# DEMOBLAZE

# Project Title: DEMOBLAZE

# Introduction:

The Demoblaze project aims to automate the testing of the Demoblaze web application, a commercial website for buying and selling products. This project encompasses UI automation to ensure the application functions correctly and meets all user requirements.

# Problem Statement :

Manual testing of the Demoblaze web application is time-consuming, error-prone, and inefficient. The increasing complexity of the application necessitates a robust automated testing solution to maintain high quality and reliability.

Objective:

The primary objective of this project is to develop an automated testing framework for UI components of the Demoblaze web application. This framework should:

* Reduce the time required for testing.
* Increase test coverage and accuracy.
* Provide comprehensive test reports.
* Facilitate continuous integration and delivery.

# Project Components:

# 1. UI Automation Framework Development:

Tools: Selenium, TestNG , JUnit, cucumber.

Description: Automate the testing of the web application UI components.

# 2. Test Case Design:

Identify and document test scenarios.

Develop detailed test cases for UI testing.

# 3. Test Case Requirements:

Functional requirements.

Non-functional requirements.

User stories and acceptance criteria.

# 4. Test Execution:

Running automated tests.

Monitoring and reporting test results.

# 5. Reporting and Documentation:

Generating test reports.

Documenting test results and any issues found.

# Test Case Design:

# UI Test Cases:

# **1. Login Functionality**

Verify that users can log in with valid credentials.

Verify error messages for invalid credentials.

# 2. Product Search:

Verify that users can search for products.

Verify search results accuracy.

Test Case Requirements:

**1. Functional Requirements**

Accurate product search.

Secure and functional user authentication.

**2. Non-Functional Requirements**

Performance under load.

Security of user data.

Test Execution:

Environment: Setup of test environments (development, staging).

Execution: Running automated test scripts.

# Reporting and Documentation:

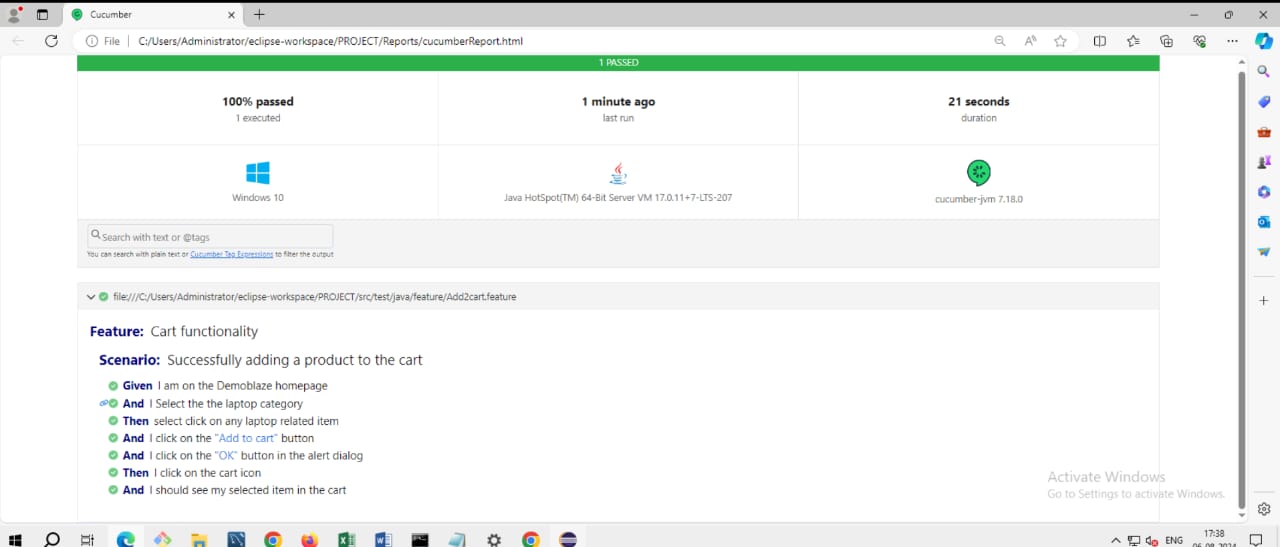
Test Reports: Generating detailed reports using tools like Allure.

