**A Smart Utility Management System**

**A PROJECT REPORT**

**for**

**Mini Project-I (K24MCA18P) Session**

**(2024-25)**

**Submitted by**

**Deepu Kumari**

**(202410116100031)**

**Anshu Patel**

**(202410116100034)**

**Dhwani Panchal**

**(202410116100063)**

**Submitted in partial fulfilment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**Under the Supervision of**

**Ms. Divya Singhal**

**Assistant Professor**



**Submitted to**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**(DECEMBER- 2024)**

# CERTIFICATE

Certified that **ANSHU PATEL <202410116100034>, DEEPU KUMARI <202410116100104>,** **DHWANI PANCHAL <202410116100063>** has/ have carried out the project work having  **“SMART UTILITY MANAGEMENT SYSTEM”** Mini Project-I, K24MCA18P) for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

**Ms. Divya Singal**  **Dr. Arun Kr.Tripathi**

Assistant Professor Dean

Department of Computer Applications Department of Computer

KIET Group of Institutions KIET Group of Institutions,

Ghaziabad Ghaziabad

# ABSTRACT

In today’s fast-paced world, accessing reliable and efficient home maintenance services is a growing challenge for customers. The Smart Utility Management System (SUMS) aims to bridge this gap by offering a digital platform that connects users with skilled professionals for services such as plumbing, painting, and general repairs. This system provides an intuitive interface for customers to book services conveniently and for service providers to manage their schedules effectively.

The system includes a robust customer management portal for service selection, a secure payment gateway for hassle-free transactions, and a feedback mechanism to ensure service quality. Additionally, a dedicated interface for service professionals allows them to showcase their skills, manage bookings, and build credibility through user ratings.

An admin dashboard ensures seamless operations by monitoring activities, resolving disputes, and maintaining service standards. By integrating features like real-time booking updates, service tracking, and notifications, the SUMS enhances transparency and user satisfaction.

This project employs modern technologies such as cloud computing for data storage, a mobile-responsive design for accessibility, and AI-based recommendations for personalized services. The Smart Utility Management System is an innovative step towards simplifying home service management, providing customers with convenience, and empowering local professionals.

# ACKNOWLEDGEMENTS

Success in life is never attained single-handedly. My deepest gratitude goes to my project supervisor, **Ms. Divya Singhal** for her guidance, help, and encouragement throughout my project work. Their enlightening ideas, comments, and suggestions.

Words are not enough to express my gratitude to Dr. Arun Kumar Tripathi, Professor and Dean, Department of Computer Applications, for his insightful comments and administrative help on various occasions.

Fortunately, I have many understanding friends, who have helped me a lot on many critical conditions.

Finally, my sincere thanks go to my family members and all those who have directly and indirectly provided me with moral support and other kind of help. Without their support, completion of this work would not have been possible in time. They keep my life filled with enjoyment and happiness.

**Anshu Patel (202410116100034)**

**Deepu Kumari (202410116100031)**

**Dhwani Panchal(202410116100063)**

# TABLE OF CONTENTS

[Certificate ii](#_Toc35124)

[Abstract iii](#_Toc35125)

[Acknowledgements iv](#_Toc35126)

[Table of Contents v](#_Toc35127)

1. Introduction 1-16
   1. Overview 2
      1. Significance of Digital Literature 4
         1. Overcoming Accessibility Barriers 4
         2. Sustaining Literary Heritage 7
      2. Audience Segmentation 10
         1. Addressing Varied Needs 10
   2. Technological Foundations 12
      1. Combining Scalability and Innovation 13
      2. User-Centric Features 13
   3. Vision and Goals 14
   4. Addressing Challenges15

Conclusion

* 1. 16

1. Feasibility Study/Literature Review 17-36
   1. Technical Feasibility 18
   2. Economic Feasibility 18 2.3 Operational Feasibility 35

2.4 Behavioral Feasibility 35

1. Project Objective
2. Hardware and Software Requirements
3. Project Flow
4. Project Outcome

References

**CHAPTER 1**

## INTRODUCTION

**1.1 Overview**

The **Smart Utility Management System (SUMS)** is a transformative platform that redefines how residential societies manage essential utilities and services. With a mission to integrate technology into everyday service management, SUMS aims to ensure accessibility, efficiency, and accountability in handling tasks such as plumbing, electrical work, cleaning, and more. It envisions a future where utility management is seamless, transparent, and universally efficient.

In today’s fast-paced world, traditional service management methods often fall short in addressing the growing complexities of modern living. SUMS is designed as a response to these challenges, bridging the gap between outdated processes and innovative digital solutions. The platform represents a perfect synergy of technology and practicality, offering a centralized, user-friendly interface that fosters streamlined operations, transparency, and superior user satisfaction.

Whether it's a resident booking a service, a society administrator overseeing operations, or a service provider managing tasks, SUMS delivers a tailored experience to make utility management efficient, reliable, and future-ready.

**1.1.1 Significance of Digital Utility Management**

The digital transformation of utility management offers a paradigm shift in how services are requested, delivered, and monitored. SUMS capitalizes on this transition by enhancing operational efficiency, accessibility, and accountability, contributing immense value to residential communities.

**Evolving Service Expectations**

Modern residents demand faster, more reliable, and flexible service options. SUMS

addresses these expectations by providing features like real-time scheduling, automated

notifications, and mobile-first accessibility.

**Enhanced Communication**  
The platform eliminates communication gaps between residents, administrators, and service providers, ensuring a smooth exchange of information and prompt issue resolution.

