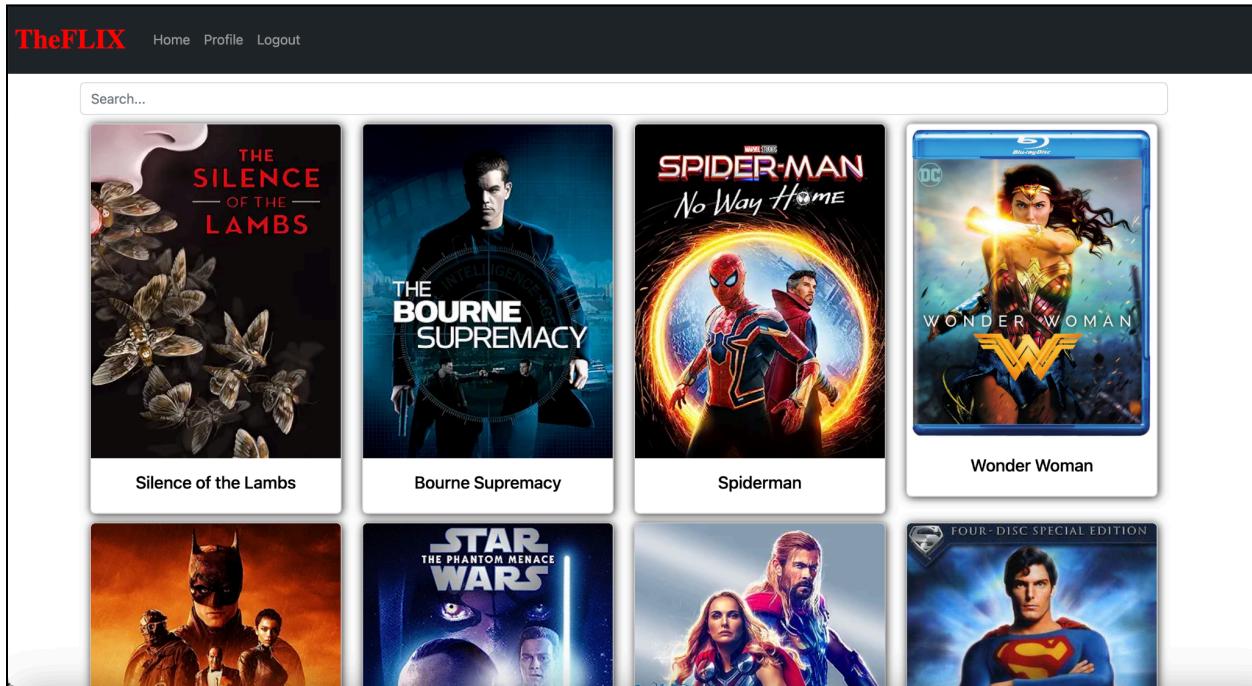


# Case Study for TheFlix Project

TheFlix - server-side component of a “movies” web application.



## Overview:

TheFlix is a web app (Front and Back End), developed using the MERN stack (MongoDB, Express, React and Node.js), that provides users with access to information about movies, directors, and genres. Users are able to create an account, update their personal data, get detailed information on selected movies and create a list of favorite movies.

## Objective:

The objective of TheFlix Front-End Application is to build the Client-side component of a Full-stack “movies” web application to showcase my knowledge as a full stack developer.

## Tools, Skills, Methodologies:

- Visual Studio Code/Eslint/Git/Github
- HTML, CSS, Javascript, Bootstrap, React, Redux

## Context:

In today's tech landscape, the scope of a JavaScript developer extends beyond just mastering frontend development. It's increasingly crucial to engage with and even construct our own APIs. With this evolving trend in mind, I previously developed a REST API that interfaces with a MongoDB database, managing an assortment of movie data.

For this particular project, I ventured into building the client-side of the application utilizing React and Redux. This integration resulted in a comprehensive web application that encompasses both client and server-side, exemplifying my proficiency in full-stack JavaScript development. The application offers users an interactive platform to explore an array of movies, their directors, and genres. It achieves this through API requests to the server-side which I engineered using Node.js, Express.js, auth.js, and passport.js, all of which are hosted on Heroku.

