

## Project Development Phase

### Sprint Delivery – III

PROJECT TITLE	Gas Leakage Monitoring and Alerting System
TEAM ID	PNT2022TMID06977

#### Introduction:

In this sprint delivery - III, we have completed creating cloudant database service in IBM cloud and we created a login page and then we created a Home Page for the users and finally, we created the Gas Reading page to see the Readings of the Gas Sensor and Also, created Valve Control Page which provides switches to control the valves of the gas pipeline.

The gas sensor data is fetched by subscribing to the device in the IBM Watson IOT platform in accordance with the MQTT protocol and the valve control operation is published back to the device in the IBM Watson IoT platform. Also, we created API Endpoint to make Mobile Application to make use of the sensor data and Current Valve State.

#### Preinstallation:

Install Node-red-dashboard to create a UI Components for the application.

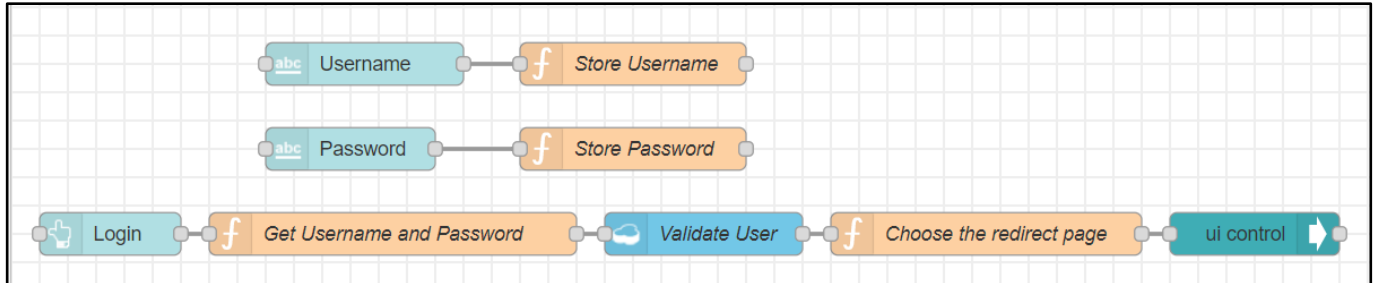
#### USER STORY/TASK 11 and 12: Creating Login Page:

##### Node-Red Steps:

1. Create a Dashboard Tab Login Page in which a create a Dashboard Group called Hello, User.
2. Now, drag a text input Node into the flow and rename it as Username and design its properties to get Username as input from the user.
3. Connect the Username Text input Node with a function which stores the user input into a global variable
4. Now, drag a text input Node into the flow and rename it as Password and design its properties to get Password as input from the user.
5. Connect the Username Text input Node with a function which stores the user input into a global variable
6. Now, drag a button node and rename it as Login and connect it with a new function which gets the global values Username and password to the next Node
7. Now, drag a Cloudant IN Node into the flow, and add your cloudant credential into the server in the properties of the Cloudant IN Node.
8. Connect the function with the Cloudant in to connect to the Cloudant Service and get the return in a function.

9. The function checks the return is true or false and sends the correct tab to be changed based on the User validation
10. Now, drag a Ui\_control Node into the flow and connect the function to it which redirects it to correct Page.

