# MINI PROJECT

## HOTEL MANAGEMENT SYSYEM

**Aim:**

The aim of this project is to develop a Hotel Management System using Java to efficiently store, manage, and display donor information.

**Algorithm:**

1) **Start**
Initialize the list of rooms and an empty booking list. Set the booking ID counter to 1.

2) **Display Menu**
Show the user a menu with options: View Available Rooms, Book a Room, Cancel Booking, View Booking Details, or Exit.

3) **View Available Rooms**
If the user selects this option, display all rooms marked as "available."

4) **Book a Room**

- Prompt the user for the room number, name, and phone number.
- Check if the selected room is available.
- If available, mark the room as booked, create a booking record, and increment the booking ID counter.
- Display the booking confirmation with the booking ID.

5) **Cancel Booking**

- Prompt the user for the booking ID.
- If the booking exists, mark the associated room as available and remove the booking record.
- Display a cancellation .

6) **View Booking Details**

- Prompt the user for the booking ID.
- If the booking exists, display the booking details.
- If the booking ID is invalid, display an error message.

**7) Stop**

**Program:**

```
import java.util.*;

public class HotelBookingSystem {
    static Scanner scanner = new Scanner(System.in);
    static List<Room> rooms = new ArrayList<>();
    static Map<Integer, Booking> bookings = new HashMap<>();
    static int bookingIdCounter = 1;

    public static void main(String[] args) {
        // Initialize some rooms
        initializeRooms();

        while (true) {
            System.out.println("\n--- Hotel Booking System ---");
            System.out.println("1. View Available Rooms");
            System.out.println("2. Book a Room");
            System.out.println("3. Cancel Booking");
            System.out.println("4. View Booking Details");
            System.out.println("5. Exit");
            System.out.print("Choose an option: ");

            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume newline

            switch (choice) {
                case 1 -> viewAvailableRooms();
                case 2 -> bookRoom();
                case 3 -> cancelBooking();
```

```
            case 4 -> viewBookingDetails();
            case 5 -> {
                System.out.println("Thank you for using the Hotel Booking System!");
                return;
            }
            default -> System.out.println("Invalid choice. Please try again.");
        }
    }
}

static void initializeRooms() {
    rooms.add(new Room(101, "Single", 1000));
    rooms.add(new Room(102, "Double", 2000));
    rooms.add(new Room(103, "Suite", 5000));
}

static void viewAvailableRooms() {
    System.out.println("\n--- Available Rooms ---");
    for (Room room : rooms) {
        if (room.isAvailable()) {
            System.out.println(room);
        }
    }
}

static void bookRoom() {
    System.out.print("\nEnter Room Number to book: ");
    int roomNumber = scanner.nextInt();
    scanner.nextLine();
```

```
    Room room = findRoomByNumber(roomNumber);
    if (room == null || !room.isAvailable()) {
        System.out.println("Room is not available!");
        return;
    }

    System.out.print("Enter your name: ");
    String name = scanner.nextLine();
    System.out.print("Enter your phone number: ");
    String phone = scanner.nextLine();

    Booking booking = new Booking(bookingIdCounter++, room, name, phone);
    bookings.put(booking.getBookingId(), booking);
    room.setAvailable(false);

    System.out.println("Room booked successfully! Booking ID: " + booking.getBookingId());
}

static void cancelBooking() {
    System.out.print("\nEnter Booking ID to cancel: ");
    int bookingId = scanner.nextInt();

    Booking booking = bookings.remove(bookingId);
    if (booking == null) {
        System.out.println("Invalid Booking ID!");
        return;
    }
```

```
        booking.getRoom().setAvailable(true);
        System.out.println("Booking cancelled successfully.");
    }

    static void viewBookingDetails() {
        System.out.print("\nEnter Booking ID: ");
        int bookingId = scanner.nextInt();

        Booking booking = bookings.get(bookingId);
        if (booking == null) {
            System.out.println("Invalid Booking ID!");
            return;
        }

        System.out.println("\n--- Booking Details ---");
        System.out.println(booking);
    }

    static Room findRoomByNumber(int roomNumber) {
        for (Room room : rooms) {
            if (room.getNumber() == roomNumber) {
                return room;
            }
        }
        return null;
    }
}

class Room {
```

```
private int number;
private String type;
private double price;
private boolean available;

public Room(int number, String type, double price) {
    this.number = number;
    this.type = type;
    this.price = price;
    this.available = true;
}

public int getNumber() {
    return number;
}

public String getType() {
    return type;
}

public double getPrice() {
    return price;
}

public boolean isAvailable() {
    return available;
}

public void setAvailable(boolean available) {
```

```
        this.available = available;
    }

    @Override
    public String toString() {
        return "Room{" +
                "number=" + number +
                ", type='" + type + '\'' +
                ", price=" + price +
                ", available=" + available +
                '}';
    }
}

class Booking {
    private int bookingId;
    private Room room;
    private String customerName;
    private String customerPhone;

    public Booking(int bookingId, Room room, String customerName, String customerPhone) {
        this.bookingId = bookingId;
        this.room = room;
        this.customerName = customerName;
        this.customerPhone = customerPhone;
    }

    public int getBookingId() {
        return bookingId;
```

```
    }

    public Room getRoom() {
        return room;
    }

    @Override
    public String toString() {
        return "Booking{" +
                "bookingId=" + bookingId +
                ", room=" + room +
                ", customerName='" + customerName + '\'' +
                ", customerPhone='" + customerPhone + '\'' +
                '}';
    }
}
```

**Output:**

```
--- Hotel Booking System ---
1. View Available Rooms
2. Book a Room
3. Cancel Booking
4. View Booking Details
5. Exit
Choose an option: 1
```

```
--- Available Rooms ---
Room{number=101, type='Single', price=1000.0, available=true}
Room{number=102, type='Double', price=2000.0, available=true}
Room{number=103, type='Suite', price=5000.0, available=true}
```

```
Enter Room Number to book: 101
Enter your name: John Doe
Enter your phone number: 1234567890
Room booked successfully! Booking ID: 1
```

**Result:**

The algorithm outlines a hotel booking system where users can view available rooms, book or cancel bookings, and check booking details, ending with program termination.