SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITAN CITIES

NALAIYA THIRAN PROJECT BASED LEARNING

ON

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

A PROJECT REPORT Submitted by

S ABDUL MAJITH - 911519205001

V BALAJI -911519205005

A DHAHA MOHAMED - 911519205006

A S MOHAMED ADHIL HUSSAIN - 911519205012

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

MOHAMED SATHAK ENGINEERING COLLEGE

KILAKARAI- 623 806

ANNA UNIVERSITY::CHENNAI 600 025

NOVEMBER 2022

Project Report

1.INTRODUCTION

1.1 Project Overview

One of the most significant issues associated with smart city applications is solid waste management, which has a negative impact on our society's health and the environment. Mainly there are three types of sources where garbage is generated viz. residential, commercial and industrial. The garbage produced in the residential area can be collected directly from home or by making an arrangement for mass collection in that area and can be lifted using vehicles. In case of restaurants, malls and other commercial establishment garbage can be collected directly from the unit using vehicles. To overcome this situation an efficient Smart Waste Management System for Metropolitan Cities has to be developed using Internet of Things. Internet of Things (IOT) can be used effectively to manage this waste as many effective methods can be found out easily.

1.2 Purpose

The Purpose of this Project to Detect the Garbage Level in SmartBin and makes the Alert when Garbage is Full. This Action might help to the user who can use the SmartBin. The App Might show the location of the Bin and where the user can know the status of bin and use the Bin in the Right time.

2. LITERATURE SURVEY

2.1 Existing problem

The garbage management in cities should be effectively and efficiently implemented. various proposals were put forward and some of them are already implemented. But we cannot consider it as an effective one. So, a survey was done among different proposals and this survey paper includes survey among different methods for smart garbage management in cities using IoT. This section discusses about the existing approaches in the field of smart waste management.

2.2 References

Insung Hong et.al [1] has suggested that replacing SGS (Smart Garbage Sensor) instead of RFID garbage collecting system helps to improve their energy efficiency up to 16% and can reduce the food waste reduction .Inside the SGS they have installed SGBs (Smart Garbage Bins) to control the energy efficiency of the system**.**

Dario Bonino et.al [2] has suggested that it provides end – to - end security and privacy that is built upon dynamic federation smart city platform. Its benefits is that it has good dependability and has resilience on failure of a system over a particular month. It focuses on the collection of wastages and accomplishment of ontology method.

A lvaro Lozano Murciegoet.al [3] has suggested that to collect the dustbins that are been filled using a truck. The main advantage is that it reduces the fuel cost of the trucks rather than travelling a long distance it makes the path simpler and easier to reach the dustbin using route optimization.

Theodoros Anagnostopoulos et.al [4] has suggested that it first starts with an assumption that the smart city must include the IoT base. It uses dynamic scheduling. It is based on the fact that the garbage will be collected only when it is fully filled or the maximum capacities of the dustbins are filled.

Rachael E. Marshall et.al [5] outlines that the smart waste management system in the high salaried countries and a developing countries.

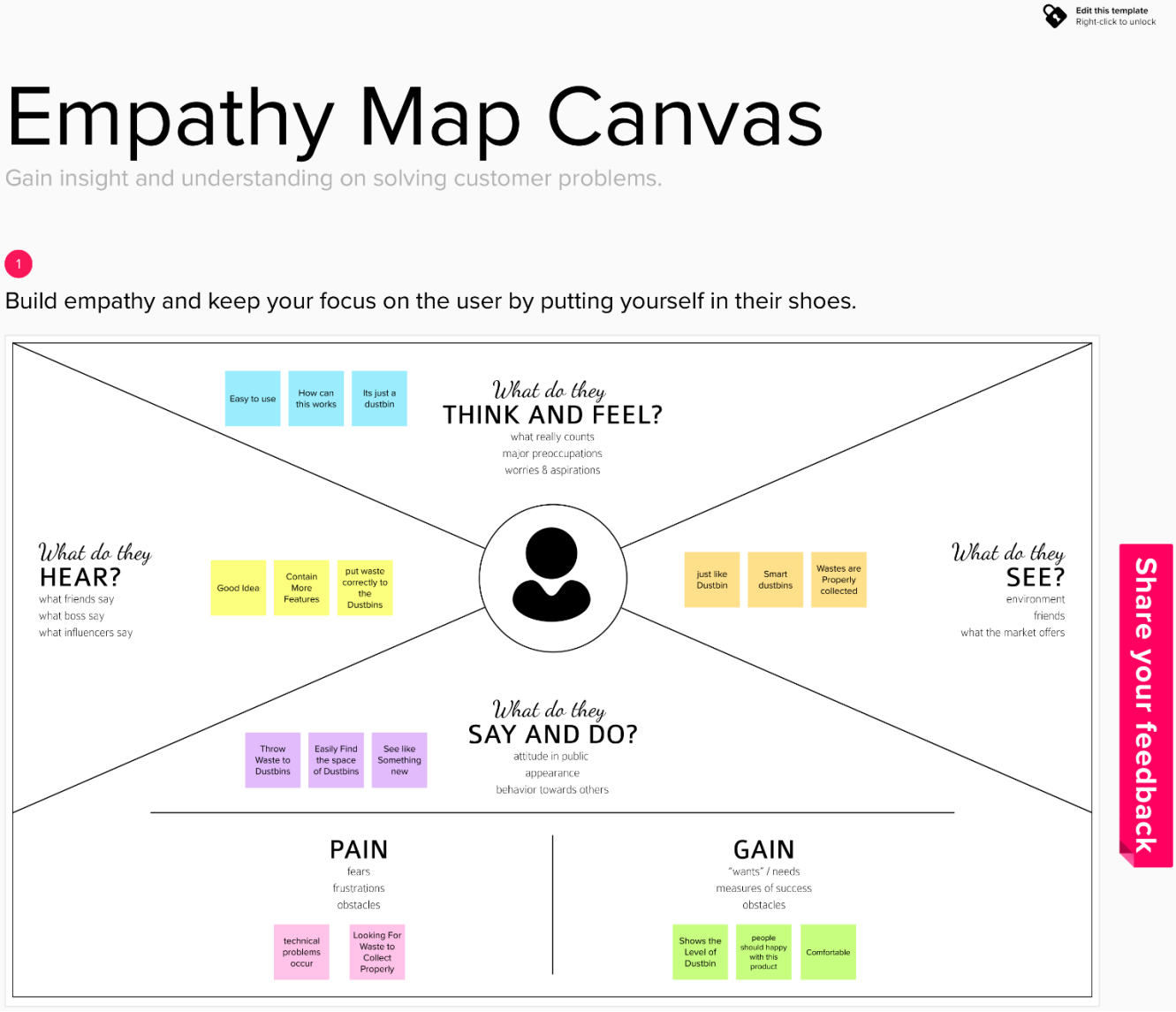
Lilliana Abarca Guerrero et.al [6] outlines the fact that the developing countries undergo a prominent factor of affecting the waste management systems due to rising population levels and rapidly growing urbanization. The collaborator of the waste management is many such as household, industry sectors, educational and research intuitions etc. Produces huge number of wastages. Collecting, transferring, Transportation of the wastages and they are finally disposed in an open land.

2.3 Problem Statement Definition

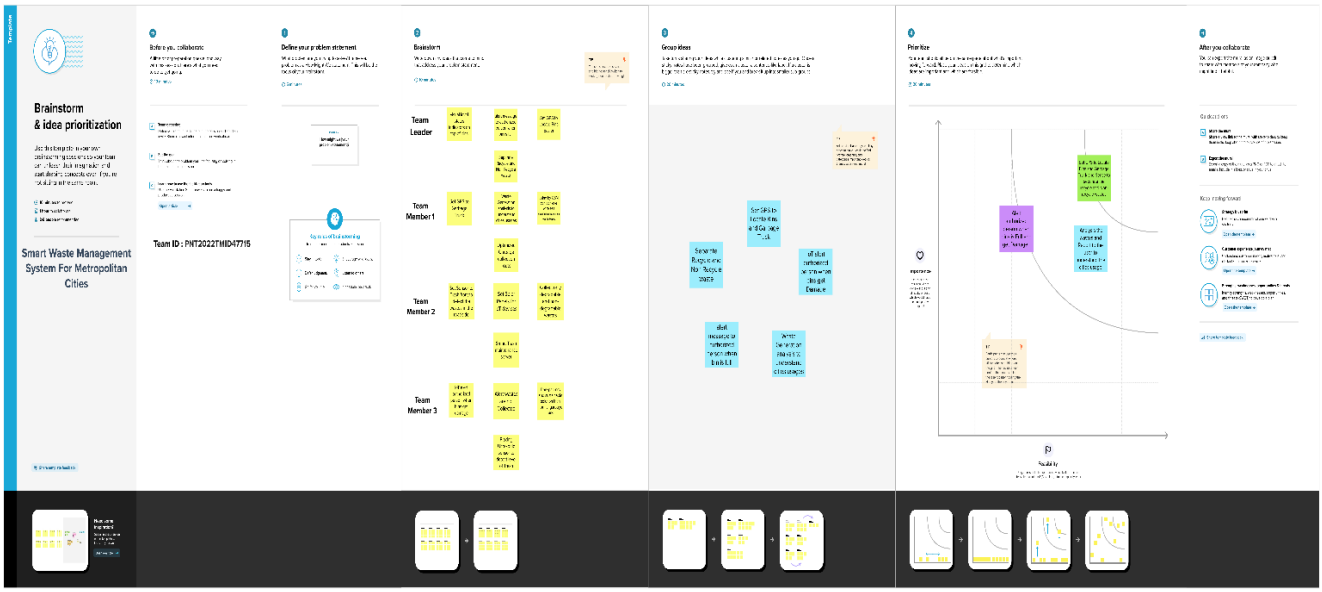
Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Customer	Puts a Wastes in Garbage	I can't	The Garbage is Full	Frustrated
PS-2	Customer	Put a waste in Garbage truck	I can't	The Garbage Truck is Loaded the wastes and Gone	Sad

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



3.2 Ideation & Brainstorming

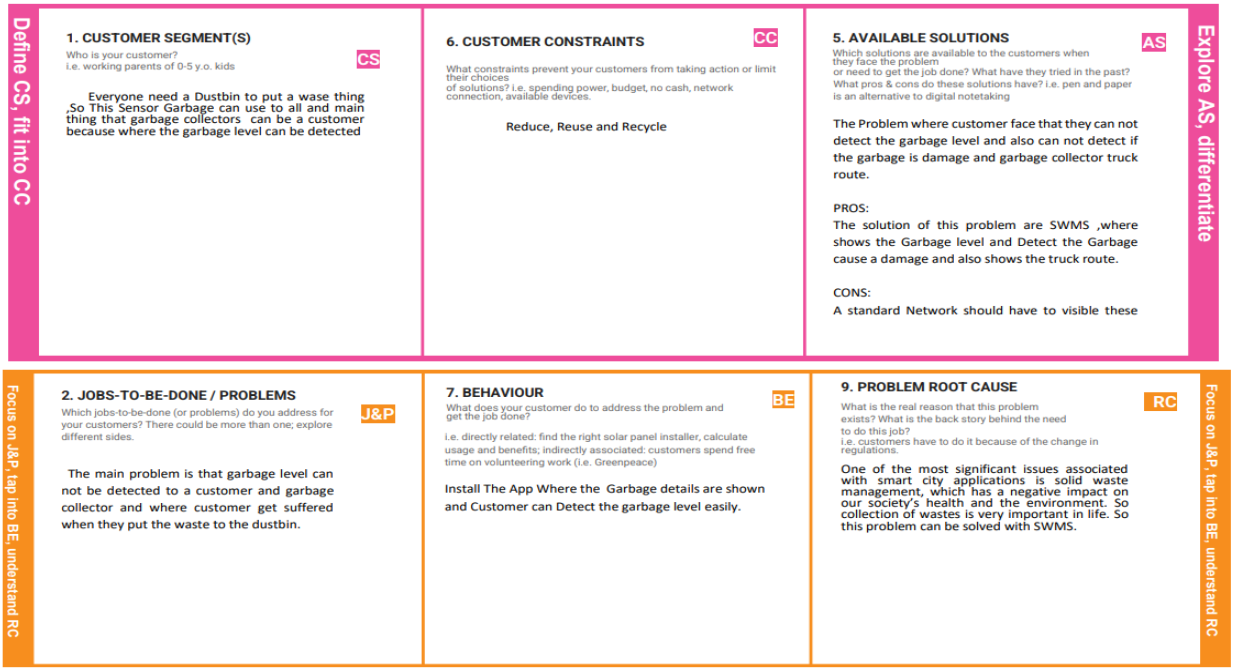


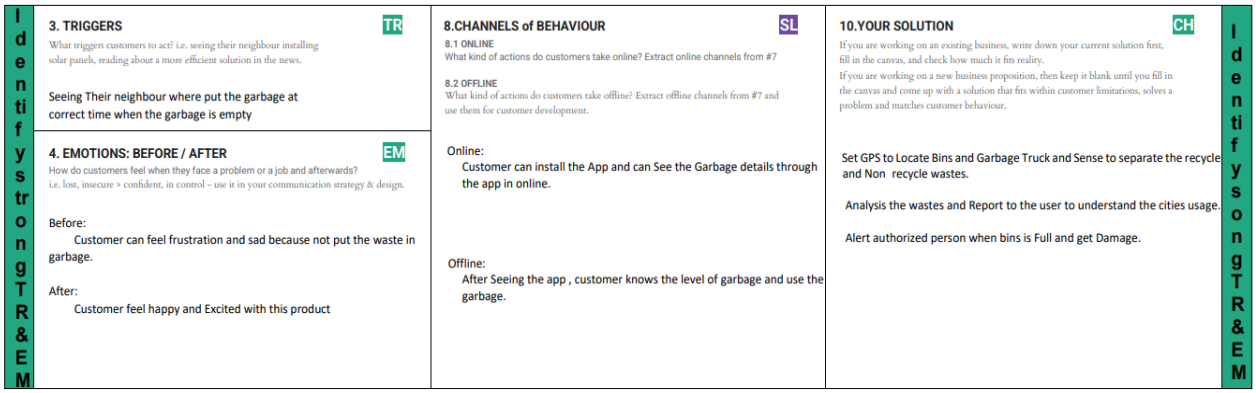
3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	This Project proposed with the problem of waste management in Smart Cities ,Where the Garbage Collection is not optimized. with this project ,Where the Garbage levels, Garbage Truck route and Garbage damage can be shown to the user
2.	Idea / Solution description	Whenever the Garbage get full or Damage in Bin can be Notified to the authorized person or user. It can be able to automate the solid waste monitoring process and management of the collection.
3.	Novelty / Uniqueness	The Uniqueness of this Project is That SWM makes notification to Authorized person or user when bin get full or cause damage.
4.	Social Impact / Customer Satisfaction	User can be satisfied as where the waste can be collected Properly and know about the Garbage level.

5.	Business Model (Revenue Model)	Waste Management organizes its operations into two reportable business segments: 1.Solid Waste, comprising the Company's waste collection, transfer, recycling and 2. Corporate and Other, comprising the Company's other activities and its recycling brokerage services ,as well as various corporate functions.
6.	Scalability of the Solution	One of the most significant issues associated with smart city applications is solid waste management, which has a negative impact on
		our society's health and the environment. In case of restaurants, malls and other commercial establishment garbage can be collected directly from the unit using vehicles But House and Streets where garbage levels can not be detected. To overcome this situation an efficient Smart Waste Management System For Metropolitan Cities has to be developed using Internet of Things.

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	**User Registration**	User able to register an account with email, password and confirm password.
FR-2	**Login**	User able to login into the system with email and password. The system shall display error message if the user enters wrong information.
FR-3	**Display Map Views**	User able to see the map views. The system require user to enable GPS. User able to see the dustbins' location. User able to see the dustbin's status. The system allows user gets the route direction from user's current location to dustbin's location.
FR-4	**Logout**	User able to logout the system.
FR-5	**Detect Dustbins' Volume**	The sensor shall detect the distance of dustbin and calculate in term of volume. The system able to update the dustbins' status to real-time database.

4.2 Non-Functional requirements

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

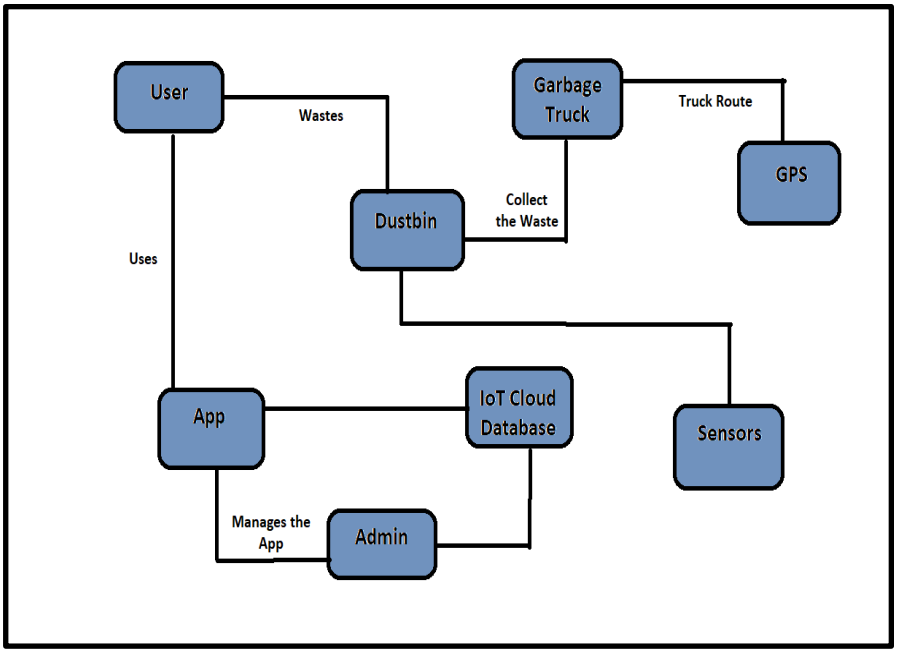
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The system can be accessed via internet. The system can be used in any android devices.
NFR-2	**Security**	The system can be only accessed by registered email and password.
NFR-3	**Reliability**	A reduction in the number of waste collections needed by up to 80%, resulting in less manpower, emissions, fuel use and traffic congestion.
NFR-4	**Performance**	The system can be accessed within 24 hours per day and 365 days per year.
NFR-5	**Availability**	the Availability of resources and the environmental conditions of a given society is essential for the development of an appropriate waste management system.
NFR-6	**Scalability**	Garbage level can be Detected at any time and where if the garbage cause damage can be notified to user.

5. PROJECT DESIGN

5.1 Data Flow Diagrams

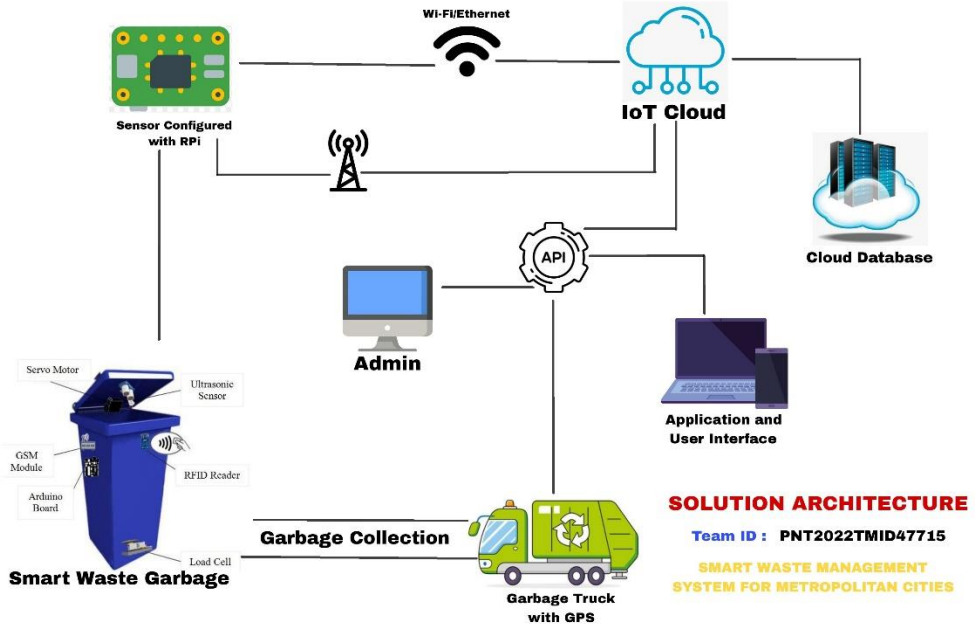
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Data Flow Diagram (Level 0):



5.2 Solution & Technical Architecture

Solution Architecture:



Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

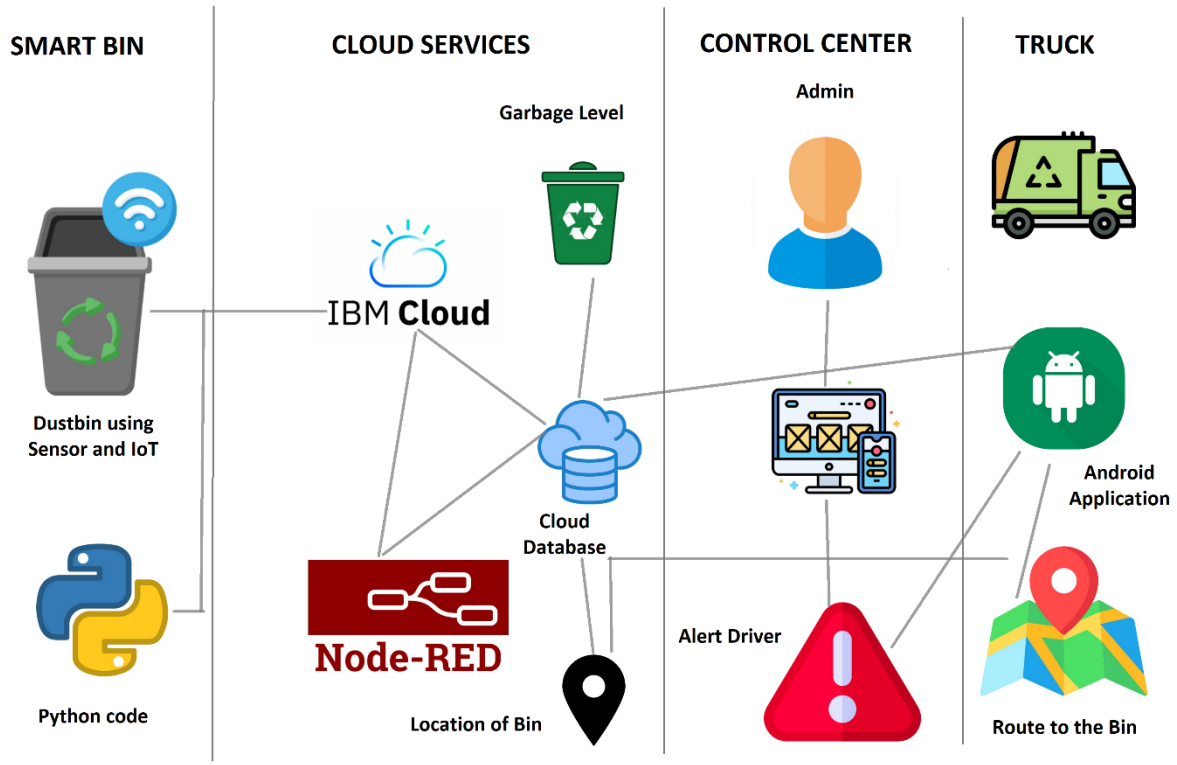


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	Application	User Interact with admin and know the status of the Dustbin	HTML, Python
2.	GPS	They receive the data from the satellites regarding the location of garbage to be collected	GPS satellites
3.	Ultrasonic Sensor	It measures the distance of garbage to the base of it using ultrasonic waves	A Transducer to send and receive ultrasonic pulses
4.	Cloud Database	For the storage of user info, location of garbage bins etc.,	IBM DB2, IBM Cloud etc.
5.	Lithium Ion Battery	A lithium-ion (Li-ion) battery is an advanced battery technology that uses lithium ions.	The positive to the negative electrode

6.	Wi-Fi Module	The ESP8266 Wi-Fi Module is a self-contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your Wi-Fi network.	IEEE 802 protocol
7.	Transport	A vehicle for the collection of bins.	Garbage truck

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Template for software development that is designed by a social network of software developers.	Python
2.	Security Implementations	provides the technical security policies, requirements, and implementation details for eliminating the security weaknesses	GSM/GPRS.
3.	Scalable Architecture	scalable architecture supports higher workloads without any fundamental changes to it.	Node Red.
4.	Availability	The quality or state of being available trying to improve the availability of affordable housing	Cloud, DB.
5.	Performance	The execution of an action	IBM Watson IoT Platform.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Admin	Login (App/Web Server)	USN-1	Admin gives User details for every Workers and manage them and manage garbage level when it full.	I can manage Web account/dashboard	High	Sprint-1
Customer	Login	USN-2	Customer where login to the App and can see the garbage level and put the wastes.	I can see the App and Put the Wastes	High	Sprint-2
Truck Driver	Worker	USN-3	As a Truck driver can pick the Wastes from the Garbage at a correct time when it full.	I can Collect the Wastes	High	Sprint-3

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Co-Admin	Worker	USN-4	As Co-Admin gets the notification when the garbage is damage or full and can show city people as Wastes usage.	I can show the Usage of People's Wastes	Medium	Sprint-2
Customer Care	Worker	USN-5	Customer care where handles the Customers questions and Problems facing from the Smart bins and gives some instruction.	I can help the customer's facing problems	Medium	Sprint-4

6.PROJECT PLANNING & SCHEDULING

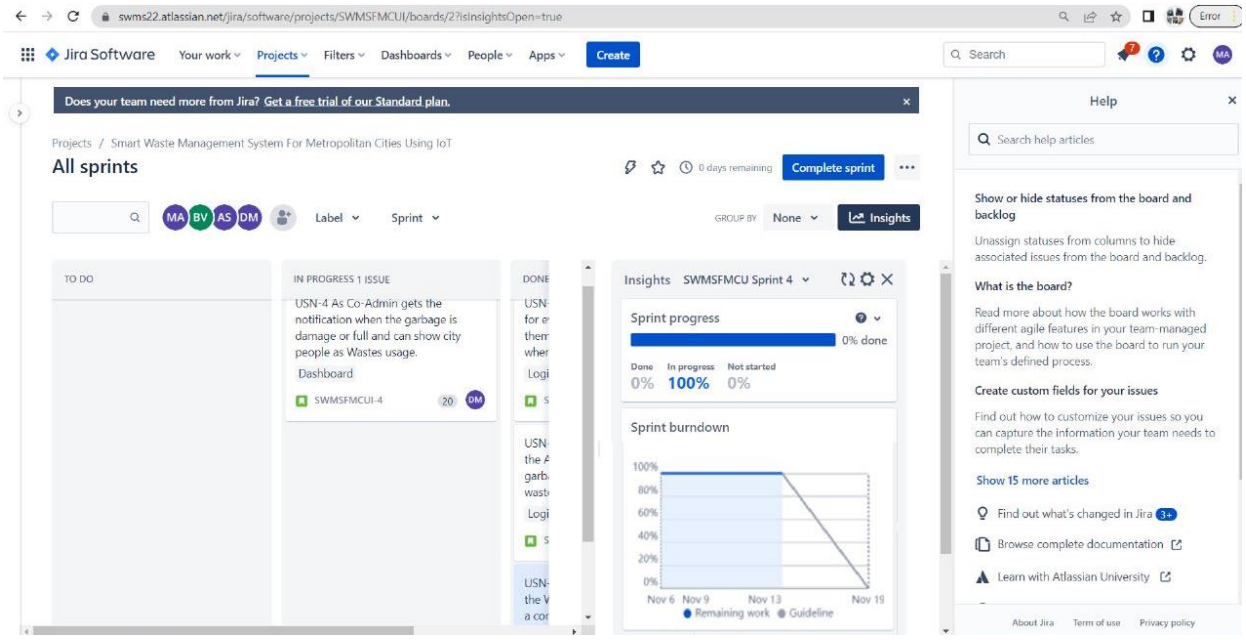
6.1 Sprint Planning & Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	Admin gives User details for every Workers and manage them and manage garbage level when it full.	20	High	Mohamed Adhil
Sprint-2	Login	USN-2	Customer where login to the App and can see the garbage level and put the wastes.	20	High	Balaji
Sprint-3	Dashboard	USN-3	As a Truck driver can pick the Wastes from the Garbage at a correct time when it full.	20	Low	Majith
Sprint-4	Dashboard	USN-4	As Co-Admin gets the notification when the garbage is damage or full and can show city people as Wastes usage.	20	Medium	Dhaha

6.2 Sprint Delivery Schedule

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	!9 Nov 2022

6.3Reports from JIRA



7.CODING & SOLUTIONING

7.1 Simulation Creation

Creating SmartBin with Ultrasonic sensor and ESP32 microcontroller in Wokwi for Simulating the SmartBin in Smart waste management System. Here there are four SmartBins are created and connected with IoT Watson Platform and Node Red.

Code:

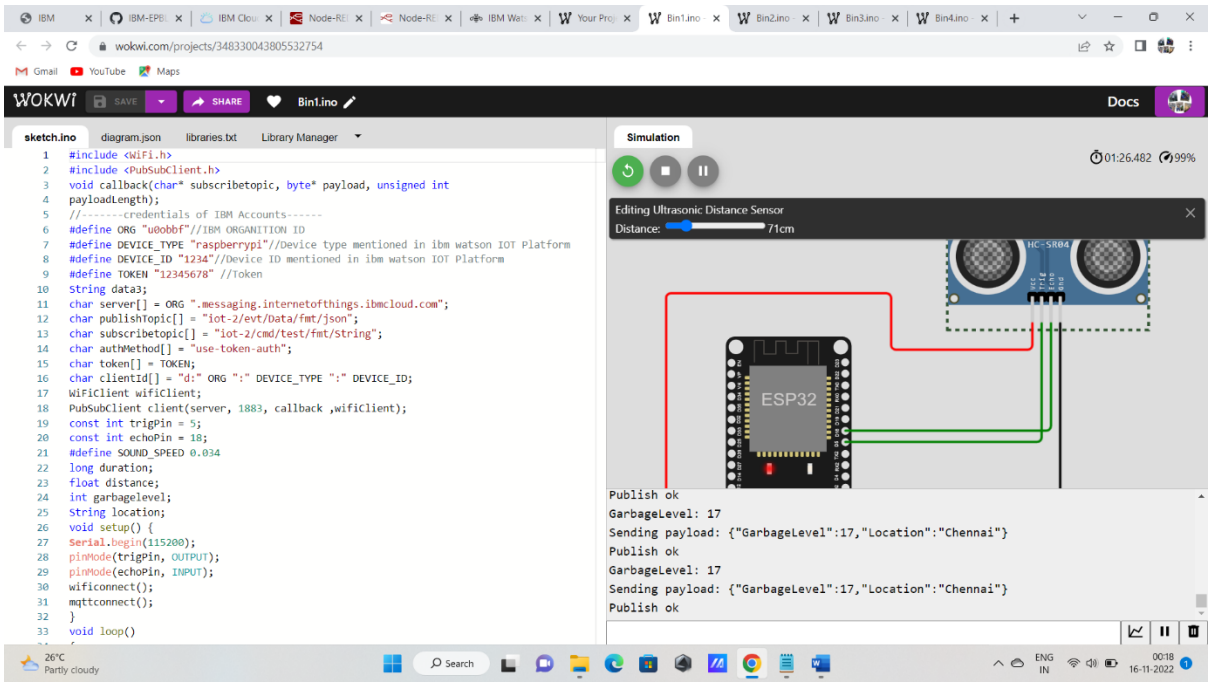
```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "u0obbf"//IBM ORGANITION ID
#define DEVICE_TYPE "raspberrypi"//Device type mentioned in ibm watson IOT
Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT Platform
#define TOKEN "12345678" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
int garbagelevel;
String location;
void setup() {
Serial.begin(115200);
```

```
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
garbagelevel = distance*100/400;
if(distance<350)
{
Serial.print("GarbageLevel: ");
Serial.println(garbagelevel);
PublishData(garbagelevel);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
if(distance>350)
{
Serial.println("ALERT!!");
Serial.print("GarbageLevel: ");
Serial.println(garbagelevel);
delay(1000);
PublishData(garbagelevel);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
void PublishData(int dist) {
mqttconnect();
if(distance<350)
{
String payload = "{\"GarbageLevel\":";
payload += dist;
```

```
payload += ",";
payload += "\"Location\":";
payload +=  "\"Chennai\"";
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(distance>350)
{
String payload = "{\"GarbageLevel\":";
payload += dist;
payload += ",\"ALERT!!\":""\"Garbage Level is Full\"";
payload += ",";
payload += "\"Location\":";
payload +=  "\"Chennai\"";
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");
}
}
}
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()
{
Serial.println();
```

```
    Serial.print("Connecting to ");
    WiFi.begin("Wokwi-GUEST", "", 6);
    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)
    {
    Serial.print("callback invoked for topic: ");
    Serial.println(subscribetopic);
    for (int i = 0; i < payloadLength; i++) {
    //Serial.print((char)payload[i]);
    data3 += (char)payload[i];
    }
    Serial.println("data: "+ data3);
    data3="";
    }
```

Images :

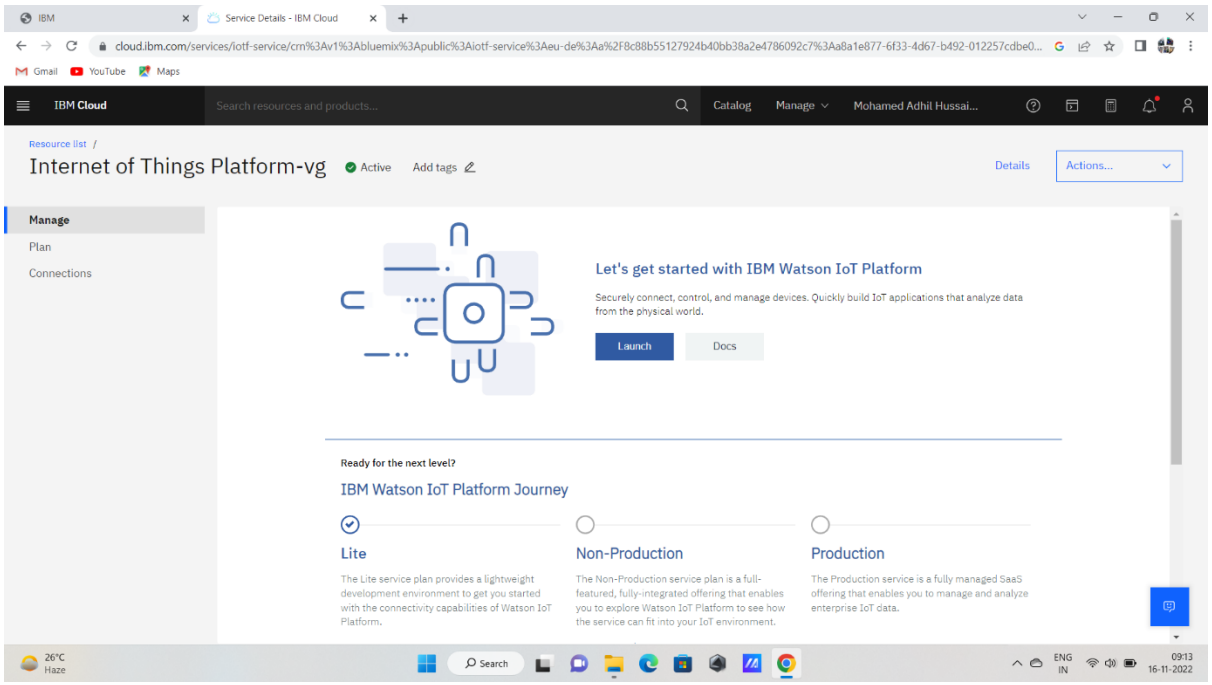


7.2 Create Device and Workflow of IoT Scenarios

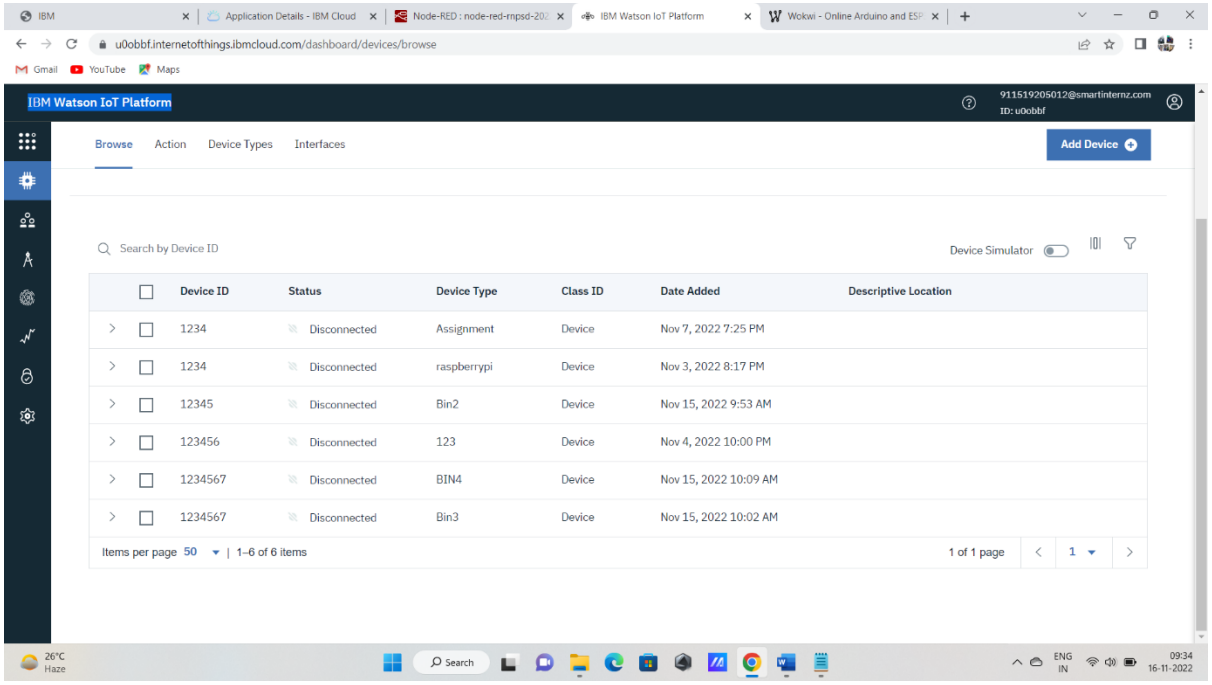
Create device in the IoT Watson platform, workflow for IoT scenarios using local node red

<u>IBM Watson IoT Platform:</u>

In IBM Watson IoT Platform, There are four Devices where Created for four SmartBins and can be Executed and The Images are Shown Below,

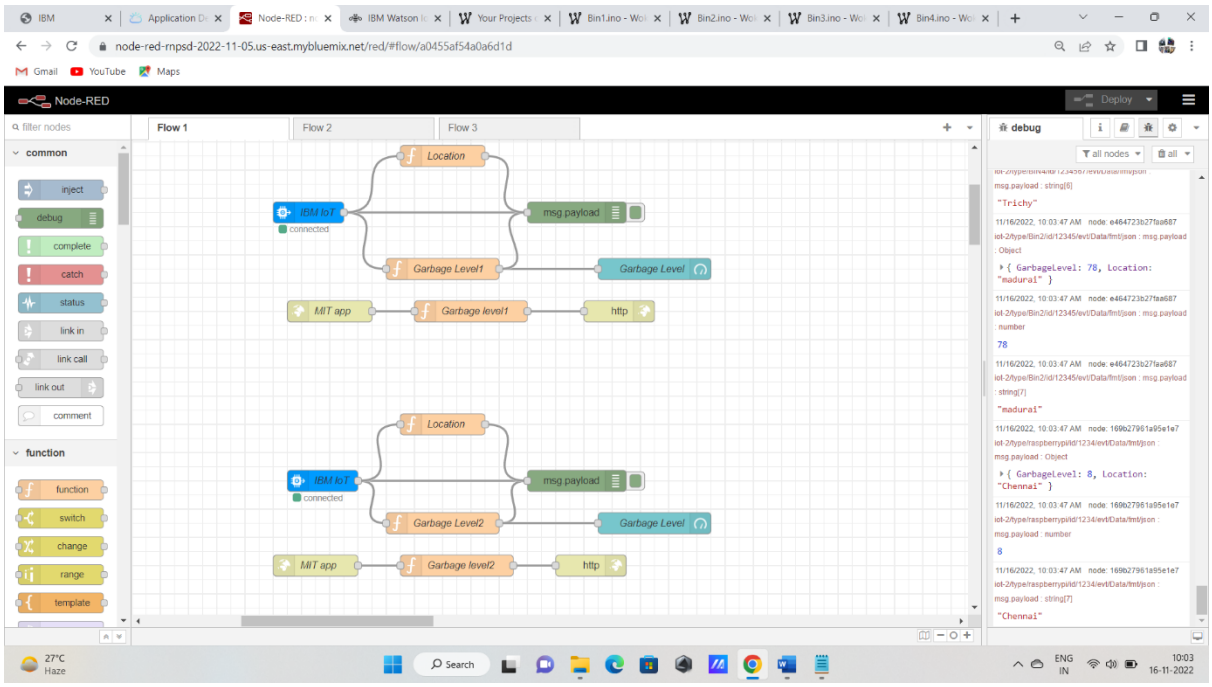


This Device Were Connected to Wokwi for Simulation and Node-RED for workflow. The device which shows the data of the Garbage level and shows the Alert message of Garbage and Authorized Person can be Seen.



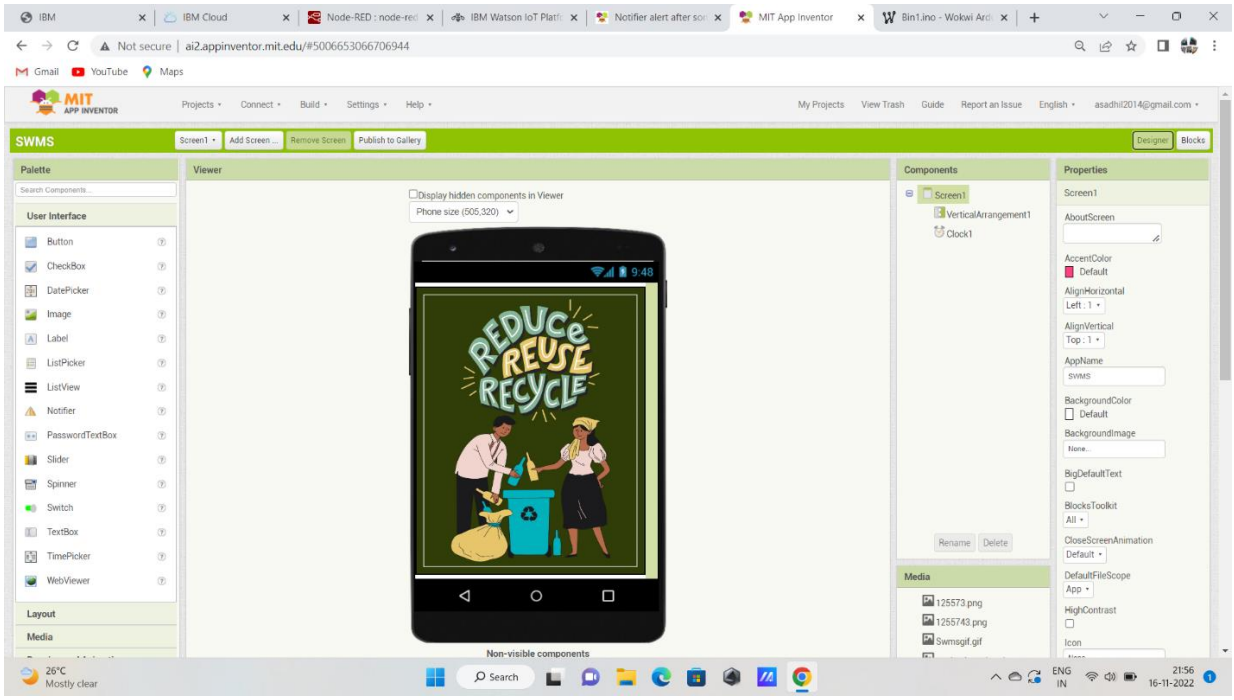
Node-RED:

The Workflow of Smart waste Management System Can be Executed in Node-RED. Here There Lots of Nodes were connected such as IBM IoT, Garbage Level, Location and Garbage level Chart. These Nodes are connected to Device and Wokwi to Show the Garbage level and Alert Messages.

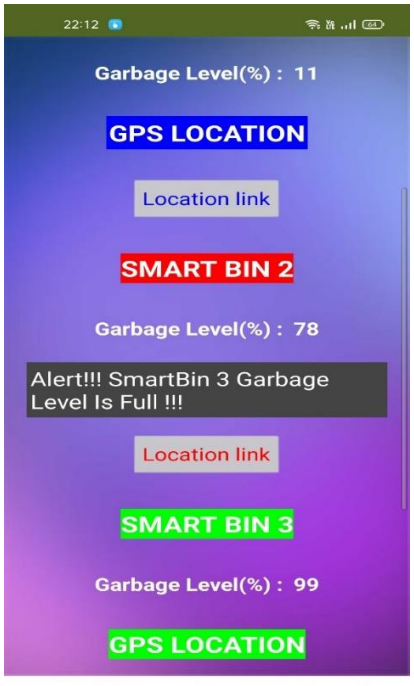


7.3 MIT App Inventor

The Application for Smart Waste Management System Using MIT app Inventor. These App can be used by user where can check the Status level of Garbage and Location (Random) of the Smart Bin. If The Garbage level of Smart Bin is Full, then It Shows Alert Message in the App.



Here There are Three Screen were used and each screen contain some Action as Screen 1: Picture of waste management. Screen 2: Login Page as contain Username and Password. Screen 3: Garbage level and Location (Random) of SmartBin



8. TESTING

8.1 Test Cases

				Date	03-Nov-22								
				Team ID	PNT2022TMID47715								
				Project Name	Project - Smart Waste Management System For Metropolitan Cities								
				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
App_Login_ID_001	UI	Home Page	Verify the 10sec image of Screen	User use the App	1.Download The App 2.Click on the App 3. There is A image At 10sec	http://ai2.appinventor.mit.edu/#5006653066706944	Screen 1 Contain 10 Sec of images and where get to next screen	Working as expected	Pass	This Test is Successfully Excuted	N	Null	Mohamed Adhil
App_Login_ID_002	Functional	Login page	verify the username and Password with Valid Credential	User use the login credential	1.Enter the Username and password	Username: SWMS password: 1234	Correct username and password shows the 'Login Succesfull'	Working as expected	pass	This Test is Successfully Excuted	N	Null	Moamed Adhil
App_Login_ID_003	Functional	Login page	Verify the username and Password with inValid Credential	User use the login credential	1. Enter the Username and Password	Username: SWMS password: 1234	InCorrect username and password shows the "invalid username or password"	Working as expected	pass	This Test is Successfully Excuted	N	Null	Balaji
App_Working_ID_001	Functional	Main page	Verify that App show the garbage level and location	Check the level of garabge	1.See the Garbage level and location	Garbgae level : 0 and location	Shows the correct vaue of garbage level and Location	Working as expected	pass	This Test is Successfully Excuted	N	Null	Majith
App_Working_ID_00	Functional	Main page	verify that App shows the Alert when garbage level is full	Check the Alert message	1.See the Alert message of Garbage level	Alert Message "Alert! Garbage Level is Full"	Shows the Alert Message	Working as expected	Pass	This Test is Successfully Excuted	N	Null	Dhaha

8.2 User Acceptance & Performance Testing

User Acceptance:

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 For Metropolitan Cities 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	2	3	20

Duplicate	1	0	0	0	1
External	2	3	0	2	7
Fixed	11	5	4	19	39
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	2	1	0	3
Totals	24	14	9	25	72

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	3	0	0	3
Client Application	2	0	0	20
Security	2	0	0	2
Outsource Shipping	6	0	0	6
Exception Reporting	8	0	0	8
Final Report Output	2	0	0	2
Version Control	1	0	0	1

Performance Testing

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	Smart waste Managen	New	No Changes	No Changes	Moderate	6.56 Sec	>5 to 10%	GREEN	As we have seen the chnages
2	Smart waste Managen	Exisiting	No Changes	No Changes	No Changes	0.25ec	No Changes	GREEN	No Changes

NFT - Detailed Test Plan

S.No	Project Overview	NFT Test approach	umptions/Dependencies/R	Approvals/SignOff
1	Smart waste Management System Fo	Load Changes	App get Slow when Alert Messa	Approval from Admin(Team Lead)

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 Managen	App Can be Slow at aler	Each alert get Individual	Smooth in Alert message	Code is Changed to indivi	Get to Individual Code	Slow At message of Alert	Approval from Admin(Team Lead)

9. RESULTS

9.1 Performance Metrics

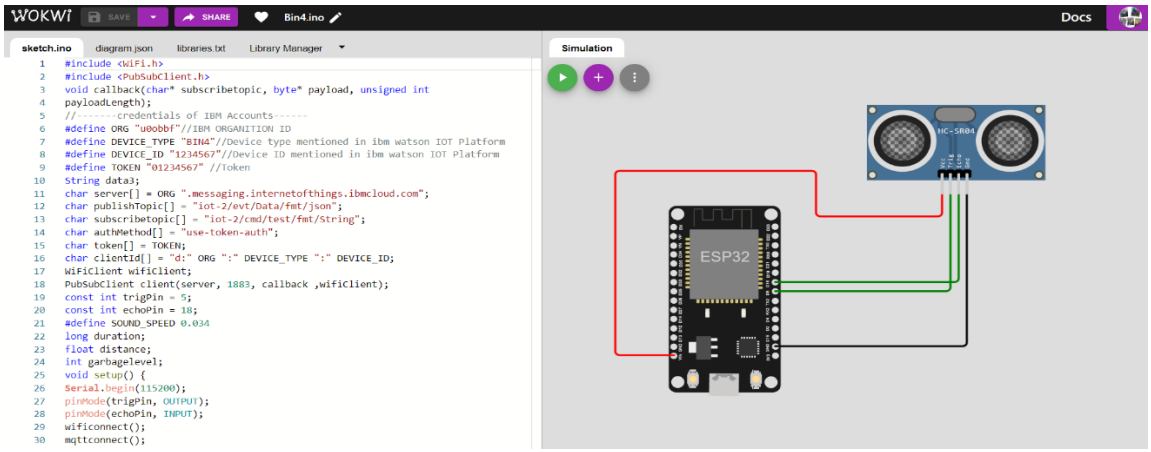
Wokwi for Simulation

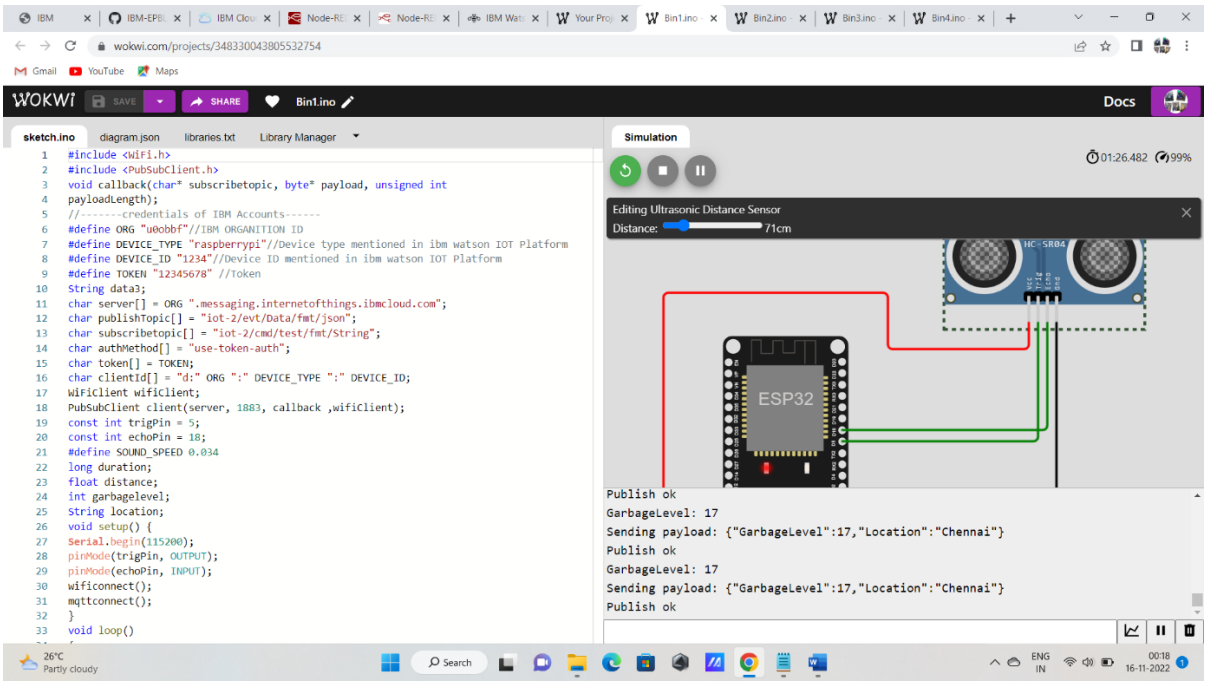
The User Interact with Wokwi Software as where the IoT of Smart Waste Management System Contain ESP32 Microcontroller and Ultrasonic Sensor are Connected. These Sensor Were used to Calculate the Level of Garbage and can be Detected with Code

The level of Garbage where user need to operate the Senor because of Simulation. Then This Wokwi were connected with IBM IoT Watson platform using code of device Type, ID, Authentication Token.

The Following Images Shows the User Interaction with Ultrasonic Sensor in Wokwi and the Link are given below.

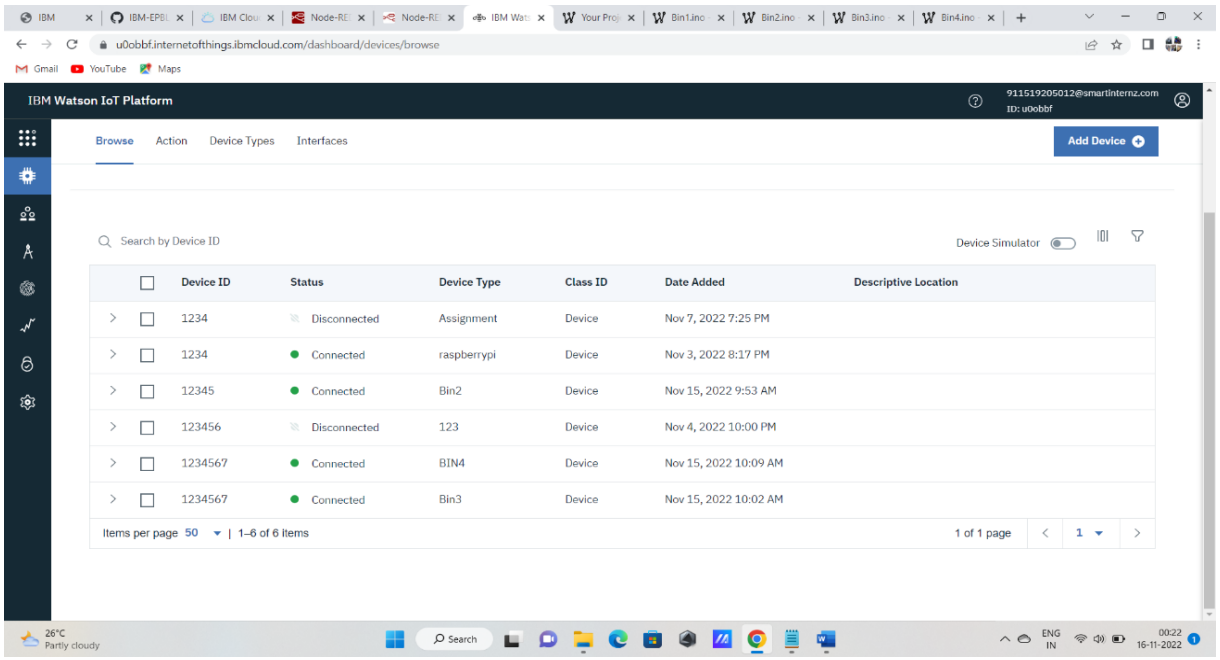
SmartBin 1: https://wokwi.com/projects/348330043805532754

