

**Student name: Andrew Clarke**

**Student number: N11270179**

## **Introduction**

### **Purpose and Description**

This webapp is designed to produce information about volcanoes around the world to users in an effective and visually appealing design. The webapp is entirely designed around presenting historical volcano data, including the location and population density around volcanic sites.

Users can access a comprehensive list of volcanoes through the website, and by registering can access additional features, such as the population density around selected volcano sites.

The design of this webapp featured many off-the-shelf React libraries creating an easier developer experience which allows for greater iterative improvements in future. The user was provided with the ability to search for volcanoes by region, as well as optionally searching by subregion if the selected region features multiple subregions. Multiple modules were utilised to make the application look more cohesive and to make it more efficient for developers by using React bootstrap and bootstrap CSS for a uniform design.

Volcanos

localhost:4173

Home Volcano list logout

## Volcano Insights: Explore the Earth's Fiery Heart

Delve into the heart of volcanic activity with our webapp, providing real-time data and comprehensive analysis tools for researchers and enthusiasts alike. Uncover the secrets of volcanic behavior and gain valuable insights into one of nature's most captivating phenomena.



Home Check out the volcanos

© 2024 Andrew Clarke

Volcanos

localhost:4173/volcano/21

Home Volcano list Register Login

**Name:** Adams

Country: United States

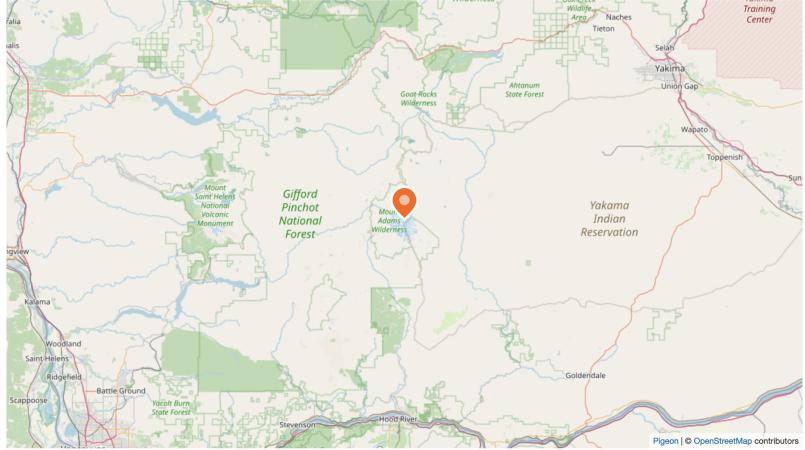
Region: Canada and Western USA

Sub region: USA (Washington)

Last eruption: 950 CE

Summit: 3742

Elevation: 12277



Home Check out the volcanos Login here Register your account here

© 2024 Andrew Clarke

## Completeness and Limitations

My application includes the following features:

Grade 4: I have built a React based application which utilises the data endpoints laid out in the assignment specification and presents the data in a minimal, readable manner. The application uses appropriate navigation.

Grade 5: My application has successfully implemented the user registration and login endpoints, and the authenticated data endpoint provided within the assignment specification.

The table component utilised does not excessively query the server. Additionally, the website features a minimal, clean design that is easy to navigate. I have also implemented a map and a chart component as required.

Grade 6-7: All the basics specified to meet a mark of grade 5 have been included and are working smoothly. Additionally, React Router has been utilised to handle navigation, along with controlled forms for input. The map and chart components are working effectively, with the chart component effectively showing the population density data.

Based on this assessment, my assessment of my application aligns with the requirements set out at the grade 6-7 level.

---

## **Use of End Points**

**/countries**

Provides an array of all available counties.

A screenshot of a web browser window titled "Volcanos". The address bar shows "localhost:4173/volcanos". The main content area has a header "Volcanos" and includes search fields for "Search country..." and "Select distance". Below these are four columns: "Name", "Country", "Region", and "Subregion". A message "No data" is displayed. At the bottom, there are links to "Home", "Check out the volcanos", and copyright information "© 2024 Andrew Clarke".

