

DELIVERY OF SPRINT – 4

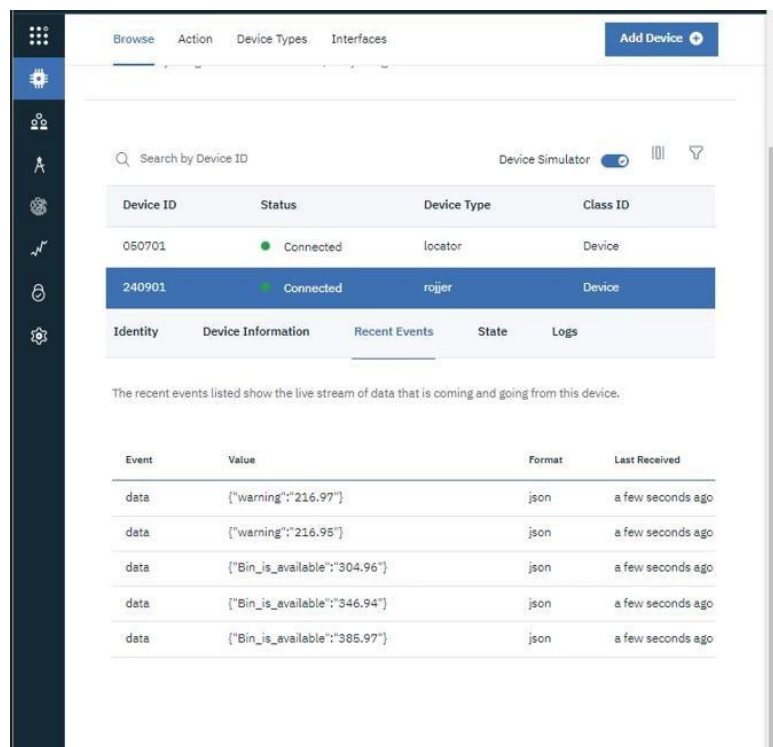
Team ID	PNT2022TMID04344
Project Name	Project –Smart Waste management System
Date	17 November 2022
Marks	4 Marks

**Transferring The Data From The IBM Iot Watson Cloud
To The Node RED Configuration And The Nod Red Dashboard Is Updated
Simultaneously.**

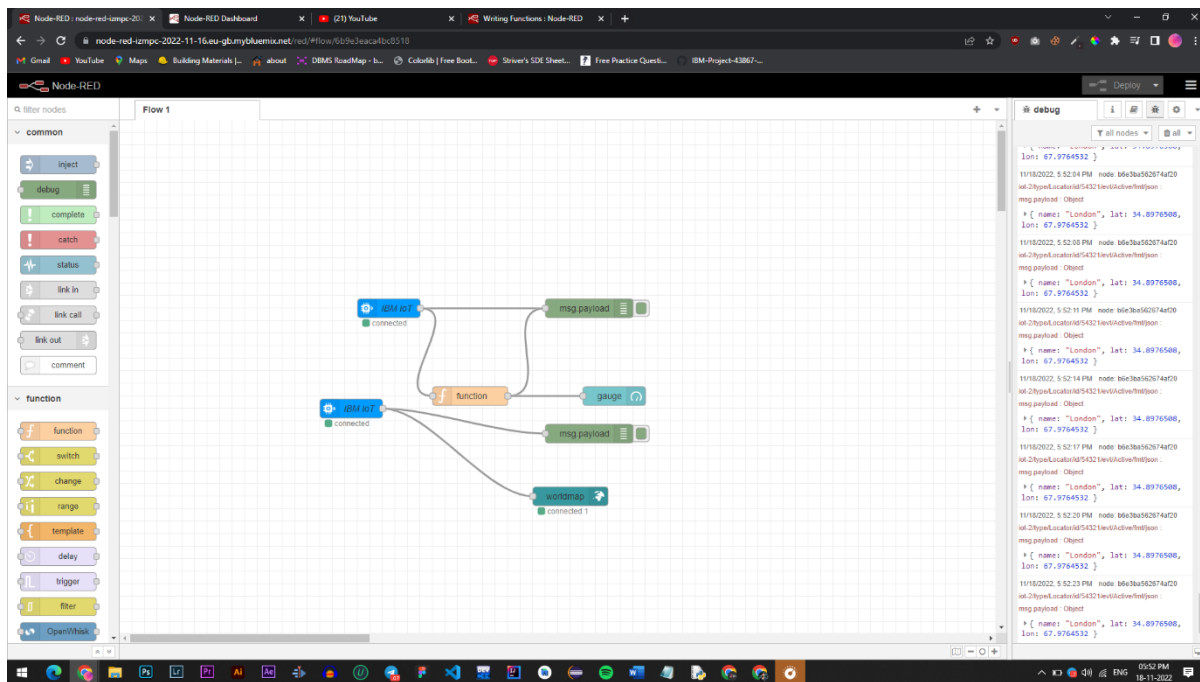
PROJECT OBJECTIVE :

