IoT Domain Analyst

Lab Record — Lab 2

- 15 February 2021

Programme	:	B.Tech(CSE)	Semester	:	Winter 2020–21
Course Title	:	IoT Domain Analyst – Lab	Code	:	ECE3502
			Slot	:	L5+L6
Name	:	**Aadhitya Swarnesh**	Registration. No	:	[redacted]
Faculty (s)	:	[redacted]	Expt. No	:	**2**

Experiment 1 :

Create a Workflow for the generation of One-Time Passwords (OTP) using Node-Red and Concepts of IoT.

Aim :

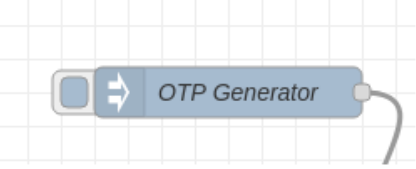
To design a workflow to generate One Time Passwords using Javascript function nodes and display it in the Dashboard with visualisations.

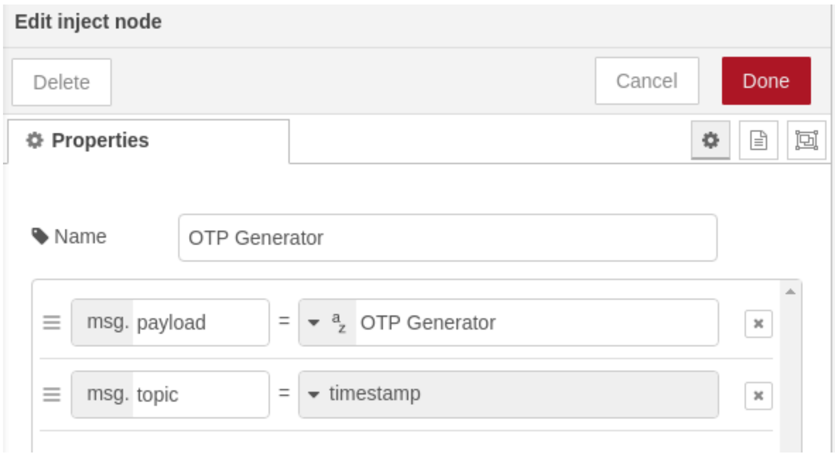
Description :

- ⋆ We will use the inject node to trigger the working function to generate the OTP. We will set a time based trigger event so that a new OTP is generated for every few minutes.
- ⋆ We will use a function node where the Javascript function will be written which will be responsible to generate the OTP number within the specified range.
- ⋆ We will use the debug node to display the OTP in the debug window. Every time a new OTP is generated, this node displays it in the debug window.
- ⋆ We use the Chart node to visualise the range of values of the OTP's generated.
- ⋆ We also have an audio out node to have a voice output of the OTP which was generated by the function.

Flow Diagrams :

