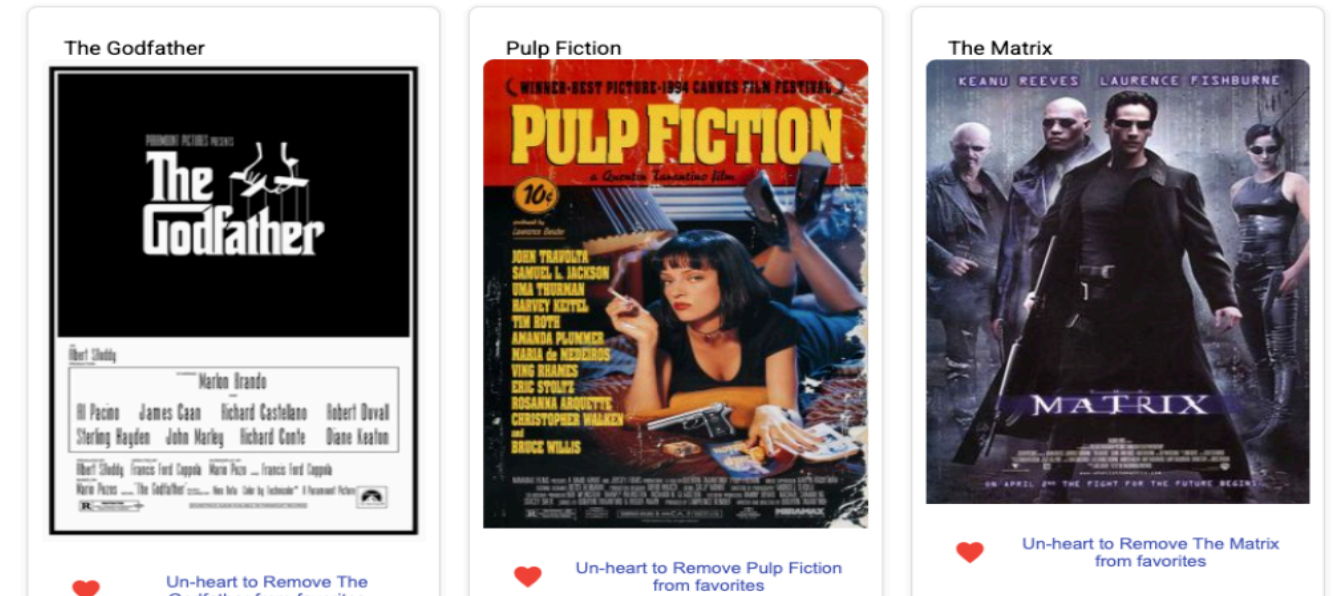


# Angular MyFlix Case Study

## Favorite Movies



## ExecutiveSummary

*A responsive single-page movie app that helps users browse, explore, and save their favorite films.*

### Introduction

Angular MyFlix is a movie app I built using Angular and TypeScript. It lets users view detailed movie information, explore genres and directors, manage their profile, and save favorites. The app connects to a secure REST API and provides a fast, user-friendly experience on both desktop and mobile screens.

## **The Problem**

Movie databases can feel overwhelming or outdated, with hard-to-read layouts or missing features. I wanted to create a streamlined, visually clear app that allows users to explore content and personalize their experience without feeling lost or frustrated.

## **My Role**

I built the entire frontend application using Angular. I connected it to the MyFlix API, styled the interface using Angular Material, and made sure features like login, movie filters, profile editing, and favorite management worked smoothly across all devices.

## **Timeline**

This project took about 3 weeks from setup through testing and deployment.

## **Tools and Skills Used**

Frameworks and Tools:

- Angular
- TypeScript

UI Library:

- Angular Material

Backend Integration:

- REST API using Node.js, Express, and MongoDB
- JWT Authentication for secure login sessions

Other Tools:

- RxJS for handling data streams
- TypeDoc for documentation

Skills I practiced:

Component-based frontend development, secure user authentication, API integration, profile and data management, responsive design



# Key findings

## Key Finding 1:

Users appreciated being able to filter by genre and director to explore content that matched their interests

## Key Finding 2:

Profile editing and favorite movie management helped users feel more connected to the app

## Key Finding 3:

A clean layout using Angular Material made the app easier to navigate, especially on smaller screens

