SAMPLE QA REPORT

This is what you get from us.

A real report from real test runs: automated, exploratory and accessibility testing of the checkout in our lab demo store, a small shop with bugs modelled on ones we often find in client stores. Client reports use the same format and stay confidential.

Report QAL-LAB-2026-11-02 · version 1.1 · prepared by QA Labs

Summary

RELEASE RECOMMENDATION

Do not release.

- Shoppers can set their own price or make the total negative (QA-001, QA-005).
- Declined cards and double clicks create orders that shouldn't exist (QA-002, QA-004).
- The last item in stock can be sold twice during a rush (QA-003).

All P1 defects must be fixed and retested before release. The seven automated-test defects have since been fixed and verified; six lower-severity findings remain open.

DEFECTS FOUND	FOUND BY	AUTOMATED PASS RATE	STABILITY
13: 2 critical, 3 high, 4 medium, 4 low	7 automated tests, 5 exploratory, 1 accessibility scan	1 / 8 (12.5%) → 8 / 8 after fixes	Each test run 10×: 70 / 70 failed before, 80 / 80 passed after

Test cases run

Eight automated checks, each tied to a business rule of the checkout. Every one was run 10 times to rule out timing flukes.

ID	Business rule checked	First run	Retest
TC-01	A paid order is created and stock goes down	Pass	Pass (10/10)
TC-02	The server ignores a price sent by the browser	**Fail · QA-001**	Pass (10/10)
TC-03	Zero, negative and fractional quantities are rejected	**Fail · QA-005**	Pass (10/10)
TC-04	Only one coupon per order	**Fail · QA-006**	Pass (10/10)
TC-05	Tax is charged on the discounted subtotal	**Fail · QA-007**	Pass (10/10)
TC-06	A declined card creates no order and keeps the stock	**Fail · QA-002**	Pass (10/10)
TC-07	Two shoppers cannot both buy the last unit	**Fail · QA-003**	Pass (10/10)
TC-08	Double-clicking Place order creates one order	**Fail · QA-004**	Pass (10/10)

Defects

SEVERITY: HOW BAD THE IMPACT IS

Critical — Money, data or security is directly at risk. Blocks release.

High — A core flow breaks or produces wrong results for many users. Blocks release.

Medium — Wrong behaviour with a workaround, or limited reach. Fix in the current sprint.

Low — Cosmetic, clarity or best-practice issue. Schedule when convenient.

PRIORITY: HOW SOON TO FIX IT

P1 — Fix before release

P2 — Fix this sprint

P3 — Backlog

They are rated separately: QA-007 is medium severity but P1, because wrong tax is a legal risk.

ID	Defect	Severity	Priority	Found by	Status
QA-001	The server accepts a price sent by the browser	Critical	P1	Automated	Fixed · verified
QA-002	A declined card still creates an order and removes stock	Critical	P1	Automated	Fixed · verified
QA-003	Two shoppers can both buy the last unit	High	P1	Automated	Fixed · verified
QA-004	Double-clicking Place order creates two orders	High	P1	Automated	Fixed · verified
QA-005	Zero, negative and fractional quantities are accepted	High	P1	Automated	Fixed · verified
QA-006	Coupons stack without limit	Medium	P2	Automated	Fixed · verified
QA-007	Tax is charged before the discount	Medium	P1	Automated	Fixed · verified
QA-008	Add to cart crashes while the cart is still being created	Medium	P2	Exploratory	Open
QA-009	An unknown coupon is ignored without any message	Medium	P2	Exploratory	Open
QA-010	Ordering with an empty cart shows “Payment failed: empty_cart”	Low	P3	Exploratory	Open
QA-011	A coupon can be applied to an empty cart	Low	P3	Exploratory	Open
QA-012	Adding the same item twice creates two cart lines	Low	P3	Exploratory	Open
QA-013	Page has no main landmark	Low	P3	Accessibility	Open

QA-001 **Critical** **P1** Cart API · Automated Fixed · verified

The server accepts a price sent by the browser

STEPS TO REPRODUCE

1. Create a cart.
2. Add 1 × Cap ($24.00) with an extra field in the request: price = 0.01.
3. Read the cart totals.

Reproducibility: Always (10/10)

EXPECTED

Subtotal $24.00 (the catalogue price).

ACTUAL

Subtotal $0.01, total $0.01.

EVIDENCE

```
POST /api/carts/{cart}/items {"sku":"CAP","qty":1,"price":0.01}
→ 200 {"subtotal":0.01,"tax":0,"total":0.01}
```

BUSINESS IMPACT

Anyone who edits the request can buy any product for a cent. Direct revenue loss.

