

C Programming Project Report :

Name: Jiya Singh

Global Id: 590025179

Course Code: CSEG1032

Batch: 20

Semester -1

Project idea:

Show Ticket Booking
System

Abstract:

This project presents a console-based Show Ticket Booking System developed in C. The program allows users to select a show, view seats, book tickets with time slots and seat categories, and order food using an interactive text interface with coloured output.

It applies core C concepts such as loops, conditionals, input handling, and macros for colours.

The system also calculates the total amount payable and displays booking details, demonstrating how basic C programming can create a real-time user application.

Objective:

The objective of this project is to apply fundamental C programming concepts to design and implement an interactive, console-based ticket booking system. It aims to:

1. Use constants and macros (such as colour codes) to enhance display information clearly.
2. Implement a menu-driven program that allows users to perform operations like choosing a show, viewing seats, booking tickets, and ordering food.
3. Handle user input and validate selections for show options, seat numbers, food menu items, and timings.

4. Use conditional statements and loops to process bookings, calculate prices, and continuously allow user interaction.

5. Maintain and display booking details, including seat numbers, timing, user name, and total payable amount.

SYSTEM DESIGN AND ALGORITHM:

The system follows a procedural design with global and local variables used to maintain booking details and user selections.

Show and Seat Configuration

- Constants and macros (ANSI color codes) are used to improve the visual interface.
- Fixed seat layout (5×5) is displayed using nested loops to simulate seat numbering.

Global/Local Variables for Booking State:

- Integer variables track user inputs such as selected show, seat number, time slot, total cost, booking status, and repeated choices for food.
- Variables like seatnumber, timeslot, and bookingstatus help maintain the booking state throughout the program.

User Information Storage:

- A character array (name[50]) is used to store the customer's name during ticket booking.

Menu-Driven Algorithm:

- A while loop continuously displays the main menu and executes user-selected operations (choose show, view seats, book ticket, order food, view booking data).
- Conditional statements validate input and ensure that the user cannot proceed without selecting required options.

Price Calculation and Booking Output:

- Ticket price is calculated based on selected seat category (Classic or Recliner).

- Food prices are accumulated using a loop until ordering is completed.
- Final payable amount and booking details are displayed once the user confirms the booking.

Major Functions:

The core functionality of the console-based Show Ticket Booking System is implemented through menu-driven program logic and specific operations that manage booking, seat layout, pricing, and user input.

Show Selection:

Allows the user to choose from multiple available shows.

Validates the input and stores the selected show so that other features (seat booking, food ordering) can only be accessed afterward.

```
printf(YELLOW"Available Shows:\n"); // available shows show kar raha hai
printf("1. Movie: Thamma\n");
printf("2. Movie: Kantara\n");
printf("3. Comedy Show: Kapil Sharma Live\n");
printf("Select a show (1-3): "); //show choose karne ke liye
scanf("%d", &show);
printf("-----\n");
```

Seat Display:

Displays a 5×5 seat layout using nested loops. Seat numbers are printed sequentially, helping users choose a valid seat in the booking process.

```
if(show==-1)    //as show ki value change hojaegi choose krne par and hamne show ko initialize kara tha -1 se
printf("First choose your show Please\n ");
else
{ num=1;
  printf(GREEN"Seats for this shows:\n"); //seats ka arrangement show kardega
  printf("-----\n");
  for(int i = 0; i < rows; i++)
  {
    for(int j = 0; j < column; j++)
    {
      printf("%d ", num);
      num++;
    }
    printf("\n");
  }
}
```

Ticket Booking:

Handles booking operations including:

- Selecting a valid time slot
- Choosing a seat category (Classic or Recliner)

- Entering customer name
Stores the booking status, seat number, and computes the ticket price based on category.

Food Order Menu:

Provides a food and beverage list and allows the user to add multiple items.

