

### Use Dashboard Nodes for Creating UI (Web Apps)

Team ID	PNT2022TMID04391
Project Name	Smart waste management system for metropolitan cities
Date	10 November 2022

Step 1: Open Node red and pick and place blocks according to python script flow

Step2: Make sure necessary blocks are installed in Node Red

Step 3: After creating the flow click on deploy

Step 4: Output is displayed in Node-red Debug window

Step 5: Also, web UI can also be seen by the URL followed by/ui

# Screenshots: NODE – RED FLOW

The screenshot displays the Node-RED web interface with four parallel flows. Each flow consists of an inject node, a function node (labeled Distance 1, 2, 3, and 4), and a msg.payload node. The function nodes are connected to a [get] /sensor node and an http node. The debug console on the right shows the execution of these flows, displaying the msg.payload and the resulting http response.

**Flow 1:**

```
graph LR
    inject1[inject] --> f1[function 1]
    f1 --> msg_payload1[msg.payload]
    f1 --> load_cell1[LOAD CELL 1]
    msg_payload1 --> http1[http]
    load_cell1 --> http1
```

**Flow 2:**

```
graph LR
    inject2[inject] --> f2[function 2]
    f2 --> msg_payload2[msg.payload]
    f2 --> load_cell2[LOAD CELL 2]
    msg_payload2 --> http2[http]
    load_cell2 --> http2
```

**Flow 3:**

```
graph LR
    inject3[inject] --> f3[function 3]
    f3 --> msg_payload3[msg.payload]
    f3 --> load_cell3[LOAD CELL 3]
    msg_payload3 --> http3[http]
    load_cell3 --> http3
```

**Flow 4:**

```
graph LR
    inject4[inject] --> f4[function 4]
    f4 --> msg_payload4[msg.payload]
    f4 --> load_cell4[LOAD CELL 4]
    msg_payload4 --> http4[http]
    load_cell4 --> http4
```

**Debug Console:**

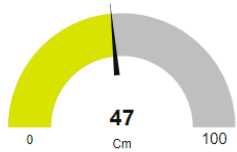
```
msg.payload : string[36]
"alert :No need to collect right now"
11/12/2022, 11:19:35 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : undefined
undefined
11/12/2022, 11:19:36 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : Object
* { dist: 48, load: 7 }
11/12/2022, 11:19:37 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : number
48
11/12/2022, 11:19:44 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : string[36]
"alert :No need to collect right now"
11/12/2022, 11:19:45 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : undefined
undefined
11/12/2022, 11:19:46 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : Object
* { dist: 18, load: 9 }
11/12/2022, 11:19:47 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : number
18
11/12/2022, 11:19:54 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : string[36]
"alert :No need to collect right now"
11/12/2022, 11:19:55 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : undefined
undefined
11/12/2022, 11:19:56 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : Object
* { dist: 38, load: 13 }
11/12/2022, 11:19:57 AM node: msg.payload
cat: 2type:BIN4Id:BIN4Id:evl:toTSensor:fmj:son :
msg.payload : number
38
```

## NODE – RED UI

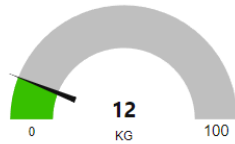
≡ control

control

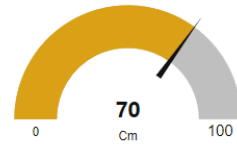
Distance 1



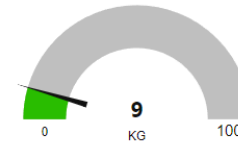
LOAD CELL 1



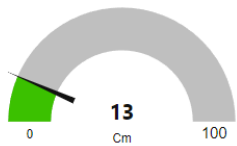
Distance 2



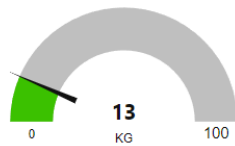
LOAD CELL 2



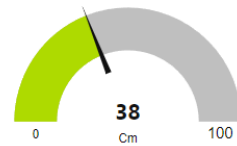
Distance 3



LOAD CELL 3



Distance 4



LOAD CELL 4

