

# **ECOVAULT**

## **MINI PROJECT REPORT**

Submitted to

**RENUKA , SIDHARTH , KRISHNA**

**(SNG23MCA-2071, SNG23MCA-2077, SNG23MCA-2050)**

to

*the APJ Abdul Kalam Technological University*

*in partial fulfillment of the requirements for the award of the Degree*

*of*

*Master Of Computer Applications*



**Department of Computer Applications**

**Sree Narayana Gurukulam College of Engineering,**

**Kadayiruppu, 623811**

**2023-2025**

## DECLARATION

We undersigned hereby declare that the project report “WASTE MANAGEMENT SYSTEM (ECOVAULT)”, submitted for partial fulfilment of the requirements for the award of degree of Master Of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Asst.Prof. Meera K Chandran**. This submission represents our ideas in our own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kadayiruppu

Renuka, Sidharth, Krishna

Date:

**SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING,  
KADAYIRUPPU, KOLENCHERY-623811  
DEPARTMENT OF COMPUTER APPLICATIONS**



**CERTIFICATE**

This is to certify that the project report entitled “WASTE MANAGEMENT SYSTEM (ECOVAULT)” submitted by **RENUKA KAMATH , SIDHARTH SURESH , KRISHNA ANILKUMAR** , to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications is a bonafide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

**Asst.Prof. Meera K Chandran**

Project Guide

Department of

Computer Applications

**Prof.Dr. Sandhya R**

Head Of the Department

Department of Computer Applications

CONTENTS		
No	TITLE	Page
	ACKNOWLEDGEMENT	6
	ABSTRACT	7
Chapter 1. INTRODUCTION		9
1.1	Project Background	9
Chapter 2. SYSTEM STUDY AND ANALYSIS		10
2.1	Analysis Of Existing System	10
2.2	Drawbacks Of Existing System	10
2.3	Problem Statement	12
2.4	problem System	13
2.5	Feasibility Studys	14
2.6	Project Planning	16
2.7	Development Environment	17
2.8	BackEnd	19
2.9	Hardware Requirement	20
2.10	Software and Hardware Specification	21
Chapter 3. SYSTEM WORKFLOW		22
3.1	Usecase Diagram	22
Chapter 4. SYSTEM DESIGNS		23
4.1	Module Descriptions	23
4.2	Input Design	24
4.3	Database Design	24
4.4	Output Design	26
4.5	User Interface Design	26

CONTENTS		
No	TITLE	Page
Chapter 5. AGILE TECHONOLGY OVERVIEW		27
5.1	Introduction to Scrum	27
5.2	Principles or Methodology Used	27
5.3	Sprint	29
Chapter 6.CODING		36
6.1	Coding Standards Followed	36
Chapter 7.CODING REVIEW AND TESTING		37
7.1	Code Review	37
7.2	Testing Process	38
Chapter 8.IMPLEMENTATION		39
8.1	New System Description	39
8.2	New System Implementation	39
Chapter 9. CONCLUSION		40
9.1	Increment Definition	41
	Further Enhancement	41
Chapter 10. APPENDIX		42
10.1	Screen Shots	43
10.2	Sample Code	4

## ACKNOWLEDGEMENT

We offer our humble salutations to the Almighty God whose abundant grace and mercy enabled the successful completion of our project. We take this opportunity to express our deep sense of gratitude, our whole hearted thanks and profound respect to all those who guided and inspired me in the completion of this projectwork.

We owe my deep sense of gratitude to the Management of Sree Narayana Gurukulam College of Engineering, providing the best suited academic environments for the fulfilment of my project. We would like to place on record my sincere thanks to **Dr.S JOSE** Principal of our institution, for his valuable comments and suggestions to this project.

We wish to express our sincere thanks to our Head of the Department **Prof. (Dr.) Sandhya R**, for providing me the opportunity to undertake this project. We are deeply indebted to our project guide **Asst.Prof. Meera K Chandran**, and project coordinator **Dr.AnoopKumar M** in the Department of Computer Applications for providing we with valuable advice and guidance during the course of the project. We thank all faculty members of our department and friends for extending their cooperation during our project.

Finally, We convey our thanks to each and every one, who helped us directly and indirectly to carry out this project successfully.

## **ABSTRACT**

Ecovault is a waste management application designed to streamline waste collection and categorization through a network of users, factories, and delivery personnel. The system comprises three main modules: the User Module, Delivery Module, and Factory Module, with the Factory Module further classifying waste into biogas, human waste, and plastic waste. The day begins as users encounter various household waste types and utilize Ecovault to manage their disposal needs. Users register on the app with their phone number, name, and location, then log in to submit waste collection requests to the nearest factory. Each request includes details such as waste type (biogas, human, or plastic), amount, and location. Based on the provided details, Ecovault identifies and notifies the nearest factory, which sends a confirmation response to the user that they are ready for waste pickup. Meanwhile, delivery personnel register on the app, providing their location, type of vehicle, and availability status (online or offline). The system assigns tasks to delivery personnel based on availability, proximity, and vehicle type. If no delivery personnel are available within a 10-meter range, the waste collection schedule rolls over to the next day. When a delivery personnel accepts a request, the app calculates the time required to reach the user based on the distance and factors such as traffic and network conditions (e.g., 1 km typically equates to 8 minutes). The user receives live updates on the delivery personnel's ETA. Upon arrival, the delivery personnel verifies the user's OTP, picks up the waste, and transports it back to the factory, where they receive payment based on distance and completed tasks. Factories update the status of delivery personnel to "available" post-completion. Additionally, users with regular waste collection needs can book monthly schedules, specifying the collection dates, waste type, and amount. Ecovault, thus, provides a comprehensive solution for managing waste collection through efficient coordination between users, delivery personnel, and factories, promoting a cleaner and more sustainable environment.

**ECOVAULT**



# 1. INTRODUCTION

In Ecovault is an innovative waste management solution developed to address the growing need for efficient waste disposal and resource management. In many urban and rural areas, the accumulation of household and industrial waste has become a significant environmental challenge leading to pollution, health hazards, and resource depletion. Ecovault is designed as a digital platform to streamline and simplify the waste management process, making it accessible, user-friendly, and environmentally responsible.

The project focuses on bridging the gap between users who need waste disposal services and facilities that process and recycle waste. By leveraging a structured system comprising three main modules—User Module, Delivery Module, and Factory Module—Ecovault ensures seamless communication and task delegation across different roles. The Factory Module further categorizes waste into types such as biogas, human waste, and plastic waste, enabling efficient processing and recycling.

The platform allows users to register, log in, and request waste pickups directly from the nearest processing facilities. Delivery personnel, equipped with real-time location tracking, handle waste collection tasks based on proximity, availability, and vehicle type. Through efficient scheduling, automated assignment, and an optimized payment system, Ecovault provides a smooth, reliable waste collection experience, reducing delays and improving operational efficiency.

This project not only enhances convenience for users and efficiency for service providers but also contributes to broader sustainability goals by promoting systematic waste segregation and eco-friendly disposal practices. Ecovault stands as a transformative solution that encourages responsible waste management and a cleaner, healthier environment.

## 1.1 PROJECT BACKGROUND

India's With rapid urbanization and population growth, waste management has become a critical issue impacting both urban and rural communities. Traditional waste management systems often struggle with inefficiencies, inadequate infrastructure, and a lack of systematic waste categorization, leading to environmental pollution, health risks, and missed opportunities for recycling and resource recovery. In response to these challenges, Ecovault was conceived as a comprehensive digital solution designed to modernize and improve waste management processes.

