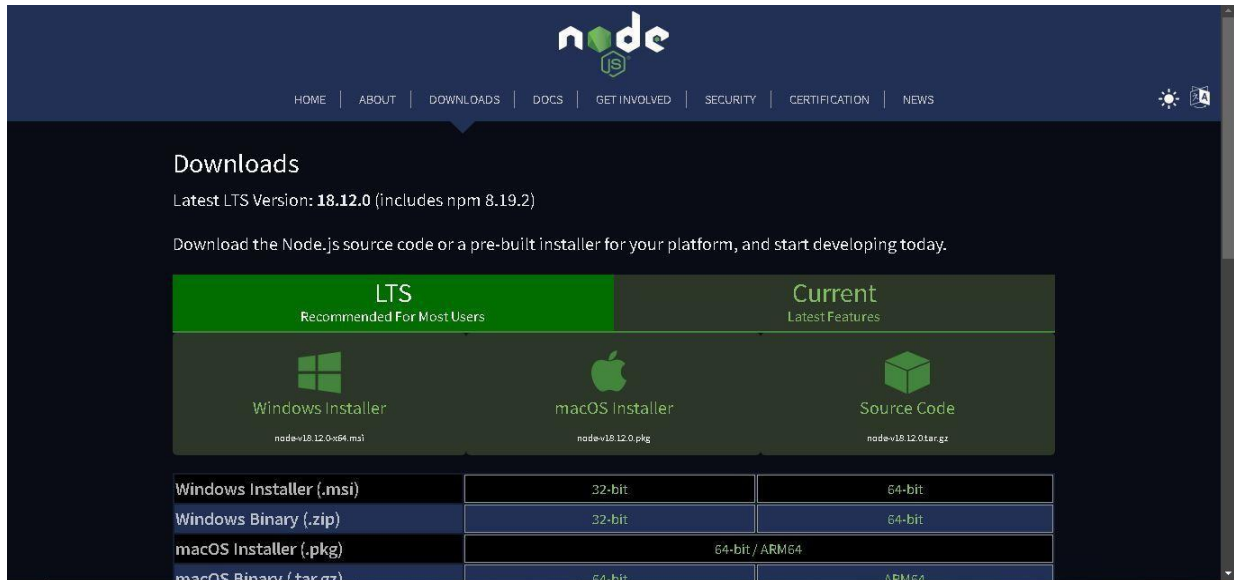


SPRINT-2

Date	05 November 2022
Team ID	PNT2022TMID10541
Project Name	Real Time River Water Quality Monitoring And Control System

STEP 1: Download and Install node.js.



