PSP [20ES104] COURSE PROJECT REPORT

On

"Online Movie Ticket Booking System"

Developed By:

H.T.NO	STUDENT NAME
2203A51653	T.Sai Kumar
2203A51642	M.Vishwa Teja
2203A51654	T.Rasanya
2203A51659	CH.Akshith Reddy
2203A51646	N. Srikanth

Under the Guidance of

V. Sudhakar
Assistant Professor

Submitted to

Department Computer Science and Artificial Intelligence
SR University
Ananthasagar(V), Hasanparthy(M), Hanamkonda(Dist.) – 506371

www.sru.edu.in
June 2023

Department of Computer Science and Artificial Intelligence

CERTIFICATE

This is to certify that the PSP course project report entitled **“Online Movie Ticket Booking system”** is a record of bonafide work carried out by the student(s) T.Sai Kumar M.Vishwa Teja, T.Rasanya, CH.Akshith Reddy ,N. Srikanth bearing roll number(s) 2203A51653,2203A51642,2203A51654,2203A5159,2203A51646 of Computer Science and Artificial Intelligence department during the academic year 2022-23.

Supervisor

(V. Sudhakar)

INDEX

PROBLEM STATEMENT:

****Develop a C Application for booking movie tickets, which allows storing and managing the details of the reservations made by customers.**

The application should store the following information for each reservation: customer's Name, Age, Aadhar No., Seat Number, Date of Booking, Movie Name, Theater Name, and Ticket Price. The data should be stored in dynamically allocated memory using a structure

MODULES:

Movie Ticket Booking System. It allows users to view available seats in a theater, select seats, buy selected seats, and exit the program. Here's a breakdown of the code:

1. DisplayTheaterView():
   - This function displays the current theater seating arrangement.
   - It uses ANSI escape codes to set colors for available, selected, and bought seats.

2.ISSeatAvailable():

- This function checks if a seat at a given row and column is available.
- It returns 1 if the seat is available, otherwise 0.

3. SelectSeat():
   - This function marks a seat at a given row and column as selected.
4. BuySeat():
   - This function marks a seat at a given row and column as bought.
   - It also displays a success message with the seat details.
5. ClearInputBuffer():
   - This function clears the input buffer to prevent any unwanted input.
6. ProcessPayment():
   - This function simulates the payment processing.
   - It calculates the total price based on the number of selected seats.
   - It asks the user for their bank account number and PIN.
   - It returns 1 to simulate a successful payment.

The main function:

- Initializes the theater seating arrangement with available seats.
- Defines variables and arrays for tracking the selected seats and user input.
- Uses a do-while loop to present a menu of options and handle user input.
- The options include viewing available seats, selecting seats, buying selected seats, and exiting the program.
- Based on the user's choice, the corresponding functions are called to perform the necessary operations.
- The loop continues until the user chooses to exit the program.

To develop the Movie Ticket Booking System application, you would need knowledge in the following areas:

1. C Programming Language: You should be familiar with the C programming language syntax, data types, control structures (such as loops and switch-case), functions, arrays, and structures.
2. Input/Output Handling: Understanding how to read user input from the console using functions like scanf and fgets, and displaying output using functions like printf.
3. Arrays: Working with two-dimensional arrays to represent the theater seating arrangement.
4. Functions: Defining and implementing functions for different tasks, such as displaying the theater view, checking seat availability, selecting seats, buying seats, clearing the input buffer, and processing payment.
5. Conditional Statements: Using conditional statements (if-else) to make decisions based on certain conditions, such as checking seat availability and handling user options.
6. Loops: Utilizing loops (for, while, or do-while) for repetitive tasks, such as iterating over seats or displaying menu options repeatedly until the user chooses to exit.
7. ANSI Escape Codes: Understanding how to use ANSI escape codes to set colors and formatting in the console for displaying the theater view.
8. Basic Algorithmic Knowledge: Implementing basic algorithms for sorting and searching could be required when adding additional functionality, such as sorting or searching passenger data.
9. Basic File Handling (Not present in the provided code): If you want to add functionality to store and retrieve passenger data from files, you would need knowledge of file handling operations like fopen, fread, fwrite, and fclose.
10. Integrated Development Environment (IDE): Familiarity with a C-compatible IDE, such as Visual Studio, Code::Blocks, or Xcode, to write, compile, and debug the code efficiently.
11. Debugging Skills: Knowing how to identify and fix common programming errors, such as syntax errors, logical errors, or memory-related issues.

