Development of Online Hotel Reservation Management System

Md. Mehedi Hasan

ID# 21103032

A Practicum in the Partial Fulfillment of the Requirements
for the Award of Bachelor of Computer Science and Engineering (BCSE)

Department of Computer Science and Engineering
College of Engineering and Technology
IUBAT–International University of Business Agriculture and Technology

Summer 2024

Online Hotel Reservation Management System

Md. Mehedi Hasan

A Practicum in the Partial Fulfillment of the Requirements for the Award of Bachelor of Computer Science and Engineering (BCSE)

The practicum has been examined and approved,

Prof. Dr. Utpal Kanti Das

Chairman

Shahinur Alam

Co-supervisor, Coordinator and Assistant Professor

Afsana Akter Lija

Supervisor and Designation of the Supervisor

Department of Computer Science and Engineering

College of Engineering and Technology

IUBAT–International University of Business Agriculture and Technology

Summer 2024

Letter of Transmittal

17 October 2024

The Chair

Practicum Defense Committee

Department of Computer Science and Engineering

IUBAT–International University of Business Agriculture and Technology

4 Embankment Drive Road, Sector 10, Uttara Model Town

Dhaka 1230, Bangladesh.

Subject: Letter of Transmittal.

I respectfully wish to present to you my practicum report, and it is with great enthusiasm and delight that I share this opportunity with you. The report is titled: **"Development of Online Hotel Reservation Management System."** for the fulfillment of my Practicum course.

Working as an intern at ImpleVista was an immensely enjoyable and fulfilling experience throughout the course of two months. I have had ample opportunities to apply my academic knowledge in a real-world business setting throughout this practicum. I tried my hardest to complete this practicum report. I am now anticipating your thoughtful evaluation of this practicum report.

I would be eternally appreciative if you could review this report and assess my work.

Regards,

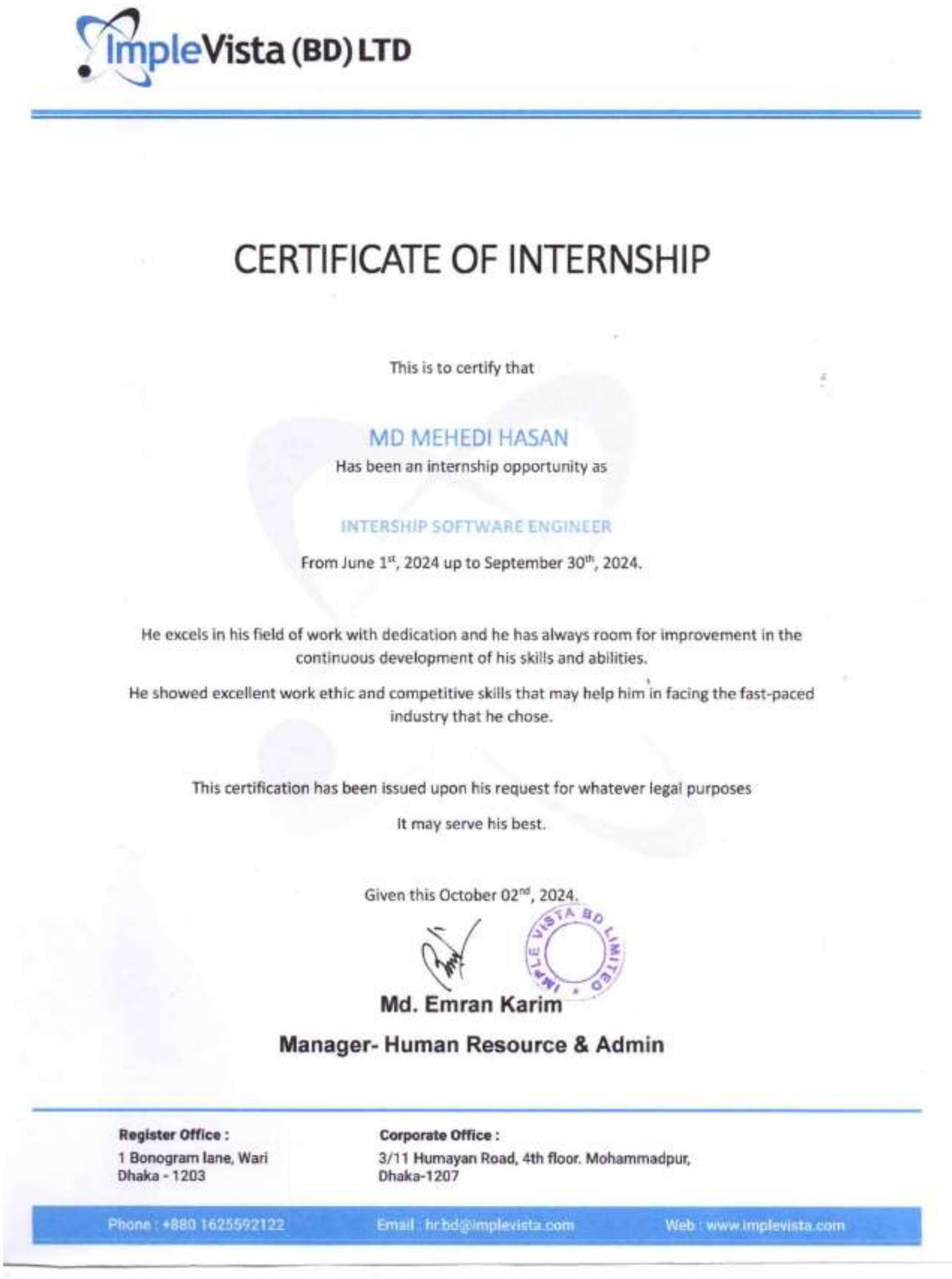
Yours sincerely,

____

Md. Mehedi Hasan

ID# 21103032

Organization Certificate



ImpleVista (BD) LTD

CERTIFICATE OF INTERNSHIP

This is to certify that

MD MEHEDI HASAN

Has been an internship opportunity as

INTERSHIP SOFTWARE ENGINEER

From June 1st, 2024 up to September 30th, 2024.

He excels in his field of work with dedication and he has always room for improvement in the continuous development of his skills and abilities.

He showed excellent work ethic and competitive skills that may help him in facing the fast-paced industry that he chose.

This certification has been issued upon his request for whatever legal purposes

It may serve his best.

Given this October 02nd, 2024.

IMPLE VISTA BD LIMITED

Md. Emran Karim

Manager- Human Resource & Admin

Register Office :
1 Bonogram lane, Wari
Dhaka - 1203

Corporate Office :
3/11 Humayan Road, 4th floor. Mohammadpur,
Dhaka-1207

Phone : +880 1625592122 Email : hr.bd@implevista.com Web : www.implevista.com

Student’s Declaration

The development of an online employee administration system is the subject of this report. I, Md. Mehedi Hasan, ID:21103032, a student enrolled in the College of Engineering and Technology (CEAT) of the International University of Business, Agriculture, and Technology's Bachelor of Computer Science and Engineering (BCSE) program, hereby declare the following. It possesses the necessary qualifications to fulfill both the BCSE (Bachelor of Computer Science and Engineering) half requirement and the CSC 490 Practicum internship prerequisites. The report and project regarding the development of an online hotel management system for ImpleVista were both authored by me. The modules and methods for this project are all developed subsequent to comprehensive examination and internet investigation. It is not yet suitable for additional accolades, exhibitions, or applications. I am certain that this online hotel reservation management system provides a reliable and efficient solution.

Md. Mehedi Hasan

ID# 21103032

Supervisor's Certification

This is to certify that Md. Mehedi Hasan, ID: 21103032 of IUBAT—International University of Business, Agriculture, and Technology—partially fulfilled the Bachelor of Computer Science and Engineering (BCSE) program requirement for the practicum report "Development of Online Hotel Management System" at the Department of Computer Science and Engineering (CSE), IUBAT. I have supervised the methodology employed in the compilation of the report. Based on the available information and in alignment with his declaration, he compiled this report, the contents of which have not been previously submitted for credit toward a degree, diploma, or certification. At this moment, he is ready to submit the report. Regardless of the outcome of his future endeavors, I wish the best.

Afsana Akter Lija

Lecturer

Department of Computer Science and Engineering

IUBAT–International University of Business Agriculture and Technology

Abstract

This practicum work presents an Online Hotel Reservation Management System designed to streamline hotel operations and enhance guest experiences. The system serves two primary user groups: hotel guests and administrative staff. For guests, the platform offers a user-friendly interface with features including registration, room browsing with advanced filtering, detailed room information, and a seamless booking process. Additional functionalities include service overviews and a contact form. The user journey guides guests from room selection to payment, with a profile section for managing bookings and personal information. On the administrative side, the system provides a comprehensive dashboard with real-time insights into key metrics such as occupancy rates and user statistics. Administrators can manage rooms, handle booking requests, and access user information efficiently. The system aims to improve operational efficiency, enhance guest satisfaction, and provide valuable business insights. Key benefits include streamlined booking processes, personalized guest experiences, and improved administrative workflows. This Online Hotel Management System represents a significant advancement in hotel operations, offering potential for increased direct bookings and improved online presence. Future enhancements could include integration with third-party booking platforms and mobile app functionality. Overall, this practicum demonstrates the successful development of a comprehensive solution addressing the needs of both hotel guests and administrators, potentially leading to improved efficiency and guest satisfaction in the hospitality industry.

Acknowledgments

In the name of the Almighty, whose benevolence and magnanimity transcend all boundaries. I would like to extend my sincere appreciation to the late Professor Dr. Md. Alimullah Miyan, who served as the inaugural Vice-Chancellor and founder of the International University of Business, Agriculture, and Technology (IUBAT), for bestowing upon me the opportunity to enroll in this renowned and visually captivating non-governmental institution. IUBAT, the International University of Business, Agriculture, and Technology, is privileged to have as honorary vice chancellor Professor Dr. Abdur Rab. Permit me to extend my gratitude and sincere appreciation to him.

I am writing to extend my sincere appreciation to the Chairman of the Department of Computer Science and Engineering at IUBAT, the International University of Business, Agriculture, and Technology, Prof. Dr. Utpal Kanti Das, for his unwavering support in funding my studies and for inspiring me to envision a promising future in the realm of cutting-edge technology.

I am exceedingly grateful to Rashedul Islam and Shahinur Alam, assistant professor of computer science and engineering and the esteemed coordinator of IUBAT, for his outstanding guidance and support throughout this semester.

I am indebted to the Managing Director of ImpleVista for his invaluable collaboration.

I am delighted to extend my sincere appreciation and admiration to our distinguished faculty members, with a special mention to Afsana Akter Lija, a lecturer in the Department of Computer Science and Engineering at IUBAT, for his indispensable scholarly counsel and steadfast dedication to the compilation of this report.

I wish to express my gratitude to my parents and instructors for serving as tremendous inspiration throughout my academic journey. Their unwavering support has been instrumental in facilitating my progress to this stage.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction:

In the rapidly evolving digital landscape, the hospitality industry has increasingly turned to online platforms to streamline operations and enhance guest experiences. This report focuses on the development and implementation of a comprehensive Online Hotel Management System, designed to meet the dynamic needs of both hotel administrators and guests.

Our system is a web-based platform that facilitates efficient hotel operations management while providing a user-friendly interface for potential guests to explore, book, and manage their stays. The project aims to create a seamless connection between the hotel's operational requirements and guest services, leveraging technology to improve overall efficiency and customer satisfaction.

The system is divided into two main components: the user-facing website and the admin panel. The user side allows potential guests to browse hotel information, search for and book rooms, and manage their reservations. It includes features such as detailed room listings, service descriptions, and a secure booking process. The admin panel, on the other hand, provides hotel staff with tools to manage rooms, bookings, and guest information, as well as access to important business metrics and reports.

Throughout this report, we will explore the system's architecture, key features, and the rationale behind various design decisions. We will also discuss the challenges encountered during development and the solutions implemented to address them. By the end of this document, we aim to demonstrate how this online hotel management system can significantly enhance operational efficiency and guest satisfaction in the modern hospitality landscape.

This project represents a significant step forward in integrating digital solutions into hotel management, offering benefits such as increased booking efficiency, improved customer service, and streamlined administrative processes. As we delve into the details of this system, we will highlight how it addresses current industry needs and positions hotels for success in an increasingly digital-first market.

1.2 Objectives of OHRMS

Objectives

1. To develop a user-friendly and efficient online platform for hotel management and guest services.
2. To streamline the room booking process, making it easier for potential guests to find, select, and reserve accommodations.
3. To provide a comprehensive admin panel that enables hotel staff to effectively manage rooms, bookings, and guest information.
4. To implement a secure user authentication system for both guests and administrators.
5. To create a detailed room management system that allows for easy updates of room information, availability, and pricing.
6. To design an intuitive interface for guests to view hotel services and amenities.
7. To develop a contact system that allows guests to easily communicate with hotel staff.
8. To implement a user profile system that stores booking history and personal preferences.
9. To create a dashboard for administrators that provides key business metrics and occupancy data.
10. To ensure the system is scalable and can accommodate future growth and additional features.
11. To implement a secure payment gateway for processing room bookings.
12. To develop a responsive design that functions well on both desktop and mobile devices.
13. To create a flexible system that can be easily customized for different types of accommodations.
14. To improve operational efficiency by automating routine tasks such as booking confirmations and availability updates.
15. To enhance the guest experience through personalized services and easy access to information.

These objectives aim to create a comprehensive, efficient, and user-friendly online hotel management system that benefits both the hotel administration and its guests.

1.3 Source of the Report

This practicum report has been developed as an essential segment of the internship program of IUBAT. This internship has been completed under the IP Link Network which is an ISP company. The company has observed their duty with the internship accessory. The respectable faculty members Prof. Dr. Utpal Kanti Das, Chairman; Shahinur Alam, Coordinator and Assistant professor; and Afsana Akter Lija, Supervisor, Department of Computer Science and Engineering, IUBAT are the persons who are responsible for the gracious deliberation and supervise the practicum program academically from the institution. The your-organization-name has given a chance to perform the internship program with them and contributed the essential information for making this practicum report.

1.4 Methodologies

1. Development Stack: MERN (MongoDB, Express.js, React.js, Node.js)
    - MongoDB: Used as the database to store and manage hotel data, user information, and bookings.
    - Express.js: Employed as the backend framework to create a robust and scalable server-side application.
    - React.js: Utilized for building the dynamic and responsive user interface.
    - Node.js: Used as the runtime environment for the server-side application.
2. Agile Development Methodology
    - Implemented Scrum framework for iterative and incremental development.
    - Conducted regular sprint planning, daily stand-ups, and sprint retrospectives.
    - Utilized user stories and product backlog for feature prioritization.
3. Version Control
    - Used Git for version control and GitHub for repository hosting.
    - Implemented feature branching and pull request reviews for code quality assurance.
4. RESTful API Design
    - Designed and implemented RESTful APIs for communication between frontend and backend.
    - Used JSON for data exchange between client and server.

5. Responsive Web Design
    - Employed CSS frameworks (e.g., Bootstrap or Material-UI) for responsive design.
    - Implemented mobile-first approach to ensure compatibility across devices.
6. Security Measures
    - Implemented JWT (JSON Web Tokens) for user authentication and authorization.
    - Used bcrypt for password hashing.
    - Implemented input validation and sanitization to prevent common web vulnerabilities.

7. Testing
    - Conducted unit testing using Jest for backend and React Testing Library for frontend.
    - Performed integration testing to ensure proper communication between components.
    - Implemented end-to-end testing using tools like Cypress.
8. Code Quality
    - Used ESLint for code linting to maintain consistent code style.
    - Implemented Prettier for code formatting.
9. Continuous Integration and Deployment (CI/CD)
    - Utilized tools like Jenkins or GitHub Actions for automated testing and deployment.
    - Implemented staging and production environments for thorough testing before release.
10. Database Design
    - Designed efficient MongoDB schemas to optimize data storage and retrieval.
    - Implemented indexing for improved query performance.
11. State Management

- Used Redux for global state management in the React application.

12. API Documentation
    - Created comprehensive API documentation using tools like Swagger.
13. Performance Optimization
    - Implemented lazy loading and code splitting in React for improved load times.
    - Used caching strategies to reduce database queries and improve response times.
14. Accessibility
    - Followed WCAG guidelines to ensure the application is accessible to users with disabilities.
15. Scalability Considerations
    - Designed the architecture to be scalable, considering potential future growth.
    - Implemented load balancing strategies for handling increased traffic.

These methodologies were chosen and implemented to ensure the development of a robust, scalable, and user-friendly online hotel management system using the MERN stack.

1.4.1 Primary Sources.

Primary Sources

1. User Requirements Gathering
    - Conducted interviews with hotel managers and staff to understand operational needs
    - Surveyed potential hotel guests to identify key features for the user-facing website
    - Analyzed feedback from existing hotel booking platforms to identify areas for improvement
2. Industry Standards and Best Practices
    - Reviewed hospitality industry guidelines for online booking systems
    - Studied recommendations from organizations like the World Tourism Organization (UNWTO) on digital tourism services
3. Technical Documentation
    - Official documentation for MongoDB, Express.js, React.js, and Node.js

- API documentation for payment gateways and other third-party services integrated into the system

4. Regulatory Compliance
    - Local and international data protection regulations (e.g., GDPR, CCPA)
    - Payment Card Industry Data Security Standard (PCI DSS) for handling financial transactions
5. Competitor Analysis
    - Examined features and user interfaces of leading hotel booking websites
    - Analyzed admin panels of existing hotel management systems
6. Academic and Industry Research
    - Reviewed academic papers on hotel management systems and e-commerce in hospitality
    - Studied industry reports on trends in online hotel bookings and management
7. User Testing Feedback
    - Conducted usability testing sessions with potential users and hotel staff
    - Gathered and analyzed feedback from beta testers of the system
8. Performance Metrics
    - Collected and analyzed data on website performance, user engagement, and conversion rates during the development and testing phases
9. Design Guidelines
    - Material Design guidelines for creating a consistent and intuitive user interface
    - Web Content Accessibility Guidelines (WCAG) for ensuring accessibility
10. Security Protocols
    - OWASP (Open Web Application Security Project) guidelines for web application security
    - Best practices for secure user authentication and data protection in web applications

These primary sources provided invaluable insights and data that guided the development of the online hotel management system, ensuring that it meets industry standards, user needs, and technical requirements.

1.4.2 Secondary Sources.

The steps secondary sources are given below:

1. Industry Reports and Market Analysis
    - Hospitality industry reports from organizations like Deloitte and PwC
    - Market analysis documents on the hotel booking software industry
2. Technology Blogs and Articles
    - Web development blogs discussing best practices in MERN stack development
    - Articles on emerging trends in hotel management software
3. Case Studies
    - Published case studies on successful implementations of hotel management systems
    - Analysis of digital transformation in the hospitality industry
4. Books
    - "Hotel Management and Operations" by Michael J. O'Fallon and Denney G. Rutherford
    - "Web Development with Node and Express" by Ethan Brown
5. Online Courses and Tutorials
    - Udemy and Coursera courses on MERN stack development
    - YouTube tutorials on building e-commerce platforms and booking systems
6. Conference Proceedings
    - Papers from hospitality technology conferences
    - Presentations from web development and software engineering symposiums
7. Industry Magazines and Publications
    - "Hotel Management" magazine
    - "Hospitality Technology" publication
8. Online Forums and Discussion Boards
    - Stack Overflow discussions on MERN stack development challenges
    - Reddit threads on hotel management software features and user experiences
9. Whitepapers
    - Technical whitepapers on database optimization for booking systems

- Industry whitepapers on improving customer experience in online hotel bookings

10. Podcasts
    - Web development podcasts discussing modern JavaScript frameworks
    - Hospitality industry podcasts exploring digital trends in hotel management
11. Webinars
    - Recorded webinars on building scalable web applications
    - Online seminars discussing the future of hotel technology
12. Software Reviews
    - User reviews of existing hotel management systems on platforms like Capterra and G2
    - Professional reviews of web development tools and frameworks
13. Academic Journals
    - Journal of Hospitality and Tourism Technology
    - International Journal of Contemporary Hospitality Management
14. Government and NGO Reports
    - Tourism department reports on digital adoption in the hospitality sector
    - World Tourism Organization (UNWTO) publications on digital transformation in tourism

These secondary sources provided additional context, theoretical frameworks, and industry insights that complemented the primary sources and informed various aspects of the project's development and implementation.

