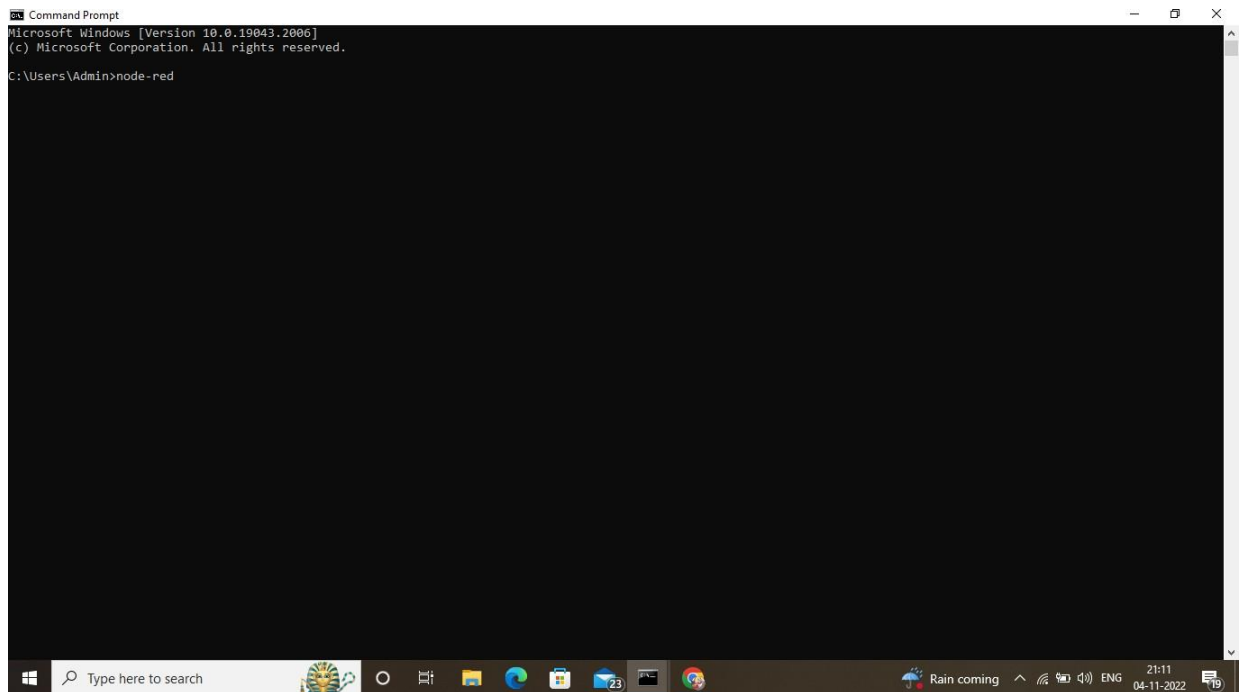


Project Development Phase

Delivery of Sprint 2

DATE	05 NOVEMBER 2022
TEAM ID	PNT2022TMID21494
PROJECT NAME	GAS LEAKAGE DETECTION AND ALERTING
MAXIMUM MARKS	20

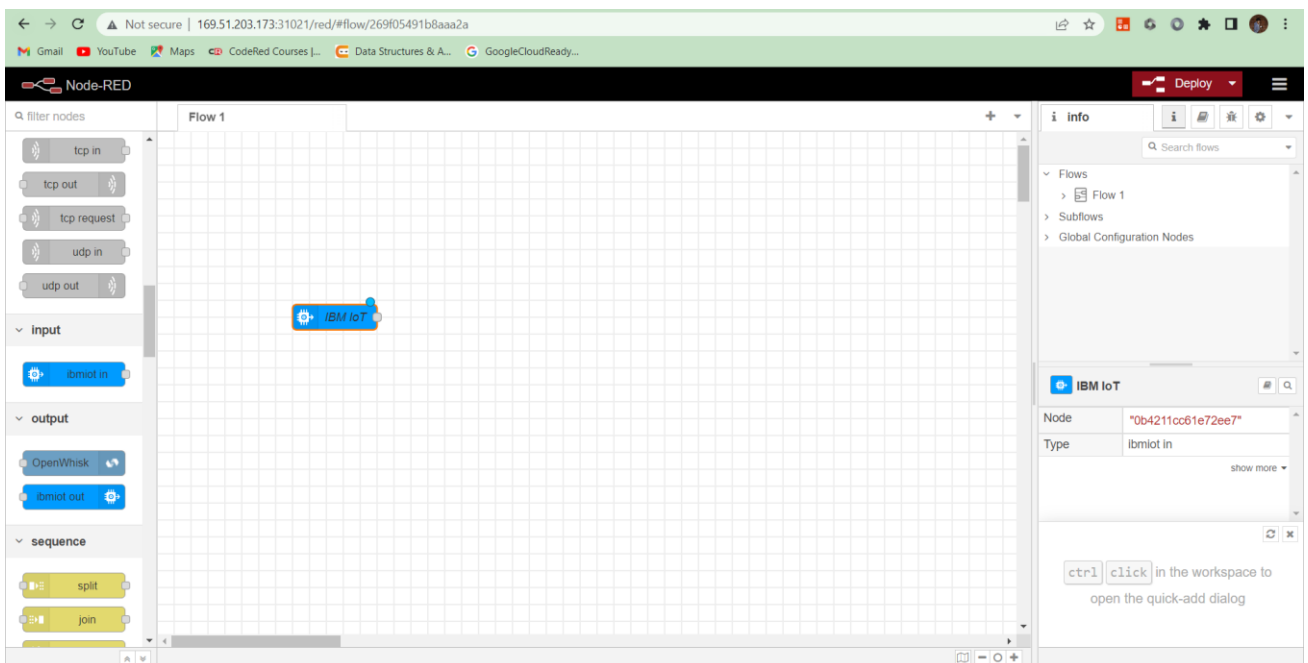
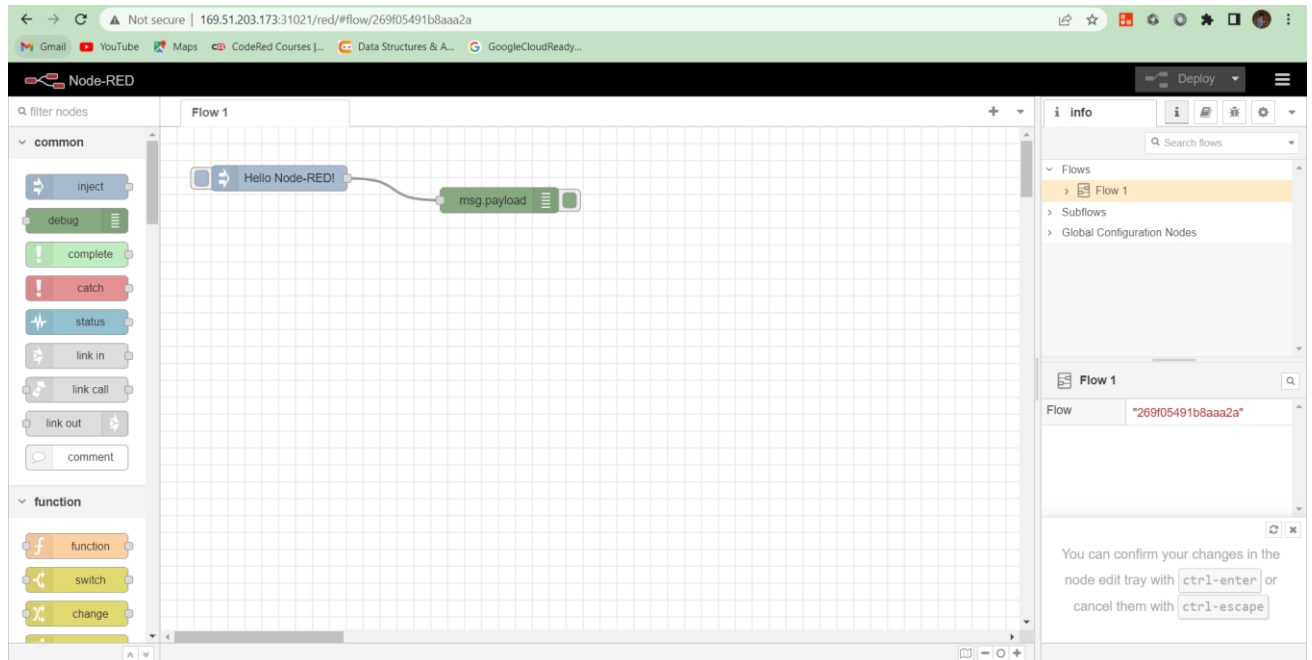
Step1: Install node red and open node red in command prompt



```
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

IBM Watson IoT Platform

kgowtham1@student.tce.edu
ID: 0z10vb


Browse IBM Cloud Apps + Generate API Key

Browse API Keys

Type the app description to search for

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	
0 results				



There are no API Keys

Generate API Key

0 Simulations running

Step 4: Click on generate API keys

The screenshot shows the IBM Watson IoT Platform interface. A modal dialog titled "Generate API Key" is open, with tabs for "Information" and "Permissions". The "Permissions" tab is active, showing a role selection dropdown set to "Visualization Application". Below the dropdown, a link provides more information about roles. At the bottom of the dialog are "Back" and "Generate Key" buttons. Below the dialog, the "Browse API Keys" section is visible, featuring a search bar and a table of API keys. The table has one entry: "1 Simulation running". The bottom of the image shows a Windows taskbar with the date 04-11-2022 and time 21:10.

