

JavaScript ES6 and OOP Part 2 Guided Project

Introduction

This week, we will be building a **Movie Db application**. The goal of this project is to create a web-based application that allows users to browse and search for information about a collection of movies. The application should provide details about each movie, including its title, genre, director, release year, plot, cast, and a poster image.

While building this application, we will be using various concepts like `map()`, `filter()`, `find()`, arrow functions, objects, encapsulation etc.

Housekeeping points

- This is a minimal example and may not follow some standard practices.
- We focus on the main flow, and not much error handling.

Problem Statement

In this project, we will build a **Movie Db application** that helps users browse and search for information about a collection of movies. The application should have the following features :

1. **Movie Listing:** Display a list of movies with basic information, including the movie title, genre, director, and a small poster image.
2. **Search Functionality:** Allow users to search for movies by entering keywords (e.g., title, genre, director, cast) in a search bar. The application should update the movie list in real-time as users type in their search queries.
3. **Movie Details:** When a user clicks on a movie card, display detailed information about the selected movie, including the full plot, cast, and a larger poster image. Include a "Back" button to return to the list of movies.
4. **Filter:** Implement the ability to filter movies by title, genre, director or cast, enabling users to select a specific title or genre or director or cast and see only movies in that category.

Functional Requirements:

- The application should be built using HTML, CSS, and JavaScript.
- Movie data, including details and images, should be stored in a JS file in JSON format.
- The user interface should be clean, intuitive, and responsive.
- Ensure that the application is well-tested and functions smoothly, including handling edge cases (e.g., no search results found).