MediQ
A RAG-based Drug Information Chatbot.

Summary of Works Day 1
(20/08/2025)

Activities Performed

1. Conducted project overview meeting with mentors (CTS & Rec) to clarify objectives and expectations.
2. Finalized the Problem Statement: To build a RAG-based Drug Information Chatbot that provides accurate, explainable, and citation-supported answers for medical queries.
3. Discussed Domain Overview: Focused on healthcare, pharmaceutical prescribing information, and regulatory compliance to ensure patient safety and trust.
4. Conducted Literature Survey: Reviewed existing approaches (traditional search engines, rule-based systems, basic chatbots) and identified their limitations in accuracy, personalization, and explainability.

Decisions Taken

1. Decided on a multi-agent architecture using CrewAI to handle query classification, retrieval, reasoning, deduplication, and answer generation.
2. Selected Gemini API for natural language understanding and answer generation instead of OpenAI.
3. Chose ChromaDB as the vector database for efficient retrieval and traceability. - Finalized SentenceTransformers for embedding generation.
4. Selected Streamlit for interactive frontend development.

Outcome

1. - Clear problem statement and objectives defined.
2. Literature survey completed to highlight gaps in existing systems.
3. Tech stack finalized and roles assigned to each team member.
4. Architecture diagram prepared and refined with mentor’s feedback.

Summary of Works Day 2
(21/08/2025)

Objectives

1. Develop the four key agents of the RAG-based system: Query Classifier, Retriever, Deduplication, and Reasoning & Domain Agent.
2. Implement data ingestion pipeline for handling PDFs, OCR processing, and structured storage.
3. Set up ChromaDB for persistent vector storage and retrieval.
4. Enable embeddings for semantic search using SentenceTransformer.

Activities Performed

1. Developed Query Classifier Agent for intent detection (dosage, interactions, side effects, etc.) and entity extraction (drug names).
2. Implemented Retriever Agent that performs vector similarity search using embeddings and returns top-k relevant chunks with section metadata.
3. Built Deduplication Agent to merge duplicate chunks and remove low-confidence results for clarity.
4. Created Reasoning & Domain Agent powered by Gemini API to analyze retrieved context and form reasoning-driven responses.
5. Designed Data Ingestion Agent: extracted raw text, figures, and tables from drug PDFs using PyMuPDF (fitz), Camelot, and Tesseract OCR for scanned content.
6. Implemented text preprocessing including line-break correction, hyphenation fixes, and chunking into manageable segments with overlap.
7. Generated vector embeddings using SentenceTransformer ('all-MiniLM-L6-v2') and stored them in ChromaDB for efficient retrieval.

Technical Details

1. PDF Handling: Extracted structured text and assets using PyMuPDF, Camelot for tables, and Pillow with Tesseract OCR for figures or scanned pages.
2. Embedding Model: SentenceTransformer (all-MiniLM-L6-v2) used for dense vector representation of text chunks.
3. Vector Database: ChromaDB set up with cosine similarity for efficient retrieval.
4. Gemini API: Integrated with the Reasoning Agent to refine and generate context-aware answers.
5. Chunking Strategy: Split text into 2000-character segments with 200-character overlap for balanced context and efficiency

Challenges & Resolutions

1. Handling scanned PDFs with very low text extraction quality – resolved using Tesseract OCR fallback.
2. Duplicate chunks leading to redundancy in retrieval – solved with Deduplication Agent.
3. Managing metadata consistency (page numbers, section headers) – introduced heading detection and metadata sanitization pipeline.

Outcome

Day 2 successfully established the foundation of our RAG-based drug information chatbot. By completing the agents and ingestion pipeline, we enabled end-to-end query understanding, retrieval, reasoning, and clean output generation with proper metadata. This setup ensures that further integration and enhancements can proceed smoothly in the next phases.

Summary of Works Day 3
(22/08/2025)

Objectives

1. Integrate all developed agents (Query Classifier, Retriever, Deduplication, Reasoning & Domain Agent) into a unified pipeline.
2. Implement vector search mechanism with ChromaDB for efficient retrieval.
3. Introduce confidence-based filtering to enhance reliability of responses.
4. Conduct initial testing of end-to-end query answering pipeline with drug-related queries.

Activities Completed:

1. Integrated Query Classifier output (intent + entities) with the Retriever for focused search in ChromaDB.
2. Retriever Agent successfully embedded queries and fetched top-k relevant chunks along with section metadata.
3. Deduplication Agent merged redundant chunks and removed low-confidence retrievals to clean up results.
4. Reasoning & Domain Agent analyzed the curated chunks and generated meaningful answers using Gemini API.
5. Implemented confidence threshold (set at 0.6) to ensure only reliable passages were passed to reasoning step.
6. Performed initial pipeline testing with queries like 'Humira dosage' and 'Can I take Rinvoq with aspirin?'.
7. Verified end-to-end flow: User Query → Classification → Retrieval → Deduplication → Reasoning → Final Answer.

Technical Workflow:

Step 1: User enters query (e.g., "What are the side effects of Humira?").
Step 2: Query Classifier detects intent (side effects) and extracts entity (Humira).
Step 3: Retriever embeds query and fetches top-k similar chunks from ChromaDB with metadata.
Step 4: Deduplication Agent removes repeated and irrelevant chunks, keeping only strong matches above threshold.
Step 5: Reasoning Agent leverages Gemini to form structured, natural-language answers.
Step 6: Final answer is displayed with citations (section numbers, page references).

