Nalaiya Thiran

Smart Waste Management System For Metropolitan Cities

PROJECT REPORT

College Name: PAAVAI ENGINEERING COLLEGE

Team ID: PNT2022TMID13797

Team Members:

1. RAM KUMAR.M(TEAM LEADER)
2. NAVEEN. S
3. UDAYAN.R
4. PRAVEEN KUMAR.R

Project Guide:

Industry mentor: Mr. Dinesh

Faculty Mentor: Mr. RAMALINGAM N.M.K

Project Report Format

1. **INTRODUCTION**
    - 1.1. Project Overview
    - 1.2. Purpose
2. **LITERATURE SURVEY**
    - 2.1. Existing problem
    - 2.2. References
    - 2.3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    - 3.1. Empathy Map Canvas
    - 3.2. Ideation & Brainstorming
    - 3.3. Proposed Solution
    - 3.4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    - 4.1. Functional requirement
    - 4.2. Non-Functional requirements
5. **PROJECT DESIGN**
    - 5.1. Data Flow Diagrams
    - 5.2. Solution & Technical Architecture
    - 5.3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
    - 6.1. Sprint Planning & Estimation
    - 6.2. Sprint Delivery Schedule
    - 6.3. Reports from JIRA
7. **CODING & SOLUTIONING**
    - 7.1. Feature 1
    - 7.2. Feature 2

1. INTRODUCTION

1.1 Project Overview

Garbage collection is becoming a major challenge in megacities due to their higher population density and increasing urbanization. Waste management is a concern, equitable participation of the government and public is required to manage and solve waste management difficulties. A simple technique can be adopted with the aid of IOT and the concept of Smart cities in order to improve the present system and take a step ahead in managing waste management challenges utilizing IOT. We propose a smart garbage truck collecting/tracking system that is powered by an ESP8266 and sends real-time truck data to the government via an MIT app. Additionally, the garbage level within the bin is monitored via a ULTRASONIC SENSOR, which sends real-time data to the app, which is accessible to the government and the public of that specific area/locality. Using this system, the government may track whether or not the trash truck is collecting rubbish according to its route/schedule, and if a bin is full midway or empty. Our method solves two problems: one is cost efficiency, and the other is ensuring that waste is collected from all bins in that area/region. One of the most interesting aspects of our technology is that we can follow the truck's location.

1.2 Purpose

Many times, in our city, we encounter garbage cans or dustbins put in public locations that are overflowing. It causes unsanitary circumstances for people, adds unpleasantness to the area, and serves as a breeding ground for mosquitoes. To avoid all of these scenarios, we will construct a project named Smart Waste Management System For Metropolitan Cities. This method also aids in the monitoring of false reports, which can assist to prevent corruption in the entire management system. Both the municipality and the public may monitor the dust bins. This approach minimizes the total number of trips made by garbage collection vehicles, lowering the overall cost of waste collection. It is ultimately beneficial in maintaining it ultimately contributes to the preservation of cleanliness in society. As a result, the smart garbage management system improves garbage collection efficiency. These dust bin models may be used in any smart city across the world. A waste collecting and monitoring crew that is assigned to gather rubbish from the city can be adequately led for collection.

2. LITERATURE SURVEY

2.1 Existing problem

The increasing volume and complexity of waste associated with the modern economy is posing a serious risk to ecosystems and human health. Poor waste management causes air pollution, water pollution, and soil contamination, ranging from non-existent collection systems to poor disposal. Open and unclean landfills pollute drinking water and can promote infection and disease transmission. The spread of trash pollutes ecosystems and harmful compounds from electronic waste or industrial junk endangers urban people's and the environment's health.

2.2 References

S.NO	PROJECT NAME	AUTHOR	OBJECTIVE/OUTCOME
1	IoT Based Smart Garbage System.	1) T.Sinha 2) R.M Sahuother	IoT Based Smart Garbage System which indicates directly that the dustbin is filled to a certain level by the garbage and cleaning or emptying them is a matter of immediate concern. This prevents lumping of garbage in the roadside dustbin which ends up giving foul smell and illness to people. The design of the smart dustbin includes a single by ultrasonic sensor which configured with Arduino Uno with this research ,it is sending SMS to the Municipal Council that particular dustbin is to overflow.

2	Raspberry pi-based smart waste management system using Internet of Things.	1)Shaik Vaseem Akram 2)Rajesh Singh	Nowadays it is becoming a difficult task to distinguish wet and dry waste. The new waste management system covers several levels of enormous workforce. Every time labourerS must visit the garbage bins in the city area to check whether they are filled or not. The data communicates to the cloud server for real-time monitoring of the system. With the real-time fill level information collected via the monitoring platform, the system reduces garbage overflow by informing about such instances before they arrive.
3	Smart Waste Management System.	1) Sanjiban Charkraborty	This Waste management is one of the serious challenges of the cities,the system now used in cities, we continue to use an old and outmoded paradigm that no longer serves the entail of municipalities, Still find over spilled waste containers giving off irritating smells causing serious health issues and atmosphere impairment
4	Smart Solid Waste Management.	1) Mohd Helmy Abd Wahab.	At the time of trash diposal,the material to be recycled could be identified using RFID technology.

5	Analysis of Load cell	1) Ranjeet Kumar 2) Sandeep Chhabra	Load Cells 4.1 General Load Cell related information A load cell is meant to measure the size of a mass but actually is a force sensor which transforms force into an electrical signal. The load cell needs the earth gravity to work. Every mass is attracted by the earth gravimetric field, that force is named “load”.

2.3 Problem Statement Definition

The objective of the project is to utilize sensors, intelligent monitoring systems, and mobile applications to address the challenges of waste management in metropolitan cities. Sensors are the first smart waste management technology to improvethe effectiveness of the garbage collecting process. These devices help optimize the best possible route containing fully filled containers and create smart schedules for drivers.