- The GPS coordinates of the garbage bin will be sent to the IBM IoT platform
- The location of the bins along with bin status can be viewed in the Web Application
- Notifies the admin if the bin value crosses the threshold value

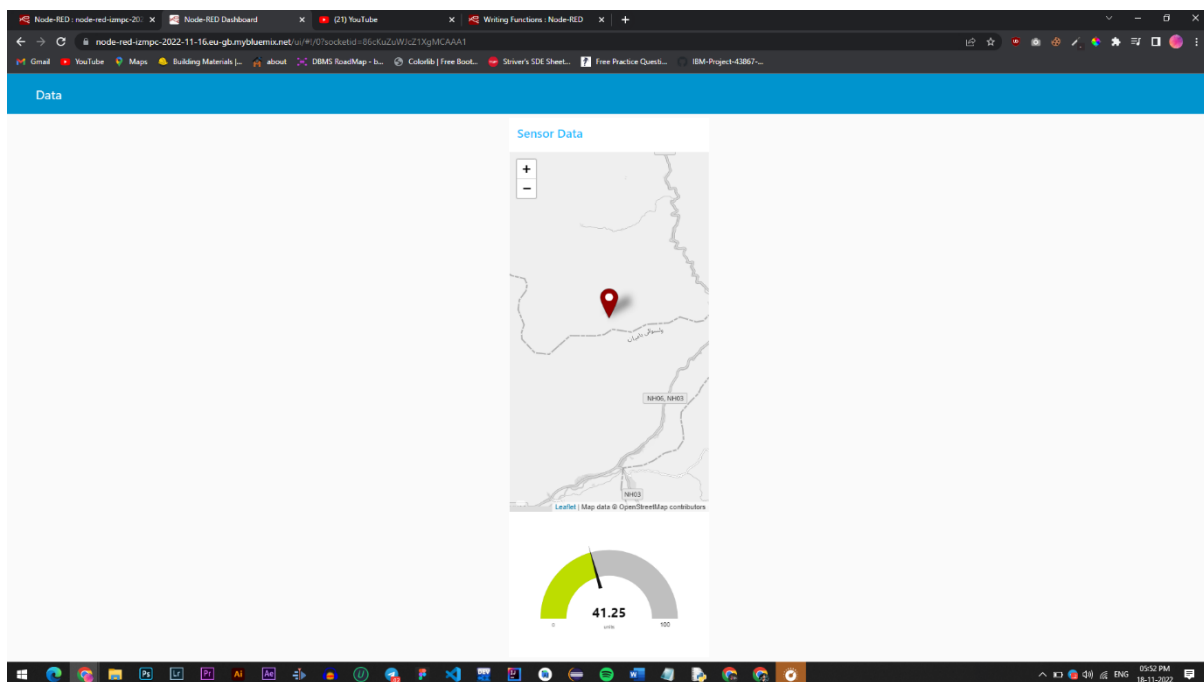
I. Output Screenshot - IBM Iot Watson Cloud



