Project -Advanced RAG Agent Chat System

Target Project Name: Advanced RAG Agent Chat System

Project Goal: To implement a comprehensive advanced RAG system that is production-ready, featuring cutting-edge RAG techniques. This system allows users to chat with an AI assistant that answers questions grounded in uploaded documents and external sources.

Core Tech Stack: Python/FastAPI (Backend), **OpenAI Agents** for advanced reasoning, **ChromaDB** (Production Vector DB), **FAISS** (Demo Vector DB), **PostgreSQL 16**, and **OpenAI o3** (with reasoning) / **GPT-4.1-mini** models, **LangChain** (for demonstrations).

Learning Outcomes: Upon completion, you will be able to architect and implement an **Advanced RAG Pipeline** featuring **Hybrid Search**, **Query Optimization (HyDE)**, and **LLM-based Reranking**.

Portfolio Highlight	Description
Advanced RAG Pipeline	Implemented the **Query Optimization Layer (HyDE)**, **Hybrid Search Layer (RRF)**, and **Reranking Layer (LLM-based)**, achieving up to **+30% accuracy** improvements compared to Naive RAG.
Agentic Orchestration	Utilized **OpenAI Agents (v0.0.14)** and the **o3 model with reasoning** to dynamically plan and execute the RAG workflow via **Tool Calling**.
Engineering & Deployment	Built on **Python FastAPI** and deployed using **Docker Compose**. Implemented **WebSocket** for real-time, streaming AI responses.
Data Intelligence	Used **Vision-based parsing (GPT-4.1-mini)** to handle complex document structures (tables, headers), ensuring **metadata-rich chunking** and high data quality.
Model Customization Concept	Understood the need for model customization and mastery of **PEFT/LoRA** techniques, recognizing they offer cost-efficient specialization without increasing inference latency.