

# ROOMZEE: A Cross-Platform Room Rental and Booking System Using Modern Web and Mobile Technologies

<sup>1</sup>Dr. Dinesh D. Patil, <sup>2</sup> Shruti Mangesh Bunde, <sup>3</sup>Nidhi Vinodsingh Pardeshi,  
<sup>4</sup>Manasvi Rajesh Bauskar

<sup>1</sup>HOD, <sup>2</sup>Student, <sup>3</sup>Student, <sup>4</sup>Student <sup>1</sup>Department of Computer Science and Engineering,

<sup>1</sup>HSM's Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal, India

<sup>1</sup>dineshonly@gmail.com, <sup>2</sup>shrutibunde349@gmail.com, <sup>3</sup>nidhipardeshi516@gmail.com, <sup>4</sup>bauskarmanasvi@gmail.com

**Abstract-** The increasing demand for rental accommodation in urban areas has made traditional room searching methods inefficient and time-consuming. Conventional approaches rely heavily on brokers and manual processes, often resulting in higher costs, lack of transparency, and delayed communication. To address these challenges, this paper presents Roomzee, a cross-platform room rental and booking system designed to simplify and digitize the process of finding and booking rental spaces. The proposed system integrates both web and mobile platforms, allowing users to search, view, and book rooms in real time. The frontend of the application is developed using React, providing a responsive and user-friendly interface [6]. The backend services are implemented using Supabase, which offers secure authentication, real-time database management, and efficient data handling capabilities. The use of cloud-based architecture ensures scalability, reliability, and continuous availability of the system [4]. The development process follows Agile methodology to support iterative improvements and adaptability throughout the software development lifecycle [14]. The system reduces dependency on intermediaries and improves overall efficiency in rental management. The results demonstrate enhanced user experience, faster booking operations, and improved data accessibility compared to traditional methods. Furthermore, the system is scalable and can be extended with advanced features such as online payment integration and intelligent recommendation systems.

**Index Terms—**Room Rental System, Cross-Platform Application, React, Firebase, Cloud Computing, Booking System, Agile Methodology, Web and Mobile Application, Real-Time Database, Authentication System.

## I. INTRODUCTION

In recent years, the demand for rental accommodation has increased rapidly due to urbanization, migration for education and employment, and changing lifestyles. However, the traditional process of finding rental rooms is still largely dependent on brokers, local contacts, and manual searching. This approach is not only time-consuming but also lacks transparency, often leading to higher costs and limited choices for users. With the advancement of web and mobile technologies, digital platforms have emerged as an effective solution to simplify rental processes. These platforms provide users with the ability to search properties, compare options, and make decisions more efficiently. Cloud-based systems further enhance these platforms by enabling real-time data access, scalability, and secure data management [4]. Additionally, modern frontend technologies allow the development of responsive and user-friendly interfaces, improving overall user experience [6].

Despite these advancements, many existing systems still face challenges such as poor integration between web and mobile platforms, lack of real-time booking functionality, and dependency on intermediaries. Moreover, security and authentication remain critical concerns in online systems, which require reliable backend services [9]. To overcome these limitations, this paper introduces Roomzee, a cross-platform room rental and booking system designed to provide a seamless and efficient solution for users. The system integrates web and mobile applications into a unified platform, allowing users to search, view, and book rooms in real time. The development of the system follows established software engineering principles to ensure reliability and maintainability [1]. Furthermore, Agile methodology is adopted to enable iterative development and continuous improvement throughout the project lifecycle [14].

The main objective of this system is to simplify the rental process, reduce dependency on brokers, and provide a transparent and user-friendly platform. By leveraging modern technologies and cloud-based

architecture, Roomzee aims to deliver an efficient, scalable, and secure solution for room rental management.

## II. LITERATURE SURVEY

The development of online rental and booking systems has gained significant attention in recent years due to the increasing demand for digital solutions in the housing sector. Various researchers and platforms have attempted to address the challenges associated with traditional rental systems. Sharma and Kaur [11] proposed an online rental management system based on cloud computing technologies.

Their system improved accessibility and data management; however, it lacked real-time booking capabilities and user interaction features. Similarly, Kumar et al. [12] developed a web and mobile-based rental platform that provided better accessibility but faced limitations in terms of user interface design and system responsiveness. Singh and Verma [13] focused on the design and implementation of online booking systems with an emphasis on backend efficiency. While their system ensured proper data handling, it did not fully support cross-platform integration, which is essential in modern applications. Cloud computing concepts discussed by Erl et al. [4] highlight the importance of scalable and distributed architectures, which are crucial for handling large user data and real-time operations.

In addition, modern frontend technologies such as React enable the development of dynamic and responsive user interfaces, improving usability and performance [6]. Backend services like Firebase provide secure authentication and real-time database management, which are essential for modern web applications [9]. Agile development methodologies further enhance system development by promoting iterative improvements and flexibility [15].

Despite these advancements, existing systems still face several limitations, including lack of centralized booking, limited cross-platform support, dependency on intermediaries, and inefficient user experience. The proposed Roomzee system addresses these gaps by integrating modern technologies into a unified, scalable, and user-friendly platform.

