Movie App Documentation

REACT GHR_SWD2_S2D

Group members:

- Asmaa Gameel
- El-Shimaa Ebrahim
- Nourhan Magdy
- Saif Toaima
- Youssef Mohamed
- Yasmeen Mohamed

Contents:

1. Introduction

The **React Movie App** is a modern web application designed to allow users to discover, explore, and search for movies. It pulls data in real-time from The Movie Database (TMDB) API, showcasing trending movies, search results, and detailed information about each movie, including ratings, descriptions, and release dates.

The app is built with a focus on clean UI, responsive design, and smooth navigation. It's designed as a single-page application (SPA), leveraging React's capabilities for creating dynamic user experiences. This app serves as a practical demonstration of how to integrate third-party APIs into a React project and manage dynamic content effectively.

2. Features

The React Movie App includes the following core features:

- **Trending Movies**: Displays a list of movies that are trending globally for the week, fetched dynamically from the TMDB API.
- **Search Functionality**: Users can search for movies by their title, with results being fetched in real-time based on user input.
- **Movie Details**: Clicking on a movie leads to a detail page with comprehensive information such as title, release date, rating, and synopsis.
- **Pagination**: Handles multiple pages of movie listings, making it easy for users to navigate through large sets of results without overwhelming the interface.
- **Login System**: Users can log in to access personalized features (authentication system added).

- **Responsive UI**: The application is designed to work seamlessly on both desktop and mobile devices, ensuring a smooth user experience across various screen sizes.
- **Smooth Navigation**: React Router is utilized to create a single-page application feel, with transitions between the homepage, search results, and movie details being smooth and intuitive.

3. Technologies Used

The project leverages the following technologies:

- **React**: The core framework used to build the app's user interface, manage component states, and handle rendering efficiently.
- **React Router**: Manages client-side routing, allowing seamless navigation between different views such as home, search, and movie details pages.
- **Axios**: A lightweight library used for making HTTP requests to the TMDB API, simplifying API interaction.
- **TMDB API**: Provides real-time movie data including trending movies, search results, and detailed movie information.
- **CSS**: Used for styling the app to ensure a clean, responsive, and user-friendly design.
- **SCSS (Sass)**: Enhances CSS with variables, nesting, and more for better styling.
- **JavaScript ES6+**: Modern JavaScript syntax and features are employed throughout the project, enhancing performance and readability.

4. API Integration (TMDB)

The **React Movie App** relies on the **TMDB API** to retrieve real-time movie data. The API allows the app to pull a wide variety of information about movies, including trending movies, specific movie details, and search results based on user input.

Key API Endpoints Used:

- **Trending Movies**: Fetches the most popular movies globally within the current week.
- **Search Movies**: Allows users to search for specific movies by title.
- **Movie Details**: Fetches detailed information about a movie based on its unique ID, including the synopsis, release date, rating, and more.

The app interacts with the API using Axios, which allows for simple and clean API calls with built-in handling for promises. Each API call includes the TMDB API key, which is stored securely in the .env file to avoid exposing sensitive information.

5. Component Overview

The application is divided into multiple reusable components, which are organized into logical modules for ease of maintenance and scalability. Here's an overview of the key components used in the app:

Main Components:

- **MovieCard**: Displays an individual movie in a list or grid format. Each card includes a poster, title, and rating.

- **SearchBar**: Allows users to input a movie title and triggers the search function, updating the results dynamically.
- **MovieList**: A container that renders a list of movies (e.g., trending movies or search results).
- **MovieDetail**: Displays detailed information about a selected movie, such as its synopsis, release date, and rating.
- **LoginForm**: A form for user authentication, allowing them to log in and access personalized features.
- **Pagination**: Handles navigation between different pages of movie results when there are too many to display on a single page.

Pages:

- **Home**: The homepage that displays trending movies fetched from the API.
- **SearchResults**: Shows search results when a user searches for a movie by its title.
- **MovieDetail**: A dedicated page for detailed information on an individual movie when clicked on from the movie list.
- **LoginPage**: Allows users to log in using the authentication provider.

Each component is designed to be modular and reusable, allowing for easy expansion or modification in future iterations of the app.

6. Routing

The **React Movie App** utilizes **React Router** to enable smooth, client-side navigation between different views, mimicking the behavior of a single-page application (SPA). This allows users to move between pages such as the homepage, search results, and movie details without reloading the entire page.

Key Routes:

- **Home ("/")**: Displays the trending movies.
- **Movie Details ("/movie/**

"): Displays detailed information about a specific movie based on its unique ID from the TMDB API.

- **Search Results ("/search")**: Displays results based on the user's search query.
- **Login ("/login")**: Displays the login form.

React Router ensures a seamless user experience by enabling deep linking and browser history support, making navigation between various pages smooth and intuitive.

7. State Management

State management in the app is handled using React's built-in hooks, primarily useState and useEffect, which allow for efficient state handling and lifecycle management.

Key State Variables:

- **Movies Data**: Stores the list of trending movies or search results fetched from the API.
- **Movie Details**: Stores the detailed information of the selected movie.
- **Search Query**: Manages the current input from the search bar.
- **Pagination State**: Keeps track of the current page for both trending movies and search results.
- **User Authentication**: Manages logged-in state for users.

State updates are managed via asynchronous API calls, with the app re-rendering components dynamically based on the incoming data. The useEffect hook is crucial in this process, fetching data when components mount or when certain dependencies (such as search input) change.

8. Styling

The **React Movie App** is styled using CSS, with a focus on a clean and responsive design. Key aspects of the styling include:

Global Styling:

- The app applies global styles to ensure consistency in typography, colors, and spacing across the entire app.

Responsive Design:

- The layout is fully responsive, ensuring the app works seamlessly on both mobile and desktop devices. Flexbox and CSS Grid are used to achieve responsive designs for the movie list, detail pages, and overall layout.

Component-Level Styling:

- Each major component, such as the MovieCard, SearchBar, and Pagination, has its own specific styles to ensure modular and maintainable CSS. This separation of concerns allows for easy adjustments and reusability of styles across different parts of the application.
- **Login Page**: Styled to offer a clean, user-friendly interface for authentication.

Theme:

- The app primarily uses a neutral and dark color scheme, providing a modern and cinematic feel to align with the movie content.