Frontend Mentor Exercise: Movees Catalog

# Description

A basic catalog site in the style of IMDB that includes search filtering, poster images, and a details page containing more information on a selected movie.

# Mockups

* Home page mockup: [Home Page Mockup](https://github.com/jrusc257/frontend-mentor-movees/blob/main/dev-resources/Mockups/Movees-Mockup---home.png)
* Details page mockup: [Details Page Mockup](https://github.com/jrusc257/frontend-mentor-movees/blob/main/dev-resources/Mockups/Movees-Mockup---details.png)

# Key Concepts

* Angular CLI
* Angular Routing
* Angular Components
* Angular Pipes

# Setup

For this project you will need to set up Node.js and install the Angular CLI, instructions for which you can find here: <https://angular.io/guide/setup-local>

# Requirements

Over the course of this project you will develop a basic Angular site to display a collection of movie details from a JSON file, and enable the user to click on a poster image to see more information on a movie details page. This site does NOT need to be responsive, instead focus on the Angular functionality.

* Generate a new Angular app on your local machine
* Generate a component to serve as the home page.
* Generate a component to serve as the details page.
* Set up two routes for the home page component and details page component respectively.
* Import the JSON file to use as our faux “database”.
* Filter the list of movies on the homepage based on the “search” input.
  + You only need to filter based on matching characters in the title of the movie
  + This challenge is a little more open-ended, so it’s up to you to decide how to address the problem.
* Set up the poster thumbnail to link to the details page.
  + Pass a “movieID” to the details page component (through the router as a route param) to determine which movie details to display.

# Stretch Goals

* Add your own movie, complete with details, to the JSON file.
* Add two radio buttons next to “search” to sort by title (alphabetical) or sort by score (highest first).
* Load the JSON file asynchronously.
* Create a “suggest!” button to take the user to a random movie details page
* Show either a “fresh” or “rotten” icon next to the Rotten Tomatoes score depending on whether it is above or below 50% (you will need to modify a movie’s score to test this)

# Review Process

At the end of the exercise we will perform a group code review. Be ready to explain any problems you encountered, how you overcame them, and why you chose to implement your prototype in the way that you did.

# Resources

* Displaying a list in Angular: <https://angular.io/tutorial/toh-pt2>
* Creating an Angular component: <https://angular.io/tutorial/toh-pt3>
* Routing/navigation: <https://angular.io/tutorial/toh-pt5>
  + (note, the base repo already has a routing module)
* Importing a JSON file: <https://www.elite-corner.com/2020/07/different-ways-to-read-local-json-file-in-angular-with-example.html>