| **Area** | **Change** | **Why it matters** |
| --- | --- | --- |
| **Runtime env-vars** | Add a *single* Zod (or EnvSafe) validator that’s imported by both Vite and Remix. Fail fast on missing or malformed values. Also expand example.env with *all* required variables, grouped by concern. | Prevents silent deploy failures & documents the contract for users. |
| **Global error handling** | Export ErrorBoundary & CatchBoundary in app/root.tsx. Log the error (server) and show a minimal UX (client). | Without this, unhandled exceptions bubble to Shopify, giving merchants a blank screen. |
| **Security headers & CSP** | Drop @fastify/helmet (or remix-utils cspHeader) into your entry.server.tsx. Add a default CSP that works with the Shopify App Bridge script. | Shopify apps are rendered in an <iframe>; solid CSP/frames headers avoid click-jacking & rogue scripts. |
| **Consistent logging** | You ship **Pino** and a hand-rolled color logger. Pick one (Pino + pino-pretty in dev) and delete the other. Wrap in a small adapter (logger.info({ctx}, msg)), so template users can swap providers. | Avoids duplicate deps & unclear guidance to adopters. |
| **Loader/action return helpers** | Replace plain objects (return { products }) with json({ products }) from @remix-run/node. That sets Content-Type, status codes and lets you add headers (e.g. caching) later. | Subtle but important once you enable caching / response headers. |
| **CI skeleton** | Add a minimal GitHub Actions workflow: pnpm install --frozen-lockfile, pnpm run lint, pnpm run test, pnpm run build. | Templates that build in CI give immediate confidence and show newcomers the happy path. |

**✅ Medium-priority improvements**

| **Scope** | **Suggestion** |
| --- | --- |
| **Database** | Include one *real* pg-migrate migration (e.g. create shopify\_sessions table) so users see the pattern. |
| **Docker** | A docker-compose.yml with Postgres 16 + Node service running pnpm prod lets devs spin up quickly/local CI runners pass. |
| **GraphQL types** | You already depend on @shopify/api-codegen-preset. Wire it into pnpm run graphql-codegen so \*.graphql files generate TS types automatically (fed into GraphQLProductManager). |
| **Testing examples** | Ship one integration-style test that hits a loader with a mocked authenticate.admin to demonstrate pattern; gives new users a blueprint. |
| **Pre-commit hooks** | Add husky + lint-staged to run eslint --fix & prettier --write on changed files only. |
| **Docs** | In the README: \* Project goals & folder map \* “How to upgrade Shopify API version” \* “How to add new PG migrations” section. |
| **Monitoring** | Show how to initialise Bugsnag only when the BUGSNAG\_KEY var is present; otherwise use a no-op stub. |

**🧹 Polish & housekeeping**

* Delete sample files you don’t intend to ship (example.sql, empty PGShopSessionManager placeholder, etc.).
* Add eslint-config-prettier *and* .eslintrc.cjs that extends both Remix & Prettier rules so they don’t fight.
* root.tsx should declare lang="en" on <html> for accessibility.
* Pin Node engines in package.json ("engines": { "node": ">=20.9 <21" }) to avoid minor-breaking changes in new LTS.
* Consider renaming DataManagers to Repositories (clearer DDD term), but that’s taste.