Ecovault focuses on creating a networked system that connects waste producers—individuals, households, and businesses—with factories capable of processing and categorizing waste into specific types, such as biogas, human waste, and plastic. By integrating real-time data and location-based services, the platform allows users to easily schedule waste pickups, ensuring that waste is directed to the nearest and most suitable facility. Delivery personnel are integral to the system, handling waste collection with optimized routes and availability tracking to minimize delays and ensure timely pickups.

The system's background lies in the need to overcome the limitations of manual waste collection and inconsistent waste disposal practices. Ecovault's design emphasizes simplicity, accessibility, and automation, making waste management more manageable and environmentally sustainable. By categorizing waste at the source, promoting scheduled pickups, and providing real-time tracking, Ecovault aims to reduce the environmental footprint of waste while contributing to the development of a cleaner and more sustainable community.

## **2.SYSTEM STUDY AND ANALYSIS**

### **2.1 ANALYSIS OF EXISTING SYSTEM**

The existing system, represented by the Haritha kerma sena initiative in Kerala, focuses primarily on the collection and disposal of plastic waste, addressing a specific aspect of waste management.

While it has been effective in reducing plastic pollution and promoting responsible disposal practices within the community, its limited scope means that other types of waste—such as biodegradable and human waste—are not adequately managed. This restriction prevents Harithasena from fully addressing the diverse waste generation from households and businesses, which often produce mixed waste streams requiring different disposal methods. Consequently, there is a need for a more comprehensive waste management system that can handle multiple waste types, optimize collection and processing, and promote sustainable practices on a broader scale.

### **2.2DRAWBACKS OF EXISTING SYSTEM**

#### **Limited Waste Type Coverage:**

Harithasena focuses exclusively on plastic waste, neglecting other significant waste types such as biodegradable, human, and electronic waste, which also contribute to environmental pollution and require specialized disposal methods.

**Inadequate Waste Segregation:**

Due to its focus on only one type of waste, the existing system lacks the necessary infrastructure and processes to handle mixed or segregated waste efficiently, limiting its adaptability for diverse waste management needs.

**Restricted Environmental Impact:**

By addressing only plastic waste, the current system has a limited scope in reducing overall environmental pollution. Other harmful waste types remain unmanaged, which diminishes the overall environmental benefits.

**Lack of User Engagement Features:**

The system does not incorporate features like reporting or incentivizing responsible waste disposal practices. Without user-driven functionalities, community engagement and active participation are limited.

**Limited Scalability:**

Due to its focus on a single waste type, the current system may face challenges if expanded to accommodate multiple types of waste, as it lacks a framework for handling diverse waste streams and the infrastructure to support such scalability.

**2.3 PROBLEM STATEMENT**

In contemporary waste management systems, the challenges of efficiently managing and categorizing diverse waste types have become increasingly pronounced. Existing systems, such as Harithasena in Kerala, primarily focus on a single type of waste (plastic), which limits their ability to address the broader spectrum of waste produced by households and businesses. The lack of a comprehensive, multi-stream waste management platform exacerbates these problems, making it difficult to efficiently collect, process, and recycle different types of waste.

**Key Problems:****1. Limited Waste Type Coverage:**

Existing systems primarily focus on plastic waste, neglecting biodegradable, human, and other types of waste that are equally important for effective waste management.

## 2. **Inefficient Waste Segregation:**

Without an integrated approach to handling various waste types, waste is often mixed, leading to inefficiencies in disposal, recycling, and processing.

## 3. **Lack of User Engagement:**

There is minimal user involvement in waste management processes, and no tools for reporting littering or incentivizing responsible waste disposal.

## 4. **Fragmented Waste Collection:**

The absence of coordination between users, delivery personnel, and factories results in fragmented waste collection services, causing delays and missed pickup opportunities.

## 5. **Limited Real-Time Tracking:**

Lack of real-time tracking and updates on waste collection schedules and delivery status hampers the ability to adapt to changes and make timely decisions regarding waste management.

## 6. **Inconsistent Service Quality:**

Variations in service quality across different delivery personnel and waste processing facilities lead to inconsistencies in waste collection and processing, undermining users' confidence in the system.

## 7. **Inadequate Planning for Regular Waste Collection:**

The existing systems do not allow for long-term waste collection planning, such as pre-scheduled pickups or management of recurring waste collection needs.

## **2.4 PROPOSED SYSTEM**

The proposed **EcoVault** system is an innovative, app-based platform designed to offer a comprehensive and user-friendly solution for managing waste collection and disposal. The key components and features of the proposed system are as follows:

### **Multimodal Waste Collection:**

The app will enable users to request waste collection for various waste types, including biodegradable, human, and plastic waste. The system will identify the most appropriate collection vehicle and route based on the user's location, type of waste, and available delivery personnel, ensuring efficient and timely pickups.

**Real-time Tracking and Updates:**

EcoVault will integrate real-time tracking of delivery personnel and waste collection vehicles. Users will receive live updates on the status of their waste collection, including estimated arrival times, delays, and any adjustments to the schedule, allowing them to plan accordingly.

**Interactive Maps**

The app will feature interactive maps that display waste collection routes, nearby processing facilities, and delivery personnel locations. This helps users track the status of their waste pickup in real-time and find available collection points nearby.

**Waste Categorization and Comparison:**

The system will allow users to categorize their waste as biodegradable, human, or plastic waste and provide suggestions for the most efficient way to dispose of each type. Users will also be able to compare different waste processing options available based on location, cost, and efficiency, helping them make informed decisions.

**User Profiles:**

Users can create personalized profiles, which store their preferences, such as the type of waste generated, preferred collection schedules, and recycling habits. This personalization will streamline the request process for recurring waste collection needs and offer tailored waste management solutions.

**Citizen Reporting and Incentives:**

The app will include a citizen reporting feature that allows users to report littering or improper waste disposal in their areas. Offenders may be fined, while users who actively engage in keeping the environment clean can earn rewards, incentivizing responsible waste management.

**Scheduled Waste Collection:** For users with regular waste collection needs, EcoVault will offer the ability to pre-schedule waste pickups. Users can specify the type and amount of waste, along with the date and time of collection, ensuring a more organized and efficient waste management process.

**2.5 FEASIBILITY STUDY**

A feasibility study is a critical step in assessing the viability of a project. In the context of an app-based travel guide, here are key elements to consider

### **2.5.1 Technical Feasibility**

**App Development:** Evaluate the technical requirements for developing the app, including the choice of platforms (iOS, Android), required features, and integration with transportation data sources.

**Data Sources:** Assess the availability and accessibility of transit data (e.g., real-time updates, schedules, fares) from various transportation providers.

**Scalability:** Consider the scalability of the app to accommodate a growing user base and increasing data volumes.

#### **Economic Feasibility**

**App Cost-Benefit Analysis:** An in-depth analysis of the costs involved in development, maintenance, and marketing against the expected benefits in terms of user adoption and improvements in urban transportation efficiency.

**Return on Investment (ROI):** Assessing the potential financial returns and the time required to recoup the initial investment.

