Sprint 2

Date	5 November 2022
Team ID	PNT2022TMID20530
Project Name	Smart waste management system for metropolitan cities
Story Points	15

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
# watson device details
organization = "evd8ss"
devicType = "abcd"
deviceId = "1234"
authMethod= "token"
authToken= "12345678"
#generate random values for random variables (temperature&humidity)
def myCommandCallback(cmd):
  global a
  print("command recieved:%s"%cmd.data['command'])
  control=cmd.data['command']
  print(control)
```

```
deviceOptions={"org": organization, "type": devicType, "id": deviceId, "auth-
method":authMethod,"authtoken":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:
  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 %"
```

try:

```
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=10.678991,long=78.177731):
```

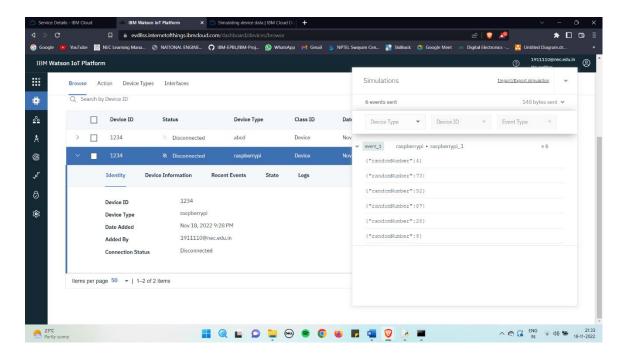
```
print("Gandigramam, Karur")
 print("published distance = %s " %distance,"loadcell:%s " %loadcell,"lon = %s "
%long,"lat = %s" %lat)
 print(load)
 print(dist)
 print(warn)
 time.sleep(10)
 success=deviceCli.publishEvent ("IoTSensor", "json", warn, qos=0, on publish=
myOnPublishCallback)
 success=deviceCli.publishEvent ("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

Output in python IDLE:



IBM Watson IOT platform:



Node Red Platform:

