

CIVIC ISSUE REPORTING AND TRACKING

**A Dissertation submitted in partial fulfillment of the requirement
for the award of the degree of
MASTER OF COMPUTER APPLICATIONS
of
Visvesvaraya Technological University**



**By
SRAJAN KUMAR
1BI24MC144
Under the Guidance of**

Internal Guide:

Prof. N RAJESHWARI
Assistant Professor
Department of MCA
BIT, Bengaluru-04.



DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

BANGALORE INSTITUTE OF TECHNOLOGY
(An Autonomous Institute Affiliated to VTU)
K. R. Road, V. V. Pura, Bengaluru – 560004

December - 2025

CIVIC ISSUE REPORTING AND TRACKING

A Dissertation submitted in partial fulfillment of the requirement
for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

of

Visvesvaraya Technological University

By

SRAJAN KUMAR

1BI24MC144

Under the Guidance of

Internal Guide:

Prof. N RAJESHWARI

Assistant Professor

Department of MCA

BIT, Bengaluru-04.



DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

BANGALORE INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated To VTU)

K. R. Road, V. V. Pura, Bengaluru – 560004

December - 2025



**DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS
BANGALORE INSTITUTE OF TECHNOLOGY**

(An Autonomous Institute Affiliated To VTU)
K. R. Road, V. V. Pura, Bengaluru – 560004

CERTIFICATE

This is to certify that Srajan Kumar bearing IBI24MCI44 has completed his final semester project work entitled “Civic issue reporting and tracking” as a partial fulfillment for the award of Master of Computer Applications degree, during the academic year 2025 - 2026 under our supervision.

Rajeshwari N

Prof N RAJESHWARI
Assistant professor
Department of MCA
BIT, Bengaluru-560004

Madhukar K

Dr. MADHUKAR K
Professor & Head
Department of MCA
BIT, Bengaluru -560004

Shashi S

PRINCIPAL
Bangalore Institute of Technology

*R.R. Road, V.V.Pura
Dr. SHANTHALA S
Principal
BIT, Bengaluru -560004*

Name of the examiner

Signature with date

1.

2.



BANGALORE INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated To VTU)

K. R. Road, V.V. Pura, Bangalore – 560 004, India. Phone: 26613237 Fax: 22426796

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

Project Completion Certificate

Dear Sir/Madam,

We certify that **SRAJAN KUMAR** bearing **1BI24MC144** from Department of MCA, Bangalore Institute of Technology Bangalore, has successfully completed his project work with **CIVIC ISSUE REPORTING AND TRACKING** from 24th September 2025 to 27th December 2025 under the guidance of Prof. N RAJESHWARI.

We found that he was hard working. We wish him/her all the best for his future endeavors.



Dr. MADHU H. K.
Professor & Head
Department of MCA
BIT, Bengaluru-560004

Vision : To transform young graduates into skilled computer professionals to meet industrial and societal needs.
Mission: To enhance the Teaching learning process to meet quality education in the field of computer applications.
To impart the knowledge in current technologies to meet the industrial needs.
To inculcate ethical values and leadership qualities for the betterment of society.

BANGALORE INSTITUTE OF TECHNOLOGY
(An Autonomous Institute Affiliated To VTU)
K.R. Road, V.V. Pura, Bengaluru-560004



Vision

To establish and develop the Institute as a centre of higher learning, ever abreast with expanding horizon of knowledge in the field of engineering and technology, with entrepreneurial thinking, leadership excellence for life-long success and solve societal problem.

Mission

- Provide high quality education in the engineering disciplines from the undergraduate through doctoral levels with creative academic and professional programs.
- Develop the Institute as a leader in Science, Engineering, Technology and management, Research and apply knowledge for the benefit of society.
- Establish mutual beneficial partnerships with industry, alumni, local, state and central governments by public service assistance and collaborative research.
- Inculcate personality development through sports, cultural and extracurricular activities and engage in the social, economic and professional challenges.

BANGALORE INSTITUTE OF TECHNOLOGY
(An Autonomous Institute Affiliated To VTU)
K.R. Road, V.V. Pura, Bengaluru-560004

Department of Master of Computer Applications



Vision

To transform young graduates into skilled computer professionals to meet Industrial and societal needs.

Mission

- To enhance the Teaching learning process to meet quality education in the field of Computer applications
- To impart the knowledge in current technologies to meet the industrial needs
- To inculcate ethical values and leadership qualities for the betterment of society

PEO's

PEO1: To develop quality application software with innovative ideas to meet the industrial requirements

PEO2: To imbibe the current technologies and to adopt in computing profession as per the changing needs

PEO3: To progress in their career with leadership qualities and ethical values that enhances self and societal growth

PROGRAM OUTCOME

PO1	Foundation Knowledge: Apply knowledge of mathematics, programming logic and coding fundamentals for solution architecture and problem solving.
PO2	Problem Analysis: Identify, review, formulate and analyse problems for primarily focussing on customer requirements using critical thinking frameworks.
PO3	Development of Solutions: Design, develop and investigate problems with an innovative approach for solutions incorporating ESG/SDG goals
PO4	Modern Tool Usage: Select, adapt and apply modern computational tools such as development of algorithms with an understanding of the limitations including human biases.
PO5	Individual and Teamwork: Function and communicate effectively as an individual or a team leader in diverse and multidisciplinary groups. Use methodologies such as agile.
PO6	Project Management and Finance: Use the principles of project management such as scheduling, work breakdown structure and be conversant with the principles of Finance for profitable project management.
PO7	Ethics: Commit to professional ethics in managing software projects with financial aspects. Learn to use new technologies for cyber security and insulate customers from malware.
PO8	Life-long learning: Change management skills and the ability to learn, keep up with contemporary technologies and ways of working.



BANGALORE INSTITUTE OF TECHNOLOGY

(An Autonomous Institute Affiliated To VTU)

K. R. Road, V.V. Pura, Bangalore – 560 004, India. Phone: 26613237 Fax: 22426796

DEPARTMENT OF MASTER OF COMPUTER APPLICATIONS

CO-PO MAPPING FOR THE ACADEMIC YEAR 2025-26 (ODD)

BATCH: 2024

Semester: III

Subject: Project Work

Subject Code: MPRJ384

CO-POs with Bloom's Level:

CO	Statement	POs
1	Communication and present the ideas to defend the project effectively, clearly and coherently in both the written and oral forms.	PO4, PO7 PO8
2	Co-relate different areas for knowledge to generate, develop and evaluate ideas and information so as to apply these skills to the project task	PO3, PO4, PO5
3	Apply complex solving skills to develop a solution using modern tool	PO2, PO3, PO4
4	Formulate and reflect on their own or with the team to take appropriate actions to improve it.	PO5, PO8

CO-PO Mapping:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8
CO1				3			2	2
CO2			2	3	3			
CO3		3	2	3				
CO4					3			3

DECLARATION

I, Srajan Kumar Student of 3rd Semester MCA, Bangalore Institute of Technology, bearing USN **1BI24MC144** hereby declare that the project entitled **Civic Issue Reporting and Tracking** has been carried out by me under the supervision of internal Guide Prof. N RAJESHWARI Assistant Professor and submitted in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications by the Vivesvaraya Technological University during the academic year 2025-2026. This report has not been submitted to any other Organization / University for any award of degree or certificate.

Name: Srajan Kumar

Signature: 

ACKNOWLEDGEMENT

The satisfaction and happiness that accompany the successful completion of any task would be incomplete without expressing gratitude to the people who made it possible..

I consider it a privilege to express my sincere gratitude to **Dr.Shanthala S**, Principal, Bangalore Institute of Technology, Bengaluru, for providing all the necessary facilities required for the successful completion of this project.

I extend my heartfelt thanks to **Dr. H. K. Madhu**, Head of the Department, Master of Computer Applications, Bangalore Institute of Technology, Bengaluru, for his continuous encouragement and valuable guidance throughout the course of this project.

I sincerely convey my respect and gratitude to my internal guide **Prof. N. Rajeshwari**, Assistant Professor, Department of MCA, Bangalore Institute of Technology, Bengaluru, for her constant support, guidance, and cooperation during the development of this project.

I also thank all the teaching and non-teaching staff of the Department of MCA for their assistance and support, which contributed significantly to the successful completion of this project.

Finally, I express my sincere gratitude to my family and friends for their continuous encouragement and moral support throughout this endeavor.

Srajan Kumar

1BI24MC144

ABSTRACT

CleanStreet is a web-based civic complaint management system developed to improve the way urban issues are reported, tracked, and resolved. In many cities, civic problems such as garbage dumping, potholes, broken streetlights, and drainage issues are still handled through manual or fragmented processes, leading to delayed responses, lack of transparency, and poor communication between citizens and authorities. CleanStreet addresses these challenges by providing a centralized, digital platform for efficient civic issue reporting and monitoring.

The system enables citizens to register, log in, and report civic issues by submitting descriptions, images, and geo-tagged locations through an intuitive web interface. Each complaint is stored securely and assigned a status that can be tracked in real time. Administrators can review complaints, assign them to appropriate volunteers or departments, update resolution progress, and manage users. Volunteers can view assigned issues and update their work status, ensuring accountability and faster resolution.

CleanStreet is developed using modern web technologies, with a responsive frontend for seamless user interaction and a backend that manages authentication, complaint processing, and data storage. Interactive dashboards and visual analytics provide insights into complaint trends, priority distribution, and resolution performance.

By promoting transparency, community participation, and efficient coordination, CleanStreet enhances urban governance and public service delivery. The system demonstrates how digital solutions can empower citizens, support authorities, and contribute to cleaner, safer, and smarter cities.

TABLE OF CONTENTS

SL. NO.	CONTENTS	PAGE NO.
1	INTRODUCTION	1
	1.1 PROJECT DESCRIPTION	1
2	LITERATURE SURVEY	2
	2.1 EXISTING SYSTEM	4
	2.2 PROPOSED SYSTEM	5
	2.3 FEASIBILITY STUDY	6
	2.4 TOOLS AND TECHNOLOGIES USED	8
	2.5 HARDWARE AND SOFTWARE REQUIREMENTS	9
3	SOFTWARE REQUIREMENTS SPECIFICATION	10
	3.1 FUNCTIONAL REQUIREMENTS	11
	3.2 NON-FUNCTIONAL REQUIREMENTS	12
4	SYSTEM DESIGN	14
	4.1 SYSTEM PERSPECTIVE	15
	4.2 FLOW CHART	16
5	DETAILED DESIGN	17
	5.1 USECASE DIAGRAM	17
	5.2 DATA FLOW DIAGRAM	18
	5.3 CLASS DIAGRAM	20
6	IMPLEMENTATION	23
	6.1 CODE SNIPPETS	23
	6.2 SCREENSHOTS	31
7	SOFTWARE TESTING	38
8	CONCLUSION	41
9	FUTURE ENHANCEMENT	42
APPENDIX A	BIBLIOGRAPHY	43
APPENDIX B	USER MANUAL	45

LIST OF FIGURES

Sl. No.	TOPIC		PAGE NO.
4	4.2	FLOW CHART	16
5	5.1	USE CASE DIAGRAM	17
	5.2	DATA FLOW DIAGRAM	18
	5.3	CLASS DIAGRAM	20
	5.4	SEQUENCE DIAGRAM	21
	5.5	ACTIVITY DIAGRAM	22

CHAPTER 1

INTRODUCTION

1.1 PROJECT DESCRIPTION

Clean Street is a civic issue reporting and tracking web application designed to enhance communication between citizens and local governing bodies for faster and more transparent resolution of public infrastructure problems. The platform provides a structured digital channel through which users can report common civic issues such as uncollected garbage, damaged roads, water leakage, and faulty streetlights along with supporting images and precise location details.

The system captures user-submitted data and securely stores it in a centralized database, where it is organized based on issue type, geographic location, priority level, and time of submission. This structured approach allows authorities and volunteers to easily review, assign, and act upon complaints without manual intervention. Automated routing ensures that each reported issue reaches the appropriate responsible unit based on location and category.

