**A Project Report**

**on**

***“HOTEL BOOKING SYSTEM”***

Submitted in partial fulfillment of the requirements

For the award of the degree of Bachelor of Technology in Computer Science & Engineering

AKS UNIVERSITY, SATNA

**B. Tech (CSE) 6th Semester**

#### Submitted by

##### AYUSH CHAURASIYA (B2255R10106007)

##### HIMANSHU GAUTAM (B2255R10106110)

**Under The Guidance of Approved by**

Mr. Sajal Kar Prof. (Dr.) Akhilesh A. Waoo

(Assistant Professor) (Associate Dean & Head CS/IT)

**Department of Computer Science & Engineering**

**AKS University, Satna (M.P.)**

## CERTIFICATE

This certify that the project report entitled “**HOTEL BOOKING SYSTEM**” submitted by partial fulfilment of the requirement for the degree of Bachelor of Engineering in Technology in **Jan-June 2025** AKS University, Satna is a bonafide project work carried out by **AYUSH CHAURASIYA (B2255R10106007),** under my supervision. The subject of the project report has been approved by supervisor. This is also to certify that it is his/her original work and no part of this project is report has been submitted for any other degree/diploma.

All the assistance the and help received during the course of the investigation has been duly acknowledged.

1. I am satisfied that the report presented by **AYUSH CHAURASIYA (B2255R10106007)** is worthy of consideration for award of the degree.
2. I certify:
   1. That he/she pursued the prescribed course for project.
   2. That he/she bears good moral character.

**Place: AKSU, Satna ………………………………**

**Date:…/…./……… External’s Signature**

……………………………... ………………………………

**Supervisor Head of Department**

**(Mr. Sajal Kar)**  **Prof. (Dr.) Akhilesh A. Waoo**

(Assistant Professor) (Asso. Dean, Professor, CS/IT)

## CERTIFICATE

This certify that the project report entitled “**HOTEL BOOKING SYSTEM**” submitted by partial fulfilment of the requirement for the degree of Bachelor of Engineering in Technology in **Jan-June 2025** AKS University, Satna is a bonafide project work carried out by **HIMANSHU GAUTAM (B2255R10106110),** under my supervision. The subject of the project report has been approved by supervisor. This is also to certify that it is his/her original work and no part of this project is report has been submitted for any other degree/diploma.

All the assistance the and help received during the course of the investigation has been duly acknowledged.

1. I am satisfied that the report presented by **HIMANSHU GAUTAM (B2255R10106110)** is worthy of consideration for award of the degree.
2. I certify:
   1. That he/she pursued the prescribed course for project.
   2. That he/she bears good moral character.

**Place: AKSU, Satna ………………………………**

**Date:…/…./……… External’s Signature**

……………………………. ………………………………

**Supervisor Head of Department**

**(Mr. Sajal Kar)**  **Prof. (Dr.) Akhilesh A. Waoo**

(Assistant Professor) (Asso. Dean, Professor, CS/IT)

**CERTIFICATE BY THE CANDIDATE**

I certify that the project report entitled “**HOTEL BOOKING SYSTEM**” is my own work conducted under the supervision of **Mr. Sajal Kar** (Supervisor), Department of Computer Science, AKS University, Satna (M.P.) for partial fulfilment of the requirement for the degree in Bachelor of Engineering in Technology in **Jan-June 2025.**

I further certify that to the best of my knowledge and belief the project report does not contain any part of this work which has been submitted for the award of any degree either in this university or in any other University/ Deemed University/ Institutes.

* AYUSH CHAURASIYA (B2255R10106007) - B.Tech (CSE) 6th Semester

**Place:** AKSU, Satna

**Date:**…./…../………

…………………………..

Signature of Candidate

Ayush Chaurasiya

**CERTIFICATE BY THE CANDIDATE**

I certify that the project report entitled “**HOTEL BOOKING SYSTEM**” is my own work conducted under the supervision of **Mr. Sajal Kar** (Supervisor), Department of Computer Science, AKS University, Satna (M.P.) for partial fulfilment of the requirement for the degree in Bachelor of Engineering in Technology in **Jan-June 2025.**

I further certify that to the best of my knowledge and belief the project report does not contain any part of this work which has been submitted for the award of any degree either in this university or in any other University/ Deemed University/ Institutes.

* HIMANSHU GAUTAM (B2255R10106110**)** - B.Tech (CSE) 6th Semester

**Place:** AKSU, Satna

**Date:**…./…../………

…………………………..

Signature of Candidate

Himanshu Gautam

## SELF DECLARATION

I hereby declare that the work presented in this project entitled **“HOTEL BOOKING SYSTEM "** towards the partial fulfilment of the requirement for the award of **Degree in B. Tech** in Department of Computer Science, **AKS University, Satna (M.P.)** is an authentic record of my own work.

I have not submitted the matter embodied in the project for the award of any other degree or diploma to any other institute or university.

…………………………..

Signature of Candidate

Ayush Chaurasiya

## 

## SELF DECLARATION

I hereby declare that the work presented in this project entitled **“HOTEL BOOKING SYSTEM "** towards the partial fulfilment of the requirement for the award of **Degree in B. Tech** in Department of Computer Science, **AKS University, Satna (M.P.)** is an authentic record of my own work.

I have not submitted the matter embodied in the project for the award of any other degree or diploma to any other institute or university.

…………………………..

Signature of Candidate

Himanshu Gautam

## ACKNOWLEDGEMENT

It is a great for me in taking this opportunity to express my sincere thanks and ineptness to **Prof. (Dr.) Akhilesh A. Waoo**, Head of the Department of CSE, AKS University, Satna (M.P.)

I consider myself lucky enough to have such a great project. This project would add as an asset to my profile.

At this moment of accomplishment, first of all I pay homage to my guide, **Mr. Sajal Kar** from AKS University Satna (M.P.). This work would not have been possible without his guidance, support and encouragement. Under his guidance I successfully overcame many difficulties and learned a lot.

I am deeply and forever indebted to my parents for their love, support and encouragement throughout my entire life.

.

**ABSTRACT**

The rapid advancement of information technology has revolutionized the hospitality industry, making online hotel booking systems an essential tool for both service providers and customers. This project, “Hotel Booking System,” presents a comprehensive, user-friendly web application designed to streamline the process of hotel room reservations, management, and administration.

Developed using widely adopted technologies—HTML, CSS, JavaScript, PHP, and MySQL—the system offers a robust backend for data management and a responsive, intuitive frontend for seamless user interaction. The application supports two primary user roles: regular users and administrators. Regular users can register, log in, browse available rooms, make bookings, and manage their reservations, including cancellations. The administrator panel empowers hotel staff to oversee all bookings, manage room inventory, and handle cancellations, ensuring operational efficiency and real-time data accuracy.

Security and data integrity are prioritized throughout the system. User authentication is implemented with password hashing, and session management ensures that sensitive operations are accessible only to authorized users. The database schema is carefully designed to maintain relational integrity between users, rooms, and bookings, reducing redundancy and preventing conflicts such as double bookings. The user interface is crafted for clarity and ease of use, incorporating modern design principles to enhance the overall user experience.

The Hotel Booking System is intended for deployment on local servers using platforms like XAMPP, making it accessible for small to medium-sized hotels and educational purposes. The modular architecture allows for easy scalability and future enhancements, such as payment gateway integration, customer feedback modules, and advanced analytics.

This project not only demonstrates the practical application of web development and database management skills but also addresses real-world challenges faced by the hospitality industry. By automating reservation processes and centralizing management tasks, the system reduces manual workload, minimizes errors, and improves customer satisfaction. The report documents the complete lifecycle of the project, from requirements analysis and system design to implementation, testing, and future recommendations, providing a valuable reference for students, developers, and hotel operators interested in digital transformation of hospitality services.

## TABLE OF CONTENT

* Introduction……...………………………….………………**1**
* Objective of The Project…………….……….…………….. **2**
* Scope of The Project…..……………………………..….… **4**
* Need of The Project…. ………………………………...…. **6**
* Main Module………………………….…………………… **8**
* Technical Overview ………………………………………. **13**
* Design and Development Process…..….…………… …..... **17**
* Benefits and Applications………………………..…………**20**
* Definition of Problem………………………………………**25**
* Logic of the Project…………………………………………**29**
* Error handling………………………………………………**34**
* Non-Functional Requirements:……………………………..**37**
* System Configuration……………………………………….**39**
* Block Diagram………………………………….……….… **40**
* DFD Diagram………………………………………….....…**41**
* ER Diagram…………………………………………………**42**
* Output Screen …..……………………………….………… **43**
* CODE………………………………………………………..**46**
* Project Explanation…..…………………………….……… **.53**
* Conclusion…………………………….………….…………**59**
* Bibliography and References……………………….………**60**

**Introduction**

The hospitality industry has experienced a profound transformation in recent years, largely driven by the integration of digital technologies into traditional business processes. Among these advancements, online hotel booking systems have emerged as a cornerstone of modern hotel management and customer service. These systems not only enhance the efficiency of hotel operations but also significantly improve the guest experience by providing convenience, transparency, and accessibility.

This project, titled “Hotel Booking System,” is conceived to address the growing need for an automated, reliable, and user-friendly platform for hotel room reservations and management. The traditional methods of booking rooms—such as phone calls, in-person visits, or email requests—are often time-consuming, prone to human error, and limited by operational hours. In contrast, an online booking system offers 24/7 accessibility, real-time availability updates, and a streamlined workflow for both customers and hotel staff.

The primary objective of this project is to develop a comprehensive web application that facilitates the entire process of hotel room booking, from user registration and authentication to room selection, reservation, and cancellation. The system is designed to cater to two main user groups: guests (regular users) and hotel administrators. Guests can easily browse available rooms, make bookings, view their reservations, and cancel if necessary. Administrators are equipped with tools to manage room inventory, oversee all bookings, handle cancellations, and ensure that the hotel’s resources are utilized efficiently.

To achieve these goals, the system is built using a combination of HTML, CSS, and JavaScript for the frontend, and PHP with MySQL for the backend and database management. This technology stack is chosen for its wide adoption, ease of deployment, and flexibility, making the solution suitable for small to medium-sized hotels as well as educational demonstrations.

Security, usability, and scalability are central to the system’s design philosophy. Features such as password hashing, session management, and role-based access control are implemented to protect user data and prevent unauthorized access. The database is structured to maintain relational integrity and support future enhancements, such as integration with payment gateways or customer feedback modules.

