Environmental Monitoring and Pollution Prediction MLOPS Final Project

Task 1: Managing Environmental Data with DVC

Step 1: Research Live Data Streams:

The list of currently usable data streams is as follows:

OpenWeatherMap API:

- Gathered weather parameters like temperature, humidity, and air pollutant contents.

AirVisual API (IQAir):

- Offering air quality data as AQI and pollutant concentration.

EPA AirNow API:

- It gives air quality data from different U.S locations.

NOAA Weather API:

- Provides weather-related data.

Chosen API:

The OpenWeatherMap API was chosen for its simplicity, universality, and comprehensive weather and air quality data.

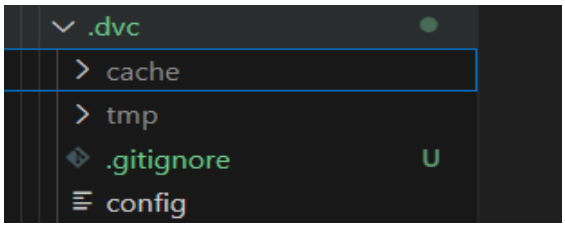
Comparison Table:

API	Features	Data Scope	Ease of Integration
OpenWeatherMap	Weather, air quality, pollution data	Global	High
AirVisual (IQAir)	Air quality, pollutant concentration	Global	Moderate
EPA AirNow	Air quality data from U.S. locations	U.S. specific	Moderate
NOAA Weather	Weather-related data	U.S. and oceans	Moderate

Step2: DVC was initialized

dvc init

- git commit -m "Initialize DVC in the repository"

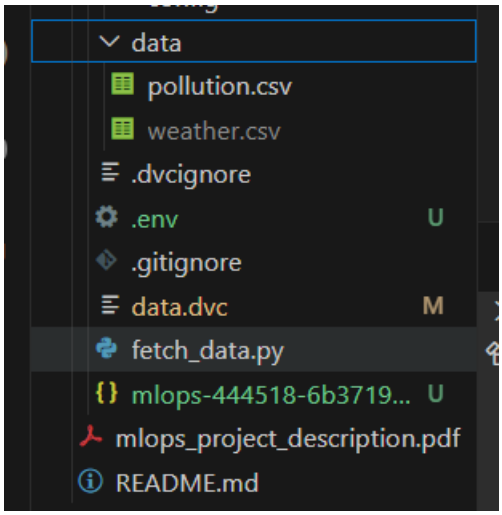


Step 3: Remote Storage Configuration

- Creating a folder first
- Folder Location: https://drive.google.com/drive/folders/12N4Q6EgP7rO38_5rHDpMVjm9BJtSkNRs?dmr=1&ec=wgc-drive-globalnav-goto
- dvc remote add -d gdrive_remote gdrive://12N4Q6EgP7rO38_5rHDpMVjm9BJtSkNRs

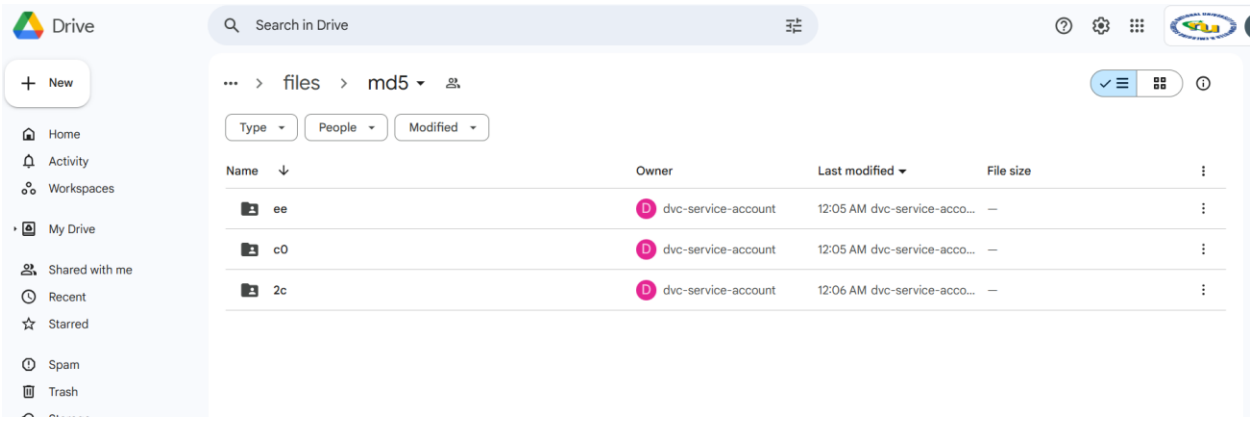
Step 4: Data Collection Script

- Fetch_data.py is a Python program created to pull weather and pollution data from OpenWeatherMap API.
- After that 2 files were created



