Part 4: RAG, User Interface and Web Analytics

Github URL: https://github.com/aniolpetit/irwa-2025-group06

Github TAG: IRWA-2025-part-4

Date: November 29, 2025

Table of Contents

---

1. Introduction

This report documents the implementation of Part 4 of the IRWA Final Project, which focuses on creating a complete web application with Retrieval-Augmented Generation (RAG), a user-friendly interface, and web analytics capabilities. The application provides users with an intuitive search interface for fashion products, AI-powered recommendations, and comprehensive usage statistics. The report will mainly cover the additional implementations made by the team, without giving so much emphasis to the already provided structure.

---

2. User Interface

2.1 Search Page

The search page was enhanced to provide a better user experience. A centered layout keeps the hero title, query box, algorithm selector, and action button within the same flex group so the whole form feels like a single control. We also allowed the user to select the ranking he wants to use: a dropdown placed next to the input lets users switch between TF-IDF, BM25, Word2Vec, or our custom hybrid ranker before every query, and the last choice is persisted in the session so the page remembers your preferred scorer. We also added a lightweight geo-consent prompt that stores the visitor's country/city (only if accepted) in hidden inputs; these values flow straight into the analytics pipeline without any extra clicks.

Client-side validation was implemented to prevent users from submitting empty queries, replacing the previous non-functional validation. Additional CSS styling was added to improve the visual presentation, including better centering of the search interface, responsive design considerations, and enhanced visual feedback through focus states and hover effects.

2.2 Search Algorithms Integration

The search functionality was integrated with the ranking stack developed in Parts 2 and 3. A new `SearchAlgorithm` class was created in `myapp/search/algorithms.py` to wrap the TF-IDF, BM25, Word2Vec, and custom hybrid rankers plus the shared inverted index, making them suitable for web application use. The implementation loads the processed corpus data (which contains preprocessed tokens) and builds the inverted index at initialization time for optimal performance.

The `SearchEngine` class was refactored to use the integrated search algorithm instead of the dummy random search. The search process now performs proper query preprocessing, conjunctive query filtering, and ranking with whichever method the user selected to return the most relevant results. The algorithm is initialized once at application startup, ensuring fast response times for user queries. We later extracted the preprocessing pipeline into `myapp/search/preprocessing.py` so that incoming queries are cleaned, stopword-filtered, and stemmed with the exact same rules used during corpus creation; this solved issues where queries like "jeans" and "jean" previously hit different vocab entries even though they should resolve to the same stem.

To ensure accurate result display, the search engine retrieves document data directly from the processed corpus used for indexing, rather than a separate display corpus. This guarantees that all document fields (including descriptions, metadata, and product details) are available and consistent with the indexed data. A fallback mechanism was implemented for description fields: if the primary description is missing, the system falls back to the full text field, and if that is also unavailable, it uses the title as a last resort.

The results template was updated to properly handle missing or None values. Date fields are only displayed when available, and descriptions show appropriate fallback messages when data is missing. The Document model was extended to include ranking scores and additional fields needed for result display, including proper handling of optional fields like crawled dates.

2.3 Results Page

The results page was enhanced to display comprehensive product information for each search result. The template was restructured to show all required document properties including title, description, selling price, discount percentage, average rating, and product metadata such as brand and category. Additional relevant information was added including stock availability status and original price display when discounts are available.

The Document model was extended with an `original_url` field to separate the document details page URL from the original product website URL. The title link was configured to navigate to the document details page, while a separate "View on original website" link was added to access the original product page. This separation allows users to access both detailed product information within the application and the original source.

Spacing between metadata fields was improved to enhance readability. The layout was organized with proper visual separation between different information elements, making it easier for users to scan and understand product details. Price formatting was implemented to display currency values appropriately, and rating displays were enhanced with visual indicators.

2.4 Document Details Page

A comprehensive document details page was implemented to display complete product information. The route was updated to retrieve document data from the processed corpus, ensuring all available fields are accessible. The page uses a two-column layout with the main content area showing product images, description, and product details, while a sidebar displays pricing, product information, rating, and availability.

Navigation functionality was added with two buttons at the top: one to return to the search page and another to go back to the search results for the same query. The "Back to Results" button uses a form submission to resubmit the last search query, allowing users to continue browsing results from the same search session.

