/\*

Group ID= 07

Members:

    1)Arunabh Aditya            Roll Number: 114026   PRN:1032230178

    2)M Aryane                  Roll Number: 114029   PRN:1032232583

    3)Akshay Anurag             Roll Number: 114028   PRN:1032231965

    4)Fardeen Ali               Roll Number: 114027   PRN:1032232038

    5)Ayush Gaikwad             Roll Number: 114039   PRN:1032232473

    6)Shriharsh Deshmukh        Roll Number: 114038   PRN:1032231397

Problem Statement:

write a C program to reserve train tickets with following functionality:

    1)User System:

        i)Login

        ii)signup

        iii)Reset Password

    2)Admin Controls

        i)add new trains

        ii)add delayed trains

        iii)see delayed trains

    3)Tickets reservation:

  i)With option to select preferred class, compartment,seat type (seater or sleeper) and number of seats

ii)With payment system.(with payment failure handling case).

        iii)Once reserved, get PNR.

    4)Get Information of reservations (By entering PNR):

        i)To know in which compartment, class seat is reserved.

        ii)To know if ticket is confirmed or in waiting list.

        iii)Get reservation id or referrence number of the reservation.

        iv)To know all about train (max speed, total dist to be travelled )

        V)Option to print tickets in PDF format

    5)Checking Train Status:

        i)To check delayed trains

    6)Option to cancel tickets:

        i)option to cancel ticket.

        ii)with proper refund messages.

    \*\*\*ADDITIONAL FEATURES\*\*\*

    1)QR code payment format

    2)User profile update

    3)Admin panel

    4)ability to cancel tickets or go back to main menu at any step

    5)Proper error handling.

    6)proper user navigation allowing a smoother program experience.

    7)Proper PDF generation.

8)Proper user interface allowing user to perform all actions without having to restart the program.

    9)Menu Based program.

    \*\*\*\*\*CAUTIONS\*\*\*\*\*

    1)Input should be of the type that is asked.

    2)All input should be in lowercase (unless stated otherwise and as per choice for name and other user details).

Input Required:

    as per the program flow and user wishes.

Algorithms used (modules used):

    1)External library:

        i)qrcodegen.h

        ii)pdfgen.h

    2)Internal header files:

        i)stdio.h

        ii)stdlib.h

        iii)conio.h

        iv)string.h

        v)time.h

        vi)windows.h

        vii)ctype.h

Conclusion:

    Thus implemented a complete error safe train ticket booking application with TUI (terminal user interface) allowing a menu based program.

\*/

//Header Files and #define

#include <stdio.h>

#include <stdlib.h>

#include <conio.h>

#include <string.h>

#include <time.h>

#include <windows.h>

#include "header/qrcodegen.h"

#include <ctype.h>

#include "header/pdfgen.h"

#define MAX\_PASSENGERS 100

//structure definations total 6 structures:

typedef struct {

    char userId[10];

    char email[100];

    char password[100];

    char name[100];

    int age;

    char gender[10];

    long long phone;

} User;

typedef struct {

    char trainId[10];

    char startingPoint[100];

    char destination[100];

    char departureTime[100];

    int cost;

    int compartments;

    char seatType[100];

    char name[100];

    int maxSpeed;

    int totalDist;

} Train;

typedef struct {

    long long pnr;

    char userId[10];

    char name[100];

    char trainId[10];

    int compartment;

    char seatType[10];

    char status[10];

    float cost;

    int seats;

    int noOfSeats;

    char date[20];

    int age;

    char gender[10];

    long long phone;

} Reservation;

typedef struct {

    char trainId[20];

    char startingPoint[20];

    char destination[20];

    char departureTime[15];

    int cost;

    int compartments;

    char seatType[20];

    int isDelayed;

    int delayTime;

