



AI Solutions Engineer Specialization

Program Curriculum

12 weeks · live cohort · product-driven · from scratch (no prior experience required)

Program overview

A 12-week, live, product-driven program that takes a complete beginner to a hireable, business-facing AI engineer. Learners build and ship a real AI product every week, ending with a flagship capstone and a public portfolio of deployed work.

Outcome scope: junior-to-mid AI Solutions Engineer / freelance AI builder. A graduate can turn a business problem into a scoped AI solution, build it (LLM apps, RAG, agents), prove it works (evaluation, guardrails), deploy and monitor it in production, and present it to a non-technical stakeholder.

Learner prerequisites: none. Basic computer literacy and strong logical thinking are sufficient. The technical foundation is taught live in Week 1.

How the program is taught: the product-driven method

This is the spine of the program, not a slogan. Every week is framed the same way — Product goal, Skills unlocked just-in-time, and what the learner Ships. The teaching rules:

1. **Build first, theory just-in-time.** No concept is taught for its own sake. Learners meet RAG because their product can't answer from a client's documents yet; they meet evaluation because their product is unreliable. The unfinished product creates the need; the lesson fills it.
2. **Every week ships a working increment.** By Friday the product does something it couldn't on Monday. There are no theory-only weeks.
3. **One flagship product grows across the program.** From Week 5 onward, a single product gains capabilities, gets hardened, gets deployed, and becomes the capstone — so learners live the real arc of solutions engineering.
4. **The user/business is always in the room.** Every build starts from who it's for and what problem it solves.
5. **Proof over certificates.** Every increment is public on GitHub; the portfolio builds continuously and is the credential.

Structure at a glance

Phase	Weeks	The product arc
Foundations	1	Get build-ready; ship your first program
Build with AI	2–4	Ship small AI products; end with your first deployed, live app

Edversity.

The Solutions Core	5–7	Start the flagship product; give it memory (RAG) and the ability to act (agents)
Ship It	8–10	Take the flagship from “works on my machine” to production: evaluate, harden, deploy, monitor
Capstone & Career	11–12	Make it client-grade against a real brief; portfolio, demo day, placement

Each week: roughly 2 hours of theory, 6–8 hours of building, and one graded shipped increment on GitHub.

Week-by-week curriculum

Phase: Foundations

Week 1 — Get build-ready

Product goal: Ship your first working program to a public GitHub repo — prove you can build and publish.

Skills unlocked (taught just-in-time): Command line and VS Code; Git/GitHub (the portfolio home from day one); Python fundamentals (variables, loops, conditionals, functions, lists/dicts, files, error handling, virtual environments, pip); how the web works (HTTP, APIs, JSON); and a no-math conceptual grip on how LLMs actually work (tokens, context windows, embeddings, why they hallucinate, what they are good and bad at).

Ships as: A small Python program with a clean README, live on the learner's GitHub.

The conceptual grip on how LLMs work is what separates an engineer from a button-pusher.

Phase: Build with AI

Week 2 — Your first AI product

Product goal: A working tool that talks to an LLM and does something useful.

Skills unlocked (taught just-in-time): Calling LLM APIs (Claude/OpenAI) from Python; messages, system prompts, parameters, streaming; handling API keys safely (environment variables, never hardcoded); reading API documentation like an engineer.

Ships as: A command-line AI tool on GitHub.

Week 3 — Making AI reliable enough to be a product

Product goal: A tool that returns structured, dependable output a business could plug into a workflow.

Skills unlocked (taught just-in-time): Prompting as engineering (clear instructions, few-shot examples, role design); getting structured JSON out of an LLM and parsing it in code; prompt patterns for extraction, classification, and summarization; token and cost awareness.



Ships as: A structured-data extraction product (e.g., messy text / invoices / resumes into clean JSON).

Week 4 — Ship it to the world (first deployed product)

Product goal: A live, shareable AI app with a public URL — the first thing a learner can send to family, friends, or a client.

Skills unlocked (taught just-in-time): Building a simple web interface (Streamlit/Gradio); wiring a front end to an LLM back end; deploying to the internet on a free or low-cost host.

Ships as: A deployed AI web app with a live link.

This early win is the single biggest driver of completion — protect it.

Phase: The Solutions Core (the flagship product begins)

Week 5 — Give your product memory (RAG part 1)

Product goal: The flagship product can answer questions from a business's own documents, not just the model's training data.

Skills unlocked (taught just-in-time): The problem RAG solves; embeddings and vector databases (Pinecone/Chroma/FAISS) in concept and practice; the pipeline — chunk, embed, store, retrieve, generate.

Ships as: Flagship v1 — a “chat with your documents” product over a real document set.

