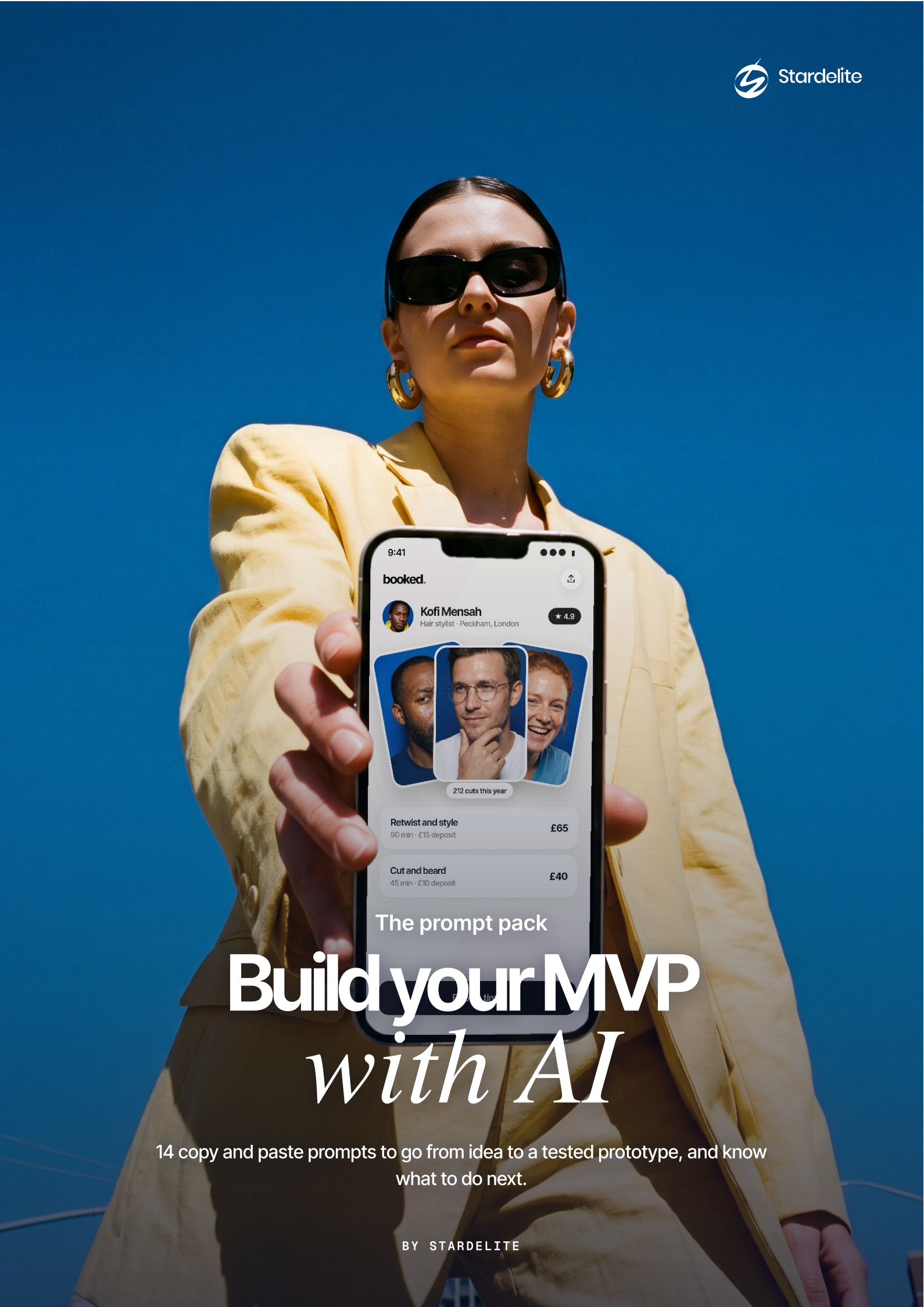
Stardelite



The prompt pack

Build your MVP *with AI*

14 copy and paste prompts to go from idea to a tested prototype, and know what to do next.