**Operational Efficiency**

By automating key tasks and centralizing data, SUMS reduces administrative burden,

eliminates inefficiencies, and improves the overall workflow for all stakeholders.

**1.1.1.1 Overcoming Traditional Challenges**

SUMS tackles key barriers in conventional utility management:

**Communication Inefficiencies**

Residents and service providers often face delays and miscommunication. SUMS

introduces real-time updates and automated alerts to keep all stakeholders informed.

**Scheduling Conflicts**

With a streamlined booking system, SUMS eliminates scheduling overlaps, ensuring

that residents receive timely service.

**Lack of Transparency**

SUMS provides detailed status updates, cost breakdowns, and feedback mechanisms,

promoting trust and accountability.

**1.1.1.2 Enabling Seamless Service Delivery**

The platform ensures efficient and high-quality service delivery:

**Service Provider Integration**

Skilled professionals can easily manage their schedules, track performance, and receive

ratings, motivating them to deliver exceptional service.

**Scalable Design**

The system can accommodate the needs of expanding societies and evolving

technologies, ensuring long-term relevance.

**Data Analytics**

Actionable insights help administrators optimize resource allocation, improve service

quality, and address emerging trends effectively.

**1.1.2 Audience Segmentation**

SUMS is designed to cater to diverse users, addressing their unique needs and expectations:

**Residents**

Offering a user-friendly portal for requesting, scheduling, and tracking services, along

with tools for feedback and issue escalation.

**Service Providers**

Providing a dedicated interface to manage bookings, monitor performance, and receive

real-time updates.

**Society Administrators**

Enabling centralized management of utilities, detailed reporting, and efficient resource

allocation.

**1.2 Technological Foundations**

The SUMS platform is built on a robust, scalable technological infrastructure that ensures reliability and adaptability.

**Frameworks**

Leveraging Django for its secure, modular, and scalable architecture, enabling rapid

development and system stability.

**Database Management**

Initially using SQLite for simplicity, with future scalability supported by PostgreSQL to

manage higher transaction volumes.

**Cloud-Based Infrastructure**

Hosting on AWS or Google Cloud ensures reliability, scalability, and consistent

performance under high traffic.

**1.2.1 Combining Innovation and Scalability**

The SUMS platform balances innovation with scalability:

**Modular Architecture**

Designed to adapt to future demands, including integrating IoT for smart home

solutions or AI-driven predictive maintenance.

**Performance Optimization**

Caching techniques and database indexing ensure swift response times and optimal

platform performance.

**1.2.2 User-Centric Features**

SUMS prioritizes user needs through its intuitive, efficient, and accessible design:

**Mobile-First Design**

Ensures users can access the platform anytime, anywhere, from their preferred devices.

**Real-Time Updates**

Automated notifications and live tracking enhance communication and reduce response

times.

**Personalization**

Tools like feedback forms, bookmarks, and customizable dashboards allow users to

tailor their experience.

**1.3 Vision and Goals**

SUMS envisions a world where utility management is effortless, transparent, and universally efficient.

**Seamless Access**

Empowering users with tools to manage utilities without barriers.

**Operational Excellence**

Providing administrators with actionable insights and data-driven decision-making

tools.

**Technological Integration**

Building a foundation for advanced technologies, ensuring the platform remains future

ready.

**1.4 Addressing Challenges**

While SUMS is a transformative platform, it must address key challenges for sustained

success:

**User Adoption**

Educating users on the benefits of the platform to drive widespread adoption and

engagement.

**Scalability**

Ensuring the platform can handle increased demand as residential societies expand.

**Technological Upkeep**

Continuously upgrading features and maintaining system reliability to meet evolving

user expectations.

**1.5 Conclusion**

The **Smart Utility Management System** represents a pioneering step in digital utility management. By merging cutting-edge technology with a user-centric approach, SUMS

transforms the way services are managed, ensuring efficiency, accountability, and accessibility.

With its mission to simplify and innovate, SUMS sets a benchmark for modern utility solutions, fostering well-managed, harmonious communities and setting the stage for

future advancements in service management.

**Chapter 2**

**FEASIBILITY STUDY**

## 2.1 Technical Feasibility

### This aspect assesses whether the available technology, tools, and resources are sufficient to build and deploy the platform.

### Technology Availability

### Frontend Technologies: Modern web development frameworks such as ReactJS, Angular, or Vue.js can be used to build a responsive and user-friendly interface for residents and service providers.

### Backend Technologies: Technologies like Node.js, Django, or Spring Boot provide robust platforms for developing scalable and efficient backend systems.

* **Database Management**: Relational databases like MySQL, PostgreSQL, or non- relational databases like MongoDB can handle user data, service requests, and transaction details.

### Notification System: Integration of services like Firebase Cloud Messaging or Twilio for automated notifications.

### Payment Integration: Secure payment gateways such as Stripe, PayPal, or Razorpay can ensure smooth financial transactions.

### Cloud Infrastructure: Platforms like AWS, Google Cloud, or Microsoft Azure can be utilized for hosting, ensuring high availability and scalability.

### Development Feasiblity

### Skills and Expertise: The required programming languages (e.g., JavaScript, Python, SQL) and frameworks are widely used, making it easy to assemble a competent development team.

### APIs and Integrations: Existing APIs for payment processing, geolocation, and messaging reduce the need for custom development and expedite project timelines.

### Platform Compatibility: The system can be designed to work across web browsers and mobile devices using responsive design and frameworks like React Native or Flutter.

### Scalability

### The platform’s architecture can be designed for scalability to handle increased users, service requests, and data as the system expands to more residential societies.

