

SPRINT – 4

Assignment Date	15 NOVEMBER 2022
Team ID	PNT2022TMID41897
Project Tittle	Smart Waste Management system for metropolitan cities

Create a node red application UI.

Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
import sys

#Provide your IBM Watson Device Credentials
organization = "106aa2"
deviceType = "Smartgarbagebins"
deviceId = "Bin1"
authMethod = "token"
authToken = "Sakthi@2001"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "lighton":
        print("led in on")
    else :
        print ("led is off")
    try:
        deviceOptions = {"org": organization, "type": deviceType, "id":
            deviceId, "auth-method":authMethod, "auth-token": authToken}
        deviceCli = ibmiotf.device.Client(deviceOptions)
```

```
#..... except
```

Exception as e:

```
print("Caught exception connecting device: %s" % str(e))
```

```
sys.exit()
```

```
#Connect and send a datapoint "hello" with value "world" into the  
cloud as an event of type "greeting" 10 times deviceCli.connect()
```

```
while True:
```

```
    #Get Sensor Data from DHT11    time.sleep(5)
```

```
    ult_son=random.randint(0,80)    weight=random.randint(0,100)
```

```
    lat = round(random.uniform(12.03, 13.50), 6)    lon =
```

```
    round(random.uniform(80.80, 85.90), 6)    data = {'Ultrasonic' :
```

```
    ult_son, 'Weight' : weight , 'lat' : lat,'lon':lon}
```

```
    #print data    def
```

```
myOnPublishCallback():
```

```
    print ("Published Ultrasonic :%s Cm" %ult_son, "Weight:%s kg "  
    %weight, "lat: %s" %lat,"lon: %s" %lon)    success =
```

```
    deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
```

```
    on_publish=myOnPublishCallback)    if not success:5    print("Not  
connected to IoT")    time.sleep(1)
```

```
    deviceCli.commandCallback = myCommandCallback #
```

Disconnect the device and application from the cloud

```
deviceCli.disconnect()
```

```

[A testipy - D:\testipy (196)]
File Edit Format Run Options Window Help
import random
import sys
#Provide your IBM Watson Device Credentials
organization = "a7mbe1"
deviceType = "testipagebina"
deviceId = "B1a1"
authMethod = "token"
authToken = "a1a1a1a200a1"
# Initialize MQTT
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == "lighton":
        print("led is on")
    else:
        print("led is off")
try:
    deviceOptions = {"org": organization, "type": deviceType, "id": deviceId, "auth-method": authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
#.....
except Exception as e:
    print("Caught exception connecting device: %s" % str(e))
    sys.exit()
#Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times
deviceCli.connect()
while True:
    #Get Sensor Data from IHT11
    time.sleep(5)
    ult=random.randint(0,80)
    weight=random.randint(0,100)
    lat = round(random.uniform(12.00, 12.50), 6)
    lon = round(random.uniform(80.00, 85.00), 6)
    data = {'Ultrasonic': ult, 'weight': weight, 'lat': lat, 'lon': lon}
    #print data
    def myOnPublishCallback():
        print("Published Ultrasonic: %s On" % ult, "weight: %s" % weight, "lat: %s" % lat, "lon: %s" % lon)
    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
    on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoT")
        time.sleep(1)
    deviceCli.commandCallback = myCommandCallback
# Disconnect the device and application from the cloud
deviceCli.disconnect()

```

Ln 37 Col 13

Python Output:

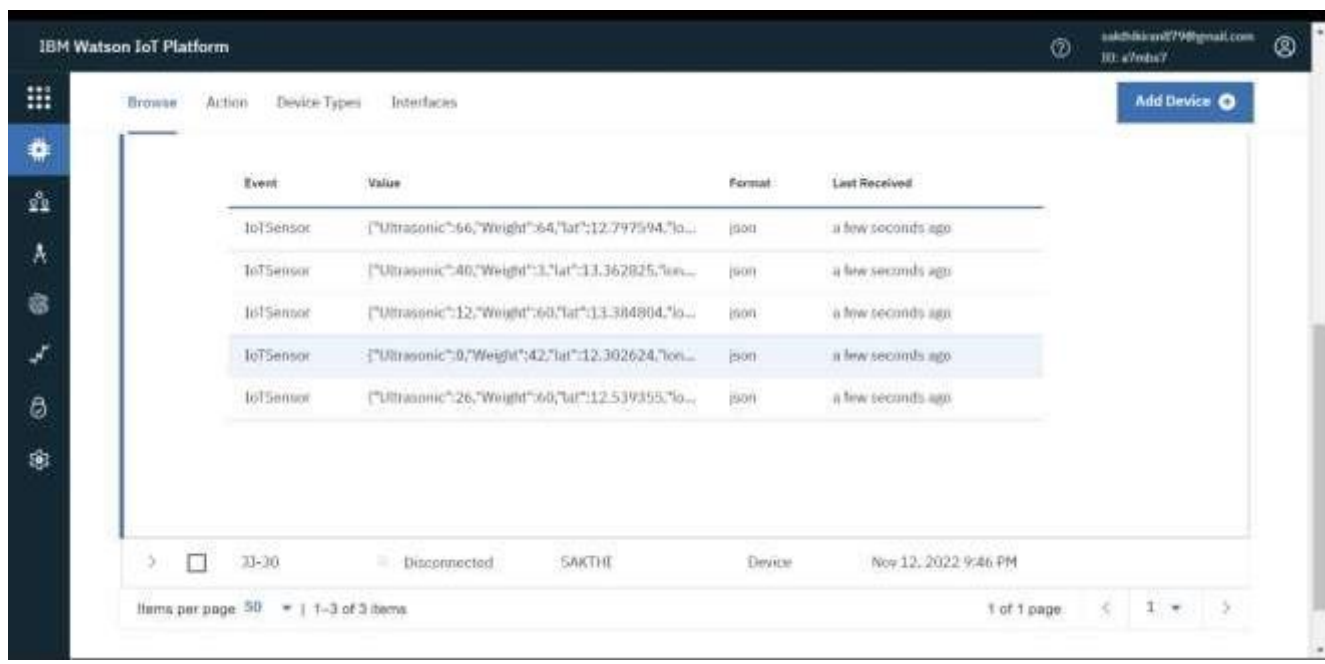
```