Chapter 2

Organizational Overview

ImpleVista is an innovative IT solutions provider that specializes in leveraging SAP S/4Hana to streamline business operations, offering a native cloud solution for efficient management of SAP and Non-SAP landscapes. Their platform, simplifies business accounting and management, while their Business Analytics services provide actionable insights for strategic growth. With a focus on cloud computing, mobile application design, IoT solutions, and e-commerce, ImpleVista empowers businesses to flourish by transforming their core competencies into market effectiveness. Their team of dedicated engineers utilizes the latest technologies to build fast, user-friendly apps and websites, ensuring businesses can harness the power of data for demand forecasting and predictive analytics. ImpleVista stands out for its commitment to enhancing online identity and keeping clients informed with industry-specific expert articles.

2.1 Mission

Implevista's mission is to provide top-notch digital marketing solutions to businesses. They prioritize understanding each client's unique objectives and tailor their strategies accordingly. By leveraging innovative techniques, they aim to boost online presence and drive measurable results.

2.2 Vision

Implevista envisions a world where businesses harness the power of digital marketing to reach global audiences effectively. Their expertise in the local market allows them to create campaigns that resonate with Bangladeshi audiences. Through clear communication and transparent processes, they guide businesses toward success.

2.3 Organization Services

1. Website Design & Development: A well-designed and developed website can be a powerful tool for capturing attention, engaging visitors, and driving successful conversions. Implevista specializes in creating user-friendly websites that align withourbusiness goals.
2. Content Marketing: Effective content marketing goes beyond promoting products or services—it's about tellingourbrand's story, providing value, and building a loyal audience. Implevista crafts compelling content that resonates withourtarget audience.
3. Social Media Marketing: Social media marketing is an ever-evolving landscape that demands innovation, expertise, and a deep understanding of audience dynamics. Implevista helpsournavigate this space to maximizeourbrand's impact.
4. PPC Advertising (Pay-Per-Click): Implevista leverages PPC advertising to drive targeted traffic and achieve measurable results in today's competitive digital landscape. Their strategies ensure efficient use ofouradvertising budget.
5. Search Engine Optimization (SEO): Implevista specializes in SEO services that enhanceouronline visibility and drive qualified traffic toourwebsite.

Figure 1: Organization Logo

2.4 Address of Office

3/4 Humayon Road, Mohammadpur, Dhaka, Bangladesh.

Chapter 3

Requirement Engineering

Within the engineering design process, requirement engineering constitutes the procedure of recognizing, capturing, and sustaining requirements. Requirements engineering can be defined in various ways, but they all share a common concept: it involves the process of understanding user expectations from a computer system and interpreting these requirements into design considerations. Software engineering, which primarily focuses on the procedures for developing systems aligned with consumer preferences, is closely intertwined with requirement engineering. Requirements divided into four types.

They are:

1. User Requirements
2. System Requirements
3. Functional Requirements
4. Non-Functional Requirements

3.1 User requirements

1. Register and log in to access the website
2. Browse hotel overview, rooms, services, and contact information
3. Search and filter available rooms
4. View detailed room information
5. Book rooms online
6. Submit personal information for booking
7. Complete payment process
8. Access personal profile and booking history
9. Contact hotel through a message form

3.2 System requirements

1. Web server to host the website
2. Database to store user, room, booking, and other relevant data
3. Payment gateway integration for secure transactions

4. Email system for notifications and confirmations
5. Content management system for easy updates of hotel information
6. Responsive design for compatibility across devices (desktop, tablet, mobile)

3.3 Functional requirements

User Side:

1. User registration and authentication system
2. Room search and filtering functionality
3. Room booking system with availability check
4. User information collection form
5. Secure payment processing
6. User profile management
7. Booking history display
8. Contact form for user inquiries

Admin Side:

1. Admin authentication system
2. Dashboard with business analytics and graphs
3. Room management (add, update, delete)
4. Booking request management (view, accept, reject)
5. User information management
6. Admin profile management

3.4 Non-functional requirements

1. Performance: The website should load within 3 seconds and handle at least 1000 concurrent users
2. Security: Implement HTTPS, secure data storage, and protection against common web vulnerabilities (e.g., SQL injection, XSS)
3. Reliability: System uptime of at least 99.9%

4. Scalability: Able to handle increasing numbers of users and bookings without performance degradation
5. Usability: Intuitive interface design for easy navigation and booking process
6. Accessibility: Comply with WCAG 2.1 guidelines for accessibility
7. Data Integrity: Ensure accurate and consistent data across all system components
8. Compatibility: Function correctly on major web browsers (Chrome, Firefox, Safari, Edge)
9. Responsiveness: Adapt to different screen sizes and devices
10. Compliance: Adhere to relevant data protection regulations (e.g., GDPR if applicable)
11. Maintainability: Well-documented code and system architecture for easy updates and maintenance
12. Localization: Support for multiple languages if catering to international customers

3.5 Software Requirements

1. Operating System:
   - Windows 10 or higher / macOS / Linux (Ubuntu 18.04 or higher)
2. Backend:
   - Node.js: JavaScript runtime environment (v14.17.0 or higher)
   - Express.js: Web framework for Node.js
   - MongoDB: NoSQL database (v4.2 or higher)
   - Mongoose: ODM for MongoDB
3. Frontend:
   - React.js: Frontend JavaScript library (v17 or higher)
   - TypeScript: Superset of JavaScript for type safety (optional but recommended)
   - Redux: State management library (optional but recommended for complex state handling)
   - Tailwind CSS / Bootstrap: CSS frameworks for styling
4. Middleware:
   - JWT (JSON Web Token): For authentication
   - Multer: For file uploads

- Dotenv: For environment variable management

5. Development Tools:
    - Visual Studio Code: Code editor
    - Postman: API testing tool
    - Git: Version control system
    - npm / Yarn: Package managers
    - Webpack / Parcel: Module bundlers for efficient asset management
6. Browser:
    - Google Chrome / Mozilla Firefox (latest version)
7. Server Hosting:
    - AWS / DigitalOcean: Cloud service for deploying the application
    - Nginx / Apache: Web server for hosting the application
8. Version Control:
    - GitHub / GitLab: For code repository management
9. Testing:
    - Jest / Mocha: Testing frameworks for unit and integration tests
    - Cypress: End-to-end testing

3.6 Hardware Requirements

1. Development Machine:
    - Processor: Intel Core i5 or higher / AMD Ryzen 5 or higher
    - RAM: 8GB minimum (16GB recommended for smoother multitasking)
    - Storage: SSD with at least 256GB free space (512GB recommended)
    - Graphics: Integrated graphics is sufficient, but a dedicated GPU (like Nvidia GTX series) is beneficial for design tasks
    - Display: 1080p resolution or higher
2. Server:
    - Processor: Intel Xeon or equivalent
    - RAM: 16GB minimum (32GB recommended for handling multiple requests)
    - Storage: SSD with RAID configuration for reliability (500GB or more)

- Network: High-speed internet connection with low latency (minimum 100 Mbps)

3. Backup and Recovery:
    - External hard drive or cloud storage for regular backups (at least 1TB capacity)
4. Peripherals:
    - Keyboard and Mouse: Standard or ergonomic design
    - Monitor: Dual monitors recommended for efficient multitasking
    - UPS: Uninterruptible Power Supply for protection against power outages

These requirements will ensure that the Online Hotel Management System operates smoothly and efficiently throughout the development and deployment phases.

3.7 Use Case Diagram

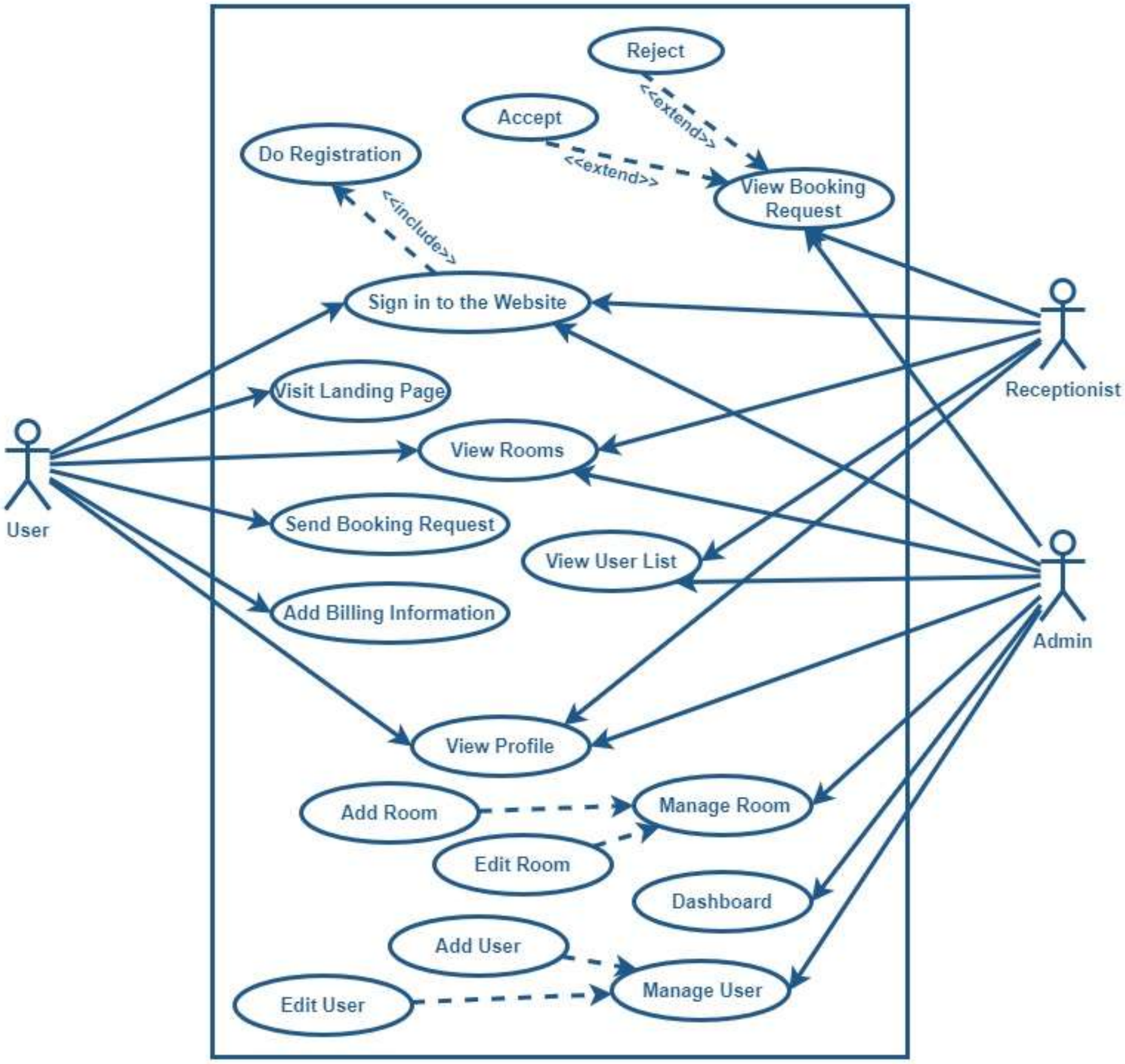


Figure 2: Use Case Diagram

This is a use case diagram for an Online Hotel Management System, depicting the interactions between different user roles (User, Receptionist, Admin) and the various functionalities they can access.

Actors:

User: A guest or customer who interacts with the website for booking rooms and managing their profile.

Receptionist: A hotel staff member responsible for managing booking requests and user information.

Admin: The system administrator responsible for managing the overall hotel operations, including users, rooms, and bookings.

Use Cases:

Do Registration

Actor: User

Description: The user can register on the website to create an account. This step is necessary to gain access to further functionalities like booking rooms.

Relationship: Includes "Sign in to the Website" (as part of the registration process, the user might be redirected to the sign-in page).

Sign in to the Website

Actors: User, Receptionist, Admin

Description: All users (User, Receptionist, Admin) need to sign in to access personalized and restricted areas of the website.

Visit Landing Page

Actor: User

Description: After signing in, the user can view the landing page, which provides an overview of the hotel's services and offerings.

View Rooms

Actors: User, Receptionist, Admin

Description: All users can view the available rooms along with their details. The user can filter and search for rooms, while the receptionist and admin can view additional details and manage room data.

Send Booking Request

Actor: User

Description: The user can select a room and send a booking request for it. This is the initial step in the booking process.

Relationship: Extends "View Rooms" (as viewing rooms is a prerequisite for sending a booking request).

Add Billing Information

Actor: User

Description: The user provides billing or payment details after submitting a booking request. This is necessary to confirm the booking.

Relationship: Extends "Send Booking Request" (as billing information is provided after sending a booking request).

View Profile

Actors: User, Admin

Description: The user can view and manage their personal information and booking history. The admin can also view the user's profile for administrative purposes.

View Booking Request

Actors: Receptionist, Admin

Description: Receptionists and admins can view all booking requests submitted by users. This includes details like the requested room, check-in/check-out dates, and user information.

Relationship: Extends "View Rooms" (as booking requests are associated with room details).

Accept / Reject Booking Request

Actors: Receptionist, Admin

Description: Receptionists and admins have the authority to accept or reject booking requests based on availability and other criteria.

Relationship: Extends "View Booking Request" (as accepting or rejecting a request is an action taken after viewing it).

View User List

Actors: Receptionist, Admin

Description: Receptionists and admins can view the list of all registered users along with their details.

Add & Edit Room

Actor: Admin

Description: The admin can add new rooms to the system or edit the details of existing rooms.

Dashboard

Actor: Admin

Description: The admin can view the dashboard, which provides an overview of hotel operations, such as occupancy rates, the number of users, and other key metrics.

Add & Edit User

Actor: Admin

Description: The admin can add new users (e.g., other admins or staff) and edit existing user information.

Relationships:

Include: Represents a mandatory use case that must be included as part of another use case.

For example, "Do Registration" includes "Sign in to the Website," meaning that after registering, the user will typically need to sign in.

Extend: Represents an optional use case that extends the functionality of another use case.

For example, "Accept" and "Reject" booking requests extend "View Booking Request," indicating that these actions are taken after reviewing the request details.

Summary:

This diagram visually outlines how different roles interact with the system's features, showcasing the flow of actions such as registration, booking, room management, and user management. The relationships between use cases help clarify the sequence and dependencies of these actions, making it easier to understand the system's functionality from both user and administrative perspectives.

Chapter 4

System Planning

4.1 System Project Planning

Project planning is a fundamental component of successful project management, playing a pivotal role in guiding a project from conception to completion. It serves as the blueprint that outlines the project’s objectives, scope, deliverables, and the detailed steps necessary to achieve them. By establishing clear timelines, resource allocations, and budget estimates, project planning ensures that all team members and stakeholders are aligned and aware of their roles and responsibilities. This alignment is essential for minimizing misunderstandings and conflicts that can arise from ambiguous expectations or unclear communication.

Moreover, thorough project planning allows for the identification and assessment of potential risks that could impede progress. By anticipating these challenges in advance, project managers can devise contingency plans and risk mitigation strategies, thereby reducing the likelihood of costly delays or failures. This proactive approach not only safeguards the project’s timeline but also helps in maintaining the quality of the deliverables by ensuring that all necessary resources, including time, budget, and personnel, are utilized effectively.

In addition to risk management, project planning also plays a critical role in resource optimization. It enables project managers to allocate resources efficiently, avoiding overallocation or underutilization that can lead to project bottlenecks or unnecessary expenses. By maintaining a clear and organized plan, teams can work more cohesively, ensuring that everyone is working towards the same goals with a shared understanding of the project’s priorities. Ultimately, project planning enhances the overall efficiency and effectiveness of the project, ensuring that it is completed on time, within budget, and to the satisfaction of all stakeholders.

4.2 Function Oriented Metrics

The function point estimation offers more significant information. Five attributes are evaluated in order to determine function points. The attributes are as follows:

The number of external inputs (EI) – This refers to the number of different items that users submit into the program to offer relevant data. These inputs must be distinguished from plain questions or information requests.

The number of external outputs (EO) – This counts the number of times the software returns information to the user.

The number of external inquiries (EQ) – An online input that prompts an instantaneous software response in the form of an online output is referred to as an inquiry. Every inquiry's findings are added together separately.

The number of internal logical files (ILF) – In this case, the database table that the program uses to update the input is counted together with a file.

Numbers of external interface files (EIF) – The total number of machine-readable interfaces that are used to transfer data from one system to another is determined.
To determine the formula's FP count,

4.3 Function Point Estimation

Below is a comprehensive breakdown of each component, including the specific functions, their counts, complexities, and the corresponding function points. This detailed estimation helps in calculating the Unadjusted Function Points (UFP) of the system.

Table 1 : Functional Point Estimation

Component	Description	Count	Complexity	FP Weight	Total FP
External Inputs (EI)	Inputs that provide data into the system, usually through forms.				
Registration Form	Allows new users to register by providing personal details.	1	Average	4	4
Login Form	Enables existing users and admins to access their accounts.	1	Low	3	3
Booking Information Form	Users provide detailed booking data including dates, guests, etc.	1	High	6	6
Billing Information Form	Users input payment information to finalize their booking.	1	Average	4	4
Contact Form	Allows users to send inquiries directly to the hotel administration.	1	Average	4	4
Admin Room Management	Admin inputs for adding or updating room details, pricing, and status.	1	High	6	6
Receptionist Check-In Form	Allows receptionists to manage check-ins and check-outs.	1	High	6	6
Subtotal EI		7			33
External Outputs (EO)	Outputs generated by the system, providing essential information.				
Booking Confirmation	Displays booking confirmation after the booking process.	1	Average	5	5
Payment Receipt	Generated receipt of payment for the user’s records.	1	Average	5	5

Admin Reports	Reports on bookings, room availability, and occupancy rates.	1	High	7	7
User Profile Output	Displays user information and booking history on the profile page.	1	Average	5	5
Booking Status Update	Provides users and admins with booking status notifications.	1	Low	4	4
Room Availability Alerts	Notifications for users on room availability changes.	1	Low	4	4
Subtotal EO		6			30
External Inquiries (EQ)	Input-output combinations that provide quick results to user actions.				
Room Search	Search functionality for users to find rooms based on criteria.	1	Average	4	4
Filter Options	Allows users to filter rooms based on price, amenities, and more.	1	Low	3	3
Booking Inquiry	Users can check booking status or details directly.	1	Average	4	4
Admin Inquiry for Reports	Admins inquire for specific reports based on dates or criteria.	1	High	6	6
Payment Inquiry	Allows users to check transaction status or payment history.	1	Low	3	3
Subtotal EQ		5			20

Internal Logical Files (ILF)	Files that are maintained internally by the system.				
User Database	Stores all registered user data, including bookings and history.	1	High	7	7
Room Database	Detailed room information, including descriptions, prices, and status.	1	High	7	7
Booking Database	Contains all booking records, including dates, statuses, and payments.	1	High	7	7
Admin and Staff Records	Admin and staff management data, including roles and access levels.	1	Average	5	5
Service Management File	Stores data on additional services offered by the hotel.	1	Low	4	4
Subtotal ILF		5			30
External Interface Files (EIF)	Files that the system interacts with but does not maintain.				
Payment Gateway Interface	Interfaces with external services for secure payment processing.	1	Average	5	5
Third-Party Room Listings	Integrates with external platforms for additional room visibility.	1	Low	4	4
External Analytics	Pulls data from third-party analytics for business insights.	1	Low	4	4
Subtotal EIF		3			13

Calculation of Unadjusted Function Points (UFP)

Summing all function points from the individual components gives us the total UFP:

Total UFP = 33+30+20+30+13=126

Average VAF = 1

AFP = UFP * VAF = 126 * 1 = 126

Effort = 126/20 = 6.3 (person months)

Effort = 6.3*160 = 1008 (person hours)

Insights from the Function Point Estimation

Total UFP: 126, reflecting a comprehensive and functionally rich system with multiple complex components.

