VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgavi-590018

A PROJECT REPORT

On

"Smart Waste Reporting and Management using MERN"

A Dissertation Submitted in partial fulfillment of the requirement for the degree of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE & ENGINEERING

Submitted by

Abhishek	**(1RG22CS003)**
Abhishek H	**(1RG22CS004)**
Adarsha S R	**(1RG22CS006)**
Dinakar D B	**(1RG22CS030)**

Under The Guidance of

Mrs. Bhagyashri Wakde

Assistant Professor, Dept. of CSE

RGIT, Bengaluru-32

Department of Computer Science & Engineering

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

Cholanagar, R.T. Nagar Post, Bengaluru-560032

2025-2026

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University)

Cholanagar, R.T. Nagar Post, Bengaluru-560032

Department of Computer Science & Engineering

CERTIFICATE

Phase-II

This is to certify that the Project Report titled **"Smart Waste Reporting and Management using MERN"** is a bonafide work carried out by **Mr. Abhishek (USN 1RG22CS003), Mr. Abhishek H (USN 1RG22CS004), Mr. Adarsha S R (USN 1RG22CS006) and Mr. Dinakar D B (USN 1RG22CS030)** in partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of the **Visvesvaraya Technological University, Belagavi**, during the year **2025-2026.** It is certified that all corrections/suggestions given for Internal Assessment have been incorporated in the report. This project report has been approved as it satisfies the academic requirements in respect of project work (15CSP85) prescribed for the said degree.

Signature of Guide	Signature of HOD	Signature of Principal
Mrs. Bhagyashri Wakde	**Dr. Arudra A**	**Dr. D G Anand**
Assistant Professor	Head Of Department	Principal
Dept. of CSE, RGIT, Bengaluru	Dept. of CSE, RGIT, Bengaluru	RGIT, Bengaluru

External Viva

Name of the Examiners	**Signature with date**
1.	
2.	

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgavi-590018

RAJIV GANDHI INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

DECLARATION

We hereby declare that the project work entitled **"Smart Waste Reporting and Management using MERN"** submitted to the **Visvesvaraya Technological University, Belagavi** during the academic year **2025-2026**, is record of an original work done by us under the guidance of **Mrs. Bhagyashri Wakde,** Assistant Professor, Department of Computer Science and Engineering, RGIT, Bengaluru in the partial fulfillment of requirements for the award of the degree of **Bachelor of Engineering** in **Computer Science & Engineering.** The results embodied in this project have not been submitted to any other University or Institute for award of any degree or diploma.

Abhishek	**(1RG22CS003)**
Abhishek H	**(1RG22CS004)**
Adarsha S R	**(1RG22CS006)**
Dinakar D B	**(1RG22CS030)**

ACKNOWLEDGEMENT

We take this opportunity to thank our college **Rajiv Gandhi Institute of Technology, Bengaluru** for providing us with an opportunity to carryout this project work.

We express our gratitude to **Dr. D G Anand**, Rector, RGIT, Bengaluru to Principal, RGIT, Bengaluru for providing the resources and support without which the completion of this project would have been a difficult task.

We extend our sincere thanks to **Dr. Arudra**, Associate Professor and Head, Department of Computer Science and Engineering, RGIT, Bengaluru, for being a pillar of support and encouraging us in the face of all adversities.

We would like to acknowledge the thorough guidance and support extended towards us by **Mrs. Bhagyashri Wakde,** Assistant Professor, Dept. of CSE, RGIT, and **Dr. Latha P H,** Associate Professor, Dept. of CSE, RGIT, Bengaluru. Their incessant encouragement and valuable technical support have been of immense help. Their guidance gave us the environment to enhance our knowledge and skills and to reach the pinnacle with sheer determination, dedication and hard work.

We also want to extend our thanks to the entire faculty and support staff of the Department of Computer Science and Engineering, RGIT, Bengaluru, who have encouraged us throughout the course of the Bachelor's Degree. We want to thank our family for always being there with full support and for providing us with a safe haven to conduct and complete our project. We are ever grateful to them for helping us in these stressful times. Lastly, we want to acknowledge all the helpful insights given to us by all our friends during the course of this project.

Abhishek	**(1RG22CS003)**
Abhishek H	**(1RG22CS004)**
Adarsha S R	**(1RG22CS006)**
Dinakar D B	**(1RG22CS030)**

ABSTRACT

Smart cities integrate multiple web and mobile solutions to create comfortable and sustainable human habitats. One of the key components of such cities is an environmentally friendly, efficient, and effective garbage management system. The existing garbage collection process, which relies on routine truck rounds conducted daily or weekly, often fails to cover all zones of the city and leads to inefficient utilization of government resources. The proposed system introduces an online-based smart waste management solution that enables real-time monitoring and analysis of waste collection activities. In this system, an admin can manage bins, drivers, user complaints, and driver work reports through a centralized web application. The system aims to provide a cost-effective, intelligent, and resource-optimized solution for managing the large volume of garbage generated daily while improving convenience for citizens. Drivers are guided by predictive routes generated through the system for efficient garbage collection, and they update the status of their work in real time. The entire platform is developed using the MERN (MongoDB, Express, React, Node.js) stack, with a React.js-based web application designed for both the workforce and the public. Users can easily report complaints and interact with smart bins, contributing to a cleaner and smarter city environment.

CONTENTS

LIST OF FIGURES / ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Overview

Proper waste management is a basic requirement in any kind of an environment. Usually cleaning in these environments are done in the morning and the afternoon. If you take an urban city like Colombo usually there are about 1,200,000 to 1,500,000 employees heading for their workstations every morning. For all those people, there are just not enough garbage bins available. On the streets of urban cities, hundreds of people are passing the same location around one minute.

Around 95% of people are carrying food covers, polythene bags, and plastic bottles. If they dispose all them at once, the bins will be filled in several minutes. When they fill up people just litter their trash around the garbage bins because there is nowhere else to put them. The obvious solution to this is for the cleaning staff to stay near garbage bins every day till they fill up to clean them. This is not a real solution. It takes way more cleaning staff and costs a lot of money. So, it is impractical. This is simply not enough. There are some notable negative effects when considering the garbage bins always being full.

One of the main effects is the surrounding area starting to smell and be very unpleasant. When the garbage bins are full people put their trash on sides of the garbage bins. When this is done for some time, first it starts to smell bad. So, others who come later tend not to go close and throw their trash in the direction of the garbage bins. If there are any leftover food items, throwing it causes them to spill. This attracts animals like cats, dogs, and flies. And these animals spill them even more. Another negative effect is the diseases that spread. It's not just the garbage that spread them, but the animals also can be a source. To address these challenges, smart waste management solutions can be implemented to optimize the collection process and reduce environmental impact. One effective approach is to introduce sensor-enabled smart bins, which can detect when they are full and notify waste collection teams in real-time. This ensures that bins are emptied before they overflow, minimizing littering and unpleasant odors. Additionally, data analytics can be used to track waste generation patterns, allowing for more efficient scheduling of cleaning staff based on peak usage times and areas with high foot traffic. Integrating mobile apps for employees and residents to report overfull bins can also enhance responsiveness.

1.2 Motivation

The increasing population and urbanization in cities have resulted in a tremendous rise in waste generation, leading to challenges in maintaining public hygiene and cleanliness. Traditional waste management systems often rely on fixed schedules and manual monitoring, which are inefficient and time-consuming. Overflowing garbage bins, foul odors, and unhygienic surroundings have become common problems that affect both the environment and human health.

The motivation behind this project arises from the need to create an intelligent, automated, and cost-effective system that ensures timely waste collection and better coordination between authorities and citizens. With the help of digital tools and real-time updates, this project aims to transform the conventional garbage collection process into a smart, data- driven operation that minimizes human effort while maximizing efficiency and cleanliness. This project is motivated by the vision of **creating cleaner, greener, and smarter cities** through the integration of web-based technology and data-driven decision-making. By adopting a digital approach, the system aims to not only improve efficiency but also promote environmental awareness, public participation, and sustainable urban development for future generations.

1.3 Problem Identification

The current waste management system faces several limitations, including lack of real-time monitoring, delayed garbage pickup, and poor coordination between different departments. Overflowing bins not only cause bad odor and attract animals but also create serious health hazards for residents. Manual collection routes are often inefficient, leading to unnecessary fuel consumption and labor costs. Citizens have no structured way to report unclean areas or track complaint progress. These issues highlight the urgent need for a system that offers automation, transparency, and data-based decision-making. Therefore, this project identifies the main problems as inefficient resource allocation, lack of communication, and absence of a digital platform for waste management, all of which can be resolved through a smart web-based application.

These recurring problems highlight the urgent need for a digital, data-driven, and location-based waste management system that can automate key processes, ensure real-time communication, and optimize resource allocation. The proposed system addresses these gaps by using the MERN technology stack to provide an integrated platform for administrators, drivers, and citizens, enabling transparency, timely garbage collection, and efficient city- wide waste management.

1.4 Scope

The scope of this project is to design and develop a **smart waste reporting and management system** that effectively connects three primary users — administrators, drivers, and the general public — through a single online platform. The system enables administrators to monitor bins, manage driver routes, and track reports; drivers to receive assigned tasks and update the completion status; and the public to raise complaints and view their updates. This project can be further extended by integrating IoT-based sensors for automatic garbage level detection and AI-based route optimization.

Its adaptability ensures that it can be implemented not only in urban cities but also in smaller towns aiming to adopt smart city technologies. Overall, the system's scope contributes to sustainable waste management by ensuring real-time monitoring, efficient collection, and cleaner environments. In summary, the scope of this project goes beyond just digitalizing the waste collection process—it aims to establish a **smart, interconnected ecosystem** where technology bridges the gap between citizens and authorities. By leveraging the MERN stack, the system ensures real-time communication, transparency, and efficient management of resources, contributing to cleaner cities and improved living standards.

1.5 Objective and Methodology

The main objective of this project is to create a digital waste management system that improves operational efficiency, enhances public participation, and reduces the burden on municipal authorities. The system aims to automate key processes such as complaint registration, route management, and bin monitoring using an online platform. The methodology involves the use of the **MERN stack**—React.js for developing an interactive user interface, Node.js and Express.js for server-side logic, and MongoDB for secure and scalable data storage. The software development process follows standard stages, including requirement analysis, design using UML and data flow diagrams, coding, and testing. The design focuses on modular structure and ease of use, while the testing phase ensures the system's accuracy, performance, and reliability.

This methodological approach guarantees a well-structured, efficient, and user-friendly waste management solution. Overall, this methodology ensures that the final product is robust, user-friendly, and scalable, capable of being deployed in both small municipalities and large urban areas. By leveraging the MERN stack's efficiency, this project not only provides a technological solution for waste management but also establishes a foundation for future integration with IoT and AI to enable predictive analytics and smart city automation.

1.6 Outcome

The developed system successfully demonstrates a real-time, web-based solution for effective garbage collection and monitoring in urban areas. It allows administrators to manage waste collection activities from a central dashboard, drivers to follow optimized routes, and citizens to actively participate in maintaining city hygiene by reporting unclean areas. The system significantly reduces manual intervention, enhances operational efficiency, and promotes transparency between the public and authorities. It achieves high accuracy, data security, and user satisfaction, leading to better cleanliness and environmental quality. The project fulfills its goal of providing a sustainable and cost-effective waste management system that aligns with the concept of smart cities. In the future, this system can be extended by integrating IoT devices, sensors, and AI analytics to make it more intelligent and adaptive.

1.7 Introduction Summary

In today’s rapidly urbanizing world, efficient waste management is one of the major challenges faced by cities. Traditional waste collection systems are outdated, relying heavily on manual monitoring and fixed schedules that do not adapt to real-time needs. The **Smart Waste Reporting and Management System using MERN React.js** offers a digital solution by connecting citizens, drivers, and administrators through a single online platform. This system enhances waste collection efficiency through route optimization, real-time updates, and data analysis. It improves public hygiene, reduces government expenses, and promotes eco-friendly practices. By integrating technology with urban services, the project contributes to the vision of creating sustainable and smart cities where cleanliness and community involvement go hand in hand.

CHAPTER 2

LITERATURE REVIEW

2.1 Related Work

The paper by S. Lokuliyana et al. on “IGOE IoT Framework for Waste Collection Optimization” presents a complete IoT-based framework for efficient waste collection in smart cities. It uses a sensor network deployed at disposal sites to detect waste levels and notify authorities about collection needs. A mobile application enables citizens to report overflow at authorized sites or illegal dumping, while authorities can communicate alerts and updates back to the public.

Optimization algorithms are applied to determine the most efficient truck collection routes, and the system performance is analyzed in terms of delay in waste collection, effective collection rate, and overall process efficiency. In contrast, R. Fujdiak et al., in “Using Genetic Algorithms for Advanced Municipal Waste Collection in Smart City,” focus on applying genetic algorithms to optimize waste collection logistics.

Their approach leverages IoT to collect massive data and uses techniques such as Floyd-Warshall, Dijkstra, TSP formulation, crossover, and mutation to calculate more efficient garbage-truck routes. The implementation is performed within an open-source integrated simulation framework to facilitate future modifications, with simulations demonstrating improvements in route efficiency.

Finally, T. Anagnostopoulos et al., in “Top-k Query based Dynamic Scheduling for IoT-enabled Smart City Waste Collection,” propose a dynamic scheduling model using top-k queries on real-time sensor data to address near real-time waste collection challenges. The city is divided into multiple sectors, each containing intermediate waste depots, and a heterogeneous fleet of trucks serves these depots.

Cloud middleware aggregates and cleans sensor data to feed a scheduling engine implemented in OpenIoT, which dynamically assigns the nearest available truck to bins based on top-k queries. A spatial database stores real-time bin information, and an Android app provides drivers with a user-friendly GUI to interact with the system. Together, these studies highlight various IoT-driven strategies—framework-based optimization, genetic algorithm routing, and top-k query dynamic scheduling—for improving waste collection efficiency in smart cities.

2.2 Existing System

In the current waste management system, employees and citizens heading to their workstations or passing through urban streets often encounter insufficient garbage bins. Hundreds of people may pass the same location within a short period, typically around a minute, leading to bins quickly reaching full capacity.

The conventional approach is for cleaning staff to stay near garbage bins daily and empty them once full. However, this method is inefficient and impractical, as it requires constant monitoring and manual intervention.

Several negative effects arise when garbage bins remain full:

- Surrounding areas become unpleasant and unhygienic, often producing bad odors.
- People frequently place trash beside overflowing bins, creating additional litter.

Disadvantages of the existing system include:

- Time-consuming and less effective operations: Trucks often empty containers regardless of whether they are full or not.
- High operational costs due to unnecessary trips and manpower.
- Unhygienic environment and a negative impact on the city's appearance.
- Health hazards: Bad smells can spread disease or cause discomfort.
- Increased traffic and noise pollution due to frequent garbage collection.

2.2 Proposed System

The proposed system introduces an IoT-enabled smart waste management solution aimed at improving the efficiency of garbage collection and overall city hygiene. Solid waste management in this system is broadly categorized into segregation, collection, and transportation.

- Data Collection and Storage: Sensors installed in dustbins will monitor fill levels in real-time and send the data to a central server, which stores it in a database.
- Data Analytics and Dashboards: The collected data is analysed and displayed on two separate dashboards—one for the workforce (garbage collectors) and one for clients (citizens or administrators). Reports generated through data analytics can be monitored by administrators via the admin dashboard.
- Optimized Routing: Based on real-time data, garbage trucks are assigned efficient routes using routing algorithms integrated with Google Maps API, ensuring that all necessary bins are serviced before reaching the dumping site.

Advantages of the Proposed System:

- Provides real-time information on the fill levels of dustbins.

- Allows deployment of dustbins based on actual needs, avoiding unnecessary placements.
- Reduces costs and optimizes resources by eliminating redundant collection trips.
- Improves environmental quality, reduces odors, and maintains cleaner cities.
- Enables intelligent management of municipal waste services.
- Ensures effective usage of dustbins, preventing overflow and littering.

2.3 Software Requirements

Software Stack (MERN):

- **Front-End:** React JS, CSS3, Bootstrap
- **Back-End:** Node JS, Express JS
- **Database:** MongoDB

Development Tools:

- **IDE:** Visual Studio Code
- **Database Management:** MongoDB

2.4 Hardware Requirements

- **Processor:** Intel Core i3 2.10 GHz
- **Installed Memory (RAM):** 4 GB
- **Hard Disk:** 160 GB
- **Operating System:** Windows 7 / 10

2.5 Literature Review Summary

The literature review highlights various research efforts and technological advancements aimed at improving urban waste management through smart solutions. Traditional waste collection methods are often inefficient, leading to delayed collection, overflowing bins, and increased pollution. Several studies have proposed using **Internet of Things (IoT)** technologies, **Machine Learning (ML)**, and **Artificial Intelligence (AI)** to automate waste monitoring and optimize collection processes.

Key works reviewed include systems that use **smart bins with sensors** to monitor fill levels, **GPS-enabled vehicles** for optimized route planning, and **data analytics** for predicting waste generation trends. Some models integrate **mobile applications** to enable citizens to report waste-related issues, enhancing civic participation and transparency.