## /volcanoes

Provides information about volcanoes based on the country requested, with an optional parameter to retrieve information about the population density within a specified distance of the

volcano.

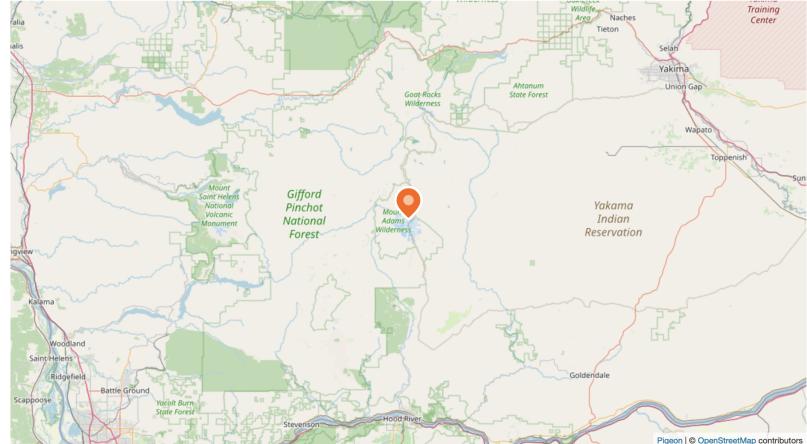
Name	Country	Region	Subregion
Adams	United States	Canada and Western USA	USA (Washington)
Crater Lake	United States	Canada and Western USA	USA (Oregon)
Craters of the Moon	United States	Canada and Western USA	USA (Idaho)
Agriyan	United States	Japan, Taiwan, Marianas	Izu, Volcano, and Mariana Islands
Buldir	United States	Alaska	Aleutian Islands
Ahyi	United States	Japan, Taiwan, Marianas	Izu, Volcano, and Mariana Islands
Asuncion	United States	Japan, Taiwan, Marianas	Izu, Volcano, and Mariana Islands
Daikoku	United States	Japan, Taiwan, Marianas	Izu, Volcano, and Mariana Islands
Buzzard Creek	United States	Alaska	Alaska (eastern)
Atka Volcanic Complex	United States	Alaska	Aleutian Islands

## /volcano/{id}

The endpoint provides an easy way to find all location about a particular volcano based on the identification number provided, allowing for users to search a database of 1343 volcanoes.

Volcanos

Name: Adams  
Country: United States  
Region: Canada and Western USA  
Sub region: USA (Washington)  
Last eruption: 950 CE  
Summit: 3742  
Elevation: 12277