3. IDEATION & PROPOSED SOLUTION

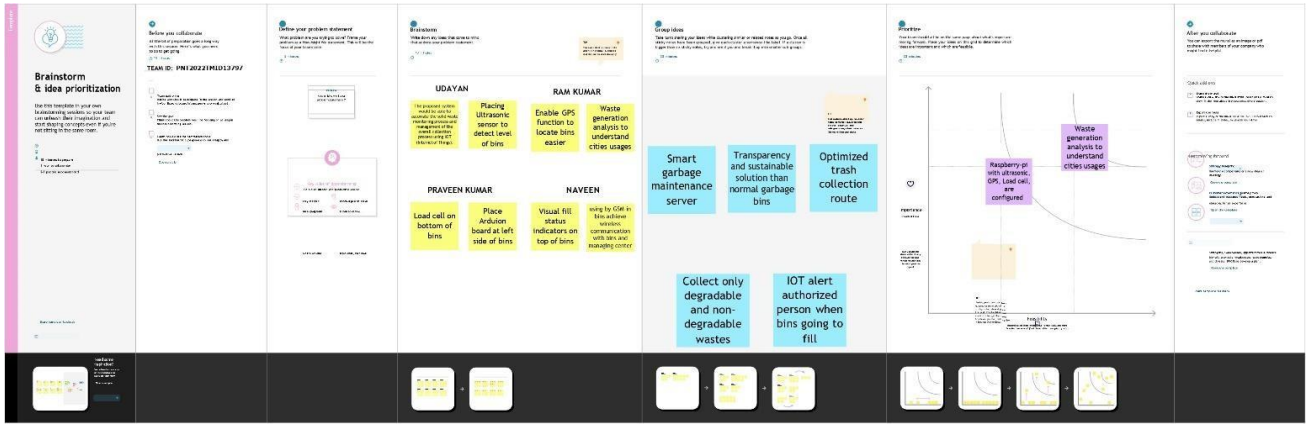
3.1 Empathy Map Canvas

EMPATHY MAP

Smart Waste Management System For Metro Cities using IOT

TEAM ID : PNT2022TMID13797

3.2 Ideation & Brainstorming

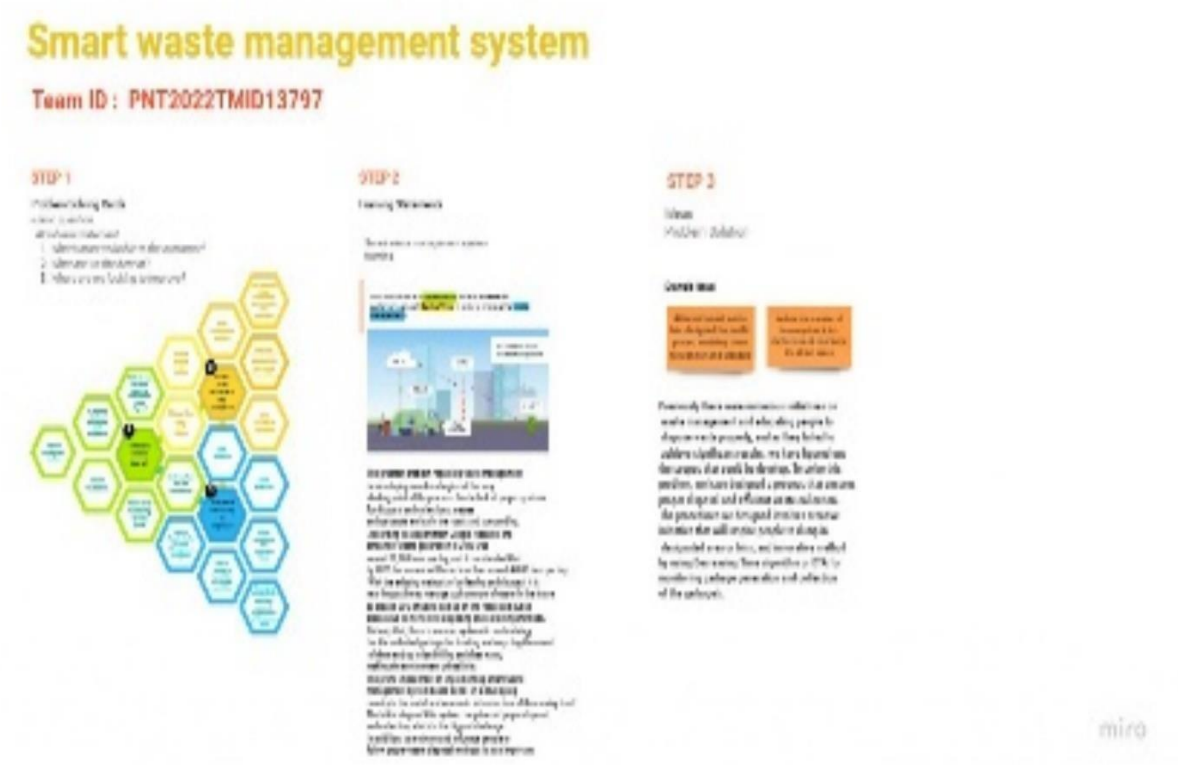


3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	This project deals with the problem of waste management in smart cities, where the garbage collection system is not optimized. This project enables the organizations to meet their needs of smart garbage management systems. This system allows the authorized person to know the fill level of each garbage bin in a locality or city at all times, to give a cost-effective and time-saving route to the truck drivers
2.	Idea / Solution description	The key research objectives are as follows: • The proposed system would be able to automate the solid waste monitoring process and management of the overall collection process using IOT (Internet of Things). • The Proposed system consists of main subsystems namely Smart Trash System (STS)and Smart Monitoring and Controlling Hut (SMCH). • In the proposed system, whenever the waste bin gets filled this is acknowledged by placing the circuit at the waste bin, which transmits it to the receiver at the desired place in the area or spot. • In the proposed system, the received signal indicates the waste bin status at the monitoring and controlling system.
3.	Novelty / Uniqueness	We are going to establish SWM in our college but the real hard thing is that janitor (cleaner) doesn't know to operate these things practically sohere our team planned to build a wrist band to them, that indicate via light blinking when the dustbin fill and this is Uniqueness we made here beside from project constrain

4.	Social Impact / Customer Satisfaction	From the public perception as worst impacts of present solid waste disposal practices are seen direct social impacts such as neighborhood of landfills to communities, breeding of pests and loss in property values
5.	BusinessModel (Revenue Model)	Waste Management organizes its operations into two reportable business segments: Solid Waste, comprising the Company's waste collection, transfer, recycling and resource recovery, and disposal services, which are operated and managed locally by the Company's various subsidiaries, which focus on distinct geographic areas; and Corporate and Other, comprising the Company's other activities, including its development and operation of landfill gas-toenergy facilities in the INDIA, and its recycling brokerage services, as well as various corporate functions.
6.	Scalability of the Solution	In this regard, smart city design has been increasingly studied and discussed around the world to solve this problem. Following this approach, this paper presented an efficient IoTbased and real-time waste management model for improving the living environment in cities, focused on a citizen perspective. The proposed system uses sensor and communication technologies where waste data is collected from the smart bin, in real-time, and then transmitted to an online platform were citizens can access and check the availability of the compartments scattered around a city.

3. 4 Problem Solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional requirement

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	Detailed bin inventory	All monitored bins and stands can be seen on the map, and you can visit them at any time via the Street View feature from Google. Bins or stands are visible on the map as green, orange or red circles. You can see bin details in the Dashboard – capacity, waste type, last measurement, GPS location and collection schedule or pick recognition.
FR-2	Real time bin monitoring	The Dashboard displays real-time data on fill-levels of bins monitored by smart sensors. In addition to the % of fill-level, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software. Sensors recognize picks as well; so, you can check whenthe bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty ones.
FR-3	Expensive bins	We help you identify bins that drive up your collection costs. The tool calculates a rating for each bin in terms of collection costs. The tool considers the average distance depo-bindischarge in the area. The tool assigns bin a rating (1-10) and calculates distance from depo-bin discharge.
FR-4	Adjust bin distribution.	Ensure the most optimal distribution of bins. Identify areas with either dense or sparse bin distribution. Make sure all trash types are represented within a stand. Based on the historical data, you can adjust bin capacity or location where necessary.

FR-5	Eliminate inefficient picks.	Eliminate the collection of half-empty bins. The sensors recognize picks. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you collect are.. The report shows how full the bin was when picked. You immediately see any inefficient picks below 80% full.

FR-6	Plan waste collection routes.	The tool semi-automates waste collection route planning. Based on current bin fill-levels and predictions of reaching full capacity, you are ready to respond and schedule waste collection. You can compare planned vs. executed routes to identify any inconsistencies

4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	IoT device verifies that usability is a special and important perspective to analyze user requirements, which can further improve the design quality. In the design process with user experience as the core, the analysis of users' product usability can indeed help designers better understand users' potential needs in waste management, behavior and experience
NFR-2	**Security**	Use reusable bottles Use reusable grocery bags Purchase wisely and recycle Avoid single use food and drink containers
NFR-3	**Reliability**	Smart waste management is also about creating better working conditions for waste collectors and drivers. Instead of driving the same collection routes and servicing empty bins, waste collectors will spend their time more efficiently, taking care of bins that need servicing
NFR-4	**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 Sensoneo's Smart Waste Management Software System, a powerful cloud-based platform, for datadriven daily operations, available also as a waste management app. Customers are hence provided data-driven decision making, and optimization of waste collection routes, frequencies, and vehicle loads resulting in

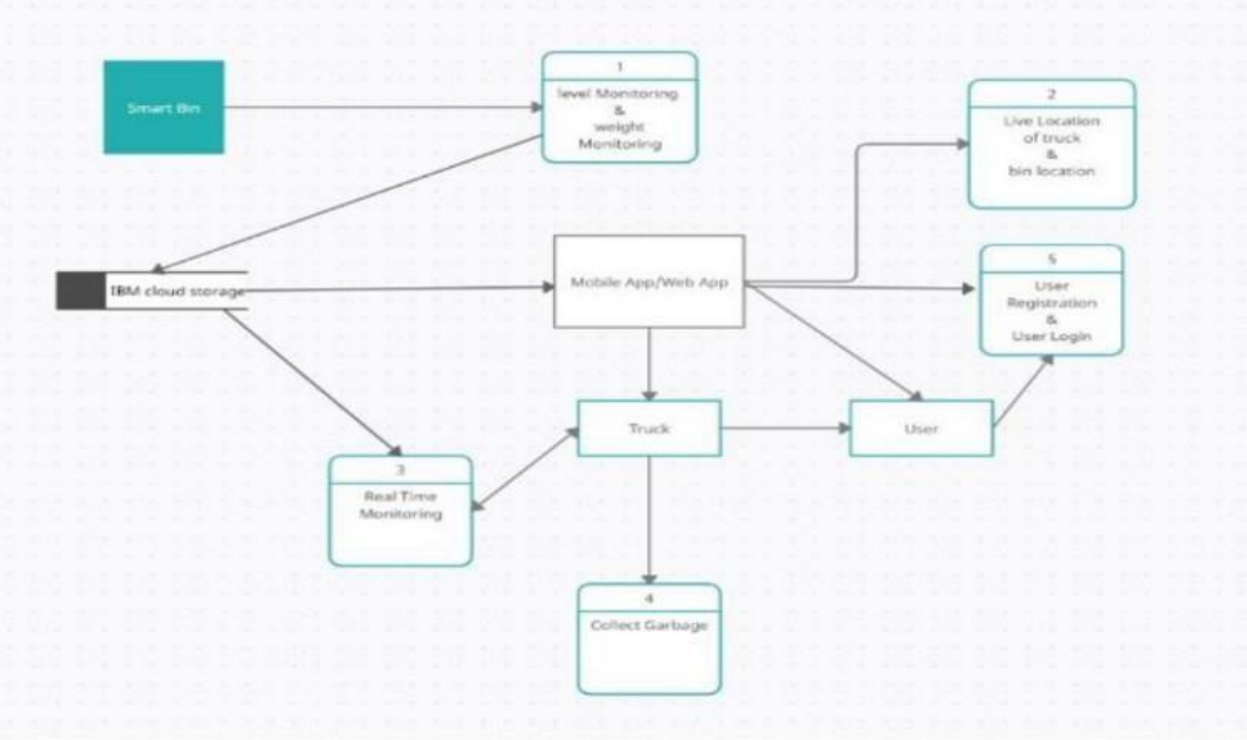
		route reduction by at least 30%.
NFR-5	**Availability**	By developing & deploying resilient hardware and beautiful software we empower cities, businesses, and countries to manage waste smarter.

