DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSCRIBE TO IBM IOT PLATFORM

Team ID	PNT2022TMID35909
Project Name	GAS LEAKAGE MONITORING AND ALERTING SYSTEMS FOR INDUSTRIES

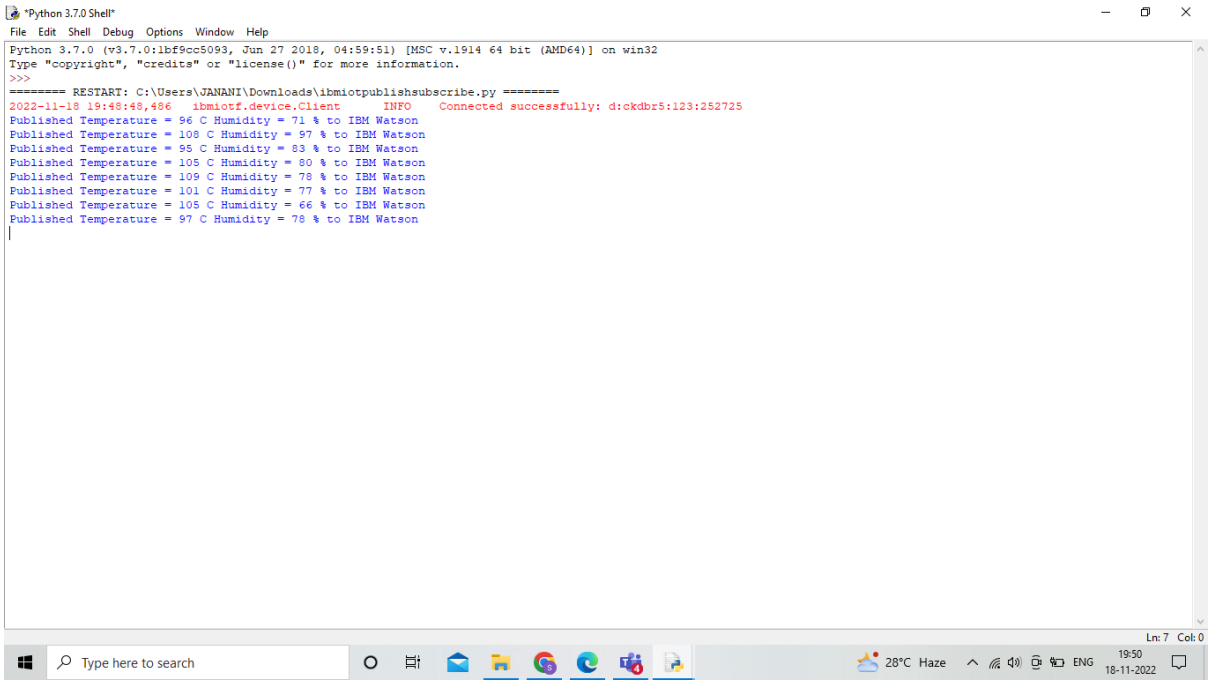
PYTHON CODE:

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

#Provide your IBM Watson Device Credentials
organization = "ckdbr5"
deviceType = "123"
deviceId = "252725"
authMethod = "token"
authToken = "27252527"
```

```
# Initialize GPIO


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")

    #print(cmd)

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(92,110)
        Humid=random.randint(60,100)


        data = { 'temp' : temp, 'Humid': Humid }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%"
% Humid, "to IBM Watson")


        success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
        if not success:
            print("Not connected to IoTF")
        time.sleep(10)


        deviceCli.commandCallback = myCommandCallback


# Disconnect the device and application from the cloud
deviceCli.disconnect()
```

OUTPUT:



PUBLISH DATA TO THE IBM CLOUD