### Cloud-based solutions ensure scalability for storage, computing power, and network requirements.

## 2.2 Economic Feasibility

Economic feasibility examines the financial aspects of the project, including initial costs operational expenses, and potential revenue generation.

### 2.2.1 Initial Development Costs

**Development Costs**:

* Hiring a development team or outsourcing to a software company.
* Estimated cost: **$20,000 - $50,000**, depending on features and complexity.

**Infrastructure Costs** :

* Cloud hosting, databases, and API subscriptions.
* Estimated cost: **$5,000 - $10,000** for the first year.

**Testing and Deployment**:

* Costs for quality assurance, user testing, and initial rollout.
* Estimated cost: **$3,000 - $5,000**.
* **Scalability Benefits**: Incremental resource allocation ensures cost efficiency as the platform grows.

### 2.2.2 Ongoing Operational Costs

**Maintenance Updates** :

* Regular updates, bug fixes, and feature enhancements.
* Estimated cost: **$10,000 annually**.

**Customer Support**:

* A support team to handle queries and troubleshooting.
* Estimated cost: **$15,000 annually**.

**Marketing and Outreach**:

* Promoting the platform to new residential societies and service providers.
* Estimated cost: **$5,000 annually**.

**2.2.3 Revenue Generation**

**Subscription Fees**:

* Charging residential societies or individual residents a subscription fee for using the platform.
* Example: $5 per household per month.

**Service Provider Fees**:

* Charging service providers a percentage commission or subscription fee for accessing the platform.
* Example: 10% commission on completed services.

**Premium Features**:

* Offering premium features like priority booking, advanced analytics, or additional advertising space for service providers.

.

## Operational Feasibility

Operational feasibility evaluates whether the Smart Utility Management System can be implemented and sustained in real-world conditions, aligning with the needs of stakeholders and the resources available.

### 2.3.1 Alignment with Stakeholder Needs

The platform’s user-friendly design allows users to explore, read, and manage books effortlessly:

**Residents**:

* The platform simplifies utility service management, aligning with thedemand for convenience, transparency, and accountability.
* Real-time updates, easy scheduling, and a feedback system improve satisfaction and adoption rates.
* **Service Providers**:
* The system offers a centralized dashboard to manage schedules, track requests, and receive feedback, streamlining their operations.
* **Society Administrators**:
* Administrators can efficiently monitor and manage service operations, reducing manual intervention and improving decision-making.
* **Responsive Design**: Ensures smooth access across all devices, including desktops, tablets, and mobile phones.

### 2.3.2 Implementation Viability

* **Deployment**: The system can be deployed on a scalable cloud platform to ensure high availability and accessibility for users.
* **Operational Processes**: A dedicated support team can handle platform maintenance, address user queries, and onboard new residents and service providers.
* **Training and Onboarding**: Conducting workshops, providing user manuals, and offering 24/7 customer support ensures that all stakeholders are well-equipped to use the system effectively.

## 2.4 Behavioral Feasibility

### Behavioral feasibility evaluates how stakeholders (residents, service providers, and administrators) will interact with and accept the system.

### 2.4.1 Acceptance by Residents

### Ease of Use: The platform’s intuitive interface ensures that residents with varying levels of technical literacy can easily navigate the system.

### Convenience: The ability to schedule services, track progress, and provide feedback will likely appeal to residents, increasing adoption rates.

### Trust and Transparency: Real-time updates, a feedback system, and clear service pricing build trust and promote transparency.

### 2.4.2 Acceptance by Service Providers

### Efficiency in Scheduling: Providers will benefit from automated scheduling, which minimizes conflicts and optimizes their workflow.

### Feedback Mechanisms: Ratings and reviews allow service providers to improve their offerings, increasing their competitiveness.

### Revenue Opportunities: A centralized platform increases visibility and helps providers expand their client base, motivating them to participate actively.

**2.4.3 Acceptance by Society Administrators**

### Streamlined Operations: Administrators gain better visibility into service requests and provider performance, reducing manual effort and improving overall management.

### Data-Driven Decisions: Analytics and reporting modules enable administrators to make informed decisions for optimizing service delivery.

**Chapter 3**

**PROJECT OBJECTIVE**

The Smart Utility Management System aims to address inefficiencies in managing utility services within residential societies. The key objectives of the project are:

**3.1 Primary Objective**

* To develop a centralized platform that simplifies the process of booking, tracking, and providing feedback for utility services such as plumbing, electrical repairs, and cleaning.

### 3.1.2 Specific Goals

**Ease of Service Booking** :

* Enable residents to quickly and conveniently book utility services through a web based platform.
* Offer a user-friendly interface with clear service categories and booking options.

**Efficient Scheduling and Conflict Resolution**:

* Enable residents to quickly and conveniently book utility services through a web-based platform.
* Provide automatic notifications to users and service providers in case of changes or delays**.**

### 3.1.3 Real-time Service Tracking:

* Allow users to monitor the status of their requests (e.g., Pending, In Progress, Completed) in real time.
  + 1. **Transport Feedback and Rating Mechanism:**
* Facilitate a system where residents can leave feedback and rate service providers based on their experience.
* Aggregate ratings to ensure quality assurance and accountability among service providers.

* + 1. **Enhanced User Experience:**
* Ensure seamless navigation and accessibility for all users.
* Minimize manual intervention by automating routine processes like notifications, reminders, and updates.
* **Adjustable Reading Interface**: Personalization tools like font size adjustment, day/night mode, and theme selection cater to individual reading comfort.

