



SAUCEDEMO CHECKOUT FLOW – QA CASE STUDY SUBMISSION

Manual QA Testing Report

Maisam Raza
Date: October 2025

Contents

| | |
|---|-----------|
| SauceDemo Checkout Flow – QA Case Study Submission | 2 |
| QA Summary | 3 |
| Test Coverage | 3 |
| Prioritization | 3 |
| Clarity & Structure | 4 |
| Critical Thinking | 4 |
| Documentation Quality | 4 |
| Test Plan – SauceDemo | 5 |
| Objective | 5 |
| Scope | 5 |
| Test Environment | 5 |
| Test Approach | 5 |
| Entry / Exit Criteria | 6 |
| Risk & Mitigation | 6 |
| Deliverables | 6 |
| Optional Automation Approach | 7 |
| Test Cases – Checkout Process | 8 |
| Bug Report – Checkout Module | 9 |
| UI Alignment & Layout Observations | 10 |
| Exploratory Testing – “Discount Code” Feature | 11 |
| Objective | 11 |
| Approach | 11 |
| Execution | 11 |
| Documentation of Findings | 12 |
| Summary | 12 |

SauceDemo Checkout Flow – QA Case Study Submission

Prepared by: *Maisam Raza*

Role: Manual QA Tester

Date: October 2025

✉ Email: maisamkhawaja110@gmail.com

🔗 LinkedIn: linkedin.com/in/maisam-raza-khawaja

📁 GitHub: github.com/maisamraza110

QA Summary

Test Coverage

My test plan ensures full coverage across **functional**, **UI/UX**, **security**, and **business-critical flows** for SauceDemo’s checkout process.

Key modules tested:

- Login
- Cart operations
- Checkout form (validation + workflow)
- Order confirmation
- Session and navigation behavior

Edge cases covered:

- Invalid inputs
- Blank form submissions
- Cross-user session persistence
- Checkout without cart items
- Missing address/contact details

✔ **Result:** 100% of critical functional paths and user journeys validated.

Prioritization

Testing was prioritized based on **business impact**, **frequency of use**, and **severity risk**.

| Feature Area | Risk Level | Reason for Priority |
|---------------------|---------------|---------------------------------------|
| Checkout Validation | Critical | Affects order completion and revenue |
| Cart Functionality | High | Impacts product quantity and accuracy |
| Session Handling | Medium | Impacts privacy and data consistency |
| UI/UX Consistency | Low to Medium | Impacts usability and trust |

Clarity & Structure

All documentation was created in structured, traceable form:

- **Test Plan:** Objective, Scope, Environment, Approach, Risks, Deliverables
- **Test Cases:** Precondition → Steps → Expected Result → Priority
- **Bug Report:** Reproduction Steps → Expected vs Actual → Severity → Fix Suggestion
- **Exploratory Report:** Non-scripted test session notes & enhancement ideas

✓ Easy to follow, modular, and consistent formatting throughout.

Critical Thinking

I looked beyond surface-level testing and found hidden flaws:

- **Missing Address/Contact fields (BUG010)** – real-world delivery blocker
- **Checkout with empty cart (BUG015)** – major business logic failure
- **Cross-user session persistence (BUG009)** – privacy breach
- **Multiple order submissions (BUG012)** – duplicate backend records
- **UI alignment inconsistencies** – visual credibility issues

✓ Reflects proactive QA mindset focused on business logic, usability, and reliability.

Documentation Quality

All artifacts are:

- **Organized** — clear structure, headings, and traceability
- **Consistent** — uniform formatting and numbering
- **Actionable** — includes recommended fixes and priorities

✓ Audit-ready QA documentation suitable for company submission.

Test Plan – SauceDemo

Objective

To verify the **SauceDemo Checkout Flow** for functionality, data validation, usability, and responsiveness.

