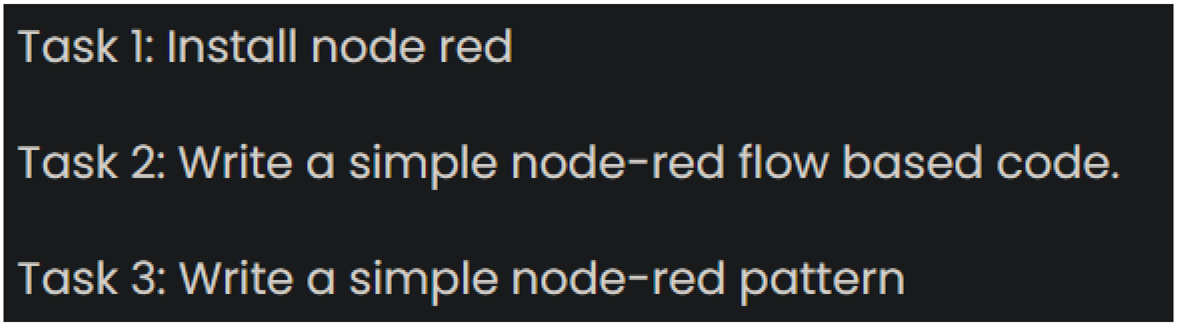
Lab 10

ICS423 - Internet of Things

Jayant Kolapkar - 2021BCS0132

Question



Task 1: Install node red

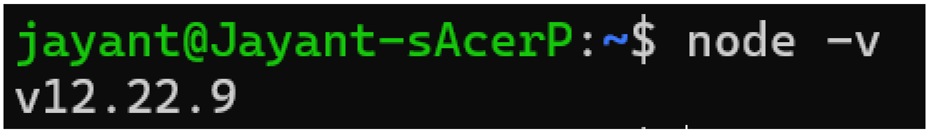
Task 2: Write a simple node-red flow based code.

Task 3: Write a simple node-red pattern

Task 1

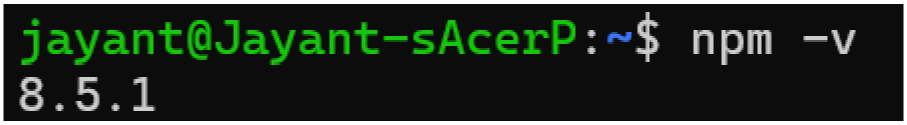
Algorithm

1. Install node.js



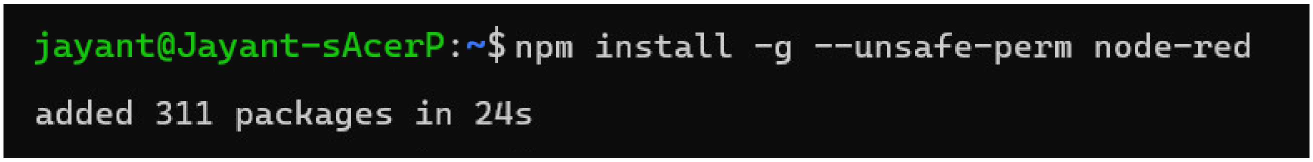
```
jayant@Jayant-sAcerP:~$ node -v
v12.22.9
```

2. Install npm



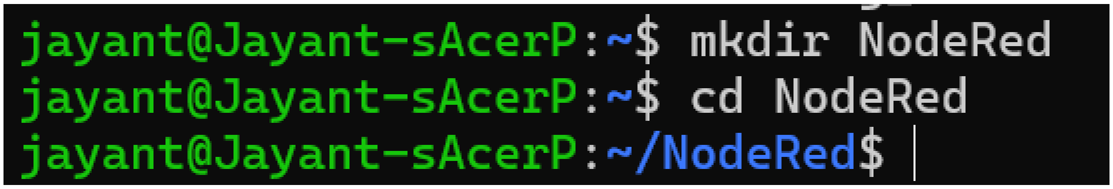
```
jayant@Jayant-sAcerP:~$ npm -v
8.5.1
```

