جامعة الحسين التقنية
Al Hussein Technical University

QA Automation & API Testing Technical Report

Supervisor: Dr. Ashraf Smadi	
Zaid Abdullah Shishani	202508096

10/2/206

Acknowledgment

I would like to express my sincere gratitude to **Al Hussien Technical University** for providing a supportive academic environment and valuable resources that contributed significantly to the successful completion of this graduation project.

I am especially grateful to **Dr. Ashraf Smadi** for his guidance and for his continuous support and professional supervision throughout the project. His expertise and encouragement played a major role in strengthening the quality of the work and ensuring that the project was completed with a high standard of professionalism.

Contents

1. Title Page

Project name: Software Testing Assignment – Complete QA Suite
Test coverage: Full-Stack System (Demo Application)
Testing types: Manual Testing, UI Automation, API Functional Testing, and API Performance Testing(K6)

Prepared by: Zaid Abdullah Shishani
Role: QA Engineer
Date: 07/feb/2026

2. Introduction

This report documents the work completed for my program graduation project in the field of software quality assurance. The project involved testing a full-stack demo system that includes multiple backend services and a web-based user interface. The main goal was to apply real QA practices used in industry, including Manual Testing, UI automation, API testing, and performance testing(K6), to evaluate the system's functionality and reliability and overall quality.

3. Project Overview and Tool Selection

The system under test is a full-stack demo application that includes multiple backend services exposed through APIs, along with a user-facing web interface. Even though the application is a demo system, it was treated as a real production-like project by applying structured QA practices, including both manual and automated testing.

Tool Selection Rationale

The following tools and strategies were selected based on their reliability for professional QA workflows:

- **Manual Testing:** was used as the starting point to understand the system's behavior and to validate requirements, explore edge cases, and identify critical user flows before automation.

- **Selenium WebDriver with Microsoft Edge:** was used to automate UI testing and simulate real user interactions in a stable browser environment.
- **TestNG:** was selected as the test framework to manage execution, structure test suites, group scenarios.
- **Postman:** was used to design and validate API requests, including authentication flows, positive scenarios and negative test cases.
- **Newman:** was used to execute Postman collections through the command line, enabling repeatable runs and supporting CI/CD integration.
- **k6:** was used to perform performance testing, including smoke test and load test and stress test scenarios using scripts and threshold evaluation.
- **Nvidia app:** was used to record and collect evidence.

4. Test Data Preparation and Test Case Design

Test data preparation was handled in a structured way to ensure consistent execution and strong coverage across the different services in the system.

Manual Testing:

- Manual testing was used as the first step to understand the system behavior and confirm that the main flows worked correctly before automation.
- Exploratory testing helped identify issues that are not always obvious from requirements alone.
- Manual test scenarios were also used to define and prioritize the most critical automation cases later in the project.

UI Automation:

- The UI automation suite was built using a clean and modular structure to keep the code maintainable and easy to extend.

- The **Page Object Model (POM)** approach was applied to separate UI locators from test logic which improves the readability and reduces duplication.
- Test cases focused on the most important user journeys, and assertions were written carefully to ensure stability. As a result, automation runs remained consistent without flaky failures.

API Testing:

- The API testing suite included **more than 20 requests** that covered authentication flows and positive scenarios and negative validation cases.
- **Token-based authentication (Bearer/JWT)** was implemented and validated to ensure secure access to protected endpoints.
- Data-driven testing was applied using external **CSV and JSON** datasets to simulate multiple realistic input combinations without rewriting requests.
- Environment variables were used to support execution across different configurations, making the collection reusable and fully compatible with Newman CLI execution.

Performance K6 Testing:

- K6 was used to Validate and check API responsiveness and the performance of DummyJSON endpoints during smoke test and load test and stress test conditions.

5. Test Results

A) Manual Testing:

Manual test execution was performed on the SauceDemo web application to validate core functional flows and UI behavior. A total of **23 manual test cases** were executed covering critical areas such as login, product browsing, sorting, cart behavior, checkout, UI check.

During execution, **11 defects were identified** and documented in formal bug reports with reproduction steps, expected vs actual results, severity, priority, and supporting screenshots

Metric	Count
Total Manual Test Cases Designed	23
Total Manual Test Cases Executed	23
Passed	12
Failed	11
Blocked	0
Not Executed	0
Total Defects Logged	11

Pass Rate

- Pass Rate: 52%
- Fail Rate: 48%

Examples of defects reported include:

- Item not added to cart (High)
  bug report
- Cart not opening after adding items (High)
  bug report
- Checkout fields not working properly (High)
  bug report
- Incorrect product description/details (Medium)
  bug report
- Sorting not applied correctly (Medium)
  bug report
- UI alignment issues (Low)

Severity Distribution

Severity	count
High	4
Medium	3
Low	4
Total	11

Conclusion

Manual testing execution confirmed that the system contains multiple functional defects impacting core user flows, particularly in **cart operations** and **checkout completion**. The presence of **4 high-severity defects** indicates that the application is not ready for release without fixing the identified issues and performing re-testing.

B) UI Automation

Execution Summary

UI automation testing was executed on the SauceDemo web application using a Selenium + TestNG framework. The automation suite focused on validating key UI functionality through repeatable test execution.

The automated test run completed successfully with **11 automated test executed**, and the result was **PASS** with no failures, skips, or retries or blocked.

C) API Testing – Postman + Newman

API testing was executed for the DummyJSON API using Postman. The test suite was designed to validate key API areas including **Authentication, Products, Carts, Users**, and also included **negative test scenarios** and **data-driven execution**.

The collection was executed via Newman (CLI runner) to ensure repeatability and CI/CD readiness. The Newman report confirms that the full run completed successfully with **0 failed tests** and **0 skipped tests** and **0 blocked.**

API Newman Execution Metrics report

Metric	Result
Total Requests Executed	135
Total Iterations	5
Total Assertions	145
Failed Tests	0
Skipped Tests	0
Total Run Duration	29.5s
Average Response Time	143ms
Total Data Received	926.97 KB

Pass Rate

- API Request Success Rate: 100%
- Assertion Pass Rate: 100%
- Failed Assertions: 0

Conclusion

The API test execution was completed successfully using Newman. All requests and assertions passed, and the run produced a stable result with no skipped or failed tests. Performance indicators from the report also show strong responsiveness, with an average response time of 143 ms, supporting the reliability of the tested endpoints

D) Performance K6 Testing:

Execution Summary

Performance testing was executed using **k6** to evaluate the responsiveness and stability of the DummyJSON API under different load conditions.

Three performance scenarios were executed: **Smoke Test, Load Test, and Stress Test**, to validate baseline performance, typical user load behavior, and system behavior under high concurrency.

Test Scenarios Executed

Scenario	Load profile	Purpose
Smoke	1 VU for 30 seconds	Validate baseline stability and quick verification
Load	Ramp up to 15 VUs, hold, ramp down	Validate performance under expected load
Stress	Ramp up to 80 VUs, then ramp down	Identify system behavior under heavy traffic

Execution Results Summary

1) Smoke Test Results

Metric	Result
Total Requests	30
Errors	0%
Checks Passed	30
Checks Failed	0
Avg Response Time	9.43 ms
p95 Response Time	13.58 ms

2) Load Test Results

Metric	Result
Total Requests	3062
Iterations	1531
Max Virtual Users	15
Total Checks	3062
Passed Checks	3037
Failed Checks	25
Avg Response Time	170.95 ms
p95 Response Time	280.20 ms

3) Stress Test Results

Metric	Result
Total Requests	19614
Iterations	9807
Max Virtual Users	80
Total Checks	39228
Failed Checks	0
Avg Response Time	158.21 ms
p95 Response Time	200.03 ms
Max Response Time	760.41 ms

Thresholds & Pass/Fail Status

Scenario	Threshold	Result
smoke	http_req_failed<0.1%	Passed
smoke	P95 < 800ms	Passed
load	http_req_failed<0.1%	Passed
load	P95 < 1200ms	Passed
stress	P95 < 200ms	Achieved

Conclusion

The DummyJSON API demonstrated strong performance and stability across smoke, load, and stress conditions. Response times remained fast, error rate remained at 0%, and all performance KPIs were met. This indicates that the tested endpoints can handle both expected and high concurrency loads with consistent responsiveness.

6. Risks, Limitations, and Recommendations

Risks:

1. **Critical user flow failures (Cart & Checkout)**
   Several defects were identified in the cart and checkout workflows, including cases where items are not added to the cart, the cart cannot be opened, and checkout can proceed with invalid conditions. These issues directly impact the core purchase flow and represent a high release risk.

2. **Potential revenue and user trust impact**
   Failures in cart functionality and checkout completion may lead to lost transactions, customer dissatisfaction, and reduced confidence in the system's reliability.

3. **Data integrity and UI correctness issues**
   Defects related to corrupted product names, mismatched product images, and incorrect item descriptions can mislead users and reduce the perceived quality of the platform.

4. **Performance results variability**
   Performance testing was executed on a public demo API. Results may vary due to shared usage, server-side variability, and network conditions. This introduces risk when comparing results across different runs or environments.

5. **Limited automation coverage**
   UI automation execution was successful; however, the number of automated scenarios is currently limited. This may reduce regression detection effectiveness when future changes are introduced.

Recommendation:

The executed testing activities provided strong coverage across manual testing, API validation, UI automation, and performance testing. API and performance results were stable, and the automation execution completed successfully.

However, manual testing revealed multiple defects affecting the cart and checkout workflows, including several high severity issues that can prevent users from completing essential actions. For this reason, release approval is recommended only after fixing high severity defects, followed by re-testing and a full regression run to ensure stability.

7. References and Evidence

Testing Type	Evidence
Manual Test cases	https://docs.google.com/spreadsheets/d/1SBf8Q8PIgJ5jFuH_0_WDGu40U2JkOMxg/edit?usp=sharing&ouid=109518821743621391389&rtpof=true&sd=true
Bug Reporting	https://drive.google.com/file/d/12hLticmY_oTQkWx9zBsD2pxfZdKpgixY/view?usp=sharing
UI Automation	https://drive.google.com/file/d/1-NR-1GlDEqypMW4waX520dn3DCEMnDNJ/view?usp=sharing
API testing Newman and Postman	https://drive.google.com/file/d/136sDOhusMW9PD7i3cuV3RfuUSPo4Q1Br/view?usp=sharing
Performance Testing	https://drive.google.com/drive/folders/12n-JZa0mPePJV7Nf2nDVTum6-U6FehgU?usp=sharing
Manual Testing Excution	https://drive.google.com/drive/folders/1MPAtl7XsgIwbD4Q_kzq6T4E2ScESo5aq?usp=sharing