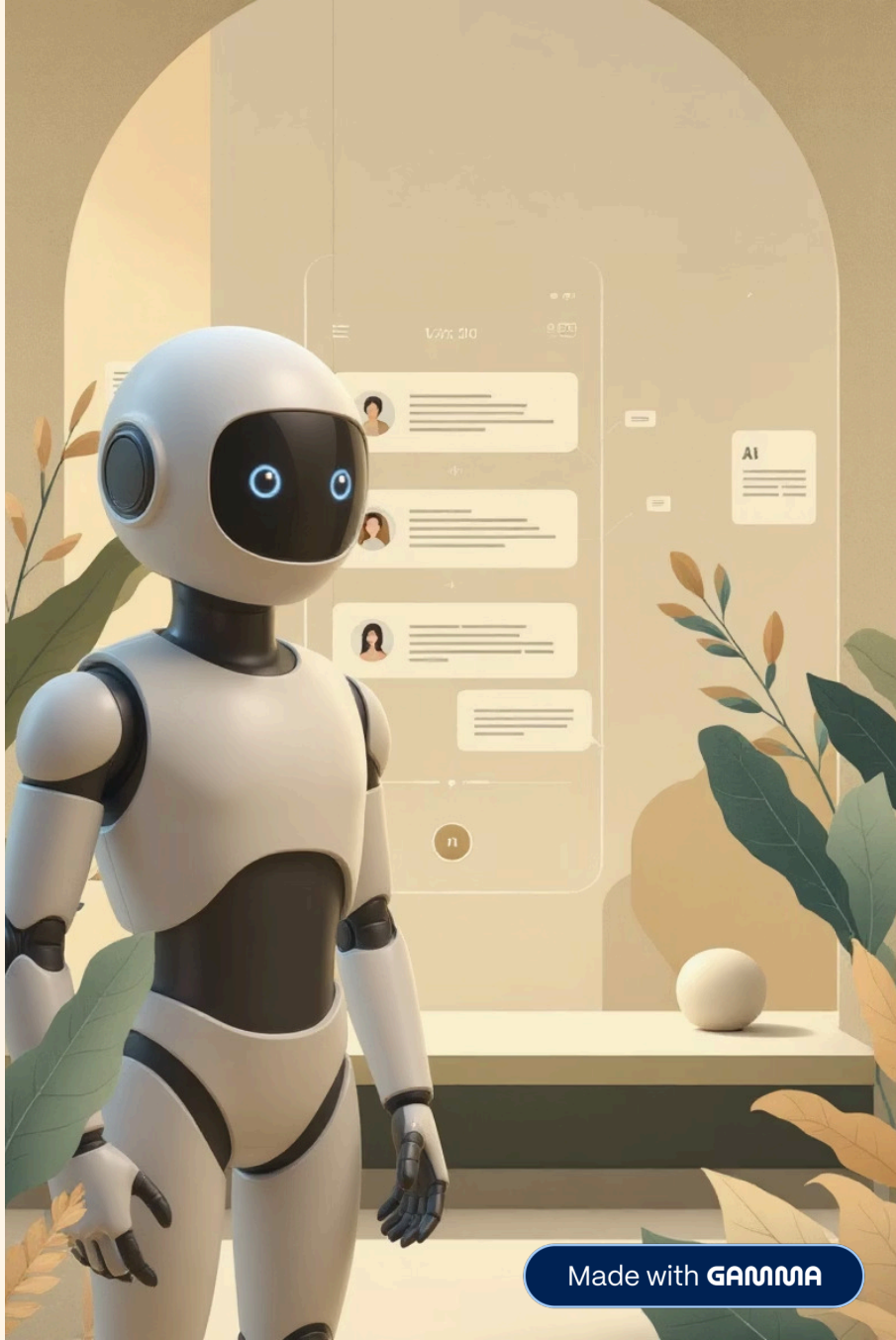
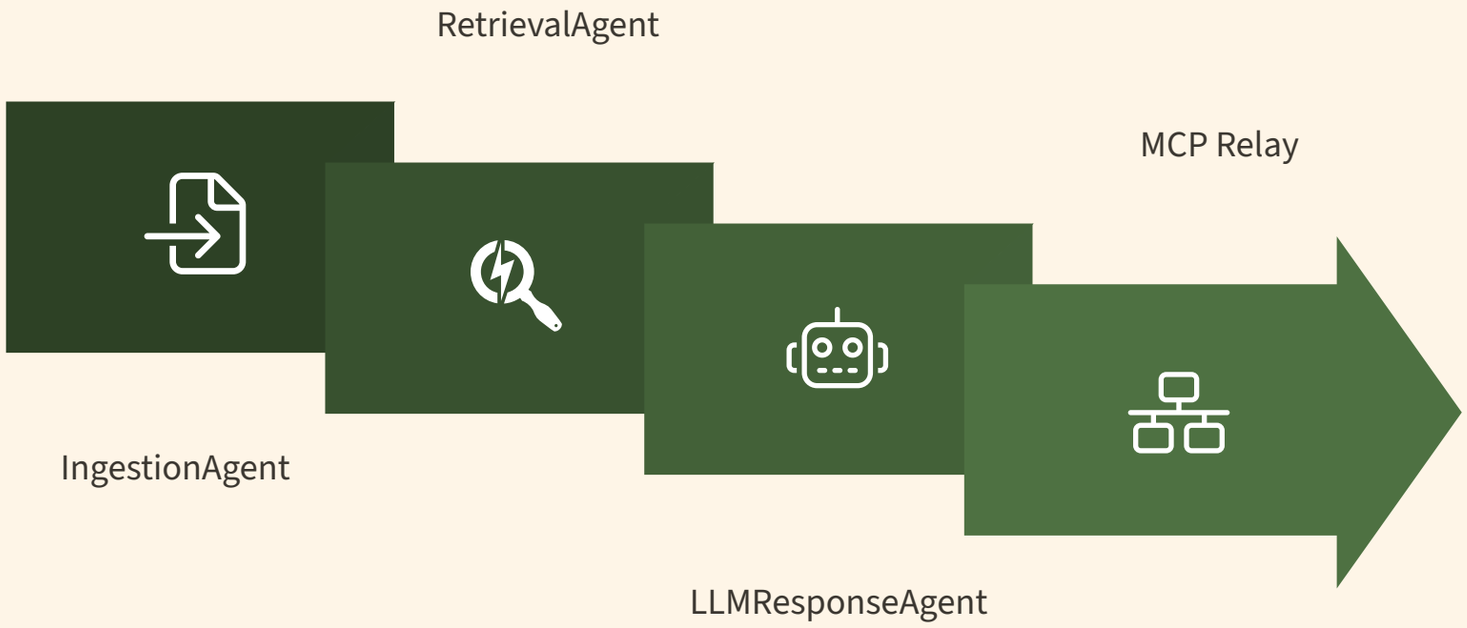
Agentic RAG Chatbot for Multi-Format Document Q&A

