WEB TECHNOLOGY PROJECT

1. Problem Description and Key Requirements

Problem Description

The goal is to design and implement a **hotel booking management system** that allows users to search for hotels, book rooms, and make secure payments. The system must handle user authentication, role-based access control, booking lifecycle management, and payment verification. Hotel managers should be able to manage hotel details, rooms, and monitor reservations efficiently, while users need a smooth experience from booking to payment confirmation.

Key Business Requirements

- **User Management:**
    - Secure user registration, login, and password recovery.
    - Two-factor authentication for enhanced security.
    - Role-based access control (e.g., admin, hotel manager, customer).
- **Hotel & Room Management:**
    - Add, update, and delete hotel and room records.
    - Manage room availability, pricing, and capacity.
- **Booking Management:**
    - Search for available rooms by date and capacity.
    - Reserve, update, or cancel bookings.
    - Track booking status (pending, confirmed, canceled).
- **Payment Processing:**
    - Record payments with different methods.
    - Validate payment confirmations and update booking status.

- **Audit and Reporting:**
    - Log key events (bookings, payments, user changes).
    - Generate occupancy and revenue reports.

Quality Attributes

- **Security:** Data encryption, role-based access, two-factor authentication.
- **Performance:** Fast room search and booking confirmation.
- **Scalability:** Support many concurrent users and bookings.
- **Reliability:** Ensure booking and payment data consistency.
- **Usability:** Intuitive user interface for customers and admins.
- **Maintainability:** Modular design for easy updates.

2. Domain & Data Modeling

a. ER-Style Diagram

- **Users** (id, username, email, password, fullName, phoneNumber, enabled, twoFactorSecret, …)
- **Roles** (name)
- **User_Roles** (user_id → Users.id, role → Roles.name)
- **Hotels** (id, name, region, address, description, rating, created_at, updated_at)
- **Rooms** (id, hotel_id → Hotels.id, roomNumber, roomType, price, capacity, status)
- **Bookings** (id, user_id → Users.id, room_id → Rooms.id, checkInDate, checkOutDate, guests, status, payment_id → Payments.id)

- **Payments** (id, booking_id → Bookings.id, amountPaid, paymentMethod, transactionId, paymentStatus, confirmed)

b. Principal Domain Concepts

- **Actors:**
    - *Customer:* browses hotels, books rooms, makes payments.
    - *Hotel Manager:* manages hotels, rooms, and bookings.
    - *Admin:* manages users, roles, and audits.
- **Processes:**
    - Authentication and authorization.
    - Searching for rooms by availability and filters.
    - Booking workflow (create → confirm → complete/cancel).
    - Payment processing and validation.
    - Notifications for booking updates and confirmations.
- **Data Objects:**
    - User profile and credentials.
    - Hotel and room data.
    - Booking records and payment transactions.

c. Refined Conceptual → Technology-Oriented Schema

- **Primary Keys:**
    - Users.id, Roles.name, Hotels.id, Rooms.id, Bookings.id, Payments.id.
- **Foreign Keys & Constraints:**
    - User_Roles.user_id → Users.id (FK)
    - Rooms.hotel_id → Hotels.id (FK)
    - Bookings.user_id → Users.id (FK)
    - Bookings.room_id → Rooms.id (FK)
    - Bookings.payment_id → Payments.id (FK, optional if payment done later)
    - Payments.booking_id → Bookings.id (FK, unique constraint to ensure 1-to-1 link)
- **Indexes:**
    - On Users.email and Users.username (unique).
    - On Hotels.name and Hotels.region (for search).
    - On Bookings.status and Bookings.checkInDate (for fast filtering).

Link to repo: https://github.com/UMURERWLISAORNELLA/hotel_booking