PROJECT DEVELOPMENT PHASE
SPRINT-2

TEAM ID	PNT2022TMID37762
PROJECT NAME	Smart waste management system for metropolitan cities
DATE	5 November

PYTHON CODE:

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


organization = "bbzx4u"
devicType = "esp32_rasp"
deviceId = "123456789"
authMethod= "token"
authToken= "pnZ4GlTK5m&)t@P(gV"

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,"authtoken":authToken}
        deviceClient = 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 20 seconds
deviceClient.connect()


while True:

     distance= random.randint(10,70)
```

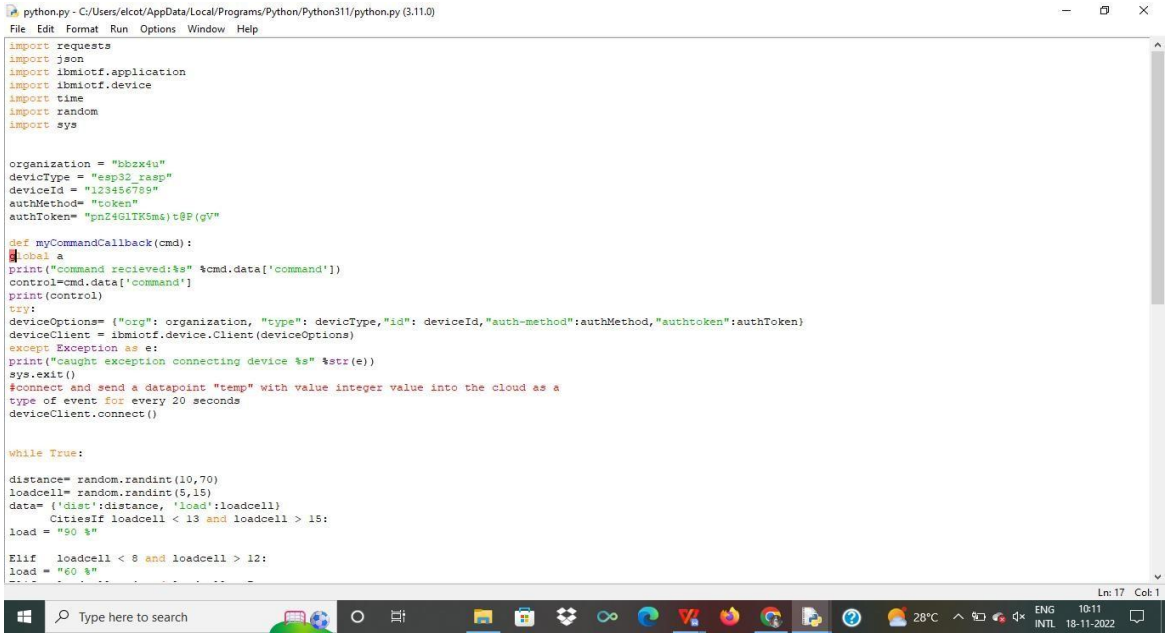
```
      loadcell= random.randint(5,15)
      data= {'dist':distance, 'load':loadcell}
     CitiesIf loadcell < 13 and loadcell > 15:
      load = "90 %"

Elif loadcell < 8 and loadcell > 12:
      load = "60 %"
Elif loadcell < 4 and loadcell > 7:
      load = "40 %"

else:
      load = "0 %"

if distance < 15:
      dist = 'Risk warning:' 'Dumpster poundage getting high, Time to collect :) 90 %'

elif distance < 40 and distance >16:
      dist = 'Risk warning:' 'dumpster is above 60%'

elif distance < 60 and distance > 41:
      dist = 'Risk warning:' '40 %'

else:
      dist = 'Risk warning:' '17 %'

If load == "90 %" or distance == "90 %":
      warn = 'alert :' ' Dumpster poundage getting high, Time to collect :)'
Elif load == "60 %" or distance == "60 %":
      warn = 'alert :' 'dumpster is above 60%'
else :
      warn = 'alert :' 'No need to collect right now '

def myOnPublishCallback(lat=12.678991,long=87.177731):
      print("Chennai, Porur")
      print("published distance = %s " %distance, "loadcell:%s " %loadcell, "lon = %s
" %long,"lat = %s" %lat)
      print(load)
      print(distance)
      print(warn)
time.sleep(20)

