

# SmartRentConnect: A Web-based Rental Property Management Platform

Asawari Bhalerao<sup>1</sup>, Dhiraj Chaudhary<sup>2</sup>, Vaidehi Kulkarni<sup>3</sup>, Martand Deshmukh<sup>4</sup> and Vishnu Bansode<sup>5</sup>  
<sup>1-5</sup>Department of Information Technology, International Institute of Information Technology, Hinjewadi Phase-1, Pune, Maharashtra, India.

Email: asawarib@isquareit.edu.in, dhirajchaudhary199@gmail.com, kulkhushi04@gmail.com, martanddeshmukh20@gmail.com, bansodevishnu24@gmail.com

**Abstract**— Rental property management is often performed using manual methods or separate systems, which creates difficulties in handling property bookings, payment processing, complaint tracking, and communication between tenants and property owners. To address these issues, SmartRentConnect is proposed as a centralized web-based rental management platform that simplifies rental-related activities through an integrated system. The platform enables users to browse rental properties, book flats and paying guest (PG) accommodations, manage bookings, submit complaints, and provide feedback through ratings. The system supports a hybrid payment mechanism by integrating Razorpay for secure online transactions while also allowing cash payment options with owner approval. In addition, Cloudinary is integrated for efficient management of property images and media files. To improve accessibility and ease of use, a guidance feature is included to assist users in understanding system functionalities and navigation. SmartRentConnect is implemented using React with TypeScript for frontend development and Spring Boot with Java for backend operations. MySQL is used for database management, while JWT-based authentication ensures secure access control and protection of user data. The proposed platform improves coordination among tenants, property owners, and administrators while providing an efficient, secure, scalable, and user-friendly solution for modern rental property management [2], [5].

**Index Terms**— SmartRentConnect, Rental Property Management, Web-Based System, Booking Management, Hybrid Payment System, Razorpay, JWT Authentication, Cloudinary, React, Spring Boot.

## I. INTRODUCTION

### A. Overview

The increasing demand for rental accommodations such as flats and paying guest (PG) facilities has led to significant growth in the rental housing sector. Managing rental activities through manual processes or multiple independent systems often creates challenges in property booking, payment processing, complaint resolution, and communication among stakeholders. Existing rental platforms generally focus on basic functionalities such as property listing and booking, while providing limited support for integrated payment management, issue tracking, and centralized coordination. As a result, users may face difficulties in accessing information, monitoring transactions, and managing rental activities efficiently.

To address these challenges, SmartRentConnect is introduced as a centralized rental property management platform that brings together tenants, property owners, and administrators within a unified environment. The platform enables users to explore available properties, manage bookings, raise complaints, submit feedback, and perform payments through both online and offline modes. Developed using React with TypeScript, Spring Boot, Java, and MySQL, the system aims to provide a secure, reliable, and user-friendly solution that improves efficiency and coordination in rental property management [5], [6].

### *B. Objective*

The primary objective of SmartRentConnect is to develop a centralized web-based rental management platform that simplifies the interaction between tenants, property owners, and administrators by integrating all major rental-related functionalities into a single system. The specific objectives of the proposed system are as follows:

- To provide an organized platform where users can search, view, and book rental properties such as flats and PG accommodations based on their requirements.
- To improve rental management efficiency by reducing dependence on manual work and disconnected systems.
- To implement a hybrid payment system that supports both secure online payments using Razorpay and offline cash payments with owner approval.
- To provide complaint management and feedback mechanisms that improve transparency, communication, and service quality between users and property owners.
- To ensure data security through JWT-based authentication and role-based access for tenants, property owners, and administrators [14].
- To provide a user-friendly interface that allows smooth navigation and improves overall user experience.
- To maintain efficient storage and management of property-related images and files using Cloudinary integration.

### *C. Problem Definition*

Rental property management in many situations is still dependent on manual coordination or separate systems for property searching, booking, payment processing, and complaint management. This creates difficulties in maintaining organized records and results in poor communication between tenants and property owners [1], [4]. Tenants often experience problems in finding suitable rental accommodations with proper details regarding location, pricing, and facilities. In addition, payment management becomes difficult when flexible payment options are unavailable or poorly managed. Property owners face challenges in maintaining booking records, handling multiple users, responding to complaints, and managing property-related data efficiently. The absence of a centralized system also results in delays in complaint resolution, inefficient tracking of transactions, and poor coordination among stakeholders. Furthermore, maintaining images, records, and user-related information across multiple sources creates scalability and management problems. Therefore, there is a need for a secure, centralized, and scalable rental management system that simplifies booking, complaint handling, payment processing, and communication between all users.