CHAPTER 3

OVERALL DESCRIPTION OF THE PROPOSED SYSTEM

3.1 Module Description

The system enables communication between the general public, workers, and administrators using real-time garbage update information. Citizens can report issues or overflowing bins to the administration, while workers and administrators can coordinate effectively to ensure timely waste collection. This interconnected communication improves responsiveness, efficiency, and overall management of the city's waste collection system.

3.2 System Features

In the life cycle of software development, problem analysis provides the foundation for the design and development phases. The purpose of analyzing the problem is to gather sufficient information to design an effective and efficient system. Large problems are typically sub-divided into smaller, manageable components, making them easier to understand and solve. Similarly, in this project, all tasks have been carefully categorized and divided into smaller modules to facilitate systematic development and implementation.

System Modules:

- **Administrator**
  a. Login
  b. Create Garbage bin
  c. Update/Delete garbage bin
  d. Assign best route for drivers
  e. Manage driver
  f. View Garbage Report
  g. View complaints from public
- **General Public**
  a. Register
  b. Login
  c. Register complaint
  d. My complaint & status
- **Driver**
  a. Login
  b. Check daily work updates
  c. Choose best route
  d. Update garbage load

Modules Description:

- **Administrator**
  e. Login
     i. In log in page, admin can manage all information. They can update or edit any information.
  f. Create Garbage bin
     i. Admin can create the garbage available in the city with details and location
  g. Update/Delete garbage bin
     i. Admin can View and update the garbage available in the city with details and location
  h. Assign best route for drivers
     i. Admin can update the garbage best route to reach the bin of the city quickly
  i. Manage driver
     i. Admin can Create & View and update the driver login details
  j. View Garbage Report
     i. Admin can View work report of garbage all updates from the driver
  k. View complaints from public
     i. Admin can View the complaint and update status of the complaints
- **General Public**
  l. Register
     i. User can register to garbage application
  m. Login
     i. User can login to garbage application
  n. Register complaint
     i. User can register to complaints about the cleaning of the garbage
  o. My complaint & status
     i. User can view the status of the complaints about the cleaning of the garbage
- **Driver**
  p. Login
     i. Drive can login to garbage application
  q. Check daily work updates
     i. Drive can update the daily work status
  r. Choose best route
     i. Drive can view the route for the garbage bins
  s. Update garbage load

Drive can update the load for the garbage bins.

CHAPTER 4

SYSTEM ANALYSIS

4.1 Introduction to System Analysis

System Analysis is a crucial phase in the software development life cycle that focuses on understanding and defining the requirements of a system. It involves examining the current system, identifying problems, and determining user needs to design an effective solution. The primary goal of system analysis is to provide a clear and structured understanding of the problem, enabling developers to create a system that is efficient, reliable, and meets user expectations.

System analysis typically includes activities such as:

- Studying the existing system to identify limitations and inefficiencies
- Gathering requirements from users, stakeholders, and other sources
- Dividing complex problems into smaller, manageable modules
- Designing a logical structure that serves as a blueprint for system development

In the context of this project, system analysis helps in planning and developing a smart waste management system, ensuring effective communication between workers, the general public, and administrators, while optimizing waste collection and resource usage.

4.2 Feasibility Study

During the **system analysis phase**, the feasibility of the project is carefully examined to ensure that the proposed system is viable and does not impose undue burden on the organization. A **business proposal** is also developed, including a general project plan and cost estimates. Understanding the major requirements of the system is essential for performing an effective feasibility analysis. The three key considerations in this phase are:

- **Economical**

  This study evaluates the economic impact of the system on the organization. The funds available for research and development are limited, so expenditures must be justified. In this project, the system development remains within budget, as most technologies used are freely available, with only certain customized products requiring purchase.

- **Technical**

  Technical feasibility ensures that the system can be implemented with the available technical resources without placing excessive demands on the client or organization.

The proposed system requires only minimal changes to existing infrastructure, making it technically feasible and easy to implement.

- **Social**

  Social feasibility assesses the **level of acceptance by users**. Users must be trained to operate the system efficiently and should view it as a helpful tool rather than a threat. Proper education, familiarization, and confidence-building ensure that users accept the system and can provide constructive feedback, improving its effectiveness.

Non-Functional Requirements

Non-functional requirements define the **quality attributes** of the system, specifying how well the system performs its functions. These attributes influence both system design and execution. The non-functional requirements for this project include:

- **Accuracy**

  The system will be accurate and reliable based on its design. In case of any discrepancies, alternative methods will be provided to solve the problem.

- **Usability**

  The system will be simple and easy to use, providing a comfortable and intuitive interface for users to interact with the system efficiently.

- **Accessibility**

  The system will be accessible through the internet, ensuring continuous availability without known issues.

- **Performance**

  The system will deliver optimal performance while executing its intended functionalities.

- **Reliability**

  The system will be reliable under all circumstances, with mechanisms in place to handle any issues effectively.

- **Security**

  The system will be highly secure. Users will require registration with a username and password. Proper **authorization and authentication** will be enforced based on user type and permissions. The system will be designed to prevent misuse and ensure data integrity.

4.3 System Analysis Summary

The system analysis focuses on understanding the current issues in urban waste management and identifying functional and non-functional requirements for an efficient solution. Existing manual processes often lead to delayed waste collection, lack of monitoring, and poor coordination between citizens and authorities. The proposed **Smart Waste Reporting and Management System** addresses these problems by enabling users to report waste through image uploads, which are processed and forwarded to municipal authorities for quick action. The system integrates modules for user management, complaint tracking, and real-time updates, supported by a backend database for data storage and retrieval. It ensures reliability, scalability, and user-friendly interaction while minimizing operational inefficiencies and promoting cleaner, smarter urban environments.

CHAPTER 5

SYSTEM DESIGN

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer's requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data.

5.1 UML Diagrams:

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

There are various kinds of methods in software design:

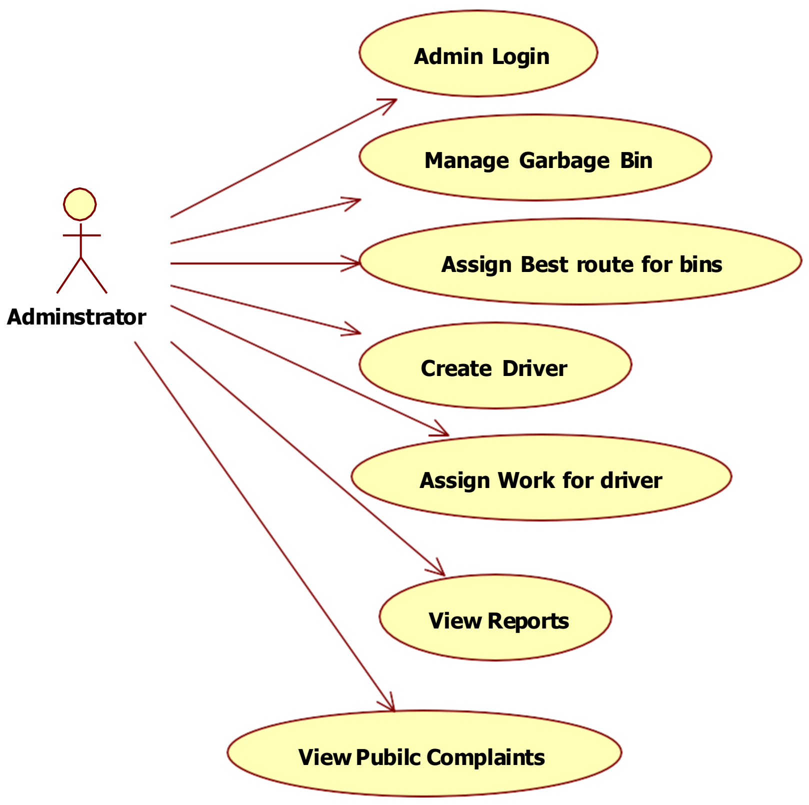
- Use case Diagram
- Sequence Diagram
- Collaboration Diagram

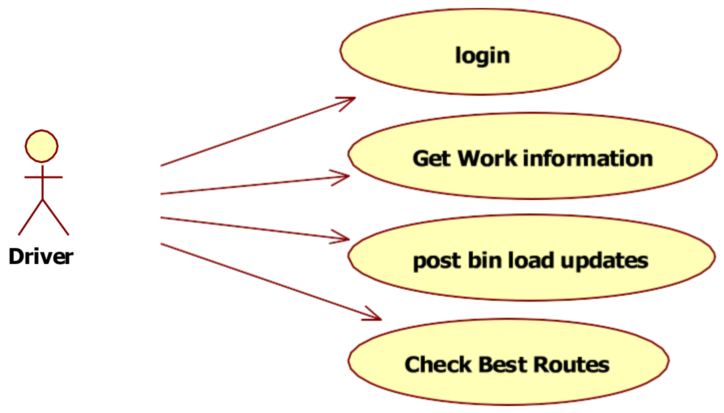
5.1.1 Use case Diagrams:

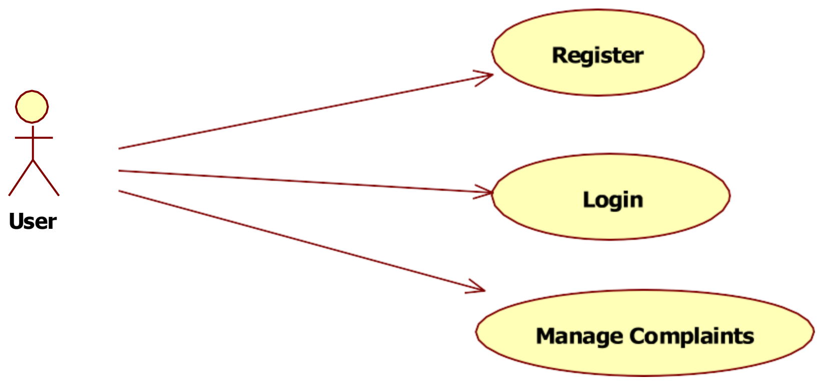
Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what's called an actor. Use case diagram can be useful for getting an overall view of the system and clarifying that can do and more importantly what they can't do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

- The purpose is to show the interactions between the use case and actor.
- To represent the system requirements from user's perspective.
- An actor could be the end-user of the system or an external system.

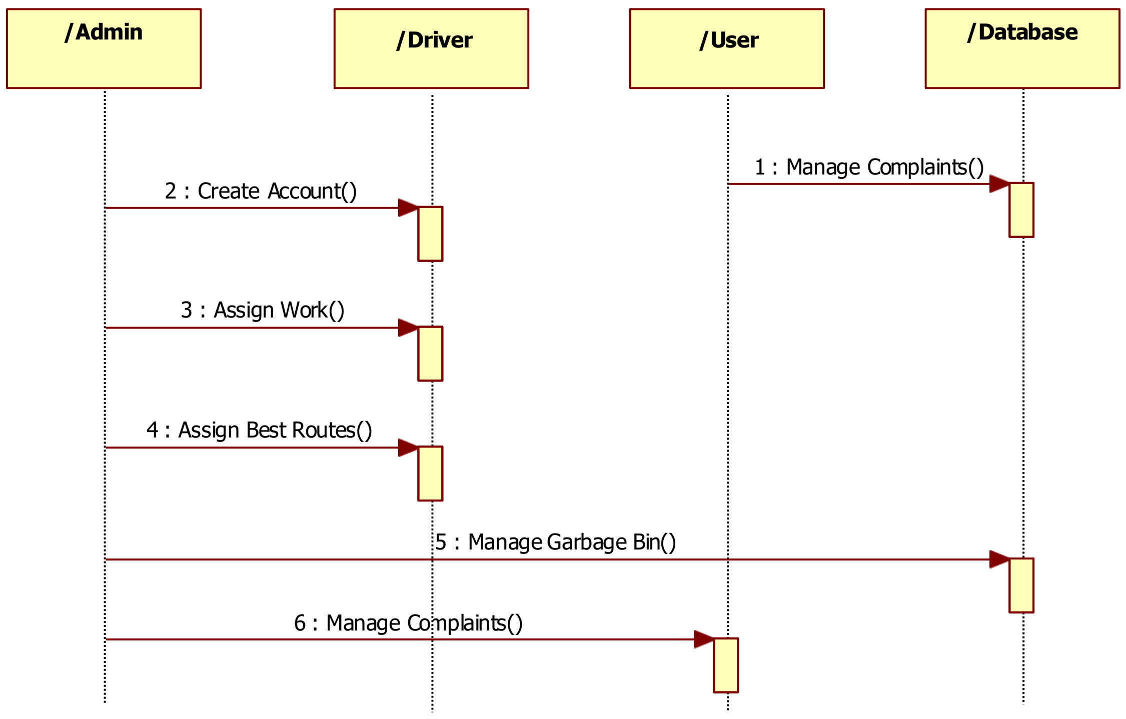

Adminstrator
Admin Login
Manage Garbage Bin
Assign Best route for bins
Create Driver
Assign Work for driver
View Reports
View Pubilc Complaints


Driver
login
Get Work information
post bin load updates
Check Best Routes


User
Register
Login
Manage Complaints

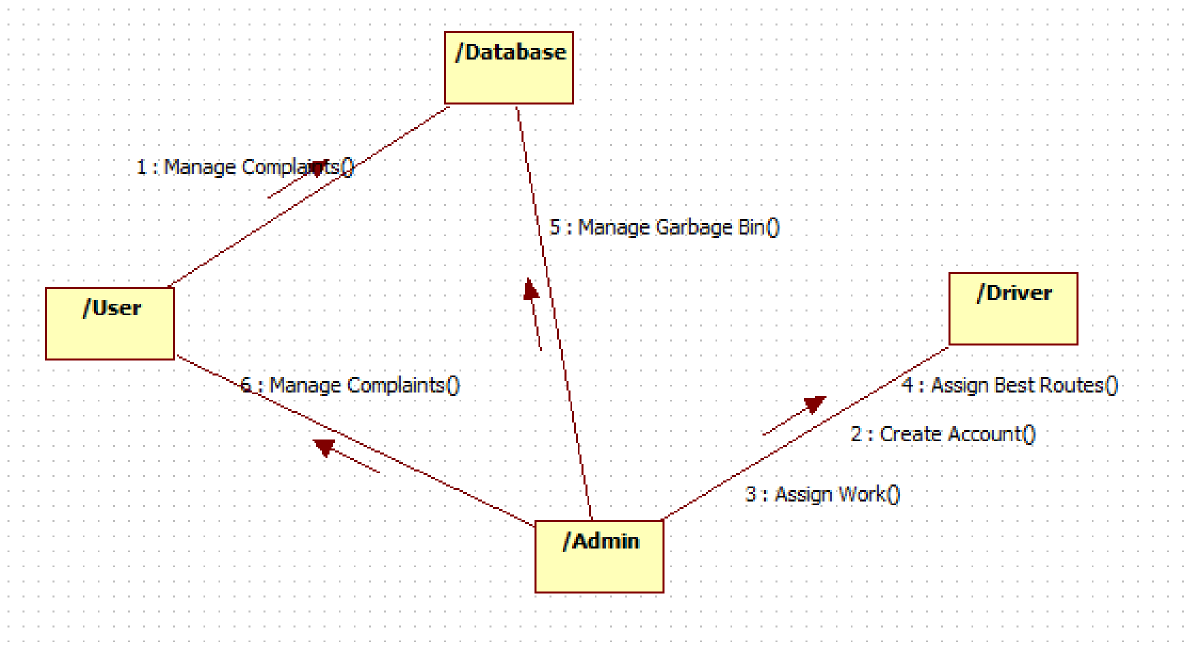
5.1.2 Sequence Diagram:

Sequence diagram and collaboration diagram are called INTERACTION DIAGRAMS. An interaction diagram shows an interaction, consisting of set of objects and their relationship including the messages that may be dispatched among them.
A sequence diagram is an introduction that empathizes the time ordering of messages. Graphically a sequence diagram is a table that shows objects arranged along the X-axis and messages ordered in increasing time along the Y-axis.

