Epam Movie Database and Review Platform

Welcome to the Epam Movie Database and Review Platform! This project is designed to offer a comprehensive movie database where users can explore movie information, write reviews, and engage with fellow movie enthusiasts.

Functional and non-functional requirements

**Functional Requirements:**

1.User Registration and Authentication:

* Users must be able to create accounts using email and password.
* Users should be able to log in securely with their credentials.
* User sessions should be maintained across the platform.

2.Movie Search and Display:

* Users can search for movies by title, genre, director, and other criteria.
* Detailed movie information including cast, crew, plot, and release details should be displayed.

3.Review and Rating System:

* Users can write reviews for movies they've watched.
* Users can rate movies using a star-based rating system.
* Average ratings should be calculated and displayed for each movie.

4.User Profiles:

* Users can create, update, and manage their profiles.
* User profiles should display reviewed movies and ratings.

5.Backend API Endpoints:

* API endpoints for user registration, login, movie search, movie details, review submission, and user profile management should be available.

**Non-Functional Requirements:**

Performance:

* The platform should have fast response times for user interactions.
* Database queries and API calls should be optimized for efficiency.

Scalability:

* The system should be designed to handle increased user traffic and data growth.

Security:

* User passwords should be securely hashed and stored.
* Sensitive user data should be protected against unauthorized access.
* Input validation and sanitization should be in place to prevent common security vulnerabilities.

Reliability:

* The system should be available and responsive with minimal downtime.
* Data integrity and consistency should be maintained.

User Experience:

* The user interface should be visually appealing, consistent, and easy to navigate.
* User interactions should be intuitive and provide clear feedback.

Data Backup and Recovery:

* Regular data backups should be performed to ensure data integrity and facilitate recovery in case of data loss.

Maintainability:

* Code should be well-structured and modular for ease of maintenance and future enhancements.

Use cases

1. User Registration and Authentication:

Use Case: User Registration

Actor: User

Description: User creates a new account on the platform.

Flow:

1.User provides registration details (username, email, password).

2.System validates input data.

3.User account is created and stored in the database.

Use Case: User Login

Actor: User

Description: User logs into their account.

Flow:

1.User enters their login credentials.

2.System verifies the credentials.

3.User is granted access and receives an authentication token.

2. Movie Search and Display:

Use Case: Search Movies

Actor: User

Description: User searches for movies based on specific criteria.

Flow:

1.User enters search criteria (e.g., title, genre).

2.System queries the database for matching movies.

3.System displays a list of matching movies to the user.

Use Case: View Movie Details

Actor: User

Description: User selects a movie to view its detailed information.

Flow:

1.User clicks on a movie from the search results.

2.System retrieves and displays the movie's detailed information.

3. Review and Rating System:

Use Case: Write Review

Actor: User

Description: User writes a review for a movie.

Flow:

1.User selects a movie they want to review.

2.User enters their review text.

3.User submits the review.

4.System stores the review in the database.

Use Case: Rate Movie

Actor: User

Description: User rates a movie by assigning a star rating.

Flow:

1.User selects a movie they want to rate.

2.User selects a star rating (1 to 5 stars).

3.User submits the rating.

4.System calculates the average rating for the movie.

4. User Profiles:

Use Case: Update Profile

Actor: User

Description: User updates their profile information.

Flow:

1.User navigates to their profile page.

2.User modifies profile details (e.g., name, profile picture).

3.User saves the changes.

4.System updates the user's profile information.

**CRC cards**

Here you can find the CRC cards for this project <https://echeung.me/crcmaker/?share=>==

Here is the Class Diagram for this project

