Software Test Description



**Otaku House Website**

**TABLE OF CONTENT**

1. **Introduction**
   1. **Scope**
2. Product Display and Listing:

* The website shall display the latest products available for purchase, which includes various categories such as figures, puzzles, and apparel.
* Each product should have an image, name, price, and rating displayed on the initial listing page.

1. Product Search Functionality:

* Users shall be able to search for products using keywords or product names.
* The search function should return relevant products based on the query input by the user.

1. Product Detail Page:

* Upon clicking a product, users shall be taken to a product detail page that includes an enlarged image, detailed description, price, availability status, and user reviews.
* The product detail page will allow users to select the quantity and add the product to their shopping cart.

1. Shopping Cart:

* Users shall be able to view items they have added to their shopping cart.
* The cart will display the product image, name, price, selected quantity, and the subtotal for all items.
* Users can proceed to checkout from the shopping cart page.

1. User Reviews:

* Registered users shall have the option to write and submit reviews for products.
  1. Glossary

a. API : A set of routines, protocols, and tools for building software and applications. It allows different software programs to communicate with each other.

b. Browser Compatibility: The ability of a website to function properly across different web browsers.

c. Cart: A virtual basket that holds items a user intends to purchase.

d. Checkout: The process through which a customer finalizes their purchase, including payment and delivery information.

e. Functional Testing: A type of testing that involves ensuring that the functions of the software application operate in conformance with the requirement specification.

1. Overview
   1. Background

Otaku House is an e-commerce platform dedicated to fans of anime, manga, and Japanese pop culture.

The website has become a popular destination for purchasing anime figures, apparel, and collectibles. It boasts a wide range of products that cater to a diverse fanbase, from casual enthusiasts to serious collectors.

* 1. Goals

The primary goals of the software testing for Otaku House are as follows:

1. Verify Functional Integrity:

* To ensure that all features of the website, such as product listings, search functionality, user accounts, shopping cart operations, and checkout processes, work as intended.

1. Enhance User Satisfaction:

* To provide a seamless, intuitive, and enjoyable shopping experience for all users, regardless of their device or browser choice.

1. Ensure Security Compliance:

* To protect sensitive user information, including payment details, and to comply with data protection regulations.

1. Ensure Security Compliance:

* To confirm that the website loads and performs quickly and reliably, even under high traffic conditions.

1. Optimize Performance:

* To ensure that users can navigate the website easily to find products, information, and customer support.

1. Facilitate Easy Navigation:

* To integrate a user review system that is simple to use and fosters authentic feedback on products.

1. Test strategy
   1. Test tree
2. Login page:
   1. Sign in with email

* Login with existing username and password.
* Login with correct username and wrong password.
* Login with not exist username.

1. Search product:
   1. With account

* Search valid product with login account
* Search valid product without login account

1. Product detail page:
   1. Product image and information

* Checking the correctness of products

1. Buy a product:
   1. Logging in to the account
   2. Not Logging in to the account
2. Shopping Cart
   1. Cart persistence
3. Not Logging in to account
   * Check if Items remain in the Cart after exit from the website
4. Logging in to account
   * Check if Items remain in the Cart after exit from the website
5. Update quantity:

* The price is change after update the quantity
* the updating quantity is saved after Logging out and login again

1. Remove item from Cart
2. if there is 1 item

* The Cart is “empty “and the subtotal is 0 and navigate to “start shopping page.”

1. if there is more than 1 item

* Removing some of the items and the subtotal is updating, and the page still Exists. (Cart persistence).

1. Checkout:
2. Logging in to account
3. Without logging to account

Non functional tests:

1. Maintainability Testing:

* Change Management Testing: Evaluating the ease with which new changes can be implemented into the existing system without introducing new issues.
* Documentation Review: Checking that all system documentation is clear, correct, and helpful.

1. Localization Testing:

* Currency Testing: If the website supports multiple currencies, test that all currency-related functionality works as expected.

1. Compatibility Testing:

* Browser Compatibility: Ensure that the website works across different web browsers like Chrome, Firefox, Safari, and Edge.

E2E tests:

1. Complete Purchase Cycle:

* User logs into their account.
* User searches for a specific product.
* User selects a product and adds it to the cart.
* User proceeds to checkout.
* User fills in payment information (mock or sandbox environment).
* User completes the purchase.
* Verify confirmation and summary of the order is displayed.
* User logs out.

1. Product Review by Logged-In User:

* User logs into their account.
* User navigates to a previously purchased product.
* User submits a review and rating for the product.
* Verify the review is posted under the product.

1. Traceability

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Parent Requirement** | **Child Requirement** | **Explanation Function** | **Script** | **Tag** | **Test Case** | **Purpose** |
| 1. Login Page | 1.1. Sign in with email | Login with existing username and password. | TS001 | TC001 | Verify user can log in with valid credentials | To ensure that users with valid credentials can access their accounts. |
| 1. Login Page | 1.1. Sign in with email | Login with correct username and wrong password. | TS002 | TC002 | Ensure login fails with incorrect password | To confirm the system security by not allowing access with incorrect passwords. |
| 1. Login Page | 1.1. Sign in with email | Login with not exist username. | TS003 | TC003 | Attempt login with a username that does not exist | To verify the system handles non-existent user data correctly. |
| 2. Search Product | 2.1. With account | Search valid product with login account | TS004 | TC004 | Search for a product while logged in | To check if the search functionality works as intended for logged-in users. |
| 2. Search Product | 2.1. With account | Search valid product without login account | TS005 | TC005 | Search for a product without being logged in | To determine if non-logged-in users can still search for products. |
| 3. Product Detail | 3.1. Product image and info | Checking the correctness of products | TS006 | TC006 | Validate the product details displayed | To confirm that product details are accurate and match the inventory information. |
| 4. Buy a Product | 4.1. Logging in to the account | — | TS007 | TC007 | Purchase flow with a logged-in user | To ensure the buying process is seamless for users who are logged in. |
| 4. Buy a Product | 4.2. Not Logging in | — | TS008 | TC008 | Purchase flow without logging in | To verify that users can buy products even if they are not logged into the site. |
| 5. Shopping Cart | 5.1. Cart persistence | Check if items remain after exit (not logged in) | TS009 | TC009 | Ensure cart items persist for guests after website exit | To test if the shopping cart retains items for non-logged-in users after exit. |
| 5. Shopping Cart | 5.1. Cart persistence | Check if items remain after exit (logged in) | TS010 | TC010 | Ensure cart items persist for users after website exit | To ensure that the cart's contents are saved for logged-in users upon reentry. |
| 6. Update Quantity | — | Price changes after quantity update | TS011 | TC011 | Verify price update when quantity changes | To check if the total price updates correctly as the item quantity changes. |
| 6. Update Quantity | — | Quantity saved after logout and login | TS012 | TC012 | Confirm quantity persistence through log out/in cycle | To ensure the updated quantity is stored and displayed after user re-logs in. |
| 7. Remove Item | 7a. If 1 item | Cart is "empty" after item removal | TS013 | TC013 | Remove the only item in cart and check if empty | To verify that the cart displays as empty and subtotal is zero after removal. |
| 7. Remove Item | 7b. If more than 1 item | Subtotal update and cart persistence after remove | TS014 | TC014 | Remove items from a cart with multiple items | To confirm the cart updates subtotal correctly and persists after item removal. |
| 8. Checkout | 8a. Logging in to account | — | TS015 | TC015 | Checkout process for a logged-in user | To validate the checkout process for users with an account. |
| 8. Checkout | 8b. Without logging in | — | TS |  |  |  |

1. Test results