PROJECT REPORT

SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITIAN CITIES USING IOT

TEAM ID - PNT2022TMID40285

TEAM MEMBERS

ILAMPIRAI R

AMREEN A

KALAIVANI E

ASWINI V

KALAIARASI D

1. INTRODUCTION

- Smart waste management system in metropolitan towns is approximately the usage of generation and facts to create a extra efficient waste enterprise primarily based totally on IOT generation

1.1 Project Overview

- The venture is primarily based totally on a real-time clever rubbish bin mechanism for stable waste control in clever towns

1.2 Purpose

- A waste control machine is the approach an company makes use of to dispose reduce, reuse and prevent waste

2. LITERATURE SURVEY

2.1 EXISTING SYSTEM

Heavy metals and other toxic compounds from landfills, pollution.

2.2 REFERENCES

1. Smart Waste Management: Garbage Monitoring Using Iot 1Mrs Sarmila SS , 2 Siva Kumar V, V3asanth Kumaur P K 1Assistant Professor .Department of Computer Science and Engineering K.L.N. College of Engineering Madurai, India ISSN: 2348 – 8387 (APRIL ,2018)

2. Review Paper on Implementation of Automatic Waste Management System Using IOT & Android for Smart Cities Pulkit Bindal1 , Utkarsh Srivastava2 , Chirag Agarwal3 , Himanshu Gupta4 , Chhaya Sharma5 1,2,3,4

Department of Computer Science and Engineering, Raj Kumar Goel Institute of Technology, GhaziaD ISSN: 2349-6002 (MAY 2022)

3. Location Based Garbage Management System for Smart City Harini P K S1 , Ramya S1 , Yamini R2 1 Student, Dept. of Computer Science and Engineering, Adhiyamaan College of Engineering, Hosur, India (november-2020)

4. IoT Enabled Smart Waste Bin with Real Time Monitoring for efficient waste management in Metropolitan Cities Manju Mohan1 , RM. Kuppan Chetty1 , Vijayram Sriram2 , Mohd. Azeem2 , P. Vishal2 and G. Pranav2 1Centre for Automation and Robotics (ANRO), School of Mechanical Sciences, Hindustan Institute of Technology and Science, Padur, Chennai –603103 ISSN : 2619-8150 Volume 1, Number 3, (September 2019)

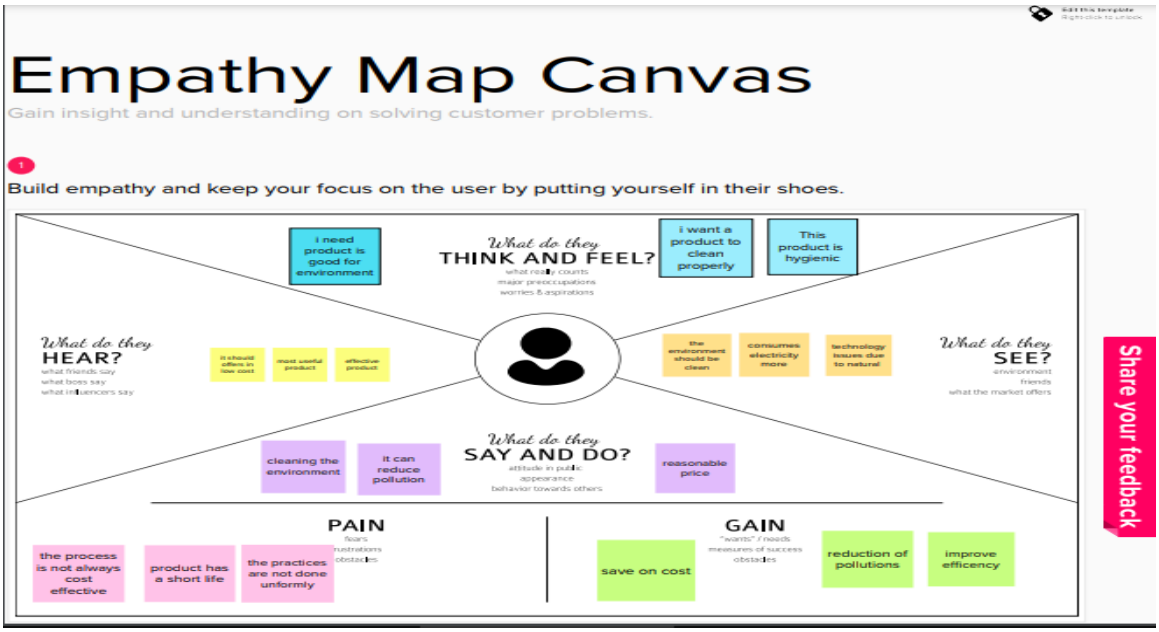
5. Smart Waste Management System using IOT Tejashree Kadus1 , Pawankumar Nirmal2 , Kartikee Kulkarni3 Department of Mechanical Engineering MIT Academy of Engineering, Pune Savitribai Phule University (April 2020)

