinary Course

AEC : Ability Enhancement Course

SEC : Skill Enhancement Course

Semester IV						
Course Code	**Course Type**	**Course Title**	**Theory/ Practical**	**Marks**	**Credits**	**Lectures/ Week**
PUSCA401	MAJOR	Enterprise Resource Planning	Theory	100	2	4
PUSCA402	MAJOR	Advanced JAVA	Theory	100	2	4
PUSCA403	MAJOR	Advanced Database Management System	Theory	100	2	4
PUDSE404D/ PUDSE404U/ PUDSE404F	DISCIPLINARY MINOR	Track1- Introduction to Data Science Track2- UI-UX Design Track-3 Full Stack Development Paper-2	Theory/ Practical	150	2+2	4+2
PUSCA405	SEC	Mini Project	Theory/ Practical	50	2	2
PUSCA406P	MAJOR	Enterprise Resource Planning Practical	Practical	50	2	2
PUSCA407P	MAJOR	Advanced JAVA Practical	Practical	50	2	2
PUSCA408P	MAJOR	Advanced Database Management System Practical	Practical	50	2	2
PUAEC	AEC	Languages (To be taken from pool)	Theory	100	2	3
PUIDC	IDC	To be taken from pool	Theory	100	2	4
Total				850	22	33
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.

SEMESTER III

Course Code	Course Type	Course Title	Evaluation Pattern
PUSCA301	MAJOR	Database Management System	A
PUSCA302	MAJOR	Data Structure using Python	A
PUSCA303	MAJOR	Business Accounting	A
PUDSE304D/ PUDSE304U/ PUDSE304F	DISCIPLINARY MINOR	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3 Full Stack Development paper-1	A
PUDSE304D/ PUDSE304U/ PUDSE304F	DISCIPLINARY MINOR PRACTICAL	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3 Full Stack Development paper-1	D
PUSCA305	SEC	Mini Project	B
PUSCA306P	MAJOR	Database Management System	D
PUSCA306P	MAJOR	Data Structure using Python	D
PUSCA308P	MAJOR	Business Accounting	D

SEMESTER IV

Course Code	Course Type	Course Title	Evaluation Pattern
PUSCA401	MAJOR	Enterprise Resource Planning	A
PUSCA402	MAJOR	Advanced JAVA	A
PUSCA403	MAJOR	Advanced Database Management System	A
PUDSE404D/ PUDSE404U/ PUDSE404F	DISCIPLINARY MINOR	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3 Full Stack Development paper-1	A
PUDSE404D/ PUDSE404U/ PUDSE404F	DISCIPLINARY MINOR PRACTICAL	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction Design Track-3 Full Stack Development paper-1	D
PUSCA405	SEC	Mini Project	B
PUSCA406P	MAJOR	Enterprise Resource Planning Practical	D
PUSCA407P	MAJOR	Advanced JAVA Practical	D
PUSCA408P	MAJOR	Advanced Database Management System Practical	D

SEMESTER III

BOS	Computer Science
Class	S. Y. B.C.A
Semester	III
Subject Name	Database Management System
Subject Code	PUSCA301
Level of the Subject	Basic

Course Objectives:

1. Introduction of the concept of the DBMS with respect to the relational model,
2. Specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases.

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Introduction to DBMS and Data Models	1.1	Introduction to DBMS : Data, ,Database, Application of DBMS ,DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction/ Data independence,	10L
		1.2	DBMS Architecture, Client/Server Architecture, Three –Tier architecture Data models - (relational, hierarchical, network Object Oriented).	
		1.3	Entity Relationship Model – ER diagram Entities and types, attributes and types, entity sets, relations and Notations, relationship sets, aggregation / generalization.	
2	Introduction to Database Languages and Constraints	2.1	DDL Statements :Creating Databases, Using Databases, data types, Creating Tables Altering Tables(alter with add columns, alter to drop columns), Renaming Tables, Dropping Tables.	10L
		2.2	DML Statements : insert, select, update, delete, unique records, conditional select, Clauses-where, aggregate functions (count, min, max, avg, sum), group by clause, having clause, order by, distinct	
		2.3	Relational Constraints - primary key, referential integrity(foreign key), unique constraint, Not Null constraint, Check constraint, default constraint.	
3	Normalization and Join Operation	3.1	Schema refinement and Normal forms(Normalization):Functional	10L

			dependencies(Anomalies), first, second, third, and BCNF normal forms based on primary keys.	
		3.2	Relational Algebra and Join : fetching operations (selection, projection) Joining Tables –equi join(With ansi, non ansi, using clause), natural joins, inner join, outer join (left outer, right outer, full outer), not equi join, cross join.	
		3.3	Database Protection: Security Issues, Threats to Databases, Security Mechanisms, Role of DBA	
4	Subqueries, View and Database Functions	4.1	Functions – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now)	10L
		4.2	Subqueries – subqueries with IN, NOT IN, Nested query, query to find second highest salary, third highest salary.	
		4.3	Views - Introduction to View, storing complex queries, creating, altering dropping the view, Create view through Join and Subquery Operation.	
Total No. of Lectures				**40L**

Course Outcomes:

1. Describe the basic concept of DBMS and the DBMS Architecture.
2. Develop the skills of Database Languages with DDL And DML Commands.
3. Identify how to make use of Constraints to limit/restrict data.
4. Apply the Normalization concept while designing the database.
5. Choose various Join Operations according to situations.
6. Design and create databases using Subqueries and View.

