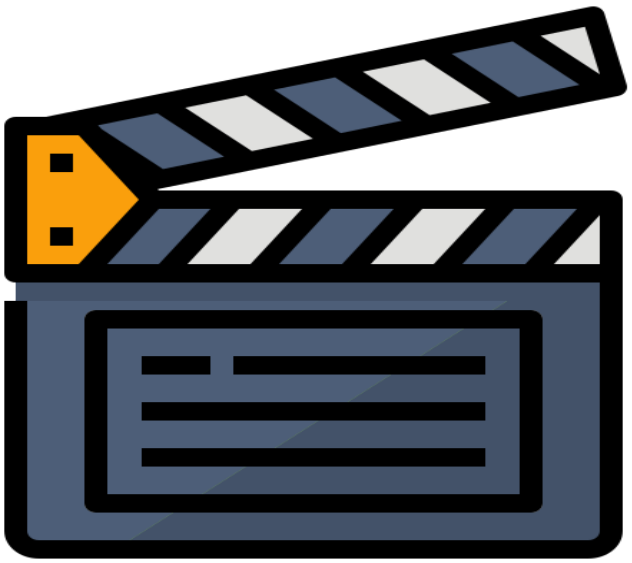
2023

CAB230 Assignment 2 Client Side



CAB230

Movies API – Client Side Application

Yeeun Lee(Evelyn)

N11355191

5/12/2023

Contents

Introduction

Purpose & description

It involved the development of a react-based web application that allows users to search and analyze movies. The application has five main pages: Home, Movies, Details, Registar and Login. Routers are used in the navigation bar to move between these pages.

On the Movie Search page, users can use the search bar or drop-down menu to filter movie titles. The Aggrid package filters and displays the desired movie and basic information about it. The Movie Details page contains details about the selected movie, including release date, cast and director, posters and summaries.

The line chart on the last page allows authorized users to view movies for specific members and other cast members(director, editor, etc.). The JWT is used for authentication during the login process.

To make the search process more efficient, the infinite grid package allows users to search 12,184 movies on the server. The reactstrap package provides a corresponding component to the bootstrap to ensure application responsiveness. Tables displaying movie information are rendered using the Ag-Grid React package. Overall, the app offers users a comprehensive search and analysis tool for movies, along with a variety of features that enhance the user experience.

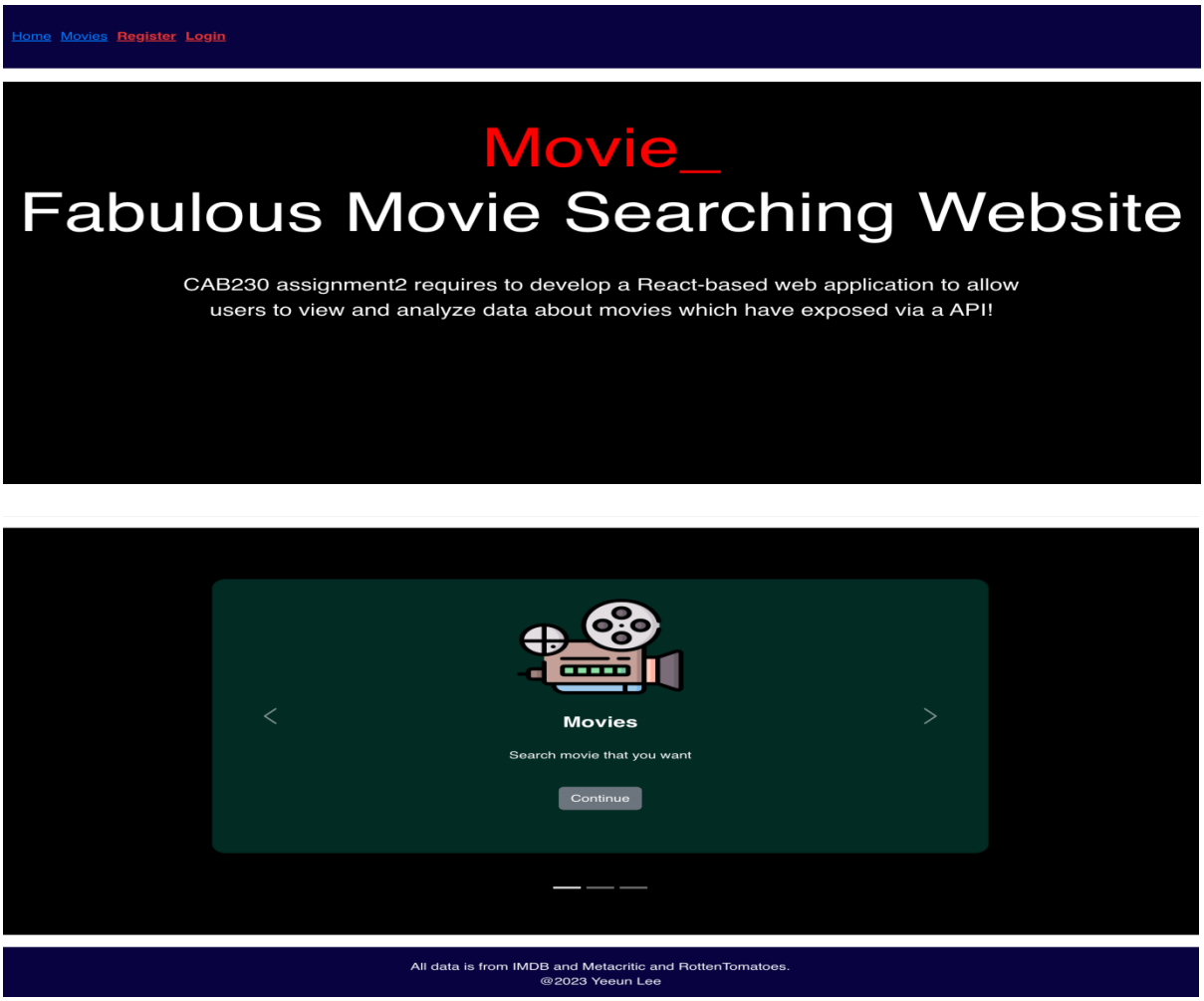


Figure 1 - App Homepage

Completeness and Limitations

I have completed my project at least at level 6, potentially level 7. Based on the first sentence of the criteria for level 6, I have met all the requirements for level 5 by including all the basic elements that allow users to explore and analyze movies. Navigation is handled through the corresponding router, and my website uses headers to navigate clearly, with visual and text-based prompts that guide the user through the experience. In terms of design, we focused on using a uniform color scheme and positioning to ensure that users can comfortably view the screen.

I believe this web application can support the push towards grade 7, if not grade 6, although there may be some inexperience in certain areas of 'Refresh' operation. However, by using the title of the movie, year, page parameter in the infinite Ag-grid, users can request values from the server through the URL and filter them.

Use of End Points

/movies/search

This page handles /movies/search endpoint, fetches 12184 movies from the server, not just the existing 100, and filters movies through two searchbars.

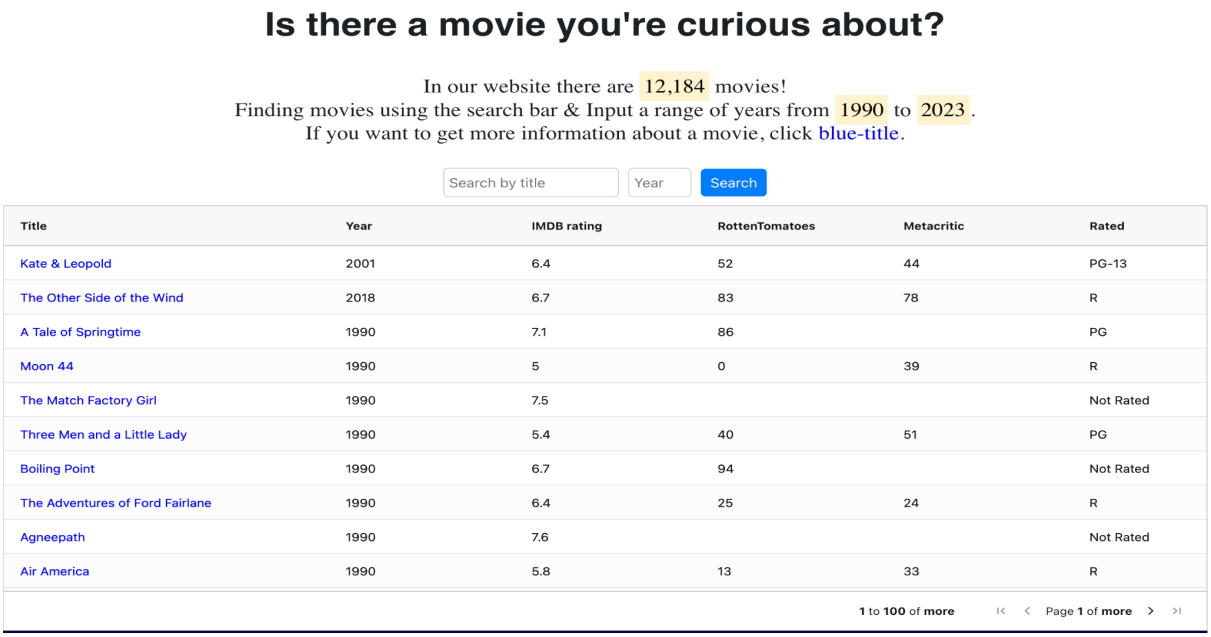


Figure 2-1 - Movies Page

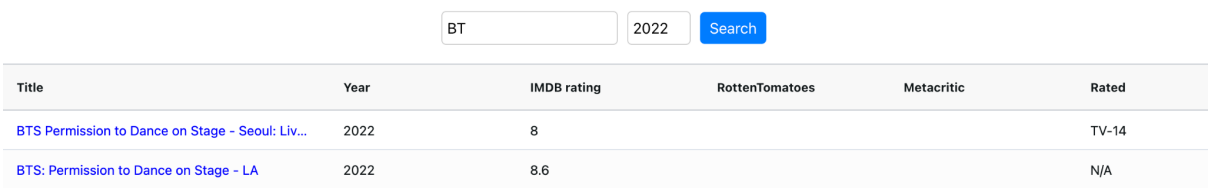


Figure 2-2 - Search (server request)

