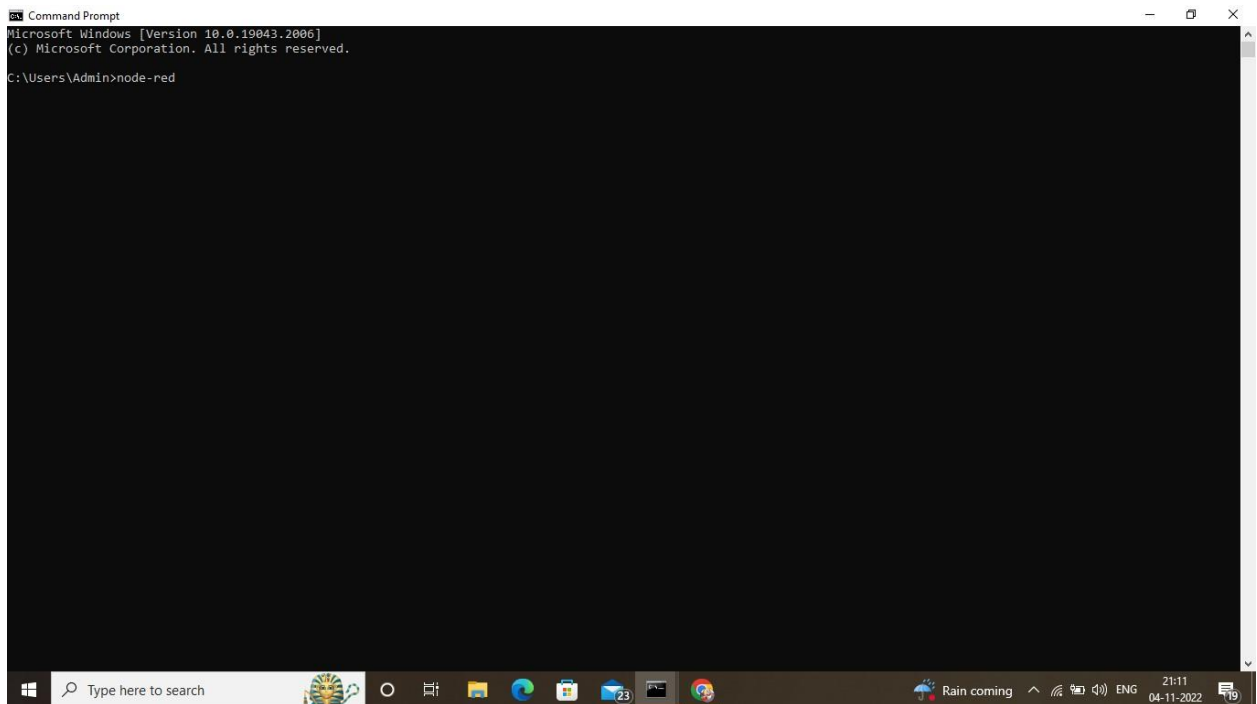


Project Development Phase

Delivery of Sprint 2

DATE	05 NOVEMBER 2022
TEAM ID	PNT2022TMID28798
PROJECT NAME	GAS LEAKAGE DETECTION AND ALERTING
MAXIMUM MARKS	20

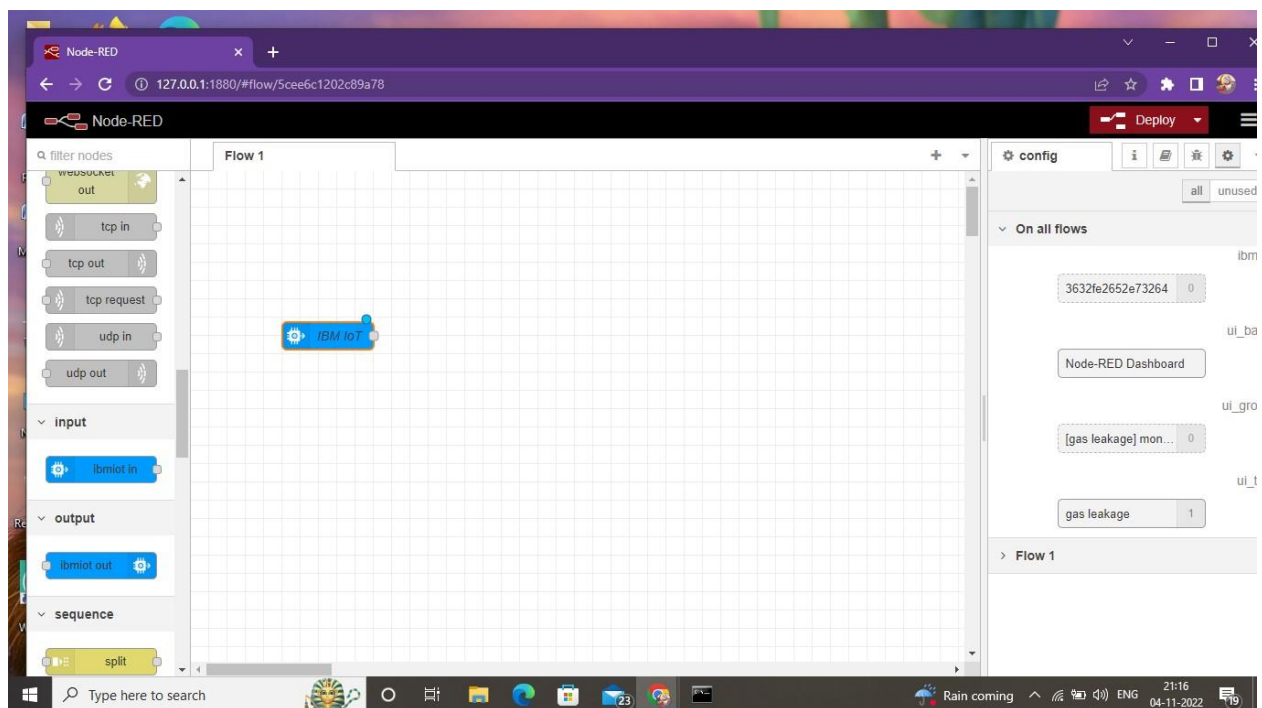
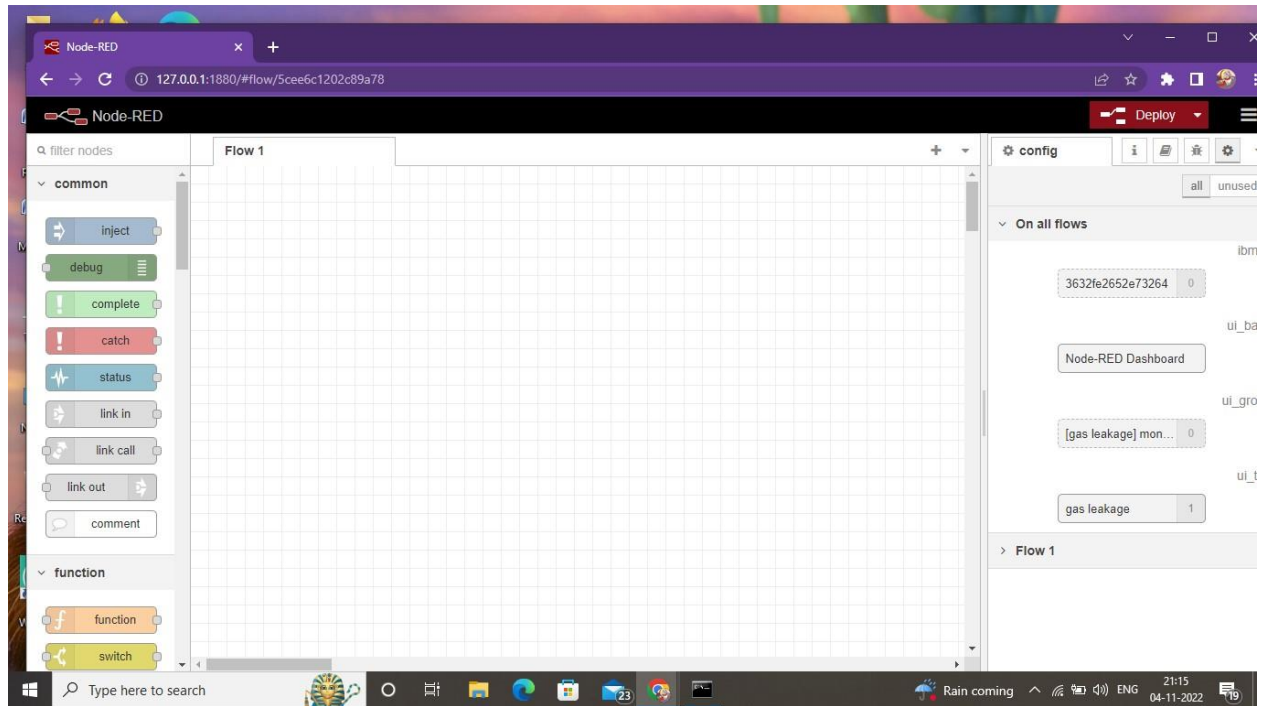
Step1: Install node red and open node red in command prompt



```
Command Prompt
Microsoft Windows [Version 10.0.19043.2006]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Admin>node-red
```

Step 2: Select IBM IoT input in node



Step 3: In IBM Watson platform, go to apps

Service Details - IBM Cloud | IBM Watson IoT Platform

6a4pz2.internetofthings.ibmcloud.com/dashboard/apps/browse

410719106050@smartinternz.com
ID: 6a4pz2

IBM Watson IoT Platform

Browse | IBM Cloud Apps | [+ Generate API Key](#)

This table shows a summary of the API keys that have been added for the organization. It can be filtered, organized, and search on using different criteria. To get started, you can add API keys by clicking Generate API Key, or by using the API. For more information about adding API keys, see [API key connection](#).

