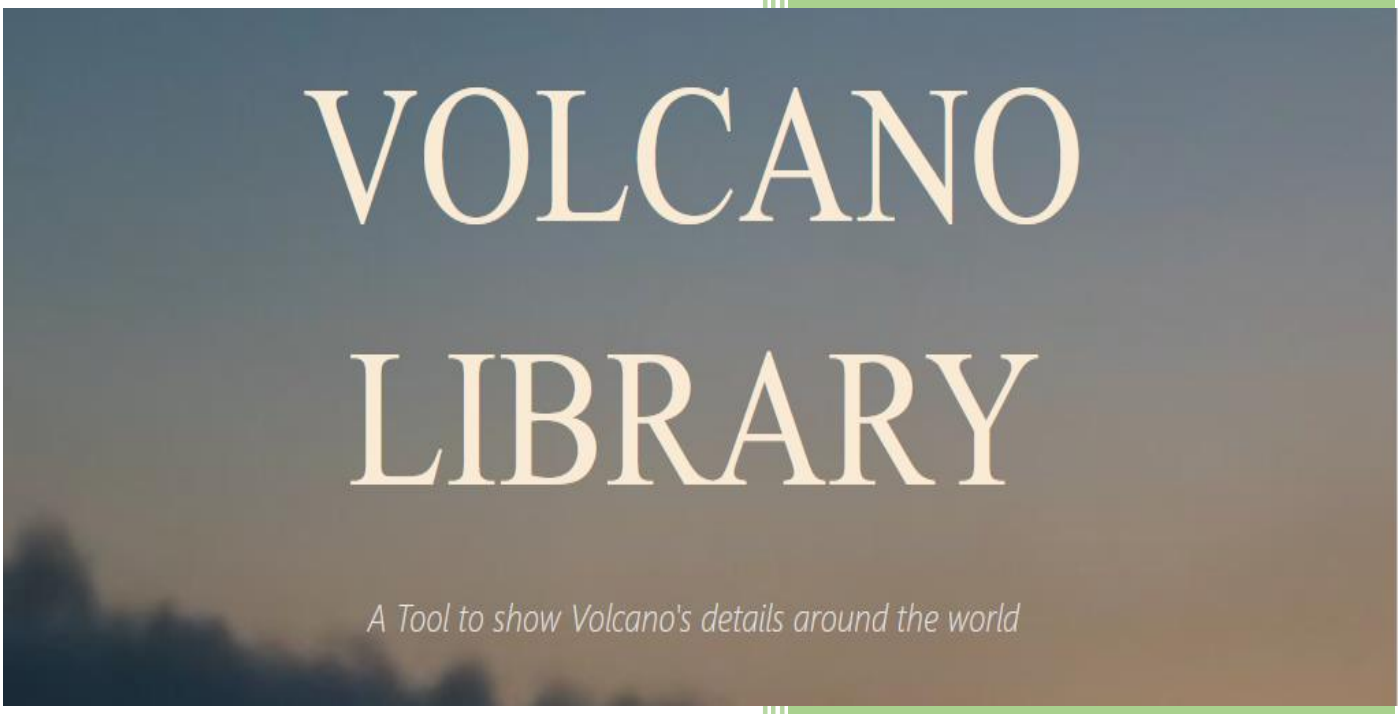
2022

CAB230 Assignment 1 Client Side



CAB230

Volcano API – Client Side Application

<Duy Pham (Daniel)>

<n10640754>

4/17/2022

Contents

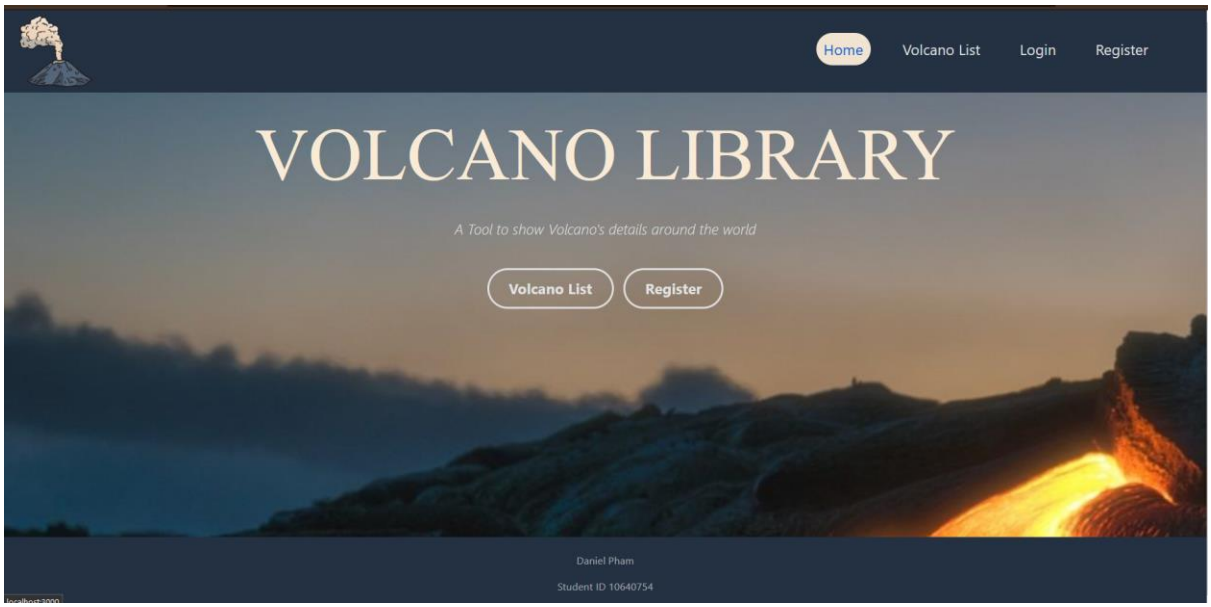
This template is adapted from one created for a more elaborate application. The original author spends most of his professional life talking to clients and producing architecture and services reports. You may find this a bit more elaborate than you are used to, but it is there to help you get a better mark

This report should be around 10 pages or so including screenshots – there is no formal page limit, and the length will depend a lot on the number of screen shots, but you won't get any extra marks for a really long report.

Introduction

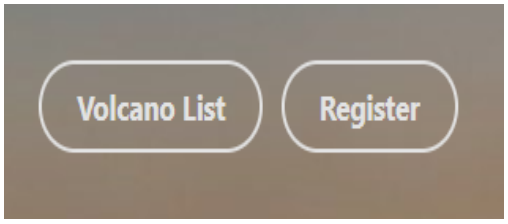
Purpose & description

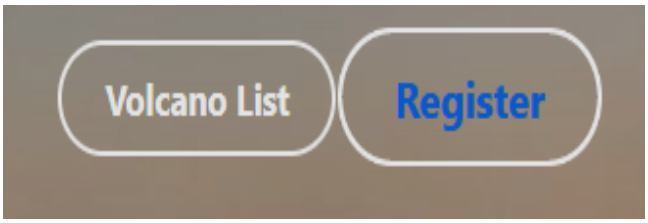
Volcano Library website is a React-based web application which allows users to achieve a decent data about one specific volcano or a list of volcanoes in a country in the world (with the assistance from a provided REST API.



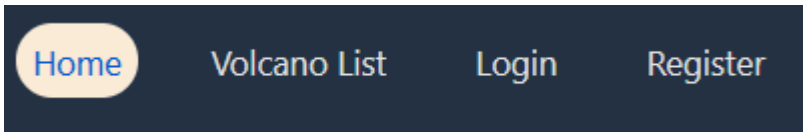
Beside qualified all requirements from assignment's criteria (in functionality perspective) the project also provides a user-friendly and modern design with the approach of using animation on button interactions. For example:

These 2 buttons in homepage got bigger when move-in the cursor





Navigation bar's buttons got hover effect when move-in cursor



The application was also used all recommended modules to build on perspectively functions (Using **pigeon-maps.js** for **map, ag-grid** for table of volcanoes and **react-chartjs-2** for **bar chart** of population density). The design of application is completely depended only on configuration of CSS properties, images and color pallete combinations.

Completeness and Limitations

Right now on the application the user can navigate through any pages demonstrated on navigation bar with an easy understanding of how the system works (handled by React Router). User can experience a search form (form provided with country options and distance options for population within) powered by Ag-Grid components, the user can also sort and filter any columns existed in the

table, for short, the ag-grid table is working smoothly with the search form and responsive filter options (to control user inputs). The website design is generally clean, uncluttered and there's close alignment between chosen components and the data they are displaying. Map component to display volcano location and a Bar Chart to show population density is successfully applied to the project (user can see the features when click on one of the volcano item in the returned list in Ag-Grid table). Furthermore, Login/Register/Logout system is working fine and logically link to each other by using page transition (for example: after successfully register the website will refresh to login page straight away and if Login successfully some components will change or disappeared or updated to be showed), all error during registering and logging in will pop up.

Use of End Points

/countries

When user click to the country name selector

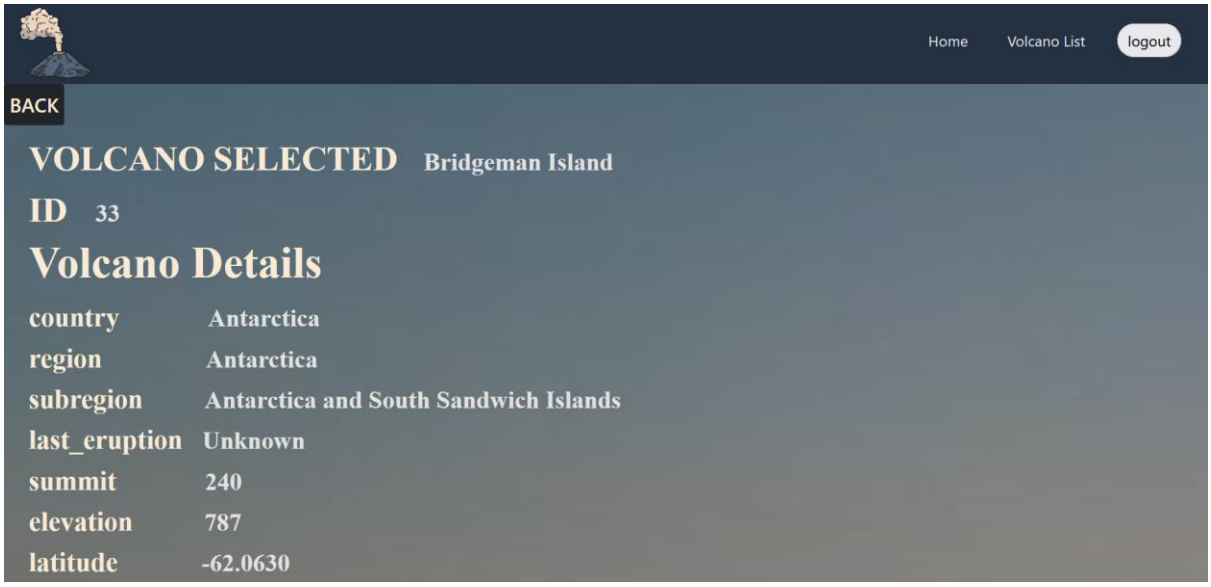
```
//get countries option
useEffect(() => {
    fetch(`http://sefdb02.qut.edu.au:3001/countries`)
    .then(res => res.json())
    .then(res => setCountryNames(res))
}, []);
```

