

GenAI engineering is more than aggressive prompt tweaking.

You already build with AI. You can get a prototype working in an afternoon. Then real users arrive and it breaks. The data is messier than your demo. The model says something it should not say. Nobody notices for a week.

This course teaches everything that comes after that moment. You pick one project you care about. You take it from an empty folder to a real product, live on your own domain. A pipeline checks every change you make. That takes 17 modules. Each one gives you a lesson and a lab, and every lab moves your own project forward.

We build a web app together. The same method works for an agent, an API, or a RAG service.

83K+

GitHub stars

40K+

Developers subscribed

2

Books on RAG & prompting

17

Deep-dive modules

Materials are the cheap part

Any AI model can write you a good outline of what production-ready means. You would read it, agree with all of it, and still ship the same fragile app. Reading is not what changes the way you build. Correction is. Three parts of this course correct you, and none of them fit inside a PDF.

Cloach reads the prompts you actually write

Cloach is a coach that sits in your terminal while you work through a lab. It knows which exercise you're on. It reads the prompts you send to Claude Code. When a prompt gives you a bad result, Cloach tells you why *your* prompt failed. A video can show you a good prompt. Only Cloach can look at yours.

People who answer when you're stuck

One evening you'll hit a problem no lesson covered. That happens to all of us. You post it in the members-only Discord, and someone answers the same day. You don't lose a week waiting.

You finish with a product, not notes

At the end you have an application other people can visit and use. It runs in a container. It has tests, auth, and a security review. And it's yours, so you keep building on it after the course ends.

What you walk away with

A real product in production

No more toy dashboard clones. You design, secure, and deploy a live application that holds up when real users and strange edge cases arrive.

A method that outlives the tools

You stop rewriting your workflow every time a new model ships. The system design and the method stay the same underneath.

Prompts your agent follows

You write the system prompts, context files, and test gates that a coding agent has to obey. What comes back is code you would actually merge.

Say hello to Cloach

We didn't record a set of videos and wish you luck. We built you software.

Cloach runs in your terminal, next to your project. It knows the module, the lab, and the exercise you're on. It reads your prompts and shows you what is weak in them. When you get stuck, it nudges you, and it does not hand you the answer. It also keeps score, so you can watch your prompting get better week by week.

We built Cloach with the same method this course teaches. The assistant helping you learn is itself an example of the thing you're learning to build.

```
|> cloach
M6 · E2 · 12 prompts · 75% pass

you > add auth to the app

cloach > That's going to
guess. It doesn't know
your session model, or
that you already have a
users table in schema.sql.

Name the file. Name the
contract. Name the failure
you want handled.
Try again. I'll score it.
```

What's included

Lessons and hands-on labs

Every module gives you a short lesson and a lab. In the lab you build the next piece of your own project. You never build a throwaway demo.

Cloach, your terminal coach

Cloach walks each lab with you, reviews your prompts as you write them, and tracks your progress (see above).

A private Discord

A members-only space where you can share what you're building and get unstuck the same day.

Live Q&A webinars

When a question keeps coming up in Discord, we run a live session on it. The course keeps answering what people really hit.

A 30-day refund

If the course is not worth your time and your money, you get all of it back within 30 days. You only have to ask.

Founding-member pricing

Everyone on the waiting list gets the launch price, the lowest it will ever be. Twenty people join the alpha for free.

Who's teaching this

Nir Diamant

Nir writes open-source AI guides for developers. They have 83K+ GitHub stars, and 40,000 AI engineers read his newsletter. He wrote two books, one on RAG and one on prompting.

Roi Halamish

Roi owns several software companies. He puts AI development techniques to work in production, with real teams, on real deadlines.

1

PART 1

Setup & Foundations

From zero to a project that boots: environment, AI tools, your project chosen and scoped, a scaffold that runs.

1 • Welcome to the AI-Native World

What it means to build with AI as a professional, how the course runs, and your first session with Cloach.

2 • Your Dev Environment

Install and verify everything, then write your first CLAUDE.md, the briefing document the AI reads on every session.

3 • Choosing Your Project

Pick a real project you'll still want to open at module 17. Scope it to a first version you can actually finish.

4 • The Spec Layer & a Working Scaffold

Write the documents the AI reads every session, then turn them into a project that boots.

5 • Spec-Driven Development

Make the spec the source of truth. You define what the app must do as a precise, testable contract. The AI builds against it, and the two stay in sync as your project grows.

2

PART 2

Building the Project (the Vibe Phase)

Ship features fast by driving the AI well. Then find the exact point where pure vibe coding stops working.

6 • Prompt Engineering for Developers

Get the code you want out of the model, then vibe the essentials of your app onto the scaffold.

7 • Context Engineering

Control everything the model sees, so it stops guessing. You'll ship your second feature with far less effort than the first.

8a • Claude Code Setup

Make the tool fit your project: settings, a grown CLAUDE.md, and your own reusable commands.

8b • Hooks

Put your rules in the runtime around the model, so they hold whether or not it decides to cooperate.

9 • Minimum Friction

Wire the AI to your files and to the tools you already use. You stop leaving the terminal to get work done.

10 • Primitives: MCPs, Sub-agents, and Skills

The three building blocks the AI uses to act, delegate, and remember. You learn how to pick the right one for a job.

3

PART 3

Research, Plan, Execute

The methodology that takes over when vibing stops scaling, plus the daily rhythm to run it.

11 • Research, Plan, Execute

Understand the problem first. Plan against a real acceptance test. Then build in ordered chunks and verify each one before you move on. You'll still use this method years after the course.

12 • Working with AI: Sessions, Agents, and the Daily Rhythm

How to spend a working day across sessions and parallel agents. We build the rhythm around the way model focus and memory really behave.

4

PART 4

Hardening the Build

Make it survive: data, tests, code quality, and security, each applied with the method from Part 3.

13a · Database & Data Flow

Design your schema with AI assist and map how data moves, before a wrong assumption costs you a rebuild.

13b · Testing: Your Safety Net

Integration tests over real user flows, with the test gate wired into your workflow so running them isn't optional.

13c · Code Quality & Review

Linting, formatting, AI-assisted code review, and paying down the technical debt the vibe phase left behind.

13d · Security

Auth, secrets, and input validation against a real contract. Then you run a full security audit of your own project. That includes the new failure modes that appear once an AI can act on your system.

5

PART 5

Ship It

From your laptop to live with users: deploy, debug, validate, launch.

14 · Deploy to Production

Containerize with Docker and set up CI/CD. Deploy to the cloud behind a real domain, with a green pipeline that gates every change.

15a · Testing & Debugging

Debug real production issues with AI, and turn every bug you fix into a permanent regression test.

15b · Validation & Security in Production

Catch the quiet decisions the AI made for you. Check its output before a user ever sees it. Then defend the live project once real attackers arrive.

16 · Bootstrap Into Any Field

Ship a feature in a field that is new to you. Web for a data person, RAG for a web developer. You get good enough, fast enough, to judge what the AI gives back.

17 · Launch Day

Make your project reachable to the people, agents, and crawlers that find it from the outside. Then ship it.

Start with the first module, free

Join the waiting list to lock founding-member pricing and enter for one of twenty free alpha seats.

diamant-ai.com/courses