Project Report

1. INTRODUCTION

- Smart waste management system in metropolitan cities is about using technology and data to create a more efficient waste industry based on IOT technology

1.1 Project Overview

- The project is based on a real-time smart garbage bin mechanism for solid waste management in smart cities

1.2 Purpose

- A waste management system is the strategy an organization uses to dispose of, reduce, reuse and prevent waste

2. LITERATURE SURVEY

2.1 Existing problem

Heavy metals and other toxic compounds from landfills, pollution.

2.2 References

1. M.C. Xenia, E. D’souza, K. D. Woelorm, R. Nii Adjei-Laryea, and E. Baah-Nyarkoh, "A Proposed IoT Based Smart Waste Bin Management System with An Optimized Route: A Case Study of Ghana", *2020 Conference on Information Communications Technology and Society (ICTAS)*, pp. 1-5, 2020.

2. Z. Hisham Che Soh, M. Azeer Al-Hami Husa, S. Afzal Che Abdulla,h and M. Affandi Shafie, "Smart Waste Collection Mon-itoring and Alert System via IoT", *2019 IEEE 9th Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 50-54, 2019.

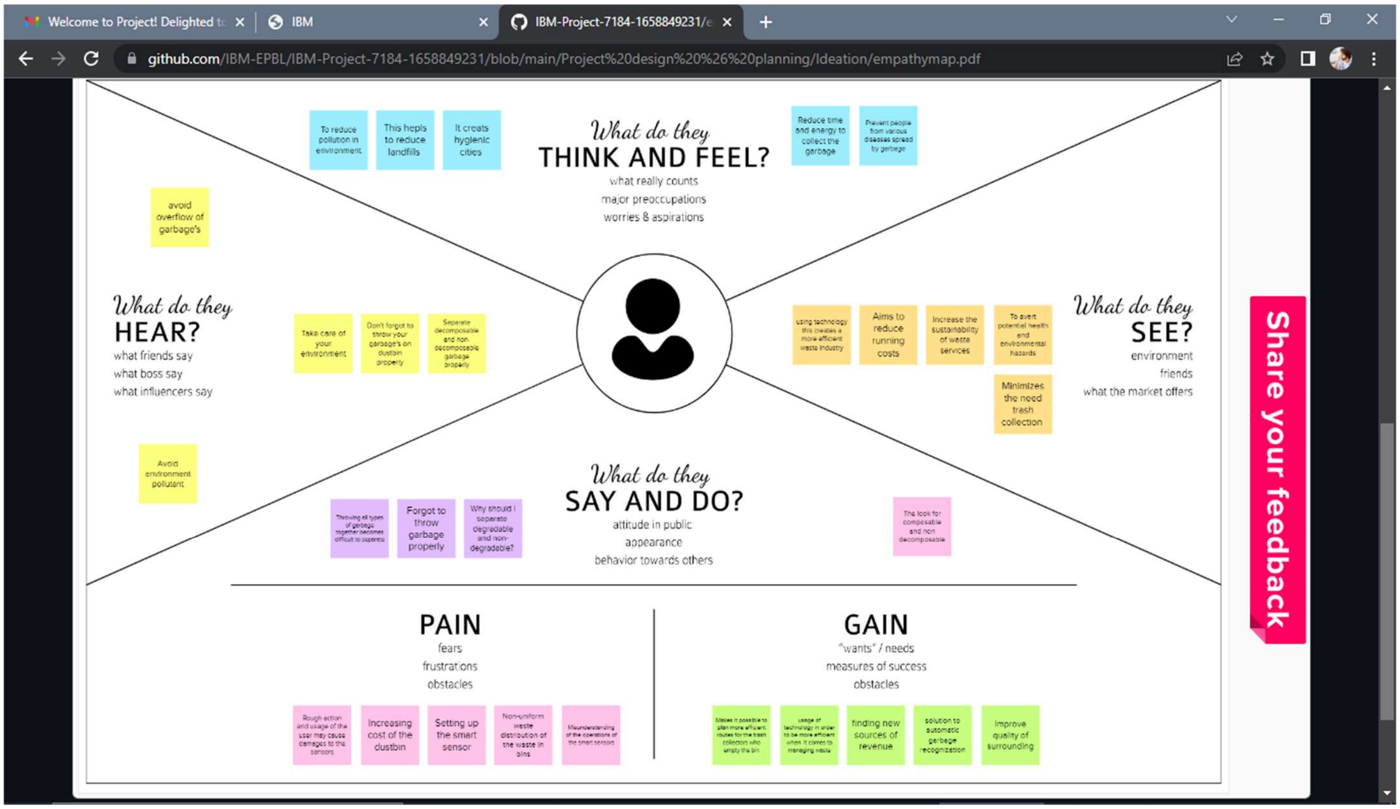
3. E. Al-Masri, I. Diabate, R. Jain, M. H. Lam, and S. Reddy Nathala, "Recycle.io: An IoT-Enabled Framework for Urban Waste Management", *2018 IEEE International Conference on Big Data (Big Data)*, pp. 5285-5287, 2018.

2.3. Problem Statement Definition

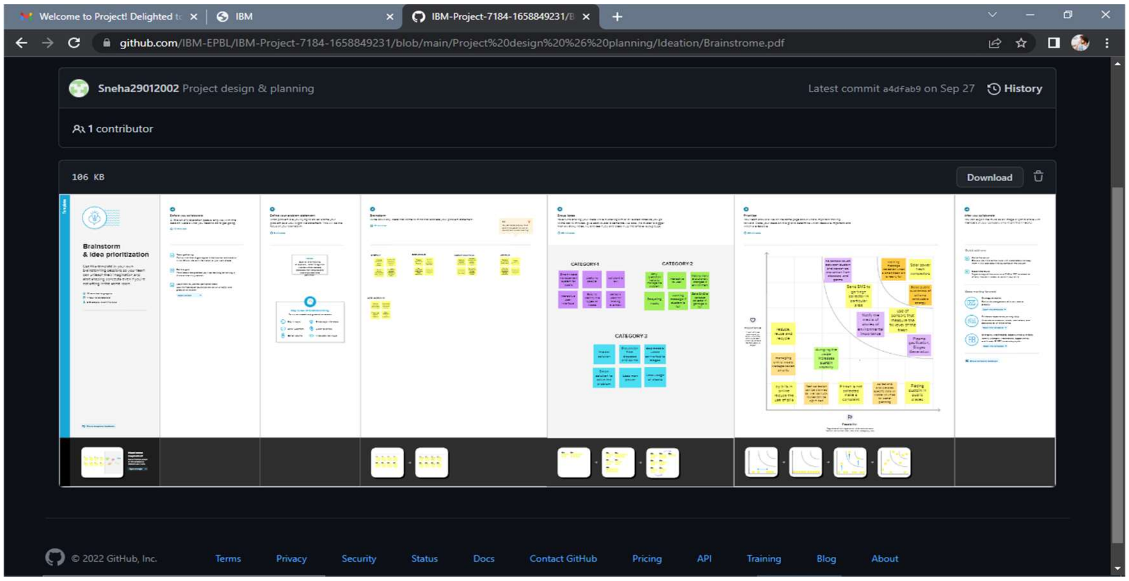
- Indiscriminate disposal of waste is a major issue in most developing countries' urban centers and poses a serious threat to the healthy living of the citizens. The fill level of waste in each of the containers, which are strategically situated across the communities, is detected using sensors.