BY STARDELITE

How to use this

- Open Claude (or any chat AI you like) and start a new chat for each step.
- Copy a prompt, replace everything in [square brackets] with your own details, then send.
- Go in order. Each step uses what the last one gave you.
- Every prompt comes with a real example. We used these exact prompts to build **booked.**, a booking app for independent hair stylists, for our carousel.

WHAT'S INSIDE

1 **Shape your idea**
Brief and riskiest assumptions

2 **Build a prototype**
Build spec, design, demo data

3 **Test with people**
Testers, script, feedback

4 **Check what's missing**
Security, payments, data, scale

5 **Build it properly**
Handoff, hiring, first 90 days

Step 1: Shape your idea

Get clear on what you are building and who it is for, before you build anything.

PROMPT 1 The one page product brief

```
I want to build [your product in one sentence]. It's for [who it's for]. The main thing it lets them do is [the core job]. Turn this into a one page product brief with these sections: the problem, who it's for (primary and secondary users), the v1 features (only what's needed to solve the core problem), what's out of scope for now, and the main user flow from first visit to the moment they get value. Ask me up to 3 questions if anything important is unclear.
```

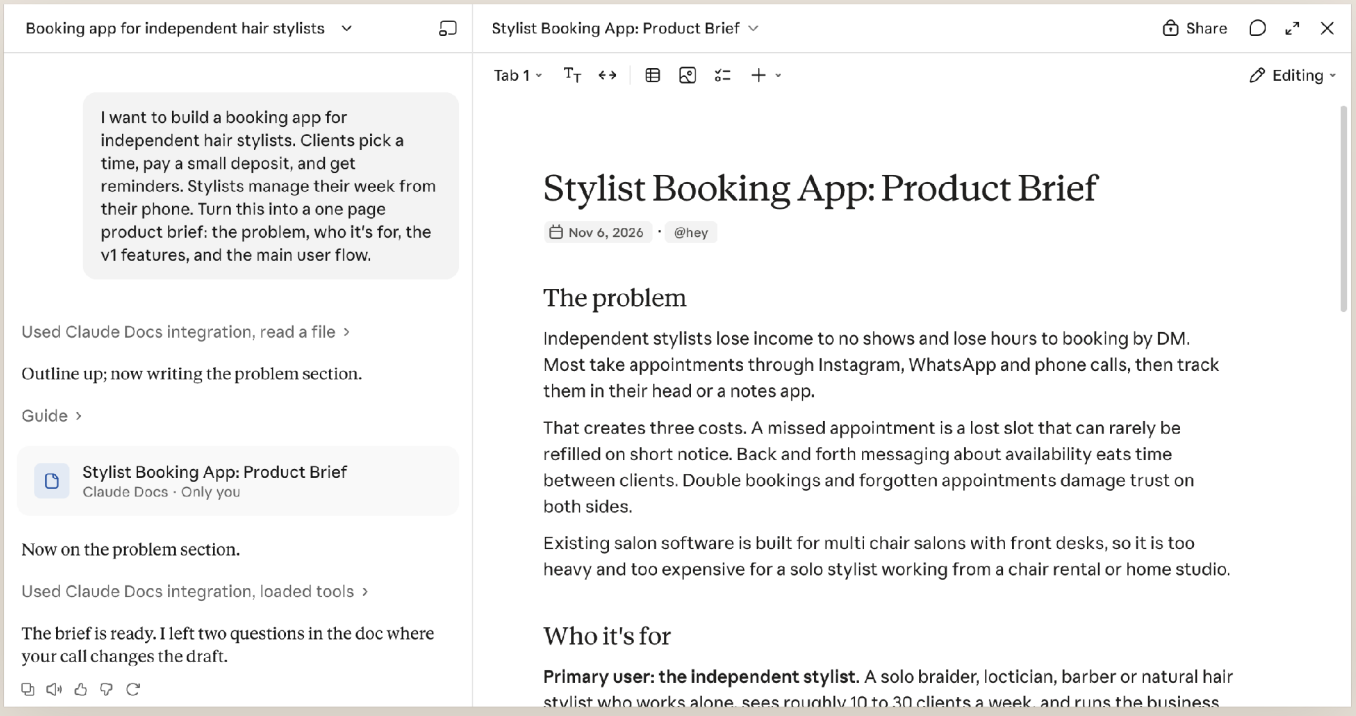
What you get: a brief you can hand to anyone, plus the questions you hadn't thought about.

