BoniBuddy app snapshot

What it is

FastAPI + Jinja web app that helps students going for "bone" meals coordinate and see who’s headed to resta

Who it’s for

Students using Slovenian subsidized meals ("bone"), focused on Ljubljana restaurants.

What it does

- Live waiting board showing per-restaurant counts for now/+30/+60 minute buckets.
- Join or leave a time bucket; auto-moves members forward as time passes.
- Matchmaking with mutual gender preference plus city/location filters; shows match page on pair.
- Web push opt-in per request id; sends best-effort push on matches when VAPID keys set.
- Searchable restaurant list seeded from data/restaurants.json (Ljubljana).
- PWA assets (manifest, service worker, icons) for add-to-home experience.

How it works

- FastAPI app (app.py) serves HTML (Jinja templates) and static/PWA assets.
- engine_web.py holds domain logic and in-memory stores: restaurants, waiting slots, requests, push subs.
- Matching: form submit -> add_request_with_pairs -> checks waiting list under lock -> pairs users on mutua
- Waiting board: get_waiting_board aggregates counts per restaurant/bucket; UI renders via template routes
- Env flags for GA4 and VAPID are optional and non-blocking.

How to run

1. python3 -m venv .venv && source .venv/bin/activate
2. pip install -r requirements.txt
3. uvicorn app:app --reload --port 8000
4. Open http://localhost:8000/
5. Optional: set VAPID_PRIVATE_KEY, VAPID_SUBJECT, GA4_MEASUREMENT_ID, GA4_API_SECRET

Not found in repo

- Pricing/hosting details
- Authentication or user identity beyond Instagram handle
- Persistent storage/DB (state is in-memory)
- Coverage outside Ljubljana restaurant seed