Week 6 — Make the memory trustworthy (RAG part 2)

Product goal: The product stops making things up and starts citing its sources — the difference between a demo and something a business trusts.

Skills unlocked (taught just-in-time): Why naive RAG fails (bad chunking, irrelevant retrieval, hallucination); improving retrieval quality; grounding and source citation; assembling clean pipelines with LangChain/LlamaIndex.

Ships as: Flagship v2 — a grounded, source-citing assistant over a realistic knowledge base.

Week 7 — Give your product the ability to act (agents)

Product goal: The product moves from answering to doing — completing multi-step tasks by using tools.

Skills unlocked (taught just-in-time): Agents that reason, plan, and act; tool use / function calling (calling APIs, querying data, triggering actions); multi-step workflows with LangGraph; integrating real business systems via APIs and webhooks, with n8n as glue where it speeds delivery.

Edversity.

Ships as: Flagship v3 — an agent that completes a real end-to-end task (e.g., research, summarize, draft, send) using at least one external tool.

Phase: Ship It (the production layer)

Week 8 — Prove it works (evaluation)

Product goal: Move from “it worked when I tried it” to measured, demonstrable quality a client can't argue with.

Skills unlocked (taught just-in-time): Designing evaluations — test sets, ground truth, pass/fail rubrics; LLM-as-a-judge; measuring faithfulness, relevance, accuracy; catching hallucinations and regressions systematically; tooling such as RAGAS for RAG quality.

Ships as: An evaluation suite that scores the flagship product with real numbers.

Week 9 — Make it bulletproof (guardrails & reliability)

Product goal: The product behaves safely on bad input, edge cases, and adversarial users — production-grade, not demo-grade.

Skills unlocked (taught just-in-time): Input/output validation, guardrails, fallback behavior; prompt-injection and jailbreak awareness with basic adversarial self-testing; retries and graceful degradation; responsible-AI basics (bias, privacy, what not to ship).

Ships as: A hardened flagship plus a documented log of the failure modes the learner found and fixed.

Week 10 — Put it in production (deploy & observe)

Product goal: The product runs reliably in the real world and the learner can see what it's doing after launch.

Skills unlocked (taught just-in-time): Proper deployment (environment management, secrets, intro to Docker, cloud deploy); observability — logging, tracing, monitoring LLM apps (Langfuse/Arize-style); cost monitoring and optimization.

Ships as: A deployed, monitored flagship with logging and a basic activity dashboard.

Phase: Capstone & Career

Week 11 — Capstone: scope it like a real engagement

Product goal: Take a real, client-style brief and turn it into a scoped solution — problem, architecture, success criteria.



Skills unlocked (taught just-in-time): The core solutions-engineering skill — translating a business problem into a system; architecture decisions and trade-offs. Begin the end-to-end build applying the full arc.

Ships as: A scoping document, an architecture, and the first working slice of the capstone.

Week 12 — Capstone: build, evaluate, ship

Product goal: A finished, client-grade AI product — built, evaluated, hardened, deployed, monitored.

Skills unlocked (taught just-in-time): Final build under code review and technical feedback from instructors; professional write-up (README, architecture diagram, evaluation results, demo video).

Ships as: The completed flagship capstone, public and runnable.

Core toolchain (what a graduate can put on a resume)

- Python
- LLM APIs (Claude, OpenAI)
- LangChain / LangGraph
- LlamaIndex
- RAG (retrieval-augmented generation)
- Embeddings & vector databases (Pinecone / Chroma / FAISS)
- Agents & function calling / tool use
- Streamlit / Gradio
- APIs & webhooks; n8n for workflow glue
- Evaluation (RAGAS, LLM-as-judge, eval harnesses)
- Guardrails & adversarial testing
- Docker (intro) & cloud deployment
- Observability (Langfuse / Arize-style logging, tracing, monitoring)
- Git / GitHub

What a graduate walks away with

A public, runnable portfolio:

- A command-line AI tool (Week 2)
- A structured-data extraction product (Week 3)
- A deployed AI web app (Week 4)

Edversity.

- The flagship: a grounded, agentic, evaluated, guarded, deployed, monitored AI product (Weeks 5–10)
- The capstone: a real-world, client-grade solution against a real brief (Weeks 11–12)

Assessment model

- Build-along, not watch-along: every week ships a graded increment to GitHub; the portfolio builds continuously.
- Phase checkpoints: end of Build with AI (deployed app), Solutions Core (acting agent), Ship It (evaluated and monitored production system).
- The capstone is the primary graduation criterion — a real, working, documented product.
- No written exams. Everything is “show me it runs,” because that is how the industry evaluates too.