**Objective of the Project**

**To Develop an Automated Hotel Booking Platform:**

The foremost objective is to design and implement an automated hotel booking system that replaces traditional, manual reservation methods with a digital, web-based solution. This system aims to streamline the entire process of booking, managing, and canceling hotel room reservations for both guests and hotel staff.

**To Enhance User Convenience and Accessibility:**

The project seeks to provide guests with a convenient and accessible interface for booking rooms at any time and from any location. By enabling users to browse available rooms, check real-time availability, and complete reservations online, the system removes the constraints of physical presence and business hours.

**To Offer a User-Friendly Experience:**

A key goal is to ensure that the platform is intuitive and easy to use for all types of users, including those with limited technical skills. The interface is designed to be clean, responsive, and straightforward, guiding users through registration, login, room selection, booking, and management of reservations with minimal effort.

**To Provide Real-Time Room Availability:**

The system is designed to update room availability instantly as bookings are made or canceled. This prevents double-booking, reduces errors, and builds trust with users by displaying only the rooms that are truly available at any given moment.

**To Empower Hotel Administrators with Management Tools:**

The project aims to equip hotel administrators with a powerful admin panel that allows them to manage all aspects of hotel operations efficiently. This includes adding, editing, or removing rooms, monitoring all bookings, handling cancellations, and managing user accounts from a centralized dashboard.

**To Ensure Data Security and Integrity:**

Protecting user data is a critical objective. The system uses secure authentication methods, such as password hashing and session management, to prevent unauthorized access. The database is structured to maintain relational integrity, ensuring that all transactions are accurately recorded and that sensitive information is safeguarded.

**To Reduce Administrative Workload:**

By automating repetitive and time-consuming tasks, the system significantly reduces the administrative burden on hotel staff. This allows employees to focus on delivering quality service and attending to guest needs rather than managing paperwork and manual records.

**To Support Scalability and Future Enhancements:**

The architecture of the system is modular and scalable, allowing for the easy addition of new features such as online payment integration, customer feedback modules, or advanced reporting tools. This ensures that the system can grow and adapt to the evolving needs of the hotel industry.

**To Improve Operational Efficiency:**

The project aims to streamline hotel operations by centralizing data, automating booking workflows, and providing actionable insights through analytics. This leads to better resource utilization, faster response times, and more effective decision-making for hotel management.

**To Increase Customer Satisfaction:**

By offering a transparent, reliable, and flexible booking process, the system enhances the overall guest experience. Features such as instant booking confirmation, easy cancellation, and the ability to view or modify reservations contribute to higher levels of customer satisfaction and loyalty.

**To Serve as an Educational Demonstration:**

The project also has an academic objective: to illustrate the practical application of web development, database management, and software engineering principles. It provides a comprehensive case study for students, educators, and aspiring developers interested in building real-world software solutions.

**Scope of the Project**

**Covers the Complete Room Booking Lifecycle:**

The scope of this hotel booking system encompasses every stage of the room reservation process, from user registration and login to room selection, booking confirmation, and cancellation. The platform is designed to handle both guest and administrative activities, ensuring a seamless experience for all stakeholders involved in hotel operations.

**Caters to Multiple User Roles:**

The system is built to support two primary user roles: guests (regular users) and hotel administrators. Guests can browse available rooms, make bookings, and manage their reservations, while administrators have access to powerful management tools for overseeing room inventory, handling bookings, and monitoring system usage.

**Web-Based Accessibility:**

The project is developed as a web application, making it accessible from any device with an internet connection and a web browser. This ensures that both guests and hotel staff can interact with the system from desktops, laptops, tablets, or smartphones, thereby increasing the system’s reach and usability.

**Real-Time Data Management:**

The system provides real-time updates on room availability, booking status, and user activity. Whenever a booking is made or canceled, the database is updated instantly, ensuring that all users see the most current information. This minimizes the risk of double-booking and enhances trust in the system’s reliability.

**Centralized Administration:**

The scope includes a centralized admin panel where hotel staff can manage all aspects of the hotel’s room inventory, monitor bookings, handle cancellations, and view user information. This centralization streamlines hotel operations and enables more effective resource management.

**Secure User Authentication and Authorization:**

The system incorporates secure login and registration mechanisms, with password hashing and session management to protect user credentials and prevent unauthorized access. Role-based access control ensures that only authorized users can perform sensitive operations, such as managing rooms or viewing all bookings.

**Database Integration:**

The project utilizes a relational database (MySQL) to store and manage all data related to users, rooms, and bookings. The database design ensures data consistency, integrity, and scalability, supporting the addition of new features or increased user load in the future.

**User-Friendly Interface:**

The application features a modern, intuitive user interface designed to make navigation and interaction straightforward for users of all technical backgrounds. Clear menus, responsive design, and informative feedback messages contribute to a positive user experience.

**Scalability for Future Enhancements:**

The system’s modular architecture allows for easy expansion. Future enhancements could include online payment integration, customer feedback and review modules, automated email notifications, advanced analytics, and support for multiple hotels or branches.

**Educational and Demonstrative Value:**

In addition to its practical application, the project serves as a comprehensive demonstration of web development, database design, and software engineering best practices. It is suitable for academic purposes, training, and as a foundation for more complex hotel management systems.

**Target Audience:**

The system is intended for use by small to medium-sized hotels, guesthouses, and educational institutions seeking to modernize their booking processes. It is also valuable for students and developers interested in learning about full-stack web application development in a real-world context.

**Need of the Project**

The hospitality industry is one of the fastest-growing sectors worldwide, driven by increased travel, tourism, and business activities. As hotels strive to attract and retain guests, the efficiency and convenience of their booking processes have become critical factors in delivering superior customer experiences. Traditional methods of hotel room booking—such as walk-ins, phone calls, or manual record-keeping—are increasingly inadequate in meeting the demands of modern travelers and hotel managers. This is where the need for an automated, web-based Hotel Booking System becomes evident.

**Addressing Customer Expectations**

Today’s customers expect instant access to information and the ability to make reservations anytime, anywhere. A Hotel Booking System enables guests to browse available rooms, compare prices, view amenities, and secure bookings in real time, all from the comfort of their homes or mobile devices. This level of convenience is not possible with manual systems, which are limited by working hours, human error, and slow response times. Furthermore, customers appreciate transparency in pricing and room availability, which a digital system can provide with up-to-date data.

**Enhancing Operational Efficiency**

For hotel staff and management, a manual booking process is labor-intensive and error-prone. It often leads to issues such as double bookings, misplaced records, and inefficient room allocation. An automated Hotel Booking System streamlines these operations by centralizing all booking data, automating availability checks, and generating reservation confirmations instantly. This reduces the administrative burden on staff, minimizes mistakes, and allows employees to focus on delivering better service to guests.

**Data Management and Reporting**

Hotels generate large volumes of data related to reservations, guest preferences, occupancy rates, and revenue. Managing this information manually is not only time-consuming but also increases the risk of data loss or inconsistency. A computerized Hotel Booking System securely stores all data in a centralized database, making it easy to retrieve, update, and analyze information. Built-in reporting tools can provide valuable insights into business performance, occupancy trends, and customer behavior, empowering managers to make informed decisions and devise effective marketing strategies.

**Security and Reliability**

Security is a major concern in the hospitality industry, especially when handling sensitive guest information and payment details. Manual systems are vulnerable to unauthorized access and data breaches. A well-designed Hotel Booking System incorporates robust authentication, data encryption, and access controls, ensuring that only authorized personnel can view or modify sensitive information. Regular backups and fail-safe mechanisms further enhance the reliability of the system.

**Scalability and Future Growth**

As hotels expand their operations, either by increasing room inventory or adding new locations, manual systems struggle to keep up with the growing complexity. A scalable Hotel Booking System can easily accommodate additional rooms, users, and features without significant rework. It also supports integration with other services, such as payment gateways, review platforms, and third-party travel agencies, opening up new avenues for business growth**.**

**Main Module**

The main module of the Hotel Booking System is the central component that integrates all critical functionalities required for efficient hotel room reservation, management, and administration. It is designed to ensure seamless interaction between users and the system, providing a robust foundation for both guest-facing and administrative operations. Below are the key sub-modules and their detailed descriptions:

**1. User Registration and Authentication Module**

**User Registration:**

Allows new users (guests) to create an account by providing necessary details such as username, password, and contact information. The system validates the input to prevent duplicate accounts and ensures all mandatory fields are filled.

**Authentication:**

Implements secure login functionality using password hashing and session management. Only registered users can access booking features, while administrators have elevated privileges.

**Session Management:**

Maintains user sessions to keep users logged in securely until they choose to log out or their session expires.

**Password Management:**

Includes features for password reset or change, ensuring users can recover or update their credentials securely.

**2. Room Management Module**

**Room Inventory Management:**

Enables administrators to add, edit, or delete room records. Each room is defined by attributes such as room number, type (AC/Non-AC, Single/Double/Suite), price, and current status (available/booked).

**Room Availability:**

Administrators can update the availability status of rooms based on bookings, maintenance, or other operational needs.

**Room Categorization:**

Supports categorization of rooms for easier filtering and searching (e.g., by type, price range, amenities).

**3. Room Browsing and Search Module**

**Room Listing:**

Displays all available rooms to users, with clear information on type, price, amenities, and current availability.

**Search and Filter:**

Users can search for rooms based on criteria such as room type, price, and availability dates. Filters help users quickly find rooms that match their preferences.

**Room Details View:**

Provides detailed information about each room, including images, descriptions, and features.

**4. Booking and Reservation Module**

**Booking Form:**

Allows users to select a room, specify check-in and check-out dates, and submit a booking request.

**Availability Check:**

The system checks the selected room’s availability for the given dates before confirming the booking, preventing double-booking and date conflicts.

**Booking Confirmation:**

Upon successful booking, the system updates the room’s status in the database and provides instant confirmation to the user.

**Booking Validation:**

Ensures that booking requests are valid (e.g., check-out date is after check-in, room is available, user is authenticated).

**5. Booking Management and Cancellation Module**

**User Booking Management:**

Users can view their current and past bookings, including reservation details such as room number, dates, and status.

**Booking Cancellation:**

Users can cancel bookings within allowed policies. The system updates the room’s availability and logs the cancellation.

**Admin Booking Oversight:**

Administrators can view all bookings, filter by date or status, and manage or cancel bookings as needed for operational reasons.

**6. Admin Dashboard Module**

**Overview Panel:**

Provides a summary of key metrics such as total rooms, active bookings, available rooms, and recent activity.

**Management Tools:**

Quick access to room management, booking management, and user management features.

**Reporting:**

Displays basic reports on occupancy rates, booking trends, and user activity. Advanced analytics can be added as a future enhancement.

