

Smart Utility Management System

A. Ravi Kiran¹, B. Preethika Reddy², B. Chinna Madhu³, Ch. Rama Krishna⁴

^{1,2,3,4}*School of Engineering B. Tech, Computer Science-AIML, Malla Reddy University, Hyderabad*

Abstract: The Smart Utility Management System is a user-friendly web platform designed to make life easier for residents in gated communities. Built with the MERN stack, this system comes pre- set with all the flats in the community, so residents just need to log in with their secure credentials. Service providers like plumbers, AC repair technicians, cleaning staff, and maids can register on the platform to offer their services. Residents can quickly check who's available, book appointments, and pay for services—all in one place. The platform also prevents double bookings, so residents and service providers can manage their time more efficiently. By bringing all these utilities together in a single, easy-to-use system, the Smart Utility Management System makes everyday living more convenient and stress-free for everyone in the community.

I. INTRODUCTION

The Smart Utility Management System is a web-based platform designed to streamline utility service management in gated communities. It provides a centralized solution for residents, workers, and administrators, enabling seamless communication, service scheduling, and payment handling. Developed using the MERN stack (MongoDB, Express.js, React.js, Node.js), this system offers a dynamic and user-friendly interface, ensuring efficient management of day-to-day utilities. By addressing common challenges in gated communities, the Smart Utility Management System aims to enhance convenience, productivity, and satisfaction for all stakeholders. The platform is tailored to overcome the inefficiencies of traditional utility management systems by integrating features such as real-time interaction, dynamic scheduling, and secure payment handling. By adopting modern web technologies, the project ensures scalability and adaptability for various gated communities, regardless of size or requirements.

II. LITERATURE REVIEW

The concept of digital utility management systems has gained significant attention with the advancement of web technologies and smart city initiatives. Several researchers have proposed web-based platforms to improve service management, complaint handling, and scheduling efficiency in residential communities. M. T. Rahman and R. Akter (2018) proposed a web-based smart home utility management system that focused on automating household service requests. Their system emphasized centralized monitoring and digital record maintenance. However, it lacked real-time scheduling optimization and advanced user-role management features.

S. S. Awasthi and P. Agarwal (2020) developed a complaint management system specifically for gated communities. Their research highlighted the importance of structured grievance redressal mechanisms and administrative monitoring. While effective in complaint handling, the system did not integrate booking and payment functionalities into a unified platform.

R. Kumar, A. Sharma, and P. Singh (2019) designed an online service booking management system aimed at reducing manual scheduling conflicts. Their study introduced automated booking workflows but did not address community-based customization or role-based access control in depth.

Further research by P. Singh and N. Verma (2020) focused on digital utility platforms within smart city infrastructure. Their work emphasized scalability and digital transformation but lacked a practical implementation model tailored to gated residential communities.

A. Patel, R. Shah, and K. Mehta (2021) proposed a smart residential utility management system integrating complaint tracking and service management. However, their model did not fully

incorporate secure payment systems or real-time worker availability tracking.

Similarly, R. Sharma and A. Gupta (2021) developed a community-based utility management platform emphasizing transparency and digital coordination. While effective in coordination, the system required further enhancement in scalability and technology stack modernization.

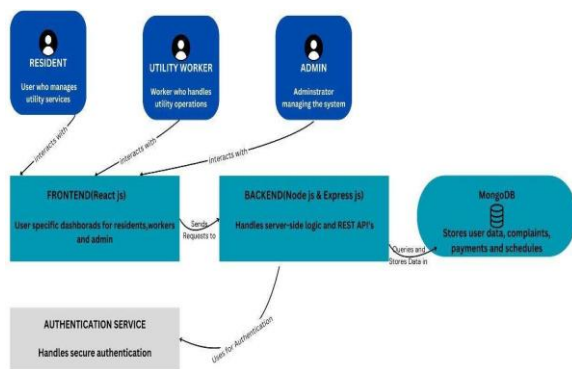
III. PROBLEM IDENTIFICATION

Residents in gated communities often face difficulties in finding and managing utility services such as plumbing, cleaning, and AC repair due to unorganized communication, scheduling conflicts, and payment issues. Service providers also struggle with handling appointments efficiently, leading to double bookings and time wastage.

