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 Computer Science

S.Y.B.Sc.Computer Science

PCACS/BSCCS/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,	Industry Representative	Absent

	Security Solution Architect, cloud Google Google	(Industry/Corporate/Allied Sector)	
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	Mrs. Soly Zachariah	Faculty Specialization	
12	Mrs. Ramya S. Kumar	Faculty Specialization	
13	Mrs. Sujata Shahabade	Faculty Specialization	Sujata
14	Mrs. Sreevidya T.V.	Faculty Specialization	
15	Mr. Omkar Sherkhane	Faculty Specialization	
16	Mr. Abhijeet Salvi	Faculty Specialization	

1. INTRODUCTION TO BACHELORS IN COMPUTER SCIENCE (C.S.) PROGRAM

B.Sc. in Computer Science is a three years undergraduate programme that concentrates on creating links between theory and practice. Computer science is about understanding computer systems and networks at a deep level. It covers a wide variety of software and hardware technologies and their applications. Students will also gain practical problem solving and program design skills; Students learn how to think more carefully and how to solve problems more effectively. They will not only develop a diverse set of skills to prepare for their curriculum and for employment, but will also be encouraged to launch their own startups or venture into new types of careers using their interdisciplinary training. Our curriculum exposes students to modern advancements and new sub-fields of computer science. As we all know degrees in Computer Science lead to rewarding and lucrative careers, excellent placement and incubation assistance is provided.

Program outcomes

Sr No	PSO Title	POs in brief
PO1	Industry knowledge	Graduates would be fully equipped with the industry relevant domain knowledge.
PO2	Modelling to complex activities.	Apply appropriate techniques, resources, and modern tools including prediction and modelling to complex activities.
PO3	Career opportunities	Graduates would be proficient to take up a career of their choice whether it is corporate leadership, entrepreneurship or higher studies.
PO4	Develop Innovative Ideas	Graduates would be capable of using technology for developing innovative ideas to solve problems.
PO5	Design of conceptual system	Apply mathematical foundations, algorithmic principles, and analysis techniques in the modeling and designing of computational systems
PO6	Life- long learning	Engage themselves in lifelong learning to keep up with the pace of changing technology.
PO7	Soft skill Excellence	Graduates would be able to exhibit excellent soft skills and would be competent enough to work in teams to accomplish the desired goal.
PO8	Ethical responsibilities	Take ethical responsibilities, human and professional values and make their contribution to the society.

2. Program Specific Outcomes

Sr No	PSOs in brief
PSO1	Professionally trained in the areas of programming, multimedia, animation, web designing, Internet of things, networking and mobile app development.
PSO2	Ability to comprehend and write effective project reports in a multidisciplinary environment in the context of changing technologies.
PSO3	Develop an ability to use appropriate techniques, skills and tools required for computing problems.
PSO4	Ability to provide socially acceptable technical solutions to complex computer science problems with the application of modern and appropriate techniques.

Course Structure

		Semester III				
Course Code	Course Type	Course Title	Theory/ Practical	Marks	Credits	Lectures / Week
PUSCS301	MAJ	Operation Research	Theory	100	2	4
PUSCS302	MAJ	Data Structure Using Python	Theory	100	2	4
PUSCS303	MAJ	Operating System	Theory	100	2	4
PUSCS304D/ PUSCS304U/ PUSCS304F	DISCMIN	Data Visualization using python/Interaction Design/Full stack Development Paper-I	Theory/ Practical	100	3	3+2
PUSCS305	SEC	Mini Project	Theory	100	2	3
PUSCS306P	MAJ	Practicals(PUSCS301)	Practical	50	2	2
PUSCS307P	MAJ	Practicals(PUSCS302)	Practical	50	2	2
PUSCS308P	MAJ	Practicals(PUSCS303)	Practical	50	2	2
PUAEC30	AEC	Languages(to be taken from pool)	Theory/Practical	100	2	3
PUIDC30	IDC	To be taken from the Pool	Theory/ Practical	100	3	3+2
		Total		850	22	34
All Subjects having Field Project as part of Continuous Assessment-2						

Abbreviations:

SEC: Skill Enhancement Course
AEC: Ability Enhancement Course
IDC: Interdisciplinary Course

Course Structure

		Semester- IV				
Course Code	**Course Type**	**Course Title**	**Theory/ Practical**	**Marks**	**Credits**	**Lectures/ Week**
PUSCS401	MAJ	Advanced JAVA	Theory	100	2	4
PUSCS402	MAJ	Computer Networks	Theory	100	2	4
PUSCS403	MAJ	Advanced Database Management System	Theory	100	2	4
PUSCS404D	DISCMIN	Track1-Data Science Introduction to Data Science Track2-UI-UX UX-UI Design Track-3 Full Stack Development paper-2	Theory/ Practical	100	3	3+2
PUSCS405	SEC	Swayam course(NPTEL)	Theory	100	2	3
PUSCS406P	MAJ	Practicals(PUSCS401)	Practical	50	2	2
PUSCS407P	MAJ	Practicals(PUSCS402)	Practical	50	2	2
PUSCS408P	MAJ	Practicals(PUSCS403)	Practical	50	2	2
PUAEC40	AEC	Emotional Intelligence	Practical	100	2	2
PUIDC40	IDC	To be taken from the pool	Theory/ Practical	100	3	3+2
		Total		850	22	33
All Subjects having Field Project as part of Continuous Assessment-2						

Abbreviations:

SEC: Skill Enhancement Course
AEC: Ability Enhancement Course
IDC: Interdisciplinary Course

