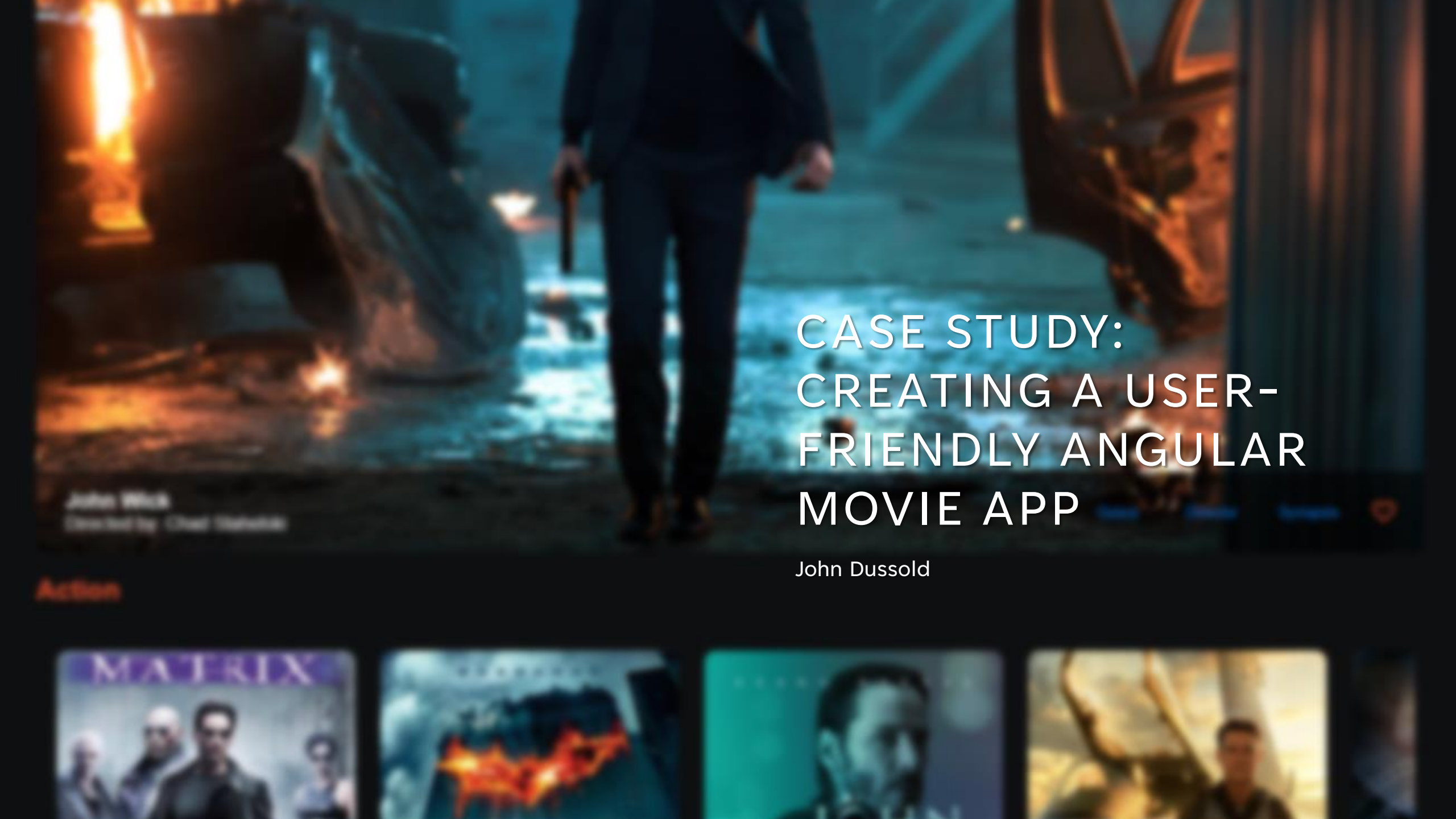


CASE STUDY: CREATING A USER-FRIENDLY ANGULAR MOVIE APP

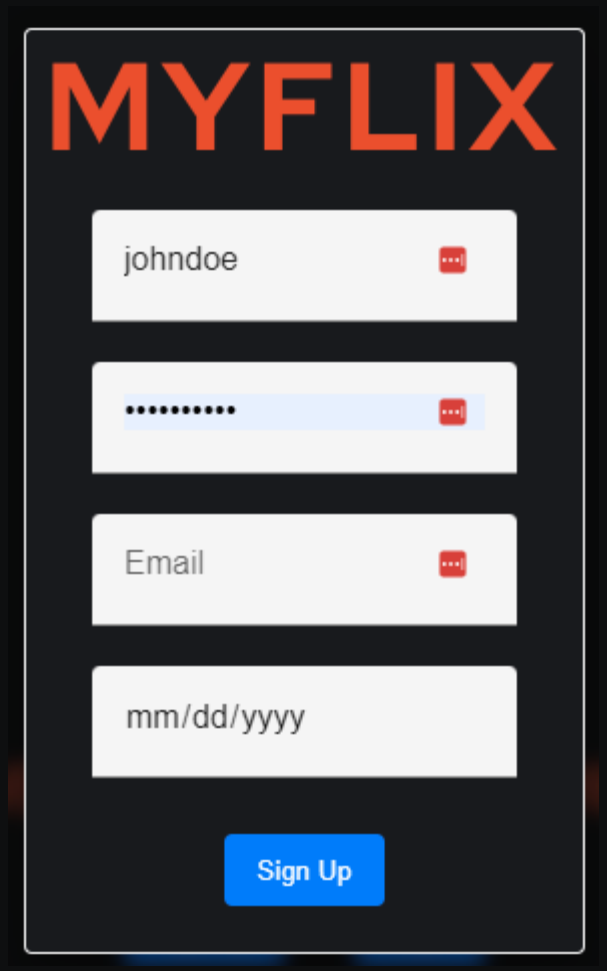
John Dussold

OVERVIEW

This case study details the creation of a user-friendly movie information application using Angular and a custom-built server-side API. The project's main objective was to design an intuitive interface that allows users to access movie details, manage their favorite selections, and update profile information.

PURPOSE & CONTEXT

The purpose of this project was to build a responsive and engaging movie application that allows users to sign up, log in, view movies, manage favorites, and edit their profiles. The application was designed to provide a user-friendly interface, enhancing the accessibility of movie content while demonstrating Angular's capabilities for creating dynamic and interactive web applications.

