SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

Domain – Internet of Things (IoT)

Project Report

Team ID : PNT2022TMID15172

Team Members:

Team Leader: Koliyar Nikhil Durairaj

Team Member 1: Obuli Hariharan R

Team Member 2: Vasanthan V

Team Member 3: Poovarasan P

Industry Mentor(s) Name: Dinesh

Faculty Mentor(s) Name: K JASMINE

College: Kumaraguru College of Technology

Coimbatore

2022 – 2023

CONTENT

1. INTRODUCTION

Waste has become a major concern for all of us due to global population growth and industrialization of countries. Over the years, in the era of globalization, academics have come to the conclusion that waste management alone is not enough to effectively treat and dispose of waste. Researchers have developed IoT-based smart waste management initiatives and solutions with the help of technology that streamlines the time and energy required to provide waste management services and reduces the amount of waste produced. Unfortunately, many variables, including the socioeconomic environment, prevent developing countries from implementing those current solutions. We have focused on developing an intelligent Internet of Things based waste management system for developing countries like India to ensure effective household waste disposal, collection, transportation, and recycling.

1.1. Project Overview

IOT-based dustbins are used in this smart waste management project to collect waste and monitor its volume inside the bin. Two ultrasonic sensors are used in the system, which is controlled by the Node MCU. An ultrasonic sensor detects the amount of waste in the bin, and another detects the person approaching the bin to remove the waste. This detection enables automatic opening and closing of the lid. The lid is connected to a servo motor which helps in closing and opening the lid. This device will inform the authorities concerned about the amount of garbage in the bin. Applications are used to monitor and store IoT data.

1.2. Purpose

We amalgamate technology along with waste management in order to effectively create a safe and a hygienic environment. Smart waste management is about using technology and data to create a more efficient waste industry. Based on IoT (Internet of Things) technology, smart waste management aims to optimize resource allocation, reduce running costs, and increase the

sustainability of waste services. This makes it possible to plan more efficient routes for the trash collectors who empty the bins, but also lowers the chance of any bin being full for over a week. A good level of coordination exists between the garbage collectors and the information supplied via technology. This makes them well aware of the existing garbage level and instigate them whenever the bins reach the threshold level. They are sent with alert messages so that they can collect the garbage on time without littering the surrounding area. The fill patterns of specific containers can be identified by historical data and managed accordingly in the long term. In addition to hardware solutions, mobile applications are used to overcome the challenges in the regular waste management system, such as keeping track of the drivers while they are operating on the field. Thus, smart waste management provides us with the most optimal way of managing the waste in an efficient manner using technology.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

In the existing system garbage is collected by the corporation weekly once or twice. Sometimes the garbage stinks and overflows from the bin and spread over the roads and pollutes the environment. This also produces a heavy air pollution and routes to various air-borne diseases Many a times the street dogs and other animals eat these waste and scatter these waste around the surroundings which creates the spread of various diseases and situation of unclean environment.

Disadvantages of existing system:

- Time consuming and less effective.
- Overflow of waste from the bin.
- Unhygienic Environment and look of the city.
- Stinky smell and unpleasant situations.

PROPOSED SYSTEM

In this proposed system there will be no issues repeated that of previous system. In this system the bin is designed in such a way that when the waste level reaches the threshold limit automatically closes the bin and intimates the alert to the admin. The bins are provided with low cost embedded device which helps in tracking the level of the garbage bins and a unique ID will be provided for every dustbin in the city. These details can be accessed by the concern authorities from their place with the help of internet and an immediate action can be made to clean the bin. The admin can monitor the level of the bin and can trace the location where it exists.

Advantages:

- Real time information on the fill level of the dustbin.
- Deployment of dustbin based on the actual needs.
- Cost Reduction and resource optimization.
- Improves Environment quality.

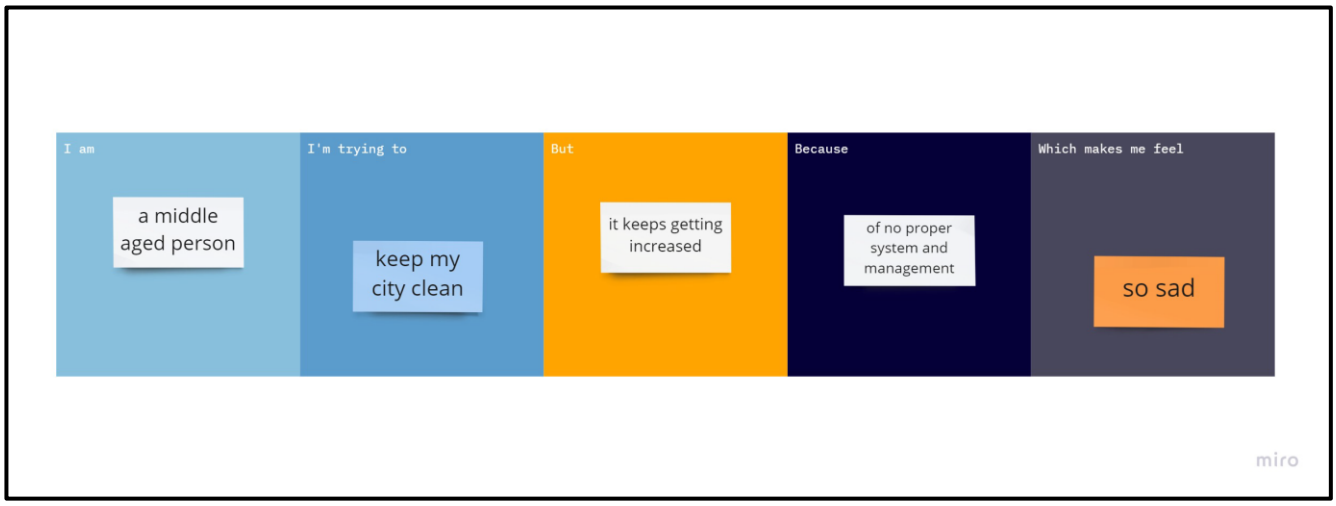
2.2 References

PAPER TITLE	AUTHOR	OUTCOME
IoT Based Smart Garbage System.	14. T.Sinha 15. 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.
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.

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.
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.
Analysis of Load cell.	16. Ranjeet Kumar 17. 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”.

Problem Statement

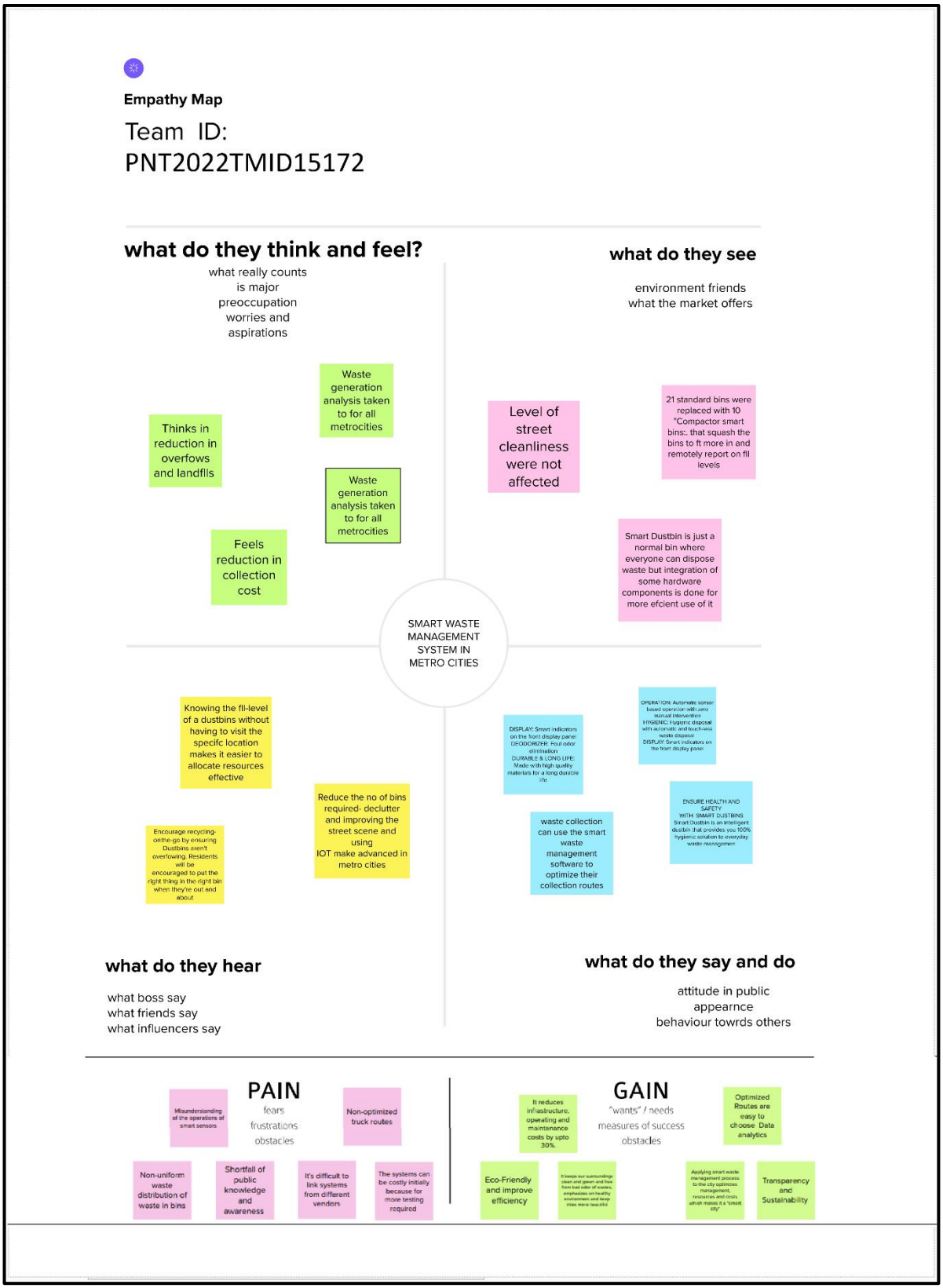
The waste management system provided earlier are not very reliable, efficient, cost effective and does not have any advanced processing features like automatic close of bin and alert intimations system. The following is a well articulated problem statement that allows you to find the ideal solution for the challenges faced.