STEP 2: Setup node.js and configure command prompt for error check. Open node-red from the generated link.

```
npm
C:\Users\Ajay>npm install -g --unsafe-perm node-red
npm WARN deprecated @types/keyv@4.2.0: This is a stub types definition. keyv provides its own type definitions, so you do not need this installed.

added 292 packages, and audited 293 packages in 5m

39 packages are looking for funding
  run `npm fund` for details

5 vulnerabilities (4 low, 1 moderate)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

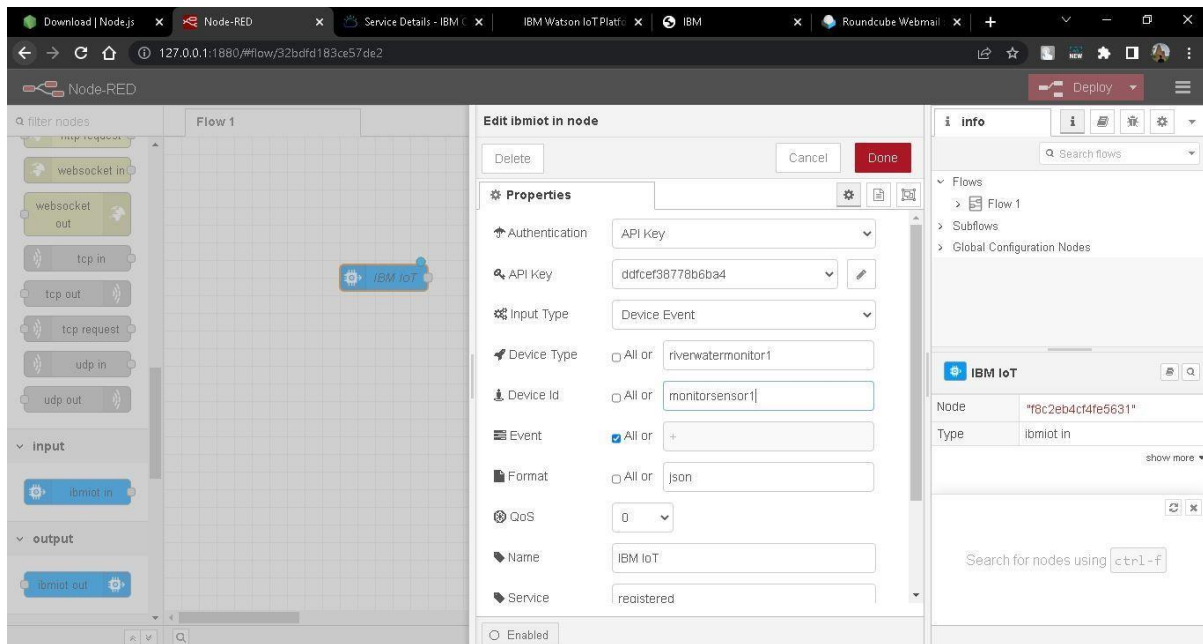
C:\Users\Ajay>node-red
3 Nov 14:35:28 - [info]

Welcome to Node-RED
=====

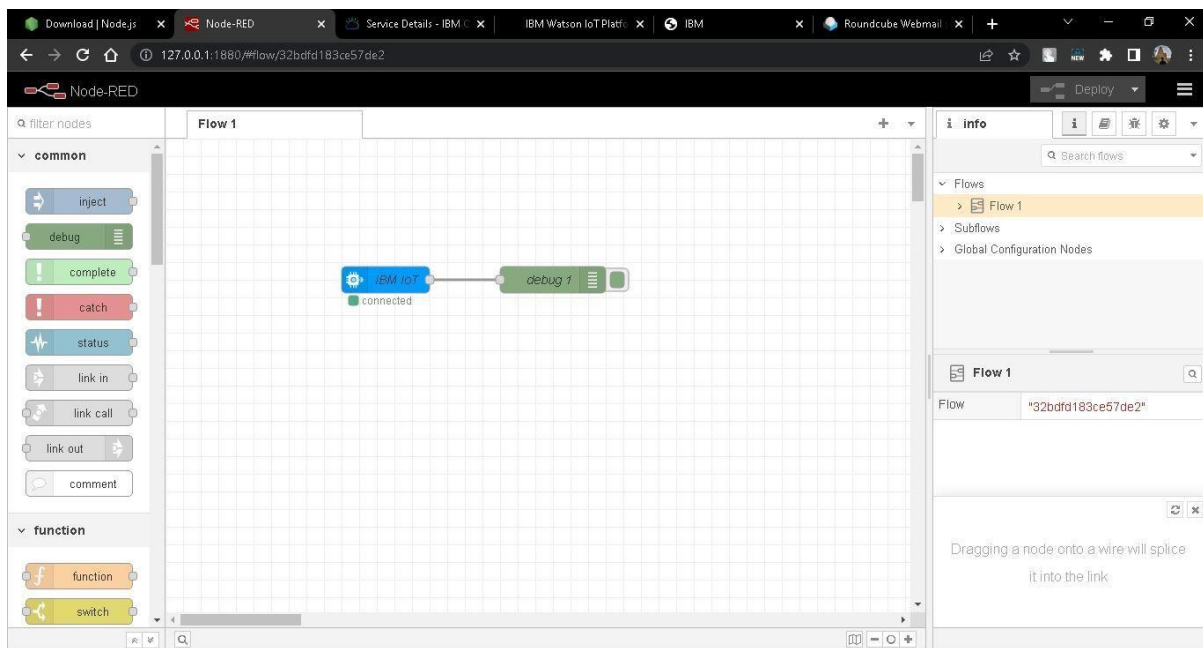
3 Nov 14:35:29 - [info] Node-RED version: v3.0.2
3 Nov 14:35:29 - [info] Node.js version: v18.12.0
3 Nov 14:35:29 - [info] Windows_NT 10.0.19043 x64 LE
3 Nov 14:35:44 - [info] Loading palette nodes
```

STEP 3: Generating API key and Authentication token.

