A

Project

On

HotelHQ - Restaurant Hotel Sales/Inventory System

For the partial fulfillment of the requirements for the degree of Bachelor
of Computer Application under Pokhara University

Submitted to

Department of Computer Application

National Academy of Science and Technology

Under the Supervision of

Mr. Sunil Bahadur Bist

Lecturer, Department of Computer Application

Submitted By

Gaurav Kathayat (23530007)

Motiram Mahato (23530015)

Pradip Panthi (235300037)

Bachelor of Computer Application, 4th Semester

May, 2025

National Academy of Science and Technology

(Affiliated to Pokhara University)

Accredited by University Grants Commission (UGC), Nepal (2022)

Uttarbehedi 4 Dhangadhi, Kailali, Nepal

Abstract

HotelHQ is a web-based Hotel and Restaurant Inventory Sales Management System designed to address the challenges of daily operations in the bustling hospitality industry, where accurate records of sales and inventory are crucial for the smooth operation of the business and enhancing the customer experience. Our system primarily handles room and table bookings, inventory status, menu/item ordering, and billing.

We will also explore optional features, such as the visual representation of sales data and the generation of specific insights from this data. These features will be implemented only if time and resources permit, utilizing libraries such as Pandas, Matplotlib, and Scikit-learn. These visualizations may include charts or dashboards to better display the sales data.

Keywords: HotelHQ, Web-based System, Scikit-learn, Flask Framework, SQL Database, Privilege-Based Access System

1. Introduction

The HotelHQ project is aimed at developing a robust and efficient platform to manage hotel and restaurant operations, enhancing both business and customer experiences. This system will provide seamless handling of bookings, inventory, ordering, and billing. By streamlining day-to-day tasks in the hospitality industry, HotelHQ ensures smooth operations, optimizing both customer satisfaction and business workflows.

HotelHQ is built using HTML, CSS, and JavaScript for the front-end, offering an intuitive and user-friendly interface. On the back-end, Python Flask provides a lightweight framework for handling business logic, while MariaDB is used for securely managing and storing data. The system will implement privilege-based access for login, ensuring that different users have different levels of access. For example, sales information will not be available to customers or waitstaff, but managers and owners can log in at any time to view this data. This ensures that sensitive information is only accessible to authorized personnel.

Optional features such as data visualization and sales insights are planned for future implementation, leveraging libraries such as Pandas, Matplotlib, and Scikit-learn. These tools will enable the system to generate actionable insights through visual representations of sales data.

With this system, business owners can ensure that all tasks are organized efficiently, allowing them to focus on maintaining a well-run and successful operation.

2. Problem Specification

The hospitality industry, particularly in hotel and restaurant operations, faces multiple challenges that hinder efficiency, increase operational costs, and reduce customer satisfaction. Managing bookings, inventory, orders, and billing manually or through disjointed systems results in confusion, errors, and delays, affecting the smooth running of the business. These inefficiencies often lead to customer dissatisfaction and lost revenue opportunities.

The main problems that the current systems or manual methods in the hospitality industry face include:

> ➢ Inefficient Booking and Reservation Management
>
> ➢ Lack of Real-Time Inventory Management
>
> ➢ Cumbersome Billing Process
>
> ➢ Lack of Data Insights
>
> ➢ Lack of Access Control
>
> ➢ Poor Record Maintenance
>
> ➢ Difficulty in Prioritizing Business Needs

3. Objective

The primary objective of the HotelHQ project is to create a comprehensive, user-friendly, and efficient web-based platform that addresses the key issues faced by the hospitality industry in managing bookings, inventory, ordering, and billing. By streamlining operations and providing real-time insights, HotelHQ will enhance the overall customer experience while improving business efficiency.

Objectives of the project:

- To develop a secure login and registration system with privilege-based access, ensuring role-specific access for managers, owners, staff, and customers.

- To enable efficient management of room and table bookings, inventory, and orders, with a real-time overview of the entire system.

- To implement a billing and payment management system, reducing the errors and delays associated with manual billing processes.

- To minimize the reliance on paper-based processes, enhancing operational efficiency through digital solutions.

- To improve record maintenance by using a unified system to store and manage data securely, ensuring easy retrieval and traceability.

- To provide managers and owners with visual representations of sales data and actionable insights into business performance using tools like Pandas, Matplotlib, and Scikit-learn — to be implemented as an optional feature depending on time and resource availability.