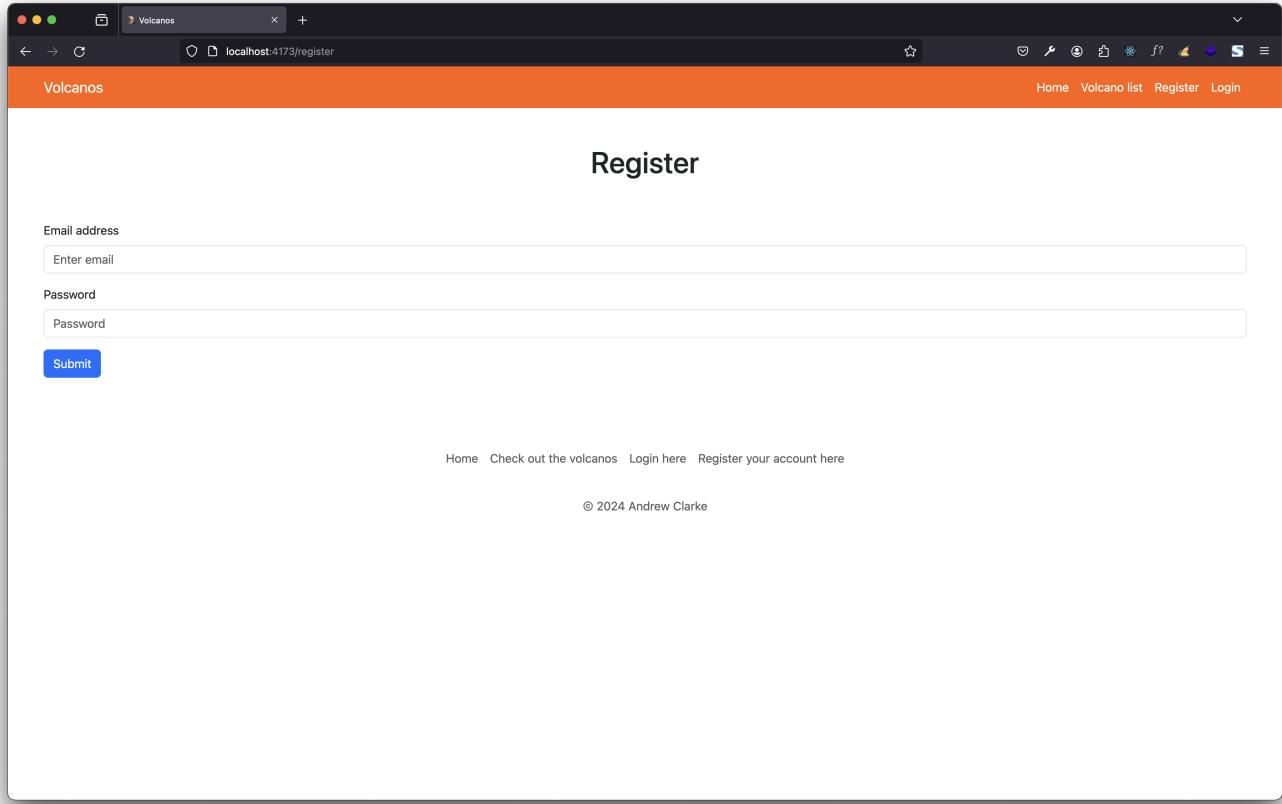
Pigeon | © OpenStreetMap contributors

[Home](#) [Check out the volcanos](#) [Login here](#) [Register your account here](#)