<input type="checkbox"/>	Key ↕	Description ↕	Role ↕	Expires ↕	
3 results					
<input type="checkbox"/>	a-6a4pz2-0v39erzrn5	API Key for the device simu...	Standard Application	-	
<input type="checkbox"/>	a-6a4pz2-6gelc7u1gi	node RED	Standard Application	-	
<input type="checkbox"/>	a-6a4pz2-nyev1qklau	API Key for the device simu...	Standard Application	-	

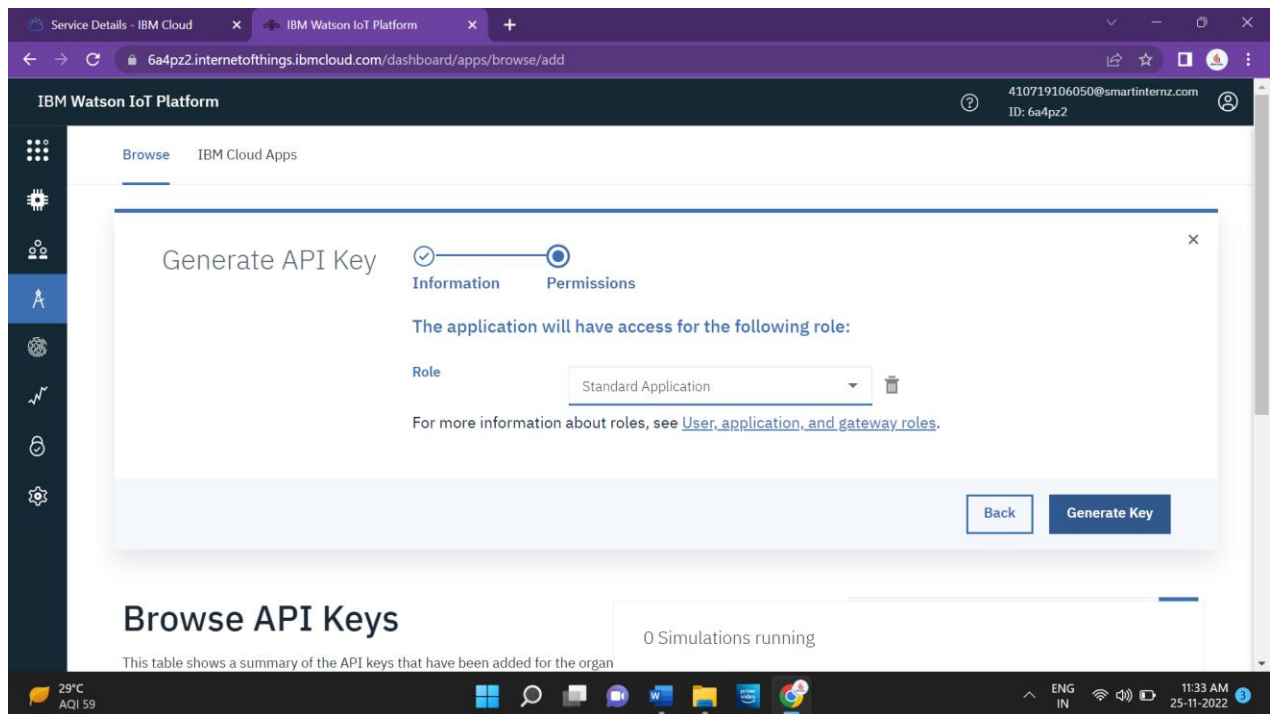
0 Simulations running

29°C
AQI 59

ENG
IN

11:32 AM
25-11-2022

Step 4: Click on generate API keys



Step 5: Generated API key Details

The screenshot displays the IBM Watson IoT Platform dashboard. The browser address bar shows the URL `6a4pz2.internetofthings.ibmcloud.com/dashboard/apps/browse/add`. The page title is "IBM Watson IoT Platform". The user is logged in as `410719106050@smartinternz.com` with ID `6a4pz2`.

The main content area shows a success message: "The API key has been added." Below this, a warning states: "Authentication tokens are non-recoverable. If you misplace this token, you will need to re-register the API key to generate a new authentication token."

The "Generated Details" section lists the following information:

Generated Details	API Key Information
API Key	Description
Authentication Token	Role
	Expires

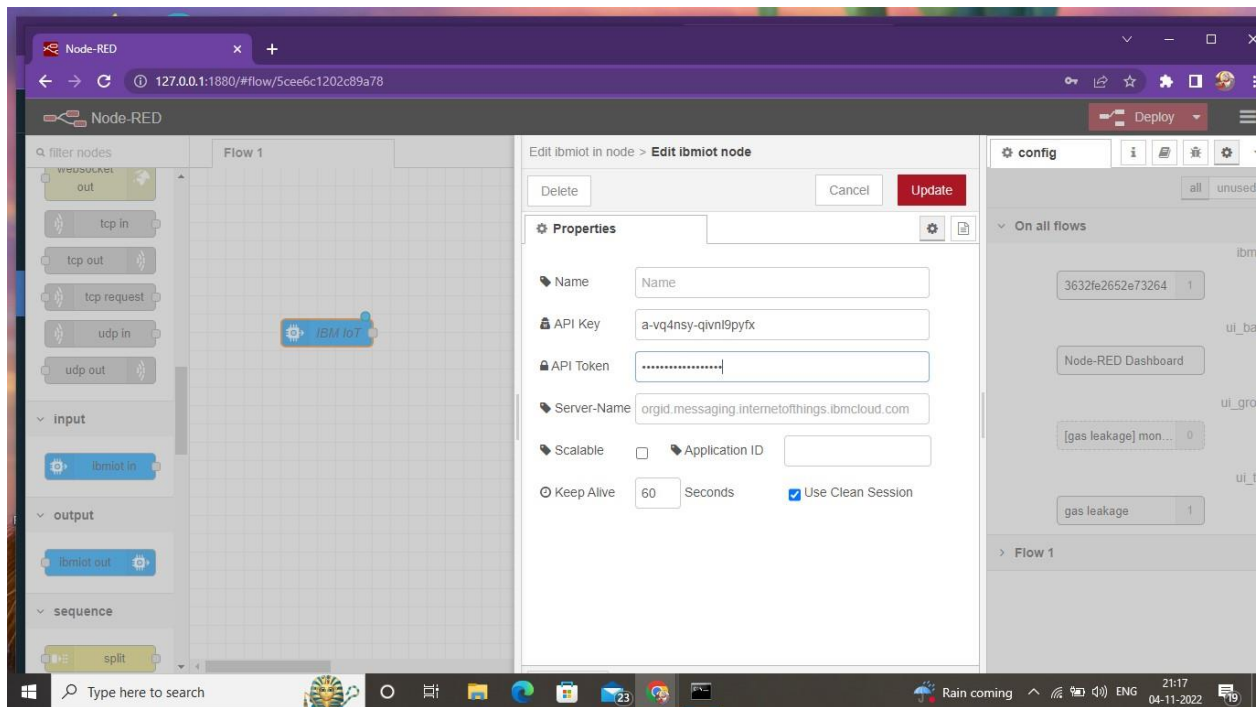
The "API Key Information" section shows the following details:

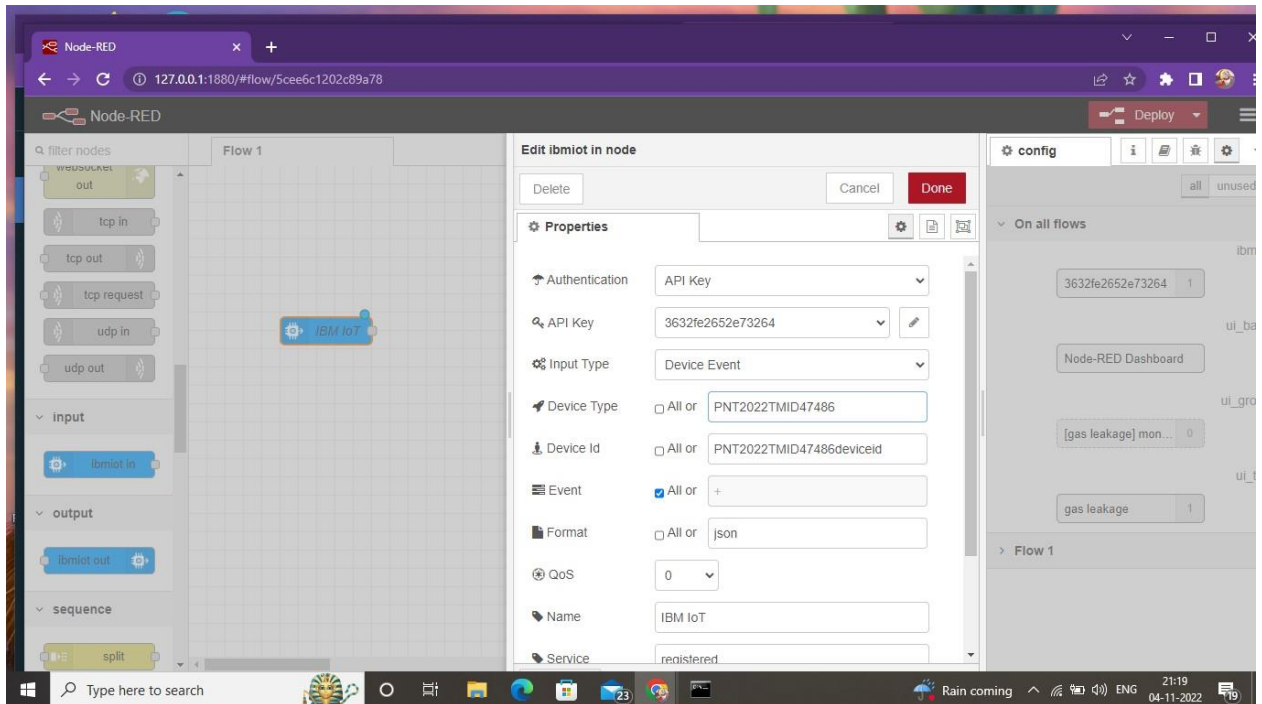
API Key Information
Description
Role
Expires

A warning icon and text are present: "Make a note of the generated authentication token. Lost authentication tokens cannot be recovered. If you lose the token, you must reregister the API to generate a new token."

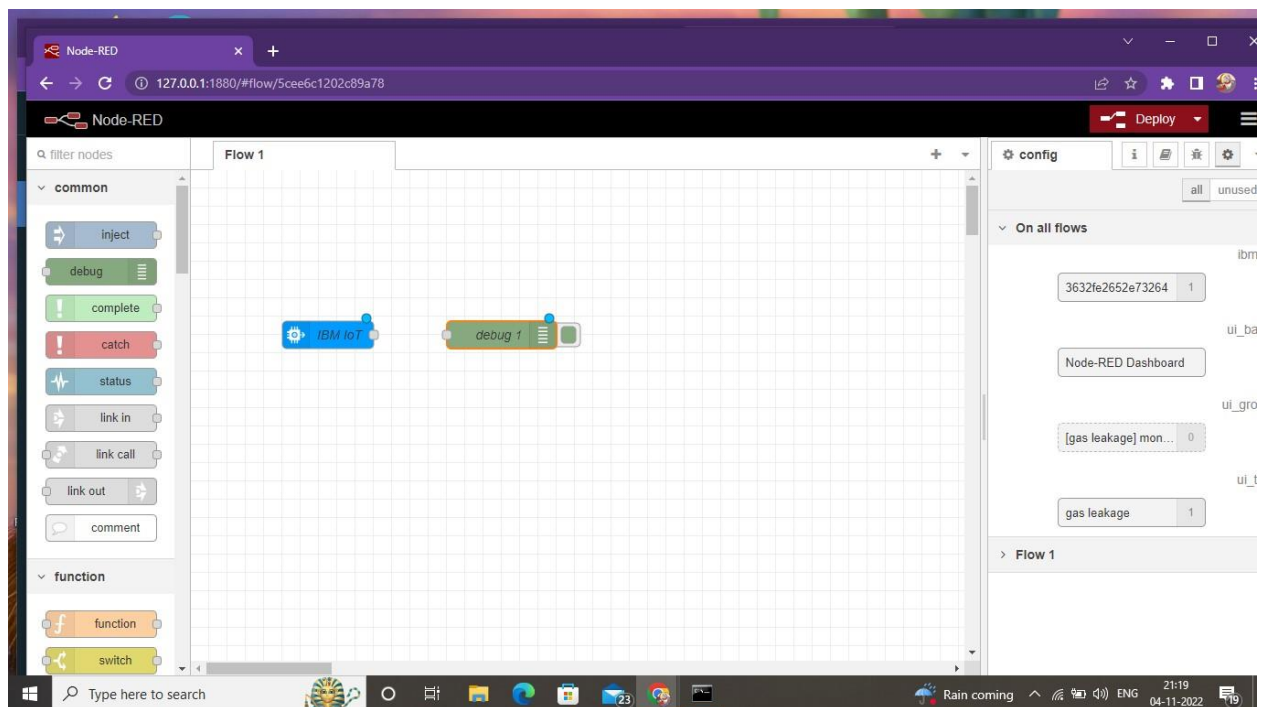
The bottom status bar shows "0 Simulations running". The system tray at the bottom indicates a temperature of 29°C, AQI of 59, and the time is 11:33 AM on 25-11-2022.

Step6: Copy and paste the generated API key in node red

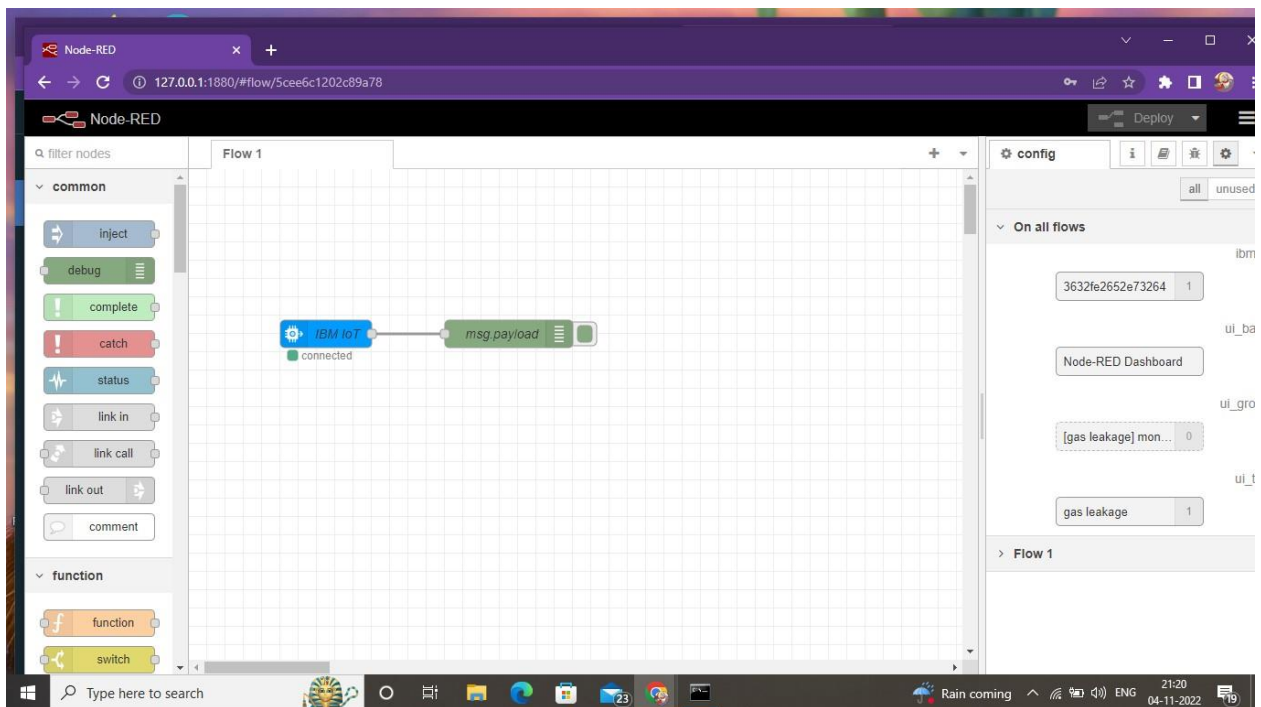
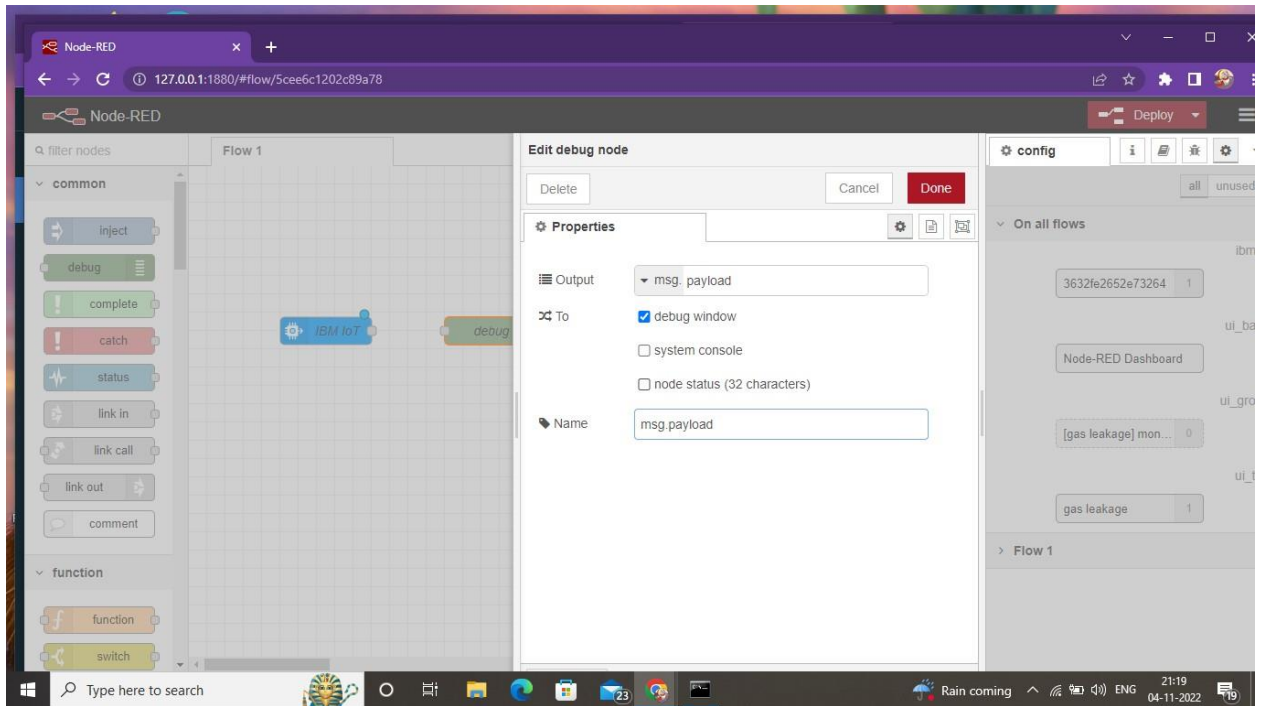