Stage 1 : Inject Node



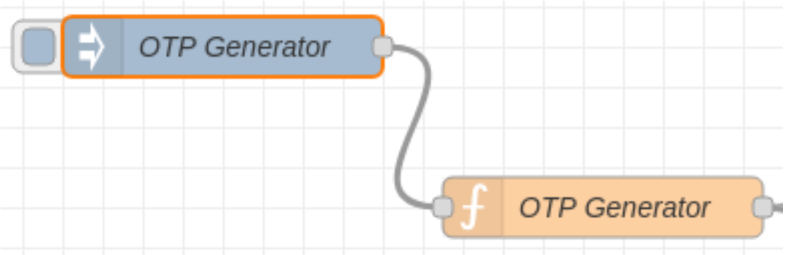

Edit inject node

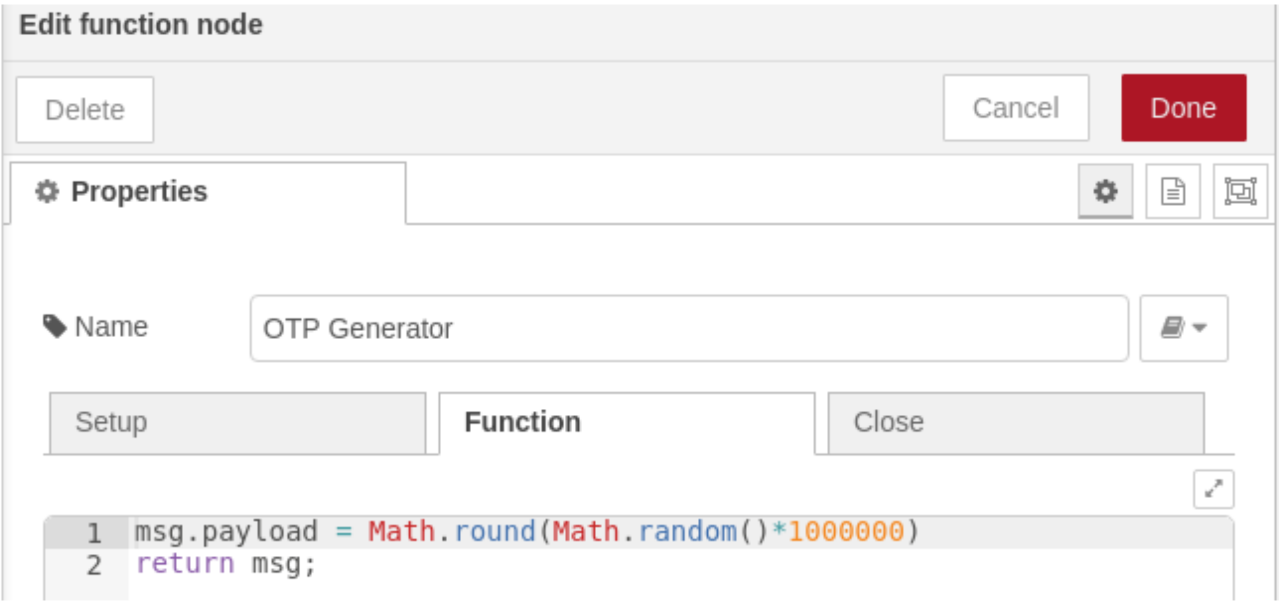
Name: OTP Generator

msg. payload = OTP Generator

msg. topic = timestamp

Stage 2 : Function Node

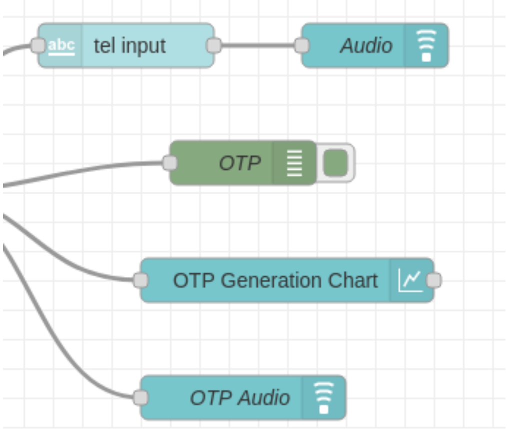



Edit function node

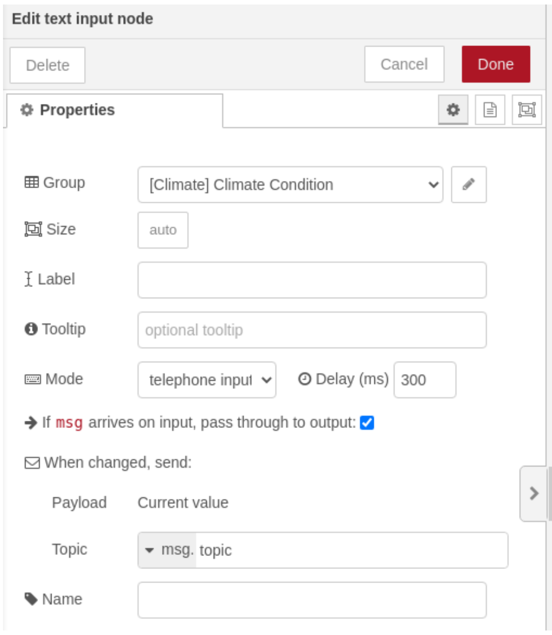
Name: OTP Generator

```
msg.payload = Math.round(Math.random()*1000000)
return msg;
```

Stage 3 : Output Nodes



1. Text Input Node : For displaying and entering the Telephone number and also to provide the audio output by transferring to the audio out node.


Edit text input node

Delete | Cancel | Done

Properties

Group: [Climate] Climate Condition

Size: auto

Label:

Tooltip: optional tooltip

Mode: telephone input | Delay (ms): 300

If msg arrives on input, pass through to output: ☑

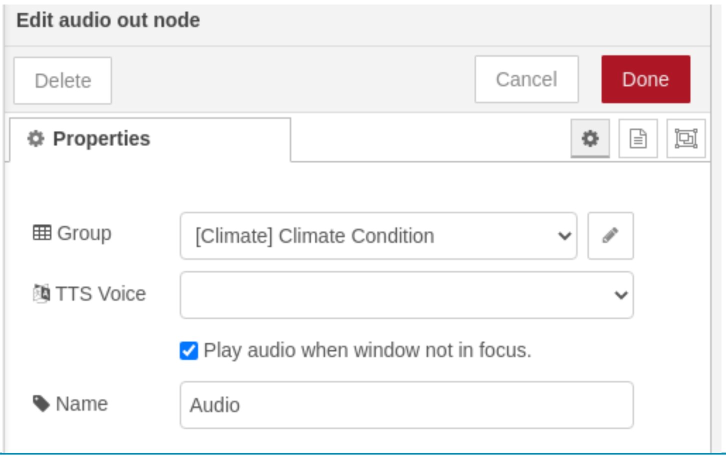
When changed, send:

Payload: Current value

Topic: msg. topic

Name:

2. Audio Out Node : This is a part of the UI Dashboard, and is used for providing the audio output. Both the audio out nodes have a similar configuration, one speaks the telephone number while the other the OTP generated.


Edit audio out node

Delete | Cancel | Done

Properties

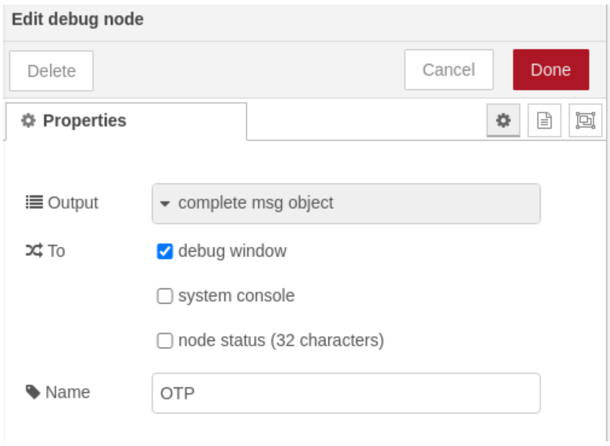
Group: [Climate] Climate Condition

TTS Voice:

☑ Play audio when window not in focus.