NFR-6	**Scalability**	Using smart waste bins reduce the number of bins inside town, cities coz we able to monitor the garbage 24/7 more cost effect and scalability when we move to smarter

5. PROJECT DESIGN

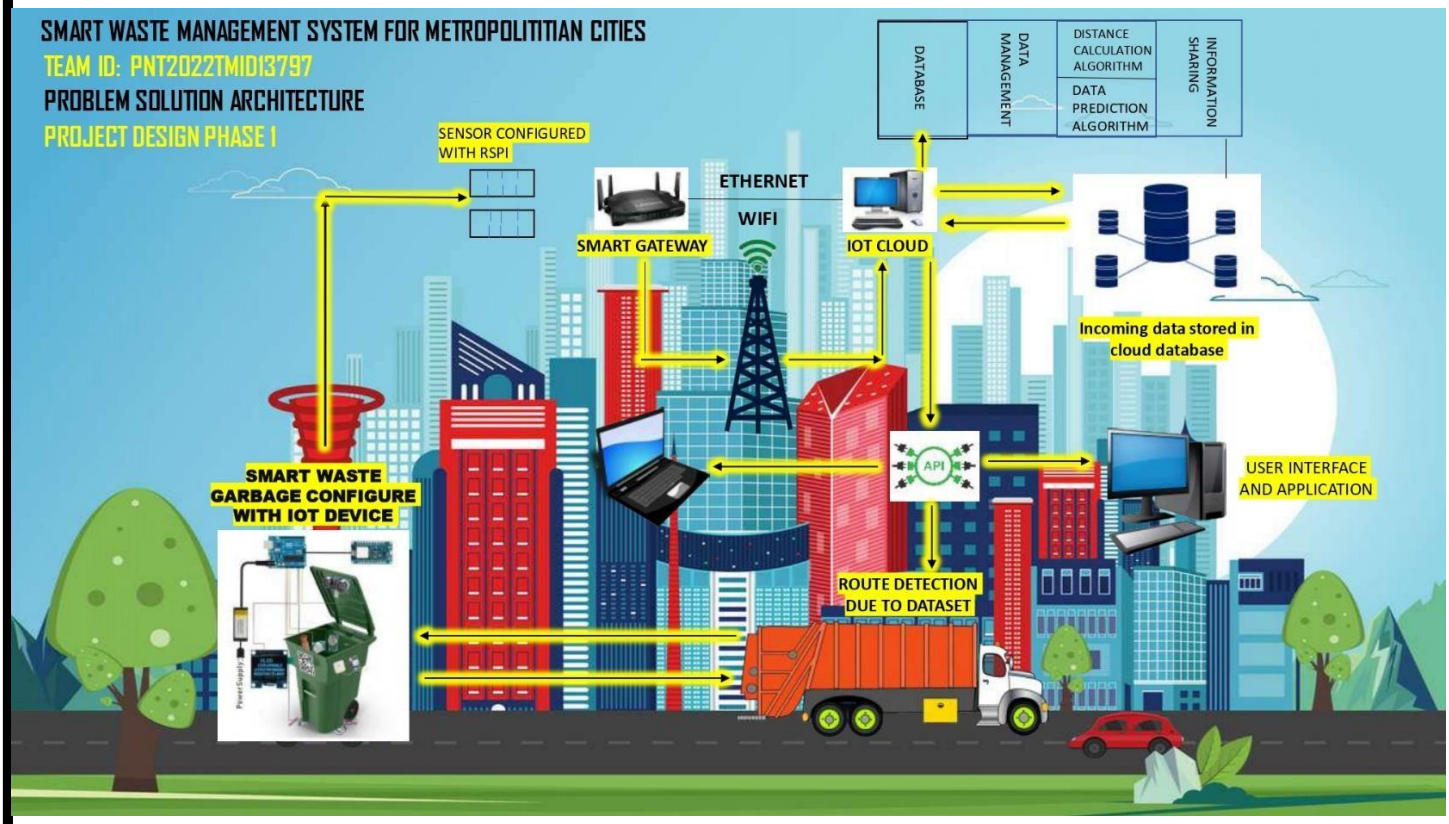
5.1 Data Flow Diagrams

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the rightamount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution & Technical Architecture

Solution Architecture:



Technical Architecture:

***NOTE:**

Here we have used ESP32 microcontroller instead of Rasbperry Pi microprocessor, cause the ESP32 has a slightly more powerful processor, with a clock speed of up to 240 MHz, compared to the Raspberry Pi Pico's maximum clock speed of 133 MHz. The Pi Pico has 2 MBof Flash memory, while the ESP 32 has 4 MB.

5.3 User Stories

User Type	Functional Requirement (Epic)	User StoryNumber	User Story / Task	Acceptance criteria	Priority	Release
Primary admin	login	USN-1	As a primary admin, I can manage the webserver	I can access account / dashboard	Medium	Sprint-2
Secondary admin	login	USN-2	As a secondary admin, I can Monitor the trucks	I can access the location	High	Sprint-1
Truck driver	dashboard	USN-3	As a truck driver, I reach to the correct destination to collect the garbage.	I can register and access thelocation via dashboard	Medium	Sprint-2
Local garbage collector	dashboard	USN-4	As a local garbage collector, I collect the garbage from the bins	I can register and access thelocation	Medium	Sprint-2
Municipality	Login	USN-5	As a municipality, we will look over the smooth management of the whole process	I can monitor the whole system	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User StoryNumber	User Story/ Task	Story Points	Priority	TeamMembers
Sprint-1	Circuit Design	USN-1	To design a circuit in Tinkercad or Wokwiusing Arduino/ESP32 board and sensors.	5	High	UDAYAN.R RAM KUMAR.M PRAVEEN KUMAR .R NAVEEN.S .
Sprint-1	Code	USN-2	To develop a python code to show the location (latitude and longitude) along with the bin values.	7	High	RAM KUMAR NAVEEN
Sprint-1	Code	Child issueN-3	To develop a python code to Show and mark the bin location in Map	3	Low	RAM KUMAR
Sprint-1	Code	Child issueN-4	To develop the python code to print the bin location(latitude and longitude)	3	High	RAM KUMAR
Sprint-1	Code	Child issueN-5	To develop a code to generate a random bin status values.	2	Medium	NAVEEN S
Sprint-2	Device Creation	USN-6	To create a device in the IoT Watson Cloudplatform and cloudant database.	2	High	NAVEEN S

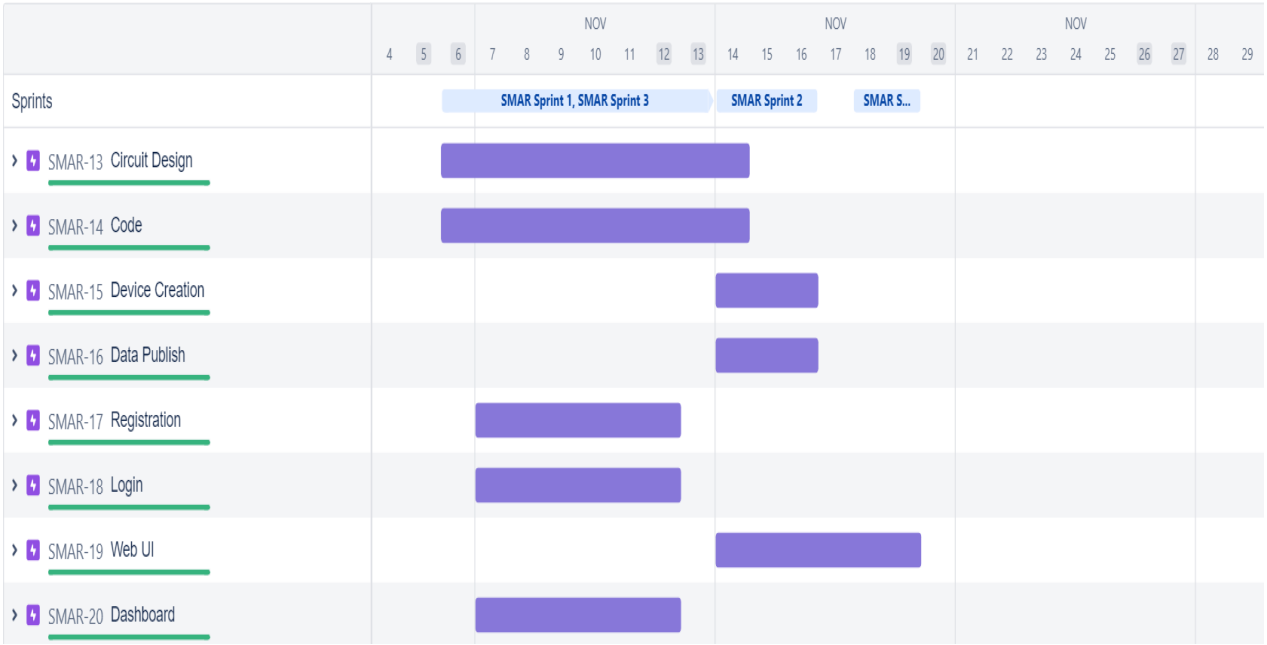
Sprint-2	Data Publish	USN-7	Publish the data such as location (Latitude and longitude) and bin status values in IoT Watson Cloud platform.	2	High	NAVEEN S
Sprint-2	Data Publish	USN-8	Store the user data such as user name, password and Gmail in the Cloudant database.	3	Medium	NAVEEN S
Sprint-2	Web UI	USN-9	To create the process workflow for IoT situations using local Node-red Application.	5	High	PRAVEEN KUMAR
Sprint-3	Registration	USN-10	As a user, I can register for the application by entering my email and password.	2	Medium	UDAYAN R
Sprint-3	Login	USN-11	As a user, I can log into the application through Gmail.	3	High	UDAYAN R
Sprint-3	Dashboard	USN-12	As a user, I can able to interact with the dashboard	5	Low	UDAYAN R
Sprint-4	Web UI	USN-13	Create a dashboard using Node-red application	7	High	PRAVEEN KUMAR
Sprint-4	Web UI	USN-14	To make the user interact with the web based dashboard software	5	High	PRAVEEN KUMAR