© 2024 Andrew Clarke

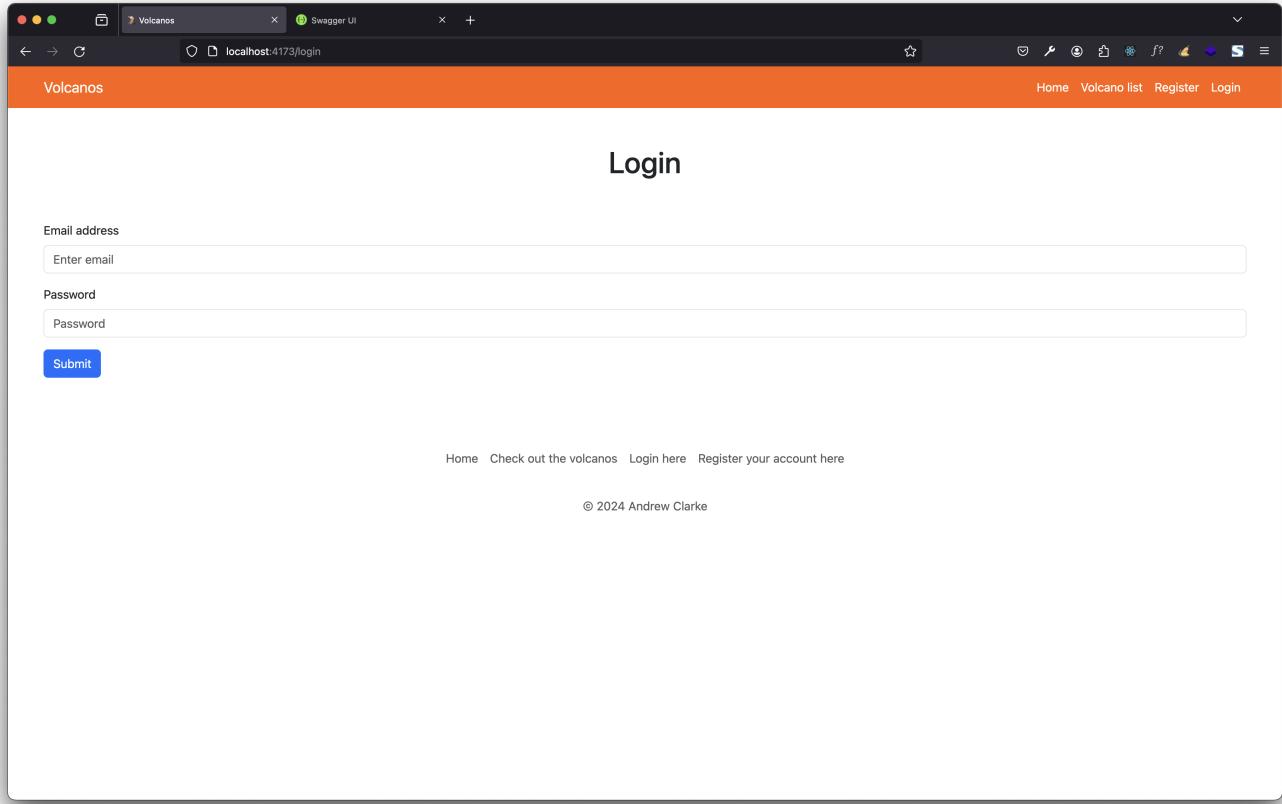
## /user/register

Allows user to register an account by providing an email and a password and sends the data to the API.



## /user/login

Allows user to login to their account by providing their email and password to receive an authentication token for future authentication requests.



## Modules used

### react

A front end library for to make modern single pages applications and reactive webpages.

<https://react.dev/> (Meta Open Source, 2024)

<https://www.npmjs.com/package/react> (Node Package Manager, 2024e)

### react-router-dom

A react extension framework that provides client side routing for single page applications

<https://reactrouter.com/en/main> (React Router, n.d.)

<https://www.npmjs.com/package/react-router-dom> (Node Package Manager, 2024f)

### bootstrap

A CSS framework for making websites with basic building blocks as well as providing responsive HTML and CSS layout.

<https://getbootstrap.com/> (Bootstrap, 2022)

<https://www.npmjs.com/package/bootstrap> (Node Package Manager, 2024b)

### bootstrap-icons

Provides icon for the parent framework of bootstrap.

<https://icons.getbootstrap.com/> (Bootstrap, n.d.)

<https://www.npmjs.com/package/bootstrap-icons> (Node Package Manager, 2024a)

## **react-bootstrap**

Provides a simpler way to work with both react and bootstrap with components.

<https://react-bootstrap.netlify.app/> (React Bootstrap, n.d.)

<https://www.npmjs.com/package/react-bootstrap> (Node Package Manager, 2024c)

## **chart.js**

<https://www.chartjs.org/> (chart.js, 2019)

<https://www.npmjs.com/package/chart.js?activeTab=readme> (Node Package Manager, 2023b)

## **pigeon-maps**

<https://pigeon-maps.js.org/> (Pigeon Maps, n.d.)

<https://www.npmjs.com/package/pigeon-maps> (Node Package Manager, 2024d)

## **react-select**

<https://react-select.com/home> (React Select, n.d.)

<https://www.npmjs.com/package/react-select> (Node Package Manager, 2023a)

## **react-chartjs-2**

<https://react-chartjs-2.js.org/> (React-chartjs-2, n.d.)

<https://www.npmjs.com/package/react-chartjs-2> (Node Package Manager, 2019)

Application Design

## **Navigation and Layout**

The layout for the application features a simple yet efficient design. A conscious decision to design the website with a minimal style was utilised, featuring a limited colour palette with a focus on functionality and readability for the user. The navigation bar is simple and effective, located in the same area with the same design on every webpage, with the only change being the menu item for login/logout changing depending on the user status. To make the design process easier and ensure the application featured consistent, clean features, frontend frameworks were used, including Bootstrap and React Bootstrap.

## **Usability and Quality of Design**

The display of the website features a clean and minimal, well organised design. The navigation is clear and intuitive, featuring consistent fonts, font sizes, and a cohesive colour palette across all screens. The colour palette is minimal, allowing the users to focus on the information presented.

