Assignment: RAG-Based Chat System

Objective

Build a Retrieval Augmented Generation (RAG) chat system that can input multiple document types and provide both chat and deep research functionalities.

Technical Requirements

Frontend

- Use any UI framework/library of your choice
- Options: Open WebUI, Streamlit, React, Vue.js, or any UI SDK

LLM Integration

- Integrate any LLM provider: Ollama, vLLM, OpenAI, Gemini, etc.
- Use any LLM model (3B-8B models also work), we want to see your approach.

Vector Database

- Implement any vector storage solution: Chroma, Pinecone, Neo4j, Weaviate, etc.
- Efficient document retrieval and similarity search

Document Processing

Support documents such as:

- PDF files
- PowerPoint presentations (PPT/PPTX)
- CSV and Excel files
- Word documents (DOC/DOCX)
- Text files (TXT)

Core Features

1. **Document Upload & Processing**
    - Bulk document upload capability
    - Automatic text extraction and chunking
    - Vector embeddings generation and storage
2. **Standard Chat**
    - Query uploaded documents
    - Contextual responses based on content
    - Chat history management

3. **Deep Research**
    - This is an open-ended task, would love to see your implementation of Deep Research.

Deliverables

GitHub Repository

Create a public GitHub repository containing:

- The source code
- Clear README with:
    - Setup and installation instructions
    - Architecture overview
    - Features demonstration
    - Technology stack used

Evaluation Criteria

- Code quality and organization
- Documentation clarity
- Technical approach and architecture decisions
- Feature completeness
- Innovation and problem-solving approach

Submission

- Email your GitHub repository link to sushant@dreate.ai
- Include a brief summary of your technical choices and challenges faced
- Timeline: Submit within 5 days of receiving this assignment