6.2 Sprint Delivery Schedule

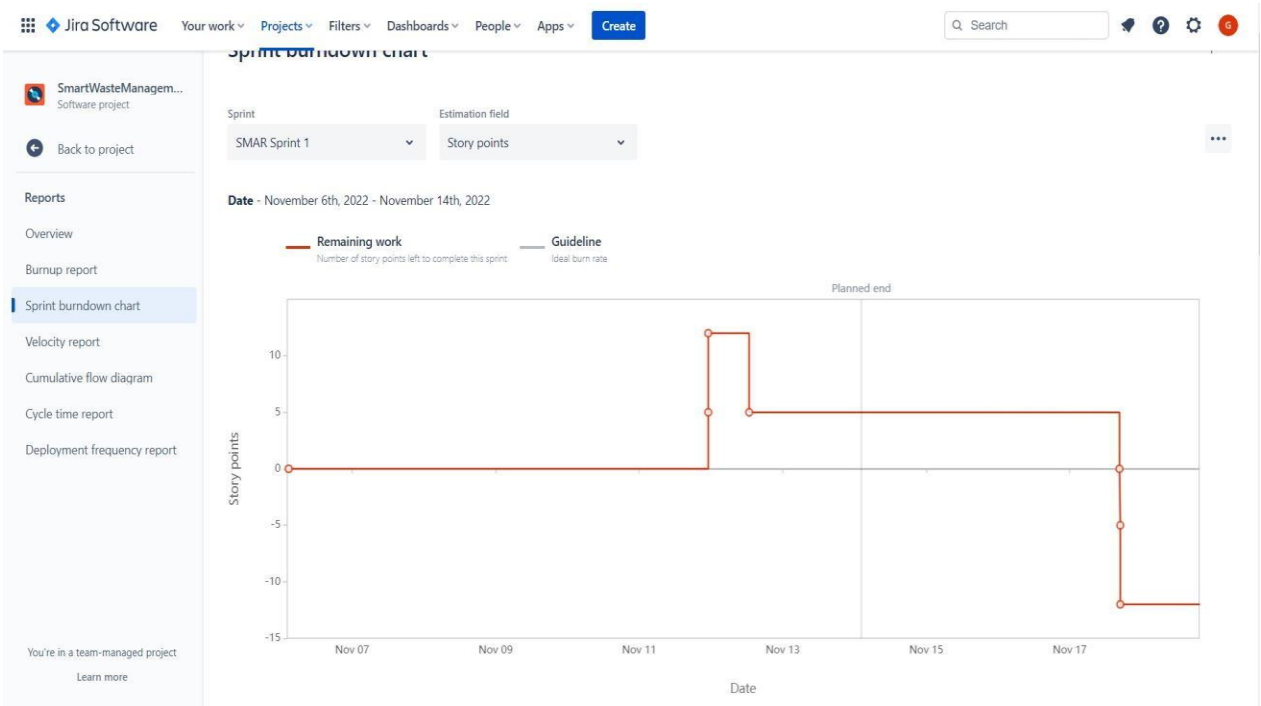
Sprint	Total StoryPoins	Duration	Sprint Start Date	Sprint End Date (Plane)	Story Points Completed (as on Planned End Date)	Sprint ReleaseDate(Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	7 Days	31 Oct 2022	06 Nov 2022	20	06 Nov 2022
Sprint-3	20	7 Days	07 Nov2022	13 Nov 2022	20	13 Nov2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.3 Reports from JIRA

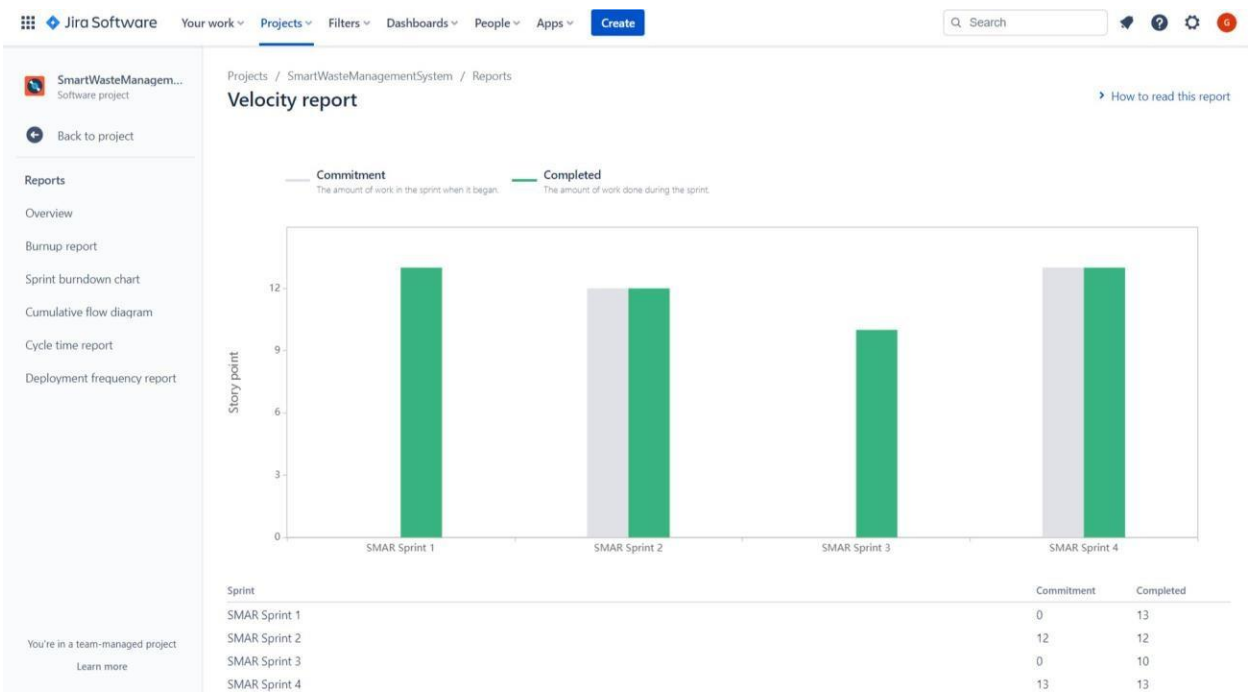
Roadmap:



Burndown chart:



Velocity Map:



7. CODING & SOLUTIONING

7.1 Feature 1-(Use of HX711 Load Cell & LCD Display)

- The output signal from the load cell is in the millivolt range, therefore we need an amplifier to raise the level so that we can process it as a digital signal later. We employ the HX711 amplifier sensor for this.
- It is specially made for amplifying signals from cells and reporting them to the microcontroller. We used load Cell to measure the load in the smart bin.
- They have many purposes including being able to tell the weight of something, weight change over time and also capable of measuring strain and load on a surface.
- They are ideal for many kinds of commercial scales that include weight, including electronic scales, price computing scales, electronic platforms scales, digital scales, parcel post scales, electronic balances, and more!

COMPONENTS:

S.NO	COMPONENTS	QUANTITY
1	Ultrasonic sensor	2
2	Servo motor	1
3	ESP32 Microcontroller	1
4	HX711 Load Cell	1
5	LCD 16*2 Display	1

DESCRIPTION:

The code will display the status of both bin level and bin load for a single smart bin. The status of the bin level is shown in the 16*2 LCD display. The data obtained from the sensors is then published to the IBM Watson Iot Cloud Platform. The sensor data can be visually depicted using node-red application.

CODE:

```
#include <ESP32Servo.h>
#include <LiquidCrystal_I2C.h>
#include <HX711.h>
#define DATA_PIN  12
#define CLOCK_PIN 14
#include <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
#define ORG "uuyxja"
#define DEVICE_TYPE "NodeMcu"
#define DEVICE_ID "12345"
#define TOKEN "23323850"
#define speed 0.034
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] ="iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=2;
const int echopin=15;
String command;
String data="";
long duration;
float dist;
LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16, 2);
Servo servo;
int trigPin1 = 2;
int echoPin1 = 15;
```

```
int trigPin2 = 18;
int echoPin2 = 5;
int duration1;
int distance1;
int duration2;
int distance2;
void setup()
{
Serial.begin(115200);
LCD.begin(16,2);
LCD.init();
LCD.backlight();
LCD.clear();
servo.attach(23);
Serial.begin(115200);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
wifiConnect();
mqttConnect();
}
void loop() {
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("subscribe to cmd OK");
}
else {
  Serial.println("subscribe to cmd FAILED"); }
  }
void publishData()
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
```

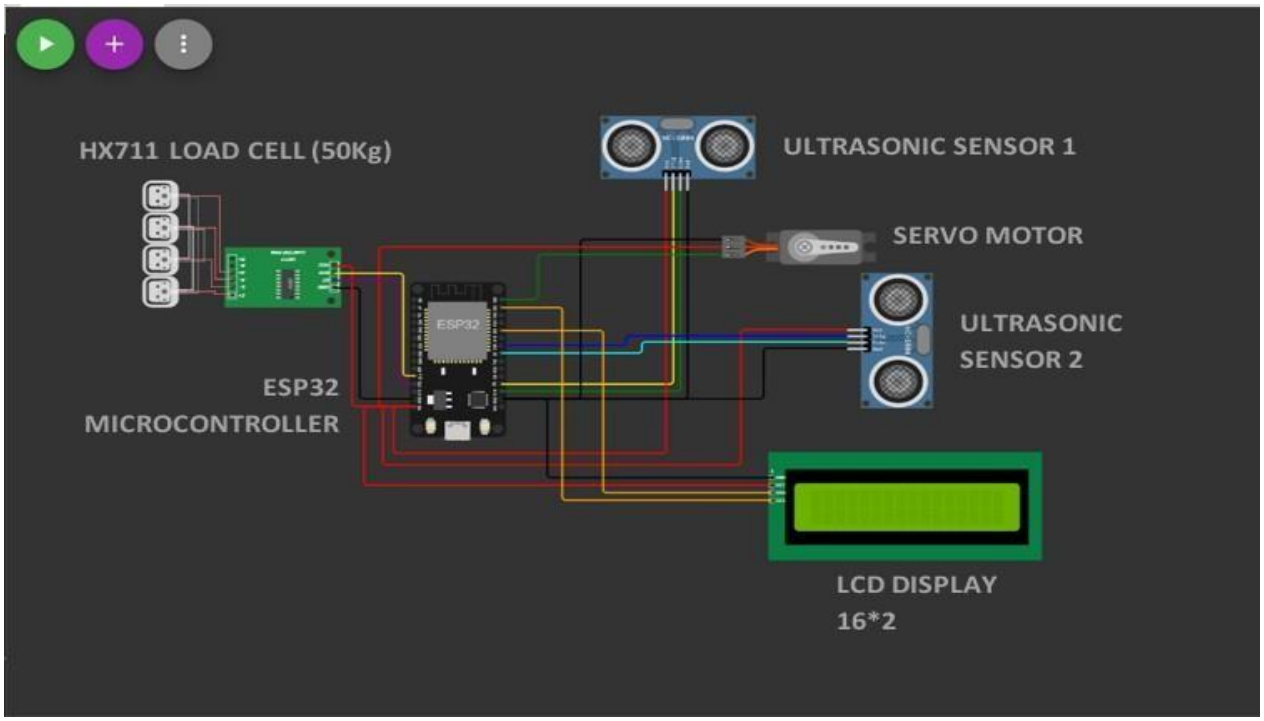
```
delayMicroseconds(10);
digitalWrite(trigPin1, LOW);
duration1 = pulseIn(echoPin1, HIGH);
//distance1= duration1*0.034/2;
delay(100);
digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
//distance2= duration2*0.034/2;
delay(100);

LCD.setCursor(0,1);
LCD.print("Fill Status ");
if(distance2>300 && distance2<=400){
  LCD.setCursor(12,1);
  LCD.print("25% ");
  String payload = "{\"Bin_Level\":";payload += "25";payload += ",\"Bin_Load \":";
  payload += "12.5";
  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");
      }
}
else if(distance2 > 200 && distance2<= 299){
  LCD.setCursor(12,1);
```

