Movie Review SPA using AngularJS

1. **Aim:**
   The aim of this project is to Create and implement a Movie Review Single Page Application (SPA) using AngularJS to provide users with an interactive platform for browsing and exploring movie details seamlessly.

2. **Introduction:**
   This project aims to leverage the power of AngularJS to build an efficient and user-friendly Movie Review SPA. Users can effortlessly navigate through a curated list of movies, clicking on each for in-depth information and reviews. The application employs AngularJS routing and services to manage dynamic content delivery, promoting a responsive and engaging experience. Through this project, we showcase the prowess of AngularJS in crafting modular and maintainable code for front-end development, providing a foundation for developers to extend and enhance movie-related features.

3. Angular features:

The following are the angular features:

- Two-Way Data Binding: AngularJS enables seamless synchronization between the model and the view, ensuring real-time updates as data changes, providing a dynamic and responsive user interface.
- Routing: AngularJS routing is employed to enable navigation between different views (e.g., movie list and movie details) without requiring a full page reload. This feature enhances the user experience in a single-page application.
- Controllers: Controllers are used to manage the logic behind different views. For instance, the MovieListController and MovieDetailController control the behavior of the movie list and movie detail views, respectively.
- Directives: AngularJS directives are utilized for extending HTML with custom behaviors. While the example is simplified, directives can be further used to create custom elements or attributes for more complex functionality.
- Services: AngularJS services, such as MovieService, are employed to encapsulate data-related operations. In this case, the service manages the retrieval of movie data, promoting code separation and reusability.
- Dependency Injection: AngularJS's dependency injection system is leveraged, allowing components like controllers and services to declare their dependencies, making it easier to manage and test.

4. Components and Functions:

The following components and functions are used in this project:

- MovieListController: Manages the behavior of the movie list view.

  Functions: Retrieves a list of movies from the MovieService. Binds the list of movies to the $scope for rendering in the view.

- MovieDetailController: Controls the movie detail view.
  Functions: Retrieves detailed information about a specific movie based on the movie ID from the route parameters.Binds the movie details to the $scope for rendering in the view.

- MovieService: A service responsible for handling movie-related data operations.

  Functions: getMovies(): Fetches a list of movies and getMovieById(id): Retrieves detailed information about a specific movie based on the provided ID.

5. Description of Routing and Navigation:

Routing and navigation are essential concepts in web development, allowing users to move between different views or pages within a single- page application (SPA). In the context of the Movie Review SPA using AngularJS, routing and navigation are facilitated by AngularJS's built-in ngRoute module.

The following routes are defined in this project:

(i) /:This route displays the list of movies.

(ii) /movies/:id: This route displays the reviews for a given movie.

(iii) /add-review/:id:This route displays a form for users to submit a review for a given movie.

Navigation between the different routes is implemented using the **ng-href** directive. When the user clicks on this link, the Angular router will navigate to the /movies/:idroute and display the reviews for the corresponding movie.

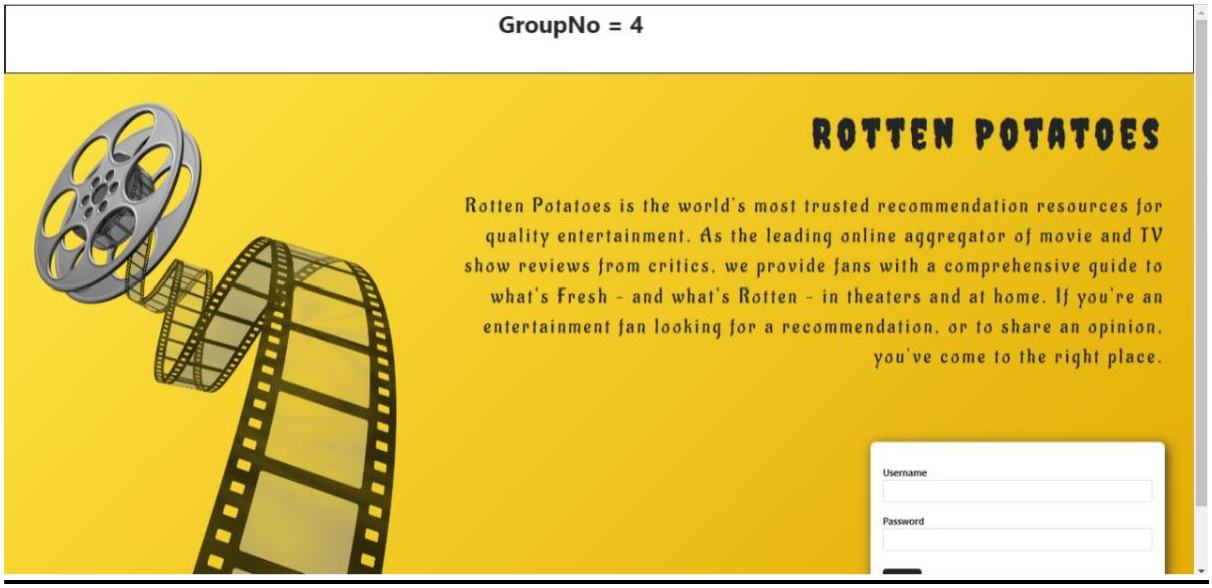
6. Description of Service and Possible Dependencies:

The following are the services that are incorporated in the project:

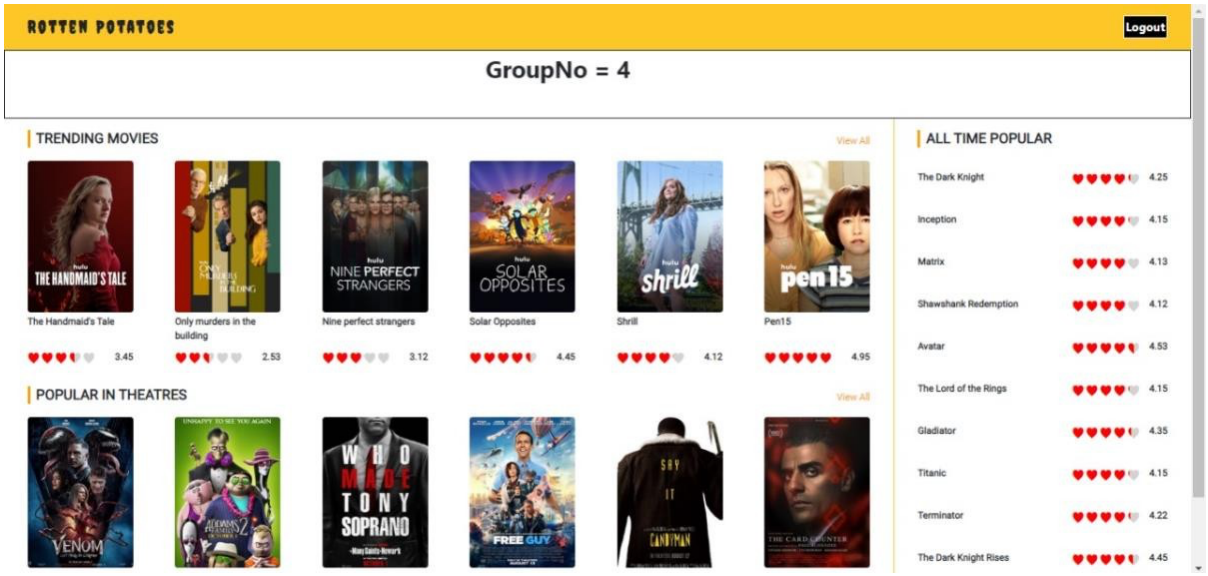
- MovieService: This service manages the movie data, including fetching movie details and reviews from a backend API.
- ReviewService: Manages the user's reviews, including posting new reviews and retrieving existing ones.
- AuthService: Handles user authentication and authorization.

- These services are injected into the "*MovieReviewController"* (Dependency) using dependency injection. This allows the controller to
  easily access the service and fetch and save movie reviews.

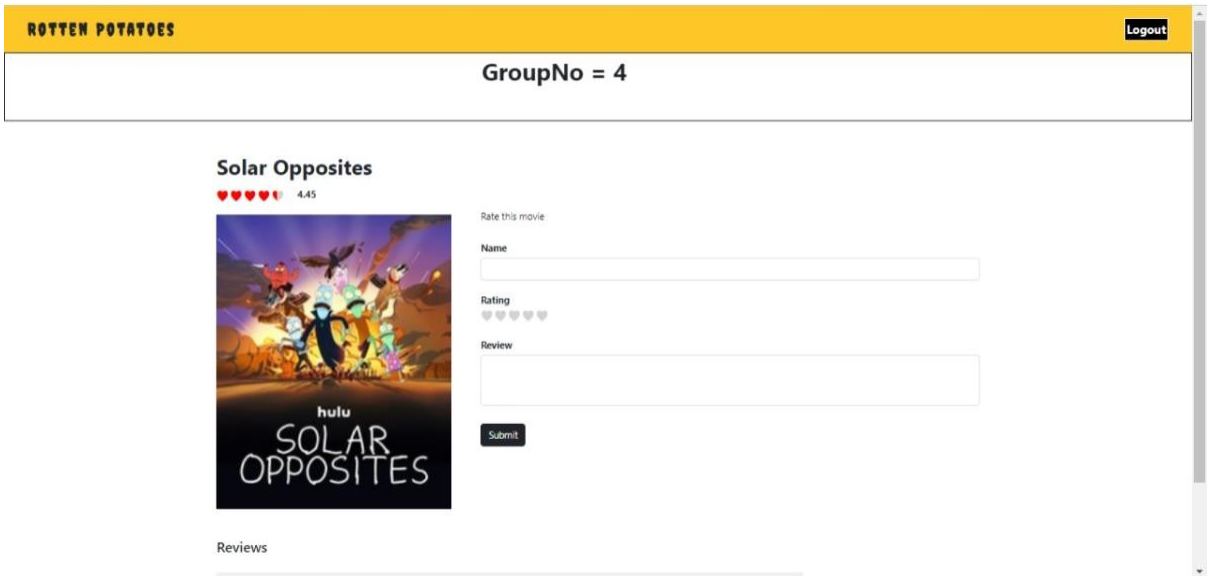
7. **Screenshots:**
   Login page:



Movie Catalogue:



Each movie's separate page:



8. **Future work:**

To modify the Movie Review SPA to a fully dynamic application and to enhance the user interface and user experience by incorporating advanced UI components and animations.

Enhanced User Interactivity: Implement more interactive features such as real-time updates, dynamic content loading, and improved user feedback.

Mobile Responsiveness: Ensure a seamless and responsive experience on various devices, especially mobile phones and tablets.

Improved Authentication and Authorization: Enhance the security features, and if not already in place, consider implementing OAuth or other secure authentication mechanisms.

9. Conclusion:

In conclusion, the Movie Review SPA is a functional and interactive web application built using AngularJs, incorporating routing, services, pipes, and structural directives. This project has demonstrated how to design and implement a Single Page Application (SPA) using AngularJS for movie reviews. The SPA provides several features that allow users to view a write movie reviews, as well as browse and search for movies. The project can be further expanded and improved to meet evolving user requirements