**HIPAA COMPLIANT RAG CLINICAL NOTE SUMMARIZATION TOOL**

6‑Week Plan (with daily tasks & artifacts)

Below is **Week 1 & 2 in full daily detail**. Weeks 3–6 include structured daily outcomes; I’ll expand those step‑by‑step with you as we proceed.

**Week 1 — PM foundations + sandbox setup (2–3 hrs/day)**

**Day 1 — Define the product (PRD v0.1)**

**Objective:** Frame the problem, users, scope, success metrics.

**Do this:**

1. Create a **GitHub repo** with two subfolders: /docs and /app.
2. In /docs, create PRD.md using the template below. Fill **Problem**, **Users**, **Jobs‑To‑Be‑Done**, **Non‑Goals**, **Assumptions**, **Success Metrics** (e.g., “**Reduce documentation time by 40% with ≥90% clinician satisfaction**; zero PHI leaks in logs”).
3. Add **Regulatory overview**: paste links to HIPAA Privacy & Security Rule summaries. 12
4. Add **De‑identification frame** (Safe Harbor vs Expert Determination) with the HHS link. 4
5. Add an **OKR** section for the 6‑week project.

**Artifact:** docs/PRD.md (start with this template)

Markdown

# PRD: HIPAA‑Compliant RAG for Clinical Documentation

## 1. Problem & Context

- Today clinicians spend X mins/note; switching costs across EHR panes, risk of copy-paste errors.

## 2. Users & JTBD

- Primary users: hospitalists, residents, scribes. JTBD: "When I draft notes, help me summarize and structure them faster without losing clinical fidelity."

## 3. Goals & Non‑Goals

- Goals: De‑identify; retrieve correctly; generate structured summaries; basic access control; audit trails.

- Non‑Goals (MVP): EHR integration, production SSO, mobile app.

## 4. Success Metrics

- Time-on-task ↓ 40%; Clinician Likert ≥ 4.5/5; No PHI in vector index; 100% actions logged.

## 5. Regulatory alignment

- HIPAA Privacy & Security: links + mapping sections.

- De-identification: Safe Harbor / Expert Determination (HHS guidance).

## 6. Requirements (Must/Should/Could)

- Must: De‑identify, encrypt PHI at rest, role-based access, audit logs, minimal-necessary retrieval.

- Should: Clinically structured sections (HPI/Assessment/Plan), terminology extraction.

- Could: Speech dictation, Keycloak OIDC, OpenSearch dashboards.

## 7. Risks & Mitigations

- Hallucinations → constrained prompts & RAG evaluation.

- False-negative de‑id → recall-biased thresholds, manual spot checks.

## 8. Milestones & Timeline

- Week-by-week roadmap.

## 9. Acceptance Criteria (MVP)

- Given , system returns summary with ; log entry created; index excludes PHI.

## 10. Measurement Plan

- Baseline note-writing time vs tool; RAG evaluation metrics; qualitative interviews.Show less

**Day 2 — Compliance & security scaffolding (Risk Register, Controls Map)**

**Objective:** Demonstrate PM rigor via a **HIPAA → controls** mapping.

**Do this:**

1. Create a docs/Risk\_Register.md listing threats using **STRIDE** categories (spoofing, tampering, etc.) and planned mitigations (auth, encryption, integrity checks, logging). 28
2. Create docs/HIPAA\_Controls\_Map.md: rows for Administrative/Physical/Technical safeguards; map our features to them with **NIST SP 800‑66 Rev.2** references (e.g., access control, audit controls, integrity, person/entity authentication, transmission security). 3
3. Add **OWASP ASVS** checklist sections you’ll verify in code review. 29

**Artifacts:** Risk Register, HIPAA Controls Map, ASVS subset.

**Day 3 — Design the system (DFD/sequence + backlog)**

**Objective:** Visualize the architecture, create a **backlog**.

**Do this:**

1. Draw a **data flow diagram** in **draw.io** showing inputs/outputs, where PHI lives (only encrypted store), and what gets indexed (de‑identified text). 27
2. Add the Mermaid **sequence diagram** I provided into docs/architecture.md and annotate trust boundaries. 20
3. In GitHub Projects, create epics: **De‑identification**, **Indexing & RAG**, **Summarization**, **Security**, **Observability**, **Evaluation**, **UX**.

**Artifact:** docs/architecture.md, GitHub Project board.

**Day 4 — Dev environment & synthetic data**

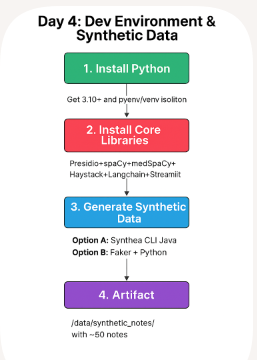
**Objective:** Set up Python env + generate fake patients/notes.

