| **Swag Labs Demo Website - Manual testing**  <https://www.saucedemo.com/> |
| --- |

## **Overview**

## Swag labs demo website is created by SauceLabs for aspiring testers to practice manual and automation testing. In this project, we will only cover manual testing of the application.

This document contains user stories, description of functional requirements, acceptance criteria and mockup of the UI design that will serve as a guide for the testers.

## **Objective**

## The objective of this project is to learn how to execute a manual testing procedure, create functional requirement specification, create test cases and how to do bug reporting.

The project will mainly be focused on testing an e-commerce demo website made by saucelabs. The demo website has provided different scenarios for each user credentials that are given.

Some of the bugs are visual, and some are functional.

## **Requirements**

Swag labs only need simple requirements for it to operate properly. First, a stable internet connection and last, any of these modern browsers on the market.

## **Limitations**

* User credential “**locked\_out\_user**”will not be tested, since it is not able to access the application using this credential.
* This project will only focus on the core functionalities of the application such as:
  + Log-in and Log-out
  + Adding / Removing items to cart
  + View product details
  + View shopping cart
  + Checkout process

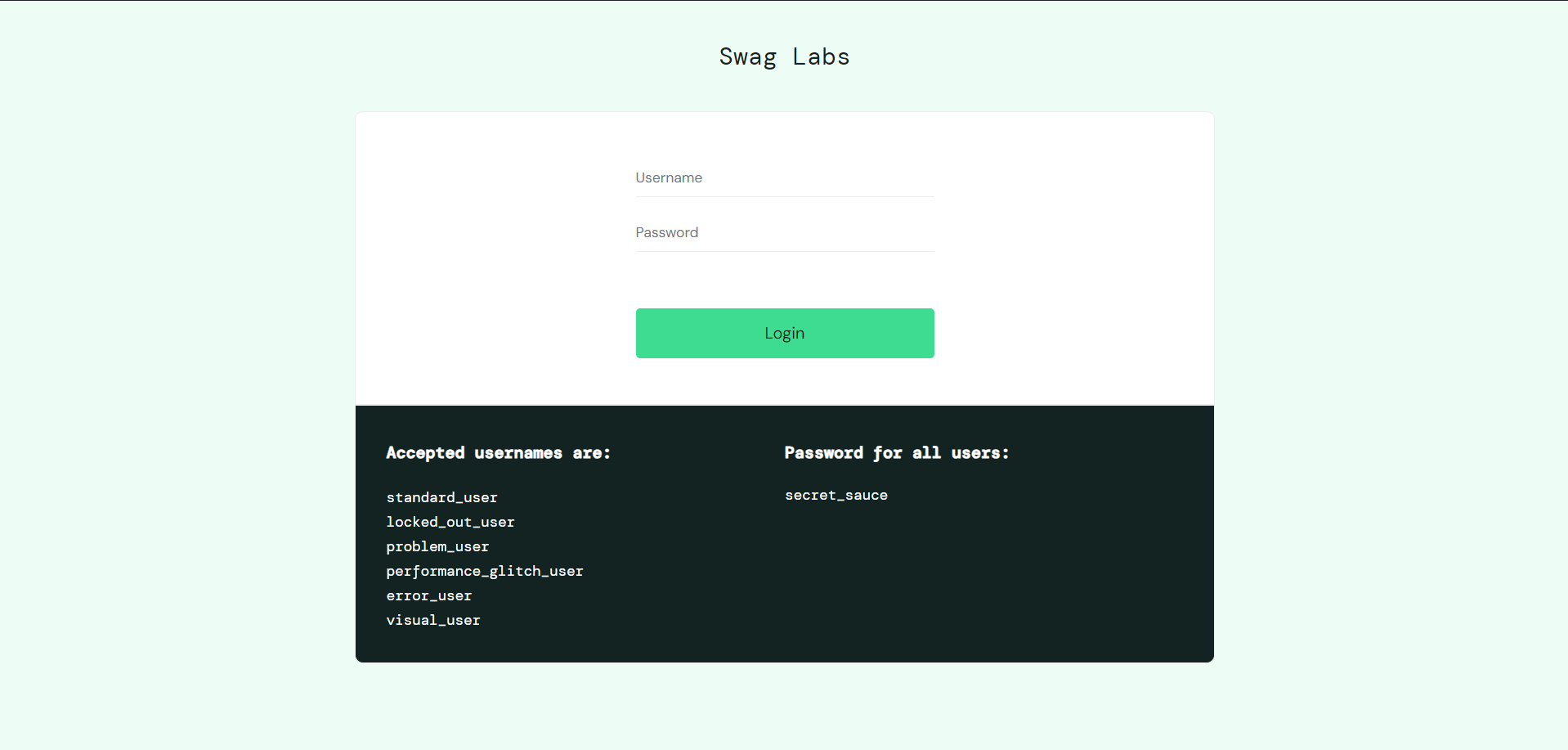
## **User Interface Guidelines**

### **1. Landing page / Log-in page**

The landing page contains a card centered in the screen that contains the username and password fields.

