Agentic RAG Chatbot for Multi-Format Document QA

Ajay Kumar | July 2025 | GitHub Submission

System Architecture Overview

- User sends query via Streamlit UI
- Input passes through LangChain agents with MCP (Model Context Protocol)
- Documents are chunked, embedded, and stored in FAISS
- Agentic workflow routes through Retriever → RAGChain → Prompt Template → OpenAI LLM
- Response shown in UI

Tech Stack Used

- LangChain (Agents, Chains, Tools)
- FAISS for Vector Store
- OpenAI API (GPT 3.5/4)
- Streamlit (for UI)
- Python (Backend logic)
- MCP (Model Context Protocol for context routing)

MCP Message Flow

- Example JSON:
- {
- "from": "retriever",
- "to": "ragchain",
- "data": "top 3 relevant chunks"
- }
- → Follows step-by-step agent conversation model
- → Maintains context flow and decision memory

UI Preview

- Screenshot of Chatbot Interface (added manually in README)
- Paste the screenshot image in this slide if needed.

Challenges & Future Scope

- Implementing multi-agent coordination with MCP
- Document conversion: PDF, DOCX, TXT formats
- Scalability: multiple queries and sessions
- Future: Add voice input, chatbot memory, file upload interface