**ASSIGNMENT2:**

**1.**

#include <stdio.h>

#include <string.h>

#define MAX\_SEATS 10

struct Passenger {

char name[50];

int age;

char contact[15];

int seatNumber;

} passengers[MAX\_SEATS];

int totalSeats, bookedSeats = 0;

void addBusDetails()

{

printf("Enter Total Seats: ");

scanf("%d", &totalSeats);

}

void bookSeat()

{

if (bookedSeats >= totalSeats)

{

printf("No seats available!\n");

return 0;

}

printf("Enter Name: ");

scanf(" %[^\n]", passengers[bookedSeats].name);

printf("Enter Age: ");

scanf("%d", &passengers[bookedSeats].age);

printf("Enter Contact: ");

scanf("%s", passengers[bookedSeats].contact);

passengers[bookedSeats].seatNumber = bookedSeats + 1;

printf("Seat %d booked!\n", passengers[bookedSeats].seatNumber);

bookedSeats++;

}

void cancelSeat()

{

int seatNum, i, found = 0;

printf("Enter Seat Number to Cancel: ");

scanf("%d", &seatNum);

for (i = 0; i < bookedSeats; i++) {

if (passengers[i].seatNumber == seatNum) {

printf("Seat %d canceled!\n", seatNum);

for (i=bookedseats;i < bookedSeats - 1; i++) passengers[i] = passengers[i + 1]

found = 1;

break;

}

}

if (!found)

{

printf("Seat not found!\n");

}

void displaySeats() {

if (bookedSeats == 0)

{

printf("No bookings yet.\n");

return;

}

printf("\nSeat | Name | Age | Contact\n");

for (int i = 0; i < bookedSeats; i++)

printf("%d | %s | %d | %s\n", passengers[i].seatNumber, passengers[i].name, passengers[i].age, passengers[i].contact);

}

int main()

{

int choice;

addBusDetails();

while (1) {

printf("\n1. Book\n2. Cancel\n3. Display\n4. Exit\nChoice: ");

scanf("%d", &choice);

if (choice == 1) bookSeat();

else if (choice == 2) cancelSeat();

else if (choice == 3) displaySeats();

else if (choice == 4) break;

else printf("Invalid choice!\n");

}

return 0;

}

**4.**

#include <stdio.h>

#define MAX\_BUSES 5

struct Bus {

int id;

float latitude, longitude;

char timestamp[20];

int distanceToStop;

};

struct Bus buses[MAX\_BUSES];

int totalBuses=0;

void updateBusLocation() {

if (totalBuses >= MAX\_BUSES)

{

printf("Bus list full!\n");

return;

}

printf("Enter Bus ID: ");

scanf("%d", &id);

printf("Enter Latitude: ");

scanf("%f", &latitude);

printf("Enter Longitude: ");

scanf("%f", &longitude);

printf("Enter Timestamp (HH:MM): ");

scanf("%s", &timestamp);

printf("Enter Distance to Stop (km): ");

scanf("%d", &distanceToStop);

totalBuses++;

printf("Bus location updated\n");

void displayBusInfo()

{

if (totalBuses == 0)

{

printf("No buses available.\n");

return;

}

printf("\nBus ID | Location (Lat, Long) | Time | Distance | ETA (mins)\n");

for (int i = 0; i < totalBuses; i++)

{

int eta = buses[i].distanceToStop \* 2;

printf("%d | (%f, %f) | %s | %d | %d mins\n")

buses[i].id, buses[i].latitude, buses[i].longitude, buses[i].timestamp,

buses[i].distanceToStop, eta);

}

}

int main()

{

int choice;

while (1) {

printf("\n1. Update Bus Location\n2. Show Bus Info\n3. Exit\nChoose: ");

scanf("%d", &choice);

if (choice == 1) updateBusLocation();

else if (choice == 2) displayBusInfo();

else if (choice == 3) break;

else printf("Invalid choice!\n");

}

return 0;

}

**3.**

#include <stdio.h>

#include <string.h>

#define MAX\_BOOKINGS 10

struct Ticket {

char name[50], contact[15], date[15], busType[10], paymentMethod[20], transactionID[20];

int age, ticketID;

float price, amountPaid;

};

void bookTicket()

{

if (totalBookings >= MAX\_BOOKINGS)

{

printf("No more tickets available!\n");

return;

}

else

printf("Enter Passenger Name: ");

scanf(" %[^\n]", t->name);

printf("Enter Age: ");

scanf("%d", &t->age);

printf("Enter Contact Number: ");

scanf("%s", t->contact);

printf("Enter Travel Date (DD/MM/YYYY): ");

scanf("%s", t->date);

printf("Enter Bus Type (AC/Non-AC/Sleeper): ");

scanf("%s", t->busType);

printf("Ticket Price: ₹%.2f\n", t->price);

printf("Enter Payment Method (UPI/Card/Cash): ");

scanf("%s", t->paymentMethod);

printf("Enter Transaction ID: ");

scanf("%s", t->transactionID);

t->amountPaid = t->price;

totalBookings++;

printf("\nTicket Booked Successfully! Ticket ID: %d\n", t->ticketID);

}

void displayBookings()

{

if (totalBookings == 0)

{

printf("No tickets booked yet.\n");

return;

}

printf("\nBooked Tickets:\n");

for (int i = 0; i < totalBookings; i++)

{

struct Ticket \*t = &tickets[i];

printf("%d | %s | %d | %s | %s | ₹%f | %s | %s\n",)

}

void cancelTicket() {

if (totalBookings == 0) {

printf("No tickets to cancel.\n");

return;

}

int id, found = 0;

printf("Enter Ticket ID to Cancel: ");

scanf("%d", &id);

for (int i = 0; i < totalBookings; i++) {

if (tickets[i].ticketID == id) {

printf("Ticket %d canceled. Refund of ₹%.2f processed.\n", id, tickets[i].price);

for (int j = i; j < totalBookings - 1; j++) tickets[j] = tickets[j + 1];

totalBookings--;

break;

}

}

if (!found) printf("Ticket ID not found!\n");

}

int main()

{

int choice;

while (1) {

printf("\n1. Book Ticket\n2. View Bookings\n3. Cancel Ticket\n4. Exit\nChoose: ");

scanf("%d", &choice);

if (choice == 1) bookTicket();

else if (choice == 2) displayBookings();

else if (choice == 3) cancelTicket();

else if (choice == 4) break;

else printf("Invalid choice! Try again.\n");

}

return 0;

}