4. Related Work Review

The related work review examines existing hotel and restaurant management systems to identify strengths, weaknesses, and opportunities for improvement in our proposed HotelHQ system. By analyzing similar platforms and tools, we aim to incorporate successful strategies while addressing shortcomings observed in current implementations. This review helps us shape a more user-centric, efficient, and functional system. The similar platforms are given below:

1. Hotelogix Property Management System

Hotelogix is a cloud-based hotel management software widely used in the hospitality industry for room booking, inventory tracking, and billing.

Strategies:

    i.   Real-time room availability and booking.

    ii.   Integration with POS systems and third-party platforms.

    iii.   Multi-user access and role-based permissions.

Drawbacks:

    i.   Primarily subscription-based with high monthly costs.

    ii.   Complexity in customization for small hotels/restaurants.

    iii.   Limited insight generation unless paired with premium modules

2. Odoo Hotel Management Module

Odoo, an open-source ERP platform, includes modules for hotel and restaurant management.

Abilities:

    i.   Modular architecture allows integration of inventory, booking, billing, and CRM.

    ii.   Open-source and customizable.

    iii.   Rich dashboard and reporting features.

Drawbacks:

    i.   Requires technical expertise to set up and configure.

    ii.   Not tailored out-of-the-box for the hospitality domain.

    iii.   Some features hidden behind paid enterprise version.

3. eZee Absolute Hotel Management System

eZee Absolute offers web-based hotel property management, including restaurant POS, housekeeping, and online bookings.

Abilities:

i. Multi-device access, including mobile.

ii. Integrated POS for restaurant operations.

iii. Reports and analytics for managerial decision-making.

Drawbacks:

i. Primarily targeted at large-scale hotels.

ii. Can be overwhelming for small restaurants or individual hotels.

iii. No local customization or offline mode.

5. System Requirements

The HotelHQ system requires both functional and non-functional requirements to ensure reliability, scalability, and user satisfaction. The system is designed to be simple for end-users while being robust enough for administrative and managerial operations.

Functional Requirements

- User Registration and Login with Role-based Access
  (Customer, Waiter, Manager, Owner)
- Room and Table Booking Module
  (Book, View Availability, Cancel)
- Menu Item Browsing and Ordering
- Inventory Management
  (Add, Update, Remove Items by Admin/Manager)
- Sales Billing System
  (Generate and Store Bills)
- Privilege-based Data Access
  (Sales data visible only to managers and owners)
- Admin Dashboard for Managing Inventory and Orders
- (Optional) Visual Sales Reports and Business Insights using
  Pandas, Matplotlib, Scikit-learn

Non-Functional Requirements

- Secure Authentication

  (Password hashing and session management using Flask)

- Responsive Frontend Interface

  (Built with HTML, CSS, JS for ease of use on all devices)

- Efficient Database Operations

  (MariaDB with optimized schema and queries)

- Modular Code Structure

  (Separation of backend logic, templates, and routes)

- Fast Page Loading and UI Feedback

- Scalable Backend Architecture

  (Supports future upgrades like analytics or real-time updates)

Requirement Prioritization Table

This table helps identify the core (high-priority) functions required for the HotelHQ Hotel and Restaurant Inventory Sales Management System and categorizes other useful enhancements based on development feasibility.

Table: Requirement Prioritization Table

Priority	Requirement	Type
High	Role-based Login (Customer, Waiter, Manager, Owner)	Functional
High	Room and Table Booking System	Functional
High	Menu Browsing and Order Placement	Functional
High	Inventory Management by Admin/Manager	Functional
High	Billing System (Invoice Generation)	Functional
High	Sales Data Access for Manager/Owner Only	Functional
Medium	Visual Sales Insights (Graphs, Reports)	Functional (Optional)
Medium	Password Hashing and Secure Login Sessions	Non-functional
Medium	Responsive UI for Multiple Devices	Non-functional
Medium	Error Handling and User Feedback Messages	Non-functional
Low	UI Animations and Aesthetic Enhancements	Non-functional
Low	AI-powered Sales Prediction (Scikit-learn)	Functional (Optional)

6. Methodology and System Design

Methodology

We follow the System Development Life Cycle (SDLC) to systematically build the HotelHQ platform. For development, we have adopted an Agile methodology approach. Agile enables us to deliver the system in iterative cycles, maintain close collaboration with stakeholders (e.g., restaurant or hotel management), and adapt quickly to evolving requirements such as new features or design tweaks.

