E-Commerce Platform Test Plan

Project: ShopNow E-Commerce Platform
Version: 2.5.0
Date: January 20, 2026
QA Lead: Sarah Chen
Status: Draft

System Overview

This document outlines test cases for the ShopNow e-commerce platform. The system handles user authentication, product browsing, shopping cart management, payment processing, and order fulfillment. Testing covers web and mobile interfaces with focus on security, performance and user experience. Critical areas include payment gateway integration with Stripe/PayPal and role-based access for customers, vendors, and administrators.

Test Environment Configuration

Component	Version/Details	Status
Frontend	React 18.2, Node 18.x	Ready
Backend	Python 3.11, Django 4.2	Ready
Database	PostgreSQL 15.2	Ready
Payment Gateway	Stripe API v2023-10-16	Sandbox Mode

Test Cases

TC-001: User Login with Valid Credentials
Description: verify user can login using correct email and password
Steps: 1) Navigate to login page 2) enter email: testuser@example.com 3) enter password 4) click Login button
Expected: User should be redirected to dashboard, welcome message displayed
Priority: High | **Type:** Functional

TC-002: Login Attempt with Invalid Password
check system handles wrong password correctly. User tries to login with email test@shop.com but enters WRONG password like "wrongpass123". System should show error message "Invalid credentials" and NOT

allow access. Account should not be locked after first failed attempt but track the attempt.
Priority: High **| Type:** Security

TC-003: Password Reset with Email Verification
User forgot password and needs to reset it. Go to forgot password link, enter registered email, check that system sends reset email (might take 1-2 min in test env), email should have reset link that expires in 1 hour. Click link, enter new password min 8 chars with 1 uppercase 1 number 1 special char. Confirm password match. After reset user should be able to login with NEW password and OLD password should not work anymore!!!
Priority: Medium

TC-004: Payment Processing - Valid Credit Card
Preconditions: User logged in, items in cart (total $45.99)
Test the checkout flow with Stripe test card 4242 4242 4242 4242. Expiry: any future date, CVV: any 3 digits. Fill billing address, select shipping standard ($5.99). Order total should be $51.98. Click Pay Now - payment should process in under 3 seconds. User gets confirmation email and order appears in Order History with status "Processing". Stripe dashboard should show transaction.
Priority: CRITICAL **| Type:** Integration

TC-005: Payment Failure - Declined Card
Use Stripe test card 4000 0000 0000 0002 (declined card). Go through checkout process, when payment is submitted it should be DECLINED. Check error message shows "Your card was declined". Cart should still have items, user should be able to try different payment method. No order should be created in database. No confirmation email sent. User account balance unchanged.
Priority: High **| Type:** Error Handling

TC-006: Shopping Cart - Add Multiple Items and Update Quantity
Start with empty cart. Add product "Wireless Mouse" (price $29.99) qty 1. Then add "USB Cable" ($9.99) qty 2. Cart should show 3 items total, subtotal $49.97. Update mouse quantity to 3. Subtotal should recalculate to $109.96. Remove USB cable from cart. Cart should show only mouse qty 3 for $89.97. Test that cart persists if user logs out and logs back in - items should still be there!!!
Priority: Medium **| Type:** Functional

TC-007: Checkout with Empty Cart (Edge Case)
This is edge case testing - user tries to access checkout URL directly without items in cart. Maybe they bookmark the checkout page or use back button after completing order. System should detect empty cart and redirect to cart page with message "Your cart is empty. Please add items before checkout." Should NOT show payment form or allow payment submission. Check that direct API calls to payment endpoint also fail if cart empty.
Priority: Medium **| Type:** Edge Case

TC-008: Admin Dashboard Access Control (Role-Based)
Description: Verify role-based access permissions work correctly
Test Scenarios:
1) Login as CUSTOMER role (user@test.com) - try to access /admin/dashboard URL directly - should get 403

Forbidden or redirect to home page
2) Login as VENDOR role (vendor@test.com) - should access /vendor/dashboard but NOT /admin/dashboard
3) Login as ADMIN role (admin@test.com) - should access both /admin/dashboard and /vendor/dashboard
Check that API endpoints also enforce same permissions - customer should not be able to call admin APIs even with valid auth token
Priority: CRITICAL **| Type:** Security, Authorization

TC-009: SQL Injection Prevention in Search
Security testing for SQL injection vulnerabilities in product search. Enter malicious input in search box: ' OR '1'='1 Also try: admin'-- and '; DROP TABLE products;-- System should treat these as literal search strings and NOT execute as SQL commands. Should return no results or safe error message. Check database logs to confirm no SQL errors occurred. Also test search API endpoint with same payloads using Postman.
Priority: CRITICAL **| Type:** Security Vulnerability Testing

TC-010: Performance - Product Listing Page Load Time
Test page load performance for product catalog. Navigate to /products page with 50 items per page. Measure page load time using browser DevTools Network tab. Page should load completely in under 2 seconds on good internet connection. Images should lazy load. Check that database queries are optimized - use Django Debug Toolbar to verify no N+1 query problems. Test with 3G throttling - should still be usable in under 5 seconds.
Priority: Medium **| Type:** Performance

TC-011: Error Handling - Network Timeout During Payment
Simulate network failure during payment processing. User submits payment but API call to Stripe times out (can simulate by blocking network in DevTools after clicking Pay). System should show user-friendly error message "Payment processing timeout. Please check your connection and try again." Should NOT charge customer multiple times if they retry. Implement idempotency - check if order ID already processed. Log error to monitoring system but don't expose technical details to user.
Priority: High **| Type:** Error Handling, Reliability

TC-012: Mobile Responsiveness - Checkout Flow on iPhone
Test complete checkout on mobile device (iPhone 13, iOS 16, Safari). All steps should work: browse products, add to cart, view cart, enter shipping address (check that form fields are properly sized for mobile keyboard), enter payment info, submit order. Check that buttons are tappable (min 44px touch target), text is readable without zooming, no horizontal scrolling, images scale properly. Test both portrait and landscape orientations. Also test on Android Chrome for comparison.
Priority: High **| Type:** Cross-Platform, UI/UX

Testing Notes & Assumptions

- All tests should be run on staging environment before production deployment
- Payment testing uses Stripe test mode - no real charges will occur
- Test data will be reset every Monday at 2 AM EST
- Security tests (TC-009) require approval from security team lead before execution
- Performance benchmarks based on AWS t3.medium instance specs

- Known issue: Password reset emails may be delayed 2-3 minutes in test environment
- Cross-browser testing required: Chrome 110+, Firefox 115+, Safari 16+, Edge 110+