

## Project development phase

### Sprint - 3

Date	17 November 2022
Team ID	PNT2022TMID36769
Project Name	Project - Industry-specific intelligent fire management system
Maximum Marks	20 marks

▼ IN Sprint 2 31 Oct – 5 Nov (2 issues)

IN-4 In industry, sensor sense the fire and smoke. **SENSOR & ACTUATOR**

IN-5 If the sensor detected the fire, next step is extinguishing the fire with the help of Sprinkler. **SENSOR & ACTUATOR**

⇒ Configure the connection security and create API keys that are used in the Node-RED service for accessing the IBM IoT Platform.



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-dvo306-lmrrimazvy

Authentication Token 08+bzC68Sx0bZYDSCL



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 Standard 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.

1 Simulation running

27°C Cloudy

2021

12:11:2022

## US-2 Create a Node-RED service

The screenshot displays the Node-RED web interface in a browser. The URL is `node-red-uwrpn-2022-11-12.eu-gb.mybluemix.net/red/#flow/caec888b429a5cb`. The interface includes a left sidebar with node categories (common, function), a central workspace for 'Flow 1', and a right sidebar for debugging.

**Flow 1 Diagram:**

- IBM IoT** node (connected) receives data from the IBM Watson IoT Platform.
- The data is split into three parallel paths:
  - Temperature node** → **Temperature** output node
  - Flame level** node → **Flame level** output node
  - Gas level node** → **Gas level** output node
- A **msg.payload** node is also connected to the main flow.

**Debug Console:**

The debug console shows the following log entries:

```
msg.payload : Object
  { Temperature: 60, Flame level: 30, Gas level: 37 }
11/13/2022, 7:08:36 PM node: aa0ccf3bbeaf648e
lot-
2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json
: msg.payload : Object
  { Temperature: 30, Flame level: 41, Gas level: 40 }
11/13/2022, 7:08:36 PM node: aa0ccf3bbeaf648e
lot-
2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json
: msg.payload : Object
  { Temperature: 65, Flame level: 89, Gas level: 77 }
11/13/2022, 7:08:36 PM node: aa0ccf3bbeaf648e
lot-
2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json
: msg.payload : Object
  { Temperature: 97, Flame level: 97, Gas level: 11 }
11/13/2022, 7:08:41 PM node: aa0ccf3bbeaf648e
lot-
2/type/sona22devicetype/id/sona22/evt/event_1/fmt/json
: msg.payload : Object
  { Temperature: 49, Flame level: 81, Gas level: 50 }
```

**Fig1** - Monitoring the sensor values - Temperature, Flame Level, Gas Level. These values are randomly generated by IBM WATSON IOT PLATFORM.

## debug

11/13/2022, 7:09:56 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

▶ { Temperature: 50, Flame level: 65, Gas level: 69 }

11/13/2022, 7:09:56 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

▶ { Temperature: 23, Flame level: 24, Gas level: 59 }

11/13/2022, 7:09:59 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

▶ { Temperature: 66, Flame level: 82, Gas level: 20 }

11/13/2022, 7:10:04 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

▶ { Temperature: 58, Flame level: 62, Gas level: 41 }

11/13/2022, 7:10:08 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

▶ { Temperature: 20, Flame level: 82, Gas level: 83 }

11/13/2022, 7:10:15 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

▶ { Temperature: 18, Flame level: 27, Gas level: 46 }

11/13/2022, 7:10:19 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

▶ { Temperature: 5, Flame level: 98, Gas level: 84 }

11/13/2022, 7:10:25 PM node: aa0ccf3bbeaf648e

iot-2/type/sona22devicetype/id/sona22/evt/event\_1/fmt/json : msg.payload : Object