Evaluation Pattern

Marking Code	Marking Scheme
A	60 Marks Final Exam, 20 Marks Internal Exam, 20 Marks Project.
B	50 Marks Continuous Exam, 50 Marks Practical 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
PUSCS301	MAJ	Operation Research	A
PUSCS302	MAJ	Data Structure Using Python	A
PUSCS303	MAJ	Operating System	A
PUSCS304D/PUSCS304U/PUSCS304F	DISCMIN	Track1-Data Science Data Visualization using python Track2-UX-UI Interaction design Track-3 Full Stack Development I	A
PUSCS305	SEC	Mini Project	C
PUSCS306P	MAJ	Practicals(PUSCS301)	D
PUSCS307P	MAJ	Practicals(PUSCS302)	D
PUSCS308P	MAJ	Practicals(PUSCS303)	D
PUAEC30	AEC	Languages(to be taken from pool)	C

SEMESTER IV

Course Code	Course Type	Course Title	Evaluation Pattern
PUSCS401	MAJ	Advanced JAVA	A
PUSCS402	MAJ	Computer Networks	A
PUSCS403	MAJ	Advanced Database Management System	A
PUSCS404-	DISCMIN	Track1-Data Science Introduction to Data Science Track2-UI-UX UX-UI Design Track-3 Full Stack Development II	A
PUSCS405	SEC	Swayam course(NPTEL)	C
PUSCS406P	MAJ	Practicals(PUSCS401)	D
PUSCS407P	MAJ	Practicals(PUSCS402)	D
PUSCS408P	MAJ	Practicals(PUSCS403)	D
PUAEC40	AEC	Emotional Intelligence	C

SEMESTER III

BOS	Mathematics and Statistics
Class	S.Y. B.Sc. C.S.
Semester	III
Course Name	Operation Research
Course Type	Major
Course Code	PUSCS301
Level of the Course	Medium
Credits	4 Theory + 1 Practicals

Course Objectives:

1. Knowledge of formulating mathematical models for quantitative analysis of managerial problems in industry and use resources (capitals, materials, staffing, and machines) more effectively.
2. Skills in the use of Operations Research approaches and computer tools in solving real problems in industry. Mathematical models for analysis of real problems in Operations Research.

Unit No.	**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	Transportation and Assignment	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.	10
		2.3	Transportation Problems- Maximization & Minimization Type Problems. Balanced and Unbalanced problems. Prohibited Transportation Problems, Unique or Multiple Optimal Solutions.	
		2.4	InitialFeasibleSolution(IFS)by:a. NorthWestCornerRule(NWCR)b. LeastCostMethod(LCM)c.Vogel's Approximation Method(VAM).Maximum 5x5TransportationMatrix. Finding OptimalSolutionbyModified Distribution(MODI)Method.(u,vand Δ).MaximumTwoIterations(i.e. MaximumTwoLoops)afterIFS)	
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	Job Sequencing and Theory of Games	4.1	Job Sequencing Problem. Processing Maximum 9 Jobs through Two Machines only. Calculations of Idle Time, Elapsed Time etc.	10
		4.2	Processing Maximum 6 Jobs through Three Machines only. Calculations of Idle Time, Elapsed Time etc.	
		4.3	Theory of Games: Introduction Terminology of Game Theory: Players, Strategies, Play, Payoff, Payoff matrix, Maximin, Maximax, Saddle Point.	
		4.4	Numericals based on: Two Person Zero Sum Games Pure Strategy Games (Saddle Point available)	
			Total number of Lectures	**40**

Course Outcomes:

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, 3^{rd}.
5. Operations Research, P. K. Gupta D. S. Hira Sultan Chand & Sons.

<table>
<tr>
<td><b>1</b></td>
<td><b>CASE STUDY 1</b>
The data for a PERT network is given in the following table. Answer the following questions based on the data. (time in weeks)
<table>
<tr><th>Activity</th><th>Optimistic time
(a)</th><th>Most likely time
(m)</th><th>Pessimistic time
(b)</th></tr>
<tr><td>1-2</td><td>4</td><td>5</td><td>6</td></tr>
<tr><td>1-3</td><td>4</td><td>6</td><td>8</td></tr>
<tr><td>2-3</td><td>3</td><td>4</td><td>5</td></tr>
<tr><td>2-4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td>3-5</td><td>1</td><td>3</td><td>5</td></tr>
<tr><td>4-6</td><td>1</td><td>2</td><td>3</td></tr>
<tr><td>5-6</td><td>2</td><td>4</td><td>6</td></tr>
<tr><td>6-7</td><td>1</td><td>2</td><td>3</td></tr>
</table>
</td>
</tr>
<tr>
<td><b>2</b></td>
<td><b>CASE STUDY 2</b>
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.</td>
</tr>
</table>

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

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**

BOS	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	III
Course Name	Data Structure Using Python
Course Type	Major
Course Code	PUSCS302
Level of the Subject	Basic
Credit points	4 Theory + 1 Practicals

Course Objectives:

1 To Understand algorithms and its analysis procedure.

2 Introduce the concept of data structures through ADT including List, Stack, Queues and develop application 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 ArraysMulti-Array ADT, Implementing Multi Arrays .	
		1.3	**Algorithm Analysis:** Complexity Analysis-Big-O Notation, Evaluating Python Code,Evaluating Python List, Amortized Cost, Evaluating Set ADT, 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, More Ways to Build Linked Lists, Applications-Polynomials.	
		2.3	**Advanced Linked List**: Doubly Linked Lists-Organization and Operation,Circular Linked List-Organization and Operation, Multi Lists	
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, AVL tree.	10
		4.2	**Advanced Sorting and Heaps:** Merge Sort, Quick Sort, , And Heaps 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 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	The Bus reservation system facilitates the passengers to enquire about the bus available on the basis of source and destination, booking and cancellation of tickets, enquire about the status of the booked ticket, etc. The aim of the case study is to design and develop a data structure maintaining the records of different bus, bus status, and passengers. The record of a bus includes its number, name, source, destination, and days on which it is available, whereas record of bus status includes dates for which tickets can be booked, total number of seats available, and number of seats already booked. The bus reservation system performs following operations: 1. Searching bus- It enables the passengers to enquire about the bus available on the basis of source and destination 2. Booking ticket - Reserving seat(s) 3. Cancellation of ticket - Cancellation of a ticket to update the bus status