Below the log-in section, credentials are provided by SauceLabs to access their demo site.

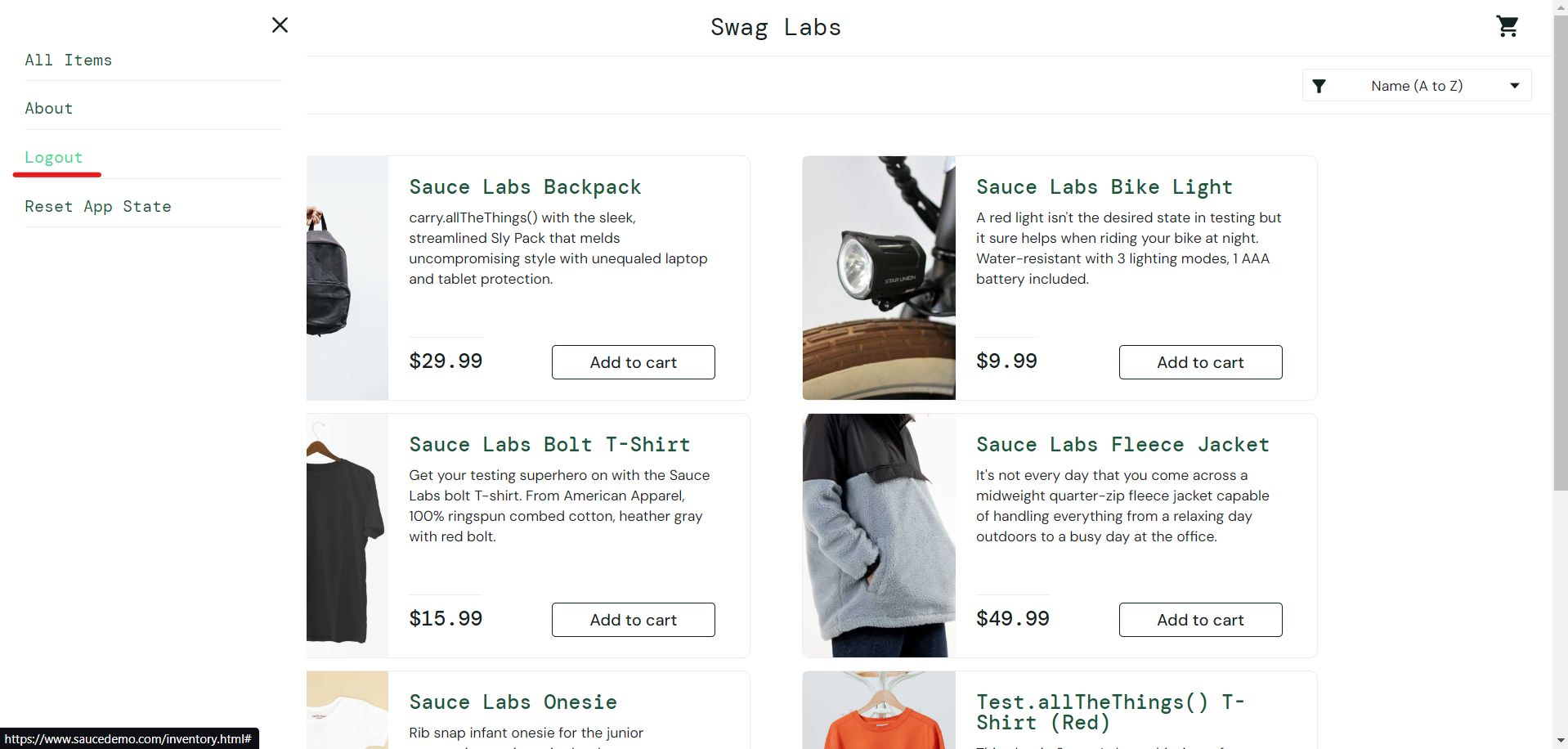
**Landing page / Log-in page UI mockup**



### **2. Log-out**

The logout function can be located in the upper left section of the navigation bar under the hamburger menu.

