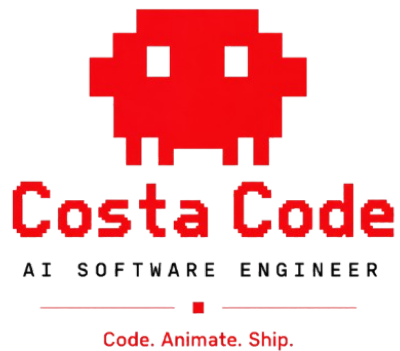




Costa Code

AI SOFTWARE ENGINEER — INSIDE COSTA STUDIO

The complete brand system in both themes, how to build anything from a single prompt, and how to turn it into income.

DOCUMENT

Brand & Business Guide

EDITION

English

VERSION

v3 — Web + CLI

AUTHOR

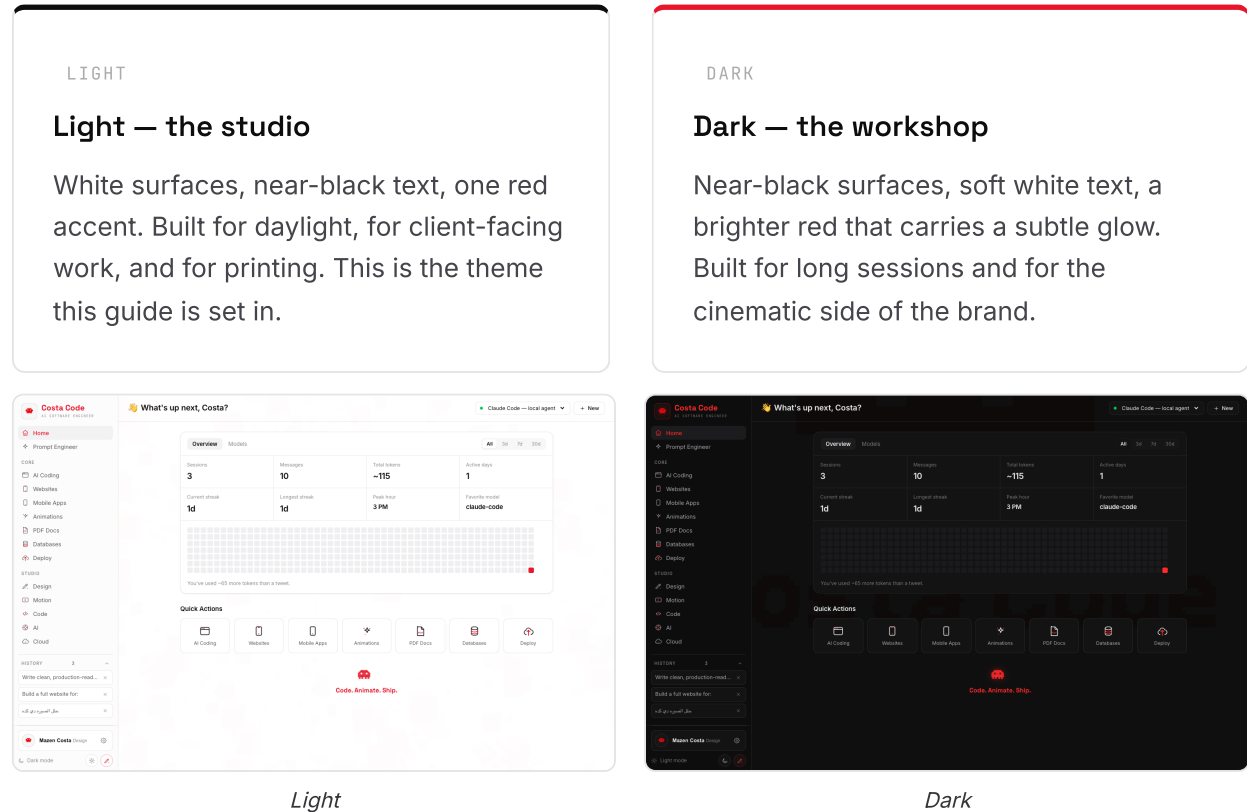
Mazen Costa

Contents

The dual identity — one brand, two skins	01
Colour system — light & dark	02
Typography & the rules that hold it together	03
The mascot — ten states, one silhouette	04
Motion — what actually moves, and why	05
Icons & logo in both themes	06
The Website — light mode	07
The Website — dark mode	08
The Terminal — the same brand in a shell	09
One prompt → anything	10
A real build, start to finish	11
Getting the most out of it	12
Turning it into income — the honest map	13
Productised services you can sell	14
Selling it as a subscription	15
Pricing & packaging	16
The 30-day plan	17

The dual identity

Costa Code is not a dark app with a light option bolted on. Both themes are complete, deliberate designs built from the same set of tokens — every colour, border, shadow and glow has a defined value in each. Nothing is simply inverted.



The rule that keeps both honest: **one red element per component**. Red marks the single most important thing on screen — never decoration, never a gradient, never a glow for its own sake.

Colour system

Light palette

					
#E8162A	#FFFFFF	#F7F7F8	#0D0D0D	#86868E	#E4E4E7
Accent	Surface	Panel	Text	Muted	Border

Dark palette

					
#FF2A2A	#0A0A0A	#131314	#EDEDED	#77777E	#1F1F22
Accent	Surface	Panel	Text	Muted	Border

ROLE

LIGHT

DARK

Accent

The one thing that matters on this screen

#E8162A

#FF2A2A

Surface

The page itself

#FFFFFF

#0A0A0A

Panel

Cards, inputs, raised areas

#F7F7F8

#131314

Text

Primary reading colour

#0D0D0D

#EDEDED

Muted

Labels, hints, secondary info

#86868E

#77777E

Border

Separation without weight

#E4E4E7

#1F1F22

Both accents are the same red at different brightness: **#E8162A** holds up against white, **#FF2A2A** stays vivid against near-black. Using one value for both would make it either muddy or blinding depending on the theme.

Typography

Three faces, three jobs. No decorative fonts, no more than three weights in play at once.

FACE	ROLE	WHERE	WEIGHTS
Space Grotesk	Display	Brand name, hero headline, section titles	600 / 700
Inter	Body	All interface text, paragraphs, buttons	400 / 500 / 600
JetBrains Mono	Mono	Code, labels, keys, stats, the terminal	400 / 500 / 700
Cairo	Arabic	All Arabic text, including this guide	400 / 600 / 700

Costa Code

Space Grotesk — Display

Build anything with a single prompt.

Inter — Body

costa build acme-landing

JetBrains Mono — Mono

ابني أي حاجة يبرومبت واحد

Cairo — Arabic

Rules worth keeping

Labels and eyebrows are mono, uppercase, wide letter-spacing — they read as system text, not as content.

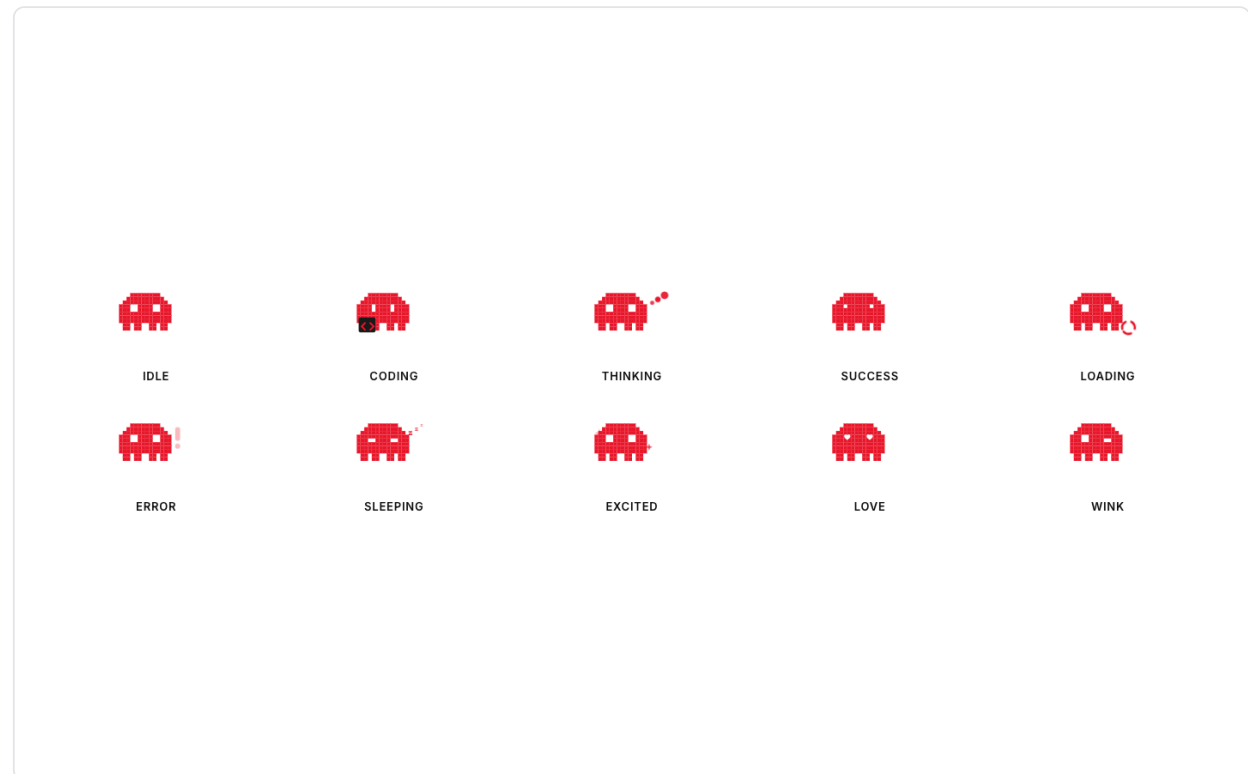
Headings never use the body face; body text never uses the display face.