	4. Status of the bus - It includes dates for which tickets can be booked, total number of seats available, and number of seats already booked.
2	Mr. Rakesh is a freelancer who develop different kind of project according to the client requirement currently he is working on two different project : **Graphical Calculator** The Graphical Calculator project makes use of a variety of data structures and algorithms to construct a calculator that can handle mathematical expressions and display the results in a graphical user interface (GUI). **College Data Management** Colleges have multiple departments where every department offers many courses. These departments have a head (HOD) and various instructors. Even though there are many instructors, one instructor can only work in one department. As you can see the organization structure of a college is quite complicated and requires a lot of effort to manage.

Practicals

No.	Details	CO
1	Implement Linear Search to find an item in a list.	CO5
2	Implement binary search to find an item in an ordered list.	CO5
3	Implement Sorting Algorithms for Bubble Sort & insertion sort.	CO2, CO3
4	Implement Sorting Algorithms for quicksort and merge sort.	CO2, CO3
5	Implement use of Sets and various operations on Sets.	CO1
6	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)	CO4
7	Implement Program for -	
a	Infix to Postfix conversion	CO4
b	Postfix Evaluation	CO4
8	Implement the following	

a	A queue as a list which you add and delete items from.	CO1, CO4
b	A circular queue. (The beginning items of the queue can be reused).	CO4
9	Implement Linked list and demonstrate the functionality to add and delete items in linked list	CO1, CO4
10	Implement Binary Tree and its traversals.	CO6
11	Recursive implementation of	
a	Factorial	CO4
b	Fibonacci	CO4
c	Tower of Hanoi	CO4
12	Write a program to display the adjacency matrix for a given Graph.	CO1
	Total No. of Lectures	20

BOS	Computer Science
Class	S.Y.B.Sc.C. S.
Semester	III
Course Name	Operating System
Course Type	Major
Course Code	PUSCS303
Level of the Subject	Moderate
Credit points	4 Theory + 1 Practicals

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	File System, Introduction to Linux	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	Linux Basic , Shell , Process and Advance Commands.	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.
5. Silberschatz, A., Galvin, P.B., & Gagne, G. (2018). Operating System Concepts (10th ed.). Wiley.

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: he 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. ortability: 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

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.C.S
Semester	III
Course Name	Data Visualization using Python
Course Type	Discipline Minor
Course Code	PUSCS304D
Level of the Subject	Medium
Credit points	4 Theory + 1 Practical

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 Visualization 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

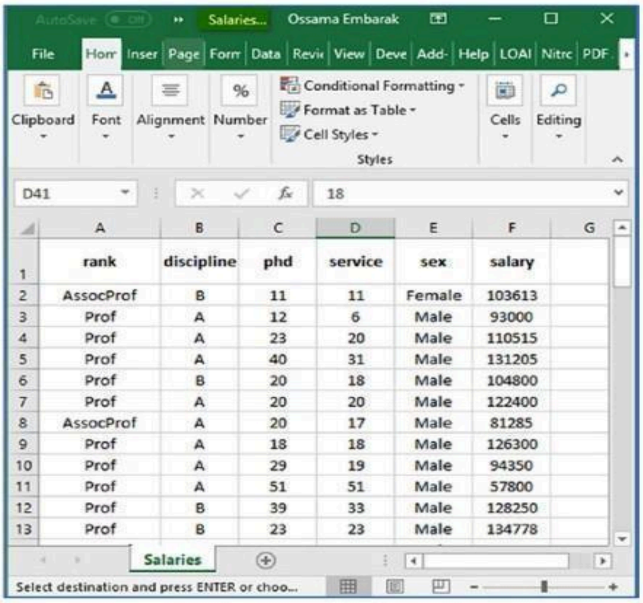
References:

1. Dr. Ossama Embarak, "Data Analysis and Visualization using Python", APress
2. Igor Milovanović , Dimitry Foures , Giuseppe Vettigli, "Python Data Visualization
3. Cookbook", Packt Publishing
4. Healy, K. (2019). Data Visualization: A Practical Introduction. Princeton University Press.
5. Kandel, S., Heer, J., Plaisant, C., & Kennedy, J. (2012). Designing Interactive Visualization Tools to Support Creativity and Insight. ACM.
6. Murray, S. (2017). Interactive Data Visualization for the Web: An Introduction to Designing with D3. O'Reilly Media.

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:



	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 Gnatt charts.
Total No. of Lectures	**20**

BOS	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	III
Course Name	Interaction Design
Course Type	Discipline Minor
Course Code	PUSCS304U
Level of the Subject	Intermediate
Credit points	4 Theory + 1 Practical

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	10
		4.2	Introduction, AgileUX, Design Patterns, Open Source Resources,Tools for Interaction Design	
	Total Lectures			60

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. Cooper, A., Reimann, R., & Cronin, D. (2014). About Face: The Essentials of Interaction Design (4th ed.). Wiley.
3. Preece, J., Rogers, Y., & Sharp, H. (2015). Interaction Design: Beyond Human-Computer Interaction (4th ed.). Wiley.
4. Norman, D.A. (2013). The Design of Everyday Things: Revised and Expanded Edition. Basic Books.
5. Tognazzini, B., & Nielsen, J. (2001). Tog on Interface. Addison-Wesley Professional.