KNOWLEDGE REQUIRED TO DEVELOP THIS APPLICATION

- Control Statements (if, if-else, switch)
- Loop Statements (while/do while, for)
- Arrays (1D/2D-arrays)
- Strings (Strings and Table of strings) and its functions (strcpy, strcmp)
- Functions (Any type of user defined functions)
- Structure (structures and nested structures)
- Pointers (pointer to strings and pointers to structures)
- Dynamic Memory Allocation (malloc/ calloc/ realloc)

SOURCE CODE [.C FILE]:

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

#define ROWS 10
#define COLS 10
void displayMovies(char movies[3][50])
{
    printf("\n--- Available Movies ---\n");
    for (int i = 0; i < 3; i++)
    {
        printf("%d. %s\n", i + 1, movies[i]);
    }
    printf("------------------------\n");
}

typedef struct
{
    int row;
    int col;
} Seat;

// Function to display the theater view
void displayTheaterView(char theater[ROWS][COLS])
{
    system("clear"); // Clear the console screen
    printf("\n--- Theater View ---\n");

    // Display column numbers
    printf("   ");
    for (int j = 0; j < COLS; j++)
    {
        printf("%-3d", j + 1);
    }
    printf("\n");

    // Display theater seats
    for (int i = 0; i < ROWS; i++)
    {
        printf("%-3d", i + 1); // Display row number
        for (int j = 0; j < COLS; j++)
        {
            if (theater[i][j] == 'A')
                printf("\033[41m   \033[0m"); // Red background for available seats
            else if (theater[i][j] == 'S')
                printf("\033[42m   \033[0m"); // Green background for selected seats
            else if (theater[i][j] == 'B')
                printf("\033[43m   \033[0m"); // Yellow background for bought seats
        }
        printf("\n")
```

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

// Function to check if a seat is available
int isSeatAvailable(char theater[ROWS][COLS], int row, int col)
{
   return theater[row][col] == 'A';
}

// Function to mark a seat as selected
void selectSeat(char theater[ROWS][COLS], int row, int col)
{
   theater[row][col] = 'S';
}

// Function to mark a seat as available
void cancelSeat(char theater[ROWS][COLS], int row, int col)
{
   theater[row][col] = 'A';
}

// Function to buy a seat
void buySeat(char theater[ROWS][COLS], int row, int col)
{
   theater[row][col] = 'B';
   printf("\n\033[33;1mSeat at Row %d, Column %d has been successfully bought.\033[0m\n",
row + 1, col + 1); // Yellow color for success message
}

// Function to clear the input buffer
void clearInputBuffer()
{
   while (getchar() != '\n')
      ;
}

// Function to handle customer login
int customerLogin()
{
   char username[20];
   char password[20];

   printf("\n--- Customer Login ---\n");
   printf("Enter your username: ");
   scanf("%s", username);

   printf("Enter your password: ");
   scanf("%s", password);

   // You can add authentication logic here to validate customer credentials
```

```
  // For simplicity, returning 1 for successful login
    return 1;
}

// Function to handle administrator login
int adminLogin()
{
    char username[20];
    char password[20];

    printf("\n--- Administrator Login ---\n");
    printf("Enter your username: ");
    scanf("%s", username);

    printf("Enter your password: ");
    scanf("%s", password);

    // You can add authentication logic here to validate administrator credentials
    // For simplicity, we are using hardcoded credentials

    if (strcmp(username, "admin") == 0 && strcmp(password, "password") == 0)
    {
        return 1;
    }
    else
    {
        printf("\n\033[31;1mInvalid username or password. Login failed.\033[0m\n"); // Red
color for error message
        return 0;
    }
}

// Function to update the movie ticket rate
void updateTicketRate(int *ticketRate)
{
    printf("\n--- Update Ticket Rate ---\n");
    printf("Current Ticket Rate: %d rupees\n", *ticketRate);
    printf("Enter the new ticket rate: ");
    scanf("%d", ticketRate);
    printf("\nTicket Rate updated successfully.\n");
}

// Function to update the snack items
void updateSnackItems(char snacks[3][20])
{
    printf("\n--- Update Snack Items ---\n");
    printf("Current Snack Items:\n");
    for (int i = 0; i < 3; i++)
    {
        printf("%d. %s\n", i + 1, snacks[i]);
    }
    printf("Enter the new snack items:\n");
    clearInputBuffer();
```

```
 for (int i = 0; i < 3; i++)
    {
        printf("Snack %d: ", i + 1);
        fgets(snacks[i], sizeof(snacks[i]), stdin);
        snacks[i][strcspn(snacks[i], "\n")] = 0; // Remove the newline character
    }
    printf("\nSnack items updated successfully.\n");
}

// Function to add movies available to watch
void addMovies(char movies[3][50])
{
    printf("\n--- Add Movies ---\n");