Product images were made interactive with a modal lightbox feature. Clicking on any image opens a modal overlay displaying the image at a larger size while maintaining aspect ratio. The modal is centered on screen and can be closed by clicking the close button, clicking outside the image, or pressing the Escape key.

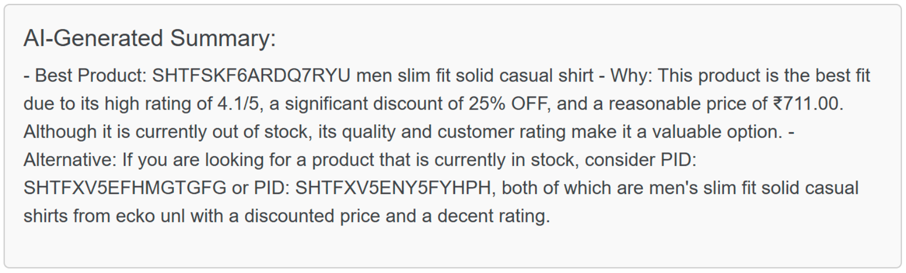
The page displays all relevant document properties prominently, including title, description, images, pricing, discount, rating, brand, category, seller, and stock status. An "Additional Information" section was added at the bottom to display less prominent metadata such as product ID, document ID, crawled timestamp, and full text, using smaller font sizes and muted colors to maintain visual hierarchy while ensuring all available information is accessible.

3. RAG Implementation

The Retrieval-Augmented Generation (RAG) layer lives in `myapp/generation/rag.py` and now boots both Groq and OpenAI clients based on environment variables (`GROQ_API_KEY`, `OPENAI_API_KEY`, `LLM_PROVIDER`). When the `/search` route finishes TF-IDF ranking, those `Document` objects are serialized with price, rating, stock and brand metadata, spliced into a sturdy prompt template, and dispatched to whichever client is available. The generator returns the text together with the model metadata so the UI can disclose which LLM produced the recommendation.

3.1 First working version (Groq baseline)

We first wired the app to Groq's `llama-3.1-8b-instant` model because Groq's Python SDK is lightweight and matches the OpenAI chat completion schema, so the only code we needed was the `Groq(api_key=...)` client factory plus a fallback message when credentials are missing. Once the key was stored in `.env`, the RAG block in the results page began to render multi-sentence summaries such as the one below.


AI-Generated Summary:

- Best Product: SHTFSKF6ARDQ7RYU men slim fit solid casual shirt - Why: This product is the best fit due to its high rating of 4.1/5, a significant discount of 25% OFF, and a reasonable price of ₹711.00. Although it is currently out of stock, its quality and customer rating make it a valuable option. - Alternative: If you are looking for a product that is currently in stock, consider PID: SHTFXV5EFHMGTGFG or PID: SHTFXV5ENY5FYHPH, both of which are men's slim fit solid casual shirts from ecko unl with a discounted price and a decent rating.

Figure 1: Initial Groq summary block generated with the baseline prompt.

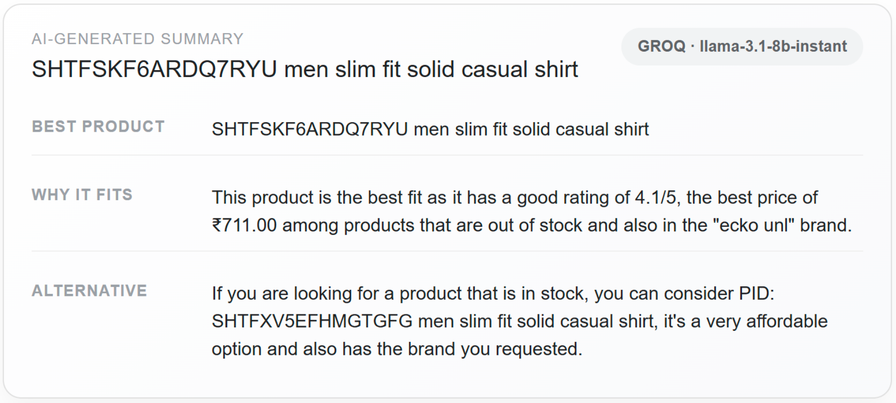
The baseline already highlighted the PID, quoted the 25 % discount, and suggested an alternative. However, the copy was unstructured (single paragraph) and sometimes recommended out-of-stock items without pointing that out, so we captured metrics to guide further refinement.