PROMPT 2 Poke holes in it

```
Here is my product brief: [paste the brief]. Act as a sceptical investor and an experienced product manager. List the 5 riskiest assumptions in this idea, ranked from most to least dangerous. For each one, tell me the cheapest way to test it in 7 days without writing code.
```

What you get: the things most likely to kill the idea, and how to check them this week.

REAL EXAMPLE



Our real Claude chat: the prompt and the brief it wrote for booked.

Step 2: Build a prototype

Turn the brief into something people can actually tap through. It doesn't need to be perfect. It needs to work.

PROMPT 3 The build spec

```
Using this brief: [paste the brief], write a build spec for an AI app builder. List every screen in the main flow. For each screen give: its purpose, what's on it, the main action, what happens when the user taps it, and the empty, loading and error states. Then list the data the app needs to store. Keep it to the v1 features only.
```

What you get: a clear plan to paste into your AI app builder (Lovable, Bolt or Replit) or into Claude itself.

PROMPT 4 The design direction

```
Give my app [app name] a design direction that feels [3 words, for example: warm, premium, simple]. Suggest a name check, a colour palette with hex codes, one font pairing, and one hero element that makes the main screen memorable (for example a photo stack, a big number or a card). Keep it realistic for a first version.
```

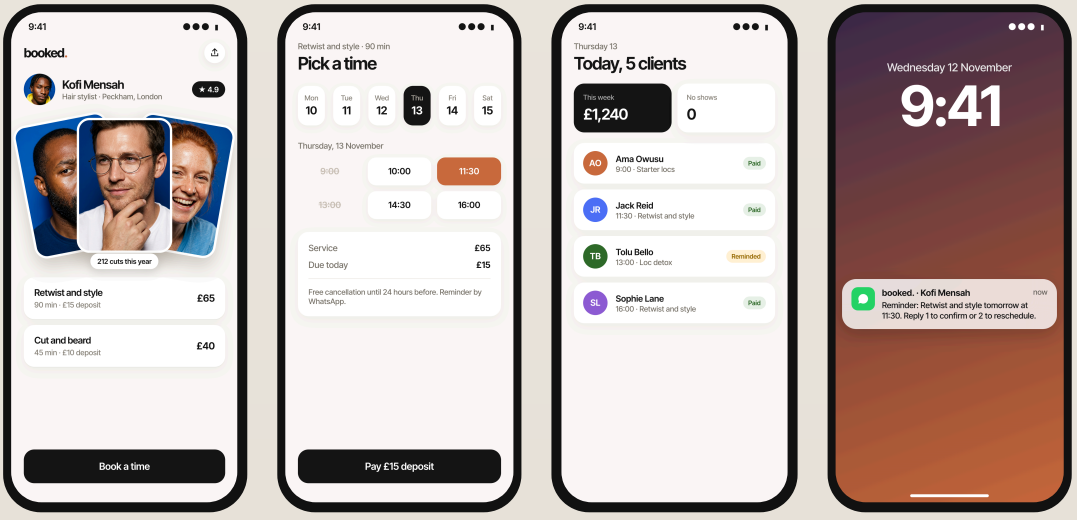
What you get: a look that feels like a real product, not a template.

PROMPT 5 Realistic demo data

```
Create realistic sample data for my prototype of [app name]: [number] users, [number] [main items, for example bookings or orders], with believable names from [your market], prices in [currency], dates in the next two weeks, and a few edge cases (a cancellation, a late payment, an empty profile). Format it as a table I can paste into the app.
```

What you get: a prototype that feels alive when people test it.