Name: Audio

3. Debug Node : This outputs the generated OTP in the debug window.



Edit debug node

Output: complete msg object

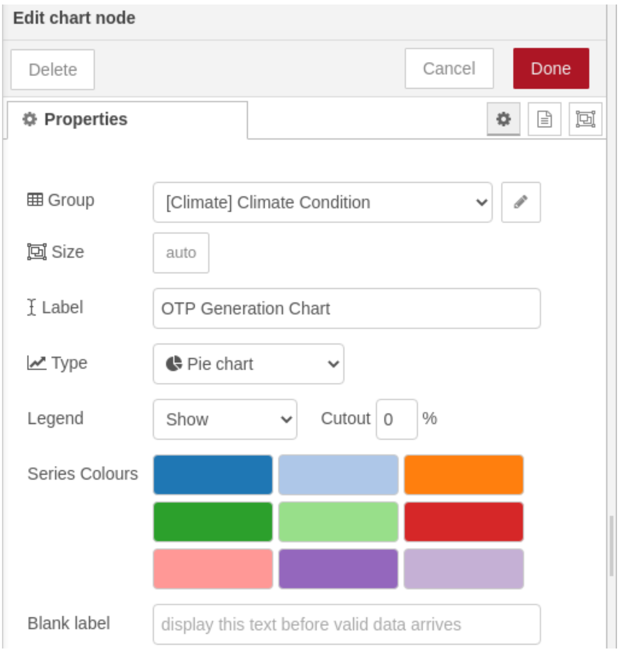
To: ☑ debug window

☐ system console

☐ node status (32 characters)

Name: OTP

4. Chart Node : Visualisation for the OTP's generated until now. Tells us about the range of the values generated by the JS function.



Edit chart node

Group: [Climate] Climate Condition

Size: auto

Label: OTP Generation Chart

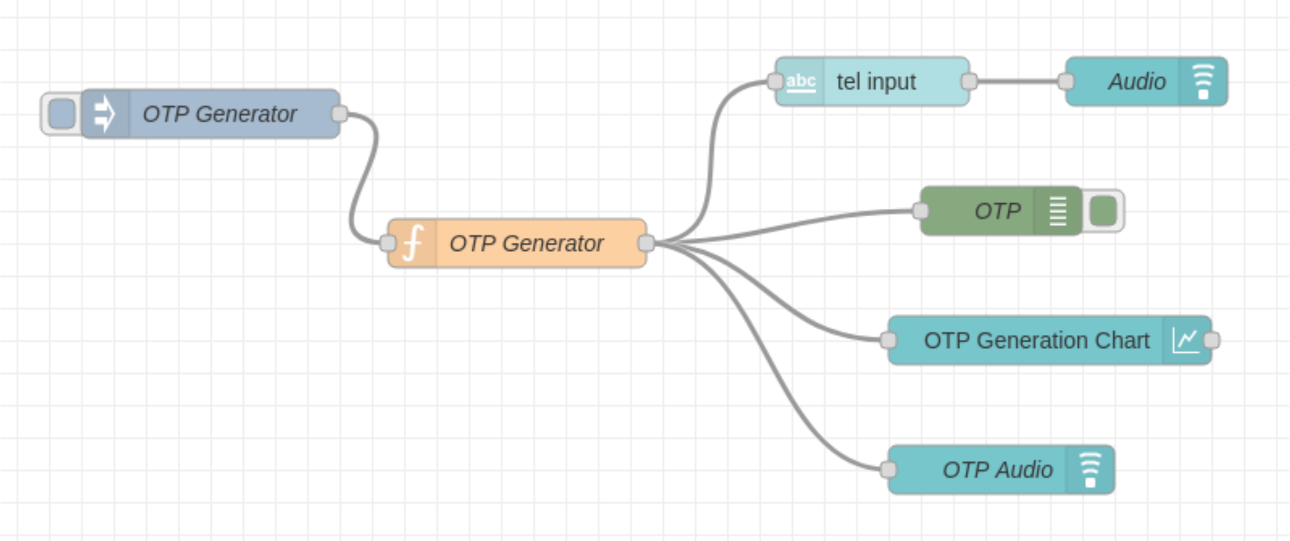
Type: Pie chart

Legend: Show   Cutout 0 %

Series Colours

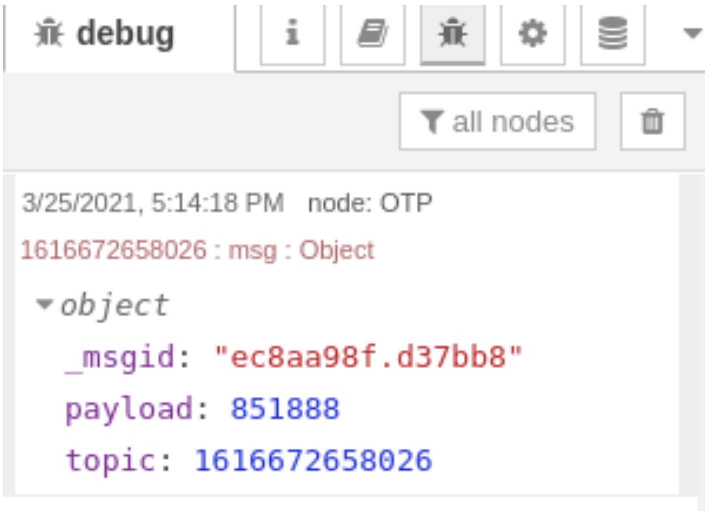
Blank label: display this text before valid data arrives

Stage 4 : Complete Flow



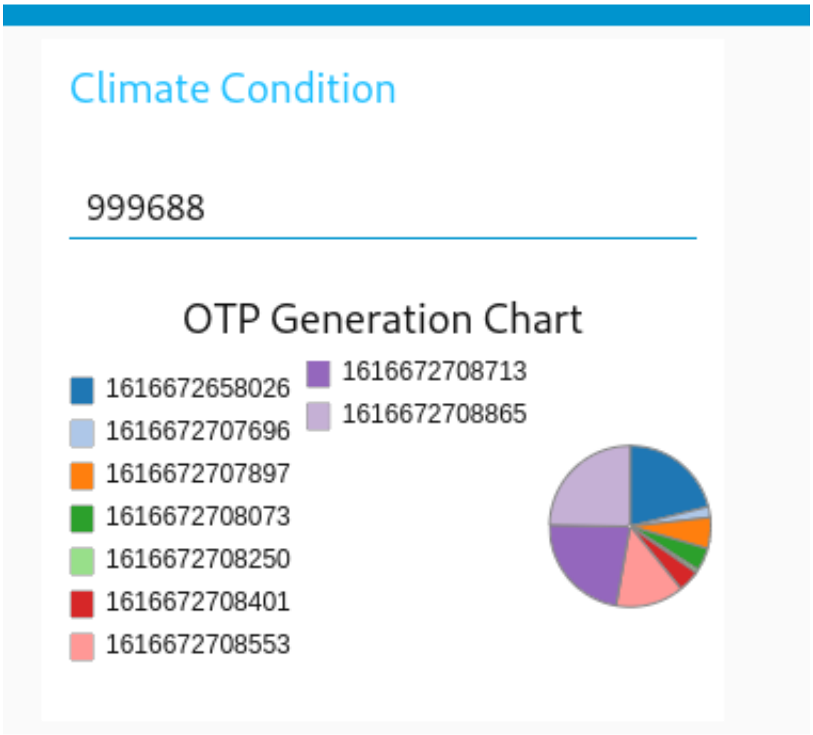
Result :

The following image denotes an OTP that was generated by the function and given as the output in the Debug Window :



```
3/25/2021, 5:14:18 PM   node: OTP
1616672658026 : msg : Object
▾object
  _msgid: "ec8aa98f.d37bb8"
  payload: 851888
  topic: 1616672658026
```

The following image is the output obtained in the Dashboard component due to the text box, audio out, and the chart nodes :



Climate Condition

999688

Experiment 2 :

Create a Workflow for retrieving the Environmental Conditions of a Location and visualise it using Node-Red and Concepts of IoT.

Aim :

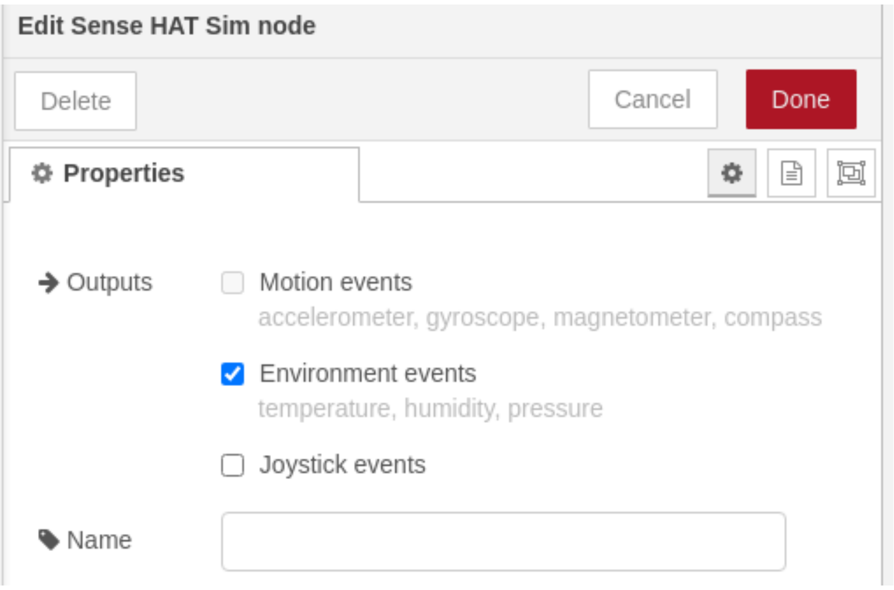
To design a workflow to retrieving the Environmental Conditions of a Location and using Weather API's and it using and display it in the Dashboard with visualisations.