References:

1. Ramez Elmasri & Shamkant B.Navathe, Fundamentals of Database Systems, Pearson Education, Sixth Edition, 2010
2. Ramakrishnam, Gehrke, Database Management Systems, McGraw‑Hill, 2007 3. Joel Murach, Murach's MySQL, Murach, 2012
3. Database System Concepts, Korth
4. Robert Sheldon, Geoff Moes, Beginning MySQL, Wrox Press, 2005.
5. "Fundamentals of Database Systems" by Ramez Elmasri and Shamkant B. Navathe

CASE STUDY	
1	Design E-R-Data Model Based on following case study. Company has Employees, departments, and projects . Company is organized into departments , Department controls a number of projects, Number of employees working with multiple projects, Employee: store each employee's name, Contact number, address, salary, sex (gender), and birth date , Keep track of each employee belongs to department also notate the relationship amongst each relation also predict the attributes of each entity.
2	Assume you have been asked to design the database structure for a job site. What NORMALIZED table data model / table design would you recommended for storing the personal data of the candidates like name , address, and phone number and their skill in multiple languages so as to ensure that when a company searches based on a particular set of skills, the query is quick to return the result. Note that each candidate would have multiple Skills.(Hint : You will need to have 3 tables in the design)

BOS	Computer Science
Class	S. Y. B.C.A
Semester	III
Subject Name	Database Management System Practicals
Subject Code	PUSCA306P
Level of the Subject	Basic

Practical No.	**Details**	**Hours**
1	Write a DBMS Query to perform DDL Command. i) Create ii) Alter(add, drop, rename) iii) Drop	2L
2	Write a DBMS Query to perform DDL Command i) insert ii) Select iii) Update iv) Delete	2L
3	Write a DBMS Query to perform Clauses i) Where ii) Group by with all aggregate functions iii) Having iv) order By v) Distinct	2L
4	Write a DBMS Query to perform Constraint i) Unique ii) NOT NULL iii) Default iv)Check v) Primary Key vi) Foreign Key	2L
5	Write a DBMS Query to perform Join Operations: i) Equi Join(ansi, non ansi, using clause) ii)Inner Join iii) Natural Join.	2L
6	Write a DBMS Query to perform Join Operations:	2L

	i) Non-Equi Join, ii)Cross Join, iii) Outer Join(Left,Right,Full).	
7	Write a DBMS Query to perform All String Functions:	2L
8	Write a DBMS Query to perform Following Functions ii) Math Functions iii)Date Functions	2L
9	Write a DBMS Query to perform Subqueries: i) IN and NOT IN ii) Second Highest iii) Third Highest	2L
10	Write a DBMS Query to perform View: i) Using Join Operation ii) Using Subqueries.	2L
	TOTAL NO OF LECTURES	20L

BOS	Computer Science
Class	S.Y. BCA
Semester	III
Subject Name	Data Structures Using Python
Subject Code	PUSCA302
Level of the Subject	Basic

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	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	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.

BOS	Computer Science
Class	S.Y.B.C.A
Semester	III
Subject Name	Data Structure Using Python Practical
Subject Code	PUSCA307P
Level of the Subject	Basic

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

BOS	Accountancy
Class	S.Y.B.C.A
Semester	III
Subject Name	Business Accounting
Subject Code	PUSCA303
Level of the Subject	Basic

Course Objectives:

1. To introduce students to basic accounting principles and terminology.
2. To provide an overview of financial statements and their interpretation.

Unit No.	Name of Unit	Topic No.	Name of Topic	Hours
1	Introduction	1.1	Meaning of Bookkeeping and Accounting, Characteristic, Advantages and Disadvantages of Accounting, Difference between Book Keeping and Accounting.	10
		1.2	Accounting principles: Introductions to Concepts and conventions.	
		1.3	Introduction to Accounting Standards: Meaning, Advantages and Disadvantages.	
		1.4	International Financial Reporting Standards (IFRS),Accounting in Computerized Environment	
2	Accounting Transactions	2.1	Journal Entries- Purchase, Sale, Income, Expenditure, Capital, Drawings (Trade Discount and Cash Discount)	10
		2.2	Ledger	
3	Trial Balance and Introduction to Final Account	3.1	Trial Balance - Introduction and Preparation of Trial Balance	10
		3.2	Introduction to Final Accounts	

4	Preparation Final Accounts of Company	4.1	Preparation of Trading Account, Profit and Loss Account and Balance Sheet.	10
		4.2	Adjustment: Closing stock, Outstanding, Prepaid, Depreciation, Bad Debt and RDD	
TOTAL NUMBER OF LECTURES				**40**

Course Outcomes:

1: Understand the purpose and importance of accounting standards in ensuring consistency and transparency in financial reporting.
2: Compare and contrast IFRS with other accounting standards, assess its impact on global financial reporting.
3: Explain how accounting processes are facilitated and enhanced through computerized systems, including software applications.
4: Use knowledge of accounting principles to record various types of transactions accurately.
5: Analyze transaction data, prepare a trial balance, and identify errors in accounting records..
6: Apply higher-order thinking skills to prepare final accounts accurately, incorporating necessary adjustments and interpreting financial information effectively.

