Technical Specification — Harry-Store

Overview

Harry-Store is a Next.js 15 app-router ecommerce storefront with Tailwind CSS, providing product listings, product details, cart and wishlist with persistence, a news feed via RSS, and merchant-specific configuration for name, currency, locale, and product source.

Architecture

- Framework: Next.js 15 (App Router), React 19, TypeScript
- Styling: Tailwind CSS v4 (inline @import in `globals.css`)
- State: React Context (`StoreProvider`) persisted to localStorage
- Images: `next/image` with local images in `public/images/*`
- Data: static seed in `src/data/products.ts` or configurable JSON source via `src/data/catalog.ts`
- Notifications: `ToastProvider` client component
- News: server component parses RSS on-demand using `rss-parser`

Key Modules

- `src/app/*` — server/client components per route
  - `/` lists featured products
  - `/men` , `/women` filter by gender
  - `/product/[handle]` fetch by product `id`
  - `/cart` and `/cart/success`
  - `/wishlist` (server page rendering a client component)
  - `/news` RSS aggregation
- `src/components/Header.tsx` — navbar with counts for cart/wishlist and link to News
- `src/components/ProductGrid.tsx` , `ProductCard.tsx` , `Hero.tsx`
- `src/store/StoreProvider.tsx` — cart and wishlist state, actions, localStorage persistence
- `src/components/ToastProvider.tsx` — transient toasts
- `src/data/products.ts` — static products
- `src/data/catalog.ts` — load products from static or JSON file
- `src/config/merchant.ts` — merchant selection and config

Merchant Config

`getMerchantConfig()` returns `{ siteName, description, currency, locale, productsSource, productsJsonPath }` based on `NEXT_PUBLIC_MERCHANT` . Used in:

- Metadata in `layout.tsx`
- Price labels in UI
- Product loaders (static vs JSON)

Data Contracts

Product interface (`src/types/product.ts`):

- id, slug, name, description, priceCents, imageUrl, category, gender

Cart/Wishlist Behavior

- Add to cart increments existing quantities
- Wishlist is a set of productIds
- Counts displayed in header via context
- Toasts on add/remove/checkout

Checkout Flow

- Mock checkout in `CheckoutClient.tsx`:
  - Simulates network delay, clears cart, shows toast, redirects to `/cart/success`

News Feed

- `/news` uses `rss-parser` to aggregate from multiple feeds (Vogue, Harper’s Bazaar, ELLE)
- Dedupe by link/title; shows up to 20 items

Build and Deploy

- Dev: `npm run dev`
- Prod: `npm run build && npm run start`
- Ensure `NEXT_PUBLIC_MERCHANT` is set if using merchant-specific config

Security/Privacy Considerations

- No PII is stored; localStorage contains cart/wishlist (ephemeral)
- No server-side secrets in repo; RSS requires no secrets
- CSP can be added if blocking extension injections is needed