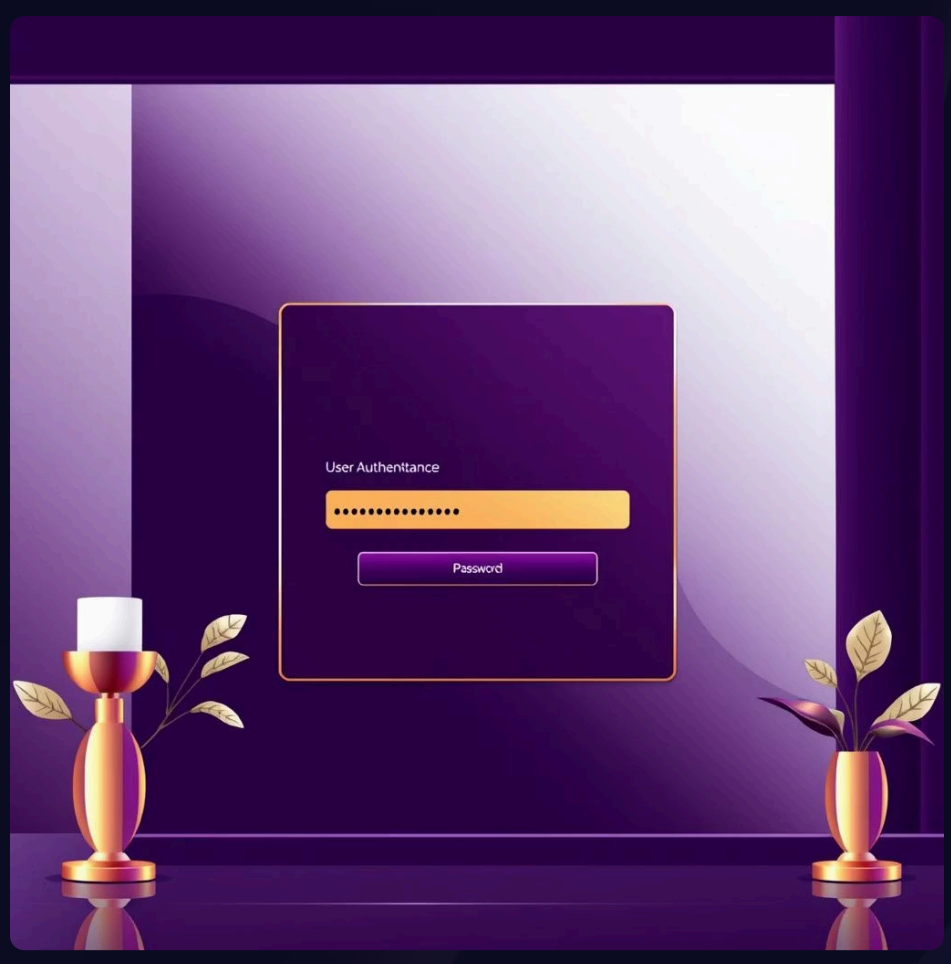
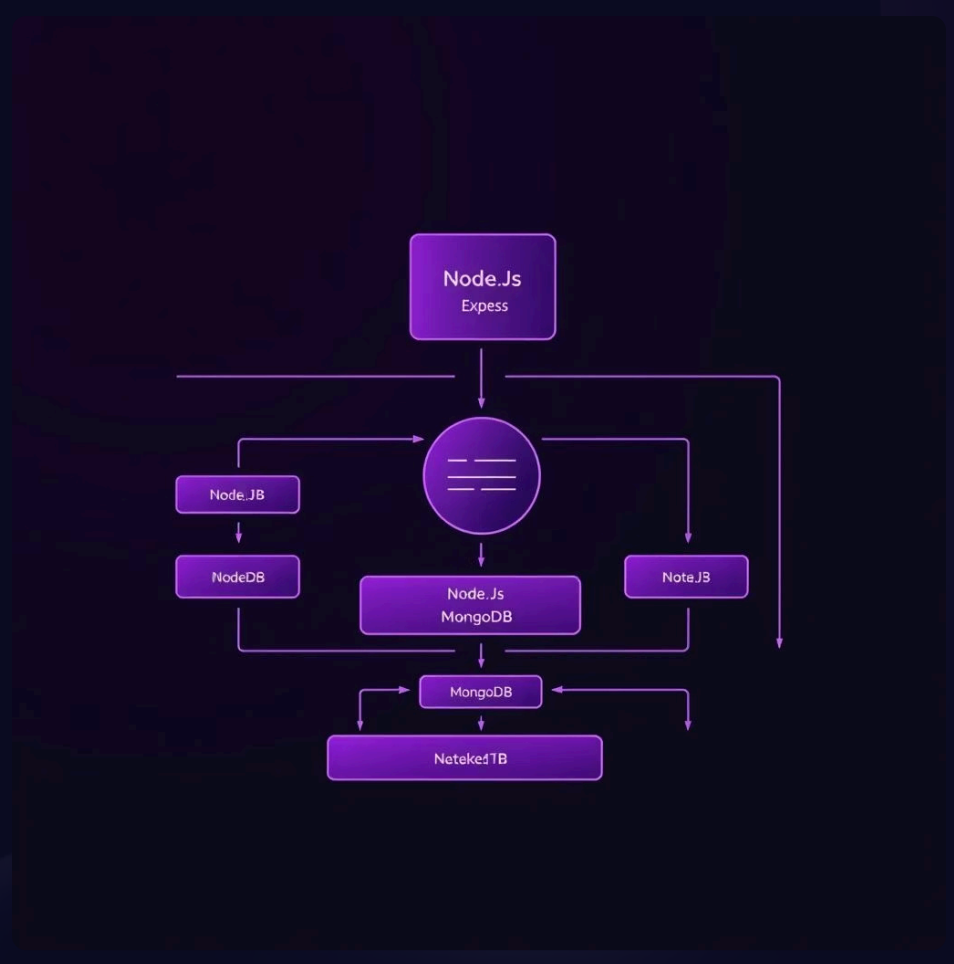
Netflix Clone

