

# FULL STACK CASE STUDY



Developer: Stephanie Duda



# OVERVIEW

FlixList is a responsive single-page web application designed for movie enthusiasts, offering a rich and interactive user experience.

Developed as part of a full-stack JavaScript project, FlixList seamlessly integrates with an existing server-side codebase, providing users with a platform to explore, save, and learn more about their favorite movies.

[VISIT SITE](#)

Username: User 1

Password: password1





# PURPOSE

FlixList was conceived as a personal project during my web development course at CareerFoundry. The goal was to demonstrate proficiency in full-stack JavaScript development. The project involved building the complete server-side and client-side for the application from scratch, as outlined in the project brief for the API.

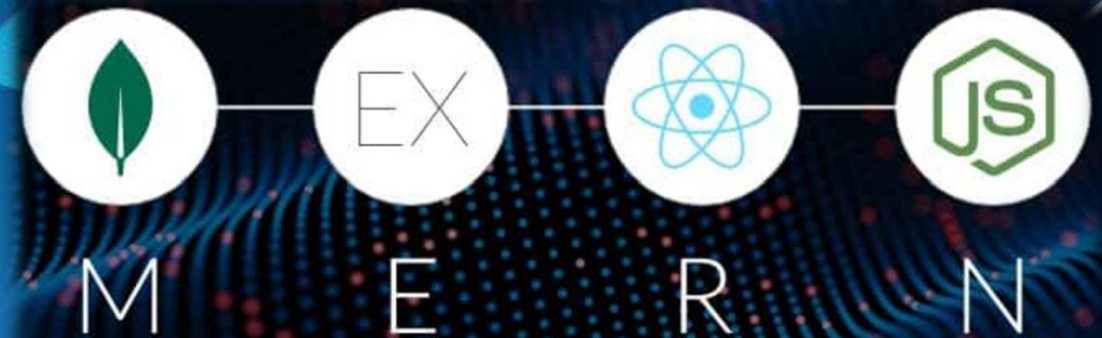
**CAREERFOUNDRY**  
**Launch Your Life-Changing  
Career in Tech**  
Learn in-demand skills and how to use AI effectively and ethically—all online, flexibly-paced, with no need to quit your job. Go from beginner to job-ready tech professional in as little as five months!

**API**



# OBJECTIVE

The primary aim was to create a feature-rich web app that caters to movie enthusiasts, allowing them to access detailed information about movies, directors, and genres. The comprehensive tech stack, known as the MERN stack, was employed to achieve this, emphasizing the importance of both server-side and client-side development.



# APPROACH

## Server-Side

I initiated the project by developing a RESTful API using Node.js and Express. The API interacts with a MongoDB database, employing CRUD methods for data retrieval and storage. To ensure security, I implemented user authentication and authorization through basic HTTP authentication and JWT.

**ENDPOINTS HERE**  
(documentation.html)

**BUSINESS LOGIC:** I structured the server-side code to handle the essential functions of FlixList. This involved setting up models to keep the data consistent. I used MongoDB, a flexible database, to store information.

**DEPLOYMENT:** For hosting FlixList, I chose MongoDB because it provides more flexibility. I deployed the app on Heroku and stored the data in MongoDB Atlas, ensuring a reliable and scalable setup.

### **SECURITY:**

Login Security: HTTP authentication

Authorization: JWT tokens

CORS Setup: Applied CORS to prevent security issues related to cross-origin requests.

Password Protection: Implemented strong password hashing for user account security.

**TESTING AND DEPLOYMENT:** Tested the app's API using Postman to make sure it works smoothly and handled data accurately. I then deployed the app on Heroku for online access. MongoDB Atlas hosted the database, ensuring data reliability



```
{
  _id: ObjectId("64ffelaaff16555747ef1f1"),
  Title: 'Captain Marvel',
  Description: 'Carol Danvers becomes one of the universes most powerful heroes when Earth is caught in the middle of a galactic war between two alien races.',
  Genre: {
    Name: 'Action',
    Description: 'Movies in the action genre are fast-paced and include a lot of action like fight scenes, chase scenes, and slow-motion shots. They can feature superheroes, martial arts, or exciting stunts. These high-octane films are more about the execution of the plot rather than the plot itself.'
  },
  Director: {
    Name: 'Anna Boden',
    Bio: 'Anna Boden is an American film director, cinematographer, editor, and screenwriter best known as the co-writer of the 2019 film Half Nelson. She is known for her collaborations with fellow filmmaker Ryan Fleck. While studying film at NYU, Boden met Ryan Fleck on the set of a student film. Soon they began dating and decided to collaborate. Together they made the short documentaries Half You Seen This Man? and Young Rebels before they wrote and he directed the short film Gowanus, Brooklyn, a sample feature aiming to attract potential financiers to their undeveloped script, Half Nelson.'
    Birth: '1979',
    Death: 'null'
  },
  imagePath: '#',
}
```

1	1	Robin Hood: Prince of Thieves	Robin Hood decides to fight back as an outlaw when faced with the tyranny of the
2	2	Hook	When Captain James Hook kidnaps his children, an adult Peter Pan must return
3	3	Legend	A young man must stop the Lord of Darkness from destroying daylight and mar
4	4	Beetlejuice	The spirits of a deceased couple are harassed by an unbearable family that has
5	5	Inglourious Basterds	In Nazi-occupied France during World War II, a plan to assassinate Nazi leaders
6	6	Captain Marvel	Carol Danvers becomes one of the universe's most powerful heroes when Earth
7	7	Pitch Perfect	Beca, a freshman at Barden University, is cajoled into joining The Bellas, her sch
8	9	Harry Potter and the Prisoner of Azkaban	Harry Potter, Ron and Hermione return to Hogwarts School of Witchcraft and W
9	10	The Fifth Element	In the colorful future, a cab driver unwittingly becomes the central figure in the s
10		Robin Hood: Prince of Thieves	In 1938, after his father goes missing while pursuing the

# Server-Side

```
body Cook

Pretty Raw Preview Visuanze

1 [
2 {
3   "Title": "Robin Hood: Prince of Thieves",
4   "Description": "Robin Hood decides to fight back as an outlaw when faced with the tyranny of the
5     Sheriff of Nottingham and also to win back the love of the beautiful Maid Marian.",
6   "Director": "Kevin Reynolds",
7   "Genre": "Action"
8 },
9 {
10  "Title": "Hook",
    "Description": "When Captain James Hook kidnaps his children, an adult Peter Pan must return to
      Neverland and reclaim his youthful spirit in order to challenge his old enemy.",
```



# APPROACH

## Client -Side

Moving on to the client-side, I employed React and React-Redux to craft a user-friendly single-page application. This includes various views such as the main view, single movie view, login view, registration view, and profile view, each contributing to an enhanced user experience.

[VISIT SITE](#)

Username: User 1  
Password: password1

### The Build

My aim was to grasp the MVC architecture design pattern. I opted for React due to its speed and ease of maintenance. To handle build operations, I employed Parcel.

### Components

After the initial build, I created components for different views and used hooks to manage the application state. Axios was used to fetch data from the previously developed API.

