

Of course. Here is a comprehensive project plan document for **HotelWay** that combines all your choices and requirements into a single, actionable roadmap.

Project Plan: HotelWay

1.0 Project Overview

This document outlines the development plan for **HotelWay**, a full-stack web application for hotel management. The platform will provide distinct portals for customers, hotel owners, and administrators to manage bookings, properties, and user activities. The goal is to build a modern, responsive, and feature-rich application based on the provided UI mockups and feature requirements.

2.0 Technology Stack

- **Front-End:** React
- **Styling:** Tailwind CSS
- **Back-End:** Java with Spring Boot
- **Database:** MySQL
- **Animations:** Framer Motion (or a similar library)
- **Data Visualization:** Recharts or Chart.js

3.0 Development Phases & Timeline

This project is broken down into manageable phases. The timeline is a suggestion and can be adjusted based on your pace.

Phase 1: Foundation & Setup (Week 1)

- **Goal:** Prepare the complete development environment.
- **Tasks:**
 1. Install necessary software: Java JDK 17+, IntelliJ IDEA, Node.js, VS Code, and MySQL Server.
 2. Create the hotelway database in MySQL and execute the SQL script to create all the tables defined in Section 4.0.
 3. Generate the Spring Boot project using Spring Initializr with Spring Web, Spring Data JPA, and MySQL Driver dependencies.
 4. Initialize the React project using `npx create-react-app hotelway-frontend` and set up Tailwind CSS.

Phase 2: Back-End Core - Authentication & Hotels (Weeks 2-4)

- **Goal:** Build the fundamental API endpoints.
- **Tasks:**
 1. Implement user registration and login (Authentication) functionality, including security and OTP logic.
 2. Build the API endpoints for managing hotels and rooms (Create, Read, Update, Delete operations).
 3. Develop the admin logic for hotel verification.

Phase 3: Front-End UI - Customer Portal (Weeks 5-7)

- **Goal:** Build the static user-facing pages.
- **Tasks:**

1. Develop reusable React components (Navbar, Buttons, Cards, Forms).
2. Build the static versions of the Homepage and the Customer Hotel View page based on the provided images.
3. Set up client-side routing using react-router-dom to navigate between pages.

Phase 4: Full-Stack Integration - Bookings (Weeks 8-9)

- **Goal:** Connect the front-end and back-end to enable core functionality.
- **Tasks:**
 1. Connect the React login/signup forms to the back-end authentication API.
 2. Fetch and display the list of trending destinations and hotels on the homepage.
 3. Implement the complete booking flow: check availability, select dates, and create a booking.

Phase 5: Dashboards - Hotel & Admin Portals (Weeks 10-12)

- **Goal:** Build the management interfaces for hotel owners and administrators.
- **Tasks:**
 1. Develop the Hotel Portal UI, including charts for revenue and occupancy.
 2. Implement features for hotel owners to manage room status, view bookings, and see guest reviews.
 3. Build the Admin Dashboard UI for managing users, hotels, and complaints.

Phase 6: Advanced Features & Polishing (Weeks 13-14)

- **Goal:** Add the final features and refine the user experience.
- **Tasks:**
 1. Implement the Community Blog system, including the submission form for users and the approval queue for admins.
 2. Integrate smooth animations and transitions.
 3. Perform thorough testing across different browsers and devices to ensure responsiveness.
 4. Refine the UI to ensure all components are well-placed and visually appealing.

4.0 MySQL Database Schema

- **users:**
 - id (Primary Key), phone_number, name, gender, password, role
- **hotels:**
 - id (PK), name, location, owner_id (Foreign Key to users), status
- **rooms:**
 - id (PK), hotel_id (FK to hotels), room_number, status, price
- **bookings:**
 - id (PK), user_id (FK to users), room_id (FK to rooms), check_in_date, check_out_date
- **reviews:**
 - id (PK), booking_id (FK to bookings), rating, comment
- **complaints:**
 - id (PK), user_id (FK to users), details, status
- **blog_posts:**
 - id (PK), user_id (FK to users), title, content, status, created_at

5.0 API Endpoint Plan (Spring Boot)

A preliminary list of API routes to be built:

- **Authentication (/api/auth):** POST /signup, POST /login
- **Hotels (/api/hotels):** GET /, GET /{hotelId}, POST / (for registration), PUT /{hotelId}
- **Rooms (/api/rooms):** GET /hotel/{hotelId}, POST /, PUT /{roomId}
- **Bookings (/api/bookings):** POST /, GET /user/{userId}
- **Admin (/api/admin):** GET /hotels/pending, PUT /hotels/approve/{hotelId}, GET /users
- **Blog (/api/blog):** GET /, POST /, GET /admin/pending, PUT /admin/approve/{postId}

6.0 Key Features & Milestones

This plan incorporates all requested features into a cohesive system.

- **Authentication:** Secure sign-up and login for all users.
- **Customer Portal:** Allows users to browse and book destinations.
- **Hotel Portal:** A comprehensive dashboard for hotel owners to manage their property, including:
 - Property and employee management.
 - Billing and POS integration.
 - Housekeeping and guest review management.
- **Admin Dashboard:** A central hub for admins to oversee the entire platform, including user, hotel, and complaint management.
- **Community Blog:** A space for verified guests to share their experiences, moderated by admins.