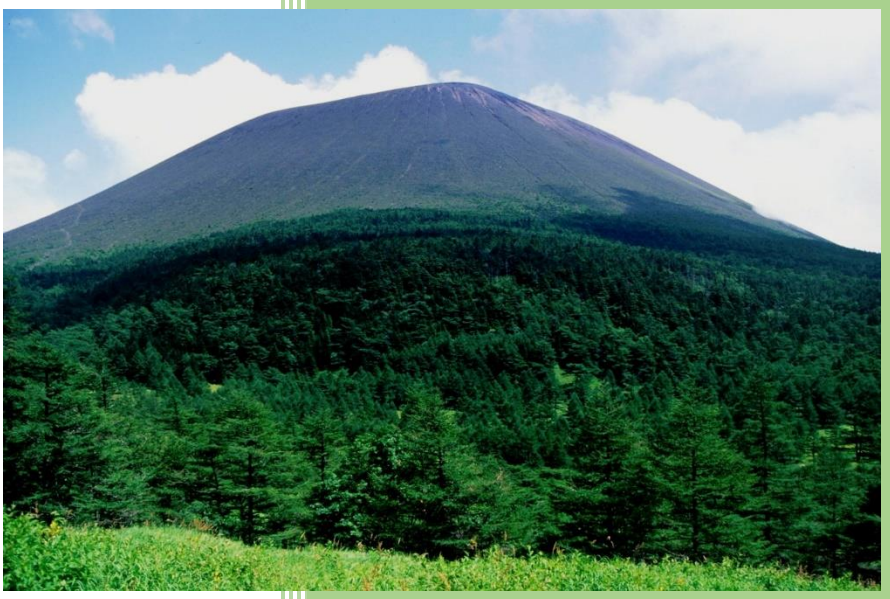
2024

CAB230 Assignment 2



CAB230

Volcano API – Client Side Application

Matthew Chambers

N11318546

5/10/2024

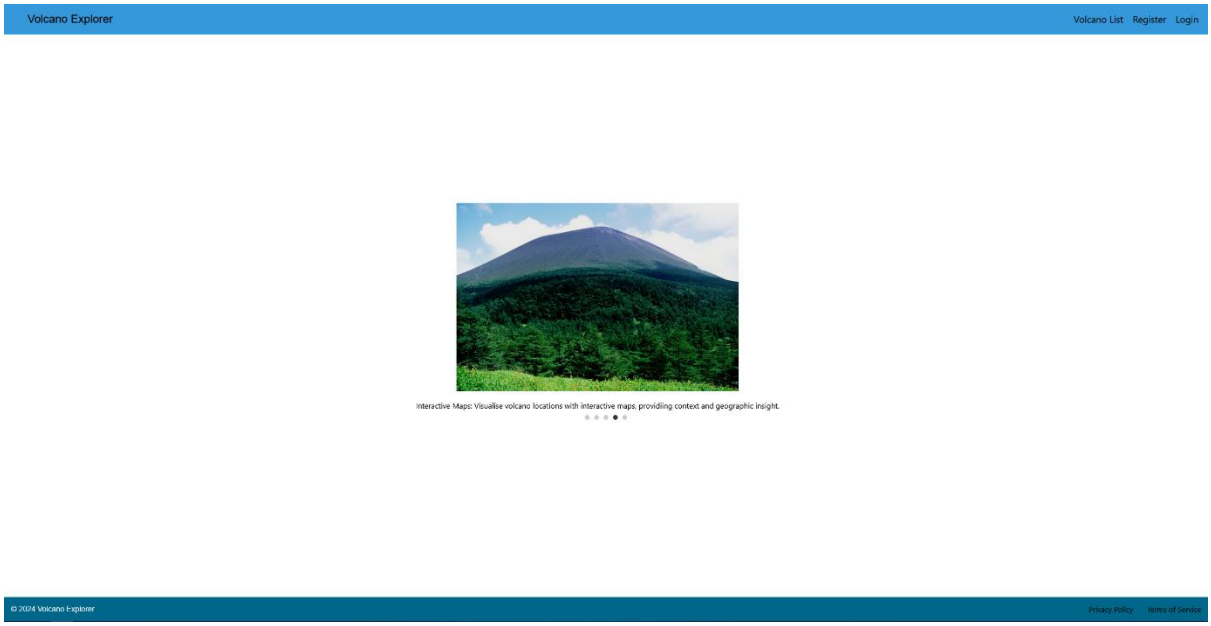
Contents

Introduction

Purpose & description

This React-based application can be broken down into 4 key main functionalities. The first of which is the graphical user interface (GUI) in which the user will interact with and serve as the base of all other functionalities of the app. From this GUI, the user should be able to access pages such as log in, registration, the volcano list, and each volcano's individual page. In order to be able to have the data to populate these pages, the apps will interact with the REST API and fetch the required information such as names of volcanoes and their region based upon a user selected country. And finally, the user should be able to register for an account and log in to access authenticated data such as population density around volcanoes.

