

2021

CAB230 Assignment 1 Client Side



CAB230

Happiness API – Client Side
Application

Hyeonwoo Lee (Leo)

n10661531

5/10/2021

Contents

Introduction	2
Purpose & description.....	2
Completeness and Limitations.....	2
Use of End Points	3
/rankings	3
/countries.....	3
/factor/{year}	4
/user/register	5
/user/login	6
Modules Used	6
Ag-grid-react	6
Module 2	Error! Bookmark not defined.
Module n	Error! Bookmark not defined.
Application Design	7
Navigation and Layout	7
Usability and Quality of Design	8
Accessibility	8
Technical Description	10
Architecture	11
Test plan.....	13
Difficulties / Exclusions / unresolved & persistent errors.....	14
Extensions (Optional).....	15
User guide	16
References	17
Appendices as you require them	17

Introduction

Purpose & description

The purpose of this app is about showing data to users which is about 'happiness survey'. This web application supposed to make client could analyze data in various ways. Such as user could get a data by year or country and so on. The 'happiness survey' data is from QUT's provided API.

When user goes into this app, they will head to main page, at this page there will be some visible buttons that could be provide to user choices. There is menu that are ranking, search, register and logins. And on the main home page, the feature called "Do you want to see extra Factor lists? than sign in! There is further information!" is mentioned so that users can expect and induce sign-up. Also, beside that comment there are login and register button to let user login or register immediately.



Completeness and Limitations

From my judgment, the basics are in place and work smoothly however, there were some trying make specification in some part. Firstly, the get error part in register, what user must insert while register, it is email and password. In this part if email form is "abc" it supposed to catch it is not valid form because email supposed to be "abc@example.com". It was one of limited part of this app. The second one is about designed because if the tap size get smaller some characters and buttons are overlapped so it is har to see and hard to process function.

Use of End Points

/rankings

The screenshot shows the /rankings page with a sidebar menu (Home, Ranking, Search, Register, Login) and a main content area. The main area has a 'Rankings' header with a dropdown menu set to '2019' and a sub-header '156 of informations'. Below this is a table with 4 columns: Rank, Country, Score Count, and Country. The table lists the top 5 countries for 2019: Finland (Rank 1, Score 7.769), Denmark (Rank 2, Score 7.600), Norway (Rank 3, Score 7.554), Iceland (Rank 4, Score 7.494), and Netherlands (Rank 5, Score 7.488).

The browser's network inspector is open, showing a GET request to `http://131.181.190.87:3000/rankings?year=2019` with a status of 200 OK. The response is a JSON array of 156 objects, each representing a country's ranking for the year 2019.

In Ranking page, it shows the GET method, as it shows, the data that requested went correctly into the API link.

/countries

The screenshot shows the /countries page with a sidebar menu (Home, Ranking, Search, Register, Login) and a main content area. The main area has a 'Search' header with a dropdown menu set to 'Australia' and a sub-header '6 of informations'. Below this is a table with 4 columns: Year, Rank, Score Count, and Country. The table lists the top 4 countries for the year 2017: Australia (Rank 12, Score 7.223), Australia (Rank 11, Score 7.228), Australia (Rank 10, Score 7.272), and Australia (Rank 10, Score 7.284).

The browser's network inspector is open, showing a GET request to `http://131.181.190.87:3000/rankings?country=Australia` with a status of 304 Not Modified. The response is a JSON array of 6 objects, each representing a country's ranking for the year 2017.

In search page, it shows the GET method, as it shows, the data that requested (by dropdown bar) went correctly into the API link.

Search

Country

Australia

6 of informations

Year	Rank	Score	Count	Country
2020	12	7.223		Australia
2019	11	7.228		Australia
2018	10	7.272		Australia
2017	10	7.284		Australia

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Meth...	Domain	File	Initiator	Type	Transferred	Size
200	GET	131.181.19...	rankings?country=Austr	main.chunk.js...	json	244 B (raced)	2 B
200	GET	131.181.19...	rankings?country=Austra	main.chunk.js...	json	244 B (raced)	2 B
200	GET	131.181.19...	rankings?country=Austral	main.chunk.js...	json	244 B (raced)	2 B
200	GET	131.181.19...	rankings?country=Australi	main.chunk.js...	json	244 B (raced)	2 B
304	GET	131.181.19...	rankings?country=Australia	main.chunk.js...	json	cached	372 B

30 requests 678.03 KB / 60.72 KB transferred Finish: 2.75 min DOMContentLoaded: 241 ms load: 294 ms

Filter Headers

GET http://131.181.190.87:3000/rankings?country=Australia

Status 304 Not Modified

Version HTTP/1.1

Transferred 560 B (372 B size)

Referrer Policy strict-origin-when-cross-origin

In search page, it shows the GET method, as it shows, the data that requested (by search) went correctly into the API link.

/factor/{year}

Factors

2018 limit

156 of informations

Rank	Country	Score	Economy
1	Finland	7.632	1.305
2	Norway	7.594	1.456
3	Denmark	7.555	1.351
4	Iceland	7.495	1.343
5	Switzerland	7.487	1.420

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter URLs

Status	Meth...	Domain	File	Initiator	Type	Transferred	Size
200	GET	localhost:3...	favicon.ico	FaviconLoader...	html	cached	1.59 ...
200	OPTI...	131.181.19...	2020	fetch	plain	320 B	0 B
304	GET	131.181.19...	2018	main.chunk.js...	json	cached	23.39...
304	OPTI...	131.181.19...	2018	fetch	plain	320 B	0 B
304	GET	131.181.19...	2018	main.chunk.js...	json	cached	23.84...

11 requests 552.42 KB / 769 B transferred Finish: 26.78 s DOMContentLoaded: 228 ms load: 277 ms

Filter Headers

GET http://131.181.190.87:3000/factors/2018

Status 304 Not Modified

Version HTTP/1.1

Transferred 24.02 KB (23.84 KB size)

Referrer Policy strict-origin-when-cross-origin

In this factor page, it shows the GET method, as it shows, the data that requested (by year dropdown bar) went correctly into the API link.

The screenshot shows the 'Factors' web application. On the left is a navigation menu with links: Home, Ranking, Search, Factors, and Logout. The main content area has a title 'Factors' and a dropdown menu set to '2018' with a '5' limit. Below this is a table with 5 rows of data:

Rank	Country	Score	Economy
1	Finland	7.632	1.305
2	Norway	7.594	1.456
3	Denmark	7.555	1.351
4	Iceland	7.495	1.343
5	Switzerland	7.487	1.420

Below the table, the Chrome DevTools Network tab is open, showing a list of requests. The selected request is a GET request to 'http://131.181.190.87:3000/factors/2018?limit=5' with a status of 304 Not Modified.

In this factor page, it shows the GET method, as it shows, the data that requested (by year dropdown bar and limit dropdown bar) went correctly into the API link.

/user/register

The screenshot shows the 'Register' web application. On the left is a navigation menu with links: Home, Ranking, Search, Register, and Login. The main content area has a title 'Register' and a form with fields for 'Email address' (containing 'abcde@example.com') and 'Password' (represented by dots). A 'Register' button is at the bottom of the form.

Below the form, the Chrome DevTools Network tab is open, showing a list of requests. The selected request is a POST request to 'http://131.181.190.87:3000/user/register' with a status of 201 Created. The 'Request' tab shows the JSON body:

```

{
  "email": "abcde@example.com",
  "password": "password"
}

```

The 'Response' tab shows the JSON body:

```

{
  "message": "User created"
}

```

In this Register page, it shows the POST method, and it gets user data into local storage and send to API than get response of error check. In image above user data went correctly into the API link.

[/user/login](#)

The screenshot shows a web browser with a login page. The page has a teal background and a dark sidebar on the left with links: Home, Ranking, Search, Register, and Login. The main content area is titled "Login" and contains two input fields: "Email address:" with the value "abcde@example.com" and "Password:" with masked characters. A "Login" button is below the fields.

Below the page, the browser's developer tools are open, showing the Network tab. A list of requests is displayed, with the last one being a POST request to "131.181.190.87:3000/login". The request details are expanded, showing the Headers tab. The headers include "Host: 131.181.190.87:3000", "Filename: /user/login", and "Address: 131.181.190.87:3000". The Status is "200 OK". The Response tab shows a JSON object with the following data:

```
JSON
{
  "email": "abcde@example.com",
  "password": "password",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbWFnZWVudC6iOiJmY2RlQGV4YW1wbGUuY290tiwiZXhwIjoxNjIwNjYxNTYxLCJpYXQiOiJlODUxNjF9.5P-R4BonVDo_UR--q-mln1oeAdliuITJaArmtYbSyA",
  "token_type": "Bearer",
  "expires_in": 86400
}
```

In this Login page, it shows the POST method, and it check user from data from local storage by token and get response of error check. The user data went correctly into the API link.

Modules Used

This is just a list of the external modules that you have used. You need not specify core React modules. In each case, ***we want the name, a brief description, and a link to the docs at npm or github or wherever.*** The example is `ag-grid-react` as most people will be using this. Just copy that style and add more as necessary.