**Log-out UI Mockup**

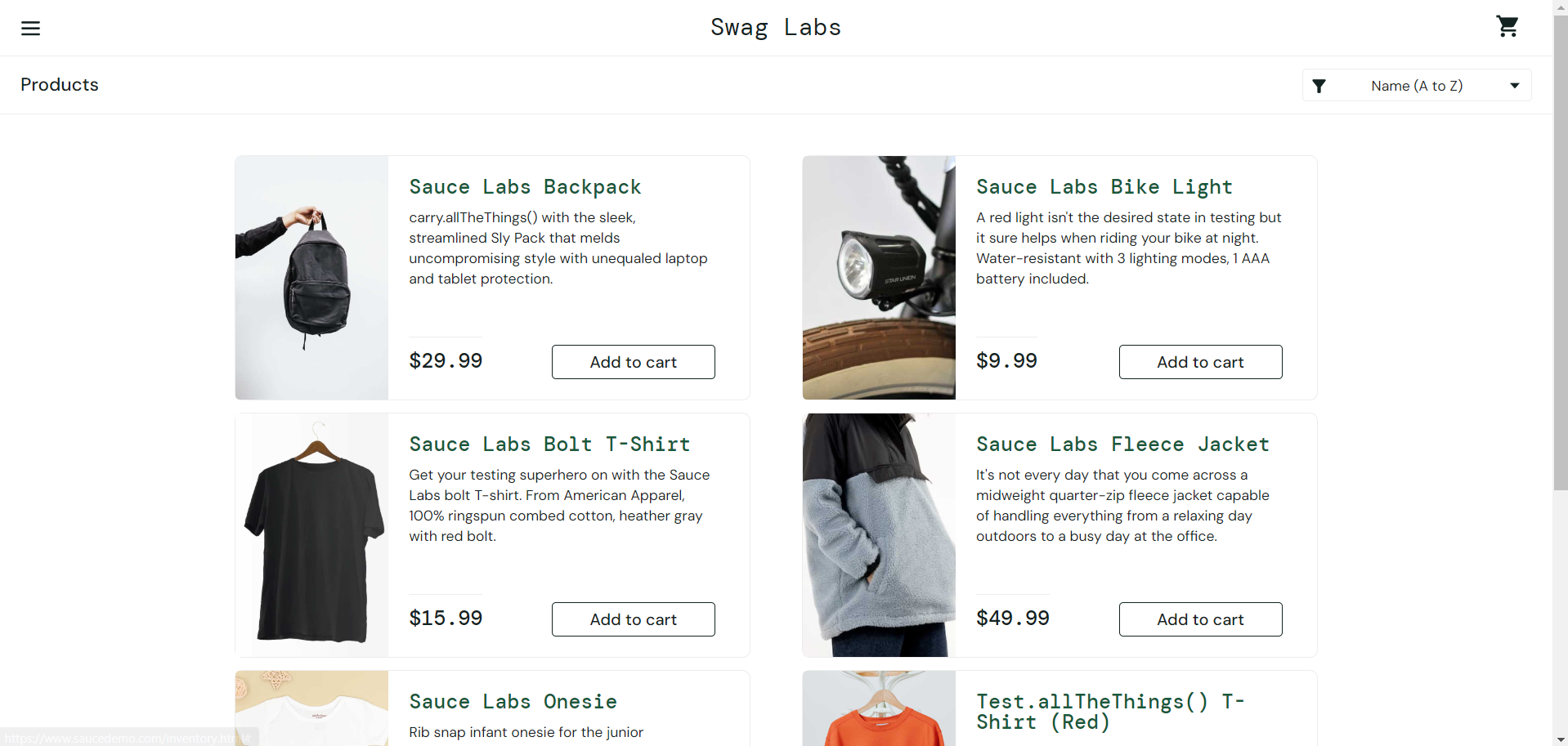


### **3. Homepage / Product listing page**

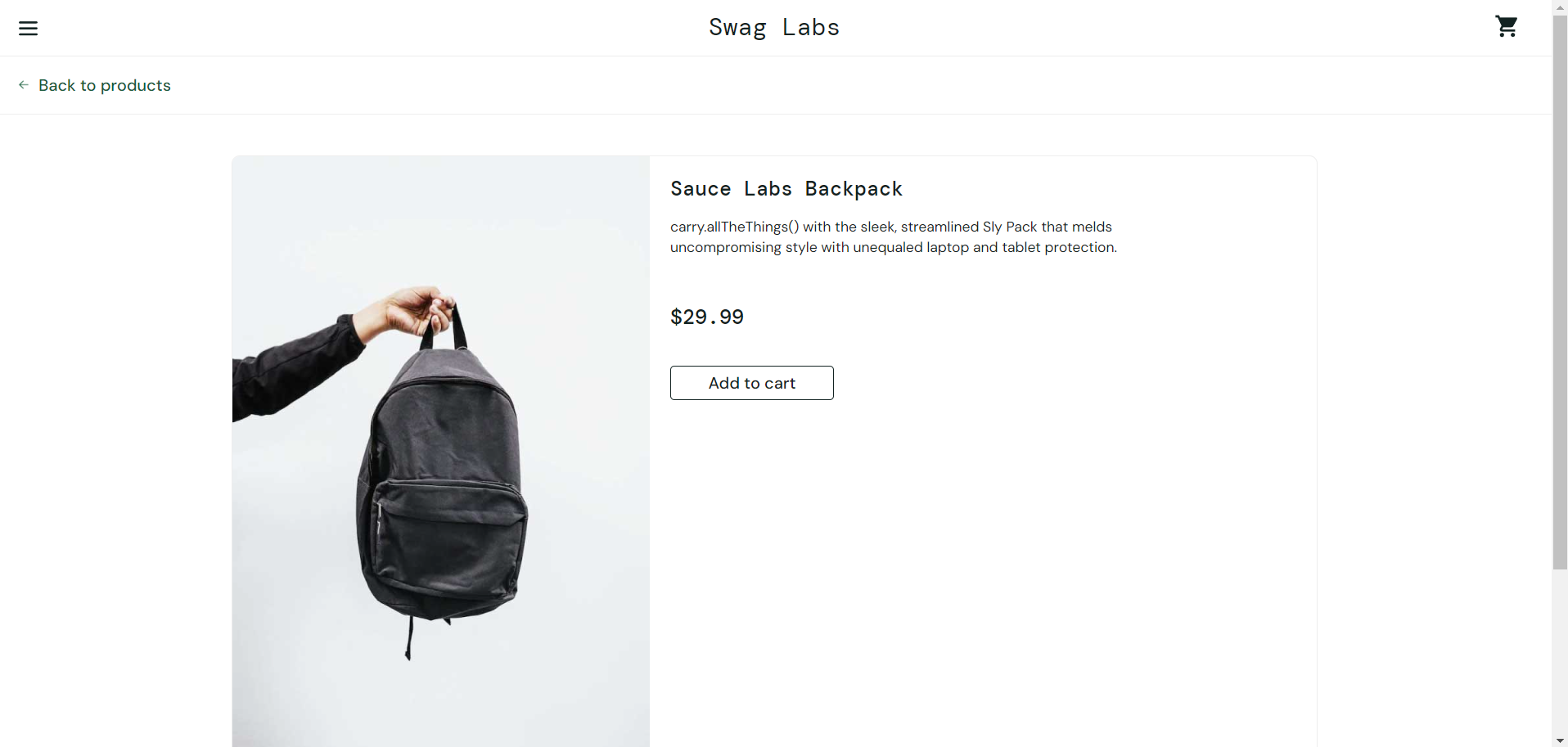
The homepage / product listing page consists of multiple cards that contain the product image, product name, description and price of an item. Each product is individually contained in a product card and the card itself also contains the **Add to cart** button.

When the product image or product name is clicked, the user will be directed to the inventory-item page where the full size of the product image and details are presented to the user.

