

Ghibli Archive



Overview

The Challenge

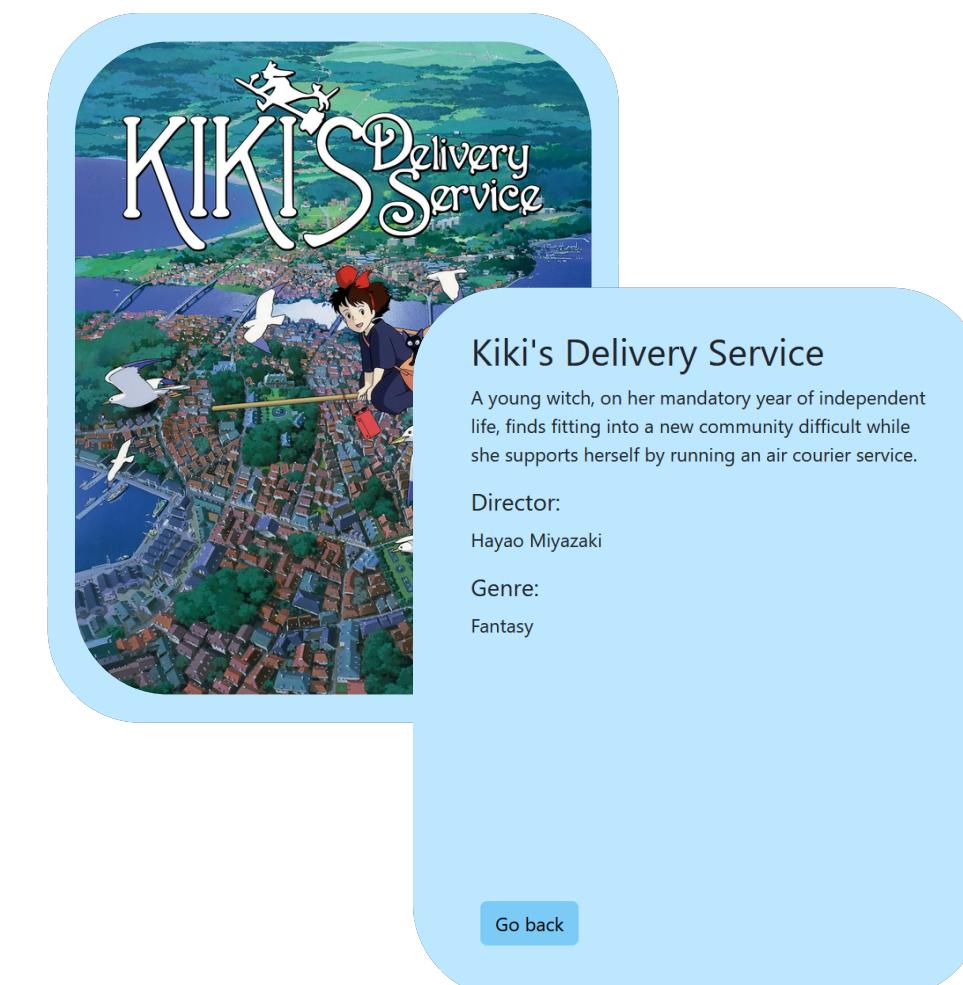
Create a full-stack application using the MERN stack application, that provides the user with access to information about different Studio Ghibli movies, their directors and movie genres. Users can create an account, update their personal info and create a list of their favorite movies.

The Goal

As a part of my web development course at CareerFoundry, my goal was to showcase my ability to build the project's backend and frontend from scratch, and combine them to add a full-stack application to display in my portfolio.

