Groovo Master Documentation

Full Codebase Overview

A complete guide for developers, teammates, and future maintainers

Introduction

This document serves as the master technical guide for the entire application. Its purpose is to give developers, teammates, and future maintainers a clear understanding of the system architecture, major components, data flow, and engineering decisions that shape the project. By consolidating all technical details into one place, this documentation provides a reliable reference for onboarding, debugging, feature expansion, and long-term maintenance.

The application is a full-stack music discovery and review platform inspired by modern social media and content-sharing ecosystems. Users can review albums, track their listening history, browse new releases, view global and personalized feeds, and interact with other users through comments, likes, and follows. The system integrates tightly with the Spotify Web API to fetch album, artist, and track metadata, while also maintaining its own internal user-generated content layer.

This document outlines the architecture, folder structure, major features, API endpoints, database design, and development standards that keep the codebase scalable, modular, and maintainable.

Project Overview

Groovo is a social music platform built with Next.js 15, MongoDB, and NextAuth. Users can discover albums, write reviews, rate albums on a 5-star system, follow other users, comment on reviews, and maintain a "Top 5" favorites list. The platform integrates with Spotify's API for album metadata and Last.fm for additional album information.

High-Level Architecture

Frontend: Next.js 15 App Router using a hybrid of React Server Components (RSC) and Client Components. Pages handle server-side data fetching, while interactive UI elements use client-side React logic.

Backend: Next.js API Routes (Route Handlers) serve as the backend layer for all application logic. These handlers manage authentication, user profiles, reviews, feed generation, and custom integrations with external APIs.

Authentication: NextAuth.js (Auth.js) configured with Google OAuth and Credentials providers. Authentication is fully JWT-based, with user identity extracted server-side from `req.user.sub` to ensure security and avoid exposing user IDs on the frontend.

Database: MongoDB Atlas as the primary database. Collections store users, reviews, comments, likes, follows, and cached album snapshots for improved performance and consistency.

External APIs: Spotify Web API for album, artist, and track metadata.
Last.fm API (optional/auxiliary) for additional metadata, similar-artists data, and future expansion.

Styling: Tailwind CSS for utility-first, responsive UI design.

Type Safety: TypeScript used across the entire codebase, including types for API responses, database schemas, hooks, and components.