PROJECT REPORT

Team ID	PNT2022TMID51107
Project Name	Smart Waste Management System for Metropolitan Cities
Date	19 November 2022
Team Lead	SAI VISHNU L
Team Member 1	MUGESH M
Team Member 2	MANOJKUMAR R
Team Member 3	NHIDHEES LAKSH KUMAR K.B

INTRODUCTION

1.1 PROJECT OVERVIEW

Project Name: Smart Waste Management System for Metropolitan Cities.

Category: Internet of Things.

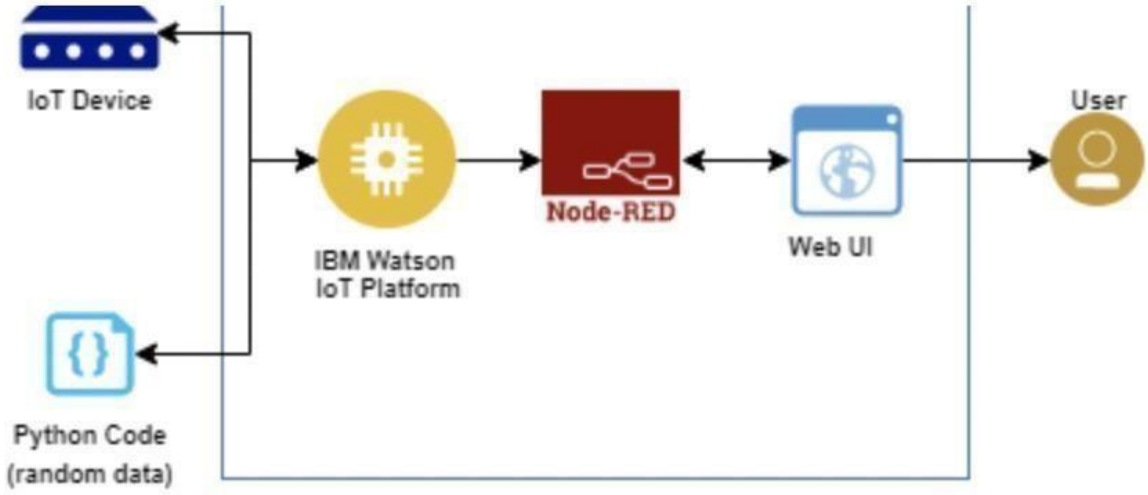
Project Description:

- Garbage level detection in bins.
- Getting the weight of the garbage in the bin.
- Alerts the authorized person to empty the bin whenever the bins are full.
- Garbage level of the bins can be monitored through a web App.
- We can view the location of every bin in the web application by sending GPS location from the device.

Skills Required:

Python, IoT Cloud Platform, IBM Cloud, Node-RED, IBM IoT Platform, IBM Node-red, IBM Cloudant DB

Technical Architecture:



1.2 PROJECT PURPOSE

Using the Internet of Things, the project's primary goal is to intelligently automate the waste management Process. The suggested system aims to assess the fill level of dry and wet dustbins and alert the garbage collection if the fill level rises beyond the 75% threshold.

Smart waste management is a concept that allows us to handle many issues that bother society, such as pollution and infections. Waste management must be handled right away to avoid irregular management, which would harm the environment. Smart cities as a concept are mostly compatible with smart waste management.

MAIN OBJECTIVES:

1. Monitoring waste management.
2. Providing smart technology for waste systems.
3. Avoiding human intervention.
4. Reducing human time and effort
5. Resulting in a healthy waste-ridden environment.

LITERATURE SURVEY

2.1 EXISTING PROBLEM

For the service providers, the procedure of garbage collection is crucial. The conventional method of manually checking the garbage in trash cans requires more human labour, takes longer, and costs more money. It is incompatible with modern technology. The improper handling of garbage—typically household waste, industrial waste, and environmental waste—is a major contributor to a number of human issues, including pollution, sickness, and a lack of hygienic conditions for all living things. We are putting out the concept of a smart waste management system to aid with the auto-management of garbage without human contact in order to maintain a clean environment in order to solve all these issues.

2.2 REFERENCES

PAPER 1:

TITLE: IoT Based Waste Management for Smart City

AUTHOR NAME: Parkash Tambare, Prabu Venkatachalam PUBLICATION

YEAR: 2016

DESCRIPTION:

The quantity of garbage created each day is increasing, and as a result, we regularly see that the trash cans or dust cans that are placed in public areas of cities are overflowing. We intend to build "IoT Based Waste Management for Smart Cities" to avoid this since it leads to unhygienic living conditions for people and offensive odours in the neighbourhood. The suggested system includes a large number of garbage cans that are dispersed across the city or on the campus. Each garbage can has a low-cost integrated gadget that monitors its level as well as a unique ID that makes it possible to track and identify it.

PAPER 2:

AUTHOR NAME: Mohammad Aazam, Marc St-Hilaire,
Chung-Horng Lung, Ioannis Lambadaris

PUBLICATION YEAR: 2016

DESCRIPTION:

The sensors in each bin of the Cloud SWAM system proposed by Mohammad Aazam et al can determine how much garbage is there within. For organic, plastic/paper/bottle/glass, and metal garbage, separate containers are available**.** In this manner, each type of garbage is already separated, and the status makes it clear how much and what kind of waste is collected. The accessibility of cloud-stored data may be advantageous to various organizations and stakeholders in various ways. As soon as the trash is picked up, analysis and planning may start, and they can continue throughout recycling and import/export-related operations. The Cloud SWAM system offers prompt waste pickup. The health, hygiene, and disposal conditions are improved by a timely and efficient garbage collection procedure.

PAPER 3:

AUTHOR NAME: K. Suresh, S. Bhuvanesh and B. Krishna Devan

PUBLICATION YEAR: 2019

DESCRIPTION:

This essay describes a method for cleaning up our atmosphere and surrounds. The waste collection and disposal system has to be upgraded for the Indian government's smart city plan to make these communities even smarter than they presently are. Self-Monitoring Automated Route Trash (SMART) dustbins are designed for usage in smart buildings, including among others universities, hospitals, and bus stations. In this project, we used the Servomotor to open the dustbin lid, the PIR and Ultrasonic sensors to detect human presence, and the Ultrasonic sensor to determine the amount of trash. A communication module is used to relay signals between two garbage cans, and the GSM module communicates the message to the operator.

2.3 PROBLEM STATEMENT DEFINITION

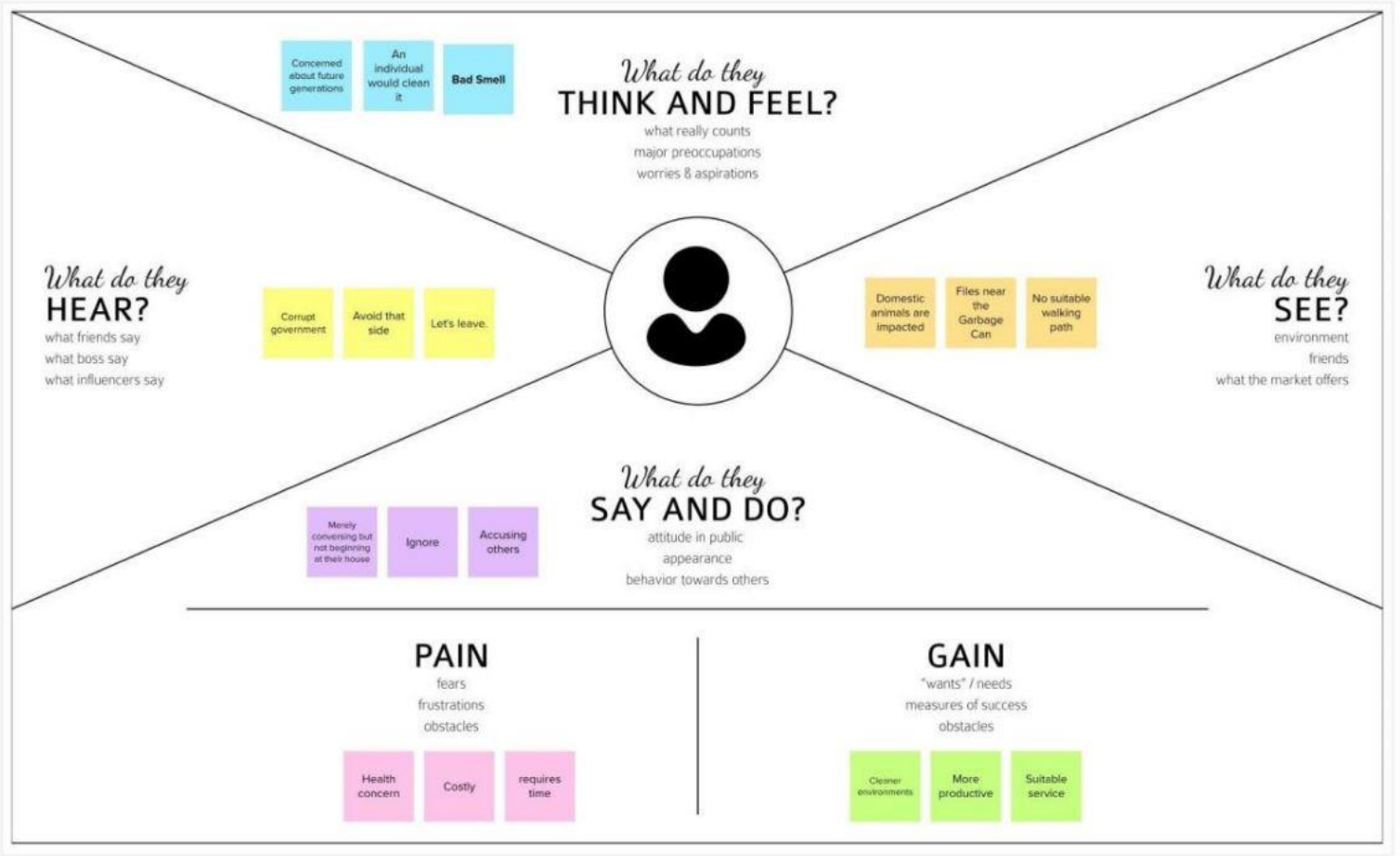
Garbage Management and Collection in Cities, Town and Villages is a major concern and emerging problem in the Smart City paradigm. Also lack of proper resource distribution in the process of Garbage collection is a great risk to sanitation, cleanliness and health.

