“CleanSweep : Urban Cleanliness and Monitoring system”

A Minor Project Report Submitted to

Rajiv Gandhi Proudyogiki Vishwavidyalaya

Towards Partial Fulfillment for the Award of

Bachelor of Technology
In
Computer Science & Information Technology

Submitted by:

Yash Patel (0827CI221154)
Umesh Panchal (0827CI221142)
Shreya Verma (0827CI221124)
Raina Dhanyase (0827CI221112)

Guided by:

Nidhi Nigam
Asst. Professor
CSIT Department

Acropolis Institute of Technology & Research, Indore
Jan- June 2025

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Batch 2022-26

DECLARATION

We hereby declare that the work, which is being presented in this project entitled "**Urban Cleanliness and Monitoring System**" in partial fulfillment of the requirements for the award of degree of **Bachelor of Technology in Computer Science and Information Technology**, is authentic record of work carried out by us.

Place: CSIT, Indore

Date:

Signature of Students

Yash Patel (0827CI221154)
Umesh Panchal (0827CI221142)
Shreya Verma (0827CI221124)
Raina Dhanyase (0827CI221112)

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Batch 2022-26

RECOMMENDATION

This is to certify that the work embodied in this project entitled **"Urban Cleanliness and Monitoring System"** submitted by **Yash Patel (0827CI221154) , Umesh Panchal (0827CI221142) , Shreya Verma (0827CI221124) and Raina Dhanyase (0827CI221112)** is a satisfactory account of the bonafide work done under the supervision of **Asst. Prof. Nidhi Nidhi Nigam** , is recommended towards partial fulfillment for the award of the Bachelor of Technology in Computer Science & Information Technology degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

Project Guide :
Asst. Prof. Nidhi Mam

Project Coordinator :
Asst. Prof. Ashwinee Gadwal

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Batch 2022-26

CERTIFICATE

The Project entitled **"Urban Cleanliness and Monitoring System"** submitted by **Yash Patel (0827CI221154) , Umesh Panchal (0827CI221142) , Shreya Verma (0827CI221124) and Raina Dhanyase (0827CI221112)** has been examined and is hereby approved towards partial fulfillment for the award of **Bachelor of Technology** in **Computer Science & Information Technology,** for which it has been submitted. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted

Internal Examiner

Date:

DEPARTMENT OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Batch 2022-26

STUDENT UNDERTAKING

This is to certify that project entitled **"Urban Cleanliness and Monitoring System"** has developed by us under the supervision of **Asst. Prof. Nidhi Nidhi Nigam** . The whole responsibility of work done in this project is ours. The sole intension of this work is only for practical learning and research.

We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work found then we are liable for explanation to this.

Yash Patel	**Umesh Panchal**	**Shreya Verma**	**Raina Dhanyase**
Date:	**Date:**	**Date:**	**Date:**

ACKNOWLEDGEMENT

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

We owe a debt of sincere gratitude, deep sense of reverence and respect to our guide **Nidhi Mam , Asst. Professor** and mentor **Ashwinee Gadwal** , **Asst. Professor** , AITR, Indore for their motivation, sagacious guidance, constant encouragement, vigilant supervision and valuable critical appreciation throughout this project work, which helped us to successfully complete the project on time.

We express profound gratitude and heartfelt thanks to **Prof. (Dr.) Shilpa Bhalerao HOD CSIT**, AITR Indore for her support, suggestion and inspiration for carrying out this project. We would be failing in our duty if do not acknowledge the support and guidance received from **Prof. (Dr.) S. C. Sharma, Director**, AITR, Indore whenever needed. We take opportunity to convey my regards to the **Management of Acropolis Institute, Indore** for extending academic and administrative support and providing us all necessary facilities for project to achieve our objectives.

We are grateful to our parent and family members who have always loved and supported us unconditionally.

Yash Patel	**Umesh Panchal**	**Shreya Verma**	**Raina Dhanyase**
Enrollment No: 0827CI221154	**Enrollment No: 0827CI221142**	**Enrollment No: 0827CI221124**	**Enrollment No: 0827CI221112**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviations

Abbr1: AI – Artificial Intelligence

Abbr2: API – Application programming Interface

Abbr3: CSS – Cascading Style Sheets

Abbr4: JSON- Java Script Object Notation

Abbr5: CSV – Comma Separated Value

Abbr6: UML – Unified Modeling Language

Abbr7: JWT – JSON Web Token

ABSTRACT

CleanSweep is a digital platform designed to simplify the process of reporting and managing waste-related issues. It allows citizens to easily report problems such as uncollected garbage, overflowing bins, or illegal dumping through a user-friendly interface. Complaints are categorized into three main areas: household waste, public area cleanliness, and unauthorized dumping . Once a complaint is submitted, it is automatically directed to the appropriate municipal department for prompt action. Users can track the status of their complaints in real-time, ensuring transparency and accountability. Administrators have access to a centralized dashboard to monitor complaints, assign tasks to waste management teams, and analyze data for informed decision-making. By digitizing the complaint process, CleanSweep reduces paperwork and accelerates response times. The platform encourages community engagement by enabling citizens to actively participate in maintaining a clean environment. Overall, CleanSweep aims to enhance the efficiency of waste management services and promote a cleaner ,healthier community.

CHAPTER 1

Chapter 1: Introduction

1.1 Overview

CleanSweep is a smart web-based complaint management system tailored to modernize urban waste monitoring and resolution. It empowers citizens to easily report cleanliness-related issues and enables municipal authorities to efficiently track, manage, and resolve them. The platform bridges the communication gap between the public and civic bodies by incorporating features such as area-based complaint logging, status tracking, AI-powered complaint prioritization, and an interactive 3D geolocation interface using CesiumJS. By digitizing and streamlining complaint handling, CleanSweep aims to enhance public sanitation standards, increase civic engagement, and improve accountability and responsiveness of city administrations of complaints **.**

1.2 Existing Systems

- **Swachhata App (Swachh Bharat Mission – Urban):**
  Launched by the Ministry of Housing and Urban Affairs (MoHUA), the Swachhata App is a nationwide platform enabling citizens to report civic-related issues such as garbage dumps, unclean public toilets, and more. Users can upload photographs of issues, which are then geo-tagged and forwarded to the respective municipal authorities for resolution. The app provides real-time updates on complaint status and allows users to reopen unresolved complaints, ensuring transparency and accountability in urban sanitation services .
- **Indore 311 App :**
  Developed by the Indore Municipal Corporation, the Indore 311 App facilitates direct communication between citizens and municipal authorities. Residents can report issues like garbage accumulation, potholes, and health hazards by capturing photos and submitting complaints through the app. The system ensures prompt redressal, with complaints typically addressed within

24 hours. The app also allows users to access information about various public services, enhancing civic engagement and service delivery

- **Recykal :**

  Recykal is a Hyderabad-based digital platform focusing on waste management and recycling. It connects waste generators, such as households and businesses, with recyclers and waste processors. Through its marketplace, Recykal facilitates the collection and recycling of various waste streams, including plastic, paper, and e-waste. The platform has collaborated with numerous brands and service providers to promote a circular economy and sustainable waste management practices .

1.3 Problem Statement

Urban municipalities face numerous challenges in managing sanitation complaints, including delayed responses, lack of real-time visibility into complaint locations, and inefficient resource allocation. Users often experience a lack of acknowledgment or transparency regarding the status of their complaints. Authorities struggle with disorganized complaint data, manual prioritization, and the absence of a centralized dashboard for complaint resolution tracking. These issues lead to reduced civic participation, unresolved complaints, and a decline in public trust. There is an urgent need for an intelligent, centralized, and interactive system that facilitates swift reporting, efficient tracking, and effective resolution of sanitation-related issues .

1.4 Proposed System

CleanSweep offers a comprehensive and intelligent solution to address these gaps in the waste management ecosystem .

For Citizens (Users) :

- Ability to submit complaints through a dynamic web form with area-based categorization.
- Real-time complaint status tracking (Pending, In Progress, Completed).
- Optional image upload and custom category support.

- Notifications and updates on complaint progress

For Authorities (admin) :

- Dashboard to manage complaints by status.
- Ability to update complaint status and view associated metadata (location, area, timestamp).
- Real-time 3D map visualization (CesiumJS) to identify complaint hotspots and location-based trends.
- AI-based model to prioritize complaints from CSV input using ML algorithm .

1.5 Need and Scope

1.5.1 Need :

India generates approximately 62 million tonnes of waste annually, with only about 70% being collected and a mere 12 million tonnes treated. The remainder often ends up in landfills or is improperly disposed of, leading to environmental pollution and health hazards. Rapid urbanization and population growth exacerbate this issue, overwhelming existing waste management infrastructures. Additionally, many municipalities lack efficient systems for residents to report and track waste-related complaints, resulting in delayed responses and unresolved issues. There's a pressing need for a streamlined, transparent, and user-friendly platform that empowers citizens to actively participate in waste management and enables authorities to address complaints promptly .

1.5.2 Scope :

The CleanSweep project aims to develop a comprehensive digital platform that facilitates the reporting, tracking, and management of waste-related complaints. Key features will include user registration, complaint submission with geotagging and photographic evidence, real-time status updates, and administrative dashboards for monitoring and resolution. The system will cater to various stakeholders, including citizens, municipal authorities, and waste management personnel, ensuring seamless communication and accountability. By integrating modern technologies and adhering to the Solid Waste Management Rules, 2016, CleanSweep seeks to enhance

operational efficiency, promote community engagement, and contribute to a cleaner, healthier urban environment .

1.6 Report Organisation

- Chapter 1 introduces the CleanSweep project, discussing the current system's limitations, the need for modernization, and the proposed smart solution.

- Chapter 2 provides a literature survey and background of the project, discussing the software engineering paradigms followed and detailing the software and hardware technologies used in building the system .

- Chapter 3 covers the system analysis, including requirement gathering and feasibility studies – technical, financial, and operational .

- Chapter 4 outlines the project planning process, including scope definition, work breakdown structure, timeline and scheduling, risk analysis, and quality management .

- Chapter 5 details the system design, including UML diagrams, ER diagrams, data flow diagrams, and the data dictionary .