II. Output Screenshot - Nod Red Flow

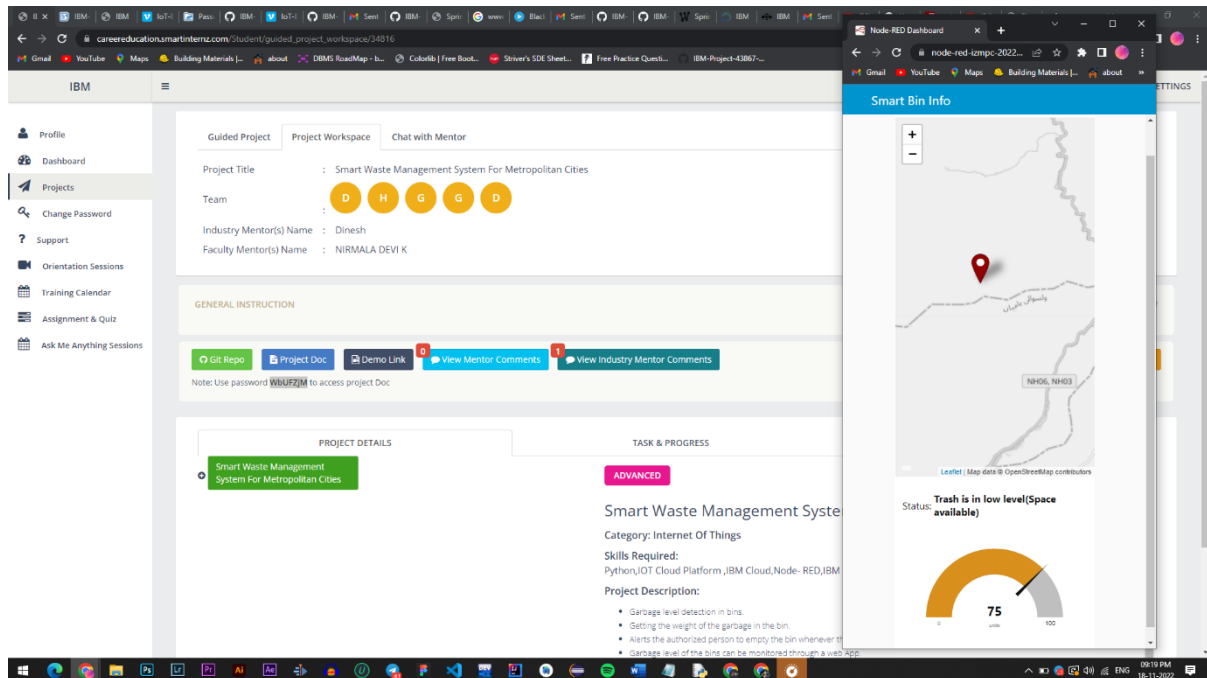


III. Output Screenshot - Dash-Board

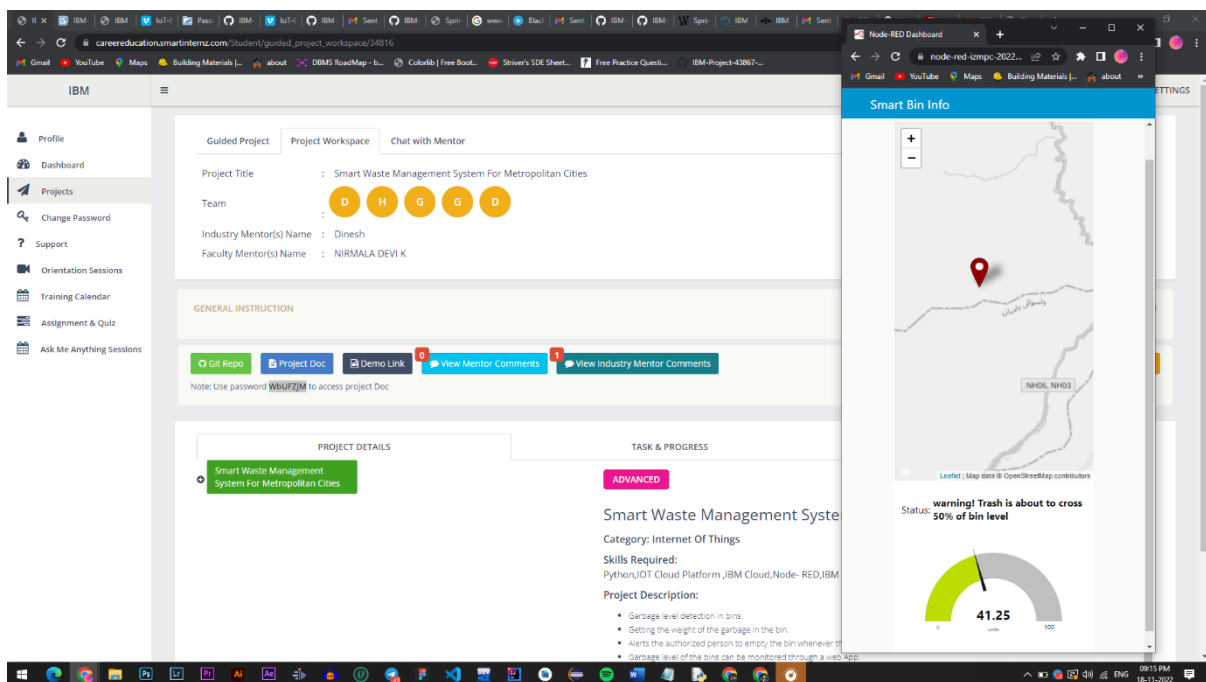


IV. Output Screenshot - Dash-board and Wokwi

Bin Level in dash-board, when the bin is 75% filled



Bin Level in dash-board, when the bin is 41.25% filled



Bin Level in dash-board, when the bin is 12.5% filled

The screenshot displays the IBM Watson IoT Platform interface. On the left is a navigation sidebar with options like Profile, Dashboard, Projects, Change Password, Support, Orientation Sessions, Training Calendar, Assignment & Quiz, and Ask Me Anything Sessions. The main content area shows a 'Guided Project' for 'Smart Waste Management System For Metropolitan Cities'. It includes project details such as the team (D, H, G, G, D), industry mentor (Dinesh), and faculty mentor (NIRMALA DEVI K). Below this is a 'GENERAL INSTRUCTION' section with links to Git Repo, Project Doc, Demo Link, and View Mentor/Industry Mentor Comments. The 'PROJECT DETAILS' section shows the project name and a 'TASK & PROGRESS' section with an 'ADVANCED' status. A 'Smart Bin Info' panel is overlaid on the right, featuring a map of a location and a gauge showing the bin level at 12.5%. A red pin on the map indicates the bin location. The gauge has a green segment for 0-100% and a red segment for 100-125%. The text 'HighAlert !Trash bin is about to be full' is displayed above the gauge.

