

Project Development Phase

Sprint Delivery – IV

PROJECT TITLE	Gas Leakage Monitoring and Alerting System
TEAM ID	PNT2022TMID06977

Introduction:

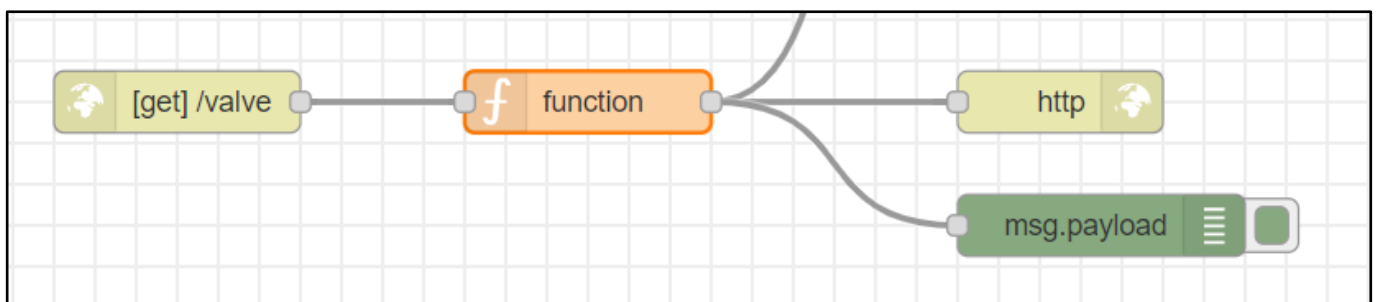
In this sprint delivery – IV, we have completed creating a Mobile Application using MIT App Inventor which communicates with the Node-Red for Backend. It has a login page, a Main page to display Sensor Data and Current Valve State and a page to control Valves of the Gas Pipeline. Also, in this Sprint we have created API Endpoints in Node Red to make the mobile Application to Communicate with the Backend.

USER STORY/TASK 19: Creating valve Endpoint:

Node-Red Steps:

1. Now, drag a http in Node into the flow and create an API Endpoint which is used to accept the update of valve state whenever it is called and name the Endpoint.
2. Now drag a function which takes input from the http in and creates the msg.payload according to the needed format to publish the data in IBM Watson IoT Platform Device.
3. 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 publishing them to Device in the IBM Watson IoT Platform.
4. Now, drag a http response Node into the flow, and connect the function to it to complete the API Request with a success code to the mobile Application
5. Connect this function to the already created IBM IoT Out Node and make the data to be published to the Device in the IBM Watson IoT Platform.

Node-Red Flow:



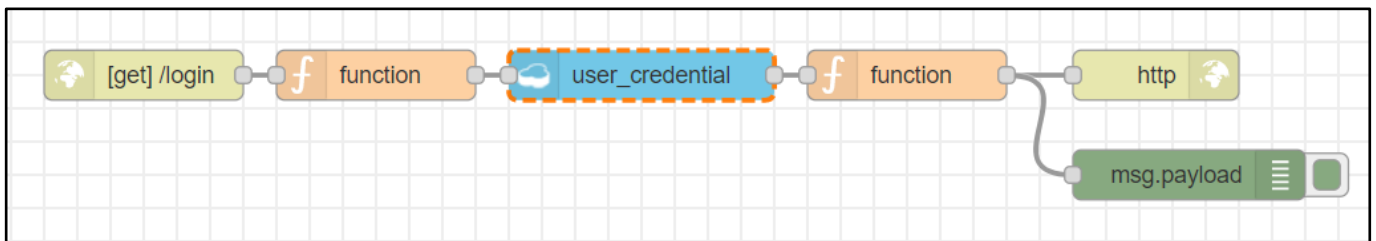
Valve Endpoint Node Connections

USER STORY/TASK 20: Creating Login Endpoint:

Node-Red Steps:

1. Now, drag a http in Node into the flow and create an API Endpoint which is used to receive the login credential from the Mobile Application and name the Endpoint.
2. Now drag a function which takes input from the http in and creates the msg.payload that contains the username and password from the http In Node combined into a neat Json Format to send to the Cloudant Database.
3. Now, drag a Cloudant In Node into the Flow and search the data in the Database and receive the result in a function and create a msg.payload to send the result of Login Attempt.
4. 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 result of the Login Attempt to the Mobile Application

