Requirements Gathering (SauceDemo Project)

◆ 1. Functional Requirements

These define what the application (SauceDemo) *must do*. Based on your manual test cases, the following key functionalities are required:

1.1 Login Page

- The system must allow users to log in with valid credentials.

- The system must display an error message when incorrect credentials are entered.

- The "Username" and "Password" fields must be mandatory.

- The "Login" button should only be clickable when both fields are filled.

- The user must be redirected to the product inventory page after successful login.

- Locked users must be restricted from accessing the system and shown a proper error message

---

1.2 Product Listing Page

- The system must display a list of all available products with name, description, and price.

- Each product must have an "Add to Cart" button.

- The "Remove" button should appear after adding a product to the cart.

- The "Cart Icon" should display the total number of items added.

- The sorting dropdown must allow sorting by **Name (A–Z / Z–A)** and **Price (Low–High / High–Low)**.

- Each product's image and details must be clickable and lead to its product detail page.

1.3 Shopping Cart

- The cart page must list all products added by the user.

- Users must be able to remove items from the cart.

- The "Continue Shopping" button should return the user to the product list.

- The "Checkout" button must navigate to the checkout page.

1.4 Checkout Process

- The checkout page must include fields for **First Name**, **Last Name**, and **Postal Code** (all required).

- The "Continue" button should proceed only when all fields are filled.

- The system must display a summary of order details before final confirmation.

- The "Finish" button should confirm the purchase and display a "Thank You" message.

- The "Back Home" button should redirect to the product inventory page.

1.5 Logout

- The user must be able to log out using the menu option.

- After logout, the system should return to the login page and clear session data.

◆ 2. Non-Functional Requirements

These describe how the system should perform rather than what it should do.

Category	Requirement
Performance	Pages should load within 3 seconds on standard internet speed.

Security	Login credentials must be validated securely; user sessions should expire after inactivity.
Usability	All buttons and forms must be intuitive and clearly labeled.
Compatibility	The application should work on all major browsers (Chrome, Edge, Firefox).
Reliability	The system should maintain data consistency across pages (e.g., items remain in cart).
Accessibility	UI elements (buttons, links, input fields) should be visible and accessible to all users.

◆ 3. User Stories & Use Cases

User Story 1 – Login

As a **standard user**, I want to log into the application using valid credentials so that I can access the product inventory page.

Use Case Steps:

1. User navigates to the login page.

2. User enters valid username and password.

3. User clicks on the login button.

4. System verifies credentials and redirects to inventory page.

Alternate Flow:
If credentials are invalid → show error message: *"Username and password do not match."*

User Story 2 – Add Product to Cart

As a **logged-in user**, I want to add products to my cart so that I can purchase them later.

Use Case Steps:

1. User views the product list.

2. User clicks "Add to Cart" on a product.

3. System increases the cart count and displays the item in the cart.

Alternate Flow:
If product already added → "Remove" button replaces "Add to Cart".

User Story 3 – Checkout

As a **user with items in the cart**, I want to enter my personal details and complete checkout so that I can finalize my order.

Use Case Steps:

1. User navigates to the cart and clicks "Checkout".

2. User fills in required fields.

3. User reviews order summary.

4. User clicks "Finish".

5. System confirms order and displays thank-you message.