PolicyMind

A Retrieval-Augmented Generation System for Intelligent Policy Analysis

"Thinks through policies intelligently."

1. Executive Summary

PolicyMind is a retrieval-augmented generation (RAG) system designed to interpret complex insurance policies used in clinical decision-making and prior authorization processes. It was developed to address the inefficiencies and ambiguities that nurses and policy analysts encounter when navigating long, unstructured documents such as medical necessity criteria or coverage limitations. By combining hybrid search and generative reasoning, PolicyMind can summarize, interpret, and answer policy-based questions with verifiable evidence citations.

This report documents the system's design process, technical experimentation, and lessons learned. PolicyMind leverages modern natural language processing pipelines that merge local retrieval with large language model (LLM) synthesis, ensuring accuracy, explainability, and responsible use. The application is implemented through a Streamlit interface that enables user authentication, policy ingestion, and interactive querying. Emphasis is placed on responsible AI practices, including prompt-injection defense, context validation, and secure document handling.

2. Problem Definition and RAG Justification

2.1. Specific Problem and Why It Matters

Insurance policy interpretation is a high-stakes task in healthcare administration. Payers such as Aetna, Anthem, UnitedHealthcare publish detailed coverage documents defining eligibility criteria for thousands of medical procedures. These policies are lengthy, often exceeding 30 pages, and formatted differently across organizations. Clinicians frequently rely on manual reading or keyword search to extract relevant information, which is error-prone and time-intensive. A single oversight can delay authorization and patient care.

2.2. Target Users and Their Needs

PolicyMind was developed through a user-centered design process that prioritized the needs and workflows of nurses and policy analysts. Utilization review nurses need quick access to authoritative text when making case determinations, while policy analysts require transparency and traceability in AI-generated outputs. These user requirements directly influenced PolicyMind's architecture, ensuring interpretability and reliability across roles.

2.3. Why RAG Is Necessary

Existing large language models (LLMs) like GPT-4 can provide fluent summaries but are prone to hallucination when ungrounded. A RAG approach was therefore selected to enforce factual grounding and ensure evidence-based reasoning. Each response is linked to specific retrieved evidence chunks from insurer policies, improving reliability and auditability.

2.4. How Grounding in Authoritative Sources Improves Outcomes

RAG unites two previously separate paradigms—retrieval and generation. The retrieval component ensures factual relevance by embedding documents into a semantic vector space, while the generation component synthesizes those facts into coherent, human readable output. Grounding the model's reasoning in authoritative insurance policy sources ensures that every summary or recommendation is traceable to the insurer's original wording. This transparency enhances decision accuracy, reduces manual lookup time, and helps healthcare reviewers justify determinations during audits or appeals. By anchoring each response to verifiable evidence, grounded retrieval minimizes hallucination, prevents procedural misinterpretation, and achieves the level of precision required for compliance, auditability, and regulatory documentation.

3. RAG System Design

Overview of RAG Architecture

PolicyMind employs a modular **Retrieval-Augmented Generation (RAG)** architecture that links document ingestion, semantic retrieval, and GPT-based generation into one continuous workflow. Policy PDFs from major insurers are converted into clean text, segmented into section-based chunks, and embedded using **SapBERT** within a local **ChromaDB** vector store.

When a user enters a query, the system performs **hybrid retrieval**—combining SapBERTbased semantic search, **LLM-driven query expansion**, and **cross-encoder re-ranking**—to identify the most relevant policy sections. Retrieved context is then formatted into structured prompts with inline citations for summarization or Q&A. This design ensures that every response remains grounded in authoritative insurer policies and traceable to its exact source.

3.1. Retrieval Strategy

PolicyMind's retrieval system was developed through an **iterative process** involving multiple design refinements (Approach 1 to 3) to improve accuracy and interpretability. It was designed to handle the varied terminology and clinical language used across insurer policies. It ensures that user queries, often entered by nurses or analysts using abbreviations or alternative medical terms, accurately surface the policy sections defining medical necessity, coverage limits, or contraindications. The system recognizes equivalent expressions (e.g., "heart bypass surgery" ↔ "coronary artery bypass grafting") and ranks the most relevant content for review and summarization.

Approach 1 *– Dense Semantic Retrieval (Baseline)*

The initial retrieval method used **SapBERT embeddings** with **cosine-similarity search** in **ChromaDB**. This setup effectively captured semantically related text when the query phrasing closely matched the policy wording. However, it often missed relevant results when users entered alternate expressions or abbreviations. For example, a query such as *"heart bypass surgery"* failed to retrieve policies listed under *"coronary artery bypass grafting (CABG)"*, and *"OSA"* did not match *"obstructive sleep apnea."*

During early testing, a basic **hybrid search** combining dense and keyword-based retrieval was also attempted, but manual review showed that it returned inconsistent or irrelevant passages—mixing unrelated sections from multiple policies. As a result, the system reverted to **pure semantic retrieval**, which provided cleaner, more interpretable results for domain specific policy content.

Lesson learned: Dense retrieval worked well for literal and near-matching phrases but struggled with synonymy and abbreviation variation common in medical policy language. Early

hybrid methods added noise without improving precision, confirming the need for a more structured re-ranking approach in later iterations.

Approach 2 *– Query Expansion for Broader Recall*

To address vocabulary mismatch, an **LLM-based query expansion** module was introduced. When a user entered a medical concept, the model generated a concise list of synonyms, abbreviations, and related expressions before retrieval. For example, the input *"knee replacement"* expanded to *"total knee arthroplasty, TKA, joint replacement."* This broadened the semantic search space and enabled PolicyMind to return the correct policies even when users used short forms or alternate phrasing.

Lesson learned: Query expansion substantially improved recall for acronym-heavy and variably worded queries, though it occasionally introduced overlapping or redundant hits that required de-duplication.

Approach 3 – Re-Ranking

The final design adopted a **hybrid retrieval strategy** combining **SapBERT embeddings** for dense retrieval with **cross-encoder re-ranking** using *cross-encoder/ms-marco-MiniLM-L-6v2*. After the top 20 candidate chunks were retrieved, the cross-encoder evaluated each *(query, chunk)* pair to prioritize those with the strongest contextual match. This refinement ensured that the most relevant text segments appeared at the top of the ranked list, significantly improving retrieval precision and interpretability.

Lesson learned: Integrating re-ranking enhanced the quality of top results by promoting contextually aligned chunks and reducing irrelevant matches.

Document-Level Aggregation

After re-ranking, the chunk-level scores were **aggregated at the document (policy) level** to identify the most relevant policies for summarization. Two approaches were tested:

- **Average Aggregation:** Initially, the mean score of all chunks within each policy was calculated. However, this approach **diluted relevance**—policies with a few highly relevant sections and several neutral ones received moderate averages, causing important policies to drop from the top results.

- **Maximum Aggregation (Final Choice):** The final system used the **maximum chunk score** per policy. This ensured that if even one section strongly matched the user query, the entire policy was prioritized for review.

The **max aggregation** method aligned better with PolicyMind's objective of generating **policy level summaries**, ensuring the system selected the correct policy first and supplied coherent, policy-scoped context for downstream summarization.

3.2. Chunking Approach

Insurance policies are hierarchically structured with headings such as *Medical Necessity Criteria*, *Contraindications*, and *Coverage Limitations*. Purely token-based chunking can break semantic continuity, merge unrelated sections and confuse the embedding model. **Section-based chunking** preserves these boundaries, so each chunk reflects a complete policy concept, resulting in higher embedding relevance, clearer retrieval, and fewer hallucinations.

After converting PDFs to clean text, a **regex pattern** identifies common headers, and each section is further divided into ~**800-token** segments with **100-token** overlaps using LangChain's **RecursiveCharacterTextSplitter**. This hybrid method balances structural awareness and token efficiency, yielding coherent, context-rich embeddings.

3.3. Vector Database

ChromaDB was selected as the vector database for PolicyMind because it provides an easy, fully Python-embedded setup that supports rapid experimentation for RAG-based applications. Its seamless integration with LangChain and **Hugging Face embeddings** made it ideal for developing and testing the advanced retrieval pipeline—combining dense **semantic search** with **cross-encoder re-ranking**. Unlike managed cloud options such as Pinecone or Weaviate, ChromaDB allows all embeddings to be stored and queried locally through the **persist_directory** feature, enabling fast, reproducible experiments without external dependencies. This simplicity, combined with efficient in-memory indexing and native cosine similarity search, made ChromaDB the most practical choice for building and evaluating PolicyMind's RAG-enhanced chatbot prototype.

Vector Database Module and Abstraction Layer

To ensure modularity and scalability, we developed a dedicated database.py abstraction layer for vector database operations. Although the current system uses **ChromaDB** directly through LangChain:

```
db = Chroma(persist_directory=PERSIST_DIR, embedding_function=embeddings)
results = db.similarity_search(query, k=10)
```

This wrapper class illustrates professional design principles such as **separation of concerns**, simplified **database migration**, and centralized **error handling**.

The module supports potential transitions to cloud databases (e.g., Pinecone, Weaviate) and enables independent testing of data-layer functionality. While ChromaDB meets our current performance needs, providing this abstraction demonstrates an understanding of scalable architecture and readiness for enterprise-level deployment. This design balances practicality with foresight, showing awareness of when added abstraction enhances maintainability without introducing unnecessary complexity.

