Scania Assignment

Introduction:

In this assignment, the goal is to automate test cases for www.scania.com, a prominent website, using Selenium as the primary automation tool. The task involves identifying functional flows, automating them, and providing a comprehensive report. This introductory section will outline the approach, tool selection, and reasons for choosing Selenium.

Task Explanation:

The task involves the following key steps:

- Open www.scania.com home page.
- Identify 2 test cases, mostly concerned with functional flow.
- Automate them with any automation tool of your choice.
- Write a detailed report outlining the testing approach, logic, challenges faced, and the methodology used.

Tool Selected:

Research on Testing Tools

Before diving into the automation process, a thorough examination of various testing tools was conducted. This exploration encompassed tools requiring coding for testing, such as Selenium, Appium, Playwright. Katalon, StackBrowser, Cypress, TestCafe, UiPath, and those that don't require any coding.

Selection Criteria:

Among the plethora of testing tools, Selenium and Playwright emerged as the top contenders based on the following criteria:

Community Support:

Both Selenium and Playwright boast large and active communities. This ensures a wealth of resources, forums, and documentation for problem-solving and continuous improvement.

Open Source:

Selenium and Playwright are open-source tools, promoting accessibility and flexibility. Open-source tools typically have a broader user base and continuous updates.

Cross-Browser Compatibility:

Selenium and Playwright support multiple browsers, ensuring comprehensive test coverage across different platforms.

Multi-Language Support:

Selenium and Playwright support multiple programming languages, allowing teams to choose the language they are most comfortable with or the one that aligns with the project's tech stack.

Choice of Selenium:

While both Selenium and Playwright presented compelling features, Selenium was ultimately selected as the primary tool for this assignment due to the following reasons:

Established Ecosystem:

Selenium has been an industry-standard for web automation, with a well-established ecosystem and a vast user base.

Mature and Stable:

Selenium has undergone extensive development and testing over the years, making it a mature and stable choice for automation projects.

Wide Adoption:

Selenium is widely adopted across industries, and its compatibility with various programming languages and browsers makes it suitable for diverse projects.

Abundant Learning Resources:

The abundance of tutorials, documentation, and online courses makes it easier for team members to learn and apply Selenium effectively.

In conclusion, Selenium was chosen as the primary automation tool for its open-source nature, community support, cross-browser compatibility, multi-language support, and well-established reputation in the field of web automation.

Testing Approach:

Installation:

Selenium Installation:

Installed Selenium using the pip install selenium command.

Ensured the compatibility of the installed Selenium version with the chosen WebDriver (ChromeDriver in this case).

WebDriver Setup:

1)Downloaded the appropriate version of ChromeDriver.

2)Configured the ChromeDriver executable in the system's PATH variable.

We can also use webdriver manager instead.

With WebDriverManager, you no longer need to manually download and configure the WebDriver executable. It dynamically fetches the appropriate WebDriver binary based on the browser and version specified in your code. This simplifies the setup process and ensures compatibility.

Configuration:

Browser Configuration:

Maximized the browser window for better visibility and consistent testing conditions.

Using command driver.maximize()

Automation:

Introduction:

To establish a comprehensive testing approach for the Scania website, I initiated the testing process with exploratory testing to gain a fundamental understanding of its basic functionalities. This initial phase allowed me to navigate through the website, identify key features, and observe user interactions. The insights gained from exploratory testing served as a foundation for crafting meaningful and effective test cases.

Exploratory Testing:

During exploratory testing, I systematically explored various sections of the Scania website, engaging with its interface, and scrutinizing the behavior of different elements. This phase was crucial in uncovering potential areas of interest and understanding the overall user experience.

Test Case Selection:

Upon acquiring a solid understanding of the website's functionality, I transitioned to the test case development phase. To ensure the creation of valuable and impactful test cases, I focused on identifying critical user journeys and scenarios. Two test cases were meticulously chosen to reflect functional flows that are integral to the user experience on the Scania website.

Test Cases:

1. **Test Case: Purchase Flow for a Gift in the Scania Lifestyle Webshop:**

   Objective:
   To verify that a user can successfully navigate to the Scania Lifestyle Webshop, select a gift category, choose a specific gift, and add it to the shopping cart. The test will also check if the product page displays correctly and if the selected item is added to the cart successfully.

   Steps:
   Open the www.scania.com website.
   Hover the mouse on the plus icon at the top right corner.
   Click on the "Scania Lifestyle Webshop" link.
   On the shopping page, click on the "Gifting" text link.

