Case Study: The Agentic Book

Build AI agents you can actually run — a curriculum for people and their coding tools

Julian M. Kleber | Builder | agentic-book.org

julian.kleber@sail.black | Last updated: August 15, 2026

Outcome snapshot

- Published a **17-chapter, five-movement curriculum** — foundations, language models, knowledge, agents, and production — on a public living-edition site.
- Made the book readable by **two audiences at once**: humans in the browser, and coding agents via a structured edition with **stable section ids** and an **MCP** server.
- Shipped the **waitlist** flow (Supabase) plus public source hosting so readers can open and build examples from day one.
- Positioned around the thing tutorials miss: **production** — logs, cost, safety rails, deploys — instead of a clever demo.

CONTEXT

Tutorials show a chat loop and call it an agent. Most guides stop at a clever demo. Real agent products need retrieval, tools, tests, cost control, and a team that speaks the same language. Engineers putting agents into products lack a single path from foundations to shipping, and the tools they build with cannot read most book content at all.

PROBLEM DEFINITION

In the status quo, builders face:

- scattered fragments: blog posts, leaderboards, and repo examples instead of one ordered sequence,
- no shared vocabulary from math-to-models through production operations,
- books that are static — agents and editors cannot query the chapters while you build,
- demos that stop before reliability, cost, and safety matter.

APPROACH

The Agentic Book is a curriculum, not a pile of tabs:

- **Five movements** — I Foundations, II Language models, III Knowledge, IV Agents, V Production — so you always know what comes next.
- **Plain language with definitions** — “an agent is a loop over a model, context, and tools.”
- **Two readers** — a browser edition for people and a structured, searchable edition for coding agents.
- **Agent-native plumbing** — stable section ids, a machine index, and an MCP server so an editor assistant can open the TOC, search, and walk a chapter without scraping a PDF.
- **Living edition** — early chapters ship as they go live; reader margin notes go straight into the backlog.

SYSTEM OUTLINE

- **Web edition**: Next.js site with a flip-book reader, per-page folios, and reading progress.
- **Structured edition**: machine-readable chapter source with stable ids alongside the prose.
- **MCP server**: queryable table of contents and course mode for coding agents.
- **Waitlist**: email capture with account page (Supabase-backed).
- **Fast Math**: KaTeX-rendered formulas where the math movement needs them.

IMPLEMENTATION STACK

- **Frontend**: Next.js App Router, FlipBook-inspired reader UI.
- **Data / auth**: Supabase (Postgres + Auth) for waitlist and accounts.
- **Content**: Markdown chapter source, stable section ids, machine index.
- **Agent access**: MCP server exposing TOC and course-mode navigation.
- **Quality**: reader margin notes funnel into the editorial backlog; deploy via public GitHub hosting.

RESULTS (HONEST LAUNCH STATE)

- **Living edition is live** at agentic-book.org with the full curriculum mapped across 17 chapters.
- **Two-reader architecture shipped**: browser edition for people, structured edition + **MCP** for coding tools.
- **Waitlist and account flow live**; early chapters open as they ship.
- **Public source hosting** so readers can run and build the examples, not just read markup.
- No inflated sales numbers: the honest launch claim is a shipped curriculum, an agent-readable edition, and a running waitlist — not vaulted reader counts.

WHAT THIS CASE STUDY DEMONSTRATES

- Product sense for a **two-audience format**: content engineered for humans and for AI tools as first-class readers.
- Pragmatic **agent-native design**: stable ids, machine index, and MCP over "AI-washed" static prose.
- Full-stack delivery: Next.js, Supabase waitlist/accounts, Markdown pipeline, MCP, and public hosting.
- Honest scoping: ship the curriculum and the structure first; let chapters open as they go live.

WHAT I WOULD IMPROVE NEXT

- Open chapters publicly at a steady cadence and refresh them from reader margin-notes.
- Grow the machine-readable edition (per-chapter embeddings search, agent "course mode" assessments).
- Add exercises and checkpoints with automated grading behind a logged-in learner account.

LINKS

- Site: agentic-book.org
- GitHub: github.com/agentic-book-org/agentic-book