/volcanoes

When user press search with corresponded chosen options. The application will connect to this endpoint

```
//get volcanoes in chosen country option and distance option
useEffect(() => {
  fetch(`http://sefdb02.qut.edu.au:3001/volcanoes?country=${searchOption.countryName}&populatedWithin=${searchOption.distanceOption}`)
  .then(res => res.json())
  .then(res => res)
  .then(res =>
    res.map(volcano => {
      return{
        name: volcano.name,
        country: volcano.country,
        region: volcano.region,
        subregion: volcano.subregion,
        id: volcano.id
      }
    }))
    .then(volcanoes => setRowData(volcanoes));
}, [searchOption]);
```

/volcano/{id}

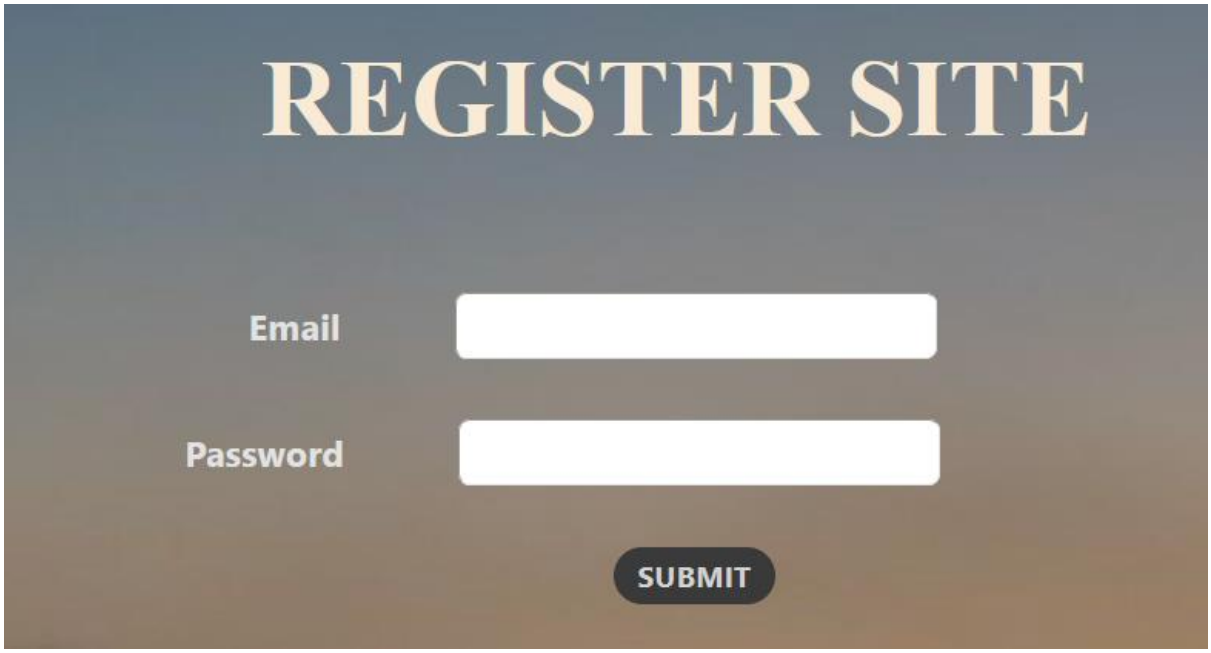


When user click-on one of the volcano item in the table

```
//get volcano from provided ID
function fetchVolcano () {
  const url = `http://sefdb02.qut.edu.au:3001/volcano/${volcanoID}`;
  const headers = {
    accept: "application/json",
    "Content-Type": "application/json",
    Authorization: `Bearer ${token}`
  }

  if(token != `undefined`)
  {
    return fetch(url,{headers})
    .then((res) => res.json())
    .then((res) => res)
  }
  else if (token == "undefined"){
   return fetch(url)
   .then((res) => res.json())
   .then((res) => res)
  }
}
```

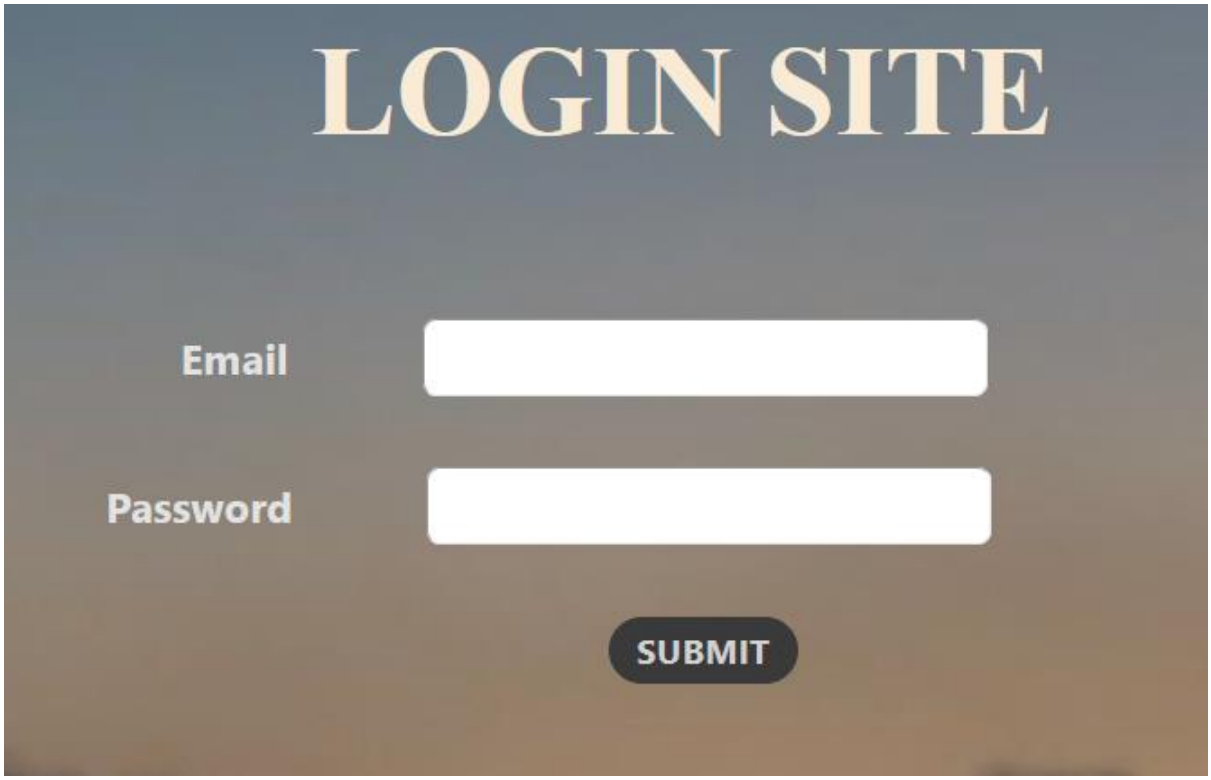
/user/register



When user click-on submit button

```
//post register request
function postRegister(){
  fetch( 'http://sefdb02.qut.edu.au:3001/user/register', {
    method: 'POST',
    headers: {
      accept: 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      "email": registerData.email,
      "password": registerData.password,
    })
  })
  .then(res => res.json())
  .then(data => setResponse(data))
}
```

/user/login



When user click-on submit button

```
//POST method
function postLogin() {
  fetch( 'http://sefdb02.qut.edu.au:3001/user/login', {
    method: 'POST',
    headers: {
      accept: 'application/json',
      'Content-Type': 'application/json',
    },
    body: JSON.stringify({
      "email": loginData.email,
      "password": loginData.password,
    })
  })
  .then(res => res.json())
  .then(data => setResponse(data))
}
```

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-dom

Provides dynamic routing for web application

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

React-chartjs-2

charting library for react web dev.

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

Pigeon-maps

provide a performance-first React-centric extendable map engine

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

React

allows developers to create large web applications that can change data, without reloading the page

https://reactjs.org/

Ag-grid-community

fully-featured and highly customizable JavaScript data grid.

https://www.npmjs.com/package/ag-grid-community

Bootstrap

front-end framework used to create modern websites and web apps.

https://getbootstrap.com/

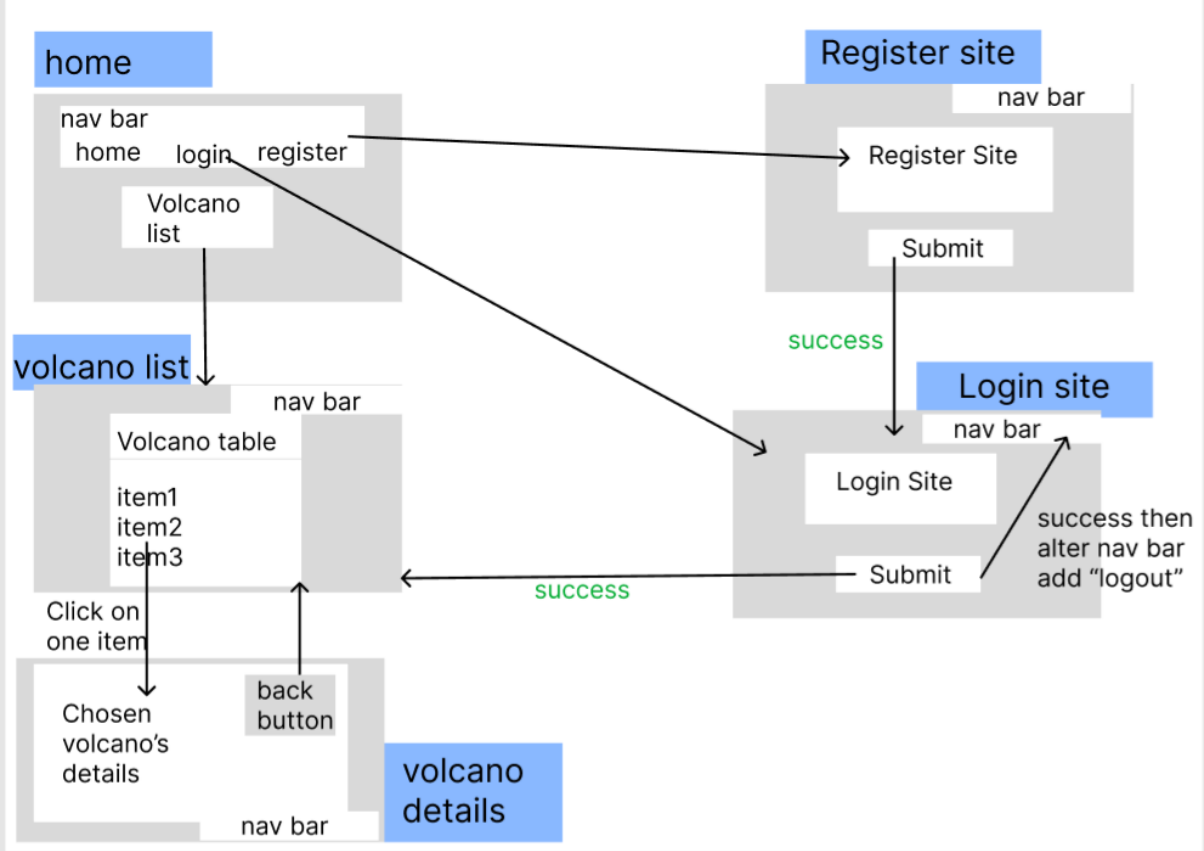
Reactstrap

Provide abilities to build forms

https://reactstrap.github.io/?path=/story/home-installation--page

Application Design

Navigation and Layout



As shown above (my sketch for mockup) for each page, it will have a header containing navigation bar (user can explore the map through here). Each navigation bar's button will lead to responsive page site. In special cases, with register site if user register successfully the site will automatically move the view to Login Site and if in login site user again successfully login it will automatically move to volcano list site, notice that with every wrong input (fail login/register) user will see an error output popping up. Furthermore, for the login site (in success login cases) navigation bar will be altered to have extra Logout button and also hide away register and login options.

