Associate Developer Guide — Harry-Store

Purpose: Get you productive quickly reading and extending this Next.js 15 ecommerce app.

1. What this app does

- Lists men/women apparel with product detail pages
- Cart and wishlist with local persistence
- Toast notifications for user feedback
- Fashion news via RSS on /news
- Merchant config to change name, currency/locale, and product source (static or JSON)

2. Quick start

- Dev: `npm install` then `npm run dev`
- Prod: `npm run build` then `npm run start`
- Open `http://localhost:3000`

3. Read order (30–60 min)

- `src/app/layout.tsx` — global layout, providers, metadata
- `src/components/Header.tsx` — navigation and counts
- `src/store/StoreProvider.tsx` — cart/wishlist state and actions
- `src/components/ToastProvider.tsx` — notifications
- `src/data/products.ts` — sample products
- `src/data/catalog.ts` — static vs JSON product loading
- Pages:
  - `src/app/page.tsx` — home with featured products
  - `src/app/men/page.tsx` , `src/app/women/page.tsx`
  - `src/app/product/[handle]/page.tsx` — product details
  - `src/app/cart/page.tsx` , `src/app/cart/CheckoutClient.tsx` , `src/app/cart/success/page.tsx`
  - `src/app/wishlist/*`
  - `src/app/news/page.tsx` — RSS parsing

4. Architecture at a glance

- Next.js App Router: server components by default; client components with `"use client"`
- Styling: Tailwind v4 via `src/app/globals.css`
- State: Context in `StoreProvider` with localStorage hydration
- Data: `catalog.ts` chooses static or JSON products based on `src/config/merchant.ts`
- Images: Located under `public/images/men|women` ; served by `next/image`

5. Merchant configuration

- File: `src/config/merchant.ts`
- Select merchant via env `NEXT_PUBLIC_MERCHANT`
- Controls: `siteName` , `description` , `currency` , `locale` , `productsSource` , `productsJsonPath`
- Used in metadata, price formatting, and product loaders

6. Key flows

- Add to cart
  - UI: Product Actions -> `useStore().addToCart()` -> context updates -> localStorage persisted
  - Toast shown via `useToast()`
- Checkout (mock)

    - Button -> `CheckoutClient.tsx` -> simulate delay -> clear cart -> toast -> redirect `/cart/success`
- Wishlist
    - Add/remove via `useStore()` ; mirrored to localStorage; toasts for feedback
- News
    - Server component parses multiple RSS feeds using `rss-parser` , dedupes, displays 20 items

7. Common edits

- Add a product (static): `src/data/products.ts` , place image file under `public/images/{men|women}`
- Switch to JSON products: set `productsSource: "json"` and `productsJsonPath` , then create the JSON file with the `Product` shape
- Add a page: create a new folder in `src/app` with `page.tsx`
- Change currency/locale or site name: update `src/config/merchant.ts`

8. Conventions

- TypeScript for all code; explicit props for components
- Descriptive variable names, early returns
- Minimal inline comments; prefer clear code and short docstrings
- Tailwind classes for styling; avoid deep overrides

9. Troubleshooting

- Port 3000 in use: stop other Next.js/Vite servers or use `set PORT=3001; npm run start`
- Images not loading: ensure local paths under `public/images/*` and referenced `imageUrl` match
- No news: some feeds rate-limit or change; try again or add another feed URL in `src/app/news/page.tsx`
- Dev overlay errors from extensions: test in Incognito without extensions

10. Next steps / roadmap

- Replace mock checkout with Stripe/PayPal
- Faceted filters (category, price)
- Search, pagination
- i18n using `locale` from merchant config
- Promo banners and coupon codes

Appendix: Product shape

```
id: string;
slug: string;
name: string;
description: string;
priceCents: number;
imageUrl: string;
category: "tshirts"|"shirts"|"jackets"|"pants"|"hoodies"|"dresses"|"skirts"|"accessories";
gender: "men"|"women";
```