A prominent goal during development of the GUI for this app was to prioritise accessibility. A minimalist design approach was used to ensure ease of navigation and understanding for all users, regardless of technical proficiency. For pages in which data had to be displayed, it was ensured that the data was always shown in clean and concise layouts, negating unnecessary complexity. Additionally, for some volcanoes it was found that their population density graphs could not be easily read with a typical linear scaling graph. To counter this, it was chosen to use a logarithmically scaling graph to ensure that all data is clearly visible.



Completeness and Limitations

The implementation of the app ensures that all functionalities are smoothly operational. Page routing is effectively handled through React Router, allowing users to move through all sections of the app with little complication. Controlled forms are used to ensure that user's input is effectively and efficiently managed, which enhances the user experience. The accessibility-focused design of the GUI is well-considered, contributing to its overall ease of use. However, additional accessibility functionality that could greatly aid some users is absent. This could potentially include functionality such as text enlargement or colour adjustments for the visually impaired. Overall, it is believed that the application implementation meets the requirements as it delivers a fully function and user-friendly experience.

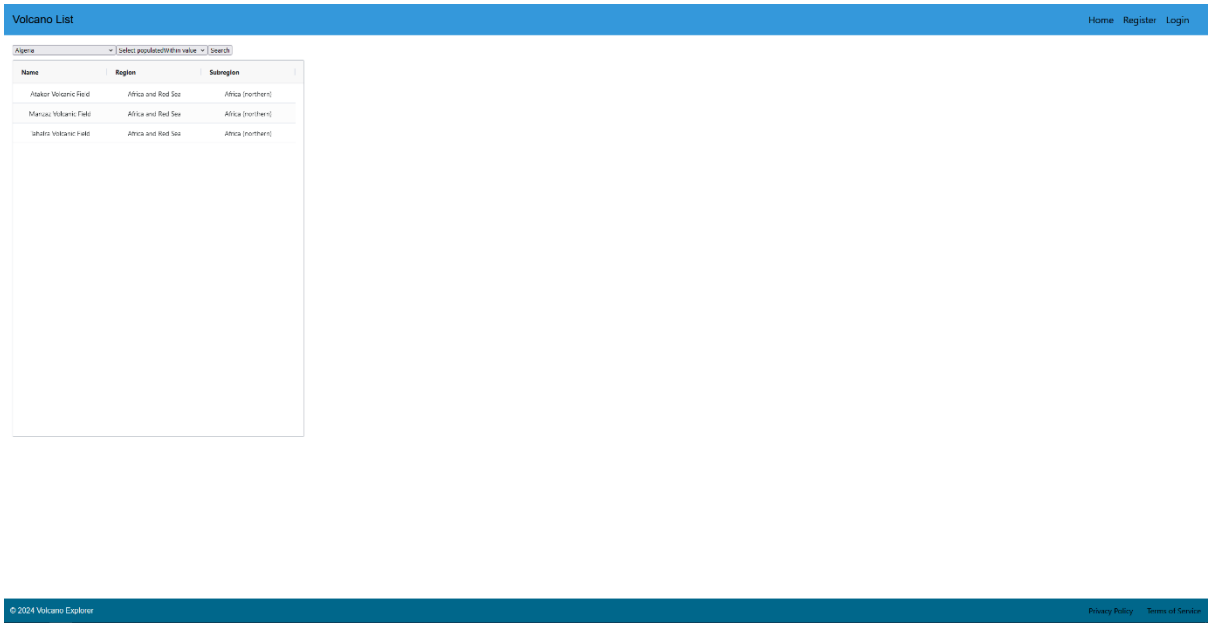
Use of End Points

/countries

The countries endpoint of the API was used to populate the drop down menu for when the user is wanting to select a country to search for a volcano.