**7. User Profile Management Module**

**Profile Editing:**

Allows users to update their personal information, such as name, contact details, and password.

**Account Security:**

Users can change their password and review recent login activity for security purposes.

**8. Security and Access Control Module**

**Role-Based Access Control:**

Differentiates between regular users and administrators, ensuring that only authorized personnel can access sensitive features.

**Input Validation and Sanitization:**

Protects against common security threats such as SQL injection and cross-site scripting (XSS).

**Session Security:**

Implements measures to prevent session hijacking and unauthorized access.

**9. Database Management Module**

**Data Storage:**

Utilizes a relational database (MySQL) to store all information related to users, rooms, and bookings.

**Data Integrity:**

Enforces relational integrity through foreign key constraints and transaction management.

**Backup and Recovery:**

Supports database backup and recovery procedures to prevent data loss.

**10. Reporting and Analytics Module (Optional/Future Enhancement)**

**Booking Reports:**

Generates reports on booking statistics, occupancy rates, and revenue trends.

**User Analytics:**

Analyzes user behavior, booking patterns, and system usage for business insights.

**11. Notification Module (Optional/Future Enhancement)**

**Email/SMS Notifications:**

Sends automated notifications to users regarding booking confirmations, cancellations, or special offers.

**Admin Alerts:**

Notifies administrators of important system events, such as new bookings or cancellations.

**12. Audit and Logging Module (Optional/Future Enhancement)**

**Activity Logging:**

Maintains logs of user and admin actions for security audits and troubleshooting.

**Change History:**

Tracks changes to room inventory, bookings, and user accounts.

**Integration and Workflow**

All these modules are integrated to provide a seamless workflow:

Users interact with the registration, login, room browsing, booking, and profile modules.

Administrators use the admin dashboard, room management, booking management, and reporting modules.

The security and database modules underpin the entire system, ensuring safe and reliable operations.

**Technical Overview**

**1. Technology Stack Overview:**

The Hotel Booking System is developed using a full-stack web technology approach, combining client-side and server-side technologies to deliver a seamless, interactive, and secure experience. The core technologies used are:

Frontend: HTML5, CSS3, JavaScript

Backend: PHP (Hypertext Preprocessor)

Database: MySQL (Structured Query Language)

Web Server: Apache (bundled in XAMPP)

Development Tools: Code editors such as VS Code and phpMyAdmin for database management.

**2. Frontend Architecture:**

The frontend is responsible for presenting information to users and capturing user input. It is built using:

HTML5: For semantic and well-structured markup of pages, forms, and content.

CSS3: For styling, layout, and responsiveness, ensuring the platform looks consistent across devices including desktops, laptops, tablets, and smartphones.

JavaScript: For client-side interactivity, such as form validation and dynamic content updates, improving user experience.

**3. Backend Architecture:**

The backend handles business logic, data processing, and communication with the database. Key aspects include:

PHP: Processes user requests, manages sessions, handles authentication, and generates dynamic content. PHP scripts are modular for maintainability.

Session Management: PHP sessions maintain user login states and enforce role-based access control.

Security: Passwords are securely hashed using PHP’s built-in functions. Input is validated and sanitized to prevent SQL injection and cross-site scripting (XSS).

**4. Database Design and Management:**

MySQL is used as the relational database management system. The schema supports all system requirements:

Tables:

users (user\_id, username, password\_hash, role, contact\_info)

rooms (room\_id, room\_number, type, price, status, description, image)

bookings (booking\_id, user\_id, room\_id, check\_in, check\_out, status, timestamp)

Relationships: Foreign keys maintain referential integrity between users, rooms, and bookings.

Indexes: Indexes on frequently queried fields improve query performance.

**5. Authentication and Authorization:**

User Authentication: Registration and login use secure password hashing and validation.

Role-Based Access Control: Users have roles (guest or admin) that restrict access to certain features. Only administrators can access room and booking management tools.

**6. Room and Booking Management:**

Room Management: Administrators can add, edit, and delete room records, including details like type, price, and availability.

Booking Process: Guests can search rooms, select dates, and book available rooms. The system checks availability and prevents overlapping bookings.

Booking Confirmation: Bookings are confirmed instantly, and users can view or cancel their reservations.

Cancellation: Users can cancel bookings, which updates room availability accordingly.

**7. Admin Dashboard and Management Tools:**

Dashboard: Provides an overview of current bookings, room status, and user statistics.

Management: Admins can manage rooms, bookings, and users from a centralized interface.

**8. Security Measures:**

Input Validation and Sanitization: All user inputs are validated on client and server sides.

Prepared Statements: Used for all database queries to prevent SQL injection.

Password Security: Passwords are hashed and never stored in plain text.

Session Security: Sessions are regenerated after login to prevent fixation attacks.

Error Handling: Sensitive information is hidden from users; errors are logged for developers.

**9. Deployment and Environment:**

The system is designed to run on a local server environment using XAMPP, which includes Apache, PHP, and MySQL.

Installation instructions cover setting up the environment, importing the database, and configuring the application.

**10. Modularity and Code Organization:**

The codebase is organized into modules for users, rooms, bookings, and admin functions to facilitate maintenance and scalability.

Reusable functions and components improve code clarity and reduce duplication.

**11. Testing and Quality Assurance:**

Unit testing of individual modules ensures correctness.

Integration testing verifies end-to-end workflows such as registration, booking, and cancellation.

User acceptance testing confirms usability and functionality.

Security audits check for vulnerabilities and ensure data protection.

**12. Documentation and Support:**

Comprehensive user and technical documentation is provided, including installation steps, module descriptions, and troubleshooting guides.

**Design and Development Process**

**1. Requirement Gathering and Analysis:**

The project started with an in-depth analysis of the needs of both hotel guests and administrators. Key requirements included a visually appealing landing page, seamless booking experience, secure authentication, real-time room availability, and efficient backend management. Industry research and user feedback were considered to ensure the system would be practical, intuitive, and competitive.

**2. Landing Page Planning and Design:**

Recognizing the critical importance of the landing page as the system’s entry point, the design focused on capturing visitor attention, conveying the hotel’s unique value, and guiding users toward action.

The landing page (index.html) was structured with clear navigation, compelling headlines, and high-quality visuals to highlight rooms, facilities, food, and contact information.

A prominent call-to-action (CTA), such as “Login” or “Book Now,” was placed in the navigation bar and within the modal, following best practices to maximize user engagement and conversion.

The design is responsive, ensuring optimal display and usability across desktops, tablets, and smartphones.

Visual hierarchy was established by placing the most important information—such as the hotel’s unique selling points and booking options—at the top of the page, making it immediately visible to visitors.

The modal login/signup form was integrated for quick access, reducing friction in the user journey.

**3. User Interface (UI) and User Experience (UX) Development:**

HTML5 was used to structure content semantically, while CSS3 provided styling and responsive layouts.

JavaScript enhanced interactivity, powering features like the modal, smooth scrolling, and form validation.

The landing page was paired with a dedicated stylesheet (style1.css) to maintain a distinct, welcoming look, while system pages shared a uniform theme for brand consistency.

Accessibility was prioritized, with readable fonts, clear color contrast, and keyboard navigation support.

**4. Backend Architecture and Logic:**

PHP was selected for server-side scripting due to its integration with MySQL and ease of deployment.

The backend was modularized into components for authentication, room management, booking processing, and administrative control.

Security measures included password hashing, session management, and input validation to safeguard user data.

**5. Database Schema Design:**

MySQL was used to create normalized tables for users, rooms, and bookings.

Relationships were defined using foreign keys to maintain data integrity and ensure accurate tracking of reservations and user activity.

**6. Integration of Frontend and Backend:**

The landing page’s login and signup modal was linked to backend authentication scripts, enabling seamless transition from visitor to registered user.

Forms and navigation were wired to backend endpoints for booking, user management, and admin functions.

**7. Testing and Quality Assurance:**

Each feature was tested individually (unit testing) and as part of the overall workflow (integration testing).

The landing page was evaluated for fast load times, mobile compatibility, and clear CTAs, aligning with industry standards for hotel websites.

User acceptance testing involved collecting feedback to refine usability and address any issues.

**8. Deployment and Configuration:**

The project was deployed on a local XAMPP environment for development and testing.

Configuration included setting up the database, adjusting PHP settings, and organizing project files for easy maintenance.

**9. Documentation and Continuous Improvement:**

Comprehensive documentation was created, including system architecture, installation steps, and user guides.

The landing page’s role as the system’s first impression and conversion driver was highlighted, with references to best practices in hotel website design.

**Benefits and Applications**

**Benefits of the Project**

The Hotel Booking System offers numerous advantages that benefit both hotel management and guests, transforming traditional booking processes into efficient, user-friendly, and revenue-optimizing operations.

**1. 24/7 Accessibility and Convenience**

The system enables guests to book rooms anytime and from anywhere with an internet connection, eliminating the constraints of office hours and physical presence. This constant availability increases booking opportunities, caters to international travelers across time zones, and enhances overall customer satisfaction.

**2. Real-Time Room Availability and Updates**

By providing instant updates on room availability, the system prevents double bookings and reservation conflicts. Guests see accurate, up-to-date information, which builds trust and reduces administrative overhead caused by manual corrections.

**3. Streamlined Booking Process**

The automated booking workflow simplifies the reservation experience for guests, allowing them to search, select, and confirm rooms quickly. This reduces the time and effort compared to traditional phone or in-person bookings, enhancing user experience and encouraging repeat business.

**4. Reduced Administrative Workload**

Hotel staff benefit from automation of repetitive tasks such as booking management, cancellations, and room inventory updates. This frees up resources to focus on guest services and operational improvements, increasing overall productivity.

**5. Improved Accuracy and Error Reduction**

Automation minimizes human errors common in manual booking systems, such as overbooking or incorrect record keeping. The system enforces validation and consistency checks, ensuring reliable reservation management and reducing guest dissatisfaction.

**6. Centralized Management and Data Access**

The system consolidates guest information, booking records, and room data into a single database. This centralization facilitates efficient management, quick retrieval of information, and better coordination among hotel departments like front desk, housekeeping, and management.

**7. Revenue Optimization and Dynamic Pricing**

Hotels can adjust room rates dynamically based on demand, seasonality, or special events. This flexibility helps maximize revenue during peak periods and attract bookings during off-peak times with competitive pricing strategies.

**8. Enhanced Guest Experience**

By providing features such as instant booking confirmation, easy cancellation, and personalized services based on guest preferences, the system elevates the overall guest experience. Access to detailed room information and seamless communication further contribute to customer satisfaction.