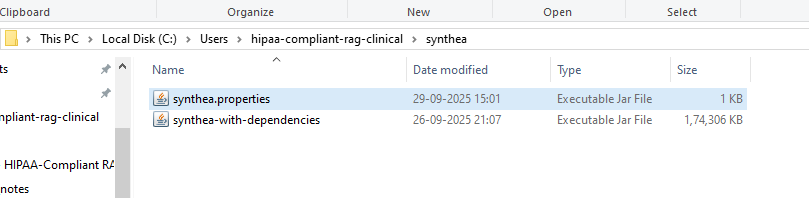
**Do this (step‑by‑step):**

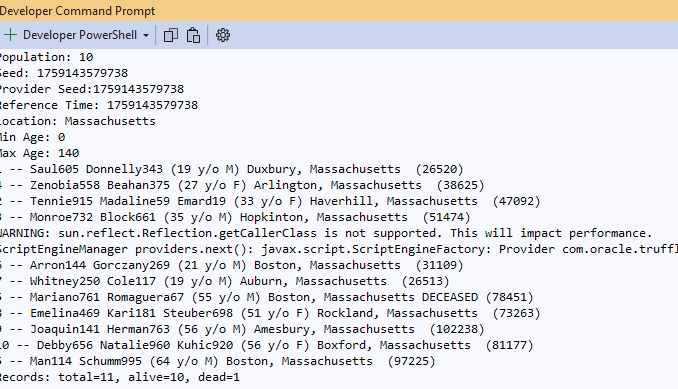
1. Install Python 3.10+, pyenv/venv.
2. pip install core libs: presidio-analyzer presidio-anonymizer spacy scispacy medspacy haystack-ai langchain chromadb faiss-cpu sentence-transformers transformers cryptography opentelemetry-api opentelemetry-sdk streamlit streamlit-authenticator
3. Generate **synthetic records** with **Synthea** (CLI Java app): run for a small population, export CSV/CCDA/FHIR; we will extract clinical notes fields. 22
   * If Synthea feels heavy today, start with a handful of **hand‑crafted synthetic notes**; we’ll swap later.

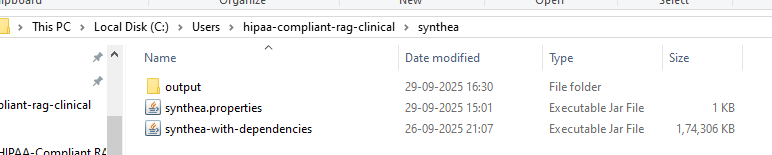
**Artifact:** /data/synthetic\_notes/ folder with ~50 notes.

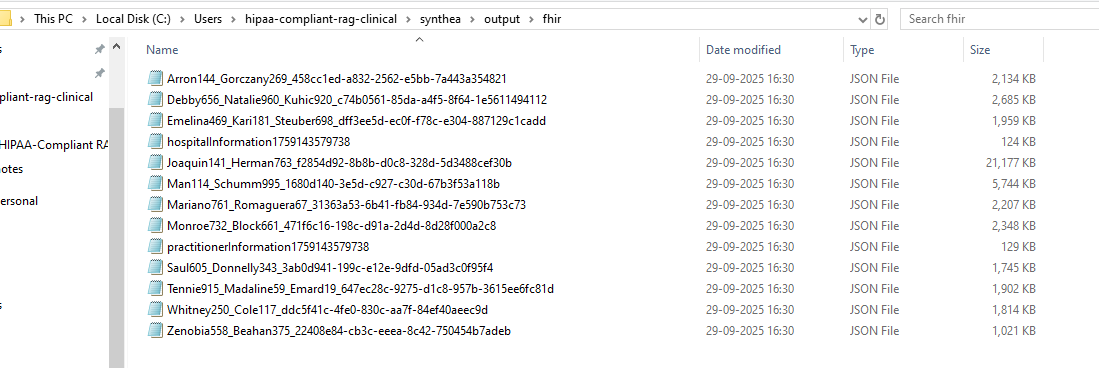
FYI: **MIMIC** and **i2b2/n2c2** are real de‑identified corpora but require training/DUA; you’ll mention this in your README to show compliance awareness.











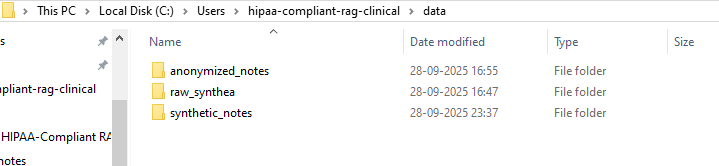
**Day 5 — De‑identification spike (text only)**

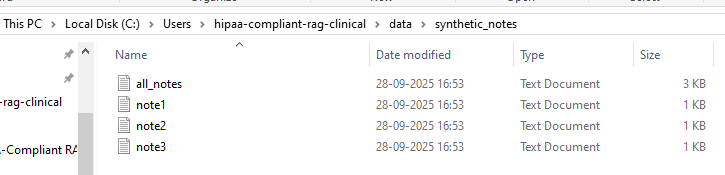
**Objective:** Build the first de‑id pipeline.

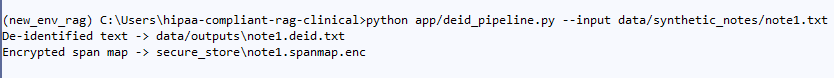
**Do this:**

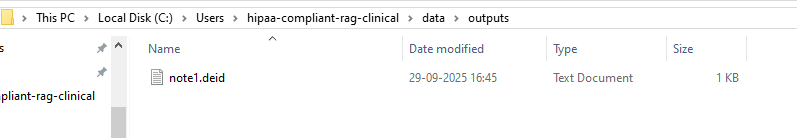
1. Create app/deid\_pipeline.py that:
   * Loads a note → runs **Presidio Analyzer** with recognizers (US phone, SSN, names, etc.) and **medspaCy sectionizer** to preserve clinical sections. 216
   * Applies **anonymizer** operators (replace with tags like [NAME], [PHONE]). 21
   * Outputs **(a)** de‑identified text for indexing and **(b)** an encrypted JSON object that maps PHI spans (for potential re‑link if a clinician is authorized). **Encrypt with Fernet** and store under /secure\_store/
2. Add unit tests for a few cases (names, phone, dates).

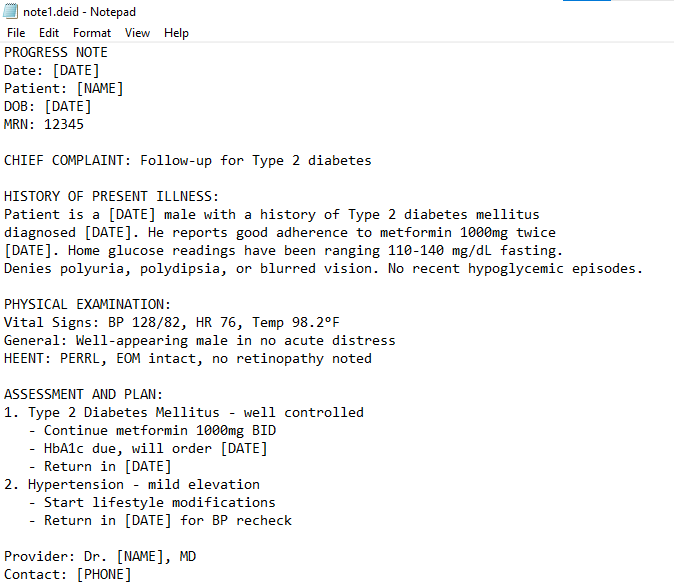
**Artifact:** deid\_pipeline.py, sample before/after, basic tests.

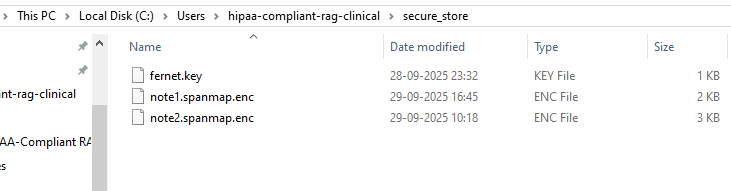












Day 5: De-identification pipeline

- PHI detection via Microsoft Presidio with predefined recognizers (names, phones, dates, emails, SSN).

- Section preservation via regex headers (upgradeable to medspaCy Sectionizer).

- Masking with tags: [NAME], [PHONE], [DATE], [EMAIL], [SSN].

- Encrypted span map written to /secure\_store using Fernet (key stored outside Git).

- Only de-identified text in /data/outputs is used for indexing (Weeks 2+).

**Week 2 — Indexing, retrieval, and first summary (2–3 hrs/day)**

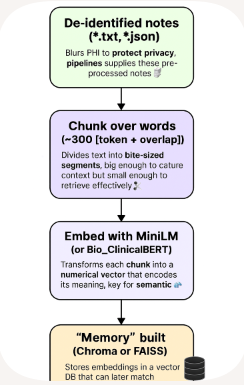
**Day 6 — Vector store & embeddings**

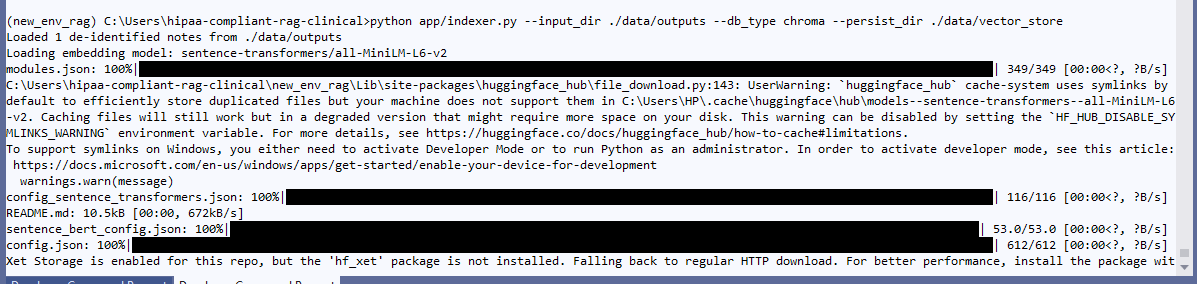
**Objective:** Create a local vector DB and embed de‑identified chunks.

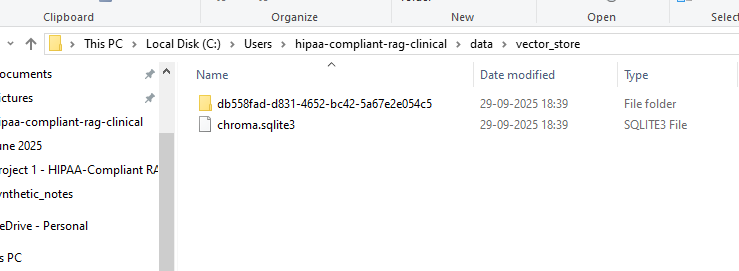
**Do this:**

1. Choose **FAISS** (fast) or **Chroma** (simple persistence). Initialize a collection/index. 910
2. Chunk notes (200–400 tokens with overlap).
3. Use a small open embedding (e.g., sentence-transformers/all-MiniLM-L6-v2) or derive embeddings from **Bio\_ClinicalBERT** if you prefer domain flavor; store vectors + metadata (note\_id, section). 26

**Artifact:** indexer.py and a local persisted DB.







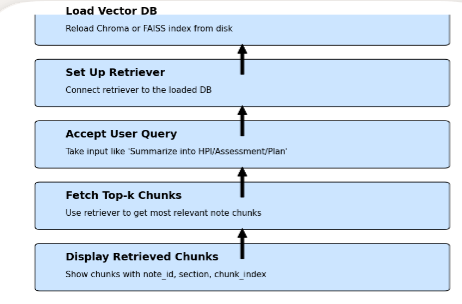
**Day 7 — Retriever + RAG baseline (Haystack or LangChain)**

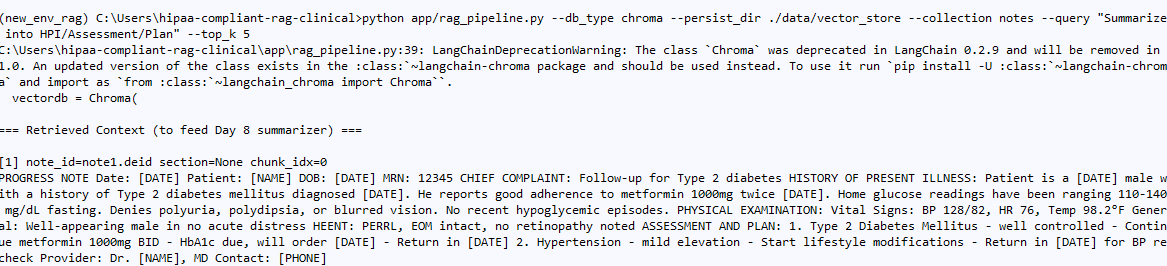
**Objective:** Query → retrieve top‑k chunks → pass to generator.

**Do this:**

1. With **Haystack**: define DocumentStore (FAISS/Chroma), Retriever (dense), pipeline for retrieval. 7*or* with **LangChain**: VectorStoreRetriever + chain. 8
2. Write rag\_pipeline.py that given a question (“Summarize the note into HPI/Assessment/Plan”), returns retrieved context.

