



EAST WEST UNIVERSITY

CSE103: Structured Programming  
[Fall 2024]

Project Report  
Hotel Management System

Course Code : CSE103  
Course Title : Structured Programming  
Instructor Name : Md. Asif Khan Rifat  
Section : 29  
Group Number : 05  
Group Name : Team Scorpion  
Date of Submission: 16/01/2025

Submitted by:

Student ID	Student Name	Contribution
2024-3-60-447	Riad Mahmud	Initialization, Checkout
2024-3-60-448	Al Amin	Login, Display Room
2024-3-60-449	Sajid Ahnaf Sahaf	Book Room, File Handling
2024-3-60-450	Syeda Raisa Tahsin	Add Room, Edit Room

# Introduction

The objective of this project is to develop a Hotel Management System that allows hotel administrators and users to manage room bookings, check-ins, and check-outs efficiently. The system aims to streamline hotel operations and enhance customer service.

The project involves creating a software application for managing hotel operations. It includes functionalities such as adding rooms, booking rooms, checking in and checking out guests, and storing customer and room information.

Programming Language: C

IDE: Code Blocks, VS Code.

# Problem Statement

Managing a hotel with numerous rooms and guests can be challenging. Manual record-keeping is prone to errors, and tracking room availability, bookings, and customer details becomes cumbersome. This project aims to automate these processes, reducing errors and improving efficiency. This project streamlines these processes by providing a computerized solution that is both efficient and user-friendly.

An automated Hotel Management System ensures accurate records, efficient room allocation, and better customer service. It helps hotel staff manage tasks effectively and provides guests with a seamless booking experience.

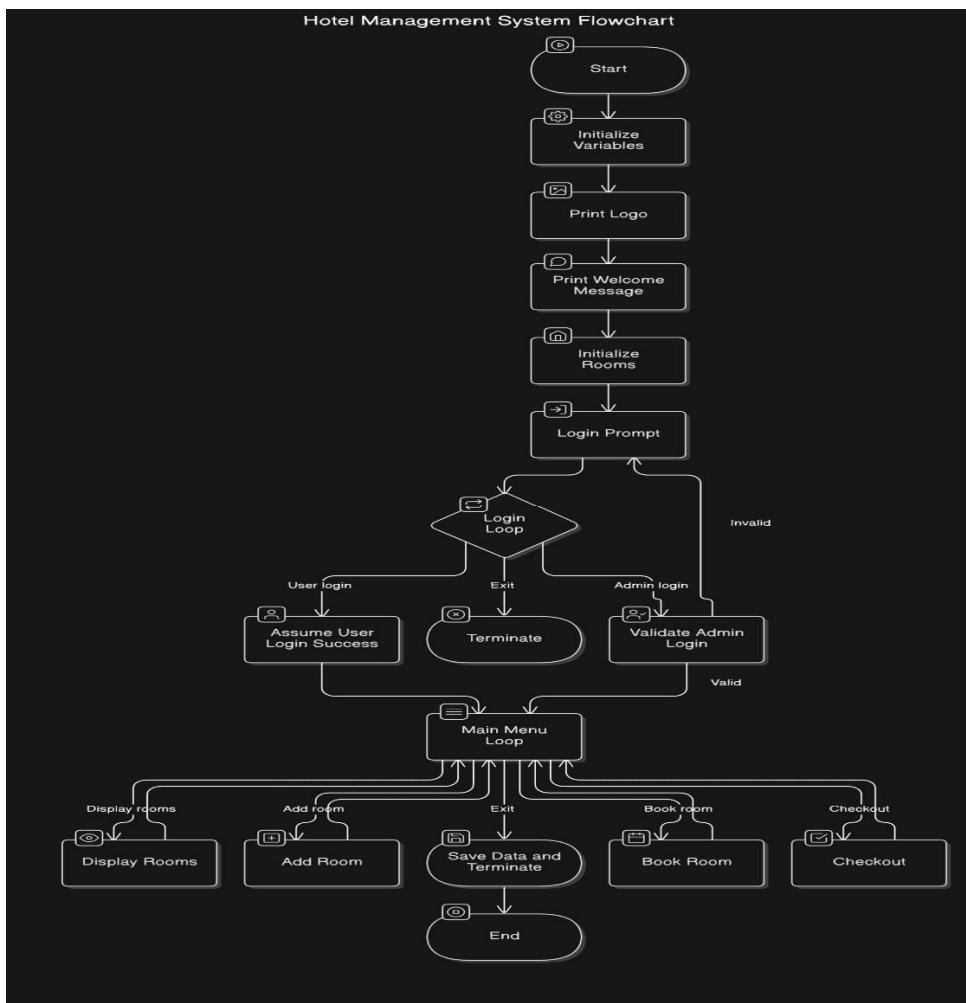
# Functional Requirements

The core functionalities of the Hotel Management System include:

1. User Authentication (Admin/User Login)
2. Displaying Available Rooms
3. Adding New Rooms (Admin only)
4. Booking Rooms
5. Checking Out Guests (Admin only)
6. Saving and Displaying Room and Customer Details

# Project Design

## 1. Flow chart



## 2.Pseudocode

1. Start
2. Declare Constants and Structures  
Define constants (e.g., MAX\_ROOMS, MAX\_FLOORS)  
Define structure Room, structure Customer
3. Global Variables Declaration  
Declare array hotelRooms[MAX\_ROOMS],  
array customers[MAX\_CUSTOMERS],  
integer roomCount, integer userCount
4. Function Prototypes Declaration  
Declare function print\_logo(), login(),  
showAdminMenu(), showUserMenu(), initializeRooms(),  
AddRoom(), edit\_Room(), displayRooms(), book\_room(), checkout(),  
calculatePrice(), isPeakSeason(), savefile(),  
saveAndDisplayRoomDetails(), saveAndDisplayBookedUsers(),  
displaySavedUsers()
5. Main Function  
Call print\_logo(), initializeRooms()  
Print "Welcome to Scorpio Stay Hub!"  
Prompt for login type (Admin/User)  
Call login() and store result in isLoggedIn  
Login Validation  
If isLoggedIn and userType == Admin  
Print "Admin login successful!"  
Else If isLoggedIn and userType == User  
Print "User login successful!"  
Else  
Print "Login failed!"  
Exit

## Menu Loop

```
While choice is not equal to exit value (e.g., 5)
If userType == Admin
    Call showAdminMenu()
Else If userType == User
    Call showUserMenu()
Get user input for menu choice
Switch choice
Case 1: Call displayRooms()
Case 2: If userType == Admin then Call AddRoom()
Case 3: Call book_room()
Case 4: If userType == Admin then Call checkout()
Case 5: Call savefile()
Case 6: Call saveAndDisplayBookedUsers()
Case 7: Call displaySavedUsers()
Case 8: Call saveAndDisplayRoomDetails()
```

## 6. Function Definitions

Define function print\_logo(), login(),  
showAdminMenu(), showUserMenu(), initializeRooms(),  
AddRoom(), edit\_Room(), displayRooms(), book\_room(), checkout(),  
calculatePrice(), isPeakSeason(), savefile(),  
saveAndDisplayRoomDetails(), saveAndDisplayBookedUsers(),  
displaySavedUsers()

## 7. End

### 3.Modular Design

The project is divided into multiple functions, each responsible for specific tasks:

`main()`:

Controls the overall flow of the program.

`initializeRooms()`:

Initializes room details.

`login()`:

Handles user authentication.

`showAdminMenu()`, `showUserMenu()`:

Display menus for admin and user.

`AddRoom()`:

Adds a new room (Admin only).

`displayRooms()`:

Displays available rooms.

`book_room()`:

Books a room.

`checkout()`:

Checks out a guest (Admin only).

`calculatePrice()`, `isPeakSeason()`:

Calculates room price based on duration and season.

`savefile()`, `saveAndDisplayRoomDetails()`, `saveAndDisplayBookedUsers()`,

`displaySavedUsers()`:

Handle file operations for saving and displaying details.

# Code Implementation

## 1. main () function :

```
int main()
{
    int choice = 0, a = 0;

    print_logo();
    printf("\n\nWelcome to the Hotel %s\n", HOTEL_NAME);
    // printf("Enter the total number of floors in the hotel: ");
    // scanf("%d", &MAX_FLOORS);
    // getchar();

    initializeRooms(); // first e rooms Initialize kotechi based on the total
number of floors.
    printf("\nPlease login to access the system:\n");
    int loginType = 0;
    while (loginType != 3)
    {
        // login type: Admin or User
        printf("1. Admin Login\n");
        printf("2. User Login\n");
        printf("3. Exit\n");
        printf("Enter your choice (1 for Admin, 2 for User): ");
        scanf("%d", &loginType);
        getchar();

        int isLoggedIn = 0;

        // Login validation for Admin or User
        if (loginType == 1)
        {
            isLoggedIn = login(ADMIN_USERNAME, ADMIN_PASSWORD);
            if (isLoggedIn)
            {
                printf("\nAdmin login successful!\n");
                break;
            }
            else
            {
                printf("\nAdmin login failed. Please try again.\n");
            }
        }
        else if (loginType == 2)
    }
```

```

{
    isLoggedIn = 1; // Assume user login is always successful for now
    if (isLoggedIn)
    {
        printf("\nUser login successful!\n");
        break;
    }
    else
    {
        printf("\nUser login failed. Please try again.\n");
    }
}
else if (loginType == 3)
{
    printf("\nExiting system. Goodbye!\n");
    return 0;
}
else
{
    printf("\nInvalid choice. Please try again.\n");
}
}

// Menu loop for Admin or User
while (choice != 7)
{
    if (loginType == 1) // Admin menu
    {
        showAdminMenu();
    }
    else // User menu
    {
        showUserMenu();
    }

    scanf("%d", &choice);
    getchar();

    switch (choice)
    {
        case 1: // Display rooms
            displayRooms();
            break;
        case 2: // Add a room (Admin only)
            if (loginType == 1)
            {

