#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct consumerData{

    char name[20];

    char contact[11];

    char password[20];

    struct consumerData \*loc;

};

struct consumerData \*head = NULL;

void home() {

    int choice;

    printf("\nFollowing are the stations:");

    printf("\n1. Huda City Centre");

    printf("\n2. Patel Chowk");

    printf("\n3. INA");

    printf("\n4. Hauz Khas");

    printf("\n5. Mandi House");

    printf("\n6. Rajiv Chowk");

    printf("\n7. Kashmere Gate");

    printf("\n8. Chandni Chowk");

    printf("\n9. New Delhi");

    printf("\n10. Rajendra Place");

    printf("\n11. Lajpat Nagar");

    char s[20];

    char d[20];

    while (1) {

        printf("\n");

        printf("\n1. Starting and Destination Point");

        printf("\n2. Find Shortest Route");

        printf("\n3. Exit");

        printf("\nEnter your choice: ");

        scanf("%d", &choice);

    switch (choice) {

            case 1:

                printf("\nEnter Starting Station: ");

                scanf("%s",&s);

                printf("\nEnter Destination Station: ");

                scanf("%s",&d);

                break;

            case 2:

                // function

                printf("\nIn Progress");

                break;

            case 3:

                printf("\nThanks for Visiting. Goodbye!\n");

                return;

            default:

                printf("Invalid choice. Please try again.\n");

        }

    }

}

void del(char delConsumer[]){

    struct consumerData \*temp = head;

    struct consumerData \*prev = NULL;

    while(temp!=NULL && strcmp(temp->name,delConsumer)!=0){

        prev = temp;

        temp = temp->loc;

    }

    if(temp==NULL){

        printf("\nConsumer not Found\n");

        return;

    }

    if(prev==NULL){

        head = temp->loc;

    }

    else{

        prev->loc = temp->loc;

    }

    free(temp);

    printf("\nConsumer %s is Deleted Successfully !\n",delConsumer);

}

void reg(){

    struct consumerData \*p = (struct consumerData \*)malloc(sizeof(struct consumerData));

    printf("\nEnter Name : ");

    scanf("%s",&p->name);

    printf("Enter Conatct: ");

    scanf("%s",&p->contact);

    printf("Enter Password: ");

    scanf("%s",&p->password);

    if(head == NULL){

        head = p;

        p -> loc = NULL;

    }

    else{

        struct consumerData \*temp = head;

        while(temp->loc != NULL){

            temp = temp->loc;

        }

        temp->loc = p;

        p->loc = NULL;

    }

    printf("\nRegistered Successfully !\n");

}

void supervisor(){

    char sname[20] = "Shivang";

    char spassword[20] = "1234";

    char name[20];

    char password[20];

    printf("\nWelcome\n");

    printf("\nEnter Name: ");

    scanf("%19s",&name);

    printf("Enter Password: ");

    scanf("%19s",&password);

    if(strcmp(name,sname)==0 && strcmp(password,spassword)==0){

        printf("\nLogin Successfully!\n");

        shome();

    }

    else{

        printf("\nInvalid Input!\n");

    }

}

void shome(){

    int choice;

    char delConsumer[20];

    while(1){

    printf("\nEnter 1 to add Consumer\n");

    printf("Enter 2 to delete Consumer\n");

    printf("Enter 3 to Log Out\n");

    printf("Enter Choice: ");

    scanf("%d",&choice);

    switch(choice){

        case 1:

            reg();

            break;

        case 2:

            printf("\nEnter the name of Consumer: ");

            scanf("%s", &delConsumer);

            del(delConsumer);

            break;

        case 3:

            printf("\nLogging Out...\n");

            return;

        default:

            printf("\nInvalid Input. Please try again.\n");

    }

    }

}

void login(){

    char name[20];

    char password[20];

    printf("\nEnter Your Name: ");

    scanf("%s",&name);

    printf("Enter Password: ");

    scanf("%s",&password);

    search(name, password);

}

void search(char name[], char password[]){

    struct consumerData \*temp = head;

    while(temp != NULL){

        if(strcmp(name,temp->name)==0 && strcmp(password,temp->password)==0){

            break;

        }

        else{

            temp = temp->loc;

        }

    }

    if(temp == NULL){

        printf("\nYou Need to Register!\n");

    }

    else{

        printf("\nLogin Successfully!\n");

        home();

    }

}

void saveToFile() {

    FILE \*fp;

    fp = fopen("DelhiMetroOutput.txt", "w");

    if (fp == NULL) {

        printf("Error opening file for writing.\n");

        return;

