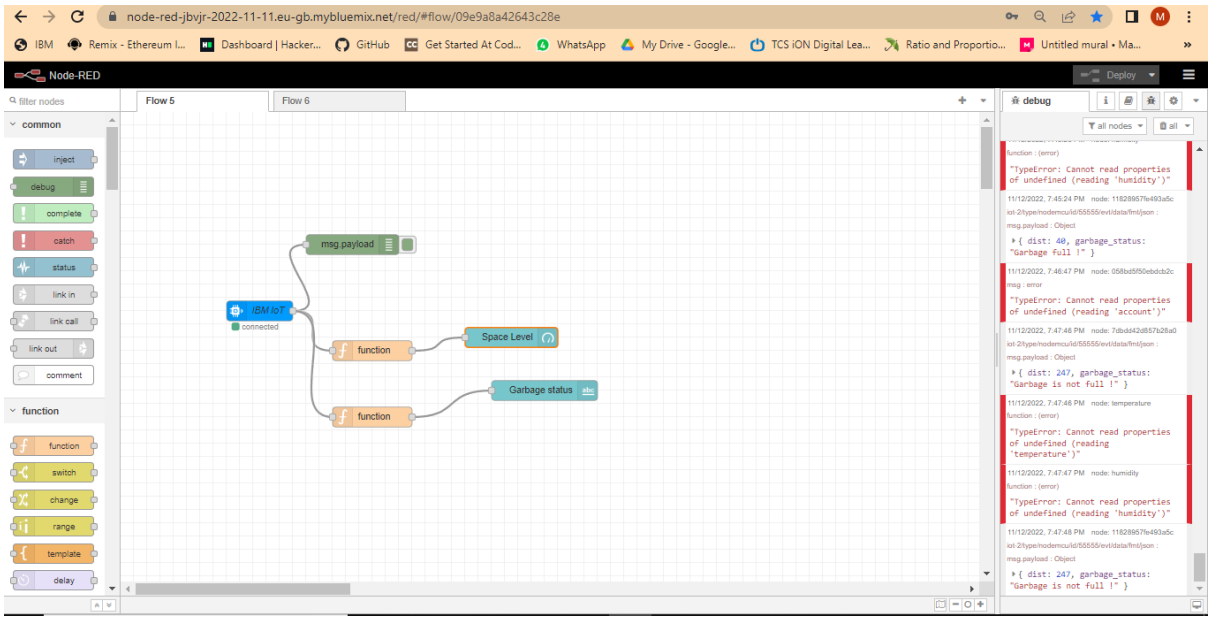
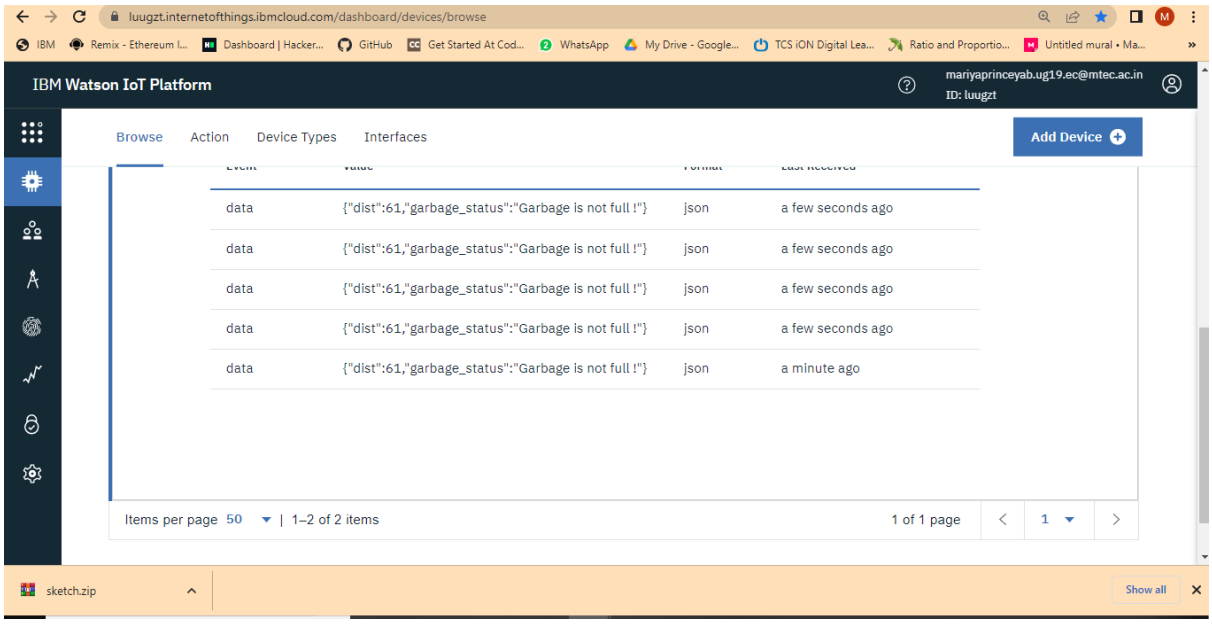
SPRINT-2

