# Course Syllabus: Fine-tuning & RAG Specialist

**Course Title:** LLM Customization: Mastery in Fine-tuning and Retrieval-Augmented Generation (RAG)

**Target Audience:** This course is for AI engineers, machine learning engineers, and data scientists with a strong foundation in deep learning frameworks (PyTorch or TensorFlow), and a solid understanding of LLM basics.

**Course Level:** Expert.

**Duration:** 10 Weeks

**Course Description:** This curriculum is a deep dive into the two most critical techniques for customizing LLMs: **fine-tuning** and **Retrieval-Augmented Generation (RAG)**. You will learn the theoretical underpinnings of each method, their trade-offs, and how to build, evaluate, and deploy production-grade systems. The course emphasizes practical skills in data preparation, model optimization, and architecting robust RAG pipelines. By the end, you will be able to select the right strategy for any business problem and implement it with expert-level proficiency.

---

## Learning Objectives

Upon successful completion of this course, students will be able to:

- Differentiate between fine-tuning and RAG, and determine the optimal approach for a given use case.
- Design and execute data preparation pipelines for both fine-tuning and RAG systems.
- Apply various fine-tuning techniques, including Parameter-Efficient Fine-Tuning (PEFT) like LoRA, on large models with limited resources.
- Architect and implement scalable RAG pipelines, including advanced retrieval strategies and vector database management.
- Evaluate the performance of fine-tuned and RAG systems using a variety of specialized metrics.
- Implement a hybrid fine-tuning and RAG solution to achieve superior performance in domain-specific applications.
- Deploy and monitor LLM customization pipelines in a production environment.

---

## Course Structure: A Step-by-Step Learning Path

### Part 1: The RAG Specialization (Weeks 1-4)

This section focuses on the end-to-end process of building, optimizing, and evaluating Retrieval-Augmented Generation systems.

#### Week 1: RAG Fundamentals & Architecture

- The RAG paradigm: why it's essential for grounded and up-to-date responses.
- The RAG pipeline: from document ingestion to response generation.
- Key components: chunking strategies, embedding models, vector databases, and the LLM.
- **Hands-on Lab:** Build a basic RAG system using an open-source framework like **LangChain** or **LlamaIndex**.

#### Week 2: Advanced Retrieval Techniques

- **Chunking optimization:** techniques for splitting documents to preserve context.
- **Hybrid search:** combining keyword search (BM25) with semantic search.
- **Re-ranking:** using models to reorder retrieved documents for better relevance.
- **Hands-on Project:** Upgrade your RAG system to use a hybrid search and re-ranking pipeline.



#### Week 3: Vector Databases & Data Management

- Deep dive into vector databases (e.g., **Pinecone**, **ChromaDB**, **Weaviate**).
- Indexing, querying, and updating a vector store at scale.
- Data engineering for RAG: building pipelines to manage the knowledge base.
- **Hands-on Project:** Deploy a vector database and build an automated pipeline to ingest and embed documents for your RAG system.

#### Week 4: Evaluating & Optimizing RAG

- The unique challenges of evaluating RAG systems.
- Key metrics: context relevance, faithfulness, and answer correctness.
- Using specialized evaluation frameworks (e.g., RAGAs) to automate the process.
- **Hands-on Project:** Implement an evaluation pipeline to measure and optimize the performance of your RAG application.

---

### Part 2: The Fine-tuning Specialization (Weeks 5-7)

This section focuses on the technical process of fine-tuning models, from data preparation to resource-efficient training.

**Week 5: Fine-tuning Fundamentals & Data Preparation**

- Fine-tuning vs. RAG: when to choose one over the other.
- Data preparation: creating high-quality, task-specific datasets for fine-tuning.
- **Instruction fine-tuning:** formatting data for instruction-following models.
- **Hands-on Lab:** Clean and format a custom dataset for a fine-tuning task.

**Week 6: Parameter-Efficient Fine-tuning (PEFT)**

- The challenge of fine-tuning large models.
- Introduction to PEFT: LoRA, QLoRA, and other adapter-based methods.
- Using frameworks like **Hugging Face PEFT** to fine-tune a model with minimal compute resources.
- **Hands-on Project:** Fine-tune an open-source LLM for a specific tone or style using a PEFT method.

**Week 7: Full Fine-tuning & Evaluation**

- The full fine-tuning process: training the entire model on a custom dataset.
- Evaluating fine-tuned models: perplexity, accuracy, and human evaluation.
- **Hands-on Lab:** Conduct a full fine-tuning run on a smaller model and evaluate its performance.

---

**Part 3: Hybrid Systems & Production Deployment (Weeks 8-10)**

This final section focuses on combining both techniques and the MLOps principles needed to deploy these systems professionally.

**Week 8: Hybrid Architectures**

- When to combine RAG and fine-tuning: the best of both worlds.
- Architecting a system where a fine-tuned model acts as the "generator" in a RAG pipeline.
- **Hands-on Project:** Build a hybrid system that uses a fine-tuned LLM and a RAG pipeline to answer domain-specific questions with high accuracy.

**Week 9: MLOps for RAG & Fine-tuning**

- The MLOps lifecycle for LLM customization.
- **Containerization with Docker** for consistent and reproducible deployments.
- Monitoring model performance, data drift, and cost in production.
- **Hands-on Lab:** Dockerize your hybrid RAG and fine-tuning application.

**Week 10: Final Capstone Project**

- **Capstone Project:** Design, build, and deploy a comprehensive LLM solution that addresses a specific business problem using a combination of fine-tuning and RAG.
- Present the project, including a detailed analysis of your design choices, performance metrics, and a comparison of different approaches.

---

## Assignments & Grading

- **Weekly Hands-on Labs:** 20%
- **Intermediate Projects (Weeks 4 & 6):** 30%
- **Final Capstone Project:** 40%
- **Code Quality & Documentation:** 10%