

House of Vanya by Alice — Complete Application Capabilities Dossier

Document type: Certification-grade feature and capability reference
Audience: Reviewers, certification auditors, founders, engineering, operations
Last updated: Wave 24 (AI Saree Try-On, Measurements Wizard, Security Hardening)

This dossier enumerates **every** customer-facing feature, every admin tab, every sub-panel, and every AI/automation capability that ships in the House of Vanya by Alice atelier platform. For each item it explains **what** it does, **how** it does it, and **why** it exists.

1. Platform Overview

House of Vanya by Alice is a luxury Indian saree atelier delivered as a content-rich, AI-augmented e-commerce experience. The product strategy combines:

- **An editorial lookbook front of house** — collections curated by occasion, fabric, drape, capsule, and lookbook story rather than by SKU dumps.
- **An AI concierge (“Mika”)** — a Gemini-powered stylist who knows the catalogue, the customer’s profile, their measurements, their order history, and the entire feature map of the site.
- **A full operations back-of-house** — multi-warehouse inventory, RMA/refund engine, marketing/promotions composer, fraud monitoring, finance, dynamic pricing, churn prediction, supplier purchase orders, shipping labels, CMS, and enterprise multi-store registry.
- **Identity-locked AI media generation** — an approved-models registry plus a Replicate face-swap pipeline so the brand never publishes synthetic faces that drift from approved likenesses.
- **An AI Saree Try-On** — credit-gated, gender-gated, identity-preserving on-model rendering of any catalogue saree onto the shopper’s own photo.

1.1 Technology stack (what powers it)

Layer	Technology	Why
Frontend / SSR	TanStack Start v1 on React 19 + Vite 7	File-based routing, server functions, SEO-friendly SSR
Styling	Tailwind v4 + semantic tokens in src/styles.css	Editorial parchment/clay/gold

Layer	Technology	Why
		palette, Cormorant Garamond + Inter
Runtime	Cloudflare Workers (edge), nodejs_compat	Global latency, no cold infra
Data	Lovable Cloud (Supabase Postgres + Auth + Storage + Realtime)	RLS-secured, audited, with migrations under version control
AI	Lovable AI Gateway (Gemini 3 Flash, Gemini 2.5 Flash, Gemini 3.1 Flash Image)	Single billing, no vendor lock-in, low latency
Media transforms	Replicate (face-swap, identity-locked image generation)	Brand-safety guardrail
Payments	Razorpay / UPI / Stripe / Cash on Delivery	India-first plus international
Logistics	Shiprocket integration + in-house 4x6 label generator	Domestic and international parcel handoff
Notifications	WhatsApp Business + Email (Resend) + SMS + Web Push	Lifecycle and transactional

2. Customer-Facing Experience

Every route below lives under `src/routes/`. Authenticated routes live under `src/routes/_authenticated/` and are gated by the integration-managed route guard which redirects unauthenticated visitors to `/auth`.

2.1 Discovery surfaces

Route	What it does	How	Why
<code>/ (index)</code>	Editorial homepage with hero, occasion strip, recommended rail, continue-shopping rail, flash-sale countdown,	Server-rendered with loader data. Occasion filter strip sits on the parchment band beneath the hero so it never overlaps the model.	First impression must feel like an atelier, not a marketplace.