3. IDEATION & PROPOSED SOLUTION

3.1. Empathy Map Canvas


Welcome to Project! Delighted tc
IBM
IBM-Project-7184-1658849231/e
github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/Project%20design%20%26%20planning/Ideation/empathymap.pdf
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
To reduce pollution in environment
This hepls to reduce landfills
It creats hyglenic cities
Reduce time and energy to collect the garbage
Prevent people from various diseases spread by gerbage
What do they
HEAR?
what friends say
what boss say
what influencers say
avoid overflow of garbage's
Take care of your environment
Don't forgot to throw your garbage's on dustbin properly
Seperete decomposable and non-decomposable garbage properly
Avoid environment pollutant
What do they
SEE?
environment
friends
what the market offers
using technology this creates a more efficient waste Industry
Aims to reduce running costs
Increase the sustainability of waste services
To avert potential health and environmental hazards
Minimizes the need trash collection
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
Throwing all types of garbage together becomes difficult to saperate
Forgot to throw garbage properly
Why should I separate degradable and non-degradable?
The look for composable and non decomposable
PAIN
fears
frustrations
obstacles
Rough action and usage of the user may cause damages to the sensors
Increasing cost of the dustbin
Setting up the smart sensor
Non-uniform waste distribution of the waste in bins
Misunderstanding of the operations of the smart sensors
GAIN
"wants" / needs
measures of success
obstacles
Makes it possible to plan more efficient routes for the trash collectors who empty the bin
usage of technology in order to be more efficient when it comes to managing waste
finding new sources of revenue
solution to automatic gerbage recognization
improve quality of surrounding
Share your feedback

3. 2. *Ideation & Brainstorming*


Welcome to Project! Delighted t
IBM
IBM-Project-7184-1658849231/
github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/Project%20design%20%26%20planning/Ideation/Brainstrome.pdf
Sneha29012002 Project design & planning
Latest commit a4dfab9 on Sep 27
History
1 contributor
106 KB
Download
Brainstorm & idea prioritization
CATEGORY 1
CATEGORY 2
CATEGORY 3
© 2022 GitHub, Inc.
Terms
Privacy
Security
Status
Docs
Contact GitHub
Pricing
API
Training
Blog
About

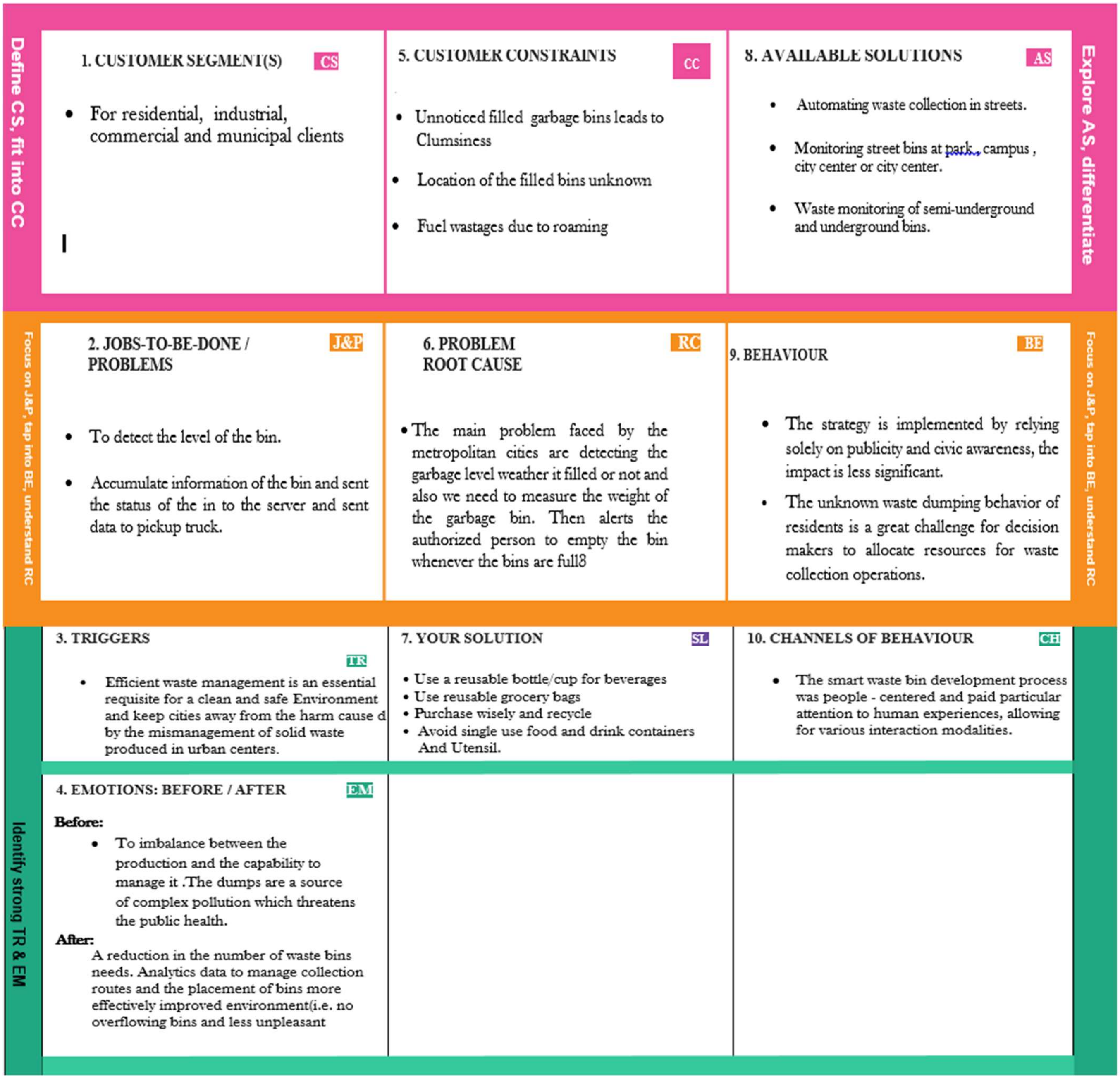
3. 3. *Proposed Solution*

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement(problem to be solved)	Indiscrimi issue in most developing countries' urban center sand most developing countries and it poses a serious threat to 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.

