ROOM BOOKING SYSTEM

A PROJECT REPORT

Submitted by

BOOBALAN M (920422205020)

DARIN VIDHU A (920422205022)

SOUNDRA PANDIAN K (920422205105)

KARTHICK M (920422205304)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECNOLOGY

DEPARTMENT OF INFORMATION TECHNOLOGY

KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution - Affiliated to Anna University, Chennai)

K.VELLAKULAM, VIRUDHUNAGAR - 625 701

NOVEMBER 2024

KAMARAJ COLLEGE OF ENGINEERING AND TECHNOLOGY

(An Autonomous Institution- Affiliated to Anna University, Chennai)

K.VELLAKULAM, VIRUDHUNAGAR - 625 701

BONAFIDE CERTIFICATE

Certified that the project report "**ROOM BOOKING SYSTEM MEAN STACK (MONGO DB, EXPRESS JS, ANGULAR, NODE.JS)"** is the bonafide work **of "BOOBALAN M (920422205020) DARIN VIDHU A (920422205022) SOUNDRA PANDIAN K (920422205105) & KARTHICK M (920422205305)"** who carried out the project work under my supervision.

SIGNATURE
Dr. E. VAKAIMALAR
Head of the Department,
Associate Professor,
Dept. of Information Technology,
Kamaraj College of Engg & Tech,
K.Vellakulam,
Virudhunagar - 625 701.

SIGNATURE
Dr. R. ARTHY
SUPERVISOR,
Assistant Professor,
Dept. of Information Technology,
Kamaraj College of Engg &Tech,
K.Vellakulam,
Virudhunagar - 625701.

Abstract

This project presents the design and development of a full-stack web application using the MEAN stack, an innovative combination of MongoDB, Express.js, Angular, and Node.js. The project, titled ROOM BOOKING SYSTEM, focuses on creating a digital platform that facilitates the process of booking rooms in hotels or rental properties. It typically includes a user-friendly interface where customers can browse available rooms, check amenities, and make reservations online. Key features often include real-time room availability tracking, user accounts for booking history, and customer reviews to assist in decision-making. For hotel managers, the system provides tools for room management, booking oversight, and analytics to optimize operations. The integration of mobile applications and payment services further streamlines the booking process, improving customer convenience and expanding reach. Overall, a room booking system enhances the accommodation experience by making it faster, easier, and more accessible. The use of the MEAN stack provides a unified development environment, with JavaScript powering both the frontend and backend. MongoDB, a NoSQL database, stores dynamic room and customer data, while Express.js and Node.js handle server-side logic and API requests. Angular.js delivers a seamless, responsive, and interactive user experience on the frontend.

Acknowledgment

First of all, We would like to thank Lord Almighty for his abundant grace and countless blessings in making this work a great success.

We Wish to express our thanks and sincere gratitude to our Head of the Department **Dr. E. VAKAIMALAR**, for giving this opportunity to do this project as a team.

We would like to thank **Dr. R. ARTHY**, for her valuable advice and guidance in making the project possible

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	3
	LIST OF TABLES	6
	LIST OF FIGURES	8
1	**INTRODUCTION**	**7**
	1.1 HTML	7
	1.2 CSS	8
	1.3 JavaScript	9
	1.4 MEAN Stack	10
2	**METHODOLOGY**	**12**
	2.1 Objective	**12**
	2.2 Problem Statement	**13**
	2.3 Block Diagram	**16**
	2.4 Module Explanation	
3	**RESULTS AND DISCUSSION**	**24**
4	**CONCLUSION**	**25**
5	**REFERENCES**	**26**

LIST OF FIGURES

FIGURE NO.	TITLE	PAGE NO.
2.1	Block Diagram of the project	13
3.1	The Register page of the room booking system	17
3.2	The Login page of the room booking system	19
3.3	The Home page of the room booking system	21

CHAPTER 1

INTRODUCTION

1.1. HTML

In a room booking system built using HTML, the structure typically includes sections for user registration, login, room display, a booking cart, and a reservation form. The registration and login forms collect user details like email, username, and password to manage user authentication. The homepage features a room display where accommodation options, such as single rooms or suites, are listed with descriptions, prices, and buttons to add them to the booking cart. The cart section displays the selected rooms, their quantities, and the total price, allowing users to review their booking before proceeding.

The "Book Now" section collects additional information such as check-in and check-out dates, guest details, and payment method through a form. These elements work together to create a streamlined, user-friendly interface for browsing room options and making reservations, with the data dynamically handled through backend systems connected to a database for storing user, room, and booking information. Additional features could include real-time booking confirmation and a notification system to inform users about their reservation status, enhancing the overall user experience.