[A testipy (196)]
File Edit Shell Debug Options Window Help
Published Ultrasonic: 119 On weight: 94 kg lat: 12.042004 lon: 81.75998
Published Ultrasonic: 128 On weight: 96 kg lat: 12.152704 lon: 81.549303
Published Ultrasonic: 140 On weight: 117 kg lat: 12.3003 lon: 82.557953
Published Ultrasonic: 150 On weight: 94 kg lat: 12.723003 lon: 85.083139
Published Ultrasonic: 128 On weight: 113 kg lat: 12.262493 lon: 81.709150
Published Ultrasonic: 134 On weight: 91 kg lat: 12.912404 lon: 84.487678
Published Ultrasonic: 149 On weight: 121 kg lat: 12.063965 lon: 81.004956
Published Ultrasonic: 10 On weight: 68 kg lat: 12.910533 lon: 82.580219
Published Ultrasonic: 18 On weight: 57 kg lat: 12.281238 lon: 81.501333
Published Ultrasonic: 178 On weight: 84 kg lat: 12.713313 lon: 83.590587
Published Ultrasonic: 184 On weight: 126 kg lat: 12.016945 lon: 85.233295
Published Ultrasonic: 15 On weight: 118 kg lat: 12.902573 lon: 84.252482
Published Ultrasonic: 122 On weight: 127 kg lat: 12.88435 lon: 80.935229
Published Ultrasonic: 113 On weight: 93 kg lat: 12.468756 lon: 82.385753
Published Ultrasonic: 122 On weight: 61 kg lat: 12.465391 lon: 81.9176
Published Ultrasonic: 18 On weight: 6 kg lat: 11.31219 lon: 84.673749
Published Ultrasonic: 14 On weight: 41 kg lat: 12.997688 lon: 83.85843
Published Ultrasonic: 173 On weight: 21 kg lat: 12.765918 lon: 81.121809
Published Ultrasonic: 176 On weight: 89 kg lat: 12.485634 lon: 85.797305
Published Ultrasonic: 167 On weight: 30 kg lat: 12.466667 lon: 85.585160
Published Ultrasonic: 154 On weight: 33 kg lat: 12.295682 lon: 82.997133
Published Ultrasonic: 121 On weight: 61 kg lat: 12.349312 lon: 81.816453
Published Ultrasonic: 115 On weight: 165 kg lat: 12.42828 lon: 85.101653
Published Ultrasonic: 177 On weight: 96 kg lat: 12.350439 lon: 84.924801
Published Ultrasonic: 141 On weight: 99 kg lat: 12.122203 lon: 82.085698
Published Ultrasonic: 180 On weight: 47 kg lat: 12.320411 lon: 85.912242
Published Ultrasonic: 158 On weight: 86 kg lat: 12.025076 lon: 82.442491
Published Ultrasonic: 131 On weight: 58 kg lat: 12.573036 lon: 82.023028
Published Ultrasonic: 161 On weight: 92 kg lat: 12.175432 lon: 85.656471
Published Ultrasonic: 153 On weight: 120 kg lat: 12.421729 lon: 83.681535
Published Ultrasonic: 174 On weight: 145 kg lat: 12.469798 lon: 84.773026
Published Ultrasonic: 174 On weight: 143 kg lat: 12.357722 lon: 83.80607
