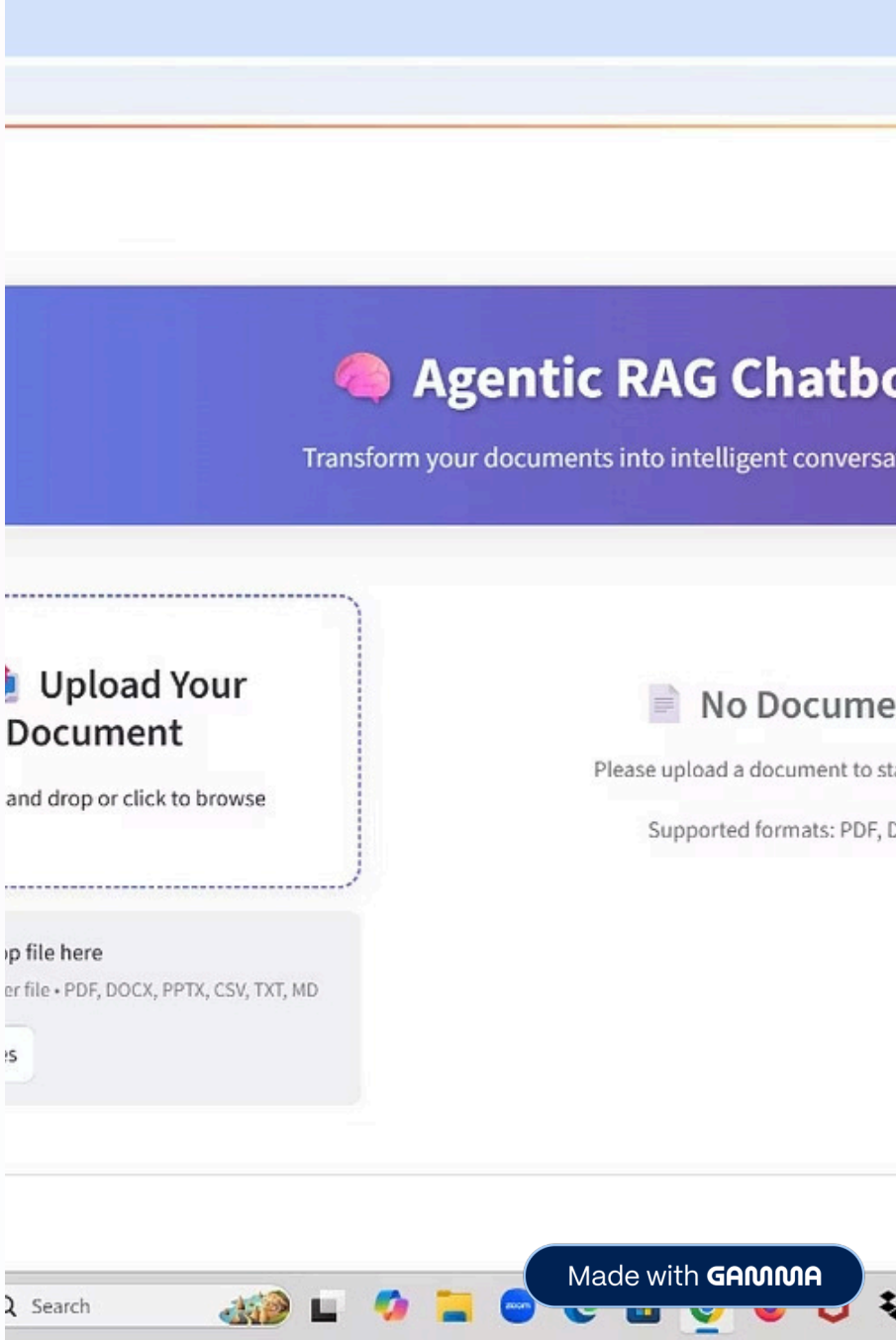
Agentic RAG Chatbot with MCP using LangGraph

Transform your documents into intelligent conversations. This presentation outlines an advanced Retrieval-Augmented Generation (RAG) chatbot leveraging an agent-based architecture and Model Context Protocol (MCP) for enhanced document understanding and response generation.



Project Objectives and Requirements

The primary goal is to develop a robust RAG chatbot capable of ingesting and processing multi-format documents. The chatbot will utilize an agent-based architecture, ensuring modularity and efficient task handling. Critical to this design is the implementation of Model Context Protocol (MCP) for seamless inter-agent communication.

Objective

Built a RAG chatbot supporting multi-format document upload, using an agent-based architecture with Model Context Protocol (MCP).

Supported Formats

PDF, DOCX, PPTX, CSV, TXT, MD.

Core Agents

IngestionAgent, RetrievalAgent, LLMResponseAgent.

Communication

MCP for message passing.