1.2. CSS

In a room booking system, CSS plays a crucial role in creating a visually appealing and responsive user interface. It helps style the layout by organizing sections such as room listings, booking cart, and reservation forms, ensuring a clean and user-friendly design. CSS is used to define consistent typography, button styles, and color schemes that align with the brand's identity. Flexbox or CSS Grid layouts allow for easy alignment of items, making the room display or cart items responsive across different screen sizes.

Hover effects and transitions on buttons provide an interactive experience, while form fields for login, registration, and booking are styled for clarity and accessibility. Media queries ensure the system works seamlessly on both desktop and mobile devices, improving usability. CSS also enhances the user experience by maintaining a balance between aesthetics and functionality, allowing users to easily browse and interact with the system. Lastly, CSS animations can be added to highlight important actions like booking confirmations or cart updates.

1.3. JAVASCRIPT

JavaScript is essential in a room booking system for handling dynamic interactions and enhancing the overall user experience. It enables real-time updates, such as adding or removing rooms from the booking cart without refreshing the page, making the reservation process smoother. JavaScript also manages form validation for registration, login, and booking, ensuring users input valid data before submission. Through event listeners, users can interact with buttons and room listings, with JavaScript handling the logic behind placing bookings, updating quantities, and calculating total prices dynamically.

AJAX or Fetch API is often used to communicate with the backend server, enabling seamless interactions like fetching available rooms from a database or submitting reservations. Additionally, JavaScript can store booking data temporarily using local storage or session storage, allowing users to navigate the site without losing their selections. It also powers user feedback mechanisms, such as displaying booking status, success messages, or error prompts in real time. Integrating JavaScript with a payment gateway further enables secure and efficient payment processing, creating a responsive, interactive, and user-friendly room booking platform.

1.4. MEAN Stack

- **MongoDB**
  - A NoSQL database that stores data in flexible, JSON-like documents.
  - Provides high scalability and performance for handling large amounts of data.
  - Ideal for applications with evolving data structures, like food items and orders.
- **Express.js**
  - A web application framework for Node.js that simplifies server-side development.
  - Facilitates the creation of RESTful APIs for handling requests and responses.
  - Supports middleware functions for streamlined request processing and error handling.

- **Angular**
  - A front-end framework developed by Google for building dynamic web applications.
  - Utilizes a component-based architecture for improved code organization and reusability.
  - Provides two-way data binding for real-time synchronization between the model and view.

- **Node.js**
  - A JavaScript runtime environment that allows server-side scripting.
  - Enables the development of scalable network applications capable of handling multiple connections.
  - Facilitates the use of JavaScript across both the frontend and backend, enhancing efficiency.

The MEAN stack, comprising MongoDB, Express.js, Angular, and Node.js, offers a robust framework for developing a food ordering system. MongoDB serves as the database to store user information, menu items, and order details, providing flexibility and scalability. Express.js facilitates the creation of RESTful APIs that handle backend operations such as user authentication and order processing. Angular powers the frontend, delivering a responsive and interactive user interface for browsing menus and placing orders. Together, the MEAN stack accelerates development, enabling rapid prototyping and the integration of third-party services for secure transactions.

CHAPTER 2

METHODOLOGY

2.1 Objective

The main objective is to build a fully functional room booking system using the MEAN stack. The application must allow users to browse available rooms, manage their bookings, and complete reservations seamlessly.

2.2 Problem Statement

The problem statement given to us is to develop a room booking system website using the MEAN stack, focused on managing and reserving accommodations online.