Uses loops to accept choices repeatedly and calculates the total food bill until the user selects “Done ordering”.

```
total=0; //0 isiliye initialize kiya taaki garbage value na ae
r=0; // aage jaake iska loop chala ke baar chalaya taaki customer 1 se zada food option choose kar sake
printf(WHITE"====SELF SERVICE====\n"); //food services ke liye
printf("==== FOOD & BEVERAGE MENU =====\n");
printf("1.] Basic Salted Popcorn = RS.100\n");
printf("2.] Cheese Popcorn = RS.150\n");
printf("3.] PEPSI CAN = RS.80\n");
printf("4.] DIET COKE CAN = RS.80\n");
printf("5.] Basic Burger = RS.150\n");
printf("6.] **SPECIAL OFFER** ( Basic Popcorn + Drink) = RS.160\n");
printf("7.] Done ordering\n"); //jab order karna khtm hojaye
```

Show Booking Data:

Displays complete booking details such as:

- Customer name
- Seat number and show timing
- Ticket price + food total

Acts only if booking has already been made; otherwise shows an error message.

Main Algorithm (Overall System Workflow):

The execution of the Show Ticket Booking System is controlled through a continuous menu-driven algorithm that manages user selections, booking operations, and final ticket display. This procedural flow ensures ease of navigation and clear task execution:

Display Welcome Screen

Show a greeting message using coloured output to enhance user interaction.

Initialize Booking Variables:

Set default values such as show = -1 (no show selected), bookingstatus = 0 (no ticket booked), and cost counters (total = 0, ticketprice = 0).

Main Menu Loop:

Execute continuously while the user does not choose Exit.

Display options: Choose Show, Show Seats, Book Ticket, Order Food, View Booking, and Exit.

```
while(q==1) //baar baar loop chalane ke liye
{
    printf(BLUE"MAIN MENU\n"); //main menu me enter hone ke liye
    printf("PLEASE SELECT YOUR OPTION:\n");
    printf("1). Choose your Show\n");
    printf("2). Available Seats\n");
    printf("3). Book Ticket\n");
    printf("4). Order FOOD\n");
    printf("5). Show my booking data\n");
    printf("6). Exit\n");

    scanf("%d", &choice);
}
```

Show Selection:

If the user selects this option:

- . Display available shows.
- . Read user input and validate it.
- . Store the selected show; if invalid, show an error.

Seat Display:

If selected only after choosing a show:

- Use nested loops to print seat numbers (1-25).
- Help the user visualize available seat positions.

Ticket Booking:

If selected:

- Prompt for a valid time slot.
- Ask for seat number and categorize it (Classic/Recliner).
- Read customer name.
- Confirm booking and set bookingstatus = 1.

```

else
{
    if(seatnumber >= 1 && seatnumber <= 19)
        ticketprice = 400;
    else
        ticketprice = 700;

    printf("Enter your first name: "); //name daaldo ticket ke liye
    scanf("%s", name);

    printf(MAGENTA"Seat Number %d is successfully booked for %s\n",
           seatnumber, name);

    bookingstatus = 1;
}

```

Food Ordering:

If selected:

- Display a food menu.
- Allow repeated item selection using a loop.
- Update food total until the user selects “Done Ordering”.

Display Booking Data:

If selected:

- If a booking exists, show complete details: name, seat number, time, ticket cost + food total.
- If no booking exists, prompt the user to book a ticket first.

- ```
printf(WHITE "Seat Number %d is BOOKED for %s, timing %d:00, Your TOTAL PAYABLE AMOUNT is %d\n",
 seatnumber, name, timeslot, total + ticketprice); //saari info show krdega ki kisne,konsi seat,konsa show book kia
```

## Exit Program:

If the user selects Exit:

- Display a Thank You message and terminate the main loop, ending the program.

This structured workflow simplifies expansion and debugging, allowing easy upgrades such as seat availability tracking,

payment integration, or online booking support.