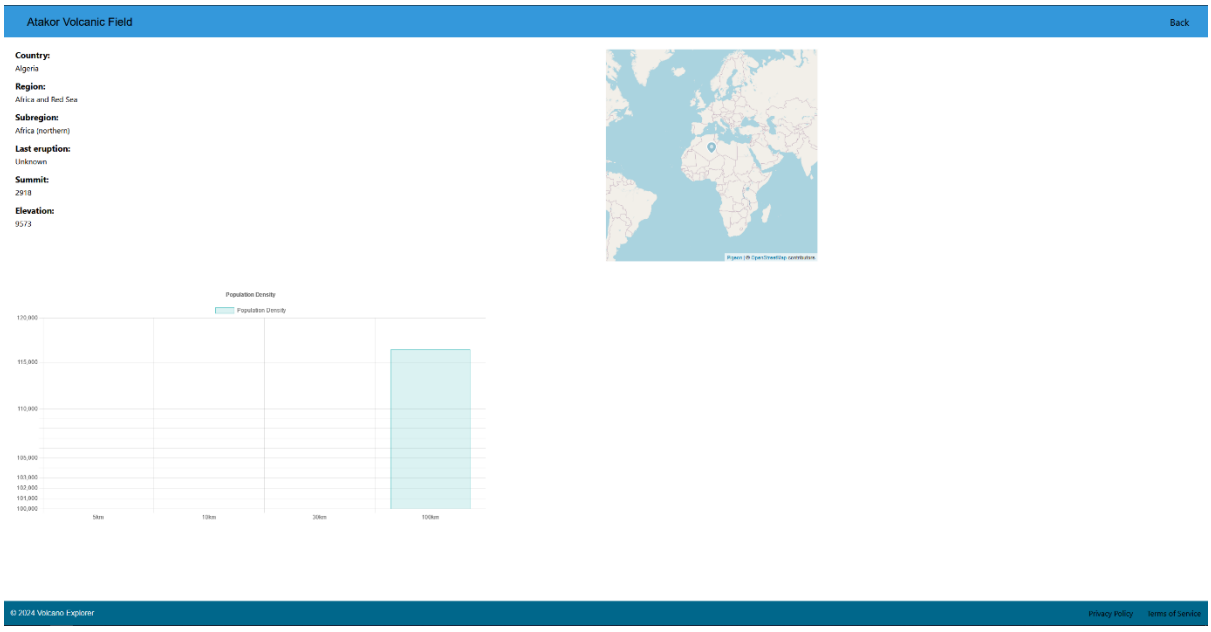
/volcanoes

The volcanoes endpoint is used when the user searches after inputting their desired country and ‘populatedWithin’ value.



/volcano/{id}

This endpoint is used when the user has clicked on an individual volcano in the table and is redirected to the volcano’s individual page.



/user/register

This API endpoint is used when the user has inputted their email and password, and clicks the register button.

/user/login

This endpoint is used very similarly to the user registration, with the difference being that the JSON web token given as a response by the server is stored for later use.

Modules Used

Ag-grid-react

Module to provide fully-featured table components, including sorting and filtering.

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

React Router

Module to provide page routing.

https://reactrouter.com/en/main

Pigeon Maps

Module to provide interactable, performance driven maps.

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

Chart.js

Module to provide charts to visualise data.

https://www.chartjs.org/

Application Design

Navigation and Layout

The application ensures accessibility to all available pages via the top menu, which can be found in the top left corner of the screen. The design process of this application followed a logical sequence, beginning with the home page followed by the registration and log in pages. Then, the volcano list page and finally the individual volcano page. The application was designed in this order to ensure easy and consistent site navigation for users. Originally page navigation was intended to be facilitated through a drop-down menu, however this was decided against due to a limited number of pages to navigate which a drop-down menu is more suited to. Instead of a drop-down menu, available pages to navigate are displayed in the top left corner whilst the name of the current page is in the top right.

Usability and Quality of Design

The application functions effectively in terms of useability, however it lacks modern stylistic elements. A prime example of this is the log in and register pages, where the page features solely the fields for user input and the corresponding button. In many modern instances, web designers choose to have a much more styled page having functionalities such as using different platform to sign in. Another hinderance within the app is navigation can at time be confusing most namely due to how fast pages load. Although all accessible pages are consistently shown in the same position throughout all pages, it can be awkward to use. As previously stated, a drop-down menu for navigation was considered but deemed impractical due to the limited number of pages, presenting a usability challenge regardless of the chosen approach. Despite these negative qualities of the app, it is still a functional and fast application that implements all necessary functionalities.