Clean Street promotes active community participation by enabling users to track the progress of reported issues in real time. Features such as voting and commenting allow community members to highlight critical problems and share additional observations, helping authorities prioritize tasks that affect a larger population. Status updates provide transparency and keep users informed throughout the resolution process.

An administrative dashboard is included to support monitoring and decision-making. Administrators can oversee complaint trends, manage users and zones, analyze resolution performance, and generate reports for evaluation and planning purposes. These insights help improve operational efficiency and accountability in civic services.

The application is built using a modern web technology stack, with a responsive frontend for seamless user interaction and a high-performance backend for secure data handling and fast processing.

CHAPTER 2

LITERATURE SURVEY

E-Governance Systems for Civic Complaint Management: A Survey

S. Nam, J. R. Pardo – *Government Information Quarterly*, 2018

This study reviews digital civic complaint management systems adopted by municipal authorities. It concludes that online grievance platforms with features such as real-time status tracking and feedback mechanisms significantly enhance transparency, accountability, and citizen engagement, leading to more efficient resolution of civic issues.

Smart City Applications for Urban Issue Reporting and Monitoring

UN Department of Economic and Social Affairs – *UN E-Government Survey*, 2020

This survey examines smart city initiatives across multiple countries, focusing on ICT-enabled platforms for reporting and monitoring urban issues. The findings highlight the importance of centralized complaint systems and public dashboards in improving service delivery and urban governance.

Smart Waste Management Systems: A Survey of Recent Technologies

A. K. Gupta, S. Sharma – *International Journal of Environmental Science and Technology*, 2021

This paper analyzes intelligent waste management solutions used in urban environments. It emphasizes the role of citizen-reported complaints, geo-tagged images, and administrative dashboards in improving sanitation monitoring and reducing response time.

Citizen Participation and Crowdsourcing in Urban Governance

D. Brabham – *Journal of Public Administration Research and Theory*, 2017

This research explores crowdsourcing-based civic platforms and highlights how public participation through voting and feedback mechanisms helps authorities prioritize critical issues and supports faster decision-making.

Transparency and Accountability in Digital Public Service Platforms

J. C. Bertot, P. T. Jaeger, J. M. Grimes – *Government Information Quarterly*, 2019

The study focuses on transparency mechanisms in digital public service systems. It concludes that providing citizens with access to complaint status and administrative actions increases trust and strengthens accountability in government services.

Mobile-Based Civic Issue Reporting Applications: A Systematic Review

M. Aloudat, S. Michael – *IEEE Access*, 2021

This review evaluates mobile and web-based civic issue reporting applications with emphasis on usability, image-based reporting, and notification features. The study finds that mobile accessibility and simplified interfaces encourage higher citizen participation.

Community-Driven Urban Problem Identification Systems

H. Zhang, Y. Liu – *Sustainable Cities and Society*, 2022

This research analyzes community-reported urban issues across cities and demonstrates that combining citizen input with analytical tools helps identify high-priority problems and supports data-driven governance.

Data Visualization Techniques for Urban Service Analytics

L. Chen, R. Kumar – *Sustainable Cities and Society*, 2023

The study highlights the role of interactive dashboards and visual analytics in understanding complaint trends, service performance, and area-wise issue distribution, aiding effective administrative planning.

Impact Analysis of Digital Grievance Redressal Systems in Municipal Services

A. Singh, P. Verma – *Journal of Urban Technology*, 2019

This paper evaluates digital grievance redressal systems using a mixed-method approach. The results show measurable improvements in response efficiency, citizen satisfaction, and overall service quality.

2.1 EXISTING SYSTEM

In many urban administrations today, civic issue reporting and street cleanliness management are still handled using traditional or partially digitized methods. Municipal bodies often rely on manual complaint registers, phone-based reporting, basic web portals, or standalone spreadsheets to record issues related to garbage accumulation, road damage, drainage problems, and streetlight failures. While these methods were sufficient in the past when complaint volumes were limited, they struggle to cope with the growing scale and complexity of modern cities.

The primary limitation of existing systems lies in their restricted monitoring and analysis capability. Most platforms focus only on registering complaints and maintaining basic records, without considering broader factors that influence civic issue severity and resolution. Geographic patterns, complaint frequency, priority levels, response time trends, and public feedback are often overlooked. As a result, authorities lack a clear understanding of which areas require urgent attention and how resources should be allocated.

This leads to two major operational challenges:

- **Delayed Issue Resolution** – Without real-time tracking and prioritization, complaints may remain unattended for long periods, allowing problems to escalate.
- **Inefficient Resource Utilization** – Poor visibility into complaint patterns results in uneven deployment of sanitation workers and maintenance teams.

Another significant drawback is the lack of integration between complaint reporting and administrative action. In many cases, complaint registration systems operate separately from internal municipal workflows. This disconnect slows down decision-making, causes delays in task assignment, and reduces responsiveness when sudden increases in civic issues occur.

Overall, existing civic issue management systems function in a largely reactive manner. They respond to problems only after citizens report them repeatedly, rather than predicting recurring issues or identifying high-risk zones in advance. This makes it difficult for city authorities to maintain consistent cleanliness standards and deliver timely public services.

2.2 PROPOSED SYSTEM

The proposed Clean Street system transforms traditional civic issue management from a manual and reactive process into a smart, data-driven, and proactive solution. Instead of relying on isolated complaint registers or basic reporting portals, the system integrates modern web technologies, analytics, and intelligent processing to enable efficient reporting, monitoring, and resolution of civic issues.

At the core of the proposed system is a centralized digital platform that captures civic complaints along with essential contextual information such as geographic location, images, issue category, and priority level. By analyzing historical complaint data, location-based patterns, response timelines, and recurrence frequency, the system provides meaningful insights into urban cleanliness and infrastructure problems. This analytical approach enables authorities to identify high-risk zones, recurring issues, and performance gaps more accurately than traditional methods.

A key strength of the proposed system lies in its seamless integration of issue reporting with administrative action. Complaints submitted by users are automatically routed to the appropriate municipal department or volunteer group based on location and issue type. The system continuously tracks complaint status and updates both users and authorities in real time, ensuring transparency and accountability throughout the resolution process.

Unlike existing systems that operate independently, the proposed solution integrates complaint management with monitoring dashboards and analytics. Administrators can view area-wise trends, response efficiency, and workload distribution through interactive dashboards. This enables informed decision-making, optimized workforce allocation, and timely resolution of critical issues without unnecessary delays or resource wastage.

The Clean Street system is designed to be flexible and scalable, allowing integration with existing municipal workflows and databases. Its real-time data handling capability enables quick adaptation to sudden increases in complaints, seasonal cleanliness challenges, or emergency situations. By shifting from reactive responses to proactive planning, the proposed system helps improve service efficiency, enhance citizen satisfaction, and support cleaner, better-managed urban environments.

2.3 FEASIBILITY STUDY

Before developing the Clean Street civic issue reporting and tracking system, it is essential to evaluate whether the proposed solution is practical, achievable, and beneficial for municipal authorities and citizens. A feasibility study helps assess the technical readiness, economic viability, operational suitability, and implementation timeline of the system to ensure that the effort and resources invested will deliver meaningful outcomes.

Technical Feasibility:

From a technical perspective, the proposed Clean Street system is feasible and well within current technological capabilities. The application can be developed using modern and widely adopted web technologies such as React.js for the frontend and Fast API for the backend, ensuring responsiveness and high performance. MongoDB can be used as the database to store user details, complaints, images, and location data efficiently.

Features such as location-based reporting, image uploads, real-time status updates, and analytics dashboards can be implemented using existing APIs, mapping services, and visualization libraries. Most municipal environments already have the required infrastructure, including internet connectivity, servers or cloud platforms, and basic IT systems, making deployment technically achievable without the need for specialized hardware.

Economic Feasibility:

Economically, the proposed system is cost-effective and offers long-term benefits. While there may be initial expenses related to system development, deployment, and user training, these costs are offset by improved operational efficiency. Faster resolution of civic issues reduces repeated complaints, manual workload, and unnecessary resource usage.

By enabling better monitoring and prioritization, Clean Street helps municipal bodies optimize workforce deployment and reduce maintenance inefficiencies. Improved service delivery also increases citizen satisfaction and trust, which adds indirect value to governance. Over time, the savings in operational costs and improved efficiency justify the initial investment.

Operational Feasibility:

The Clean Street system is designed to integrate smoothly into existing municipal workflows without causing disruption. The user interface is simple and intuitive, allowing citizens, volunteers, and administrators to use the system with minimal training. Since the platform consolidates reporting, tracking, and monitoring into a single system, users do not need to switch between multiple tools.

Automation of complaint routing, status updates, and reporting reduces manual effort and administrative burden. This allows municipal staff to focus on decision-making and field operations rather than data entry and tracking, resulting in faster responses and improved service quality.

Schedule Feasibility:

The proposed system follows a modular development approach, making it suitable for phased implementation. Initial deployment can focus on core features such as user registration, complaint submission, and status tracking. Advanced features like analytics dashboards, community voting, and administrative reports can be added in later stages.

With proper planning and resource allocation, the initial version of Clean Street can be developed and deployed within a few months. This phased rollout allows early benefits while providing flexibility for future enhancements and scalability.

2.4 TOOLS AND TECHNOLOGIES

Frontend Technologies

- **React.js** – Used to build a dynamic and component-based user interface for reporting and tracking civic issues.
- **React Router DOM** – Used for client-side routing and smooth navigation between application pages.
- **JavaScript (ES6)** – Used to implement frontend logic and interactive functionalities.
- **CSS** – Used for styling the application and designing responsive layouts.
- **Axios** – Used for handling HTTP requests between the frontend and backend services.
- **Open Layers (OL)** – Used for map integration to enable location-based civic issue reporting.
- **Bootstrap Icons** – Used to provide icons for navigation and user actions.

Backend Technologies

- **Node.js** – Used as the server-side runtime environment for handling requests.
- **Express.js** – Used to develop RESTful APIs and manage backend routing.
- **JavaScript** – Used for implementing backend logic and server-side operations.
- **Multer** – Used for handling image uploads related to civic issue reports.
- **Nodemailer** – Used to send email notifications for user registration and complaint updates.
- **Environment Variables (.env)** – Used to securely store configuration and sensitive data.

Database

- **MongoDB** – Used to store user data, civic complaints, images, votes, comments, and administrative logs.

Development Tools

- **Visual Studio Code** – Used as the primary development environment.
- **Git & GitHub** – Used for version control and source code management.

2.5 HARDWARE AND SOFTWARE REQUIREMENTS

- RAM: 8.00 GB
- Storage: 20 GB free disk space (for project files, node modules, images, and database data)
- Operating System: Windows 11 (64-bit)

Software requirements:

- Operating System: Windows 11 (64-bit)
- **Programming and backend:** For this project, Node.js is used as the server-side runtime environment and Express.js is used as the backend framework for handling APIs and server logic.
- **Libraries and Packages:**
 - o Express.js: Used to build RESTful APIs and manage backend routing.
 - o Multer: Used for handling image uploads related to civic issue reports.
 - o Node mailer: Used to send email notifications for user registration and complaint updates.
- **Frontend:** React.js is used for building the user interface along with JavaScript (ES6) and CSS for interactivity and styling.
- **Database:** MongoDB is used to store user details, civic complaints, images, votes, comments, and administrative logs.

CHAPTER 3

SOFTWARE REQUIREMENTS SPECIFICATION

The purpose of the Clean Street system is to provide a smart, reliable, and user-friendly digital platform for reporting, tracking, and managing civic issues in urban areas. The system aims to replace manual and fragmented complaint-handling processes with a centralized, transparent, and data-driven solution that improves coordination between citizens and municipal authorities.

By enabling citizens to report issues such as garbage accumulation, potholes, water leakage, and malfunctioning streetlights along with location and image evidence, the platform helps authorities respond more efficiently. The system also supports real-time monitoring and community participation, allowing issues to be prioritized based on urgency and public feedback.

Through structured data collection and analytics, the Clean Street system helps municipal bodies:

- Reduce delays in addressing civic problems.
- Improve allocation of manpower and resources.
- Increase transparency and accountability in issue resolution.
- Enhance citizen satisfaction and trust.

Overall, the goal of the system is to simplify civic issue management, improve operational efficiency, and support cleaner, well-managed urban environments through technology-driven decision-making.

3.1 Functional Requirements

The Clean Street system provides a set of core features that directly support effective civic issue reporting, tracking, and management. These features are designed to improve communication between citizens and municipal authorities and ensure timely resolution of civic problems.

1. User Registration and Login

- Users shall be able to register and log in securely using valid credentials.
- The system shall support role-based access for citizens, administrators, and volunteers.

2. Civic Issue Reporting

- Users shall be able to report civic issues by providing details such as issue title, description, and category.
- The system shall allow users to upload images as evidence of the reported issue.
- Users shall be able to select the issue location using OpenStreetMap for accurate geographic tagging.

3. Automated Complaint Storage and Processing

- Submitted complaints shall be automatically stored in a centralized database.
- The system shall organize complaint data based on issue type, location, priority level, and time of submission.

4. Real-Time Complaint Tracking

- Users shall be able to track the current status of their complaints in real time.
- Complaint statuses such as submitted, in progress, and resolved shall be visible to both users and administrators.

5. Community Participation Features

- Users shall be able to vote on reported issues to highlight high-priority problems.
- The system shall allow users to add comments to provide feedback or additional information on complaints.

6. Administrative Management and Assignment

- Administrators shall be able to view, assign, and update complaints through an admin dashboard.

- The system shall enable administrators to monitor complaint resolution progress and user activities.

7. Reporting and Data Export

- Administrators shall be able to generate reports related to complaints and system usage.
- The system shall support exporting reports for record-keeping and analysis purposes.

8. System Integration Support

- The system shall support API-based integration to enable future connectivity with other municipal or civic management systems.

3.2 NON-FUNCTIONAL REQUIREMENTS

In addition to the core functional features, the Clean Street system must satisfy several non-functional requirements to ensure reliability, performance efficiency, security, and overall user satisfaction. These requirements define how well the system performs under different conditions.

1. Performance

- The system shall respond to user actions such as complaint submission, status updates, and dashboard loading within an acceptable time limit.
- Complaint data retrieval and status updates shall be processed efficiently even when a large number of users access the system simultaneously.

2. Usability

- The user interface shall be simple, intuitive, and easy to understand, requiring minimal training for users.
- Clear labels, icons, alerts, and guided workflows shall assist users in reporting and tracking civic issues.
- The system shall be accessible to users with basic computer and internet knowledge.

3. Security

- All communication between the client and server shall be secured using SSL/TLS encryption.
- User authentication and authorization shall be enforced to prevent unauthorized access.
- Sensitive data such as user credentials and complaint details shall be securely stored and protected against misuse.

4. Scalability

- The system shall support multiple concurrent users without degradation in performance.
- The architecture shall allow future expansion to handle increased complaint volume, additional users, and new features.

5. Reliability

- The system shall remain operational with minimal downtime.
- Proper error handling and validation mechanisms shall be implemented to ensure consistent and reliable system behavior.

CHAPTER 4

SYSTEM DESIGN

The Clean Street system is a smart, web-based civic engagement platform designed to assist citizens and municipal authorities in reporting, tracking, and managing public issues efficiently. It acts as a digital coordination system that captures real-world civic problems, organizes them systematically, and supports informed decision-making instead of relying on manual follow-ups or fragmented communication.

The system is developed using a modular architecture in which each component performs a specific function while remaining seamlessly connected with other modules. This modular design ensures that complaint reporting, location tracking, administrative management, and analytics operate independently but function together as a unified system. Such an approach makes the platform reliable for current use and flexible enough to accommodate future enhancements.

At the core of the system, a Node.js and Express.js-based backend handles all server-side operations, including user authentication, complaint processing, image uploads, and status updates. Complaints submitted by users are securely stored in a MongoDB database along with relevant details such as images, geographic coordinates, timestamps, and user information. Backend services also manage role-based access for citizens, administrators, and volunteers.

Before complaints are made visible on dashboards, the data undergoes validation and processing to ensure accuracy and consistency. Location details are captured using OpenStreetMap through OpenLayers, enabling precise geographic tagging of issues. This allows administrators to identify area-wise problem patterns and assign tasks effectively.

The frontend, developed using React.js, provides a clean and intuitive interface that allows users to report issues, track complaint status, and interact with the system without technical complexity. Interactive dashboards display complaint trends, issue categories, and resolution progress in a user-friendly manner. Notifications and alerts guide users through each action and keep them informed about updates.

4.1 SYSTEM PERSPECTIVE

This system architecture illustrates how the Clean Street civic issue reporting and tracking application operates in a structured and coordinated manner. The process begins when the **User** interacts with the system through the **Front End**, which provides interfaces for registration, login, complaint submission, complaint tracking, and community participation.

The **Front End**, developed using React.js, acts as the interaction layer between the user and the system. Users submit civic issue details such as issue description, images, and geographic location through this interface. The frontend then forwards the collected data to the **Backend** for further processing.

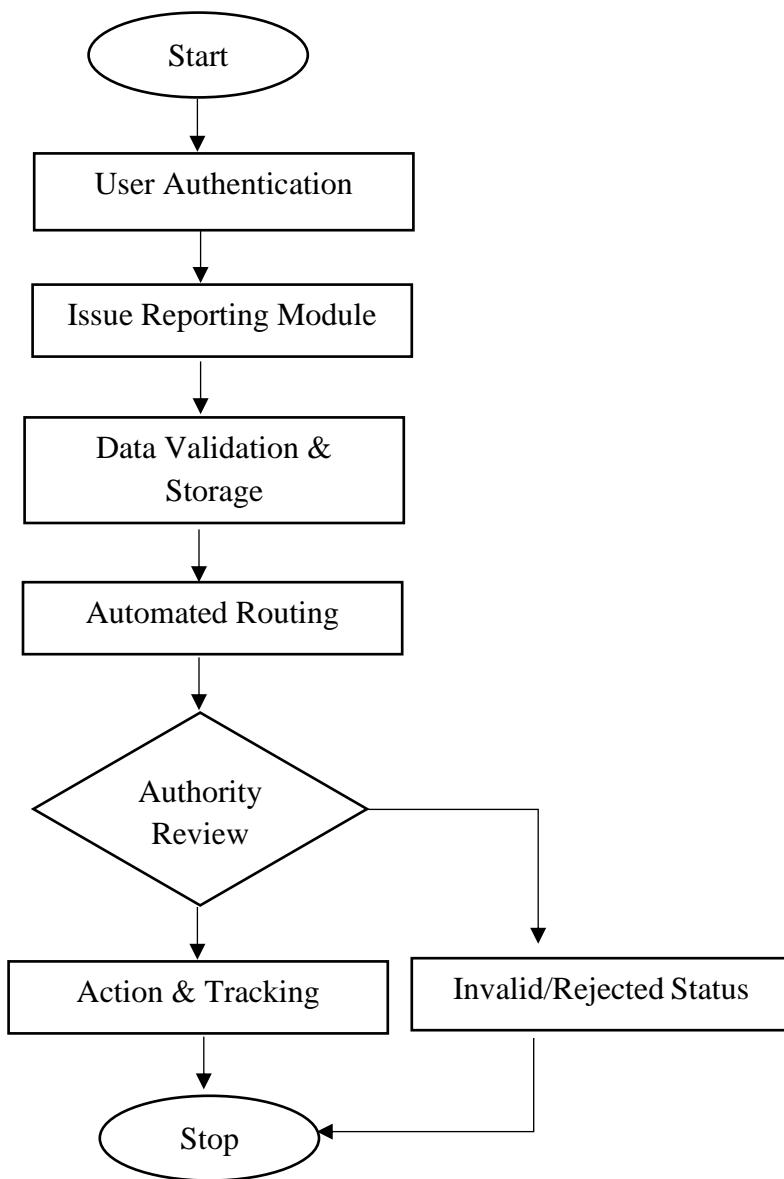
The **Backend**, implemented using Node.js and Express.js, serves as the core processing unit of the system. It validates incoming data, manages user authentication, processes image uploads, and handles complaint-related operations. Location details provided by users are processed using **OpenStreetMap through OpenLayers** to ensure accurate geographic tagging of complaints.

Within the backend, the **Complaint Processing Module** manages tasks such as storing complaints in the database, assigning issues based on location and category, updating complaint status, and managing administrative actions. All complaint data, user details, votes, comments, and logs are stored securely in the **MongoDB Database**.

Once the backend completes processing, the updated complaint status and relevant information are sent back to the **Front End**, where the results are displayed to users in a clear and understandable format. Users can view real-time updates on their complaints, while administrators can access dashboards and reports to monitor system performance and issue resolution progress.

Thus, the system architecture demonstrates how Clean Street integrates user interaction, backend processing, location services, and data storage into a unified platform for efficient civic issue management.

4.2 FLOW CHART

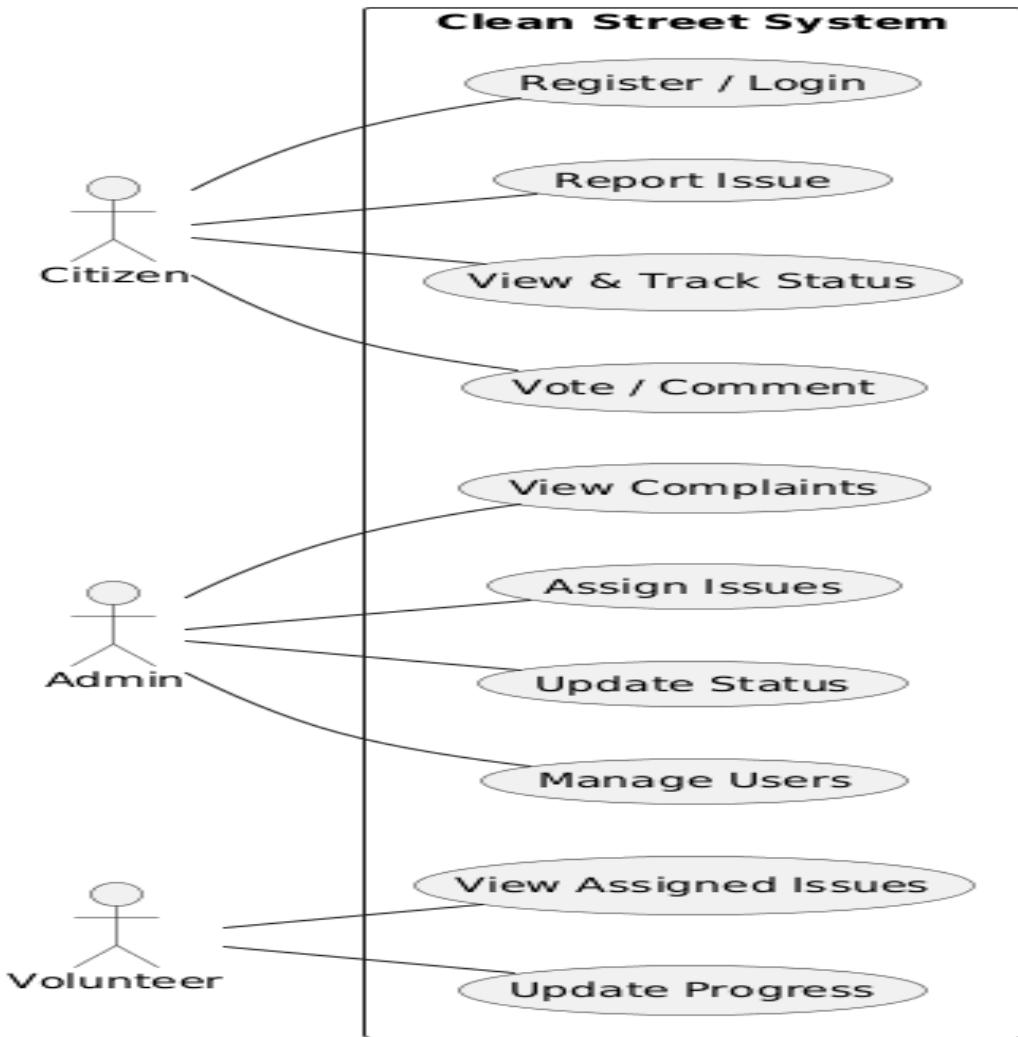


The flow chart of the Civic Issue Reporting System illustrates the complete workflow involved in reporting, tracking, and resolving civic problems within an urban environment. The process begins with user authentication, where citizens or administrators log into the system. After successful login, users can report civic issues by selecting the issue category, entering location details, and uploading relevant images or descriptions.

CHAPTER 5

DETAILED DESIGN

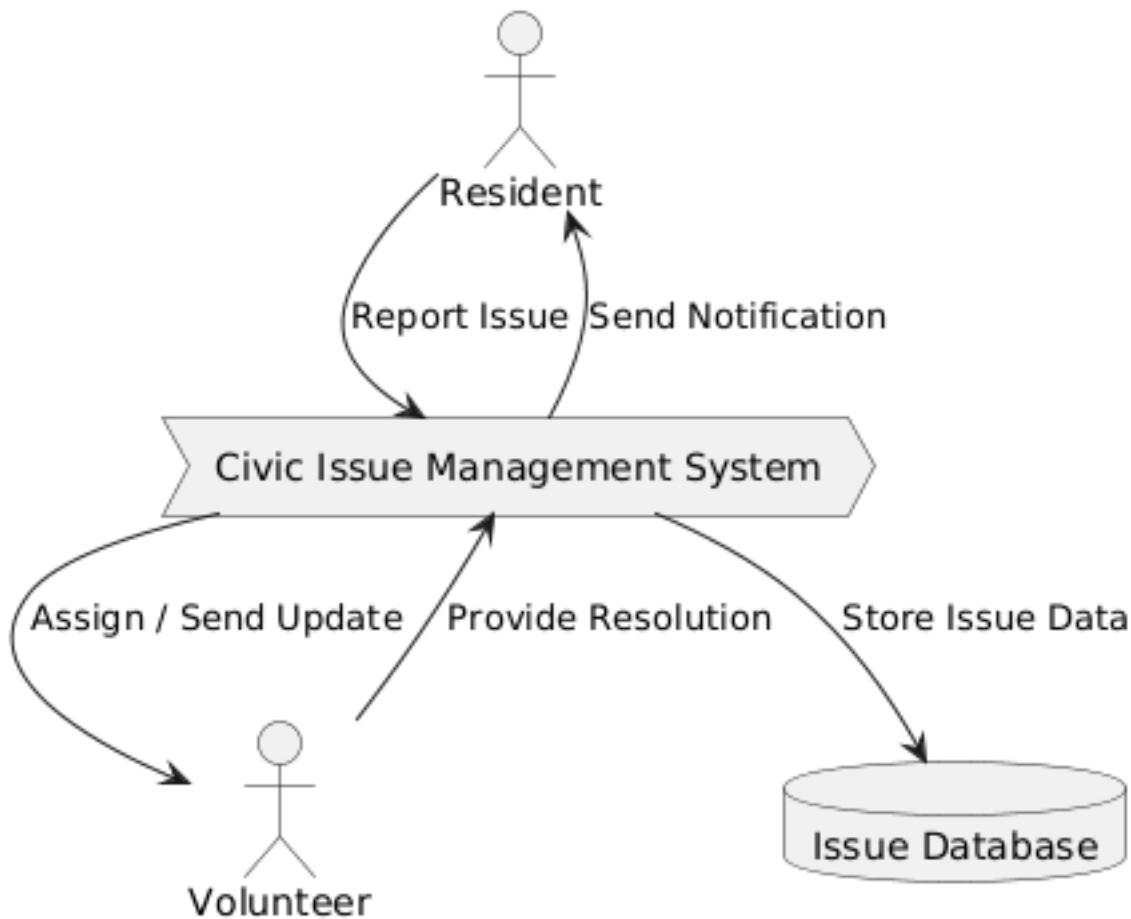
5.1 USE CASE



The use case diagram illustrates the interaction between different actors and the Clean Street system. The primary actors are Citizen, Admin, and Volunteer. Citizens use the system to register or log in, report civic issues, track complaint status, and participate through voting and comments. Administrators manage the system by viewing complaints, assigning issues, updating status, and managing users. Volunteers access assigned issues and update progress after taking action. This diagram highlights the coordinated workflow that supports efficient reporting, monitoring, and resolution of civic issues.

5.2 DATA FLOW DIAGRAM

Level 0 DFD - Civic Issue Reporting System



The Level 0 Data Flow Diagram illustrates the overall working of the Civic Issue Reporting System and shows how information moves between users and the system. A resident reports a civic issue to the Civic Issue Management System, which processes the request and sends notifications back to the resident regarding status updates. The system stores all complaint details in the Issue Database for record keeping and tracking. Volunteers interact with the system by receiving assigned issues, updating progress, and providing resolution details. Through this flow, the system ensures efficient issue reporting, assignment, monitoring, and resolution while maintaining centralized data storage.

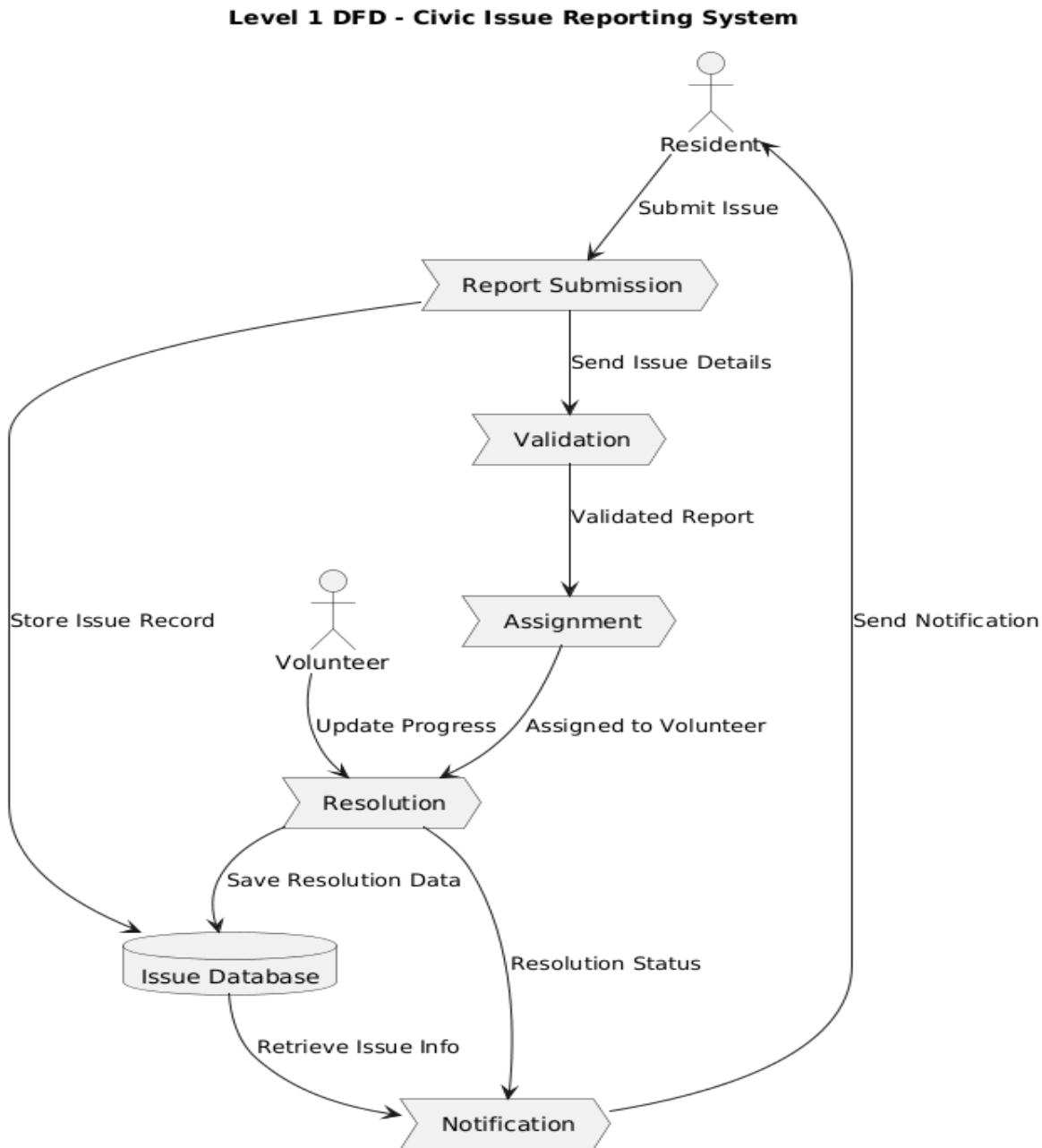
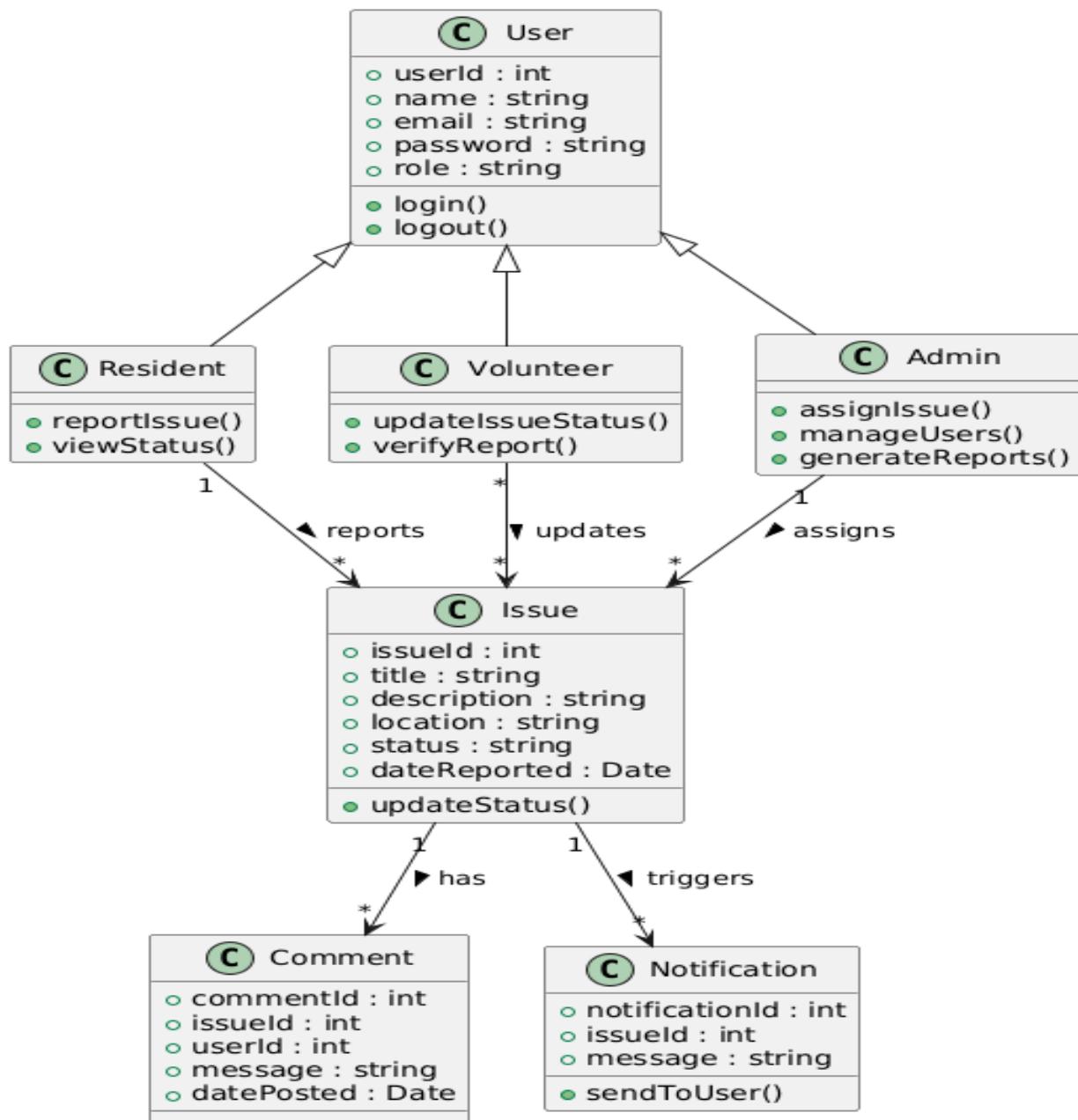


Figure 5.2.1:Level 1 DFD

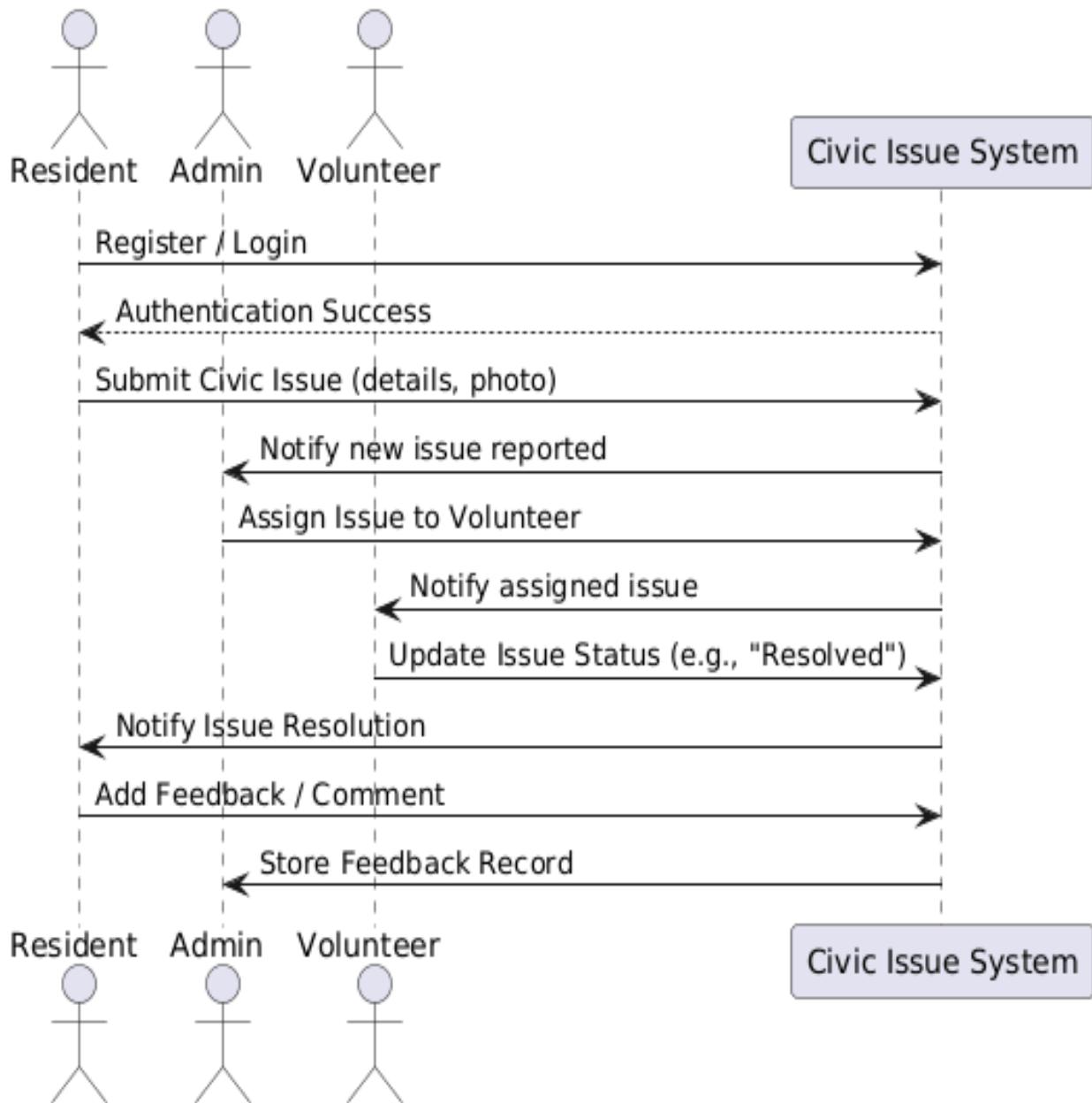
The diagram shows how a resident submits a civic issue, which is validated and assigned to a volunteer. The volunteer resolves the issue and updates the database. Finally, the system sends a notification to the resident about the issue status.

5.3 CLASS DIAGRAM



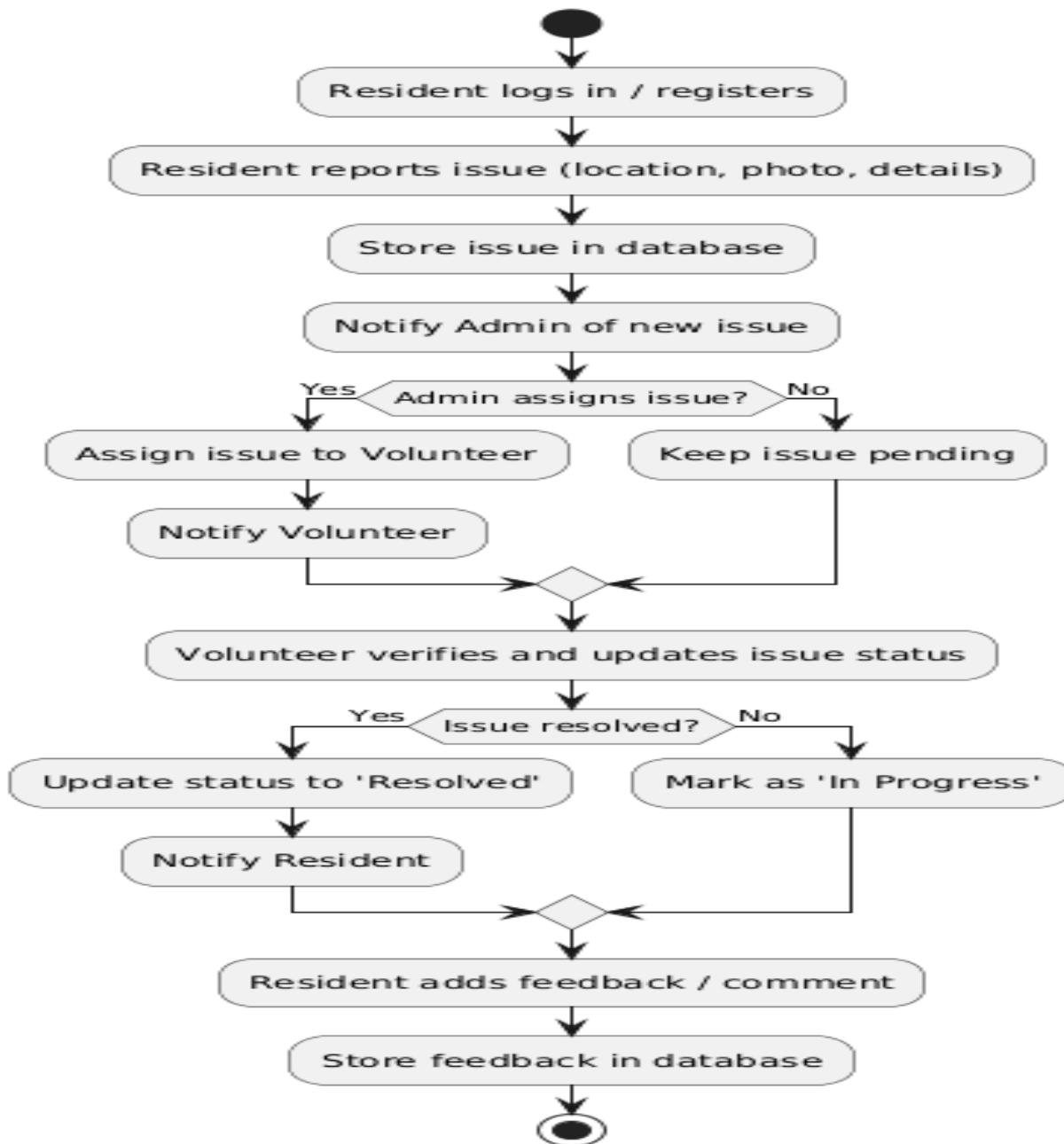
The diagram shows a Civic Issue Reporting System where users (Residents, Volunteers, and Admins) interact with reported issues. Residents report issues, Volunteers update their status, and Admins assign and manage them. Each issue can have comments and notifications for updates.

5.4 SEQUENCE DIAGRAM



The sequence diagram shows how a resident reports a civic issue to the Clean Street system. After login, the issue is submitted and the admin is notified. The admin assigns the issue to a volunteer, who updates the status after resolution. The system then informs the resident, who can provide feedback.

5.5 ACTIVITY DIAGRAM



The diagram shows how a resident reports a civic issue through the system. The issue is stored and the admin is notified, who may assign it to a volunteer or keep it pending. The volunteer updates the issue status as *In Progress* or *Resolved*. Once resolved, the resident is notified and can submit feedback, which is saved in the system.

