Green University of Bangladesh

Department of Computer Science and Engineering (CSE)

Faculty of Science and Engineering

Semester: (Spring, Year: 2023), B.Sc. in CSE (Eve)

Lab Project Report

Course Title: Data Structure lab

Course Code: CSE-106 **Section: PC-221-EM**

Project Proposal Name: Movie Ticket Booking System

Student Details

Name		ID
1	Tanvir Khan Shykat	221915004

Date Course : 18-05-2023

Submission Date : 15-06-2023

Course Teachers Name : Md. Abu Rumman Refat (Lecturer , CSE)

[For Teachers use only: Don’t Write Anything inside this box]

Project Proposal Status

Marks: ………………………. Comments: ……………………

Signature: …………………….. Date: …………………………..

LAB PROJECT REPORT

ON

Table of Contents

Chapter : 01

Introduction

1.1 Introduction:

The Movie Ticket Booking System is a software application designed to facilitate the process of booking movie tickets. It provides an efficient and user-friendly platform for moviegoers to browse available movies, choose seats, and make ticket reservations. The system aims to streamline the ticket booking process, eliminate long queues, and enhance the overall movie-watching experience for customers.

1.2 Design Goals/ Objective:

Movie Listing: The system provides a comprehensive list of movies currently playing in theaters. Users can view details such as movie titles, genres, synopses, ratings, and showtimes.

Seat Selection: Users can select their preferred seats from an interactive seating layout. The system displays seat availability in real-time, allowing customers to choose their desired seats for a specific showtime.

Multiple Payment Options: The booking system supports various payment methods, including credit/debit cards, mobile wallets, and online banking. Users can securely make payments within the application.

User Registration and Authentication: To enhance user experience and offer personalized services, the system allows users to register accounts. Registered users can save their preferences, access booking history, and receive special offers.

Discounts and Promotions: The system incorporates promotional offers, discounts, and loyalty programs to incentivize users. Special discounts may be offered for specific movies, showtimes, or group bookings.

Booking Management: The system includes an administrative panel that enables theater staff to manage movie listings, showtimes, seat availability, and reservations. Staff members can also generate reports, analyze booking trends, and make necessary adjustments to optimize operations.

Ticket Confirmation and Notifications: After successful booking, customers receive a confirmation email or SMS containing their ticket details, including movie title, showtime, seat numbers, and barcode for easy check-in.

Chapter : 02

Design/Development/Implementation of the Project.

2.1 : Implemention of the Project:

```
#include <stdio.h>
#include <stdlib.h>

#define MAX_SEATS 10

struct Movie {
    char title[100];
    int availableSeats;
    float ticketPrice;
};

struct Movie movies[] = {
    {"Avengers: The Godfather", MAX_SEATS, 7.0},
    {"The Lion King", MAX_SEATS, 10.0},
    {"The Matrix", MAX_SEATS, 11.0},
    {"Alien", MAX_SEATS, 13.0}
};

void displayMovies() {
    printf("Available Movies:\n");
    printf("-----------------\n");

    int numMovies = sizeof(movies) / sizeof(movies[0]);

    for (int i = 0; i < numMovies; i++) {
        printf("%d. %s\n", i + 1, movies[i].title);
    }

    printf("-----------------\n");
}
```

```
int bookTicket(int movieIndex, int numTickets) {
    if (movieIndex < 0 || movieIndex >= sizeof(movies) / sizeof(movies[0])) {
        printf("Invalid movie selection.\n");
        return 0;
    }

    struct Movie* selectedMovie = &movies[movieIndex];

    if (numTickets > selectedMovie->availableSeats) {
        printf("Insufficient seats available.\n");
        return 0;
    }

    float totalPrice = numTickets * selectedMovie->ticketPrice;

    printf("Booking Details:\n");
    printf("-----------------\n");
    printf("Movie: %s\n", selectedMovie->title);
    printf("Number of Tickets: %d\n", numTickets);
    printf("Total Price: $%.2f\n", totalPrice);
    printf("-----------------\n");

    selectedMovie->availableSeats -= numTickets;

    return 1;
}

int main() {
    int choice, numTickets;

    while (1) {
        displayMovies();

        printf("Enter movie number (0 to exit): ");
        scanf("%d", &choice);

        if (choice == 0) {
            printf("Thank you. Exiting the program.\n");
            break;
        }

        printf("Enter number of tickets: ");
        scanf("%d", &numTickets);

        if (bookTicket(choice - 1, numTickets)) {
            printf("Ticket(s) booked successfully!\n");
        } else {
            printf("Ticket booking failed. Please try again.\n");
        }

        printf("-----------------\n");
    }

    return 0;
}
```

Chapter : 03
Performance Evaluation

1.1 : Simulation Environments:

algorithmic flow of the movie ticket booking system :

1.Define the necessary header files: stdio.h and stdlib.h.