References:

1. Ashok Banerjee, (2002), Financial Accounting (a managerial emphasis), Excel Books
2. Anil Choudhary, (2007), Fundamental of Accounting and Financial Analysis, Pearson Education
3. T.P. Ghosh, (2011),Indian Accounting Standards and IFRS for non-finance executives, By, Taxman
4. P.C. Tulsian,(2002), Financial Accounting, Pearson Publications, New Delhi R.L Gupta and
5. M. Radhaswamy, (2014), Advanced Accountancy, New Delhi, S. Chand and Sons (P) Ltd.,

CASE STUDY	
1	ABC Retail Store is a small business that sells clothing and accessories. Analyze their accounting transactions and prepare final accounts for the year ending December 31, 2023. **Accounting Transactions:** 1. **January 1, 2023**: ABC Retail Store started its business with an initial capital of Rs.50,000. 2. **January 5, 2023**: Purchased inventory worth Rs.10,000 on credit from XYZ Wholesale Supplier. 3. **January 10, 2023**: Sold goods worth Rs.8,000 for cash. 4. **January 15, 2023**: Paid Rs.5,000 to XYZ Wholesale Supplier.

5. **February 1, 2023**: Purchased additional inventory worth Rs.15,000 on credit from XYZ Wholesale Supplier.
6. **February 10, 2023**: Sold goods worth Rs.12,000 on credit.
7. **February 20, 2023**: Received Rs.10,000 from credit customers.
8. **February 25, 2023**: Paid Rs.8,000 to XYZ Wholesale Supplier.
9. **December 31, 2023**: Counted the inventory and found it to be Rs.20,000.

2

XYZ Services Agency is a consulting firm providing marketing and advertising services. Let's examine their trial balance and prepare final accounts for the year ended December 31, 2023.

Trial Balance as at December 31, 2023:

Account Title	Debit (Rs)	Credit (Rs)
Cash	25,000	-
Accounts Receivable	15,000	-
Office Supplies	2,000	-
Prepaid Rent	4,000	-
Equipment	40,000	-
Accumulated Depreciation - Equipment	-	5,000
Accounts Payable	-	10,000
Salaries Expense	20,000	-
Rent Expense	8,000	-
Advertising Expense	5,000	-
Utilities Expense	3,000	-
Depreciation Expense	5,000	-
Interest Expense	2,000	-
Revenue	-	100,000
	1,29,000	**1,29,000**

BOS	Computer Science
Class	S.Y. B.C.A
Semester	III
Subject Name	Business Accounting Practicals
Subject Code	PUSCA308P
Level of the Subject	Basic

Practical No	Details	Hours
1	Create a new company in Tally and set up its chart of accounts according to standard accounting principles.	**2**
2	Enter and post journal entries for the following transactions: purchase of inventory on credit, sale of goods for cash, and payment of rent.	**2**
3	Record a cash receipt transaction and allocate it to appropriate ledger accounts in Tally.	**2**
4	Generate a trial balance for the company you created in Tally and ensure that it balances.	**2**
5	Use Tally to prepare a sales invoice for a hypothetical sale transaction, including details such as item description, quantity, rate, and taxes.	**2**
6	Enter and process a payment voucher for various expenses incurred by the company, ensuring accurate allocation to respective expense ledger accounts.	**2**
7	Perform inventory management tasks in Tally, such as creating stock items, recording stock movements (e.g., purchases, sales, returns), and checking stock levels.	**2**

8	Set up GST (Goods and Services Tax) in Tally for the company and record transactions including GST implications.	**2**
9	Generate financial reports such as a profit and loss statement and a balance sheet using Tally for the company's fiscal period.	**2**
10	Reconcile bank transactions in Tally by matching the company's bank statement with the ledger entries and resolving any discrepancies.	**2**
Total No. of Lectures		**20**

BOS	Computer Science
Class	S.Y.B.C.A
Semester	III
Course Name	Data Visualization using Python
Course Code	PUDSE304D
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	10

			transparency and size of axis labels.	
		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:

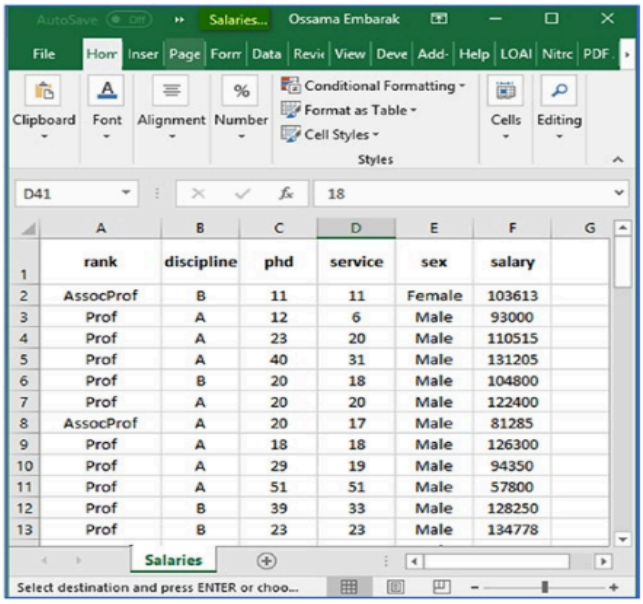
1. Explain the need of Data Visualization and the use of Python
2. Describe plotting of data using graphs and charts

3. Create 3D Visualizations , animations and generate Captchas
4. Analyze data and use appropriate graphs and charts
5. Apply different customization techniques to the graphs to make data more meaningful 6. Compare different plotting techniques

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. "Data Visualization with Python and JavaScript: Scrape, Clean, Explore & Transform Your Data" by Kyran Dale
4. "Storytelling with Data: A Data Visualization Guide for Business Professionals" by Cole Nussbaumer Knaflic
5. "Python Data Visualization: An In-Depth Guide to Matplotlib for Beginners" by Benjamin Walter Keller

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)	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:



	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

BOS	Computer Science
Class	S.Y.B.C.A
Semester	III
Course Name	Data Visualization using Python - Practical
Level of the Subject	Medium

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

BOS	Computer Science
Class	S.Y.B.C.A
Semester	III
Subject Name	Interaction Design
Subject code	PUDSE304U
Level of The Subject	Intermediate

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	**10L**
		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	**10L**
		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	**10L**
		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	**10L**
		4.2	Introduction,AgileUX,Design Patterns,Open Source Resources,Tools for Interaction Design	
	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) "Observing the User Experience: A Practitioner's Guide to User Research" by Mike Kuniavsky, Andrea Moed, and Elizabeth Goodman
3) "Interviewing Users: How to Uncover Compelling Insights" by Steve Portigal
4) "Contextual Design: Defining Customer-Centered Systems" by Hugh Beyer and Karen Holtzblatt
5) "Interaction Design: Beyond Human-Computer Interaction" by Jenny Preece, Helen Sharp, and Yvonne Rogers

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.

BOS	Information Technology
Class	S.Y.B.Sc.I.T
Semester	III
Subject Name	Interaction Design - Practical
Level of The Subject	Intermediate

Practical No	Details	Hours
1.	Introduction to Interaction Design.	2
2.	Create surveys to gather quantitative data on user preferences and behaviors.	2
3.	Perform contextual inquiries by observing users in their natural environment.	2
4.	Design responsive layouts for a website, considering different screen sizes and orientations.	2
5.	Prototype a mobile-first approach by designing the mobile version of an interface first and then scaling up for larger screens.	2
6.	Develop scenarios or user stories to illustrate how each persona would interact with the product or service.	2
7.	Create user flow diagrams to visualize the paths users take through an app or website.	2
8.	Use Figma or Adobe XD's diagramming tools to map out the user journey from entry point to conversion.	2
9.	Design and prototype microinteractions such as button presses, menu toggles, and scroll animations to enhance user engagement.	2
10.	Work collaboratively on a design project with team members, utilizing Figma's real-time collaboration features such as shared editing and commenting.	2
Total No.of Lectures		20

BOS	**Computer Science**
Class	**S.Y.B.C.A**
Semester	**III**
Course Name	**Full stack Development Paper-I**
Course Code	**PUDSE304F**
Level of the Subject	**Medium**
Credit points	**2**

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	No of Lectures
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	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. 1. Create different Components as per requirements. 2. Create a React FORM for taking adventure trips details. 3. Create routes for different types activates
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. Create different Components as per requirements. Create a React FORM for taking Books details Create routes for different types of Tour

BOS	**Computer Science**
Class	**S.Y.B.C.A**
Semester	**III**
Course Name	**Front End development with React.js - Practical**
Level of the Subject	**Medium**
Credit points	**2**

Sr. No.	Practical's name	
1.	Create a new React project using Create React App.	2
2.	Review ES6 features like arrow functions, template literals, destructuring, spread syntax,	2
3.	**Templating using JSX:** 1. Create a simple React component using JSX. 2. Use expressions, logical operators, attributes, and children in JSX.	**2**
4.	**Understanding Components and their types:** 1. Explore the concept of components in React. 2. Understand the difference between functional components and class components	**2**
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.	**2**
6.	**Stateful Function Components with Hooks:** 1. Create stateful function components using useState. 2. Utilise useEffect for handling side effects and cleanup.	**2**
7.	**Lifecycle Methods and Event Handling:** 1. Understand the lifecycle of a React component. 2. Handle events using the React event system.	**2**
8.	**Accessing the DOM Element using the React ref prop:** Use the ref prop to access and manipulate DOM elements in React.	**2**
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.	**2**