Theme: Internet of Things

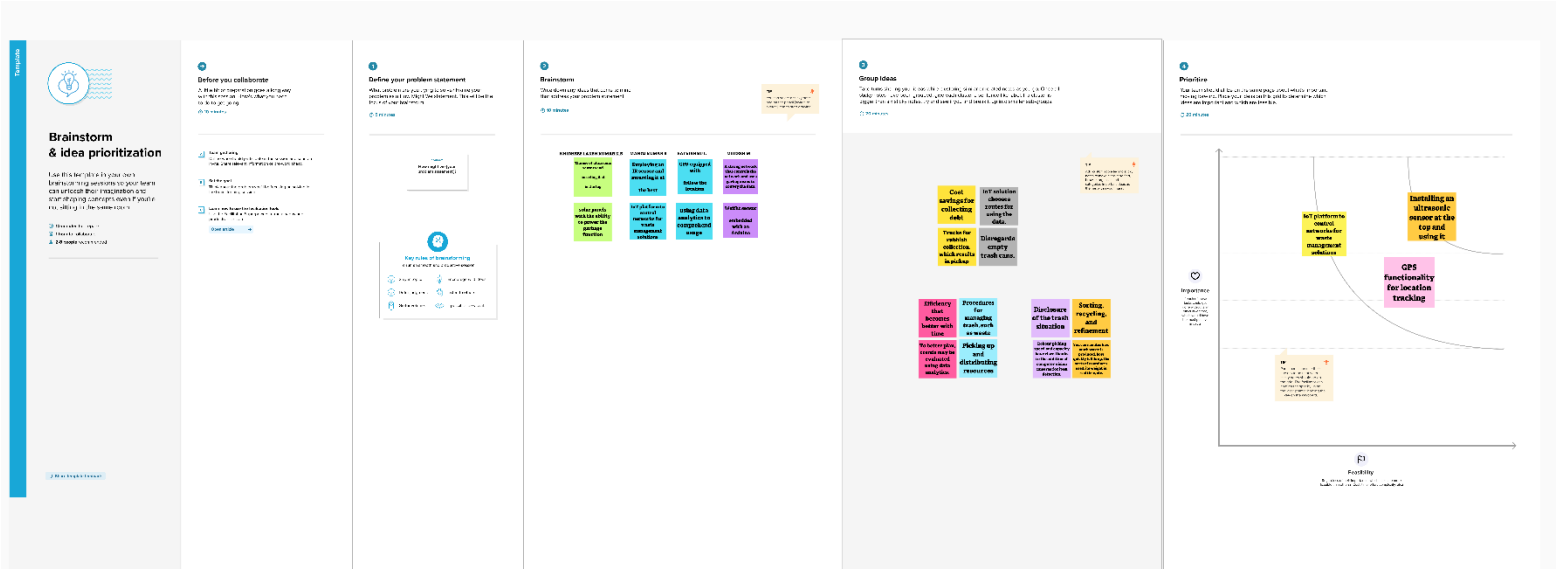
Technologies: LoRa, Smart Mesh, RF, WiFi

IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTORMING

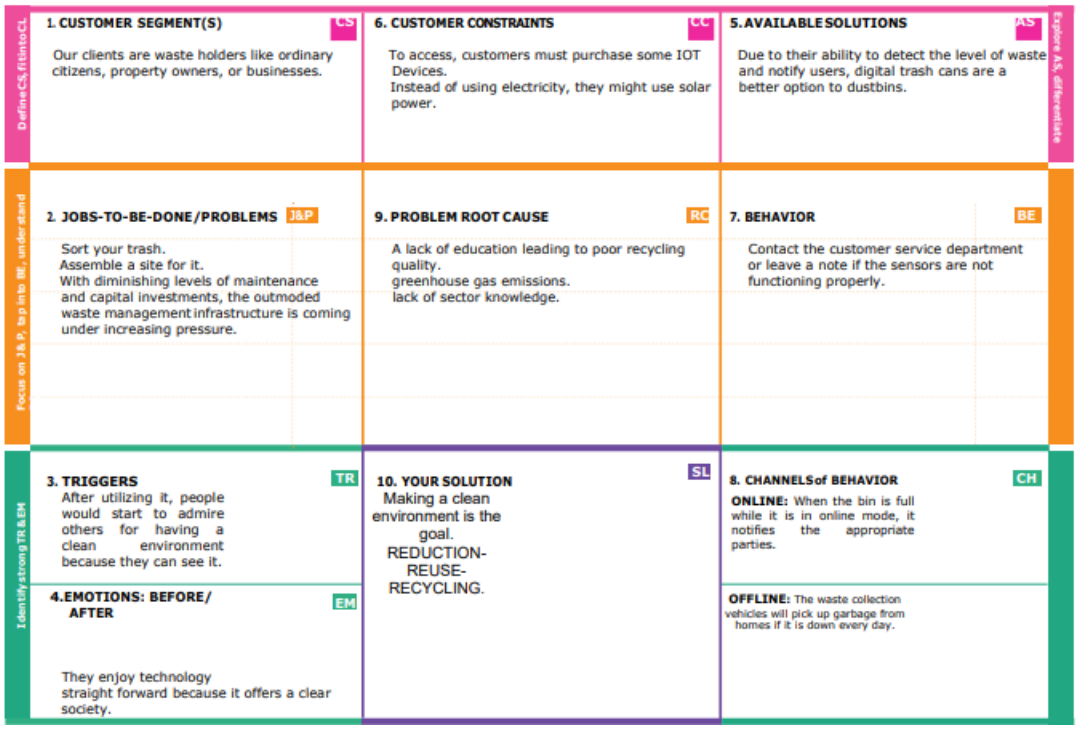


3.3 PROPOSED SOLUTION

S.NO	PARAMETER	DESCRIPTION
1.	Problem Statement (Problem to be solved)	This project addresses the issue of waste management in smart cities with inefficient garbage collection systems. This project helps the enterprises to fulfil their requirements for intelligent garbage management solutions. The authorised person can use this method to provide truck drivers with a time- and money-saving route by always knowing the level of fill in each garbage can in a neighbourhood or city.
2.	Idea / Solution description	The key research objectives are as follows: • The suggested system would be able to use IOT (Internet of Things) to automate the solid waste monitoring process and control of the whole collection operation. • The major components of the proposed system are the Smart Trash System (STS) and the Smart Monitoring and Controlling Hut (SMCH). • In the suggested system, the circuit is placed at the garbage bin to recognise when it is full. The circuit then communicates this information to the receiver, which is located at the desired location in the area or spot. • In the suggested method, the monitoring and control system's monitoring and controlling system receives a signal that shows the state of the trash bin.
3.	Novelty / Uniqueness	We intend to implement SWM in our college, but the real challenge is that the janitor (cleaner) doesn't know how to use this equipment practically. In this case, our team planned to build a wristband for them that alerts them when the dustbin is full via light blinking. This is another unique decision we made here in addition to the project's limitations.