Usability and Quality of Design

- The design can be considered as clean and modern. With a suitable consistent color palette, the overall web application brings a professional look for user. Arrangement of widget UIs is logical and clear. But in some parts, the design color is a little bit moldy and dark to see data (for example: the population bar chart's column color). Furthermore, the font size in each page is not very uniform which could bring an awkward experience for some users. In future, design could improve by having a more professional design sheet with a better color scheme and contrast
- Navigation has been done in a simple way and easy to understand the pattern. But could be noticed that there are some lacking of notifications to user between the transition of pages (from login page automatically switch to volcano list and from register page automatically switch to login site). In general, user still can understand of what's going on with the web environment and aware of what kind of information will be delivered when click to what button on navigation bar. The solution to improve from this issue would be experience more modern components from popular/trendy CSS frameworks (Foundation, Bulma,.etc.).

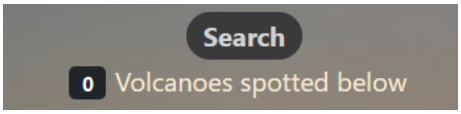
- In general, the application has met the requirements both on function's effectiveness and general design's aesthetic compared to other general web applications on internet.

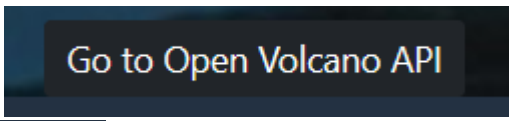
Accessibility

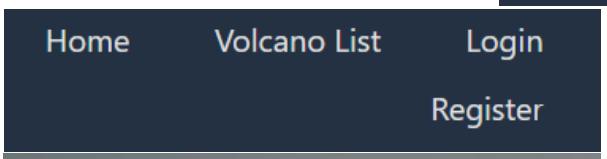
list of Priority 1 Accessibility Requirements:

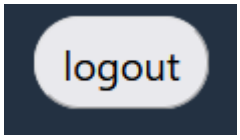
- Provide a text equivalent for every non-text element – alternatives to images, symbols, scripts, graphical buttons, sounds, audio and video files and so on. (**achieved**)
    - Applied on every selector 

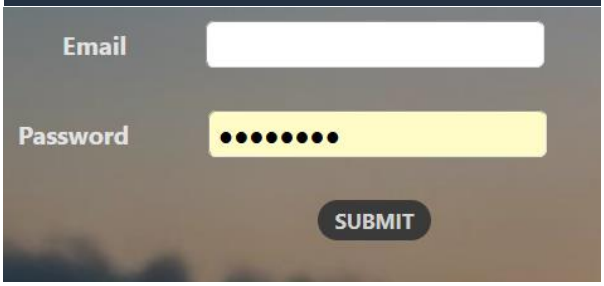
    - Buttons applied:



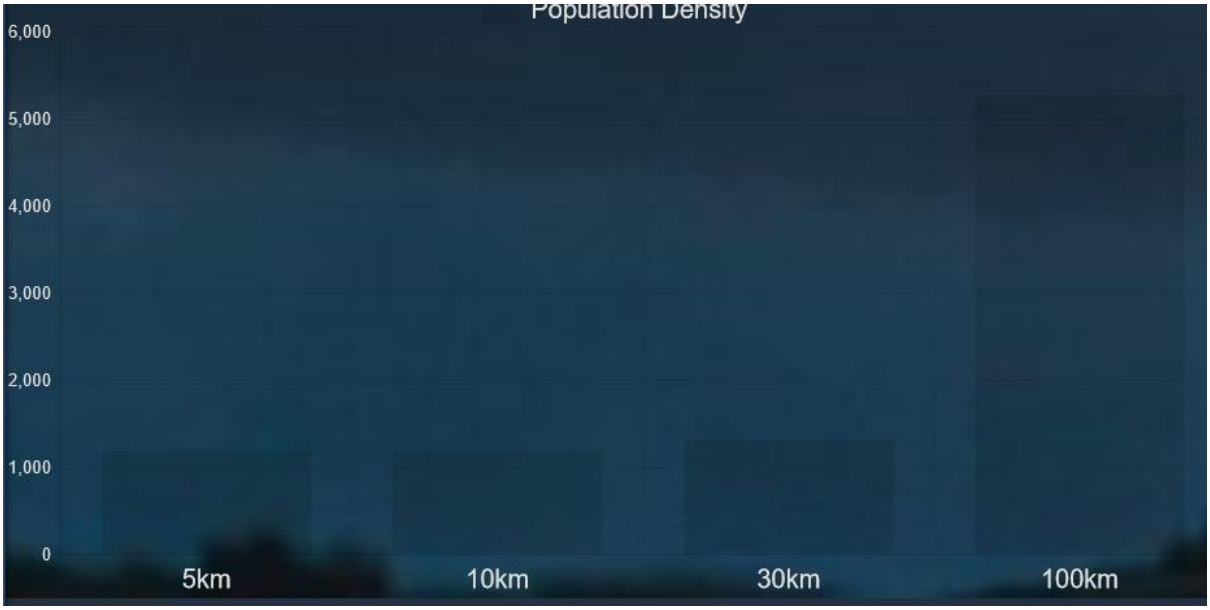




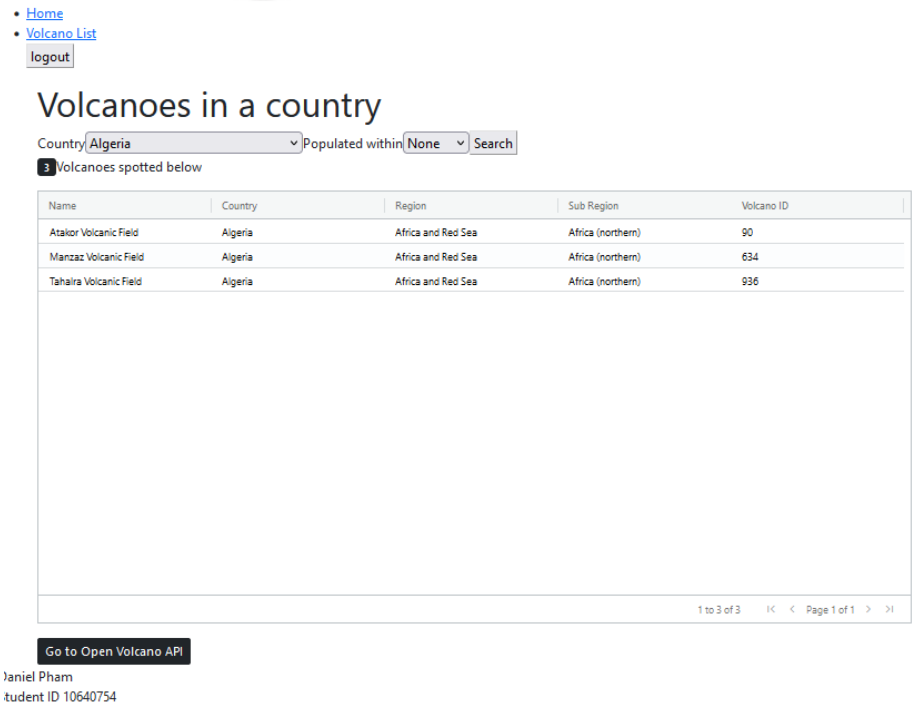


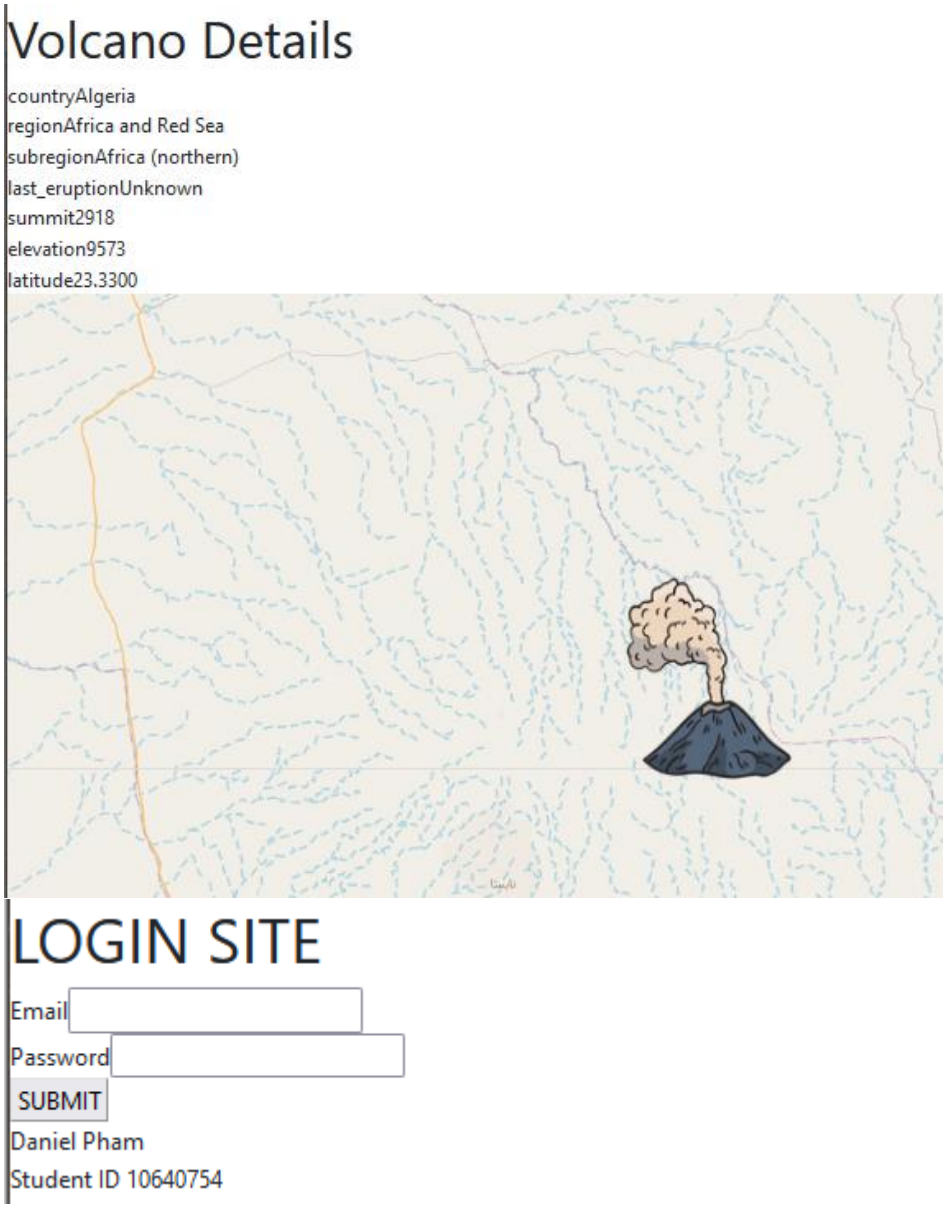


- Ensure that all information conveyed with color is also available without color, for example from context or markup. (**Mostly**)
  The bar chart can barely see without color (could be clearer without the background img)