- Chapter 6 describes the implementation phase, outlining how each module of CleanSweep was developed and integrated.

- Chapter 7 presents the testing strategy, methods used (unit testing, integration testing, etc.), and sample test cases with results.

- Chapter 8 concludes the report by summarizing the outcomes and suggesting potential future enhancements to extend the system's capabilities.

- References include all sources such as websites, research papers, books, and tools that were used throughout the development of CleanSweep .

CHAPTER 2

Chapter 2: Literature Survey

2.1 Study

1. In urban management, addressing public hygiene and waste-related issues through digital solutions has become essential due to growing population density and increasing complaint volume.

2. CleanSweep aims to resolve day-to-day municipal challenges by automating waste complaint registration, tracking, and resolution using a centralized, digital approach.

3. Traditional manual systems involve paperwork, delayed responses, and lack of centralized data, making complaint management inefficient and difficult to monitor in real-time.

4. This application proposes a centralized web-based platform, reducing the dependency on manpower and paperwork, while improving transparency and response time.

5. The primary objective is to provide an intuitive and efficient interface for both citizens and authorities to handle public complaints swiftly and reliably.

6. Our end goal is to streamline the complaint management cycle—from submission to resolution—by eliminating scattered data handling, reducing workload, and enhancing citizen satisfaction.

2.2 Problem Methodology

The development of the CleanSweep system adopts an **Agile Software Development Life Cycle (SDLC)** approach, emphasizing iterative progress and stakeholder

collaboration. Initially, the team conducts **requirement gathering** sessions with citizens and municipal authorities to identify essential features such as complaint registration, geolocation tracking, and real-time status updates. Following this, the **design phase** involves creating wireframes for user interfaces and establishing the system architecture, ensuring alignment with user needs. The **development phase** proceeds in short, iterative cycles called sprints, where specific functionalities are implemented and integrated into the system. Each sprint is accompanied by **testing activities**, including unit and integration testing, to ensure software quality and reliability. Upon completion of development, the system undergoes **deployment**, making it accessible to end-users. Post-deployment, the team engages in **maintenance and feedback collection**, addressing any issues and incorporating user suggestions for continuous improvement. This Agile methodology facilitates flexibility, encourages user involvement, and ensures the delivery of a responsive and efficient waste complaint management system.

2.3 Software Engineering Paradigm

The principle of software design mainly includes the following:

1. **Agile Development :** Agile practices are used for rapid iteration, continuous feedback, and modular release of CleanSweep features.
2. **Reliability :** The system is tested for consistent performance, including fault handling (e.g., invalid data, offline file uploads) and secure complaint management.
3. **Incremental Development :** The project was developed in modules such as user login, complaint form, admin dashboard, CesiumJS map, and AI-based complaint prioritization—each built and tested in stages.
4. **Collaboration :** Team members (frontend, backend, AI, research) collaborate using GitHub, Postman, and Trello for task management, ensuring alignment with project goals.

5. **Continuous Testing and Improvement :** Testing is carried out in each sprint to ensure platform stability and accuracy in complaint data, location tracking, and status updates .
6. **Focus on User-Centric Design :** UI is designed to be simple and responsive using Tailwind CSS, ensuring accessibility for users from both technical and non-technical backgrounds

2.4 Software Development Life Cycle

2.4.1 Requirement Analysis :

- Requirements gathered from students, local residents, and municipal teams via Google Forms and interviews.
- Functional needs included: Complaint registration with image upload, category-based filtering, area-specific submission, status updates, and real-time map .

2.4.2 Feasibilty Study :

- Analyzed technical feasibility using open-source stack (React, Node, MongoDB) .
- Financial feasibility was supported through free-tier cloud services (Render, Railway, Vercel).
- Evaluated future scalability with integration potential for AI and PWA/mobile versions .

2.4.3 System Design :

- Created ER diagrams, complaint lifecycle flowcharts, and MongoDB schema using Mongoose .
- Designed responsive UI for mobile and desktop .
- Prepared CesiumJS 3D layout for complaint location visualization .

2.4.4 Implementation :

- Frontend: React + Vite + Tailwind CSS .
- Backend: Node.js + Express.js + MongoDB .

- Modules: User Auth, Complaint CRUD, Admin Panel, File Upload, Map API, AI script integration .

2.4.5 Testing :

- Unit tests on key backend routes using Postman and Jest .
- Integration tests between frontend forms and backend API .
- CesiumJS and file handling tested with multiple location data .
- Admin tested for proper status filtering and file metadata validation .

2.4.6 Deployment :

- Frontend hosted on Vercel, backend on Railway .
- MongoDB Atlas used for secure and scalable database storage .
- GitHub used for version control and CI/CD deployment pipelines .

2.4.7 Maintenance :

- Bug tracking and updates handled via GitHub Issues .
- Planned feature updates include analytics dashboard, push notifications, and AI enhancements .
- Feedback collected for iterative UI/UX improvements .

2.5 Technology Methodology

2.5.1 Hardware Requirements

The used hardware for development of application system:

1. RAM: 16 GB

2. Processor: i5 9^{th} generation or above

2.5.2 Software Requirements

The required software for development environment:

- **Operating System** : Windows 11 Pro
- **Database**: MongoDB Atlas (Cloud-based NoSQL database)
- **Backend**: Node.js (v18.16.0) with Express.js framework
- **Frontend**: React (utilizing Vite for fast development) with Tailwind CSS for styling)
- **3D Mapping**: CesiumJS for geospatial visualization .
- **Code Editor**: Visual Studio Code (v1.83.0) .
- **API Testing Tool** : Postman
- **Deployment** : Vercel for frontend deployment, Railway for backend deployment .
- **Version Control** : Git with GitHub for source code management .

CHAPTER 3

Chapter 3 : Analysis

3.1 Identification of System Requirements

In the development of the CleanSweep system, identifying comprehensive system requirements is crucial to ensure the platform effectively addresses the needs of both citizens and municipal authorities in managing waste-related complaints. These requirements encompass functional and non-functional aspects, ensuring the system's efficiency, reliability, and user-friendliness .

The system requirements serve multiple purposes :

- **Foundation for System Design**: They provide a blueprint for architects and developers to design a system that meets specified functionalities and performance criteria.
- **Guidance for Integration and Testing**: Clear requirements facilitate the integration of various system components and establish benchmarks for testing and validation processes .
- **Stakeholder Communication**: They act as a reference point for discussions among stakeholders, ensuring that the system aligns with user expectations and regulatory standards .
- **Basis for Validation and Acceptance**: Defined requirements enable stakeholders to validate the system's performance and accept its deployment based on predefined criteria .

During the system definition phase, requirements are carefully documented to ensure consistency and traceability throughout development, enabling CleanSweep to remain robust, scalable, and adaptable to evolving user needs.

3.2 Functional Requirements

- **User Registration and Authentication** : Citizens can create personal accounts by providing necessary details. Once registered, they can securely log in to access the system's features .

- **Complaint Submission:** Users can lodge complaints related to waste management issues by filling out a form detailing the problem and its location .
- **Complaint Tracking:** After submission, users can monitor the status of their complaints in real-time, receiving updates as the issue progresses through various stages .
- **Geolocation Integration:** The system utilizes geolocation to pinpoint the exact location of reported issues, aiding in efficient resolution by the concerned authorities .
- **Administrative Dashboard:** Administrators have access to a comprehensive dashboard where they can view, assign, and manage complaints, ensuring timely action and resolution .
- **Waste Collector Interface:** Assigned waste collectors can view their tasks, update the status of complaints they've addressed, and communicate any challenges faced during resolution .
- **Notification System:** The platform sends timely notifications to users about the status of their complaints, ensuring they are informed throughout the resolution process .
- **Data Analytics and Reporting:** The system generates reports and analytics on complaint trends, helping authorities identify recurring issues and plan preventive measures .
- **Feedback Mechanism:** After a complaint is resolved, users can provide feedback on the resolution process, helping improve service quality .
- **Multi-Platform Accessibility:** The application is accessible via various devices, including desktops, tablets, and smartphones, ensuring users can report issues conveniently .

3.3 Non Functional Requirements

- **Scalability :** The system should efficiently handle an increasing number of users and complaint submissions over time. This ensures continued performance as the user base and data volume grow **.**
- **Performance :** The application must respond promptly to user actions, with minimal loading times. Quick interactions enhance user experience and satisfaction .
- **Reliability :** CleanSweep should operate consistently without unexpected failures or crashes. Users should trust the system to be available whenever needed .
- **Security:** User data must be protected through secure authentication methods like JWT or session-based logins. Safeguarding information builds user trust and complies with data protection standards .
- **Usability :** The interface should be intuitive, clean, and accessible across various devices. A user-friendly design ensures that users can navigate and utilize the system effectively.

3.4 Identification of System Requirements

The feasibility study evaluates the practicality and viability of implementing the CleanSweep system, ensuring that it meets technical standards, remains financially sustainable, and operates effectively within the intended environment .

3.4.1 Technical Feasibility

CleanSweep is developed using modern and widely adopted technologies :

- **Frontend**: Built with React (Vite) and styled using Tailwind CSS, ensuring a responsive and user-friendly interface .
- **Backend**: Implemented with Node.js (v18.16.0) and Express.js, providing a robust server-side framework .
- **Database**: Utilizes MongoDB Atlas for scalable and flexible data storage .
- **3D Mapping**: Incorporates CesiumJS for geospatial visualization of waste complaint locations .

- **AI Integration (Optional)**: Employs Python 3.10 with TF-IDF/Embedding-based machine learning models for advanced data analysis .

These technologies are open-source and have extensive community support, making them suitable for developing a scalable and maintainable system .

3.4.2 Financial Feasibilty

The project leverages cost-effective resources :

- **Development Tools**: Utilizes free and open-source tools like Visual Studio Code (v1.83.0) for coding and Postman for API testing .
- **Deployment**: Frontend is deployed on Vercel, and the backend is hosted on Railway, both offering free tiers suitable for initial deployment phases .
- **Version Control**: Employs Git and GitHub for source code management, which are free for public repositories .

By utilizing these resources, the development and deployment of CleanSweep incur minimal costs, making it financially feasible for municipal bodies and organizations with limited budgets .