/movies/data/{imdbID}

Figure 3 below shows that the Movie Detail page handles the /movies/data/{imdbID} endpoint. This page displays information and ratings for a selected movie, as well as tables showing the people involved in making and starring in the movie. The information displayed on the page is filtered using the imdbID of the particular movie selected on the previous page.

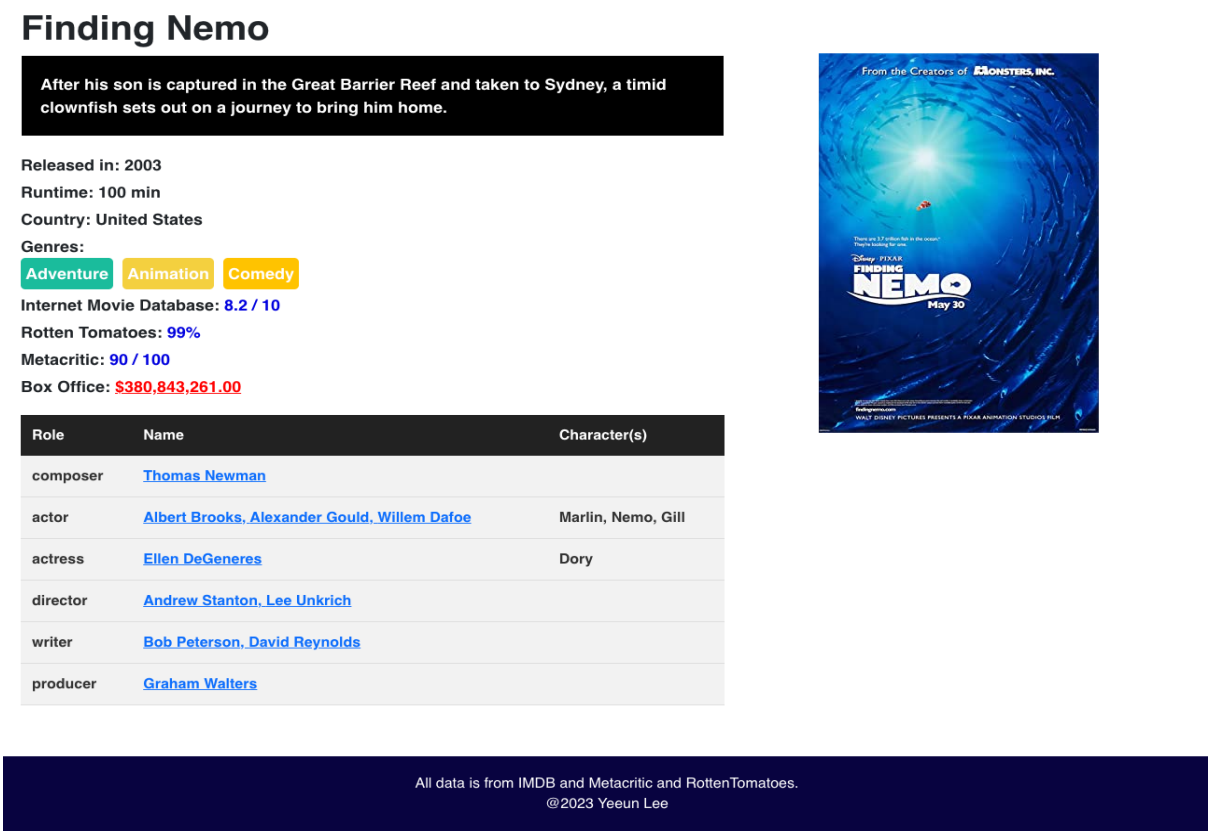


Figure 3 - Movie Detail Page

/people/{id}

Figure 4 shows that the last page includes /people/{id} endpoints to provide more detailed information about the people involved in making a particular movie. This allows users to obtain information about individuals such as actors, writers, directors, and more, using their respective IMDB IDs

The information is presented in two ways through the use of a button that toggles between Ag Grid and line chart views. If there is a large amount of data in Ag Grid, the line chart may provide a more readable and user-friendly visualization option. (Figure 4-1 & 4-2)

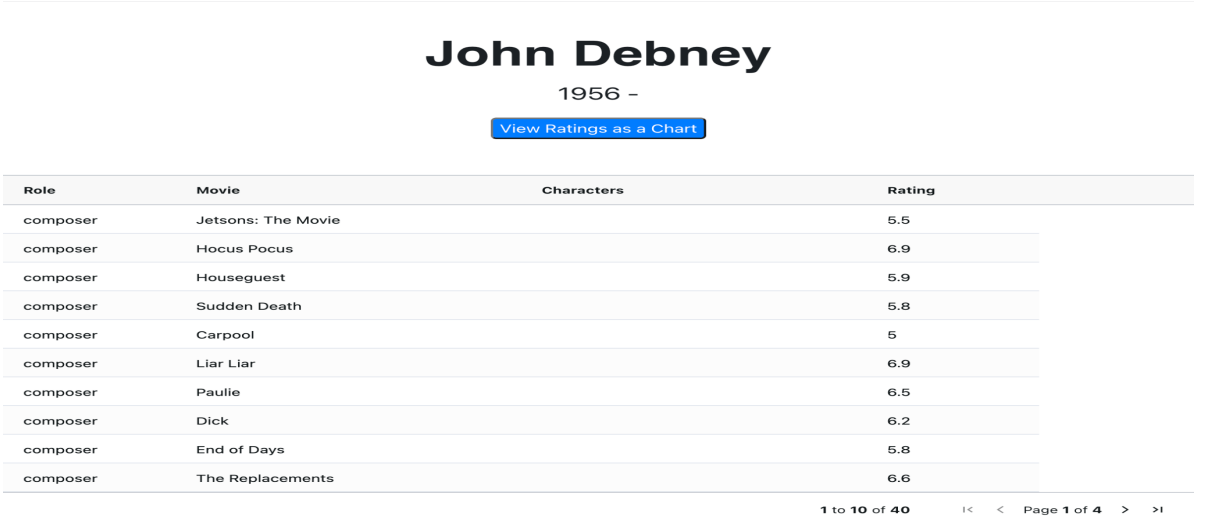


Figure 4-1 - Movie People Page (Ag-grid)