SUGGESTED FIX

Ignore client-sent prices; always read the unit price from the catalogue on the server.

CAUGHT BY

Security testing →

QA-002 **Critical** **P1** Checkout · Automated Fixed · verified

A declined card still creates an order and removes stock

STEPS TO REPRODUCE

1. Add the last Logo tee (stock 1) to a cart.
2. Check out with a card that is declined.
3. List the cart's orders and read the stock.

Reproducibility: Always (10/10)

EXPECTED

HTTP 402, no order, stock still 1.

ACTUAL

402 returned, but a pending $21.64 order exists and stock is 0.

EVIDENCE

```
POST /api/carts/{cart}/checkout {"paymentToken":"tok_declined"}
→ 402 {"error":"card_declined"}

GET /api/orders?cart={cart}
→ 200 [{"status":"pending","total":21.64}]

GET /api/products  (Logo tee)
→ 200 {"stock":0}
```

BUSINESS IMPACT

Unpaid orders reach fulfilment and items show as sold out when they are not.

SUGGESTED FIX

Charge the card first; create the order and take stock only after the payment succeeds.

CAUGHT BY

API testing →

QA-003 **High** **P1** Checkout · Automated Fixed · verified

Two shoppers can both buy the last unit

STEPS TO REPRODUCE

1. Two carts each hold the last Logo tee.
2. Both check out at the same moment.

Reproducibility: Always (10/10)

EXPECTED

One order succeeds (201), the other is refused as out of stock (409).

ACTUAL

Both returned 201 and stock fell to -1.

EVIDENCE

```
POST /api/carts/{cart A}/checkout  (same moment)
→ 201 {"status":"paid","total":21.64}

POST /api/carts/{cart B}/checkout  (same moment)
→ 201 {"status":"paid","total":21.64}

GET /api/products  (Logo tee)
→ 200 {"stock":-1}
```

BUSINESS IMPACT

Overselling during busy sales: refunds, cancelled orders and unhappy customers.

SUGGESTED FIX

Re-check and take stock inside a lock or database transaction at the moment of purchase.

CAUGHT BY

Performance testing →

QA-004 **High** **P1** Checkout UI · Automated Fixed · verified

Double-clicking Place order creates two orders

STEPS TO REPRODUCE

1. Add a mug to the cart in the browser.
2. Double-click Place order.
3. List the orders.

Reproducibility: Always (10/10)

EXPECTED

1 order.

ACTUAL

2 paid orders of $13.53, so the shopper is charged twice.

EVIDENCE

```
Browser: double-click Place order, then GET /api/orders
→ 200 [{"status":"paid","total":13.53},{"status":"paid","total":13.53}]
```

BUSINESS IMPACT

Double charges, chargebacks and support tickets.

SUGGESTED FIX

Disable the button after the first click and send an idempotency key the server enforces.

CAUGHT BY

Web application testing →

QA-005 **High** **P1** Cart API · Automated Fixed · verified

Zero, negative and fractional quantities are accepted

STEPS TO REPRODUCE

1. Add a mug with quantity 0, then -3, then 2.5.

Reproducibility: Always (10/10)

EXPECTED

Each request is refused with HTTP 400.

ACTUAL

All accepted (200); the order total went negative.

EVIDENCE

```
POST /api/carts/{cart}/items {"sku":"MUG","qty":0}
→ 200 {"subtotal":0,"total":0}

POST /api/carts/{cart}/items {"sku":"MUG","qty":-3}
→ 200 {"subtotal":-37.5,"total":-3.09}

POST /api/carts/{cart}/items {"sku":"MUG","qty":2.5}
→ 200 {"subtotal":-6.25,"total":-0.52}
```

BUSINESS IMPACT

Shoppers can lower what they pay, and order data becomes unreliable.

SUGGESTED FIX

Validate quantity on the server as a whole number from 1 to the per-order limit.

CAUGHT BY

API testing →

Would your checkout pass these checks?

Get a free QA review →

QA-006 **Medium** **P2** Promotions · Automated Fixed · verified

Coupons stack without limit

STEPS TO REPRODUCE

1. Add 2 × Cap ($48.00).
2. Apply SAVE10 (10% off).
3. Apply WELCOME5 ($5 off).

Reproducibility: Always (10/10)

EXPECTED

The second coupon is refused (409); discount stays $4.80.

ACTUAL

Accepted (200); discount $9.80.

