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

Team ID	PNT2022TMID04728
Project Name	Smart Farmer-IoT Enabled Smart Farming Application

PYTHON CODE:

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



#Provide your IBM Watson Device Credentials
organization = "rr454u"
deviceType = "sensor_1"
deviceId = "sensor"
authMethod = "token"
authToken = "uQ@5ONlr2&eKy*pof*"



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:


      temperature=random.randint(0,100)
      humidity=random.randint(0,100)
      soil= random.randint(0,100)


      data = {'temperature' : temperature, 'humidity': humidity ,'soil':soil}
      #print data
      def myOnPublishCallback():
         print ("Published Temperature = %s C" % temperature, "Humidity = %s %%" % humidity, "soil Moisture = %s %%"% soil,"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()
```