10.	**Installing and Setting up Redux:** 1. Install the Redux library and set up the Redux store. 2. Add middleware to handle asynchronous actions.	**2**
TOTAL NO OF LECTURES		**20**

SEMESTER IV

BOS	Computer Application
Class	S.Y.BCA
Semester	IV
Subject Name	Enterprise Resource Planning
Subject code	PUSCA401
Level of The Subject	Intermediate

Course Objectives:

1.Integrate and manage core business processes across various departments within an organization.

2.Understand the ERP Modules,Technologies and Security measures.

Unit No	Name of Unit	Topic No	Name of Topic	Hours
1	Introduction to ERP and ERP Modules	1.1	Introduction Enterprise Resource Planning (ERP) : introduction, history, advantages Enterprise : Basic concepts of ERP Risks and benefits of ERP Components and architecture of ERP systems. ERP Implementation Life Cycle	10
		1.2	Overview of various modules in an ERP system (e.g., finance, human resources, manufacturing, supply chain management, customer relationship management) Functionalities and interdependencies among modules	
2	ERP Implementation, Customization and Configuration	2.1	Phases of ERP implementation (planning, analysis, design, development, testing, deployment, maintenance) ERP implementation methodologies (e.g., Waterfall, Agile) Challenges and risks in ERP implementation	10
		2.2	Customization vs. configuration Tailoring ERP systems to meet organizational requirements Tools and techniques for customization and configuration	
		2.3	ERP Marketplace and Functional Modules Marketplace : overview,	

			dynamics, changing ERP market Indian ERP Scenario Functional modules of ERP software Integration of ERP, SCM and CRM	
3	ERP Related Technologies ERP Integration:	3.1	ERP Related Technologies- Business Process Reengineering (BPR), Online Analytical Processing (OLAP), Data Warehousing, Data Mining, Applications of ERP	10
		3.2	Integration of ERP with other enterprise systems (e.g., Customer Relationship Management (CRM), Supply Chain Management (SCM), Business Intelligence (BI)) Middleware technologies for integration	
4	ERP Security and Control: ERP Maintenance and Support:	4.1	Security issues in ERP systems Role-based access control Security measures and best practices	10
		4.2	Routine maintenance tasks Troubleshooting common issues Help desk support for ERP users	
TOTAL NO. OF LECTURES				40

Course Outcomes:

1. Understanding of the fundamental concepts of Enterprise Resource Planning, including its purpose, benefits, and components.
2. Implementation of methodologies and phases
3. Understand Overview of various modules Functionalities and interdependencies in an ERP system
4. Understanding Tools and techniques for customization and configuration
5. learn how to troubleshoot common issues, Security issues and challenges encountered during ERPimplementations
6. Applying Business related tools and ERP Middleware technologies for integration to get better insight in business

References:

1.Basic Text & Reference Books :- Alexis Leon : Enterprise Resource Planning, Tata McGraw-Hill, New Delhi 1st and 2nd editions. Internet based resource.
2.Alexis Leon, ERP demystified, second Edition Tata McGraw-Hill, 2008.
3.Jagan Nathan Vaman, ERP in Practice, Tata McGraw-Hill, 2008
4.Alexis Leon, Enterprise Resource Planning, second edition, Tata McGraw-Hill, 2008.
5.Vinod Kumar Grag and N.K. Venkitakrishnan, ERP- Concepts and Practice, PHI, 2006.

CASE STUDY	
1	XYZ Enterprises is a mid-sized manufacturing company specializing in the production of automotive components. With multiple departments and manufacturing facilities, the company faced challenges in managing its operations efficiently. The lack of integration between various business processes, including manufacturing, inventory management, sales, and accounting, resulted in inefficiencies and data silos. Challenges: 1.Disparate Systems: XYZ Enterprises used multiple standalone systems for different business functions, leading to data duplication and inconsistency. 2.Lack of Real-time Visibility: Due to the disjointed systems, the company lacked real-time visibility into its operations, making it difficult to make informed decisions. 3.Manual Processes: Many processes, such as inventory management, procurement, and order processing, were largely manual, leading to errors and delays. 4.Scalability: The existing systems were not scalable enough to support the company's growth plans and increasing transaction volumes.
2	The client is a prominent name in the fabrication industry having expertise in heavy industry fabrication, lifting equipment, and industrial doors in the European region. They specialize in warehouse loading and unloading docks along with the metal Fabrication equipment needed by the various industries. Their manufacturing process begins after a job site visit and obtaining all the necessary measurements from the client. However, they were facing communication issues due to their manual process of managing data in Excel sheets and in hard copies of paper. There was a lack of availability of production status across various departments, and obtaining data from each department was a manual process. This led to miscommunication and uninformed decision-making. Sometimes, different departments were not informed about new quotations or lacked information related to resources. To address these challenges, we implemented an ERP solution that automated the process of receiving quotations, project management, field services management, and product warranty module integration. Upon implementation of cloud cloud-based ERP system helped them get real-time data on a single dashboard and ensured that all departments working more efficiently. Challenges Faced By Metal Fabrication Industry ● The organization was facing challenges in processing quotations as per demand due to the distributed work among departments and a manual process. There was lack of communication between departments resulting in delays in sending the final quotation to clients. ● The progress status of projects is not readily available and requires manual updates from each department. This process is time-consuming and delays the overall progress of