**9. Marketing and Promotional Opportunities**

The platform allows hotels to promote special offers, packages, and loyalty programs directly to guests. This targeted marketing increases engagement and helps build long-term customer relationships.

**10. Cost Savings and Increased Profit Margins**

Direct booking through the system reduces reliance on third-party travel agencies, lowering commission fees. Automation also decreases operational costs by minimizing manual labor and errors, contributing to higher profitability.

**11. Scalability and Future Growth**

The modular design of the system supports easy integration of additional features such as payment gateways or customer feedback modules. This scalability ensures the hotel can adapt to changing business needs and technological advancements without major overhauls

**Applications of the Project**

**1. Hotel Room Booking and Reservation:**

The system enables guests to conveniently search for available rooms and make reservations online at any time. Real-time updates ensure that room availability is always accurate, reducing the risk of double-booking. Instant booking confirmations provide assurance to guests. This application streamlines the entire reservation process for both users and hotel staff.

**2. Centralized Guest and Booking Management:**

Hotel administrators can manage all guest profiles, booking records, and room statuses from a single dashboard. This centralization eliminates the need for manual record-keeping and scattered information. It allows for quick access to guest details and booking histories. As a result, hotel operations become more organized and efficient.

**3. Automation of Administrative Tasks:**

The system automates repetitive administrative functions such as updating room status, sending booking confirmations, and processing cancellations. This reduces the workload on hotel staff and minimizes human error. Automated processes also speed up service delivery. Staff can then dedicate more time to enhancing guest experiences.

**4. Flexible Stay Management:**

The platform supports various booking durations, including hourly, daily, or extended stays. This flexibility allows hotels to cater to different types of guests, from business travelers to vacationers. Room pricing and availability can be adjusted based on the length of stay. It helps maximize occupancy and revenue for the hotel.

**5. Enhanced Guest Experience:**

Guests benefit from a user-friendly interface that provides detailed information about rooms, facilities, and pricing. The ability to book, modify, or cancel reservations online adds convenience and transparency. Prompt notifications and confirmations improve guest confidence. Overall, the system fosters a positive and memorable stay experience.

**6. Data Collection and Analytics:**

The system collects valuable data on bookings, guest preferences, and occupancy trends. Hotel management can analyze this data to make informed decisions about pricing, promotions, and resource allocation. Analytics help identify peak seasons and guest demographics. This data-driven approach supports strategic planning and business growth.

**7. Revenue and Rate Management:**

Hotels can use the system to implement dynamic pricing strategies, adjusting room rates based on demand, season, or special events. This helps optimize occupancy and maximize revenue. The platform also allows for the easy creation of promotional offers. Automated rate management reduces errors and enhances profitability.

**8. Multi-Channel Integration:**

The system can be integrated with other platforms such as property management systems (PMS) and online travel agencies (OTAs). This ensures that inventory and bookings are synchronized across all channels. It prevents overbooking and expands the hotel’s market reach. Centralized integration simplifies overall management.

**9. Customer Relationship Management (CRM):**

By maintaining detailed guest profiles and booking histories, the system enables hotels to offer personalized services and targeted promotions. CRM features help build stronger relationships with guests. Loyalty programs and special offers can be managed more effectively. This leads to increased guest retention and satisfaction.

**10. Feedback and Review Collection:**

After their stay, guests can provide feedback or reviews through the system. This feature helps hotels monitor service quality and identify areas for improvement. Positive reviews can be showcased to attract new customers. Continuous feedback collection supports ongoing service enhancement.

**11. Secure Payment Processing:**

The platform supports secure online payment options, allowing guests to pay for their bookings safely and conveniently. Integration with trusted payment gateways ensures data protection and fraud prevention. This builds trust with guests and streamlines the check-in process. Multiple payment methods cater to diverse customer preferences.

**12. Paperless Operations:**

By digitizing all records and processes, the system reduces the need for paper-based documentation. This not only saves physical storage space but also supports environmental sustainability. Digital records are easier to manage, search, and back up. Paperless operations enhance efficiency and compliance.

**Definition of Problem**

**Inefficiency of Traditional Booking Methods:**

Many hotels, especially small and medium-sized establishments, still rely on manual or semi-automated processes for room reservations. These methods often involve phone calls, emails, or in-person interactions, which are time-consuming, prone to human error, and inefficient for both guests and staff.

**Limited Accessibility and Convenience:**

Traditional booking systems restrict guests to making reservations only during business hours and often require direct communication with hotel staff. This lack of 24/7 accessibility can result in lost business opportunities and inconvenience for guests who wish to book rooms outside regular working hours or from remote locations.

**Risk of Double Booking and Data Inconsistency:**

Manual record-keeping and fragmented systems increase the risk of double booking, where two guests are assigned the same room for overlapping dates. Such errors can lead to dissatisfied customers, operational challenges, and potential financial losses for the hotel.

**Cumbersome Cancellation and Modification Processes:**

In the absence of an automated system, canceling or modifying a booking can be a complicated process that requires multiple communications between guests and hotel staff. This not only frustrates customers but also increases the administrative workload.

**Lack of Centralized Management and Oversight:**

Without a unified platform, hotel administrators struggle to maintain an up-to-date view of room availability, booking status, and customer information. This fragmentation makes it difficult to manage resources efficiently, track performance, or generate useful business insights.

**Security and Data Privacy Concerns:**

Storing sensitive customer information in paper registers or unsecured digital files exposes both the hotel and its guests to potential data breaches. Manual systems are often not equipped with adequate security measures, making them vulnerable to unauthorized access and misuse of information.

**Difficulty in Scaling Operations:**

As a hotel grows, the limitations of manual or basic booking systems become more apparent. Managing a larger inventory of rooms, handling increased booking volumes, and accommodating more users become challenging without an automated, scalable solution.

**Lack of Integration with Modern Technologies:**

Many existing systems do not integrate with online payment gateways, customer feedback tools, or analytics platforms. This limits the hotel’s ability to offer a seamless, modern guest experience and to make data-driven business decisions.

**Inadequate Reporting and Analytics:**

Manual systems rarely provide comprehensive reporting or analytics features. This makes it difficult for hotel management to analyze booking trends, occupancy rates, customer preferences, or operational bottlenecks, hindering strategic planning and growth.

**Customer Dissatisfaction and Lost Revenue:**

The combined effect of slow, error-prone, and inconvenient booking processes is lower customer satisfaction. Dissatisfied guests are less likely to return or recommend the hotel, resulting in lost revenue and a diminished reputation.

**Need for an Educational and Demonstrative Solution:**

There is also a need for a practical, real-world project that demonstrates the application of web development and database management skills. Such a project can serve as a learning tool for students and as a prototype for hotels seeking to digitize their operations.

**Problem Statement**

**Inefficiency of Manual Booking:**

Many hotels still use manual reservation methods, leading to delays and frequent errors. This slows down operations and frustrates both guests and staff.

**Limited Guest Accessibility:**

Guests can only book during office hours, which restricts convenience and may cause lost bookings, especially for international customers.

**Risk of Double Booking:**

Without a centralized system, overlapping reservations and inaccurate room availability often occur, causing operational issues and guest dissatisfaction.

**Complicated Cancellation Process:**

Cancelling or modifying bookings requires manual intervention, which is time-consuming and prone to mistakes.

**Lack of Centralized Data:**

Fragmented records make it difficult to track bookings, generate reports, and make data-driven decisions.

**Security Concerns:**

Sensitive guest data stored manually or insecurely is vulnerable to loss or unauthorized access.

**Scaling Challenges:**

Manual systems become inefficient as the hotel grows, limiting the ability to handle more bookings effectively.

**Problem Analysis**

**Manual Processes Cause Inefficiency:**

Traditional booking methods require significant time and effort from both staff and guests. This leads to slow service, frequent errors, and increased operational costs, making it hard to deliver a smooth guest experience.

**Limited Accessibility Reduces Bookings:**

Without an online system, guests can only book during hotel working hours, which restricts convenience and may result in lost business, especially from international or last-minute travelers.

**Data Inconsistency and Double Bookings:**

When records are kept manually or in separate files, there is a high risk of overlapping reservations and inaccurate room status. This can result in guest dissatisfaction and financial losses for the hotel.

**Fragmented Data Hinders Management:**

Without a centralized database, it is challenging to generate accurate reports, analyze trends, or make informed business decisions. Fragmented data also complicates day-to-day management and planning.

**Security and Privacy Risks:**

Storing sensitive guest information in paper files or unsecured digital formats exposes the hotel to risks of data loss, breaches, and non-compliance with privacy regulations.

**Scalability Issues:**

As the number of rooms and guests increases, manual systems become harder to manage, leading to more errors and inefficiencies. This limits the hotel’s ability to grow and remain competitive.

**Logic of the Project**

**1. User Registration and Authentication**

**Registration Process:**

When a new user visits the hotel booking system, they have the option to register for an account. The registration form collects essential information such as the user’s full name, email address, and a password. The system first checks if the provided email address already exists in the database to prevent duplicate accounts. If the email is unique and all required fields are filled correctly, the password is securely hashed using a modern cryptographic algorithm before storing it in the database. This ensures that even if the database is compromised, user passwords remain protected. Upon successful registration, the user receives a confirmation message and can proceed to log in.

**Login Process:**

For authentication, the user enters their registered email and password. The system retrieves the corresponding hashed password from the database and verifies it using a secure comparison function. If the credentials match, a session is created for the user, which is maintained throughout their interaction with the system. This session management ensures that only authenticated users can access their personal dashboard, view bookings, or make reservations. If authentication fails, the user is prompted with an error message and given the option to retry.

**Session Security:**

Sessions are managed using secure, unique session IDs. After successful login, the session ID is regenerated to prevent session fixation attacks. Sessions are also set to expire after a period of inactivity to reduce the risk of unauthorized access.

**2. Room Browsing and Selection**

**Room Listing and Filtering:**

The landing page and dedicated “Rooms” section display all available rooms, each with details such as room type, price per night, amenities, and images. Users can browse through the list or apply filters to narrow down options based on their preferences, such as price range, room type, or specific amenities like Wi-Fi or a swimming pool.

**Room Selection:**

Once a user finds a suitable room, they select it and are prompted to enter their desired check-in and check-out dates. This information is crucial for the next step, which involves checking the room’s availability for the specified period.

**3. Room Availability and Booking Logic**

**Availability Check Algorithm:**

The core logic for booking revolves around ensuring that no two bookings for the same room overlap in time. When a user submits their desired dates, the system queries the database for all existing bookings for that room. It checks whether the requested check-in and check-out dates overlap with any existing reservations.

If there is no overlap, the room is considered available, and the booking process can proceed.

