Assignment

CSA0805 – Python Programming

Register Number	**192311290**
Name	**M. Poojith Ganesh**

1. **Title: HTML Page Scraper**: Develop a Python program that scrapes data from HTML web pages using the **`requests`** and **`Beautiful Soup`** modules, extracting specific elements such as links, images, or text, and saving them to a file or database.

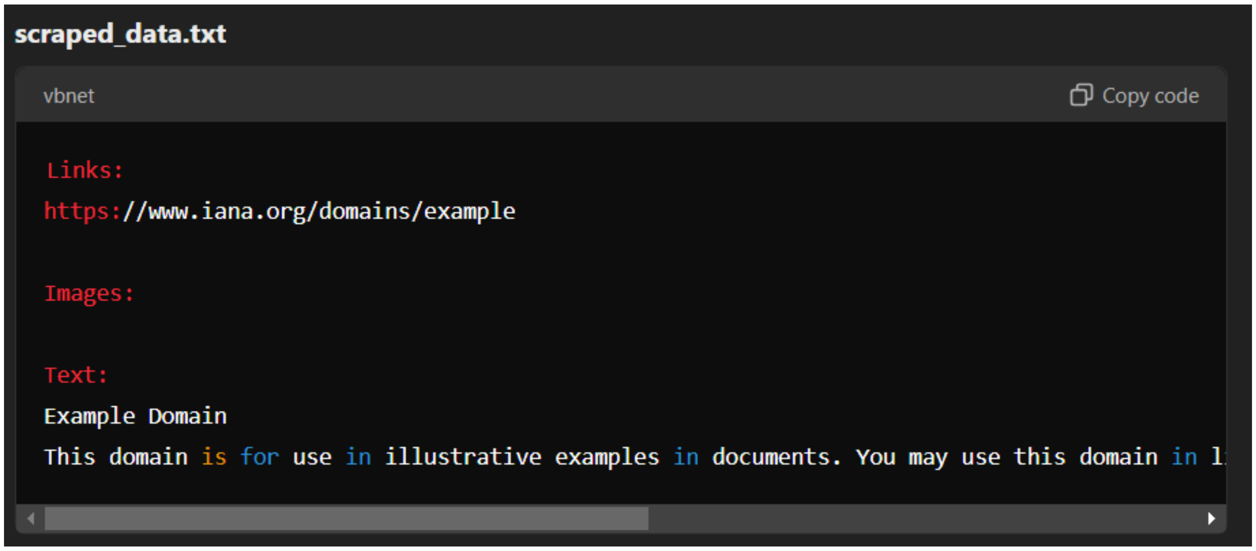
Problem Statement:

1. **Fetch HTML Content**: Use the ‘`requests`’ module to send an HTTP request to a web page URL and retrieve the HTML content of the page.
2. **Parse HTML**: Employ the ‘`Beautiful Soup`’ module to parse the HTML content, allowing easy extraction of specific elements such as links, images, or text.
3. **Extract Data**: Identify and extract the desired elements from the HTML, which might include:
    a. Links (‘`<a>`’ tags with ‘`href`’ attributes)
    b. Images (‘`<img>`’ tags with ‘`src`’ attributes)
    c. Text content from specific tags (e.g., ‘`<p>`’, ‘`<h1>`’, etc.)
4. **Save Data**: Choose a method to save the extracted data. Options include writing to a local file (e.g., a text file or CSV) or inserting the data into a database.
5. **Handle Exceptions**: Implement error handling to manage potential issues such as network errors, invalid URLs, or changes in the HTML structure of the page.

Code:

```
import requests
from bs4 import BeautifulSoup


# Function to scrape a webpage
def scrape_webpage(url):
    try:
        # Send a GET request to the webpage
    response = requests.get(url)
    response.raise_for_status()  # Raise an exception for HTTP errors

    # Parse the HTML content using BeautifulSoup
    soup = BeautifulSoup(response.text, 'html.parser')

    # Extract all links, images, and text
    links = [a['href'] for a in soup.find_all('a', href=True)]
    images = [img['src'] for img in soup.find_all('img', src=True)]
    text = soup.get_text()

    # Save the extracted data to a file
    with open('scraped_data.txt', 'w') as file:
        file.write('Links:\n')
        for link in links:
            file.write(f'{link}\n')

        file.write('\nImages:\n')
        for image in images:
            file.write(f'{image}\n')

        file.write('\nText:\n')
        file.write(text)

    print('Data successfully scraped and saved to scraped_data.txt')

except requests.exceptions.RequestException as e:
    print(f'Error fetching the webpage: {e}')
    except Exception as e:
        print(f'An error occurred: {e}')

# Example usage
url = 'https://example.com'  # Replace with the URL you want to scrape
scrape_webpage(url)
```

Output Screen Shots:



```
Links:
https://www.iana.org/domains/example

Images:

Text:
Example Domain
This domain is for use in illustrative examples in documents. You may use this domain in l
```

Conclusion: Develop a Python program to scrape and extract data from HTML web pages. Using '`requests`' to download the page and `Beautiful Soup` to parse the HTML allows you to easily locate and retrieve specific elements such as links, images, or text. Storing the extracted data in a file or database enables further use and analysis. Implementing error handling ensures that your scraper remains robust and can handle various issues that may arise during the scraping process. This approach provides a solid foundation for web scraping tasks, making it possible to gather and utilize data from web pages effectively.