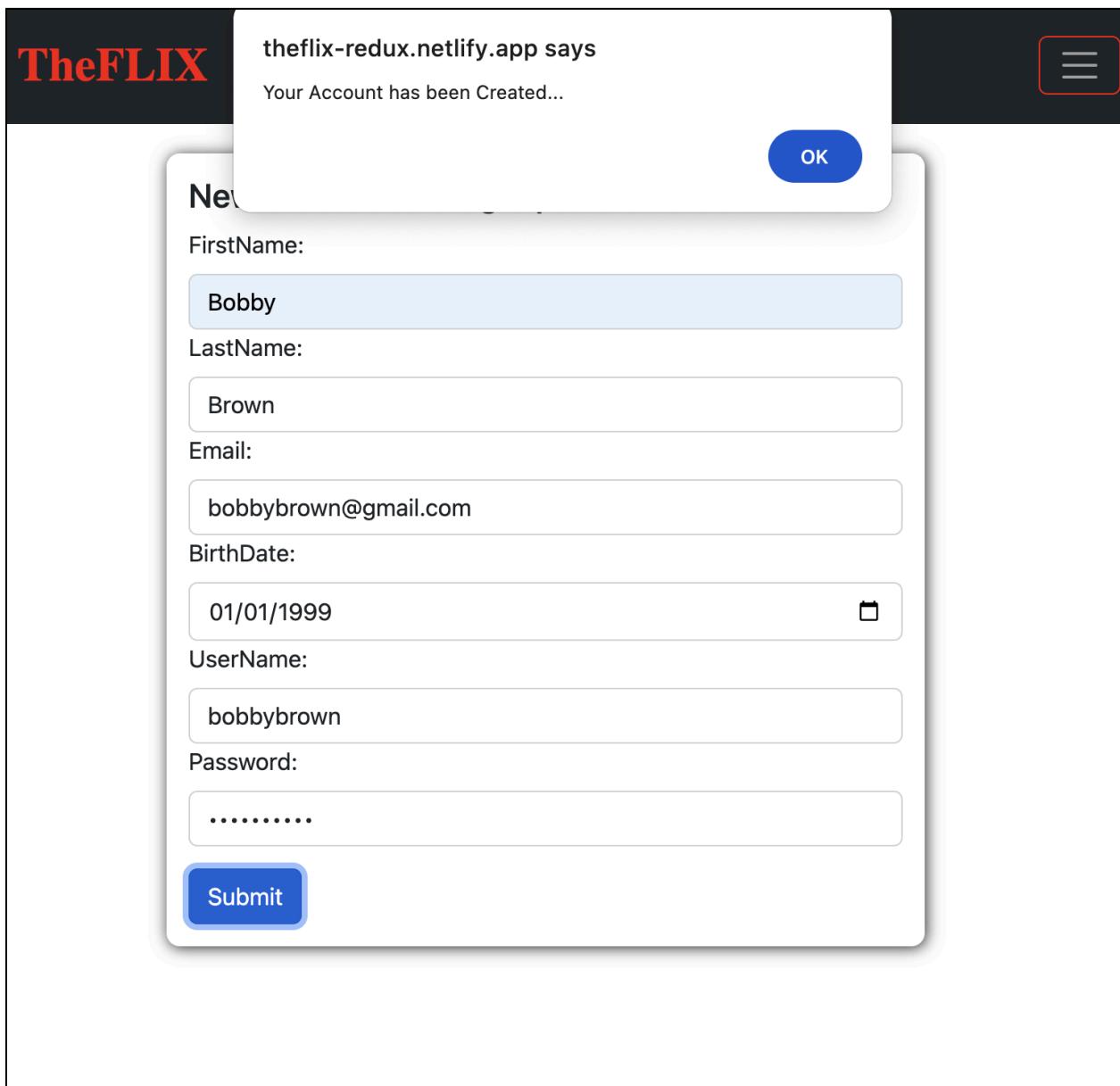
Furthermore, these API requests retrieve data from a MongoDB non-relational database, which is cloud-hosted via MongoDB Atlas. To delve deeper into the server-side functionalities, feel free to visit my portfolio at [www.terryrossi.com](http://www.terryrossi.com).