**Artifact:** rag\_pipeline.py with CLI for quick tests.



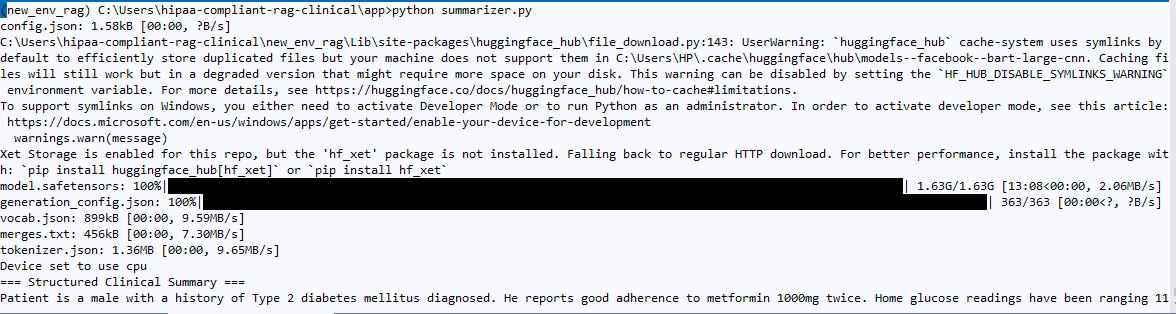


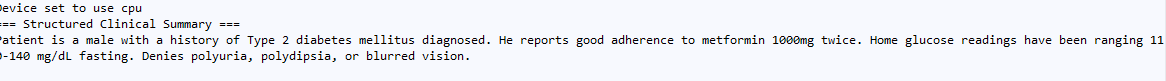
**Day 8 — Summarization model & prompt**

**Objective:** Produce a structured clinical summary.

**Do this:**

1. Load **facebook/bart-large-cnn** or **google/flan-t5-base** via transformers. 1112
2. Create a **prompt template** that enforces structure:
   * Input: retrieved snippets + instructions
   * Output sections: **Chief Complaint, HPI, PMH, Medications, Allergies, Assessment, Plan**, bullets only, “**Do not invent**—answer only from context.”
3. Try 2–3 chunking and top\_k settings; record qualitative differences.





**Artifact:** summarizer.py and example outputs.

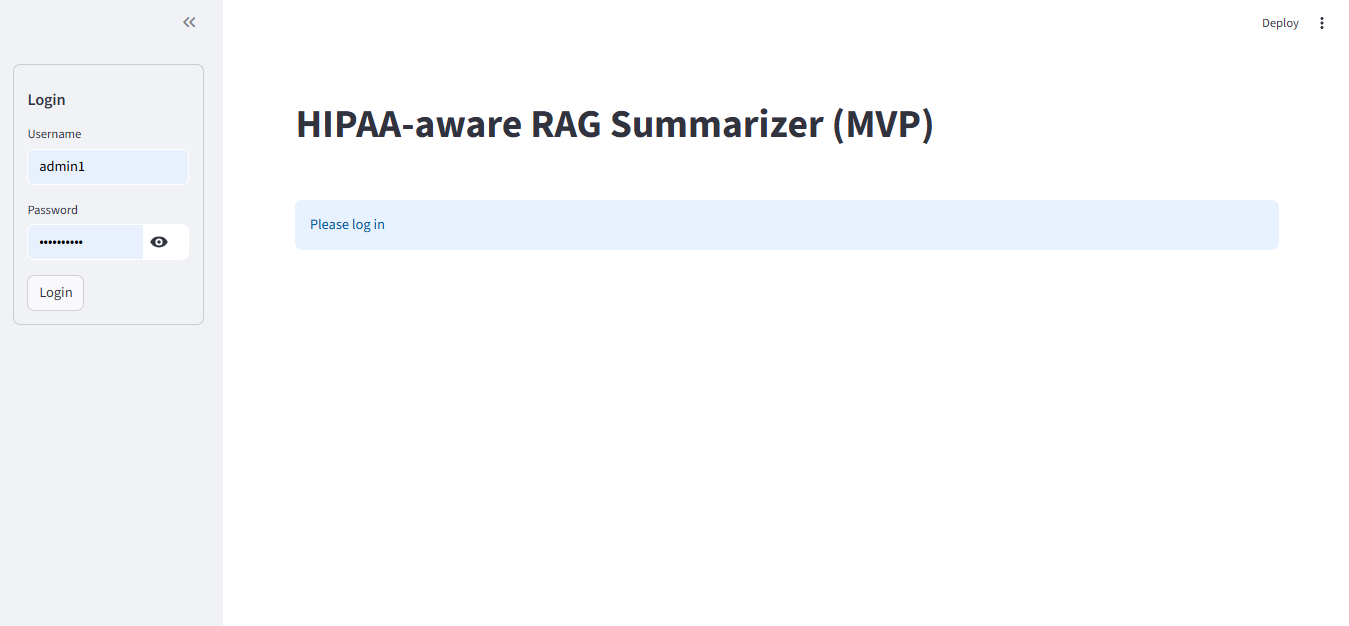
**Day 9 — UI v0 (Streamlit) + simple auth**

