GeoWeather

GeoWeather is a Python CLI app that lets users check weather and explore city data. You will:

- Get current weather or forecasts from OpenWeather API
- Search and filter cities in a local PostgreSQL database
- Display results in tables and plots
- Practice Python, SQL, and data visualization

CLI Commands and Functionalities

1. API-Driven Commands
    - 1.1. **`weather <city>`**
        - 1.1.1. Prints current weather status for a city
        - 1.1.2. Uses **`--units`** parameter to switch units (metric/imperial)
        - 1.1.3. Optional: caches API responses locally to reduce API calls
    - 1.2. **`forecast <city>`**
        - 1.2.1. Plots forecast: temperature, wind, rain for the next days
        - 1.2.2. Uses **`--units`** parameter to switch units (metric/imperial)
        - 1.2.3. Optional: save plot as PNG
    - 1.3. **`overview <country>`**
        - 1.3.1. Prints current temperature for the largest 3 cities in the country
        - 1.3.2. Uses **`--units`** parameter to switch units (metric/imperial)
        - 1.3.3. Optional: plots temperatures from highest to lowest

2. Database-Focused Commands
    - 2.1. **`top_cities <country> --n <number>`**
        - 2.1.1. Lists top N most populous cities in a country
        - 2.1.2. Default N=5
        - 2.1.3. Plots a pie chart
    - 2.2. **`city_search <partial_name>`**
        - 2.2.1. Lists all cities matching a partial name
        - 2.2.2. Includes population, country, coordinates
    - 2.3. **`nearby <lat> <lng> --radius <km>`**
        - 2.3.1. Finds all cities within X km of given coordinates
        - 2.3.2. Default radius 50 km
    - 2.4. **`population_stats <country>`**
        - 2.4.1. Shows average, max, min population for cities in a country

3. Bonus
    3.1. **multi_weather <city1> <city2> ...**
        3.1.1. Show current weather for multiple cities
        3.1.2. Plot multiple forecasts on one chart
    3.2. **coldest_capital <continent>**
        3.2.1. Show which capital city has the lowest temperature

Suggested Folder Structure

```
geo-weather/
├─ main.py                 # CLI entry point
├─ requirements.txt
├─ docker-compose.yml
├─ config/
│   ├─ init_db.sql
│   └─ data.csv
└─ utils/
    ├─ db.py               # DB connection and queries
    ├─ weather_api.py      # OpenWeather API calls
    └─ plot.py             # plotting helper
```

Learning Outcomes

- How to pass arguments to a Python script
- How to query a database and filter results
- How to call an external REST API
- How to organize and manipulate JSON with pandas
- How to visualize time-series data