2.3 PROBLEM STATEMENT

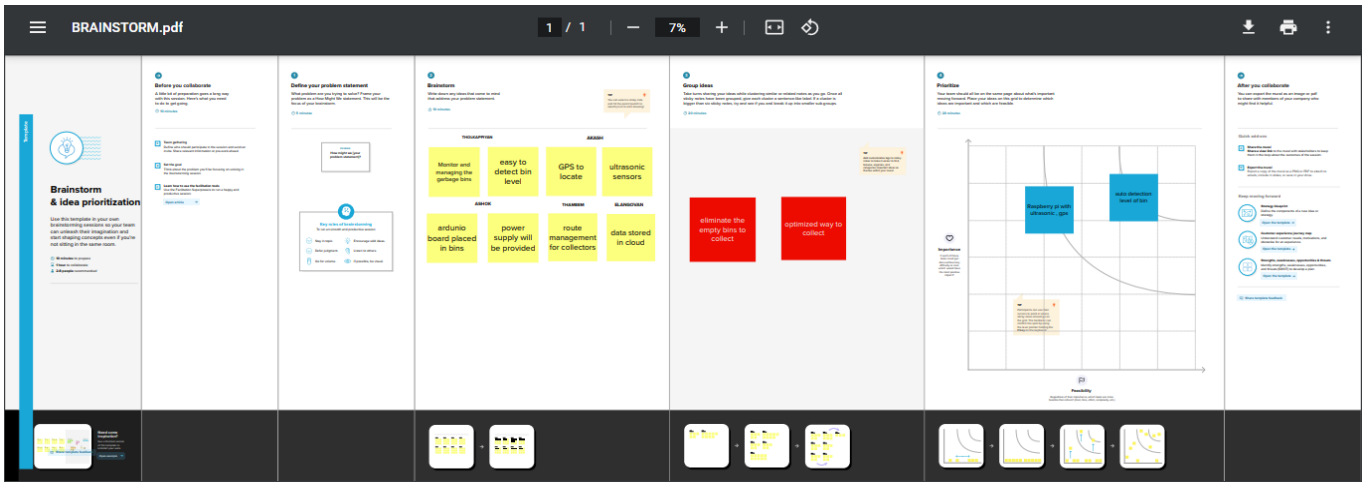
Indiscriminate disposal of waste is a major issue in most developing countries' urban centers and poses a serious threat to the healthy living of the citizens. The fill level of waste in each of the containers, which are strategically situated across the communities, is detected using sensors.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



3.2 IDEATION AND BRAINSTROMING



3.3 PROPOSED SOLUTION

S.No	Parameter	Description
1.	Problem Statement	This project addresses the issue of waste management in smart cities with

	(Problem to be solved)	inefficient garbage collection systems. With the help of this initiative, the enterprises may get the intelligent garbage management solutions they require. This technology enables the authorised person to provide truck drivers with a time- and cost-efficient route by always knowing the level of fill in each garbage can in a neighbourhood or city.
2.	Idea / Solution description	The following are the main research goals: • The proposed system would be able to use IOT to control the complete collection process and automate the solid waste monitoring procedure (Internet of Things). • The circuit at the garbage bin, which communicates it to the receiver at the desired location in the area or spot, is placed at the waste bin in the proposed system to acknowledge whenever the waste bin is filled. • In the suggested system, the signal received from the monitoring and control system indicates the status of the waste bin.
3.	Social Impact / Customer Satisfaction	According to popular perception, the direct social effects of current solid waste disposal procedures, such as the proximity of landfills to neighbourhoods, the development of pests, and the decline in property values, are the worst effects.
4.	Business Model (Revenue Model)	Solid Waste, which includes the Company's waste collection, transfer, recycling, and resource recovery, as well as its

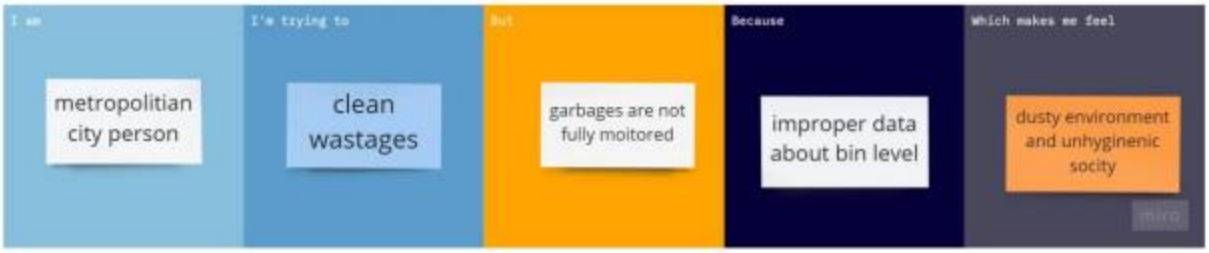
		resource recovery and disposal services, are run and managed locally by the Company's various subsidiaries, which concentrate on specific geographic areas. Corporate and Other, which includes the Company's other activities, such as the development and operation of landfill gasto-energy facilities
5.	Scalability of the Solution	In order to address this issue, smart city design is being researched and discussed more and more globally. Following this methodology, this article proposed a powerful IoT-based, realtime trash management model with an emphasis on citizens to enhance urban living conditions. The proposed method makes use of sensor and communication technologies, collecting garbage information from the smart bin in real-time and sending it to an internet site that city residents may access to see whether the compartments are still available.