REAL EXAMPLE



The booked. prototype: booking page, pick a time, the stylist's day and the WhatsApp reminder.

Step 3: Test with people

Put it in front of 5 to 10 people who match your user. Watch where they get stuck.

PROMPT 6 Find your testers

```
My product is for [who it's for] in [city or country]. Give me 10 specific places,
online and offline, where I can find 5 to 10 of these people this week. Then write a
short, friendly message I can send asking for 15 minutes of their time to try a prototype.
No hard selling.
```

PROMPT 7 The test script

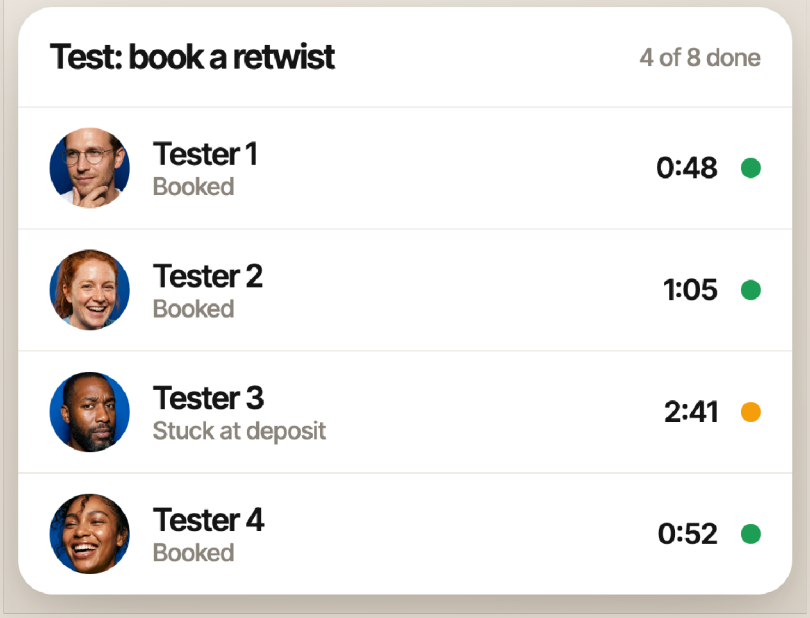
```
Write a 15 minute user test script for my prototype of [app name]. Include a short intro
that puts the tester at ease, 3 tasks for them to complete without help (based on this
main flow: [paste the flow]), and 5 open questions that don't lead them to the answer I
want. Add notes on what I should watch for during each task.
```

PROMPT 8 Make sense of the feedback

```
Here are my raw notes from [number] user tests: [paste notes]. Group the feedback into
patterns. For each pattern, tell me how many testers hit it, quote them where useful, and
rate how much it hurts the main flow. Finish with the top 3 fixes I should make before
anything else, and what I should NOT build yet.
```

What you get: clear next moves, not a pile of opinions.

REAL EXAMPLE



Our test panel. Tester 3 got stuck at the deposit, so that screen got fixed first.

Step 4: Check what's missing

AI gets you surprisingly far. These are the things that look fine in a demo and break with real users.

PROMPT 9 The pre launch audit

```
Before real customers use [app name], what could break? Review it for security, payments, user data and performance. Here is what it does: [paste the brief and the build spec]. Reply as a short checklist grouped under those 4 headings, and mark the 3 items that would hurt most if they failed.
```

PROMPT 10 What breaks at 1,000 users

```
Imagine [app name] goes from 10 users to 1,000 in one month. Walk me through what is most likely to break first: technically, in operations and in support. For each one, tell me the early warning sign and what a proper fix looks like.
```

What you get: an honest list of what your prototype can't do yet.

REAL EXAMPLE

Before real customers use this, what could break? Check security, payments, user data and performance. Reply here in chat as a short checklist, no doc.