[Ag-grid-react](#)

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

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

[reactstrap](#)

Module quickly build forms. Especially to use on Badge function.

<https://reactstrap.github.io/>

[ag-grid-community/dist/styles/ag-grid](#)

Module from AG Grid Community to build valuable data lines.

<https://www.ag-grid.com/javascript-grid/modules/>

ag-grid-community/dist/styles/ag-theme-alpine-dark.css

Module from AG Grid Community to change data lines colour.

<https://www.ag-grid.com/javascript-grid/themes-provided/>

react-chartjs-2

The npm packages to create chart in React js.

<https://www.npmjs.com/package/react-chartjs-2>

react-router

Module to use router function in react.

<https://reactrouter.com/>

react-dom

The package that can be used in react about DOM specific methods.

<https://reactjs.org/docs/react-dom.html>

Application Design

Navigation and Layout

The process of design could be divided by two big part, the first part is login and logout statement part, second part is the data showing part. Firstly, I made a homepage, there are navigation var and home page contents, in here user could do login through login option, if logged in, login button and register button will be disappear and factors button and logout button will come out which is the only function that allows to log in users so, users could go to factors or logout pages.

In data showing part, there are some cases which are ranking page, search page and factor page. In ranking page data showed but it could be sorted by years and in search page, data could be shown through searching by specific countries. At last, in factors page, data will be shown by years and limit's dropdown var selection.

The mock designed.

First menu nav var shows the register and login button before logged in. And after logged in, navigation var will become to factors button and logout button.



If user do register, page will go immediately into login page.

Register

Email address:
ccc@gmail.com

Password:
...

Register

=====>>

Login

Email address:

Password:

Login

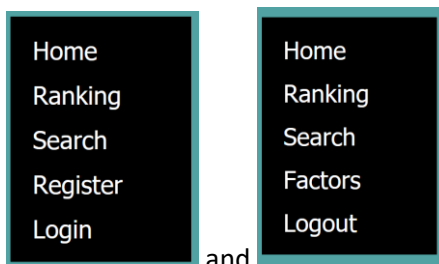
Usability and Quality of Design

- Is your display well organized? Is the layout clean or clumsy?
My display is well organized because I choose to use nav by a vertical version which is for organize well. However, when screen size gets smaller than expected size, designs are overlapped and crashes. So, it does not look well organized.



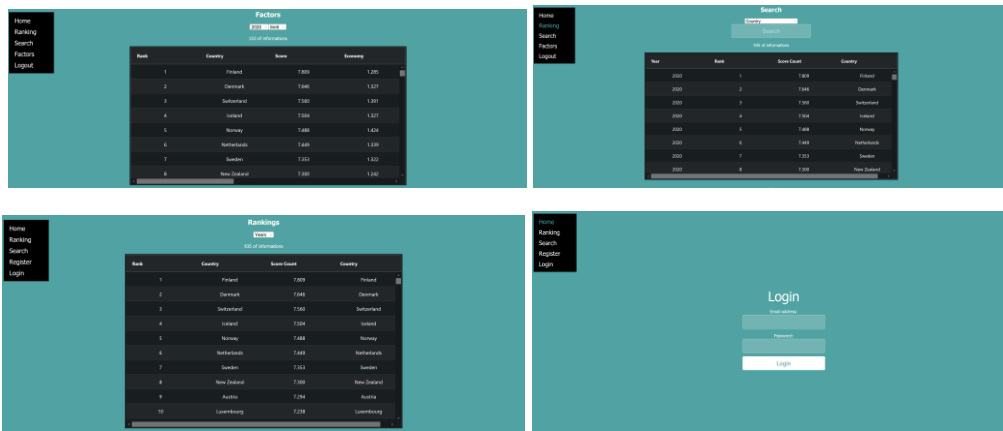
<< == if it website's page become smaller, than overlapped like this.

- Is the navigation clear and intuitive?
Yep, because as picture downside, the main options are all for link to necessary functions. menus are considered by user's logged in statement, so if user does not log in it does not show invalid menu buttons like factors and logout.

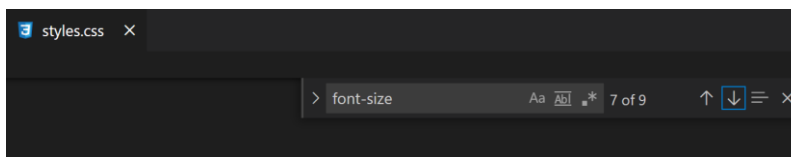


- Is the application consistent with user expectations from other apps?
Basic parts are generally similar to other apps; however, the weakness part of this app is the specifications the amount of specification details are lacked in our app. Such like find ID password function, video function sound function and so on.

