CAB230 ASSIGNMENT 2 CLIENT-SIDE

Volcano API – Client-Side Application

Samuel van Zuylen
N11078472
05/05/2024

Contents

Introduction

Purpose & Description

The aim of this project is to develop a web application using React.js that provides users with information on over 1,300 different volcanoes from around the world. This information is sourced from a REST API that contains a large database of volcano data. The application allows users to easily search for volcanoes by country and offers a simple and intuitive interface to view specific information such as the volcano's name, location, summit, elevation, and even the population density of the surrounding area.

What sets this application apart is its thoughtful approach to user experience and responsive design. The navigation is straightforward, allowing users to move seamlessly from the home page to the volcanoes page, where they can find a search bar for exploring volcanoes by country. The search functionality includes an autocomplete feature, making it easier for users to select countries that match their input. Additionally, users can refine their search by setting a distance based on population density, to limit their search to those that are close to civilisation.

Some access to volcano data is limited, thus introducing the need for an authentication service. This is also incorporated into the website and uses the same REST API to send and receive data on users' account information in order to authenticate users before displaying this non-public information.

Completeness & Limitations

The app successfully implemented several key features that contribute to a user-friendly experience. The overall simplistic design and layout, combined with a responsive component, allows for the application to be accessible on a variety of devices with different screen resolutions and aspect ratios. React router is used to provide a flexible and efficient way of handling navigation and routing within the application, which is used to simulate pages without requiring full page reloads. Controlled forms were used in key components such as the sign up / login pages, as well as the search bar system for the volcanoes table component. The volcano page was fully implemented to include an interactive map component that displays the location of the volcano on the world map with a marker. This page also contains a bar chart to visually display the population density of the surrounding area of the volcano.

All features that were included in the original design were effectively implemented into the application in a way that maximises both functionality and user experience. However, a noteworthy limitation of the application is the potential for many extra features that could've been added to enhance user experience. Some extra features were considered such as the use of more external API's such as a google images API which could be used to display an image of each volcano. Many more features were contemplated however it was concluded that adding extra features could've 'cluttered' the website and potentially taken away from the core purpose and simplicity of the application, leading to a less user-friendly experience.

Use of End Points

/countries

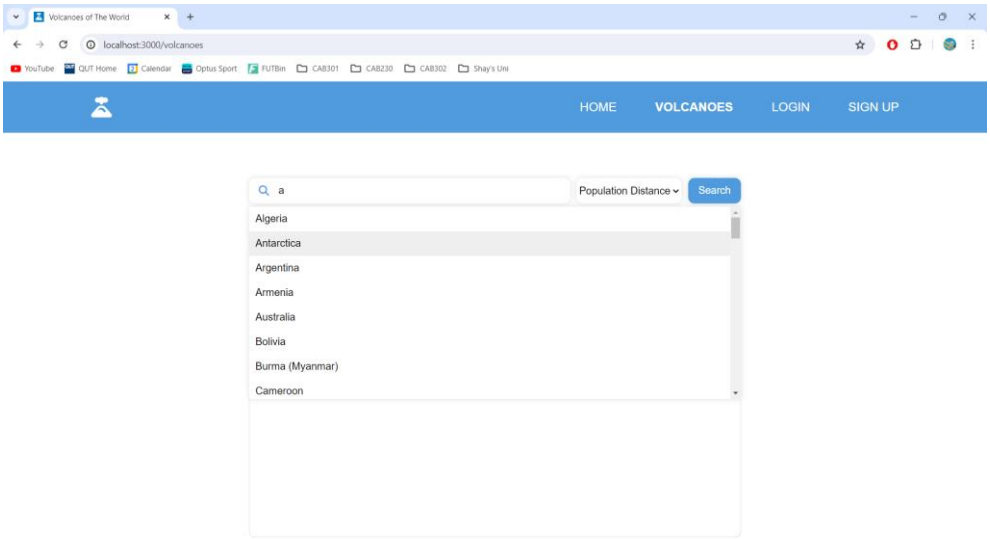

Figure 1 - Country Endpoint Usage

The /country endpoint is used to retrieve country name data. It is called when the user searches for a country. The response body includes a list of country names, which are then sorted based on the user's input into the search bar and are displayed in the dropdown list as shown in Figure 1.

/volcanoes

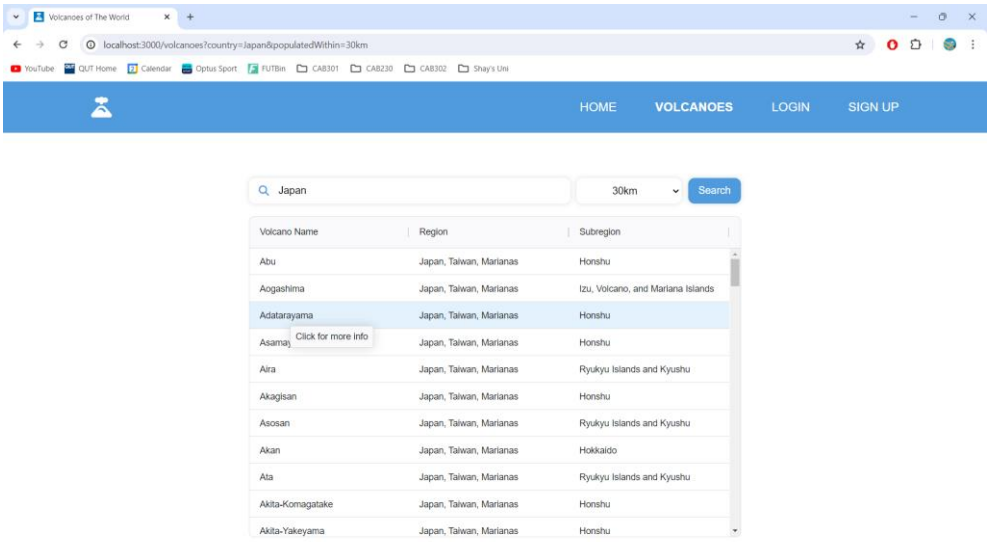

Figure 2 - Volcanoes Endpoint Usage

The /volcanoes endpoint is accessed once the user has clicked on the search button. The fetch request is sent to the API with the country name and population distance (if specified). The response body contains a list of volcanoes that match the search parameters, and the information is then displayed in a table for the user to browse as seen in Figure 2.

/volcano{id}

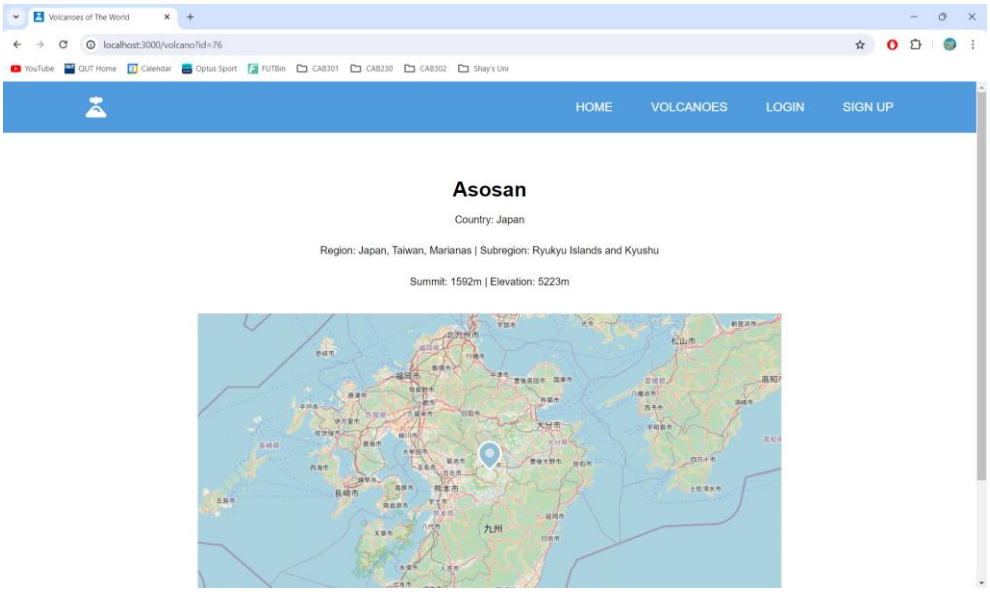

Figure 3 - Volcano{ID} Endpoint Usage

When the user clicks on a volcano from the table list, a new page is opened, which sends an API request to the /volcano endpoint with the ID of the selected volcano. If the user has been authenticated (logged in), then the request body also includes an authentication token. The response data on that volcano such as the name, region, and location is then used to display various components to the user, which will be explained in more detail.

/user/register

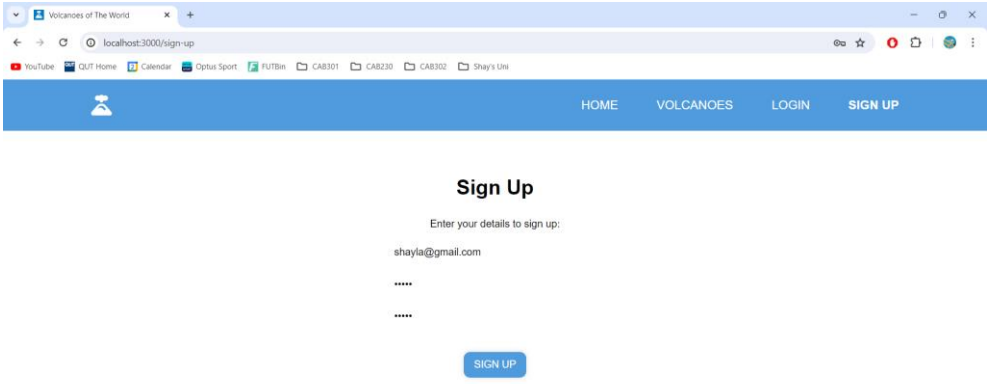

Figure 4 - User/Register Endpoint Usage

When a user attempts to sign up to the website, a POST call is sent to the /user/register endpoint to create a new user account. The body of the request includes an email and a password, which will be used to identify the user. The response object includes a message stating whether the action was successful or not. This sign-up form is shown in Figure 4.

