### Test Plan for Sauce Labs Demo Website

#### ****1. Introduction****

This test plan outlines the strategy and approach for testing the Sauce Labs demo website. It focuses on validating the functionality, usability, and appearance of the website's front end across key workflows. The plan highlights how different types of changes (UI or backend) will influence the testing focus. **This plan excludes security testing, backend-specific testing (beyond appearance validation of backend data), and performance testing.**

#### ****2. Testing Strategy****

##### **2.1. General Testing Approach**

* **Scope**: Testing will be conducted exclusively on **Windows** using **Chrome** and **Opera** browsers.
* **Focus Areas**:
  + Functional workflows across the cart and checkout pages.
  + Visual consistency and appearance of UI elements.
  + Cross-browser checks for UI changes.
  + Data accuracy and appearance validation after backend changes.

##### **2.2. Testing Focus Based on Change Type**

**UI Changes:**

* Ensure UI elements render correctly on both **Chrome** and **Opera**.
* Verify visual alignment, fonts, colors, and responsiveness across:
  + **Cart Page**
  + **Checkout Information Page**
  + **Checkout Overview Page**
  + **Checkout Complete Page**
* Check usability features, including button clicks, navigation links, and form fields.

**Backend Changes:**

* Focus on validating data-dependent elements:
  + Correct product details (name, price, description) displayed on the cart and checkout pages.
  + Proper calculation of totals in the checkout workflow.
  + Consistency of data between steps (cart → checkout overview → checkout complete).
* Ensure error handling and appropriate messaging if backend issues occur

**3. Test Cases and Scenarios**

##### **3.1. Functional Testing**

 **Login Page**:

* Test login functionality with valid credentials.
* Verify error messages for invalid or empty username and password fields.
* Check the UI of the login form (e.g., placeholders, alignment, and responsiveness).

 **Cart Page**:

* Add and remove items from the cart.
* Verify the cart updates with correct quantities and prices.
* Validate navigation to the checkout process.

 **Checkout Information Page**:

* Test form field inputs (first name, last name, postal code).
* Ensure appropriate error messages for missing or invalid inputs.
* Validate navigation to the checkout overview page.

 **Checkout Overview Page**:

* Confirm product details, quantity, and prices match the cart.
* Validate correct calculation of totals (item total, tax, final total).
* Test navigation to the checkout complete page.

 **Checkout Complete Page**:

* Verify the "Thank You" message and order confirmation.
* Validate navigation back to the home page.

 **Left Sidebar**:

* Verify the sidebar opens and closes correctly.
* Check the functionality of links in the sidebar (e.g., logout, navigation links, all idems).
* Test visual consistency and responsiveness.

 **Footer**:

* Verify all footer links (e.g., Twitter, Facebook, LinkedIn) open the correct pages.
* Check the alignment and display of footer elements across supported browsers.
* Test that the footer remains visible and properly aligned on all pages.

##### **3.2. Visual Testing**

* Ensure consistent rendering of elements across **Chrome** and **Opera**:
  + Buttons, forms, and links.
  + Product images, text alignment, and colors.
  + Responsive layout for different window sizes.
* Verify visual integrity after backend changes, focusing on data-dependent elements (e.g., product information).

#### ****4. Test Environment****

* **Platform**: Windows
* **Browsers**: Chrome (latest version), Opera (latest version)
* **Tools**: TestRail for test case management and reporting.

#### ****5. Deliverables****

* **Test Cases**: Documented in TestRail for all scenarios.
* **Test Execution Report**: Including pass/fail results, screenshots for defects, and testing logs.
* **Defect Log**: Reported in TestRail or a defect management tool.

#### ****6. Pass/Fail Criteria****

* **Pass**: All UI elements render correctly, and workflows function as intended without data or visual inconsistencies.
* **Fail**: Any defect in functionality, data accuracy, or UI appearance.

#### ****7. Exclusions****

This test plan does **not include**:

* **Security Testing**: Testing for vulnerabilities such as SQL injection, XSS, CSRF, etc.
* **Backend Testing**: Detailed validation of APIs or database operations beyond appearance and data verification in the UI.
* **Performance Testing**: Load, stress, or scalability testing for backend or frontend.

#### ****8. Risks and Mitigation****

* **Risk**: UI inconsistencies between browsers.
  + **Mitigation**: Conduct cross-browser testing for every UI change.
* **Risk**: Backend data inaccuracies impacting the UI.
  + **Mitigation**: Validate all data-driven elements thoroughly after backend changes.