4.	Social Impact / Customer Satisfaction	According to popular opinion, the direct social effects of current solid waste disposal procedures, such as the proximity of landfills to neighbourhoods, the development of pests, and the decline in property values, are the worst effects.
5.	Business Model (Revenue Model)	Solid Waste, which consists of the Company's waste collection, transfer, recycling, resource recovery, and disposal services. Corporate and Other, which consists of the Company's other activities, such as the development and operation of landfill gas-to-energy facilities in India. recycling brokerage services and various corporate functions.
6.	Scalability of the Solution	In order to address this issue, smart city design is being researched and debated more and more globally. Following this methodology, this article proposed a powerful IoT-based, real-time trash management model with an emphasis on citizens to enhance urban living conditions. The suggested method makes use of sensor and communication technologies, collecting garbage information from the smart bin in real-time and sending it to an internet site that city residents may visit to see whether the compartments are still available.

3.4 PROBLEM SOLUTION FIT



REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT(FR)

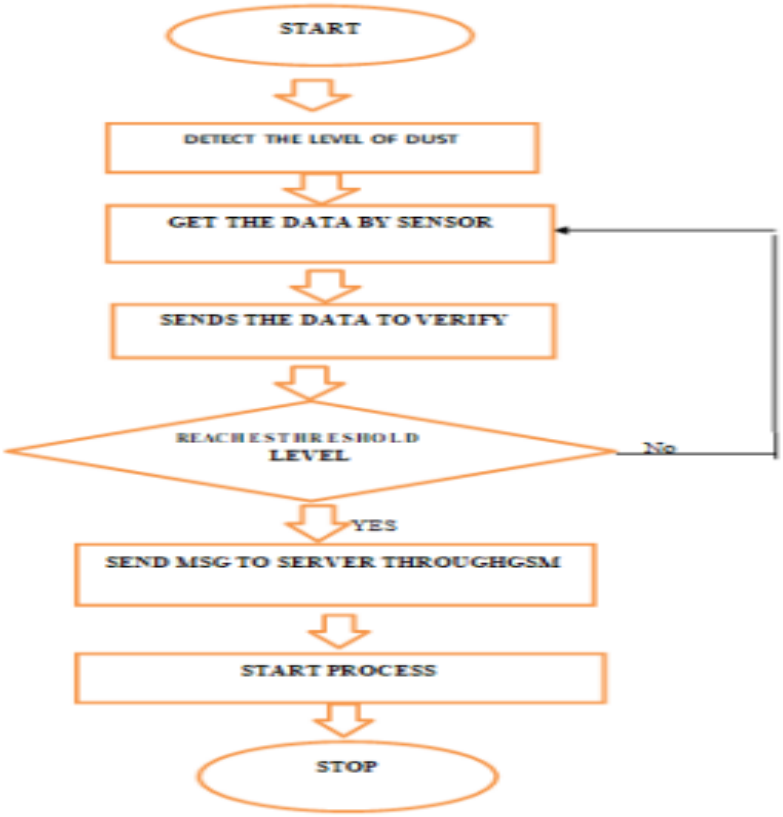
FR NO.	Functional Requirement (Epic)	Sub Requirement (Story / Sub Task)
FR-1	User Registration	Download application through online registration.
FR-2	User Confirmation	Confirmation via E-mail Confirmation via OTP
FR-3	Cloud	Details are stored in cloud & update periodically.
FR-4	Notifier	When bins are filled, notifications should be sent automatically.
FR-5	Sensor	Bins are filled sensor will monitor the wastage.

4.2 NON-FUNCTIONAL REQUIREMENTS (NFR)

FR NO.	Non-Functional Requirement	Description
NFR-1	Usability	It will stop overflowing of dustbins along roadsides and localities as smart bins are manage at real time.
NFR-2	Reliability	Details are maintained.
NFR-3	Performance	It seeks to produce both a clean and green environment.
NFR-4	Availability	For a clean atmosphere, this approach is accessible to all urban residents.
NFR-5	Scalability	The solution's scalability is achieved by continuing to monitor bins and offer guiding services based on the database.

PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Table-1: Components & Technologies:

S. No	Component	Description	Technology
1.	User Interface	IOT cloud platform from IBM.	MQTT Protocol
2.	Application Logic-1	Sensors are used to collect the data from the trash bins.	Python
3.	Application Logic-2	IOT is used to monitor the acquired data.	IBM Watson STT service

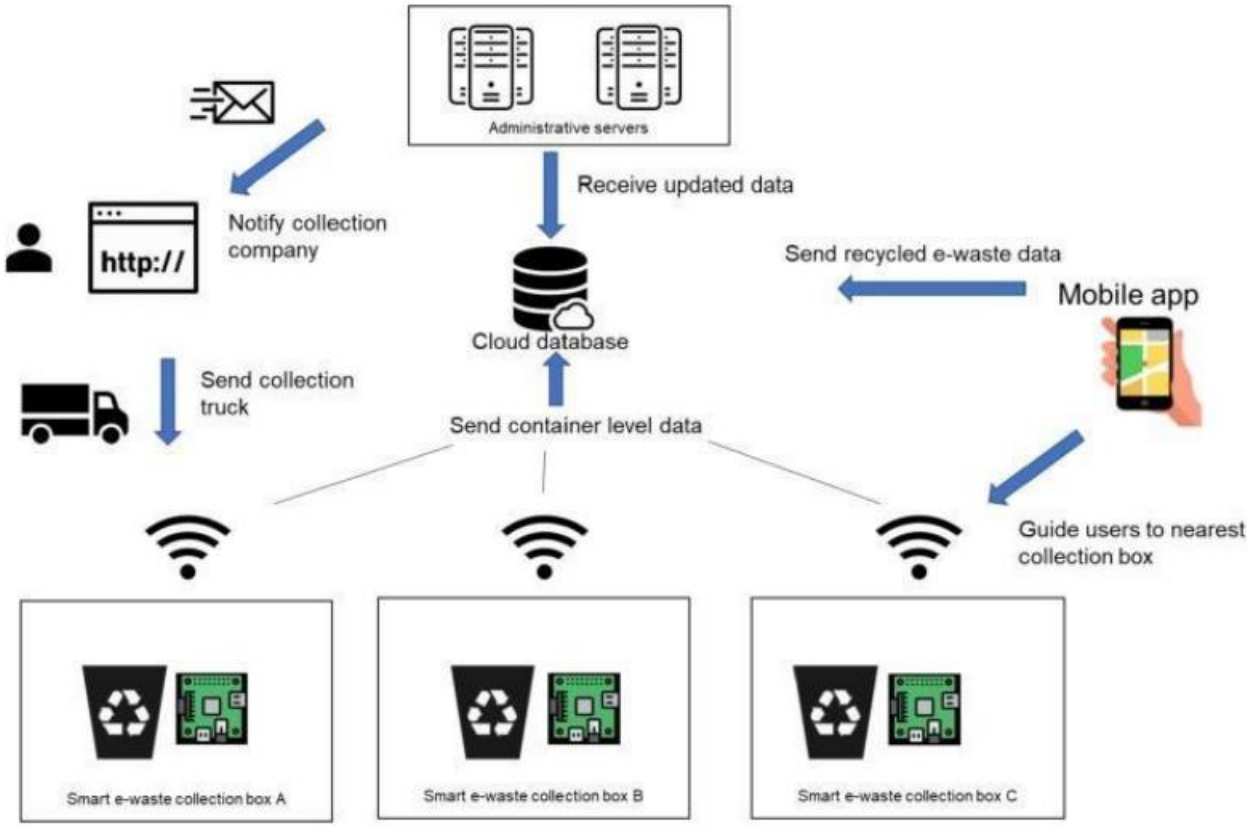
4.	Application logic -3	A warning message will be sent to the employees to dispose of the waste based on the data.	IBM Watson Assistant
5.	Database	□ MySQL is a relational database that is based on a tabular design. □ NoSQL is non-relational and has a document-based design.	MySQL, NoSQL
6.	Cloud Database	This module will continuously display real-time garbage bin status updates on the web application and send client-side notifications in addition to receiving real-time status updates from all of the garbage cans (Municipal Corporation, Garbage collector truck drivers etc.) mobile application.	IBM DB2, IBM Cloud
7.	File storage	Data storage makes it simple to back up files for storage and speedy recovery in the case of an unanticipated computer failure or cyberattack.	IBM Block Storage or Other Storage Service.
8.	External API 1	A project's internal resources can be made accessible to outside users or apps through external APIs.	IBM Weather API, etc.
9.	External API 2	You can use external API to access resources provided by third parties using RESTful web services.	Aadhar API, etc.
10.	Machine Learning Model	Planning is made effective by the right algorithm. It will provide direction for the excellent character, the best course of action, and the order of waste collection.	Python IDLE or Anaconda navigator or Jupitar.
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud **Cloud Server Configuration:** The process of delivering an application using one or more cloud-based hosting models,	Cloud server- MySQL

		such as software as a service (SaaS), platform as a service (PaaS), or infrastructure as a service (IaaS), is known as cloud deployment. Local Server. **Configuration:** Access to data and objects in a collection of Windows folders known as data directories is restricted unless you use a local server.	Local server-HTTP

