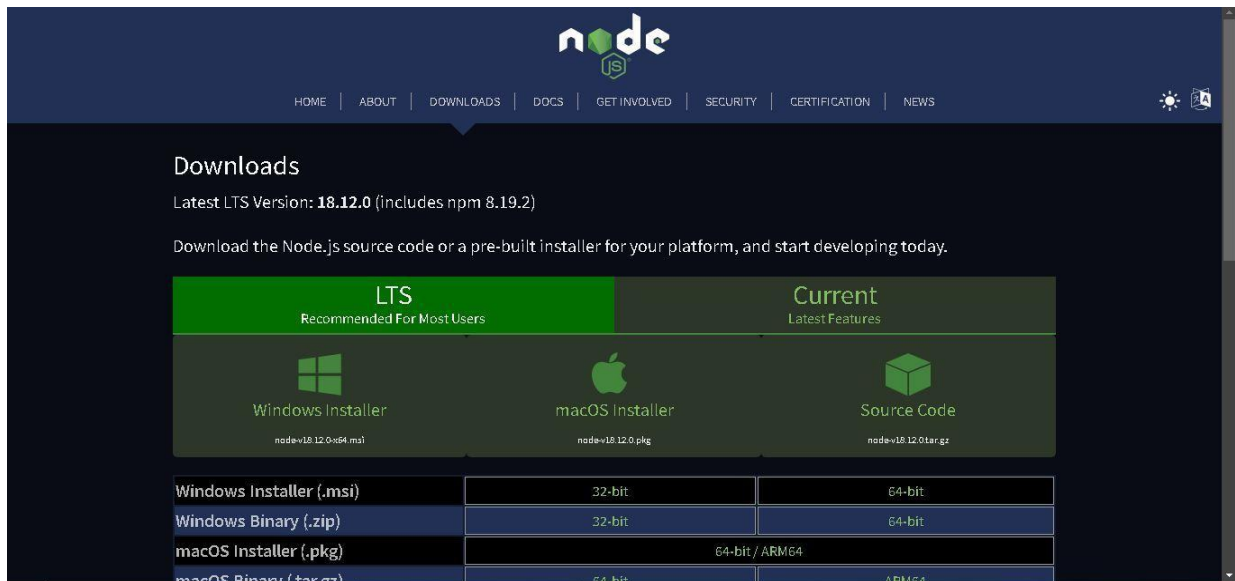


Date	09 November 2022
Team ID	PNT2022TMID38427
Project Name	Real Time River Water Quality Monitoring And Control System