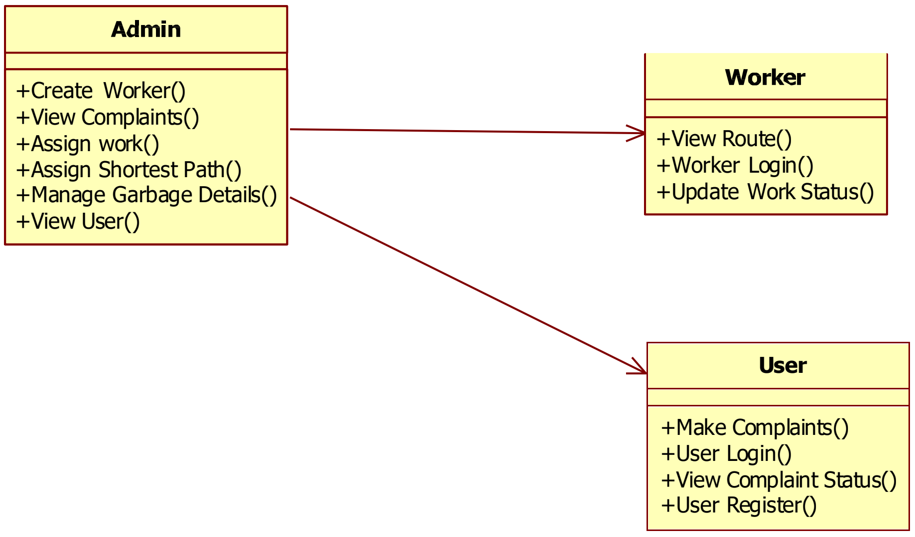


5.1.3 Collaboration Diagram:

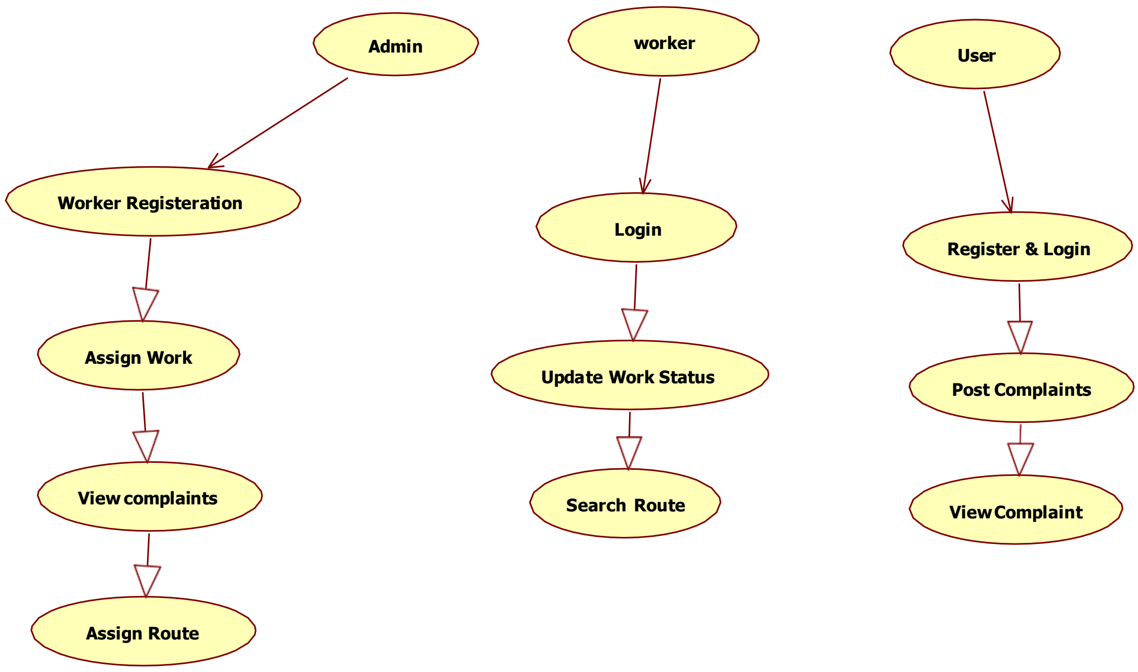
A collaboration diagram is a type of visual presentation that shows how various software objects interact with each other within an overall IT architecture and how users can benefit from this collaboration. A collaboration diagram often comes in the form of a visual chart that resembles a flow chart.



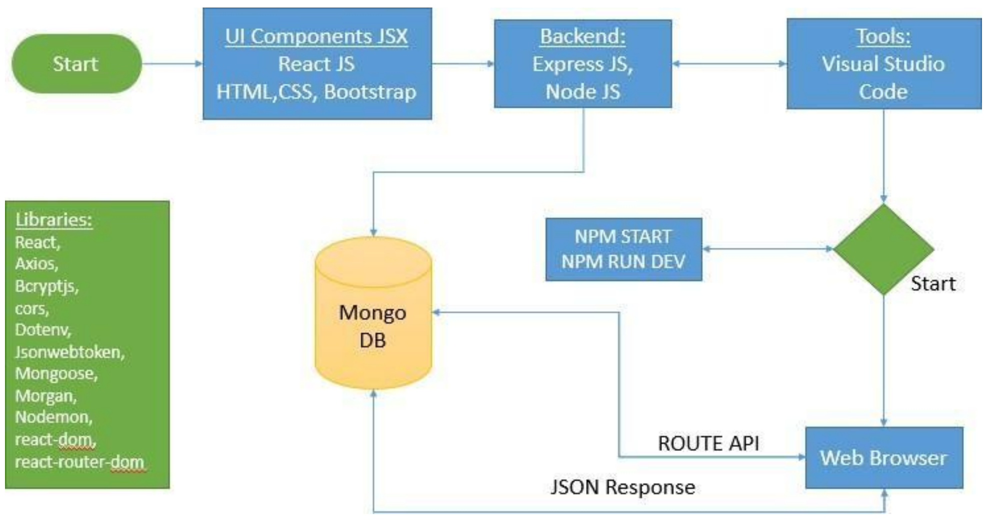
5.1.4 Class Diagram:



5.1.5 Activity Diagram:

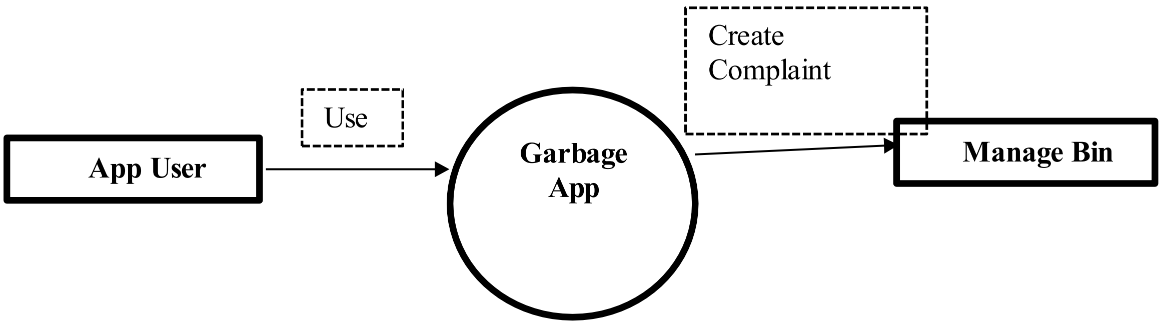


5.1.6 Work Flow Diagram:

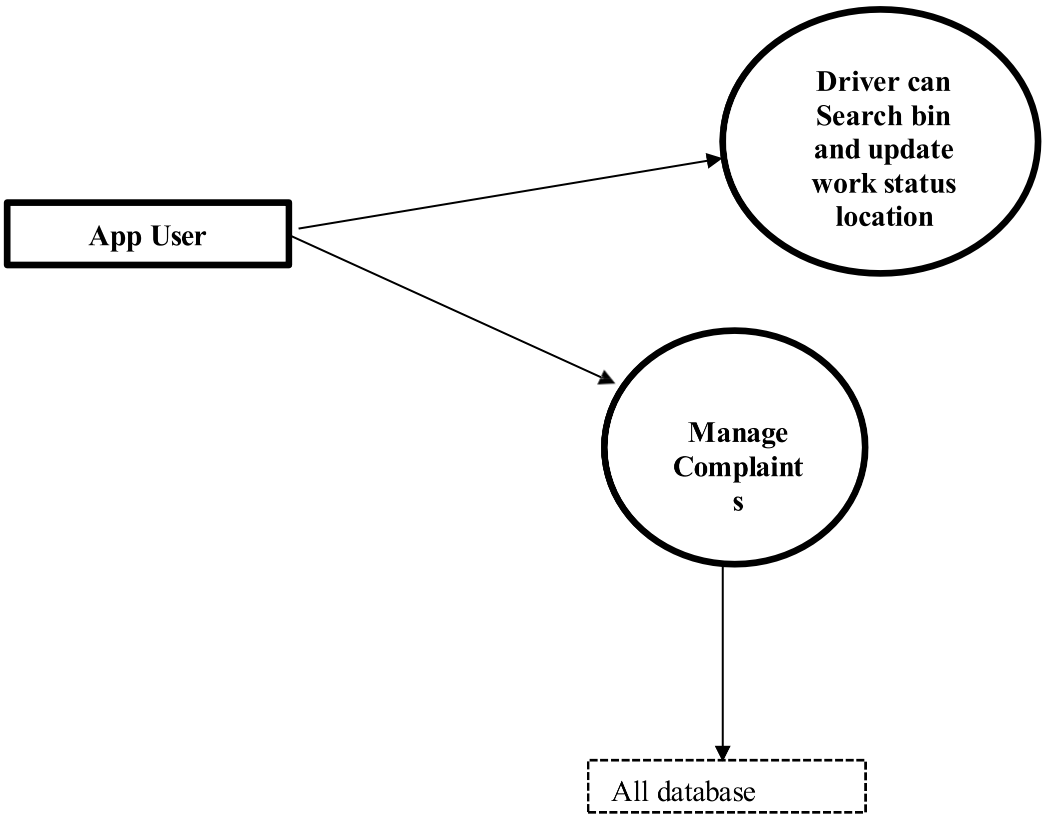


5.1.7 Data Flow Diagram:

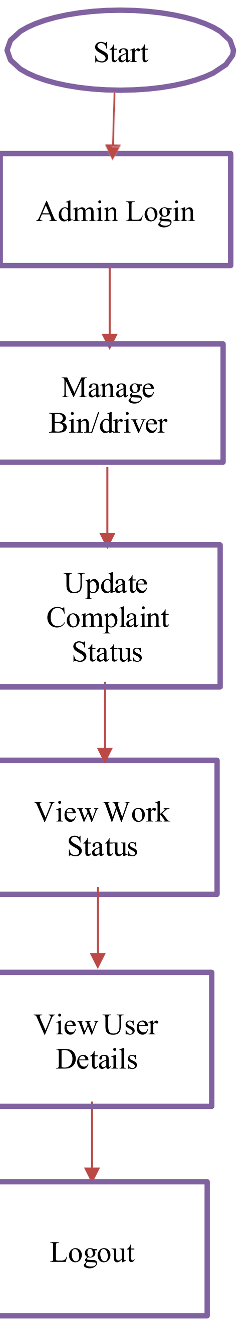
0-Level DFD



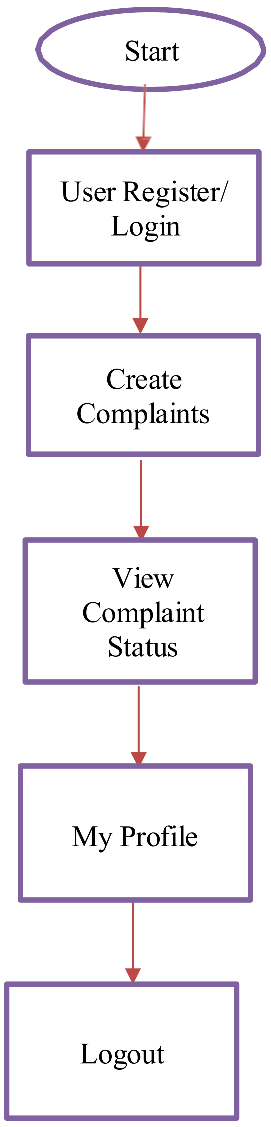
1-Level DFD



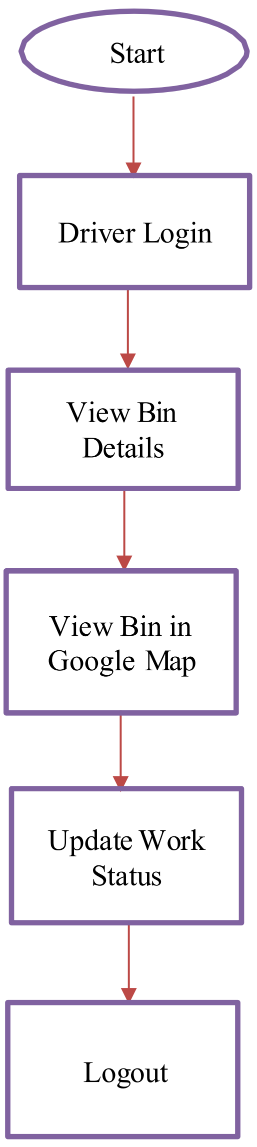
2-Level Admin DFD



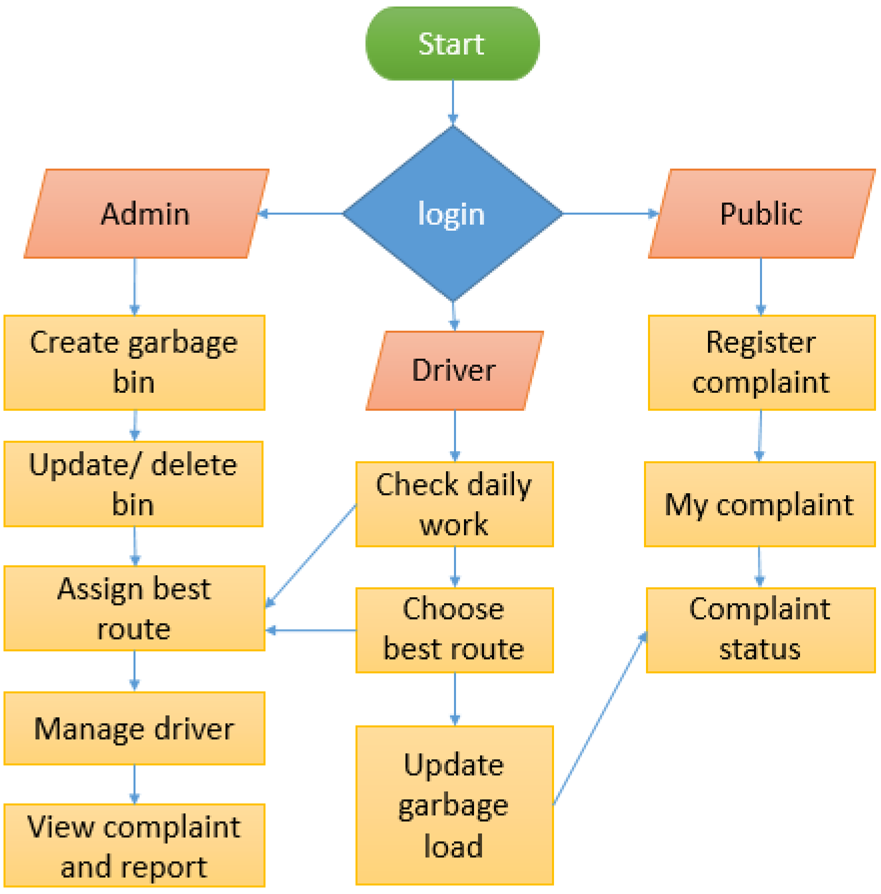
2 Level User DFD



2 Level Driver DFD



5.1.8 FLOW DIAGRAM



5.1.9 Table Design:

User & Admin & Worker Login

Id	Username	Password
Int	Varchar	Varchar
100	100	100
Primary key		

User Register

User ID	Name	Email Id	Password	Mobile	Address	City	Question 1	Question 2
Int	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar
100	100	100	100	100	100	100	100	100
Primary key								

Garbage

ID	Place	Locality	Landmark	Driver ID	Address 1	Address 2	Lat	Long
Int	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar	Varchar	Varc har
100	100	100	100	100	100	100	100	100

Primary key								

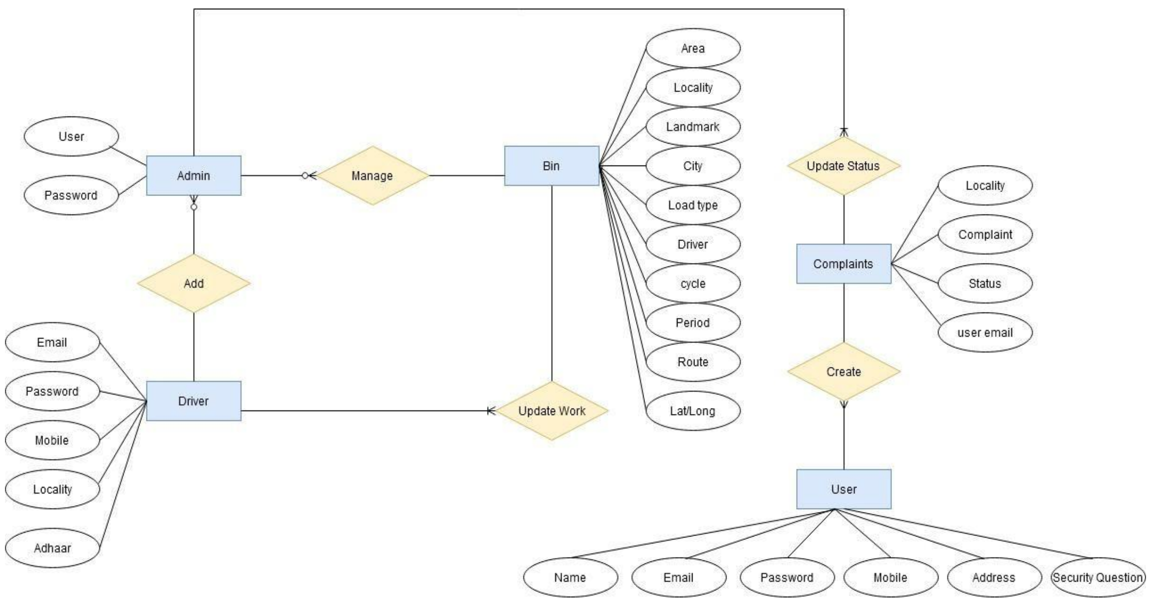
Worker Status

ID	Name	Email Id	Address	Status
Int	Varchar	Varchar	Varchar	Varchar
100	100	100	100	100
Primary key				