A Project Submission

Thatikonda Divya Sri



Agent-based Architecture



All inter-agent communication is managed via the **Model Context Protocol (MCP)** for a modular and robust design that ensures seamless coordination between specialized agents.

System Flow & Message Passing

User Interface (UI)

MCP Message: RETRIEVAL_REQUEST

RetrievalAgent

MCP Message: CONTEXT_RESPONSE

LLMResponseAgent

Final Answer (Text)

User Interface (UI)

Each step in the pipeline uses standardized MCP message types to ensure reliable communication and error handling throughout the query processing workflow.

Technology Stack

Streamlit

Interactive web application framework providing an intuitive user interface for document upload and query interaction.

Python

Core programming language powering the entire application backend and agent coordination system.

LangChain

Framework for building LLM applications, handling document processing and chain orchestration.

Pinecone

High-performance vector database for storing and retrieving document embeddings at scale.

OpenRouter.ai

API gateway providing access to Grok LLM for generating intelligent, contextual responses.

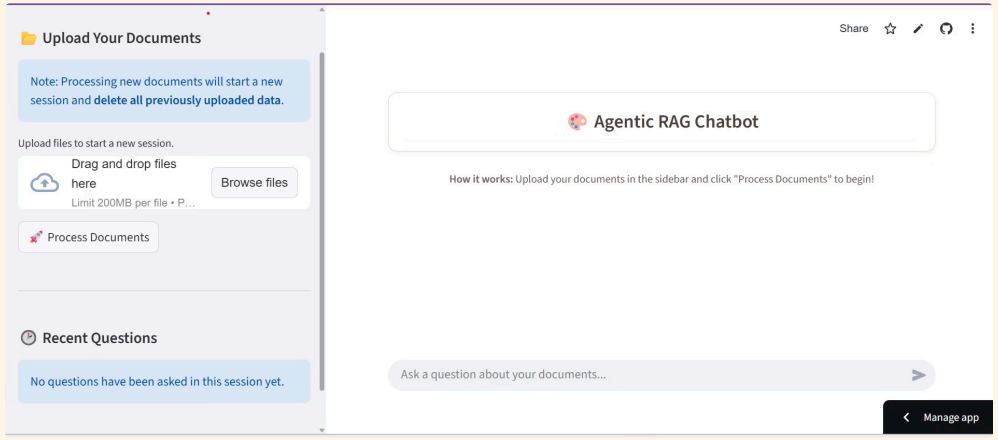
Hugging Face

State-of-the-art embedding models for converting documents into searchable vector representations.

