



QA Automation Baseline

An executable demonstration of risk-based test design, boundary coverage, and CI-ready evidence for a small pricing component.

Demonstration scope

Axtryll built a small quote-calculation component and an automated Node.js test suite. The suite covers nominal behavior, currency rounding, policy limits, and invalid inputs. It was executed locally on September 4, 2026.

12

automated checks

12

checks passed

0

failures

Coverage strategy

- Happy paths for base pricing and allowed discounts.
- Boundary values for minimum and maximum seat counts and discount limits.
- Validation failures for zero, fractional, excessive, and negative inputs.
- Deterministic two-decimal currency rounding.

Execution evidence

Area	Checks	Result	Risk addressed
Calculation	3	Passed	Incorrect subtotal, discount, or currency rounding
Allowed boundaries	3	Passed	Valid edge inputs rejected
Seat validation	3	Passed	Invalid quantity accepted
Price and discount validation	3	Passed	Invalid commercial rules accepted

CI pipeline design

Commit	Static checks	Unit tests	Artifact	Release gate
Pull request	Lint + type checks	12 deterministic cases	Machine-readable report	Block on failure

Next production steps

- Add API contract tests and authentication/authorization cases.
- Add browser coverage for the highest-value user flows.
- Track flaky-test rate, escaped defects, duration, and failure ownership.
- Run the suite on pull requests and retain artifacts for auditability.

Evidence note

This is original demonstration work, not a client project. The source file is retained with the portfolio workspace and the reported 12/12 result comes from an actual local test run.