# A Web-Based Smart Rental Management System Using React and Secure Payment Integration

Umamaheswararao Mogili[1*], I. Sharmila[2], S. Manga[3], G. Padma[4], M. Nandini[5], P. Dileep[6]

[1*]*Assistant Professor, Department of Computer Science and Engineering (AI&ML), Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

[2,3,4,5,6] *B.Tech, Department of Computer Science and Engineering, Avanthi's St Theressa Institute of Engineering and Technology, Garividi, Andhra Pradesh, India*

**Abstract: SmartRent is a comprehensive web-based house rental and owner–tenant management system designed to modernize and simplify the traditional rental process. In many regions, house rental activities are still handled through brokers, manual records, and offline communication. This approach often results in high commission charges, lack of transparency, fake or outdated listings, and inefficient communication between owners and tenants. With the rapid growth of digital technologies and internet usage, there is a strong need for a centralized and reliable rental management platform. Smart Rent is developed to address these challenges by providing a secure, user-friendly, and efficient digital solution. The primary objective of the Smart Rent system is to create a single platform where property owners and tenants can interact directly without the involvement of brokers. The system enables owners to list their rental properties by providing complete and accurate details such as house location, rent amount, description, availability, and images. At the same time, tenants can easily search for houses based on their preferences, view detailed property information, and shortlist suitable options. By bringing both parties onto one platform, Smart Rent reduces time, cost, and dependency on intermediaries. A key feature of Smart Rent is its secure user authentication and role-based access control. Users are required to register and log in to the system, ensuring that only authorized individuals can access the platform. Based on their role, users are provided with dedicated dashboards. Owners have access to features such as adding new house listings, updating existing listings, viewing booking requests, and communicating with tenants. Tenants, on the other hand, can search and filter houses, add properties to a Wish list, send booking requests, and chat with owners. This role-based design improves usabilityand ensures that userscan access only the features relevant to them. The SmartRent platform is a cutting-edge online solution designed to revolutionize the rental experience for house and flat owners, as well as potential clients. By providing a user-friendly interface and streamlining the rental process, SmartRent aims to reduce vacancies, increase efficiency, and improve customer satisfaction.**

**Keywords: Smart Rent, React.js, Node.js, MongoDB, Razorpay, Online Rental System, Secure Payment.**

## I. INTRODUCTION

Finding a rental house is a common but challenging task in many cities and towns. In most cases, tenants depend on brokers or local agents to find suitable houses. This process is often time-consuming and costly due to high commission charges. Moreover, tenants may face issues such as fake listings, incorrect house details, and delayed communication. With the growth of the internet and web technologies, there is a need for a digital solution that simplifies the rental process. Smart rent is designed to provide an online platform that connects house owners and tenants directly. By using smart rent, owners can easily manage their properties, and tenants can quickly find houses that match their requirements. The system ensures better communication, transparency, and data security. Communication between owners and tenants is a critical aspect of house rentals. In traditional systems, communication usually happens through phone calls or messaging applications, which are unorganized and difficult to track. Smart rent overcomes this issue by integrating a real-time chat system within the platform. Using real-time communication technology, tenants and owners can exchange messages instantly. All chat conversations are stored securely in the database, allowing users to refer back to previous discussions whenever required. This feature improves transparency, trust, and efficiency in decision-making. From a technical perspective, smart rent is developed using modern web technologies to ensure scalability, performance, and security. The frontend of the application is built using react, which provides a responsive and interactive user interface. The backend is developed using node.js and express, which handle application logic, authentication, and api requests efficiently. Mongodb is used as the database to store user details, house information, booking records, and chat messages. Real-time chat functionality is implemented using (link unavailable), enabling seamless and instant communication between users.

The smart rent system provides various features such as user registration, property listing, property search, booking management, and payment processing.

