**Test Plan for Book Cart Web Application**

**1. Introduction**

Book Cart is a web application that allows users to search for books, add them to a cart, and purchase them. This test plan outlines the testing strategy, tools, environment, scenarios, success criteria, risks, and limitations for ensuring the quality of the Book Cart application.

**1.1 Scope of Testing**

This test plan covers the following functionalities:

* User registration and login
* Book search and filtering
* Adding and removing books from the cart
* Purchase process
* Data validation

The following functionalities are out of scope for this test plan:

* Performance testing under high load
* Security testing

**2. Testing Strategy**

**2.1 Types of Testing**

The following types of testing will be performed:

* **Functional Testing:** Verifying that all core functionalities of the application work as expected.
* **UI Testing:** Checking the user interface for usability and consistency.
* **Regression Testing:** Ensuring that new changes or bug fixes do not introduce new issues or regressions.
* **Negative Testing:** Testing the application with invalid or unexpected inputs to ensure it handles errors gracefully.

**2.2 Testing Tools**

The following tools will be used for testing:

* **Selenium:** For automating UI tests across different browsers.
* **Mocha:** A JavaScript test framework for structuring and running tests.
* **Chai:** An assertion library for writing test expectations.

**2.3 Testing Environment**

The tests will be executed in the following environment:

* **Browsers:** Chrome (latest version)
* **Operating System:** Windows 10
* **Test Data:** Test users and books will be created specifically for testing purposes.

**3. Testing Scenarios**

**3.1 Test Cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Description | Priority | Preconditions | Steps | Expected Result |
| TC\_LOGIN\_001 | User Login - Valid Credentials | High | User account exists | 1. Navigate to the login page.  2. Enter valid email and password.  3. Click the login button. | User is successfully logged in and redirected to the home page. |
| TC\_LOGIN\_002 | User Login - Invalid Password | High | User account exists | 1. Navigate to the login page.  2. Enter valid email and incorrect password.  3. Click the login button. | Error message is displayed, and user is not logged in. |
| TC\_LOGIN\_003 | User Login - Non-existent User | High | N/A | 1. Navigate to the login page.  2. Enter non-existent email and password.  3. Click the login button. | Error message is displayed, and user is not logged in. |
| TC\_REG\_001 | User Registration - Valid Data | High | N/A | 1. Navigate to the registration page.  2. Enter valid user details. 3. Click the register button. | User account is created, and user is logged in. |
| TC\_REG\_002 | User Registration - Existing Email | High | User account with the email exists | 1. Navigate to the registration page.  2. Enter existing email and other details.  3. Click the register button. | Error message is displayed, indicating that the email is already in use. |
| TC\_REG\_003 | User Registration - Invalid Password Format | High | N/A | 1. Navigate to the registration page.  2. Enter details with an invalid password format. 3. Click the register button. | Error message is displayed, indicating password format requirements. |
| TC\_SEARCH\_001 | Book Search - By Title | High | Books exist in the catalog | 1. Navigate to the home page.  2. Enter a book title in the search field.  3. Click the search button. | Books matching the title are displayed. |
| TC\_SEARCH\_002 | Book Search - By Author | High | Books exist in the catalog | 1. Navigate to the home page.  2. Enter an author's name in the search field.  3. Click the search button. | Books by the specified author are displayed. |
| TC\_SEARCH\_003 | Book Search - No Results | High | Books exist in the catalog | 1. Navigate to the home page.  2. Enter a non-existent book title or author.  3. Click the search button. | A message is displayed indicating no books were found. |
| TC\_CART\_001 | Add to Cart - Single Book | High | User is logged in, book exists | 1. Navigate to a book details page.  2. Click the "Add to Cart" button. | The book is added to the user's cart. |
| TC\_CART\_002 | Add to Cart - Multiple Books | High | User is logged in, books exist | 1. Navigate to a book details page.  2. Click the "Add to Cart" button multiple times. | Multiple copies of the book are added to the user's cart. |
| TC\_CART\_003 | Remove from Cart - Single Book | High | User is logged in, cart contains books | 1. Navigate to the cart page.  2. Click the "Remove" button for a book. | The book is removed from the cart. |
| TC\_PURCHASE\_001 | Purchase - Valid Data | High | User is logged in, cart contains books | 1. Navigate to the cart page.  2. Proceed to checkout.  3. Enter valid payment and shipping information. 4. Complete the purchase. | Order is successfully placed, and confirmation is displayed. |
| TC\_PURCHASE\_002 | Purchase - Empty Cart | High | User is logged in, cart is empty | 1. Navigate to the cart page.  2. Proceed to checkout. | Error message is displayed, indicating that the cart is empty. |
| TC\_PURCHASE\_003 | Purchase - Invalid Payment Information | High | User is logged in, cart contains books | 1. Navigate to the cart page.  2. Proceed to checkout.  3. Enter invalid payment information. 4. Complete the purchase. | Error message is displayed, indicating invalid payment information. |

**4. Success Criteria**

* All test cases must pass without any critical or high-severity defects.
* All core functionalities must work correctly on the supported browsers.
* The application must be user-friendly and intuitive.

**5. Risks and Limitations**

* Potential issues with the test environment, such as browser updates or network connectivity problems.
* Limited documentation for the application, which may require additional time for exploration and understanding.
* The test plan may need to be updated as the application evolves and new features are added.

**6. Defect Management**

Any defects found during testing will be reported using a defect tracking tool (e.g., Jira). The defect reports will include detailed steps to reproduce the issue, expected behavior, actual behavior, severity, and priority.

**7. Reporting**

Test results will be documented and summarized in a test report. The report will include the number of test cases executed, the number of passed and failed test cases, a summary of defects found, and any other relevant information.