### 🎯 Objective:

Master the next level of GenAI — including model fine-tuning, retrieval-augmented generation (RAG), intelligent agent systems, and safety strategies. This phase prepares you for enterprise AI, production use-cases, and innovation-level work.

## 🔍 Curriculum Breakdown

### 1. Fine-Tuning Language Models

#### 📌 1.1 PEFT (Parameter-Efficient Fine-Tuning)

* Overview: updating only a small subset of model parameters
* LoRA: inserts trainable low-rank matrices into frozen layers
* QLoRA: quantized LoRA for low-resource environments
* Adapters: modular plug-in layers trained per task

#### 📌 1.2 Fine-Tuning Options

* Full fine-tuning vs. PEFT
* Trade-offs: memory, compute, dataset size
* Saving & sharing with Hugging Face Hub

#### 📌 1.3 Tools & Frameworks

* Hugging Face PEFT + Transformers
* bitsandbytes for quantization
* LLaMA.cpp, GGML for lightweight deployment

📌 Mini Project: Fine-tune a 7B+ model with QLoRA on a custom task (e.g., Q&A or summarization)

### 2. Advanced Training Techniques

#### 📌 2.1 RLHF – Reinforcement Learning from Human Feedback

* Learning from ranked outputs and preferences
* Architecture: Supervised fine-tuning → reward model → policy optimization
* OpenAI’s use case in ChatGPT

#### 📌 2.2 DPO – Direct Preference Optimization

* Simplifies RLHF: uses preference pairs directly for training
* More sample-efficient and stable than PPO

#### 📌 2.3 Instruction Tuning & Constitutional AI

* Tuning models on instructional datasets (e.g., FLAN, Dolly)
* Anthropic's Constitutional AI: aligning LLMs with ethical frameworks

📌 Notebook: Visualize preference data → apply DPO → evaluate response quality

### 3. Retrieval-Augmented Generation (RAG)

#### 📌 3.1 Why RAG?

* Combine LLM generation with custom external knowledge
* Solve LLM hallucination and fact inconsistency

#### 📌 3.2 RAG Workflow

* Query → embedding → vector search → context retrieval → prompt injection → answer
* Support for PDFs, websites, internal knowledge bases

#### 📌 3.3 Key Components

* Embedding Models: BGE, OpenAI, SentenceTransformers
* Vector DBs: FAISS, Pinecone, ChromaDB
* Orchestration: LangChain, LlamaIndex

📌 Hands-On: Build a RAG chatbot that answers from uploaded documents

### 4. AI Agents & Multi-Agent Systems

#### 📌 4.1 Agent Frameworks

* AutoGPT, BabyAGI: LLM-based task agents
* Pluggable tools (browsing, calculator, file manager)
* Role memory and goal decomposition

#### 📌 4.2 Architectures

* Reactive: act based on current state only
* Deliberative: internal state + planning
* BDI (Belief-Desire-Intention): structured decision making

#### 📌 4.3 Multi-Agent Collaboration

* A2A (Agent-to-Agent communication)
* MCP (Multi-Agent Control Protocols)
* Use-cases: collaborative writing, research, design

📌 Lab: Create a 2-agent system: one for planning, one for execution, working on a shared task (e.g., blog generation)

### 5. Application Areas (Real-World)

| Domain | Use Cases |
| --- | --- |
| Code | Autocomplete, bug fixing, test generation (Codex, Copilot) |
| Science | Drug discovery, protein folding (AlphaFold, GenSLMs) |
| Creativity | Visual art, storytelling, music composition (MuseNet, MidJourney, SDXL) |
| Search & Productivity | Semantic search, smart emails, summarization |

📌 Challenge: Build a GenAI app for your chosen domain with at least one RAG or Agent capability

### 6. Ethics, Alignment & AI Safety

#### 📌 6.1 Model Bias & Safety

* Real-world risks: discrimination, misinformation
* Detection of offensive, biased, or harmful outputs

#### 📌 6.2 Red Teaming & Testing

* Adversarial inputs: jailbreak prompts, misleading prompts
* Safety checkers, filters, moderation pipelines

#### 📌 6.3 Transparency & Interpretability

* Feature attribution, model explanations
* Explainable AI (XAI) for large language models

📌 Reflection Exercise: Analyze a GenAI system for safety & ethical flaws and propose mitigations

### 7. Tools & Platforms

#### ✅ Fine-Tuning & Experimentation

* Hugging Face Transformers & PEFT
* bitsandbytes, Accelerate, AutoTrain

#### ✅ Retrieval & Tool Use

* LangChain, LlamaIndex
* FAISS, Pinecone, ChromaDB

#### ✅ Agent Frameworks

* Auto-GPT, BabyAGI, CrewAI (experimental)
* Plug-in tools (Google Search, Python, file systems)

#### ✅ Tracking & Monitoring

* Weights & Biases
* TensorBoard, MLflow

#### ✅ APIs & Hosted Models

* OpenAI, Anthropic, Cohere, Together.ai

### 🧪 Capstone Project

Project: *Build an Intelligent GenAI Assistant with Memory + Retrieval*

* Use RAG for custom knowledge access
* Add tool-use via LangChain agents or AutoGPT
* Include prompt logic + memory
* Implement safety filters and logging
* Share app + source on GitHub and demo link