STEP 4: Edit IBM IOT in node.



STEP 5: Connect Ibm iot in and debug 1 and deploy.



STEP 6: Edit gauge node (here the gauge nodes are named as Temperature, pH and Turbidity).

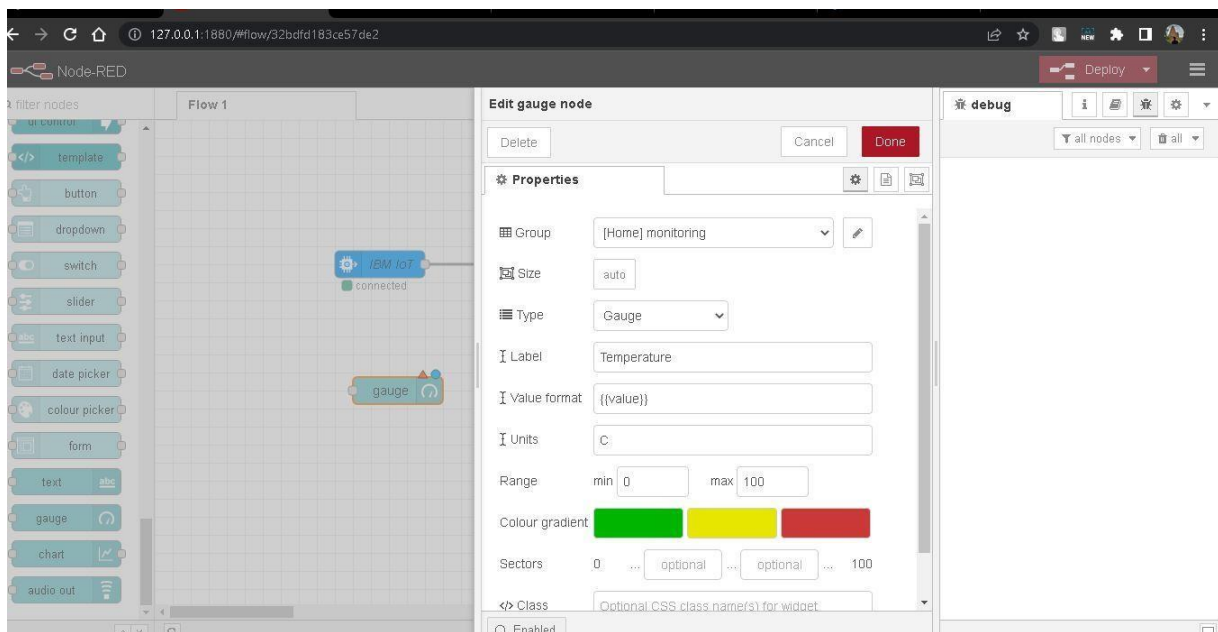


Fig 1

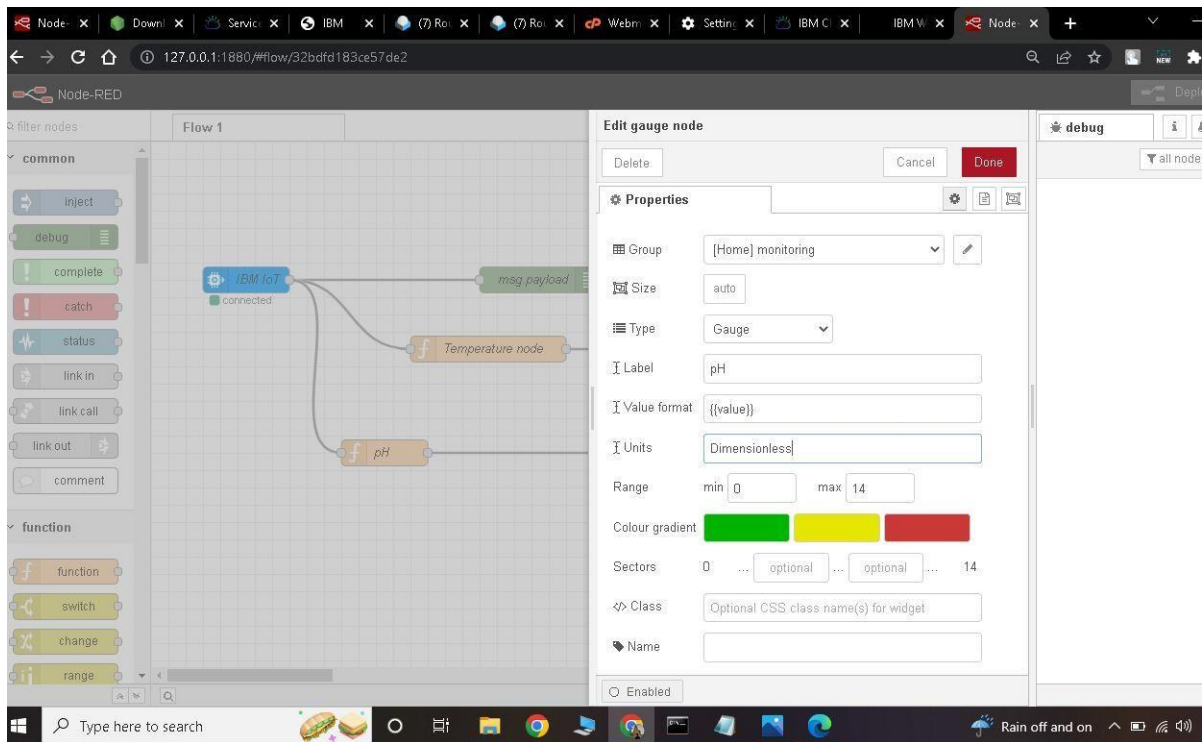


Fig 2

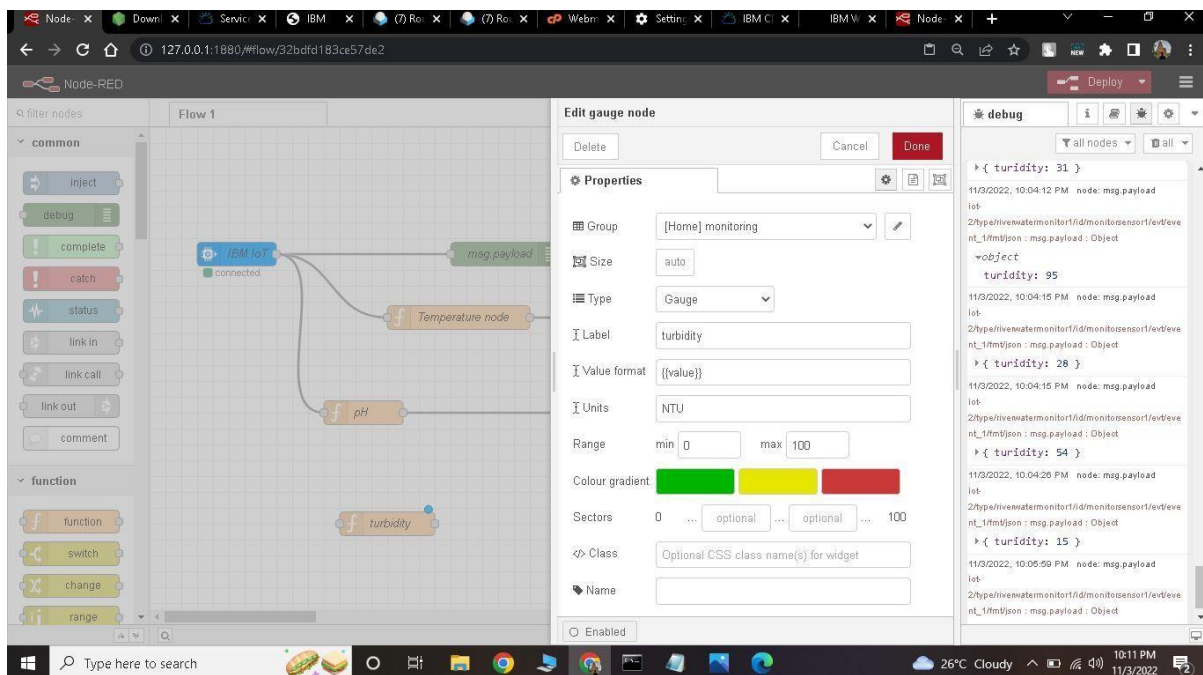


Fig 3

STEP 7: Simulated program to get the random values.

