



STRIPE × SUPABASE • AI PAYMENTS FOR THE REAL ECONOMY • BERLIN • 24 OCT 2026

Describe your business. Get software that fits it.

One paragraph in, a booking app out: courts, bikes or stylists. Customers pay the owner through Stripe Connect. Supabase makes double bookings impossible.

• AI: spec, not code

• Stripe Connect

• Supabase Postgres

Paras Adlakha • Harshita Adlakha

The screenshot shows the 'Cadence Ride' booking interface. At the top, there's a header with the app name and a tagline 'Pick your bike, Ride to the beat.' Below this, there are two main options: 'Rhythm Ride' (45 min, from €18) and 'Hill Climb' (60 min, from €22). A calendar view shows the current date as Monday, 26 Oct. Below the calendar, there are two time slots: '07:00' (18 free) and '18:30' (15 free). A section titled 'Choose your bike' shows a grid of 18 bikes, with bike 3 selected. To the right of the bike grid, there are checkboxes for 'Free', 'Taken', and 'Yours'. Below the bike grid, there's a section for 'Your booking' which includes fields for Name (Ana Silva), Email (ana@example.com), and Shoe size for clip-in shoes (Choose...). The total price is €18, and there's a button to 'Pay €18 and book'. At the bottom, there's a note: 'Cancel up to 12 hours before class for a full refund. Payments by Stripe go straight to Cadence Ride.'

github.com/ParasAdlakha/fitted

The problem

small businesses get software that doesn't fit

34M

EU enterprises in 2024; 99% have fewer than 50 people.

Eurostat, via Portugal Global

15–25%

of salon appointments are no-shows (10–20% at fitness studios) when nobody pays upfront.

Vendor data, SchedulingKit · directional

~45%

fewer no-shows when customers pay a deposit or upfront when they book.

Square data, via SchedulingKit · directional

OPTION A · VERTICAL SOFTWARE

One industry, one subscription

Court software for the padel club, salon software for the salon, studio software for the gym. Expensive, rigid, and built for someone else's business.

OPTION B · GENERIC TOOLS

Calendar link + WhatsApp + bank transfer

No bike map, double-booked courts, nobody paid upfront, no-shows. It doesn't understand the business at all.

AI adapts the product one engine, three businesses

Kiez Padel Club
Four courts in Französerberg, black, pay, pay.

50 min match (50 min - 1 hour 15 min)
50 min practice (50 min - 1 hour 15 min)

Your booking
50 min match
Sat 18:00-19:30
Court 4
€40

Times
08:00, 08:30, 09:00, 09:30, 10:00, 10:30, 11:00, 11:30, 12:00, 12:30, 13:00, 13:30, 14:00, 14:30, 15:00, 15:30, 16:00, 16:30, 17:00, 17:30, 18:00, 18:30, 19:00, 19:30, 20:00, 20:30, 21:00, 21:30

Choose your court

RESOURCE • PADEL CLUB

"4 courts, 2 indoor, €32 a match, €40 after 5pm"
→ court map, 90-min slots, peak price, indoor + €4

Cadence Ride
Pick your bike. Ride to the beat.

Rhythm Ride (45 min - 1 hour 15 min)
Hill Climb (45 min - 1 hour 15 min)

Your booking
45 min
Sat 18:30-19:15
Bike 3
€18

Choose your bike
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18

CLASS • SPIN STUDIO

"18 bikes in 3 rows facing the instructor"
→ bike map, class schedule, live taken seats

Studio Mira
Cuts and colour in Neuville.

Cut and style (120 min - 1 hour 15 min)
Skin fade (120 min - 1 hour 15 min)
Full colour (120 min - 1 hour 15 min)

Your booking
120 min
Sat 18:45-19:45
Stylist Mira
€110

Choose your stylist
Mira (Cuts and colour), Jonas (Cuts and fades), Lea (Junkies style)

STAFF • SALON

"Mira does colour, Jonas and Lea do fades"
→ stylist cards, services limited to the right people

HOW

The model fills in a **strict JSON schema** (OpenAI structured outputs), and our normalizer checks and repairs it. A single tested renderer runs every spec. **No generated code.** Changes are plain words too: "Add a Sunday 11:00 class. Front-row bikes €4 more."

Stripe and Supabase power it

01 · AI

Describe

Paragraph → spec, about \$0.002

02 · SUPABASE

Publish

Business row + spec as jsonb, versioned

03 · STRIPE CONNECT

Onboard

Express account, Stripe-hosted KYC

04 · SUPABASE

Hold

book_slot() holds the unit 15 min

05 · STRIPE CONNECT

Pay

Checkout → owner's account, 3% fee

STRIPE CONNECT

- **Express account per business**, onboarding and payouts handled by Stripe
- **Destination charges:** `transfer_data.destination` = the owner
- **Platform revenue:** `application_fee_amount` = 3%
- **Webhooks:** session completed, expired (frees the slot), account updated

SUPABASE · THE DATABASE IS THE REFEREE

```
constraint no_overlap exclude using gist (  
  business_id with =,  
  resource_key with =,  
  tstzrange(starts_at, ends_at) with &&  
) where (status <> 'cancelled')
```

Two people tap bike 3 at the same second: Postgres accepts one, the other gets "just taken". RLS locks every table; one schema serves every kind of business.

How we keep costs low

COST PER BUSINESS

AI to set up the app (one call)	\$0.002
AI per plain-words change	\$0.002
AI per customer booking	\$0
Generated code to host or deploy	none
Infrastructure per extra business	one row

- **AI at setup, not per booking.** Cost stays flat as bookings grow
- **Small model, minimal reasoning,** strict schema: ~2,500 tokens, 8–12 s
- **Metered and shown** to the owner on the dashboard (ai_calls)
- **Revenue from payments:** a 3% Connect fee, no subscription to justify

EXAMPLE • PADEL CLUB

300 bookings × €32 = €9,600 a month through Stripe. 3% = **€288 a month** to the platform, against **\$0.012** of AI spent once.

Why us

systems thinkers who ship

ENGINE • STRIPE • SUPABASE

Paras Adlakha

Aerospace, electronics, systems engineering. Treats a booking like a safety-critical record: one source of truth, constraints in the database, nothing trusted from the model without checks.

RESEARCH • PRODUCT • AI SPEC

Harshita Adlakha

System dynamics, socio-technical systems. Mapped dozens of small businesses onto three shapes (resource, class, staff), the insight that lets one engine serve them all.

TRACK RECORD

Yesterday: **LineCheck**, an AI inspector aircraft technicians call on FaceTime (Beta Fund × AWS × OpenAI hackathon, SF). Today: Fitted, with 12 tests passing and a working Stripe Connect flow.

Try it: describe a business → book → pay with 4242 4242 4242 4242

github.com/ParasAdlakha/fitted