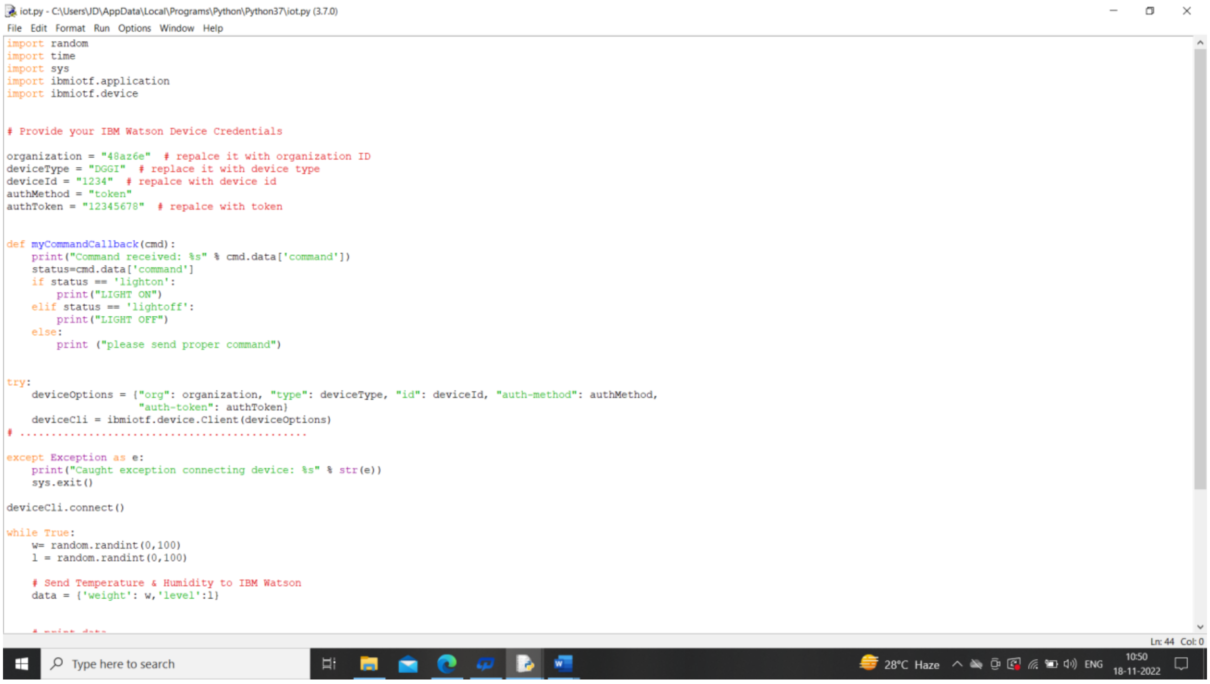
DEVELOP THE PYTHON SCRIPT

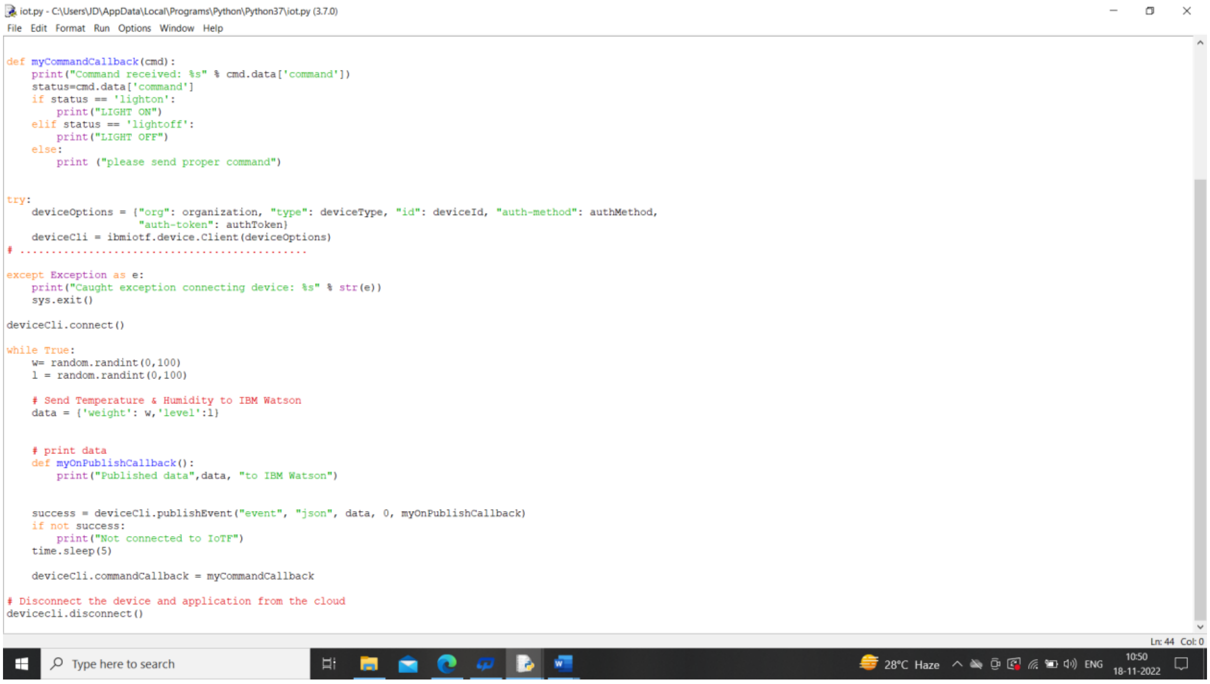
TEAM ID	PNT2022TMID06962
PROJECT NAME	Smart Waste managmentsystem for metropolitan cities

PYTHON CODE:



iot.py - C:\Users\JD\AppData\Local\Programs\Python\Python37\iot.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 44 Col: 0
Type here to search
28°C Haze
ENG
10:50
18-11-2022

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


# Provide your IBM Watson Device Credentials

organization = "48az6e"  # repalce it with organization ID
deviceType = "DGGI"  # replace it with device type
deviceId = "1234"  # repalce with device id
authMethod = "token"
authToken = "12345678"  # repalce with token


def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == 'lighton':
        print("LIGHT ON")
    elif status == 'lightoff':
        print("LIGHT OFF")
    else:
        print ("please send proper command")


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:
    w= random.randint(0,100)
    l = random.randint(0,100)

    # Send Temperature & Humidity to IBM Watson
    data = {'weight': w,'level':l}


    # print data
```



iot.py - C:\Users\JD\AppData\Local\Programs\Python\Python37\iot.py (3.7.0)
File Edit Format Run Options Window Help
Ln: 44 Col: 0
Type here to search
28°C Haze
ENG
10:50
18-11-2022

```
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == 'lighton':
        print("LIGHT ON")
    elif status == 'lightoff':
        print("LIGHT OFF")
    else:
        print ("please send proper command")


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:
    w= random.randint(0,100)