Key Contributors:

External Inputs (33 points): The system requires a significant number of inputs, particularly for user interactions, such as bookings, payments, and admin management tasks.

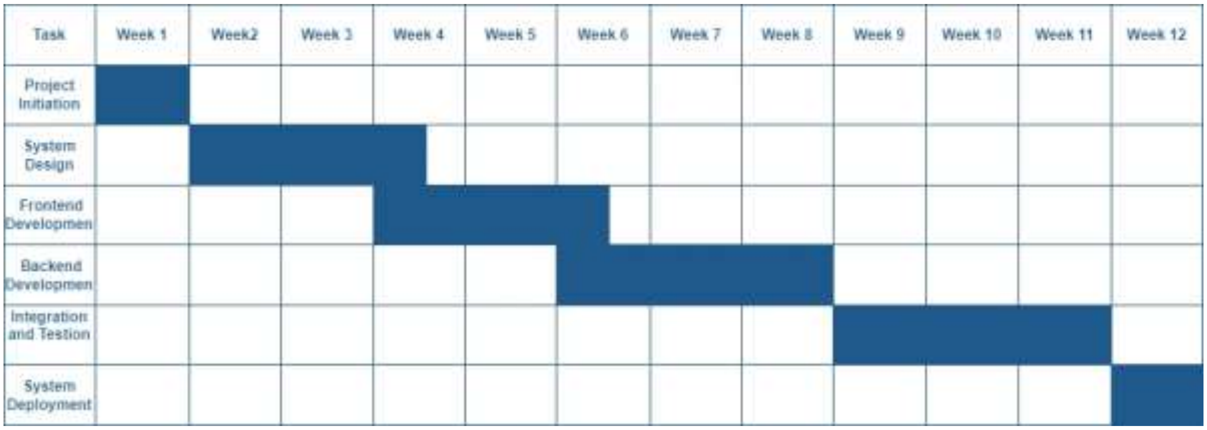
External Outputs (30 points): The system generates various outputs, including confirmations, receipts, and detailed reports essential for user and admin roles.

Internal Logical Files (30 points): Managing internal data, especially related to user accounts, bookings, and room management, is a critical aspect of the system.

Overall Complexity: The system's complexity is primarily driven by its diverse functionality, the need to handle data securely, and the requirement to provide real-time information to multiple user roles.

This detailed function point analysis is crucial for estimating the project's scope, development effort, and resource allocation, ensuring successful completion within the planned timeline.

4.4 Project scheduling Chart



Task	Week 1	Week2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12
Project Initiation	■											
System Design		■	■	▌								
Frontend Developmen				■	■	▌						
Backend Developmen						■	■	■				
Integration and Testion									■	■	■	
System Deployment												■

Figure 3 : Project Scheduling Chart

Week 1: Project Initiation

Tasks:

Project kick-off and requirement gathering.

Define project scope, objectives, and key deliverables.

Set up the development environment.

Week 2-3: System Design

Tasks:

Design system architecture and database schema.

Create UI/UX prototypes and wireframes.

Finalize design with stakeholder feedback.

Week 3-6: Frontend Development

Tasks:

Implement user registration and login pages.

Develop Home, Rooms, Services, and Contact pages.

Create Room Details, Booking Information, Billing, and User Profile pages.

Ensure integration with backend APIs.

Week 5-8: Backend Development

Tasks:

Set up backend services (Express.js, MongoDB).

Develop API endpoints for user, room, booking, and billing management.

Implement admin and receptionist functionalities.

Integrate payment gateway.

Week 7-10: Integration and Testing

Tasks:

Integrate frontend with backend services.

Conduct end-to-end testing.

Perform security testing and optimization.

Week 10-11: System Deployment

Tasks:

Deploy the application to a live environment.

Finalize domain and SSL configurations.

Conduct final testing and troubleshooting.

Week 11-12: Post-Deployment and Handover

Tasks:

Provide training and documentation.

Conduct User Acceptance Testing (UAT).

Finalize project handover to stakeholders.

Offer post-deployment support.

This schedule aligns with the provided chart and ensures that all critical tasks are completed within the 12-week timeline.

4.5 Cost Estimation

4.5.1. Personnel Costs

Since I have haldled all aspects of the project, this will be the main cost component.

Table 2 : Personal Cost

Task	Estimated Hours/Week	Weeks	Total Hours	Hourly Rate (BDT)	Total Cost (BDT)
Project Management	5	12	60	2,000	120,000
Frontend Development	20	8	160	2,000	320,000
Backend Development	20	8	160	2,000	320,000
UI/UX Design	10	4	40	2,000	80,000
Testing and Integration	15	4	60	2,000	120,000
System Deployment	10	1	10	2,000	20,000
Total Personnel Cost			490 Hours		980,000

4.5.2. Software Costs

These are costs associated with software tools needed for development, design, and testing.

Table 3 : Software Cost

Item	Cost (BDT)
Design Tools (e.g., Adobe XD, Figma)	15,000
IDEs and Development Tools (e.g., WebStorm, Visual Studio)	20,000
Testing Tools (e.g., Postman)	10,000
Version Control Hosting (e.g., GitHub)	5,000
Total Software Cost	50,000

4.5.3. Hardware Costs

Assumingouralready have basic hardware, this might be minimal.

Table 4 : Hardware Cost

Item	Cost (BDT)
Development Laptop	100,000
Backup Storage Device	20,000
Total Hardware Cost	120,000

4.5.4. Miscellaneous Costs

These include overheads such as internet, utilities, and potential office space.

Table 5 : Miscellaneous Cost

Item	Cost (BDT)
Internet & Utilities	15,000
Miscellaneous (e.g., Travel, Stationery)	10,000
Total Miscellaneous Cost	25,000

4.5.5. Contingency

Given that you're managing everything, a small contingency fund can cover unexpected issues.

Table 6 : Contingency

Contingency (10%)	Amount (BDT)
10% of Total Cost	117,500

4.5.6. Total Estimated Cost

Combining all the above costs gives the total estimated cost for the project.

Table 7 : Total Estimated Cost

Category	Cost (BDT)
Personnel Cost	980,000
Software Cost	50,000
Hardware Cost	120,000
Miscellaneous Cost	25,000
Contingency	117,500
Total Project Cost	1,292,500 BDT

Summary

Total Estimated Cost: 1,292,500 BDT

This estimation assumes a moderate level of complexity and that all tasks are handled by one individual. Adjustments may be needed based on actual time spent on each task.

Chapter 5

Risk Management

Risk Management involves identifying, assessing, and mitigating potential risks that could affect the project's success. In the context of the Online Hotel Management System, risk management is essential to ensure that the project is completed on time, within budget, and meets the desired quality standards.

5.1 Stages of Risk Management

The stages of risk management in a project like the Online Hotel Management System are crucial for systematically identifying, assessing, mitigating, and controlling risks throughout the project's lifecycle. Understanding these stages helps in ensuring that risks are managed effectively to avoid negative impacts on the project's success. Here's a detailed breakdown of each stage:

1. **Risk Identification**

- **Objective:** The first stage of risk management involves identifying all possible risks that could affect the project. These risks could be related to the project scope, timeline, resources, technology, or external factors.
- **Activities:**
  - **Brainstorming:** Engage the project team, stakeholders, and experts in brainstorming sessions to list potential risks.
  - **Checklists:** Use standardized risk checklists that include common risks encountered in similar projects.
  - **Interviews and Surveys:** Conduct interviews and surveys with team members and stakeholders to gather insights on potential risks.
  - **SWOT Analysis:** Analyze the project's Strengths, Weaknesses, Opportunities, and Threats to uncover risks.
  - **Document Review:** Review project documents, such as the project charter, scope, and requirements, to identify risks.
- **Output:** A comprehensive list of potential risks that could affect the project.

2. **Risk Assessment**

Objective: Once risks are identified, the next stage involves assessing each risk to understand its potential impact on the project and the likelihood of its occurrence.

Activities:

Risk Analysis: Analyze each identified risk to determine its probability and impact. This can be done qualitatively (using descriptive scales) or quantitatively (using numerical data).

Risk Categorization: Group risks into categories (e.g., technical, operational, financial) to better understand and manage them.

Risk Prioritization: Rank risks based on their severity and the urgency with which they need to be addressed. High-priority risks require immediate attention, while low-priority risks can be monitored over time.

Risk Mapping: Create a risk matrix that plots risks on a grid based on their likelihood and impact. This visual tool helps prioritize risks more effectively.

Output: A prioritized list of risks, each with an assigned probability and impact score, ready for further analysis and mitigation.

3. **Risk Mitigation Planning**

Objective: In this stage, strategies are developed to manage each identified risk. The goal is to either reduce the likelihood of the risk occurring or minimize its impact if it does occur.

Activities:

Avoidance: Modify the project plan to eliminate the risk entirely. For example, if a certain technology poses a high risk, choose a more reliable alternative.

Reduction: Implement actions to reduce the likelihood or impact of the risk. This could involve additional training, more thorough testing, or using better tools.

Transfer: Transfer the risk to a third party, such as by outsourcing a risky component of the project or purchasing insurance.

Acceptance: Accept the risk if it cannot be avoided, reduced, or transferred. Prepare contingency plans to manage the risk if it materializes.

Contingency Planning: Develop backup plans and allocate resources for managing risks that cannot be mitigated upfront.

Output: A risk management plan that outlines the strategies for handling each risk, along with contingency plans and assigned risk owners.

4. **Risk Monitoring and Control**

Objective: This stage involves the ongoing monitoring of risks throughout the project to ensure that they are being managed effectively and to identify any new risks that may arise.

Activities:

Regular Risk Reviews: Schedule regular meetings to review the status of identified risks and discuss any new risks that have emerged.

Risk Audits: Conduct periodic audits to assess the effectiveness of risk management strategies and make necessary adjustments.

Performance Metrics: Track key performance indicators (KPIs) related to risk management, such as the number of risks identified, mitigated, or realized.

Risk Reassessment: Continuously reassess risks based on project progress and changes in the project environment.

Issue Resolution: Address any issues that arise as a result of risks materializing, following the predefined mitigation or contingency plans.

Output: Updated risk register, regular risk reports, and a controlled environment where risks are managed proactively.

5. **Risk Communication**

Objective: Throughout the risk management process, effective communication is essential to ensure that all stakeholders are informed about risks, their potential impact, and the strategies in place to manage them.

Activities:

Risk Reporting: Provide regular risk reports to stakeholders, including updates on the status of high-priority risks.

Stakeholder Engagement: Engage stakeholders in discussions about risks, ensuring that their concerns are addressed, and they are aware of the risk management plan.

Transparency: Maintain open lines of communication about risks, their management, and any changes to the risk management plan.

Escalation: Establish clear escalation procedures for critical risks that require attention from higher management or stakeholders.

Output: Clear and consistent communication about risks, ensuring that everyone involved in the project is informed and aligned with the risk management strategy.

6. **Risk Response and Resolution**

Objective: When a risk materializes, the project team must respond effectively to mitigate its impact and resolve any issues that arise.

Activities:

Activate Contingency Plans: Implement the pre-defined contingency plans to manage the impact of the risk on the project.

Adjust Project Plan: Make necessary adjustments to the project plan, timeline, and resources to accommodate the impact of the risk.

Resource Allocation: Allocate additional resources or reassign existing ones to address the issues caused by the risk.

Communication: Inform stakeholders about the risk event, the response strategy, and any changes to the project plan.

Output: Successful resolution of the risk event with minimal disruption to the project, along with updates to the risk management plan.

7. **Post-Project Risk Review**

- **Objective:** After the project is completed, a thorough review of the risk management process is conducted to assess its effectiveness and identify lessons learned.
- **Activities:**
    - **Lessons Learned:** Document lessons learned from managing risks throughout the project. Identify what worked well and what could be improved.
    - **Risk Management Report:** Prepare a comprehensive report detailing the risks encountered, how they were managed, and the outcomes.
    - **Process Improvement:** Use insights gained from the post-project review to improve risk management processes for future projects.
    - **Knowledge Sharing:** Share the lessons learned and best practices with other teams and future projects to build organizational knowledge.

- **Output:** A finalized risk management report, documentation of lessons learned, and improved risk management practices for future projects.

By following these detailed stages of risk management, a project like the Online Hotel Management System can better anticipate, manage, and mitigate risks, leading to a higher likelihood of project success.

5.2 Categories of Risk

When managing a software project like an Online Hotel Management System, understanding and categorizing risks is crucial to ensure the project's success. Below are detailed explanations of the various categories of risk:

1. **Technical Risks**

Complexity of Technology: The use of new or unproven technology can lead to unexpected problems. For example, integrating a new payment gateway or using a specific framework like MERN (MongoDB, Express.js, React.js, Node.js) for the first time may introduce challenges.

System Integration: Risks related to integrating different components or systems. Inourproject, integrating the front-end with the back-end or connecting third-party services like payment processors can pose risks.

Performance Issues: There is a risk that the system might not meet performance requirements, such as speed, responsiveness, or scalability. This is particularly important for the user experience in a web-based application.

2. **Project Management Risks**

Scope Creep: The risk of the project growing beyond its initial scope due to the addition of new features or requirements, leading to increased timelines and costs.

Scheduling Risks: Delays in the project timeline can occur due to various factors, including resource availability, unexpected technical challenges, or underestimating the complexity of tasks.

Resource Allocation: Mismanagement of resources (like time, personnel, or budget) can lead to project delays or the inability to deliver the project within the planned scope.

3. **Operational Risks**

Human Resources: The risk of losing key team members, such as the developer (since it's a single-person project), or not having access to the necessary skill sets.

Process Failures: Risks arising from inadequate processes or the failure to follow established processes. For instance, skipping testing phases or not documenting the code properly can lead to system failures.

Infrastructure Issues: Any problems with the infrastructure that supports the project, such as development tools, hosting services, or version control systems. Downtime in the server hosting the development environment can be a significant risk.

4. **Financial Risks**

Budget Overruns: The risk of exceeding the planned budget due to underestimations, unexpected expenses, or scope creep.

Return on Investment (ROI): The risk that the project may not generate the expected returns, especially if it is a commercial project where profitability is a key concern.

Funding Shortages: If the project is self-funded or dependent on external investors, there's a risk that funding might run out before project completion.

5. **Legal and Compliance Risks**

Regulatory Compliance: Failure to comply with legal requirements, such as data protection laws (e.g., GDPR), can result in legal penalties. For a hotel management system, handling customer data securely is critical.

Intellectual Property Risks: Risks related to the use of third-party software, libraries, or tools without proper licensing, which could lead to legal disputes.

Contractual Obligations: Not meeting contractual obligations with clients, vendors, or partners, which could result in penalties or loss of business.

6. **External Risks**

Market Risks: Changes in the market that could affect the project's viability, such as a downturn in the hospitality industry or the emergence of a strong competitor with a better offering.

Political Risks: Changes in regulations or political instability in the region where the project is being developed or deployed can impact project timelines and success.

Environmental Risks: Natural disasters or other environmental factors that could disrupt the project's progress, such as severe weather affecting internet connectivity or power supply.

7. **Security Risks**

- **Cybersecurity Threats:** The risk of external attacks, such as hacking, which could compromise sensitive customer data or disrupt service. For a hotel management system, ensuring secure transactions and data storage is paramount.
- **Data Breaches:** The risk of unauthorized access to sensitive information, leading to data loss or legal repercussions.
- **Vulnerability Exploitation:** The risk that security flaws in the software or infrastructure could be exploited by malicious entities.

Each of these risk categories must be carefully assessed and managed throughout the project lifecycle. Implementing a comprehensive risk management plan that includes risk identification, assessment, mitigation strategies, and continuous monitoring is crucial for successfully delivering the Online Hotel Management System.

5.3 Risk Management Strategies

Effective risk management strategies are crucial for ensuring the successful completion of a software project like the Online Hotel Management System. Below are detailed strategies for managing different types of risks throughout the project lifecycle:

1. Risk Identification

- **Continuous Monitoring:** Establish a process for ongoing risk identification. Regularly assess the project environment, technology, market trends, and team dynamics to spot potential risks early.
- **Brainstorming Sessions:** Conduct regular brainstorming sessions with stakeholders, including the project team, clients, and vendors, to identify potential risks. This collaborative approach can uncover risks that might not be apparent initially.
- **SWOT Analysis:** Utilize SWOT (Strengths, Weaknesses, Opportunities, Threats) analysis to identify internal and external risks. This strategic tool can help in recognizing areas that require more attention.

2. Risk Assessment

- **Risk Prioritization:** Once risks are identified, prioritize them based on their potential impact and likelihood. High-impact, high-probability risks should be addressed first, while low-impact, low-probability risks can be monitored.
- **Risk Matrix:** Create a risk matrix that categorizes risks based on their severity and likelihood. This visual tool helps in understanding which risks require immediate attention and which can be addressed later.
- **Quantitative Analysis:** For significant risks, conduct a quantitative analysis to estimate the potential financial impact. This analysis helps in understanding the cost implications of risks and preparing appropriate mitigation strategies.

3. Risk Mitigation Strategies

- **Avoidance:** In some cases, avoiding the risk entirely is the best strategy. For example, if using an unproven technology poses a significant risk, opting for a more stable and proven solution can mitigate that risk.
- **Reduction:** Implement strategies to reduce the likelihood or impact of risks. This could involve more thorough testing, additional training for the team, or adopting more reliable development tools and technologies.
- **Transfer:** Transfer the risk to a third party, such as purchasing insurance or outsourcing certain aspects of the project. For example,ourmight use a third-party service for payment processing to mitigate the risks associated with handling financial transactions in-house.
- **Acceptance:** For risks that cannot be avoided, reduced, or transferred, develop a plan to accept them and prepare contingency plans. This might include allocating a portion of the budget as a contingency fund or preparing a disaster recovery plan.

4. Risk Monitoring and Control

- **Regular Reviews:** Conduct regular risk reviews throughout the project lifecycle to monitor the status of identified risks and identify any new risks. This ensures that risks are continually managed and mitigated.
- **Risk Owner Assignment:** Assign a specific team member as the owner of each risk. The risk owner is responsible for monitoring the risk, implementing mitigation strategies, and reporting on the risk's status.

