Course Syllabus: LLM Application Engineer

Course Title: LLM Application Engineering: Building Production-Ready Applications

Target Audience: This course is for experienced software developers and engineers with a strong foundation in Python and a desire to specialize in building applications powered by Large Language Models (LLMs). This is not an introductory course to machine learning.

Course Level: Advanced Intermediate to Expert.

Duration: 12 Weeks

Course Description: This curriculum provides a comprehensive, project-based pathway to becoming an LLM Application Engineer. The focus is on the software engineering principles and MLOps practices needed to take a generative AI idea from a prototype to a robust, scalable, and secure production application. You'll master the use of frameworks, advanced prompt engineering, data integration, and deployment pipelines, with a strong emphasis on real-world challenges like cost optimization, model monitoring, and security.

Learning Objectives

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

- Architect and develop full-stack applications that effectively leverage LLMs.
- Master advanced prompt engineering techniques and apply them programmatically.
- Build sophisticated data pipelines to augment LLMs with private and real-time data.
- Implement MLOps best practices for LLM deployment, monitoring, and maintenance.
- Optimize LLM applications for cost, performance, and scalability.
- Secure LLM applications against common vulnerabilities like prompt injection.
- Develop and manage a portfolio of professional-grade LLM applications.

Course Structure: A Step-by-Step Learning Path

Part 1: Foundational Skills & Development (Weeks 1-4)

This section builds the core development skills for LLM-based applications, including framework usage and advanced prompting.

Week 1: LLM Fundamentals for Developers

- The LLM landscape: an overview of key models (GPT, Gemini, Llama, etc.).
- Understanding tokenization, context windows, and model parameters.
- The developer's role: from model trainers to application builders.
- **Hands-on Lab:** Use a Python SDK to interact with an LLM API and explore key parameters.

Week 2: Advanced Prompt Engineering

- **The anatomy of a prompt:** role, task, context, and format.
- Advanced prompting techniques: Chain-of-Thought, few-shot prompting, and persona.
- **Prompt templating** and management for dynamic applications.
- **Hands-on Lab:** Develop a set of advanced prompts for a complex problem, like generating a structured report.

Week 3: LLM Orchestration Frameworks

- Introduction to popular frameworks like **LangChain**, **LlamaIndex**, and **Haystack**.
- The core components of these frameworks: chains, agents, tools, and memory.
- Building a basic, multi-step application with an orchestration framework.
- **Hands-on Project:** Create a simple tool that uses a chain to perform a task (e.g., summarizing an article and then generating a social media post).

Week 4: LLM-Powered APIs & Full-Stack Apps

- Designing a RESTful API with **FastAPI** to serve your LLM application.
- Building a simple front-end with **Streamlit** or **Gradio** to interact with the API.
- **Hands-on Project:** Transform your previous LLM chain into a functional web application with a user interface.

Part 2: Data, Integration, and Optimization (Weeks 5-8)

This section focuses on the engineering challenges of data integration, performance, and cost management.

Week 5: Retrieval-Augmented Generation (RAG)

- The RAG paradigm: the key to grounding LLMs with private data.
- Understanding vector embeddings, vector databases, and similarity search.
- Building a complete RAG pipeline from data ingestion to query response.
- **Hands-on Project:** Build a chatbot that can answer questions based on a set of documents, using a vector database for retrieval.

Week 6: Cost, Performance, and Token Management

- Strategies for cost optimization: managing API calls, caching, and model selection.
- Techniques for improving latency and throughput.
- **Prompt compression** and context window management.
- **Hands-on Lab:** Analyze the cost of your RAG application and implement caching to reduce API calls.

Week 7: Building Multi-Agent Systems

- Introduction to multi-agent collaboration with frameworks like **CrewAI** or **AutoGen**.
- Designing agent roles, tasks, and communication protocols.
- Building a team of agents to perform a complex, collaborative task.
- **Hands-on Project:** Create a crew of agents to perform a multi-step task, like generating a market research report.

Week 8: Advanced Data Integration

- Integrating LLMs with external tools and APIs (e.g., search engines, calendars, email clients).
- The concept of a "tool-calling" LLM and its importance in building agents.
- Building custom tools for your specific application needs.
- **Hands-on Project:** Add a new tool to your multi-agent system, allowing it to interact with a third-party API.

Part 3: MLOps, Security, and Professional Skills (Weeks 9-12)

This final section covers the essential engineering skills for deploying, monitoring, and securing LLM applications in a production environment.

Week 9: LLMOps Fundamentals

- The LLMOps lifecycle: from development to monitoring and maintenance.
- **Containerization with Docker** for consistent and reproducible deployments.
- Building CI/CD pipelines for your LLM application.
- **Hands-on Lab:** Dockerize your entire web application for easy deployment.

Week 10: Cloud Deployment & Scaling

- Deploying LLM applications on a major cloud platform (AWS, GCP, Azure).
- Using managed services for model serving and inference.
- Strategies for scaling your application to handle high traffic and user demand.
- **Hands-on Project:** Deploy your Dockerized application to a cloud service and set up auto-scaling.

Week 11: Security & Responsible AI

- **Prompt Injection Attacks:** Understanding the threat and implementing defenses.
- Securing API keys and sensitive data.
- Implementing content moderation and guardrails to filter harmful outputs.
- **Hands-on Lab:** Implement security measures to prevent prompt injection and filter toxic content.

Week 12: Final Capstone Project & Career Skills

- **Capstone Project:** Design, build, and deploy a complete, professional-grade LLM application. This project should solve a real-world problem and demonstrate mastery of all the skills learned.
- Building a professional portfolio and resume tailored for LLM Application Engineer roles.
- Interview preparation and understanding the current industry landscape.

techupgrad