**RAG Pipeline Implementation**

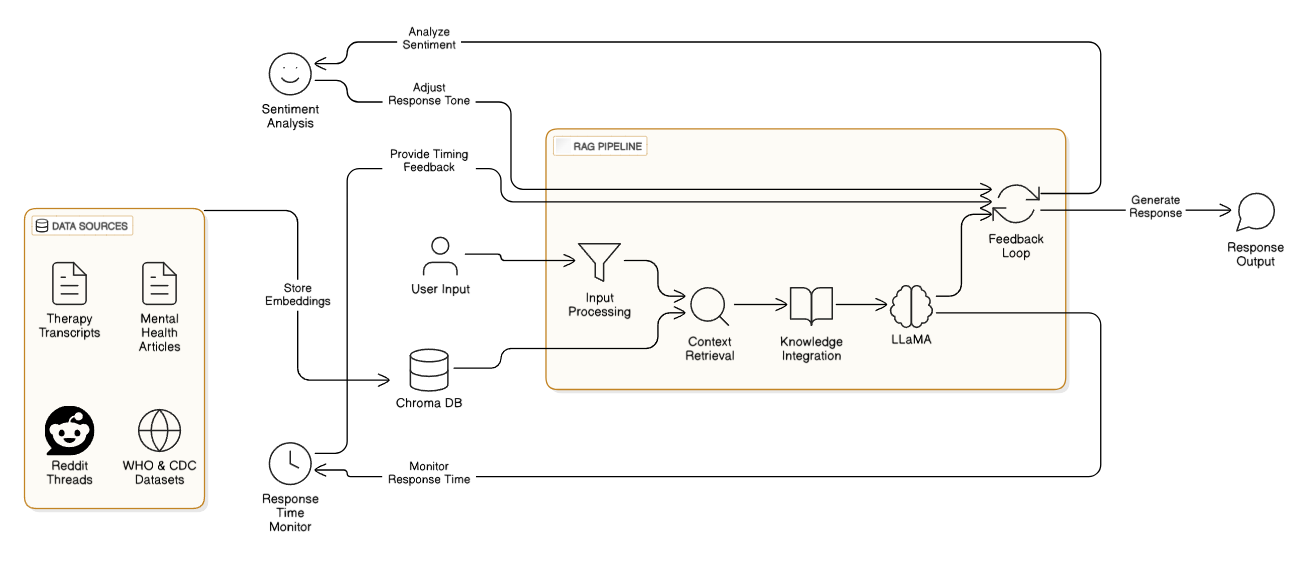
The system is designed around a straightforward but effective RAG pipeline. The architecture includes three key phases: document ingestion and embedding, vector-based retrieval, and contextual response generation.

**Data and Documents Used for Retrieval**  
The retrieval mechanism is powered by a single-source knowledge base: a plain text file titled **mental\_health\_resources.txt**. This document is composed of curated psychological advice, coping strategies, and mental health best practices sourced from credible public resources. These entries are segmented into logical chunks so that the chatbot can identify and retrieve only the most relevant passages during an interaction.

**Retrieval Implementation**  
To facilitate efficient and meaningful retrieval, **ChromaDB** is used as the vector database. The text data from mental\_health\_resources.txt is first split into manageable chunks using **LangChain's RecursiveCharacterTextSplitter**, which ensures logical sentence boundaries are maintained. Each chunk is then converted into an embedding using the all-MiniLM-L6-v2 model from the Hugging Face sentence-transformers library. These embeddings are stored in ChromaDB, enabling the system to conduct similarity searches and retrieve the most relevant context based on user queries.

**LLM Used for Generation**

The generation layer is powered by the open-source llama3 model, which is served locally through the **Ollama** framework. This choice allows the application to function entirely offline or in privacy-sensitive environments, making it highly suitable for personal use cases. The retrieved chunks from the knowledge base are appended to the user's question and sent to the model as contextual information to guide its response.

****