Lab 5 Part 2 - Oil Wells Analysis and Visualization

Important Files and Folders

1. `oil-wells-app/.env`: File with environment variables.
2. `oil-wells-app/requirements.txt`: File with dependencies to be installed for the project.
3. `oil-wells-app/`: Directory containing the source code for the web app.
4. `docs/`: Documentation including meeting minutes and README in PDF format.
5. `notebooks/`: Experimental usage and testing of concepts.

Setup

1. Navigate to the `oil-wells-app` directory:

```
cd oil-wells-app
```

2. Create and activate a virtual environment:

```
virtualenv venv
source venv/bin/activate  # or "venv\Scripts\activate" on Windows
```

3. Install the necessary libraries:

```
pip install -r requirements.txt
```

Running the Web Application

1. Ensure that:
   - All the important files and folders listed above are present at the correct location.
   - The virtual environment is created and activated.
   - `Apache2.4` is installed on your system and the service is running.
2. To run the script:
   - **Navigate to the Project Directory**: First, ensure you are in the /oil-wells-app directory. You can change to this directory using the following command:

     ```
     cd oil-wells-app
     ```

   - **Setting up Reverse Proxy Settings for Apache Server**: To configure the Apache Server to act as a reverse proxy for the FastAPI application, add the following snippet to the httpd.conf file:

```
<VirtualHost *:80>
   ServerName localhost:8000

   # Preserve the original host header when proxying requests
   ProxyPreserveHost On

   # Proxy all requests to the FastAPI application running on port 8000
   ProxyPass / http://127.0.0.1:8000/
   ProxyPassReverse / http://127.0.0.1:8000/
</VirtualHost>
```

Location of `httpd.conf` file:

- On Linux: The `httpd.conf` file is typically located in the `/etc/apache2/` directory.
- On Windows: You can usually find it in the `conf` directory inside the Apache installation folder, such as `C:\Program Files\Apache Software Foundation\Apache2.4\conf`.
- On macOS: The file can be found in the `conf` directory inside the Apache installation folder, similar to the Windows location.

- **Start the web app**: Run the following command to start the FastAPI web application:

```
uvicorn app.main:app
```

This command launches the FastAPI application, making it available for requests on port 8000 by default. If you wish to use a different port, you can specify it using the --port option followed by the desired port number.

3. Script Operation and Communication with Apache Server:

   The script starts a FastAPI application using uvicorn on port 8000. It sets up Apache to act as a reverse proxy, forwarding requests to the FastAPI app. This allows Apache to handle incoming HTTP requests and route them to the FastAPI application, which processes the requests and generates responses. Apache then sends these responses back to the client.

About Python Files

File Name	Purpose
`crud.py`	This file contains function definitions responsible for querying and processing oil wells data.
`database.py`	In this file, a connection is established between Python and the SQL server. It contains code to handle database connections, execute SQL queries, and manage transactions.
`main.py`	Entry point for the FastAPI web application, handling basic routing and middleware configurations. It serves the home page with the oil wells analysis and visualization, and redirects any incorrect routes to the home page for a seamless user experience.

File Name	Purpose
`mapgen.py`	This module creates a Folium map by processing well data and converting it into GeoJSON format. It performs format conversion on the well data to generate a GeoJSON representation suitable for mapping. The module then plots markers on the map, along with tooltips and popups to provide additional information about each well. Finally, it returns an HTML representation of the map embedded in the generated webpage.
`model.py`	In this file, the schema for data storage in the MySQL server is defined. It includes the structure of database tables and models, and it creates these tables if they do not already exist in the database.
`schema.py`	This file defines Pydantic models, which are used for data validation and format conversion. Pydantic models ensure that the data sent to and received from the API endpoints conforms to a specified schema.
`settings.py`	Here, environment credentials required for connecting to the MySQL database is set up. It handles configuration settings and environment variables used throughout the application.