CHAPTER - 1

INTRODUCTION

1.1 Project Overview

Iot-based smart waste management system helps in monitoring the fill level and collection of bins in a efficient way by placing a ultrasonic sensor and a gps tracker in every dumpster in town these sensors are integrated with a mobile application where the weight of the dumpsters are updated to the municipal corporation who is responsible for that particular area.The employee can plan his collection operation accordingly and the gps sensor helps him plan the collection making it more efficient.

1.2 Purpose :

Municipal Solid Waste Management is of critical concern and needs attention.The rapid urbanisation and industrialization has led to increased solid waste generation, about 2.1 billion tonnes of municipal solid waste is generated annually around the globe. So the traditional methods of waste collection have become inefficient and costly.The main objective is to maximise waste collection and optimise the work for municipal corporations. In order to overcome the disadvantages of traditional systems ,technology paves the way.Creates a clean as well as green environment. Less amount of fuel consumed by vehicles can save a large amount of money.It will stop overflowing of dustbins along roadsides and localities. The filling and cleaning time of smart bins will be reduced thus making empty and clean dustbins available to common people. Employment of health workers remains while more employment opportunities for technical personnel increases.

CHAPTER - 2

LITERATURE SURVEY

2.1 Existing problem :

● Inefficient way to identify the fill level of the bins for collection
● There is Fixed routine for waste collection regardless of the fill levels of the trashcan
● Wastage of resources
● Missed pick-ups,causing unclean environment

2.2 References :

- Zhang, A., Venkatesh, V. G., Liu, Y., Wan, M., Qu, T., & Huisingh, D. (2019). Barriers to smart waste management for a circular economy in China. *Journal of Cleaner Production*, *240*, 118198.

- Samih, H. (2019). Smart cities and internet of things. *Journal of Information Technology Case and Application Research*, *21*(1), 3-12.

- Abdullah, N., Alwesabi, O. A., & Abdullah, R. (2019). IoT-based smart waste management system in a smart city. Advances in Intelligent Systems and Computing, 843, 364–371. https://doi.org/10.1007/978-3-319- 99007-1_35

- Soh, Z. H. C., Husa, M. A. A. H., Abdullah, S. A. C., & Shafie, M. A. (2019, April). Smart waste collection monitoring and alert system via IoT.

- Aazam, M., St-Hilaire, M., Lung, C. H., Lambadaris, I., (2016). Cloud-based smart waste management for smart cities. 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD). Toronto, ON, Canada. https://doi.org/10.1109/CAMAD.2016.7790356

2.3 Problem statement definition

Municipal Solid Waste Management is of critical concern and needs attention.The rapid urbanisation and industrialization has led to increased solid waste generation, about 2.1 billion tonnes of municipal solid waste is generated annually around the globe.The Municipal trash bins and landfills packed like sardines are creating serious pollution and health hazards such as dysentery, diarrhoea and amoebic dysentery, plague, salmonellosis, trichinosis etc.Major challenges include not being able to monitor municipal trash cans in every community and, in the worst case scenario, not being able to reach rural locations. As a result, the old ways of collecting waste are now both expensive and ineffective.

