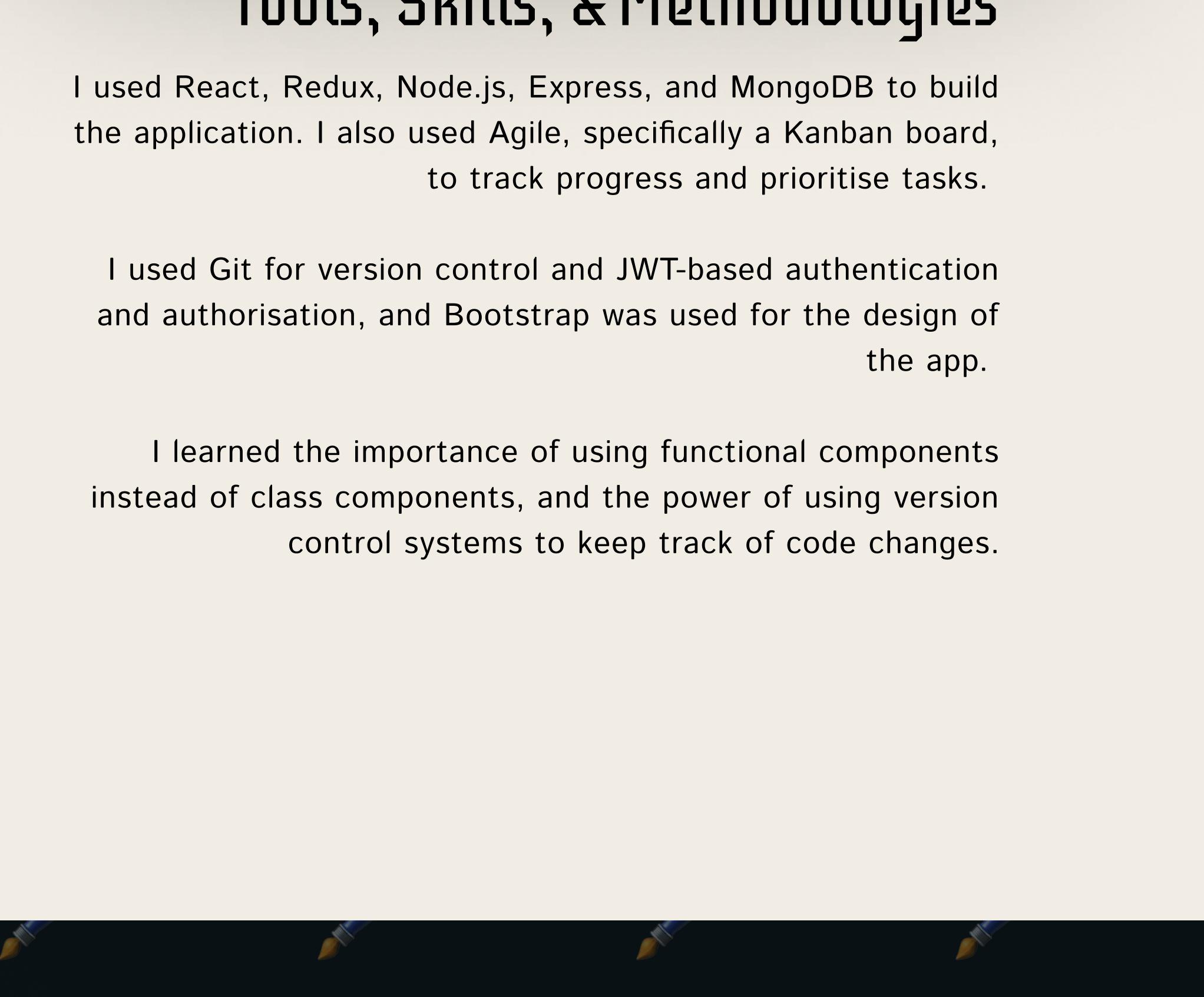


# SCYNAPSE

## For Sci-fi Lovers

### Overview

Synapse is a web application that allows science-fiction enthusiasts to access information about different sci-fi movies, directors, and genres, update their personal information and create a list of their favourite movies.



### Purpose & Context

The purpose of the project was to create a web application that provides movie information to users and allows them to create a list of their favourite movies. I was given a project brief, and some user stories and technical requirements to complete this project.

