tests\Meta Tag Test\meta_tags_test.py

```python
#
#  Course: COSC 4P02
#  Assignment: Group Project
#  Group: 9
#  Version: 1.0
#  Date: April 2024
#
import requests # Import the requests module to make HTTP requests.
from bs4 import BeautifulSoup # Import the BeautifulSoup class from the bs4 module.

URL = "https://group9website-gth5dkhfajb4d4g9.canadaeast-01.azurewebsites.net/index.php" # Our website URL to be tested. NOTE: This URL will change before the end of the project, as we will be changing Azure Subscriptions. The tests remain the same, and can be run on the new URL.

def test_open_graph_meta_tags():
    #
    # Test Case: Testing the meta tags are present on the hosted website. This will confirm the meta tags are present and correct.
    # Execution: python -m pytest meta_tags_test.py -k "test_open_graph_tags" -s -v # Only use -s to view the contents of the text in the test.
    # The method will scrape the page source of the website and check for the presence of Open Graph meta tags.
    # Expected Result: Pass. The meta tags are present and correct.
    #
    response = requests.get(URL) # Make a GET request to the specified URL.
    assert response.status_code == 200, "Website is not accessible" # Check if the response status code is 200 (OK).

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

    og_title = soup.find("meta", property="og:title") # Find the Open Graph title meta tag.
    og_description = soup.find("meta", property="og:description") # Find the Open Graph description meta tag.
    og_url = soup.find("meta", property="og:url") # Find the Open Graph URL meta tag.

    assert og_title and og_title.get("content").strip(), "Missing or empty og:title" # Check if the Open Graph title meta tag is present and has content.
    assert og_description and og_description.get("content").strip(), "Missing or empty og:description" # Check if the Open Graph description meta tag is present and has content.
    assert og_url and og_url.get("content").strip(), "Missing or empty og:url" # Check if the Open Graph URL meta tag is present and has content.

    expected_title = "SmartSummaries: AI-Powered Newsletter & Social Media Content Generator" # The expected title for the Open Graph title meta tag.
    expected_description = "Create engaging newsletters and social media posts effortlessly with AI-driven automation, summarization, and scheduling." # The expected description for the Open Graph description meta tag.
    expected_url = "https://group9website-gth5dkhfajb4d4g9.canadaeast-01.azurewebsites.net/index.php" # The expected URL for the Open Graph URL meta tag.
    actual_title = og_title.get("content").strip() # Get the content of the Open Graph title meta tag.
    actual_description = og_description.get("content").strip() # Get the content of the Open Graph description meta tag.
    actual_url = og_url.get("content").strip() # Get the content of the Open Graph URL meta tag.

    assert actual_title == expected_title, f"og:title does not match. Expected: '{expected_title}', Got: '{actual_title}'"
    assert actual_description == expected_description, f"og:description does not match. Expected: '{expected_description}', Got: '{actual_description}'"
    assert actual_url == expected_url, f"og:url does not match. Expected: '{expected_url}', Got: '{actual_url}'"

    print(f"\nOpen Graph Meta Tags Found:") # Print a message indicating that the Open Graph meta tags were found.
    print(f"Title: {og_title.get('content')}") # Print the content of the Open Graph title meta tag.
    print(f"Description: {og_description.get('content')}") # Print the content of the Open Graph description meta tag.
    print(f"URL: {og_url.get('content')}") # Print the content of the Open Graph URL meta tag.


```