Day 3 – Modular Enterprise Architecture (Detailed Revision)

1. What Was the Goal of Day 3?

The goal of Day 3 was to move from a single-file RAG script to a modular, production-style architecture. Instead of putting all logic in one file, we separated responsibilities into different layers.

2. Why Modular Architecture Matters

In real systems, components evolve independently. The embedding model might change, the vector database might change, or the LLM provider might change. If everything is tightly coupled, updates become risky and difficult. Modular design allows independent modification without breaking the entire system.

3. Separation of Concerns (Core Concept)

Separation of concerns means each part of the system has one responsibility. Example: The embedding service only converts text into vectors. The retriever only performs similarity search. The LLM service only generates answers.

4. VectorStore (Core Layer)

The VectorStore handles document indexing, persistence, and file hash validation. It ensures embeddings are built once and reloaded when needed. This mirrors how production systems manage indexing.

5. EmbeddingService

EmbeddingService loads the embedding model and provides a clean method to encode text. If we switch to OpenAI embeddings in the future, we only modify this file.

6. RetrieverService

RetrieverService calculates cosine similarity and selects top-k relevant chunks. This isolates retrieval logic from generation logic.

7. LLMService

LLMService communicates with the LLM (Ollama in this case). It builds prompts and generates responses. If we switch LLM providers, only this service changes.

8. main.py (Orchestration Layer)

main.py wires everything together. It initializes services and controls the application flow. In production, this layer would be replaced or wrapped by an API server like FastAPI.

9. Why This Is Enterprise-Level

This structure supports scalability, testing, maintainability, and deployment. Each component can be independently tested or replaced.

10. Example Scenario

If tomorrow we move from local Ollama to a cloud LLM API, we only update LLMService. The retriever and vector store remain unchanged. This is the power of clean architecture.

Final Understanding After Day 3

You now understand how to design AI systems with modular architecture. You can separate indexing, retrieval, and generation layers. This prepares you for adding memory, agents, and API deployment in future stages.