

Fuel Around and Find Out

The itch

- I have a **ton of ideas**. I finish them to ~90%, then move on.
- This time: go to **100%**, put it **live**, and **find out** what happens.
- Side quest: learn how **AI behaves in production**, and whether you can **test** it.

> The real problem: on long rides I bonk if I **under-fuel** and if I **over-fuel**.
> What if someone could parse your route, ask a few questions and get you a plan?

BANANAGOAT.

- Teamed up with a friend who does **marketing / branding / SEO**.
- **2 months**, **~380 name ideas**.
- **bananagoat.com**: **Banana** (athlete fuel) + **Goat**.

Tone: technical but friendly, data-driven. The fun is already in the name.

Positioning: personalised fuelling, built for confidence over hype.

The product

Upload a route, set effort + a few preferences, get a fuelling plan.

- **Route:** distance, elevation, terrain, detected climbs
- **Conditions:** temperature, wind, humidity
- **Athlete:** weight, fitness, goal, foods you have

You get carbs/hour + an eat/drink schedule, a route-profile sheet, and a fuelling sticker for your bike.

Today: a website with a prototype form that emails you the plan.



The tech

```
def on_route_submit(gpx, athlete):
    route   = parse_gpx(gpx)
    route   = calculate_gradient(route)
    climbs  = detect_climbs(route)
    climbs  = match_climbs_in_db(climbs)
    places  = reverse_geo(route)
    weather = fetch_weather(route, date)

    energy  = estimate_energy(route, climbs, weather)
    perf    = calc_athlete_performance(athlete, energy)

    plan = llm_fueling_plan(route, climbs, weather, energy, perf)
    plan.pack_list = compute_pack_list(plan)

    sticker = sticker_renderer(route, climbs, plan)
    send_email(athlete, plan, sticker)
```

The AI part

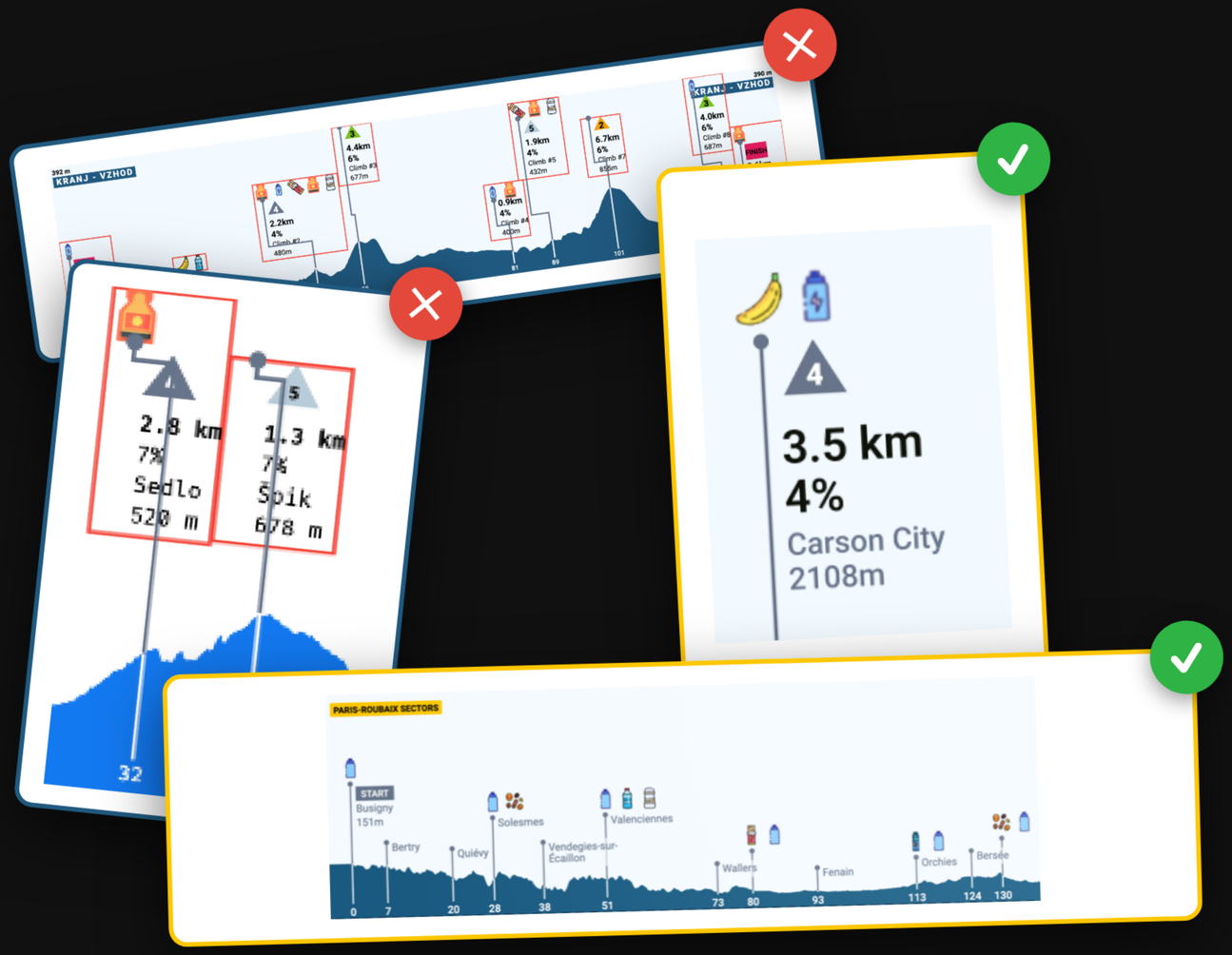
- One **LLM** writes the plan. Everything else (geo, climbs, physics) is **my own code**.
- Routed through **OpenRouter**: swap models with a one-line slug change (`openai/gpt-5-mini` today).
- Output is **structured JSON** validated against a Pydantic schema, not free text.
- **Prompt is versioned**; the deterministic numbers are computed, not hallucinated.

How I test it (yes, you can test an LLM):

- **LangChain + LangSmith** harness, swap the eval model with one env var.
- **Deterministic validators** + **LLM judges** on a **different** model family (Claude) for nutrition rules.

What were the challenges

- **2D sticker layout:** placing climb tiles on the elevation profile without overlaps.
- **AI prompt engineering:** consistent, safe, structured plans.
- **Altitude & aerodynamics:** real GPX is noisy; calculating wind and draft is not so trivial.
- **Determinism:** same route twice could give **two different plans**. For a **trust** product, that is the scary one.



The real bottleneck: getting users

A **single Facebook post** attracted **~330 users** · **0 ads** · **0 marketing campaigns**

- Funnel: **330** land, **165** touch the widget, **83** finish the form, **~19** come back.
- **~23%** who get a plan come back for another. Engaged sessions **~6 min**.

> The product works for people who reach it. Getting them there is the unsolved problem.

What's next?

- **Getting out there: user growth.** If that's your thing, come find me after.
- Find a **nutrition expert** to refine the fuelling strategy.
- **App** (mobile / web / desktop) with generative AI.

Fuel around and find out. 🍌🐐



bananagoat.com