POSSUMTREE PUBLISHING

Agent built SaaS projects for passive income

Book Proposal

Author: Leon A. Murray · 2026-08-17

The Book

An AI agent is the build team and a simple SaaS is the product: with current agents doing the scoping, coding, testing, and upkeep, one person with initiative can find a narrow market, build a simple recurring-revenue service to serve it, and run it as income — without a technical co-founder, venture capital, or a software background. This book lays out the whole sequence in simple steps — market discovery first, the anatomy of a simple SaaS applied generally to most services, an agent-led build method, honest pricing and launch, day-to-day operation, and the real math of passive income — then walks through two complete build examples and hands over the resource materials to start this week.

Who It's For

Primary: non-technical entrepreneurs and solo founders — side-hustlers, career changers, freelancers, and aspiring indie builders — who have initiative and everyday skill with AI chat tools but have never shipped software, and who want an income stream they own rather than another platform gig. Secondary: technically capable builders who lack the market-side method. The book assumes no code, no marketing degree, and no startup vocabulary — only willingness to follow simple steps and test what people will pay for. Every chapter answers the reader's three questions: what do I find, what do I build, what do I automate?

Why It's Different

What is on the shelf today:

1. SaaS and indie-hacking books — written for people who can code or hire; the solo non-technical reader is not in the picture.
2. AI-tool and prompt books — useful for productivity, but they teach usage, not a business; no market, no pricing, no product.
3. Make-money-online books — dropshipping, affiliate links, content mills; the reader never owns a product that pays them twice.
4. Technical build books — frameworks and code for developers; market discovery and income math are out of scope.
5. AI-vendor content — blog posts and white papers that each push one tool; the closest thing to this topic and exactly what a buyer cannot trust.

The empty position this book occupies: the complete, vendor-neutral, market-first route from initiative to a simple agent-built SaaS that people pay for — the essential anatomy applied generally, the agent build method, two worked builds, and honest passive-income math — written for a reader who has never written code. It is the only book on the shelf that gives the non-technical reader the whole order and then shows them the two examples end to end.

Why a reader picks this book over a free AI blog post: the blog post wants engagement and sells a tool; this book wants the reader to own a product. The book treats the agent as the employee and the reader as the employer — and the reader feels the difference in the first chapter.

The Structure — Twelve Chapters

01. The Agent-Owned Micro-SaaS
02. Finding a Market First
03. The Anatomy of a Simple SaaS
04. Your AI Build Team
05. From Market to Specification
06. Building the Service
07. Launch, Pricing, and First Customers
08. Running It Day to Day
09. Build Example One: The Reminder Service
10. Build Example Two: The Monitoring Service
11. The Math of Passive Income
12. The Agent-Built Portfolio

How the Book Is Built

Four parts, 12 chapters. Each entry gives the chapter's job and its core argument.

PART I — THE MAP (why this book exists)

Ch 1 — The Agent-Owned Micro-SaaS
Job: define the micro-SaaS and the shift that makes it a one-person business: AI agents that scope, code, test, and maintain software. Introduce the composite founder Priya and her first product, InvoiceNudge — a polite invoice-follow-up service for freelancers — which carries examples through the book. Preview the book's sequence: market, spec, build, launch, run, math.
Argues: the bottleneck used to be building software; agents remove it, so the bottleneck is now finding a market people will pay for — which is exactly what this book teaches.

Ch 2 — Finding a Market First
Job: the market-first discipline: pains and jobs-to-be-done, niche sizing, evidence that people already pay (tools, forums, competitors with customers), and the 100-customer test. Include the markets to walk past: crowded, free, or hobby-sized.
Argues: product-market fit starts before the product; a service nobody needs is a hobby, and

this book will not help you build one.

Ch 3 — The Anatomy of a Simple SaaS
Job: the outline of essential aspects applied generally to most services: the problem it solves, accounts and data, the core loop, billing and payments, the dashboard, notifications, hosting, security, and support. This is the general template the rest of the book — and both build examples — use.
Argues: every simple SaaS is the same machine; learn the anatomy once and any service — reminder, monitor, scheduler, tool — is a variation on it.

PART II — THE METHOD (how to build it)

Ch 4 — Your AI Build Team
Job: map agents onto the roles a small team used to fill — research, planning, coding, testing, documentation, operations support — name the tool categories by job (chat assistants, coding agents, workflow automations, testing agents), and set the review gate: agents propose, you dispose. Honest limits: agents hallucinate, drift, and cannot decide what the product should be.
Argues: the human supplies judgment, taste, and market sense; the agents supply the labor; and the review gate is what keeps the labor honest.