2. Define a constant MAX_SEATS to represent the maximum number of seats available in a movie.

3. Define a structure Movie with the following fields:
‘Title’: A character array to store the movie title.
‘availableSeats’: An integer to store the number of available seats for the movie.
‘ticketPrice’: A float to store the price of each ticket for the movie.

4.’ Create an array’` movies’’ of type’’ struct ‘’Movie to store’ the available movies. Initialize it with the movie titles, maximum seats, and ticket prices.

5.Implement the main() function as the entry point of the program:
Declare variables choice and numTickets to store the user's input.
Start an infinite loop.
Call the displayMovies() function to show the available movies.
Prompt the user to enter the movie number they want to book tickets for (0 to exit).
If the user enters 0, print a farewell message and break the loop to exit the program.
Prompt the user to enter the number of tickets they want to book.
Call the bookTicket() function with the appropriate arguments based on the user's inputs.

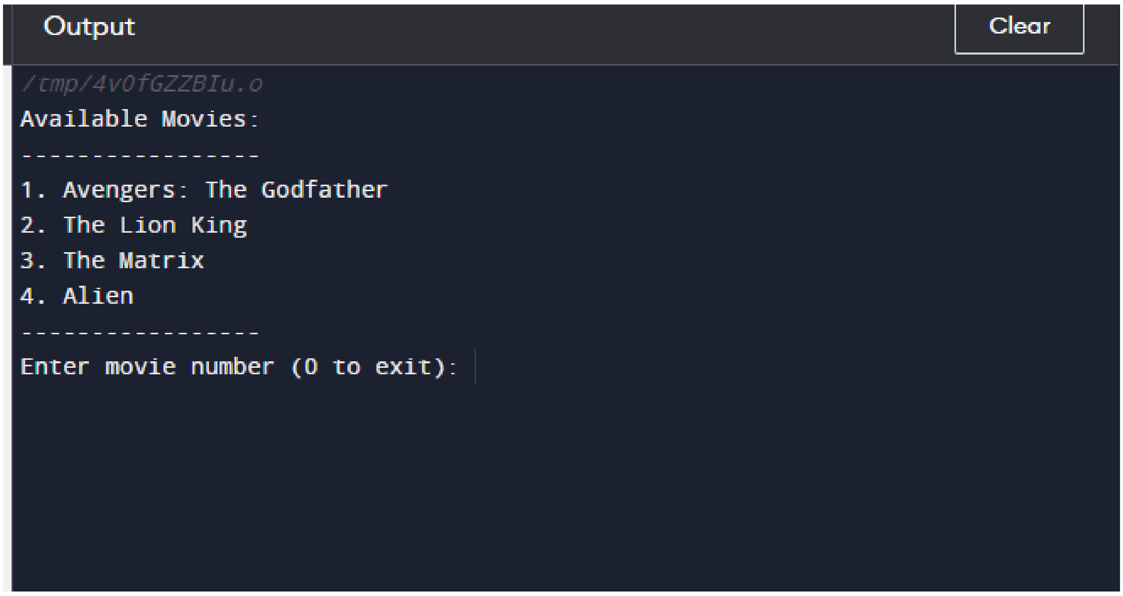
If the booking is successful (the function returns 1), print a success message. Otherwise, print a failure message.

Print a separator line before the next iteration of the loop.