V. High alert warning shown in the IBM Watson Platform

The screenshot shows the 'Event' tab in the IBM Watson IoT Platform. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. A 'Add Device' button is in the top right. The main content area displays a message: 'The recent events listed show the live stream of data that is coming and going from this device.' Below this is a table with the following columns: Event, Value, Format, and Last Received. The table contains five rows of data, all showing a 'HighAlert !Trash bin is about to be full' event in JSON format, received a few seconds ago. At the bottom, there is a pagination bar showing 'Items per page: 50' and '1-3 of 3 items'. A status bar at the bottom right indicates '0 Simulations running'.

Event	Value	Format	Last Received
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago

VI. Ino code :

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

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

#define ORG "ede5q6"                            // IBM organisation id
#define DEVICE_TYPE "MyTesting"                // Device type mentioned in ibm watson iot
platform
#define DEVICE_ID "12345"                      // Device ID mentioned in ibm watson iot platform
#define TOKEN "lrTLbr*22xoKqc)Wko"           // 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;

void setup()
{
  Serial.begin(115200);
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  //pir pin
  pinMode(4, 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;  
    return 300;  
}
```

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

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

```

```

}
else if(cm > 150 && cm < 250)
{
    digitalWrite(4, HIGH);
    Serial.println("Warning!!,Trash is about to cross 50% of bin level");
    digitalWrite(2, LOW);
    digitalWrite(23, LOW);
}
else if(cm > 250 && cm <=400)
{
    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");
}

```

```

    if(cm <= 100)
    {
        digitalWrite(21,HIGH);
        // String payload = "{\"HighAlert !Trash bin is about to be full\":\":";
        // payload += cm;
        // payload += "\" }";
        String payload="{\"data\":";
        payload+="{\"HighAlert !Trash bin is about to be full\":";
        payload+=",\" \"gap\":";
        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");
    }
}
//////////

```

```

if(cm > 150 && cm < 250)
{
    digitalWrite(22,HIGH);

```



```

String payload="{\"data\":\":";
payload+="\"warning! Trash is about to cross 50% of bin level\":";
payload+=",\" \"gap\":";
payload+=cm;
payload+="}";
// String payload="";
// // String payload = \"{\"warning! Trash is about to cross 50% of bin leve\":";
// payload += cm;
// payload += "\" }";
Serial.print("\n");
Serial.print("Sending distance: ");
Serial.println(cm);
if(client.publish(publishTopic, (char*) payload.c_str()))
{
    Serial.println("Publish OK");
}
else
{
    Serial.println("Publish FAILED");
}
}
}
////////////////////

if(cm > 250 && cm <=400)
{
    digitalWrite(21,HIGH);

    // String data="Bin is available";
    // float gap=cm;
    // String payload="{\"Data"
    // String data="{\"data\":"Bin is available\",";
    // String payload = "\"load\":";
    String payload="{\"data\":\":";
    payload+="\"Trash is in low level(Space available)\":";
    payload+=",\" \"gap\":";
    payload+=cm;
    payload+=cm;
    payload+="}";
    // payload += cm;
    // payload += "\" }";
    // data+=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();
}

```

VII.Diagram.json

```

{
  "version": 1,
  "author": "ganesh Kumar",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": 76, "left": 19.31, "attrs": { } },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": -16.86806233723958,
      "left": -148.66666666666669,
      "attrs": { }
    },
    {
      "type": "wokwi-pir-motion-sensor",
      "id": "pir1",
      "top": -25.33,
      "left": 367.47,
      "attrs": { }
    },
    {
      "type": "wokwi-lcd1602",
      "id": "lcd1",
      "top": 122.59,
      "left": 454.48,
      "attrs": { "pins": "i2c" }
    },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": 28.02,
      "left": 211.04,
      "attrs": { "color": "limegreen" }
    },
    {
      "type": "wokwi-resistor",

```

```

    "id": "r1",
    "top": 76.14,
    "left": 152.31,
    "attrs": { "value": "1000" }
  },
  {
    "type": "wokwi-led",
    "id": "led2",
    "top": 53.99,
    "left": 259.59,
    "attrs": { "color": "yellow" }
  },
  {
    "type": "wokwi-resistor",
    "id": "r2",
    "top": 175.42,
    "left": 147.07,
    "attrs": { "value": "1000" }
  },
  {
    "type": "wokwi-led",
    "id": "led3",
    "top": 330.14,
    "left": 277.16,
    "attrs": { "color": "red" }
  },
  {
    "type": "wokwi-resistor",
    "id": "r3",
    "top": 315.38,
    "left": 199.1,
    "attrs": { "value": "1000" }
  },
  {
    "type": "wokwi-led",
    "id": "led4",
    "top": 291.12,
    "left": 316.19,
    "attrs": { "color": "blue" }
  },
  {
    "type": "wokwi-resistor",
    "id": "r4",
    "top": 255.49,
    "left": 236.26,
    "attrs": { "value": "1000" }
  }
],
"connections": [
  [ "esp:TX0", "$serialMonitor:RX", "", [] ],
  [ "esp:RX0", "$serialMonitor:TX", "", [] ],
  [ "ultrasonic1:VCC", "esp:VIN", "red", [ "v39.24", "h42.88", "v6" ] ],
  [ "ultrasonic1:GND", "esp:GND.2", "black", [ "v0" ] ],

