

HitByGlam | Technical Handover

A buyer-safe technical handover covering the application package, buyer-owned services, deployment, verification, security, and maintenance.

Product and system overview

HitByGlam is a single-page web application with public discovery pages, authenticated client and professional workspaces, a separate owner area, real-time data, private files, subscriptions, and featured-placement payments.

The transferred application package uses React and TypeScript for the browser application, Vite for build output, Convex for application data and server functions, Clerk for member login, and Stripe for professional subscriptions and featured visibility. The buyer should create and control every production account used after transfer.

Project structure

- src/: public routes, client workspace, professional workspace, owner interface, shared components, product configuration, and styling.
- convex/: data schema, queries, mutations, server actions, authorization, billing, seed logic, owner operations, and smart helpers.
- public/: route fallback files, brand assets, fonts, icons, and other static files.
- tests/: focused checks for explainable matching and workflow helpers.
- docs/: product requirements and implementation evidence used during handover.

Important product boundaries

- Stripe is used for HitByGlam professional plans and labelled featured placement.
- Client appointments, contracts, deposits, service payments, payouts, escrow, and wallets are outside the current product.
- Availability is a request signal and not a guaranteed appointment calendar.
- Smart matching is explainable and deterministic. The product does not require generative AI, facial analysis, health inference, or image-based suitability decisions.

Buyer-owned services and configuration

Create fresh buyer-owned accounts for hosting, member login, application data, payments, email, and storage. Do not reuse seller credentials in the live environment.

- Browser configuration: VITE_CONVEX_URL and VITE_CLERK_PUBLISHABLE_KEY.
- Server-only configuration: CLERK_SIGNING_KEY, STRIPE_SECRET_KEY, STRIPE_WEBHOOK_SECRET, SITE_URL, APP_ID, and APP_NAME.
- Owner access: transfer the initial owner login privately, then change it immediately after first access.
- Payments: connect buyer-owned products, prices, webhook endpoint, and billing portal settings before accepting live purchases.

Important: Use obvious placeholders in setup files. Keep every reusable secret in the server-side deployment environment and never in browser variables, public files, or documentation.

Deployment sequence

- 1 Open the transferred project folder and use the included pnpm lockfile.
- 2 Create an ignored local environment file from the provided example and add only public browser values.
- 3 Configure server-only values in the buyer-owned backend environment.
- 4 Deploy the backend functions and confirm the production site URL.
- 5 Run the owner-controlled seed only against the intended buyer environment.
- 6 Run the focused smart-feature tests, type check, production build, and direct-route checks.
- 7 Publish the static output to the buyer's hosting account with route fallback enabled.
- 8 Complete the functional checks below before inviting real users.

Functional verification

- Member registration, client onboarding, professional onboarding, and owner login.
- Professional submission, owner approval, public profile visibility, portfolio, services, packages, and availability signals.
- Inspiration boards, Style Brief creation, explained matching, invitations, proposals, revisions, messages, and outcomes.
- Professional plans, featured-placement checkout, signed webhook processing, entitlements, and billing portal access.
- Private file access, public-field privacy, reports, reviews, data export and deletion, rate limits, and owner audit records.
- Responsive layouts, keyboard focus, reduced motion, image text alternatives, and direct URL refreshes.

Security, backups, and maintenance

- Rotate every shared credential, payment setting, webhook secret, owner credential, and deployment token after transfer.
- Keep payment webhooks signature-verified and idempotent.
- Review permissions for public profiles, private briefs, conversations, proposals, verification files, and owner sessions.
- Review upload types and limits, rate limits, sanitization, retention, export, deletion, and backup procedures before launch.
- Monitor failed payments, webhook failures, reports, storage use, privacy requests, and owner audit activity.
- Keep service dependencies and browser packages updated through a tested maintenance process.

Transfer and support boundary

The transfer includes the application package, domain, buyer guide, technical handover guide, and agreed setup assistance. Codeblix can provide up to 30 days of reasonable technical handover support.

The buyer owns deployment, service accounts, credential security, backups, monitoring, legal review, marketing, professional recruitment, customer support, third-party fees, and ongoing maintenance after handover.