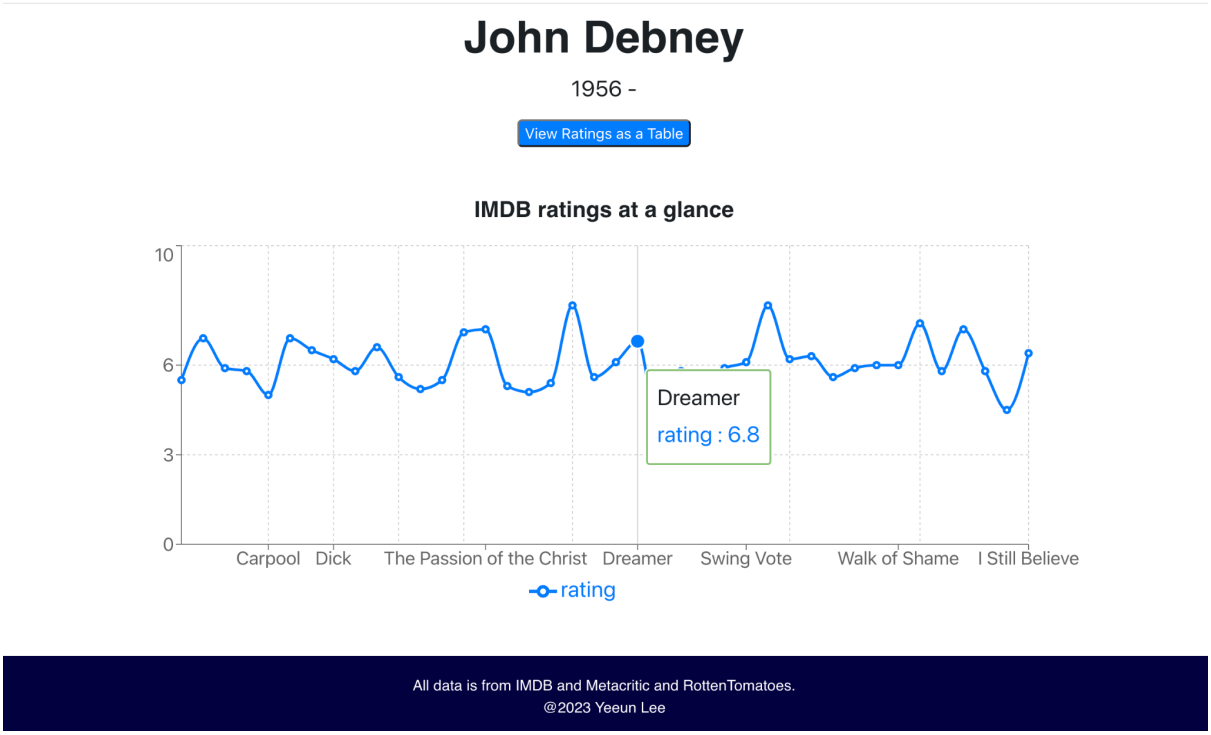


Figure 4-2 - Movie People Page (line chart)

/user/register

The Register page handles /user/register endpoints. Use this page to register emails that do not currently exist. In addition, it has the ability to check your password again.

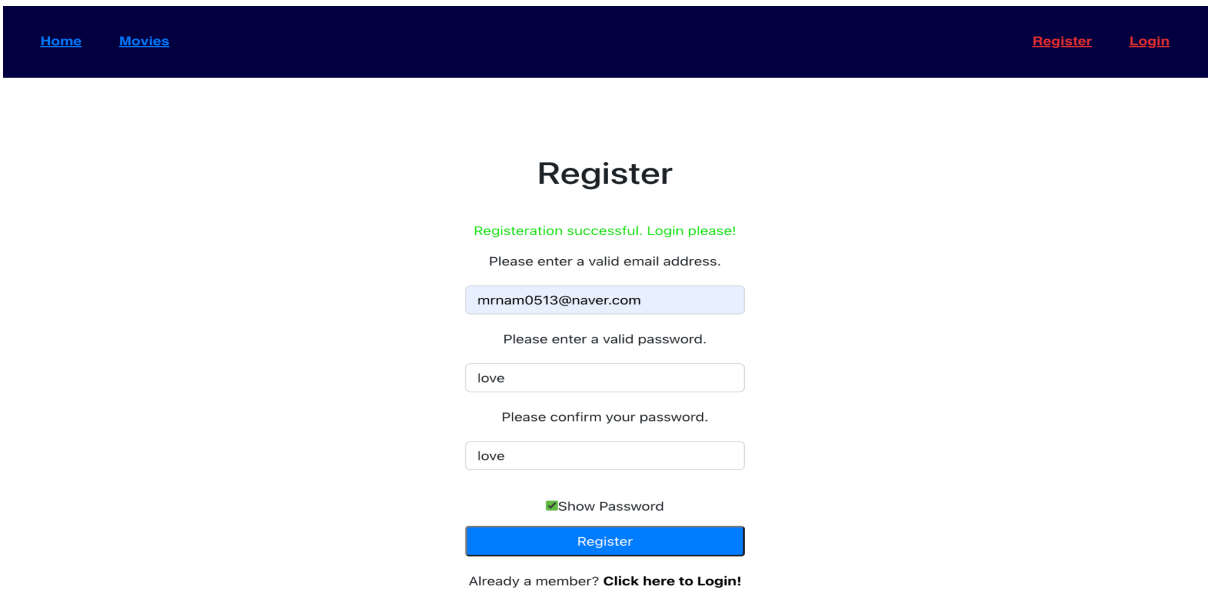


Figure 5 - Register Page (successful register)

/user/login

The login page handles the /user/login endpoints. Use the registered email and password.

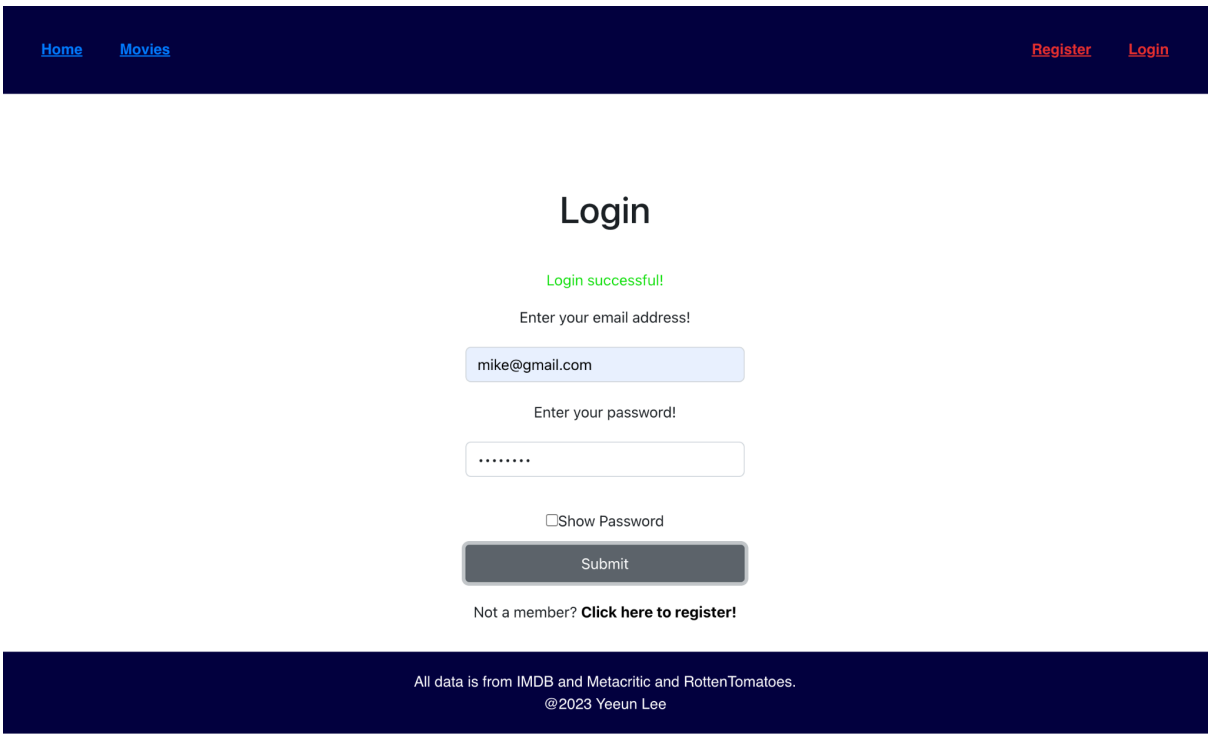


Figure 6 - Login Page

/user/refresh

Like the registration, login page, used POST, and handles the /user/refresh end point. 1 minute before the token expires, alert indicates to refresh the token.

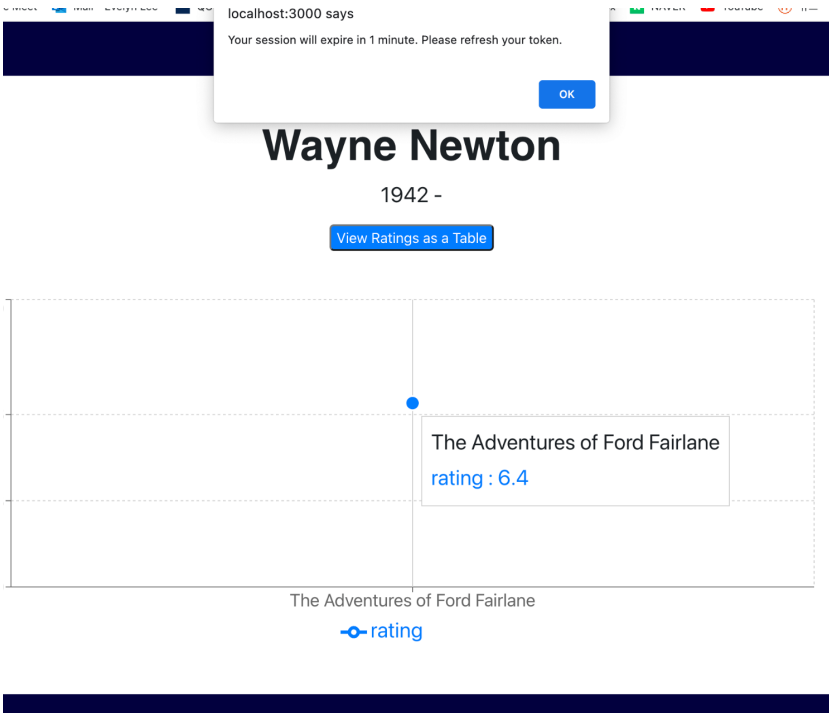


Figure 7 - Alert, before the token is expired(9min)

/user/logout

The logout is also indicated by POST that the endpoint should handle /user/logout. But in the web I created, a button with the ability to remove tokens, which is the logic required to log out, so that when a user login and gets a token, the logout button is visible.