/user/login

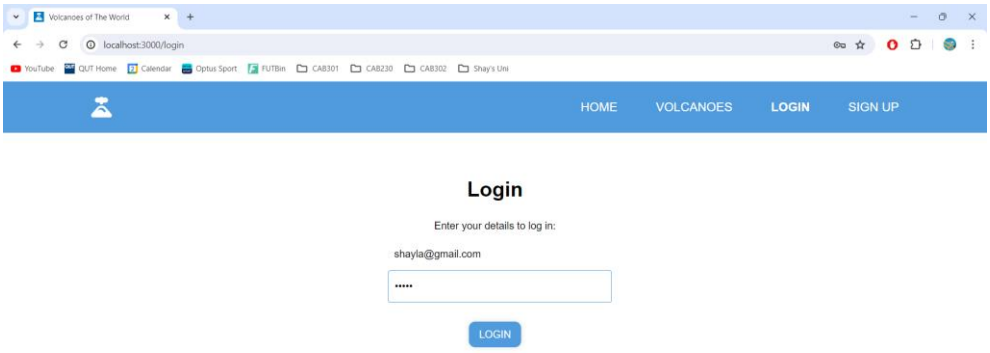

Figure 5 - User/Login Endpoint Usage

Once the user has successfully signed up, they can use that same information to log in and be authenticated. The email and password are sent off in the POST request body to the /user/login endpoint where the user account is verified. The response body indicates whether the sign in action was successful, and if so, a token is also provided which is stored in local storage for future use. See Figure 5 for the login form.

Modules Used

Ag-Grid-React

A data grid component for React that allows developers to implement complex data tables with features like sorting, filtering, grouping, and more.

Documentation: https://www.ag-grid.com/react-grid/

React-Router

A routing library for React applications that enables developers to create single-page applications with multiple navigation features.

Documentation: https://reactrouter.com/en/main

React-Chart-JS

A library that allows you to create various types of charts in React applications. It supports a wide range of chart types, including line, bar, pie, doughnut, and provides customization options for chart appearance and behaviour.

Documentation: https://react-chartjs-2.js.org/

Pigeon-Maps

A lightweight map component for React applications that provides a simple API for integrating maps into React applications, with features like custom markers, overlays, and interactions.

Documentation: https://pigeon-maps.js.org/docs/

Application Design

Navigation & Layout

A key design consideration that needed to be made was deciding how to handle navigation between application pages / views. It was important to make the navigation system simple, user-friendly, and responsive to be usable on multiple device displays such as monitors, iPads, phones, etc. Figure 6 shows a design sketch highlighting the layout of the responsive header navbar to be used, which can support both wide and narrow-screen devices.

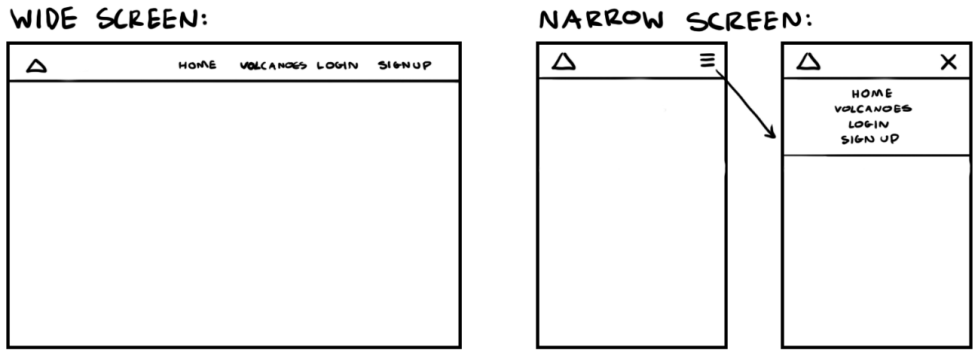

Figure 6 - Responsive Navigation Design Sketch

Including a footer as well as a header for navigation purposes is a common practice when building websites and was also considered. However, it was deemed to be unnecessary for this application as no pages will be large enough to require it. If the application was expanded to include more information on each page to the point where scrolling is required by the user, then a footer with similar navigation links should be implemented.

The next design step was to plan out the overall structure of each page, and the links between them. It was established that the home, volcanoes, login, and sign-up pages were to be included in the navbar, with the volcano icon (top-left) also functioning as a link to the home page. The page which shows the information in relation to a volcano with a specific ID, can only be accessed by clicking on the rows in the table once the user has searched for them.

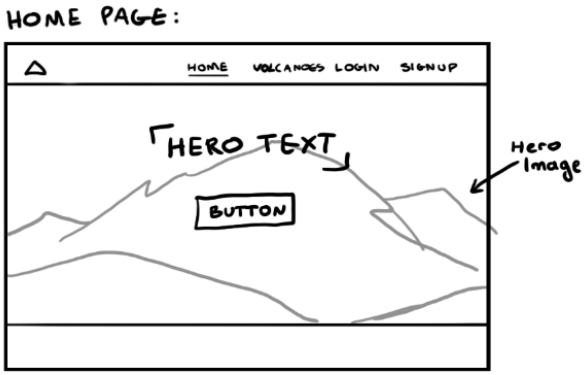

Figure 7 - Home Page Design Layout Initial Sketch

A very common practice for professional websites is to include a 'Hero Image' on the front page of your website. The purpose of this is to stand out and be memorable, whilst also making it clear to the user the exact purpose of the website. This design layout for the home page of the website is shown in Figure 7. When searching online for interesting volcano images, the following two photographs of Mount Bromo in Indonesia were identified as potential hero-image worthy shots.

Figure 8 - Proposed Hero Image Photographs

The sign-up and login pages were very simple designs, as this is a very common requirements of websites and applications of all types, and there is already a highly universal layout. There will simply be a form that includes a message telling the user to enter their details, followed by a text field for an email, and a text field for a password in which the characters are hidden when the user types for privacy. The sign-up page will also include a second password field for the user to re-enter their password to ensure they have not made any mistakes.

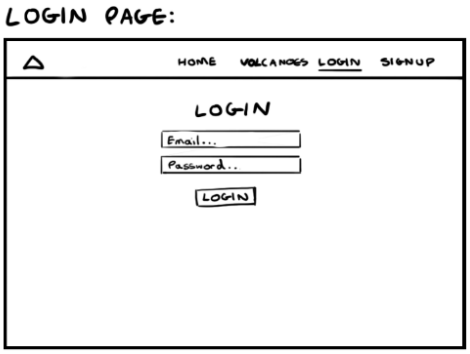

Figure 9 - Login Page Design Layout Initial Sketch

On the volcanoes page, it was quickly established that the best method of displaying the volcano data was in a table, as there was to be a long list of volcanoes with their names, region, and subregion to be displayed. Figure 10 shows this table layout.

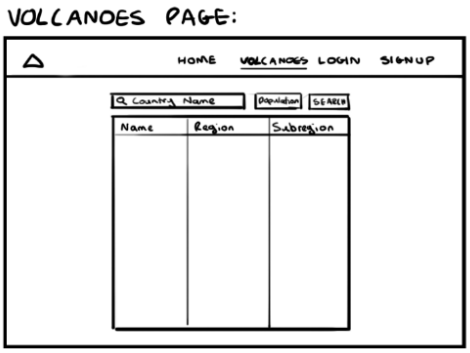

Figure 10 - Volcanoes Page Design Layout Initial Sketch

There were however many considerations to be made when it came to the method for searching for these volcanoes. As per the API request, the user would be required to select a country, and optionally a population distance of four set distances. As the population distance only has four specified options, a dropdown menu was the best option for this selection. However, the country name was more complicated due to the large number of options. A search was an obvious choice, although that would require the user to manually input a country name without any spelling mistakes or the API request would fail. With these two things in mind, a search bar with an autocomplete dropdown-like component was the best method.

The layout of the search form component was also considered, as it would need to be compatible with multiple device screens. The following design approach was made to accommodate both wide and narrow screens:

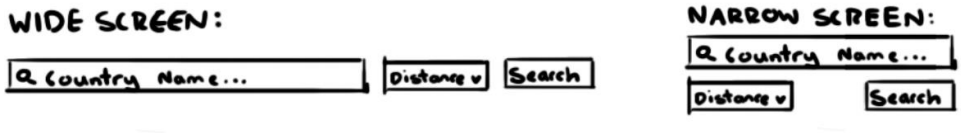

Figure 11 - Volcano Search Bar Layout Sketch

The volcano page required three main components; an information section (text), interactive map, and a graph to display population density data. The simplest and cleanest way to present these components was to lay them out vertically in the centre of the page, as this would create clear separation between components and be easily adaptable to both wide and narrow screens. This layout is shown in the design sketch in Figure 12.

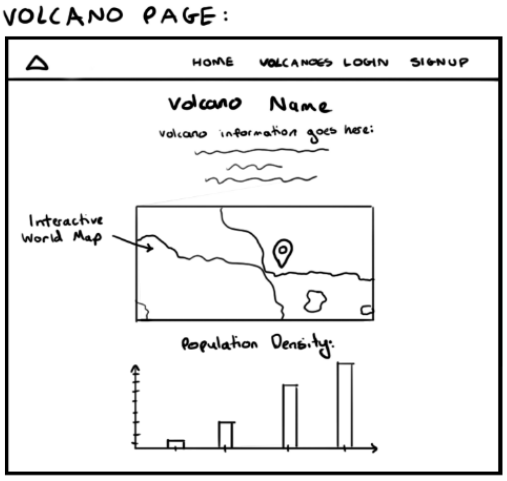

Figure 12 - Volcano Page Design Layout Initial Sketch

Usability & Quality of Design

The navigation system was handled with a simple responsive navbar within the header of each page. It was designed in a very simple and clean way that includes a website / company logo, with each relevant page name displayed horizontally. The narrow screen alternative allows for users to click on a menu icon that will display a dropdown with the same page names vertically. The design is very intuitive and consistent among other applications as it is a very common method of handling navigation between pages.