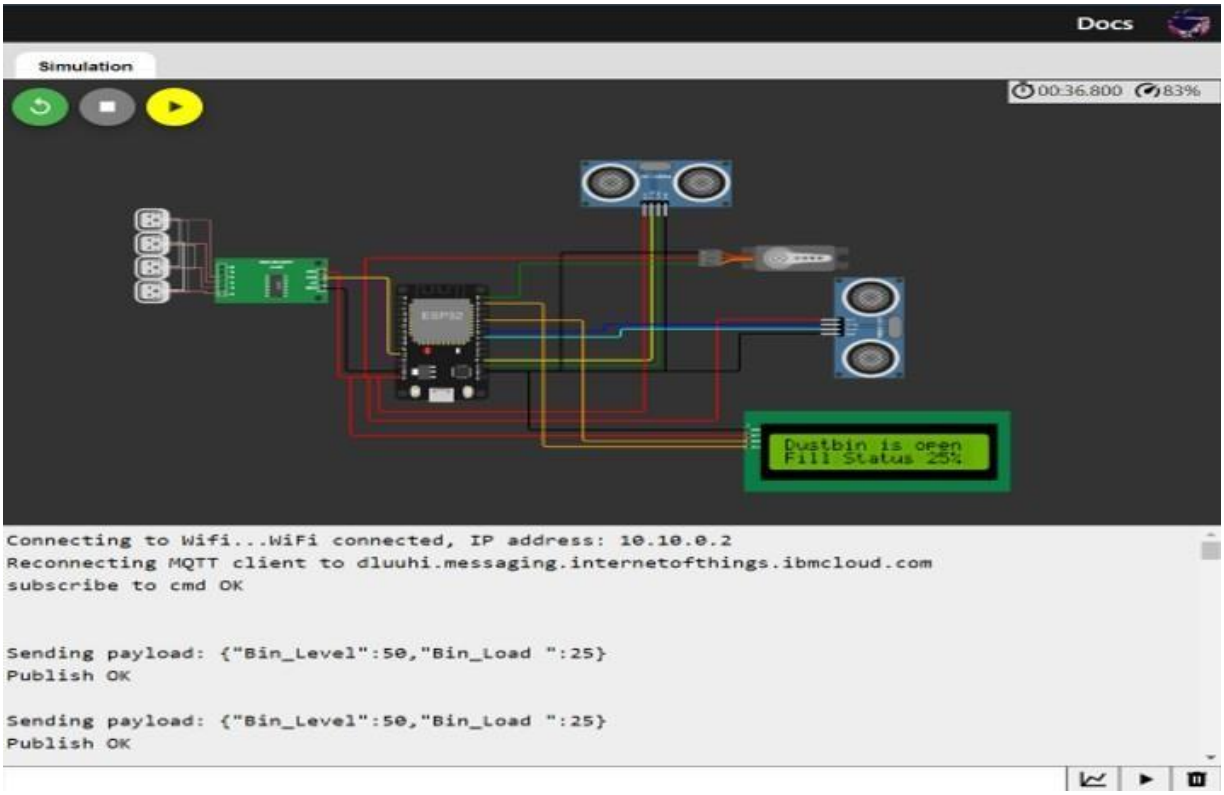
```
  LCD.print("50%");
  String payload = "{\"Bin_Level\":";payload += "50";payload += ",\"Bin_Load \":";
  payload += "25";payload += "}";
  Serial.print("\n");
  Serial.print("Sending payload: ");
  Serial.println(payload);
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
    } else {
      Serial.println("Publish FAILED");
      }
}
else if(distance2 >50 && distance2<=199){
  LCD.setCursor(12,1);
  LCD.print("75%");
  String payload = "{\"Bin_Level\":";payload += "75";payload += ",\"Bin_Load \":";
  payload += "37.5";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");
      }
}
else{
  LCD.setCursor(12,1);
  LCD.print("100%");
  String payload = "{\"Bin_Level\":";
  payload += "100";payload += ",\"Weight \":"; payload += "50";payload += "}";
  Serial.print("Sending payload: ");
  Serial.println(payload);
```

```
  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish OK");
    } else {
      Serial.println("Publish FAILED");
      }
}
if(distance1<=50){
  LCD.setCursor(0,0);
  LCD.print("Dustbin is open ");
  servo.write(90);
}else{
  LCD.setCursor(0,0);
  LCD.print("Dustbin is close ");
  servo.write(0);
} }
```

CIRCUIT DIAGRAM:



OUTPUT:



7.2 Feature 2 - (Python program to display the Location of Smart bin for garbage collectors and Truck drivers.)

DESCRIPTION:

The below code indicates the position (Latitude and Longitude) of each Smart Bin that has been deployed throughout the city, together with the percentage indicating the state of each bin. The garbage collectors can easily tell which bins are fully filled or which bins need to be collected thanks to the bin level's different colour display on the map based on its bin status.

For instance, if the trash can is 75% full(indicate in yellow colour), the smart bin will fill to its threshold level if more garbage is added. In contrast, a 100% fill level indicates that the bin has reached its maximum capacity and is displayed in red colour on the map so that garbage collector can collect the trash.

CODE:

```
import requests
import urllib.parse
import folium
address = 'kodambakkam'
address1 = 'virugambakkam'
address2 = 'Royapettah'
address3 = 'Ambattur'
address4 = 'Coimbatore'
str1 = input("Enter the string:")
if(str1 == address):
  url = 'https://nominatim.openstreetmap.org/search/' + urllib.parse.quote(address)
+'?format=json'
  response = requests.get(url).json()
  a = response[1]["lat"]
  b = response[1]["lon"]
  m = folium.Map(location =[a,b],zoom_start=15)
  folium.Marker(location =[a,b],popup=" kodambakkam",tooltip="click for more
```

```
info").add_to(m)
  folium.Marker(location =[13.0559, 80.2265],popup=" ",tooltip="click for more
info").add_to(m)
folium.Marker(location =[13.0418, 80.2341],popup=" ",tooltip="click for more
info").add_to(m)
folium.Marker(location =[13.0532, 80.1922],popup=" ",tooltip="click for more
info").add_to(m)
m
if(str1 == address1):
  url1 = 'https://nominatim.openstreetmap.org/search/' + urllib.parse.quote(address1)
+'?format=json'
  response = requests.get(url1).json()
  a = response[1]["lat"]
  b = response[1]["lon"]
  m = folium.Map(location =[a,b],zoom_start=15)
  folium.Marker(location =[a,b],popup=" ",tooltip="click for more info").add_to(m)
folium.Marker(location =[22.6992, 75.8671],popup=" ",tooltip="click for more
info").add_to(m)
folium.Marker(location =[13.0436, 80.1817],popup=" ",tooltip="click for more
info").add_to(m)
#folium.Marker(location =[13.0532, 80.1922],popup=" ",tooltip="click for more
info").add_to(m)
m
if(str1 == address2):
 url2 = 'https://nominatim.openstreetmap.org/search/' + urllib.parse.quote(address2)
+'?format=json'
  response = requests.get(url2).json()
  a = response[1]["lat"]
  b = response[1]["lon"]
```