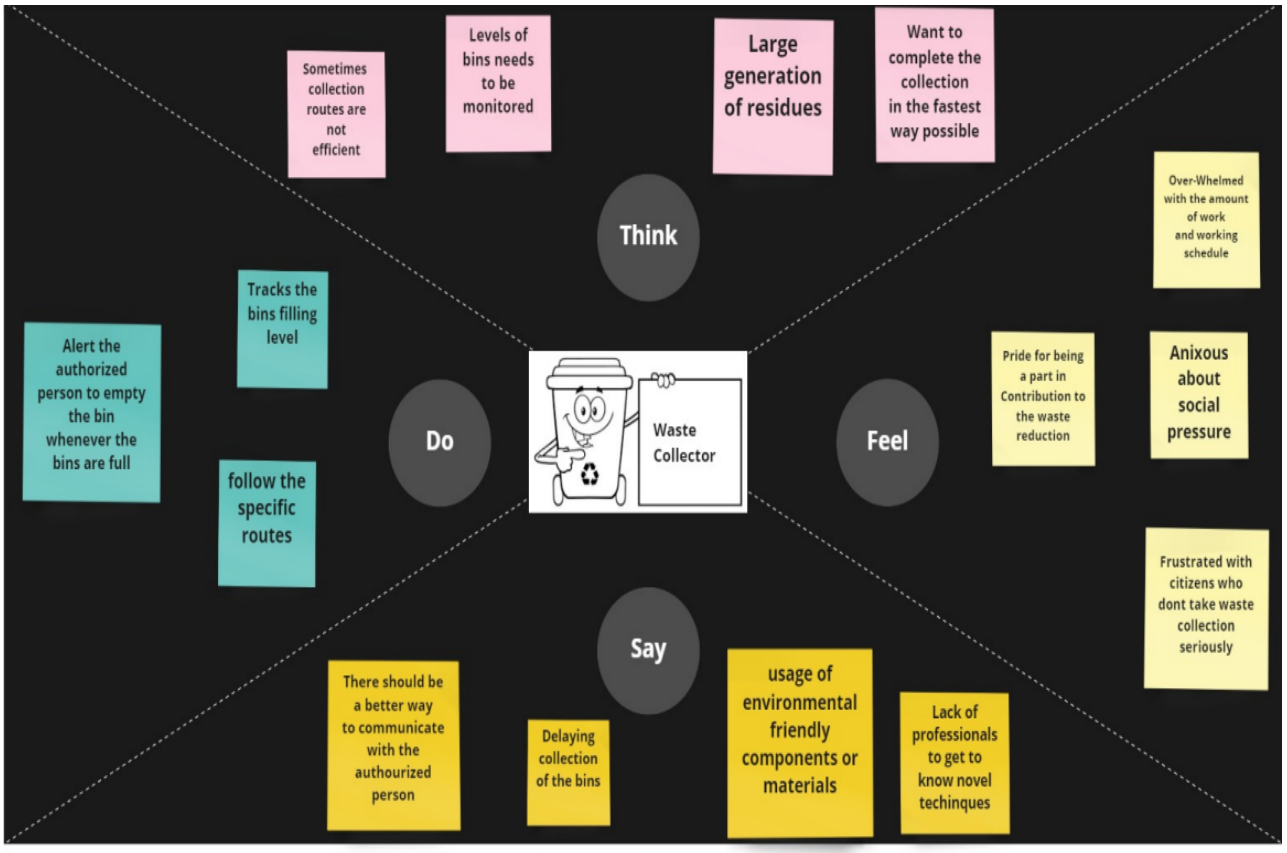
2.4 Idea Description

Solid waste collection is a great challenge for our modern society. So to overcome this challenge we propose a real time waste management system that uses bins integrated with sensors which will provide the real time information about the fill level of the bins and air quality of the surroundings to the admin through an application. The concerned employee can check the status of the bins and plan his collection operation accordingly and results are notified to the municipal corporation if the bin value crosses the threshold value. The system also provides route optimization for the collection of bins. The weight sensor computes the weight of the bin every 12 hours and the air quality sensor detects if the air is contaminated and updates the information to the server and gps sensor provides location of the bins.

CHAPTER - 3

IDEATION AND PROPOSED SOLUTION

3.1 Empathy Map Canvas :



Pain

- People not understanding the importance of waste mangement
- Not knowing which conatiners are empty and full
- Not being sure about the time taken
- Need for skilled professionals and usage of techniques

Gain

- More efficient collection system by using new technologies and techniques
- Help reduce waste and pollution
- Greener technology

3.2 Ideation and Brainstorming :

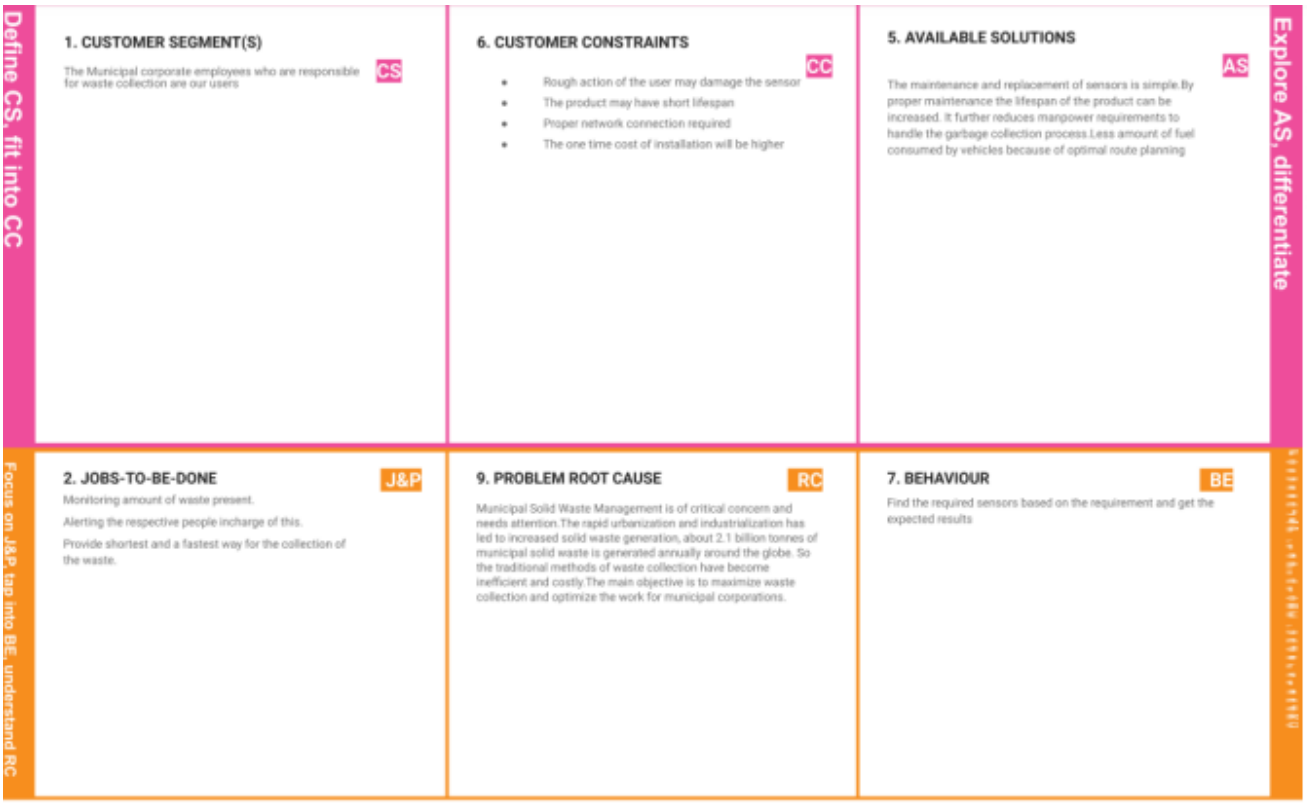
IoT smart waste management products are designed to help monitor the amount of waste using sensors and send those data to the waste collector . As a result, trash collectors can easily monitor the waste from anywhere without any hassle.

3.3 Proposed Solution :

S.No.	Parameter	Description
1	Problem Statement	Municipal Solid Waste Management is of critical concern and needs attention.Major challenges include not being able to monitor municipal trash cans in every community and, in the worst case scenario, not being able to reach rural locations. As a result, the old ways of collecting waste are now both expensive and ineffective
2.	Idea / Solution description	A dynamic waste management system that uses bins integrated with sensors which will provide the real time information about the fill level of the bins and air quality of the surroundings of bins to the admin through an application. The concerned employee can check the status of the bins and plan his collection operation accordingly and results are notified to the municipal corporation if the bin value crosses the threshold value. The system also provides route optimization for the collection of bins.