Step 7:after completing all the details click in done button

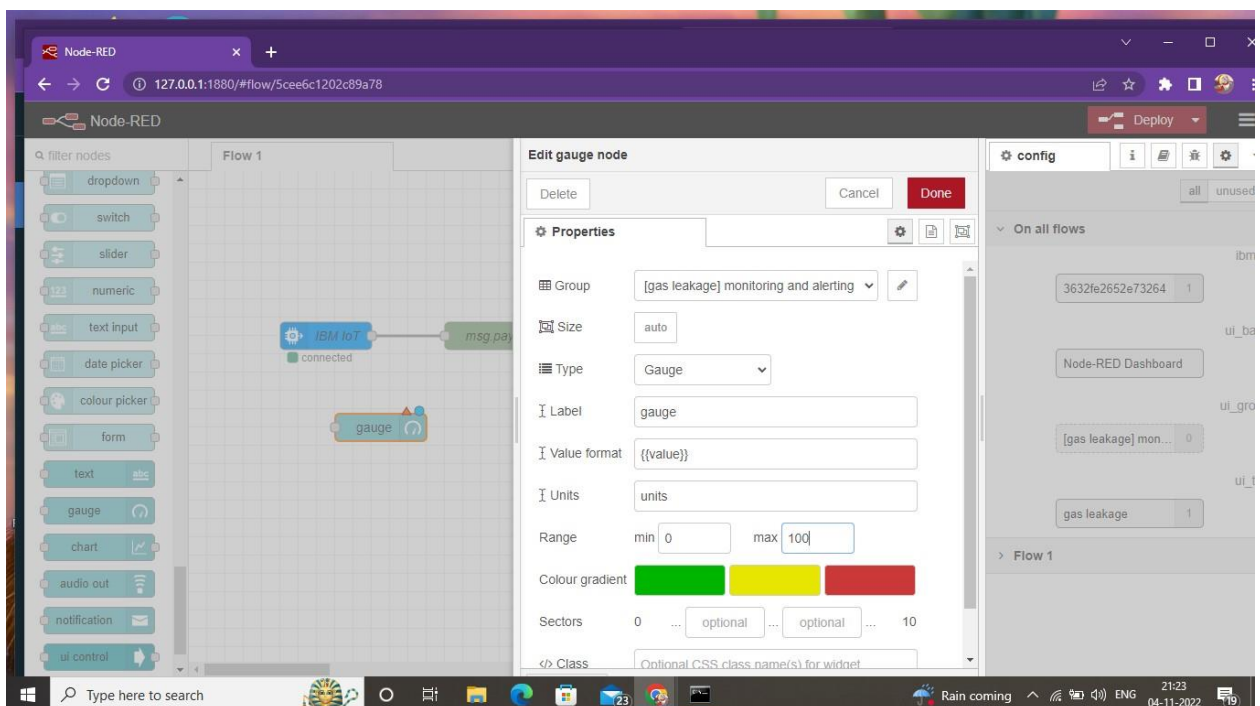
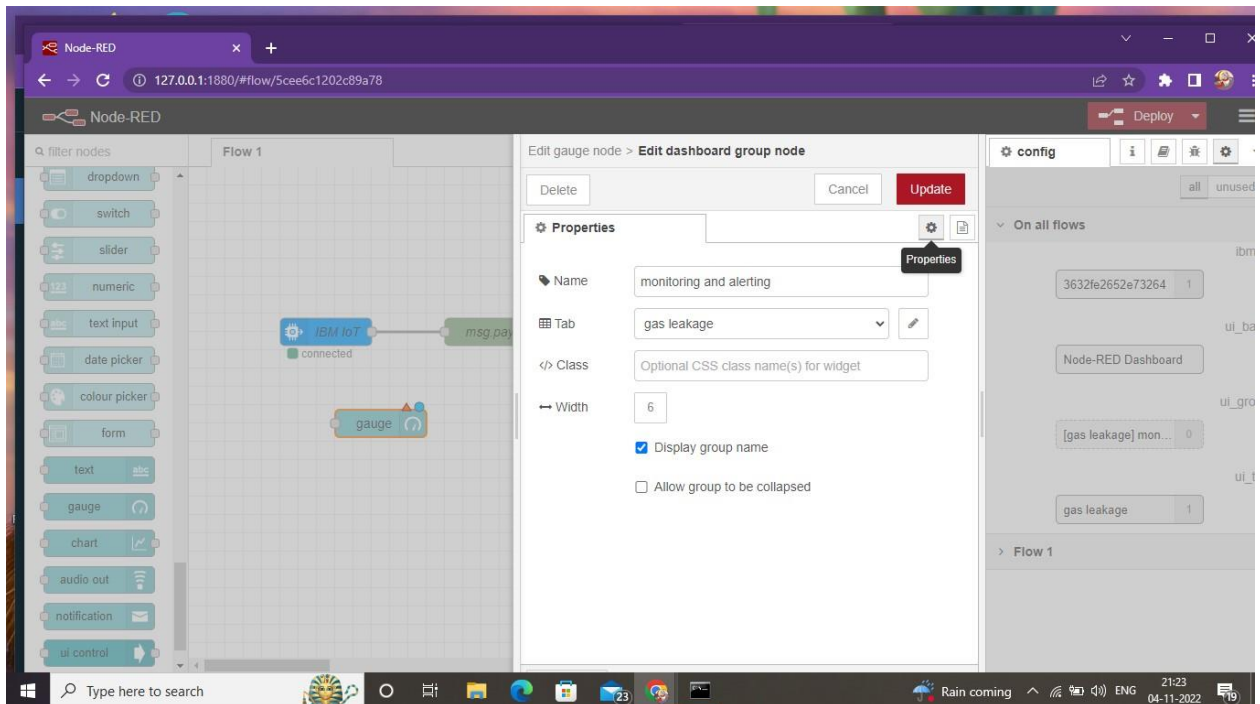


Step 8: add debug to the IBM iot and rename as msg.payload and click on done.



Step 10:

Click gauge from the dashboard node and fill the details



Add functions to the gauge

Step 11:

The screenshot displays the Node-RED web interface in a browser window. The address bar shows the URL `127.0.0.1:1880/#flow/5cee6c1202c89a78`. The interface includes a left sidebar with a 'filter nodes' search bar and a list of nodes categorized under 'function' and 'network'. The main workspace, titled 'Flow 1', contains a flow diagram with the following nodes: an 'IBM IoT' node (blue) with a 'connected' status indicator, a 'msg.payload' node (green), a 'gauge' node (teal), and a 'function 1' node (orange). The flow is configured as follows: the 'IBM IoT' node connects to the 'msg.payload' node, which then connects to the 'gauge' node. The 'gauge' node connects to the 'function 1' node. The right sidebar shows the 'config' tab with a 'Deploy' button and a list of nodes under 'On all flows'. The list includes: '3632fe2652e73264' (1), 'Node-RED Dashboard', '[gas leakage] mon...' (1), and 'gas leakage' (1). The bottom status bar shows the system clock as 21:24 on 04-11-2022, along with weather information 'Rain coming' and system icons.