Figure 8 - Logout (When user login)

Modules Used

Ag-grid-react

Module to provide fully-featured table components, including infinite scrolling.
https://www.ag-grid.com/react-grid/

Ag-gid-community/all-modules

Module with themes for Ag grid.
https://www.npmjs.com/package/ag-grid-community

Reactstrap

Module was instrumental in enhancing the functionality and user experience of the web application, providing efficient and user-friendly tools for building and presenting information to users.
https://reactstrap.github.io/?path=/story/home-installation--page

Bootstrap

Bootstrap offers a collection of pre-designed and reusable components, such as buttons, forms, and navigation, along with a responsive grid system and CSS styles, that can speed up and simplify the process of building and styling modern, mobile-first web applications.
https://getbootstrap.com/

JWT-decode

For decoding JWT on client side for finding expiry dates.
https://www.npmjs.com/package/jwt-decode

Recharts

React components to build line chart quickly.
https://recharts.org/en-US

Application Design

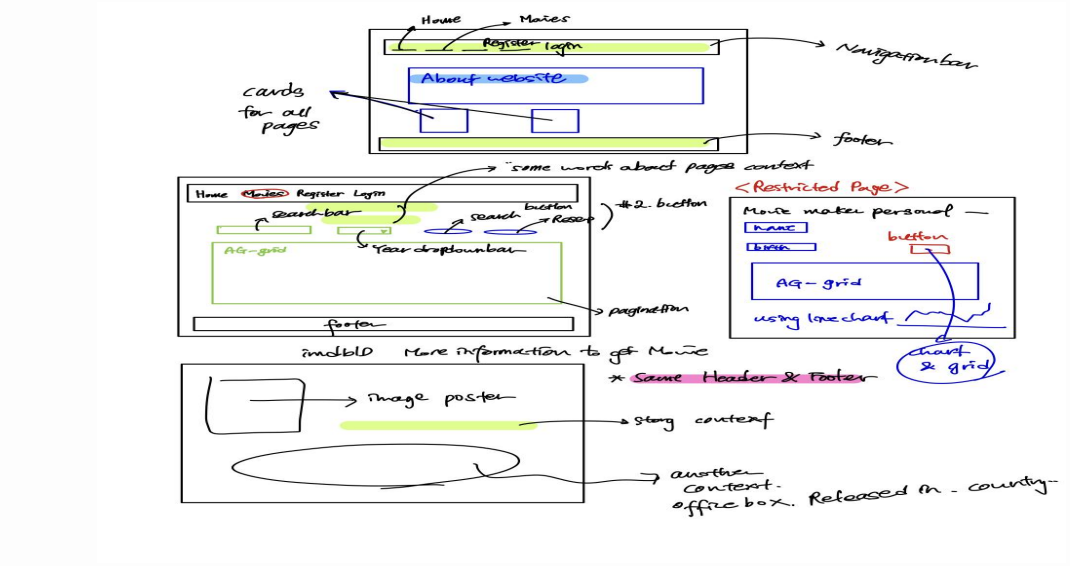
Navigation and Layout

The entire website was briefly sketched, divided into several detailed models. At first, my goal was to post a movie poster slide on the homepage. However, the given API was unable to post the poster on the home screen, so we used card components to organize the slides. This slide provided a brief description of the information on the web page.

There are a total of four items on navigation-bars, Home, Movies, Register, and Login, which provide a path to various pages. The home page provides a brief description of the web page and a slide below that introduces the purpose of the web site. There are two ways to navigate to the Search for Movies page: the navigation bar and the buttons on the slide. Users who enter the movie page can filter the movie they want by entering the title and year, and the search button next to it is convenient. When you find the movie you want and select the title, the Movie Details page appears to display the details of a particular movie. There is also a table showing the names of the people who made the movie, and you can go

to the next page by selecting the names of the people from that table. The last page is visible only to the logged-in authorized user and better communicates information by visualizing the name, date of birth, and past work of a particular film worker with tables and line diagrams.

The design of the registration page and the login page is very similar because email and password require the same input. If the user has not already registered, there is a link to the registration page. When the user logs in successfully, the 'Register' and 'Login' options on the navigation bar change to 'Logout'. Refresh is possible in 10 minutes, but when the token expires on my web page, I notify the user with a warning and the user is redirected to the login page.



Overall, the location or arrangement of the items are different from the sketch

Figure 9-1 - Sketch All pages design

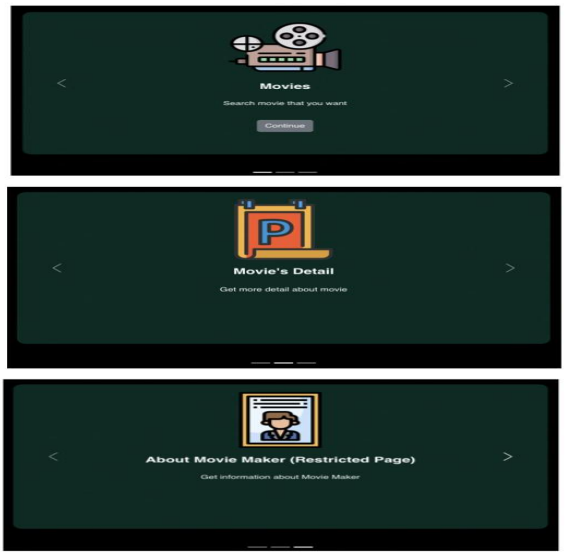


Figure 9-2 - Design Slides

Usability and Quality of Design

The overall design of the application is aimed at minimalism and a clean look with headers, footers and consistent colour configurations across all sites. The black background on the home screen makes the slide stand out and provides a specific and consistent design by applying the same margins to the middle, bottom, and top of the screen.

On the Movie page, the input layout is integrated with consistent margin spacing to label all navigation and element labels to improve visual comfort and increase usefulness. The text consists of larger and smaller titles depending on the importance of the information. The desired location for the title and year search bar was the upper right header, but it was centred due to CSS restrictions. However, it allowed users to focus their eyes on one side.

The design of the registration page and the login page is emphasized, making it visually simple and effective. Like the rest of the web, I linked logins and registrations through links and placed everything centrally. However, except for the home screen, the background colour was white and there was a lot of space, so the completeness seemed low. So, the items in the navigation bar, instead of being skewed to one side, are positioned in two to be symmetrical. In addition, I added a simple button to the personal details page to attract your attention, and added dynamic features using reactive line charts to reduce the space in the margins.

Through this challenge, it became clear that using fewer colors and simple designs does not necessarily give you a vintage look. Overall, the design of the application is simple and functional, but it has the potential to become a much more impressive website with improved skills and experience.

Accessibility

The edited list of Priority 1 Accessibility Requirements is here:

- Provide a text equivalent for every non-text element – alternatives to images, symbols, scripts, graphical buttons, sounds, audio and video files and so on.
  Every element on the page had a text label and never only showed color or shape. It provides sufficient context for users to understand the information provided for AG Grid components that use standardized rules.

- Ensure that all information conveyed with color is also available without color, for example from context or markup.
  Designed the app to deliver information through color as well as other means such as text and icons. For example, a color was used to highlight a specific button, but text was displayed inside the button to indicate the purpose of each button. This allows users who cannot distinguish between colors to use it effectively.

- 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.
  Designed the app to read even if the style sheet is disabled or not supported by your browser. HTML tags are used and content is organized into logical structures to make it easy to read and understand.

- Ensure that text equivalents are updated when dynamic content changes.
  Include dynamic content, such as filtering(using infinite Ag-grid scroll), where search results change. To reflect your changes, we have made sure that the text for these elements is updated in real time.

- Avoid causing the screen to flicker.
  Designed the app to prevent unnecessary blinking that may cause inconvenience to some users.

- Use the clearest and simplest language appropriate for a site's content.
  I don't think it's easy to represent terms that some users may find difficult to understand. For example, Ag-grid header name, "Rated"(The advisory categories for films and computer games). As a whole, used simple language, and added explanation like, user must log-in to see the next page.

- For tables, identify row and column headers – clearly differentiated from the data.

  For tables, identify row and column headers. That is, it is clearly distinguished from the data. A table is included to display the data, and the row and column headers must be clearly identified and distinguished from the data.

Technical Description

Architecture

The application is controlled using components that contain the state of the data, which is passed to the page.

If multiple elements are included in one file during app development, it can become very bulky and confusing. To avoid confusion, group files together and divide them into folders(figure 10-2). The src directory contains two folders, Components and Pages, respectively (Figure 10-3 & 10-4). The Component folder(Figure 10-3) contains components that are used globally for the web, and the Page folder(Figure 10-4) contains pages that you use directly as items in the navigation bar. In addition, the three files that import data from the API (apiMovies.js, apiMovieDetail.js, apiPeopleId.js) are located in the same location as the App.js to avoid confusion with others.