Description :

- ⋆ We will use the Sense HAT Simulation node to get the weather data (environmental events) from their servers.
- ⋆ We will use a function node where the Javascript function will be written which will be responsible to get the pressure, temperature and the humidity values from the Sense HAT node.
- ⋆ We will use the Debug nodes to display the environmental values in the debug window.
- ⋆ We use the Gauge nodes to visualise the range of values of all the environmental values obtained.

Flow Diagrams :

Stage 1 : Sense HAT Simulation Node

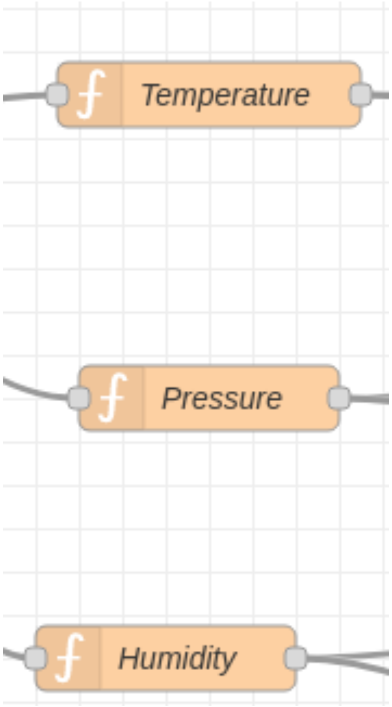

Edit Sense HAT Sim node

Outputs
☐ Motion events
accelerometer, gyroscope, magnetometer, compass
☑ Environment events
temperature, humidity, pressure
☐ Joystick events

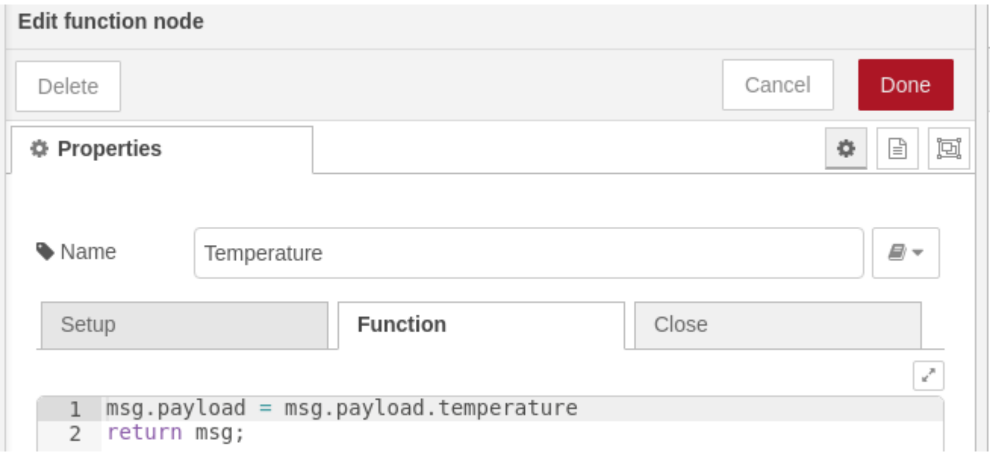
Name

Stage 2 : Function Nodes

The function nodes responsible for the JavaScript Function to retrieve the values from the API and then to display them in the debug window and also in the UI dashboard.

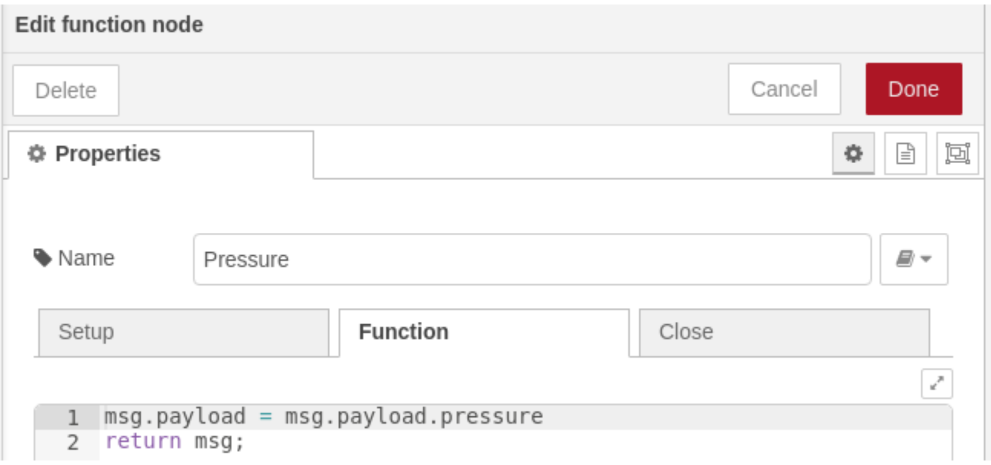


The following diagrams denote each of these function nodes including the Java script code written in each of them :