- **Performance Indicators:** Establish key performance indicators (KPIs) related to risk management, such as the number of risks identified, mitigated, or realized. Tracking these KPIs helps in understanding the effectiveness of the risk management process.

5. Communication and Reporting

- **Clear Communication Plan:** Develop a clear communication plan that outlines how risks will be communicated to stakeholders. This includes regular status updates, risk reports, and escalation procedures for critical risks.
- **Transparency:** Maintain transparency with all stakeholders regarding the risks the project faces. This ensures that everyone is aware of potential challenges and can contribute to the risk management process.
- **Risk Register:** Maintain a risk register that document all identified risks, their assessment, mitigation strategies, and current status. This register should be accessible to all project stakeholders and regularly updated.

6. Contingency Planning

- **Develop Contingency Plans:** For high-priority risks, develop contingency plans that outline the steps to be taken if the risk materializes. This ensures that the project can continue with minimal disruption.
- **Budget Contingency:** Allocate a portion of the project budget as a contingency fund to cover unexpected costs related to risks. This financial buffer provides flexibility in dealing with unforeseen challenges.
- **Backup Systems:** Implement backup systems and procedures to ensure business continuity in case of a major risk event. For example, regular backups of the project's codebase and database can prevent data loss in the event of a system failure.

7. **Risk Response Strategy**

- **Proactive Response:** Encourage a proactive approach to risk management, where the team actively seeks out potential risks and addresses them before they become issues. This mindset helps in reducing the overall impact of risks.
- **Agile Response:** In an Agile project environment, regularly review and adjust risk management strategies based on the project's progress. This iterative approach ensures that risks are managed in real-time.

- **Escalation Procedures:** Establish clear escalation procedures for critical risks that cannot be managed at the project level. This ensures that these risks are brought to the attention of senior management or other relevant stakeholders quickly.

8. **Post-Project Risk Review**

- **Lessons Learned:** After the project is completed, conduct a post-project review to assess how risks were managed. Identify what worked well and what could be improved for future projects.
- **Documentation:** Document the entire risk management process, including all identified risks, mitigation strategies, and outcomes. This documentation serves as a valuable resource for future projects.
- **Continuous Improvement:** Use the insights gained from the post-project review to improve risk management processes for future projects. Continuously refining the risk management approach ensures that the team is better prepared for future challenges.

By following these detailed risk management strategies, our can effectively minimize the impact of risks on our Online Hotel Management System project, ensuring a smoother development process and a successful project outcome.

Chapter 6

Analysis & Modelling

6.1 Analysis Modeling

An approach used in software engineering called analysis modeling is used to depict and explain component interactions, system behavior, and requirements. It comprises understanding, assessing, and elucidating system requirements and design decisions via the creation of visual models or representations of the system's behavior, structure, and functioning. Among the analytical modeling methods accessible are entity-relationship diagrams, swim lane diagrams, use case diagrams, activity diagrams, and data flow diagrams. Through the software development process, analysis modeling helps to validate system behavior, get insights into system requirements, and improve stakeholder communication.

6.2 Software Analysis Pattern

- The software analysis pattern's objectives are:
- Boost cooperation and communication
- Reduce the complexity of complicated systems
- Encourage upgradability and maintainability
- Encourage optimal procedures
- Offer Remedies
- Boost the Caliber of Software

6.3 Activity Diagram

An activity diagram is a type of behavioral diagram in UML (Unified Modeling Language) that visually represents the flow of activities or actions within a system, showcasing how different processes interact and sequence over time. It depicts the step-by-step workflow of system operations, including decisions, parallel processes, and data flow, making it ideal for modeling complex business processes, use cases, or any sequence of tasks. Activity diagrams use symbols like actions, decisions, forks, joins, and start/end points to illustrate the overall logic and behavior of a system, helping developers, analysts, and stakeholders better understand and communicate the dynamics of the system’s functionality.

6.3.1 Activity Diagram for User Authentication

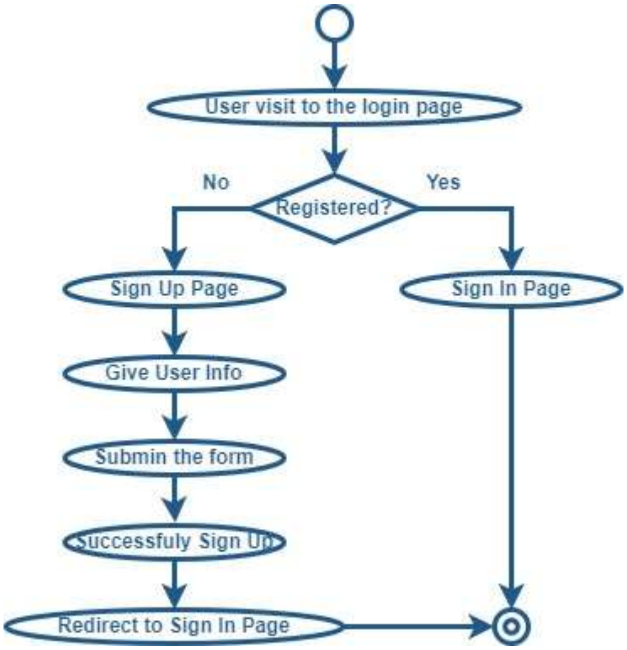


Figure 4 : Activity Diagram (User Authentication)

6.3.2 Activity Diagram for Booking Request

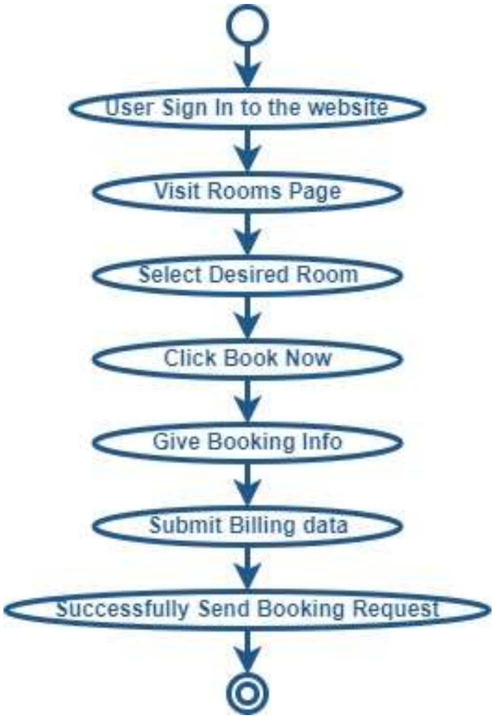


Figure 5 : Activity Diagram (Booking Request)

6.3.3 Activity Diagram for Cancel Booking Request

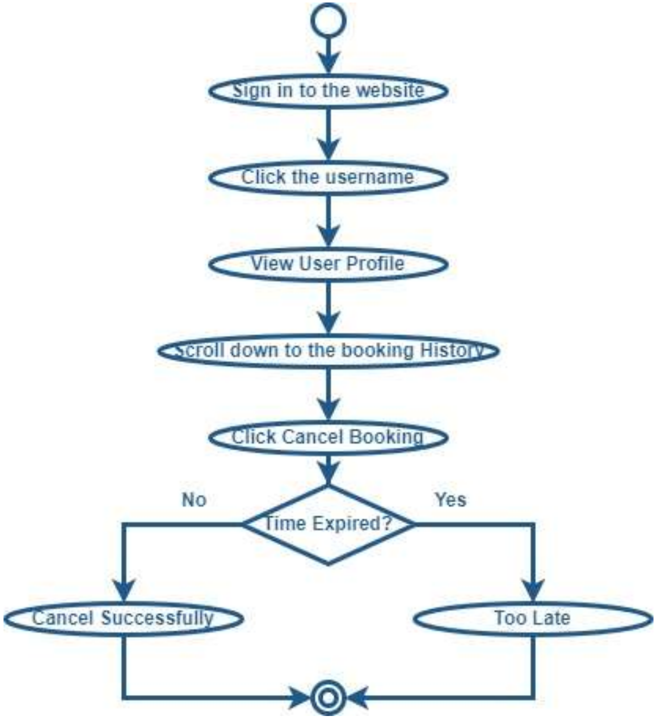


Figure 6 : Activity Diagram (Cancel Booking Request)

6.3.4 Activity Diagram for Manage Booking request

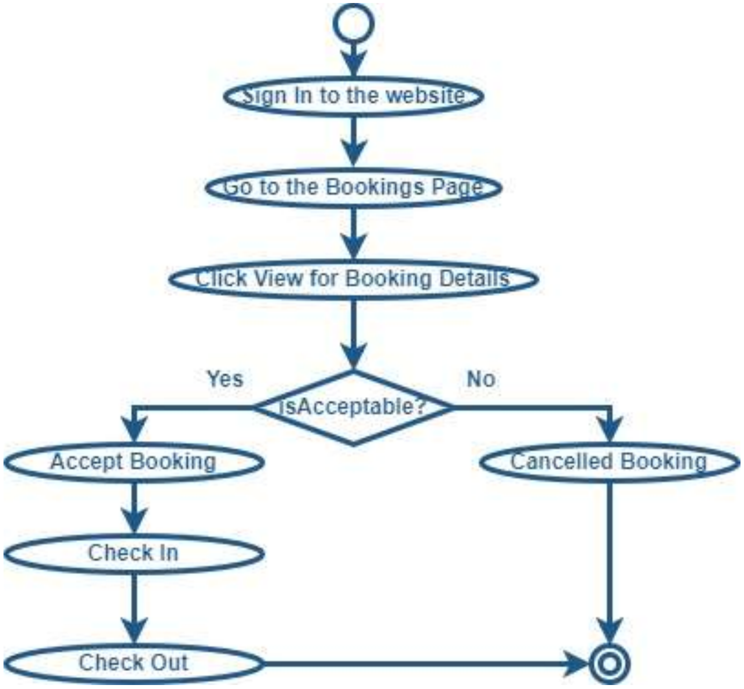


Figure 7 : Activity Diagram (Manage Booking Request)

6.3.5 Activity Diagram for Manage Rooms

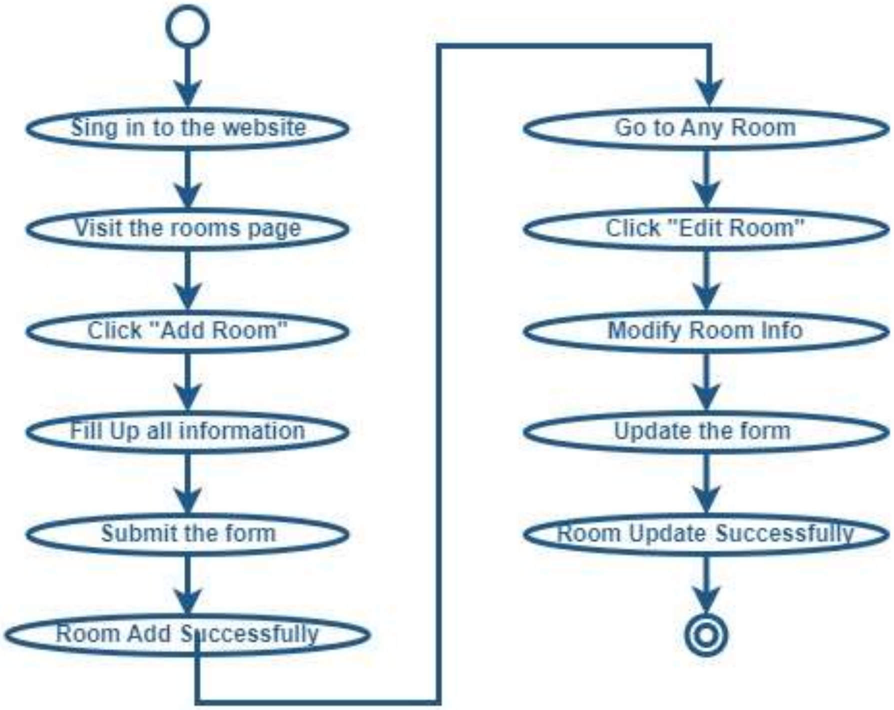


Figure 8 : Activity Diagram (Manage Rooms)

6.3.6 Activity Diagram for Manage Users

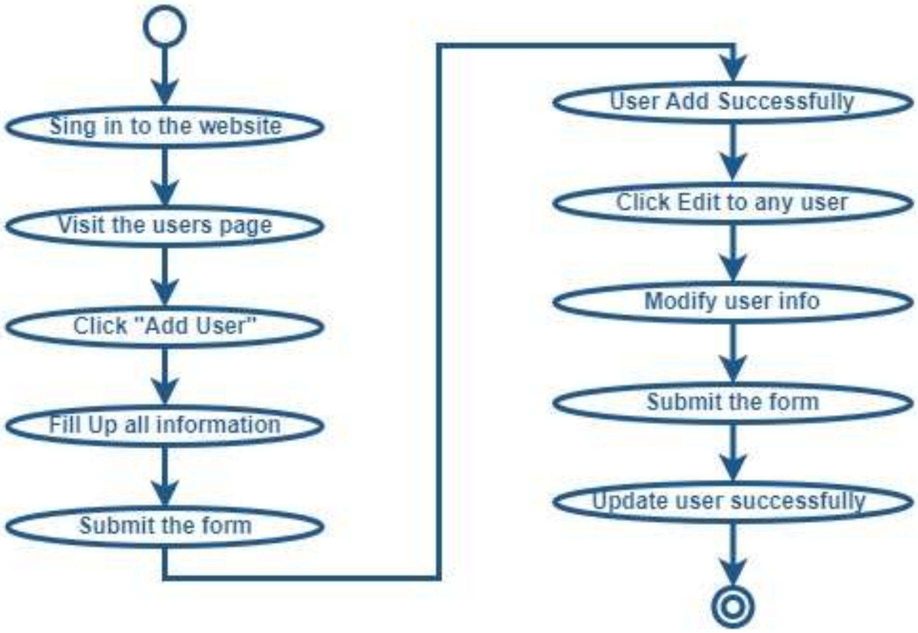


Figure 9 : Activity Diagram (Manage Rooms)

6.4 Entity Relationship Diagram

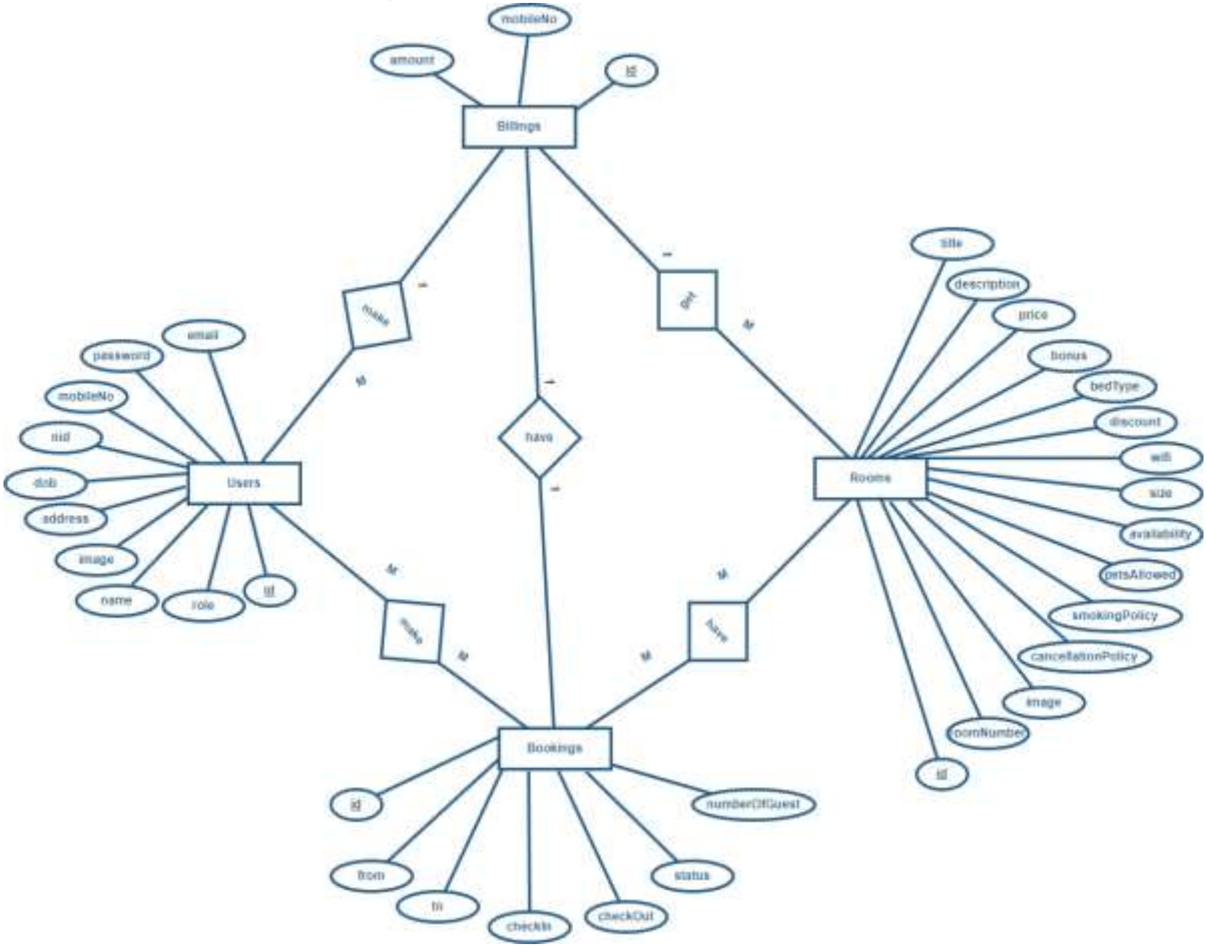


Figure 10 : Entity Relationship Diagram

6.5 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of how data moves through a system, illustrating the flow of information between processes, data stores, external entities, and data inputs/outputs. It provides a high-level view of the system’s structure, focusing on how data is processed and shared rather than on the sequence of tasks. DFDs use standardized symbols: circles or ovals for processes, arrows for data flow, open-ended rectangles for data stores, and squares for external entities (like users or other systems). By mapping out these components, a DFD helps identify data dependencies, streamline processes, and uncover inefficiencies, making it an essential tool for system analysis and design.