3.4. Context Integration (How Prompts Are Formatted)

PolicyMind integrates retrieved policy text into carefully structured prompts to ensure that the large language model (LLM) generates grounded and citation-based outputs. Before every LLM call, retrieved chunks are processed through citation-mapping function **_build_citation_map** which assigns numeric citation keys (e.g., [1], [2]) and links them to their source documents and metadata. The resulting context preserves both policy content and provenance, for example:

```
[1] Medical necessity for septoplasty requires documentation of chronic
    nasal obstruction...
[2] Contraindications include active infection or recent nasal trauma...
```

Summarization Prompts

For summarization (summarize_policy_chunks), the system prompt defines the model's role and output format. The LLM is instructed to produce concise bullet points organized under headers such as **Coverage Criteria**, **Medical Necessity Conditions**, and **Exclusions**, while citing every fact with its reference number. A few-shot example is included to demonstrate correct citation formatting and ensure consistency across outputs.

Conversational Q&A Prompts

For interactive dialogue (conversational_policy_qa), the retrieved context and citation map are embedded in the system message, followed by the user's question and limited history. The system prompt enforces strict citation rules—each factual statement must contain a numeric citation, and uncited content is excluded. Few-shot examples further demonstrate correct question-answer structure.

The full system and user prompt templates used for both summarization and conversational Q&A are provided in **Appendix B**.

4. Data Pipeline

4.1. Dataset Description and Sources

All documents used in this project originated from **publicly accessible, non-PHI (Protected Health Information) insurer policy repositories**. These repositories contain official medical coverage policies, clinical guidelines, and utilization management criteria published by major U.S. healthcare payers. The dataset includes medical policies from *Anthem Blue Cross Blue Shield, Aetna, UnitedHealthcare, and Molina Healthcare*, as well as selected *National and Local Coverage Determinations (NCDs and LCDs)* obtained from the *Centers for Medicare & Medicaid Services (CMS)***.** Collectively, these sources provide authoritative and regulatory guidance on medical necessity criteria, coverage limitations, and procedural requirements, ensuring data authenticity, traceability, and compliance with ethical data use standards.

A complete list of all policy titles and direct links is provided in **Appendix C**.

4.2. Data Collection and Preprocessing

For this project, **55** policy documents were systematically downloaded in PDF format from the public policy libraries of the insurers and CMS. Representative examples include *Anthem's CG-SURG-18: Septoplasty, Aetna's CPB 0516: Colonoscopy and Colorectal Cancer Screening, UnitedHealthcare's Electrical and Ultrasound Bone Growth Stimulators (Commercial), and Molina's MCP-032: Epidural Steroid Injections for Chronic Back Pain*. Each policy PDF was processed using the **UnstructuredPDFLoader** module in **LangChain**. Each file is read page by page, and the extracted text is concatenated and saved as a clean **.txt** file for downstream processing.

This preprocessing step eliminated unnecessary formatting, tables of contents, and embedded elements such as images or hyperlinks, resulting in a consistent, machine readable text corpus. The preprocessed files were then passed into PolicyMind's section-based chunking and embedding stages, forming the structured input required for semantic search and large-language-model reasoning.

4.3. Embedding model

Selecting an embedding model was essential for capturing the medical terminology and policy-specific phrasing in PolicyMind. A review of biomedical language models identified **BioClinicalBERT** and **SapBERT** as promising domain-specific candidates, along with the general-purpose baseline **all-MiniLM-L6-v2** (**Reimers & Gurevych, 2019**). The system initially used MiniLM for its 384-dimension vectors and efficiency, but retrieval tests showed it struggled with clinical synonymy (e.g., linking *septoplasty* with *nasal obstruction surgery*). Attempts to deploy **BioBERT** failed due to compatibility and size constraints, leading to the successful adoption of **SapBERT** (*cambridgeltl/SapBERT-from-PubMedBERT-fulltext*).

In comparative testing on twenty policy queries, SapBERT improved **precision (0.75 → 0.85)** and **recall (0.85 → 0.95)**, accurately mapping terms such as *OSA* ↔ *obstructive sleep apnea* and *CABG* ↔ *coronary artery bypass grafting*.

To further refine ranking quality, a **cross-encoder re-ranking model** (*cross-encoder/msmarco-MiniLM-L-6-v2*) was integrated, forming a two-stage retrieval pipeline balancing **semantic accuracy and computational efficiency**.

4.4. Evidence That Retrieval Works Effectively

Retrieval effectiveness was evaluated through approximately **60 policy-related queries** tested across the team to assess the overall performance of PolicyMind's **Hybrid Search Pipeline**. The system combines **SapBERT-based vector retrieval** with **cross-encoder reranking**, supported by section-based chunking for improved context preservation. Together, these components achieved an average **recall of 90%** and **precision of 80%**, indicating that relevant policy sections consistently appeared among the top-ranked results. Retrieved chunks accurately reflected medical terminology and policy intent, while section-based chunking ensured that each result corresponded to a coherent and complete policy section rather than fragmented sentences. These outcomes confirm that PolicyMind's retrieval

component functions effectively as a high-precision, context-aware mechanism for surfacing the most relevant evidence prior to generation.