3. Install node-red dependency



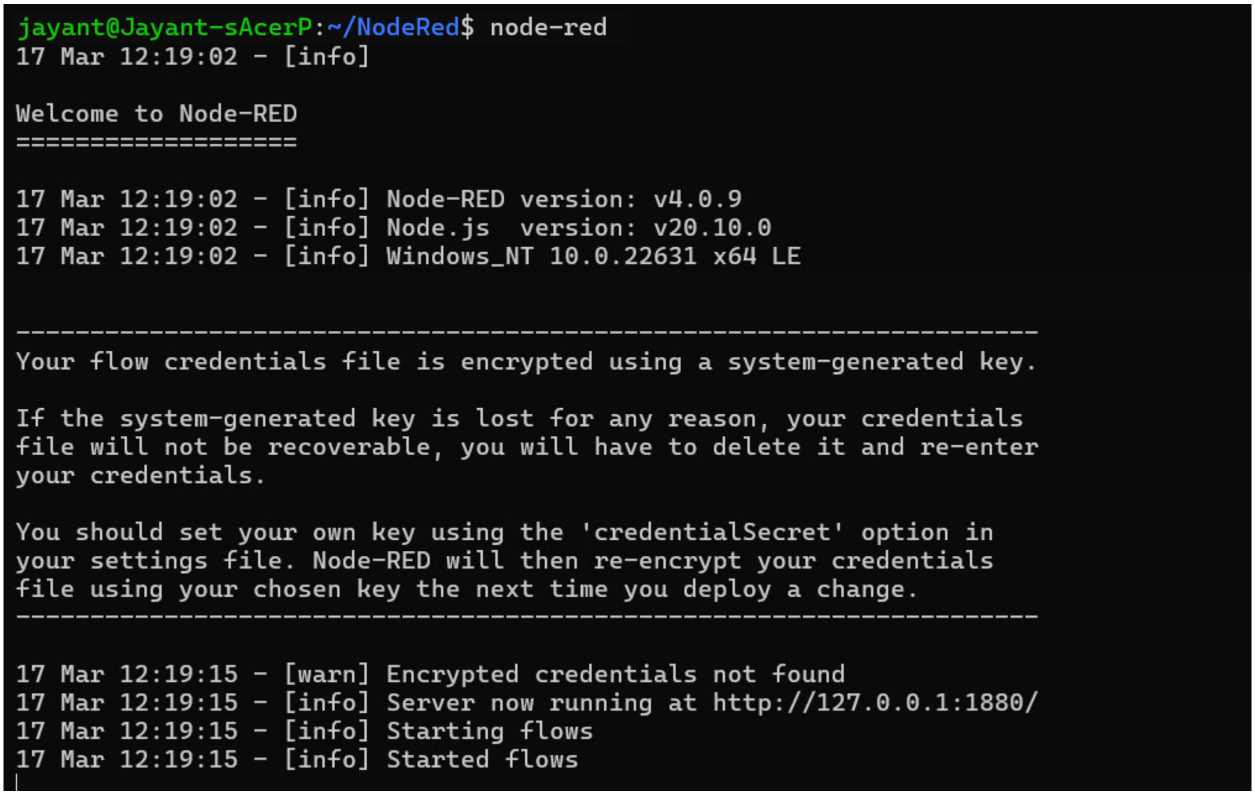
```
jayant@Jayant-sAcerP:~$ npm install -g --unsafe-perm node-red

added 311 packages in 24s
```

4. Create a folder for NodeRed



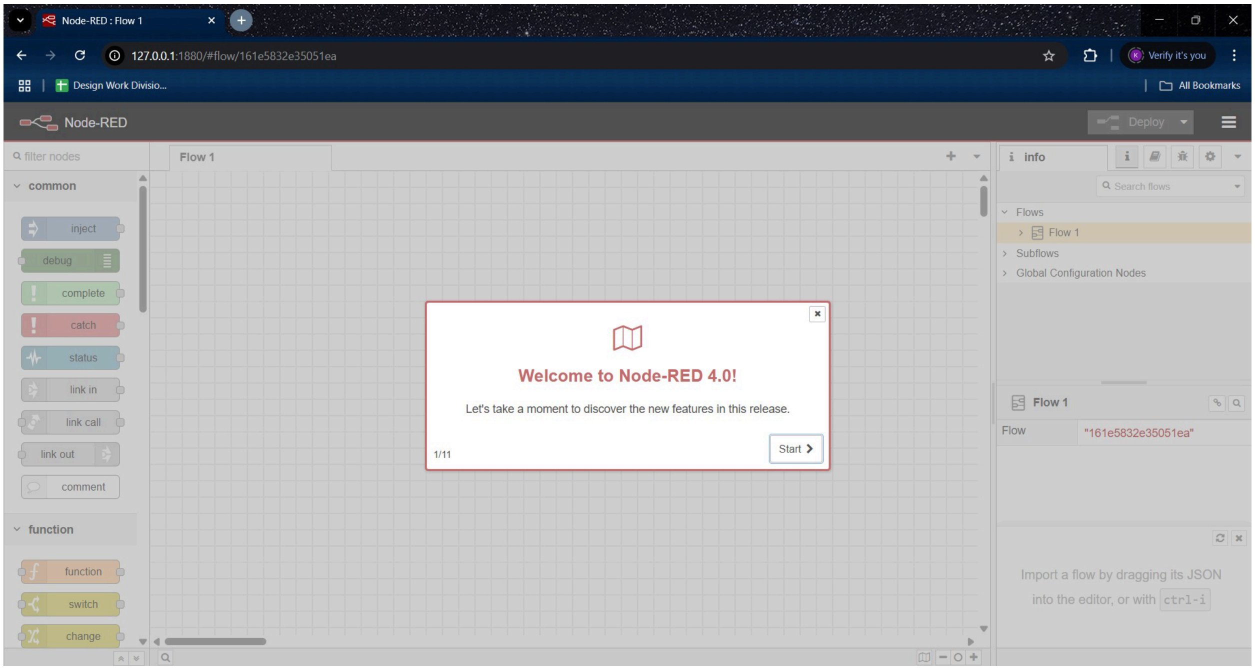
```
jayant@Jayant-sAcerP:~$ mkdir NodeRed
jayant@Jayant-sAcerP:~$ cd NodeRed
jayant@Jayant-sAcerP:~/NodeRed$
```

5. Start node-red using the node-red command. This will create a new flow.



```
jayant@Jayant-sAcerP:~/NodeRed$ node-red
17 Mar 12:19:02 - [info]

Welcome to Node-RED
===================

17 Mar 12:19:02 - [info] Node-RED version: v4.0.9
17 Mar 12:19:02 - [info] Node.js  version: v20.10.0
17 Mar 12:19:02 - [info] Windows_NT 10.0.22631 x64 LE


---------------------------------------------------------------------
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
---------------------------------------------------------------------

17 Mar 12:19:15 - [warn] Encrypted credentials not found
17 Mar 12:19:15 - [info] Server now running at http://127.0.0.1:1880/
17 Mar 12:19:15 - [info] Starting flows
17 Mar 12:19:15 - [info] Started flows
```

6. Now open your browser and visit http://127.0.0.1:1880/. Your node-red will be running here.



127.0.0.1:1880/#flow/161e5832e35051ea

Welcome to Node-RED 4.0!

