Develop A Python Script To Publish And Subscribe To IBM IoT Platform

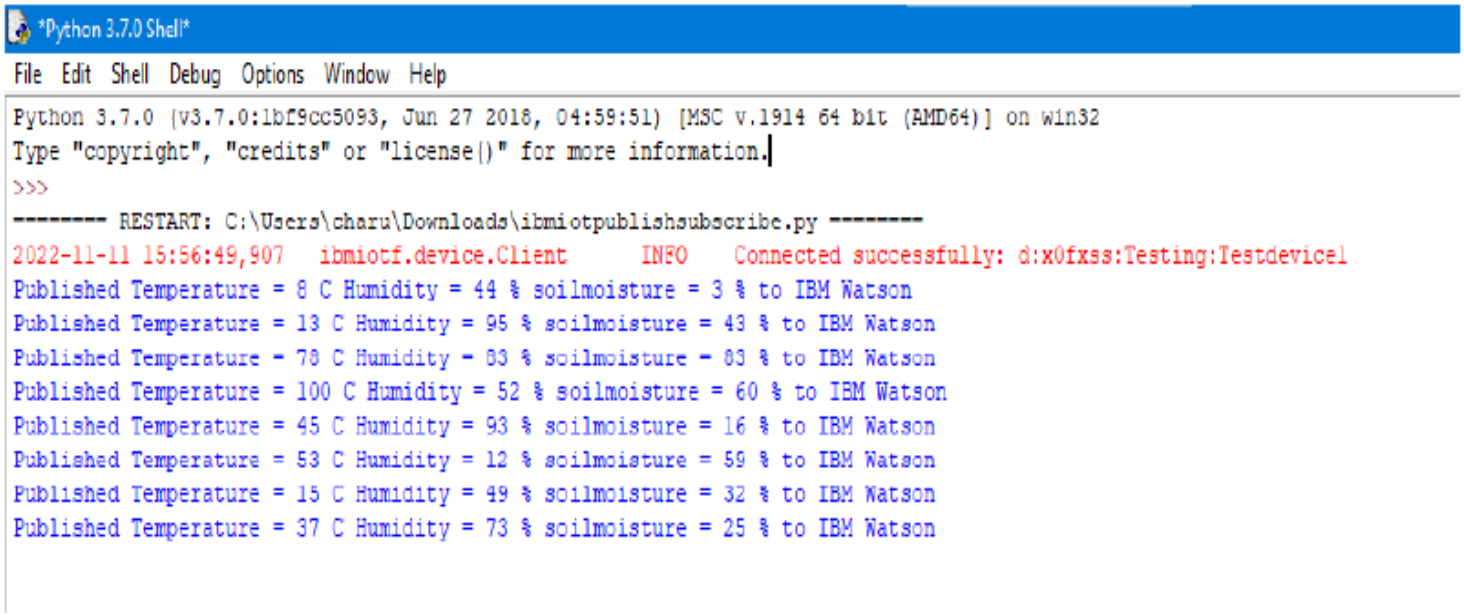
TEAM ID: PNT2022TMID34370

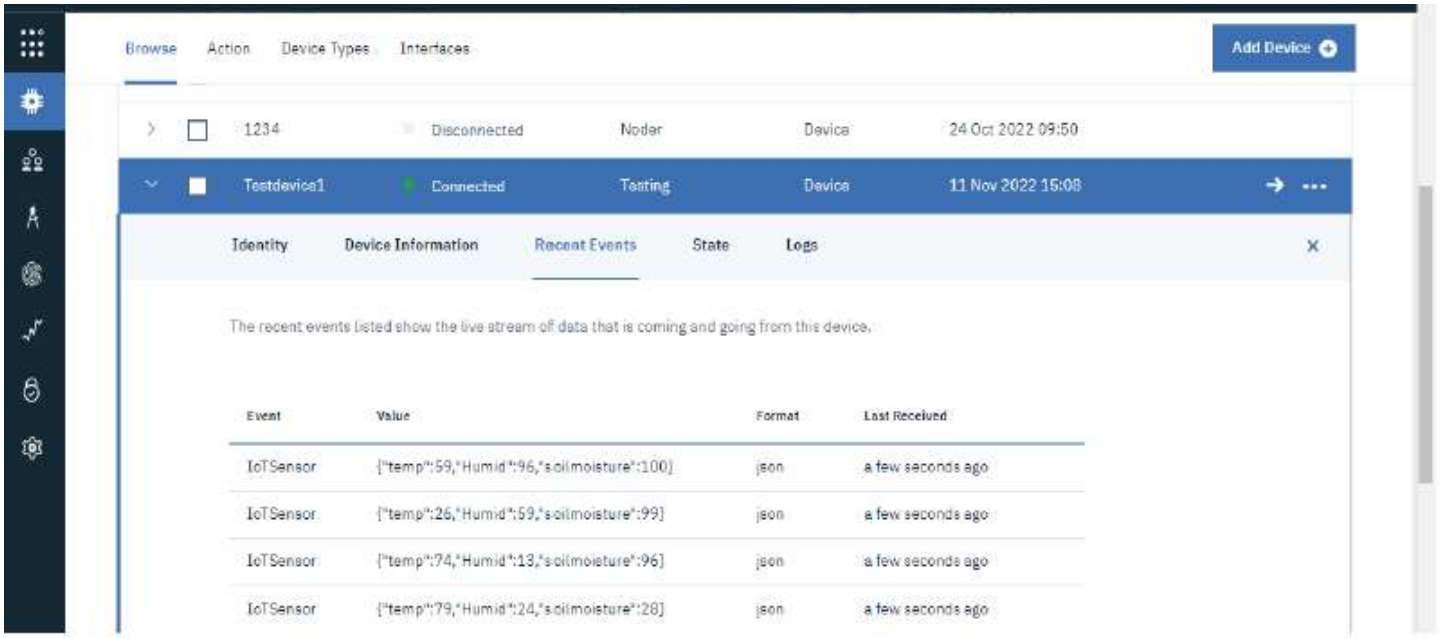
PYTHON CODE;

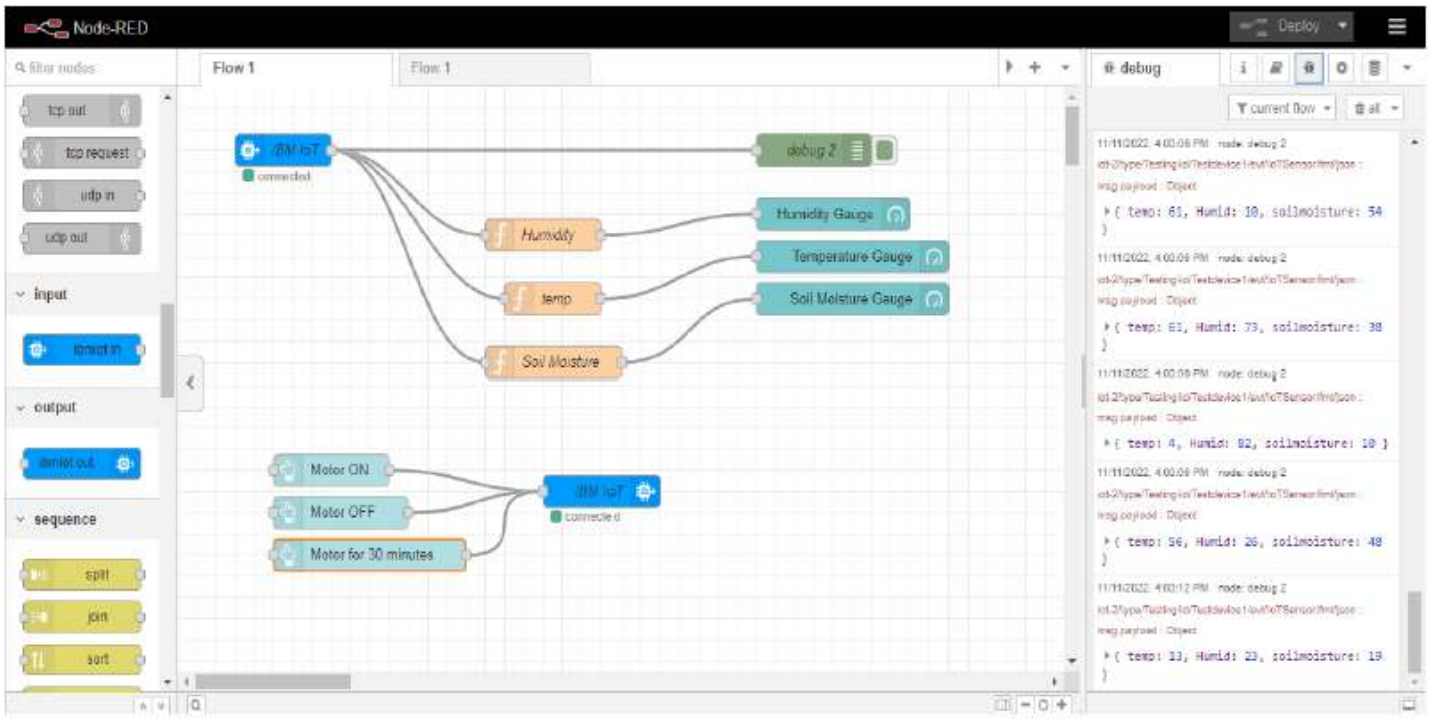
```
import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "x0fxss" #replace the ORG ID

deviceType = "Testing"#replace the Device type wi

deviceId = "Testdevice1"#replace Device ID

authMethod = "token"

authToken = "123456789"

#Replace the authtoken # Initialize GPIO #Receives Command from Node-red

def myCommandCallback(cmd):

        print ("Command received: %s" % cmd.data['command'])

        status=cmd.data['command']

        if status=="motoron":

                print ("motor is on")

        elif status == "motoroff" :

                print ("motor is off")

        elif status == "motor30" :

                print ("motor is on for 30 minutes")

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

        temp=random.randint(0,100)

        Humid=random.randint(0,100)

        soilmoisture=random.randint(0,100)

        data = { 'temp' : temp, 'Humid': Humid, 'soilmoisture': soilmoisture }       #print data

        def myOnPublishCallback():

                print ("Published Temperature = %s C" % temp, "Humidity = %s %%" % Humid, "soilmoisture = %s %%" %soilmoisture, "to IBM Watson")

        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)

        if not success:

                print("Not connected to IoTF")

                time.sleep(5) deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()
```

