Location Analysis API

This project is a Python backend built with FastAPI that provides analysis for a given geographical coordinate (latitude and longitude).

Features

1. **Finds Nearby Infrastructure**: Uses the Google Maps Places API to identify the nearest points of interest (schools, parks, hospitals, etc.).
2. **Local Climate Data Integration**: Reads a local NetCDF file (`Durham_AT_RH_JulAug2024.nc`) to find the most recent air temperature and relative humidity for the nearest grid cell.
3. **AI-Powered Summaries**: Leverages an AI model to generate a short, descriptive paragraph about the infrastructure and how the local climate might affect it.
4. **Categorization for Web**: Includes a `category` field in the response, making it easy to filter and display data on an interactive website.
5. **Simple API Endpoint**: Exposes a single, easy-to-use endpoint: `/analyze_location`.

Setup and Installation

Follow these steps to get the application running on your local machine.

1. Create a Project Folder

Create a new folder on your computer (e.g., `location_api`) and save the `main.py`, `create_dummy_netcdf.py`, `requirements.txt`, and `README.md` files inside it.

2. Create a Virtual Environment

It's highly recommended to use a virtual environment to manage project dependencies. Open your terminal in the project folder and run:

```
# Create a virtual environment named 'venv'
python -m venv venv

# Activate the virtual environment
# On macOS and Linux:
source venv/bin/activate
# On Windows:
.\venv\Scripts\activate
```

3. Install Dependencies

Install all the required Python libraries using the `requirements.txt` file.

pip install -r requirements.txt

4. Set Environment Variables

This application requires API keys for Google Maps and OpenAI. You need to set them as environment variables.

Create a file named `.env` in the root of your project directory and add your keys like this:

GOOGLE_MAPS_API_KEY="your_actual_google_maps_api_key"
OPENAI_API_KEY="your_actual_openai_api_key"

The application uses `python-dotenv` to load these automatically.

Note: You must have the "Places API" enabled in your Google Cloud project for the Google Maps integration to work.

5. Generate the Dummy Climate Data

The application requires a local NetCDF file for climate data. Run the included script to generate a sample file for testing.

python create_dummy_netcdf.py

This will create the `Durham_AT_RH_JulAug2024.nc` file in your project directory.

How to Run the Application

Once the setup is complete, you can start the FastAPI server using `uvicorn`.

uvicorn main:app --reload

The server will start, and you should see output similar to this in your terminal:

INFO: Uvicorn running on http://127.0.0.1:8000 (Press CTRL+C to quit)

Using the API

You can now send requests to the API.

Example Request

Open your web browser or use a tool like `curl` to make a GET request to the `/analyze_location` endpoint.

Browser URL (using coordinates for Chapel Hill, NC):

http://127.0.0.1:8000/analyze_location?lat=35.9132&lon=-79.0558

Example JSON Response

If successful, you will receive a JSON response structured like this:


```
{
  "infrastructure": {
    "name": "UNC-Chapel Hill",
    "type": "University"
  },
  "climate_data": {
    "air_temperature_celsius": 27.81,
    "relative_humidity_percent": 68.45,
    "last_updated_utc": "2024-08-31T23:00:00"
  },
  "ai_summary": "UNC-Chapel Hill, a university, primarily serves the community by providing a space for its designated activities. With the current relative humidity at 68.5% and temperature at 27.8°C, visitors might experience certain environmental effects. For instance, elevated humidity can increase the perceived temperature, potentially causing discomfort during outdoor activities and affecting sensitive electronic equipment or building materials over time.",
  "category": "University"
}
```


Interactive API Documentation

FastAPI automatically generates interactive API documentation. You can access it at the following URL while the server is running:

- **Swagger UI**: http://127.0.0.1:8000/docs