Experiment – 9

Aim: Automating web applications for testing purposes Using Selenium.

Tools: Selenium

Learning Objective:

- **Understand Selenium Architecture**:
  - Explain the components of Selenium (WebDriver, IDE, Grid) and how they interact.
- **Set Up Selenium Environment**:
  - Install and configure Selenium WebDriver and the necessary browsers for testing.
- **Write Basic Test Scripts**:
  - Create and execute simple test scripts using Selenium WebDriver in a chosen programming language (e.g., Java, Python).
- **Locate Web Elements**:
  - Use various strategies (ID, name, XPath, CSS selectors) to identify and interact with web elements.
- **Perform User Interactions**:
  - Simulate user actions such as clicking, typing, and selecting from dropdowns.

Theory:

What is Selenium?

Selenium is an open-source suite of tools designed for automating web browsers. It allows developers and testers to write scripts in various programming languages to interact with web applications, simulating user behavior for testing purposes.

Key Components of Selenium

1. **Selenium WebDriver**:
   - The core component that allows for direct communication with browsers.
   - Supports multiple programming languages (Java, Python, C#, Ruby, etc.).
   - Interacts with the browser via its native support (no need for a separate server).
2. **Selenium IDE**:
   - A browser extension for recording and playback of tests.

Useful for beginners to create simple test cases without programming knowledge.

3. **Selenium Grid**:
    - A tool for running tests on multiple machines and browsers in parallel.
    - Facilitates cross-browser testing by distributing tests across different environments.

How Selenium Works

- **Browser Interaction**: Selenium uses WebDriver to send commands to the browser and retrieve results. It mimics user actions like clicking, entering text, and navigating pages.
- **Element Identification**: Elements on a web page can be located using various strategies:
    - **ID**: The most efficient method for locating elements.
    - **Name**: Useful for form inputs.
    - **XPath**: Allows for complex queries to locate elements based on their XML path.
    - **CSS Selectors**: A powerful method to select elements using CSS syntax.

Advantages of Using Selenium

- **Open Source**: No licensing fees, which makes it accessible.
- **Cross-Platform Compatibility**: Works on different operating systems (Windows, macOS, Linux).
- **Supports Multiple Browsers**: Compatible with major browsers like Chrome, Firefox, Safari, and Edge.
- **Integration**: Can be integrated with testing frameworks (JUnit, TestNG) and CI/CD tools (Jenkins, GitLab).