The screenshot shows the IBM Watson IoT Platform interface. On the left, there's a sidebar with navigation icons. The main area displays a table of devices. A modal window is open for the device 'monitorsensor1' under the 'riverwatermonitor1' group. The modal shows the 'Events' tab with one event named 'event_1' set to trigger '20 x Every Minute'. The payload is a JSON object with three fields: 'temperature', 'pH', and 'turbidity', each assigned a random value between 0 and 100. The 'Send' button is visible in the top right of the modal.

Event type name	Frequency
event_1	20 x Every Minute

Payload

```
{
  "temperature": random(0, 100),
  "pH": random(0, 100),
  "turbidity": random(0, 100)
}
```

STEP 8: Generate debug message from IBM Watson IoT Platform and connect the nodes.

The screenshot shows the Node-RED interface. A flow named 'Flow 1' is visible. It starts with an 'IBM IoT' node (connected). The output of this node is connected to a 'msg.payload' node. From 'msg.payload', the flow branches into four parallel paths, each leading to a sensor node: 'Temperature node', 'pH', 'Humidity', and 'turbidity'. Each sensor node is connected to a corresponding output node (Temperature, Humidity, and turbidity). The 'debug' tab on the right shows a list of messages, including the payload object and the sensor values.

```
{
  "turbidity": 31
}
```

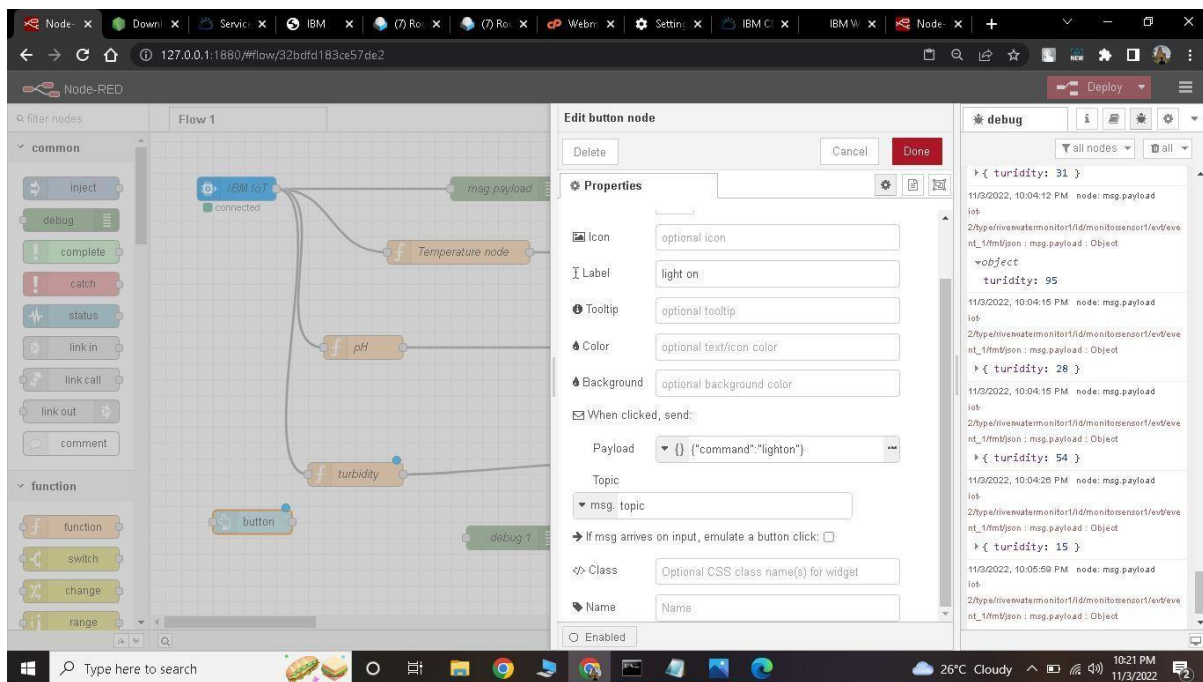
```
{
  "turbidity": 95
}
```

```
{
  "turbidity": 28
}
```