### **2.5.2 Operational Feasibility**

**User Acceptance:** Evaluating the likelihood of user acceptance and adoption through surveys, focus groups, or market research.  
**Integration with Existing Systems:** Ensuring compatibility and seamless integration with existing transportation infrastructure and services

### **2.5.3 Legal Feasibility**

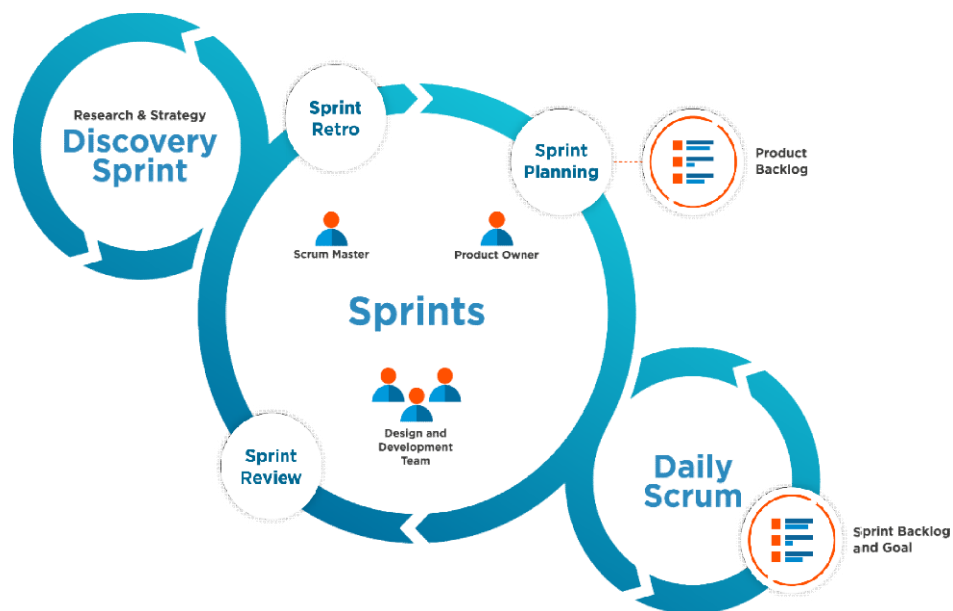
**Based Compliance:** Ensuring compliance with local regulations and laws related to data privacy, transportation services, and app development.

**Intellectual Property:** Addressing any potential legal issues related to intellectual property rights, especially if integrating with existing transportation services.

## 2.6 Project Planning

The project will go through the following stages of development in its software development life cycle.

Figure 2.6.1 Project Planning



## **2.7 Development Environment/ Technology/Framework**

### **2.7.1 Sublime Text**

Sublime Text is a highly regarded text editor known for its speed, versatility, and extensive customization options. Despite being classified as a text editor rather than a full-fledged integrated development environment (IDE), it packs a punch in terms of functionality and coding capabilities. One of the standout features of Sublime Text is its remarkable speed and performance. It's designed to be incredibly fast and responsive, allowing for seamless navigation and editing, even when working with large files. The editor's lightweight and efficient core contribute to its swift and fluid coding experience.

### **2.7.2 HTML**

Hypertext Mark-up Language (HTML) is the standard mark-up language for creating webpages and web applications. With Cascading Style Sheets (CSS) and JavaScript it forms a triad of cornerstone technologies for the World Wide Web. Web browsers receive HTML documents from a web server or from local storage and render them into multimedia webpages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

### **2.7.3 CSS**

CSS, or Cascading Style Sheets, is a crucial technology used in web development to define the visual appearance of web pages. It works hand-in-hand with HTML and allows developers to control the layout, colours, fonts, and other visual aspects of elements on a webpage. With CSS, you can specify how HTML elements should be displayed, giving you the ability to create visually appealing and well-structured websites. It offers a wide range of styling options, such as setting element dimensions, margins, and paddings, defining borders and backgrounds, and controlling typography and text formatting.



#### **2.7.4 GIT- version control**

### **VERSION CONTROL SYSTEM – VCS**

A version control system (VCS) allows you to track the history of a collection of files. It supports creating different versions of this collection. Each version captures a snapshot of the files at a certain point in time and the VCS allows you to switch between these versions. These versions are stored in a specific place, typically called a repository.

### **LOCALIZED AND CENTRALIZED VCS**

A localized version control system keeps local copies of the files. This approach can be as simple as creating a manual copy of the relevant files. A centralized version control system provides a server software component which stores and manages the different versions of the files. A developer can copy (checkout) a certain version from the central server onto their individual computer.

Both approaches have the drawback that they have one single point of failure. In a localized version control system, it is the individual computer and in a centralized version control system it is the server machine. Both systems make it also harder to work in parallel on different features.

### **DISTRIBUTED VCS**

In a distributed version control system each user has a complete local copy of a repository on his individual computer. The user can copy an existing repository. This copying process is typically called cloning and the resulting repository can be referred to as a clone. Every clone contains the full history of the collection of files and a cloned repository has the same functionality as the original repository.

### **GIT**

Git is currently the most popular implementation of a distributed version control system. Git originates from the Linux kernel development and was founded in 2005 by Linus Torvalds. Nowadays it is used by many popular open source projects, e.g., the Android or the

Eclipse developer teams, as well as many commercial organizations. The core of Git was originally written in the programming language C, but Git has also been re-implemented in other languages, e.g., Java, Ruby and Python.

## **2.8 BACKEND**

### **2.8.1 DATABASE SERVERS**

A database server is used to store data in a database. Users can access the data and manipulate it. There are many types of databases. Here we have used the MySQL database.

### **2.8.2 MySQL DATABASE**

MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

The MySQL database is open source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

The MySQL Database Server is very fast, reliable, scalable, and easy to use. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention.

### **2.8.3 PHP**

PHP, which stands for Hypertext Pre-processor, is a popular server-side scripting language used for web development. It is widely recognized for its versatility and is often embedded within HTML code or used in conjunction with various web frameworks. PHP enables developers to create dynamic web pages and build powerful web applications. It offers a broad range of features, including database integration, file handling, session management, and form processing. With PHP, you can generate dynamic content, interact with databases, handle user input, and perform various server-side tasks.

## **2.9 REQUIREMENT SPECIFICATION**

### **2.9.1 SOFTWARE REQUIREMENT**

The software requirements specification (SRS) is a means of translating the ideas in the minds of clients into a formal documentation. This document forms the development and software validation. The basic reason for the difficulty in software requirement specification comes from the fact that there are three interested parties—the client, the end users and the software developer. The requirements document has to be such that the client and the user can understand easily and the developers can use it as a basis for software development. Due to the diverse parties involved in software requirement specification, a communication gap exists. This gap arises when the client does not understand software or the software development processor when the developer does not understand the client's problem and application area. SRS bridges this communication gap.

Problem analysis is done to obtain a clear understanding of the needs of the clients and the users, and what exactly is desired from the software. Analysis leads to the actual specification. People performing the analysis called analysts, are also responsible for specifying the requirements. The software project is initiated by the client's needs. In the beginning these needs are in the minds of various people in the client organization. The requirement analyst has to identify their requirements by talking to these people and understanding their needs. These people and the existing documents about the current mode of operation are the basic source of information for the analyst.

## **2.9.2 HARDWARE REQUIREMENT**

Requirements analysis is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. This step acquiring all the facts problem specification such as identifying the desired result determining what information is needed to produce these results and figuring out what process must be carried out to proceed to get the accurate result.