The home page layout is very common consisting of a simple hero image acting as a background behind some hero text and a simple button. It is very clean and professional looking, and nothing a user would not expect. The login and signup forms were also implemented using a very universal and common approach. Error messages are displayed to the user to help identify problems such as a wrong password input.

The search form on the volcanoes page was designed with many considerations that had to be made regarding different types of input fields and overall layout. The final approach is very intuitive and would be understandable to the majority of users requiring information on volcanoes. The table displays important information clearly to the user and the hover overlay indicates to the user that there is an extra page for more information. The layout of the volcano page is very simple and a common approach to displaying information on a website.

Overall, all components within the application are designed with consistent colours, fonts, font-sizes, and page layouts. This results in a final product that is cohesive, clean, professional, and understandable to any users that may wish to access the volcano information provided by the website.

Accessibility

The following bullet points analyse the accessibility quality of the application in relation to the priority 1 checklist of accessibility guidelines from w3.org.

1.1 Not all images used in the application have text equivalents such as the hero image on the home page or the interactive map on the volcano page. Although, the home page icon has an alternative text link within the navbar, and all other important buttons and actions use text to describe their purpose. Further implementation could help to further satisfy this guideline.

2.1 All important information is displayed without colour being an important factor. This guideline is mostly applicable to components such as graphs in which colour is used to decipher information. Therefore, black & white alternatives are not required for this application and this guideline is met.

6.1 The JS files within the application are organised in a way such that the CSS files are not required for the main purpose of the website to be accessible. Although less structured, all functionality of the website is visible and readable, as seen in Figure 13. This guideline is passed.

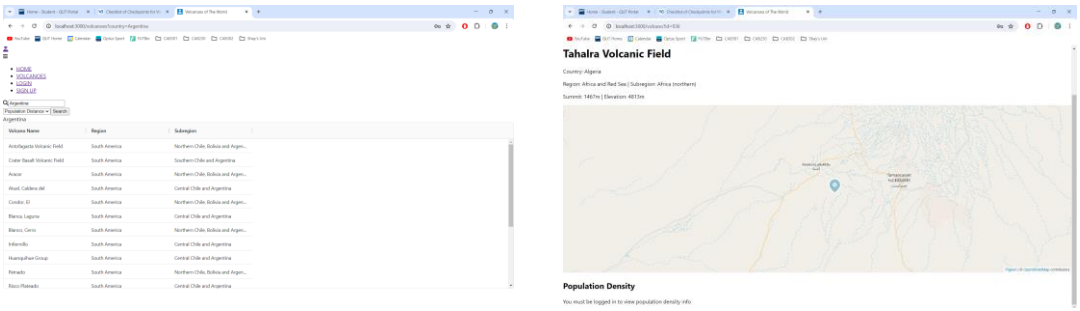

Figure 13 - Application Screenshots Without CSS

4.1 Most dynamically updated changes made to the website will also update any text relevant to the change, such as the sign up and login features, which will dynamically update the label message to display any error messages or if the action was successful. Some further implementations could better the application in relation to this guideline, such as text equivalent captions for the interactive map on the volcano page.

7.1 All dynamically updating actions used such as navigation are done in non-glitchy and smooth ways that do not cause the screen to flicker in any scenarios that were tested. As per user tests, this guideline is satisfied, though future bug testing would improve confidence.

14.1 All core functionality is described with simplistic language that should be understandable to all users. This includes most buttons being one or two words that accurately describe the action, and headings / labels being short, descriptive, and easy to read.

5.1 All columns in the volcanoes table are labelled and marked clearly with obvious separation between the table content and headings. This guideline is also passed.

Technical Description

Architecture

The source code for the React application was organised in a way very similar to the automatically generated project. It consists of a public folder which is used to store the html index file, and image files such as photographs and the favicon.

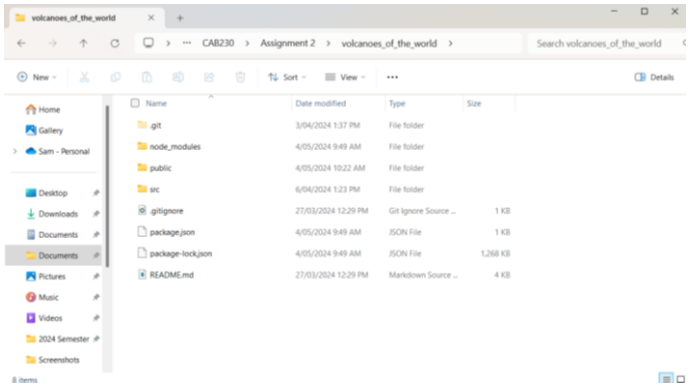

Figure 14 – React Application Code Folder Structure

The src folder contains all the JavaScript and CSS files required for the main functionality and customisation of the website. The App.js file is the main component that handles the dynamic navigation between pages of the website by displaying different components based on the current page index. The components responsible for these pages are found in the pages folder. The components folder includes all smaller JS and CSS files used to create individual components that may or may not be repeated throughout the website such as the navbar, search bar, table, graph, etc.

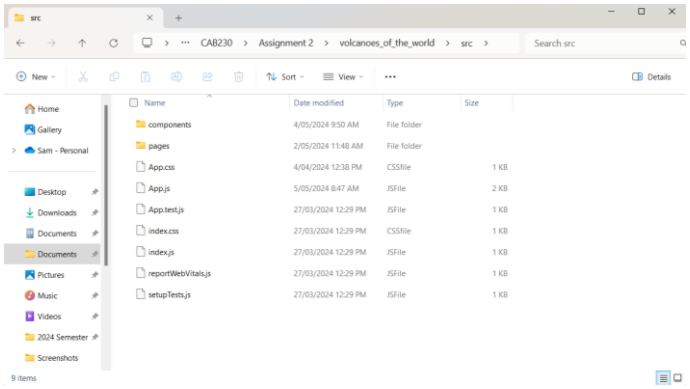

Figure 15 - src Folder Structure

The overall structure of the organisation method is simple and clear to understand and make changes to when required. Breaking up pages into smaller components is a very effective way of simplifying the page layouts, and having these types of components separated in different folders is a common React practice for readability.

However, all individual components are lumped into the one folder and not organised. This could cause future problems if the application was to be expanded and the number of component files increased. Another small problem with the method is that each JS file has a respective CSS file attached. This does not allow for common components to share CSS styles, which could add to workload when adding extra components or pages that wish to reuse styles.

Test Plan

The following tests were manually conducted to check the outcome of various scenarios when interacting with the website.

Task	Expected Outcome	Result	Screenshot (Appendix A)
Access each page using navbar	Display is updated to show new page	PASS	-
Enter character into volcanoes search bar	List of countries with that letter are displayed	PASS	1
Enter second character into search bar	List of countries is updated	PASS	2
Press enter when searching for country	Top country in list replaces user input in search bar	PASS	3
Click on country in list when searching for country	Clicked country replaces user input in search bar	PASS	4
Select population distance in dropdown	Specified population distance is displayed in button label	PASS	5
Search button is pressed without population distance selected	All volcanoes in specific country are listed in table	PASS	6
Search button is pressed with population distance selected	All volcanoes in specific country that have a population within that distance are listed in table	PASS	7
Hover over volcano row in table	“Click for more info” label is shown	PASS	8
Access volcano info by clicking on it in table	Volcano page is displayed	PASS	9
View population density data when logged in	Graph showing population density of the volcano is displayed	PASS	10
View population density data when not logged in	Message saying “You must be logged in” is displayed	PASS	11
Signup using valid email and password	User is taken to login page	PASS	-
Signup using an already used email	Displays message “User already exists”	PASS	12
Signup with incorrect confirmation password	Displays message “Passwords do not match”	PASS	13
Login to account with correct information	User is taken to home page + navbar displays “Logout”	PASS	14
Login to account using incorrect information	Displays message “Incorrect email or password”	PASS	15
Attempt to sign up without internet access	Displays message “An error occurred, please try again”	PASS	16
Attempt to search for volcano without internet access	No countries are displayed, and error message is logged to terminal	PASS	-

Difficulties / Exclusions / Unresolved & Persistent Errors

Some roadblocks were encountered whilst implementing functionality for the website. Though challenging and sometimes time consuming, there were no problems that could not be overcome. An example of one particular difficulty were the login & logout functions, as the program was required to keep track of whether the user was logged in or not. The first way that this was achieved, was simply through the use of a useState variable in the navigation bar JS file. This worked for the most part but could not account for page refreshes or new tabs. This problem was solved by also keeping track of the logged in state using local storage.

One outstanding bug that was identified within the program, is the action of logging out whilst on a volcano page. The action of logging out clears the id from the search parameter, thus losing the volcano currently being inspected. Though when this happens, the user can simply search for the volcano again without problems.

User Guide

When the application is first open, the home page is displayed to the user, as shown in Figure 16. Hovering over the hero text or explore button will cause them to slightly expand, giving an interactive feel to the website. Clicking the explore button will take the user to the volcanoes page.

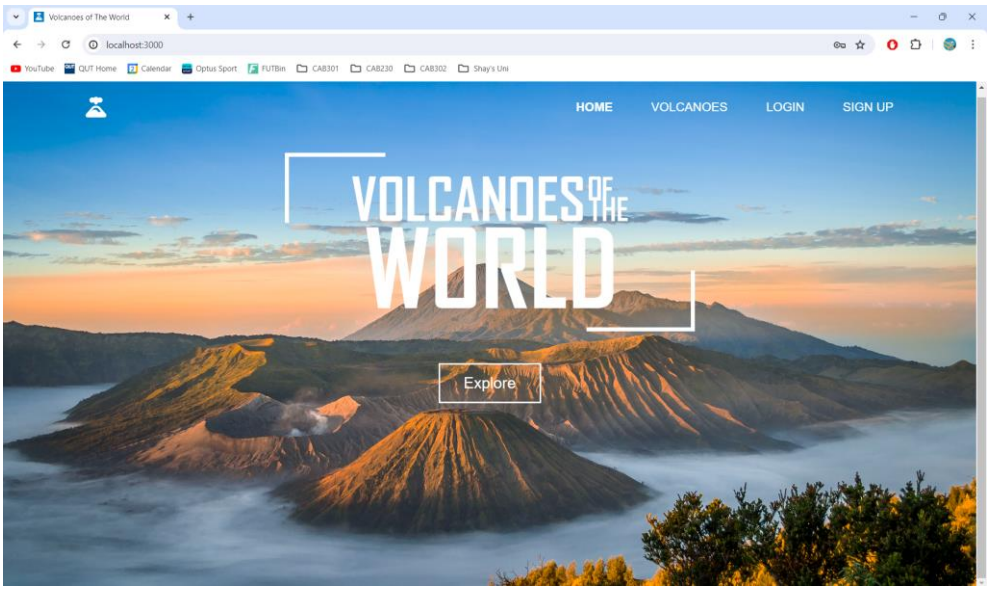

Figure 16 – Wide Screen Home Page Screenshot

The responsive nature of the navbar can be visualised by resizing the window to be narrower. When the screen width is below 960px, the navbar changes to that showed in Figure 17.