	work. ● Field data, including various measurements required for metal fabrication industries including manufacturing lifting equipment, and industrial doors, are collected in hard copy and distributed among departments, which consumes a significant amount of time. ● The warranty for products cannot be determined instantly due to the manual process, and all documents need to be checked to figure out the warranty. ● The organization was lacking proper planning as there is no real-time update from each department, resulting in delays and inefficiencies. ● The manual workflow for quotation preparation is taking up a lot of time, and the inflow of requirements is high compared to the monthly basis on which clients use to prepare quotations. As a result, the organization is unable to meet the monthly demand for the project inflow.

BOS	Computer Application
Class	S.Y.B.C.A
Semester	IV
Subject Name	Enterprise Resource Planning
Subject code	PUSCA406P
Level Of the Subject	Advanced

Practical

Practical No.	Topic	No. of Lectures
1	Introduction to Zoho Erp .Configure basic Account Setup and Configuration.	2
2	Add and manage customer records in the CRM module	2
3	Create custom fields to capture additional customer information.	2
4	Create sales orders for customer purchases.Convert sales orders into invoices upon shipment or delivery.	2
5	Create tasks for Zoho CRM events.	2
6	Add leads from online forms and Add new leads to your mailing list.	2
7	Add new Zoho CRM contacts to your SMS app.	2
8	Create new deals in Zoho CRM when a document is signed and Export leads or contacts to a spreadsheet.	2
9	Create workflows to automate repetitive tasks and processes.	2
10	Record financial transactions such as invoices, payments, and expenses..	2
NO. OF LECTURES		20

BOS	Computer Science
Class	S.Y B.C.A
Semester	IV
Subject Name	Advanced Java
Subject code	PUSCA402
Level of the Subject	Medium

Course Objectives:

1.Understand the fundamentals of Swing, JDBC (Java Database Connectivity) & Multithreading.
2.Develop dynamic web applications using Java servlets/ JSP as the backbone for web application control.

Unit No.	Name of Unit	Topic No.	Name of Topic	Hours
1	Swing Components	1.1	Swing Components – I: Introduction to JFC and Swing, Features of the Java Foundation Classes, Swing API Components, JComponent Class, Windows, Dialog Boxes, and Panels, Labels, Buttons, Check Boxes, Menus, Pane, JScrollPane, Desktop pane, Scrollbars, Lists and ComboBoxes, Text-Entry Components.	10
		1.2	Swing Components – II: Toolbars, Implementing Action interface, Colors and File Choosers, Tables and Trees, Printing with 2D API and Java Print Service API.	
2	Threads and Multithreading	2.1	Threads and Multithreading, The Lifecycle of a thread, Creating and running threads, Creating the Service Threads, Schedules Tasks using JVM, Thread-safe variables, Synchronizing threads, Communication between threads	10

		2.2	JDBC: JDBC Introduction, JDBC Architecture, Types of JDBC Drivers, The Connectivity Model, The java.sql package, Navigating the ResultSet object's contents, Manipulating records of a ResultSet object through User Interface ,The JDBC Exception classes, Database Connectivity, Data Manipulation (using Prepared Statements, Joins, Transactions, Stored Procedures), Data navigation	
3	Servlet	3.1	Servlet : Introduction , Web Terminology, Servlet API, types of servlet, servlet life cycle, servlet examples. Servlet Collaboration : RequestDispatcher, sendRedirect	10
		3.2	Session Tracking : Session Techniques, Cookie class , Hidden form field , URL rewriting & HttpSession interface. CRUD Operations with servlet.	
4	Java Server Pages	4.1	Introduction To Java Server Pages: Why use Java Server Pages? Disadvantages Of JSP, JSP v\s Servlets, Life Cycle of a JSP Page, How does a JSP function? How does JSP execute? About Java Server Pages Getting Started With Java Server Pages: Comments, JSP Document, JSP Elements, JSP GUI Example.	10
		4.2	Action Elements: Including other Files, Forwarding JSP Page to Another Page, Passing Parameters for other Actions, Loading Java bean. Implicit Objects, Scope And El Expressions: Implicit Objects, Character Quoting Conventions, Unified Expression Language [UnifiedEl], Expression Language	
Total No.of Lectures				**40**

Course Outcomes:

1. Understand the basics of Java Foundation Classes (JFC) and Swing.
2. Implement simple GUI applications using Swing components and understand event handling.
3. Demonstrate the process of creating and running threads in Java, both by extending the Thread class and implementing the Runnable interface.