## 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.

```

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.

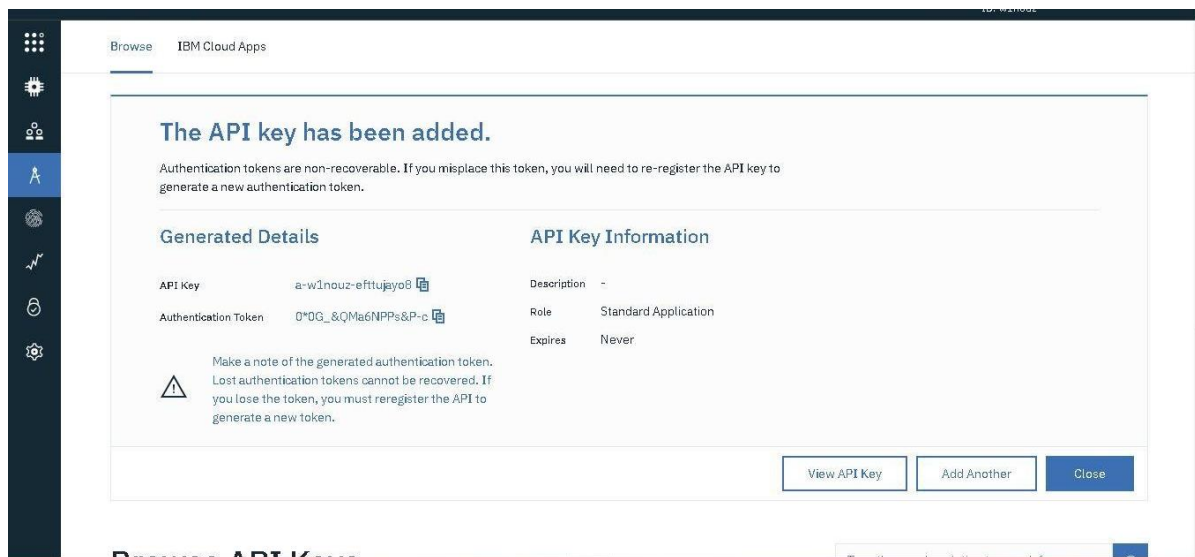
C:\Users\Ajay>node-red
3 Nov 14:35:28 - [info]

Welcome to Node-RED
=====

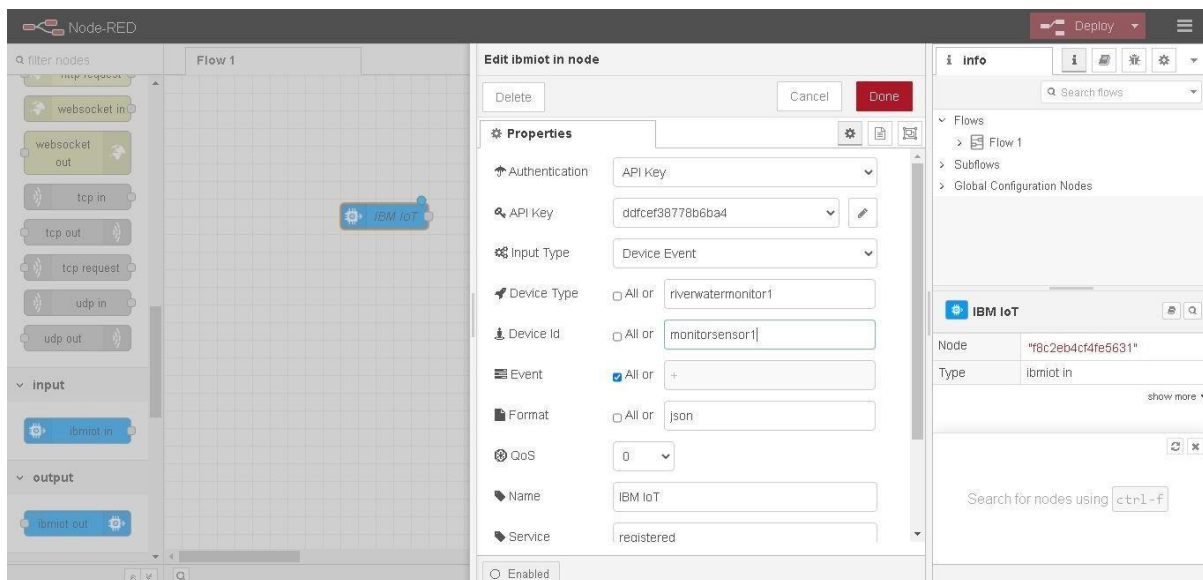
3 Nov 14:35:29 - [info] Node-RED version: v3.0.2
3 Nov 14:35:29 - [info] Node.js version: v18.12.0
3 Nov 14:35:29 - [info] Windows_NT 10.0.19043 x64 LE
3 Nov 14:35:44 - [info] Loading palette nodes