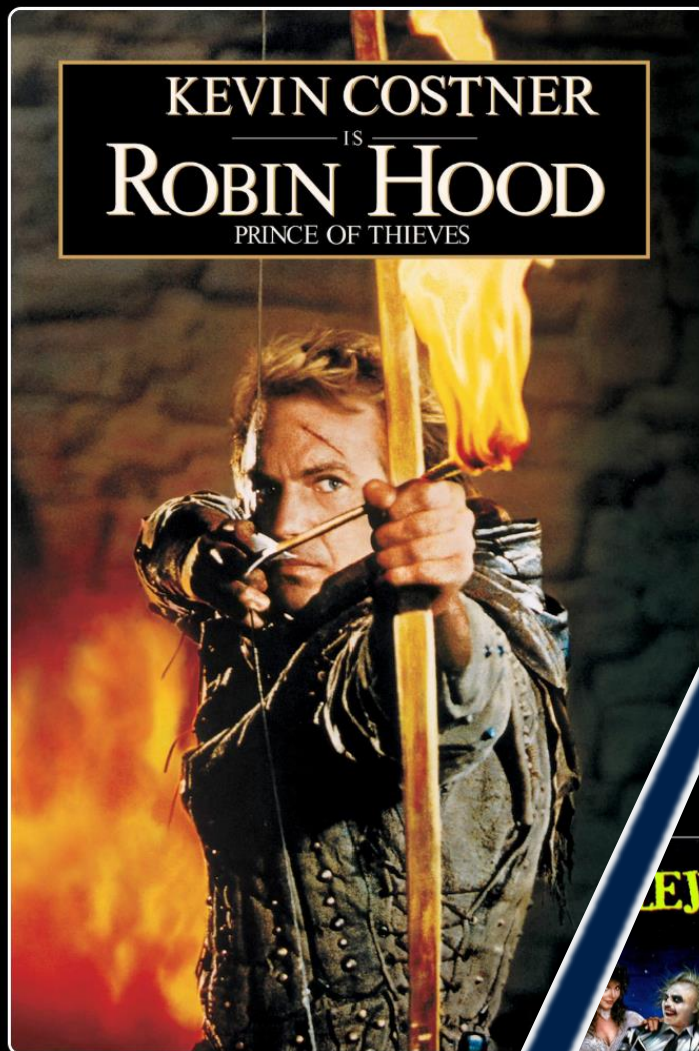
### Design

To ensure a consistent and appealing layout, React Bootstrap came in handy for designing pages and cards. App-routing on the client-side added an authentication layer for accessing views.

### Final Product

To enhance state management and scalability, Redux was integrated into the application. The completed project found its home on Netlify, making it accessible online.

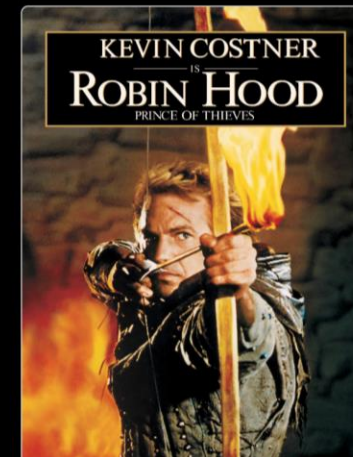




**Title:** Robin Hood: Prince of Thieves

**Description:** Robin Hood decides to fight back as a  
the Sheriff of Nottingham and also to win back the l

**Director:** Kevin Reynolds



Welcome to  
**myFlix App**

Login

[New User? Sign Up](#)

Filter by Genre:

All Genres



# CHALLENGES

While constructing the API was a somewhat smooth process, navigating React and Redux on the client-side presented significant challenges. Troubleshooting with my tutor and mentor was invaluable in overcoming these hurdles. This was both my favorite and most challenging project. Deploying the project on Netlify added another layer of complexity, but through collaborative learning and perseverance, I successfully conquered these challenges.

# DURATION

The development of the client-side component took longer than expected, reflecting the time invested in understanding React and React-Redux. My issues with deployment also contributed to the extended duration.

# TECHNOLOGIES

## Server-Side Development

- Node.js
- Express
- Mongo DB
- MERN stack

## Client-Side Development

- React
- Bootstrap

## Additional Technologies

- Postman
- Heroku
- Axios
- Redux
- JWT Authentication

# CREDITS

•**Lead Developer:** Stephanie Duda

•**Tutor:** Andrew Muscara

•**Mentor:** Sarah Rohl