4. Learn about JDBC architecture, different driver types, connectivity models, and the essential `java.sql` package.
5. Implement Session Tracking and CRUD Operations using Servlets.
6. Collaborate between servlets using RequestDispatcher and sendRedirect.

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
4)The Java Tutorials of Sun Microsystems Inc.
5)"Java: The Complete Reference" by Herbert Schildt

CASE STUDY
tudy: Swing Components in Java wing is a powerful framework for creating window-based applications. Unlike AWT (Abstract wing Toolkit), Swing provides platform-independent and lightweight components. Let's delve into a o where a software development team is building a graphical user interface (GUI) application using components. *enario:* The team is developing a music player application. They need to design a sleek and iendly interface that allows users to browse their music library, create playlists, and control playback. components come to the rescue! JButton (Play/Pause Button): ○ The team decides to use a JButton for the play/pause functionality. ○ They customize the button's appearance, such as setting an icon for "play" and "pause." JComboBox (Genre Selection): ○ To filter music by genre, the team employs a JComboBox. ○ Users can select genres like "Rock," "Pop," or "Jazz."

ng a Student Database Management System

case study, we'll create a web-based application using Java Servlets and JDBC. Our goal is to e student information efficiently. Here are the steps we'll follow:

se Setup:

- We'll use a PostgreSQL database to store student details.
- Create a table named studentdetails with columns:
  - stuid: Student ID
  - stuname: Student Name
  - email: Email ID
  - phonenum: Phone Number
- Insert some sample data into the table.

BOS	Computer Application
Class	S.Y.B.C.A
Semester	IV
Subject Name	Advanced JAVA Practical
Subject code	PUSCA407P
Level Of the Subject	Advanced

Practical No	Details	Hours
1	**Swing Components-I :** **Assignment :** Design a simple calculator by using respective swing components & listeners. Tools : JFrame, JLabel , JButton , ActionListener	2
2	**Swing Components – I**: **Assignment**: Create a Java Swing application that displays a window with a label, a button, and a checkbox. When the button is clicked, show a dialog box with a message. **Tools**: Use JFrame, JLabel, JButton, and JCheckBox.	2
3	**Swing Components – II**: **Assignment**: Design a toolbar with buttons for common actions (e.g., save, open, print). Implement an action listener to handle button clicks **Tools**: Utilize JToolBar, JButton, and ActionListener.	2
4	**Multithreading**: **Assignment**: Create a multithreaded Java program that simulates a simple task scheduler. Implement thread-safe variables and demonstrate communication between threads. **Tools**: Use Thread, Runnable, and synchronization techniques.	2
5	**JDBC**:	2

	Assignment: Develop a Java application that connects to a database using JDBC. Retrieve data from a table and display it in a user interface (e.g., a list or table). **Tools**: Utilize java.sql package, ResultSet, and PreparedStatement.	
6	**Servlet Basics**: **Assignment**: Write a simple servlet that responds to HTTP requests. Display a welcome message when the servlet is accessed **Tools**: Use HttpServlet and implement the doGet method.	2
7	**Servlet Collaboration**: **Assignment**: Create two servlets. Use RequestDispatcher to forward a request from one servlet to another. Implement a redirection using sendRedirect. **Tools**: Utilize RequestDispatcher and HttpServletResponse.	2
8	**Session Tracking**: **Assignment**: Develop a web application that demonstrates different session tracking techniques. Use cookies, hidden form fields, and URL rewriting. **Tools**: Explore HttpSession, Cookie, and URL rewriting.	2
9	**CRUD Operations with Servlet**: **Assignment**: Build a servlet-based application that performs CRUD (Create, Read, Update, Delete) operations on a database table (e.g., user records). **Tools**: Combine servlets, JDBC, and database operations.	2
10	**CRUD Operations with JSP**: **Assignment**: Build a servlet-based application that performs CRUD (Create, Read, Update, Delete) operations on a database table (e.g., user	2

	records). **Tools**: Combine JSP, JDBC, and database operations.	
TOTAL NO OF LECTURES		**20**

BOS	Computer Science
Class	S.Y B.C.A
Semester	IV
Subject Name	Advanced Database Management Systems
Subject code	PUSCA403
Level of the Subject	Medium

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	10L
		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,	10L
		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	10L
		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	10L
		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:

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:

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.

