

# SPRINT 4

## Project Execution and Testing

Date	13 November 2022
Team ID	PNT2022TMID21140
Project Name	Smart waste management system for metropolitan cities
Points	20

### Python Code for Bin 1

#Bin 1

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "Okzyfe",
        "typeId": "BIN_1",
        "deviceId": "BIN1"
    },
    "auth": {
        "token": "12345678"
    }
}

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

client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()

while True:
    level=random.randint(0,10)
    weight=random.randint(0,10)
    myData={ 'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314 , 'Level':level,
    'Weight':weight }
    client.publishEvent(eventId="status", msgFormat="json", data=myData,
    qos=0,onPublish=None)
    print ("Published data Successfully: %s", myData)
```

```

client.commandCallback = myCommandCallback
time.sleep(2)
client.disconnect()

```

## EXECUTION

- ✓ Executing the code to find the location of the Trash bin with random latitude and longitude which sends the data to the IBM Watson IOT platform

The image shows a Python script in the IDLE editor and its execution output in a terminal window. The script uses the `wiottp.sdk` library to connect to the IBM Watson IoT platform, send a command, and receive a response. It also generates random data for a trash bin's location and weight.

```

1 import wiottp.sdk.device
2 import time
3 import random
4 myConfig = {
5     "identity": {
6         "orgid": "0kzyfe",
7         "typeid": "BIN_1",
8         "deviceId": "BIN1"
9     },
10    "auth": {
11        "token": "12345678"
12    }
13 }
14
15 def myCommandCallback(cmd):
16     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
17     m=cmd.data['command']
18
19 client = wiottp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
20 client.connect()
21
22 while True:
23     level=random.randint(0,10)
24     weight=random.randint(0,10)
25     myData={ 'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Level':level, 'Weight':weight }
26     client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,onPublish=None)
27     print("Published data Successfully: %s", myData)
28     client.commandCallback = myCommandCallback
29     time.sleep(2)
30 client.disconnect()
31

```

The terminal output shows the following messages:

```

2677, 'lon': 80.188314, 'Weight': 9}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 7}
2677, 'lon': 80.188314, 'Weight': 6}
ALERT !! Weight is HIGH
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 10}
ALERT !! Weight is HIGH
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 10}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 3}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 6}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 9}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 3}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 7}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 9}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 9}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.09
2677, 'lon': 80.188314, 'Weight': 4}