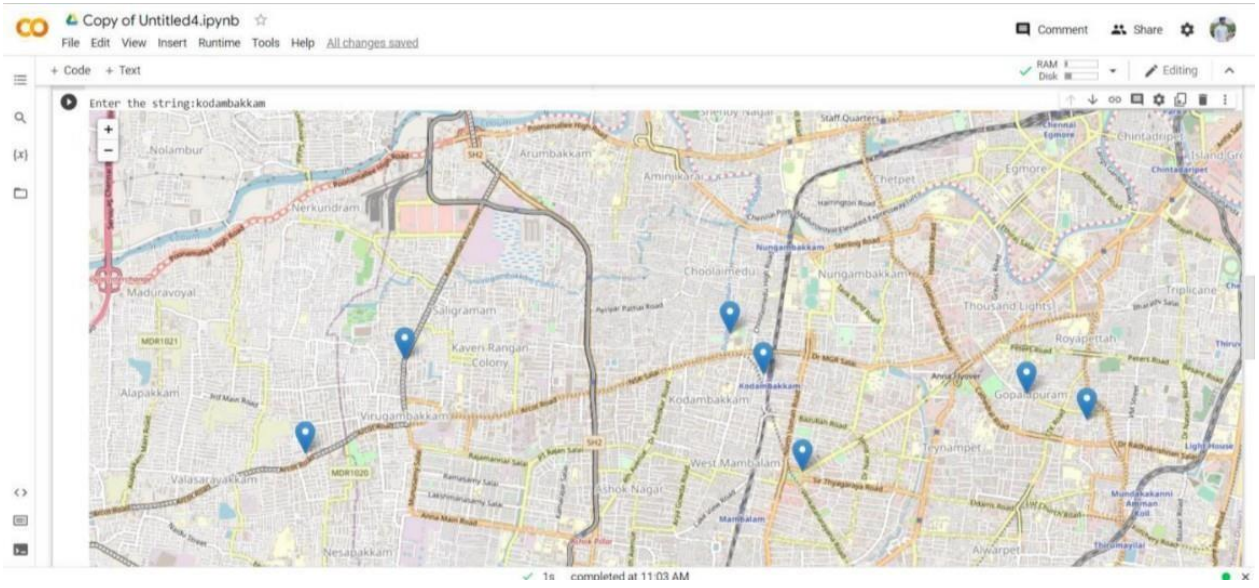
```
  m = folium.Map(location =[a,b],zoom_start=15)
  folium.Marker(location =[a,b],popup=" Royapettah",tooltip="click for more info").add_to(m)
folium.Marker(location =[ 13.0497, 80.2577],popup=" kodambakkam",tooltip="click for more
info").add_to(m)
folium.Marker(location =[13.047059,80.264052],popup="Harshitha's home",tooltip="click for
more info").add_to(m)
m
if(str1 == address3):
  url2 = 'https://nominatim.openstreetmap.org/search/' + urllib.parse.quote(address3)
+'?format=json'
  response = requests.get(url2).json()
  a = response[1]["lat"]
  b = response[1]["lon"]
  m = folium.Map(location =[a,b],zoom_start=15)
  folium.Marker(location =[a,b],popup=" Ambattur",tooltip="click for more info").add_to(m)
folium.Marker(location =[13.0892, 80.1613],popup=" ",tooltip="click for more
info").add_to(m)
folium.Marker(location =[13.1142, 80.1527],popup=" ",tooltip="click for more info").add_to(m)
folium.Marker(location =[13.1197, 80.1501],popup=" ",tooltip="click for more info").add_to(m)
m
if(str1 == address4):
  url3 = 'https://nominatim.openstreetmap.org/search/' + urllib.parse.quote(address4)
+'?format=json'
  response = requests.get(url3).json()
 a = response[1]["lat"]
 b = response[1]["lon"]
m = folium.Map(location =[a,b],zoom_start=15)
folium.Marker(location  =[a,b],popup="Coimbatore",tooltip="click  for  more  info").add_to(m)
folium.Marker(location =[11.0242, 77.0028],popup=" ",tooltip="click for more info").add_to(m)
```

folium.Marker(location =[10.822477, 77.016144],popup=" ",tooltip="click for more info").add_to(m)

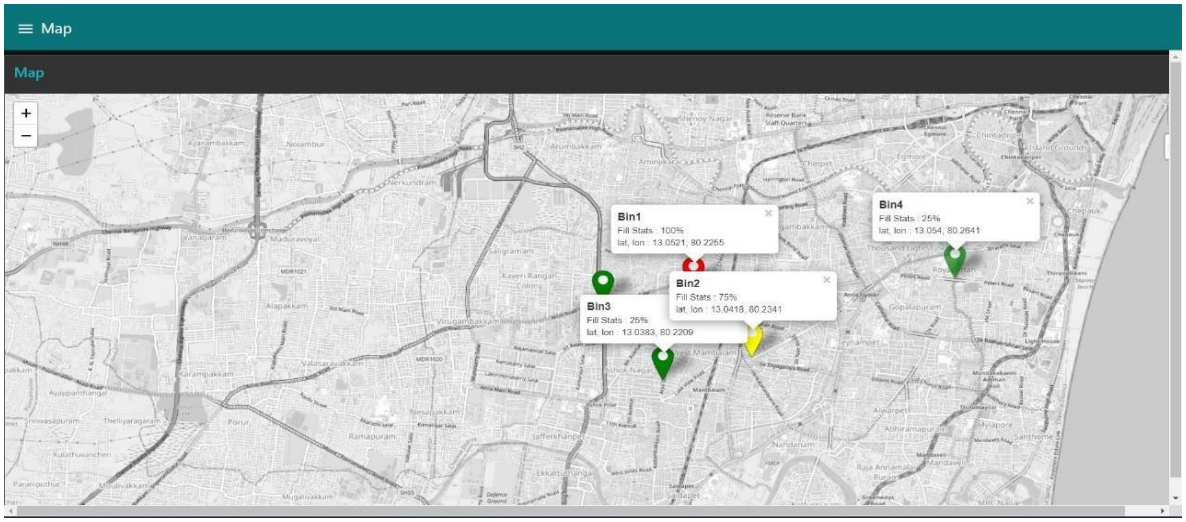
folium.Marker(location =[11.0102, 76.9504],popup=" ",tooltip="click for more info").add_to(m)

m

DEMO OUTPUT IN GOOGLE COLAB:



FINAL OUTPUT IN NODE-RED WEB UI:



8. TESTING

8.1 Test Cases

Test Scenarios

1 Verify user is able to see login form page
2 Verify user is able to login to application or not?
3 Verify user is able to navigate to the Login page.
4 Verify user is able to provide the required information.
5 Veriify login page elements

Dashboard

1 Verify user is able see the dashboard page
2 User can view the visual data displayed on the screen.

MAP

1 User can view the Bin Location on the map

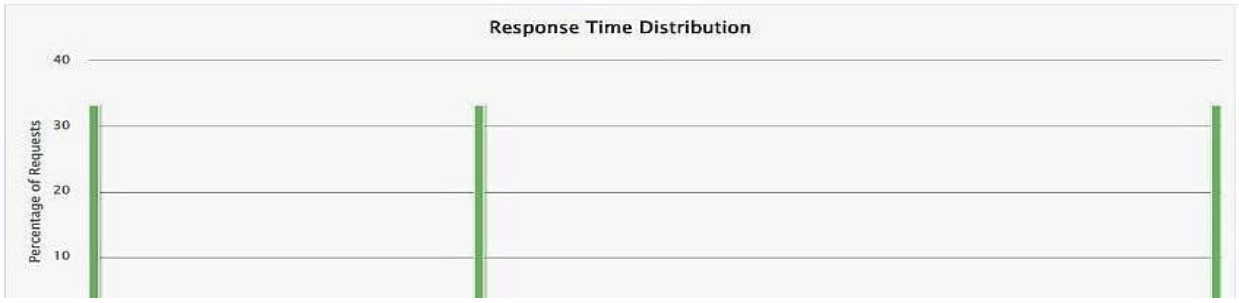
9. RESULTS

9.1 Performance Metrics

			NFT - Risk Assessment						
S.No	Project Name	Scope/feature	Functional Changes	Hardware Changes	Software Changes	Impact of Downtime	Load/Voluem Changes	Risk Score	Justification
1	Management System	Existing	Low	No Changes	Moderate	-	>5 to 10%	ORANGE	As we have seen the chnages

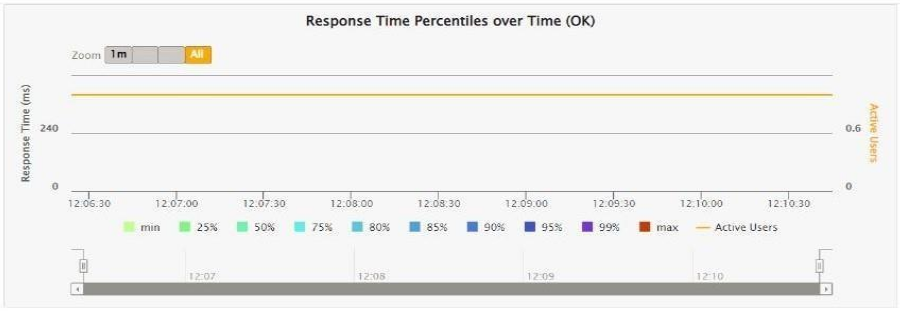
	NFT - Detailed Test Plan			
S.No	Project Overview	NFT Test approach	umptions/Dependencies/R	Approvals/SignOff
1	Smart Waste Management System F	Load	if the login page crashes/ devel	approved

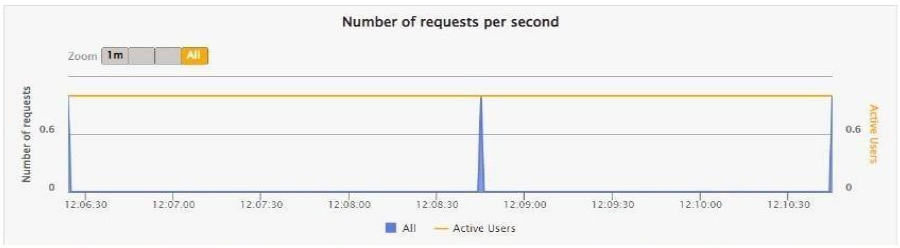
	End Of Test Report							
S.No	Project Overview	NFT Test approach	NFR - Met	Test Outcome	GO/NO-GO decision	Recommendations	Identified Defects (Detected/Closed/Open)	Approvals/SignOff
1	Smart Waste Manager	load	Met	system is functional and is supportin	GO Decisio	None	closed	Approved



Gatling Documentation

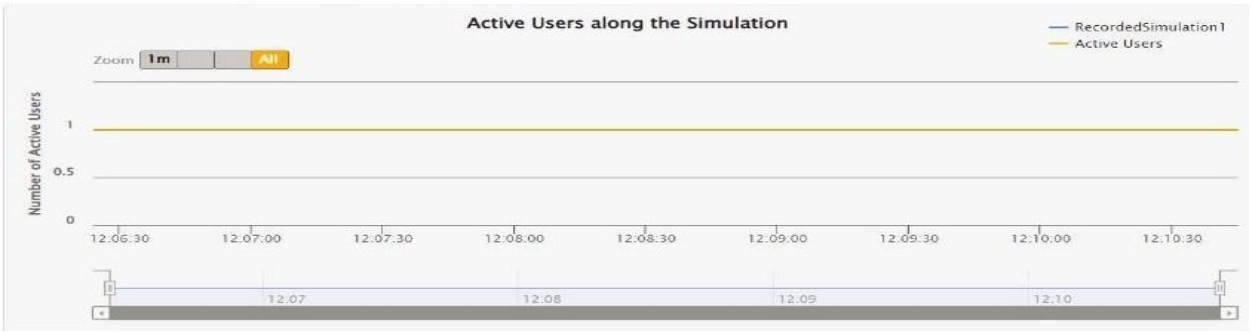
Ranges
Stats
Active Users
Requests / sec
Responses / sec