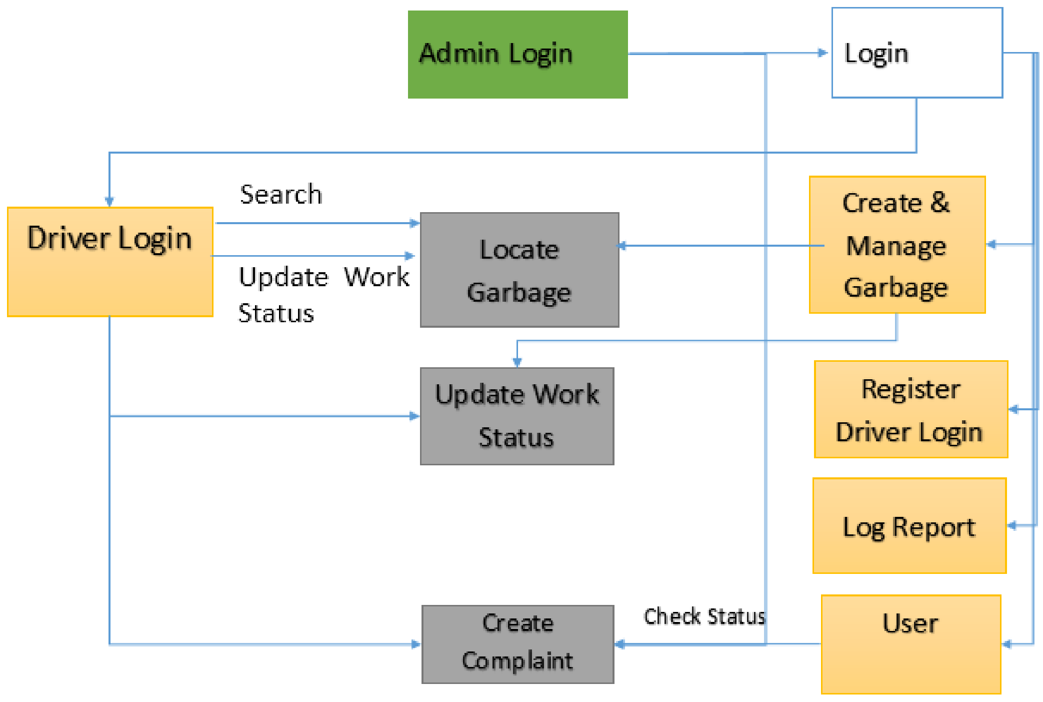
User Complaints

ID	Name	Email Id	Address	Status
Int	Varchar	Varchar	Varchar	Varchar
100	100	100	100	100
Primary key				

5.1.10 ER Diagram



5.1.11 Architecture Design:



5.1.12 RISK ANALYSIS TABLE

RISK INFORMATION SHEET
DESCRIPTION: The UI design is developed as a reuseable component. The remaining functionalities are developed based on user's performance or application with which integrated. Improper authentication or apk security led to misuse of the details present within the application.
REFINEMENT/ CONTEXT: Sub condition 1: Fleid conditions are incorporated to provide proper guidance to users. Sub condition 2: User' security are incorporated to provide invalid security purposes. Sub condition 3: UI design is not a language or domain specific. Target environment support will be difficult.
MITIGATION/ MONITORING: 1. Validation of fields such as email and password and display of error message has to

be indicated properly. 2. Monitoring the whole process which be invalid credentials to be shows on the error of invalid users 3. It will be safe on apk builds. There is no hack of application inactivity it's on fully security of apk build process. 4. Developing language specific or application specific modules may lead of rebuild some modules.
MANAGEMENT/ CONTINGENCY PLAN/ TRIGGER: When an unauthorized user tries to access the details/ services provided by the application, a warning message or alert has to be given to the user and stop the user from accessing the data.
CURRENT STATUS: validations are verified.

5.1.13 PERFORMANCE CHART

- Number of Test case Scenario – 03
- Number of Test case Scenario Tested – 16
- Number of Test case Scenario Passed – 16
- Efficiency – 100%

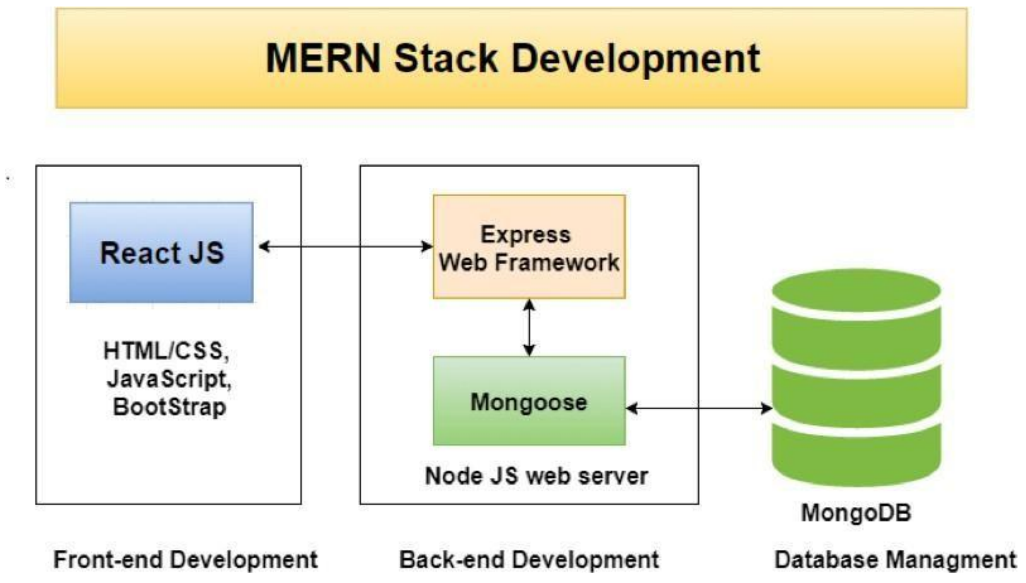
5.2 System Design Summary

The system design defines the overall architecture and interaction between various modules of the **Smart Waste Reporting and Management System**. It follows a **three-tier architecture** comprising the frontend, backend, and database layers. The **frontend**, built using React.js, provides an interactive user interface for citizens to register complaints and track their status. The **backend**, developed with Node.js and Express.js, handles business logic, routing, and communication between the client and server. The **database**, implemented using MongoDB, stores user details, complaint records, and system logs efficiently. The design also incorporates RESTful APIs for seamless data exchange, ensuring scalability, modularity, and secure access. Additionally, the system supports real-time updates, automated complaint processing, and integration with authorities for faster response, providing a robust and intelligent solution for urban waste management.

CHAPTER 6

IMPLEMENTATION DETAILS

6.1 MERN:



6.2 React.js (Frontend Development)

React.js is a powerful JavaScript library developed by Facebook for building dynamic, interactive, and scalable user interfaces for web applications.

Key Features:

- **Declarative UI:** Simplifies the process of designing interfaces by defining what the UI should look like based on the application state.
- **Component-Based Architecture:** Divides the UI into reusable, independent components for improved scalability and maintenance.
- **Virtual DOM:** Optimizes rendering by updating only the parts of the DOM that change.
- **JSX Syntax:** Enables embedding HTML-like code directly within JavaScript for better readability and integration of logic with layout.
- **Unidirectional Data Flow:** "Props down, events up" design ensures predictable data flow and easier debugging.
- **Rich Ecosystem:** Libraries such as React Router, Redux, Axios, and Material-UI enhance functionality and streamline development.
- **Server-Side Rendering (SSR) & Static Site Generation (SSG):** Improves performance and SEO optimization.

- **Developer Tools:** React DevTools provide easy inspection of component hierarchy, props, and state.
- **Cross-Platform Support:** React Native and Electron enable reuse of code for mobile and desktop applications.
- **Continuous Improvement:** Actively maintained with frequent updates and feature enhancements.
- **Backend Integration:** Communicates with Node.js/Express backend through HTTP requests for CRUD operations.
- **Development Server:** The command npm run dev provides live reloading, debugging, and real-time updates during development.

6.3 Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) is a styling language used to control the visual presentation of HTML or XML content.

Key Features:

- **Separation of Content and Presentation:** HTML defines the structure, while CSS defines the styling and layout.
- **Reusability:** Styles can be applied across multiple web pages using external .css files.
- **Responsive Design:** Adapts layouts for different screen sizes and devices, improving accessibility.
- **Flexibility:** Supports multiple media types including screen, print, and voice.
- **Cascade and Specificity:** Determines the priority of style rules when multiple styles apply to an element.
- **Rapid Updates:** A single modification in a CSS file can affect multiple pages, saving time and effort.

6.4 Bootstrap

Bootstrap is a popular front-end framework used to develop responsive and visually consistent web interfaces.

Key Features:

- **Responsive Grid System:** Enables flexible layouts using a row and column structure.
- **Pre-Styled Components:** Offers built-in components such as buttons, forms, navigation bars, and modals for faster development.
- **Utility Classes:** Provides quick styling for spacing, alignment, and typography

without custom CSS.

- **Integration with React:** Can be used via React-Bootstrap or standard CSS/JS modules.
- **Customization:** Default styles can be overridden to align with specific branding or design requirements.

6.5 Node.js Server

Node.js is a powerful JavaScript runtime environment that executes server-side code outside of a web browser.

Key Features:

- **JavaScript Runtime:** Built on Chrome's V8 engine, enabling high-speed execution of JavaScript on the server side.
- **npm (Node Package Manager):** Simplifies dependency and package management.
- **Event-Driven & Asynchronous:** Efficiently handles multiple requests without blocking operations.
- **Single-Threaded Event Loop:** Designed for non-blocking I/O operations, improving scalability.
- **Customizable Start Script:** The command npm start runs the start script defined in package.json.
- **Cross-Platform Compatibility:** Works seamlessly on Windows, macOS, and Linux.
- **Built-in Modules:** Provides modules for file systems, networking, HTTP, and path handling.
- **Scalability & Performance:** Ideal for building APIs, real-time applications, and microservices.

6.6 Express.js

Express.js is a lightweight and flexible Node.js web framework that simplifies server-side application development.

Key Features:

- **Routing:** Simplifies handling of HTTP requests (GET, POST, PUT, DELETE).
- **Middleware Architecture:** Allows modular handling of tasks like authentication, logging, and data validation.
- **Request & Response Handling:** Provides easy access to query parameters, headers, and cookies, and supports sending JSON/HTML responses.
- **Template Engine Support:** Compatible with popular template engines such as Pug, EJS, and Handlebars.

- **Error Handling:** Offers built-in mechanisms for managing errors and exceptions.
- **Extensible:** Supports third-party middleware and plugins to extend functionality.

6.7 MongoDB

MongoDB is a NoSQL, document-oriented database that stores data in a flexible, JSON-like format known as BSON.

Key Features:

- **Schema-less Design:** Allows dynamic and flexible data models without predefined structures.
- **Collections & Documents:** Analogous to tables and rows in relational databases but more flexible.
- **High Scalability:** Supports horizontal scaling through sharding.
- **Replication & High Availability:** Uses replica sets to ensure data redundancy and fault tolerance.
- **Rich Query Language & Aggregation:** Enables advanced filtering, grouping, and data transformation.
- **Geospatial & Full-Text Search:** Supports efficient location-based and keyword searches.

  Strong Community Support: Backed by a large community and an ecosystem of tools for monitoring and deployment.

6.8 Program Code

Module.js

```
const mongoose = require('mongoose');
const binSchema = mongoose.Schema({
    adminemail: {
        type: String,
        required: true,
    },
    area: {
        type: String,
        required: true,
    },
    locality: {
        type: String,
        required: true,
    },
    landmark: {
        type: String,
        required: true,
    },
    city: {
        type: String,
        required: true,
    },
    loadtype: {
        type: String,
        required: true,
    },
    driveremail: {
        type: String,
        required: true,
    },
    cycleperiod: {
        type: String,
        required: true,
    },
    bestroute: {
        type: String,
        required: true,
    },
    lat: {
        type: Number,
        default: 0,
    },
    long: {
        type: Number,
        default: 0,
    },
    dateCreated: {
        type: Date,
        default: Date.now
    }
})


binSchema.virtual('id').get(function () {
    return this._id.toHexString();
});

binSchema.set('toJSON', {
    virtuals: true,
});


exports.Bin = mongoose.model('Bin', binSchema);
```

Router.js

```
const {Bin} = require('../models/bin');
const express = require('express');
const router = express.Router();
const auth = require('../helpers/jwt');

// name description mechanicname service available  locality address city
mobile

router.get(`/`,  async (req, res) =>{
    const binList = await Bin.find();

    if(!binList) {
        res.status(500).json({success: false})
    }
    res.status(200).send(binList);
})



router.get(`/:id`, async (req, res) =>{
    const binList = await Bin.findById(req.params.id);

    if(!binList) {
        res.status(500).json({success: false})
    }
    res.send(binList);
})


router.post('/', async (req,res)=>{
    let bin = new Bin({
        adminemail: req.body.adminemail,
        area: req.body.area,
        locality: req.body.locality,
        landmark: req.body.landmark,
        city: req.body.city,
        loadtype: req.body.loadtype,
        driveremail:  req.body.driveremail,
        cycleperiod:  req.body.cycleperiod,
        bestroute: req.body.bestroute
        //mobile: req.body.mobile,
        //status: req.body.status
    })
    bin = await bin.save();

    if(!bin)
    return res.status(400).send('the Bin cannot be created!')
    res.send(bin);

})
```

```
router.delete('/:id', auth, (req, res)=>{
    Bin.findByIdAndRemove(req.params.id).then(bin =>{
        if(bin) {
            return res.status(200).json({success: true, message: 'the bin is
deleted!'})
        } else {
            return res.status(404).json({success: false , message: "bin not
found!"})
        }
    }).catch(err=>{
       return res.status(500).json({success: false, error: err})
    })
})



router.put('/:id',async  (req,  res)=>  {
    const bin = await Bin.findByIdAndUpdate(
        req.params.id,
        {
            adminemail: req.body.adminemail,
            area: req.body.area,
            locality: req.body.locality,
            landmark: req.body.landmark,
            city: req.body.city,
            loadtype: req.body.loadtype,
            driveremail:  req.body.driveremail,
            cycleperiod:  req.body.cycleperiod,
            bestroute:  req.body.bestroute
        },
        { new: true}
    )

    if(!bin)
    return res.status(400).send('the Bin cannot be created!')

    res.send(bin);
})




router.put('/map/:id',async (req, res)=> {
    const bin = await Bin.findByIdAndUpdate(
        req.params.id,
        {
            lat: req.body.lat,
            long: req.body.long
        },
        { new: true}
    )
    if(!bin)
    return res.status(400).send('the Bin cannot be created!')

    res.status(200).send(bin);

})



router.put('/status/:id', auth, async (req, res)=> {
    const bin = await Bin.findByIdAndUpdate(
        req.params.id,
        {
            status: req.body.status
        },
        { new: true}
    )

    if(!bin)
    return res.status(400).send('the bin cannot be created!')

    res.send(bin);
})


module.exports =router;
```

