Automation Testing of an

E-commerce website

# Table of Contents

PROJECT CHARTER 4

PROJECT PARTICIPANTS 5

PROJECT DESCRIPTION 6

MILESTONE 7

ASSUMPTIONS, CONSTRAINTS, DEPENDENCIES, IMPACTS AND RISKS 8

TEST STRATEGY 9

INTRODUCTION 10

SCOPE  10

FUNCTIONAL MODULES TESTED  11

NON-FUNCTIONAL ASPECTS COVERED  11

TESTING METHODOLOGIES 12

MANUAL TESTING APPROACH  13

AUTOMATION TESTING STRATEGY  13

INTEGRATION OF MANUAL AND AUTOMATED TESTING  14

TOOLS AND TECHNOLOGIES USED  16

SELENIUM WEBDRIVER FOR TESTING 17

CUCUMBER FOR BEHAVIOUR -DRIVEN DEVELOPMENT (BDD)  17

PLAYWRIGHT FOR CROSS-BROWSER END-TO-END TESTING  18

JMETER FOR LOAD AND PERFORMANCE TESTING  19

EXCEL FOR MANUAL TEST CASE DOCUMENTATION  20

TEST CASE DESIGN AND EXECUTION  21

TEST SCENARIOS AND TEST CASE PREPARATION  22

TEST DATA MANAGEMENT  22

TEST EXECUTION PROCESS  23

TEST RESULTS AND REPORTING  24

AUTOMATION TEST REPORTS (SELENIUM, PLAYWRIGHT)  25

MANUAL TESTING OUTCOME SUMMARY  28

LOAD TESTING RESULTS USING JMETER 29

REPORTING AND DOCUMENTATION  31

**PROJECT CHARTER**

|  |  |  |
| --- | --- | --- |
| PROJECT PARTICIPANTS | | |
| **PROJECT TITLE** | Automation Testing Documentation  (https://demowebshop.tricentis.com/) | |
| **STAKEHOLDER** | Project Sponsor | Mphasis Limited |
| **PROJECT TEAM** | Project Manager | DINESH KANNAN M |
| Team Members | GOKULKANNAN MUTHUVEL  SOWMIYA P |
| Email Address | <gokulkannan.muthuvel@mphasis.com>**,**  <sowmiya.p@mphasis.com> |

**PROJECT PARTICIPANTS**

**PROJECT DESCRIPTION**

**Objective**

The goal of the testing phase is to ensure the website (<https://demowebshop.tricentis.com>) functions correctly, reliably, and securely. This involves validating the accuracy of various functionalities, including user interactions, shopping cart operations, and the checkout process, as well as ensuring seamless API integration.

**Description and background**

Testing is crucial to identify and rectify issues before deployment. Given the complexity of e-commerce websites, thorough testing will minimize errors and enhance the system's efficiency. The testing phase will leverage automated tools and methodologies to validate the system's performance and reliability.

**Scope**

* **System Testing**

Validate the complete system's functionality, including the checkout process and overall user experience

* **User Acceptance Testing (UAT)**

Confirm that the system meets user requirements and expectations. API Development: Developing secure, scalable endpoints for model access and integration.

**Deliverables**

* **Test Plan:** Detailed plan outlining the testing strategy, objectives, scope, and schedule.
* **Test Cases:** Comprehensive test cases covering unit, integration, system, and UAT.
* **Test Reports:** Summary of test execution and results, including defect reports.
* **Automated Test Scripts:** Scripts for automated testing using tools like Selenium, Cucumber, J Meter and Playwright.

**MILESTONE**

**Schedule**

|  |  |
| --- | --- |
| Start Date | 15-04-2025 |
| End Date | 20-06-2025 |

**Progress milestones**

The progress milestones associated with this project are as follows:

|  |  |
| --- | --- |
| **Week 1** | Define testing goals, create test plan, set up test environment, develop test cases. |
| **Week 2** | Execute unit testing, log defects, review and refine test cases. |
| **Week 3** | Execute integration testing, log defects, prepare test data for system testing. |
| **Week 4** | Execute system testing, log defects, conduct performance testing |
| **Week 5** | Generate test reports, review and finalize documentation, prepare for deployment. |

# **ASSUMPTIONS, CONSTRAINTS, DEPENDENCIES,** **IMPACTS AND RISK**

|  |  |
| --- | --- |
| **ASSUMPTIONS** | * The website can handle various user interactions seamlessly, ensuring a smooth shopping experience. * Users can access the website securely and perform transactions without any issues. * The testing environment accurately simulates real-world conditions to validate the website's performance. * The system adheres to regulatory compliance frameworks to ensure the security and privacy of user data. |
| **CONSTRAINTS** | * The website must accurately process user inputs and transactions without compromising data integrity and security. * Automated testing tools require adequate test data to efficiently validate functionalities. * The API and website should provide scalability and real-time responsiveness for a smooth user experience. * Compliance with financial and data protection regulations must be ensured during testing and deployment. |
| **RISKS** | * Potential security breaches and compliance issues with regulations. * Automated testing tools may fail to identify certain edge cases or bugs. * Errors in transaction processing and user interactions could impact the user experience. |

**TEST STRATEGY**

**INTRODUCTION**

The project provides a comprehensive overview of the testing activities conducted on the demowebshop.tricentis.com website. It includes details on the scope of the project, the methodologies employed, the tools and technologies used, and the results obtained from various testing phases. The primary goal of the project is to ensure the website's functionality, performance, and reliability across different environments and platforms. The testing activities include both functional and non-functional aspects, covering user interactions, performance, security, usability, and compatibility.

The tools used in the project include Selenium WebDriver, Cucumber, Playwright, JMeter, and Excel. The results provide insights into the website's performance and reliability, with reports highlighting test execution status, defects identified, and performance metrics. Overall, the project aims to provide a clear account of the testing activities, ensuring thorough validation of the website.

**SCOPE OF THE PROJECT**

The scope of this project encompasses both functional and non-functional testing of the demowebshop.tricentis.com website. The primary objective is to ensure the website's functionality, performance, and reliability across different environments and platforms. This involves validating the core features of the website, assessing its performance under various conditions, ensuring security measures are in place, and verifying compatibility across multiple devices and browsers.

Functional testing aims to verify that all user interactions and business processes on the website work as intended. Non-functional testing focuses on aspects such as performance, load handling, security, usability, and compatibility, which are crucial for providing a seamless and secure user experience.

**FUNCTIONAL MODULES TESTED**

* User Registration and Login: Validating account creation, login, and profile management.
* Product Search and Filtering: Ensuring accurate and responsive search functionality.
* Shopping Cart and Checkout Process: Testing cart management and purchase completion.
* Payment Gateway Integration: Verifying secure payment processing.
* Order History and Tracking: Checking order details and tracking updates.
* User Profile Management: Ensuring users can update personal information securely.