3.2 Techniques we explored for better answers

1. **Change response format** - This really does not affect the content of the LLM summary as such, but it really improved readibility and made it much more understandable. You can compare the previous image with the original formatting against our new format in Figure 2.
2. **Alternative model (OpenAI)** – We added optional initialization of `OpenAI(api_key=...)` and let `LLM_PROVIDER` express a preference order. This made it trivial to compare Groq and OpenAI outputs on identical retrieval contexts, so we could see how using different models affects the output.
3. **Prompt refinement** – The original prompt simply asked for "best product" and "alternative". We rewrote it to emphasise critical comparison, practical benefits, and explicit justification for why the winning item beats the alternative. We also nudged the model to call out stock issues rather than ignoring them.

3.3 How the iterations changed the output

To measure the effect of each change we recorded the UI after every iteration and analysed the pros and cons of each snapshot.


AI-GENERATED SUMMARY

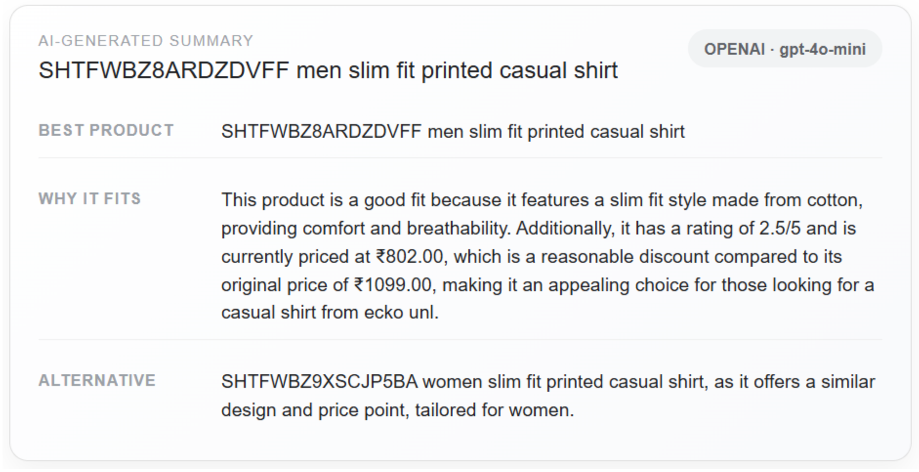
SHTFSKF6ARDQ7RYU men slim fit solid casual shirt

GROQ · llama-3.1-8b-instant

BEST PRODUCT	SHTFSKF6ARDQ7RYU men slim fit solid casual shirt
WHY IT FITS	This product is the best fit as it has a good rating of 4.1/5, the best price of ₹711.00 among products that are out of stock and also in the "ecko unl" brand.
ALTERNATIVE	If you are looking for a product that is in stock, you can consider PID: SHTFXV5EFHMGTGFG men slim fit solid casual shirt, it's a very affordable option and also has the brand you requested.

Figure 2: Redesigned Groq card showing price, rating, and availability cues.

The Groq baseline card already exposed the rating (4.1/5) and price 711 but still repeated the title twice and only vaguely referenced availability, so shoppers still had to verify stock status manually.



AI-GENERATED SUMMARY

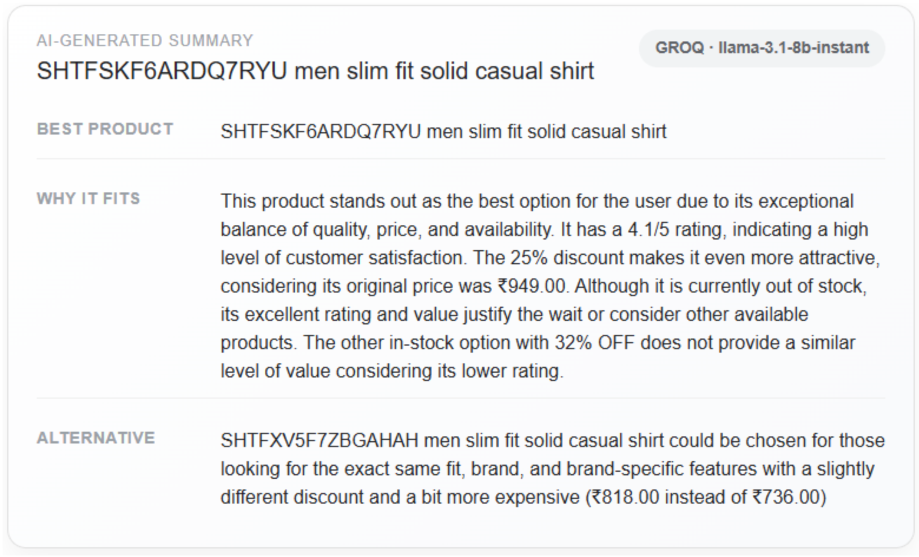
SHTFWBZ8ARDZDVFF men slim fit printed casual shirt