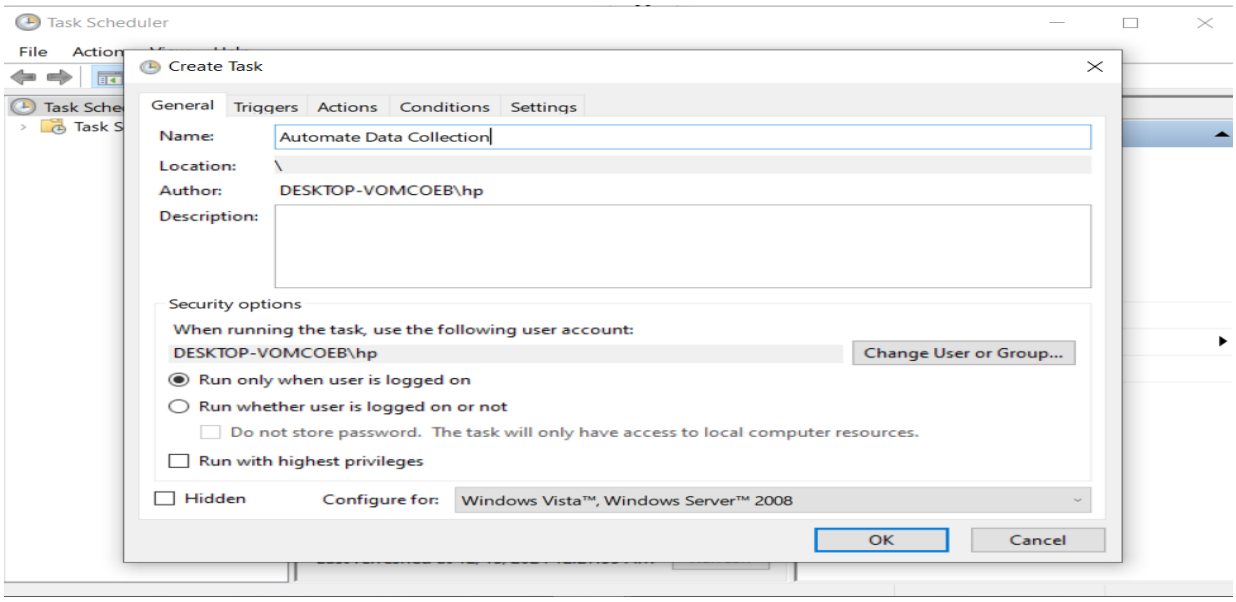
Step 5: Version Control with DVC

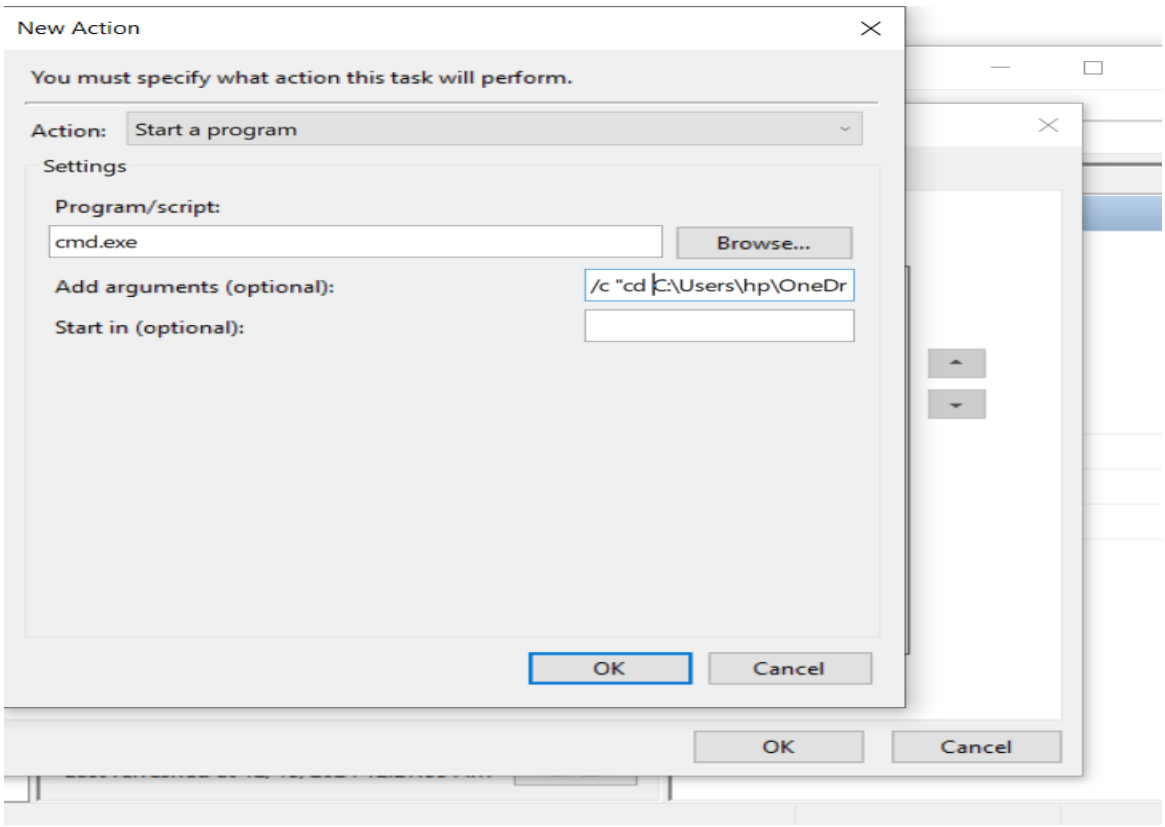
- Add data files to the DVC repository for tracking: these are the weather and pollution data.
- Commit changes to Git so that changes can be tracked in version history.
- Push the data onto Google Drive ensuring it is hosted remotely.