Compontent.js

```
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
import  "./css/bootstrap.min.css";
import "./css/owl.carousel.min.css";
import "./css/font-awesome.min.css";
import "./css/animate.css";
import "./css/font-awesome.min.css";
import "./css/lineicons.min.css";
import "./css/magnific-popup.css";
import "./css/style.css";
import "./js/jquery.min.js";
import "./js/bootstrap.bundle.min.js";
{/*
import "./js/waypoints.min.js";
import "./js/jquery.easing.min.js";
import "./js/owl.carousel.min.js";
import  "./js/jquery.magnific-popup.min.js";
*/}
import imgSmall from "./img/core-img/logo-small.png";
import imgBg from "./img/bg-img/9.png";
import Logout from './Logout.jsx';
import Title from './Title.jsx';

// useremail  complaint mobile lat long status

const PostComplaint = () => {
  const binarea =
(document.cookie.replace(/(?:(?:^|.*;\s*)binarea\s*=\s*([^;]*).*$)|^.*$/,
'$1'));
  const userEmail =
decodeURIComponent(document.cookie.replace(/(?:(?:^|.*;\s*)email\s*=\s*([^;]*
).*$)|^.*$/, '$1'));

  const [formData, setFormData] = useState({
    binarea: '',
    useremail: '',
    complaint: ''
  });



  const postComplaintData = async () => {
    const token = localStorage.getItem('token');
    try {
      const response = await fetch('http://localhost:4000/api/v1/complaint/',
{
        method: 'POST',
        headers: {
          'Content-Type': 'application/json',
          'x-auth-token': token,
        },
        body: JSON.stringify({
          ...formData,
          binarea: binarea,
          useremail: userEmail,
        }),
      });

      if (response.ok) {
        console.log('Bin data posted successfully!');
        // Handle success, e.g., redirect to another page
        alert('Requested Successful');
        window.location.href = "view_complaints_user";

      } else {
        console.error('Error posting bin data:', response.statusText);
      }
    } catch (error) {
      console.error('Error posting bin data:', error.message);
    }
  };
  const handleInputChange = (e) => {
    const { name, value } = e.target;
    setFormData({ ...formData, [name]: value });
  };


  // OnForm Submit
  const handleSubmit = (e) => {
    e.preventDefault();

    postComplaintData();
  };


  return (
    <div>
        <div>

        <div className="header-area" id="headerArea">
        <div className="container h-100 d-flex align-items-center justify-
content-between">

        <div className="header-area" id="headerArea">
        <div className="container h-100 d-flex align-items-center justify-
content-between">
            <div className="logo-wrapper" style={{color:'#020310'}}><img
src={imgSmall} alt=""/> <Title /> </div>

            <div className="suha-navbar-toggler" data-bs-toggle="offcanvas"
data-bs-target="#suhaOffcanvas" aria-
controls="suhaOffcanvas"><span></span><span></span><span></span></div>
        </div>
        </div>

{/* tabindex="-1" */}
        <div className="offcanvas offcanvas-start suha-offcanvas-
wrap"  id="suhaOffcanvas" aria-labelledby="suhaOffcanvasLabel">
      <button className="btn-close btn-close-white text-reset" type="button"
data-bs-dismiss="offcanvas" aria-label="Close"></button>

      <div  className="offcanvas-body">
        <div className="sidenav-profile">
          <div className="user-profile"><img src={imgBg} alt=""/></div>
          <div className="user-info">
            <h6 className="user-name mb-1">Garbage management</h6>

          </div>
        </div>

        <ul className="sidenav-nav ps-0">
          <li><Link to="/user_home"><i className="lni lni-
home"></i>Home</Link></li>
          <li><Logout /></li>
          </ul>
      </div>
    </div>
      </div>
    </div>
    <div className="page-content-wrapper">
      <div className="top-products-area py-3">
        <div className="container">
          <div className="section-heading d-flex align-items-center justify-
content-between">
            <h6>Post complaint</h6>
          </div>
        {/* Form Scrip Start*/}
        <div className="profile-wrapper-area py-3">
          <div className="card user-data-card">
            <div className="card-body">
              <form  onSubmit={handleSubmit}>

              <div className="mb-3">
                  <div className="title mb-2"><span>Complaint</span></div>
                  <input className="form-control"
                    name="complaint" id="complaint"
                    value={formData.complaint}
                    onChange={handleInputChange}     type="text"  />
                </div>




                <button className="btn btn-success w-
100" type="submit">Submit</button>
              </form>
            </div>
          </div>
        </div>
        {/* Form Scrip End*/}



        </div>
      </div>
    </div>

            <div className="footer-nav-area" id="footerNav">
              <div className="container h-100 px-0">
```

```
                <div className="suha-footer-nav h-100">
                  <ul className="h-100 d-flex align-items-center justify-
content-between ps-0">
                    <li className="active"> <Link to="/user_home" ><i
className="lni lni-home"></i>Home </Link> </li>
                    <li><Logout /></li>


                  </ul>
                </div>
              </div>
            </div>


</div>
</div>
  )
}

export default PostComplaint
import React, { useState } from 'react';
import { Link } from 'react-router-dom';
import  "./css/bootstrap.min.css";
import "./css/owl.carousel.min.css";
import "./css/font-awesome.min.css";
import "./css/animate.css";
import "./css/font-awesome.min.css";
import "./css/lineicons.min.css";
import "./css/magnific-popup.css";
import "./css/style.css";
import "./js/jquery.min.js";
import "./js/bootstrap.bundle.min.js";
{/*
import "./js/waypoints.min.js";
import "./js/jquery.easing.min.js";
import "./js/owl.carousel.min.js";
import  "./js/jquery.magnific-popup.min.js";
*/}
import imgSmall from "./img/core-img/logo-small.png";
import imgBg from "./img/bg-img/9.png";
import imgMech from "./img/mechanic.png";
import imgRequest from "./img/destination.png";
import imgProfile from "./img/user.png";
import newcomplaint from "./img/bg-img/garbage.png";
import mycomplaint from "./img/bg-img/mycomplaint.png";
import Logout from './Logout';
import Title from './Title';

const UserHome = () => {
  return (
    <div>
```

```
        <div>

        <div className="header-area" id="headerArea">
        <div className="container h-100 d-flex align-items-center justify-
content-between">

        <div className="header-area" id="headerArea">
        <div className="container h-100 d-flex align-items-center justify-
content-between">
            <div className="logo-wrapper" style={{color:'black'}}><img
src={imgSmall} alt=""/> <Title /></div>

            <div className="suha-navbar-toggler" data-bs-toggle="offcanvas"
data-bs-target="#suhaOffcanvas" aria-
controls="suhaOffcanvas"><span></span><span></span><span></span></div>
        </div>
        </div>

{/* tabindex="-1" */}
        <div className="offcanvas offcanvas-start suha-offcanvas-
wrap"  id="suhaOffcanvas" aria-labelledby="suhaOffcanvasLabel">
      <button className="btn-close btn-close-white text-reset" type="button"
data-bs-dismiss="offcanvas" aria-label="Close"></button>

      <div  className="offcanvas-body">
        <div className="sidenav-profile">
          <div className="user-profile"><img src={imgBg} alt=""/></div>
          <div className="user-info">
            <h6 className="user-name mb-1">Garbage management</h6>

          </div>
        </div>

        <ul className="sidenav-nav ps-0">
          <li><Link to="/user_home"><i className="lni lni-
home"></i>Home</Link></li>
          <li><Logout /></li>
          </ul>
      </div>
    </div>
      </div>
    </div>
    <div  className="page-content-wrapper">
      <div className="top-products-area py-3">
        <div className="container">
          <div className="section-heading d-flex align-items-center justify-
content-center">
            <h4><div className="text-success">User Home</div></h4>
          </div>

          <div className="row g-3">
```

```
            <div className="col-6 col-md-12">
              <div className="card horizontal-product-card">
                <div className="card-body d-flex align-items-center">
                  <div className="card-body"><img src={newcomplaint}
className="img-fluid" style={{width:64, height:64}} />{" "}
                  <Link className="text-dark" to="/view_bin_user">
                   New Complaint </Link>
                    </div>
                </div>
              </div>
            </div>
            </div>

          <div className="row g-3">
            <div className="col-6 col-md-12">
              <div className="card horizontal-product-card">
                <div className="card-body d-flex align-items-center">
                  <div className="card-body"><img src={mycomplaint}
className="img-fluid" style={{width:64, height:64}} />{" "}
                  <Link className="text-success" to="/view_complaints_user">
                   My Complaints  </Link>
                    </div>
                </div>
              </div>
            </div>
            </div>

          <div className="row g-3">
            <div className="col-6 col-md-12">
              <div className="card horizontal-product-card">
                <div className="card-body d-flex align-items-center">
                  <div className="card-body"><img src={imgProfile}
className="img-fluid" style={{width:64, height:64}} />{" "}
                  <Link className="text-success" to="/user_profile">
                   My Profile  </Link>
                    </div>
                </div>
              </div>
            </div>
            </div>

            </div>
            </div>


            </div>

            <div className="footer-nav-area" id="footerNav">
              <div className="container h-100 px-0">
                <div className="suha-footer-nav h-100">
                  <ul className="h-100 d-flex align-items-center justify-
```

```
content-between ps-0">
                    <li className="active"> <Link to="/user_home" ><i
className="lni lni-home"></i>Home </Link> </li>
                    <li><Logout /></li>


                  </ul>
                </div>
              </div>
            </div>


</div>
</div>
  )
}

export default UserHome
```

6.9 Implementation details Summary

The implementation phase focused on translating the system design into a fully functional and interactive web application using modern technologies. The frontend was developed using React.js, offering a responsive and dynamic user interface that enables users to upload waste images, register complaints, and track the status of their reports. The backend was built with Node.js and Express.js, ensuring efficient request handling, routing, and server-side operations. MongoDB was employed as the database to securely store user information, complaint records, and image references. Communication between the frontend and backend was facilitated through RESTful APIs, ensuring seamless data exchange and system integration. Machine learning models were incorporated to enhance the accuracy of waste image classification, while the admin dashboard provided municipal authorities with tools to monitor reports, track complaint locations, and manage cleaning operations. Overall, the implementation achieved scalability, security, and user engagement, delivering a practical and intelligent solution for sustainable waste management.

CHAPTER 7

SYSTEM TESTING

The primary purpose of system testing is to identify and correct errors in the software. Testing involves examining the software thoroughly to discover any faults or weaknesses in components, sub-assemblies, assemblies, or the final product.
System testing ensures that the software:

- Meets its specified requirements
- Satisfies user expectations
- Operates reliably without failing in an unacceptable manner

It is a critical process in the software development life cycle that validates the **functionality**, **performance**, and **reliability** of the system. Various types of tests are conducted, each addressing specific aspects or requirements of the software to ensure comprehensive quality assurance.

Types of Tests

7.1 Unit Testing

Unit testing involves designing test cases to validate that the internal program logic functions correctly and that program inputs produce valid outputs. It tests individual software units after the completion of each unit and before integration.
Unit testing is a **structural (white-box)** testing approach that relies on knowledge of the code structure. It ensures that each component performs as specified, with clearly defined inputs and expected results. Unit tests also verify that all decision branches and internal code flows work accurately.

Test Strategy and Approach:

- Field testing performed manually with detailed functional tests.

Test Objectives:

- Ensure all field entries work properly.
- Verify that all pages are activated from identified links.
- Ensure entry screens, messages, and responses are not delayed.

Features to be Tested:

- Verify that entries are in the correct format.
- Prevent duplicate entries.
- Ensure all links navigate to the correct pages.

7.2 Integration Testing

Integration testing evaluates whether combined software components function correctly as a unified system. Even if individual units pass unit testing, integration testing identifies errors that occur due to interactions between components.

It is **event-driven**, focusing on the basic outcomes of screens, fields, and interfaces.

Objective:

- Verify correct interaction between components and detect interface defects.

Test Results:

- All test cases passed successfully with no defects encountered.

Functional Testing

Functional testing systematically demonstrates that the software operates as specified by business and technical requirements, system documentation, and user manuals.

Key Focus Areas:

- **Valid Inputs:** Must be accepted.
- **Invalid Inputs:** Must be rejected.
- **Functions:** Must be exercised thoroughly.
- **Outputs:** Must be generated correctly.
- **System Procedures:** Interfaces with other systems or processes must function correctly.

Functional testing covers business process flows, data fields, predefined processes, and successive operations to ensure completeness.

System Testing

System testing ensures that the entire integrated software system meets the specified requirements. It verifies the configuration to produce predictable results and emphasizes process flows, integration points, and predefined links.

Example: Configuration-oriented system integration tests.

White Box Testing

White box testing examines the internal workings, structure, and code of the software. It is used to test areas not reachable from the black-box level and ensures that internal logic works as intended.

Black Box Testing

Black box testing evaluates software without any knowledge of its internal structure or code. Tests are designed based on specifications or requirements documents, providing inputs and verifying outputs without considering internal implementation.

7.3 Acceptance Testing

User Acceptance Testing (UAT) is a critical phase that requires end-user participation. It ensures that the system meets functional requirements and is ready for deployment.

Test Results:

- All test cases passed successfully with no defects encountered.

7.4 Test Case

A **test case** is a set of instructions designed to verify that a particular feature or functionality of an application works as expected. It defines the **input**, **action**, and **expected response** to determine whether the system behaves correctly.

Purpose:

To provide a systematic way to validate a specific test objective and ensure that the expected behavior of the system is satisfied.

Components of a Test Case:

- **Input:** The data or conditions provided to the system.
- **Action:** The steps to perform on the system using the given input.
- **Expected Response:** The anticipated outcome that confirms whether the system behaves correctly.

Characteristics of a Good Test Case:

- **Accurate:** Clearly addresses the intended purpose.
- **Economical:** Contains no unnecessary steps; concise and efficient.
- **Traceable:** Can be linked back to specific requirements or user stories.
- **Repeatable:** Can be executed multiple times with consistent results.
- **Reusable:** Can be reused for future testing scenarios if needed.

Well-written test cases ensure **consistency, reliability, and traceability**, helping detect defects early and maintain high-quality software.

S. NO	SCENARIO	INPUT	EXPECTED OUTPUT	ACTUAL OUTPUT
1	Admin login Details	Admin enter login details	Login successfully or if incorrect login details "Login unsuccessfully"	Login successfully or Login unsuccessfully
2	Create garbage bin	Admin created by all garbage bin details like bin id, bin area, etc.,	All the products Create successfully	Created successfully or created unsuccessfully

3	Update And delete garbage bin	Admin can update and delete the garbage bin	If any changes or non-available garbage bin admin can edit or delete products	Updated successfully or unsuccessfully Delete successfully or unsuccessfully
4	Assign Best route for drivers	Admin will assign a best short route for drivers	Admin will assign route for drivers	Assigned successfully Or assigned unsuccessfully
5	Manage driver	Admin will manage the driver details when they are changed.	Admin will manage the details	View and maintain the driver details
6	View garbage report	Admin check and view all garbage report details	Admin view garbage report details	View all garbage report details
7	View complaint from public	Admin can view public complaint details	Admin view and check the complaint details	View all complaint details
9	Public Login	Username Password and Key Verification	If correct directed to home page otherwise show “Invalid Login”	Login successfully or Login unsuccessfully
10	Public register	Username Password other details and Key Verification	All the public details register successfully	Register successfully Or Register unsuccessfully
11	Register complaint	Public created by complaint based on garbage	All the complaint Create successfully	Created successfully or created unsuccessfully
12	My complaint and status	Public will check complaint and their status	Public view complaint and their status details	View all complaint status details
13	Driver Login	Username Password and Key Verification	If correct directed to home page otherwise show “Invalid Login”	Login successfully or Login unsuccessfully

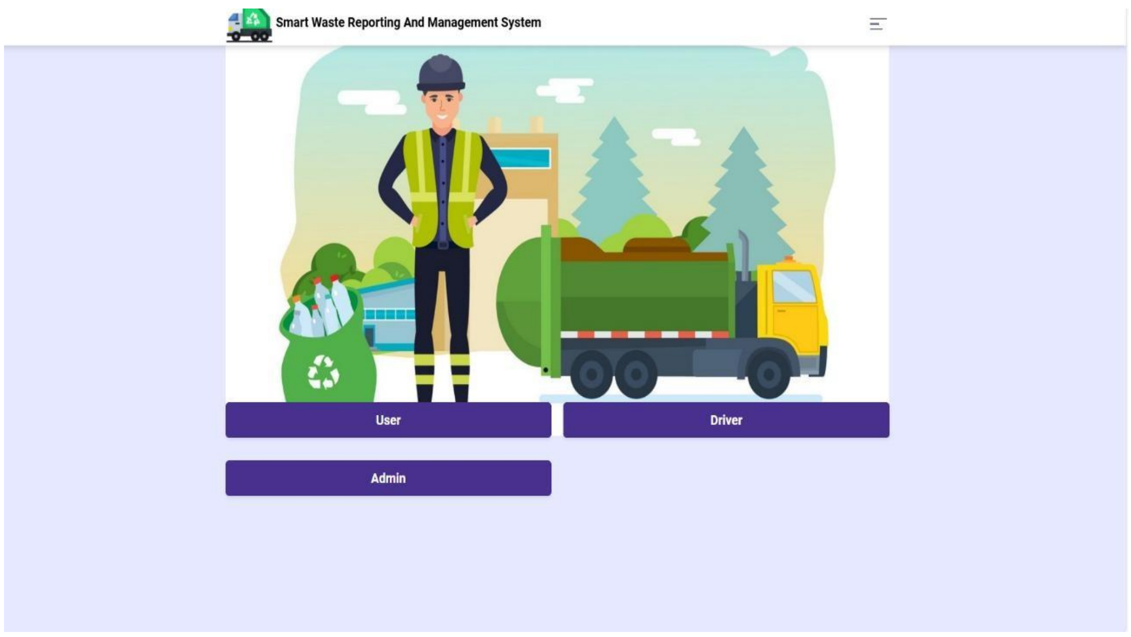
14	Check daily works	Driver once login check the all-daily works	View all daily works from the admin	View all daily works and updates
15	Choose best route	Driver will choose short and best route for reached area	View all routes from the driver choose best route	View and select best route
16	Update garbage load	Driver will upload garbage load weight and their status	Updated Garbage load and status	Status updated successfully or unsuccessfully