If there is an overlap, the system informs the user that the room is unavailable for the selected dates and prompts them to choose different dates or another room.

**Booking Confirmation:**

If the room is available, the system creates a new booking record in the database. This record includes the user’s ID, room ID, check-in and check-out dates, and booking status (e.g., confirmed, pending, or cancelled). The room’s status is updated as necessary to reflect its occupancy during the booked period. A confirmation message is displayed to the user, and an optional email notification can be sent.

**Atomic Transactions:**

To maintain data integrity, especially in high-traffic environments, the booking and room status update are performed as a single, atomic transaction. This ensures that either both actions succeed or neither does, preventing inconsistencies such as double bookings.

**4. Booking Management (View, Cancel, Modify)**

**Viewing Bookings:**

Authenticated users can access their personal dashboard to view a list of all their current and past bookings. The system retrieves this information from the database and displays it in an organized manner, showing details such as room type, stay dates, booking status, and total cost.

**Cancellation Process:**

If a user wishes to cancel a booking, they can do so from their dashboard. The system updates the booking status in the database to “cancelled” and releases the room for those dates, making it available for other guests. A cancellation confirmation is shown to the user, and the system may also send an email notification.

**Modification Process:**

If the system allows modifications, users can request to change their booking dates or room selection. The system re-runs the availability check for the new dates or room. If available, the booking record is updated; if not, the user is prompted to select alternative options.

**5. Administrative Functions**

**Admin Authentication:**

Administrators log in using special credentials with elevated privileges. Their sessions are managed separately to ensure that only authorized personnel can access sensitive management features.

**Room Management:**

Admins can add new rooms, edit existing room details, or remove rooms from the system. When adding or editing a room, the admin specifies details such as room number, type, price, amenities, and uploads images. All changes are immediately reflected in the system and made available to users.

**Booking Oversight:**

Administrators have access to all bookings in the system. They can filter bookings by date, status, or guest, and can intervene if necessary (e.g., force-cancel a booking or resolve conflicts).

**User Management:**

Admins can view all registered users, manage user accounts, and address issues such as duplicate registrations or misuse of the system.

**6. Security and Data Integrity**

**Input Validation and Sanitization:**

All user inputs are validated on both the client side (using JavaScript) and server side (using PHP) to prevent invalid data entry and protect against security threats like SQL injection and cross-site scripting (XSS).

**Password Security:**

Passwords are never stored in plain text. They are hashed using secure algorithms, and password verification is done using proper cryptographic functions.

**Session Management:**

Sessions are securely managed, with session IDs regenerated after login and set to expire after inactivity. Sensitive operations require re-authentication or session validation.

**Error Handling:**

The system handles errors gracefully, providing informative messages to users without exposing sensitive system details. All critical errors are logged for administrative review.

**7. Notifications and Feedback**

**Booking and Cancellation Confirmations:**

Users receive immediate feedback on the status of their bookings or cancellations. The system displays confirmation messages and can send notifications via email for important actions.

**User Feedback:**

After their stay, users may be prompted to provide feedback or reviews, which can be used by hotel management to improve services.

**8. Data Analytics and Reporting**

**Data Collection:**

The system logs all bookings, user activity, and room occupancy data. This information is stored securely and can be used for generating reports.

**Reporting:**

Administrators can generate reports on occupancy rates, booking trends, and revenue, helping them make informed business decisions and optimize hotel operations.

**9. Integration and Scalability**

**Modular Design:**

The system is designed in a modular way, allowing for easy integration of new features such as payment gateways, third-party booking platforms, or advanced analytics tools.

**Scalability:**

The database and application logic are structured to handle increased numbers of rooms, users, and bookings as the hotel grows.

**Error Handling**

**1. Philosophy and Objectives**

Error handling in the Hotel Booking System is designed to ensure the application remains robust, user-friendly, and secure even when unexpected situations arise. The primary objectives are to:

Prevent system crashes and data corruption.

Provide clear, actionable feedback to users.

Maintain data security and privacy.

Enable administrators and developers to identify and resolve issues efficiently.

Preserve a seamless user experience, even in the face of errors.

**2. Types of Errors Addressed**

**a. User Input Errors**

Examples: Missing required fields, invalid email formats, password mismatches, invalid date ranges (e.g., check-out before check-in), or attempting to book for past dates.

Impact: Can lead to failed bookings, user frustration, or inconsistent data if not properly handled.

**b. Authentication and Authorization Errors**

Examples: Incorrect login credentials, expired sessions, unauthorized access to admin features, or attempts to perform restricted actions.

Impact: Can compromise security or disrupt user workflow.

**c. Database and Backend Errors**

Examples: Database connection failures, failed queries, constraint violations (e.g., duplicate entries), or attempts to access non-existent records.

Impact: May result in data loss, incomplete transactions, or system downtime.

**3. Error Handling Implementation Strategies**

**a. Input Validation and Sanitization**

All user inputs are validated on the client side (using JavaScript) for immediate feedback and on the server side (using PHP) for security.

Validation includes checking for required fields, correct data formats, logical date ranges, and password strength.

Sanitization removes or escapes potentially dangerous input to prevent attacks such as SQL injection or XSS.

**b. Use of Try-Catch Blocks and Exception Handling**

Critical operations (such as database queries, file uploads, and session management) are wrapped in try-catch blocks.

When exceptions are caught, the system logs detailed error information (timestamp, user, operation, stack trace) in a secure location for admin review.

Users receive a generic, non-technical error message to avoid exposing sensitive system details.

**c. Graceful Degradation and Fallbacks**

If a non-critical feature fails (e.g., sending a confirmation email), the core booking process continues, and the user is informed of the partial failure.

The system always provides a way to return to a safe state, such as the home page or previous step.

**d. User-Friendly Error Messages**

Error messages are clear, concise, and instructive (e.g., “Please enter a valid email address” or “Selected room is not available for these dates”).

The system highlights erroneous fields and, where possible, suggests corrective actions.

**e. Logging and Monitoring**

All critical errors are logged with detailed context, including user actions, input data, and system state.

Logs are accessible only to authorized administrators for debugging and auditing.

System health and error rates are monitored, and alerts are sent to admins if thresholds are exceeded.

**f. Security-Focused Handling**

Error messages never reveal sensitive information such as database structure, file paths, or system internals.

All failed login attempts are monitored, and repeated failures may trigger account lockout or CAPTCHA to prevent brute-force attacks.

Session management ensures that expired or invalid sessions are promptly invalidated.

**4. User Experience in Error Handling**

**Immediate Feedback:**

Users receive instant feedback on form errors before submission (client-side) and after submission (server-side), reducing frustration.

**Guided Recovery:**

After an error, users are offered options to retry, correct input, or contact support. Navigation aids (e.g., “Return to Home” or “Try Again” buttons) are provided.

**Non-Functional Requirements:**

**1. Security:**

The system must enforce secure login with unique credentials for all user levels (guests, staff, managers, owners). Sensitive data, such as passwords and payment information, must be encrypted both in transit and at rest. Only authorized users should access specific system functions, and all external communications must use secure protocols (e.g., HTTPS).

**2. Performance:**

The user interface should load within 2 seconds, and database queries must return results within 5 seconds. Login validation and booking confirmation should be processed within 3 seconds. The system must handle at least 20 transactions during peak hours without significant delays.

**3. Reliability and Availability:**

The system should be available 24/7, with minimal downtime. In case of system failure, recovery should occur within one hour, and the database should be backed up at least every hour to prevent data loss.

**4. Scalability:**

The system must be able to accommodate growth in the number of users, rooms, and transactions. It should support up to 10,000 registered members and be capable of scaling to handle increased load as the hotel expands.

**5. Usability:**

The interface should be intuitive and easy to use for all types of users, including guests and administrators. A user manual or help section should be provided to assist users in navigating the system. Error messages must be clear and guide users to resolve issues.

**6. Maintainability:**

The system should be designed for easy maintenance and updates. Code should be well-documented, and the architecture should allow for the addition of new features or changes with minimal disruption.

**7. Data Integrity:**

All data entered into the system should be validated to prevent inconsistencies or corruption. The system must ensure that booking and cancellation operations are atomic and consistent, preventing double bookings or data loss.

**8. Portability:**

The software should run on any standard Microsoft Windows environment and be compatible with common web browsers to ensure accessibility for all users.

**9. Safety:**

The system must have measures in place to prevent unauthorized access and protect against data breaches. Different user levels should have access only to the functions relevant to their roles, enforced through login credentials and access controls.

**10. Testability:**

The system should be testable, allowing for verification of all functional and non-functional requirements. Automated and manual testing should be possible to confirm performance, security, and correctness.

**System Configuration**

|  |  |
| --- | --- |
| **Operating System** | Windows 11, Windows XP, Windows7 etc. |
| **Language** | HTML, CSS, PHP |
| **Database** | MySql |
| **Browser** | Any of Mozilla, Opera, Chrome etc |

|  |  |
| --- | --- |
| Processor | AMD Ryzen, Pentium III 500MHz or more |
| RAM | 255-512 MB |
| Hard disk | 105.1 GB |
| Monitor | 32-Bit Color Monitor |
| Keyboard | PS2 or USB |
| Mouse | PS2 or USB |
| Additional Software | VS Code , XAMPP |

**Block Diagram**

USER

WEB INTERFACE (FRONTEND)

BACKEND (PHP)