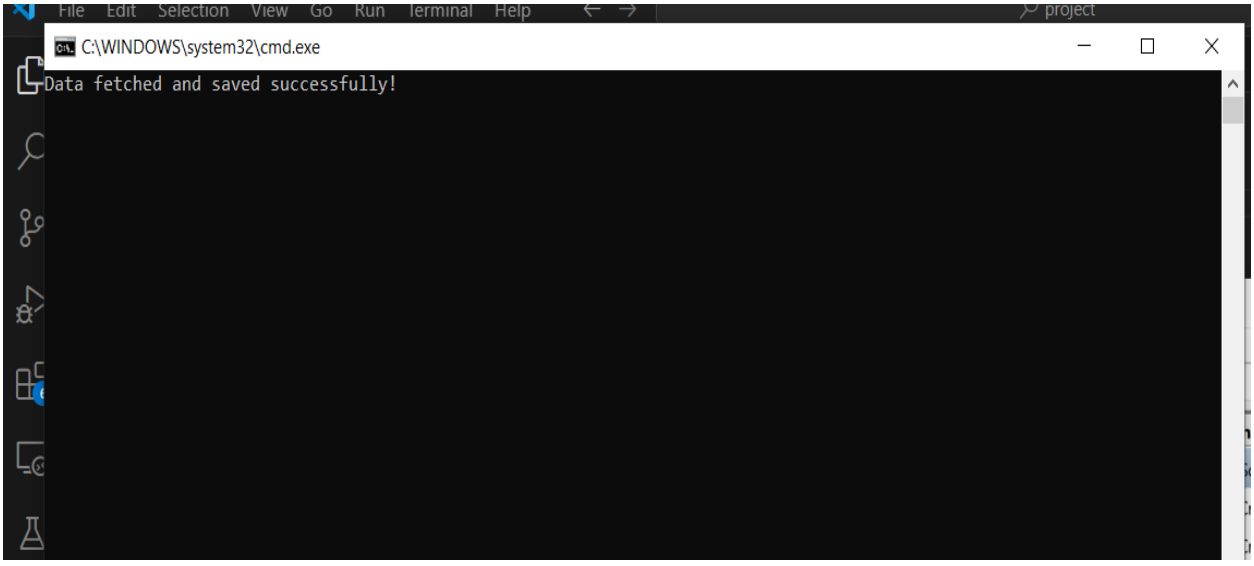


Step 6: Automate Data Collection

- Used Windows Task Scheduler to schedule work to run the script at intervals.
- Configured it to collect data and update the DVC repository automatically.