```

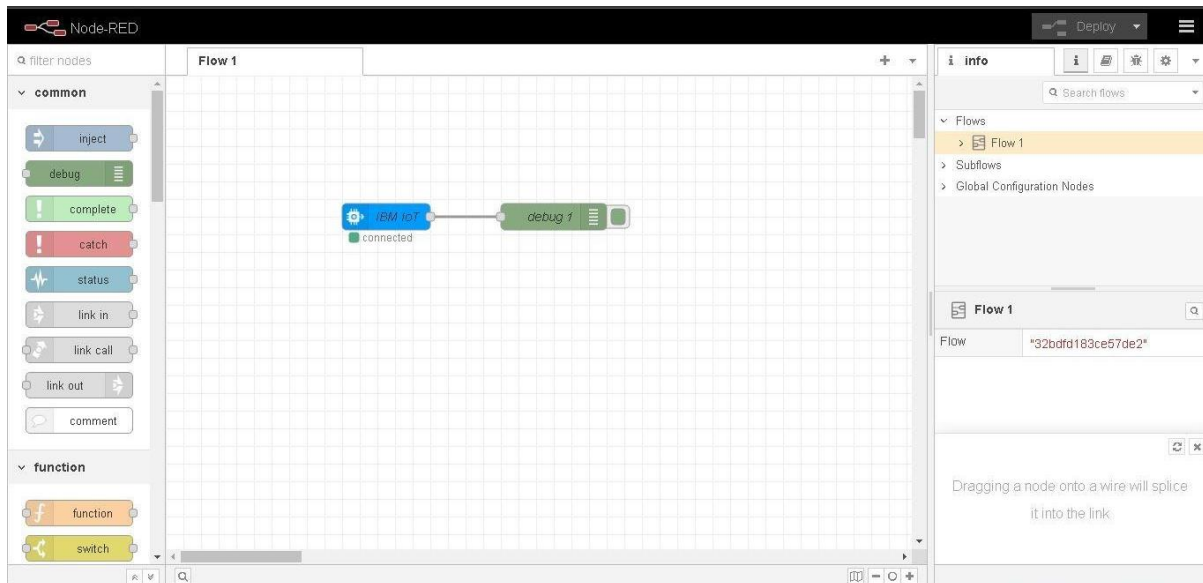
### STEP 3: Generating API key and Authentication token.



