

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.

googlesearch_app.py

```
#!/usr/bin/env python3
import argparse
from ddgs import DDGS    # pip install ddgs

def google_search(query, num_results=10):
    with DDGS() as ddgs:
        return list(ddgs.text(query, max_results=num_results))

def main():
    parser = argparse.ArgumentParser(description="Search from the command line")
    parser.add_argument("query", help="Search query")
    parser.add_argument("-n", "--num", type=int, default=10,
                        help="Number of results (default: 10)")

    args = parser.parse_args()

    results = google_search(args.query, args.num)
    print(f"\nTop {len(results)} results for: {args.query}\n")
    for i, r in enumerate(results, 1):
        print(f"{i}. {r['title']} → {r['href']}")

if __name__ == "__main__":
    main()
```

Output:

```
python googlesearch_app.py "ultrasonic testing" -n 5
```

```
PS C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project> python googlesearch_app.py "ultrasonic testing" -n 5

Top 5 results for: ultrasonic testing

1. Ultrasonic testing → https://en.wikipedia.org/wiki/Ultrasonic\_testing
2. Automated Ultrasonic Testing (AUT): Everything You Need to Know → https://www.linkedin.com/pulse/automated-ultrasonic-testing-aut-everything-you-need-know-beqpf
3. Ultrasonic Testing (UT) - Ocean Corporation → https://oceancorp.com/ultrasonic-testing/
4. Benefits of Ultrasonic Testing - OnestopNDT → https://www.onestopndt.com/ndt-articles/benefits-of-ultrasonic-testing
5. Ultrasonic Testing - UT Inspection Services | MISTRAS United... → https://www.mistrasgroup.com/united-kingdom/how-we-help/field-inspections/traditional-ndt/ultrasonic-testing/
```

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

location.py

```
#!/usr/bin/env python3
"""
weather.py - Command Line Weather Info Tool
"""

import argparse
import requests
from datetime import datetime

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)

    if response.status_code == 200:
        return response.json()
    else:
        raise Exception(f"API request failed with status code {response.status_code}")

def main():
    parser = argparse.ArgumentParser(description="Get weather details of a city by city ID")
    parser.add_argument("--apikey", required=True, help="OpenWeatherMap API key")
    parser.add_argument("--cityid", required=True, help="City ID")
    args = parser.parse_args()

    try:
        data = get_weather(args.apikey, args.cityid)
```

```

city = data["name"]
country = data["sys"]["country"]
temp_min = data["main"]["temp_min"]
temp_max = data["main"]["temp_max"]
pressure = data["main"]["pressure"]
humidity = data["main"]["humidity"]
windspeed = data["wind"]["speed"]
clouds = data["clouds"]["all"]
sunrise =
datetime.utcfromtimestamp(data["sys"]["sunrise"]).strftime('%Y-%m-%d %H:%M:%S')
sunset =
datetime.utcfromtimestamp(data["sys"]["sunset"]).strftime('%Y-%m-%d %H:%M:%S')

print(f"\nWeather details for {city}, {country}\n")
print(f"Temperature : Min {temp_min}C / Max {temp_max}C")
print(f"Humidity      : {humidity}%")
print(f"Pressure       : {pressure} hPa")
print(f"Windspeed      : {windspeed} m/s")
print(f"Cloud Cover    : {clouds}%")
print(f"Sunrise        : {sunrise} UTC")
print(f"Sunset         : {sunset} UTC\n")

except Exception as e:
    print(f"Error: {e}")

if __name__ == "__main__":
    main()

```

Output:

python location.py "Berlin"

```

PS C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project> python location.py "Berlin"

Location details for: Berlin

Address   : Berlin, Deutschland
Latitude  : 52.5173885
Longitude : 13.3951309
PS C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project> python location.py "Mumbai"

Location details for: Mumbai

Address   : Mumbai, Mumbai Suburban, Maharashtra, India
Latitude  : 19.054999
Longitude : 72.8692035

```

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).