    clearInputBuffer();
    for (int i = 0; i < 3; i++)
    {
        printf("Movie %d: ", i + 1);
        fgets(movies[i], sizeof(movies[i]), stdin);
        movies[i][strcspn(movies[i], "\n")] = 0; // Remove the newline character
    }
    printf("\nMovies added successfully.\n");
}

int main()
{
    char theater[ROWS][COLS];

    // Initialize the theater view with available seats
    for (int i = 0; i < ROWS; i++)
    {
        for (int j = 0; j < COLS; j++)
        {
            theater[i][j] = 'A';
        }
    }

    int numOfSelectedSeats = 0, numOfSeats;
    int option, row, col;
    int ticketRate = 150; // Default ticket rate

    // Default snack items
    char snacks[3][20] = {
        "Popcorn",
        "Soda",
        "Candy"};

    // Default movies
    char movies[3][50] = {
        "Movie 1",
        "Movie 2",
        "Movie 3"};
 printf("\n--- Welcome to Movie Ticket Booking System ---\n");

    do
    {
        // Display the menu
        printf("\nOptions:\n");
        printf("1. View Available Seats\n");
        printf("2. Select Seat(s)\n");
        printf("3. Cancel Selected Seats\n");
        printf("4. Buy Selected Seats\n");
        printf("5. Administrator Login\n");
        printf("6. Show Available Movies\n");
        printf("7. Exit\n");
        printf("Enter your option: ");
        scanf("%d", &option);

        switch (option)
        {
        case 1:
            // Display the theater view
            displayTheaterView(theater);
            break;
        case 2:
        {
            // Select seat(s)
            int numOfSeats;
            printf("\nEnter the number of seats to select: ");
            scanf("%d", &numOfSeats);
            if (numOfSeats <= 0)
            {
                printf("\n\033[31;1mInvalid number of seats. Please try again.\033[0m\n"); // Red
color for error message
                break;
            }

            printf("\nEnter the row and column of the seats (separated by a space):\n");
            clearInputBuffer();
            for (int i = 0; i < numOfSeats; i++)
            {
                scanf("%d %d", &row, &col);
                row--; // Adjust row number to 0-based index
                col--; // Adjust column number to 0-based index
                if (row >= 0 && row < ROWS && col >= 0 && col < COLS)
                {
                    if (isSeatAvailable(theater, row, col))
                    {
                        selectSeat(theater, row, col);
                        numOfSelectedSeats++;
                        printf("\n\033[32;1mSeat at Row %d, Column %d has been
selected.\033[0m\n", row + 1, col + 1); // Green color for success message
                    }
```

```
  else
                    {
                        printf("\n\033[31;1mSeat at Row %d, Column %d is not available.\033[0m\n",
row + 1, col + 1); // Red color for error message
                    }
                }
                else
                {
                    printf("\n\033[31;1mInvalid seat selection.\033[0m\n"); // Red color for error
message
                }
            }
            break;
        }
        case 3:
        {
            // Cancel selected seats
            if (numOfSelectedSeats > 0)
            {
                printf("\nEnter the row and column of the seats to cancel (separated by a space):\n");
                clearInputBuffer();
                for (int i = 0; i < numOfSelectedSeats; i++)
                {
                    scanf("%d %d", &row, &col);
                    row--; // Adjust row number to 0-based index
                    col--; // Adjust column number to 0-based index
                    if (row >= 0 && row < ROWS && col >= 0 && col < COLS)
                    {
                        if (theater[row][col] == 'S')
                        {
                            cancelSeat(theater, row, col);
                            numOfSelectedSeats--;
                            printf("\n\033[33;1mSeat at Row %d, Column %d has been successfully
canceled.\033[0m\n", row + 1, col + 1); // Yellow color for success message
                        }
                        else
                        {
                            printf("\n\033[31;1mSeat at Row %d, Column %d is not selected.\033[0m\n",
row + 1, col + 1); // Red color for error message
                        }
                    }
                    else
                    {
                        printf("\n\033[31;1mInvalid seat selection.\033[0m\n"); // Red color for error
message
                    }
                }
            }
            else
            {
                printf("\n\033[31;1mNo seats are currently selected.\033[0m\n"); // Red color for
```

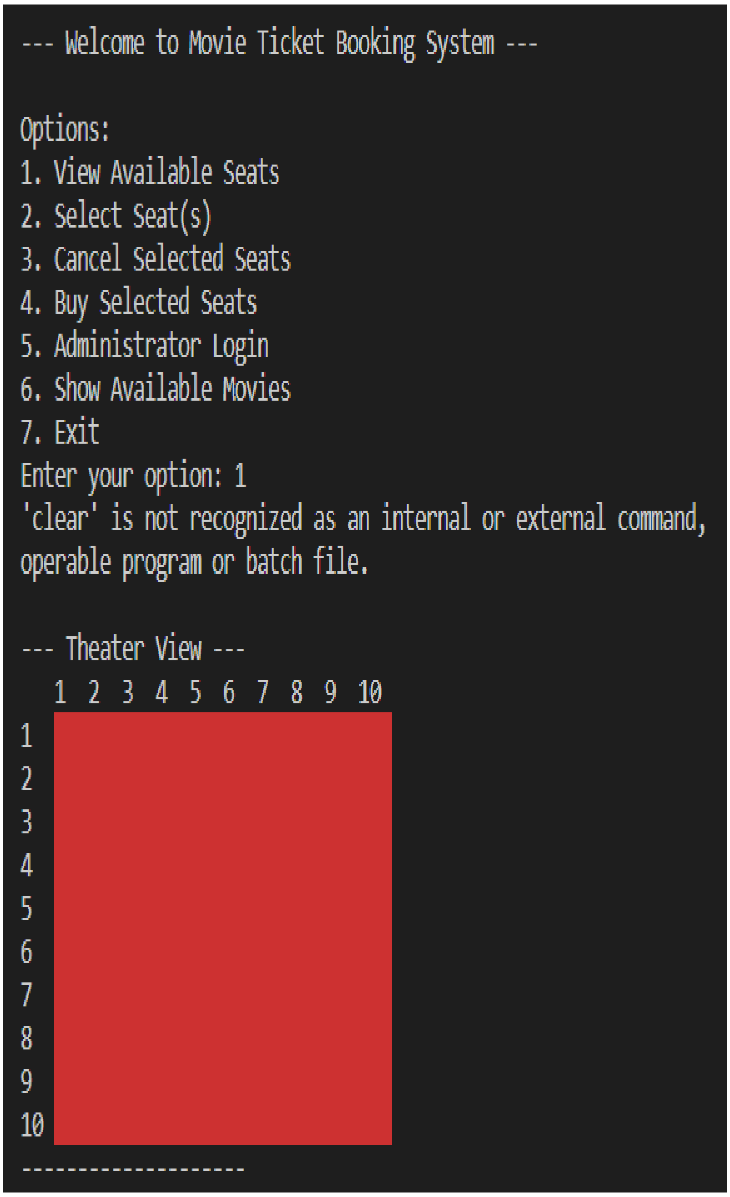
```
error message
            }
            break;
        }
        case 4:
        {
            int seatPrice = 150;
            int totalPrice = seatPrice * numOfSelectedSeats;

            printf("\n--- Bank Account Payment ---\n");
            printf("Total Amount: %d rupees\n", totalPrice);

            char accountNumber[20];
            char pin[5];

            printf("Enter your bank account number: ");
            scanf("%s", accountNumber);

            printf("Enter your PIN: ");
            scanf("%s", pin);

            // Simulating payment processing
            printf("\nProcessing payment...\n");

            if (numOfSelectedSeats > 0)
            {

                 theater[row][col] = 'B';
                printf("\n\033[33;1mSeat at Row %d, Column %d has been successfully
bought.\033[0m\n", row + 1, col + 1);
            }
            else
            {
                printf("payment failed");
            }
            break;
        }
        case 5:
        {
            // Administrator login
            if (adminLogin())
            {
                int adminOption;
                do
                {
                    printf("\nAdministrator Options:\n");
                    printf("1. Update Ticket Rate\n");
                    printf("2. Update Snack Items\n");
                    printf("3. Add Movies\n");
                    printf("4. Exit Administrator Menu\n");
                    printf("Enter your option: ");
                    scanf("%d", &adminOption);
```

```
  switch (adminOption)
                    {
                    case 1:
                        updateTicketRate(&ticketRate);
                        break;
                    case 2:
                        updateSnackItems(snacks);
                        break;
                    case 3:
                        addMovies(movies);
                        break;
                    case 4:
                        printf("\nExiting Administrator Menu...\n");
                        break;
                    default:
                        printf("\n\033[31;1mInvalid option. Please try again.\033[0m\n"); // Red color
for error message
                    }
                } while (adminOption != 4);
            }
            break;
        }
        case 6:
            displayMovies(movies);
            break;
        case 7:
            printf("\nThank you for using Movie Ticket Booking System!\n");
            break;
        default:
            printf("\n\033[31;1mInvalid option. Please try again.\033[0m\n"); // Red color for error
message
        }
    } while (option != 6);

    return 0;
}
```