This project is a testament to my expertise in the full spectrum of full-stack JavaScript development, encompassing areas like APIs, web servers, frameworks, databases, business logic, authentication, and data security.

The ensemble of technologies I utilized for this project is referred to as the MERN stack, an acronym for MongoDB, Express, React, and Node.js. This stack symbolizes the modern approach to building powerful and efficient web applications.

Significant emphasis was placed on the app's progressive features, particularly the dynamic header. It integrates a responsive hamburger menu that smartly adapts to different user states, catering to registered, unregistered, logged-in, or logged-out users for a seamless experience.

## Registration



Logged out Header

## Logged in Header

Search...



## Login

Existing User. Please Login...

UserName:

bobbybrown

Password:

.....

**Submit**

## User Profile

TheFLIX



### User Profile

FirstName:

Bobby

LastName:

Brown

Email:

bobbybrown@gmail.com

**Update User**

## Movie Detail (with option to favorite the movie and list of similar movies)

# TheFLIX



Wonder Woman

Genre: Sci-fi

Description: The Last of the Amazons

Director: Clint Eastwood

[Favorite](#)

[Back to Movie List](#)

---

### Similar Movies :



## User Profile showing list of favorite movies

TheFLIX

User Profile

FirstName:

Bobby

LastName:

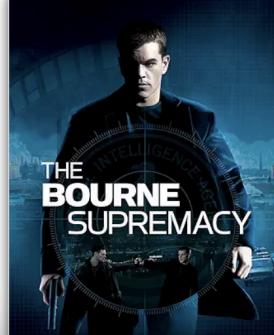
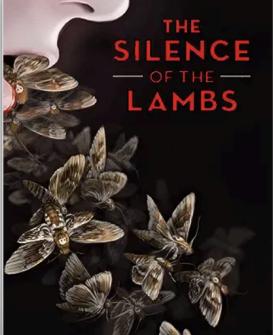
Bondurant

Email:

bbondurant@gmail.com

Update User

Favorite Movies :



## Outcome:

The completed app boasts an aesthetic reminiscent of Netflix's visual style, yet unique in its own right. Functionality aligns seamlessly with the initial requirements, having undergone thorough testing. User interfaces are intuitive, with clear controls and error messages. The application's flow across various components is smooth, adhering to set requirements. Overall, the app stands as a robust showcase of my grasp on technologies like JavaScript, React, and Redux.

## **Challenges:**

- Transitioning the app from pure React to React with Redux posed a significant learning curve but proved to be an enriching experience.
- A notable challenge was in API integration and error handling from the backend. The backend, initially not fully optimized for React/Redux, exhibited some discrepancies in API call expectations and error messaging, necessitating backend modifications for compatibility.
- Drawing from my background as a former Database Administrator, I observed that utilizing a non-relational database like MongoDB presented unique challenges. In certain scenarios, it required more time and effort compared to traditional relational databases, which I found to be generally more robust and reliable.

## **Future Steps:**

Reflecting on the project, my newfound proficiency in React/Redux has illuminated areas for improvement in the server-side architecture, particularly in enhancing its communication with the frontend. This project was instrumental in deepening my understanding of the intricacies involved in API interactions with React and Redux, and it will significantly influence my approach to API design in the future for enhanced compatibility and efficiency. Additionally, drawing on my experience as a former DBA, I've recognized the unique challenges and opportunities presented by using a non-relational database like MongoDB. While it offers flexibility, it can also be more time-consuming and less reliable than traditional relational databases in certain scenarios. Moving forward, I aim to leverage these insights to optimize database choices and integration, ensuring a robust and seamless application experience.