VEEFYED SENIOR DATA ANALYST – TECHNICAL ASSESSMENT REPORT BY KELVIN MENSAH

04/01/2026 – 05/01/2026

Tools & Technologies: Python 3, Requests, BeautifulSoup (bs4), Pandas for inspection and sampling, Pydantic for schema validation, Google Custom Search API for enrichment, Jupyter Notebook for iterative validation

DAY 1 TASK

Access & Feasibility

- Authentication was explored but required delayed approval and was not viable.
- robots.txt allowed unrestricted crawling.
- Page inspection revealed ld+json metadata and confirmed that all required fields (excluding pricing) were publicly accessible.

Conclusion: Authentication was unnecessary.

Scraper Architecture

A multi-stage pipeline was implemented to improve reliability:

Skincare category filtering → Product URL collection→ Product page visits → Field extraction

- Categories were extracted from site navigation and filtered using skincare-specific URL paths.
- The first 10 products per category were collected to ensure diversity.
- Pagination handling increased coverage beyond the initial render limit.
- Result: 100 product listings → 95 unique products after deduplication.

Deduplication

Performed strictly by product URL, not name, to reflect real ecommerce behavior where products appear in multiple categories.

Extraction & Validation Logic

- Deterministic DOM traversal (no heuristic text guessing).
- Ingredients extracted from structured <ul><li> blocks when present.
- Unicode-aware normalization prevented parsing errors.
- Pydantic validators enforced non-empty names, valid URLs, and consistent field types.
- Manual sampling in Pandas identified and corrected brand normalization issues with code

- Missing data was explicitly stored as null.
- Final output saved as: **day_1_final_scraped.csv**

DAY 2 TASK

Enrichment Architecture

Existing dataset: **day_1_final_scraped.csv**

Select 20 products (oversampling) → Normalize product names → Generate search queries → Google Custom Search API → Parse results → Extract enrichment fields → Validate & score confidence → Select top 10 → Save structured output (CSV/JSON)

API Usage

- Product names were normalized to remove size and promotional text.
- Multiple targeted queries were generated per product (minimum one API call per product).
- Results were parsed to extract the appropriate fields

Validation Logic

- Manufacturer pages accepted only on official brand domains.
- Marketplaces, forums, and community sites were excluded.
- Barcodes/SKUs accepted only when numeric length constraints were met.
- Ingredient lists accepted only if INCI-formatted and sourced from official or reputable databases.
- Country of origin recorded only when explicitly stated.
- Unverifiable fields were left null.

Reliability Determination

- External ingredient lists were compared against the baseline scraped dataset (day_1_final_scraped.csv).
- Confidence was based on ingredient overlap ratios, not exact matches.
- Products were ranked by aggregate confidence scores.
- Only the top 10 highest-confidence enriched products were retained.
- Final output saved as: **day_2_final_enriched.csv**