WeatherApp

WeatherApp is a simple and interactive graphical weather application built using **Python**, **Tkinter**, and the **OpenWeatherMap API**. It leverages the `geopy` library for geocoding, `timezonefinder` for determining time zones, and various built-in Python libraries to fetch and display weather information for a specified city.

Project Overview

The WeatherApp allows users to enter the name of a city and fetches real-time weather data for that location. The app displays details such as temperature, weather conditions, wind speed, humidity, pressure, and a brief description of the current weather. It also shows the current time in the chosen location, formatted based on the respective time zone.

Features

- **Real-time Weather Data**: Fetches current temperature, weather condition, wind speed, humidity level, pressure, and a description of the weather.
- **Geocoding with OpenCage**: Converts city names into geographical coordinates (latitude and longitude).
- **Timezone Awareness**: Automatically fetches and displays the current time for the specified city using its coordinates.
- **User-friendly Interface**: Built using Tkinter, the app provides an intuitive user experience with a clean interface.
- **Custom Image Assets**: Includes custom-designed images for search icons, logos, and layout.

Technologies Used

- **Python**: The core language used to build the app.
- **Tkinter**: A Python GUI library to create the graphical interface.
- **Geopy**: To geocode the city names into latitude and longitude using the OpenCage API.
- **TimezoneFinder**: To determine the timezone based on the city's coordinates.
- **Requests**: To handle API calls to OpenWeatherMap.
- **Pytz**: For managing time zones accurately.

How It Works

1. The user enters a city name into the input box in the app.
2. The app uses the **OpenCage API** to fetch the geographical coordinates of the specified city.
3. The app then determines the city's timezone using the **TimezoneFinder** library.
4. With the obtained coordinates, the app fetches weather information using the **OpenWeatherMap API**.
5. The app displays the current temperature, feels-like temperature, weather condition, wind speed, humidity, pressure, and a brief description of the weather.
6. The app also updates the current local time for the chosen city.

Credits

This project was created with the help of a YouTube tutorial by **Parvat Computer Technology**. Special thanks to them for providing the guidance and resources to build this app!

Screenshot