**NON-FUNCTIONAL ASPECTS COVERED**

* Performance Testing: Assessing responsiveness and stability.
* Load Testing: Simulating high traffic conditions.
* Security Testing: Identifying vulnerabilities and ensuring data protection.
* Usability Testing: Evaluating user experience and interface design.
* Compatibility Testing: Ensuring functionality across different browsers and devices.

**TESTING**

**METHODOLOGIES**

**MANUAL TESTING APPROACH**

Objective: Validate the functionality of the demowebshop.tricentis.com website through hands-on testing.

Process:

* Test Case Design: Test cases were meticulously designed based on user stories and requirements, covering various functionalities such as user registration, product search, shopping cart management, and payment processing.
* Execution: Test cases were executed manually to simulate real user interactions and identify defects.
* Defect Logging: Screenshots were taken during manual testing to document defects and unexpected behaviors. These screenshots should be included in the defect logs and test reports to provide visual evidence of issues encountered, such as error messages, broken links, or incorrect data displays.
* Advantages: Manual testing allowed for detailed observation of user interactions and provided flexibility to adapt to new scenarios, ensuring the application behaves as expected.

**AUTOMATION TESTING STRATEGY**

Objective: Increase efficiency, coverage, and repeatability of tests through automation. Tools Used: Selenium WebDriver, Playwright, and Cucumber for Behavior-Driven Development (BDD).

Process:

* Script Development: Automated test scripts were developed to cover various user scenarios.
* Execution: Selenium WebDriver facilitated cross-browser testing, ensuring the website functions correctly on different browsers such as Chrome, Firefox, Safari, and Edge. Playwright was used for end-to-end testing across multiple browsers, providing a comprehensive assessment of the website's functionality. Cucumber enabled the creation of test scenarios in plain language, enhancing collaboration between technical and non-technical stakeholders.
* Reporting: Screenshots were captured during automated test execution to provide visual confirmation of test results and defects. These screenshots should be included in the automation test reports to illustrate the outcomes of the tests, such as successful form submissions, navigation flows, and error messages.
* Advantages: Automation testing helped in quickly identifying regressions and ensuring consistent test execution.

**INTEGRATION OF MANUAL AND AUTOMATED TESTING**

Objective: Leverage the strengths of both manual and automated testing methodologies for comprehensive coverage. Approach: Hybrid testing strategy.

Process:

* Initial Validation: Manual testing was used for initial validation and exploratory testing, allowing testers to uncover complex issues and validate new features.
* Automated Regression: Automated tests were employed for regression testing and repetitive tasks, providing quick and reliable results.
* Reporting: Screenshots from both manual and automated testing should be compiled into comprehensive reports to provide a clear and detailed account of the testing activities. These reports should include visual evidence of the test scenarios executed, defects identified, and the state of the application at various points.
* Advantages: By combining both approaches, the team could achieve a balance between thoroughness and efficiency in their testing efforts, ensuring the overall quality of the website.

**TOOLS AND**

**TECHNOLOGIES USED**

**SELENIUM WEBDRIVER FOR TESTING**

Selenium WebDriver was used for automating web application testing, facilitating cross-browser testing and regression testing. It ensured the website functions correctly on different browsers such as Chrome, Firefox, Safari, and Edge. Automated test scripts covered various user scenarios, including form submissions, navigation, and data validation.

**CUCUMBER FOR BEHAVIOUR-DRIVEN DEVELOPMENT (BDD)**

Cucumber was employed for Behavior-Driven Development (BDD), allowing the creation of test scenarios in plain language. This approach enhanced collaboration between technical and non-technical stakeholders by making test cases more understandable and accessible. Test scenarios were written in Gherkin syntax, describing the expected behavior of the application in a given context.

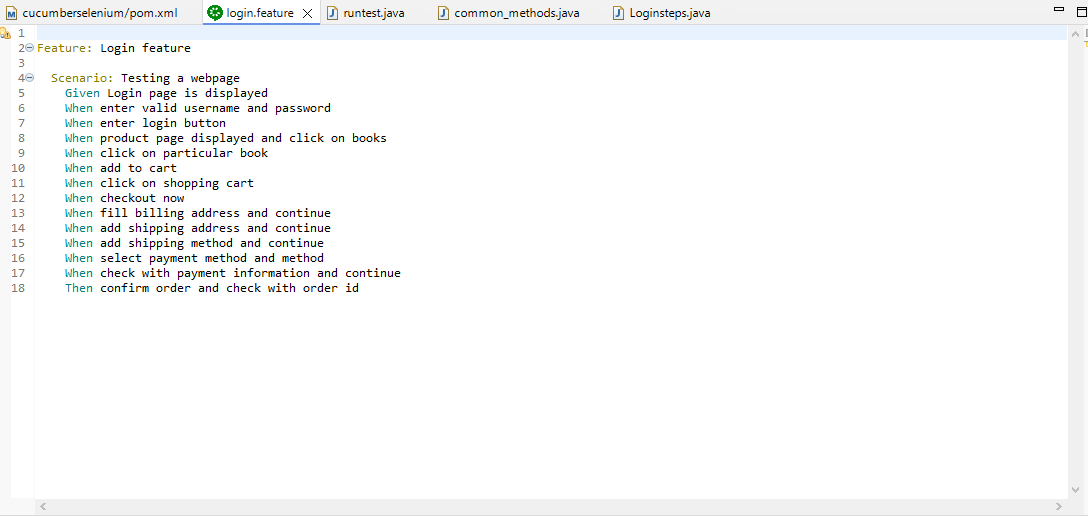


Fig. Feature file of Selenium and Cucumber

**PLAYWRIGHT FOR CROSS-BROWSER END-TO-END TESTING**

Playwright was utilized for end-to-end testing across multiple browsers, ensuring the website's functionality and performance were consistent across different browser environments. Playwright's capabilities allowed for testing complex user interactions, such as drag-and-drop, file uploads, and dynamic content loading.