Numbers in the dashboard are mono so columns line up.

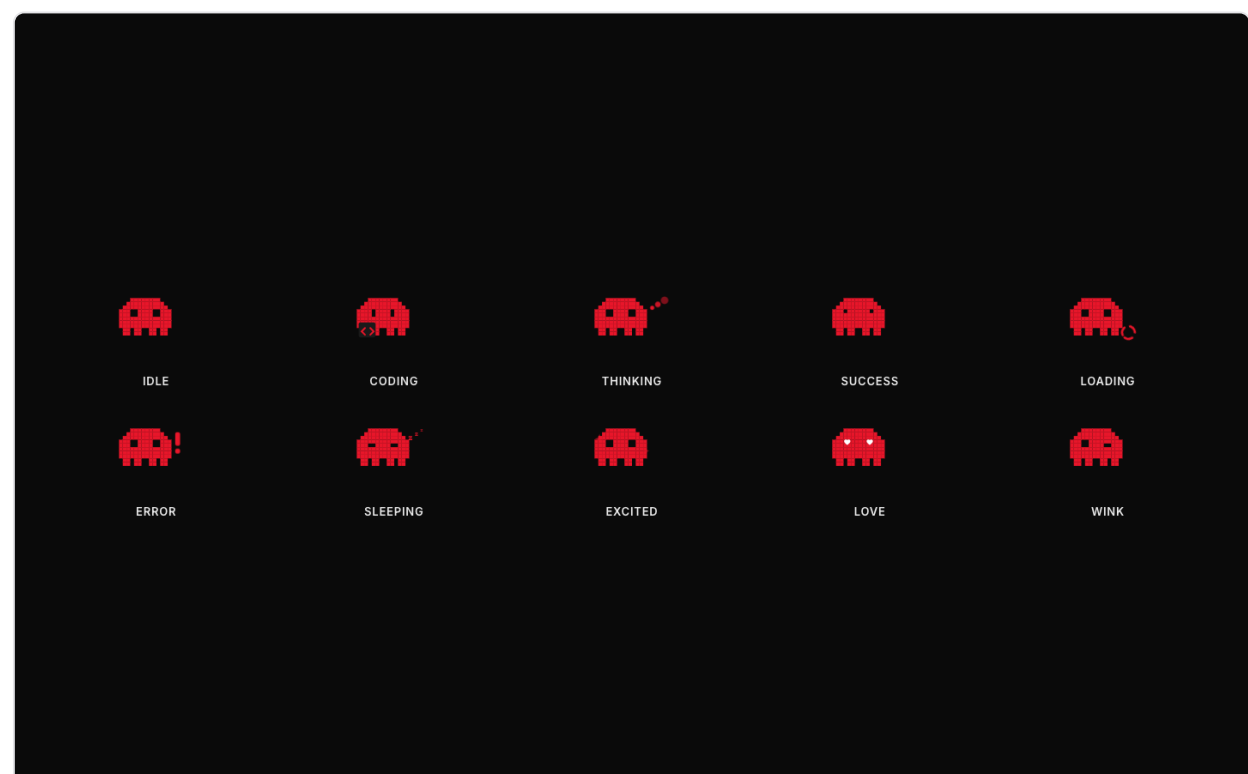
Arabic always sets in Cairo — never a fallback, never a Latin face doing Arabic badly.

The mascot

Every expression is generated from one bitmap grid in code. The silhouette is identical across all ten — only the eye row and a small accessory layer change. That is why it can never drift off model, at any size, in either theme.



Light



The ten states

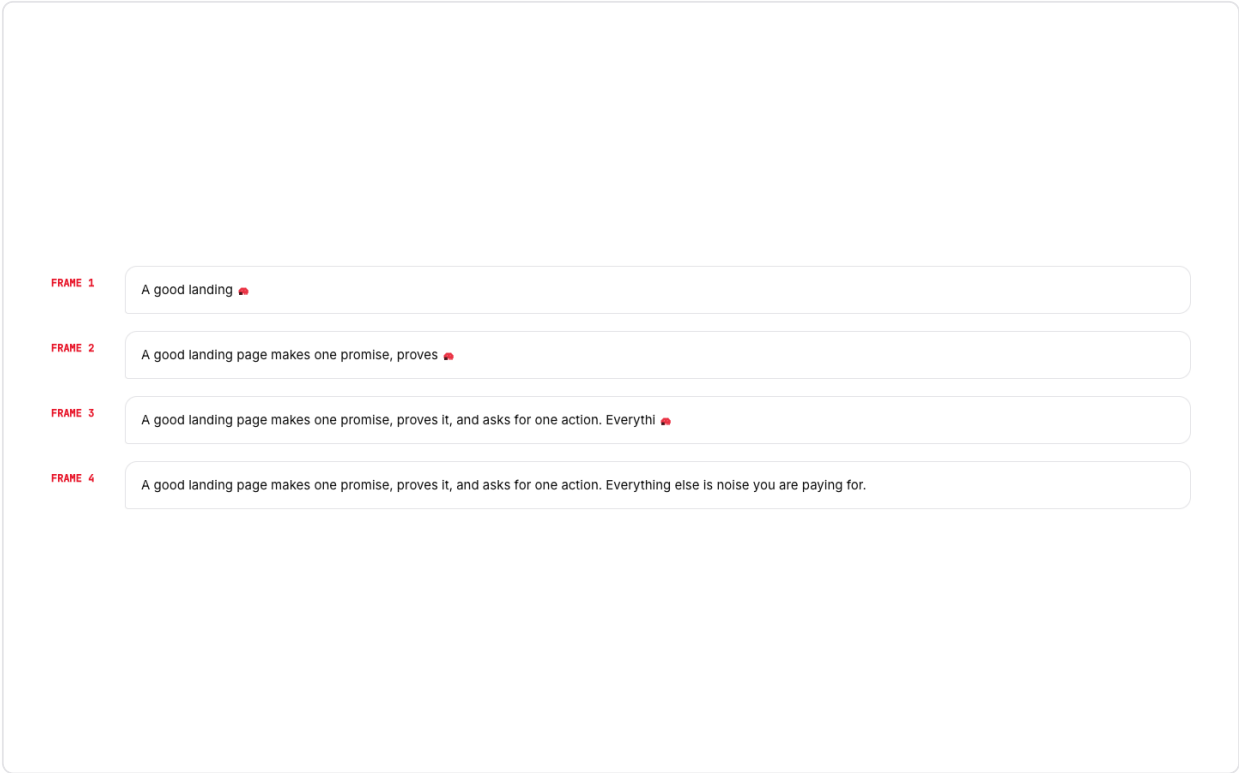
STATE	TRIGGER
Idle	At rest — a slow breathing scale
Thinking	From Send until the first character arrives
Coding	While the reply types itself out
Success	Reply finished cleanly — a brief pop
Error	Request failed — a brief shake
Loading	Work in progress with no output yet
Sleeping	Two minutes with no interaction
Excited / Love / Wink	Reserved for celebration and personality moments

Motion

Motion in Costa Code exists to tell you what the system is doing. If an animation does not answer "what is happening right now?", it does not ship.

The typewriter reveal

The local engine returns a whole answer in one burst. Shown raw, a long reply would simply appear. Costa Code buffers it and reveals it at a steady pace — faster when there is a backlog — so it always reads as being written. The mascot walks along at the end of the text as the cursor.



