

# Marketplace Web App Documentation

## Overview

A comprehensive online marketplace built with Supabase (Postgres + RLS + storage) and a Vite-powered React frontend. The platform supports authenticated sellers, chat, POS, analytics, and modern UI controls. Use this document to understand every page/function, verify behavior, and tick off what works.

## Tech Stack & Architecture

- **Frontend:** Vite + React 18 + TypeScript, Tailwind CSS with custom tokens (`src/index.css`, `tailwind.config.ts`). Uses Radix UI primitives wrapped in `/components/ui`, plus Sonner for toast and React Query for data caching.
- **Backend/Data:** Supabase handles auth, Postgres storage, RLS, migrations in `supabase/migrations`. Tables include `profiles`, `products`, `orders`, `order_items`, `payments`, `variants`, `analytics_daily`, `product_views`, `profile_views`, `receipt_snapshots`, `pos_sessions`, `pos_payment_methods`, `locations`, `categories`, etc.
- **Integrations:** Supabase Client in `src/integrations/supabase/client.ts`, Edge function `analyze-product` for Gemini Vision; geolocation APIs for location features; Supabase Storage for images.
- **Observability:** `src/lib/monitoring` tracks errors and Supabase connectivity. Theme state managed via `next-themes` (ThemeProvider + toggle). POS offline sync stored with IndexedDB/outbox pattern (`offlineSync`).

## Pages & Functional Matrix

### Global UI

- Responsive Navbar with theme toggle, mobile sheet menu, and gradient backdrop.
- Footer with gradient stripe, links to pages, and social icons.
- Layout uses `main` container padded at top to accommodate sticky nav.
- Dark mode works via CSS variables and `@layer` base tokens.

### Home (/)

- Hero block gradient background, search box, location selector, stats, CTA.
- Product grid: paginated list, skeleton loader, and fallback state when empty.
- Categories section: chips linking to `/category`, showing counts.
- Quick stats section (static numbers) and CTA to start selling.
- Footer & global nav repeated.

### Authentication (/auth)

- Supabase sign-in/sign-up flows with session persistence.
- Auth guard ensures some routes redirect when unauthenticated.

### Product Detail (/products/:productId)

- Data fetch includes product info, seller profile, images, reviews, variants.
- Image gallery with thumbnails opens `ImageLightbox` (keyboard & click navigation). Click handler uses `lightboxOpen` state.
- Seller badge card with avatar, review stats, company info, chat button.
- Ratings display and reviews list (with submission form requiring auth).
- Analytics: product view tracking writes to `product_views`, auto-creates chat conversation/message for seller.
- Related products grid showing similar items by category with pagination.
- Receipt print integration via `printHtml` and `buildReceiptHtml` (dynamic import + fallback static import warnings from Vite due to mixed usage).

### Product Form (/products/new/edit & edit)

- Business type banner (from `localStorage` via `/sell` page or Settings) shows currently selected type; redirect to `/sell` if missing.
- AI analysis card integrates Gemini Vision (image analysis). Suggests description, category, subcategory, variants.
- Category select filtered by business type keywords; fallback to all categories. Subcategories automatically loaded from products matching selected category.
- Variant suggestion chips appear via `variantSuggestions` (mapped by business type) and can be toggled for creation (default price = main price, stock 0). Selected names inserted into `public.variants` after product creation/update.
- Saves to Supabase `products` table with denormalized category and location text; reruns fallback to handle schema caches.
- Supports image uploads to Supabase Storage, preview, status, location select (from `locations` table).

### Business Type Selection (/sell, Settings)

- Selecting business type stores `businessType` & `businessTypeKeywords` in `localStorage` for filtering ProductForm.
- Settings page uses stored value, allows change, and stores keywords for category filtering.
- ProductForm reads keywords to show only relevant categories/variants.

### Settings (/settings)

- Business type selector, receipt settings (business name, theme, accent, include tax, footer note).
- Store location block toggle `display_location`, show `lat/long/district` fields, button to Use My Current Location (prompts geolocation, saves coordinates, auto-assigns district `abdi asiis` when `lat 2.0370-2.0575` & `lon 45.3280-45.3600`).
- Distances stored in `profiles.latitude`, `longitude`, `location_name`, `display_location`.
- Saves to Supabase via `profiles` table.
- Receipt preview buttons use `buildReceiptHtml` to show templates.


### Category Page (/category/:categoryName)

- Gradient hero with search/filters, dynamic subcategory list, sort dropdown, location filter.
- Active filters shown as chips with remove actions + "Clear all".
- Product grid similar to home, with skeleton loader and empty state.
- Additional info section describing the category.

### Profile (/profile/:userId)

- Hero card with avatar, gradient background, badges for username, views, rating, product count.
- CTA buttons: owners see Edit/Profile/Business settings; others see Chat button.
- Stats cards: Profile Views, Average Rating (with counts), Active Products.
- Recent reviews list with avatars and rating visualization.
- Product grid listing user's products.
- Display location indicator when `display_location true` (MapPin icon and district name).
- Name change restricts to once per 7 days (uses `full_name_changed_at` timeline).