#### Node-Red Flow:



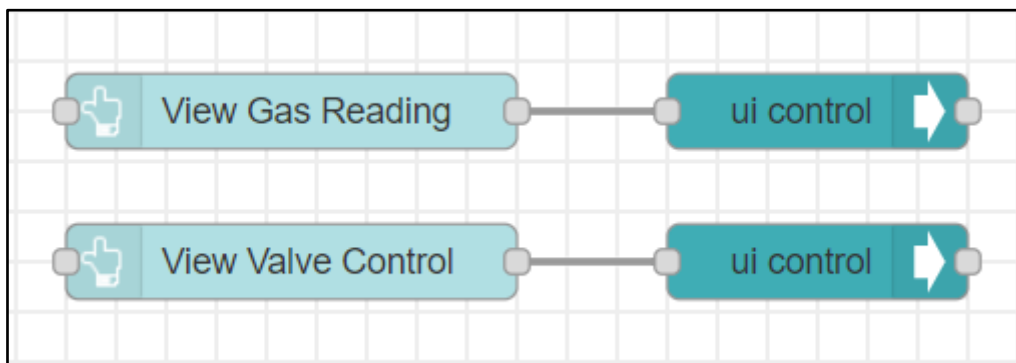
Login page Nodes Connections

#### USER STORY/TASK 13: Creating Home Page:

##### Node-Red Steps:

1. Create a Dashboard Tab Home Page in which a create a Dashboard Group called Welcome, User.
2. Now, drag a button into the flow and rename it as View Gas Reading and place it in the Welcome, User Dashboard group and design its properties which returns Gas Reading Tab when clicked.
3. Now, drag a ui\_control node into the flow and which is connected to the View Gas Sensor Reading button to redirect to Gas Reading Tab When the button is clicked.
4. Now, drag a button into the flow and rename it as View valve Control and place it in the Welcome, User Dashboard group and design its properties which returns Valve Control Tab when clicked.
5. Now, drag a ui\_control node into the flow and which is connected to the View Valve Control button to redirect to Valve Control Tab When the button is clicked.