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	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	III
Course Name	Full stack Development Paper-I
Course Type	Discipline Minor
Course Code	PUSCS304F
Level of the Subject	Medium
Credit points	4 Theory + 1 Practical

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	10
			its benefits, Environment setup for React development, Refresher on ES6 concepts,	
			Create React App, Folder Structure.	
		1.2	**Templating using JSX:** Understanding	
			component architecture and its	
			significance, Introduction to components	
			And their types, Working with React create Element 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,	
			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 system and using synthetic events. Passing arguments to event handlers. Understanding the significance of lifecycle methods in handling events.	
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 Hooks: In-depth exploration of useState and useEffect hooks.Understanding useContext and useReducer hooks, Creating custom hooks.	

		3.3	Routing with React Router: Setting up React Router for navigation in singlepage 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 :
Travelduck 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:

Travelduck's 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.

Case study 2 :

SnowShow is the most prominent Polish winter tour provider, and also one of the biggest in Europe. They target 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 name**
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	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	IV
Course Name	Advanced Java
Course Type	Major
Course Code	PUSCS401
Level of the Subject	Advanced
Credit points	4 Theory + 1 Practical

Course Objectives:

1. To explore advanced topics of Java programming for solving problems.
2. To enable students to develop web-based applications and enterprise applications.

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Servlet	1.1	Introduction to Java Servlets: The Need for Dynamic Content, Java Servlet Technology. Servlet API: Interfaces & Classes in javax.servlet package and javax.servlet.http package. Servlet Types and Lifecycle: Servlet Interface, GenericServlet, HttpServlet, The Servlet Life Cycle, Reading HTML Form Data in Servlet	10
		1.2	Working With Servlets: Using Deployment , Descriptor (web.xml), Using Annotations (@WebServlet). Request Dispatcher: Resquest Dispatcher, Request dispatcher interface, Methods of request dispatcher, request dispatcher applications Cookies: kinds of cookies, where cookies are used, creating cookies using servlets Session : What are sessions, life cycle of HTTP session , Aservelet session example	

2	JSP	2.1	Working With Servlets: Using Deployment Descriptor (web.xml), Using Annotations (@WebServlet). Request Dispatcher: Resquest DispatcherInterface, Methods of Request Dispatcher, Request Dispatcher Application. COOKIES: Kinds of Cookies, Where Cookies Are Used? Creating Cookies Using Servlets. SESSION: What Are Sessions? Lifecycle Of Http Session, A Servlet Session Example.	10
		2.2	JSP Actions: jsp:include, jsp:useBean, jsp:setProperty, jsp:getProperty, Jsp:forward, jsp:plugin, jsp:element, jsp:attribute, jsp:body, jsp:text Expression Language: Introduction ofExpression Language, Implicit objects of EL, Precedence of operators in EL, Reserved words in EL.	
3	JDBC & JavaBean	3.1	Introduction to JDBC: Introduction, Architecture, Types of Drivers, Driver Manager class, Connection interface. Types of Statements: Statement, Prepared Statement, Callable Statement. ResultSet & its Types: ResultSet, Read Only ResultSet, Updatable ResultSet, Forward Only ResultSet, Scrollable ResultSet.	10
		3.2	Working with Metadata: ResultSet Metadata, Database Metadata, SavePoint, Batch Processing, Storge Images (BLOB) & Store Files (CLOB), Transaction Management. Java Beans: Introduction, why use Java Bean? Java Bean properties, Advantages, Disadvantages, JavaBean Example.	
4	Struts Framework	4.1	Struts 2 Introduction: Basic MVC, Architecture, Struts 2 framework, Core Components: Interceptors, ValueStack, OGNL, ActionContext, ActionInvocation,	10
		4.2	Struts 2 Action: Action Interface, Action Support Class. Struts 2 Configuration File:	

			struts.xml: package element, action element, result element. Struts 2 Validation: Custom & Bundled.	
Total No. of Lectures				40

Course Outcomes:

1. Enumerate the concept of request-response architecture.
2. Demonstrate the Database connectivity with GUI applications.
3. Develop the JSP applications to solve real life problems.
4. Analyze the concept of JavaBean with examples.
5. Build various Servlet applications to solve real life problems.
6. Elaborate the concept of Struts and construct its applications.

References:

1. Cay S. Horstmann, Gary Cornell, Core Java™
2. Volume II–Advanced Features Prentice Hall PTR,9th Edition
3. Herbert Schildt, Java2: The Complete Reference, Tata McGraw-Hill,5th Edition
4. Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD) ,3rd Edition
5. The Java Tutorials: http://docs.oracle.com/javase/tutorial/
6. STRUTS 2 IN ACTION, By Donald Brown, Chad Michael Davis, Scott Stanlick · 2008

Case Studies

1. A university's computer science department wants to develop an Online Bookstore Management System to facilitate the ordering and management of books for students and faculty. The university already has a database containing information about books, authors, and publishers. Your task is to design a JDBC-based system to interact with the database and implement the necessary functionalities for the Online Bookstore Management System.
2. A university has decided to implement an Online Learning Management System to enhance its educational offerings. The system aims to provide a platform for instructors to create and manage online courses, students to enroll and participate in these courses, and administrators to oversee the entire learning process.