3.4 PROBLEM SOLUTION FIT

CUSTOMER PROBLEM

The main problem of customers is improper maintenance of the garbage bins.

It leads to various problems like unhygienic environment , soil pollution and etc.



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

FR No.	Functional Requirement(Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Bin inventory	The Dashboard shows data on the amount of fill in bins as it is being tracked by smart sensors. The application also forecasts when the bin will be full based on past data, which is one of the capabilities that even the greatest waste management software does not offer. As picks are also recognised by the sensors, you can determine when the bin was last emptied. You can get rid of the overflowing bins and cease collecting halfempty ones with

		real-time data and predictions
FR-2	Bin inventory	On the map, you can see every monitored bin and stand, and you can use Google Street View at any time to visit them. On the map, bins or stands appear as green, orange, or red circles. The Dashboard displays information about each bin, including its capacity, trash kind, most recent measurement, GPS location, and pickup schedule.
FR-3	Optimize the route to collect	Route planning for rubbish pickup is semi-automated using the tool. You are prepared to act and arrange for garbage collection based on the levels of bin fill that are now present and forecasts of approaching capacity. To find any discrepancies, compare the planned and actual routes

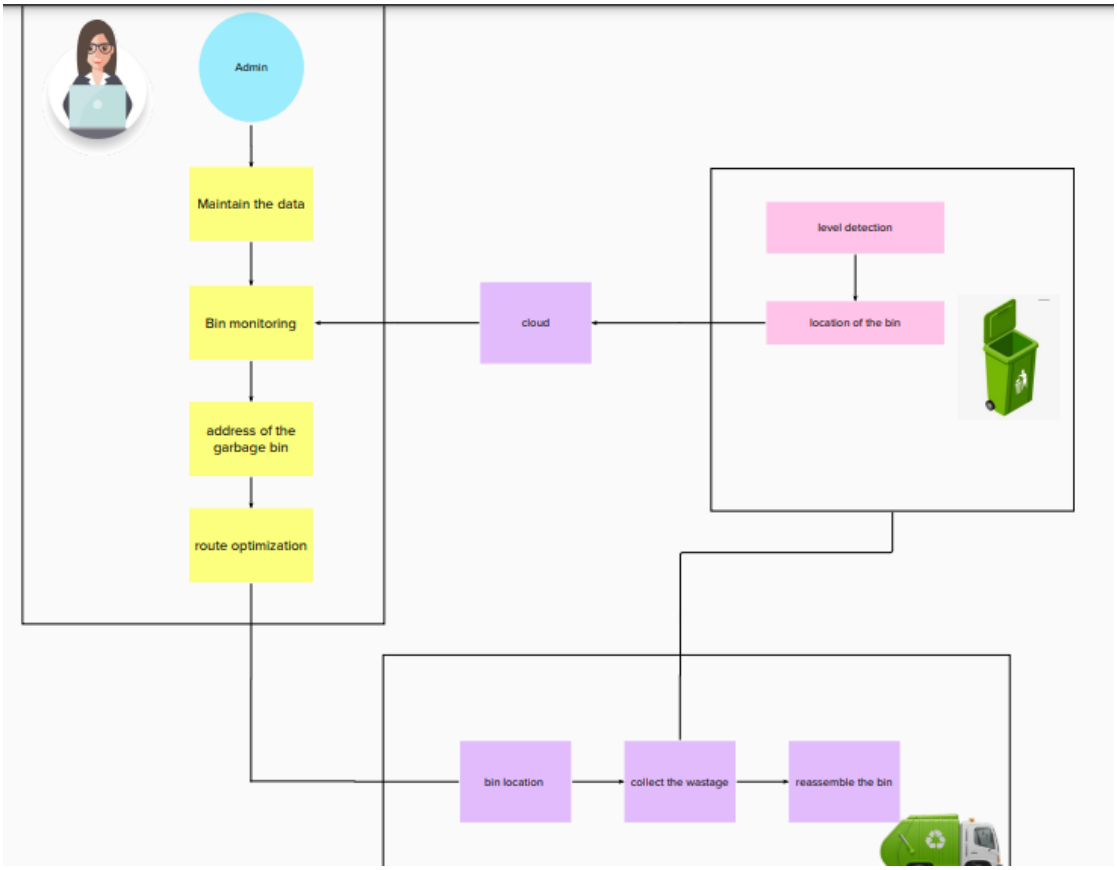
4.2 NON FUNCTIONAL REQUIREMENTS

NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Usability is a unique and significant perspective to examine user requirements, which can further enhance the design quality, according to IoT devices. The study of customers' product usability can help designers better

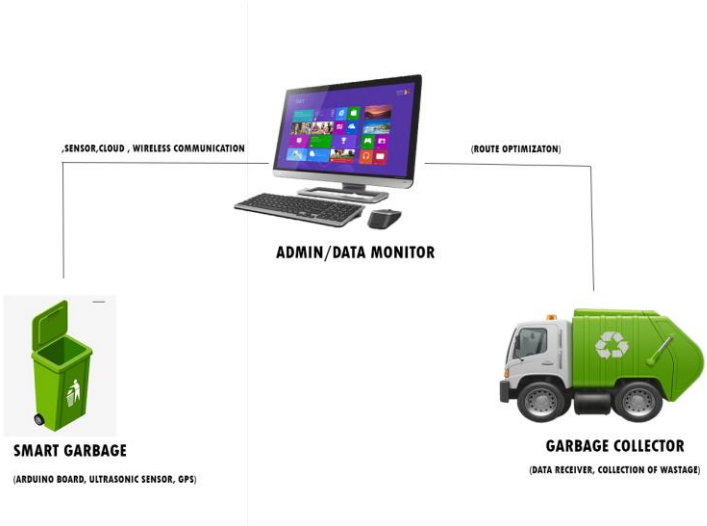
		understand users' possible demands in waste management, behaviour, and experience during the design process, which places a focus on the user experience
NFR-2	Reliability	Creating better working conditions for waste collectors and drivers is another aspect of smart waste management. Waste collectors will use their time more effectively by attending to bins that require service rather than travelling the same collection routes and servicing empty bins
NFR-3	Performance	The Smart Sensors assess the fill levels in bins along with other data numerous times per day using ultrasound technology. The sensors feed data to Smart Waste Management Software System, a robust cloud -based platform with data - driven daily operations and a waste management app, using a variety of IoT networks. As a result, customers receive data - driven decision - making services, and waste collection routes, frequency, and vehicle loads are optimised, resulting in at least a 30% route reduction.
NFR-4	Scalability	We can add more bins into this system

5. PROJECT DESIGN

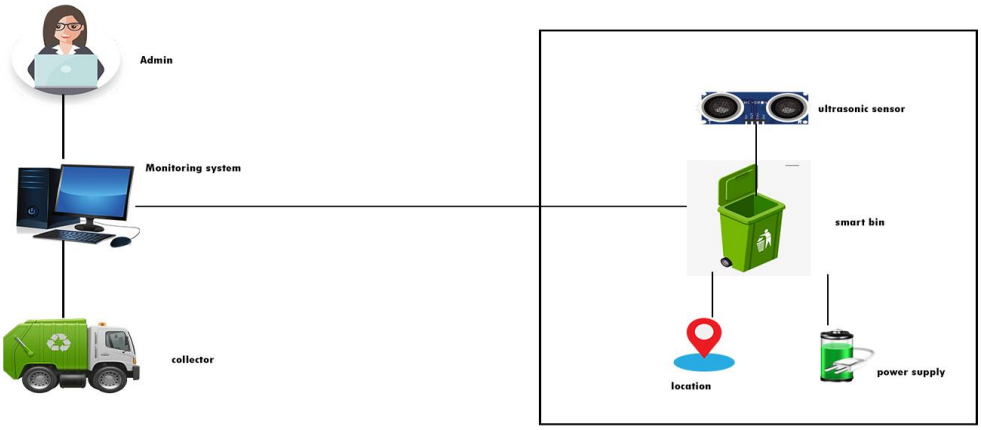
5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE



SMART WASTE MANAGEMENT SYSTEM USING IOT



TEAM ID : PNT2022TMID40268

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Numbe	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2

		USN-4	As a user, I can register for the application through Gmail		Mediu m	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
	Customer (Web user)					

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATING

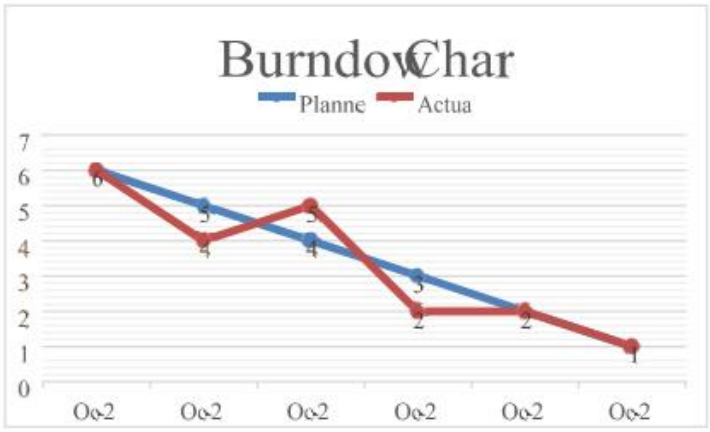
Sprint	Function al Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a Administrator, I need to give user id and passcode for ever workers over there in municipality	10	High	MEMBER 1
Sprint-1	Login	USN-2	As a Co-Admin, I'll control the waste level by monitoring them vai real time web portal. Once the filling happens, I'll notify trash truck with location of bin with bin ID	10	High	MEMBER 2

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	MEMBER 3
Sprint-3	Dashboard	USN-4	As a Local Garbage Collector, I'll gather all the waste from the garbage, load it onto a garbage truck, and deliver it to Landfills	20	Medium	MEMBER 4
Sprint-4	Dashboard	USN-5	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems	20	High	MEMBER 5

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	19 Nov 2022

6.3 REPORTS FROM JIRA



B3 24-10-2022

	A	B	C	D	E	F
1	Time		Tasks			
2	Day	Dates	Planned	Actual		
3	Mon	Oct-22	6	6		
4	Tue	Oct-22	5	4		
5	Wed	Oct-22	4	5		
6	Thurs	Oct-22	3	2		
7	Friday	Oct-22	2	2		
8	Saturday	Oct-22	1	1		
9						
10						

7. CODING & SOLUTIONING
(Explain the features added in the project along with code)

7.1 feature 1

```
#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 "siala1" // IBM organisation id
#define DEVICE_TYPE "SmartBin" // Device type
mentioned in ibm watson iot platform
#define DEVICE_ID "2901" // Device ID mentioned in
ibm watson iot platform
#define TOKEN "IBMproject" // 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;
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");
```

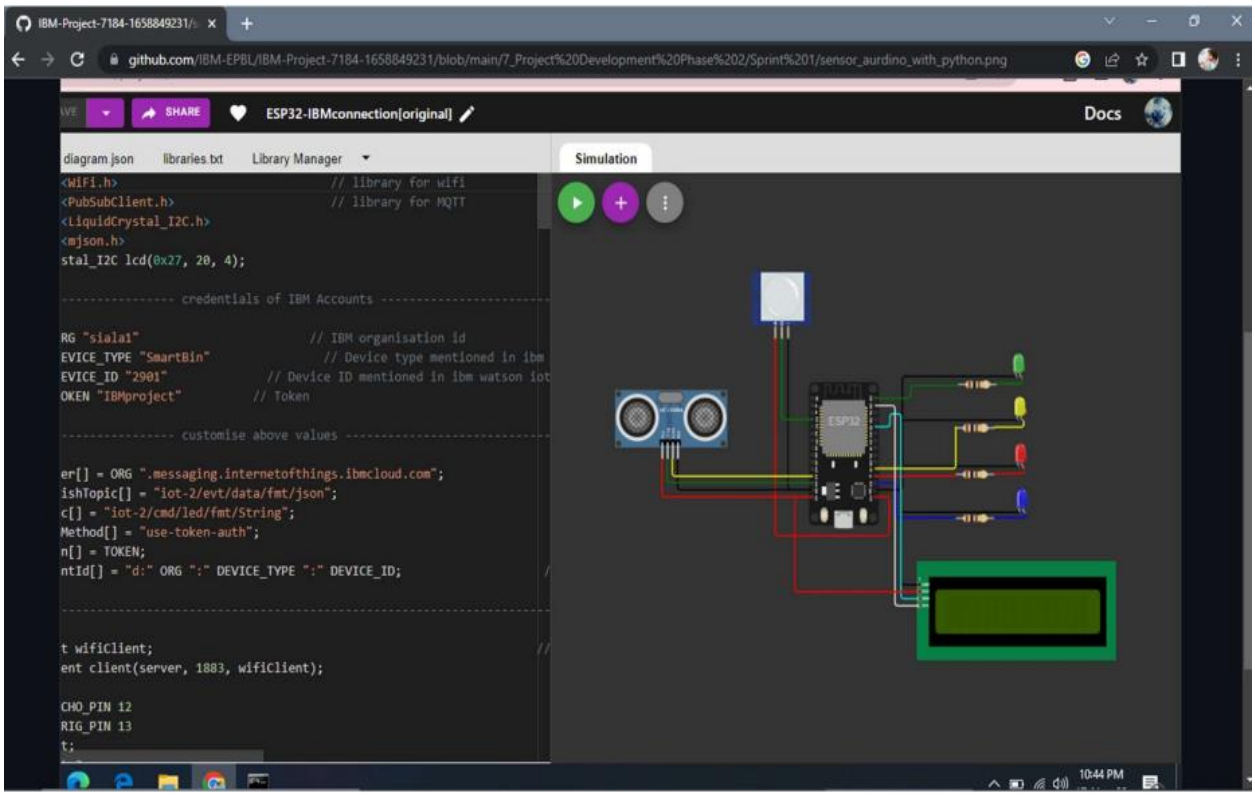
```
Serial.println("Lid Opened");
digitalWrite(15, HIGH);



if(digitalRead(34)== true)
{
if(cm <= 60) //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 > 60 && cm < 120)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of
bin
level");
digitalWrite(2, LOW);
digitalWrite(23, LOW);

}
else if(cm > 120)
{
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 <= 60)
{
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 <= 120)
```

```
{
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
{
digitalWrite(23,HIGH);
String payload = "{\"Safe\":";
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);
```

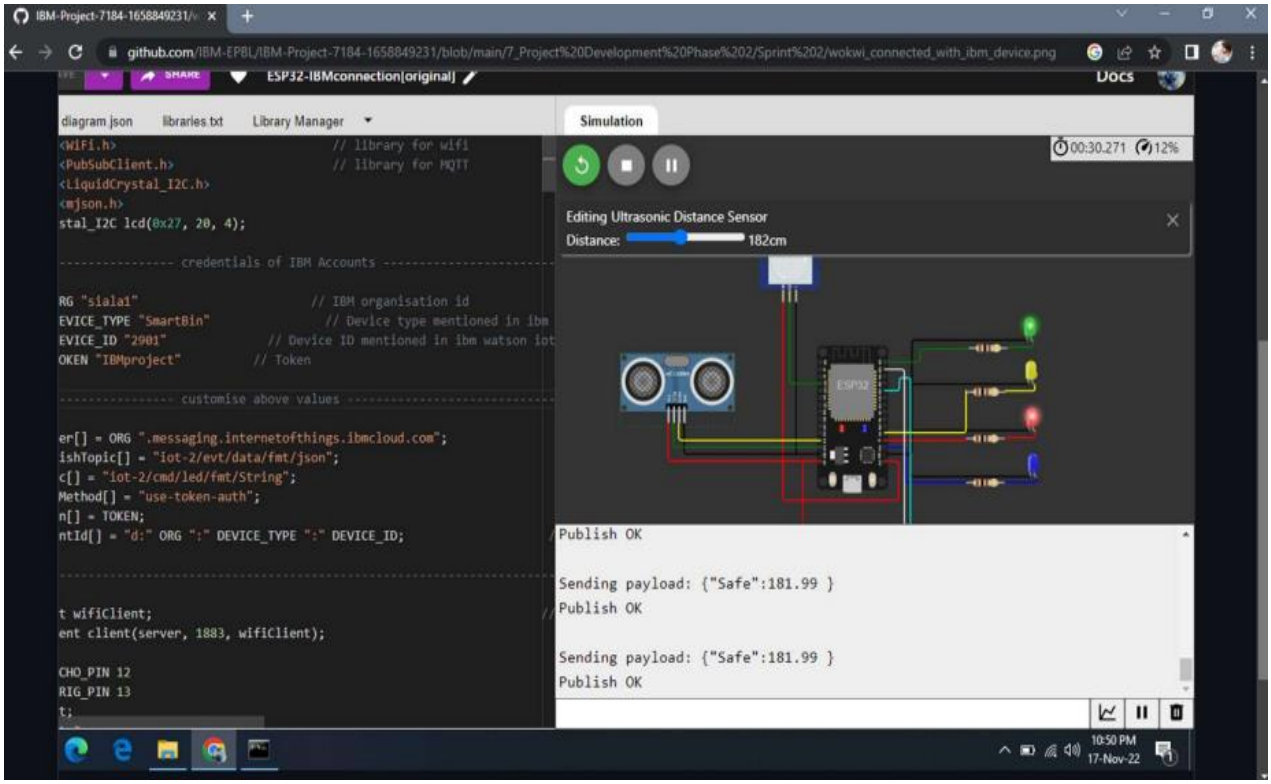
```
 lcd.print("Inches");
 lcd.setCursor(4,0);
 lcd.setCursor(12,0);
 lcd.print("cm");
 lcd.setCursor(1,1);
 lcd.print(inches,  1);
 lcd.setCursor(11,1);
 lcd.print(cm,  1);
 lcd.setCursor(14,1);
 delay(1000);
 lcd.clear();
}
//handles commands from user side
void callback(char* subscribetopic, byte* 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=="\"Seal  Bin\"")
{
Serial.println("Sealed");

}


}
data3="";
}
```