CHAPTER 6

IMPLEMENTATION

6.1 CODE SNIPPETS

```
import React, { useEffect, useState } from "react";
import { Link } from "react-router-dom";
import { useAuth } from "../context/AuthContext";
import "./Home.css";
import axios from "axios";
export default function Home() {
  const { user } = useAuth();
  const [loading, setLoading] = useState(true);
  const [stats, setStats] = useState({
    totalUsers: 0,
    totalIssues: 0,
    postalCodes: 0,
  });
  const [recentlySolved, setRecentlySolved] = useState([]);
  useEffect(() => {
    const fetchData = async () => {
      const usersRes = await axios.get(
        "http://localhost:5000/api/auth/public/users-count"
      );
      setStats({
        ...stats,
        totalUsers: usersRes.data.count,
      });
      setRecentlySolved([
        ...recentlySolved,
        usersRes.data.user,
      ]);
    };
    fetchData();
  }, []);
  return (
    <div>
      <h1>Welcome to Civic Issue Reporting and Tracking</h1>
      <p>User:<br/>{user}</p>
      <table border="1">
        <thead>
          <tr>
            <th>Category</th>
            <th>Count</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>Total Users</td>
            <td>{stats.totalUsers}</td>
          </tr>
          <tr>
            <td>Total Issues</td>
            <td>{stats.totalIssues}</td>
          </tr>
          <tr>
            <td>Postal Codes</td>
            <td>{stats.postalCodes}</td>
          </tr>
        </tbody>
      </table>
      <table border="1">
        <thead>
          <tr>
            <th>Issue ID</th>
            <th>Title</th>
            <th>Status</th>
          </tr>
        </thead>
        <tbody>
          <tr>
            <td>1</td>
            <td>Water Supply Problem</td>
            <td>Solved</td>
          </tr>
          <tr>
            <td>2</td>
            <td>Road Construction</td>
            <td>In Progress</td>
          </tr>
          <tr>
            <td>3</td>
            <td>Garbage Collection</td>
            <td>Solved</td>
          </tr>
        </tbody>
      </table>
    </div>
  );
}
```

```
);

const totalUsers = usersRes.data.count || 0;

try {

let totalUsers = 0;

try {

const res = await axios.get(

"http://localhost:5000/api/auth/allusers"

);

totalUsers = Array.isArray(res.data) ? res.data.length : 0;

} catch (err) {

console.warn("Users endpoint not found");

}

const issuesRes = await axios.get(

"http://localhost:5000/api/issues/public"

);

const all = issuesRes.data.issues || [];

const seen = new Set();

const uniqueIssues = [];

for (const issue of all) {

if (issue?._id && !seen.has(issue._id)) {

seen.add(issue._id);

uniqueIssues.push(issue);

}

}
```

```
}

const totalIssues = uniqueIssues.length;

const postalRes = await axios.get(
  "http://localhost:5000/api/auth/public/postal-codes"
);

const postalCodes = postalRes.data.postalCodes.length;

const resolvedIssues = uniqueIssues

.filter((i) => (i.status || "").toLowerCase() === "resolved")

.sort((a, b) => new Date(b.createdAt) - new Date(a.createdAt))

.slice(0, 3);

setRecentlySolved(
  resolvedIssues.map((i) => ({
    id: i._id,
    title: i.title,
    date: new Date(i.createdAt).toLocaleDateString(),
  }))
);

// Set Stats

setStats({
  totalUsers,
  totalIssues,
  postalCodes,
});
```

```
    } catch (err) {  
  
      console.error("Error loading homepage data:", err);  
  
    } finally {  
  
      setLoading(false);  
  
    }  
  
  };  
  
  fetchData();  
  
}, []);  
  
return (  
  
  <div className="landing-container">  
  
    {/* SECTION 1: HERO */}  
  
    <section className="home-section hero">  
  
      <div className="hero-overlay"></div>  
  
      <div className="hero-content">  
  
        <div className="hero-left">  
  
          <h1>  
  
            Clean<span className="highlight">Street</span>  
  
          </h1>  
  
          <p className="caption">Report. Track. Resolve.</p>  
  
          <p className="brief">  
  
            A community-driven platform to build cleaner, smarter  
  
            neighborhoods. Report issues like potholes and garbage in  
  
            real-time and watch them get resolved.  
  
        </div>  
  
      </div>  
  
    </section>  
  
  </div>  
  
);
```

```
</p>

<div className="hero-actions">

  { !user ? (
    <>
      <Link to="/register" className="btn primary-btn">
        Get Started
      </Link>
      <Link to="/login" className="btn outline-btn">
        Login
      </Link>
    </>
  ) : (
    <>
      <Link to="/report-issue" className="btn primary-btn">
        Report Issue
      </Link>
      <Link to="/complaints" className="btn outline-btn">
        View Complaints
      </Link>
    </>
  )
}

</div>
</div>
```

```
</div>

<div className="scroll-indicator">

  <span>Scroll to Explore</span>

  <div className="arrow">↓</div>

</div>

</section>

/* SECTION 2: LIVE STATISTICS */

<section className="home-section stat-section">

  <div className="section-headers">

    <h2>Our Impact</h2>

    <p>Real-time data empowering our city</p>

  </div>

  <div className="stat-grids">

    <div className="stats-card">

      <div className="icons">

        <i class="bi bi-people-fill"></i>

      </div>

      <h3>{loading ? "..." : stats.totalUsers }+</h3>

      <p>Active Citizens</p>

    </div>

    <div className="stats-card featured">

      <div className="icons">

        <i class="bi bi-megaphone-fill"></i>

      </div>
```

```
</div>

<h3>{loading ? "..." : stats.totalIssues}+</h3>

<p>Issues Reported</p>

</div>

<div className="stats-card">

  <div className="icons">

    <i class="bi bi-geo-fill"></i>

  </div>

  <h3>{loading ? "..." : stats.postalCodes}</h3>

  <p>Postal Zones</p>

</div>

</div>

</section>

/* SECTION 3: RECENT ACTIVITY */

<section className="home-section recent-section">

  <div className="content-wrapper">

    <h2>Recently Solved</h2>

    <p className="subtitle">See what's happening in your neighborhood</p>

    <div className="recent-list">

      {recentlySolved.map((item) => (

        <div key={item.id} className="recent-item">

          <div className="status-indicator resolved">✓</div>
```

```
<div className="recent-info">

  <h4>{item.title}</h4>

  <span className="date">{item.date}</span>

</div>

</div>

))}

</div>

</div>

<footer className="footer">

  <p>

    © {new Date().getFullYear()} CleanStreet. Together for a cleaner

    tomorrow.

  </p>

</footer>

</section>

</div>

);

}
```

6.2 SCREENSHOTS

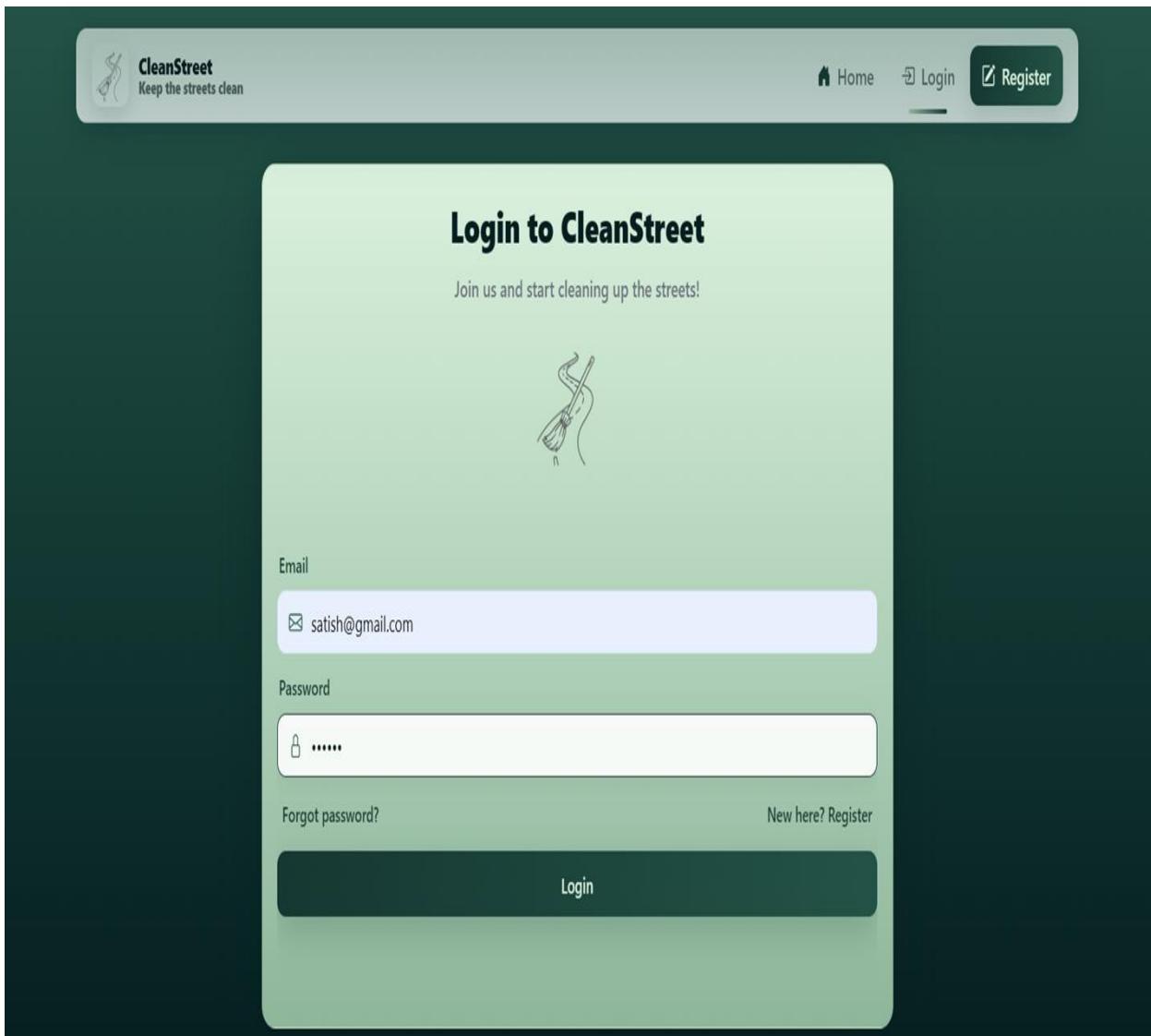
STEP-1

- The user opens the CleanStreet application and navigates to the Create Account (Register) page to begin the registration process.

The screenshot shows the 'Create Account' page of the CleanStreet mobile application. At the top, there is a logo with a broom icon and the text 'CleanStreet Keep the streets clean'. To the right are links for 'Home', 'Login', and a prominent 'Register' button. The main form is titled 'Create Account' with the sub-instruction 'Join us and start cleaning up the streets!'. It contains fields for 'Full Name' (with placeholder '@ alex'), 'Username' (with placeholder '@ alex12'), 'Email' (with placeholder 'alex@gmail.com'), 'Phone (Optional)' (with placeholder '9234552523'), 'Location' (with placeholder '3rd Cross Road, Banashankari 1st Sta...'), and 'Password' (with placeholder '.....'). Below the form is a link 'Already have an account? Login' and a large dark green 'Register' button at the bottom.

STEP-2

- The user navigates to the **Login** page of the CleanStreet application and enters their registered **email ID and password** to access the system



STEP-3

- After logging in, the user reports a civic issue by entering issue details such as title, type, priority, description, and selecting the location on the map before submitting the report

CleanStreet
Keep the streets clean

Home Dashboard Report Issue View Complaints Profile Logout

Report Civic Issue

Help make your community cleaner and safer

Issue Details

ISSUE TITLE * Streetlight is not working

ISSUE TYPE * Broken Streetlight

PRIORITY LEVEL * Medium

LANDMARK (OPTIONAL) Sudhamnagar

ADDRESS (CLICK ON MAP) 9th Cross Road, Thyagarajnagar, Bengaluru West City Corporation, Bengaluru, Bangalore South