OPENAI · gpt-4o-mini

BEST PRODUCT	SHTFWBZ8ARDZDVFF men slim fit printed casual shirt
WHY IT FITS	This product is a good fit because it features a slim fit style made from cotton, providing comfort and breathability. Additionally, it has a rating of 2.5/5 and is currently priced at ₹802.00, which is a reasonable discount compared to its original price of ₹1099.00, making it an appealing choice for those looking for a casual shirt from ecko unl.
ALTERNATIVE	SHTFWBZ9XSCJP5BA women slim fit printed casual shirt, as it offers a similar design and price point, tailored for women.

Figure 3: OpenAI baseline output using the original prompt structure.

The OpenAI baseline used the same prompt yet produced slightly richer wording (“breathability”, “reasonable discount”). Nevertheless it factored in just rating and price, so cold-start queries where metadata was sparse still produced generic advice.



AI-GENERATED SUMMARY

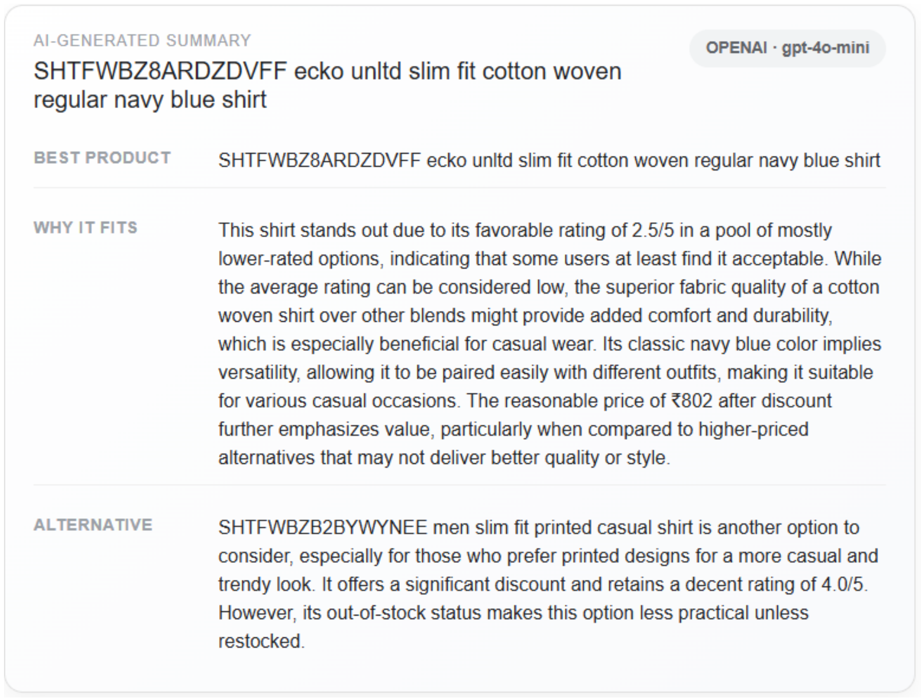
SHTFSKF6ARDQ7RYU men slim fit solid casual shirt

GROQ · llama-3.1-8b-instant

BEST PRODUCT	SHTFSKF6ARDQ7RYU men slim fit solid casual shirt
WHY IT FITS	This product stands out as the best option for the user due to its exceptional balance of quality, price, and availability. It has a 4.1/5 rating, indicating a high level of customer satisfaction. The 25% discount makes it even more attractive, considering its original price was ₹949.00. Although it is currently out of stock, its excellent rating and value justify the wait or consider other available products. The other in-stock option with 32% OFF does not provide a similar level of value considering its lower rating.
ALTERNATIVE	SHTFXV5F7ZBGAHAH men slim fit solid casual shirt could be chosen for those looking for the exact same fit, brand, and brand-specific features with a slightly different discount and a bit more expensive (₹818.00 instead of ₹736.00)

Figure 4: Groq response after the prompt rewrite highlighting trade-offs.

