From Idea to Live Product in 3.5 Hours

How one business strategist built an interactive AI setup wizard for non-technical professionals using Claude Code.

6,500+	3.5 hrs	~$67	16
Lines of Code	Build Time	Subscription Cost	Setup Steps

Harrison Painter | LaunchReady.ai | March 2026
launchready.ai/build

The Problem

Non-technical professionals want to use Claude Code. They hear about it, they get excited, they try to set it up. Then they hit a wall.

Download Homebrew. Install Node. Configure a terminal. Set an API key. Every step assumes you already know what you are doing. Most people do not. They get stuck at step two, close the tab, and never come back.

There was no guided path. No one had built an onboarding experience that treated these users like the smart, capable professionals they are -- just ones who have never opened a terminal before.

I decided to build one myself.

The Decision

I could have written a blog post. A PDF guide. A YouTube walkthrough. All of those would have been easier. But I had watched real people try to follow written instructions and fail. The gap was not information. The gap was experience design.

I needed an interactive wizard that held someone's hand through every single step, showed them their progress in real time, and never left them wondering what to do next.

This was not a coding decision. It was a design decision. I was not building for myself. I was building for someone who has never seen a command line.

[Level 5: Captain (Design Thinker)]

> **About the 7 Levels**
>
> The 7 Levels of AI is a proficiency framework developed by LaunchReady.ai. It maps practical AI skill to the human capabilities that drive it. Each level represents a distinct way of working with AI, from basic awareness to full orchestration. Level callouts throughout this case study show where specific decisions and skills fall on the framework.

What I Built

An interactive, 16-step setup wizard that walks non-technical professionals through building a personalized AI assistant with Claude Code. Live at launchready.ai/build.

The application includes:

- An interactive multi-step form with progressive disclosure (6,500+ lines of production code)
- A live preview panel that builds a personalized CLAUDE.md file in real time as the user answers questions
- 16 micro-steps across two phases: brain file builder and tool installation
- Platform toggle (Mac/Windows) with conditional instructions
- Save and resume via localStorage. Close the tab, come back later, pick up where you left off.
- Mobile detection with a graceful boundary. Phase 1 works on mobile. Phase 2 requires desktop.
- Copy-to-clipboard for all terminal commands
- Name suggestion buttons for the AI assistant with encouragement messaging at every step
- Responsive design, accessibility (ARIA, keyboard nav, screen reader support), and SEO (HowTo structured data)
- A Cloudflare Worker serverless backend with Turnstile bot protection, Kit API integration, rate limiting, and CORS protection

I also built a complete book design spec, a memory system overhaul, a Kit email tag, a legal disclaimer, and GA4 analytics events in the same session. But the setup wizard was the main event.

How It Happened

9:00 PM -- Started the session. Brainstormed the book spec for "Human IS the Loop," fixed the memory system, built a session-close skill.

9:45 PM -- Shifted focus to the setup page. Researched UX patterns. Studied 200+ onboarding flow analyses, configurator UX studies, and conversion optimization research. Wrote a detailed build plan.

10:15 PM -- Ran a security audit on my own plan before writing a single line of code. The plan scored 65 out of 100. I identified 21 vulnerabilities. Rewrote the plan with all fixes. New score: 87 out of 100.

[Level 3: Lieutenant (Critical Thinker)]

10:30 PM -- Dispatched three parallel AI agents to build HTML, CSS, and JavaScript simultaneously. Three agents, working in parallel, each with clear instructions and a shared architecture document.

[Level 6: Admiral (Systems Integrator)]

10:45 PM -- All three agents completed. Integration review found 30+ selector mismatches between files. Dispatched two more agents to reconcile CSS and JS against the HTML source of truth.

11:00 PM -- Live testing and iteration for a full hour. Fixed preview panel sizing, scroll behavior, step ordering, the platform toggle, copy text, and encouragement messaging. Every fix came from actually clicking through the wizard and asking: would a non-technical person understand this?

12:00 AM -- Deployed the Cloudflare Worker. Configured Turnstile, Kit integration, and environment variables. Security hardened with rate limiting, Turnstile enforcement, and CORS restrictions.

12:15 AM -- Created the Kit tag, drafted the first drip email, added a legal disclaimer. Pushed to production.

12:30 AM -- Live.

What 3.5 Hours Actually Looks Like

The AI wrote the code. But the code was maybe 30% of what mattered. Here is what I actually spent my time on.

UX research and planning (~30 min). I studied 200+ onboarding flow analyses before writing a single line of the plan. I needed to understand why setup wizards fail, what progressive disclosure actually means in practice, and how to design for someone who has never opened a terminal.

[Level 5: Captain (Design Thinker)]

Security thinking (~15 min). I audited my own plan before building anything. Found 21 issues. Rewrote the entire approach. This is the step most people skip. It is the step that matters most when you are collecting email addresses and running API integrations.

[Level 3: Lieutenant (Critical Thinker)]

Design decisions and iteration (~60 min). Live testing every step. Clicking through the wizard as a non-technical user would. Finding the places where someone would get confused, stuck, or frustrated. Fixing them in real time. This was the longest phase and the most important one.