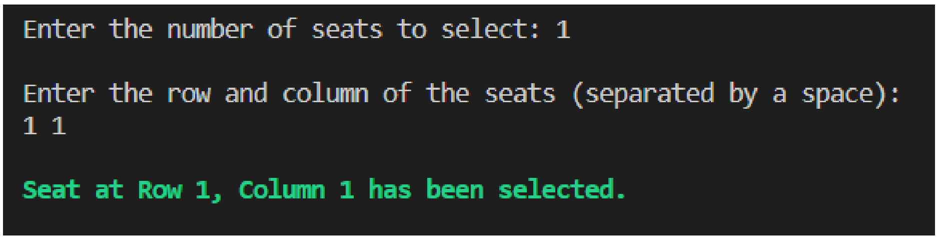
SOURCE CODE [HEADER FILE]:

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

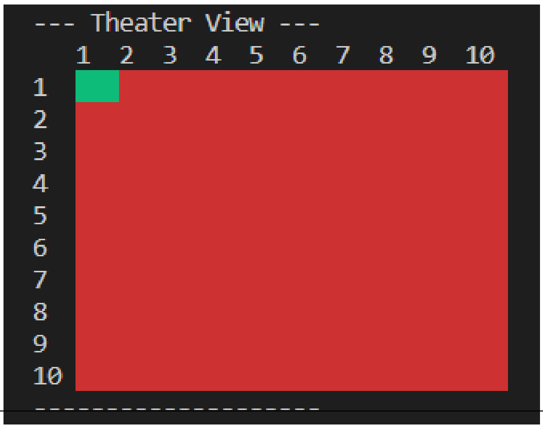
RESULTS:



```
--- Welcome to Movie Ticket Booking System ---

Options:
1. View Available Seats
2. Select Seat(s)
3. Cancel Selected Seats
4. Buy Selected Seats
5. Administrator Login
6. Show Available Movies
7. Exit
Enter your option: 1
'clear' is not recognized as an internal or external command,
operable program or batch file.

--- Theater View ---
   1  2  3  4  5  6  7  8  9  10
1  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
2  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
3  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
4  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
5  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
6  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
7  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
8  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
9  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
10 🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
--------------------
```



```
Enter the number of seats to select: 1

Enter the row and column of the seats (separated by a space):
1 1

Seat at Row 1, Column 1 has been selected.
```



```
--- Theater View ---
   1  2  3  4  5  6  7  8  9  10
1  🟩🟥🟥🟥🟥🟥🟥🟥🟥🟥
2  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
3  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
4  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
5  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
6  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
7  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
8  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
9  🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
10 🟥🟥🟥🟥🟥🟥🟥🟥🟥🟥
--------------------
```