Group project

Weather forecast

Endrik Meitinger (22-616-759), Paul Spielberger (18-619-098), Roxane Monnier (20-207-924), Yueheng Niu (23-603-194), Moritz Türk (21-615-935)

Description

The purpose of this program is to provide information on current weather conditions and forecasts, as well as some suggestions about the most suitable activities. Designed for easy interaction, it allows the user to input choices and city names to retrieve relevant data from the Weather API. We use an API for the weather data, which can be found at this link: https://www.weatherapi.com/api-explorer.aspx#current

Environment Variable Management

The script starts by importing necessary libraries, including dotenv for accessing environment variables, os for operating system interactions, requests for making API calls, and matplotlib.pyplot for plotting graphs. The .env file stores the API key, and the script loads this key into the token variable using load_dotenv().

API Integration: Two API endpoints are defined: current_url for current weather data and forecast_url for weather forecasts. These URLs are used to fetch weather data from the Weather API.

How does it work ?

The user is first asked which option he would like to choose (enters an integer between 1 and 3):

1. Get current weather
2. Get forecast weather
3. Exit the program

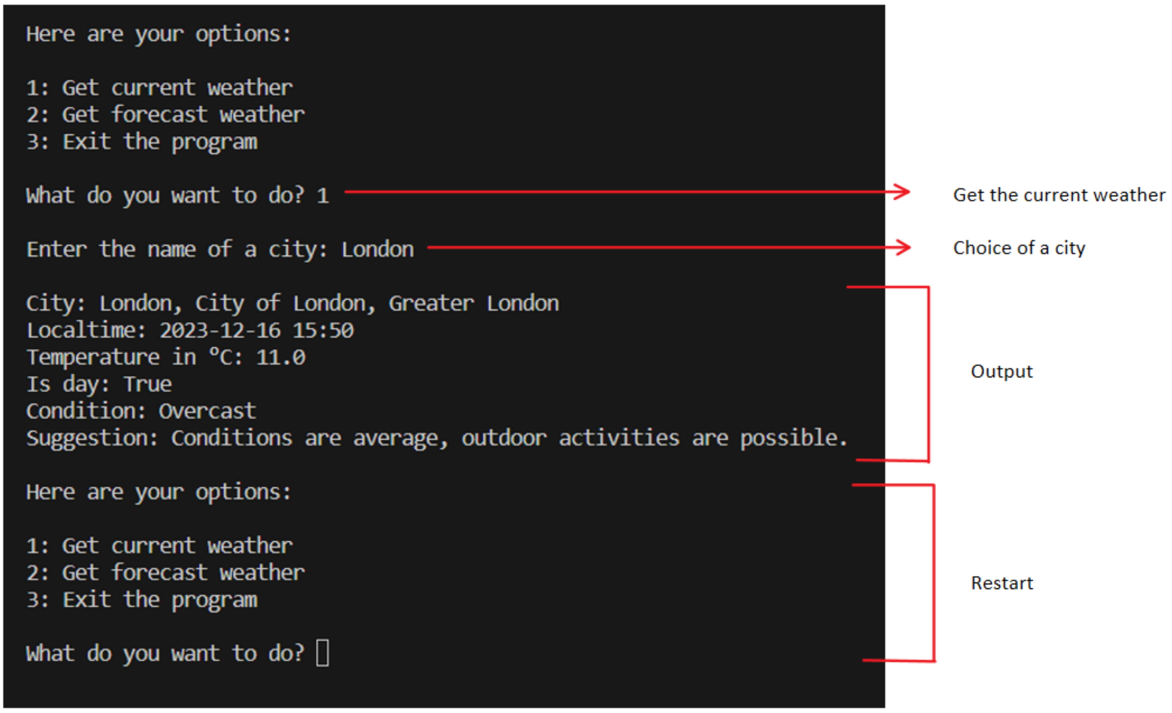
Afterwards, the user enters the city for which he wants weather information. If he chose option 2 in the first step, he would have to enter the number of days for the forecast (the free version of our API limits it to 3 days).