6.5.1 Context Level Diagram (DFD 0)

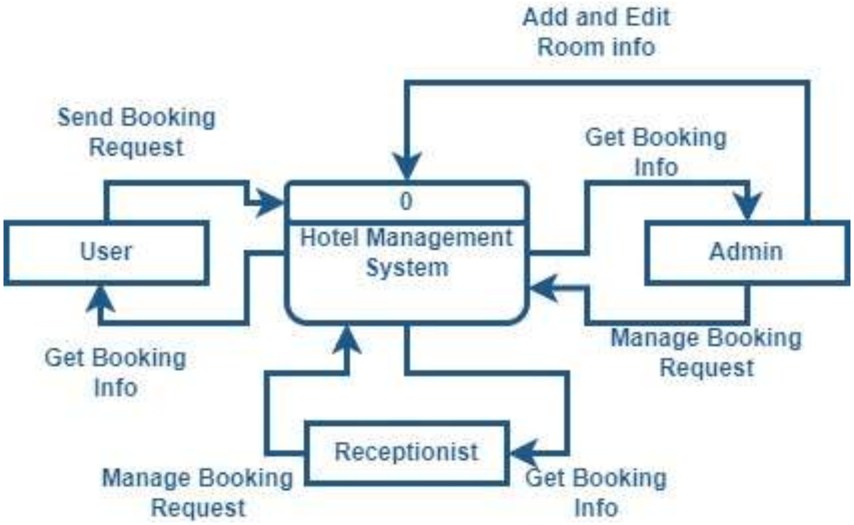


Figure 11 : Context Level Diagram

6.5.2 Level 1 DFD

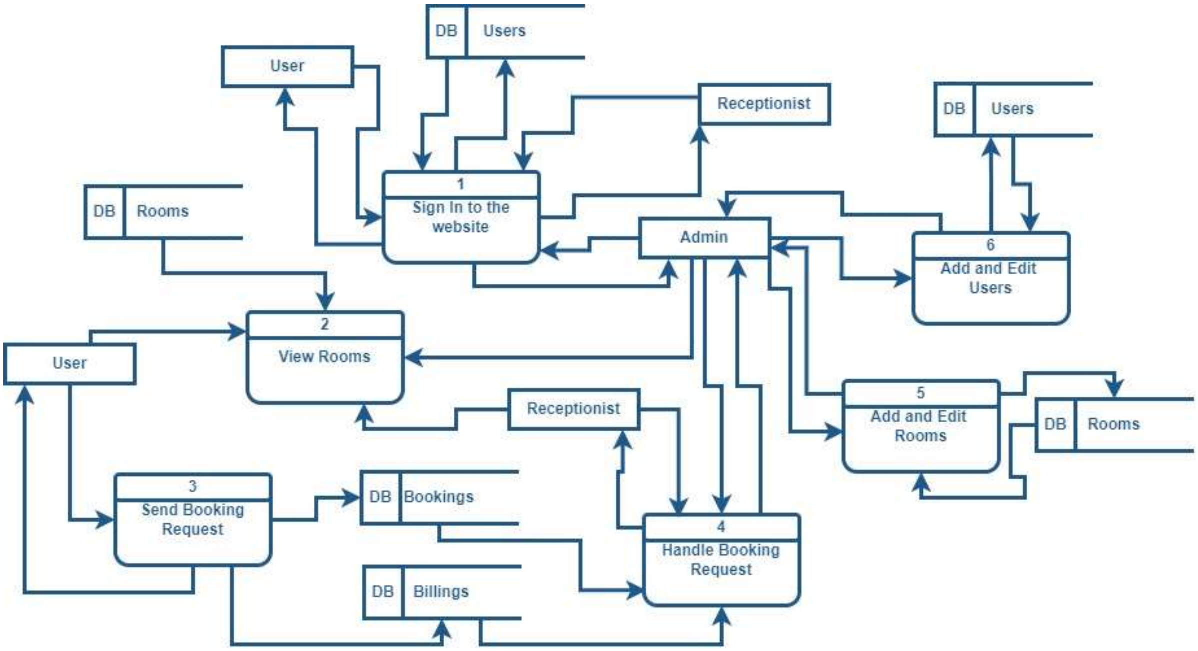


Figure 12 : Level 1 DFD Diagram

6.5.3 Level 2 Process (Authentication)

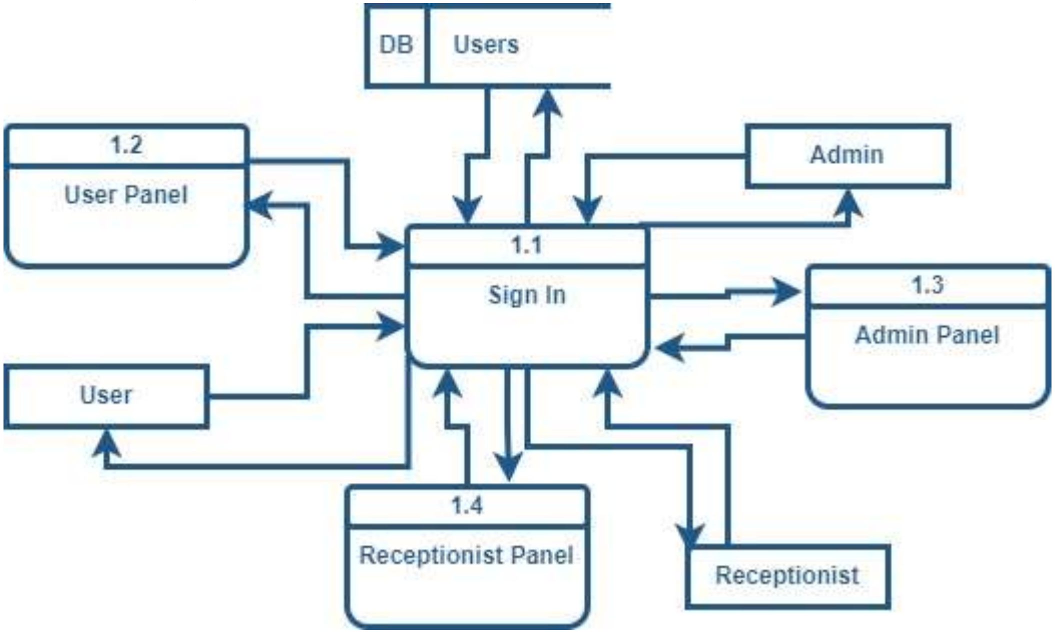


Figure 13 : Level 2 DFD (User Authentication)

6.5.3 Level 2 Process (Viewing Room Details)

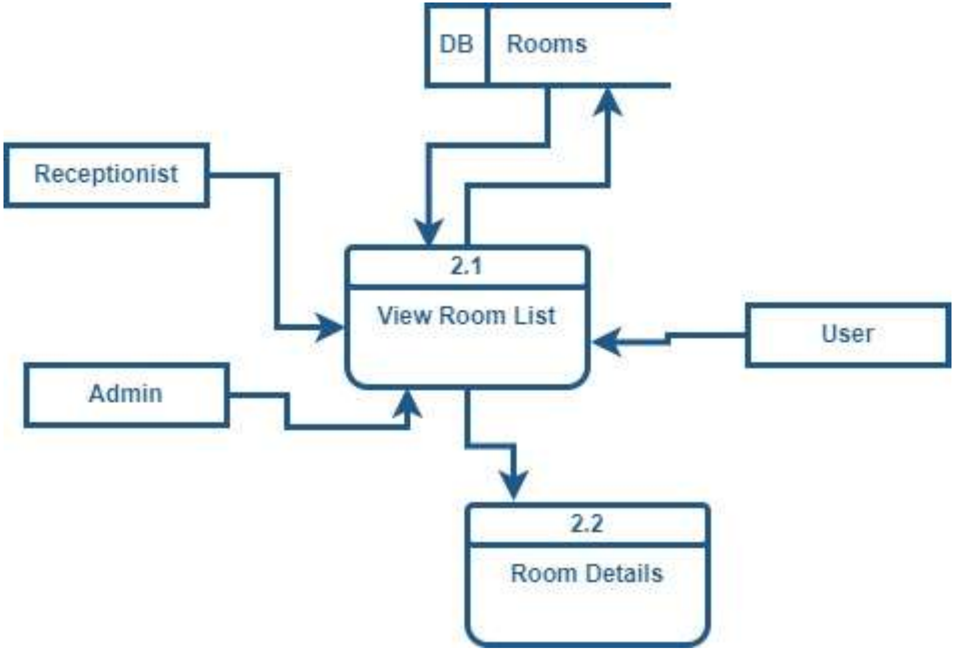


Figure 14 : Level 2 DFD (View Room Details)

6.5.3 Level 2 Process (Booking Request)

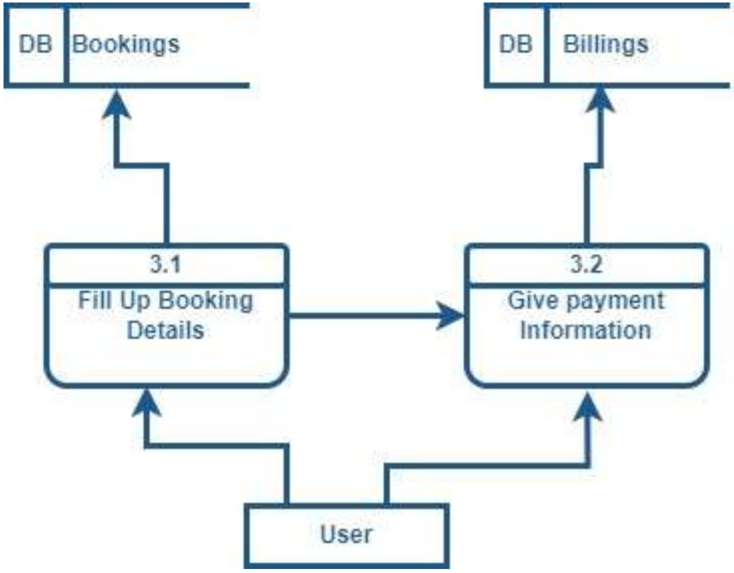


Figure 15 : Level 2 DFD (Booking Request)

6.5.3 Level 2 Process (Booking Details)

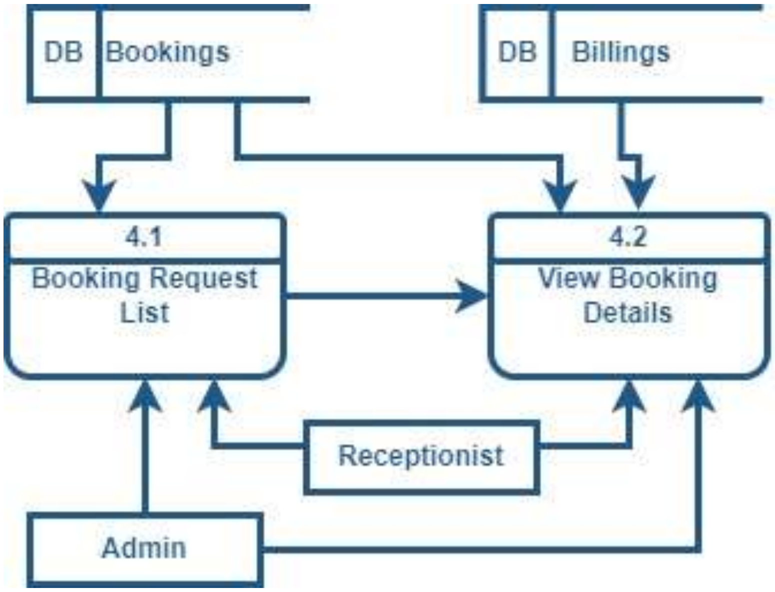


Figure 16 : Level 2 DFD (Booking Details)

6.5.3 Level 2 Process (Manage Rooms)

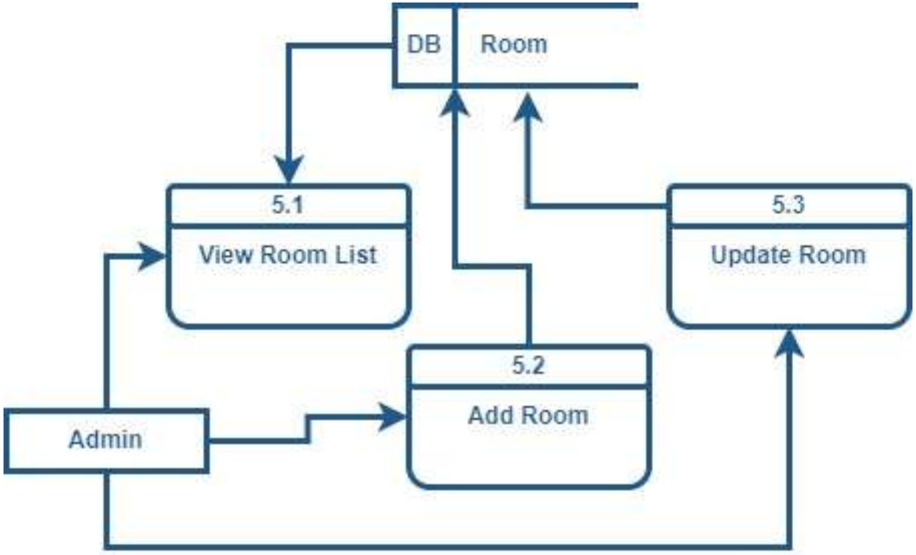


Figure 17 : Level 2 DFD (Manage Rooms)

6.5.3 Level 2 Process (Manage User)

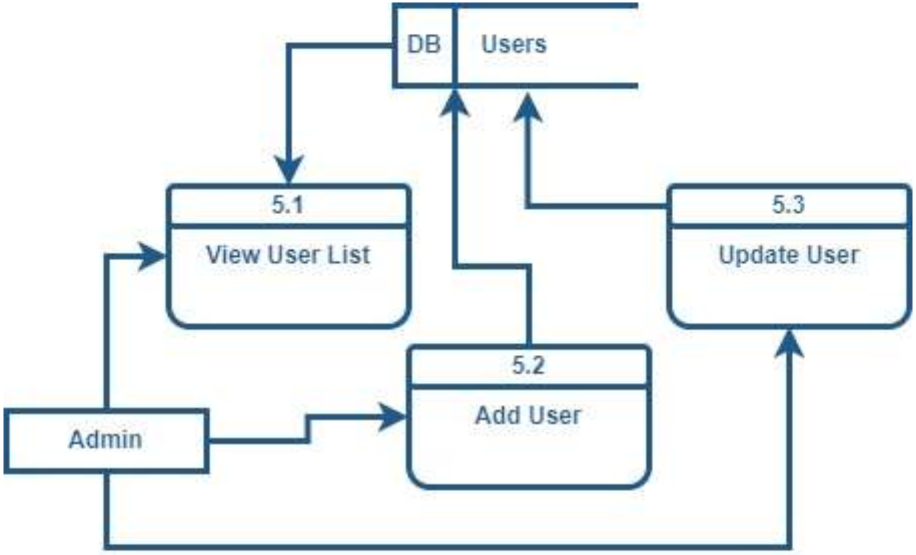
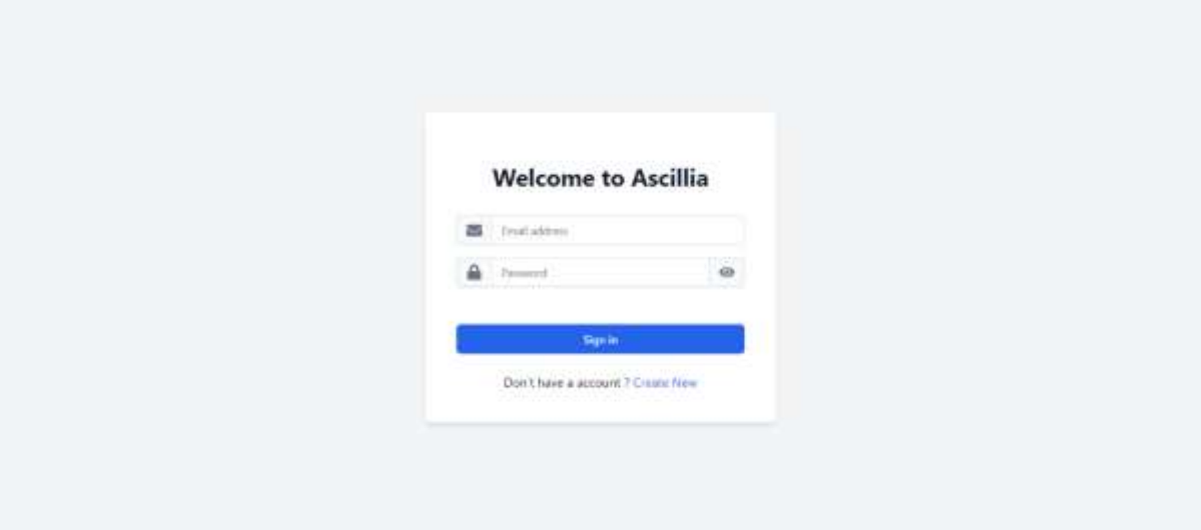


Figure 18 : Level 2 DFD (Manage User)