2	Idea / Solution description	Reduce, Recycle & Reuse – Recycling not only saves energy but also prevents the materials from going to landfills & incineration and provides raw materials for new products. Installing more bins for collecting recycling like paper ,g lass ,plastics , etc.
3	Novelty / Uniqueness	Raise public awareness of utilizing renewable energy, improves street sanitation, encourage recycling , collect and analyze area-specific data on waste volumes for better planning .
4	Social Impact / Customer Satisfaction	Impacts of present waste disposal practices are seen direct social impacts such as neighborhood of landfills to communities, breeding of pests and loss in property values. Waste also represents an economic loss and burden to our society.
5	Business Model (Revenue Model)	Waste management generate revenue through the provision of various waste management and disposal services and recycling solutions to residential,

		commercial, industrial and municipal cl,ients.
6	Scalability of the Solution	The implementation of our solution helps to promote separation by residents. With sufficient, capacity, there is always enough space for people’s trash-general and separable. The need-driven waste collection eliminates unnecessary traffic blockage and overflowing bins. Cities, as a result, becomes cleaner and free or litter, trash garbage and solutions like recycling are promoted

3.4. *Problem Solution fit*



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

- For residential, industrial, commercial and municipal clients

I

5. CUSTOMER CONSTRAINTS CC

- Unnoticed filled garbage bins leads to Clumsiness
- Location of the filled bins unknown
- Fuel wastages due to roaming

8. AVAILABLE SOLUTIONS AS

- Automating waste collection in streets.
- Monitoring street bins at park , campus , city center or city center.
- Waste monitoring of semi-underground and underground bins.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- To detect the level of the bin.
- Accumulate information of the bin and sent the status of the in to the server and sent data to pickup truck.

6. PROBLEM ROOT CAUSE RC

- The main problem faced by the metropolitan cities are detecting the garbage level weather it filled or not and also we need to measure the weight of the garbage bin. Then alerts the authorized person to empty the bin whenever the bins are full8

9. BEHAVIOUR BE

- The strategy is implemented by relying solely on publicity and civic awareness, the impact is less significant.
- The unknown waste dumping behavior of residents is a great challenge for decision makers to allocate resources for waste collection operations.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

- Efficient waste management is an essential requisite for a clean and safe Environment and keep cities away from the harm cause d by the mismanagement of solid waste produced in urban centers.

7. YOUR SOLUTION SL

- Use a reusable bottle/cup for beverages
- Use reusable grocery bags
- Purchase wisely and recycle
- Avoid single use food and drink containers And Utensil.

10. CHANNELS OF BEHAVIOUR CH

- The smart waste bin development process was people - centered and paid particular attention to human experiences, allowing for various interaction modalities.

4. EMOTIONS: BEFORE / AFTER EM

Before:

- To imbalance between the production and the capability to manage it .The dumps are a source of complex pollution which threatens the public health.

After:

A reduction in the number of waste bins needs. Analytics data to manage collection routes and the placement of bins more effectively improved environment(i.e. no overflowing bins and less unpleasant

4. REQUIREMENT ANALYSIS

4.1. Functional requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Detailed bin inventory.	All monitored bins and stands can be seen on the map, and you can visit them at any time via the Street View feature from Google. Bins or stands are visible on the map as green, orange or red circles. You can see bin details in the Dashboard – capacity, waste type, last measurement, GPS location and collection schedule or pick recognition.
FR-2	Real time bin monitoring.	The Dashboard displays real-time data on fill-levels of bins monitored by smart sensors. In addition to the % of fill-level, based on the historical data, the tool predicts when the bin will become full, one of the functionalities that are not included even in the best waste management software.. Sensors recognize picks as well; so you can check when the bin was last collected. With real-time data and predictions, you can eliminate the overflowing bins and stop collecting half-empty ones.

FR-3	Expensive bins.	We help you identify bins that drive up your collection costs. The tool calculates a rating for each bin in terms of collection costs. The tool considers the average distance depo - bin discharge in the area. The tool assigns bin a rating (110) 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.	Eliminate the collection of half-empty bins. The sensors recognize picks. By using real-time data on fill-levels and pick recognition, we can show you how full the bins you collect are. The report shows how full the bin was when picked. You immediately see any inefficient picks below 80% full.
FR-6	Plan waste collection routes.	The tool semi-automates waste collection route planning. Based on current bin fill-levels and predictions of reaching full capacity, you are ready to respond and schedule waste collection. You can compare planned vs. executed routes to identify any inconsistencies

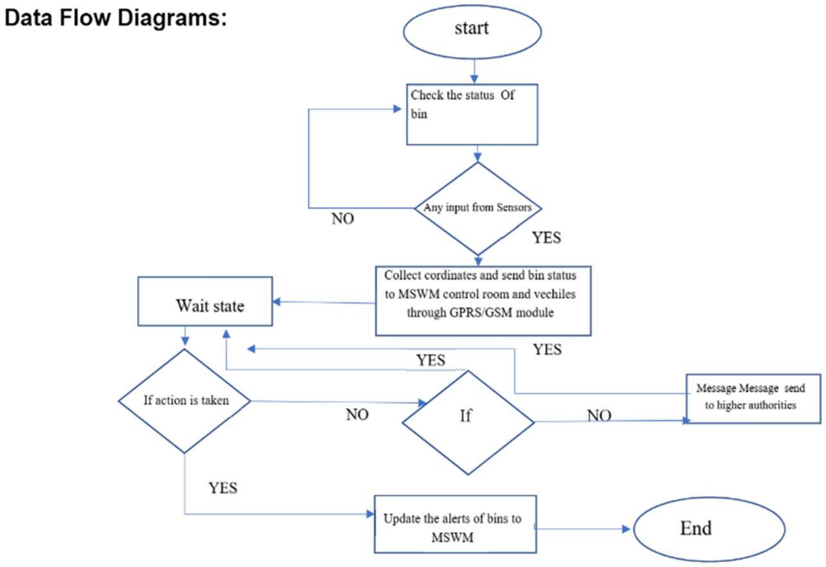
4. 2. Non-Functional requirements

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 Avoid single use food and drink containers.
NFR-3	Reliability	Smart waste management is also about creating better working conditions for waste collectors and drivers. Instead of driving the same collection routes and servicing empty bins, waste collectors will spend their time more efficiently, taking care of bins that need servicing.