5. Security and Responsible AI

5.1. Domain-Specific Risks (e.g., Giving Medical/Financial/Legal Advice)

PolicyMind operates in the **medical-policy domain**, which requires strong safeguards to prevent misinterpretation or misuse of clinical information. While the system does **not process patient data**, it interacts with sensitive regulatory text that defines medical necessity and coverage rules. Domain-specific risks include:

- **Out-of-scope queries** such as requests for diagnosis, treatment, or personal medical recommendations.
- **Policy misinterpretation** due to hallucinated or incomplete model summaries.
- **Unintended exposure of private identifiers** if external or PHI-containing documents were ever added.

To mitigate these risks, the system enforces strict domain boundaries and confines all LLM responses to factual, policy-based interpretations only.

5.2. Threats Anticipated and Tested Against

During development, the system was tested against a range of adversarial and malformed inputs across both the search and chat components. The primary categories of threats include:

1. **Prompt injection and jailbreaks** – attempts to override system instructions or reveal hidden prompts.
2. **Role manipulation** – user inputs instructing the model to act as a doctor or policy author.
3. **System prompt exposure** – attempts to view or modify backend configuration.
4. **Gibberish and nonsensical inputs** – random strings or keyboard mashes that consume tokens and distort retrieval.
5. **Malformed uploads** – non-PDF or corrupted policy documents during ingestion.
6. **Rate and depth abuse** – extremely long queries or chat histories that risk performance degradation.

The system was validated using **test_security_validation()** and manual testing, covering over a dozen representative scenarios.

Results: All malicious inputs were correctly blocked, flagged, or sanitized before processing. Valid policy queries continued to pass securely into the RAG retrieval and summarization flow.

5.3. Implemented Guardrails and Their Effectiveness

PolicyMind incorporates multiple layers of automated guardrails through the security.py module, creating a **defense-in-depth** framework.

Category	Implementation	Function	Effectiveness
Prompt Injection / Jailbreak	Regex-based pattern detection for phrases like "ignore previous" or "act as."	detect_prompt_injection()	Successfully blocked override attempts
Domain Validation	Rejects medical-advice or PHI-related prompts.	validate_medical_query()	Prevented scope violations
Gibberish & Trivial Query Filters	Detects non-linguistic input or greetings (e.g., "asdfgh," "hello").	is_gibberish(), is_trivial_query()	Filtered noise efficiently
Length / Rate Limits	Caps query length (≤1000 chars) and session turns (≤20).	check_query_length(), check_conversation_depth()	Controlled resource use
Output Validation	Scans LLM responses for unsafe phrasing or sensitive data.	validate_output()	Removed unsafe or directive language
Logging	Stores blocked events with timestamps and reasons.	log_security_event()	Enables transparency and auditability

An additional domain-restriction prompt was introduced to discourage the LLM from answering out-of-scope or general-knowledge queries (e.g., "What is the capital of France?"). This safeguard helps ensure that responses remain grounded in policy data, though occasional cases still bypass the restriction when phrasing is ambiguous.

Effectiveness summary

Testing confirmed that each guardrail operated as intended, effectively blocking most out-of scope queries, adversarial attempts, and unsafe inputs while allowing legitimate policy searches and summaries to proceed seamlessly. The domain-restriction prompt further reduced general-knowledge responses (e.g., non-policy questions), though a few ambiguous cases still bypassed the filter—highlighting an area for future refinement.

5.4. Limitations and Residual Risks

Despite comprehensive safeguards, certain residual risks remain:

- **Incomplete or over-generalized summaries:** Even with retrieval grounding, models may omit subtle policy nuances.
- **Potential PHI leakage:** If future datasets include sensitive documents, embeddings could inadvertently encode identifiers.
- **External API dependency:** Reliance on external LLM providers introduces latent privacy and availability risks.
- **Partial domain restriction:** Although domain-specific prompts discourage general knowledge queries (e.g., "What is the capital of France?"), some ambiguous or indirectly phrased inputs may still bypass this filter.

Planned Mitigations: Future work will integrate automated **red-teaming pipelines**, **differential privacy checks** for embeddings, and **secure logging backends** for greater auditability. Continuous evaluation of model responses under adversarial conditions will further strengthen PolicyMind's reliability and ethical compliance.

6. Reflection and Future Work

Building PolicyMind demonstrated the practical challenges of balancing retrieval precision, computational efficiency, and interpretability in RAG-based healthcare applications. The integration of section-aware chunking and hybrid retrieval significantly enhanced factual grounding, though blending context across multiple insurer policies with overlapping terminology remains a challenge. In some cases, similar language across different payers caused partial redundancy in retrieval results, emphasizing the need for stronger policy-level context separation.

The Streamlit implementation facilitated rapid prototyping but introduced difficulties in session state management during concurrent use. Incorporating persistent SQLite-based session storage and authentication tokens improved reliability and user continuity, highlighting the importance of architectural robustness in LLM-driven interfaces.