## IMPLEMENTATION DETAILS

### Key Language Features Used:

#### Constants and Macros

- #define is used to create ANSI color codes (e.g., RED, GREEN, YELLOW), improving the user interface without changing program logic.
- These macros allow easy customization of display formatting.

```
//colours add kardiye thode
#define RED "\x1b[31m"
#define GREEN "\x1b[32m"
#define YELLOW "\x1b[33m"
#define BLUE "\x1b[34m"
#define MAGENTA "\x1b[35m"
#define CYAN "\x1b[36m"
#define WHITE "\x1b[37m"
```

## Variables for Booking State

- Integer variables track program status, including:
  - show, seatnumber, timeslot, ticketprice, total, bookingstatus
  - Loop control variables (q, r) for menu repetition and food ordering.
- A character array name[50] stores the customer's name, enabling ticket personalization.

```
int q=1,r=0,num =1;
int rows = 5, column = 5;
int choice,show=-1;
int seatnumber, total = 0, foodChoice;
char name[50];
int bookingstatus=0;
int ticketprice,timeslot;
```

## Standard Library Functions

- stdio.h provides printf, scanf, and gets/puts-like functionality for user interaction, menu display, and reading inputs.
- Loop constructs from <stdio.h> and standard C syntax enable seat layout generation and repeated food ordering.

## Procedural Program Structure

- The system uses nested loops for seat numbering and food selection.

- Conditional statements (if-else) validate time slots, show selection, and seat categories.
- The menu-driven logic inside a continuous while loop allows repeated operations until the user chooses to exit.

### User Interface Enhancement

- ANSI color codes make error messages, menus, and confirmations visually distinct.
- This improves readability and gives a better experience even in a text-only environment.

## Control Structures and Program Logic

### Control Structures

- The system relies heavily on loops (while, for) and conditional statements (if-else).

- . A while loop repeatedly displays the main menu until the user selects Exit.
- . for loops are used to print the seat layout (1-25) in a  $5 \times 5$  grid.
- . if-else statements validate selections such as show choice, time slot, seat category, and food items.

## Program Logic and Validation

The system ensures correct booking through a series of checks:

### Show Selection Validation

- . Ensures that the user picks only from the available shows (options 1-3).
- . Prevents further actions (like booking or food ordering) without choosing a show first.

## Seat Booking and Category Check

- Validates whether the seat number is within 1-25.
- Determines ticket price based on the category:
  - Seat 1-19 → Classic
  - Seat 20-25 → Recliner

```
printf("Enter seat number (1-25) for booking:\n ");
printf("(1-19) are CLASSIC (Rs.400/-)\n");
printf("(20-25) are RECLINER (Rs.700/-)\n ");
scanf("%d", &seatnumber);
```

◦

## Time Slot Validity

- Only accepts valid fixed slots (e.g., 09:00, 11:00, 13:00, etc.).
- If invalid, the program prompts the user again without exiting the menu loop.

```
printf("PLEASE SELECT YOUR TIME SLOT:\n");
printf("09:00 11:00 13:00\n");
printf("15:00 17:00 18:00\n");
printf("ENTER YOUR SLOT TIME IN --:00 FORM\n");
scanf("%d", ×lot);
```

- 

## Food Ordering Logic

- A loop repeatedly accepts food choices.
- Each valid item adds to the total bill; selecting option 7 ends the order.
- Invalid entries display error messages and prompt re-entry.

## User Input Handling

- scanf is used to read show numbers, seat numbers, time slots, and food choices.
- A character array stores the customer's name, allowing personalized booking.

- Booking details are only displayed if bookingstatus is set, preventing incomplete or accidental viewing.

This structured combination of loops, checks, and input handling ensures that only valid operations can proceed, making the system user-friendly and robust.

# Platform Considerations and Program Termination:

## Platform Considerations

- The program uses ANSI escape color codes to display text in different colors, improving readability in a console environment.
- These escape sequences work on most Linux/Unix terminals and modern Windows terminals.
- Because the program does not depend on system-specific commands, it remains portable across platforms without modification.

## Program Termination

- . The main menu runs inside a loop until the user chooses the Exit (Option 6).
- . Once Exit is selected:
  - A closing message is displayed through colored text for better interaction.
  - The loop condition is updated, stopping further execution.
- . All allocated resources are automatically cleared on exit since the program only uses stack variables and no dynamic memory.

# PROBLEM DEFINITION

This project creates a simple, text-based Pong game in C for two players. It runs in the terminal without graphics libraries, making it ideal for beginners. The program handles player names, paddle movement, ball motion, collisions, scoring, and win detection, all within a single C file that works on any standard compiler.