## III. SYSTEM ARCHITECTURE

### Architecture Overview:

The system architecture of the proposed Roomzee platform is designed using a modern client-server model integrated with cloud-based backend services. The architecture ensures scalability, real-time data processing, and secure user interactions. The overall system consists of multiple layers, including the user interface layer, application layer, backend services, and database layer. The architecture is structured to support both web and mobile platforms, providing seamless accessibility to users.

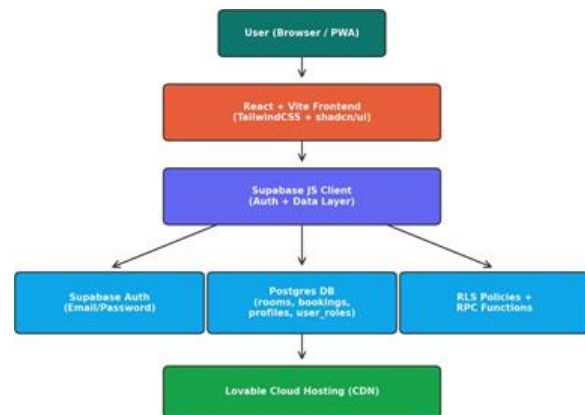


Fig 1: System Architecture

### 1. User Layer (Browser / PWA)

The system is accessed by users through a web browser or Progressive Web Application (PWA). This layer allows users to interact with the system for searching rooms, viewing details, and performing booking operations.

### 2. Frontend Layer (React + Vite)

The frontend of the system is developed using React with Vite, which provides a fast and responsive user interface. TailwindCSS and modern UI libraries are used to enhance the visual design and user experience. This layer handles user interactions and communicates with backend services [6].

### 3. Application Layer (Supabase JS Client)

The Supabase JavaScript client acts as the bridge between the frontend and backend services. It manages API calls, authentication requests, and database operations. This layer ensures efficient communication and data handling within the system.

**4. Authentication Module (Supabase Auth)**

User authentication is handled using Supabase Auth, which provides secure login and registration functionality using email and password. This ensures that only authorized users can access system features.

**5. Database Layer (PostgreSQL Database)**

The system uses a PostgreSQL database managed through Supabase to store application data such as user profiles, room listings, bookings, and user roles. The database supports structured data storage and efficient querying.

**6. Security Layer (RLS Policies & RPC Functions)**

Row Level Security (RLS) policies are implemented to restrict data access based on user roles. RPC (Remote Procedure Call) functions are used to perform secure and optimized database operations. This enhances system security and data integrity.

**7. Hosting Layer (Cloud Hosting / CDN)**

The application is deployed on cloud hosting platforms with Content Delivery Network (CDN) support. This ensures high availability, faster content delivery, and scalability of the system [4].

**IV. METHODOLOGY**

The development of the Roomzee system follows an Agile-Iterative Software Development Methodology, which emphasizes incremental development, continuous feedback, and adaptability. Instead of building the entire system in a single phase, the project was divided into multiple smaller iterations (sprints), where each sprint focused on delivering a specific functional component of the system.

Agile methodology enables flexibility in development, allowing changes to be incorporated based on testing results and user feedback. This approach improves system quality, reduces risks, and ensures timely completion of the project [14], [15].

**A. Agile Development Approach**

The system was developed through a series of structured sprints, where each sprint contributed to a specific module of the application. This iterative approach ensured that each feature was properly developed, tested, and refined before moving to the next stage.

Sprint No.	Goal / Module	Description	Duration
Sprint 1	Project Setup & UI Design	Initial project setup, design system creation, navbar and homepage (hero section) development	2 Days
Sprint 2	Authentication & Database	Database schema design, user authentication (signup/login) using Superbase	3 Days
Sprint 3	Room Listing Module	Development of room listing, search filters, and room detail page	3 Days
Sprint 4	Booking System	Implementation of booking flow with conflict detection logic	2 Days
Sprint 5	Admin Dashboard	Development of admin panel with role-based access control	2 Days
Sprint 6	UI Enhancement	UI improvements, branding (logo, favicon), responsiveness optimization	2 Days
Sprint 7	Testing & Deployment	System testing, bug fixing, deployment, and documentation	2 Days

Table 1: Sprint Planning Table

**B. Methodology Explanation**

Each sprint followed a structured cycle consisting of:

- Planning – Defining sprint goals and features
- Development – Implementing functionalities
- Testing – Verifying functionality and fixing issues
- Review – Evaluating output and preparing for next sprint

This iterative cycle ensured continuous improvement and helped in identifying issues at an early stage. The modular development approach also improved maintainability and scalability of the system.

### C. Advantages of Agile in This Project

- Faster development through incremental releases
- Flexibility to incorporate changes
- Early detection of errors
- Better collaboration and task management
- Improved overall system quality

## V. RESULTS AND DISCUSSIONS

The Roomzee system was tested across multiple modules to evaluate its performance, usability, and reliability. Different test cases were designed to verify the functionality of core features such as authentication, room listing, booking, and database operations. The testing process included unit testing, integration testing, and system testing to ensure that all components work together seamlessly. The results indicate that the system performs efficiently and meets the expected requirements.

