IDEA TO INVOICE · 09

Build the Thing

A field guide for the moment after validation. How to turn a real idea into the smallest thing that lets a real customer pay a real price — without building more than you need.

AUTHOR

Ted Milton
Founder, Expedient AI

EDITION

09 of 12

FORMAT

Field guide · PDF
12 pages

READ ME FIRST

Before you start

This guide starts where Edition 08 ends. It assumes the score said “go” and the work begins this week.

You have done the validation work. The idea survived. Now you need to turn it into something a customer can pay for, in a way that does not bankrupt you before the first invoice lands. This guide covers the four decisions that matter in the first month: build or buy, the offer in plain words, the smallest working prototype, pricing that you can defend, and the boring admin that nobody wants to write but everyone needs. None of it costs more than £20 a month.

This is for you if

You have completed the four checks in Edition 08 and the score said “go” or “pivot with the fixed input”.

You have a specific customer, a specific problem, and a specific price in mind.

You have at least twenty hours a week to give this for the next three months.

You can ship something ugly, then improve it, instead of waiting for it to be perfect.

You are willing to be the customer support, the salesperson, and the founder for the first year.

This is not for you if

You have not validated yet. Go back to Edition 08. The build decisions below assume the work has been done.

You are looking for a no-code miracle. The tools help, they do not replace the work.

You are building a hardware product, regulated service, or anything that needs formal certifications. Different guide.

You have outside money committed and an investor breathing down your neck. Their expectations are not your friend in week one.

You want a five-minute AI hack that builds the whole thing. There is no such hack.

> The honest summary: the first version is the one nobody sees. Its job is to confirm the customer will pay and to find the parts of the offer that do not survive contact with reality. Build small, charge early, fix what breaks.

WHAT IS IN HERE

Contents

Five chapters, two worksheets, one warning that may save you from a real mistake. Read in order, then keep the worksheets for the build.

How to use this guide

Chapter 1 gives you the build-or-buy decision framework. Chapter 2 is the offer in plain words, with prompts that do not produce slop. Chapter 3 covers the smallest working prototype using AI-assisted builders. Chapter 4 is pricing, with the maths done honestly. Chapter 5 covers the legal and admin essentials. The two worksheets are the build plan and the pricing scenarios. Fill both in before you spend a penny on building.

Tools used in this guide

Claude for drafting offer copy and admin documents. v0.dev free tier for landing pages and simple interfaces. Bolt.new free tier for app prototypes. ChatGPT for pricing scenario maths. None of these require a paid plan to complete this guide. Total monthly cost during the build: £0–£20.

CHAPTER 1

What to build vs. what to buy

Every component of your product is a build-or-buy decision. Get each one right and the cost is manageable. Get them wrong and you spend six months on plumbing.

A solopreneur cannot build everything. The question is not “should I code it myself”. The question is: for each piece of the product, is the marginal value of building it higher than the marginal cost of buying a working version? Almost every product is a stack of ten to twenty components, and the right answer for each one is different. The matrix below is the first cut of the decision.

The component matrix

COMPONENT	DEFAULT ANSWER	BUILD WHEN	BUY WHEN
Landing page	Buy	You are a designer-founder and it is your differentiator	Always. A Carrd or Notion page is fine for week one.
Authentication	Buy	You are selling to enterprise with SSO requirements	Always. Clerk, Auth0, Supabase Auth: all free tiers.
Database	Buy	Never, in year one	Supabase, Airtable, or Firebase free tier.
Payments	Buy	Never	Stripe or Lemon Squeezy. Subscription billing is solved.
Email transactional	Buy	Never	Resend or Postmark. Free tiers cover the first thousand sends.
Core product logic	Build	Always	There is no off-the-shelf version of your specific idea.
Reporting / dashboards	Build lite	The view of the data is your differentiator	Buy a Metabase instance or use the database UI.
Customer support tool	Buy	Never, in year one	A shared inbox works until you have fifty customers.
Scheduling / booking	Buy	Your entire product is the scheduling	Calendly, Cal.com free tier, or SavvyCal.
Document e-signature	Buy	Never	Documenso free tier covers small volumes.