Route	What it does	How	Why
	weather-drape card, journal previews.		
/shop	Master catalogue with filter rail (occasion, fabric, drape, price band, Bayesian min-rating, min-discount, eco).	Behaviour-driven discovery (products.function s.ts) ranks by recency-weighted views, conversions, and rating priors.	Replaces vanity “newest first” with relevance.
/collections	Curated collection landing pages.	Static SEO content + CMS overlay.	SEO and storytelling per collection.
/occasion/\$slug, /fabric/\$slug, /drapes/\$slug, /drapes	Filtered facets with editorial intros and drape atlas guides.	Slug routes resolve to filtered shop queries with hand-written copy.	Long-tail SEO + concierge education.
/lookbook, /lookbook/\$slug	Photographic lookbook stories with shoppable product overlays.	Block-based content from lookbooks.function s.ts.	Drives aspirational discovery and bundle revenue.
/capsule	Seasonal capsule wardrobe view.	Curated subset of catalogue.	Concentrates merchandising for launches.
/atelier, /story	Brand narrative pages.	Static content.	Trust + provenance.
/journal, /journal/\$slug	Long-form editorial articles.	Markdown-backed posts.	SEO + thought leadership.
/celebrity-looks	Rail of celebrity drape inspirations with shoppable matches.	Curated rail + product joins.	Aspirational pull-through to PDP.
/for-you	Personalised “fashion feed” of products and looks.	Per-user ranking using profile, RFM segment, recently viewed, wishlist signals.	Personalisation flywheel.
/visual-search	Upload an image,	Embedding-	Removes the

Route	What it does	How	Why
	find similar sarees.	based similarity over catalogue vectors.	keyword barrier for fashion search.
/live, /live/\$slug	Live-shopping pages.	Reels and live sessions managed in admin.	Social commerce.
/with/\$influencer	Influencer storefronts with curated edits.	influencer_storefronts table + creator profile.	Creator-led commerce.
/community/looks	UGC outfit-sharing feed (moderated).	community.functions.ts + admin UGC moderation.	Social proof + community.
/compare	Side-by-side product comparison.	Local state + product joins.	Reduces decision friction on PDP.
/reviews	Aggregated review feed, including video reviews.	reviews.functions.ts.	Trust and social proof.
/api-docs	Developer-facing API documentation page.	Static / OpenAPI-style.	Enterprise integration story.

2.2 Product detail and purchase path

Route	Capability	How / Why
/product/\$id	Full PDP: gallery, 360° spin viewer (Spin360.tsx), fabric/drape/measurement spec, fit guidance, Bayesian “Top rated” badge, Frequently Bought Together, “Complete the look” rail, gift wrap option, video reviews, share, wishlist, save-for-later, “Try this saree with AI” launcher, alerts (back-in-stock, price drop).	All of these are PDP modules wired to dedicated server functions. The Try-On launcher gates on settings.tryon.enabled and the shopper’s credit wallet.

Route	Capability	How / Why
/cart	Cart with quantity, gift-wrap toggle, save-for-later, applied coupon preview, recommended add-ons.	Persistent via cart.tsx (localStorage for guests, server for signed-in).
/checkout	Address with OpenStreetMap map picker (AddressMapPicker.tsx), saved addresses, shipping method, payment method (Razorpay / UPI / Stripe / COD), gift wrap, order modification window. COD toggle is controlled centrally via SiteSettings.	Single page with progressive disclosure.
/checkout/success	Order confirmation, WhatsApp/email/SMS notifications, recommended next purchases.	Triggered by notify-dispatch.server.ts.
/subscribe	Email/SMS subscription opt-in with consent capture.	Compliance trail logged.
/wishlist/shared	Public shareable wishlist URL.	Token-scoped link; recipient sees curated picks.
/contact, /faq, /policies/*	Support, FAQ, shipping/returns/privacy/terms/international policy pages.	Content-managed via admin FAQs and policy pages.

2.3 AI features (shopper side)

Feature	What	How	Why
Mika AI Stylist (MikaChat.tsx)	Floating chat concierge that recommends sarees, books appointments, answers policy questions, looks	Server fn mika.functions.ts calls Gemini 3 Flash via Lovable AI Gateway. System prompt injects shopper	One concierge that knows the whole atelier — no dead-end “I can’t help with that”.