## Accessibility

According to W3C (W3C, 2021), Priority 1 Checkpoints outlined are must be satisfied by web content developers. The checkpoints are outlined below:

General	Yes	No
1. Provide a text equivalent for every non-text element.	Yes	
2. Ensure that all information conveyed with colour is also available without colour.		No
3. Clearly identify changes in the natural language of a document's text and any text equivalents.		No
4. Organise documents so they may be read without style sheets.	Yes	
5. Ensure that equivalents for dynamic content are updated when the dynamic content changes.	Yes	
6. Until user agents allow users to control flickering, avoid causing the screen to flicker.	Yes	
7. Use the clearest and simplest language appropriate for the site's content.	Yes	
Images	Yes	No
1. Provide redundant text links for each active region of a server-side image map.		No
2. Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	Yes	
Tables	Yes	No
1. For data tables, identify row and column headers.	Yes	
2. For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.	Yes	
Otherwise	Yes	No
1. If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.		No

Note: Anything else included within priority 1 otherwise listed was deemed not applicable to this application. These checkpoints have been taken directly from the source (W3C, 2021).

Based on the guidelines provided from W3C, the application does meet most of the major requirements that must be implemented by web content developers. However, some major checkpoints have not been met, meaning that accessibility for some users may not be available.

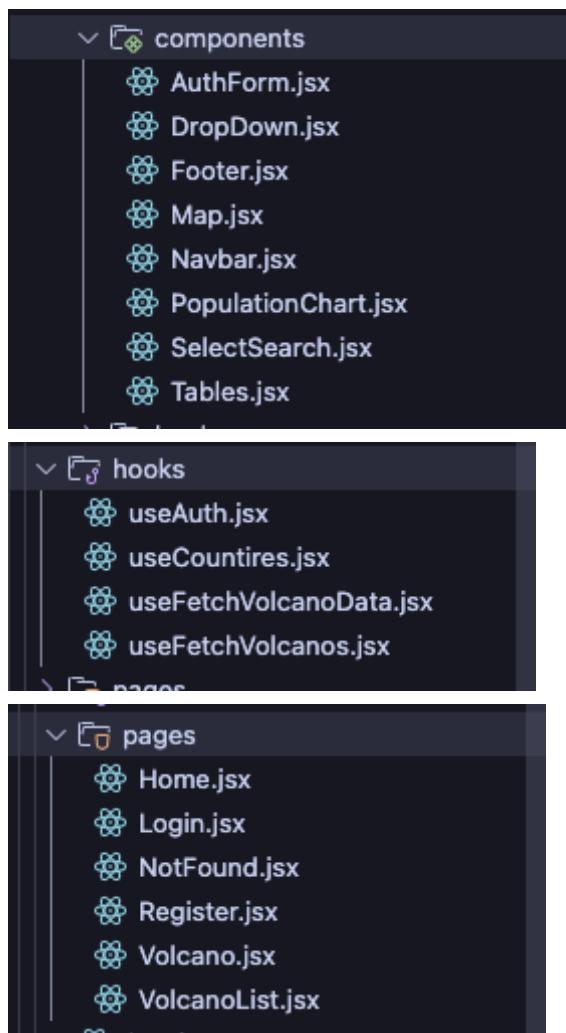
## Technical Description

### Architecture

The architecture I went with was a simple one based on the size of this web app. Everything happens in three main folders, first folder called components hold highly reusable atomic components. Second folder is the hooks folder, which holds all my custom hooks that deal with all the business logic for consuming the REST API such as login, and get countries as an example. The last folder pages is used to structure to the look of the page by using components, hooks and page specific logic that I could not justify making a hook.

