

Test Scenarios

The following test plan outlines various functional and non-functional tests identified through research. These tests can be further refined and augmented by leveraging Artificial Intelligence to uncover additional test scenarios, edge cases, and potential areas of failure. Furthermore, the "3 Amigos" process, involving collaboration between Business Analysts, Developers, and Testers, can significantly enhance the quality and coverage of these tests by bringing diverse perspectives and a shared understanding of the requirements. This collaborative approach ensures that tests are aligned with business objectives, technically feasible, and adequately address potential risks. The integration of AI and collaborative ceremonies represents a proactive and comprehensive strategy for ensuring the quality and reliability of the system under test.

1. UI/Visual Tests examples

Test ID	Test Description	Test Steps	Expected Results
UI - 01	Verify initial page load	1. Navigate to http://localhost:4000	Page loads with "Buy Stock" and "Sell Stock" buttons visible
UI - 02	Verify inventory table display	1. Navigate to http://localhost:4000	Table displays with columns: Stock Type, Price, Quantity, and Sale Date
UI - 03	Verify table data	1. Navigate to http://localhost:4000	Table shows existing inventory items with correct data
UI - 04	Verify "Buy Stock" button functionality	1. Click "Buy Stock" button	Buy form/modal appears with fields for Stock Type, Price, and Quantity
UI - 05	Verify "Sell Stock" button functionality	1. Click "Sell Stock" button	Sell form/modal appears with fields for Stock Type, Price, and Quantity

2. Functionality Tests - Buy Stock examples

Test ID	Test Description	Test Steps	Expected Results
BUY - 01	Buy wine stock with valid data	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter 100 as Price 4. Enter 50 as Quantity 5. Submit form	1. Form submits successfully 2. New entry appears in table 3. Table shows "wine" with price 100, quantity 50, and today's date
BUY - 02	Buy whisky stock with valid data	1. Click "Buy Stock" button 2. Select "whisky" as Stock Type 3. Enter 150 as Price 4. Enter 30 as Quantity 5. Submit form	1. Form submits successfully 2. New entry appears in table 3. Table shows "whisky" with price 150, quantity 30, and today's date
BUY - 03	Buy stock with empty fields	1. Click "Buy Stock" button 2. Leave fields empty 3. Submit form	Form shows validation errors and doesn't submit
BUY - 04	Buy stock with negative price	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter -100 as Price 4. Enter 50 as Quantity 5. Submit form	Form shows validation error for Price field and doesn't submit
BUY - 05	Buy stock with zero quantity	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter 100 as Price 4. Enter 0 as Quantity 5. Submit form	Form shows validation error for Quantity field and doesn't submit

3. Functionality Tests - Sell Stock examples

Test ID	Test Description	Test Steps	Expected Results
SELL - 01	Sell wine stock with valid data	1. Click "Sell Stock" button 2. Select "wine" as Stock Type 3. Enter 30 as Price 4. Enter -35 as Quantity (negative to indicate selling) 5. Submit form	1. Form submits successfully 2. New entry appears in table 3. Table shows "wine" with price 30, quantity -35, and today's date
SELL - 02	Sell whisky stock with valid data	1. Click "Sell Stock" button 2. Select "whisky" as Stock Type 3. Enter 150 as Price 4. Enter -20 as Quantity (negative to indicate selling) 5. Submit form	1. Form submits successfully 2. New entry appears in table 3. Table shows "whisky" with price 150, quantity -20, and today's date
SELL - 03	Sell stock with empty fields	1. Click "Sell Stock" button 2. Leave fields empty 3. Submit form	Form shows validation errors and doesn't submit
SELL - 04	Sell stock with negative price	1. Click "Sell Stock" button 2. Select "wine" as Stock Type 3. Enter -100 as Price 4. Enter -50 as Quantity 5. Submit form	Form shows validation error for Price field and doesn't submit
SELL - 05	Sell stock with positive quantity	1. Click "Sell Stock" button 2. Select "wine" as Stock Type 3. Enter 100 as Price 4. Enter 50 as Quantity (should be negative for selling) 5. Submit form	Form shows validation error for Quantity field or converts to negative and submits

4. Edge Cases examples

Test ID	Test Description	Test Steps	Expected Results
EDGE - 01	Test with very large numbers	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter 999999999 as Price 4. Enter 999999999 as Quantity 5. Submit form	System handles large numbers appropriately
EDGE - 02	Test with special characters	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter "\$100" as Price 4. Enter "50*" as Quantity 5. Submit form	System shows validation errors for invalid characters
EDGE - 03	Test with decimal values	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter 100.50 as Price 4. Enter 50.5 as Quantity 5. Submit form	System handles decimal values appropriately

5. More examples of some other edge test cases

Test ID	Test Description	Test Steps	Expected Results
---------	------------------	------------	------------------

EDGE - 04	Maximum integer values	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter Number.MAX_SAFE_INTEGER (9007199254740991) as Price 4. Enter Number.MAX_SAFE_INTEGER as Quantity 5. Submit form	System either handles these extremely large numbers appropriately or provides clear validation error
EDGE - 05	Boundary values combination	1. Click "Buy Stock" button 2. Select "wine" as Stock Type 3. Enter "0.01" as Price (minimum valid price) 4. Enter "999999999" as Quantity (very large) 5. Submit form	System handles combination of boundary values appropriately
EDGE - 06	Non-existent stock type	1. Click "Buy Stock" button 2. Try to select a non-existent stock type (if possible) 3. Enter "100" as Price 4. Enter "50" as Quantity 5. Submit form	System prevents selection of invalid stock types

6. API test examples

Test ID	Test Description	Test Steps	Expected Results
API - 01	Get all stocks without filter	1. Send GET request to http://localhost:4001/stocks	1. Status code 200 2. Response body contains array of stock objects 3. Each stock has type, price, quantity, and date properties
API - 02	Get stocks with valid client_id	1. Send GET request to http://localhost:4001/stocks?client_id=valid_id	1. Status code 200 2. Response body contains only stocks for the specified client
API - 03	Get stocks with invalid query parameter format	1. Send GET request to <a href="http://localhost:4001/stocks?client_id=<script>alert('xss')</script>">http://localhost:4001/stocks?client_id=<script>alert('xss')</script>	1. Status code 400 or sanitised parameter handling 2. No XSS vulnerabilities triggered
API - 04 - Happy path	GET /stocks endpoint	1. Send GET request to http://localhost:4001/stocks?client_id=	API returns list of stocks in JSON format
API - 05 - Happy path	POST /stocks endpoint with valid data	1. Send POST request to http://localhost:4001/stocks with valid stock data	API creates new stock and returns success response
API - 06 - Happy path	POST /stocks endpoint with invalid data	1. Send POST request to http://localhost:4001/stocks with invalid stock data	API returns appropriate error response

7. Performance test examples

Test ID	Test Description	Test Steps	Expected Results
PERF - 01	Page load time	1. Measure time to load the application	Page loads within acceptable time (< 2 seconds)

PERF - 02	Form submission time	1. Submit a form to buy/sell stock 2. Measure response time	Form submission completes within acceptable time (< 1 second)
-----------	----------------------	--	---

8. Concurrency test examples

Test ID	Test Description	Test Steps	Expected Results
CONC - 01	Multiple simultaneous transactions	1. Open multiple browser instances 2. Submit buy/sell transactions simultaneously	All transactions are processed correctly