## **2.10 SOFTWARE AND HARDWARE SPECIFICATION**

### **► HARDWARE REQUIREMENTS [ DEVELOPER SIDE]**

Display	13.5 Inch or Above
Hard Disk	250GB
Processor	Intel i3 or Above
RAM	Minimum 4GB

### **► SOFTWARE REQUIREMENTS [ DEVELOPER SIDE]**

IDE	SUBLIME TEXT
SERVER SIDE	Python
DATABASE SERVER	MySQL
OPERATING SYSTEM	MICROSOFT WINDOWS

### 3 SYSTEM WORK FLOW

#### 3.1 USECASE DIAGRAM

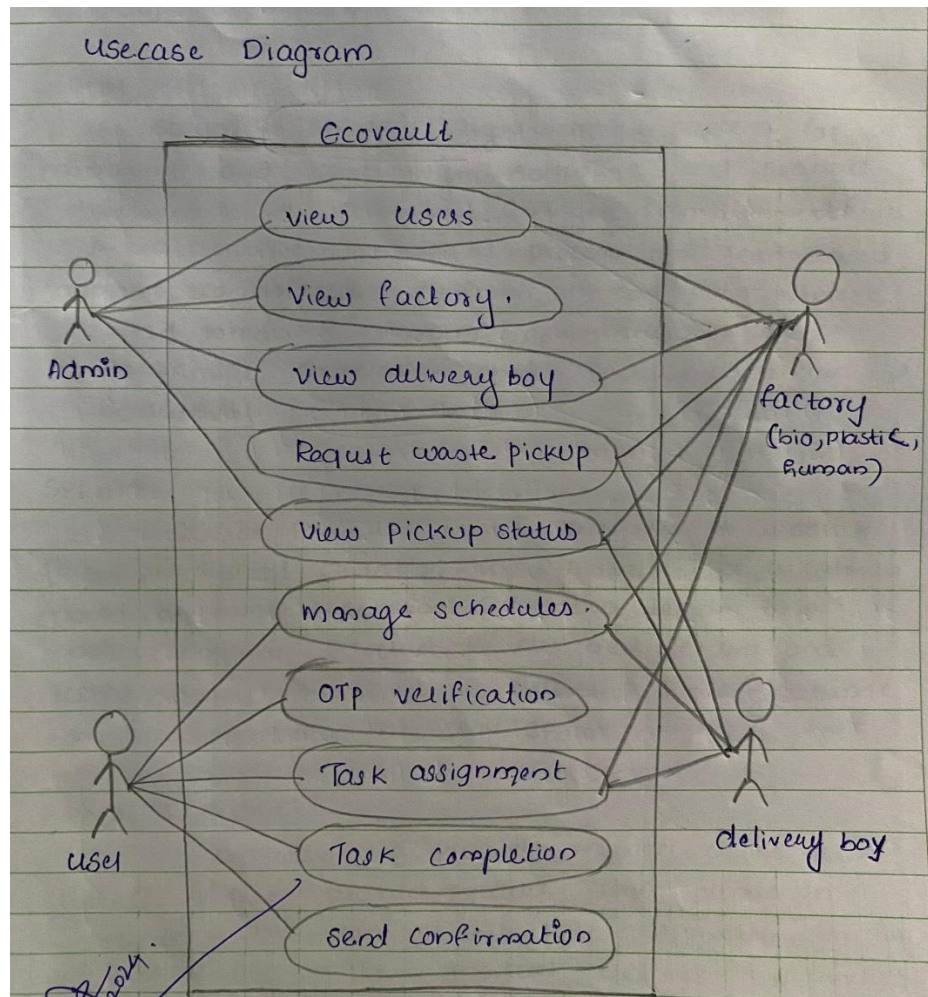


Figure 3.1 Use Case Diagram

## 4 SYSTEM DESIGN

The system design for the "Traveling Route Identification System" (Travel Guide) project is structured to deliver a comprehensive and user-centric solution for navigating urban transportation efficiently. The user interface is designed with map integration and intuitive navigation, ensuring ease of use as users input destinations and receive optimized routes. Real-time tracking, powered by GPS technology, facilitates live updates on the locations and estimated arrival times of buses, cabs, metros, and other transportation modes. The system relies on a robust database to store and manage comprehensive information on transportation routes, schedules, fares, and user profiles. An advanced route planning algorithm optimizes journeys based on user preferences, time constraints, and real-time traffic data, supporting seamless integration of multiple transportation modes. Modules for fare and time comparison assist users in making informed decisions, while a notification system provides alerts and personalized notifications. Security measures, including user authentication and data encryption, prioritize the protection of user information. A feedback and rating system encourages user engagement, contributing to continuous improvements in service quality. Accessibility features, scalability considerations, and integration APIs are incorporated to ensure adaptability, user-friendliness, and future expansion. Streamlined maintenance and updates, facilitated by automated mechanisms and version control, underscore the commitment to delivering a cutting-edge and reliable urban navigation solution for users in India.

### 4.1 MODULE DESCRIPTIONS

#### 1. User Module

The **User Module** allows end-users to interact with the system for waste management services. Its primary functions include:

**Registration:** Users can register by entering their phone number, name, and location.

**Login:** Registered users log into the app to access waste management services.

**Send Waste Pickup Request:** Users can submit a request specifying the type, weight, and location of waste. The request details help in assigning the nearest factory and appropriate vehicle.

**Payment & Feedback:** After waste pickup, users can complete payments based on distance and amount of waste collected. They can also provide feedback on the service.

**Logout:** Users can log out of their accounts after completing their tasks.

## 2. Admin/Factory Module

**Track Delivery:** Users can track the assigned delivery boy's estimated arrival time based on distance and location.

**OTP Verification:** Users provide a One-Time Password (OTP) to the delivery boy upon arrival, ensuring secure waste pickup.

The **Admin Module** is used by the factory to manage user requests and handle the operations related to waste collection. Key functionalities include:

**Manage Registration & Login:** Admins can log into the app to oversee waste management operations.

**Request Management:** Admins receive requests from users and assign the appropriate factory or vehicle type (e.g., for biogas, human waste, or plastic waste).

**Assign Delivery Boy:** Based on delivery boys' availability and location, requests are assigned to the nearest available delivery person.

**Status Updates:** The factory provides real-time updates on request statuses, including pickup and delivery confirmations.

**Manage Payments & Reports:** Admins process payments to delivery staff and generate reports on collections and operational efficiency.

**Logout:** Admins can log out upon completion of their tasks.

## 3. Delivery Boy Module

The **Delivery Boy Module** enables delivery staff to register and manage waste pickups based on their assigned tasks. This module includes:

**Registration:** Delivery boys register with phone number, vehicle type, and location.

**Set Availability:** Delivery boys mark their availability status, indicating whether they are online (available) or offline.

**Accept or Decline Pickup Task:** Based on location and availability, delivery boys are assigned waste pickup tasks that they can accept or decline.

**OTP Verification & Pickup:** Upon reaching the user location, delivery boys request the OTP for verification before collecting waste.

**Waste Delivery to Factory:** Once collected, the waste is transported to the assigned factory, where the delivery is confirmed.

**Receive Payment:** Delivery boys receive payments based on completed tasks and distance covered.

**Status Update:** After completing a task, delivery boys update their availability status for the next task.

## **4.2 INPUT DESIGN**

Input design converts user-oriented inputs to computer- based format, which requires careful attention. The collection of input data is the most expensive part of the system in terms of the equipment used and the number of people involved. In input design, data is accepted for computer processing and input to the system is done through mapping via some map support or links. The input design should require careful attention. In this project we use the input design as forms format. We give all the input in this forms format and the data is accepted for computer processing. All the data can be given very easily.

