The AI Agents Book

An Yepas Production ®

Index:

1. Introduction

What You Are Really Building

At some point, you probably had a moment like this.

You asked an AI to do something that felt complex. Maybe it was to debug a piece of code, design a system, or explain a difficult concept. And for a moment, it worked. It gave you an answer that felt surprisingly good. Structured, clear, even insightful.

And then, a few steps later, something broke.

It misunderstood something simple. It forgot a detail you just mentioned. It gave an answer that sounded correct but clearly was not. Or it simply lost track of what it was doing.

That moment is confusing, because the system seems capable, and yet unreliable. It feels intelligent, and yet inconsistent.

It is tempting to explain this as a limitation of the model, as if the technology is simply not advanced enough yet. But that explanation is incomplete, the deeper reason is this:

What you are interacting with is not a complete system, it is a powerful component inside a system that does not yet exist.

Most introductions to artificial intelligence begin by explaining models. They describe how neural networks are trained, how data is processed, and how performance is measured. This is useful, but it does not explain the experience you just had.

Because the problem is not only how good the model is.

It is what the model is missing.

A language model, taken on its own, is a remarkable but limited thing. It can transform text into text with surprising fluency. It can simulate reasoning, explain ideas, and generate code. But it does not persist knowledge across time. It does not act in the world. It does not verify whether what it produced is correct. It does not organize work into a coherent sequence of steps.

And yet, those are exactly the things you expect from something that feels intelligent.

This gap between what the model can do and what you expect it to do is where most confusion comes from.

The central idea of this book is that this gap is not a flaw.

It is the starting point.

Once you understand that the model is only one part of the system, a different way of thinking becomes possible. Instead of asking how to make the model smarter, you begin to ask how to build a system around it that compensates for its limitations.

If the model cannot remember, you design memory.

If it cannot act, you give it tools.

If it cannot verify, you create feedback loops.

If it cannot structure work, you introduce planning.

What emerges from this is not just a better prompt or a better API call.

It is a different kind of system entirely.

This is what you are going to build. Not a chatbot, not a wrapper around an API(explain with a short phrase what API in Leaman terms), but a system that operates over time, that interacts with its environment, that makes decisions under uncertainty, and that remains useful even when it is not perfectly correct.

Each chapter in this book exists for a reason. It is not there to introduce a topic, but to solve a problem that arises from the limitations of the previous layer. You will see how each piece connects to the next, and why it is needed.

This way of learning is intentional.

Because the field you are entering is changing quickly. Frameworks will come and go. Libraries will evolve. Interfaces will be replaced. But the underlying structure of these systems, the reasons they work and the reasons they fail, changes much more slowly.

If you understand that structure, you are not dependent on any specific tool.

You can build, adapt, and improve.

You do not need a background in machine learning to follow this path.

What you need is something else.

You need to be willing to look at systems not just when they work, but when they break. Because those moments, the ones that feel confusing or inconsistent, are the ones that reveal what is really happening.

By the end of this book, you should be able to look at any modern AI system and see beyond its surface. You should be able to understand what parts of the system are responsible for its behavior, where its limitations come from, and how you might design something similar yourself.

That is the goal.

Not just to use these systems, but to understand them deeply enough to build them.