NFR-4	Performance	The Smart Sensors use ultrasound technology to measure the fill levels (along with other data) in bins several times a day. Using a variety of IOT networks ((NB- IoT , GPRS), the sensors send the data to Sensor's Smart Waste Management Software System, a powerful cloud-based platform , for data driven daily operations, available also as a waste management app. Customers are hence provided data-driven decision making, and optimization of waste collection routes, Frequencies , and vehicle loads resulting in route reduction by at least 30%.

5. PROJECT DESIGN

5.1. Data Flow Diagrams


Data Flow Diagrams:
start
Check the status Of bin
Any input from Sensors
NO
YES
Collect cordinates and send bin status to MSWM control room and vechiles through GPRS/GSM module
Wait state
YES
YES
If action is taken
NO
If
NO
Message Message send to higher authorities
YES
Update the alerts of bins to MSWM
End

5.2. Solution & Technical Architecture

Technical Architecture:

Table-1 : Components & Technologies:

S .No	Component	Description	Technology
1.	User Interface	Web Portal	HTML ,CSS , NodeRed , JAVA SCRIPT or on
2.	Application Logic-1	To calculate the distance of dreck and show the real time level in web portal. Information getting via ultrasonic sensor and the alert message activate with python script to web portal.	Ultrasonic sensor/ Python.
3.	Application Logic-2	To calculate the weight of the garbage and show the real time weight in web portal, this info getting via load cell and the alert message activate with python to web portal.	Load cell/Python.
4.	Application Logic-3	Getting location of the Garbage.	GSM / GPS.
5.	Cloud Database.	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
6.	File Storage	File storage requirements	Git hub , Local file system.
7.	External API1.	Firebase is a set of hosting services for any type of	Firebase.

		Application. It offers NoSQL and real-time hosting of databases, content, social authentication, and notifications, or services, such as a real-time communication server.	
8.	Ultrasonic Sensor.	To throw alert message when garbage is getting full.	Distance Recognition Model.
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: localhost Cloud Server Configuration: localhost, Firebase.	Localhost, Web portal.

Table-2: Application Characteristics:

S. No	Characteristics	Description	Technology
1.	Open-Source Frameworks	NodeRed, Python, IBM Simulator.	IoT
2.	Security Implementations	Raspberry Pi is connected to the internet and for example used to broadcast live data, further security measures are recommended and use the UFW(uncomplicated Firewall).	IoT

3.	Scalable Architecture	Raspberry pi: Specifications Soc: rspi ZERO W CPU: 32-bit computer with a 1 GHz ARMv6 RAM: 512MB Networking: Wi-Fi Bluetooth: Bluetooth 5.0, Bluetooth Low Energy (BLE). Storage: MicroSD GPIO: 40-pin GPIO header, populated	IoT
S. No	Characteristics	Description	Technology
		Ports: micro HDMI 2.0, 3.5mm analogue audiovideo jack, 2x USB 2.0, 2x USB 3.0, Ethernet Dimensions: 88mm x 58mm x 19.5mm, 46g	
4.	Availability	These smart bins use sensors like ultrasonic and load cell to send alert message about the trash level recognition technology, and artificial intelligence, enabling them to automatically sort and categorize recycling litter into one of its smaller bin.	IoT.
5.	Performance	Number of request:RPI manages to execute 129139 read requests per second. Use of Cache:512mb Use of CDN's:Real time	IoT/Web portal.

5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	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		Medium	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)						
Customer Care Executive						
Administrator						

6. <u>PROJECT PLANNING & SCHEDULING</u>

6 1. Sprint Planning & Estimation

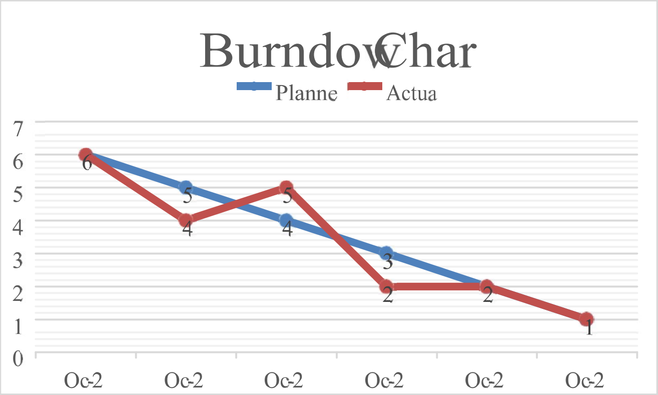
Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Login	USN-1	As a Administrator, I need to give user id and passcode for ever workers over there in municipality	10	High	MEMBER 1
Sprint-1	Login	USN-2	As a Co-Admin, I'll control the waste 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

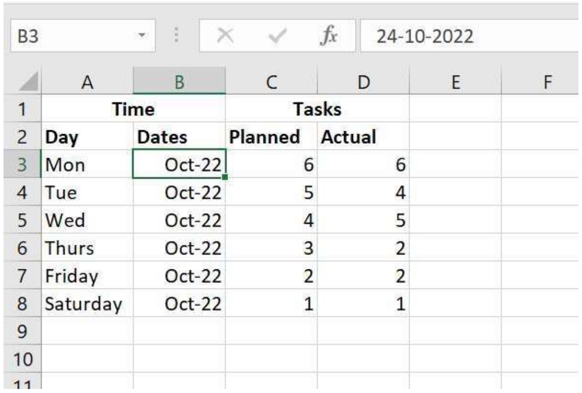
Sprint-3	Dashboard	USN-4	As a Local Garbage Collector, I’II gather all the waste from the garbage, load it onto a garbage truck, and deliver it to Landfills	20	Medium	MEMBER 4
Sprint-4	Dashboard	USN-5	As a Municipality officer, I'll make sure everything is proceeding as planned and without any problems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





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)