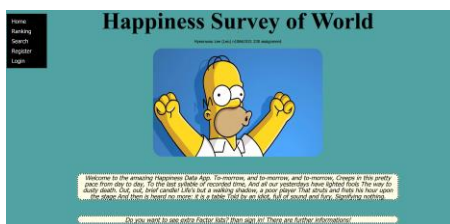
- Is the visual design consistent across screens? Are there too many colors?
This app's pages are tried to design by same concept which is like same background color same way to feature data.



- Do you use a small number of fonts and font sizes?
In this app only two fonts were used "Merriweather", serif and "Source Sans Pro", sans-serif. However, to fit handle data case by case, several font sizes were used.



- Comment on the usability of your design – are there compromises that make it awkward to use? How might you improve those?
As this application is for data analysis, people may feel it is difficult. By putting an animated character image on the main screen that looks happy, it makes a connection with happiness, and at the same time, it allows users to feel less difficult, use comfortably, and increase accessibility.



Accessibility

- Provide a text equivalent for every non-text element – alternatives to images, symbols, scripts, graphical buttons, sounds, audio and video files and so on.
YES
- Ensure that all information conveyed with color is also available without color, for example from context or markup.
YES
- 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.
NO
- Ensure that text equivalents are updated when dynamic content changes.
YES
- Avoid causing the screen to flicker.
YES
- Use the clearest and simplest language appropriate for a site's content.
YES
- For tables, identify row and column headers – clearly differentiated from the data.
YES

From question “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.” It is not available in this app. Because style sheet and document must be connected with one another. It will interrupt to do code if I use this on app.

The part that helps to design is especially the question “Ensure that all information conveyed with color is also available without color, for example from context or markup.”. It can be used without color like markup, design process became simple and efficient.

The question “Avoid causing the screen to flicker.” Is one of my problem by miss code, how ever I fixed this by using the comment history. push () function which is from react-router module.

Technical Description

Architecture

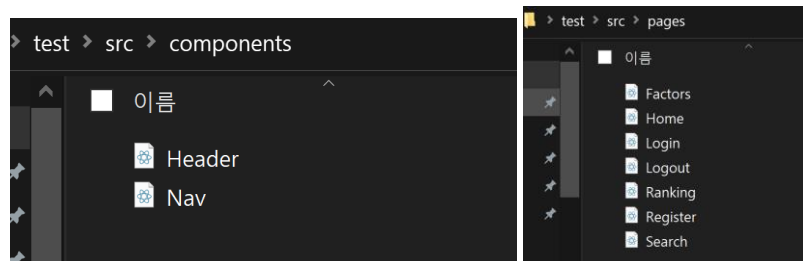
```
import Home from "../pages/Home";
import Ranking from "../pages/Ranking";
import Search from "../pages/Search";
import Factors from "../pages/Factors";
import Register from "../pages/Register";
import Login from "../pages/Login";
import Logout from "../pages/Logout";

export default function App() {
  return (
    <Router>
      <div className="App">
        <Header />
        <Switch>
          <Route exact path="/">
            <Home />
          </Route>
          <Route path="/Ranking">
            <Ranking />
          </Route>
          <Route path="/Search">
            <Search />
          </Route>
          <Route path="/Factors">
            <Factors />
          </Route>
          <Route path="/Register">
            <Register />
          </Route>
          <Route path="/Login">
            <Login />
          </Route>
          <Route path="/Logout">
            <Logout />
          </Route>
        </Switch>
      </div>
    </Router>
  );
}
```

As this left picture shows, this web is constructed like this. There are several imports to show a different page. Which means there are extra js react pages.

node_modules	2021-05-07 오후 6:42	파일 폴더	
public	2021-04-28 오후 5:04	파일 폴더	
src	2021-05-10 오전 10:06	파일 폴더	
package	2021-05-07 오후 6:42	JSON 원본 파일	1KB
package-lock	2021-05-07 오후 6:42	JSON 원본 파일	683KB

components	2021-05-10 오전 10:13	파일 폴더	
pages	2021-05-10 오전 10:33	파일 폴더	
App	2021-05-09 오후 3:25	JavaScript 원본 ...	2KB
index	2021-04-28 오후 4:50	JavaScript 원본 ...	1KB
styles	2021-05-10 오전 10:12	CSS 원본 파일	3KB



If goes into src folder there are some piles firstly, there is components folder, inside that, there is a folder which are navigation var. Secondly, there is a folder which name is pages. These are as mentioned above, files for each individual pages. And there is App file. This file is main of this application. Every information gathered in this file and processed. And the last one is styles file which is for website designing CSS file.

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Compiled successfully!