### STEP 4: Edit IbmIoT in node.



**STEP 5:** Connect Ibmiot in and debug 1 and deploy.



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

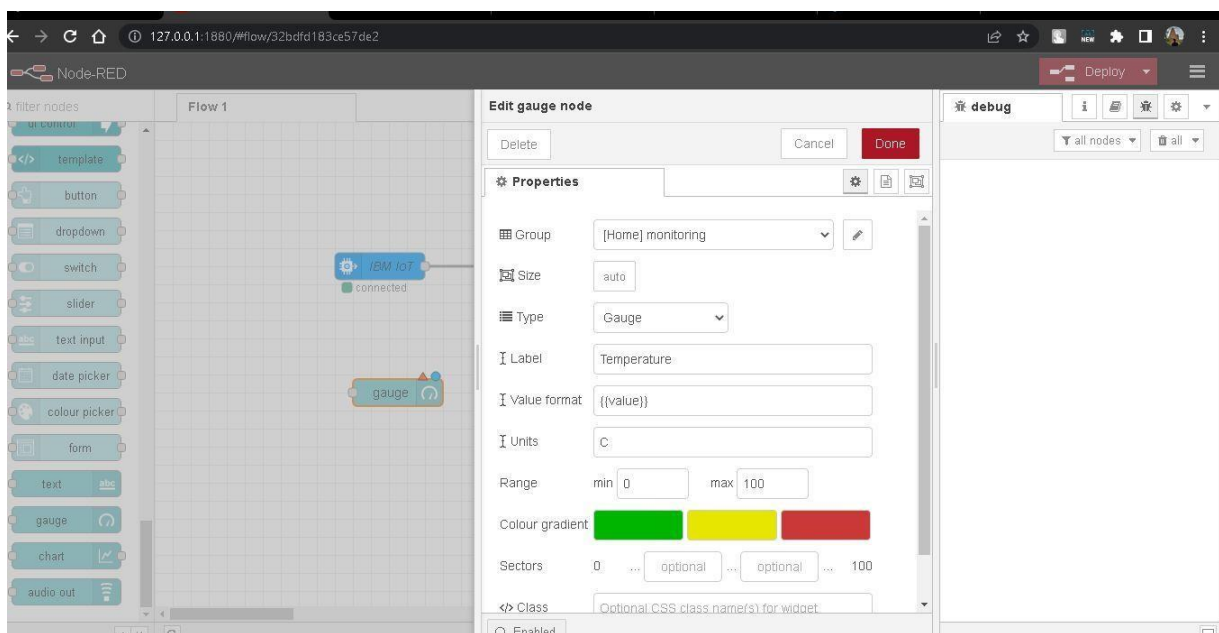


Fig 1

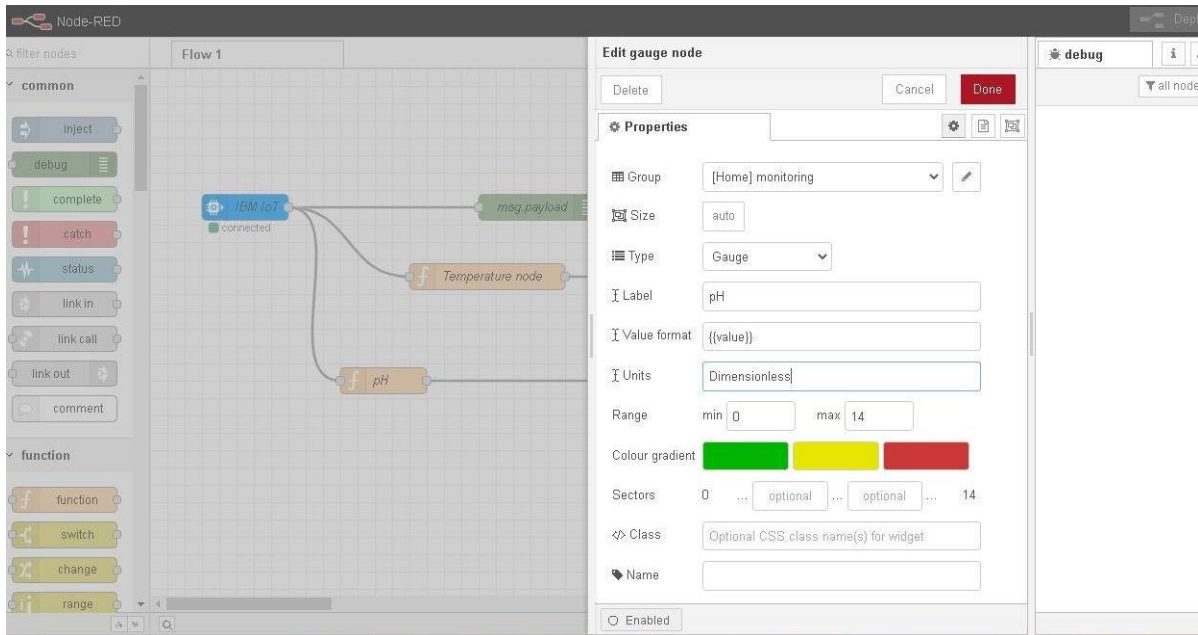


Fig 2

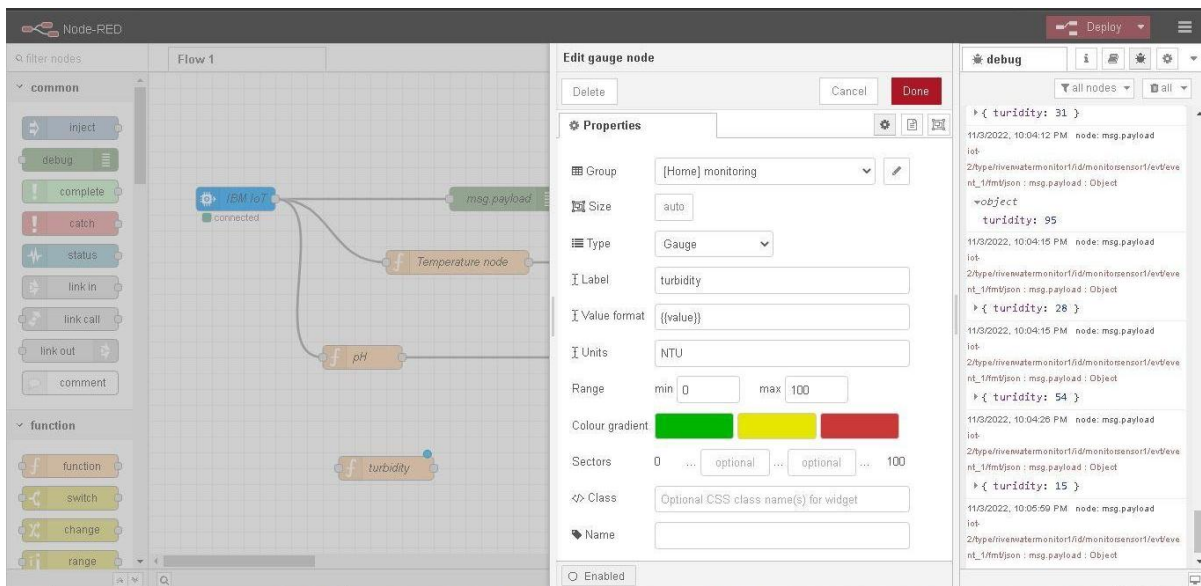


Fig 3

## STEP 7: Simulated program to get the random values.

The screenshot shows the IBM Watson IoT Platform interface. On the left, a sidebar contains navigation icons. The main area displays a table of devices. A modal window is open for the device 'monitorsensor1' under the group 'riverwatermonitor1'. The modal shows the 'Events' configuration for 'event\_1', set to a frequency of '20 x Every Minute'. The 'Payload' section contains a JSON object with random values for 'temperature', 'pH', and 'turbidity'.

IBM Watson IoT Platform

Browse Action Device Types Interface

This table shows a summary of all devices that have been searched on using different criteria. To get started, you can search for devices by name, type, or by using API.

Search by Device ID

Device ID	Status
monitorsensor1	Dis...

Items per page 50 | 1-1 of 1 item

Events 1

Event type name event\_1 Frequency 20 x Every Minute Send

Payload

You can override field values in the event payload that is sent by this device. Specify the override values in the editor window.

```
{
  "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 configured. It starts with an 'IBM IoT' node (connected). The output of this node is split into four parallel paths. Each path consists of a function node (labeled 'Temperature node', 'pH', 'turbidity') followed by a corresponding output node (labeled 'Temperature', 'Humidity', 'turbidity'). The 'debug' tab on the right shows the messages being processed, including the 'turbidity' values.

Node-RED

Flow 1

common

- inject
- debug
- complete
- catch
- status
- link in
- link call
- link out
- comment

function

- function
- switch
- change
- range

IBM IoT (connected)

msg.payload

Temperature node

Temperature

pH

Humidity

turbidity

turbidity

debug

all nodes