#### Node-Red Flow:



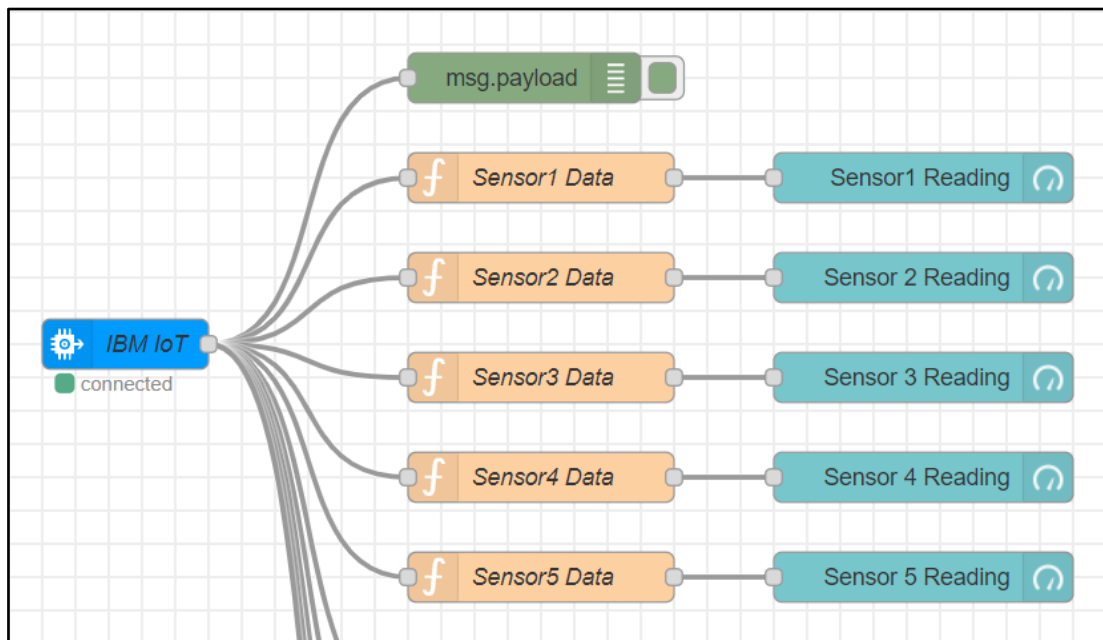
Home Page Node connections

## USER STORY/TASK 14 and 15: Creating Gas Reading Page:

### Node-Red Steps:

1. Create a Dashboard Tab Gas Reading Page in which a create a Dashboard Group called Gas Sensor values
2. Now, drag an IBM IoT IN Node into the flow and edit its property and enter your IBM Watson IoT Platform Device Credential which is created in the previous part of the project. On Deploy, the Node must be stay connected to the Device.
3. Now drag a Debug node into the flow which is connected to the IBM IoT IN Node to print the incoming Json format data from the IBM Watson IoT Platform Device
4. Now, insert 5 functions at the out of IBM IoT IN Node to...
  - a. To fetch the Sensor Reading from the Json Format message
  - b. Send those values as message Payload
  - c. Declare those values as Global for future Usage.
5. Now, insert 5 Gauge Node into the flow which are connected to the 5 functions respectively and visualize the data sent by the function.

### Node-Red Flow:



Gas Reading Page Node connections

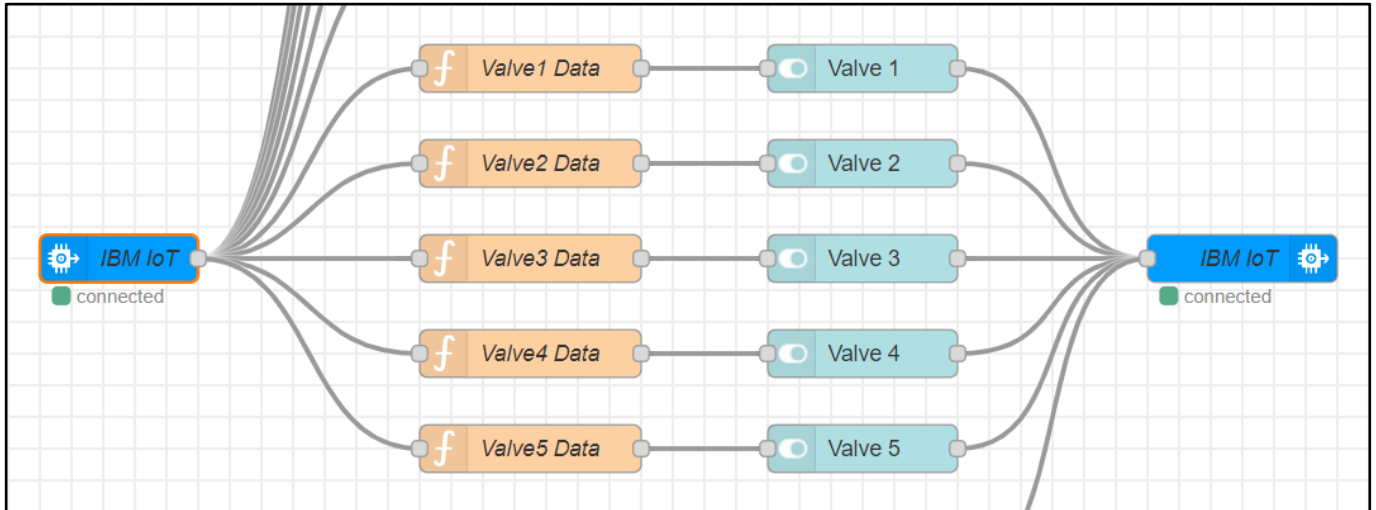
## USER STORY/TASK 16 and 17: Creating Valve Control Page:

### Node-Red Steps:

1. Create a Dashboard Tab Valve Control Page in which a create a Dashboard Group called Valves
2. Now, insert 5 functions at the out of IBM IoT IN Node to...
  - a. To fetch the Current Valve state from the Json Format message
  - b. Send those values as message Payload in defined format at the Dashboard Switch Node
  - c. Declare those values as Global for future Usage.
3. Now, drag 5 Dashboard Switch into the flow, and connect them to the respective functions and define its properties as...
  - a. Display the Input Data as the Dasboard Switch State.
  - b. Send json fomate data as output when clicked to change the current state.
  - c. Define the ON and OFF State Payload to send when clicked.