Ch 5 — From Market to Specification
Job: turn a found market into a one-page spec an agent can build from: the customer, the pain, the core loop, the data fields, the screens, and a pricing stub. Include the discipline of scoping small.
Argues: the spec is the contract between you and your agents; a vague spec produces a vague product, and the money is lost in the gap.

Ch 6 — Building the Service
Job: the general build sequence: scaffold the project, accounts and login, the core feature, billing, the dashboard, notifications, then test with agents until the flow is boring. Define the minimum paying version and the rule that every extra feature is debt.
Argues: build the smallest version people will pay for, and no more; the first version's job is to collect evidence, not to impress.

Ch 7 — Launch, Pricing, and First Customers
Job: pricing strategy for a simple SaaS — value-based, tiered, annual discount — the landing page and waitlist that pre-sell, the first ten customers and why they matter more than the first hundred, and onboarding that turns signups into renewals.
Argues: launch is a process that starts weeks before you have a product; the first ten customers teach the product what it is.

Ch 8 — Running It Day to Day
Job: the recurring rhythm: the weekly review, uptime and backups, monitoring and alerts, support on a schedule, update discipline, and letting agents automate the operation itself. Include what to do when it grows — and when it does not.
Argues: a SaaS pays in arrears — the machine only pays while it runs — so the operation

must be a rhythm one person can sustain alongside a life.

PART III — THE TWO BUILD EXAMPLES (proof the method transfers)

Ch 9 — Build Example One: The Reminder Service
Job: complete walk-through of InvoiceNudge, a polite invoice-follow-up service for freelancers: the market evidence, the one-page spec, the build with agents, pricing, launch, first customers, and the first month's numbers (illustrative). This is the small, fast, subscription-shaped build.
Argues: the method in Parts I and II is enough to ship a real, paying service in weeks — demonstrated end to end in one chapter.

Ch 10 — Build Example Two: The Monitoring Service
Job: complete walk-through of Trackline, a price-and-availability monitor for a niche market: a different data model (scheduled data pulls), a different billing shape (usage tiers), the same anatomy. Show the spec, the build, the launch, and what changed from the first example.
Argues: the anatomy transfers; the author's job is the same every time — find the pain, write it down, build the smallest paying version, then run it.

PART IV — THE INCOME (make it a business)

Ch 11 — The Math of Passive Income
Job: honest numbers: revenue per customer, churn, the path from first dollar to meaningful income, the real time budget, taxes and entity basics, and the definition of passive income as recurring, not effortless. Every figure labeled illustrative unless cited.
Argues: the honest math is what separates a business from a lottery ticket; run the numbers before you build, and again after launch.

Ch 12 — The Agent-Built Portfolio
Job: close the loop — find, build, launch, run, improve, and back to find — the 90-day first-product plan (days 1-30: market and spec; days 31-60: build and launch; days 61-90: first customers and the first month's numbers), the deliberate path to a second product, and where this way of working is headed.
Argues: one product is a job; the loop is the asset — and the book ends on the reader's first step this week.

Contract: 12 chapters; 2,000 words each; 24,000 words total. Typeset at 250 words per page: 24,000 / 250 = 96 text pages; plus ~8 front-matter pages (title, copyright, contents, introduction) and ~10 back-matter pages (resource appendix, glossary, index) = ~114 typeset pages.

Per-chapter frame (fixed, 600 words, every chapter): hook 150, What the Research Says sidebar 120, Real Voices vignette 120, Key Point callout 60, For Today 80, Chapter Summary 50, bridge 20.

Body (1,400 words per chapter, four sections; section budgets sum exactly to 1,400 per chapter — no rescaling pass needed):