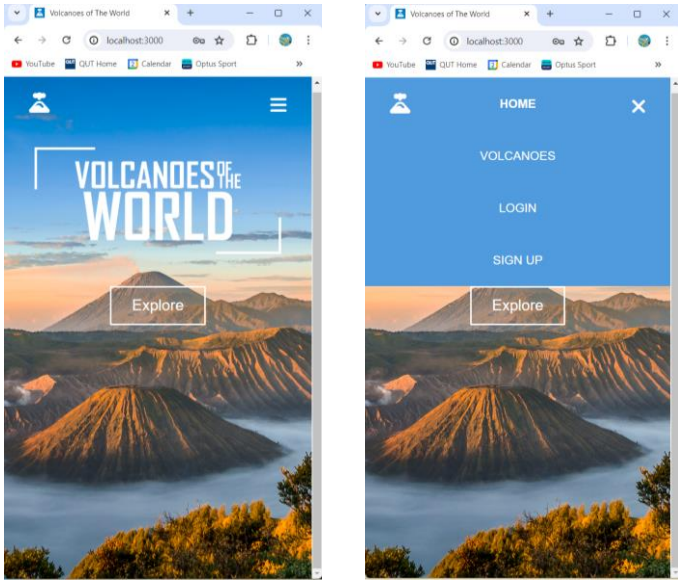

Figure 17 - Narrow Screen Home Page Screenshot (Including Dropdown Menu Before & After)

Clicking on the sign-up button in the navbar will open the sign-up page, where the user must input an email address that has not been used before, followed by a password that is to be entered twice. If this is done correctly, the account is created, and the user is automatically taken to the login page.

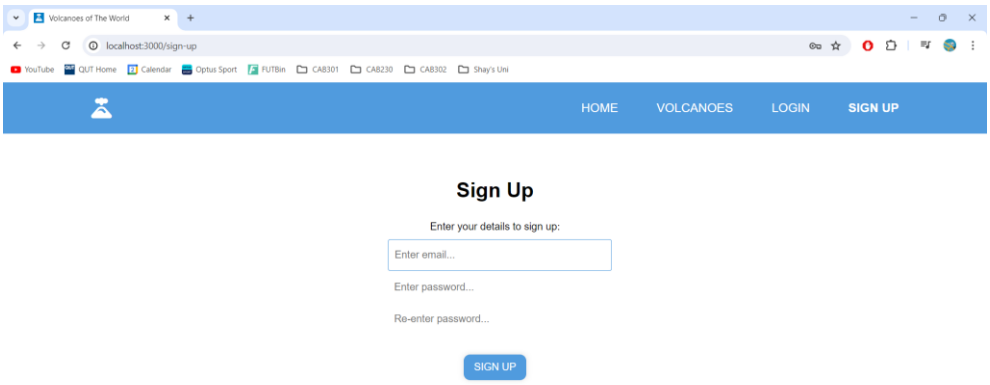

Figure 18 - Sign-Up Page Screenshot

The login page can also be accessed via the navbar and is in a very similar format to the sign-up page. The user can input an email and password, and if they match, the user is successfully logged in, and taken back to the home page.

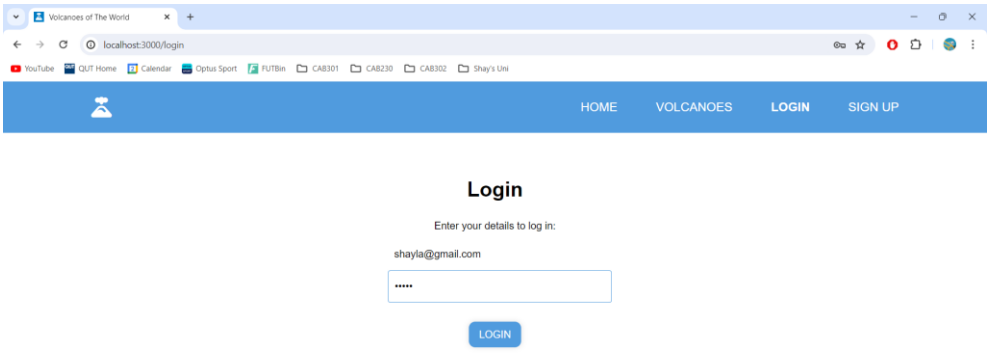

Figure 19 - Login Page Screenshot

Once the user has logged in, the navbar is altered to display a 'logout' button instead of login and sign-up. Clicking this new button will sign the user out of the account and revert to the original state.

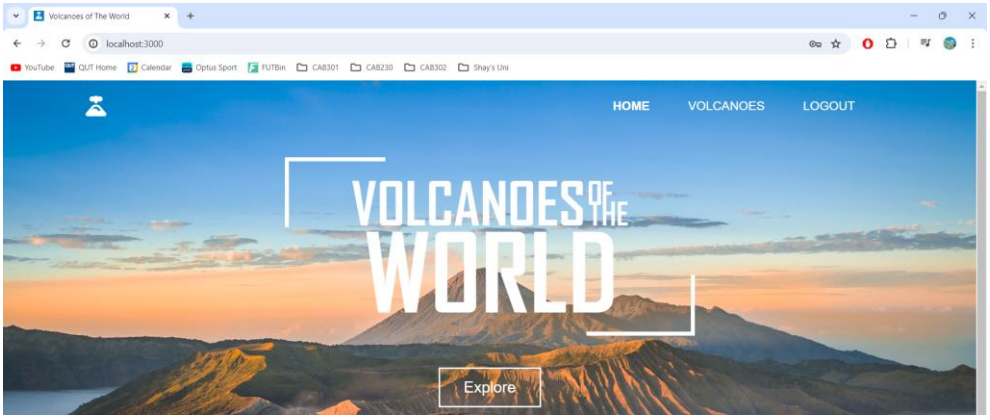

Figure 20 - Logged In Home Page Screenshot

The volcanoes page can be accessed via the navbar or by the explore button on the home page. When first opened, the following search bar, dropdown, button, and empty table will be dsipalyed. The search bar form is responsive, meaning on a narrow screen, the searchbar will re-position itself to be above the dropdown and search button in order to properly fit inside the window, and be an appropriate readable size.

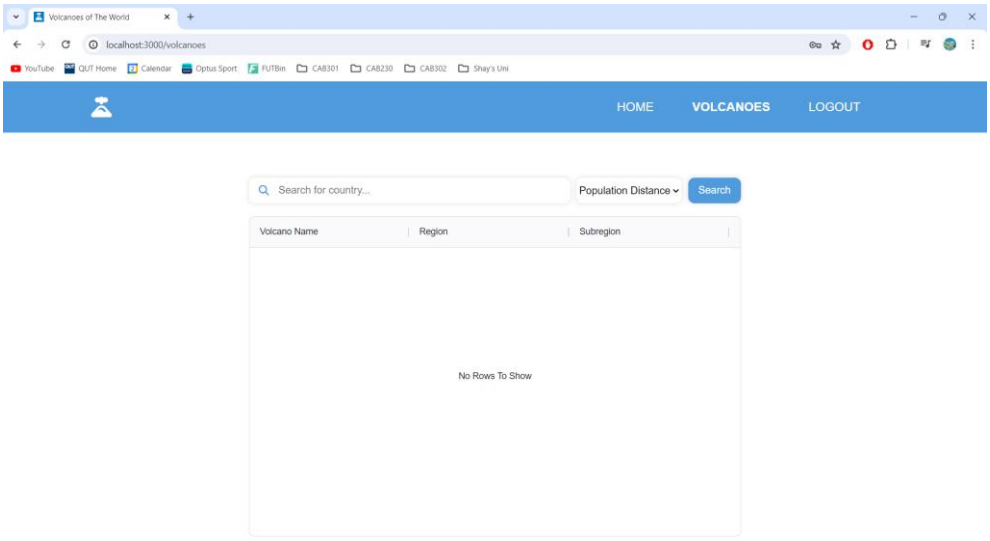

Figure 21 - Wide Screen Volcanoes Page Screenshot

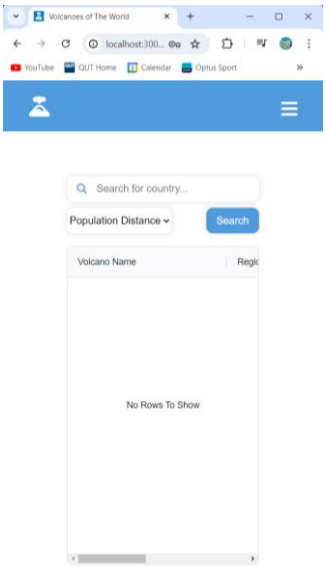

Figure 22 - Narrow Screen Volcanoes Page Screenshot

If the user begins to type into the search bar, a list of autocomplete options are displayed in a dropdown-like menu. The user can select one from this list with their mouse, keep typing, or press the ENTER key to automatically select the top option.

