|  |  |
| --- | --- |
| **Project Name** | Smart Waste Management System For Metropolitan Cities |
| **Team ID** | PNT2022TMID20896 |

**Sprint-2**

# Python Code

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

#Provide your IBM Watson Device Credentials organization

= "2melo1" deviceType =

"waste" deviceId = "1234" authMethod = "token" authToken = "12345678"

# Initialize GPIO

def myCommandCallback(cmd): print("Command received: %s" % cmd.data['command']) status=cmd.data['command'] if status=="waste level":

print ("waste level monitored") else :

print ("weight level monitored"

#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 DHT11

level=random.randint(0,100) weight=random.randint(0,100)

data = { 'level' : level, 'weight': weight } #print data

def myOnPublishCallback():

print ("Published Level = %s %%" % level, "Weight = %s %%" % weight, "to IBM Watson"

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

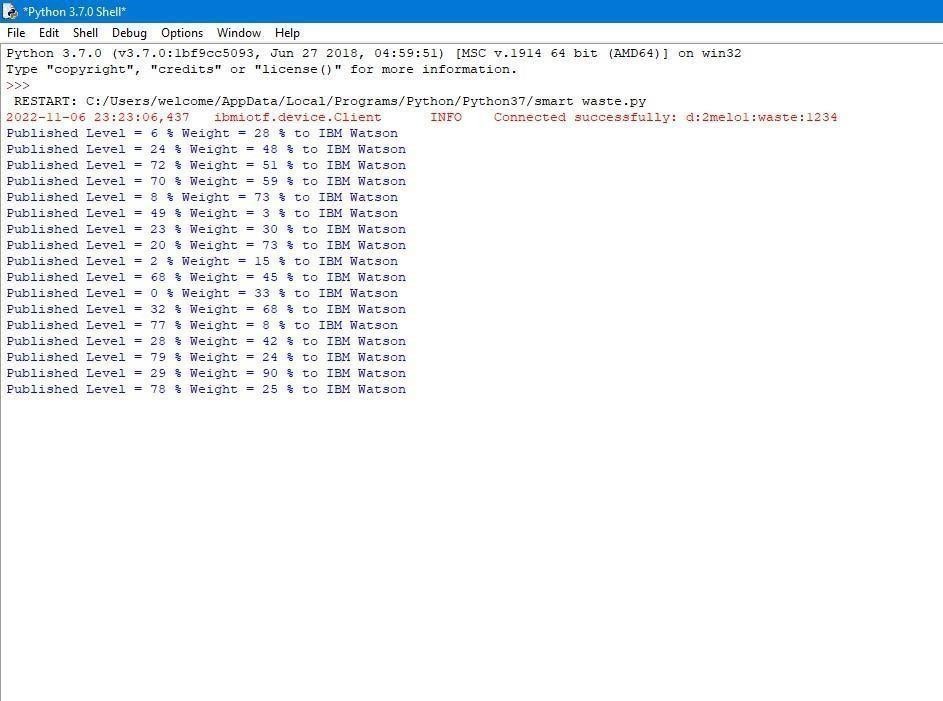
if not success: print("Not connected to IoTF")

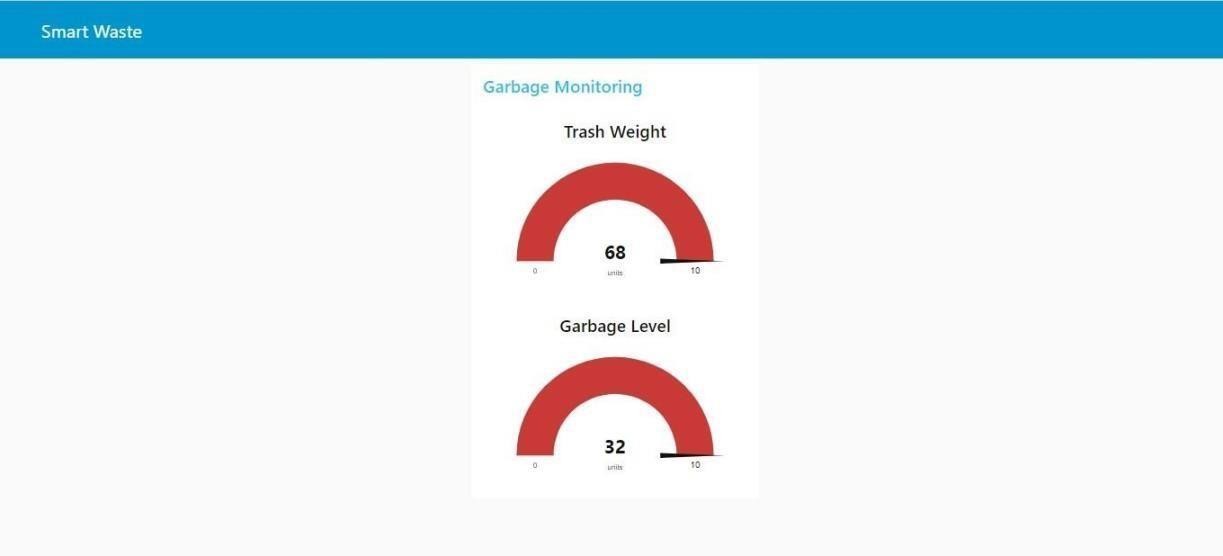
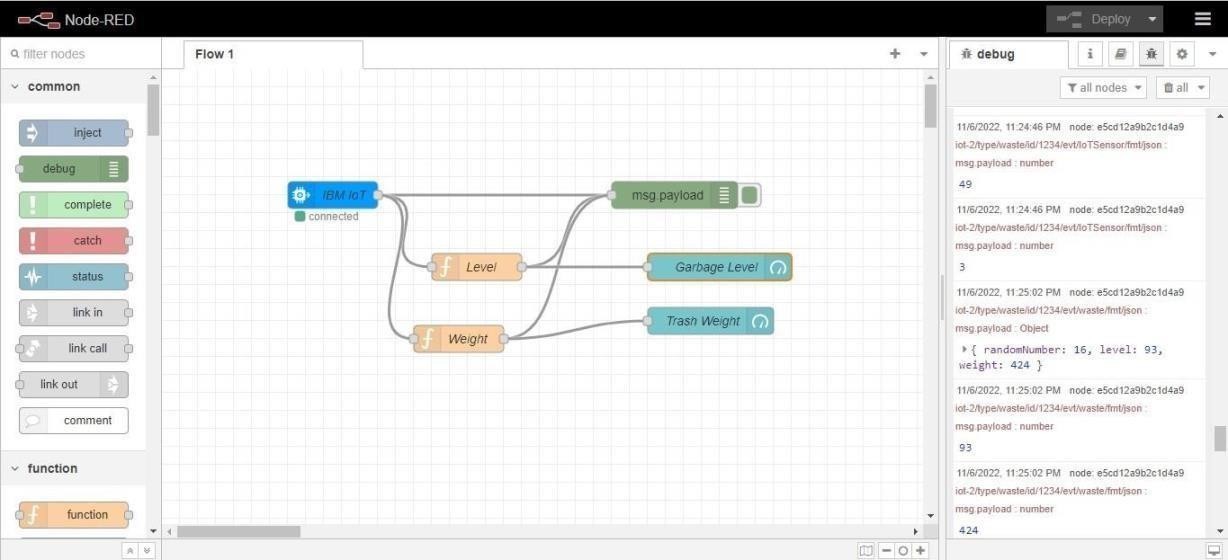
time.sleep(20)

deviceCli.commandCallback = myCommandCallback

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

# OUTPUT:



**NODE RED INPUT AND OUPUT:**