Published Ultrasonic: 168 On weight: 115 kg lat: 12.100682 lon: 84.964771
Published Ultrasonic: 165 On weight: 129 kg lat: 12.585533 lon: 85.863665
Published Ultrasonic: 118 On weight: 100 kg lat: 12.489128 lon: 82.152936
Published Ultrasonic: 168 On weight: 70 kg lat: 12.16198 lon: 82.081189
Published Ultrasonic: 160 On weight: 118 kg lat: 12.471483 lon: 80.985271
Published Ultrasonic: 118 On weight: 65 kg lat: 12.623412 lon: 81.955765
Published Ultrasonic: 178 On weight: 60 kg lat: 12.266666 lon: 83.276115
Published Ultrasonic: 144 On weight: 36 kg lat: 12.599836 lon: 85.200905
Published Ultrasonic: 152 On weight: 47 kg lat: 12.716786 lon: 82.161738
Published Ultrasonic: 168 On weight: 178 kg lat: 12.48661 lon: 81.426264
Published Ultrasonic: 183 On weight: 129 kg lat: 12.444952 lon: 84.373957
Published Ultrasonic: 168 On weight: 131 kg lat: 12.134231 lon: 82.323981

```

Ln 45 Col 0

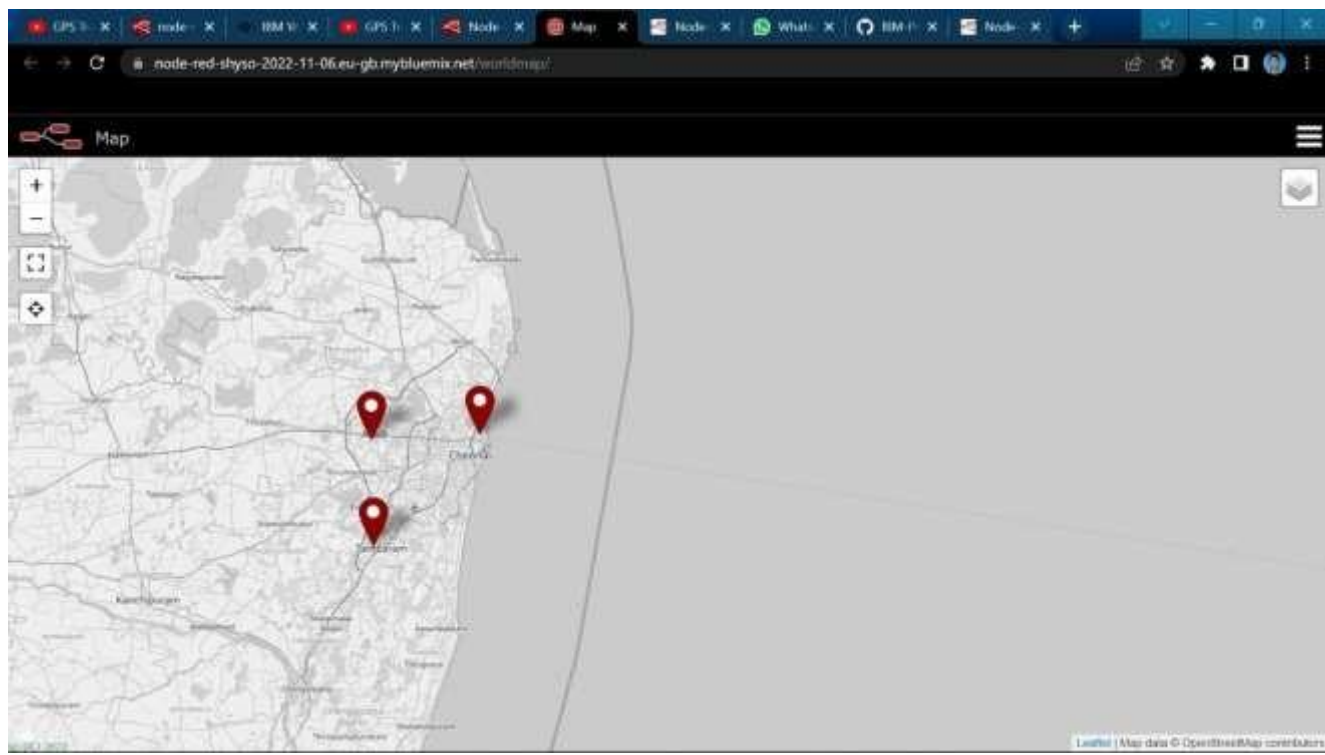
IBM Watson Output:



The screenshot shows the IBM Watson IoT Platform interface. At the top, there's a navigation bar with 'Browse', 'Action', 'Device Types', and 'Interfaces'. A sidebar on the left contains various