Finally, the user gets all the information he asked for. He can also ask for another option and another city, unless his option is 3, then the program ends.

Current weather

The output shows some essential information about the temperature, the condition, if it is actually day or night.
In addition, it also provides suggestions of activities regarding the actual weather condition (more information in the corresponding section).

Here is an example of output:



```
Here are your options:

1: Get current weather
2: Get forecast weather
3: Exit the program

What do you want to do? 1

Enter the name of a city: London

City: London, City of London, Greater London
Localtime: 2023-12-16 15:50
Temperature in °C: 11.0
Is day: True
Condition: Overcast
Suggestion: Conditions are average, outdoor activities are possible.

Here are your options:

1: Get current weather
2: Get forecast weather
3: Exit the program

What do you want to do? 
```

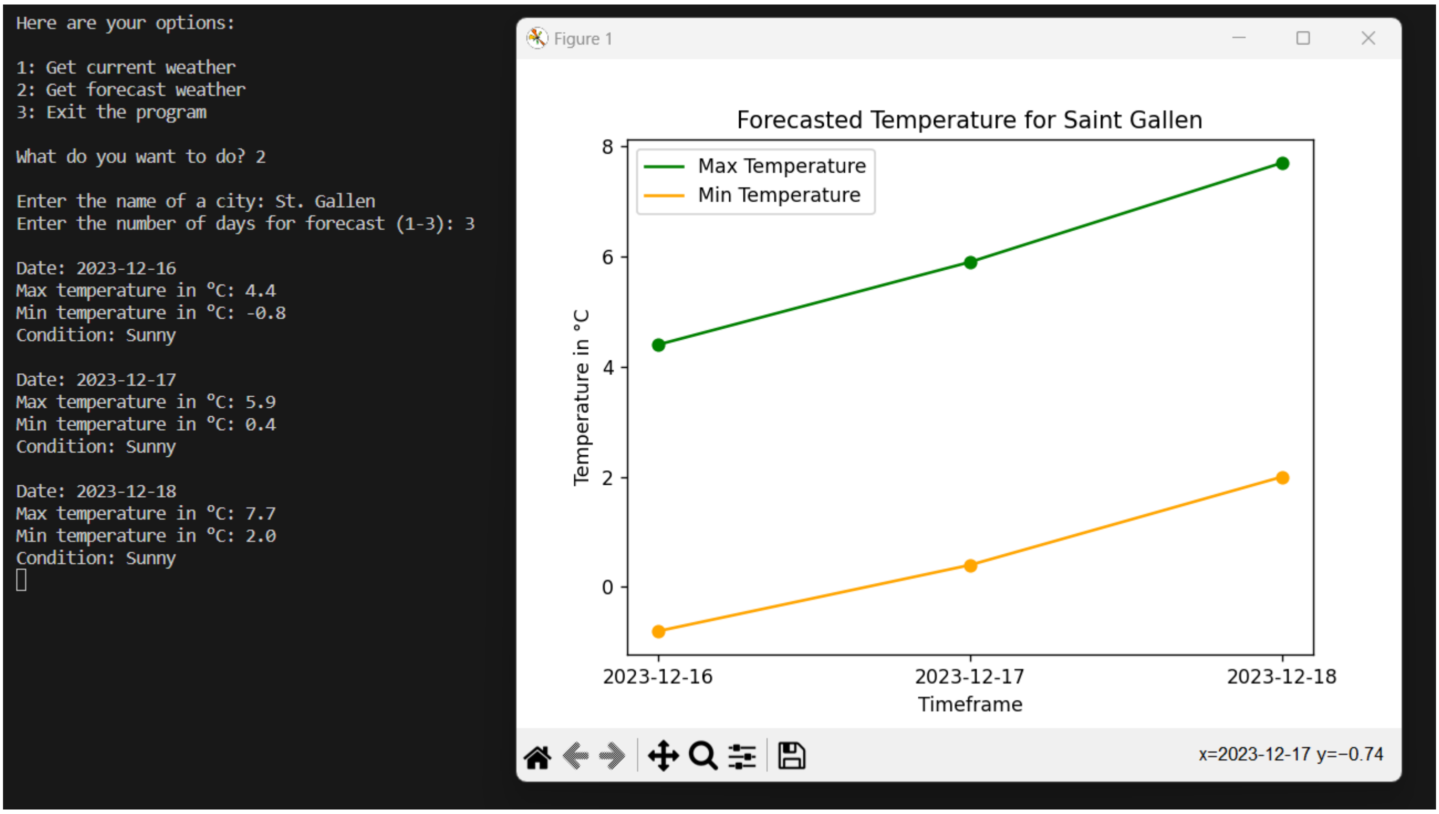
Forecast

If the user picks the 2nd option they are asked to input the city they want to get the weather information for and the number of days the forecast should be long. These inputs are then assigned to the variables ‘q’ for the location and ‘days’ for the number of days selected. Once this has been done the API call is made using the above mentioned forecast_url and the relevant information is retrieved. Before printing the results it is checked if the API call resulted in code 200, which means the call was successful.

Then the retrieved information is printed. For each day of the forecast the date, the maximum and minimum temperature and a short description of the expected weather is printed.

This option also provides graphs of expected temperatures for the next few days. The graph provides information on the maximum and minimum temperatures for every day of the forecast. Additionally, every rainy day is marked with a blue dot.

Here's an example:



```
Here are your options:

1: Get current weather
2: Get forecast weather
3: Exit the program

What do you want to do? 2

Enter the name of a city: St. Gallen
Enter the number of days for forecast (1-3): 3

Date: 2023-12-16
Max temperature in °C: 4.4
Min temperature in °C: -0.8
Condition: Sunny

Date: 2023-12-17
Max temperature in °C: 5.9
Min temperature in °C: 0.4
Condition: Sunny

Date: 2023-12-18