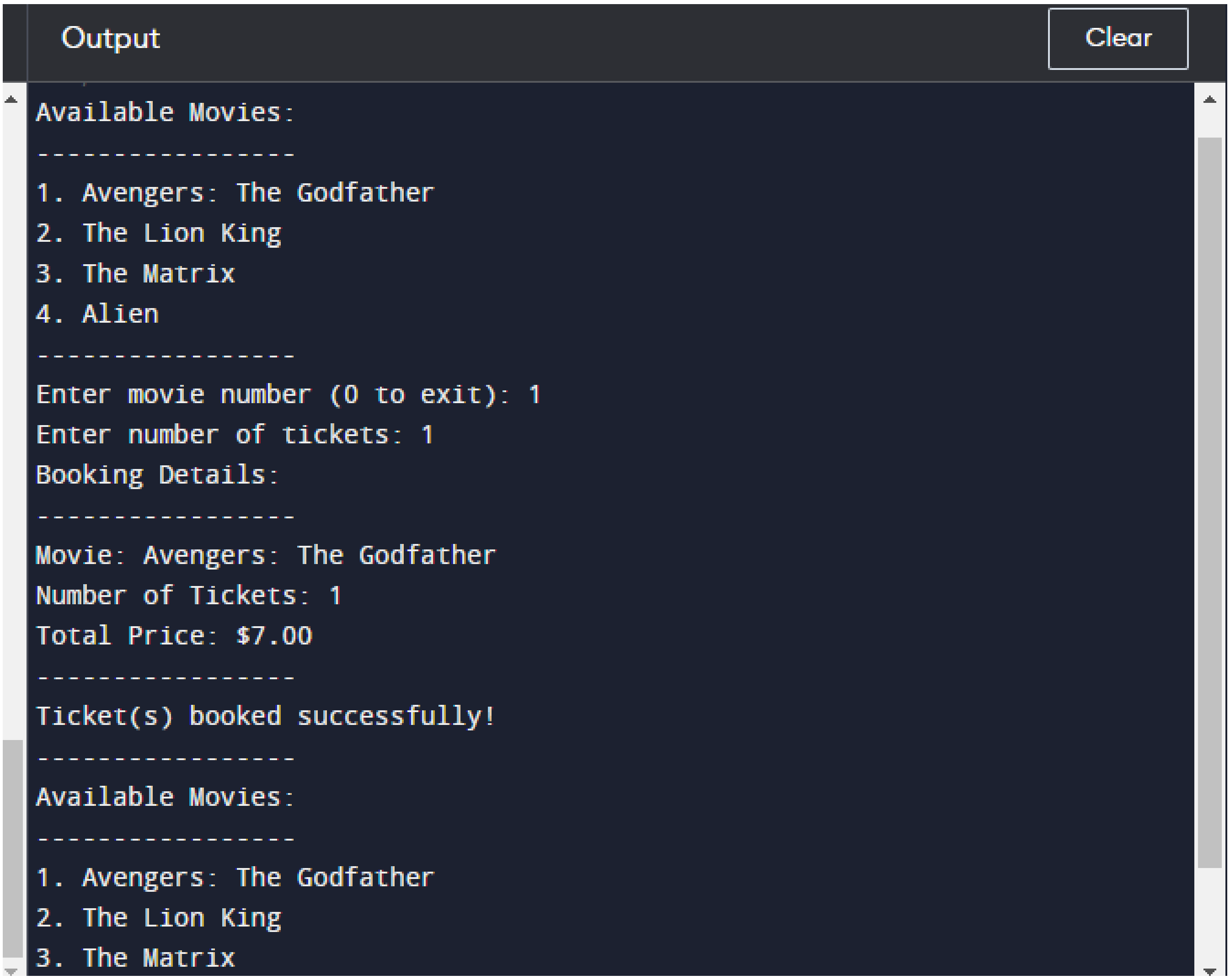
6.Return 0 from the main() function to indicate successful program execution.

1.2 : Result:

1.3 Output:



```
/tmp/4v0fGZZBIu.o
Available Movies:
-----------------
1. Avengers: The Godfather
2. The Lion King
3. The Matrix
4. Alien
-----------------
Enter movie number (0 to exit):
```



```
Available Movies:
-----------------
1. Avengers: The Godfather
2. The Lion King
3. The Matrix
4. Alien
-----------------
Enter movie number (0 to exit): 1
Enter number of tickets: 1
Booking Details:
-----------------
Movie: Avengers: The Godfather
Number of Tickets: 1
Total Price: $7.00
-----------------
Ticket(s) booked successfully!
-----------------
Available Movies:
-----------------
1. Avengers: The Godfather
2. The Lion King
3. The Matrix
```

Chapter : 04

Conclusion

4.1 : Introduction:

The Movie Ticket Booking System project aims to revolutionize the way movie tickets are booked by leveraging technology and providing a seamless, user-friendly platform for movie enthusiasts. It brings efficiency, convenience, and an enhanced movie-going experience for both customers and theater operators.

4.2 : Practical Implication and scope of future work:

1.User Registration and Authentication:

Implement a user registration system to allow users to create accounts.

Develop an authentication mechanism to verify user credentials during login.

2.Seat Selection:

Create a visual representation of the theater seating layout.

Allow users to select seats based on availability.

Implement seat locking to prevent double bookings during the ticket reservation process.

3.Payment Integration:

Integrate with a payment gateway to securely process transactions.

Provide various payment options, such as credit cards, mobile wallets, or online banking.

Handle payment failures and provide appropriate error messages to the user.

4.Booking Confirmation and Ticket Generation:

Generate unique booking reference numbers for each transaction.

Send booking confirmations to users via email or SMS, including details such as movie title, showtime, seat numbers, and total price.

5.User Profile Management:

Allow users to view and update their profile information.

Provide options to manage personal preferences, such as favorite genres or notification settings.

Display the user's booking history and provide the ability to cancel or modify bookings if permitted.

6.Admin Dashboard and Management:

Create an admin interface to manage movie listings, showtimes, and seating arrangements.

Enable administrators to add, edit, or remove movies and their details.

Allow administrators to manage showtimes, adjust seat availability, and generate reports.

7.Security Measures:

Implement secure coding practices to protect against common web vulnerabilities like SQL injection Apply data encryption techniques to safeguard sensitive user information.

8. Automation and Integration: It provides a foundation for gathering system information, which canbe extended to perform additional actions or integrate with other monitoring tools.