

Manual Testing for NopCommerce E-commerce Platform Using Jira

Table of Content:

1. Overview
2. Objectives
3. Environment and Tools
4. Methodologies
5. Backlog Organization
6. Key Features
 - 6.1. User Stories
 - 6.2. Test Cases
7. Test Metrics and Reporting
8. Conclusion

1. Overview

This project is a comprehensive manual testing initiative for the nopCommerce e-commerce platform. It was designed and implemented using Jira. The project focuses on validating important functionality features such as:

- User registration,
- Login,
- Logout,
- Account Management,
- Product search,
- Product catalog,
- Wishlist,
- and Add-to-cart.

2. Objectives

- Ensure Quality: Validate that the nopCommerce platform meets the required functionality and user experience expectations.
- Documentation: Create detailed test cases and scenarios for each major feature.
- Agile Process Implementation: Demonstrate the use of agile practices by organizing and managing test cases as part of the project backlog in Jira.
- Skill Showcase: Highlight proficiency in manual testing, test case design, and agile project management using Jira.

3. Tools and Environment

- Manual Testing: All test cases were executed manually.
- Jira: Used for managing overall backlog (user stories, test cases, and bugs).
- nopCommerce Platform: The live e-commerce website serving as the test subject.
- Browsers: Google Chrome Version 133.0.6943.99 (Official Build) (64-bit).
- Operating System: Windows 11 (primary).

4. Methodologies

1. Agile Approach:
 - The project was organized using agile principles with a defined backlog, sprint planning, and iterative execution.
2. Test Case Design:
 - Each functionality (e.g., registration, search, add to cart) was broken down into user stories and corresponding test cases.
3. Jira for Test Management:
 - Test cases were documented as tasks.
 - Backlog items were organized into Epics and grouped with labels/components for better traceability.

5. Backlog Organization

- Epics:

Major features were grouped under epics such as "User Registration", "Logged In User", "Product Catalog & Search", "Wishlist", and "Cart & Checkout".

- User Stories and Acceptance Criteria:

Each feature was detailed using user stories that followed the format:

“As a [user], I want [feature] so that [benefit].”

Acceptance criteria were defined to specify the conditions for test success.

- Test Cases:

Detailed test cases were created for each user story and documented in Jira as tasks for individual test steps. For example:

- Registration Test Cases,
- Login & Logout Test Cases,
- User Account Management Test Cases,
- Product Search,
- Product Details Test Cases,
- Add To Cart & Checkout Test Cases,
- Wishlist Test Cases.

- Bugs:

Detailed bug reports were created for each defect and documented in Jira as issue.

6. Key Features

1. User Registration

- Epic

Manage and validate all scenarios related to user sign-up, including form validations and email confirmations.

- User Story

As a new customer, I want to register an account so that I can create a personalized profile, receive special offers, and make purchases on the website.

Acceptance Criteria:

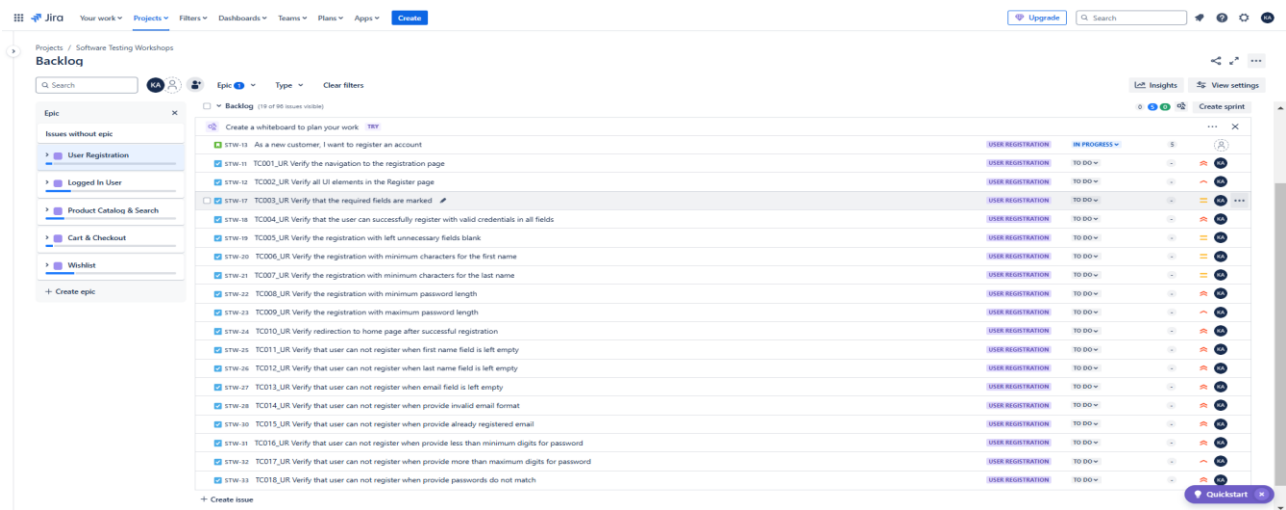
AC1: The registration page should be accessible from the homepage.

AC2: All required fields (e.g., First Name, Last Name, Email, Password) must be present and validated.

AC3: Upon submitting valid registration data, the system should create a new account and display a success message.

AC4: An email confirmation (if applicable) is sent to the user's provided email address.

AC5: Invalid or missing data should trigger appropriate error messages.



User Registration Epic

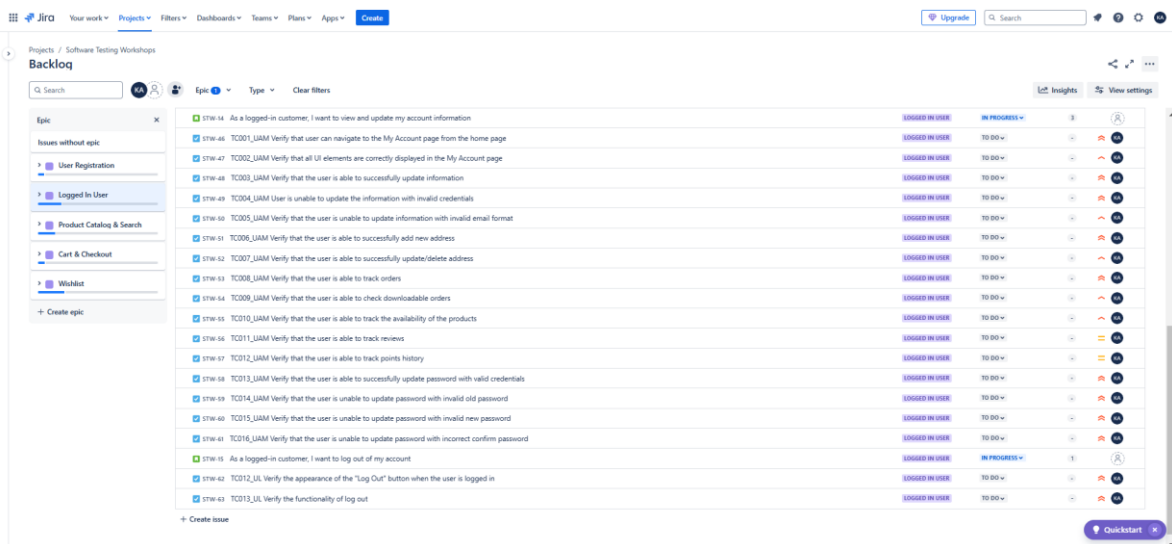
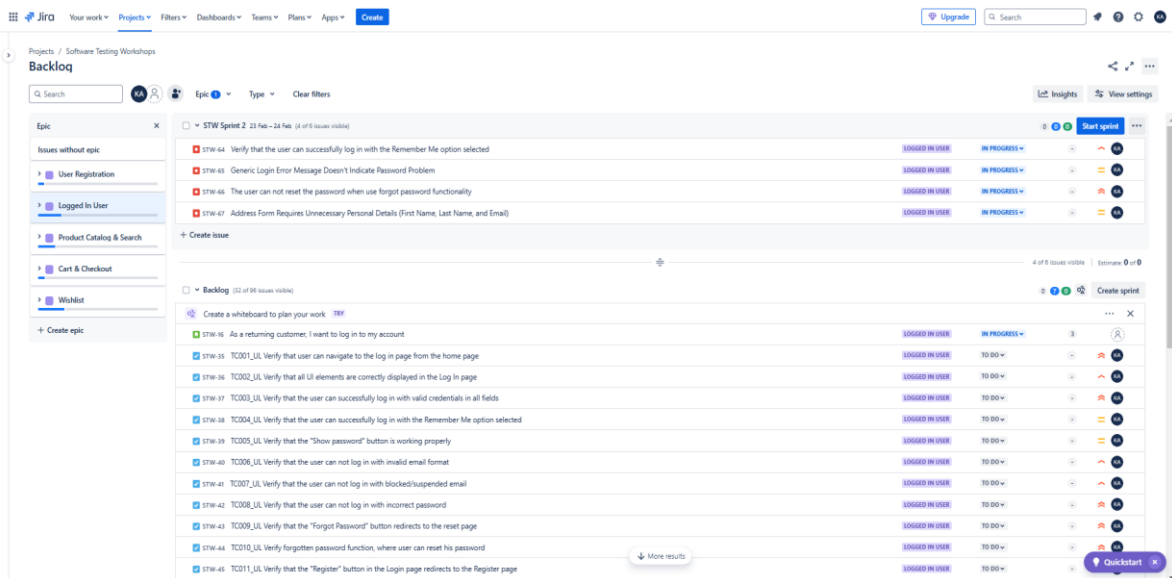
2. Logged In User

- Epic

Ensure that login, logout, and account management features function correctly and securely.

- User Stories

- As a returning customer, I want to log in to my account so that I can access my personalized dashboard, view order history, and make new purchases.
- As a logged-in customer, I want to view and update my account information so that I can manage my personal details, addresses, and order history effectively.
- As a logged-in customer, I want to log out of my account so that I can ensure the security of my personal information when I am done browsing.



Logged In User Epic

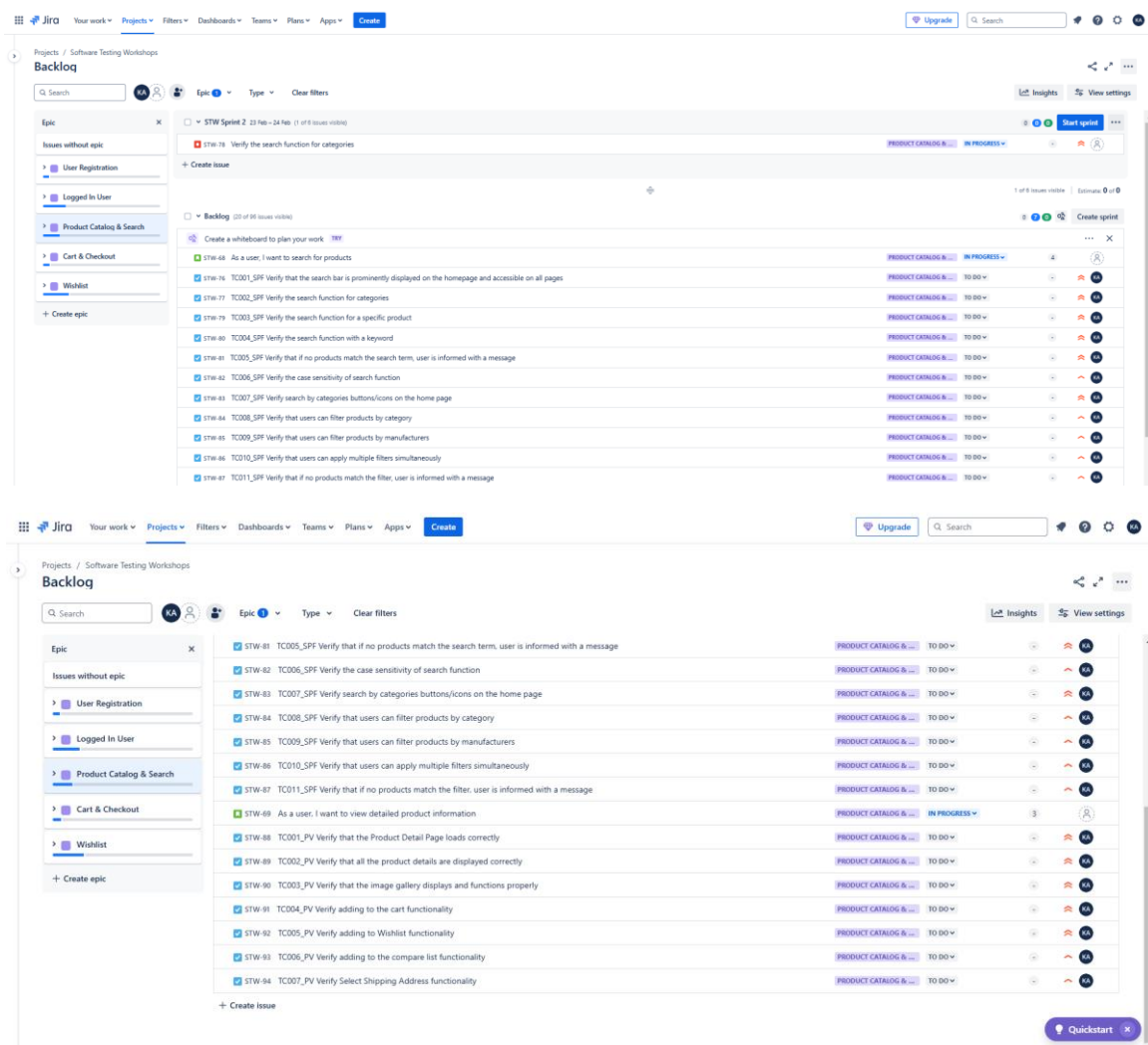
3. Product Catalog & Search

- Epic

Validate the search functionality, filters, and product listing details, ensuring accurate and user-friendly navigation.