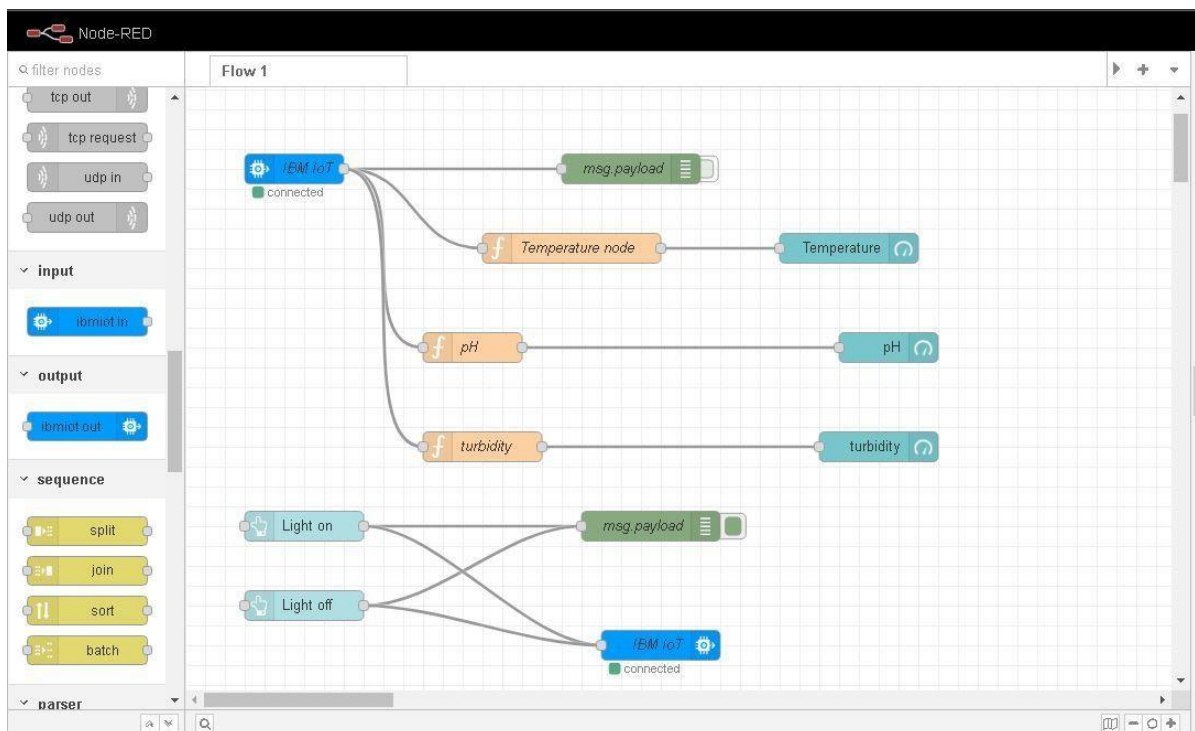
```
{
  "turbidity": 54
}
```

```
{
  "turbidity": 15
}
```