Step 12:

Check the values from debug messages

The screenshot shows the Node-RED web interface in a browser. The main workspace displays a flow named 'Flow 1' with the following nodes: an 'IBM IoT' node (blue), a 'msg.payload' node (green), a 'gauge' node (blue), and a 'function 1' node (orange). The 'IBM IoT' node is connected to 'msg.payload', which is connected to 'gauge'. The 'gauge' node is connected to 'function 1'. The left sidebar shows the 'function' and 'network' node categories. The right sidebar shows the 'Debug messages' panel, which displays a list of messages received from the 'node: debug 1' node. The messages are as follows:

- 11/4/2022, 9:15:23 PM node: debug 1
iot-
2/type/PNT2022TMID47486/id/PNT2022TMID47486dev
ceid/evt/event_1/fmt/json : msg.payload : Object
↳ { temperaturer: 52 }
- 11/4/2022, 9:15:26 PM node: debug 1
iot-
2/type/PNT2022TMID47486/id/PNT2022TMID47486dev
ceid/evt/event_1/fmt/json : msg.payload : Object
↳ { temperaturer: 46 }
- 11/4/2022, 9:15:29 PM node: debug 1
iot-
2/type/PNT2022TMID47486/id/PNT2022TMID47486dev
ceid/evt/event_1/fmt/json : msg.payload : Object
↳ { temperaturer: 12 }
- 11/4/2022, 9:20:09 PM node: msg.payload
iot-
2/type/PNT2022TMID47486/id/PNT2022TMID47486dev
ceid/evt/event_1/fmt/json : msg.payload : Object
↳ { temperaturer: 67 }
- 11/4/2022, 9:20:12 PM node: msg.payload
iot-
2/type/PNT2022TMID47486/id/PNT2022TMID47486dev

Step 12: Edit function node

The screenshot shows the Node-RED web interface in a browser. The left sidebar contains a 'function' category with various nodes. The main workspace, labeled 'Flow 1', contains an 'IBM IoT' node (blue) connected to a 'msg.payload' node (green). Below these, there is a 'gauge' node (blue) and a 'function 1' node (orange). The right sidebar shows the 'debug' console with a list of messages. The first message is from 'node: debug 1' at 9:15:29 PM, showing a temperature of 12. The second message is from 'node: msg.payload' at 9:20:09 PM, showing a temperature of 67. The third message is from 'node: msg.payload' at 9:20:12 PM, showing a temperature of 44. The fourth message is from 'node: msg.payload' at 9:20:15 PM, showing a temperature of 98. The fifth message is from 'node: msg.payload' at 9:20:18 PM, showing a temperature of 98.

The screenshot shows the Node-RED web interface with the 'function 1' node selected and its configuration panel open. The configuration panel shows the 'function' tab with a text area containing the following code:

```
return msg.payload.temperature;
```

The right sidebar shows the 'debug' console with a list of messages. The first message is from 'node: msg.payload' at 9:20:15 PM, showing a temperature of 98. The second message is from 'node: msg.payload' at 9:20:18 PM, showing a temperature of 73. The third message is from 'node: msg.payload' at 9:20:21 PM, showing a temperature of 78. The fourth message is from 'node: msg.payload' at 9:20:24 PM, showing a temperature of 44. The fifth message is from 'node: msg.payload' at 9:20:27 PM, showing a temperature of 98.

Node-RED interface showing a flow with an IBM IoT node connected to a function node named "temperature node". The function node is configured to process messages on the "On Message" trigger.

Edit function node

Properties

Name: temperature node

Setup On Start On Message On Stop

```
1 msg.payload = msg.payload.temperature
2 return msg;
```

debug

11/4/2022, 9:32:15 PM node: msg.payload
lol-
2/type/PNT2022TMID47486/Id/PNT2022TMID47486devi
ceid/ev/event_1/fmt/json : msg.payload : Object
↳ { temperature: 61 }

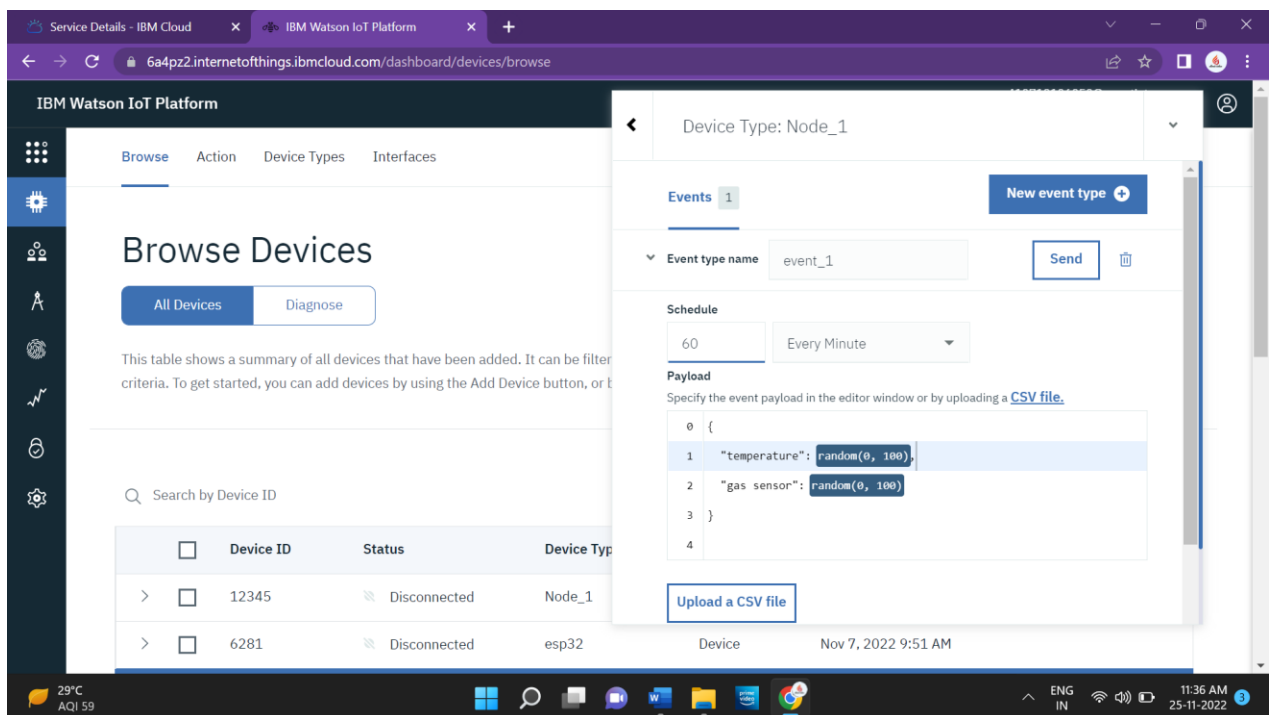
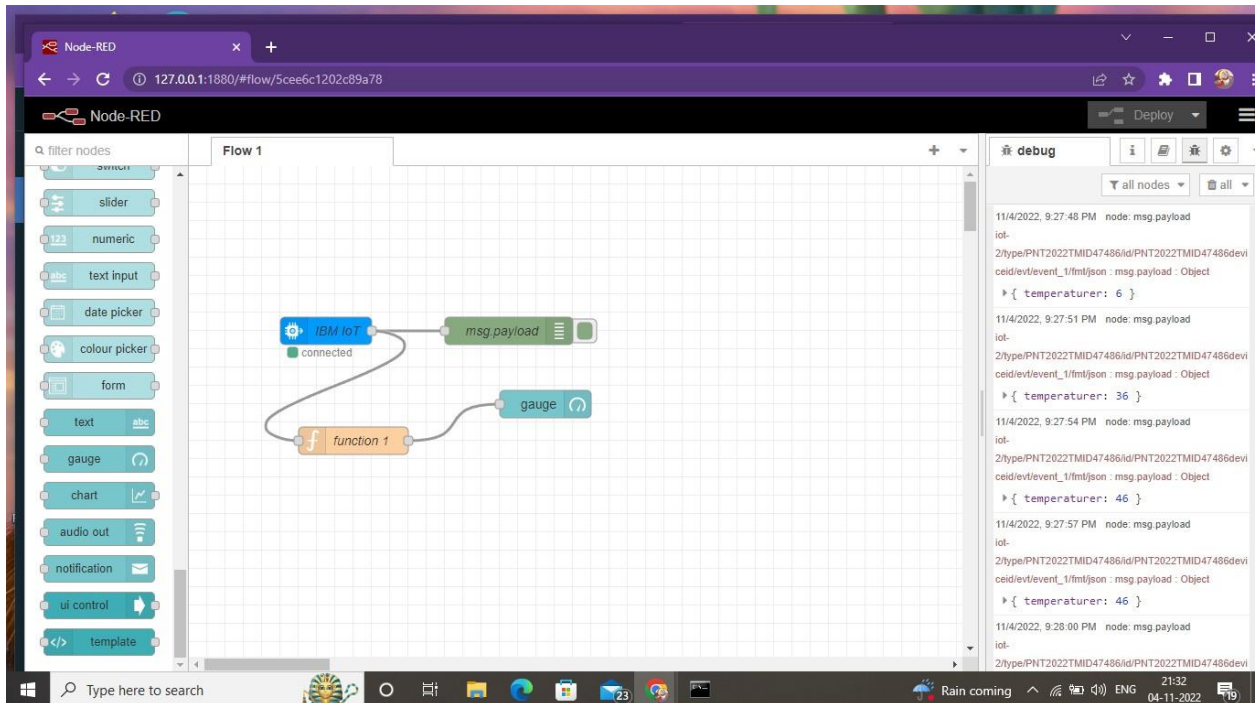
11/4/2022, 9:32:18 PM node: msg.payload
lol-
2/type/PNT2022TMID47486/Id/PNT2022TMID47486devi
ceid/ev/event_1/fmt/json : msg.payload : Object
↳ { temperature: 47 }

