**Mastering Generative AI: LLMs, RAG Systems, and Agentic Intelligence**

Module 1: Foundations of LLMs and Generative AI

 Evolution of language models: GPT, PaLM, LLaMA, Claude

 Architecture breakdown: transformers, embeddings, attention

 Pretraining vs. fine-tuning vs. instruction-tuning

 Ethical foundations and governance strategies

 📘 Lab: Build a mini GPT-like transformer using HuggingFace

Module 2: Retrieval-Augmented Generation (RAG) Systems

 Why RAG: context-aware generation and limits of closed-book LLMs

 Vector databases (FAISS, Weaviate, Chroma) + embedding techniques (FastEmbed, OpenAI, Cohere)

 LangChain pipeline construction (multi-folder ingestion, chunking, retriever + LLM orchestration)

 Optimization: latency, relevance, and grounding

 🔧 Workshop: Develop a scalable RAG chatbot with fallback and summarization layers

Module 3: Agentic AI and Autonomous Task Planning

 What makes an agent: memory, planning, tools, reflexes

 Frameworks: LangChain agents, CrewAI, AutoGPT, OpenAgents

 Tool invocation, chaining, and adaptive goal refinement

 Guardrails, risk management, and ethical deployment

 🤖 Lab: Deploy an agent that autonomously updates a report by parsing PDFs and searching the web

Module 4: Strategy, Platformization, and Real-World Impact

 Architecting AI platforms for scale and resilience

 POD team models for GenAI delivery

 Self-healing systems and proactive orchestration

 Cross-functional collaboration and KPI alignment

 🎯 Capstone: Propose and prototype an enterprise-grade AI agent ecosystem tailored to real org needs

**🛠 Bonus Tracks**

* LangChain Deep Dive: Callbacks, Prompts, Streaming
* Advanced RAG: Hybrid search, reranking, citation injection
* Strategic Playbooks: Risk governance for AI deployment

Absolutely, Naveen — let’s design a hands-on course that builds not just comprehension but implementation muscle across LLMs, Retrieval-Augmented Generation (RAG), and Agentic AI. Given your technical command over LangChain, FAISS, and embedding pipelines, we’ll anchor this in practical labs, sandbox builds, and deployable artifacts.

**🛠️ Course Title**

**“Applied AI Systems: Hands-On with LLMs, RAG and Autonomous Agents”**

**📚 Course Overview**

Structured over **5 project-driven modules**, each with code labs, architectural design, and evaluative checkpoints.

**Module 1: LLM Fundamentals + Customization**

**Concepts**

* Tokenization, embeddings, transformer anatomy
* Differences between base, fine-tuned, and instruction-tuned models
* Evaluation metrics for generative output

**Hands-On**

* Deploy GPT2/BERT using HuggingFace
* Train a small transformer from scratch on a custom corpus
* Fine-tune an open-source model with instruction prompts

**Tools**

* 🤖 HuggingFace Transformers, SentenceTransformers, Weights & Biases

**Module 2: Building RAG Pipelines (Retrieval-Augmented Generation)**

**Concepts**

* What RAG solves: grounding, hallucination, domain adaptation
* Document ingestion, chunking strategies
* Embed + index → retrieve → generate workflow

**Hands-On**

* Use LangChain + FastEmbed to build a vector store from multi-folder PDFs
* Integrate FAISS/Chroma retriever with OpenAI/Cohere LLM
* Add metadata filters, hybrid search, summarization post-processing

**Tools**

* 📦 LangChain, FAISS, Chroma, FastEmbed, Streamlit

**Module 3: Agentic AI & Task-Chaining**

**Concepts**

* Reflex vs memory-driven agents
* Tool calling, planning, adaptive loops
* Risk handling and error boundaries

**Hands-On**

* Build LangChain Agent with toolset: Google Search, Calculator, PDF parser
* Use CrewAI to assign roles, goals, and task graphs
* Build an Agent that reads legal PDFs and updates a report autonomously

**Tools**

* 🧠 LangChain Agents, CrewAI, Serper.dev, LlamaIndex

**Module 4: Deployment and Evaluation**

**Concepts**

* Latency optimization, guardrails, caching
* Prompt templates, fallback logic
* Safety and explainability strategies

**Hands-On**

* Deploy RAG Agent as a REST API
* Add LangSmith callbacks and tracing
* Use Streamlit to build an interactive chat UI with live metrics

**Tools**

* 🚀 LangServe, LangSmith, Streamlit, Docker

**Module 5: Strategy, Platformization, and Capstone**

**Capstone Project** Design and build a **modular GenAI agent** for a real-world use case (e.g., legal assistant, incident responder, policy checker). Includes:

* Use-case mapping → design → architecture → deployment
* Trade-off reasoning: latency vs grounding vs autonomy
* Documentation and org-level presentation

**🧩 Optional Add-Ons**

| **Track** | **Focus** |
| --- | --- |
| 📊 Governance Lab | Ethical risk analysis, compliance design |
| 🔍 RAG+ | Citation injection, reranking, provenance checks |
| 📦 POD Enablement | Strategic alignment, cross-functional delivery models |

If you’d like, I can turn this into a GitHub repo template with starter notebooks, sandbox setups, and README walkthroughs — or even align it to onboarding tracks for internal teams. Just say the word.