```
{ turbidity: 31 }
```

11/3/2022, 10:04:12 PM node: msg.payload  
iot-2/type/riverwatermonitor1/d/monitorsensor1/ev/veve  
nt\_1/mtdjson: msg.payload: Object  
object  
turbidity: 95

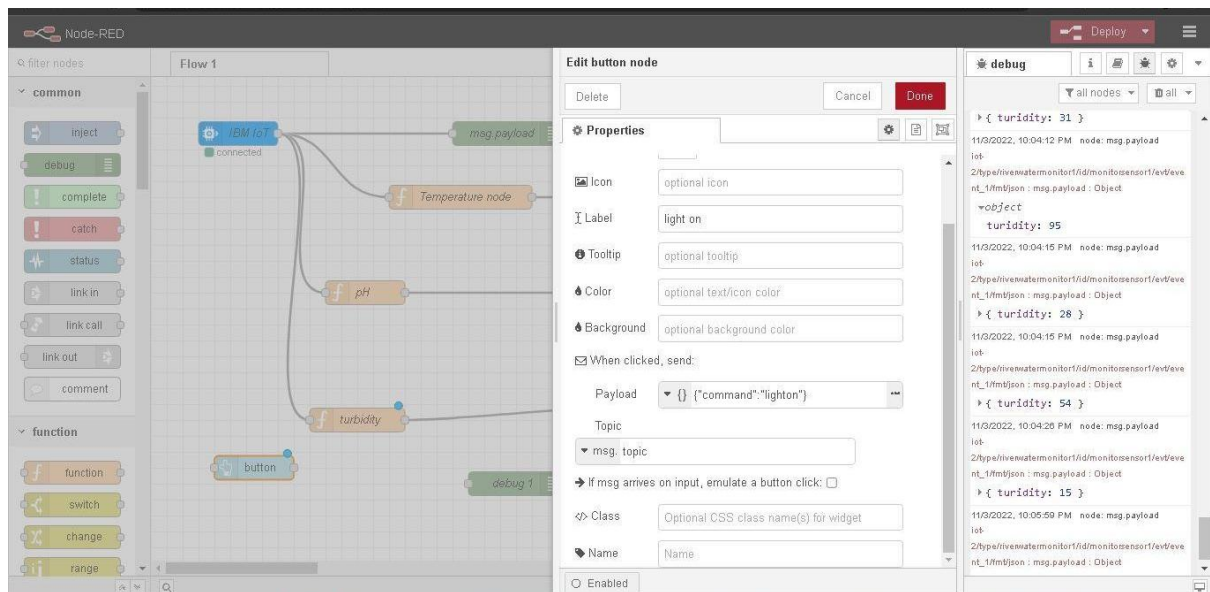
11/3/2022, 10:04:15 PM node: msg.payload  
iot-2/type/riverwatermonitor1/d/monitorsensor1/ev/veve  
nt\_1/mtdjson: msg.payload: Object  
{ turbidity: 28 }

11/3/2022, 10:04:15 PM node: msg.payload  
