Develop a Python Script to Publish and Subscribe to IBM IoT Platform

Date	10 November 2022
Team ID	PNT2022TMID30750
Project Name	**Project -** SmartFarmer - IoT Enabled Smart Farming Application

- Developed a Python Script to Publish and Subscribe to IBM IoT Platform.

Python Code:

```
import time
import sys
import ibmiotf.application
import ibmiotf.device
import random
#Providing the  IBM Watson Device Credentials
organization = " o58zsl "
deviceType = " abcd "
deviceId = "1234"
authMethod = "token"
authToken = "12345678"
global y

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

```
    status=cmd.data['command']
    if status=="motoron":
        print ("motor is on")
    if status=="motoroff" :
        print ("motor is off")
    if status=="manual" :
        print ("Motor Control is in Manual Mode")
    if status=="automatic" :
        print ("Motor control is in Automatic Mode")
        if soilmoisture > 600:
            print ("motor is on")

    #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()
deviceCli.connect()

while True:
        #Get Sensor Data from DHT11
        temp=random.randint(0,100)
        Humid=random.randint(0,100)
        soilmoisture=random.randint(0,1023)
        Phlevel=random.randint(0,14)
        y=soilmoisture

        data = { 'temp' : temp, 'Humid': Humid,'soilmoisture' : soilmoisture ,'Phlevel' :
Phlevel }
        #print data
        def myOnPublishCallback():
            print ("Published Temperature = %s C" % temp, "Humidity = %s %%" %
Humid,"Soil Moisture is %s %%" % soilmoisture,"PH level is %s" %Phlevel ,"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

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