3.	Novelty / Uniqueness	Shortest path is discovered using the route algorithm.The cleanliness of the air is also evaluated around the trash cans using a quality sensor.It is easy to install in any type of container Live dashboard which displays the real time fill level of garbage bins.
4.	Social Impact / Customer Satisfaction	It creates a clean as well as green environment. It consumes less amount of fuel which can save money. It also stops the overflowing of dustbins along roadsides and localities. It increases the employment opportunities for technical personnel.
5.	Business Model (Revenue Model)	The sensors can be monitored remotely using a software which tracks the data real time and provides statistics of the usage. Alerts are triggered to respective persons when needed. The state of the dustbin can be viewed through a dashboard with various details about the events. The shortest route for the collection of waste is provided and can be viewed at real-time.
6.	Scalability of the Solution	The components used for the building up the dustbin is cheap and the solution is effective as the components are easily available. The sensors in the dustbin collects the data and sends it to the cloud. Node Red makes the runtime environment scalable and supports a bunch of users to access at the same time and IBM Cloud supports thousands of users to access the cloud simultaneously. The system is capable of handling multiple requests and handles data without any flaw. Thus sensors can be handled and viewed remotely there is a vast growth in our product that will be scalable and useful.

3.4 Problem Solution Fit :



CHAPTER - 4

REQUIREMENT ANALYSIS

4.1 Functional Requirements :

FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub Task)
FR-1	Detailed bin inventory	The IoT device consists of Ultrasonic sensor, IR sensor, Weight sensor. To send data to the cloud GPRS/GSM is used. The Dashboard displays information about the bin, including its capacity, most recent measurement, GPS location, and collection schedule.
FR-2	Real time bin monitoring.	Retrieving information from database for calculation garbage bin which fulfils the condition for garbage collection, example: Collect garbage from bins whose level is over 80% of bin. Garbage collection live monitoring is done using a client side script
FR-3	Adjust bin distribution.	Determine which regions have a dense or sparse distribution of bins. You can make any necessary adjustments to bin position or capacity based on past data.
FR-4	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.
FR-5	Plan waste collection routes.	Route planning for waste pickup is semi-automated using the tool. It is prepared to act and arrange for garbage collection based on the level of bin filling at the moment and projections of when capacity will be reached.

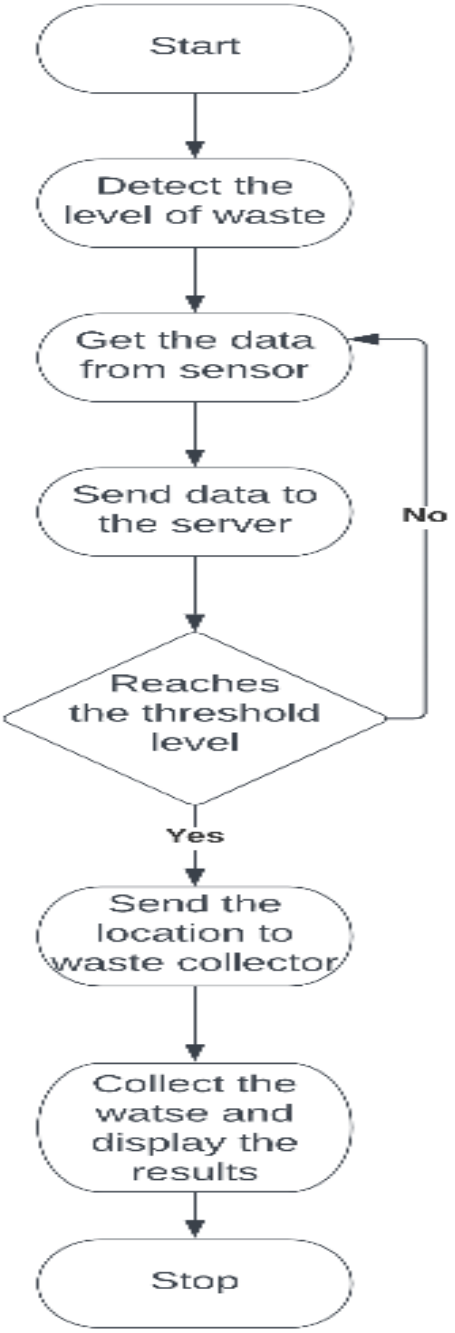
4.2 Non-Functional Requirements:

FR	Non-Functional Requirement	Description
NFR-1	Usability	Usability includes easy learning ability, efficiency in use,memory ability, lack of errors in operation and subjective pleasure.
NFR-2	Security	Sensitive and private data must be protected from their production until the decision-making and storage stages.
NFR-3	Reliability	The shared protection achieves a better trade-off between costs and reliability. The model uses dedicated and shared protection schemes to avoid farm service outages.
NFR-4	Performance	The idea of implementing integrated sensors with sensing the amount of waste and environmental or ambient parameters in farming will be more efficient for overall monitoring.
NFR-5	Availability	Automatic adjustment of farming equipment made possible by linking information like crops/weather and equipment to auto-adjust temperature, humidity, etc.
NFR-6	Scalability	Scalability is a major concern for IoT platforms. It has shown that different architectural choices of IoTplatforms affect system scalability and that automatic real time decision-making is feasible in an environment composed of dozens of thousands.