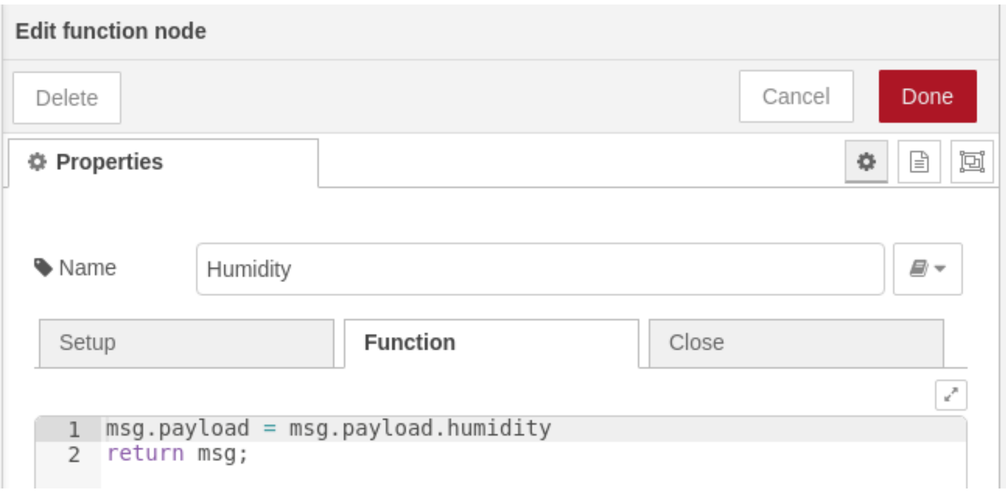
Name: Temperature

```
msg.payload = msg.payload.temperature
return msg;
```



Name: Pressure

```
msg.payload = msg.payload.pressure
return msg;
```

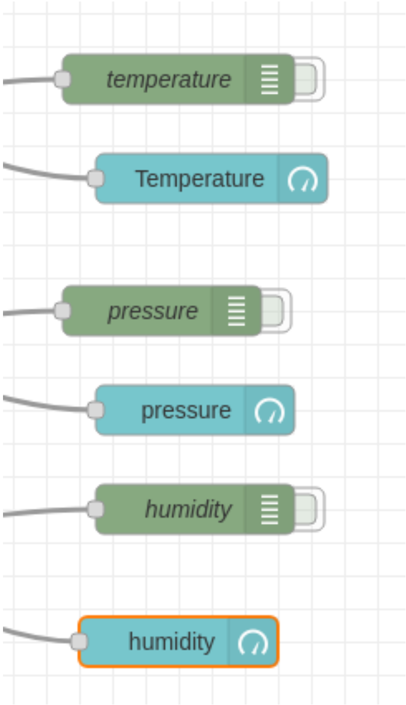


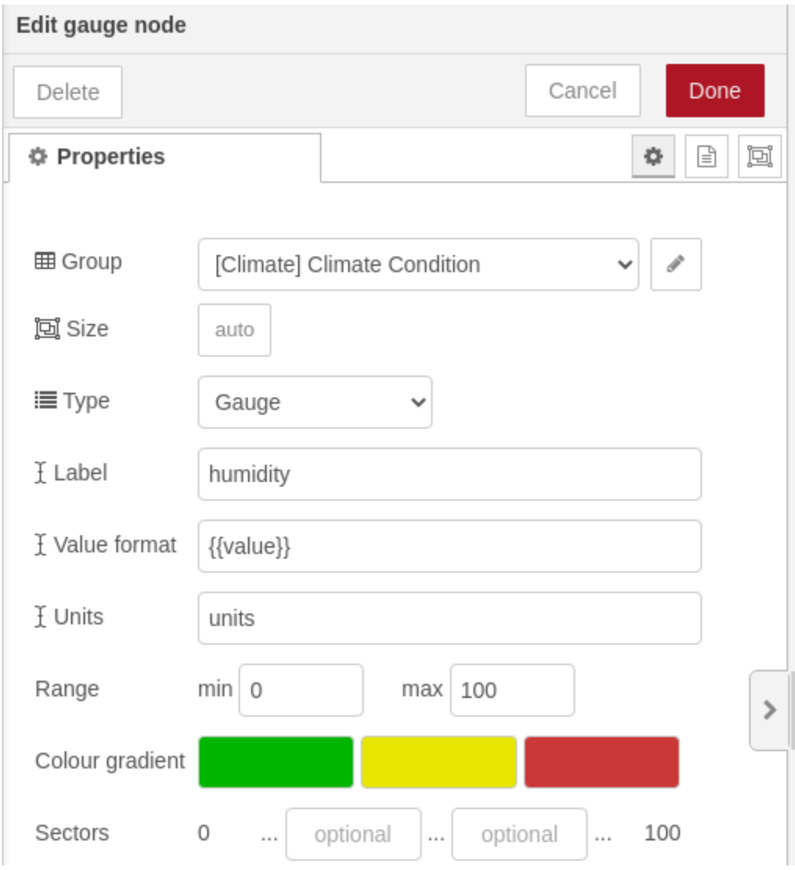
Name: Humidity

```
msg.payload = msg.payload.humidity
return msg;
```

Stage 3 : Output Nodes

This is a collection of 2 nodes used for each of the function nodes specified before, the nodes are a debug node, which displays the corresponding detail of the environmental condition in the debug window, and then the Gauge nodes which display the same data on the form of a gauge in the dashboard component of the node-red UI.





