Develop a Python Script

Team ID	PNT2022TMID22430
Project Name	Project – Smart waste management system for Metropolitan cities

Develop a python code for publishing the bin values to the IBM IoT Platform.

The Python script is developed and published to cloud.

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

#Provide your IBM Watson Device Credentials
organization = "32ws5h"
deviceType = "Ultrasonic_sensor"
deviceId = "554517"
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
   print("Message received from IBM IOT Platform : %s" % cmd.data['ALERT'])
   status=cmd.data['ALERT']
   if status=="BIN FULL":
      print ("Empty the bin immediately")

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()

#SENSOR DATA
```

```
Garbage_level=0
Garbage_weight=0
while True:
    #Get Sensor Data from DHT11

    Garbage_level=Garbage_level+random.randint(90,110)
    Garbage_weight=Garbage_weight+random.randint(60,100)


    data = { 'Garbage level(%)' : Garbage_level, 'Garbage weight(g)': Garbage_weight , 'Location': "10, Gandhi nagar, Adayar"}

    def myOnPublishCallback():
      print ("Published Garbage level(%) = %s %" % Garbage_level, "Garbage weight(g) = %s %" % Garbage_weight, "to IBM Watson")

    success = deviceCli.publishEvent("Ultrasonic_sensor", "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()
```

python script.py - C:/Users/Jayasri/Desktop/python script.py (3.11.0)

File Edit Format Run Options Window Help

```
authMethod = "token"
authToken = "12345678"

# Initialize GPIO
def myCommandCallback(cmd):
    print("Message received from IBM IOT Platform : %s" % cmd.data['ALERT'])
    status=cmd.data['ALERT']
    if status=="BIN FULL":
        print ("Empty the bin immediately")

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()

#SENSOR DATA

Garbage_level=0
Garbage_weight=0
while True:
        #Get Sensor Data from DHT11

        Garbage_level=Garbage_level+random.randint(90,110)
        Garbage_weight=Garbage_weight+random.randint(60,100)


        data = { 'Garbage level(%)' : Garbage_level, 'Garbage weight(g)': Garbage_weight , 'Location': "10, Gandhi nagar, Adayar"}

        def myOnPublishCallback():
            print ("Published Garbage level(%) = %s %" % Garbage_level, "Garbage weight(g) = %s %" % Garbage_weight, "to IBM Watson")

        success = deviceCli.publishEvent("Ultrasonic_sensor", "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()
```