Accessibility

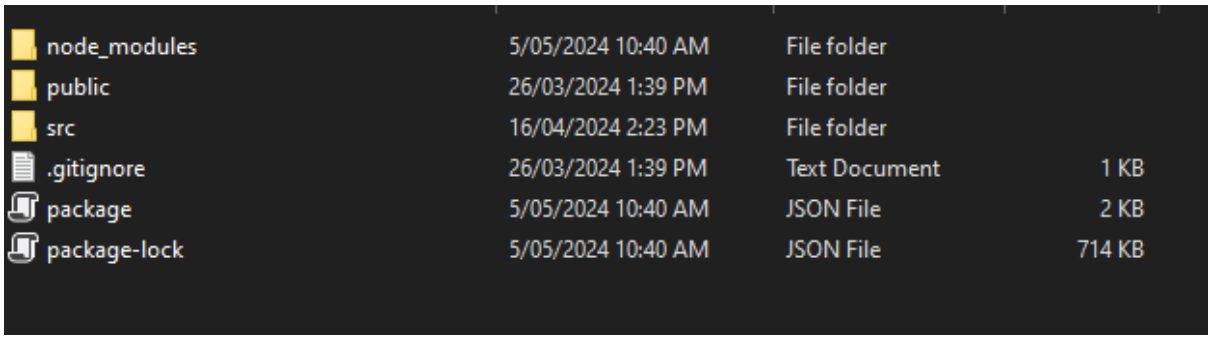
Due to the design being heavily focused on accessibility, it is important to evaluate upon a known accessibility criteria checklist. The checklist used in the evaluation is the W3C priority 1 (W3C, n.d.). After evaluating the application based upon the checklist, it reveals areas for improvement. For features such as the map displayed in the individual volcano page, there is no text equivalent which could potentially pose challenges for users who may be vision impaired. Additionally, the standard style that Chart.js uses for bar charts includes opaque bars which may also further hinder vision impaired users. To remedy this, it was chosen to have all bars the same colour in hopes that it would aid colour blind users to distinguish the data.

However more positively, the application incorporates several accessibility considerations to benefit users with differing needs. It avoids screen flickering or rapidly changing colours, helping minimise eye discomfort or seizures for those with photosensitive epilepsy. Additionally, language throughout the application is kept concise and clear allowing a wide range of users to use it. Tables within the application are structured to identify column easily using bold lettering and segmentation. Therefore, The application has many accessibility accommodations allowing a wide range of users to be able to use it, however there is room for improvement in some areas for the visually impaired.

Technical Description

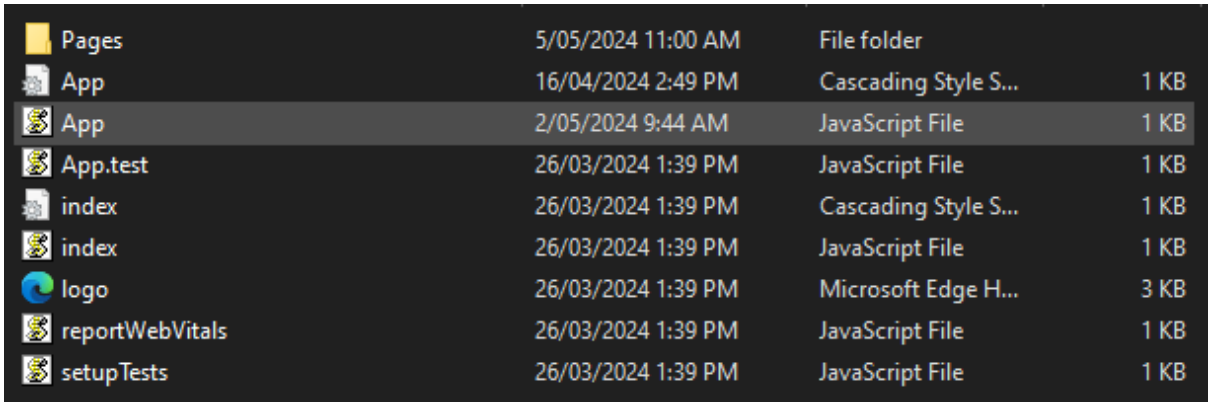
Architecture

The overall architecture of the application follows fairly standard layout conventions, with most relevant code residing in the source directory.

