TruPharma

Drug Label Evidence RAG System

CS 5588 | Spring 2026 | Week 4 Integration Report
Team: Salman Mirza, Amy Ngo, Nithin Songala
Module Owner: Salman Mirza (RAG LLM)

Live App: https://trupharm.streamlit.app
Repository: https://github.com/SalmanM1/CS5588-Deployment

1. Where the Module Fits in the Capstone Architecture

The RAG LLM module is the core intelligence layer of TruPharma. It sits between the Streamlit UI (frontend) and two external services: the openFDA Drug Label API (data source) and Google Gemini 2.0 Flash (optional LLM). The module orchestrates the full pipeline from user question to grounded, citation-enforced answer.

```
User --> Streamlit UI --> RAG Engine (rag_engine.py)
                              |
                              +---> openFDA API (fetch drug label records)
                              +---> Chunk & Index (FAISS dense + BM25 sparse)
                              +---> Hybrid Retrieval (reciprocal rank fusion)
                              +---> Answer Generation (Gemini LLM / extractive)
                              +---> Logging (logs/product_metrics.csv)
```

The system fetches real-time FDA drug label records, chunks text across 10 selected label fields (e.g., drug_interactions, dosage_and_administration, warnings), indexes them with dual retrieval (FAISS inner-product search + BM25 Okapi), and retrieves top-K evidence via hybrid reciprocal rank fusion. Answers include inline citation IDs linking to specific label sections, and the system refuses to answer when evidence is insufficient.

2. Supported User Workflow

Step	User Action	System Response
1	Opens trupharm.streamlit.ap	Displays query interface with example questions
2	Types a drug question	Converts question to openFDA search query
3	Clicks Search	Fetches records, runs hybrid retrieval, generates answer
4	Reviews Response Panel	Answer with inline citations (e.g., [doc_id::field])
5	Expands Evidence Panel	Source text chunks, field names, confidence scores
6	Checks Metrics Panel	Latency, records fetched, retrieval method, confidence

A Stress Test page (sidebar) runs three automated scenarios: drug interactions, dosage/warnings, and an out-of-scope refusal test, validating correctness and latency.

3. Application Interface

The Streamlit application has two pages:

- Primary Demo (streamlit_app.py): Two-column layout with query input and response on the left, evidence artifacts

and pipeline metrics on the right. Includes a sidebar with example queries and optional Gemini API key input.

- Stress Test (pages/stress_test.py): Automated scenario validation running three test queries, displaying pass/fail, latency, and evidence summaries.

The app uses a warm color theme (peach/orange accents) defined in .streamlit/config.toml with custom CSS for Times New Roman body text. Material Icons are preserved for Streamlit's built-in UI elements. The live app is accessible at https://trupharm.streamlit.app.

4. Logging Example

All interactions are logged to logs/product_metrics.csv. The file currently contains 20 interaction records (well exceeding the 5-record minimum). Sample rows:

Timestamp	Query (truncated)	Latency (ms)	Conf.	Evidence	Method
2026-02-10 14:23	Drug interactions for ibuprofen	4,523	0.78	5	hybrid
2026-02-10 15:01	Dosage for acetaminophen + warni	3,892	0.82	5	hybrid
2026-02-10 16:15	Safety warnings for caffeine pro	5,102	0.74	4	hybrid
2026-02-11 09:45	Warnings for aspirin during preg	4,202	0.80	5	hybrid
2026-02-11 10:30	Overdosage symptoms for diphenhy	3,654	0.76	4	hybrid
2026-02-11 14:12	Projected cost of AMR to GDP in	2,104	0.00	0	hybrid
2026-02-12 08:05	Aspirin overdosage & when to sto	5,891	0.82	5	hybrid

Logged fields: timestamp, query, latency_ms, evidence_ids, confidence, num_evidence, num_records, retrieval_method, llm_used, and answer_preview. Note: Row 6 shows the refusal case (confidence = 0.0, out-of-scope question).

5. Production Failure Scenario & Mitigation

Scenario: openFDA API returns 0 results for an obscure or misspelled drug name.

Without mitigation, the pipeline would have no documents to index, potentially crashing or hallucinating an answer with no supporting evidence.

Implemented mitigations:

- Empty result detection: If the API returns 0 records or HTTP 404, the system returns "Not enough evidence in the retrieved context" instead of proceeding.
- Confidence scoring: The heuristic confidence drops to 0.0 when no relevant evidence is found, providing a clear trust signal to the user.
- Logging: Failed queries are logged with confidence=0.0 and empty evidence_ids, enabling post-hoc analysis of query coverage gaps.
- Graceful UI: The frontend displays the refusal message rather than crashing, and the evidence panel shows "No evidence found."

Future improvement: Add fuzzy drug-name matching and spell-check suggestions before querying the API, reducing zero-result queries caused by typos.

6. Deployment Readiness Plan

Architecture & Data Flow

Three-tier design: Streamlit UI -> RAG Engine -> External APIs. The RAG Engine (rag_engine.py) calls openfda_rag.py for API fetching, chunking, and indexing. Data flows: user query -> openFDA search -> fetch records -> chunk (250-word windows, 40-word overlap) -> index (FAISS + BM25) -> hybrid retrieval -> answer generation -> CSV logging.

Aspect	Current Approach	Production Path
Hosting	Streamlit Community Cloud (free)	Containerized on AWS/GCP
Data	Real-time openFDA API (no local storage)	Add Redis caching for common drugs
Scaling	Single instance, API pagination	Horizontal scaling + API key for limits
Monitoring	CSV-based logging	Cloud logging (CloudWatch) + alerting
CI/CD	GitHub auto-deploy on push	GitHub Actions for testing + deployment

7. Impact Evaluation

Metric	Before (Manual)	After (TruPharma)	Improvement
Time-to-answer	10-15 min scanning PDFs	< 5 sec per query	~99% reduction
Citation coverage	Manual copy-paste	Automatic inline citations	Full traceability
Refusal accuracy	User may miss gaps	System refuses at conf=0	Prevents misinfo
Trust indicators	None	Confidence, IDs, fields	Transparent basis

Workflow improvement: Pharmacists, clinicians, and regulatory analysts no longer need to manually scan lengthy drug label PDFs. The RAG system retrieves relevant label sections in seconds and produces answers citing exactly which label field and document the information came from.

Trust indicators: Every answer includes (1) inline citation IDs linking to specific label sections, (2) a heuristic confidence score (0-1), (3) the count of evidence chunks retrieved, and (4) a clear refusal message when evidence is insufficient. These indicators allow users to verify answers against source material and trust the system for clinical and compliance decisions.