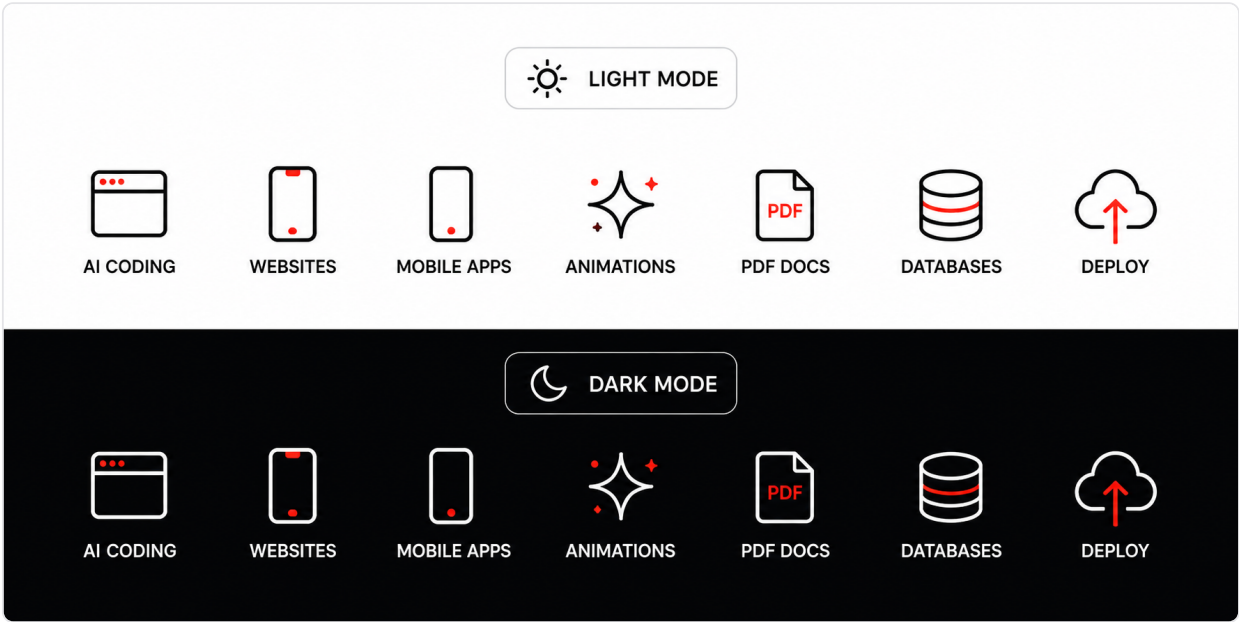
The typewriter reveal

ELEMENT	DURATION	EASING	WHAT IT SAYS
Mascot breathe	4s	ease-in-out, infinite	The app is alive and idle
Mascot bob	0.85s	ease-in-out, infinite	Thinking or writing
Hero float	5s	ease-in-out, infinite	The chat is ready for you
Message in	0.28s	ease-out-expo	A new message arrived
Panel reveal	0.18–0.22s	ease-out-expo	Something opened
Button press	0.12s	scale 0.96	Your click registered

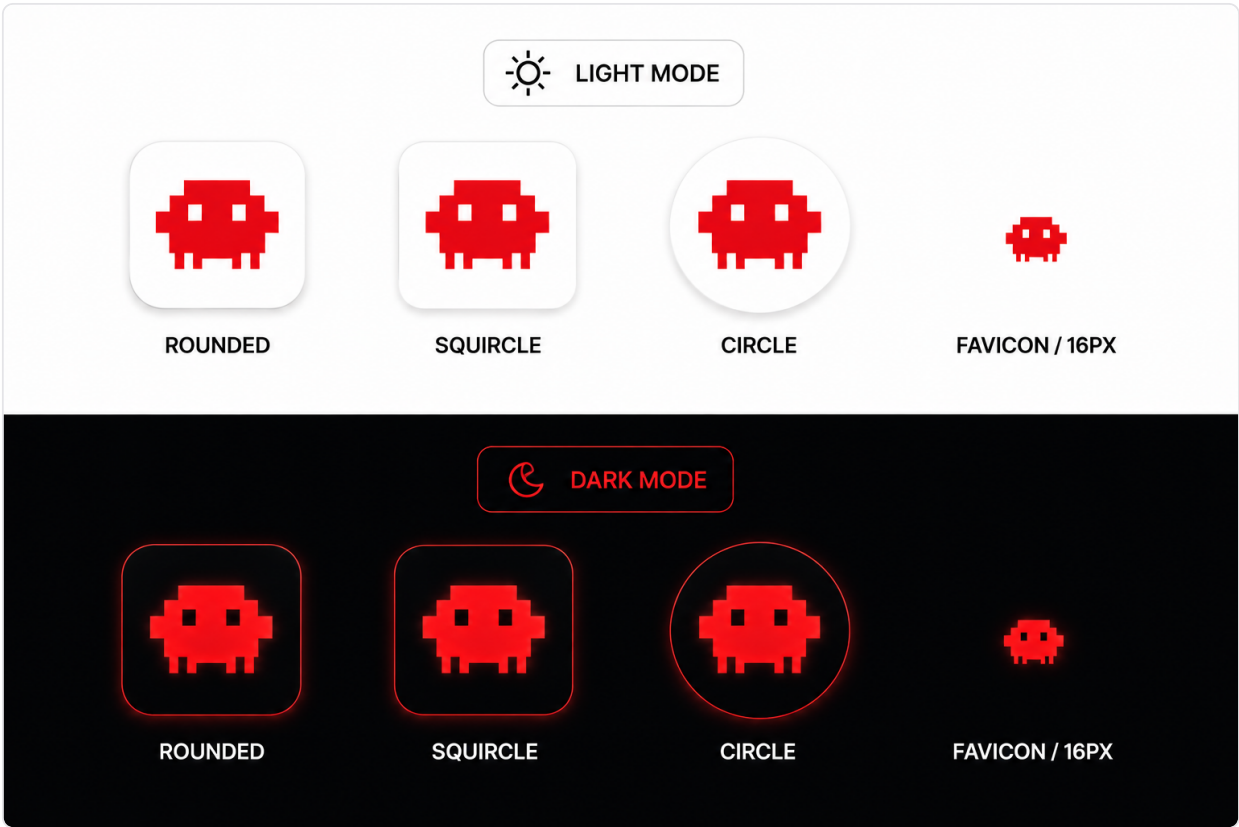
Every one of these is disabled instantly by Settings → Motion → Reduced, and by the operating system's own reduce-motion setting. Motion is never the only way information is communicated.

Icons & logo

Icons are hand-drawn line art rendered as inline SVG: 1.5px strokes, rounded caps, a monochrome outline with exactly one red detail. They inherit the theme colour, so a single definition serves both.



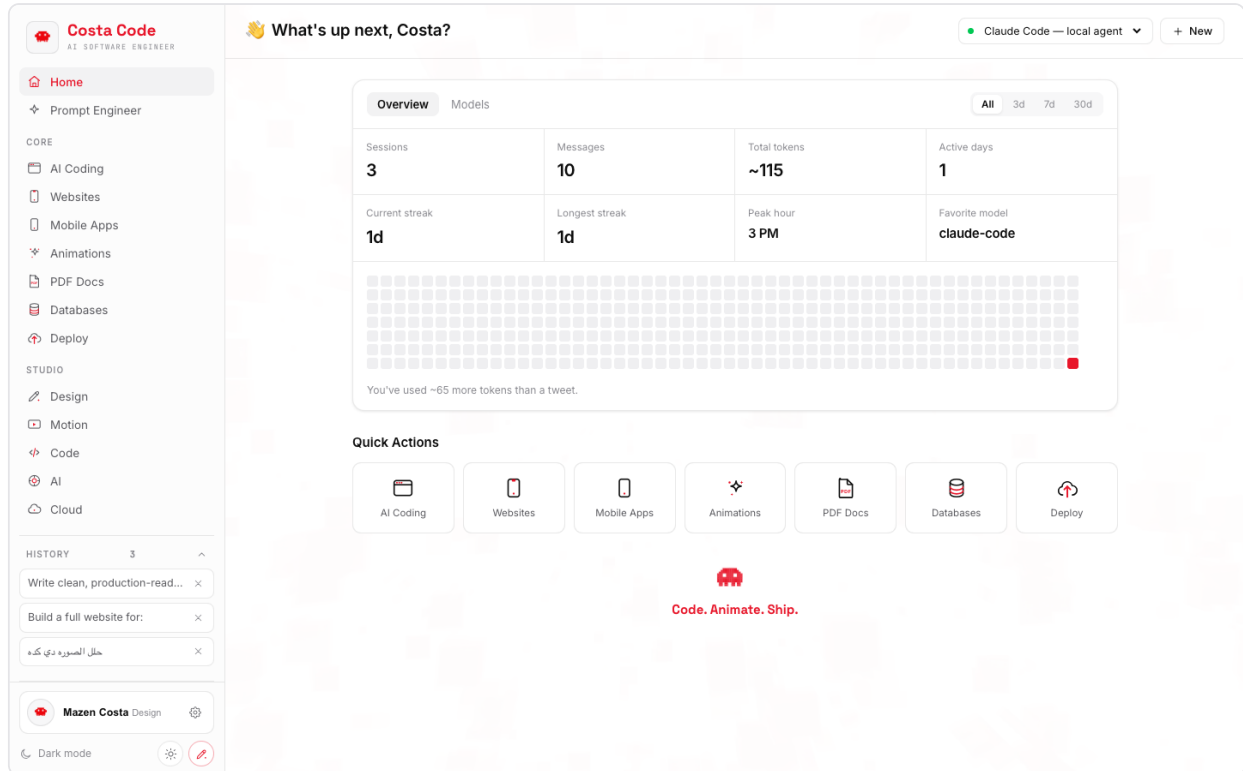
Logo framings



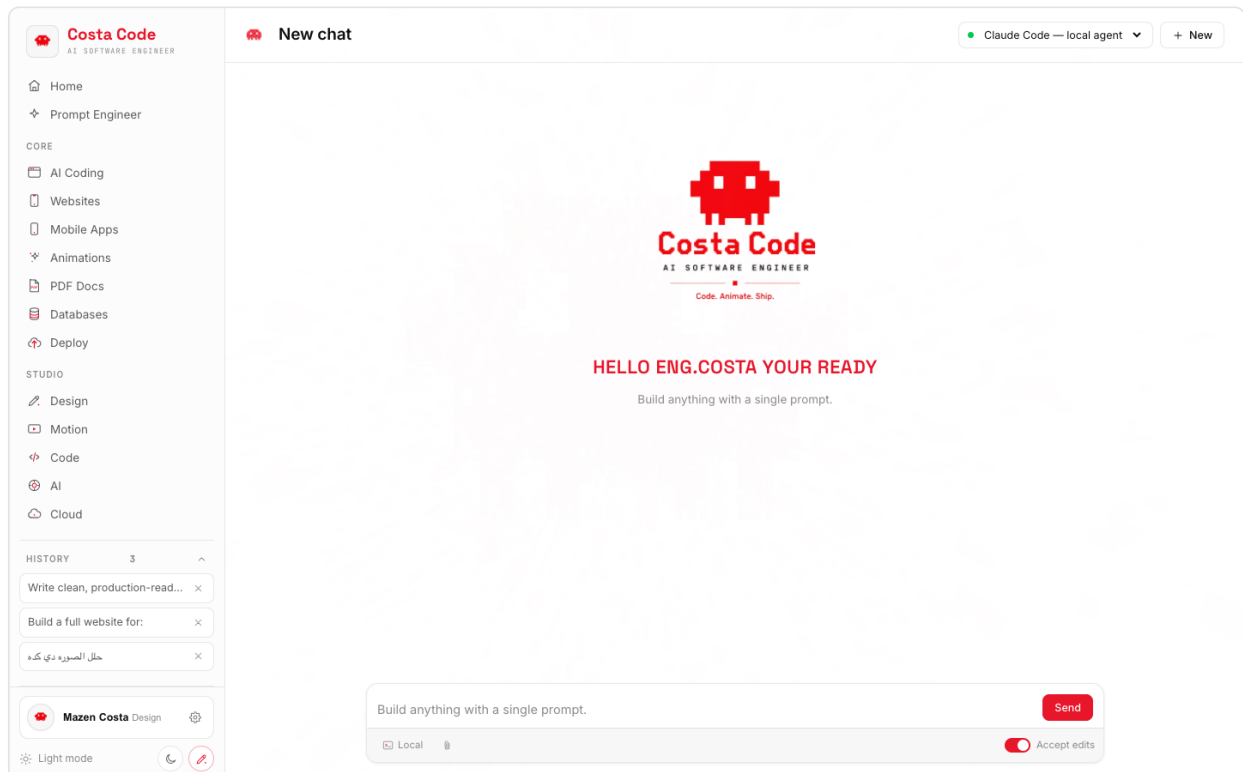
In dark mode the mark carries a soft red glow; in light mode it sits clean with a hairline border.
Same asset, two treatments.

The Website — light

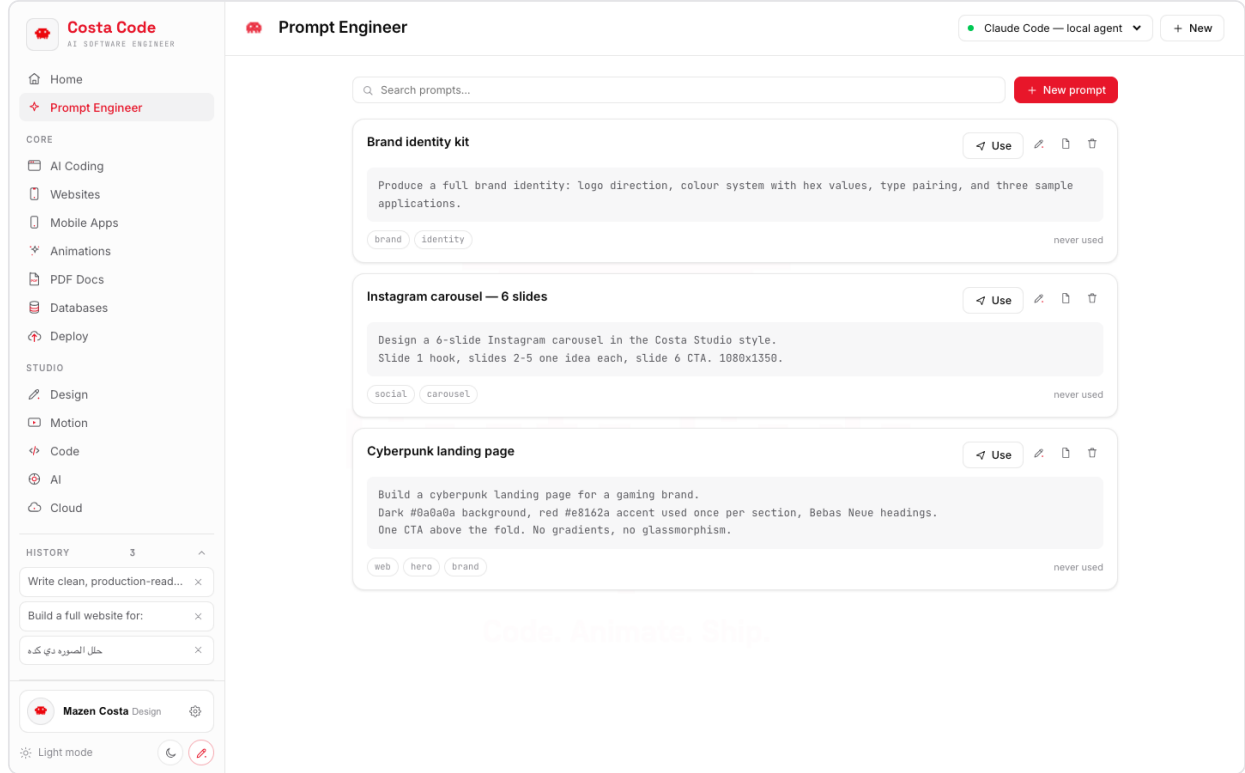
The same five screens, in both themes. Note that nothing moves position and nothing loses contrast — the layout is identical, only the surface changes.



