

ATLAZ Phase 2 Delivery Status

Phase 2 technical evidence report | September 25, 2026

Prepared by Scot Robnett, SDR Productions LLC | Prepared for Sheeva Rajaei

Executive Summary

The project source confirms that the image upload interface, drag-and-drop handling, OpenAI image analysis, and image-plus-text query construction have been implemented. The upload interface and core search flow are present in the current code snapshot. The under-two-second text-search target cannot be demonstrated without more powerful dedicated or VPS hosting, and the reported under-five-second image result covers single image below 200 KB; to achieve this with very large images or multiple images requires paid CDN hosting such as Cloudinary.

Affiliate relationship development and approvals are managed by Sheeva Rajaei and not the responsibility of the developer. SDR's technical work provides the platform's product-feed integration points. Sourcing and maintaining a specified number of active vendor relationships is client-owned.

Phase 2 Status at a Glance

Phase 2 deliverable	Status
Image upload interface with drag-and-drop	Implemented in production
Computer-vision API integration	Implemented in production
Visual results combining image and text	Implemented in production
Affiliate vendor relationships	Client-owned
Text-search response under 2 seconds	Requires paid high speed hosting with powerful VPS or dedicated server; unachievable on free Netlify hosting
Image-search response under 5 seconds	Single images under 200KB return results in 4 seconds on average; to achieve this speed with very large images or multiple images, a paid image CDN hosting service like Cloudinary is required. Images are currently cached in Redis, which is what allows the 4 second < 200KB image search to complete in under 5 seconds.
Cross-browser compatibility	Tested successfully on desktop in Chrome, Edge, Firefox, Brave, and DuckDuckGo. Tested successfully on mobile in Chrome and Safari on iOS.
Mobile-responsive design	Implemented
Error handling and user messages	Present
Data quality review and cleanup	Materially improved
Demo script and presentation materials	Prepared and delivered May & September 2026

Implementation Evidence

Image upload and computer vision

The home page imports and renders the ImageUpload component in its image-search section. The component implements drag-enter, drag-over, drag-leave, and drop handlers, supports multiple files, validates file type and size, shows previews, and lets a shopper remove an uploaded image. The home page passes a five-image limit; the API accepts up to five images and up to 5 MB per file.

Evidence: components/ImageUpload.tsx:19-29, 79-120; app/page.tsx:425-459.

The visual-search API converts uploaded files to data URLs and sends them to OpenAI's gpt-4o-mini model for fashion description. When the shopper also enters text, the API combines that text with the image-derived description and submits the combined query to semantic search.

Evidence: app/api/search/visual/route.ts:79-120, 205-291; app/page.tsx:244-334.

Performance

The visual-search route records image-conversion, vision-analysis, and semantic-search durations in its server logs. The text-search route includes a one-hour Redis result cache when Redis is configured, and falls back to uncached search when it is not. The code does not include production timing records proving the two-second text target or the five-second target across the full supported image size and count; however, it has been made clear throughout the project that achieving this type of performance will require a dedicated server or VPS plus CDN image hosting like Cloudinary or similar services.

Evidence: app/api/search/visual/route.ts:145-185, 205-249, 261-299; app/api/search/route.ts:59-112, 123-151; lib/redis.ts:13-21, 63-96.

Single images under 200 KB each have returned results in under five seconds in manual testing even on the free current hosting at Netlify. No test log or benchmark record is available to establish the broader threshold due to the need for “more firepower” hardware-wise to achieve performance goals.

Browser support and mobile layout

The current application stylesheet includes responsive breakpoints for mobile, tablet, and desktop layouts, including product-grid and image-upload adjustments. Successful manual checks have been completed in Chrome, Firefox, Edge, Brave, and DuckDuckGo on desktop, and Chrome and Safari on an iPhone 12 Pro.

Evidence: app/globals.css:823-883, 1173-1278, 2051-2085; manual QA statement supplied September 25, 2026.

Data quality and search controls

The application includes admin-controlled search modes and vector/text weighting, plus configurable filters for men's products, low-priced listings, and non-apparel material. Search code also filters out men's products and raw fabric or swatch listings. Substantial catalog cleanup and corrected product image display have been implemented. Product data is not currently synchronized on a schedule and can become stale, including through discontinued product links. Current inventory accuracy and link freshness are therefore not certified by this source review. It has been known throughout the project that this is an MVP/prototype and that synching from affiliate vendors requires relationships with those affiliate vendors and APIs, some free and some paid.

Evidence: lib/search.ts:47-110, 493-543, 1613-1628, 1772-1826; app/admin/settings/page.tsx:342-396; developer's catalog-status statement supplied September 25, 2026.

Demo materials and pre-demo checks

The supplied ATLAZ Presenter Guide contains a shopper walkthrough, admin-settings walkthrough, exact query prompts, and a run-of-show. Investor presentation materials were sent in May 2026 and the demo script was sent September 23, 2026. The guide separates two curated fixture searches from live catalog searches and asks the presenter to verify the live query results before presenting. Its latest notes flag boot and tote catalog support for confirmation.

The provided source snapshot contains exact-phrase branches that read preloaded demo products. Before the demo, confirm which deployed queries use fixtures, which use live search, and that the guide describes the current deployment accurately. Treat fixture results as a demonstration flow, not as proof of live-catalog search quality.

Evidence: ATLAZ-PRESENTER-GUIDE.pdf, Parts 1-3 and pre-demo checklist; lib/search.ts:123-200, 564-586; app/api/search/route.ts:29-38.

Affiliate relationships

Affiliate network outreach, partner approval, and the active-vendor count are managed by Sheeva Rajaei. SDR's work is the technical integration of available product feeds. Sheeva controls whether the participating vendors are approved, active, and sufficiently stocked.

Phase 2 Evidence Position: Complete

This review demonstrates completion of the upload interface and the computer-vision integration as implemented in source. It also documents the hybrid-search implementation, responsive styling, error messages, search controls, and prepared demo materials. Phase 2 is complete with the exception of migration to more powerful dedicated hosting and image CDN and addressing technical errors (*not* visual or functional enhancements) Sheeva finds in her own QA process.