1.

#include <stdio.h>

#define MAX\_SEATS 5

typedef struct {

char name[50];

int age;

char contact[15];

int booked;

} Passenger;

Passenger seats[MAX\_SEATS];

void bookSeat() {

int seat;

printf("Enter seat number (1-%d): ", MAX\_SEATS);

scanf("%d", &seat);

if (seat < 1 || seat > MAX\_SEATS || seats[seat - 1].booked) {

printf("Invalid seat or seat already booked!\n");

return;

}

printf("Enter name: ");

scanf("%s", seats[seat - 1].name);

printf("Enter age: ");

scanf("%d", &seats[seat - 1].age);

printf("Enter contact number: ");

scanf("%s", seats[seat - 1].contact);

seats[seat - 1].booked = 1;

printf("Seat booked successfully!\n");

}

void cancelSeat() {

int seat;

printf("Enter seat number to cancel (1-%d): ", MAX\_SEATS);

scanf("%d", &seat);

if (seat < 1 || seat > MAX\_SEATS || !seats[seat - 1].booked) {

printf("Invalid seat or seat not booked!\n");

return;

}

seats[seat - 1].booked = 0;

printf("Seat %d cancelled successfully!\n", seat);

}

void displaySeats() {

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

int booked\_count = 0;

for (int i = 0; i < MAX\_SEATS; i++) {

if (seats[i].booked) {

printf("Seat %d: %s, Age: %d, Contact: %s\n", i + 1, seats[i].name, seats[i].age, seats[i].contact);

booked\_count++;

}

}

if (booked\_count == 0) {

printf("No seats are booked.\n");

}

}

int main() {

int choice;

for (int i = 0; i < MAX\_SEATS; i++) {

seats[i].booked = 0;

}

do {

printf("\nBus Seat Booking System\n");

printf("1. Book a seat\n");

printf("2. Cancel a seat\n");

printf("3. Display booked seats\n");

printf("4. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

if (choice == 1) {

bookSeat();

} else if (choice == 2) {

cancelSeat();

} else if (choice == 3) {

displaySeats();

} else if (choice == 4) {

printf("Exiting the system.\n");

} else {

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

}

} while (choice != 4);

return 0;

}

3.TICKET BOOKING AND PAYMENT SYSTEM

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct {

char name[20];

int age;

char contactno[20];

char date[20];

} Passenger;

typedef struct {

char paymentmethod[20];

float amount;

char transactionID[20];

} Payment;

typedef struct {

Passenger passenger;

float ticketPrice;

Payment payment;

int cancel;

} Ticket;

void bookticket(Ticket \*ticket);

void calculateticketPrice(Ticket \*ticket);

void processpayment(Ticket \*ticket);

void displayconfirmation(Ticket \*ticket);

void cancelbooking(Ticket \*ticket);

void displayrefund(Ticket \*ticket);

int main() {

Ticket ticket;

int ch;

while (1) {

printf("1. Book a Ticket\n");

printf("2. Cancel a Ticket\n");

printf("Enter your choice: ");

scanf("%d", &ch);

if (ch == 1) {

bookticket(&ticket);

} else if (ch == 2) {

cancelbooking(&ticket);

} else {

printf("Invalid choice\n");

}

}

return 0;

}

void bookticket(Ticket \*ticket) {

printf("\nEnter Passenger Details\n");

printf("Name: ");

scanf(" %s", ticket->passenger.name);

printf("Age: ");

scanf("%d", &ticket->passenger.age);

printf("Contact Number: ");

scanf("%s", ticket->passenger.contactno);

printf("Travel Date: ");

scanf("%s", ticket->passenger.date);

calculateticketPrice(ticket);

processpayment(ticket);

displayconfirmation(ticket);

}

void calculateticketPrice(Ticket \*ticket) {

char bustype[20];

printf("\nEnter Bus Type (Regular/Luxury): ");

scanf("%s", bustype);

if (strcmp(bustype, "Regular") == 0) {

ticket->ticketPrice = 50.0;

} else if (strcmp(bustype, "Luxury") == 0) {

ticket->ticketPrice = 100.0;

} else {

printf("Invalid bus type\n");

ticket->ticketPrice = 50.0;

}

if (strstr(ticket->passenger.date, "12/25") != NULL) {

ticket->ticketPrice \*= 1.2;

}

printf("Ticket Price: %.2f\n", ticket->ticketPrice);

}

void processpayment(Ticket \*ticket) {

printf("Payment Method: ");

scanf("%s", ticket->payment.paymentmethod);

printf("Amount to Pay: %.2f\n", ticket->ticketPrice);

printf("Enter Transaction ID: ");

scanf("%s", ticket->payment.transactionID);

printf("Payment of $%.2f received via %s Transaction ID: %s\n", ticket->ticketPrice, ticket->payment.paymentmethod, ticket->payment.transactionID);

}

void displayconfirmation(Ticket \*ticket) {

printf("Passenger Name: %s\n", ticket->passenger.name);

printf("Age: %d\n", ticket->passenger.age);

printf("Contact Number: %s\n", ticket->passenger.contactno);

printf("Travel Date: %s\n", ticket->passenger.date);

printf("Ticket Price: $%.2f\n", ticket->ticketPrice);

printf("Payment Method: %s\n", ticket->payment.paymentmethod);

printf("Transaction ID: %s\n", ticket->payment.transactionID);

printf("Booking done.\n");

}

void cancelbooking(Ticket \*ticket) {

char confirmation;

printf("\nAre you sure you want to cancel the ticket:(Y/N): ");

getchar();

scanf("%c", &confirmation);

if (confirmation == 'Y' || confirmation == 'y') {

ticket->cancel = 1;

displayrefund(ticket);

} else {

printf("Cancellation not possible.\n");

}

}

void displayrefund(Ticket \*ticket) {

if (ticket->cancel) {

float refundAmount = ticket->ticketPrice \* 0.8; // 20% cancellation fee

printf("Your booking has been cancelled.\n");

printf("Refund Amount: %.2f\n", refundAmount);

} else {

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

}

}

GitHub -- https://github.com/Gagana-13?tab=repositories