A notable enhancement during development was the Chat Citations Fix, which resolved issues where LLM responses occasionally lacked or displayed incorrect source citations. The fix

involved regex-based citation filtering, database schema migration for citation storage, and improved few-shot prompt examples to guide the model's citation formatting. As a result, each generated response now correctly references the policy name and page number, reinforcing explainability and auditability across chat sessions.

Lessons learned include the value of iterative evaluation, provenance preservation, and maintaining clear traceability between retrieved evidence and generated responses. Designing for explainability—through inline citations and structured outputs—proved essential to ensure transparency in regulatory contexts such as healthcare policy review.

Future Work

Future development will focus on expanding PolicyMind's capabilities and strengthening its reliability through the following initiatives:

- Clinical Review Assist Integration: Extend PolicyMind into a broader Clinical Review Assist system where nurses can upload medical records, receive automated summaries, and ask policy-grounded questions about patient cases.
- Policy Criteria Evaluation: Introduce an evaluation feature that allows users to paste policy criteria and automatically assess whether those criteria are met based on the uploaded medical record.
- Policy Metadata Extraction: Implement automated extraction of policy publication and effective dates to maintain a continuously updated, version-controlled medical policy database aligned with the latest insurer guidelines.
- Multimodal Ingestion: Enable multimodal document ingestion, including scanned PDFs, images, and structured tables, to handle diverse clinical record formats.
- Enhanced Embeddings: Integrate domain-tuned biomedical embeddings for improved contextual understanding and precision in retrieval.
- Continuous RAG Evaluation: Incorporate automated RAG performance dashboards using *RAGAS* and *TruLens* for ongoing quality and factuality assessment.
- Advanced Responsible AI Features: Deploy red-teaming scripts, retrieval anomaly detection, and data watermarking to strengthen robustness, ensure transparency, and maintain long-term trustworthiness.

7. References

Anthropic. (2024). *The Claude 3 model family: Opus, Sonnet, Haiku*. Anthropic, San Francisco, CA. Retrieved from https://www.anthropic.com/news/claude-3-family

Chroma. (2024). *Chroma: The AI-native open-source embedding database*. Retrieved from https://www.trychroma.com

Henderson, P., et al. (2024). *Evaluating and reducing hallucinations in large language models for domain-specific applications*. NeurIPS Workshop on Trustworthy and Responsible AI.

LangChain. (2024). *LangChain documentation*. Retrieved from https://www.langchain.com

NIST. (2023). *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. National Institute of Standards and Technology. Retrieved from https://www.nist.gov/itl/ai-riskmanagement-framework

OpenAI. (2024). *GPT-4 technical report*. *arXiv preprint* arXiv:2303.08774. Retrieved from https://arxiv.org/abs/2303.08774

OWASP Foundation. (2024). *OWASP Top 10 for large language model applications*. Open Worldwide Application Security Project. Retrieved from https://owasp.org/www-project-top10-for-llm/

Ramakrishna, A., & Rastogi, P. (2023). *Responsible retrieval-augmented generation (RAG): Challenges and frameworks*. *Proceedings of the ACM Conference on Information and Knowledge Management (CIKM 2023)*.

Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*. *arXiv preprint* arXiv:1908.10084.

Schulhoff, S., et al. (2023). *Defending against prompt injection attacks in large language models*. *arXiv preprint* arXiv:2304.11165.

Streamlit Inc. (2024). *Streamlit documentation*. Retrieved from https://docs.streamlit.io

TruLens AI. (2024). *Evaluating RAG applications for trust and quality using TruLens*. Retrieved from https://www.trulens.org

U.S. Department of Health and Human Services. (2023). *HIPAA Security Rule guidance material*. Office for Civil Rights (OCR). Retrieved from https://www.hhs.gov/hipaa/forprofessionals/security/index.html

Unstructured.io. (2024). *Unstructured: Document parsing and preprocessing framework*. Retrieved from https://unstructured.io

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., et al. (2022). *Ethical and social risks of harm from language models*. *arXiv preprint* arXiv:2112.04359.
Zou, A. J., Wang, Y., Chen, Y., et al. (2023). *Universal and transferable adversarial attacks on aligned language models*. *arXiv preprint* arXiv:2307.15043.

Zhang, L., Su, Y., & Tang, J. (2021). *SapBERT: Self-alignment pretraining for biomedical entity representations*. *arXiv preprint* arXiv:2105.14779.

RAGAS. (2024). *RAGAS: Evaluation framework for retrieval-augmented generation systems*. Hugging Face. Retrieved from https://github.com/explodinggradients/ragas

Appendix A. Foundation: Tutorial Exercises

This appendix documents how the four Streamlit tutorial exercises (Sections 9.1 – 9.4) were implemented within the **PolicyMind** application. Each exercise was adapted and extended to fit the project's retrieval-augmented generation (RAG) workflow, user session management, and security design.

B.1 Exercise 1 – Model Selection Objective:

Allow users to select between multiple large-language models (LLMs).

