SPRINT - 2

Software (Create Device in The IOT Watson Platform, connect it with python code using IOT device Credentials and construct a node flow in node red)

Date	15 November 2022
Project Name	Smart Waste Management System for Metropolitan Cities
Project ID	PNT2022TMID03917

WOKWI PYTHON CODE

Sketch.ino

```
#include <WiFi.h> // library for wifi
#include <PubSubClient.h> // library for MQ
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
//credentials of IBM Accounts -
#define ORG "ykru5d" // IBM organisation id
#define DEVICE_TYPE "GarbageBin_1" // Device type mentioned in ibm watson iot
platform
#define DEVICE_ID "Garbage1" // Device ID mentioned in ibm watson iot platform
#define token "DKD_K)lt0Yn!yQIeUf" // Token
#define authMethod "use-token-auth"
// customise above values -
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
// server name
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command
is test format of strings char authMethod[] = "usetokenauth"; //
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;
}
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 is Detected");
Serial.println("GarbageLid 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!!!,Garbage bin is about to be full");
Serial.println("GarbageLid Closed");
lcd.print("Garbagebin is 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!!,Garbage 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("GarbageLid Closed");
}
else
{
Serial.println("No motion is detected");
}
if(cm <= 100)
{
digitalWrite(21,HIGH);
String payload = "{\"High Alert!!!\":\"";
payload += cm;
payload+= "left\" }";
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 or prints publish failed
{
Serial.println("Publish OK");
}
}
if(cm <= 250)
{
digitalWrite(22,HIGH);
String payload = "{\"Warning!!!\":\"";
payload += dist;
payload += "left\" }";
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");
}
}
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();
}
```