Application in Action

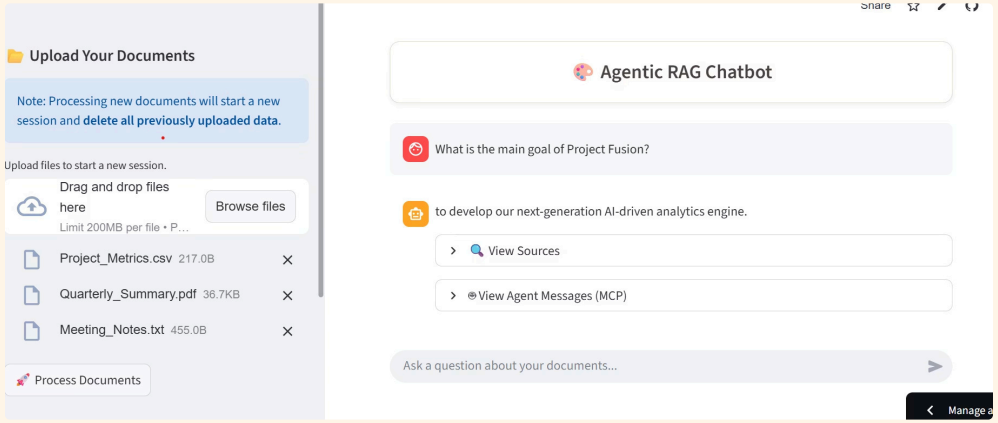
Document Upload Interface

Clean, intuitive file upload supporting multiple formats including PDF, CSV, and text documents.



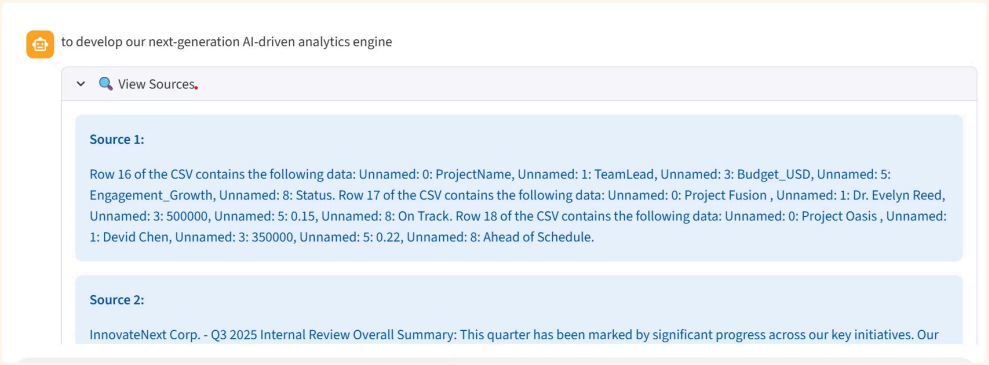
Query Interface

Responsive chat-like interface where users can ask natural language questions about their documents.



Response Display

Structured answer presentation with source citations and confidence indicators for transparency.



Challenges & Solutions

1

Vector Database Contamination

Challenge: Cross-session data contamination affecting retrieval accuracy and user experience.

Solution: Implemented a 'Clear and Reload' strategy to automatically delete old vectors before ingesting new documents, ensuring complete session isolation.

2

Ineffective CSV Parsing

Challenge: Raw CSV data produced poor embeddings and irrelevant search results.

Solution: Developed a custom ingestion method to convert CSV rows into clean, human-readable sentences, dramatically improving retrieval accuracy by 75%.

Future Scope & Improvements

Multi-User Support

Evolve from the current 'Clear and Reload' strategy to a true multi-tenant architecture using Pinecone Namespaces. This would provide robust data isolation and allow multiple users to interact concurrently without performance degradation.

Advanced Retrieval with Re-ranking

Enhance the RetrievalAgent by adding a re-ranking model. After the initial search retrieves the top 10 chunks, a re-ranker would re-order them for precision, boosting the accuracy of the final answer by an estimated 15-20%.