    }

    struct consumerData \*temp = head;

    while (temp != NULL) {

        fprintf(fp, "%s %s %s\n", temp->name, temp->contact, temp->password);

        temp = temp->loc;

    }

    fclose(fp);

}

void loadFromFile() {

    FILE \*fp;

    fp = fopen("DelhiMetroOutput.txt", "r");

    if (fp == NULL) {

        printf("File not found.\n");

        return;

    }

    while (1) {

        struct consumerData \*p = (struct consumerData \*)malloc(sizeof(struct consumerData));

        if (fscanf(fp, "%s %s %s", p->name, p->contact, p->password) != 3) {

            free(p);

            break;

        }

        if (head == NULL) {

            head = p;

            p->loc = NULL;

        } else {

            struct consumerData \*temp = head;

            while (temp->loc != NULL) {

                temp = temp->loc;

            }

            temp->loc = p;

            p->loc = NULL;

        }

    }

    fclose(fp);

}

int main(){

    loadFromFile();

    printf("\n\tWelcome to Delhi Metro System\n");

    int choice;

    while(1){

        printf("\nEnter 1 to Register\n");

        printf("Enter 2 to Login\n");

        printf("Enter 3 to Login as Supervisor\n");

        printf("Enter 4 to Log Out\n");

        printf("Enter choice: ");

        scanf("%d",&choice);

        switch(choice){

            case 1:

                reg();

                break;

            case 2:

                login();

                break;

            case 3:

                supervisor();

                break;

            case 4:

                printf("\nLogging Out....");

                saveToFile();

                exit(0);

            default:

                printf("Invalid Input");

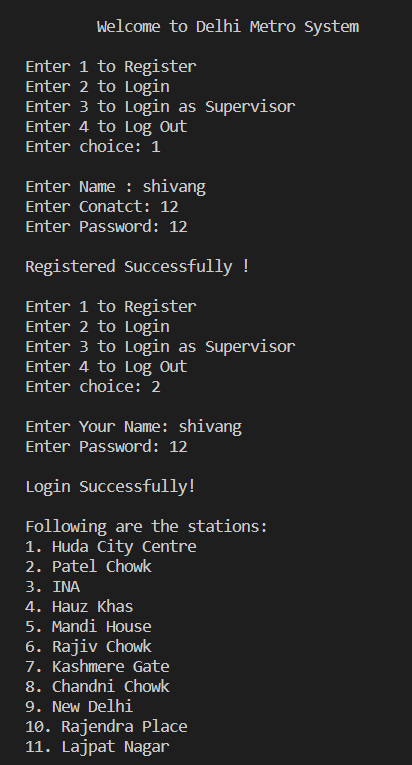
        }

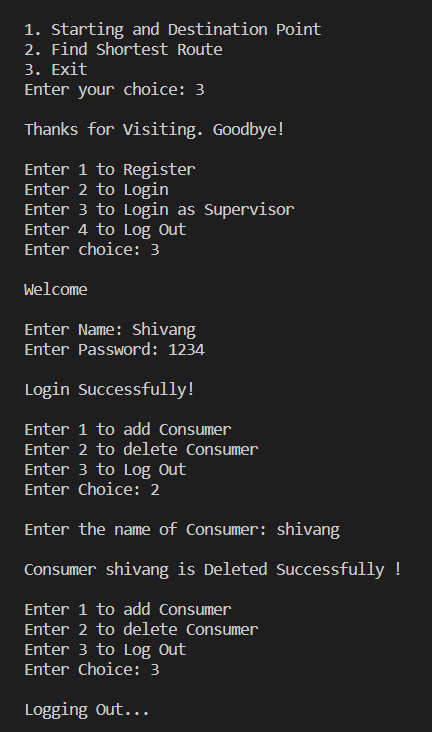
    }

    return 0;

}

Output :





Write Up about Project :  
  
Project on ‘Selection of Shortest Route between two Stations of Delhi Metro’.

Require ‘Graph Theory’.

Here are the features required to make the project:

**1. Data Representation:**

You need to represent the Delhi Metro network as a graph. Each station will be a node, and the connections (lines between stations) will be edges. You can create a data structure to represent this

**2. Data Collection:**

You'll need information about the Delhi Metro network, including station names, line information, and the connections between stations

**3. Algorithm Selection:**

Choose an appropriate graph search algorithm to find the shortest route between two stations

**4. User Interface:**

Create a user interface that allows users to input their source and destination stations.

**5. Path Finding:**

Implement the selected algorithm to find the shortest route between the source and destination stations.

**6. Display Results:**

Display the shortest route to the user

Flow Chart :

Delete Consumer

Add Consumer

Display of Shortest Route

Stations Selection

Display of Stations

User Interface

Delhi Metro

Project