Experiment-1

1. **Introduction**

* **Purpose**

The **Hotel Management System (HMS)** is designed to automate the operations of a hotel, including room booking, check-in/check-out, billing, and customer management. The system will enhance efficiency, reduce human errors, and improve customer satisfaction.

* **Scope**

The **Hotel Management System (HMS)** aims to streamline hotel operations by integrating room management, reservations, billing, and staff administration into a single platform. It ensures efficient handling of guest check-ins and check-outs, reduces manual workload, and enhances customer experience. Additionally, it provides reporting tools for analytics and decision-making, making it suitable for hotels of all sizes.

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

1. **Functional Requirements**

* **User Authentication** – Secure login for different roles.
* **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.

1. **Design Requirements**

The system should have:

* Responsive design with intuitive navigation
* Smooth booking, check-in, and payment workflows
* Normalized structure for efficient data retrieval
* Microservices architecture for handling high traffic

1. **Software Requirements**

* For frontend, HTML, CSS, JavaScript (React/Angular/Vue) will be used. And for backend, Node.js, Python (Django/Flask), or Java (Spring Boot) can be used.
* DBMS is needed to store the data of previous transactions done. Database can be made by MySQL / SQLite.
* As for operating system, Windows or Linux can be used.
* **Security Measures:** HTTPS, Role-Based Access Control, Data Encryption. These will help in secure transactions.