Feature	What	How	Why
	up orders, and deep-links into any feature using <code>[[link:/path Label]]</code> tappable chips.	measurements, RFM tier, saved addresses, last order, wishlist, locale, and full feature map. Supports English, Hindi, Kannada, Telugu.	
AI Saree Try-On (/try-on/plans, /try-on/\$productId)	Upload your photo, pick a saree, get a draped render with your face/skin/body preserved.	tryon.functions.ts runs a Gemini 2.5 Flash gender-classifier gate (rejects males with a polite, non-pejorative message; rejects no-person uploads). On pass, calls gemini-3.1-flash-image with an identity-preservation system prompt. Stores uploads in a private per-user Supabase bucket. Deducts credits only on success. Plans: Basic (5), Standard (15), Premium (30).	Removes the “will it look right on me?” objection without inventory cost. Female-only because the catalogue is women’s sarees.
AR Mirror (/ar-try-on)	Live webcam AR overlay for drape positioning.	WebGL + pose estimation client-side.	Real-time fit perception.
AI Outfit Critique + Blouse-Size Prediction	Upload an outfit, get a stylist critique; or get a Gemini-predicted blouse size from measurements.	ai-fashion.functions.ts.	Mika-grade advice on demand.

Feature	What	How	Why
Weather-Based Drape Card	Homepage card suggesting drape and fabric for today's weather at the shopper's pincode.	Open-Meteo API + drape-matcher mapping.	Contextual relevance.
AI Visual Search	Upload a photo or paste a link to find visually similar sarees.	Image embeddings + cosine similarity over catalogue.	Inspiration → SKU in one tap.
Identity-Locked Generation	Marketing imagery uses only approved models from APPROVED_MODEL S registry, optionally face-swapped via Replicate.	face-swap.functions.ts, approved-models.ts.	Brand-safety + likeness consent.

2.4 Authenticated account hub (/_authenticated/account.*)

Route	Capability
account.profile	Name, contact, language, currency, marketing consents, social linkage (Google, Facebook, Apple, Mobile OTP).
account.measurements	3-step guided wizard (Essentials → Blouse → Petticoat) with validated ranges and progress tracking. Saved measurements auto-inject into Mika's system prompt.
account.addresses	Address book with default selection and map-picker support.
account.orders, account.orders.\$id	Order list and detail with timeline, invoice PDF, modification window, and re-order.
account.exchanges	Self-service returns & exchanges via ReturnRequestPanel.tsx — supports refund or exchange, photo upload, reason taxonomy, and pickup scheduling.
account.payments	Saved payment instruments.

Route	Capability
account.subscriptions	Subscription plans (incl. Try-On credit packs).
account.wishlist	Multi-wishlist groups with sharing.
account.alerts	Back-in-stock and price-drop alerts.
account.rewards	Loyalty tier, points balance, cashback ledger (2% auto-accrual).
account.referrals	Referral codes, status, payouts.
account.appointments	Booked stylist appointments.

3. Admin Console (/_authenticated/admin.*)

The admin app is gated by RBAC (has_role security-definer function over the user_roles table; absolutely no role flags on profiles). Sensitive vault access requires WebAuthn (Fingerprint Vault). Every write is audit-logged.

The console is organised by domain. Below is every tab, every sub-panel, every section.

3.1 Admin home — admin.index

- **Executive dashboard** with revenue, AOV, CR, GMV, refund ratio, RFM mix.
- **5-stage conversion funnel chart** (View → PDP → Cart → Checkout → Paid).
- **New approved models queue** — surfaces newly-added entries from APPROVED_MODELS for moderation/use.
- Quick links to every sub-tab.

3.2 Catalogue & content

Tab	Sub-sections	What it does
admin.products	Product list, create/edit drawer, bulk CSV import, variant editor, media uploader, SEO meta, eco/sustainability flags.	Master catalogue CRUD.
admin.inventory	Per-SKU stock, low-stock thresholds, reserve counts.	Stock truth across warehouses.
admin.warehouses	Warehouse CRUD with “Share pool” toggle for shared vs siloed inventory; address;	Multi-warehouse fulfilment.