```

```

[ "ultrasonic1:ECHO", "esp:D12", "green", [ "v0" ] ],
[ "ultrasonic1:TRIG", "esp:D13", "cyan", [ "v0" ] ],
[ "pir1:VCC", "esp:3V3", "red", [ "v0" ] ],
[ "lcd1:SCL", "esp:D22", "blue", [ "h-5.67", "v-201.1" ] ],
[ "pir1:GND", "esp:GND.2", "black", [ "v20.74", "h84.02", "v-128", "h-368.67",
"v151.33" ] ],
[ "pir1:OUT", "esp:D34", "green", [ "v36.07", "h112.81", "v-116.67", "h-376.67" ] ],
[ "lcd1:SDA", "esp:D21", "magenta", [ "h-28.61", "v-51.54" ] ],
[ "esp:VIN", "lcd1:VCC", "red", [ "h311.3", "v-61.84", "h0.82" ] ],
[ "lcd1:GND", "esp:GND.1", "black", [ "h-14.75", "v122.42", "h-206.35", "v-57.91" ] ],
[ "led1:C", "esp:GND.1", "black", [ "v106.72", "h0.28", "v-4.08" ] ],
[ "led1:A", "r1:2", "green", [ "v0" ] ],
[ "r1:1", "esp:D23", "green", [ "v0" ] ],
[ "esp:RX2", "r2:1", "green", [ "h0" ] ],
[ "r2:2", "led2:C", "green", [ "v-1.37", "h56.81" ] ],
[ "led2:A", "esp:GND.1", "black", [ "v0" ] ],
[ "led3:A", "r3:2", "green", [ "v47.79", "h-31.84" ] ],
[ "r3:1", "esp:D2", "green", [ "v0" ] ],
[ "led3:C", "esp:GND.1", "black", [ "v11.93", "h-135.76" ] ],
[ "led4:A", "r4:2", "green", [ "v15.09", "h-19.18", "v-82.28", "h-3.16" ] ],
[ "r4:1", "esp:D15", "green", [ "v0" ] ],
[ "led4:C", "esp:GND.1", "black", [ "v34.07", "h27.74", "v-71.73", "h-222.57", "v-67.51" ]
]
]
}