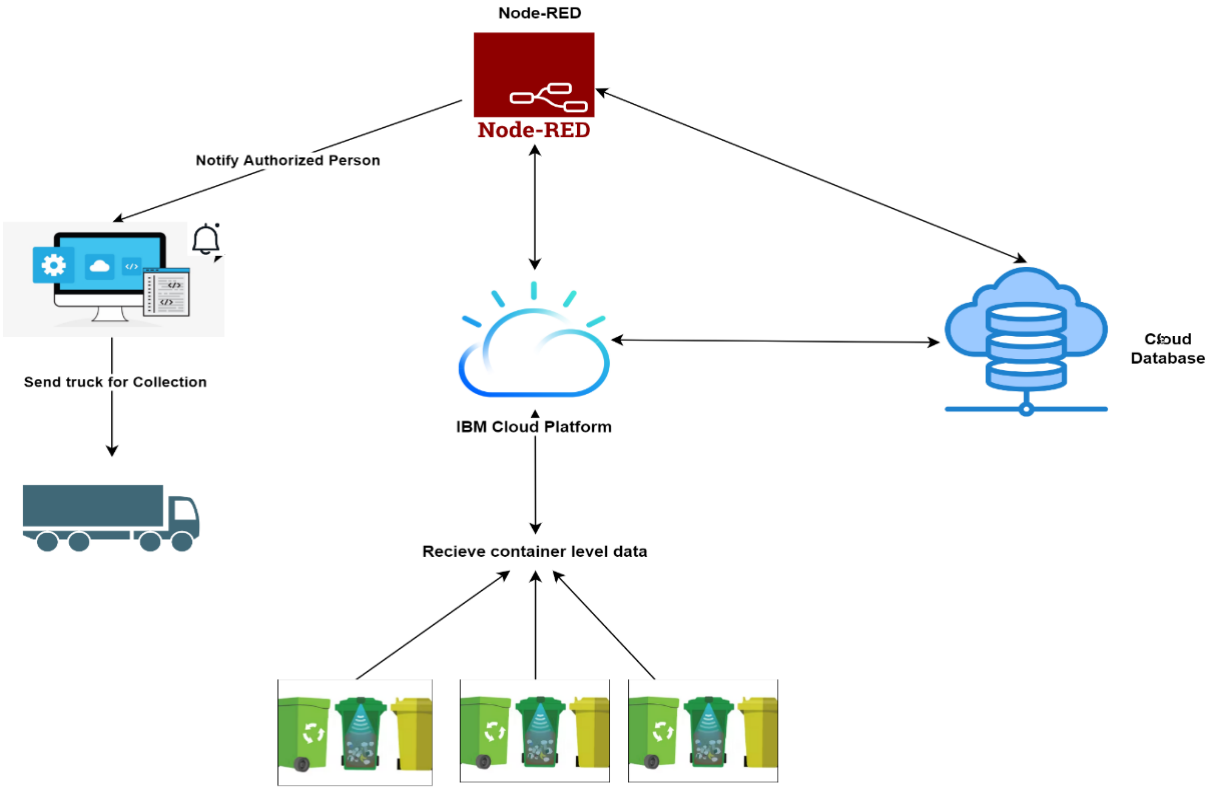
CHAPTER - 5

PROJECT DESIGN

5.1 Data Flow Diagrams :



5.2 Solution and Technical Architecture :



5.3 Circuit Connections :

5. User Stories :

UserStories

User Type	Functional Requirement (Epic)	User Story number	User Story/Task	Acceptance Criteria	Priority	Release
Customer (Mobile user)	Registration	1	Can register for the application by entering my email,password,and confirming my password.	Can Access my account / dashboard	High	Sprint-1
		2	Will receive confirmation email once I have registered for the application	Receive Confirmation email and click confirm	High	Sprint-1
		3	Can register for the application Through Facebook	Can register & access the dashboard with Facebook Login	Low	Sprint-2
		4	Can Register for the application through Gmail		Medium	Sprint-1
	Login	5	Can Login to the application by entering email & password.		High	Sprint-1

CHAPTER - 6

PROJECT PLANNING AND SCHEDULING

6.1 Sprint Planning and Estimation :

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password and login with the registered mail and password.	2	High	Daadreyaa D Geethanjali M Deepasree R
Sprint-1	Web UI for monitoring bin level	USN-2	As a user, I can see a dashboard that displays the bin level.	1	Low	Deepasree R Aruna S
Sprint-2	Garbage level detection in bins	USN-3	As a user, I can see the weight of the bins from the data collected from a weight sensor.	2	High	Geethanjali M Deepasree R Aruna S
Sprint-2	Sending alert messages to the authorized person	USN-4	As a user, I can receive alerts when the bin is full.	1	Low	Deepasree R Daadreyaa D
Sprint-3	GPS location tracking of the bins	USN-5	As a user, I can track the location of the bins.	2	Medium	Aruna S Geethanjali M
Sprint-4	Shortest Route Planning for garbage collection	USN-6	As a user, I can take the shortest path to collect the garbage.	1	Medium	Daadreya D Geethanjali M

6.2 Sprint Delivery Schedule :

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	31 Oct 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	07 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	14 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)