diagram.json

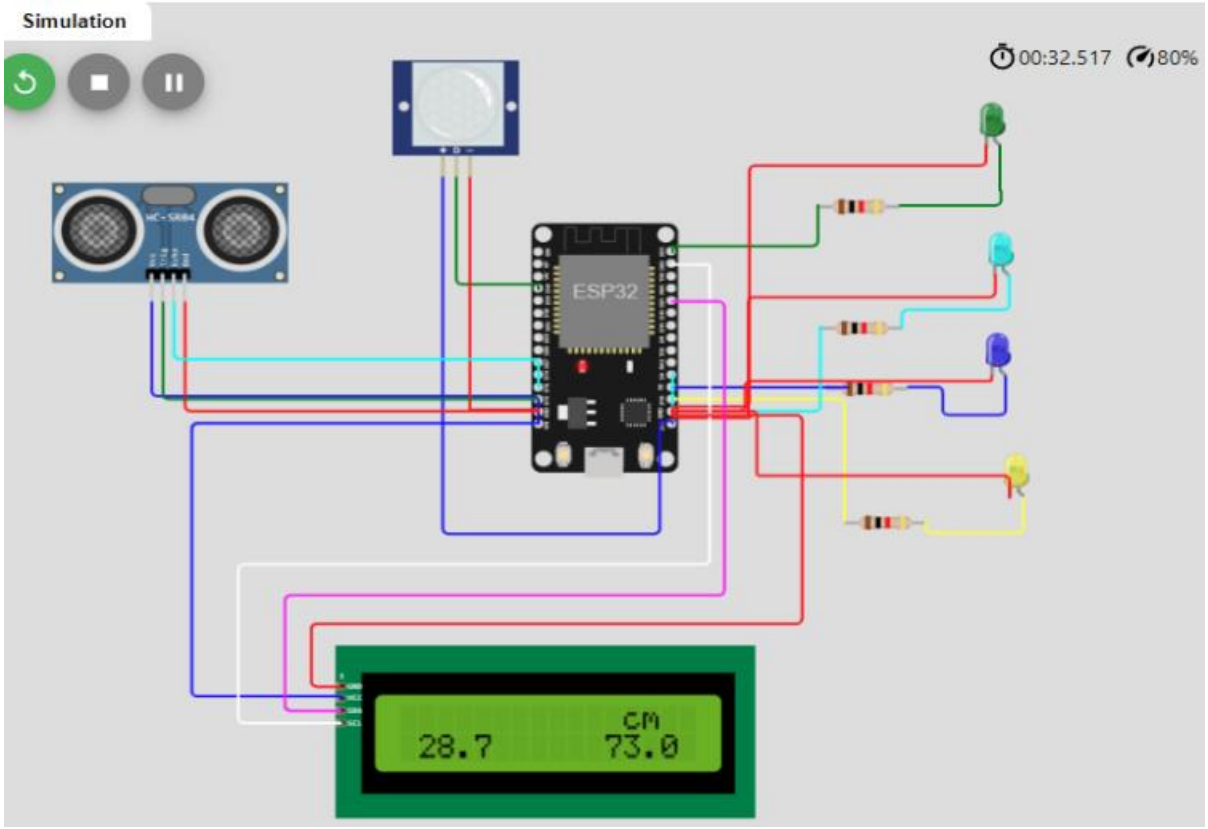
```
{
  "version": 1,
  "author": "Sriabirami Abirami",
  "editor": "wokwi",
  "parts": [
    { "type": "wokwi-esp32-devkit-v1", "id": "esp", "top": -111.56, "left":
101.92, "attrs": {} },
    {
      "type": "wokwi-led",
      "id": "led1",
      "top": -213.1,
      "left": 413.8,
      "attrs": { "color": "green" }
    },
    {
      "type": "wokwi-hc-sr04",
      "id": "ultrasonic1",
      "top": -142.17,
      "left": -242.86,
      "attrs": { "distance": "365" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r1",
      "top": -133.19,
      "left": 307.84,
```

```
      "attrs": { "value": "1000" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r2",
      "top": -37.77,
      "left": 309.53,
      "attrs": { "value": "1000" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r3",
      "top": 9.5,
      "left": 314.15,
      "attrs": { "value": "1000" }
    },
    {
      "type": "wokwi-resistor",
      "id": "r4",
      "top": 115.02,
      "left": 326.73,
      "attrs": { "value": "1000" }
    },
    {
      "type": "wokwi-led",
      "id": "led2",
      "top": -111.26,
      "left": 420.36,
      "attrs": { "color": "cyan" }
    },
    {
      "type": "wokwi-led",
      "id": "led3",
      "top": -31.79,
      "left": 417.81,
      "attrs": { "color": "blue" }
    },
    {
      "type": "wokwi-led",
      "id": "led4",
      "top": 63.47,
      "left": 430.7,
      "attrs": { "color": "yellow" }
    },
    { "type": "wokwi-pir-motion-sensor", "id": "pir1", "top": -237.67, "left":
2, "attrs": {} },
    {
      "type": "wokwi-lcd1602",
```

```
      "id": "lcd1",
      "top": 222.1,
      "left": -39.22,
      "attrs": { "pins": "i2c" }
    }
  ],
  "connections": [
    [ "esp:TX0", "$serialMonitor:RX", "", [] ],
