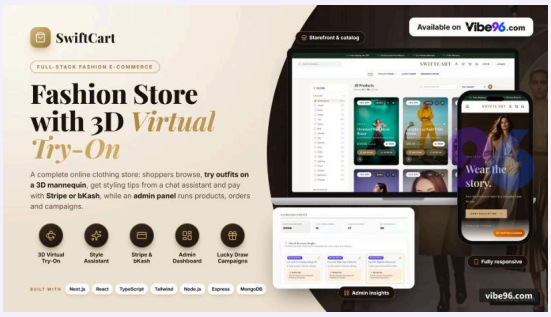


VIBE96 QUALITY REPORT

SwiftCart

Full-stack fashion e-commerce platform(Next.js, Express, MongoDB) with storefront, admin panel, payments and AI helpers.

Product ID	**B0446-76031**
Builder	**Nishat Raihana Prantik**
Version reviewed	**1.0.0 · released 11 Nov 2026**
Stack	**Next.js 16.4.2 (App Router, React 19) frontend with Express 4.22 backend · JavaScript (Node.js 20.9+) with TypeScript 5.9 on the frontend · MongoDB with Mongoose 8 (Atlas in production); minimum server version not stated in the digest**
Built with	**OpenAI Codex, Claude Code, Cursor, GitHub Copilot, Google Antigravity**
Review period	**26 Oct 2026 – 11 Nov 2026 · 3 rounds of changes**

80 / 100

Passed with 4 warnings

Pass mark 75 and no failing check.

Scores

Security	81
Code structure	78
Docs & completeness	73
Dependency health	72
Build & deploy	80

✓ **AI code audit**
6 checks · 26 Oct 2026

✓ **Installed & run**
11 Nov 2026

✓ **Demo vs delivered**
11 Nov 2026

✓ **Final sign-off**
11 Nov 2026

At a glance

- ✓ Installed and run from the delivered code — 8 of 8 checks answered yes.
- ✓ Live demo compared with the delivered code — 7 of 7 checks answered yes.
- ! 11 of 15 promised items delivered (setup guide, example config and the rest).
- ✓ Security scored 81/100 with no failing check.
- ! 4 open warnings — explained in part 1.

Check this report is genuine. Scan the code or open vibe96.com/report/QR-26-00120. The online copy always shows the latest status.

PART 1

AI code audit

The full source archive was read by our AI auditor against each check. It reads the code — it does not run it; running it is part 2.

CHECK	WHAT WE FOUND	SCORE	RESULT
Secrets & credentials	Every secret-scan candidate is a form field, a call to a password helper, or a read from process.env, so no real API key, JWT secret, SMTP password or database URI is committed. Both .env.example files contain only blank or placeholder values. The remaining issues are low-risk: a demo-login feature that could expose an admin account if enabled in production, and a dev-only reset token echoed in the UI.	92	Pass
Dependency vulnerabilities	Both the frontend and backend have package-lock.json files, and no wildcard or 'latest' versions are used. Caret ranges are used everywhere, and a few packages (multer 1.x, next-auth alongside a custom auth stack, helmet 7, express 4) are deprecated or behind current majors. I could not confirm a known critical vulnerability in a production path from the manifests alone.	72	Warning
Authorisation & access control	The visible code shows sound practices: bcrypt-style password matching, JWT with tokenVersion revocation, and a documented rule to guard every mutating route. However, none of the route files (adminRoutes, orderRoutes, etc.) were included in the digest, so per-route guards and ownership checks could not be traced. The one-click demo admin login is a risk if enabled in production, and the live demo likely has it on.	70	Warning
Database access rules	The project uses MongoDB via Mongoose with no raw SQL visible, and the shown auth code uses object-based, parameterized queries with atomic updates and lockout. However, most controllers, models and routes were not in the digest, so injection and scoping of per-user queries could not be fully verified, and several schema/doc inconsistencies exist.	82	Warning
Code structure & duplication	The project separates routes, controllers, services, models and middleware, and the frontend is split by feature. The digest shows some god-style files and duplicated helpers, and the seed scripts repeat password logic, which would make ongoing maintenance harder.	78	Warning
Build reproducibility	Manifests, scripts and env examples are coherent: Node >=20.9 is declared in both package.json files, Next 16.4.2 matches eslint-config-next, and lockfiles are present. Build output and vendored folders are not committed. HowToSetup.md and several referenced files were not in the digest, so setup order and prerequisites could not be fully confirmed.	80	Pass

Open warnings

Problems that did not stop the listing. They stay on the product page until a later version fixes them.