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

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

CHAPTER - 7

CODING AND SOLUTIONING

7.1 Feature 1 (Arduino code):

```
#include <WiFi.h>
#include <PubSubClient.h>
#define echo 5
#define trigger 18

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

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

#define ORG "rz7zse"//IBM ORGANITION ID
#define DEVICE_TYPE "sampleDevice"//Device type mentioned in ibm watson IOT Platform
#define DEVICE_ID "sampleId"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "sampleToken"     //Token


//-------- Customise the 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 subscribetopic[] = "iot-2/cmd/test/fmt/String";// cmd  REPRESENT command type AND COMMAND IS TEST OF FORMAT STRING
char authMethod[] = "use-token-auth";// authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;//client id


//---------------------------------------
```

```
WiFiClient wifiClient; // creating the instance for wificlient
PubSubClient client(server, 1883, callback ,wifiClient); //calling the predefined client id by passing parameter like server id,portand wificredential


void setup()// configureing the ESP32
{

  Serial.begin(115200);
  delay(10);
  Serial.println();
  wificonnect();
  mqttconnect();
  pinMode(echo, INPUT);
  pinMode(trigger, OUTPUT);

}


void loop()// Recursive Function
{

  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger, LOW);
  float duration = pulseIn(echo, HIGH);
  float distance = duration * 0.0343 / 2;
  PublishData(distance);
  delay(1000);
  if (!client.loop())
{
    mqttconnect();

  }
```

```
 delay(500);
}



/*.....................................retrieving to Cloud.....................*/


void PublishData(float distance) {
  mqttconnect();//function call for connecting to ibm
  /*
     creating the String in in form JSon to update the data to ibm cloud
  */
  String payload = "{\"distance\":";
  payload += distance;
  payload += "}";

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


  if (client.publish(publishTopic, (char*) payload.c_str())) {
    Serial.println("Publish ok");// if it sucessfully upload data on the cloud then it will
print publish ok in Serial monitor or else it will print publish failed
  }

 else

{
    Serial.println("Publish failed");

  }

}
```

```
void mqttconnect()
{

  if (!client.connected())
{

    Serial.print("Reconnecting client to ");
    Serial.println(server);
    while (!!!client.connect(clientId, authMethod, token))

 {
      Serial.print(".");
      delay(500);

    }

     initManagedDevice();
     Serial.println();

}

}
void wificonnect() //function defination for wificonnect

{

  Serial.println();
  Serial.print("Connecting to ");

  WiFi.begin("Wokwi-GUEST", "", 6);//passing the wifi credentials to establish the
connection
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
```

```
}

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

}


void initManagedDevice() {

  if (client.subscribe(subscribetopic)) {

    Serial.println((subscribetopic));
    Serial.println("subscribe to cmd OK");
  } else
{

    Serial.println("subscribe to cmd FAILED");
  }
}

void callback(char* subscribetopic, byte* payload, unsigned int payloadLength)
{
  String data = "";
  Serial.print("callback invoked for topic: ");
  Serial.println(subscribetopic);
  for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data += (char)payload[i];
}

  Serial.println(data);

}
```

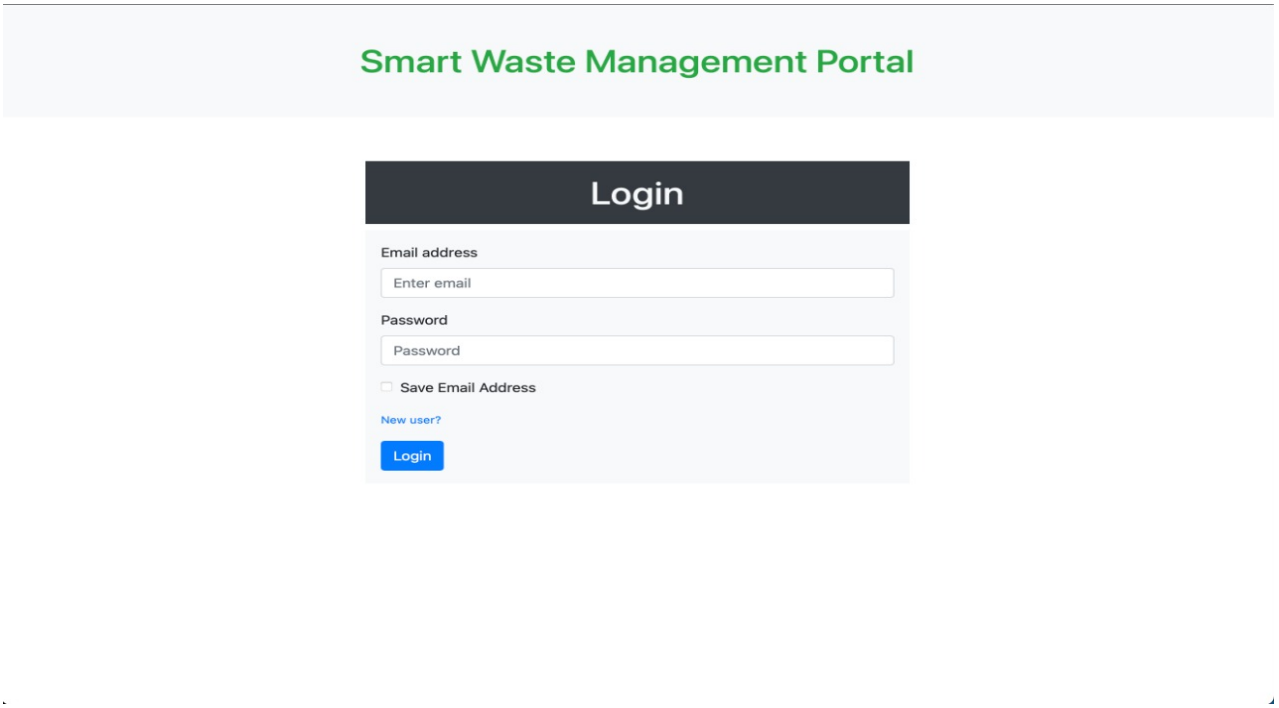
7.2 Feature 2(Python stimulation):