## **4.3 DATABASE DESIGN**

Database design is the process of producing a detailed data model of a database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

The process of doing database design generally consists of a number of steps which will be carried out by the database designer. Usually, the designer must:

- Determine the data to be stored in the database.
- Determine the relationships between the different data elements.
- Superimpose a logical structure upon the data on the basis of these relationships.



## NORMALIZATION

Normalization is the process of efficiently organizing data in a database. There are two goals of the normalization process: eliminating redundant data (for example, storing the same data in more than one table) and ensuring data dependencies make sense (only storing related data in a table). Both of these are worthy goals as they reduce the amount of space a database consumes and ensure that data is logically stored.

The database community has developed a series of guidelines for ensuring that databases are normalized. These are referred to as normal forms and are numbered from one (the lowest form of normalization, referred to as first normal form or 1NF) through five (fifth normal form or 5NF). In practical applications, you'll often see 1NF, 2NF, and 3NF along with the occasional 4NF. Fifth normal form is very rarely seen.

- **First Normal Form(1NF):** First normal form (1NF) sets the very basic rules for an organized database. Create separate tables for each group of related data and identify each row with the primary key.
- **Second Normal Form(2NF):** Second normal form (2NF) further addresses the concept of removing duplicate data.
- **Third Normal Form(3NF):** Third normal form (3NF) goes one large step further. Meet all the requirements of the second normal form and remove columns that are not dependent upon the primary key.
- **Boyce-Codd Normal Form (BCNF or 3.5NF):** The Boyce-Codd Normal Form, also referred to as the "third and half (3.5) normal form.
- **Fourth Normal Form(4NF):** Finally, fourth normal form (4NF) has one additional requirement. A relation is in 4NF if it has no multi-valued dependencies.

In my project, I have tried to follow the Third Normal Form(3NF) for all the tables.

#### **4.4 OUTPUT DESIGN**

The output design of the Ecovault Waste Management System is centered on providing clear, real-time information and feedback to users, admins, and delivery staff. Users receive notifications for registration, waste pickup confirmations, OTP verification, payment receipts, and service feedback prompts. Admins have a dashboard with real-time tracking of active requests, delivery assignments, status updates, and access to performance and financial reports. Delivery staff are notified of task assignments, route and ETA details, OTP verification prompts, and payment confirmations post-task. Automated scheduling reminders and traffic adjustment notifications ensure smooth operations and timely waste pickups. This output design enhances transparency, user experience, and operational efficiency across the system.

#### **4.5 USER INTERFACE DESIGN**

User interface design or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to it self. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements to create a system that is not only operational but also usable and adaptable to changing user needs.

## **5 AGILE TECHNOLOGY OVERVIEW**

### **5.1 Introduction to Scrum**

Scrum is a subset of agile. Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum itself is a simple framework for effective team collaboration on complex products. Scrum co-creators Ken Schwaber and Jeff Sutherland have written The Scrum Guide to explain Scrum clearly and succinctly. This Guide contains the definition of Scrum.

### **5.2 Principles or Methodology Used**

The SCRUM methodology is defined by team rules, events(ceremonies), artifacts and roles.

#### **5.2.1 Scrum Team**

A scrum team is a collection of individuals(typically between five and nine members)working together to deliver the requested and committed product increments. To work effectively it is important for a scrum team that everyone within the team follow a common goal. The Scrum Team share different tasks and responsibilities related to the delivery of the product. Each role is closely inter-related. It is recommended for Scrum team members work together in the same location whenever possible. There are 3 roles in a scrum team:

**1.The Product Owner:** The product owner is a project's key Stake holder-Usually an internal or external customer or a scope person for the customer. There is only one product owner who conveys the overall mission and vision of the product which the team is building. The product owner is ultimately accountable for managing the product backlog and accepting completed increments of work.

**2.The Scrum Master:** The scrum master Is the servant leader to the product owner,

development team and organization. With no hierarchical authority over the team but rather more of a facilitator, the scrum master ensures that the team adheres to scrum theory, practices and rules. The scrum master protects the team by doing anything possible to help the team perform of the highest level. This may include removing impediments, facilitating meetings and helping the product owner groom the backlog.

**3. The Development Team:** is a self-organizing, cross functional group armed with all of the skills to deliver shippable increments at the completion of each sprint.

### 5.2.2 Scrum events

Scrum events are time-boxed events that means in a project, every scrum events have a pre defined maximum duration. These events enable transparency on the project progress to all who are involved in the projects. The vital events of scrum are –the Sprint.

1. **The Sprint:** A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprint are usually 2-4 weeks long but can be as short as 1 week.
2. **Sprint Planning:** Sprint Planning team meetings are time-boxed events that determine which product backlog items will be delivered and have the work will be achieved.
3. **The daily Stand-up:** The daily stand-up is a short communication meeting in which each team member quickly and transparently covers progress since the last stand-up, planned work before the next meeting and any impediments that may be blocking his or her progress.
4. **The Sprint Review:** The sprint review is the “show and tell” of demonstration events for the team to present the work completed during the sprint. The product owner checks the work against predefined acceptance criteria or either accept or reject the work. The stake holder or client give feedback to ensure that the delivered incremental must be a business model.
5. **Retrospective:** The retrospective or retro is the final team meeting in the sprint to determine what went well, what didn’t go well and how the team can improve the next sprint. Attended by the team and the scrum master, the retrospective is an important opportunity for the team to focus on its overall

performance and identify strategies for continuous improvement on its process.

### 5.2.3 Scrum artifacts

Scrum artifacts are designed to increase transparency of information related to the delivery of the project, and provide opportunities for inspection and adaptation. They are management products useful for the creation of the specialist product of the project. There are 3 artifacts in scrum

1. **Product Backlog:** an extended list of everything that might be needed in the final product.
2. **Sprint Backlog:** Selected items for the product backlog to be delivered through a sprint, along with the task for delivering the item and realizing the sprint goal.
3. **The Sprint Review:** Increment set of all the product backlog items so far in the project.

### 5.2.4 Scrum rules

The rules of Agile Scrum Should be completely up to the team and governed for what works best for their processes. The best agile will tell teams to start with basic scrum events listed above and then inspect and adopt based on your team's unique needs so there is continuous improvement in the way teams work together.

## 5.3 SPRINT

### 5.3.1 Product backlog

In the simplest definition the Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user centric e.g. in the form of user stories. Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the development Team collaborate on the details of Product Backlog items. A Scrum product

backlog contains descriptions of the functionality desired in an end product. Agile backlog prioritization is the next step. The Product Backlog is the tool used by the product owner to keep track of all of the features that stakeholders would like to see implemented in the product whereas the Sprint Backlog is a subset of the Product Backlog representing the current active Sprint iteration. However, the product owner prioritizes it/

FEATURE ID	USER STORY	ESTIMATED HOURS	PRIORITY
1	As a user, I want to registerfor an account and login to my account, I want to login successfully I can send request for pickup waste and submit the details (location amount ,type)	8 hrs	high
2	As a factory manage I want the system assign the nearest factory based on waste type and assign the delivery boy based in available , location and vehicle	8 hrs	High
3	As a user I want to see time for delivery boy pickup , delivery person reach the location and collect the waste	4 hrs	low
4	As a delivery boy I want to verify the OTP before pickup and the collect the waste and it will deliver to factory	8 hrs	Medium
5	As a delivery boy I want my payment calculate based on the distance and complete task the payment has given by factory	8 hrs	Medium
6	As a delivery boy I want to update my available status after the completing task	4 hrs	low

**Table 5.1 Product backlog**

### 5.3.2 sprint planner

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box. The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. As the Development Team works, it does so with the Sprint Goal always in mind.