The build rule of thumb

Build only what the customer can see and feel. Build the thing that, if it is missing or wrong, makes them leave. Buy everything else, even when buying feels expensive, because buying saves the time you need to spend on the part that matters. A solo founder with six months of runway should spend four of them on the core product and the other two on everything else combined.

When AI-assisted building beats paying

Two years ago, building anything meant hiring a developer or learning to code. In 2026, a non-technical founder can ship a working web app in a weekend using Claude, v0, or Bolt. The output is not as polished as a custom build, but for the first version it does not need to be. It needs to work, it needs to load in under three seconds, and it needs to do the one thing the customer is paying for. Everything else can wait.

The exception is anything that handles money, identity, or compliance. Use Stripe, Clerk, and a regulated vendor. The cost of getting those wrong personally is not worth the saving.

> If a vendor offers a free tier that covers the first hundred customers, take it. You can swap in a paid alternative in month four when you have revenue. Switching costs are low in the first year.

CHAPTER 2

Your offer, in words people understand

If your customer cannot repeat your offer back to you in one sentence, the offer is not ready. AI is good at drafting. It is bad at clarity. That is your job.

The single biggest reason a customer does not buy is that they did not understand the offer in the first three seconds. Most solopreneur offers are vague because the founder has been living inside the idea for months and assumes context the customer does not have. The fix is one sentence, in plain words, that a stranger can read once and repeat back to a friend. This chapter gives you the prompts that get you there, and the editing rules that stop AI from making it worse.

The one-sentence test

Your offer, in one sentence, must answer four questions. Who is it for. What problem it solves. What they get. What they pay. If any of the four is missing, the offer is not finished. Test it by reading the sentence to a friend who is not in your industry and asking them to repeat it back. If they cannot, the sentence needs another edit.

Bad: “A platform that empowers creators to monetise their workflow.” Better: “A weekly email that gives UK freelance photographers three new client leads, for £29 a month.” The second one is answerable. The first one is not.

Prompts that do not produce slop

Most AI-written offers sound the same because the prompts are the same. The prompts below are designed to break the pattern. Run them in Claude or ChatGPT and treat the output as raw material, not copy.

```
You are writing a landing-page headline for [PRODUCT]. The customer is [WHO]. The
problem is [PROBLEM]. Give me 20 options, each under 12 words, each using a different
rhetorical pattern: (a) a number, (b) a question, (c) a forbidden action, (d) a
before-and-after, (e) a named authority, (f) a specific time period, (g) a
counterintuitive claim, (h) a direct address, (i) a comparison, (j) a single word
followed by a colon. Do not write features. Write outcomes. No exclamation marks.
```

Read the 20 options. Pick the three that are specific, measurable, and would make the customer nod. Discard the rest. Combine the best parts into one sentence. That sentence is the working headline. The version on the website is the eighth draft, not the first.

The editing rules

- ✓ If a sentence works in a competitor's ad, delete it. It is also the sentence the customer has stopped reading.
- ✓ Replace adjectives with numbers. “Affordable” becomes “£29 a month”. “Fast” becomes “results in 48 hours”. Numbers are the cheapest voice you can buy.
- ✓ Delete the first paragraph of any draft. It is almost always throat-clearing.
- ✓ Read it aloud. Anywhere you stumble, the reader will too.
- ✓ Show it to one stranger, get their first reaction, then ask what they think it costs. If their guess is off by more than 30%, the offer is unclear.

One-pager and landing page

Once the sentence is settled, write a one-page document covering: who it is for, the problem in their words, what they get, what they pay, what happens after they pay, and how to cancel. Keep it to one printed page. The landing page on the website is the same content, broken into the same sections, with the headline first and the cancellation policy visible. Trust is built by showing the exit as clearly as the entry.

CHAPTER 3

A working prototype in a weekend

The first version of the product is the one nobody sees. Its job is to confirm a customer will pay, not to win design awards.

A working prototype in 2026 means a usable interface that does the one thing the customer is paying for, with payment connected and a real way to contact you. It is not pretty. It is not feature-complete. It does the core job and it does not crash. The good news is that three AI-assisted tools can get you there in a weekend if you go in with the right brief. The bad news is the same as it has always been: the brief is the work.