Select the category "For the Little Scania Fan."
Choose the first item from the list of gifts.
Verify that the product page for the selected item is displayed.
Add the selected item to the shopping cart.
Open the shopping cart and verify that the selected item is present.

Expected Results:
The product page for the selected item should display correctly.
The selected item should be added to the shopping cart.
The shopping cart should contain the selected item.

2. **Test Case: Apply for a Temporary IT Position - I-Talent Program**

Objective:
To validate the process of applying for a temporary IT position, specifically for the "I-Talent Program," on the www.scania.com website. The test includes navigating through career-related pages, selecting job criteria, verifying the job description page, and completing the application process.

Steps:
Open the www.scania.com website.
Click on the "Accept" button for cookies.
Click on the "Career" text link.
Click on the "Available Positions" link.
Click on the "Employment Type" dropdown and select "Temporary."
Click on the "Job Category" dropdown and select "IT."
From the search results, select the job titled "I-Talent Program."
Verify that the job description page for "I-Talent Program" is available.
Click on the "Apply" button.
If the application process is not completed, go back to the job description page.
Click on the text link "Directly Apply Here.

Expected Results:
The job description page for the "I-Talent Program" should be available.
The application process should be completed successfully.

Script Development Approach:

In developing these scripts, I utilized the Selenium framework, a powerful tool for automating web applications. The scripts are written in Python, leveraging the Selenium WebDriver library for interaction with web elements**.**

Key features of the scripts include:

- Locator Strategies: Employing various locator strategies (XPath, Link Text, etc.) to identify and interact with specific elements on the webpage.
- Explicit Waits: Using explicit waits (e.g., WebDriverWait) to ensure that the script interacts with elements only when they are ready, enhancing script reliability.

- User Actions: Simulating user interactions such as clicks, dropdown selections, and form submissions to replicate real user behavior.

Script Execution and Verification:

The scripts can be executed in any Selenium-compatible environment. They undergo thorough testing to validate their effectiveness and reliability in replicating the specified test cases. Additionally, the scripts include verification steps to ensure that the expected outcomes align with the actual behavior of the website.

Challenges Faced:

Element Interactions:

The website may use dynamic elements that require careful handling to avoid ElementNotInteractableException. Employed WebDriverWait to mitigate this.

Browser Compatibility:

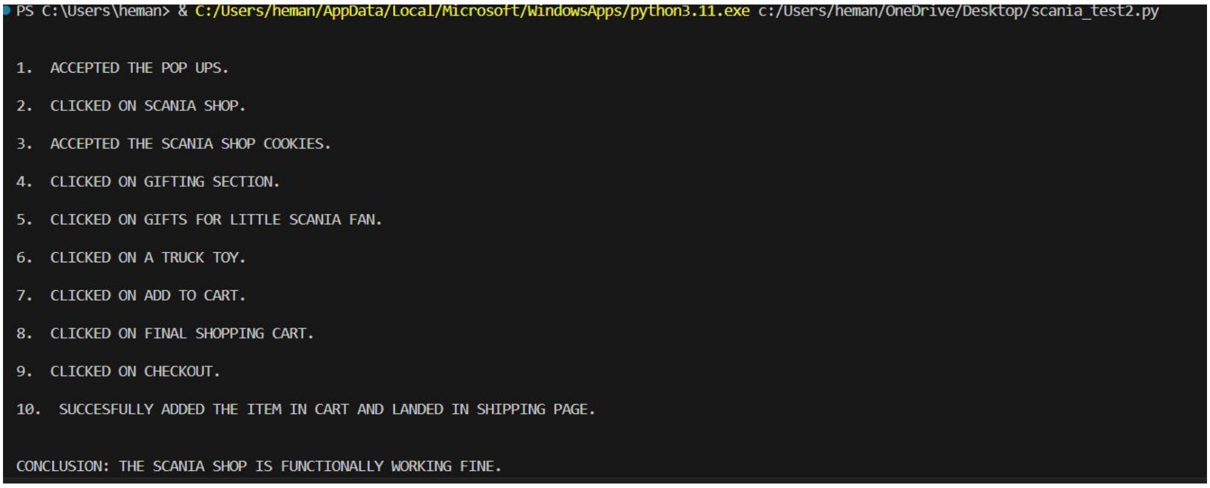
Ensured compatibility with different browser versions by using the appropriate WebDriver version.

Handling Asynchronous Behavior:

Dealt with asynchronous loading of elements by using explicit waits (WebDriverWait).