The system also maintains records of transactions and bookings, which help both property owners and tenants manage their rental activities efficiently. This project is developed using modern web technologies such as the mongodb database, express. Js, React, And Node.Js, which together form the MERN stack. These technologies help create a responsive, scalable, and user-friendly web application. In today's digital world, technology plays an important role in simplifying everyday activities. The traditional method of renting houses or rooms often involves manual processes, paperwork, and time-consuming communication between owners and tenants. These traditional systems may lead to inefficiency, lack of transparency, and difficulty in managing rental records. To overcome these issues, an online rental management system is required the Smart Rent: a secure web-based rental management system is designed to simplify and automate the process of property rental. This system allows property owners to list their rental properties online and enables users to search, view, and book available properties easily through a web platform. It also provides a secure payment system using online payment gateways such as razor pay to ensure safe and reliable transactions.

## II. LITERATURE REVIEW

A literature review provides an understanding of existing systems, technologies, and research related to house rental management platforms. It helps identify the strengths and weaknesses of current approaches and highlights the need for an improved system. This section discusses traditional rental systems, existing online rental platforms, related studies, limitations, and the research gap that led to the development of the Smart Rent system. The rental market has undergone significant transformation in recent years due to rapid technological advancements and changing consumer behavior. Traditional rental systems are largely manual and broker-driven, involving lengthy procedures, high commission fees, and limited access to accurate property information. These issues often result in inefficiencies, lack of transparency, and dissatisfaction among both property owners and tenants. As a result, the demand for digital solutions that simplify and streamline rental processes has increased significantly. This literature review examines the evolution of online rental platforms, their benefits and challenges, and the role of property management software in modern rental systems.

The rental market has undergone significant transformation in recent years due to rapid technological advancements and changing consumer behavior. Traditional rental systems are largely manual and broker-driven, involving lengthy procedures, high commission fees, and limited access to accurate property information. Verma and K. Patel developed an online rent payment system integrated with digital payment gateways for secure transactions. S. Reddy and P. Naidu implemented a cloud-based rental management system to improve scalability and data storage efficiency. [1] L. Wang and J. Chen proposed a web platform that enables landlords to manage multiple properties and track tenant details in real time. N. Singh and R. Kaur developed a rental web application using modern JavaScript frameworks for better user interface and faster performance. [2] K. Brown and J. Taylor introduced a system that improves communication between landlords and tenants through messaging and notification features. V. Prakash and S. Reddy designed a MERN stack–based rental system that supports secure login, property listing, and online rent payment. [3] P. Sharma and A. Mehta developed an online property listing system that allows users to search rental houses based on location, price, and facilities. R. Gupta and

S. Jain proposed a digital rental management platform that stores tenant and property information using a centralized database. [4] T. Anderson and M. Clark designed a secure web application for property management with login authentication and role-based access control. D. Lee and H. Kim introduced a smart rental system with automated notifications and reminders for rent payments.
[5] Rajesh Kumar and S. Sharma developed an online rental management system that allows tenants to view property details and make rent payments digitally. A. Patel and R. Singh proposed a web-based house rental platform that simplifies communication between landlords and tenants using a centralized database. [6] Johnson and L. Smith designed a property management system with secure user authentication and role-based access for owners and tenants. K. Gupta and P. Verma introduced a cloud-based rental. [7] Ian Sommerville He explained the principles of software engineering and system development. His work helped in designing structured web applications and developing efficient software systems for online services.[8] Martin Fowler He contributed to the development of modern web application architectures and explained the importance of scalable backend systems using technologies such as Node.js and micro services. [9] Brendan Eich He developed JavaScript, which became the foundation for modern frontend frameworks like React.js, widely used for building interactive web applications. [10] Ryan Dahl He created Node.js, which allows developers to build scalable and high-performance backend servers. This technology is widely used in modern web applications like online rental platforms. Some of the sample artificial intelligence, machine learning and deep learning models for prediction for fire detection are described in details [11-18]. [19] Dwight Merriman He was one of the creators of MongoDB, a NoSQL database designed for handling large amounts of unstructured data efficiently, which is useful for storing user data, property details, and booking information.[20] Harshil Mathur and Shashank Kumar They co-founded Razorpay and contributed to the development of secure digital payment systems that allow safe and efficient online transactions in web applications.[21] Jeffrey Dean He contributed to large-scale distributed computing systems, improving the performance of web applications. [22] Gavin King He developed frameworks and tools for database management and enterprise applications. [23] Erich Gamma He contributed to design patterns in software engineering, which help developers build structured and reusable software systems.

