

Complete QA & Selenium Automation Study Report

Coverage: From First Topic to Last Topic Discussed

Focus: Manual + Automation Testing (Mobile & Web)

1. Mobile Application Testing Process

Mobile application testing ensures that an app works correctly across devices, OS versions, screen sizes, and network conditions. Testing starts from requirement analysis and continues until production release.

Types of Mobile Testing:

- Functional Testing – verifies features like login, forms, notifications
- UI/UX Testing – layout, alignment, responsiveness
- Compatibility Testing – devices, OS versions
- Performance Testing – speed, load, stress
- Security Testing – data protection, session handling

2. Manual vs Automation Testing

Manual testing is best for exploratory, UI, and usability testing, while automation is best for regression, repetitive tasks, and critical flows like login. Both approaches are used together in real QA projects.

3. Selenium Automation Introduction

Selenium is an open-source automation tool used for automating web browsers. It supports multiple languages including Python. Selenium only performs actions that are explicitly coded by the tester.

4. Environment Setup & Issues Faced

During setup, issues such as missing Maven, Java version mismatch, and command not found errors were encountered. The approach was changed to Python-based Selenium setup using virtual environments for simplicity.

5. Login Automation Using Selenium (Python)

Login automation involved opening the browser, navigating to the login URL, locating username and password fields, entering credentials, and submitting the form. Explicit waits were used to handle dynamic loading.

6. Common Selenium Errors & Fixes

- NoSuchElementException – locator incorrect or element not loaded
- TimeoutException – wait condition not met
- ElementNotInteractable – element hidden or disabled

7. Dashboard and Sidebar Automation

After login, the dashboard loads with a sidebar. Sidebar items like Notice are dynamic and open panels instead of navigating to new pages. Proper waits and JavaScript clicks are required.

8. Notice / Send Message Feature Automation

The Notice feature opens a message panel where messages are sent to users. Automation includes clicking Notice, opening Send Message, filling subject and message, and clicking Send.

9. Using Chrome Inspect Tool

Chrome DevTools Inspect was used to identify correct locators. Steps included highlighting elements, checking attributes like id, name, aria-label, detecting iframes, and validating XPath using `$x()` in console.

10. Key QA Learnings & Best Practices

- Selenium does not perform actions automatically
- Inspecting DOM is critical before automation
- Dynamic apps require explicit waits
- Debugging is part of automation, not failure

Conclusion

This document covers the complete learning journey from mobile testing fundamentals to real-world Selenium automation challenges. It establishes a strong QA foundation for advanced automation frameworks and professional testing practices.