

# VIT-IoT (INDUSTRY CERTIFICATE INTERNSHIP PROGRAM)

## ASSIGNMENT -3

NAME: CHITTARANJAN TADIKONDA

MAIL ID: [crchintu07@gmail.com](mailto:crchintu07@gmail.com)

Develop a code to upload the water tank level and light intensity values to the IBM IoT platform and visualize them in the web application.

Python Code:

---

```
import wiotp.sdk.device
import time
import random
myConfig = {
    "identity": {
        "orgId": "k1qu6q",
        "typeId": "RPI",
        "deviceId": "003"
    },
    "auth": {
        "token": "789789789"
    }
}

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:
    wlevel=random.randint(0,100)
    light=random.randint(0,100)
    myData={'Water_Level':wlevel, 'Light_Intensity':light}
    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()
```

## Code:

```
import wiotp.sdk.device
import time
import random

myConfig = {
    "identity": {
        "orgId": "klqu6q",
        "typeId": "RPI",
        "deviceId": "003"
    },
    "auth": {
        "token": "789789789"
    }
}

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:
    wlevel=random.randint(0,100)
    light=random.randint(0,100)
    myData={'Water_Level':wlevel, 'Light_Intensity':light}
    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()

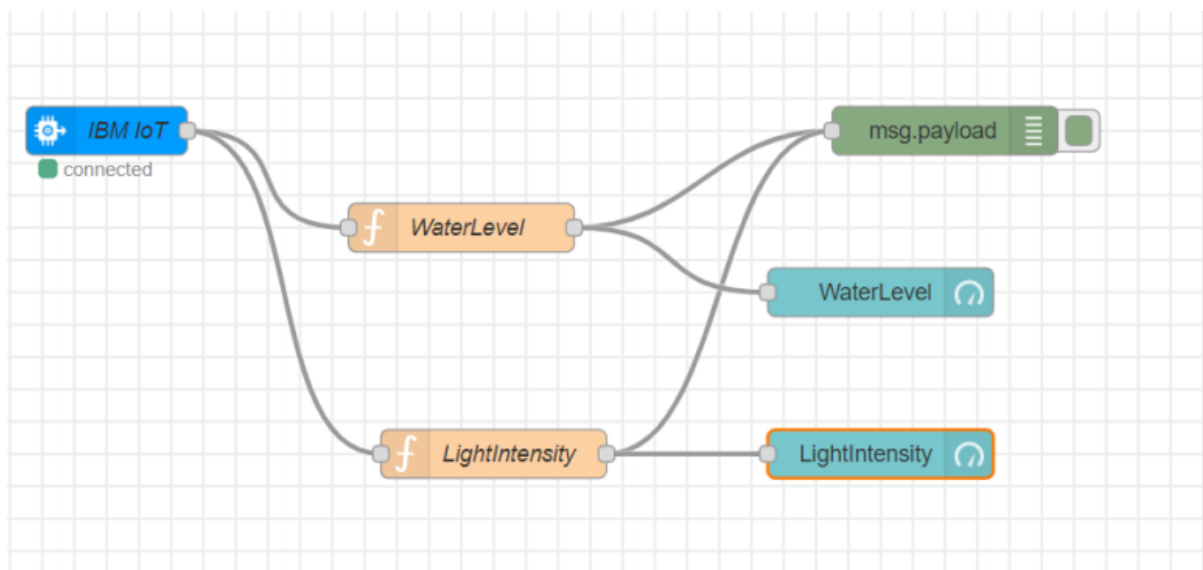
```

```

Published data Successfully: %s {'Water_Level': 41, 'Light_Intensity': 53}
Published data Successfully: %s {'Water_Level': 2, 'Light_Intensity': 4}
Published data Successfully: %s {'Water_Level': 14, 'Light_Intensity': 89}
Published data Successfully: %s {'Water_Level': 1, 'Light_Intensity': 54}
Published data Successfully: %s {'Water_Level': 52, 'Light_Intensity': 97}
Published data Successfully: %s {'Water_Level': 99, 'Light_Intensity': 77}
Published data Successfully: %s {'Water_Level': 30, 'Light_Intensity': 73}
Published data Successfully: %s {'Water_Level': 91, 'Light_Intensity': 85}
Published data Successfully: %s {'Water_Level': 45, 'Light_Intensity': 98}

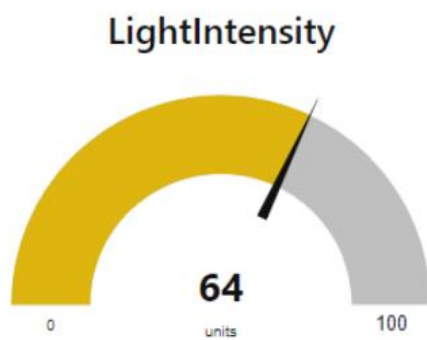
```

Output of python code: sending its data values to the device

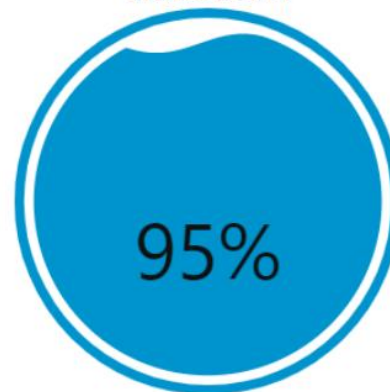


Node Red Flow chart

Sensor



WaterLevel



webpage is also receiving the same data produced by the random variables in python