Documentation: Maintaining documentation of test cases, execution, and results.

# Screenshots:

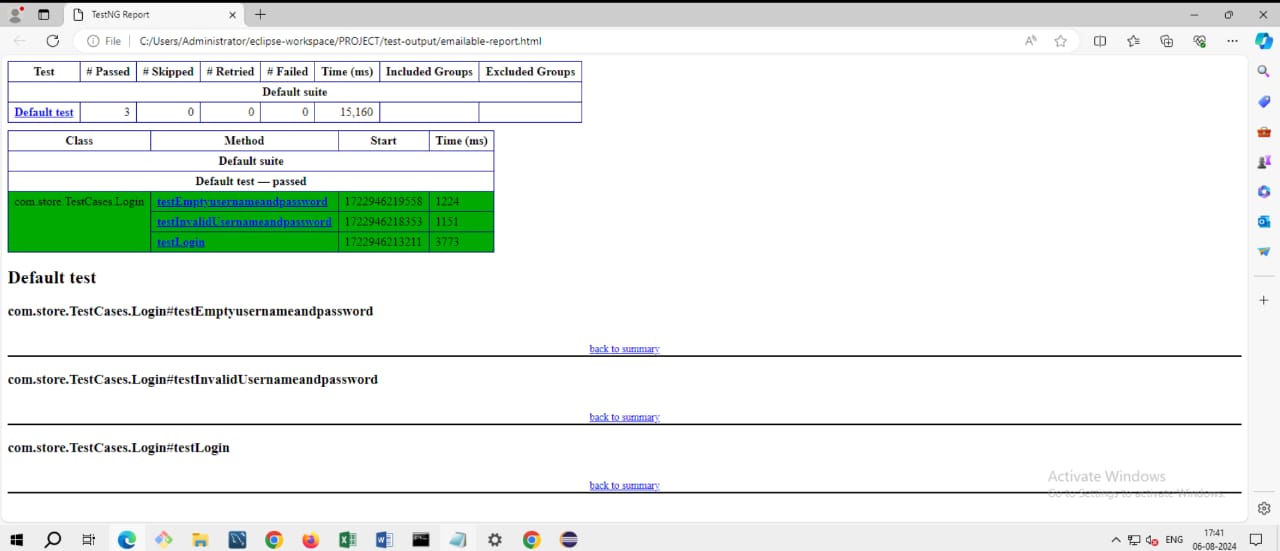
# CUCUMBER:

This report provides an overview of test execution and highlights any issues encountered during the testing process of cucumber.