```

```

        AddRoom();
    }
    else
    {
        printf("Access denied. Only Admin can add rooms.\n");
    }
    break;
case 3: // Book a room
    if (loginType == 1)
    {
        edit_Room();
    }
    else
    {
        printf("Access denied. Only Admin can add rooms.\n");
    }
    break;
case 4:
    book_room();
    break;
case 5:
    if (loginType == 1)
    {
        editBookingDetails();
    }
    else
    {
        printf("Access denied. Only Admin can perform checkout.\n");
    }
case 6:
    if (loginType == 1)
    {
        checkout();
    }
    else
    {
        printf("Access denied. Only Admin can perform checkout.\n");
    }
    break;
case 7:// Exit
    printf("Exiting the system. Goodbye!\n");
    save_book_users_data();
    save_Room_Details_data();
    return 0;
case 8:

```

```

        printf("Enter the index for student: ");
        scanf("%d", &a);
        print_bill(a);
        break;
    case 9:
        display_Room_Details_data();
        break;
    case 10:
        display_book_users_data();
        break;
    default: // Invalid input
        printf("Invalid choice. Please try again.\n");
    }
}
return 0;
}

```

The main() function displays the hotel logo, initializes the rooms in the hotel, manages the login type (Admin or User) and performs login validation, based on the user type (Admin or User), it displays different menus and handles various operations like displaying rooms, adding rooms, booking rooms, checking out, saving data, and displaying booked user data.

Screenshot:

```

(
 )\  ( ) ( / ( ) ) \ ( )\ ) ( ) / ( ( ) ( ) ( ) ( ) ( ) ( )
((_) ( ( ( / ( ( ) \ O | / ((_) / ( _ ) ) \ ) \ O ) ( ) \ ( ) / ( ) \ O )
) \ _ _ ) \ ) ( _ ) ) \ ( _ ) ) / ( _ ) / ( _ ) ) ( ) \ ( ) / ( ) ( ) / ( ) / ( )
( ( / | ( ( ) ( ( ) ( ( ) | - ( ( ) | | | - ( ) | | | - ( ) ( ( ) | | | - ( )
| ( _ / - \ _ ` - < - / - ` | | | / - ) | - | ' / - ) - | - | - |
\ _ \ _ \ _ \ _ , - / - \ _ \ _ \ _ , - | | | - \ _ | | \ _ \ _ \ _ , - | \ _ \ _ |
by Team Scorpion

Welcome to the Hotel Paradise

Please login to access the system:
1. Admin Login
2. User Login
3. Exit
Enter your choice (1 for Admin, 2 for User): |

```

```
( )\ ) ) ( /(\ ) )\ ((/\\ ( ( /(( ( ) ( /(
((_) ( ( ( /(\ ( )\()| /(((_) /(_))))\ )\()(( ))\(( /(\ )\()
)\_\_)\\ )\()())\((\_)//(\_)_ | ( ) /((\_)_)((\_)\\ /((\_)_)|_)/
((/_|((_)((_)_((_)_|((_)_| | | _ ( )| | _ ((\_)_)((\_)_|_|_
| (/_ - \/_ ^_-< /_ ^_- | | / -) | | . / -) ^_- | | -|
\_\_\_/\_,/_/\_/\_\_,/_/ | | \_\_ | \_\_ | \_\_ \_\_ | \_\_ | \_\_ |
```

by Team Scorpion

Welcome to the Hotel Paradise

Please login to access the system:

1. Admin Login
2. User Login
3. Exit

Enter your choice (1 for Admin, 2 for User): 1

Enter Username: admin

Enter Password: admin123

Admin login successful!

```
*****
*      Welcome to HOTEL Paradise      *
*****
*  1. Display Available Rooms      *
*  2. Add a Room                  *
*  3. Edit Room details           *
*  4. Room Booking                *
*  5. Edit Booking Details        *
*  6. Checkout                    *
*  7. Exit                        *
*****
```

Enter your choice: ■

```
( )\ ) ) ( / ( ) )\ ( ()\ \ ( ( / ( ( ( ) ( ( ) ( / ( )
((_) ( ( ( / ( ( ) \ () | / ((_) / ( _))\ ) \ () ( ) )\ ( / ( ) \ () )
) \ _ )\ ) ( _))\ ) ( _)/ ( _))_ ( _)/ ( ( _)) ( () \ / ( ( ) ( _)) / /
( / _ )| ( ( ) ( ( ) | ( ( ) | _| _ ( ) | _| ( ( _)) ( ( ) | _| _| ( )
| ( _| - \ _| ^ _| < _| / _| | _| / _| ^ _| < _| / _| | _| | _| | _| | _| | _|
```

by Team Scorpion

Welcome to the Hotel Paradise

Please login to access the system:

1. Admin Login
2. User Login
3. Exit

Enter your choice (1 for Admin, 2 for User): 2

User login successful!

```
*****
*      Welcome to HOTEL Paradise      *
*****
*  1. Display Available Rooms      *
*  4. Room Booking                 *
*  7. Exit                         *
*****
```

Enter your choice: ■

## 2. initializeRooms() function :

```
void initializeRooms()
{
    int roomNumber = 1;
    char *type[] = {"single", "double", "deluxe"};
    char *view[] = {"sea view", "city view"};
    char *ac_type[] = {"ac", "non ac"};
    int basePrice[] = {1350, 1200, 1000, 1500};
    int basePrice1[] = {3000, 5000};
    for (int floor = 1; floor <= MAX_FLOORS; floor++) // total floor in building
    {
        for (int i = 1; i < 5; i++) // akta floor e 4 ta room
        {
            hotelRooms[roomCount].roomNumber = (floor * 100) + i; // one kore
            room number bariteche // new fix room number by floor

            hotelRooms[roomCount].floor = floor; // for each iteration floor
            remain same
            // (rand() % 2 == 0)                                // amni akt logic to get
            value for other parameter // By sAhAf@@
            strcpy(hotelRooms[roomCount].type, type[rand() % 3]);
            strcpy(hotelRooms[roomCount].view, view[rand() % 2]);
            strcpy(hotelRooms[roomCount].ac_type, ac_type[rand() % 2]);
            if ((!strcmp(hotelRooms[roomCount].view, "Sea View") &&
!strcmp(hotelRooms[roomCount].ac_type, "AC Room")) ||
!strcmp(hotelRooms[roomCount].type, "deluxe"))
            {

                hotelRooms[roomCount].basePrice = basePrice1[rand() % 2];
            }
            else
            {
                hotelRooms[roomCount].basePrice = basePrice[rand() % 4];
            }
            hotelRooms[roomCount].isOccupied = 0;
            roomCount++;
        }
    }
    // printf("Default rooms initialized!\n");
}
```

The initializeRooms() function initializes the room details such as room number, type, view, and base price.

### 3. login() function:

```
int login(char *username, char *password)
{
    char inputUsername[20], inputPassword[20];

    printf("Enter Username: ");
    scanf("%s", inputUsername);
    printf("Enter Password: ");
    scanf("%s", inputPassword);

    if (strcmp(username, inputUsername) == 0 && strcmp(password,
inputPassword) == 0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

The login() function validates the user credentials for admin and user login.

### 4. showAdminMenu() function, showUserMenu() function:

```
void showAdminMenu()
{
    printf("\n*****\n");
    printf("*      Welcome to HOTEL %s      *\n", HOTEL_NAME);
    printf("*****\n");
    // printf("*          ADMIN          *\n");
    printf("* 1. Display Available Rooms      *\n");
    printf("* 2. Add a Room      *\n");
    printf("* 3. Edit Room details      *\n");
    printf("* 4. Room Booking      *\n");
    printf("* 5. Edit Booking Details      *\n");
    printf("* 6. Checkout      *\n");
    printf("* 7. Exit      *\n");
    printf("*****\n");
    printf("Enter your choice: ");
}
```

```

void showUserMenu()
{
    printf("\n*****\n");
    printf("*      Welcome to HOTEL %s      *\n", HOTEL_NAME);
    printf("*****\n");
    printf("*  1. Display Available Rooms      *\n");
    printf("*  4. Room Booking                 *\n");
    printf("*  7. Exit                         *\n");
    printf("*****\n");
    printf("Enter your choice: ");
}

```

The showAdminMenu() function, showUserMenu() function display menu options for admin and user respectively.

### 5. AddRoom() function:

```

void AddRoom()
{
    int roomNumber, floor, basePrice, isOccupied;
    char type[20], view[20], ac_type[20];

    if (roomCount < MAX_ROOMS) // Ensure roomCount is less than MAX_ROOMS
    {
        // Prompt user for room details
        printf("Enter Room Number: ");
        scanf("%d", &roomNumber);
        printf("Enter Floor: ");
        scanf("%d", &floor);
        getchar();
        printf("Enter Room Type (e.g., Single, Double, Deluxe): ");
        scanf("%[^\\n]s", type);
        getchar();
        printf("Enter View (e.g., Sea, City): ");
        scanf("%[^\\n]s", view);
        getchar();
        printf("Enter AC type (AC, Non AC): ");
        scanf("%[^\\n]s", ac_type);
        getchar();
        printf("Enter Base Price: ");
        scanf("%d", &basePrice);
        printf("Is the room occupied? (1 for Yes, 0 for No): ");
        scanf("%d", &isOccupied);

        // Populate hotelRooms array
    }
}

```

```

        hotelRooms[roomCount].roomNumber = roomNumber;
        hotelRooms[roomCount].floor = floor;
        hotelRooms[roomCount].basePrice = basePrice;
        hotelRooms[roomCount].isOccupied = isOccupied;
        strcpy(hotelRooms[roomCount].type, type);
        strcpy(hotelRooms[roomCount].view, view);
        strcpy(hotelRooms[roomCount].ac_type, ac_type);

        roomCount++;
        printf("\n\n ----- Room added successfully.----- \n");
        printf("Room Number: %d\nFloor: %d\nRoom Type: %s\nView: %s\nAC Type:
%s\nBase Price: %d\nOccupied: %d\n",
               roomNumber, floor, type, view, ac_type, basePrice,
               isOccupied);
    }
    else
    {
        printf("Sorry, Can not add more room ! Please try later.\n");
    }
}

```

The AddRoom() function allows admin to add new rooms to the system.

#### 6. editRoom() function:

```

void edit_Room()
{
    int roomNumber, floor, basePrice, isOccupied, edit = 0;
    char type[20], view[20], ac_type[30];
    printf("Enter Room Number: ");
    scanf("%d", &roomNumber);
    for (int i = 0; i < roomCount; i++)
    {
        if (hotelRooms[i].roomNumber == roomNumber &&
            hotelRooms[i].isOccupied == 0) // Ensure roomCount is less than MAX_ROOMS
        {
            printf("Enter Floor: ");
            scanf("%d", &floor);
            getchar();
            printf("Enter Room Type (e.g., Single, Double,): ");
            scanf("%[^\\n]s", type);
            getchar();
            printf("Enter View (e.g., Sea, City): ");
            scanf("%[^\\n]s", view);
            getchar();
            printf("Enter AC type (AC, Non AC): ");

```

```

        scanf("%[^\\n]s", ac_type);
        getchar();
        printf("Enter Base Price: ");
        scanf("%d", &basePrice);
        printf("Is the room occupied? (1 for Yes, 0 for No): ");
        scanf("%d", &isOccupied);

        hotelRooms[i].floor = floor;
        hotelRooms[i].basePrice = basePrice;
        hotelRooms[i].isOccupied = isOccupied;
        strcpy(hotelRooms[i].type, type);
        strcpy(hotelRooms[i].view, view);

        printf("\n\n    ----- Room edited successfully.----- \n");
        printf("Room Number: %d\nFloor: %d\nRoom Type: %s\nView: %s\nBase
Price: %d\nOccupied: %d\n",
               roomNumber, floor, type, view, basePrice, isOccupied);
        edit = 1;
    }
}
if (!edit)
{
    printf("Cannot add more rooms. Maximum capacity reached.\n");
}
}

```

## 7. displayRooms() function:

```

void displayRooms()
{
    printf("\nRoom Data:\n");
    printf("-----\n");
    printf(" | Room No | Floor | Type      | View       | AC Type      |
Occupied | Price   | \n");
    printf("-----\n");
    for (int i = 0; i < roomCount; i++)
    {
        printf(" | %-7d | %-5d | %-8s | %-10s | %-11s | %-8s | %-6d | \n",
               hotelRooms[i].roomNumber,
               hotelRooms[i].floor,
               hotelRooms[i].type,
               hotelRooms[i].view,
               hotelRooms[i].ac_type,

```

```

        hotelRooms[i].isOccupied ? "Yes" : "No",
        hotelRooms[i].basePrice);
    }

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

```

The displayRooms() function displays the list of available rooms.

#### 8. book\_room() function:

```

void book_room()
{
    int roomNumber = 0;
    char type[30], view[30], ac[30];
    int foundroom[MAX_ROOMS];
    int foundcount = 0;
    int room_found = 0, stayDuration, month;

    printf("Enter the room type (single/double): ");
    scanf("%[^\\n]s", type);
    getchar();

    // Filter by type and store it in custom local array
    for (int i = 0; i < roomCount; i++)
    {
        if (hotelRooms[i].isOccupied == 0 && strcmp(hotelRooms[i].type, type)
== 0)
        {
            foundroom[foundcount++] = i; // adding all required type to
            foundroom
        }
    }

    printf("Do you want ac or non ac room? -> ");
    scanf("%[^\\n]s", ac);
    getchar();

    // Filter by ac
    for (int i = 0; i < foundcount;)
    {
        if (strcmp(hotelRooms[foundroom[i]].ac_type, ac) != 0)
        {
            for (int j = i; j < foundcount - 1; j++)
            {

```

```

        foundroom[j] = foundroom[j + 1]; // match share gulo remove
    }
    foundcount--;
}
else
{
    i++;
}
}

printf("Do you prefer Sea View or City View? ");
scanf("%[^\\n]s", view);
getchar();

// Filter by view
for (int i = 0; i < foundcount;)
{
    if (strcmp(hotelRooms[foundroom[i]].view, view) != 0)
    {
        for (int j = i; j < foundcount - 1; j++)
        {
            foundroom[j] = foundroom[j + 1]; // removing the not match
item
        }
        foundcount--;
    }
    else
    {
        i++;
    }
}

if (foundcount == 0)
{
    printf("No available rooms match your criteria.\n");
}
else
{
    printf("\nRoom Data:\n");
    printf("-----\n");
    printf("-----\n");
    printf(" | Room No | Floor | Type      | View          | AC Type      |\n");
Occupied | Price   | \n");
    printf("-----\n");
    printf("-----\n");
}

```

```

for (int i = 0; i < foundcount; i++)
{
    printf("| %-7d | %-5d | %-8s | %-10s | %-11s | %-8s | %-6d |\n",
           hotelRooms[foundroom[i]].roomNumber,
           hotelRooms[foundroom[i]].floor,
           hotelRooms[foundroom[i]].type,
           hotelRooms[foundroom[i]].view,
           hotelRooms[foundroom[i]].ac_type,
           hotelRooms[foundroom[i]].isOccupied ? "Yes" : "No",
           hotelRooms[foundroom[i]].basePrice);
}

printf("-----\n");
printf("Enter the Room Number to book: ");
scanf("%d", &roomNumber);
getchar();
// printf("%d\n", roomNumber);
}

for (int i = 0; i < max_user; i++)
{
    c[i].roomNumber = 0;
}

for (int i = 0; i < foundcount; i++)
{
    int idx = foundroom[i];
    if (hotelRooms[idx].roomNumber == roomNumber)
    {
        // printf("%d\n", hotelRooms[foundroom[i]].roomNumber);
        room_found = 1;
        for (int j = 0; j < max_user; j++)
        {
            if (c[j].roomNumber == 0)
            {
                // printf("%d", j);
                // By sAhAf☺
                printf("Name: ");
                scanf("%[^\\n]s", c[j].name);
                getchar();

                printf("Phone Number: ");
                scanf("%[^\\n]s", c[j].phoneNumber);
            }
        }
    }
}

```

```

        getchar();
        printf("Email: ");
        scanf("%[^\\n]s", c[j].email);
        getchar();
        printf("Enter Address: ");
        scanf("%[^\\n]s", c[j].address);
        getchar();
        printf("Enter ID Type (e.g., Passport, Driver's License):
");
        scanf("%[^\\n]s", c[j].idType);
        getchar();
        printf("Enter ID Number: ");
        scanf("%[^\\n]s", c[j].idNumber);
        getchar();
        printf("Enter the duration of stay (in days): ");
        scanf("%d", &stayDuration);
        getchar();
        printf("Enter the current month (1-12): ");
        scanf("%d", &month);
        getchar();
        printf("Enter Check-in Date (dd/mm/yyyy): ");
        scanf("%[^\\n]s", c[j].checkInDate);
        getchar();
        printf("Enter Check-out Date (dd/mm/yyyy): ");
        scanf("%[^\\n]s", c[j].checkOutDate);
        getchar();
        c[j].roomNumber = roomNumber;
        c[j].base_price = hotelRooms[idx].basePrice;
        c[j].total_price =
calculatePrice(&hotelRooms[foundroom[i]], stayDuration, month);
        hotelRooms[foundroom[i]].isOccupied = 1;
        // printf("Mr/Ms %s. You room %d have successfully been
booked! \nTotal price for %d days is: %d\n",
           //      c[j].name, roomNumber, stayDuration,
totalPrice);
        printf("\n\n\n");
        printf("*****\n*****\n");
        printf("          Dear %s, \n", c[j].name);
        printf("          Your booking for room number %d \n",
roomNumber);
        printf("          The total price for your stay of %d days
is: %d \n", stayDuration, c[j].total_price);
        printf("          Thank you for choosing our hotel. We
look forward to your stay.\n");

```