BOS	Computer Science
Class	S.Y.B.C.A
Semester	III
Subject Name	Advanced Database Management Systems Practical
Subject code	PUSCA408P
Level Of the Subject	Medium

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	2

	a.Creating functions. b.Recursive functions.	
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.C.S.
Semester	IV
Course Name	Introduction to Data Science
Course Code	PUDSE404D
Level of the Subject	Moderate
Credit points	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,	10L
		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	10L
		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 Bigdata Analytics.	3.1	Introduction : Who uses PySpark Features , Advantages, PySpark Architecture , Modules and Packages	10L
		3.2	PySpark RDD : RDD creation, RDD operations PySpark DataFrame: Difference of 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	10L
		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. "Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking" by Foster Provost and Tom Fawcett
4. "Data Science: An Introduction" by Michael T. Lash
5. "Ethics of Big Data: Balancing Risk and Innovation" by Kord Davis

CASE STUDY	
1)	The provided data set appears to represent measurements of sepal and petal dimensions for various iris flowers, with the corresponding species labeled. Each row in the dataset represents a single observation or sample. Here's a summary of the dataset: 1.sepal_length: Length of the sepals in the iris flower. 2.sepal_width: Width of the sepals in the iris flower. 3.petal_length: Length of the petals in the iris flower. 4.petal_width: Width of the petals in the iris flower.

5.species: The species of the iris flower.

	sepal_length	sepal_width	petal_length	petal_width	species
2	5.1	3.5	1.4	0.2	setosa
3	4.9	3.0	1.4	0.2	setosa
4	4.7	3.2	1.3	0.2	setosa
5	4.6	3.1	1.5	0.2	setosa
6	5.0	3.6	1.4	0.2	setosa
7	5.4	3.9	1.7	0.4	setosa
8	4.6	3.4	1.4	0.3	setosa
9	5.0	3.4	1.5	0.2	setosa
10	4.4	2.9	1.4	0.2	setosa

2) 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.

	user	django	avenger
0	Sam	7.5	10.0
1	Max	7.0	7.0
2	Robert	7.0	8.0
3	Toby	9.0	8.5
4	Julia	6.0	10.0
5	William	8.0	6.0
6	Jill	6.5	7.0

BOS	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	IV
Course Name	Introduction to Data Science - Practicals
Course Code	PUDSE404D
Level of the Subject	Moderate
Credit points	3

Practical No	**Details**	**Hours**
1	Write a NumPy program to swap rows and columns of a given array in reverse order.	2
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	2
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.	2
4.	Perform various data transformation techniques on numerical and date time data.	2
5.	Perform Univariate , Bivariate and Multivariate graphical and non graphical analysis on Sample Superstore dataset.	2

6.	Practical to implement Data Manipulation using Pandas Techniques.(Handling of missing value , Renaming a column, Data Extraction, Categorical encoding)	2
7.	Setup Github Account, loading data from different source files formats (csv, excel) and summarizing data with statistics.	2
8.	Perform the Data Recommendation technique of some random data.	2
9.	Create a dataframe using PySpark and perform basic operations.	2
10.	Practical to implement case study on Analyzing Unstructured Data using Text Mining.	2
TOTAL NO. OF LECTURES		20

BOS	Computer Science
Class	S.Y.B.C.A
Semester	IV
Subject Name	UX/UI Design
Subject Code	PUDSE404U
Level of the Subject	Basic

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.	10L
		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	10L
		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, Wireframes & Prototyping	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 usage in visual design, Using colors, Textures, Forms, Design principles, Visual design tools.	10L

		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,	10L
		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. "Observing the User Experience: A Practitioner's Guide to User Research" by Mike Kuniavsky, Andrea Moed, and Elizabeth Goodman
4. "Interviewing Users: How to Uncover Compelling Insights" by Steve Portigal
5. "Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Rules" by Jeff Johnson

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.

BOS	Computer Science
Class	S.Y.B.C.A
Semester	IV
Subject Name	UX/UI Design - Practicals
Subject Code	PUDSE404U
Level of the Subject	Basic

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

BOS	**Computer Science**
Class	**S.Y.B.C.A**
Semester	**IV**
Course Name	**Full stack Development Paper-II**
Course Code	**PUDSE404F**
Level of the Subject	**Moderate**
Credit points	**2**

Course Objectives:

1. Understand the JavaScript and technical concepts behind Node JS
2. Understand and use the Event Emitter , Buffers, Streams, and Pipes

Unit No.	Name of Unit	Topic No	Content	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,	10

			Installing Node.js with NPM Creating Routes and Handling Requests	
		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 NUMBER 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.Node.js Design Patterns" by Mario Casciaro
2.Node.js Web Development" by David Herron
3."Express in Action" by Evan Hahn:
4.Node.js 8 the Right Way: Practical, Server-Side JavaScript That Scales" by Jim R. Wilson
5."Node.js Web Development: Server-side Development with Node 10 made easy, 4th Edition

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.

BOS	**Computer Science**
Class	**S.Y.B.C.A**
Semester	**IV**
Course Name	**Back End development with Node.js - Practicals**
Course Code	**PUDSE404F**
Level of the Subject	**Moderate**
Credit points	**2**

Practical No.	Details	Hours
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
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	2
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	2
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	2

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)	2
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	2
7	Node.js Crypto: 1. Generate a hash of a string using the Node.js crypto module	2
8	Node.js TLS/SSL: 1. Connect to a secure HTTPS website and print the response using the Node.js tls module	2
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	2
10	Performing CRUD operations using database (Create, Read, Update, Delete),	2
Total No of Lectures		20