Python AI Engineer — Skills ↔ Super-Skills Mapping API (Take-Home)

Goal

Build a small, production-minded API that ingests raw skills, maps them to canonical **super-skills (parent skills)**, and exports a 100-row Excel with fully mapped examples. We're looking for clean code, pragmatic design, and thoughtful trade-offs (accuracy, performance, DX).

Tech

- **Python** (Frameworks of candidate's choice)
- **Database: MongoDB** recommended (or PostgreSQL if you justify it)
- **Containerization:** Docker
- **Cloud deploy:** any provider (**GCP** recommended)

Requirements

1) Domain & Endpoints

Implement a REST API for **Mappings**.

Mappings (one skill → multiple super-skills)

- POST /mappings – upsert { skillId, Array[superSkillId]}

2) Data Model (suggested)

- **Skill**: view rawskills_samples.xlsx (Refer to references)
- **SuperSkill**: view metadata_superSkill.json (Refer to references)
- **Mapping**: { _id, skillId, Array[superSkillId]}

3) Business Logic (must-haves)

- **No duplicate Skills** by normalized (case/space/punct-insensitive). Upsert accordingly.
- **One skill** → **Many super-skill**; enforce uniqueness at DB level.
- **Validation & errors**: clear 4xx on bad input; 409 on duplicate conflicts

6) Containerization & Deploy

- **Dockerfile** for the API;

- **Environment** via .env
- **GCP (recommended):** provide commands to build, push, deploy; set env vars; attach Mongo Atlas or a managed instance. Alternatives (Render/Fly.io/Azure) acceptable with docs.

Deliverables

- Public GitHub repo with:
    - **README**: setup, run, test, example .env, **design decisions** and **trade-offs** (brief but specific), and deployment steps.
    - **API docs** (sample curl/Postman).
    - **DB schema note** (collections, indexes).
    - Returns an Excel with **600 sample skills fully mapped**.
        - Sheet: **MappedSamples** with columns (in order):
            - _id (From mongodb)
            - skill_raw
            - skill_super (Array)

Evaluation Criteria

- **Mapping correctness**
- **Completeness** (all required endpoints + Excel export).
- **Code quality & structure** (readability, separation of concerns).
- **Validation, errors** done right.
- **Query efficiency & indexes** justified.
- **Thoughtful README** (assumptions, corner cases, next steps).

References:

- rawskills_samples.xlsx
- https://drive.google.com/file/d/10F8ykZL5pA6idsXa2y6uXU0SN3BCMUvq/view?usp=drive_link