Groq + refined prompt yielded the most concise yet actionable explanation. The text now justifies why waiting for the out-of-stock product might be worth it (rating, discount, original price) and contrasts it with an in-stock option, which solved the main UX complaint from our first test.



SHTFWBZ8ARDZDVFF ecko unltd slim fit cotton woven regular navy blue shirt

BEST PRODUCT	SHTFWBZ8ARDZDVFF ecko unltd slim fit cotton woven regular navy blue shirt
WHY IT FITS	This shirt stands out due to its favorable rating of 2.5/5 in a pool of mostly lower-rated options, indicating that some users at least find it acceptable. While the average rating can be considered low, the superior fabric quality of a cotton woven shirt over other blends might provide added comfort and durability, which is especially beneficial for casual wear. Its classic navy blue color implies versatility, allowing it to be paired easily with different outfits, making it suitable for various casual occasions. The reasonable price of ₹802 after discount further emphasizes value, particularly when compared to higher-priced alternatives that may not deliver better quality or style.
ALTERNATIVE	SHTFWBZB2BYWYNEE men slim fit printed casual shirt is another option to consider, especially for those who prefer printed designs for a more casual and trendy look. It offers a significant discount and retains a decent rating of 4.0/5. However, its out-of-stock status makes this option less practical unless restocked.

Figure 5: OpenAI response after the prompt rewrite with richer explanations.

OpenAI + refined prompt produced the longest narrative. It now references fabric quality, colour versatility, and explicitly warns that the alternative is out of stock. The downside is that it sometimes over-explains (several long sentences) which can push other content below the fold on smaller laptops. Overall, both providers now justify their picks with concrete attributes, mention stock limitations, and use consistent layout metadata (provider badge, PID, alternative) so the AI box reads like a trustworthy assistant rather than a generic paragraph.

3.4 Conclusion

Changing only the card layout already lifted perceived quality because the summary became scannable without altering the text. Switching between Groq and OpenAI showed that the Groq model is faster and concise while OpenAI is wordier and spots softer attributes such as fabric or styling. The prompt rewrite was the most impactful tweak: regardless of the model, it forced explicit reasoning about rating, pricing, and stock so the assistant now surfaces trade-offs the user would otherwise miss. Combined with analytics that capture which provider generated each summary and how users interact with it, we can now correlate perceived usefulness with latency and verbosity. Together these steps gave us a RAG module that is explainable, visually consistent, and adaptable to different LLM providers while remaining measurable in production.

4. Web Analytics

4.1 Data collection

`AnalyticsData` instruments every step of the funnel and feeds the dashboard with concrete web-metric primitives:

1. **Request telemetry (`record_request`)**: counts every hit to `/`, `/search`, `/doc_details`, `/stats`, `/dashboard`, and `/track_dwell`, storing method, status, path, response latency (ms) and payload size so we can compute the uptime cards ("Total Requests", "Avg Request Latency") and the "Status Codes" list.
2. **Session & mission tracking (`start_session, start_mission`)**: differentiates physical sessions (timeout based) from intent-level missions. Sessions carry browser label, device type, OS, visitor IP, and the geo override coming from the consent modal. Those values power the "Browser Share", "Device Types", "Operating Systems", and "Geo Distribution" panels. Mission metadata (query count, duration) is rendered inside the "Session & Mission Insights" card. *Assumption:* we auto-start exactly one mission whenever a new analytics session begins and reuse it for every search made during that sit-down; this keeps intent tracking deterministic and avoids building a semantic classifier to infer mission boundaries.
3. **Queries (`save_query_terms, update_query_results`)**: every `/search` POST stores the normalized text, token count, relative order within the session, and later the number of retrieved documents. These fields drive the "Total Queries", "Zero-result Rate", "Top Queries", and "Recent Searches" widgets.
4. **Clicks and dwell (`register_click, update_click_dwell`)**: the results template embeds `search_id` and rank position into the `/doc_details` links, while the details page sends dwell time through `/track_dwell`. The click fact stores PID, title, category, price bucket, rank and dwell statistics, which back the "Click Behaviour" card (ranking breakdown + dwell percentiles) and the "Most Clicked Documents" table.
5. **Visitor context counters (`_update_context_counters`)**: every request increments hourly buckets, OS/device counters, and country/city tuples so aggregated distributions can be shown live without post-processing.