iot-2/type/riverwatermonitor1/d/monitorsensor1/ev/veve  
nt\_1/mtdjson: msg.payload: Object  
{ turbidity: 54 }

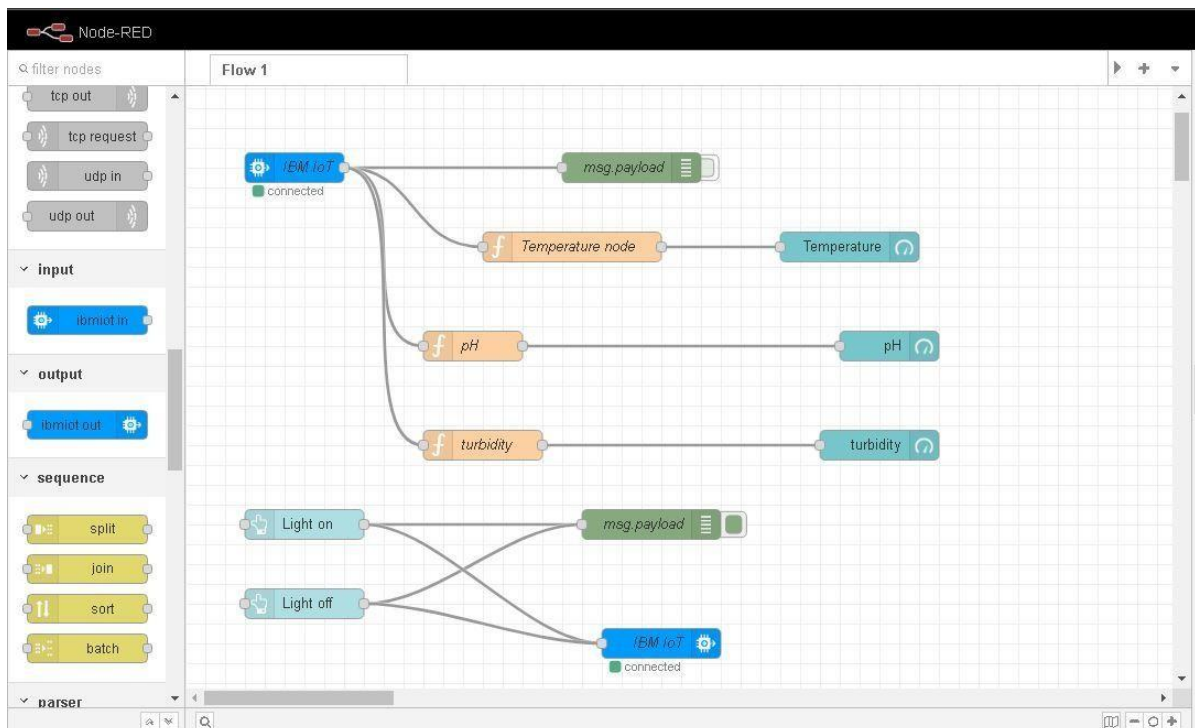
11/3/2022, 10:04:20 PM node: msg.payload  
iot-2/type/riverwatermonitor1/d/monitorsensor1/ev/veve  
nt\_1/mtdjson: msg.payload: Object  
{ turbidity: 15 }

11/3/2022, 10:05:59 PM node: msg.payload  
iot-2/type/riverwatermonitor1/d/monitorsensor1/ev/veve  