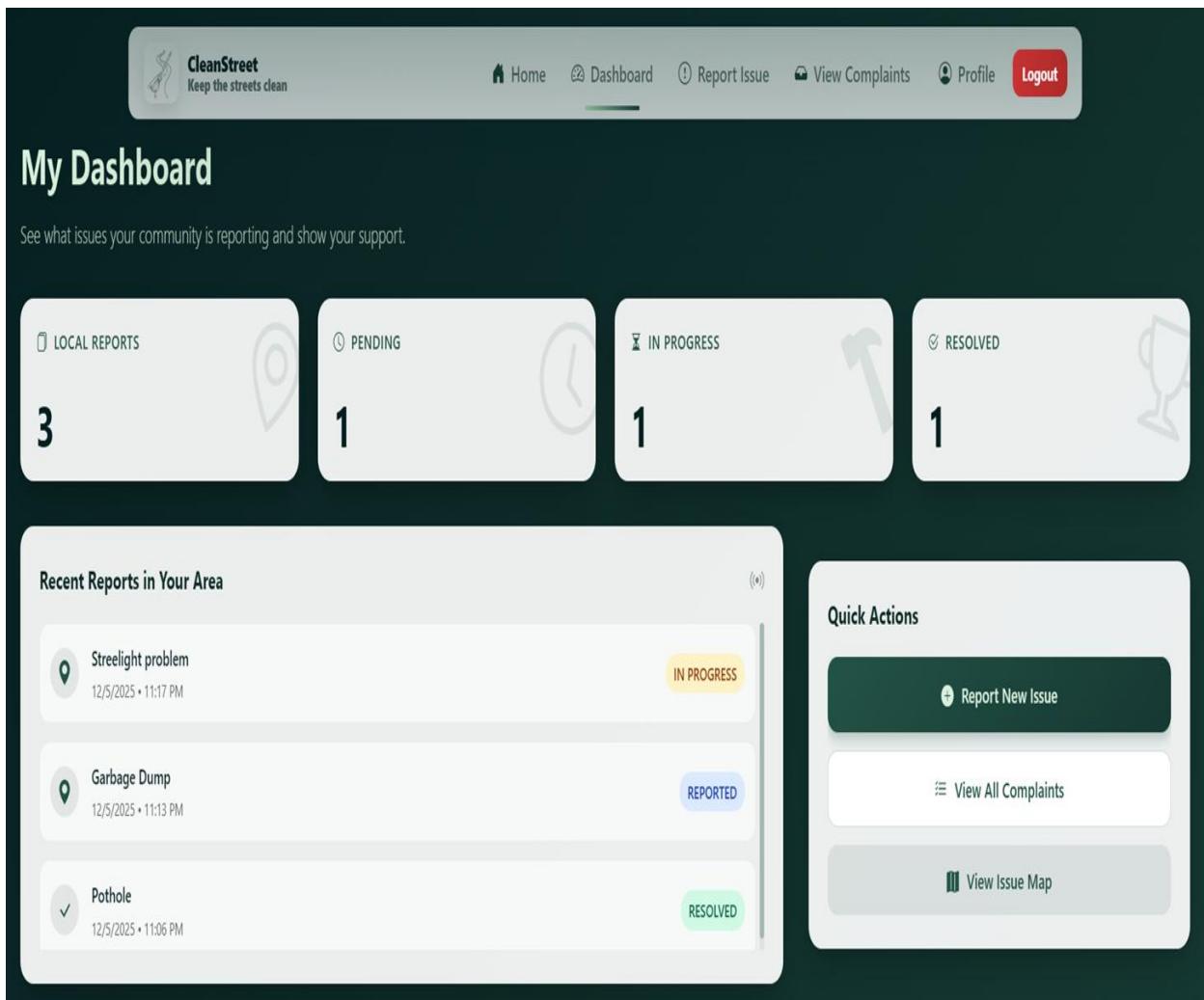
DESCRIPTION *

Select Location on Map

The map displays a street layout with several roads labeled: 3rd, 5th, 10th, and 12th Cross Roads; 4th, 9th, and 10th Main Roads; and 4th, 5th, 6th, 7th, 8th, 9th, and 10th Thirthanahalli. Landmarks visible include the 'Abhaya Lakshmi Narasimha Temple' and 'Channammana Kere Grounds'. A green marker indicates the reported issue is located on 'Vidya Peetha Main Road' near '1st Cross Road'.

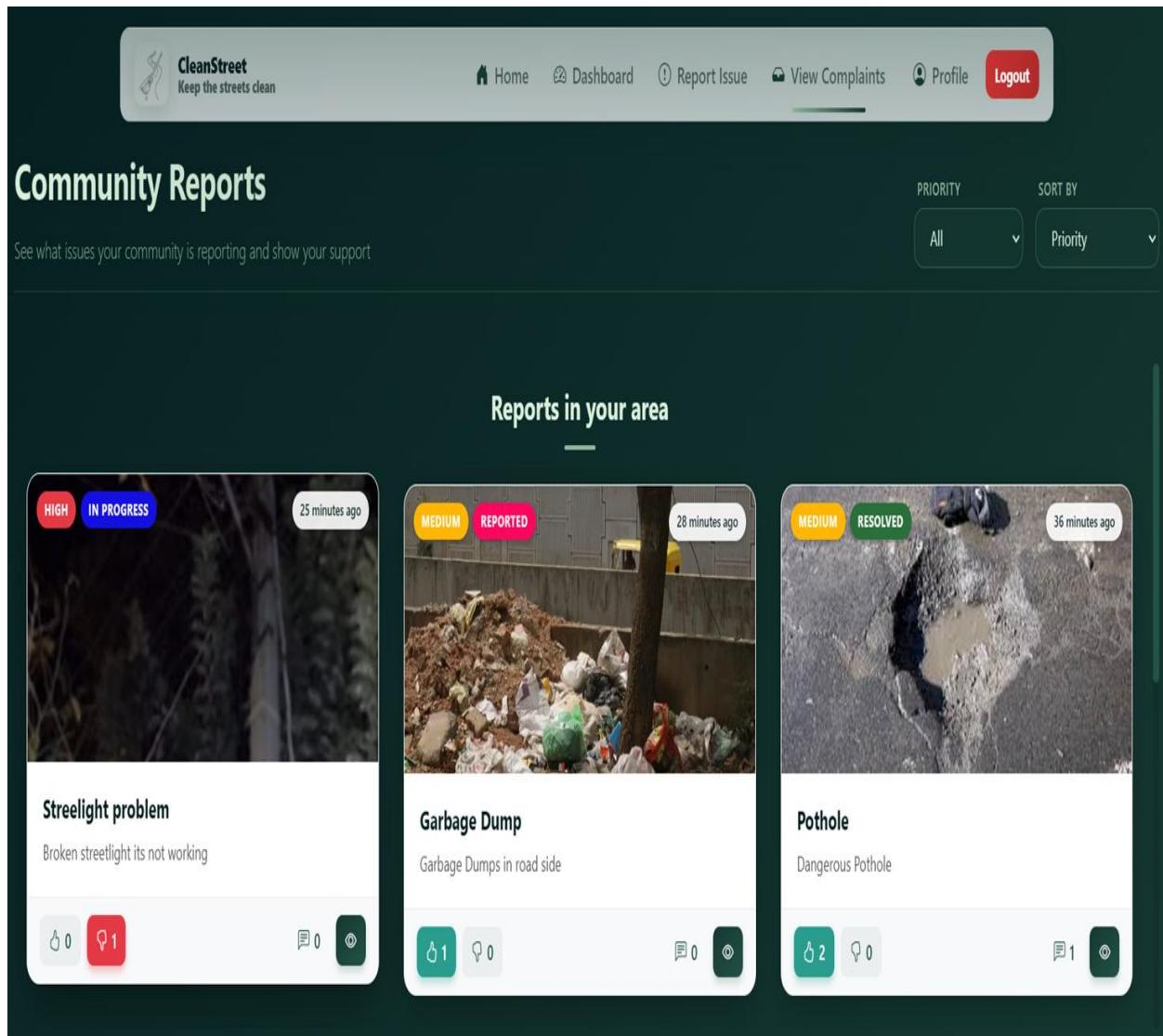
STEP-4

- After submitting an issue, the user views the **Dashboard** to track reported issues, check their status (pending, in progress, resolved), and access quick actions such as reporting a new issue or viewing complaints.



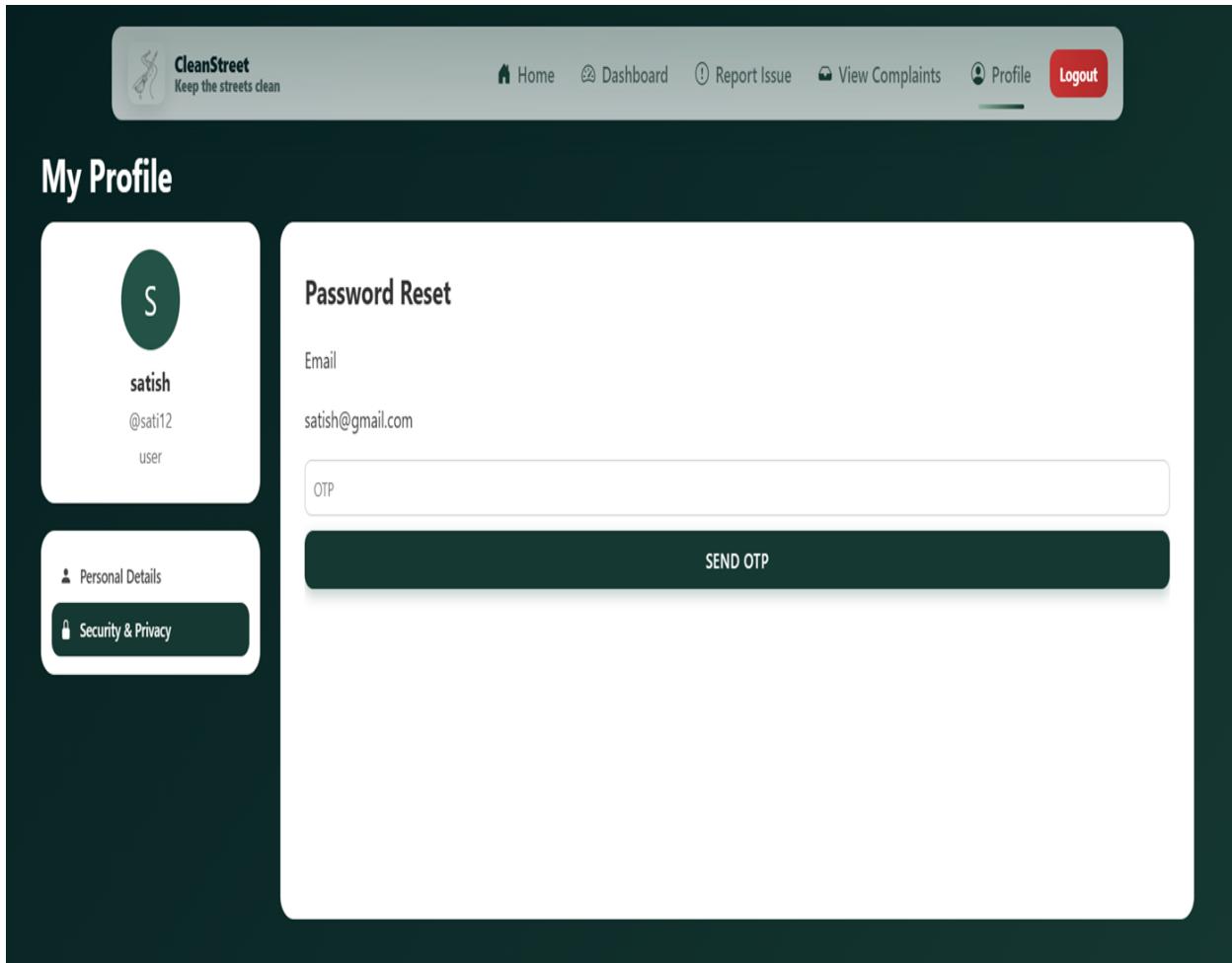
STEP-5

- The user views **Community Reports** to see issues reported in their area, filter them by priority or status, and track updates such as reported, in-progress, or resolved issues.



STEP-6

- The user accesses the **Profile** section to view personal details and perform security actions such as **password reset** by requesting and verifying an OTP.



STEP -7

- The **Admin/Region Manager** accesses the **Region Manager** panel to view users in a specific region, check their status, search users, and perform actions such as managing or removing user accounts.

The screenshot shows the 'Region Manager' section of the CleanStreet application. At the top, there's a navigation bar with links for Home, Overview, Manage Complaints, Users (which is currently selected), Reports, Profile, and Logout. Below the navigation, it says 'Managing Region: 560050'. A search bar is present with the placeholder 'Search by name or email...'. The main area displays a table of user details:

USER DETAILS	POSTAL CODE	STATUS	ACTIONS
arthik arthik@gmail.com	560050	• ACTIVE	
alex alex@gmail.com	560050	• ACTIVE	
satish satish@gmail.com	560050	• ACTIVE	

At the bottom of the table, it says 'End of list'.

CHAPTER 7

SOFTWARE TESTING

Software testing is a critical phase in the development of the Clean Street civic issue reporting and tracking system. Since the system is used to report, monitor, and manage public issues, any malfunction can affect user trust and delay issue resolution. Testing is therefore conducted systematically to ensure the application is reliable, secure, and user-friendly before deployment.

Objectives of Testing

- **Functional Accuracy:** Ensuring that features such as user registration, complaint submission, image upload, location selection, and status tracking work correctly.
- **Data Validation:** Verifying that incorrect or incomplete inputs are handled properly without system failure.
- **Usability:** Confirming that users can easily navigate the interface and perform actions without confusion.
- **Security:** Ensuring secure handling of user data and protected access to administrative functions.
- **System Stability:** Checking that the system performs consistently under normal usage conditions.