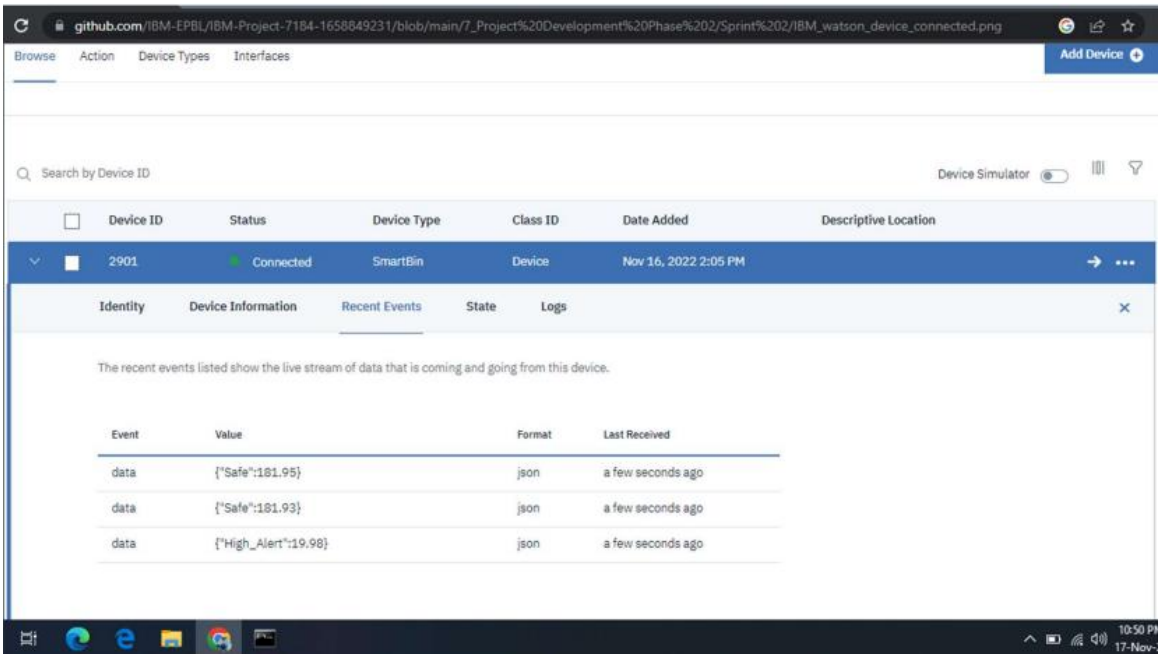


Sensor-ardunio connections

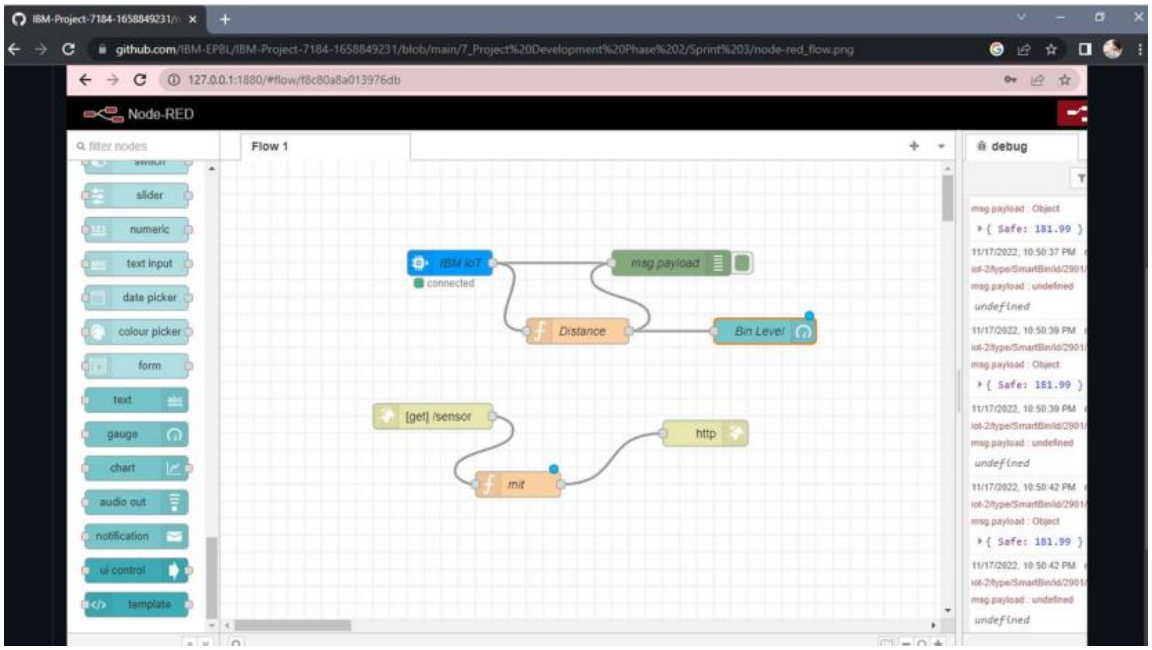
7.2 Feature 2



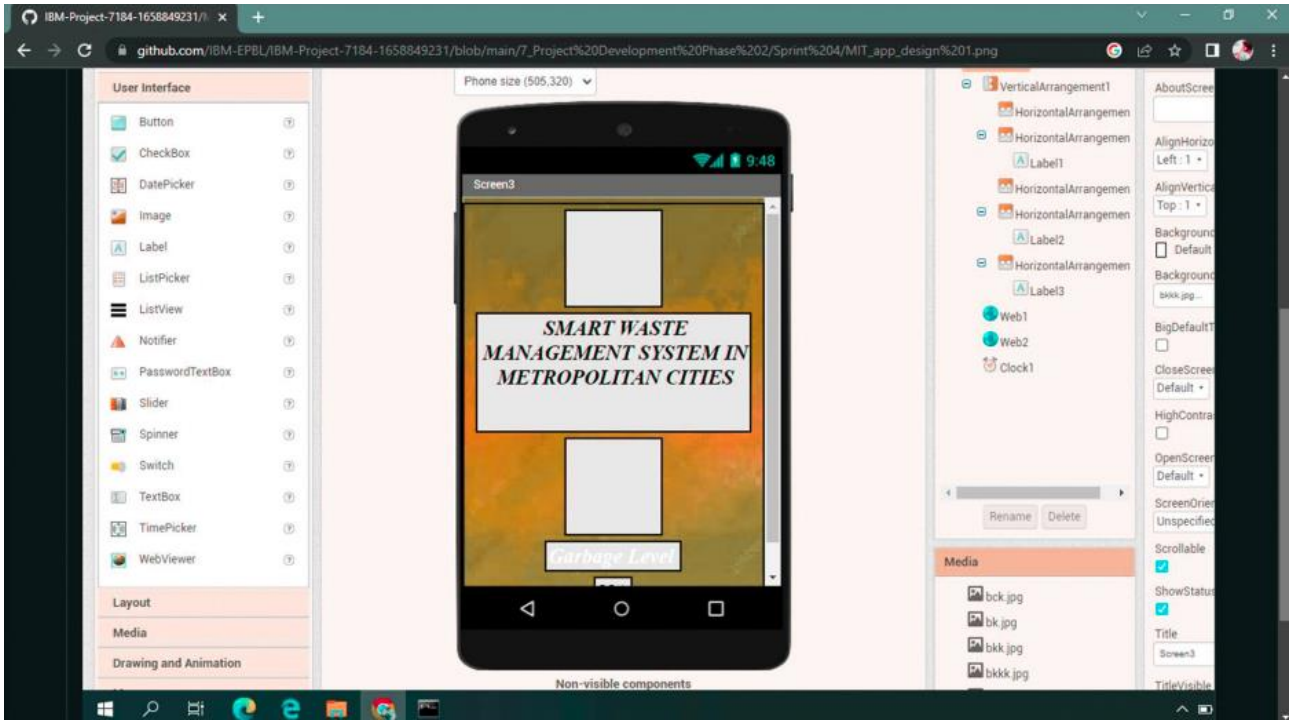
WOWKI OUTPUT



IBM Watson Device connected



Nod red connections



MIT App

8. TESTING

8.1 TESTING CASES

Component	Test Case Scenario	Pre-Requisite	Availability	Test Condition	Expected Result	Actual Result	Status	Comments	Accessed B
Ultrasonic Sensor	When Bin is empty	Ultrasoncic sensor PIR Motion Sensor Garbage Bins	Bin is accessible to users	Bin Level == 0	Displays Bin level and space left	Working as expected	Pass		User
Ultrasonic Sensor	When bin level is below 50 %	Ultrasoncic sensor , PIR Motion Sensor , Garbage Bins	Bin is accessible to users	Bin Level < 50	Displays Bin level and space left	Working as expected	Pass		User
Ultrasonic Sensor	When bin level is above 50	Ultrasoncic sensor , PIR Motion sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level > 50	Displays Bin level and space left	Working as expected	Pass		User
Ultrasonic Sensor	When bin level is below 75 %	Ultrasoncic sensor , PIR Motion sensor , Garbage Bins	Bin is accessible to users and the admin gets warning about the bin level	Bin Level < 75	Displays Bin level and space left	Working as expected	Pass		User
Ultrasonic Sensor	When bin level is above 75 %	Ultrasoncic sensor , PIR Motion sensor , Garbage Bins	Bin is not accessible to the users, the admin recieves High alert and seals the the bin to avoid overflow.	Bin Level > 75	Displays Bin is FULL and Seals the bin.	Working as expected	Pass	The system starts to sense the level once the Bin is emptied partially or fully	User/Admi

8.2 User Acceptance Testing

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	7	0	0	7
Client Application	51	0	0	51

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

9. RESULTS

9.1 Performance results

The smart waste management solution to make the waste collection process more efficient is sensors. Sensors can measure the fill level of the containers and provide updated information at any time and notify waste management services to empty them when they are full or almost full. These devices help optimize the best possible route containing fully filled containers and create smart schedules for drivers.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES

➨It saves time and money by using smart waste collection bins and systems equipped with fill level sensors. As smart transport vehicles go only to the filled containers or bins. It reduces infrastructure, operating and maintenance costs by up to 30%.

