Take-Home Assignment: Test Automation Challenge

Test Case 1: Verify that the base page can be accessed

- **Objective:** Ensure that the target page for testing is working and can be accessed.
- **Expected Results:** The page returns status code 200 and the main components are present.
- **Reason:** To ensure that the page is working. If it is not working, there is no reason to test the rest of the cases.
- **Test Steps:**
    1. Send request to the URL: https://nocnoc.com/pl/All?area=search&st=%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8%AD%E0%B8%87%E0%B8%97%E0%B8%B3%E0%B8%99%E0%B9%89%E0%B8%B3%E0%B8%AD%E0%B8%B8%E0%B9%88%E0%B8%99
    2. Verify the status code.
    3. Locate the main components
        - NocNoc logo
        - Search bar
        - Product list container
        - Login button
    4. Validate the results.

Test Case 2: Validate the presence of SEO related elements

- **Objective:** Ensure SEO-related elements (meta tags, title, headings, etc.) are correctly rendered and optimized on the page after applying the filtering criteria.
- **Expected Results:** SEO-related elements such as the meta description, page title, and heading tags should be present.
- **Reason:** Ensure that the page gets good ranking by search engines, improving visibility in search results.
- **Test Steps:**
    1. Locate the SEO-related elements
    2. Verify their existence.

Test Case 3: Validate infinite scrolling functionality

- **Objective:** Ensure infinite scrolling, or dynamic product list population, is working.
- **Expected Results:** Scrolling correctly load more part of the page and the products are added (More children in the product list every scroll).
- **Reason:** If this function doesn't work, users may miss out on products that match their interests, resulting in the loss of customers.
- **Test Steps:**
  1. Let the page load the initial set of product.
  2. Scroll to the bottom.
  3. Let the page load more products.
  4. Validate the results.
  5. Repeat 1-4 five times.

Test Case 4: Validate Sorting by Price

- **Objective:** Ensure sorting results by ascending / descending price orders the products correctly.
- **Expected Results:** Products are sorted in the correct order based on the selected sorting preference.
- **Reason:** Sorting helps users prioritize results based on their preferences.
- **Test Steps:**
  1. Let the page load the initial set of product.
  2. Select price sorting option based on {{***Test input***}}.
  3. Observe the updated order of results.
  4. Validate the results.
  5. Repeat 1-4 until every {{***Test input***}} cases are covered.

Test input	Expected output
ราคา: จากน้อย-มาก	The page displays products ordered by price (low to high).
ราคา: จากมาก-น้อย	The page displays products ordered by price (high to low).

Test Case 5: Validate Sorting by New Arrival

- **Objective:** Ensure sorting results by arrival date orders the products correctly from new to old.
- **Expected Results:** Products are sorted in the correct order based on arrival date.
- **Reason:** Sorting helps users prioritize results based on their preferences.
- **Test Steps:**
  1. Let the page load the initial set of product.
  2. Select the ***arrival date*** sorting option
  3. Observe the updated order of results.
  4. Validate the results.

Test Case 6: Validate Sorting by Amount sold

- **Objective:** Ensure sorting results by amount sold orders the products correctly from the most to the least sold.
- **Expected Results:** Products are sorted in the correct order based on amount sold.
- **Reason:** Sorting helps users prioritize results based on their preferences.
- **Test Steps:**
    1. Let the page load the initial set of product.
    2. Select the ***amount sold*** sorting option
    3. Observe the updated order of results.
    4. Validate the results.

Test Case 7: Validate Brand Filter functionality

- **Objective:** Ensure filtering results by brand displays the list of products correctly.
- **Expected Results:** Products are filtered correctly based on selected choice.
- **Reason:** Filtering helps users prioritize results based on their preferences.
- **Test Steps:**
    1. Locate the filtering chekbox brand dropdown menu on the left side of the page.
    2. Select each cases specified in the {{***Test Input}}***.
    3. Observe the updated results.
    4. Validate the results.
    5. Repeat 1-4 until the last {{***Test Input}}***.

Test input	Expected output
Filter “**ยี่ห้อ**”	The page displays only the products that match the selected brand.
Filter multiple “**ยี่ห้อ**”	The page displays only the products that match the selected brands.

Test Case 8: Validate Price range Filter Functionality

- **Objective:** Ensure filtering results by the price range option displays the list of products correctly.
- **Expected Results:** Products are filtered correctly based on selected price range.
- **Reason:** Filtering helps users prioritize results based on their preferences.
- **Test Steps:**
  1. Locate the price range filter menu on the left side of the page.
  2. Test inputting invalid bounds specified in the {{***Test Input***}}.
  3. Input the valid range specified in the {{***Test Input***}}.
  4. Observe the updated results.
  5. Validate the results.

Test input	Expected output
(input test) {{**Lower bound - 100**}} in min "ราคา"	Input automatically changes to the lowest available price of all products
(input test) {{**Upper bound + 100**}} in max "ราคา"	Input automatically changes to the highest available price of all products
(search test) Filter "ราคา" by range Q1 – Q3	The page displays only the products that match the selected price range.

Test Case 9: Validate the Mobile application download link

- **Objective:** Verify that the links to the iOS and Android apps are functional.
- **Expected Results:** All the links respond with status code 200.
- **Reason:** Mobile app links enhance user accessibility. If the links are broken, accessibility is compromised, leading to a poor user experience.
- **Test Steps:**
  1. Locate links at the top part of the page.
  2. Send request to the link.
  3. Validate the results.

Test Case 10: Validate Language button functionality

- **Objective:** Ensure that the TH / EN language button is working correctly.
- **Expected Results:** The button works correctly.
- **Reason:** The language button functionality is crucial for ensuring that users can easily navigate and interact with the site in their preferred language.
- **Test Steps:**
    1. Locate the language button on the top-right part of the page.
    2. Click the button and select the opposite country.
    3. Validate the results.