icons. The main area displays a table of IoT sensor data. The table has four columns: 'Event', 'Value', 'Format', and 'Last Received'. Below the table, there's a status bar showing '20-20', 'Disconnected', 'SAKTHI', 'Device', and 'Nov 12, 2022 9:46 PM'. At the bottom, there's a pagination bar showing 'Items per page: 50' and '1-3 of 3 items'.

Event	Value	Format	Last Received
IoT Sensor	["Ultrasonic":66,"Weight":64,"lat":12.797594,"lon...	json	a few seconds ago
IoT Sensor	["Ultrasonic":46,"Weight":1,"lat":13.362825,"lon...	json	a few seconds ago
IoT Sensor	["Ultrasonic":12,"Weight":60,"lat":13.384804,"lon...	json	a few seconds ago
IoT Sensor	["Ultrasonic":0,"Weight":42,"lat":12.302624,"lon...	json	a few seconds ago
IoT Sensor	["Ultrasonic":26,"Weight":60,"lat":12.539355,"lon...	json	a few seconds ago

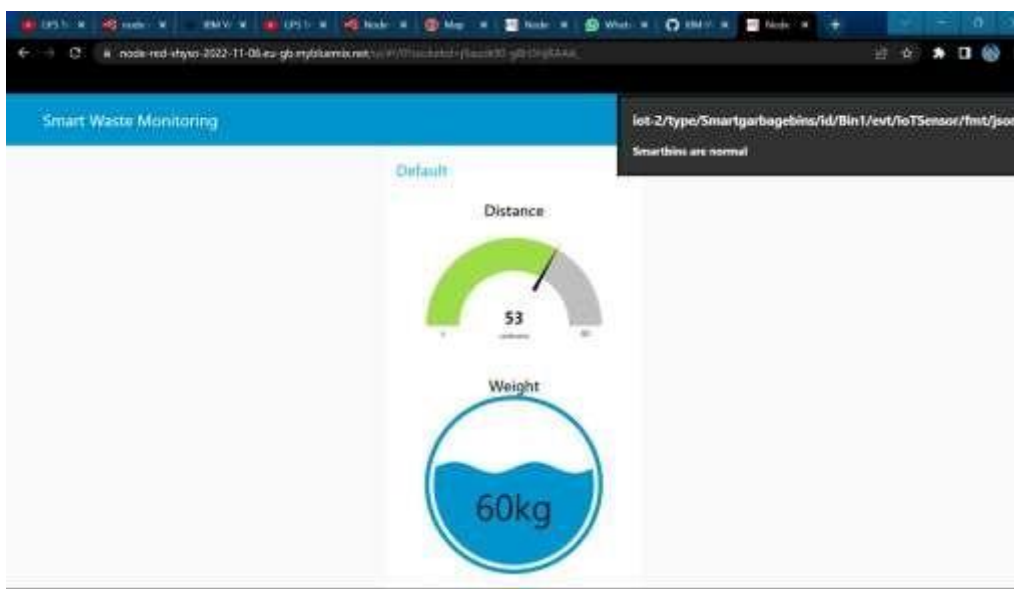
Web UI map:

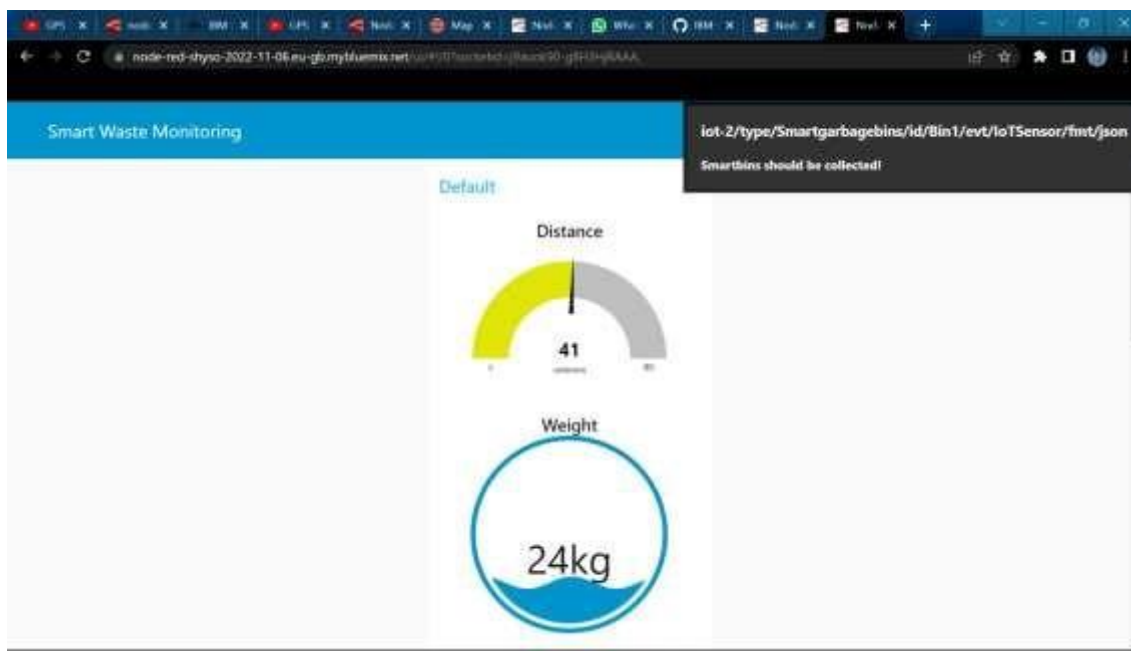
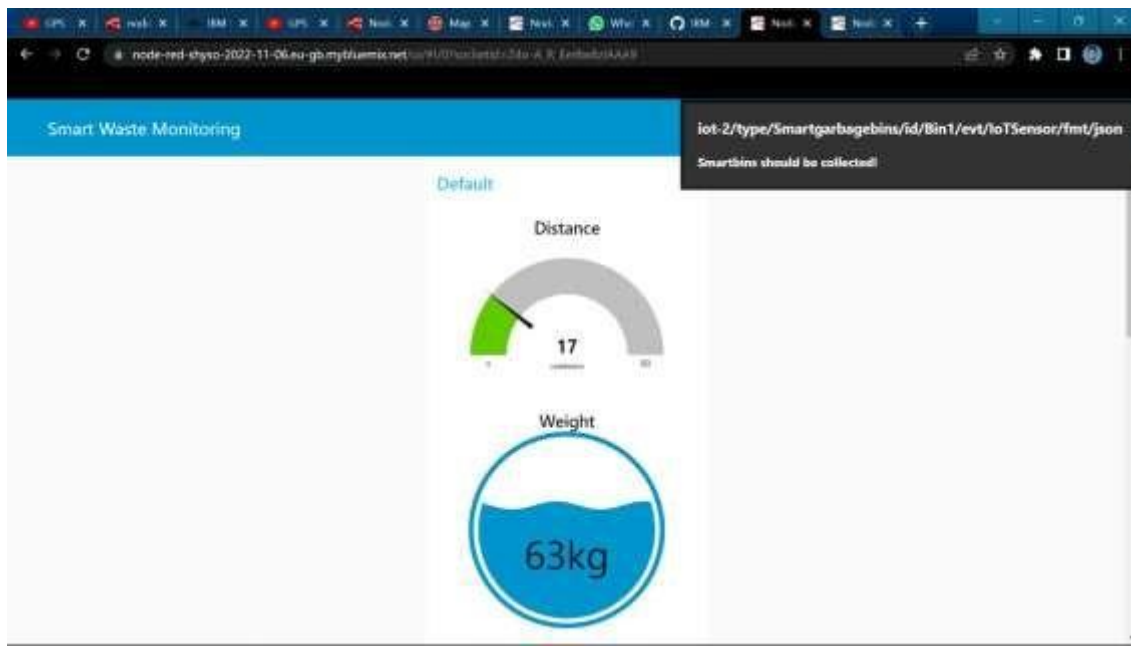




Smart bins are in normal weight:

Smart bins should be collected:





Result:

The node red application was successfully created and the weight of the smart bins are collected and the alert notifications are created successfully.