Therefore, if the page has the necessary elements, I can use the elements in the Component folder to keep the all files sized properly.

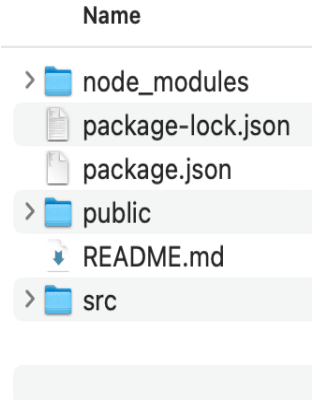


Figure 10-1 Whole

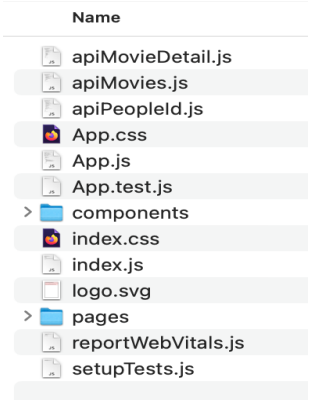


Figure 10-2 src

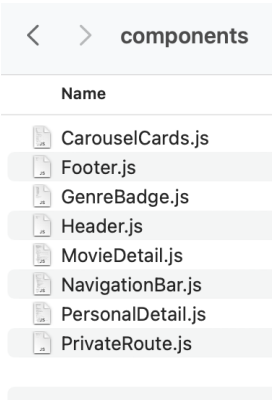


Figure 10-3 components

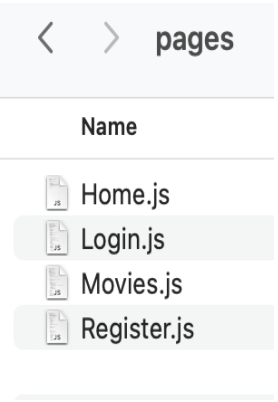


Figure 10-4 pages

Test plan

Task	Expected Outcome	Results	Appendix B
Load Home Page (click slide the carousel cards)	The home page is shown + carousel cards slide moving smoothly	OK	01
Use navbar, nav to movies	movies page shown	OK	02
Search a movie title with search-bar Search a movie using year drop-bar	can input letters and click /input year	OK	03
Press "Search" button	The results update, by search bar and year bar	OK	04
Using three parameter for request to server using dynamic url	Updates Ag grid with filtered movies	OK	04
"Wrong" title handled smoothly	Table renders no rows	OK	05
Click pagination and move next page	Can go 1 to 122page	OK	06
Click row to see specific movie-detail page	Click movie title cell, then move next page	OK	04
Movie detail page displays info	With more details about movie, poster, released in, plot, table(movie maker)etc	OK	07
Click table "Name" row to see individual movie maker *actor, editor, director..)	Click link, then go to next page	OK	07
Navigate to login	Goes to login page	OK	02
Navigate to register	Goes to register page	OK	02
link from register to login (Click here to login!)	Login page is displayed	OK	08
After login	Page goes Home page	OK	09
Failed to submit empty form (auth)	"Request body incomplete, both email and pw are required" error message appear	OK	13
Enter in correct name and password to login	"Login successfully"	OK	12
Email/PW entered does not exist/wrong	"Invalid email or password"	OK	14
Entering existing username on register	"User already exists"	OK	15
After, user login successfully	Navigation bar change (logout appear)	OK	09
When logged in, last page of personal detail page is displayed.	Yes, the Ag-grid and line chart are displayed	OK	10
After expired token, get refresh token (not need to login)	Not to login after the token is expired	No	
1min before, token is expired, alert message appear to refresh	Alert message appear	OK	11
User can log out (successfully)	When logout, then Register/Login appear on navigation bar	OK	

Difficulties / Exclusions / unresolved & persistent errors /

It was easy to import an existing list of 100 movies into AgGrid and filter them using the search bar and year drop-down. However, an error occurred when implementing infinite scrolling. It took a lot of time to update movie titles in real time at the URL and improve the process of requesting data from the server. I had to change the variable several times and check for errors using console.log(). There were some errors when using the rowData property in the "Infinite" state. Therefore, I defined the rowData property, but it was not used in the file, which resulted in one warning.

Although we have successfully implemented search and filtering, there are still visually unnecessary elements. I need to adjust the AgGrid page to show the number of filtered videos after searching, but I don't know how to fix it.

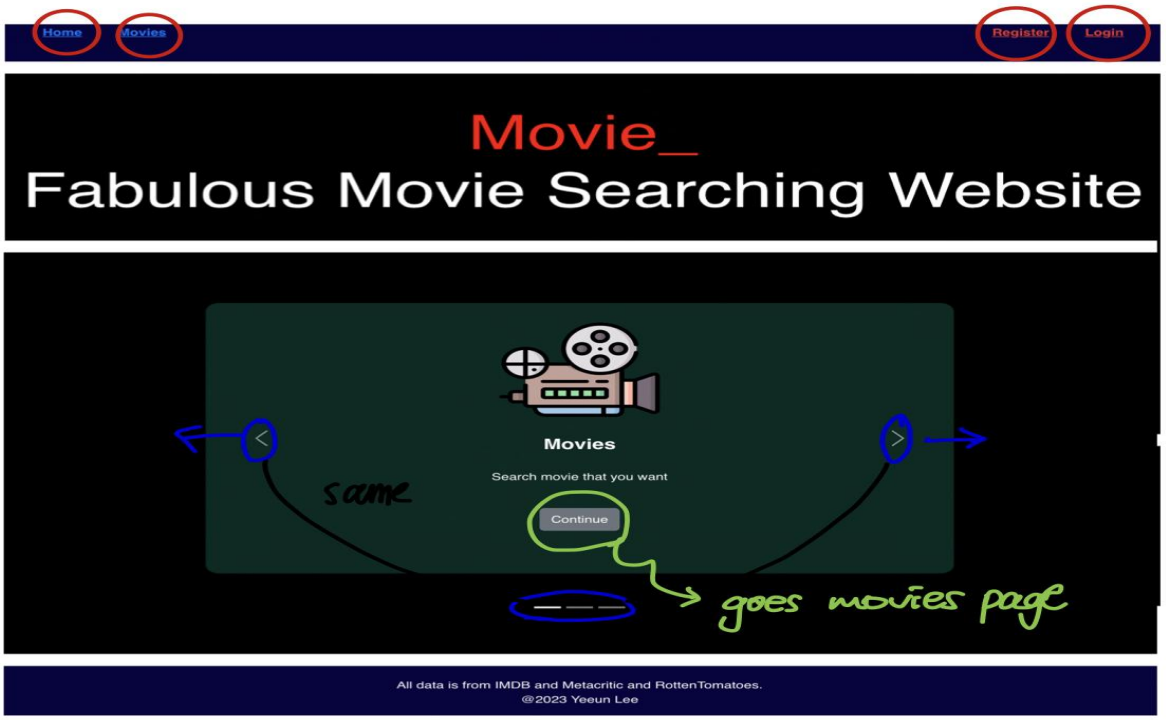
Also, there seems to be a problem with the refresh token. It will be issued when you log in, but it will not be reissued after 10 minutes, and you will be logged out immediately

Extensions (Optional)

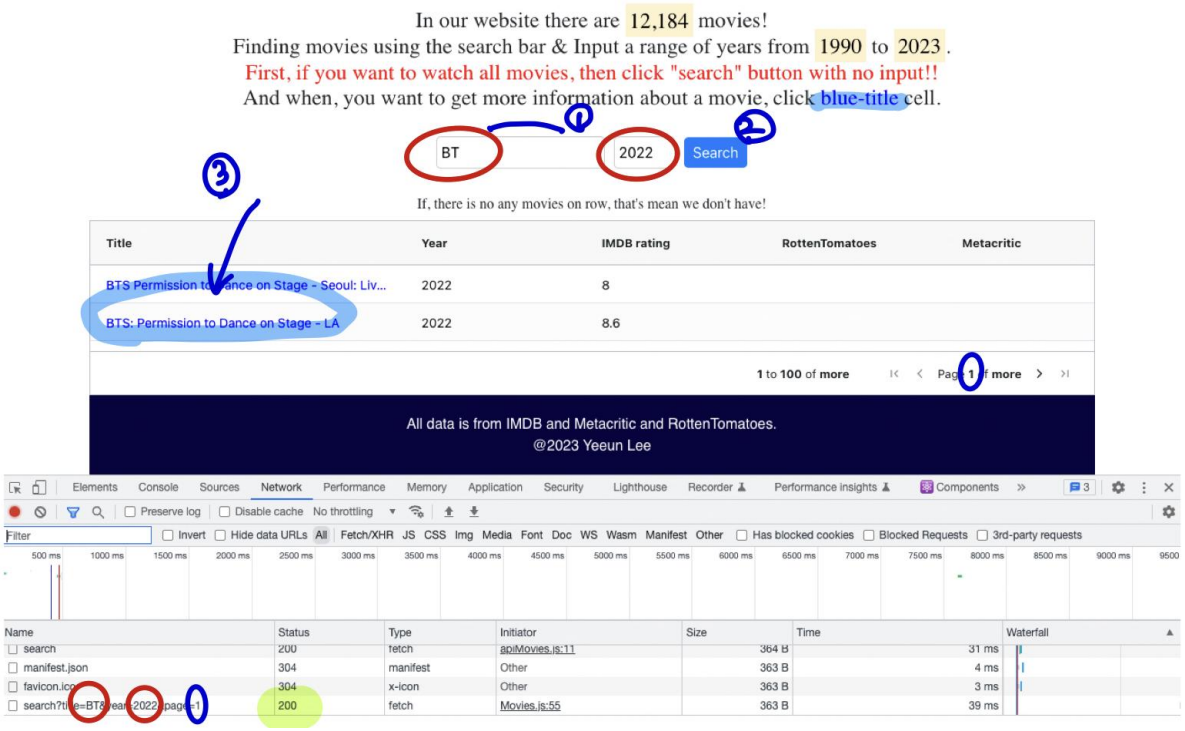
The extension that I want to do is add input validation to the authentication form using a regular expression. Alternatively, if an external module exists, I would like to introduce an external module. The user logs in with a common email and password, so a library must be available to verify the field requirements. And also, know how to do handle infinite Ag-grid scroll more accurately for improving my web page.