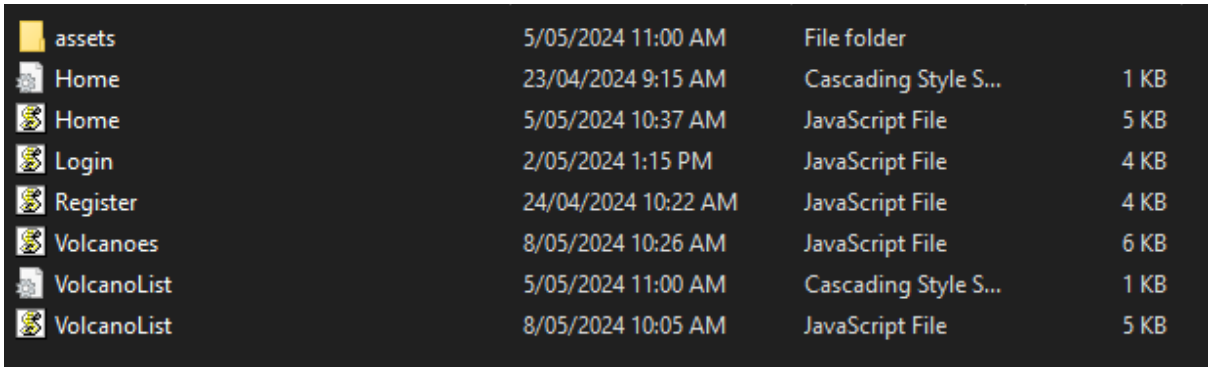


node_modules	5/05/2024 10:40 AM	File folder	
public	26/03/2024 1:39 PM	File folder	
src	16/04/2024 2:23 PM	File folder	
.gitignore	26/03/2024 1:39 PM	Text Document	1 KB
package	5/05/2024 10:40 AM	JSON File	2 KB
package-lock	5/05/2024 10:40 AM	JSON File	714 KB

Within the source directory is most of the auto-generated files as well as the App.js and its corresponding CSS file. With this application implementation, the App.js file serves as the router for all site navigation. Most notably, the source directory also contains the Pages directory which holds each page of the application as individual components and their respective CSS files. Within the Pages directory, exists another assets directory which simply holds the image files that are displayed on the home page. This hierarchal organisation allows for clear separation between different pages, allowing for streamlined management and easy modification when necessary.



Pages	5/05/2024 11:00 AM	File folder	
App	16/04/2024 2:49 PM	Cascading Style S...	1 KB
App	2/05/2024 9:44 AM	JavaScript File	1 KB
App.test	26/03/2024 1:39 PM	JavaScript File	1 KB
index	26/03/2024 1:39 PM	Cascading Style S...	1 KB
index	26/03/2024 1:39 PM	JavaScript File	1 KB
logo	26/03/2024 1:39 PM	Microsoft Edge H...	3 KB
reportWebVitals	26/03/2024 1:39 PM	JavaScript File	1 KB
setupTests	26/03/2024 1:39 PM	JavaScript File	1 KB



assets	5/05/2024 11:00 AM	File folder	
Home	23/04/2024 9:15 AM	Cascading Style S...	1 KB
Home	5/05/2024 10:37 AM	JavaScript File	5 KB
Login	2/05/2024 1:15 PM	JavaScript File	4 KB
Register	24/04/2024 10:22 AM	JavaScript File	4 KB
Volcanoes	8/05/2024 10:26 AM	JavaScript File	6 KB
VolcanoList	5/05/2024 11:00 AM	Cascading Style S...	1 KB
VolcanoList	8/05/2024 10:05 AM	JavaScript File	5 KB