- Now, drag an IBM IoT Out Node into the flow and edit its property and enter your IBM Watson IoT Platform Device Credential which is created in the previous part of the project. On Deploy, the Node must be stay connected to the Device.
- Connect Output of all 5 Switches to the IBM IoT Out Node to send the User Valve input to the IBM Watson IoT Device.

#### Node-Red Flow:



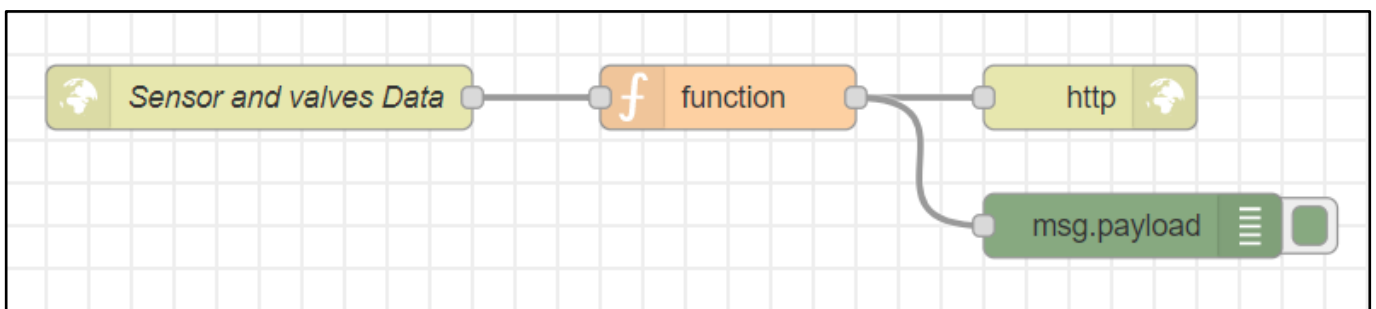
Valve Control Page Node connections

#### USER STORY/TASK 18: Creating sensorData Endpoint:

##### Node-Red Steps:

- Now, drag a http in Node into the flow and create an API Endpoint which is used to send the sensor data and the current Valve State whenever it is called and name the Endpoint.
- Now drag a function which takes input from the http in and creates the msg.payload that contains the current Sensor value and the Current State of the Valve which are obtained from the stored global variable and combined into a neat Json Format to send to the mobile Application.
- Now, drag a Debug Node into the flow, and Connect a debug Node to print the msg.payload to verify the Json Format and the message before sending it in the response body to requesting Mobile Application.
- Now, drag a http response Node into the flow, and connect the function to it to complete the API Request with a success code and response body containing the requested data to the mobile Application

