HOTEL BOOKING WEBSITE

PROJECT REPORT

Submitted by

PRERANA REDDY MALE (B192082)

ALLE SINDHUJA (B191852)

LAVANYA VATTIKOTI (B191796)

Of

Bachelor of Technology

Under the guidance of

Mrs.U.NAGAMANI

Asst.Prof.CSE, RGUKT BASAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

BASAR, NIRMAL (DIST), TELANGANA – 504107

HOTEL BOOKING WEBSITE

RGUKT BASAR

Project Report submitted to
Rajiv Gandhi University of Knowledge Technologies, Basar
for the partial fulfillment of the requirements
for the award of the degree of
Bachelor of Technology in
Computer Science & Engineering
by

PRERANA REDDY MALE (B192082)
ALLE SINDHUJA (B191852)
LAVANYA VATTIKOTI (B191796)

Under the Guidance of

Mrs.U.NAGAMANI
Asst.Prof.CSE, RGUKT BASAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

BASAR – JULY 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

BASAR

CERTIFICATE

This is to certify that the Mini Project Report entitled **'HOTEL BOOKING WEBSITE'** submitted by **Prerana Reddy Male B192082, Alle Sindhuja B191852 , Lavanya Vattikoti B191796,** Department of Computer Science and Engineering, Rajiv Gandhi University Of Knowledge Technologies, Basar; for partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering; is a bonafide record of the work and investigations carried out by him/her/them under my supervision and guidance.

PROJECT SUPERVISOR:
Mr.B.VENKAT RAMAN
Assistant Professor

HEAD OF THE DEPARTMENT:
Mr.V.REVYA NAIK
Assistant Professor

PROJECT COORDINATOR:
Mrs.U.NAGAMANI
Assistant Professor

EXTERNAL EXAMINER:
Mr.P.LAXMI NARAYANA
Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

BASAR

DECLARATION

We hereby declare that the work which is being presented in this mini project entitled, " **HOTEL BOOKING WEBSITE**" submitted to RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, BASAR in the partial fulfillment of the requirements for the award of the degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING, is an authentic record of my/our own work carried out under the supervision of "Mrs.U.Nagamani", Assistant Professor in Department of Computer Science and Engineering, RGUKT, Basar.

The matter embodied in this project report has not been submitted by me/us for the award of any other degree.

Place: Basar

Date: 02-08-2024

PRERANA REDDY MALE (B192082)

ALLE SINDHUJA (B191852)

LAVANYA VATTIKOTI (B191796)

ACKNOWLEDGEMENT

We would like to express our deep gratitude to our project guide **Mrs.U.NAGAMANI**, Assistant Professor, Department of Computer Science and Engineering, RGUKT BASAR, for her guidance with unsurpassed knowledge and immense encouragement.

We are grateful to Mr. REVYA NAIK, Head of the Department, Computer Science and Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the Director and Administration, RGUKT, BASAR for their encouragement and cooperation to carry out this work.

We express our thanks to all teaching faculty of Department of CSE, whose suggestions during reviews helped us in accomplishment of our project.

We would like to thank all non-teaching staff of the Department of CSE, RGUKT, BASAR for providing great assistance in accomplishment of our project.

We would like to thank our parents, friends, and classmates for their encouragement throughout our project period. At last but not the least, we thank everyone for supporting us directly or indirectly in completing this project successfully.

PROJECT STUDENTS:

PRERANA REDDY MALE (B192082)

ALLE SINDHUJA (B191852)

LAVANYA VATTIKOTI (B191796)

ABSTARCT

The modern hospitality industry demands efficient and user-friendly systems to manage reservations and bookings. Traditional manual booking processes are prone to errors, time-consuming, and often result in double bookings or missed reservations. To address these challenges, this project aims to develop a robust and dynamic hotel booking website using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js.

This hotel booking website allows users to search for available rooms, make reservations, and manage their bookings seamlessly. It also provides hotel administrators with tools to manage room availability, view booking reports, and update hotel information in realtime. The front-end of the application is built with React.js, providing a responsive and interactive user interface. The back-end is powered by Express.js and Node.js, ensuring efficient handling of user requests and business logic. MongoDB is used as the database to store user details, room information, and booking records securely.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

1.1 PROBLEM OBJECTIVE

The objective of the hotel booking website, developed using the MERN stack, is to create a comprehensive platform that streamlines the process of finding and booking accommodations. The website aims to offer an intuitive and user-friendly interface that enhances the user experience by providing advanced search and filter options, enabling users to quickly find hotels that meet their specific requirements.

- The platform is designed to cater to both customers and hotel administrators. Customers can securely book rooms, manage their accounts, and access customer support with ease. The website also includes features for hotel administration, allowing hoteliers to manage bookings, update room availability, and handle payments securely through integrated payment gateways.
- The project prioritizes security and reliability, ensuring that user data and transactions are protected. By leveraging the capabilities of the MERN stack, the website aims to provide a seamless, efficient, and secure booking experience for users, ultimately enhancing the overall process of finding and booking hotel accommodations online.

1.2 PROBLEM STATEMENT

- In the rapidly evolving digital era, the hospitality industry requires an efficient and user-friendly platform for managing hotel bookings. Current booking systems often suffer from fragmented interfaces, outdated availability information, and lack of comprehensive search and filter options, leading to a frustrating user experience. Additionally, hotel administrators face challenges in managing bookings, updating room availability, and handling secure transactions.
- The objective of this project is to develop a robust hotel booking website using the MERN stack that addresses these challenges. The website aims to offer an intuitive interface with advanced search and filter options, ensuring users can easily find accommodations that meet their specific needs. Real-time availability and secure booking features will enhance user confidence and streamline the booking process.
- For hotel administrators, the platform will provide efficient tools to manage bookings, update room statuses, and ensure secure payment processing. By prioritizing security and reliability, the website will protect user data and transaction integrity, offering a seamless and trustworthy experience for both customers and hotel managers.

1.3 SCOPE OF THE PROJECT

- **User Authentication and Management**:
  - Login with Credentials: Users authenticate themselves using their credentials (username and password) to access personalized features.
  - User Roles: Different user roles include regular users (travelers) and administrators (hotel management).
  - User Profiles: Each user can manage their profile information, preferences, and booking history.

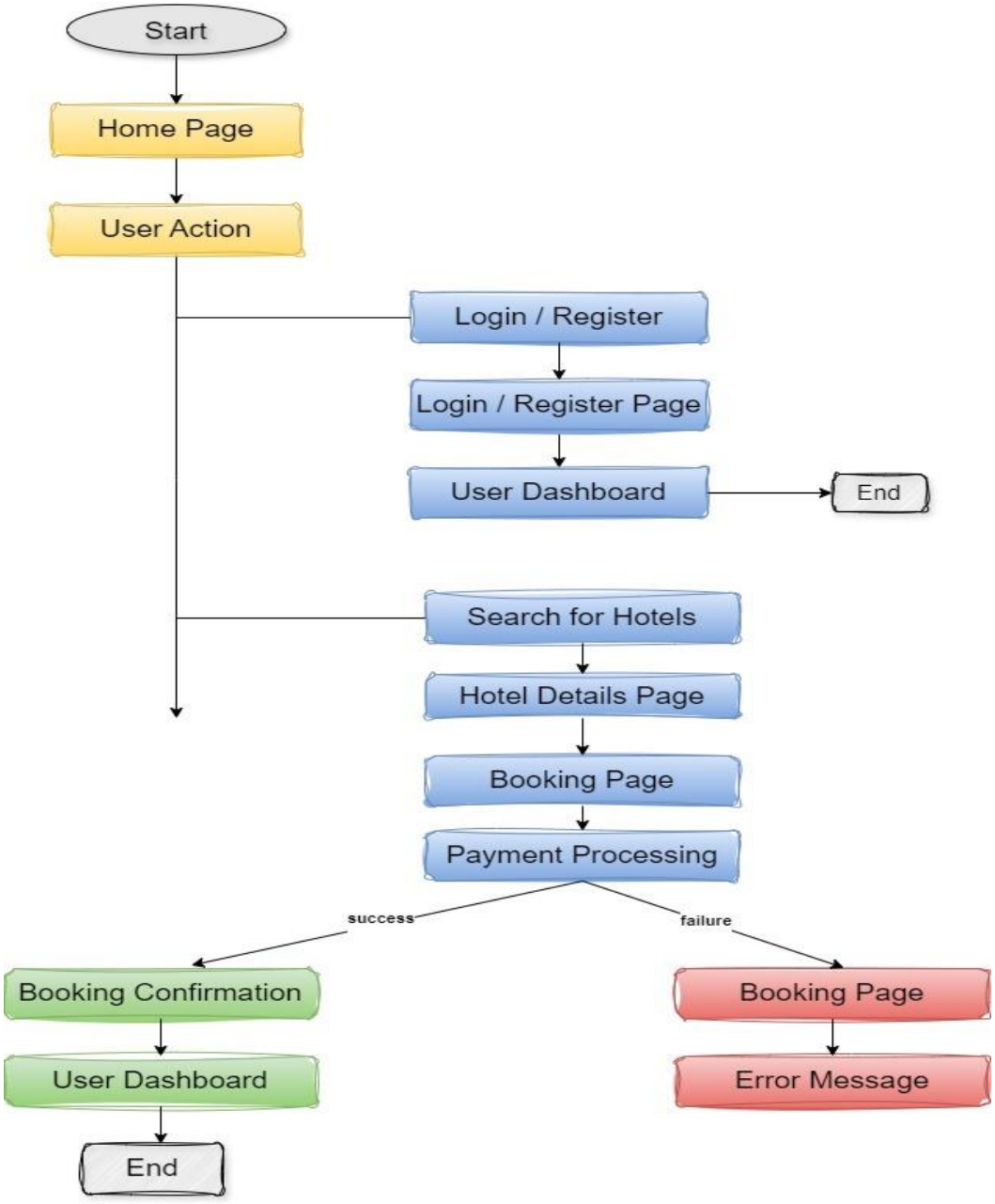
- **Booking and Reservation System**:
  - Room Search and Selection: Users can search for hotels based on criteria such as location, dates, room types, and amenities.
  - Room Details and Booking: Detailed room listings provide information on room types, prices, availability, and amenities. Users can select rooms and proceed to booking.
  - Booking Management: Users can view and manage their bookings, including cancellations and modifications, through their accounts.