You can now view react in the browser.

Local:      http://localhost:3001
On Your Network:  http://172.19.12.251:3001

Note that the development build is not optimized.
To create a production build, use npm run build.
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ireun\Desktop\test> npm install
```

Application could be handled in terminal through code like npm install and npm start than it could be available to open application.

```

3 export default function Logout() {
4   function loggoutt() {
5
6     localStorage.removeItem('token')
7     window.location.href = "/";
8   }
9 }

```

Login service: storage token of user information and authorize from API.

Register service: storage user information to local storage for login.

Logout service: Delete data (token, user information) from local storage.

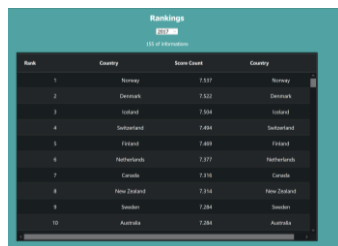
Search: Show data with sort by country search engine and country dropdown bar.

Rank: Show data with Year dropdown bar selection

Factors: Show data with Year dropdown bar selection and limit dropdown bar selection.

Test plan

1.

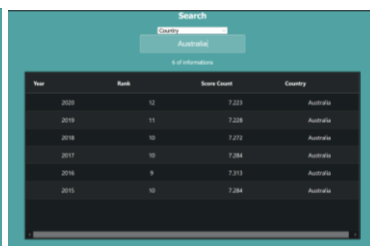


Rankings

10 of 10 countries

Rank	Country	Score Count	Country
1	Norway	7,537	Norway
2	Denmark	7,522	Denmark
3	Iceland	7,504	Iceland
4	Switzerland	7,484	Switzerland
5	Finland	7,489	Finland
6	Netherlands	7,377	Netherlands
7	Canada	7,319	Canada
8	New Zealand	7,314	New Zealand
9	Sweden	7,284	Sweden
10	Australia	7,284	Australia

2.



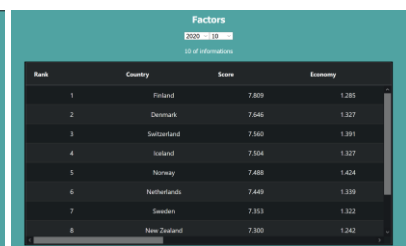
Search

Country: Australia

10 of 10 countries

Year	Rank	Score Count	Country