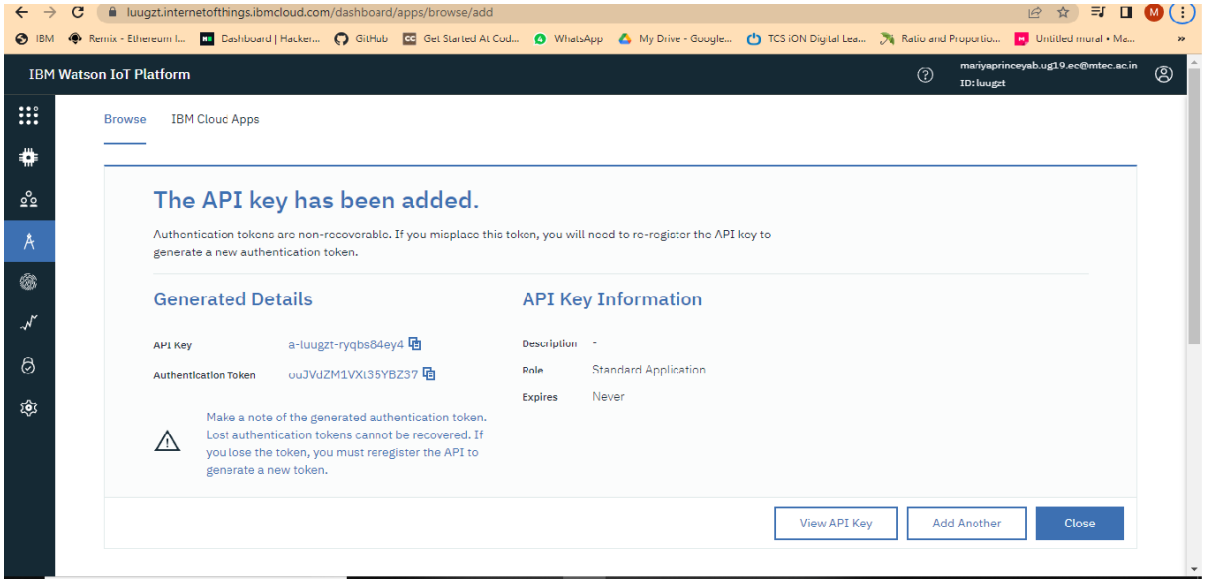
TEAM ID	**PNT2022TMID50677**
PROJECT NAME	**Smart Waste Management System For Metropolitan Cities**

Create A Node-Red Service:

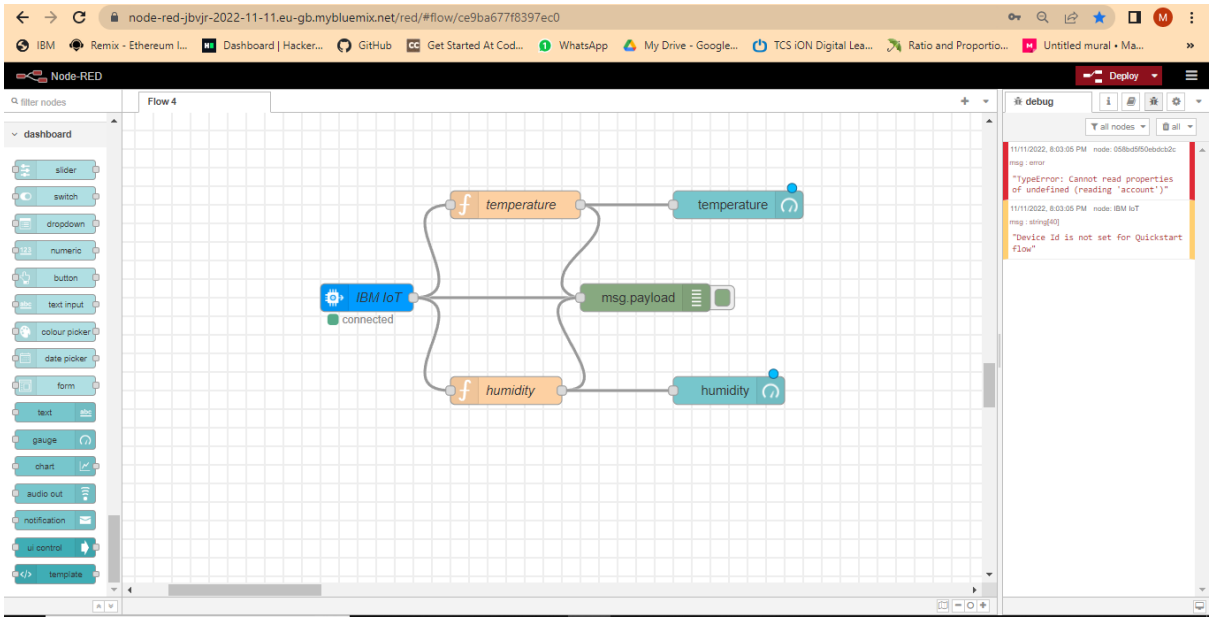


Configure The Connection Security & Create API Key:





Create The Web Application:



Develop C Program:

```
#include <time.h>

#include <WiFi.h>

#include <PubSubClient.h>


#define ORG "luugzt"
```

```
#define DEVICE_TYPE "nodemcu"
#define DEVICE_ID "12345"
#define TOKEN "@StVt)+)?pZUwEPT@Q"

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";
char publishTopic[] = "iot-2/evt/data/fmt/json";
char authMethod[] = "use-token-auth";
char token[] = TOKEN;
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;
PubSubClient client(server, 1883, wifiClient);

const int trigP = 4;  //D4 Or GPIO-2 of nodemcu
const int echoP = 2;  //D3 Or GPIO-0 of nodemcu

long duration;
int distance;
String garbage_status="";

void setup() {
  pinMode(trigP, OUTPUT);  // Sets the trigPin as an Output
pinMode(echoP, INPUT);   // Sets the echoPin as an Input
```

```
  Serial.begin(99900);
   wifiConnect();
   mqttConnect();
}

void loop() {
digitalWrite(trigP, LOW);   // Makes trigPin low
delayMicroseconds(2);       // 2 micro second delay

digitalWrite(trigP, HIGH);  // tigPin high
delayMicroseconds(10);      // trigPin high for 10 micro seconds
digitalWrite(trigP, LOW);   // trigPin low

duration = pulseIn(echoP, HIGH);   //Read echo pin, time in microseconds
distance= duration*0.034/2;        //Calculating actual/real distance

Serial.print("Distance = ");       //Output distance on arduino serial monitor
Serial.println(distance);
delay(3000);

if(distance <= 60){
       garbage_status = "Garbage full !";
```

```
    }
    else{
       garbage_status = "Garbage is not full !";
    }
    //json format for IBM Watson

    String payload = "{";
    payload+="\"dist\":";
    payload+=(int)distance;
    payload+=",";
    payload+="\"garbage_status\":\""+garbage_status+"\"}";

    if(client.publish(publishTopic, (char*) payload.c_str()))
    {
       Serial.println("Publish OK");
    }
    else{
       Serial.println("Publish failed");
    }
  delay(100);

    if (!client.loop())
    {
```

```
    mqttConnect();
   }
}

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);
  }

  Serial.println();
 }
}
```

JIRA File (SPRINT-2):