* + 1. **Scalability and Reliability:**
* Design the system to accommodate the growing needs of residential societies by supporting multiple users and service categories.
* Ensure secure and reliable data management using modern tools like MongoDB Atlas and Firebase Authentication.
  + 1. **Empowerment of Service Providers:**

Provide service providers with tools to manage their schedules, view booking details, and improve their performance through feedback.

The overarching aim of the project is to create a smart, scalable solution that enhances convenience for residents while ensuring efficient management of utility services.

## Chapter 4: HARDWARE AND SOFTWARE REQUIREMENTS

## 4.1 Hardware Requirements

To ensure the development and deployment of a robust and efficient Smart Utility Management System, the following hardware and software requirements are specified:

**4.1.1 Development Environment**

### Processor: Intel i5/i7 or AMD equivalent (quad-core or higher)

### RAM: Minimum 8 GB (16 GB recommended for seamless multitasking)

### Storage: At least 256 GB SSD or 500 GB HDD for storing development tools, project files, and database backups.

### Display: Full HD (1920x1080) monitor or higher resolution for better interface design and debugging.

* **Internet Connectivity**: Stable and high-speed broadband for real-time testing, API integrations, and cloud deployments.

### Client Requirements

### Device: Desktop, laptop, tablet, or smartphone.

### Browser: Any modern web browser like Google Chrome, Mozilla Firefox, Microsoft Edge, or Safari.

### Internet Connection: Reliable connection for accessing the web-based system.

* + 1. **Server Requirements (If Deployed Online)**
* **Cloud Hosting:** AWS, Google Cloud, or Azure to host the backend and database.
* **Node.js Environment:** For backend application deployment.
* **Database Storage:** Cloud-supported MongoDB instance for data storage and retrieval.
  + 1. **Minimum Specifications:**
* 2 vCPU processors
* GB RAM
* 50 GB SSD storage

## 4.2 Software Requirements

**4.2.1 Programming Languages and Frameworks**

* **Frontend Development:**
* **Languages:** HTML, CSS, JavaScript.
* **Frameworks:** React.js (optional) for building a dynamic and responsive user interface.

### 4.2.2 Backend Requirements

* **Framework:** Node.js with Expresss.js for creating RESTful APIs and server-side logic.
* **Programming Language:** JavaScript.

### 4.2.3 Database

* **MongoDB Atlas:** A cloud-based No SQL database for storing user data, service requests, feedback, and scheduling details.

### 4.2.4 Development Tools

* **Code Editor:** Visual Studio Code(VS Code) for efficient code writing and debugging.
* **Version Control:** GitHub or Git for version tracking, collaboration, and repository management.

### 4.2.6 Authentication and Security

* **Firebase Authentication:**
* For secure authentication , password management, and role-based access.
* **CORS (Cross-Origin Resource Sharing):** To manage API security and prevent unauthorized access.

**4.2.7 Utilities and APIs**

* **Nodemon:** For automatic server restarts during development.
* **Postman**: To tes and debug API endpoints

**4.2.8 Operating System**

**Nodemon :**

For automatic server restarts during development.

**Postman:**

To test and debug API endpoints.

**Chapter 5**

**PROJECT FLOW**

## The Smart Utility Management System follows a systematic and user-centric workflow, ensuring efficient operation and smooth user experiences. The following outlines the step-by-step project flow:

## 5.1 User Authentication and Registration

**Process:**

* Residents or administrators create an account by entering personal details such as name, email, and phone number.
* Authentication is handled using Firebase Authentication to ensure secure login credentials.

• **Outcome:**

* Successful login gives users access to their personalized dashboard based on their role (Resident/Admin).

## 5.2 Dashboard Access

* **Resident dashboard**:
* Displays available utility services (e.g., plumber, electrician, cleaner).
* Shows the history of booked services, current bookings, and feedback submissions.
* **Admin Dashboard**:
* Allows tracking of all service requests, managing service providers, and resolving disputes.

## 5.3 Sevice Request Creation

**Process:**

* Residents select the type of service required (e.g., plumbing, painting) and specify details such as preferred date, time, and additional requirements.
* Booking details are verified for conflicts, and the system assigns a service provider.
* **Outcome**:
* A confirmed booking is created with real-time status updates and notifications.

## 5.4 Service Provider Assignment

* **Process**:
* The system automatically matches the service request with an available provider based on location, expertise, and schedule.
* **Outcome**:
* The provider is notified of the new assignment and can accept or reschedule as necessary.

