Movie Catalog App

Description

Welcome to the Movie Catalog App, your gateway to an extensive collection of movies from various genres and eras! This application is designed to provide users with a seamless browsing experience, allowing them to explore and discover their favourite movies effortlessly. With features such as browsing, searching, filtering, and user authentication, the Movie Catalog App offers a personalized movie discovery journey tailored to your preferences.

Additions Info

- Language - JavaScript
- Lib - React (version 18 or later)
- Node Version - 20 or later
- Dataset - movies.json
- You may use any lib you deem necessary for the completion of this project.

Tasks

1. Browse and Search:
   - Task: Implement a page where users can browse through a paginated list of movies.
     Subtasks:
     - Fetch movie data from the provided dataset.
     - Display movie posters, titles, and basic information in a grid or list format.
     - Implement pagination functionality to navigate through multiple pages of movie listings.

   - Task: Enable users to search for specific movies using a search bar.
     Subtasks:
     - Implement a search functionality that filters movies based on title or keyword matches.
     - Update the displayed movie list dynamically as the user types in the search bar.

2. Filtering Options:
   - Task: Provide basic filtering options to refine movie selection.
     Subtasks:
     - Implement filters for genre, release year, and IMDb rating.
     - Allow users to select one or more filters to narrow down the displayed movie list.

- Task: Implement at least one challenging filtering option.
  Subtasks:
  - Example: Implement a dynamic range slider for filtering movies based on length (e.g., movies between 90 to 120 minutes).
  - Example: Implement a multi-select dropdown for filtering movies by multiple genres simultaneously.

3. User Authentication:

- Task: Develop user authentication functionality for signing up and signing in.
  Subtasks:
  - Create a user registration form with fields for email address and password.
  - Implement backend validation to ensure unique email addresses and secure password storage.
  - Allow users to sign up, and save their authentication details in a JSON file for verification.
  - Enable users to sign in with their credentials, verifying them against the JSON file.
  - Implement error handling for invalid credentials or registration attempts.

Submission

- Please create a zip file containing your project and send it to samit.n@agilophiles.com by Friday, 24th May 2024.
- Additionally, include a document detailing your thought process during the implementation process. This document should explain the rationale behind your decisions, such as the choice of libraries or the layout design. Additionally, discuss any further enhancements or improvements you would have made with more time available. Your insights into the development process will provide valuable context for evaluating your work.

Disclaimer

The dataset used in this project is sourced from the open-source platform (Kaggle) and may contain content that some individuals may find offensive or inappropriate. It is the responsibility of the user to handle the data sensitively and ensure compliance with relevant guidelines and regulations. Agilophiles is not responsible for any offensive or inappropriate content within the dataset.