## Weather Monitoring

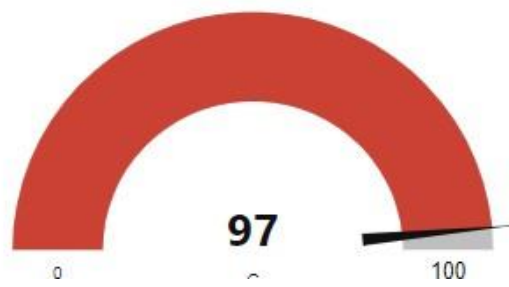
Flame level



Gas level

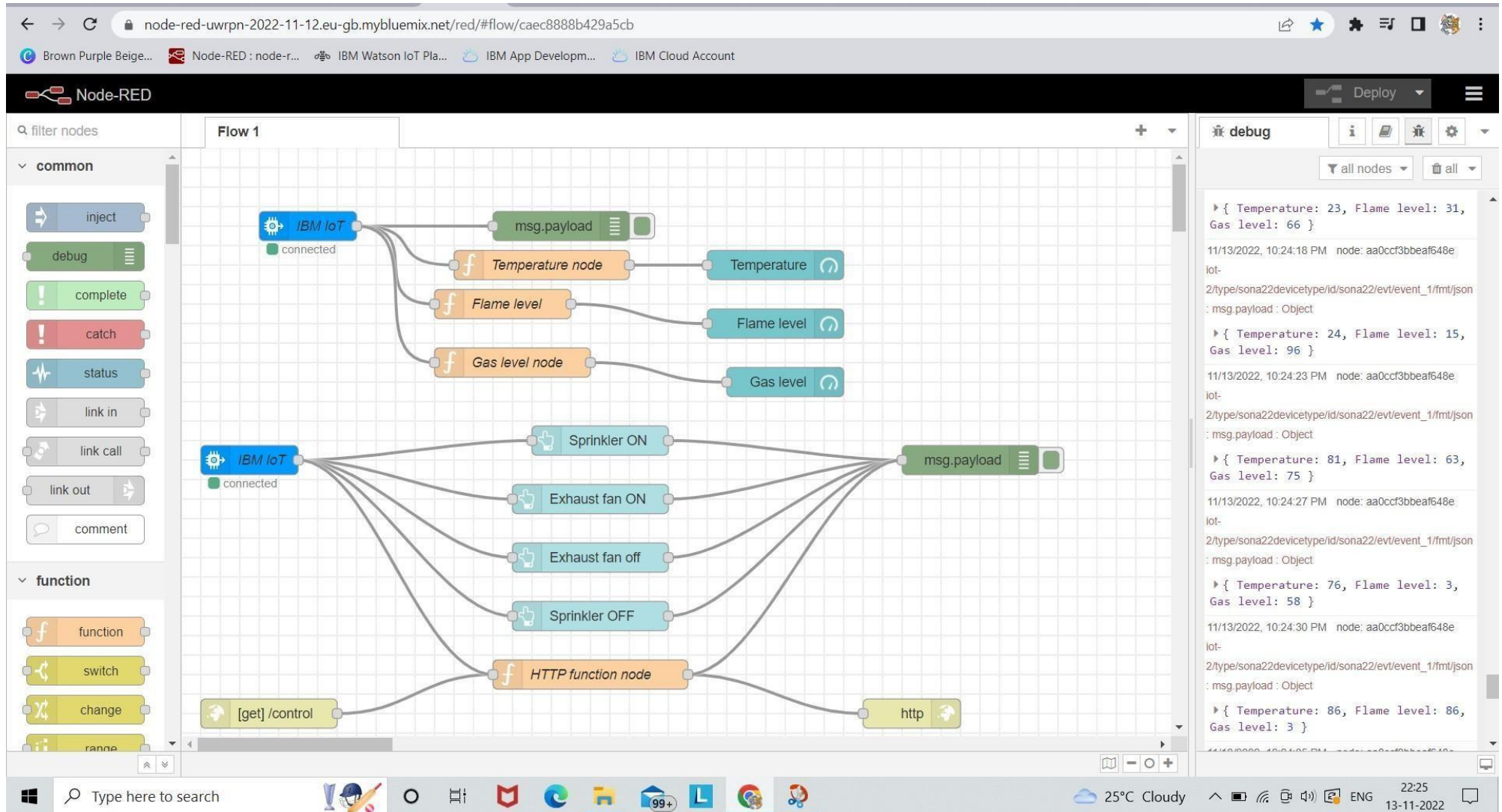


Temperature





**Fig 2 - Temperature, Flame Level, Gas Level values displayed in deploy tab in node-red**



**Fig 3 : Using HTTP in and HTTP response in network option ,<https://node-red-uwrpn-2022-11-12.eu-gb.mybluemix.net/red/#flow/caec8888b429a5cb> will display sensor values from the IBM WATSON IOT PLATFORM**