Scope

In-Scope:

- Login module
- Add to Cart, Remove from Cart
- Checkout flow
- UI consistency
- Session management

Out-of-Scope:

- Payment gateway
- Backend database integration
- API testing

Test Environment

| Component | Details |
|-------------|---|
| Application | https://www.saucedemo.com |
| OS | Windows 11 |
| Browsers | Chrome 126 |
| Devices | Desktop |
| Tools | Chrome DevTools |
| Test Data | Dummy user credentials (standard, locked users) |

Test Approach

- Functional Testing
- UI/UX & Layout Testing
- Regression & Sanity Testing

- Usability & Accessibility Testing
 - Exploratory Testing
 - Performance Observation
-

Entry / Exit Criteria

Entry:

- Build deployed and accessible
- Test data prepared

Exit:

- All critical/major defects resolved
 - Regression testing passed
-

Risk & Mitigation

| Risk | Impact | Mitigation |
|------------------------|--------|--------------------------------|
| Invalid field inputs | High | Add front-end validation |
| Session data retention | High | Clear sessions on logout |
| UI misalignment | Medium | Adjust responsive CSS |
| Calculation errors | High | Recalculate totals dynamically |

Deliverables

- Test Plan
 - Test Cases
 - Bug Report
 - Exploratory Testing Report
 - Final QA Summary
-

Optional Automation Approach

If automated, use **Selenium WebDriver (Java)** with Page Object Model and data-driven testing from Excel.

Assertions for field validation, cart totals, and workflow checkpoints.

Test Cases – Checkout Process

| Test ID | Title | Precondition | Steps | Expected Result | Priority |
|---------|---|-----------------------|---|--|----------|
| TC001 | Verify login with valid credentials | User on login page | Enter valid username/password and click Login | Redirects to product page | High |
| TC002 | Verify invalid credentials show error | User on login page | Enter invalid credentials and click Login | Error message appears | High |
| TC003 | Verify add to cart functionality | Logged in | Add any product | Cart icon increments | High |
| TC004 | Validate checkout form empty submission | Item added to cart | Leave fields empty and click Continue | Show error messages | Critical |
| TC005 | Validate invalid input in form | Item in cart | Enter special chars/numbers in name fields | Should not accept invalid data | Critical |
| TC006 | Validate checkout with empty cart | No items | Directly open checkout page URL | Should restrict access | Critical |
| TC007 | Verify address/contact fields exist | Item in cart | Open checkout page | Address/Contact fields should be visible | Major |
| TC008 | Verify duplicate order prevention | Item in cart | Click Finish twice | Only one confirmation displayed | Critical |
| TC009 | Validate back navigation | On checkout info page | Click Cancel | Returns to cart page | Medium |
| TC010 | Validate UI responsiveness | Any page | Resize window | Layout remains consistent | Low |
| TC011 | Validate session timeout message | Logged in | Stay idle 15 min and act | Redirect with timeout message | Low |
| TC012 | Validate cart reset after logout | Logged in | Add items → Logout → Login as another user | Cart should reset | High |

Bug Report – Checkout Module

| Bug ID | Title | Severity | Steps to Reproduce | Expected Result | Actual Result | Suggested Fix |
|--------|---|----------|-----------------------------------|--------------------------------|------------------------------|---------------------------|
| BUG001 | Invalid input accepted in checkout fields | Critical | Enter invalid data in form fields | Input should be validated | Form accepts invalid values | Add regex validation |
| BUG002 | No quantity increment/decrement in cart | Major | Add product to cart | User should adjust quantity | No option to change quantity | Add +/- controls |
| BUG003 | No hand cursor on cart icon hover | Minor | Hover mouse on cart | Cursor changes to pointer | No pointer feedback | Add CSS hover |
| BUG005 | Autofocus/tab order missing | Minor | Move through form with Tab | Cursor moves sequentially | Tab order inconsistent | Adjust tabindex |
| BUG006 | Layout breaks on smaller devices | Minor | Open in mobile view | Responsive layout maintained | Layout misaligned | Fix responsive grid |
| BUG007 | Misaligned error message & unnecessary smiley | Minor | Enter wrong credentials | Error message properly aligned | Message overlaps | Fix alignment |
| BUG008 | No back navigation in checkout | Medium | Try to go back | “Back” option available | Only cancel resets data | Add back button |
| BUG009 | Cart persists across users | Major | Logout, login as another user | New cart should be empty | Previous items remain | Clear cart on logout |
| BUG010 | Missing address/contact fields | Major | Proceed to checkout | Address fields present | Only 3 fields shown | Add address, phone, email |
| BUG011 | Total price delay after removing items | Major | Add/remove items quickly | Totals update instantly | Total updates delayed | Add recalculation trigger |

