DEVELOP A PYTHON SCRIPT TO PUBLISH AND SUBSRIBE TO IBM IOT PLATFORM.

Team ID	PNT2022TMID04728
Project Name	Smart Farmer-IOT Enabled Smart FarmingApplication

Python code:

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


#Provide your IBM Watson Device Credentials
organization = "49x4b9"
deviceType = "weather_monitor"
deviceId = "weather_today"
authMethod = "token"
authToken = "S*xL?JyVVKPwTGH_IK"


# Initialize GPIO

temp=random.randint(0,100)
pulse=random.randint(0,100)
oxygen= random.randint(0,100)
lat =  17
lon = 18


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

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

    success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0, on_publish=myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(1)

    deviceCli.commandCallback = myCommandCallback

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