A

# PROJECT REPORT

ON

# “Hotel Booking System”

SUBMITTED BY:

**Mr. Ruturaj Sonawane (2124UCEM1105)**

SUBJECT:

**CORE C++ PROGRAMMING**

Under the guidance of **Miss. ISHWARI TIRSE.**

**Department of Computer Science and Engineering**

**Sanjivani Rural Education Society's**

**SANJIVANI UNIVERSITY**

**KOPARGAON-423601, DIST: AHMEDNAGAR 2024-2025**

# INDEX

| SR. NO | CONTENT | PAGE NO. |
|---|---|---|
| 1. | INTRODUCTION | 3 |
| 2. | CODE | 4 |
| 3. | OUTPUT | 13 |
| 4. | CONCLUSION | 16 |

# INTRODUCTION

A **Hotel Booking System** is a software application that allows customers to book hotel rooms, view available rooms, cancel bookings, and check reservation details. The system manages rooms, customer information, and booking data to provide seamless functionality for both customers and hotel administrators. This project simulates the operations of a hotel, with features such as room availability checks, booking management, and viewing current bookings.

In this project, we focus on creating a **console-based** hotel booking system in **C++**. The system allows users to interact with the hotel's room inventory, book rooms, view bookings, and cancel bookings, all while maintaining the state of the rooms and bookings in memory.

Key features of the system:

1. **View Available Rooms**: Displays rooms that are available for booking.
2. **Book a Room**: Allows users to book a room if it's available.
3. **Cancel a Booking**: Allows users to cancel an existing booking, making the room available again.
4. **View Bookings**: Displays a list of all bookings with customer details and booking dates.

# CODE

```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
using namespace std;

class Room {
private:
    int roomNumber;
    string roomType;
    float price;
    bool isAvailable;

public:
    // Constructor
    Room(int rNum, string rType, float rPrice)
        : roomNumber(rNum), roomType(rType), price(rPrice), isAvailable(true) {}

    // Getter and Setter methods
    int getRoomNumber() const { return roomNumber; }
    string getRoomType() const { return roomType; }
    float getPrice() const { return price; }
    bool getAvailability() const { return isAvailable; }
    void setAvailability(bool availability) { isAvailable = availability; }

    // Display room details
    void displayRoom() const {
        cout << "Room Number: " << roomNumber
             << ", Type: " << roomType
             << ", Price: $" << price
             << ", Status: " << (isAvailable ? "Available" : "Booked") <<
endl;
    }
};

class Booking {
private:
    string customerName;
    string contactNumber;
    int roomNumber;
    string checkInDate;
    string checkOutDate;

public:
    // Constructor
    Booking(string name, string contact, int roomNum, string checkIn,
string checkOut)
```

```
        : customerName(name), contactNumber(contact),
roomNumber(roomNum),
          checkInDate(checkIn), checkOutDate(checkOut) {}

    // Display booking details
    void displayBooking() const {
        cout << "Customer Name: " << customerName
             << ", Contact: " << contactNumber
             << ", Room Number: " << roomNumber
             << ", Check-In: " << checkInDate
             << ", Check-Out: " << checkOutDate << endl;
    }

    int getRoomNumber() const { return roomNumber; }
};

class HotelManagementSystem {
private:
    vector<Room> rooms;
    vector<Booking> bookings;

public:
    // Load rooms into the system
    void loadRooms() {
        rooms.push_back(Room(101, "Single", 100.0f));
        rooms.push_back(Room(102, "Double", 150.0f));
        rooms.push_back(Room(103, "Suite", 250.0f));
    }

    // Display available rooms
    void displayAvailableRooms() {
        bool available = false;
        for (const auto& room : rooms) {
            if (room.getAvailability()) {
                room.displayRoom();
                available = true;
            }
        }
        if (!available) {
            cout << "No rooms are available." << endl;
        }
    }

    // Book a room
    void bookRoom(string name, string contact, int roomNumber,
string checkIn, string checkOut) {
        for (auto& room : rooms) {
            if (room.getRoomNumber() == roomNumber &&
room.getAvailability()) {
                room.setAvailability(false);
                bookings.push_back(Booking(name, contact, roomNumber, checkIn, checkOut));
                cout << "Room booked successfully!" << endl;
                return;
            }
        }
        cout << "Sorry, this room is either unavailable or does not exist." << endl;
    }

    // View all bookings
    void viewBookings() const {
        if (bookings.empty()) {
            cout << "No bookings yet." << endl;
            return;
        }
        for (const auto& booking : bookings) {
            booking.displayBooking();
        }
    }

    // Cancel a booking
    void cancelBooking(int roomNumber) {
        for (auto it = bookings.begin(); it != bookings.end(); ++it) {
            if (it->getRoomNumber() == roomNumber) {
                bookings.erase(it);
                for (auto& room : rooms) {
                    if (room.getRoomNumber() == roomNumber) {
                        room.setAvailability(true);
                    }
                }
                cout << "Booking canceled successfully!" << endl;
                return;
            }
        }
        cout << "No booking found for this room." << endl;
    }
};

int main() {
    HotelManagementSystem hms;
    hms.loadRooms();  // Load rooms into the system

    int choice;
    do {
        cout << "\nHotel Booking System\n";
        cout << "1. View Available Rooms\n";
        cout << "2. Book a Room\n";
        cout << "3. View Bookings\n";
        cout << "4. Cancel Booking\n";
        cout << "5. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case 1: {
                cout << "Available Rooms:\n";
                hms.displayAvailableRooms();
                break;
            }
            case 2: {
                string name, contact, checkIn, checkOut;
                int roomNumber;
                cout << "Enter your name: ";
                cin.ignore();  // To clear the buffer
                getline(cin, name);
                cout << "Enter your contact number: ";
                getline(cin, contact);
                cout << "Enter room number to book: ";
                cin >> roomNumber;
                cout << "Enter check-in date (dd/mm/yyyy): ";
                cin.ignore();  // To clear the buffer
                getline(cin, checkIn);
                cout << "Enter check-out date (dd/mm/yyyy): ";
                getline(cin, checkOut);
```

```
                hms.bookRoom(name, contact, roomNumber, checkIn, checkOut);
                break;
            }
            case 3: {
                cout << "Booking Details:\n";
                hms.viewBookings();
                break;
            }
            case 4: {
                int roomNumber;
                cout << "Enter room number to cancel booking: ";
                cin >> roomNumber;
                hms.cancelBooking(roomNumber);
                break;
            }
            case 5:
                cout << "Thank you for using the Hotel Booking System!\n";
                break;
            default:
                cout << "Invalid choice. Please try again.\n";
        }
    } while (choice != 5);
    return 0;
}
```