Expand all groups | Collapse all groups

Requests	Executions					Response Time (ms)							
	Total	OK	KO	% KO	Cnt/s	Min	50th pct	75th pct	95th pct	99th pct	Max	Mean	Std Dev
All Requests	3	3	0	0%	0.011	62	196	324	426	447	452	237	162
request_0	1	1	0	0%	0.004	62	62	62	62	62	62	62	0
request_3	1	1	0	0%	0.004	196	196	196	196	196	196	196	0
request_11	1	1	0	0%	0.004	452	452	452	452	452	452	452	0

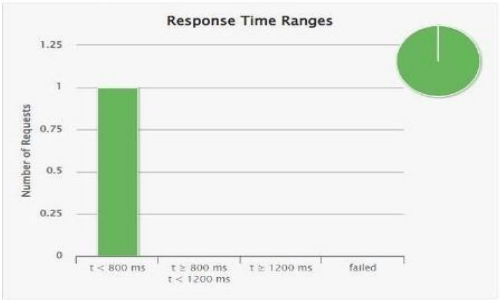


Gatling Documentation

request_0
request_3
request_11

RecordedSimulation1

Global | Details



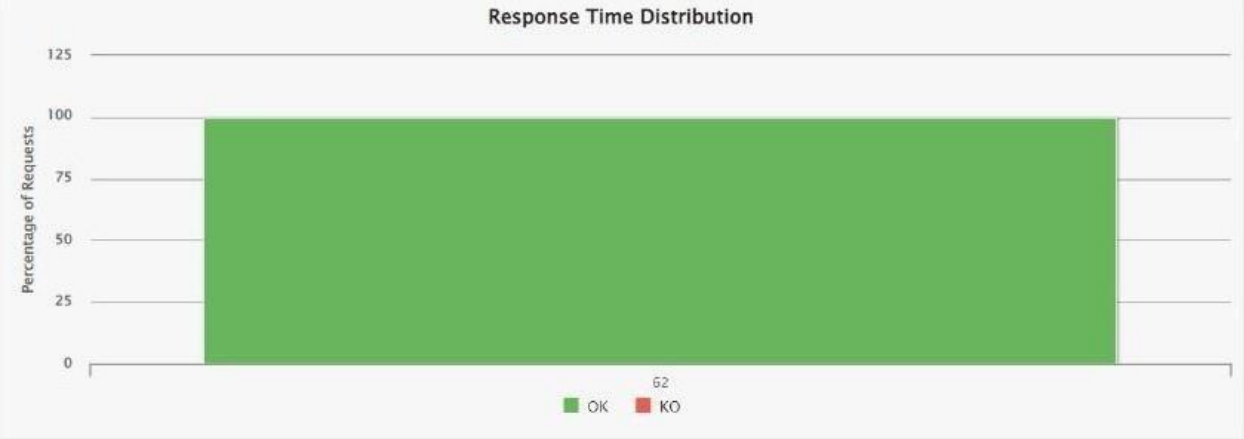
Stats

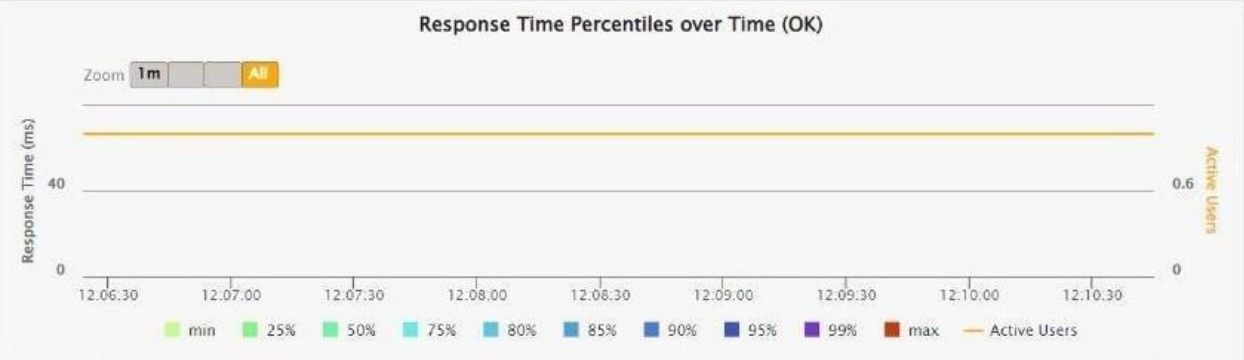
Executions

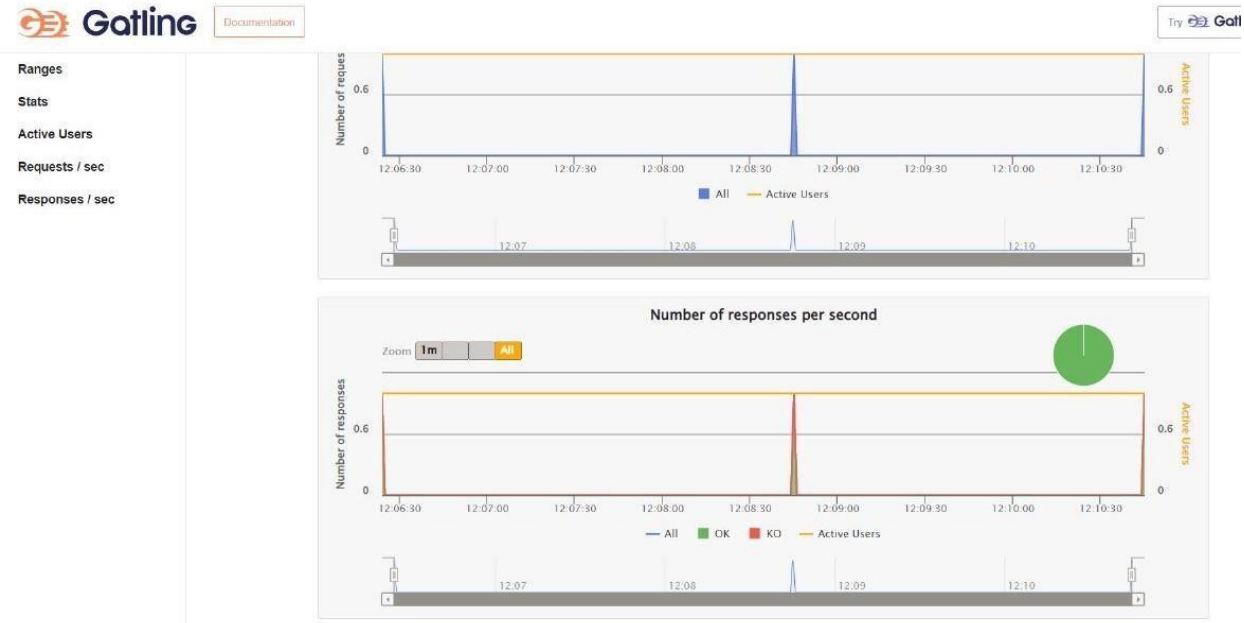
	Total	OK	KO
Total count	1	1	0
Mean count/s	0.004	0.004	-

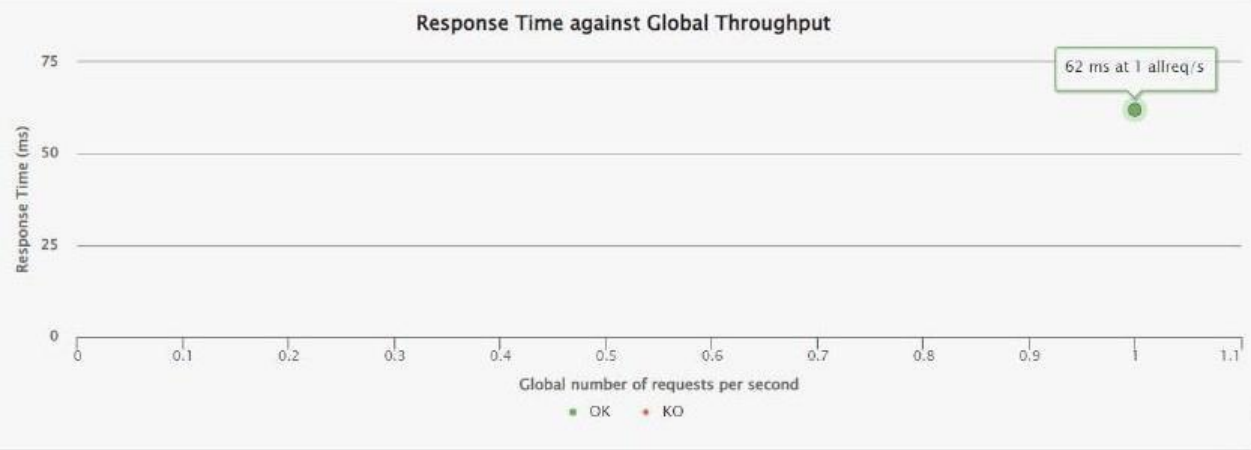
Response Time (ms)