    l = random.randint(0,100)

    # Send Temperature & Humidity to IBM Watson
    data = {'weight': w,'level':l}


    # print data
    def myOnPublishCallback():
        print("Published data",data, "to IBM Watson")


    success = deviceCli.publishEvent("event", "json", data, 0, myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(5)

    deviceCli.commandCallback = myCommandCallback

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

CODE

```
import random

import time

import sys

import ibmiotf.application

import ibmiotf.device



# Provide your IBM Watson Device Credentials


organization = "48az6e"  # repalce it with organization ID

deviceType = "DGGI"  # replace it with device type

deviceId = "1234"  # repalce with device id

authMethod = "token"
```

```
authToken = "12345678"  # repalce with token



def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']
    if status == 'lighton':
        print("LIGHT ON")
    elif status == 'lightoff':
        print("LIGHT OFF")
    else:
        print ("please send proper command")



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:
    w= random.randint(0,100)
    l = random.randint(0,100)

    # Send Temperature & Humidity to IBM Watson
    data = {'weight': w,'level':l}



    # print data
    def myOnPublishCallback():
        print("Published data",data, "to IBM Watson")



    success = deviceCli.publishEvent("event", "json", data, 0,
myOnPublishCallback)
    if not success:
        print("Not connected to IoTF")
    time.sleep(5)

    deviceCli.commandCallback = myCommandCallback

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