Candidate Take-Home — Geo Ingestion Lite (FastAPI + PostGIS)

Timebox: 3–5 hours • **Deliverables:** GitHub repo + README

Goal

Design a small FastAPI service using PostGIS for basic geospatial tasks. No prior PostGIS experience required. Focus on correctness, clarity, and documentation.

User Stories

1. Create feature: POST /features with { name, lat, lon } → returns { id }.
2. Process feature: dev-only POST /features/{id}/process builds 500 m buffer + area in PostGIS.
3. Query: GET /features/{id} returns status + buffer_area_m2.
4. Nearby: GET /features/near?lat&lon;&radius;_m returns features within radius using ST_DWithin.

What You Must Implement

• Alembic migration: enable PostGIS, create tables (features, footprints) with geography columns and GIST indexes.
• Service layer: implement SQL functions (point insert, buffer/area, ST_DWithin).
• Endpoints: connect routes to service functions (already scaffolded).
• Tests: make provided smoke test pass; add one small unit test if time allows.

Endpoints

POST /features	Body {name,lat,lon} → {id}
POST /features/{id}/process	Dev-only; compute buffer + area, set status=done
GET /features/{id}	Return status, buffer_area_m2
GET /features/near?lat&lon&radius_m	Nearby features ordered by distance
GET /healthz /readyz	Basic health endpoints

Suggested Schema

features(id UUID PK, name TEXT, status TEXT, attempts INT, created_at TIMESTAMPTZ, updated_at TIMESTAMPTZ, geom geography(Point,4326)) + GIST index
footprints(feature_id UUID PK FK, buffer_m INT, area_m2 DOUBLE PRECISION, created_at TIMESTAMPTZ, geom geography(Polygon,4326)) + GIST index

PostGIS Cheat-Sheet

```
Enable: CREATE EXTENSION IF NOT EXISTS postgis;
Point: ST_SetSRID(ST_MakePoint(lon, lat),4326)::geography
Nearby: ST_DWithin(f.geom, ref.g, :radius)
Dist: ST_Distance(f.geom, ref.g)
Buffer: ST_Buffer(geom, :buffer_m)
Area: ST_Area(geom)
```

Evaluation Rubric

Architecture (30%): clean boundaries, correct PostGIS usage.
Code quality (30%): FastAPI best practices, typing, config.
Reliability & ops (20%): health/readiness endpoints, logs, retries.
Testing & docs (20%): smoke test passes, clear README.

Getting Started

1. docker compose up --build
2. POST /features body: {"name":"Site A","lat":45.5017,"lon":-73.5673}
3. POST /features/{id}/process
4. GET /features/{id} and GET /features/near?lat=..&lon;=..&radius;_m=1000
5. pytest

Submission

GitHub repo link + README (setup, run instructions, trade-offs). Optional: short video walkthrough.