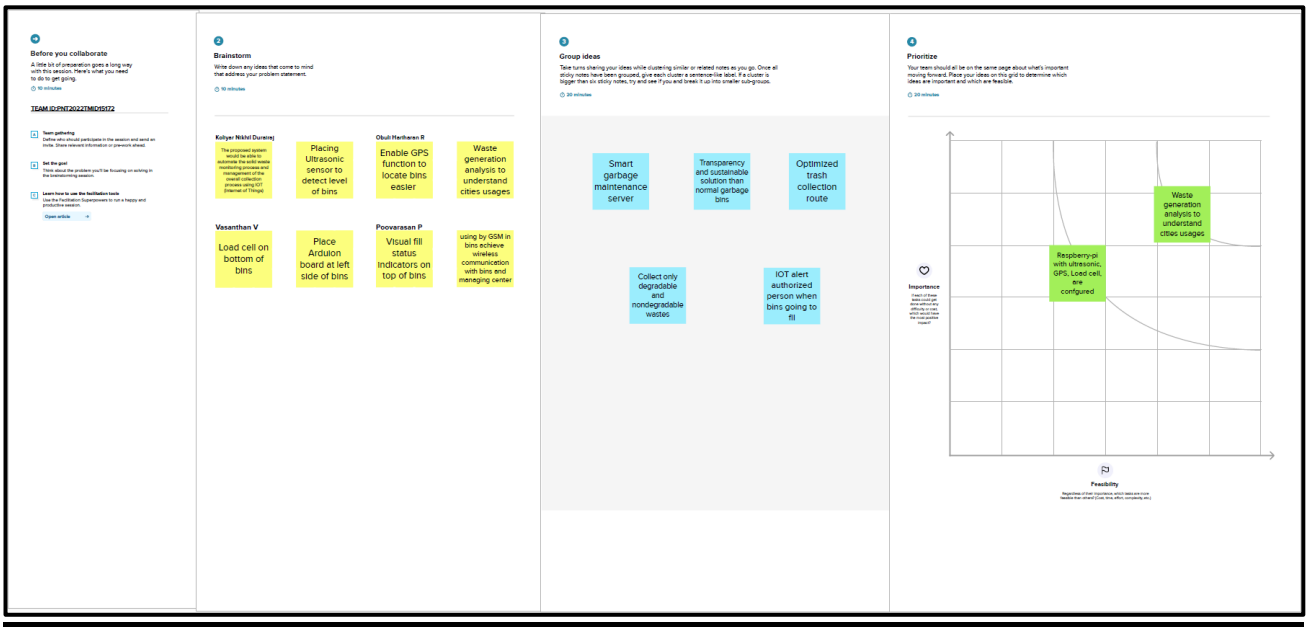
3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Empathy map on Smart waste Management System for Metropolitan cities.



3.2. Ideation and Brainstorming

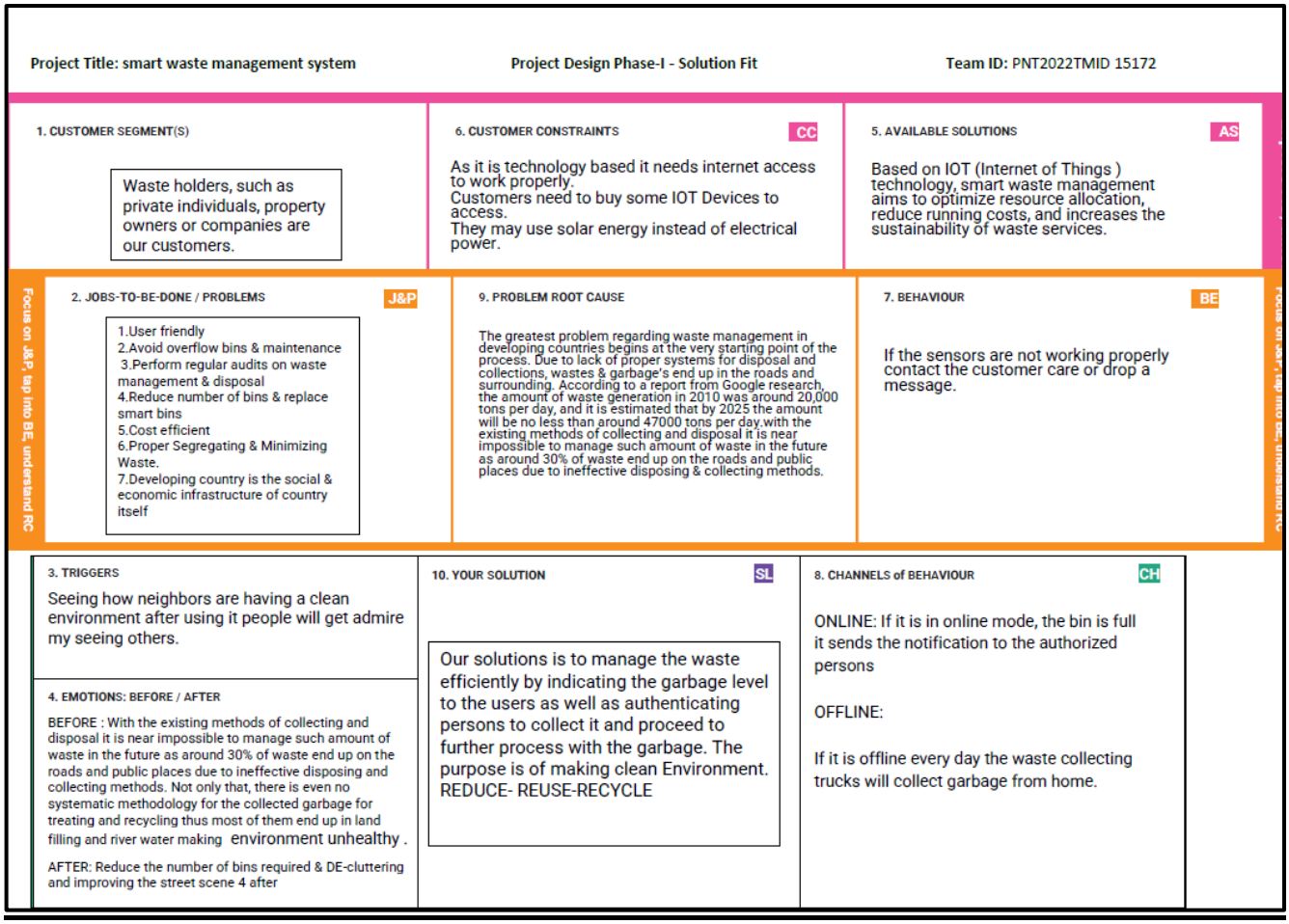


3.3. PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Rubbish and waste can cause air and water pollution. Rotting garbage is also known to produce harmful gases mix with the air and cause breathing problem in people. Due to improper waste disposal, we may face several problems like unpleasant odour and health problems
2.	Idea / Solution description	To solve this problem of waste management for disposal using a smart refuse bin built with technologies like Sensors, Arduino Yun. Garbage truck Weighing Mechanisms. AI Recycling Robots.
3.	Novelty / Uniqueness	We are going to establish SWM in our college but the real hard thing is that janitor (cleaner) don't know to operate these thing practically so here 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	n From the public perception as worst impacts of present solid waste disposal practices are seen direct social impacts such as neighbourhood of landfills to communities, breeding of pests and loss in property values
5.	Business Model (Revenue Model)	It generates revenue through the provision of various waste management and disposal services. Recycling solutions to residential ,commercial ,industrial ,and municipal clients
6.	Scalability of the Solution	Installing more bins fir collecting recyclables like paper, glass, plastic . Recycling not only save energy but also prevent the material from going to landfills & Incineration and provides raw materials for new products.

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.	Waste which are filled in bins are monitored by sensors. Based on the previous data, the tool predicts when will the bin fill. Smart sensor recognize each and every action takesplace. Hence it will check the last collected data. With the real time data & predictions, we can eliminate the overflowing of bins.
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-bin-discharge 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 un-efficient picks.	The sensor recognize picks. By the data filled on the bin, pick recognition, we can show how full the bins you collect are. Eliminates the collection of empty bins.