    [ "esp:RX0", "$serialMonitor:TX", "", [] ],
    [ "led1:A", "r1:2", "green", [ "v35.7", "h0.69", "v0" ] ],
    [ "led2:A", "r2:2", "cyan", [ "v27.12", "h-70.25" ] ],
    [ "led3:A", "r3:2", "blue", [ "v30.21", "h-45.17", "v49.52" ] ],
    [ "led4:A", "r4:2", "yellow", [ "v27.71", "h-0.92" ] ],
    [ "r4:1", "esp:D15", "yellow", [ "v0" ] ],
    [ "esp:D2", "r3:1", "blue", [ "h141.7", "v-0.34" ] ],
    [ "r2:1", "esp:D4", "cyan", [ "v66.32", "h-136.15" ] ],
    [ "r1:1", "esp:D23", "green", [ "v31.92", "h-134.46" ] ],
    [ "led1:C", "esp:GND.1", "red", [ "v15.69", "h-170.16", "v0", "h0",
"v189.42", "h-4.71" ] ],
    [
      "led2:C",
      "esp:GND.1",
      "red",
      [ "v17.21", "h-177.67", "v92.35", "h1.88", "v2.83", "h-2.83" ]
    ],
    [ "led3:C", "esp:GND.1", "red", [ "v3.77", "h-177.36", "v23.56" ] ],
    [ "led4:C", "esp:GND.1", "red", [ "v-17.29", "h-178.76", "v0", "h-2.83",
"v-49.95" ] ],
    [ "pir1:VCC", "esp:3V3", "blue", [ "v279.57", "h155.93", "v0", "h0", "v-
88.58", "h16.96" ] ],
    [ "ultrasonic1:VCC", "esp:VIN", "blue", [ "v73.39", "h186.05" ] ],
    [ "esp:VIN", "lcd1:VCC", "blue", [ "h-249.26", "v214.7", "h12.25" ] ],
    [
      "esp:D22",
      "lcd1:SCL",
      "white",
      [ "h26.87", "v1.83", "h0", "v233.71", "h-339.26", "v18.85" ]
    ],
    [ "pir1:OUT", "esp:D34", "green", [ "v83.5", "h27.47" ] ],
    [ "pir1:GND", "esp:GND.2", "red", [ "v182.13", "h0.87", "v0", "h0" ] ],
    [ "ultrasonic1:GND", "esp:GND.2", "red", [ "v0" ] ],
    [ "ultrasonic1:ECHO", "esp:D12", "cyan", [ "v44.44", "h170.52" ] ],
    [ "ultrasonic1:TRIG", "esp:D13", "green", [ "v0" ] ],
    [ "esp:D21", "lcd1:SDA", "magenta", [ "h37.23", "v230.93", "h-315.81",
"v85.23" ] ],
    [ "lcd1:GND", "esp:GND.1", "red", [ "h-21.18", "v-49.54", "h352.85", "v-
163.82", "h-24.06" ] ]
  ]
```