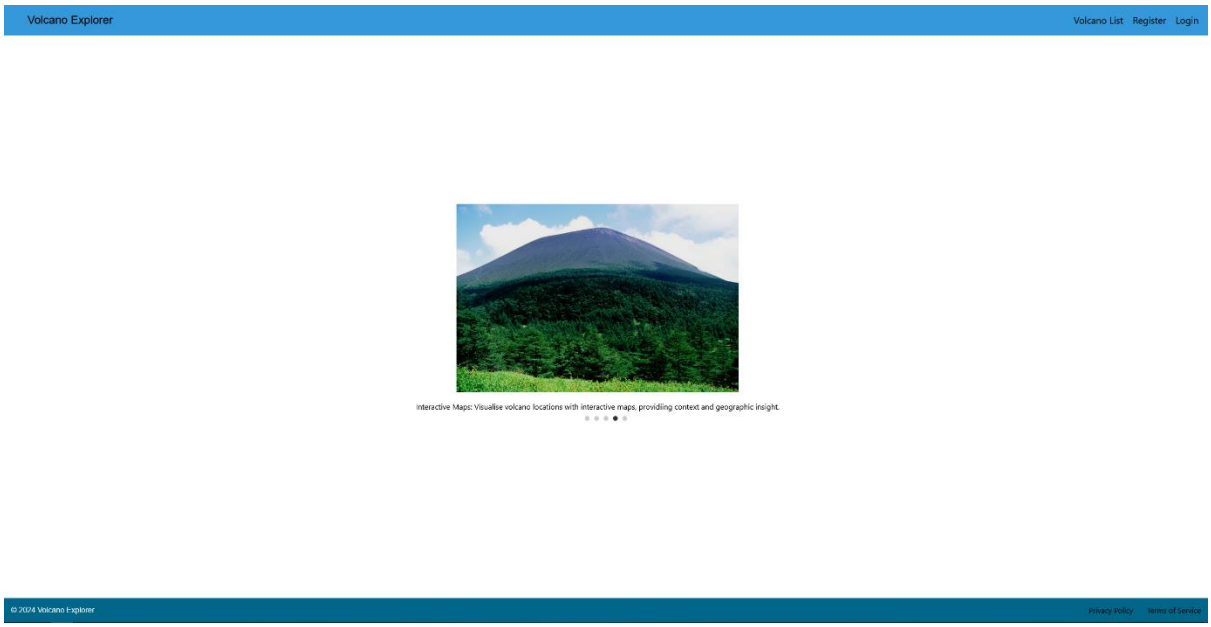
Test Plan

Task	Expected Outcome	Result
Click Volcano List	User is navigated to Volcano List page	**PASS**
Click Register	User is navigated to Register page	**PASS**
Click Login	User is navigated to Login page	**PASS**
Enter registration details, click register button	Upon successful registration, user is redirected to login page	**PASS**
Enter login details, click login button	Upon successful login, user is acknowledged with “login successful” message.	**PASS**
Navigate to Volcano List, Click Search	Table is populated with results of Algeria with no specified populatedWithin Value	**PASS**
Select country, Click search	Table is populated with results related to specified country	**PASS**
Select country and Populated within value, click search	Table is populated with correct values	**PASS**
Click on Name of volcano within Table	User is redirected to individual Volcano page	**PASS**
Evaluate displayed data	Correct data is displayed for chosen volcano	**PASS**
Evaluate map	Correct location is displayed on Map, and is interactable (can be zoomed and moved)	**PASS**
Evaluate population density (logged in)	Chart is shown with correct values	**PASS**
Evaluate population density (not logged in)	Chart is not shown.	FAIL – Empty chart is shown.

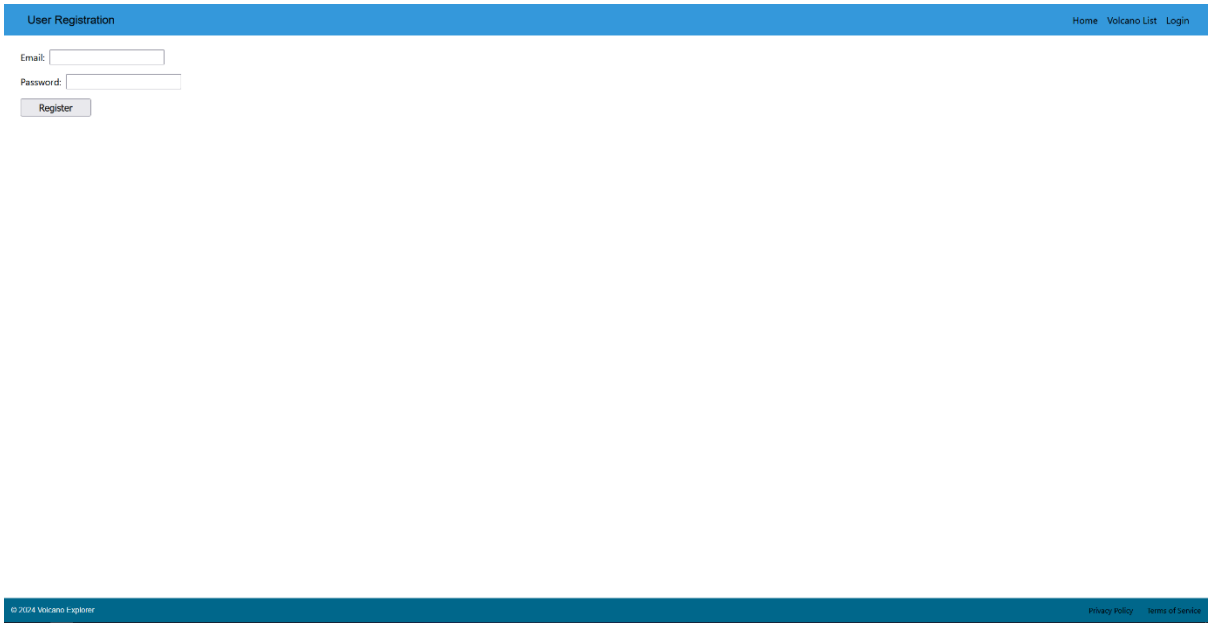
Difficulties / Exclusions / unresolved & persistent errors

