# Preface

## Why I wrote this book

This book came about because I needed a way to copy my knowledge to my colleagues.

I’ve been using Large Language Models (LLMs) for quite a while now and discovered how to use them in building applications by failing a lot. I don’t think I can protect anyone from making their own mistakes, but I can show what I learned and explain what worked for me.

## Who this book is for

I wrote this book for developers who want to use an LLM in their application to solve specific issues that can’t be solved with normal program logic. All examples in the book are in C# because I think you can find plenty of Python examples online but not enough in a more enterprise-oriented language like C#.

## How to get the most from this book

As with many things in programming, you must build it yourself to fully understand what you’re working with. You can gain the most from this book by running the examples and building your pet project with the patterns in your back pocket.

The examples are on Github: <https://github.com/wmeints/effective-llm-applications>. I’ve provided enough instructions to get the samples up and running in a few minutes.

Make sure you are familiar with the C# programming language and ASP.NET Core. If you’re unfamiliar with the concepts in either one, I highly recommend reading about them before diving into this book.

Although I will explain what an LLM is, I’m not going too deep into this topic. I believe there are other books available that can help you learn more about its intrinsics.

## What this book covers

Chapter 1, “Understanding Large Language Models,” introduces large language models (LLMs) through the lens of my personal experience, explaining what they are, their capabilities, and why they're transforming how we build applications. This foundational chapter is essential for understanding the core concepts, terminology, and practical considerations that will help you make informed decisions about using LLMs in your projects, even if you're completely new to working with these models.

Chapter 2, “Getting Started with Semantic Kernel,” covers Semantic Kernel, a powerful framework that simplifies building applications with LLMs by providing robust tools, abstractions, and patterns for integrating AI capabilities into your projects. This chapter walks you through setting up your development environment and creating your first Semantic Kernel project, giving you hands-on experience with the framework we'll build upon throughout the rest of the book.

Chapter 3, “The Art of Prompt Engineering”, dives into the art of prompt engineering, teaching you how to craft effective prompts that get reliable, high-quality responses from LLMs by covering key concepts like temperature, templates, and advanced techniques. This chapter is crucial because the ability to write good prompts is the foundation of working with LLMs - without this skill, you'll struggle to get consistent results no matter what frameworks or patterns you use.

Chapter 4, “Enhancing LLMs With Tools”, explores how to enhance LLMs by giving them access to external tools and skills, showing you how to build custom tools, integrate APIs, and manage memory and context to create more capable AI systems. LLMs become dramatically more powerful when they can interact with external tools and data - understanding these patterns will let you build AI assistants that can take actual actions and work with real-world data.

Chapter 5, “Retrieval Augmented Generation (RAG)”, shows you how to supercharge your LLM applications by grounding them in your own data using Retrieval Augmented Generation (RAG), going from basic vector embeddings all the way to building a working domain-specific chatbot. If you want your LLM applications to give accurate responses based on your company's documents, internal knowledge, or any specific dataset, this chapter is essential since it teaches you the complete RAG architecture from preprocessing to efficient retrieval and context integration.

Chapter 6, “Working with structured output”, teaches you how to get structured output from LLMs, enabling you to reliably integrate AI responses into your applications through systematic prompting techniques and robust error handling. This chapter helps you bridge the gap between AI capabilities and your existing codebase. Mastering this skill will let you build reliable AI features that work seamlessly with your applications.

Chapter 7, “Prompt Chaining Workflows”, delves into the essential pattern of prompt chaining, showing you how to break down complex tasks into manageable sequences of prompts that build upon each other's outputs to refine the rough version of a result into a more refined piece of information.

Chapter 8, “Intelligent Request Routing Workflows”, teaches you how to build systems that intelligently route requests between different LLM agents and endpoints based on their content. We’ll cover what an agent design pattern is, and how to let an agent route request to get more effective answers from the LLM.

In Chapter 9, “LLM Orchestration Workflows”, you'll learn how to design and implement robust LLM orchestration workflows that can handle complex document processing tasks at scale, including key patterns for state handling, and error recovery. This chapter is essential for anyone building production-grade LLM applications that need to reliably process large volumes of requests while maintaining observability and performance - skills that become critical as your LLM applications grow beyond simple single-request implementations.

Chapter 10, “Artist and Critic Workflows”, explores the powerful "artist and critic" pattern where LLMs are used both to generate content and to critically evaluate and refine that content through iterative feedback loops. This chapter is crucial for developers who want to build self-improving systems that can generate high-quality content autonomously, making it especially valuable for applications in content creation, code generation, or any scenario where output quality is important.

Chapter 11, “Building Basic Agents,” introduces the fundamentals of building autonomous agents with LLMs. It covers the core loop of observation, reasoning, and action along with essential patterns for managing agent memory and state. This chapter is essential for developers looking to create AI assistants that can perform complex tasks independently, providing the foundation you'll need before diving into multi-agent systems in the following chapter.

Chapter 12, “Building Multi-Agent Teams”, tackles the advanced topic of building multi-agent teams, exploring how to coordinate multiple AI agents working together with defined roles, communication protocols, and strategies for handling resource conflicts and team dynamics. Although the multi-agent is still in its infancy, reading this chapter will help you understand where current generative AI research is moving towards.

## System requirements