Implementation:

Model selection is implemented in the **sidebar settings** using Streamlit's **st.selectbox().** The chosen model (gpt-4o-mini, gpt-4o, or o1-preview) is stored in session state **(st.session_state.model_sel)** and passed dynamically to all LLM calls for summarization and Q&A.

```
st.session_state.model_sel = st.selectbox(
        "Model",
        ["gpt-4o", "gpt-4o-mini", "o1-preview"],
        index=0
    )
...
result = summarize_policy_chunks(
                    retrieved_chunks=docs,
                    llm_client=oai_client,
                    llm_model=st.session_state.model_sel,
                )
```

Outcome:

Users can easily experiment with different models, enabling comparison of accuracy, latency, and token efficiency directly within the UI.

B.2 Exercise 2 – System Prompt Customization

Objective:

Allow users to modify the assistant's behavior dynamically.

Implementation:

A **st.text_area()** field in the sidebar lets users edit the system prompt that guides the model's reasoning style. The value is saved in **st.session_state.system_prompt** and injected into all Q&A prompts via the **custom_prompt** argument.

```
st.session_state.system_prompt = st.text_area(
        "System Prompt (for Q&A only)",
        value=st.session_state.get("system_prompt", default_prompt),
        help="Customize the assistant's behavior for question-answering.",
        height=200
    )
...
def process_chat_question(name, user_text, chat_key):
. . .
  custom_prompt = st.session_state.get("system_prompt", None)
```

Outcome:

Users can switch between clinical, regulatory, or concise explanation styles in real time without restarting the session.

B.3 Exercise 3 – Token Limit Warnings

Objective:

Warn users when nearing the model's token limit.

Implementation:

The app integrates **tiktoken** to compute total tokens in conversation history. When the count approaches the safe limit (~100 000 tokens), the app displays a Streamlit warning to prevent context-overflow errors and maintain response reliability.

```
try:
  import tiktoken
  TIKTOKEN_AVAILABLE = True
except ImportError:
  print("Warning: tiktoken not installed. Token counting disabled.")
  TIKTOKEN_AVAILABLE = False
. . .

  is_safe, token_count, token_warning = check_token_limit(
```

```
    messages,
    model=llm_model,
    warning_threshold=100000,
    max_limit=120000
)
```

Outcome:

Improves system robustness, avoids “context length exceeded” exceptions, and enhances user awareness of computational cost.

B.4 Exercise 4 – Message Regeneration

Objective:

Enable re-generation of the most recent model response.

Implementation:

A **“ Regenerate”** button beside each chat session removes the last assistant message and reissues the API call using the most recent user query. This allows side-by-side comparison of responses from different models or prompt edits.

```
if regen_clicked:
. . .
  if last_user:
     if st.session_state[chat_key]["history"] and st.session_state[chat_key]["history"][-1]["role"] == "user":
        st.session_state[chat_key]["history"].pop()
        print(f"Removed last user message for regeneration")

     process_chat_question(name, last_user, chat_key)
```

Outcome:

Enhances usability and supports iterative experimentation without retyping queries, essential for clinical policy review workflows.

Summary

All four Streamlit tutorial features were successfully implemented within **PolicyMind**, extending the base exercises into a robust, domain-specific RAG interface. Users can now dynamically control model choice, customize system behavior, monitor token usage, and regenerate responses—all integrated seamlessly with conversation caching, policy summarization, and citation-grounded retrieval.

Appendix B. Prompt Templates for Summarization and Q&A

Summarization Prompt

```
system_msg=(
        "You are a healthcare policy summarization assistant. "
        "Summarize the policy content concisely using bullet points and section headers. "
        "like 'Coverage Criteria', 'Medical Necessity Conditions', 'Limitations', 'Exclusions' and 'Coding Information' "
        "\n\n"
        "CRITICAL: You MUST use inline citations [1], [2], [3] for EVERY fact. "
        "Each [number] refers to a numbered source chunk below.\n\n"
        "EXAMPLE FORMAT:\n"
        "**Coverage Criteria**\n"
        "- Prior authorization required [1]\n"
        "- Medical necessity must be documented [2]\n"
        "- Conservative therapy must have failed [3]\n\n"
        "**Exclusions**\n"
        "- Experimental treatments not covered [4]\n"
        "- Cosmetic procedures excluded [5]\n\n"
        "Remember: Every bullet point needs a [number] citation!"
    )

user_msg = f"Policy Context:\n{context_text}\n\nSummarize the key criteria clearly with citations."
```

