INDEX

SR NO.	NAME OF PROJECT	DATA STRUCTURE IMPLEMENTED
1.	RESTAURANT LOGIN , ORDERING AND BILLING SYSTEM	STRUCT
2.	BANK MANAGEMENT SYSTEM WITH FINANCE ADVISOR	STACK
3.	GENERAL KNOWLEDGE QUIZ GAME	QUEUE
4.	SINGING COMPETITION – CONTESTANTS AND AUDIENCE LOGIN +VOTING	LINKED LIST
5.	STUDENT DATABASE MANAGEMENT SYSTEM IN ASCENDING FASHION	TREES
6.	SHORTEST PATH CALCULATION FOR COURIER SERVICE	GRAPHS

TOPIC: **STRUCT**

- SUBJECT: Restaurant Login, Ordering and Billing System

- DESCRIPTION:

This project consists of a following features:

- ✓ Restaurant login
- ✓ Food Ordering
- ✓ Billing System

Here a user enters the restaurant portal and enters the *name, email id, phone number* and a *password* is generated for the user in the portal and he can order now from the menu.

First, we have created struct of food item with the help of nested structs and initialized them in a function named menu and the menu details are also printed in this function so the user can choose from a delicious choice of menu.

Functions used :-

- menu()
- bill()
- passswordgenerator()

Then we have created a function called billing for calculating the total cost generated according to the choices of user and we print the bill and show to user. Price of each item is initialized here and is calculated including GST using appropriate mathematical formulas.

Then we have created a function that generates password for user on random basis using rand function. By this way a complete billing system is created with the help of struct in command line.

TOPIC: **STACK**

- SUBJECT: Bank Management System with Finance Advisor

- DESCRIPTION:

Here we are creating a Bank Management System that has following features:-

- Create new account
- Update information of existing account
- For transactions
- Checking details of existing account
- View customer’s list
- Exist

Now according to the choices of user we are providing financial advice on the basis of the status of the bank account of that user.

To start with, we created structs for various details of the user details and their account details. Now in this project we are using stack data structure which is based on LIFO principle that basically says that the element added last is taken out first . One of the well known use of stack is for Polish expressions .

Functions used:-

• fordelay(int j) • new_acc() • view_list() • edit(void) • transact(void) • erase() • see(void) • interest(float t,float amount,int rate)	• menu(void) • isempty() • isfull() • traverse() • peek() • pop() • push(int data) • call(int c)

Polish Notation in the data structure is a method of expressing mathematical, logical, and algebraic equations universally when parsing mathematical expressions.We have implemented the financial plans using Polish expressions and logical formulation of C code to achieve the output in the stack. And at the end we have stored all the data of the user with the help of file handling in a text file.

TOPIC: **QUEUE**

IMPLEMENTATION: Quiz Game by using Circular Queue

DESCRIPTION:-

A queue can be defined as an ordered list which enables insert operations to be performed at one end called **REAR** and delete operations to be performed at another end called **FRONT**.Queue is referred to be as First In First Out list.

There are four different types of queues in data structures:

- Simple Queue
- Circular Queue
- Priority Queue
- Double-Ended Queue (Deque)

In this project we have used the concept of Circular Queue in which we have used operations of enqueue ,dequeue and traversal. It is a Quiz of user specified subject GK, SCIENCE and SPORTS .User needs to answer 5 questions correctly to win .

A circular Queue is created which keeps count of number of correct questions answered and if answered wrong then 1 point is deducted and accordingly queue is dequeued. After every question there is a scoreboard which is shown which is a pattern generator and shows progress in pattern with each correct answer.

When user answers 5 correct questions then circular queue becomes full and pattern is completed .

Functions in the program are:

- Struct()
- isFull()
- isEmpty()
- Enqueue(key)
- Dequeue()
- Traversal()
- Peek()
- Pattern()
- main()

TOPIC: **LINKED LIST**

IMPLEMENTATION: SINGING COMETITION AND LIVE VOTING

DESCRIPTION:-

A linked list is a linear data structure that includes a series of connected nodes. Here, each node stores the data and the address of the next node.

We give the address of the first node a special name called HEAD. Also, the last node in the linked list can be identified because its next portion points to NULL.

Linked lists can be of multiple types: singly, doubly, and circular linked list. In this article, we will focus on the singly linked list.

Functions in program are:

- Voter_Information()
- Voter_Insert()
- Voting()
- Admin()
- Show()
- Winner()
- Stop()
- Not_Again()
- Exit()
- Search()
- Main()

This program is an online voting portal which asks user to login by giving their Aadhar id, name and DOB. If user enters invalid data then it declares that you have entered wrong data so that only those people can vote who have already created their profile.

After successful login, the list of candidates is shown and user can give vote to his favoured candidate.

At then end when every user gives their votes then that candidate who gets maximum number of votes becomes the winner.

Here linked list is used to change nodes and pointers according to the number of votes given during live voting.

TOPIC: **TREES**

A tree is non-linear and a hierarchical data structure consisting of a collection of nodes such that each node of the tree stores a value and a list of references to other nodes

This data structure is a specialized method to organize and store data in the computer to be used more effectively. It consists of a central node, structural nodes, and sub-nodes, which are connected via edges. We can also say that tree data structure has roots, branches, and leaves connected with one another.

In our Project, we have created a student database management system in which we have used a Binary Search Tree to Store the Names of students and displayed them alphabetically using Inorder Traversal of the tree.

This was done by comparing the ASCII values of the first letters of name of all the students and storing them in the tree accordingly.

‘createNode’ function will create a node that will store the student’s data.

‘insert’ function will check where to add the new student’s data in the BST.

‘findNode’ function will help to find the student in existing tree.

‘deleteNode’ function will delete the student’s data from the tree.

‘inorderTraversal’ function will traverse the data of students alphabetically.

‘fileRead’ function is used to read the contents of the file.

‘fileSave’ function is used to write the given student’s data in the file.

Lastly all the required output or student description in ascending order are at stored in file which will be generated via file handling.

TOPIC: **GRAPH**

A Graph is a non-linear data structure consisting of vertices and edges. The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph. More formally a Graph is composed of a set of vertices (V) and a set of edges (E). The graph is denoted by G (E, V).

In this project, we have created an application to calculate the shortest path for courier service. We have created hotels as vertices and roads as edges that connect the hotels. For the purpose of finding the shortest path, we have used Bellman Ford Algorithm in graph theory.

We have created a 'display' function which will print the adjacency matrix of the generated graph where 'printGraph' function is used to show the edges and vertices of the generated graph.

'minPath' is a function which will return the minimum path between given two source and destination indices.

'printSolution' will print the shortest path between the given indices of matrix.

'writeToFile' function will append the shortest possible path values to the file which will generate a text file.