## 5.5 Real-TimeStatus Tracking

**Process:**

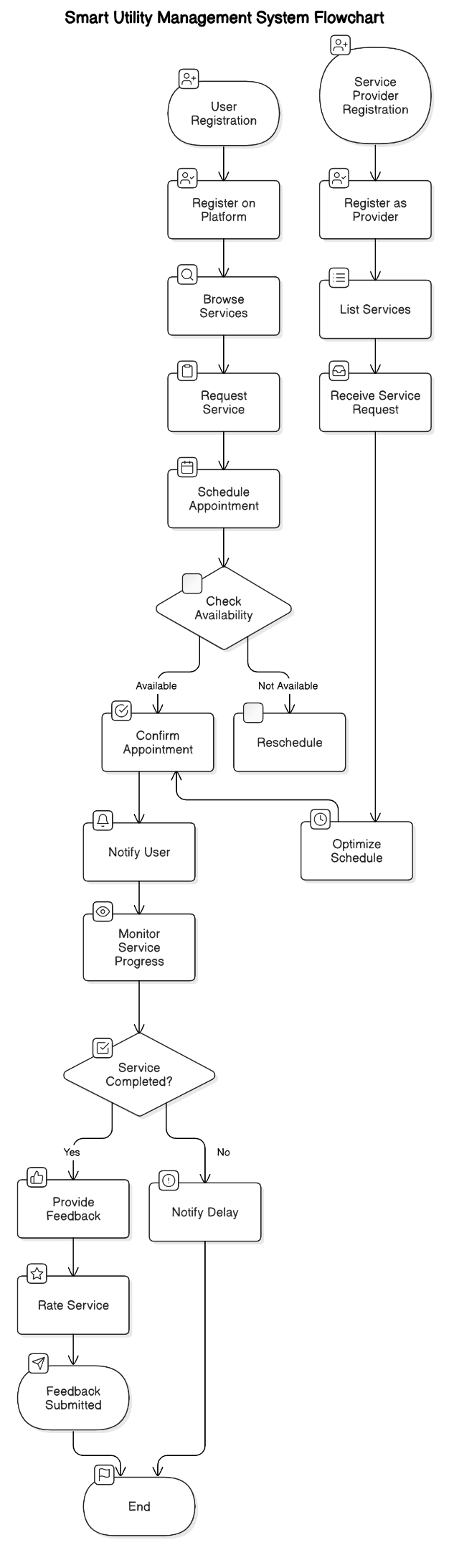
* Users can track the status of their service request via the dashboard (e.g., Pending, In Progress, Completed).
* Notifications are sent at critical stages, such as service start and completion.

**Outcome:**

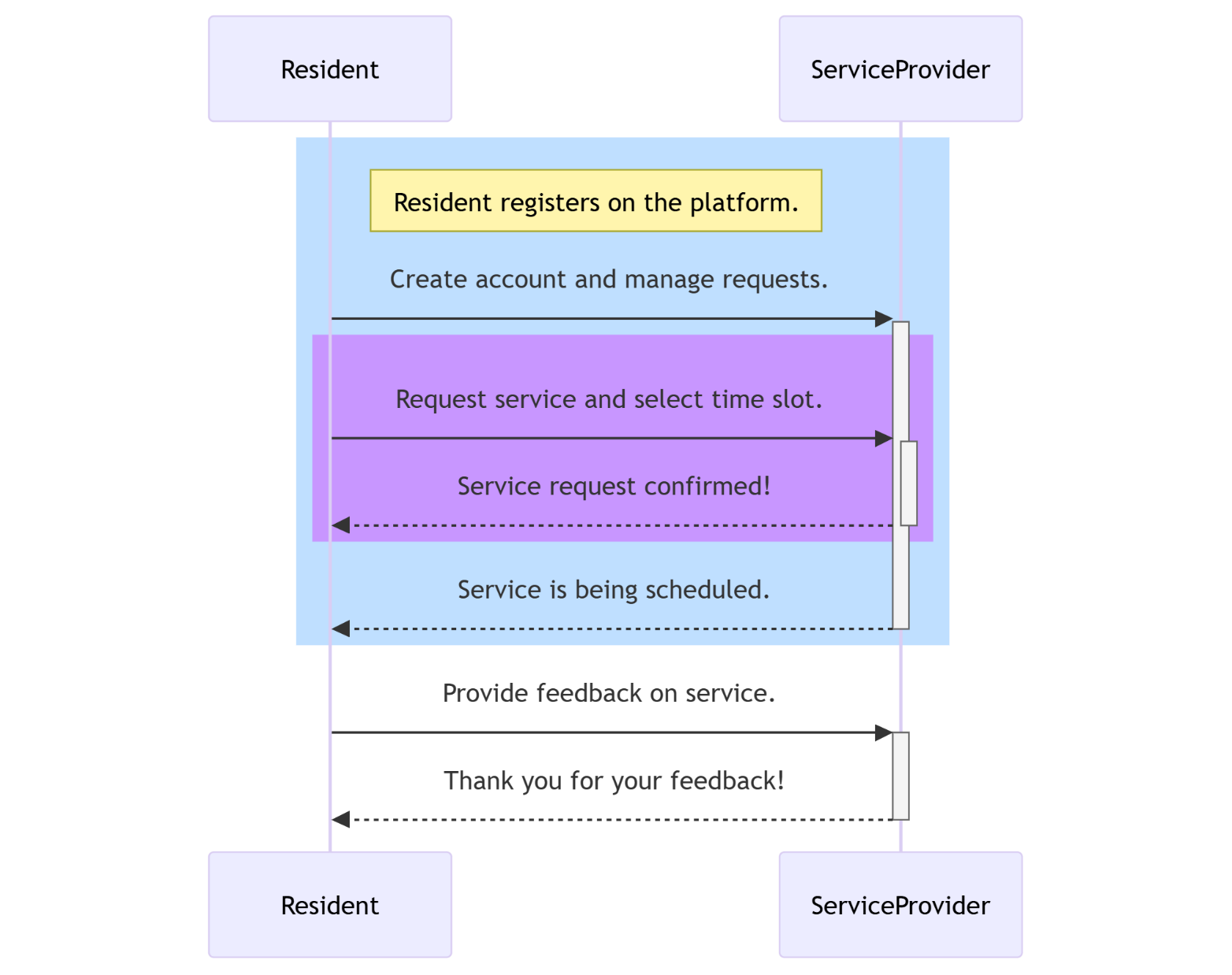
* Transparency in service delivery is maintained, enhancing user trust and satisfaction.

.