DATABASE SERVER

ADMIN INTERFACE

**DFD DIAGRAM**

Manage Booking

Customer

Room Data

User Data

Manage Profile

Registration

Cancellation

Booking Data

Cancel Booking

**ER DIAGRAM**

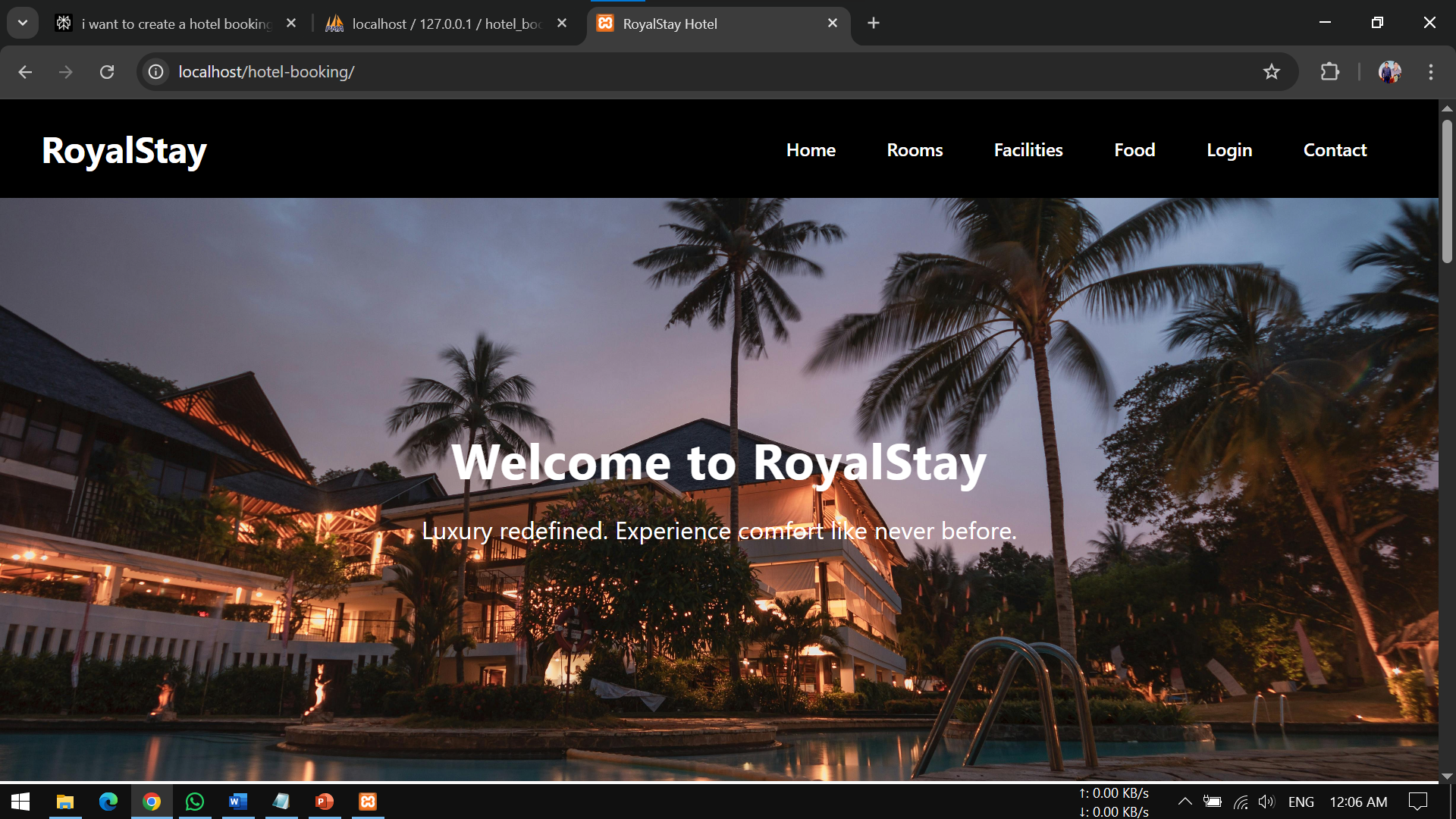
DATABASE

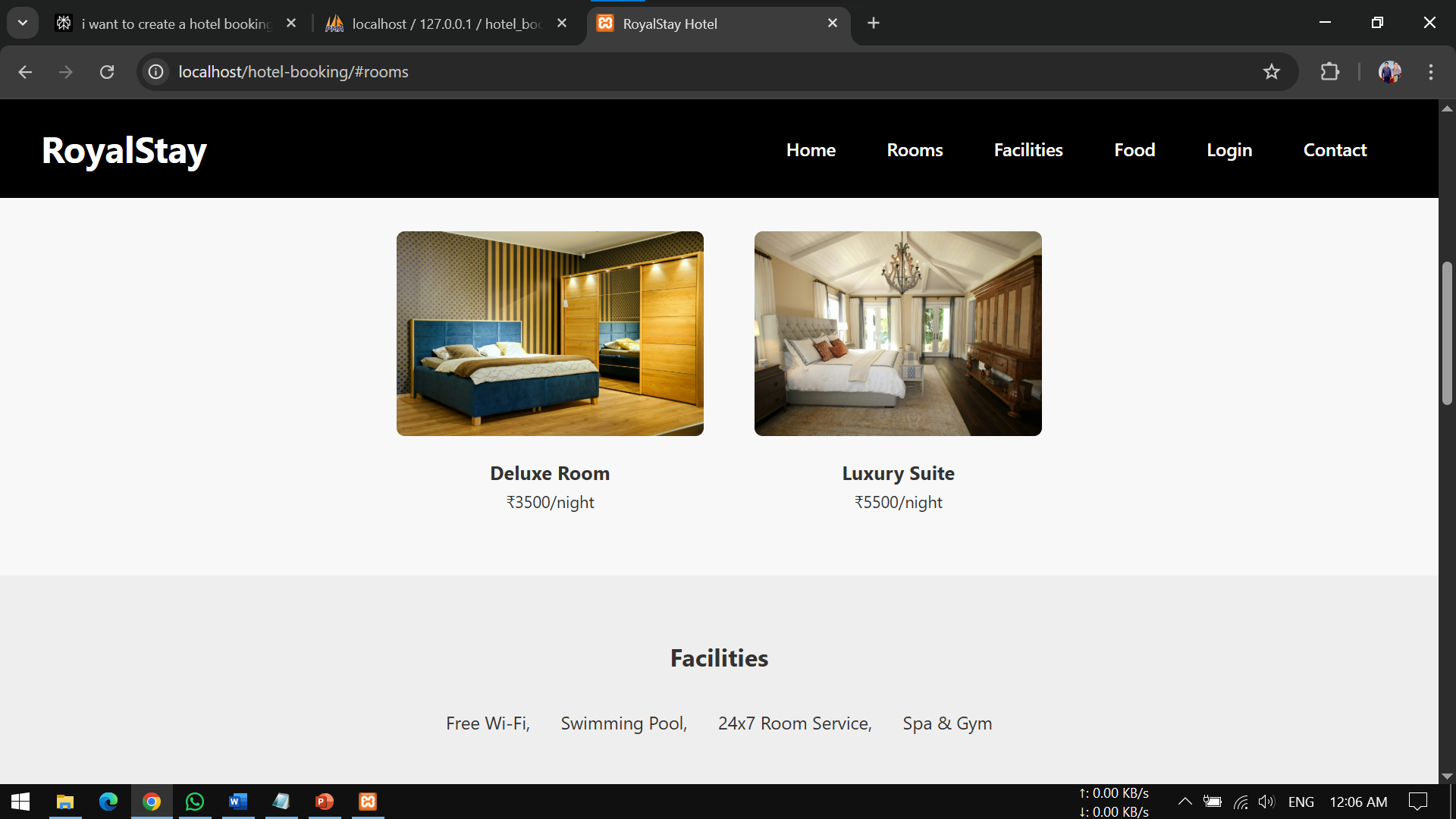
USERS

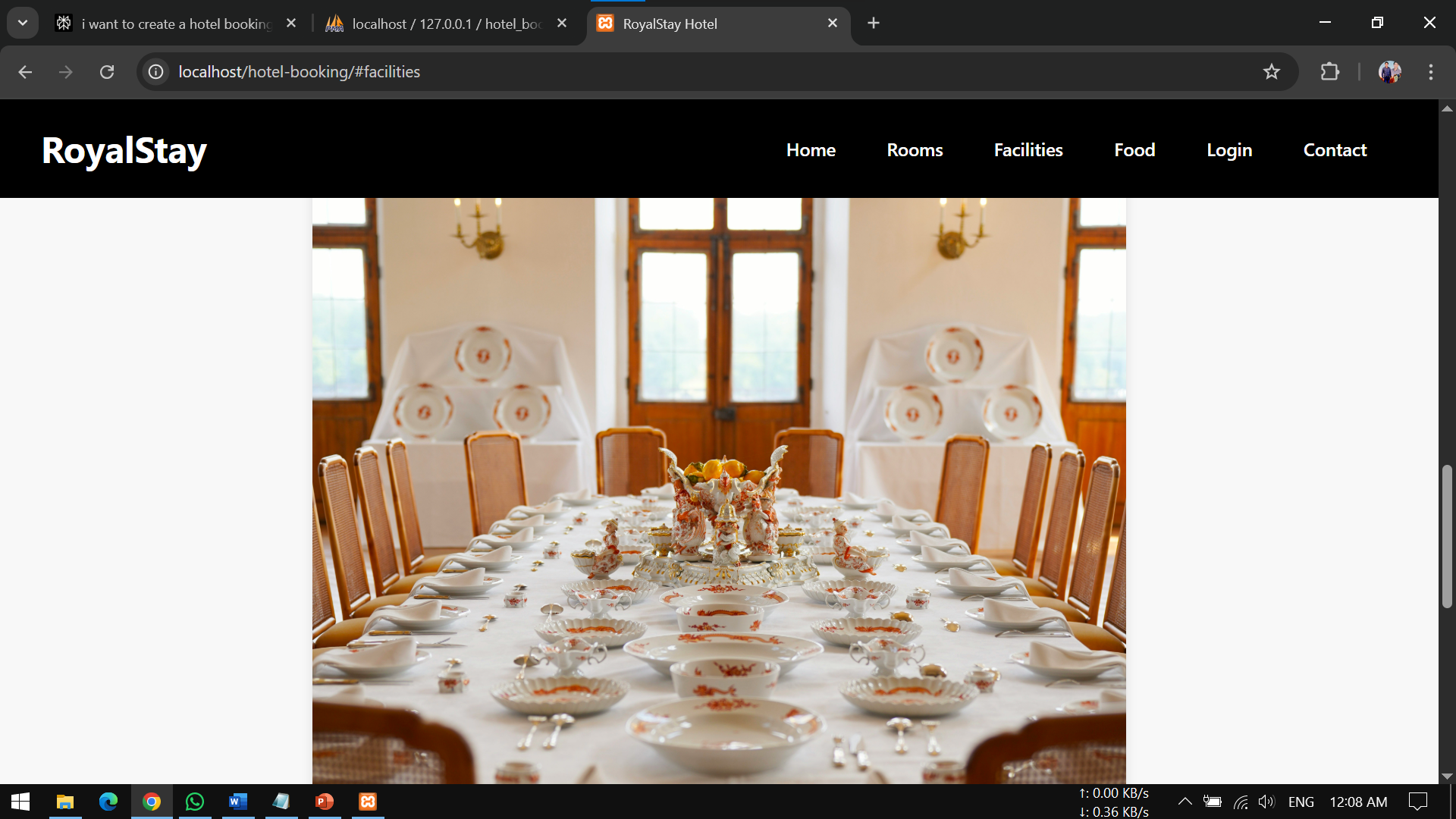
ROOMS

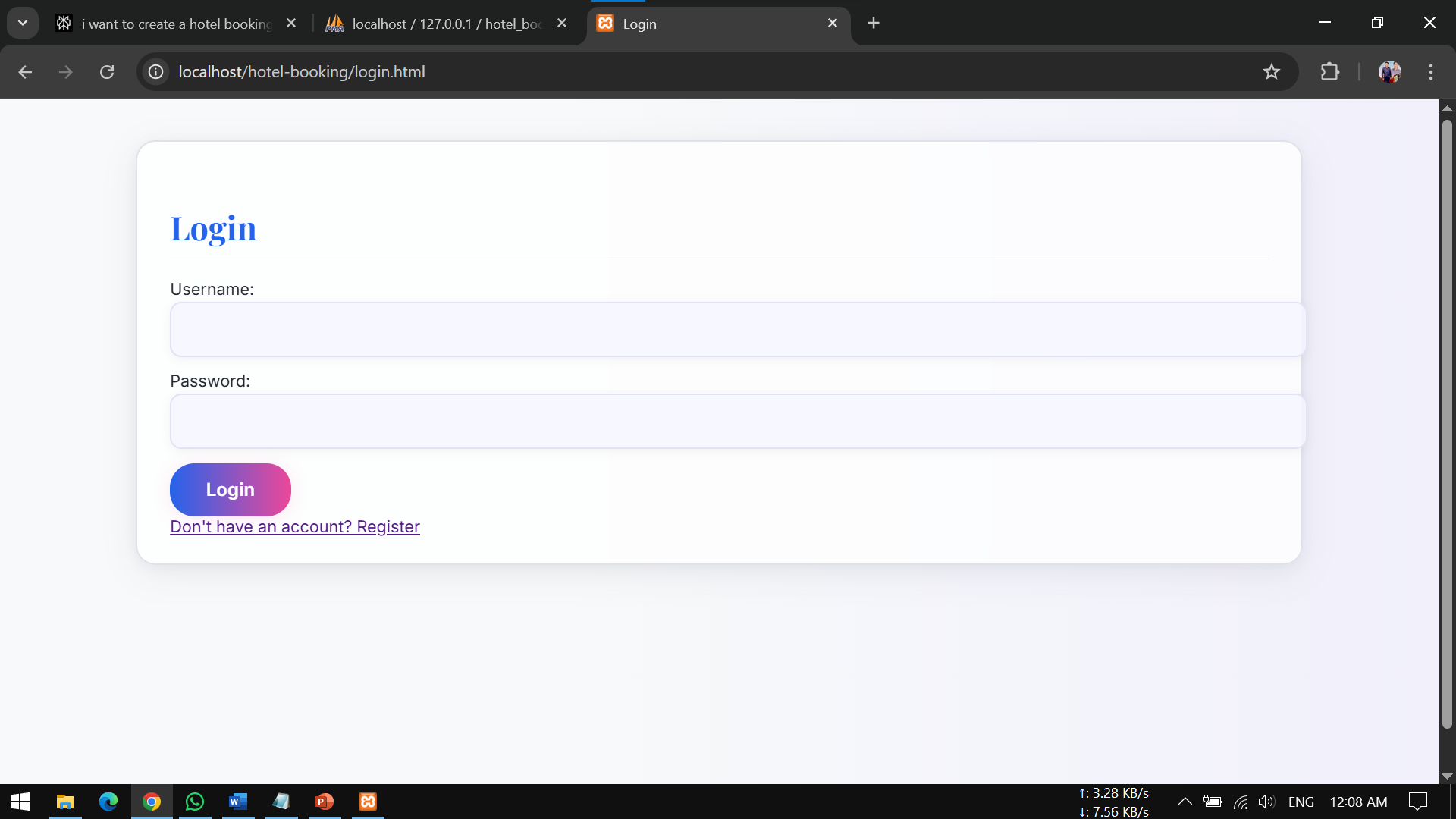
BOOK

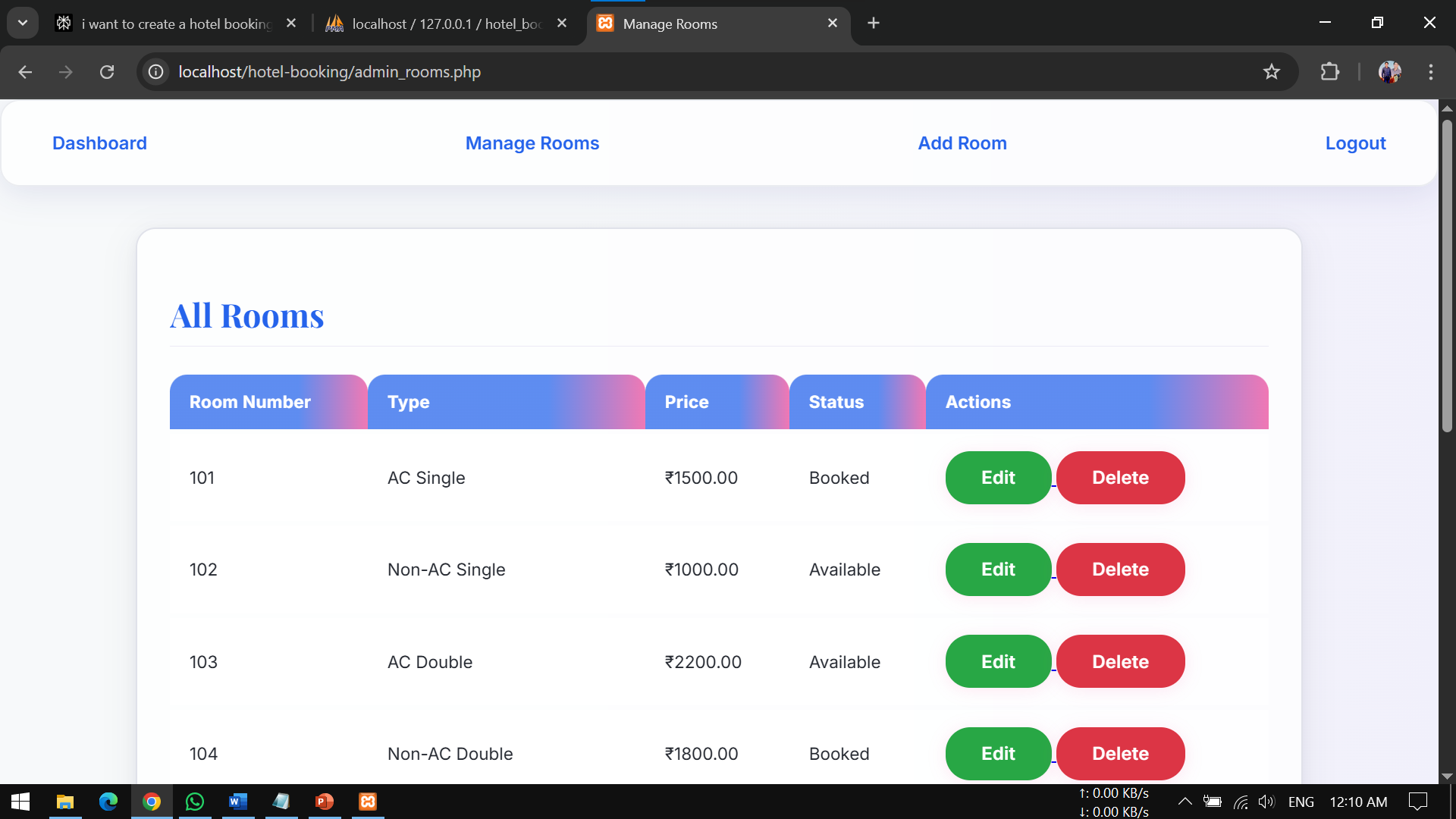
**Output Screen**









****

**CODE**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>RoyalStay Hotel</title>

<link rel="stylesheet" href="style1.css" />

</head>

<body>

<header>

<nav>

<div class="logo">RoyalStay</div>

<ul class="nav-links">

<li><a href="#home">Home</a></li>

<li><a href="#rooms">Rooms</a></li>

<li><a href="#facilities">Facilities</a></li>

<li><a href="#food">Food</a></li>

<li><span class="login-link" onclick="openAuth()"><a href="http://localhost/hotel-booking/login.html">Login</a></span></li>

<li><a href="#contact">Contact</a></li>

</ul>

</nav>

</header>

<section id="home" class="hero">

<div class="hero-text">

<h1>Welcome to RoyalStay</h1>

<p>Luxury redefined. Experience comfort like never before.</p>

</div>

</section>

<section id="rooms" class="section">

<h2>Our Rooms</h2>

<div class="cards">

<div class="card1">

<img src="images/room1.jpg" alt="Deluxe Room" width="100px" height="100px">

<h3>Deluxe Room</h3>

<p>₹3500/night</p>

</div>

<div class="card2">