3.4.3 Operational Feasibility

CleanSweep is designed to integrate seamlessly into existing municipal operations :

- **User Roles**: Supports multiple user roles, including citizens, administrators, and waste collectors, each with tailored interfaces and functionalities .
- **Complaint Management**: Streamlines the process of lodging, tracking, and resolving waste-related complaints, enhancing transparency and accountability .
- **Data Analytics**: Provides insights into waste management patterns, aiding in decision-making and resource allocation .

The system's design aligns with current operational workflows, ensuring that it can be adopted with minimal disruption and training .

CHAPTER 4

Chapter 4 : Project Planning

The CleanSweep project follows a structured approach to ensure efficient execution and timely delivery. The planning begins with requirement gathering and analysis, where system requirements are identified and input from stakeholders such as municipal authorities, developers, and local citizens is collected. Once requirements are defined, the system's architecture is designed, including the frontend, backend, database, authentication mechanisms, and integration with a 3D geolocation interface. Wireframes and mockups are created to visualize the user interface. During the development phase, the frontend is built using React and Tailwind CSS, and the backend is developed using Node.js and Express.js, with MongoDB used for database management and secure authentication implemented. The system undergoes rigorous testing, including unit, integration, and system testing, to ensure performance, security, and quality. Once testing is complete, the system is deployed, and users are introduced to its functionality. Post-deployment, continuous maintenance and support are provided to address any issues, gather user feedback, and improve the platform as needed. With well-defined phases, clear milestones, and assigned responsibilities, CleanSweep ensures smooth and timely project planning .

Proposed TimeTable

Module	**Duration (Weeks)**	**Activities & Deliverables**

Module 1: Project Planning	1	- Define project scope & goals. - Identify stakeholders: Admins, Citizens, Developers. - Finalize system requirements & use cases. - Create project roadmap & timeline.
Module 2: Requirements Gathering	2	- Gather input from city officials and users. - Define complaint lifecycle and statuses. - Identify features (e.g., complaint form, map view, admin dashboard).
Module 3: Design & Development	5	- Design UI/UX wireframes for citizen and admin panels. - Develop backend APIs (Node.js + Express). - Build frontend (React + Tailwind CSS). - Set up MongoDB schema and connections.
Module 4: Authentication Setup	3	- Implement secure login for Admins and Users. - Add role-based access control. - Ensure password protection and session management.

Module 5: omplaint Management & 3D Visualization	6	- Implement complaint submission, tracking & status update system. - Integrate CesiumJS 3D map for location-based complaint view. - Develop admin features for filtering, sorting, and managing complaints.
Module 6: Testing & Validation	4	- - Perform unit testing, integration testing, and system testing. - Validate AI-based priority classification from CSV uploads. - Ensure geolocation accuracy and admin workflows.
Module 7: Deployment & Feedback	3	- Deploy CleanSweep platform to live environment. - Gather feedback from early users and city staff. - Finalize user guide, documentation, and performance improvements.

CHAPTER 5

Chapter 5 : Design

5.1 Introduction to UML

Unified Modeling Language (UML) is a standard way to visually represent how a system is designed. Think of it like a blueprint for software, similar to how architects use blueprints for buildings planning .

UML isn't a programming language; instead, it's a visual tool that helps developers, stakeholders, and system architects understand and communicate the structure and behavior of a system. By using UML diagrams, we can plan, design, and analyze our system effectively before actual development begins .

5.2 UML Diagrams

For complex applications like CleanSweep, it's essential to have clear communication among various teams. UML diagrams serve this purpose by providing a visual representation of the system's components and their interactions .

UML Diagrams are broadly categorized into :

1. **Structural Diagrams:** These depict the static aspects of the system, such as its components and their relationships. Examples include :
    - Class Diagram
    - Component Diagram
    - Deployment Diagram
2. **Behavioral Diagrams:** These illustrate the dynamic behavior of the system, showing how it responds to events. Examples include :
    - Use Case Diagram
    - Activity Diagram
    - Sequence Diagram

5.2.1 Use Case Diagarm

A Use Case Diagram shows how different users (actors) interact with the system. It helps in identifying the functionalities the system should offer .

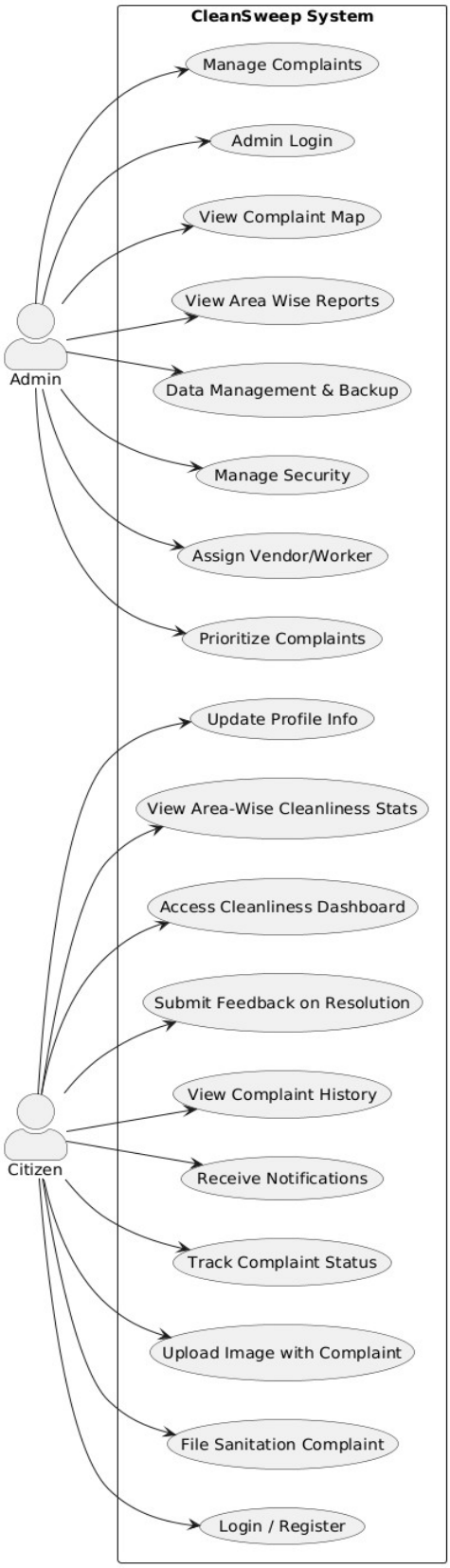
In the context of CleanSweep, the primary actors and their interactions are :

CITIZEN –

- Register and log in to the system .
- Submit waste-related complaints .
- Track the status of their complaints .

ADMINISTRATOR –

- Manage user accounts .
- Assign complaints to appropriate personnel .
- Monitor overall system performance .


CleanSweep System
Manage Complaints
Admin Login
View Complaint Map
View Area Wise Reports
Data Management & Backup
Manage Security
Assign Vendor/Worker
Prioritize Complaints
Update Profile Info
View Area-Wise Cleanliness Stats
Access Cleanliness Dashboard
Submit Feedback on Resolution
View Complaint History
Receive Notifications
Track Complaint Status
Upload Image with Complaint
File Sanitation Complaint
Login / Register
Admin
Citizen

5.2.2 Class Diagram

A Class Diagram shows the internal structure of the system by defining its classes, their attributes, functions, and how they interact with each other. For the CleanSweep system, it helps in designing the backend logic and mapping how users, complaints, and services work together.

- **User**

  i) Attributes: userId, name, email, password

  ii) Methods: login(), register()

  iii) Role : Can register, log in, and submit complaints.

- **Admin**

  i) Attributes: adminId, username, password

  ii) Methods: login(), updateComplaintStatus()

  iii) Role: Can manage and update complaint statuses.

- **AI Model**

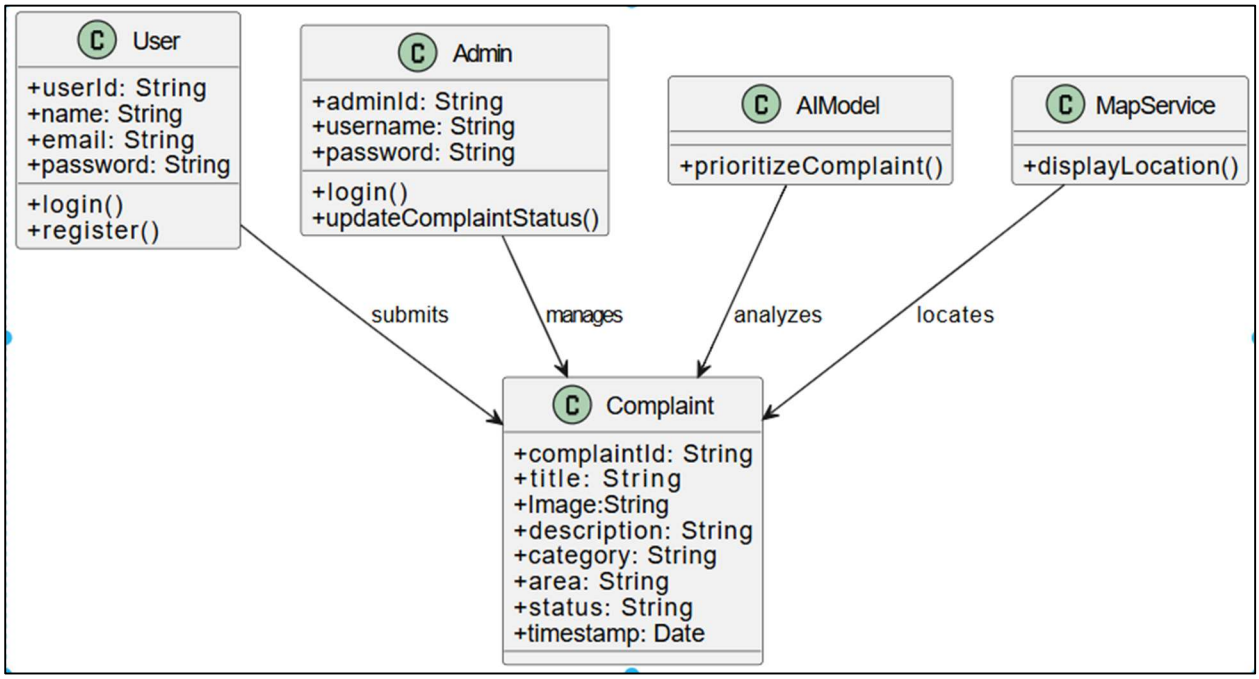
  i) Method: prioritizeComplaint()

  ii) Role: Uses ChatGPT API to analyze and rank complaints.