While Agile is our core approach, we've also incorporated aspects of the Waterfall Model in the early planning and design phases to ensure:

- Simple and easy-to-follow structure.
- Reduced development risks.
- Clear role assignments and team collaboration.
- Efficient resource usage.
- Transparent tracking and documentation.
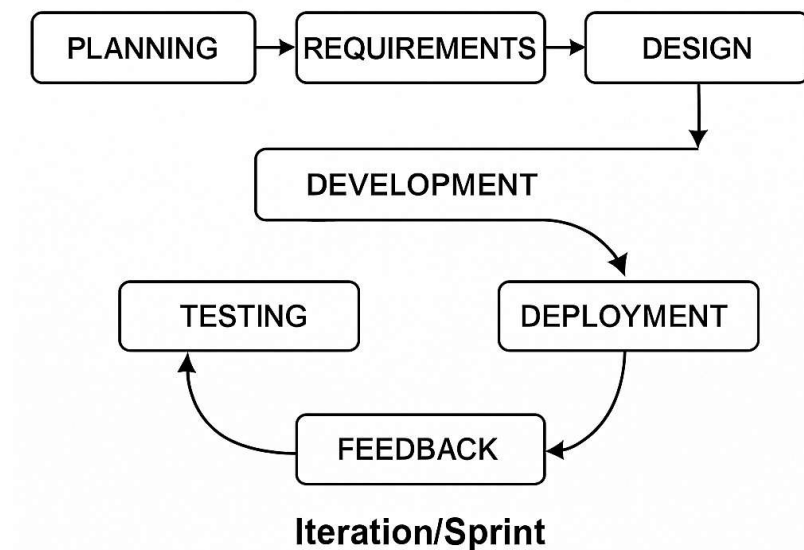
Hybrid SDLC Model – Agile with Waterfall Phases

PLANNING → REQUIREMENTS → DESIGN → DEVELOPMENT → DEPLOYMENT → FEEDBACK → TESTING

Iteration/Sprint

Figure 6.1: Hybrid SDLC Model -Agile with Waterfall Phases

System Design

The design of the HotelHQ Hotel and Restaurant Inventory Sales Management System is user-friendly and role-based, allowing access privileges based on user type (e.g., customer, waiter, manager, owner).

Frontend:

- Built using HTML, CSS, and JavaScript.
- Focus on responsiveness and ease of use.
- Separate interfaces for different user roles (e.g., booking portal for customers, inventory dashboard for managers).

Backend:

- Developed using Python (Flask).
- Handles login sessions, role-based routing, order processing, inventory control, and billing logic.

Database:

MariaDB is used for storing:

- User credentials and roles.
- Inventory status.
- Booking details.
- Order histories and billing records.
- Sales data for analytics.

System Architecture:

The high-level system architecture of HotelHQ consists of:

- Frontend (Client-side): HTML, CSS, JavaScript-based UI tailored for each user role.
- Backend (Server-side): Flask framework in Python handling APIs, logic, and session control.
- Database Layer: MariaDB for relational data handling—CRUD operations, analytics, and transactional records.

- Role-Based Access Control: Ensures data privacy and functionality control. For example, only managers and owners can view or analyze sales data, while customers and waiters have limited access to ordering and booking features.