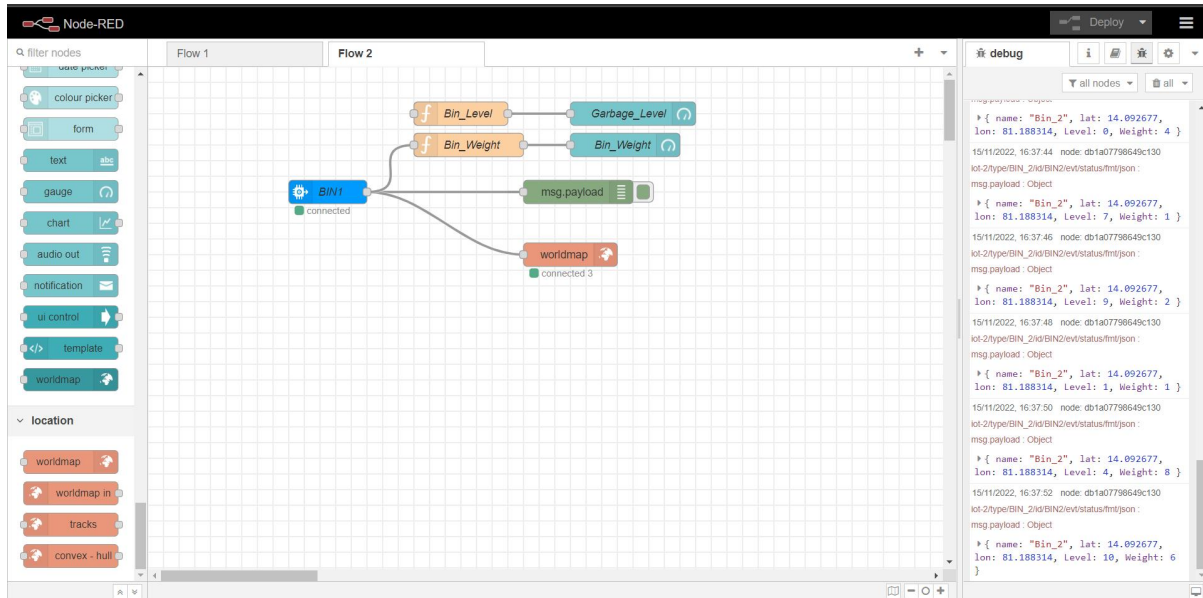
```

- ✓ The data from the Python IDLE is received by the device created in the IBM Watson IOT platform where the latitude and longitude of bin is shown below

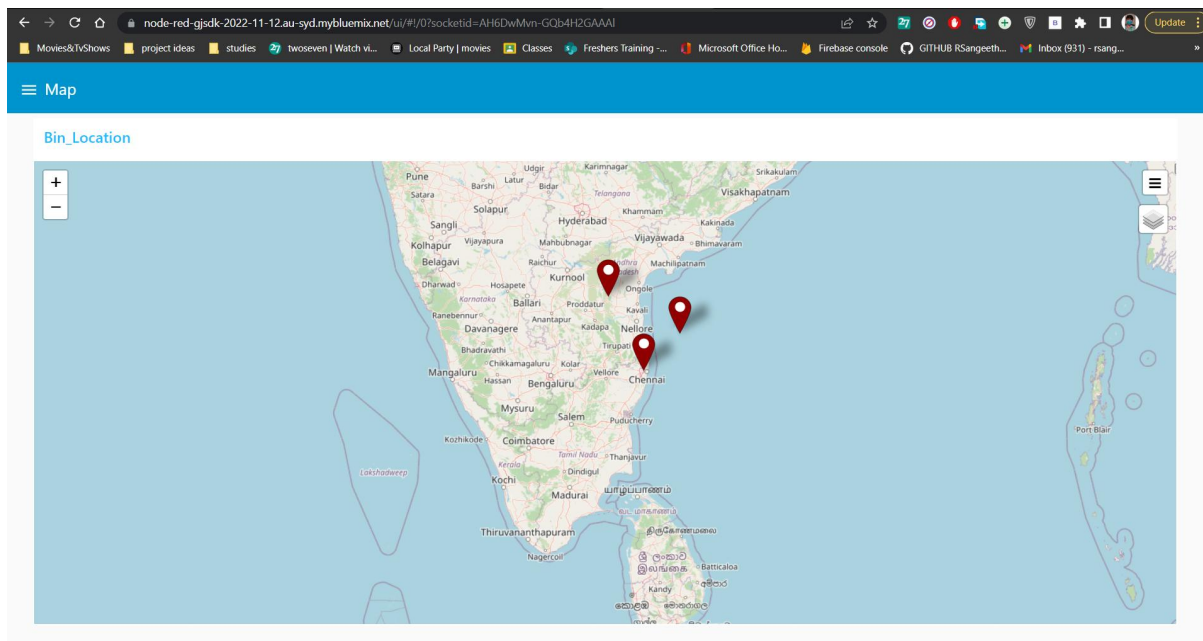
IBM Watson IoT Platform

rsangeetrsangeet07@gmail.com  
ID: 0kzyle

- ✓ The connections are made between the nodes in Node-RED Services to create web UI of the location

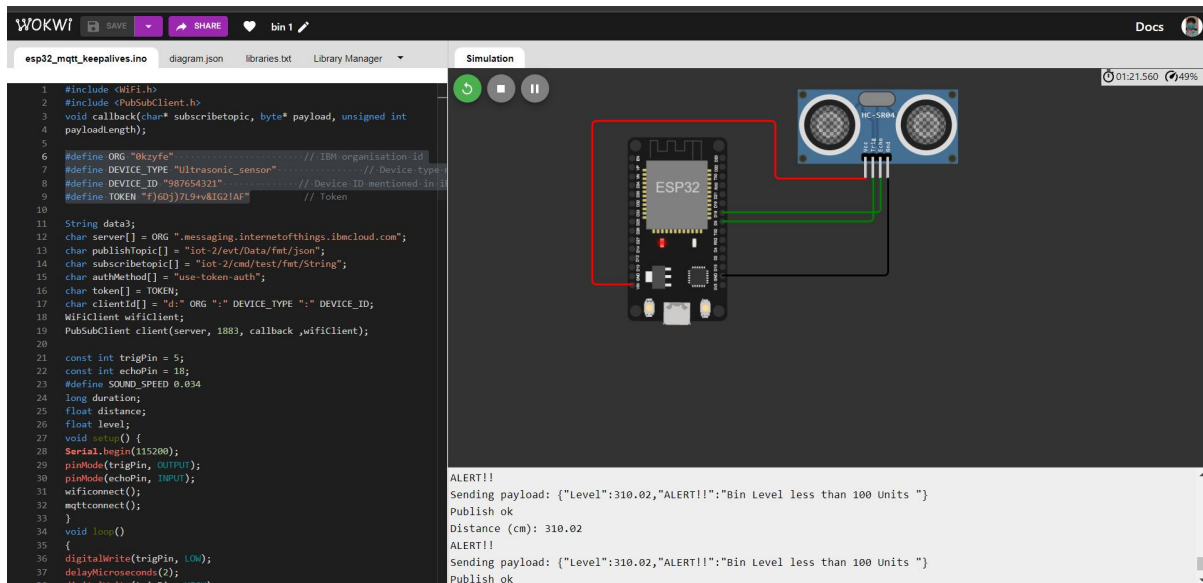


