Project Technology & Feature List

This document outlines the core technologies and application features based on the provided project files, with

Core Technologies (The "Engine")

This is the underlying stack and tooling used to build and run your application.

* **Project Structure:** Monorepo managed with **pnpm** and **Turborepo**.
* **Runtimes:**
    * **Node.js**: The primary runtime for the entire project.
    * **Bun**: Used as the runtime for the dedicated backend service.
* **Languages:**
    * **TypeScript**: The primary language for the entire monorepo.
    * **SQL**: Used for database functions and queries.
* **Frameworks:**
    * **Next.js (v15)**: The core React framework for the main web application, using **Turborepo** (and Turbop
    * **Vite**: Used as the build tool for the separate ‘backend‘ service.
* **UI & Styling:**
    * **React (v19)**: The foundational UI library.
    * **Tailwind CSS**: The utility-first CSS framework for all styling.
    * **Radix UI**: Used for accessible, unstyled UI primitives (e.g., ‘Dialog‘, ‘Select‘, ‘Dropdown‘).
    * **shadcn/ui Utilities**: The project uses utilities common to ‘shadcn/ui‘ like ‘class-variance-authority‘, ‘clsx‘, a
* **Database & Backend-as-a-Service (BaaS):**
    * **Firebase**: Used as the core BaaS for database, authentication, and backend services (replacing Supaba
    * **Prisma**: The primary ORM for database access and schema management.
    * **Firebase Cloud Functions**: Serverless functions for AI tasks, search, and other backend logic (replacing
* **Authentication:** **Clerk** is the primary service for user authentication and management.
* **Payments & Monetization:** **Stripe** is used for all payments, including subscriptions and creator payouts
* **File Storage:** **AWS S3** (or a compatible service like **Cloudflare R2**) is used for file and media upload
* **AI & Embeddings:** **OpenAI** client is used for AI features, including generating vector embeddings for se
* **State Management:**
    * **Jotai**: Used for global client-side state management.
    * **TanStack Query**: Used for server-side state, caching, and data fetching.
* **In-App Editors:**
    * **Monaco Editor**: Provides a rich, VS Code-like text editor in the browser.
    * **CodeSandbox Sandpack**: Used to create live, runnable code sandboxes.
* **Analytics:**
    * **Amplitude**: For product analytics.
    * **PostHog**: For product analytics and session recording.
* **Transactional Email:** **Resend** and **React Email** are used for sending emails.
* **Testing:** **Vitest** is the framework used for unit and integration tests.

Application Features (The "Experience")

This is what users can actually *do* with your application, based on the file list.

* **Component Marketplace (Core Feature):**
    * **Browse Components:** Users can browse components listed on the main page, by category/collection, or
    * **Component Pages:** Each component has a dedicated public page showing its details and demos.
    * **Interactive Previews:** Users can view interactive component demos.
    * **User Profile Pages:** Public-facing pages that list all components and bundles published by a specific use

* **Creator Studio & Publishing:**