## III. METHODOLOGY

Methodology describes the systematic approach followed to design, develop, and implement the Smart Rent – House Rental & Owner–Tenant Management System. A well-defined methodology helps in understanding how the project was planned, executed, tested, and evaluated. The development of Smart Rent was carried out using a modular and incremental approach inspired by the Agile development methodology. This approach allows the project to be developed in small stages, making it easier to manage complexity, identify issues early, and incorporate improvements based on testing.

### 3.1. Database Design

The database is designed to store and manage application data efficiently. Main collections include: Users (Landlord / Tenant details), Properties (House information, rent, location), Rental agreements, Payments and transaction records.

### 3.2. System Implementation

In this phase, the actual coding and development of the system are carried out.
Frontend Development: React.js is used to create responsive pages such as login, property listing, dashboard, and payment pages.
Backend Development: Node.js and Express.js are used to create REST APIs for data communication.
Payment Integration: Razorpay payment gateway is integrated to allow secure online rent payments.
MongoDB is used because it supports flexible and scalable data storage.

Based on this study, the functional requirements of the Smart Rent system were defined. These requirements include user registration and login, role-based access for owners and tenants, house listing management, house search and filtering, booking request handling, and real-time chat functionality. Non-functional requirements such as security, performance, usability, and scalability were also identified during this phase.

### 3.3. Testing

Different testing methods were used, including unit testing, integration testing, and functional testing. Test cases were written for login, booking requests, and chat functionality to ensure reliability. Users can easily access property information and complete bookings through the web interface. The results demonstrate that the system effectively reduces manual work and simplifies the rental management process. Users can easily access property information and complete bookings through the web interface. The integration of the Razorpay payment gateway enabled secure online rent payments. The payment process was successfully tested and transactions were recorded in the database. API testing was performed using Postman to verify data transfer between the frontend and backend.

### 3.4. Implementation Details

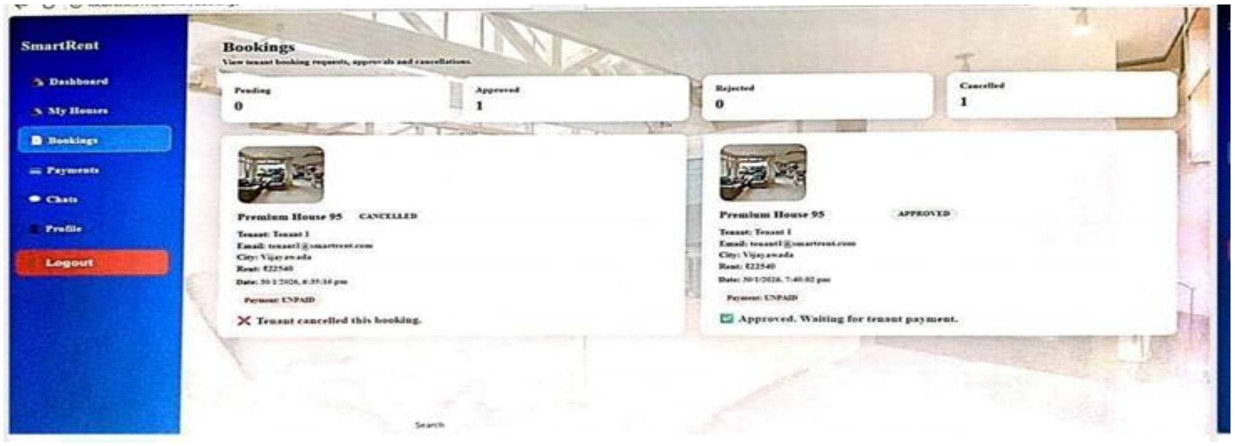
The application implements secure authentication, role-based access, and CRUD operations for house listings. Booking requests are processed and stored with status updates. Chat messages are exchanged in real time and saved in the database for future reference

