Release Readiness Report

ShopLab v1.0 · seeded defects

06 Nov 2026, 09:46 AM IST
Prepared by Sharan L · R2C method

Demonstration report generated automatically from real test runs against ShopLab, a demo application with synthetic data. Not a client engagement.

No-Go

Do not release

- 6 open Critical/High defect(s): D01, D03, D05, D06, D08, D14
- performance requirement REQ-13 fails: Product browsing stays under 800 ms (95th percentile) with 50 users
- security requirement REQ-16 fails: Errors are handled safely without exposing internals
- security requirement REQ-17 fails: Security headers and secure session cookies are set
- security requirement REQ-18 fails: User input cannot inject code (SQL or script)

3/18	1 · 5 · 7 · 1	36/70	1485.7 ms
REQUIREMENTS PASSING	OPEN DEFECTS C · H · M · L	AUTOMATED CHECKS PASSING · 1 SKIPPED BY DESIGN	P95 AT 50 USERS

Executive summary

Do not release ShopLab v1.0. Testing found **14 open defects**, **6 of them Critical or High**. The most serious: any signed-in customer can read other customers' orders, names and postal codes; attackers can keep guessing passwords until they take over a customer's account; shoppers can stack one 10% code into 20% off or more, losing margin on every order. 15 of 18 requirements fail and 7 of 7 business risks remain open. Fix the Critical and High defects first, then re-run the full regression before reconsidering release.

Open defects (fix before release)

ID	SEVERITY	DEFECT AND BUSINESS IMPACT	EVIDENCE
D06	Critical	**Any user can read another user's order** Any signed-in customer can read other customers' orders, names and postal codes.	api:test_user_cannot_read_another_users_order
D01	High	**Account never locks after repeated wrong passwords** Attackers can keep guessing passwords until they take over a customer's account.	api:test_account_locks_after_five_failed_logins

D03	High	**Same discount code can be applied twice** Shoppers can stack one 10% code into 20% off or more, losing margin on every order.	api:test_same_discount_code_cannot_be_applied_twice
D05	High	**Quantity accepts 0 and negative numbers** A negative quantity can reduce the order total, letting shoppers pay less than the price.	api:test_invalid_quantity_is_rejected[zero] api:test_invalid_quantity_is_rejected[negative] +1 more
D08	High	**Stock goes negative under concurrent orders** During busy periods the shop sells items it does not have, forcing cancellations.	api:test_concurrent_orders_never_oversell api:test_database_has_no_negative_stock_after_concurrency
D14	High	**Customer name rendered as HTML (stored XSS)** A malicious name can run code in the browser and steal another session's data.	ui:a customer name with HTML is shown as text, not run (stored XSS) [desktop-chromium]
D02	Medium	**Sign-up accepts invalid email and 1-character passwords** Customers create accounts that cannot receive email and are trivial to break into.	api:test_signup_rejects_invalid_email[a@b] api:test_signup_rejects_invalid_email[no-tld] +6 more
D04	Medium	**Tax truncated instead of rounded** Tax is under-collected by a cent on many orders, creating accounting mismatches.	api:test_totals_for_several_items api:test_tax_is_rounded_to_nearest_cent +1 more
D07	Medium	**Malformed JSON returns 500 with stack trace** Error pages reveal internal code paths that help attackers map the system.	api:test_malformed_json_returns_400_without_internals[broken-brace] api:test_malformed_json_returns_400_without_internals[plain-text] +1 more
D09	Medium	**Listed price differs from database** Shoppers see one price and are charged another, causing complaints and disputes.	api:test_listed_prices_match_product_detail api:test_listed_prices_match_database
D10	Medium	**Product list slow under load** At 50 simultaneous shoppers pages take over 1.4 seconds, so campaigns lose customers.	perf:load p95 1485.7 ms > 800 ms
D11	Medium	**Place order button off-screen on phones** Phone shoppers cannot tap 'Place order' and cannot complete a purchase.	ui:full order: add, cart, checkout, confirmation [phone-chromium] ui:"Place order" is fully visible on a phone [phone-chromium]
D12	Medium	**Unlabelled fields and low-contrast prices** Screen-reader and low-vision users struggle to sign in or read prices (WCAG 2.2 failures).	ui:form fields on /login have visible labels [desktop-chromium] ui:form fields on /signup have visible labels [desktop-chromium] +1 more