Feature ID	Sprint	Start Date	End Date	Estimated Work(hrs)	Sprint Goal	Acceptance criteria
01	1,2	28/08/24	11/09/24	8	Set up user Registration And login Functionality	User can register And login with Valid users
02	3,4,5	12/09/24	25/09/24	8	Waste request Submit, assign Factory based Location and Assign task	User can submit Waste request are Assign to nearest Factory based On waste type And location
03	6,7	26/9/24	9/10/24	8	Time calculation for pickup, Otp verification Before collecting.	User can Estimation Arrival time for Delivery otp Verification is

						Function before collection
04	8,9,10	12/10/24	28/11/24	8	Develop confirmation notification for waste collection and payment calculation for delivery personal	User receive collection . confirmation ,delivery personal receive payment based on distance and completed task
05	11,12,13	29/10/24	8/11/24	8	Implement monthly Collection scheduling update delivery personal availability add factory	Users can schedule monthly pickup,delivery personal status updates works factorized see real time status updates

**Table 5.2 Sprint Planner**



### 5.3.3 Ideal Burn Down Chart

A burn down chart is a graphic representation of how quickly the team is working through a customer's user stories, an agile tool that is used to capture a description of a feature from an end-user perspective. The burn down chart shows the total effort against the amount of work for each iteration. The quantity of work remaining is shown on a vertical axis, while the time that has passed since beginning the project is placed horizontally on the chart, which shows the past and the future. The burn down chart is displayed so everyone on the team can see it and is updated regularly to keep it accurate. There are two variants that exist for a burn down chart. A sprint burn down is for work remaining in the iteration. When illustrating the work remaining for the entire project, the chart is called a product burn down.

The burn down chart has several points. There's an x-axis, which is the project or iteration timeline. The y-axis is the work that needs to get done in the project. The story point estimates for the work that remains is represented by this axis. The project starting point is the farthest point to the left of the chart and occurs on day zero of the project or iteration. The project end point is farthest to the right and marks the final day of the project or iteration.

There is an ideal work remaining line, which is a straight line connecting the start point to the end point. It shows the sum of the estimates for all the tasks that need to be completed. At the end point, the ideal line crosses the x-axis and shows there is no work left to be done. This line is based on estimates and therefore not always accurate. Then there is the actual work remaining line that shows the actual work that remains in the project or iteration.

At the beginning the actual work remaining and the ideal work remaining are the same, but as the project or iteration progresses the actual work line will fluctuate above and below the ideal work line. Each day a new point is added to this line until the project or iteration is done to make sure it's as accurate as possible. If the actual work line is above the ideal work line, it means there is more work left than originally thought. In other words, the project is behind schedule. However, if the actual work line is below the ideal work line, there is less work left than had been predicted and the project is ahead of schedule.

### 5.3.4 Git Hub Registration

GitHub is an online-browser based distributed version control system for software developers using the Git revision control system. The service provides free public repositories, issue tracking, graphs, code review, downloads, wikis, collaborator management, and more. GitHub offers free accounts for users and organizations working on public and open source projects, as well as paid accounts that offer unlimited private repositories and optional user management and security features. It hub account creation includes the following steps:

- Go to the GitHub sign up page, then Enter a username, valid email address, and password. Use at least one lowercase letter, one numeral, and seven characters.
- Review carefully the GitHub Terms of Service and Privacy Policy before continuing and Choose a plan. Hereby anyone can finish the account creation procedure.
- You can store a variety of projects in GitHub repositories, including open source projects.
- In the upper-right corner of any page, click, and then click New repository.
- Type a short, memorable name for your repository followed by Optionally, add a description of your repository, public or private repository.
- Select Initialize this repository with a README. Finally Click Create repository.
- After creation, need to collaborate members by the admin.
- In the left sidebar, click Collaborators and teams.
- Under” Collaborators”, type the name of the person you’d like to give access to the repository, then click Add collaborator.
- Next to the new collaborators name, choose the appropriate permission level: Write, Read or Admin.
- The user will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.

## **6. CODING**

The above system is translated into a machine-readable form which is termed as coding. It is basically translating the human readable format to a machine friendly one. The code generation step performs this task:

The following points are considered while converting the system design into coding:

- Are the initializations correct?
- Are the data types properly assigned?
- Is memory leak being dealt with?
- Does it comply with the coding standard?

### **6.1 CODING STANDARDS FOLLOWED**

Coding standardization basically is the efficiency of our code which has been converted from the system design. The efficiency primarily depends upon:

- Readability: The code should be readable with proper indentation and spacing to make the contents clear of all the modules.
- Portability: The code is portable enough as it will work on various platform given all the necessary dependencies are installed?
- Debug Easily: The coding should be error-free as much as possible.

## **7.CODE REVIEW AND TESTING**

### **7.1 CODE REVIEW**

Software Testing is the process of executing a program or system with the intent of finding errors. Testing involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. The scope of software testing includes examination of code as well as execution of that code in various environments and conditions as well as examining the quality aspects of code: does it do what it is supposed to Do and do what it needs to do. Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases.

Testing Objectives include:

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has a probability of finding an as yet undiscovered error.
- A successful test is one that uncovers an undiscovered error.

Testing Principles are:

- All tests should be traceable to end use requirements
- Tests should be planned long before testing begins
- Testing should begin on a small scale and progress towards testing in large
- Exhaustive testing is not possible
- To be most effective testing should be conducted by an independent third party.

Implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main workload, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned and controlled, it can cause chaos and confusion. The implementation stage involves the following tasks:

- Careful planning.
- Investigation of system and constraints. of methods to achieve the changeover.
- Training of staff in the changeover phase.
- Evaluation of the changeover method.

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the same time users are trained in the new procedures.

## **7.2 TESTING PROCESS**

Testing helps not only to uncover errors introduced during coding, but also locates errors committed during the previous phases. Thus the aim of testing is to uncover requirements, design or coding errors in the program. Software Testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting anomaly for the software engineer.

## **UNIT TESTING**

This is the first of testing. In this different modules are tested against the specification produces during the design of the modules. It refers to the verification of single program module in an isolated environment. Unit testing focuses on the modules independently of one another to locate errors.

In our project we test each module and each forms individually. Each forms may have tested using appropriate values. The input screens need to be designed very carefully and logically. While entering data in the input forms, proper validation checks are done and messages will be generated by the system if incorrect data has been entered.

## **VALIDATION CHECKS**

As a web application developer, form validation is a crucial part of your work, and it should not be underrated as it could lead to security flaws in your application. You should consider it a must if you're striving to provide a professional end user experience.

- Custom error messages
- Basic form validation

It's always preferred to load common libraries and helpers in the constructor of the controller itself as it's a nice habit to avoid code duplication elsewhere in controller methods. We load the form and URL helpers so that we can use the utility methods provided by those helpers throughout the rest of the application. The form validation library in the constructor, you can access it using the form validation convention.

