

GROUP 4

PROJECT

OUR TEAM

SHIRLLIE PRAISE APIYO - 667438

ABEDINEGO ISHIMWE - 666221

EDDY WAWERU - 665939

PAUL MUEMA - 665506

INTRODUCTION

Our project is about developing an automated testing framework for a web application, covering unit, integration, and end-to-end (E2E).

The framework will ensure efficient and effective testing, reduce manual testing efforts, and improve overall testing quality.

PROJECT OVERVIEW

1. Project Goal: Develop an automated testing framework to ensure efficient, effective, and high-quality testing.
2. Scope: Covers unit, integration, and end-to-end (E2E).
3. Tools: Python, Pytest, Allure and Selenium.
4. Objective: Reduce manual testing efforts, increase coverage, and detect issues early.

TOOLS & TECHNOLOGIES

Python: A programming language for scripting tests.

Pytest: Framework for writing simple, scalable test cases.

Trello: documenting our processes

Selenium: Tool for browser automation to simulate user actions and testing.

Git: For version control and collaboration.

Continuous Integration (CI): Using allure ci-cd.

FRAMEWORK OVERVIEW

Testing Types:

1. Unit Testing: Verifies individual components in isolation.
2. Integration Testing: Ensures components work together correctly.
3. End-to-End Testing: Simulates real user scenarios.
4. Report generation: Giving out a report using allure.
5. Automation Strategy: Explanation of how Pytest and Selenium are used together

REQUIREMENTS

FUNCTIONAL REQUIREMENTS

1. The framework should support automated execution of unit, integration and E2E testing.
2. It should be able to generate detailed test reports.
3. Tests should run on multiple browsers (e.g., Chrome, Firefox).
4. The framework should identify and log failed test cases for debugging.
5. The framework should handle test data creation and cleanup.

REQUIREMENTS

NON-FUNCTIONAL REQUIREMENTS

1. Performance: Tests should execute within a reasonable time frame.
2. Usability: Easy to understand and extend for new test cases.
3. Scalability: Framework should handle a growing number of test cases.
4. Reliability: Must be robust and consistently produce accurate results.
5. Maintainability: Code should be well-documented and structured for easy maintenance.

CHALLENGES

1. Tool Familiarity: Learning Pytest and Selenium from scratch.
2. Synchronization: Managing dynamic elements and waiting times in Selenium.
3. Test Flakiness: Handling inconsistent test results.
4. Browser Compatibility: Ensuring tests run on different browsers smoothly.
5. CI Integration: Setting up automated testing in a CI pipeline.

ACHIEVEMENTS

1. Coverage: Improved test coverage compared to manual testing.
2. Efficiency: Significant reduction in time spent on repetitive testing tasks.
3. Reliability: Fewer bugs reaching production due to automated regression testing.
4. Learning: Gained hands-on experience with industry-standard tools.

NEXT STEPS /CONCLUSION

1. Summary: Automation improved testing efficiency and reliability.
2. Next Steps: Further optimize and scale the framework.
3. Integrate performance testing.
4. Continue refining skills with Pytest, Selenium and GitHub Workflow.

THANK YOU
SO MUCH