Let's take a moment to discover the new features in this release.

1/11

Start

Flow "161e5832e35051ea"

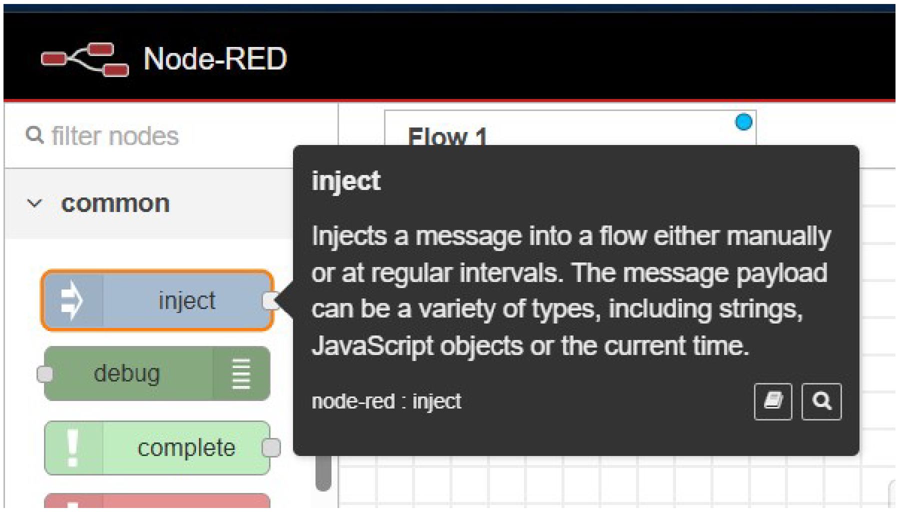
Import a flow by dragging its JSON into the editor, or with ctrl-i

Task 2

Algorithm

1. Open the node-red editor at http://127.0.0.1:1880/.

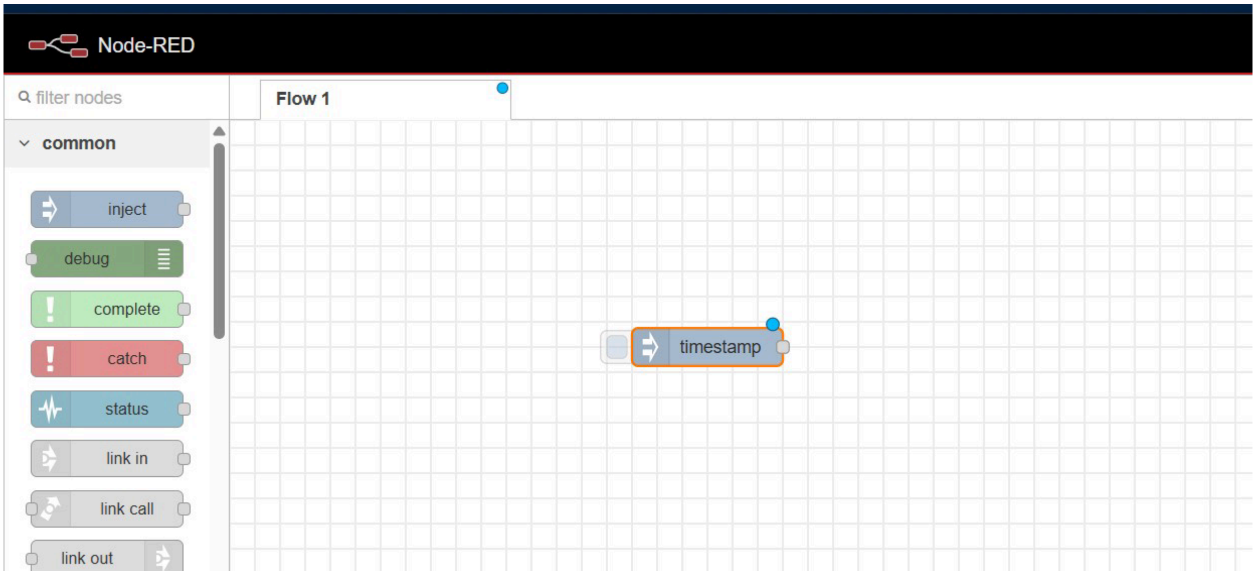
2. Drag inject node to trigger the flow manually.



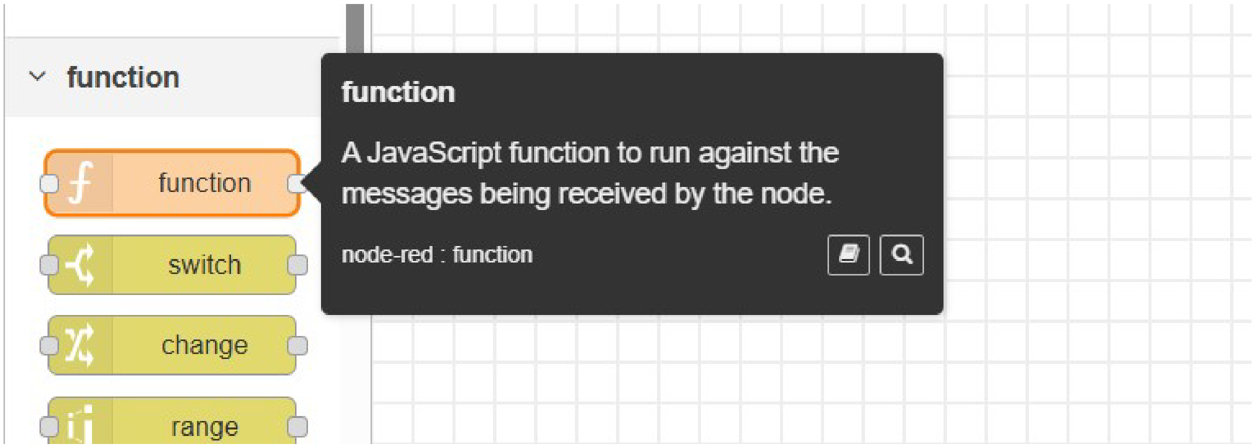
inject

Injects a message into a flow either manually or at regular intervals. The message payload can be a variety of types, including strings, JavaScript objects or the current time.

