

The Future Belongs to *AI Builders*.

Not the ones who use AI. The ones who build with it.

Agentic Sprint for .NET Developers

Zero to One for Developers



Master Generative AI, Agentic AI & Intelligent Automation

6 Weeks | 18 Live Sessions | Hands-On Assignments | Mentorship | Community

What Is **SIDILLUS** **AGENTIC SPRINT?**

A 6 week, 18 session sprint that gives .NET developers a clear, hands-on understanding of building with LLMs and AI agents. The essential concepts and a working feel for the Microsoft AI stack, at a pace built for understanding, not rushing.

Real C# in every session, kept deliberately light, local models, retrieval, tool calling, MCP, agents, and a first look at evaluation, observability, and architecture. So you walk into deeper programs ready, not overwhelmed before you start.

Program Overview

A clear, unrushed on-ramp to agentic AI

One unified curriculum, 6 milestones, 18 sessions, across 6 weeks, built to give working .NET developers a solid, hands-on understanding of the Microsoft AI stack, without the depth or pace of a full production engineering program.

Milestone	Theme	Length	Roadmap & Outcome
1	Foundations for .NET	3 Sessions	Understand what LLMs actually do, how to prompt them well, and make your first real model call from C# using Microsoft's unified AI abstraction.
2	Local Models & RAG	3 Sessions	Run a model on your own machine, then get a first working feel for retrieval grounding answers in your own documents, kept simple on purpose.
3	Tools & MCP	3 Sessions	Let a model call a C# function, then understand the Model Context Protocol before using one from your own app, no need to build a server yet.
4	Building Agents	3 Sessions	See where agent frameworks in .NET came from with a look at Semantic Kernel, then build your first real agent with the Microsoft Agent Framework.
5	Quality & Production Awareness	3 Sessions	A first, honest look at evaluation, observability, and packaging, enough to know these matter and roughly how they work, not a deep production build.
6	Capstone & Career	3 Sessions	Pick something small and real, build it over 3 focused days, and present it, then map what a deeper program would add on top.

How This Program Is Built

- **Real C# in every session, kept light on purpose**, one clear idea plus a small, achievable build, not a production project.
- **A working AI-connected API by Session 3**, a simple RAG assistant by Session 6, and your first agent by Session 10, each one small enough to finish without rushing
- **Every session ends with a small deliverable**, not a lecture by Session 18 you'll have touched the full Microsoft AI stack once and understood why each piece exists.
- **This is a foundation, not a finish line**: it's built to prepare you for deeper, more advanced agentic AI programs, not replace them.



PHASE 1

LLM & Prompting Foundations

Milestone 1: Foundations for .NET

Sessions 1 to 3

MILESTONE ROADMAP

Understand what LLMs actually do, how to prompt them well, and make your first real model call from C# using Microsoft's unified AI abstraction.

Session 1

LLM Fundamentals for .NET Developers

- What a large language model actually does · tokens and context windows in plain terms
- Chat vs completion vs embedding models · the handful of providers you'll actually use
- Where LLMs fit into a .NET app, just another HTTP API call, nothing magical

ASSIGNMENT:

Call one LLM provider's API directly from a .NET console app using `HttpClient`, print the response, and note what a request/response actually looks like

Session 2

Prompt Engineering & Getting Useful Output

- Writing a clear prompt: system, user, and a bit of structure
- Iterating on a prompt until the output is actually usable
- Asking for JSON back and reading it into a simple C# object

ASSIGNMENT:

Write one prompt for a real task you do at work, run it 3 times with small tweaks, and get one version to return clean JSON you can deserialize.

Session 3

Microsoft.Extensions.AI: One Interface, Any Provider

- Why Microsoft.Extensions.AI exists - the IChatClient interface
- Swapping providers (OpenAI, Azure OpenAI, Anthropic) without rewriting your code
- Wiring an AI client into a minimal ASP.NET Core API with DI

ASSIGNMENT:

Build a one-endpoint Minimal API that answers a question via IChatClient . Swap the provider once through configuration and confirm it still works.

Milestone 2: Local Models & RAG

Sessions 4 to 6

MILESTONE ROADMAP

Run a model on your own machine, then get a first working feel for retrieval grounding answers in your own documents, kept simple on purpose.

