

# myFlix Angular Application

01

## Overview

This client-side application, built with Angular, allows registered users to view a catalog of movies and manage their personal profile information. The application communicates with a RESTful API and connected database to retrieve movie data and persist user-specific updates, providing a seamless and dynamic user experience.

02

## Purpose

This movie application was developed as a personal project during my Full Stack Development program with CareerFoundry, designed to use the RESTful API that I created in a previous project and develop an understanding of Angular's framework to build out a client-side application.

## Technologies

\* Angular

\* Typescript

\* Angular Material

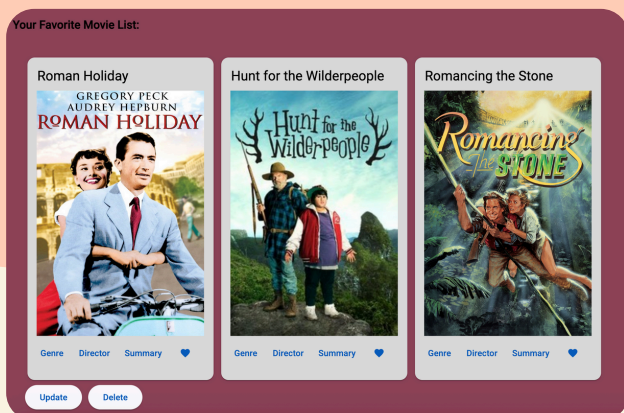
03

## Objective

The objective of this project was to learn the basics of Angular through developing a client side application using different assets of the Angular Library such as Angular Forms and Angular Material.

## Project Developer

I served as the primary developer for this project, applying key concepts provided through CareerFoundry's curriculum.



# The Meet Application

Continued

04

## Angular with Typescript

I've come to understand the many advantages of using TypeScript, particularly its ability to catch errors during compilation rather than at runtime—a major benefit for maintaining reliable code. Transitioning from React with JavaScript, however, initially made TypeScript feel more rigid, as it requires explicit type definitions and a clear understanding of variable structures. Despite the steeper learning curve, I recognize how valuable TypeScript is in modern development. Its strong typing system not only improves code quality but also aligns with industry best practices, as many companies have adopted it to reduce errors and enhance maintainability.

05

## Angular Architecture

I really appreciated the structure of Angular's architecture, which allows developers to clearly define and organize components throughout an application. Early on, I realized the importance of having a solid organizational plan to ensure that components remained modular and easy to manage—without becoming tangled. I also enjoyed working with Angular's data binding features, which enable seamless communication between components and create a dynamic, interactive user experience.

One key architectural decision involved handling API endpoints from a backend I had previously developed. Since there wasn't a dedicated endpoint for a user's favorite movies, I retrieved all movies from the API and then filtered them on the client side. While this approach worked for a smaller application, I noted that adding a dedicated "favorites" endpoint would improve efficiency and reduce load times in future iterations. Additionally, I appreciated that Angular communicates directly with the server rather than relying on a virtual DOM. For this project, that direct interaction contributed to smooth performance and responsive updates, reinforcing my overall positive experience with Angular's architecture and design principles.



# The Meet Application

Continued

06

## Angular Forms

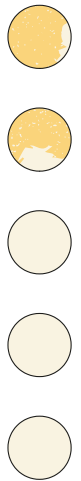

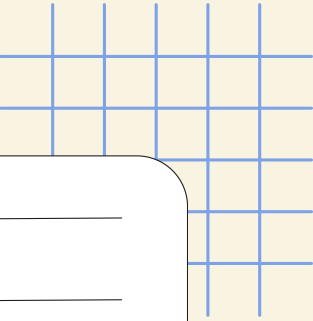
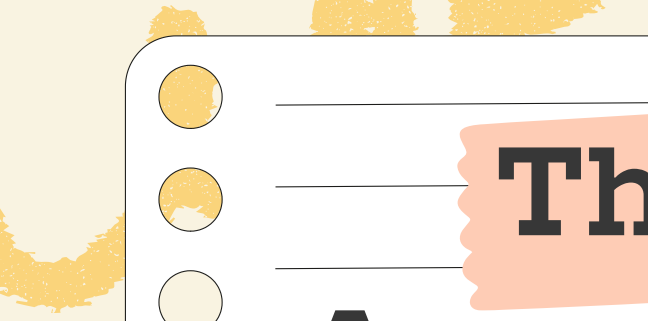

Angular offers two primary approaches to building forms: template-driven and reactive. Template-driven forms are defined within the template, while reactive forms are managed within the component's model, making them more powerful, scalable, and easier to maintain. These Angular forms are part of Angular's framework and can be implemented efficiently with minimal code once imported.