### Objective

The objective of the project is to create a web application that meets the needs and expectations of the target users according to the given user stories, providing users with information about movies, directors and genres, and allowing users to update their personal information and create a list of their favourite movies.

### Tools, Skills, & Methodologies

I used React, Redux, Node.js, Express, and MongoDB to build the application. I also used Agile, specifically a Kanban board, to track progress and prioritise tasks.

I used Git for version control and JWT-based authentication and authorisation, and Bootstrap was used for the design of the app.

I learned the importance of using functional components instead of class components, and the power of using version control systems to keep track of code changes.

## PHASE 1

### Design & Planning

★ USER STORIES ★

### Designing Synapse

The design and planning phase of the project was focused on understanding the needs and expectations of the target users.

To do this, I created user flows for the user stories that described the various features and functionality of the app.

I also designed an interface in Figma, taking into account UX and UI principles to help make the app accessible and usable.

These materials helped me to stay focused on the goal of the project and ensure that the app met the needs of the users.

### Tracking Progress

To keep track of my progress and prioritise tasks, I used a Kanban board, which helped me to stay organised and focused throughout the project.

## PHASE 2

### Front-End Development

### Summary

For the front-end of the app, I was responsible for building the client-side using React. My tasks included implementing various interface views, handling data through the REST API endpoints, and ensuring a polished, accessible, and responsive user experience on any device.

### Decisions Made

One important decision I made was to use Redux for state management, as I believed it would provide the most efficient and scalable solution for Synapse.

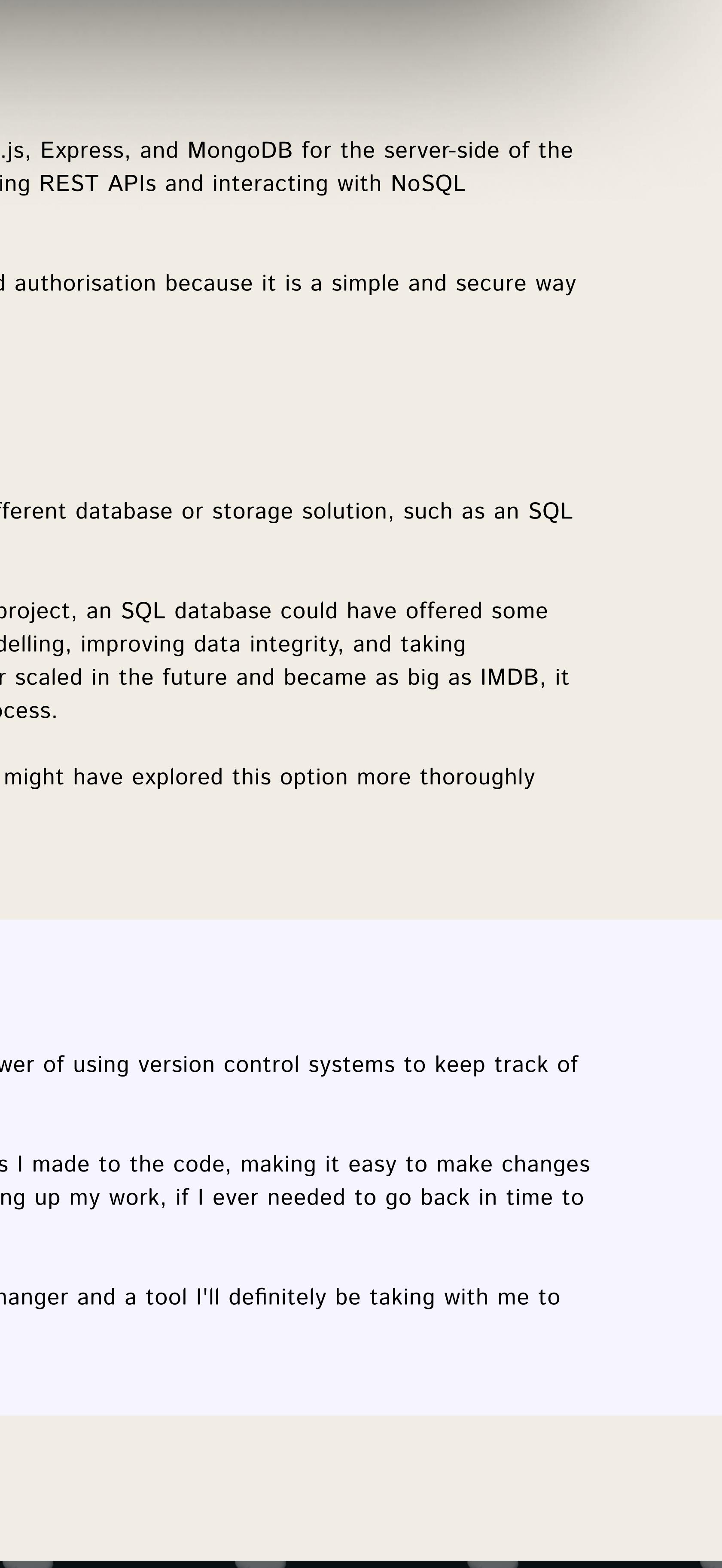
This decision ended up being quite challenging, as I had to spend a significant amount of time learning and implementing Redux. However, the benefits of using Redux made it worth the effort, as it allowed me to manage the app's global state in a predictable and maintainable way.