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

if not success:
    print("not connected to ibmiot")
time.sleep(40)
```

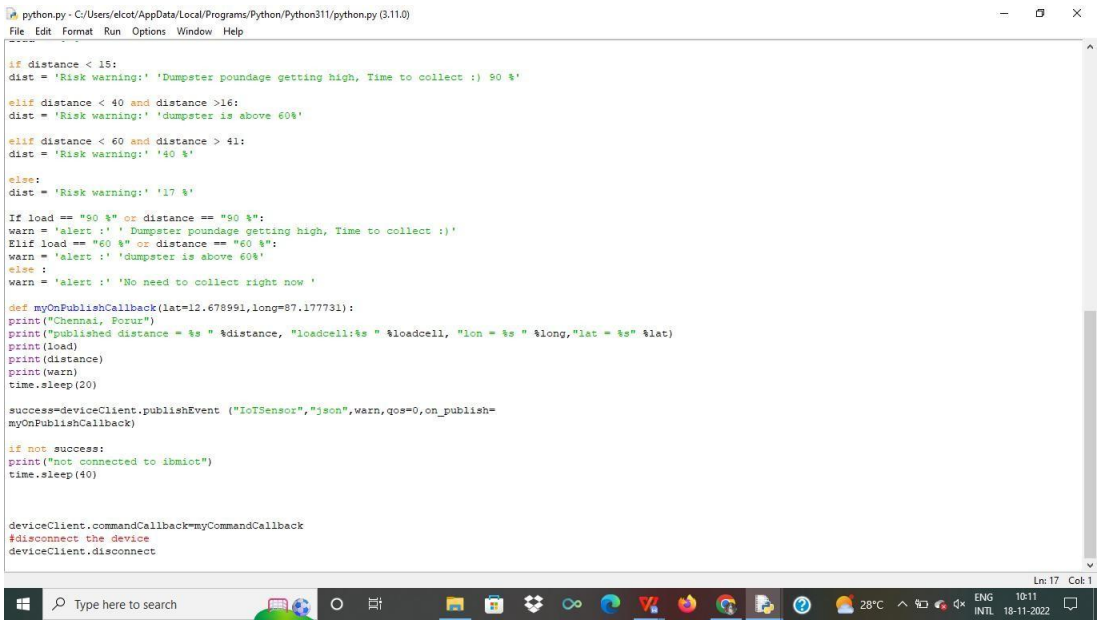
deviceClient.commandCallback=myCommandCallback

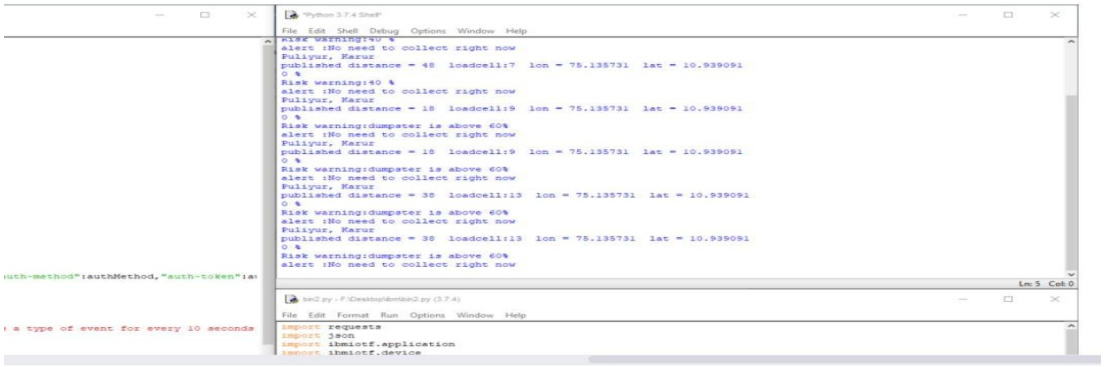
#disconnect the device

deviceClient.disconnect

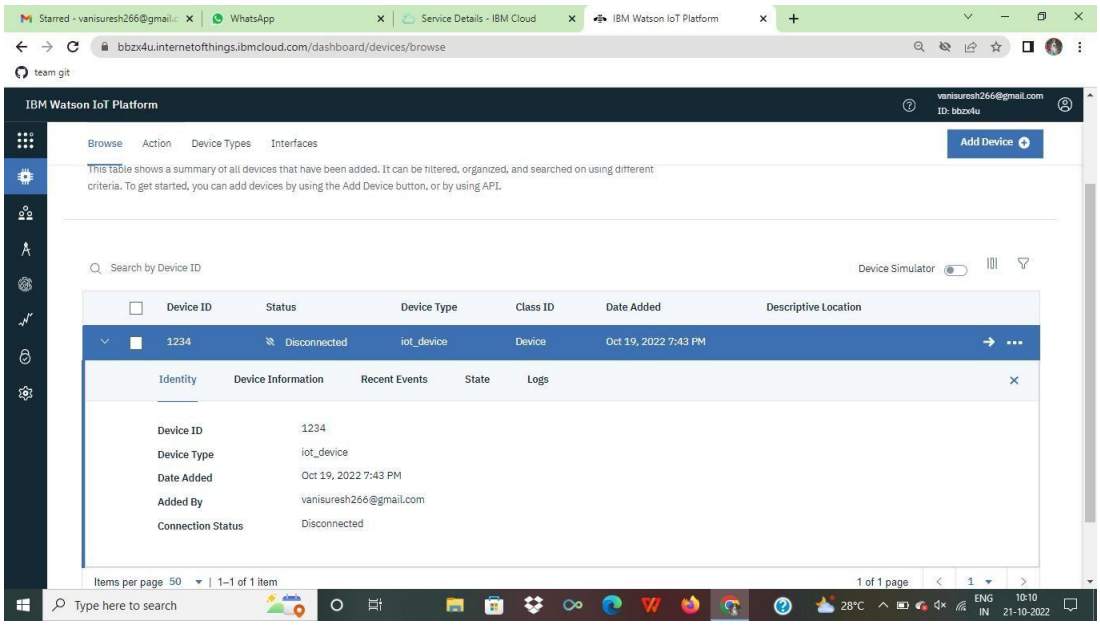
OUTPUT IN PYTHON IDLE:







IBM IOT PLATFORM:



NODE-RED PLATFORM :