- User Stories

- As a user, I want to search for products so that I can quickly find the items I need without browsing through multiple pages.
- As a user, I want to view detailed product information so that I can make informed decisions before adding a product to my cart.



Product Catalog Epic

4. Cart & Checkout

- Epic

Confirm that the add-to-cart, checkout process, and order confirmation are executed without issues.

- User Story

As a user, I want to add products to my shopping cart so that I can compile a list of items I wish to purchase in one transaction.

Acceptance Criteria:

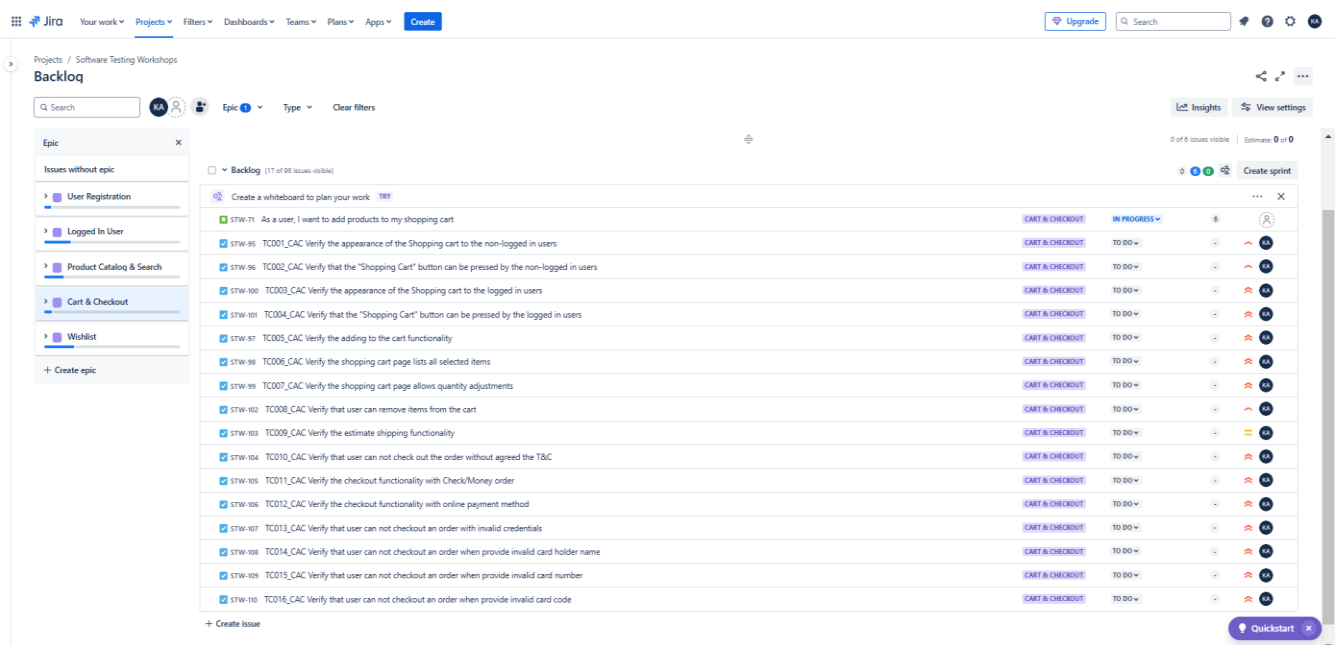
AC1: An “Add to Cart” button is clearly visible on the product details page and possibly on product listings.

AC2: When a product is added, the cart icon (usually in the header) updates to reflect the current number of items.

AC3: A confirmation message or mini cart popup appears after adding an item.

AC4: The shopping cart page lists all selected items, allows quantity adjustments, and displays the subtotal and total price.

AC5: Users can remove items from the cart directly from the shopping cart page.



Cart & Checkout Epic

5. Wishlist

- Epic

Verify that users can add, view, and remove items from the Wishlist, with session persistence and duplicate prevention.

- User Story

As a logged-in user, I want to add products to my Wishlist so that I can save items for future purchase or reference.

Acceptance Criteria:

AC1: A Wishlist icon or “Add to Wishlist” button is available on product listings and product details pages.

AC2: When a product is added, the user receives a confirmation message that the item has been saved.

AC3: The Wishlist is persistent across user sessions and can be accessed from the user account menu.

AC4: Users can remove products from their Wishlist.

AC5: The Wishlist displays key product details (image, name, price) and links to the product details page.

The screenshot displays the Jira interface for a project named 'Software Testing Workshops'. The 'Backlog' view is active, showing a list of issues under the 'Wishlist' epic. The epic is titled 'STW-70 As a logged-in user, I want to add products to my Wishlist' and is currently in 'IN PROGRESS' status. Below the epic, there are seven issues, each with a checkbox, a title, and a status. The issues are:

- STW-113 TC001_WL Verify the appearance of the Wishlist button (WISHLIST, TO DO)
- STW-111 TC002_WL Add Product to Wishlist (WISHLIST, TO DO)
- STW-112 TC003_WL Add Product to Wishlist (WISHLIST, TO DO)
- STW-114 TC004_WL Remove Product from Wishlist (WISHLIST, TO DO)
- STW-115 TC005_WL View Wishlist Contents (WISHLIST, TO DO)
- STW-116 TC006_WL Wishlist Persistence Across Sessions (WISHLIST, TO DO)
- STW-117 TC007_WL Prevent Duplicate Entries in Wishlist (WISHLIST, TO DO)

The interface also includes a sidebar with navigation links for 'User Registration', 'Logged In User', 'Product Catalog & Search', 'Cart & Checkout', and 'Wishlist'. A 'Quickstart' button is visible at the bottom right.

Wishlist Epic

7. Test Metrics and Reporting

Test Coverage:

- Total Test Cases: 93
- Test Cases Executed: 93
- Pass Rate: 93%
- Fail Rate: 7%

Defect Metrics:

- Bugs Reported: 5
- Critical: 1
- Major: 2
- Minor: 2

8. Conclusion

This project demonstrates my ability to plan, document, and execute manual test cases for an e-commerce platform. Using Jira to manage rapid tests, I successfully created detailed user stories, organized test cases, and ensured that the core functionality of the nopCommerce platform was fully validated. This project not only demonstrates my technical testing skills, but also demonstrates my ability to effectively manage and communicate project status.

[Jira Backlog View](#)