HOTEL MANAGEMENT SYSTEM

EXPERIMENT-5

1) Introduction

Purpose

The HMS is meant to automate all the operations in a hotel, including room booking, check-in/check-out, billing, and customer management. The system will improve efficiency, reduce human errors, and improve customer satisfaction.

Scope

The HMS aims to make hotel management simple by combining room management, reservations, billing, and staff administration. It will ensure proper handling of check-ins and check-outs and significantly reduce the workload undertaken by individuals in such hotels. Reporting tools for analytics and decision-making make it suitable for hotels of any size.

Definition

- **HMS** - Hotel Management System
- **GUI** - Graphical User Interface
- **DBMS** - Database Management System
- **API** - Application Programming Interface

2) Functional Requirements

- **Room Management** – Add, update, and delete room details.
- **Reservation System** – Check room availability, book rooms, and modify bookings.
- **Billing & Payment** – Generate bills, accept payments, and issue invoices.
- **Customer Management** – Store and manage customer records.
- **User Authentication & Role Management** – Implement login with different roles (admin, receptionist, staff, customer).
- **Inventory Management** – Track hotel supplies, food, and housekeeping essentials.
- **Staff Management** – Maintain employee details, shift schedules, and payroll processing.

- **Guest Feedback System** – Allow customers to provide ratings and reviews.
- **Notification System** – Send email/SMS reminders for booking confirmation, check-in/out reminders, and payment receipts.

3) Design Requirements

The system should have:

- **Responsive design** with intuitive navigation.
- **Smooth booking, check-in, and payment workflows.**
- **Normalized structure** to allow efficient retrieval of data.
- **Microservices architecture** to handle large traffic.
- **Cloud Deployment** – Consider AWS, Azure, or Google Cloud for hosting.
- **Third-Party Integrations** – Integrate with payment gateways (PayPal, Stripe) and travel agency APIs (Booking.com, Expedia).

4) Performance Requirements

- The system should process room bookings in less than **2 seconds**.
- It should support **at least 500 concurrent users** without performance degradation.
- Payment transactions should be completed within **5 seconds** to ensure a smooth user experience.

5) Security Requirements

- **Data Encryption** – All sensitive data, including payment and customer details, must be encrypted using **AES-256**.
- **Authentication & Authorization** – Implement **multi-factor authentication (MFA)** for admins and role-based access control (RBAC).
- **Audit Logs** – Maintain **detailed logs** of all transactions, modifications, and login activities.
- **Backup & Recovery** – The system should perform **automatic database backups** every **24 hours** with quick recovery options.

6) Usability Requirements

- The system should have a **user-friendly interface** with minimal training required.

- The design should be **accessible** for users with disabilities (WCAG compliance).
- **Multilingual Support** should be available for international users.

7) Hardware Requirements

- **Server:** Minimum **8-core processor**, **16GB RAM**, and **500GB SSD storage**.
- **Client Devices:** Compatible with desktops, tablets, and mobile devices.
- **Internet Connection:** Minimum **10 Mbps speed** for smooth cloud-based operation.

8) Software Requirements

- **Frontend Technologies:** HTML, CSS, JavaScript (React/Angular/Vue).
- **Backend Technologies:** Node.js, Python (Django/Flask), or Java (Spring Boot).
- **Database:** MySQL / SQLite for storing transaction data.
- **Operating System:** Windows or Linux.
- **Security Measures:** HTTPS, Role-Based Access Control, Data Encryption for secure transactions.

9) Legal and Compliance Requirements

- The system must comply with **GDPR** (for data protection) and **PCI-DSS** (for secure payments).
- The software license should clearly define usage policies for hotels.
- Customer data retention policies must comply with **local regulations**.

10) External Interface Requirements

10.1 User Interfaces

- The system should have a **web-based GUI** that is intuitive and responsive for both desktop and mobile users.
- The **login screen** must support multiple user roles (Admin, Receptionist, Staff, Customer).
- A **dashboard** should display key statistics like total bookings, revenue, and occupancy rate.

10.2 Hardware Interfaces

- The system should support barcode or **RFID scanners** for quick check-in and check-out.
- It should be compatible with **POS (Point-of-Sale) terminals** for restaurant and spa billing.

10.3 Software Interfaces

- Integration with **third-party payment gateways** (PayPal, Stripe, Razorpay).
- Connection to **email and SMS services** (Twilio, SendGrid) for notifications.

10.4 Communication Interfaces

- The system should use **RESTful APIs** for data exchange between frontend and backend.
- Secure transactions should be conducted over **HTTPS with SSL/TLS encryption**.