Test Case	Description	Expected Output	Result
User Registration	New user signup	Account created successfully	Passed
User Login	Login with valid credentials	Successful authentication	Passed
Room Listing	Display available rooms	Rooms displayed correctly	Passed
Search Filter	Apply filters (price/location)	Filtered results shown	Passed
Room Booking	Book available room	Booking confirmed	Passed
Conflict Detection	Book already booked room	Error / restriction message	Passed
Database Storage	Store user & booking data	Data stored correctly	Passed
Admin Access	Role-based dashboard access	Admin panel accessible	Passed

Table 2: Testing Results Table

### A. Performance Analysis:

The system demonstrated efficient performance in handling user requests and database operations. The integration of Superbase enabled real-time data synchronization, ensuring that updates such as room availability and bookings are reflected instantly. The frontend developed using React provided a smooth and responsive user experience, reducing loading time and improving navigation efficiency [6]. The cloud-based backend architecture ensured high availability and scalability, allowing the system to handle multiple users simultaneously [4].

### B. Discussion:

From the results, it is evident that the proposed system successfully addresses the limitations of traditional rental methods. The booking system with conflict detection ensures data consistency and prevents double bookings. Additionally, the authentication system enhances security by restricting unauthorized access.

Compared to existing systems discussed in the literature survey, Roomzee provides:

- better user interface
- real-time booking capability
- improved system integration
- reduced dependency on intermediaries

The testing results confirm that the system is reliable, efficient, and user-friendly. However, certain features such as online payment integration and advanced recommendation systems can be further improved in future versions.

## VI. CONCLUSIONS:

The proposed Roomzee system successfully demonstrates how modern web and mobile technologies can be utilized to simplify the process of room rental and booking. The system effectively overcomes the limitations of traditional methods, which are often time-consuming, less transparent, and dependent on intermediaries such as brokers.

By integrating a cross-platform architecture, the system allows users to access services seamlessly through web and mobile interfaces. The use of React for frontend development ensures a responsive and user-friendly interface [6], while the implementation of Supabase as a backend service provides secure authentication, real-time database management, and efficient data handling. Additionally, the adoption of cloud-based architecture enhances system scalability, reliability, and performance [4].

The Agile development methodology used in this project enabled iterative improvements, efficient task management, and early detection of issues, contributing to the overall quality of the system [14]. The testing results confirm that the system performs effectively across all major functionalities, including user authentication, room listing, booking, and conflict detection.

Overall, Roomzee provides a reliable, scalable, and efficient solution for rental management. The system not only improves user experience but also reduces operational complexity and time consumption. In the future, the system can be enhanced by integrating features such as online payment gateways, AI-based room recommendations, and advanced filtering options to further improve functionality and user engagement.

## VII. ACKNOWLEDGEMENT

We extend our sincere gratitude to Dr. Dinesh D Patil, Head of Department and Prof. R P Chaudhari, Assistant Professor, Department of Computer science and Engineering at Shri Sant Gadge Baba College of Engineering and Technology, Bhusawal, for their expert guidance and unwavering support throughout this research. Their insightful feedback significantly strengthened our technical approach and paper organization.

## REFERENCES

1. I. Sommerville, "Software Engineering," 10th ed., Pearson Education, 2016.
2. R. S. Pressman and B. R. Maxim, "Software Engineering: A Practitioner's Approach," 8th ed., McGraw-Hill, 2015.

3. A. Dennis, B. H. Wixom, and D. Tegarden, "Systems Analysis and Design: An Object-Oriented Approach with UML," Wiley, 2018.
4. T. Erl, R. Puttini, and Z. Mahmood, "Cloud Computing: Concepts, Technology & Architecture," Prentice Hall, 2013.
5. M. Fowler, "Patterns of Enterprise Application Architecture," Addison-Wesley, 2002.
6. Meta Platforms Inc., "React – A JavaScript Library for Building User Interfaces," 2024.
7. Meta Platforms Inc., "React Native Documentation," 2024.
8. Node.js Foundation, "Node.js Documentation," 2024.
9. Google Developers, "Firebase Authentication and Firestore Documentation," 2024.
10. Google Cloud, "Cloud Firestore: NoSQL Document Database," 2024.
11. S. Sharma and P. Kaur, "Online Rental Management System Using Cloud Computing," International Journal of Computer Applications, vol. 182, no. 45, 2019.
12. R. Kumar et al., "A Smart Rental System Using Web and Mobile Technologies," International Journal of Engineering Research & Technology (IJERT), 2020.
13. A. Singh and M. Verma, "Design and Implementation of Online Booking Systems," IJRASET, 2021.
14. K. Schwaber and J. Sutherland, "The Scrum Guide," Scrum.org, 2020.
15. P. Abrahamsson et al., "Agile Software Development Methods: Review and Analysis," VTT Publications, 2002.