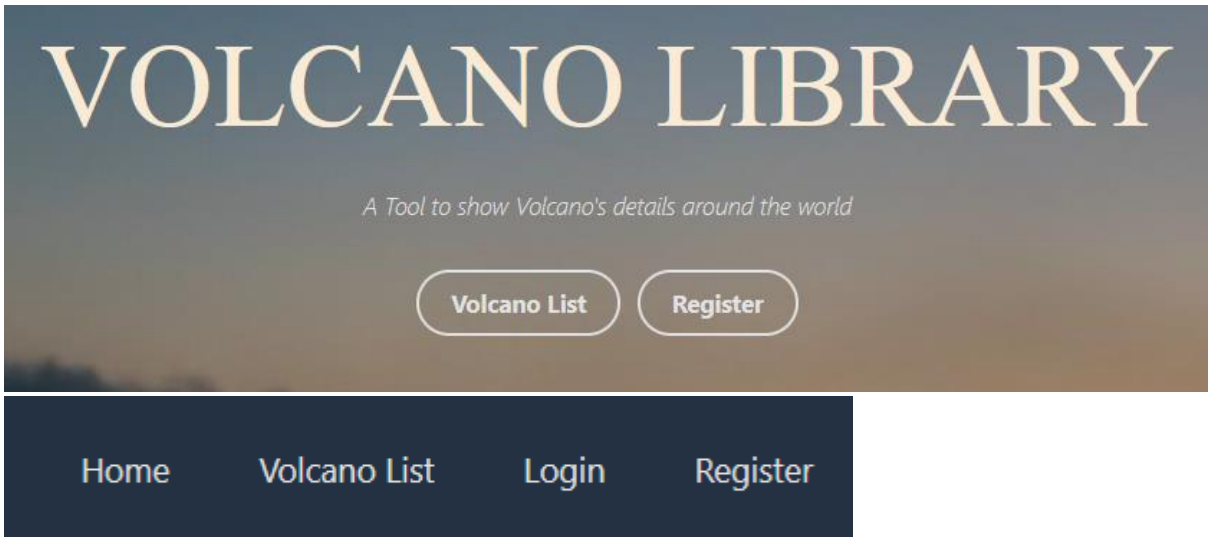


- Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document (**achieved**). below are example of component with details but without stylesheet

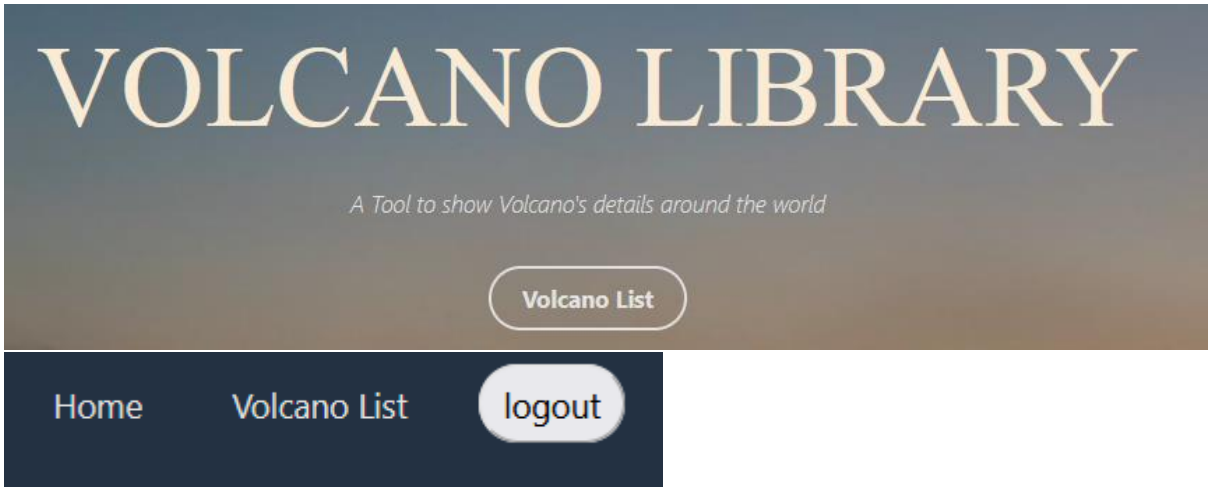




- Ensure that text equivalents are updated when dynamic content changes (**achieved**).
  When login the register and login will disappear but Logout button will show up
  Before:



After:



- Avoid causing the screen to flicker(**achieved**).
- Use the clearest and simplest language appropriate for a site's content(**achieved**).
- For tables, identify row and column headers – clearly differentiated from the data(**achieved**).

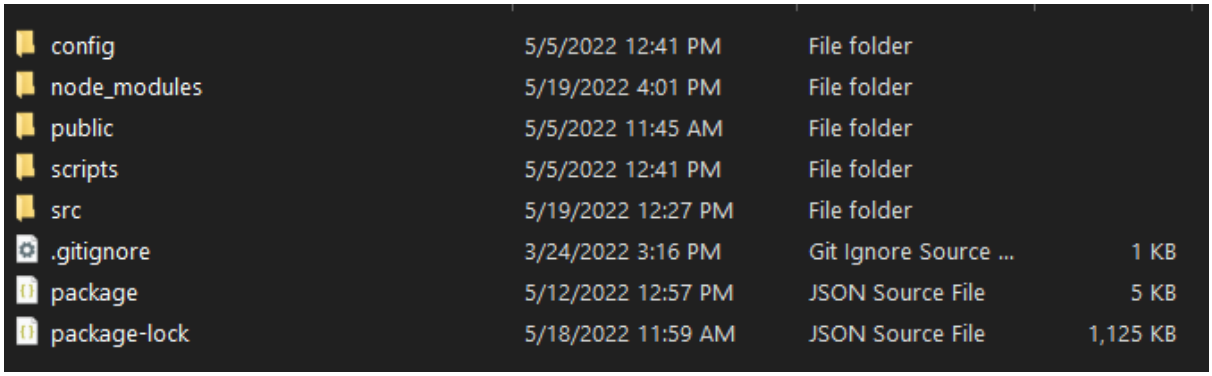
Name	Country	Region	Sub Region	Volcano ID
Abu	Japan	Japan, Taiwan, Marianas	Honshu	1
Aogashima	Japan	Japan, Taiwan, Marianas	Izu, Volcano, and Marian...	16
Adatarayama	Japan	Japan, Taiwan, Marianas	Honshu	30
Asamayama	Japan	Japan, Taiwan, Marianas	Honshu	65
Aira	Japan	Japan, Taiwan, Marianas	Ryukyu Islands and Kyus...	68
Akagisan	Japan	Japan, Taiwan, Marianas	Honshu	75
Asosan	Japan	Japan, Taiwan, Marianas	Ryukyu Islands and Kyus...	76
Akan	Japan	Japan, Taiwan, Marianas	Hokkaido	78
Ata	Japan	Japan, Taiwan, Marianas	Ryukyu Islands and Kyus...	83
Akita-Komagatake	Japan	Japan, Taiwan, Marianas	Honshu	85
Akita-Yakeyama	Japan	Japan, Taiwan, Marianas	Honshu	100
Akusekijima	Japan	Japan, Taiwan, Marianas	Ryukyu Islands and Kyus...	105
Azumayama	Japan	Japan, Taiwan, Marianas	Honshu	168
Bandaisan	Japan	Japan, Taiwan, Marianas	Honshu	203
Chachadake [Tiatia]	Japan	Kuril Islands	Kuril Islands	244

1 to 15 of 112 Page 1 of 8

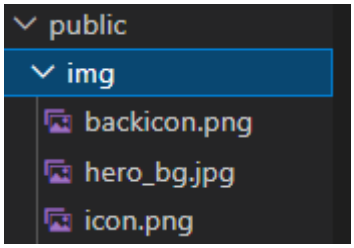
Technical Description

Architecture

Overall view of the application folder:

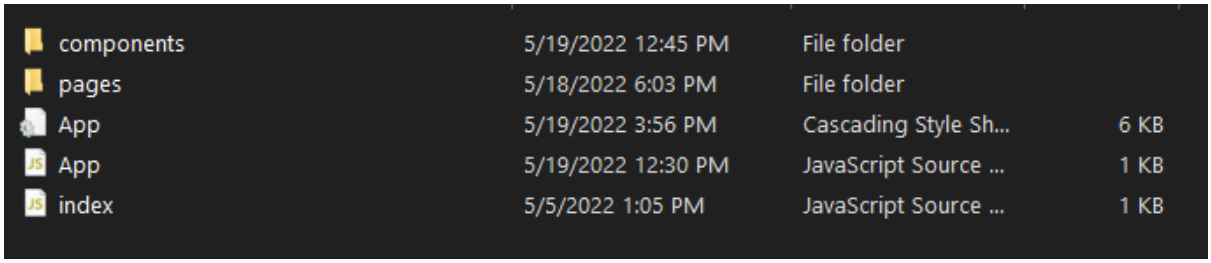


For **public folder** I used it stored img used for the application

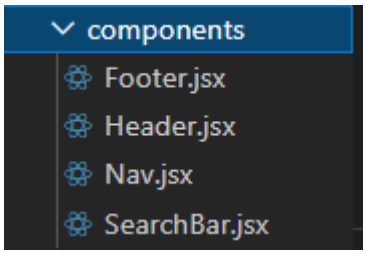


Source code folder:

I have divided into these main parts which are components, pages, App.js and App.css files



for the **components folder** I have stored all components that frequently appears across the web application (navigation bar, footer and header) and some other complicated and massive component like SearchBar.jsx



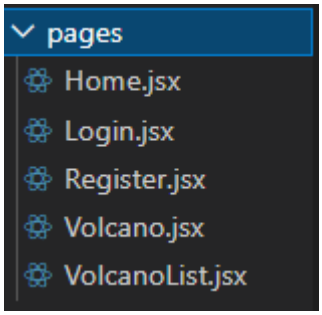
Index.js is just a support file to render app

```
import React from "react";
import ReactDOM from "react-dom";

import App from "./App.js";

const rootElement = document.getElementById("root");
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  rootElement
);
```

And for **pages folder** I have stored all pages that can navigate into for a better management



App.css for all the style sheets and finally was **App.js** is for keeping the routing source code looks clean

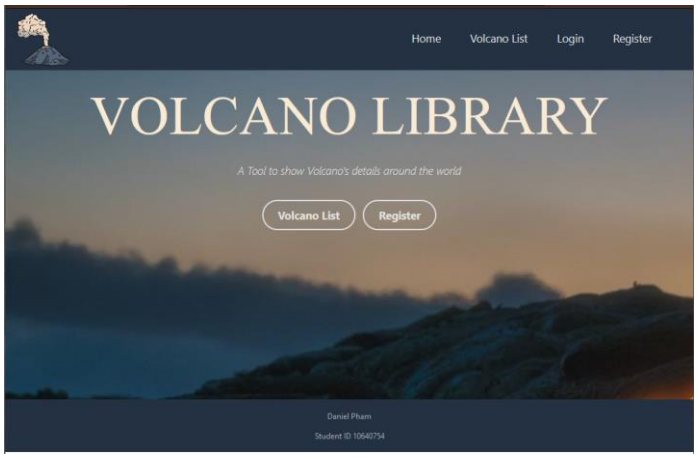
```
export default function App() {
  return (
    <BrowserRouter>
    <div className="App">
    <Header />
    <Routes>
              <Route path="/" element={<Home />} />
              <Route path="/login" element={<Login />} />
              <Route path="/register" element={<Register />} />
              <Route path="/volcanolist" element={<VolcanoList />} />
              <Route path="/volcano" element={<Volcano />}/>
    </Routes>
    <Footer />
    </div>
    </BrowserRouter>
  );
}
```