11/4/2022, 9:32:21 PM node: msg.payload
lol-
2/type/PNT2022TMID47486/Id/PNT2022TMID47486devi
ceid/ev/event_1/fmt/json : msg.payload : Object
↳ { temperature: 10 }

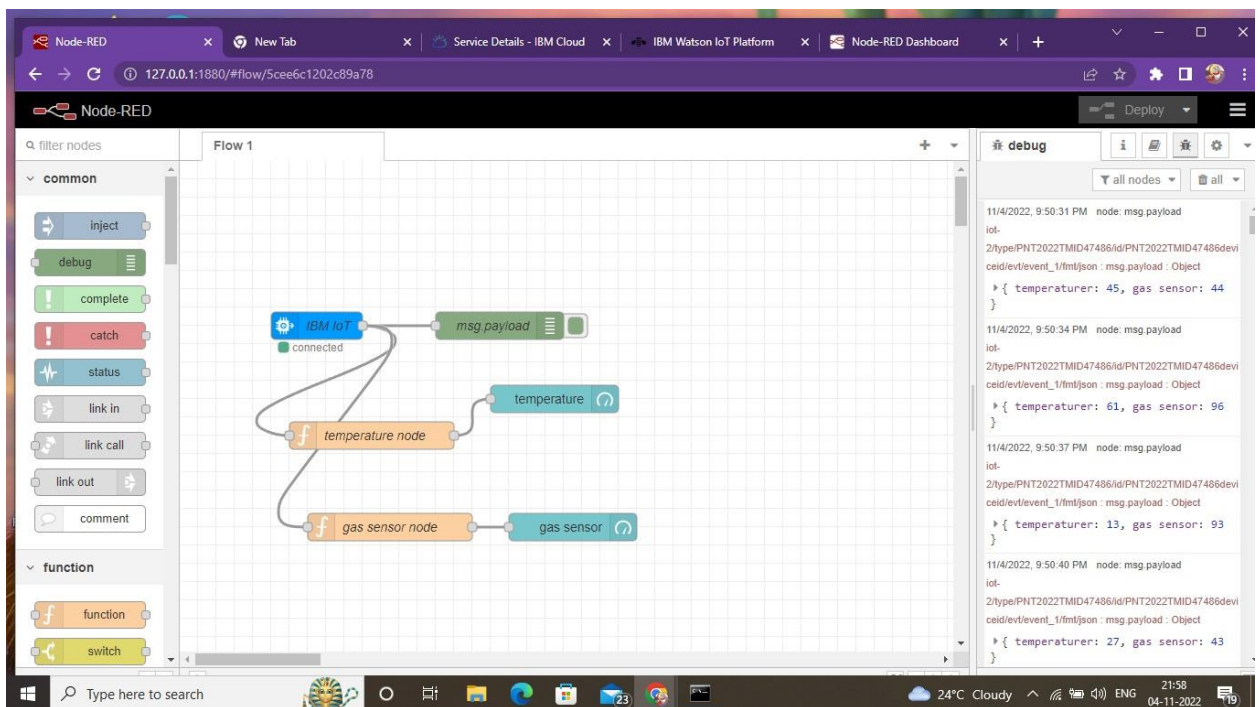
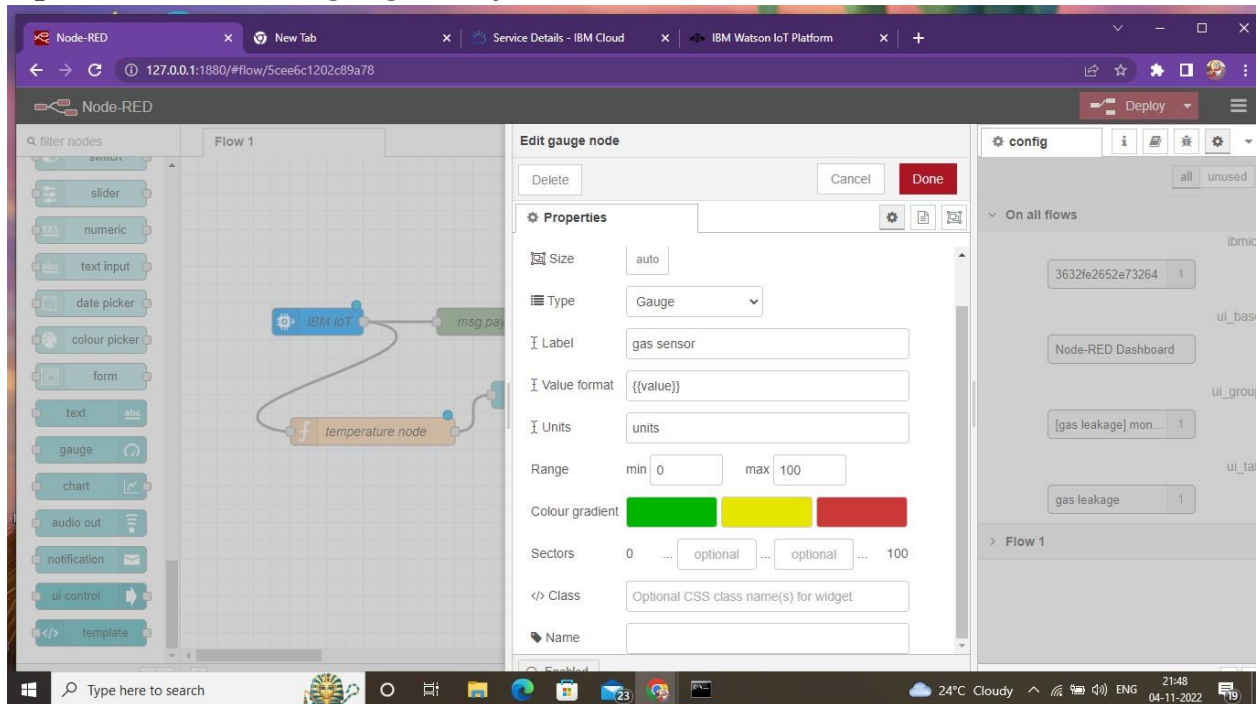
11/4/2022, 9:32:24 PM node: msg.payload
lol-
2/type/PNT2022TMID47486/Id/PNT2022TMID47486devi
ceid/ev/event_1/fmt/json : msg.payload : Object
↳ { temperature: 83 }

11/4/2022, 9:32:27 PM node: msg.payload
lol-
2/type/PNT2022TMID47486/Id/PNT2022TMID47486devi

Step 13: Connect them



Step 14: Add another gauge and functions



Node-RED interface showing a flow named "Flow 1" and a debug console.

Flow 1:

- Input: **IBM IoT** (connected)
- Processing:
 - temperature node** (function node)
 - gas sensor node** (function node)
- Output: **msg.payload** (message node)
- Destination: **temperature** and **gas sensor** (output nodes)

Debug Console:

11/4/2022, 9:50:31 PM node: msg.payload
iot-
2/type/PNT2022TMD47486/id/PNT2022TMD47486devi
ceid/evt/event_1/fmt/json : msg.payload - Object
 { temperaturer: 45, gas sensor: 44
 }

11/4/2022, 9:50:34 PM node: msg.payload
iot-
2/type/PNT2022TMD47486/id/PNT2022TMD47486devi
ceid/evt/event_1/fmt/json : msg.payload - Object
 { temperaturer: 61, gas sensor: 96
 }

11/4/2022, 9:50:37 PM node: msg.payload
iot-
2/type/PNT2022TMD47486/id/PNT2022TMD47486devi
ceid/evt/event_1/fmt/json : msg.payload - Object
 { temperaturer: 13, gas sensor: 93
 }

11/4/2022, 9:50:40 PM node: msg.payload
iot-
2/type/PNT2022TMD47486/id/PNT2022TMD47486devi
ceid/evt/event_1/fmt/json : msg.payload - Object
 { temperaturer: 27, gas sensor: 43
 }