### *D. Purpose*

The purpose of SmartRentConnect is to simplify rental property management through a centralized web-based platform. It enables tenants, property owners, and administrators to manage property searching, booking, payments, and complaint handling within a single system. The platform improves transparency, communication, and service quality through feedback and complaint management features. In addition, it supports both online payments through Razorpay and offline cash payments, providing flexibility and convenience for users. Overall, SmartRentConnect aims to deliver a secure, organized, and efficient rental management solution.

### *E. Scope*

The scope of SmartRentConnect is to provide a centralized web-based platform for rental property management. The system supports property browsing, booking, payment processing, complaint handling, and feedback management for tenants, property owners, and administrators. It includes hybrid payment support through Razorpay and cash payment options, secure JWT-based authentication, and Cloudinary integration for media management. The current implementation focuses on essential rental management functionalities, while advanced features such as recommendation systems, predictive analytics, and mobile applications are considered for future development [5].

## II. LITERATURE SURVEY

A review of existing rental management platforms and web-based accommodation systems shows that technology has improved the process of property searching, tenant management, and booking activities. Many systems have reduced manual effort and improved accessibility for users; however, limitations still exist in areas such as complaint management, flexible payment handling, centralized coordination, and secure data organization. Several research studies have proposed systems for managing rental activities using different technologies and approaches. However, most solutions focus only on selected functionalities instead of providing a complete rental management environment.

TABLE I. LITERATURE SURVEY TABLE

| Sr. No. | Existing System (Year)                                        | Technology / Technique              | Limitations                                         |
|---------|---------------------------------------------------------------|-------------------------------------|-----------------------------------------------------|
| 1       | Web-Based PG Management System (2023)                         | HTML, CSS, JavaScript, MySQL (CRUD) | Limited payment integration, no complaint handling  |
| 2       | Online Rental House Management System (2024)                  | PHP, MySQL, Search & Filtering      | No real-time communication, limited payment options |
| 3       | RENTEASE (2025)                                               | MERN Stack, JWT Authentication      | Incomplete payment gateway, no complaint tracking   |
| 4       | Web-Based House Rental Management System (2023)               | JSP, Servlets, MySQL                | Outdated UI, limited scalability                    |
| 5       | Rental Property Management System Using Web Technology (2024) | React, Spring Boot, REST APIs       | No hybrid payment system, no user guidance          |
| 6       | Web-Based Room Booking and Management System (2025)           | Node.js, Express, MySQL             | No feedback support, limited payment handling       |

### A. Literature Gap

From the review of existing systems, it is observed that many rental management solutions focus only on selected modules such as booking or listing while ignoring features like complaint handling, hybrid payment support, guidance assistance, and centralized management. SmartRentConnect attempts to address these limitations by combining booking, complaint management, feedback systems, secure authentication, and flexible payment options into a single platform [1]–[5].

## III. METHODOLOGY

### A. System Requirements

The successful implementation of SmartRentConnect depends on a set of hardware, software, functional, and non-functional requirements that ensure smooth operation of the platform. Since the system is developed as a web-based application, it is designed to be accessible through commonly available devices and internet connectivity without requiring specialized infrastructure.

Hardware Requirements:

- A personal computer or laptop with a minimum of 8 GB RAM is recommended for smooth system performance during development and execution.
- Sufficient storage space is required to manage application files, property images, booking information, and related records efficiently.
- A stable internet connection is necessary for accessing the platform, performing bookings, processing payments, and interacting with cloud-based services.

Software Requirements:

- Frontend: The frontend of the system is developed using React with TypeScript to provide an interactive and responsive user experience. Tailwind CSS is used to create a structured and visually organized interface, while Vite supports faster development and execution.
- Backend: The backend implementation is carried out using Spring Boot and Java, where business logic, request handling, user authentication, and communication between system modules are managed efficiently.
- Database: MySQL is used as the relational database management system for storing user information, booking records, complaint details, payment information, and property-related data. JPA/Hibernate supports efficient interaction between application components and the database.
- External Services: The platform integrates Razorpay for secure online payment transactions and Cloudinary for handling property images and media storage in an organized manner.

### Functional Requirements:

- Users should be able to register and log in securely into the system.
- Tenants should be able to browse properties and perform booking operations according to their preferences.
- Property owners should be able to add, manage, and update rental property information.
- The platform should support complaint submission and complaint tracking mechanisms.
- Users should be able to provide ratings and feedback for service improvement.
- The system should support both online payment and cash payment mechanisms.
- Administrators should be able to manage users, bookings, and platform activities effectively.

