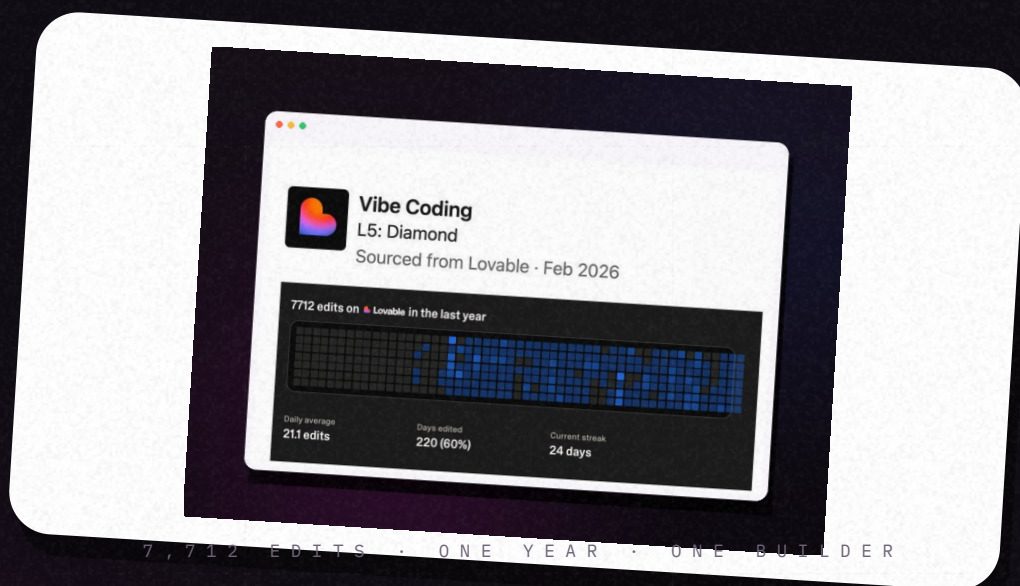


Prompt to Production.

Field notes from 22 shipped Lovable builds

How to prompt like a designer, protect your credits like an operator, and walk out with a product you actually own.



7,712 EDITS · ONE YEAR · ONE BUILDER

KZ

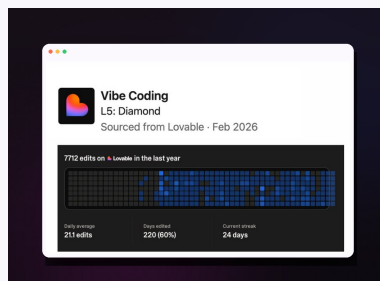
Kristian Zwart

LOVABLE VIBE CODING L5 DIAMOND · 7,712 EDITS / 12 MONTHS

I didn't read about Lovable. I *lived in it*.

Hi – I'm Kris. Twenty years a designer, and for the last year one of Lovable's most active builders: 7,712 edits, 220 days, 22+ shipped products – from a grief-tech app AWS put in their Activate portfolio, to a CRM valued at £1.5M, to a voice assistant that raised £90k.

That volume teaches you things the launch videos don't: where Lovable is genuinely magic, exactly where it breaks, and the discipline that separates a demo from a product. These are those field notes – twelve of them, plus the prompts I actually reuse. Everything below has shipped.



THE RECEIPTS – SOURCED FROM LOVABLE, FEB 2026

INSIDE THIS EDITION

03–04

Mindset & setup

Prompt like a designer · the knowledge file

05–11

The 12 field notes

Direct → Protect → Production

12–14

The toolkit

6 reusable prompts · failure modes · proof

Prompt like a designer, *not a user.*

The gap between a mediocre Lovable build and a beautiful one isn't credits or luck – it's the brief. Users describe widgets. Designers describe intent. Three shifts:

1

Mood before mechanics

Open every project by defining how it should feel, not what it should contain. "Warm, unhurried, confident – a private members' club, not a SaaS dashboard" shapes a thousand small decisions no feature list can reach.

