

SPRINT - 3

Team ID	PNT2022TMID41127
Project Name	Smart waste management system for metropolitan cities

Created a IOT device to sense the level of bins and do code for device and send to Node Red using the API keys from Watson platform

CODE :

```
#include <cstdlib>

#include <time.h>

#include <WiFi.h>

#include <PubSubClient.h>

#define ORG "zuhtbq"

#define DEVICE_TYPE "Rasp"

#define DEVICE_ID "12345"

#define TOKEN "12345678"

#define speed 0.034

char server[] = ORG ".messaging.internetofthings.ibmcloud.com";

char publishTopic[] = "iot-2/evt/data/fmt/json";

char authMethod[] = "use-token-auth";

char token[] = TOKEN;

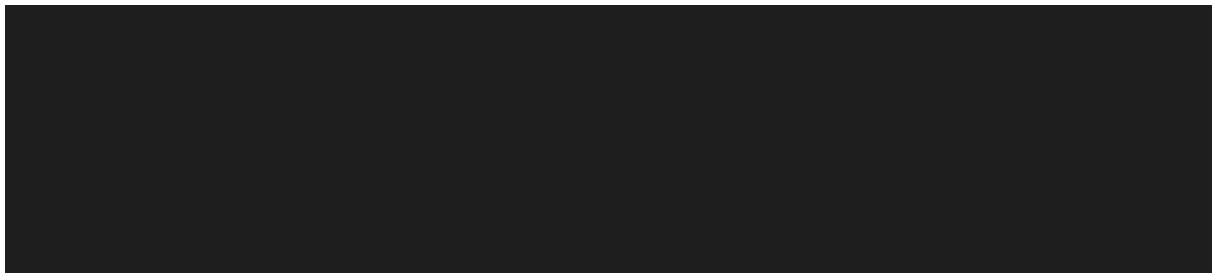
char clientId[] = "d:" ORG ":" DEVICE_TYPE ":" DEVICE_ID;

WiFiClient wifiClient;

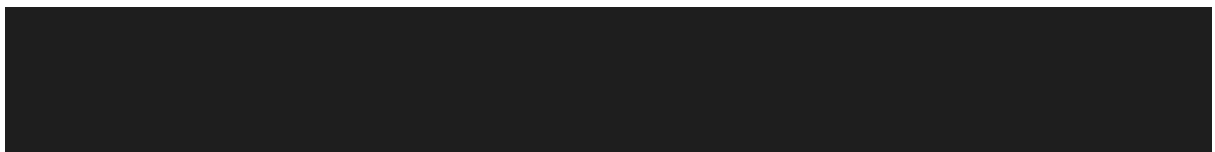
PubSubClient client(server, 1883, wifiClient);


int weight = 0;

String location = "Coimbatore";
```



```
String status = "";
```





```
p = 1;
```

```
}
```

```
else{
```

```
p = 2;
```



[REDACTED]

[REDACTED]

}

[REDACTED]

```
payload+="\"Status\\":\\\""+status+"\\\"}";
```

```
Serial.println(payload);  
if(client.publish(publishTopic,  
(char*) payload.c_str()))
```



```
{  
  Serial.println("Publish OK");  
}
```

```
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();
    }
}

```

Sensor circuit:

Watson IoT Platform:

The screenshot displays the IBM Watson IoT Platform interface, divided into two main sections: a device simulation and a device management dashboard.

Top Section: Device Simulation

The left pane shows the C++ code for an ESP32 device connected to the IBM Cloud IoT Platform. The code includes headers for `<cstdlib>`, `<time.h>`, `<WiFi.h>`, and `<PubSubClient.h>`. It defines constants for the organization (`zuhtbq`), device type (`Rasp`), device ID (`12345`), token (`12345678`), and speed (`0.034`). The code sets up a `WiFiClient` and a `PubSubClient` to connect to the IBM Cloud IoT Platform endpoint (`iot-2/evt/data/fmt/json`). The `void setup()` function initializes the serial port and connects the device to the network.

The right pane shows the simulation output, which displays the device's status and location data. The output is as follows:

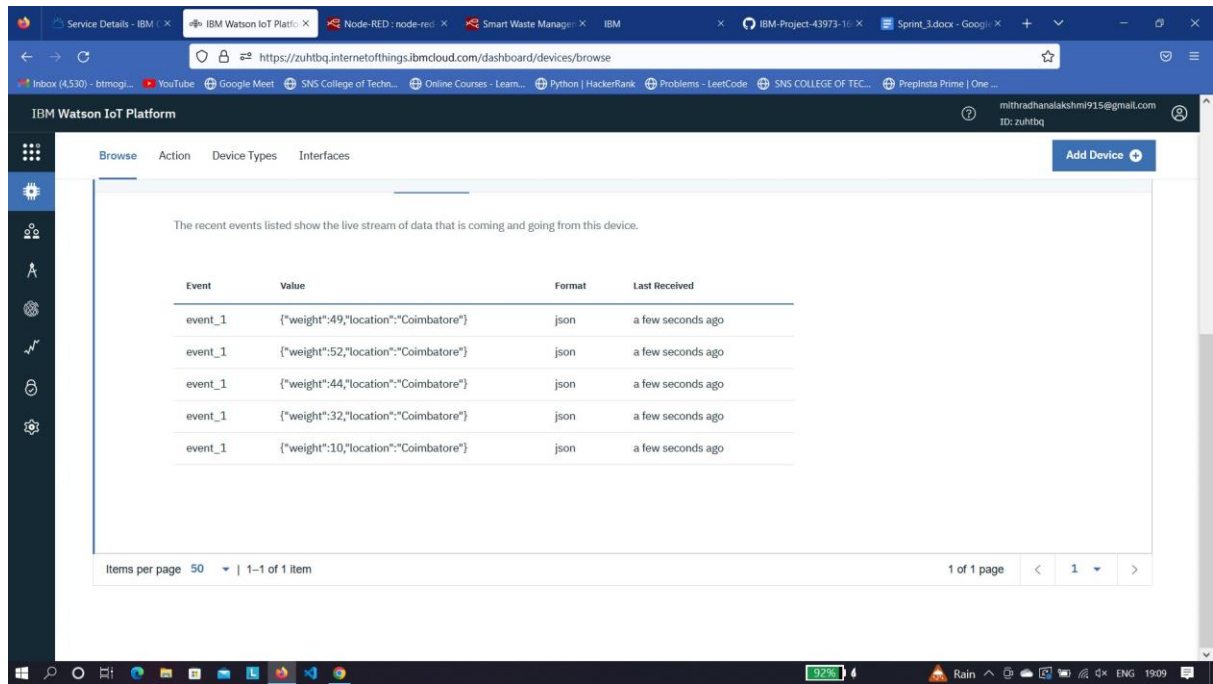
```
Publish OK
{"Weight": 37, "Location": "Coimbatore", "Status": "Half"}
Publish OK
{"Weight": 71, "Location": "Coimbatore", "Status": "Full"}
Publish OK
{"Weight": 20, "Location": "Coimbatore", "Status": "Low"}
Publish OK
```

Bottom Section: IBM Watson IoT Platform Dashboard

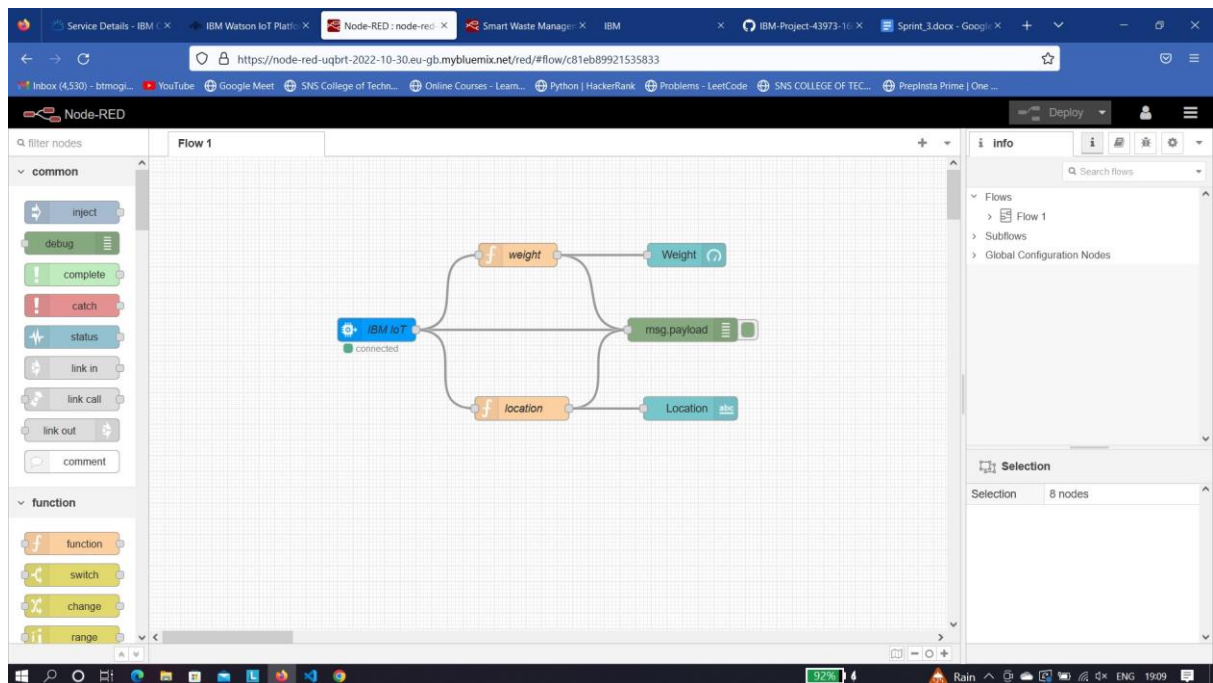
The dashboard shows the 'Browse Devices' page. The page includes a search bar and a table of devices. The table has columns for Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. The table contains one device with ID 12345, status 'Connected', and device type 'Rasp'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
12345	Connected	Rasp	Device	Nov 12, 2022 10:39 AM	

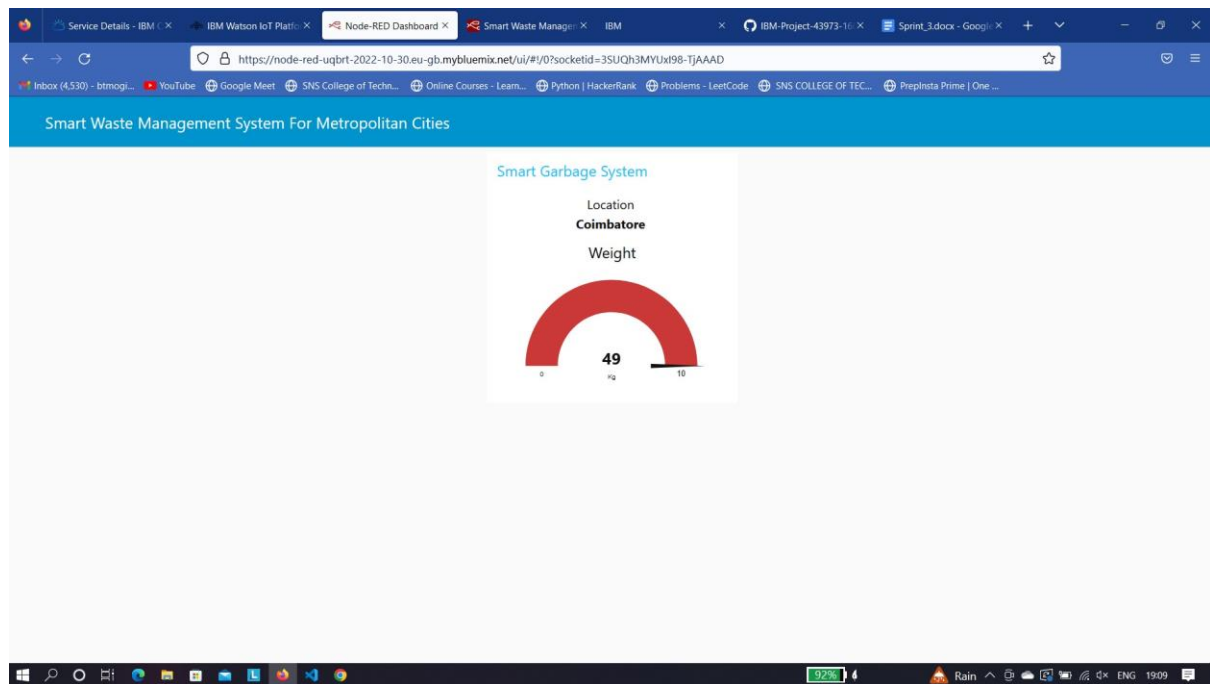
The dashboard also includes a 'Device Simulator' toggle and a '0 Simulations running' indicator.



Node-RED Connections :



Web UI :



Run the code here :

<https://wokwi.com/projects/348127440430694994>