The Process

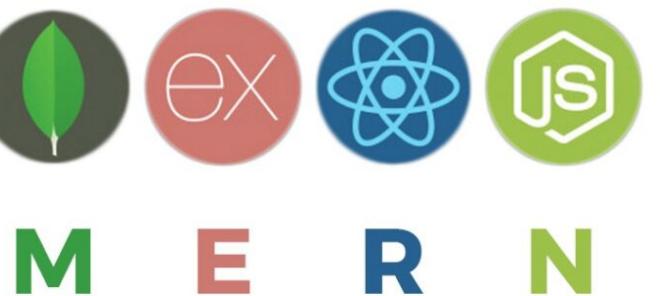
Created the database where the movies and users information would be stored. Then, using a frontend framework (React) to code the client-side of the application. Also wrote user stories to better understand how users would interact with the service. During the whole process I tested and debug multiple features.



Technologies

MERN Stack

MongoDB
Express
React
Node.js



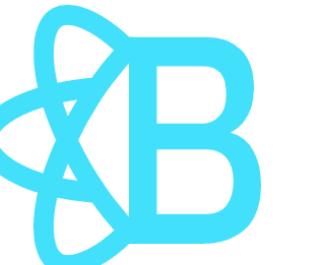
Render

Server hosting service



React Bootstrap

for website styling

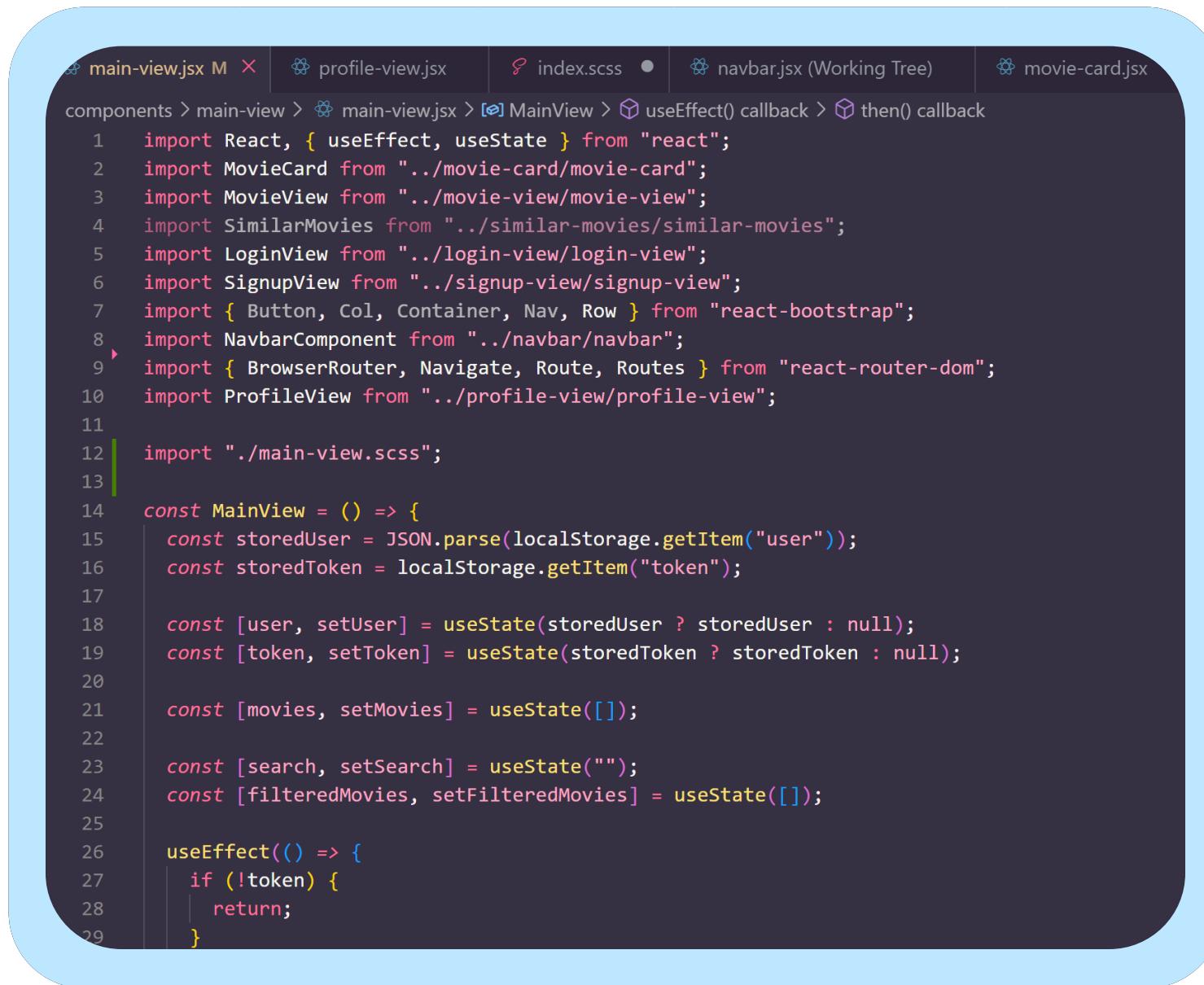


Github

code storing and
management



Approach



```
main-view.jsx M | profile-view.jsx | index.scss | navbar.jsx (Working Tree) | movie-card.jsx
components > main-view > main-view.jsx > MainView > useEffect() callback > then() callback
1 import React, { useEffect, useState } from "react";
2 import MovieCard from "../movie-card/movie-card";
3 import MovieView from "../movie-view/movie-view";
4 import SimilarMovies from "../similar-movies/similar-movies";
5 import LoginView from "../login-view/login-view";
6 import SignupView from "../signup-view/signup-view";
7 import { Button, Col, Container, Nav, Row } from "react-bootstrap";
8 import NavbarComponent from "../navbar/navbar";
9 import { BrowserRouter, Navigate, Route, Routes } from "react-router-dom";
10 import ProfileView from "../profile-view/profile-view";
11
12 import "./main-view.scss";
13
14 const MainView = () => {
15   const storedUser = JSON.parse(localStorage.getItem("user"));
16   const storedToken = localStorage.getItem("token");
17
18   const [user, setUser] = useState(storedUser ? storedUser : null);
19   const [token, setToken] = useState(storedToken ? storedToken : null);
20
21   const [movies, setMovies] = useState([]);
22
23   const [search, setSearch] = useState("");
24   const [filteredMovies, setFilteredMovies] = useState([]);
25
26   useEffect(() => {
27     if (!token) {
28       return;
29     }
29 }
```

Server-side

I started development with the server-side. I build a RESTful API using Node.js and Express, which uses endpoints to identify the resources on the server the client wants to access. The API interacts with the database MongoDB, and can be accessed through HTTP requests such as GET and POST. CRUD Method (Create, Read, Update, Delete) are used to retrieve and save information.