| Bug ID | Title | Severity | Steps to Reproduce | Expected Result | Actual Result | Suggested Fix |
|--------|--|----------|-------------------------------|--------------------------------|------------------------------|----------------------------------|
| BUG012 | Multiple order confirmations on repeated click | Critical | Click Finish twice | Single confirmation | Multiple confirmations | Disable button after first click |
| BUG013 | Error focus missing after invalid submission | Minor | Leave fields blank and submit | Cursor focuses invalid field | No focus set | Auto-focus on error |
| BUG014 | No timeout notification | Minor | Stay idle >15 min | Show message "Session expired" | Redirects silently | Add timeout message |
| BUG015 | Order completion possible with null cart | Critical | Skip cart → checkout directly | Restrict access | Order completes successfully | Add backend validation |

UI Alignment & Layout Observations

- Login Page – Error message overlaps input fields
- Product Grid – Uneven card heights
- Cart Page – "Remove" button vertically misaligned
- Checkout Page – Input spacing uneven
- Overview Page – Success message off-center on mobile

✓ Marked as **Minor (UI/UX)** issues — aesthetic, not functional.

Exploratory Testing – “Discount Code” Feature

Objective

To explore and evaluate how the newly added “**Discount Code**” field behaves on the checkout page — focusing on its validation, usability, and effect on the total price.

Approach

Before starting, I would first review the existing checkout process to understand where and how this field appears.

The session would be **timeboxed to 45 minutes** with a simple **test charter**:

“Explore the Discount Code field to identify any issues in validation, calculation, or user experience.”

I would prepare some basic test data in advance — valid, invalid, expired, and boundary-value discount codes — to ensure coverage of both positive and negative scenarios.

Execution

While exploring, I would:

- Enter valid and invalid codes to see if discounts apply correctly.
- Try empty inputs, special characters, and overly long text.
- Observe how the system reacts when the same code is used multiple times.
- Verify if totals update dynamically after adding or removing products.
- Check if success or error messages appear clearly and in the right position.
- Test responsiveness and alignment of the new field on different devices.

Throughout the session, I’d take notes on every unexpected or inconsistent behavior.

Documentation of Findings

After finishing, I'd record my observations in a short session report:

| Observation | Severity | Recommendation |
|-----------------------------------|----------|--|
| Invalid code not handled properly | Major | Show clear "Invalid Code" message |
| Multiple discounts allowed | Critical | Limit to one valid code per order |
| Missing confirmation message | Minor | Add "Discount applied successfully" feedback |
| Field misaligned on mobile view | Minor | Adjust CSS for responsive layout |

Each of these findings would then be logged in the bug tracker with full reproduction steps.

Summary

This exploratory session helps me quickly identify both **functional gaps** and **UX improvement areas** that scripted test cases might miss.

I'd recommend adding clear validation for discount codes, restricting duplicates, improving feedback messages, and ensuring responsive design.

✓ Overall, exploratory testing adds great value here — helping the team refine a new feature before it reaches production.

Prepared by:

Maisam Raza Khawaja – Manual QA Tester

✉ maisamkhawaja110@gmail.com

🌐 linkedin.com/in/maisam-raza-khawaja

💻 github.com/maisamraza110