Throughout the completion of this application, many difficulties were faced and ultimately overcome. Most namely of which were to do with Pigeon Maps and Chart.js. With Pigeon Maps, the issue was getting it to display the correct location for the given latitude and longitude values from the server response. Often the shown location would be outside of the map and appear as solid grey. This issue was corrected via utilising JavaScript's 'ParseFloat' to convert the given values into floats. With Chart.js, the issue was getting it to display the chart. Many online tutorials depicting how to display a bar chart using Chart.js are for older version, which is syntactically outdated. This issue was resolved by going directly to the official documentation and looking at the provided examples. Ultimately, all specified functionality was successfully implemented but due to time constraints the level of style and detail could not be realistically achieved. In addition to this, the time constraints also allowed for a known bug to not be successfully resolved. As shown in the test plan, when the user is not logged in the population density chart is still shown but without any data.

User guide

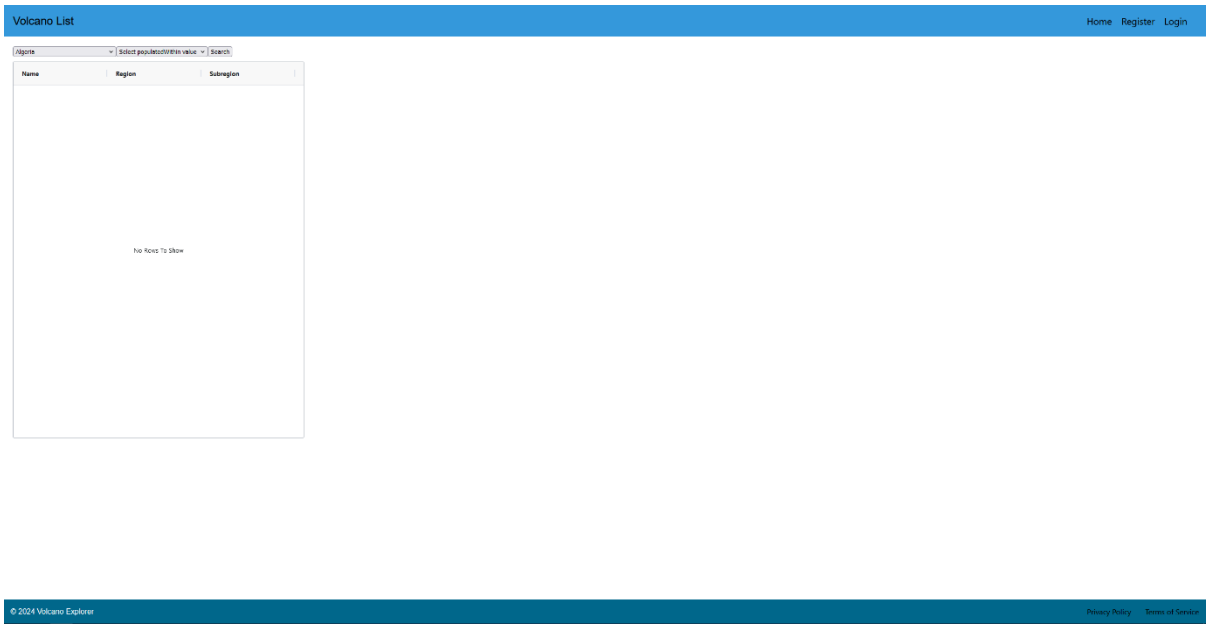


1. From the Home page, users can choose to navigate to either the volcano list, the registration page, or the login page. For this user guide, navigate to the registration page.