Chapter 7

Interface Design

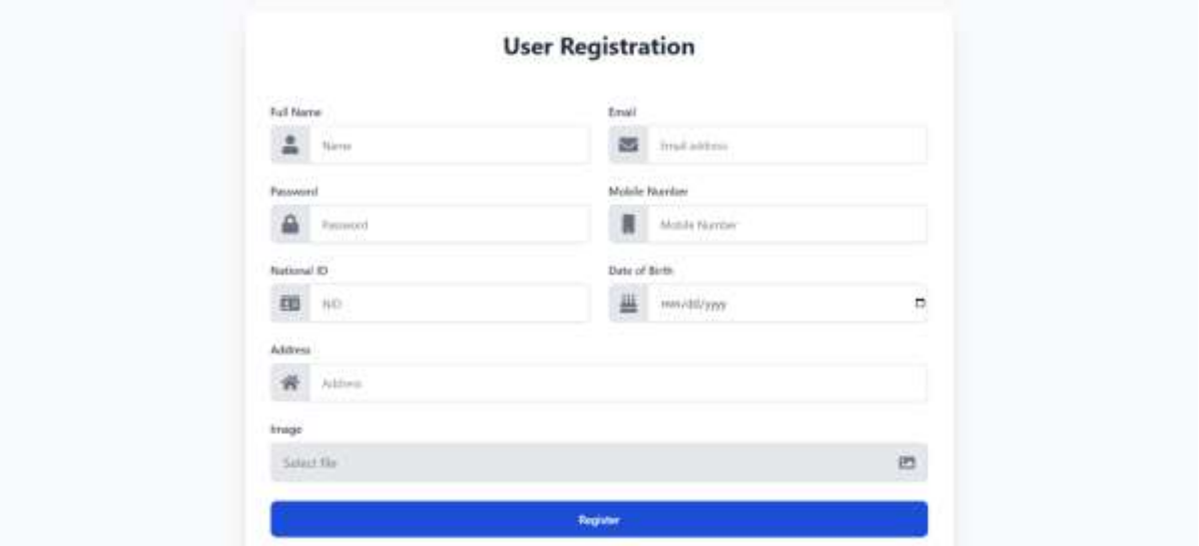
7.1 Sign in page



Welcome to Ascillia

Figure 19 : Sign in Page

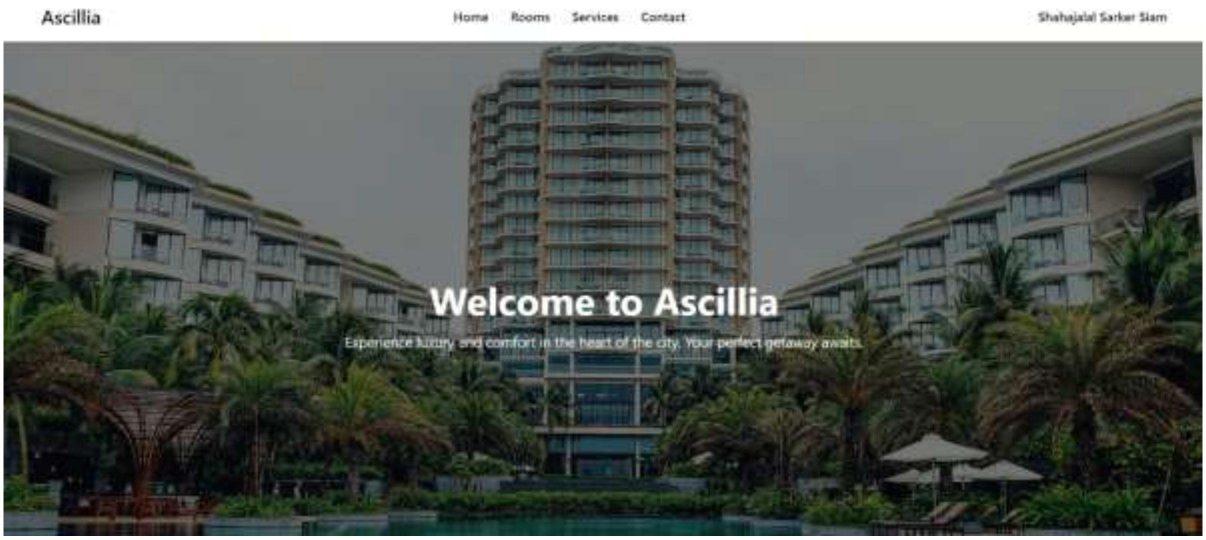
7.2 Sign up page



User Registration

Figure 20 : Sign up Page

7.3 User Home Page

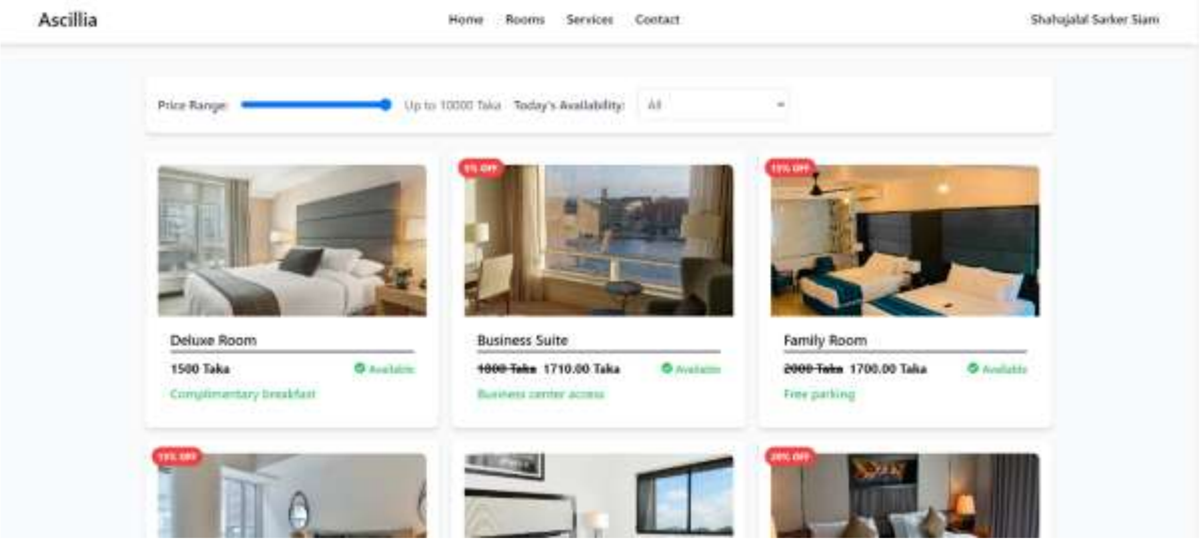


Welcome to Ascillia

Experience luxury and comfort in the heart of the city. Your perfect getaway awaits.

Figure 21 : User Home Page

7.4 Rooms Page



Price Range: Up to 10000 Taka Today's Availability: All

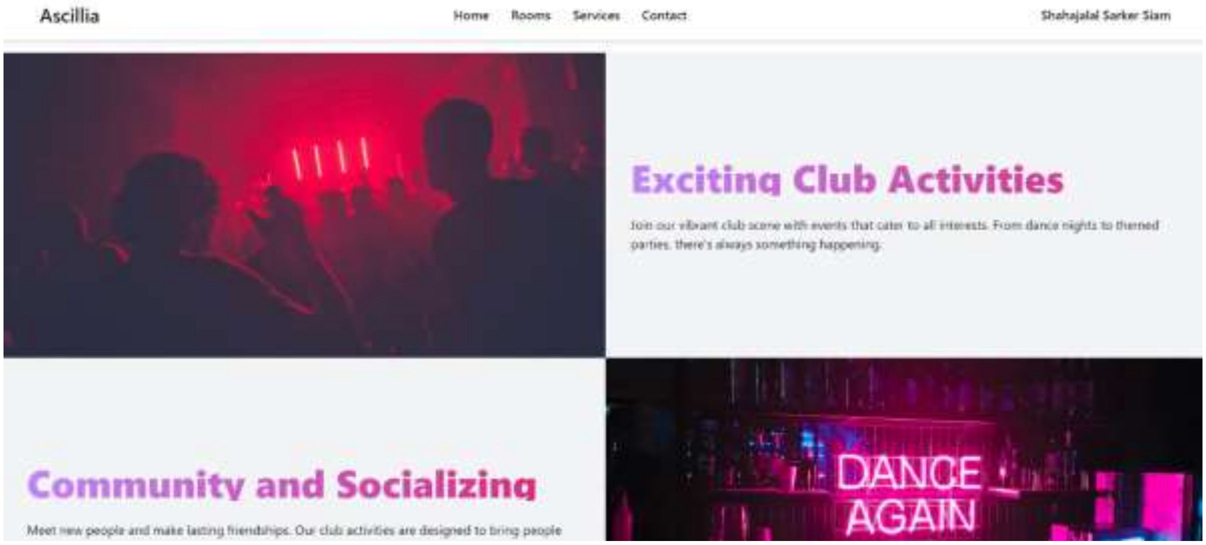
Deluxe Room
1500 Taka ✓ Available
Complimentary breakfast

Business Suite
~~1800 Taka~~ 1710.00 Taka ✓ Available
Business center access

Family Room
~~2000 Taka~~ 1700.00 Taka ✓ Available
Free parking

Figure 22 : Rooms Page

7.5 Services Page



Exciting Club Activities

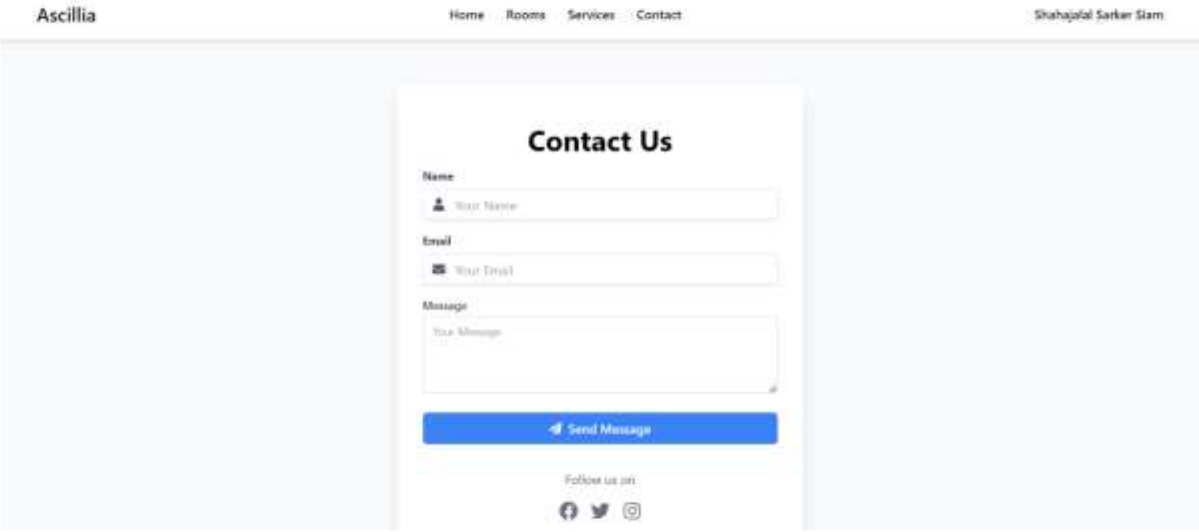
Join our vibrant club scene with events that cater to all interests. From dance nights to themed parties, there's always something happening.

Community and Socializing

Meet new people and make lasting friendships. Our club activities are designed to bring people

Figure 23 : Services Page

7.6 Contact Page



Contact Us

Name

Your Name

Email

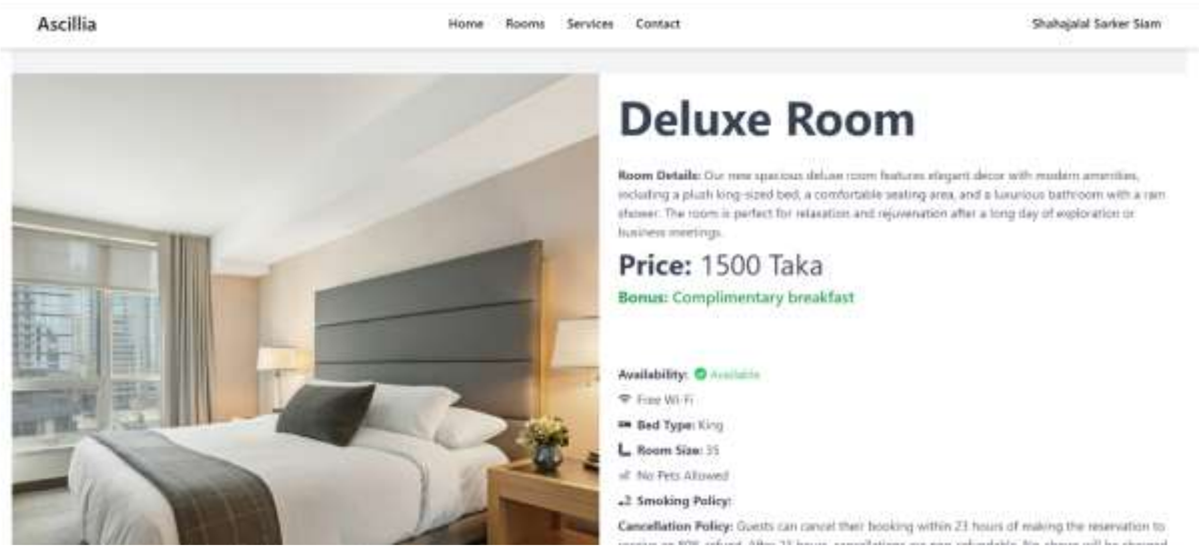
Your Email

Message

Your Message

Figure 24 : Contact Page

7.7 Room Details Page



Deluxe Room

Room Details: Our new spacious deluxe room features elegant decor with modern amenities, including a plush king-sized bed, a comfortable seating area, and a luxurious bathroom with a rain shower. The room is perfect for relaxation and rejuvenation after a long day of exploration or business meetings.

Price: 1500 Taka

Bonus: Complimentary breakfast

Availability: ✓ Available

Free Wi-Fi

Bed Type: King

Room Size: 35

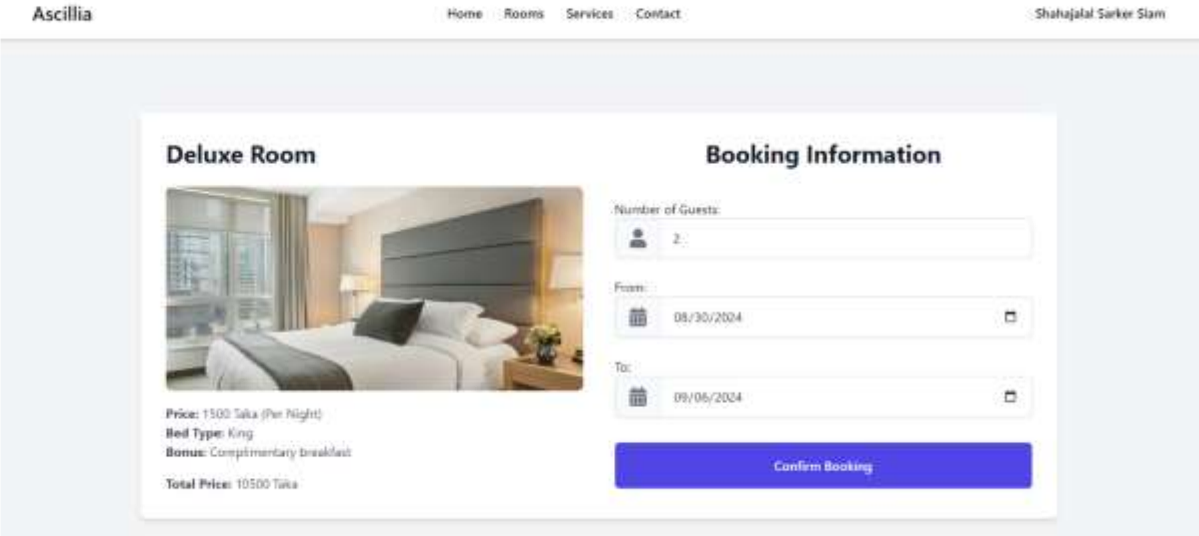
No Pets Allowed

Smoking Policy:

Cancellation Policy: Guests can cancel their booking within 23 hours of making the reservation to

Figure 25 : Room Details Page

7.8 Booking Info Page



Deluxe Room

Price: 1500 Taka (Per Night)
Bed Type: King
Bonus: Complimentary breakfast

Total Price: 10500 Taka

Booking Information

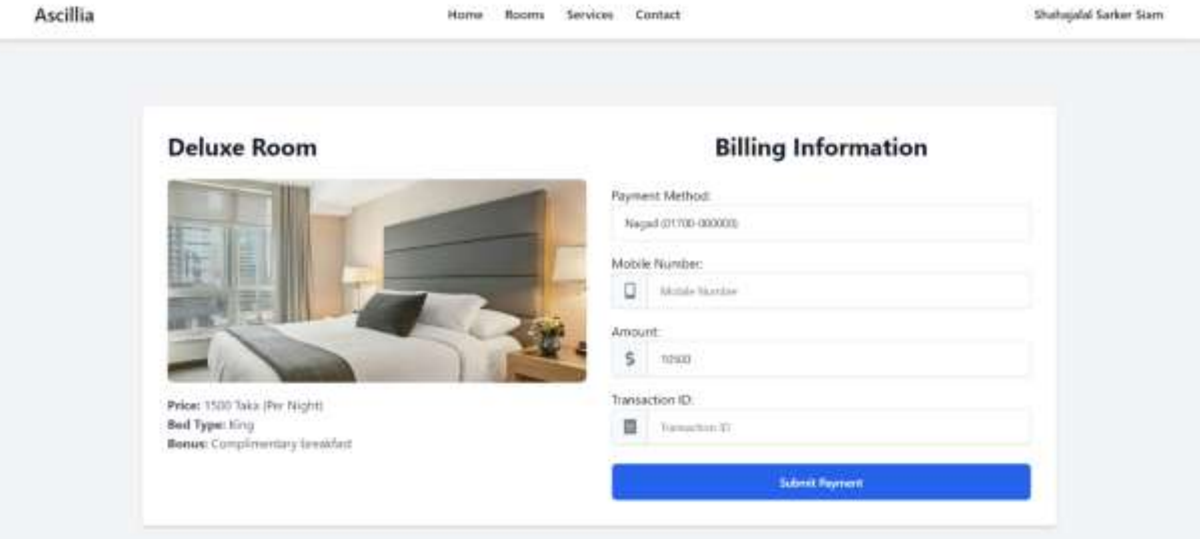
Number of Guests: 2

From: 08/30/2024

To: 09/06/2024

Figure 26 : Booking Info Page

7.9 Billing Form



Deluxe Room

Price: 1500 Taka (Per Night)
Bed Type: King
Bonus: Complimentary breakfast

Billing Information

Payment Method: Nagad (01700-000000)

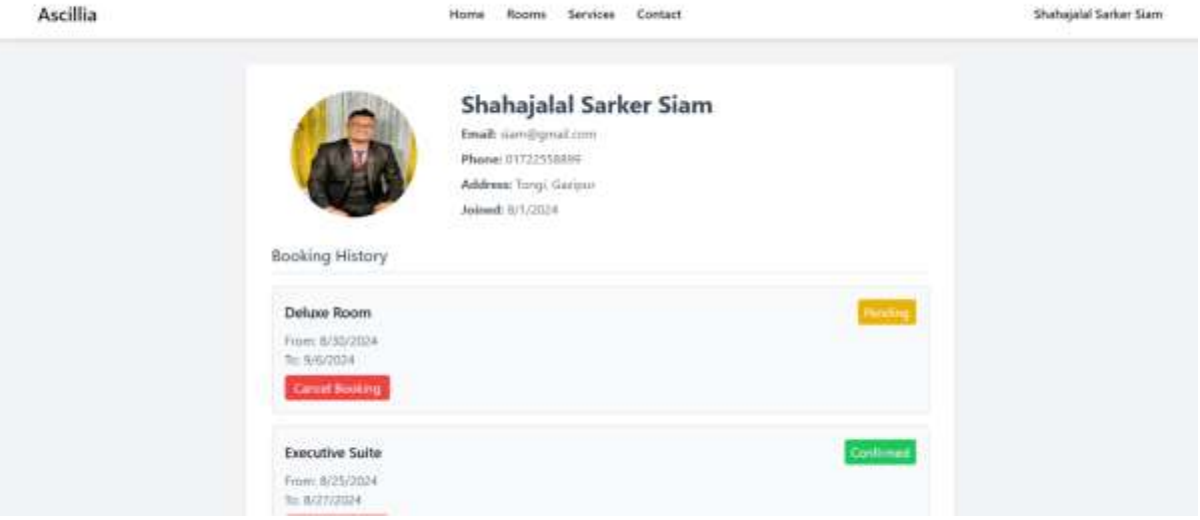
Mobile Number: Mobile Number

Amount: $ 10500

Transaction ID: Transaction ID

Figure 27 : Billing Info Page

7.10 User Profile



Shahajalal Sarker Siam

Email: siam@gmail.com
Phone: 01722558889
Address: Tongi, Gazipur
Joined: 8/1/2024