2.3 Block Diagram

2.3.1.Frontend

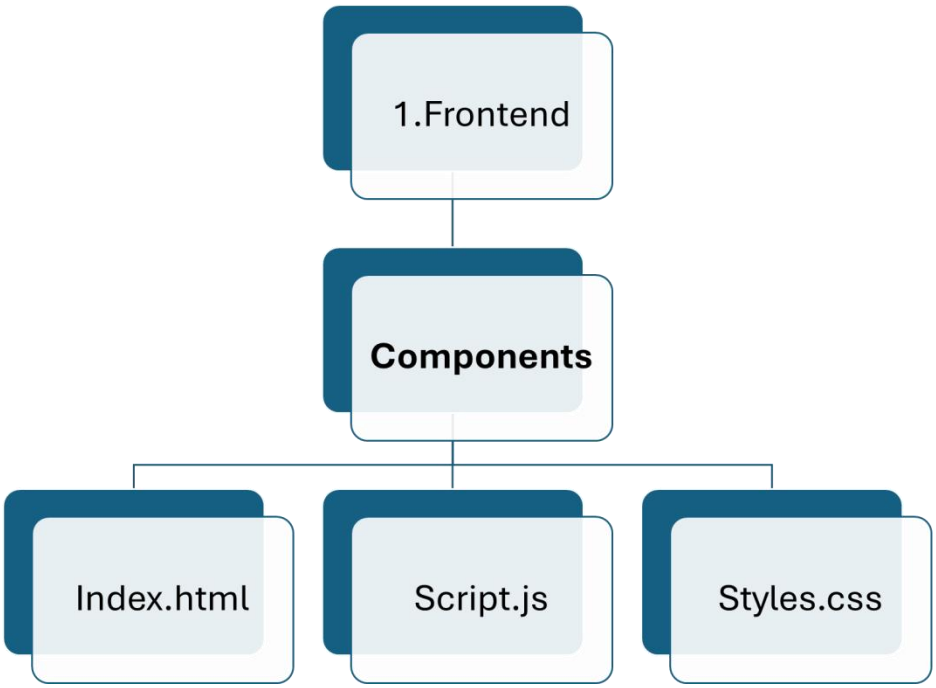


Figure 2.1 Block Diagram for Frontend

2.3.2 Backend

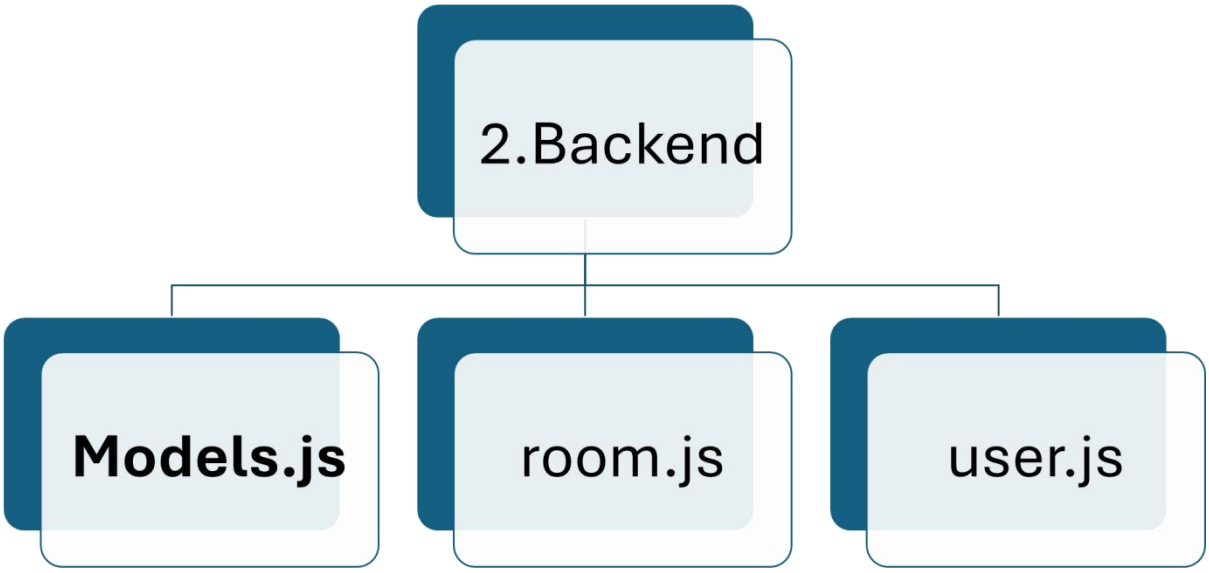


Figure 2.2 Block Diagram for Backend

2.3.3 Routes

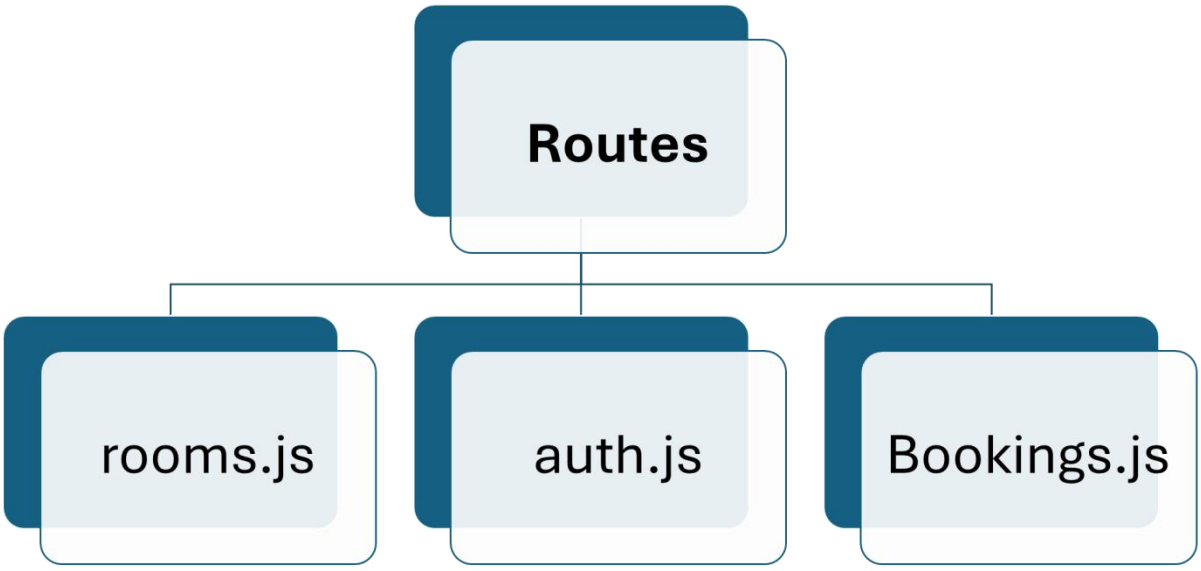


Figure 2.3 Block Diagram for Routes

2.4 Module Explanation

1. Login page

2. Register page

3. Room booked

4.Setup page