The screenshot shows a file explorer interface with a dark theme. The root directory is 'client'. It contains several subfolders and files:

- 'dist'
- 'node\_modules'
- 'public'
- 'src'
  - 'components'
  - 'hooks'
  - 'pages'
    - 'App.jsx'
    - 'index.css'
    - 'main.jsx'
  - '.dockerignore'
  - '.env.development'
  - '.env.production'
  - '.eslintrc.cjs'
  - 'build.dockerfile'
  - 'dockerfile'
  - 'index.html'
  - 'makefile'
  - 'nginx.conf'
  - 'NOTES.md'
  - 'package-lock.json'
  - 'package.json'
  - 'README.md'
  - 'sbom.json'
  - 'vite.config.js'



## Test Plan

Task	Expected Outcome	Result
List number of results (/volcanoes)	Result count limited as per selection.	PASS
Click pagination arrow in table to load next page of results (/volcanoes)	Next set of results is displayed.	PASS
Click dropdown to display and select countries available (/volcanoes)	All available countries are displayed and the countries can be selected by the user one at a time.	PASS
Click table item to navigate to /volcano/{id} (/volcanoes)	Selecting table option will successfully navigate the user to the appropriate volcano page associated with its identification number.	PASS
Handle invalid url input	Display an error message for all urls not valid.	PASS
User can register successfully (/user/register)	User can input their email and password into the relevant fields and the data will be sent to the API.	PASS

Task	Expected Outcome	Result
User can login successfully (/user/login)	User can input their email and password into the relevant fields which will be sent to the API and will receive a token back.	PASS
User can logout successfully	User will be logged out of their account successfully, allowing for another user to log in.	PASS

## Difficulties/Exclusions/Unresolved & Persistent Errors

There were no major roadblocks that required resolution within this application. There was no functionality laid out in the assessment guidelines that were not finished, and there were no technical issues encountered. Additionally, there were no outstanding bugs within the application.

## References

Bootstrap. (n.d.). *Bootstrap Icons*. Icons.getbootstrap.com. <https://icons.getbootstrap.com/>

Bootstrap. (2022). *Bootstrap*. Getbootstrap.com. <https://getbootstrap.com/>

chart.js. (2019). *Chart.js | Open source HTML5 Charts for your website*. Chartjs.org. <https://www.chartjs.org/>

Grypachevska, T. (2021). Recent Fagradalsfjall volcanic eruption in Iceland. In *Unsplash.com*. <https://unsplash.com/photos/brown-and-black-mountain-under-white-clouds-80x3QULJDN4>

Meta Open Source. (2024). *React*. React.dev. <https://react.dev/>

Node Package Manager. (2019, September 20). *react-chartjs-2*. Npm. <https://www.npmjs.com/package/react-chartjs-2>

Node Package Manager. (2023a, November 6). *react-select*. Npm. <https://www.npmjs.com/package/react-select>

Node Package Manager. (2023b, December 4). *chart.js*. Npm. <https://www.npmjs.com/package/chart.js?activeTab=readme>

Node Package Manager. (2024a, January 3). *bootstrap-icons*. Npm. <https://www.npmjs.com/package/bootstrap-icons>

Node Package Manager. (2024b, February). *bootstrap*. Npm. <https://www.npmjs.com/package/bootstrap>

Node Package Manager. (2024c, March). *react-bootstrap*. Npm.

<https://www.npmjs.com/package/react-bootstrap>

Node Package Manager. (2024d, April 19). *pigeon-maps*. Npm.

<https://www.npmjs.com/package/pigeon-maps>

Node Package Manager. (2024e, April 24). *react*. Npm. <https://www.npmjs.com/package/react>

Node Package Manager. (2024f, May 10). *react-router-dom*. Npm.

<https://www.npmjs.com/package/react-router-dom>

Pigeon Maps. (n.d.). *Demo | Pigeon Maps*. Pigeon-Maps.js.org. <https://pigeon-maps.js.org/>

React Bootstrap. (n.d.). *React Bootstrap | React Bootstrap*. React-Bootstrap.netlify.app.

<https://react-bootstrap.netlify.app/>

React Router. (n.d.). *Home v6.4.1*. Reactrouter.com. <https://reactrouter.com/en/main>

React Select. (n.d.). *React-Select*. React-Select. <https://react-select.com/home>

React-chartjs-2. (n.d.). *react-chartjs-2 | react-chartjs-2*. React-Chartjs-2.Js.org. <https://react-chartjs-2.js.org/>

W3C. (2021). *Checklist of Checkpoints for Web Content Accessibility Guidelines 1.0*. W3.org.

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