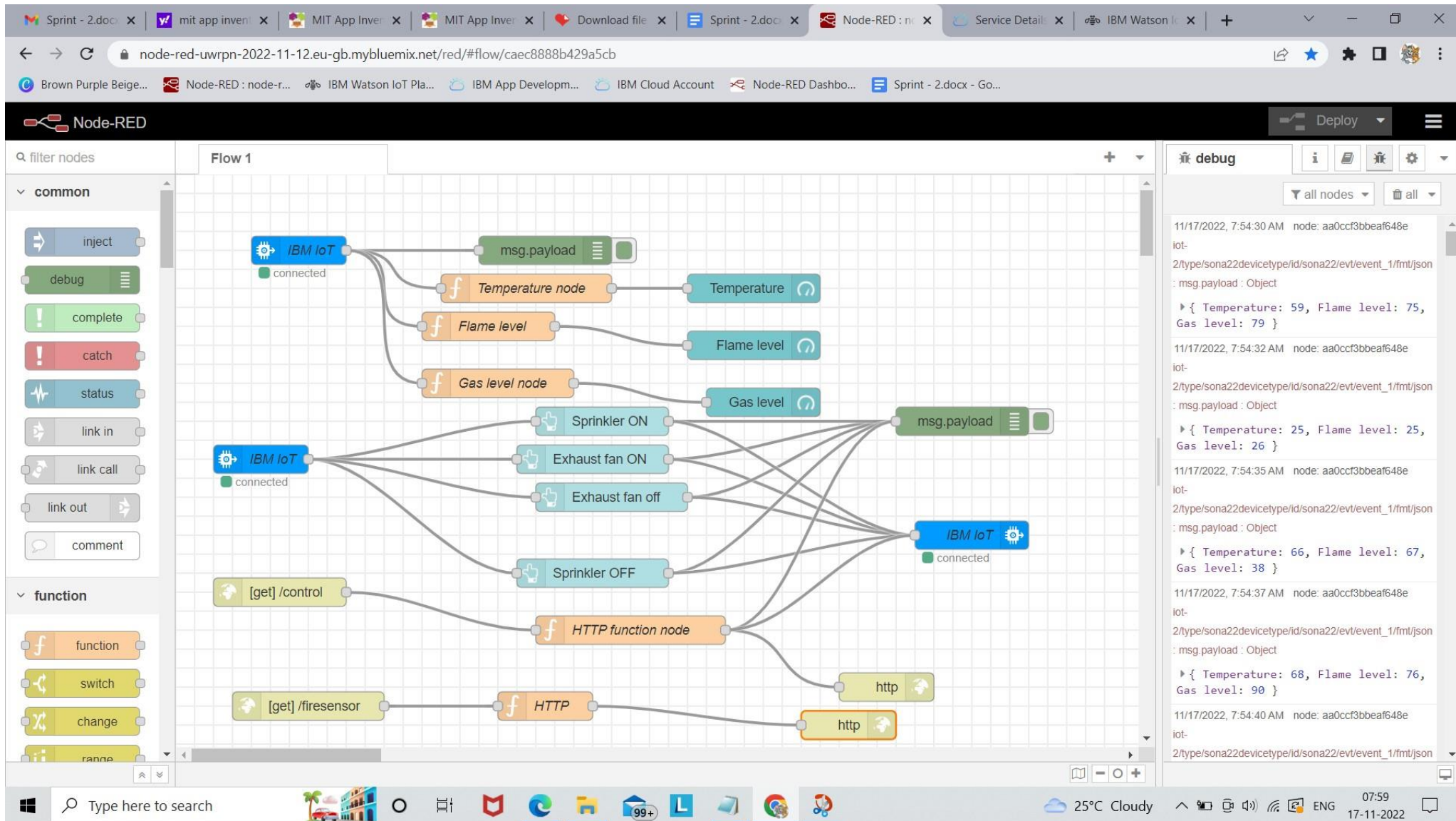
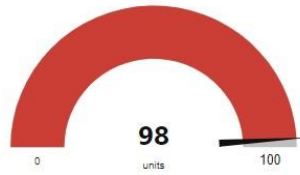