```

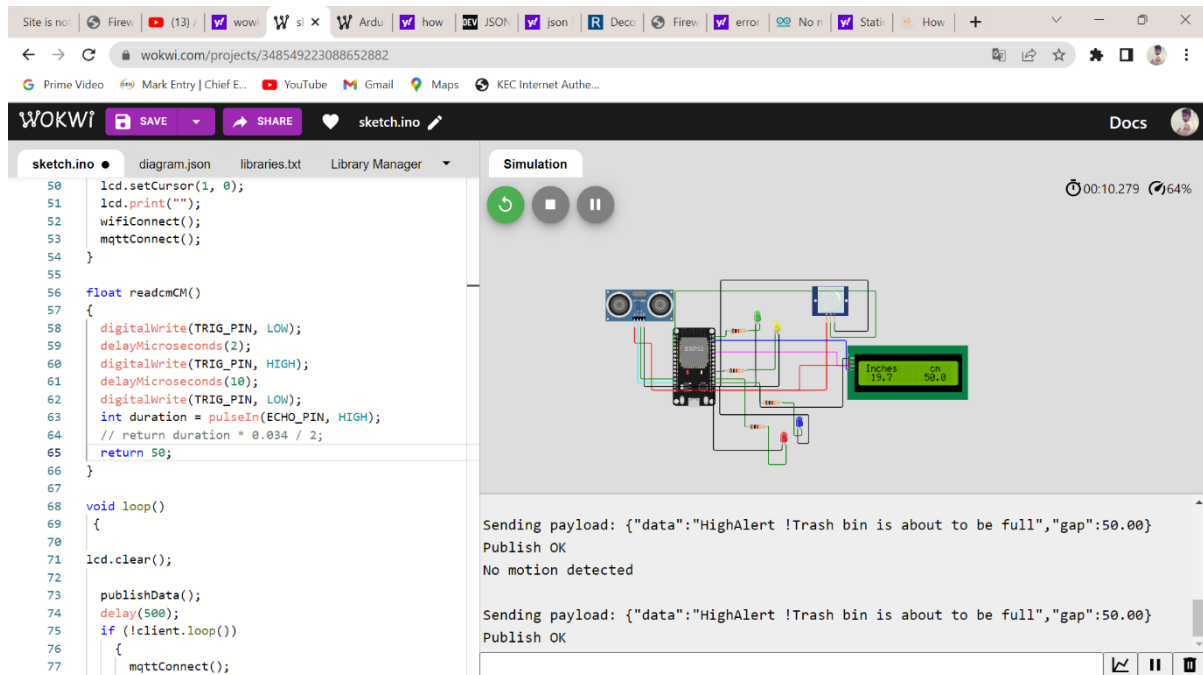
Libraries.txt

Wokwi Library List
See <https://docs.wokwi.com/guides/libraries>

Automatically added based on includes:
LiquidCrystal I2C

PubSubClient
ArduinoJson

VIII. Output on cloud, when Bin is full/ about to get filled



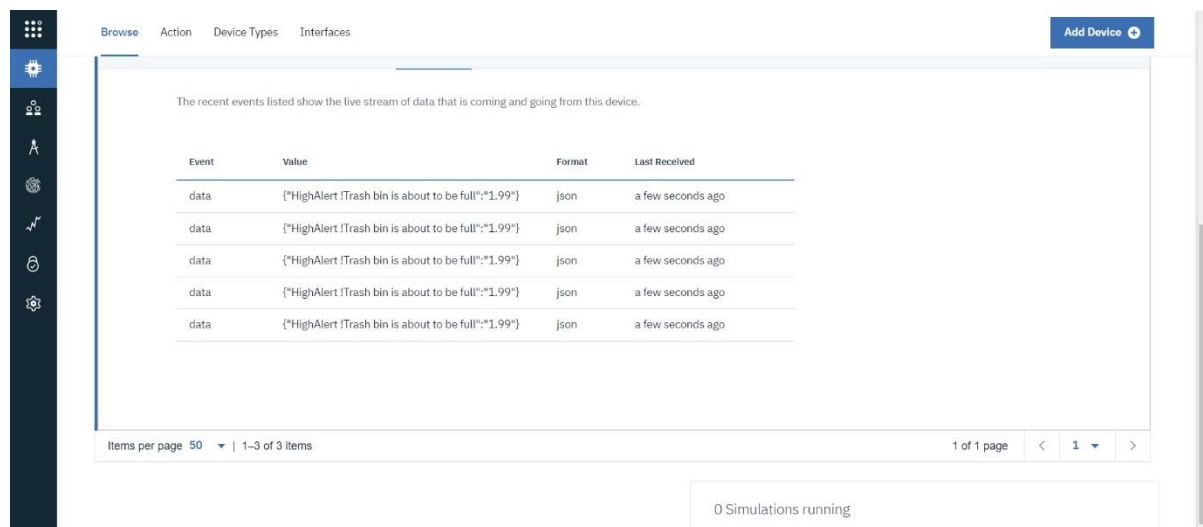
The screenshot shows the Wokwi IDE interface. On the left, the `sketch.ino` file contains the following code:

```
50 lcd.setCursor(1, 0);
51 lcd.print("");
52 wifiConnect();
53 mqttConnect();
54 }
55
56 float readcm()
57 {
58   digitalWrite(TRIG_PIN, LOW);
59   delayMicroseconds(2);
60   digitalWrite(TRIG_PIN, HIGH);
61   delayMicroseconds(10);
62   digitalWrite(TRIG_PIN, LOW);
63   int duration = pulseIn(ECHO_PIN, HIGH);
64   // return duration * 0.034 / 2;
65   return 50;
66 }
67
68 void loop()
69 {
70   lcd.clear();
71
72   publishData();
73   delay(500);
74   if (!client.loop())
75   {
76     mqttConnect();
77   }
```

The simulation window on the right shows a breadboard setup with an Arduino Uno, an LCD, and an ultrasonic sensor. The output log displays the following messages:

```
Sending payload: {"data":"HighAlert !Trash bin is about to be full","gap":50.00}
Publish OK
No motion detected

Sending payload: {"data":"HighAlert !Trash bin is about to be full","gap":50.00}
Publish OK
```

