Experiment-1

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 reduces 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

1. **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**

1. **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

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.