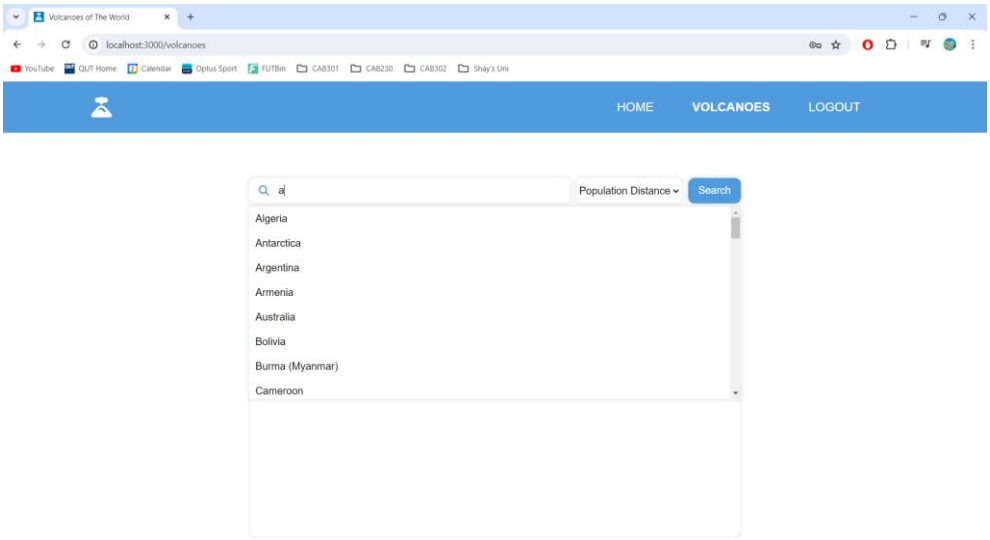

Figure 23 - Volcanoes Search-bar Autocomplete Dropdown Screenshot

The population distance button is a dropdown, clicking on it will open a list of set distances including 5km, 10km, 30km, and 100km. The user can select one of these or leave it on the 'Population Distance' option leave it unspecified.

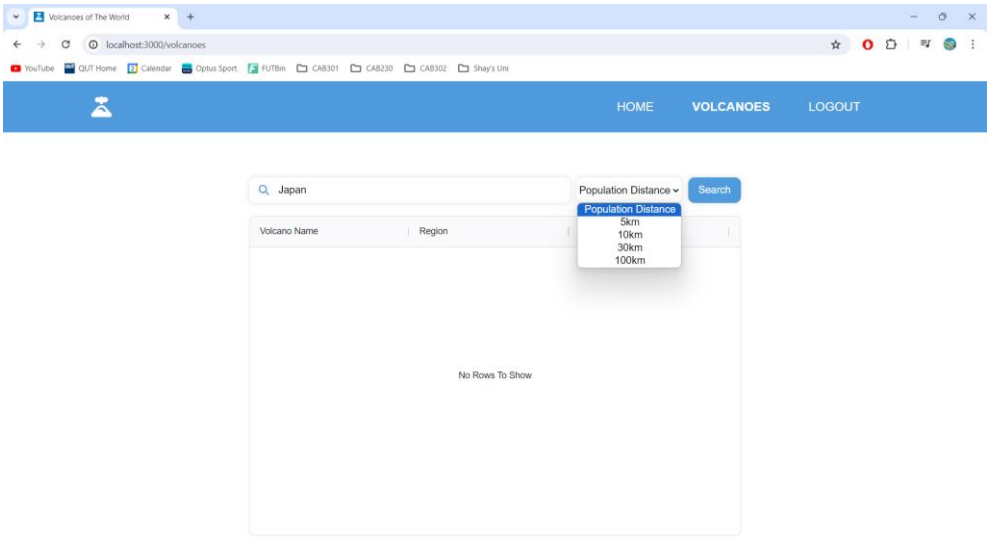

Figure 24 - Population Distance Selection Screenshot

The user can then click the search button to display results in the table that match the given input fields.

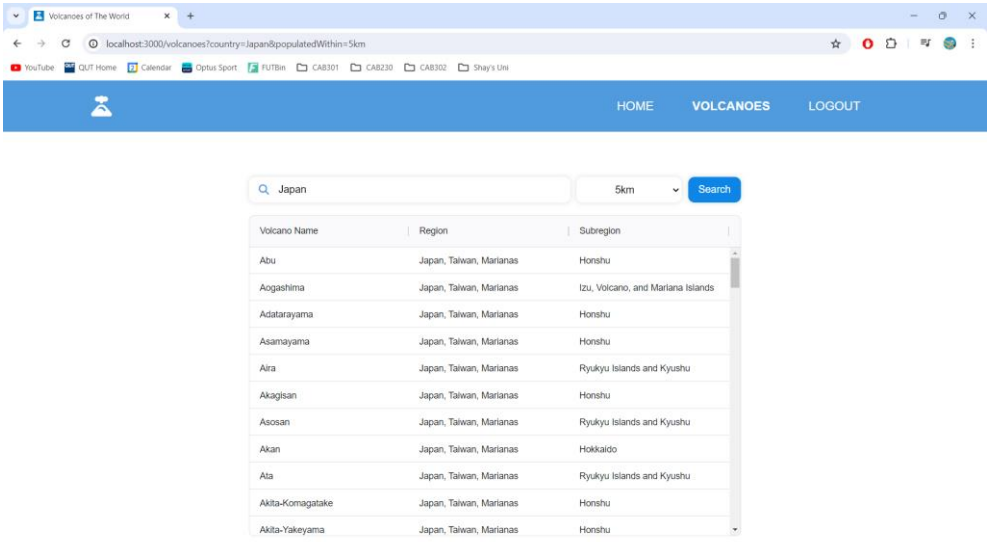

Figure 25 - Volcanoes Page with Table Results Screenshot

Highlighting over a volcano in the table will show a label that says “Click for more info”. Clicking on a volcano will take the user to the information page for that specific volcano.

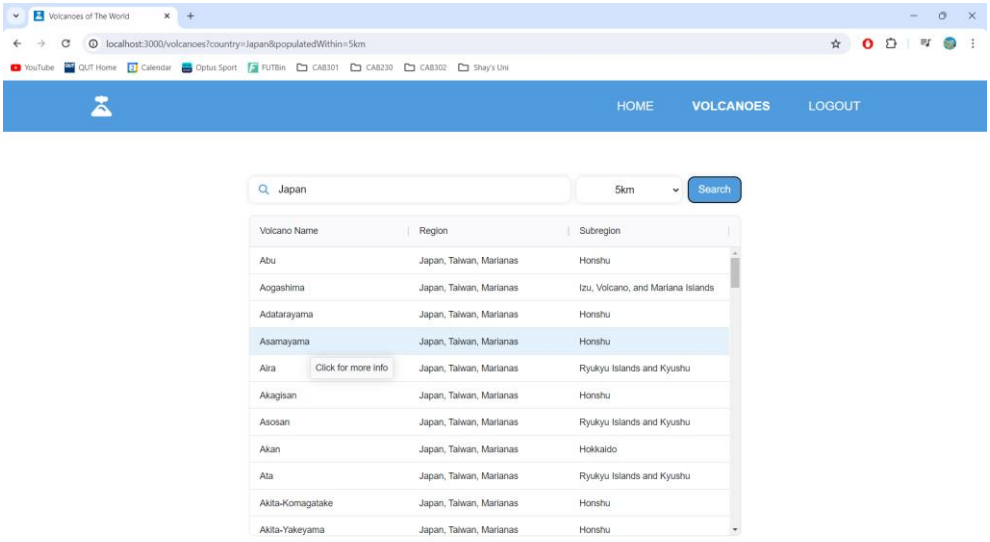

Figure 26 - Volcanoes Table Row Hover Screenshot

The volcano page displays some basic information about the volcano such as the name, region, subregion, summit, and elevation. There is also an interactive map that shows the position of the volcano using its longitude and latitude coordinates.

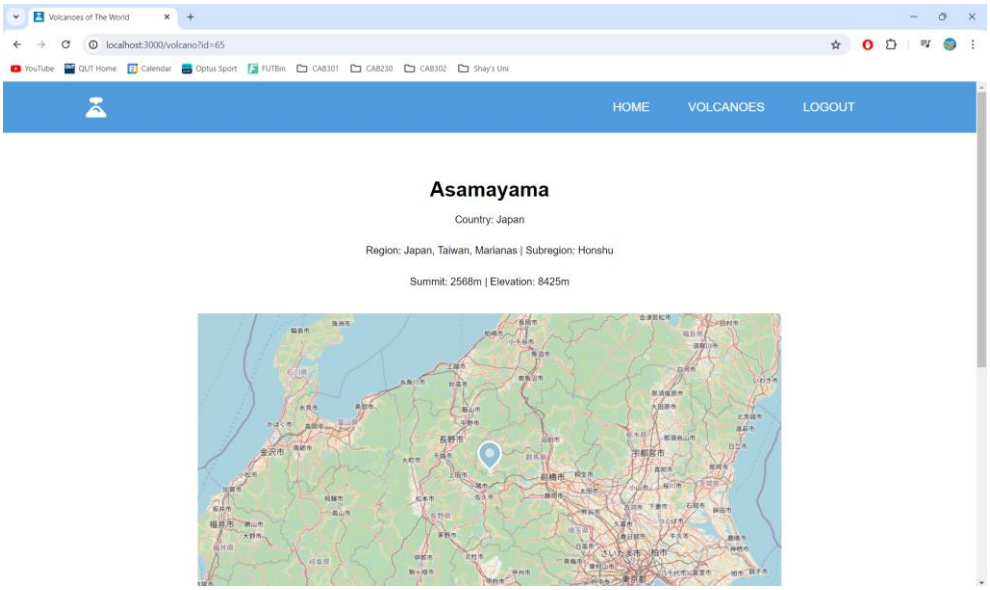

Figure 27 - Volcano Information Page Top Screenshot

Scrolling down, if the user is logged in, will show a graph displaying the population density of the area surrounding the volcano. If the user is not logged in, a simple message is displayed instead, reminding the user to log in to view population density info.