2

Moments, not pages

Describe what happens: "when the quote is accepted, the summary card settles into place and the total gently counts up." You're briefing motion, sequence and emotion in one sentence.

3

Name your references

Anchor taste to things that exist: a product ("Linear's restraint"), a system ("shadcn defaults"), an era ("Swiss poster, 1968"). Shared references collapse ten rounds of "not quite" into one.

“

The best Lovable builders sound like art directors, not ticket queues.

Win the build before *the first prompt*.

Every project starts with a knowledge file – the standing brief Lovable reads on every single edit. Ten minutes here saves hundreds of corrective prompts later. Mine always contains:

```
PROJECT          what it is, for whom, in one sentence
TONE             three feeling words + one reference product
TOKENS          exact hex colours, both fonts, radius, spacing
LAYOUT LAW      one idea per view · generous whitespace
COPY VOICE      how buttons, errors and empty states speak
HARD RULES      never change: nav, tokens, auth flow
STACK           backend, auth, payments – named up front
```

- PASTE INTO PROJECT SETTINGS → KNOWLEDGE. UPDATE AS YOU DECIDE.

zwartifydesign.

The HARD RULES line matters most. Lovable is an enthusiastic collaborator – it will happily "improve" things you considered finished. Rules in the knowledge file are cheaper than rules repeated in forty prompts.

IN PRACTICE

BRIEF

"Make a nice modern landing page for my startup"



✓ ART DIRECTION

"Calm, premium landing for busy founders
– Linear-like restraint, our tokens,
hero + 3 proof cards only."

12 field notes. *Three phases.*

The arc every successful build follows. Direct sets taste and intent. Protect keeps the build (and your credits) intact. Production turns a Lovable project into a product you own.

PHASE 1 · DIRECT

Notes 1–4

Prompts that carry taste, not just tasks

PHASE 2 · PROTECT

Notes 5–8

Workflow discipline – versions, batches, guardrails

PHASE 3 · PRODUCTION

Notes 9–12

Own the stack: backend, auth, payments, exit

Most people learn Phase 1 from posts. Almost nobody talks about Phase 3 – which is exactly why founders end up locked in.

Taste is a *prompt*.

FIELD NOTE 01

Open with the vibe contract

First prompt of every build: the feeling, the reference, the audience, the one thing this product must never feel like. Everything Lovable makes afterwards is negotiating with this paragraph.

TRY: "Calm and premium, like Linear meets a boutique hotel. Never busy."

FIELD NOTE 02

One scene per prompt

Big multi-feature prompts produce mush. Build the way you'd storyboard: one screen, one moment, one prompt. Ten small briefs beat one epic – in quality and in credits.

TRY: "Build only the empty state of the inbox. Nothing else yet."

IN PRACTICE

EVERYTHING

"Homepage, pricing, about, blog, dashboard – go"



✓ ONE SCENE

"Just the hero. Headline, sub, one CTA, our tokens. Stop there."

Brief the *behaviour*.

FIELD NOTE 03

Prompt verbs, not adjectives

"Make it nicer" is a coin flip. "Make it feel lighter – more air, thinner borders, calmer shadows" is a direction. When a screen is wrong, describe the feeling gap, not the CSS.

TRY: "This feels heavy. More whitespace, softer elevation, quieter chrome."

FIELD NOTE 04

Give the AI a spine

Tell Lovable to act as your design partner and to push back when something conflicts with the knowledge file. It genuinely will – and it catches your own inconsistencies.

TRY: "You are my senior design partner. Challenge briefs that break our rules."

“*A prompt is a creative brief. Write it like the budget depends on it – because in credits, it does.*”

 **zwartifydesign.**

Credits are a *workflow*.

FIELD NOTE 05

Skeleton first, beauty second

Generate structure and navigation with placeholder content, test every route, pin that version – then design. Styling a broken skeleton is how people burn a month of credits in a weekend.