## **8.IMPLEMENTATION**

After the system has been tested, the implementation type or the changeover technique from the existing system to the new system is a step-by-step process. In the system at first only a module of the system is implemented and checked for suitability and efficiency. When the end user related to the particular module is satisfied with the performance, the next step of implementation is preceded.

Implementation to some extent is also parallel. For instance, modules which are not linked, with other modules are implemented parallel and the training is the step-by-step process. Backups are necessary since any time unexpected events may happen. And so during the program execution, the records are stored in the workspace. This helps to recover the original status of the records from any accidental updating or intentional deletion of records.

## **8.1 NEW SYSTEM DESCRIPTION**

College voting application is a system which developed with the intention of making the elections in a college online. This android system will enable the colleges to conduct elections online and the students will be able to vote through their mobile. The system is implemented in a safe and secure way that is suited for the safe conduction of election.

## **8.2 NEW SYSTEM IMPLEMENTATION**

An Implementation plan is a management tool for a specific policy measure, or package of measures, designed to assist agencies to manage and monitor implementation effectively.

Implementation plans are intended to be scalable and flexible; reflecting the degree of urgency, innovation, complexity or sensitivity associated with the particular policy measure.

The implementation stage involves following tasks:

- Careful planning
- Investigation of system and constraints
- Design of method to achieve the change over phase
- Evaluation of the change over method

## **9.CONCLUSION**

### **9.1 INCREMENT DEFINITION**

The Faculty Appraisal System is a valuable tool for educational institutions seeking to streamline and enhance their faculty appraisal process. By leveraging technology, it offers a range of features and benefits that promote fairness, transparency, and efficiency. The system eliminates manual tasks, standardizes evaluations, and provides accurate performance assessments, ultimately contributing to improved faculty management and institutional growth.

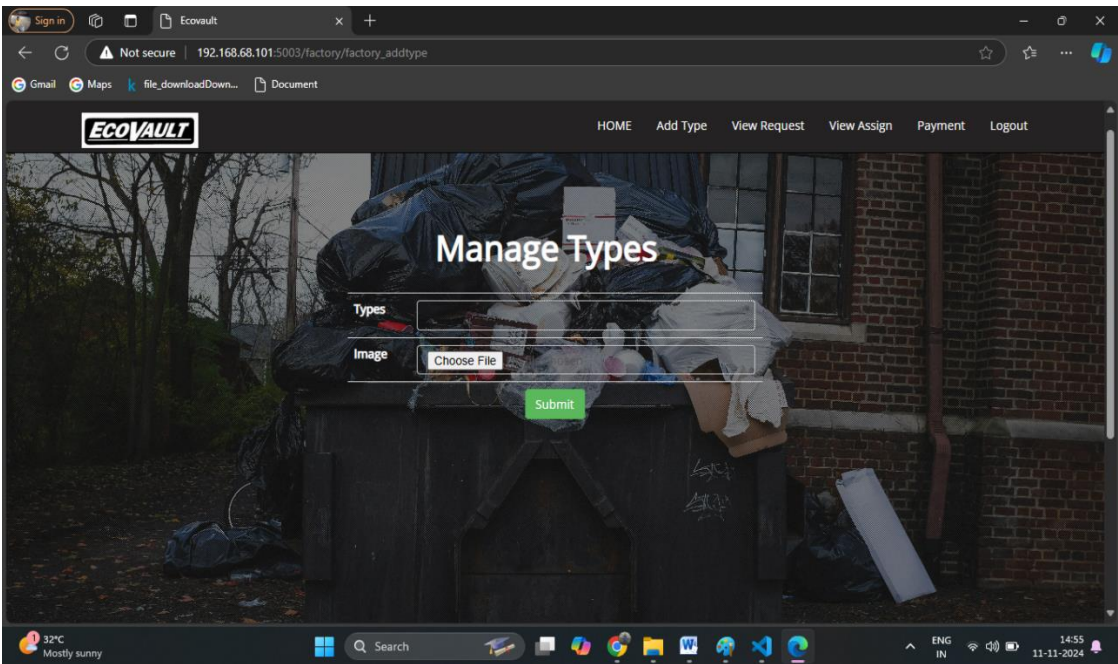
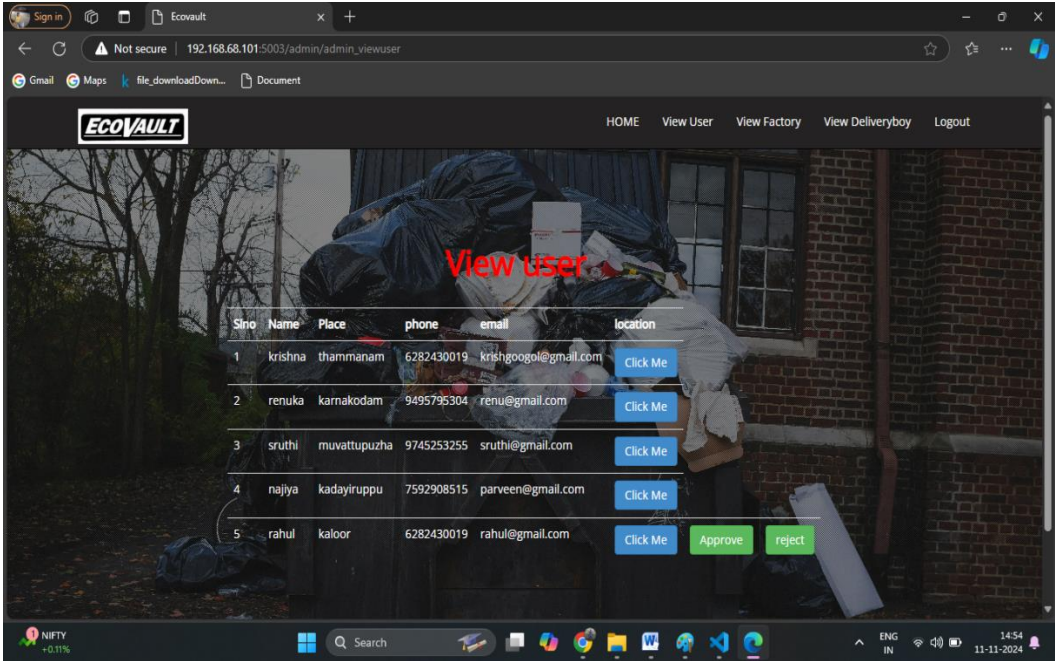
## **FURTHER ENHANCEMENT**

Every Edition of a book comes with new topics and modifications if any errors are present. In the similar way, in near future, my website will overcome any flaws if occurred, following are the Enhancements to the website.

1. Enhanced User Interface: Improve the user interface of the Faculty Appraisal System to enhance usability and user experience.
2. Multi-level Approval Workflow: Implement a multi-level approval workflow feature to accommodate different levels of hierarchy within the institution. This will enable a smooth and efficient appraisal process, ensuring appropriate reviews and approvals are obtained at each stage.
3. 360-Degree Feedback: Introduce a 360-degree feedback mechanism where faculty members receive feedback not only from their superiors but also from peers, subordinates, and students. This comprehensive feedback approach provides a holistic assessment of faculty performance and fosters a culture of continuous improvement.
4. Automated Reminders and Notifications: Enhance the system by incorporating automated reminders and notifications to keep faculty members and administrators informed about upcoming appraisal deadlines, pending reviews, and important updates. This feature will help in ensuring timely completion of appraisals and reducing manual follow-up efforts.