Home — usage dashboard, activity heatmap, Quick Actions

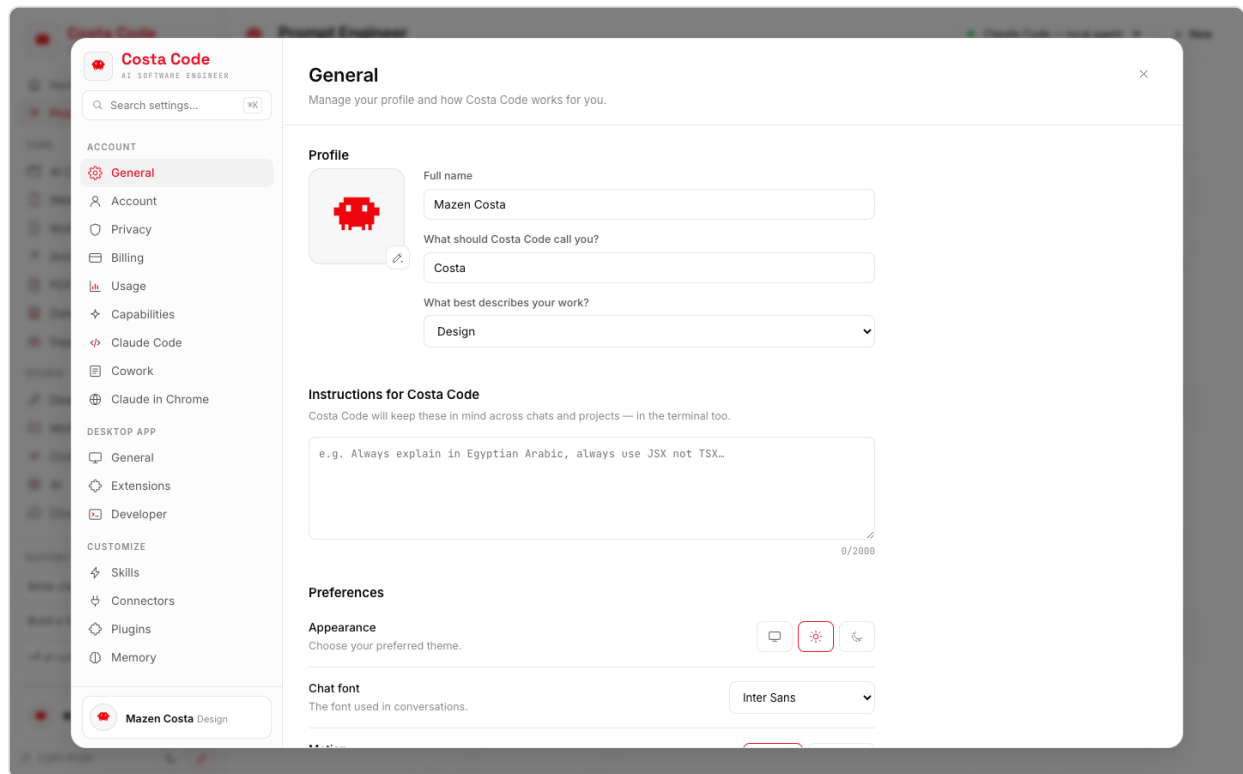


Chat — the animated logo lockup and your greeting

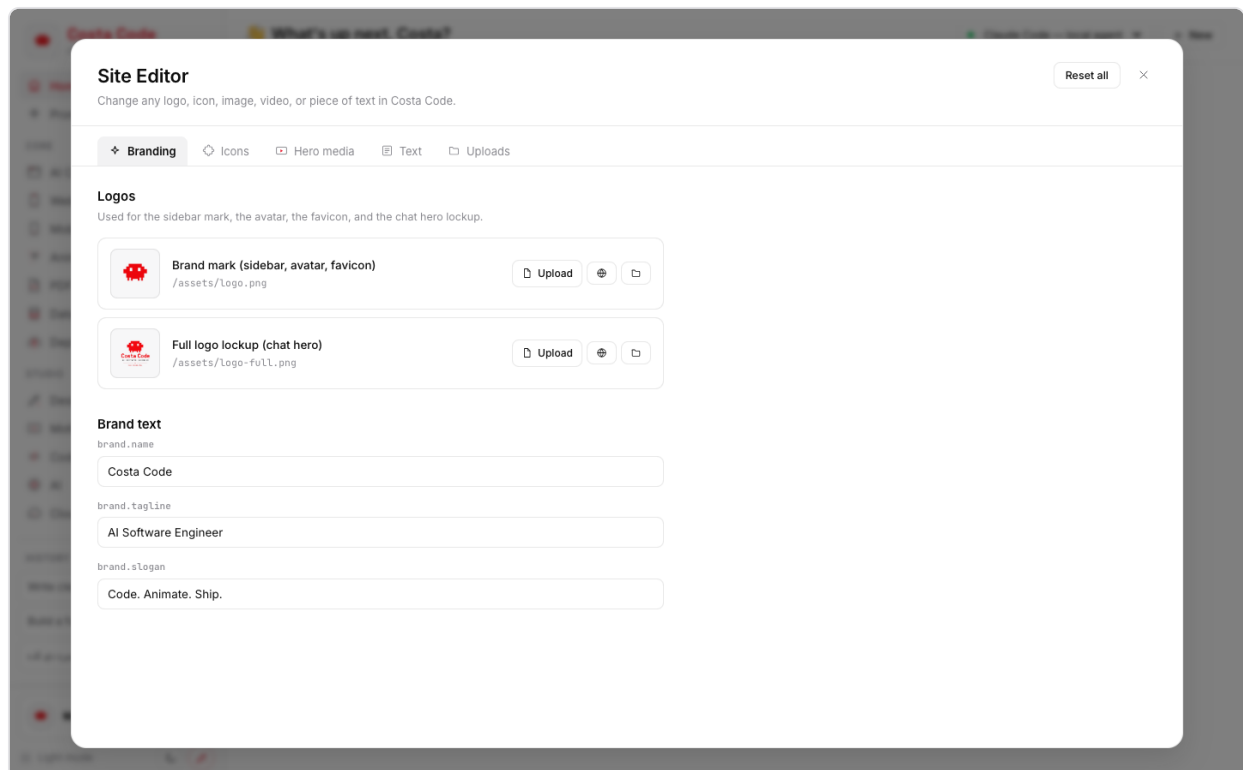


Prompt Engineer — your reusable prompt library

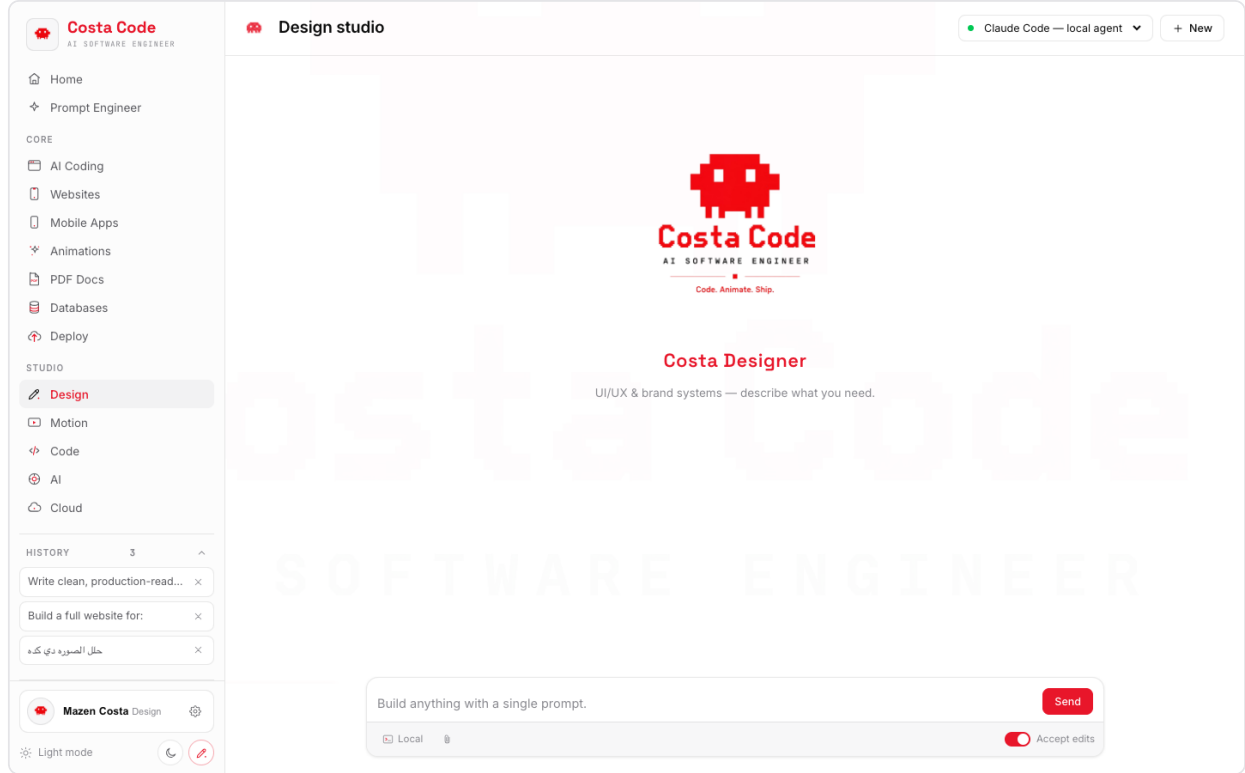
The Website — light



Settings — profile, standing instructions, appearance

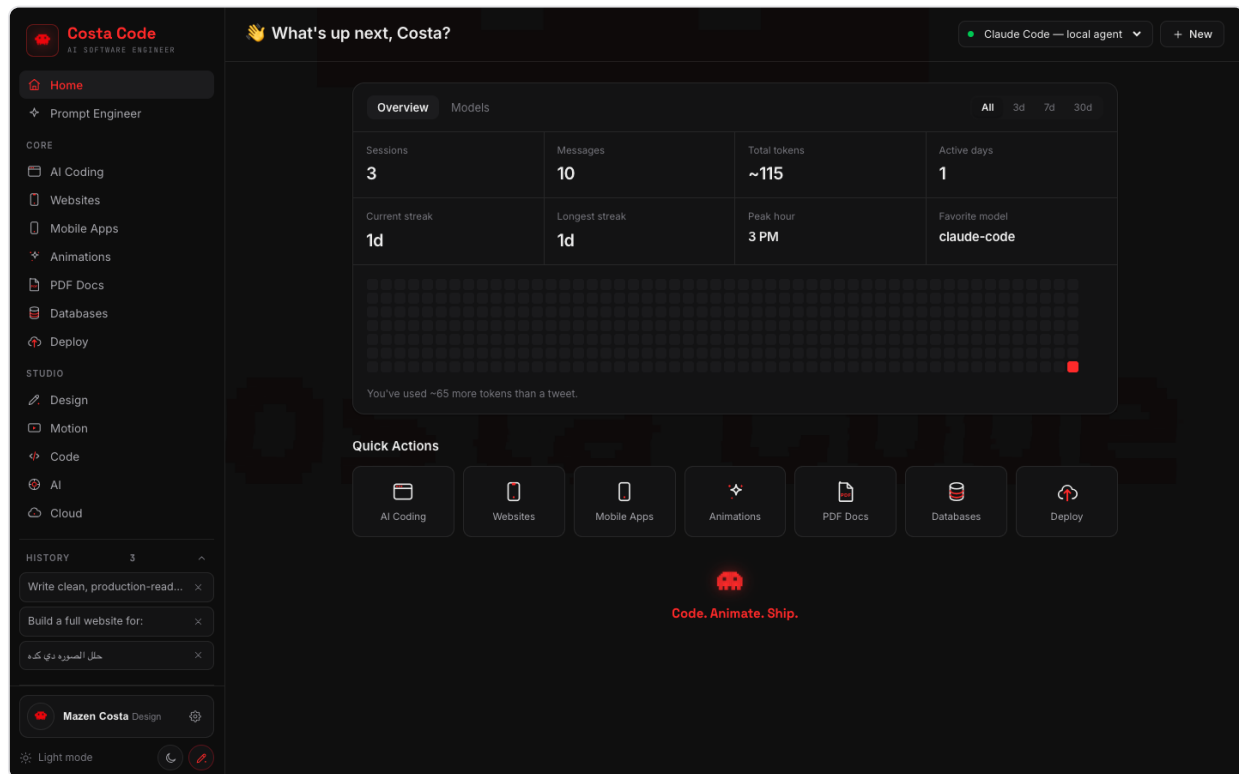


Site Editor — change any logo, icon, image or word

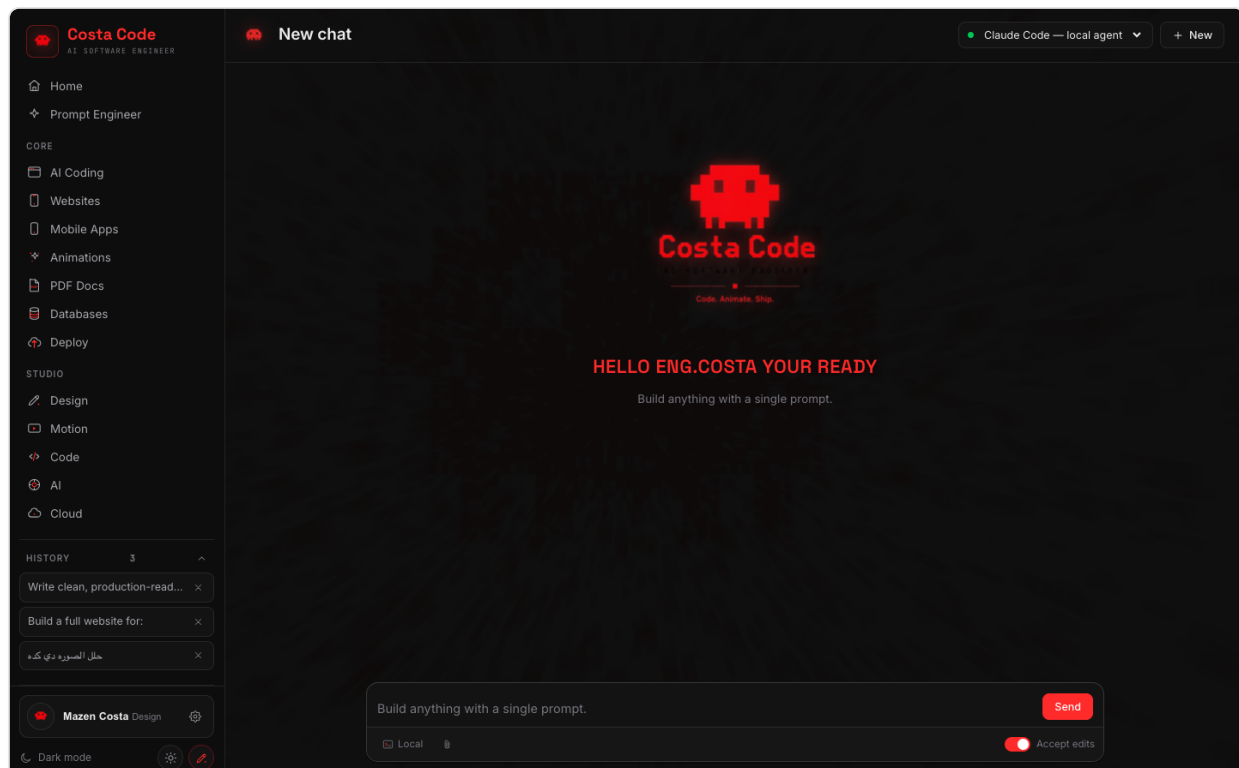