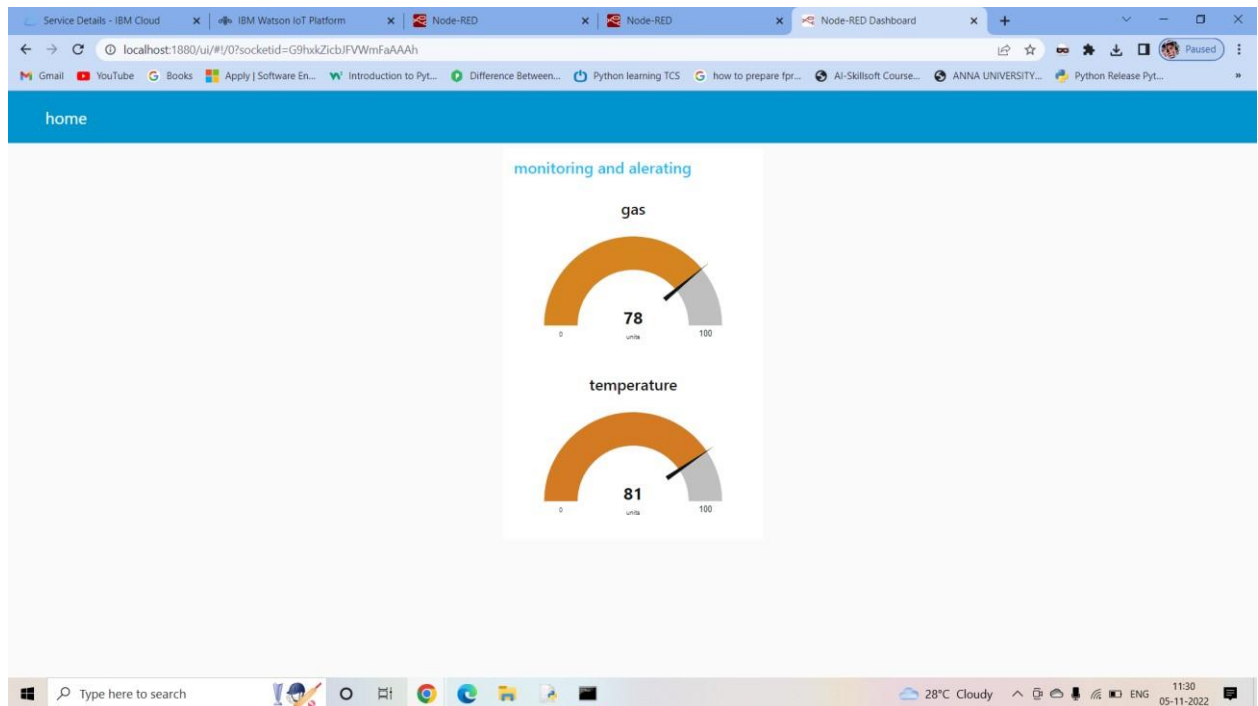
Step15: finally add alarm on and off buttons to IBM iot and debug.step

The screenshot displays the Node-RED web interface in a browser. The top bar shows the URL `127.0.0.1:1880/#flow/27b4c9cc6b1a7d1f` and a "Successfully deployed" status. The interface is divided into several sections:

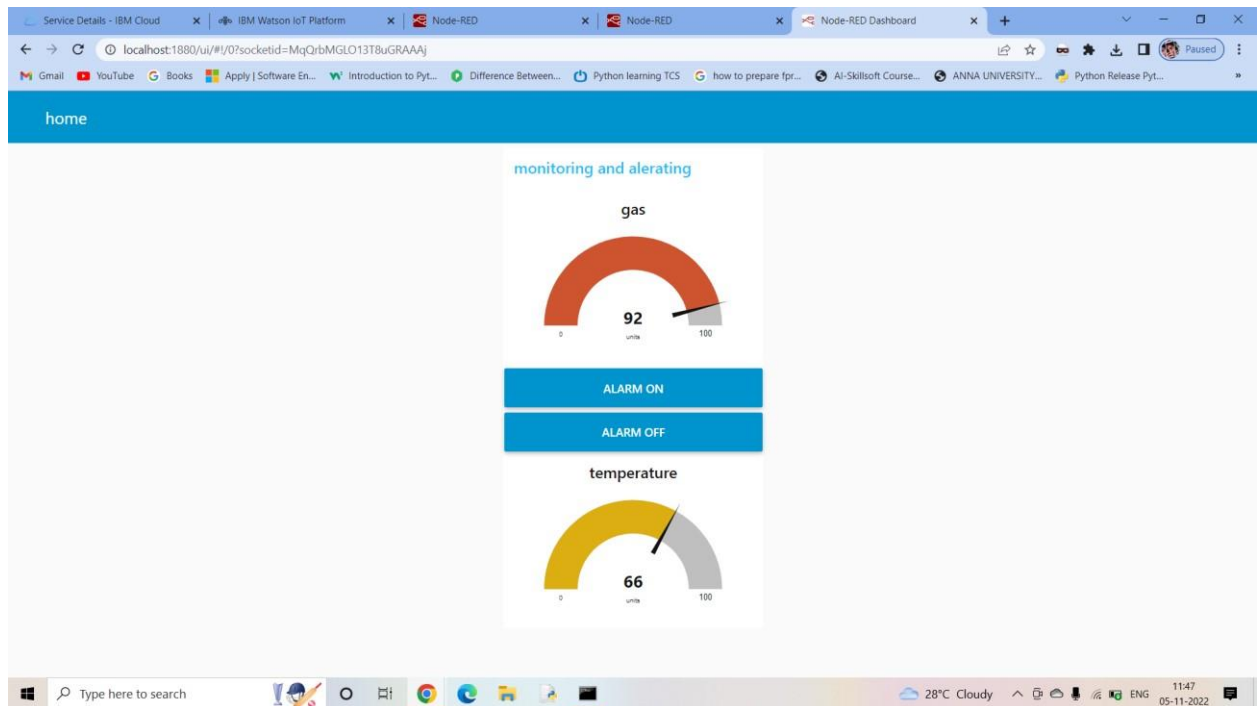
- Left Panel (Nodes):** Contains "common" nodes (inject, debug, complete, catch, status, link in, link call, link out, comment) and "function" nodes (function, switch, change, range).
- Flow 1:** A flow starting with an "IBM IoT" node (connected). It branches into two paths:
 - Top path: "msg.payload" node → "temperature node" (function) → "temperature" output node.
 - Bottom path: "msg.payload" node → "gas node" (function) → "gas" output node.
- Flow 2:** A flow starting with an "IBM IoT" node (connected). It branches into two paths:
 - Top path: "Alarm On" button node → "msg.payload" node.
 - Bottom path: "Alarm Off" button node → "msg.payload" node.
- Right Panel (Debug Console):** Shows a log of messages. The first two messages are from Flow 1:

```
{ temperature: 51, gas: 24 }  
11/5/2022, 11:51:18 AM node: msg.payload  
27b4c9cc6b1a7d1f: msg.payload Object  
27b4c9cc6b1a7d1f: msg.payload Object  
{ temperature: 26, gas: 64 }  
11/5/2022, 11:51:21 AM node: msg.payload  
27b4c9cc6b1a7d1f: msg.payload Object  
27b4c9cc6b1a7d1f: msg.payload Object  
{ temperature: 86, gas: 94 }  
11/5/2022, 11:51:24 AM node: msg.payload  
27b4c9cc6b1a7d1f: msg.payload Object  
27b4c9cc6b1a7d1f: msg.payload Object  
{ temperature: 23, gas: 78 }  
11/5/2022, 11:51:27 AM node: msg.payload  
27b4c9cc6b1a7d1f: msg.payload Object  
27b4c9cc6b1a7d1f: msg.payload Object  
{ temperature: 81, gas: 65 }  
11/5/2022, 11:51:33 AM node: msg.payload  
27b4c9cc6b1a7d1f: msg.payload Object  
27b4c9cc6b1a7d1f: msg.payload Object  
{ temperature: 3, gas: 94 }
```

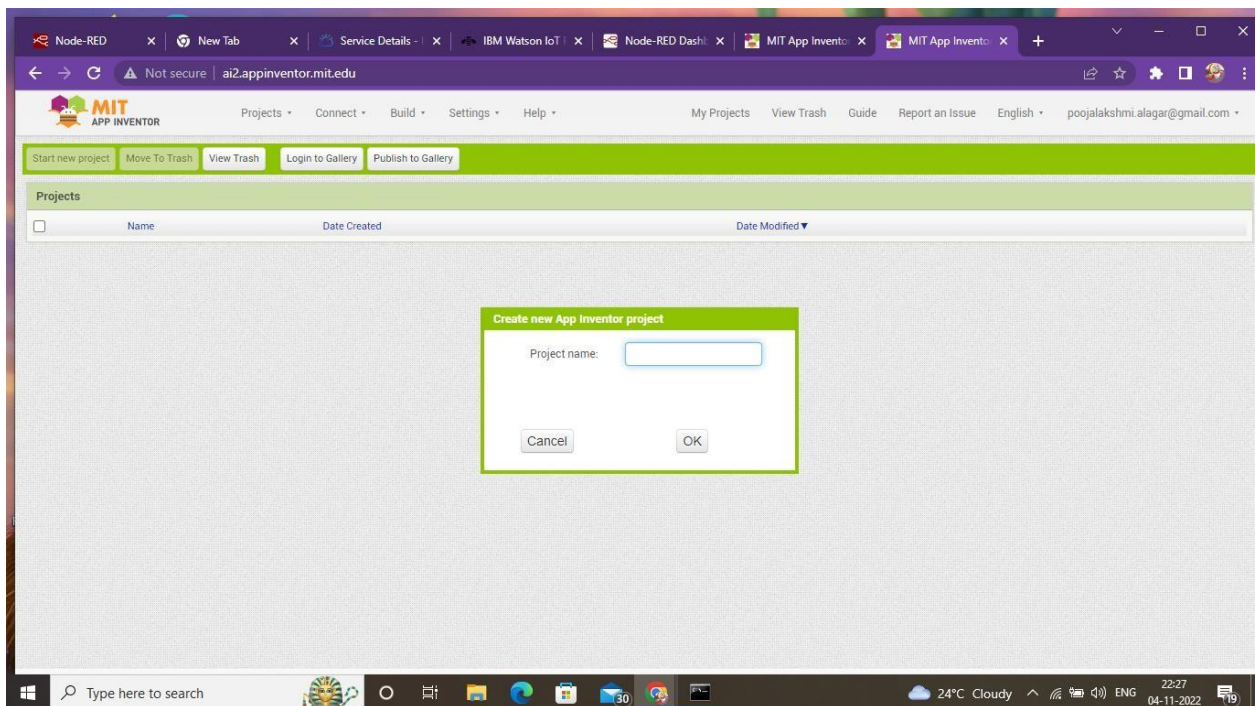
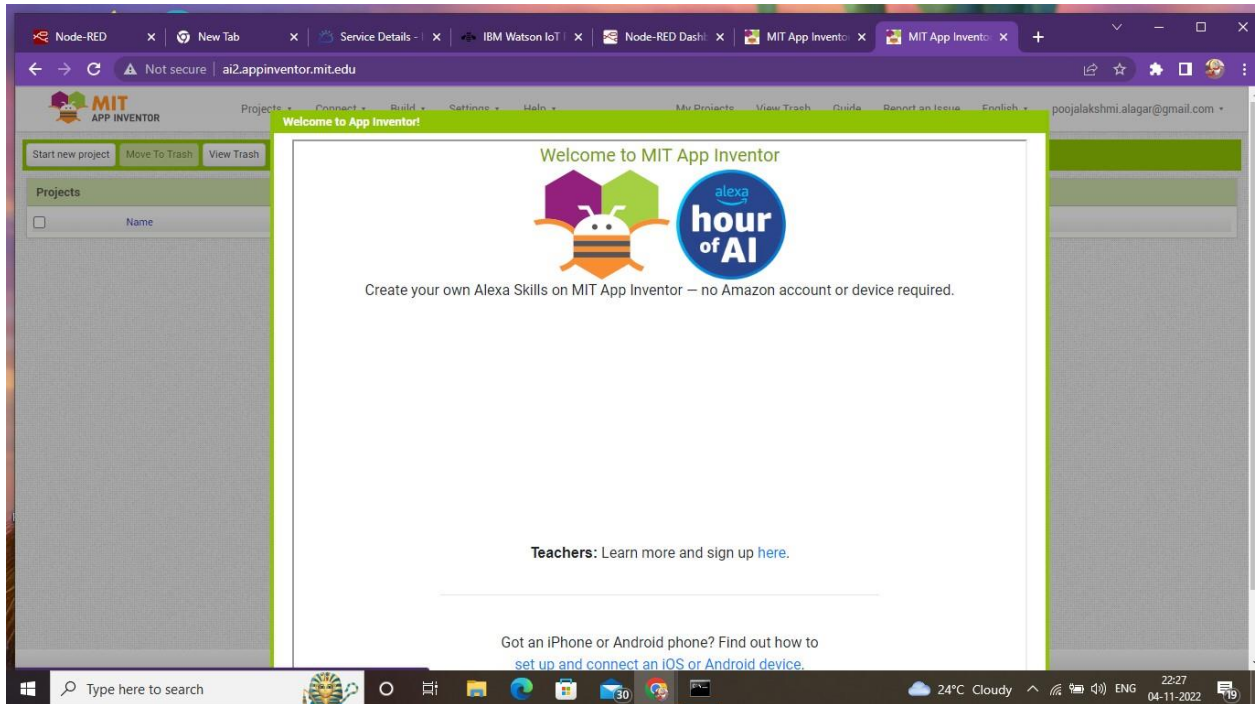
Step16: Output from node red

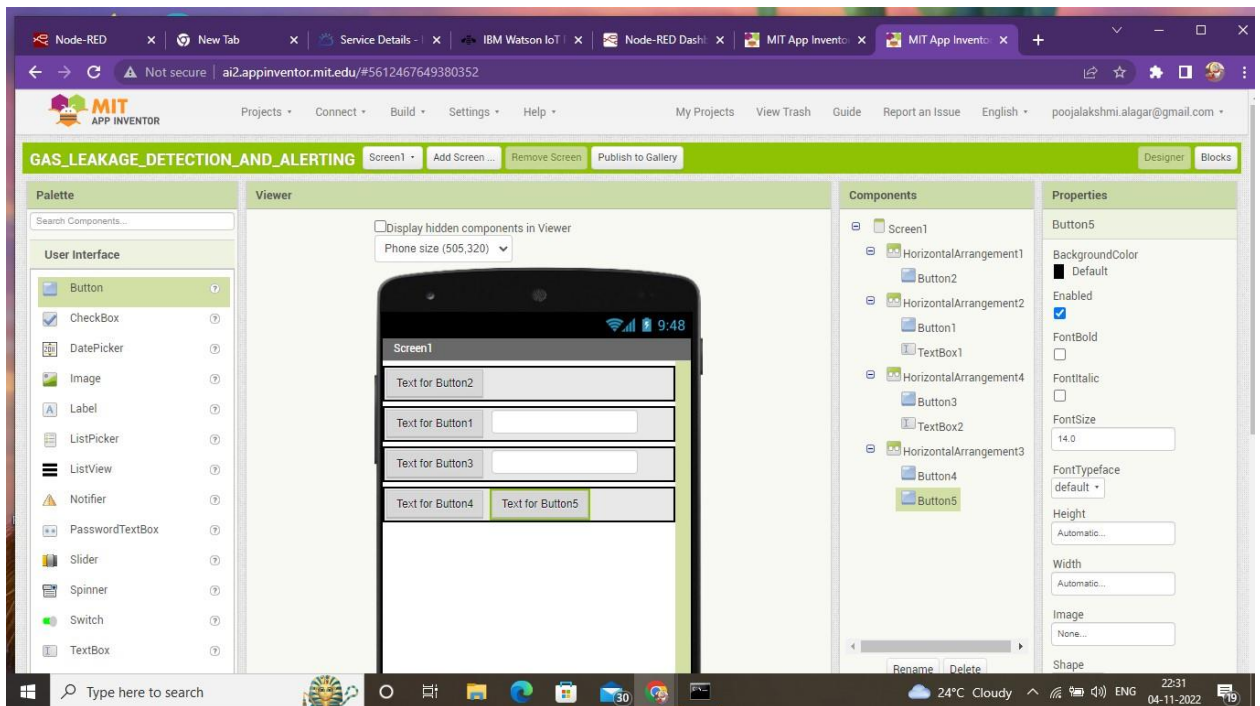
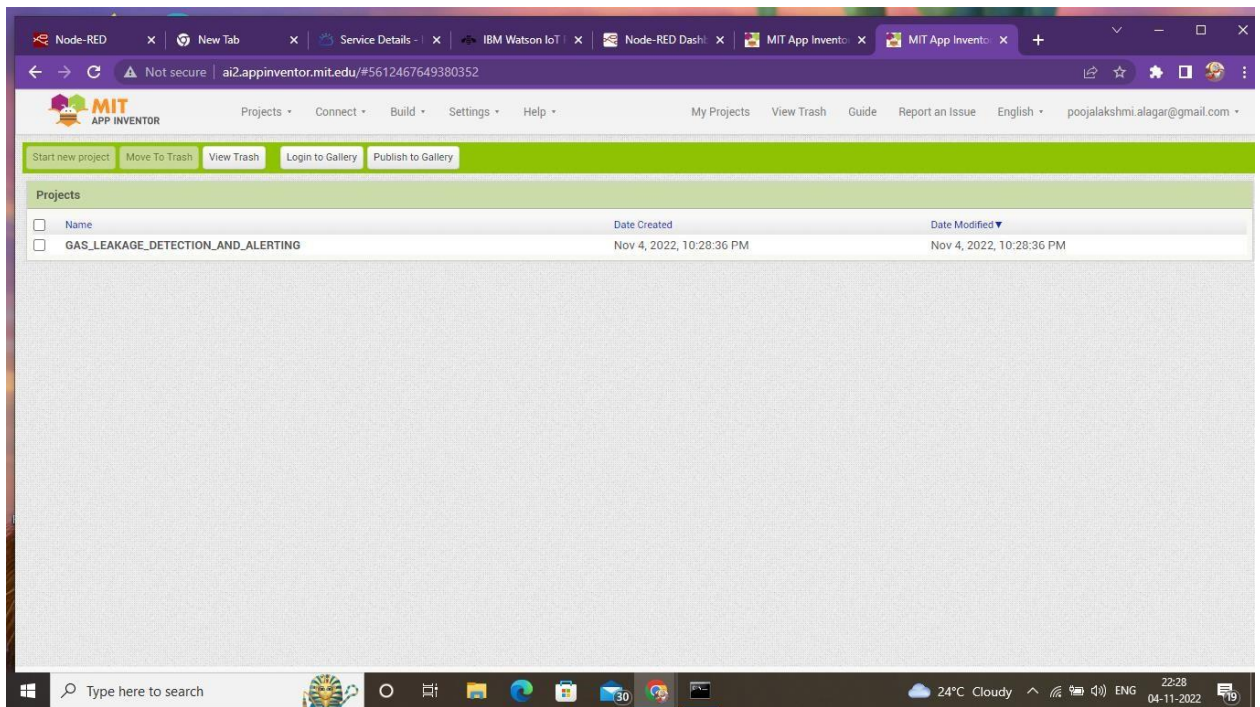


Step17: Output with light on and off button



Step 18: Login to MIT app inventor and design





Step 19: The Output