2020	12	7,223	Australia
2019	11	7,228	Australia
2018	10	7,272	Australia
2017	10	7,284	Australia
2016	9	7,313	Australia
2015	10	7,284	Australia

3.



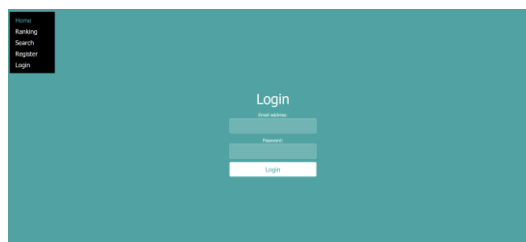
Factors

2020 10

10 of 10 countries

Rank	Country	Score	Economy
1	Finland	7,819	1,285
2	Denmark	7,546	1,327
3	Switzerland	7,560	1,381
4	Iceland	7,504	1,327
5	Norway	7,489	1,424
6	Netherlands	7,449	1,339
7	Sweden	7,313	1,322
8	New Zealand	7,300	1,242

4.



Home
Ranking
Search
Register
Login

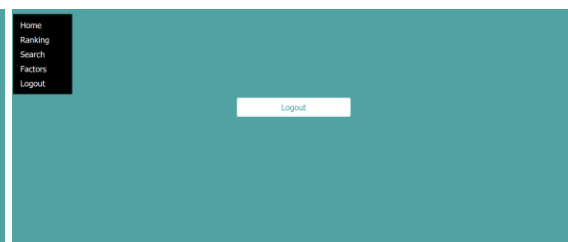
Login

Email address

Password

Login

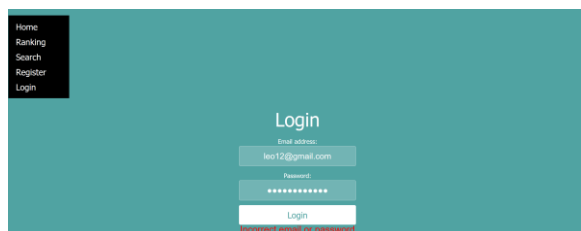
5.



Home
Ranking
Search
Factors
Logout

Logout

6.



Home
Ranking
Search
Register
Login

Login

Email address

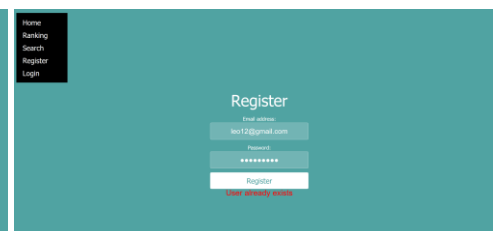
leo12@gmail.com

Password

Login

Invalid email or password

7.



Home
Ranking
Search
Register
Login

Register

Email address

leo12@gmail.com

Password

Register

Your already exists

8.



Cases	/Task	/expected outcome	/result	/screenshot.
Positive cases	/Get year value.	/Got year value and show correct data table	/ Pass	/ 1.
Positive cases	/Get country value.	/Got country value and show correct data table	/Pass	/ 2.
Positive cases	/Get year and limit value.	/Got year and limit value and show correct data table	/Pass	/3.
Positive cases	/Click Login	/User created and goes through homepage	/Pass	/4.
Positive cases	/Click Logout	/delete the token that saved in local storage and goes homepage	/Pass	/5.
Negative cases	/Login error	/show error response if login failed	/Pass	/ 6.
Negative cases	/Register error	/show error response if register failed	/Pass	/7.