### 3.5. Deployment

After successful testing, the system is deployed on a servers users can access it online. The deployment ensures the system is stable, secure, and available for real-time use. Secure Data Storage and Sharing in Multi-Cloud Environment In the cloud storage is also described to store the predicted data in a secured way [24-28].

## IV. RESULTS & DISCUSSION

The Smart Rent system was implemented successfully. Users were able to register, login, list houses, search properties, send booking requests, and communicate through chat. The system performed efficiently and met all project objectives. The results show that Smart Rent effectively addresses the problems of the traditional rental system. By eliminating brokers and providing direct communication, the system improves transparency and reduces costs. Centralized data storage improves management and reliability. The Smart Rent system was successfully developed and implemented as a web-based application. The system allows users to register, search rental properties, book available houses, and make secure online payments. The platform provides a user-friendly interface that helps both tenants and property owners manage rental activities efficiently. During testing, the application performed all major functions successfully, including user authentication, property listing, booking management, and payment processing. The integration of the payment gate way using Razorpay enabled secure and reliable online transactions. The system also records booking details and payment transaction logs in the database. The results demonstrate that the system effectively reduces manual work and simplifies the rental management process. Users can easily access property information and complete bookings through the web interface. The results demonstrate that the system effectively reduces manual work and simplifies the rental management process. Users can easily access property information and complete bookings through the web interface the integration of the Razorpay payment gateway enabled secure online rent payments. The payment process was successfully tested and transactions were recorded in the database. API testing was performed using Postman to verify data transfer between the frontend and backend.



Bookings

View tenant booking requests, approvals and cancellations.

| Pending | Approved | Rejected | Cancelled |
|---|---|---|---|
| 0 | 1 | 0 | 1 |

Premium House 95 CANCELLED

Tenant: Tenant 1
Email: tenant1@smartrent.com
City: Vijayawada
Rent: ₹22540
Date: 30/1/2026, 6:35:16 pm

Payment: UNPAID

Tenant cancelled this booking.