In this application, I used Angular forms to handle both user sign-up and login functionality. The curriculum provided guidance for building the sign-up form, and I applied the same principles to develop the login form independently. Overall, I found Angular's form modules intuitive and efficient to work with. The built-in features and add-ons provided by Angular significantly enhance development productivity and contribute to its strength as a comprehensive front-end framework.

07

## Angular Material

I found this Angular Material library to be highly intuitive and well-documented, which made the implementation process smooth and enjoyable. Given its robust set of features and flexibility, I would have liked to spend even more time exploring its full capabilities. While some of the naming conventions for components were occasionally confusing or not entirely reflective of their functionality, the overall responsiveness and performance of the elements were excellent. This has been one of my favorite design libraries to work with, and it provided a great balance between ease of use and creative flexibility.



# The Meet Application



Final Thoughts



08

## Overall





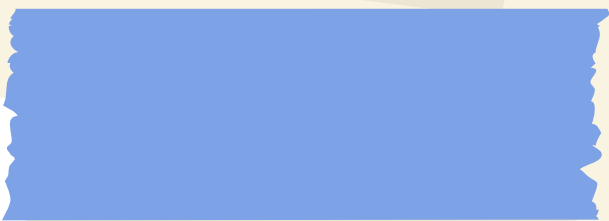

Overall, I really enjoyed working with Angular. Compared to React, I found its structure to be more clearly defined and easier to organize. I especially appreciated that components in Angular can communicate bidirectionally, rather than being limited to parent-to-child data flow. This project also gave me valuable experience in retrieving and displaying data from a database, as well as implementing user customization features to enhance the overall experience.



09

## Learning

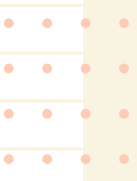
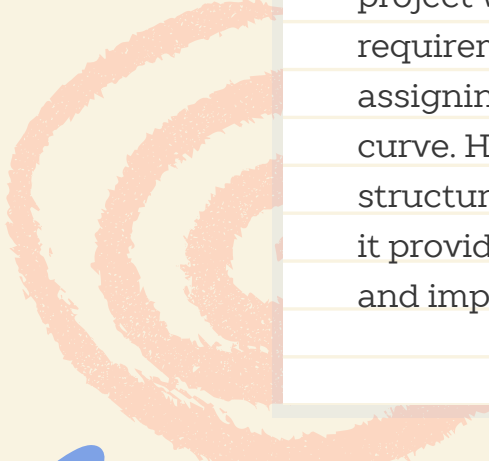

Utilizing Angular Material proved to be a valuable and insightful experience. Once I became familiar with the framework, I was impressed by how intuitive and versatile it was, offering a wide range of components that were easy to implement and customize. I particularly appreciated the design process aspect—seeing visual results come together quickly was both motivating and creatively fulfilling. Overall, Angular Material enhanced the development experience by streamlining the UI design process while maintaining a polished, cohesive look across the application.



10

## Challenges

One of the main challenges I encountered during this project was working with TypeScript. Adapting to its requirement of defining variable types before assigning values presented a significant learning curve. However, as I became more familiar with its structure and purpose, I came to appreciate the value it provides in preventing errors early in development and improving overall code reliability.



## The Future

In the future, I would like to expand this project by adding more movies to the database, giving users a broader selection to explore. I especially enjoyed working with Angular Material and would love to continue refining the design elements of this application. Enhancing the navigation bar and experimenting with additional layout and interaction techniques would also provide valuable experience in user interface design and improve overall usability.