### Chat & Conversations

- `chat_conversations` and `chat_messages` tables with RLS ensure only participants read messages.
- Product view triggers conversations; seller sees message  viewed your product: ... (makes viewer aware). Viewer sees their own chat only.
- `/conversations` and `/chat/:userId` pages allow message streaming + send.

### POS (/pos, /pos/dashboard, /pos/inventory)

- Multi-order tabs (create, close, switch) for concurrent carts.
- Cart component with line selection, quantity change buttons, discount logic, order-level discount, checkout button.

- Numpad allows editing selected line's qty/price/discount; highlight indicates selection.
- Payment modal supports splitting payments, change due for cash overpayment, confirm only when total covered.
- Checkout flow: creates orders, order\_items, payments, prints receipt (if settings exist), stores receipt\_snapshots, and queue offline data when offline.
- Offline sync caches products + orders, replays when back online (offlineSync utilities).
- POS sessions/pos\_payment\_methods tables added via migrations for future session/cash control.

## Analytics & Rollups

- Daily rollups (analytics\_daily) computed from profile views, product views, orders, revenue; refreshed via stored function refresh\_daily\_rollups. RLS allows sellers to view only their rollups.
- product\_cart\_adds, product\_views, profile\_views track user actions; indexes dedupe per day via view\_day\_utc column + trigger function set\_view\_day\_utc (see migration 20251112161500\_add\_view\_rate\_limits.sql).
- Order rules enforced by trigger trg\_enforce\_order\_rules: restricts status transitions and ensures payments cover totals before completing.

## Database & Migrations Highlights

- profiles: base columns + company\_name, contact\_number, display\_location, latitude, longitude, location\_name (migrations 20251112090000, 20251112171000).
- products: columns for category, subcategory, category\_id, location\_id, plus statuses and triggers.
- orders, order\_items, payments: RLS ensures created\_by, is\_order\_owner helper, snapshot/triggers, receipt prints.
- variants: each product variant per ProductForm, RLS ensures seller-specific.
- pos\_sessions, pos\_payment\_methods created to tie payments to sessions.
- receipt\_snapshots: stores reprint data.
- categories, locations: lookup tables for filtering.

## Full Feature Checklist (tick or mark)

### Auth & Profiles

- ☐ Supabase Auth sign-in/up. (Check /auth.)
- ☐ Profile creation on signup (trigger handle\_new\_user).
- ☐ Profile edit flow (image upload, company, contact, bio).
- ☐ Full-name cooldown (only change once per week).
- ☐ Display location toggle + geolocation capture (Settings).
- ☐ Profile view tracking (stored in profile\_views).

### Products & Listings

- ☐ Business Type stored (localStorage + Settings).
- ☐ Category list filtered by business type keywords.
- ☐ Subcategory suggestions from existing products (ProductForm).
- ☐ Variant template chips appear by business type.
- ☐ AI analyzer prefill works (description, category, variant).
- ☐ Image upload + preview uses Supabase Storage.
- ☐ Location select uses locations table.
- ☐ Product creation/update handles category/location fallback.
- ☐ Product detail gallery + lightbox.
- ☐ Product view tracking writes to product\_views (unique per day).
- ☐ Related products show active items by category.
- ☐ Receipt print + snapshot on checkout.

### POS / Sales

- ☐ Multi-order tabs.
- ☐ Cart with line selection, discount, totals.
- ☐ Numpad editing of qty/price/discount.
- ☐ Payment modal support change due.
- ☐ Order insertion respects RLS, triggers, snapshots.
- ☐ Offline sync queue for orders when offline.
- ☐ Analytics refresh via refresh\_daily\_rollups.

### Business Logic

- ☐ Business type selection accessible via Settings/Sell.
- ☐ Category filter, variant suggestions linked to business type.
- ☐ Settings location detection (Abdi Asis district).
- ☐ Chat conversation auto-created on product view (seller notified).
- ☐ Name change limited to 7 days.
- ☐ Product viewer events track to seller chat.
- ☐ Receipt settings apply per seller.

### Admin/Other

- ☐ Dashboard/Receipts page for sellers (RLS).
- ☐ POS inventory page.
- ☐ Chat and conversation listing (RLS).
- ☐ Monitoring logs communication.
- ☐ Build process passes (npm run build).

## Verification Actions

1. Walk through each page, mark checkboxes above for functioning items.
2. Use Supabase Studio to confirm table entries (profiles with coords, product\_views, orders).
3. Inspect migrations applied (supabase/migrations + config.toml referencing project).
4. Run POS flows: create order, pay, check receipts saved.
5. Enable display location and allow permission to test district detection.
6. Switch business type in Settings, verify ProductForm's category/variant filtering.
7. Validate chat creation when viewing another's product (see chat\_messages).
8. Confirm analytics rollups via refresh\_daily\_rollups function if accessible.

## Next Steps

- Tick the checklist in this document (convert to PDF to mark).
- Note sections needing repairs (e.g., chunk size warnings, dynamic import mix).
- Use this doc as the authoritative list when triaging bugs or new features.