

ATLAZ: What the investment built

Scope additions, implemented capabilities, and a market-based rebuild estimate

Prepared September 25, 2026 | Based on the project plan and quote dated December 1, 2025, the supplied source repository, and the owner’s implementation and QA statements

Executive summary

The ATLAZ codebase now contains a multi-part apparel discovery application: a responsive storefront, natural-language and semantic search, image upload and OpenAI vision analysis, hybrid retrieval, adjustable search controls, user accounts, saved ratings and feedback, product quality tools, and admin workflows. Several implementation choices extend or deepen the original MVP description, particularly the operator-tunable search stack, multi-level query refinement, computer-vision product enrichment, and community-aware rating logic.

Replacement-cost perspective For a professional to recreate the implemented breadth from the stated project baseline, this report uses a planning range of 160-260 specialist hours. At a modeled independent-specialist rate of \$50-\$100/hour, that is \$8,000-\$26,000. At a modeled agency rate of \$100-\$150/hour, that is \$16,000-\$39,000. The agency comparison is about 5.3-13 times the \$3,000 investment listed in the quote’s executive summary and payment schedule.

This is an estimated replacement effort and market-cost comparison, not a time sheet, formal bid, valuation of intellectual property, or assertion that every feature is defect-free. Known limitations are identified below. AI-assisted development can reduce the builder’s time; it does not make an equivalent system cheaper for a client to commission and integrate from scratch.

Where implementation goes beyond the baseline

The original plan already included semantic search, responsive UI, Google sign-in and preferences, user feedback, image search, hybrid image-plus-text search, quality review, and investor demo support. Those are not presented as scope additions simply because substantial implementation exists. The table separates genuine additions from richer implementations and substitutions. In-scope work such as caching and pagination is included in the overall replacement estimate, but is not claimed as an extra:

Implemented capabilities beyond the baseline

Capability	What exists	Why it adds value / evidence
Operator-tunable retrieval and relevance controls	Admin settings and the search layer expose automatic/hybrid/vector modes, vector/text weighting, similarity thresholds, category and product-quality filters.	Incremental operating control beyond the proposal's search/performance outcomes. Evidence: app/admin/settings/page.tsx; lib/search.ts; supabase-migration.sql.
Multi-level natural-language refinement	Users can refine an existing result set through successive intent parsing, semantic re-ranking, and a breadcrumb/history flow (up to three levels).	Distinct refinement implementation; it is not the specified "More Like This" product-to-product recommendation. Treat it as a different implementation. Evidence: components/RefinementBox.tsx; components/RefinementBreadcrumb.tsx; lib/search.ts; src/hooks/useRefinementHistory.ts.
Reusable computer-vision catalog enrichment	In addition to the in-scope upload analysis, scripts and schema support verified product colors, image descriptions, and image-description embeddings for catalog items.	Additional reusable product-enrichment work beyond analyzing the shopper's uploaded image. Evidence: app/api/admin/analyze-images/route.ts; scripts/migrations/2026-05-add-image-description.sql; scripts/migrations/2026-05-update-hybrid-match-products-v2.sql; lib/search.ts.
Richer relevance feedback and community safeguards	The in-scope binary vote concept is implemented as persisted 1-5 star ratings. Low ratings can open a text-feedback prompt; search can apply individual rating preferences and a community suppression rule.	Enhanced implementation of an in-scope requirement: a rating scale and aggregate suppression logic extend beyond thumbs up/down. Current code suppresses a product after at least 10 ratings when 51% or more are rated 2 stars or lower. Evidence: app/api/ratings/route.ts; app/api/ratings/stats/route.ts; components/ProductCard.tsx; lib/search.ts.
In-result product administration and catalog-quality tooling	Admin users can adjust a product's category and remove a product from its result card; the repo includes cleanup, classification, color, image, and import-maintenance scripts.	Operator tooling that extends the proposal's data-quality review into product-level administration and repeatable maintenance. Evidence: components/ProductCard.tsx; app/api/admin/products/[id]/route.ts; scripts/cleanup-nonfashion-products.mjs; scripts/cleanup-duplicate-products.sql.

Scope distinction These are codebase-backed capabilities. The report does not assign extra value to affiliate partnerships: vendor relationship acquisition and approvals remain client-managed, as stated for this project.

Delivered system breadth and effort model

The estimate below models the effort a replacement developer or small team would need to reproduce a system of similar breadth from the project’s stated baseline. It counts integrated workstreams once and is deliberately expressed as a range.

Workstream	Hours	Included examples
Application foundation, storefront, responsive behavior, integration	45-70	Product grid, upload/search experience, core API/UI integration, responsive behavior and application wiring.
Search and query refinement	30-45	Natural-language/semantic search, multi-step refinement, ranking, filtering, result handling and pagination.
Admin controls and catalog quality	20-35	Search tuning, product moderation, category controls, enrichment/cleanup tooling and database integration.
Identity, preferences and feedback	20-35	Google authentication integration, preference schema, persisted ratings, feedback capture and filtering behavior.
Vision analysis and multimodal search	20-35	Upload handling, vision API integration, product image enrichment and hybrid retrieval implementation. Known live errors limit the completion claim.
Caching, QA, handoff and demonstration materials	25-40	Cache integration, cross-browser/mobile checks as owner-reported, integration stabilization, documentation and demo support.
Modeled replacement effort	160-260	Approximate person-hours; workstream estimates are not recorded historical labor.

Market rate references and modeled cost

Upwork’s 2026 published directional ranges show software developers at \$10-\$100/hour, AI engineers at \$35-\$60/hour, and specialized development/AI/consulting work often at \$75-\$150+ per hour. Current Clutch Chicago software-agency listings include examples at \$100-\$149/hour and \$150-\$199/hour. These are advertised benchmarks and listings, not guaranteed quotes; actual rates vary by location, experience, team composition, and scope.

Comparison basis	Estimated amount	Calculation
Experienced independent specialist (modeled \$50-\$100/hour)	\$8,000-\$26,000	160-260 hours × modeled rate
Small software agency (modeled \$100-\$150/hour)	\$16,000-\$39,000	160-260 hours × modeled rate
Quoted project investment	\$3,000	Executive summary and payment schedule; three \$1,000 milestones

Agency-rate comparison On these assumptions, the \$3,000 investment represents roughly 8%-19% of the modeled agency replacement cost, or about \$13,000-\$36,000 below that range. This arithmetic compares project price with estimated replacement cost; it does not establish an invoice entitlement or guarantee a third-party bid.

Completion boundaries and client-owned dependencies

Area	Status / boundary
Visual search reliability	Upload and vision-analysis has been implemented.
Performance targets	There is sub-5-second processing for single images below 200 KB. The under-2-second text-search target cannot be substantiated across the deployed service tiers without moving the application to a more powerful dedicated server or VPS. Hosting and database plan upgrades are identified as dependencies.
Affiliate relationships	Vendor recruitment, approvals, and active relationship counts are client-managed. The developer replacement-cost estimate excludes vendor acquisition.
Data freshness	Catalog quality controls and cleanup are present; scheduled synchronization is not currently in place. Discontinued items can therefore leave stale links. Syncing products on a schedule requires vendor relationships.
Phase 1 distinctions	The semantic refinement flow is not the specified “More Like This” recommendation feature but allows users to narrow search results down and be much more selective, still using only natural language.
QA basis	Browser/device coverage is based on manual testing Chrome, Firefox, Edge, Brave and DuckDuckGo on desktop and Chrome/Safari on iPhone 12 Pro.

Pricing basis and interpretation

- The proposal estimates 80-100 total development hours and lists a \$3,000 total in both its executive summary and payment schedule. That equates to about \$30-\$37.50 per proposal-estimated hour before accounting for features beyond the plan.
- The rebuild estimate values the breadth of implemented functionality at public market rates. It does not claim the code was manually written line by line, certify production readiness, price unfinished fixes as completed, or replace formal procurement quotes.
- The estimate excludes third-party hosting/API costs, ongoing maintenance, formal security certification, and client-side affiliate relationship development. The proposal itself excludes third-party operating costs and future maintenance beyond its stated support period.

Selected source references

- Project scope: “Natural Language Apparel Shopping App MVP Development Project Quote,” December 1, 2025, supplied PDF; see Phase 1, Phase 2, payment schedule, exclusions, and conclusion.
- Code evidence locations are named beside each capability above and refer to the supplied repository snapshot. Source files demonstrate implementation structure, not live deployment state or end-to-end pass results.
- Upwork, “Hourly Rates: Average Rates by Skill and Experience in 2026” (published May 12, 2026): <https://www.upwork.com/resources/upwork-hourly-rates/>
- Upwork, “How Much Do Freelancers Make in 2026?": <https://www.upwork.com/resources/how-much-do-freelancers-make>
- Clutch, “Top Software Developers in Chicago” (listing page updated September 20, 2026): <https://clutch.co/developers/chicago>