EVIDENCE

```
POST /api/carts/{cart}/coupons {"code":"SAVE10"}
→ 200 {"coupons":["SAVE10"],"discount":4.8,"total":47.16}

POST /api/carts/{cart}/coupons {"code":"WELCOME5"}
→ 200 {"coupons":["SAVE10","WELCOME5"],"discount":9.8,"total":42.16}
```

BUSINESS IMPACT

Margins leak on every order that finds a second code.

SUGGESTED FIX

Enforce one coupon per order on the server, including the same code twice.

CAUGHT BY

Software QA testing →

QA-007 **Medium** **P1** Totals · Automated Fixed · verified

Tax is charged before the discount

STEPS TO REPRODUCE

1. Add 2 × Cap ($48.00) and apply SAVE10 (−$4.80 → $43.20).
2. Read the tax at 8.25%.

Reproducibility: Always (10/10)

EXPECTED

Tax $3.56 (8.25% of $43.20), total $46.76.

ACTUAL

Tax $3.96 (8.25% of $48.00), total $47.16.

EVIDENCE

```
GET /api/carts/{cart}  (2 × Cap, SAVE10)
→ 200 {"subtotal":48,"discount":4.8,"tax":3.96,"total":47.16}
```

BUSINESS IMPACT

Every discounted order is over-taxed: customers are overcharged and tax filings are wrong. Medium severity, but P1 because of the legal exposure.

SUGGESTED FIX

Calculate tax on the subtotal after discounts.

CAUGHT BY

Regression testing →

QA-008 **Medium** **P2** Storefront · Exploratory Open

Add to cart crashes while the cart is still being created

STEPS TO REPRODUCE

1. Throttle the network so creating the cart takes 1.5 s.
2. Open the store and click Add to cart straight away.

Reproducibility: Always on a slow connection (cart request delayed 1.5 s)

EXPECTED

The click waits for the cart (or the button is disabled until it exists), then the mug is added.

ACTUAL

The request goes to /api/carts/undefined/items (404) and the page throws a JavaScript error; nothing is added and no message is shown.

EVIDENCE

```
POST /api/carts/undefined/items
→ 404 Console: Cannot read properties of undefined (reading 'toFixed')
```

BUSINESS IMPACT

Shoppers on mobile data tap Add to cart and nothing happens; some leave.

SUGGESTED FIX

Disable Add to cart until the cart exists, and handle failed responses with a visible message.

CAUGHT BY

Web application testing →

QA-009 **Medium** **P2** Promotions · Exploratory Open

An unknown coupon is ignored without any message

STEPS TO REPRODUCE

1. Add a mug.
2. Enter BOGUS as the coupon and click Apply.

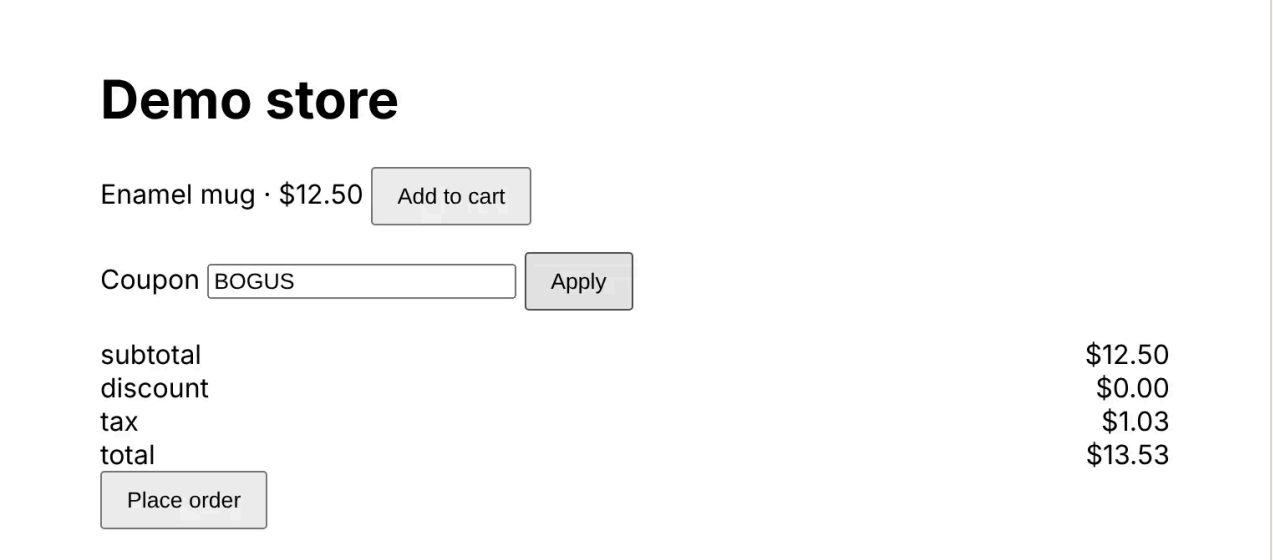
Reproducibility: Always

EXPECTED

A clear, announced message such as “This coupon code isn’t valid”.

ACTUAL

Nothing happens: the totals don’t change and no message appears (the API answers 404 unknown_coupon). Also fails WCAG 3.3.1 Error Identification.



Screenshot, Chromium 1280 px.

BUSINESS IMPACT

Shoppers think the site is broken and abandon, or contact support.

SUGGESTED FIX

Show the error next to the coupon field and announce it with role=“alert”.

CAUGHT BY

Usability testing →

QA-010 **Low** **P3** Checkout UI · Exploratory Open

Ordering with an empty cart shows “Payment failed: empty_cart”

STEPS TO REPRODUCE

1. Open the store.
2. Click Place order without adding anything.

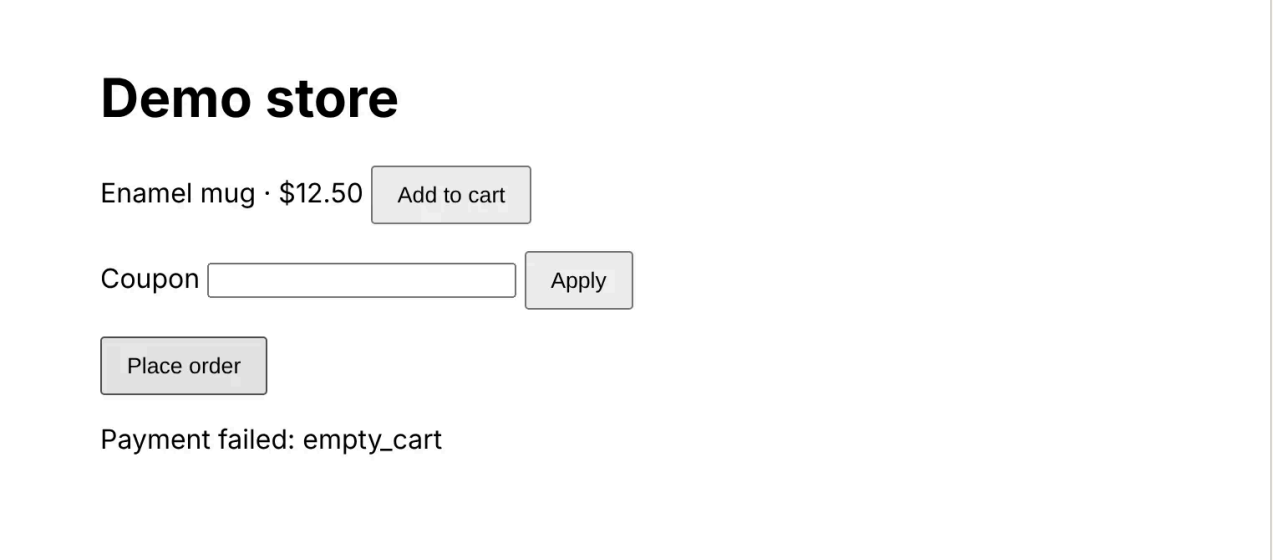
Reproducibility: Always

EXPECTED

Place order is disabled until the cart has an item.

ACTUAL

The button is enabled and the shopper sees a payment error with an internal code.



Screenshot, Chromium 1280 px.

BUSINESS IMPACT

Confusing, and it suggests a payment problem that doesn’t exist.

SUGGESTED FIX

Disable Place order for an empty cart and replace internal codes with plain messages.

CAUGHT BY

Usability testing →

QA-011 **Low** **P3** Promotions · Exploratory Open

A coupon can be applied to an empty cart

STEPS TO REPRODUCE

1. Create a cart.
2. Apply SAVE10 before adding items.

Reproducibility: Always

EXPECTED

Refused until the cart has items (or kept and re-validated at checkout).

ACTUAL

Accepted (200).

EVIDENCE

```
POST /api/carts/{empty cart}/coupons {"code":"SAVE10"}
→ 200 {"coupons":["SAVE10"],"subtotal":0,"total":0}
```

BUSINESS IMPACT

Low on its own, but it makes the stacking bug (QA-006) easier to abuse.

SUGGESTED FIX

Validate coupons against the cart contents, and again at checkout.

CAUGHT BY

API testing →

QA-012 **Low** **P3** Cart API · Exploratory Open

Adding the same item twice creates two cart lines

STEPS TO REPRODUCE

1. Add a mug.
2. Add a mug again.

Reproducibility: Always

EXPECTED

One line: Enamel mug × 2.

ACTUAL

Two separate lines of Enamel mug × 1.

EVIDENCE

```
POST /api/carts/{cart}/items {"sku":"MUG","qty":1}  (twice)
→ 200 {"items":["MUG ×1","MUG ×1"]}
```

BUSINESS IMPACT

Cluttered cart and harder quantity changes; can confuse stock limits.

SUGGESTED FIX

Merge lines for the same product and increase the quantity.

CAUGHT BY

Software QA testing →

QA-013 **Low** **P3** Accessibility · Accessibility Open

Page has no main landmark

STEPS TO REPRODUCE

1. Run axe-core 4.14 on the store, empty and with an item in the cart.

Reproducibility: Always

EXPECTED

Content sits inside landmarks (header, main), so screen-reader users can jump straight to it.

ACTUAL

axe: landmark-one-main and region (best-practice, moderate) on the heading, product line, coupon field and totals.

BUSINESS IMPACT

Slower navigation for screen-reader users; not a WCAG A/AA failure on its own.

SUGGESTED FIX

Wrap the store content in <main> (and the title in <header>).

CAUGHT BY

Accessibility testing →

Exploratory and accessibility testing

EXPLORATORY SESSION

Charter: explore the storefront as a real shopper would — empty states, wrong input, slow networks, keyboard only and a phone-sized screen — to find what the scripted tests don't cover.

Found: five defects (QA-008 to QA-012), including a crash on slow connections that no automated test caught.

CHECKED AND WORKING

- ✓ Coupon codes with spaces and lower case (" save10 ") are accepted correctly
- ✓ Keyboard only: Tab reaches Add to cart, Coupon, Apply and Place order in order, and Enter works
- ✓ No horizontal scrolling at 375 px (phone width)
- ✓ Order results are announced to screen readers (role="status")

ACCESSIBILITY (WCAG 2.2 AA)

Automated: axe-core 4.14.0, empty cart and with an item. 0 WCAG A/AA violations, 19 rules passed, 2 best-practice issues (no main landmark, content outside landmarks: QA-013).

Manual: one WCAG failure that tools can't detect: an invalid coupon gives no error message (3.3.1 Error Identification, QA-009). Keyboard access and status announcements work.

Automated scanners can't detect every accessibility problem (QA-009 is an example), which is why we always pair them with manual checks.

Retest after fixes

The fixed build of the store resolves QA-001 to QA-007. We ran the full suite again, 10 times over, and every check passed. The tests stay in the store's pipeline, so these bugs can't quietly return. QA-008 to QA-013 are open and scheduled by priority.

FIRST RUN	RETEST	REPEATED 10×	SUITE TIME
1 / 8 passed	**8 / 8 passed**	**80 / 80 passed**	**3.2 seconds**

Scope, environment and sign-off

IN SCOPE

- Cart, coupons, totals and checkout: API and browser
- Exploratory testing, desktop and 375 px
- Keyboard and automated accessibility checks

NOT IN SCOPE

- The real payment provider (a test stand-in was used)
- Load testing (see our peak-sale load test)
- Penetration testing, Safari, Firefox and real devices

ENVIRONMENT

Build	Lab demo store, buggy build, then fixed build
Date	2 November 2026
Browser	Chromium (Playwright 1.56)
Runtime	Node.js 22
Screens	1280 px desktop, 375 px phone
Accessibility	axe-core 4.14.0
Test data	3 products, 2 coupons, test cards (approved and declined)

RELEASE CRITERIA

- ✓ No open Critical or High defects
- ✓ All automated checks pass, 10 runs in a row
- ○ P2 items scheduled in the current sprint
- ○ Client acceptance

QAL-LAB-2026-11-02 · v1.1 · prepared by QA Labs

The services behind this report

Each finding above came from a different kind of testing. These are the same services we run for clients, alone or together.

- Security testing →
- API testing →
- Performance testing →
- Web application testing →
- Software QA testing →
- Regression testing →
- Usability testing →
- Accessibility testing →

Want a report like this for your product?

Start with a free QA risk review: a scoping call and a short written review of your release risks, with no obligation.

Get my free QA review → | QA testing services