Tab	Sub-sections	What it does
admin.suppliers	default-for-region. Supplier directory + purchase-order composer that syncs received stock into inventory.	Procurement & PO lifecycle.
admin.media	Media library, image uploader, alt-text editor.	Asset governance.
admin.lookbooks	Lookbook story builder (blocks, products, hotspots).	Editorial content.
admin.hero	Homepage hero CMS (image, copy, CTA).	Top-of-funnel control.
admin.drape-cms, admin.drape-images	Drape atlas content + drape image library.	Drape education.
admin.landing-pages	Block-based landing page CMS producing /p/\$slug routes.	Campaign landing pages.
admin.faqs	FAQ CRUD by category.	Self-serve support.
admin.email-templates, admin.sms-templates	Template editors with merge tokens and preview.	Lifecycle messaging.
admin.approved-models	Approved-models registry with new-model intake queue (e.g. new-model-01..10).	Identity-locked AI media.
admin.face-swap (root-level admin.face-swap.tsx)	Replicate face-swap workflow.	Brand-safe model imagery.
admin.drape-images	Drape reference imagery for matchers.	Powers Mika's drape suggestions.

3.3 Orders & fulfilment

Tab	What it does
admin.orders	Order list, filters, detail drawer, status transitions, refund triggers, fulfilment notes.
admin.returns	RMA queue with refund/exchange decisions, atomic concurrency guards, pickup scheduling.

Tab	What it does
admin.refund-rejections	Audit of rejected refunds with reason codes.
admin.shipping	Carrier configuration, zone rules, rates.
admin.labels	In-house 4×6 label generator for packing slips + Shiprocket PDF side-panel.
admin.payment-events	Webhook event log for Razorpay/Stripe with reconciliation status.

3.4 Customers & lifecycle

Tab	What it does
admin.customers, admin.customers_.\$id	Customer list + 360° profile (orders, RFM tier, lifetime value, segments, support tickets).
admin.segments	Customer segment builder (RFM, behaviour, custom predicates) with segment-scoped CSV export.
admin.loyalty	Loyalty tier rules, points multipliers, redemption.
admin.ltv	LTV cohort views.
admin.churn	Churn prediction dashboard with at-risk segments.
admin.coupons	Coupon CRUD (code, stacking rules, eligibility).
admin.promotions	Promotions composer producing email/SMS/push campaigns.
admin.campaign-roi	Campaign ROI dashboard (cost, revenue, ROAS, attributed orders).
admin.notifications	Notification log + manual broadcast.
admin.dynamic-pricing	Velocity-vs-stock dynamic pricing rules.

3.5 AI, intelligence & analytics

Tab	What it does
admin.analytics	KPI dashboards (revenue, traffic, conversion, retention).
admin.category-performance	Per-category performance with flash-sale impact.

Tab	What it does
admin.forecasts	Demand forecasting per SKU.
admin.insights (via insights.functions.ts)	AI-generated insights (“Why did velvet spike this week?”).
admin.mika	Mika conversation review, intent quality, escalation queue.
admin.llm	Encrypted LLM vault dashboard (AES-256-GCM at rest, deny-all RLS except service_role) for managing model keys, prompt versions, eval runs. WebAuthn-gated.
admin.tryon-plans	AI Saree Try-On plan manager: CRUD on Basic/Standard/Premium credit packs, subscription oversight, per-user usage tracking.
admin.sustainability	Eco-score signals per SKU and aggregate ESG view.

3.6 Marketing, social & community

Tab	What it does
admin.reels	Reel/short-video pipeline with preset + QC checks (reels-qc.ts).
admin.live	Live-shopping session scheduler.
admin.influencers	Influencer roster, payouts, attribution.
admin.ugc	UGC moderation queue for community looks.
admin.reviews	Review moderation (text + video).
admin.stylists	Stylist roster + availability.
admin.appointments	Appointment calendar across stylists.
admin.support	Live support chat handoff queue from Mika.

3.7 Governance, security & enterprise

Tab	What it does
admin.settings	Master SiteSettings editor: currency, COD enabled, early-sale access, VIP support, Try-On enabled, gift-wrap defaults, VAT/GST per locale.
admin.roles	Role assignment via user_roles table.