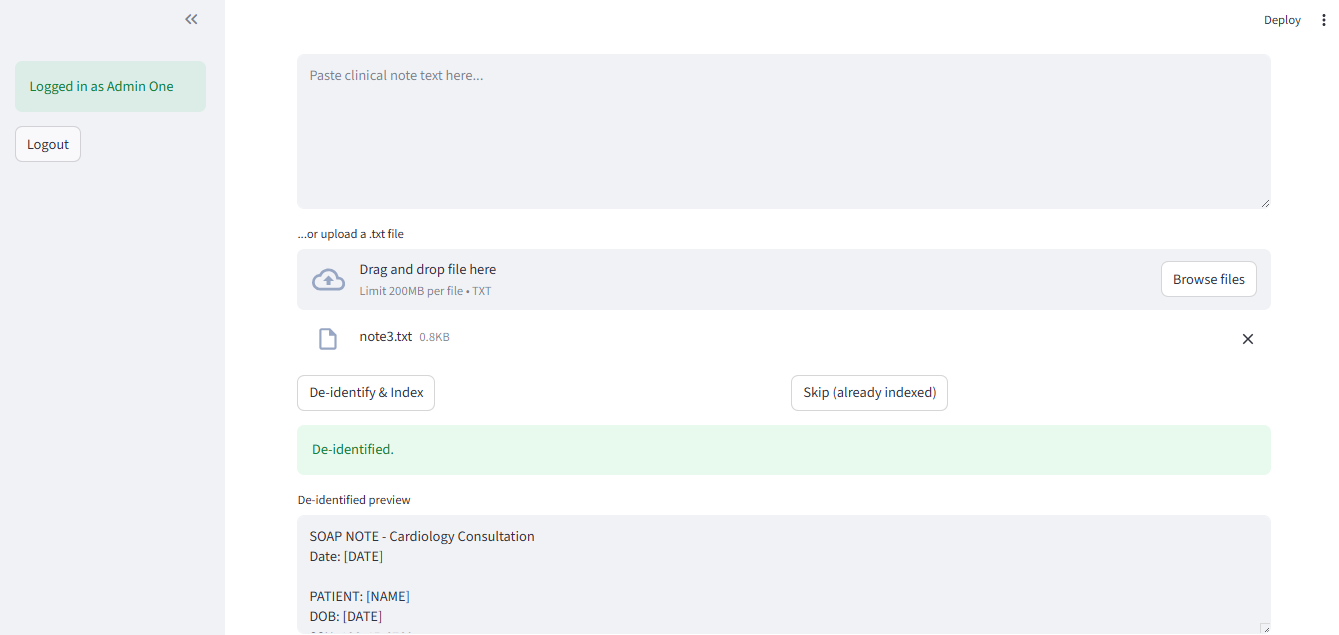
**Objective:** End‑to‑end working demo with **basic auth**.