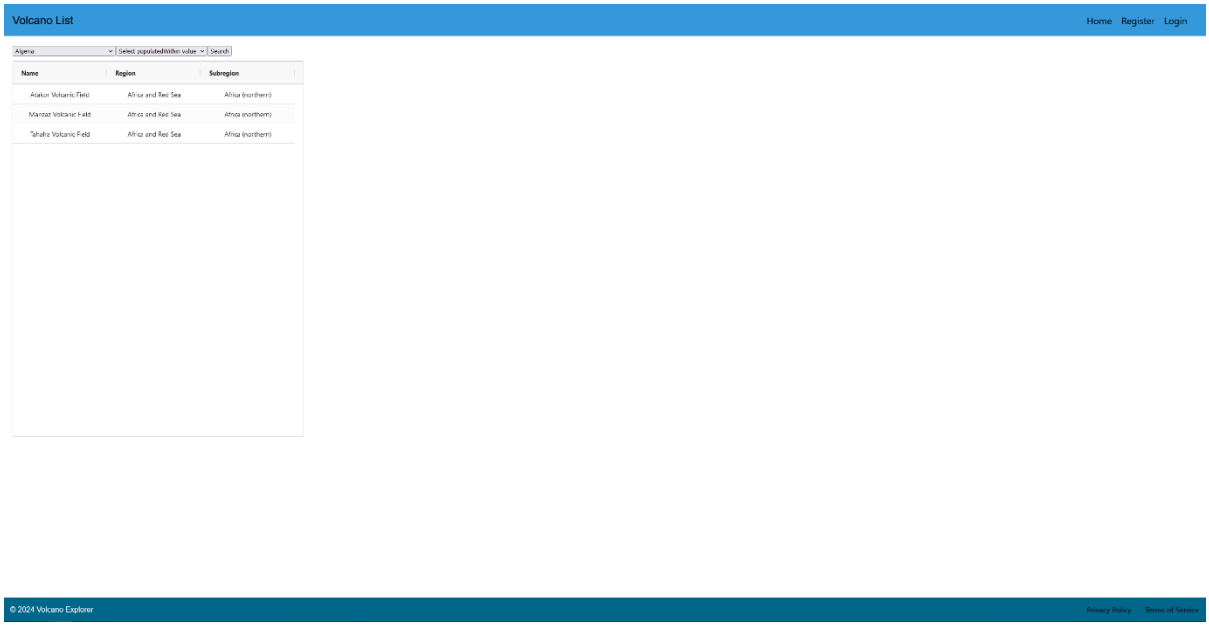


2. In the registration page, fill out the email and password fields, and select the register button.

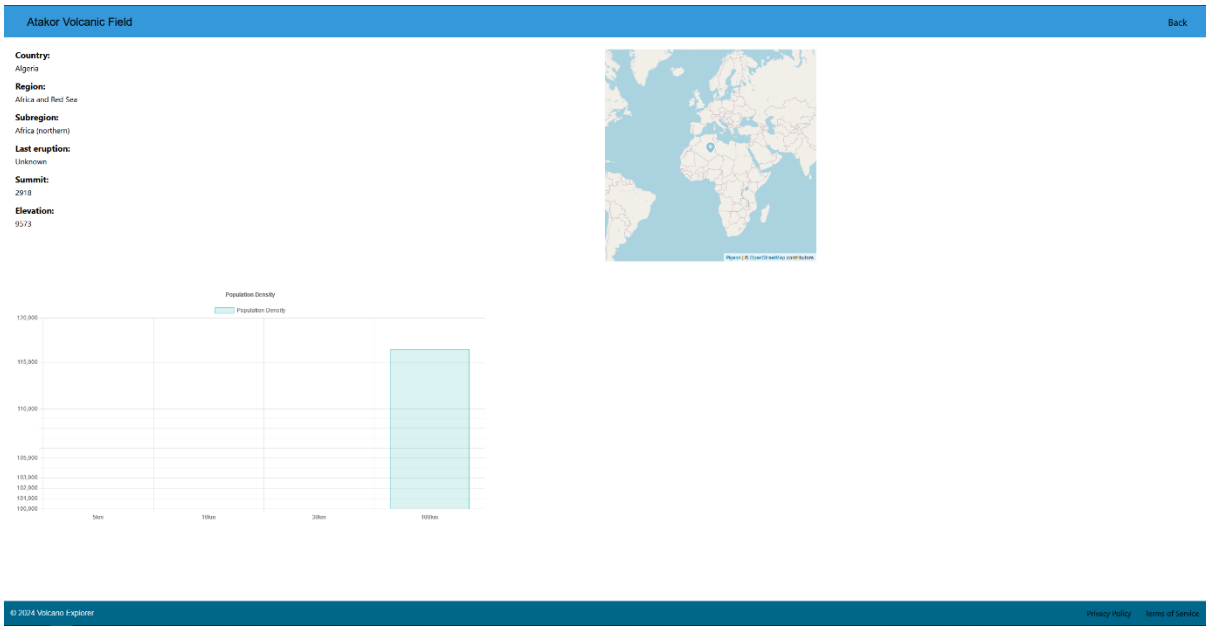
3. After successful registration, the user is redirected to the login page where they can fill in their login details.



4. After the user is logged in, they can choose to view the volcano list and select a country and a populatedWithin value.



5. After the user has clicked the search button, their results are displayed in the table. The user can then choose which volcano to view on an individual page.



6. The individual volcano page is displayed after the user has selected a volcano to view, and once done browsing can click back to be returned to the volcano list.

References

(2021). W3.org. https://www.w3.org/TR/WAI-WEBCONTENT/full-checklist