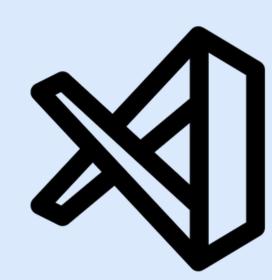# MyMovies-Angular App: a case study
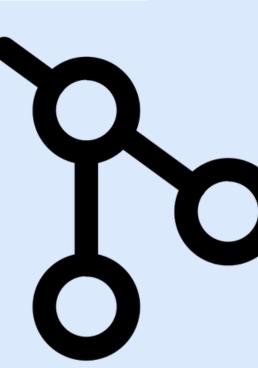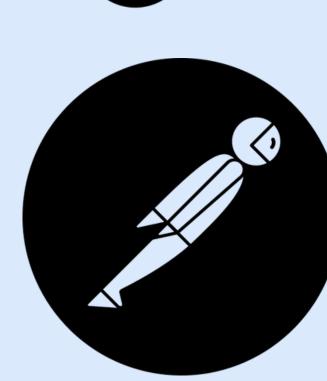
Planning

Development
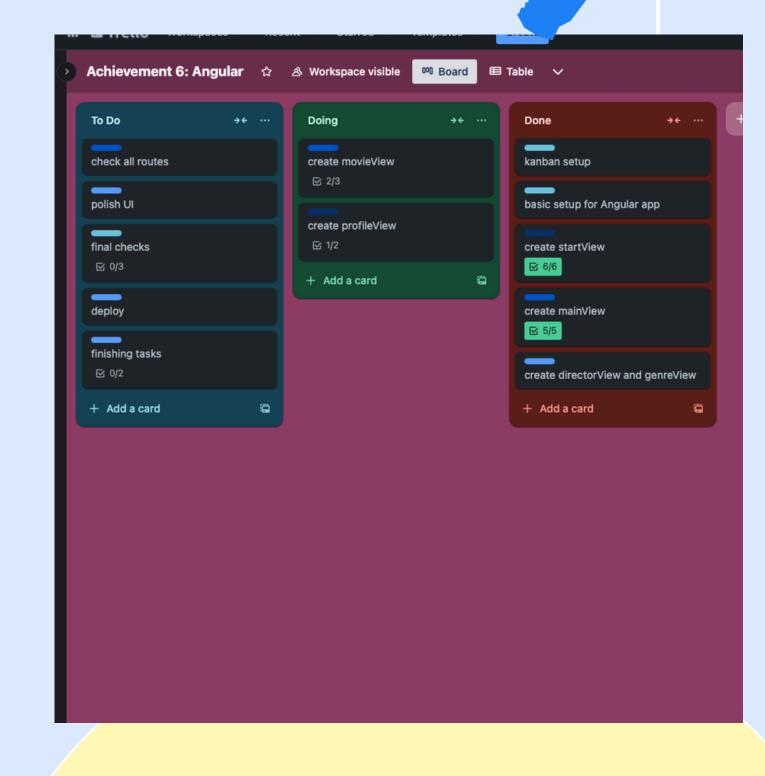
Reflection

Objective:
The primary objective of this project was to explore Angular by rebuilding the frontend of the MyMovies-Server, a custom-built RESTful API from an earlier phase of my coding journey.

Structure:
As this was my first time using Angular CLI, I found "ng generate component" to be a very powerful tool. Using this to generate components, and chatGPT to generate some of the logic made building this app extremely efficient and fast. By utilizing these powerful dev tools, I could focus on making adjustments so it met project requirements, successfully connected to the API, and provided an intuitive and appealing UI for users.

Conclusion:
Rebuilding the MyMovies frontend with Angular was a significant milestone in my coding journey. It exposed me to another prevalent frontend framework and provided an opportunity to enhance my problem-solving skills and Agile workflow methods. Moving forward, I plan to add more movies to the API to make the app more useful for users. This project not only broadened my technical skills but also reinforced the importance of adaptability and continuous learning in software development.

Design:
Using Angular Material to design this app was a delight. I really enjoyed creating a consistent, intuitive layout that mimics other popular movie apps where users scroll through movies horizontally. Crafting visually appealing movie cards and dialogs, along with implementing a fixed navbar and footer for easier navigation access, was particularly satisfying.

Conceptualization:
This MyMovies project was initially developed using React for the frontend. Out of a desire to expand my expertise and compare two major fr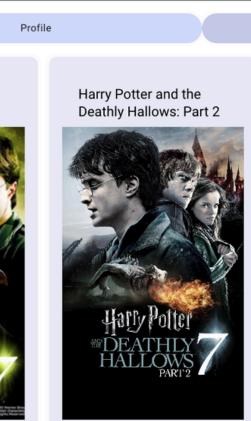ontend frameworks, I rebuilt the frontend using Angular. This new frontend would be integrated with the RESTful API I had built for the original app (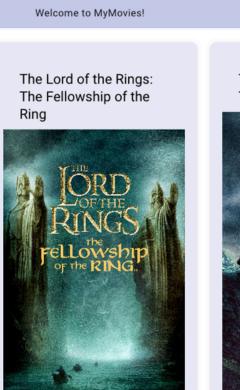using MongoDB Atlas), and provide equal functionality. I also wanted this Angular app to be a SPA (Single Page App) with horizontal scrolling for users to have a similar experience to other popular movie apps. At this point, I used Trello to map out my Kanban board to maximize efficiency and guarantee a thorough build process.
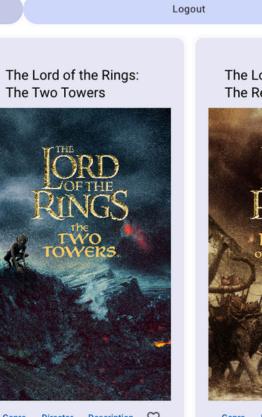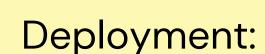
Integration:
This is where most of the challenges lay in this project. One particularly tricky issue was the movie details dialog route, which didn't work as expected. By referencing the other working routes and by troubleshooting with chatGPT, I was able to get this route working correctly. It also took a fair amount of troubleshooting and console.log additions to find the correct objects to pass for functions such as addToFavorites and removeFromFavorites.
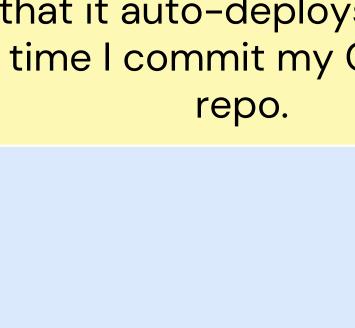
Deployment:
Once again, I chose to deploy using Vercel because it is awesome and I absolutely love that it auto-deploys every time I commit my GitHub repo.