IBM Watson IoT Platform

Browse IBM Cloud Apps

Generate API Key

Information Permissions

The application will have access for the following role:

Role Visualization Application

For more information about roles, see [User, application, and gateway roles](#).

Back Generate Key

Browse API Keys

Type the app description to search for

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](#).

API Key
1 Simulation running

1 Simulation running

Type here to search

Rain coming

ENG

21:10

04-11-2022

Step 5: Generated API key Details

IBM Watson IoT Platform

kgowtham1@student.tce.edu
ID: 0zi0vb


Browse IBM Cloud Apps

The API key has been added.

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.

Generated Details

API Key	a-0zi0vb-d6n7kcm1v
Authentication Token	m_4ppW3ubQILzxB((0

 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.

API Key Information

Description	-
Role	Visualization Application
Expires	Never

View API Key

Add Another

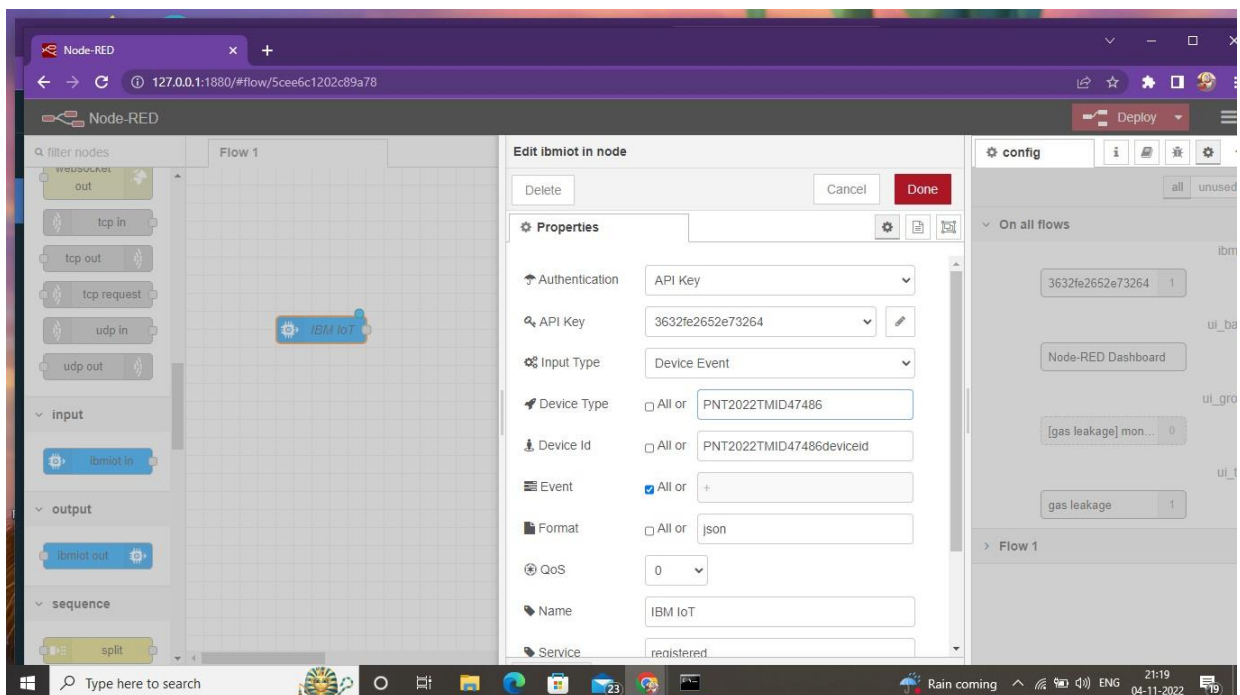
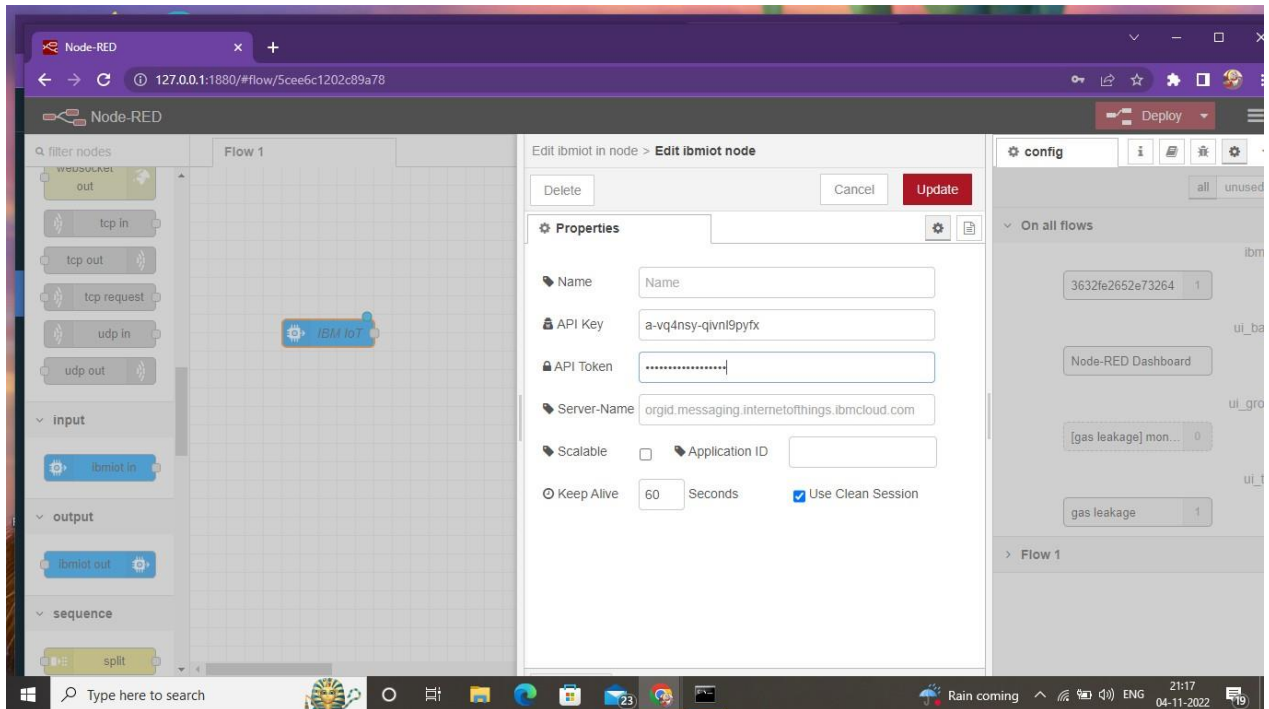
Close

Browse API Keys

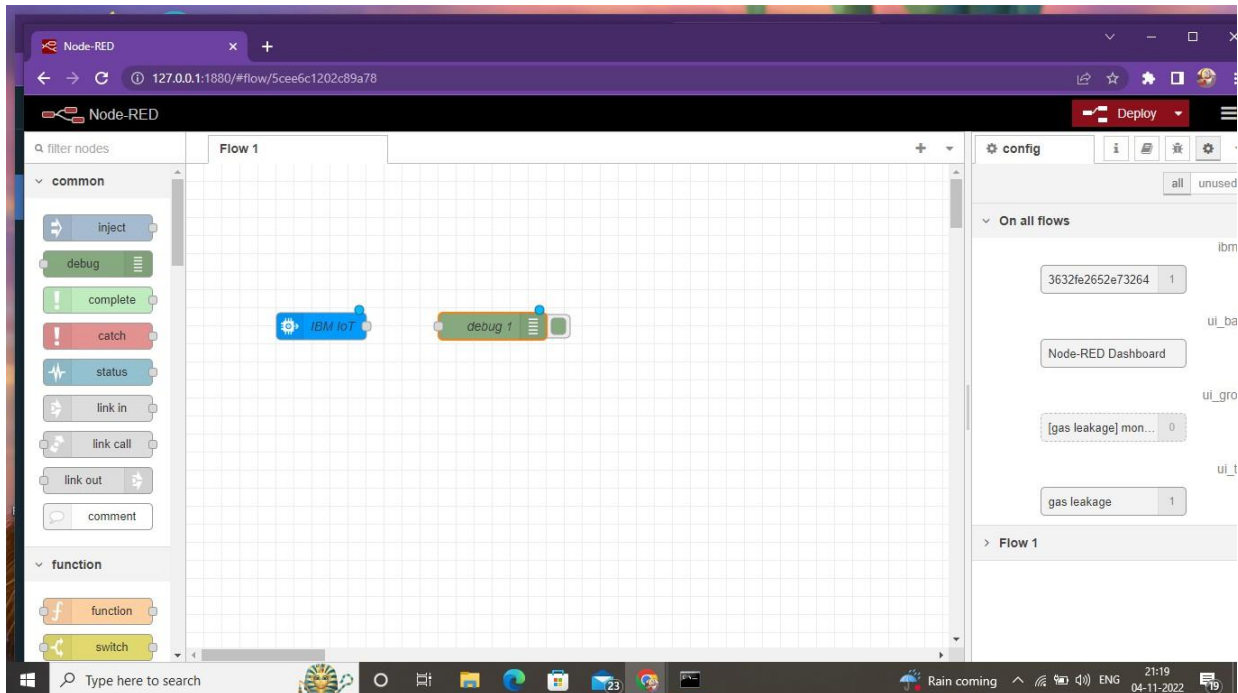
This table shows a summary of the API keys that have been added for the organization. It can be filtered

0 Simulations running

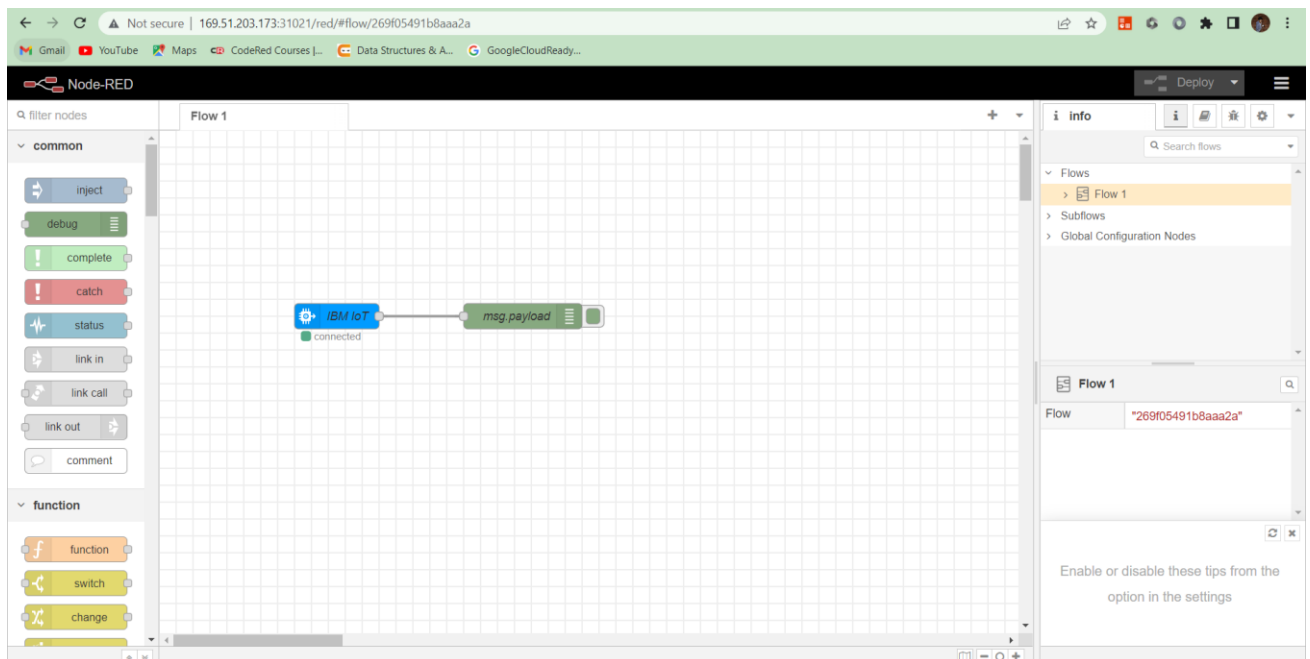
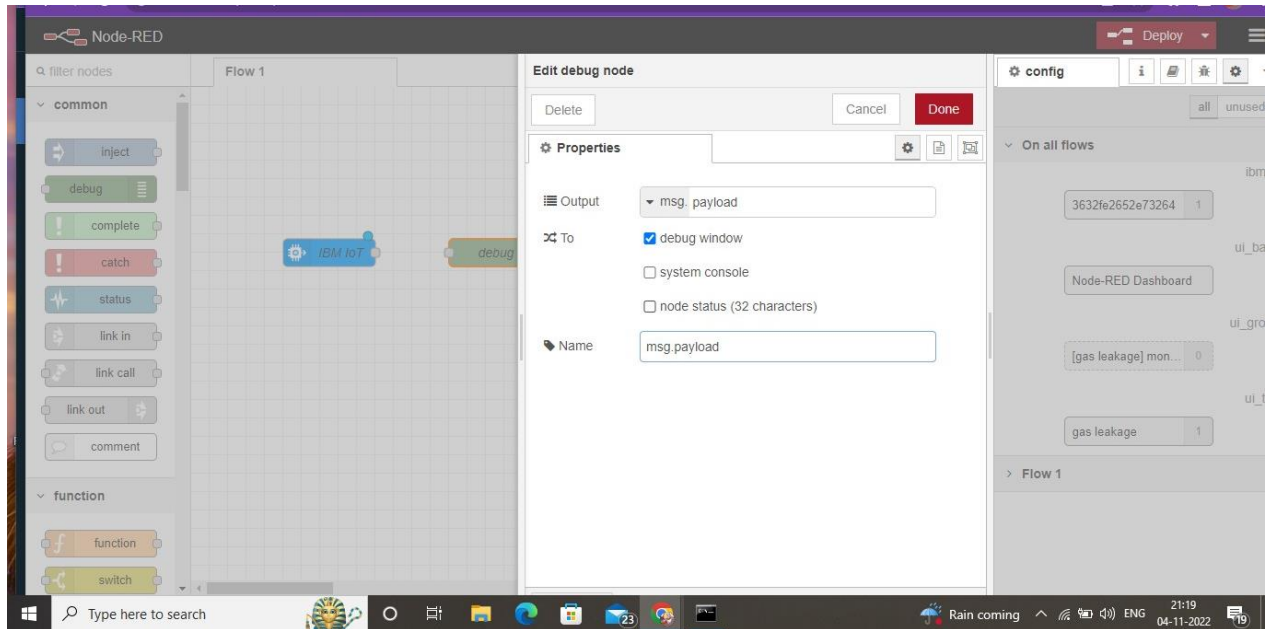
Step6: Copy and paste the generated API key in node red



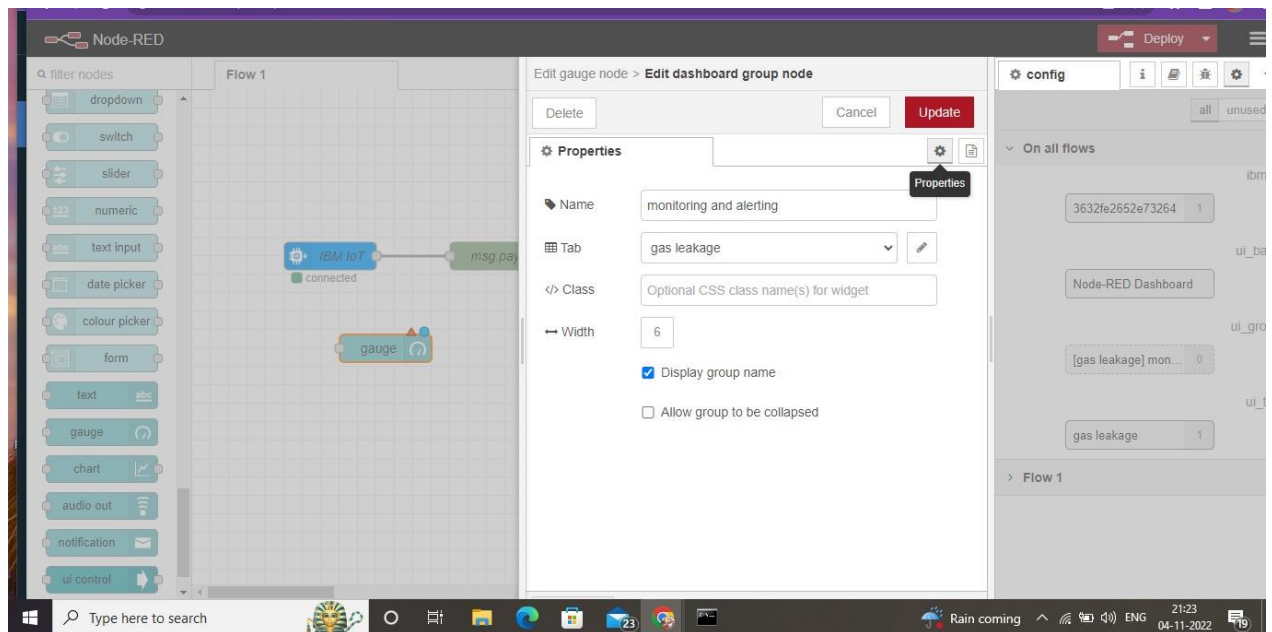
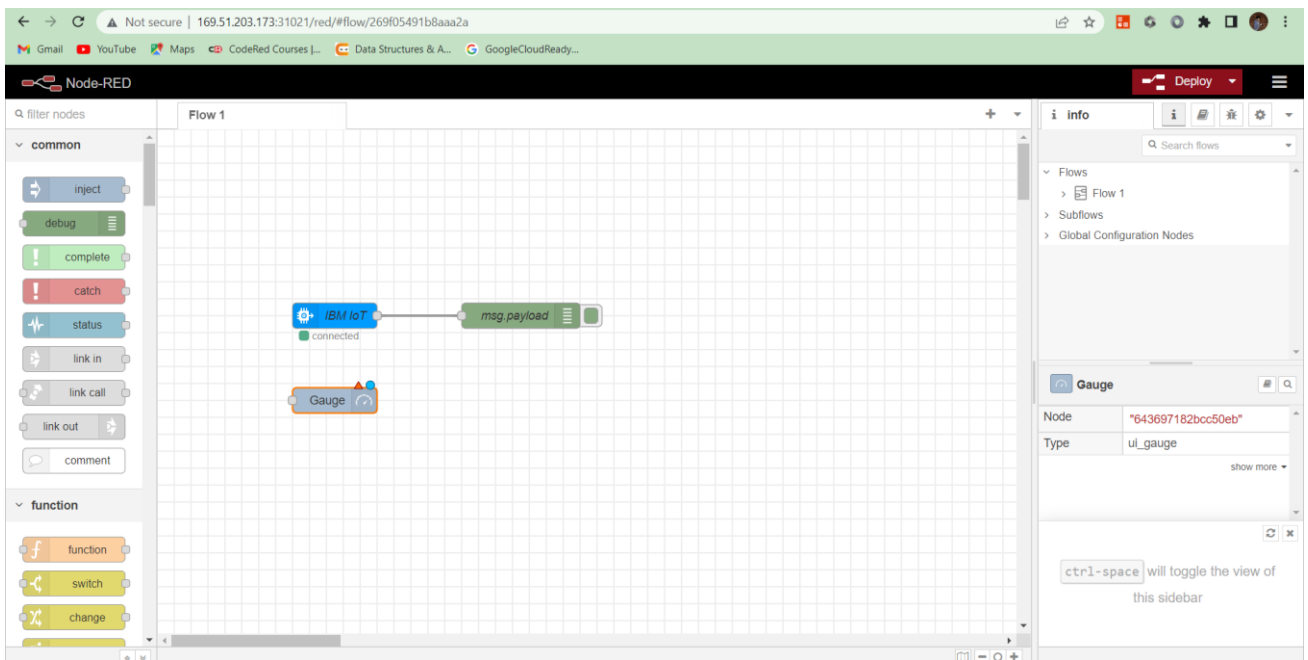
Step 7:after completing all the details click in done button



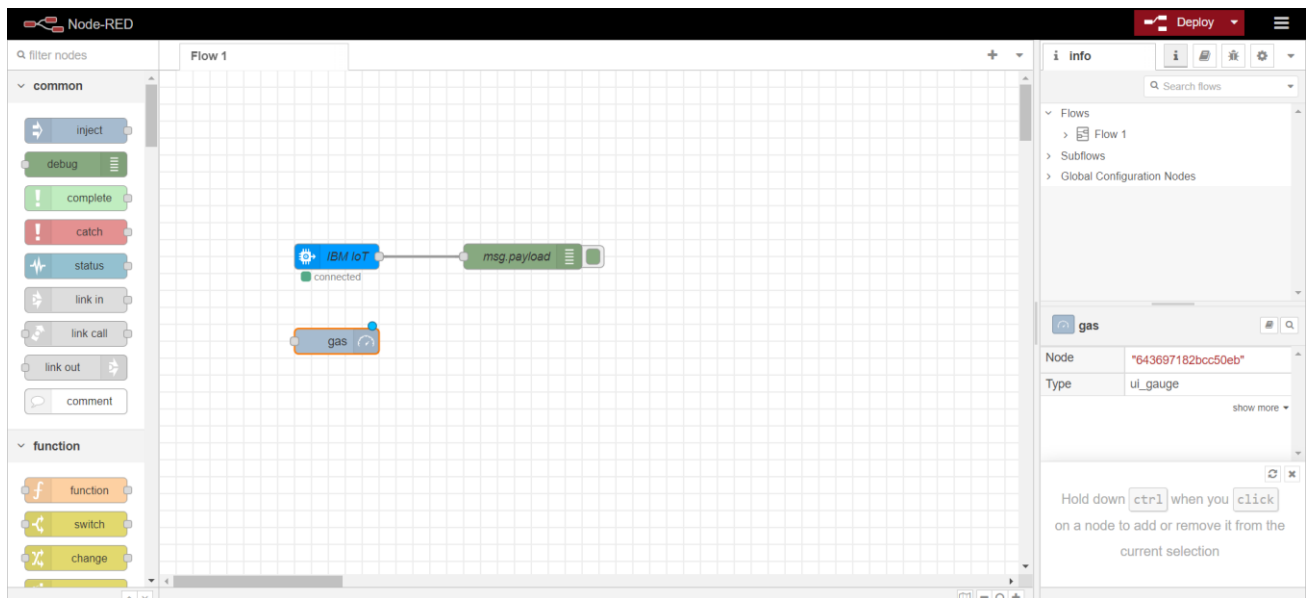
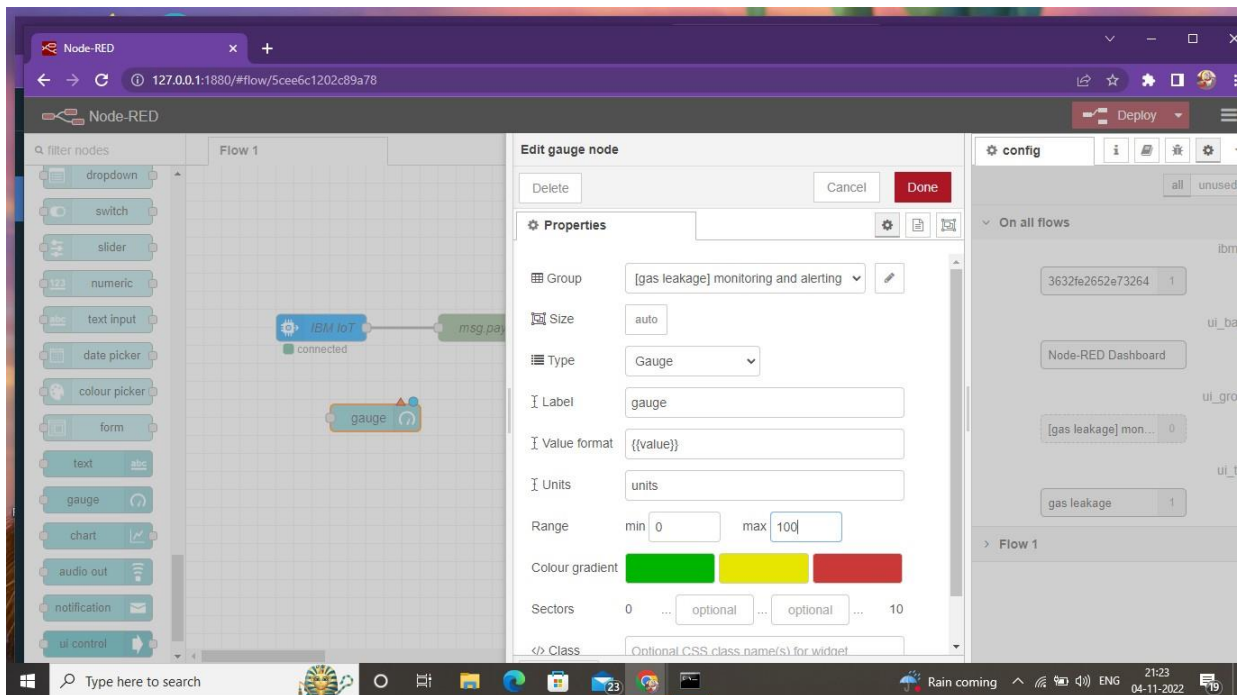
Step 8: add debug to the IBM iot and rename as msg.payload and click on done.



Step 9: Click gauge from the dashboard node and fill the details



Step 9: Click gauge from the dashboard node and fill the details



Step 10: Add functions to the gauge

The screenshot displays the Node-RED web interface. On the left, the 'function' category is expanded in the node palette, showing various nodes like 'function', 'switch', 'change', 'range', 'template', 'delay', 'trigger', 'exec', and 'filter'. The main workspace, titled 'Flow 1', contains a flow starting with an 'IBM IoT' node (labeled 'connected') connected to a 'msg.payload' node. Below these, a 'gauge' node is connected to a 'function 1' node. The right sidebar shows the 'config' tab with a list of nodes used in the flow, including '3632fe2652e73264', 'Node-RED Dashboard', '[gas leakage] mon...', and 'gas leakage'. The bottom status bar indicates the system time as 21:24 on 04-11-2022.