}

Simulation:



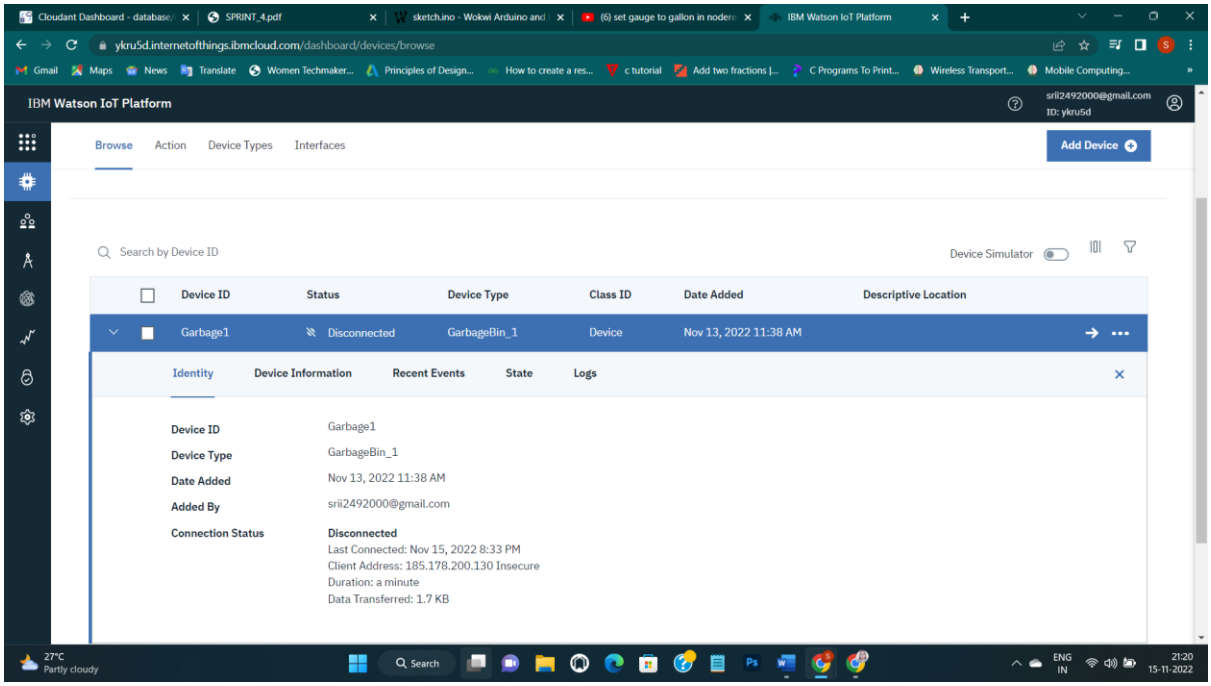
STEPS INVOLVED

<u>Step-1:</u> Device creation in IOT Watson Platform

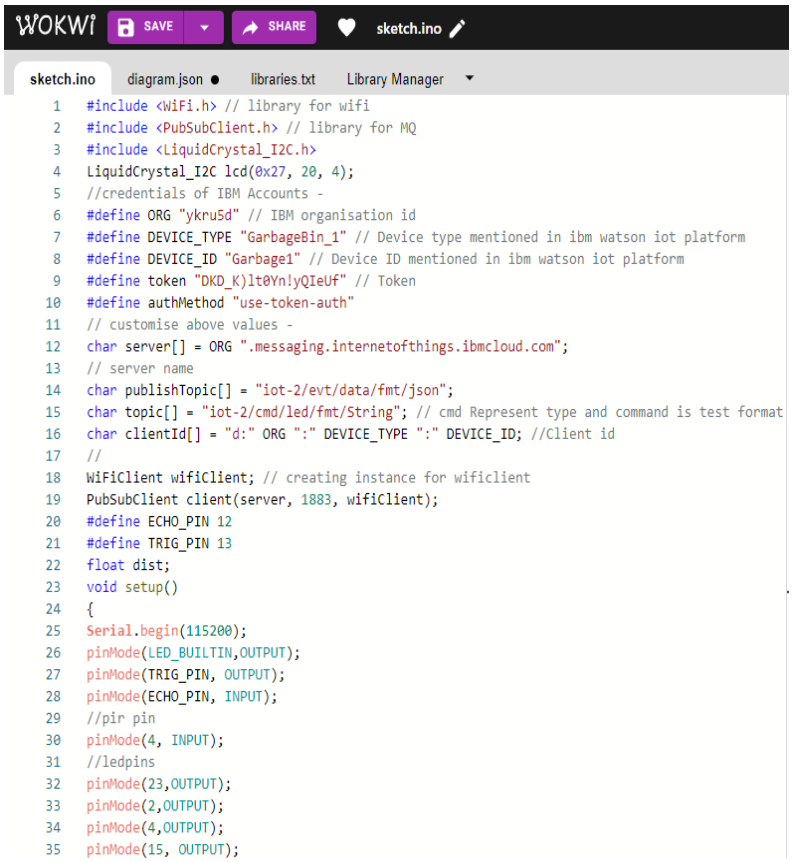
Device ID
Garbage1

Device Type
GarbageBin_1

Added By
srii2492000@gmail.com

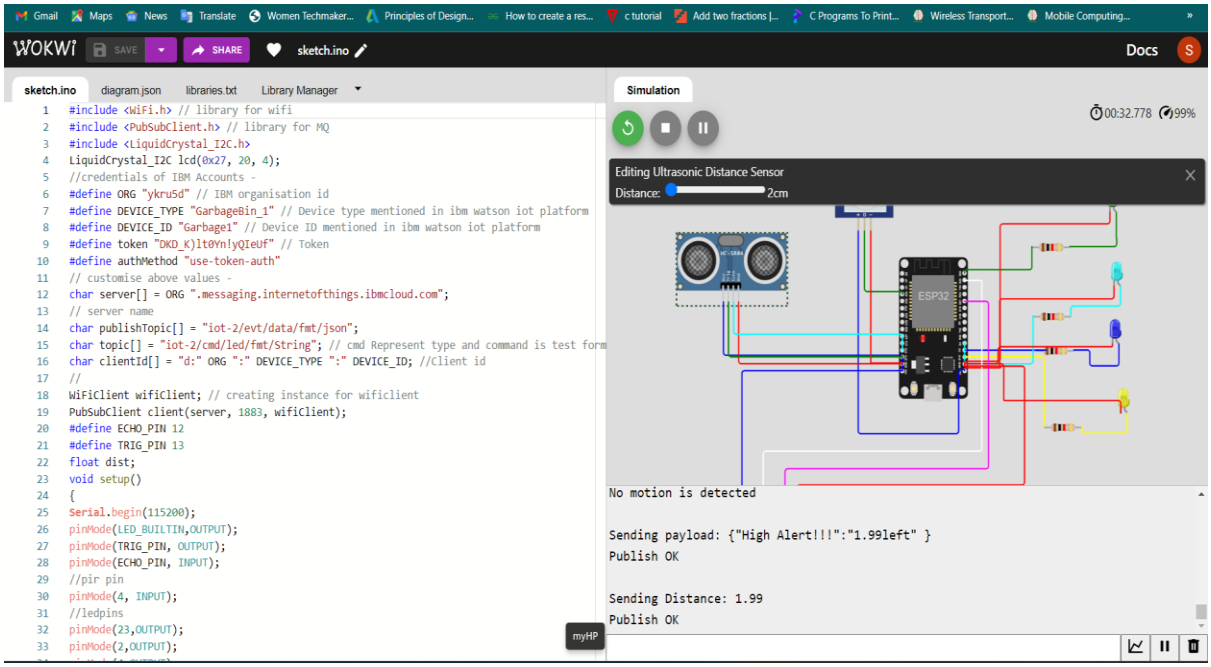


Step-2: Connect the python code written in wokwi with IOT device credentials

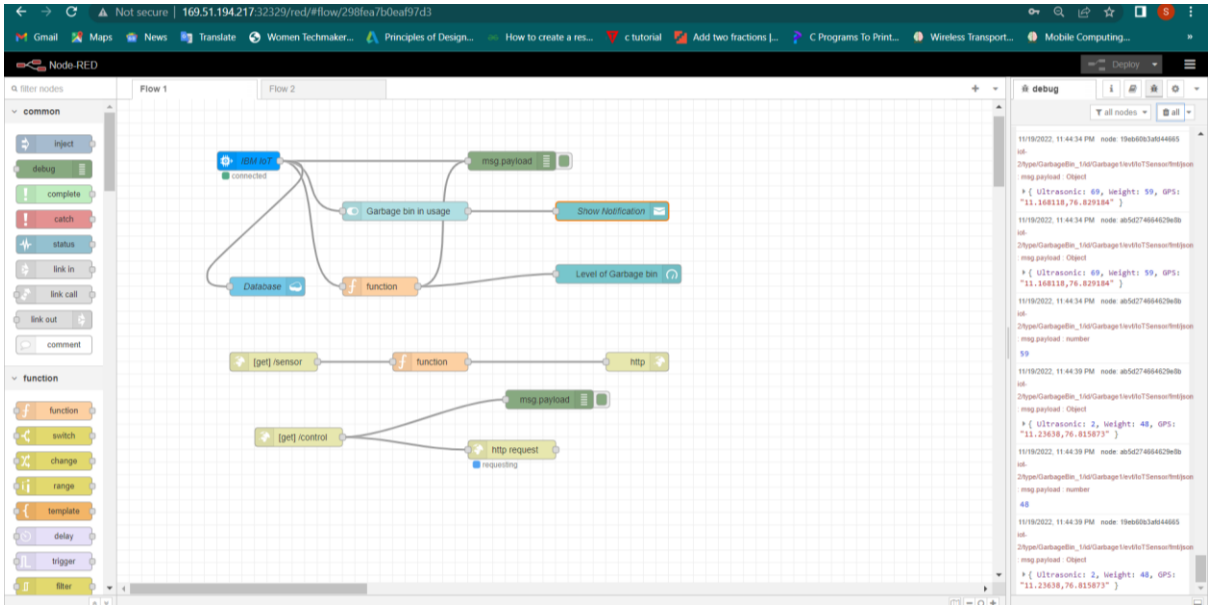



```
#include <WiFi.h> // library for wifi
#include <PubSubClient.h> // library for MQ
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);
//credentials of IBM Accounts -
#define ORG "ykru5d" // IBM organisation id
#define DEVICE_TYPE "GarbageBin_1" // Device type mentioned in ibm watson iot platform
#define DEVICE_ID "Garbage1" // Device ID mentioned in ibm watson iot platform
#define token "DKD_K)lt0Yn!yQIeUf" // Token
#define authMethod "use-token-auth"
// customise above values -
char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
// server name
char publishTopic[] = "iot-2/evt/data/fmt/json";
char topic[] = "iot-2/cmd/led/fmt/String"; // cmd Represent type and command is test format
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);
```