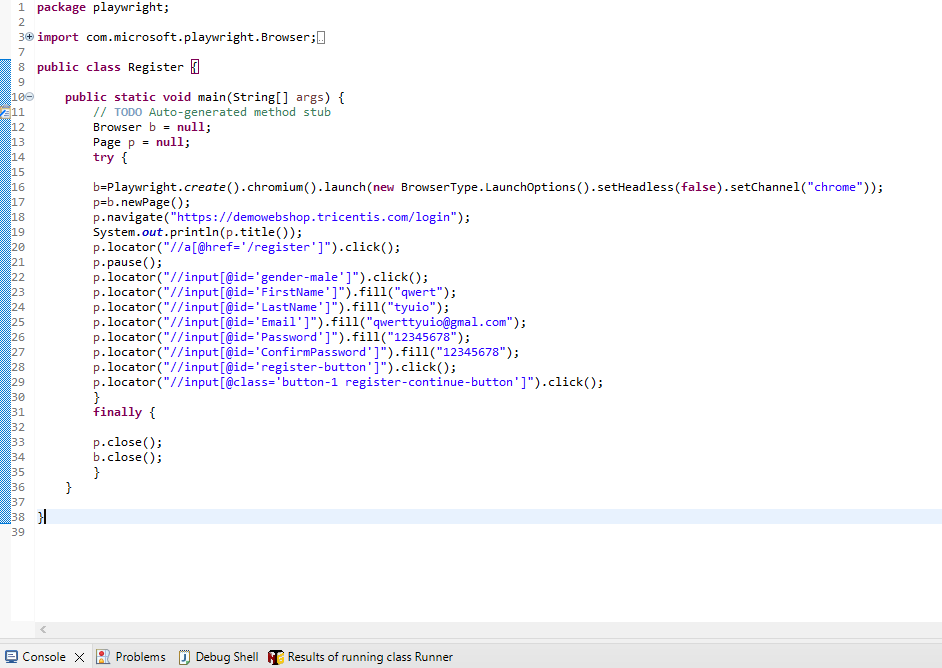


Fig. Register page of Playwright

**JMETER FOR LOAD AND PERFORMANCE TESTING**

JMeter was used to simulate load and measure the performance of the website under various conditions. It helped identify bottlenecks and optimize the application's performance by simulating multiple users accessing the website simultaneously. Load testing scenarios included stress testing, spike testing, and endurance testing to evaluate the website's stability and responsiveness.

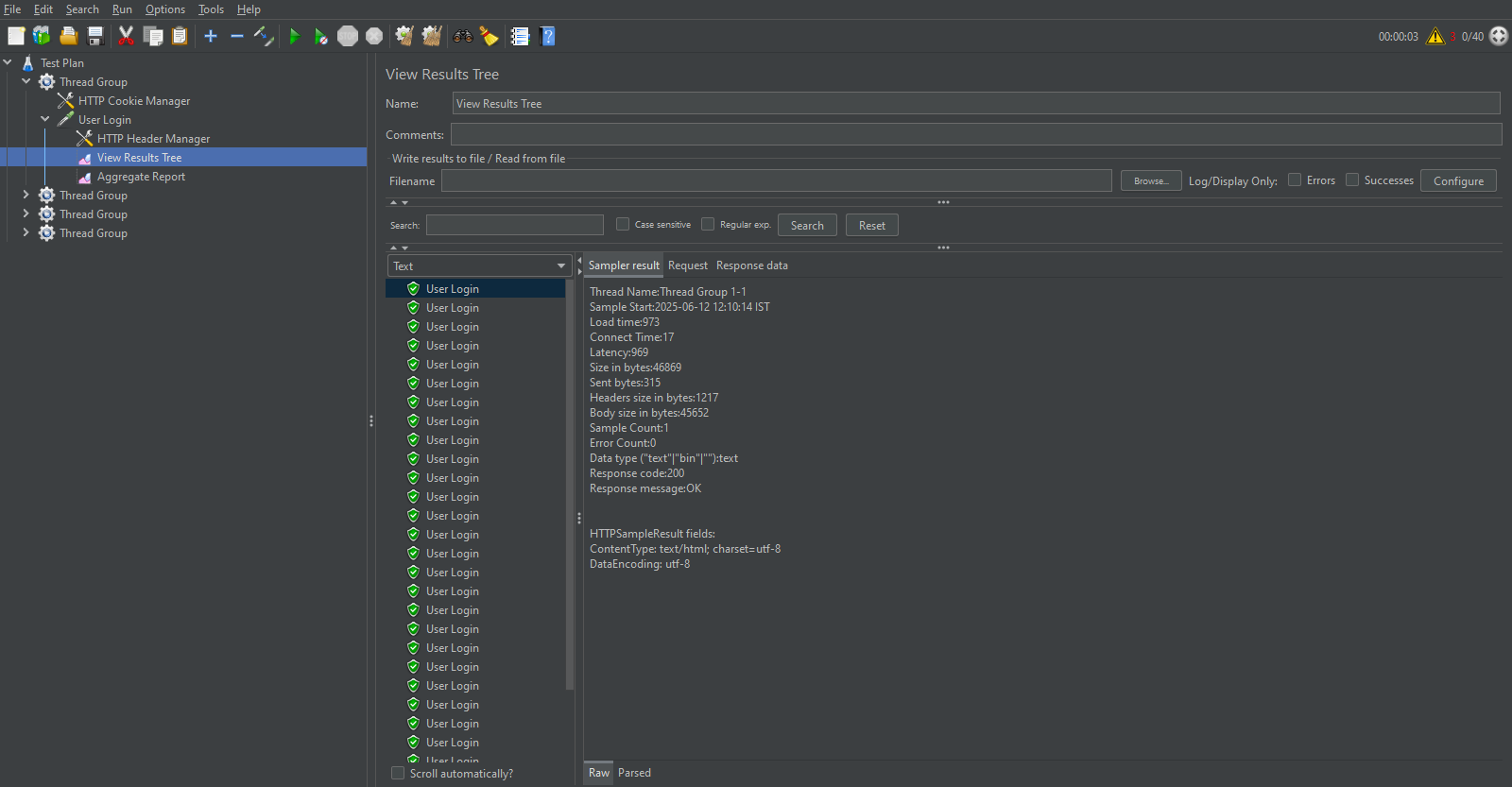


Fig. JMeter

**EXCEL FOR MANUAL TEST CASE DOCUMENTATION**

Excel was used to document manual test cases, track execution status, and record defects. It provided a structured approach to manual testing, allowing testers to organize test cases, capture test results, and log defects systematically.

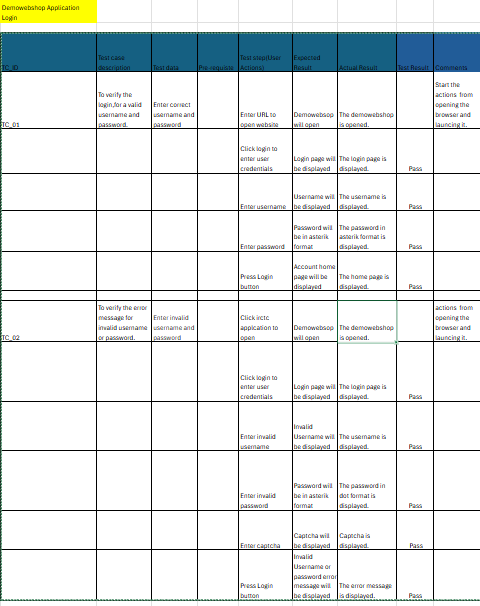


Fig. Test Case Description

**TEST CASE DESIGN**

**AND EXECUTION**

**TEST SCENARIOS AND TEST CASE PREPARATION**

Test scenarios were derived from the requirements and user stories. Detailed test cases were prepared to cover all functional and non-functional aspects of the website. Screenshots of the test case documents should be included to provide a visual reference for the test scenarios.

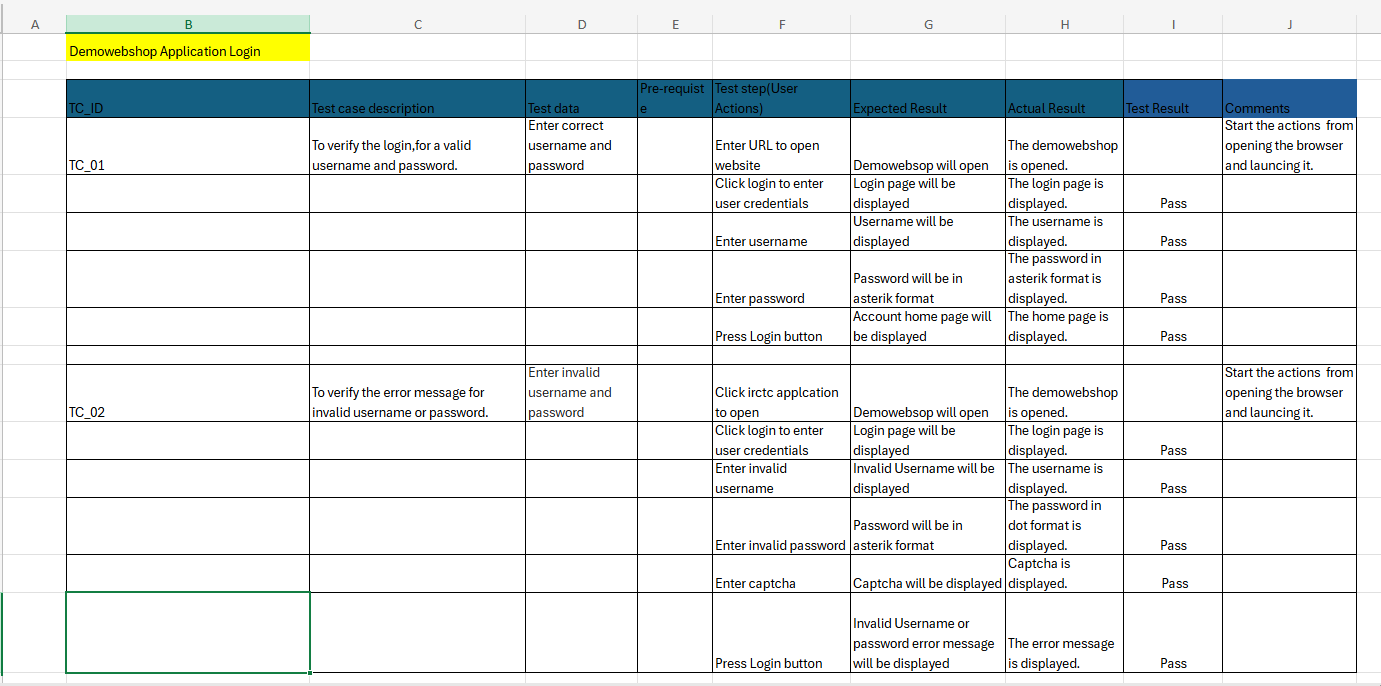


Fig. Test Case Description

**TEST DATA MANAGEMENT**

Test data was created and managed to ensure comprehensive testing. Data sets included valid, invalid, and edge cases to validate the application's behavior. Screenshots of the test data management process should be included to illustrate how data was handled.

**TEST EXECUTION PROCESS**

Test cases were executed manually and automatically. The execution process included logging defects, retesting fixed issues, and verifying the overall functionality. Screenshots of the test execution process should be included to show the steps taken and the results obtained.

**TEST RESULTS**

**AND REPORTING**

**AUTOMATION TEST REPORTS (SELENIUM, CUCUMBER AND PLAYWRIGHT)**

Automation test reports were generated using Selenium and Playwright. These reports provided insights into test execution status, pass/fail rates, and identified defects. Screenshots from the automation tests should be included in the reports to provide visual confirmation of the results.