Node-Red Flow:



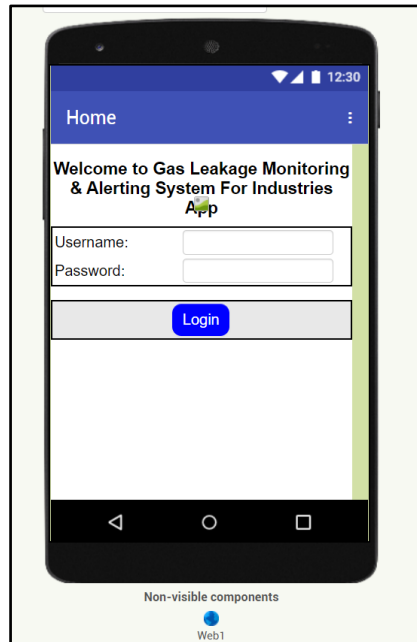
Login Endpoint Node Connections

USER STORY/TASK 21: Creating Login Page

MIT App Inventor Steps for Design:

1. Create a Screen for the login Page and change adjust the properties to change the design to make simple Login page
2. Now, drag an image to Screen and change its properties to paste a neat image for the app
3. Now, drag a Label to Screen and change its properties to make a Username label.
4. Now, drag a Text Input to Screen and Change its properties to make user to enter the Username.
5. Now, drag a Label to Screen and change its properties to make a Password label.
6. Now, drag a Text Input to Screen and Change its properties to make user to enter the Password.
7. Now, drag a Button to the Screen and Change its properties to create a login Button.
8. Now, drag a Web Component into the Screen to make API Endpoint Request to Backend.

MIT App Inventor Design



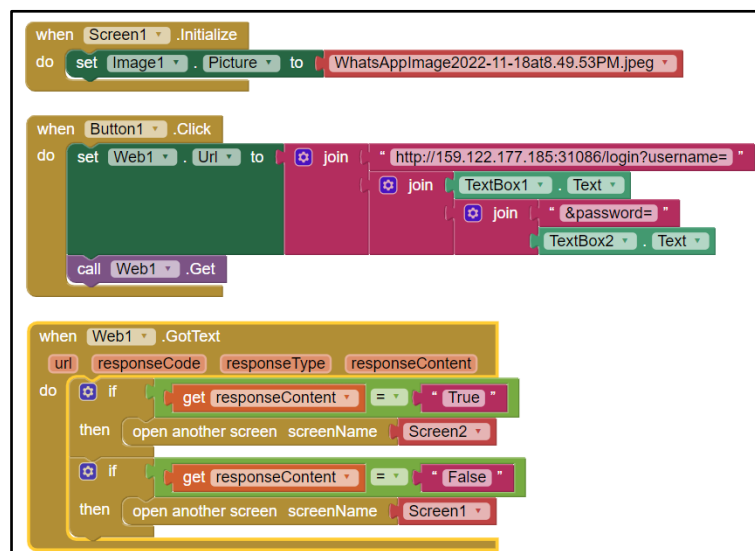
Login Page Design in MIT App Inventor

USER STORY/TASK 22: Creating Login Page

MIT App Inventor Steps for Blocks:

1. Drag a Block to set image with the Gas Pipe Industry Image when the Screen is initialized.
2. Drag a Block that takes action when the login Button is Clicked.
3. When the Login Button is clicked it takes the input in the username and password input field and make a API Request to the Endpoint to verify the Credential.
4. Drag a block to receive the response of the endpoint and redirect to the home page of the reponse is positive.

MIT App Inventor Blocks:



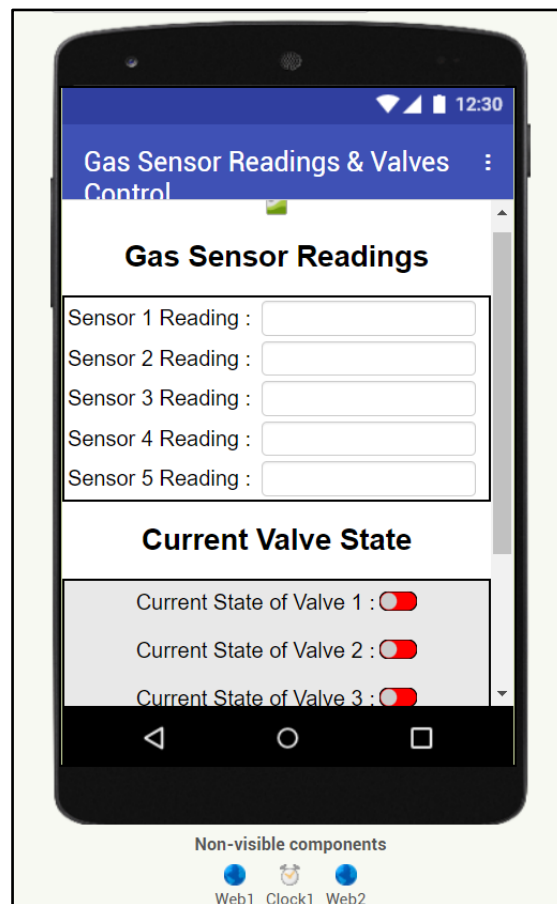
Login Page Blocks in MIT App Inventor

USER STORY/TASK 23: Creating Home Page

MIT App Inventor Steps for Design:

1. Create a Screen for the login Page and change adjust the properties to change the design to make simple Login page
2. Now, drag an image to Screen and change its properties to paste a neat image for the app.
3. Now, drag a Label to the Screen and change its properties to make neat Heading as Gas Sensor Reading.
4. Now, drag 5 Labels to the Screen to label the 5 Sensor Values and change its properties.
5. Now, drag 5 Text box to the Screen to display 5 Sensor reading fetched from the Backend and change its properties.
6. Similarly, drag a Label to the Screen and Change its properties to make a neat heading as Current Valve State.
7. Similarly, drag 5 Switches to the Screen to display the current state of the valve and change its properties.
8. Now, drag a button to the Screen and change its properties to make Update valve State.
9. Now, drag 2 Web Components into the Screen to make API Endpoint Request to Backend.
10. Now, drag a clock Components into the Screen to make repeated request within the track of time to Backend.

MIT App Inventor Design



Home Page Design in MIT App Inventor

USER STORY/TASK 24: Creating Home Page

MIT App Inventor Steps for Blocks:

1. Drag a Block to set image with the Gas Pipe Industry Image when the Screen is initialized. And, set all 5 Switches to be Disabled when the Screen is initialized.
2. Drag a Block to set Timer and make Repeated API Request to backend at regular interval of 1 second to the Backend to fetch the Data from the Node-Red.
3. Drag a block to receive the response of the endpoint and read the Json Format in the response body and set the Textbox with sensor value that is read and Also Set the Switch State based on the read response content.
4. Drag a Block that takes action when the Update Valve State Button is Clicked which redirects the current Screen to the Update Valve State page.
5. Drag a Block that takes action when the back Button is pressed which redirects the screen to the Login Page.

MIT App Inventor Blocks:

The image shows four MIT App Inventor blocks:

- when Screen2.Initialize**
 - do set Image1.Picture to WhatsAppImage2022-11-18at8.49.53PM.jpeg
 - do set Switch1.Enabled to false
 - do set Switch2.Enabled to false
 - do set Switch3.Enabled to false
 - do set Switch4.Enabled to false
 - do set Switch5.Enabled to false
- when Clock1.Timer**
 - do set Web2.Url to http://159.122.177.185:31086/sensorData
 - do call Web2.Get
- when Button1.Click**
 - do open another screen screenName Screen3
- when Screen2.BackPressed**
 - do open another screen screenName Screen1

The image shows a large MIT App Inventor block for the **when Web2.GotText** event. It contains the following logic:

- do** set TextBox1.Text to look up in pairs key "Sensor1" pairs call Web2.JsonTextDecode jsonText get responseContent
- if** look up in pairs key "Valve1" pairs call Web2.JsonTextDecode jsonText get responseContent = "Opened"
 - then** set Switch1.On to true
 - else** set Switch1.On to false
- do** set TextBox6.Text to look up in pairs key "Sensor2" pairs call Web2.JsonTextDecode jsonText get responseContent
- if** look up in pairs key "Valve2" pairs call Web2.JsonTextDecode jsonText get responseContent = "Opened"
 - then** set Switch2.On to true
 - else** set Switch2.On to false
- do** set TextBox3.Text to look up in pairs key "Sensor3" pairs call Web2.JsonTextDecode

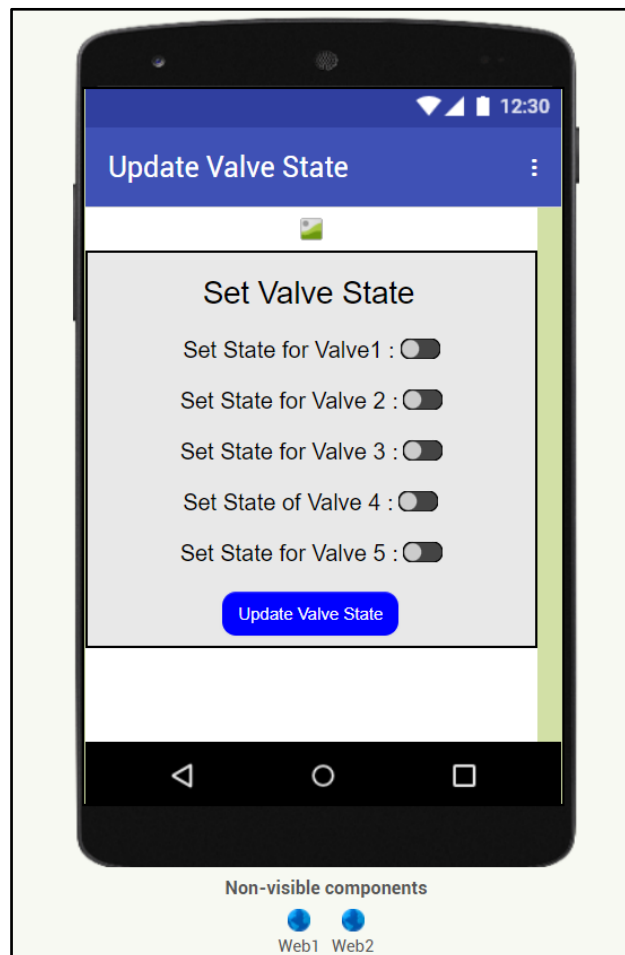
Home Page Blocks in MIT App Inventor

USER STORY/TASK 25: Creating Update Valve State Page

MIT App Inventor Steps for Design:

1. Create a Screen for the update valve state Page and change adjust the properties to change the design to make simple update valve state page.
2. Now, drag an image to Screen and change its properties to paste a neat image for the app.
3. Now, drag a Label to the Screen and change its properties to make neat Heading as Set valve State.
4. Now, drag 5 Switches to the screen to display the current state and get state input from the user and change its properties to make a neat look.
5. Now, drag a button into the screen and rename it as Update Valve State and change its properties for a good look.
6. Now, drag 2 Web Components into the Screen to make API Endpoint Request to Backend.

MIT App Inventor Design:



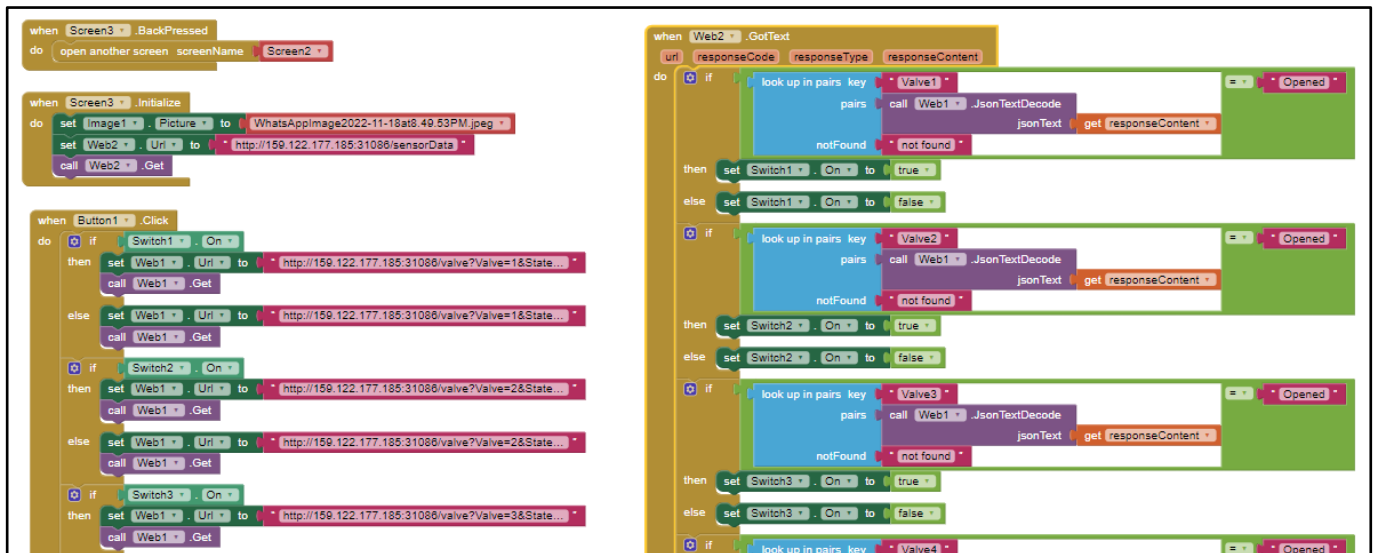
Update Valve State Design in MIT App Inventor

USER STORY/TASK 26: Creating Home Page

MIT App Inventor Steps for Blocks:

1. Drag a Block to set image with the Gas Pipe Industry Image when the Screen is initialized and make API Request to Node-Red Backend to get the current state of the 5 Switches.
2. Drag a block to receive the response of the endpoint and read the Json Format in the response body and Set the Switch State based on the read response content.
3. Drag a Block that takes action whenever the Update Valve State Button is Clicked which makes API Request to Node-Red Backend to update the State inputted by the user in the 5 Switches.
4. Drag a Block that takes action when the back Button is pressed which redirects the screen to the Home Page.

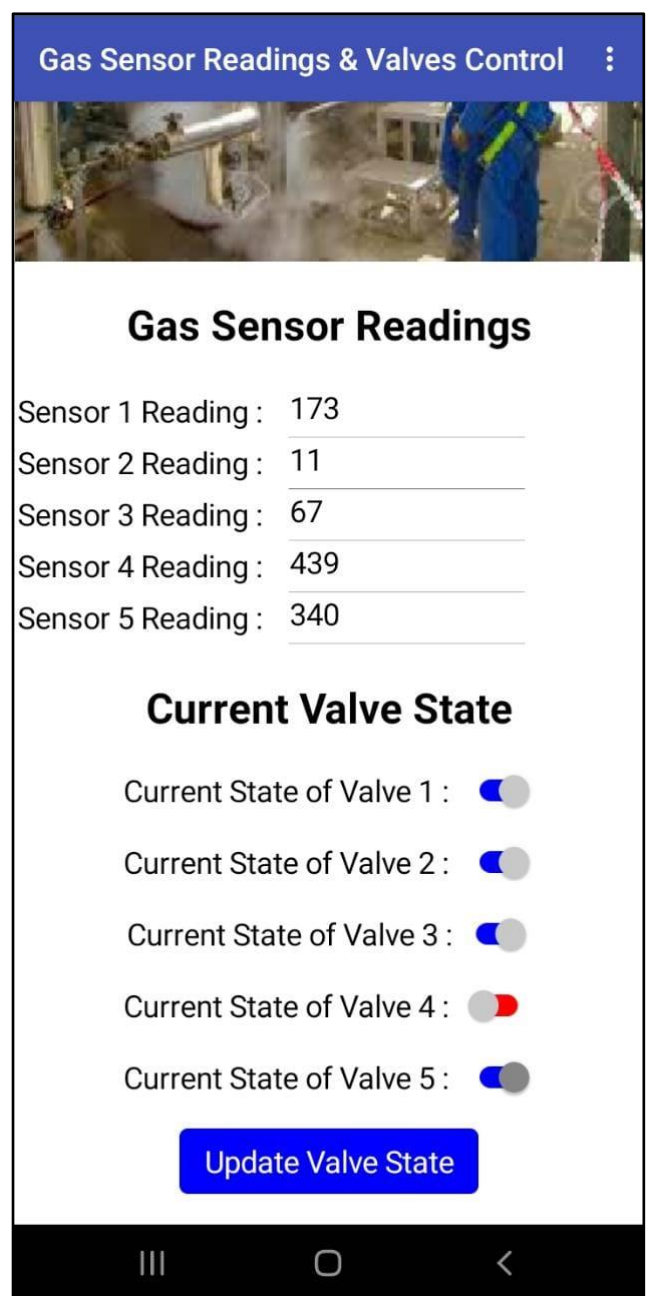
MIT App Inventor Blocks:



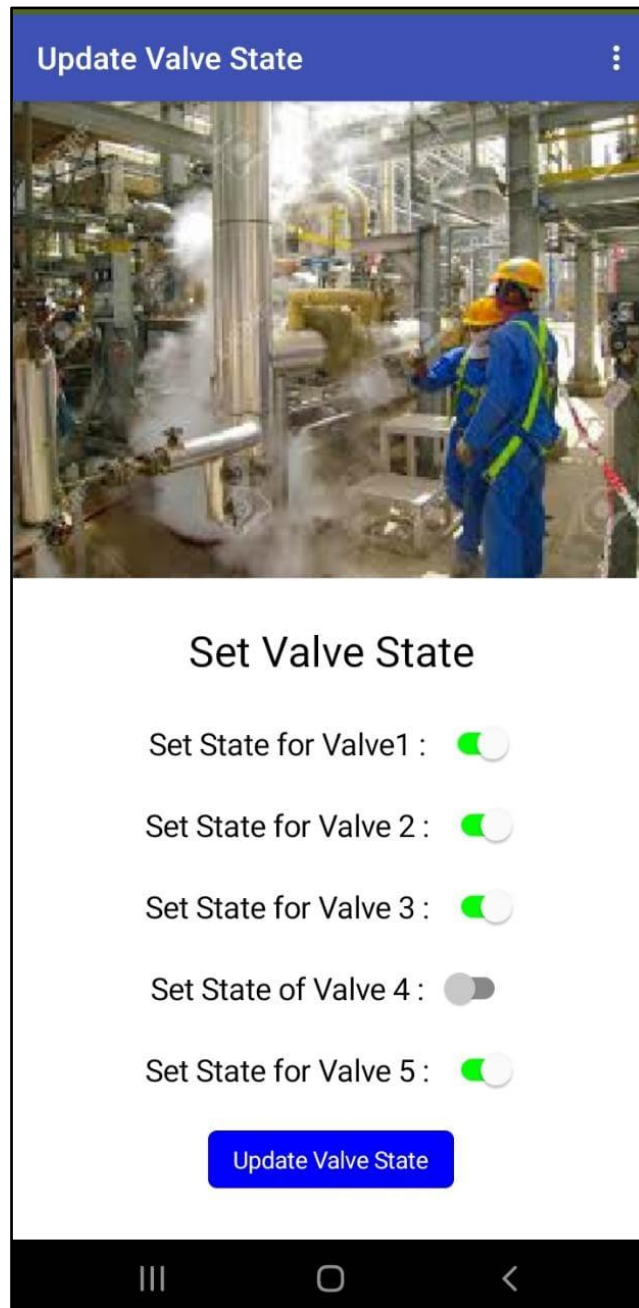
Update Valve State Blocks in MIT App Inventor

Output of Mobile Application:

Screenshot of Login Page and Home Page:



Screenshot of Update Valve State Page:



Project Link:

Link of the .apk File of this Mobile Application:

https://github.com/IBM-EPBL/IBM-Project-2995-1658493718/blob/main/Project%20Development%20phase/Sprint%20-%204/PNT2022TMID06977_Nalaiya_Thiran.apk

Demo Link:

Link of the Demo Video of the Project:

<https://drive.google.com/drive/folders/1ZesOrCqdZJQKc8DVe00X15nj0hnsbTc?usp=sharing>