# OUTPUT

```
Hotel Booking System
1. View Available Rooms
2. Book a Room
3. View Bookings
4. Cancel Booking
5. Exit
Enter your choice: 1
Available Rooms:
Room Number: 101, Type: Single, Price: $100, Status: Available
Room Number: 102, Type: Double, Price: $150, Status: Available
Room Number: 103, Type: Suite, Price: $250, Status: Available

Hotel Booking System
1. View Available Rooms
2. Book a Room
3. View Bookings
4. Cancel Booking
5. Exit
Enter your choice: 2
Enter your name: John Doe
Enter your contact number: 123-456-7890
Enter room number to book: 102
Enter check-in date (dd/mm/yyyy): 10/10/2024
Enter check-out date (dd/mm/yyyy): 15/10/2024
Room booked successfully!

Hotel Booking System
1. View Available Rooms
2. Book a Room
3. View Bookings
4. Cancel Booking
5. Exit
Enter your choice: 3
Booking Details:
Customer Name: John Doe, Contact: 123-456-7890, Room Number: 102, Check-In: 10/10/2024, Check-Out: 15/10/2024

Hotel Booking System
1. View Available Rooms
2. Book a Room
3. View Bookings
4. Cancel Booking
5. Exit
Enter your choice: 4
Enter room number to cancel booking: 102

Booking canceled successfully!
```

# CONCLUSION

The **Hotel Booking System** is a simple yet effective demonstration of a real-world application using C++. It covers key concepts like **object-oriented programming (OOP)**, **file handling**, and **user input/output**. In this project, we've created a basic hotel management system that can handle room bookings, cancellations, and viewing of bookings.

While this is a basic implementation, the system can be extended in many ways, including:

- Adding a **database** or **file system** for persistent storage.
- Implementing **payment processing**.
- Adding **room availability checks** for different dates.
- Creating a **Graphical User Interface (GUI)**.

This project provides a solid foundation for understanding how to design and implement more complex systems, making it an excellent learning experience for C++ programming.