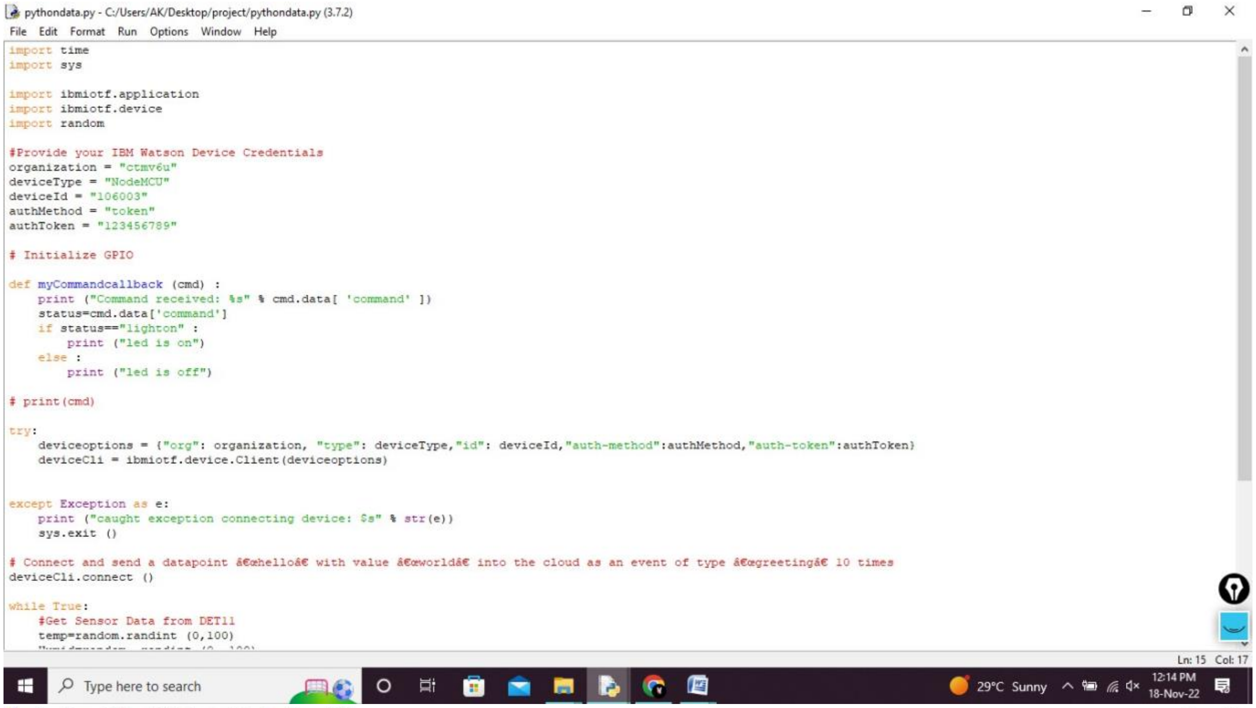
PYTHON SCRIPT

TEAM ID	PNT2022TMID48693
PROJECT NAME	PROJECT- Smart waste management system for metropolitan cities