TRY: "Placeholders only. No styling yet. I want to test the flow first."

FIELD NOTE 06

Pin before you experiment

Before any risky prompt – refactors, new libraries, layout rethinks – pin the version. Lovable's history is your free undo; treat pins like a climber treats anchors.

IN PRACTICE

YOLO

"Redesign everything, add auth + payments, make it pop"



✓ FENCED

"Pin current. Then: restyle ONLY /pricing to match the hero. Nothing else."

Change one thing *at a time*.

FIELD NOTE 07

The guardrail prompt

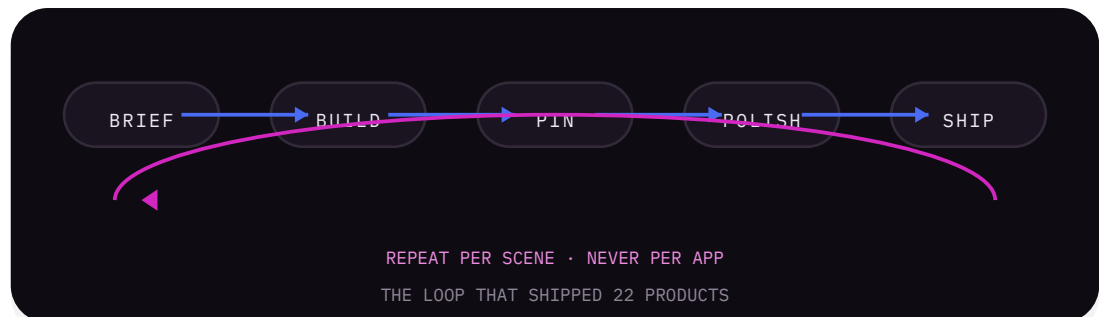
When editing anything specific, fence the blast radius explicitly. This single habit ends the "fixed one screen, broke three" cycle that makes people quit Lovable.

TRY: "Change ONLY the pricing card. Touch no other file, style or route."

FIELD NOTE 08

Polish by hand, not by prompt

Spacing, font sizes, copy tweaks – use Visual Edit and direct edits. Prompts are for decisions; pixels are cheaper by hand. My rule: if the change is under 8px, it isn't a prompt.



Own your *stack*.

FIELD NOTE 09

Choose your backend on day one

Lovable will happily wire Supabase – great. But decide deliberately: Supabase, Convex or custom, named in the knowledge file. Products that "grew whatever backend happened" are the ones that get stuck.

TRY: "All data through a clean API layer – I may swap the backend later."

FIELD NOTE 10

Auth and payments are products

Don't bolt them on last. Brief login, empty states, failed payments and cancellations with the same design care as the homepage – this is where "demo" and "product" part ways.

IN PRACTICE

VAGUE

"Add Supabase or whatever works"



✓ DELIBERATE

"Supabase, named in knowledge. Every query behind /lib/api – swappable later."

Build your *exit*.

FIELD NOTE 11

Custom domain, week one

A real domain changes how everyone treats the build – users trust it, you finish it, investors take it seriously. It's a \$12 psychological upgrade.

FIELD NOTE 12

Know how you'd leave

Export the repo, read it, make sure it runs outside Lovable. Not because you will leave – because a product you could take anywhere is a product you actually own.

TRY: "Audit the codebase for anything that only works inside this platform."

“*Rented velocity feels wonderful – right up until the day you need to move.*”

 **zwartifydesign.**

Six prompts I reuse *every build.*

PROJECT OPENER

Act as my senior design partner. Read the knowledge file. Confirm the vibe in one sentence before building anything.

SCENE BRIEF

Build [screen] for [user] in [moment]. One idea, generous whitespace, our tokens. Describe your plan first in 3 bullets.

GUARDRAIL EDIT

Change only [component]. No other files, styles, routes or copy. If that's impossible, stop and tell me why.

FEELING FIX

[Screen] feels [wrong feeling]. Make it feel [right feeling] – adjust space, weight and motion, not the layout.