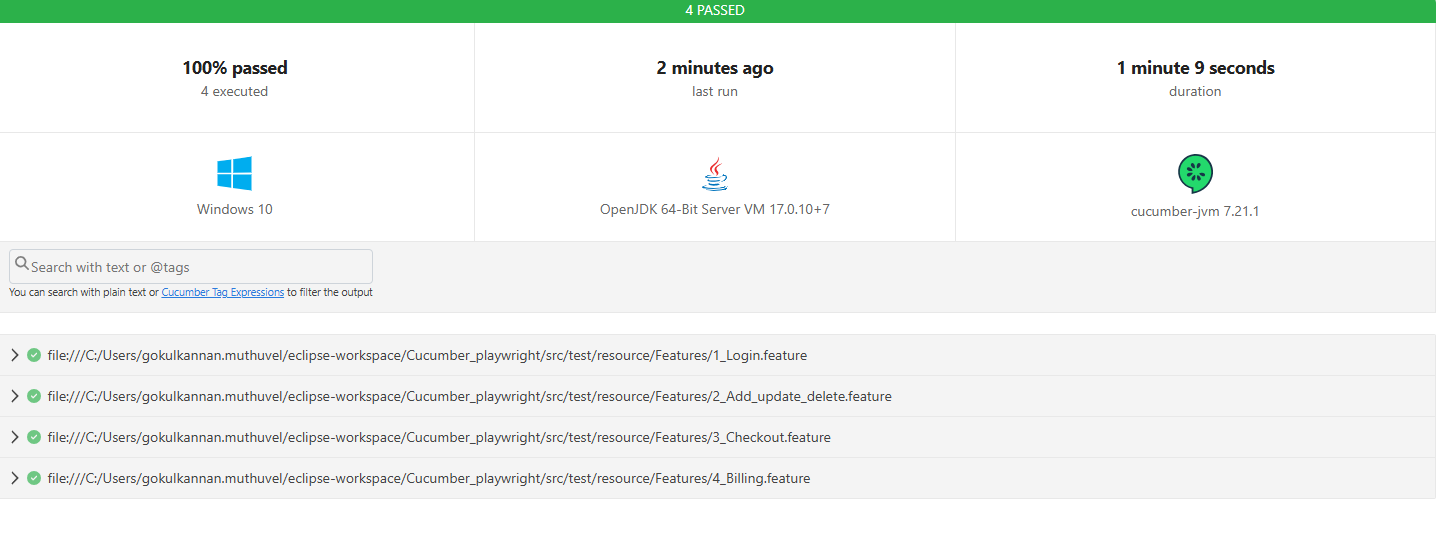


Fig. Output Screenshot of Cucumber

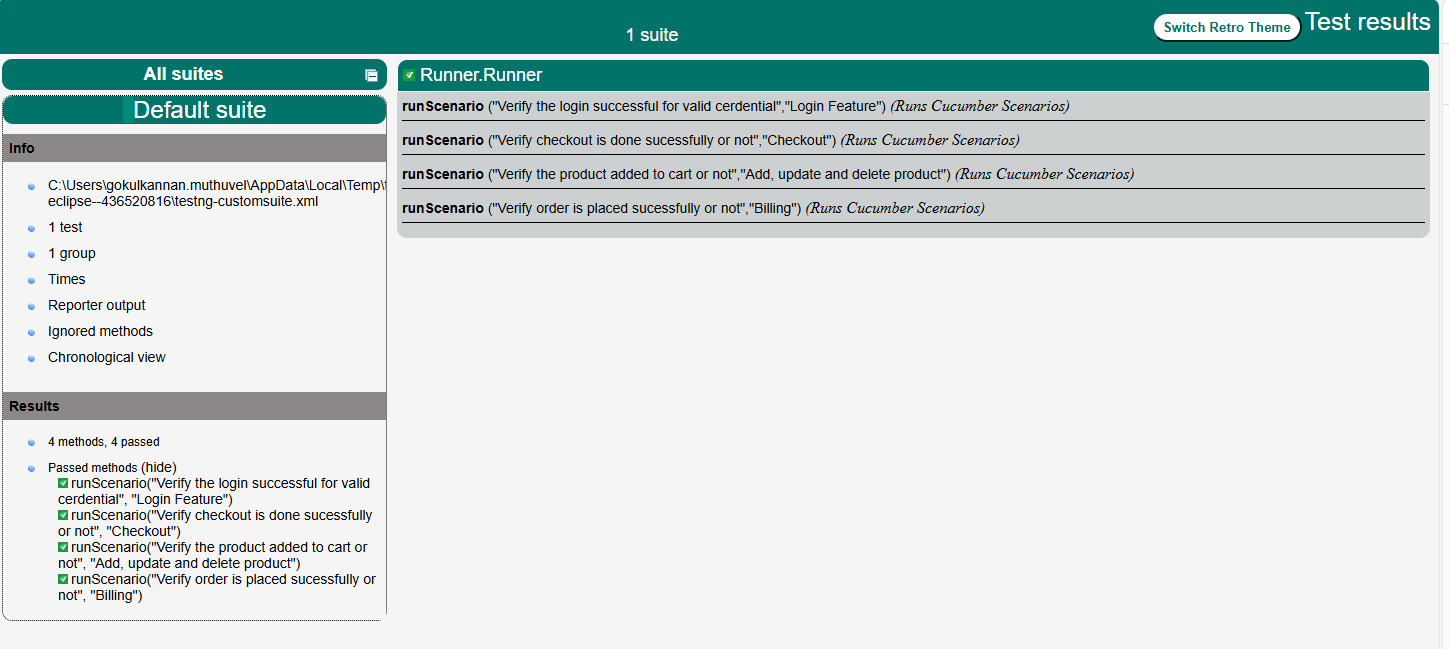


Fig. TestNG reports (XML report)

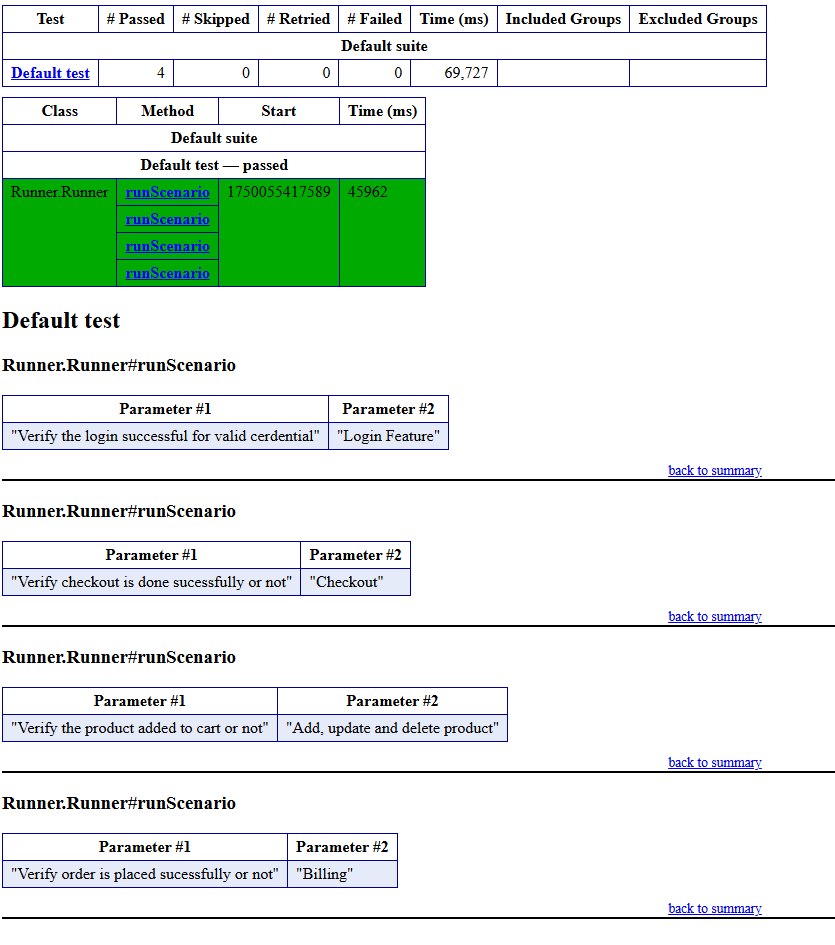


Fig. TestNG reports (HTML report)