Studio — five specialist agents

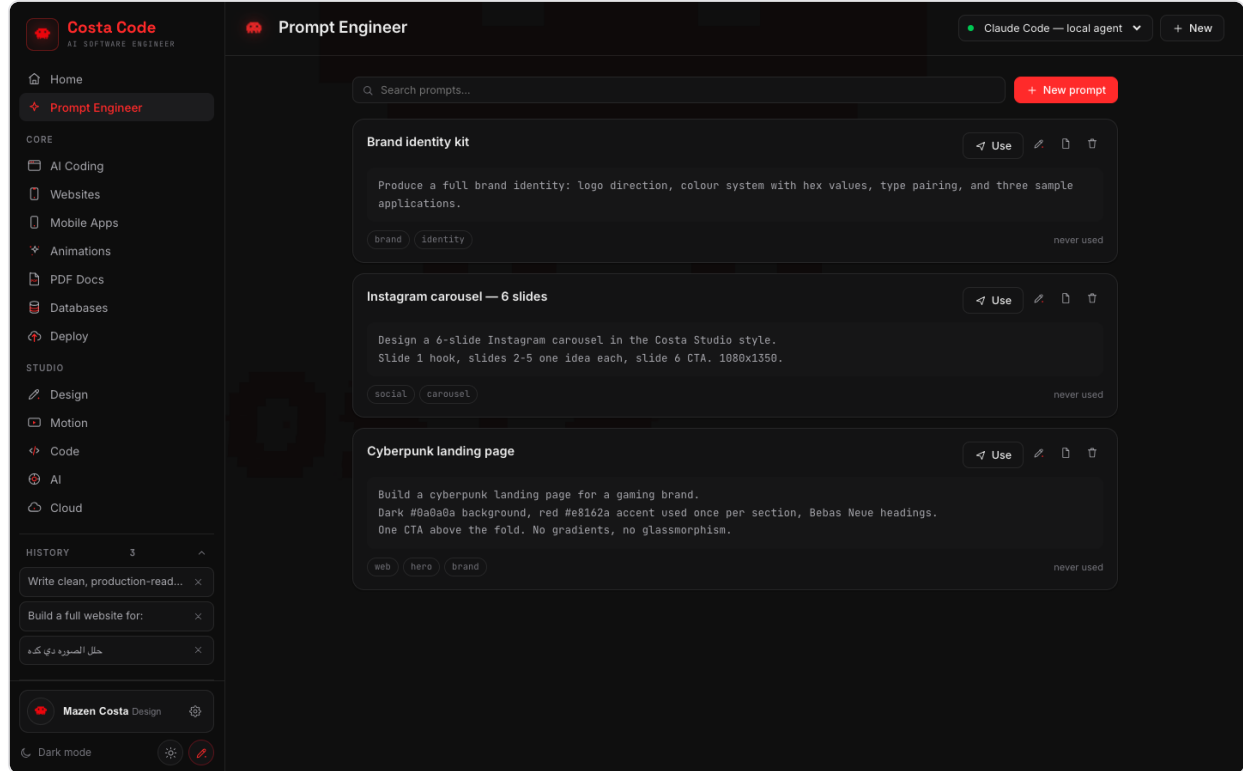
The Website — dark



Home — usage dashboard, activity heatmap, Quick Actions

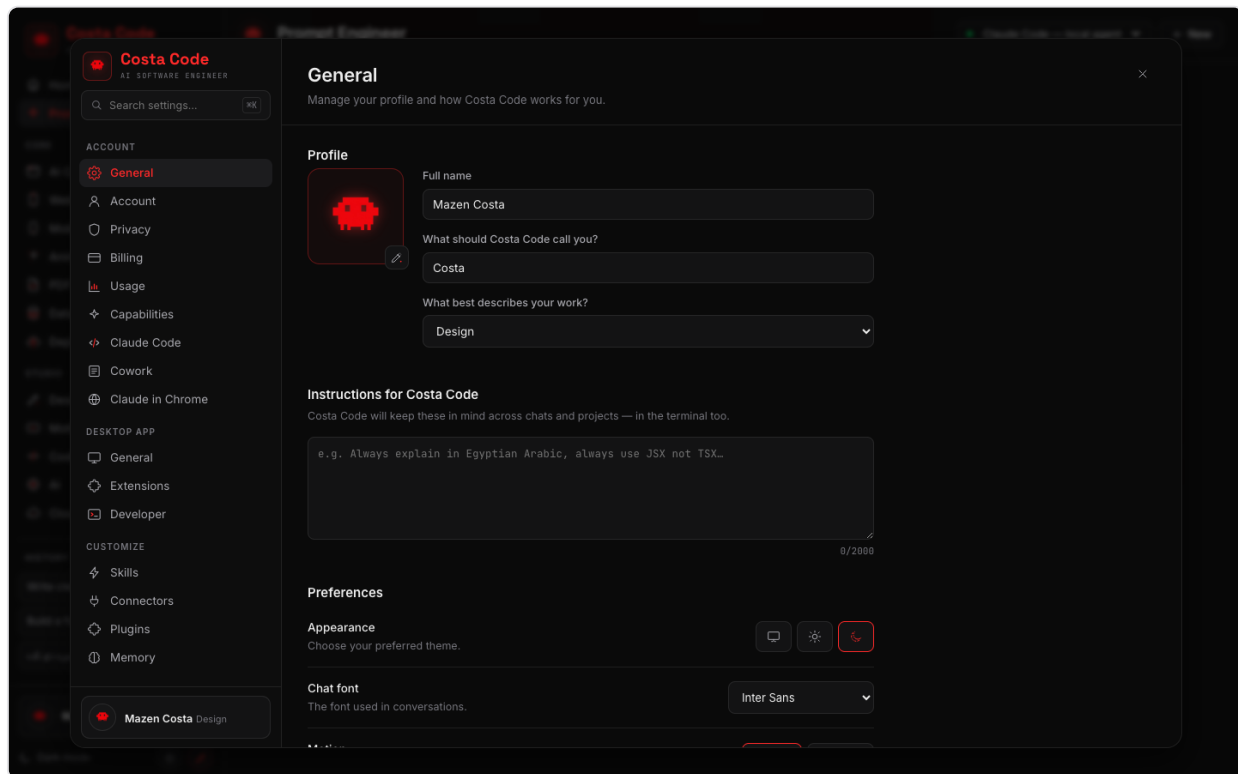


Chat — the animated logo lockup and your greeting

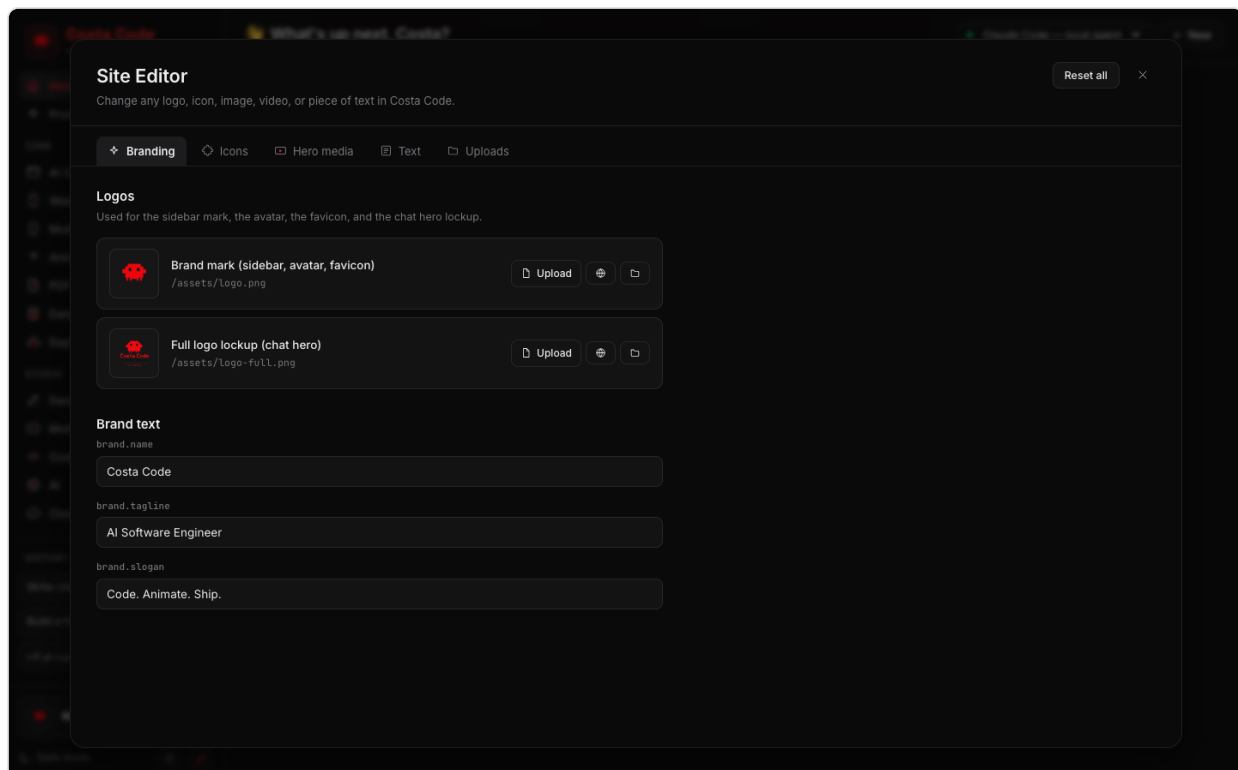


Prompt Engineer — your reusable prompt library

The Website — dark



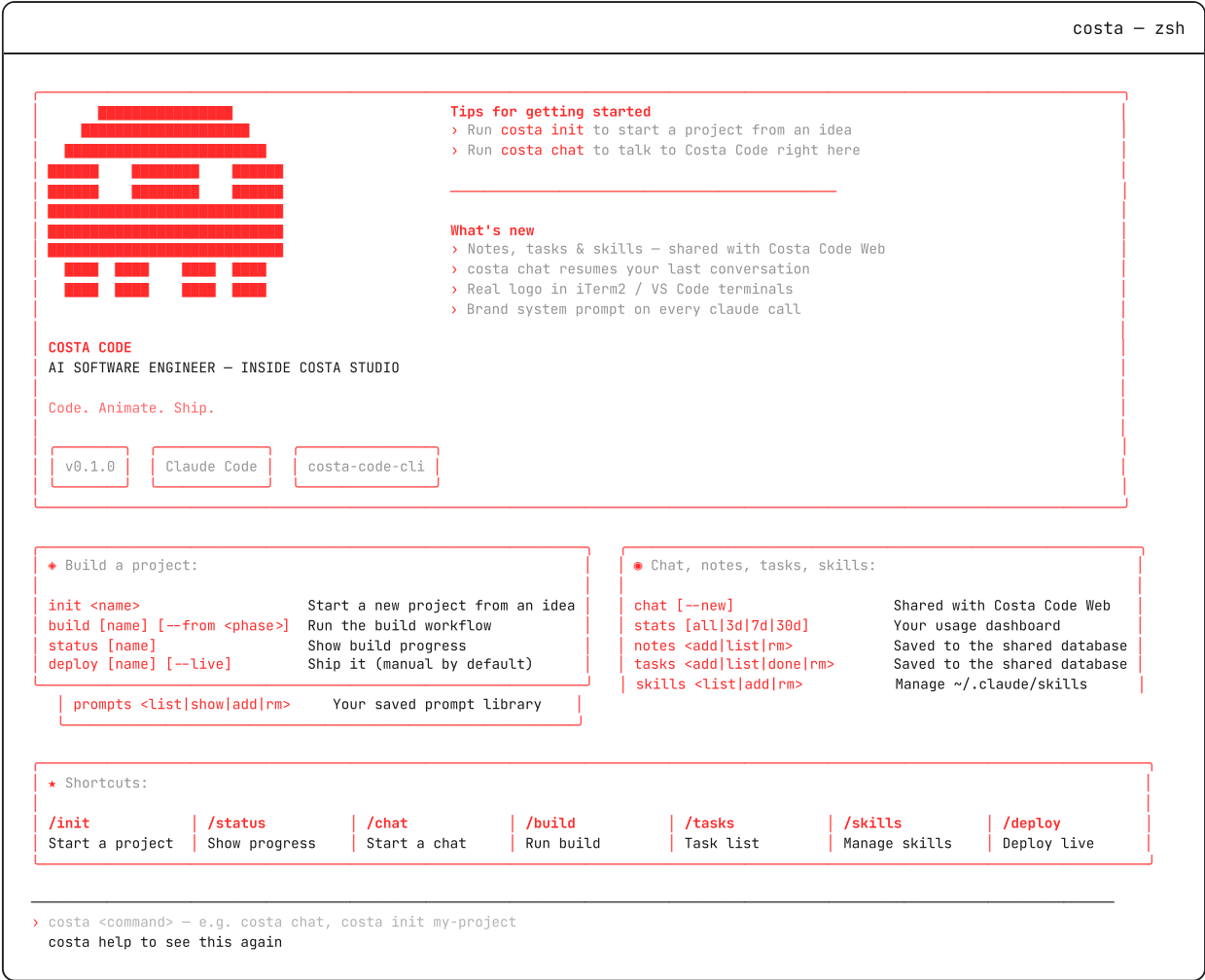
Settings — profile, standing instructions, appearance



Site Editor — change any logo, icon, image or word

The Terminal

The same brand, drawn with box characters. Red accent, mono type, the mascot rendered as a real inline image where the terminal supports it and as pixel art everywhere else.



costa help

The Terminal

```

costa — zsh

Costa Code — All time

Sessions      3      Current streak  1d
Messages     10      Longest streak  1d
Total tokens  ~115     Peak hour       3 PM
Active days   1      Favorite model  claude-code

Activity
.....

You've used ~65 more tokens than a tweet.