7.1Feature 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
```

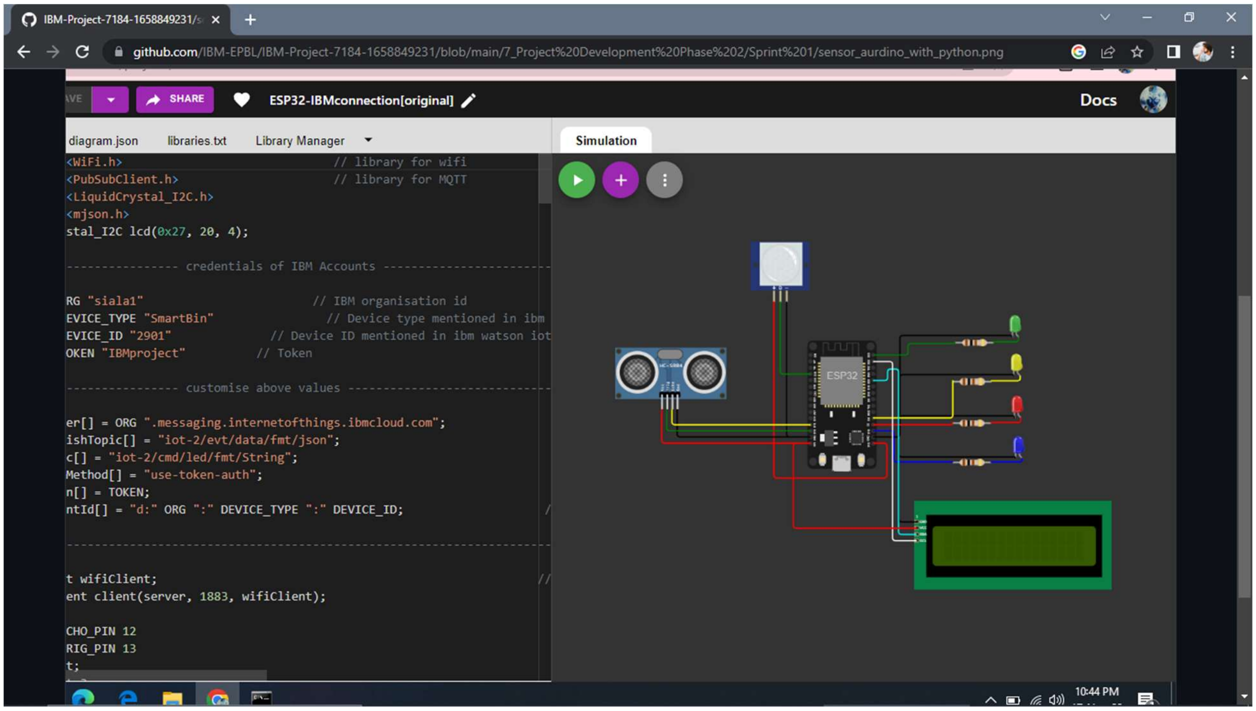
```
char authMethod[] = "use-token-auth";                                    //
authentication method
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;
//Client id

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

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);
```

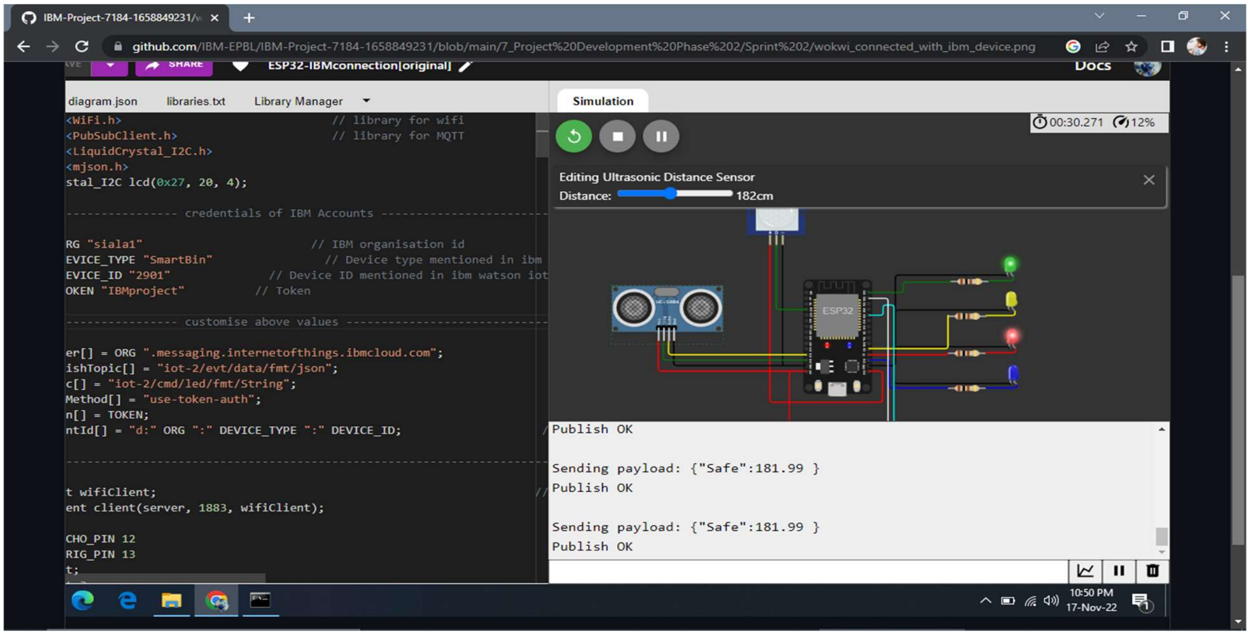
```
if (client.publish(publishTopic, (char*) payload.c_str()))        // if
data is uploaded to cloud successfully,prints publish ok else prints
publish failed
{
Serial.println("Publish OK");
}

}

  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="";
}
```


IBM-Project-7184-1658849231/s
github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/7_Project%20Development%20Phase%202/Sprint%201/sensor_aurdino_with_python.png
SHARE
ESP32-IBMconnection[original]
Docs
diagram.json
libraries.txt
Library Manager
Simulation
ESP32
10:44 PM

```
<WiFi.h>                               // library for wifi
<PubSubClient.h>                       // library for MQTT
<LiquidCrystal_I2C.h>
<mjson.h>
stal_I2C lcd(0x27, 20, 4);

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

RG "siala1"                         // IBM organisation id
EVICE_TYPE "SmartBin"                 // Device type mentioned in ibm
EVICE_ID "2901"                 // Device ID mentioned in ibm watson iot
OKEN "IBMproject"             // Token