nt\_1/mtdjson: msg.payload: Object

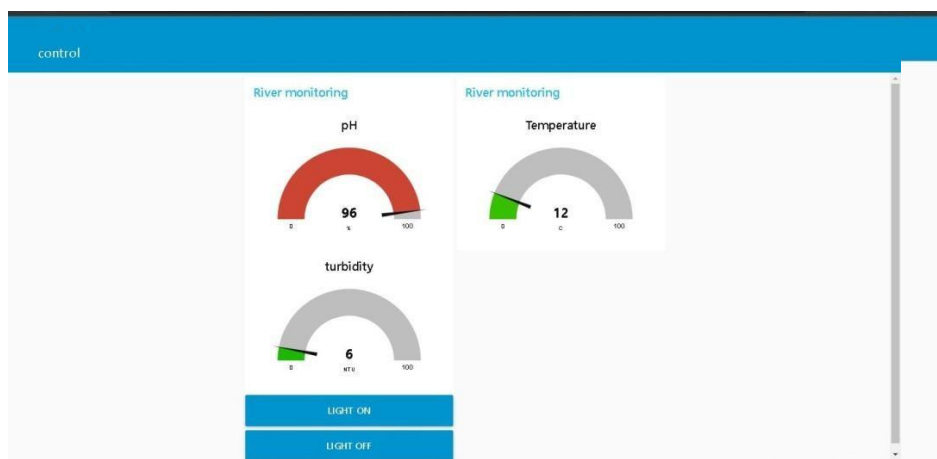
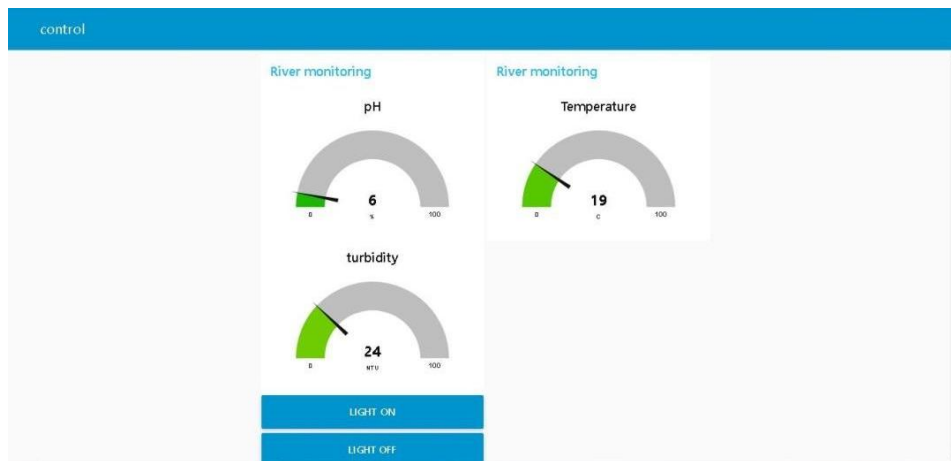
## STEP 9: Edit button mode [light ON and light OFF].



## STEP 10: Entire flow diagram in Node-RED.

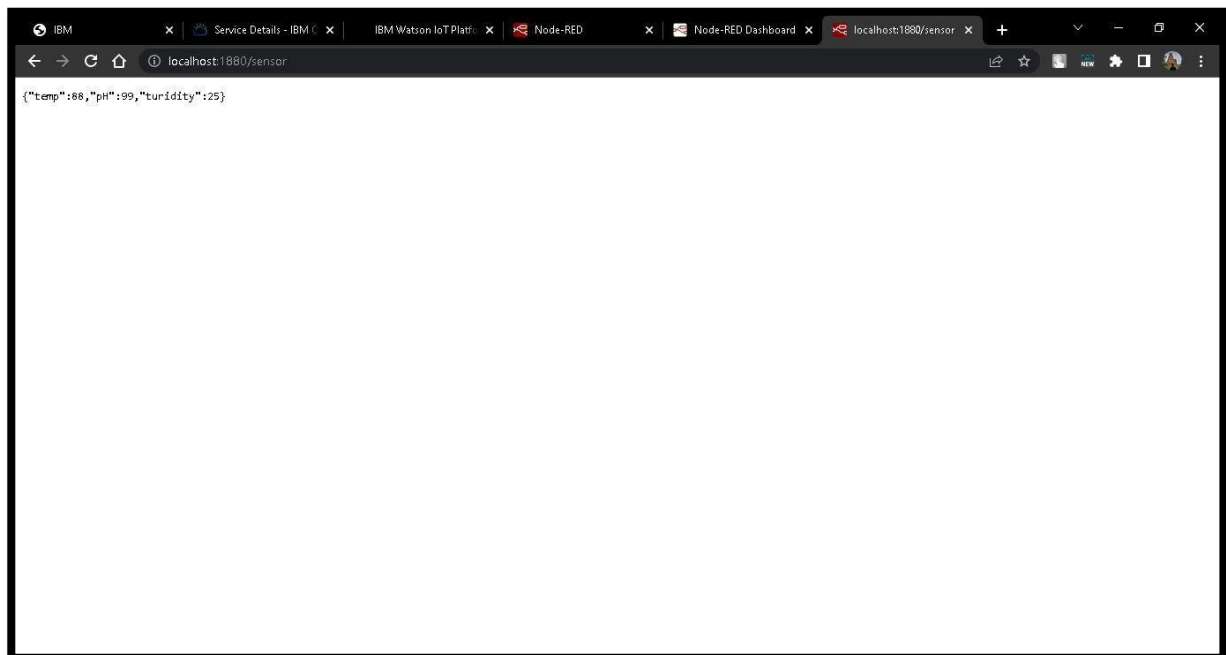
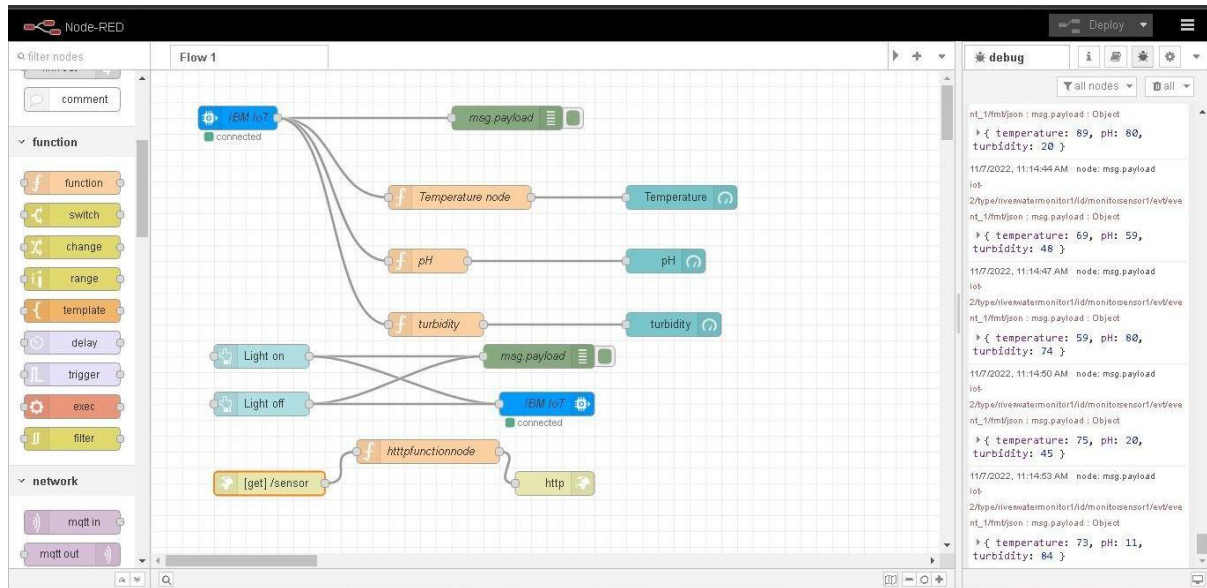


## STEP 11: Generate the output from recent events.





## 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.