```

        printf("*****\n");
        printf("\n");
        printf("\t\tYour bill have been printed\n\n\n");
        print_bill(j);
        userCount++;
        break;
    }
}
break;
}
}
if (room_found == 0 && foundcount != 0)
{
    printf("The given room doesn't meet your requirement.\n");
}
}

```

The book\_room() function handles room booking by capturing customer details and updating room status.

#### 9. checkout() function:

```

void checkout()
{
    int roomNumber = 0;
    printf("Enter Room number: ");
    scanf("%d", &roomNumber);
    for (int i = 0; i < roomCount; i++)
    {
        if (roomNumber == hotelRooms[i].roomNumber)
        {
            if (hotelRooms[i].isOccupied)
            {
                hotelRooms[i].isOccupied = 0;
                printf("You have been successfully checked out from Room %d!\n",
                hotelRooms[i].roomNumber);
            }
            else{
                printf("Room %d is already Unoccupied. So checkout is not
possible.\n", hotelRooms[i].roomNumber);
            }
            break;}}}

```

The checkout() function allows admin to check out guests and update room status.

## 10. calculatePrice() function:

The calculatePrice() function calculates the total price based on room type, stay duration, and peak season.

## 11. isPeakSeason() function:

The isPeakSeason() function checks if the given month is within the peak season.

## 12. File Operations:

```
save_book_users_data() and display_book_users_data()  
ave_Room_Details_data() and display_Room_Details_data()  
and print_bill()
```

# Testing and Results

Test Cases:

Test Case	Test Input	Expected Output	Actual Output
1	Admin Login with correct credentials	Admin login successful!	Admin login successful!
2	User Login	User login successful!	User login successful!
3	Display Rooms	Displays list of available rooms	Displays list of available rooms
4	Add a Room (Admin only)	Room added successfully message	Room added successfully
5	Book a Room with valid room number	Booking confirmation message	Booking confirmation
6	Checkout (Admin only)	Checkout confirmation message	Checkout confirmation message
7	Save Room Details	Room details saved successfully	Room details saved successfully
8	Save Booked Users	Booked users saved successfully	Booked users saved successfully

## Results and Screenshots:

### Test case 1:

```
Welcome to the Hotel Paradise

Please login to access the system:
1. Admin Login
2. User Login
3. Exit
Enter your choice (1 for Admin, 2 for User): 1
Enter Username: admin
Enter Password: admin123

Admin login successful!
```

### Test case 2:

```
Welcome to the Hotel Paradise

Please login to access the system:
1. Admin Login
2. User Login
3. Exit
Enter your choice (1 for Admin, 2 for User): 2

User login successful!
```

Test case 3:

```
*****
*      Welcome to HOTEL Paradise      *
*****
* 1. Display Available Rooms      *
* 4. Room Booking                *
* 7. Exit                         *
*****
```

Enter your choice: 1

Room Data:

Room No	Floor	Type	View	AC Type	Occupied	Price
101	1	deluxe	city view	ac	No	3000
102	1	deluxe	sea view	ac	No	3000
103	1	double	sea view	non ac	No	1200
104	1	double	city view	non ac	No	1500
201	2	double	sea view	non ac	No	1350
202	2	single	sea view	ac	No	1200
203	2	double	sea view	non ac	No	1350
204	2	deluxe	city view	non ac	No	3000
301	3	deluxe	sea view	non ac	No	3000
302	3	deluxe	city view	non ac	No	3000
303	3	deluxe	city view	ac	No	5000
304	3	single	sea view	non ac	No	1500
401	4	deluxe	sea view	non ac	No	3000
402	4	double	city view	non ac	No	1500
403	4	deluxe	city view	non ac	No	3000
404	4	double	city view	ac	No	1000
501	5	single	sea view	ac	No	1000
502	5	double	sea view	ac	No	1200
503	5	deluxe	city view	ac	No	3000
504	5	single	city view	non ac	No	1350
601	6	single	city view	ac	No	1000
602	6	single	sea view	ac	No	1350
603	6	deluxe	sea view	ac	No	5000
604	6	deluxe	city view	non ac	No	3000

Test case 4:

```
Admin login successful!  
*****  
*      Welcome to HOTEL Paradise      *  
*****  
* 1. Display Available Rooms      *  
* 2. Add a Room                  *  
* 3. Edit Room details          *  
* 4. Room Booking                *  
* 5. Edit Booking Details        *  
* 6. Checkout                   *  
* 7. Exit                      *  
*****  
Enter your choice: 2  
Enter Room Number: 100  
Enter Floor: 1  
Enter Room Type (e.g., Single, Double): single  
Enter View (e.g., Sea, City): sea view  
Enter AC type (AC, Non AC): ac  
Enter Base Price: 1100  
Is the room occupied? (1 for Yes, 0 for No): 0
```

----- Room added successfully.-----

```
Room Number: 100  
Floor: 1  
Room Type: single  
View: sea view  
AC Type: ac  
Base Price: 1100  
Occupied: 0
```

Test case 5:

```
*****
*      Welcome to HOTEL Paradise      *
*****
* 1. Display Available Rooms      *
* 2. Add a Room                  *
* 3. Edit Room details           *
* 4. Room Booking                *
* 5. Edit Booking Details         *
* 6. Checkout                     *
* 7. Exit                         *
*****
Enter your choice: 4
Enter the room type (single/double): single
Do you want ac or non ac room? -> ac
Do you prefer Sea View or City View? sea view

Room Data:
-----
| Room No | Floor | Type      | View       | AC Type    | Occupied | Price   |
| 202     | 2      | single    | sea view   | ac          | No        | 1200    |
| 501     | 5      | single    | sea view   | ac          | No        | 1000    |
| 602     | 6      | single    | sea view   | ac          | No        | 1350    |
| 902     | 9      | single    | sea view   | ac          | No        | 1000    |
-----
Enter the Room Number to book: 202
Name: Tahsin
Phone Number: 12345678
Email: tt123@gmail.com
Enter Address: Dhaka
Enter ID Type (e.g., Passport, Driver's License): passport
Enter ID Number: 18001234
Enter the duration of stay (in days): 1
Enter the current month (1-12): 1
Enter Check-in Date (dd/mm/yyyy): 1/1/2025
Enter Check-out Date (dd/mm/yyyy): 2/1/2025
-----
*****
Dear Tahsin,
Your booking for room number 202
The total price for your stay of 1 days is: 1380
Thank you for choosing our hotel. We look forward to your stay.
*****
Your bill have been printed
```

Test case 6:

```
Enter Room number: 1

*****
*      Welcome to HOTEL Paradise      *
*****
* 1. Display Available Rooms      *
* 2. Add a Room                  *
* 3. Edit Room details           *
* 4. Room Booking                *
* 5. Edit Booking Details         *
* 6. Checkout                     *
* 7. Exit                         *
*****
Enter your choice: 6
Enter Room number: 202
You have been successfully checkout from Room 202!
```

Test case 7:

```
*****
*      Welcome to HOTEL Paradise      *
*****
*  1. Display Available Rooms      *
*  2. Add a Room                  *
*  3. Edit Room details           *
*  4. Room Booking                *
*  5. Edit Booking Details        *
*  6. Checkout                    *
*  7. Exit                        *
*****
Enter your choice: 7
Exiting the system. Goodbye!
Booked Users saved and displayed successfully.
-----
Room details saved and displayed successfully.
```

Test case 8:

```
*****
*      Welcome to HOTEL Paradise      *
*****
*  1. Display Available Rooms      *
*  2. Add a Room                  *
*  3. Edit Room details           *
*  4. Room Booking                *
*  5. Edit Booking Details        *
*  6. Checkout                    *
*  7. Exit                        *
*****
Enter your choice: 7
Exiting the system. Goodbye!
Booked Users saved and displayed successfully.
-----
Room details saved and displayed successfully.
```

# Conclusion

The Hotel Management System project successfully automates hotel operations, including room booking, check-in, and check-out processes. It ensures efficient management and provides a better user experience for both hotel staff and guests.

The project enhanced our understanding of C programming, modular design, file handling, user authentication, and practical application of data structures.

Working on this project as a team improved our collaboration skills, communication, and problem-solving abilities. We learned to divide tasks effectively and integrate our work seamlessly.

We mostly faced challenges with handling file operations and ensuring data persistence, managing user authentication and session handling, and thoroughly debugging and testing the program.

The opportunities for improvement include implementing a graphical user interface for better user experience, adding features like payment processing and advanced room search filters, and optimizing code for better performance and scalability.

This project provided valuable insights into software development and project management. It was a great learning experience, and we are proud of the results we achieved as a team.

# References

YouTube, ChatGPT, Teach Yourself C by Herbert Schildt