- ✓ The location of the bin is shown in The World Map



- ✓ The SENSOR values sent to the IBM Watson IOT platform by two ways:
  - From ESP32-Wokwi
  - From Python IDLE-Random Values

- ✓ To send sensor readings from ESP32-Wokwi



Run The ESP32 code Here : [Wokwi](#)

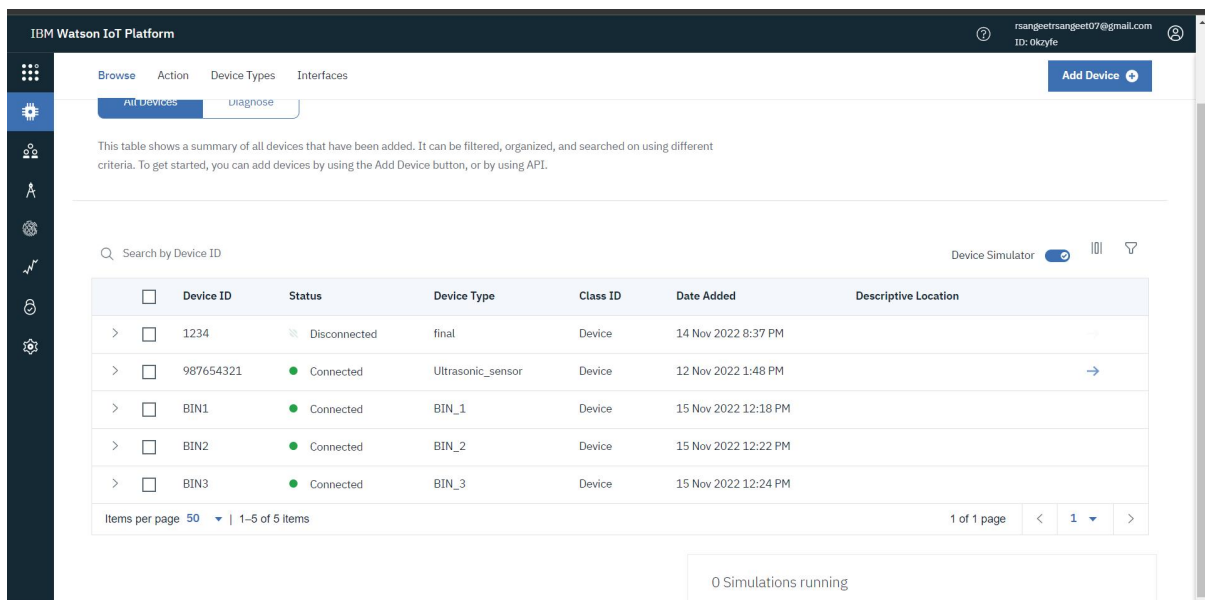
- ✓ Run the Python using IDLE to send the Random Values of Level and Weight of the Trash in the bin to IBM Watson IOT platform

## TESTING

- ✓ Executing codes for multiple Bins