Content and copywriting (~20 min). Writing 16 steps of encouragement messaging. Every step needed a headline, instructions, and a reason to keep going. The tone had to be warm without being condescending.

Integration and deployment (~15 min). Cloudflare Worker, Kit email integration, Turnstile bot protection, environment variables, CORS configuration. This was the fastest part because the architecture decisions were already made.

Quality judgment throughout. Deciding what "good" looks like for a non-technical user is not something AI can do. I was the quality bar. Every decision about what to simplify, what to cut, and what to explain more came from human judgment about a human audience.

What This Would Cost Traditionally

Approach	Estimated Cost	Timeline
Mid-level freelancer ($75/hr, 60-80 hrs)	$4,500 - $6,000	2-3 weeks
US-based agency	$8,000 - $15,000	3-6 weeks
Harrison + Claude Code	**~$67 (3.5 hrs of subscription time)**	**3.5 hours**

Freelancer estimate based on 2026 rates from Arc.dev and ZipRecruiter. Agency estimate based on pricing from Apexure, Landingi, and Tapflare for custom interactive landing pages with API integrations. These are not inflated numbers. A 16-step interactive wizard with a live preview panel, serverless backend, bot protection, and email integration is not a simple landing page.

The Decisions That Shaped It

Name the assistant first. The very first step asks the user to name their AI assistant. Not configure settings. Not install tools. Name it. This creates emotional ownership before the user invests any effort. It turns "set up a developer tool" into "build your assistant."

Micro-steps for tool installation. In early testing, a real user struggled with a step that said "download all 5 tools." That single observation changed the architecture. Each tool became its own step with its own confirmation button. More steps, less confusion.

Never auto-advance while typing. Users were getting yanked to the next step mid-sentence. The form was advancing automatically after a field was filled. I changed every transition to a manual "Continue" button. Slower, but respectful of the user's pace.

VS Code as home base, not Terminal. Terminal is only for initial setup. The test step and all future use happens inside VS Code. For a non-technical user, VS Code is familiar. Terminal is not. Meet people where they are.

Security before launch. During testing, I found that the email capture gate could be bypassed on a Turnstile failure. I fixed it so Turnstile is mandatory, failures show a retry prompt, and the Worker rate-limits by IP. This took 15 minutes. Skipping it would have been a real vulnerability.

What This Means for You

The speed gets the attention. But the speed is not the point.

The research mattered. The security audit mattered. The hour of live testing and iteration mattered. The decision to name the assistant first mattered. Every one of those was a human call that AI could not have made on its own.

The AI could not have made any of those decisions. It did not know the audience. It did not feel the friction in the UX. It did not understand why a non-technical professional would quit at step three.

I knew those things. And because I knew them, I could direct five AI agents in parallel to build something that would have taken a team of developers weeks.

That is Level 5.

The human IS the loop.

Tools: Claude Code (Opus 4.6), VS Code, GitHub, Cloudflare Workers, Cloudflare Turnstile, Kit, GA4

The 7 Levels of AI

The 7 Levels of AI is a proficiency framework that maps practical AI skill to the human capabilities that drive it. Each level builds on the one before it.

Level 1: Cadet (AI Aware)

You know AI exists and you have tried it. You type requests the way you would type into a search engine. The outputs feel hit-or-miss because they are.

Human skill: Self-awareness. Knowing what you do not know.

Level 2: Ensign (Prompt Engineer)

You give AI clear instructions with context, constraints, and format. Your results are better than most because your inputs are better.

Human skill: Structured thinking. You organize your thoughts before giving them to AI.

Level 3: Lieutenant (Critical Thinker)

You use AI as a thinking partner. You ask follow-up questions, stress-test ideas, and push back on weak answers. Most people quit when AI gives a bad answer. You iterate.

Human skill: Self-management. Frustration tolerance and persistence when AI underperforms.

Level 4: Commander (Context Engineer)

You manage the conversation itself. You know when to start fresh, how to carry forward what matters, and why a clean session with good context beats a long one with a full memory.

Human skill: Systems awareness. You see the conversation as a system with constraints and limits.

Level 5: Captain (Design Thinker)

You design AI experiences for others. You think about what data AI needs, how workflows should be structured, and how to scope access responsibly.

Human skill: Design thinking. You work backward from the outcome and design the system to produce it.

Level 6: Admiral (Systems Integrator)

You document your best AI processes into reusable workflows. Your results are consistent because the system is consistent. You build infrastructure that compounds.

Human skill: Stakeholder navigation. Building AI systems for organizations requires trust and buy-in.

Level 7: Mission Director (AI Orchestrator)

You chain workflows into pipelines that run with minimal human intervention. You design feedback loops. You change how organizations work. The job of the future is yours because you are the most human, not the most technical.

Human skill: Inspirational leadership. Culture change and psychological safety at scale.

Find your level.

Take the free AI Proficiency Assessment at assess.launchready.ai

Harrison Painter

AI Business Strategist | LaunchReady.ai

AI@HarrisonPainter.com | 317.871.1000

linkedin.com/in/harrisonpainter | launchready.ai

Book a consultation: launchready.ai/book-a-call