User guide

1. Upon loading the app, the user will be greeted with the Home page and can navigate using the navbar as shown!

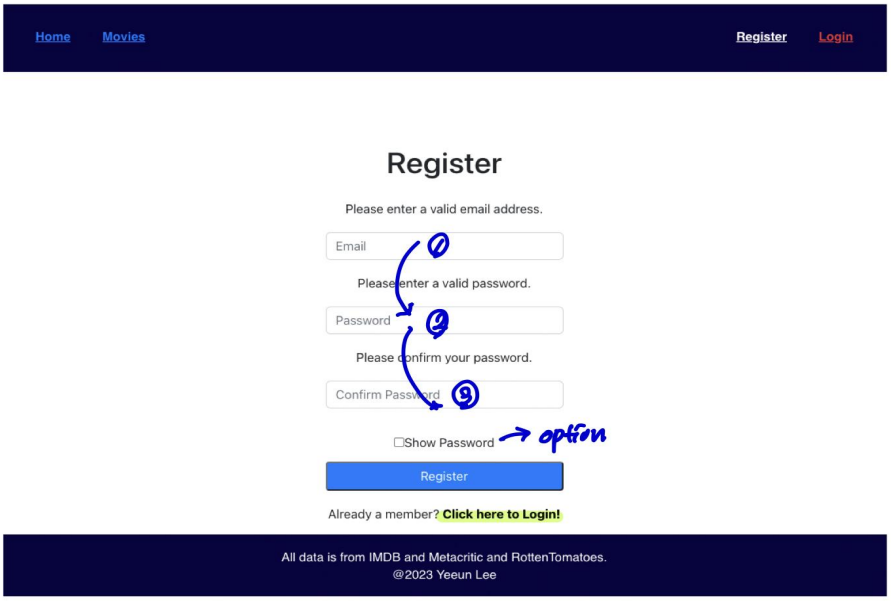


2. Entering an input in the search bar or year bar on the Movies page will change the Ag-grid with each letter input.



3. When, user login, then last page “Person detail page” is displayed. In here, user can see two kind of type data using button.

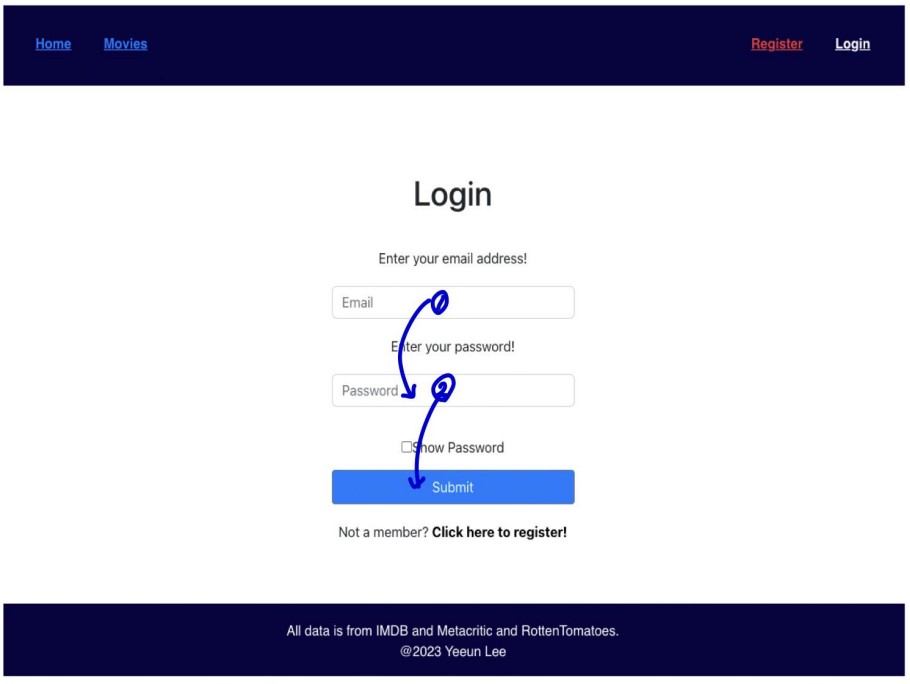
4. Enthering a valid email and password on the Register page will successfully register a new account. #show password checkbox is option



5. And, click “Click here to Login!” then page is changed.

   Entering a valid email and password on the Login page will log the user in and then goes Home page.

   In the navigation menu will change: appear “Logout”.



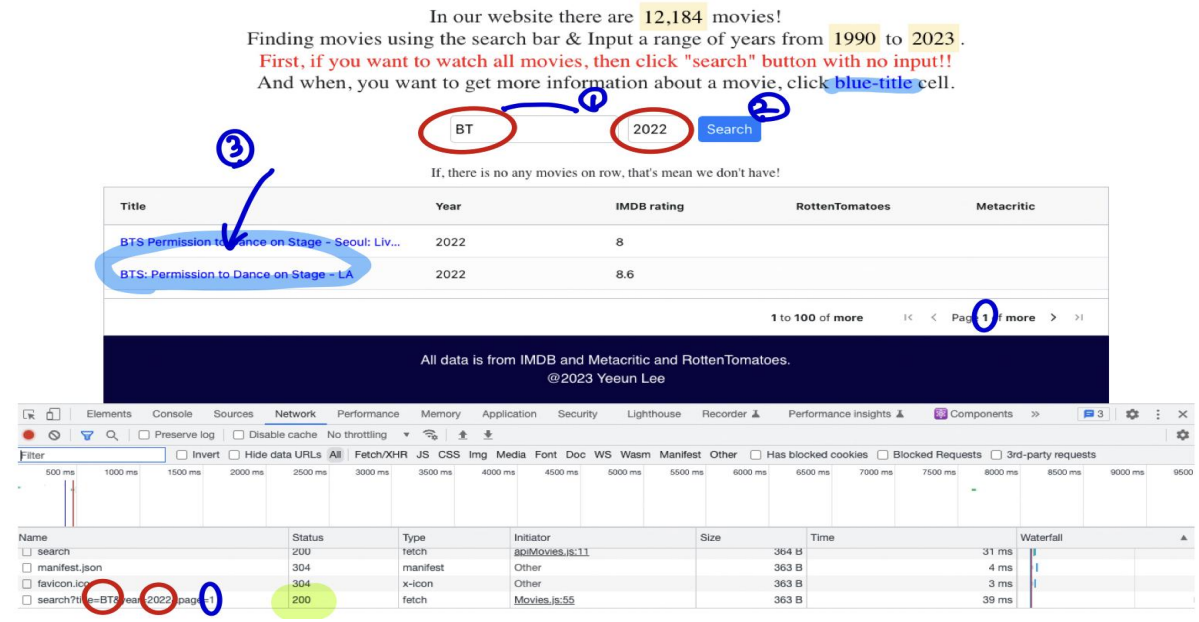
Appendices as you require them

01.

