

Requirements & Design

User Stories

As a user, I want to register and log in so that I can access all features.

As a user, I want to browse a catalog of movies with details (title, genre, description, etc) so that I can discover new content.

As a user, I want to search by title, director, or genre so that I can find specific movies quickly.

As a user, I want to rate movies (1-5 stars) so that the system can learn my preferences.

As a user, I want personalized recommendations based on my ratings and history so that I get relevant suggestions.

AI Assisted User Stories

- As a new user, I want to register with email and password so that I can create an account. (Must-have; Acceptance: Email verification sent, duplicate emails blocked.)
- As a registered user, I want to log in securely so that I can access my profile. (Must-have; Acceptance: JWT token issued, sessions expire after 30 mins.)
- As a user, I want to view a catalog of movies with filters by genre so that I can browse easily. (Should-have; Acceptance: Paginated list, 10 items per page.)
- As a user, I want to search movies by title or director so that I can find content quickly. (Must-have; Acceptance: Results in <1 sec, fuzzy matching.)
- As a user, I want to rate a movie (1-5 stars) so that my preferences are recorded. (Must-have; Acceptance: Rating stored in DB, average updated.)
- As a user, I want recommendations based on my ratings so that I discover new movies. (Must-have; Acceptance: At least 5 suggestions, using simple algorithm like item similarity.)

Functional Requirements

User Management: Secure registration/login (use of email/password, JWT authentication), profile editing, etc

Catalog: Display a list of movies with filters/sorting and synopsis

Search: Key-word based, case-insensitive

Ratings: Store in Database, average ratings per item

Recommendations: A simple popularity based system

AI Assisted Functional Requirements

- Backend: REST API endpoints (e.g., /users/register, /movies/search, /recommendations/get).
 - Database: Store users (id, email, password_hash), movies (id, title, genre, description), ratings (user_id, movie_id, score).
 - Frontend: Web app with pages for login, catalog, search, profile, recommendations.
-

Non-Functional Requirements

Performance: Response time < 2 seconds

Security: Encrypt passwords, protect against some owasp vulnerabilities

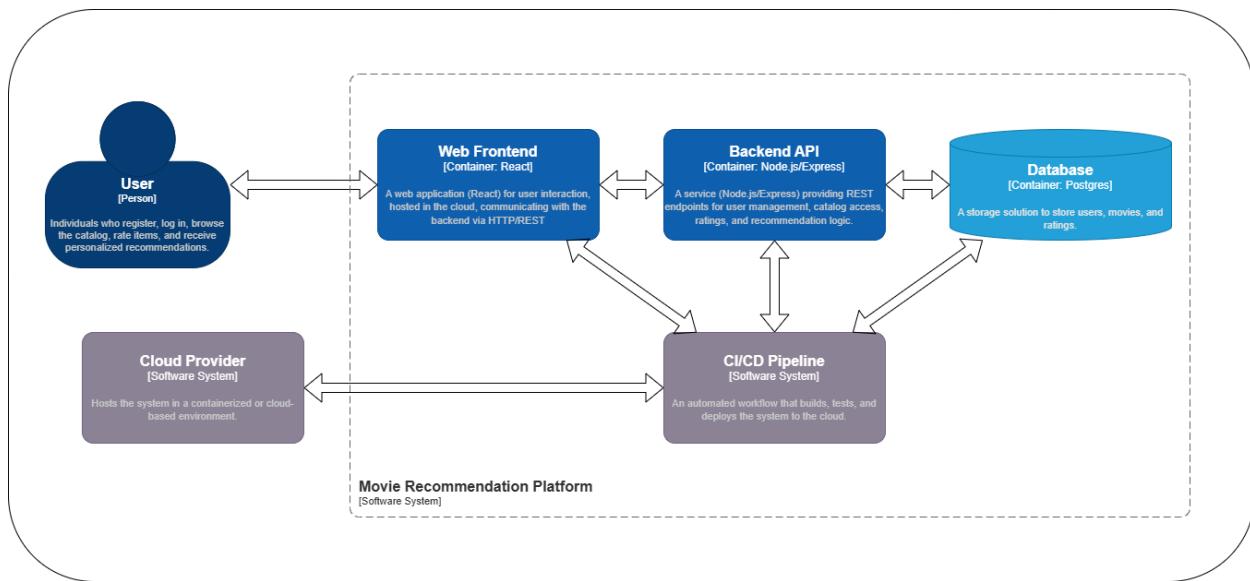
Usability: Intuitive UI, mobile-responsive