Tab	What it does
admin.owners	Workspace owner management.
admin.secrets	Vault-managed secret editor (WebAuthn-gated).
admin.security	Security posture dashboard, audit log viewer.
admin.fraud	Weighted fraud-risk scoring dashboard (velocity, geo mismatch, device, payment failures).
admin.activity	Cross-system audit feed.
admin.feature-flags	Feature-flag CRUD with useFeatureFlag client hook.
admin.enterprise	Multi-store registry + POS / ERP / CRM integration endpoints.

4. AI Saree Try-On — End-to-End

Why: Removes the largest objection in online saree purchase (“will it suit me?”) without inventory or photoshoot cost.

What it does: Renders the selected catalogue saree draped on the shopper’s own photo while preserving face, skin tone, and body shape.

How it works:

1. **Entry** — TryOnLauncher.tsx on PDP. Gated by settings.tryon.enabled and the shopper’s credit wallet.
2. **Plans** — /try-on/plans lists Basic (5 images), Standard (15), Premium (30) sold via purchaseTryonPlan (Razorpay/Stripe). Credits land in tryon_subscriptions.
3. **Flow** — /try-on/\$productId: upload → guideline review → disclaimer acknowledgement → generate.
4. **Gender gate** — Gemini 2.5 Flash classifier inspects the upload. If male, the request is rejected with code FEMALE_ONLY and a polite, non-pejorative message (“The AI saree try-on is tailored to the women’s saree drape... If this is a gift, please ask the recipient to upload her photo”). No credit consumed, image not stored. If no person is detected, code NO_PERSON with a helpful retry hint.
5. **Generation** — On pass, image is uploaded to a private tryon-uploads Supabase bucket (RLS scoped to auth.uid()), then sent to gemini-3.1-flash-image with an identity-preservation system prompt. Output is stored in private tryon-outputs bucket. Credit is decremented atomically only on success.

6. **History** — listMyTryonJobs and deleteMyTryonJob provide a private gallery with delete-and-purge.
 7. **Admin** — /admin/tryon-plans manages plans and per-user usage. Master toggle in /admin/settings.
-

5. Mika AI Stylist — Capability Map

- **Model:** google/gemini-3-flash-preview via Lovable AI Gateway (no per-tenant key needed).
 - **Context injection:** name, locale, RFM tier, saved measurements (bust/waist/height/blouse/petticoat), saved addresses, last order, wishlist, recent views.
 - **Capabilities (CAPABILITIES + FEATURE AWARENESS blocks in the system prompt):** recommend sarees by occasion/fabric/budget; explain drapes; advise length and blouse sizing from measurements; book stylist appointments; look up orders securely (getOrderForMika with auth guard); start a return; check delivery to pincode; surface flash sales; route to AR/AI try-on; explain policies; switch language (English, Hindi, Kannada, Telugu).
 - **Inline routing:** Mika emits [[link:/path|Label]] tokens; MikaChat.tsx parses them into tappable navigation chips so any feature is one tap away.
 - **Escalation:** /admin/support queue receives handoff requests with full transcript.
-

6. Commerce Mechanics

- **Cart:** persistent cart with gift wrap, save-for-later, FBT bundles, complete-the-look rail.
 - **Coupons:** stacking rules, customer-segment scoping, expiry, first-order-only flags.
 - **Loyalty + Cashback:** tier-based points multipliers; 2% cashback auto-accrual ledger (cashback.functions.ts).
 - **Referrals:** per-user referral code, payout ledger.
 - **Subscriptions:** Try-On credit packs and curated drops.
 - **Pricing:** dynamic pricing rules (velocity vs stock), multi-currency display, per-locale VAT + GST.
 - **Checkout:** Razorpay, UPI, Stripe, COD (toggleable centrally), OpenStreetMap address picker with pincode validation.
 - **Order modification:** time-boxed self-service modification window.
 - **Returns & exchanges:** self-service via ReturnRequestPanel.tsx, pickup scheduling, atomic concurrency guards, refund-rejection audit.
-

7. Logistics & Fulfilment

- **Multi-warehouse** with shared-pool toggle.
 - **Shiprocket** integration with PDF labels.
 - **In-house 4x6 label generator** for direct dispatch.
 - **Purchase orders** to suppliers with synced receipt-to-stock.
 - **Low-stock alerts** hourly cron (low-stock-alerts.ts).
 - **Delivery promise:** pincode-aware ETA on PDP and checkout (pincode.functions.ts).
-