Fig 4 : Monitoring the temperature ,flame and gas sensor

## Weather Monitoring

Flame level



Gas level



EXHAUST FAN ON

EXHAUST FAN OFF

SPRINKLER ON

SPRINKLER OFF

Temperature



Fig 5

: Properties of ibm iot node



Edit ibmiot in node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖨

🔑 Authentication

API Key

▼

🔑 API Key

d709a7c4dcd74895

▼

✎

⚙ Input Type

Device Event

▼

🔑 Device Type

☐ All or

sona22devicetype

👤 Device Id

☐ All or

sona22

📋 Event

☒ All or

+

📄 Format

☐ All or

json

⚙ QoS

0

▼

🔑 Name

IBM IoT

🔑 Service

registered

Use the Input Type property to configure this node to receive Events sent by IoT Devices. Commands sent to IoT Devices. Status.

☐ Enabled

Fig 6 : Properties of temperature node

Edit gauge node

Delete

Cancel

Done

⚙ Properties

⚙

📄

🖼

📊 Group

[control] Weather Monitoring

▼

✎

📏 Size

auto

☰ Type

Gauge

▼

🏷 Label

Temperature

🏷 Value format

{{value}}

🏷 Units

C

Range

min

0

max

100

Colour gradient

Sectors

0

...

optional

...

optional

...

100

🏷 Name

☐ Enabled

Fig 7: Properties of Flame

Edit gauge node

Delete

Cancel

Done

Properties

Group

[control] Weather Monitoring

Size

auto

Type

Gauge

Label

Flame level

Value format

{{value}}

Units

units

Range

min

0

max

100

Colour gradient

Sectors

0

...

optional

...

optional

...

100

Name

Fig 8 : Properties of Gas level node

Edit gauge node

Delete

Cancel

Done

Properties

Group

[control] Weather Monitoring

Size

auto

Type

Gauge

Label

Gas level

Value format

{{value}}

Units

units

Range

min

0

max

10

Colour gradient

Sectors

0

...

optional

...

optional

...

10

Name

Enabled

Fig 9 - Properties of IBM IOT are shown. The API key, Device Type, Device ID are taken from IBM IOT WATSON PLATFORM.

## Edit function node

Delete

Cancel

Done


### Properties



 Name

Temperature node



 Setup

On Start

**On Message**

On Stop

```
1 msg.payload=msg.payload.Temperature
2 global.set("t",msg.payload)
3 return msg;
```





Edit function node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔗

📌 Name

Flame level

📄

⚙️ Setup

On Start

On Message

On Stop

1 msg.payload=msg.payload["Flame level"]

2 global.set("f",msg.payload)

3 return msg;

○ Enabled

Edit function node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔗

📌 Name

Gas level node

📄

⚙️ Setup

On Start

On Message

On Stop

1 msg.payload=msg.payload["Gas level"]

2 global.set("g",msg.payload)

3 return msg;

○ Enabled

**Fig 10** - Properties of HTTP request with method GET and url control

The image shows a software interface for editing an HTTP node. The title bar reads "Edit http in node". Below the title bar are three buttons: "Delete", "Cancel", and "Done". The "Done" button is highlighted in red. The main area is titled "Properties" and contains three fields: "Method" with a dropdown menu set to "GET", "URL" with a text input field containing "/control", and "Name" with a text input field containing "Name". At the bottom left, there is a checkbox labeled "Enabled" which is checked.