```
Command Prompt - python BII x Command Prompt - python BII x Command Prompt - python BII x Command Prompt - python ult x + v
ALERT !! Weight is HIGH
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 10}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 9}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 3}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 7}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 8}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 9}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 0}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 7}
ALERT !! Weight is HIGH
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 10}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 9}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 6}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 5}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 0}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 8}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 11}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 6}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 0}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 0}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 8}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 2}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 4}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 3}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 0}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 3}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 1}
Published data Successfully: %s {'name': 'Bin_1', 'lat': 13.092677, 'lon': 80.188314, 'Weight': 1}
```

- ✓ For Example, Three IOT device is create in Watson IOT platform and also connected with the sensor Devices or IDLE . It shows the device is connected

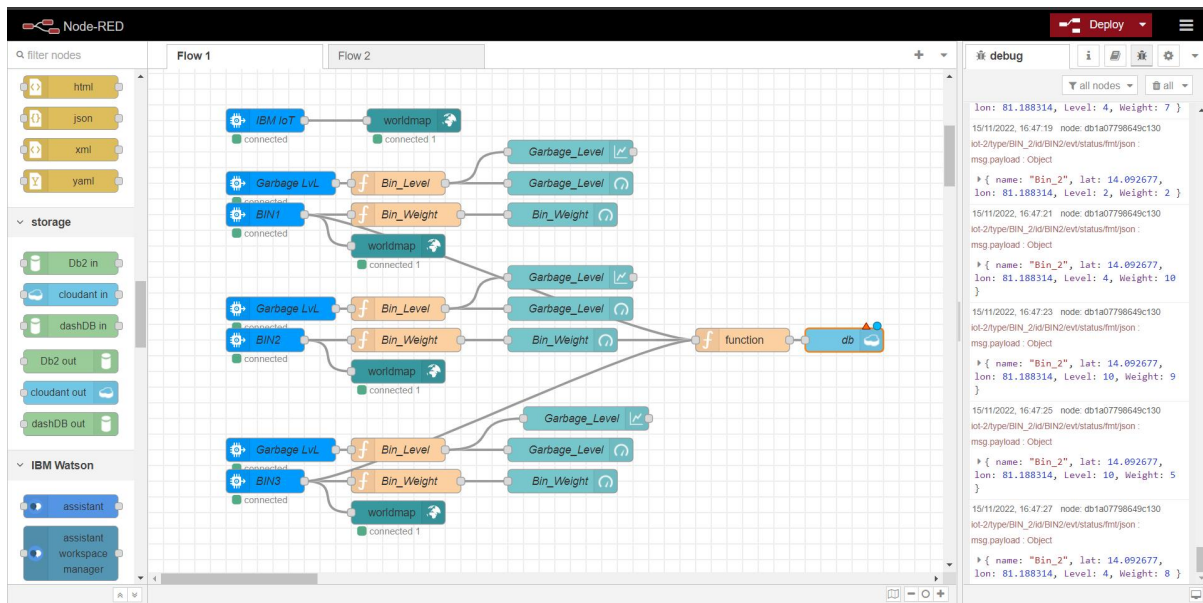


The screenshot shows the IBM Watson IoT Platform interface. The top navigation bar includes 'Browse', 'Action', 'Device Types', and 'Interfaces'. The 'All Devices' tab is selected. A table displays a summary of added devices, which can be filtered, organized, and searched. The table has columns for Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. There are five devices listed: 1234 (Disconnected), 987654321 (Connected), BIN1 (Connected), BIN2 (Connected), and BIN3 (Connected). The status of each device is indicated by a colored dot (grey for disconnected, green for connected). The 'Descriptive Location' column shows the device type (final, Ultrasonic\_sensor, BIN\_1, BIN\_2, BIN\_3). The bottom of the interface shows '0 Simulations running'.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
1234	Disconnected	final	Device	14 Nov 2022 8:37 PM	
987654321	Connected	Ultrasonic_sensor	Device	12 Nov 2022 1:48 PM	
BIN1	Connected	BIN_1	Device	15 Nov 2022 12:18 PM	
BIN2	Connected	BIN_2	Device	15 Nov 2022 12:22 PM	
BIN3	Connected	BIN_3	Device	15 Nov 2022 12:24 PM	

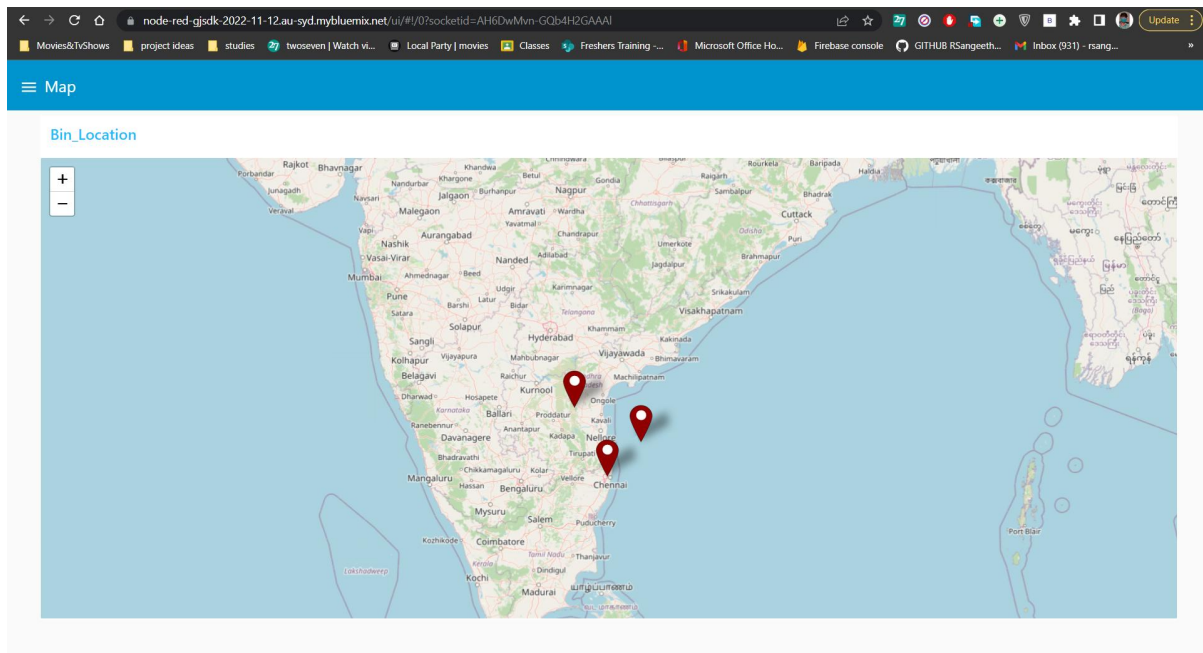
- ✓ The payload received to the IOT device in IBM platform as shown below
- ✓ To Monitor the level and Weight of the bins , the Node-RED platform is used.  
The Nodes are used to make connections.
- ✓ IBM IOT Node is to receive message from IBM IOT device.
- ✓ Several Functions , Templates , Gauge , Graph Nodes are used to create UI