7.5 System Testing Summary

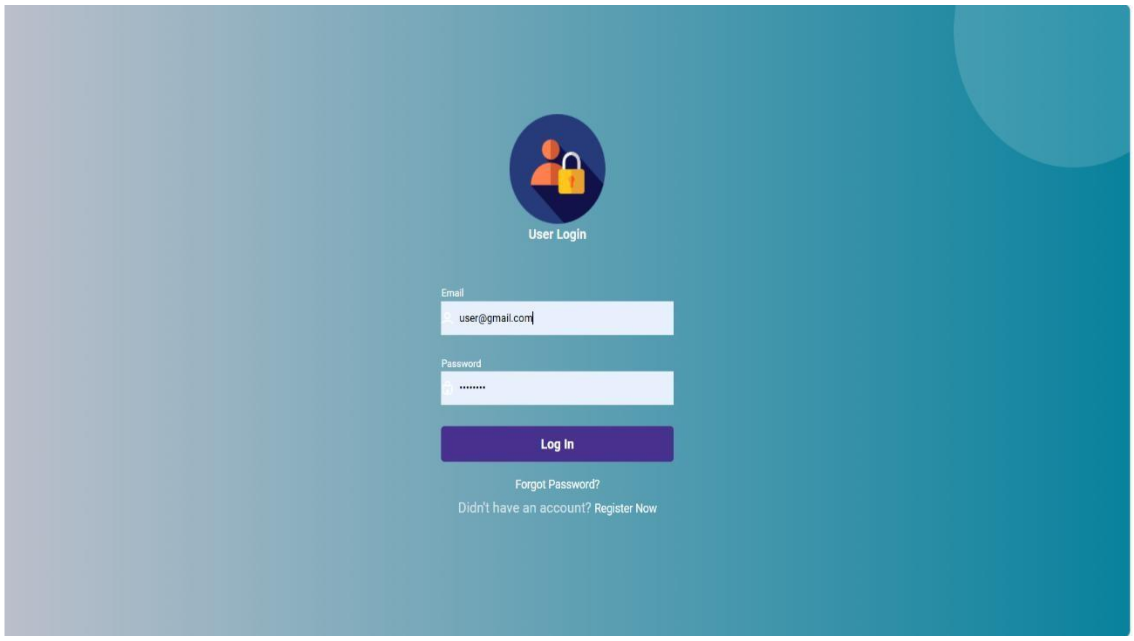
System testing was carried out to verify that the Smart Waste Reporting and Management System performs as intended and complies with all defined requirements. A comprehensive range of testing methods—unit testing, integration testing, functional testing, and user acceptance testing (UAT)—was applied to ensure overall system reliability and performance. Unit testing verified the functionality of individual modules such as user registration, image upload, and complaint tracking. Integration testing validated the seamless interaction between the frontend, backend, and database components. Functional testing confirmed that all features operated according to the specified requirements, while UAT ensured usability and satisfaction from the end-user perspective. All test cases were executed successfully without major issues, confirming that the system is stable, efficient, and ready for deployment in a real-world smart city environment.

CHAPTER 8

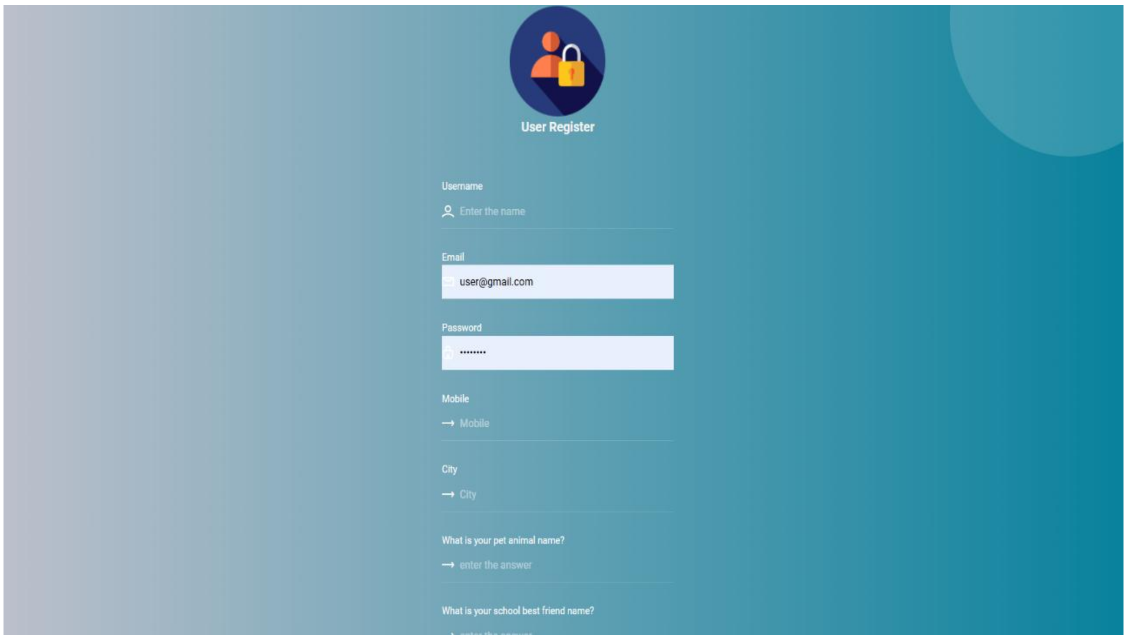
SAMPLE OUTPUTS



Snapshot 8.1 Dashboard



Snapshot 8.2 User Login



User Register

Username

Enter the name

Email

user@gmail.com

Password

••••••••

Mobile

Mobile

City

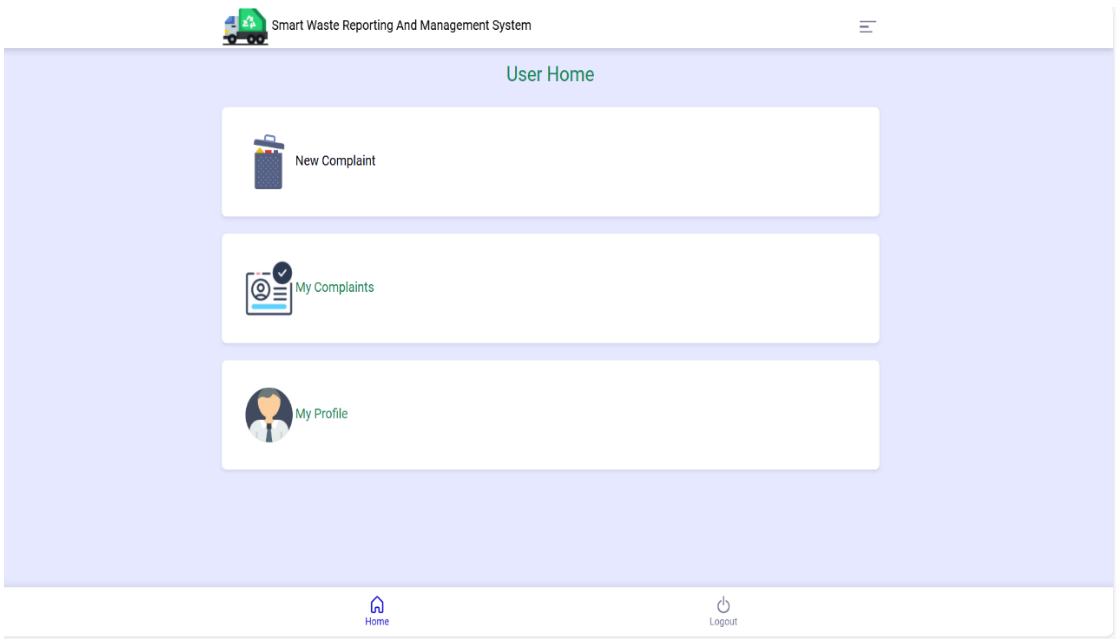
City

What is your pet animal name?

enter the answer

What is your school best friend name?

Snapshot 8.3 User Registration



Smart Waste Reporting And Management System

User Home

New Complaint

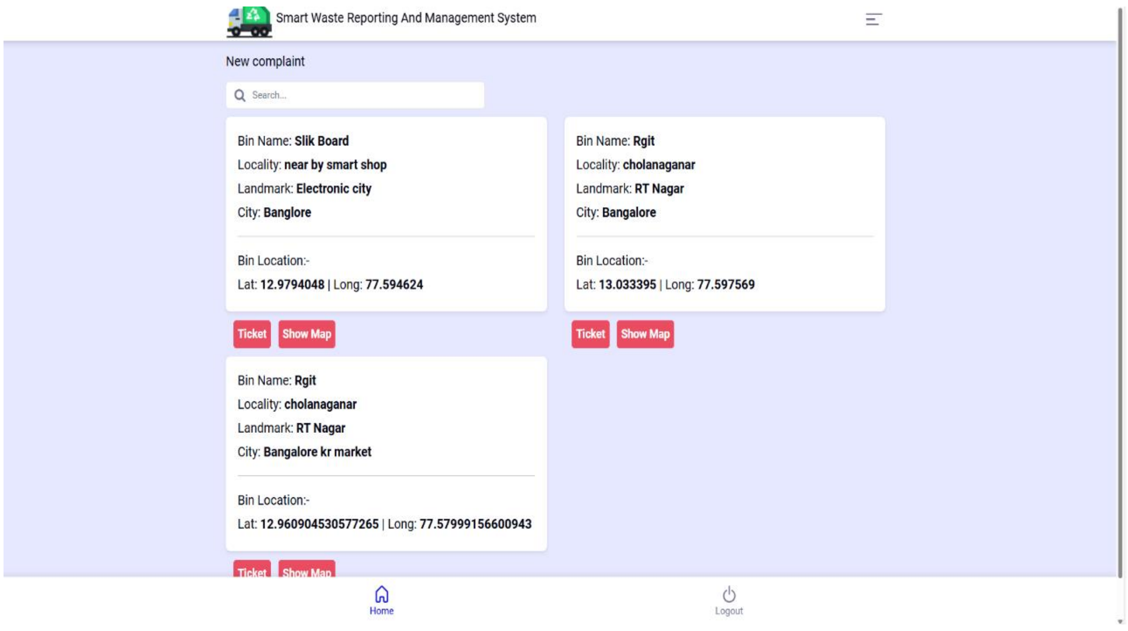
My Complaints

My Profile

Home

Logout

Snapshot 8.4 User Home


Smart Waste Reporting And Management System

New complaint

Search...

Bin Name: **Slik Board**
Locality: **near by smart shop**
Landmark: **Electronic city**
City: **Banglore**

Bin Location:-
Lat: **12.9794048** | Long: **77.594624**

Ticket Show Map

Bin Name: **Rgit**
Locality: **cholanaganar**
Landmark: **RT Nagar**
City: **Bangalore**

Bin Location:-
Lat: **13.033395** | Long: **77.597569**

Ticket Show Map

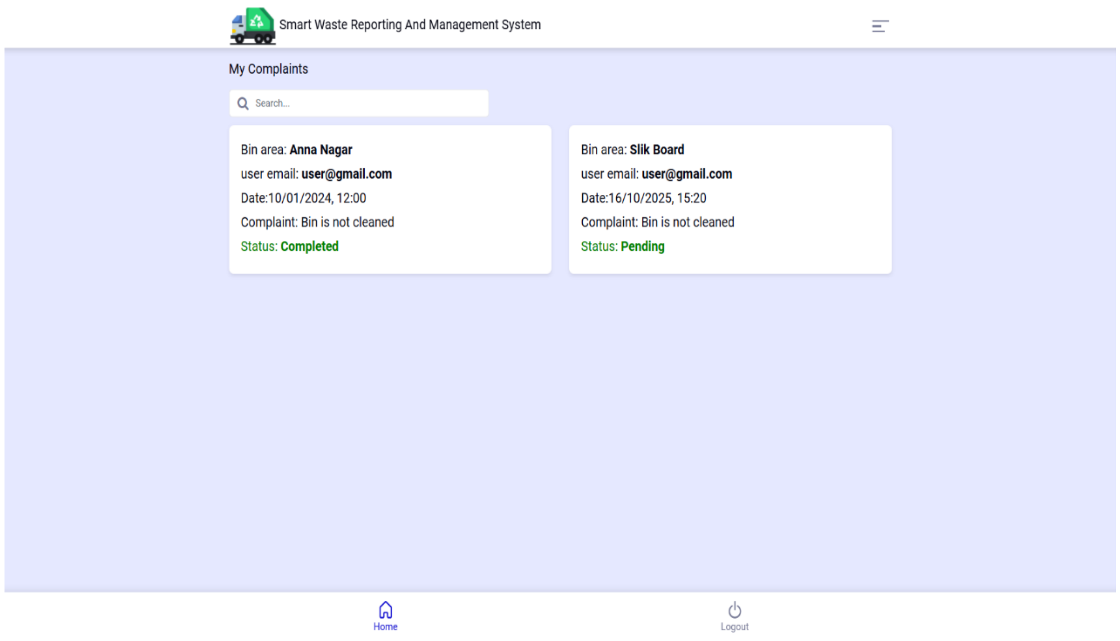
Bin Name: **Rgit**
Locality: **cholanaganar**
Landmark: **RT Nagar**
City: **Bangalore kr market**

Bin Location:-
Lat: **12.960904530577265** | Long: **77.57999156600943**

Ticket Show Map

Home Logout

Snapshot 8.5 User's New Complaints


Smart Waste Reporting And Management System

My Complaints

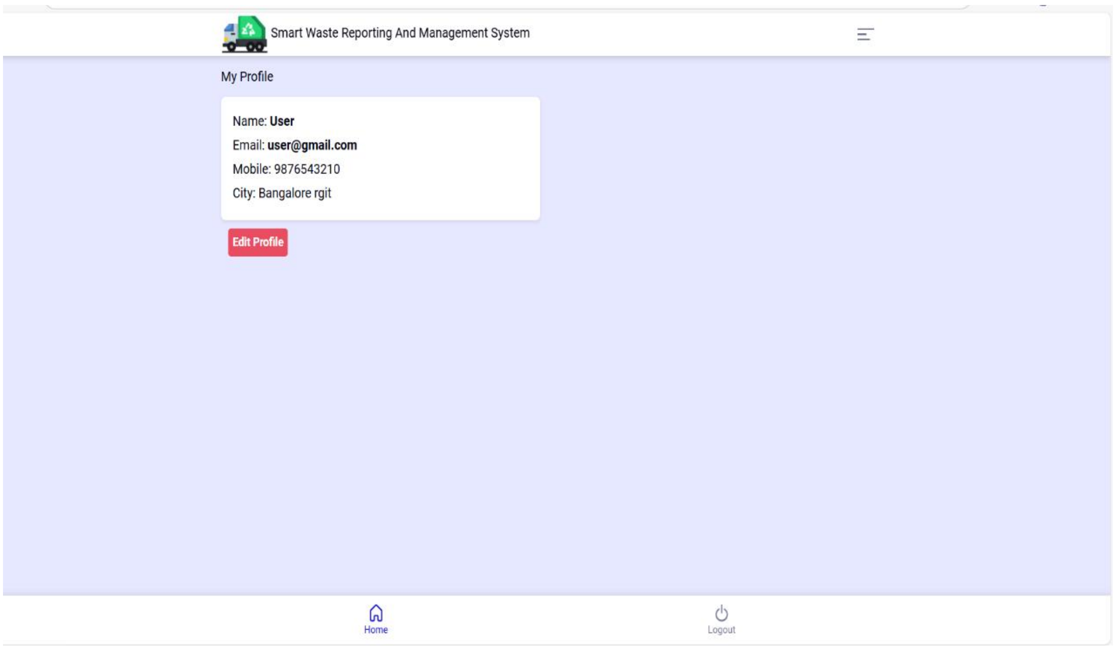
Search...