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Screenshots

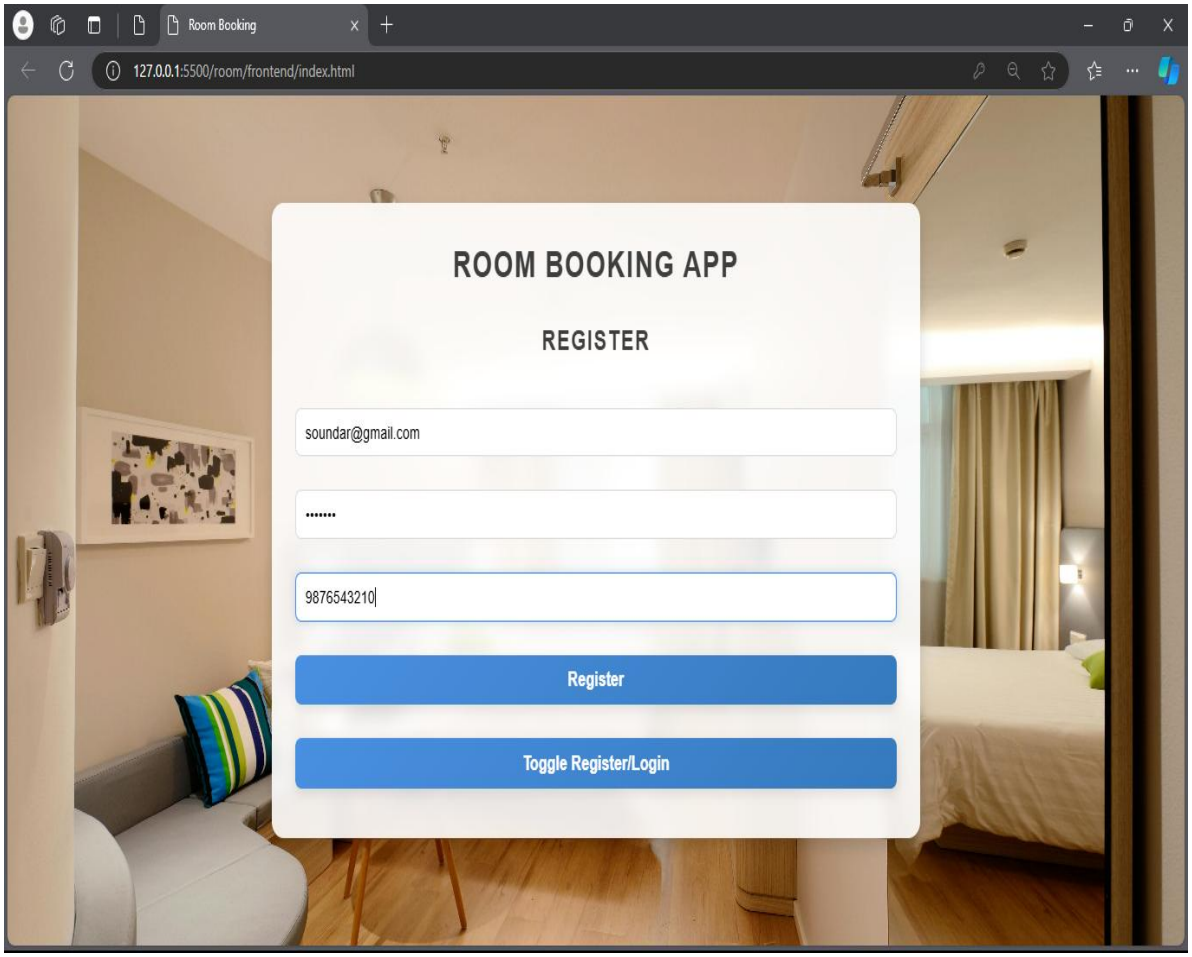


Figure 3.1 Register page

The register page (**FIGURE 3.1**) enables customers to create a new account by providing necessary information such as name, email, and password. It ensures a seamless onboarding process for users, allowing them to access hotel services easily.