- **MapService**

  i) Method: displayLocation()

  ii) Role: Displays complaint locations on the 3D map.

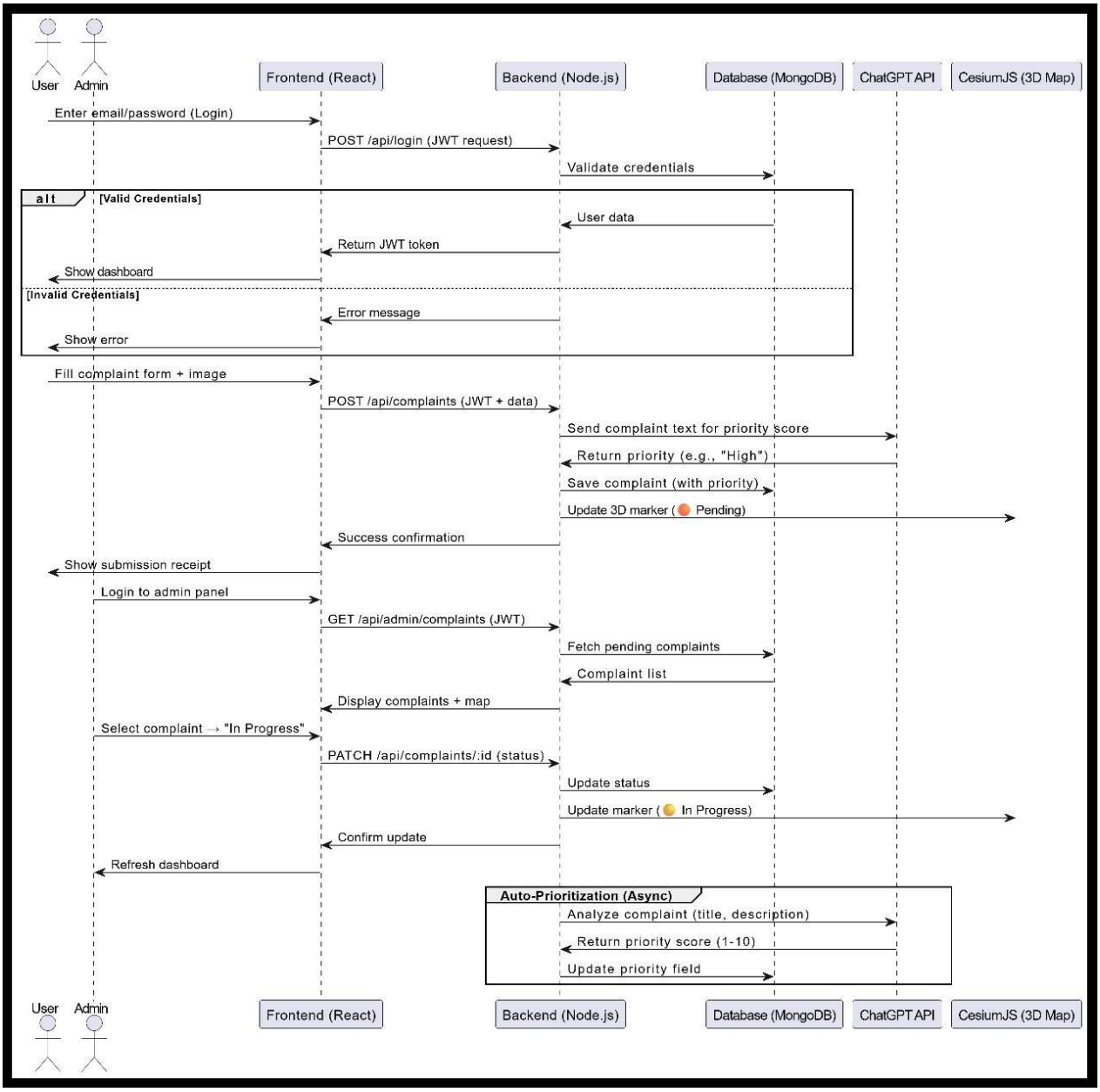


5.2.3 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describe how-and in what order- a group of objects works together .

Sequence diagram are sometimes known as an event diagrams or event scenarios. Sequence diagram are time focus .

A Sequence Diagram shows how different components of the CleanSweep system interact with each other step-by-step over time. It captures the flow of messages between actors like the User, Admin, Backend, Database, and AI Model. For example, when a user submits a complaint, the diagram shows how the frontend sends the data to the backend, stores it in the database, and gets analyzed by the AI model. It helps in understanding the dynamic behavior of the system. This diagram is useful for developers to design and debug the real-time flow of actions in the app .

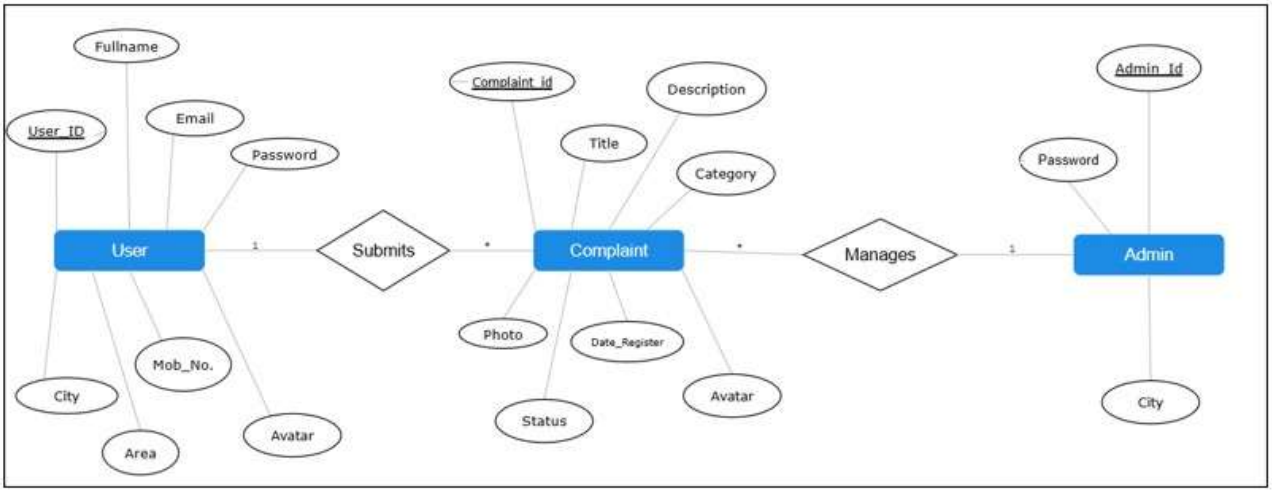


5.2.4 ER Diagram

ER Diagram is a visual representation of data that describes how data is related to each other. In ER Model, we disintegrate data into entities, attributes and setup relationships between entities, all this can be represented visually using the ER diagram. ER Diagrams

contain different symbols that use rectangles to represent entities, ovals to define attributes and diamond shapes to represent relationships .

An ER (Entity-Relationship) Diagram represents the data structure of the CleanSweep system. It shows key entities like User, Admin, Complaint, and Area, along with their attributes and relationships. This diagram helps in designing the database by clearly defining how data is stored and connected. It acts as a foundation for creating tables in a relational database .

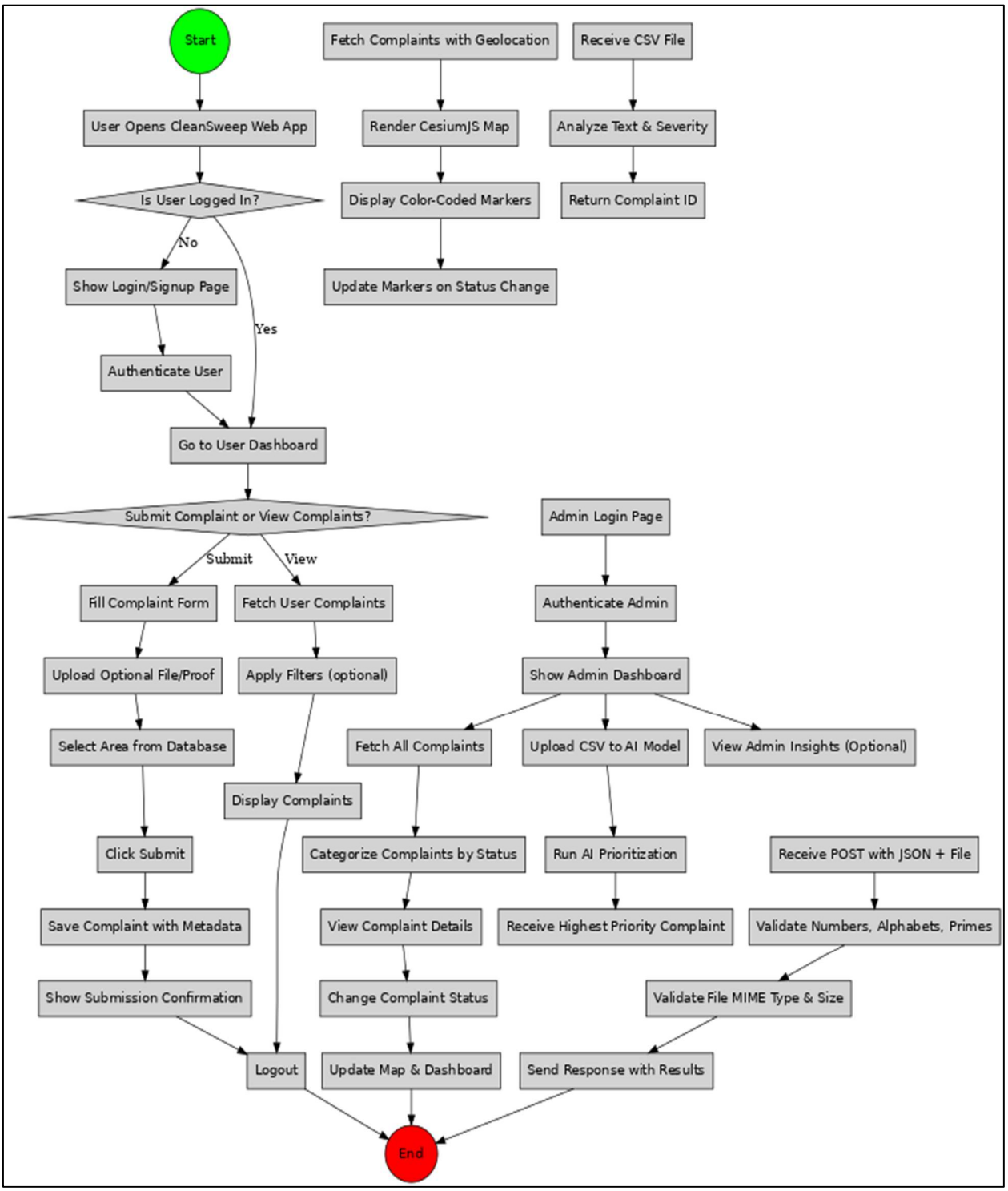


5.2.5 Activity Diagram

An activity diagram portrays the control flow from a start point to finish point showing various decision paths that exist while the activity is being executed .An activity diagram focuses on condition of flow and the sequence in which it happens. An activity diagram is a behavioral diagram i.e. it depicts the behavioral of a system .

An **Activity Diagram** illustrates the **workflow of various processes** in the CleanSweep system. It visually represents the **sequence of actions**, decisions, and parallel activities involved in key features like complaint submission, status update, and complaint resolution. For example, the flow starts from a user logging in, filling the complaint form,

submitting it, and the system validating and storing the complaint. The admin later updates the status, which is reflected back to the user. Decision nodes like "Is user authenticated?" or "Is complaint valid?" help define conditional flows. This diagram helps in understanding how the system behaves from **start to end** during a particular operation. It is useful for both developers and stakeholders to analyze system logic and identify improvements .

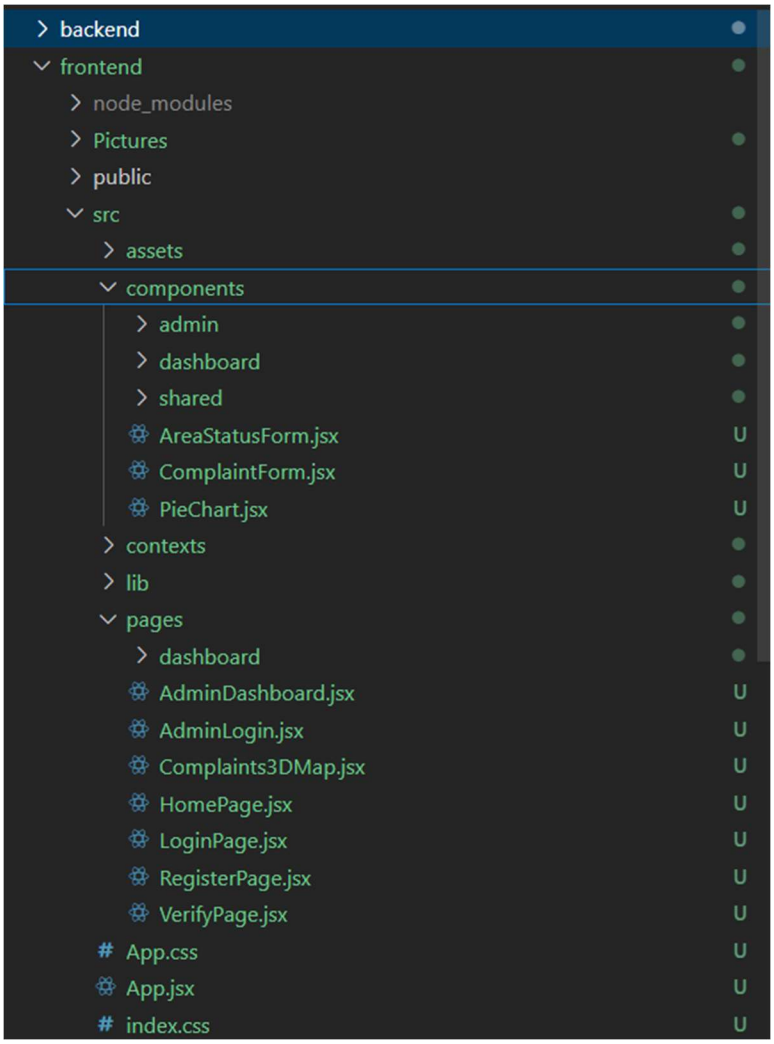

Start
User Opens CleanSweep Web App
Is User Logged In?
No
Yes
Show Login/Signup Page
Authenticate User
Go to User Dashboard
Submit Complaint or View Complaints?
Submit
View
Fill Complaint Form
Upload Optional File/Proof
Select Area from Database
Click Submit
Save Complaint with Metadata
Show Submission Confirmation
Fetch User Complaints
Apply Filters (optional)
Display Complaints
Logout
Fetch Complaints with Geolocation
Render CesiumJS Map
Display Color-Coded Markers
Update Markers on Status Change
Receive CSV File
Analyze Text & Severity
Return Complaint ID
Admin Login Page
Authenticate Admin
Show Admin Dashboard
Fetch All Complaints
Upload CSV to AI Model
View Admin Insights (Optional)
Categorize Complaints by Status
View Complaint Details
Change Complaint Status
Update Map & Dashboard
Run AI Prioritization
Receive Highest Priority Complaint
Receive POST with JSON + File
Validate Numbers, Alphabets, Primes
Validate File MIME Type & Size
Send Response with Results
End

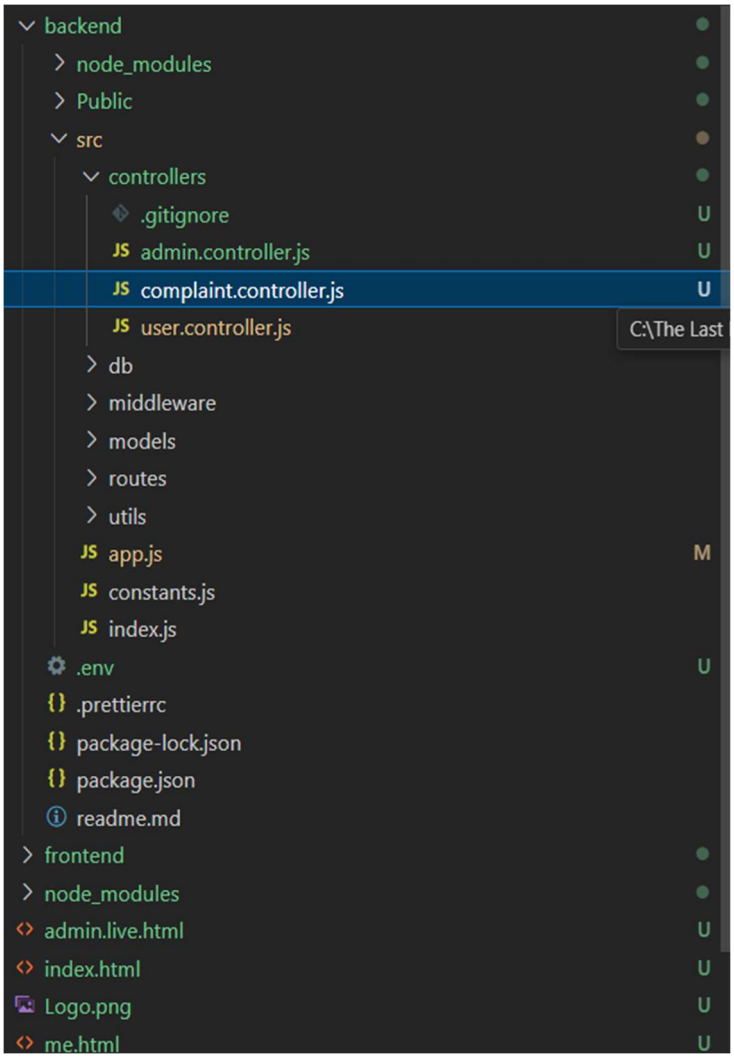
CHAPTER 6

Chapter 6 : Implementation

6.1 Coding (Main Module)

The main module of the **CleanSweep** application is developed using HTML, Tailwind CSS, and JavaScript (React) for the frontend, ensuring a user-friendly and responsive interface. Tailwind CSS is utilized to maintain a consistent and modern design theme across the platform. The backend is built with Node.js and Express.js, which manage server-side logic and API endpoints. MongoDB with Mongoose is used for database management, enabling efficient handling of user complaints, location data, and cleanup statuses. All backend dependencies are managed using npm. This full-stack architecture ensures smooth communication between the frontend and backend, resulting in a dynamic, real-time, and efficient web application tailored for smart city waste management .