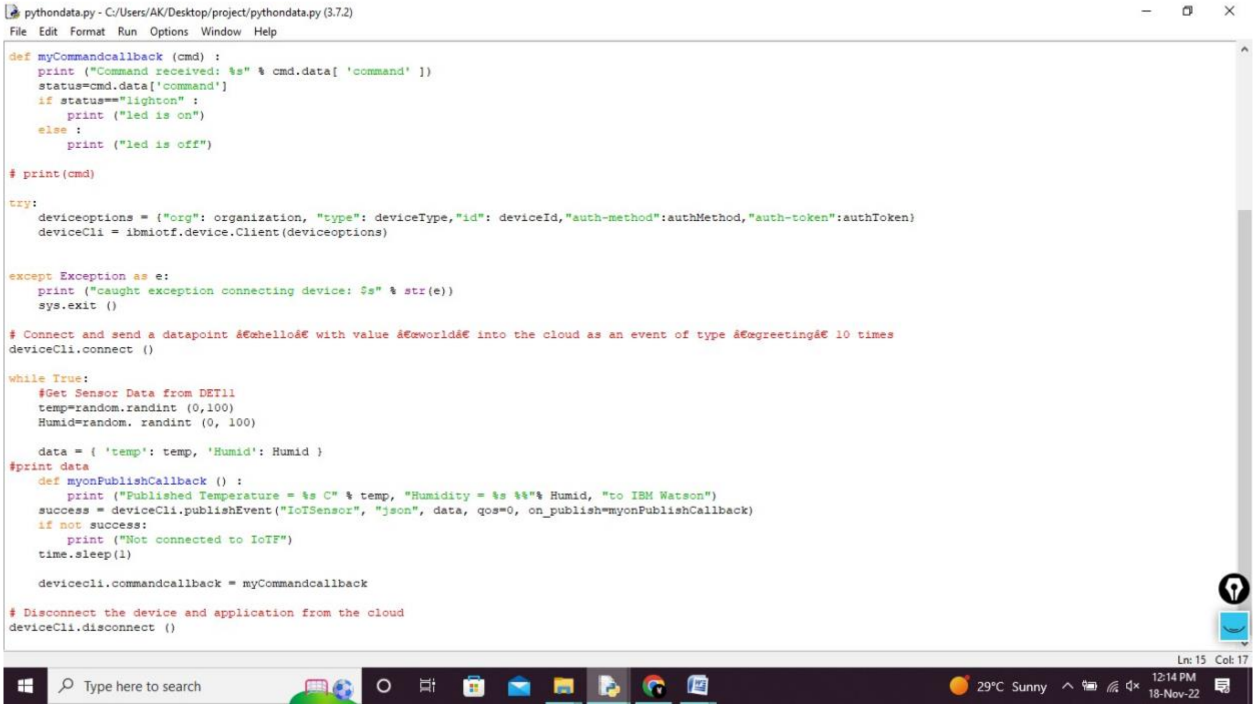
PYTHON SCRIPT:

```
import time import
sys

import ibmiotf.application
import ibmiotf.device import
random

#Provide your IBM Watson Device Credentials
organization = "fp3a19” deviceType = "sample"
deviceId = "123" authMethod = "token" authToken =
"Abcdefgh"

# Initialize GPIO

def myCommandcallback (cmd) :     print ("Command received: %s"
% cmd.data[ 'command' ])     status=cmd.data['command']     if
status=="1ighton" :        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 DET11     temp=random.randint
(0,100)
   Humid=random. randint (0, 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(1)

   devicecli.commandcallback = myCommandcallback SCREENSHOTS :
```


pythondata.py - C:/Users/AK/Desktop/project/pythondata.py (3.7.2)
File Edit Format Run Options Window Help
Ln: 15 Col: 17
Type here to search
29°C Sunny
12:14 PM
18-Nov-22

```
import time
import sys

import ibmiotf.application
import ibmiotf.device
import random

#Provide your IBM Watson Device Credentials
organization = "ctmv6u"
deviceType = "NodeMCU"
deviceId = "106003"
authMethod = "token"
authToken = "123456789"

# 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 â€œhelloâ€ with value â€œworldâ€ into the cloud as an event of type â€œgreetingâ€ 10 times
deviceCli.connect ()

while True:
    #Get Sensor Data from DET11
    temp=random.randint (0,100)
```


pythondata.py - C:/Users/AK/Desktop/project/pythondata.py (3.7.2)
File Edit Format Run Options Window Help
Ln: 15 Col: 17
Type here to search
29°C Sunny
12:14 PM
18-Nov-22

```
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 â€œhelloâ€ with value â€œworldâ€ into the cloud as an event of type â€œgreetingâ€ 10 times
deviceCli.connect ()

while True:
    #Get Sensor Data from DET11
    temp=random.randint (0,100)
    Humid=random. randint (0, 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(1)

    devicecli.commandcallback = myCommandcallback

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

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