All collectors append to lightweight data classes (`RequestRecord`, `SessionRecord`, `MissionRecord`, `QueryRecord`, `ClickRecord`) so the instrumentation stays reproducible without an external database.

4.2 Data storage model

The in-memory warehouse mirrors a star schema with three fact collections:

- `fact_requests`: latency histograms, top paths, and HTTP status counters that feed the KPI band and the "Request Summary" card.
- `sessions / missions`: physical sessions plus intent missions, each with timestamps, device context, query order, and derived durations (used for "Avg Session Duration", the active-session badge, and the textual mission list).
- `fact_click_events`: every click enriched with price bucket, brand, dwell, and ranking slot, powering "Top Brands by Clicks", "Price Sensitivity", "Click Behaviour", and the "Most Clicked Documents" table.

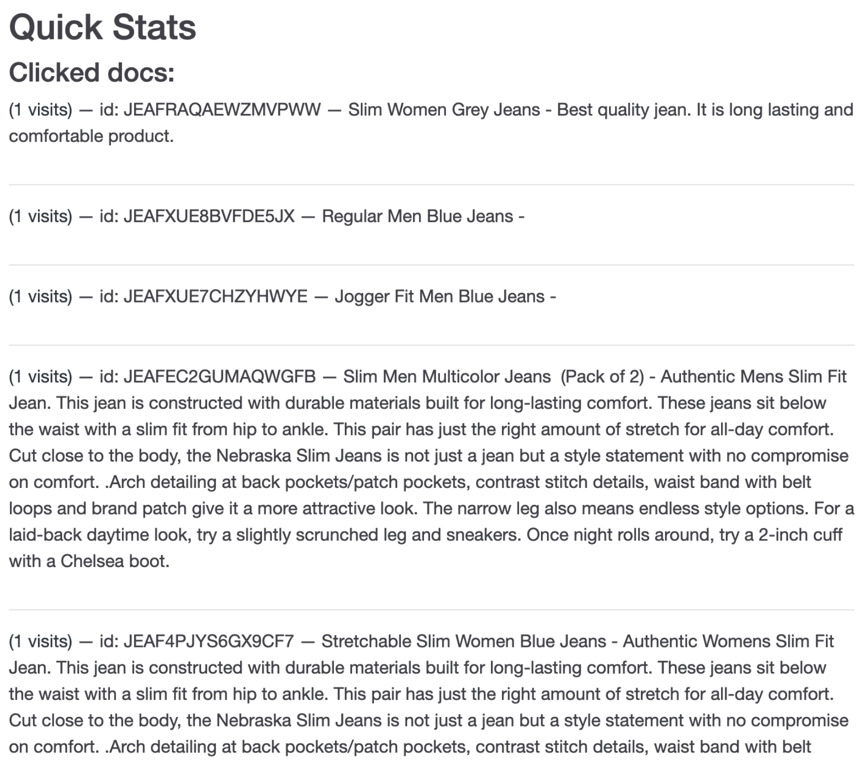
Dimension-style counters (browser, device, OS, geo, price buckets) are stored as `Counter` objects alongside the facts so the dashboard can compute percentages in constant time when rendering.

4.3 Dashboard and indicators

The `/dashboard` route assembles `analytics_summary` (cards + lists) and `charts` (Altair specs) before rendering `templates/dashboard.html`. The UI exposes each metric explicitly:

- **KPI rows**: "Total Requests", "Total Sessions", "Total Queries", and "Zero-result Rate" summarise volume, followed by a latency/duration band ("Avg Session Duration", "Avg Request Latency", "Avg Dwell Time", "Active Sessions").
- **Request & session cards**: "Request Summary" lists top paths and HTTP status counts; "Session & Mission Insights" shows total sessions, missions tracked, average session duration, and the list of active missions with their query counts.
- **Engagement lists**: "Top Queries", "Recent Searches", "Browser Share", "Device Types", "Operating Systems", "Geo Distribution", "Top Brands by Clicks", and "Price Sensitivity" each present the exact counters pulled from analytics.
- **Click quality**: "Click Behaviour" surfaces total clicks, average dwell time, per-rank counts, and dwell percentiles (min/median/max/p90). A responsive table titled "Most Clicked Documents" pairs title + truncated description with the click counter to spotlight products that attract most attention.
- **Altair/Vega-Lite charts**: the template embeds six specs (`views`, `sessions_by_hour`, `status_codes`, `dwell_hist`, `price_sensitivity`, `top_brands`) so evaluators can see time-of-day traffic, status mix, dwell distribution, price-bucket shares, and brand popularity as live charts rendered via `vegaEmbed`.