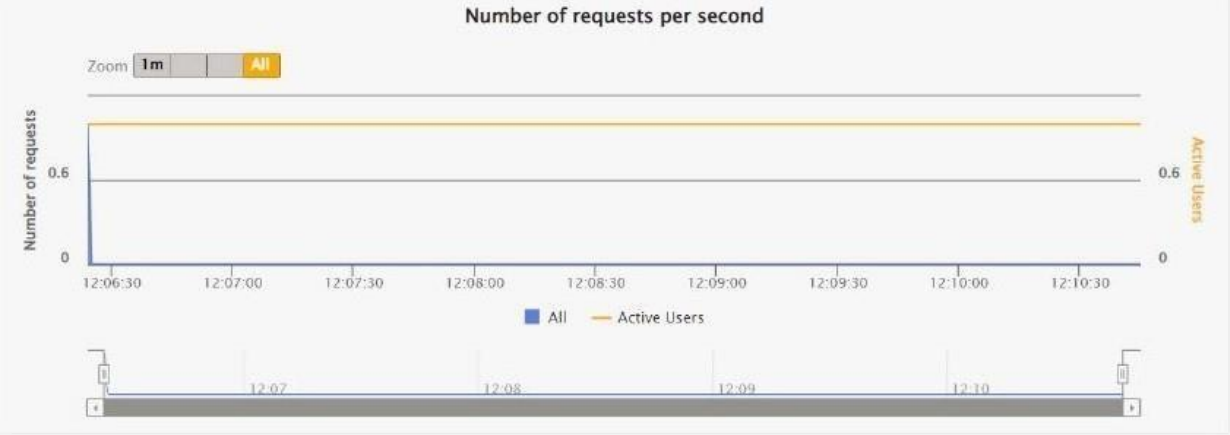
	Total	OK	KO
Min	62	62	-
50th percentile	62	62	-
75th percentile	62	62	-
95th percentile	62	62	-
99th percentile	62	62	-
Max	62	62	-
Mean	62	62	-
Standard Deviation	0	0	-

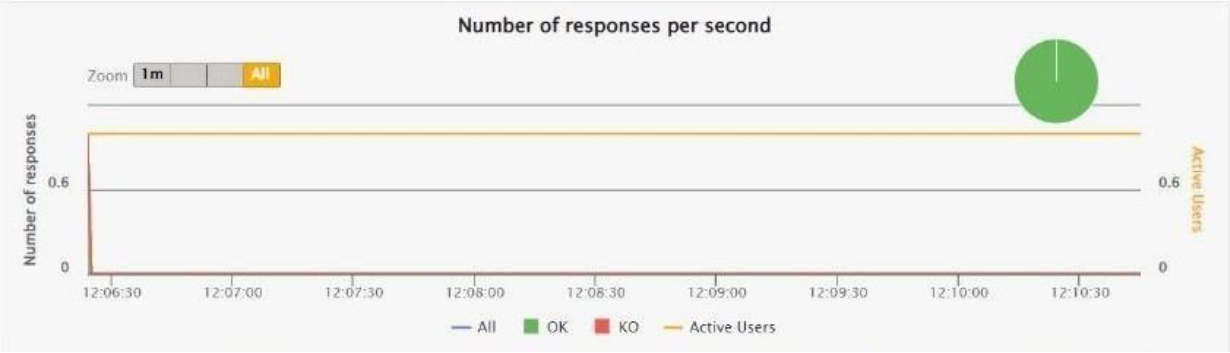

Response Time Distribution
125
100
75
50
25
0
Percentage of Requests
62
OK
KO


Response Time Percentiles over Time (OK)
Zoom
1m
All
Response Time (ms)
40
0
0.6
0
Active Users
12:06:30
12:07:00
12:07:30
12:08:00
12:08:30
12:09:00
12:09:30
12:10:00
12:10:30
min
25%
50%
75%
80%
85%
90%
95%
99%
max
Active Users


Gatling
Documentation
Try Gatl
Ranges
Stats
Active Users
Requests / sec
Responses / sec
Number of reques
0.6
0
0.6
0
Active Users
12:06:30
12:07:00
12:07:30
12:08:00
12:08:30
12:09:00
12:09:30
12:10:00
12:10:30
All
Active Users
12:07
12:08
12:09
12:10
Number of responses per second
Zoom
1m
All
Number of responses
0.6
0
0.6
0
Active Users
12:06:30
12:07:00
12:07:30
12:08:00
12:08:30
12:09:00
12:09:30
12:10:00
12:10:30
All
OK
KO
Active Users
12:07
12:08
12:09
12:10


Response Time against Global Throughput
62 ms at 1 allreq/s
Response Time (ms)
75
50
25
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
1.1
Global number of requests per second
OK
KO


Number of requests per second
Zoom
1m
All
Number of requests
0.6
0
Active Users
0.6
0
12:06:30
12:07:00
12:07:30
12:08:00
12:08:30
12:09:00
12:09:30
12:10:00
12:10:30
All
Active Users
12:07
12:08
12:09
12:10


Number of responses per second
Zoom
1m
All
Number of responses
0.6
0
Active Users
0.6
0
12:06:30
12:07:00
12:07:30
12:08:00
12:08:30
12:09:00
12:09:30
12:10:00
12:10:30
All
OK
KO
Active Users

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

- Improved quality of life
- No missed pickups
- Pollution-free and odour-free surroundings
- Efficient and effective operation

DISADVANTAGES

- It needs well-structured hardware.
- The initial installation cost will be greater .
- The system necessitates a higher number of garbage containers for separate waste collection in proportion to the city's population.
- Sensor nodes utilised in the dustbins have limited memory size.

11. CONCLUSION

This project seeks to address waste management by partnering with the use of IoT technology in order to address issues about public health and environmental damage caused by improper residential trash maintenance and disposal. The number of trips made by garbage collection vehicles is reduced as a result of this system's effective and efficient bin monitoring system, which also lowers the overall cost of waste collection. Our method can be used by the municipal authorities to control the trucks as well. By tracking the state of garbage collection in real-time and evaluating the team's performance, the suggested approach will help in overcoming any significant obstacles and preserving a clean environment.

12. FUTURE SCOPE

- The moisture sensor can be implemented hand in hand with the other sensors and the compartments for segregating the dry and wet waste can be created which will solve the issues related to waste segregation.
- Implementation of methane and smell sensors can helps in distinguishing the waste at the source and hence reducing the requirement of manpower .
- Also Image processing with a camera as a passive sensor might be employed, however this is an expensive option.

13. APPENDIX

Source Code

```
#include <ESP32Servo.h>
#include <LiquidCrystal_I2C.h>
#include <HX711.h>
#define DATA_PIN  12
#define CLOCK_PIN 14

#include  <WiFi.h>
#include <PubSubClient.h>
WiFiClient wifiClient;
#define ORG "uuyxja"
#define DEVICE_TYPE "NodeMcu"
#define DEVICE_ID "12345"
#define TOKEN "23323850"
#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char topic[] ="iot-2/cmd/home/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
PubSubClient client(server, 1883, wifiClient);
void publishData();
const int trigpin=2;
const int echopin=15;
String command;
String data="";
long duration;
float dist;

LiquidCrystal_I2C LCD = LiquidCrystal_I2C(0x27, 16, 2);
Servo servo;
int trigPin1 = 2;
```

```
int echoPin1 = 15;
int trigPin2 = 18;
int echoPin2 = 5;
int duration1;
int distance1;
int duration2;
int distance2;

void setup()
{
Serial.begin(115200);
LCD.begin(16,2);
LCD.init();
LCD.backlight();
LCD.clear();

servo.attach(23);
Serial.begin(115200);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
wifiConnect();
mqttConnect();
}

void loop() {
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("subscribe to cmd OK");
}
else {
  Serial.println("subscribe to cmd FAILED"); }
  }
void publishData()
{
  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
```

```
duration1 = pulseIn(echoPin1, HIGH);
distance1= duration1*0.034/2;
//Serial.println(distance1);
delay(100);

digitalWrite(trigPin2, LOW);
delayMicroseconds(2);
digitalWrite(trigPin2, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin2, LOW);
duration2 = pulseIn(echoPin2, HIGH);
distance2= duration2*0.034/2;
//Serial.println(distance2);
delay(100);

LCD.setCursor(0,1);
LCD.print("Fill Status ");


if(distance2>300 && distance2<=400){
  LCD.setCursor(12,1);
  LCD.print("25% ");
  String payload = "{\"Bin_Level\":";
  payload += "25";
  payload += ",\"Weight\":";
  payload += "12.5";
  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");
      }
}
else if(distance2 > 200 && distance2<= 299){
```

```
  LCD.setCursor(12,1);
  LCD.print("50%");
  String payload = "{\"Bin_Level\":";
  payload += "50";
  payload += ",\"Weight\":";
  payload += "25";
  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");
      }
}
else if(distance2 >50 && distance2<=199){
  LCD.setCursor(12,1);
  LCD.print("75%");
  String payload = "{\"Bin_Level\":";
  payload += "75";
  payload += ",\"Weight\":";
  payload += "37.5";
  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");
      }
}
else{
  LCD.setCursor(12,1);
  LCD.print("100%");
  String payload = "{\"Bin_Level\":";
```

```
    payload +=  "100";
    payload += ",\"Weight\":";
    payload += "50";
    payload += "}";
    Serial.print("Sending payload: ");
    Serial.println(payload);
    if (client.publish(publishTopic, (char*) payload.c_str())) {
      Serial.println("Publish OK");
      } else {
        Serial.println("Publish FAILED");
        }
  }
  if(distance1<=50){
    LCD.setCursor(0,0);
    LCD.print("Dustbin is open ");
    servo.write(90);
  }
  else{
    LCD.setCursor(0,0);
    LCD.print("Dustbin is close ");
    servo.write(0);
  }
}
```

GitHub & Project Demo Link

GitHub link:

https://github.com/IBM-EPBL/IBM-Project-37523-1660311128.git

Wokwi Link:

https://wokwi.com/projects/348653337941377619

Demo Link:

https://drive.google.com/file/d/1dC5N1vlBOPiquucuxFiTzV5rvTPsc0r4/view?usp=sharing -