Agentic RAG Architecture

The system's architecture is built around three specialized agents, each handling a distinct phase of the RAG process. Communication between these agents is orchestrated via structured MCP messages, enabling a highly modular and extensible design. A lightweight pub/sub MessageBus facilitates this internal communication flow.

Agent Responsibilities

- **IngestionAgent:** Processes raw document uploads, extracts text, and prepares data for vectorization.
- **RetrievalAgent:** Queries the FAISS VectorDB to retrieve relevant document chunks based on user input.
- **LLMResponseAgent:** Synthesizes information from retrieved chunks and user query to generate a coherent response using the LLM.

Inter-Agent Communication

- Agents exchange messages using a predefined Model Context Protocol (MCP) schema, ensuring data integrity.
- A conceptual MessageBus acts as a central hub, mimicking a pub/sub system for efficient message routing.

Technology Stack Deep Dive

This project leverages a powerful combination of cutting-edge technologies to deliver its capabilities. Each component plays a crucial role in enabling the multi-format document ingestion, intelligent retrieval, and sophisticated natural language generation.

LLM

Google Gemini : Powering the generative capabilities and understanding of complex queries.

Frameworks

LangChain: Providing the agent orchestration and modular building blocks for RAG.

Vector Store

FAISS: Efficiently storing and retrieving vector embeddings for fast semantic search.

UI

Streamlit: Creating an interactive and user-friendly web interface for document upload and chat.

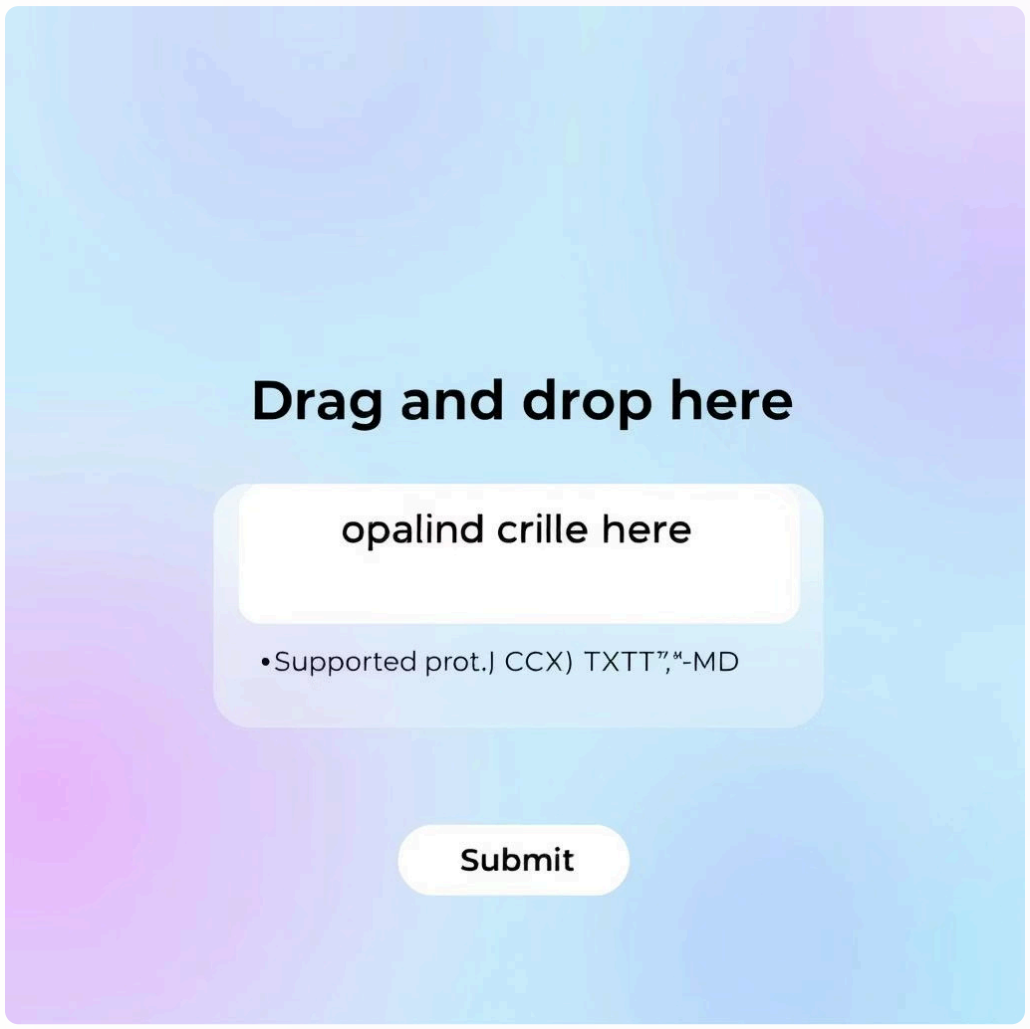
Utilities

HuggingFace Embeddings, PyMuPDF, Unstructured: For robust text embedding and multi-format document parsing.