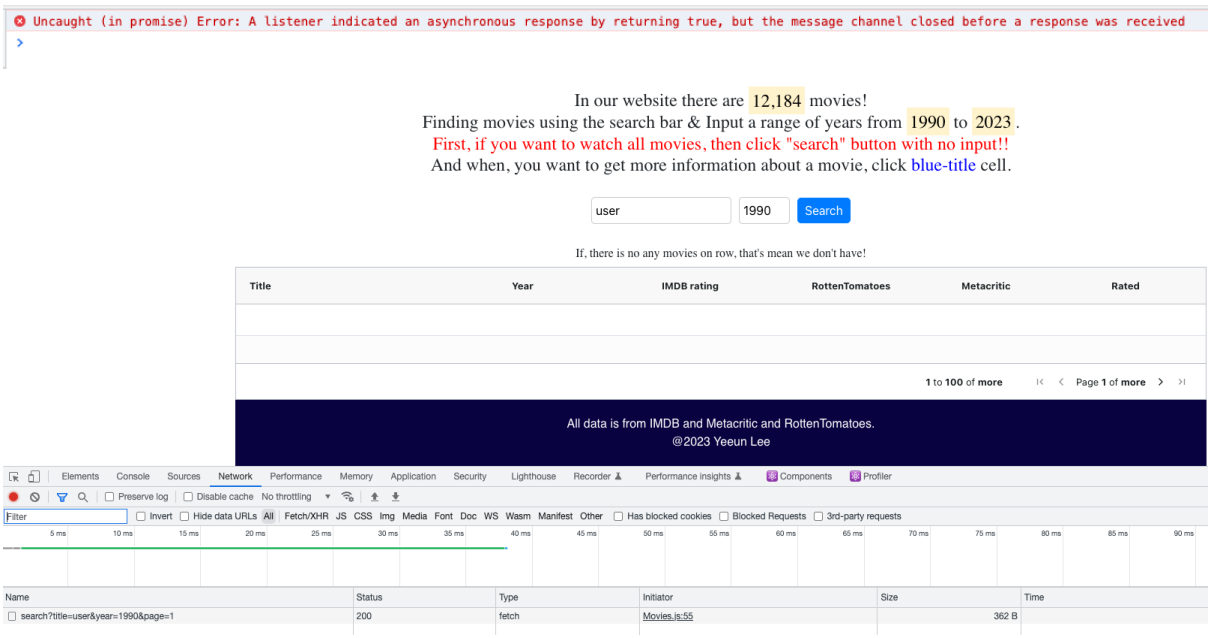
02.

03.

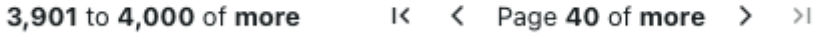
04.



05. url is updated, but there is no movies on Row.



06.



07.

Finding Nemo

After his son is captured in the Great Barrier Reef and taken to Sydney, a timid clownfish sets out on a journey to bring him home.

Released in: 2003
Runtime: 100 min
Country: United States
Genres:
Adventure Animation Comedy
Internet Movie Database: 8.2 / 10
Rotten Tomatoes: 99%
Metacritic: 90 / 100
Box Office: $380,843,261.00

Role	Name	Character(s)
composer	Thomas Newman	
actor	Albert Brooks, Alexander Gould, Willem Dafoe	Marlin, Nemo, Gill
actress	Ellen DeGeneres	Dory
director	Andrew Stanton, Lee Unkrich	
writer	Bob Peterson, David Reynolds	
producer	Graham Walters	

All data is from IMDB and Metacritic and RottenTomatoes.

08. Click here to login

Home Movies Register Login

Register

Registeration successful. Login please!

Please enter a valid email address.

mrnam0513@naver.com

Please enter a valid password.

love

Please confirm your password.

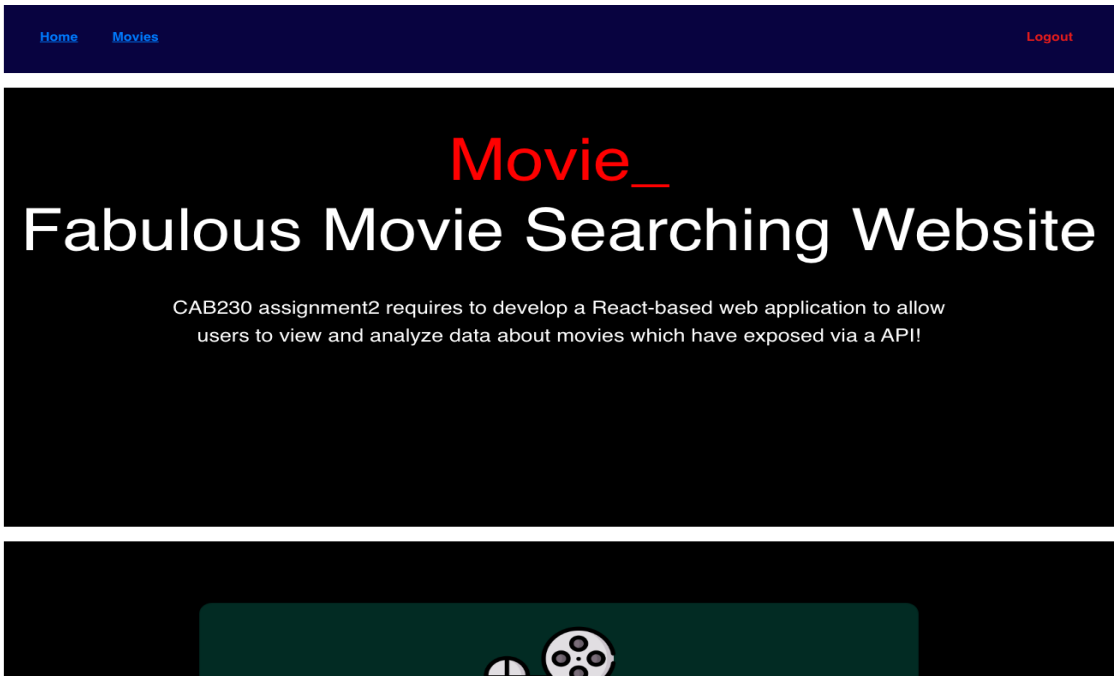
love

☑ Show Password

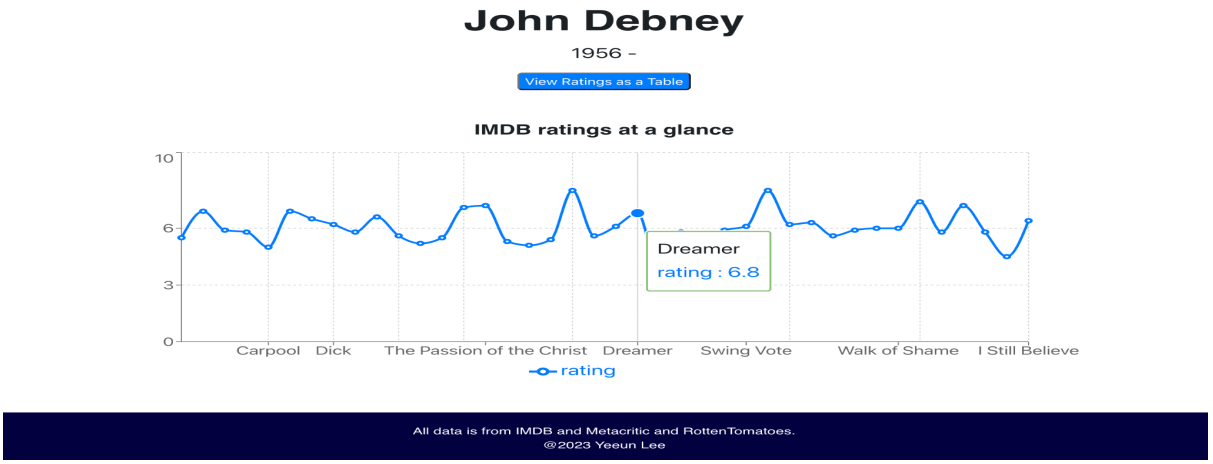
Register

Already a member? **Click here to Login!**

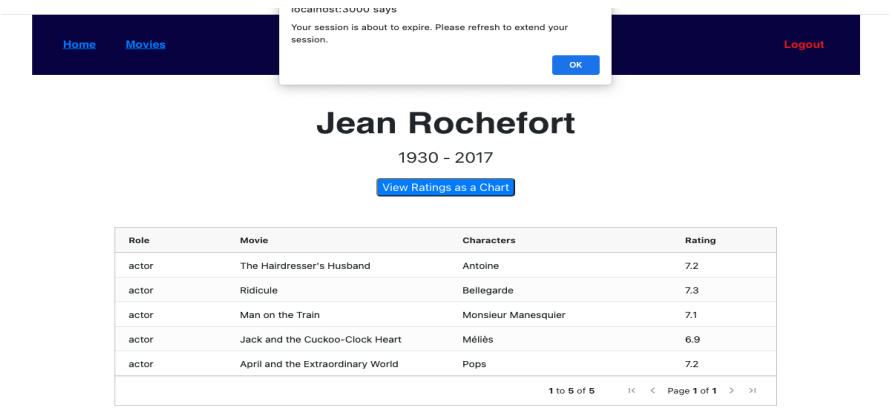
09. After login, go home page & logout appear