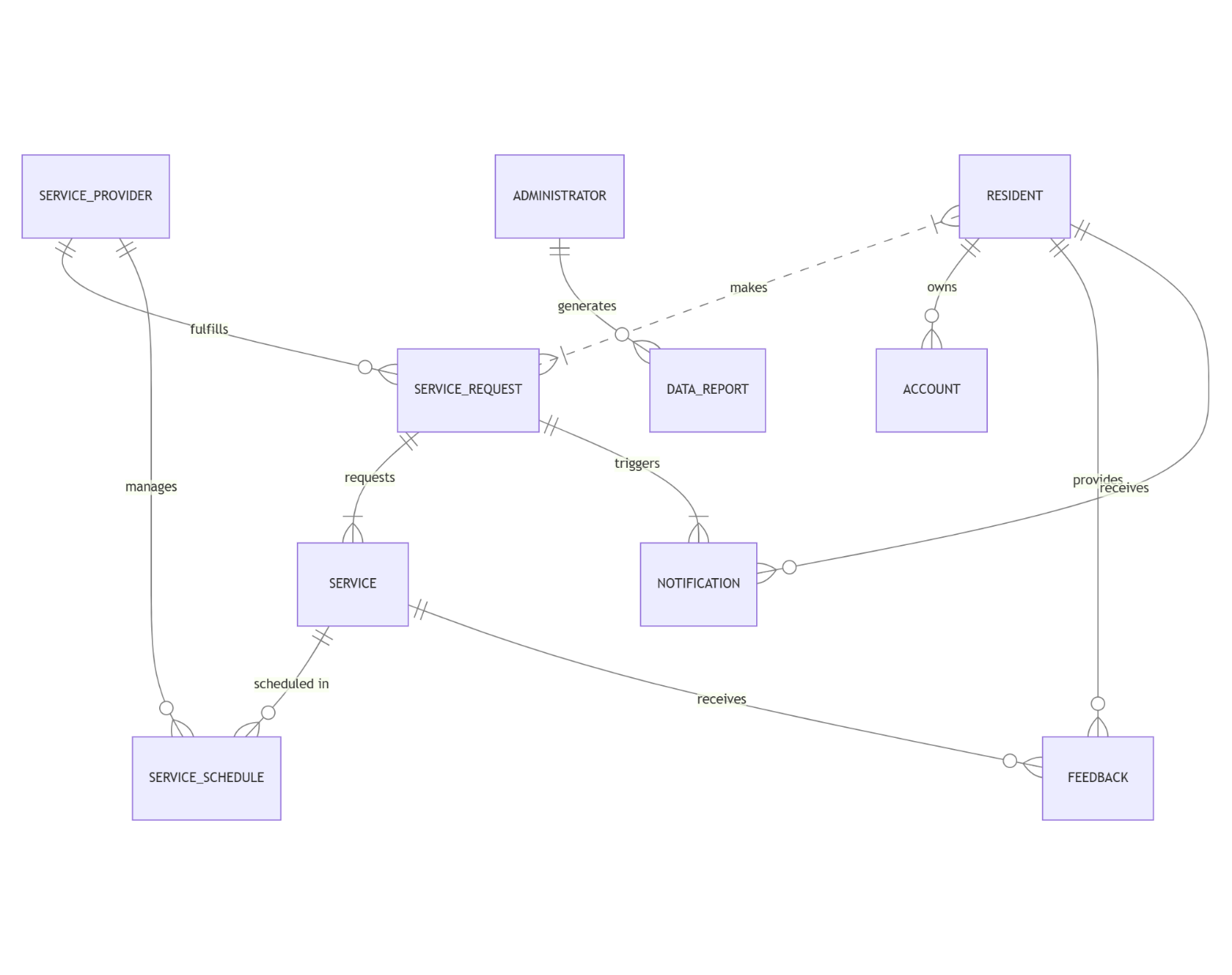
## 5.5 Data flow diagram



**5.6 Sequence diagram**



## 5.7 E-R diagram



**Chapter 6**

**PROJECT OUTCOME**

The Smart Utility Management System successfully addresses the core challenges of traditional utility service management, delivering transformative benefits for residential societies. Key outcomes of the project include: content rather than being overwhelmed by visual clutter.

**6.1. User-Centric Benefits**

**6.1.1 Simplified Registration and Service Requests**  
Residents can easily register, update profiles, and submit service requests with just a few clicks, fostering ease of use and accessibility.

**6.1.2 Personalized Experience**  
Residents receive tailored notifications and updates, creating a more personal and engaging experience.

**6.1.3 Transparent Billing and Cost Tracking**  
Residents can view detailed cost breakdowns and service histories, ensuring financial transparency and informed decision-making.

**6.1.4 Convenience and Time-Saving**  
The ability to book, track, and rate services through a unified platform significantly reduces the time and effort required for managing day-to-day utility needs.

**6.1.5 Improved Safety and Reliability**  
Verified service providers ensure a higher degree of trust, offering residents peace of mind about the quality and safety of the services rendered.

**6.2. Operational Enhancements**

**6.2.1 Efficient Service Coordination**  
The platform eliminates manual coordination efforts by automating scheduling and notifications, minimizing delays and miscommunication.

**6.2.2 Enhanced Service Provider Performance**  
Integrated dashboards for service providers enable better schedule management, tracking of requests, and performance optimization through resident feedback.

**6.2.3 Increased Accountability**  
Real-time tracking, feedback mechanisms, and performance analytics ensure that service providers maintain high service standards.

**6.2.4 Reduction in Administrative Burden**  
Society administrators benefit from streamlined workflows and automated reporting, allowing them to focus on higher-value tasks.

* + 1. **Structured Data Management**  
       The system centralizes all service data, providing easy access for audits, performance reviews, and future planning.