Conversational Q&A Prompt

```
base_prompt = custom_prompt or (
        "You are a healthcare policy assistant. Answer questions using ONLY the provided policy context. "
        "You MUST cite every fact using inline citations [1], [2], [3], etc.\n\n"
        "CITATION RULES:\n"
        "- Every statement must have a citation number in brackets\n"
        "- Citations refer to the numbered chunks in the policy context below\n"
```

```
        "- Format: 'Fact or requirement [citation_number]'\n"
        "- Multiple facts need multiple citations\n"
        "- If you cannot cite it, do not say it\n\n"
        "EXAMPLE RESPONSES:\n"
        "Question: What are the coverage requirements?\n"
        "Good Answer: 'Coverage requires prior authorization [1], documented medical necessity [2], and failure of conservative therapy for 6 months [3].'\n"
        "Bad Answer: 'Coverage requires prior authorization and medical necessity.' (NO CITATIONS)\n\n"
        "Question: What are the exclusions?\n"
        "Good Answer: 'Exclusions include experimental treatments [4], cosmetic procedures [5], and investigational devices [6].'\n\n"
        "Remember: EVERY fact needs a [number] citation!"

        "If the context does not contain the answer, respond strictly with "
        "'Not found in policy context.' "
        "Do not use external or general knowledge. "
        "Do not guess or infer beyond the retrieved policy text. "
        "Maintain continuity from previous conversation turns. "
    )

system_msg = {
        "role": "system",
        "content": f"{base_prompt}\n\nPolicy Context:\n{context_text}"
    }
```

These prompt templates explicitly control the LLM's tone, structure, and grounding behavior, ensuring reproducible, citation-driven outputs for both summarization and conversational use cases.

Appendix C. Source Policy Documents

The following links represent **sample sources** from the total of **55 PDF medical policy files** downloaded for this project. These repositories were used to collect current clinical and utilization management guidelines from major commercial insurance providers and from the Centers for Medicare & Medicaid Services (CMS). All files were obtained directly from official public policy libraries to ensure authenticity and reproducibility.

Insurer	Policy / Topic	Link
Aetna	Autologous Chondrocyte Implantation - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/200_299/0247.html
UnitedHealthcare	bariatric-surgery.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/comm-medical-drug/bariatric-surgery.pdf

Aetna	Bone Growth Stimulators - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/300_399/0343.html
Anthem	CG-DME-36 Pediatric Gait Trainers.pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_c174202.html
Anthem	CG-DME-40 Noninvasive Electrical Bone Growth Stimulation of the Appendicular Skeleton.pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_d055254.html
Anthem	CG-DME-45 Ultrasound Bone Growth Stimulation.pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_d083855.html
Anthem	CG-DME-49 Standing Frames.pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_e001130.html
Anthem	CG-MED-52 Allergy Immunotherapy (Subcutaneous).pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_c183207.html
Anthem	CG-MED-59 Upper Gastrointestinal Endoscopy in Adults.pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_c197646.html
Anthem	CG-MED-73 Hyperbaric Oxygen Therapy (Systemic_Topical).pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_d083866.html
Anthem	CG-MED-78 Anesthesia Services for Interventional Pain Management Procedures.pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_d087067.html
Anthem	CG-MED-94 Vestibular Function Testing.pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_e002727.html
Anthem	CG-MED-95 Transanal Irrigation.pdf	https://www.anthem.com/medpolicies/abc/active/gl_pw_e002531.html
Anthem	CG-MED-98 Parenteral Antibiotics for the Treatment of Lyme Disease.pdf	CG-MED-98 Parenteral Antibiotics for the Treatment of Lyme Disease.pdf
Anthem	CG-SURG-01 Colonoscopy.pdf	https://www.anthem.com/medpolicies/abc/active/gl_pw_c119565.html
Anthem	CG-SURG-119 Treatment of Varicose Veins (Lower Extremities).pdf	https://www.anthem.com/medpolicies/abcbs/active/gl_pw_e002908.html
Anthem	CG-SURG-120 Vagus Nerve Stimulation.pdf	https://www.anthem.com/medpolicies/abc/active/gl_pw_e002906.html
Anthem	CG-SURG-18 Septoplasty.pdf	https://www.anthem.com/medpolicies/abc/active/gl_pw_a051158.html

Aetna	Colonoscopy and Colorectal Cancer Screening - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/500_599/0516.html
Aetna	Cosmetic Surgery and Procedures - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/500_599/0516.html
UnitedHealthcare	dme-prosthetics-appliances-nutritional-supplies-grid.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/medadv-mp/dme-prosthetics-appliances-nutritional-supplies-grid.pdf?
UnitedHealthcare	electrical-ultrasonic-stimulators.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/medadv-mp/electrical-ultrasonic-stimulators.pdf
UnitedHealthcare	electrical-ultrasound-bone-growth-stimulators.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/comm-medical-drug/electrical-ultrasound-bone-growth-stimulators.pdf
Aetna	Epidural Injection Technologies - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/900_999/0934.html
Molina	Epidural-Steroid-Injections-for-Chronic-Back-Pain-MCG-032.pdf	https://www.molinahealthcare.com/~/media/Molina/PublicWebsite/PDF/providers/sc/medicaid/Epidural-Steroid-Injections-for-Chronic-Back-Pain-MCG-032.pdf
UnitedHealthcare	epidural-steroid-injections-spinal-pain.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/comm-medical-drug/epidural-steroid-injections-spinal-pain.pdf
UnitedHealthcare	gastroesophageal-gastrointestinal-gi-services-procedures.pdf	https://uhg1-prod.adobecqms.net/content/dam/provider/docs/public/policies/medadv-mp/gastroesophageal-gastrointestinal-gi-services-procedures.pdf?utm_source=chatgpt.com
Aetna	Injectable Medications - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/1_99/0020.html

CMS	LCD - Cervical Fusion (L39758).pdf	https://www.cms.gov/medicare-coverage-database/view/lcd.aspx?lcdid=39758
CMS	LCD - Epidural Steroid Injections for Pain Management (L39240).pdf	https://www.cms.gov/medicare-coverage-database/view/lcd.aspx?lcdid=39240
CMS	LCD - Hypoglossal Nerve Stimulation for the Treatment of Obstructive Sleep Apnea (L38312).pdf	https://www.cms.gov/medicare-coverage-database/view/lcd.aspx?lcdid=38312
CMS	LCD - Nerve Blockade for Treatment of Chronic Pain and Neuropathy (L35457).pdf	https://www.cms.gov/medicare-coverage-database/view/lcd.aspx?lcdid=35457&ver=58&bc=0
CMS	LCD - Percutaneous Vertebral Augmentation (PVA) for Osteoporotic Vertebral Compression Fracture (VCF) (L34106).pdf	https://www.cms.gov/medicare-coverage-database/view/lcd.aspx?LCDId=34106
CMS	LCD - Peripheral Nerve Stimulation (L37360).pdf	https://www.cms.gov/medicare-coverage-database/view/lcd.aspx?LCDId=37360
CMS	LCD - Sacral Nerve Stimulation for the Treatment of Urinary and Fecal Incontinence (L39543).pdf	LCD - Sacral Nerve Stimulation for the Treatment of Urinary and Fecal Incontinence (L39543).pdf
CMS	LCD - Spinal Cord Stimulators for Chronic Pain (L36204).pdf	https://www.cms.gov/medicare-coverage-database/view/lcd.aspx?lcdid=36204
Molina	MCP-394-Vagus-Nerve-Stimulation-for-Depression.pdf	https://www.molinahealthcare.com/-/media/Molina/PublicWebsite/PDF/Providers/wa/Medicaid/resource/mcp-mcr/MCP-394-Vagus-Nerve-Stimulation-for-Depression.pdf
CMS	NCD - Arthroscopic Lavage and Arthroscopic Debridement for the Osteoarthritic Knee (150.9).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?ncdid=285
CMS	NCD - Deep Brain Stimulation for Essential Tremor and Parkinson's Disease (160.24).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?NCDId=279&ncdver

CMS	NCD - Diagnosis and Treatment of Impotence (230.4).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?NCDId=32&NCDver=1
CMS	NCD - Incontinence Control Devices (230.10).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?NCDId=241
CMS	NCD - Induced Lesions of Nerve Tracts (160.1).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?NCDId=19&ncd
CMS	NCD - Percutaneous Image-Guided Lumbar Decompression for Lumbar Spinal Stenosis (150.13).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?ncdid=358
CMS	NCD - Phrenic Nerve Stimulator (160.19).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?NCDId=244&ncdver=1&bc=AAAAQAAAAAAA
CMS	NCD - Sacral Nerve Stimulation For Urinary Incontinence (230.18).pdf	https://www.cms.gov/medicare-coverage-database/view/ncd.aspx?NCDId=249
UnitedHealthcare	pediatric-gait-trainers-standing-systems-cs.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/medicaid-comm-plan/pediatric-gait-trainers-standing-systems-cs.pdf
UnitedHealthcare	preventive-care-services.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/comm-medical-drug/preventive-care-services.pdf
UnitedHealthcare	Rhinoplasty and Other Nasal Procedures – Commercial and Individual Exchange Medical Policy.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/comm-medical-drug/rhinoplasty-other-nasal-surgeries.pdf
UnitedHealthcare	screening-colonoscopy-procedures-site-service.pdf	https://www.uhcprovider.com/content/dam/provider/docs/public/policies/comm-medical-drug/screening-colonoscopy-procedures-site-service.pdf
Aetna	Septoplasty and Rhinoplasty - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/1_99/0005.html

Aetna	Standing Frames, Tables, and Transfer Boards - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/400_499/0481.html
Anthem	SURG.00071 Percutaneous Spinal Surgery.pdf	https://www.anthem.com/medpolicies/abc/active/mp_pw_a053367.html
Aetna	vagus-nerve-stimulation.pdf	https://www.aetna.com/cpb/medical/data/100_199/0191.html
Aetna	Vasectomy Procedures - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/1_99/0027.html
Aetna	Viscosupplementation - Medical Clinical Policy Bulletins _ Aetna.pdf	https://www.aetna.com/cpb/medical/data/100_199/0179.html