The three tools and what each is good for

TOOL	GOOD FOR	NOT GOOD FOR	FREE TIER
v0.dev	Landing pages, marketing UI, simple dashboards, anything visual	Complex app logic, real databases, authentication flows	~10 generations per day. Enough for one weekend.
Bolt.new	Full web apps with a backend, simple data models, payment flows	Mobile apps, anything that needs native performance	Limited credits. Enough for one prototype.
Lovable	Mid-complexity apps with clean UI, good for client-facing tools	Heavy data processing, anything that needs real infrastructure	Limited credits on the free tier.

For most solo founders in week one, the right answer is Bolt or Lovable for the working app, and v0 for the marketing page that links to it. Do not pick all three. Pick one, learn it, ship with it.

The weekend plan

1. **Friday evening.** Write the one-sentence offer from chapter 2 in plain text. List the three things the product must do for the customer to pay. Anything not on that list is not in the weekend prototype.
2. **Saturday morning.** Use v0 to generate the landing page from the one-sentence offer. Edit until the page reads cleanly and the call-to-action is unambiguous. Connect a Stripe payment link in test mode.
3. **Saturday afternoon.** Use Bolt or Lovable to generate the core product: the one screen that does the one job. Connect Stripe to capture a real payment, even if you refund it later.
4. **Sunday morning.** Write the welcome email that fires when a customer pays. Test the full flow: landing page → payment → product access → welcome email. Run it on your own phone.
5. **Sunday afternoon.** Find one stranger from your Edition 08 conversations, ask them to use it. Watch them. Note where they hesitate. Fix the worst hesitation, not all of them.

What to skip on weekend one

Skip a custom domain if the Bolt free URL is clean enough. Skip analytics beyond the basics. Skip automated onboarding beyond the welcome email. Skip mobile responsiveness beyond what the tool generates by default. Skip everything that is not the one job. You can add it in week two.

> The point of the weekend prototype is to find out whether a real person will pay a real price and use the thing. Everything else is decoration. If the prototype is ugly but works, you have a business. If it is beautiful and slow, you have a portfolio piece.

CHAPTER 4

Pricing without guessing

Most first-time pricing is set by vibes. There is a better method, and it costs an afternoon.

Pricing is the part of the business where guessing costs the most and is easiest to fix. A £5 mistake on the price, compounded across a year of customers, is the difference between a profitable business and a hobby. The method below uses three numbers — your cost to deliver, the customer's perceived value, and the competitor's price — to set a price you can defend when a customer asks why it costs that much.

The three numbers

Cost to deliver. Your time, the tool subscriptions, the third-party services, and the customer support hours, divided by the number of customers you expect to have. If the cost is £4 per customer per month, that is your floor.

Customer's perceived value. What does the customer save or earn by using your product, per month? A pricing tool that saves a freelancer two hours a month at £50 an hour is worth £100 to them. Charging £30 is a clear win. Charging £150 loses the sale. The right price is below their perceived value and above your cost, with enough margin to survive when things go wrong.

Competitor's price. If five competitors all charge £20 a month, your price needs a reason to be different. Cheaper works only if your cost allows it. More expensive works only if your offer is clearly different. Same price with a different angle is the safe default for week one.

Cost-plus vs value-based

Cost-plus sets price as cost multiplied by a margin. It guarantees you make money per customer. It often leaves money on the table when the customer would have paid more.

Value-based sets price at a fraction of the customer's perceived value. It captures more of the surplus. It is the right method when you can articulate the value clearly, which most solo founders cannot in week one.

The honest hybrid for week one: use cost-plus as your floor, value-based as your target, and competitor price as a sanity check. Aim for the highest of the three that still feels defensible. After three paying customers, you have evidence to move the price.

AI-assisted scenario modelling

Use ChatGPT or Claude with this prompt:

```
I am pricing [PRODUCT]. My cost to deliver per customer per month is £[X]. The
customer's perceived value is £[Y]. The average competitor price is £[Z]. My expected
customer count is [N]. Model three scenarios: (a) cost-plus at 3x margin, (b) value-
based at 30% of perceived value, (c) competitor-aligned at the average. Show monthly
revenue, monthly profit, and break-even customer count for each. Flag any scenario
where profit is below zero at expected customer count.
```

Read the three scenarios. Pick the one with the highest profit at the expected customer count that does not require the customer to believe a price 50% above the market average. That is your week-one price.