Practicals

Practical No	Details
1.	a. Create a simple calculator application using servlet. b. Design a simple form using HTML which takes a number as an input & perform following operations i. Calculate Factorial ii. Determine Prime Number or Not. iii. Determine Number is Armstrong or Not.
2.	a. Using RequestDispatcher Interface create a Servlet which will validate the password entered by the user, if the user has entered "Servlet" as password, then he will be forwarded to Welcome Servlet else the user will stay on the index.html page and an error message will be displayed. b. Create a servlet that uses Cookies to store the number of times a user has visited the servlet. c. Create a servlet demonstrating the use of session creation and destruction.
3.	a. Create a registration servlet 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. b. Design database for student administration. Develop servlet(s) to perform CRUD operations.
4.	a. Develop a simple JSP application to display values obtained from the use of intrinsic objects [Implicit Objects] of various types. b. Develop a simple JSP application to pass values from one page to another with validations. (Name-txt, age-txt, hobbies-checkbox, email-txt,genderradio button)
5.	a. Create a JSP page to demonstrate the use of Jsp Actions. b. Create a JSP page to demonstrate the use of Expression language.
6.	a. Create a registration and login JSP application to register and authenticate the user based on username and password using JDBC. b. Create Employees table in EMP database. Perform select, insert, update, and delete operations on the Employee table using JSP.

7.	a. Design a JDBC application using Servlet/JSP to demonstrate the concept of ResultSetMetaData. b. Design a JDBC application using Servlet/JSP to demonstrate the concept of DatabaseMetaData.
8.	a. Create a JSP page to demonstrate the use of BLOB & CLOB. b. Write a Student class with three properties. The useBean action declares a JavaBean for use in a JSP. Write a Java application to access JavaBeans Properties.
9.	a. Design application using Struts2. Application must accept the user name and greet the user when the command button is pressed. b. Design an application using Struts2 to demonstrate the use of Interceptors.
10.	a. Design an application using Struts2 to demonstrate the use of Custom Validators. b. Design an application using Struts2 to demonstrate the use of Bundled Validators.

BOS	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	IV
Course Name	Computer Networks
Course type	Major
Course Code	PUSCS402
Level of the Subject	Medium
Credit points	4 Theory + 1 Practical

Course Objectives:

1. The course introduces main concepts of networking and the layers of different models.
2. To develop an understanding of different components of computer networks, various protocols, modern technologies and their applications.

Unit No.	Name of Unit	Topic No.	Content	Hours
1	Introduction	1.1	Introduction Network Models: Introduction to data communication, Components, Data Representation, Data Flow, Networks, Network Criteria, Physical Structures, Network types, Local Area Network, Wide Area Network, Switching, The Internet, Accessing the Internet, Standards and Administration Internet Standards.	10
		1.2	Protocol Layering: Network Models: OSI Layered Architecture, TCP/IP Protocol Suite,	

		1.3	Signals: Data and Signals, Analog and Digital Data, Analog and Digital Signals, Sine Wave Phase, Wavelength, Time and Frequency Domains, Composite Signals, Bandwidth, Digital Signal, Bit Rate, Bit Length	
		1.4	Transmission of Digital Signals, Transmission Impairments, Attenuation, Distortion, Noise, Data Rate Limits, Performance, Bandwidth, Throughput, Latency (Delay)	
2	Introduction to Physical Layer	2.1	Introduction: Digital Transmission digitalto- digital conversion, Line Coding, Line Coding Schemes, analog-to-digital conversion, Pulse Code Modulation (PCM)	10
		2.2	Transmission modes and Modulation: Transmission Modes, Parallel Transmission, Serial Transmission. Analog Transmission, digital-to-analog Conversion, Aspects of Digital-to-Analog Conversion, Amplitude Shift Keying, Frequency Shift Keying, Phase Shift Keying, analog-toanalog Conversion, Amplitude Modulation (AM), Frequency Modulation (FM), Phase Modulation (PM)	
		2.3	Multiplexing:Multiplexing, frequency-Division Multiplexing, Wavelength-Division Multiplexing TimeDivision Multiplexing.	
		2.4	Transmission Media, Guided Media, Twisted- Pair Cable, Coaxial Cable, Fiber-Optic Cable. Switching, Three Methods of Switching , Circuit Switched Networks, Packet Switching	

3	Introduction to DataLink Layer	3.1	Introduction to Data-Link Layer: Services, Two Sub-layers: MAC, LLC Three Types of addresses, Address Resolution Protocol	
		3.2	Error Detection and Correction: Introduction, Types of Errors, Redundancy, Detection versus Correction, Simple parity check code, Hamming distance, Hamming code, CRC, Checksum	
		3.3	Media Access Control (MAC), random access, CSMA, CSMA/CD, CSMA/CA, controlled access, Reservation, Polling, Token Passing, channelization, FDMA, TDMA, CDMA.	
4	CSMA , Addressing and Routing	4.1	Connecting Devices and Addressing: Virtual LANs, connecting devices, Hubs, Link-Layer Switches, Routers,	10
		4.2	Routing : Unicast Routing, Least- Cost Routing, Routing Algorithms: Distance Vector Routing, Link-State Routing, Path Vector Routing	
		4.3	Introduction to Transport Layer, Transport-Layer Services, Connectionless and Connection- Oriented Protocols. UDP: Service, Port Numbers, User Datagram Protocol User Datagram TCP: Service, Port Numbers, TCP Features, Segment format, TCP connections, Error Handling in TCP	
		Total Number of Lectures		**40**

Course Outcomes:

1. Identify basic computer network technology.
2. Explain Data Communications System and its components.

3. Compare the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
5. Apply different types of network devices and their functions within a network .
6. Build the skills of subnetting and routing mechanisms.

References:

1. Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2013.
2. Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson Education, 2011.
3. Computer Network, Bhushan Trivedi, Oxford University Press
4. Data and Computer Communication, William Stallings,
5. PHI STRUTS 2 IN ACTION, By Donald Brown, Chad Michael Davis, Scott Stanlick · 2008

CASE STUDY

1) XYZ Corp, a rapidly growing enterprise with multiple departments and locations, experienced inefficiencies in communication and data sharing due to its outdated network infrastructure. As the company expanded, the existing network struggled to accommodate increased traffic, leading to slow data transfer speeds and frequent downtime. To address these issues and support future growth, XYZ Corp decided to overhaul its computer network.
2) TechHub Solutions, a growing startup with a team of 20 employees, recognized the need for a reliable and efficient network infrastructure to facilitate smooth communication and data sharing. With limited resources and technical expertise, the company aimed to implement a basic computer network solution that aligned with its operational requirements and budget constraints.