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	□ Transport, treatment, and disposal of waste together with monitoring and regulation. □ It also encompasses the legal and regulatory framework that relates to waste management encompassing guidance on recycling.”	Python
2.	Security Implementations	□ Fundamental component of data security that dictates who's allowed to access and use company information and resources. □ Firewalls use a rule-based access control model with rules expressed in an access control list.	Firewall
3.	Scalable Architecture	Using smart waste bins, reduce the number of bins	Technology used

		inside town and cities because that we can able to monitor the garbage 24/7. It will be more cost efficient and scalable when we moves to smarter.	
4.	Availability	By developing & deploying resilient hardware and beautiful software we empower cities, businesses, and countries to manage waste smarter.	IOT, RFID
5.	Performance	□ The Smart Sensors use ultrasound technology to measure the fill levels (along with other data) in bins several times a day. □ Using a variety of IoT networks (NB- IoT, GPRS), the sensors send the data to Sensor's Smart Waste Management Software System, a powerful cloud-based platform, for data-driven daily operations, available also as a waste management app.	IOT, GPRS


Administrative servers
Receive updated data
Notify collection company
http://
Cloud database
Send recycled e-waste data
Mobile app
Send collection truck
Send container level data
Guide users to nearest collection box
Smart e-waste collection box A
Smart e-waste collection box B
Smart e-waste collection box C

5.3 USER STORIES

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a customer, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
	login	USN-2	As a customer, I can login to the application by entering correct email and password.	I can access my account/dashboard.	High	Sprint-1
	Dashboard	USN-3	As a customer, I can see all the orders raised by me.	I get all the info needed in my dashboard.	Low	Sprint-2
	Order creation	USN-4	As a customer, I can place my order with the detailed description of my query	I can ask my query	Medium	Sprint-2
	Address Column	USN-5	As a customer, I can have conversations with the assigned agent and get my queries clarified	My queries are clarified.	High	Sprint-3
	Forgot password	USN-6	As a customer, I can reset my password by this option in case I forgot my old password.	I get access to my account again	Medium	Sprint-4
	Order details	USN-7	As a Customer, I can see the current stats of order.	I get a better understanding	Medium	Sprint-4
Agent (web user)	Login	USN-1	As an agent I can login to the application byentering Correct email and password.	I can access my account / dashboard.	High	Sprint-3
	Dashboard	USN-2	As an agent, I can see the order details assigned to me by admin.	I can see the tickets to which I could answer.	High	Sprint-3
	Address column	USN-3	As an agent, I get to have conversations with the customer and clear his/er doubts	I can clarify the issues.	High	Sprint-3
	Forgot password	USN-4	As an agent I can reset my password by this option in case I forgot my old password.	I get access to my account again.	Medium	Sprint-4

Admin (Mobile user)	Login	USN-1	As a admin, I can login to the application by entering Correct email and password	I can access my account/dashboard	High	Sprint-1
	Dashboard	USN-2	As an admin I can see allthe orders raised in the entire system and lot more	I can assign agentsby seeing those order.	High	Sprint-1
	Agent creation	USN-3	As an admin I can createan agent for clarifying the customers queries	I can create agents.	High	Sprint-2
	Assignment agent	USN-4	As an admin I can assignan agent for each order created by the customer.	Enable agent to clarify the queries.	High	Sprint-1
	Forgot password	USN-5	As an admin I can resetmy password by this option in case I forgot my old password.	I get access to my account.	High	Sprint-1

PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc	03 October 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	03 October 2022
Ideation	List them by organizing the brainstorming session and prioritize the top 3 ideas based on feasibility & importance.	09 October 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	03 October 2022

Problem Solution Fit	Prepare problem - solution fit document.	31 October 2022
Solution Architecture	Prepare a solution architecture document.	03 October 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application	31 October 2022
Functional Requirement	Prepare the functional requirement document	28 October 2022
Data flow diagrams	Draw the data flow diagrams and submit for review	31 October 2022
Technology architecture	Prepare the technology architecture diagram.	29 October 2022
Prepare milestone and activity list	Prepare the milestones & activity list of the project	12 November 2022
Project Development, delivery of sprints-1,2,3 & 4	Develop & submit the developed code by testing it.	05 November 2022

6.2 DELIVERY SCHEDULE:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As an administrator, I must provide user names and passwords to each employee in the municipality.	10	High	Nhidhees Laksh Kumar, Sai Vishnu, Mugesh, Manojkumar.
Sprint-1	Login	USN-2	As a Co-Admin, I'll keep an eye on the trash level via a real-time online interface. I'll let the trash truck know where the bin is located and its ID after it has been filled.	10	High	Nhidhees Laksh Kumar, Sai Vishnu, Mugesh, Manojkumar.
Sprint-2	Dashboard	USN-3	As a Truck Driver, I'll follow Co-Admin's Instruction to reach the filling bin in short roots and save time	20	Low	Nhidhees Laksh Kumar, Sai Vishnu, Mugesh, Manojkumar.
Sprint-3	Dashboard	USN-4	I collect all the waste from the garbage as a local garbage collector, load it onto a garbage truck, and bring it to landfills.	20	Medium	SaiVishnu, Nhidhees Laksh Kumar, Mugesh, Manojkumar.
Sprint-4	Dashboard	USN-5	I'll make sure everything is going as planned and without any issues as a Municipality official.	20	High	Nhidhees Laksh Kumar, Sai Vishnu, Mugesh, Manojkumar.

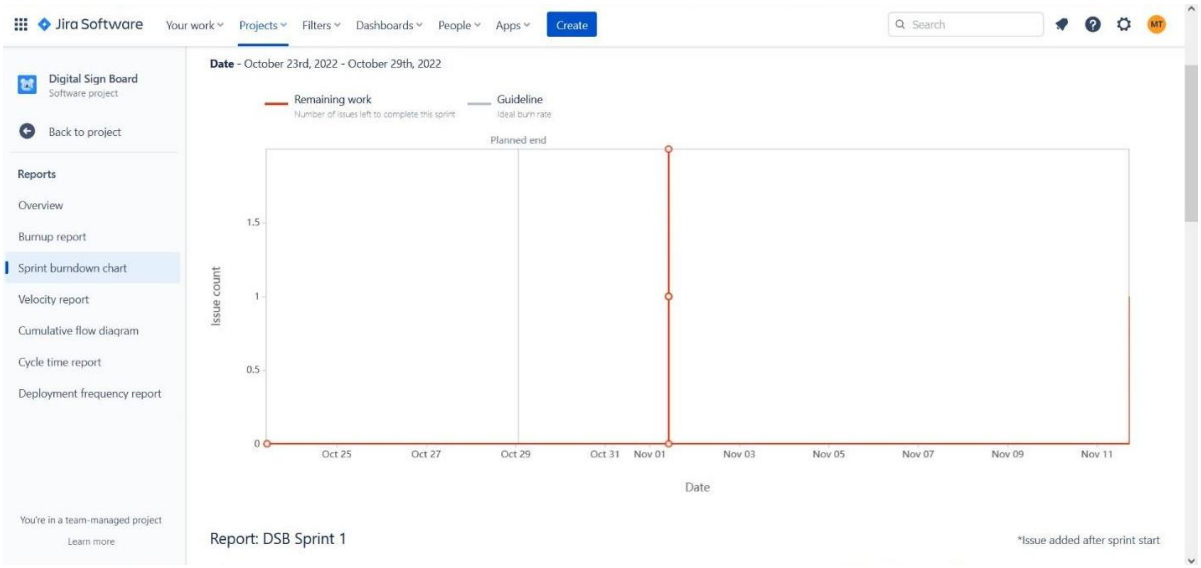
PROJECT TRACKER, VELOCITY AND BURNDOWN CHART:

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

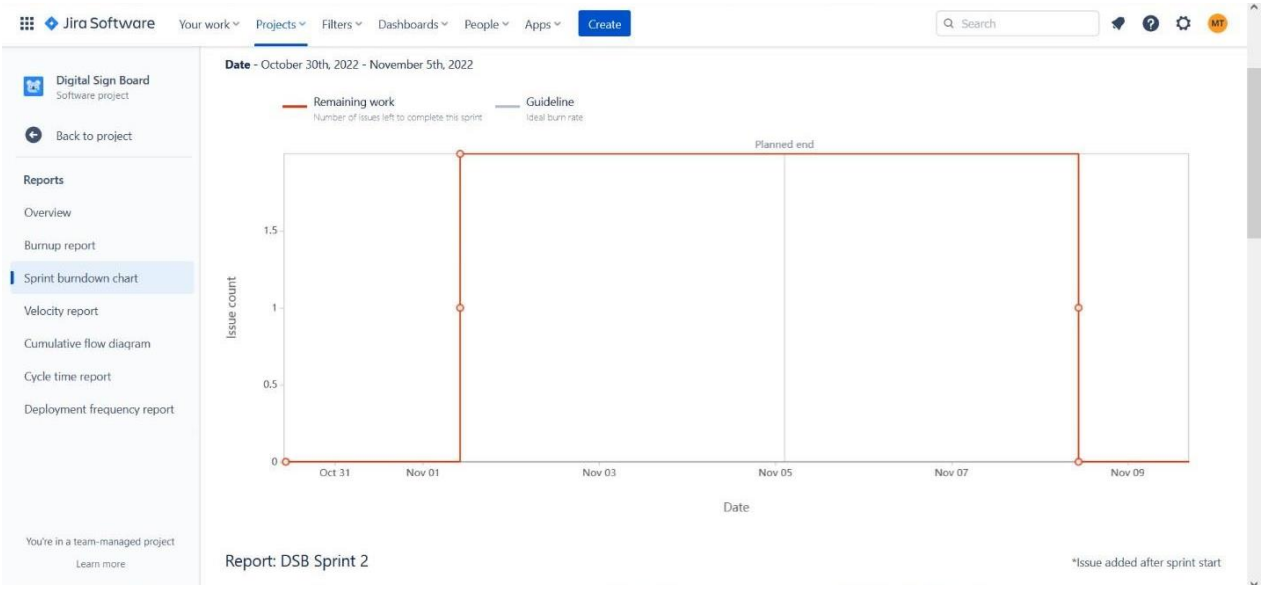
6.3 REPORTS FROM JIRA

SMART WASTE MANAGEMENT SYSTEM IN THE JIRA SOFTWARE PLATFORM:

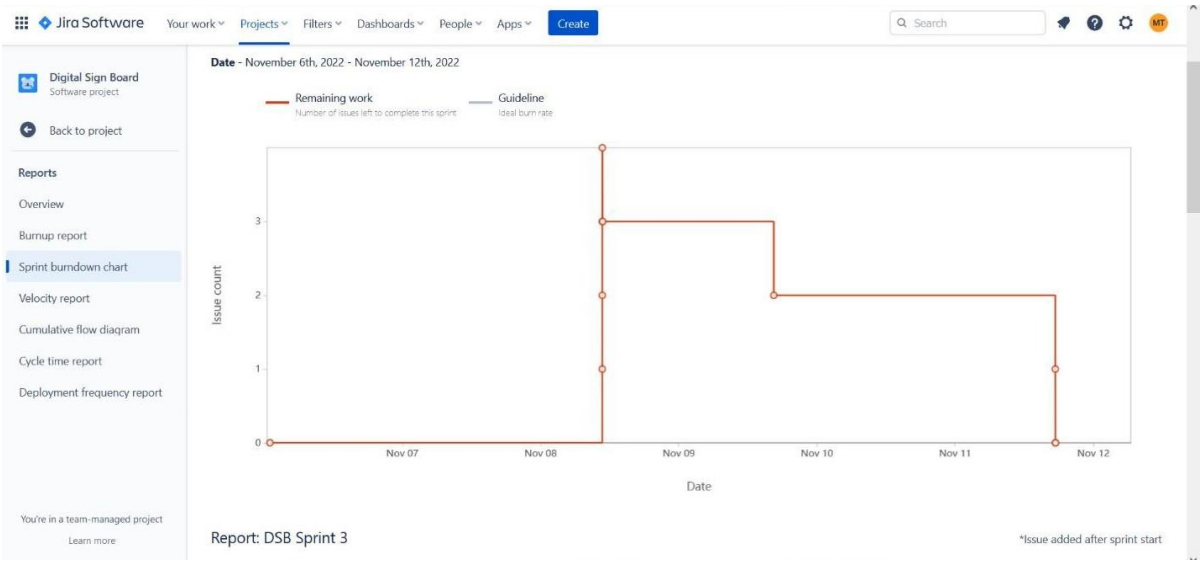
Sprint-1:



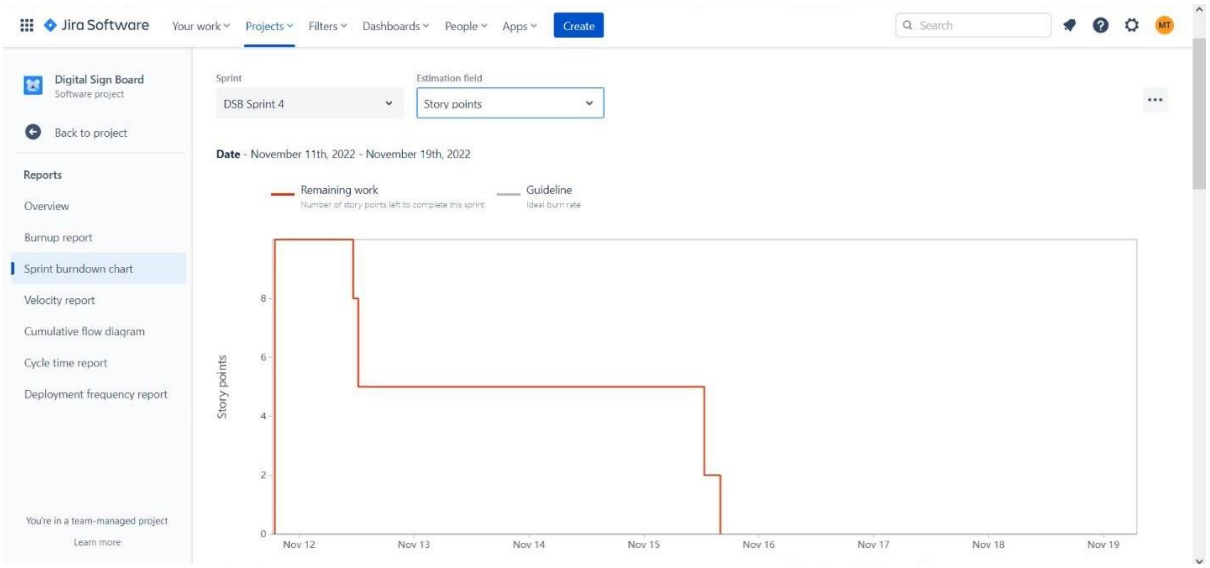
Sprint-2:



Sprint-3:



Sprint-4:



7.CODING & SOLUTIONING:

7.1 WEBSITE CODES:

WELCOME PAGE CODE (HTML):

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Smart Waste Management System For Metropolitan Cities</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
<!--===============================================================================
=======================-->
    <link rel="icon" type="image/png"
href="/static/images/icons/favicon.ico"/>
<!--===============================================================================
=======================-->
    <link rel="stylesheet" type="text/css"
href="/static/vendor/bootstrap/css/bootstrap.min.css">
<!--===============================================================================
=======================-->
    <link rel="stylesheet" type="text/css"
href="/static/fonts/font-awesome-4.7.0/css/font-awesome.min.css">
<!--===============================================================================
=======================-->
    <link rel="stylesheet" type="text/css"
href="/static/fonts/Linearicons-Free-v1.0.0/icon-font.min.css">
<!--===============================================================================
=======================-->
    <link rel="stylesheet" type="text/css"
href="/static/vendor/animate/animate.css">
<!--===============================================================================
=======================-->
    <link rel="stylesheet" type="text/css"
href="/static/vendor/css-hamburgers/hamburgers.min.css">
<!--===============================================================================
=======================-->
```

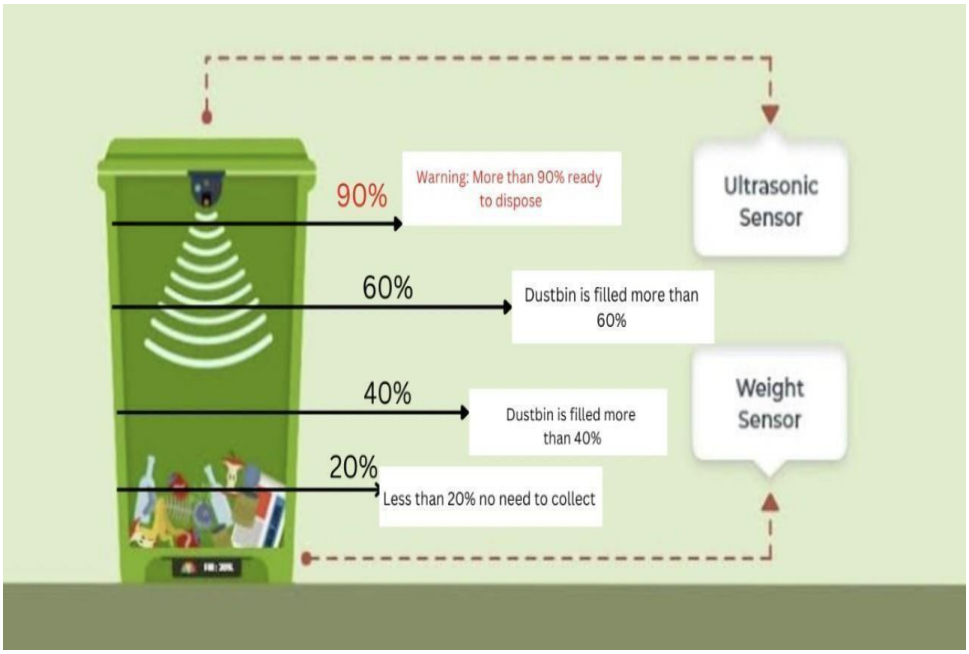
```
    <link rel="stylesheet" type="text/css"
href="/static/vendor/animsition/css/animsition.min.css">
<!--===============================================================================
=====================-->
    <link rel="stylesheet" type="text/css"
href="/static/vendor/select2/select2.min.css">
<!--===============================================================================
=====================-->
    <link rel="stylesheet" type="text/css"
href="/static/vendor/daterangepicker/daterangepicker.css">
<!--===============================================================================
=====================-->
    <link href="{{ url_for('static', path='/css/main.css') }}"
rel="stylesheet">
    <link href="{{ url_for('static', path='/css/util.css') }}"
rel="stylesheet">
<!--===============================================================================
=====================-->
</head>
<body>


