PROJECT DEVELOPMENT PHASE

PROJECT DEVELOPMENT DELIVERY OF SPRINT 2

Date	**11-11-2022**
Team ID	**PNT2022TMID34548**
Project Name	**Smart Waste Management System For Metropolitan Cities**

SPRINT DESCRIPTION:

In this Sprint, we discuss about the complete Python Code Simulation.

CODE EXPLANATION:

The below described code is what we have developed for connecting with IBM IOT cloud. The code also connects with the Node- Red Service and displays the output frequently. Once the Code is simulated, the code runs with the output of bin value with latitude and longitude and with location ID, whenever the bin value beyond the maximum value. This code links with the IBM IOT Platform and then to Node-Red, Finally the result is displayed in our Application.

PYTHON CODE:

```
#IBM Watson IOT Platform

#pip install wiotp-sdk

import wiotp.sdk.device

import time

import random

from geopy.geocoders import Nominatim




myConfig = {

   "identity": {

      "orgId": "fa4qjp",
```

```
        "typeId": "123",
        "deviceId":"1234567"
    },
    "auth": {
        "token": "12345678"
    }
}


# Initialize Nominatim API
geolocator = Nominatim(user_agent="geography")



def myCommandCallback(cmd):
    print("Message received from IBM IoT Platform: %s" % cmd.data['alert'])
    m=cmd.data['alert']
    if m =="binfull":
      print ("Empty the bin immediately")


client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
client.connect()



while True:
```

```
binlevel=random.randint(10,100)
locationId=random.randint(1,5)
if locationId == 1:
    latitude=8.2396
    longitude=77.3066
elif locationId == 2:
    latitude=8.2114
    longitude=77.3031
elif locationId == 3:
    latitude=8.3348
    longitude=77.2664
elif locationId == 4:
    latitude=8.2507
    longitude= 77.3267
elif locationId == 5:
    latitude=8.3022
    longitude=77.2231
else:
    print("No Location Found!!")

# Get location with geocode
coordinates = str(latitude)+","+str(longitude)
```

```
    # print("coordinates",coordinates)

    location = geolocator.reverse(coordinates)

    address = location.raw['address']


    # Traverse the data

    village = address.get('village', '')

    district = address.get('state_district', '')

    state = address.get('state', '')

    country = address.get('country', '')


    if binlevel >= 90:

        myData={'latitude':latitude,
'longitude':longitude,'binlevel':binlevel,'village':village,'district':district,'state':state,'country'
:country}

        client.publishEvent(eventId="status", msgFormat="json", data=myData, qos=0,
onPublish=None)

        print("!!!!!!!!!BIN IS FULL!!!!!!!!!!  ",myData)

        client.commandCallback = myCommandCallback

        time.sleep(2)

    else :

        print("BIN IS IN NORMAL LEVEL")

        time.sleep(2)


client.disconnect()
```