Security

- [ ] Booking links can't be enumerated to scrape other stylists' clients (use random slugs, rate limit).
- [ ] Reschedule and cancel links use long, single use, expiring tokens.
- [ ] Stylist login uses OTP with rate limits to block SMS pumping and account takeover.
- [ ] Every API call checks the stylist owns the booking, service or payout it touches.
- [ ] Bot protection on the booking form so fake bookings can't block a stylist's week.

Payments

- [ ] Use a hosted checkout (Stripe, Paystack, Flutterwave) so card data never touches your servers.
- [ ] Payment webhooks are verified by signature and are idempotent, so retries don't double book or double refund.
- [ ] A slot is held only briefly during checkout and released if payment fails or is abandoned.
- [ ] Refund logic around the cancellation cutoff is tested at the exact boundary and across time zones.
- [ ] Payout bank details changes require reverification, since that's the main fraud target.

Claude's real audit of booked. before any customer touched it.

Step 5: Build it properly

This is where a real development team turns a promising prototype into software customers can trust.

PROMPT 11 The developer handoff brief

```
Using everything above (brief, build spec, test findings and audit: [paste]), write a one page handoff brief a development team can quote from. Include: the goal, the users, the must have features, the nice to haves, integrations (payments, messaging, email), the platforms (web, iOS, Android), what I've already validated, and the open questions they should answer in their proposal.
```

PROMPT 12 Questions to ask any dev team

```
I'm about to hire a development team to build [app name] properly. Give me 10 questions to ask them before I sign, covering how they handle security, payments, data, testing, launch, support after launch, who owns the code, and how they price changes. For each question, tell me what a good answer sounds like and what's a red flag.
```

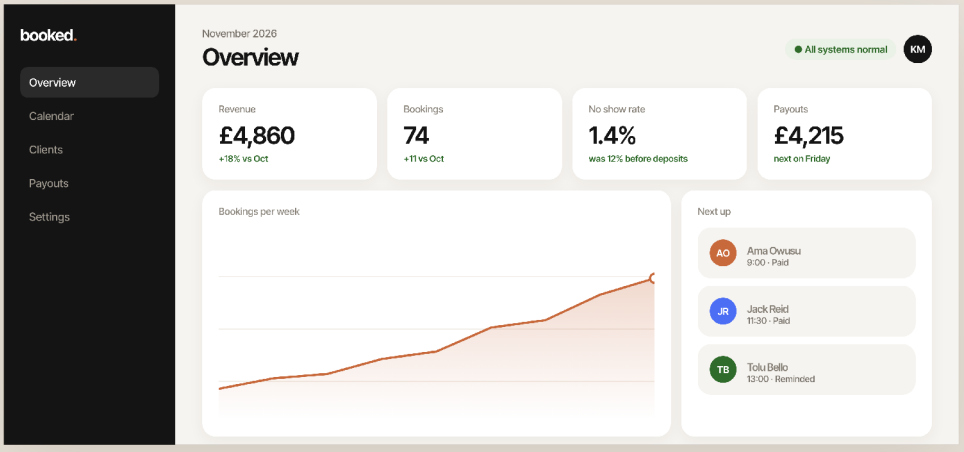
PROMPT 13 Plan the first 90 days

```
My v1 of [app name] is about to be built. Plan the first 90 days after launch: what to measure each week, the 3 numbers that show it's working, when to collect feedback, and what would make me change direction.
```

PROMPT 14 Bonus: explain it like a pitch

```
Using my brief and test findings [paste], write a 30 second explanation of [app name] I can say out loud to a customer, an investor or a developer. Plain words, no jargon, and end with what I need from them.
```