**Edit http in node**

Delete Cancel Done

**Properties**

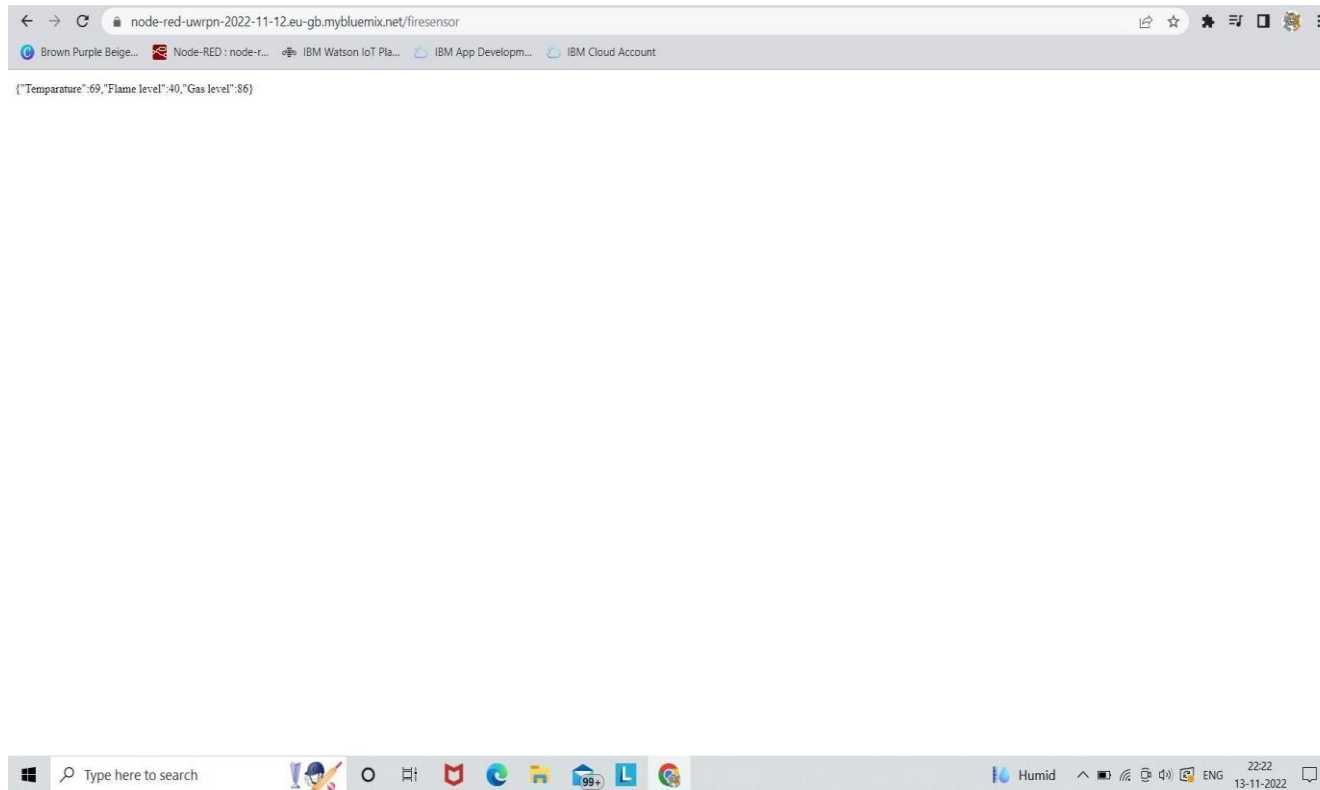
Method GET

URL /control

Name Name

Enabled

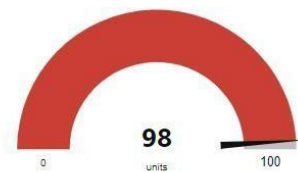
**Fig 11 : HTTP request OUTPUT**



## control

### Weather Monitoring

Flame level



Gas level



EXHAUST FAN ON

EXHAUST FAN OFF

SPRINKLER ON

**FIG 12 :** Front -end APP for our project to display the temperature ,smoke level and flame level with control buttons like Sprinkler ,exhaust fan on and off buttons

