Software Release Document

Geoair

Gianluca Bettoni

Mobina Faraji

Alessia Ippolito

Edoardo Pessina

POLITECNICO
MILANO 1863

Index

1. Introduction *1*
2. Prerequisites 3
4. Running the Application 4
5. Project Structure 4
6. Additional Notes 5
7. Data Sources (APIs) 5

Repository:

https://github.com/astroedo/air_quality_analysis

1. Introduction

Air Quality Analysis is a web application built with Dash and Flask to visualize and analyze environmental air quality pollutant data for the Lombardia region. The project uses a PostgreSQL database with PostGIS extension and visualization libraries like Plotly, Dash Leaflet, and GeoPandas.

2. Prerequisites

- **PostgreSQL (Windows):** Download and install from https://www.postgresql.org/download/windows/
- **PostGIS:** Install via StackBuilder (included with PostgreSQL installer), selecting the PostGIS extension for spatial features.
- **Python 3.8 or higher** installed.

Python dependencies: Install with:
pip install -r requirements.txt

- (Dependencies include Flask, psycopg2-binary, pandas, numpy, requests, dash, plotly, dash-leaflet, geopandas, Werkzeug, ecc.)

3. Database Setup and Data Loading

- Create the database `lombardia_air_quality` (if not existing).
- Ensure a PostgreSQL user exists with privileges on the database:
  - database = lombardia_air_quality
  - user = airdata_user
  - password = user
- Database creation, PostGIS enabling, and data loading are managed by Jupyter notebooks located in the `database/`folder.
- Open VSCode or Jupyter, navigate to the `database/` folder, and run the notebooks in order:
  - `database_station.ipynb`: Enables PostGIS (`CREATE EXTENSION postgis;`), fetches sensor data from Lombardia API, inserts into DB.
  - `database_measurement.ipynb`: Fetches sensor measurements data from Lombardia API, inserts into DB.
  - `database_user.ipynb`: Creates initial users and inserts into DB.

4. Running the Application

Start the Flask backend API (runs on port 5001):
python server.py

- API accessible at http://localhost:5001

Start the Dash frontend (runs on port 8000):
python app.py

- Dashboard accessible in a browser at http://localhost:8000

5. Project Structure

```
AIR_QUALITY_ANALYSIS/
├── requirements.txt       # Python dependencies list
│
├── database/              # Jupyter notebooks for DB setup and data loading
│   ├── database_station.ipynb
│   ├── database_measurement.ipynb
│   └── database_user.ipynb
│
├── server.py
├── app.py
│
├── pages/                 # Dash page layouts and callbacks
│   ├── home_page.py       # Home
│   ├── login_page.py      # Login
│   ├── map_page.py        # Map of the stations
│   └── graph_page.py      # Graph of the pollutants
│
├── components/
│   ├── map_component.py          # api for the home page map
│   ├── dropdown_component.py
│   │
│   ├── fetch_pollutant.py        # api for the home page map
│   └── logger.py
│
├── maps/                  # file for the map like .shp
│
└── assets/                # CSS, logo, img
```

6. Additional Notes

- VSCode is the recommended IDE; pgAdmin4 is used for DB management and spatial reference system configuration.
- The application has been tested only on Windows and macOS.
- Data updates occur by fetching Lombardia regional API data and inserting it into the database via python jupiter notebooks file in `database/` folder.
- Backend (server.py) and frontend (app.py) must be run separately.
- The logging system provides event/error tracing that gives feedback on api call.
- Logging is handled via `components/logger.py` using `setup_logging()`, which records timestamp, level, and message.
- Update Python dependencies with: pip install --upgrade -r requirements.txt
- Periodically rerun the notebooks in `database/` to update database data.

7. Data Sources (APIs)

The application loads air quality data from the official Lombardia regional open data APIs:

- **Measurement Data API**
  - Description: Provides sensor measurements data such as pollutant values, timestamps, and sensor status since 2018.
  - API documentation: https://www.dati.lombardia.it/Ambiente/Dati-sensori-aria-dal-2018/g2hp-ar79/about_data
  - API endpoint (JSON): https://www.dati.lombardia.it/resource/g2hp-ar79.json
  - Key fields: `idsensore`, `data` , `valore` , `stato` , `idoperatore`
- **Station Data API**
  - Description: Provides information about air quality monitoring stations including location, sensor types, and administrative data.
  - API documentation: https://www.dati.lombardia.it/Ambiente/Stazioni-qualit-dell-aria/ib47-atvt/about_data
  - API endpoint (JSON): https://www.dati.lombardia.it/resource/ib47-atvt.json
  - Key fields: `idsensore`, `nometiposensore`, `unitamisura`, `idstazione`, `nomestazione`, `quota`, `provincia`, `comune`, `storico`, `datastart`, `datastop`, `utm_nord`, `utm_est`, `lat`, `lng`, `location`

The Jupyter notebooks in the `database/` folder use these APIs to fetch and load data into the PostgreSQL/PostGIS database.