Date	5 November 2022
Team ID	PNT2022TMID42565
Project Name	Smart waste management system for metropolitan cities
Story Points	15

Sprint 2

Develop the python code to find the GPS location using Latitude and Longitude (random values) and send it to Node red using IBM Watson platform and view location of bins on map

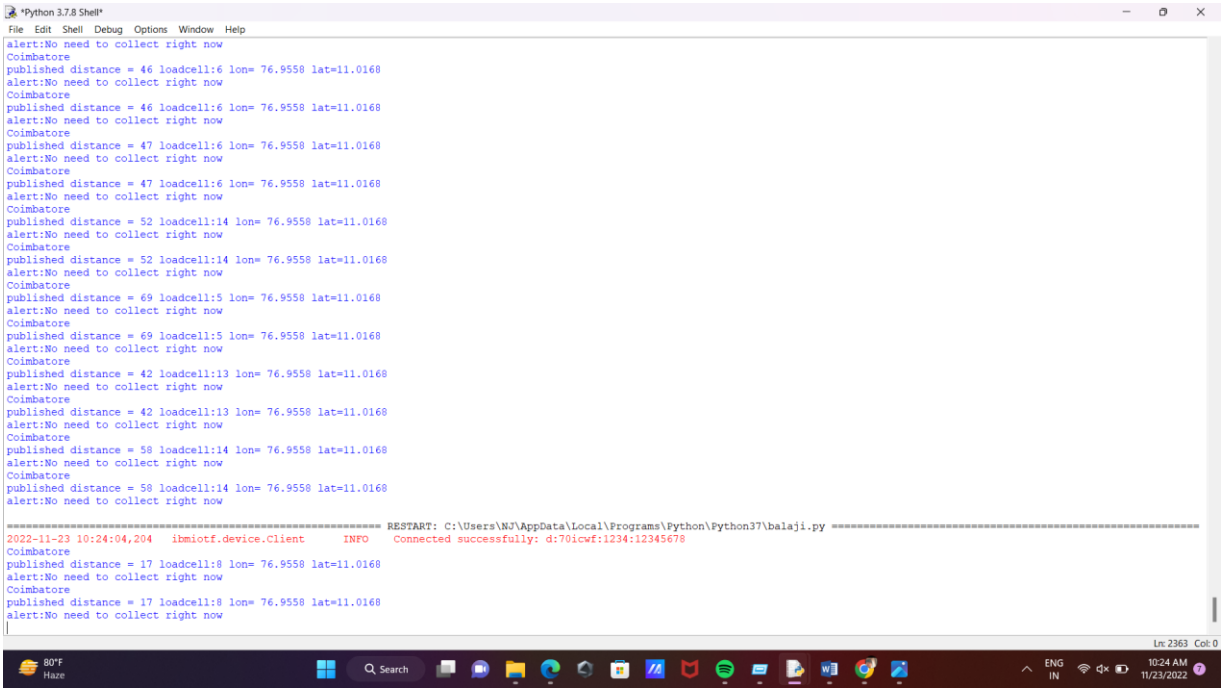
PYTHON CODE :

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


organization = "70icwf"
deviceType="1234"
deviceId="12345678"
authMethod="token"
authToken="S_OVsw4ICr5-Vk9A9x"
```

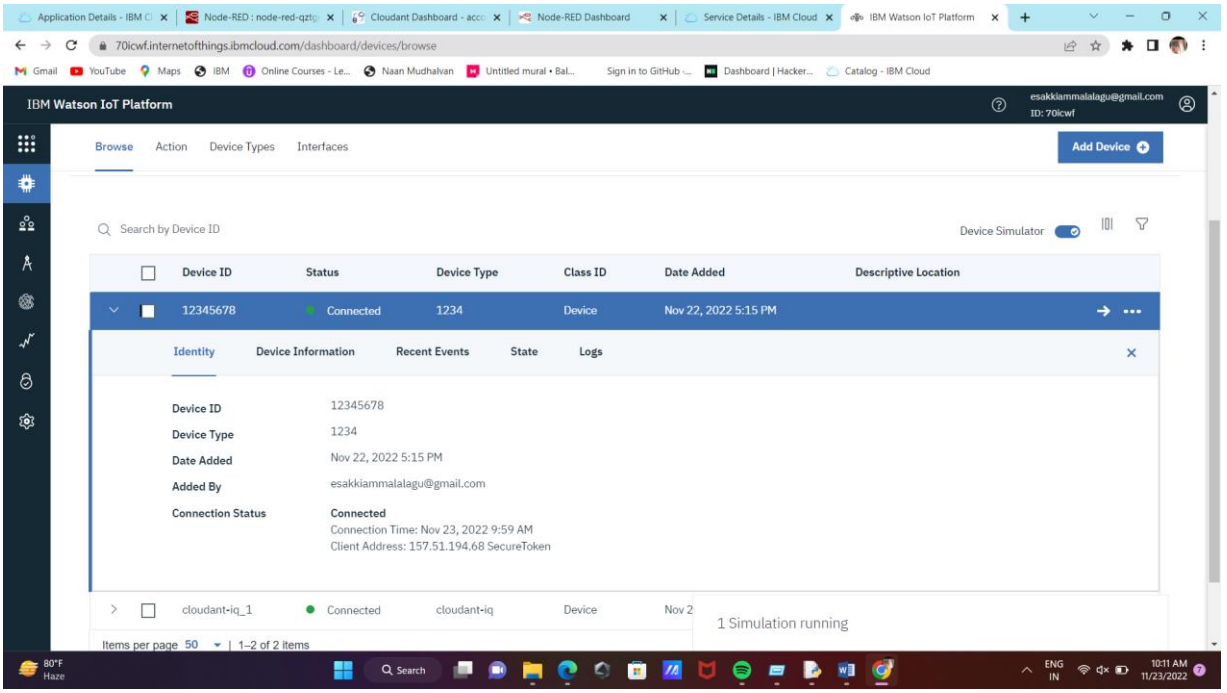
```
def myCommandCallback(cmd):
    global a
    print("Command received: %s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)
try:
    deviceOptions ={"org":organization, "type": deviceType, "id" : deviceId, "auth-method":
authMethod, "auth-token": authToken}
    deviceCli = ibmiotf.device.Client(deviceOptions)
except Exception as e:
    print("Caught exception connecting device: %s" %str(e))
    sys.exit()
deviceCli.connect()
while True:

    distance= random.randint(10,70)
    loadcell= random.randint(5,15)
    data= {'dist':distance,'load':loadcell}

    if 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 = 'Warning:' 'Trash is getting high, Time to collect 90 %'

    elif distance < 40 and distance >16:
       dist = 'Warning:' 'Trash is above 70 %'

    elif distance < 60 and distance > 41:
       dist = 'Warning:' '40 %'
    else:
       dist = 'Warning:' '17 %'
    if load == "90 %" or distance == "90 %":
```

```
        warn = 'alert:' ' Warning: Trash poundage getting high, Time to collect'
    elif load == "60%" or distance == "60 %":
        warn = 'alert:' 'Trash is above 60%'
    else :
        warn = 'alert:''No need to collect right now'
    def myOnPublishCallback(lat=11.0168,long=76.9558):
        print("Coimbatore")
        print("published distance = %s" %distance, "loadcell:%s" %loadcell, "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 connnected to ibmiot")
    time.sleep(20)
    deviceCli.commandCallback=myCommandCallback
deviceCli.disconnect()
```

Output in python IDLE :



IBM Watson IOT platform :



Node Red Platform :