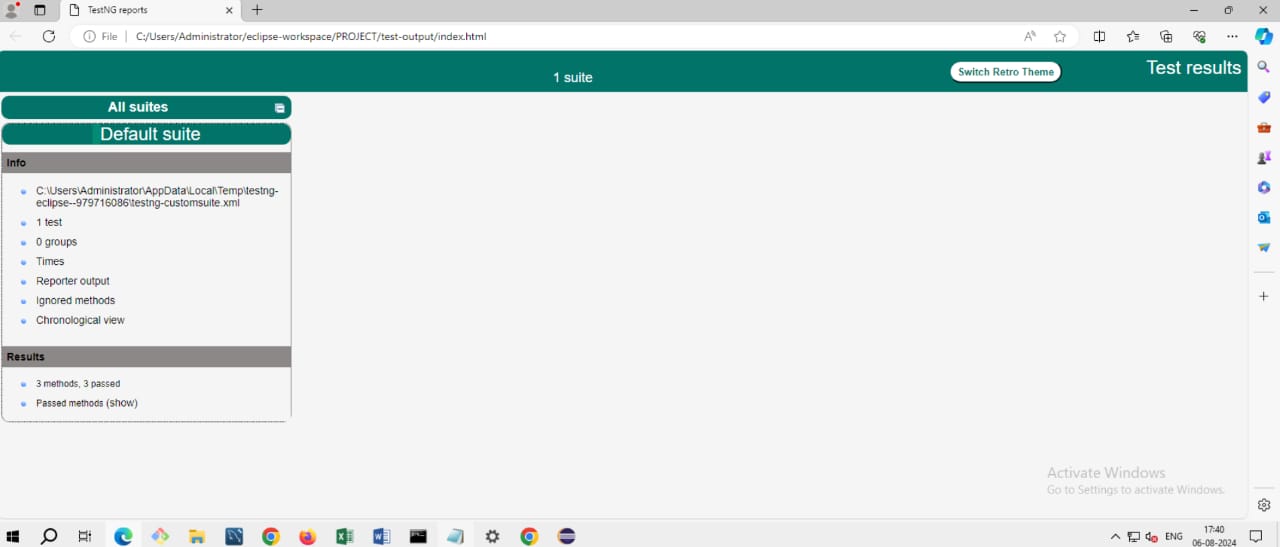


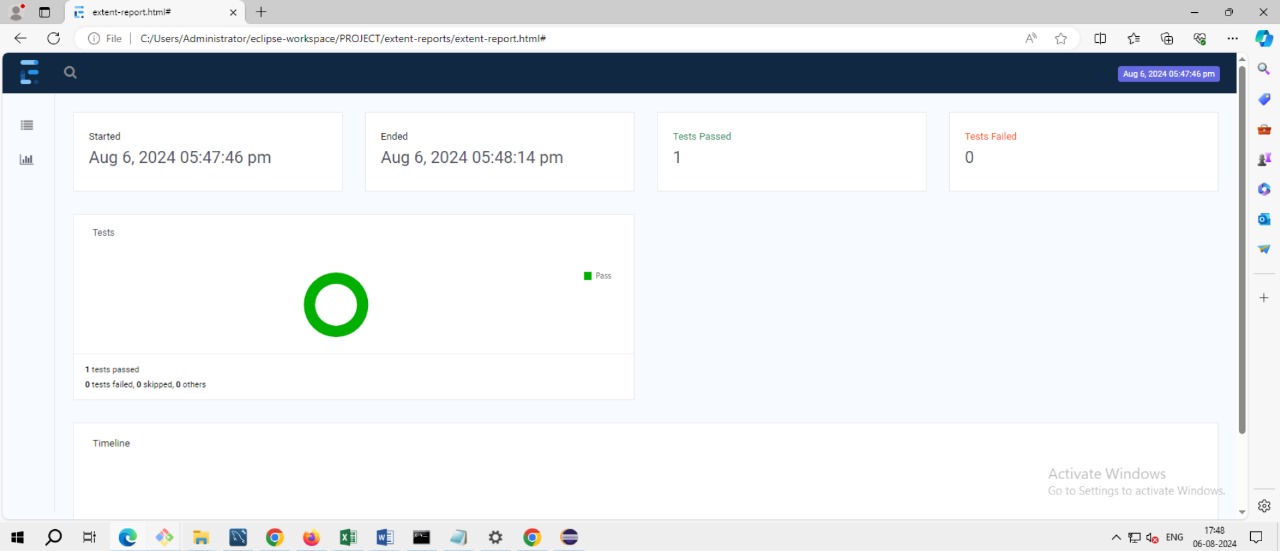
1. TESTNG:

comprehensive summary of the test execution, including details on which tests passed, failed, or were skipped.It often includes test metrics, such as execution time, number of assertions, and error messages, which help in assessing the overall quality of the application.



1. The index.html file is crucial in JUnit test reports as it provides a user-friendly, centralized view of the test results.



1. This extent-report.html summary helps in effectively communicating the results of the test execution, providing actionable insights, and facilitating informed decision-making to enhance the quality of the application.

# Project Deliverables:

1. Automated test scripts for UI

2. Test execution reports.

3. Comprehensive documentation of test cases and results.

4. Final project report with analysis and recommendations.

Evaluation Criteria:

1. Accuracy: Test cases should correctly verify functionality.

2. Coverage: Comprehensive coverage of all critical functionalities.

3. Efficiency: Reduced time and effort for test execution.

4. Maintainability: Ease of updating and maintaining test scripts.

# Conclusion:

The successful completion of the Demoblaze project will result in a robust automated testing framework that ensures high quality and reliability of the application. This will lead to faster release cycles, reduced manual testing effort, and increased confidence in the application’s performance and functionality.