    <div class="limiter">
        <div class="container-login100" style="background-image: url({{
url_for('static', path='/images/bg1.jpg') }});">
            <div class="wrap-login100 p-l-110 p-r-110 p-t-62 p-b-33">
                <form class="login100-form validate-form flex-sb flex-w">
                    <span class="login100-form-title p-b-53">
                        Sign In With
                    </span>

                    <a href="#" class="btn-face m-b-20">
                        <i class="fa fa-facebook-official"></i>
                        Facebook
                    </a>

                    <a href="#" class="btn-google m-b-20">
```

```
                        <img src="static/icons/icon-google.png" alt="GOOGLE">
                        Google
                    </a>

                    <div class="p-t-31 p-b-9">
                        <span class="txt1">
                            Username
                        </span>
                    </div>
                    <div class="wrap-input100 validate-input" data-validate =
"Username is required">
                        <input class="input100" type="text" name="username" >
                        <span class="focus-input100"></span>
                    </div>

                    <div class="p-t-13 p-b-9">
                        <span class="txt1">
                            Password
                        </span>

                        <a href="#" class="txt2 bo1 m-l-5">
                            Forgot?
                        </a>
                    </div>
                    <div class="wrap-input100 validate-input" data-validate =
"Password is required">
                        <input class="input100" type="password" name="pass" >
                        <span class="focus-input100"></span>
                    </div>

                    <div class="container-login100-form-btn m-t-17">
                        <button class="login100-form-btn">
                            Login
                        </button>
```

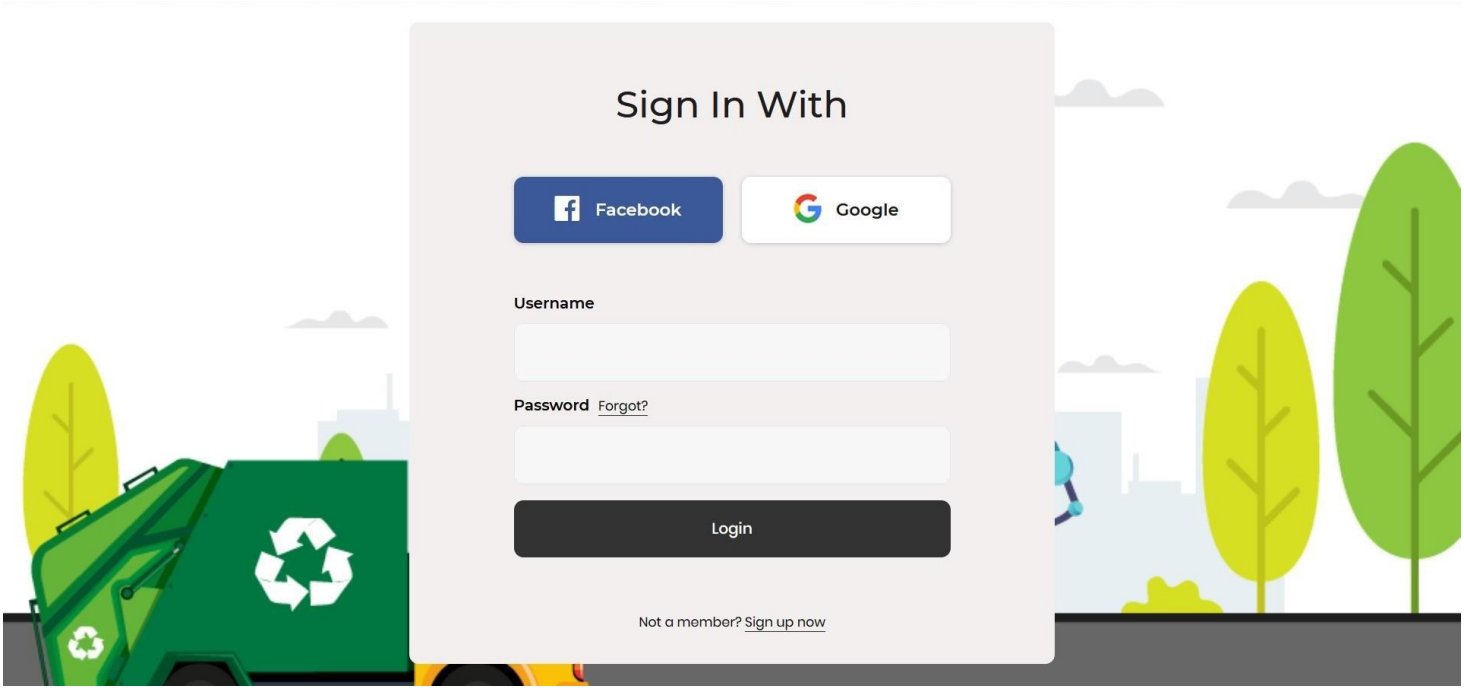
```
                    </div>

                    <div class="w-full text-center p-t-55">
                        <span class="txt2">
                            Not a member?
                        </span>

                        <a href="#" class="txt2 bo1">
                            Sign up now
                        </a>
                    </div>
                </form>
            </div>
        </div>
    </div>


    <div id="dropDownSelect1"></div>

<!--===============================================================================================-->
    <script src="/static/vendor/jquery/jquery-3.2.1.min.js"></script>
<!--===============================================================================================-->
    <script src="/static/vendor/animsition/js/animsition.min.js"></script>
<!--===============================================================================================-->
    <script src="/static/vendor/bootstrap/js/popper.js"></script>
    <script src="/static/vendor/bootstrap/js/bootstrap.min.js"></script>
<!--===============================================================================================-->
    <script src="/static/vendor/select2/select2.min.js"></script>
<!--===============================================================================================-->
    <script src="/static/vendor/daterangepicker/moment.min.js"></script>
```

```
    <script src="/static/vendor/daterangepicker/daterangepicker.js"></script>
<!--============================================================================================
========================-->
    <script src="/static/vendor/countdowntime/countdowntime.js"></script>
<!--============================================================================================
========================-->
    <script src="/static/js/main.js"></script>