Warning 1 **Well-organized layers, but several oversized files and repeated demo/password logic**

The project separates routes, controllers, services, models and middleware, and the frontend is split by feature. The digest shows some god-style files and duplicated helpers, and the seed scripts repeat password logic, which would make ongoing maintenance harder.

Warning 2 **Authz cannot be fully verified: route files not visible; demo-login admin bypass is a risk**

The visible code shows sound practices: bcrypt-style password matching, JWT with tokenVersion revocation, and a documented rule to guard every mutating route. However, none of the route files (adminRoutes, orderRoutes, etc.) were included in the digest, so per-route guards and ownership checks could not be traced. The one-click demo admin login is a risk if enabled in production, and the live demo likely has it on.

Warning 3 **Mongoose-only access looks sound; limited visibility into most controllers and some gaps in schema/rules**

The project uses MongoDB via Mongoose with no raw SQL visible, and the shown auth code uses object-based, parameterized queries with atomic updates and lockout. However, most controllers, models and routes were not in the digest, so injection and scoping of per-user queries could not be fully verified, and several schema/doc inconsistencies exist.

Warning 4 **Lockfiles present, caret ranges throughout; a few risky or deprecated packages, no confirmed critical CVE**

Both the frontend and backend have package-lock.json files, and no wildcard or 'latest' versions are used. Caret ranges are used everywhere, and a few packages (multer 1.x, next-auth alongside a custom auth stack, helmet 7, express 4) are deprecated or behind current majors. I could not confirm a known critical vulnerability in a production path from the manifests alone.

Requirement list

What the builder agreed to deliver before uploading — 11 of 15 met.

✓	Complete source for every component	Met
✓	No real credentials in the archive	Met
!	Builds from a clean machine	Partial
✓	Database script or migrations	Met
!	Authentication and roles on every mutating endpoint	Partial
!	Pinned dependencies without critical CVEs (nice)	Partial
✓	Documentation set (nice)	Met
!	Working live demo	Partial
✓	README.md	Met
✓	HowToSetup.md	Met
✓	CLAUDE.md (or AGENTS.md) (nice)	Met
✓	CHANGELOG.md (nice)	Met
✓	.env.example	Met
✓	a project file	Met
✓	a project file (nice)	Met

PART 2

Human review

Reviewers from the Vibe96 team in Dubai worked through each question below; a final reviewer signed it off.

Stage 1 — Installed and run from the delivered code

Completed 11 Nov 2026

A reviewer installed the product on a clean machine, following only the guide that ships with it.

QUESTION	ANSWER	REVIEWER'S NOTE
Packages restored without errors	Yes	
Solution builds clean	Yes	
Database created from the shipped script or migrations	Yes	
First sign-in works with the documented steps	Yes	
Setup guide is accurate and complete	Yes	
Every module in the feature list opens and works	Yes	
No real secrets in the archive	Yes	
Branding and company details can be changed without code	Yes	

Stage 2 — Live demo compared with the delivered code

Completed 11 Nov 2026

A second reviewer checked that what the listing and demo show is what the buyer receives.

QUESTION	ANSWER	REVIEWER'S NOTE
Demo URL is reachable	Yes	
Demo credentials work	Yes	
Demo runs the same version as the archive	Yes	
All five differentiators are visible in the demo	Yes	
Screenshots match the demo	Yes	
Nothing in the demo is missing from the code	Yes	
Datasheet claims are accurate	Yes	

Changes asked for during review

None — approved on the first round.

Sign-off: approved for sale on 11 Nov 2026 with 4 warnings shown publicly.

What this report covers — and what it doesn't

- **It covers** version 1.0.0 exactly as the buyer downloads it: the source archive, the setup guide and the live demo on the dates above.

- **It is not** a penetration test, a performance or load test, or a legal review of third-party package licences.
- **Scores are a point in time.** Libraries age; a later version is audited again and gets its own report.
- **Your setup matters.** Hosting, configuration and changes made after purchase are outside this review.
- **How the score is made:** each AI check is scored 0–100 and weighted (security counts most). A listing needs 75 or more, no failing check, and the human review passed.
- **Always try the live demo** before buying, and ask the builder questions from the product page.