```

costa stats — the same numbers as the web dashboard

```

costa — zsh

Prompts (3):
#9 Brand identity kit [brand, identity]
  Produce a full brand identity: logo direction, colour system with hex values, ...
#8 Instagram carousel — 6 slides [social, carousel]
  Design a 6-slide Instagram carousel in the Costa Studio style...
#7 Cyberpunk landing page [web, hero, brand]
  Build a cyberpunk landing page for a gaming brand...

```

costa prompts list — the same library as the browser

One prompt → anything

This is the core promise, and it is worth being precise about how it actually works.

1 You write one line

Plain language. No formatting, no special syntax, no prompt engineering degree required.

2 Your brand is attached

Colours, fonts, code conventions, tone — plus whatever standing instructions you saved. Automatically, every time.

3 The right specialist runs it

Pick a Studio agent and you get a designer, an animator, an engineer, an AI architect or an infrastructure lead — not a generalist.

4 Real output lands

Text streams back, or actual files are written to disk if you used the local engine.

What "anything" actually covers

PACKAGE	HOW IT WORKS
A full website	<i>Build a one-page site for a specialty coffee shop in Cairo: hero, menu, location, one booking CTA.</i>
A mobile screen	<i>Design the onboarding screen for a fitness app — dark, bold type, one primary action.</i>
An animation	<i>Add a scroll-triggered reveal to the hero, 600ms, ease-out, no bounce.</i>
A document	<i>Turn this brand guide into a client-ready PDF with a cover and contents.</i>
A database	<i>Design the schema for a booking system with tables, staff and time slots.</i>
A deploy	<i>Give me the exact steps to put this on Netlify with a custom domain.</i>

The difference between a weak result and a strong one is almost always **specificity**. "Make it nice" gives you an average of the internet. "Dark, one red accent, Bebas headings, one CTA above the fold" gives you your brand.

A real build, start to finish

This is an actual session, captured as it ran. Four commands from idea to a project on disk.

```
costa – zsh

✓ Created cafe-landing at /private/tmp/costa-demo/cafe-landing
  Brief saved to .costa/brief.md

Next: costa build cafe-landing
```

```
costa init cafe-landing --prompt "..."
```

```
costa – zsh

cafe-landing /private/tmp/costa-demo/cafe-landing
created 2026-08-06T02:40:36.978Z

  • Analyze Idea
  • Plan Project
  • Design Interfaces
  • Build Code
  • Add Animations
  • Test
  • Generate Documentation

7 phase(s) remaining – run: costa build cafe-landing
```

```
costa status cafe-landing
```

What the eight phases do

01 Analyze Idea

02 Plan Project

03 Design Interfaces

04 Build Code

05 Add Animations

06 Test

07 Documentation

08 Deploy

Each phase is a separate agent run that hands off through real files in `.costa/` — not in-memory context. So if phase four goes wrong you can read what phase three decided, fix it, and resume from there with `--from build` instead of starting over.

Getting the most out of it

Write your instructions once, properly

Settings → Instructions is the highest-leverage box in the whole product. Put your non-negotiables there: your stack, your conventions, the language you want explanations in. It applies to every message in both tools, forever.

Build a prompt library, not a habit of retyping

Every brief you write twice belongs in Prompt Engineer with tags and reference images. Six months in, this is the asset — not the app.

Use the right Studio agent

Asking a generalist for motion direction gets you generic easing. Asking Costa Motion gets you durations, curves and triggers.

Let the local engine touch files

For real project work, use the Claude Code provider with Accept edits on. Talking about code is slower than changing it.

Keep notes and tasks where the work is

Client quirks, deploy paths, decisions — in the sidebar, shared with the terminal, not scattered across chats.

Watch the dashboard

Peak hour tells you when you actually do your best work. Streaks tell you whether this became a habit or a novelty.

Turning it into income

There are three honest routes. They are not equally easy, and the differences matter.

1 • Deliver faster

Easiest — start today

Use it as your production engine. Same services you already sell, delivered in a fraction of the time. If a brand system took five days and now takes two, you have not raised your price — you have multiplied your capacity.

Formula: **Revenue = existing rate × more projects**

2 • Sell productised packages

Medium — a few weeks

Stop selling hours. Sell fixed outcomes with fixed prices and fixed turnaround, built on repeatable prompt workflows you already have saved.

Formula: **Revenue = package price × volume**

3 • Sell access as a product

Hardest — months of work

Turn Costa Code into something other people pay to use. This is a real product build, not a repackaging. See the honest requirements two pages on.

Formula: **Revenue = subscribers × monthly fee**

The arithmetic that actually matters

A freelancer delivering four projects a month at \$450 makes \$1,800. The same freelancer delivering ten of the same projects makes \$4,500 — with no new clients, no price increase, and no new skill. Speed is the most under-priced lever in freelance work, and it is the one this tool actually moves.

Productised services you can sell

Each of these maps directly onto a saved prompt workflow. Build the prompt once; sell the outcome repeatedly.

PACKAGE	INCLUDES	TYPICAL PRICE
Brand system in 5 days	Logo direction, colour system, type pairing, motion logo, social templates, brand guide PDF	\$450 – \$900
Landing page that converts	One page, copy + design + build + deploy, mobile-first	\$300 – \$700
Social content pack	12 posts + 4 carousels, on-brand, ready to schedule	\$200 – \$450
Pitch / brand deck	12–20 slides, designed, with real content not lorem	\$250 – \$600
App screen set	6–10 screens, one flow, design-system consistent	\$400 – \$900
Rescue / redesign	Audit an existing site, then rebuild it properly	\$350 – \$800

Ranges reflect MENA freelance platforms at the lower end and direct international clients at the upper end. Price on the outcome you deliver, never on the hours the tool saved you — the client is buying the result, not your process.

Selling it as a subscription

This is the route people jump to first and underestimate most. Here is what actually stands between the current build and a product other people pay for.

What has to be built first

It is local-only by design

The current build runs a real agent with file access on the server. On a public host that is remote code execution. The local engine and the file endpoints have to come out before anything is hosted.

No accounts

There is no sign-up, no sessions, no per-user data separation. Every visitor would share one database.

No billing

No payments, no plans, no usage limits, no way to cut someone off when they stop paying.

No cost control

Model calls cost money per request. Without per-user metering and caps, a handful of heavy users can cost more than everyone pays.

Support becomes your job

The moment someone pays monthly, uptime, bugs and questions are your responsibility, not a side project.

A realistic path if you want it

1. Strip the local engine and file access — keep only the API providers.
2. Add accounts and per-user data (auth + a real user column on every table).
3. Add a credit ledger: every generation deducts, every plan tops up.
4. Add payments — Stripe internationally, Paymob for Egypt, Tap for the Gulf.
5. Only then: pick pricing, and expect the first ten paying users to be people you already know.

The lower-risk version of the same idea

Sell the **output**, not the access. A prompt library, a brand system template, a set of workflows — packaged and sold once, with no servers to run, no uptime to owe anyone, and no per-request cost. Most of the upside, almost none of the liability.

Pricing & packaging

Three principles that survive contact with real clients.

1 **Price the outcome, not the input**

Nobody pays for "eight hours". They pay for a brand that makes them look serious. The faster you get there, the higher your effective rate — as long as you never bill by the hour.

2 **Always offer three tiers**

Most people pick the middle. Without a top tier there is no middle. The expensive option exists to make the middle one feel reasonable.

3 **Charge for speed separately**

Standard turnaround is included. Rush is a paid add-on, typically +40–60%. You now have the capacity to actually deliver rush work, which most freelancers cannot.

A tier structure that works

Essential

One deliverable, one revision, standard turnaround

Anchors the low end

Complete

The full package, two revisions, source files

Where most clients land

Studio

Everything, priority turnaround, ongoing support window

Makes Complete look sensible

The 30-day plan

Not a growth-hack list. Just the order that actually compounds.

Week 1 **Set it up properly**

Write your standing instructions. Save your five most-used briefs into Prompt Engineer with reference images. Run one real client job through it end to end and time it honestly.

Week 2 **Turn speed into a package**

Take the job you just did and write it up as a fixed-price, fixed-turnaround package. Three tiers. Publish it on the platforms you already use.

Week 3 **Sell it twice**

Offer it to two past clients before any stranger. They already trust you, and their feedback tells you what the packaging gets wrong.

Week 4 **Measure and cut**

Check the dashboard. Which package sold? Which prompt did you actually reuse? Delete what you did not touch and deepen what you did.



The tool is not the business. The tool is what lets you run the business at a speed nobody can match by hand.