Sprint-1

Team ID	PNT2022TMID38943
Project Name	Smart Waste Management for Metropolitan Cities

Python idle:

Project coding:

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


#Provide your IBM Watson Device Credentials
organization = "yal2ec"
deviceType = "BIN1"
deviceId = "54321"
authMethod = "token"
authToken = "12345678"


# Initialize GPIO
def myCommandCallback(cmd):
    print("Command received: %s" % cmd.data['command'])
    status=cmd.data['command']


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 ultrasonic

    distance= random.randint(5,100)
    data= {'distance':distance}
    if distance >5 and distance<=35:
      print("alert:' 'waste bin level high is 90%, Time to collect")
    elif distance>35 and distance<=50:
      print("Risk warning:' 'waste Bin is above 60%")
    elif distance >35 and distance <=70:
      print("waste Bin level is above 40%")
    elif distance >70 and distance <=85:
        print("waste Bin level is above 25%")
    elif distance >85 and distance <100:
        print("waste Bin level is above 10%")
    elif distance==100:
        print("waste Bin is Empty")
```

```
#print data
def myOnPublishCallback():
    print ("Published distance = %s " %distance ,"to IBM Watson")

success = deviceCli.publishEvent("IoTSensor", "json", data, qos=0,
on_publish=myOnPublishCallback)
if not success:
    print("Not connected to IoTF")
time.sleep(10)

deviceCli.commandCallback = myCommandCallback

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

coding screen shot: