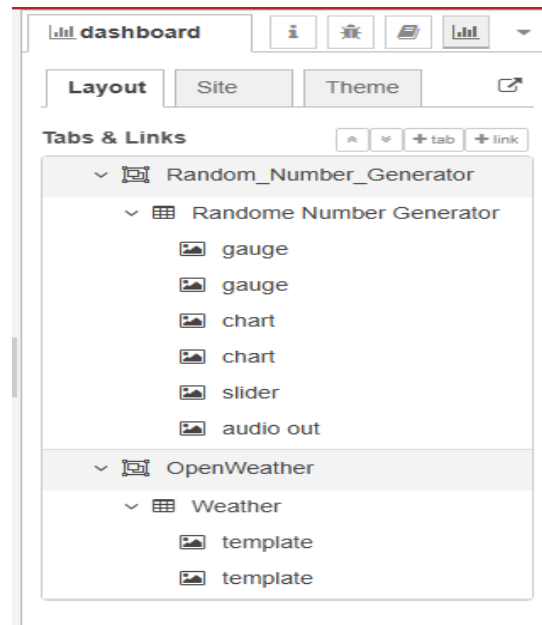


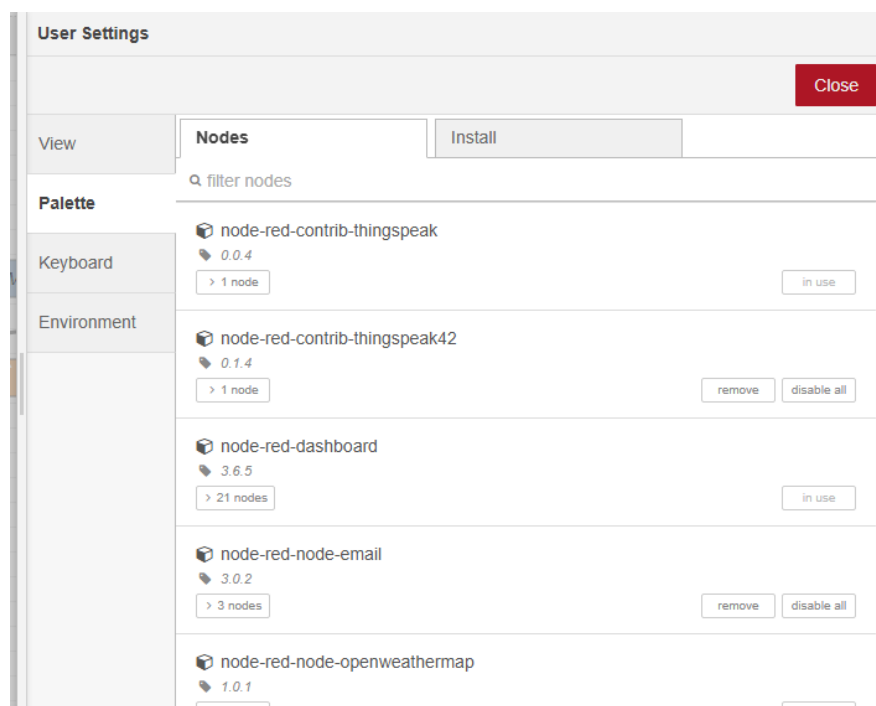
## Practical 10

**Study and implementation of processing data from different sensors and visualize data on Node-red dashboard.**

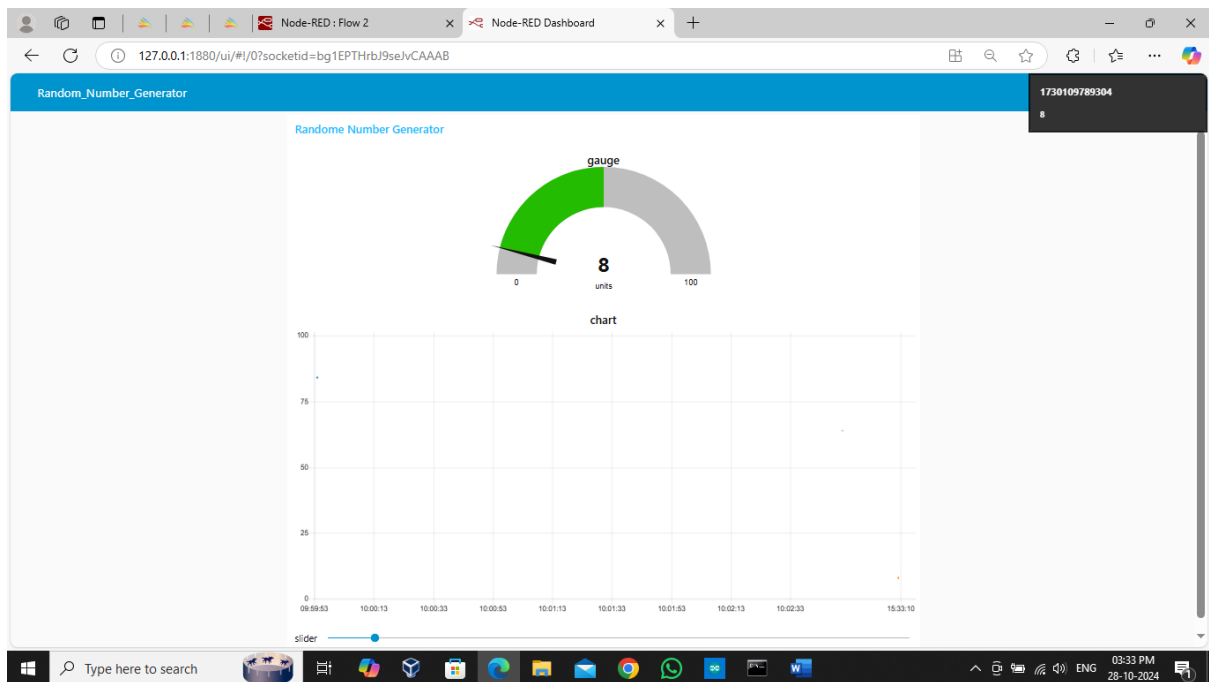
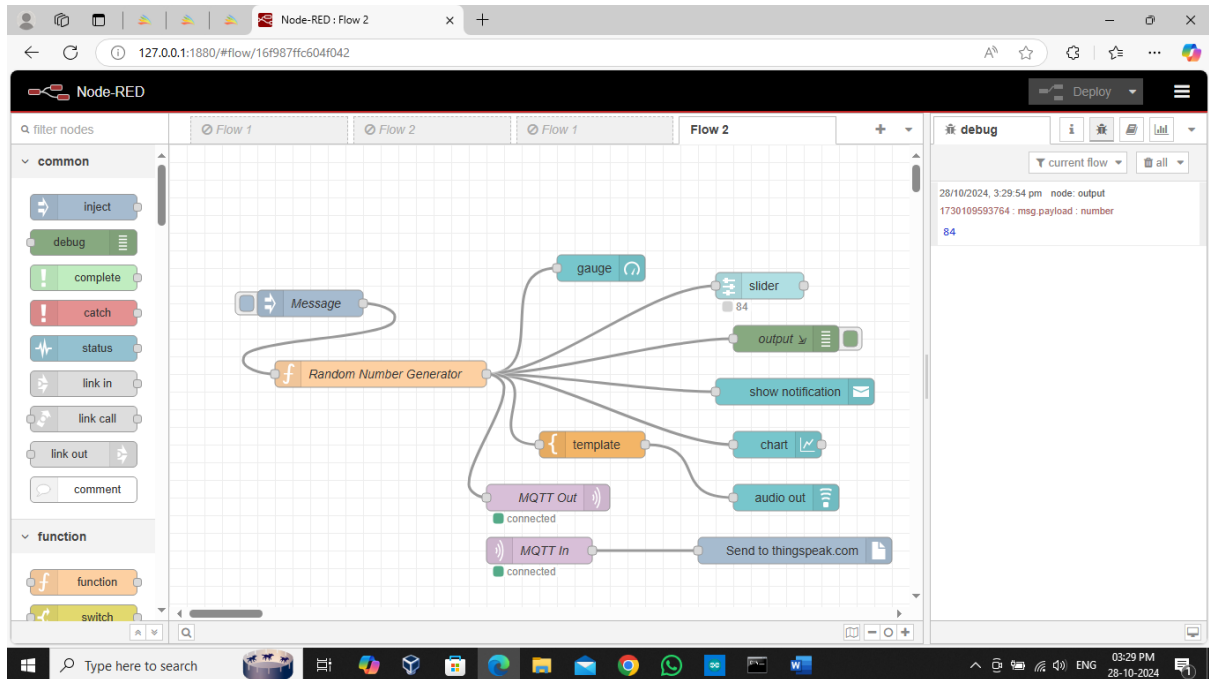
Dashboard layout:

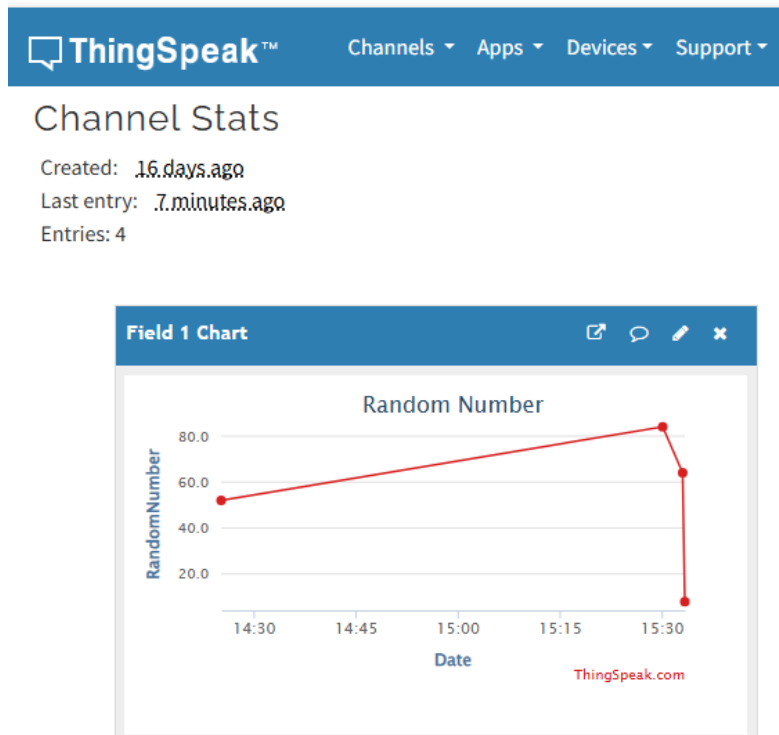


Node-Red Pallet installed module:



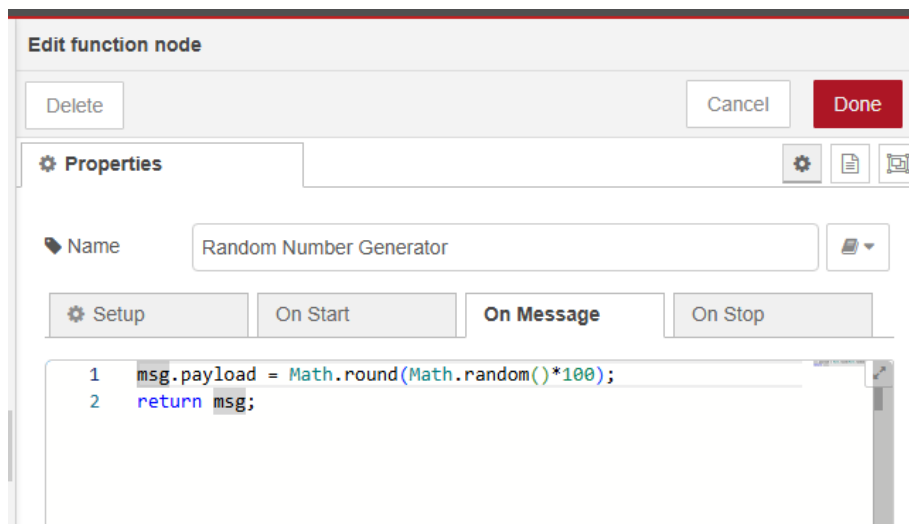
## Random Number Generator With UI





Random Number Generator Code :

```
msg.payload = Math.round(Math.random()*100);  
return msg;
```



### MQTT Configuration:

Note : Before use install Mosquitto application

### MQTT Out:

The screenshot shows the 'Edit mqtt out node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below the buttons is a 'Properties' tab with a gear icon and a document icon. The configuration fields are as follows:

- Server:** A dropdown menu set to 'Mqtt1' with a pencil icon and a plus sign.
- Topic:** A text input field containing 'randomnumber'.
- QoS:** A dropdown menu with a downward arrow.
- Retain:** A checkbox labeled 'Retain' followed by a dropdown menu with a downward arrow.
- Name:** A text input field containing 'MQTT Out'.

Below the fields is a yellow tip box that reads: 'Tip: Leave topic, qos or retain blank if you want to set them via msg properties.'

The screenshot shows the 'Edit mqtt-broker node' configuration window. At the top, there are three buttons: 'Delete', 'Cancel', and 'Update'. Below the buttons is a 'Properties' tab with a gear icon and a document icon. The configuration fields are as follows:

- Name:** A text input field containing 'Mqtt1'.
- Connection:** A tabbed interface with 'Connection', 'Security', and 'Messages' tabs. The 'Connection' tab is active.
- Server:** A text input field containing 'localhost'.
- Port:** A text input field containing '1883'.
- Connect automatically:** A checked checkbox.
- Use TLS:** An unchecked checkbox.
- Protocol:** A dropdown menu set to 'MQTT V3.1.1'.
- Client ID:** A text input field containing 'Leave blank for auto generated'.
- Keep Alive:** A text input field containing '60'.
- Session:** A checked checkbox labeled 'Use clean session'.

At the bottom of the window, there is a status bar showing 'On all flows' and a dropdown menu.

MQTT In:

The screenshot shows the 'Edit mqtt in node' dialog box. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below this is a 'Properties' tab with a settings icon, a document icon, and a preview icon. The properties are as follows:

- Server:** A dropdown menu showing 'Mqtt1' with edit and add icons.
- Action:** A dropdown menu showing 'Subscribe to single topic'.
- Topic:** A text input field containing 'randomnumber'.
- QoS:** A dropdown menu showing '2'.
- Output:** A dropdown menu showing 'auto-detect (parsed JSON object, string or buff'.
- Name:** A text input field containing 'MQTT In'.

Gauge Setting :

The screenshot shows the 'Edit gauge node' dialog box. At the top, there are three buttons: 'Delete', 'Cancel', and 'Done'. Below this is a 'Properties' tab with a settings icon, a document icon, and a preview icon. The properties are as follows:

- Group:** A dropdown menu showing '[Random\_Number\_Generator]' with edit and add icons.
- Size:** A text input field containing '20 x 5'.
- Type:** A dropdown menu showing 'Gauge'.
- Label:** A text input field containing 'gauge'.
- Value format:** A text input field containing '{{value}}'.
- Units:** A text input field containing 'units'.
- Range:** Two text input fields: 'min' with '0' and 'max' with '100'.
- Colour gradient:** Three color swatches: green, yellow, and red.
- Sectors:** A range from '0' to '100' with 'optional' text in the middle.
- Fill gauge from centre:** A checkbox that is checked.

At the bottom, there is a 'Enabled' checkbox which is checked.

Note: - Other UI components are on their default settings

## Using OpenWeatherMap API

Template Code:

```

<div>
  <p>
    <strong>Weather Icon:</strong>
    
  </p>
  <h2>Weather in <span ng-bind-html="msg.payload.location"></span></h2>
  <p><strong>Detail:</strong> <span ng-bind-
html="msg.payload.detail"></span></p>
  <p><strong>Description:</strong> <span ng-bind-
html="msg.payload.description"></span></p>
  <p><strong>Temperature:</strong> <span ng-bind-
html="msg.payload.tempc"></span>°C</p>
  <p><strong>Max Temp:</strong> <span ng-bind-
html="msg.payload.temp_maxc"></span>°C, <strong>Min Temp:</strong> <span ng-
bind-html="msg.payload.temp_minc"></span>°C</p>
  <p><strong>Humidity:</strong> <span ng-bind-
html="msg.payload.humidity"></span>%</p>
  <p><strong>Pressure:</strong> <span ng-bind-
html="msg.payload.pressure"></span> hPa</p>
  <p><strong>Wind Speed:</strong> <span ng-bind-
html="msg.payload.windspeed"></span> m/s</p>
  <p><strong>Wind Direction:</strong> <span ng-bind-
html="msg.payload.winddirection"></span>°</p>
  <p><strong>Clouds:</strong> <span ng-bind-
html="msg.payload.clouds"></span>%</p>
  <p><strong>Sunrise:</strong> <span ng-bind-html="(msg.payload.sunrise *
1000) | date:'shortTime'"></span></p>
  <p><strong>Sunset:</strong> <span ng-bind-html="(msg.payload.sunset *
1000) | date:'shortTime'"></span></p>
</div>

```

**Edit template node**

Delete Cancel Done

**Properties**

Template type: Widget in group

Group: [OpenWeather] Weather

Size: 17 x 12

Class: Optional CSS class name(s) for widget

Name: Name

Template

```
1 <div>
2   <p>
3     <strong>Weather Icon:</strong>
4     <img ng-src="http://openweatherma
5   </p>
```

☒ Pass through messages from input.

☒ Add output messages to stored state.

Enabled

**Edit openweathermap in node**

Delete Cancel Done

**Properties**

API Key: .....

Language: English

Current weather for:

Location: City, Country

City: Anand

Country: India

Name: OpenWeatherMapAPI

Enabled

**dashboard**

Layout Site Theme

**Tabs & Links**

- Random\_Number\_Generator
  - Random Number Generator
- OpenWeather
  - Weather

01:54 PM  
28-10-2024

The image displays two screenshots of a Node-RED environment. The top screenshot shows the Node-RED editor interface with a flow named 'Flow 1'. The flow consists of three nodes: 'OpenWeatherMapAPI', 'template', and 'Output (Debug)'. The 'OpenWeatherMapAPI' node is connected to the 'template' node, which is then connected to the 'Output (Debug)' node. The 'debug' sidebar on the right shows the output of the 'Output (Debug)' node, which is a JSON object containing weather data for Anand.

```
object
  id: 800
  weather: "Clear"
  detail: "clear sky"
  icon: "01d"
  tempk: 308.14
  tempc: 34.9
  temp_maxc: 34.9
  temp_minc: 34.9
  humidity: 30
  pressure: 1006
  maxtemp: 308.14
  mintemp: 308.14
  windspeed: 1.97
  winddirection: 332
  location: "Anand"
  sunrise: 1730077835
  sunset: 1730118797
  clouds: 8
  description: "The weather in Anand at coordinates: 22.5667, 72.9333 is Clear (clear sky)."
```

The bottom screenshot shows the web dashboard for the 'OpenWeather' project. The dashboard has a blue header with the title 'OpenWeather'. Below the header, there is a section titled 'Weather' with a red circular icon. The section displays the following information:

- Weather Icon:** (represented by a red circle)
- Weather in Anand**
- Detail:** clear sky
- Description:** The weather in Anand at coordinates: 22.5667, 72.9333 is Clear (clear sky).
- Temperature:** 34.9°C
- Max Temp:** 34.9°C, **Min Temp:** 34.9°C
- Humidity:** 30%
- Pressure:** 1006 hPa
- Wind Speed:** 1.97 m/s
- Wind Direction:** 332°
- Clouds:** 8%
- Sunrise:** 6:40 AM
- Sunset:** 6:03 PM