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

er[] = ORG ".messaging.internetofthings.ibmcloud.com";
ishTopic[] = "iot-2/evt/data/fmt/json";
c[] = "iot-2/cmd/led/fmt/String";
Method[] = "use-token-auth";
n[] = TOKEN;
ntId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;                        /

---------------------------------------------------------------------


t wifiClient;                                                          //
ent client(server, 1883, wifiClient);

CHO_PIN 12
RIG_PIN 13
t;
```

Sensor-Arduino connections

7.2Feature 2


IBM-Project-7184-1658849231/v
github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/7_Project%20Development%20Phase%202/Sprint%202/wokwi_connected_with_ibm_device.png
SHARE
ESP32-IBMconnection[original]
Docs
diagram.json
libraries.txt
Library Manager
Simulation
00:30.271
12%
ESP32
10:50 PM
17-Nov-22

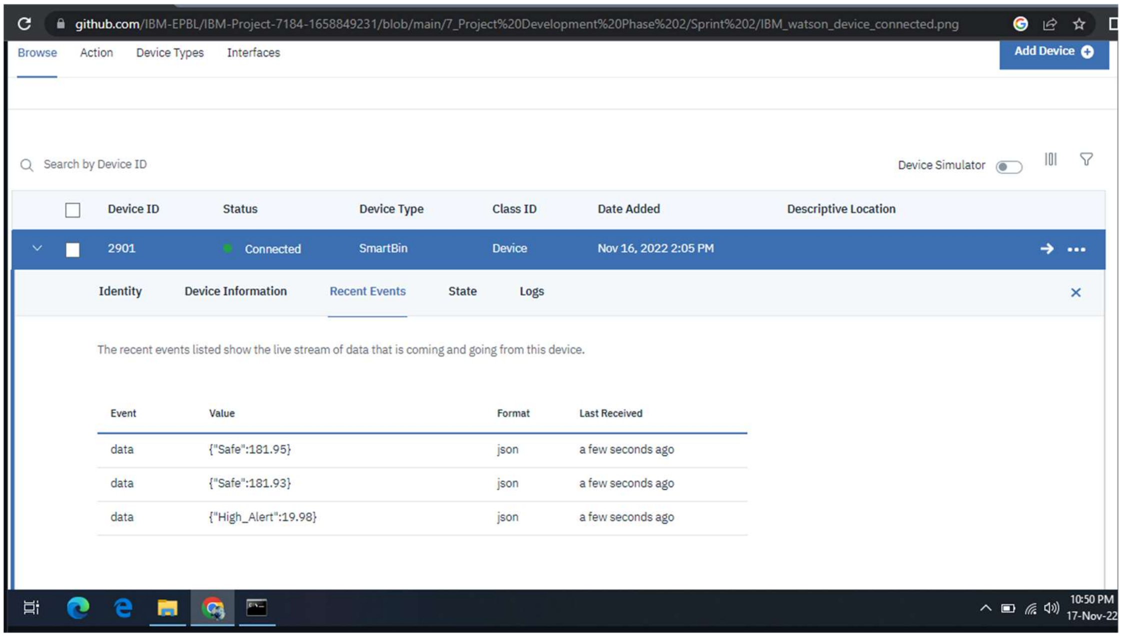
```
<WiFi.h>                               // library for wifi
<PubSubClient.h>                       // library for MQTT
<LiquidCrystal_I2C.h>
<mjson.h>
stal_I2C lcd(0x27, 20, 4);

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

RG "siala1"                         // IBM organisation id
EVICE_TYPE "SmartBin"                 // Device type mentioned in ibm
EVICE_ID "2901"                 // Device ID mentioned in ibm watson iot
OKEN "IBMproject"             // Token

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

er[] = ORG ".messaging.internetofthings.ibmcloud.com";
ishTopic[] = "iot-2/evt/data/fmt/json";
c[] = "iot-2/cmd/led/fmt/String";
Method[] = "use-token-auth";
n[] = TOKEN;
ntId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;                        /

---------------------------------------------------------------------


t wifiClient;                                                          //
ent client(server, 1883, wifiClient);

CHO_PIN 12
RIG_PIN 13
t;
```

Editing Ultrasonic Distance Sensor

Distance: 182cm

