**Movie Browser Application Documentation**

The Movie Browser Application is a web-based platform that allows users to browse, search, and view details about various movies. Users can mark their favorite movies and view them in a dedicated section. The application features a responsive design to ensure compatibility across various devices.

The project can be directly accessed with this link: [https://movie-browser-db.netlify.app](https://movie-browser-db.netlify.app/)

GitHub link: <https://github.com/jainisheridan/movie-db.git>

**Project working model:**

When the user starts the project, they will land on the homepage, which displays the header, a carousel slider, and a list of popular movies. The user can change the filters provided in the header, which will update both the movies shown in the carousel and the movie list. The carousel can be navigated using the arrows or by clicking on the slick dots. The header also features a search bar where the user can type a movie name (not case-sensitive) and either press Enter or click the search button to display the relevant movies in both the carousel and the movie list.

When the user hovers over a movie in the movie list, details such as the title, release date, ratings, and a brief overview will appear. Due to limited space in the card, the overview may not be fully visible. For more information, the user can click on the card, which will redirect them to a page displaying all the posters and detailed information. A back button is available at the top of this page to return to the previously viewed page.

Each card also features a button to add the movie to favorites. When the user clicks it, the movie is stored in the favorites section, which can be accessed from the header. Adding and removing movies from favorites can be done directly from the same page without needing to navigate to the favorites page.

**The projects consist of 5 key component:**

* **Main.js**: This is the core component that manages state and renders the movie cards, carousels, and details page.
* **Card.js**: Represents an individual movie card. Handles displaying the movie poster, title, and other details. It also manages adding/removing movies from favorites.
* **MovieDetails.js**: Displays detailed information about a selected movie, including an option to add or remove it from favorites.
* **MovieCarousel.js**: Displays a carousel of top movies for easy browsing.
* **Header.js**: Contains the navigation menu, including links to different movie categories and a search bar.

**Design Decisions:**

* **useState** is used for managing the state within components. No redux. The **Main** component holds the primary state for movie data, search query, and favorites, ensuring centralized management.
* The application uses a CSS file for styling. The design is kept minimalist with a focus on usability. The color scheme is based on a dark theme, ensuring a cinematic experience. Responsive design principles were followed to make the application accessible on mobile devices.
* The search bar allows users to search for movies by name. The search is case-insensitive and returns results as long as they match the input.
* Users can mark movies as favorites by clicking the “Add to Favorites” or “Remove from Favorites” button on the movie card to add or remove the card.

**Additional Features:**

**Movie Carousel:** The carousel on the homepage showcases top movies. This feature allows users to get a quick overview of trending movies.

**Movie Details:** The details of the movie like Title, Release date, Rating (with votes), Movie Runtime, Genres and overview. This feature allows the user to get more information about the particular movie.

**Possible Improvement:**

* Favorites are stored in the local system; we can create a database to store for longer time.
* Adding favourites button in movie details page, for more user satisfaction.
* Implementing filter for searching the movie.
* For mobile application, header isn’t responsive.