8. Marketing, CMS & Content Ops

- **Landing-page builder** (/admin/landing-pages → /p/\$slug).
 - **Promotions composer** for email/SMS/push.
 - **Campaign ROI** dashboard.
 - **Lookbook + Hero CMS** with shoppable hotspots.
 - **Drape Atlas CMS** with image library.
 - **Reels & Live** with QC pipeline.
 - **Influencer storefronts** at /with/\$influencer.
 - **TikTok Shop CSV feed**, Instagram product catalogue feed.
-

9. Notifications

Channel	Triggers
WhatsApp Business	Order placed, shipped, delivered, return approved.
Email (Resend)	Order lifecycle, marketing campaigns, password reset, OTP fallback.
SMS	OTP, delivery updates.
Web Push	Back-in-stock, price drop, flash sale.

All dispatched via notify-dispatch.server.ts from server-only paths so credentials never touch the browser.

10. Personalisation & Intelligence

- **Behaviour-driven discovery** in products.functions.ts (recency-weighted views, conversions, rating priors → Bayesian “Top rated”).
- **/for-you** personalised fashion feed.
- **Recently viewed** rail (recently-viewed.ts).

- **RFM segmentation** in customer-segments.functions.ts.
 - **LTV cohorts + churn prediction + demand forecasts.**
 - **Dynamic pricing** rules (velocity vs stock).
 - **Fraud scoring** (weighted multi-signal).
 - **AI insights** (insights.functions.ts) — natural-language explanations of dashboard swings.
-

11. Security & Compliance

- **Auth:** Email, Google, Facebook, Apple, Mobile OTP. Social redirect_uri always to a same-origin public URL; post-session navigation handles the intended destination.
 - **RBAC:** user_roles table + security-definer has_role(). **No role flags on profiles** — eliminates privilege escalation.
 - **RLS:** Every public table has explicit grants and policies. The LLM vault (app_settings) is explicit deny-all for anon + authenticated, service-role only. Anon-insertable tables (product_alerts, push_subscriptions) have tightened WITH CHECK constraints (enum/length guards, user_id IS NULL anti-impersonation).
 - **Vault:** AES-256-GCM at rest. **WebAuthn Fingerprint Vault** for unlocking secret panels in admin.
 - **Audit:** audit.functions.ts + audit.server.ts log every privileged action.
 - **Payments:** Razorpay/Stripe webhooks verified by signature in /api/public/* routes.
 - **PII:** Try-On uploads stored in private per-user buckets with RLS scoped to auth.uid().
 - **Gender gate** for AI Try-On treated as a sensitive-content control (Certification control **DP-5**).
-

12. Documentation & Certification Pack

- docs/product/feature-matrix-world-class.md — feature matrix and gap register.
 - docs/product/mika-chatbot-flow.md — Mika capability map, including Wave 24 routing tokens.
 - docs/release-notes/2026-roadmap-wave-24.md — AI Try-On + gender gate + measurements wizard + security hardening.
 - docs/diagrams/certification/ai-tryon-flow.mmd — sequence diagram.
 - docs/certification/control-matrix.md — controls **AI-1..AI-4** (Try-On safeguards), **DP-5** (sensitive content gate), **DV-1..DV-2** (vault/anon-path hardening).
 - docs/certification/README.md — certification index.
-

13. Roadmap Status Snapshot

- **Waves 20–24 shipped.** Feature matrix scorecard $\geq 90\%$; all Wave 24 rows ☐.
 - **Active gap register** (52 originally identified; closed across Waves 22–24 with remaining items intentionally N/A).
 - **TypeScript:** tsgo --noEmit exit 0.
 - **Supabase linter:** 17 → 13 findings; remainder are intentional SECURITY DEFINER helpers and the platform-default pg_trgm extension, all documented in security memory.
-

End of dossier.