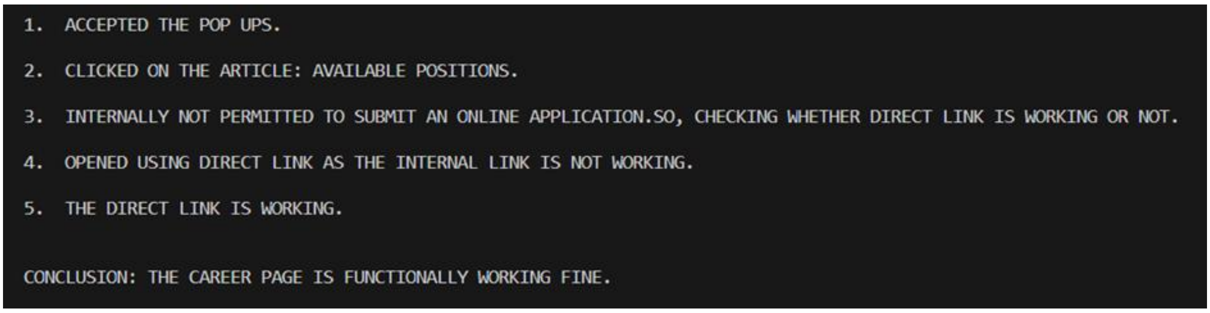
OUTPUT:

1. **Test Case: Purchase Flow for a Gift in the Scania Lifestyle Webshop:**



```
PS C:\Users\heman> & C:/Users/heman/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/heman/OneDrive/Desktop/scania_test2.py

1.  ACCEPTED THE POP UPS.

2.  CLICKED ON SCANIA SHOP.

3.  ACCEPTED THE SCANIA SHOP COOKIES.

4.  CLICKED ON GIFTING SECTION.

5.  CLICKED ON GIFTS FOR LITTLE SCANIA FAN.

6.  CLICKED ON A TRUCK TOY.

7.  CLICKED ON ADD TO CART.

8.  CLICKED ON FINAL SHOPPING CART.

9.  CLICKED ON CHECKOUT.

10.  SUCCESFULLY ADDED THE ITEM IN CART AND LANDED IN SHIPPING PAGE.

CONCLUSION: THE SCANIA SHOP IS FUNCTIONALLY WORKING FINE.
```

2. **Test Case: Apply for a Temporary IT Position - I-Talent Program:**



```
1.  ACCEPTED THE POP UPS.

2.  CLICKED ON THE ARTICLE: AVAILABLE POSITIONS.

3.  INTERNALLY NOT PERMITTED TO SUBMIT AN ONLINE APPLICATION.SO, CHECKING WHETHER DIRECT LINK IS WORKING OR NOT.

4.  OPENED USING DIRECT LINK AS THE INTERNAL LINK IS NOT WORKING.

5.  THE DIRECT LINK IS WORKING.

CONCLUSION: THE CAREER PAGE IS FUNCTIONALLY WORKING FINE.
```

Conclusion:

In undertaking the testing assignment for the www.scania.com website, the process began with a thorough phase of exploratory testing. This initial exploration allowed for a deep dive into the website's functionalities, helping to comprehend user interactions and identify potential areas of interest. The insights gained from this phase laid the groundwork for the creation of meaningful and effective test cases.

The two chosen test cases exemplify functional flows crucial to the user experience on the Scania website. The first test case revolves around the seamless purchase flow for a gift from the Scania Lifestyle Webshop, ensuring a smooth user journey from selection to cart verification. The second test case focuses on the application process for a temporary IT position, specifically the "I-Talent Program." This scenario navigates through the career section, filters positions, verifies job descriptions, and initiates the application process, with a fallback mechanism if needed.

For the implementation of these test cases, Selenium WebDriver was employed as the automation tool, leveraging Python as the scripting language. The Selenium scripts mimic user actions, interact with web elements, and verify expected outcomes. Key features, such as locator strategies, explicit waits, and simulated user actions, were incorporated to ensure the robustness and reliability of the scripts.

The code source file for these Selenium scripts is made available on GitHub, attached to this document. This allows for transparency and accessibility, facilitating collaboration and future reference.

In summary, this testing approach integrates exploratory testing insights with the automation capabilities of Selenium, providing a comprehensive strategy for validating the critical functionalities of the Scania website. By continuously refining and expanding the suite of test cases, we aim to contribute to the improvement of the website's quality, ensuring a seamless and delightful experience for users interacting with the Scania platform. This testing process aligns with best practices in quality assurance, emphasizing thorough coverage, repeatability, and adaptability to evolving web environments.