or easily reverse-engineered

The secret to building apps that can't be copied.

Why code alone doesn't protect a software business anymore, my list of boring industries nobody is building for yet and the two prompts to find your gap in one.

AI can rebuild any screen. It can't rebuild the rules.

Fadé Adeniyi

THE PREMISE

If Adobe can be rebuilt, what's safe?

In October 2026 one developer, Brandon Thomas, rebuilt seven Adobe apps and put them on GitHub for free. **Notebookcheck reported** he used Claude Opus 5.5 as his coding assistant. There's PhotoCraft for Photoshop, VectorCraft for Illustrator, FilmCraft for Premiere, LightCraft for Lightroom, a PDF app for Acrobat, EffectCraft for After Effects and DesignCraft for InDesign.

They're early and not every one is ready for daily work. But **the PhotoCraft repo** had more than 74,000 stars on GitHub when we checked on 29 October 2026. People clearly want them.

So if the screens of a giant like Adobe can be rebuilt by one person with AI, the screens of your app can too. Code on its own isn't a moat anymore.

> **THE TRAP**
>
> Building another planner, to-do app or developer tool feels safe because you understand it. It's also the first thing everyone else builds with AI, so you're competing with thousands of copies on day one.

THE MOAT

What AI can't easily copy.

AI can rebuild any screen. What it isn't aware of yet are the rules and the history inside each industry. That's where the hard, valuable work lives.

1. **Regulation.** Firms have to prove they follow the rules, not just say it. The FCA's **Consumer Duty**, for example, "sets high standards of protection for retail consumers across financial services". Software that helps a firm prove it is worth paying for.

2. **Old data.** Decades of records sit in legacy databases. Moving them safely into something new is slow, careful work a weekend clone can't fake.

3. **Integrations.** Credit checks, identity checks, banks and government systems. Everything pulls into them and each one needs access, approval and testing.

4. **Trust.** In a regulated industry, buyers pick the supplier they trust to get it right. That takes time to earn and it can't be copied from GitHub.

> **HOW I USE IT**
>
> I've spent 16 years building software for mortgage advisers, payments companies, property firms and compliance teams. None of them were exciting apps. The hard part was never the screens. It was the regulations, the old databases I had to migrate and the credit checks everything pulled into. That's exactly why those products are hard to copy.

THE LIST

Boring industries nobody is building for yet.

Here's my list. Each one answers to real rules, runs on old systems and still does too much by hand. Pick one you know or one you can get close to.

1. **Mortgage advisers.** Answer to the FCA and the Consumer Duty. Gap: keeping every client's file ready to show why the advice was right for them.
2. **Payments.** Card data falls under PCI DSS. Gap: simple tools that help small merchants handle chargebacks and keep their evidence in one place.
3. **Insurance brokers.** FCA rules on fair value. Gap: renewal reviews that show each customer is still getting a fair deal.
4. **Compliance teams.** The Money Laundering Regulations 2017 mean ongoing customer checks. Gap: monitoring that small regulated firms can afford.
5. **Care homes.** Inspected by the CQC. Gap: turning daily care notes into the evidence inspectors ask for.
6. **Lettings agents.** Right to Rent checks and a stack of safety certificates. Gap: one tracker that warns before anything lapses.
7. **Nurseries.** Ofsted and the early years framework. Gap: paperwork that eats the time staff should spend with children.
8. **Small accountants.** Making Tax Digital for Income Tax is rolling out to sole traders and landlords. Gap: helping their clients' records keep up.
9. **Infrastructure for businesses.** Buildings, energy and telecoms answer to the Building Safety Act 2022, Ofgem and Ofcom. Gap: keeping asset, maintenance and safety records ready for whoever asks.

Two prompts to go from industry to idea.

1. Find the gap. Paste this into Claude with one industry from the list. It makes Claude show its sources, so you can check the rules yourself.

```
I want to find a software gap in this industry:
[THE INDUSTRY, E.G. LETTINGS AGENTS IN THE UK]

Act as my research partner. Tell me:

THE RULES: which regulations and regulators this
industry answers to and what they make firms prove.
THE PAPERWORK: the jobs people still do by hand,
in spreadsheets or in old systems to stay compliant.
THE OLD SYSTEMS: the legacy software and data they're
stuck with and what it connects to.
THE PAIN: the three jobs that cost them the most time
or the most risk if they get them wrong.

Then give me three small software ideas that solve one
of those jobs. For each, say who pays, what rule it
helps them meet and what would make it hard to copy.

Link your sources. If you're not sure a rule applies,
say so. Don't guess.
```

2. Stress-test the idea. Paste this in the same chat once you've picked one of its ideas.

```
Here's my idea: [YOUR IDEA IN ONE LINE]
for [THE INDUSTRY].

Be my toughest critic. Could a developer with AI rebuild
the screens in a weekend? If yes, what would they still
be missing: the rules, the data, the integrations, the
trust or the approvals? List the five questions I should
ask ten people in this industry before I build anything.
```

Then talk to real people. Ten conversations with people in that industry will teach you more than any prompt. The rules they complain about are your gap.

THE TAKEAWAY

Pick one industry today.

You've got the list and two prompts. Do one thing today. Pick the industry you know best, paste the first prompt and read what Claude finds.

What this won't do: it won't make a regulated business easy. These industries are slow to buy, they'll ask hard questions and some ideas need approvals or legal advice before you launch. Claude can also get a rule wrong, so check every source it gives you. That slow, careful work is exactly what makes your app hard to copy.

AI can rebuild any screen. It can't rebuild the rules.

Sources, opened 29 October 2026: Notebookcheck, **Experienced Rust developer recreates Adobe suite with AI assistance as open source** (8 October 2026); **storytold/photocraft on GitHub**; FCA, **Consumer Duty**. Rules change, so check the regulator's own site before you build.

Building something of your own? **The Audacity to Build** is my community for people building alongside me, with the systems I actually run. It is opening soon. Join the waitlist at fadeadeniyi.com/waitlist.

Fadé Adeniyi

Fadé & The Machines