Practicals

Practical No	Details
1.	Understanding the working of NIC cards, Switches, Hub, Gateway router
2.	Crimping of Twisted-Pair Cable with RJ45 connector for Straight Through and Cross-Over, Roll-Over.
3.	Using Packet Tracer, Create a basic network of different devices(Router, Switch,PC) using appropriate network wire.
4.	Problem solving with IPv4, which will include concept of Classful addressing.
5.	Using linux-terminal or Windows-cmd, execute the following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig.
6	Using Packet Tracer, Configure the network with static routing using protocol.
7	Using Packet Tracer, Configure the network with dynamic routing using protocol.
8	Using Packet Tracer, connect a network with three layer two switches and every switch will have four computers. Verify their connectivity with each other.
9	Using packet Tracer ,create wireless network of multiple PCs using appropriate access points to block /unblock wireless connection.
10	Using wireshark ,network analyzer ,set the filter for ICMP,TCP,HTTP,UDP,FTP and perform respective protocol transactions to show /prove that the network analyzer is working .
	Total no of Lectures :20

BOS	Computer Science
Class	S.Y.B.Sc.C.S
Semester	IV
Subject Name	Advanced Database Management System
Course Type	Major
Subject code	PUSCS403
Level of The Subject	Medium
Credits	4 Theory + 1 Practical

Course Objective:

1. To Introduce basic concepts of PL/SQL with Advanced Database Techniques.
2. Understand the advances in database technology using No Sql and MongoDB.

Unit No.	**Name of Unit**	**Topic No.**	**Content**	**Hours**
1	Fundamentals of PL/SQL	1.1	Introduction to PL/SQL: PL/SQL Overview, Benefits of PL/SQL, SQL Identifiers List the different Types of Identifiers in a PL/SQL subprogram, Identify Scalar Data Types, The %TYPE Attribute, Bind Variables, Sequences in PL/SQL Expressions.	10
		1.2	Write Executable Statements:Describe Basic PL/SQL Block Syntax Guidelines, Comment Code, Deployment of SQL Functions in PL/SQL, Nested Blocks, Identify the Operators in PL/SQL.	
		1.3	Control Structures: Conditional processing UsingIF Statements, Conditional processing Using CASE Statements, Use simple Loop Statement, Use While Loop Statement, Use For Loop Statement, Describe the Continue Statement	

2	DBMS with PL/SQL	2.1	Stored Procedures and Functions: Create a Simple Procedures, Create a Simple Procedure with IN parameter, Create a Simple Function, Execute a Simple Procedure, Execute a Simple Function.	10
		2.2	Cursor: Introduction to Cursor ,Types of Cursor,Attributes of Cursor,Cursor FOR loop, Explicit Cursor Attributes, FOR UPDATE Clause and WHERE CURRENT Clause	
		2.3	Exception Handling :Understand Exceptions, Handle Exceptions with PL/SQL, Trap Predefined Oracle Server Errors, Trap Non-Predefined Oracle Server Errors, Trap User-Defined Exceptions,	
3	Advanced Database Techniques	3.1	Package: Components of a Package, Package Specification and Body, Package Constructs,	10
		3.2	Triggers: Concepts of Triggers, Create Trigger, Insert Trigger and Delete Trigger Statement, Statement Level Triggers Versus Row Level Triggers, Create Instead of and Disabled Triggers, Managing Testing and Removing Triggers.	
		3.3	NoSQL Database Concepts: Structured Vs Unstructured data, NoSQL Database concepts:NoSql Databases,NoSql Data Modeling, Benefits of NoSql,Sql Vs NoSql Database Systems.	
4	Advances in Databases	4.1	NoSql Using MongoDB: Introduction to MongoDB Shell,MongoDB Client,Basic Operation with MongoDB Shell,basic datatypes.	10

		4.2	Introduction to Data Warehouse: Characteristics, Types of Data Warehouse, Data Warehouse architecture, Data Warehouse lifecycle,Data Warehouse development.	
		4.3	Introduction to Multimedia Databases ,Mobile Databases ,Digital Databases.	
	Total No. of Lectures			40

Course Outcomes:

1. Describe fundamental concepts of PL/SQL & Advanced Database Techniques.
2. Explore database Management Systems with PL/SQL..
3. Demonstrate PL/SQL ,Advanced Database Management System Techniques.
4. Apply fundamental concepts of PL/SQL..
5. Manipulate Data using PL/SQL,MongoDB Commands.
6. Research, analyze and use emerging technologies such as NoSQL, On-Line Analytical Processing (OLAP) and Data Warehouses

References:

1. Ramakrishnam, Gehrke, “Database Management Systems”, McGraw- Hill.
2. Ivan Bayross, “SQL,PL/SQL -The Programming language of Oracle”, B.P.B. Publications
3. Michael Abbey, Michael J. Corey, Ian Abramson, Oracle 8i – A Beginner’s Guide, TataMcGraw-Hill.
4. Joel Murach, Murach’s MySQL, Mike Murach & Associates
5. A. Silberschatz, H. Korth, S. Sudarshan, Database system concepts, 5/e, McGraw Hill, 2008

Case Study:

1. XYZ hospital is a multi specialty hospital that includes a number of departments, rooms, doctors, nurses, compounders, and other staff working in the hospital. Patients having different kinds of ailments come to the hospital and get checkup done from the concerned doctors. If required they are admitted in the hospital and discharged after treatment. The aim of this case study is to design and develop a database for the hospital to maintain the records of various departments, rooms, and doctors in the hospital. It also maintains records of the regular patients, patients admitted in the hospital, the check up of patients done

by the doctors, the patients that have been operated, and patients discharged from the hospital.