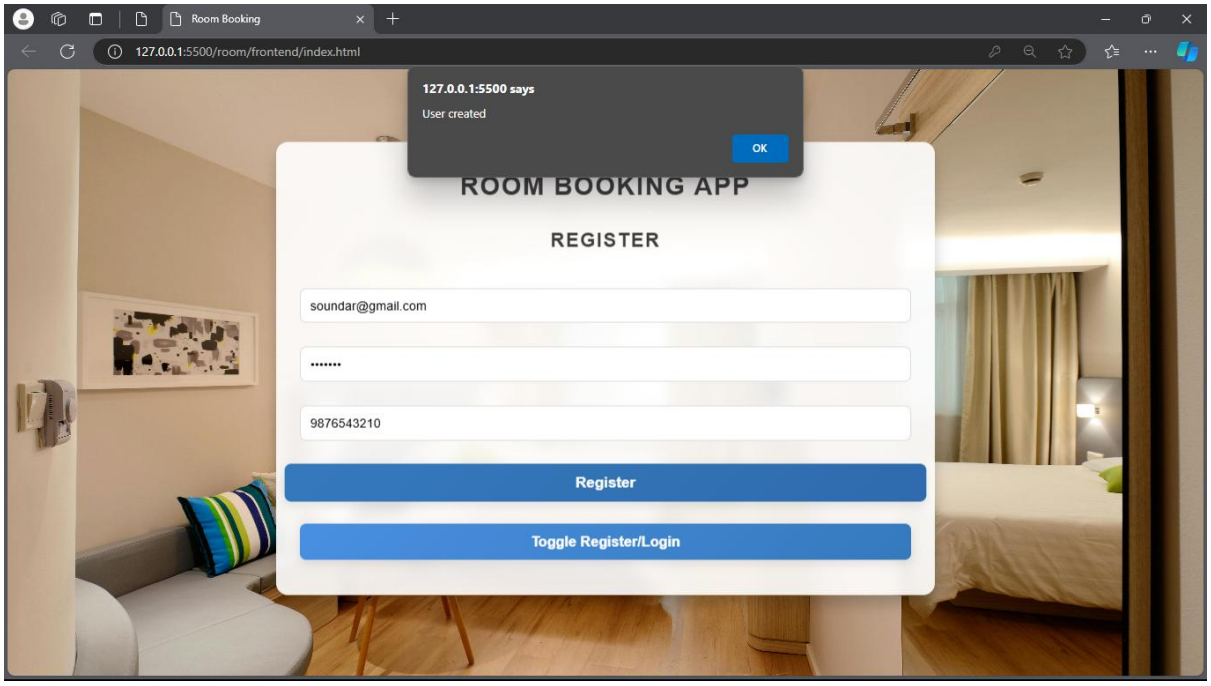


Figure 3.2 User Created

The register page (**FIGURE 3.2**) enables customers to create a new account by providing necessary information such as name, email, and password. It ensures a seamless onboarding process for users, allowing them to access hotel services easily.

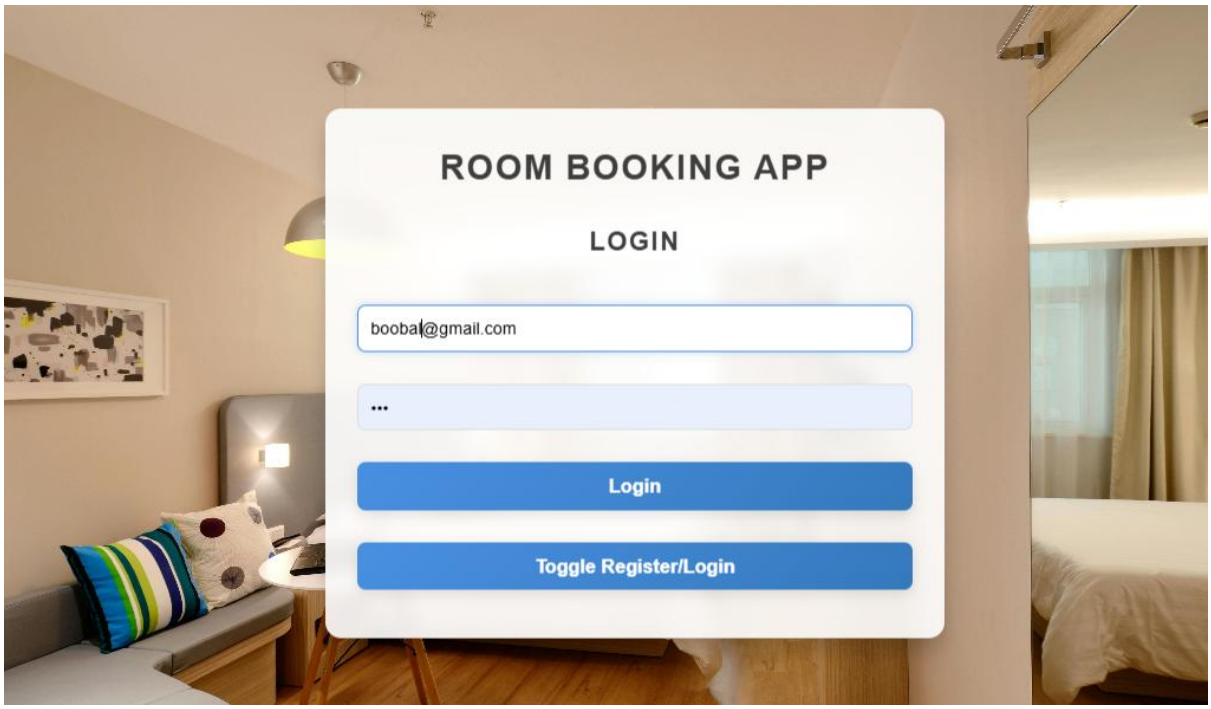


Figure 3.3 Login page

The login page (**FIGURE 3.3**) allows customers to securely access their accounts by entering their credentials. It includes fields for the username and password, along with options for password recovery and account creation.

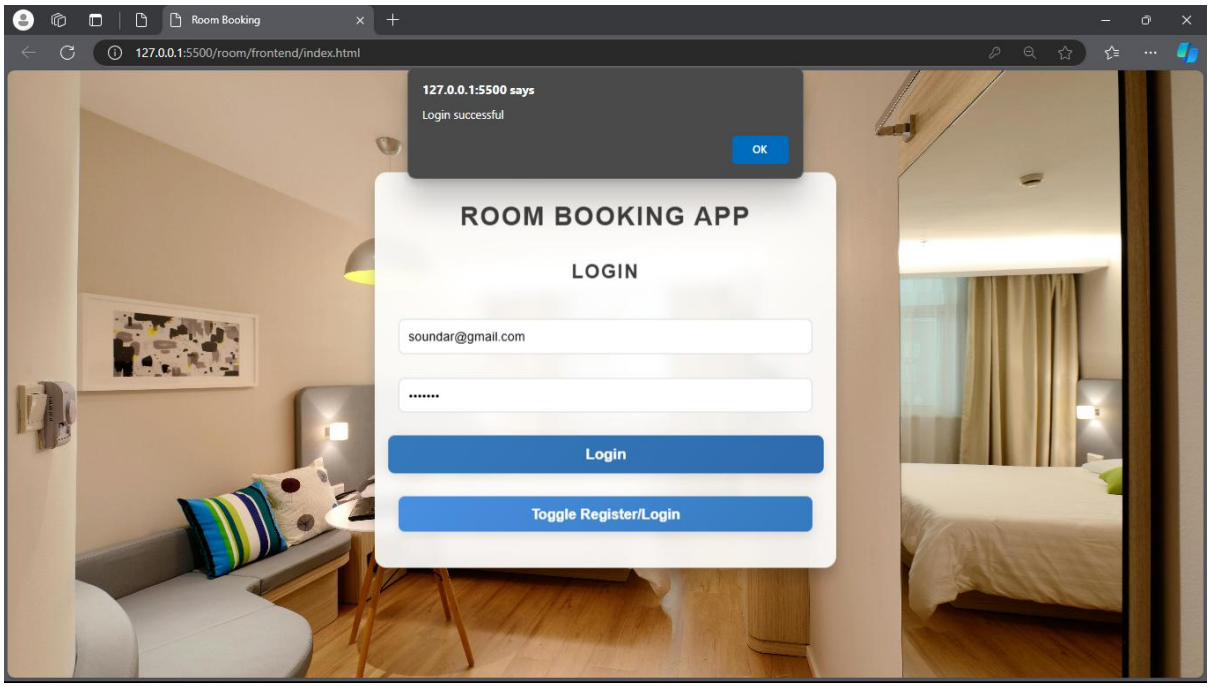


Figure 3.4 Login Successful

The login page (**FIGURE 3.4**) allows customers to securely access their accounts by entering their credentials. It includes fields for the username and password, along with options for password recovery and account creation.

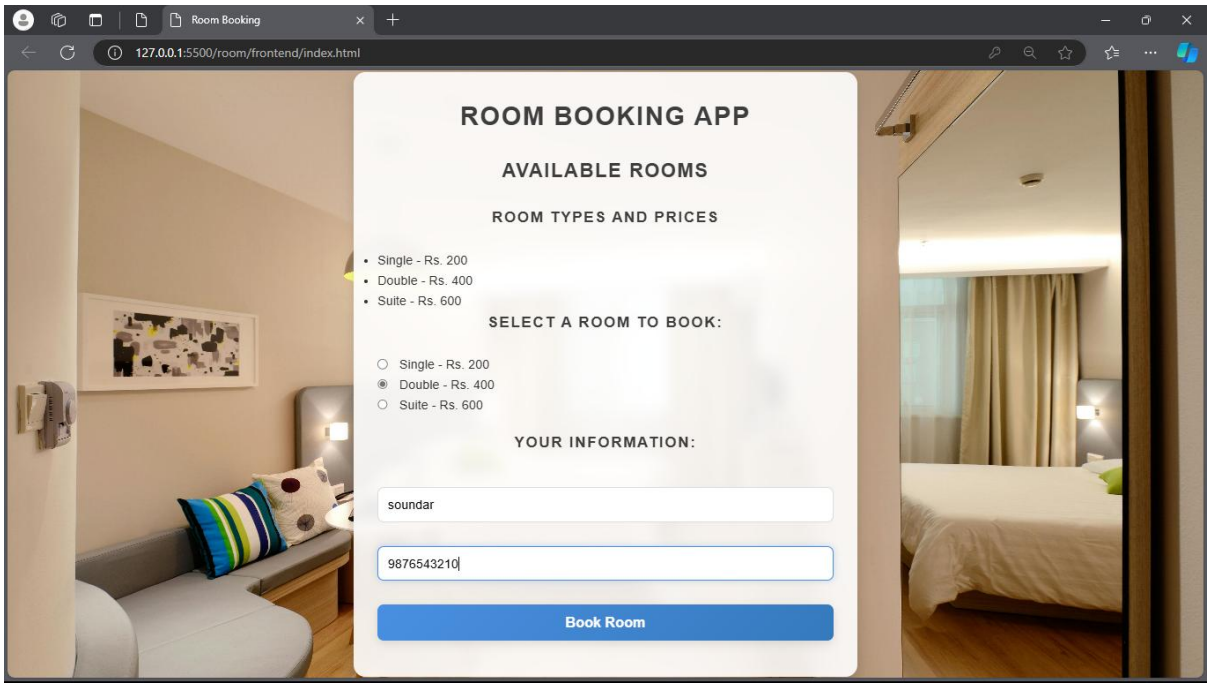


Figure 3.5 Home page

The home page (**FIGURE 3.5**) serves as the main entry point for the hotel management system, showcasing available services, room options, and special offers. It provides a user-friendly interface for customers to navigate and access information easily.

.

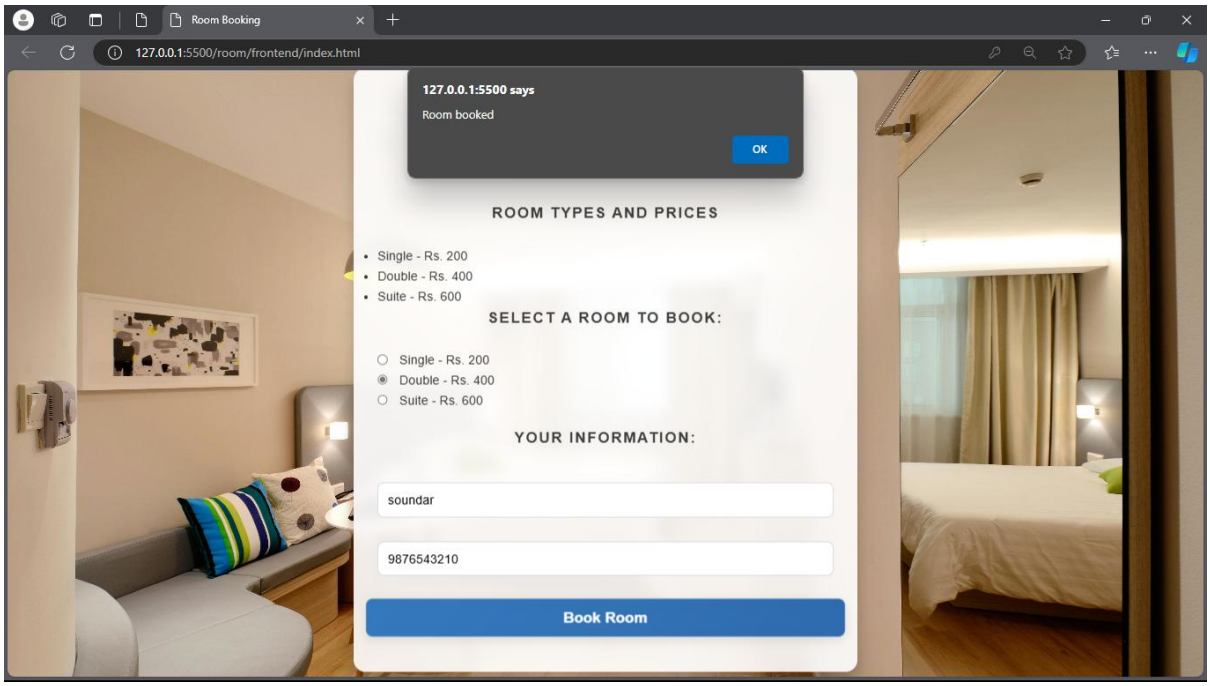
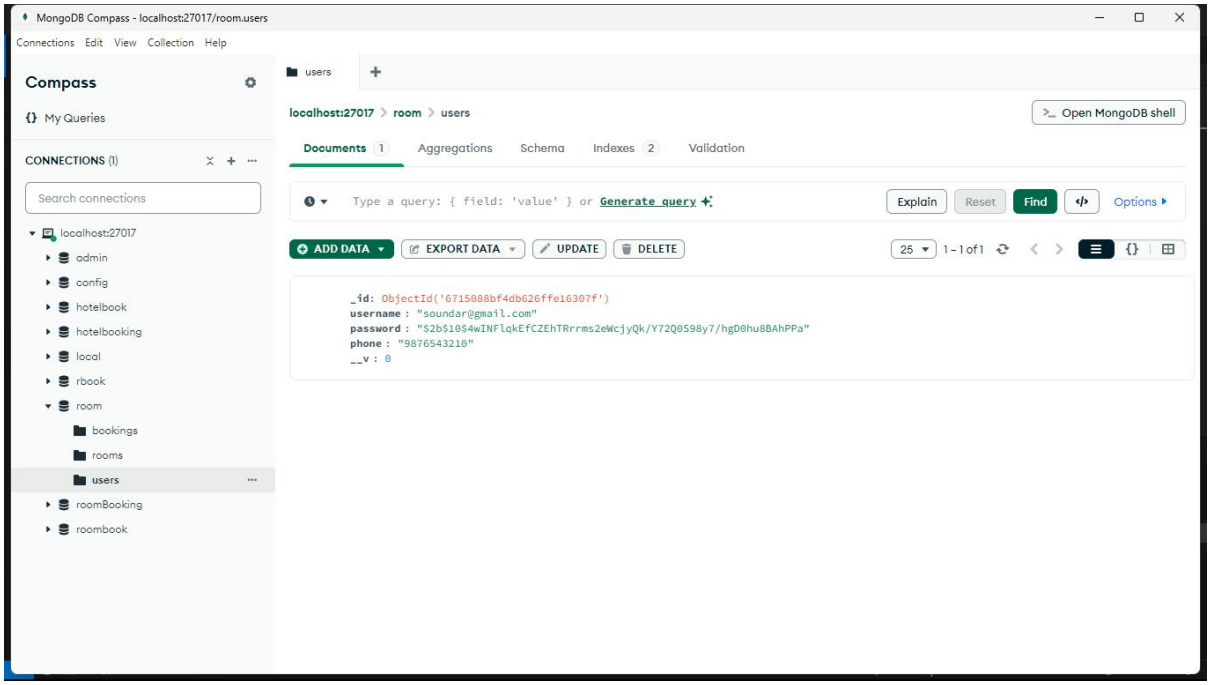


Figure 3.6 Room Booked

The room booked page (**FIGURE 3.6**) displays confirmation of the successful reservation, including details such as room type, check-in and check-out dates, and total cost. Users can also find options to modify their bookings or view additional information about their stay.



3.7. Backend connection

The backend connection page (**FIGURE 3.7**) establishes communication between the frontend and the server, enabling data retrieval and updates for room availability and bookings. It ensures secure API interactions, facilitating seamless operations such as user authentication and reservation management..

3.2 Results

Thus, we were able to create and implement a storyteller Website with the MEAN Stack structure

Frontend and Backend components:

1. **Frontend:**
    - **Components**: Register,Login,Home,Backend connection (JS files).
2. **Backend:**
    - **Controllers**: Handle backend logic (items and User controllers).
    - **Models**: Define the database schemas (items and User models in MongoDB).
    - **Routes**: Define API endpoints for itmes and users (Express.js routes).

This setup supports user authentication, items display, and essential features like login, register, and items management.

CHAPTER 4

CONCLUSION

In conclusion, the developed room booking system is well-structured with a clear separation between the frontend and backend. The frontend, built with Angular, uses components to create a dynamic and interactive user experience. The backend, managed through controllers, models, and routes with Node.js and Express.js, ensures efficient product management, order processing, and secure user authentication. This architecture, utilizing the MEAN stack, offers the scalability, flexibility, and robust foundation necessary for building a modern and full-featured room booking system.

References

- **MDN Web Docs** - Comprehensive guide for web technologies like HTML, CSS, and JavaScript.
  MDN Web Docs
- **Angular Official Documentation** - Learn about Angular components and best practices for building frontend applications.
  Angular Docs
- **MEAN Stack Guide** - Learn how to create a full-stack project using MongoDB, Express.js, Angular, and Node.js.
  MEAN Guide
- **Tailwind CSS Documentation** - For utility-first CSS framework used in the frontend.
  Tailwind CSS Docs
- **Node.js Official Docs** - Backend technology for building scalable applications.
  Node.js Documentation