**Develop a Python code**

**Team Id : PNT2022TMID19409**

**Smart Waste Management for Metropolitan Cities**

**Python code:**

import time

import sys

import ibmiotf.application

import ibmiotf.device

import random

#Provide your IBM Watson Device Credentials

organization = "uzesig"

deviceType = "Arduino"

deviceId = "12345"

authMethod = "token"

authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd):

print("Command received: %s" % cmd.data['command'])

status=cmd.data['command']

if status=="lighton":

print ("led is on")

else :

print ("led is off")

#print(cmd)

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

deviceCli.connect()

while True:

#Get Sensor Data from esp32

weightSensor=random.randint(0,100)

irSensor=random.randint(0,100)

ultrasSensor=random.randint(0,100)

data = { 'weight' : weightSensor, 'ir':irSensor, 'ultrasonic':ultrasSensor }

#print data

def myOnPublishCallback():

print ("Published Weight of Trashcan is = %s C" % weightSensor, "IR Sensor = %s %%" % irSensor, "Ultrasonic Sensor = %s %%" % ultrasSensor, "to IBM Watson")

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

if not success:

print("Not connected to IoTF")

time.sleep(1)

deviceCli.commandCallback = myCommandCallback

# Disconnect the device and application from the cloud

deviceCli.disconnect()