Scalability: Handle growing user base

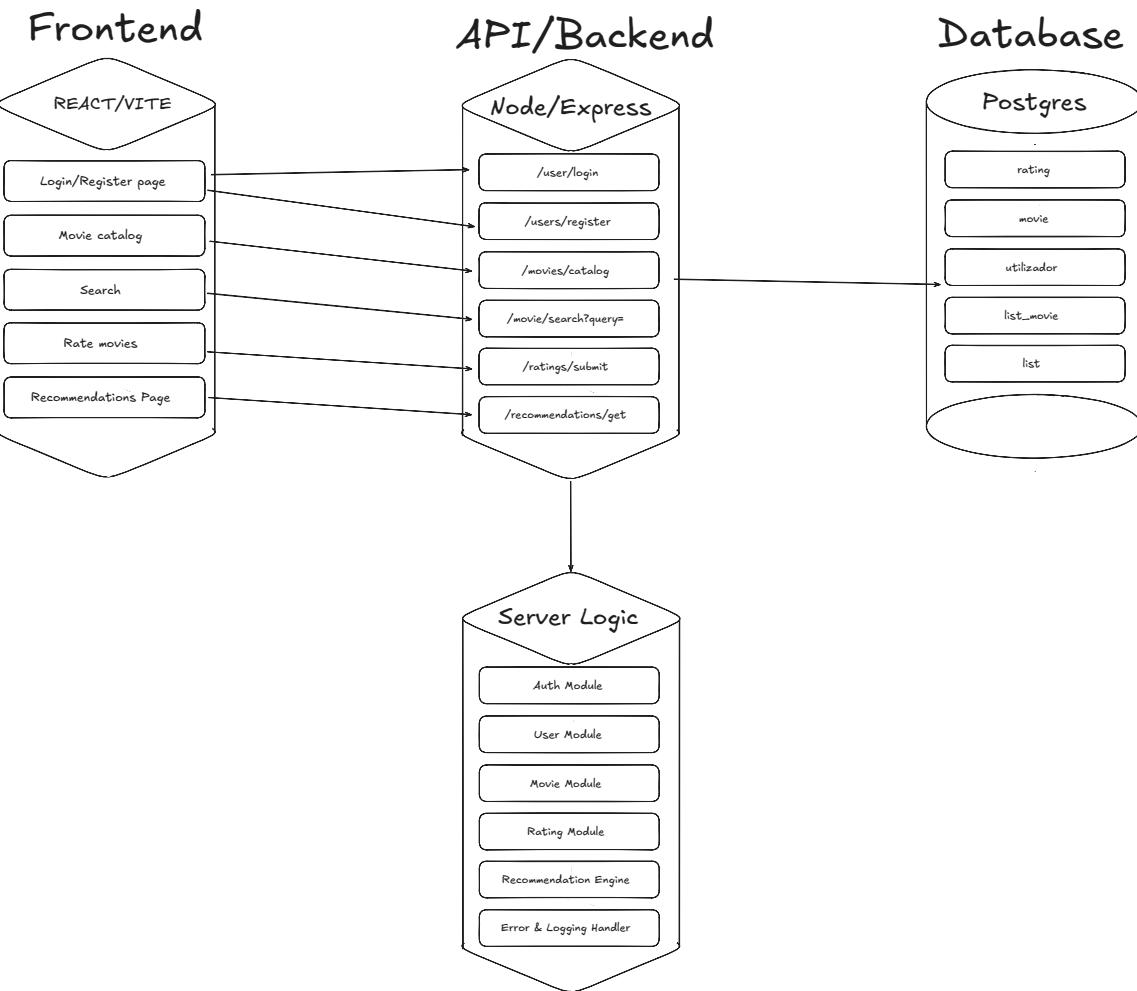
AI Assisted Non-Functional Requirements

- Reliability: 99% uptime.
 - Security: HTTPS, input validation.
 - Maintainability: Clean code, modular design.
-

