Certification Project

Q1: Create a python script called googlesearch that provides a command line utility to perform google search. It gives you the top links (search results) of whatever you want to search on google.

Q2: Create a script called location that return the location parameters of any location you want.

Q3: Create a script called weather that return the environmental parameters (temperature (min, max), windspeed, humidity, cloud, pressure, sunrise and sunset) of any location you want; after passing arguments (like user api and city id).

Q1: Google Search Script

This script uses the `googlesearch-python` package to perform Google searches from the command line. You can install the package using `pip install googlesearch-python`.

```
[john@squid certification]$ sudo pip install googlesearch-python
Collecting googlesearch-python
  Downloading googlesearch_python-1.2.4-py3-none-any.whl (4.5 kB)
Collecting beautifulsoup4>=4.9
  Downloading beautifulsoup4-4.12.3-py3-none-any.whl (147 kB)
     |████████████████████████████████| 147 kB 627 kB/s
Requirement already satisfied: requests>=2.20 in /usr/lib/python3.9/site-packages (from googlesearch-python) (2.25.1)
Collecting soupsieve>1.2
  Downloading soupsieve-2.5-py3-none-any.whl (36 kB)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/lib/python3.9/site-packages (from requests>=2.20->googlesearch-python) (1.26.5)
Requirement already satisfied: idna<3,>=2.5 in /usr/lib/python3.9/site-packages (from requests>=2.20->googlesearch-python) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in /usr/lib/python3.9/site-packages (from requests>=2.20->googlesearch-python) (4.0.0)
Installing collected packages: soupsieve, beautifulsoup4, googlesearch-python
Successfully installed beautifulsoup4-4.12.3 googlesearch-python-1.2.4 soupsieve-2.5
```

googlesearch.py:

```
import argparse
from googlesearch import search

def google_search(query, num_results):
    for result in search(query, num_results):
        print(result)

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Perform a Google search and return the top links.")
    parser.add_argument('query', type=str, help="The search query.")
    parser.add_argument('--num_results', type=int, default=10, help="Number of search results to return.")

    args = parser.parse_args()
    google_search(args.query, args.num_results)
```

```
[john@squid certification]$
[john@squid certification]$ python3 google_search_script.py openai --num_results 5
https://openai.com/
https://openai.com/
https://en.wikipedia.org/wiki/OpenAI
https://www.youtube.com/openai
https://www.instagram.com/openai/?hl=en
https://www.theguardian.com/business/article/2024/jul/25/openai-search-engine-searchgpt
[john@squid certification]$
```

Q2: Location Script

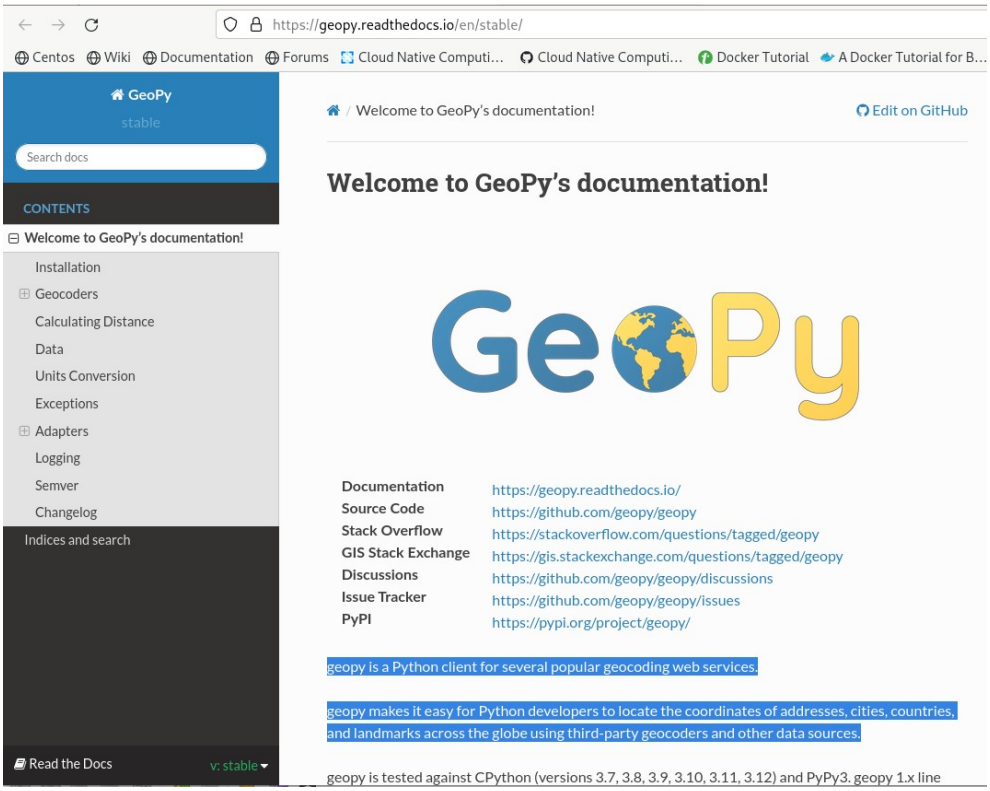
This script uses the `geopy` package to get the location parameters. You can install the package using `pip install geopy`.

```
[john@squid certification]$ sudo pip install geopy
Collecting geopy
  Downloading geopy-2.4.1-py3-none-any.whl (125 kB)
     |████████████████████████████████| 125 kB 736 kB/s
Collecting geographiclib<3,>=1.52
  Downloading geographiclib-2.0-py3-none-any.whl (40 kB)
     |████████████████████████████████| 40 kB 1.7 MB/s
Installing collected packages: geographiclib, geopy
Successfully installed geographiclib-2.0 geopy-2.4.1
```