# 10. APPENDIX

## 10.1 SCREEN SHOTS





Sign in

Not secure | 192.168.68.101:5003/factory/factory\_viewpayment

GmailMapsfile\_downloadsDownloadDocument

ECOVault

HOMEAdd TypeView RequestView AssignPaymentLogout

View Payments

Sno	Delivery boy	Place	phone No	date	Amount	Status	total distance
1	siddharth suresh	pattimattam	8590641494	2024-10-30	297	available	20 Km
2	siddharth suresh	pattimattam	8590641494	2024-10-30	297	available	20 Km

32°C Mostly sunny

Search

ENG IN14:5711-11-2024

Sign in



Not secure | 192.168.68.101:5003/factory/factory\_viewassign

GmailMapsfile\_downloadsDownloadDocument

ECOVault

HOMEAdd TypeView RequestView AssignPaymentLogout

View Assigned

Sno	Delivery boy	Place	phone No	Waste type	Image	Status	Quantity
1	siddharth suresh	pattimattam	8590641494	bio waste		paid	60
2	siddharth suresh	pattimattam	8590641494	bio waste		paid	50

32°C Mostly sunny

Search

ENG IN14:5811-11-2024

Sign in



Not secure | 192.168.68.101:5003/factory/factory\_viewrequest

GmailMapsfile\_downloadsDownloadDocument

ECOVault

HOMEAdd TypeView RequestView AssignPaymentLogout

View Request

Sno	User	Place	Location	Waste type	Image	Status	Quantity
1	renuka	karnakodam	<a href="#">Click here</a>	bio waste		assign	60
2	renuka	karnakodam	<a href="#">Click here</a>	bio waste		assign	50

32°C Mostly sunny

Search

ENG IN14:5911-11-2024

4G 3:04 1.20 KB/s

Voice LTE 90%

## Ecovault

🔍 Nearest factory



**renuka**  
**9495795345**  
**renuka@gmail.com**  
**ernalulam**



**tata**  
**9495795345**  
**tata@gmail.com**  
**ernalulam**



**birla**  
**9495795345**  
**birla@gmail.com**  
**ernalulam**

  
HOME



4G 3:05 1.10 KB/s

VoLTE 90%

Ecovault

## View Types



type:bio waste

Send request

HOME



4G 3:05 2.30 KB/s

Von LTE 90%

## Ecovault

**Delivery Boy Status: paid**

Order ID: 2

Delivery Boy Status: paid

### ***View Details***



name:renuk  
a  
place:karna  
kodam  
phone:9495  
795304  
type:bio  
waste  
status:paid  
date:  
2024-10-30

  
HOME

Status: paid 



4G 3:06 2.90 KB/s

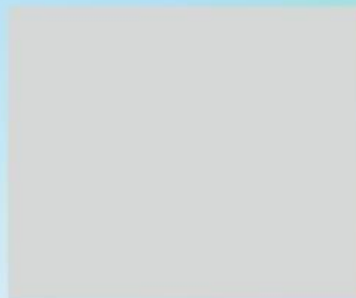
VoLTE 90%

Ecovault

## Add vehicles

vehicle type

capacity



SUBMIT

HOME



4G 3:07 2.90 KB/s

Von LTE 90%

Ecovault

**OTP**



### **Verification Code**

We have Sent a Verification Code to your mobile number

otp

PickUp SUCCESSFULLY

SUBMIT

## SAMPLE CODE

### MAIN.PY

```
from flask import Flask
from public import public
from admin import admin

from factory import factory

from api import api

app=Flask(__name__)

app.secret_key='key'

app.register_blueprint(public)
app.register_blueprint(admin,url_prefix='/admin')
app.register_blueprint(factory,url_prefix='/factory')

app.register_blueprint(api,url_prefix='/api')

app.run(debug=True,port=5003,host="0.0.0.0")
```

### public.py

```
from flask import *
from database import*
import uuid

public=Blueprint('public',__name__)

@public.route('/',methods=['post','get'])
def public_home():
    if "submit" in request.form:
        u=request.form['username']
        p=request.form['password']
        q="select * from login where username='%s' and password='%s'"%(u,p)
        res=select(q)
```



```

        if res:
            session['login_id']=res[0]['login_id']
            lid=session['login_id']

            if res[0]['usertype']=="admin":
                return redirect(url_for('admin.adminhome'))
            elif res[0]['usertype']=="factory":
                q="select * from factory where login_id='%s'"%(lid)
                res=select(q)
                if res:
                    session['factory_id']=res[0]['factory_id']
                    fid=session['factory_id']
                    return redirect(url_for('factory.factoryhome'))
            else:
                flash('invalid username and password')

        return render_template('index.html')
@public.route('/signup',methods=['post','get'])
def signup():
    if "submit" in request.form:
        f=request.form['fname']

        p=request.form['place']

        n=request.form['phone']
        e=request.form['email']
        u=request.form['uname']
        pa=request.form['pwd_confirm']
        la=request.form['latitude']
        lo=request.form['longitude']
        i=request.files['img']
        path="static/image/"+str(uuid.uuid4())+i.filename
        i.save(path)
        q="select * from login where username='%s' and password='%s'"%(u,pa)
        res=select(q)
        if res:
            flash('already exist')
        else:
            q="insert into login values(null,'%s','%s','pending')"%(u,pa)
            id=insert(q)
            q="insert into factory
values(null,'%s','%s','%s','%s','%s','%s','%s','%s')"%(id,f,p,n,e,la,lo,path
)
            insert(q)
            print(q)

```



```
flash('successfully')
return redirect(url_for('public.signup'))

return render_template('signup.html')
```

```
<h3>LOGIN !! </h3>

    <h2 ><span >SAVE EARTH</span> </h2>
    <div class="col-lg-12 col-md-12 col-sm-12" >
        <form method="post">
            <div class="form-group">
                <input type="text" class="form-control"
name="username" required="required" placeholder="Enter Your Username">
            </div>
            <div class="form-group">
                <input type="password" name="password"
class="form-control" required="required" placeholder="Enter Your Password">
            </div>

            <div class="form-group">
                <button
type="submit" name="submit" value="submit" class="btn btn-success btn-
block btn-lg">LOGIN NOW</button>
            </div>
        </form>

        <div class="form-group">
            <a href="signup">Do You Have An
Account?</a>

        </div>
    </div>
```

```

<center>
<h1>View factory</h1>
<table class="table" style="width: 500px;color: white">
  <tr>
    <th>Slno</th>
    <th>Name</th>
    <th>Place</th>
    <th>phone</th>
    <th>email</th>
    <th>location</th>

  </tr>
  {% for row in data['factory']%}
  <tr>
    <td>{{loop.index}}</td>
    <td>{{row['fname']}}</td>
    <td>{{row['fplace']}}</td>
    <td>{{row['fphone']}}</td>
    <td>{{row['femail']}}</td>

    <td><a class="btn btn-primary"
href="http://www.google.com/maps?q={{row['flatitude'] }},{{row['flongitude']
}}">Click Me</a></td>

    {% if row['usertype']=='pending'%}

    <td><a class="btn btn-success"
href="?action=accept&cid={{row['login_id']}}">Approve</a></td>
    <td><a class="btn btn-success"
href="?action=reject&cid={{row['login_id']}}">reject</a></td>
    {% endif %}
  </tr>

  {%endfor %}
</table>
</center>

```