Test plan

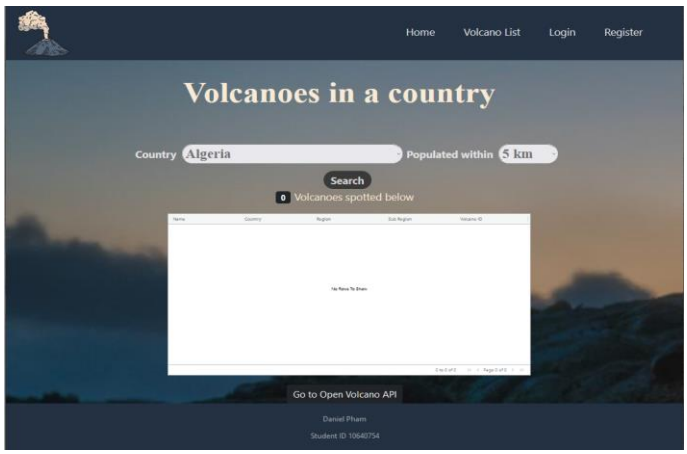
TASKS	Expected Outcome	RESULT	Screenshot/s (Appendix A)
Successfully display Home page	UI displayed with full buttons and texts	PASS	1
Successfully display Volcano list page	UI displayed with full buttons and texts	PASS	2
Successfully display Login/register page	UI displayed with full buttons and texts	PASS	3
Able to register new account	Can login with account just created	PASS	Backend (no image)
Able to login	Can unlock new content in Volcano page (bar chart) and alter navigation bar/home screen	PASS	4
Login error	pop up error if theres invalid/wrong input	PASS	5
Register error	pop up error if theres invalid/wrong input	PASS	6
Cut content	If not login navigation bar wont have logout button and no bar chart in volcano page	PASS	7
Able to choose country and population within km	2 selectors available with dropdown options	PASS	8
Able to search for volcano with provided information	Return list of volcanoes relative to information given (country option & poplation within KM option)	PASS	9
Return details of volcano when click in row data	Return Map and volcano details (switch to volcano page)	PASS	10
Map return correct volcano location	Return correct location of volcano on the map	PASS	11
Population density bar chart with correct data	Bar column of bar chart response correct to given data	PASS	12
Logout button works fine	Return navigation bar to normal and logout button disappear also hide away bar chart	PASS	1 + 7

Appendix A:

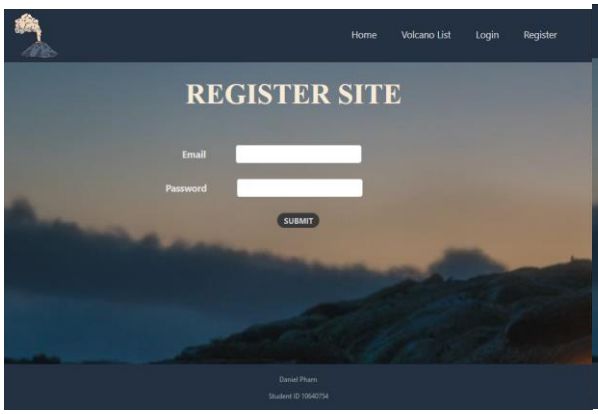
1:

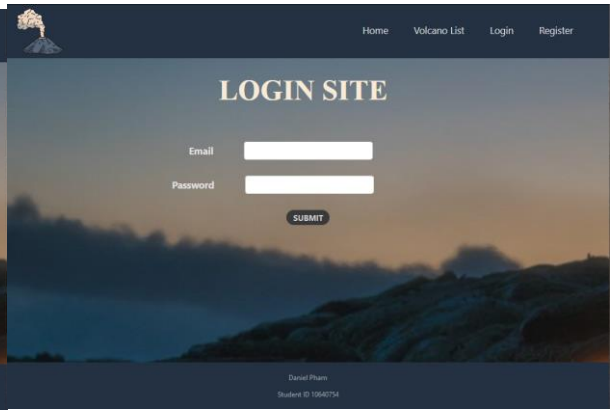


2:

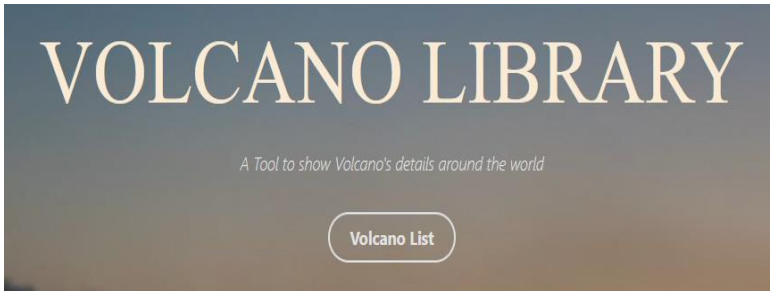


3:

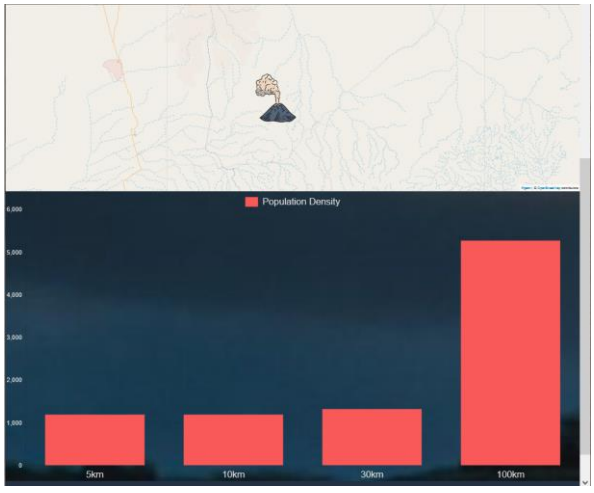

Home
Volcano List
Login
Register
REGISTER SITE
Email
Password
SUBMIT
Daniel Pham
Student ID 10640754


Home
Volcano List
Login
Register
LOGIN SITE
Email
Password
SUBMIT
Daniel Pham
Student ID 10640754

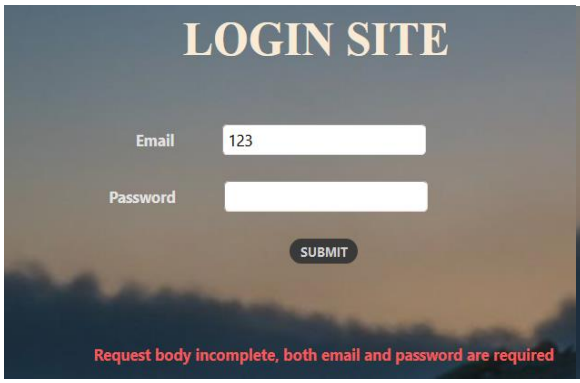
4:

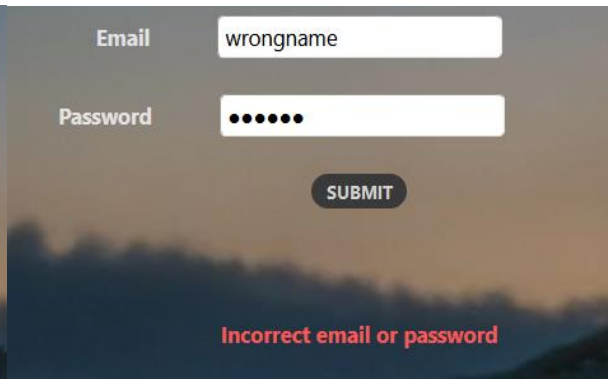

VOLCANO LIBRARY
A Tool to show Volcano's details around the world
Volcano List


Home
Volcano List
logout

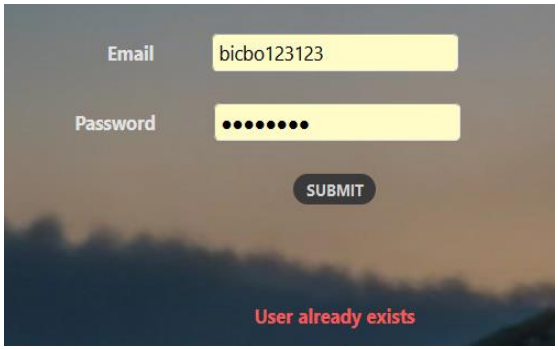

Population Density
5km
10km
30km
100km

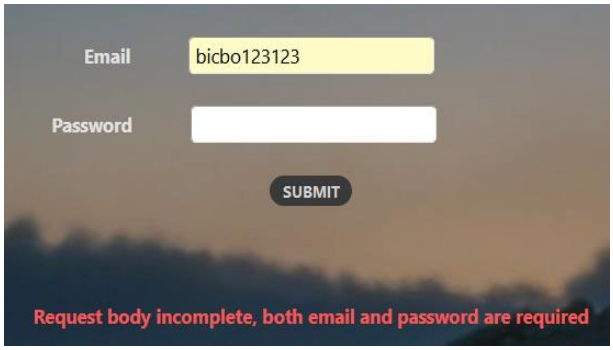
5:


LOGIN SITE
Email
123
Password
SUBMIT
Request body incomplete, both email and password are required

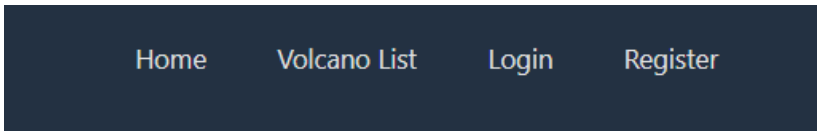

Email
wrongname
Password
SUBMIT
Incorrect email or password

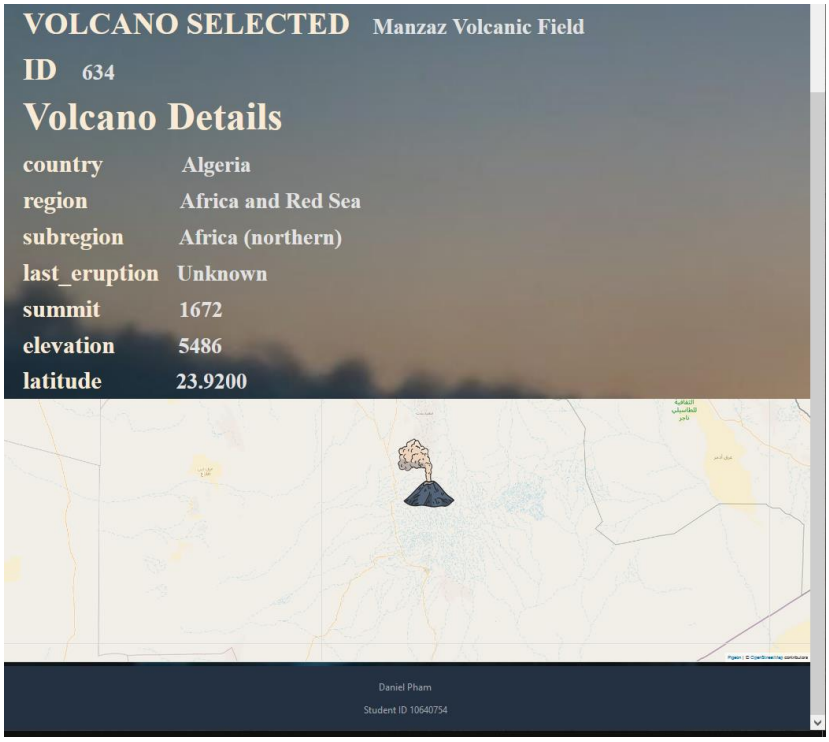
6:


Email
bicbo123123
Password
SUBMIT
User already exists

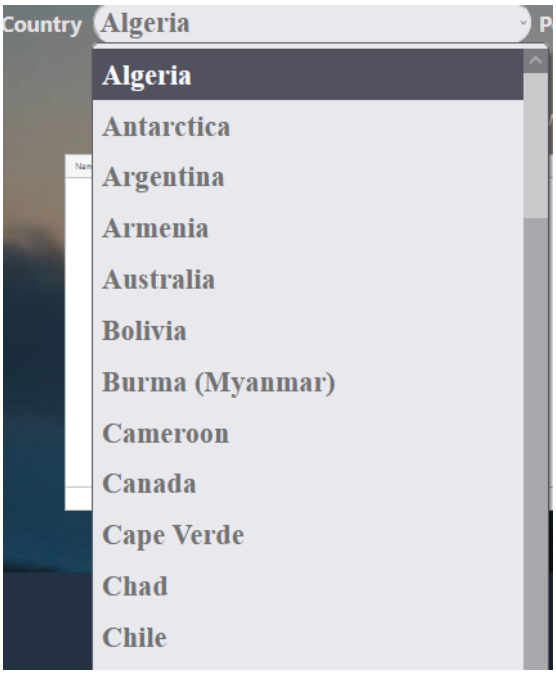

Email
bicbo123123
Password
SUBMIT
Request body incomplete, both email and password are required

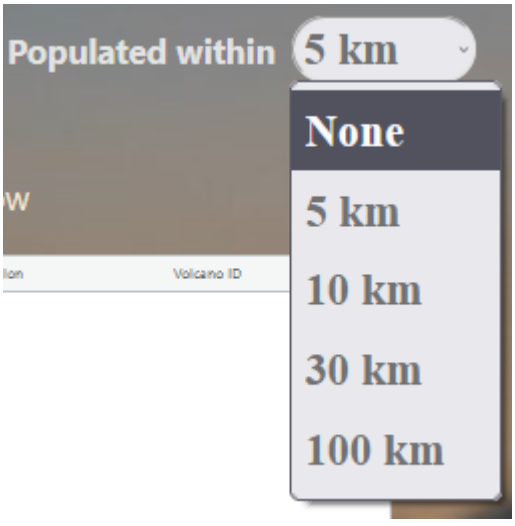
7:


Home
Volcano List
Login
Register

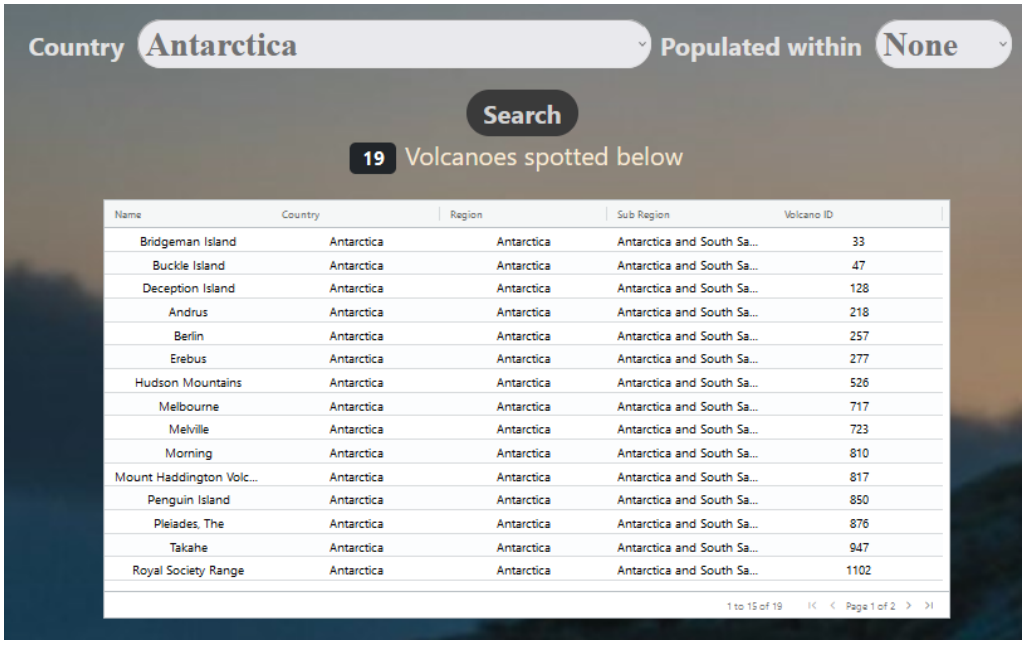

VOLCANO SELECTED Manzaz Volcanic Field
ID 634
Volcano Details
country Algeria
region Africa and Red Sea
subregion Africa (northern)
last_eruption Unknown
summit 1672
elevation 5486
latitude 23.9200
Daniel Pham
Student ID 10640754

8:

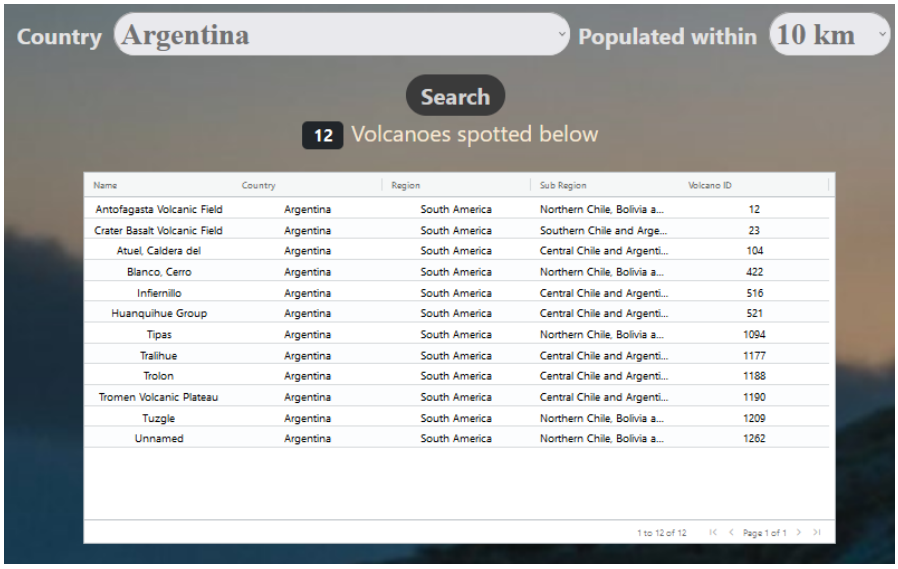

Country
Algeria
Algeria
Antarctica
Argentina
Armenia
Australia
Bolivia
Burma (Myanmar)
Cameroon
Canada
Cape Verde
Chad
Chile


Populated within
5 km
None
5 km
10 km
30 km
100 km
Volcano ID

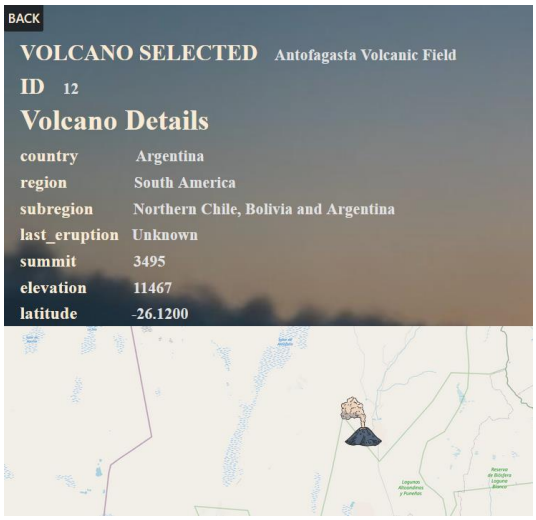
9:


Country
Antarctica
Populated within
None
Search
19
Volcanoes spotted below
Name
Country
Region
Sub Region
Volcano ID
Bridgeman Island
Antarctica
Antarctica
Antarctica and South Sa...
33
Buckle Island
Antarctica
Antarctica
Antarctica and South Sa...
47
Deception Island
Antarctica
Antarctica
Antarctica and South Sa...
128
Andrus
Antarctica
Antarctica
Antarctica and South Sa...
218
Berlin
Antarctica
Antarctica
Antarctica and South Sa...
257
Erebus
Antarctica
Antarctica
Antarctica and South Sa...
277
Hudson Mountains
Antarctica
Antarctica
Antarctica and South Sa...
526
Melbourne
Antarctica
Antarctica
Antarctica and South Sa...
717
Melville
Antarctica
Antarctica
Antarctica and South Sa...
723
Morning
Antarctica
Antarctica
Antarctica and South Sa...
810
Mount Haddington Volc...
Antarctica
Antarctica
Antarctica and South Sa...
817
Penguin Island
Antarctica
Antarctica
Antarctica and South Sa...
850
Pleiades, The
Antarctica
Antarctica
Antarctica and South Sa...
876
Takahe
Antarctica
Antarctica
Antarctica and South Sa...
947
Royal Society Range
Antarctica
Antarctica
Antarctica and South Sa...
1102
1 to 15 of 19
Page 1 of 2


Country
Antarctica
Populated within
5 km
Search
0
Volcanoes spotted below
Name
Country
Region
Sub Region
Volcano ID
No Rows To Show
0 to 0 of 0
Page 0 of 0

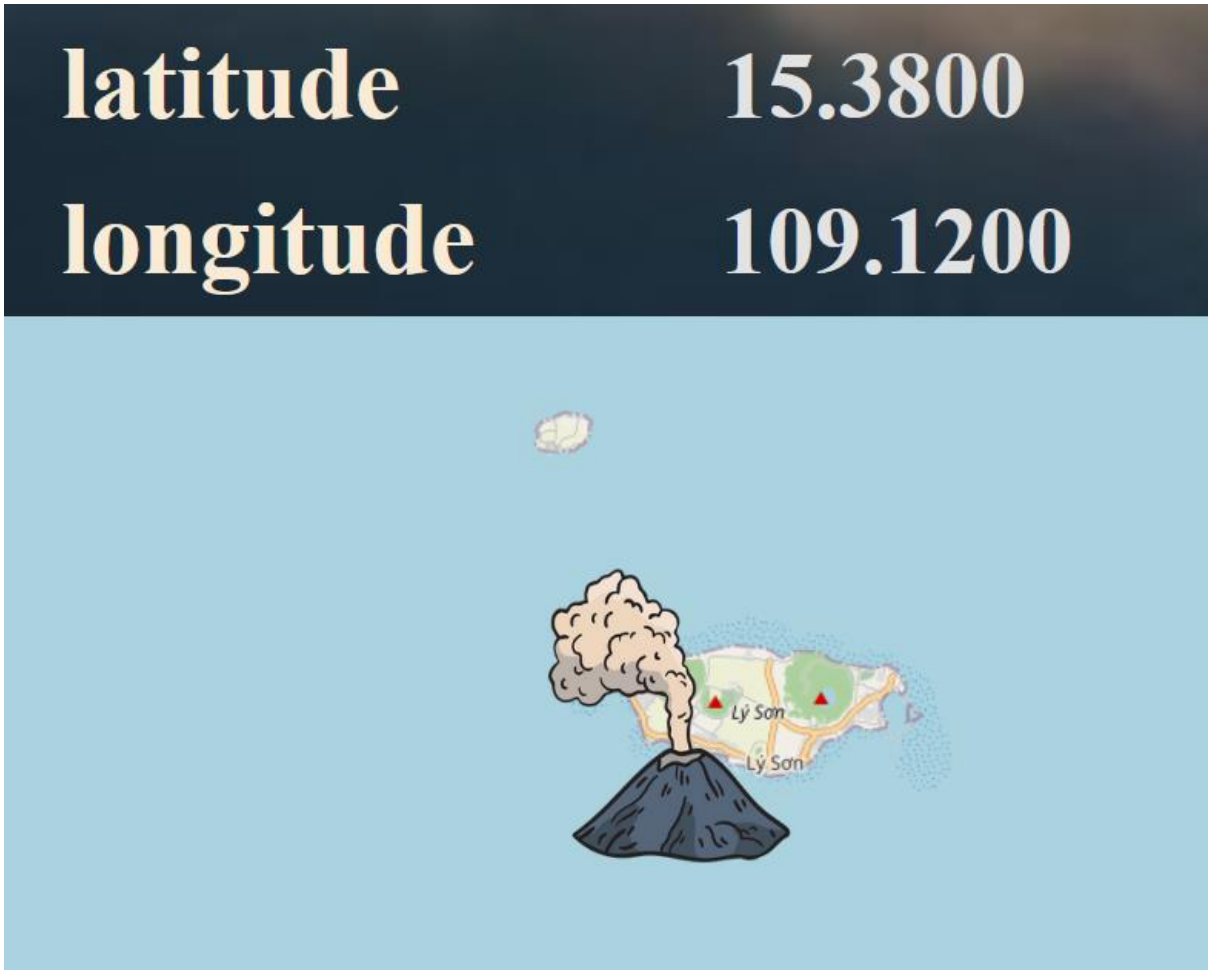


10:

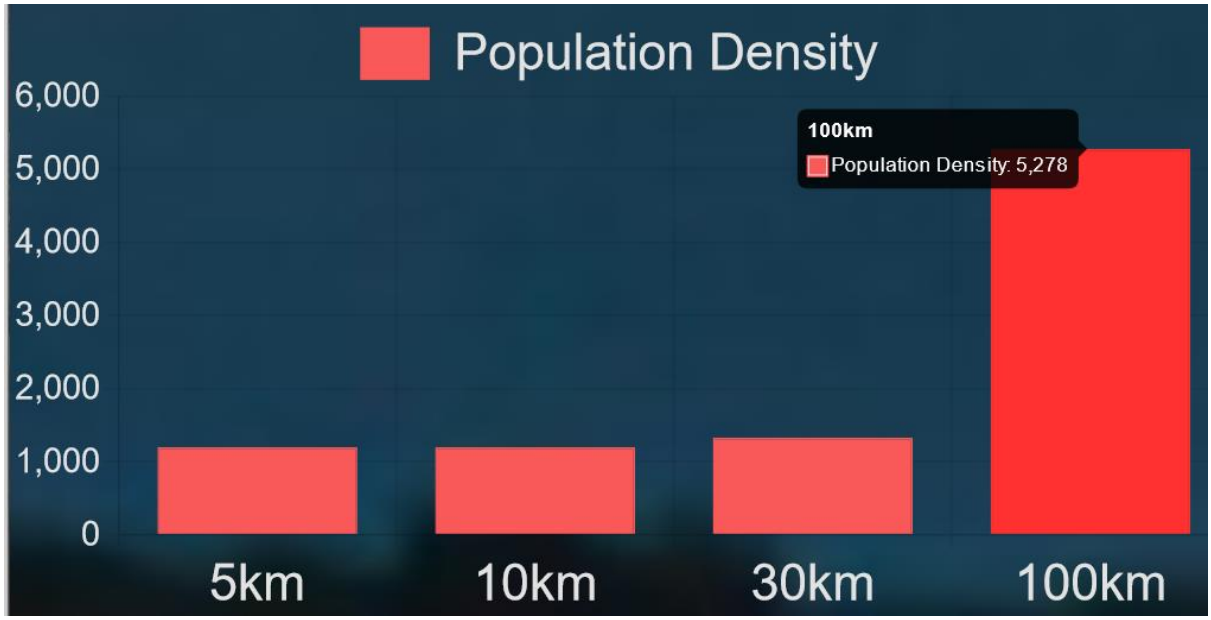


11:

Case: I choose a volcano in **Ly Son** and it responses with correct location


latitude
15.3800
longitude
109.1200
Lý Sơn
Lý Sơn

12:


Population Density
6,000
5,000
4,000
3,000
2,000
1,000
0
100km
Population Density: 5,278
5km
10km
30km
100km

Difficulties / Exclusions / unresolved & persistent errors /

Difficulties that I have faced was:

- How to show the application know that the user has logged in
  Solution: read tutorial slides and figure our using local storage to store token's contents so if the content is undefined meaning the user hasn't logged in and opposite.
- How to hide content when user hasn't logged in yet
  Solution: using ":null" syntax (for example:

```
{/* chart */}
{token!='undefined'?<Bar
data={state}
options={option_state}/>:null}
```

) the condition for **null is token's content**
- How to iterate through an object values (not all values just values in index 1 to 9) and return object value at the same time
  Solution: using **Object.keys()** and **slice()** functions, I want to do this to avoid code repeating

```
function Volcano(props) {
  return(
    <div className="Volcano">

      {Object.keys(props).slice(1,9).map(function(data, index) {
        return (
          <div>
            <label className={data}>{data}</label>
            <p1>{props[data]}</p1>
          </div>
        )})}
    </div>
  );
}
```

- I can solve mostly all technical issues by doing research on internet and tutorials on YouTube, that's also the reason why I luckily applied all required functions to my application. Furthermore, I also read docs of pigeon map and react-chartjs-2 to understand more about them and easily applied/customized them successfully.

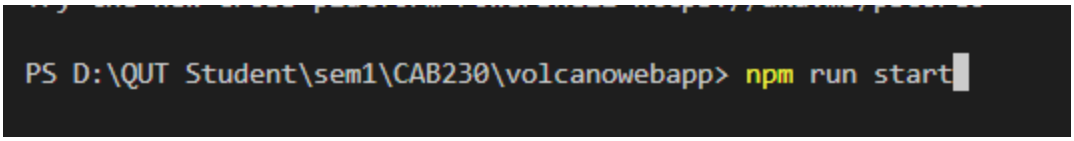
Extensions (Optional)

Potential future extensions / improvements for your app:

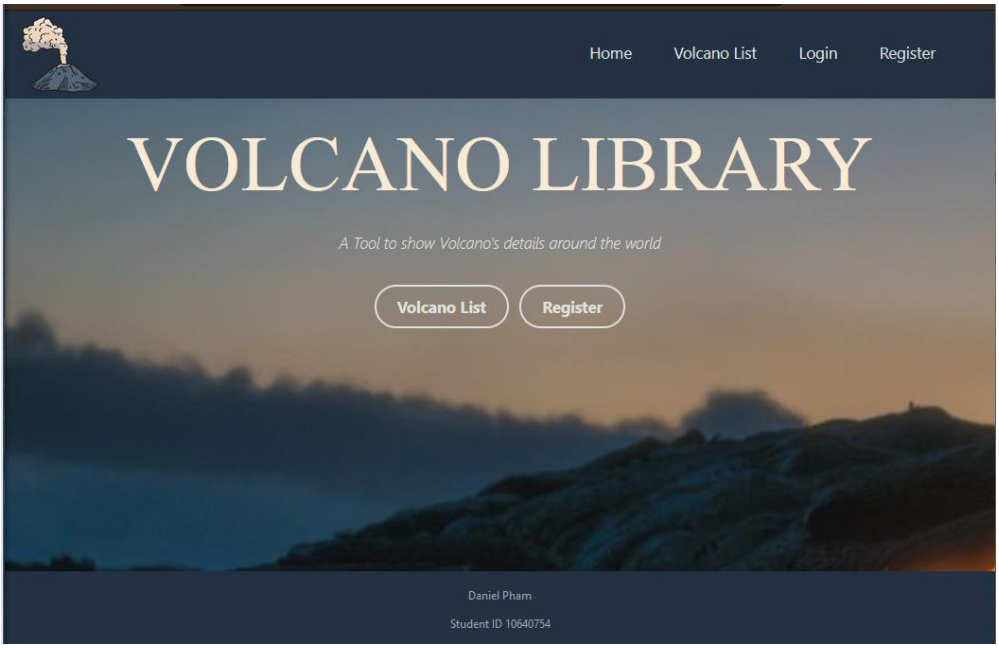
- Provide a more modern notification pop up
- More methods to announce what changing in the application
- Forgot password function for login
- Search volcano with keyword (without using ag-grid filter option) => only worked when user know specific name
- Upgrade selector components to search bar

User guide

1. Download the application to your local machine
2. Install all required modules (can be seen above **in modules used section**)
3. Open VSCODE and locate the path you stored the application then run it in the terminal