There is a need for a centralized digital platform that enables residents to easily book services, make payments, and check availability while allowing service providers to manage their schedules effectively.

The Smart Utility Management System addresses this problem by providing a secure, user-friendly web-based solution that streamlines utility service management within gated communities. Not only in gated communities but also useful in society.

IV. SYSTEM ARCHITECTURE



V. METHODOLOGY

The development of the Smart Utility Management System followed a structured software engineering approach to design and implement a scalable web-based solution using the MERN stack.

1. Research and Problem Analysis

The first phase involved identifying key challenges faced by residents and service providers in gated communities:

- Unorganized communication
- Appointment scheduling conflicts
- Double bookings
- Manual payment handling
- Lack of centralized monitoring

A literature survey was conducted using previously published research on utility management systems and complaint management platforms to understand existing solutions and limitations.

2. System Design Methodology

a) Requirement Analysis

Functional and non-functional requirements were gathered:

- Functional Requirements:
 - User login and authentication
 - Service provider registration
 - Appointment booking system
 - Real-time availability checking
 - Online payment system
 - Complaint management module
 - Admin monitoring panel

Non-Functional Requirements:

- Security
- Scalability
- Usability
- Performance efficiency

b) Software Development Model

The project follows the Agile Development Methodology, enabling:

- Incremental development
- Continuous testing
- Regular feature updates
- Flexible modifications

3. System Architecture Design

The system follows a Three-Tier Architecture:

Frontend Layer – Built using React.js

- Handles user interface and user interactions.

Backend Layer – Developed using Node.js and Express.js

- Manages business logic, authentication, and API handling.

Database Layer – MongoDB

- Stores user data, bookings, complaints, and service records.

This architecture ensures modularity and scalability.

4. Database Design Method

- MongoDB collections were designed for:
- Residents
- Service Providers
- Bookings
- Complaints
- Payments

Relationships were maintained using references and Object IDs.

Data validation rules were applied to prevent inconsistencies.

5. Implementation Methodology

The project was implemented using:

- MERN Stack (MongoDB, Express.js, React.js, Node.js)
- RESTful APIs for communication between frontend and backend
- JWT-based authentication for secure login
- Role-based access control (Admin, Resident, Worker)

6. Testing Methodology

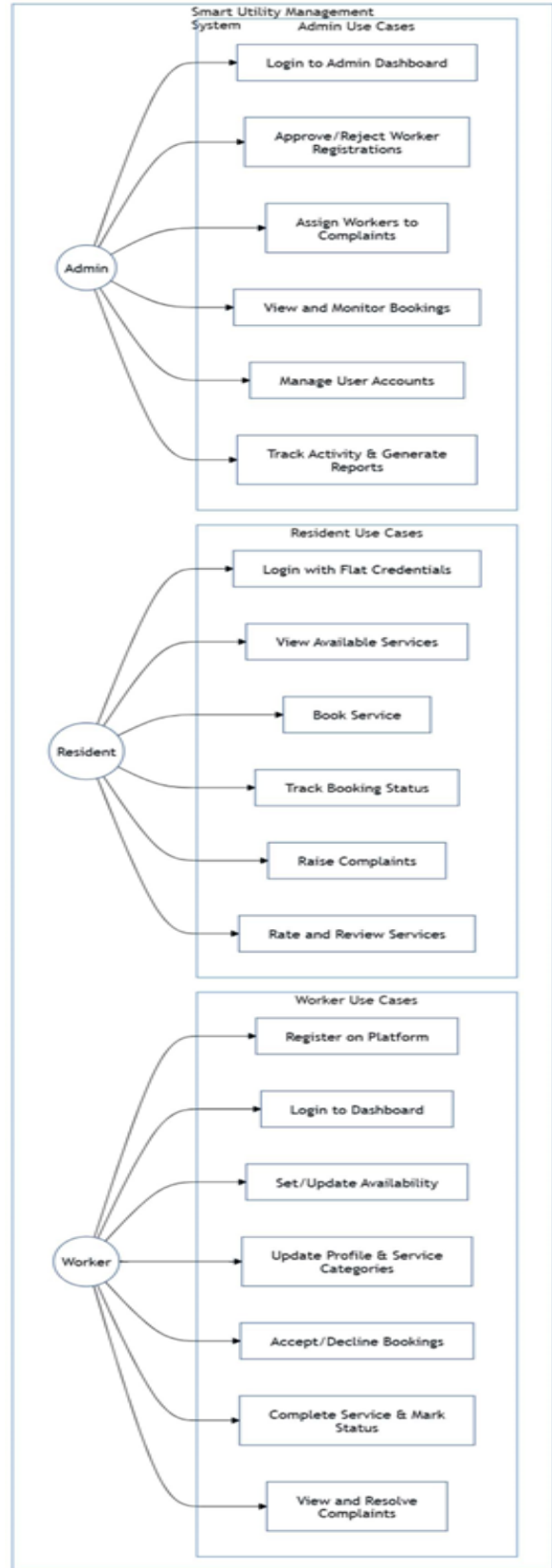
The system underwent:

- Unit Testing – Individual modules tested independently
- Integration Testing – Ensured proper communication between modules
- System Testing – Verified complete workflow

User Acceptance Testing – Ensured usability and functionality

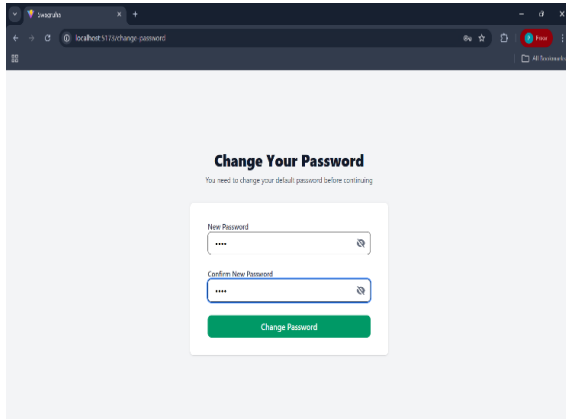
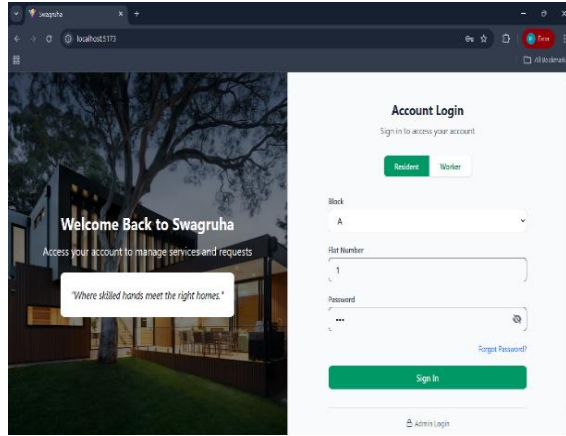
VI.SYSTEM DESIGN