Step 11: Check the values from debug messages

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 with the following nodes: an 'IBM IoT' node (blue) with a 'connected' status, a 'msg.payload' node (green), a 'gauge' node (blue), and a 'function 1' node (orange). The right sidebar shows the 'Debug messages' panel, which displays a list of messages received from the 'node: debug 1' and 'node: msg payload' nodes. The messages are JSON objects containing temperature data.

Debug messages:

- 11/4/2022, 9:15:23 PM node: debug 1
iot-2/type/PNT2022TMID47486Id/PNT2022TMID47486devId/evt/event_1/fmt/json : msg.payload : Object
{ temperaturer: 52 }
- 11/4/2022, 9:15:26 PM node: debug 1
iot-2/type/PNT2022TMID47486Id/PNT2022TMID47486devId/evt/event_1/fmt/json : msg.payload : Object
{ temperaturer: 46 }
- 11/4/2022, 9:15:29 PM node: debug 1
iot-2/type/PNT2022TMID47486Id/PNT2022TMID47486devId/evt/event_1/fmt/json : msg.payload : Object
{ temperaturer: 12 }
- 11/4/2022, 9:20:09 PM node: msg payload
iot-2/type/PNT2022TMID47486Id/PNT2022TMID47486devId/evt/event_1/fmt/json : msg.payload : Object
{ temperaturer: 67 }
- 11/4/2022, 9:20:12 PM node: msg payload
iot-2/type/PNT2022TMID47486Id/PNT2022TMID47486devId/evt/event_1/fmt/json : msg.payload : Object
{ temperaturer: 67 }

Step 12: Edit function node

The screenshot shows the Node-RED web interface. On the left, the 'function' node is selected from the palette. In the center workspace, a flow is visible: an 'IBM IoT' node (blue) is connected to a 'msg.payload' node (green), which is then connected to a 'function 1' node (orange). A 'gauge' node (blue) is also present but not connected. The right-hand 'debug' console shows a log of messages received from the IoT node, including timestamps and JSON payloads like `{ temperature: 12 }`.

