

Use Dashboard Nodes for Creating UI (Web Apps)

Team ID	PNT2022TMID37138
Project Name	Smart waste management system for metropolitan cities

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 (Flow 1 to Flow 4) and a debug console on the right.

Flow 1: Starts with an `IBM IoT` node (connected), followed by a `msg.payload` node, then a `Distance 1` node, and finally a `LOAD cell 1` node.

Flow 2: Starts with a `[get] /sensor` node, followed by a `function 1` node, and finally an `http` node.

Flow 3: Starts with an `IBM IoT` node (connected), followed by a `msg.payload` node, then a `Distance 2` node, and finally a `LOAD cell 2` node.

Flow 4: Starts with a `[get] /sensor` node, followed by a `function 2` node, and finally an `http` node.

Flow 5: Starts with an `IBM IoT` node (connected), followed by a `msg.payload` node, then a `Distance 3` node, and finally a `LOAD cell 3` node.

Flow 6: Starts with a `[get] /sensor` node, followed by a `function 3` node, and finally an `http` node.

Flow 7: Starts with an `IBM IoT` node (connected), followed by a `msg.payload` node, then a `Distance 4` node, and finally a `LOAD cell 4` node.

Flow 8: Starts with a `[get] /sensor` node, followed by a `function 4` node, and finally an `http` node.

Debug Console: Shows a series of messages from the `msg.payload` node, including alerts and data points like `{ dist: 48, load: 7 }`, `48`, `{ dist: 18, load: 9 }`, `18`, `{ dist: 38, load: 13 }`, and `38`.

NODE – RED UI

≡ control

control

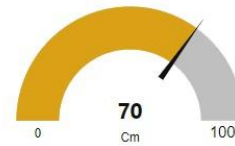
Distance 1



LOAD CELL 1



Distance 2



LOAD CELL 2



Distance 3



LOAD CELL 3



Distance 4



LOAD CELL 4