- Optional Analytics Component (if time/resources allow): Uses Pandas, Matplotlib, and Scikit-learn for sales visualization and performance insights.
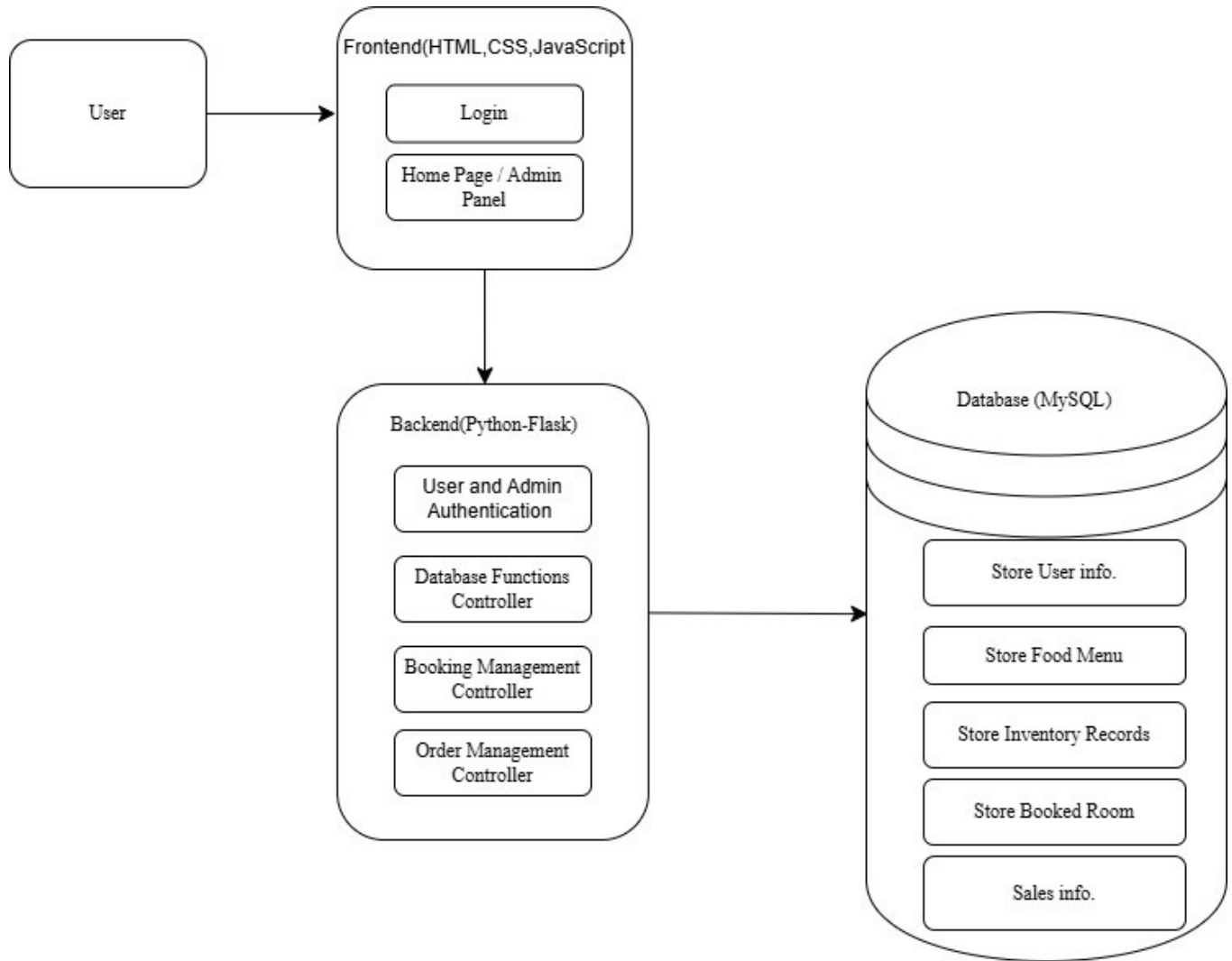
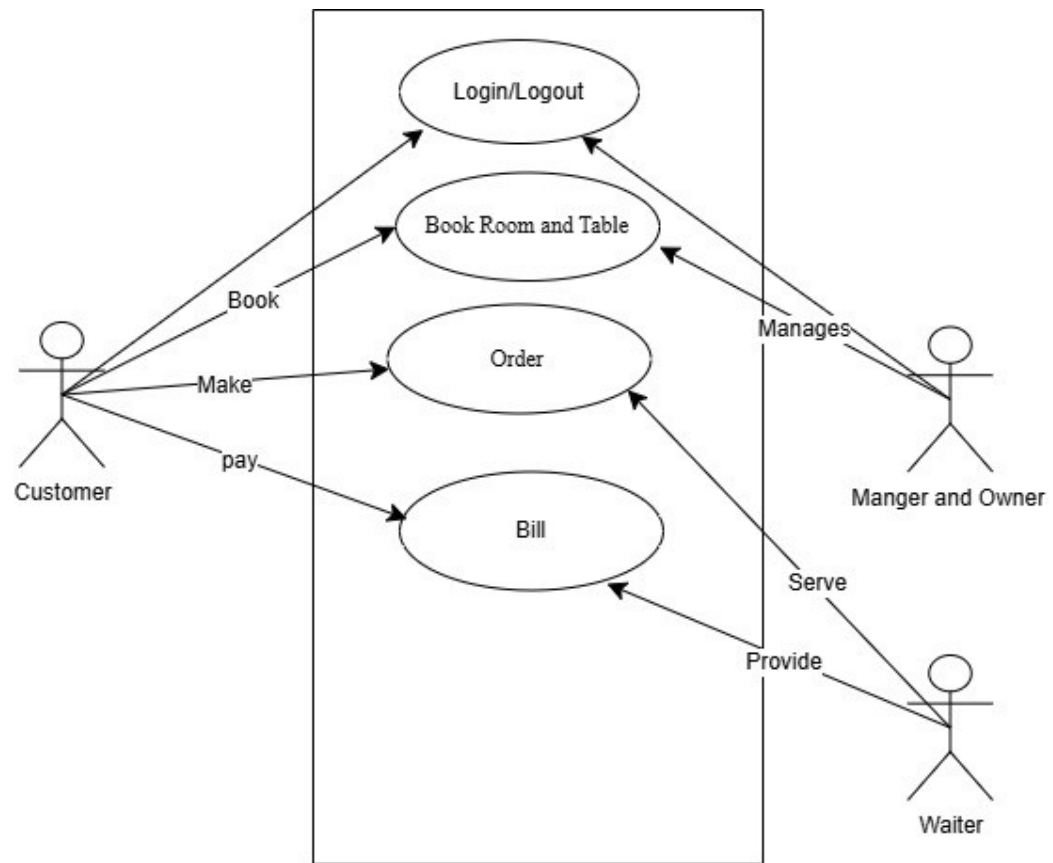

Figure 6.1.2: System Architecture

Use Case Diagram

Figure 6.1.3: Use-case Diagram

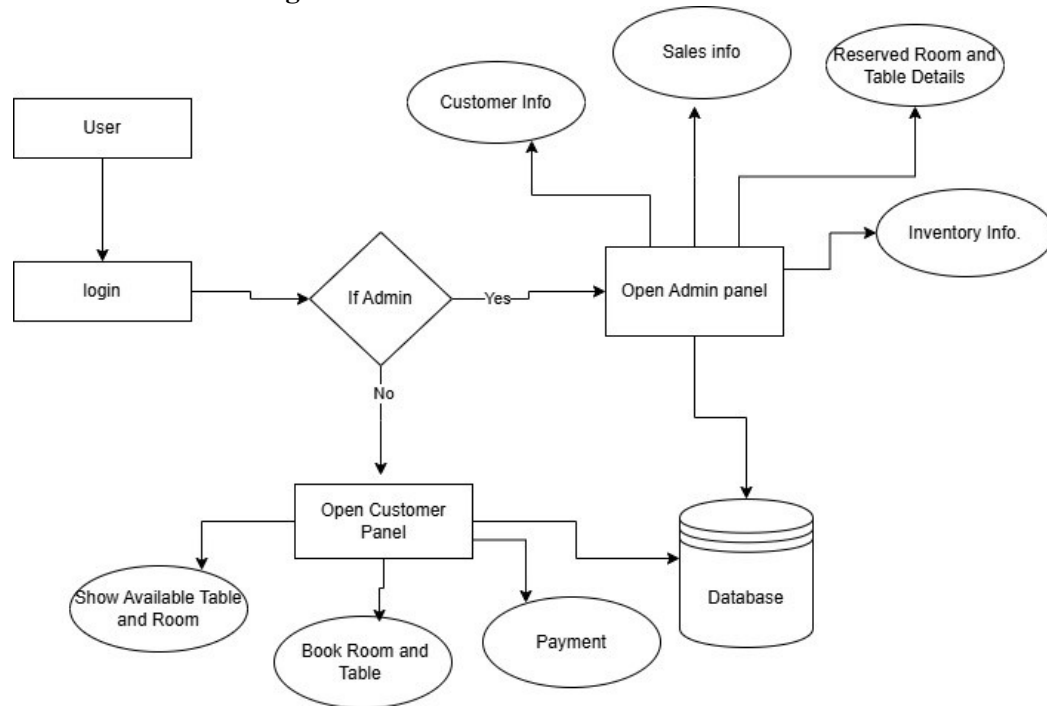**Basic Work Flow Diagram**

Figure 6.1.5: Basic Workflow Diagram

7. Project Schedule

The project schedule has been planned carefully according to the system development life cycle and project requirements. The total duration for the completion of HotelHQ is estimated to be around two months. This includes phases such as requirement analysis, proposal defense, feasibility study, system design, development (coding), testing, documentation, and final presentation.

Each phase has been allocated a specific timeframe to ensure timely delivery and efficient resource management. The team has adopted an Agile approach with overlapping tasks where applicable, allowing for iteration and flexibility during development.