- **Administrator Dashboard**:
  - Hotel Management Tools: Administrators have access to a dashboard for managing hotel listings, room availability, pricing, and special offers.
  - Booking Management: Administrators can view and manage booking requests, confirm reservations, and communicate with guests.

- Analytics and Reporting: Tools for generating reports on occupancy rates, revenue, and guest demographics help in decision-making and planning.

- **Secure Payment Processing**:
  - Transaction Security: Measures are in place to ensure the security and encryption of financial transactions to protect user data.
  - Payment Gateway Integration: Integration with secure payment gateways allows users to make online payments safely using credit/debit cards, PayPal, or other methods.

1.4 FLOW CHART



Project Outline

CHAPTER 2 :

CODE IMPLEMENTATION AND ANALYSIS

2.1 PROJECT FOLDER STRUCTURE


SHEYROOMS


CLIENT……..FRONTEND


MODELS……..BACKEND


ROUTES……BACKEND


SERVER……BACKEND


DATABASE…… BACKEND

2.2 CODE IMPLEMENTATION

2.2.1 BACKEND

Technology Stack:

- Node.js: A JavaScript runtime built on Chrome's V8 engine, used for building fast and scalable server-side applications.
- Express.js: A web application framework for Node.js, designed for building web applications and APIs with minimal setup.

Database:

- MongoDB: A NoSQL database known for its scalability and flexibility, used to store user data, hotel information, booking details, and transaction records.

Key Components and Features:

1. **User Authentication and Authorization:**
    - Implement user registration and login functionalities using JWT (JSON Web Tokens) for secure authentication.
    - Role-based access control to differentiate between regular users and hotel administrators.
2. **Hotel Management:**
    - APIs for adding, updating, and deleting hotel information.
    - Real-time updates of room availability and pricing.
    - Features for managing room types, amenities, and other hotel details.
3. **Booking Management:**
    - APIs for searching hotels based on criteria such as location, price, and amenities.
    - Functionality for creating, updating, and canceling bookings.
    - Real-time availability checks to prevent double bookings.
4. **Payment Processing:**
    - Integration with payment gateways (e.g., Stripe, PayPal) to handle secure transactions.

- APIs for processing payments, managing refunds, and handling transaction records.

5. **User Profile Management:**
    - APIs for users to view and update their profiles, including personal details and booking history.
    - Secure storage and management of user data.
6. **Admin Dashboard:**
    - APIs for administrative tasks such as managing users, viewing booking statistics, and handling customer support requests.
    - Tools for generating reports and analytics for better decision-making.
7. **Error Handling and Logging:**
    - Comprehensive error handling to ensure robust and user-friendly error messages.
    - Logging of server activity and errors for monitoring and troubleshooting.

2.2.2 FRONTEND

Technology Stack:

- React**:** A JavaScript library for building user interfaces, known for its component-based architecture and efficiency.

- Redux**:** State management tool to manage the application's state in a predictable way.
- React Router**:** For handling routing within the single-page application.
- Axios**:** For making HTTP requests to the backend API.
- Styled-components**:** For styling the components with scoped and maintainable CSS.

Key Components and Features:

1. User Interface**:**
    - Home Page**:**
        - Attractive and user-friendly design.
        - Featured hotels and promotions.
        - Search bar for finding hotels based on dates.
    - Search Results Page:
        - Display list of hotels based on search criteria.
    - Hotel Detail Page:
        - Detailed information about the selected hotel.
        - High-quality images, room types and amenities.
        - Booking form to select room type, check-in and check-out dates.
    - Booking Confirmation Page:
        - Summary of the booking details.

        - Payment form to enter payment information and complete the booking.

    - User Profile Page**:**
        - View user profile information.
        - Display booking history and manage current bookings.
    - Admin Dashboard**:**
        - Manage hotels, rooms, and bookings.

2. State Management**:**
    - Used Redux or Context API to manage the global state.
    - Handled user authentication state, booking details, and hotel data efficiently.
3. Routing**:**
    - Implemented routing using React Router.
    - Defined routes for the main pages: Home, Search Results, Hotel Detail, Booking Confirmation, User Profile, and Admin Dashboard.
4. Form Handling**:**
    - Used controlled components for handling form inputs.
5. HTTP Requests:
    - Used Axios or Fetch API to make requests to the backend API.
    - Handle loading states, errors, and responses appropriately.

6. Styling:
    - Used styled-components, CSS Modules.
7. User Authentication:
    - Implemented login and registration forms.
    - Used JWT tokens for handling authentication.
    - Protect routes that require authentication (e.g., User Profile, Admin Dashboard).
8. Notifications**:**
    - Display success, error, and informational messages using a notification library (e.g., React SweetAlerts).
9. Performance Optimization**:**
    - Used Hash loading and code splitting to improve performance.
    - Optimize images and assets for faster loading times.

Component Structure:

1. App Component**:**
    - Main component that includes routing and global state providers.
2. Header Component**:**
    - Navigation bar with links to Home, Profile, and Admin Dashboard.
    - User login/logout buttons.
3. Home Component**:**

- Search bar and featured hotels.

4. SearchResults Component**:**
    - List of hotels matching the search criteria.
    - Filters and sorting options.
5. HotelDetail Component**:**
    - Detailed information about the selected hotel.
    - Booking form for selecting room and dates.
6. BookingConfirmation Component**:**
    - Summary of booking details.
    - Payment form.
7. UserProfile Component**:**
    - User information and booking history.
8. AdminDashboard Component**:**
    - Management tools for hotels, rooms, and bookings.

CODE AND OUTPUT SAMPLES:

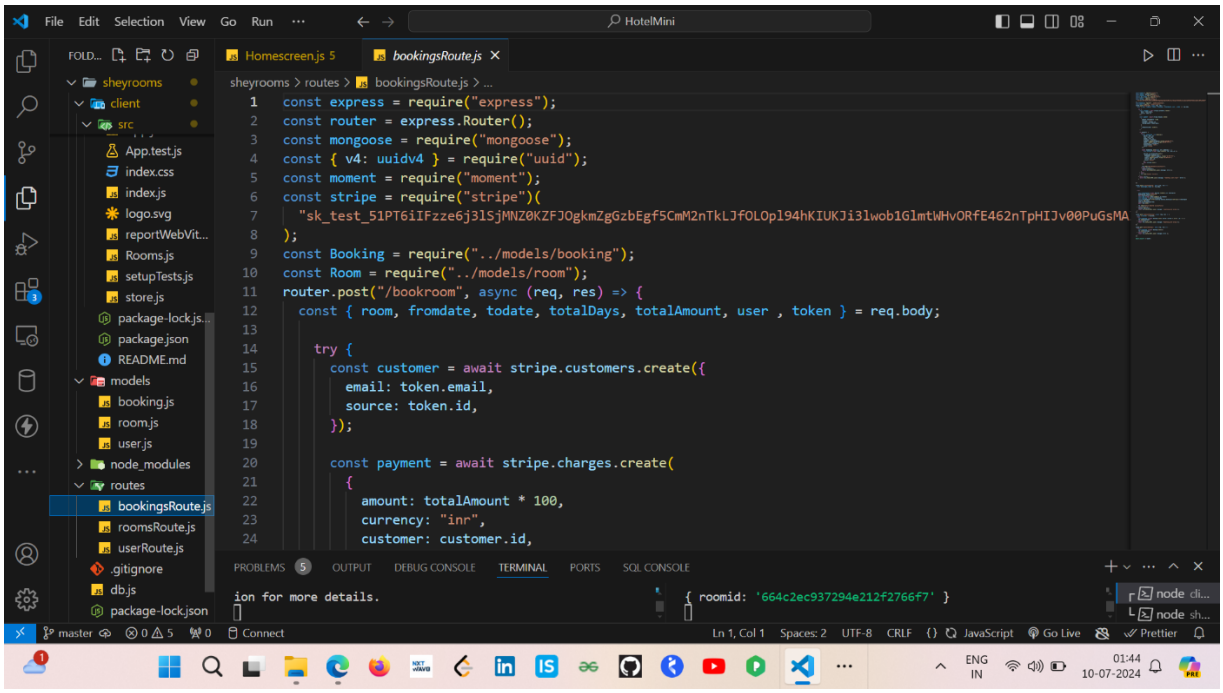


FIGURE 2.2.1 1

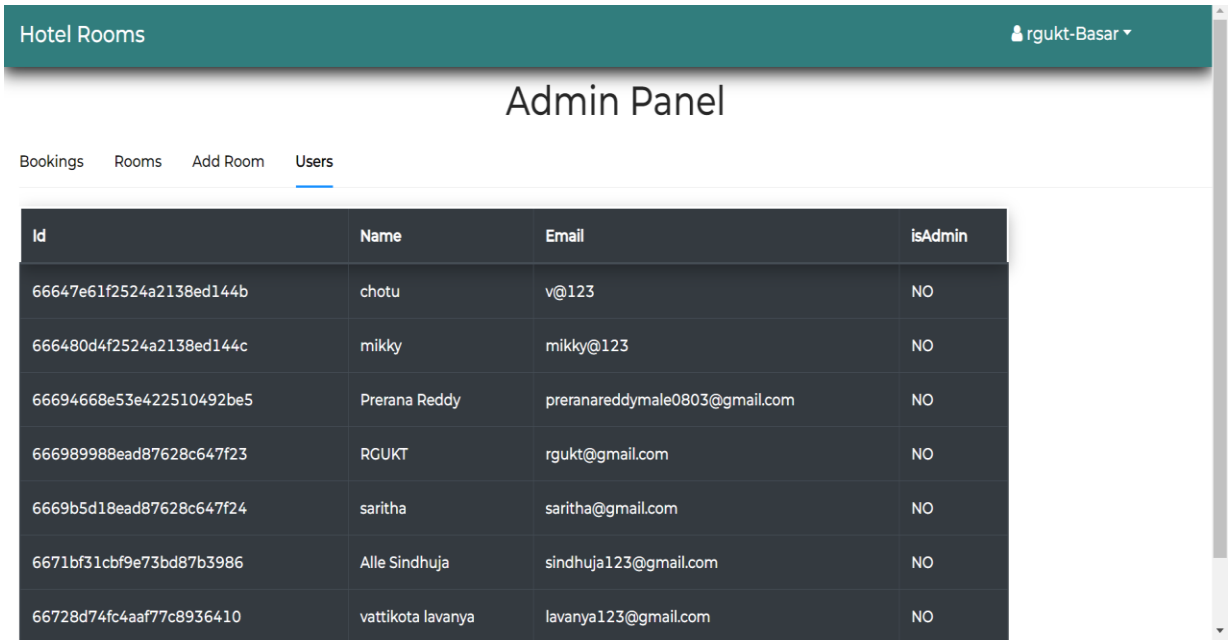


FIGURE 2.2.1 2

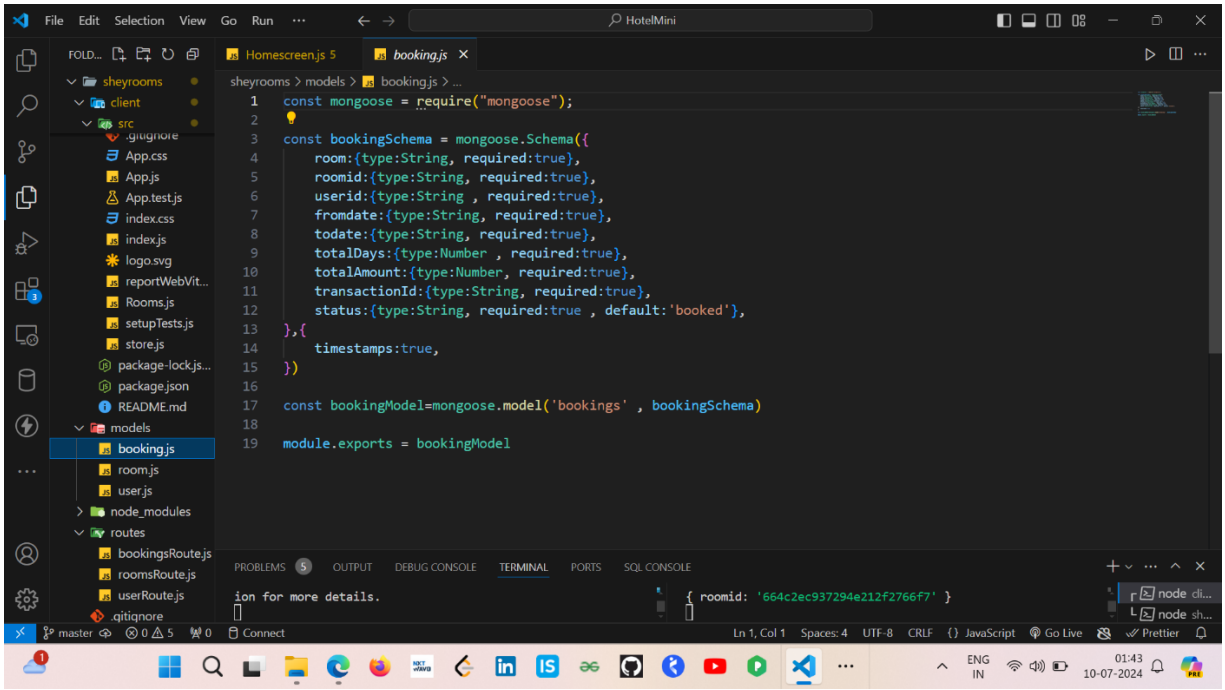


FIGURE 2.2.1 3

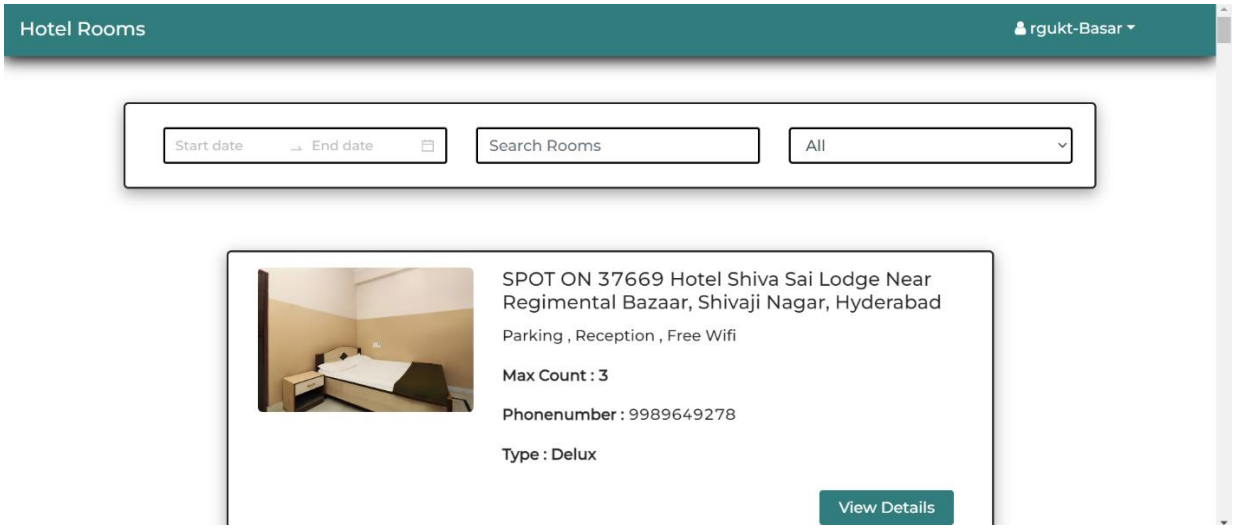


FIGURE 2.2.1 4

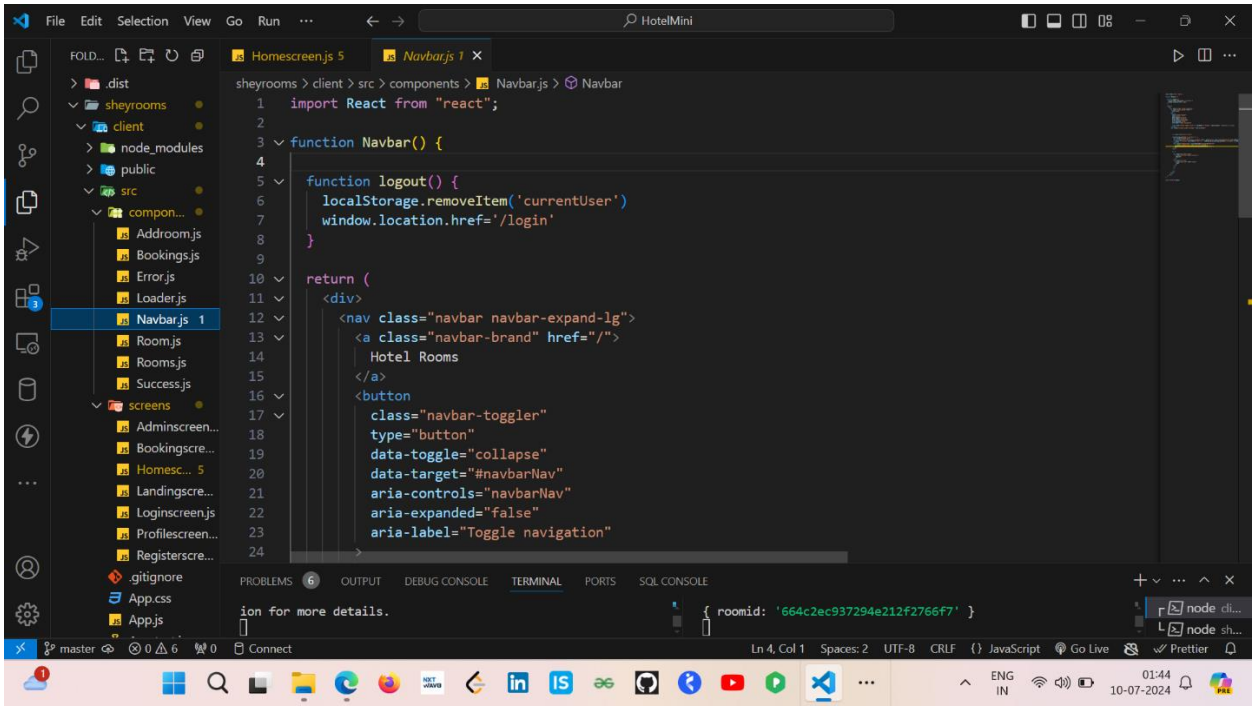


FIGURE 2.2.1 5

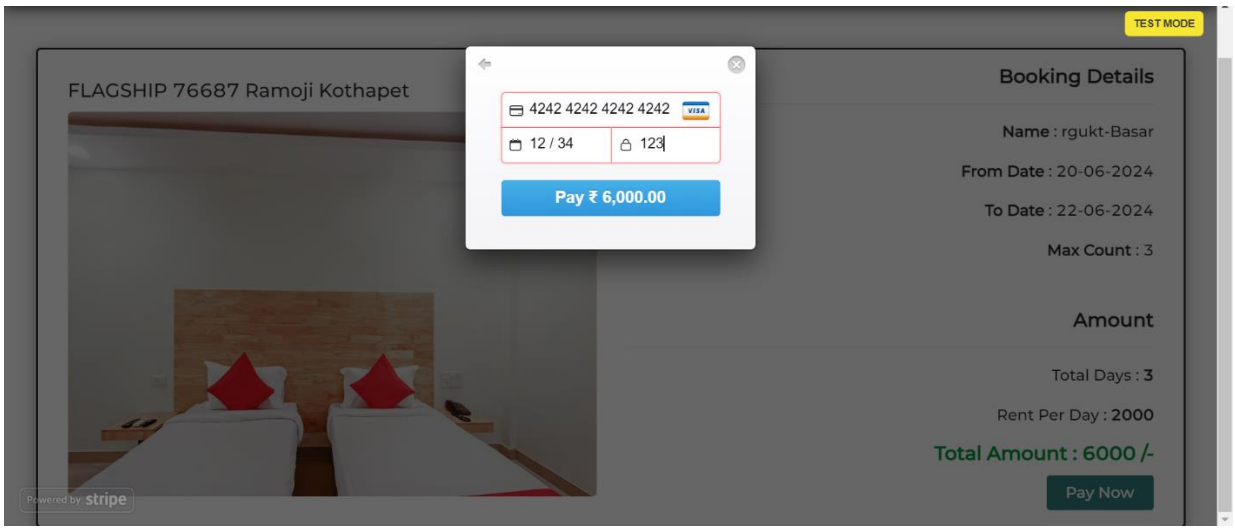


FIGURE 2.2.1 6

CHAPTER 3:

WORK BREAKDOWN STRUCTURE AND REFERENCES

A work plan is a strategic document that outlines the steps, resources, and timeline required to achieve specific goals within a project. It serves as a roadmap, detailing the tasks and activities needed to reach the desired outcomes. Key components of a work plan include objectives, deliverables, milestones, resources (such as personnel, budget, and tools), timelines, and responsibilities. By clearly defining these elements, a work plan ensures that all team members understand their roles and the project's direction. It also aids in monitoring progress, managing risks, and adjusting plans as necessary to stay on track. Ultimately, a well-constructed work plan enhances coordination, accountability, and efficiency, facilitating the successful completion of a project.

1. Project Initiation

- Define project objectives
- Identify stakeholders
- Develop project plan

2. Project Planning

- Detailed requirement gathering
- Define scope and deliverables
- Project scheduling and timeline creation
- Resource planning and allocation

- Risk management plan

3. Backend Development

- **Node.js and Express.js Setup:**
    - Set up project structure and initialize Node.js application
    - Install and configure Express.js
    - Implement middleware for request parsing, logging, etc.
- **Database Design and Integration:**
    - Design MongoDB schema for users, hotels, rooms, bookings, and transactions
    - Connect to MongoDB using Mongoose
    - Implement CRUD operations for each schema
- **API Development:**
    - User authentication and authorization (register, login, JWT token handling)
    - Hotel management (create, read, update, delete hotels and rooms)
    - Booking management (search hotels, create, update, cancel bookings)
    - Payment processing (integrate payment gateways, handle transactions)
    - User profile management (view, update profile, booking history)
    - Admin dashboard APIs (manage users, hotels, rooms, view reports)

- **Security Implementation:**
    - Implement HTTPS, data encryption, and secure authentication
    - Ensure secure data transmission and storage

4. Frontend Development

- **Setup React Application:**
    - Initialize React project using Create React App or Vite
    - Configure project structure and dependencies
- **Component Development:**
    - **Common Components:**
        - Header (navigation bar)
        - Footer (contact information, social media links)
        - Notifications (error, success messages)
    - **User Components:**
        - Home (search bar, featured hotels)
        - Search Results (hotel listings, filters, sorting)
        - Hotel Detail (hotel information, booking form)
        - Booking Confirmation (booking summary, payment form)
        - User Profile (profile details, booking history)
    - **Admin Components:**

        - Admin Dashboard (manage hotels, rooms, bookings)
        - Reports (view statistics, generate reports)

- **State Management:**
    - Implement global state management using Redux or Context API
    - Handle user authentication state, hotel data, and booking details
- **Routing:**
    - Implement routing using React Router
    - Define routes for main pages (Home, Search Results, Hotel Detail, etc.)
- **Styling:**
    - Style components using styled-components, CSS Modules, or Sass
    - Ensure responsive design for various devices

5. Integration and Testing

- **API Integration:**
    - Integrate frontend with backend APIs using Axios or Fetch API
    - Handle loading states, errors, and responses
- **Testing:**
    - Unit testing for individual components and APIs
    - Integration testing to ensure smooth interaction between frontend and backend

CHAPTER 4: CONCLUSION

The hotel booking website project, built using the MERN stack (MongoDB, Express.js, React, and Node.js), aims to revolutionize the way users search for and book accommodations online. By focusing on an intuitive user interface, real-time availability, secure transactions, and robust administrative features, the platform seeks to enhance the overall user experience for both customers and hotel managers.

The backend development focuses on creating a scalable and secure environment with efficient data management and real-time updates. Key functionalities include user authentication, hotel management, booking management, payment processing, and comprehensive security measures to protect user data and transactions.

In conclusion, this hotel booking website project endeavors to provide a seamless, efficient, and secure booking experience. By leveraging the MERN stack, it aims to meet the evolving needs of the hospitality industry, offering a reliable solution that simplifies the booking process for users and enhances management capabilities for hoteliers.