location.py:

```
import argparse
from geopy.geocoders import Nominatim

# pip install geopy

def get_location(address):
    geolocator = Nominatim(user_agent="location_script")
    location = geolocator.geocode(address)
    if location:
        print(f"Address: {location.address}")
        print(f"Latitude: {location.latitude}")
        print(f"Longitude: {location.longitude}")
    else:
        print("Location not found")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Get the location parameters of an address.")
    parser.add_argument('address', type=str, help="The address to lookup.")

    args = parser.parse_args()
    get_location(args.address)
```

```
[john@squid certification]$ python location.py "New York, USA"
Address: City of New York, New York, United States
Latitude: 40.7127281
Longitude: -74.0060152
```



Q3: Weather Script

This script uses the `requests` package to get weather data from the OpenWeatherMap API. You can install the package using `pip install requests`.

weather.py:

```
import argparse
import requests

def get_weather(api_key, city_id):
    url = f"http://api.openweathermap.org/data/2.5/weather?
id={city_id}&appid={api_key}&units=metric"
    response = requests.get(url)
    data = response.json()

    if data["cod"] != 200:
        print(f"Error: {data['message']}")
        return

    weather = data['main']
    wind = data['wind']
```

```
    clouds = data['clouds']
    sys = data['sys']

    print(f"Temperature: {weather['temp']}°C")
    print(f"Temperature (min): {weather['temp_min']}°C")
    print(f"Temperature (max): {weather['temp_max']}°C")
    print(f"Pressure: {weather['pressure']} hPa")
    print(f"Humidity: {weather['humidity']}%")
    print(f"Wind Speed: {wind['speed']} m/s")
    print(f"Cloudiness: {clouds['all']}%")
    print(f"Sunrise: {sys['sunrise']}")
    print(f"Sunset: {sys['sunset']}")

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Get the weather parameters of a location.")
    parser.add_argument('api_key', type=str, help="Your OpenWeatherMap API key.")
    parser.add_argument('city_id', type=int, help="The city ID.")

    args = parser.parse_args()
    get_weather(args.api_key, args.city_id)
```

python weather.py YOUR_API_KEY CITY_ID

Replace `YOUR_API_KEY` with your actual OpenWeatherMap API key and `CITY_ID` with the city ID of the location you want to get the weather for. You can find city IDs on the OpenWeatherMap website.

Example

If your API key is `abc123` and you want to get the weather for London, you would run:

python weather.py abc123 London

This script will print the current weather details for the specified city, including temperature, wind speed, humidity, cloudiness, pressure, sunrise, and sunset times.