LLM Orchestration

This session introduces how modern LLM-based applications are structured beyond a single prompt. We will look at how multiple components work together to build reliable, production-ready AI systems.

What We Will Cover

1. LLM as Judge

- Using LLMs to evaluate outputs
- Scoring answers for relevance and correctness
- Basic re-ranking concepts in retrieval systems

---

2. Why Orchestration is Needed

- Limitations of single-prompt systems
- Managing multi-step workflows
- Handling tools, retrieval, memory, and evaluation

---

3. LLM Pipelines and Chaining

- Sequential workflows
- Conditional logic in LLM flows
- Multi-step reasoning patterns

---

4. LangChain Architecture (Conceptual Overview)

We will understand the core building blocks:

- Chains
- Prompts
- Memory
- Tools
- Retrievers

Focus will be on how these pieces fit together in a real application.

---

5. Integration with Vector Databases

- Why embeddings are required
- Basic RAG flow
- How retrieval connects to generation

What You Should Understand After This Session

By the end of this session, you should be able to:

- Explain why orchestration is necessary in LLM applications
- Describe a basic RAG pipeline
- Identify components like retrievers, memory, and tools
- Understand how LLMs can evaluate other LLM outputs

What I Expect From You in the Next Session

Please come prepared to:

1. Think of one real-world use case (ecommerce, healthcare, fintech, education, etc.).
2. Identify where:
    - Retrieval would be needed
    - Memory might be useful
    - A tool call would improve the system
3. Sketch a simple pipeline (even rough boxes and arrows are enough).

We will build on this in the next session and move from conceptual understanding toward system-level thinking.