🔹 Project Overview

The Netflix Clone is a full-stack web application that replicates the core functionalities of Netflix. It allows users to register, log in, browse movies, search titles, and watch trailers. The project integrates React (frontend), Redux (state management), Node.js + Express (backend), and MongoDB (database), along with The Movie Database (TMDB) API for fetching real-time movie data.

🔷 Features





🎬 Frontend (React + Redux + Tailwind CSS)

User Authentication

Signup & login with JWT and cookie-based authentication.

Dynamic Routing

Implemented with React Router for navigation between login, browse, and search pages.

State Management

Redux Toolkit used for user sessions, movie lists, and search results.

Movie Categories

Fetches and displays Now Playing, Popular, Top Rated, and Upcoming movies from TMDB API.

Search Functionality

Search movies by title with instant results.

Trailer Playback

YouTube trailers embedded for selected movies.

Responsive UI

Styled with Tailwind CSS for a Netflix-like modern interface.

🎬 Backend (Node.js + Express + MongoDB)

User Management

REST APIs for registration, login, and logout.

Password Security

Encrypted user passwords using bcrypt.js.

JWT Authentication

Secure sessions with token handling.

Database

MongoDB for persistent user storage.

🔷 Tech Stack

Frontend

React.js, Redux Toolkit, Tailwind CSS, Axios, React Router, React Icons, Material UI Dialog

Backend

Node.js, Express.js

Database

MongoDB with Mongoose

APIs

TMDB API for movies & YouTube embed for trailers

Authentication

JWT + bcrypt.js

Tools

GitHub, VS Code, Postman

Project Architecture

Frontend Flow: Login/Signup

User initiates authentication from the client-side UI.

Auth API Call

Frontend sends authentication request to the backend API.

Redux Store Updates

User session and data are managed in the Redux store.

Browse Movies

User navigates through movie categories.

Fetch TMDB API

Frontend fetches real-time movie data from TMDB.

Display in UI

Movie data is rendered on the user interface.

Backend Flow: Client Request

Backend receives requests from the client.

Express Router

Requests are routed to appropriate controllers.

Controller (Auth logic)

Business logic, including authentication, is processed.

MongoDB

Data is stored and retrieved from the database.

Response with JWT/User data

Backend sends response back to the client, including JWT for authentication.

🔷 Challenges & Solutions

Challenge: Managing Multiple API Calls

Handling various API endpoints for popular, trending, and upcoming movies efficiently.

Solution: Custom React Hooks

Created modular custom React hooks (e.g., `useNowPlayingMovies`, `usePopularMovies`) for streamlined API calls and better code organization.

Challenge: Secure Authentication & Persistent Login

Ensuring user data security and maintaining login sessions across visits.

Solution: JWT with Cookie Storage & Redux

Implemented JWT for secure token-based authentication with cookie storage for persistence, and Redux for global state management of user sessions.

Challenge: Smooth UI Similar to Netflix

Achieving a modern, responsive, and visually appealing user interface that mirrors Netflix's design.

Solution: Tailwind CSS + Material UI Dialog

Utilized Tailwind CSS for rapid and responsive styling, combined with Material UI Dialog for elegant modals and an overall polished user experience.