**6.3. Technology-Driven Advancements**

* + 1. **Scalability for Expanding Communities**  
       The platform’s modular design allows for easy integration of additional features and accommodates larger societies as their needs grow.
    2. **Real-Time Data Analytics**  
       Actionable insights generated through data analytics help identify trends, improve service delivery, and forecast demand effectively.
    3. **IoT Integration Potential**  
       The system can connect with IoT-enabled devices for services like automated fault detection in plumbing or electrical systems, streamlining maintenance processes.
    4. **AI-Driven Features**  
       The foundation is set for integrating AI-powered features such as dynamic pricing, personalized recommendations, and predictive maintenance alerts, ensuring future relevance.
    5. **Cloud-Based Flexibility**  
       The use of cloud infrastructure enables seamless data access, scalability, and system reliability across all devices.
    6. **Offline and Emergency Support**  
       Features like offline booking or predefined emergency response workflows enhance reliability during unexpected situations or network outages.
    7. **Cross-Platform Consistency**  
       Uniform performance across web and mobile platforms ensures uninterrupted user engagement, regardless of device preferences.
    8. **Data Security and Privacy**  
       Advanced encryption and secure access controls protect user data, fostering trust and compliance with privacy standards.

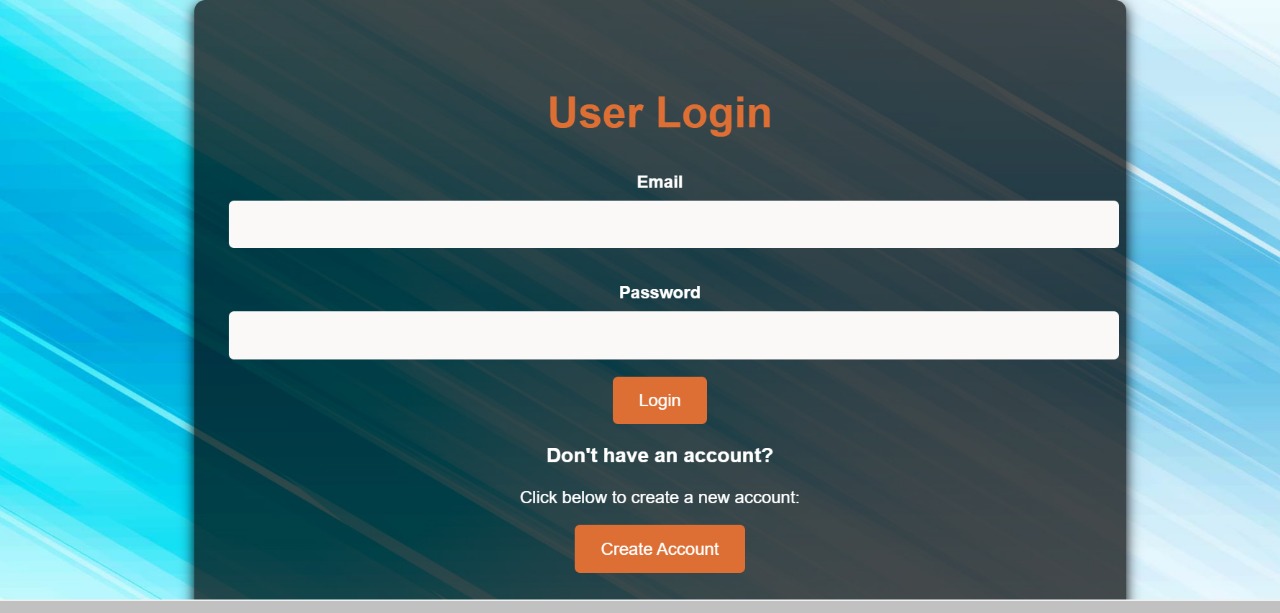
**6.4. Social and Environmental Impact**

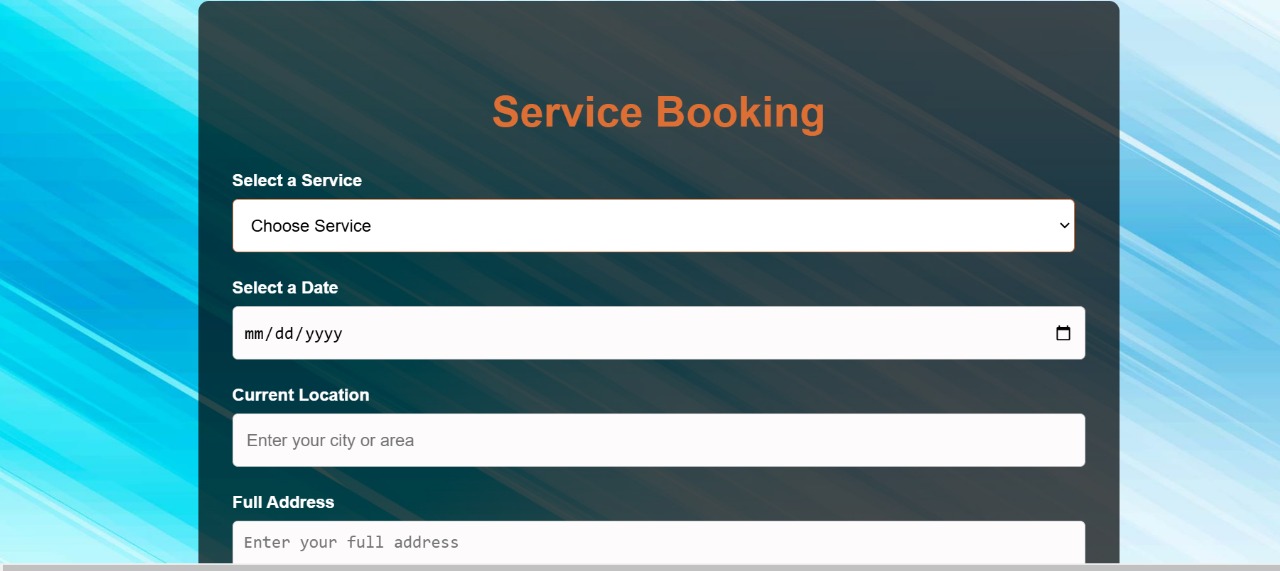
* + 1. **Community Development**  
       By offering transparency and fostering collaboration between residents and service providers, the platform strengthens community relationships.
    2. **Sustainable Practices**  
       Digitized workflows reduce paper-based processes and resource wastage, contributing to environmentally sustainable operations.
    3. **Skill Development for Providers**  
       Service providers benefit from structured feedback and insights, motivating them to improve their skills and service quality.
    4. **Support for Local Businesses**  
       The system offers a platform for skilled professionals and small businesses to expand their reach, contributing to economic growth at a community level.
    5. **Reduced Carbon Footprint**  
       Optimized scheduling reduces unnecessary travel by service providers, lowering fuel consumption and emissions.
  + .
* **Responsive Design**: 6.6 Final Deliverables (Snapshots)