➨It decreases traffic flow and consecutively noise due to less air pollution as result of less waste collection vehicles on the roads. This has become possible due to two way communication between smart dustbins and service operators.

➨It keeps our surroundings clean and green and free from bad odour of wastes, emphasizes on healthy environment and keep cities more beautiful.

➨It further reduces manpower requirements to handle the garbage collection process.

DISADVANTAGES

➨System requires more 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.

➨Wireless technologies used in the system such as zigbee and Wi-Fi have shorter range and lower data speed. In RFID based systems, RFID tags are affected by surrounding metal objects (if any).

➨It reduces man power requirements which results into increase in unemployment for unskilled people.

➨The training has to be provided to the people involved in the smart waste management system.

11. CONCLUSION

The behavior of generating garbage is too dangerous not only for today's generation, but also for future generations. It is critical to educate people and encourage them to practice recycle,reuse and reduse instead of producing waste. Waste disposal should be a priority for municipalities and governments.

12. FUTURE SCOPE

In this report, smart bin is built on a microcontroller based platform ArdunioUno board, which is interfaced with ultrasonic sensor. It will stop overflowing of dustbins along roadsides and localities as smart Dustbins are managed.

13. APPENDIX

SOURCE CODE:

```
#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 "siala1" // IBM organisation id
#define DEVICE_TYPE "SmartBin" // Device type
mentioned in ibm watson iot platform
#define DEVICE_ID "2901" // Device ID mentioned in
ibm watson iot platform
#define TOKEN "IBMproject" // 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;
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 <= 60) //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 > 60 && cm < 120)
{
digitalWrite(4, HIGH);
Serial.println("Warning!!,Trash is about to cross 50% of bin
level");
digitalWrite(2, LOW);
digitalWrite(23, LOW);

}
else if(cm > 120)
{
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 <= 60)
{
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 <= 120)
{
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
{
digitalWrite(23,HIGH);
String payload = "{\"Safe\":";
```

```
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=="\"Seal Bin\"")
{
Serial.println("Sealed");


}



}
data3="";
}
```

GITHUP LINK- https://github.com/IBM-EPBL/IBM-Project-29191-1660121920

Project demo link – _____