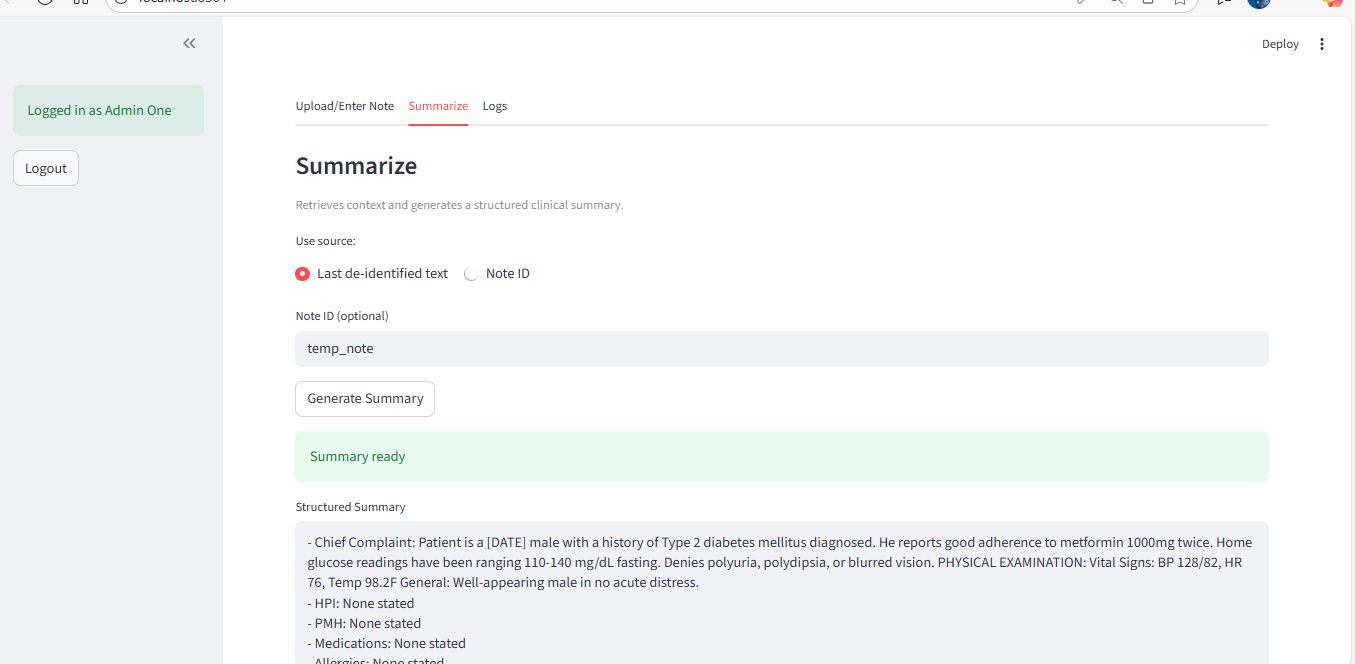
**Do this:**

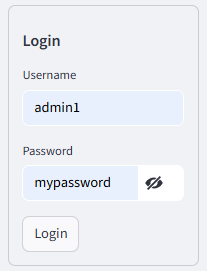
1. Build a **Streamlit** app with three tabs: **Upload/Enter Note**, **Summarize**, **Logs**.
2. Add **streamlit‑authenticator** (user YAML, hashed passwords, simple roles: clinician, admin). 16
3. Wire: paste note → de‑id → index (or skip if already indexed) → ask → summary → download.

**Artifact:** app/main.py (Streamlit) + config.yaml for users/roles.









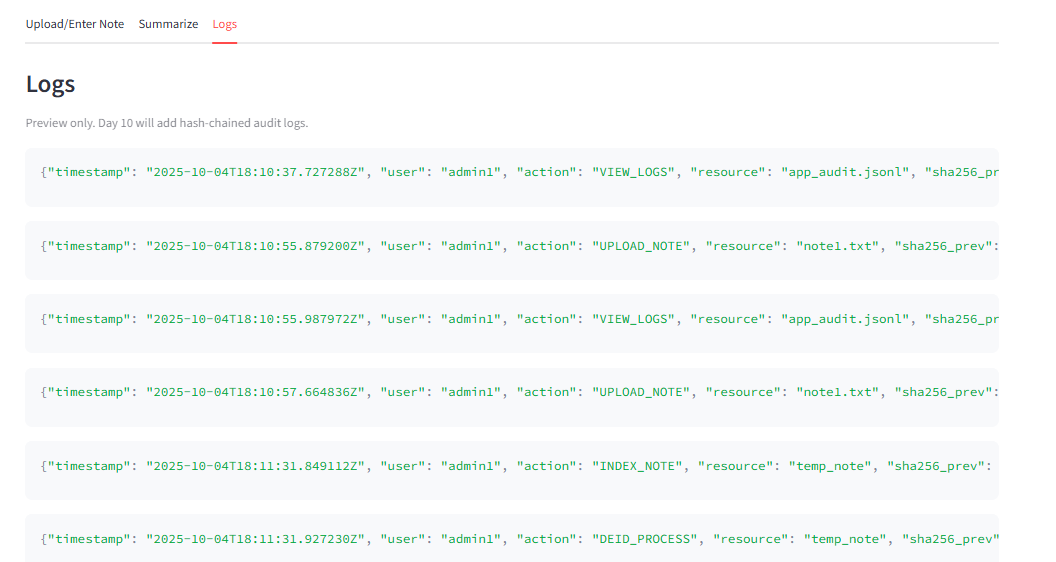
**Day 10 — Audit logging & encryption hygiene**

**Objective:** Log all actions + verify at‑rest encryption.

**Do this:**

1. Implement audit.py: append‑only JSON log with fields (timestamp, user, action, resource, sha256\_prev, sha256\_curr) to create a **hash‑chained** log (tamper‑evident). (Showcase of immutability concept.) 30
2. Generate **OpenTelemetry‑style** log attributes to ease future export to OpenSearch. 14
3. Confirm **Fernet** encryption of any stored PHI payloads, rotate keys using MultiFernet when you update. 17

**Artifact:** audit.py, logs/app\_audit.jsonl, key rotation notes.



**Week 3 — Security hardening & access control (2–3 hrs/day)**

**Day 11–12**

* Add **role checks** (only clinician can view re‑link button; admin can view logs).
* Start a mini **threat model** doc with **STRIDE** per data flow, include mitigations; export a diagram from Microsoft Threat Modeling guidance into your docs. 28
* Create **ASVS validation checklist** for auth, session mgmt, crypto storage, logging. 29

**Day 13–14**

* Optional: prototype **Keycloak** locally and document OIDC flow in your roadmap (even if you don’t ship it). 31
* Add **HTTPS (local)** via self‑signed cert for reverse proxy (document steps).