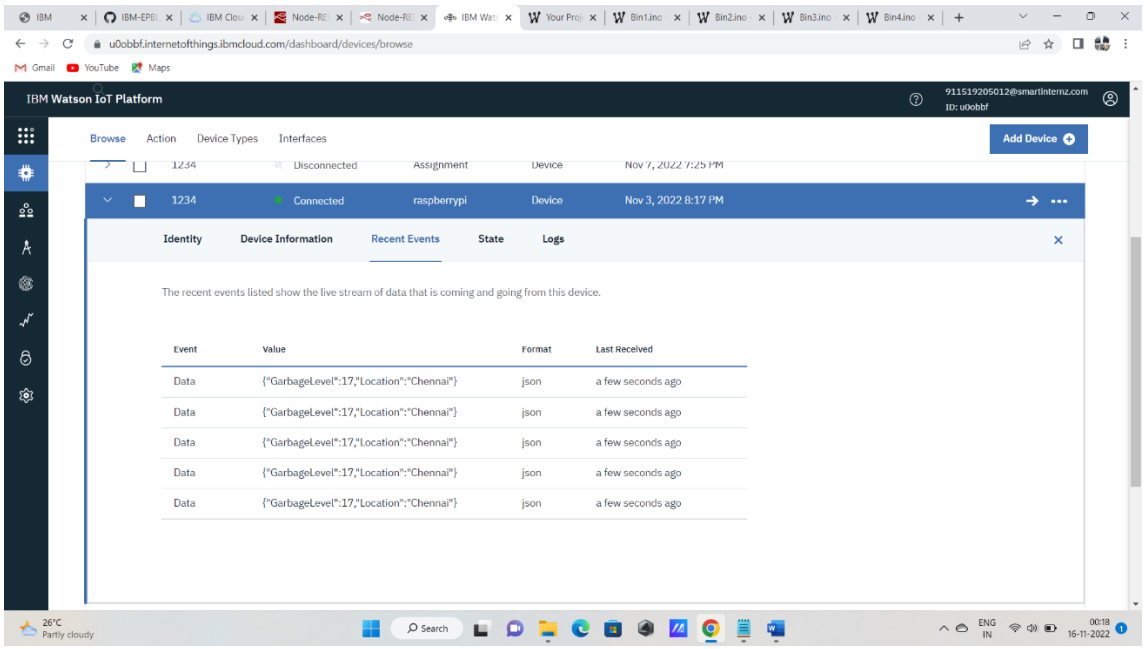




IBM Watson IoT Platform:

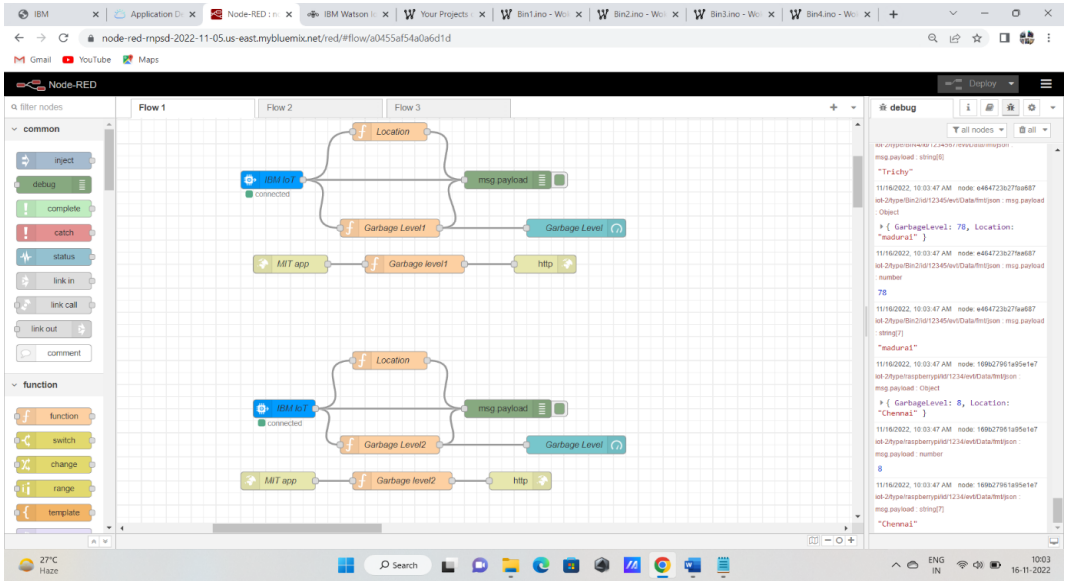
Create a device in IBM Watson IoT platform and Connect to the Wokwi. The Data's from Wokwi were send to IoT Watson Platform and The User can be seen and detected.

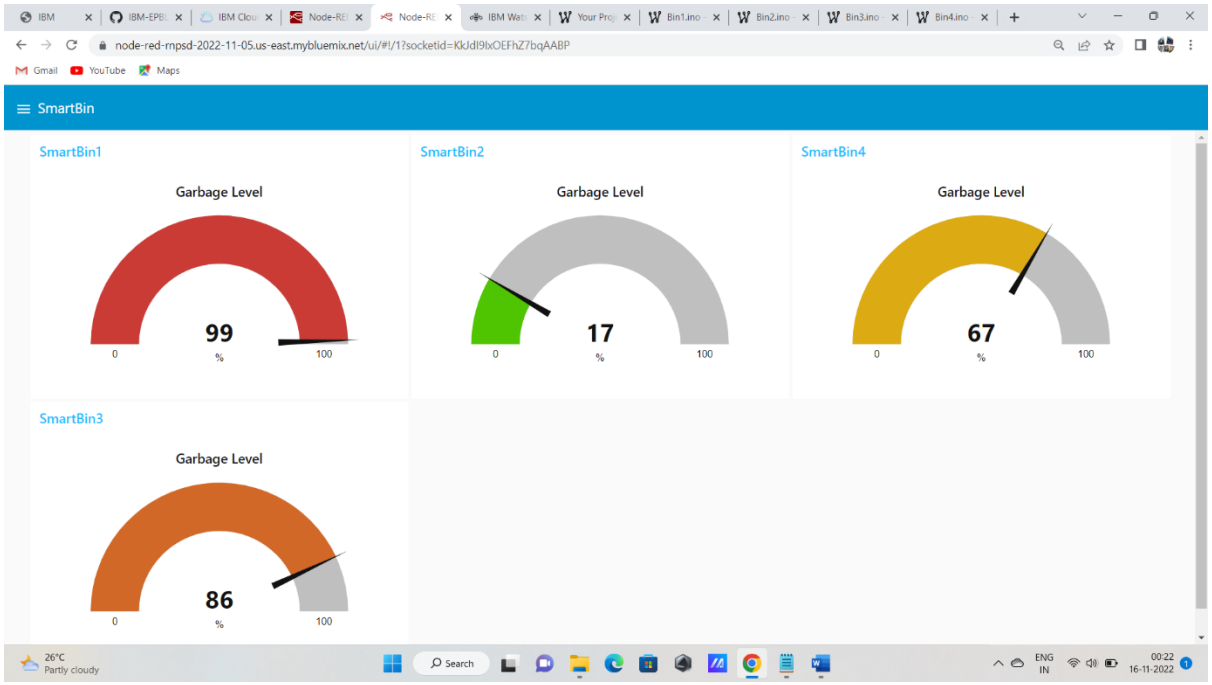




Node-RED and Dashboard

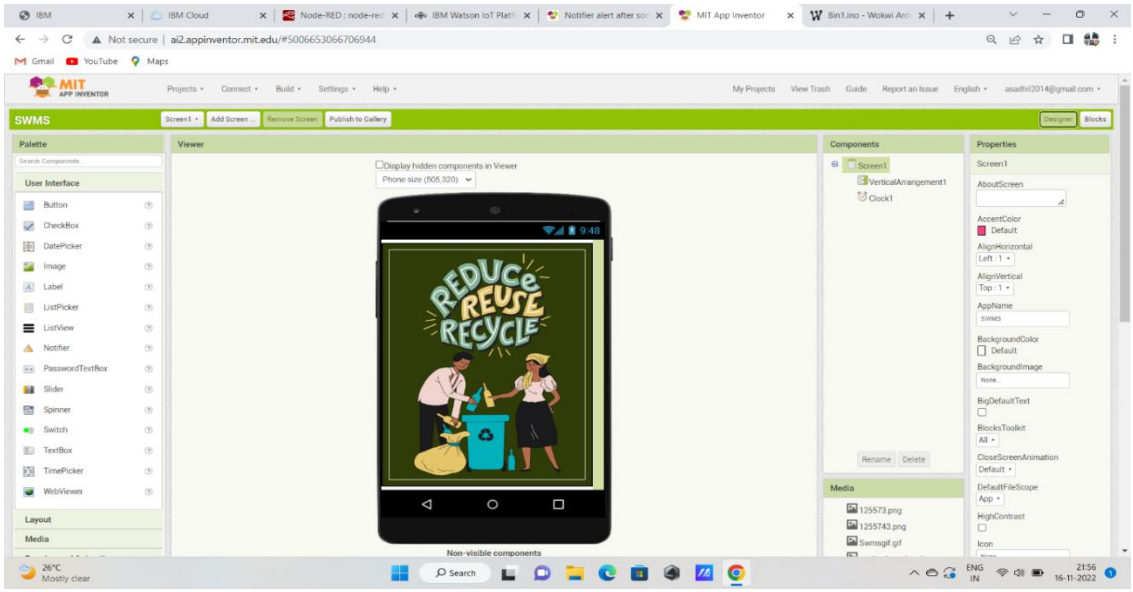
The Node-RED which is used for workflow as where the IBM Watson IoT platform and MIT app Inventor are connected with Nodes. This IBM IoT was connected with Wokwi. The Node-RED contain Dashboard as Chart for Level of the Garbage.

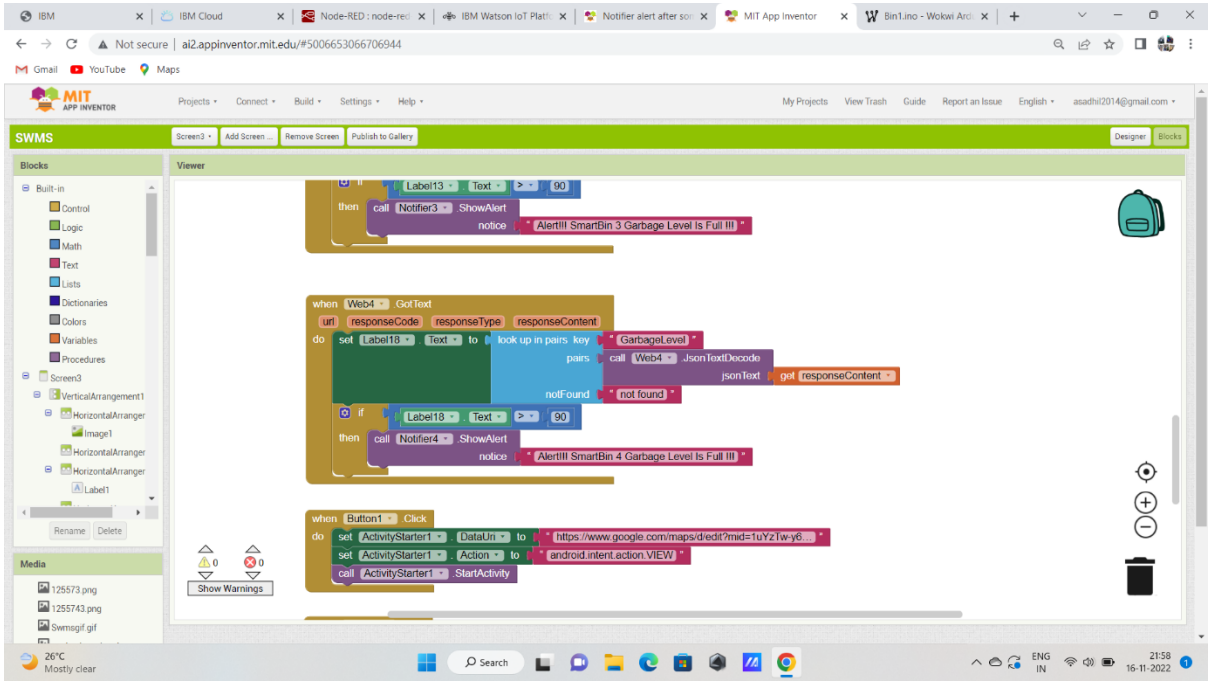


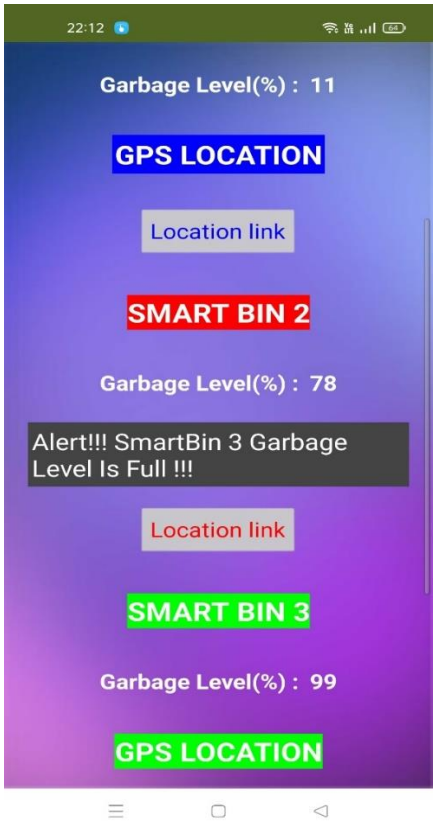


MIT App Inventor

With Help of MIT App Inventor, The User can Download the app in mobile and can be Executed to See the Level of Garbage with Alert Message and Location(Random).







10. ADVANTAGES & DISADVANTAGES

Advantages

- Reduction in Collection Cost. The solution reduces waste collection frequency dramatically, enabling you to save on fuel, labor, and fleet maintenance costs.
- No Missed Pickups.
- Reduced Overflows.
- Waste Generation Analysis.
- CO2 Emission Reduction.

Disadvantages

- Misunderstanding of the operations of smart sensors
- Non-optimized truck routes

- Setting up the smart sensor

11. CONCLUSION

The Conclusion of this project is to make a city clean, and the waste things are collected properly at a correct time. With this Ideation, the project contains Sensor to Detect the level of Garbage and shows the location of the Garbage.

So, we just made App to detect the level and location using MIT App Inventor and were connected to the IBM Watson IoT platform.

Finally, The Level of Garbage is detected and where the alert can be shown to the User and Admin when the Garbage is full.

12. FUTURE SCOPE

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

1. Change the system of user's authentication and atomic lock of bins which would help in securing the bin from any kind of damage or theft.

2. Concept of green points that would encourage the involvement of the residents or the end users making the idea successful and helping to achieve joined efforts for the waste management and hence fulfilling the idea of Swachh Bharath.

3. Having case study or data analytics on the type and times the waste is collected on the type of days or season making the bin filling predictable and removing the dependency on electronic components and fixing the coordinates.

4. Improving graphical interfaces for the Server and complete Android applications has possibility of extending the system adding other use cases and applications for smart cities.

13. APPENDIX

Source Code

```
#include <WiFi.h>
#include <PubSubClient.h>
void callback(char* subscribetopic, byte* payload, unsigned int
payloadLength);
//-------credentials of IBM Accounts------
#define ORG "u0obbf"//IBM ORGANITION ID
#define DEVICE_TYPE "raspberrypi"//Device type mentioned in ibm watson
IOT Platform
#define DEVICE_ID "1234"//Device ID mentioned in ibm watson IOT
Platform
#define TOKEN "12345678" //Token
String data3;
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/Data/fmt/json";
char subscribetopic[] = "iot-2/cmd/test/fmt/String";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
WiFiClient wifiClient;
PubSubClient client(server, 1883, callback ,wifiClient);
const int trigPin = 5;
const int echoPin = 18;
#define SOUND_SPEED 0.034
long duration;
float distance;
int garbagelevel;
String location;
void setup() {
```

```
Serial.begin(115200);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
wificonnect();
mqttconnect();
}
void loop()
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration * SOUND_SPEED/2;
garbagelevel = distance*100/400;
if(distance<350)
{
Serial.print("GarbageLevel: ");
Serial.println(garbagelevel);
PublishData(garbagelevel);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
if(distance>350)
{
Serial.println("ALERT!!");
Serial.print("GarbageLevel: ");
Serial.println(garbagelevel);
delay(1000);
PublishData(garbagelevel);
delay(1000);
if (!client.loop()) {
mqttconnect();
}
}
delay(1000);
}
```

```
void PublishData(int dist) {
mqttconnect();
if(distance<350)
{
String payload = "{\"GarbageLevel\":";
payload += dist;
payload += ",";
payload += "\"Location\":";
payload +=  "\"Chennai\"";
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(distance>350)
{
String payload = "{\"GarbageLevel\":";
payload += dist;
payload += ",\"ALERT!!\":""\"Garbage Level is Full\"";
payload += ",";
payload += "\"Location\":";
payload +=  "\"Chennai\"";
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");
}
}
}
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()
{
Serial.println();
Serial.print("Connecting to ");
WiFi.begin("Wokwi-GUEST", "", 6);
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)
{
Serial.print("callback invoked for topic: ");
Serial.println(subscribetopic);
for (int i = 0; i < payloadLength; i++) {
//Serial.print((char)payload[i]);
data3 += (char)payload[i];
}
Serial.println("data: "+ data3);
data3="";
```

}

Link:

IBM Cloud: https://cloud.ibm.com/

IoT Watson Platform: https://u0obbf.internetofthings.ibmcloud.com/dashboard/devices/browse

Node-RED: https://node-red-rnpsd-2022-11-05.us-east.mybluemix.net/

MIT App Inventor: http://ai2.appinventor.mit.edu/#5006653066706944

GitHub Link: https://github.com/IBM-EPBL/IBM-Project-10750-1659201528

Project Demo Link:

https://drive.google.com/drive/folders/1rzW4IMFIYqDDz5CqNHLSPwyOqsNZFsxR