weather.py

```
#!/usr/bin/env python3
import argparse
import requests
from datetime import datetime

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

    if response.status_code == 200:
        return response.json()
    else:
        raise Exception(f"API request failed with status code {response.status_code}: {response.text}")

def main():
    parser = argparse.ArgumentParser(description="Get weather details of a city by name")
    parser.add_argument("--apikey", required=True, help="OpenWeatherMap API key")
    parser.add_argument("--city", required=True, help="City name (e.g., Berlin)")
    args = parser.parse_args()

    try:
        data = get_weather(args.apikey, args.city)

        city = data["name"]
        country = data["sys"]["country"]
        temp_min = data["main"]["temp_min"]
        temp_max = data["main"]["temp_max"]
        pressure = data["main"]["pressure"]
        humidity = data["main"]["humidity"]
        windspeed = data["wind"]["speed"]
        clouds = data["clouds"]["all"]
        sunrise =
datetime.utcfromtimestamp(data["sys"]["sunrise"]).strftime('%Y-%m-%d %H:%M:%S')
```

```

        sunset =
datetime.datetime.utcfromtimestamp(data["sys"]["sunset"]).strftime('%Y-%m-%d %H:%M:%S')

    print(f"\nWeather details for {city}, {country}\n")
    print(f"Temperature : Min {temp_min}C / Max {temp_max}C")
    print(f"Humidity      : {humidity}%")
    print(f"Pressure       : {pressure} hPa")
    print(f"Windspeed      : {windspeed} m/s")
    print(f"Cloud Cover    : {clouds}%")
    print(f"Sunrise       : {sunrise} UTC")
    print(f"Sunset        : {sunset} UTC\n")

except Exception as e:
    print(f"Error: {e}")

if __name__ == "__main__":
    main()

```

Output:

python weather.py --apikey a02c1cfd8a65ba4df193213a6aa6d3f3 --city Berlin

```

PS C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project> python weather.py --apikey a02c1cfd8a65ba4df193213a6aa6d3f3 --city Berlin
C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project\weather.py:32: DeprecationWarning: datetime.datetime.utcfromtimestamp(
) is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime
e.fromtimestamp(timestamp, datetime.UTC).
    sunrise = datetime.datetime.utcfromtimestamp(data["sys"]["sunrise"]).strftime('%Y-%m-%d %H:%M:%S')
C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project\weather.py:33: DeprecationWarning: datetime.datetime.utcfromtimestamp(
) is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime
e.fromtimestamp(timestamp, datetime.UTC).
    sunset = datetime.datetime.utcfromtimestamp(data["sys"]["sunset"]).strftime('%Y-%m-%d %H:%M:%S')

Weather details for Berlin, DE

Temperature : Min 9.99C / Max 13.93C
Humidity    : 64%
Pressure    : 1023 hPa
Windspeed   : 2.68 m/s
Cloud Cover : 0%
Sunrise     : 2025-09-28 05:02:19 UTC
Sunset      : 2025-09-28 16:51:36 UTC

```

```
PS C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project> python weather.py --apikey a02c1cfd8a65ba4df193213a6aa6d3f3 --city Mumbai
C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project\weather.py:32: DeprecationWarning: datetime.datetime.utcfromtimestamp(
) is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime
e.fromtimestamp(timestamp, datetime.UTC).
    sunrise = datetime.datetime.utcfromtimestamp(data["sys"]["sunrise"]).strftime('%Y-%m-%d %H:%M:%S')
C:\Users\49176\Desktop\Edureka\Python\Python_intro_edureka\Project\weather.py:33: DeprecationWarning: datetime.datetime.utcfromtimestamp(
) is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime
e.fromtimestamp(timestamp, datetime.UTC).
    sunset = datetime.datetime.utcfromtimestamp(data["sys"]["sunset"]).strftime('%Y-%m-%d %H:%M:%S')

Weather details for Mumbai, IN

Temperature : Min 24.99C / Max 24.99C
Humidity    : 94%
Pressure    : 1004 hPa
Windspeed   : 6.69 m/s
Cloud Cover : 100%
Sunrise     : 2025-09-29 00:58:43 UTC
Sunset      : 2025-09-29 12:59:08 UTC
```