> The most common pricing mistake new founders make is setting a low price to “be safe”. A low price signals low value, and the customers it attracts are the most price-sensitive and the most likely to churn. Charge enough to take the work seriously and to keep the customers who do the same.

CHAPTER 4 · CONTINUED

Pricing scenarios, side by side

A worked example, the band that picks your week-one price, and the rule for changing it later.

Numbers are easier to defend than feelings. The band below is what your three scenarios should produce. The example that follows is a typical solo-founder pricing decision, fully worked, with the week-one price justified in one paragraph.

The pricing band

Floor

Cost-plus at 3x your cost to deliver. The price below which you lose money on every customer. You should never go here.

Target

Value-based at 20–40% of the customer's perceived monthly value. The price you charge if the value story is clear and you can defend it.

Ceiling

The competitor average plus the smallest premium that still feels honest. Going above this without a clearly different offer loses sales.

A worked example

A freelance photographer in Manchester is launching a weekly email that sends three new client leads, in the UK, on a recurring basis. The cost to deliver is £4 per customer per month (Stripe fees, Resend, and two hours of research at £15 an hour). The customer's perceived value is £150 a month (one new client from a single lead pays for the year). The average competitor price for a similar newsletter is £19 a month.

Cost-plus 3x: £12 a month. Value-based at 30%: £45 a month. Competitor-aligned plus a small premium: £24 a month. The week-one price is £24 a month. It is above the cost-plus floor, well below the value-based target, and a defensible £5 above the competitor average, justified by the claim of three leads per week (most competitors send one).

After fifty paying customers, revisit. If churn is below 5% a month, raise to £29. If churn is above 10%, the price is above the value, and either the offer needs sharpening or the price needs lowering.

When to change the price

Raise it when the customer acquisition cost is comfortably below half the price and churn is below 5% a month. Lower it when the take-up rate is below 2% of the people who see the offer. Change the offer when neither move is enough, because the price is not the problem.

Do not change the price for a single customer. The customer who complained is the customer you will lose to the next price change too. Change it for the cohort, not for the complaint.

Annual vs monthly

Default to monthly. Offer annual at a 15–20% discount for the customers who want it. Annual customers are stickier, which is good for retention and bad if the product needs a major redesign. Solopreneurs usually want monthly for the first six months until they trust the product will still be there in year two.

> Pricing is the lever you can pull that needs no code, no design, and no extra work. The first price you set is a guess. The third price you set, after fifty customers and a churn number, is a decision.

CHAPTER 5

The boring admin AI can handle

Terms, privacy, contracts — the parts nobody wants to write and everybody needs. AI drafts them in an hour. A human must check them.

Every solopreneur eventually needs the same documents: a terms of service, a privacy policy, a basic service agreement, sometimes a data processing agreement, and almost always a refund policy. None of these are exciting. All of them are necessary. AI is excellent at drafting the first version of each, in plain English, in under an hour. AI is not excellent at knowing when the draft is wrong, and getting them wrong is the kind of mistake a small business does not recover from quickly.

What AI can draft

- ✓ Privacy policy (UK GDPR-aware, free draft from a reputable template)
- ✓ Terms of service (one-pager, plain English, no legalese)
- ✓ Refund and cancellation policy (the one customers actually read)
- ✓ Basic service agreement (for B2B customers, with deliverables and payment terms)
- ✓ Cookie notice (one paragraph, plus the small banner)
- ✓ Data processing agreement (if you handle customer data on their behalf)
- ✓ Email template for handling customer complaints (yes, even this)

The AI prompt to draft each is straightforward. Give it the product, the customer, the price, and the data you collect, and ask for a plain-English draft under 800 words. Edit for things that are wrong. Do not delete things you do not understand.

What a human must check

> **DO NOT SKIP THIS**
> AI-drafted legal documents are starting points, not finished work. A first-draft privacy policy written by Claude is approximately 80% right and 20% wrong in ways that are easy to miss. The 20% includes: jurisdiction mismatches (UK vs US data law), missing clauses for the specific data you actually collect, inaccurate descriptions of third parties, and ambiguous language about your obligations. Get the drafts reviewed by a human — ideally a solicitor, otherwise a knowledgeable friend with relevant experience — before they go on the website. The cost of a one-hour review is £150 to £400. The cost of getting it wrong is the cost of a customer complaint at the worst possible moment, plus the ICO's attention if you are based in the UK.