node-red : inject



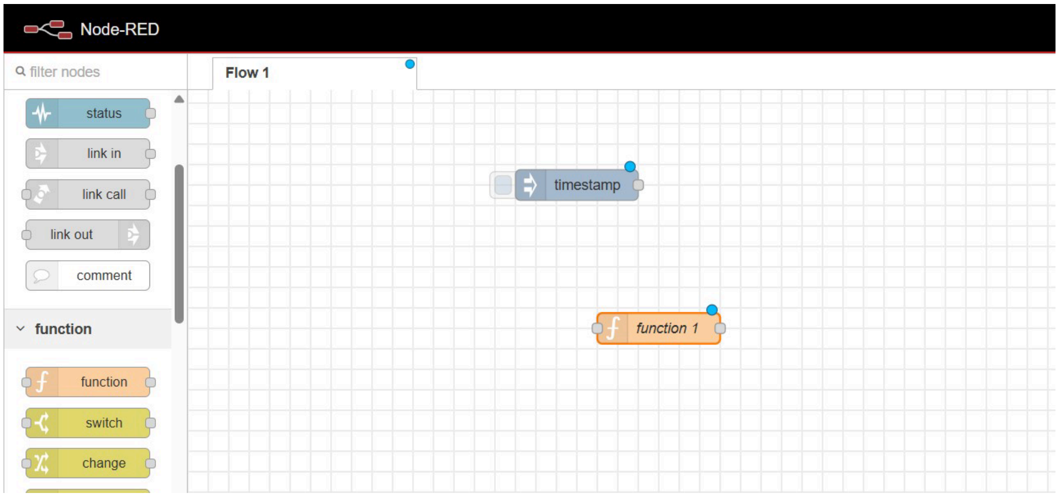
3. Drag a function node to process the data.



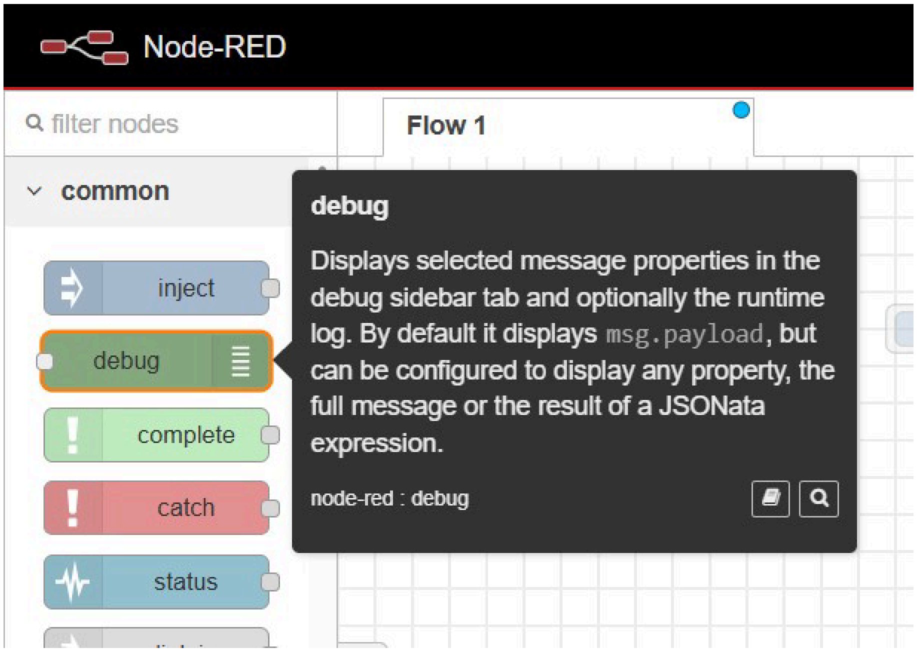
function

A JavaScript function to run against the messages being received by the node.

node-red : function



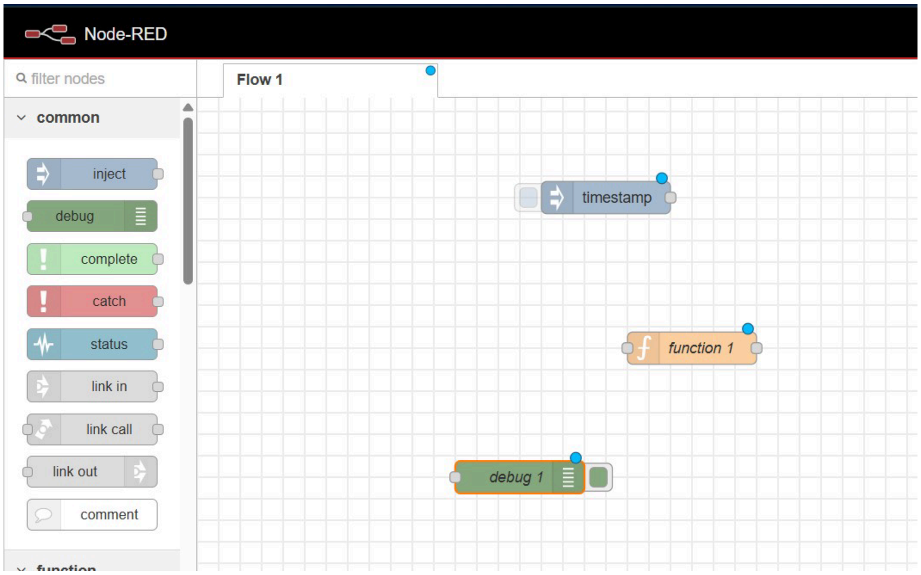
4. Drag a Debug node to log the output.



