APPLIED AND GENERATIVE AI

Building Generative AI Applications

Level: Practitioner • 2 days (expandable to 3) • Virtual, In-person

Overview

Calling a large language model is easy. Building an application on top of one that behaves reliably is the real skill, and it is what this course teaches. You will go from your first API call to a working, dependable feature that produces structured results, uses your code through tool calling, and can draw on your own data.

This is a hands-on, practitioner course. It builds in the order that keeps you on solid ground. We begin with what an LLM API actually is and get a first call working, then learn to prompt from inside an application rather than in a chat window. From there each step adds a capability that the last one makes possible: structured output so your code can trust the result, tool calling so the model can act, grounding so it can use your data, and finally the hardening that turns a prototype into something you would ship. Rather than tour every feature of every SDK, we build one real feature end to end and go deep on the handful of techniques that carry most applications. Every module ends with hands-on work.

Who Should Attend

- Developers building their first real features on top of large language models
- Engineers integrating LLMs into existing applications
- Technical staff who want a dependable, hands-on foundation before advanced or agentic work

Prerequisites

- Working proficiency in a programming language such as Python or JavaScript
- Comfortable calling web APIs and working with JSON
- *Introduction to Generative AI*, or equivalent familiarity, is helpful but not required

What You Will Learn

- Explain what an LLM API is: models, messages, tokens, and cost
- Prompt effectively from within application code, not just a chat window
- Get structured, validated output your code can act on
- Use tool and function calling to let the model call your code
- Ground the model in your own data with a simple retrieval flow
- Harden a feature with streaming, error handling, and a first look at evaluation

Course Outline

Day one: talking to the model

- How LLM Applications Work
  - What an LLM API actually is: models, messages, tokens, and cost
  - The request and response loop; system and user messages
  - Choosing a model for the job at hand

  - Lab: make your first calls to the OpenAI and Anthropic APIs
- Prompting Inside an Application
  - Designing prompts you reuse in code, not one-off chat messages
  - System prompts, examples, and instructions that hold up
  - The parameters that matter, such as temperature
  - Lab: build a small, reliable prompt-driven feature
- Getting Structured, Usable Output
  - Why free-form text is hard to build on
  - Asking for JSON and validating what comes back
  - Handling the model when it does not comply
  - Lab: return structured data your code can act on

Day two: doing more than text

- Tool and Function Calling
  - Letting the model call your code
  - Defining tools and handling the call-and-respond loop
  - Lab: give your application a tool and let the model use it
- Grounding the Model in Your Own Data
  - Why models need your data, and the retrieval idea in brief
  - A simple retrieve-then-answer flow (covered in depth in *Retrieval-Augmented Generation (RAG) with Vector Databases*)
  - Lab: answer questions over a small set of your own documents
- From Prototype to Dependable Feature
  - Streaming responses and the effect on user experience
  - Handling errors, rate limits, and cost
  - Knowing when output is good enough: a first look at evaluation
  - Lab: harden your feature and stream its output

Extended Version

The three-day version keeps the same gradient and adds a bridge toward more advanced work:

- Multi-step chains and a first simple agent loop
- Working with more than one provider or model behind a single interface
- A fuller look at evaluation and guardrails
- A capstone that takes one feature from first call to a robust, streamed, tool-using, data-grounded implementation