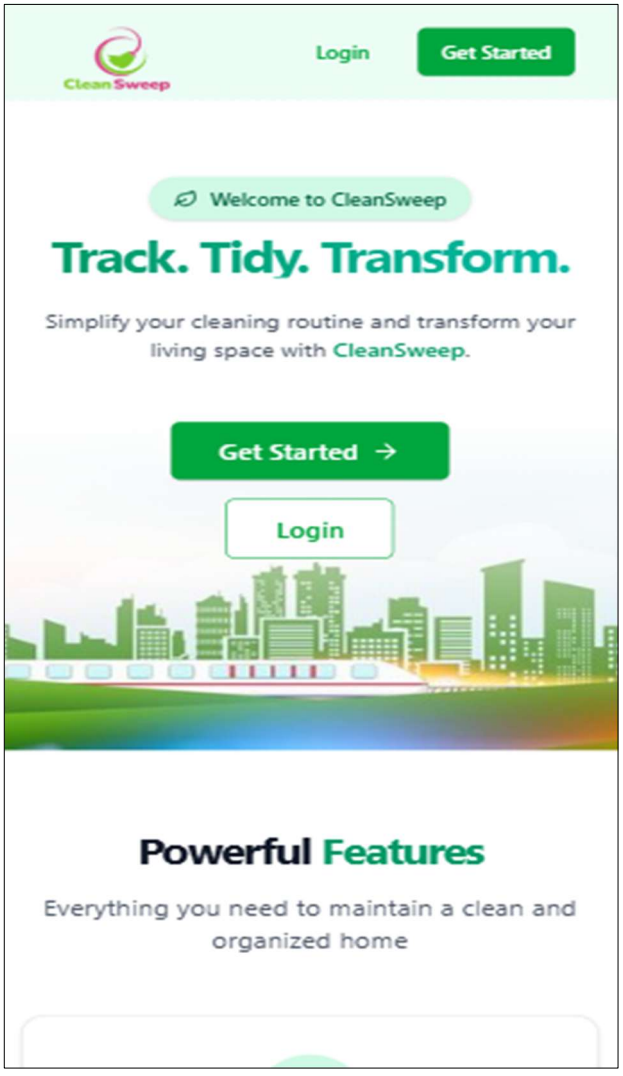

backend
frontend
node_modules
Pictures
public
src
assets
components
admin
dashboard
shared
AreaStatusForm.jsx U
ComplaintForm.jsx U
PieChart.jsx U
contexts
lib
pages
dashboard
AdminDashboard.jsx U
AdminLogin.jsx U
Complaints3DMap.jsx U
HomePage.jsx U
LoginPage.jsx U
RegisterPage.jsx U
VerifyPage.jsx U
App.css U
App.jsx U
index.css U


backend
node_modules
Public
src
controllers
.gitignore U
admin.controller.js U
complaint.controller.js U
user.controller.js
C:\The Last
db
middleware
models
routes
utils
app.js M
constants.js
index.js
.env U
.prettierrc
package-lock.json
package.json
readme.md
frontend
node_modules
admin.live.html U
index.html U
Logo.png U
me.html U

6.2 Results : Screen Shots

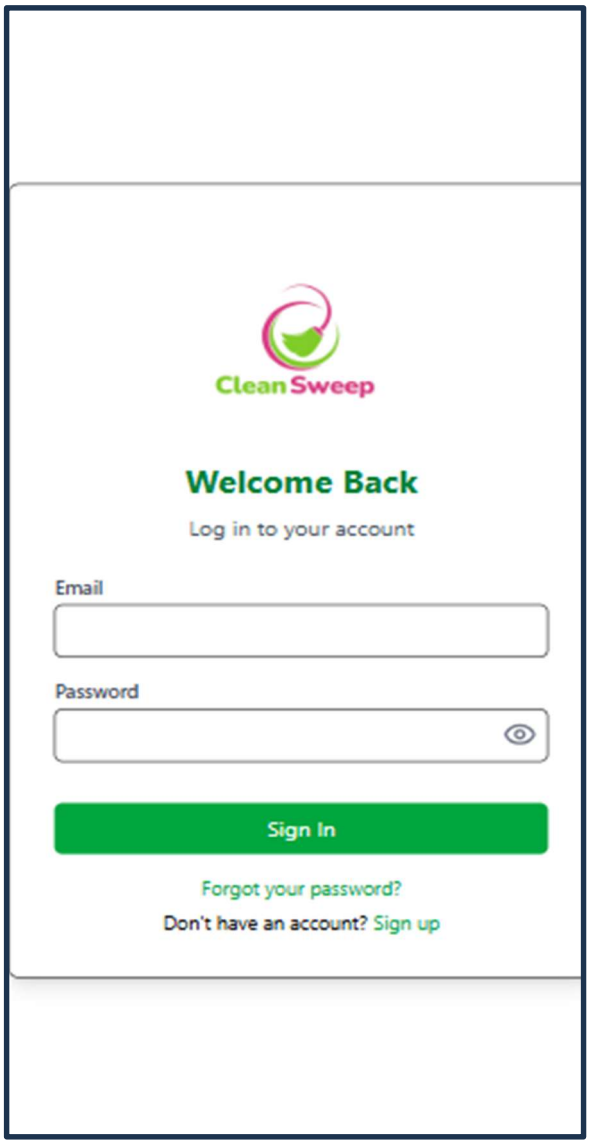
6.2.1 First Screen (Landing Page)

Displays the **CleanSweep** logo, mission statement, and a brief introduction to the app's purpose. Includes navigation options like **Login**, **Sign Up**, and **Get Started** for quick access .



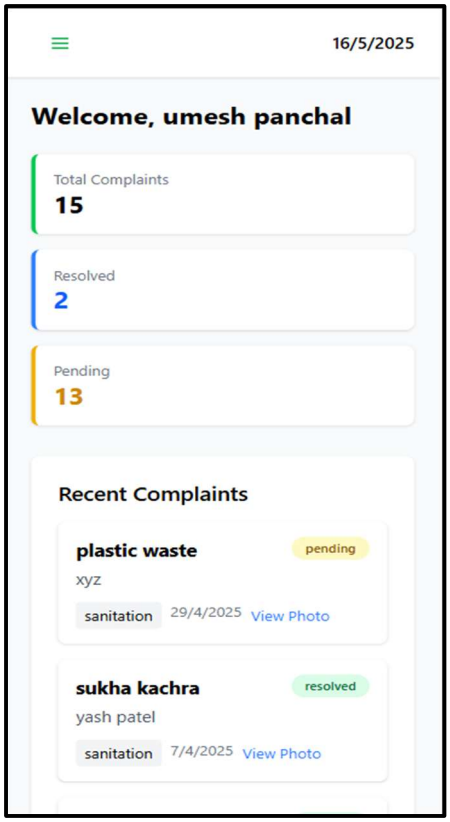
6.2.2 Signup/Login Screen

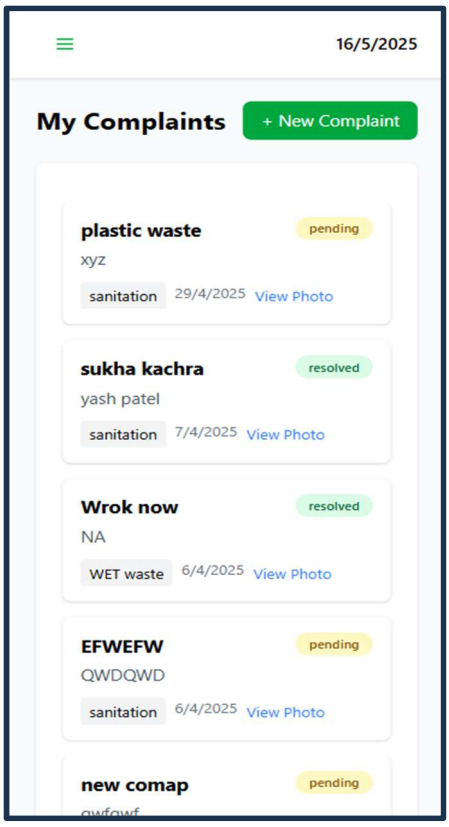
Allows users to create an account or log in using their credentials. Basic validation ensures secure and accurate input. Provides user authentication for accessing personalized features .

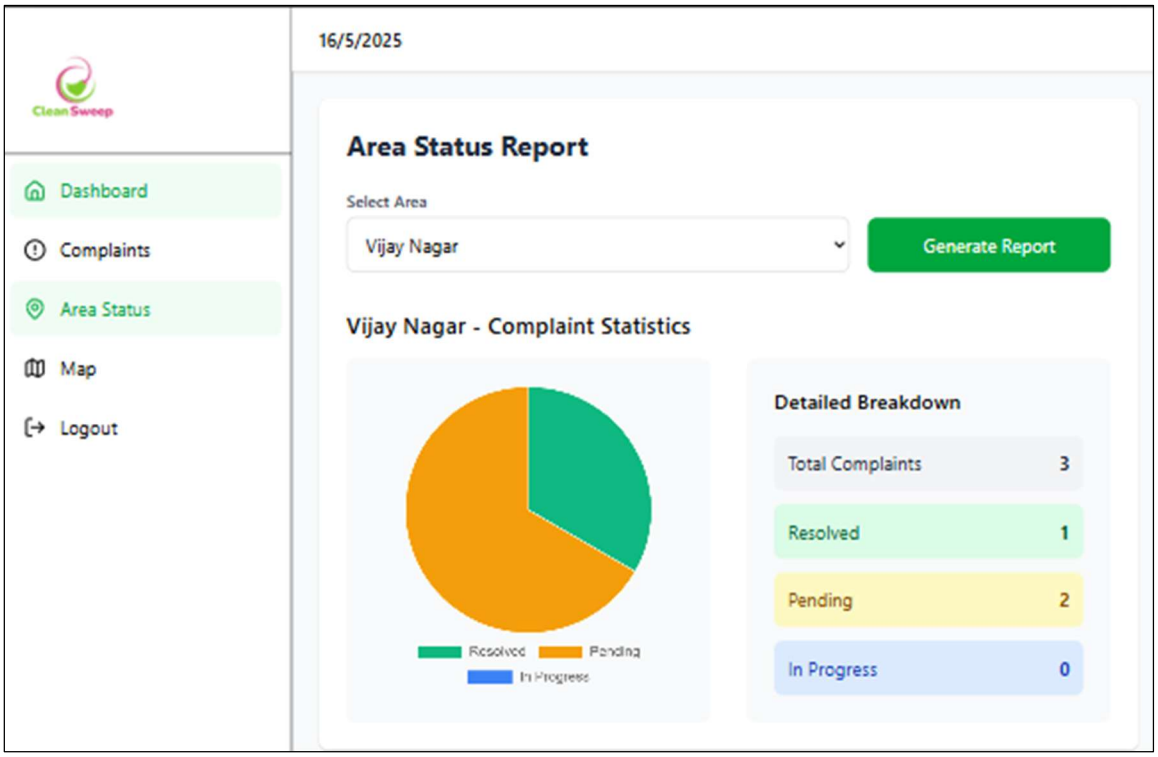


6.2.3 User Dashboard

After login, users access a dashboard displaying their submitted complaints, complaint status (Pending, In Progress, Resolved), and estimated cleanup time. A clean and intuitive layout helps users navigate easily .