Ch	Part	Body sections (word budgets)	Body sum	Total			
1	I	1.1 What a micro-SaaS i (420)	1.2 The build team that used to need five people (380)	1.3 What agents do and where you stay in charge (340)	1.4 What passive income really means (260)	1,400	2,000
2	I	2.1 Start with a pain, no a product (430)	2.2 Niches where people already pay (390)	2.3 The 100-customer test (330)	2.4 Markets to walk past (250)	1,400	2,000
3	I	3.1 The one machine under every service (450)	3.2 Accounts, data, and the core loop (380)	3.3 Billing, the dashboard, and notifications (320)	3.4 Hosting, security, and support essentials (250)	1,400	2,000
4	II	4.1 The roles agents play (440)	4.2 Choosing your tools (390)	4.3 The review gate: agents propose, you dispose (310)	4.4 What only you can do (260)	1,400	2,000
5	II	5.1 The one-page spec (430)	5.2 The core loop, written down (400)	5.3 Data fields and screens (330)	5.4 The pricing stub (240)	1,400	2,000
6	II	6.1 The build sequence, top to bottom (450)	6.2 Scaffold, accounts, and the core feature (380)	6.3 Billing and the dashboard (320)	6.4 Testing with agents until it is boring (250)	1,400	2,000
7	II	7.1 Pricing before launch (420)	7.2 The landing page and the waitlist (390)	7.3 Your first ten customers (330)	7.4 Onboarding that keeps them (260)	1,400	2,000
8	II	8.1 The weekly rhythm (430)	8.2 Uptime, backups, and monitoring (380)	8.3 Support on a schedule (340)	8.4 Automating the operation itself (250)	1,400	2,000
9	III	9.1 The market and the spec (440)	9.2 Building with agents step by step (390)	9.3 Launch and first customers (330)	9.4 What the first month taught the founder (240)	1,400	2,000
10	III	10.1 A different market, the same anatomy (450	10.2 Where this build differs (380)	10.3 Billing and data from the start (320)	10.4 What transfers from the first build (250)	1,400	2,000
11	IV	11.1 Revenue per customer, churn, and the runway (430)	11.2 The real time budget (400)	11.3 Taxes, entity, and paperwork basics (310)	11.4 Passive means recurring, not effortless (260)	1,400	2,000
12	IV	12.1 The loop: find, build, launch, run, improve (440)	12.2 The 90-day first-product plan (380)	12.3 A second product, deliberately (320)	12.4 Where this is headed (260)	1,400	2,000

Reconciliation rule (documented for safety; not needed here): if a chapter's body sections ever fail to sum to 1,400, chapter writers rescale them by the uniform factor 1,400 / body_sum, rounding to the nearest 10 with the residual absorbed into the largest sections; the 600-word frame is never rescaled. The frame and body together must always equal 2,000 words.

Research placeholders: every figure a chapter needs that is not an illustrative example is a clearly labeled research placeholder; chapter writers fill and verify them against current sources with Harvard author-date citations, per Section 6.

Research & Verification

• Scope discipline: this design is built strictly from the job text — niche AI for Entrepreneurs, audience anyone with initiative to capitalize on AI, subject an instructive manual that lays out in simple steps how to find a market and build a simple SaaS to serve it, covering the essential aspects of a SaaS applied generally, two build examples, and resource materials — plus standard domain reasoning about micro-SaaS economics and agent capabilities.
• No external statistics, vendor claims, or market figures are asserted as fact in this design.
• A live web_search pass (2025-2026) returned the following sources, which anchor the book's research thread;
• chapter writers must verify each source exists, confirm the claim against the original, prefer primary research, and cite Harvard author-date (Author, Year) at the point of use: - Freemius (2025).
• State of Micro-SaaS — how solo founders ship and operate products that once required full teams (Ch 1, 11).
• The Founder Nation (2025).
• The Micro-SaaS Trend — solo founders turning AI into one-person SaaS businesses (Ch 1, 2).

• TNW (2025).
• How AI Is Powering the Next Wave of Micro-SaaS Entrepreneurs — AI assistants collapsing the cost of building software (Ch 1, 6).
• CB Insights (2025).
• Coding AI Agents Are Taking Off — the coding-agent and copilot market scaling (Ch 4, 12).
• DEV Community (2025).
• The Solo Dev SaaS Stack — solo developers running micro-SaaS businesses (Ch 8, 12).
• Sidetool (2025).
• One-Person SaaS Success Stories — solo revenue stories for chapter writers to verify or replace with primary sources (Ch 11).
• RapidDev (2025).
• Micro SaaS Ideas for Solopreneurs — a niche-idea map for the discovery chapters (Ch 2).
• EntrepreneurLoop (2026).
• Bootstrapped SaaS Niches for Solo Founders — niche and market evidence (Ch 2).
• Searchlab (2026).
• SaaS Statistics — industry market context;
• figures to verify against primary research (Ch 11).
• BuildMvpFast (2026).
• Agentic AI Productivity Gains — adoption data pointing to McKinsey's 2025 State of AI as the primary source (Ch 4, 12).
• Practitioner discussions (r/SaaS, 2026).
• The Passive Income SaaS Fantasy — the honest counterpoint the book adopts in Ch 11.
• What the book will need researched during drafting (explicit placeholders, so chapter writers know what to verify and cite): - Current agent tool landscape for coding, testing, and operations (Ch 4, 6, 8) — names and pricing change fast;
• the book evaluates categories and provides the scorecard;
• chapter writers supply current, cited examples.
• Current statistics on SaaS pricing, churn, solo-founder outcomes, and AI-agent adoption (Ch 1, 11) — to be sourced and cited, or replaced by clearly labeled illustrative numbers.
• Billing and payments provider practices for micro-SaaS (Ch 3, 6) — the categories are stable;
• specific providers and fee structures are versioned and must be verified at writing time.
• Legal basics for a solo online business: entity choice, sales tax, and platform terms (Ch 11) — jurisdiction-specific;
• the book covers the questions to ask, with specifics marked for verification.
• Rules for chapter writers: never assert a statistic without a source;
• label every composite and every illustrative number as such;
• keep the vendor-neutral posture — no endorsements, only evaluation frameworks;
• and when in doubt, mark it as a research placeholder rather than writing a confident guess.

Tone

• Mentor to reader, plain-spoken, practical.
• Think: someone who has built and run these products sitting down with you for an hour per chapter, showing you exactly what to do next.
• Voice rules: - Short chapters, short sections, direct sentences.
• Simple steps, in order.

• No throat-clearing.
• Concrete over abstract: every idea lands in a scene — the freelancer whose invoice goes unpaid, the first customer who asks for a refund, the agent that confidently ships the wrong page.
• Honest about effort and odds: the book never promises a passive fortune from a weekend;
• it promises a working system built in simple steps, with the math to judge it.
• Vendor-skeptical: tools are named by category and scored, never endorsed;
• today's product names are research placeholders for chapter writers.
• Respectful: the reader is not naive;
• the book never talks down, never oversells, and never shames a failed first attempt.
• Every number in a worked example is labeled illustrative unless it carries a citation.
• No footnotes in the reading flow;
• sources and further reading sit in endnotes that chapter writers populate with verified citations.
• Voice sample (for chapter writers): 'Your SaaS does not need a team.
• It needs a market that hurts, a spec that says what to build, an agent that builds it, and a rhythm that keeps it running.
• That is the whole book — the rest is the order, the details, and the math.'

Narrative Arc

• The reader's journey has four movements and one steady emotional climb: Movement 1 — Map (Ch 1-3).
• The reader arrives possibly skeptical and certainly busy.
• Ch 1 reframes their situation: the build-team problem has been solved for them;
• Ch 2 replaces the fear of picking the wrong market with a concrete discovery method;
• Ch 3 shows the one machine under every simple service, so the reader can see the whole book in miniature.
• Emotional state at the end of Part I: clarity — I can see the shape of this.
• Movement 2 — Method (Ch 4-8).
• The book switches from map to construction, one layer per chapter in strict dependency order: team before spec, spec before build, build before launch, launch before operation.
• Each chapter is self-contained enough to be delegated and sequential enough to be cumulative.
• Emotional state: competence — the reader sees the system taking shape.
• Movement 3 — Proof (Ch 9-10).
• Two complete builds, end to end.
• The reader watches the same anatomy applied to a reminder service and then to a monitoring service;
• what was abstract becomes a template they can imitate.
• Emotional state: belief — if it transfers twice, it will transfer for me.
• Movement 4 — Income (Ch 11-12).
• Ch 11 reframes passive honestly: recurring, measured, real — the numbers the reader needs before and after launch.
• Ch 12 closes the loop, makes the 90-day plan concrete starting Monday, and ends on the image of the agent-built portfolio running quietly while the owner watches the monthly numbers.

• Emotional state: readiness — I start this week.
• Continuity devices the chapter writers must use: - Priya, the composite founder, appears in the method chapters (Ch 1, 5, 6, 7, 8) and in Build Example One (Ch 9);
• her second product, Trackline, carries Build Example Two (Ch 10).
• Always labeled composite — Priya, InvoiceNudge, and Trackline are not real.
• The anatomy from Ch 3 is the organizing diagram;
• build chapters and both examples reference it by name, so the reader never loses the thread.
• Each chapter's bridge (20 words) hands the reader to the next chapter's question.
• For Today (80 words) converts each chapter into one concrete reader step.
• Every money and usage figure in examples is labeled illustrative;
• research-backed numbers carry citations from the chapter writer.