

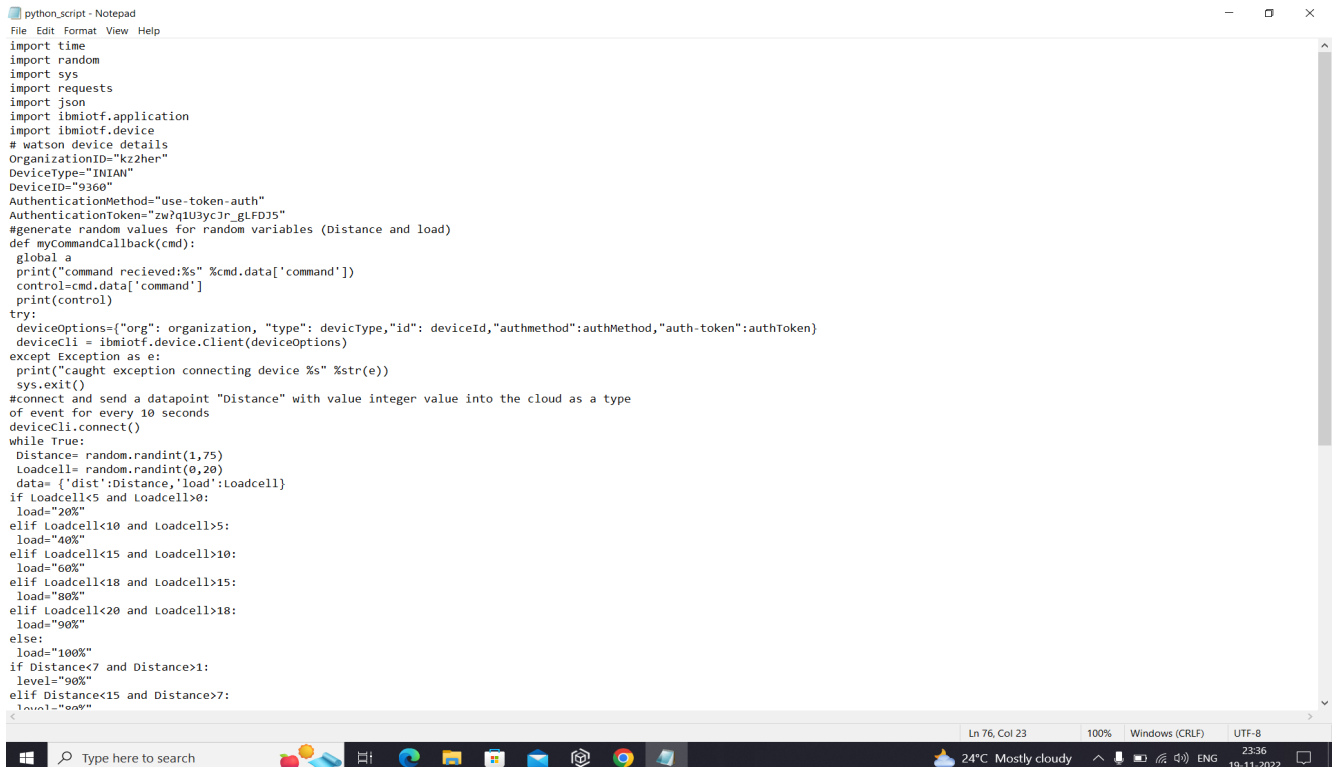
# PROJECT DEVELOPMENT PHASE

## SPRINT 3

TEAM ID	PNT2022TMID04607
PROJECT NAME	SMART WASTE MANAGEMENT SYSTEM FOR METROPOLITON CITIES

### STEP 1:

The development of interface in MIT app inverter for various input parameters.



```
python_script - Notepad
File Edit Format View Help
import time
import random
import sys
import requests
import json
import ibmiotf.application
import ibmiotf.device
# watson device details
OrganizationID="kz2her"
DeviceType="INIAN"
DeviceID="9360"
AuthenticationMethod="use-token-auth"
AuthenticationToken="zw?giU3yc3r_gLFD35"
#generate random values for random variables (Distance and load)
def myCommandCallback(cmd):
    global a
    print("command recieved:%s" %cmd.data['command'])
    control=cmd.data['command']
    print(control)
try:
    deviceOptions={"org": organization, "type": deviceType,"id": deviceId,"authmethod":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 "Distance" with value integer value into the cloud as a type
of event for every 10 seconds
deviceCli.connect()
while True:
    Distance= random.randint(1,75)
    Loadcell= random.randint(0,20)
    data= {'dist':Distance,'load':Loadcell}
    if Loadcell<5 and Loadcell>0:
        load="20%"
    elif Loadcell<10 and Loadcell>5:
        load="40%"
    elif Loadcell<15 and Loadcell>10:
        load="60%"
    elif Loadcell<18 and Loadcell>15:
        load="80%"
    elif Loadcell<20 and Loadcell>18:
        load="90%"
    else:
        load="100%"
    if Distance<7 and Distance>1:
        level="90%"
    elif Distance<15 and Distance>7:
        level="90%"
    level="90%"
```

STEP 2: Run the python code it send data to IBM IoT Watson Platform.

The screenshot shows the IBM Watson IoT Platform dashboard. The main table lists devices with columns: Device ID, Status, Device Type, Class ID, Date Added, and Descriptive Location. A device with ID 9360 is selected, and its details are shown in a sidebar. A configuration window for 'Device Type: NODE' is open, showing a schedule of 'Every Minute' and a payload of random values for Temperature, Humidity, and PIR.

Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location
4222	Connected	NODE	Device	Nov 18, 2022 3:50 PM	
8778	Disconnected	Sampledevice	Device	Nov 14, 2022 3:38 PM	
9360	Disconnected	INIIN	Device	Nov 19, 2022 11:05 PM	
9843	Disconnected	PYTHONCODE	Device	Nov 14, 2022 3:49 PM	

Device Details for ID 9360:

- Device ID: 9360
- Device Type: INIIN
- Date Added: Nov 19, 2022 11:05 PM
- Added By: jayanthma.19ece@kongu.edu
- Connection Status: Disconnected

Configuration for Device Type: NODE:

- Event type name: event\_1
- Schedule: 20 Every Minute
- Payload:
 

```

0 {
1   "Temperature": random(0, 100)
2   "Humidity": random(0, 100)
3   "PIR": random(0, 100)
4 }
5
      
```

STEP 3: Open Node-RED flow dashboard.

The screenshot shows the Node-RED flow dashboard. The main flow consists of an 'IBM IoT' node connected to a 'msg.payload' node. The flow is split into three parallel paths: 'TEMPERATURE', 'HUMIDITY', and 'PIR'. Each path has a function node that processes the data. The flow then joins back into a single path that outputs to a 'web output' node. The debug console on the right shows the output of the flow, including the device ID and the processed data for Temperature, Humidity, and PIR.

```

graph LR
    IoT[IBM IoT] --> Split(( ))
    Split --> Temp[TEMPERATURE]
    Split --> Hum[HUMIDITY]
    Split --> PIR[PIR]
    Temp --> Join(( ))
    Hum --> Join
    PIR --> Join
    Join --> WebOut[web output]
  
```

Debug Console Output:

```

msg.payload : number
493
11/19/2022, 11:12:30 PM node:
t22649a.0d0d98
iot-2?type=NODE/id/4222/ev/IoTSensor/rmt/json :
msg.payload : number
487
11/19/2022, 11:12:30 PM node:
t22649a.0d0d98
iot-2?type=NODE/id/4222/ev/IoTSensor/rmt/json :
msg.payload : number
315
11/19/2022, 11:12:30 PM node:
t22649a.0d0d98
iot-2?type=NODE/id/4222/ev/IoTSensor/rmt/json :
msg.payload : number
307
11/19/2022, 11:12:30 PM node:
t22649a.0d0d98
iot-2?type=NODE/id/4222/ev/IoTSensor/rmt/json :
msg.payload : number
70
11/19/2022, 11:12:30 PM node:
t22649a.0d0d98
iot-2?type=NODE/id/4222/ev/IoTSensor/rmt/json :
msg.payload : number
87
  
```

STEP 4: Open Node-RED user interface to show the Hazardous gases, Humidity

and Temperature value in respective charts.

