****

**COS10026 WEB TECHNOLOGY PROJECT**

**Assignment 2 UI Testing (Report)**

|  |  |
| --- | --- |
| **Members** | **Members ID** |
| Kelvin Sze Hoon LAU | 104391559 |

Table of Contents

[Introduction 1](#_Toc200564053)

[UI Testing Tool 1: Selenium 2](#_Toc200564054)

[Tool 2: Cypress 3](#_Toc200564055)

[Comparison Table 4](#_Toc200564056)

[Conclusion 5](#_Toc200564057)

[References 6](#_Toc200564058)

# Introduction

The goal of this UI Testing Report is to explore automated testing tools that improve the quality and reliability of web interfaces. Rather than performing manual verification for each feature, automated UI testing allows us to validate interactive components, ensure visual consistency, and reduce the risk of undetected bugs. For the Brew & Go Coffee website—which includes dynamic features such as member top-ups, product searches, and form submissions—UI testing is essential to ensure a smooth user experience. This report evaluates two suitable UI testing tools: Selenium and Cypress, based on their compatibility with the features implemented.

UI Testing Tool 1: Selenium  
**Description:**Selenium is a widely used open-source framework for browser automation. It supports multiple programming languages, including PHP (via PHPUnit bindings), Java, and Python, and is compatible with major browsers like Chrome, Firefox, Edge, and Safari.

**Strengths:**

* Cross-browser testing: Ensures compatibility across different devices and browsers.
* Multi-language support: Allows test scripts in various languages, making integration with backend technologies seamless.
* Robust ecosystem: Selenium WebDriver integrates with frameworks like TestNG, JUnit, and CI/CD tools for enhanced automation workflows.

**Weaknesses:**

* Slower execution: Tests run via browser automation can be slower than native JavaScript-based tools.
* Complex setup: Requires initial configuration and ongoing maintenance.
* Limited built-in debugging: Troubleshooting errors may require manual investigation or third-party logging tools.

**Use for Brew & Go:**Selenium can be utilized to test critical site functionalities, such as member registration/login forms, the password reset process, and anti-spam measures through browser simulation.

Tool 2: Cypress  
**Overview:**Cypress is a modern JavaScript-based end-to-end testing framework designed specifically for web applications. It operates directly within the browser, providing real-time reloading and built-in debugging tools for efficient testing.

**Advantages:**

* Faster test execution: Runs in the same execution loop as the application, enabling quicker feedback.
* Built-in debugging tools: Supports screenshot and video capture for enhanced visibility.
* Automatic waiting: Manages DOM loading and element availability internally, reducing test flakiness.

**Disadvantages:**

* Limited browser compatibility: Primarily supports Chrome and Chromium-based browsers.
* JavaScript-only: Tests must be written exclusively in JavaScript.
* Less support for legacy systems: Not ideal for older applications or non-SPA architectures.

# Comparison Table

|  |  |  |
| --- | --- | --- |
| **Feature** | **Selenium** | **Cypress** |
| Language Support | Multiple (Java, Python, C#, etc.) | JavaScript only |
| Browser Support | Chrome, Firefox, Safari, Edge | Chrome, Edge (limited Firefox) |
| Execution Speed | Moderate to slow | Fast |
| Test Debugging | Manual, external tools | Built-in screenshots, video capture |
| Community and Support | |  | | --- | |  |  |  | | --- | | Large and mature | | Rapidly growing |
| Learning Curve | Steep | Beginner-friendly |

# Conclusion

Selenium and Cypress both provide powerful UI testing solutions, each suited to different scenarios. Selenium is well-suited for large-scale, complex applications that require support for multiple programming languages and browsers. In contrast, Cypress is optimized for modern JavaScript-based web applications, offering a streamlined setup and rapid test execution. The choice between the two depends on factors like the technology stack and team expertise, as either tool can be effectively integrated into UI testing workflows.

# References

Atlassian 2023, ‘What is Cypress?’, *Atlassian*, viewed 11 June 2025, <https://www.atlassian.com/continuous-delivery/testing/cypress>.

BrowserStack 2024, ‘Selenium vs Cypress: A Comparison of Test Automation Tools’, *BrowserStack*, viewed 11 June 2025, <<https://www.browserstack.com/guide/selenium-vs-cypress>>.

Cypress Docs 2024, ‘Introduction to Cypress’, *Cypress.io*, viewed 11 June 2025, <https://docs.cypress.io/guides/overview/why-cypress>.

Guru99 2024, ‘What is Selenium’*,* *Guru99*, viewed 11 June 2025, <https://www.guru99.com/selenium-tutorial.html>.

MDN Web Docs 2024, ‘Automated testing with Selenium’, *Mozilla Developer Network*, viewed 11 June 2025, <<https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Cross_browser_testing/Automated_testing>>.

Testim 2023, ‘Cypress vs Selenium: Key Differences’, *Testim.io*, viewed 11 June 2025, <https://www.testim.io/blog/cypress-vs-selenium/>.