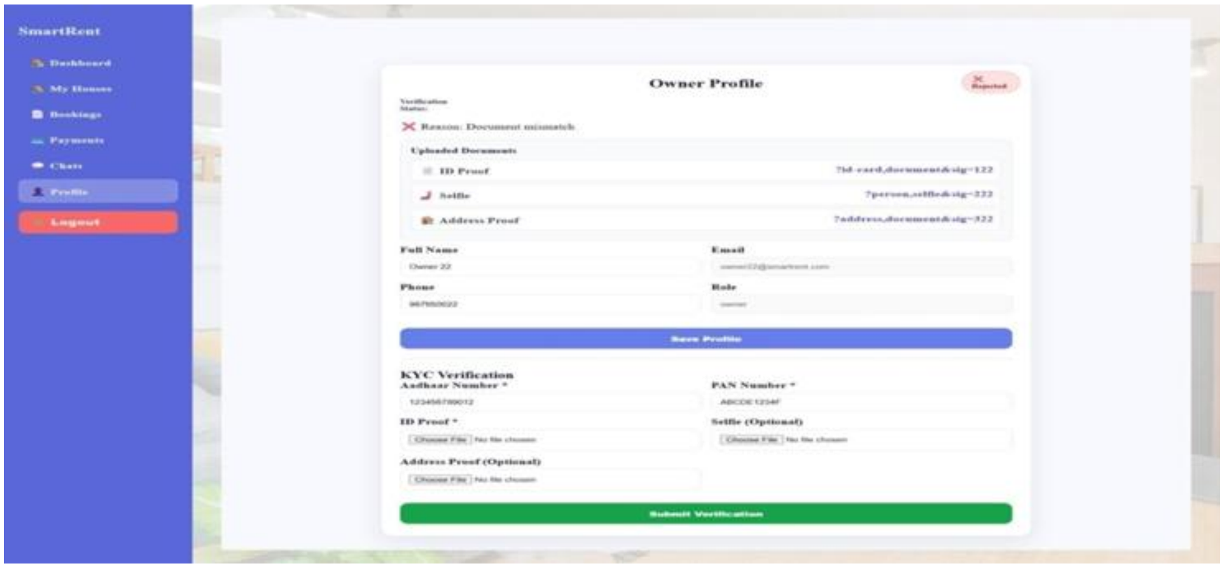
Premium House 95 APPROVED

Tenant: Tenant 1
Email: tenant1@smartrent.com
City: Vijayawada
Rent: ₹22540
Date: 30/1/2026, 7:40:02 pm

Payment: UNPAID

Approved. Waiting for tenant payment.

Fig.1. Bookings of the Rental houses



Owner Profile

Verification Status:

Reason: Document mismatch

Uploaded Documents

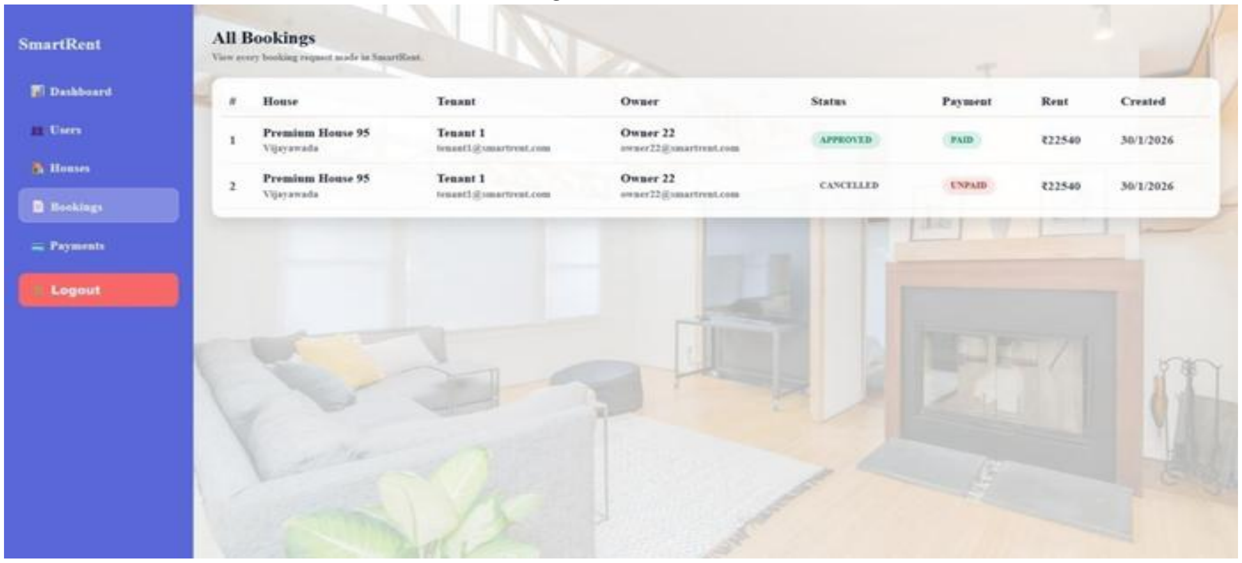
| | |
|---|---|
| ID Proof | ?id-card,document&sig=122 |
| Selfie | ?person,selfie&sig=222 |
| Address Proof | ?address,document&sig=322 |