OUTPUT :




Browse
Action
Device Types
Interfaces
Add Device
1234
Disconnected
Noder
Device
24 Oct 2022 09:50
Testdevice1
Connected
Testing
Device
11 Nov 2022 15:08
Identity
Device Information
Recent Events
State
Logs
The recent events listed show the live stream of data that is coming and going from this device.
Event
Value
Format
Last Received
IoTSensor
{"temp":59,"Humid":96,"soilmoisture":100}
json
a few seconds ago
IoTSensor
{"temp":26,"Humid":59,"soilmoisture":99}
json
a few seconds ago
IoTSensor
{"temp":74,"Humid":13,"soilmoisture":96}
json
a few seconds ago
IoTSensor
{"temp":79,"Humid":24,"soilmoisture":28}
json
a few seconds ago


Node-RED
Deploy
Flow 1
debug
tcp out
tcp request
udp in
udp out
input
mqtt in
output
mqtt out
sequence
split
join
sort
IBM IoT
connected
debug 2
Humidity
temp
Soil Moisture
Humidity Gauge
Temperature Gauge
Soil Moisture Gauge
Motor ON
Motor OFF
Motor for 30 minutes
IBM IoT
connected


```
Published Temperature = 25 C Humidity = 32 % soilmoisture = 86 % to IBM Watson
Published Temperature = 27 C Humidity = 16 % soilmoisture = 26 % to IBM Watson
Command received: motoron
motor is on
Command received: motoron
motor is on
Published Temperature = 10 C Humidity = 69 % soilmoisture = 82 % to IBM Watson
Published Temperature = 75 C Humidity = 37 % soilmoisture = 2 % to IBM Watson
Published Temperature = 63 C Humidity = 59 % soilmoisture = 11 % to IBM Watson
Published Temperature = 31 C Humidity = 20 % soilmoisture = 43 % to IBM Watson
Published Temperature = 47 C Humidity = 38 % soilmoisture = 95 % to IBM Watson
Published Temperature = 62 C Humidity = 5 % soilmoisture = 93 % to IBM Watson
Command received: motoroff
motor is off
Command received: motor30
motor is on for 30 minutes
Published Temperature = 19 C Humidity = 99 % soilmoisture = 96 % to IBM Watson
Published Temperature = 6 C Humidity = 56 % soilmoisture = 85 % to IBM Watson
```