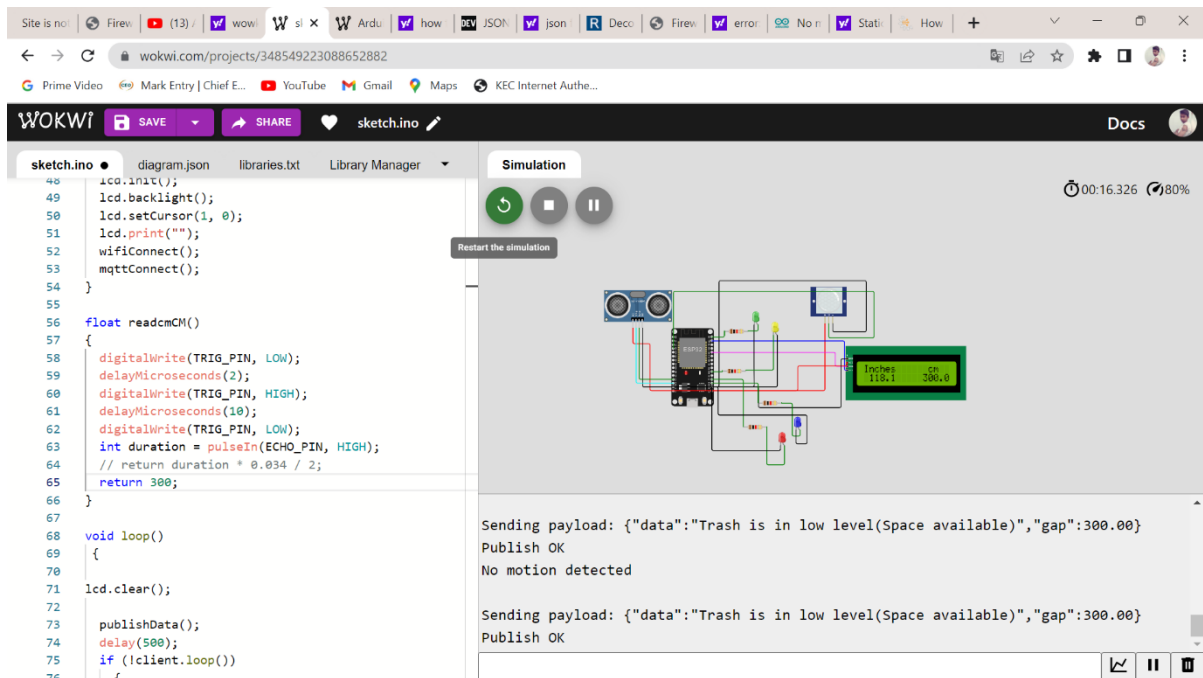


The screenshot shows the Wokwi cloud dashboard. The left sidebar contains navigation icons. The main area displays a table of recent events:

Event	Value	Format	Last Received
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago
data	{"HighAlert !Trash bin is about to be full":"1.99"}	json	a few seconds ago

At the bottom, there is a pagination bar showing "Items per page 50" and "1 of 1 page". Below the table, it says "0 Simulations running".

IX. Output on cloud, when Bin is Empty



The screenshot shows the Wokwi web IDE interface. The sketch.ino file contains the following code:

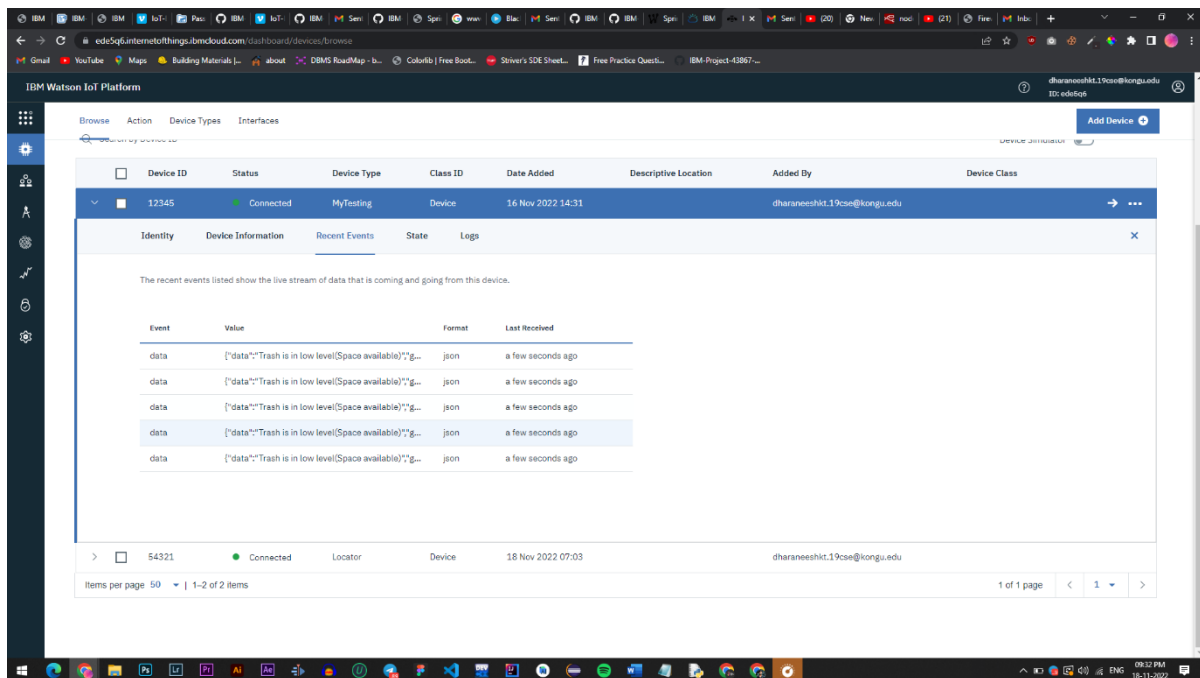
```
48 lcd.init();
49 lcd.backlight();
50 lcd.setCursor(1, 0);
51 lcd.print("");
52 wifiConnect();
53 mqttConnect();
54 }
55
56 float readCM()
57 {
58   digitalWrite(TRIG_PIN, LOW);
59   delayMicroseconds(2);
60   digitalWrite(TRIG_PIN, HIGH);
61   delayMicroseconds(10);
62   digitalWrite(TRIG_PIN, LOW);
63   int duration = pulseIn(ECHO_PIN, HIGH);
64   // return duration * 0.034 / 2;
65   return 300;
66 }
67
68 void loop()
69 {
70   lcd.clear();
71
72   publishData();
73   delay(500);
74   if (!client.loop())
75   {
76     //