Activities	May 01-09	May 10-13	May 14-20	May 21-30	June 01-30	July 01-15	July 16-22	July 23-30
Project analysis	■							
Proposal Defense	■							
Feasibility study		■						
Designing			■					
Coding				■	■			
Testing				■	■			
Final testing							■	
Documentation	■	■	■	■	■	■	■	
Final Submission								■

8. Development Cost Estimation

Development cost estimation refers to the projection of all resources—time, manpower, infrastructure, and miscellaneous expenses—needed to successfully complete the HotelHQ system. A bottom-up approach is used here, where individual roles are estimated for their effort and cost, and the total project cost is aggregated accordingly.

Effort and Timeline

Total Duration: 10 weeks

Team Members: 3

Weekly Working Hours per Member: 36

Total Estimated Effort: 36 hours × 10 weeks × 3 members = 1080 hours

Role	Duration	Estimated Cost (Rs.)
Project Manager	2.4 months	76,000
Frontend Developer	2.4 months	68,000
Backend Developer	2.4 months	70,000
QA Engineer *(shared role)*	2.4 months	28,000 *(approx.)*
UI/UX Designer *(shared role)*	2.4 months	23,000 *(approx.)*
Total		2,65,000

Table: Personnel Cost Estimation

Infrastructure and Miscellaneous Costs

Includes server hosting, electricity, internet, domain names, version control services, etc. Estimated Total: Rs. 18,500

Grand Total Development Cost

Rs. 2,83,500

9. Deliverables

The deliverables are the final tangible and intangible outputs produced during the development lifecycle of the HotelHQ project. These include all the artifacts that demonstrate the successful implementation and completion of the system.

List of Project Deliverables:

- Complete Project Report and Documentation
- Working Web-Based HotelHQ Application
- Source Code Files (Frontend + Backend + Database Scripts)
- Project Presentation Slides (for defense and evaluation)
- Deployment Instructions (if applicable)
- User and Admin Manuals (optional)

10. Conclusion

HotelHQ is a well-structured, scalable, and practical solution aimed at modernizing hotel and restaurant management through efficient handling of bookings, inventory, orders, and billing. The system incorporates essential features such as role-based access, real-time data management, secure login, and optional analytics tools to deliver improved operational efficiency and customer satisfaction.

By automating and digitizing the core operations of the hospitality business, HotelHQ reduces dependency on manual methods and minimizes errors. With the potential for further enhancements like AI-based sales forecasting and interactive dashboards, HotelHQ can evolve into a powerful all-in-one management tool tailored for the hospitality industry.

11.Refrences

[1] "System Design - System Analysis and Design," *TutorialsPoint*. Available: https://www.tutorialspoint.com/system_analysis_and_design/system_design.htm. [Accessed: May 8, 2025].

[2] "Welcome to Flask — Flask Documentation (2.0.x)," *Pallets Projects*. Available: https://flask.palletsprojects.com/en/2.0.x/. [Accessed: May 8, 2025].

[3] "MariaDB Foundation," *MariaDB.org*. Available: https://mariadb.org/. [Accessed: May 8, 2025].

[4] "Pandas - Python Data Analysis Library," *pandas.pydata.org*. Available: https://pandas.pydata.org/. [Accessed: May 8, 2025].

[5] "Matplotlib: Visualization with Python," *matplotlib.org*. Available: https://matplotlib.org/. [Accessed: May 8, 2025].

[6] "Scikit-learn: Machine Learning in Python," *scikit-learn.org*. Available: https://scikit-learn.org/stable/. [Accessed: May 8, 2025].

[7] "Hotelogix Property Management System," *Hotelogix.com*. Available: https://www.hotelogix.com/. [Accessed: May 8, 2025].

[8] "Odoo Hotel Management Module," *Odoo.com*. Available: https://www.odoo.com/page/hotel-management. [Accessed: May 8, 2025].

[9] "eZee Absolute - Hotel Management Software," *ezeeabsolute.com*. Available: https://www.ezeeabsolute.com/. [Accessed: May 8, 2025].

[10] "Agile Methodology Guide," *Atlassian*. Available: https://www.atlassian.com/agile. [Accessed: May 8, 2025].

[11] "IEEE Reference Guide," *IEEE Author Center*. Available: https://journals.ieeeauthorcenter.ieee.org/wp-content/uploads/sites/7/IEEE_Reference_Guide.pdf. [Accessed: May 8, 2025].