Small, RAG Q&A App

Goal: An intelligent web app that answers questions over a tiny local corpus (10–30 .md/.txt /.pdf files).

Core scope

- **App modes**
  - ingest “upload”, then chunk docs, create embeddings, store in vector DB;
  - Ask “natural language (en-US, en-UK)”, then answer, print top-k hits with doc_id#line and scores.
- **Guardrails:**
  - Pre-gen: detect prompt-injection/jailbreak patterns; block or neutralize.
  - Pre-/post-gen PII redaction (email/phone).
  - Refuse with a message if the grounding score is below the threshold.
- **Attribution:**
  - Align each sentence of the answer to supporting chunks(Citation); if any sentence lacks support, then hallucination=true flag in output and mark unsupported sentences.
- **Eval harness (local file-based):**
  - eval.yaml with ~15 Q&A + expected citations.
- **Observability/Costs (local logs):**
  - Log tokens and rough cost estimate
  - Prompt-cache to skip identical queries.
- **Tests:**
  - test_chunker, test_retriever, test_guardrails, test_eval_math runnable via pytest/unittest.

Implementation notes

- **Storage:** local SQLite (FAISS/HNSW table or pgvector-like lib)
- **Models:** any embedding model; generator can be local (GGUF/ONNX) or API-key-driven (behind an adapter).
- **ChatUI :** streamable response similar to ChatGPT UI. NextJS/Python preferred.

Acceptance

- Ask output always includes ≥1 citation when answering; unsafe/empty queries are refused with a clear reason.
- eval prints EM/F1 + sim
- Tests pass locally.

Deliverables

- README.md (setup, commands, examples), LICENSE, requirements/env, eval.yaml, sample corpus, and eval_report.json from a demo run.