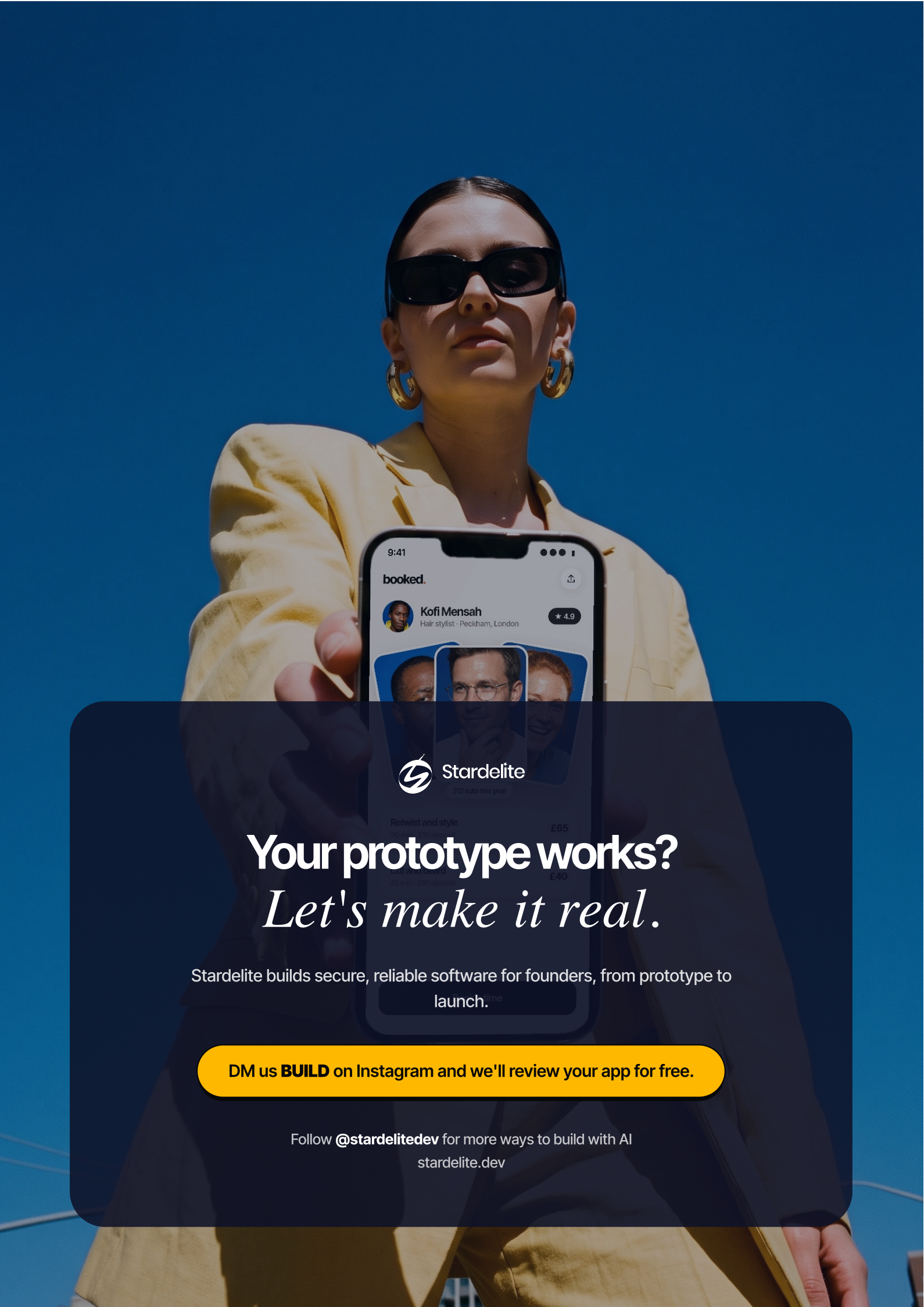
REAL EXAMPLE



What built properly looks like: the booked. dashboard with real payments, reminders and payouts.

The checklist

- [ ] I have a one page brief and know my riskiest assumptions
- [ ] I have a clickable prototype with realistic data
- [ ] 5 to 10 real people have tested it and I know the top 3 fixes
- [ ] I've checked security, payments, user data and performance
- [ ] I have a handoff brief and know what to ask a dev team



Stardelite

Your prototype works?

Let's make it real.

Stardelite builds secure, reliable software for founders, from prototype to launch.

DM us BUILD on Instagram and we'll review your app for free.

Follow **@stardelitedev** for more ways to build with AI

stardelite.dev