This screenshot shows the same Node-RED interface, but the 'function 1' node is now highlighted with a dashed orange border, indicating it is being edited. The 'debug' console on the right shows updated messages, including one with `{ temperature: 98 }` and another with `{ temperature: 73 }`. The overall flow structure remains the same: IBM IoT node connected to msg.payload node, which connects to the function node, with a gauge node also present.

← → ↻ 🔒 Not secure | 169.51.203.173:31021/red/#flow/269f05491b8aaa2a

Gmail YouTube Maps CodeRed Courses [...] Data Structures & A... GoogleCloudReady...

Node-RED

Deploy

filter nodes

Flow 1

common

inject

debug

complete

catch

status

link in

link call

link out

comment

function

function

switch

change

IBM IoT

connected

gas

Edit function node

Delete

Cancel

Done

Properties

Name

Name

Setup

On Start

On Message

On Stop

1 msg.payload = msg.payload.temperature

2 return msg;

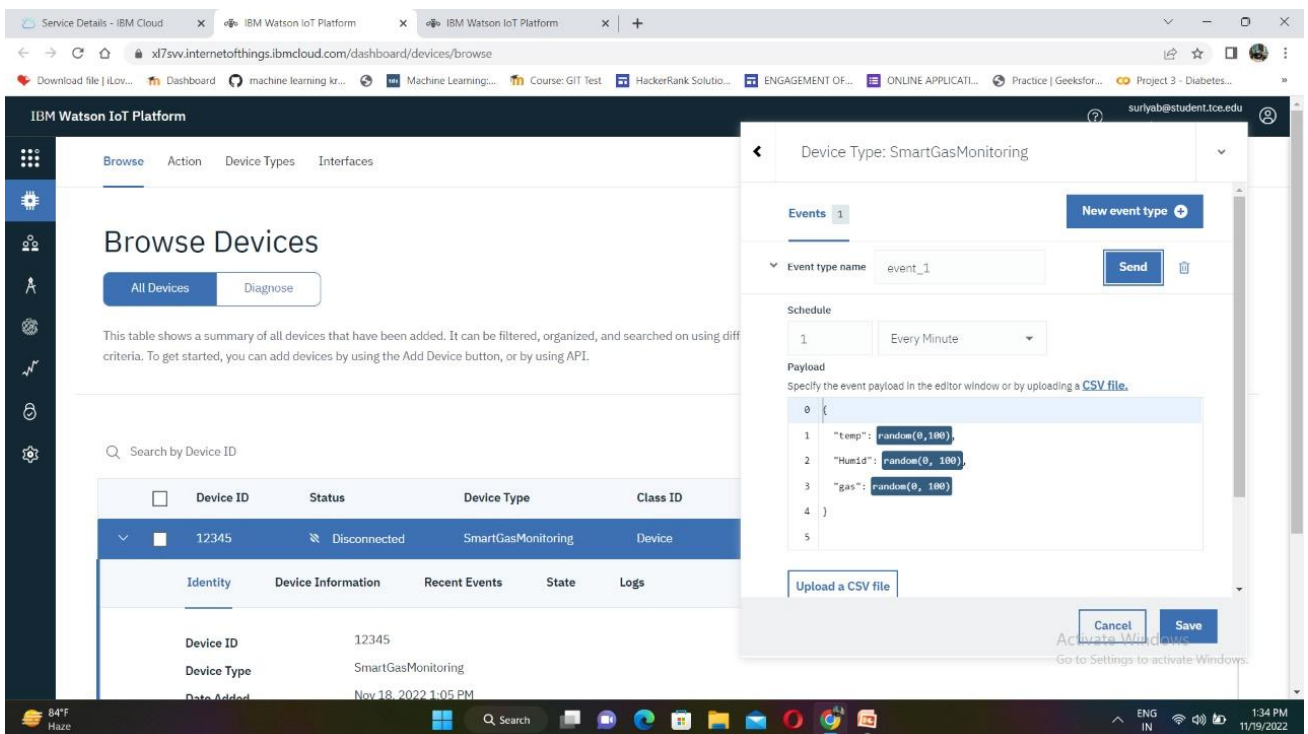
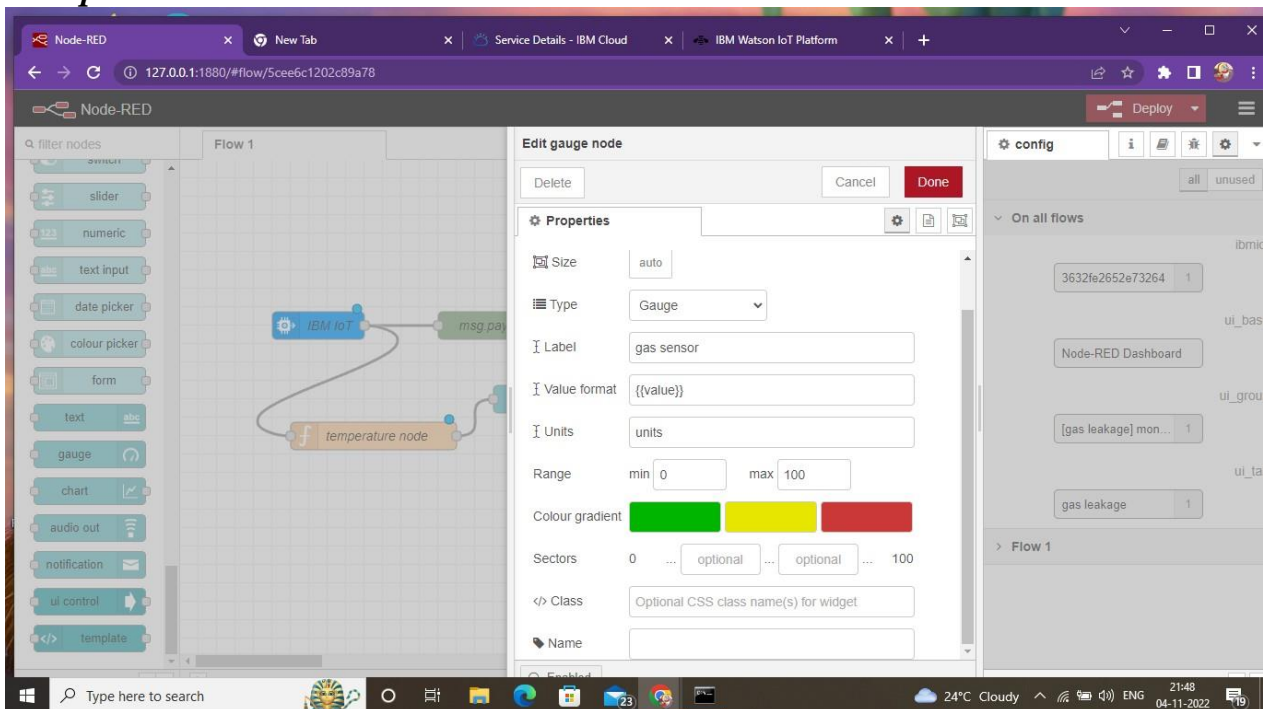
debug

all nodes

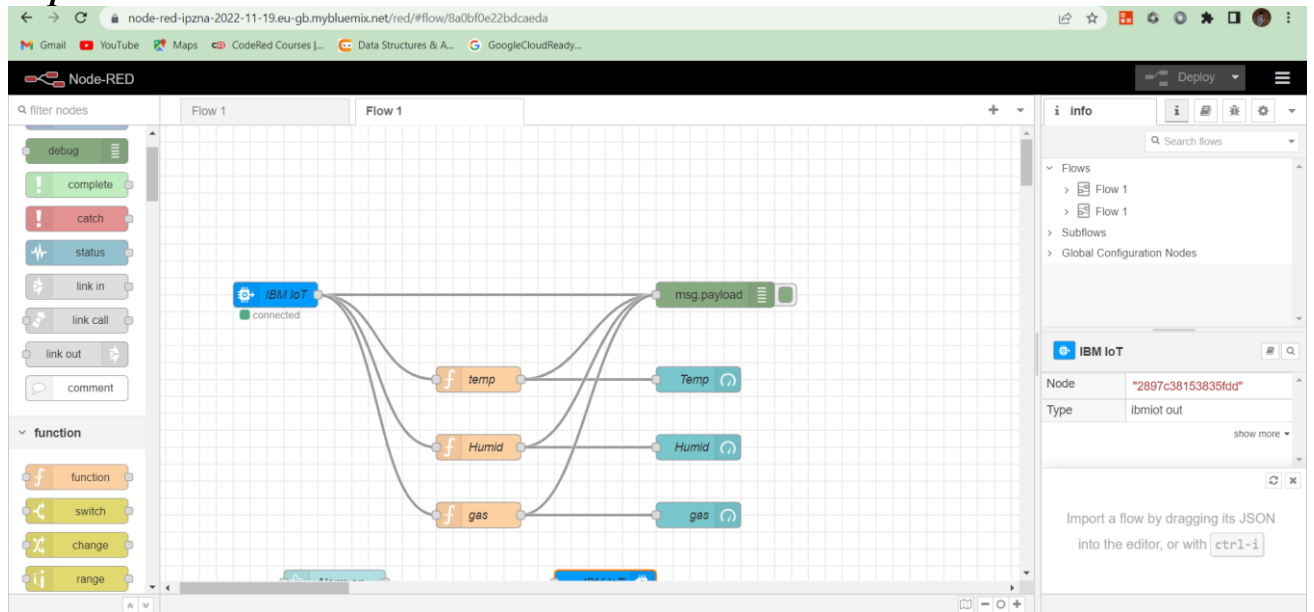
all

Enabled

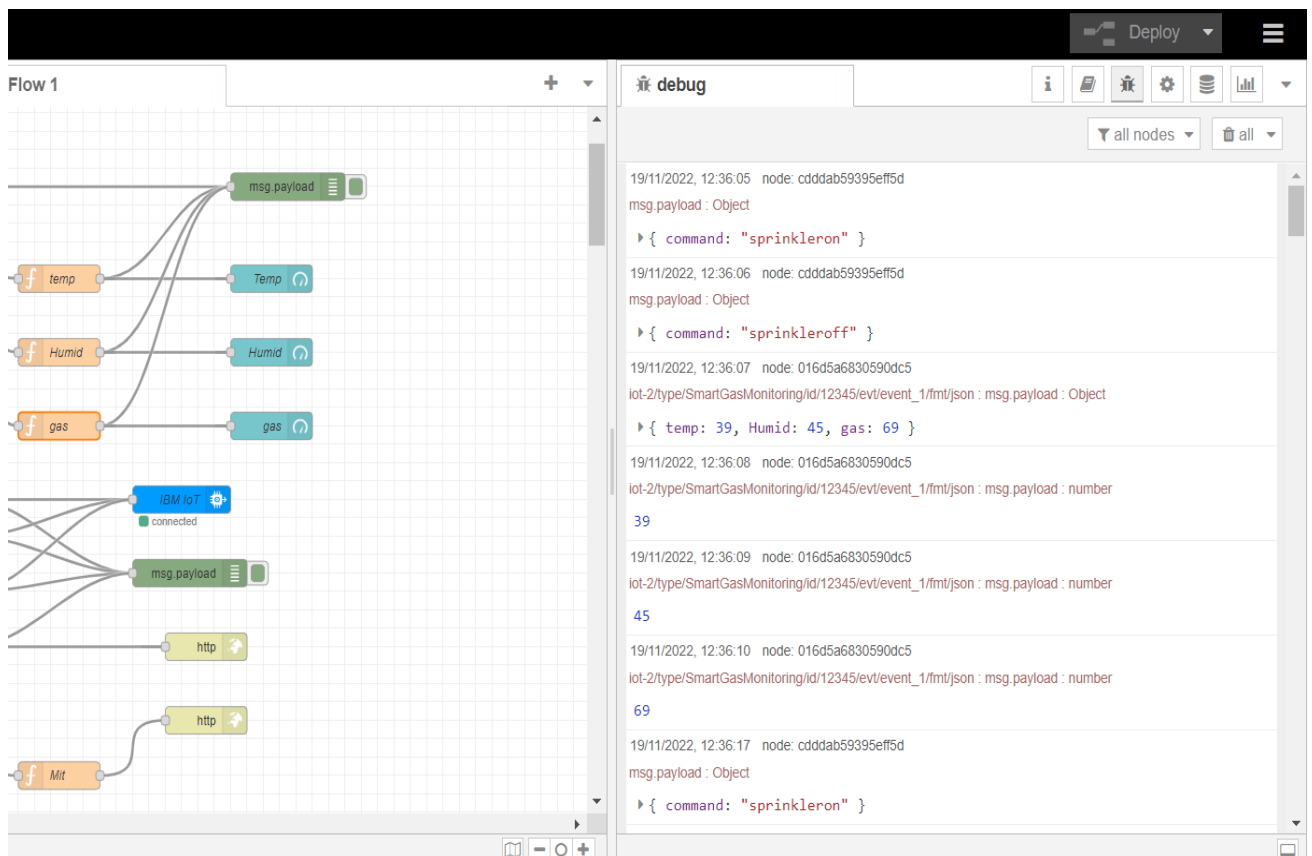
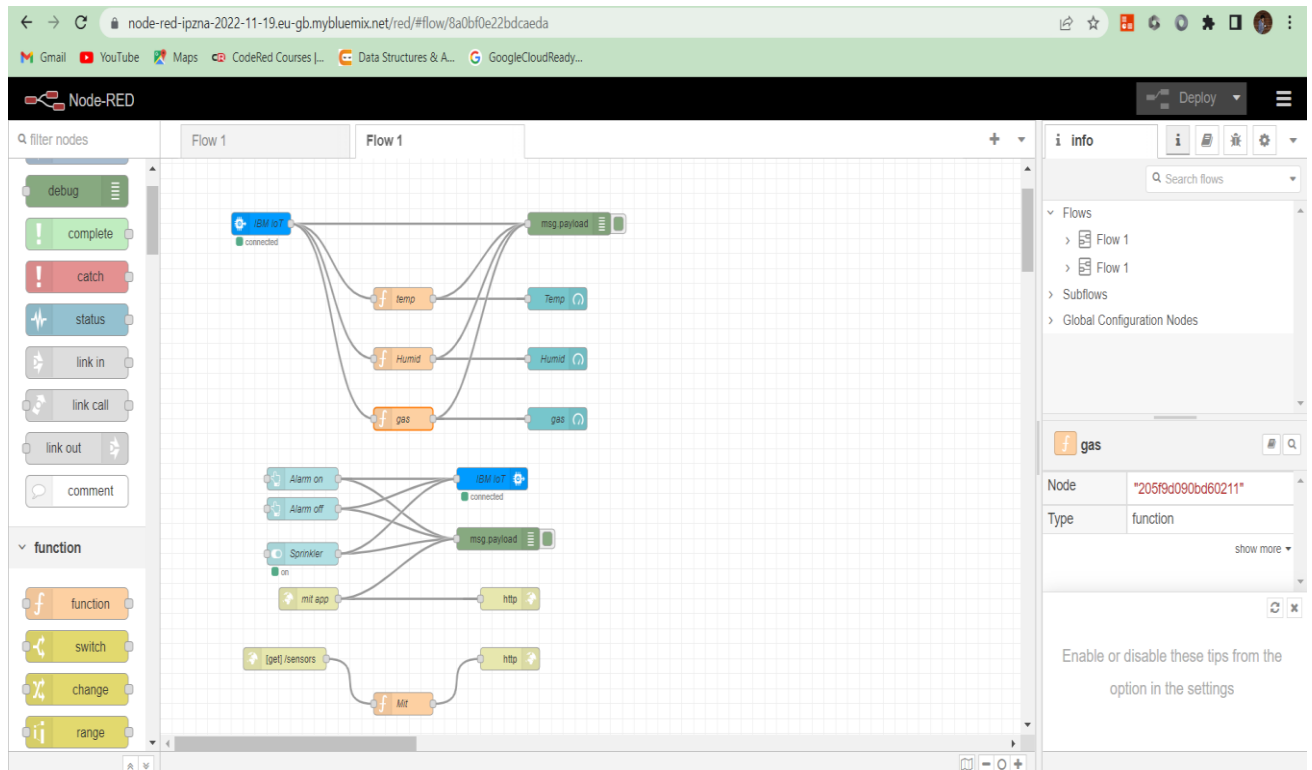
Step 13: Connect them



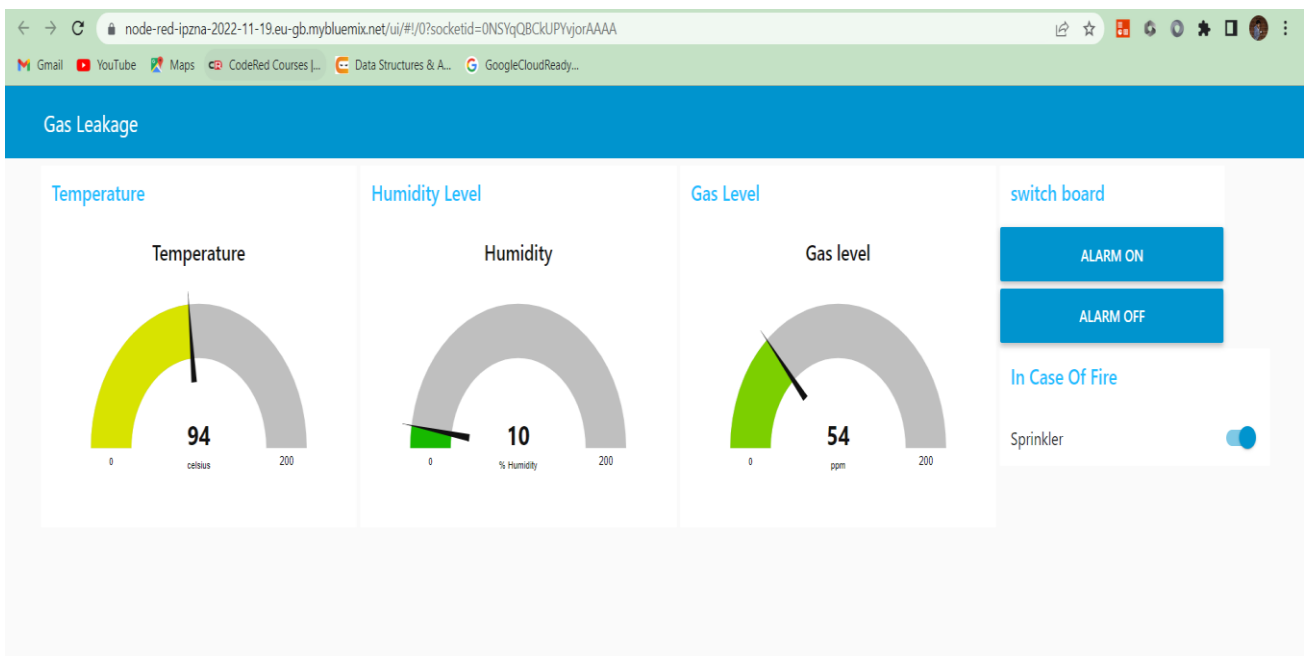
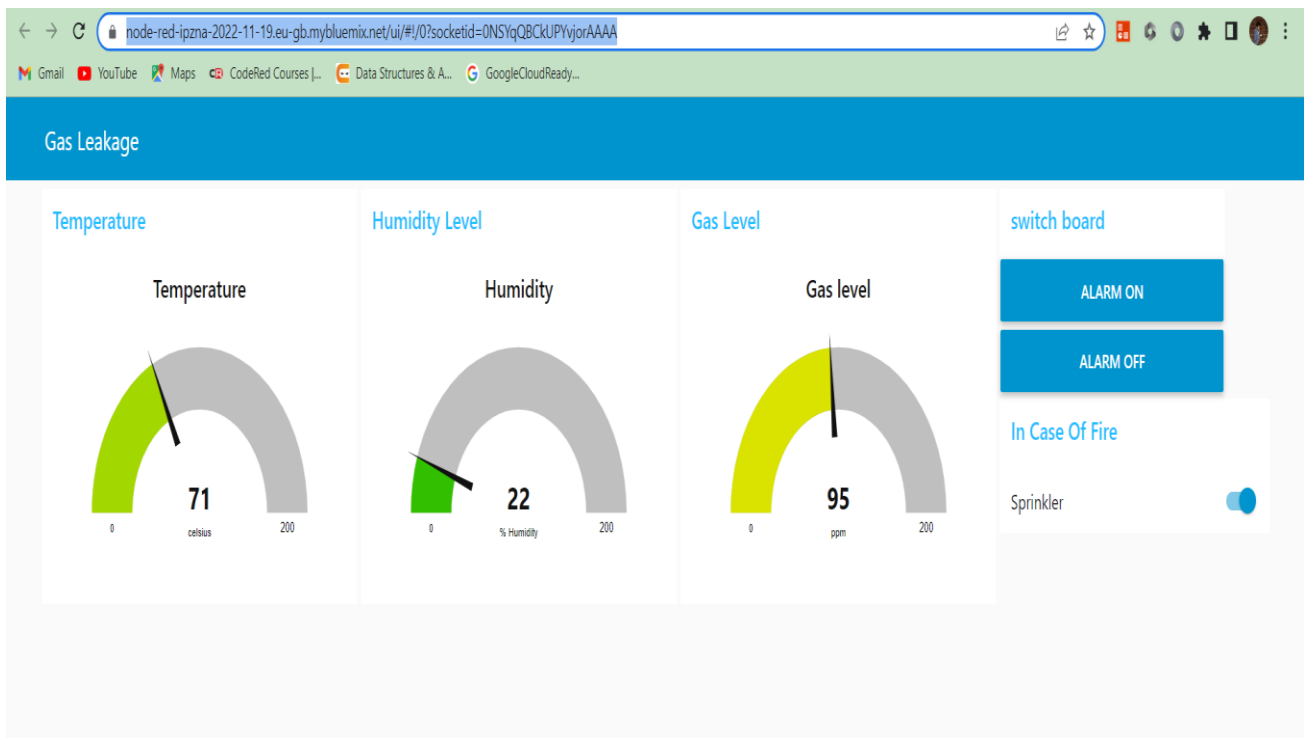
Step 13: Connect them



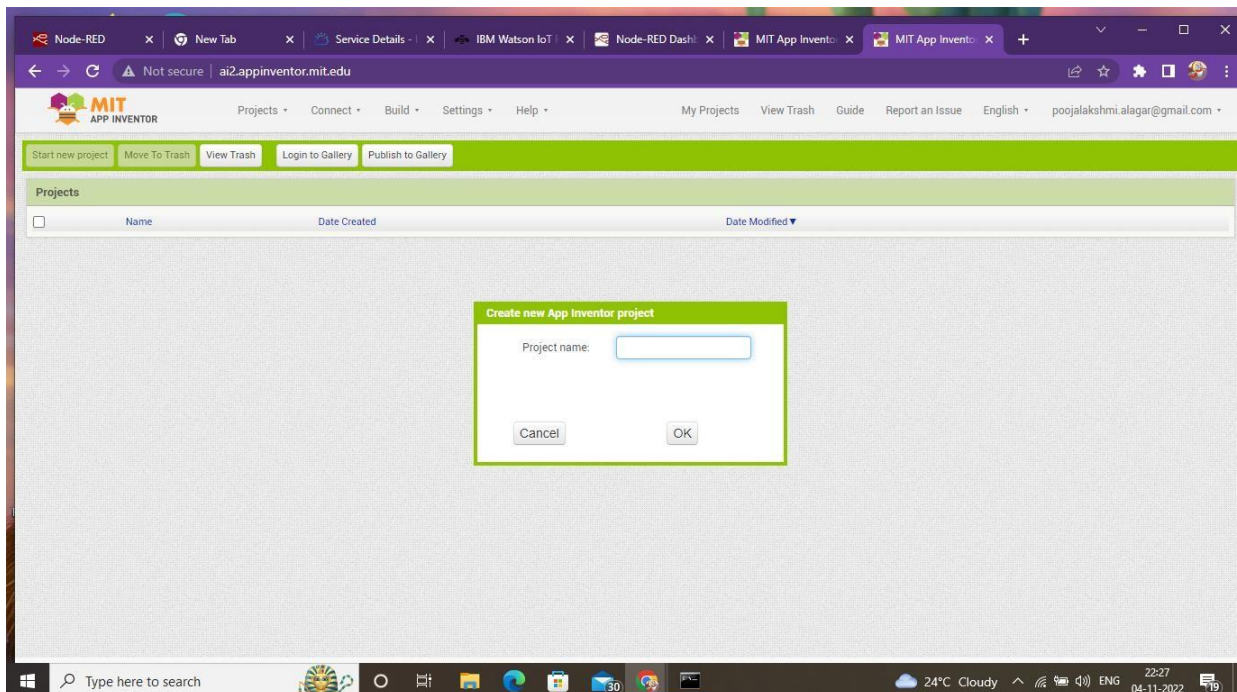
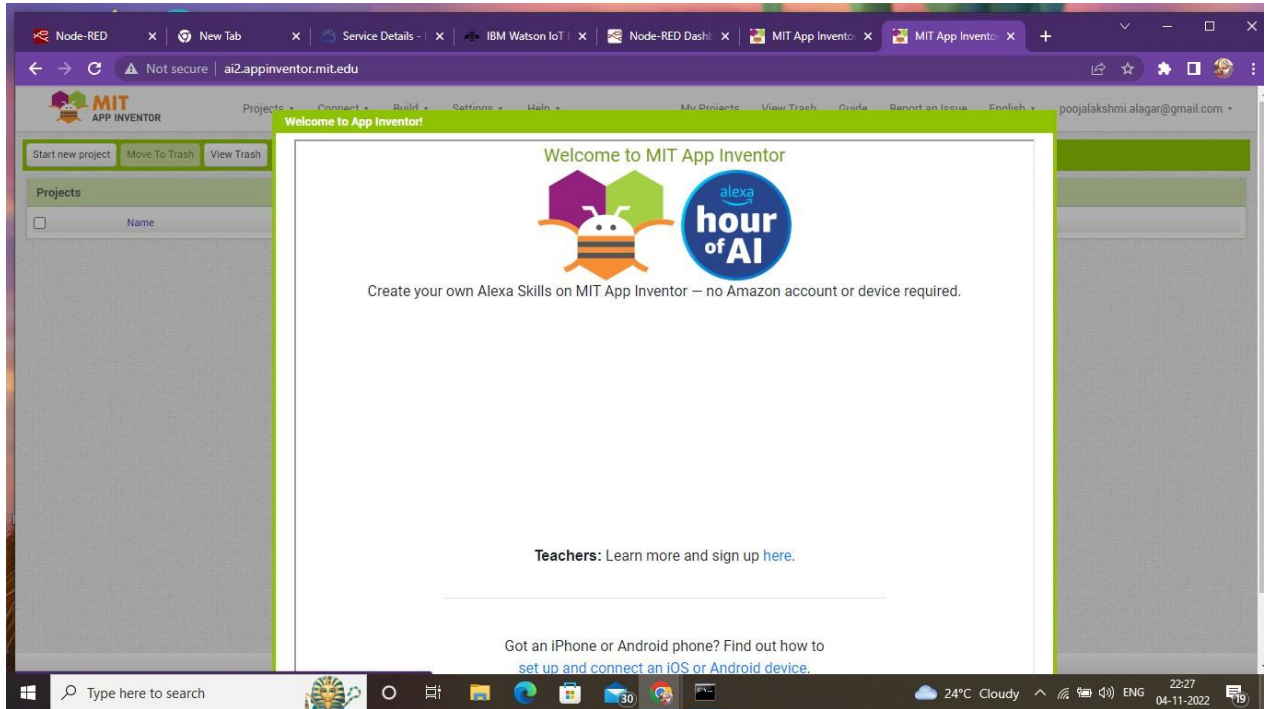
Step14: finally add alarm on and off buttons to IBM iot and debug.step

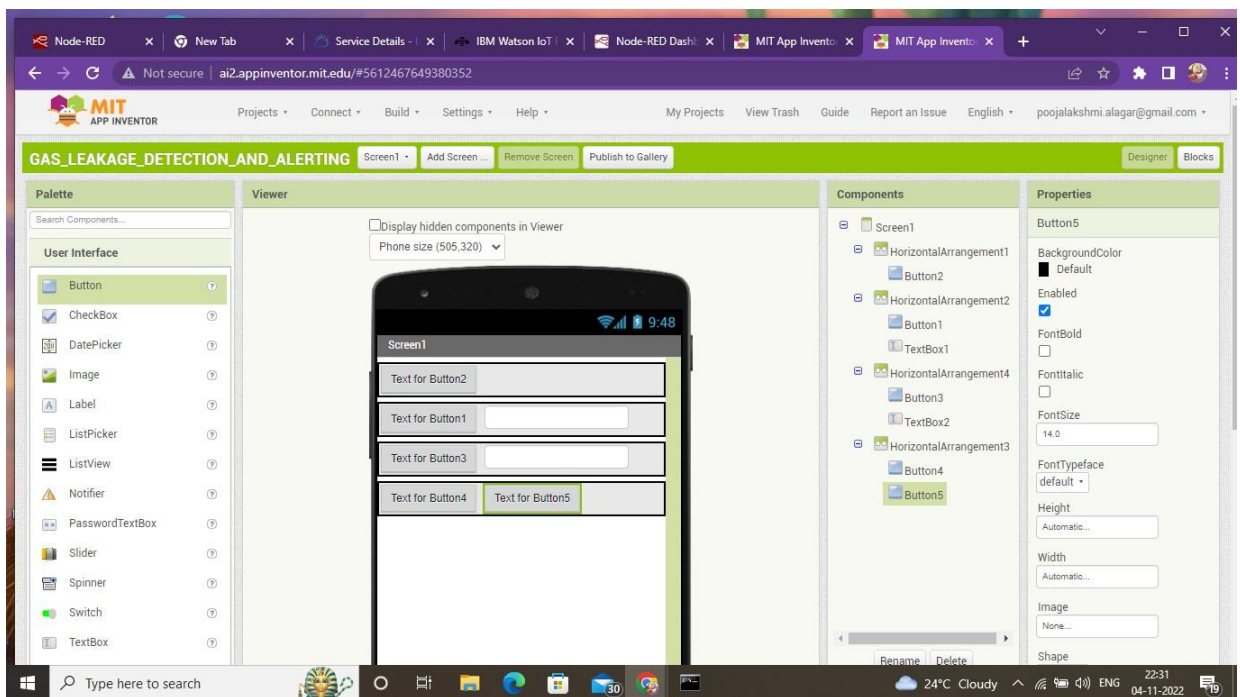
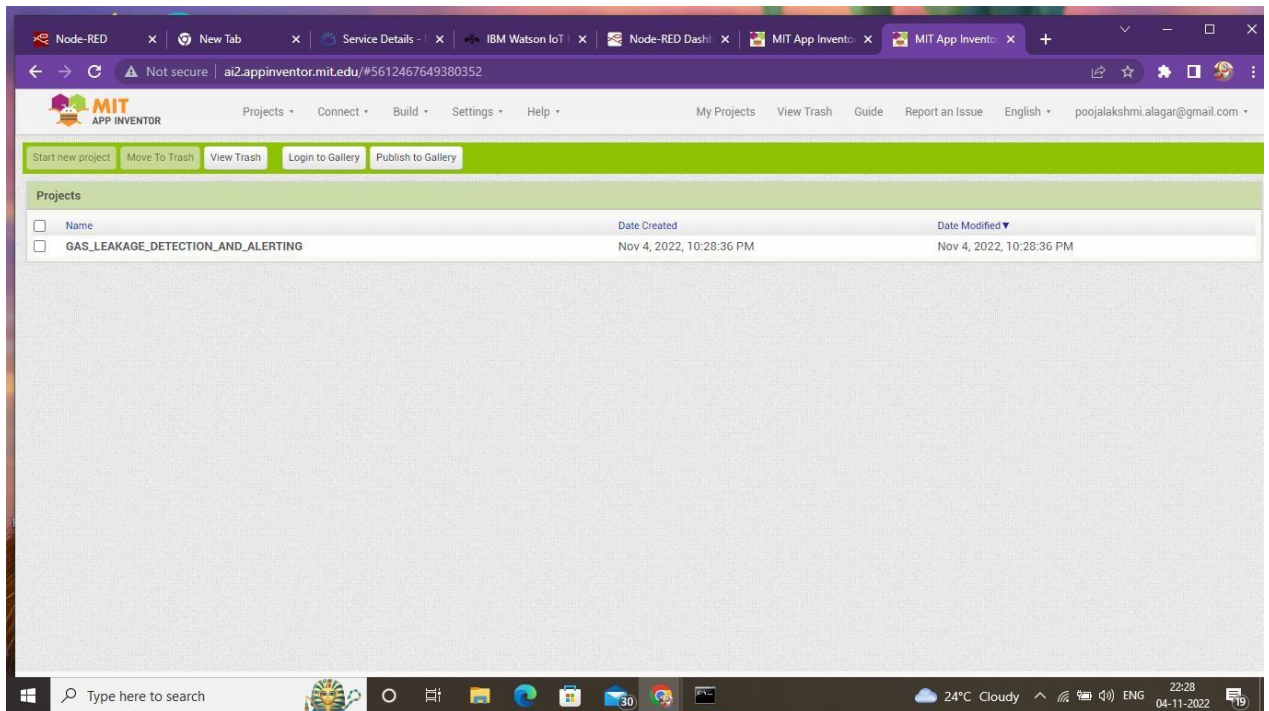


Step15: Output from node red



Step 16: Login to MIT app inventor and design





Step 19: The Output