2. Types of Testing Performed

a) Unit Testing

Each module of the Clean Street system is tested independently to ensure it functions correctly. This includes verifying whether the user authentication module handles login and registration properly, checking if the complaint submission module correctly accepts issue details and images, and ensuring that location selection using the map works as expected.

b) Integration Testing

Integration testing is performed to ensure smooth interaction between different modules of the system. This includes verifying that complaints submitted through the user interface are correctly processed by the backend, stored in the database, and reflected accurately in the admin dashboard. It also ensures that status updates made by administrators are properly displayed to users.

c) System Testing

The entire Clean Street system is tested end-to-end to simulate real user interaction. This includes user login, complaint submission with image and location, backend processing, status updates by administrators, and viewing the updated complaint status on the dashboard.

d) Performance Testing

Performance testing ensures that the system responds efficiently under normal and peak usage conditions. It verifies that multiple users can submit and track complaints simultaneously without delays and that dashboards load smoothly even with a large number of complaints.

e) Security Testing

Security testing is conducted to protect user data and system access. This includes verifying secure login authentication, ensuring that data transfer between client and server is encrypted, and confirming that only authorized users can access administrative features.

f) Usability Testing

Usability testing ensures that the application is easy to use for all users. The interface is checked for clarity, ease of navigation, and simplicity in reporting issues, uploading images, selecting locations, and tracking complaint status.

g) Validation Testing

Validation testing confirms that the system functions according to requirements. It checks whether submitted complaints are correctly stored, status updates are accurately reflected, and reports display reliable and consistent information.

Test cases:

Test ID	Test Case Description	Input	Expected Output	Actual Result	Status (Pass/Fail)
TC-01	User Login Validation	Invalid login credentials	System rejects login and shows error message	Error message displayed	Pass
TC-02	Complaint Submission Validation	Form with missing fields	System prompts user to fill required fields	Validation message shown	Pass
TC-03	Image Upload Validation	Unsupported image format	System rejects file and shows warning	Error message displayed	Pass
TC-04	Location Selection	User selects map location	Location coordinates captured correctly	Location saved successfully	Pass
TC-05	Complaint Storage	Valid complaint submission	Complaint stored with initial status	Complaint stored correctly	Pass
TC-06	Status Update	Admin updates complaint status	Updated status visible to user	Status updated successfully	Pass
TC-07	Usability Test	Non-technical user reports issue	User completes task without confusion	Task completed successfully	Pass

CHAPTER 8

CONCLUSION

In today's rapidly growing urban environments, effective management of civic issues is essential for maintaining cleanliness, safety, and quality of life. The Clean Street system developed in this project addresses this need by providing a centralized, technology-driven platform for reporting, tracking, and resolving public issues in a transparent and efficient manner.

The system replaces traditional manual complaint handling methods with a structured digital approach that enables citizens to report issues using images and location data, while allowing authorities to monitor and resolve complaints in real time. By integrating user participation, administrative control, and location-based tracking, the platform ensures better coordination and faster response to civic problems.

Beyond its functional capabilities, the project emphasizes usability and accessibility. The web-based interface is simple and intuitive, making it easy for non-technical users to report issues and track progress. At the same time, administrators benefit from dashboards and reports that support monitoring and decision-making.

Overall, the Clean Street system demonstrates how modern web technologies can be effectively applied to civic management. By improving transparency, accountability, and community involvement, the project contributes to cleaner streets, improved public services, and smarter urban governance.

In addition, the modular design of the Clean Street system allows for future enhancements without major changes to the existing structure. Features such as mobile application support, integration with smart city platforms, automated priority detection, and advanced analytics can be incorporated to further improve efficiency. This scalability ensures that the system can adapt to growing urban needs and evolving technological advancements, making it a sustainable solution for long-term civic management.

CHAPTER 9

FUTURE ENHANCEMENTS

Although the Clean Street application effectively supports civic issue reporting and monitoring, further improvements can enhance its adaptability and long-term usefulness. As cities expand and public service demands increase, the application can evolve to address emerging challenges more efficiently and proactively.

1. Use of Contextual Public Data

Future versions may utilize contextual information such as climate conditions, public gatherings, and traffic patterns to better anticipate problem-prone areas and time periods, enabling improved preparedness by authorities.

2. Automated Issue Detection

The application can be extended to receive inputs from smart devices and sensors deployed in urban spaces. This would allow certain civic problems to be identified automatically, reducing dependence on manual reporting.

3. Advanced Issue Ranking Mechanism

A rule-based or score-driven mechanism can be introduced to rank reported issues based on urgency, location sensitivity, and citizen feedback, helping officials focus on the most critical cases first.

4. Improved Stakeholder Interaction

Future enhancements can include richer interaction features such as discussion spaces, notification alerts, and structured communication channels to strengthen coordination among users, volunteers, and officials.

APPENDIX A

BIBLIOGRAPHY

Textbooks

1. Sommerville, I. (2016). *Software Engineering* (10th ed.). Pearson Education.
2. Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach* (9th ed.). McGraw-Hill.
3. Elmasri, R., & Navathe, S. B. (2017). *Fundamentals of Database Systems* (7th ed.). Pearson.
4. Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic Information Systems and Science* (4th ed.). Wiley.

Websites

1. OpenStreetMap Contributors. (n.d.). *OpenStreetMap Documentation*. Retrieved from <https://www.openstreetmap.org>
2. React Documentation. (2024). *React – A JavaScript Library for User Interfaces*. Retrieved from <https://react.dev>
3. Node.js Foundation. (2024). *Node.js Official Documentation*. Retrieved from <https://nodejs.org>
4. MongoDB Documentation. (2024). *MongoDB Manual*. Retrieved from <https://www.mongodb.com/docs>
5. Smart Cities Council. (2023). *Technology for Smart and Sustainable Cities*. Retrieved from <https://smartcouncils.com>

References

- [1] I. Sommerville, *Software Engineering*, 10th ed., Pearson Education, 2016.
- [2] R. S. Pressman and B. R. Maxim, *Software Engineering: A Practitioner's Approach*, 9th ed., McGraw-Hill, 2019.
- [3] R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed., Pearson, 2017.
- [4] P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind, *Geographic Information Systems and Science*, 4th ed., Wiley, 2015.
- [5] A. Batty et al., “Smart cities of the future,” *European Physical Journal – Special Topics*, vol. 214, no. 1, pp. 481–518, 2012.
- [6] A. Townsend, *Smart Cities: Big Data, Civic Hackers, and the Quest for a New Utopia*, W. W. Norton & Company, 2013.
- [7] Government of India, *Smart Cities Mission Guidelines*, Ministry of Housing and Urban Affairs, 2023.
- [8] J. Scholl and S. Scholl, “Smart governance: A roadmap for public service digitalization,” *Information Polity*, vol. 19, no. 3–4, pp. 163–176, 2014.
- [9] United Nations Human Settlements Programme (UN-Habitat), *People-Centered Smart Cities*, UN Publications, 2020.
- [10] M. Janssen, Y. Charalabidis, and A. Zuiderwijk, “Benefits, adoption barriers and myths of open data and open government,” *Information Systems Management*, vol. 29, no. 4, pp. 258–268, 2012.

APPENDIX B

USER MANUAL

1. Introduction

Welcome to the **Clean Street – Civic Issue Reporting and Tracking System**.

Clean Street is a web-based application developed to help citizens report public issues such as garbage accumulation, potholes, water leakage, and damaged streetlights in an easy and organized manner. The system allows users to submit complaints with images and location details, track issue status, and participate through voting and comments. Administrators can monitor, assign, and resolve issues efficiently, improving transparency and civic management.

2. System Requirements

Hardware Requirements

- Minimum 4 GB RAM (8 GB recommended)
- Dual-core processor or higher
- At least 2 GB free disk space
- Stable internet connection

Software Requirements

- Web browser (Google Chrome, Mozilla Firefox, or Microsoft Edge)
- Node.js runtime environment (configured by developer)
- MongoDB database (configured by developer)

3. Getting Started

Step 1: Access the Application

- Open the CleanStreet application in a web browser.
- The Home page displays options to log in or register.

Step 2: User Login / Registration

- Existing users enter email and password on the Login page.
- New users create an account using the Register option.
- Successful login redirects the user to the Dashboard.

Step 3: View Dashboard

- Displays summary of reported civic issues.
- Shows counts of pending, in-progress, and resolved complaints.
- Provides quick access to report issues or view complaints.

Step 4: Report a Civic Issue

- Navigate to the Report Issue page.
- Enter issue title, type, priority, and description.
- Upload an image related to the issue.

Step 5: Select Issue Location

- Choose the issue location using the interactive map.
- System captures address and coordinates automatically.

Step 6: Submit Complaint

- Review entered details.
- Submit the complaint.
- System validates data and registers the issue.

Step 7: Track Complaint Status

- Open the View Complaints page.
- Monitor status updates for submitted issues.

Step 8: Profile & Security

- Access Profile page to view user details.
- Reset password using OTP verification if required.

Step 9: Logout

- Click Logout to securely exit the system.

4. Key Features

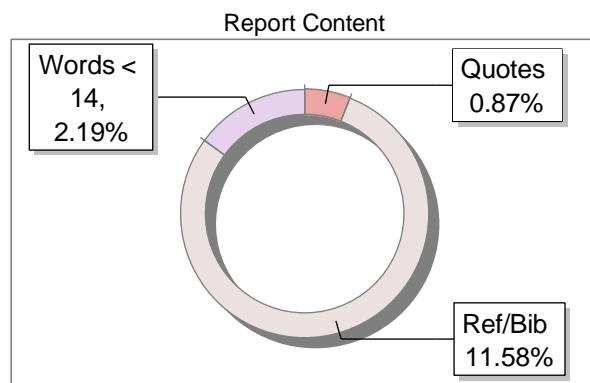
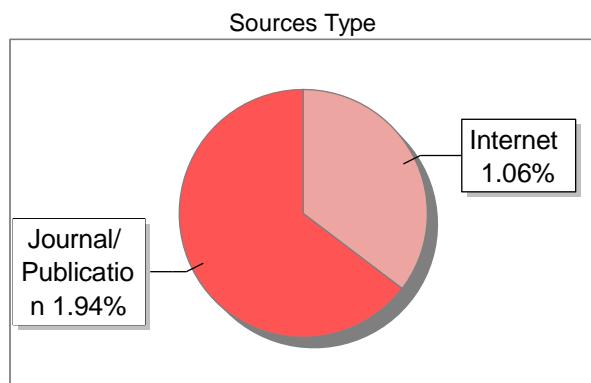
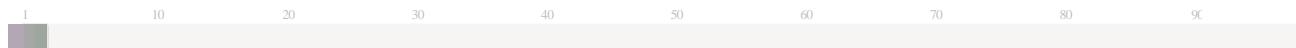
- **Civic Issue Reporting:** Allows users to report public issues with description, image, and location details.
- **Location-Based Mapping:** Uses an interactive map to accurately tag and display issue locations.
- **Real-Time Status Tracking:** Enables users to track complaint progress from submission to resolution.
- **Community Participation:** Supports voting and commenting on issues to highlight priority concerns.
- **Admin Management Panel:** Provides administrators with tools to assign issues, update status, and manage users.
- **Visual Dashboards:** Displays complaint statistics and trends for easy monitoring and decision-making

Submission Information

Author Name	Srijan kumar
Title	Civic issue
Paper/Submission ID	5119629
Submitted by	malleshakls@gmail.com
Submission Date	2026-01-05 13:42:13
Total Pages, Total Words	47, 6996
Document type	Project Work

Result Information

Similarity **3 %**



Exclude Information

Quotes	Excluded
References/Bibliography	Excluded
Source: Excluded < 14 Words	Excluded
Excluded Source	0 %
Excluded Phrases	Excluded

Database Selection

Language	English
Student Papers	Yes
Journals & publishers	Yes
Internet or Web	Yes
Institution Repository	Yes

A Unique QR Code use to View/Download/Share Pdf File





DrillBit Similarity Report

3

SIMILARITY %

5

MATCHED SOURCES

A

GRADE

A-Satisfactory (0-10%)
B-Upgrade (11-40%)
C-Poor (41-60%)
D-Unacceptable (61-100%)

LOCATION	MATCHED DOMAIN	%	SOURCE TYPE
1	www.nhmmp.gov.in	1	Publication
2	translate.google.com	1	Internet Data
3	etd.aau.edu.et	<1	Publication
4	aurangabadmahapalika.org	<1	Publication
5	moam.info	<1	Internet Data

EXCLUDED PHRASES

1 bangalore institute of technology