Challenges & Resolutions

1. Low-confidence or irrelevant chunks were being retrieved – resolved with confidence threshold (0.6) filtering.
2. Pipeline synchronization between agents caused metadata mismatch – fixed by standardizing 'chunk_id' linking.
3. Initial outputs had redundancy due to repeated dosage entries – Deduplication Agent polished results.

Outcome

Day 3 achieved full integration of all four agents into a smooth RAG-based pipeline. The system can now take a drug-related query, classify it, retrieve relevant context, remove noise, and generate a clean, reliable, and explainable answer with proper citations. This marks the first complete functional version of our Drug Information Chatbot.

Summary of Works Day 4 (23/08/2025)

On Day 4, The focus was on enhancing the dataset quality and adding new features to improve the robustness of our Agentic Multi-Agent AI system. After successfully integrating the agents and vector search pipeline on Day 3, this stage aimed at refining the system by expanding the dataset, improving diversity, and enabling more advanced functionality.

Activities performed

Dataset Enhancement:

1. Expanded the dataset by including additional PDF documents and structured files.
2. Implemented OCR-based extraction for scanned PDFs to ensure text availability.
3. Applied data cleaning and deduplication to maintain consistency.
4. Added semantic variations of queries to improve retrieval accuracy.
5. Balanced domain coverage to avoid bias towards certain categories.

Feature Addition:

1. Introduced multi-query handling allowing the system to process multiple user queries simultaneously.
2. Integrated confidence-based ranking of retrieved results for improved reliability.
3. Added support for domain-specific reasoning (heritage tourism sustainability focus).
4. Developed an analytics module for query success rate and error monitoring.
5. Enhanced user interface with query suggestions and context-aware assistance.

Outcome:

Day 4, the system had become more robust, scalable, and feature-rich. Enhancements to the dataset improved retrieval accuracy, while the new features boosted usability, reliability, and domain relevance. This set the foundation for testing and refinement on Day 5.

Summary of Works Day 5
(24/08/2025)

On Day 5, the focus was on Testing, Debugging, and Refinement of the integrated RAG-based Drug Information Chatbot. After developing the agents and integrating them into a pipeline, it was critical to validate functionality, identify bugs, and ensure smooth end-to-end performance. The day's work centered on functional testing, error resolution, and performance optimization to guarantee reliability before the final demo.

1. Functional Testing:

1. Verified that the Query Classifier Agent correctly detects intents and entities from diverse medical queries.
2. Ensured that the Retriever Agent fetched top-k relevant results with associated section/citation details.
3. Checked that the Deduplication Agent successfully removed redundant or noisy results.
4. Validated that the Reasoning Agent generated coherent and context-aware answers using Gemini API.
5. Confirmed that ingestion pipeline correctly handled PDFs with OCR, tables, and images.

2. Debugging & Issue Resolution

During testing, multiple issues were identified and resolved:

1. OCR Noise: Some scanned PDFs introduced extra line breaks; fixed by implementing hyphenation & line-break cleanup.
2. Embedding Errors: Large documents caused memory issues; solved by batching embeddings in chunks of 32.
3. Duplicate Chunks: Identified overlapping passages; improved deduplication with SHA1-based hashing.
4. Low-Confidence Results: Implemented confidence threshold filtering (0.6 similarity) to remove weak matches.
5. Latency Issues: Optimized retrieval queries in ChromaDB and cached embeddings to reduce response time

3. Refinement Activities

The system was fine-tuned for reliability and user experience:-

1. Enhanced confidence-based ranking for more accurate results.
2. Improved section mapping to maintain traceability of answers.
3. Optimized database queries for faster retrieval.
4. Polished UI on Streamlit for a smoother demo experience.
5. Added better error handling to gracefully manage API failures or missing data.

4. Outcome

By the end of Day 5, the system was stable, reliable, and demo-ready. Key achievements include:

1. Fully functional end-to-end pipeline tested across multiple queries.
2. Debugged and resolved major issues related to ingestion, retrieval, and reasoning.
3. Improved confidence in the accuracy, traceability, and usability of results.
4. System prepared for final documentation and demonstration on Day 6.

Summary of Works Day 6 (25/08/2025)

On day 6, The project was dedicated to compiling the complete report, finalizing documentation, and delivering a live demonstration of our RAG-based Drug Information Chatbot. This stage represented the culmination of all development, integration, and testing efforts conducted over the previous days.

Activities Performed

1. Compiled daily progress into a structured project report.
2. Formatted and refined the final project documentation (Problem Definition, Domain Overview, Agents, Workflow, Testing).
3. Created presentation slides summarizing the system architecture, agents, workflow, and results.
4. Prepared demonstration setup including ingestion of sample drug PDFs and live query answering.
5. Conducted mock demos to ensure smooth delivery during the final presentation.
6. Delivered the final live demo showcasing the complete workflow of the system

Challenges Faced

Some challenges included ensuring formatting consistency across the report, compressing detailed workflows into presentation slides, and making the demo run smoothly without technical delays. To address these, the team conducted multiple dry runs and finalized a clear speaking sequence for the presentation.

Outcome

Day 6 marked the successful conclusion of the project. The team produced a well-documented final report, polished presentation slides, and a functional live demo that highlighted the technical depth and practical utility of the RAG-based Drug Information Chatbot. This stage demonstrated not only the technical achievements but also the collaborative effort of the entire team.