## Web UI : Smart Waste Management

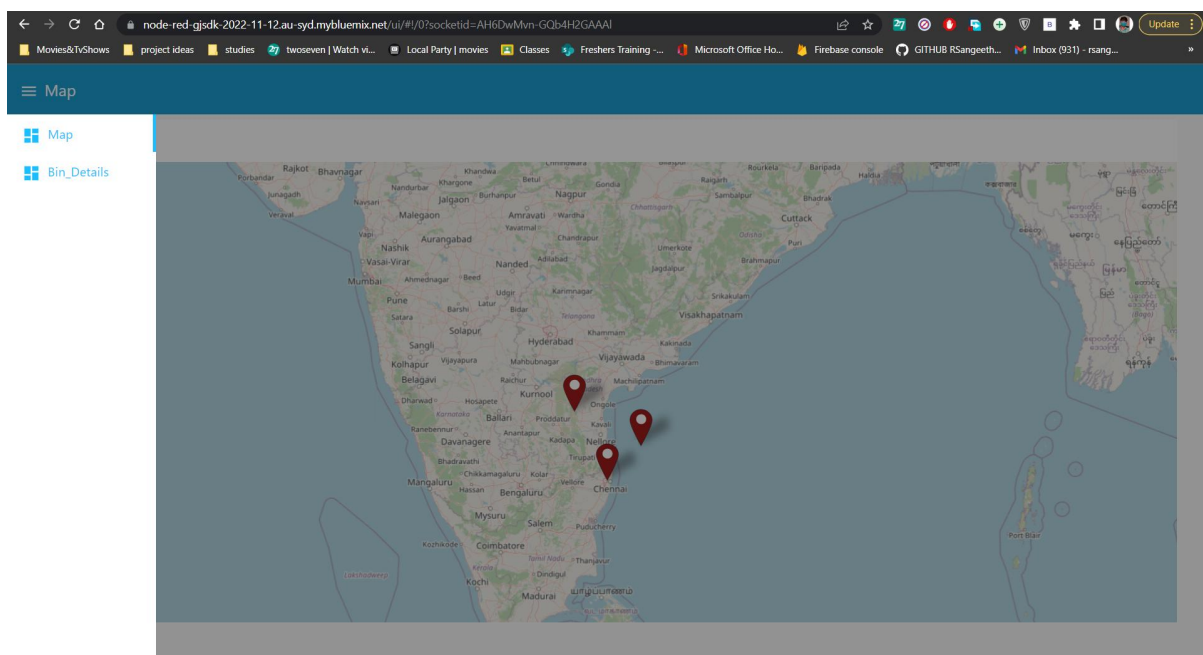


- ✓ The Web UI is created using Node-RED services , The Dashboard is used to  
Navigate between the Number of Bins

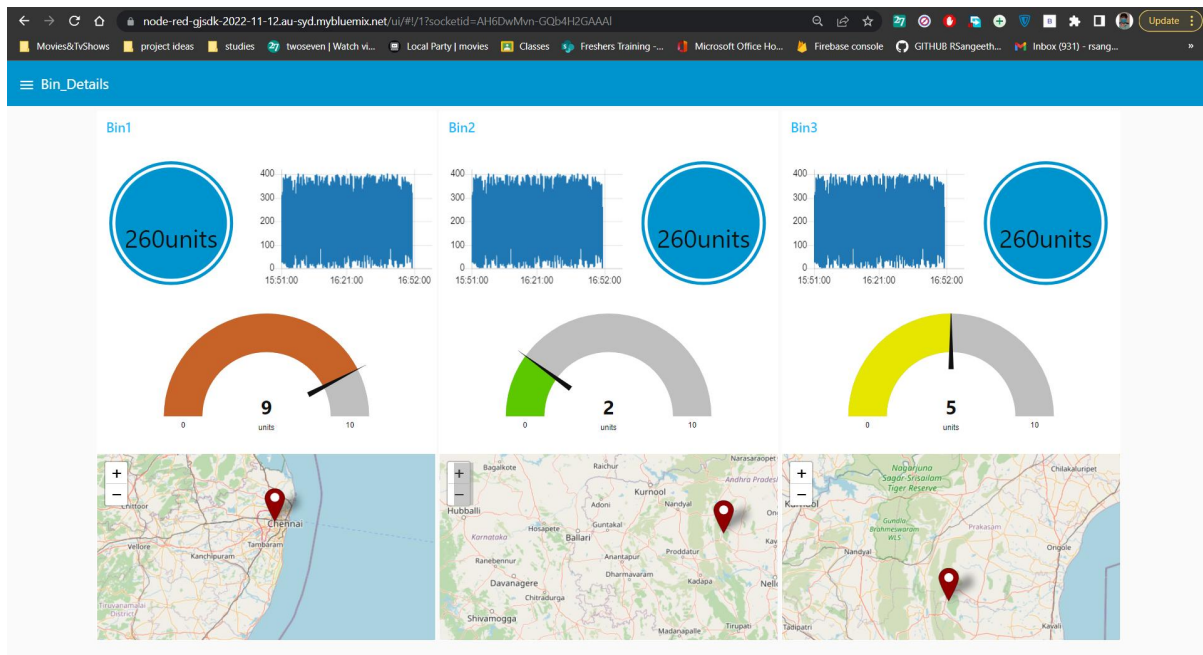




- ✓ The UI is used to monitor the garbage by weight and Level of the Trash as shown Below.
- ✓ The previous Level of the bins can also be viewed using the chart



- ✓ If the level of the Trash is FULL or the Weight of the Trash is MAXIMUM , The ALERT message is Displayed



4a1997ca-6d07-4fd6-8961-167f542618a7-blumix.cloudant.com/dashboard.html#database/project1/\_all\_docs

project1

Document ID

Options

{ } JSON

Create Document

deviceId	deviceType	payload	_id	format
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	0470b9cbbac059d400474c372d5585a3	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	150649d9eb12f72aa933a28fbaf60a24	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	15eece3341d91f4b9e5b6e2c9f2e9f5c0	json
Bin_3	Bin	{"date": "13-10-2022", "ALERT": "Bin 3 is FULL"}	310f17c359885f5b706af7128faf3fbd	json
Bin_2	Bin	{"date": "13-10-2022", "ALERT": "Bin 2 is FULL"}	3797f7920a0f1a490a96a1afe3a57324	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	3e0c113afc4b74a32d5a950618e59230	json
Bin_2	Bin	{"date": "13-10-2022", "ALERT": "Bin 2 is FULL"}	425bff2bc6db02938e18ac71185ad3bc	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	486e091eb703125a9e49bb5233da89b9	json
Bin_1	Bin	{"date": "13-10-2022", "ALERT": "Bin 1 is FULL"}	5b7bfa5f080001686443735fbd1eec85	json

Showing 5 of 8 columns. ☐ Show all columns.

Showing document 1 - 96. Documents per page: 100

✓ The Message stored in cloud contains Bin details , Date and The ALERT message



