

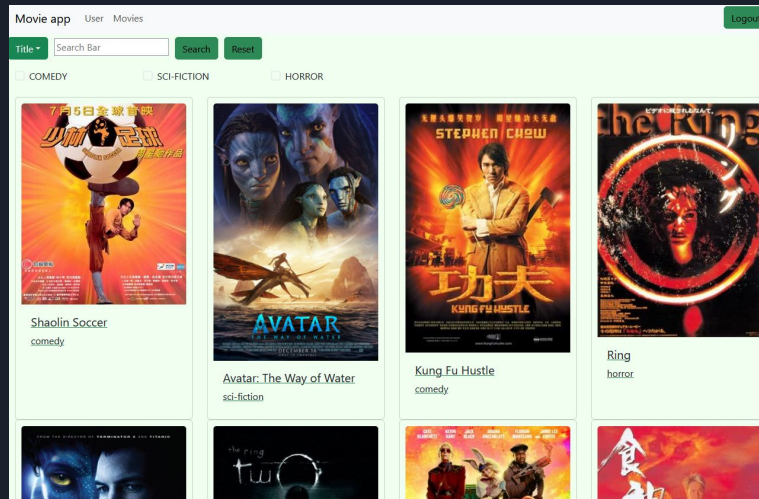


Case Study - MyFlix

Ching Hin Man

Overview

MyFlix is a web application designed with MERN (MongoDB, Express, React, Node). It provides users with access to the information about movies in the database. Users are able to check the information of the movies, including introduction, director and genre. Users also able to save their favorite movies into their favorite movies list.



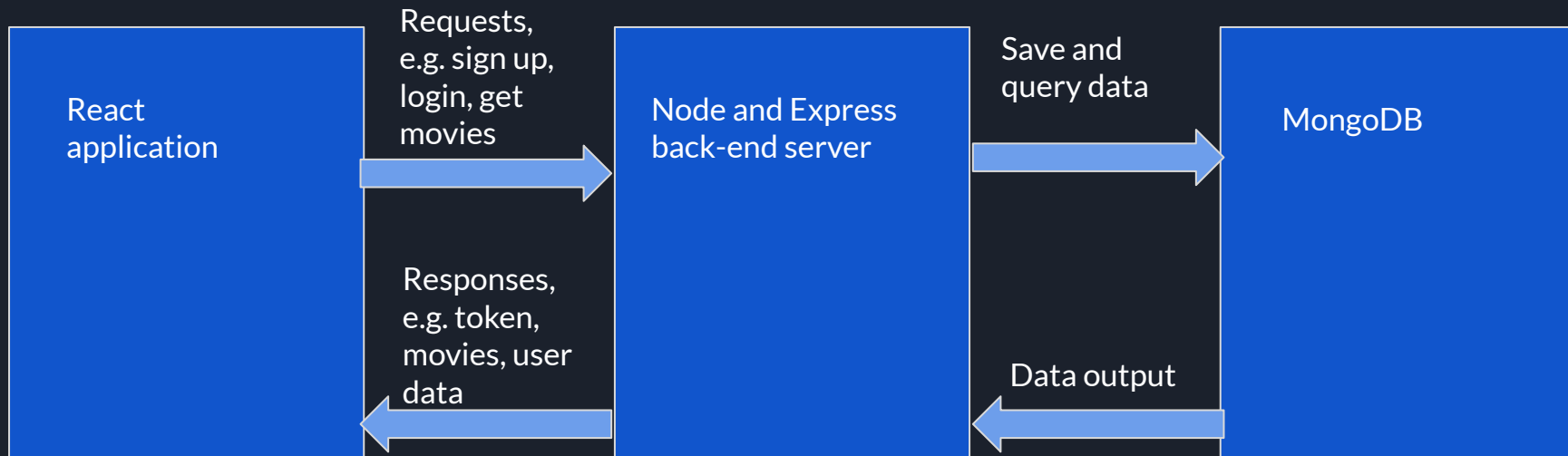


Purpose

MyFlix was a personal project built as a part of my full-stack web development program at CareerFoundry. In this project, I learned to build the back-end server with Node and Express, MongoDB as the database and build applications with React.