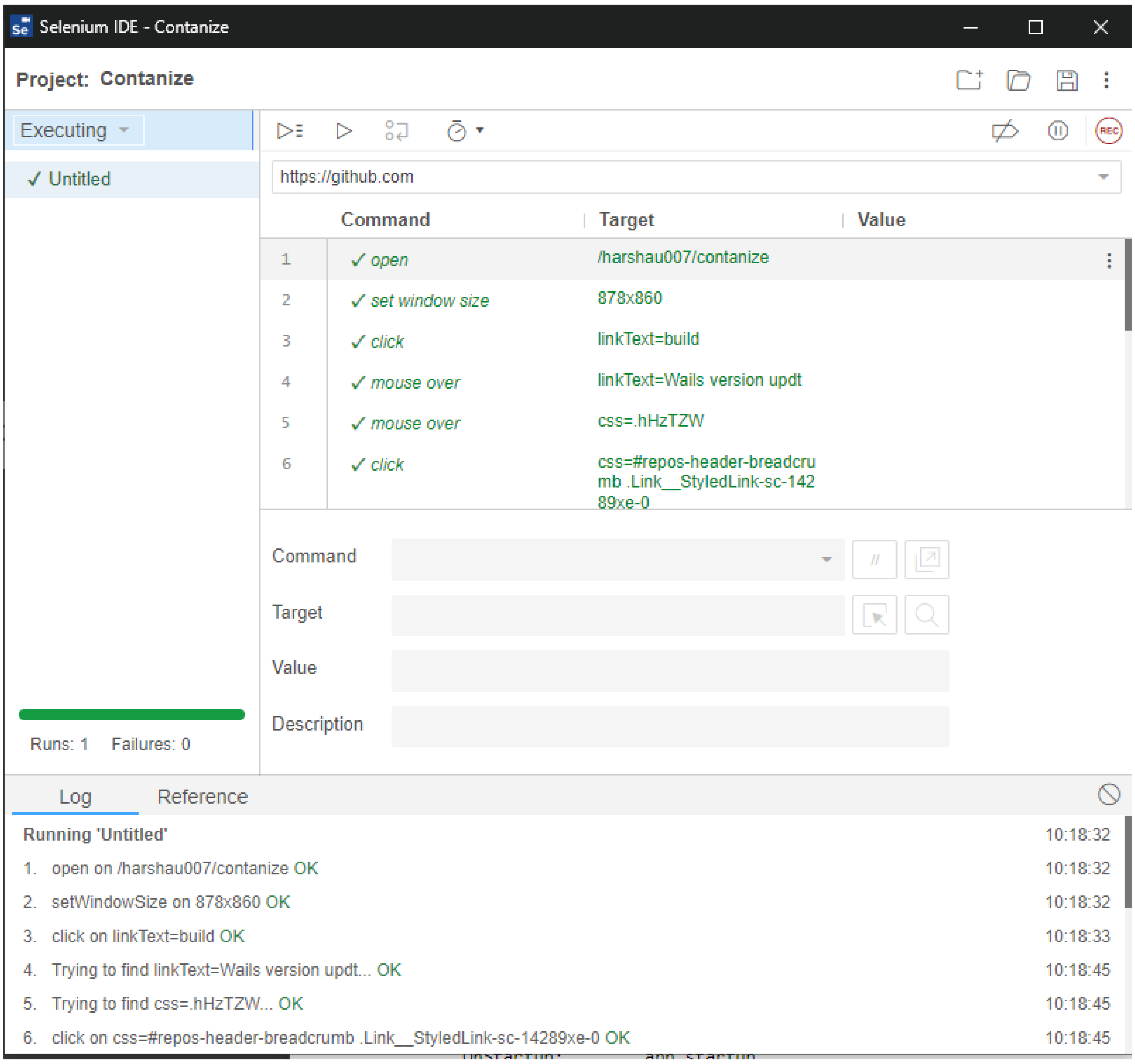
Challenges and Considerations

- **Dynamic Content**: Web applications often use dynamic content (e.g., AJAX), which can lead to timing issues.
- **Maintenance**: Test scripts may require frequent updates due to changes in the web application’s UI.
- **Complex Scenarios**: Handling advanced user interactions (drag-and-drop, file uploads) may require additional effort.

Best Practices

- **Use Page Object Model (POM)**: To enhance maintainability by separating test logic from UI element locators.
- **Implement Explicit Waits**: To deal with dynamic elements effectively and avoid race conditions.

Outputs:


Selenium IDE - Contanize

Project: Contanize

Executing

✓ Untitled

https://github.com

	Command	Target	Value
1	✓ open	/harshau007/contanize	
2	✓ set window size	878x860	
3	✓ click	linkText=build	
4	✓ mouse over	linkText=Wails version updt	
5	✓ mouse over	css=.hHzTZW	
6	✓ click	css=#repos-header-breadcrumb .Link__StyledLink-sc-14289xe-0	

Runs: 1 Failures: 0

```
Running 'Untitled'                                                          10:18:32
1. open on /harshau007/contanize OK                                         10:18:32
2. setWindowSize on 878x860 OK                                              10:18:32
3. click on linkText=build OK                                               10:18:33
4. Trying to find linkText=Wails version updt... OK                         10:18:45
5. Trying to find css=.hHzTZW... OK                                         10:18:45
6. click on css=#repos-header-breadcrumb .Link__StyledLink-sc-14289xe-0 OK  10:18:45
```

Learning Objective:

- **Comprehension of Selenium Architecture**:
  - Describe the components of Selenium and their respective roles in web automation.
- **Proficiency in Environment Setup**:
  - Successfully install and configure Selenium WebDriver and necessary browsers, demonstrating readiness for test automation.
- **Script Development Skills**:

- Write and execute basic Selenium test scripts in a selected programming language, showcasing an understanding of syntax and functionality.

- **Element Identification Techniques**:
  - Identify and interact with web elements using various locator strategies (ID, name, XPath, CSS selectors) effectively.

- **User Interaction Simulation**:
  - Simulate realistic user interactions (clicks, inputs, selections) within web applications using Selenium.

- **Wait Mechanism Implementation**:
  - Apply implicit and explicit waits to manage dynamic web elements, ensuring test stability.

- **Conclusion:-**

In conclusion, mastering Selenium equips individuals with essential skills for automated web application testing. By understanding its architecture and components, learners can effectively set up their testing environments and write robust scripts that simulate real user interactions. Proficiency in element identification, user action simulation, and handling dynamic content prepares testers to tackle the challenges posed by modern web applications.

By applying best practices, such as the Page Object Model and effective wait strategies, testers can create maintainable and reliable test suites that adapt to changes in the application. Overall, the skills gained through studying Selenium not only improve the quality of software products but also contribute to a more streamlined development process, ultimately enhancing user satisfaction and driving business success.

Correction Parameters	**Formative Assessment [40%]**	**Timely completion of Practical [40%]**	**Attendance / Learning Attitude [20%]**	
Marks Obtained				