		. 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.	Based on current bin fill-levels and predictions of reaching full capacity, we have ready to respond and schedule. We have to compare planned and executed routes to identify any inconsistencies

4.2 Non-functional Requirements:

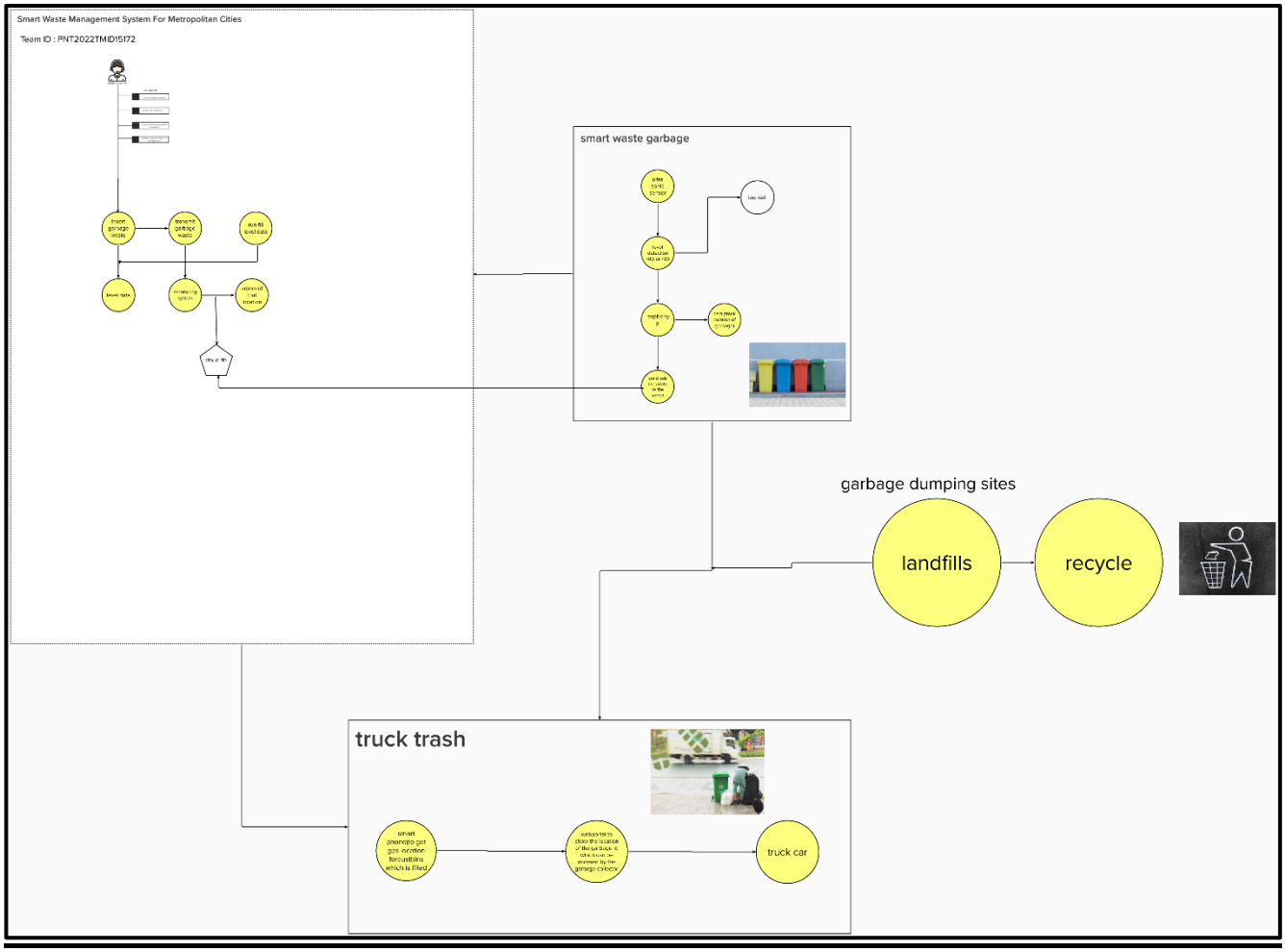
Following are the non-functional requirements of the proposed solution.

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 a reusable bottles Use reusable grocery bags Purchase wisely and recycle
NFR-3	**Reliability**	This project (Smart waste management system) is all about creating better work experience for waste collectors and drivers. Waste collector will spend their time more efficiently instead of driving the same collection routes and servicing empty bins.
NFR-4	**Performance**	By using the various IoT networks, the sensors send the data to smart waste management software system, a cloud platform, for data-driven daily operations, and available waste. User are provided with data-driven decision making, and optimization of waste collection route reduction by at least 35%

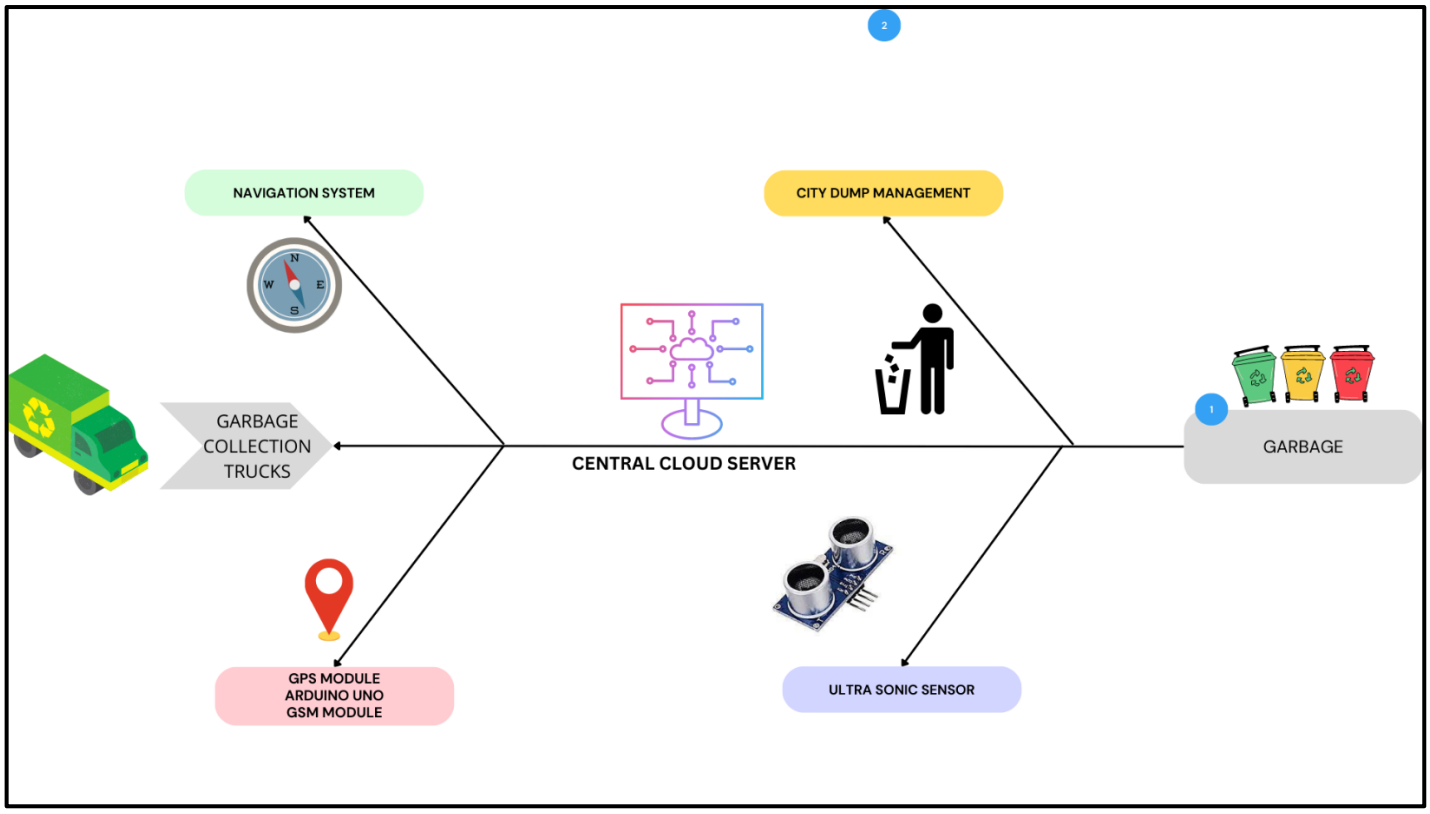
NFR-5	**Availability**	By developing & deploying resilient hardware and beautiful software we empower cities, businesses, and countries to manage waste smarter.
NFR-6	**Scalability**	Using smart waste bins reduce the number of bins inside town , cities coz we able to monitor the garbage 24/7 more cost efficient and scalability when we move to smarter.