Bin area: **Anna Nagar**
user email: **user@gmail.com**
Date:10/01/2024, 12:00
Complaint: Bin is not cleaned
Status: **Completed**

Bin area: **Slik Board**
user email: **user@gmail.com**
Date:16/10/2025, 15:20
Complaint: Bin is not cleaned
Status: **Pending**

Home Logout

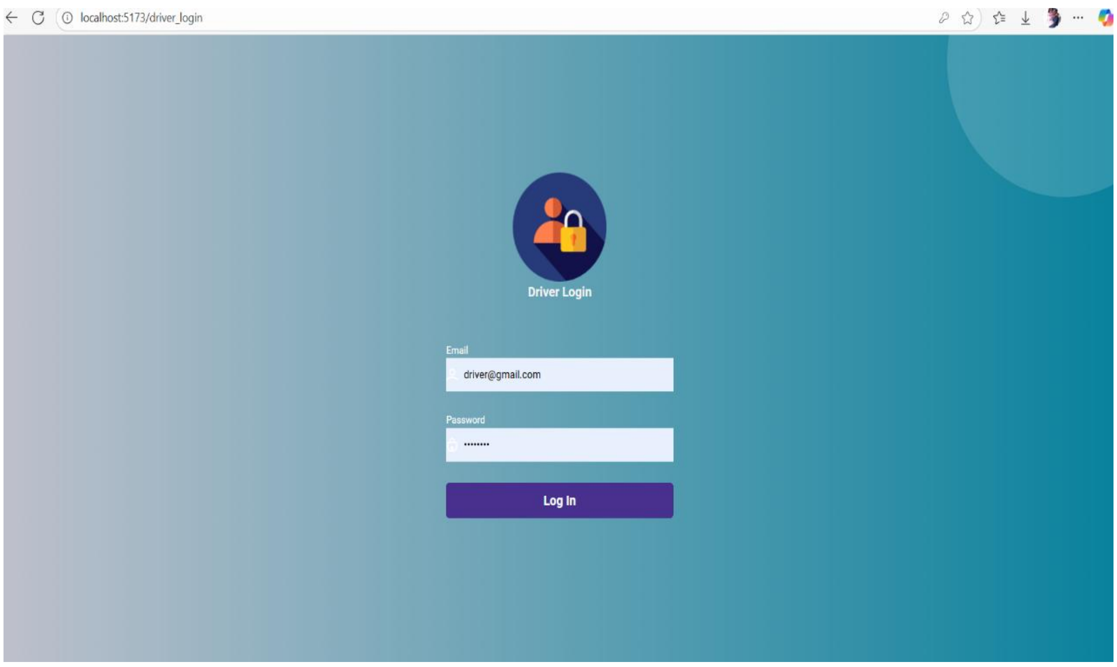
Snapshot 8.6 User's Complaints



My Profile

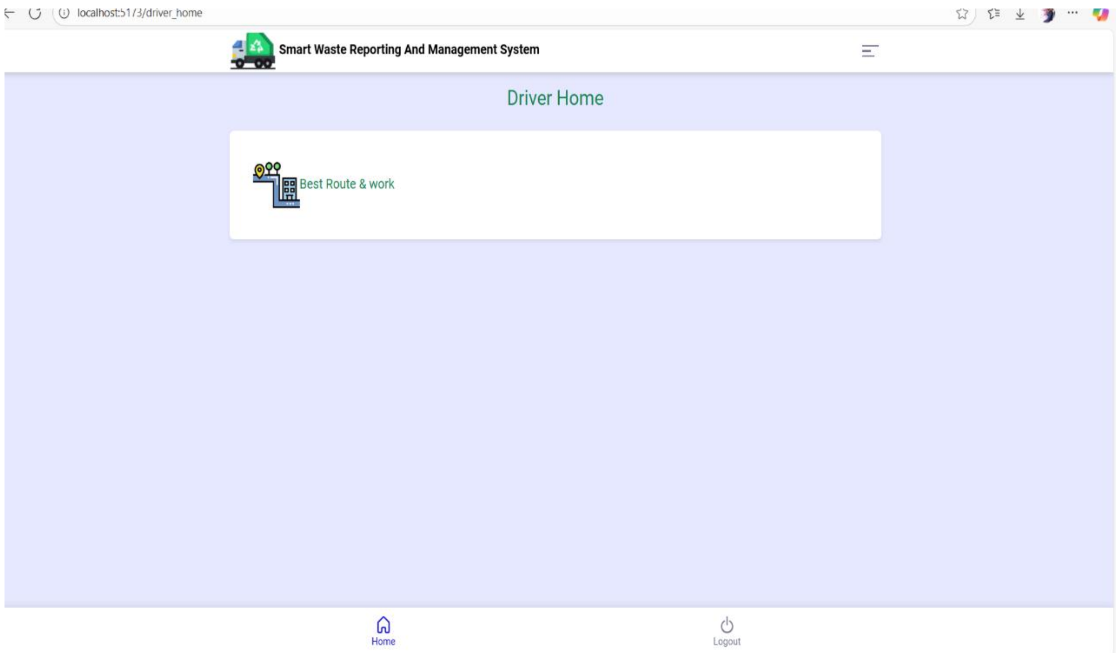
Name: **User**
Email: **user@gmail.com**
Mobile: 9876543210
City: Bangalore rgit

Snapshot 8.7 User's Profile

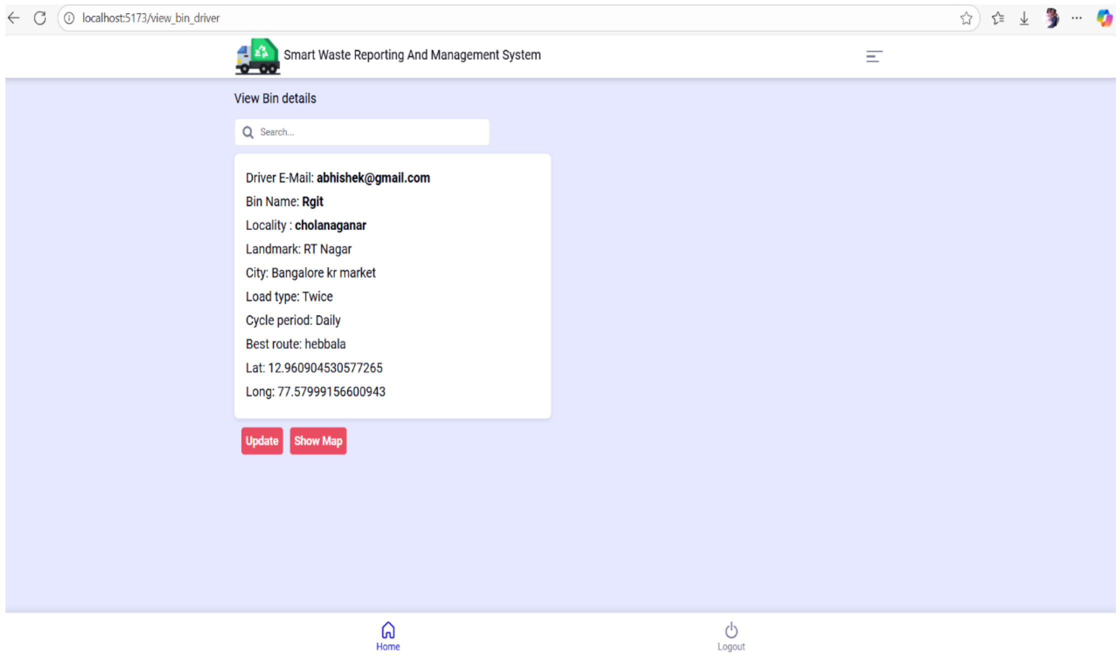


Email
driver@gmail.com
Password
••••••••

Snapshot 8.8 Driver Login

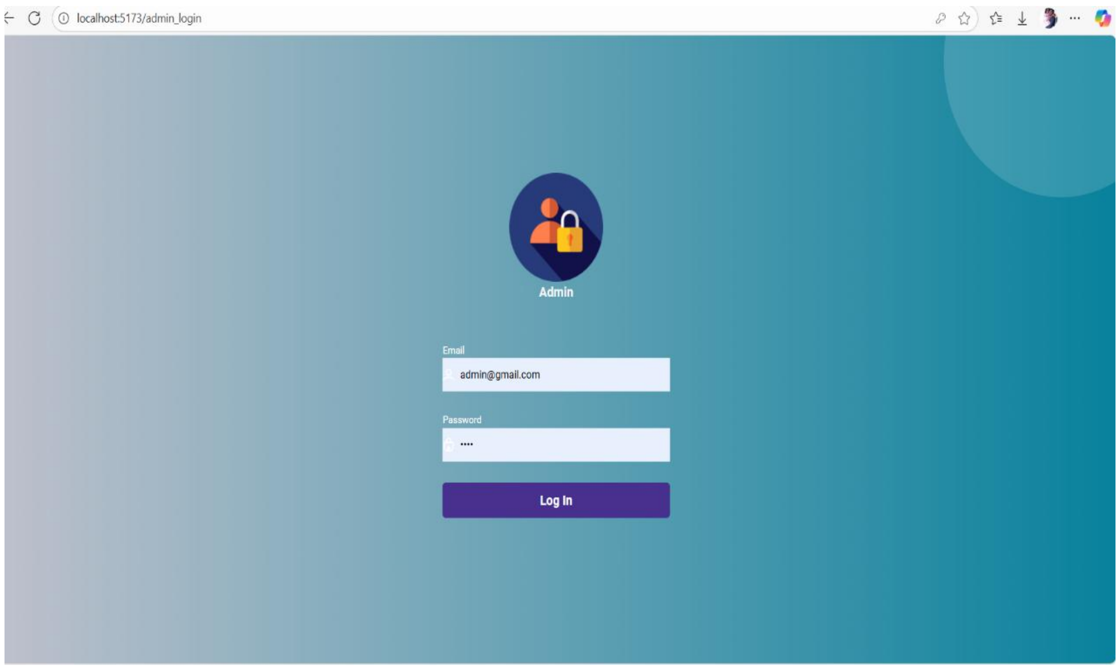

localhost:5173/driver_home
Smart Waste Reporting And Management System
Driver Home
Best Route & work
Home
Logout

Snapshot 8.9 Driver Home

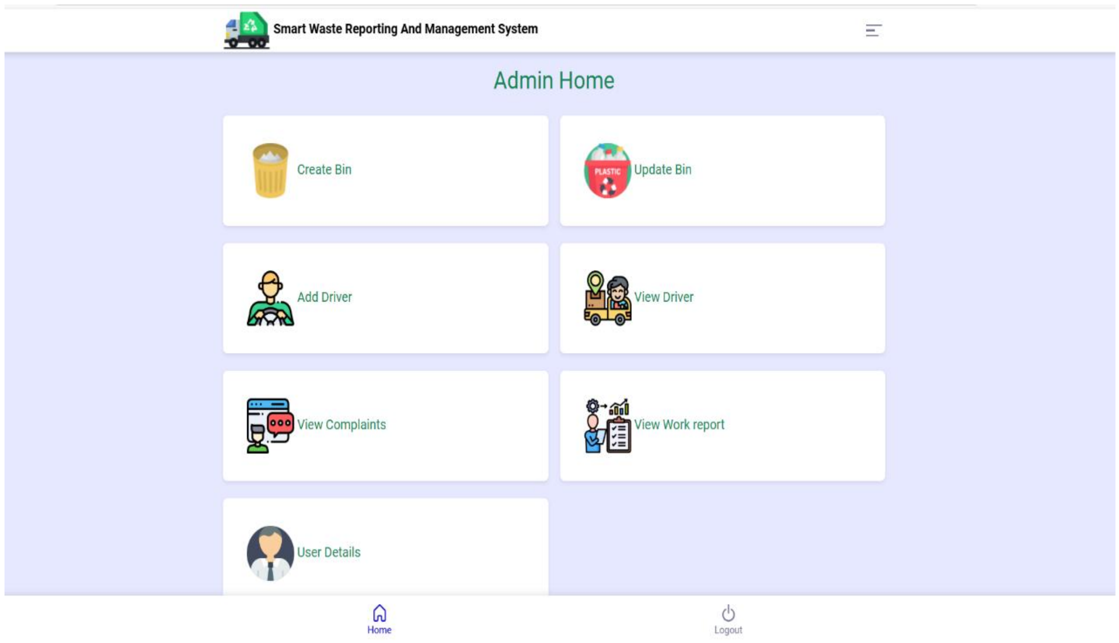

localhost:5173/view_bin_driver
Smart Waste Reporting And Management System
View Bin details
Search...
Update
Show Map
Home
Logout

Driver E-Mail: **abhishek@gmail.com**
Bin Name: **Rgit**
Locality : **cholanaganar**
Landmark: RT Nagar
City: Bangalore kr market
Load type: Twice
Cycle period: Daily
Best route: hebbala
Lat: 12.960904530577265
Long: 77.57999156600943

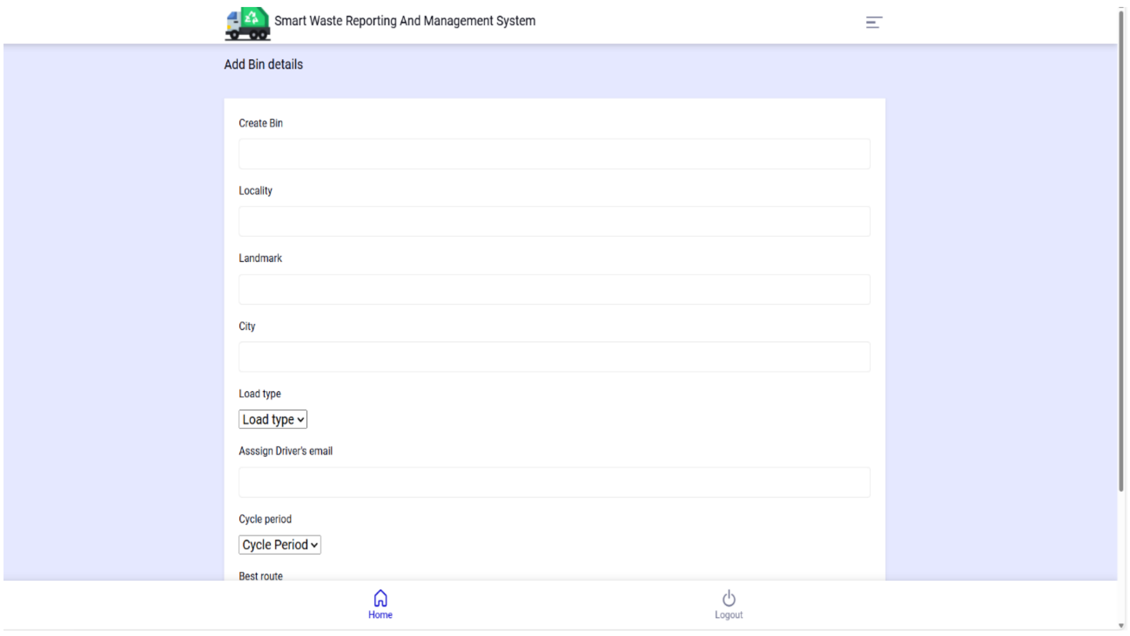
Snapshot 8.10 Driver's Best Route & Work



Snapshot 8.11 Admin Login



Snapshot 8.12 Admin Home



Add Bin details

Create Bin

Locality

Landmark

City

Load type

Load type

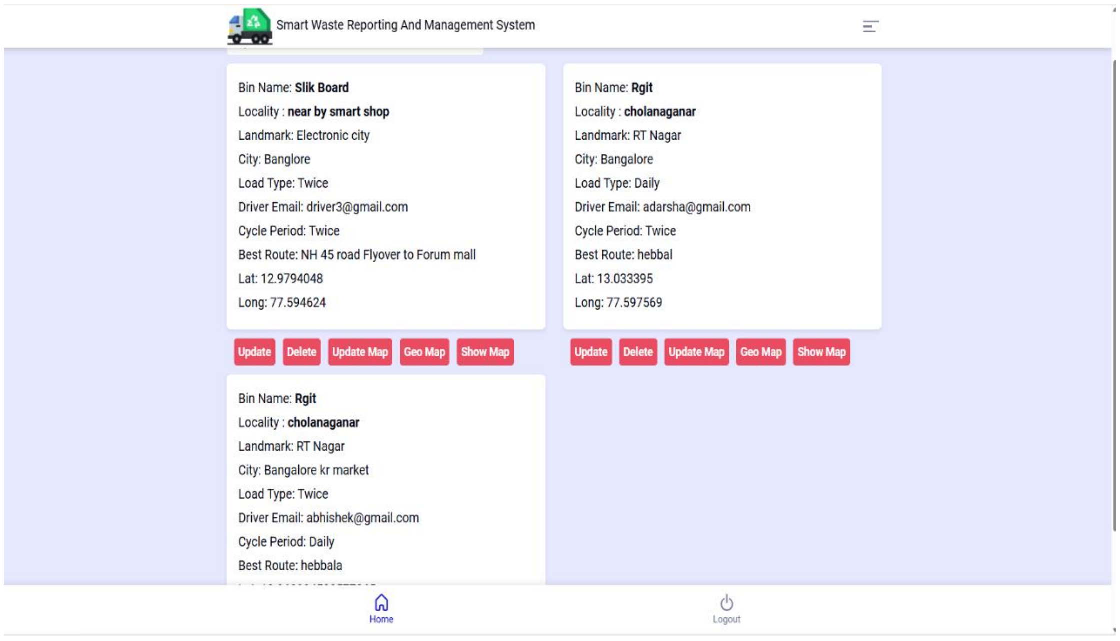
Asssign Driver's email

Cycle period

Cycle Period

Best route

Snapshot 8.13 Create Bin

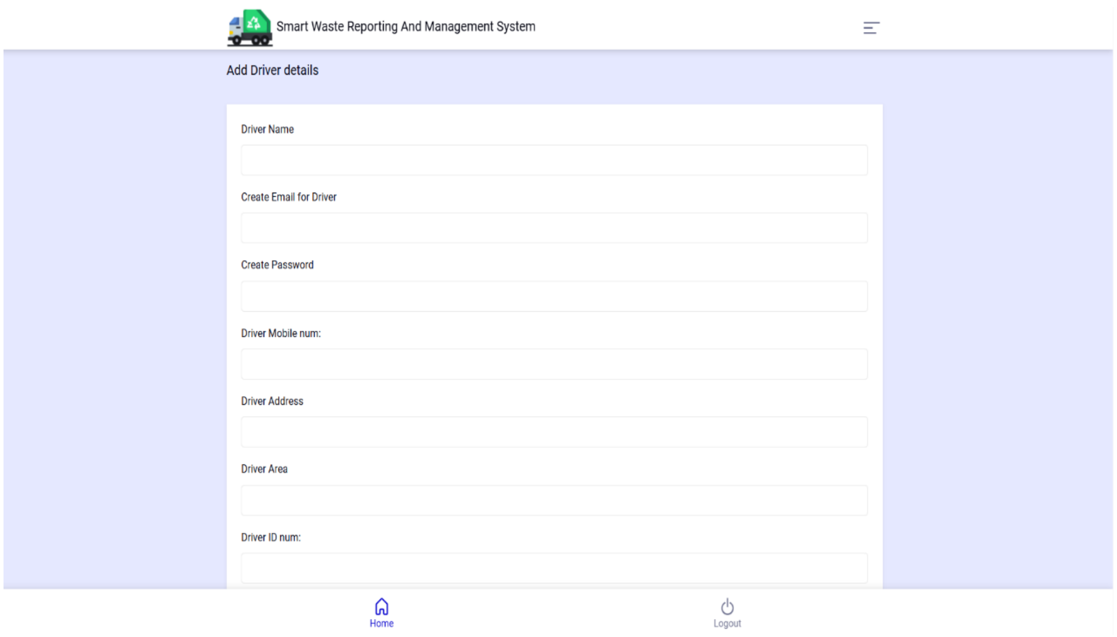


Bin Name: **Slik Board**
Locality : **near by smart shop**
Landmark: Electronic city
City: Banglore
Load Type: Twice
Driver Email: driver3@gmail.com
Cycle Period: Twice
Best Route: NH 45 road Flyover to Forum mall
Lat: 12.9794048
Long: 77.594624