2. The Career Advising Center at a University would like a “Career Advising Website” to be created so they can better manage the student requests for career advising appointments. The Career Advising Center would like to assign appropriate Industry Advisors with students based on career area.

   At various times, the Career Advising Center would want together information about the number of advising sessions that took place,the number of advising sessions that took place per career area, the total number of unique students who requested advising sessions (students can request multiple advising sessions during an academic year), the average rating given, or the Industry Advisor who was rated the highest, etc.

Practicals

Practical No	Details
1.	Writing Anonymous PL/SQL Block with basic programming construct by including following: a.Sequential Statements b. unconstrained loop c.% TYPE d.%ROWTYPE
2.	Writing PL/SQL Blocks with basic programming constructs by including following: If...then...Else, IF...ELSEIF...ELSE... END IF Case statement FOR-LOOP WHILE -LOOP
3.	Writing Exception Handling with PL/SQL. a. Exception Types (implicitly raised, Explicitly raised) a. Trapping Exceptions (WHEN exception1, WHEN OTHERS) b. Predefined Exception ● NO_DATA_FOUND ● TOO_MANY_ROWS ● INVALID_CURSOR ● ZERO_DIVIDE ● DUP_VAL_ON_INDEX
4.	Writing Procedures in PL/SQL Block (IN, OUT, INOUT parameter) a. Create an empty procedure, replace a procedure and call procedure a. Create a stored procedure and call it b. Define procedure to insert data c. A forward declaration of procedure
5.	Writing Functions in PL/SQL Block. a. Define and call a function a. Define and use function in select clause, b. Call function in dbms_output.put_line c. Recursive function d. Count Employee from a function and return value back e. Call function and store the return value to a variable

6.	Writing PL/SQL Block for: a. Declare and use Association Array b. Varray c. Nested Tables
7.	writing PL/SQL Block for Cursors a. Cursor attributes:%ROWCOUNT,%FOUND,%NOTFOUND,%ISOPEN a. Cursor with sub queries b. Combination of PL/SQL, cursor and for loop c. Parameterized cursors, Cursor Variables
8.	Study of transactions and locks.
9.	Creating and Handling Deadlock situations.
10.	Packages 1: a. Working with oracle supplied packages like DBMS_OUTPUT , etc b.Forward Declaration of packages Packages 2: a. Create and invoke a package that contains private and public constructs. b.Implement Package Functions in SQL
11	Study Basics of MongoDB: ● Start the mongo shell ● Quit the mongo shell ● Display database and collection
	Total No. Of Lectures :20

BOS	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	IV
Course Name	Introduction to Data Science
Course Type	Discipline Minor
Course Code	PUSCS404D
Level of the Subject	Moderate
Credit points	4 Theory + 1 Practical

Course Objectives:

1. Understanding basic Data Science concepts,Learning various Machine Learning Techniques to Predict the Data.
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 Bigdata Analytics.	3.1	Introduction : Who uses PySpark Features , Advantages, PySpark Architecture , Modules and Packages	10
		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	4.1	Introduction to Machine learning: Different types of Machine Learning, Linear Regression, Logistic Regression, K-means Clustering, Hierarchical Clustering	10
	Recommendation systems	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:

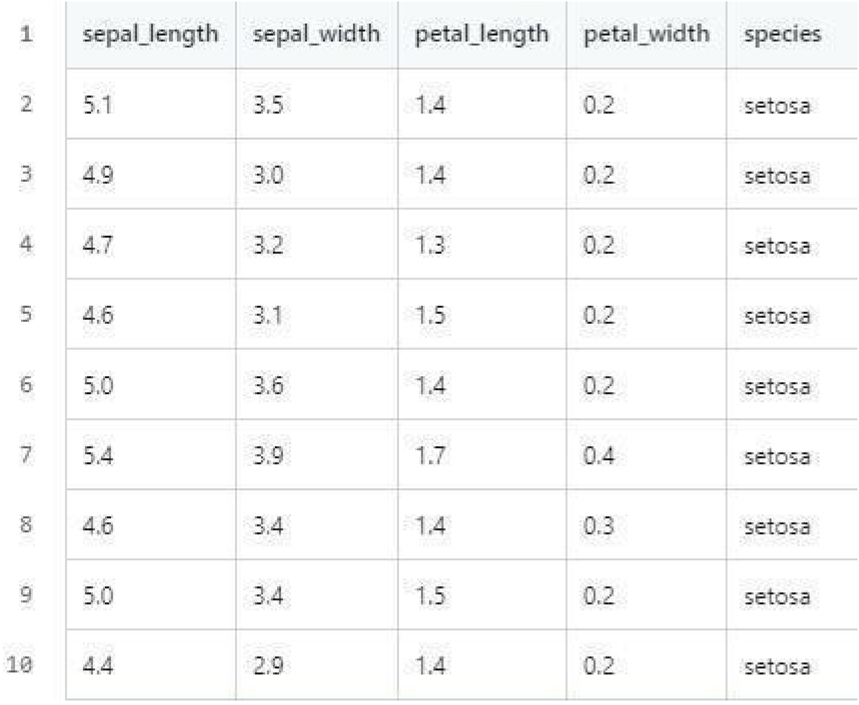
1. Enumerate the Various types of Data and Data Analysis Techniques.
2. Illustrate the Various Data transformation techniques.
3. Apply the various functions on data for data cleaning and exporting.
4. Illustrating the various data handling techniques using PySpark.
5. Analyzing the data with various 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.
6. VanderPlas, J. (2016). Python Data Science Handbook: Essential Tools for Working with Data. O'Reilly Media.