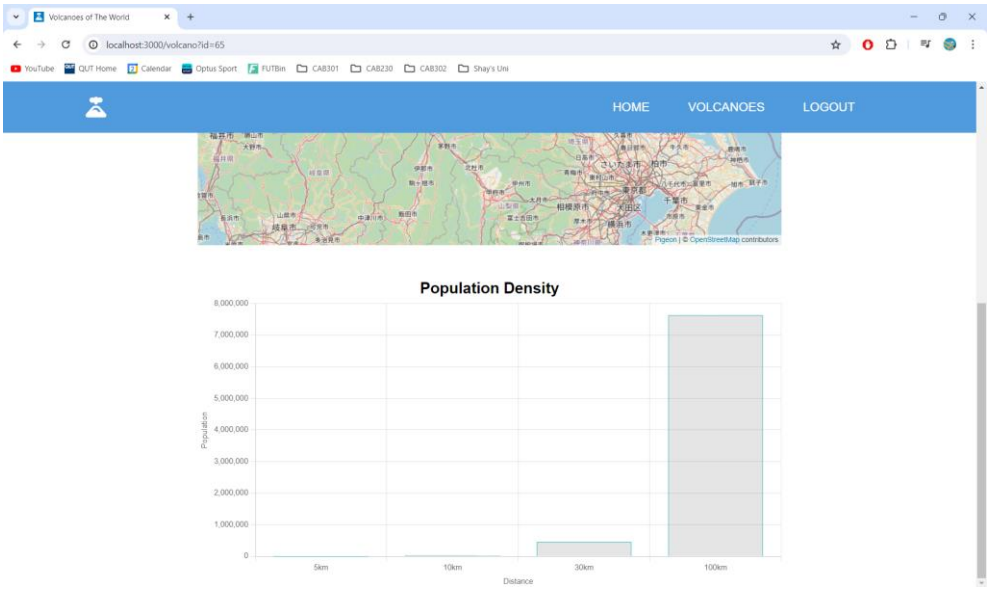

Figure 28 - Volcano Information Page Bottom Screenshot

References

Chrisinthai, (2015). Sunrise at Mount Bromo (Photograph)

iStockPhoto.com: https://www.istockphoto.com/photo/sunrise-at-mount-bromo-gm494386476-77413429

Ikunl, (2013). An aerial view of mountain tops in the fog (Photograph)

iStockPhoto.com: https://www.istockphoto.com/photo/an-aerial-view-of-mountain-tops-in-the-fog-gm181928587-27300647

W Chisholm, G Vanderheiden, I Jacobs, (2021). Checklist of Checkpoints for Web Content Accessibility Guidelines 1.0 (Document)