#### Node-Red Flow:

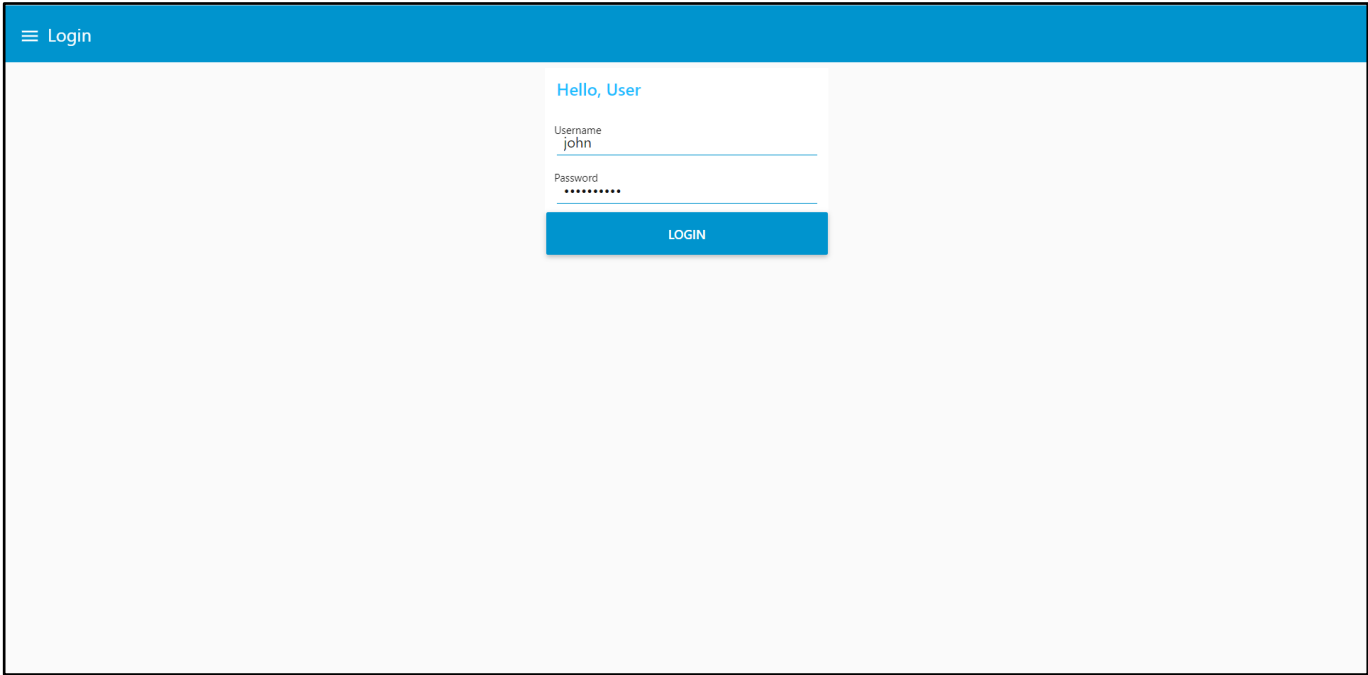


sensorData Endpoint Node Connections

Output of Sprint 3:

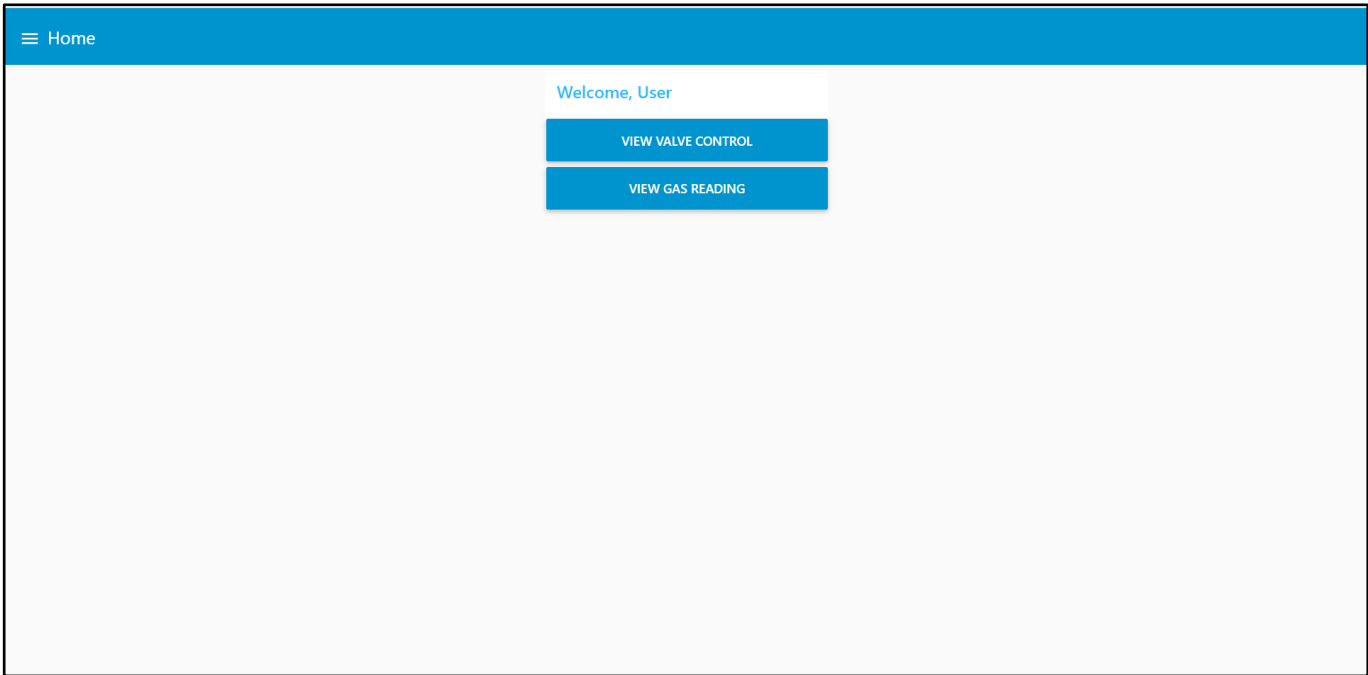
Screenshots of Web Application:

Screenshot of Login Page:



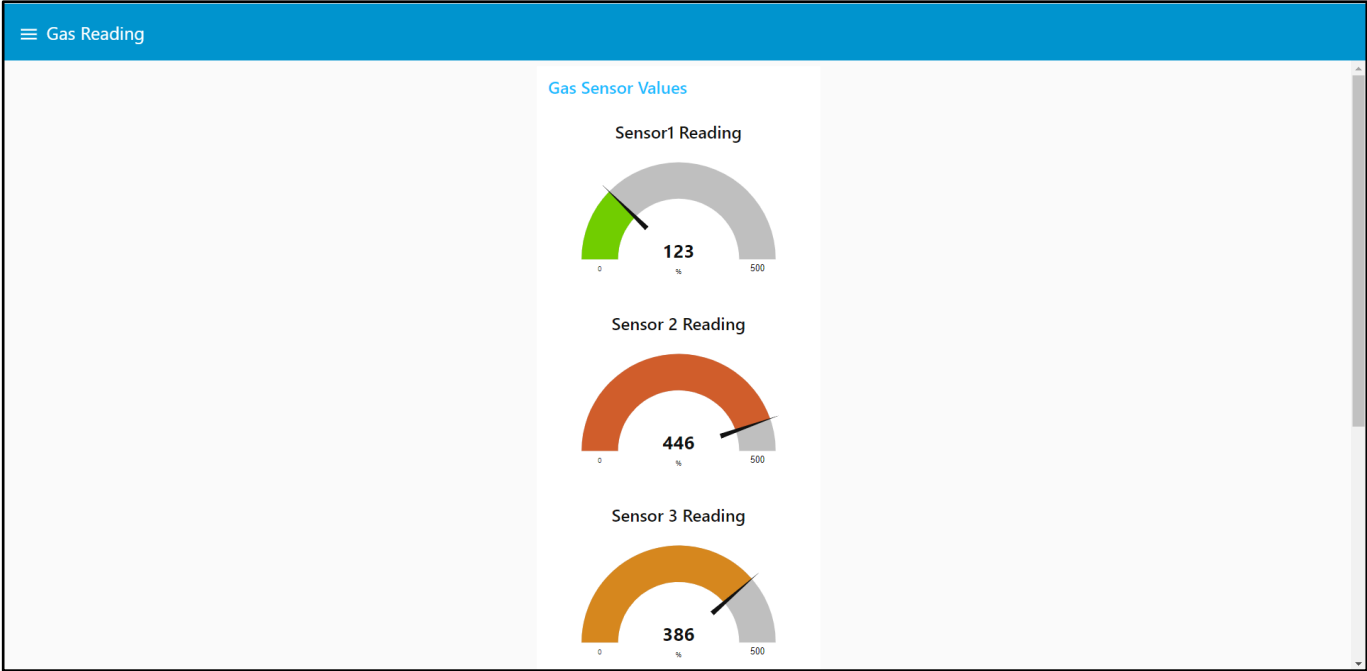
Login Page

Screenshot of Home Page:



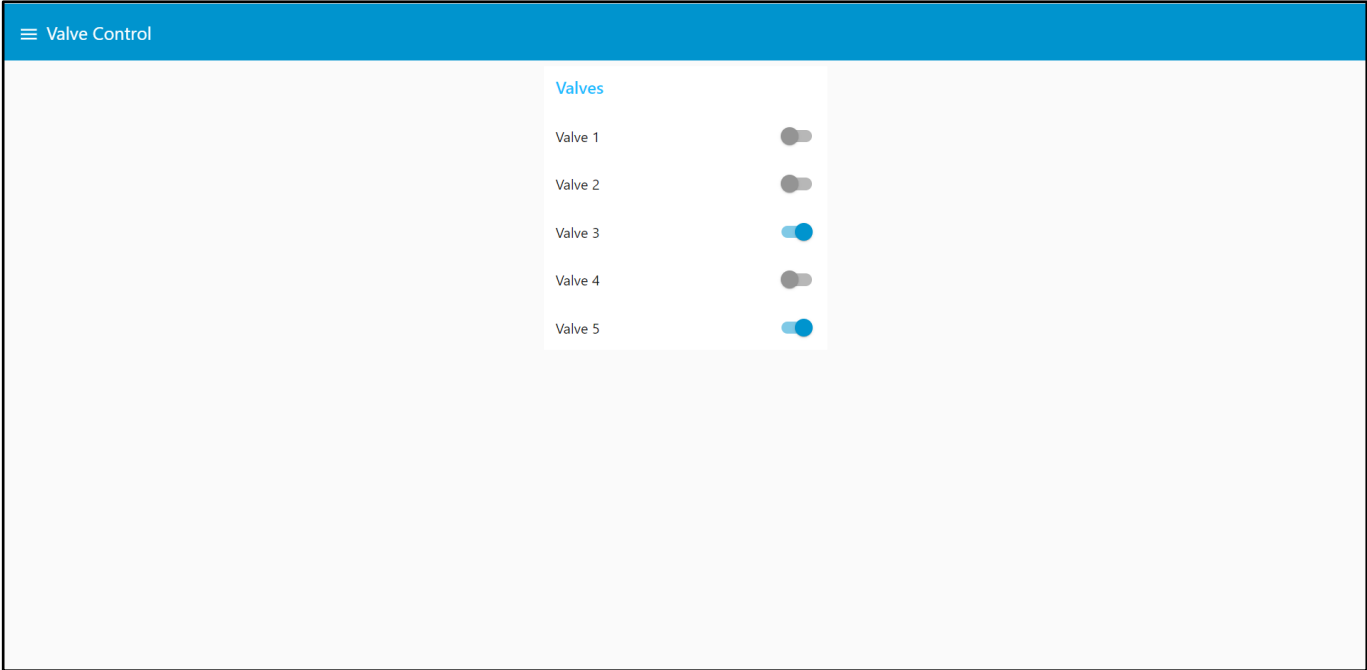
Home Page

Screenshot of Gas Reading Page:



Gas Reading Page

Screenshot of Valve Control Page:



Valve Control Page

**Project Link:**

Link of the Json File which Creates this Flow:

<https://github.com/IBM-EPBL/IBM-Project-2995-1658493718/blob/main/Project%20Development%20phase/Sprint%20-%203/flows.json>

Link of the Web Application UI: <http://159.122.177.185:31086/ui>

**Demo Link:**

Link of the Demo Video:

[https://drive.google.com/drive/folders/1ZesOrCqdZJQKc8DVe00X15nj0hnsbTc\\_?usp=sharing](https://drive.google.com/drive/folders/1ZesOrCqdZJQKc8DVe00X15nj0hnsbTc_?usp=sharing)