**Artifacts:** docs/threat\_model.md, docs/asvs\_checklist.md, Keycloak notes.

**Week 4 — Clinical NLP extras + evaluation (2–3 hrs/day)**

**Day 15–16**

* Plug **medspaCy/scispaCy** pipelines to **extract problems, meds, negated terms**; render a “**Terminology Extract**” pane in the UI. 613

**Day 17–18**

* **Evaluate RAG** using **Ragas** metrics (faithfulness, context precision/recall) with a small handcrafted QA set. Add a docs/eval\_report.md. 3233

**Day 19–20**

* UX polish: add **copy‑to‑clipboard**, **download as .docx** for summaries; export PDF for portfolio.

**Artifacts:** docs/eval\_report.md, screenshots, sample .docx.

**Week 5 — Observability & dashboards (2–3 hrs/day)**

**Day 21–22**

* Add **OpenTelemetry** logging context (trace\_id/span\_id) so logs are correlatable. 34

**Day 23–24**

* Optional: run **OpenSearch** locally (Docker compose) and ingest logs via **Data Prepper**; build a simple dashboard (counts by action, user). 15

**Artifacts:** docs/observability.md, Kibana‑style screenshots (OpenSearch Dashboards).

**Week 6 — Evidence pack & mock interview prep (2–3 hrs/day)**

**Day 25–26**

* **Case study write‑up**: problem → approach → architecture → controls mapping → results → demo link → screenshots.

**Day 27–28**

* **Measurement**: run **time‑on‑task** test—baseline vs tool on 5 notes; compute % reduction; include RAG metrics chart.

**Day 29**

* **Readiness checklist** (ASVS), final PRD v1.0, roadmap for vNext (Keycloak OIDC, FHIR export, speech dictation).

**Day 30**

* **Interview packet**: 1‑pager, slides (5–7), demo script, GitHub README polishing, **Risk register & mitigations** summary.

**Artifacts delivered** (all in /docs and /app):

* PRD v1.0, Architecture (DFD/sequence), Risk Register, HIPAA Controls Map, ASVS checklist, Threat Model, Roadmap, Test Plan & Eval Report, Observability notes, Portfolio case study.

**“How‑to” pointers you’ll reference while learning (all free)**

* **HIPAA Privacy & Security Rule summaries** (know what you’re mapping to). 12
* **De‑identification** methods (Safe Harbor / Expert Determination). 4
* **NIST SP 800‑66 Rev.2** (how orgs implement the Security Rule). 3
* **Presidio** docs (PII detection/anonymization) & **medspaCy** docs (clinical pipelines). 56
* **scispaCy** (biomedical models). 13
* **Synthea** for synthetic EHRs. 22
* **Haystack**/**LangChain** tutorials for RAG. 78
* **FAISS**/**Chroma** vector stores. 910
* **BART‑Large‑CNN**/**FLAN‑T5‑Base** model cards. 1112
* **OpenTelemetry** Python logs; **OpenSearch** ingestion. 1415
* **OWASP ASVS v5.0** (latest) for app‑sec verification. 35
* **HL7 FHIR Clinical Notes** & **US Core DocumentReference** (for standards mapping). 1918
* **Mermaid** & **draw.io** for diagrams in your docs. 2027

**Optional but nice for vNext (if time permits)**

* **Speech‑to‑text** to capture dictation locally (offline): **Vosk** or **Whisper** (open‑source). 3637
* **Keycloak** SSO/OIDC integration; document flows for enterprise readiness. 31
* **FHIR export**: wrap summaries as **DocumentReference** with metadata for EHR ingestion. 18

**What “HIPAA‑aligned” means in your readme (talking points)**

* You followed **HHS de‑identification guidance** for what you index (de‑identified text only; PHI stays encrypted). 4
* You implemented **technical safeguards** from the Security Rule: access control, unique user IDs, audit controls, integrity, transmission security; mapped via **NIST 800‑66 Rev.2**. 23
* You keep **tamper‑evident audit logs** and **minimum necessary** data flows. (Show your DFD and log samples.)

**Disclaimer**: This is a portfolio/MVP; real deployments require organizational policies, BAAs, security program, and possibly third‑party assessments.