# TITLE : Hotel Reservation System (Java)

## Abstract

This project presents a simple and efficient Hotel Reservation System developed in Java. It enables users to view available rooms, book rooms, and cancel reservations easily. The system uses Object-Oriented Programming principles and dynamic data handling to automate the room management process. It demonstrates how arrays and Java collections can be used to manage real-world operations effectively.

## Introduction

The Hotel Reservation System aims to automate the manual process of managing hotel room bookings. Using this program, hotel staff can efficiently view room availability, confirm bookings, and process cancellations. The project focuses on simplicity and clarity while implementing key Java concepts like classes, objects, and ArrayLists.

## Existing System

The existing manual booking process involves maintaining records on paper, which is prone to human error and inefficiency. It requires extra time to check room availability, update records, and handle cancellations manually.

## Proposed System

The proposed Hotel Reservation System automates the room booking process. It provides real-time availability updates, prevents double bookings, and simplifies cancellations. The system uses Java’s object-oriented structure to ensure data consistency and ease of use.

## Software Requirements

• Operating System: Windows / Linux  
• Programming Language: Java (JDK 8 or higher)  
• IDE: Eclipse / IntelliJ IDEA / BlueJ / VS Code  
• Database: Not required (in-memory data storage)

## Hardware Requirements

• Processor: 1 GHz or higher  
• RAM: 512 MB or higher  
• Hard Disk: 10 MB free space  
• Input Device: Keyboard  
• Output Device: Monitor

## Full Java Code

import java.util.\*;  
  
class Room {  
 int roomNo;  
 boolean isBooked;  
 String customerName;  
  
 Room(int roomNo) {  
 this.roomNo = roomNo;  
 this.isBooked = false;  
 this.customerName = "";  
 }  
}  
  
public class HotelReservation {  
 static ArrayList<Room> rooms = new ArrayList<>();  
  
 public static void main(String[] args) {  
 for (int i = 1; i <= 5; i++) rooms.add(new Room(i));  
 Scanner sc = new Scanner(System.in);  
 int choice;  
  
 do {  
 System.out.println("\n--- HOTEL RESERVATION SYSTEM ---");  
 System.out.println("1. View Available Rooms");  
 System.out.println("2. Book Room");  
 System.out.println("3. Cancel Booking");  
 System.out.println("4. Show All Rooms");  
 System.out.println("5. Exit");  
 System.out.print("Enter choice: ");  
 choice = sc.nextInt();  
  
 switch (choice) {  
 case 1 -> viewAvailable();  
 case 2 -> bookRoom(sc);  
 case 3 -> cancelRoom(sc);  
 case 4 -> showAll();  
 case 5 -> System.out.println("Thank you for using our system!");  
 default -> System.out.println("Invalid choice!");  
 }  
 } while (choice != 5);  
 }  
  
 static void viewAvailable() {  
 System.out.println("\nAvailable Rooms:");  
 for (Room r : rooms)  
 if (!r.isBooked) System.out.println("Room " + r.roomNo);  
 }  
  
 static void bookRoom(Scanner sc) {  
 System.out.print("Enter room number to book: ");  
 int num = sc.nextInt();  
 for (Room r : rooms) {  
 if (r.roomNo == num && !r.isBooked) {  
 System.out.print("Enter your name: ");  
 r.customerName = sc.next();  
 r.isBooked = true;  
 System.out.println("Room booked successfully!");  
 return;  
 }  
 }  
 System.out.println("Room not available or already booked!");  
 }  
  
 static void cancelRoom(Scanner sc) {  
 System.out.print("Enter room number to cancel: ");  
 int num = sc.nextInt();  
 for (Room r : rooms) {  
 if (r.roomNo == num && r.isBooked) {  
 r.isBooked = false;  
 r.customerName = "";  
 System.out.println("Booking cancelled successfully!");  
 return;  
 }  
 }  
 System.out.println("No booking found for that room!");  
 }  
  
 static void showAll() {  
 System.out.println("\nRoom Status:");  
 for (Room r : rooms) {  
 System.out.println("Room " + r.roomNo + " - " +  
 (r.isBooked ? "Booked by " + r.customerName : "Available"));  
 }  
 }  
}

## Sample Output

--- HOTEL RESERVATION SYSTEM ---  
1. View Available Rooms  
2. Book Room  
3. Cancel Booking  
4. Show All Rooms  
5. Exit  
  
Enter choice: 1  
Available Rooms:  
Room 1  
Room 2  
Room 3  
Room 4  
Room 5  
  
Enter choice: 2  
Enter room number to book: 2  
Enter your name: Riya  
Room booked successfully!  
  
Enter choice: 4  
Room 1 - Available  
Room 2 - Booked by Riya  
Room 3 - Available  
Room 4 - Available  
Room 5 - Available

## Conclusion

The Hotel Reservation System simplifies the process of managing hotel bookings. Using Java’s OOP principles, the program provides a reliable way to handle reservations, cancellations, and room availability. The project showcases how real-world systems can be implemented efficiently using simple programming logic.