**Hotel Booking System**

The **Hotel Booking System** is a comprehensive web application built using **Spring Boot** that allows users to search for hotels, view available rooms, and make bookings. It supports user authentication with role-based access for **users** and **administrators**, and provides a complete booking flow from room selection to booking confirmation. This project simulates the functionality of a real-world hotel booking portal like **Booking.com** or **OYO**, but on a smaller scale suitable for learning and portfolio demonstration.

**What to do?**

**User Module**

Manages user registration and login functionality. Handles storing user credentials securely with hashed passwords, generating JWT tokens upon successful login, and differentiating between regular users and admins using role-based access control.

**Hotel Module**

Allows admins to create, update, view, and delete hotel records. Each hotel includes information like name, city, address, description, and rating. Only admins have access to manage hotel data, while users can view it.

**Booking Module**

Lets users book available rooms for specific check-in and check-out dates. It checks for date conflicts and updates room availability. Users can also view their booking history and cancel bookings. Booking status is tracked (e.g., CONFIRMED, CANCELLED).

**Search Module**

Enables users to search for hotels based on location and view rooms available for a selected date range. Filters out rooms that are already booked during the specified period. Returns hotel and room details suitable for booking.

**Payment Module**

Simulates or integrates payment processing. Records payment details such as booking ID, amount, and payment status (SUCCESS/FAILED). Used after successful room booking as the final step before confirmation.

**Admin Dashboard Module**

Provides a consolidated view for the admin to monitor system activity such as all registered users, all bookings, hotel statistics, and payment summaries. Can be presented as REST APIs or a simple dashboard UI.

**Tech Stack:**

**Backend:** Spring Boot,JPA, REST- for building the server and APIs.

**Database:** MySQL/MongoDB - to store user, product, and order data.

**Frontend:** Thymeleaf (server-rendered pages) or React (interactive client app).

**Security:** Spring Security - for user login and role-based access.

**Razor Pay/Stripe:** For making payment after the booking

**Deployment:** Render/Railway/Netlify

**N.B:** Use JWT if you use React; for Thymeleaf, JWT is not required.

**Testing:**

* Write JUnit 5 and Mockito unit tests for backend services.(Controller, Service, Repository)
* Perform end-to-end testing for user workflows

How do I submit my work?

**For Code Submission:**

**Code:**

* Push your complete project code to GitHub
* Organize your code into /backend and /frontend folders (if using React).

**Deployment:**

* Option 1: Deploy Backend on Render/Railway/AWS EC2(Mandatory)
* Option 2: Frontend on Vercel/Netlify(Mandatory)
* Option 3: Record a 1-2 min walkthrough video and upload to Google Drive (public access) (Optional)