# Solutions

## Solution 1

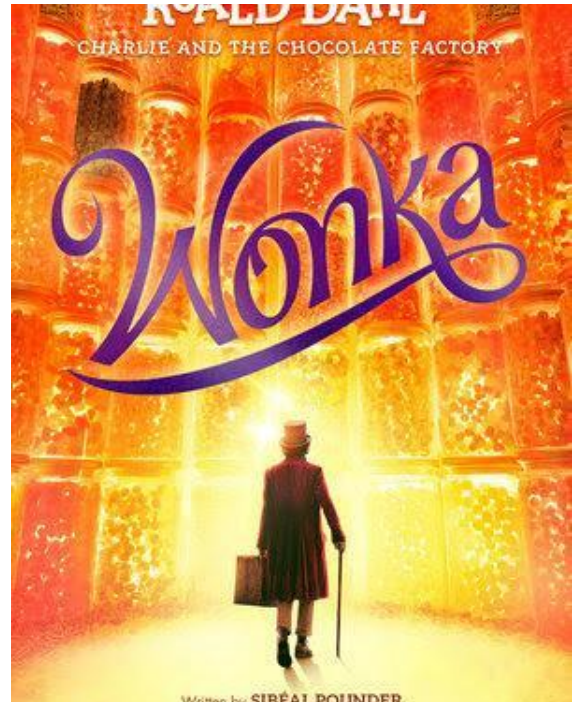
Used Angular Material to build a responsive and clean layout that works well on all devices

## Solution 2

Connected the Angular frontend to my custom-built API with secure login and JWT-based authentication

## Solution 3

Created dedicated views for genres, directors, and profiles to give users deeper ways to explore and personalize their experience



MyFlix Movies

Profile

Log Out

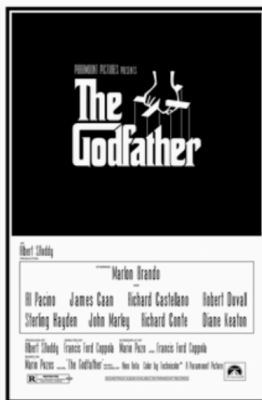
Q movies, directors genres



The Shawshank Redemption

Directed by: Frank Darabont  
Genre: Drama

Genre Director Description



The Godfather

Directed by: Francis Ford Coppola  
Genre: Gangster Film

Genre Director Description



# Impact

## Impact 1

The app helped users browse and save movie content in a simple and organized way

## Impact 2

Profile management and the ability to update personal data gave users more ownership

## Impact 3

I gained confidence in using Angular, TypeScript, and working with reusable components connected to a live API

# Case Study Introduction

## Background

This project was part of my full-stack development training. I created the Angular version of MyFlix to practice working with TypeScript, Angular components, and API integration.

It allowed me to build a fully functional movie app using a structured frontend framework that supports scalability and reusable components.

The app uses Angular Material for a clean look and connects to the custom-built MyFlix API.

## Objectives

### **Objective 1**

Build a movie browsing app using Angular with a polished, responsive interface

### **Objective 2**

Let users explore details about movies, genres, and directors

### **Objective 3**

Allow users to register, log in securely, and manage their profiles

### **Objective 4**

Enable users to add or remove movies from their list of favorites

# Problem

Movie databases can feel overwhelming or outdated, with hard-to-read layouts or missing features.

I wanted to create a streamlined, visually clear app that allows users to explore content and personalize their experience without feeling lost or frustrated.

# Recommendations

- Recommendation 1**  
Use a frontend framework like Angular to manage state and routing cleanly across views
- Recommendation 2**  
Give users clear access to profile tools so they can personalize their experience
- Recommendation 3**  
Design a mobile-first layout that makes the app accessible and intuitive regardless of screen size

# Impact of recommendation

SOLUTION	EFFECTIVENESS	IMPACT	NOTES
Angular Material layout	Highly effective ▾	Improved readability and structure across devices	Clean and mobile-friendly
Profile and favorites	Highly effective ▾	Created a more personal experience	Users could update info and curate content

SOLUTION	EFFECTIVENESS	IMPACT	NOTES
Modular routing system	Moderately effective ▾	Made navigation smoother	Helped keep user experience focused

# Analysis



## Research methods

- Studied other movie apps to find strengths and weaknesses in layout and functionality
- Tested the app on multiple screen sizes to ensure mobile responsiveness
- Gathered feedback from mentors and testers about the flow and appearance of features

## Approaches used

- Component-based architecture using Angular
- Secure connection to REST API with token authentication
- Structured UI layout with Angular Material



# Relevant facts and information

- Duration: 3 weeks
- Tools: Angular, TypeScript, Angular Material, Node.js, Express, MongoDB
- Features: Secure login, movie browsing, profile editing, favorite movie tracking

## Challenges

Setting up secure JWT authentication and connecting the frontend with the API required careful testing.

Another challenge was organizing the different views (movies, genres, directors, profile) in a way that felt simple and intuitive to users.

Making sure the app looked consistent across devices also took several design adjustments.

## Conclusion



## Summary of findings

This project gave users a smooth, reliable way to explore and organize their favorite movies. It provided a structure that worked well across devices and gave users full control of their own profiles and preferences.



# Implications of the study

This app helped me become more confident with Angular, and taught me how to manage data, design interactive views, and build secure connections to a backend. It reinforced the value of simple navigation, clean design, and giving users flexibility.

**“Building the Angular version of MyFlix helped me strengthen my frontend development skills while creating a clean, personal movie browsing experience for users.”**