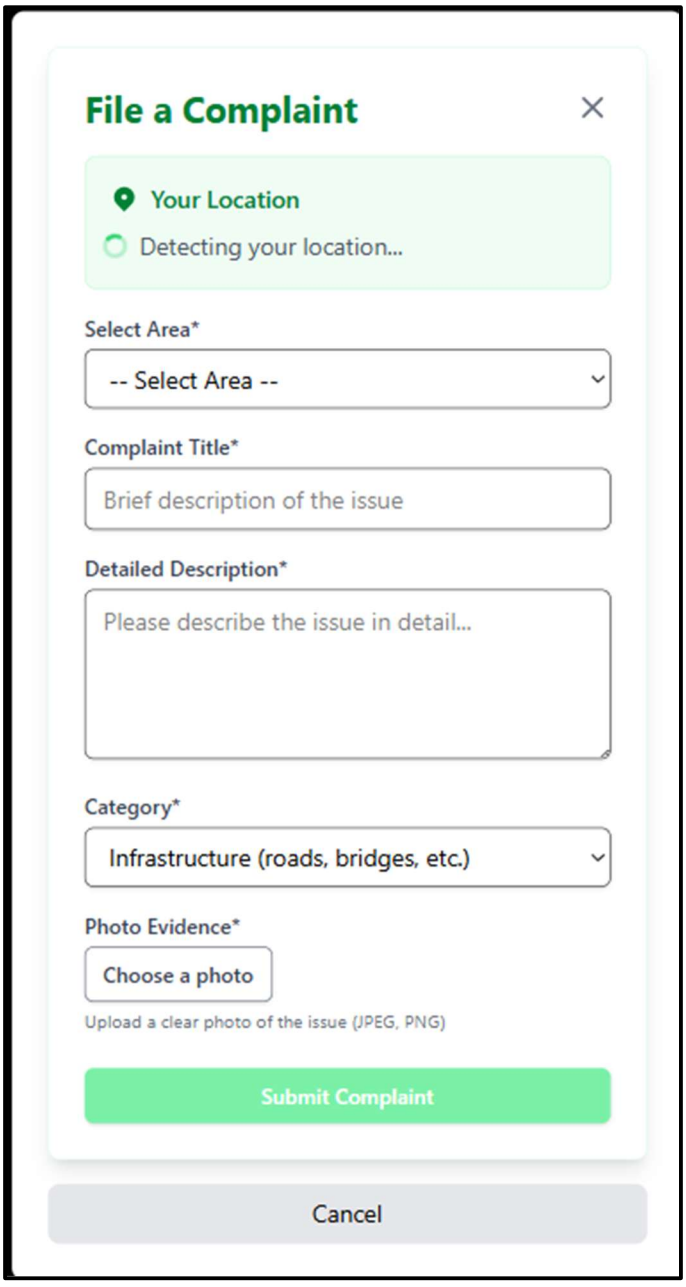





Clean Sweep
16/5/2025
Dashboard
Complaints
Area Status
Map
Logout
Area Status Report
Select Area
Vijay Nagar
Generate Report
Vijay Nagar - Complaint Statistics
Resolved
Pending
In Progress
Detailed Breakdown
Total Complaints 3
Resolved 1
Pending 2
In Progress 0

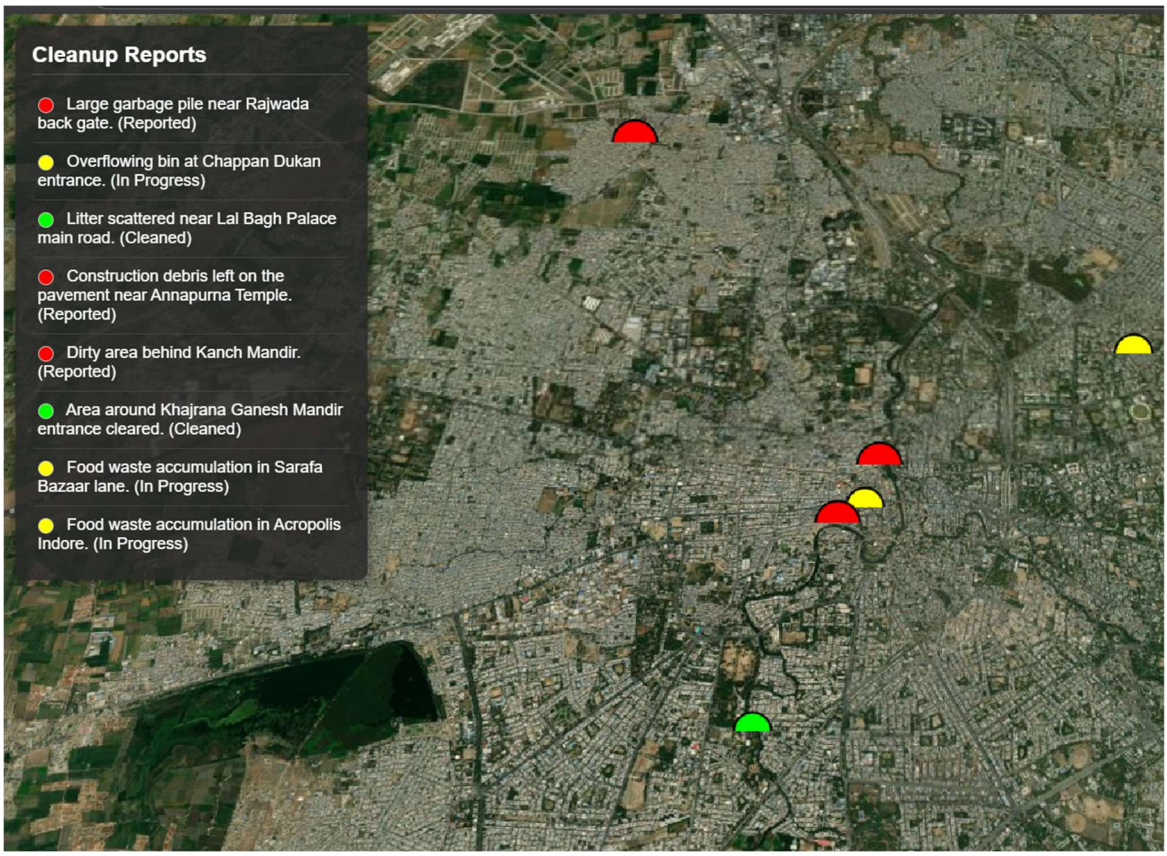
6.2.4 Complain Submission Screen

Users can submit complaints by filling out a form that includes Title, Description, Category, Area, and optionally uploading an image. Area options are dynamically fetched from the database, and users can create custom categories .



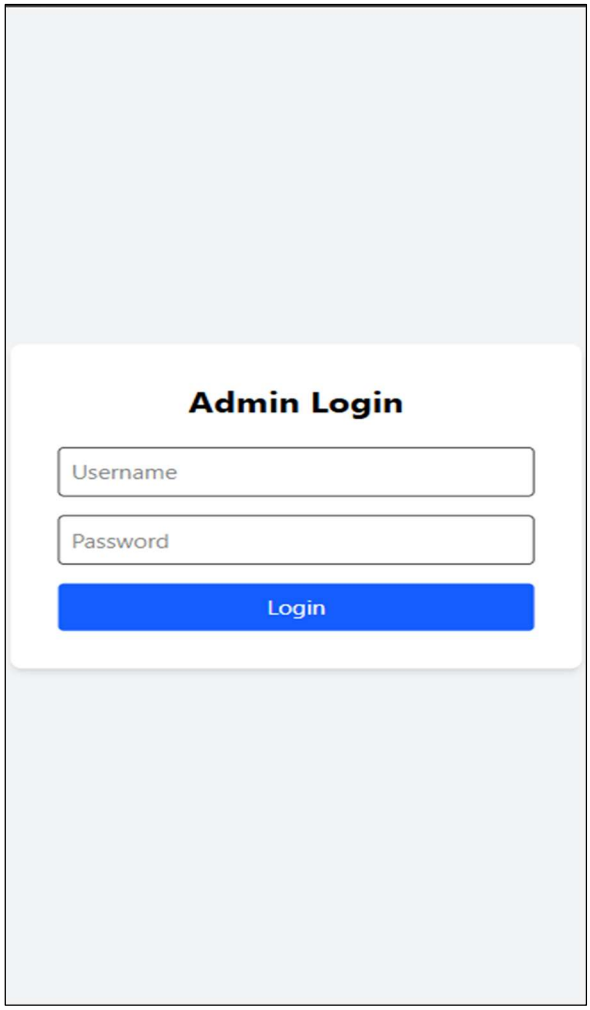
6.2.5 3D Map Interface

Visualizes complaint locations on a 3D map. Each marker indicates complaint status with color codes and popups showing complaint details. Enables interactive exploration of city cleanup issues .



6.2.6 Admin Login Screen

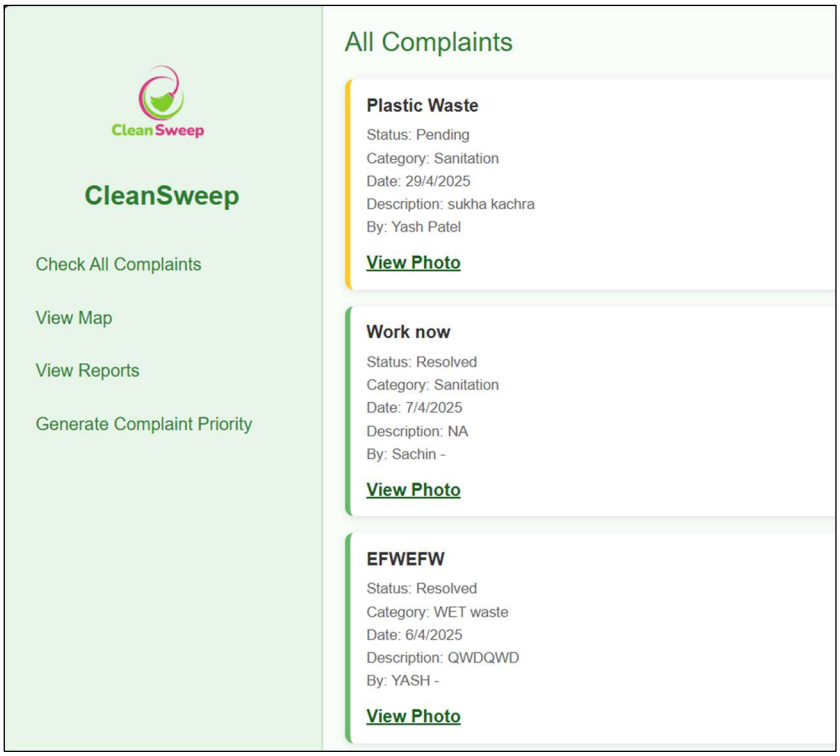
Allows administrators to securely log into the system to manage complaints and users. Validates admin credentials before granting access to backend functionalities .