W3.org: https://www.w3.org/TR/WAI-WEBCONTENT/full-checklist

Appendix

Appendix A - Test Case Screenshots

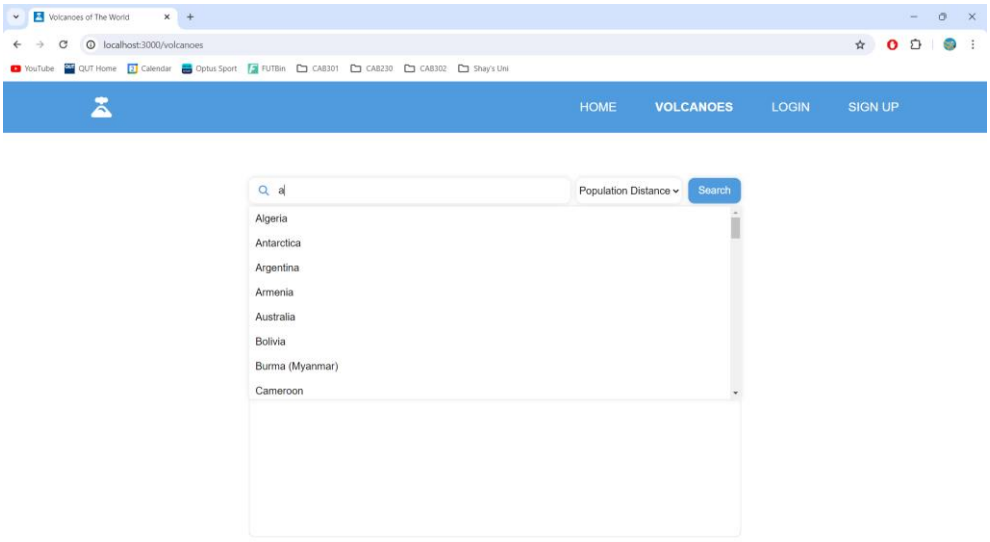

Figure 29 - Test Case Screenshot 1

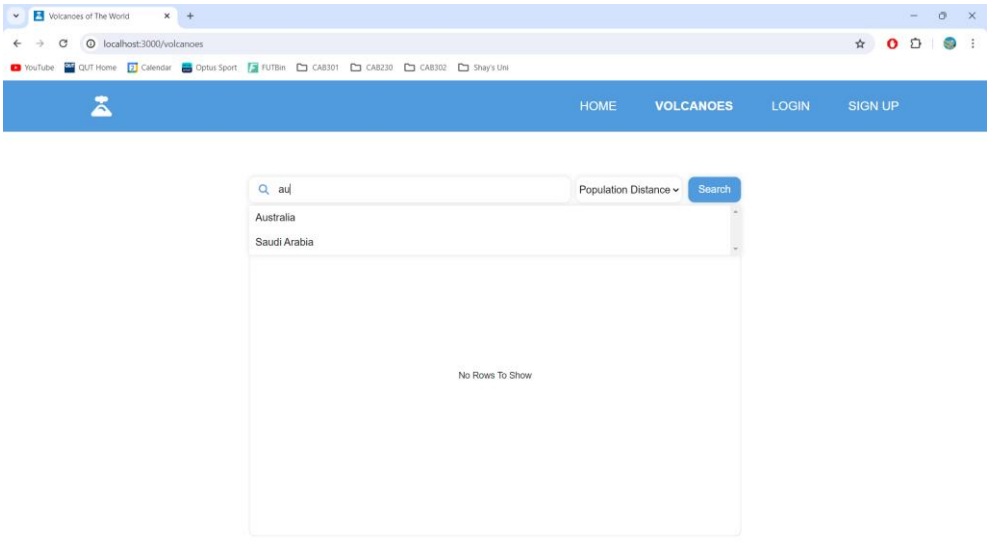

Figure 30 - Test Case Screenshot 2

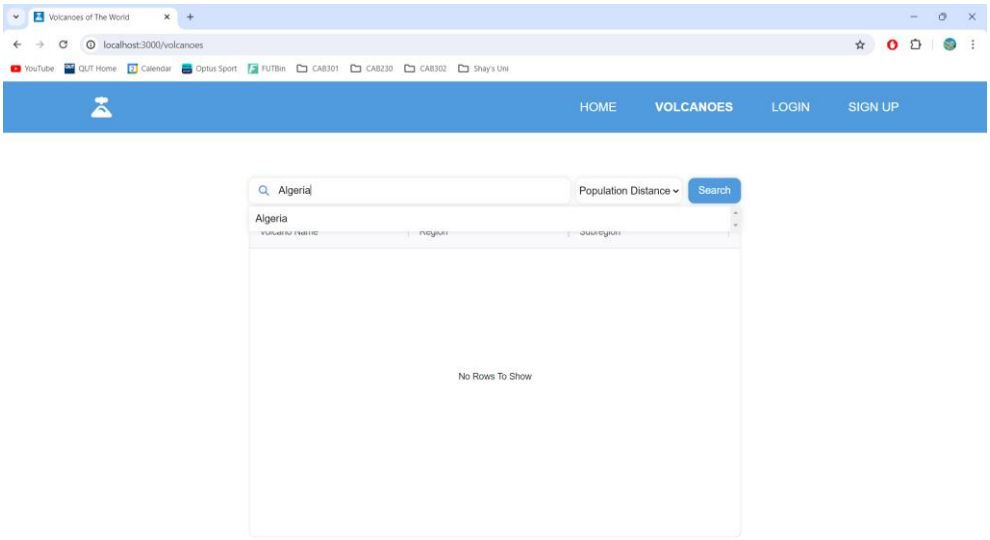

Figure 31 - Test Case Screenshot 3

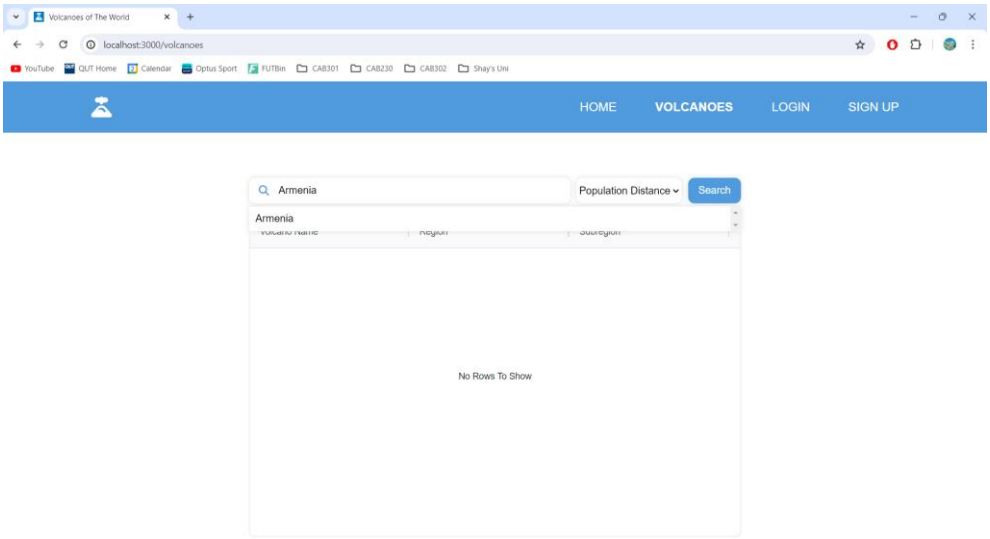

Figure 32 - Test Case Screenshot 4

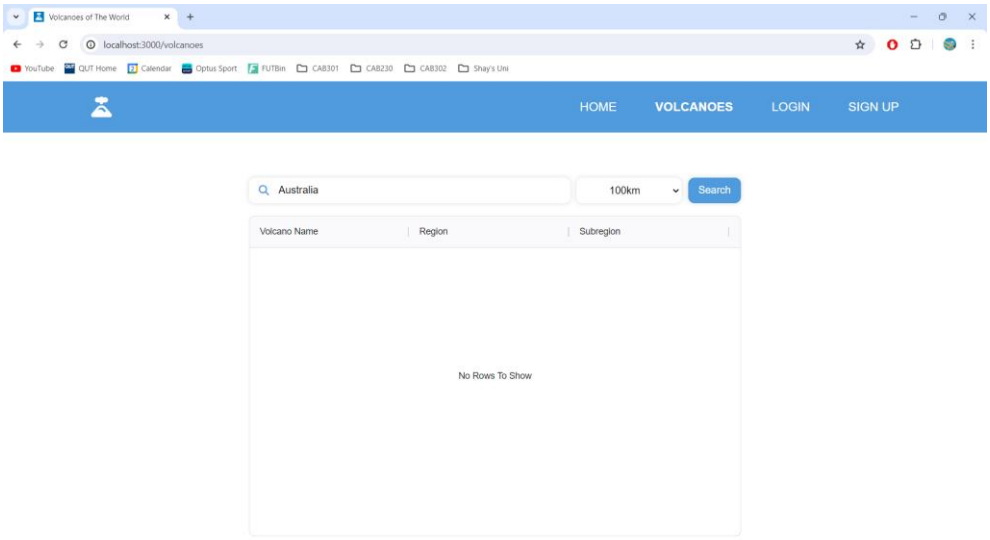

Figure 33 - Test Case Screenshot 5

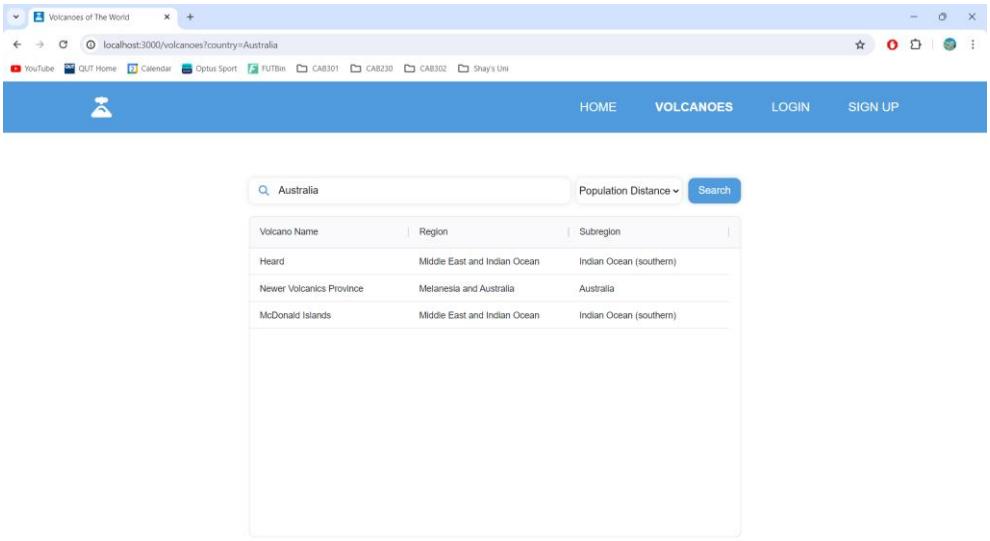

Figure 34 - Test Case Screenshot 6

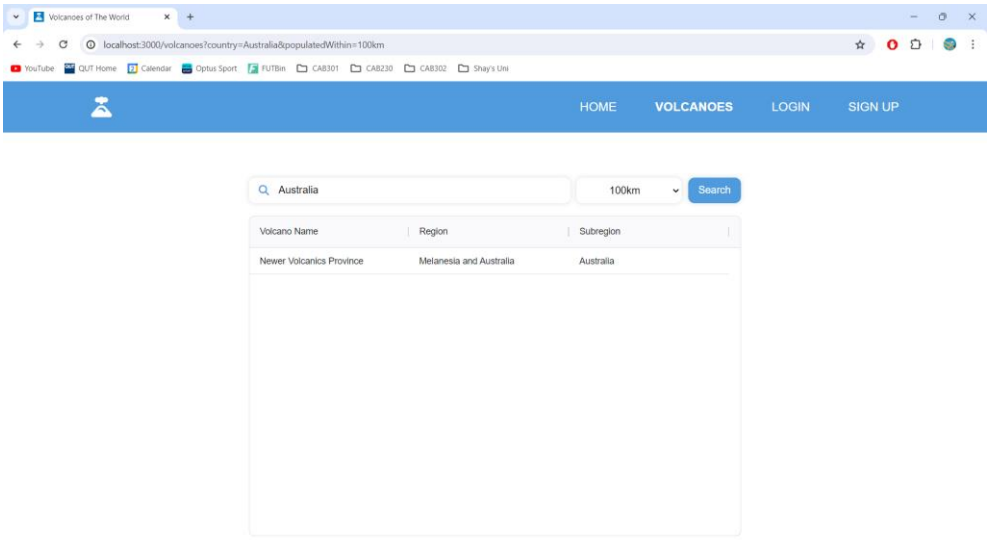

Figure 35 - Test Case Screenshot 7

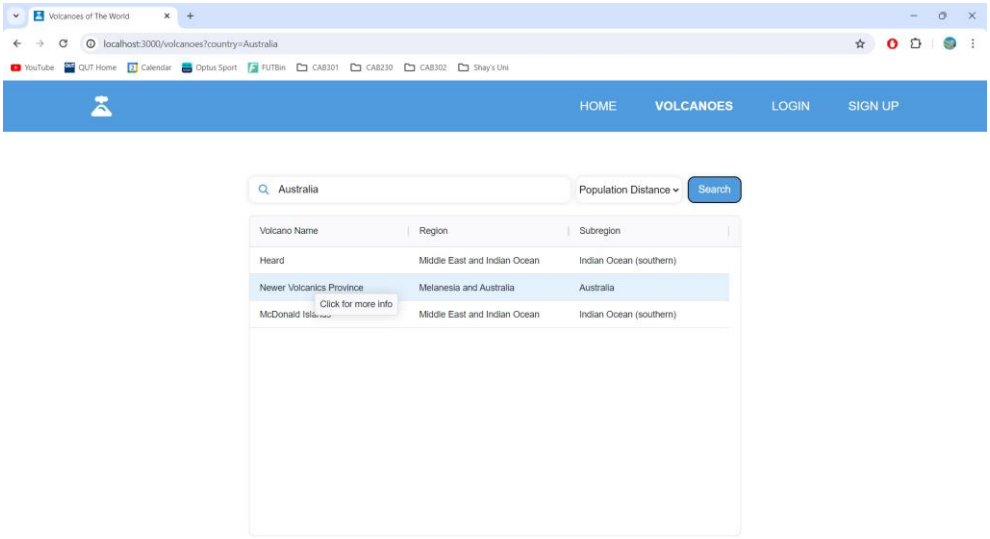

