**Movie Recommendation Platform**

**Table of Contents**

* **Overview**
* **Features**
* **Technologies Used**
* **Installation and Setup**
* **Usage**
* **Project Structure**
* **API Endpoints**

**Overview**

The Movie Recommendation Platform is a dynamic and interactive web application that allows users to explore, rate, and manage movies. With features such as a floating menu, personalized ratings, sign-up/sign-in capabilities, and a Django REST Framework (DRF)-powered backend, the platform offers a seamless movie browsing experience.

**Features**

* **User Authentication**:
  + Sign-up, sign-in, and sign-out functionality.
  + User session management using tokens (Django REST Framework).
* **Movie Ratings**:
  + Users can rate movies, and the system dynamically updates ratings.
  + Ratings are persisted in the backend database.
* **Floating Menu**:
  + A responsive floating menu for quick navigation.
* **Movie Overview**:
  + Dynamic UI updates based on user interactions.
  + Information includes movie posters, trailers, descriptions, and tags.
* **Django REST Framework Backend**:
  + API endpoints for user authentication, ratings, and movie data retrieval.
* **Frontend**:
  + Built using HTML, CSS, and JavaScript.
  + Smooth UI transitions and hover effects for better user experience.

**Technologies Used**

**Frontend**

* HTML5
* CSS3
* JavaScript (ES6)

**Backend**

* Python (Django)
* Django REST Framework

**Database**

* SQLite (Default with Django)

**Other Tools**

* LocalStorage for managing temporary user data.
* Fetch API for communication with the backend.

**Installation and Setup**

**1. Clone the Repository**

git clone <repository-url>

cd <project-folder>

**2. Set Up the Backend**

1. Create a virtual environment and activate it:
2. python -m venv venv
3. source venv/bin/activate # On Windows: venv\Scripts\activate
4. Install the required Python packages:
5. pip install -r requirements.txt
6. Apply migrations to set up the database:
7. python manage.py migrate
8. Start the Django development server:
9. python manage.py runserver

**3. Serve the Frontend**

1. Open the index.html file in your browser:
2. open frontend/index.html # On Windows: start frontend\index.html

**Usage**

1. **Sign Up**:  
   Navigate to the sign-up page, enter your details, and create an account.
2. **Sign In**:  
   Log in using your email or mobile number and password.
3. **Browse Movies**:  
   Explore the "Top Picks" section. Click "Rate" to add your rating for a movie.
4. **Rate Movies**:  
   Select a star rating from 1 to 10 and submit your review. Updates are reflected immediately.
5. **Sign Out**:  
   Use the floating menu to sign out and clear your session.

**Project Structure**

MovieRecommendation/

│

├── backend/

│ ├── manage.py

│ ├── db.sqlite3

│ ├── backend/ # Project folder

│ ├── movies/ # App for handling movie data

│ ├── users/ # App for user authentication

│

├── frontend/

│ ├── index.html

│ ├── css/

│ │ └── styles.css

│ ├── js/

│ │ └── scripts.js

│ ├── assets/

│ ├── posters/

│ ├── trailers/

│ └── icons/

│

└── README.md

**API Endpoints**

**Authentication**

* POST /register/ - Create a new user.
* POST /signin/ - Authenticate and log in.
* POST /logout/ - Log out and clear the session.

**Ratings**

* POST /rate\_movie/ - Submit or update a movie rating.
* POST /submit-rating/ - Submit user ratings.
* GET /get\_rate/ - Retrieve a user's rating for a specific movie.

**Movies**

* GET /api/movies/ - Retrieve all movies.
* GET /api/movies/<id>/ - Retrieve details for a specific movie.

**Ratings**

* POST /api/rate\_movie/ - Submit or update a movie rating.
* GET /api/get\_rate/?email=<email>&movie\_name=<movie\_name> - Retrieve a user's rating for a specific movie.

**Contributing**

Feel free to contribute to the project! Fork the repository, make your changes, and submit a pull request.