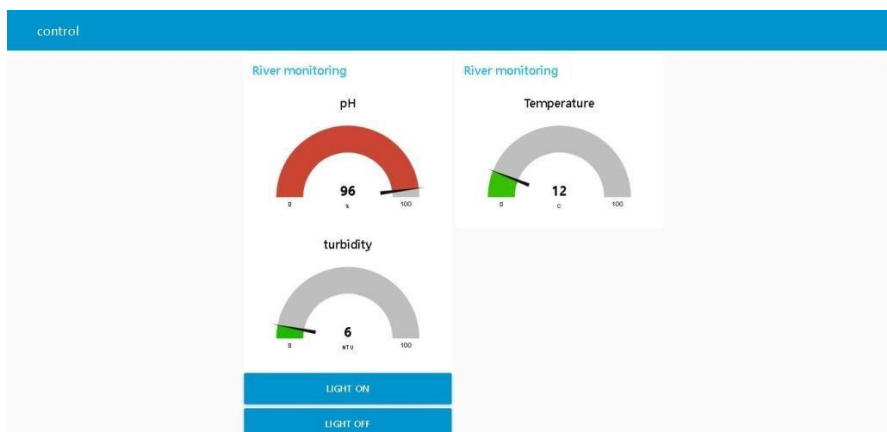
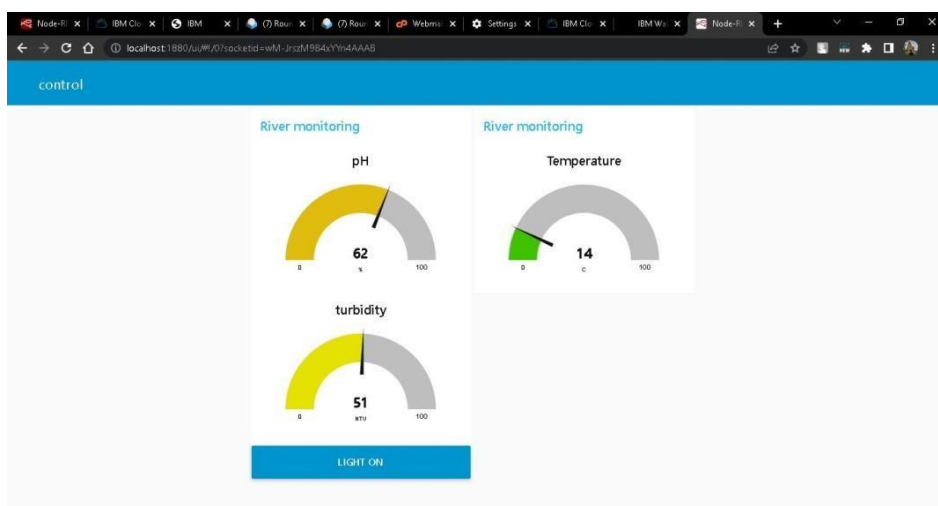
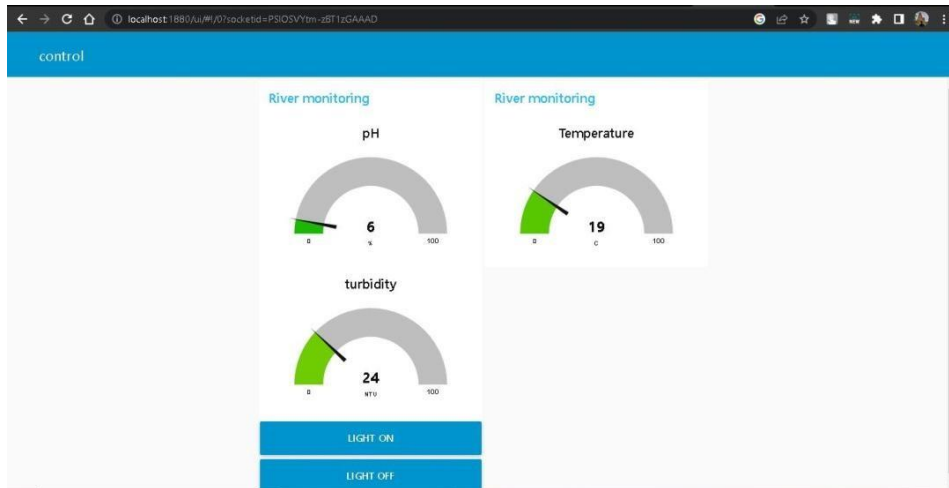
STEP 9: Edit button mode [light ON and light OFF].