debug

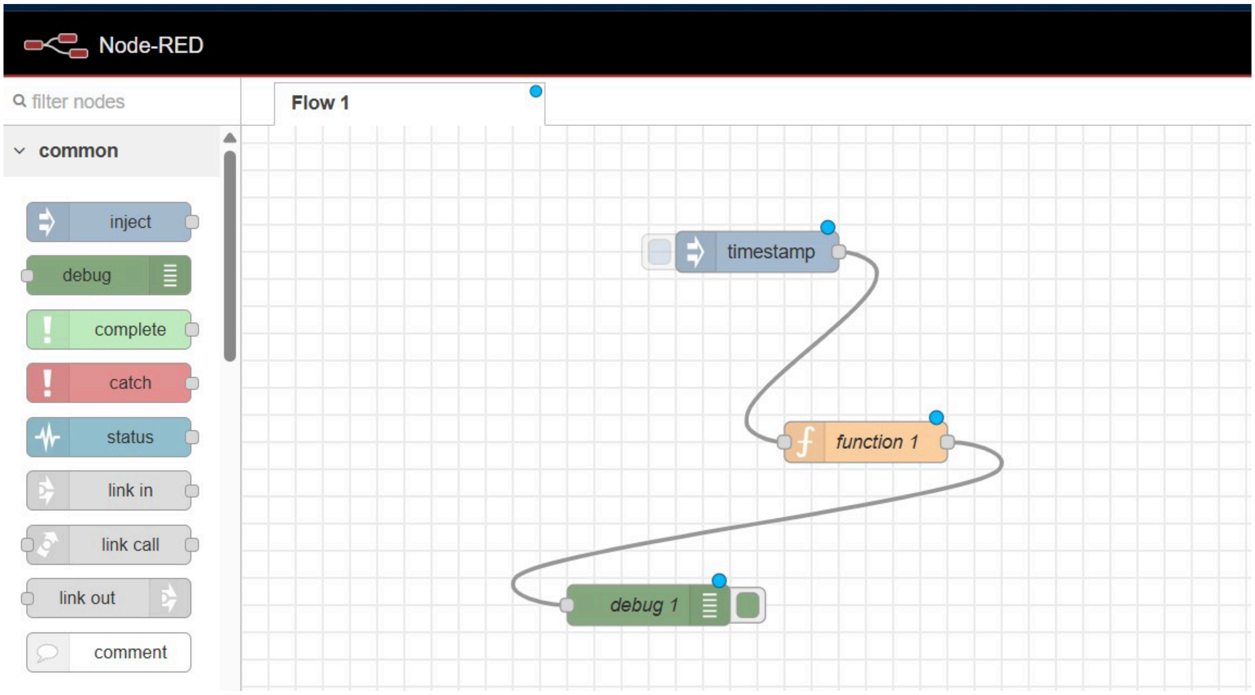
Displays selected message properties in the debug sidebar tab and optionally the runtime log. By default it displays `msg.payload`, but can be configured to display any property, the full message or the result of a JSONata expression.

