

Frontend Assignment - 3 (ReactJS/NextJS)

Prerequisite

1. All assignments must be implemented using **Next.js 15**.
2. You may choose **App Router** or **Pages Router** based on the problem statement and your preference.

Movie Search App with Favorites

Objective:

Develop a sleek, responsive movie search application using **Next.js 15**, showcasing your ability to work with APIs, implement search functionality with performance optimizations (debounce), and manage local state effectively. The app should allow users to search for movies, view details, and save their favorites locally.

Requirements:

- **Movie Search Interface:**
Integrate with an external movie database API such as:
 - [OMDB API](#)
 - TMDB API
- Users should be able to search for movies using a search input field. Implement **debounced search logic** to avoid firing API calls on every keystroke (use `useEffect` with a delay or a custom debounce hook).
- **Search Results Display:**
Show a list/grid of matching movie results, including:
 - Movie poster.
 - Title.
 - Release year or rating snippet.
- If there are no results, display a user-friendly message or animation.

- **Dynamic Movie Detail Page:**

On clicking a movie, navigate to a dedicated detail page (`/movie/[id]`) showing:

- Full movie poster.
- Title, release date, genre(s).
- IMDB rating or vote average.
- Plot/overview description.

- **Favorites Feature (Local Persistence):**

Allow users to mark movies as "Favorites". This list should:

- Be persisted in **localStorage** (bonus if Zustand or Context is used to manage state).
 - Be accessible from a "Favorites" page or a toggleable drawer.
 - Allow removal of favorite movies as well.
-



Optional Enhancements:

- Show loading spinners while searching or fetching details.
- Add infinite scroll or pagination to the search results.
- Allow filtering by genre or rating using the API if supported.

Skills Evaluated:

- API integration with external movie data providers.
- Search and debounce implementation in React.
- Dynamic routing and navigation with Next.js 15.
- Client-side state management and local persistence.
- Component-driven UI design and conditional rendering.

Bonus Optional Features to Suggest:

- Light/Dark Theme Toggle
- Responsiveness
- Custom loading skeletons.
- Add pagination or infinite scroll.