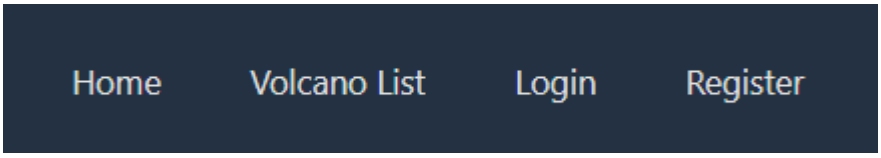


4. You now can see the homepage in http://localhost:3000/

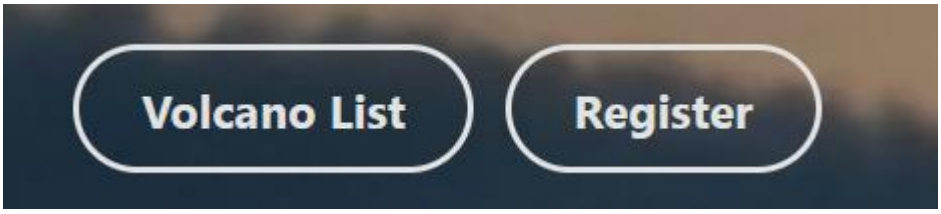


Noticed: If not connect to QUT network you need to use **Cisco**

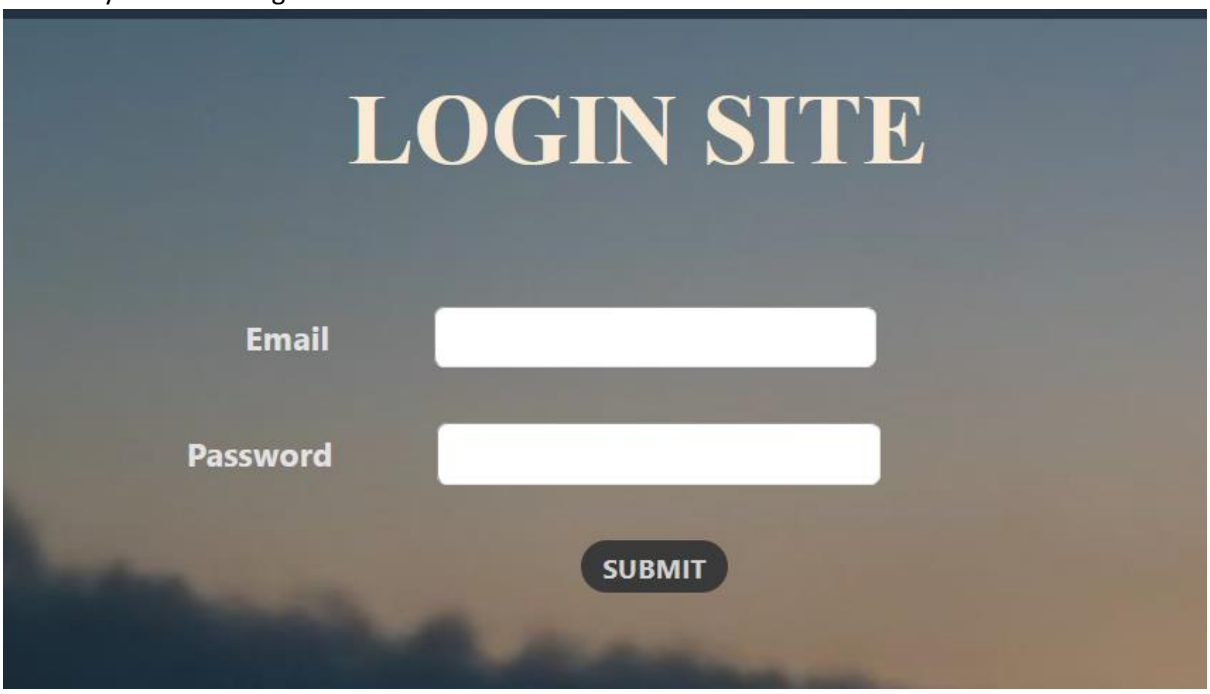
5. In this screen you can navigate to other page by using navigation bar on top



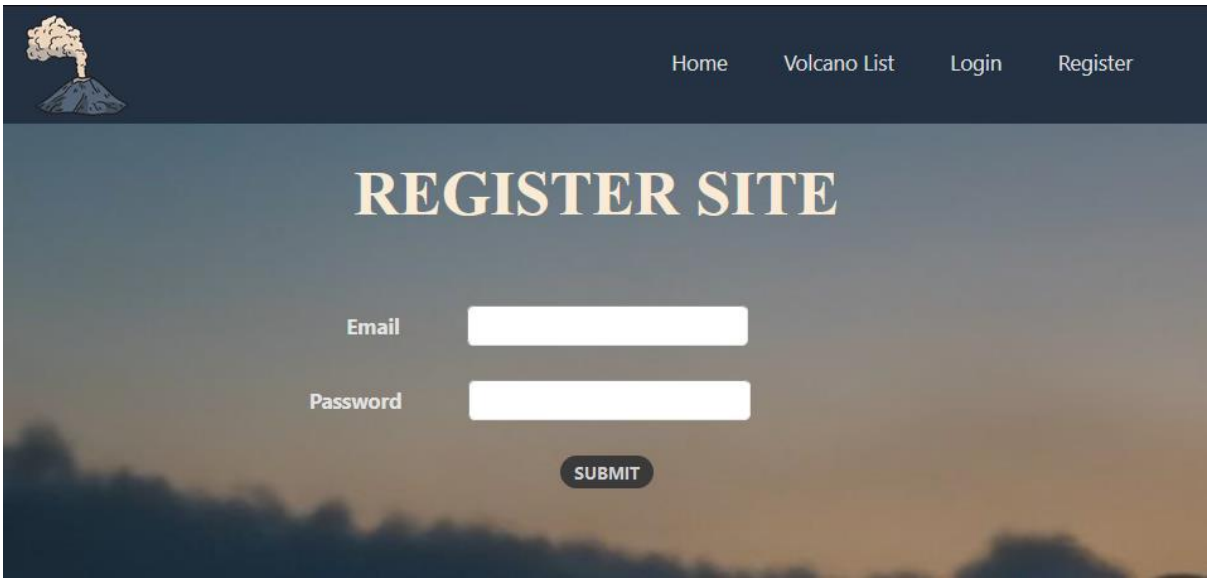
Or using these buttons



6. You can now choose volcano list to search for volcanoes and using grid table but to see cut content you have to login first:



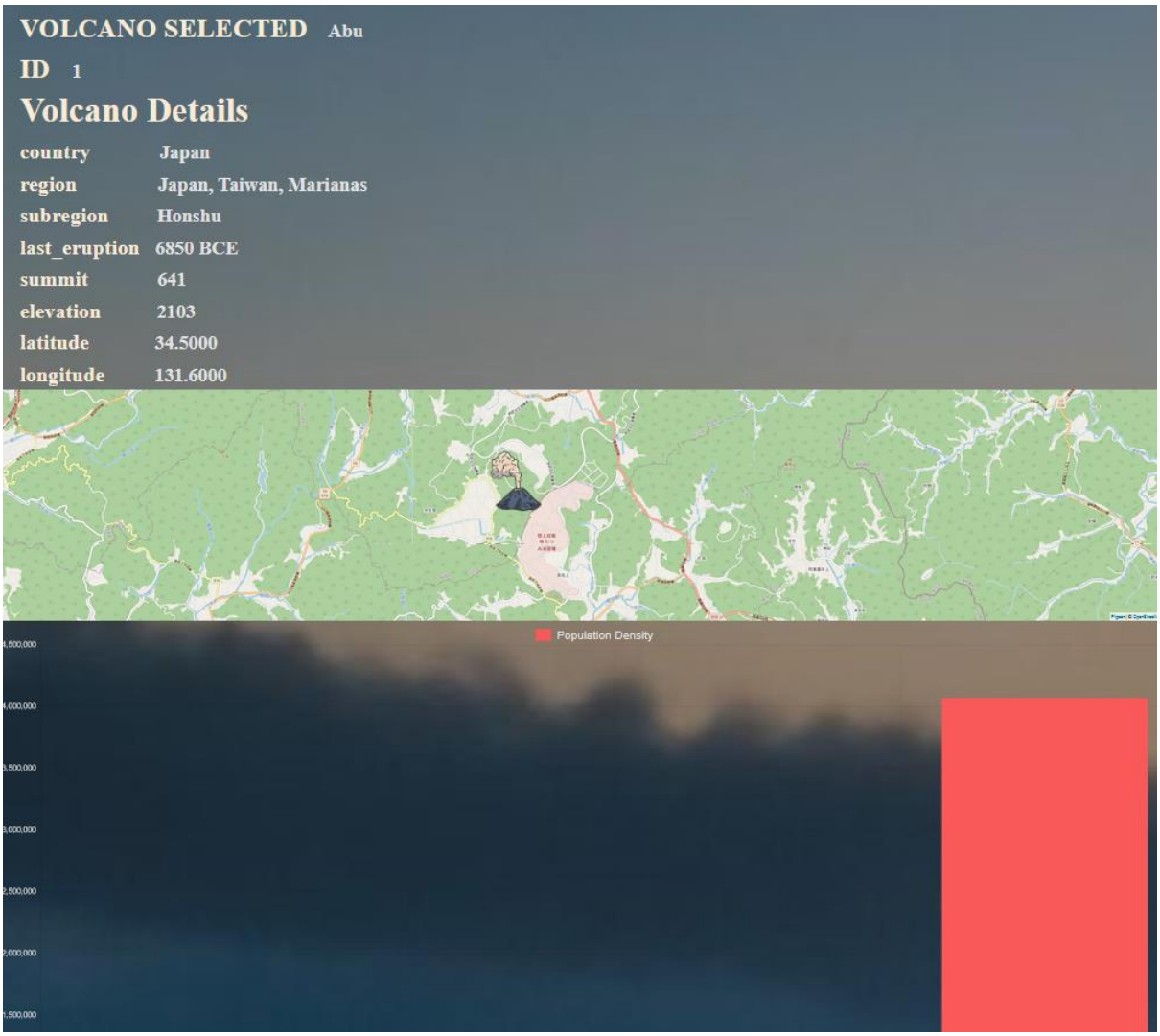
7. If you don't have an account just go straight to Register site on both Home page or navigation bar as shown above.



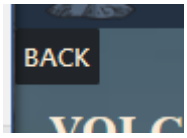
8. If successfully register an account, the screen will automatically refresh then show login site

9. When successfully login the web will refresh and switch the screen to Volcano List page

10. To see full details of a volcano just click-on the row data you want then the screen will refresh to that Volcano Page with full details



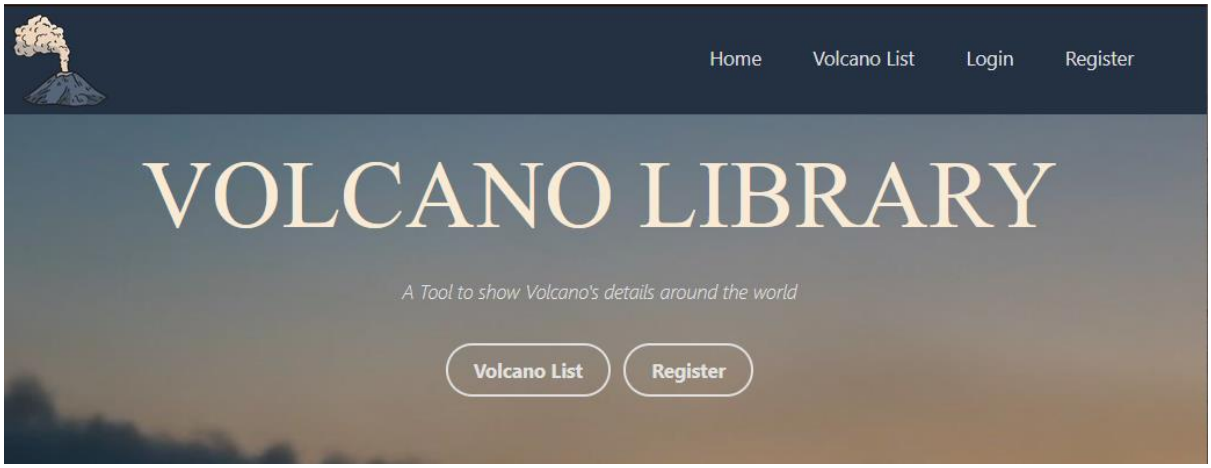
11. When you done with browsing the volcano page just simply click on Back button on top



The screen will then turn back to volcano list page

12. When you have finished with the volcano searching and want to logout, simply click on LOG OUT button on top in the navigation bar

The screen will refresh and now you turned back to homepage



References

Use a standard approach to referencing – see the guidance at https://www.citewrite.qut.edu.au/cite/.

Appendices as you require them