EAST WEST UNIVERSITY

Department of Computer Science and Engineering

CSE430 - Software Testing and Quality Assurance

Lab Topic: Selenium IDE

1. Objective

This laboratory introduces the use of Selenium IDE, a browser-based test automation tool, for recording and replaying web application interactions. Students will learn to design simple GUI test scripts, insert assertions and waits, and export tests for further automation using Selenium WebDriver.

2. Learning Outcomes

1. Install and configure Selenium IDE as a browser extension.
2. Record and replay user interactions with a website.
3. Insert and modify assertions, verifications, and wait commands.
4. Automate test cases using variables and simple control flow.
5. Export Selenium IDE tests for later use in WebDriver (Lab 7).

3. Requirements

- Mozilla Firefox browser (latest version).
- Selenium IDE browser extension.

- Internet connectivity to access demo testing websites.

4. Background Theory

Selenium IDE is a record-and-playback tool that allows testers to create automated test cases directly from browser interactions. It is widely used for rapid prototyping of test scripts without requiring programming skills.

Each Selenium IDE script consists of commands, targets, and values:

- Command: The operation to perform (e.g., open, click, assertText).
- Target: The web element locator (e.g., id=username, css=h1.title).
- Value: Optional data for input fields or variable assignments.

Assertions confirm expected behaviors, while waits help synchronize tests with dynamic web content.

5. Step-by-Step Procedure

Step 1: Install Selenium IDE

1. Open Firefox and navigate to
   🔗 https://addons.mozilla.org/en-US/firefox/addon/selenium-ide/.
2. Click Add to Firefox → confirm installation.
3. The Selenium IDE icon will appear in the top-right toolbar.
4. If installation is blocked:
    - In the address bar, type about:config.
    - Accept risk and continue.
    - Search for extensions.blocklist.enabled → double-click → set to false

Step 2: Create a New Project

1. Click the Selenium IDE icon.
2. Select Create a New Project → name it *CSE430_Lab6*.
3. Click OK to open the IDE workspace.

Step 3: Start Recording

1. In the project window, choose Record a new test in a new project.
2. Enter the website URL, e.g., https://www.google.com, and click Start Recording.
3. A new tab opens — every user action is recorded automatically (typing, clicking, navigation)

Step 4: Interact with the Website

Example (Google Search):

- Click on the search box.
- Type *Selenium IDE tutorial*.
- Press Enter or click Search.
- Optionally, open a result.

Step 5: Stop and Save

- Return to Selenium IDE → click Stop Recording.

- All actions appear in a command list (table view).
- Click Save Project (three-dot menu) to keep the .side file for reuse

6. Executing and Validating Tests

1. Click Run Current Test ▸ to replay the actions.
2. Observe the Pass/Fail status beside each step.
3. Review the log to debug failing commands.
4. Use assert or verify commands for validation:
    - assertTitle – verifies window title.
    - assertText – checks visible content.
    - verifyElementPresent – confirms presence of an element.

7. Example Script – Google Search

Step	Command	Target	Value
1	open	https://www.google.com	
2	type	name=q	Selenium IDE
3	sendKeys	name=q	${KEY_ENTER}
4	waitForElementVisible	css=h3	5000
5	assertText	css=h3	selenium.dev

Expected Result: **The test passes if search results containing "selenium.dev" appear.**

8. Practice Lab Tasks

Perform the following using Selenium IDE (Firefox):

1. **Wikipedia Search Test**
    - Open https://www.wikipedia.org/.
    - Type "Alan Turing" and click Search.
    - Use assertTitle to verify "Alan Turing - Wikipedia".

2. **Dropdown Test on Booking.com**
    - Go to https://www.booking.com/.
    - Search “Dhaka” → wait for suggestions → select → assert selection.
3. **Product Add-to-Cart Test**
    - Open https://advantageonlineshopping.com.
    - Select category → choose product → click “Add to Cart”.
    - Verify confirmation message or cart count update
      .
4. **Negative Login Validation**
    - Visit https://the-internet.herokuapp.com/login.
    - Enter invalid credentials.
    - Assert error message: “Your username is invalid!”
      .
5. **Form Submission (DemoQA)**
    - Go to https://demoqa.com/automation-practice-form.
    - Fill in fields → submit → verify confirmation modal.
6. **Responsive Navigation (W3Schools)**
    - Visit https://www.w3schools.com/.
    - Click the responsive menu → select “Learn JavaScript”.
    - Assert that heading *“JavaScript Examples”* is visible.
7. **File Upload Automation**

- Open https://the-internet.herokuapp.com/upload.
- Upload a file → assert text “File Uploaded!”

9. Exporting and Maintaining Scripts

- Go to File → Export → Java JUnit or Python pytest to reuse your script in WebDriver labs.
- Save exported files for integration in Lab 7 (Selenium WebDriver).
- Maintain clear naming conventions for elements and test files.

10. Deliverables

Each student must submit:

1. Selenium IDE project file (.side).
2. At least three completed test cases with assertions.
3. One exported script (Java/Python).
4. Screenshot of successful test execution.
5. Short report (objectives, key commands, results, issues).

11. Conclusion

Selenium IDE enables testers to quickly automate web-based tasks without writing code. It provides an excellent foundation for understanding web UI automation concepts such as element locators, assertions, waits, and reusable scripts. The skills gained in this lab will be extended in the next session on Selenium WebDriver, where students will script tests programmatically for robust automation.