Problem Statement:

The challenge is to design and develop a comprehensive hotel management system that caters to the needs of all stakeholders, including receptionists, cashiers, accounts departments, customers, managers, and trusts. The system should be easy to use, scalable, secure, and offer data analytics and reporting features to help stakeholders make informed decisions.

The system should allow receptionists to manage room bookings, check-ins, and check-outs efficiently, providing customers with a seamless experience. The cashier

should be able to handle payments and billing efficiently, while the accounts department should be able to monitor financial transactions, generate reports and invoices, and manage inventory.

Customers should be able to book rooms easily, access information about the hotel's services, and provide feedback about their experience. Managers should be able to monitor the performance of the hotel, track employee productivity, manage inventory and supplies, and ensure compliance with regulations and standards. Trusts should be able to monitor the financial performance of the hotel, ensure that the hotel's operations are aligned with their mission, and make data-driven decisions.

# Introduction:

## Purpose of this Document:

The main aim of this document is to specify all the requirements For hotel management system.

## Scope of this document –

This is for a hotel management system which will automate the major operations of the hotel.The objective of the hotel management system is to simplify the day to day processes of the system. This system will be able to handle many services to take care of the customer in a quick and efficient manner.

## Overview

Hotel management system is a tool for proper management and ease of management for customers and staff.

# General description:

The Online Hotel Management System is a new self contained software product which will be produced by the developers in order to overcome the problems that have occurred due to the current manual system. The proposed system will provide an easy access to the system and it will allow user friendly functions with attractive interface. The system will give better options for the problem of handling large scale of physical file system, errors occurring in calculations and all the other required tasks that has been specified by the client. The final outcome of this project will increase the efficiency of almost all the task done at the hotel in a much convenient manner.

**3. Functional Requirements:** Check-in and check-out

Facilitate the check-in and check-out process for users, including verifying identity and

payment information

Room amenities

Allow users to view and select room amenities

Billing and payment

Provide clear and transparent billing information and enable users to pay through various methods

Loyalty programs

Allow users to enroll in loyalty programs and receive rewards for repeat business

Feedback and reviews

Provide users with an opportunity to provide feedback and leave reviews of stay

**4 Interface Requirements:**

Web development languages like HTML, CSS, JavaScript and PHP will be used in order to develop this website. Bootstrap has also been used for the responsive design of the front end so that it is customizable to the devices having different dimension. The backend runs on MySQL provided by XAMPP.

**5.Performance Requirements:**

Response time: The system should respond quickly to user requests and actions, with minimal latency and delay.

Scalability: The system should be scalable, capable of handling a large number of users and transactions, without compromising performance or reliability.

Availability: The system should be available 24/7, with minimal downtime for maintenance or upgrades.

Reliability: The system should be reliable, with minimal errors or failures, and capable of recovering quickly from any disruptions.

Security: The system should be secure, protecting user data and transactions from unauthorized access or breaches.

Load testing: The system should undergo regular load testing, to ensure that it can handle peak loads and heavy traffic without compromising performance or availability.

Optimization: The system should be optimized for performance, with efficient algorithms, data structures, and processing techniques, to minimize resource usage and improve response time.

**6 .Design constrain:**

Hardware constraints: The system design may be constrained by the hardware resources available, such as servers, storage devices, and network equipment.

Software constraints: The system design may be constrained by the software resources available, such as the operating system, database management system, and programming languages.

Time constraints: The system design may be constrained by the project timeline, which may limit the scope, features, and functionality of the system.

Budget constraints: The system design may be constrained by the available budget, which may limit the investment in hardware, software, and personnel.

Regulatory constraints: The system design may be constrained by regulatory requirements, such as data privacy laws, security standards, and compliance regulations.

Compatibility constraints: The system design may be constrained by the need to integrate with existing systems, such as payment gateways, reservation systems, and loyalty programs.

Usability constraints: The system design may be constrained by the need to ensure usability and accessibility, such as designing for users with disabilities, elderly users, or users with limited technical skills.

**7. Non-Functional Attributes:**

Usability: The system should be user-friendly, easy to learn, and intuitive, with a well-designed user interface that enables users to perform tasks quickly and efficiently.

Reliability: The system should be reliable, with a low error rate and minimal downtime, ensuring that users can access and use the system at all times.

Security: The system should be secure, protecting user data and transactions from unauthorized access or breaches, and complying with regulatory requirements.

Performance: The system should be fast and responsive, with minimal latency and delay, enabling users to perform tasks quickly and efficiently.

Scalability: The system should be scalable, able to handle a large number of users and transactions, without compromising performance or reliability.

Maintainability: The system should be easy to maintain, with well-documented code, clear error messages, and easy-to-use debugging tools, enabling developers to identify and fix issues quickly.