</body>
</html>
```

WORKING MODEL:



APPLICATION OUTLOOK:

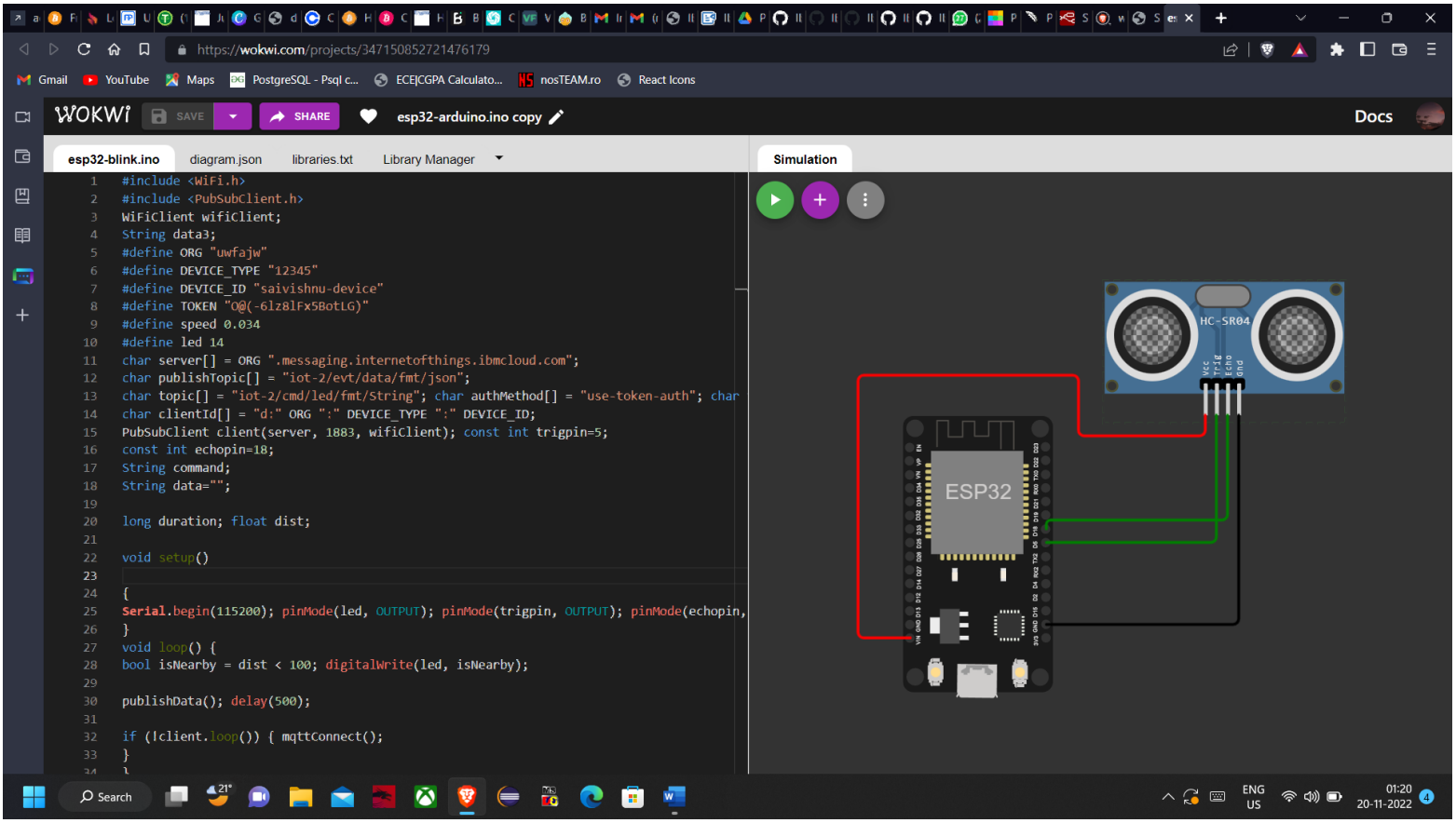


PYTHON CODE FOR HARDWARE:

```
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "uwfajw"
#define DEVICE_TYPE "12345"
#define DEVICE_ID "saivishnu-device"
#define TOKEN "O@(-6lz8lFx5BotLG)"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; char authMethod[] = "use-token-auth"; char token[] =
TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient); const int trigpin=5;
const int echopin=18;
String command;
String data="";

long duration; float dist;

void setup()

{
Serial.begin(115200); pinMode(led, OUTPUT); pinMode(trigpin, OUTPUT); pinMode(echopin, INPUT);
wifiConnect(); mqttConnect();
}
void loop() {
bool isNearby = dist < 100; digitalWrite(led, isNearby);

publishData(); delay(500);

if (!client.loop()) { mqttConnect();
}
}

void wifiConnect() {
Serial.print("Connecting to "); Serial.print("Wifi"); WiFi.begin("Wokwi-GUEST", "", 6);
while (WiFi.status() != WL_CONNECTED) { delay(500);
Serial.print(".");
}
Serial.print("WiFi connected, IP address: "); Serial.println(WiFi.localIP());
}

void mqttConnect() {
if (!client.connected()) {
```

```
Serial.print("Reconnecting MQTT client to "); Serial.println(server); while
(!client.connect(clientId, authMethod, token)) { Serial.print(".");
delay(500);
}
initManagedDevice();
Serial.println();
}
}

void initManagedDevice() { if (client.subscribe(topic)) {
// Serial.println(client.subscribe(topic));
Serial.println("IBM subscribe to cmd OK");
} else {
Serial.println("subscribe to cmd FAILED");
}
}
void publishData()
{
digitalWrite(trigpin,LOW); digitalWrite(trigpin,HIGH); delayMicroseconds(10);
digitalWrite(trigpin,LOW); duration=pulseIn(echopin,HIGH); dist=duration*speed/2; if(dist<100){
String payload = "{\"Alert Distance\":"; payload += dist;
payload += "}"; Serial.print("\n"); Serial.print("Sending payload: "); Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}

}
if(dist>100){
String payload = "{\"Distance\":"; payload += dist;
payload += "}";

Serial.print("\n"); Serial.print("Sending payload: "); Serial.println(payload);
if(client.publish(publishTopic, (char*) payload.c_str())) {
Serial.println("Publish OK");
}else {
Serial.println("Publish FAILED");
}

}

}
```

SIMULATION:


https://wokwi.com/projects/347150852721476179
WOKWI
SAVE
SHARE
esp32-arduino.ino copy
Docs
esp32-blink.ino
diagram.json
libraries.txt
Library Manager
Simulation
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
String data3;
#define ORG "uwfajw"
#define DEVICE_TYPE "12345"
#define DEVICE_ID "saivishnu-device"
#define TOKEN "O@(-6lz8lFx5BotLG)"
#define speed 0.034
#define led 14
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; char authMethod[] = "use-token-auth"; char
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient); const int trigpin=5;
const int echopin=18;
String command;
String data="";
long duration; float dist;
void setup()
{
Serial.begin(115200); pinMode(led, OUTPUT); pinMode(trigpin, OUTPUT); pinMode(echopin,
}
void loop() {
bool isNearby = dist < 100; digitalWrite(led, isNearby);
publishData(); delay(500);
if (!client.loop()) { mqttConnect();
}
HC-SR04
ESP32
Search
ENG
US
01:20
20-11-2022

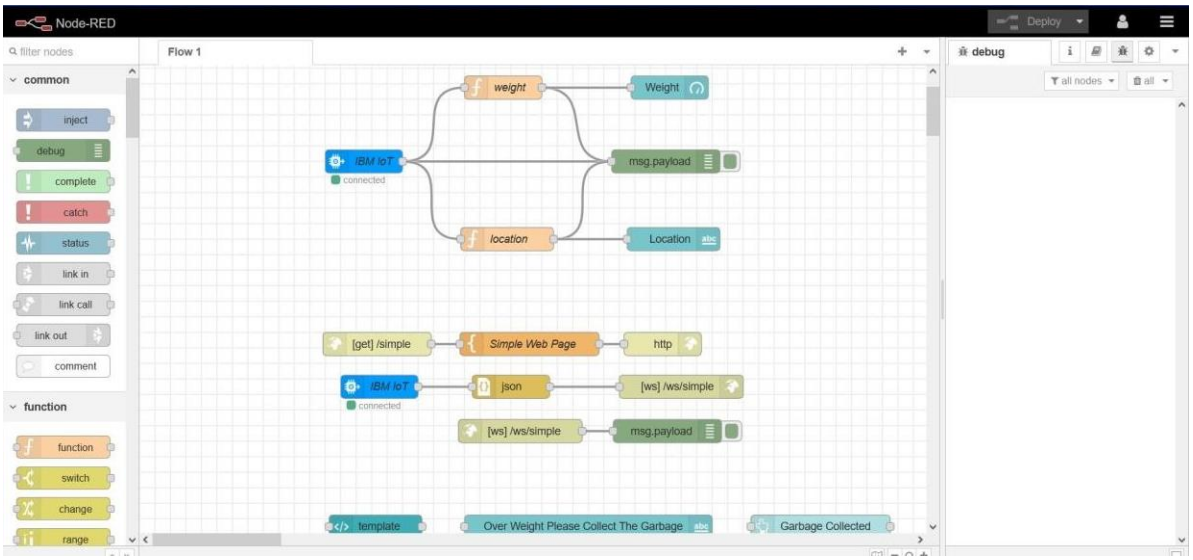
SIMULATION CODE:

```
   #include <cstdlib>
#include <time.h>
#include <WiFi.h>
#include <PubSubClient.h>

#define ORG "uwfajw"
#define DEVICE_TYPE "Rasp"
#define DEVICE_ID "12345"
#define TOKEN "12345678"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);
int weight = 0;