```
Publish OK

Sending payload: {"Safe":181.99 }
Publish OK

Sending payload: {"Safe":181.99 }
Publish OK
```

WOKWI output



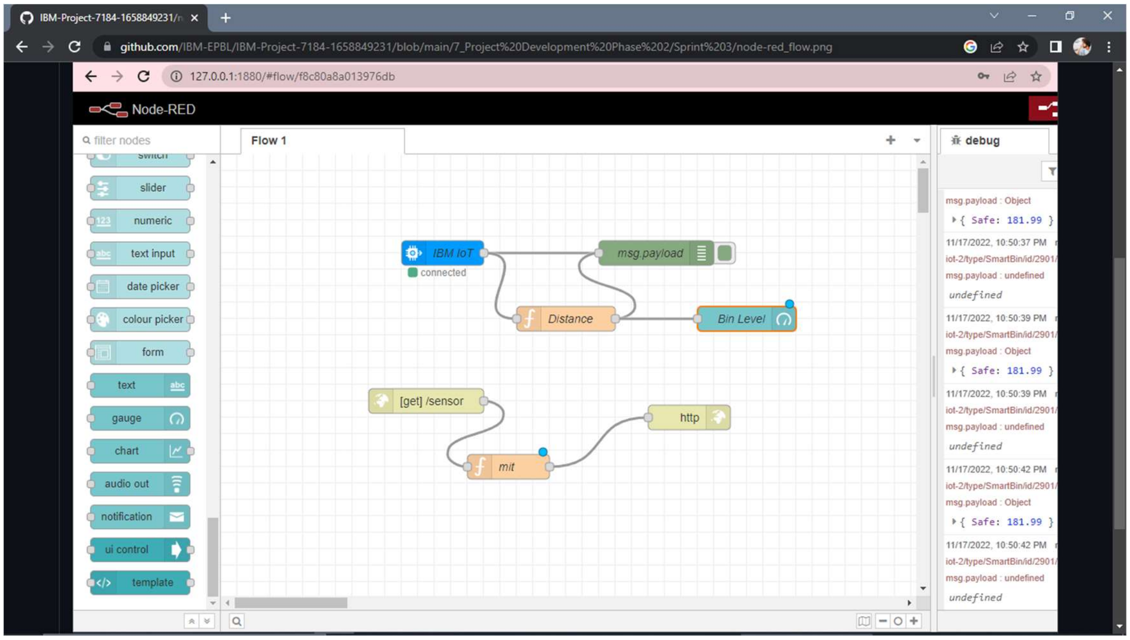
github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/7_Project%20Development%20Phase%202/Sprint%202/IBM_watson_device_connected.png

	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
□	2901	Connected	SmartBin	Device	Nov 16, 2022 2:05 PM	

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
data	{"Safe":181.95}	json	a few seconds ago
data	{"Safe":181.93}	json	a few seconds ago
data	{"High_Alert":19.98}	json	a few seconds ago

IBM Watson device connected



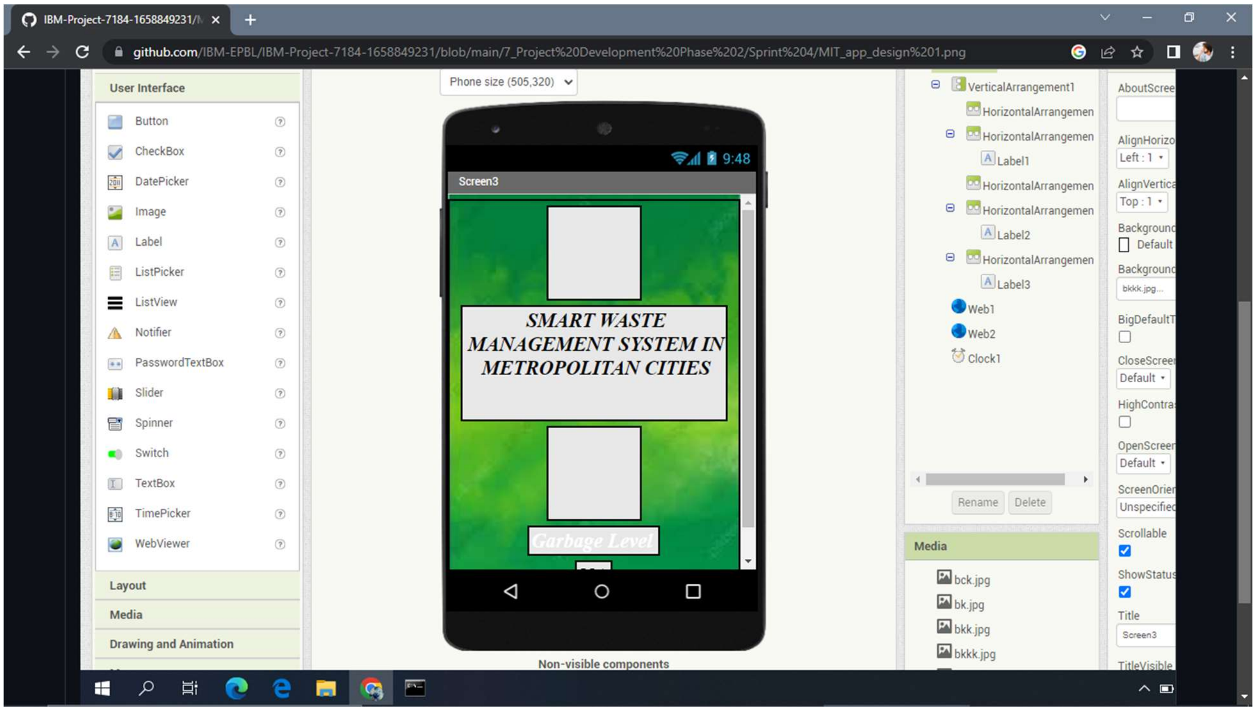
IBM-Project-7184-1658849231/n

github.com/IBM-EPBL/IBM-Project-7184-1658849231/blob/main/7_Project%20Development%20Phase%202/Sprint%203/node-red_flow.png

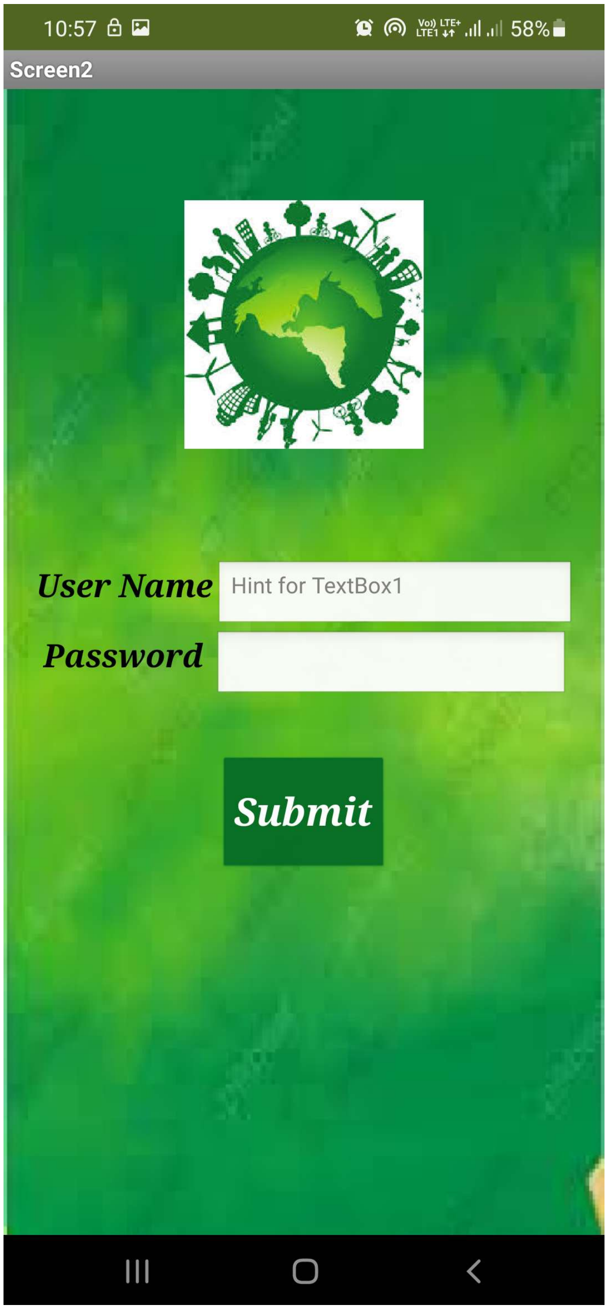
127.0.0.1:1880/#flow/f8c80a8a013976db

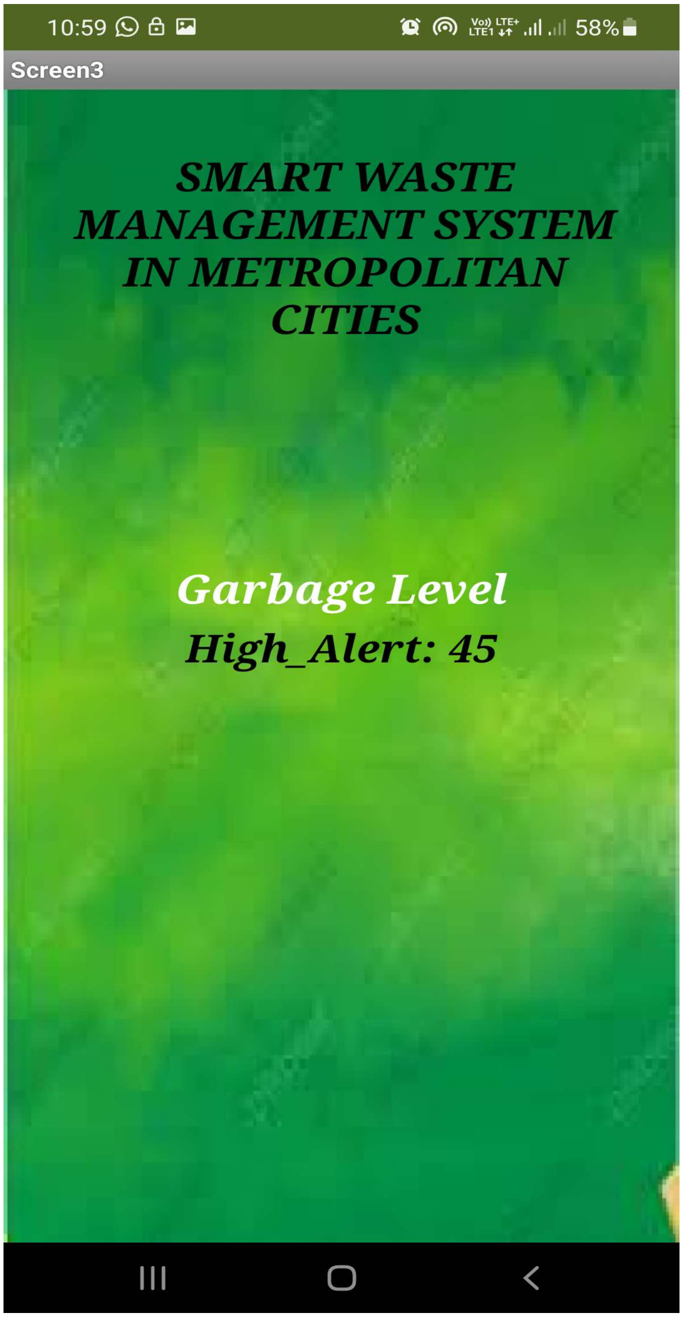
```
msg.payload : Object
▸{ Safe: 181.99 }
11/17/2022, 10:50:37 PM
iot-2/type/SmartBin/id/2901/
msg.payload : undefined
undefined
11/17/2022, 10:50:39 PM
iot-2/type/SmartBin/id/2901/
msg.payload : Object
▸{ Safe: 181.99 }
11/17/2022, 10:50:39 PM
iot-2/type/SmartBin/id/2901/
msg.payload : undefined
undefined
11/17/2022, 10:50:42 PM
iot-2/type/SmartBin/id/2901/
msg.payload : Object
▸{ Safe: 181.99 }
11/17/2022, 10:50:42 PM
iot-2/type/SmartBin/id/2901/
msg.payload : undefined
undefined
```

Node red connections



MIT app design





SMART WASTE MANAGEMENT SYSTEM IN METROPOLITAN CITIES

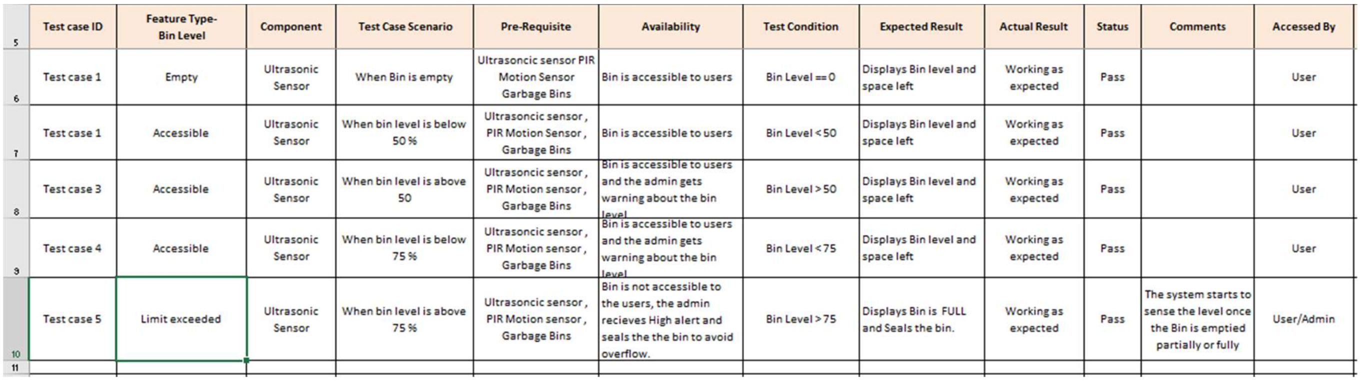
Garbage Level

High_Alert: 45

Value published in MIP

8. TESTING

8.1. Test Cases



Test case ID	Feature Type-Bin Level	Component	Test Case Scenario	Pre-Requisite	Availability	Test Condition	Expected Result	Actual Result	Status	Comments	Accessed By
Test case 1	Empty	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
Test case 1	Accessible	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
Test case 3	Accessible	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
Test case 4	Accessible	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
Test case 5	Limit exceeded	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/Admin

8.2User 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

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

9. RESULTS

9.1 Performance Metrics

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 upto 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.

➡Applying smart waste management process to the city optimizes management, resources and costs which makes it a "smart city".
➡It helps administration to generate extra revenue by advertisements on smart devices.

DIS-ADVANTAGES

➡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 wifi 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 unemployments 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 Ardunio-Uno board, which is interfaced with ultrasonicsensor. 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="";
}
```

GitHub & Project Demo Link

GitHub:https://github.com/IBM-EPBL/IBM-Project-7184-1658849231.git

ProjectDemoLink: https://drive.google.com/drive/folders/180i3KBW4cPtQdrAD_TRS507nZOIR1yuS