The honest rule: AI drafts the first version, a human reviews it, you edit the human's comments, then you publish. The AI saves you the blank-page problem. The human catches what the AI does not know. Neither alone is enough.

The minimum viable admin stack

For week one, you need exactly four documents: privacy policy, terms of service, refund policy, and one email template for handling complaints. Everything else can wait. Put all four on a single /legal page on your website. Link to them from the footer, from the checkout page, and from the welcome email. Customers will not read them. The point is that they exist and are correct when someone asks.

For week two, add the service agreement if you sell to businesses. For month two, add the data processing agreement if any of your customers are passing personal data through you. For year one, plan to review all four documents once and update them once, ideally with professional help.

What AI cannot do

AI cannot tell you whether your specific business model triggers a regulatory requirement it does not know

WORKSHEET 1 OF 2

Your build plan, on one page

Fill this in by hand before the weekend prototype. The plan is short on purpose: the constraint is your time, not your ambition.

Two short worksheets on this page. The first is the build-vs-buy decision for each component. The second is the four documents from chapter 5. Both fit on one page because the value is in the deciding, not in producing a forty-row spreadsheet nobody reads.

Build vs buy decisions

Core product logic

Landing page

Authentication

Database

Payments

Transactional email

Reporting / dashboard

Scheduling / booking

Customer support tool

My chosen AI builder

Admin document checklist

Privacy policy — drafted by

Reviewed by (human)

Published date

Terms of service — drafted by

Reviewed by

Published date

Refund policy — drafted by

Reviewed by

Complaint email template

Service agreement (B2B)

If a row is empty on launch day, that row is the gap that will hurt you. Fill it in before the first paying customer, not after.

WORKSHEET 2 OF 2

Your pricing scenarios

Fill this in by hand before you set the price on the checkout page. The numbers will not write themselves.

Two short worksheets on this page. The first captures the three numbers from chapter 4. The second is the three scenarios and your chosen week-one price. Both fit on one page because the decision is supposed to be quick. If you cannot fill it in in fifteen minutes, the inputs are not yet known.

The three numbers

My cost to deliver, per customer, per month

My customer's perceived value, per month

Average competitor price, per month

Expected customer count in month 3

Cost-plus floor (cost × 3)

Value-based target (30% of value)

Competitor ceiling (avg + premium)

The three scenarios

Scenario A — cost-plus: monthly revenue

Scenario A — monthly profit

Scenario B — value-based: monthly revenue

Scenario B — monthly profit

Scenario C — competitor-aligned: monthly revenue

Scenario C — monthly profit

Chosen week-one price

My justification in one sentence

Re-price trigger (churn or uptake %)

Re-price date (target)

The justification sentence is the one you say aloud when a customer asks why it costs that much. If you cannot say it without hesitating, the price is not yet defended.

ABOUT THIS GUIDE

Colophon

Everything you might want to know about where this came from and what to read next.

AUTHOR

Ted Milton, Founder, Expedient AI

An independent practitioner in Bristol writing field guides for one-person businesses that would rather be shipping than writing a business plan about shipping.

TOOLS USED

Claude for drafting offer copy and admin documents. v0.dev free tier for the landing page. Bolt.new free tier for the working prototype. ChatGPT for pricing scenario maths. Total monthly cost during the build: £0–£20.

EDITION

09 of 12 in the Idea to Invoice series.

Published July 2026. Printed and bound in PDF.

SERIES FORMAT

A4, twelve pages, one field guide per quarter. British English. No exclamation marks. No testimonials. Website contact only.

The Idea to Invoice series

01 — Where AI Actually Helps

02 — The 30-Day AI Pilot

03 — Your Data Your Rules

04 — Writing AI Instructions That Actually Work

05 — The AI Vendor Interrogation Kit

06 — Your First AI Employee

07 — Set Up Your Own AI Agent

08 — Is This Worth Doing?

09 — Build the Thing (this guide)

10 — Get Found

11 — Get the Conversation

12 — Close the Deal

> The next guide in the series is Edition 10 — Get Found. It picks up where this one stops, with the marketing plan that gets the product in front of the customer you have already defined. Until then: ship the smallest thing, charge a real price, and write the four documents before the first paying customer.