6.2.7 Admin Login Screen

Provides a control panel for administrators to :

- View all complaints .
- Change complaint statuses .
- View analytics on complaint trends and area-wise reports .



6.2.8 User Collection

```
_id: ObjectId('67e2e0188b223e0242e5c444')
fullName : "umesh panchal"
username : "umesh142"
email : "uprewas@gmail.com"
State : "MP"
District : "RATLAM"
Area : "JAORA"
avatar : "http://res.cloudinary.com/dpweu94ja/image/upload/v1742921751/hmr166vjb…"
password : "$argon2id$v=19$m=65536,t=3,p=4$aahuCSEWMFab5zflxuaONA$YKDUSUtltnoNvfgn…"
isVerified : true
otp : "908590"
createdAt : 2025-03-25T16:55:52.321+00:00
updatedAt : 2025-05-15T19:25:33.500+00:00
__v : 0
refreshToken : "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2N2UyZTAxODhiMjIzZTAyN…"
```

6.2.9 Complaint Collecion

```
_id: ObjectId('67ebd282fb866930b82f8abf')
title : "Hello"
description : "dqwefew"
latitude : 22.753693775502324
longitude : 75.89337421466894
photo : "http://res.cloudinary.com/dpweu94ja/image/upload/v1743508095/gx33w8ka6…"
status : "pending"
createdBy : ObjectId('67e2e0188b223e0242e5c444')
category : "infrastructure"
createdAt : 2025-04-01T11:48:18.863+00:00
updatedAt : 2025-04-01T11:48:18.863+00:00
__v : 0
```

CHAPTER 7

Chapter 7 : Testing

7.1 Introduction

Testing is a critical phase in the software development lifecycle that ensures the system functions as expected, meets user requirements, and is free of critical bugs. For CleanSweep, a platform-centric civic complaint management system, testing was carried out systematically across all layers of the MERN stack—**frontend, backend, database, and integration with AI agents and cloud-based services** .

7.2 Testing Objectives

- Verify that users can submit complaints accurately and receive responses .
- Ensure AI agents assign correct priority levels based on complaint content .
- Validate automated escalation after predefined thresholds .
- Confirm admin users can track, assign, and resolve complaints .
- Ensure data integrity and proper role-based access control .
- Validate map view, real-time updates, and feedback loop .

7.3 Types of Testing

7.3.1 Unit Testing

Each component of the application, such as form inputs, API calls, AI agent responses, and MongoDB queries, was tested in isolation using tools like **Jest** and **Mocha**. For example, individual APIs were tested to ensure they return the correct HTTP status and payloads .

7.3.2 Integration Testing

This tested the interactions between modules—such as submitting a complaint and seeing it reflected on the admin dashboard, including AI-generated priority .

7.3.3 System Testing

CleanSweep was tested end-to-end as a complete system. A user would submit a complaint, and the flow—from database insertion, AI prioritization, to map plotting and escalation email—was validated .

7.3.4 User Acceptance Testing (UAT)

Conducted with a sample group of users (students/admins) to simulate how real users would interact with CleanSweep. Feedback was collected and minor UI/UX refinements were made .

7.3.5 Performance Testing

Simulated multiple concurrent complaints to evaluate response time and server load handling. The system maintained integrity under expected load thresholds .

7.4 Test Case Table

Test Case ID	Feature	Test Scenario	Input Data	Expected Output	Status
TC001	Complaint Submission	User submits a complaint with valid details	Title, Description, Category, Area	Complaint successfully saved in DB and visible on dashboard	Pass
TC002	Field Validation	User submits complaint with missing required fields	Empty title or category	Error message shown: "Required fields missing"	Pass
TC003	AI Priority Assignment	Complaint text is processed by AI for priority sorting	"Sewage overflow, foul smell, urgent action needed"	Assigned priority: **High**	Pass
TC004	Area-wise Filtering (Admin)	Admin filters complaints for a selected area	Area = "Zone B"	Dashboard shows only complaints from Zone B	Pass
TC005	Auto-Escalation Trigger	Complaint remains unresolved past threshold time	Complaint priority = High, 24+ hours pass	Auto-email sent to higher authority with complaint details	Pass

Test Case ID	Feature	Test Scenario	Input Data	Expected Output	Status
TC006	Email Notification	New complaint is submitted	Valid complaint data	Confirmation email sent to user	Pass
TC007	Complaint Mapping	Complaint is submitted with location details	Latitude = 28.61, Longitude = 77.23	Complaint plotted on map	Pass
TC008	Admin Status Update	Officer changes status of a complaint to "In Progress"	Complaint ID, new status = "In Progress"	Complaint record updated; timestamp logged	Pass
TC009	Feedback Submission	User submits feedback after complaint is marked resolved	Rating = 4/5, comment = "Good response"	Feedback saved and visible to admin	Pass
TC010	Security & Access Control	Unauthorized user attempts to access admin dashboard	No auth token / invalid user role	Access denied / redirected to login	Pass

7.5 Result of Testing

All major test cases passed successfully. AI-priority assignments were validated against manual benchmarks, and escalation triggers fired accurately. Frontend forms validated data correctly, while backend APIs responded securely and accurately .

CHAPTER 8

Chapter 8 : Conclusion

The **CleanSweep – Smart Waste Complaint & Management System** has successfully solved several issues in the traditional public waste complaint and monitoring process. By using the latest technologies like **React**, **Node.js**, **MongoDB**, **CesiumJS**, and **AI**, the platform delivers a **digital and smart solution** that benefits both **citizens** and **municipal authorities .**

Key features such as **real-time complaint tracking**, **AI-based complaint prioritization**, and **3D mapping with CesiumJS** make it easy to handle complaints efficiently and transparently. The system **reduces paperwork**, **minimizes human error**, and **improves public service quality .**

8.1 Key Achievements

8.1.1 Faster and Transparent Complaint Handling

- Automation helped reduce delays in complaint processing and ensured proper tracking of each issue .
- Citizens can easily track their complaints in real-time with status updates .

8.1.2 Improved Efficiency for Admin

- Admin dashboard allows authorities to **view, filter, and prioritize complaints** quickly .
- Status-based filters and metadata help in making faster decisions .

8.1.3 Advanced Mapping and Geolocation

- With **CesiumJS 3D Maps**, complaints are shown visually with location and current status, improving awareness and planning .
- Real-time updates with color-coded markers help manage workload better .

8.1.4 AI Based Prioritization

- A Python model analyzes complaints and identifies the most urgent one based on category and severity .
- This helps in focusing efforts on the most critical complaints first .

8.1.5 Scalability and Flexibility

- The modular design and use of open-source tools allow future upgrades like analytics, push notifications, and heatmaps .

8.2 Future Scope

8.2.1 Push Notitification for updates

- Notify users via SMS or app when their complaint status is updated (e.g., In Progress, Completed).

8.2.2 Complaint Heatmaps

- Visualize complaint density in different areas to help municipal teams plan cleaning schedules better .

8.2.3 Advanced AI Features

- Use Natural Language Processing (NLP) to auto-categorize and tag complaints for even faster processing .

8.2.4 Progressive Web App (PWA)

- Make the system mobile-friendly for use in rural or low-internet areas .

8.2.5 Citizen Engagement and Feedback Analysis

- Analyze public feedback to measure satisfaction and improve services further .

8.3 Final Remarks

CleanSweep shows how modern technology can solve real-world civic problems like **waste management** and **public complaint handling**. It creates a bridge between citizens and authorities using smart tools like **AI**, **interactive maps**, and **automated workflows .** This project proves that with the right design and tools, we can make public services **faster**, **smarter**, and **more accountable**. CleanSweep can become a **model for smart municipal systems** and be extended to other areas like **road maintenance**, **streetlight issues**, or **water supply tracking** in the future .

References

Deshmukh, D., & Shiravale, S. (2018). Priority-based sentiment analysis of citizen complaints. In Proceedings of the 2018 International Conference on Convergence in Technology (I2CT) .

Kumar, R. (2022). Full-Stack Web Development with MongoDB, Express, React, and Node. Packt Publishing.

Jeon, W., Kim, J., & Lee, Y. (2024). A data-driven decision-making support method for priority determination for an intelligent road problem reporting system. Applied Sciences, 14(23), 10861.

Graeff, E. (2018). Civic Technology Empowerment Models. MIT Media Lab. [Online]. Available: https://civic.mit.edu

Eghan, A. (2024). Enhancing Complaints Management with Artificial Intelligence. Civica Insights. Available: https://www.civica.com

FixMyStreet. (2022). Citizen reporting platform overview.Available: https://www.fixmystreet.com

SeeClickFix. (2023). Civic engagement platform.Available: https://seeclickfix.com

OpenAI (2024). GPT models for municipal applications.Available: https://platform.openai.com/docs

CesiumJS (2023). 3D geospatial visualization for web.Available: https://cesium.com

MongoDB, Inc. (2023). The MongoDB Developer Hub. Available: https://www.mongodb.com/developer

React.js Documentation (2023). Building user interfaces in React. Meta. Available: https://react.dev

Vercel Documentation (2024). Deploying full-stack applications. Available: https://vercel.com/docs

Postman Labs (2023). Automated API testing and monitoring. Available: https://www.postman.com