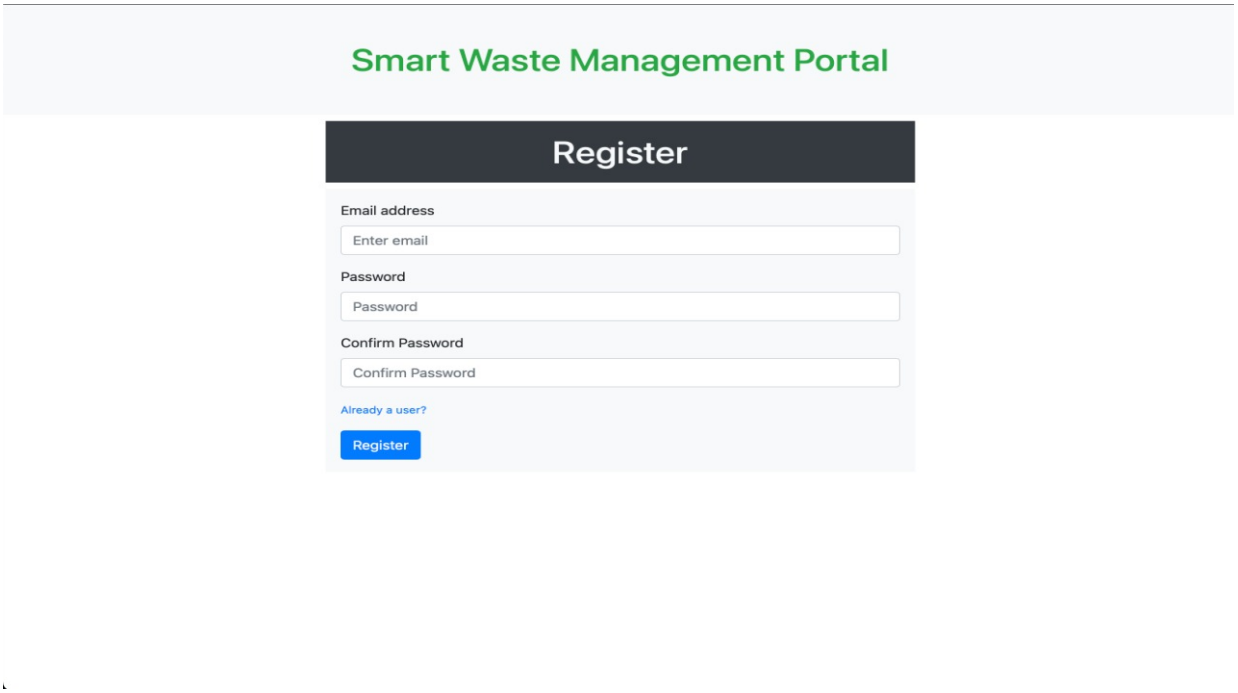
```
# Bin 1

import time
import sys
import random
from database import db
import ibmiotf.application
import ibmiotf.device

organization = "rz7zse"
deviceType = "BIN"
deviceId = "bin1"
authMethod = "token"
authToken = "bin1token"


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data)
    print(cmd)


def start():
    try:
        deviceOptions = {"org": organization, "type": deviceType,
                         "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)

    except Exception as e:
        print("Caught exception connecting device", e)
        sys.exit()

    deviceCli.connect()

    while True:
```

```
        distance = random.randint(0, 400)
        weight = random.randint(0, 250)
        lat = 13.1
        long = 80.1

        data = {"data": {"distance": distance, "weight": weight,
                    "latitude": lat, "longitude": long}}

        def myOnPublishCallback():
            print("Published Distance =", distance,
                  "Weight =",  weight, "to IBM Watson")

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

        db.child("bin").child("bin1").set(data)

        if not success:
            print("Not connected to IOTF")

        time.sleep(1)
        deviceCli.commandCallback = myCommandCallback

    deviceCli.disconnect()


start()
```

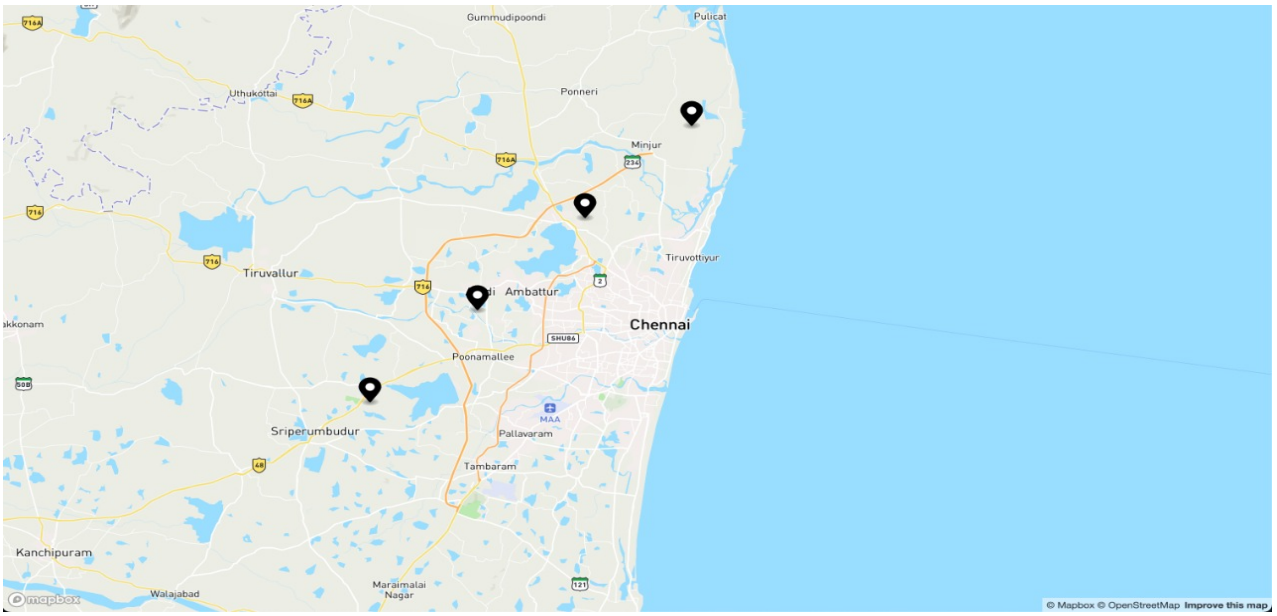
Output:

Login and register:

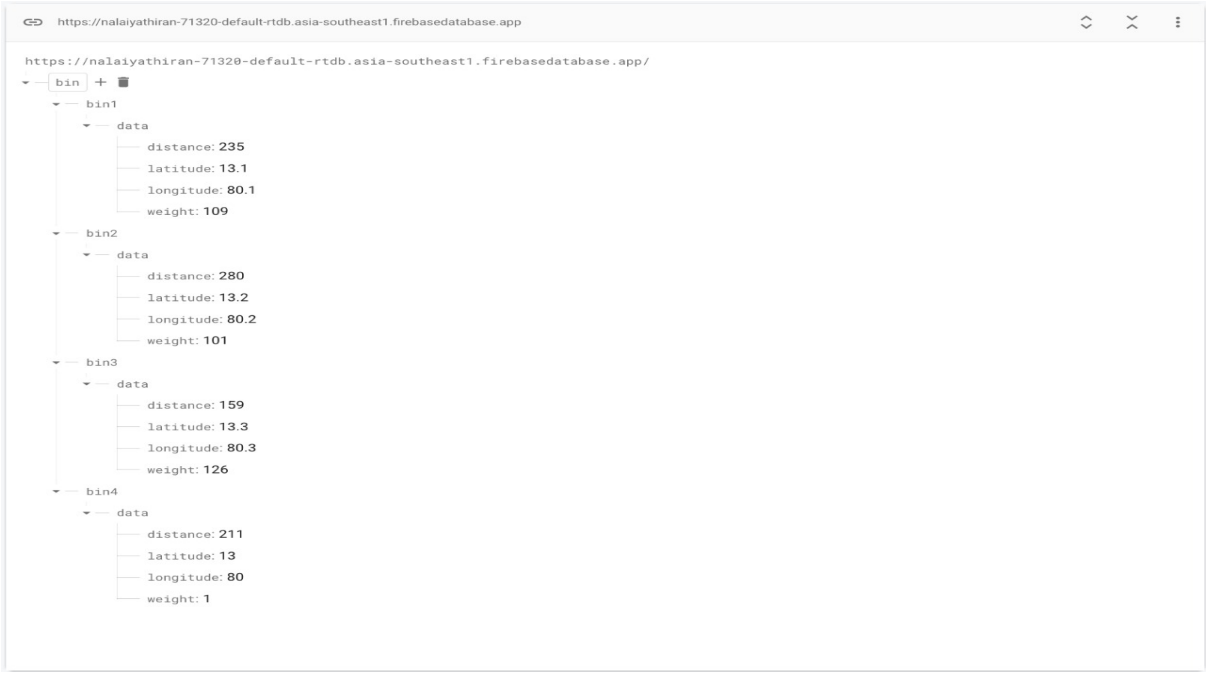




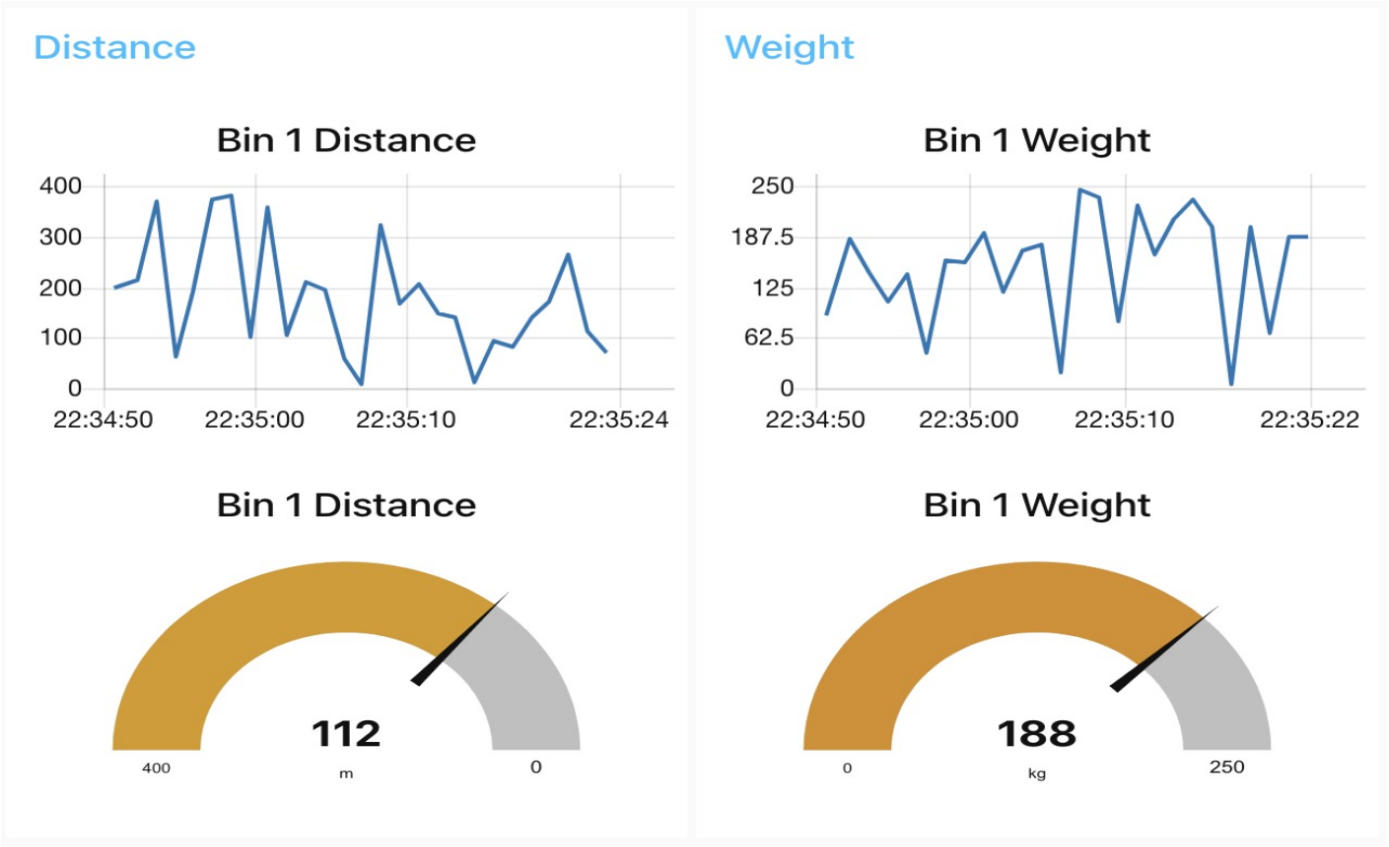
Map Plot Of Bin locations:

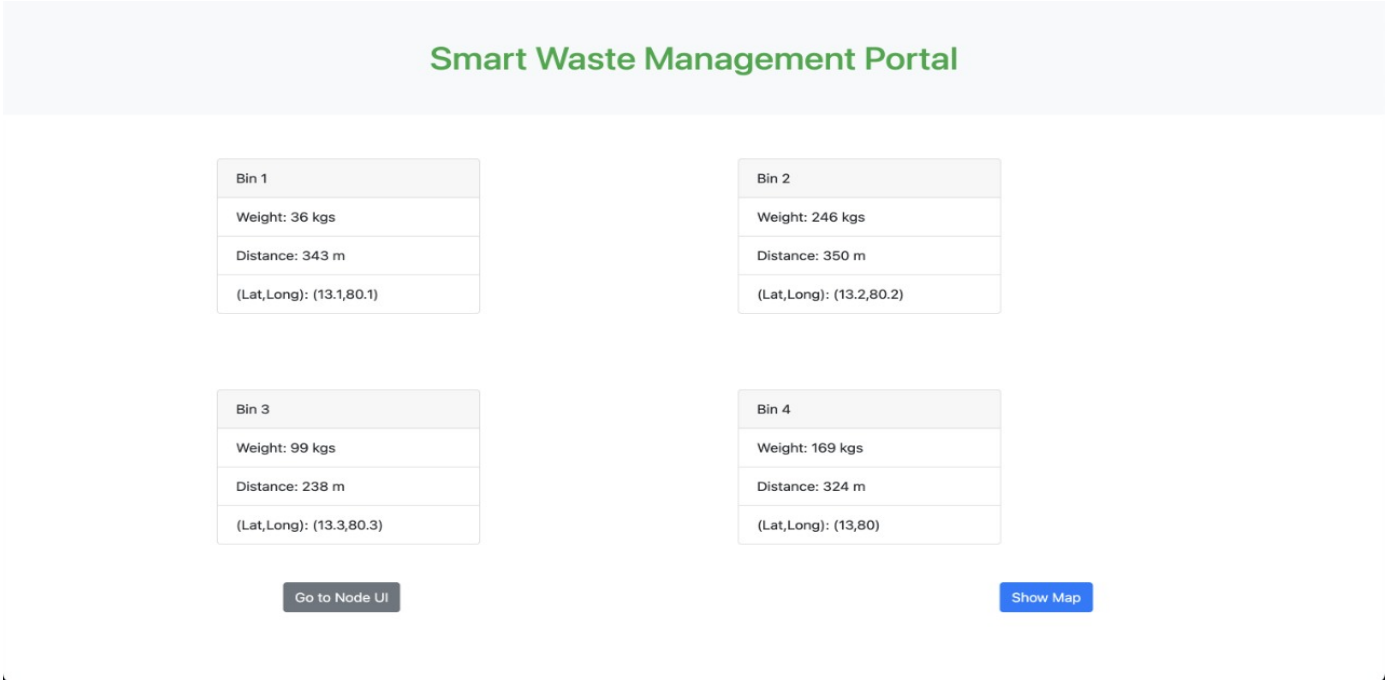


Firebase Realtime Database:



Bin level Monitoring:





CHAPTER - 8

TESTING

8.1 Test Cases :

- Check the working condition of the sensors
- Check the Performance of the app on the different internet networks
- Check whether the application is working for real-time updates
- Check response when a Login Button is pressed
- Examine the usability of the user interface.

CHAPTER - 9

ADVANTAGES AND DISADVANTAGES

Advantages

- The bins can be monitored and controlled remotely.
- Increase in convenience to corporation workers.
- Less transport and fuel cost.
- The filling and cleaning time of smart bin will also be reduced thus making empty and clean dustbins available to common people.
- Clean and green society.
- Collection operations become more efficient and Smarter.
- Easy to install to any type of containers

Disadvantages

- Lack of internet connectivity may cause issues.
- Added cost of internet and internet gateway infrastructure.
- The corporate workers wanted to adapt the use of Mobile App.

CHAPTER - 10

CONCLUSION

IoT will help to enhance smart waste management.Through this system, the information of all smart trash cans can be viewed by the person who is concerned at any time and from any location. Smart bins are used to monitor the fill level and to determine whether they are full or not. In order for the responsible authority to dispatch the garbage collection vehicle only when the dustbin is full, it will provide real-time information on the status of each and every dustbin.This Iot smart waste management system is used to identify the status of waste bins if it is empty or filled so as to customise the waste collection schedule accordingly and also save the cost of transport and fuel and build a clean and green society.

Github Link:
https://github.com/IBM-EPBL/IBM-Project-18188-1659680622

Project Demo Link:
https://github.com/IBM-EPBL/IBM-Project-7031-1658845516/tree/main/Final%20Deliverables/Demo%20Video