node-red : debug



5. Connect the nodes in the following manner.

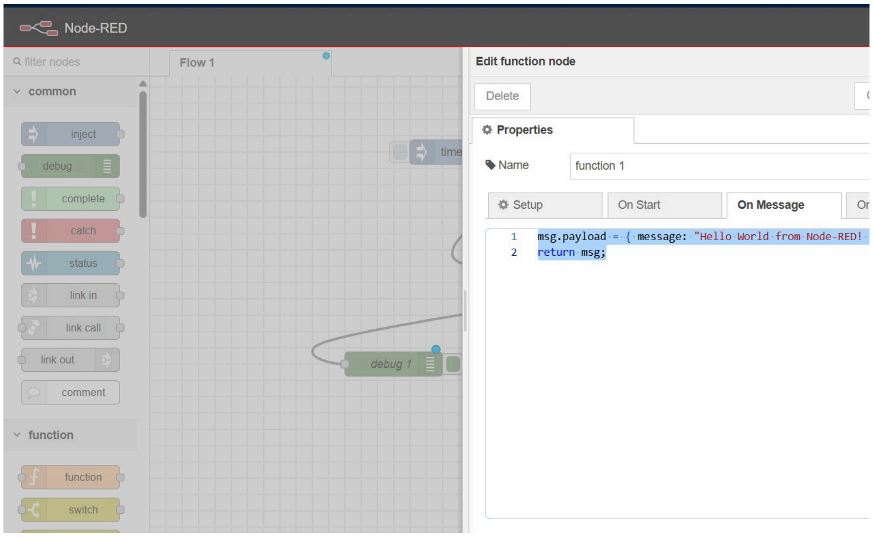
Inject -> Function -> Debug



6. Configure the function with the following js code.

Code

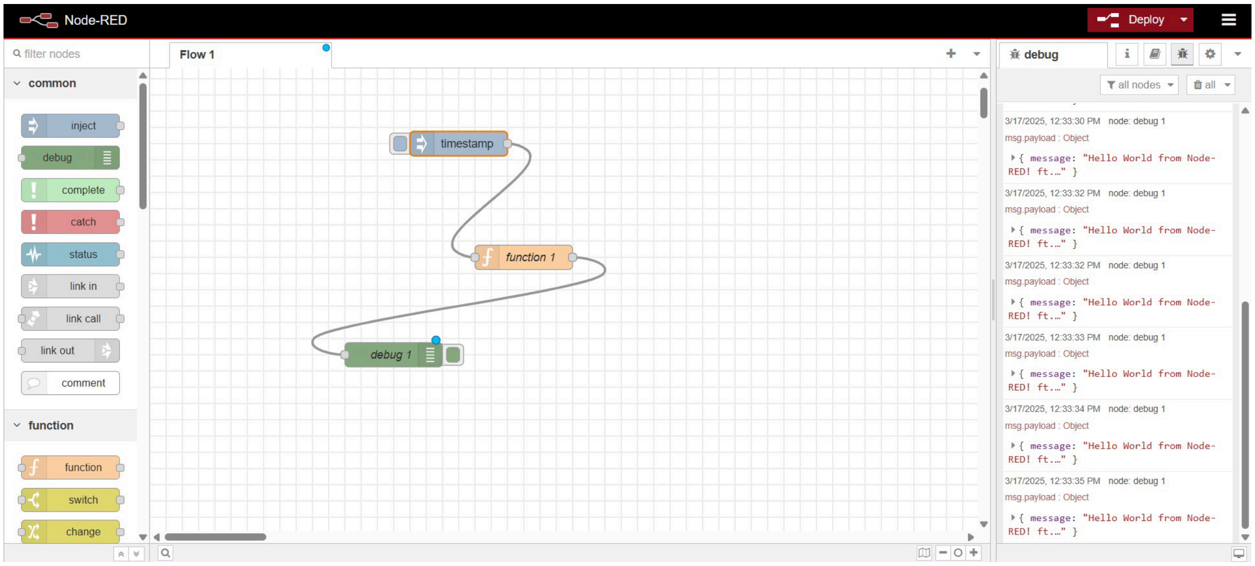
```
msg.payload = { message: "Hello World from Node-RED! ft.Jayant" };

return msg;
```



Name: function 1

```
msg.payload = { message: "Hello World from Node-RED!
return msg;
```

7. Click on the inject node to trigger the flow. Output can be viewed on the debug panel.



```
3/17/2025, 12:33:30 PM   node: debug 1
msg.payload : Object
▸{ message: "Hello World from Node-
RED! ft…" }
3/17/2025, 12:33:32 PM   node: debug 1
msg.payload : Object
▸{ message: "Hello World from Node-
RED! ft…" }
3/17/2025, 12:33:32 PM   node: debug 1
msg.payload : Object
▸{ message: "Hello World from Node-
RED! ft…" }
3/17/2025, 12:33:33 PM   node: debug 1
msg.payload : Object
▸{ message: "Hello World from Node-
RED! ft…" }
3/17/2025, 12:33:34 PM   node: debug 1
msg.payload : Object
▸{ message: "Hello World from Node-
RED! ft…" }
3/17/2025, 12:33:35 PM   node: debug 1
msg.payload : Object
▸{ message: "Hello World from Node-
RED! ft…" }
```

Task 3

Algorithm

1. We will start with creating a flow for the pattern which processes data from a sensor.

2. Flow will be:

- Inject node to simulate sensor data
- Random node to generate random temperature values
- Switch node to check if temp > 50°C
- Change node to format alert message
- Debug node to log the alert

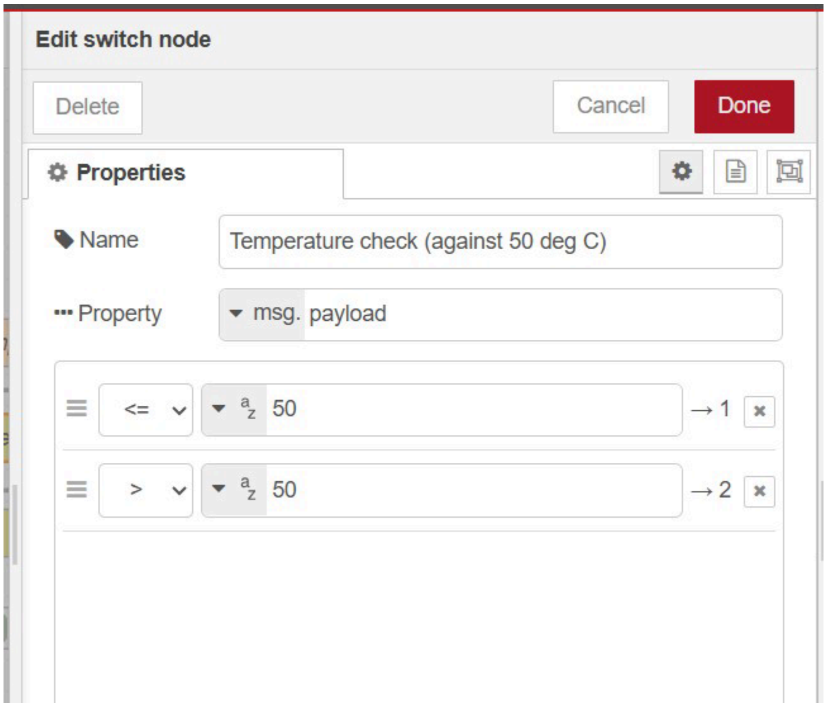
3. The nodes will do the following job.

Random Temp Generator: Generates random temperature values between 10 and

80 deg C.

Switch node: Will check the temperature values and log based on the set threshold

values (here 50 deg C).



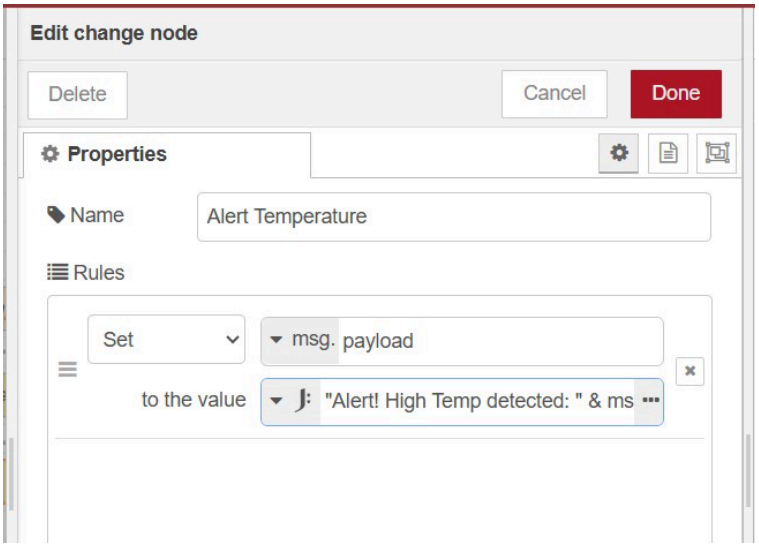
Edit switch node

Name: Temperature check (against 50 deg C)

Property: msg. payload

<=	50	→ 1
>	50	→ 2

Change node: Sets msg.payload = "Alert! High Temp detected."



Edit change node

Name: Alert Temperature

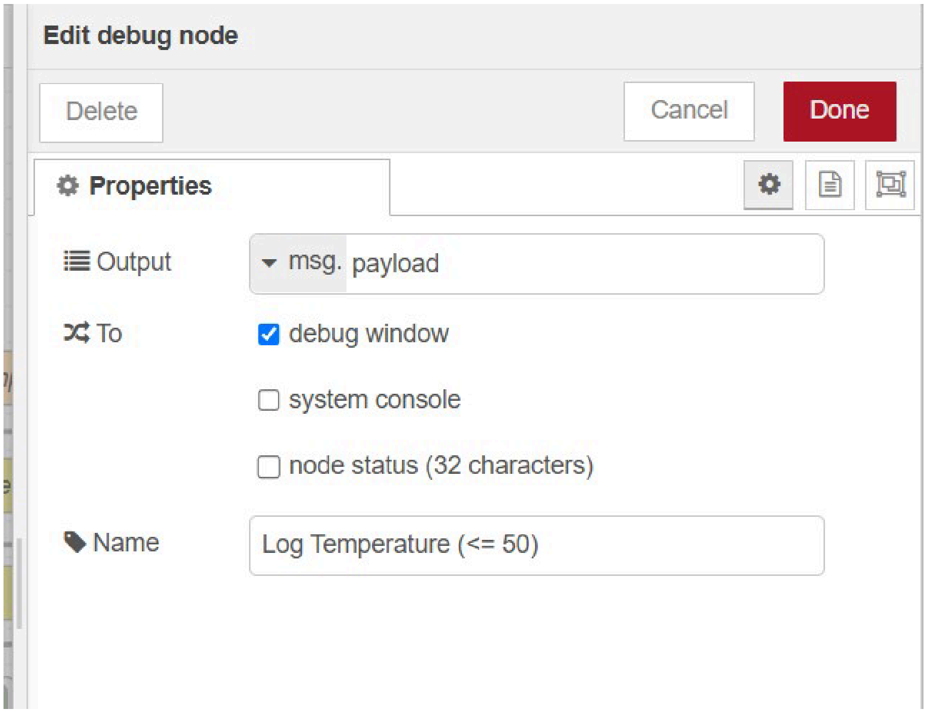
Rules:

Set msg. payload

to the value "Alert! High Temp detected: " & ms

Debug Node:

Log Temperature: If temperature is <= 50 deg C.



Edit debug node

Output: msg. payload

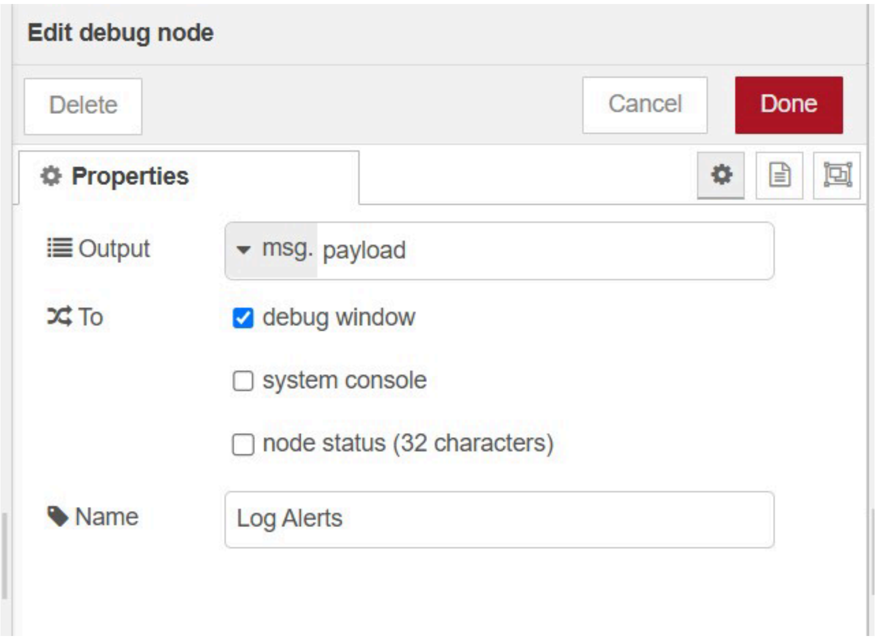
To: ☑ debug window

□ system console

□ node status (32 characters)

Name: Log Temperature (<= 50)

Alert: If Temperature is >50 degC.



Edit debug node

Output: msg. payload

To: ☑ debug window

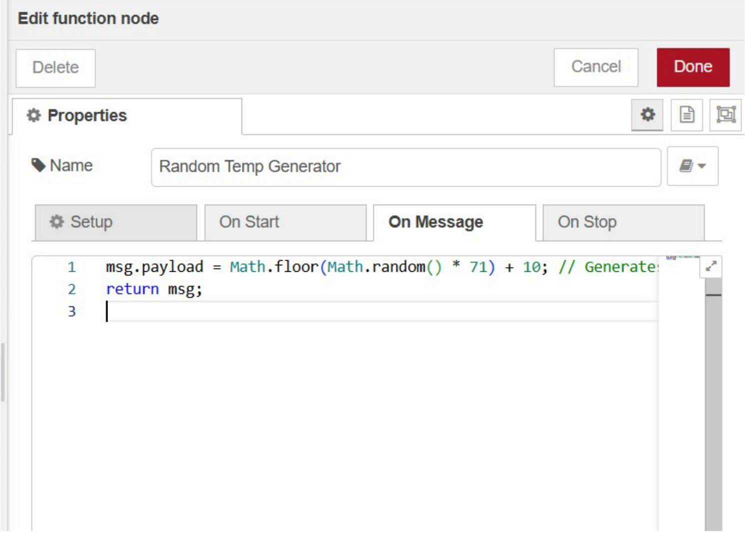
□ system console

□ node status (32 characters)

Name: Log Alerts

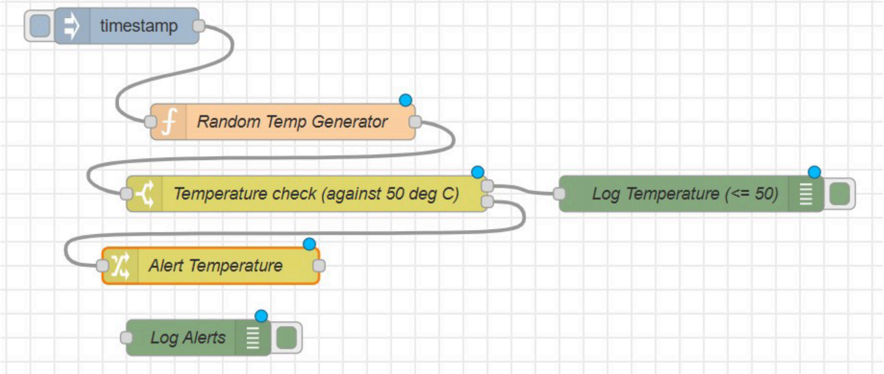
Code for Random Temp Generator function

```
msg.payload = Math.floor(Math.random() * 71) + 10; // Generates temp between 10°C - 80°C

return msg;
```



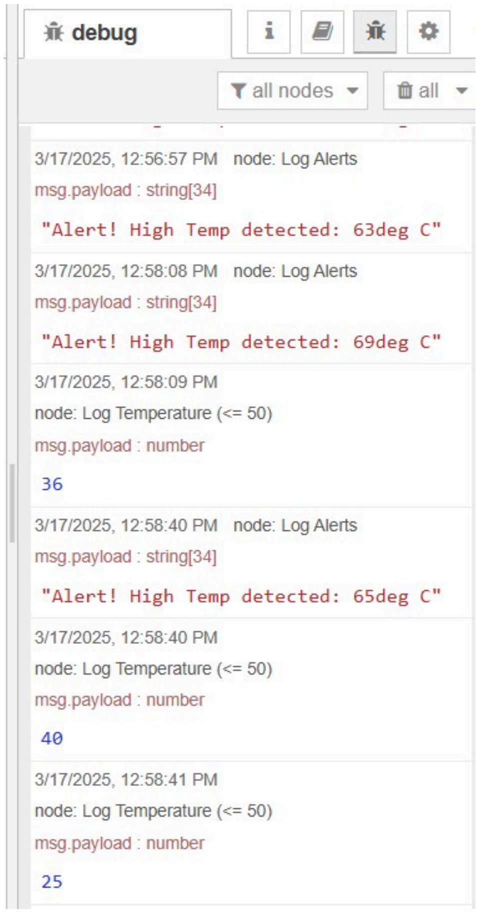
Name: Random Temp Generator

```
msg.payload = Math.floor(Math.random() * 71) + 10; // Generate
return msg;
```



4. Deploy and inject the timestamp node. Output will be visible in the debug mode.

Output



```
3/17/2025, 12:56:57 PM   node: Log Alerts
msg.payload : string[34]
"Alert! High Temp detected: 63deg C"
3/17/2025, 12:58:08 PM   node: Log Alerts
msg.payload : string[34]
"Alert! High Temp detected: 69deg C"
3/17/2025, 12:58:09 PM
node: Log Temperature (<= 50)
msg.payload : number
36
3/17/2025, 12:58:40 PM   node: Log Alerts
msg.payload : string[34]
"Alert! High Temp detected: 65deg C"
3/17/2025, 12:58:40 PM
node: Log Temperature (<= 50)
msg.payload : number
40
3/17/2025, 12:58:41 PM
node: Log Temperature (<= 50)
msg.payload : number
25
```