String location = "Coimbatore";
String status = "";



void setup() {
  Serial.begin(99900);


   wifiConnect();
   mqttConnect();
}

void loop() {

  srand(time(0));

    //initial variable
    int p;

    weight = random(0,80);
    if(weight > 0  && weight < 25){
        p = 0;
    }
    else if(weight > 25  && weight < 50){
        p = 1;
```

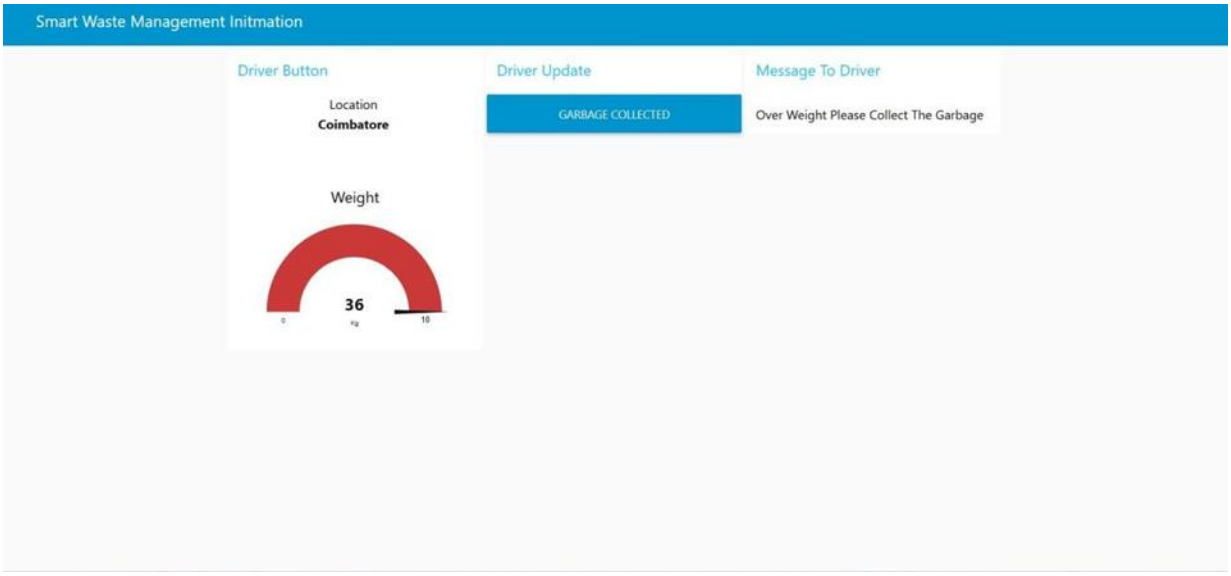
```
    }
    else{
      p = 2;
    }

    //set a quality status

    switch (p) {
    case 0:
        status = "Low";
        break;
    case 1:
        status = "Half";
        break;
    case 2:
        status = "Full";
        break;
    }

  //Obivously the output.It is like json format 'cause it will help us for future sprints
    String payload = "{";
    payload+="\"Weight \":";
    payload+=weight;
    payload+=",";
    payload+="\"Loaction\":";
    payload+="Coimbatore";
    payload+=",";
    payload+="\"Status\":\""+status+"\"}";
    Serial.println(payload);


  if(client.publish(publishTopic, (char*) payload.c_str()))
  {
    Serial.println("Publish OK");
  }
  else{
    Serial.println("Publish failed");
  }
  delay(1000);


  if (!client.loop())
  {
    mqttConnect();
  }



}
```

```
void wifiConnect()
{
  Serial.print("Connecting to ");
  Serial.print("Wifi");
  WiFi.begin("Wokwi-GUEST", "", 6);
  while (WiFi.status() != WL_CONNECTED)
  {
    delay(500);
    Serial.print(".");
  }
  Serial.print("WiFi connected, IP address: ");
  Serial.println(WiFi.localIP());

}


void mqttConnect()
{
  if (!client.connected())
  {
    Serial.print("Reconnecting MQTT client to ");
    Serial.println(server);
    while (!client.connect(clientId, authMethod, token))
    {
      Serial.print(".");
      delay(500);
    }

    Serial.println();
  }
```

Node-RED:



OUTLOOK:


Smart Waste Management Initmation
Driver Button
Location
Coimbatore
Weight
36
0
10
Driver Update
GARBAGE COLLECTED
Message To Driver
Over Weight Please Collect The Garbage

TESTING

8.1 TEST CASE:

Test case	Feature Type- Bin Level	Component	Test Case Scenario	Pre-Requisite	Availability	Test Condition	Expected Result	Actual Result	Status	Comments	Accessed By
Test case 1	Empty	Ultrasonic Sensor	When Bin is empty	Ultrasoncic sensor PIR Motion Sensor Garbage Bins	Bin is accessible to users	Bin Level == 0	Displays Bin level and space left	Working as expected	Pass		MUGESH M
Test case 2	Accessible	Ultrasonic Sensor	When bin level is below 50 %	Ultrasoncic sensor , PIR Motion Sensor , Garbage Bins	Bin is accessible to users	Bin Level < 50	Displays Bin level and space left	Working as expected	Pass		SAI VISHNU L
Test case 3	Accessible	Ultrasonic Sensor	When bin level is above 50	Ultrasoncic sensor , PIR Motion sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level > 50	Displays Bin level and space left	Working as expected	Pass		MANOJKUMAR R
Test case 4	Accessible	Ultrasonic Sensor	When bin level is below 75 %	Ultrasoncic sensor , PIR Motion sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level < 75	Displays Bin level and space left	Working as expected	Pass		NHIDHEES LAKSH KUMAR K.B
Test case 5	Limit exceeded	Ultrasonic Sensor	When bin level is above 75 %	Ultrasoncic sensor , PIR Motion sensor , Garbage Bins	Bin is not accessible to the users, the admin recieves High alert and seals the the bin to avoid overflow.	Bin Level > 75	Displays Bin is FULL and Seals the bin.	Working as expected	Pass	The system starts to sense the level once the Bin is emptied partially or fully	MANOJKUMAR R

TEST CASE 1:
WEIGHT: 0 KG
STATUS: NOT FILLED, DUSTBIN IS EMPTY

TEST CASE 2:
WEIGHT: 10KG
STATUS:20% FILLED, NOT READY TO DISPOSE

TEST CASE 3:
WEIGHT: 20KG
STATUS: 40% FILLED, NOT READY TO DISPOSE

TEST CASE 4:
WEIGHT: 30KG
STATUS: 60% FILLED, NOT READY TO DISPOSE

TEST CASE 5:
WEIGHT: 45KG
STATUS: 90% FILLED, READY TO DISPOSE

8.2 USER ACCEPTING TECHNIQUE:

Defect Analysis;

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	1	4
Client Application	47	0	2	46
Security	3	0	0	3

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	2	19
Duplicate	1	1	2	0	4
External	2	4	0	1	6
Fixed	10	2	3	20	37
Not Reproduced	0	0	2	0	2
Skipped	0	0	2	1	3
Won't Fix	0	5	2	1	8
Totals	23	16	13	25	79

Test Case Analysis;

This report shows the number of test cases that have passed, failed, and untested.

Outsource Shipping	2	0	0	2
Exception Reporting	11	0	2	9
Final Report Output	5	0	0	5
Version Control	3	0	1	2

RESULTS

9.1 PERFORMANCE METRICS

Total MSW Generated = Total tons Recycled + Total tons Recovered + Total tons Disposed
MSW = Municipal Solid Waste (does not include industrial, special and demolition wastes)

ADVANTAGES:

- . 80% reduction in the amount of rubbish pickups required.
- A reduction in the utilisation of labour, pollutants, fuel, and transportation congestion.
- A decrease in the quantity of trash cans required.
- Analytics data to better efficiently manage collection routes and bin placement

DISADVANTAGES:

- These tasks range from sorting trash collectors to the demanding work that is required in
- manufacturing and retail establishments.
- Even while waste management generates employment, it can only do so with low-quality
- employment.

CONCLUSION:

It is feasible to create a system that is more effective than the one that is already in place by monitoring the fullness of bins with the use of sensors. Our conception of a "smart waste management system" focuses on monitoring waste management, offering intelligent technology for waste systems, eliminating human involvement, minimizing human time and effort, and producing a healthy and trash free environment. The suggested approach can be put into practice in smart cities where inhabitants have busy schedules that provide little time for garbage management. If desired, the bins might be put into place in a metropolis where a sizable bin would be able to hold enough solid trash for a single unit. The price could be divided among the locals, resulting in less expensive service delivery.

GitHub Project Link:

https://github.com/IBM-EPBL/IBM-Project-41736-1660644494