Architecture



C4 Model



General Logic of the Architecture

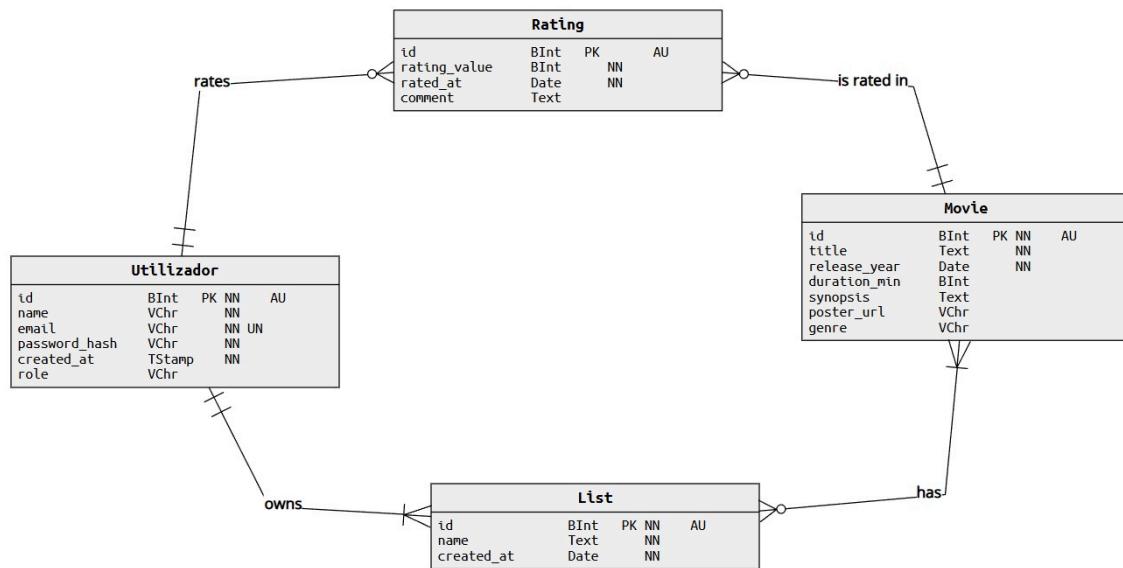
Tech Stack Proposal

Frontend: React/Vite

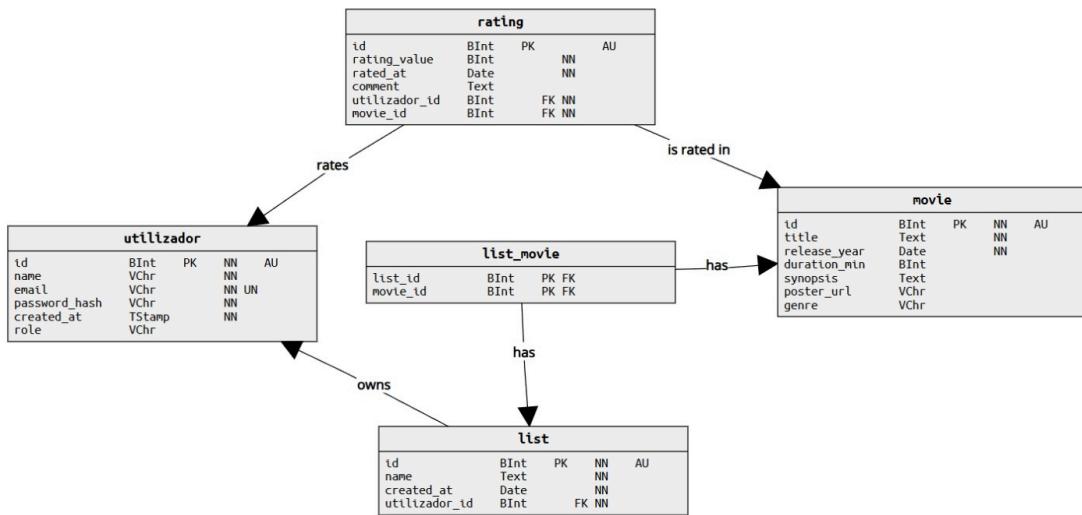
Backend: Node/Express

Database: Postgres

Diagrams



Conceptual Diagram



Physical Diagram