CASE STUDY 1

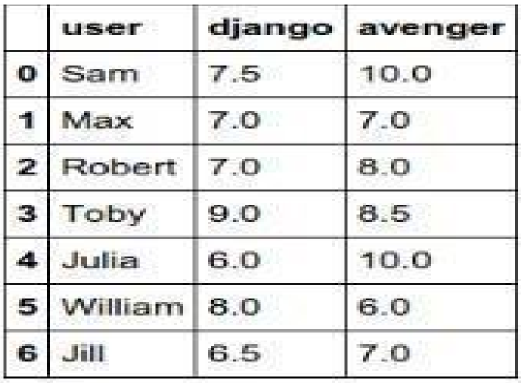
The provided data set appears to represent measurements of sepal and petal dimensions for various iris flowers, with the corresponding species labeled. Each row in the dataset represents a single observation or sample. Here's a summary of the dataset:1.sepal_length: Length of the sepals in the iris flower.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



1	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

CASE STUDY 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

Practicals

Practical No	Details	CO Mapping
1.	Write a NumPy program to swap rows and columns of a given array in reverse order.	CO2
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	CO2
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.	CO2
4.	Perform various data transformation techniques on numerical and date time data.	CO3
5.	Perform Univariate , Bivariate and Multivariate graphical and non graphical analysis on Sample Superstore dataset.	CO3
6.	Practical to implement Data Manipulation using Pandas Techniques.(Handling of missing value , Renaming a column, Data Extraction, Categorical encoding)	CO3
7.	Setup Github Account, loading data from different source files formats (csv, excel) and summarizing data with statistics.	CO4
8.	Perform the Data Recommendation technique of some random data.	CO5
9.	Create a dataframe using PySpark and perform basic operations.	CO5

10.	Practical to implement case study on Analyzing Unstructured Data using Text Mining.	CO6
	TOTAL NO. OF LECTURES : 20	

BOS	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	IV
Course Name	UX-UI Design
Course Type	Discipline Minor
Course Code	PUSCS404
Level of the Subject	Moderate
Credit points	4 Theory + 1 Practical

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, 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.	10
		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

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.

Practicals

Practical No	Details	
1.	Introduction to Figma Tool	CO1
2.	Design a UX prototype to demonstrate Font, Color, Typography.	CO2
3.	Create UX design for Login Page.	CO4
4.	Perform Boolean Operations and Design Logo.	CO4
5.	a) Perform User Research (for online shopping /any domain specific) b) System Concept Statement (prepare short summary report on what approach used)	CO3
6.	User Requirement Analysis (Draw Flow Model).	CO3
7.	Create User Personas and User Scenario.	CO2
8.	Create a Site map, Wireframe, Screens, Widgets.	CO3
9.	Setting Properties, Screen transitions , Header and Footer.	CO5
10.	Perform Usability Testing .	CO6
	Total No. Of Lectures	**20**

BOS	Computer Science
Class	S.Y.B.Sc.C.S.
Semester	IV
Course Name	Full stack Development Paper-II
Course Type	Discipline Minor
Course Code	PUSCS404
Level of the Subject	Moderate
Credit points	4 Theory + 1 Practical

Course Objectives:

1. Understand the JavaScript and technical concepts behind Node JS ,Understand and use the Event Emitter , Buffers, Streams, and Pipes
2. Build a Web Server in Node and understand how it really works ,Connect to a SQL or Mongo database in Node Understand how the MEAN stack works

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.	10
			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.,	
		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	Node.js Punycode: What is Punycode, encode, decode, toASCII, toUnicode Node.js TTY : Classes(ReadStream, WriteStream), examples.	
		3.3	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."Node.js Design Patterns" by Mario Casciaro and Luciano Mammino

2. "Node.js in Action" by Mike Cantelon, Marc Harter, and T.J. Holowaychuk

3."Fullstack Node.js: The Complete Guide to Building Production Apps with Node.js and MongoDB" by David Gochberg

4."Node.js Web Development" by David Herron

5."Node.js the Right Way: Practical, Server-Side JavaScript That Scales" by Jim R. Wilson

6. "Node.js Essentials - Second Edition" by Fabian Cook, Indrajit Das, and Arseniy Maksimov
7. "Learning Node.js Web Development" by Andrew Mead and Michael Scott

Case Studies:
The case study will involve creating a simple RESTful API for a book library management system.
A book library management system needs to be created that allows users to perform CRUD (Create, Read, Update, Delete) operations on books in a library's database. The system should also provide endpoints for managing user data and user authentication.

Potential Challenges

- Database Design: Ensure that the MongoDB collections are designed efficiently for the operations the application will perform.
- Authentication and Authorization: Implement secure authentication and authorization practices to protect user data and API endpoints.
- Error Handling: Handle errors gracefully and provide meaningful responses to clients.

Case Study: Food Delivery Service Application

A food delivery startup aimed to create an online platform that connects restaurants with customers. The application needed to provide customers with a user-friendly interface to browse restaurants, place orders, and track deliveries. Additionally, the platform needed to handle restaurant management, including menu updates and order processing.
The application offered the following features:

- User Authentication: Customers and restaurant staff could log in using JWT-based authentication.
- Restaurant Management: Restaurants could manage their menus, prices, and availability.
- Order Placement: Customers could browse restaurants and place orders.
- Order Tracking: Customers could track their orders in real-time using WebSocket connections.
- Payment Integration: The application integrated with a payment gateway to handle transactions.

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),

****