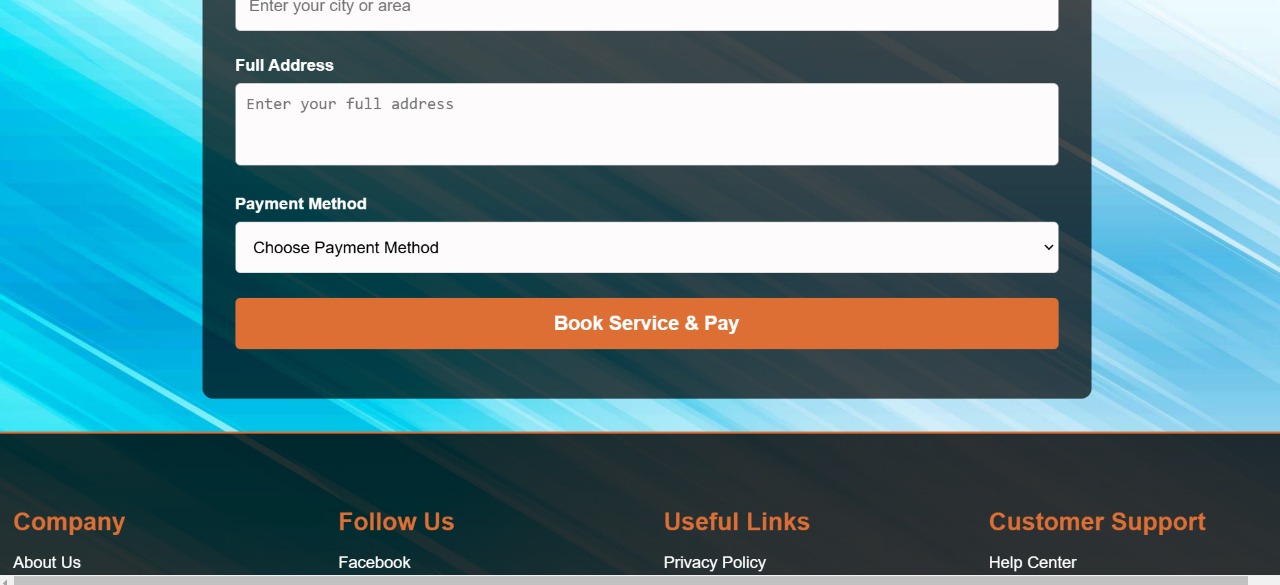
**i)**Home page:



ii)Log in page:



****



**Chapter 7 REFERENCES**

1. Django Documentation: [https://docs.djangoproject.com](https://docs.djangoproject.com/)
2. Python Official Documentation: [https://www.python.org](https://www.python.org/)
3. Bootstrap Official Site: [https://getbootstrap.com](https://getbootstrap.com/)
4. Project Gutenberg: [https://www.gutenberg.org](https://www.gutenberg.org/)
5. Gutenberg API: [https://gutendex.com](https://gutendex.com/)
6. GitHub: [https://github.com](https://github.com/)
7. SQLite Official Site: [https://www.sqlite.org](https://www.sqlite.org/)
8. Visual Studio Code: [https://code.visualstudio.com](https://code.visualstudio.com/)
9. Postman API Tool: [https://www.postman.com](https://www.postman.com/)
10. W3Schools (HTML, CSS, JavaScript): [https://www.w3schools.com](https://www.w3schools.com/)
11. Mozilla Developer Network (MDN): [https://developer.mozilla.org](https://developer.mozilla.org/)
12. Stack Overflow: [https://stackoverflow.com](https://stackoverflow.com/)
13. Python Package Index (PyPI): [https://pypi.org](https://pypi.org/)
14. Medium (Django Articles): [https://medium.com](https://medium.com/)
15. Git Official Site: [https://git-scm.com](https://git-scm.com/)
16. FreeCodeCamp: [https://www.freecodecamp.org](https://www.freecodecamp.org/)
17. Real Python Tutorials: [https://realpython.com](https://realpython.com/)
18. Bootstrap Icons: https://icons.getbootstrap.com
19. Django for Beginners by William S. Vincent: [https://djangoforbeginners.com](https://djangoforbeginners.com/) 20. Chrome DevTools: https://developer.chrome.com/docs/devtools