### Non-Functional Requirements:

- The platform should provide a responsive and easy-to-use interface for smooth navigation.
- Security should be maintained through authentication and role-based access mechanisms.
- The system should support multiple users without affecting performance.
- Reliability and availability should be maintained to ensure uninterrupted access to platform services.
- The platform should remain scalable and maintainable for future upgrades and improvements.

### B. Analysis Model

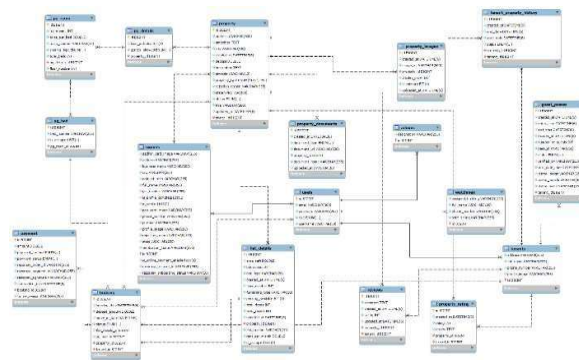


Figure 1: ER Diagram

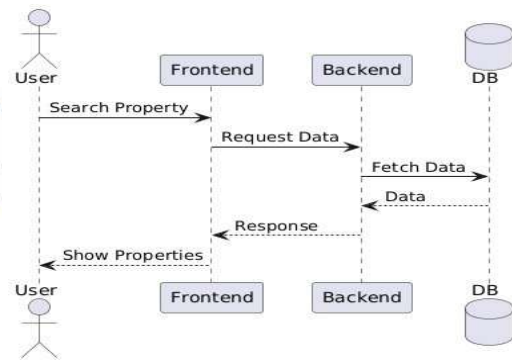


Figure 2: Sequence Diagram

### C. System Design System Architecture:

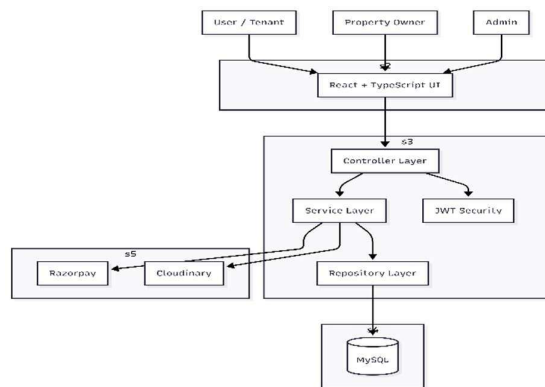
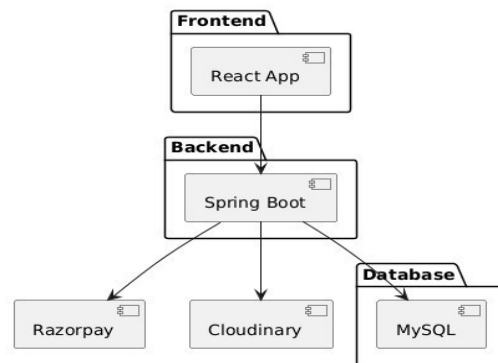


Figure 3: Architecture Figure



4: Layered Architecture Sequence Diagram

### D. Methodology

SmartRentConnect follows a structured full-stack development approach to simplify rental property management through a centralized web-based platform. The system integrates property browsing, booking, complaint handling, payment management, and communication within a single environment. Requirement analysis was performed to identify challenges such as manual coordination, inefficient booking management, limited payment flexibility, and poor complaint tracking.

The frontend is developed using React with TypeScript, while the backend is implemented using Spring Boot and Java. MySQL is used for storing user, property, booking, complaint, and payment data. JWT-based

authentication ensures secure access for tenants, property owners, and administrators. A layered architecture consisting of Controller, Service, Repository, and Entity layers improves system organization and maintainability. Razorpay is integrated for secure online transactions, while Cloudinary is used for property image management. The methodology focuses on security, scalability, and efficient coordination among all stakeholders involved in rental management.

IV. RESULTS

The developed SmartRentConnect platform successfully integrates rental property browsing, booking management, complaint handling, payment processing, and administrative monitoring into a centralized environment. The implemented modules support interaction between tenants, property owners, and administrators while improving usability and operational management. Figures 4.1–4.13 demonstrate different modules of the proposed system and validate the successful implementation of SmartRentConnect.