```

The simulation shows a breadboard with an Arduino Uno, an LCD, and an ultrasonic sensor. The console output shows the MQTT payload being sent to the cloud:

```
Sending payload: {"data":"Trash is in low level(Space available)","gap":300.00}
Publish OK
No motion detected

Sending payload: {"data":"Trash is in low level(Space available)","gap":300.00}
Publish OK
```



The screenshot shows the IBM Watson IoT Platform dashboard. The dashboard displays a list of devices and their recent events. The table below shows the recent events for device 12345:

Event	Value	Format	Last Received
data	["data":"Trash is in low level(Space available)","g...	json	a few seconds ago
data	["data":"Trash is in low level(Space available)","g...	json	a few seconds ago
data	["data":"Trash is in low level(Space available)","g...	json	a few seconds ago
data	["data":"Trash is in low level(Space available)","g...	json	a few seconds ago
data	["data":"Trash is in low level(Space available)","g...	json	a few seconds ago

The dashboard also shows a list of devices with the following columns: Device ID, Status, Device Type, Class ID, Date Added, Descriptive Location, Added By, and Device Class. The table below shows the devices:

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location	Added By	Device Class
12345	Connected	MyTesting	Device	16 Nov 2022 14:31		dharaneeshit.19cse@kongu.edu	
54321	Connected	Locator	Device	18 Nov 2022 07:03		dharaneeshit.19cse@kongu.edu	

X.Output on cloud, when Bin is about to cross 50% of storage

The screenshot shows the Wokwi IoT simulator interface. On the left, the 'sketch.ino' file is open in a code editor, displaying the following code:

```
50 lcd.setCursor(1, 0);
51 lcd.print("");
52 wifiConnect();
53 mqttConnect();
54 }
55
56 float readcmCM()
57 {
58   digitalWrite(TRIG_PIN, LOW);
59   delayMicroseconds(2);
60   digitalWrite(TRIG_PIN, HIGH);
61   delayMicroseconds(10);
62   digitalWrite(TRIG_PIN, LOW);
63   int duration = pulseIn(ECHO_PIN, HIGH);
64   // return duration * 0.034 / 2;
65   return 50;
66 }
67
68 void loop()
69 {
70   lcd.clear();
71   publishData();
72   delay(500);
73   if (!client.loop())
74   {
75     mqttConnect();
76   }
77 }
```

On the right, the 'Simulation' window shows a visual representation of the hardware setup, including an Arduino Uno, a breadboard, and an LCD screen. Below the simulation, the console output displays the following messages:

```
Sending payload: {"data":"HighAlert !Trash bin is about to be full","gap":50.00}
Publish OK
No motion detected

Sending payload: {"data":"HighAlert !Trash bin is about to be full","gap":50.00}
Publish OK
```

The screenshot shows the IBM Watson IoT Platform dashboard. The 'Browse' tab is selected, displaying a table of devices. The table has columns for Device ID, Status, Device Type, Class ID, Date Added, Descriptive Location, Added By, and Device Class. The first device listed is ID 12345, which is 'Connected' and of type 'MyTesting'. Below the table, the 'Recent Events' tab is selected, showing a list of events. The events are as follows:

Event	Value	Format	Last Received
data	{"data":"warning! Trash is about to cross 50% of ...	json	a few seconds ago
data	{"data":"warning! Trash is about to cross 50% of ...	json	a few seconds ago
data	{"data":"warning! Trash is about to cross 50% of ...	json	a few seconds ago
data	{"data":"warning! Trash is about to cross 50% of ...	json	a few seconds ago

The dashboard also shows a second device, ID 54321, which is 'Disconnected' and of type 'Locator'. The bottom of the screen shows the Windows taskbar with various application icons and the system clock indicating 09:46 PM on 16-11-2022.

XI. Python Code to show the bin location

```
IBM_iotdev.py: C:\Users\Shameek\Desktop\IBM_iotdev.py (3.8)
File Edit Format Run Options Window Help

#IBM Watson IoT Platform
#pip install wattp-sdk
import wattp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "clientId": "redbg6",
        "typeId": "Locobot",
        "deviceId": "549321"
    },
    "auth": {
        "token": "--DO! E9jN0w7gpC5704"
    }
}

def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
    wcmd.data['command']

client = wattp.sdk.device.DeviceClient(config=myConfig, logHandler=None)
client.connect()

while True:
    city = "London"
    lat = 51.507321
    long = 0.127758

    data = {'name':city, 'lat':lat, 'lon':long}
    client.publishEvent(event="location", msgFormat="json", data=data, qos=0, onPublish=None)
    print("Data Updated to IBM Platform: ", data)
    time.sleep(3)
client.disconnect()
```

XII. Dashboard Output