### Challenges

If I could do anything differently in the future, it would be to consider using a different design solution rather than relying on a library like Bootstrap.

While it was convenient to use at first, I found myself spending a lot of time trying to customise it to fit the exact look and feel I was going for.

In the future, I think I would either try using something like styled-components or even just building the styles from scratch conventionally to have more control and flexibility in the design process.



### Lessons Learned

One lesson I learned during this project was the importance of using functional components instead of class components whenever possible.

Initially, I was using class components for the majority of my UI because I was more familiar with them. However, as I continued to work on the project, I realised that functional components had several benefits, including improved readability and simplicity, and performance due to fewer overheads. I decided to refactor my code to use functional components wherever possible, and I believe it improved the efficiency and maintainability of my app while making it more future proof as functional components are the direction that React is going in.

In the future, I will make a conscious effort to use functional components as much as possible in my projects.

## PHASE 3

### Back-End Development

### Summary

For the back-end of the app, I was responsible for building the server-side component using Node.js, Express, and MongoDB.

My tasks included designing, implementing, and testing a REST API, connecting the API to a MongoDB database, and implementing user authentication and authorisation.

Method	URL	Description
GET	/movies	Returns list of all movies.
GET	/movies/:Title	Returns data about a single movie.
GET	/genres/:Genre	Returns data about a genre.
GET	/directors/:Director	Returns data about a director.
POST	/users	Register a new user.
PUT	/users/:Username	Update a user's details.
DELETE	/users	Deregister a user.
POST	/users/:Username/favorites/:MovieID	Add movie to favorites.
DELETE	/users/:Username/favorites/:MovieID	Remove movie from favorites.

### Decisions Made

In terms of decision-making, I chose to use Node.js, Express, and MongoDB for the server-side of the application because they are well-suited for building REST APIs and interacting with NoSQL databases.

I also chose to use JWT-based authentication and authorisation because it is a simple and secure way to manage user sessions.

This project taught me that Git is a total game-changer and a tool I'll definitely be taking with me to future projects.

### Lessons Learned

One lesson I learned during this project is the power of using version control systems to keep track of code changes.

With Git, I was able to easily track all the updates I made to the code, making it easy to make changes and be confident that I wasn't permanently messing up my work, if I ever needed to go back in time to an older version of the code, Git had my back.

If I'd considered potential future scaling issues, I might have explored this option more thoroughly before deciding on a database solution.

## CONCLUSION

### Closing thoughts

Building Synapse was a challenging but rewarding experience. The app was a great example of how to use React and Node.js to build a responsive and scalable web application.

Through this project, I learned the importance of user-centred design, efficient state management, and proper testing and deployment.

I also learned valuable lessons about using version control systems, code reusability and scalability.

Overall, the end product was a well-designed and functional app that serves the needs of sci-fi movie enthusiasts well.

### CREDITS

#### Full-Stack Development

Kerr Campbell

#### Tutorship

Adewunmi Bamishigbin

#### Mentorship

Ndi Selma Ekfvei