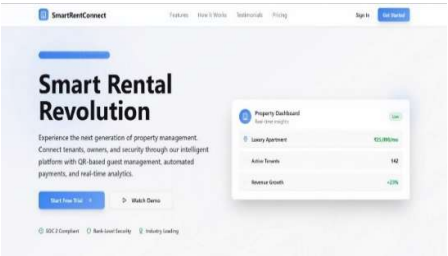


Figure 5: Home Page

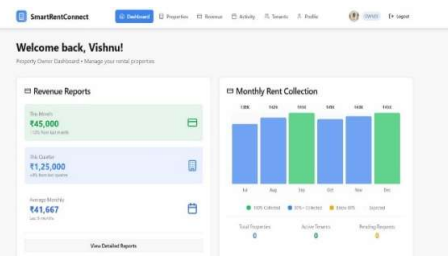


Figure 6: Owner Dashboard

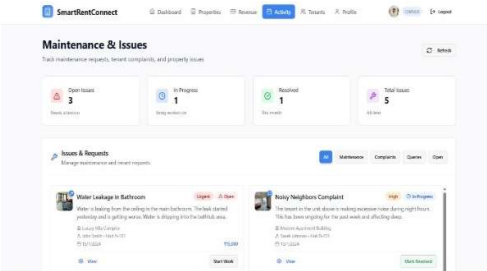


Figure 7: Maintenance & Issues Dashboard

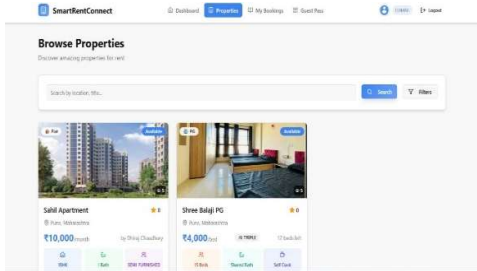


Figure 8: Property Listing Page

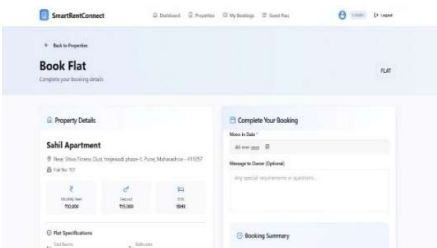


Figure 9: Flat Booking Form Page

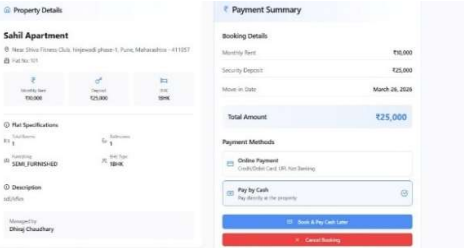


Figure 10: Payment Method Selection Page

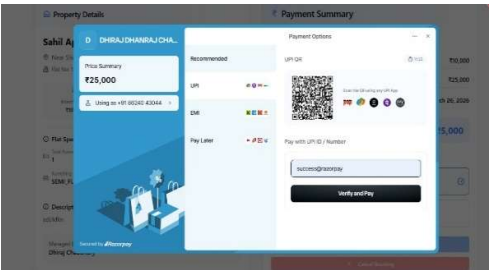


Figure 11: Payment Options Page

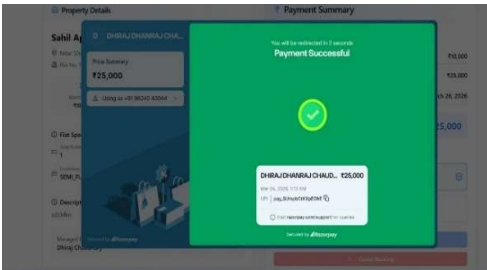


Figure 12: Payment Success Screen

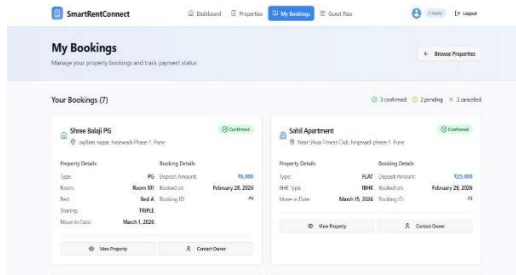


Figure 13: Tenant Booking History Page

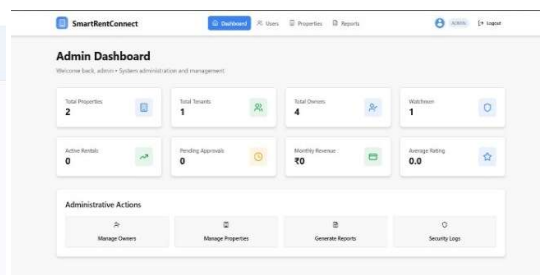


Figure 14: Admin Dashboard Page

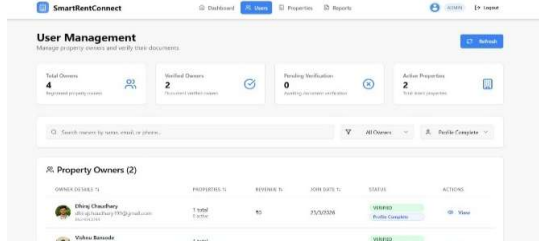


Figure 15: User Management Dashboard

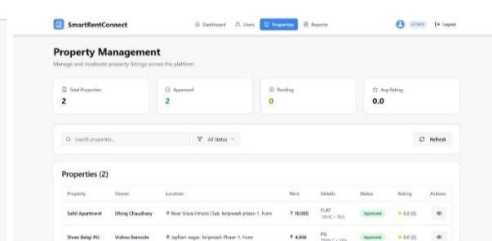


Figure 16: Property Management Dashboard

## V. DISCUSSION

The proposed SmartRentConnect platform successfully addresses major challenges in rental property management by integrating property listing, booking, complaint management, feedback collection, and hybrid payment mechanisms into a single platform. The system improves coordination among tenants, property owners, and administrators while ensuring secure access through JWT authentication.

## VI. FUTURE SCOPE

Although SmartRentConnect provides an efficient solution for rental property management, several enhancements can further improve its functionality and user experience. Future developments may include an intelligent recommendation system that suggests suitable properties based on user preferences, location, budget, and booking history. The existing guidance feature can be extended into an AI-powered chatbot to assist users with bookings, complaints, payments, and platform navigation. Additional features such as predictive analytics, SMS/email notifications, and a dedicated mobile application can further enhance accessibility, usability, and overall system efficiency. These improvements can make SmartRentConnect more intelligent, scalable, and user-friendly for modern rental management needs [2], [5].

## VII. CONCLUSION

SmartRentConnect provides a centralized and efficient solution for rental property management by integrating property browsing, booking, complaint handling, payment processing, and feedback management into a single platform. The system improves coordination among tenants, property owners, and administrators while reducing dependence on manual processes. The use of React, Spring Boot, MySQL, JWT authentication, Razorpay, and Cloudinary ensures secure, reliable, and scalable system performance. By supporting both online and offline payment methods, the platform enhances flexibility and user convenience. Overall, SmartRentConnect offers a practical and user-friendly approach to modern rental management while providing opportunities for future enhancements and scalability [2], [6].

## ACKNOWLEDGMENT

The authors express their sincere gratitude to Prof. Asawari Bhalerao for her valuable guidance, support, and encouragement throughout the development of SmartRentConnect. The authors also thank the Department of Information Technology, International Institute of Information Technology, Pune, for providing the necessary facilities and support.

## REFERENCES

- [1] K.Buradkar et al., "Property Rental Management System," *International Journal of Computer Science and Mobile Computing*, vol. 11, no. 11, pp. 177–179, 2022.
- [2] R. P. S. Manikandan et al., "House Rental Management System: Streamlining Property Management and Tenant Services," *International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)*, vol. 12, no. 5, pp. 6317–6318, 2023.
- [3] M. Wadgaonkar et al., "Rental Management System Using Machine Learning," *International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)*, vol. 8, no. 5, pp. 8363–8368, 2025.
- [4] "Enhancing House Rental Management System through Modern Web Technologies," *International Research Journal of Engineering and Technology (IRJET)*, vol. 12, no. 4, 2025.
- [5] "Property Management System in Residential Applications," *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, vol. 13, no. 4, 2024.
- [6] *Software Engineering: A Practitioner's Approach*, 8th ed. New York, NY, USA: McGraw-Hill, 2015.
- [7] *Patterns of Enterprise Application Architecture*. Boston, MA, USA: Addison-Wesley, 2002.
- [8] S. Gupta et al., "Web-Based Rental Management System for Indian Users," *International Journal of Engineering Research*, 2023.
- [9] P. Mishra and S. K. Dubey, "Design and Development of a Web-Based House Rental Management System," *International Journal of Engineering Research & Technology (IJERT)*, vol. 12, no. 7, pp. 1120–1125, 2023.
- [10] A. K. Verma and N. Singh, "Online Property Rental Management Using Full Stack Web Technologies," *International Journal of Advanced Research in Computer Science (IJARCS)*, vol. 14, no. 2, pp. 45–50, 2023.
- [11] K. Tiwari and R. Saxena, "Secure Web-Based Rental System," *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol. 15, no. 1, pp. 75–80, 2024.
- [12] M. S. Khan and A. Khan, "Rental Property Management System with Online Booking and Payment Integration," *International Journal of Engineering Applied Sciences and Technology (IJEAST)*, vol. 9, no. 2, pp. 300–305, 2024.