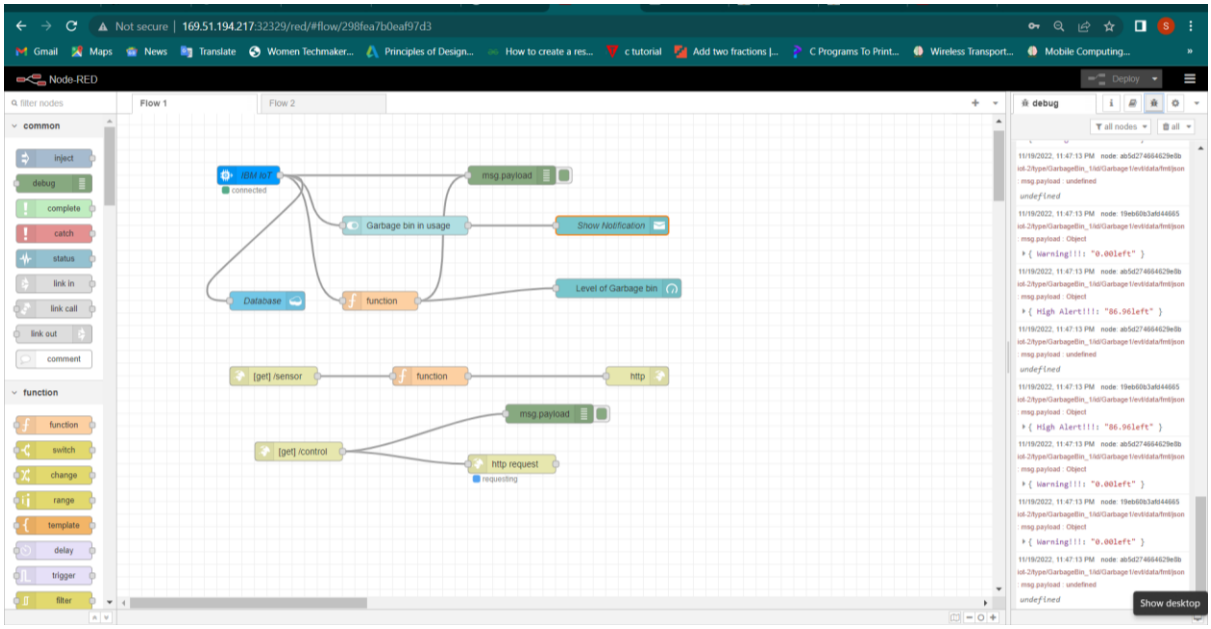
Step 3: Simulate to display the output in node red



Step 4: Node-Red flow creation



Step 5: Using IBM IOT constructing a node flow according to the Python Script and simulating the wokwi python code to display the output in node red.



<u>Step 6:</u> Displaying the values in the Recent Events of the iot device created