Bin Name: **Rgit**
Locality : **cholanaganar**
Landmark: RT Nagar
City: Bangalore
Load Type: Daily
Driver Email: adarsha@gmail.com
Cycle Period: Twice
Best Route: hebbal
Lat: 13.033395
Long: 77.597569

Bin Name: **Rgit**
Locality : **cholanaganar**
Landmark: RT Nagar
City: Bangalore kr market
Load Type: Twice
Driver Email: abhishek@gmail.com
Cycle Period: Daily
Best Route: hebbala

Snapshot 8.14 Update Bin



Add Driver details

Driver Name

Create Email for Driver

Create Password

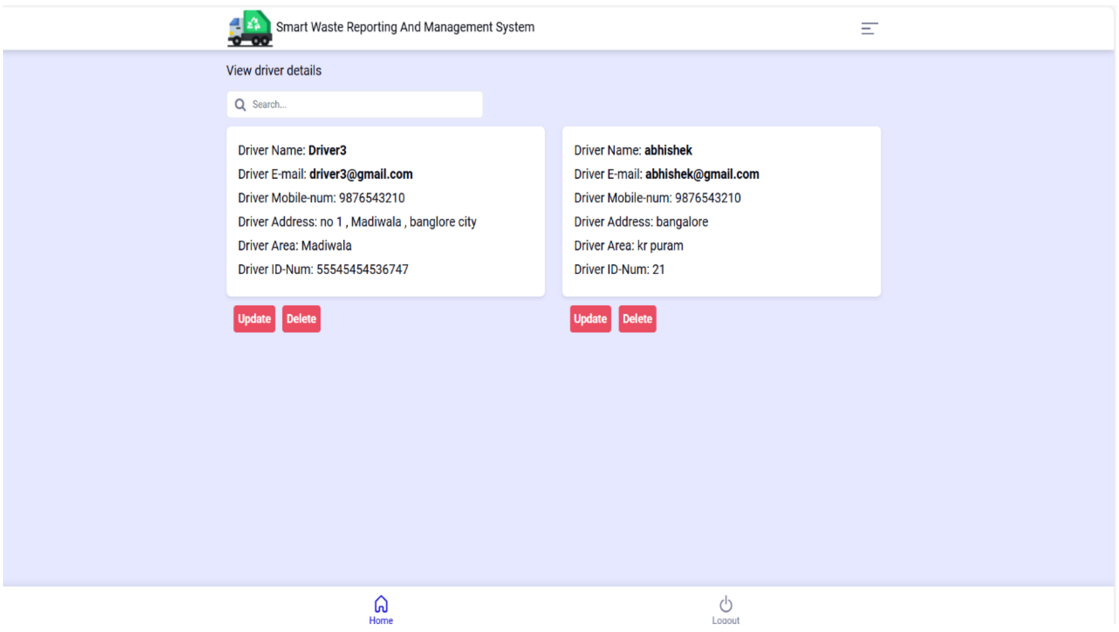
Driver Mobile num:

Driver Address

Driver Area

Driver ID num:

Snapshot 8.15 Add Driver

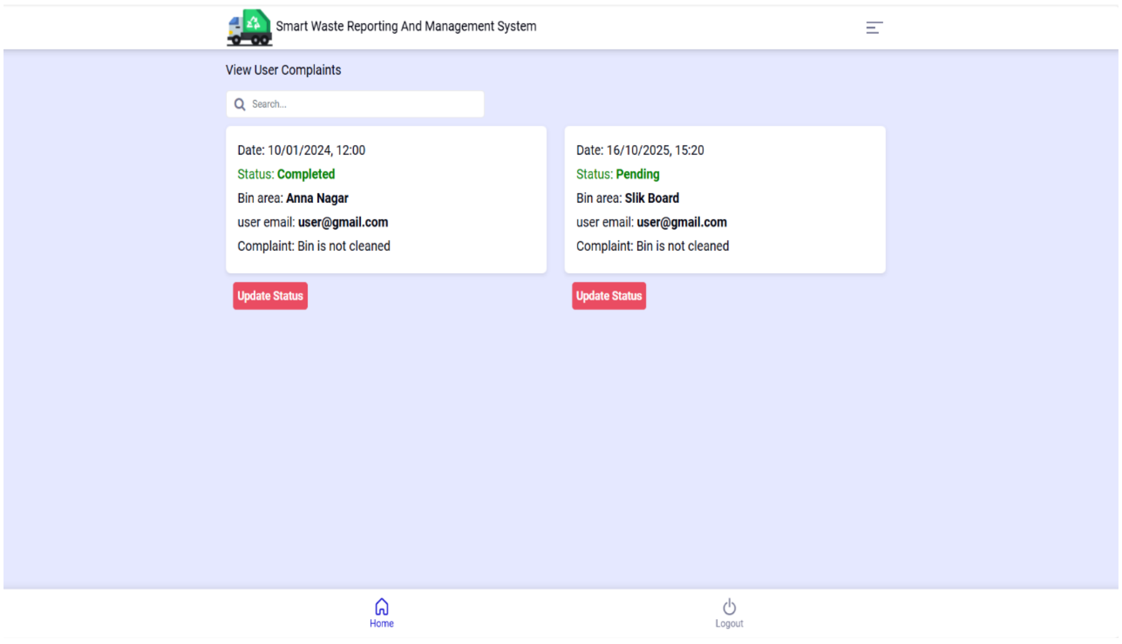


View driver details

Driver Name: **Driver3**
Driver E-mail: **driver3@gmail.com**
Driver Mobile-num: 9876543210
Driver Address: no 1 , Madiwala , banglore city
Driver Area: Madiwala
Driver ID-Num: 55545454536747

Driver Name: **abhishek**
Driver E-mail: **abhishek@gmail.com**
Driver Mobile-num: 9876543210
Driver Address: bangalore
Driver Area: kr puram
Driver ID-Num: 21

Snapshot 8.16 View Driver

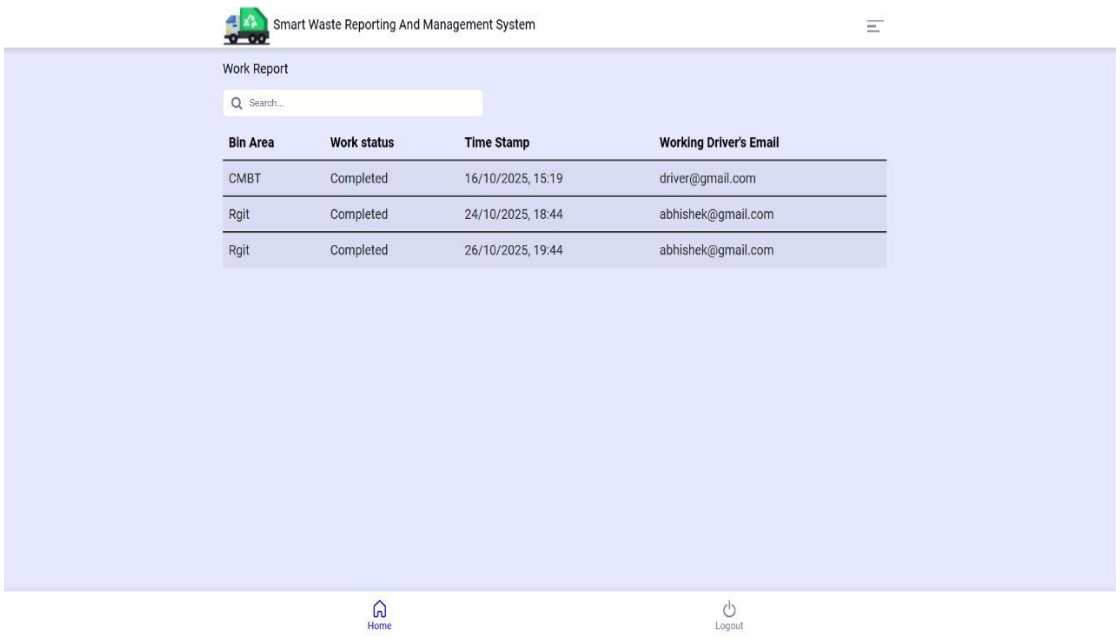


View User Complaints

Date: 10/01/2024, 12:00
Status: **Completed**
Bin area: **Anna Nagar**
user email: **user@gmail.com**
Complaint: Bin is not cleaned

Date: 16/10/2025, 15:20
Status: **Pending**
Bin area: **Slik Board**
user email: **user@gmail.com**
Complaint: Bin is not cleaned

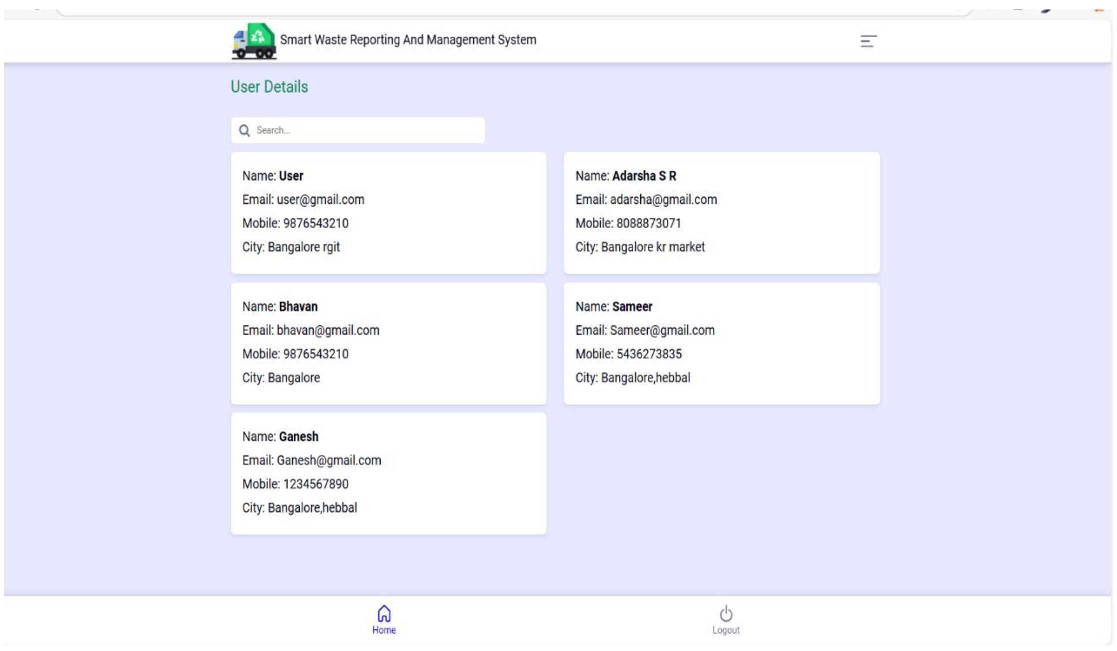
Snapshot 8.17 View Complaints



Work Report

Bin Area	Work status	Time Stamp	Working Driver's Email
CMBT	Completed	16/10/2025, 15:19	driver@gmail.com
Rgit	Completed	24/10/2025, 18:44	abhishek@gmail.com
Rgit	Completed	26/10/2025, 19:44	abhishek@gmail.com

Snapshot 8.18 View Work Report



User Details

Name: **User**
Email: user@gmail.com
Mobile: 9876543210
City: Bangalore rgit

Name: **Adarsha S R**
Email: adarsha@gmail.com
Mobile: 8088873071
City: Bangalore kr market

Name: **Bhavan**
Email: bhavan@gmail.com
Mobile: 9876543210
City: Bangalore

Name: **Sameer**
Email: Sameer@gmail.com
Mobile: 5436273835
City: Bangalore,hebbal

Name: **Ganesh**
Email: Ganesh@gmail.com
Mobile: 1234567890
City: Bangalore,hebbal

Snapshot 8.19 User Details

CONCLUSIONS

The Smart Waste Reporting and Management System for Smart Cities offers an innovative and practical solution to address inefficiencies in existing waste collection processes. By leveraging a centralized admin dashboard, driver tracking, and automated route optimization, the system enhances operational efficiency, reduces costs, and minimizes environmental impact. It promotes environmental sustainability by decreasing unnecessary vehicle movements, thereby reducing air and noise pollution and ensuring timely waste collection. This leads to cleaner streets, improved public health, and a higher quality of urban life. Additionally, the integration of data analytics empowers authorities to make data- driven decisions and allocate resources effectively. Citizens benefit from an easy-to-use complaint registration and tracking system, ensuring greater transparency and civic engagement. Overall, the system provides a cost-effective, sustainable, and intelligent framework for modern urban waste management, paving the way for cleaner and healthier smart cities.

Future Work

In the future, the system can be enhanced by integrating Internet of Things (IoT) sensors to automatically detect bin fill levels and transmit real-time data to the central server, eliminating the need for manual updates. Machine learning algorithms could be employed to predict waste generation patterns and dynamically optimize collection schedules. Furthermore, the integration of Geographic Information Systems (GIS) and AI-driven analytics can enhance route planning, reduce fuel consumption, and improve operational efficiency. A reward-based mechanism can be introduced to encourage citizens to report waste-related issues promptly, fostering active community participation. Expanding the system to include recycling and waste segregation modules would further strengthen its role in promoting sustainable and intelligent urban waste management.

REFERENCES

- "Colombo Vehicle Statistics (2015)." Indi.ca. [Online]. Available: http://indi.ca/2015/10/colombo-vehicle-statistics-2015/. [Accessed: 09-Jan-2017]

- "Population and Housing." Population and Housing. [Online]. Available: http://www.statistics.gov.lk/page.asp?page=Population%20and%20Housing/ [Accessed: 09-Jan-2017].

- Council, Colombo Municipal. "Colombo Municipal Council." Garbage Collection [Online] Available: http://colombo.mc.gov.lk/garbagecollection.php/ [Accessed: 04-Jan-2017]

- S. Lokuliyana, J. A. D. C. A. Jayakody, L. Rupasinghe, and S. Kandawala, "IGOE IoT framework for waste collection optimization," 2017 6th National Conference on Technology and Management (NCTM), Malabe, 2017, pp. 12–16.

- R. Fujdiak, P. Masek, P. Mlynek, J. Misurec and E. Olshannikova, "Using genetic algorithms for advanced municipal waste collection in Smart City," 2016 10th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Prague, 2016, pp. 1–6.

- T. Anagnostopoulos, A. Zaslavsky, A. Medvedev, and S. Khoruzhnikov, "Top-k Query based Dynamic Scheduling for IoT-enabled Smart City Waste Collection," Proc. of the 16th IEEE International Conference on Mobile Data Management (MDM 2015), Pittsburgh, US.

- "National Solid Waste Management Support Center." Ministry of Provincial Councils & Local Government. [Online]. Available: http://www.lgpc.gov.lk/eng/?page_id=1118/. [Accessed: 03-Jan-2017].

- "See Ultrasonic Sensor." PDF. [Accessed: 23-Jan-2017].

- "Raspberry Pi Zero." Raspberry Pi. [Online]. Available: [Accessed: 25-Jan-2017]. https://www.raspberrypi.org/products/pi-zero/.

- "Waypoints in directions | Google Maps JavaScript API | Google Developers." [Online]Available:https://developers.google.com/maps/documentation/javascript/examples/directionswaypoints/ [Accessed: 15-Mar-2017]

- "2.2.3 Determining current domestic waste generation per capita." [Online] Available http://iwmp.environment.gov.za/guideline/2/2_2_3/.[Accessed: 23-Apr-2017]. Integrated Waste Management Plan (IWMP).