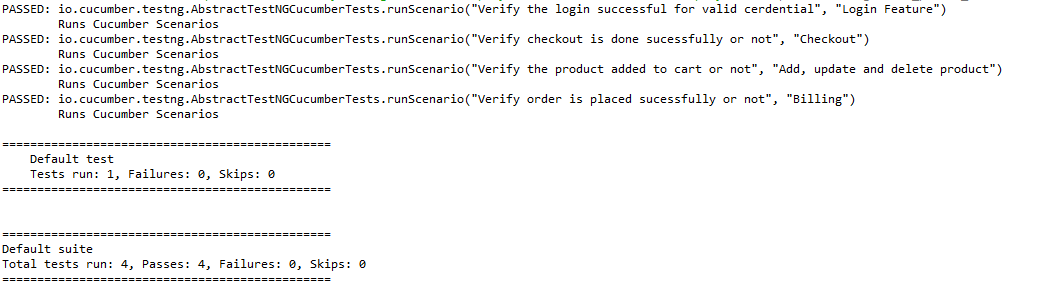


Fig. Console Output1

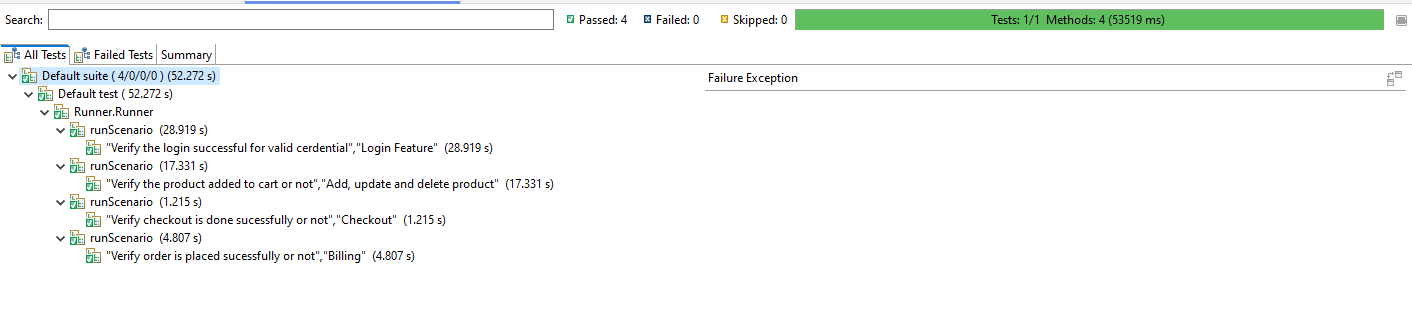


Fig. Console Output2

**MANUAL TESTING OUTCOME SUMMARY**

A summary of manual testing outcomes was documented, highlighting the test cases executed, defects found, and their resolution status. The link for the complete Test Case Description is……

([TCD.xlsx](https://mphasis-my.sharepoint.com/:x:/p/sowmiya_p/ETuJCE-GTONHlF25ojXuagABQQGg_pRhiItZ5jI78QAPXA?e=IrLNnb))

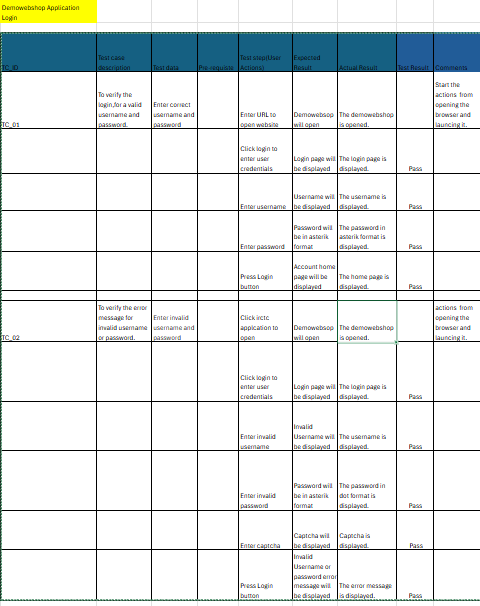


Fig. Test Case Description for Login



Fig. Test Case Description for Checkout

**LOAD TESTING RESULTS USING JMETER**

Load testing results from JMeter were analyzed to assess the website's performance under different load conditions. Key metrics included response times, throughput, and error rates. Screenshots of the JMeter results and graphs should be included in the load testing reports to visualize the performance metrics.