Full Name: Owner 22
Email: owner22@smartrent.com
Phone: 9876500022
Role: owner

KYC Verification

Aadhaar Number *: 123456789012
PAN Number *: ABCDE1234F
ID Proof *: No file chosen
Selfie (Optional): No file chosen
Address Proof (Optional): No file chosen

Fig.2 Owner Profile



All Bookings

View every booking request made in SmartRent.

| # | House | Tenant | Owner | Status | Payment | Rent | Created |
|---|---|---|---|---|---|---|---|
| 1 | Premium House 95 Vijayawada | Tenant 1 tenant1@smartrent.com | Owner 22 owner22@smartrent.com | APPROVED | PAID | ₹22540 | 30/1/2026 |
| 2 | Premium House 95 Vijayawada | Tenant 1 tenant1@smartrent.com | Owner 22 owner22@smartrent.com | CANCELLED | UNPAID | ₹22540 | 30/1/2026 |

Fig.3 All Bookings

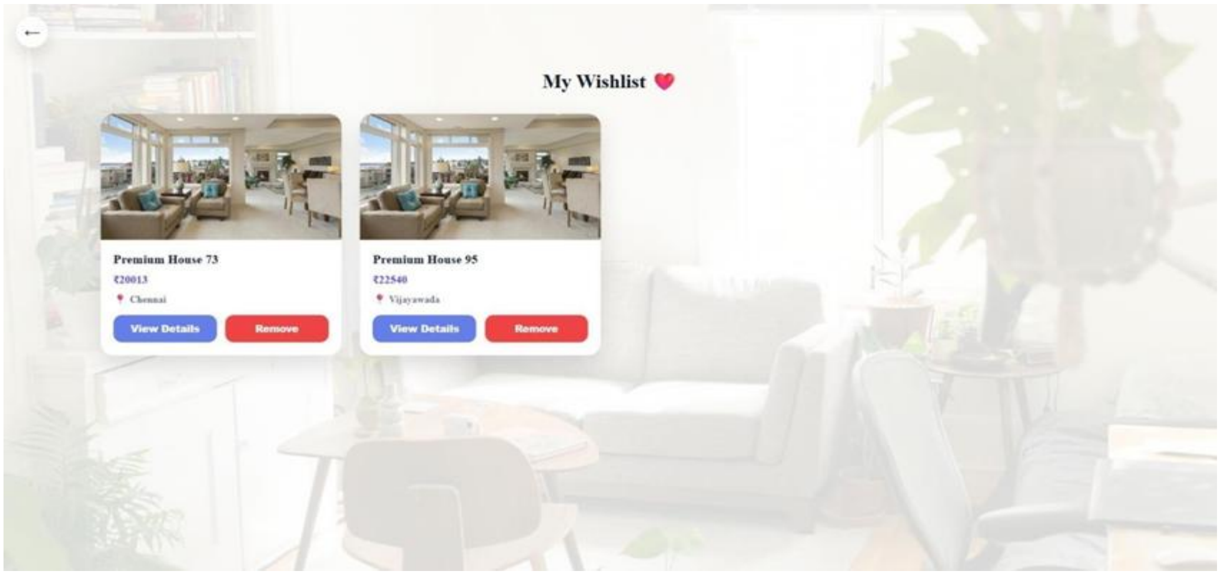


Fig.4 My Wishlist

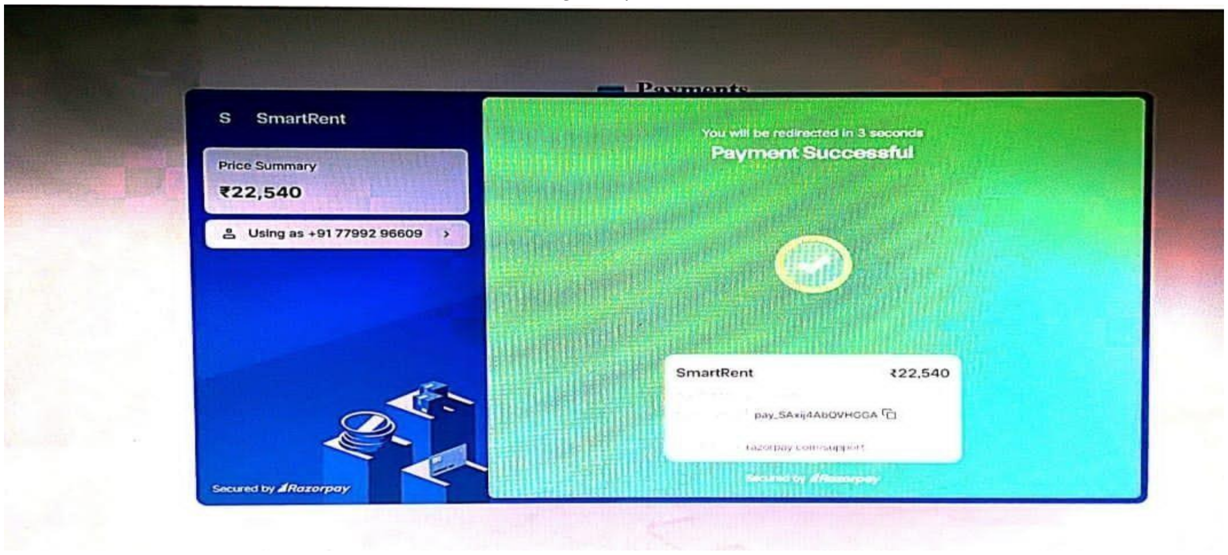


Fig.5 Payment Successful

## V. CONCLUSION

Smart Rent provides an efficient, secure, and transparent solution for house rental management. It simplifies the rental process for both owners and tenants and can be extended further to support advanced features. The project was developed using modern web technologies such as React.js for the frontend, Node.js and Express.js for the backend, and MongoDB for database management. These technologies help in creating a responsive, efficient and user-friendly application. The integration of the Razorpay payment gateway ensures secure and reliable online payment for rental transactions. This system reduces the time and effort required in the traditional rental process by providing features such as online property listing, booking requests, real-time communication, and