} DelayedTrain;

typedef struct PassengerTicket PassengerTicket;

typedef struct {

    char StartingPoint[20];

    char Destination[20];

    char PNR[20];

    char TrainName[20];

    char TrainId[20];

    char class[20];

    char Date[20];

    char dist[20];

    char Cost[20];

    char Compartment[20];

    int totalSeats;

    PassengerTicket\* passengerTickets[10];

} Ticket;

struct PassengerTicket {

    char passName[20];

    char age[20];

    char gender[20];

    char seatNum[20];

};

//Global Veriables:

int delayMilliseconds = 5000;

int logged = 0;

int \*loggedPtr = &logged;

char userId[10];

char \*userIdPtr = userId;

//All function declarations: total 48 functions

void sleepProgram(float seconds);

void PrintSleep(float seconds);

void clearTerminal();

void greenColor();

void redColor();

void resetColor();

void yellowColor();

void signup();

int userExists(const char\* email);

void login();

void copyFile(const char \*source, const char \*destination);

void resetPass();

void LogOrSign();

void adminLogin();

void addTrain();

void displayAllTrains();

int compareByCostAsc(const void\* a, const void\* b);

int compareByCostDesc(const void\* a, const void\* b);

void sortByCost(int sortOrder);

void findTrainsByDestination(const char\* destination);

void findTrainsByStartingPoint(const char\* startingPoint);

void adminControls();

void addDelayedTrain();

void displayDelayedTrains(int isAdmin);

void trainListandBook();

void findTrain(const char\* trainId);

void showCompartment( int compartment, const char\* trainId);

void randomlyBookSeats(int\* seats, int numSeats);

int isValidDate(const char \*dateStr);

void payNow(const char\* trainId, int ticketsToBuy, int choosenCompartment, int ticketsNums[MAX\_PASSENGERS], const char\* date);

void showTickets(int choosenCompartment, const char\* trainId);

int isAllDigits(const char \*str);

void paymentGateway(const char\* trainId, int ticketsToBuy, int choosenCompartment, float totalPrice, int ticketsNums[MAX\_PASSENGERS], const char\* date);

long long generate10DigitRandomNumber();

long long generate15DigitRandomNumber();

void writeReservationToFile(const Reservation\* passenger);

void confirmTickets(const char\* trainId, int ticketsToBuy, int choosenCompartment, int\* ticketsNums, float totalPrice, const char\* date);

void PrintToBeDeletedReservation(long long pnr);

void deleteReservationByPnr(long long pnr);

void cancelBooking();

void seeReservationInfo(long long pnrInfo);

void reservationInfo();

void getTicketFormat();

void getTicketFormatAfterBooking(long long pnrInfo);

int writeTicketPDF(const Ticket\* ticket);

void checkAllReservations();

void showUserInfo();

void changeUserInfo();

void mainMenu();

//PROGRAM STRUCTURE:

1)Login Signup processes

void LogOrSign()

void login()

void signup()

void showUserInfo();

2)Admin Controls

void adminControls()

void adminLogin()

void addTrain();

void addDelayedTrain()

3) display trains and booking process

void displayAllTrains();

void findTrain(const char\* trainId);

void showCompartment( int compartment, const char\* trainId);

void showTickets(int choosenCompartment, const char\* trainId);

4)Payment Process

void payNow(const char\* trainId, int ticketsToBuy, int choosenCompartment, int ticketsNums[MAX\_PASSENGERS], const char\* date);

void paymentGateway(const char\* trainId, int ticketsToBuy, int choosenCompartment, float totalPrice, int ticketsNums[MAX\_PASSENGERS], const char\* date);

void confirmTickets(const char\* trainId, int ticketsToBuy, int choosenCompartment, int\* ticketsNums, float totalPrice, const char\* date);

void writeReservationToFile(const Reservation\* passenger);

5)Reservation Information and cancellation

void reservationInfo();

void deleteReservationByPnr(long long pnr);

void cancelBooking();

void getTicketFormat();

int writeTicketPDF(const Ticket\* ticket);