Max temperature in °C: 7.7
Min temperature in °C: 2.0
Condition: Sunny
```

Additional Function

Function: activity_suggestions:

(1) Purpose:

The activity_suggestions function is designed to analyze weather data and provide practical suggestions regarding outdoor activities and car washing. This feature enhances the user experience by offering more than just weather data; it gives actionable advice based on that data.

(2) How It Works:

The function takes a dictionary weather_data containing weather information.
It extracts the current temperature (temp_c), rain condition from the text description (is_raining), and the amount of precipitation (precipitation).

(3) Logic for Suggestions:

- Outdoor Activities: If the temperature is above 20°C and it's not raining, the function suggests that it's a great day for outdoor activities.
- Car Washing: Additionally, if the precipitation is less than 0.1 mm, it suggests that it's also a good day to wash your car, assuming lower chances of rain mean your car will stay clean longer.

- Indoor Activities: If it is raining or the precipitation is more than 0.5 mm, it advises preferring indoor activities, as these conditions are less favorable for outdoor engagements.
- General Conditions: If the conditions don't meet any of the above criteria, it provides a general suggestion that outdoor activities are possible, but the conditions are average.

(4) Handling Exceptions:

The function includes error handling (try-except) for situations where the necessary data might not be present in the weather_data dictionary. This prevents the program from crashing and instead provides a message indicating insufficient data for making a suggestion.

(5) Integration in the Script:

The function is called after fetching and displaying the weather data for both current weather (option == 1) and forecast weather (option == 2).
For the current weather, it analyzes the data received from the API and prints a suggestion immediately.
For the forecast weather, it provides suggestions for each day in the forecast period based on the daily weather data.

(6) User Experience Enhancement:

By providing these suggestions, the program not only informs users about the weather conditions but also helps them plan their activities accordingly. This feature makes the program more interactive and useful in everyday life.