D13	Low	**Missing security headers** Browser protections against script injection and clickjacking are switched off.	api:test_security_headers_are_present[Content-Security-Policy] api:test_security_headers_are_present[X-Content-Type-Options] +1 more

Business risk register

ID	RISK	LEVEL	STATUS	FAILING REQUIREMENTS
R1	**Customers are charged the wrong amount** Lost revenue, refunds and chargebacks; loss of trust at the point of payment.	High	Open	REQ-07, REQ-08, REQ-09
R2	**Customer data is exposed to other users or attackers** Privacy breach, regulatory exposure (GDPR / DPDP) and reputational damage.	High	Open	REQ-10, REQ-16, REQ-17, REQ-18
R3	**Accounts are taken over by password guessing** Fraudulent orders on real customers' accounts.	High	Open	REQ-02, REQ-03
R5	**The shop slows down at peak traffic** Abandoned carts during campaigns and launches.	High	Open	REQ-13
R4	**Stock is oversold** Orders that cannot be fulfilled, cancellations and support cost.	Medium	Open	REQ-11
R6	**Phone and assistive-technology users cannot buy** Lost sales and accessibility-law exposure (e.g. European Accessibility Act).	Medium	Open	REQ-14, REQ-15
R7	**Shoppers see a different price from the one charged** Complaints and pricing disputes.	Low	Open	REQ-05

Requirements traceability

ID	REQUIREMENT	TESTS	STATUS
REQ-01	Customers can sign in with valid credentials	5	pass
REQ-02	Accounts lock after 5 failed sign-in attempts	1	fail
REQ-03	Sign-up validates email format and password strength	12	fail
REQ-04	Product catalogue and search return correct results	4	pass

REQ-05	Listed prices match the system of record	2	fail
REQ-06	Cart quantities are whole numbers from 1 to 10	10	fail
REQ-07	Totals are correct and 8% tax is rounded to the cent	3	fail
REQ-08	A discount code can be used once per order	3	fail
REQ-09	Checkout validates shipping details and places the order	8	fail
REQ-10	Customers can only see their own orders and carts	3	fail
REQ-11	Stock is never oversold, even with simultaneous orders	3	fail
REQ-12	Order data is consistent in the database	2	pass
REQ-13	Product browsing stays under 800 ms (95th percentile) with 50 users	1	fail
REQ-14	Checkout is usable on phones	3	fail
REQ-15	Pages meet WCAG 2.2 AA (labels, contrast, keyboard)	2	fail
REQ-16	Errors are handled safely without exposing internals	3	fail
REQ-17	Security headers and secure session cookies are set	4	fail
REQ-18	User input cannot inject code (SQL or script)	2	fail

Performance

MEASURE (50 USERS, 30 S)	RESULT	TARGET	STATUS
95th percentile response	1485.7 ms	< 800 ms	fail
Average response	668.6 ms	—	
Throughput	54.4 requests/s	—	
Error rate	0.00%	< 1%	pass

Test execution

SUITE	PASSED	FAILED	SKIPPED
API + database (pytest)	32	26	0
Browser (Playwright)	4	7	1
Load (k6)	0	1	0

Residual risks and recommended mitigations

RISK	SEVERITY	MITIGATION
Passwords use a fast salted hash suited to a demo, not production	Medium	Move to Argon2 or scrypt before handling real customer accounts.

No rate limiting on the sign-in and order APIs	Medium	Add per-IP and per-account rate limits at the edge or in the app.
ShopLab browser tests run on Chromium only	Low	Add Firefox and WebKit projects (already used in the main Playwright suite).
Load measured on a 2-4 core CI runner, not production-like hardware	Low	Repeat the k6 profiles on a staging environment sized like production.

Scope, method and verdict rules

Method (R2C — Risk-to-Confidence): requirements and risks identified → tests designed per risk → automated execution on every change → evidence captured (screenshots, traces, logs) → defects analysed → fixes verified → release decision.

Tools: pytest + requests + jsonschema (API and SQL checks), Playwright + axe-core (browser, phone and WCAG 2.2), k6 (load), GitHub Actions (execution on every change).

Verdict rules: No-Go — any open Critical or High defect, or a failing performance or security requirement. **Go with risks** — only Medium or Low defects open. **Go** — no open defects above Low and every requirement passes.

Source run: local run · Generated by release-report/aggregate.py and render.py · ShopLab is a demonstration application built by Sharan L.