Figure 36 - Test Case Screenshot 8

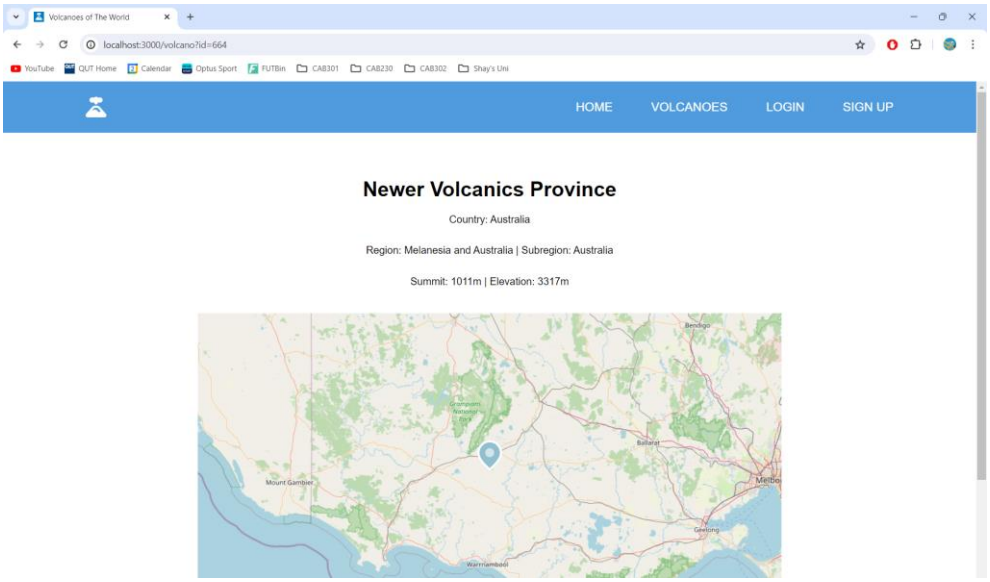

Figure 37 - Test Case Screenshot 9

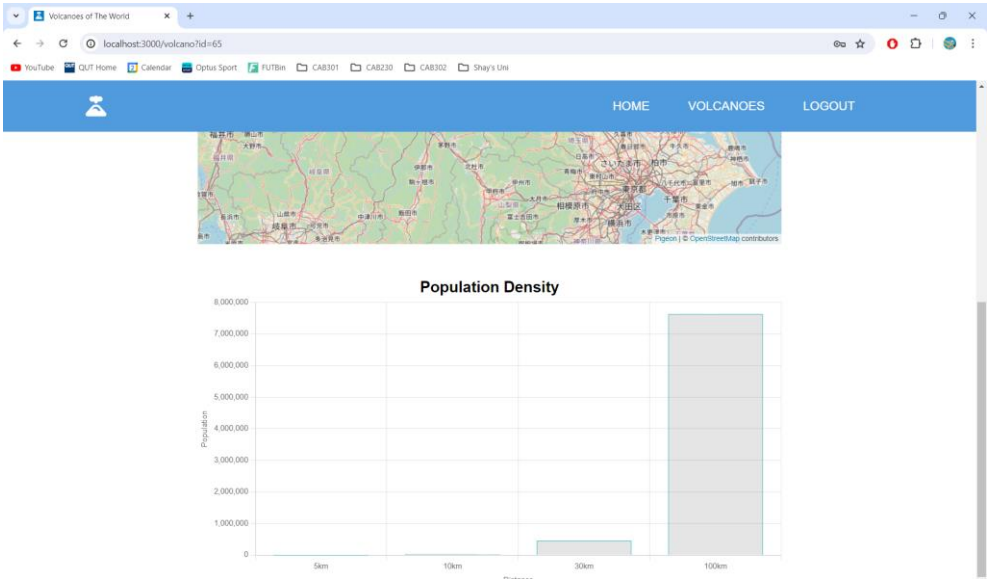

Figure 38 - Test Case Screenshot 10

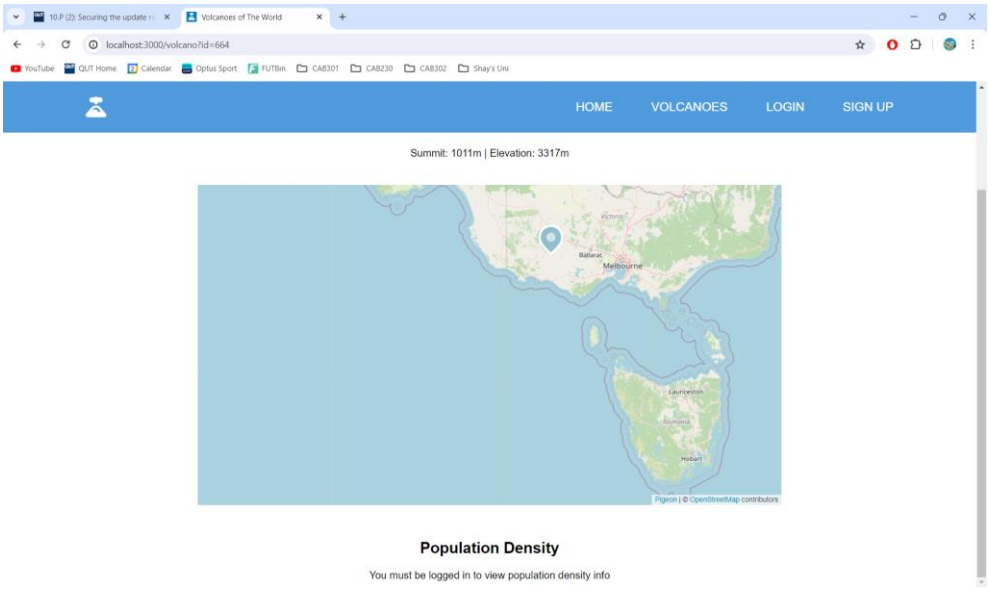

Figure 39 - Test Case Screenshot 11

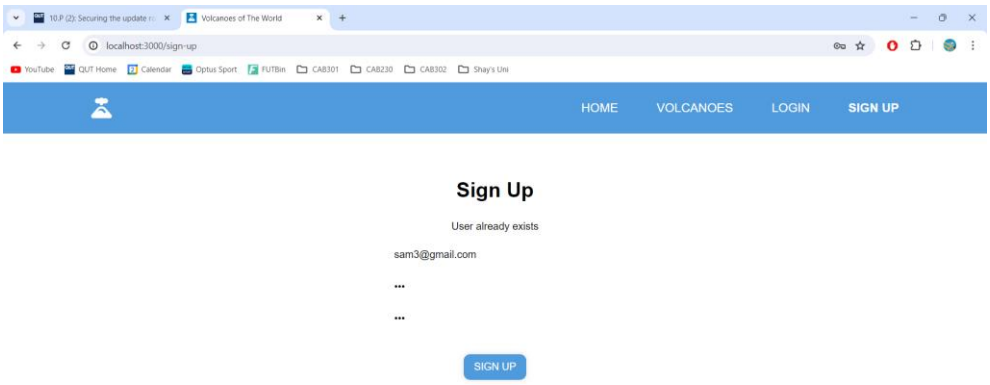

Figure 40 - Test Case Screenshot 12

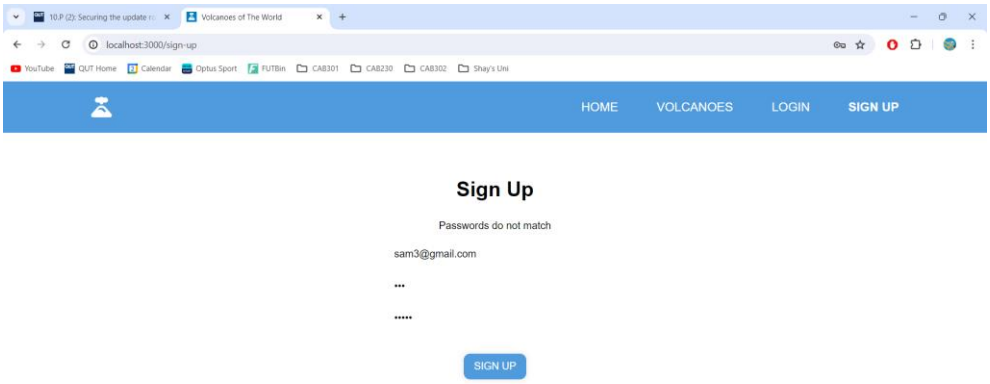

Figure 41 - Test Case Screenshot 13

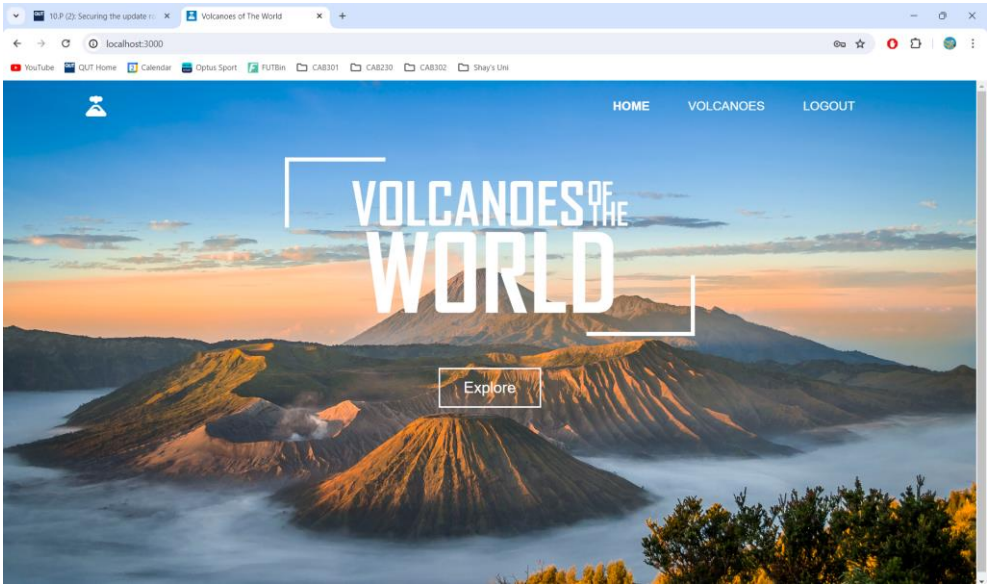

Figure 42 - Test Case Screenshot 14

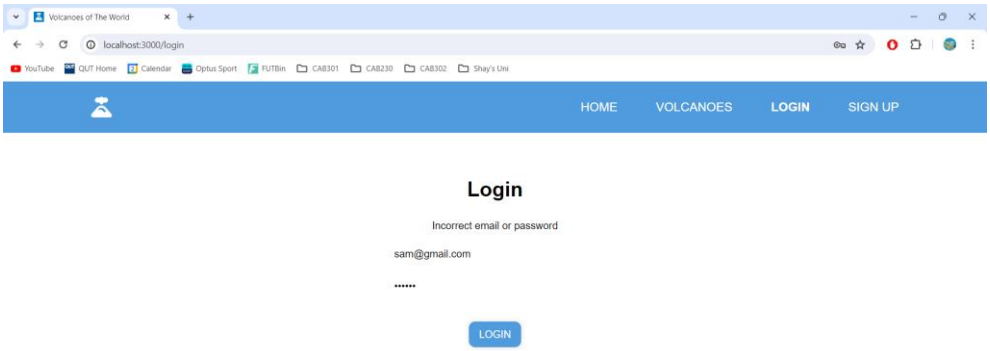

Figure 43 - Test Case Screenshot 15

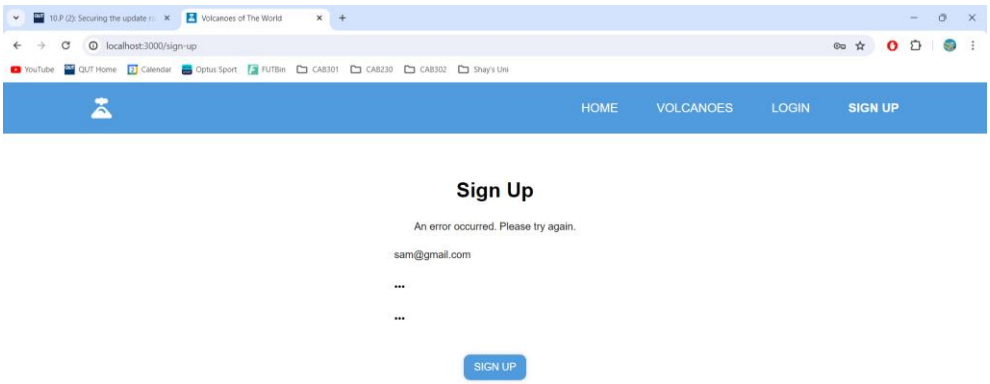

Figure 44 - Test Case Screenshot 16