Project Documentation: Nox

Project Overview

Nox is a modern movie website that displays trending movies, actors, and TV shows. It provides a user-friendly interface for browsing media content and includes authentication features for user login and registration. The frontend is built using **React.js**, while the backend uses **Node.js** with a **SQLite** database for lightweight and efficient data storage.

Technologies Used

- **Frontend:** React.js
- **Backend:** Node.js (Express.js)
- **Database:** SQLite
- **Authentication:** Custom login and registration using JWT (JSON Web Tokens)

Project Structure

The project consists of **four main pages**:

1. **Home Page**
   Displays featured and trending content, including:
   - Popular movies
   - Famous actors
   - TV series highlights
     Users can browse without logging in.
2. **Details Page**
   Shows in-depth information about a selected movie, actor, or show:
   - Cover image
   - Description
   - Cast
   - Ratings
   - Trailers or clips
3. **Login Page**
   Allows existing users to log into the platform securely using their credentials. Authenticates users via the Node.js backend and stores session tokens (e.g., in localStorage or cookies).
4. **Register Page**
   Enables new users to create an account by providing:

    - Username
    - Email
    - Password
      The data is saved securely in the SQLite database.

Authentication

The authentication system uses **JWT (JSON Web Token)** to manage sessions:

- On successful login, a token is issued by the backend and stored on the client.
- Protected routes (like accessing detailed user info) check for a valid token.
- Passwords are hashed using a secure method (e.g., bcrypt).

Backend Functionality (Node.js + SQLite)

- **User Routes:**
    - `/register` – Create a new user
    - `/login` – Authenticate a user and return a token
- **Movie/Actor/Show Routes:**
    - `/details/:id` – Get full details about a specific item

□ Goals and Features

- Showcase top-rated movies, actors, and series
- Smooth and responsive UI/UX using React components
- Simple but secure user authentication system
- RESTful API communication between frontend and backend
- Data persistence with lightweight SQLite

□ Future Improvements

- Add a search bar for quick movie/actor lookup
- Include a user profile page
- Add user reviews and ratings
- Add pagination or infinite scrolling to the content feed