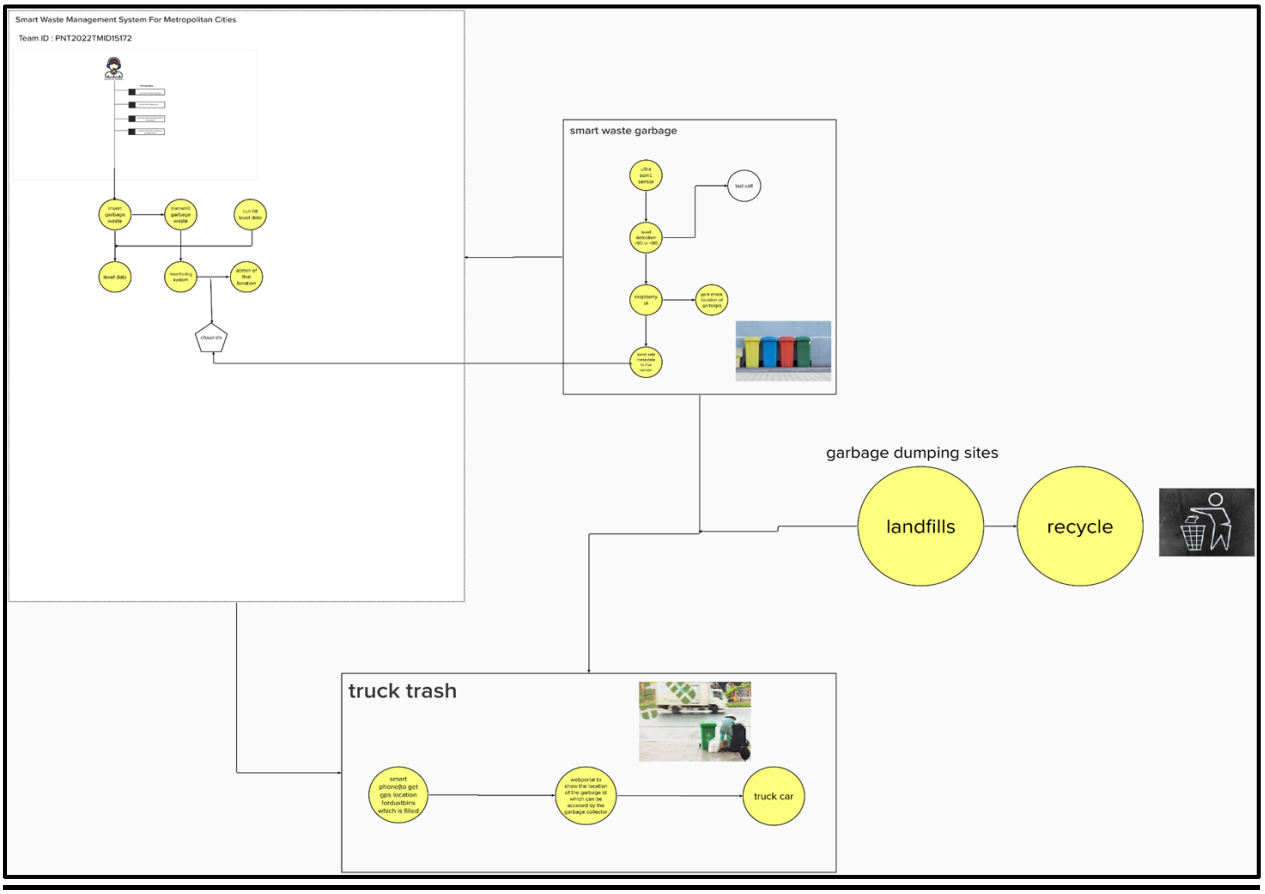
5. PROJECT DESIGN

5.1 . DATA FLOW DIAGRAM



5.2. SOLUTION ARCHITECTURE AND TECHNICAL ARCHITECTURE





5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user / Web user)	Signup	USN-1	User can signup using their email and password and confirm the details.	I can access my account / dashboard	High	Sprint-1
		USN-2	A confirmation mail is sent to the user.	I can receive confirmation email & click confirm	High	Sprint-1
	Login	USN-3	User can login using login credentials	User can log on to the website	High	Sprint-1
	Dashboard	USN-4	User can specify the location and area to check the availability of bins.	User can access dashboard and search for bins in specified areas	High	Sprint-2
		USN-5	User can post the queries and grievances in the report section	Options are provided to solve user issues	Medium	Sprint-2

6. 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.	24 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	24 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	24 SEPTEMBER 2022

Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	24 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	02OCTOBER 2022
Solution Architecture	Prepare solution architecture document.	01 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application.	15 OCTOBER 2022
Functional Requirement	Prepare the functional requirement document.	8 OCTOBER 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	15 OCTOBER 2022
Technology Architecture	Preparethetechnology architecture diagram.	17 OCTOBER 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	26 OCTOBER 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop & submit the developed code by testing it.	29 October 2022 05 November 2022 12 November 2022 19 November 2022

6.2. SPRINT DELIVERY SCHEDULE

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a Administrator, I need to give user id and pass code for ever workers over there in municipality	10	High	Koliyar Nikhil Durairaj
Sprint-1	Login	USN-2	As a Co-Admin, I'll control the waste level by monitoring them via real time web portal. Once the filling happens, I'll notify trash truck with location of bin with bin ID	10	High	Obuli Hariharan R
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	Vasanthan
Sprint-3	Dashboard	USN-4	As a Local Garbage Collector, I'II gather all the waste from the garbage, load it onto a garbage truck, and deliver it to Landfills	20	Medium	Vasanthan
Sprint-4	Dashboard	USN-5	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems	20	High	Poovarasan

Project Tracker, Velocity & Burndown Chart: (4 Marks)

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

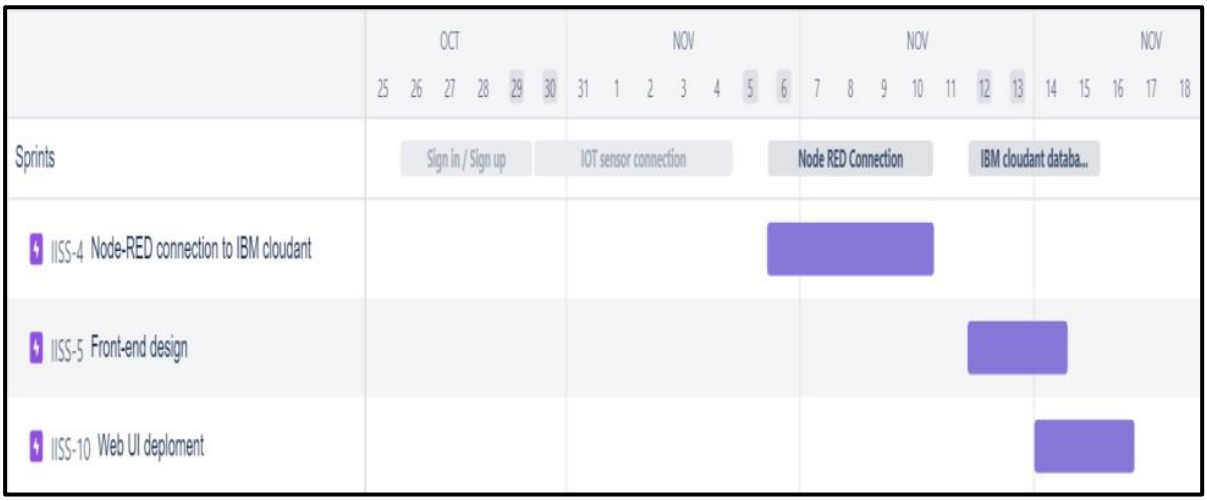
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

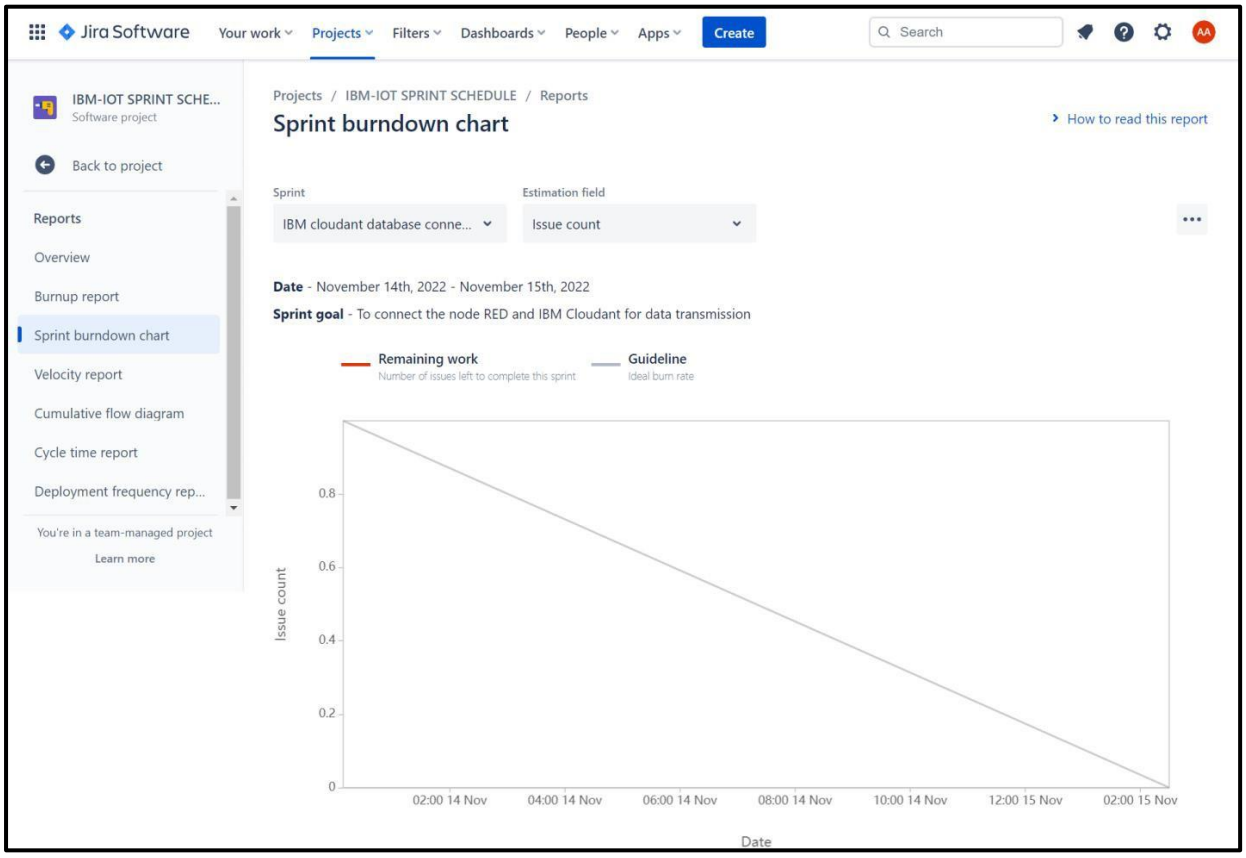
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

6.3 JIRA REPORTS

ROADMAP



SPRINT BURNDOWN CHART



7. CODING AND SOLUTION

7.1 Wokwi code for Sensor transmission

```
#include <WiFi.h>                                  // library for wifi
#include <PubSubClient.h>                          // library for MQTT
#include <LiquidCrystal_I2C.h>
#include <mjson.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

//---------------------- credentials of IBM Accounts ----------------------------

#define ORG "ffw1lq"                             // IBM organisation id
#define DEVICE_TYPE "Raspberry-pi"                    // Device type mentioned in ibm watson
iot platform
#define DEVICE_ID "12345"                  // Device ID mentioned in ibm watson iot platform
#define TOKEN "12345678"              // Token

//---------------------- customise above values ----------------------------------------
-------------

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";                // server
name
char publishTopic[] = "iot-2/evt/data/fmt/json";                                // topic
name and type of event perform and format in which data to be send
char topic[] = "iot-2/cmd/led/fmt/String";                                      // cmd
Represent type and command is test format of strings
char authMethod[] = "use-token-auth";                                           //
authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;                      //Client id

//-------------------------------------------------------------------------------------
------------------------------

WiFiClient wifiClient;                                                        // creating
instance for wificlient
PubSubClient client(server, 1883, wifiClient);

#define ECHO_PIN 12
#define TRIG_PIN 13
float dist;
String data3;
bool SealBin = true;
void setup()
```

```
{
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  //pir pin
  pinMode(34, INPUT);

  //ledpins
  pinMode(23, OUTPUT);
  pinMode(2, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(15, OUTPUT);


  lcd.init();
  lcd.backlight();
  lcd.setCursor(1, 0);
  lcd.print("");
  wifiConnect();
  mqttConnect();
}

float readcmCM()
{
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  int duration = pulseIn(ECHO_PIN, HIGH);
  return duration * 0.034 / 2;
}

void loop()
 {

  lcd.clear();

  publishData();
  delay(500);
  if (!client.loop())
    {
      mqttConnect();                                    // function call to connect to
IBM
    }
}
```

```
/* --------------------------------retrieving to cloud----------------------------------
-----------------*/

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("IBM subscribe to cmd OK");
      }
    else
      {
        Serial.println("subscribe to cmd FAILED");
      }
  }
void publishData()
{
  float cm = readcmCM();

  if(digitalRead(34))                                  //pir motion detection
  {
```

```
   Serial.println("Motion Detected");
   Serial.println("Lid Opened");
   digitalWrite(15, HIGH);



if(digitalRead(34)== true)
{
 if(cm <= 100)                                          //Bin level detection
 {
   digitalWrite(2, HIGH);
   Serial.println("High Alert!!!,Trash bin is about to be full");
   Serial.println("Lid Closed");
   lcd.print("Full! Don't use");
   delay(2000);
   lcd.clear();
   digitalWrite(4, LOW);
   digitalWrite(23, LOW);
 }
 else if(cm > 100 && cm < 180)
 {
   digitalWrite(4, HIGH);
   Serial.println("Warning!!,Trash is about to cross 50% of bin level");
   digitalWrite(2, LOW);
   digitalWrite(23, LOW);

 }
 else if(cm > 180)
 {
   digitalWrite(23, HIGH);
   Serial.println("Bin is available");
   digitalWrite(2,LOW);
   digitalWrite(4, LOW);

 }
   delay(10000);
   Serial.println("Lid Closed");
}
else
{
  Serial.println("No motion detected");
   digitalWrite(2, LOW);
   digitalWrite(15, LOW);
   digitalWrite(4, LOW);
   digitalWrite(23, LOW);
}
```

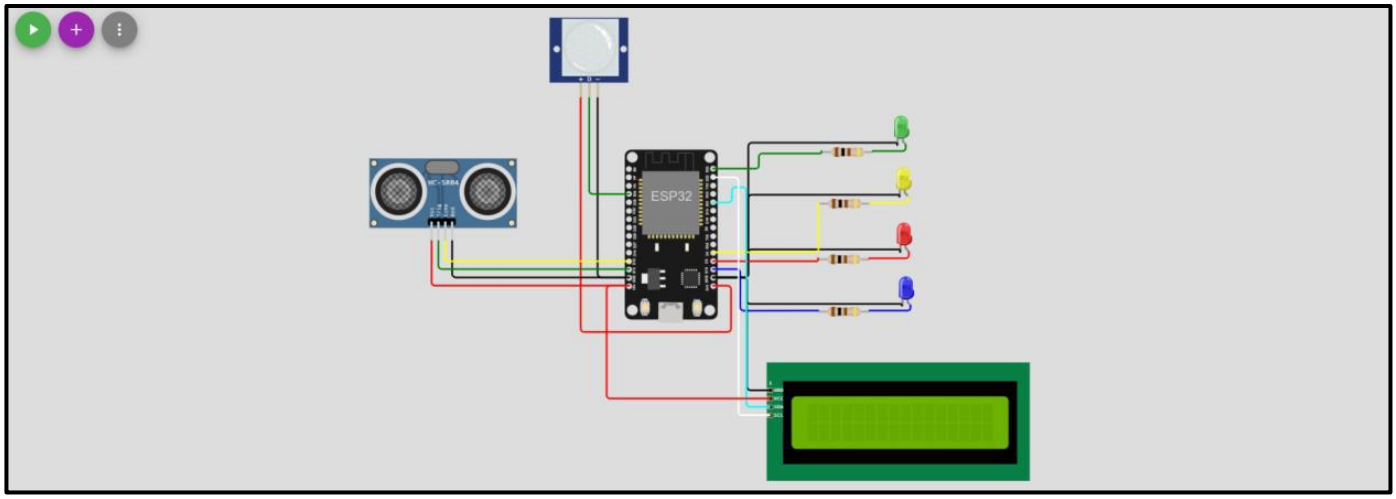
```
}
  else
  {
    digitalWrite(15, LOW);

  }


  if(cm <= 100)
{
digitalWrite(21,HIGH);
String payload = "{\"High_Alert\":";
payload += cm;
payload += " }";
Serial.print("\n");
Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str()))         // if data is uploaded
to cloud successfully,prints publish ok else prints publish failed
{
Serial.println("Publish OK");
}
}
else if(cm <= 180)
{
digitalWrite(22,HIGH);
String payload = "{\"Warning\":";
payload += cm ;
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(cm > 180)
{
digitalWrite(23,HIGH);
String payload = "{";
payload += cm;
payload += " }";
Serial.print("\n");
```

```
Serial.print("Sending payload: ");
Serial.println(payload);

if (client.publish(publishTopic, (char*) payload.c_str()))         // if data is uploaded
to cloud successfully,prints publish ok else prints publish failed
{
Serial.println("Publish OK");
}

}

  float inches = (cm / 2.54);                                      //print on lcd
  lcd.setCursor(0,0);
  lcd.print("Inches");
  lcd.setCursor(4,0);
  lcd.setCursor(12,0);
  lcd.print("cm");
  lcd.setCursor(1,1);
  lcd.print(inches, 1);
  lcd.setCursor(11,1);
  lcd.print(cm, 1);
  lcd.setCursor(14,1);
  delay(1000);
  lcd.clear();
}

//handles commands from user side

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{

  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {

    data3 += (char)payload[i];
  }
  Serial.println("data: "+ data3);

  const char *s =(char*) data3.c_str();
  double pincode = 0;


        const char *buf;
        int len;

        if (mjson_find(s, strlen(s), "$.command", &buf, &len))  // And print it
        {
```

```
            String command(buf,len);

            if(command=="\"SealBin\"")
            {
              SealBin = true;

            }


            }

  data3="";
}
```

7.2 Sensor Connection Setup

Simulation Circuit Diagram:



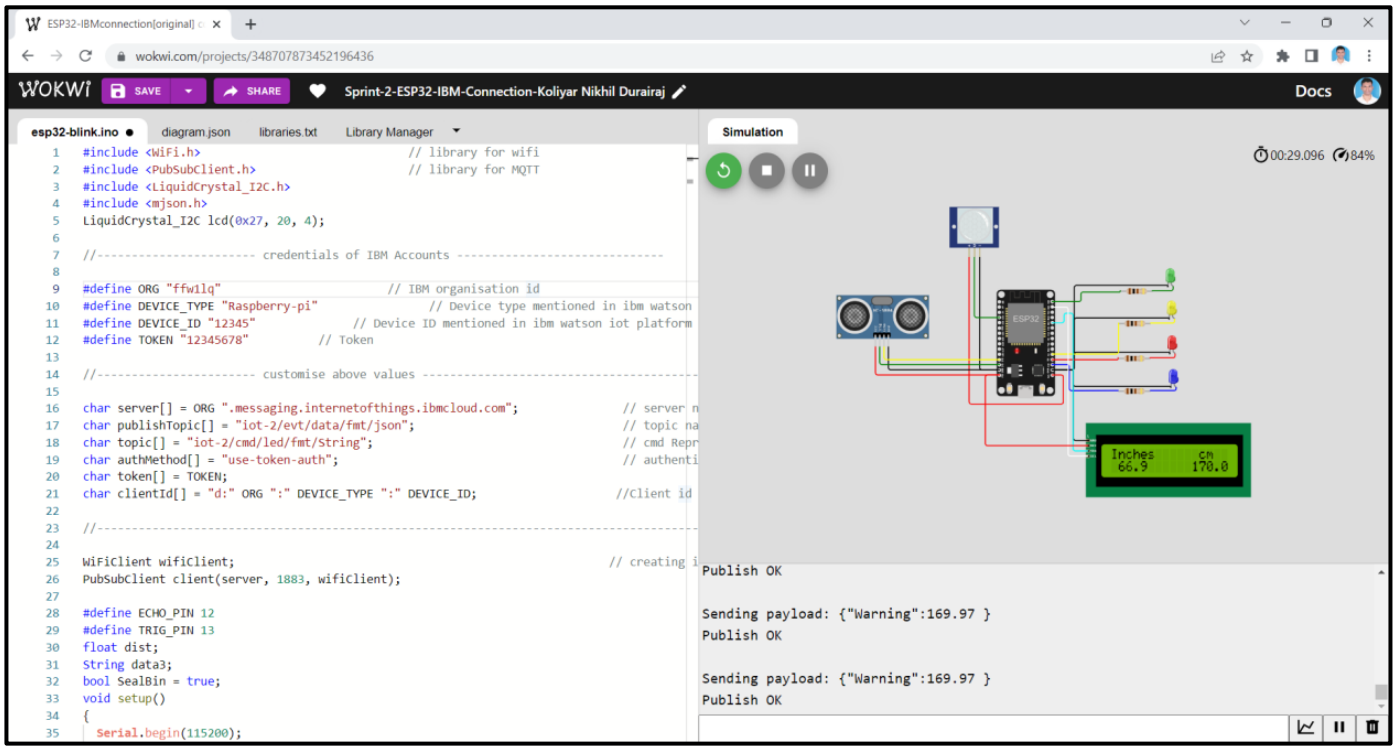
PHYSICAL COMPONENTS:

- **PIR MOTION SENSOR**
- **ULTRASONIC DISTANCE SENSOR**
- **ESP32-ARDUINO MICROCONTROLLER**

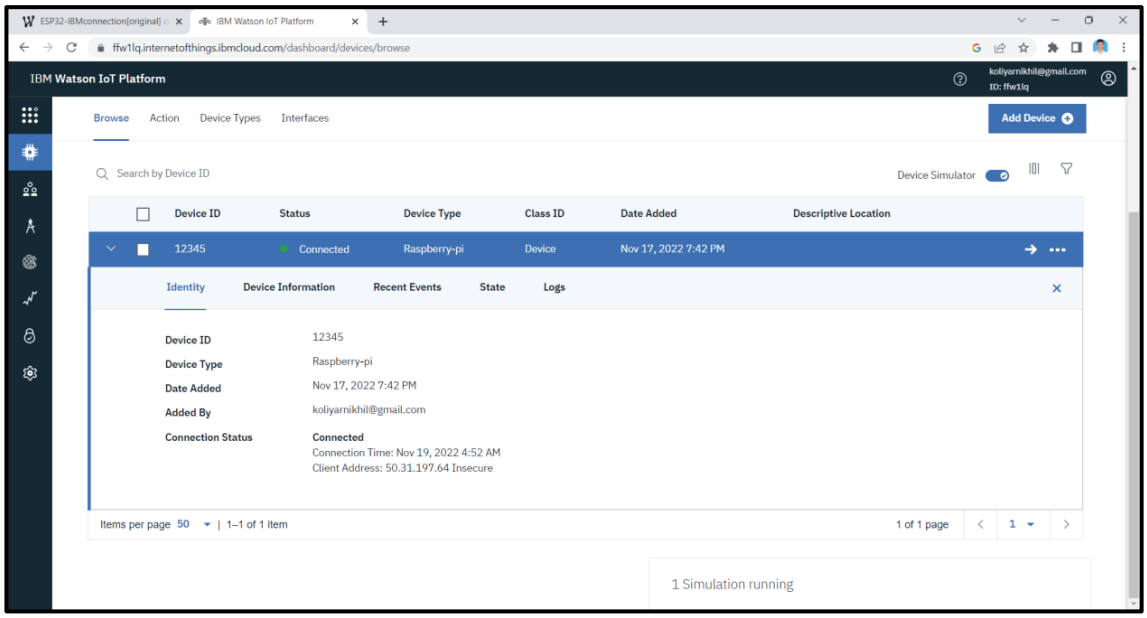
OUTPUT:

WOKWI SETUP

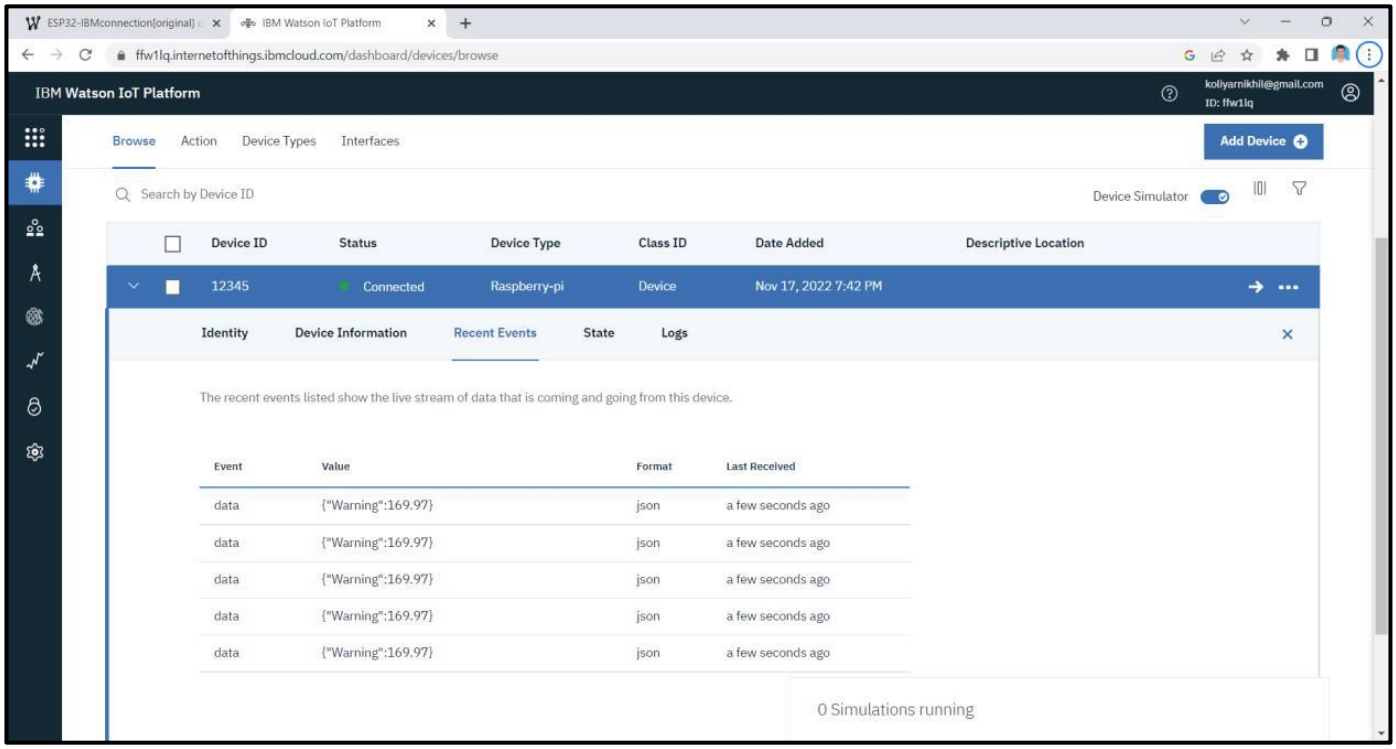
Simulation Output:



Connected with IBM Watson IoT Platform:



Sensor Data Received at IBM Watson IoT Platform:



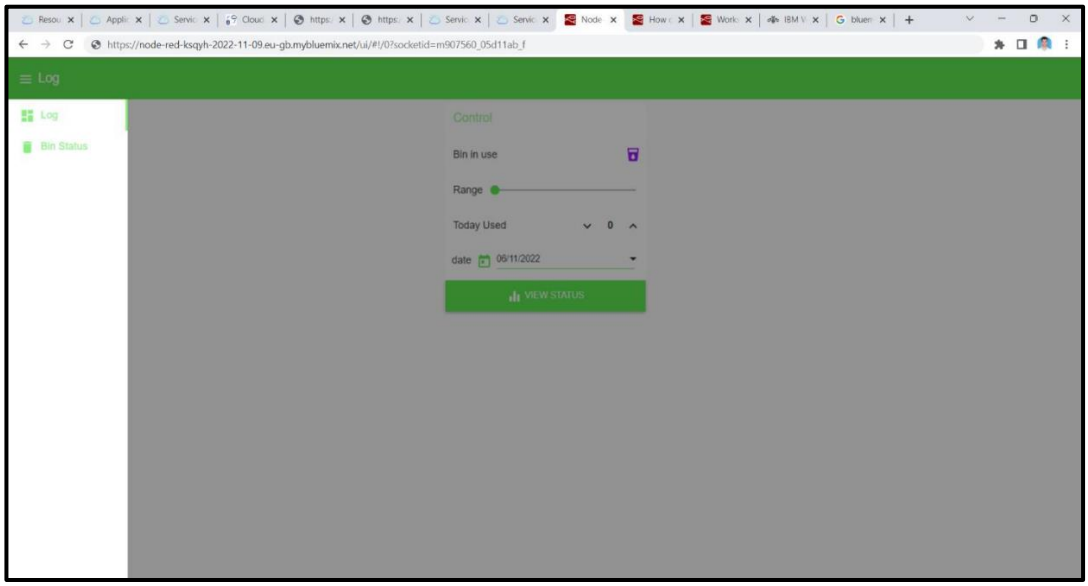
Wokwi Link:

https://wokwi.com/projects/348707873452196436

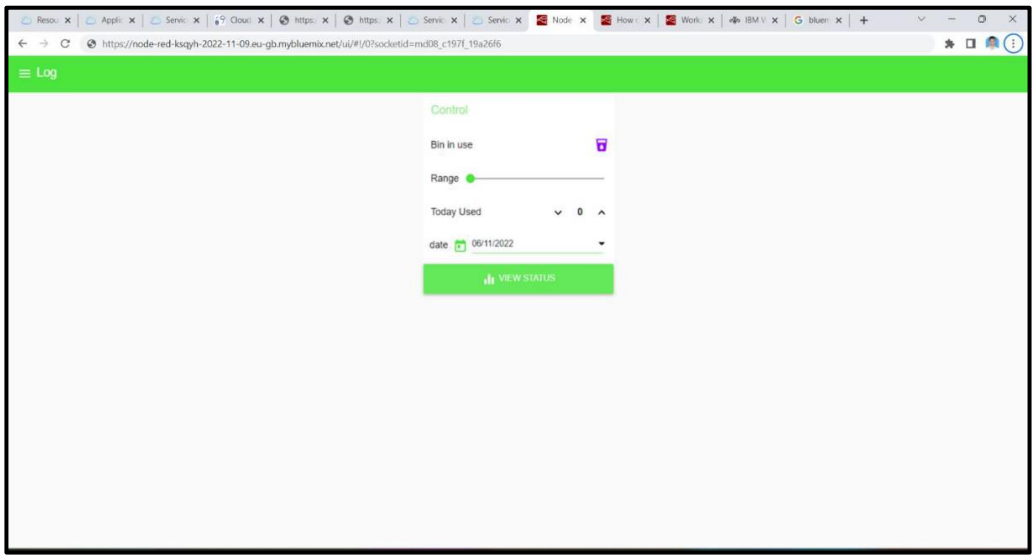
Web UI: (User Interface):

Web UI Design and Deploy:

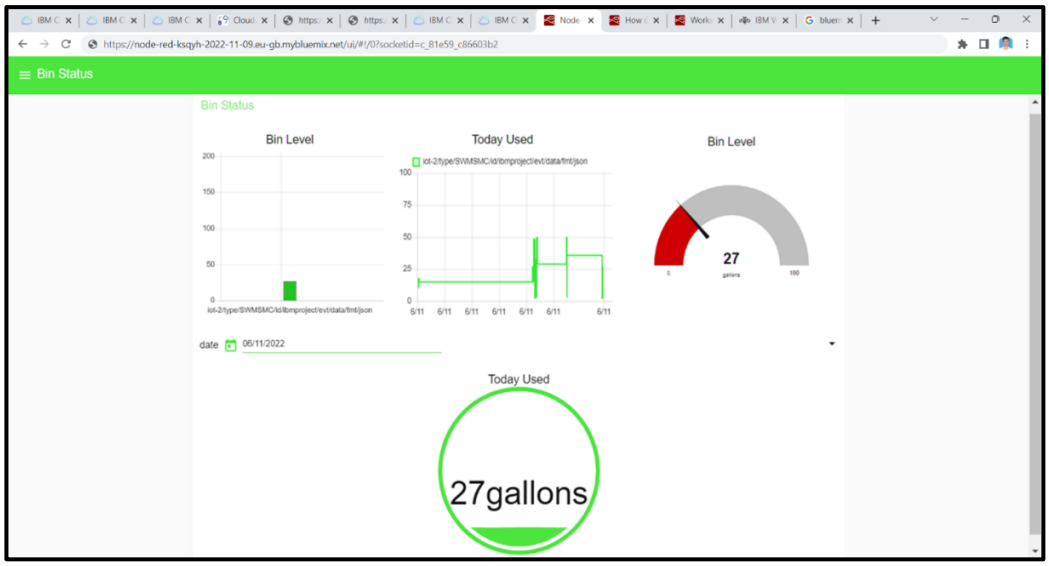
Log Menu Interface:



Control Page Interface:



Bin Status Interface:



The admin gets notification when the bin detects motion and if the bin level crosses 50 percent it indicates warning and if it crosses 90 percent it gives a High alert and closes the bin. If the admin wants to seal the bin the admin can command seal bin until it is accessed for cleaning.

Test Case:

Maximum Size of Bin: 200 cm
Minimum threshold limit of bin: 100 cm
Maximum threshold limit of bin: 180 cm
Safe limit: below 100 cm

S.no	Bin Level (cm filled)	Bin Status	Location
1	45	Safe	Kanyakumari
2	78	Safe	Coimbatore
3	112	Warning	Trichy
4	169	Warning	Chennai
5	186	Warning	Ooty
6	193	High Alert	Tirunelveli
8	0	Safe	Chengalpattu
9	35	Safe	Madurai
10	101	Warning	Salem
11	132	Warning	Thanjavore
12	158	Warning	Vellore
13	93	High Alert	Erode
14	93	High Alert	Karur
15	93	High Alert	Cuddalore
16	30	Safe	Kumbakonam
17	110	Warning	Ambur
18	180	Warning	Sivakasi
19	195	High Alert	Neyveli
20	80	Safe	Krishnagiri

Note: The bin location provided above is default. When the user access the bin , the location and status of the bin displayed to the admin.

USER ACCEPTANCE TESTING

1. Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Smart Waste Management System project at the time of the release to User Acceptance Testing (UAT).

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved.

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	3	3	20
Duplicate	1	0	3	0	4
External	2	3	0	1	6
Fixed	11	2	4	20	37
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	24	14	13	26	78

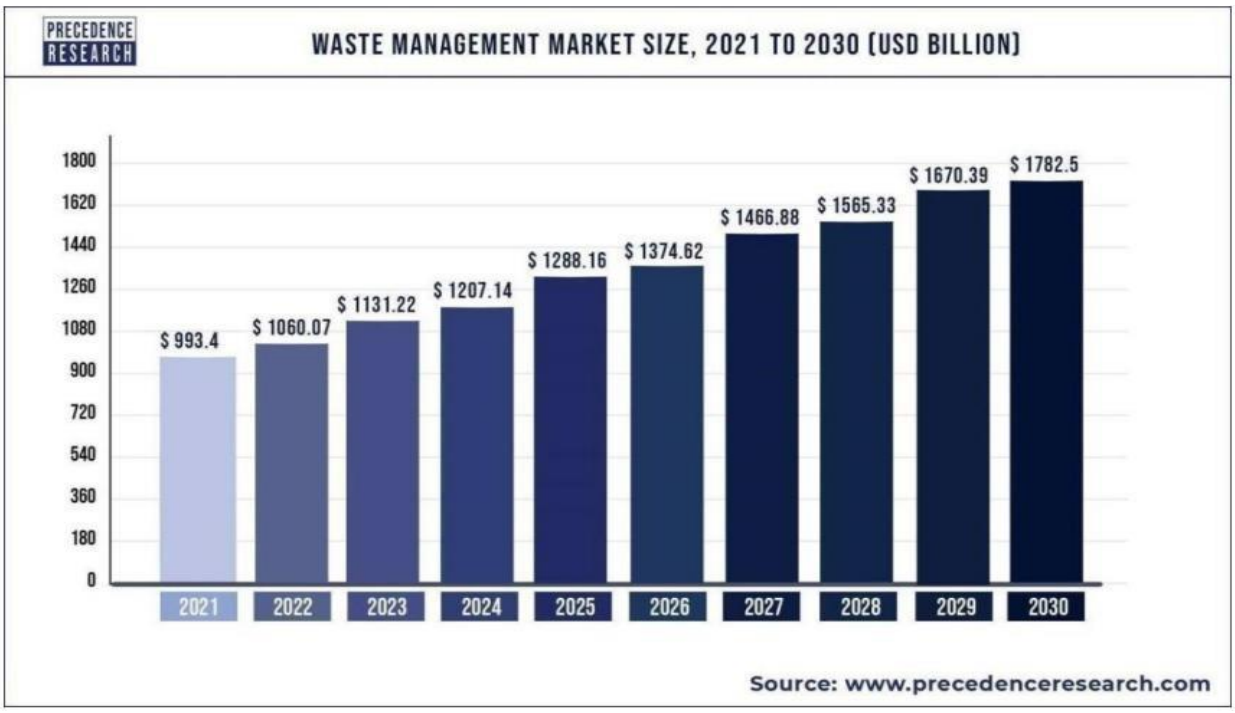
3. TEST CASE ANALYSIS

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

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51
Security	2	0	0	2
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final Report Output	4	0	0	4
Version Control	2	0	0	2

