

DevID Guardian - Advanced Stack & Tooling Blueprint

1. Project Overview

DevID Guardian is an advanced, developer-focused identity and account usage manager.

Built from scratch by Nitya without AI assistance, this project will track and organize developer IDs (like Google, GitHub, Microsoft), monitor where they are used, store associated credentials, and analyze usage, all while emphasizing security and data privacy.

2. Tech Stack (Advanced Level)

Frontend:

- Next.js (App Router for modern routing)
- TailwindCSS (utility-first CSS)
- ShadCN UI (for elegant components)
- Zustand (for state management)

Backend:

- Node.js with Express.js (for RESTful APIs)
- MongoDB (Atlas, for cloud DB) OR SQLite (for local storage)

Security & Storage:

- Web Crypto API (on frontend)
- Node.js crypto module (on backend)
- OAuth 2.0 integration (manual or with NextAuth)

Deployment:

- Vercel (frontend hosting)
- Render (backend API hosting)

Dev Tools:

- ESLint + Prettier (code quality)
- Husky (git hooks)

DevID Guardian - Advanced Stack & Tooling Blueprint

- GitHub Actions (CI/CD automation)

3. Key Features Summary

- Add/manage multiple developer IDs
- Track ID usage across platforms/tools
- Securely store passwords, API tokens
- Dashboard with filters and search
- Optional: Export data as .json or .csv
- Bonus: Activity tracking, ethical alerts

4. Security Focus

- Data stored locally or encrypted before saving to DB
- Passwords & tokens encrypted using AES (Web Crypto or Node Crypto)
- Only user has access; zero cloud sync unless added later
- Manual breach alerts or suspicious site warnings

5. Suggested Folder Structure

/dev-id-guardian
??? /app (Next.js routing)
??? /components
??? /lib (utilities like encryption)
??? /pages/api (Express APIs if hybrid)
??? /styles
??? /hooks
??? /store (Zustand logic)

6. Project Flow

1. Design rough UI
2. Build local ID manager with Next.js UI

DevID Guardian - Advanced Stack & Tooling Blueprint

3. Add secure vault for credentials
4. Implement local DB or Mongo connection
5. Add analytics, tag-based filter system
6. Test all encryption methods
7. Polish UI and deploy

7. Bonus Tools & Enhancements

- Make it a PWA for offline usage
- Desktop version using Electron
- Export/import encrypted backups
- Dashboard stats & reminders