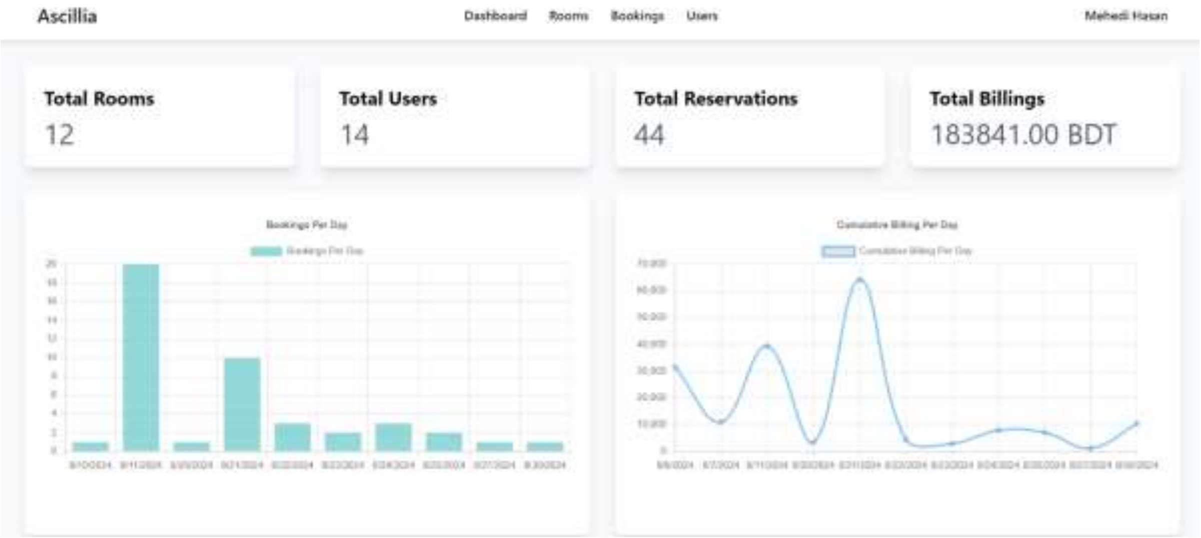
Booking History

Deluxe Room
From: 8/30/2024
To: 9/6/2024

Executive Suite
From: 8/25/2024
To: 8/27/2024

Figure 28 : User Profile

7.11 Admin Dashboard



Total Rooms
12

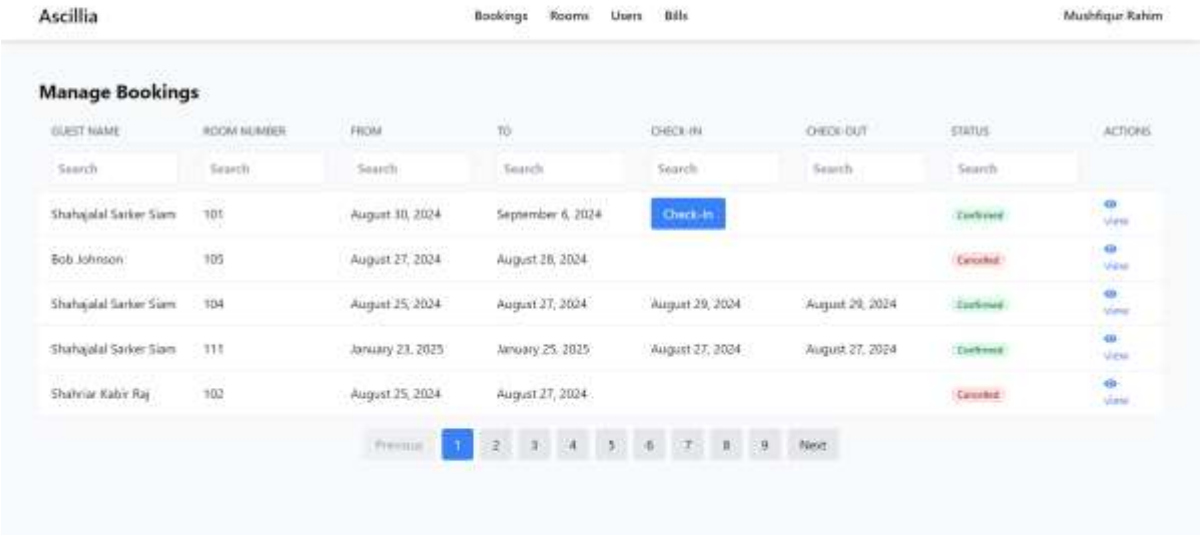
Total Users
14

Total Reservations
44

Total Billings
183841.00 BDT

Figure 29 : Admin Dashboard

7.12 Booking List

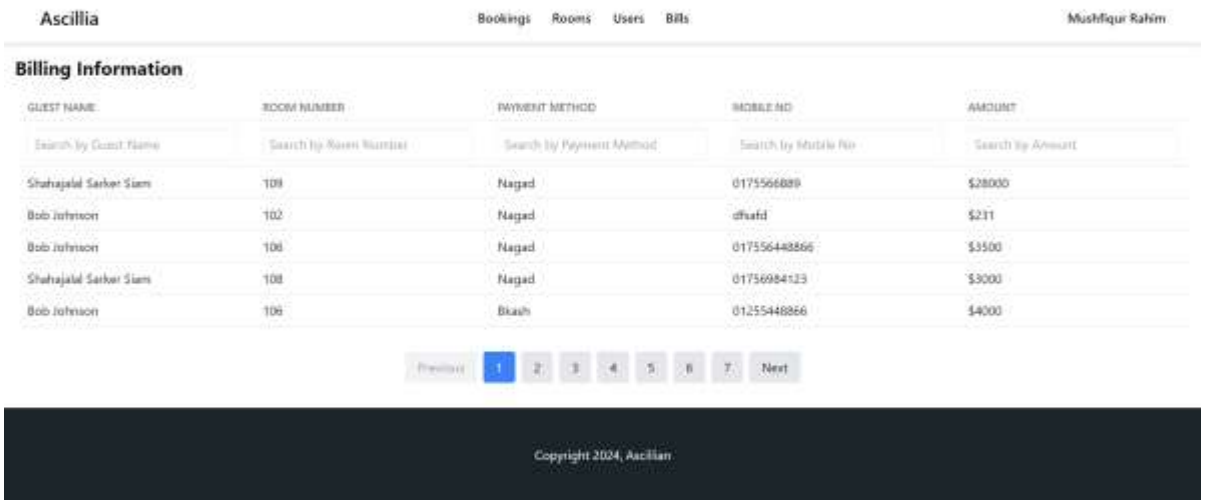


Manage Bookings

GUEST NAME	ROOM NUMBER	FROM	TO	CHECK-IN	CHECK-OUT	STATUS	ACTIONS
Search	Search	Search	Search	Search	Search	Search	
Shahajalal Sarker Siam	101	August 30, 2024	September 6, 2024	Check-In		Confirmed	View
Bob Johnson	105	August 27, 2024	August 28, 2024			Canceled	View
Shahajalal Sarker Siam	104	August 25, 2024	August 27, 2024	August 29, 2024	August 29, 2024	Confirmed	View
Shahajalal Sarker Siam	111	January 23, 2025	January 25, 2025	August 27, 2024	August 27, 2024	Confirmed	View
Shahriar Kabir Raj	102	August 25, 2024	August 27, 2024			Canceled	View

Figure 30 : Booking List

7.13 Billing List



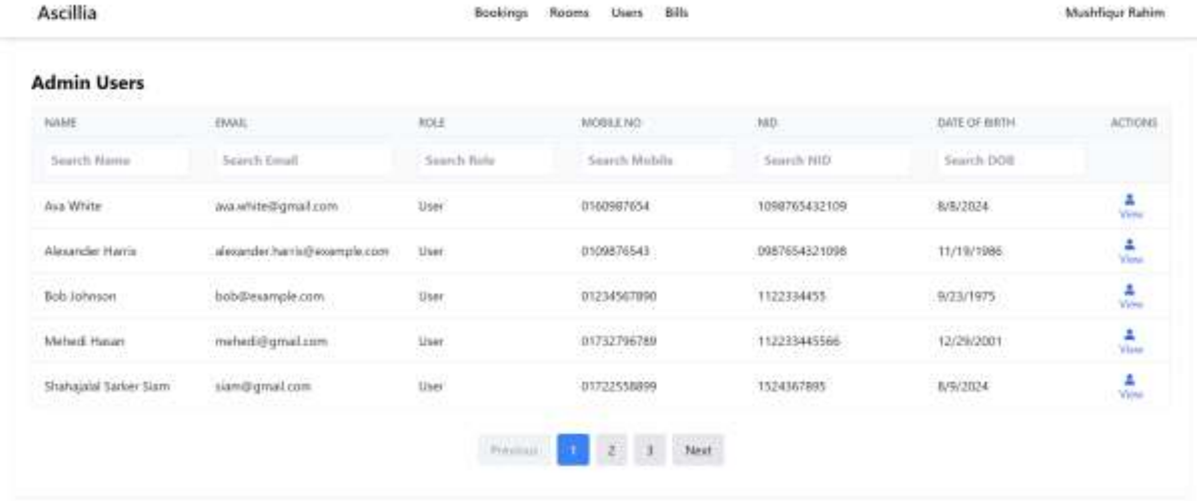
Billing Information

GUEST NAME	ROOM NUMBER	PAYMENT METHOD	MOBILE NO	AMOUNT
Search by Guest Name	Search by Room Number	Search by Payment Method	Search by Mobile No	Search by Amount
Shahajalal Sarker Siam	109	Nagad	0175566889	$28000
Bob Johnson	102	Nagad	dfsafd	$231
Bob Johnson	106	Nagad	017556448866	$3500
Shahajalal Sarker Siam	108	Nagad	01756984123	$3000
Bob Johnson	106	Bkash	01255448866	$4000

Copyright 2024, Ascilian

Figure 31 : Billing List

7.14 Users List

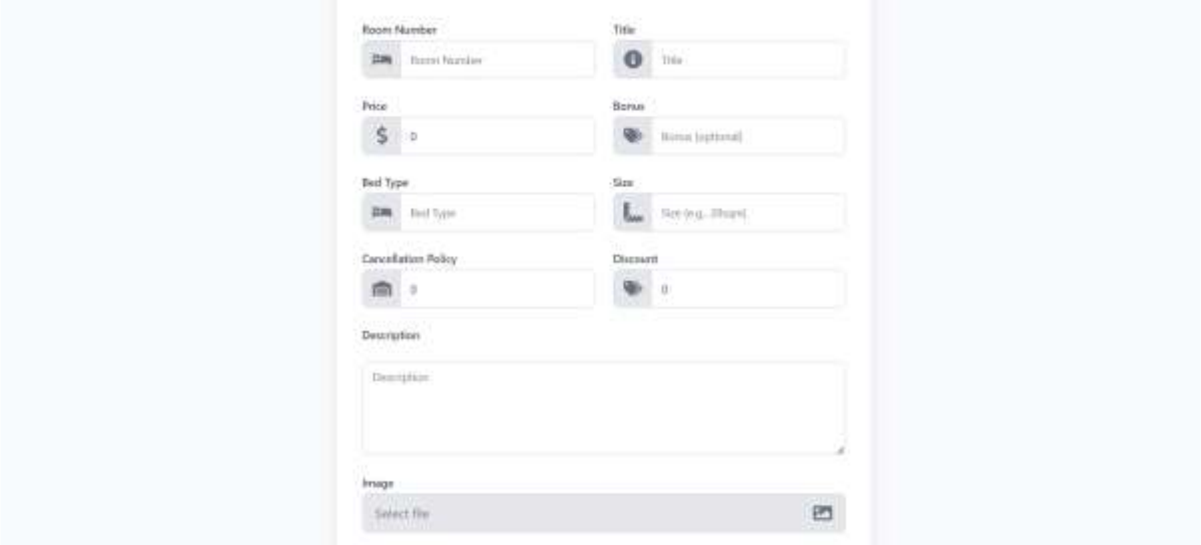


Admin Users

NAME	EMAIL	ROLE	MOBILE NO	NID	DATE OF BIRTH	ACTIONS
Search Name	Search Email	Search Role	Search Mobile	Search NID	Search DOB	
Ava White	ava.white@gmail.com	User	0160987654	1098765432109	8/8/2024	View
Alexander Harris	alexander.harris@example.com	User	0109876543	0987654321098	11/19/1986	View
Bob Johnson	bob@example.com	User	01234567890	1122334455	9/23/1975	View
Mehedi Hasan	mehedi@gmail.com	User	01732796789	112233445566	12/29/2001	View
Shahajalal Sarker Siam	siam@gmail.com	User	01722558899	1524367895	8/9/2024	View

Figure 32 : User List

7.15 Add Room Form



Room Number: Room Number

Title: Title

Price: 0

Bonus: Bonus (optional)

Bed Type: Bed Type

Size: Size (e.g., 30sqm)

Cancellation Policy: 0

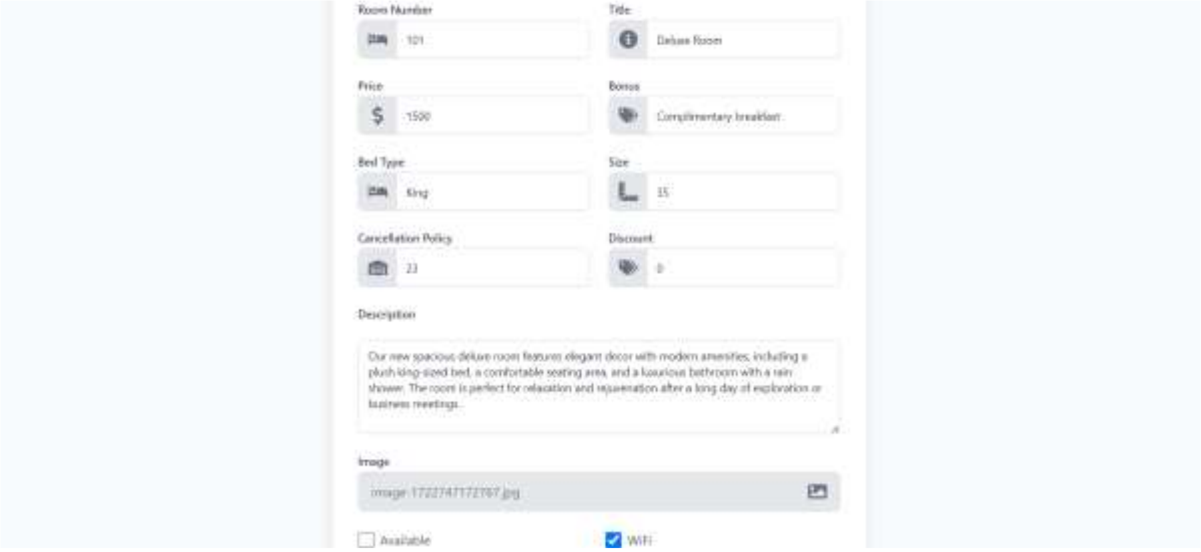
Discount: 0

Description: Description

Image: Select file

Figure 33 : Add Room Page

7.16 Update Room Form



Room Number: 101

Title: Deluxe Room

Price: 1500

Bonus: Complimentary breakfast

Bed Type: King

Size: 35

Cancellation Policy: 23

Discount: 0

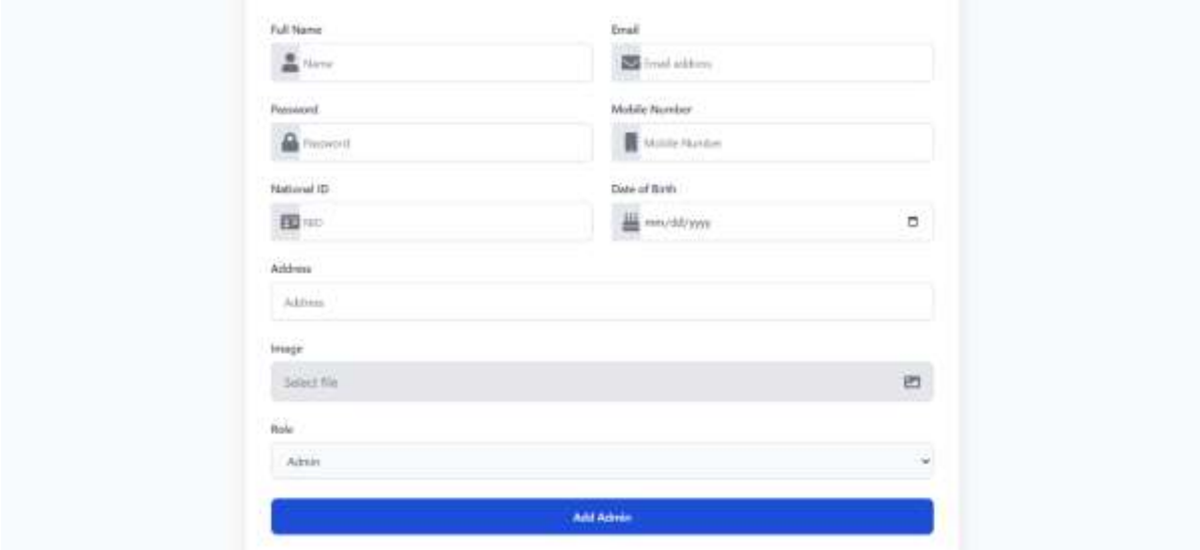
Description: Our new spacious deluxe room features elegant decor with modern amenities, including a plush king-sized bed, a comfortable seating area, and a luxurious bathroom with a rain shower. The room is perfect for relaxation and rejuvenation after a long day of exploration or business meetings.

