Python Code Development

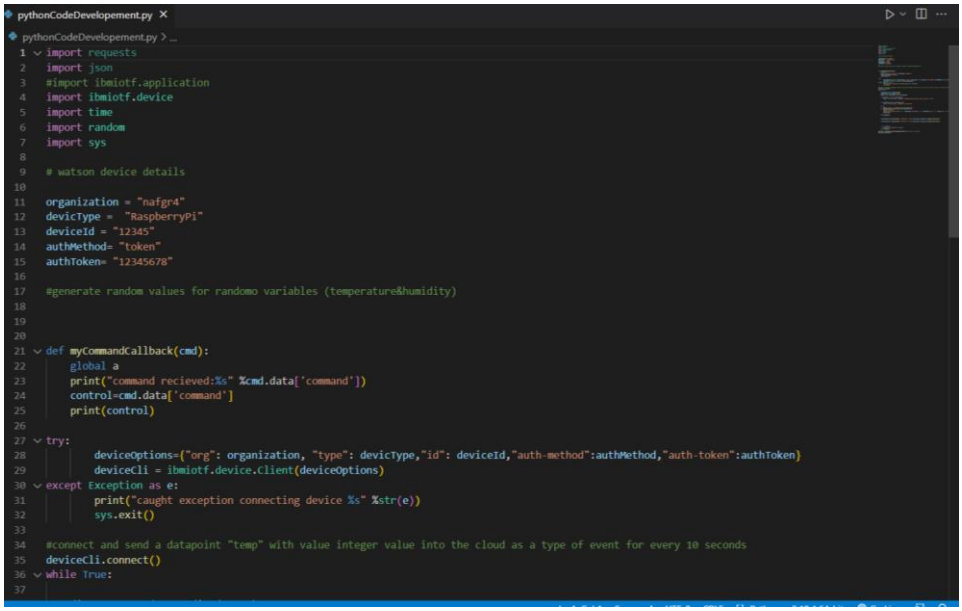
Team ID	PNT2022TMID02546
Project Name	Smart waste management system for metropolitan cities

Python Script

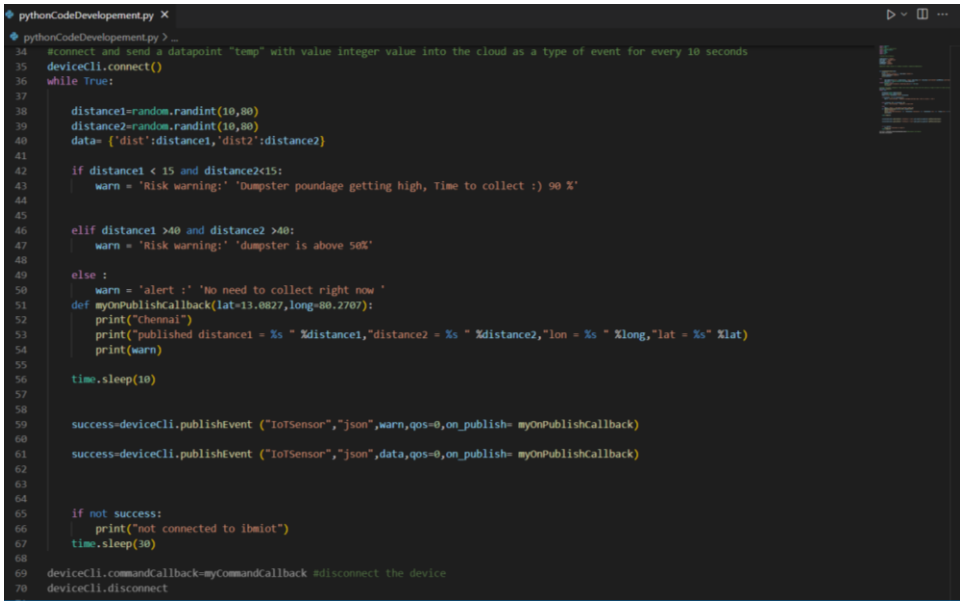
```
import requests
import json
#import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "w662fh"
devicType =  "NodeMCU"
deviceId = "12345"
authMethod= "token"
authToken= "12345678"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org": organization, "type": devicType,"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 "temp" with value integer value into the cloud
as a type of event for every 10 seconds
deviceCli.connect()
while True:

    distance1=random.randint(10,80)
```

```
    distance2=random.randint(10,80)
    data= {'dist':distance1,'dist2':distance2}

    if distance1 < 15 and distance2<15:
        warn = 'Risk warning:' 'Dumpster poundage getting high, Time to
collect :) 90 %'


    elif distance1 >40 and distance2 >40:
        warn = 'Risk warning:' 'dumpster is above 50%'

    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=13.0827,long=80.2707):
        print("Chennai")
        print("published distance1 = %s " %distance1,"distance2 = %s "
%distance2,"lon = %s " %long,"lat = %s" %lat)
        print(warn)

    time.sleep(10)


    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish=
myOnPublishCallback)

    success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish=
myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(30)

deviceCli.commandCallback=myCommandCallback #disconnect the device
deviceCli.disconnect
```

Code:

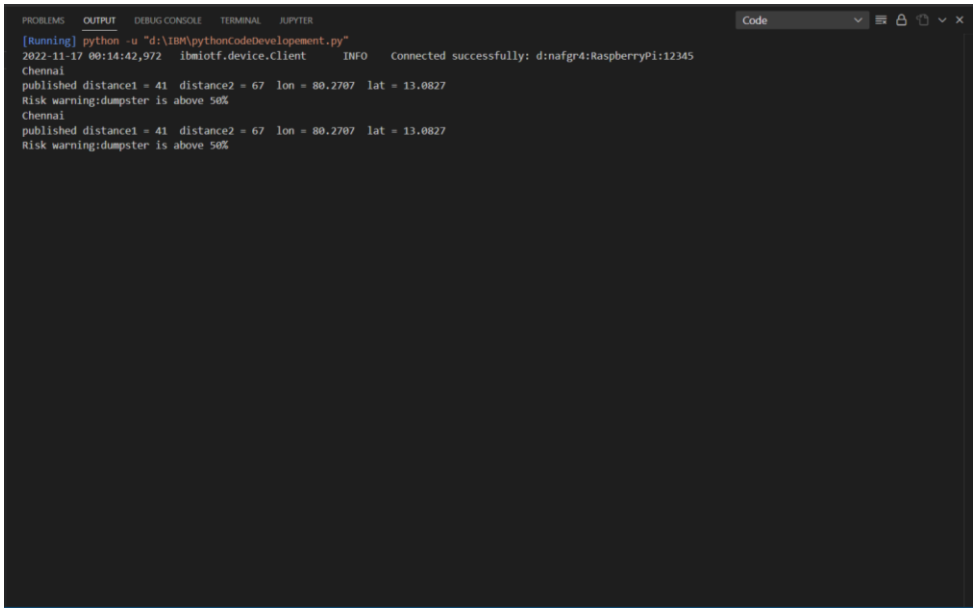


```python
import requests
import json
#import ibmiotf.application
import ibmiotf.device
import time
import random
import sys

# watson device details

organization = "nafgr4"
devicType =  "RaspberryPi"
deviceId = "12345"
authMethod= "token"
authToken= "12345678"

#generate random values for randomo variables (temperature&humidity)



def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)

try:
        deviceOptions={"org": organization, "type": devicType,"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 "temp" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()
while True:
```

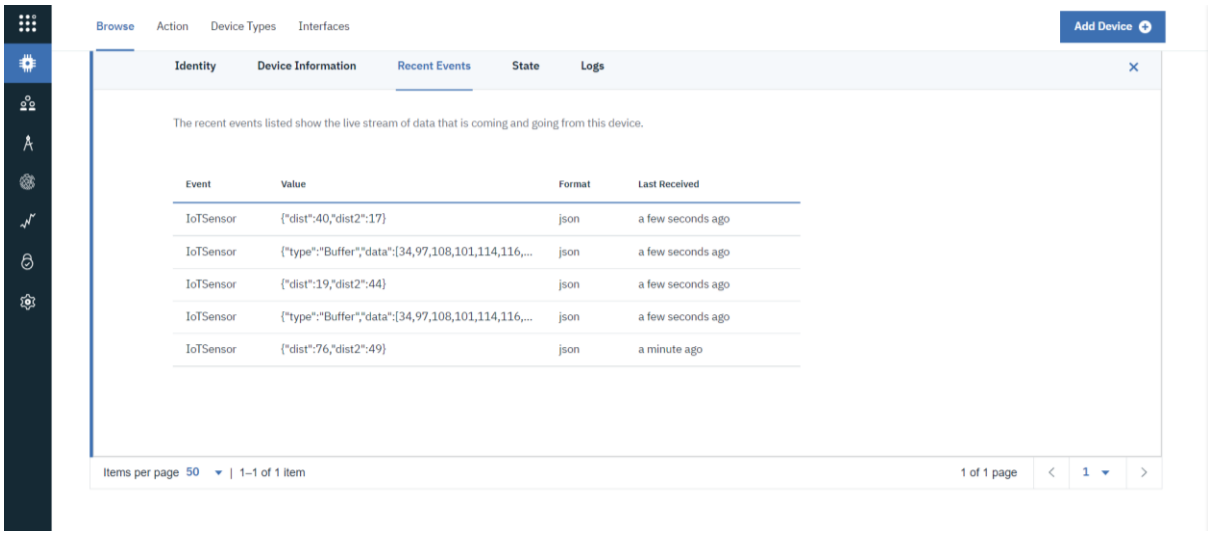


```python
#connect and send a datapoint "temp" with value integer value into the cloud as a type of event for every 10 seconds
deviceCli.connect()
while True:

    distance1=random.randint(10,80)
    distance2=random.randint(10,80)
    data= {'dist':distance1,'dist2':distance2}

    if distance1 < 15 and distance2<15:
        warn = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'


    elif distance1 >40 and distance2 >40:
        warn = 'Risk warning:' 'dumpster is above 50%'

    else :
        warn = 'alert :' 'No need to collect right now '
    def myOnPublishCallback(lat=13.0827,long=80.2707):
        print("Chennai")
        print("published distance1 = %s " %distance1,"distance2 = %s " %distance2,"lon = %s " %long,"lat = %s" %lat)
        print(warn)

    time.sleep(10)


    success=deviceCli.publishEvent ("IoTSensor","json",warn,qos=0,on_publish= myOnPublishCallback)

    success=deviceCli.publishEvent ("IoTSensor","json",data,qos=0,on_publish= myOnPublishCallback)



    if not success:
        print("not connected to ibmiot")
    time.sleep(30)

deviceCli.commandCallback=myCommandCallback #disconnect the device
deviceCli.disconnect
```

Connection To IBM Watson



IBM Watson



Node Red Output