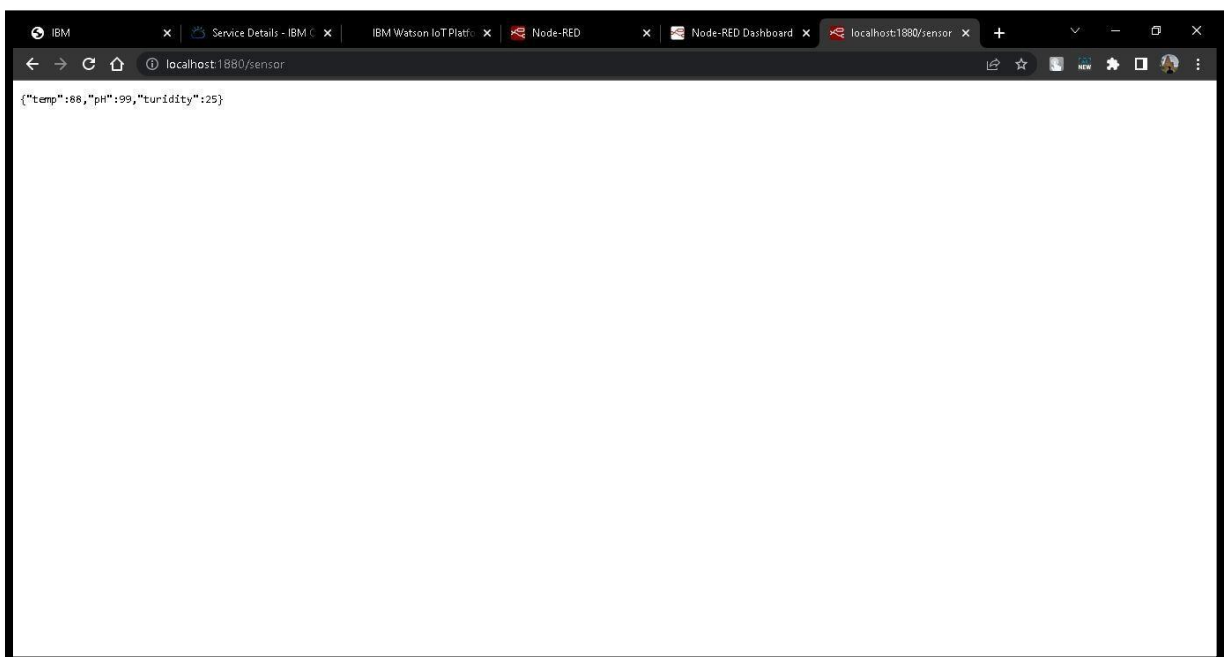
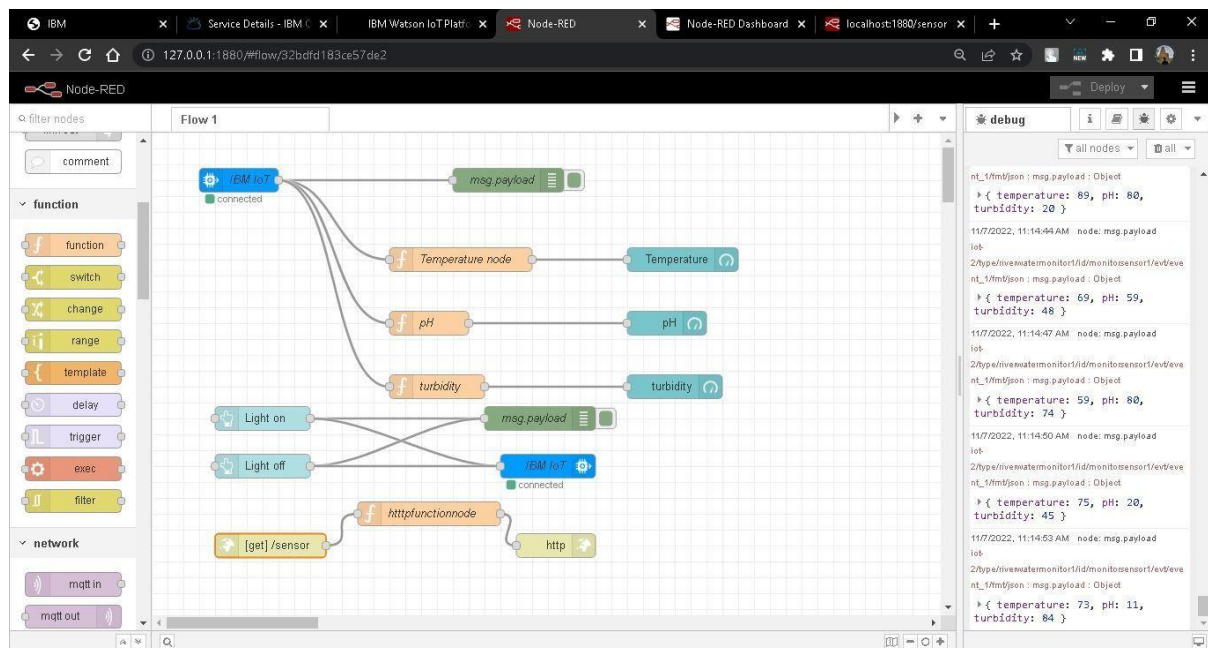
STEP 10: Entire flow diagram in Node-RED.



STEP 11: Generate the output from recent events.



STEP 12: Implementing url in the function node to generate output.



URL are:

`localhost:1880/ui`

`localhost:1880/sensor`

STEP 13: MIT app inventor to design the app.