Objective

The objective of this project is to build a fully function application for storing data into MongoDB, React application fetches data through back-end server built with Node and Express.



Back-end server

Server is the connection between React application and MongoDB. It is built by Node and Express, with REST principle. User authorization is implemented in the server using JWT. The server is host on Heroku.

Repository: <https://github.com/CoderMCH/movie-app>

API documentation: <https://mch-flux-app-813b2fce5e48.herokuapp.com/documentation.html>

Tech:

1. Node and Express for building server
2. JWT for authorization
3. MongoDB and Mongoose for database
4. Heroku for hosting server

This is document of movie app api.

Description	Method	URL	Body	Result example
Get the list of all movies	Get	/movies	-	<pre>{ "genre": { "type": "horror", "description": "Horror is..." }, "director": { "name": "Hideo Nakata", "bio": "Hideo Nakata is...", "birthday": "1961-07-19T04:00:00.000Z", "deathDate": null }, "id": "5bce7f05c8da9238fe21", "title": "Ring", "description": "Ring...", "imagePath": "https://en.wikipedia.org/wiki/Ring_(film)#/media/File:Rings (1998).jpg", "featured": false, "actors": [] }</pre>
Get the specific movie by title	Get	/movie/title	-	<pre>{ "genre": { "type": "horror", "description": "Horror is..." }, "director": { "name": "Hideo Nakata",</pre>

Front-end application

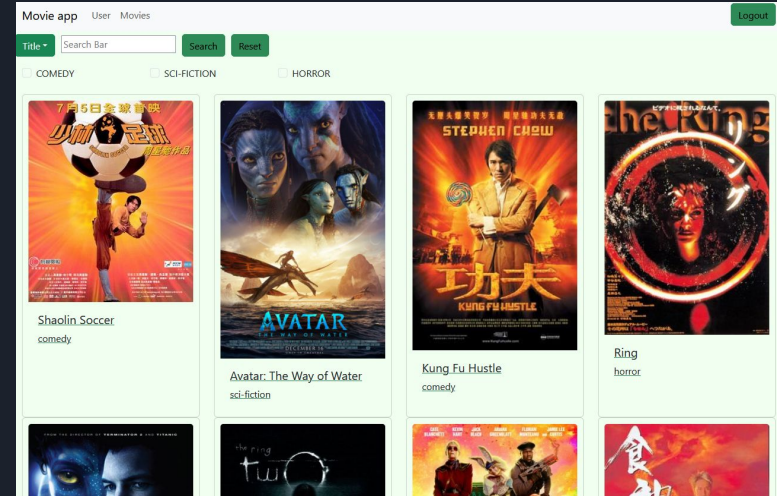
The application is build by React. Users are able to register account and login to view the movies in database. Users need to register an account for viewing movies and saving them into favorite movie list. The application is host on Netlify.

Repository: <https://github.com/CoderMCH/movie-app-client>

Link: <https://main--mch-flix.netlify.app/>

Tech:

1. Redux for data binding
2. Router for single page application routing
3. Bootstrap for styling
4. Netlify for hosting web application





Outcomes and learnings

Server side:

1. Learn Node, Express, HTTP methods and REST
2. Learn and implement JWT authorization method
3. Learn MongoDB and use Mongoose for communication between server and database

Client side:

1. Learn React Redux for data binding to UI elements
2. Learn React Router for single page web application
3. Learn React Bootstrap for UI/UX components



Future enchantments

1. Share movies to social media, including Facebook, X, Instagram
2. Login with other media account, including Google, Facebook, X, Instagram
3. Recommendation system for better user experience improvement
4. Advanced security system such as two factor authentication