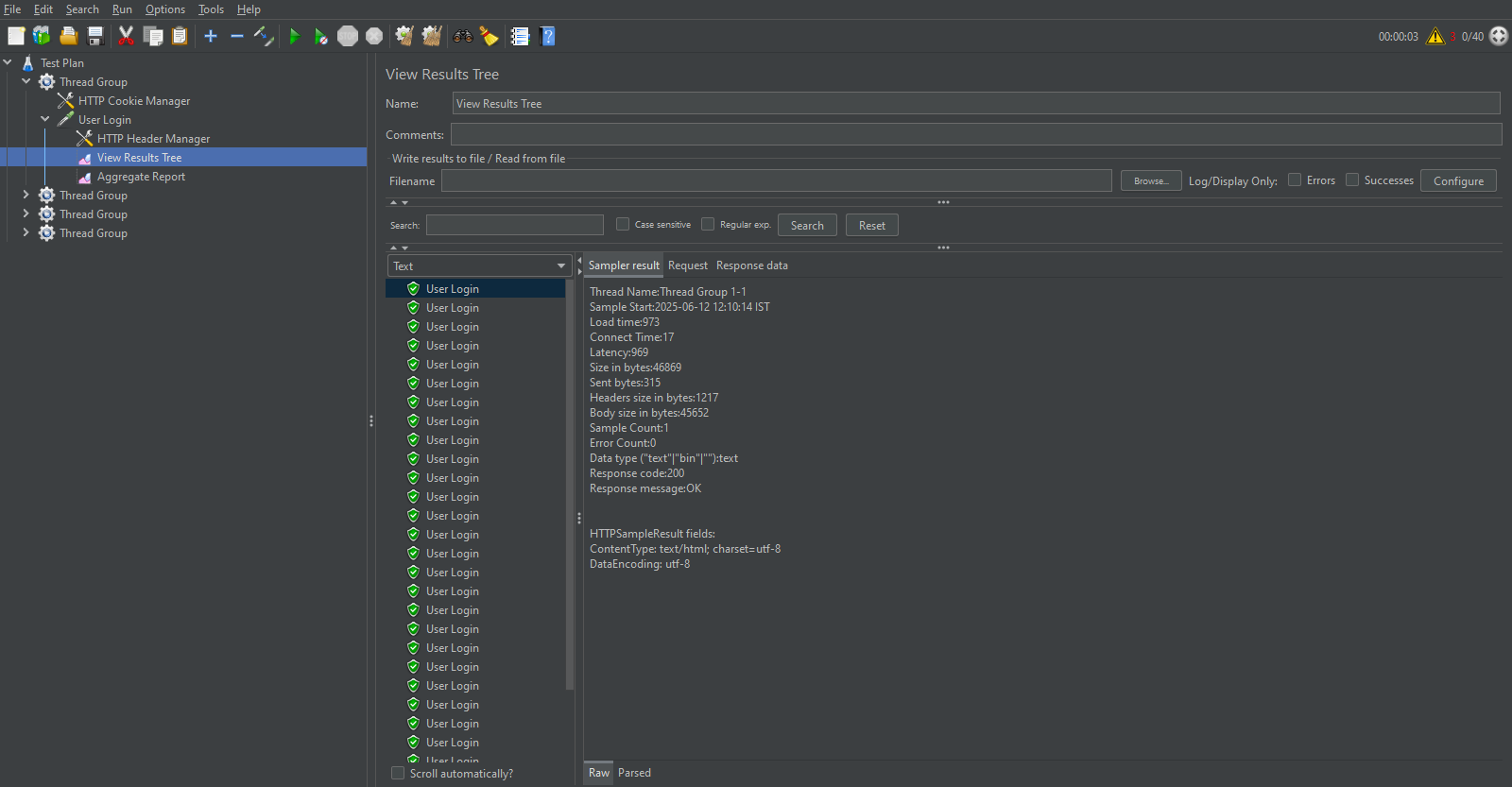


Fig. Login image of JMeter

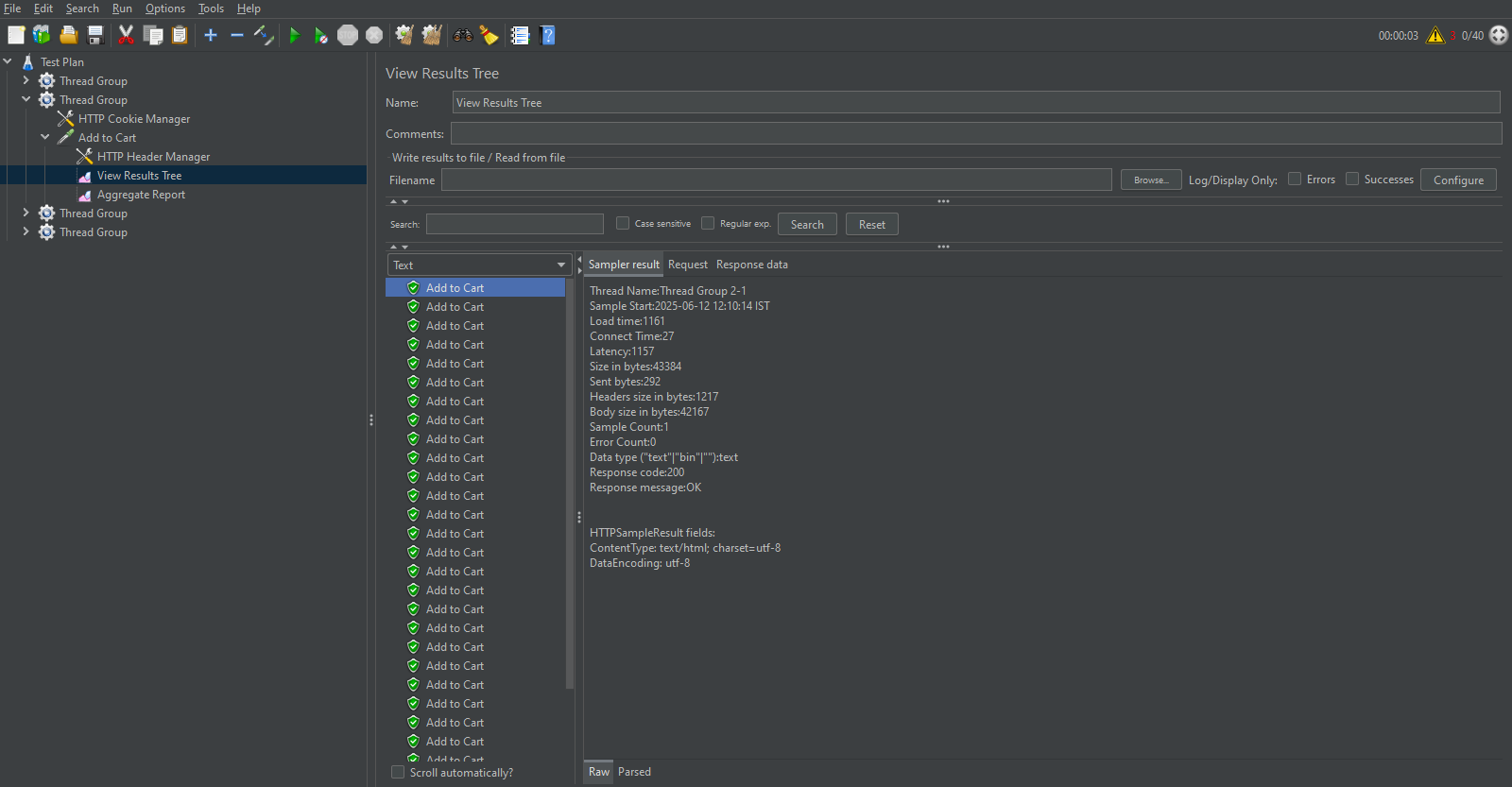


Fig. Add to cart of JMeter

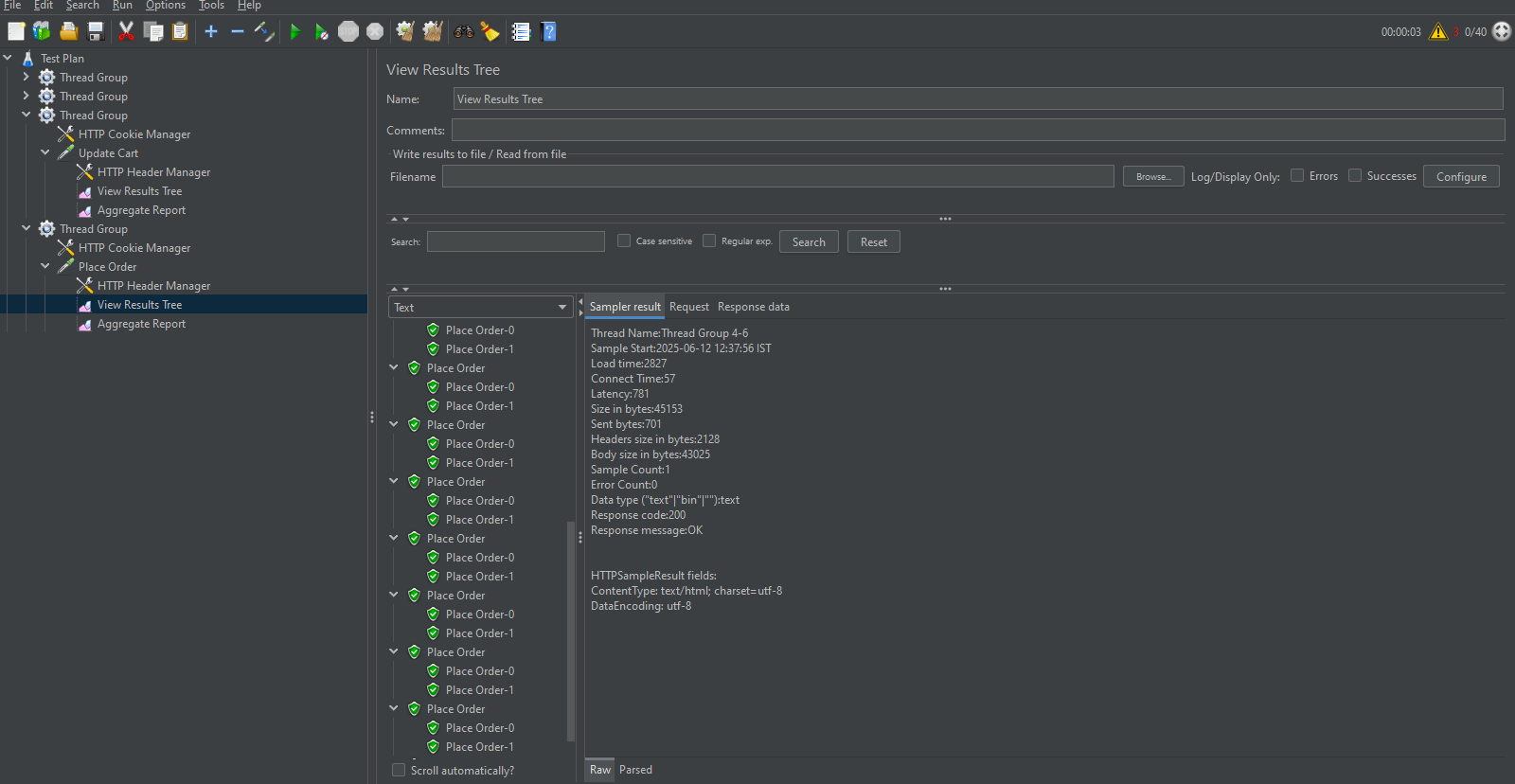


Fig. Placing order of JMeter

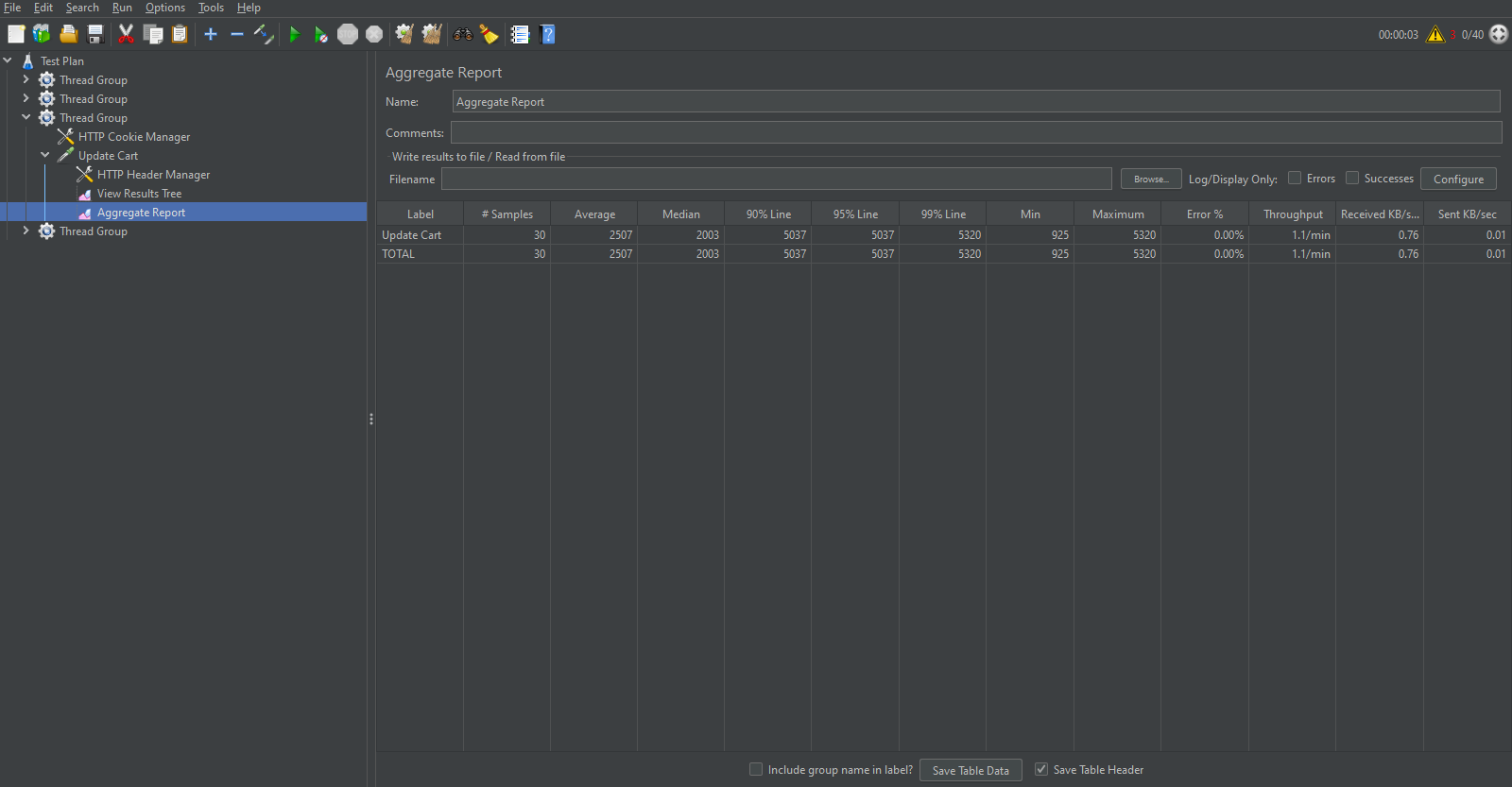


Fig. Final report page of JMeter

**REPORTING AND DOCUMENTATION**

**REPORTING**

Generating comprehensive test reports is crucial for documenting the outcomes of the testing process. The reports included:

* **Functional Test Report**: Detailed results of Selenium and Cucumber tests, including the pass/fail status of each test case, screenshots of failures, and any relevant logs.
* **Performance Test Report**: JMeter test results, including metrics such as response times, throughput, and resource utilization. The report highlighted any performance bottlenecks and provided recommendations for improvement.
* **Compatibility Test Report**: Playwright test results across different browsers and devices, detailing any compatibility issues and their impact on user experience.
* **Defect Logs**: Detailed records of issues identified during testing, including steps to reproduce, severity, status, and resolution.

**DOCUMENTATION**

Proper documentation ensures that all aspects of the testing process are recorded and can be referenced in the future. The documentation included:

* **Test Plan**: A comprehensive document outlining the strategy, scope, resources, schedule, and deliverables for the testing activities.
* **Test Cases**: Detailed scenarios for testing each feature, including preconditions, test steps, expected results, and actual results.
* **Test Summary**: An overall assessment of the testing effort, summarizing the results, defects found, and the effectiveness of the testing process.
* **User Manuals**: Guides for using the testing tools (Selenium, Cucumber, JMeter, Playwright) and understanding the test scripts.

The testing of demowebshop.tricentis.com utilized Selenium, Cucumber, JMeter, and Playwright to cover functional, performance, and compatibility aspects comprehensively. The process was well-planned and executed, with detailed reporting and documentation. Defects were systematically logged, tracked, and resolved, enhancing the website's overall quality. The testing provided valuable insights and recommendations, ensuring the website delivers a robust and reliable user experience.