<img src="images/room2.jpg" alt="Luxury Suite" width="100px" height="100px">

<h3>Luxury Suite</h3>

<p>₹5500/night</p>

</div>

</div>

</section>

<section id="facilities" class="section alt">

<h2>Facilities</h2>

<ul class="features">

<li>Free Wi-Fi,</li>

<li>Swimming Pool,</li>

<li>24x7 Room Service,</li>

<li>Spa & Gym</li>

</ul>

</section>

<section id="food" class="section">

<h2>Our Food</h2>

<p>Enjoy world-class dining from our multi-cuisine restaurant.</p>

<img src="images/food.jpg" alt="Delicious Food" class="food-img">

</section>

<div class="modal-overlay" id="authOverlay">

<div class="auth-modal">

<span class="close-btn" onclick="closeAuth()">&times;</span>

<div id="loginBox" class="loginBox">

<h2>Login</h2>

<input type="email" id="loginEmail" placeholder="Email">

<input type="password" id="loginPassword" placeholder="Password">

<div id="loginError" style="color:red; display:none;">Invalid email or password.</div>

<button onclick="validateLoginForm()">Login</button>

<p>Don't have an account? <a href="#" onclick="toggleAuth()">Sign Up</a></p>

</div>

<div id="signupBox" class="loginBox hidden">

<h2>Sign Up</h2>

<input type="text" id="signupFullName" placeholder="Full Name">

<input type="email" id="signupEmail" placeholder="Email">

<input type="password" id="signupPassword" placeholder="Password">

<input type="password" id="signupConfirmPassword" placeholder="Confirm Password">

