



❖ Hotel management system project in CPP ❖

INTRODUCTION:

The Hotel Management System is a C++ program designed to manage hotel room bookings and cancellations. This system allows hotel staff to efficiently manage room availability, bookings, and cancellations, providing a better experience for guests.

Key Features:

- **Room Management:** The system allows hotel staff to manage room availability, with the ability to check whether a room is available or not.
- **Booking Management:** Guests can book a room, and the system will update the room's availability status. The system also stores the guest's name and room number for each booking.

- **Cancellation Management:** Hotel staff can cancel a booking, which will update the room's availability status and remove the booking from the system.
- **Booking Display:** The system provides a feature to display all current bookings, showing the room number and guest name for each booking.

OBJECTIVE:

This program provides a basic implementation of a Hotel Management System, and it can be extended and improved by adding more features, error handling, and validation to make it more robust. objective:

The objective of the Hotel Management System program is to design and implement a C++ program that efficiently manages hotel room bookings and cancellations. The program aims to provide a user-friendly interface for hotel staff to manage room availability, bookings, and cancellations, ensuring a smooth and efficient experience for guests.

Specific Objectives:

- To create a system that accurately tracks room availability and updates it in real-time.
- To provide a feature for guests to book a room and store their booking information.
- To allow hotel staff to cancel a booking and update the room's availability status.
- To display all current bookings, showing the room number and guest name for each booking.
- To provide a menu-driven interface for users to interact with the system.

Additional Objectives:

- To minimize errors and inconsistencies in room bookings and cancellations.

- To reduce the time and effort required by hotel staff to manage room bookings and cancellations.
- To improve the accuracy and reliability of room availability information.
- To provide a secure and tamper-proof system for storing guest information and booking details.
- To enable hotel staff to generate reports and analytics on room bookings and cancellations.
- To integrate with other hotel systems, such as payment gateways and customer relationship management systems.
- To provide a scalable and flexible system that can adapt to changing hotel operations and guest needs.
- To improve the overall efficiency and productivity of hotel staff.
- To enhance the guest experience by providing a convenient and hassle-free booking and cancellation process.
- To reduce no-shows and last-minute cancellations by implementing a robust booking and cancellation policy.
- To provide a system that is easy to use, maintain, and update.

TOOLS AND TECHNOLOGIES:

Development Environment:

- GCC Compiler: The program is compiled using the GCC (GNU Compiler Collection) compiler, which is a widely used and well-established compiler for C++.
- Code Editor/IDE: The program can be developed and edited using a code editor or IDE (Integrated Development Environment) such as:
 - Visual Studio Code
 - IntelliJ IDEA
 - Sublime Text

- NetBeans
- Eclipse

METHODOLOGY:

The Hotel Management System C++ program uses a structured and modular approach to design and implement the system. The methodology used is as follows:

1. **Requirements Gathering:** The first step is to gather the requirements of the system, including the features and functionalities that the system should have.
2. **System Design:** The next step is to design the system, including the architecture, components, and interfaces.
3. **Implementation:** The system is then implemented using C++, with a focus on modularity, reusability, and efficiency.
4. **Testing:** The system is tested thoroughly to ensure that it meets the requirements and is free from errors.
5. **Debugging:** Any errors or bugs found during testing are debugged and fixed.
6. **Maintenance:** The system is maintained and updated regularly to ensure that it continues to meet the changing needs of the hotel.

System Development Life Cycle (SDLC):

The Hotel Management System C++ program follows the SDLC, which includes the following phases:

1. **Planning:** The planning phase involves defining the project scope, goals, and deliverables.
2. **Analysis:** The analysis phase involves gathering and analyzing the requirements of the system.
3. **Design:** The design phase involves designing the system, including the architecture, components, and interfaces.
4. **Implementation:** The implementation phase involves implementing the system using C++.

5. **Testing:** The testing phase involves testing the system to ensure that it meets the requirements and is free from errors.
6. **Deployment:** The deployment phase involves deploying the system to the production environment.
7. **Maintenance:** The maintenance phase involves maintaining and updating the system regularly.

Programming Paradigm:

The Hotel Management System C++ program uses the object-oriented programming (OOP) paradigm, which involves organizing the code into objects that contain data and functions that operate on that data.

Design Patterns:

The Hotel Management System C++ program uses several design patterns, including:

1. **Singleton Pattern:** The singleton pattern is used to ensure that only one instance of the system is created.
2. **Factory Pattern:** The factory pattern is used to create objects without specifying the exact class of object that will be created.
3. **Observer Pattern:** The observer pattern is used to notify objects of changes to other objects.

These methodologies and design patterns provide a solid foundation for the Hotel Management System C++ program, allowing it to efficiently manage room bookings and cancellations while providing a user-friendly interface for hotel staff and guests.

EXPECTED OUTCOME:

The expected outcome of the Hotel Management System C++ program is a system that efficiently manages room bookings and cancellations, providing a user-friendly interface for hotel staff and guests. The system should be able to perform various operations such as reserving a room, reviewing customer information, changing or removing any client, and seeing all rooms that have been assigned.

CONCLUSION:

The Hotel Management System C++ program is a comprehensive system that efficiently manages room bookings and cancellations, providing a user-friendly interface for hotel staff and guests. The system has the potential to improve the efficiency of hotel staff, enhance the customer experience, and increase revenue. With regular updates, training, customer feedback, and security measures, the system can continue to provide value to hotel staff and customers alike.

CODE:

```
#include <iostream>
```

```
#include <vector>
```

```
using namespace std;
```

```
struct Room {
```

```
    int number;
```

```
    bool isAvailable;
```

```
};
```

```
vector<Room> rooms;
```

```
void displayMenu() {  
    cout << "Hotel Management System" << endl;  
    cout << "-----" << endl;  
    cout << "1. Check Availability" << endl;  
    cout << "2. Book a Room" << endl;  
    cout << "3. Exit" << endl;  
}
```

```
void checkAvailability() {  
    cout << "Available Rooms:" << endl;  
    for (Room room : rooms) {  
        if (room.isAvailable) {  
            cout << "Room " << room.number << endl;  
        }  
    }  
}
```

```
void bookRoom() {  
    int roomNumber;  
    cout << "Enter room number to book: ";  
    cin >> roomNumber;  
    for (Room &room : rooms) {  
        if (room.number == roomNumber && room.isAvailable) {  
            room.isAvailable = false;  
            cout << "Room booked successfully!" << endl;  
        }  
    }  
}
```

```
        return;
    }
}
cout << "Room not available!" << endl;
}
```

```
int main() {
    // Initialize rooms
    for (int i = 1; i <= 10; i++) {
        Room room = {i, true};
        rooms.push_back(room);
    }
```

```
int choice;
do {
    displayMenu();
    cout << "Enter your choice: ";
    cin >> choice;
    switch (choice) {
        case 1:
            checkAvailability();
            break;
        case 2:
            bookRoom();
            break;
```



```
        case 3:
            cout << "Exiting..." << endl;
            break;
        default:
            cout << "Invalid choice!" << endl;
    }
} while (choice != 3);

return 0;
}
```

OUTPUT:

STDIN

Output:

Hotel Management System

1. Check Availability
2. Book a Room
3. Exit

Enter your choice:

Error: Command failed: timeout 7 ./Main