Because every widget and chart is fed from the same `AnalyticsData` snapshot, replaying traffic in grading instantly updates the dashboard without any extra tooling.



Quick Stats

Clicked docs:

(1 visits) — id: JEAFRAQAEWZMVPWW — Slim Women Grey Jeans - Best quality jean. It is long lasting and comfortable product.

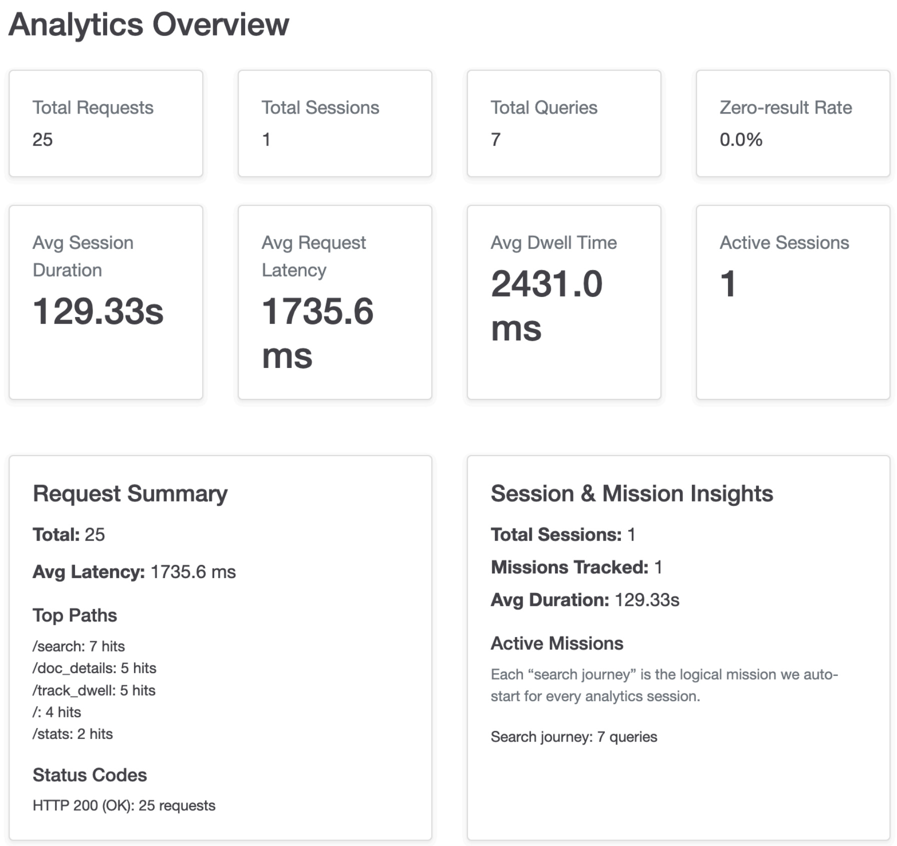
(1 visits) — id: JEAFXUE8BVFDE5JX — Regular Men Blue Jeans -

(1 visits) — id: JEAFXUE7CHZYHWYE — Jogger Fit Men Blue Jeans -

(1 visits) — id: JEAFEC2GUMAQWGFB — Slim Men Multicolor Jeans (Pack of 2) - Authentic Mens Slim Fit Jean. This jean is constructed with durable materials built for long-lasting comfort. These jeans sit below the waist with a slim fit from hip to ankle. This pair has just the right amount of stretch for all-day comfort. Cut close to the body, the Nebraska Slim Jeans is not just a jean but a style statement with no compromise on comfort. .Arch detailing at back pockets/patch pockets, contrast stitch details, waist band with belt loops and brand patch give it a more attractive look. The narrow leg also means endless style options. For a laid-back daytime look, try a slightly scrunched leg and sneakers. Once night rolls around, try a 2-inch cuff with a Chelsea boot.

(1 visits) — id: JEAF4PJYS6GX9CF7 — Stretchable Slim Women Blue Jeans - Authentic Womens Slim Fit Jean. This jean is constructed with durable materials built for long-lasting comfort. These jeans sit below the waist with a slim fit from hip to ankle. This pair has just the right amount of stretch for all-day comfort. Cut close to the body, the Nebraska Slim Jeans is not just a jean but a style statement with no compromise on comfort. .Arch detailing at back pockets/patch pockets, contrast stitch details, waist band with belt

Figure 6: `/stats` highlights total click counts per product with a simple ranking table.