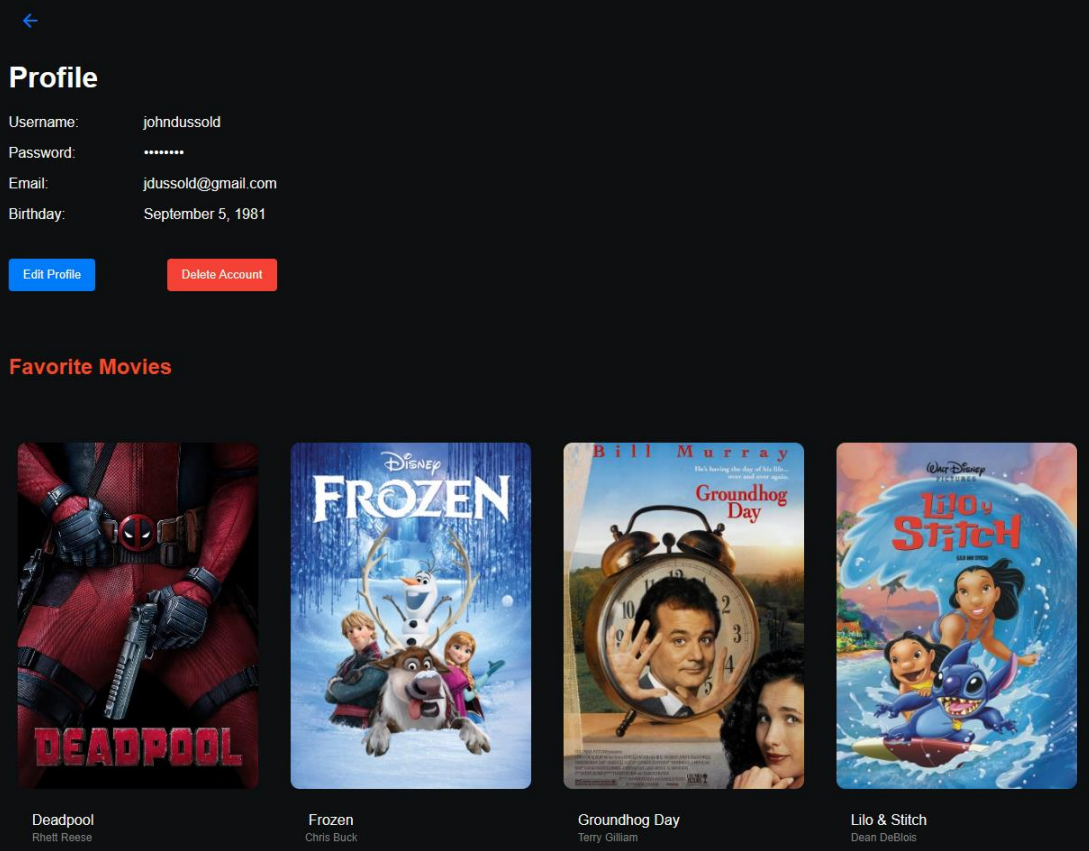


OBJECTIVE

The central objective was to create a client application using Angular that effectively communicates with the MyFlix API, a custom server-side solution. The application needed to offer users an engaging experience for accessing movie information and personalizing their interactions.

SERVER-SIDE

The server-side component involved the creation of the MyFlix API, which was developed using Node.js and