Session 4

Running Models Locally with Ollama

- Why you'd run a model locally: cost, privacy, offline use
- Installing Ollama and pulling a small open model
- Pointing your Session 3 API at Ollama instead of a cloud provider

ASSIGNMENT:

Get a local model running in Ollama and answering through your existing API. Compare one answer side-by-side with the cloud version.

Session 5

Vector Search & Microsoft.Extensions.VectorData Basics

- What an embedding is and why similarity search works, no math, just intuition
- Microsoft.Extensions.VectorData and pgvector, at a glance
- Storing a handful of records and searching them from C#

ASSIGNMENT:

Store 8–10 short pieces of text in a pgvector-backed store and run 3 searches against them. Just get it working, polish isn't the point yet.

Session 6

Your First RAG Assistant

- The RAG idea in one sentence: look it up, then answer with what you found
- A simple ingest → retrieve → answer flow in .NET
- Where RAG answers go wrong, and why that's normal at this stage

ASSIGNMENT:

Wire your Session 5 store into a simple "ask a question, get a grounded answer" endpoint. Try 5 questions and just observe what happens.

PHASE 2

Tools, MCP & Agents

Milestone 3: Tools & MCP

Sessions 7 to 9

MILESTONE ROADMAP

Let a model call a C# function, then understand the Model Context Protocol before using one from your own app, no need to build a server yet.

Session 7

Function & Tool Calling Basics

- What it means for a model to "call a function" the first real agent behaviour
- Exposing one or two C# methods as tools with Microsoft.Extensions.AI
- Watching the model decide when, and when not, to use a tool

ASSIGNMENT:

Give your chat client one simple C# tool (a stub is fine) and get the model to call it correctly on 3 different prompts

Session 8

Understanding the Model Context Protocol (MCP)

- What MCP is and the problem it solves, one standard instead of custom integrations everywhere
- Server, client, tools, and resources, the four ideas that make up MCP
- A look at a working MCP server, using an existing one rather than building your own yet

ASSIGNMENT:

Explore one existing MCP server with an inspector tool. Write down what tools and resources it exposes, and what each one does.

Session 9

Using MCP Tools from Your .NET App

- Connecting to an MCP server as a client from C#
- Letting the model discover and call MCP tools automatically
- What a connected MCP server can and can't do, the trust boundary

ASSIGNMENT:

Connect your Session 3 API to the MCP server from Session 8 as a client, and get the model to successfully use one of its tools.

Milestone 4: Building Agents

Sessions 10 to 12

MILESTONE ROADMAP

See where agent frameworks in .NET came from with a look at Semantic Kernel, then build your first real agent with the Microsoft Agent Framework.

Session 10

Agent Frameworks in .NET: From Semantic Kernel to MAF

- What makes something an "agent" rather than just a chat client with tools
- Semantic Kernel in a nutshell, what it was for, and why Microsoft built a successor
- Your first agent with the Microsoft Agent Framework: AI Agent, tools, and a simple task end-to-end

ASSIGNMENT:

Build one simple agent using the Microsoft Agent Framework that completes a small multi-step task, e.g. "look up an order and draft a reply." That's the whole assignment, get it working, not perfect

Session 11

Multiple Agents Working Together

- Why you'd split work across more than one agent
- A simple two-agent pattern: one does the work, one reviews it
- Passing information cleanly from one agent to the next

ASSIGNMENT:

Build a 2-agent workflow (e.g. Writer → Reviewer) for one small task, and run it end-to-end once.

Session 12

Keeping a Human in the Loop

- Why some agent actions need a person to approve them first
- Adding a simple approval step before anything real happens
- Remembering context across a conversation, at a basic level

ASSIGNMENT:

Add one approval checkpoint to your Session 11 workflow before its final action, and test both approving and rejecting it.

PHASE 3

Quality & Production Awareness

Milestone 5: Quality & Production Awareness

Sessions 13 to 15

MILESTONE ROADMAP

A first, honest look at evaluation, observability, and packaging, enough to know these matter and roughly how they work, not a deep production build.

Session 13

Understanding the Model Context Protocol (MCP)

- Why "it looked right once" isn't evaluation
- Writing a small set of test questions with expected answers
- Running your RAG assistant or agent against that set with `Microsoft.Extensions.AI.Evaluation` and scoring it

ASSIGNMENT:

Write 8–10 test cases for your RAG assistant or agent, run them, and mark each pass or fail.