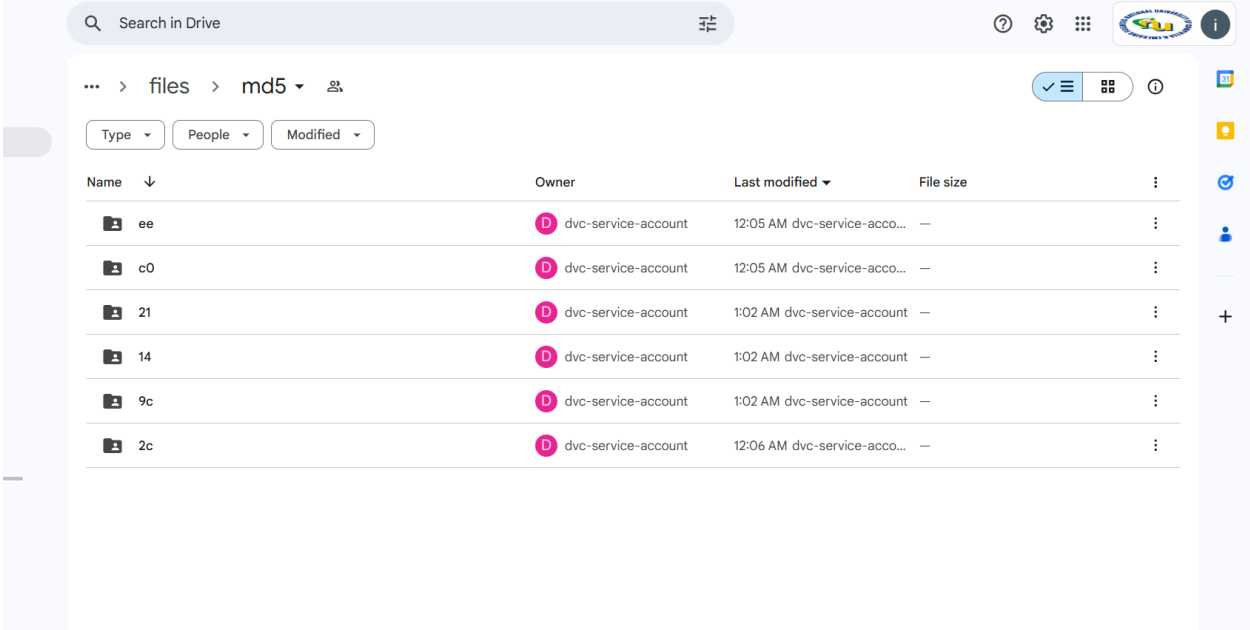
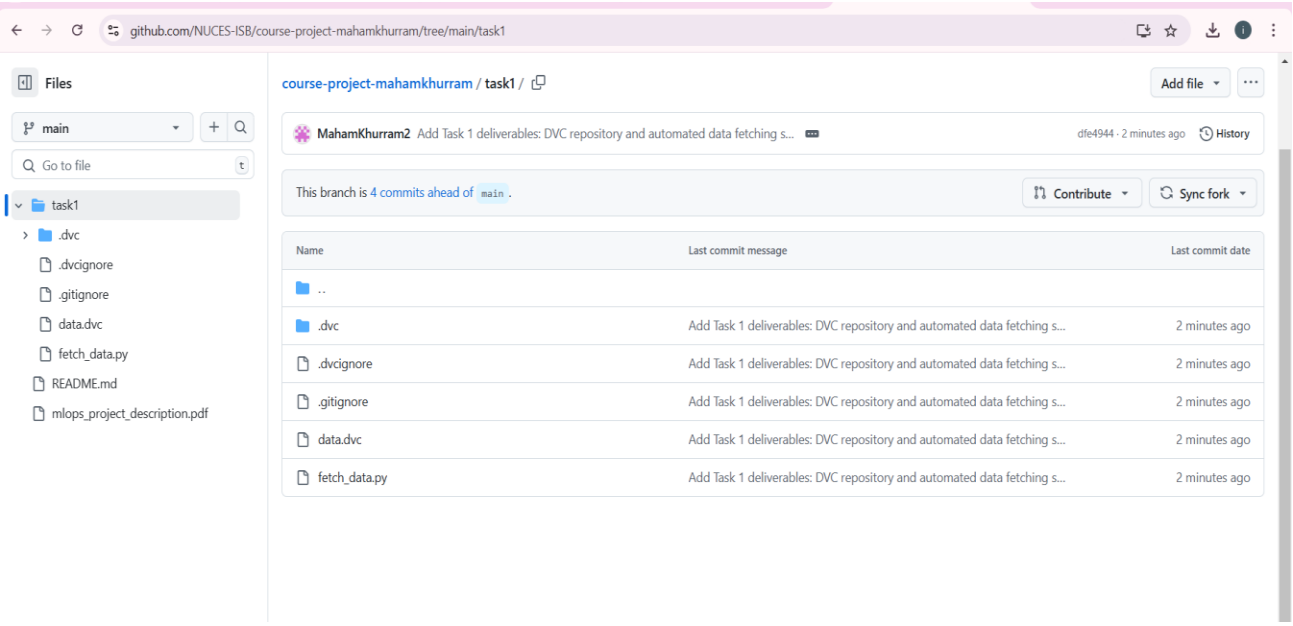






Step 7: Update Data with DVC

Used same commands to update the data directory in the DVC repository by using dvc add to stage the new data files and dvc commit followed by dvc push to version and push the changes to the remote storage.



Deliverables: