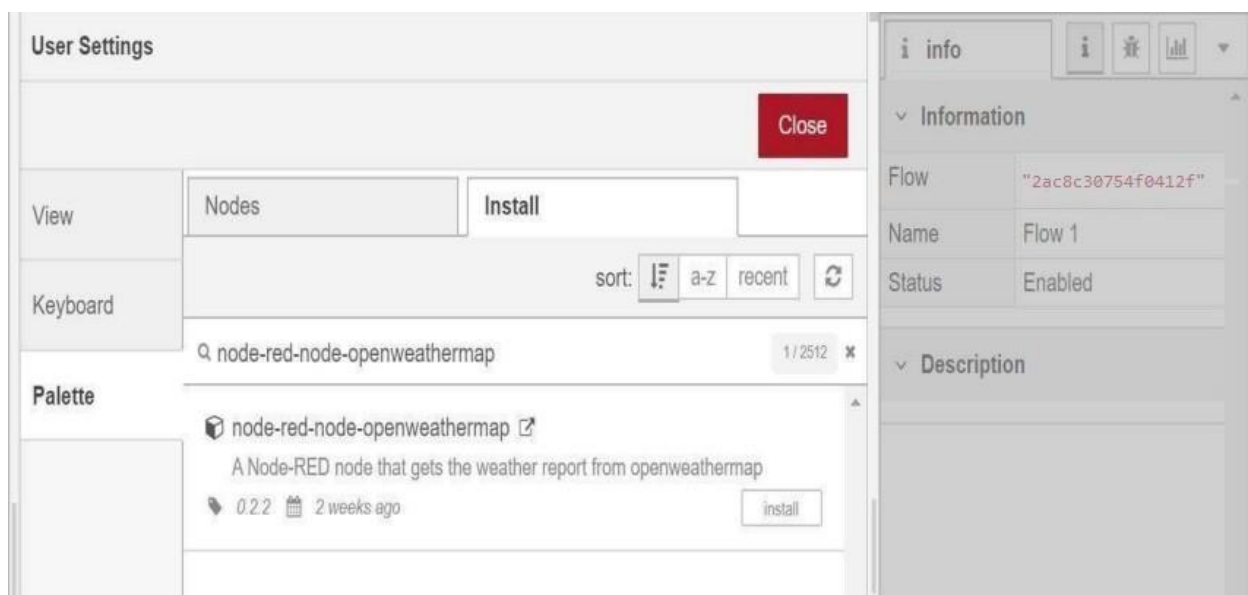


Design your UI to display the water turbidity and pH values

DATE	14 NOVEMBER 2022
TEAM ID	PNT2022TMID14081
PROJECT NAME	Real – Time River Water Quality Monitoring and Control System
MAXIMUM MARKS	4 Marks

1. Double-click the tab with the flow name, and call it Earthquake Details.
2. Click the hamburger menu, and then click Manage palette. Look for node-red-node-openweathermap to install these additional nodes in your palette.



Add an HTTP input node to your flow.

Double-click the node to edit it. Set the method to GET and set the URL to /earthquake info-hr.

1. Add an HTTP response node and connect it to the previously added HTTP input node. All other nodes introduced in this sub-section is to be added between the HTTP input node and the HTTP response node.
2. Add an HTTP request node and set the URL to

https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_hour.geojson, the Method to GET and the Return to a parsed JSON object. This will allow

Edit http request node

Delete Cancel Done

Properties

Method: GET

URL: `http://api.openweathermap.org/data/2.5/forecast?id=5249`

☐ Append msg.payload as query string parameters

☐ Enable secure (SSL/TLS) connection

☐ Use authentication

☐ Enable connection keep-alive

☐ Use proxy

Return: a parsed JSON object

Name: Name

Tip: If the JSON parse fails the fetched string is returned as-is.

info

Information

Node: "2ac8c30754f0412f"

Type: http request

show more

Description

Node Help

Sends HTTP requests and returns the response.

Inputs

url: string
If not configured in the node, this optional property sets the url of the request.

method: string

Hold down **ctrl** when you click on a node to add or remove it from the current

extracting all earthquakes that occurred within the last hour. Name this node Get

Add a **change** node. Double-click the node to modify it. Name this node **Weather** Info. In

the **Rules** section, add

rule to *Delete* msg.topic, msg.headers, msg.statusCode, msg.responseUrl and msg.redirectList

and Set payload.features.
msg.payload

```
{
  "type":properties.type,
  "magnitude": properties.mag,
  "location": properties.place,
  "longitude":geometry.coordinates[0],
  "latitude":geometry.coordinates[1],
  "depth":geometry.coordinates[2],
  "timestamp": $fromMillis( properties.time,
```