1.UML DIAGRAM

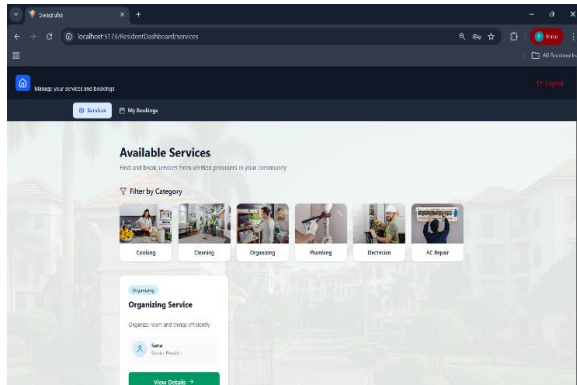


VII.RESULTS

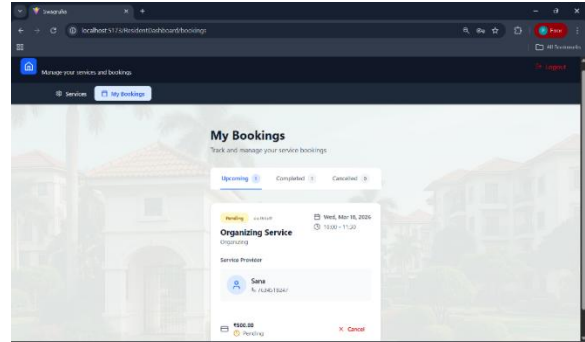
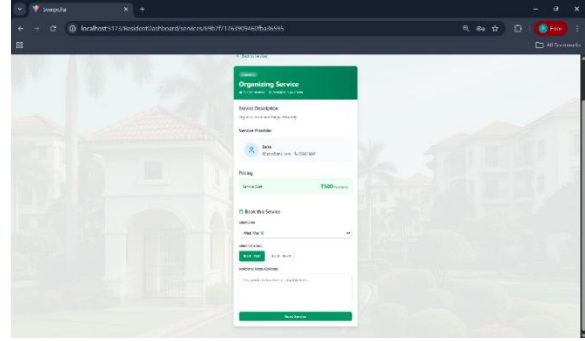
Resident Login page:



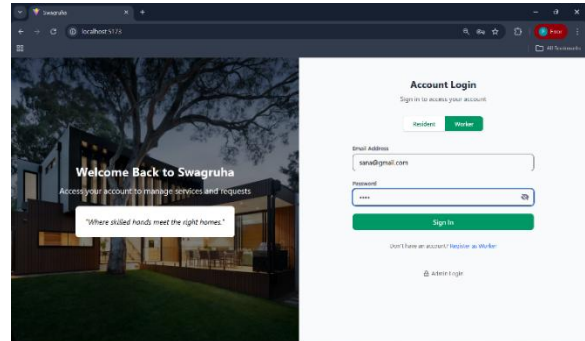
Resident Dashboard:



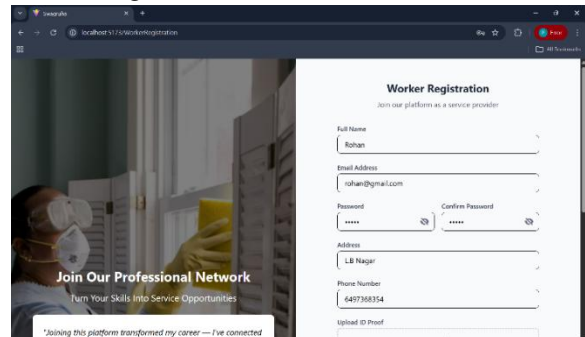
To book a service:

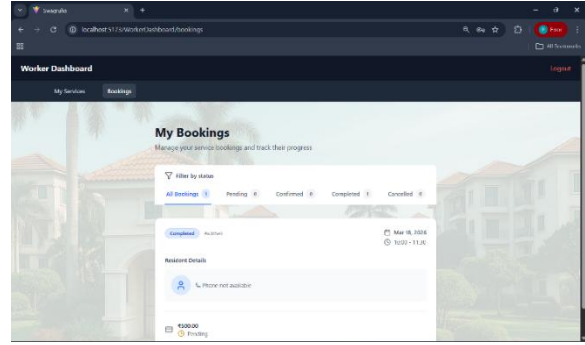
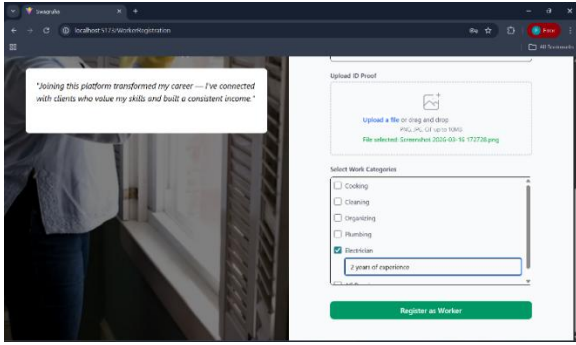


Worker Login:

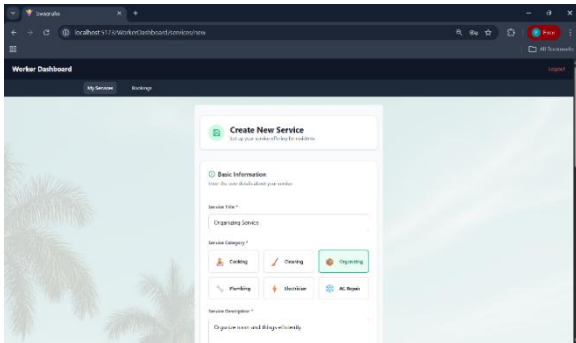


Worker Registration:

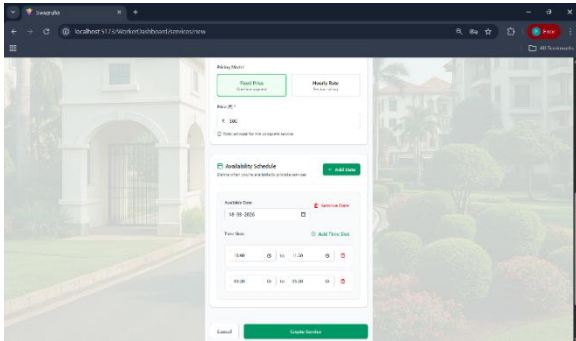
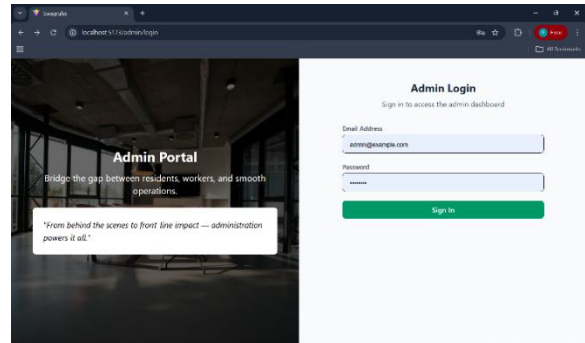




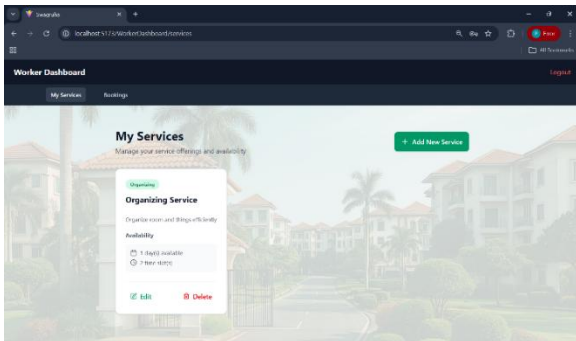
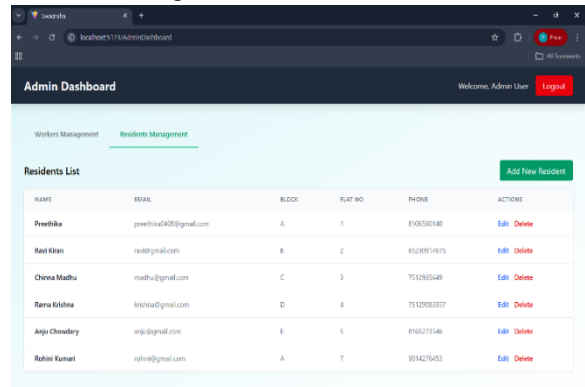
Worker Dashboard:



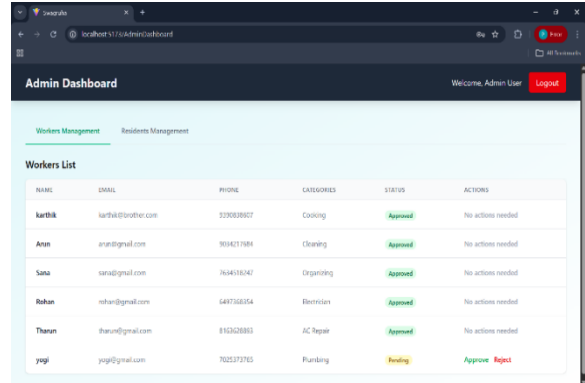
Admin login:



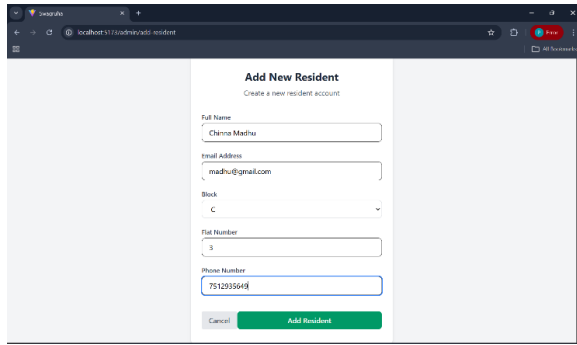
Resident Management List:



Workers Management List:



To add new resident:



The screenshot shows a web browser window with a form titled "Add New Resident". The form has a subtitle "Create a new resident account". It contains the following fields: "Full Name" with the value "Chitra Madhu", "Email Address" with the value "madhu@gmail.com", "Block" with a dropdown menu showing "C", "Flat Number" with the value "3", and "Phone Number" with the value "7512935694". At the bottom of the form, there are two buttons: "Cancel" and "Add Resident".

VIII.CONCLUSION

The Smart Utility Management System is a web-based platform developed using the MERN stack to simplify and streamline utility service management in gated communities. It provides a secure and centralized system that allows residents to book services and raise complaints, workers to manage availability and services, and administrators to monitor and control operations efficiently. By reducing manual coordination and improving transparency, the system enhances overall service delivery and user experience

IX.FUTURE SCOPE:

Future Scope of the Smart Utility Management System include integrating a secure in-app payment gateway with automated invoicing and digital receipts for seamless transactions. Real-time notifications through SMS, email, or push alerts will keep users informed about bookings, complaints, and service updates. A cross-platform mobile application with features like biometric login will improve accessibility and user convenience. Multilingual support and worker background verification will enhance inclusivity and resident safety. Additionally, AI-based service recommendations and calendar synchronization will further personalize and optimize service management.

REFERENCES:

1. M. Valenta and F. Antunes, Learning MERN Stack, Packt Publishing, 2019.
2. B. J. Patil, "Modern Web Applications Using the MERN Stack," International Journal of Emerging Technologies and Innovative Research, vol. 6, no. 5, pp. 42-47, 2021.

3. MongoDB Inc., "MongoDB Documentation," [Online]. Available: <https://www.mongodb.com/deca>
4. Express.js Team, "Express https://expressjs.com/. Node.js web application framework." [Online] Available:
5. Meta Platforms Inc., "React A JavaScript library for building user interfaces," [Online] Available:
6. Node.js Foundation, "Node.js Documentation," [Online]. Available: <https://nodejs.org/en/docs>.
7. GitLab Inc., "GitLab Docs Version Control and DevOps Lifecycle," [Online] Available: <https://docs.gitlab.com>.
8. Stripe Inc., "Stripe API Reference," [Online]. Available [tips//strips.com/docs/api](https://docs.stripe.com/api).
9. OWASP Foundation, "OWASP Top Ten Web Application Security Risks, [Online] Available. [https://ow.asp.org/www-project-top-son](https://owasp.org/www-project-top-son)
10. M. Fowler, Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
11. M. T. Rahman and R Akter. "Smart Home Utility Management Using Web-Based Systems" International Journal of Computer Applications, vol 180, no. 3, pp. 25-29, Feb. 2018.
12. S. S. Awasthi and P. Agarwal, "Design and Implementation of Complaint Management System in Gated Communities, International Journal of Scientific & Engineering Research, vol. 11. no. 8, pp. 140-145. Aug 2020.
13. Mozilla Developer Network (MDN), JWT JSON Web Tokens, [Online] Available: <https://deycheest.mozilla.org/en-US/docs/Web/HTTP/Authentication>
14. NIST. Security and Privacy Controls for Information Systems and Organizations, NIST SP 800-53 5. US Department of Commerce, 2020.