Selenium WebDriver Assignment

Automating a Google Search Using Selenium and Unittest

General Requirements:

- Use **Selenium WebDriver** with **Python** and the `unittest` framework.
- Your code should follow best practices for test automation, including the use of setup and teardown methods.
- All elements should be located using appropriate locator strategies (e.g., `By.NAME`, `By.XPATH`, `By.CSS_SELECTOR`).
- Your script must be written in a modular and readable format.
- Include meaningful comments in your code to explain your logic.

Specific Requirements:

Functional Test Flow:

1. Launch a browser and open `https://www.google.com`.
2. Locate the search input field.
3. Enter the text `"Selenium WebDriver"` and perform the search.
4. Wait for the results page to load.
5. Verify that the search term appears in the **page title**.
6. Take a screenshot of the results page and save it as `search_results.png`.
7. Click on the **first search result** (typically identified by an `<h3>` element).
8. Verify that the title of the newly opened page **does not contain** the word `"Google"`.

Optional Challenge (Bonus):

- Modify your script to run in **headless mode**, meaning the browser window will not open during test execution.

Submission Details:

- Submit a single Python file named: `test_google_search.py`.
- Ensure the file runs successfully using `unittest` (i.e., it should be executable via `python test_google_search.py`).
- Include the generated `search_results.png` in your submission.
- Submit your work to dc connect before the due date.