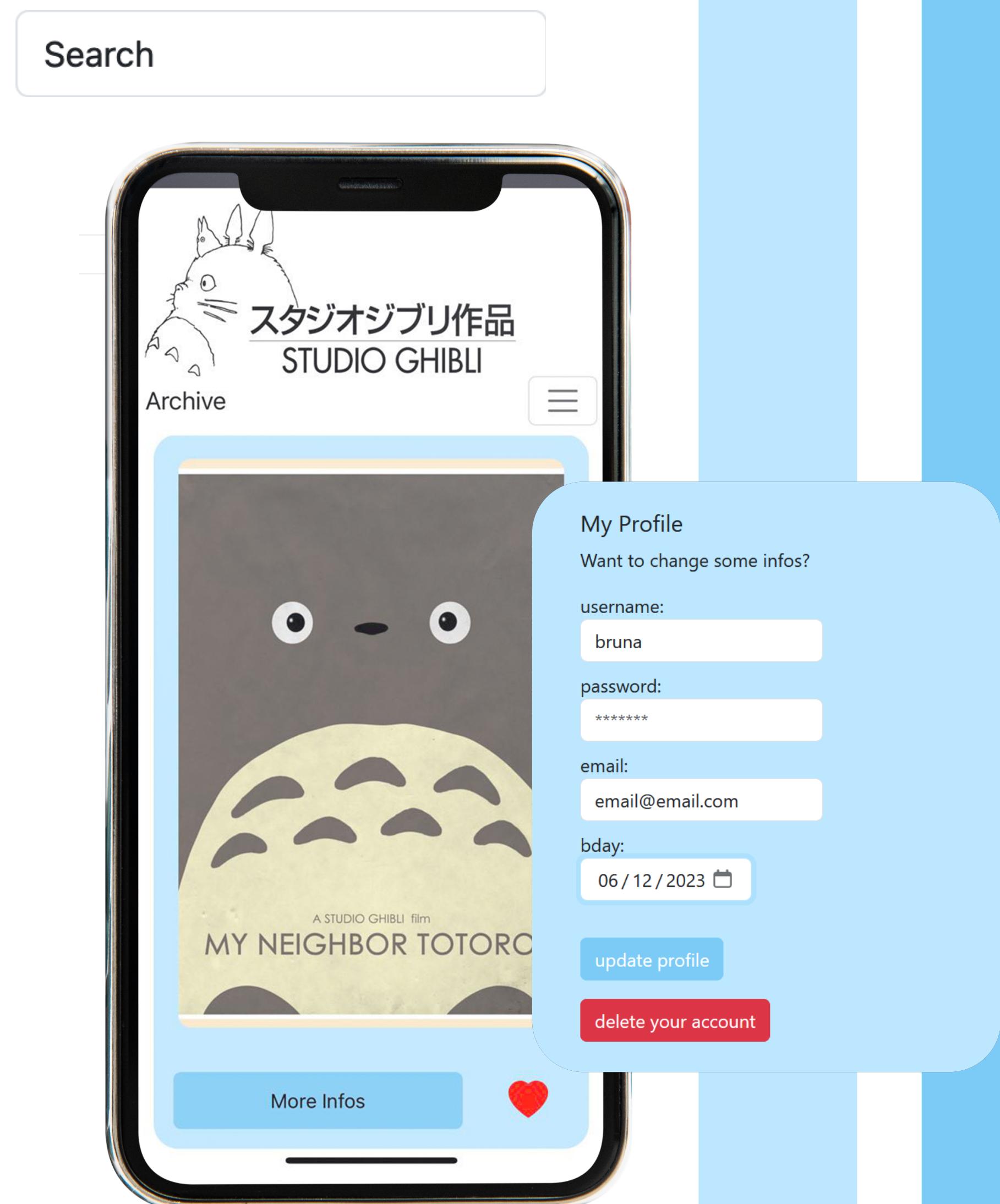
The backend is hosted in a service called Render, which enables easy deployment.

Client-side

After building the server-side, It was time for the client-side. For his project, the framework chosen was React.

The web application provides the users with multiple views, starting with the signup and login view. Once logged in, users can explore the website further, starting from the homepage, where all the movies are listed. Upon clicking on the details button for any movie, the user is redirected to the details page, with more detailed information about that title. On the profile view the user can update their information, as well as see their favorite movies list.

For the layout, React Bootstrap was used, providing a straightforward and elegant approach for the design.



Retrospective

What went well?

I learned so much about designing full-stack applications throughout this project. It was very satisfying seeing all the pieces come together and working smoothly towards the end. I feel more confident in my ability to build full-stack applications after completing this project.

Difficulties:

There was a lot of testing and debugging involved throughout the whole project. It was important to test the application in different browsers, as well as in mobile vs. desktop. Multiple users tested the application on their respective devices, and with the received feedback I gained more insight into how to better the look and feel, as well as the general user experience.

Final thoughts

I had a lot of fun building Ghibli Archive from scratch. There is still a lot of room for implementing new features in the project. As the next steps I would add more movies to the database, implement a search function also for directors and genres, as well as making some UI adjustments.