edge cases	/show graph	/show graph clearly and specifically	/False	/8.

Difficulties / Exclusions / unresolved & persistent errors /

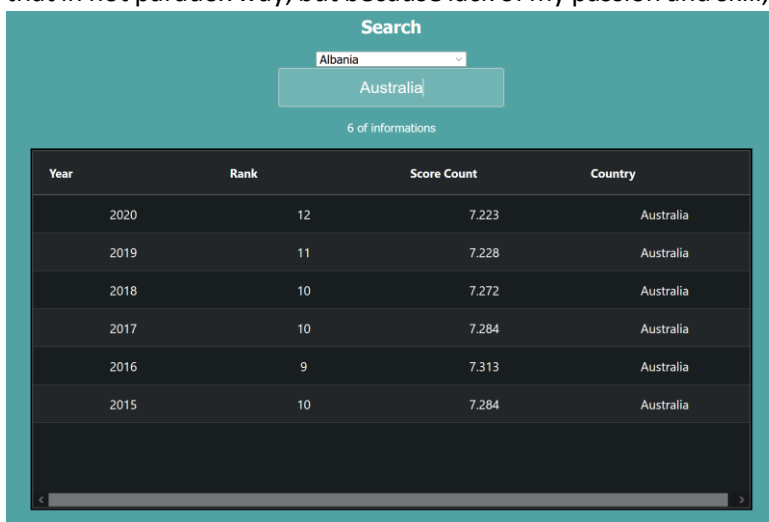
- What were your major roadblocks / how did you resolve them?

The major roadblock of this assignment is catching a null part especially in Factor function.

If webpage loaded, the year input will be null until I select from dropdown bar. And Limit dropdown bar as well. For API, Year is necessary value to get data. So, I was pretty struggling to handle this part. I put if statement and handled this like if (year == "") {year = "2020"}.

- Any functionality you didn't or couldn't finish and the technical issues encountered.

I think it is not that big error, however in search function, if I Search by dropdown bar and after that I search country by search comment input, the data are perfectly showing out. However, the dropdown var's character is still on that drop down bar. I was trying to solve that in not paradox way, but because lack of my passion and skill, I failed to solve this problem.

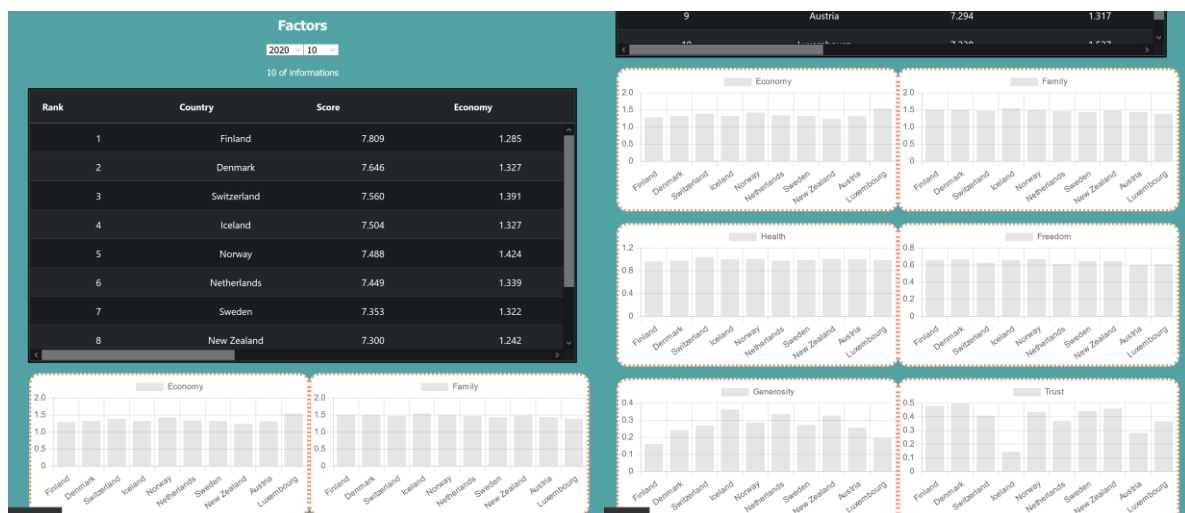
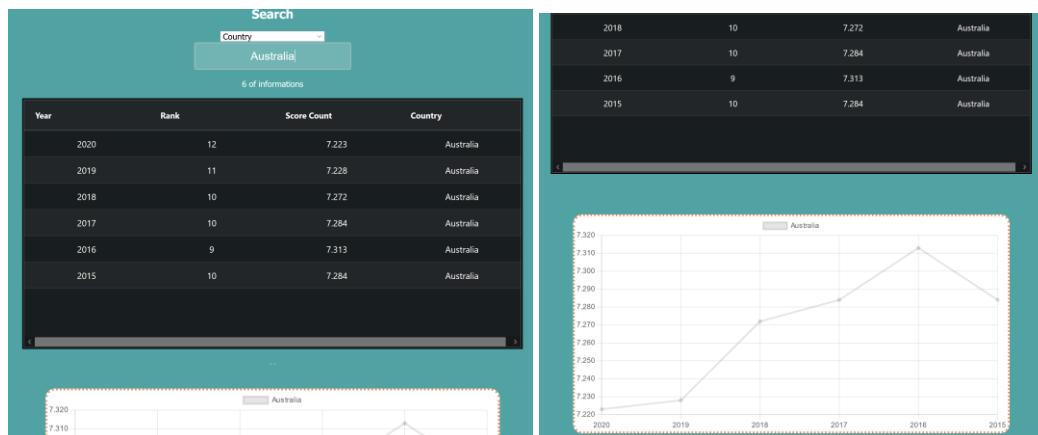


Year	Rank	Score Count	Country
2020	12	7.223	Australia
2019	11	7.228	Australia
2018	10	7.272	Australia
2017	10	7.284	Australia
2016	9	7.313	Australia
2015	10	7.284	Australia

- Are there any outstanding bugs?
There is no outstanding bug.

Extensions (Optional)

Where could you take this:



Firstly, as image above, I made graphs that reflects the current value on table to graph. There is one line graph that shows Country's information (year and ranking) on the search page. In the factors part, I made 6 bar graphs that interact with every table values.

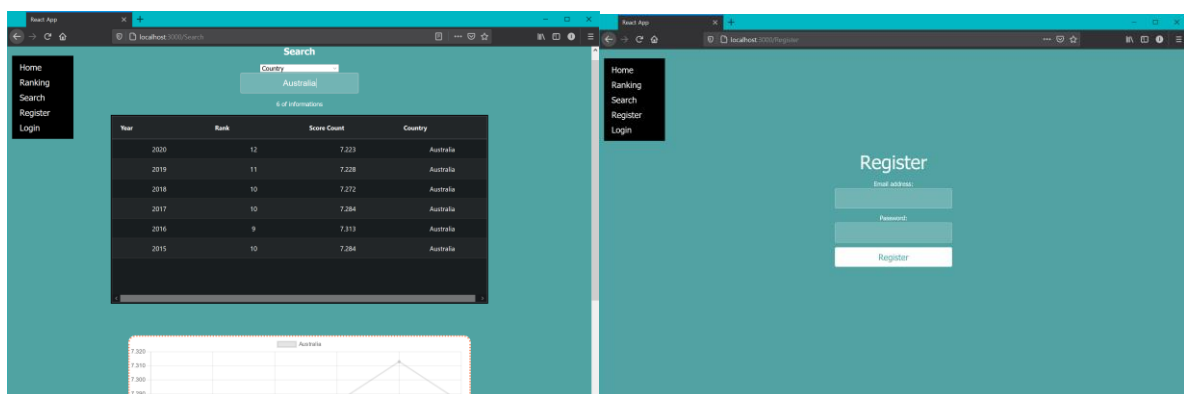
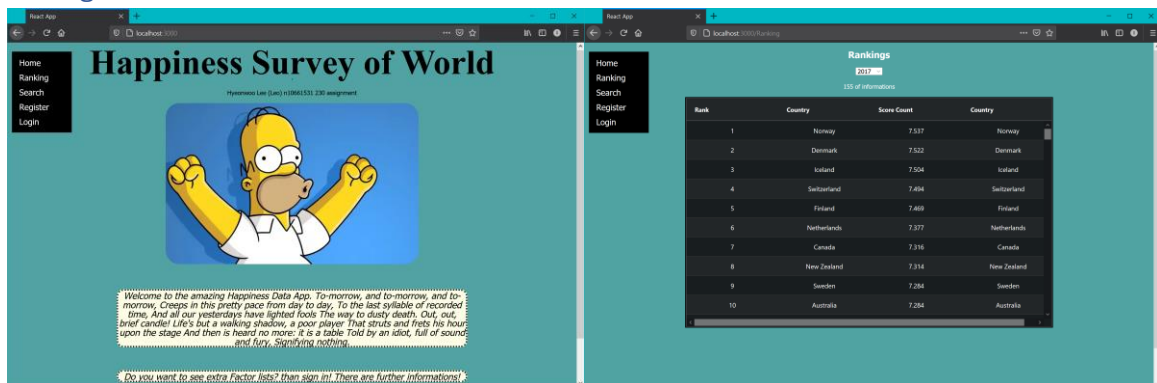
(FamilyData, EconomyData, HealthData, FreedomData, GenerosityData and TrustData)

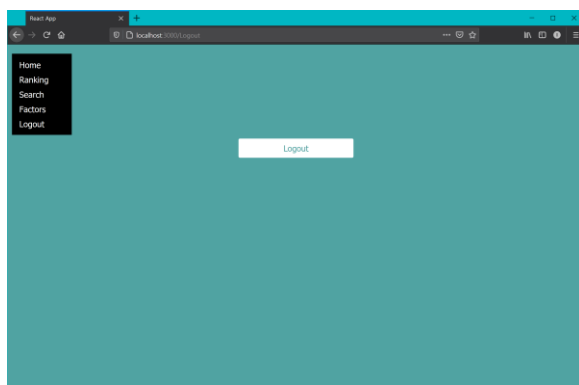
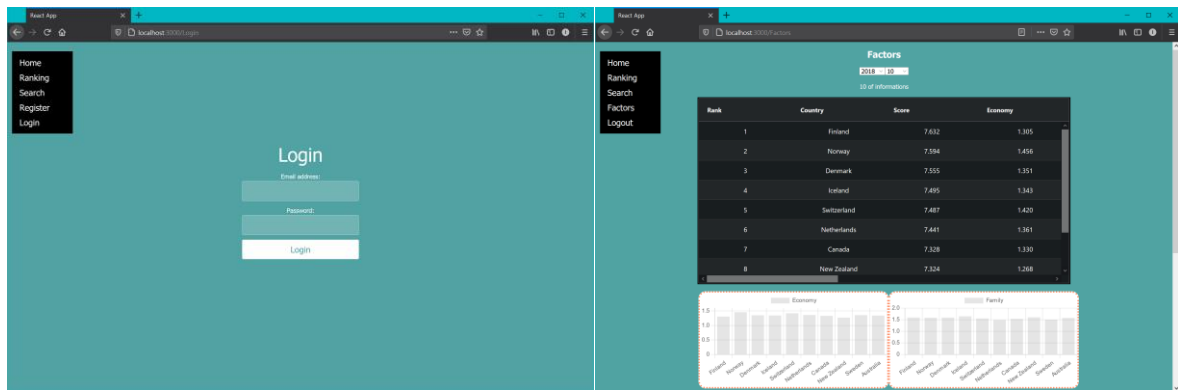
< - - - - > CountryData



Secondly, In homepage there is a button that ask to user to login and register, and if user logged in, button changes into Factors button and Logout button. I used same lgic as

User guide





For user guide, this app is looking for a user who is looking for a Happiness Survey of World. In this website, it provides several functions such like Ranking, search and Factors. But If user wants to access into factor data, user has to login into this app, by this app's login and page. If user does not have an account, user could create account in register page. User also could go to logout after they logged in. In factor page user could see and analyze the data with graphs sorting with years and limits. In ranking page it shows data and it could be short by years. In search page, user could search data by put in country name.

References

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

CSS Tutorial. (n.d.). W3schools.com. Retrieved from <https://www.w3schools.com/css/>.

HTML Tutorial. (n.d.). W3schools.com. Retrieved from <https://www.w3schools.com/html/>.

JavaScript Tutorial. (n.d.). Retrieved from <https://www.w3schools.com/js/>.

Appendices as you require them