BUG SNIPER

Reproduce: [steps]. Expected [X], got [Y]. Fix the cause, not the symptom, and tell me what was wrong in one line.

SHIP AUDIT

Review the whole app as a ruthless QA lead: broken states, overflow, contrast, dead ends. List issues by severity – don't fix yet.

COPY THEM. ADAPT THEM. THEY'RE YOURS NOW.

Where Lovable breaks *(and the fix)*.

The refactor spiral

It rewrites a large file, breaks two things, "fixes" them by breaking a third. → Roll back to your pin, split the change into three guardrailed prompts.

Style drift

Twenty edits in, buttons have three radii and four accents. → Tokens in the knowledge file + a monthly "audit against tokens, list violations" prompt.

Ghost regressions

A screen you finished last week quietly changed. → You edited without a guardrail. Pins to compare, guardrails to prevent. Every time.

The 90% wall

Demo-perfect, production-fragile: edge cases, empty states, errors. → Run the SHIP AUDIT prompt, then fix by severity. This is the real last mile.

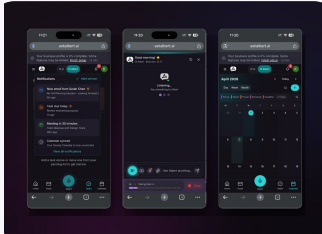
Credit evaporation

A hard bug eats thirty prompts. → Stop at three failed attempts. Reproduce precisely, use BUG SNIPER, or fix that one by hand in the editor.

The context ceiling

Very long chats get vaguer and slower. → Start a fresh chat per feature; the knowledge file carries the state so nothing is lost.

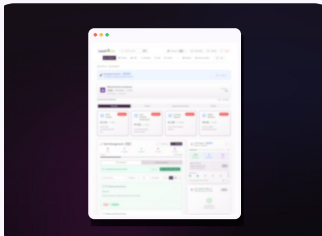
Same tool. *Different discipline.*



ASK ALBERT · AI ASSISTANT

£90k pre-seed · ~£2.5M valuation

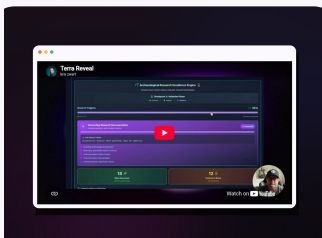
Voice-and-chat cognitive assistant, Lovable-built and investor-funded – with this exact workflow.



LEADNOMIC · SAAS CRM

£1.5M valuation · fully Lovable front-end

Proof AI-native development reaches production quality – early backers, private beta.



TERRAREVEAL · AI RESEARCH

5th of 10,000+ · OpenAI to Z

Full pipeline and product UX in Lovable – top 0.05% worldwide vs PhD-led teams, solo.

“None of these needed a bigger team. They needed twelve notes, applied without exceptions.”



Want it built for you?

I run this exact workflow as a service: the MVP Sprint – your AI product designed, built and deployed in 14 days for \$8,900 fixed. Lovable-speed, without Lovable-lock: your backend stays portable (Convex ⇌ Supabase ⇌ custom) and you own everything from day one.

Start a 2-week MVP Sprint

zwartify.com

KRIS@ZWARTIFYDESIGN.COM · KUALA LUMPUR · WORLDWIDE

*Or take the notes and outbuild everyone. Either way –
ship something real this week.*



ALL 3 BOOKS



AI PRODUCTS DESIGNED, BUILT & SHIPPED - BY ONE PERSON



- THE ZWARTIFY PLAYBOOK SERIES -

01 GET FOUND BY AI · 02 PROMPT TO PRODUCTION · 03 DESIGNING FOR
MORE FREE EDITIONS AT [ZWARTIFY.COM/#RESOURCES](https://zwartify.com/#resources)

© 2026 ZWARTIFYDESIGN SDN BHD · FREE TO SHARE, NOT TO