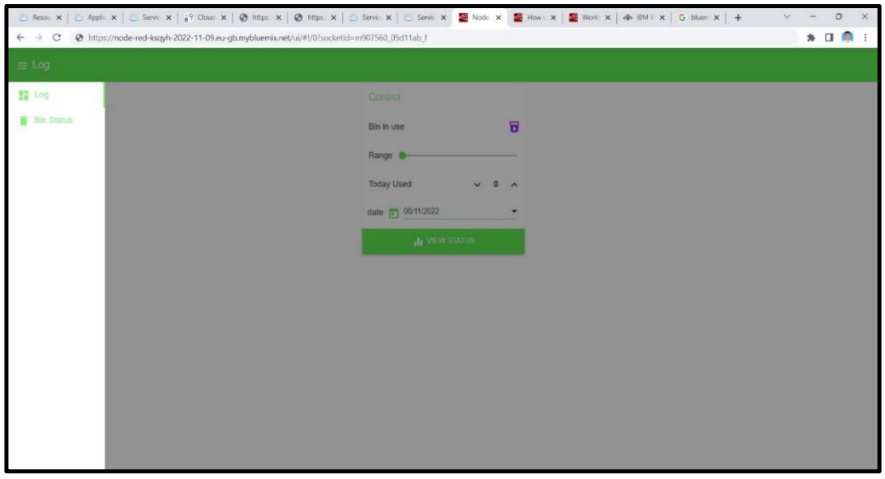
9. RESULTS

9.1. Performance Metrics

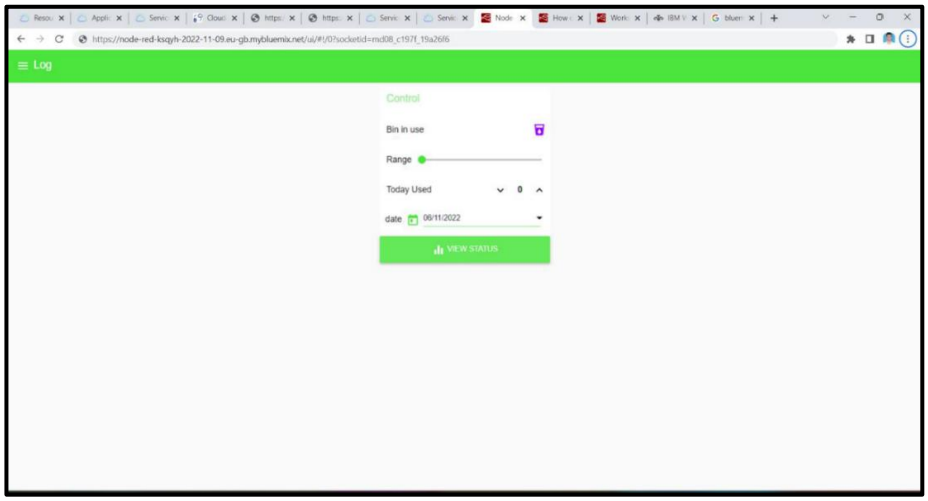


9.2 . Admin Web UI

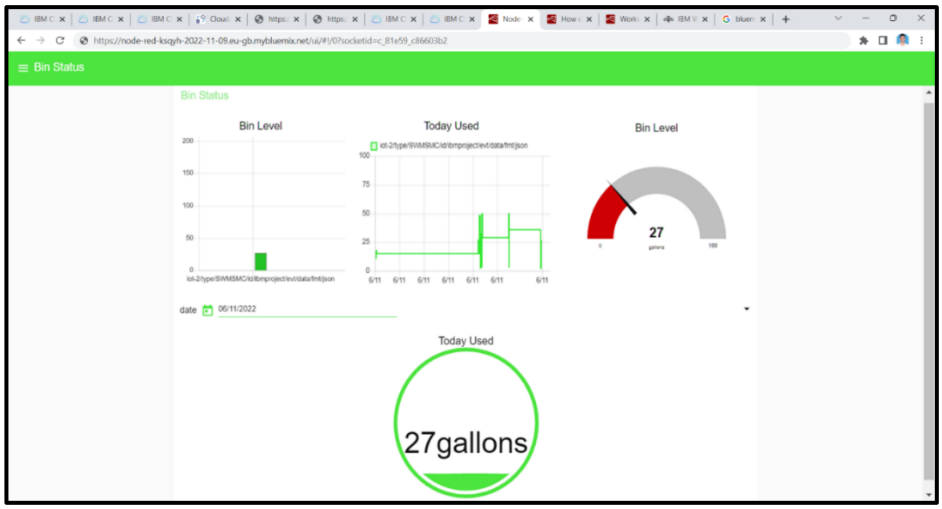
Log Menu Interface:



Control Page Interface:



Bin Status Interface:



10. ADVANTAGES AND DISADVANTAGES

10.1. ADVANTAGES

- Reduction in Collection Cost
- No Missed Pickups
- Reduced Overflows
- Waste Generation Analysis
- CO_2 Emission Reduction

10.2 DISADVANTAGES

- System requires a greater number of waste bins for separate waste collection as per population in the city.
- This results into high initial cost due to expensive smart dustbins compare to other methods. Sensor nodes used in the dustbins have limited memory size.

11. CONCLUSION:

A Smart Waste Management system that is more effective than the one in use now is achievable by using sensors to monitor the filling of bins. Our conception of a "smart waste management system" focuses on monitoring waste management, offering intelligent technology for waste systems, eliminating human intervention, minimizing human time and effort, and producing a healthy and trash- free environment. The suggested approach can be implemented in smart cities where residents 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 container would be able to hold enough solid trash for a single unit. But these may price bit high.

12. FUTURE SCOPE:

There are several future works and improvements for the proposed system, including the following:

- Changes the system of user authentication and atomic lock of bins, which would aid in protecting the bin from damage or theft.
- The concept of green points would encourage the involvement of residents or end users, making the idea successful and aiding in the achievement of collaborative waste management efforts, thus fulfilling the idea of 'Swachh Bharath'.

- Having case study or data analytics on the type and times waste is collected on different days or seasons, making the bin level predictable and remove the reliance on electronic components, and fixing the coordinates.

- Improving the Server's and Android's graphical interfaces

13. APPENDIX

- **Esp32 - Microcontroller:**

ESP32 is a series of low-cost, low-power system on a chip microcontrollers with integrated Wi-Fi and dual-mode Bluetooth.

- Memory: 320 KiB
- SRAM CPU: Tensilica Xtensa LX6 microprocessor @ 160 or 240 MHz
- Power: 3.3 V DC
- Manufacturer: Espressif Systems
- Predecessor: ESP8266

- **Sensors:**

  - PIR motion sensor: PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range.
  - Ultrasonic Distance Sensor: Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

13.1. Python Source code

```
import requests
import json
import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "ffw1lq"
devicType =  "BIN"
deviceId = "BINID"
authMethod= "token"
authToken= "12345678"

#generate random values for randomo variables (Distance and load)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org": organization, "type": devicType,"id": deviceId,"auth-
method":authMethod,"auth-token":authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
        print("caught exception connecting device %s" %str(e))
        sys.exit()

#connect and send a datapoint "temp" with value integer value into the cloud as a type of
event for every 10 seconds
deviceCli.connect()

while True:

    distance= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':distance,'load':loadcell}


    if loadcell < 13 and loadcell > 15:
        load = "90 %"
```

```
    elif loadcell < 8 and loadcell > 12:
          load = "60 %"

    elif loadcell < 4 and loadcell > 7:
          load = "40 %"
    else:
          load = "0 %"

    if distance < 15:
          dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90
%'


    elif distance < 40 and distance >16:
          dist = 'Risk warning:' 'dumpster is above 60%'

    elif distance < 60 and distance > 41:
          dist = 'Risk warning:' '40 %'
    else:
          dist = 'Risk warning:' '17 %'



    if load == "90 %" or distance == "90 %":
          warn  = 'alert :' ' Dumpster poundage getting high, Time to collect :)'

    elif load == "60 %" or distance == "60 %":

          warn = 'alert :' 'dumpster is above 60%'
    else :
          warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=10.9368,long=78.1366):
        print("Chinnavedampatti, Coimbatore, Tamil Nadu")
        print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s "
%long,"lat = %s" %lat)
        print(load)
        print(dist)
        print(warn)

    time.sleep(10)


    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)

    success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish=
myOnPublishCallback)
```

```
    if not success:
        print("not connected to ibmiot")
    time.sleep(30)



    deviceCli.commandCallback=myCommandCallback
#disconnect the device
deviceCli.disconnect()
```

Python Code Output

```
Chinnavedampatti, Coimbatore, Tamil Nadu
published Level of bin = 20% Load = 60% Latitude = 11.0779 Longitude = 76.9902
60%
20%
alert:Dustbin is almost filled
Chinnavedampatti, Coimbatore, Tamil Nadu
published Level of bin = 20% Load = 60% Latitude = 11.0779 Longitude = 76.9902
60%
20%
alert:Dustbin is almost filled
Chinnavedampatti, Coimbatore, Tamil Nadu
published Level of bin = 20% Load = 60% Latitude = 11.0779 Longitude = 76.9902
60%
20%
alert:Dustbin is almost filled
Chinnavedampatti, Coimbatore, Tamil Nadu
published Level of bin = 20% Load = 60% Latitude = 11.0779 Longitude = 76.9902
60%
20%
alert:Dustbin is almost filled
Chinnavedampatti, Coimbatore, Tamil Nadu
published Level of bin = 20% Load = 60% Latitude = 11.0779 Longitude = 76.9902
60%
20%
alert:Dustbin is almost filled
```

13.2 . GITHUB LINK

LINK: **https://github.com/IBM-EPBL/IBM-Project-8111-1658909581**