Session 14

A First Look at Observability

- Why you'll want to see what your AI app is actually doing, not just its final answer
- OpenTelemetry, in plain terms, traces, not just logs
- Turning on tracing for one AI call and looking at what it captured

ASSIGNMENT:

Turn on OpenTelemetry tracing for one endpoint, run it 3 times, and look at what shows up in the trace.

Session 15

Packaging Your App: An Introduction to .NET Aspire

- Why real apps are more than one project, an API, a database, maybe a local model
- What .NET Aspire does for you, at a glance
- Running your app's pieces together with Aspire's App Host

ASSIGNMENT:

Add your Session 3 API and vector store to a basic Aspire AppHost, and run the whole thing with one command.

Milestone 6: Capstone & Career

Sessions 16 to 18

MILESTONE ROADMAP

Pick something small and real, build it over 3 focused days, and present it, then map what a deeper program would add on top.

Session 16

Capstone: Pick Something Small and Real

- Choosing a capstone idea that's genuinely finishable in 3 days
- A one-page plan: what it does, what tools it uses, how you'll know it works

ASSIGNMENT:

Write a one-page plan for your capstone agent, keep the scope small on purpose.

Session 17

Capstone: Build It

- Building your plan end-to-end
- Testing it against a few real examples, not edge cases

ASSIGNMENT:

Get your capstone working end-to-end on 3–5 real examples.

Session 18

Capstone: Show It & What's Next

- Demoing what you built and why you made the choices you did
- Where to go next, what a deeper program would add on top of this

ASSIGNMENT:

Give a short demo (5–10 minutes) and write down 3 things you'd want to learn next.

Why Upskill With Sidillus



1:1 Mentorship

Get direct guidance and feedback while you build.



Live Classes

Learn concepts in real time and get your doubts answered immediately.



Hands-On Learning

Every session includes a practical assignment or deliverable.



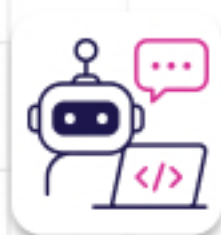
Real-World Scenarios

Apply AI concepts to business, development and automation use cases.



Community

Learn alongside others exploring AI and Agentic development.



Practical AI Skills

Go beyond prompting and learn how to design, build, test and evaluate AI systems.

Program Outcomes

By the end of Agentic Sprint, you will be able to:

- Understand how modern Generative AI systems work.
- Use LLMs more effectively through structured prompting.
- Build RAG-based knowledge assistants.
- Build your first agent with the Microsoft Agent Framework.
- Extend agents using MCP and Agent Skills.
- Write specifications before building AI systems.
- Build multi-agent workflows.
- Evaluate agent performance using structured test sets.
- Package and run a full AI-connected app with .NET Aspire
- Identify security and reliability.
- Build and present a complete working AI agent solution.



Meet the Founders

Built by People Who've Been There

Adil A Latheef Founder, Senior Software Engineer

Adil brings 8+ years of enterprise software engineering experience, with expertise in .NET and React. He secured offers from leading MNCs including Wipro, Cognizant, Infosys, Mindtree, HCL and HTC, beginning his career at Mphasis. Throughout his career, he has delivered enterprise solutions for global clients including HP and P&G, bringing production-grade engineering expertise and real-world industry experience to Sidillus.

He founded Sidillus to bring that same standard of practical, production-grade thinking into AI training for working developers.



Sibin Fawaz Co-Founder, Senior Tech Lead

Sibin is a Senior Tech Lead with extensive experience in building enterprise-scale mobile applications using React Native. He has secured opportunities with leading technology companies including IBM, Persistent Technologies, Fujitsu, and Virtusa. Throughout his career, he has delivered high-impact solutions for organizations such as Sharjah Public, Sharjah Police, and AstraZeneca, bringing deep expertise in scalable application development, enterprise architecture, and technical leadership to Sidillus.

He co-founded Sidillus to help software engineers build real AI capabilities and thrive in the next generation of software development.



The Window Is Open.

It Won't Stay That Way Forever

AI is moving fast. The people who learn to build with it, not just use it are going to have a different kind of advantage.

This program won't promise you a job. It will give you the skills, the projects and the clarity to go after one or build something of your own.

Join

Sidillus Agentic Sprint

Zero to One for Developers



Start your journey into
Generative AI, Agentic AI
and Intelligent Automation.

Learn. Build. Innovate.

Connect with us

 connect@sidillus.com

 +91 8086 808 303

 sidillus.com



Scan Here! Join Now!