<button onclick="validateSignUpForm()">Sign Up</button>

<p>Already have an account? <a href="#" onclick="toggleAuth()">Login</a></p>

</div>

</div>

</div>

<section id="contact" class="section alt">

<h2>Contact Us</h2>

<p>Email: himanshu@gmail.com | Phone: +91-9638524105</p>

<p>Email: Ayush234@gmail.com | Phone: +91-9876543210</p>

</section>

<footer>

<p>&copy; 2025 RoyalStay Hotel. All rights reserved.</p>

</footer>

</div>

</div>

<script src="script.js"></script>

</body>

</html>

Login.php

<?php

include 'db.php';

$username = $\_POST['username'];

$password = $\_POST['password'];

$sql = "SELECT \* FROM users WHERE username=?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param('s', $username);

$stmt->execute();

$result = $stmt->get\_result();

$user = $result->fetch\_assoc();

if ($user && password\_verify($password, $user['password'])) {

session\_start();

$\_SESSION['user\_id'] = $user['id'];

$\_SESSION['role'] = $user['role'];

if ($user['role'] == 'admin') {

header('Location: admin\_dashboard.php');

} else {

header('Location: rooms.php');

}

exit();

} else {

echo "Invalid username or password.";

}

$stmt->close();

$conn->close();

?>

Register.php

<?php

include 'db.php';

$username = $\_POST['username'];

$password = password\_hash($\_POST['password'], PASSWORD\_DEFAULT);

$sql = "INSERT INTO users (username, password) VALUES (?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param('ss', $username, $password);

if ($stmt->execute()) {

echo "Registration successful!";

} else {

echo "Error: " . $stmt->error;

}

$stmt->close();

$conn->close();

?>

Mybookings.php

<?php

session\_start();

include 'db.php';

if (!isset($\_SESSION['user\_id'])) {

header('Location: login.html');

exit();

}

$user\_id = $\_SESSION['user\_id'];

$sql = "SELECT b.id, r.room\_number, r.type, b.check\_in, b.check\_out, b.status

FROM bookings b

JOIN rooms r ON b.room\_id = r.id

WHERE b.user\_id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param('i', $user\_id);

$stmt->execute();

$result = $stmt->get\_result();

?>

<!DOCTYPE html>

<html>

<head>

<title>My Bookings</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="navbar">

<a href="rooms.php">Rooms</a>

<a href="my\_bookings.php">My Bookings</a>

<a href="logout.php">Logout</a>

</div>

<div class="container">

<div class="card">

<h2>My Bookings</h2>

<table>

<tr>

<th>Room Number</th>

<th>Type</th>

<th>Check-in</th>

<th>Check-out</th>

<th>Status</th>

<th>Cancel</th>

</tr>

<?php while($row = $result->fetch\_assoc()): ?>

<tr>

<td><?= $row['room\_number'] ?></td>

<td><?= $row['type'] ?></td>

<td><?= $row['check\_in'] ?></td>

<td><?= $row['check\_out'] ?></td>

<td><?= $row['status'] ?></td>

<td>

<?php if($row['status'] == 'booked'): ?>

<form action="cancel\_booking.php" method="POST">

<input type="hidden" name="booking\_id" value="<?= $row['id'] ?>">

<button type="submit">Cancel</button>

</form>

<?php else: ?>

N/A

<?php endif; ?>

</td>

</tr>

<?php endwhile; ?>

</table></div></div></body></html>

**Project Explanation**

**1. Database Connection Functions**

Purpose:

To establish a secure connection to the MySQL database so that all modules can read and write data.

How it works:

Usually, a function like connectDatabase() or a config.php file holds the connection logic using mysqli\_connect() or PDO.

Why it’s important:

Centralizes database access, making it easy to update credentials and maintain security.

**2. User Authentication Functions**

registerUser($name, $email, $password):

Validates and sanitizes input.

Hashes the password (using password\_hash()).

Inserts a new user into the users table.

loginUser($email, $password):

Fetches the user by email.

Verifies password using password\_verify().

Starts a session for the user.

logoutUser():

Destroys the session and logs the user out.

Why important:

Ensures only authorized users can access booking features and keeps user data secure.

**3. Room Management Functions**

getAvailableRooms($checkIn, $checkOut):

Queries the database for rooms not booked between the given dates.

addRoom($details):

Allows admins to add new rooms to the inventory.

editRoom($roomId, $newDetails):

Updates room information.

deleteRoom($roomId):

Removes a room, after checking for existing bookings.

Why important:

Keeps the room inventory accurate and up-to-date for both guests and admins.

**4. Booking Functions**

bookRoom($userId, $roomId, $checkIn, $checkOut):

Checks if the room is available for the given dates.

If yes, inserts a new record into the bookings table.

cancelBooking($bookingId, $userId):

Updates the booking status to "cancelled" if the user is authorized.

getUserBookings($userId):

Retrieves all bookings for a specific user.

getAllBookings():

For admins to view all bookings in the system.

Why important:

Handles the core business logic of making, viewing, and cancelling reservations.

**5. Payment Functions (if implemented)**

processPayment($bookingId, $paymentDetails):

Integrates with a payment gateway or simulates payment.

Updates the booking status to "confirmed" upon success.

getPaymentStatus($bookingId):

Checks if payment has been completed for a booking.

Why important:

Ensures bookings are only finalized after successful payment, preventing revenue loss.

**6. Admin Functions**

getAllUsers():

Retrieves all registered users for management.

updateUserRole($userId, $role):

Changes user privileges (e.g., promote to admin).

generateReport($criteria):

Aggregates data for occupancy, revenue, or booking trends.

Why important:

Gives hotel staff control over the business and supports decision-making.

**7. Utility and Helper Functions**

validateInput($data):

Cleans and checks user input to prevent SQL injection/XSS.

sendEmail($to, $subject, $message):

Sends confirmation or notification emails to users.

formatDate($date):

Converts dates to a standard format for display or database storage.

Why important:

Improves code reusability, security, and maintainability.

**8. Session Management**

startSession():

Initializes PHP sessions at the start of each script.

checkSession():

Ensures the user is logged in before accessing certain pages.

destroySession():

Logs the user out and clears session data.

Why important:

Maintains user state and protects restricted areas of the system.

**9. Frontend Elements (HTML/JS/PHP Mix)**

Forms:

For registration, login, booking, and admin tasks.

Tables/Lists:

To display rooms, bookings, and user information.

AJAX Calls (optional):

For dynamic updates without reloading the page (e.g., live room availability).

**10. Database Tables (as referenced by functions)**

users:

Stores user details and hashed passwords.

rooms:

Contains room information and status.

bookings:

Holds booking records, dates, user and room references, and status.

payments:

(If implemented) Stores payment transactions.

admin:

(Optional) For admin user management.

How These Elements Interact

User registers (registerUser) → data stored in users.

User logs in (loginUser) → session started.

User searches for rooms (getAvailableRooms) → displays available rooms.

User books a room (bookRoom) → checks for overlap, inserts booking.

User pays (processPayment) → booking confirmed.

User/admin views bookings (getUserBookings, getAllBookings).

Admin manages rooms/users (addRoom, editRoom, getAllUsers).

User logs out (logoutUser) → session destroyed.

**Conclusion**

The Hotel Booking System project successfully demonstrates the development of a comprehensive, user-friendly, and secure platform that streamlines the entire hotel reservation process. By automating key functions such as user registration, room searching, booking management, and payment processing, the system significantly reduces manual workload and minimizes errors, enhancing operational efficiency for hotel staff and convenience for guests.

The modular design and robust backend logic ensure data integrity, prevent double bookings, and safeguard sensitive information through secure authentication and encryption techniques. Additionally, the system’s scalability and maintainability make it adaptable to future enhancements, such as integrating payment gateways, adding promotional features, or expanding to multi-hotel management.

Overall, this project not only meets the functional requirements of a modern hotel booking system but also emphasizes critical non-functional aspects like security, usability, and performance. It provides a solid foundation for real-world deployment, offering a seamless experience that benefits both hotel management and customers alike.

**Bibliography and References**

Randive, G., Pawar, A., Poharkar, S., Patil, V., & Aundhakar, S. (2023). Hotel Booking System with New Features. International Research Journal of Modernization in Engineering, Technology and Science, 5(4), 7231-7238.

[PDF Reference: IRJMETS1683366037]

This paper discusses the evolution of hotel booking systems, the need for seamless user experience, and the integration of advanced features in modern reservation platforms.

Shewale, P. S., Khode, R. R., Pardeshi, A. M., Yadav, V. P., & Dashore, P. (2024). Multi-Hotel Booking System. International Journal of Novel Research and Development, 9(2), 306-307.

[PDF Reference: IJNRD2402333]

Explores the challenges of fragmented booking systems and proposes a centralized, user-friendly multi-hotel booking web application.

Chavan, S. (2024). Online Room Booking System for HotelVilla18. Gurukul International Multidisciplinary Research Journal, 12(Special Issue), 363-368.

[PDF Reference: GIMRJ2406I8V12P044]

Investigates the impact of web-based hotel booking systems on user experience and business efficiency, with a focus on interface design, real-time availability, and secure payment processing.

Patil, V., Poharkar, S., Pawar, A., Randive, G., & Aundhakar, S. (2023). Hotel Booking System with New Features. International Journal of Innovative Science and Research Technology, 2324.

[PDF Reference: IJISRT23MAR034]

Describes the implementation of an online hotel booking system, highlighting the advantages of global accessibility, convenience, and integrated management features.

Bemile, R., Achampong, A., & Danquah, E. (2014). Development of a Comparison Based Hotel and Resort Booking System in Bangladesh.

[Reference: SCIRP]

Presents the development and evaluation of a hotel and resort booking system, focusing on comparison features and usability.

Grafiati (n.d.). Bibliographies: 'Hotels – Reservation systems'.

[Reference: Grafiati]

Provides curated bibliographies and references on hotel reservation systems, including articles, books, and theses.

Academia.edu. Online Hotel Reservation System Research Papers.

[Reference: Academia.edu]

A collection of research papers on the design, development, and evaluation of online hotel reservation systems.

Promus Hotel Corp. (1995). The Evolution of Hotel Room Reservation Systems, from Pencil and Paper to Real-Time.

[PDF Reference: IJARSCT]

Reviews the historical development of hotel reservation systems and the technological advancements that shaped modern platforms.

How These References Were Used

**System Design and Features:**

References , , , and provided insight into the essential modules (registration, booking, payment, admin), usability considerations, and the importance of real-time room availability.

**Industry Context and Best Practices:**

References , , and offered foundational knowledge on hotel reservation systems, their evolution, and comparative studies of different approaches.

**Technical Implementation:**

References , , and informed the technical architecture, security practices, and integration of new features in the system.