secure payment options. It also improves transparency between property owners and tenants. In conclusion, the Smart Rental Management System provides a convenient and effective solution for managing rental properties digitally. The system can be further enhanced in the future by adding features such as mobile applications, advanced search filters, location-based services, and AI-based property recommendations.

## REFERENCES

[1] React, “React: A JavaScript library for building user interfaces,” 2024. [Online]. Available: https://react.dev/

[2] Node.js, “Node.js Documentation,” 2024. [Online]. Available: https://nodejs.org/

[3] Express.js, “Express – Fast, unopinionated, minimalist web framework for Node.js,” 2024. [Online]. Available: https://expressjs.com/

[4] MongoDB, “MongoDB Documentation,” 2024. [Online]. Available: https://www.mongodb.com/docs/

[5] Socket.IO, “Socket.IO Documentation,” 2024. [Online]. Available: https://socket.io/docs/

[6] Mongoose, “Mongoose ODM Documentation,” 2024. [Online]. Available: https://mongoosejs.com/docs/

[7] JWT, “JSON Web Tokens Introduction,” 2024. [Online]. Available: https://jwt.io/introduction/

[8] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures,” Ph.D. dissertation, Univ. of California, Irvine, 2000.

[9] I. Sommerville, Software Engineering, 10th ed. Boston, MA, USA: Pearson, 2016.

[10] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner’s Approach, 8th ed. New York, NY, USA: McGraw-Hill, 2015.

[11] Mogili, U., Ampolu, K. V., Rajasekharam, B., & Timothy, M. J. AI-Driven Interaction in AR Environments, in Journal of Digital Economy, 2024, Volume 3, Issue 1, pp. 228-234.