Image: image-1722747172767.jpg

□ Available ☑ WiFi

Figure 34 : Update Room Form

7.17 Add User Form



Full Name: Name

Email: Email address

Password: Password

Mobile Number: Mobile Number

National ID: NID

Date of Birth: mm/dd/yyyy

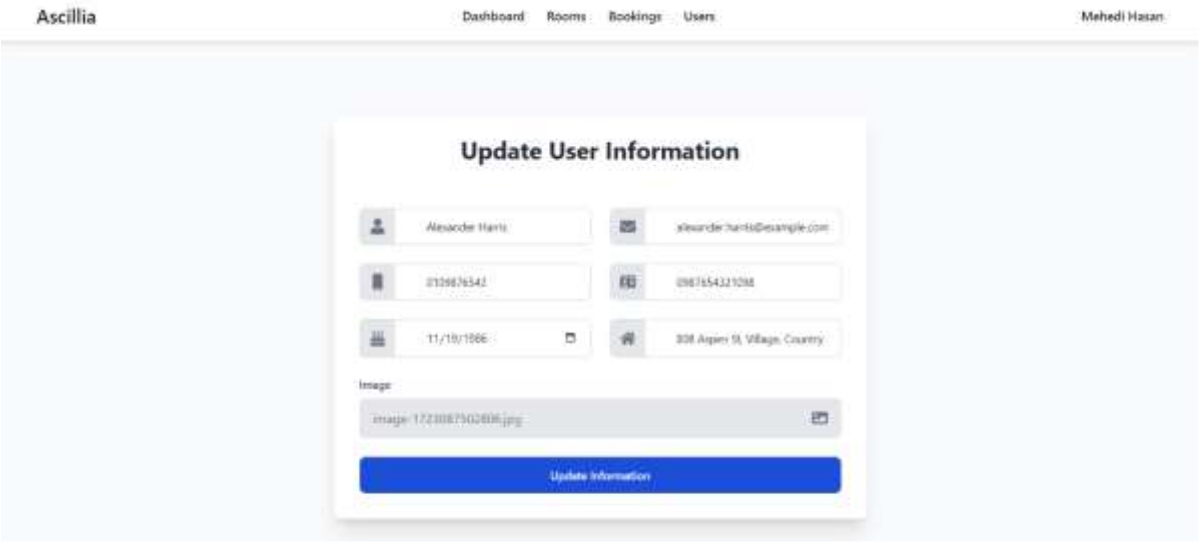
Address: Address

Image: Select file

Role: Admin

Figure 35 : Add User Form

7.18 Update User Form



Update User Information

Alexander Harris

alexander.harris@example.com

0109876542

0987654321098

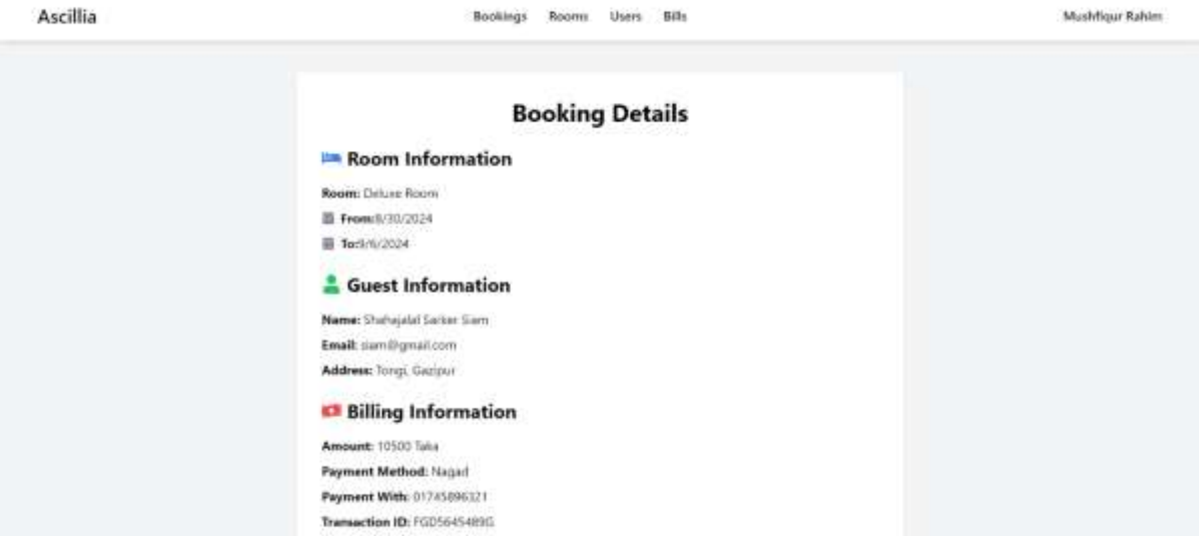
11/19/1986

308 Aspen St, Village, Country

Image: image-1723087502806.jpg

Figure 36 : Update User Form

7.19 Booking Details Form



Booking Details

Room Information

Room: Deluxe Room

From:8/30/2024

To:9/6/2024

Guest Information

Name: Shahajalal Sarker Siam

Email: siam@gmail.com

Address: Tongi, Gazipur

Billing Information

Amount: 10500 Taka

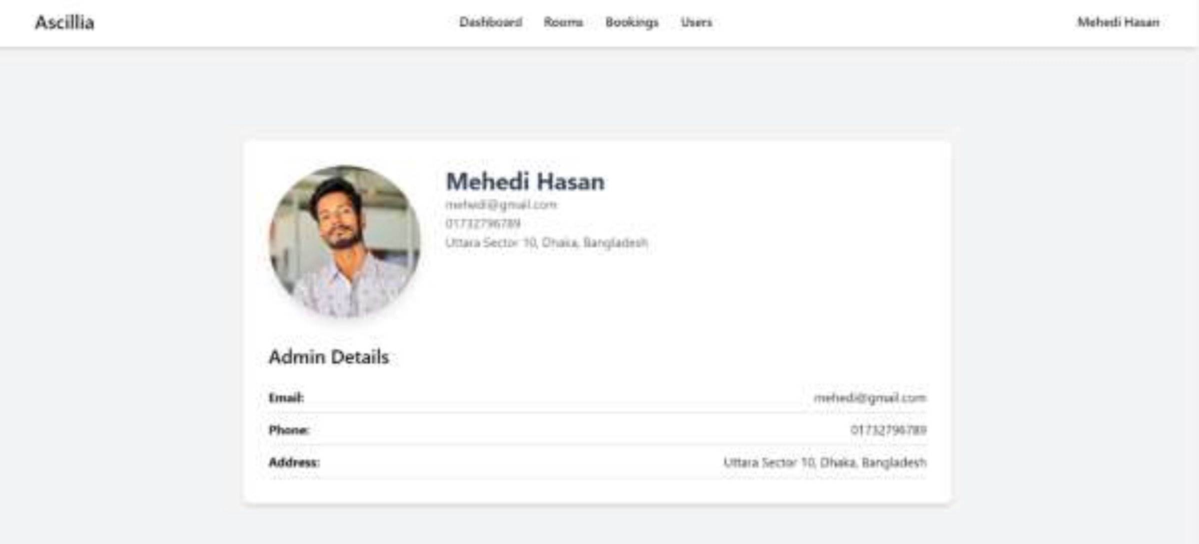
Payment Method: Nagad

Payment With: 01745896321

Transaction ID: FGD5645489G

Figure 37 : Booking Details Page

7.20 Admin Profile



Mehedi Hasan
mehedi@gmail.com
01732796789
Uttara Sector 10, Dhaka, Bangladesh

Admin Details

Email:	mehedi@gmail.com
Phone:	01732796789
Address:	Uttara Sector 10, Dhaka, Bangladesh

Figure 38 : Admin Profile Page

Chapter 8

Testing

8.1 System Testing

System testing is a subset of testing that evaluates the application or system as a whole. After integration testing is finished, it entails assessing the system overall in relation to the requirements and specifications. System testing is done to confirm that the system is functional and prepared for use.

Verification:

System testing consists of two main parts: verification and validation. Verification is the process of checking systems or software to make sure there are no errors and that they fulfill requirements. The design and code must be examined in order to identify mistakes, inconsistencies, or missing requirements. Verification is sometimes known as static testing.

Validation:

Validation is the process of evaluating the software or system to ensure that it meets the actual aims and objectives of the stakeholders. It involves executing the system or software and comparing the results against what was expected. Validation is sometimes known as dynamic testing. In system testing, both validation and verification are essential. While verification serves to ensure that the system is suitably built, validation ensures that the system meets the needs and goals of the stakeholders.

Functional testing, performance testing, stress testing, usability testing, and security testing are a few methods utilized in system testing. These techniques help evaluate the system's many components to ensure that it is safe, dependable, and able to function well in a range of situations.

Software Testing Strategy:

"Software testing strategy" refers to the process or blueprint that outlines how software will be tested to ensure that it meets quality standards and is meant for its intended application.

In the testing procedure used for this software project,

- Unit testing
- Integration testing
- Validation testing

8.2 System Testing Methodology

"Black box" software testing ignores the internal code structure and just concentrates on functionality. With this method, testers rely on the system's input and output to confirm its behavior rather than having access to the internal design and code of the application they are testing. Black box testing is a method used to evaluate the external behavior of the system and confirm that the software satisfies the specified requirements and performs as intended.

Table 8 : Testing Scenario 1

Testing Scenario No. 1	
Scenario	User Registration Testing Case
Input	User details (Name, Email, Password, Phone Number)
Desired Output	New users will be successfully registered in the system after submitting valid information for all required fields.
Actual Output	The system registers new users correctly and redirects them to the login page.
Verdict	By comparing the desired output with the actual output, the registration process works as expected, effectively registering new users.

Table 9 : Testing Scenario 2

Testing Scenario No. 2	
Scenario	User Login Testing Case
Input	Registered Email and Password
Desired Output	Registered users will be successfully logged in and directed to the home page.
Actual Output	Users log in successfully and are redirected to the home page as expected.
Verdict	By comparing the desired output with the actual output, the system is verified to work accurately in logging users in.

Table 10 : Testing Scenarion 3

Testing Scenario No. 3	
Scenario	Room Filtering on Rooms Page
Input	Filter criteria (Room type, Price range, Availability, etc.)
Desired Output	Rooms displayed should match the selected filter criteria, allowing users to find suitable rooms easily.
Actual Output	The system accurately displays rooms matching the specified filters.
Verdict	The actual output aligns with the desired output, confirming that the filtering function operates correctly and efficiently.

Table 11 : Testing Scenario 4

Testing Scenario No. 4	
Scenario	Booking Information Submission
Input	Booking details (Check-in date, Check-out date, Number of guests, Special requests)
Desired Output	Booking information should be successfully submitted, and the user should be directed to the billing page.
Actual Output	The booking information is submitted successfully, and the user is directed to the billing page.
Verdict	Comparing the desired and actual outputs verifies that the booking submission process is working as designed.

Table 12 : Testing Scenario 5

Testing Scenario No. 5	
Scenario	Payment and Booking Confirmation
Input	Payment information (Payment method, Mobile number, Amount, Transaction ID)
Desired Output	The system should process the payment and confirm the booking, updating the user profile with booking history.
Actual Output	Payment is processed, and the booking is confirmed, reflecting correctly in the user’s profile.
Verdict	By contrasting the desired output with the actual output, it is determined that the payment and booking confirmation functionalities are operating correctly.

Chapter 9

Conclusion

9.1 Conclusion

The Online Hotel Management System (OHMS) developed using the MERN stack effectively addresses the challenges of managing hotel operations digitally. The system provides a comprehensive and user-friendly platform that streamlines booking, room management, and user interactions, enhancing both the customer experience and operational efficiency for hotel staff.

Through rigorous testing, the system has proven its capability to handle key functionalities, including user registration, login, room filtering, booking submission, and payment processing. Each module operates as expected, with seamless transitions between user actions and system responses, ensuring a smooth and reliable user journey.

The OHMS's integration of critical components such as real-time booking updates, room availability tracking, and detailed billing management reflects a well-rounded approach to modern hotel management needs. The system's design prioritizes both user convenience and administrative control, providing essential tools for administrators, receptionists, and end-users alike.

Overall, the project successfully meets its objectives by providing a robust, efficient, and scalable solution for hotel management. It not only enhances the guest experience by simplifying the booking and payment process but also empowers hotel staff with better control over daily operations. This practicum has demonstrated the potential of leveraging modern web technologies to innovate within the hospitality industry, setting a solid foundation for further enhancements and future developments in hotel management systems.

9.2 Limitation

While the Online Hotel Management System (OHMS) offers a comprehensive solution for managing hotel operations, several limitations were identified during the development and testing phases:

1. **Limited Scalability for Large-Scale Operations**: The current system is designed primarily for small to medium-sized hotels. Scaling the application to handle a larger number of users, rooms, or bookings simultaneously may require additional optimizations and infrastructure upgrades.

2. **Dependency on Internet Connectivity**: The system relies heavily on internet access for real-time updates and data synchronization. Any disruption in connectivity can impact the user experience, particularly during critical operations like booking confirmations or payments.
3. **Basic Security Measures**: Although basic security protocols such as authentication and data validation are implemented, the system lacks advanced security features like two-factor authentication, encryption of sensitive data, and robust protection against cyber threats. This could expose the system to potential security vulnerabilities.
4. **Limited Payment Integration**: Currently, the system supports a limited number of payment methods (e.g., Bkash, Nagad, Rocket). The lack of integration with international payment gateways like PayPal or credit card processors restricts the system's usability for international users.
5. **Lack of Advanced Analytics and Reporting**: While the admin dashboard provides basic statistics such as occupancy rates and user counts, it lacks advanced analytics capabilities. More detailed insights and reports on revenue, customer behavior, and predictive analytics could significantly enhance decision-making for hotel management.
6. **Manual Room Availability Management**: Although the system updates room availability based on bookings, it does not automatically handle scenarios such as maintenance, cleaning schedules, or other temporary unavailability, requiring manual intervention by the admin.
7. **No Mobile Application**: The current system is a web-based application optimized for desktops and mobile browsers, but it lacks a dedicated mobile app. This limits the convenience for users who prefer a native app experience for smoother navigation and faster access.
8. **Limited Multilingual Support**: The system is primarily designed for a single language interface, which may not cater to a diverse range of international guests. Multilingual support could improve user accessibility and broaden the system's appeal.
9. **Absence of Automated Notifications**: The system does not currently support automated email or SMS notifications for booking confirmations, reminders, or payment receipts, which could enhance communication with users.

10. **Inadequate Handling of Special Requests**: The booking process does not fully accommodate special requests like dietary restrictions, special accommodations, or extra services, which could enhance the guest experience.

Addressing these limitations in future iterations of the OHMS will improve its robustness, security, scalability, and overall user experience, making it a more versatile and powerful tool for hotel management.

9.3 Future Plan

To enhance the functionality, scalability, and user experience of the Online Hotel Management System (OHMS), the following future plans have been identified:

1. **Scalability Enhancements**: Upgrade the system architecture to support large-scale hotel operations, including the ability to handle high traffic, multiple properties, and a large number of bookings. This could involve implementing microservices, optimizing database performance, and leveraging cloud-based infrastructure for better load management.
2. **Advanced Security Features**: Strengthen the system's security by implementing advanced features such as two-factor authentication (2FA), end-to-end encryption of sensitive data, role-based access control, and regular security audits. These measures will enhance protection against unauthorized access and cyber threats.
3. **Integration with International Payment Gateways**: Expand payment options by integrating international payment gateways such as PayPal, Stripe, and credit card processors. This will cater to a broader range of customers and provide a more flexible and secure payment experience.
4. **Development of a Mobile Application**: Launch a dedicated mobile application for both Android and iOS platforms to improve accessibility and user convenience. A mobile app will provide a smoother and faster user experience, with offline capabilities for certain features like room browsing and booking history.

5. **Multilingual Support**: Add multilingual support to cater to a global audience, allowing users to select their preferred language. This will make the system more user-friendly for international guests and improve overall engagement.
6. **Automated Notifications and Reminders**: Implement automated email and SMS notifications for booking confirmations, reminders, cancellations, and special offers. This will enhance communication with users and improve customer satisfaction by keeping them informed at every step of the booking process.
7. **Enhanced Analytics and Reporting**: Develop advanced analytics features to provide detailed insights into hotel performance, customer behavior, and revenue trends. This could include interactive dashboards, predictive analytics, and custom report generation to support better decision-making for hotel management.
8. **AI-Powered Room Recommendations**: Integrate AI algorithms to provide personalized room recommendations based on user preferences, booking history, and seasonal trends. This feature will enhance the user experience by helping guests find the most suitable rooms quickly.
9. **Integration with Third-Party Services**: Expand system capabilities by integrating with third-party services such as travel agencies, car rentals, and local attractions booking platforms. This will create a more comprehensive experience for guests, offering them additional services directly through the system.
10. **Automated Room Management**: Implement automated room management features to handle scenarios like maintenance, cleaning schedules, and temporary room unavailability. This will reduce manual intervention and ensure that room availability information is always accurate.
11. **Enhanced Guest Experience Features**: Introduce features like virtual room tours, real-time chat support, loyalty programs, and feedback systems to improve the overall guest experience and encourage repeat bookings.
12. **Eco-Friendly Options and Sustainability Tracking**: Add features that allow guests to choose eco-friendly options, such as opting out of daily housekeeping or selecting rooms with sustainable amenities. Include tracking of these choices to promote sustainability in hotel operations.

These future plans aim to transform the OHMS into a more powerful, secure, and user-centric platform, positioning it as a leading solution in the hospitality industry. By continuously enhancing the system based on user feedback and emerging technological trends, OHMS will offer a superior hotel management experience for both guests and administrators.

9.4 Ethical Consideration

Ethical considerations in software development are crucial to ensuring that the product is not only functional but also responsible, fair, and sustainable. In the context of an "Online Hotel Management System," various ethical concerns must be addressed throughout the development process, from protecting user data to considering the broader impact on stakeholders and society.

9.4.1 Ethical Considerations in the Software Development Process

1. **Data Privacy and Security**
   Protecting users’ personal and financial data is paramount in any system that handles sensitive information, such as a hotel management system. Developers must implement secure authentication, encryption, and data storage protocols to prevent unauthorized access, data breaches, and misuse. The system should comply with relevant data protection laws, such as GDPR or CCPA, ensuring that users' data is handled responsibly and deleted upon request or when no longer needed.
2. **Intellectual Property**
   Respecting the intellectual property rights of others is critical in software development. Developers must ensure that all third-party libraries, frameworks, and tools used in the system are properly licensed. Furthermore, any proprietary code or designs should be protected from unauthorized copying or misuse. Open-source contributions must be credited appropriately, and any original content within the system must not violate the intellectual property rights of other organizations or individuals.
3. **User Impact**
   The system must be designed with the end user in mind, ensuring it is easy to navigate, secure, and meets the users' needs. In the hotel management system, this includes

providing clear information about room availability, ensuring transparency in the booking process, and preventing any manipulative practices, such as hidden fees. Accessibility for users with disabilities should also be considered to create an inclusive experience.

4. **Bias and Fairness**
   In designing any software, it is essential to avoid introducing bias into the system. This could occur in how rooms are recommended to users or how pricing models are set. Developers must ensure that any algorithms or features are designed in a way that is fair to all users and does not discriminate based on race, gender, nationality, or any other factor. Fairness extends to ensuring equal access to services for all users, regardless of their background.
5. **Responsibility to Stakeholders**
   The development team has a responsibility to all stakeholders, including hotel owners, staff, users, and investors. The software must meet business requirements while respecting user privacy and delivering value to end-users. Transparent communication with stakeholders throughout the development process helps to ensure that their needs are met without compromising ethical standards.
6. **Professional Responsibility**
   Developers must adhere to professional codes of conduct and ethical standards, such as those outlined by organizations like the ACM (Association for Computing Machinery) or the IEEE (Institute of Electrical and Electronics Engineers). This includes maintaining integrity, ensuring the quality and security of the software, and being accountable for any issues or bugs that arise. It also means being transparent about potential limitations and challenges that the software might face.

9.4.2 Sustainability in the Software Development Process

1. **Economic Sustainability**
   Economic sustainability refers to ensuring that the software is developed in a cost-effective way while delivering long-term value. The "Online Hotel Management System" must be designed to optimize operational efficiency for the hotel, reducing costs related to manual booking management or customer support. By automating processes like room management, bookings, and billing, the system can contribute to the financial

sustainability of the hotel. Additionally, the software should be scalable to accommodate future growth without requiring extensive redevelopment or expensive upgrades.

2. **Social Sustainability**

   Social sustainability in software development refers to the positive social impact the system can have on users, employees, and society as a whole. The hotel management system can support this by enhancing customer satisfaction through an easy-to-use interface and fair booking practices. Additionally, by providing job tools for hotel staff (like receptionists and administrators), the system can help improve their efficiency and job satisfaction. Ensuring fair treatment for all users and employees, regardless of background, contributes to social sustainability.

3. **Environmental Sustainability**

   While software systems themselves do not have a direct environmental footprint, they can contribute to environmental sustainability by optimizing operations that reduce waste. For example, by digitizing booking processes, the hotel can reduce its use of paper for receipts, forms, and other administrative tasks. The software can also be hosted on energy-efficient cloud platforms that prioritize renewable energy. Moreover, features like remote check-ins or energy-saving recommendations for users could be integrated to align with the hotel's environmental goals.

By considering these ethical and sustainable principles, the development of the Online Hotel Management System can ensure that it is responsible, fair, and future-proof, contributing positively to both the business and society.

References

GeeksforGeeks. (2021). MERN Stack development: A complete guide. Retrieved from https://www.geeksforgeeks.org

W3Schools. (2023). Building a web application using React and Node.js. Retrieved from https://www.w3schools.com

Tutorialspoint. (2022). Implementing user authentication in MERN stack applications. Retrieved from https://www.tutorialspoint.com

MDN Web Docs. (2023). HTTP authentication with Node.js. Retrieved from https://developer.mozilla.org

JavaTpoint. (2021). How to create a management system using MongoDB and Express.js. Retrieved from https://www.javatpoint.com

DigitalOcean. (2022). Building a RESTful API with Node.js. Retrieved from https://www.digitalocean.com

FreeCodeCamp. (2023). A beginner's guide to setting up a full-stack MERN application. Retrieved from https://www.freecodecamp.org

Stack Overflow. (2023). Best practices for web system design in MERN. Retrieved from https://stackoverflow.com

CSS-Tricks. (2021). Responsive design techniques for web applications. Retrieved from https://css-tricks.com

DEV Community. (2022). Step-by-step guide to building a management system using React and Node.js. Retrieved from https://dev.to