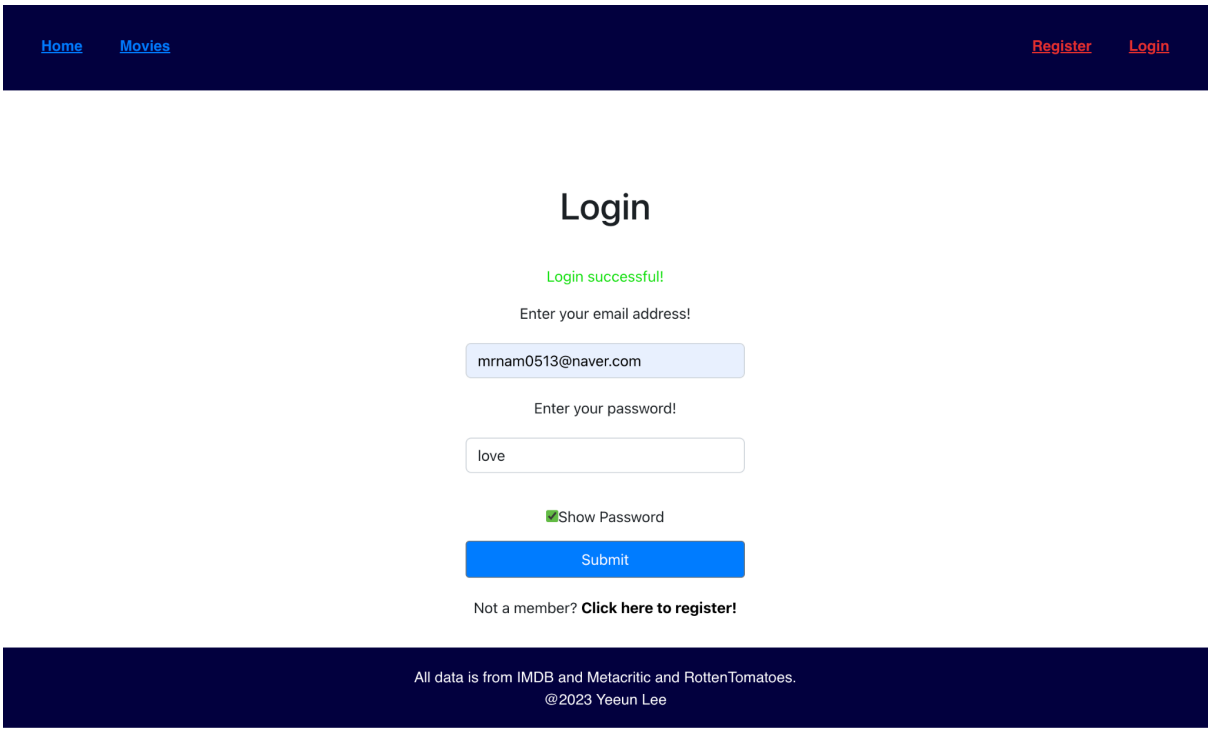
10. Restricted page



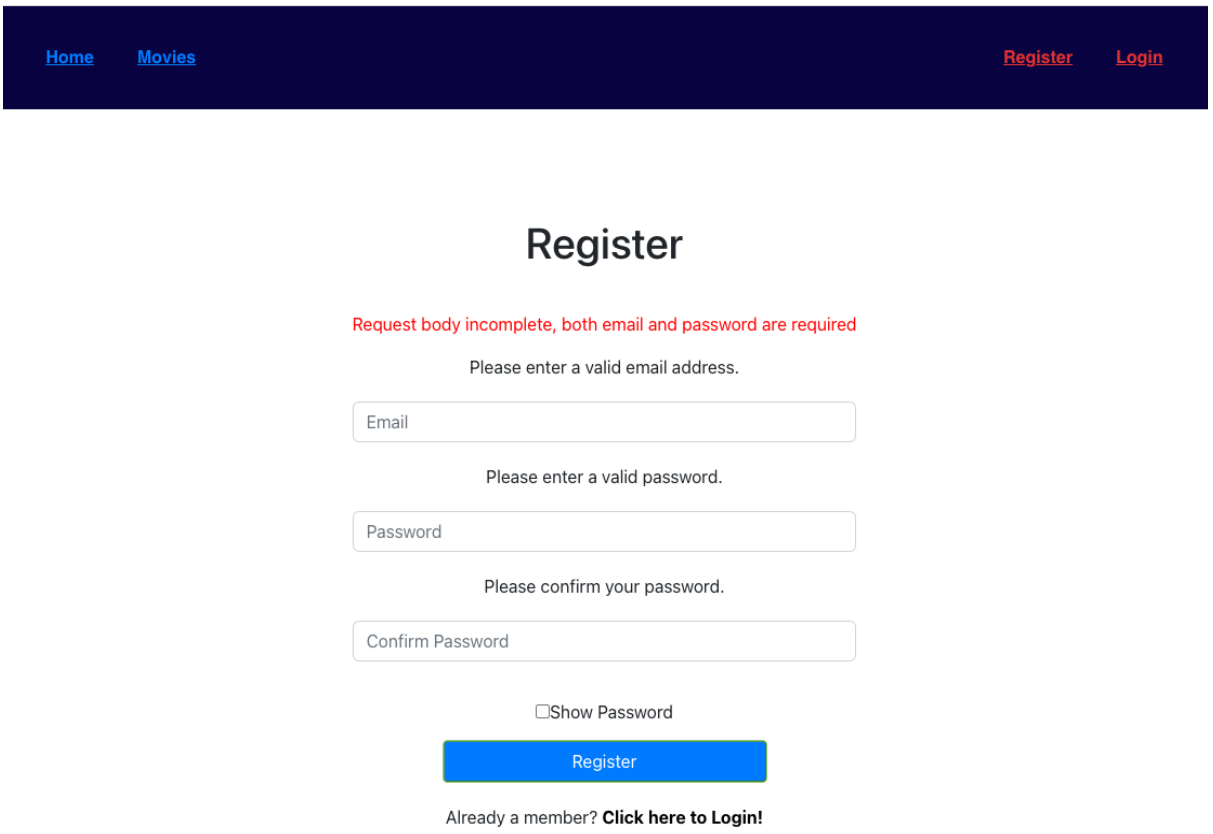
11. Alert



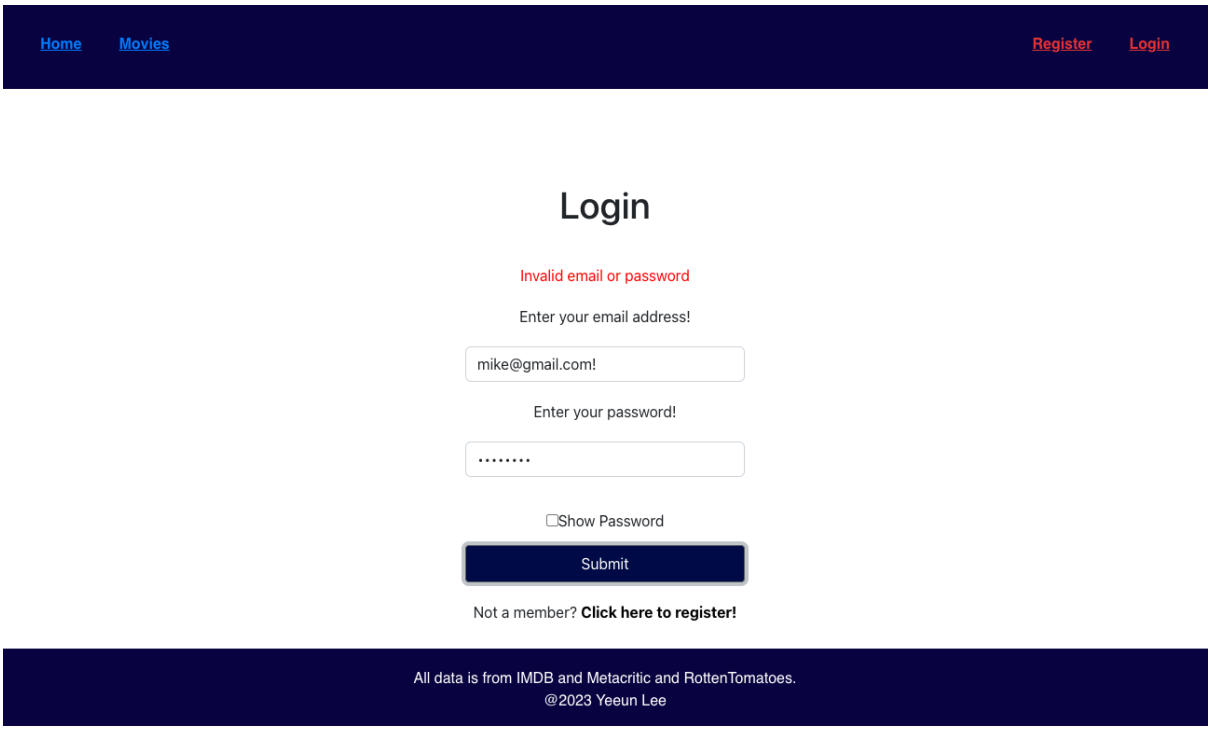
12.



13. Empty Auth Form



14. Invalid



15. Already exists