Elaine Nguyen / rbo748

353 Project

Overview

DevForum is a full-stack developer discussion platform designed using the MERN-style approach, with a slight twist: instead of MongoDB, it uses CouchDB. The app is containerized via Docker Compose, ensuring easy local development and deployment.

Backend Architecture

Express.js: Acts as the main API server (server.js), handling RESTful routes for user authentication, post creation, replies, voting, and channel management.

CouchDB (nano client): A schema-less, document-based database well-suited for nested and unstructured data (e.g., threaded replies). Managed via the Nano package.

Authentication: Secured with JWT tokens (jsonwebtoken), providing stateless user sessions and role-based access (e.g., admin routes).

bcryptjs: Used for password hashing during registration/login, enhancing security.

init-couchdb.sh: Bootstraps essential system databases when CouchDB starts.

Middleware functions include:

authenticate: Verifies JWT tokens before accessing protected routes (e.g., /postmessage, /account/update).

Vote handling: Tracks upvotes and downvotes per user to avoid duplicate voting

Frontend Architecture

Routing: react-router-dom is used to create SPA-style routes such as /login, /register, /channels, and dynamic user profiles (/user/:username).

ProtectedRoute Component: A custom wrapper that ensures only authenticated users can access certain pages (e.g., ChannelPage, AccountPage).

Containerized Deployment

Docker Compose (docker-compose.yml) sets up:

couchdb: Database container.

backend: Node.js API.

frontend: React app. Each service is isolated, linked via internal networking, and automatically initialized.

Libraries

Package	**Purpose**
express	Web server & routing
nano	CouchDB client
jsonwebtoken	Auth w/ JWT
bcryptjs	Password hashing
react-router-dom	SPA routing
react-bootstrap	UI components
cors	Enable frontend-backend communication
dotenv	Manager environment configs