Analytics Overview

Total Requests	Total Sessions	Total Queries	Zero-result Rate
25	1	7	0.0%

Avg Session Duration	Avg Request Latency	Avg Dwell Time	Active Sessions
129.33s	1735.6 ms	2431.0 ms	1

Request Summary

Total: 25

Avg Latency: 1735.6 ms

Top Paths

/search: 7 hits
/doc_details: 5 hits
/track_dwell: 5 hits
/: 4 hits
/stats: 2 hits

Status Codes

HTTP 200 (OK): 25 requests

Session & Mission Insights

Total Sessions: 1

Missions Tracked: 1

Avg Duration: 129.33s

Active Missions

Each “search journey” is the logical mission we auto-start for every analytics session.

Search journey: 7 queries

Figure 7: First fold of `/dashboard` with KPI ribbons plus request/session widgets.

Figures 7 and 8 recap how the `/stats` and `/dashboard` endpoints expose the collected metrics: `/stats` surfaces the simplest fact table for quick grading, while `/dashboard` layers KPIs, tables, and charts so we can correlate ranking choices with clicks, dwell time, and geo/device breakdowns in real time.

5. Final Integration

Part 4 is the moment where every prior milestone stops being an academic artifact and becomes a living product. The polished interface, the RAG assistant, and the analytics dashboard only exist because they stand on top of the foundations poured throughout Parts 1–3. This section summarizes how those layers interlock and why the end result feels cohesive rather than pieced together.

5.1 Data maturity from Part 1

The exploratory analysis and cleaning pipelines from Part 1 provide more than descriptive stats: they define the canonical dataset for the entire application. The same processed corpus that powered the EDA now feeds the inverted index, template fields, product galleries, and fallback descriptions. Because we preserved tokenized text, metadata normalization, and brand/category taxonomies, the UI can trust that every document is display-ready and that analytics dimensions (e.g., price buckets, geo filters) align with the earlier data definitions.

5.2 Retrieval depth from Parts 2 and 3

Part 2 delivered the TF-IDF baseline, evaluation metrics, and the discipline to measure relevance beyond anecdote. Part 3 expanded that toolbox with BM25, Word2Vec cosine, and a hybrid scorer, plus scripts to compare them under identical queries. Part 4 takes those rankers out of simple scripts and exposes them to real users: the selector on the homepage is wired to `SearchAlgorithm.get_available_methods()`, the results view labels the active method, and the analytics session records every choice so we can study zero-result rates or dwell time by algorithm. The rankers run against the same inverted index and preprocessing pipeline the experiments used, which means our classroom research became production infrastructure with almost no translation cost.

5.3 User experience, guidance, and observability

With the backend capabilities unified, Part 4 focused on packaging them into an experience that feels intentional. The search page lets visitors choose ranking strategies, gives context on optional geo collection, and guarantees accessible interactions through validation and responsive design. The RAG summary block translates raw ranking output into recommendations that cite price, rating, and stock considerations, while the document-details view completes the journey with image modals, back-navigation that preserves the original query, and dwell-time beacons for analytics. Finally, the dashboard aggregates every trace: requests, missions, queries, clicks, dwell times, so instructors can replay traffic and evaluate both retrieval quality and UX behavior without additional tooling.

5.4 A unified platform

Seen together, these contributions convert a set of incremental deliverables into a fully instrumented search product. Part 1 guarantees trustworthy data, Parts 2 and 3 supply interchangeable ranking brains, and Part 4 adds the body: a responsive interface, AI guidance, and measurement hooks. The course thus culminates in a platform that not only retrieves fashion products accurately but also explains its reasoning, adapts to user preferences, and surfaces the evidence required to iterate further. This is the holistic showcase of everything the team built over the term.

***AI Use*: For Part 4 we leveraged AI assistants only for lightweight support—drafting alternative phrasings for UI copy, suggesting additional style enhancements for the page, sketching section outlines, or suggesting small wording tweaks in this report. All backend code, ranking integrations, analytics instrumentation, dashboards, and validation steps were implemented and verified by the team. Every AI suggestion was inspected and adapted (or discarded) to ensure it matched our design and evaluation goals.**