Project Development phase

|  |  |
| --- | --- |
| Date | **09 November 2022** |
| Team ID | **PNT2022TMID41935** |
| Project Name | **Project – SMART WASTE MANAGEMENT FOR**  **METROPOLITAN CITIES** |
| Maximum Marks | **4 Marks** |

# Delivering of Sprint-3

**IBM Cloud service**

* Python code for sending location(latitude,longitude)along with the bin status
* Sending the data to the IBM watson

# Python code test code:

import time import sys

import ibmiotf.application import ibmiotf.device import random

from geopy.geocoders import Nominatim

# initialize Nominatim API

geolocator = Nominatim(user\_agent="geoapiExercises") #Provide your IBM Watson Device Credentials organization = "pb6xw8"

deviceType = "efgh" deviceId = "1234" authMethod = "token"

authToken = "12345678" 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()

# Connect and send a datapoint "hello" with value "world" into the cloud as an event of type "greeting" 10 times

print("checking connection to waston iot ")

deviceCli.connect() time.sleep(1)

while True:

name='bin' level=random.randint(1,100) weight=random.randint(1,100)

latitude=random.uniform(12.867342,13.043514) longitude=random.uniform(77.477635,77.695109) Latitude=str(latitude)

Longitude=str(longitude)

location = geolocator.reverse(Latitude+","+Longitude)

address = location.raw['address'] city = address.get('city', '') print('City : ', str(city))

#STATUS OF GARBAGE CAN

if(level<30):

level\_status="low level" print("level\_status=low level garbage")

elif(level>30)and(level<80): level\_status="medium level garbage" print("level\_status=low level garbage")

else:

level\_status="high level garbage" print("level\_status=high level garbage")

if (weight<30):

weight\_status="low level" print("weight\_status=low level garbage")

elif(weight>30)and(weight<80): weight\_status="medium level garbage" print("weight\_status=low level garbage")

else:

weight\_status="high level garbage" print("weight\_status=high level garbage")

data = { 'name' : name,'level' : level,'level\_status':level\_status,'weight': weight,'weight\_status':weight\_status, 'lat': Latitude,'lon':Longitude,'city':str(city)} #print data

def myOnPublishCallback():

print ("Published weight = %s kg" % weight,"level=%s m" %level, "latitude =

%s %%" % latitude, "longitude = %s %%" % longitude,"city=%s" %city,"to IBM Watson")

success = deviceCli.publishEvent("project", "json", data, qos=0, on\_publish=myOnPublishCallback)

if not success:

print("Not connection lost from sensor to ibm iot") time.sleep(10)

# Disconnect the device and application from the cloud deviceCli.disconnect()