[12] Timothy, M. J., Rajasekharam, B., Ampolu, K. V., & Mogili, U. Threat Detection Using AI in Cybersecurity Systems, in IJIS, 2023, Volume 7, Issue 1, pp. 1-7.

[13] Ampolu, K.V., Mogili, U., Timothy, M. J., & Rajasekharam, B. Machine Learning Models for Predictive Maintenance, in IJIS, 2022, Volume 6, Issue 4, pp. 1-7.

[14] Rajasekharam, B., Timothy, M. J., Mogili, U., Ampolu, K.V., Machine Learning Models for Predictive Maintenance, in JDE, 2023, Volume 2, Issue 2, pp. 95-101.

[15] Soujania, B., Ampolu, K. V., Timothy, M. J., & Mogili, U. (2025) Classifying Disease Information Forums through Semantic Similarity-Based Machine Learning, Science, Technology and Development Journal, Volume XIV, Issue II, pp 67-75.

[16] B Satish Kumar, Kavitha C., Mogili, U.R., S. Pallam Shetty (2022). “Application of Machine Learning To Enhance the Performance of The Prophet Routing Protocol For Delay Tolerant Networks”. Journal for Basic Sciences, Volume 23, Issue 5, 2107-2116, DOI:10.37896/JBSV23.5/2278.

[17] I. Sree Geeta, Umamaheswararao Mogili. (2022), “Use of Several Machine Learning Algorithms for Effective Prediction of Cyberbullying”, International Journal of Creative Research Thoughts, Volume 10, Issue 6, pp 17.

[18] Mogili, U., & Mohamed, A. (2023, November). Artificial intelligence and machine learning in the fields of education, medical, and smart phones. In AIP conference proceedings (Vol. 2917, No. 1, p. 050012). AIP Publishing LLC.

[19] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Boston, MA, USA: Addison-Wesley, 1994.

[20] M. Fowler, Patterns of Enterprise Application Architecture. Boston, MA, USA: Addison-Wesley, 2002.

[21] R. Elmasri and S. Navathe, Fundamentals of Database Systems, 7th ed. Boston, MA, USA: Pearson, 2016.

[22] A. Silberschatz, H. F. Korth, and S. Sudarshan, Database System Concepts, 7th ed. New York, NY, USA: McGraw-Hill, 2019.

[23] S. Krug, Don’t Make Me Think: A Common-Sense Approach to Web Usability, 3rd ed. Berkeley, CA, USA: New Riders, 2014.

[24] Adithya, P. U., Mogili, U., & Mondru, J. T. (2025) A Novel Parity Authenticator-Based Zero-Knowledge Auditing Approach for Secure Cloud Data Management, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211, Volume 13, Issue 5, pp 994-999.

[25] Mogili, U., Mohamed, A., & Kasup, C. (2023, December). Mechanism of Data Sharing Using Secured Keyword Search in Cloud Computing. In Conference of Innovative Product Design and Intelligent Manufacturing System (pp. 483-494). Singapore: Springer Nature Singapore.

[26] Anjali, S., Mogili, U., & Ampolu, K. V. (2025)

Efficient Key-Based Encryption and Authentication for Advanced Digital Forensic Storage Security, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211, Volume 13, Issue 5, pp 3097-3102.

[27] Sree, S. V. D. T., Mogili, U. M. R., & Ampoly, K. V. (2025) Enhancing Security in Wearable Computing: A Lightweight Authenticated Key Exchange Scheme, International Journal of All Research Education and Scientific Methods (IJARESM), ISSN: 2455-6211, Volume 13, Issue 5, pp 3103-3108.

[28] Kanakala Pranay Raj, Umamaheswararao Mogili. (2020), "Cloud-of-Cloud: A Novel Protocol for Secure Data Storage and Sharing in Multi-Cloud Environment", Journal of Interdisciplinary Cycle Research (JICR), Volume XII, Issue VI, pp 2201-2209, DOI:18.0002JICR.2020.V12I6.008301.3171227.