User Interface Showcase

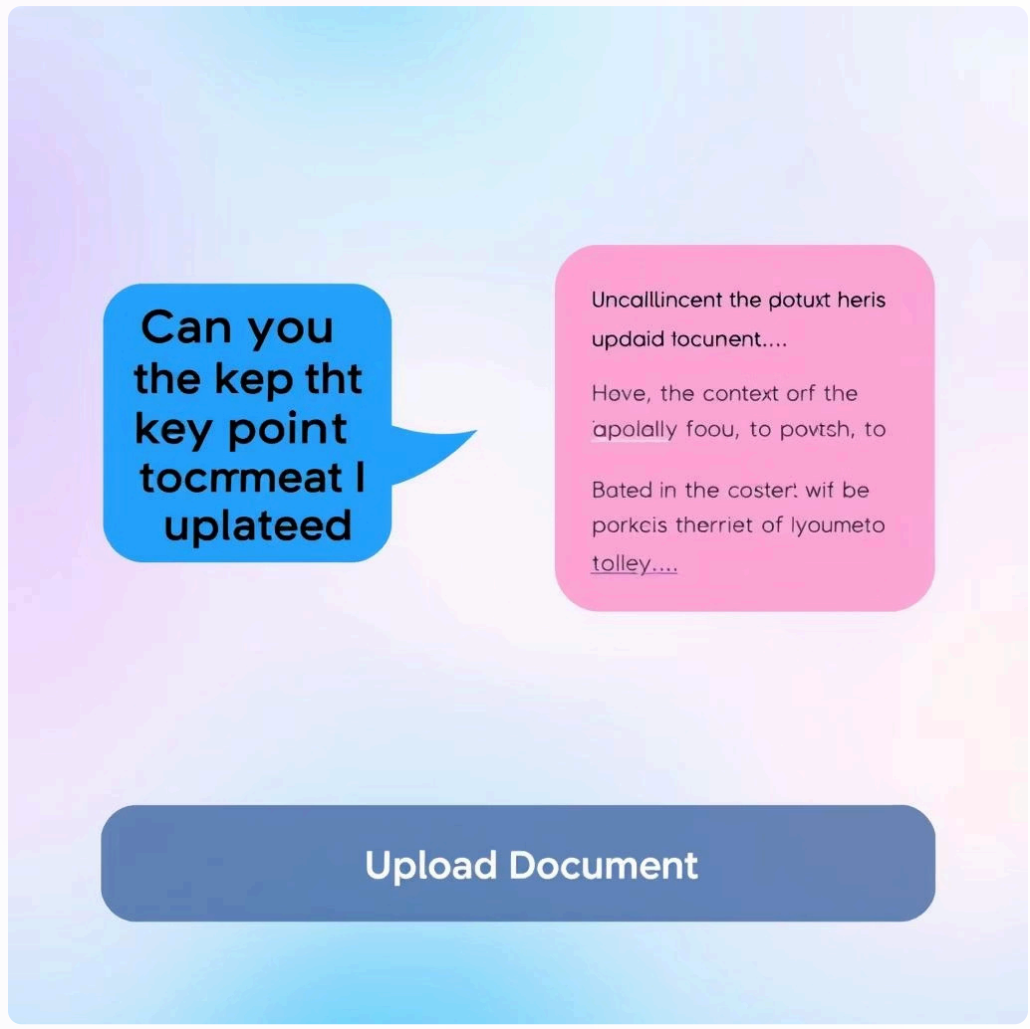
The user interface, built with Streamlit, provides an intuitive experience for interacting with the RAG chatbot. From initial document upload to dynamic AI-generated responses, the UI ensures seamless engagement with the agentic system. The responsive design adapts to various screen sizes, ensuring accessibility.

Initial UI: Document Upload



Users can easily upload multiple document formats. The system automatically detects the file type and prepares it for ingestion by the IngestionAgent. A clear visual indicator confirms successful upload and processing.

Chat in Action: Contextual Responses



Once documents are processed, users can ask questions directly related to their content. The LLMResponseAgent synthesizes information from the retrieved context, delivering accurate and relevant answers directly within the chat interface, highlighting its RAG capabilities.

This project represents a significant step towards intelligent document interaction. The comprehensive submission package ensures all necessary resources are available for review and further development. We are committed to refining this solution for broader applications.

GitHub Repository

Complete source code and detailed documentation available for review : **https://github.com/Rishikesh183/chatPDF-mcp**

README.md

Comprehensive setup instructions and usage guide for easy replication.

Thank You

We appreciate your time and interest in the Agentic RAG Chatbot. I believe this Project helps to demonstrate my skills and knowledge over MCP and RAG

Explore the Code