#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_ROWS 5

#define MAX\_SEATS\_PER\_ROW 10

#define MAX\_MOVIES 6

#define MAX\_LOCATIONS 5

#define MAX\_SNACKS 4

#define MAX\_RESERVATIONS 10

#define TICKET\_PRICE 200

int seats[MAX\_ROWS][MAX\_SEATS\_PER\_ROW] = {0};

const char\* movies[MAX\_MOVIES] =

{

"LEO",

"Jailer",

"Irugapatru",

"Jigarthanda Doublex",

"Adiye",

"Mark Antony"

};

const char\* locations[MAX\_LOCATIONS] =

{

"Trichy-LA",

"Chennai-IMAX",

"Madurai-INOX",

"Coimbatore-IMAX",

"salem-PVR"

};

const char\* snacks[MAX\_SNACKS] = {

"Popcorn",

"Soda",

"Candy",

"None"

};

struct Reservation {

char movie[50];

char location[50];

char snack[50];

int row;

int seat;

int numTickets;

int totalCost;

};

struct Reservation reservations[MAX\_RESERVATIONS];

int reservationCount = 0;

void initializeSeats()

{

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

{

for (int j = 0; j < MAX\_SEATS\_PER\_ROW; j++)

{

seats[i][j] = 0;

}

}

}

void displayMovies()

{

printf("Available Movies:\n");

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

{

printf("%d. %s\n", i + 1, movies[i]);

}

}

void displayLocations()

{

printf("Available Theater Locations:\n");

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

{

printf("%d. %s\n", i + 1, locations[i]);

}

}

void displaySnacks()

{

printf("Available Snacks:\n");

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

{

printf("%d. %s\n", i + 1, snacks[i]);

}

}

void displayAvailableSeats(int movieIndex)

{

printf("Available Seats for %s:\n", movies[movieIndex]);

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

{

printf("Row %d: ", i + 1);

for (int j = 0; j < MAX\_SEATS\_PER\_ROW; j++)

{

if (seats[i][j] == 0)

{

printf("%d ", j + 1);

}

}

printf("\n");

}

}

void displayReservations()

{

if (reservationCount == 0)

{

printf("You have no reservations.\n");

}

else

{

printf("Your Reservations:\n");

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

{

printf("\n");

printf("\*\*\*\*\*\*");

printf("\*\*\*\*\*\*");

printf("\n");

printf("Reservation %d: %s - %s - Row %d, Seat %d - Snack: %s - Tickets: %d - Total Cost: Rs. %d\n",

i + 1, reservations[i].movie,

reservations[i].location,

reservations[i].row,

reservations[i].seat,

reservations[i].snack,

reservations[i].numTickets,

reservations[i].totalCost);

printf("\n");

printf("\*\*\*\*\*\*");

printf("\*\*\*\*\*\*");

printf("\n");

}

}

}

int processPayment()

{

char cardNumber[20];

char cvv[10];

printf("Enter your credit card number: ");

scanf("%s", cardNumber);

printf("Enter your CVV: ");

scanf("%s", cvv);

if (strlen(cardNumber) == 16 && strlen(cvv) == 3) {

printf("\n");

printf("\*\*\*\*");

printf("\*\*\*\*");

printf("\n");

printf("Payment successful! Your seats are reserved.\n");

printf("\n");

printf("Total Cost: Rs. %d\n", reservations[reservationCount - 1].totalCost);

printf("\n");

printf("\*\*\*\*");

printf("\*\*\*\*");

return 1;

}

else {

printf("Payment failed. Please check your card information and try again.\n");

return 0;

}

}

int main()

{

initializeSeats();

while (1) {

printf("\n1. Display Available Movies\n2. Select a Movie\n3. Reserve Seats\n4. View Reservations\n5. Select Theater Location\n6. Select Snacks\n7. Make Payment\n8. Exit\nEnter your choice: ");

int choice;

scanf("%d", &choice);

switch (choice) {

case 1:

displayMovies();

break;

case 2: {

int selectedMovie;

displayMovies();

printf("Select a movie by entering its number: ");

scanf("%d", &selectedMovie);

if (selectedMovie < 1 || selectedMovie > MAX\_MOVIES) {

printf("Invalid movie selection. Please choose a valid number.\n");

} else {

displayAvailableSeats(selectedMovie - 1);

}

break;

}

case 3: {

int selectedMovie;

displayMovies();

printf("Select a movie by entering its number: ");

scanf("%d", &selectedMovie);

if (selectedMovie < 1 || selectedMovie > MAX\_MOVIES) {

printf("Invalid movie selection. Please choose a valid number.\n");

} else {

displayAvailableSeats(selectedMovie - 1);

int numTickets;

printf("Enter the number of tickets to be booked: ");

scanf("%d", &numTickets);

if (numTickets < 1) {

printf("Invalid number of tickets. Please book at least one ticket.\n");

break;

}

char selectedLocation[50];

displayLocations();

printf("Select a theater location by entering its number: ");

int locationChoice;

scanf("%d", &locationChoice);

if (locationChoice < 1 || locationChoice > MAX\_LOCATIONS)

{

printf("Invalid location selection. Please choose a valid number.\n");

break;

} else

{

strcpy(selectedLocation, locations[locationChoice - 1]);

}

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

printf("Enter the row and seat number you want to reserve (e.g., 1 3): ");

int row, seat;

scanf("%d %d", &row, &seat);

if (row < 1 || row > MAX\_ROWS || seat < 1 || seat > MAX\_SEATS\_PER\_ROW)

{

printf("Invalid row or seat number. Please choose valid values.\n");

break;

} else if (seats[row - 1][seat - 1] == 1)

{

printf("Seat %d in Row %d is already reserved. Please choose another seat.\n", seat, row);

break;

} else

{

seats[row - 1][seat - 1] = 1;

char selectedSnack[50];

strcpy(reservations[reservationCount].movie, movies[selectedMovie - 1]);

strcpy(reservations[reservationCount].location, selectedLocation);

displaySnacks();

printf("Select a snack by entering its number: ");

int snackChoice;

scanf("%d", &snackChoice);

if (snackChoice < 1 || snackChoice > MAX\_SNACKS) {

printf("Invalid snack selection. Please choose a valid number.\n");

seats[row - 1][seat - 1] = 0;

break;

} else {

strcpy(selectedSnack, snacks[snackChoice - 1]);

}

reservations[reservationCount].numTickets = numTickets;

reservations[reservationCount].totalCost = numTickets \* TICKET\_PRICE;

strcpy(reservations[reservationCount].snack, selectedSnack);

reservations[reservationCount].row = row;

reservations[reservationCount].seat = seat;

reservationCount++;

if (i == 0) {

printf("Reserved %d seat(s) in Row %d for %s at %s with %s successfully!\n",

numTickets, row, movies[selectedMovie - 1], selectedLocation, selectedSnack);

} else {

printf("Reserved %d seat(s) in Row %d for %s with %s successfully!\n",

numTickets, row, movies[selectedMovie - 1], selectedSnack);

}

}

}

}

break;

}

case 4:

displayReservations();

break;

case 5: {

displayLocations();

break;

}

case 6: {

displaySnacks();

break;

}

case 7: {

if (reservationCount == 0) {

printf("You don't have any reservations. Please reserve seats first.\n");

} else {

int paymentSuccess = processPayment();

if (paymentSuccess) {

reservationCount = 0;

initializeSeats();

}

}

break;

}

case 8:

printf("Thank you. Exiting the program.\n");

return 0;

default:

printf("Invalid choice. Please enter a valid option.\n");

}

}

return 0;

}