Group: [Climate] Climate Condition

Size: auto

Type: Gauge

Label: humidity

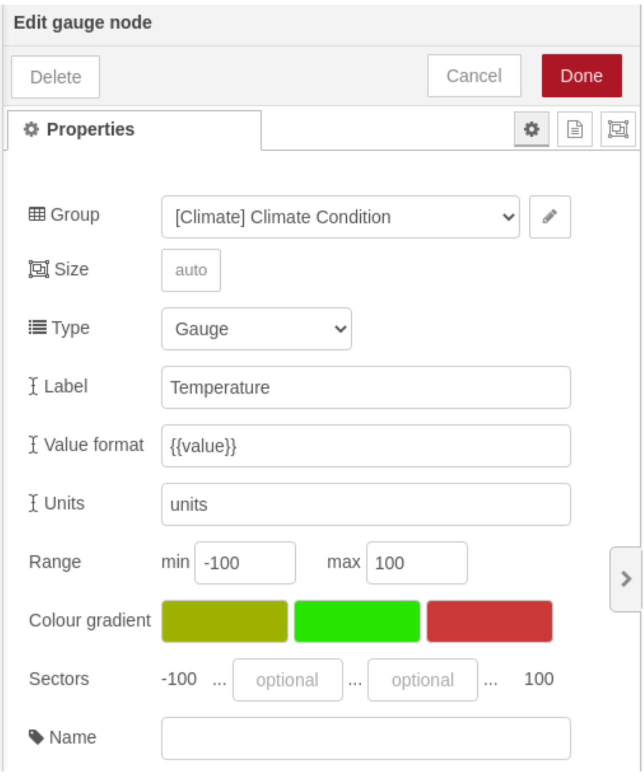
Value format: {{value}}

Units: units

Range: min 0 max 100

Colour gradient: 🟩 🟨 🟥

Sectors: 0 ... optional ... optional ... 100



Edit gauge node

Delete | Cancel | Done

Properties

Group: [Climate] Climate Condition

Size: auto

Type: Gauge

Label: Temperature

Value format: {{value}}

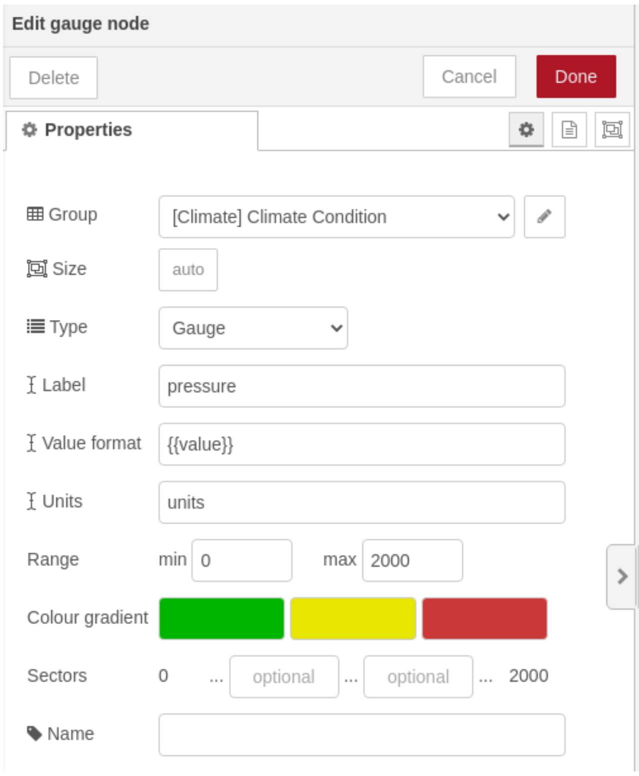
Units: units

Range: min -100 max 100

Colour gradient: 🟩 🟩 🟥

Sectors: -100 ... optional ... optional ... 100

Name:



Edit gauge node

Delete | Cancel | Done

Properties

Group: [Climate] Climate Condition

Size: auto

Type: Gauge

Label: pressure

Value format: {{value}}

Units: units

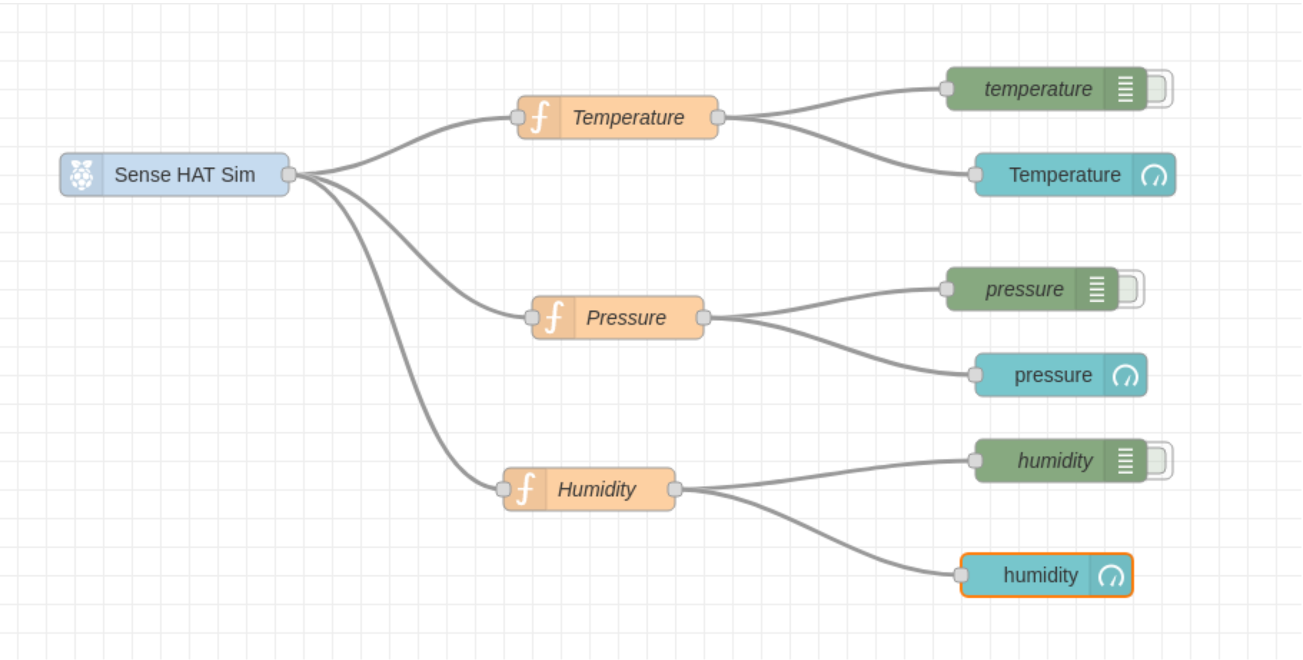
Range: min 0 max 2000

Colour gradient: 🟩 🟨 🟥

Sectors: 0 ... optional ... optional ... 2000

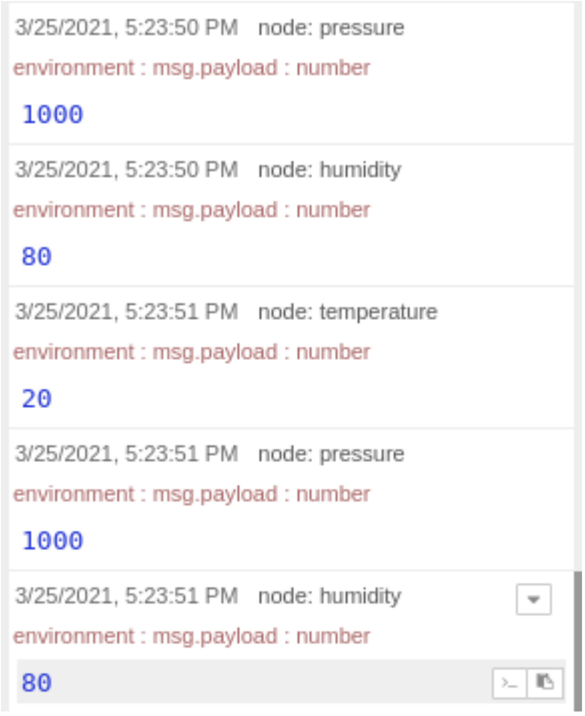
Name:

Stage 4 : Complete Workflow



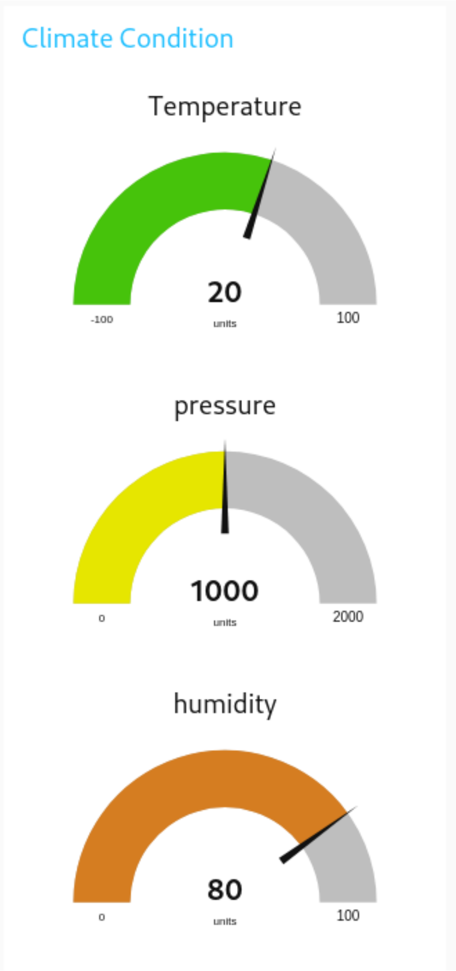
Result :

The following image denotes the environmental conditions that was collected by the function from the API and given as the output in the Debug Window :



```
3/25/2021, 5:23:50 PM   node: pressure
environment : msg.payload : number
1000

3/25/2021, 5:23:50 PM   node: humidity
environment : msg.payload : number
80

3/25/2021, 5:23:51 PM   node: temperature
environment : msg.payload : number
20

3/25/2021, 5:23:51 PM   node: pressure
environment : msg.payload : number
1000

3/25/2021, 5:23:51 PM   node: humidity
environment : msg.payload : number
80
```

The following image is the output obtained in the Dashboard component due to the gauge nodes which display the Temperature, Pressure and the Humidity values as they are obtained from the API :


Climate Condition
Temperature
20
units
-100
100
pressure
1000
units
0
2000
humidity
80
units
0
100