Express.js for routing. The API was responsible for handling user authentication, querying the MongoDB database for movie data, managing user profiles, and handling favorite selections.



CLIENT SIDE

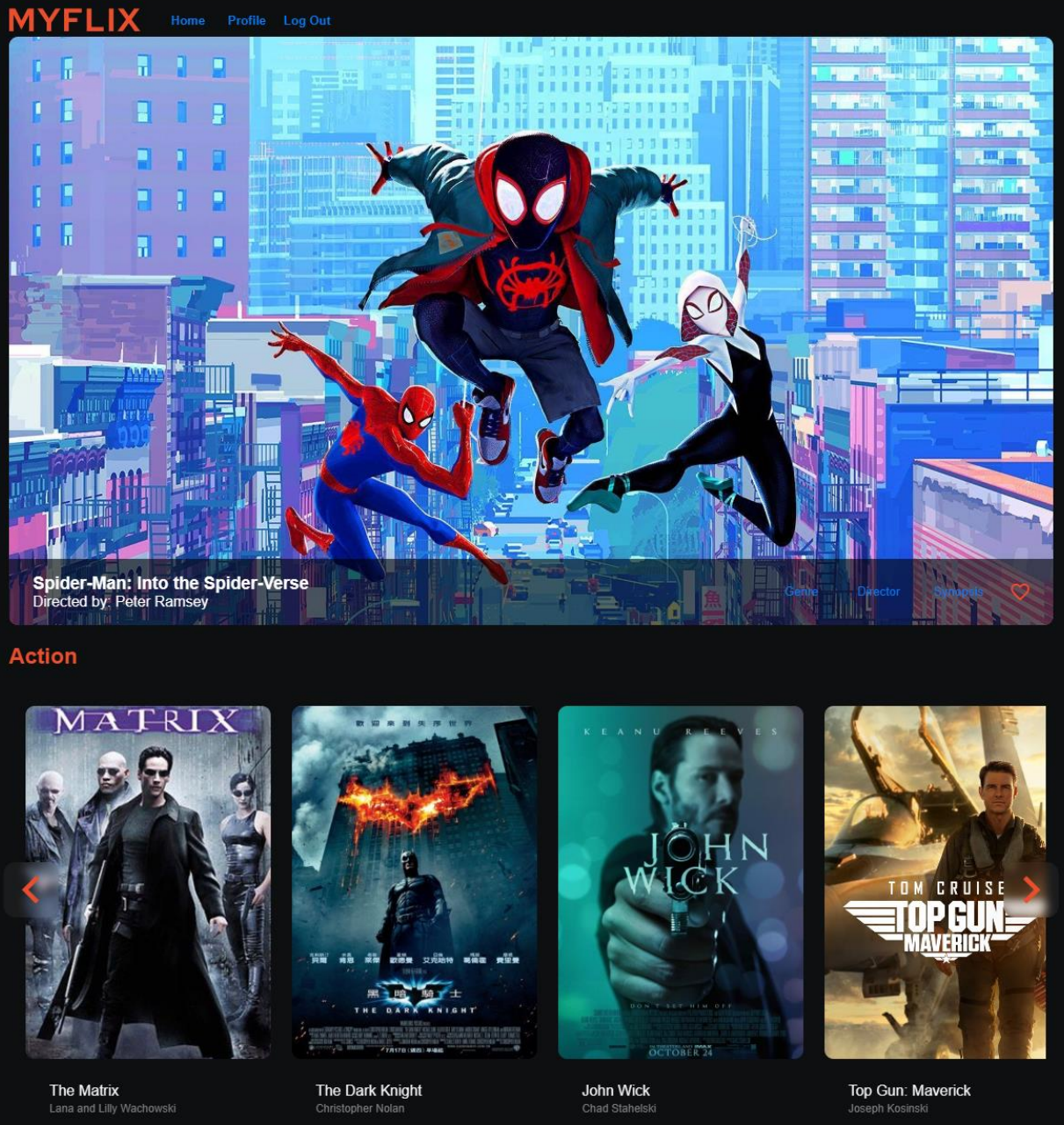
The client-side application was developed using Angular CLI version 15.2.5. It connected with the MyFlix API to deliver a dynamic and interactive movie information experience. The client-side components included:

- welcome-page: A landing page built using Angular Material components. Users could sign up or log in to access the main features of the application.
- user-registration-form: A registration form utilizing Angular Reactive Forms for creating user accounts, with fields for username, password, email, and birthday.
- user-login-form: A login form for user authentication, featuring fields for username and password.
- movie-card: Displayed movie cards on the homepage using Angular Material cards, showing key details like movie title, director, and images.
- movie-dialog: Employed Angular Material's MatDialog to present genre-specific movie information, such as genre, director, and synopsis.
- profile-page: Enabled users to edit profile information, manage favorite movies, and delete their account using Angular Reactive Forms.
- confirm-password-dialog: Provided a layer of security for sensitive actions, validating user passwords.

CHALLENGES

Several challenges were encountered during the project's development:

- Integration with Custom API: Coordinating communication between the Angular client and the custom MyFlix API required careful API design and data handling.
- Responsive UI: Designing a responsive interface that functions well across devices demanded skillful use of Angular Material components and responsive design techniques.
- Complex Forms: Implementing form validation and managing form states for registration, login, and profile editing forms necessitated in-depth knowledge of Angular Reactive Forms.
- User Authentication: Ensuring secure user authentication across the application while protecting user data and interactions posed a significant challenge. To overcome this challenge, a multi-layered authentication strategy was adopted. Passwords were securely hashed and stored in the database, protecting sensitive user information. JSON Web Tokens (JWT) were employed for session management, providing secure user authentication without relying on traditional session cookies.



DURATION

The project spanned approximately 4 weeks, covering various phases such as planning, design, development, testing, and documentation.

TECHNOLOGIES & METHODOLOGIES USED

- Front-End: Angular, Angular Material components, Angular Reactive Forms, CSS
- Back-End: Node.js, Express.js, MongoDB
- API Communication: HTTP requests, FetchAPIDataService
- User Authentication: Secure token-based authentication
- UI Design: Responsive design principles, Angular Material design
- Methodologies: Agile development, Test-driven development (TDD)

CREDITS

- Role: Lead Developer
- Tutor: Jackson Lawrence
- Mentor: Sophek Tounn

FINAL THOUGHTS

The project successfully accomplished it’s initial goal and demonstrated the development of an Angular-based movie information application that empowers users to access movie details, manage favorites, and update their profiles. Through integration with a custom API, the project accomplished its goal of providing users with an interactive movie information experience. Challenges were met with effective design and implementation, resulting in a functional and user-centric application.

FUTURE STEPS

- Additional Features: I’m considering adding features like an advanced search, recommendations based on user preferences, and social sharing of favorite movies.
- Localization: Integrate localization to make the application accessible to users from various regions by providing content in different languages.
- Performance Optimization: Continuously optimize API requests, database queries, and application performance to ensure fast loading times and responsiveness.
- Testing & Quality Assurance: Expand automated testing coverage, conduct thorough testing on different devices and browsers, and implement continuous integration and delivery practices.