**Product listing page UI mockup**



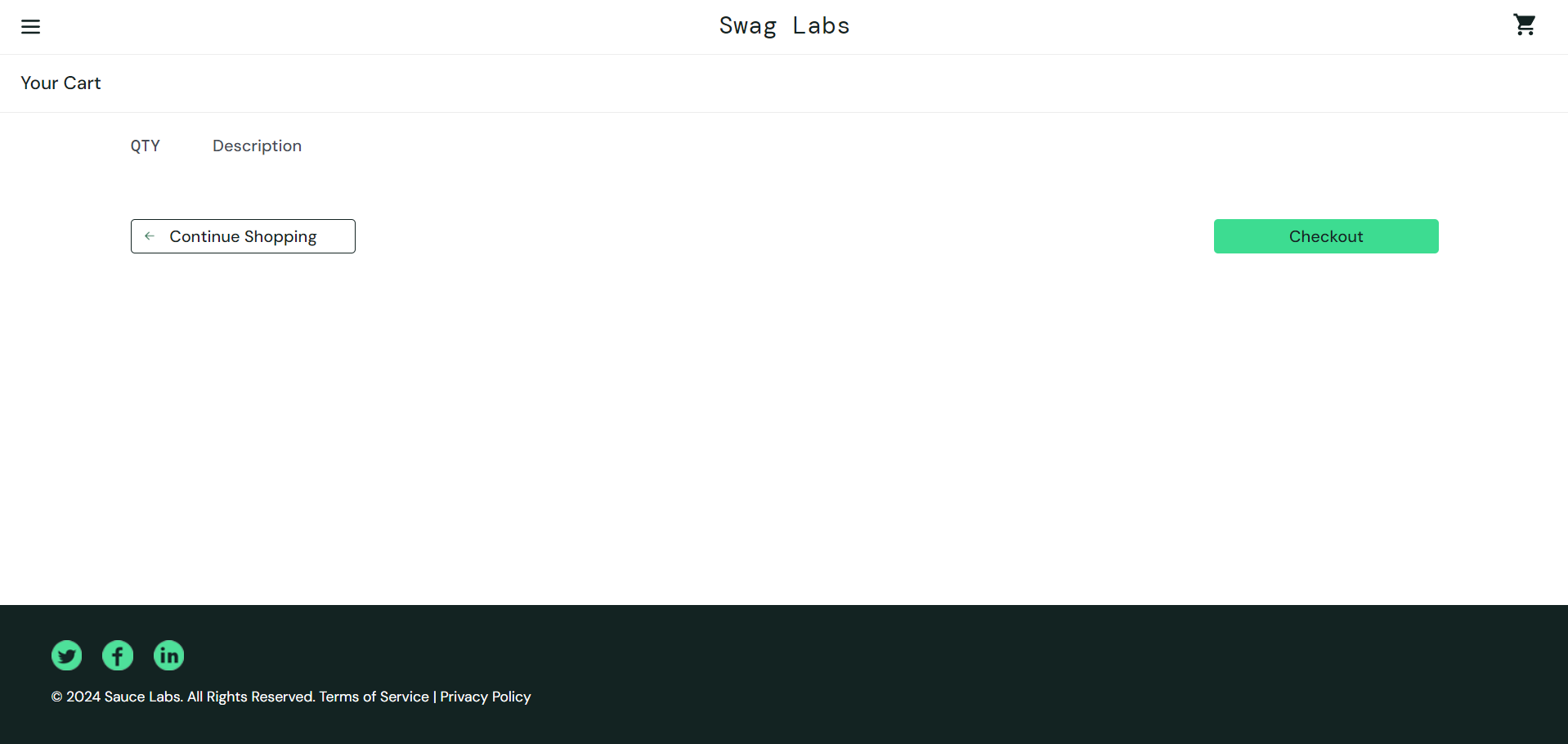
**Product details full view**



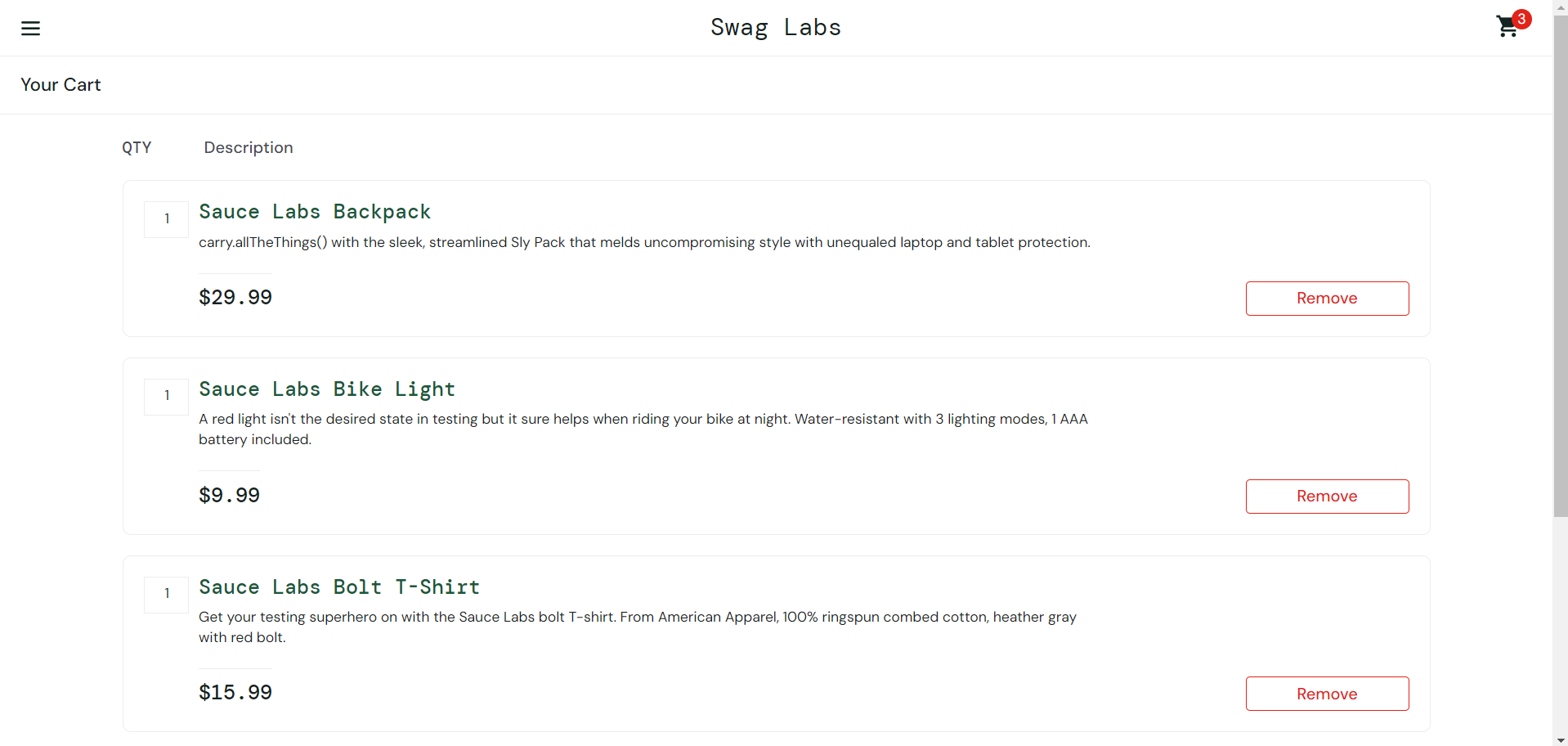
### **4. Shopping Cart**

The shopping cart section has two states: **empty** and **non-empty**. The items added by the user will be located in this section and this is also where the checkout process will be executed.

**Shopping Cart (Empty) UI mockup**



**Shopping Cart (Non-empty) UI mockup**

****

## **User Stories**

| **Function** | **User story** |
| --- | --- |
| Log-in | As an existing user, I want to be able to easily access my account by entering my login credentials. |
| Log-out | As a user, I want to be able to easily log-out of my account when I am finished with my transaction. |
| View Product Details | 1. As a user, I want to be able to view the product details by clicking on the image of the item or by clicking on its name or title. 2. As a user, I want to be able to see the actual image of the product based on its name or description. 3. As a user, I am expecting that the product image is relevant to its product name and product details. |
| Sort Items | 1. As a user, I want to be able to sort the items by name(A-Z, Z-A). 2. As a user, I want to be able to sort the items by price(lowest-highest, highest-lowest). |
| Add item to cart | As a user, I want to be able to easily add an item to my cart by clicking the “**Add to cart**” button. |
| Remove item from cart | As a user, I want to be able to easily remove an item in my cart by simply clicking the “**Remove**” button from the product list view or in the view shopping cart menu. |
| View Shopping cart | As a user, I want to be able to see my shopping cart whether it is empty or not. |
| Checkout | 1. As a user, I want to be able to check out the items I added in my shopping cart. 2. As a user, I want to be able to see the payment details of my purchase before proceeding. |

## 

## **Requirements Table (Functional)**

| **Requirement** | **Description** | **Priority** | **Acceptance Criteria** |
| --- | --- | --- | --- |
| Log-in | 1. Users should be able to log-in using provided credentials. | High | 1. Users can log-in from the homepage. 2. Users can see the product listing page after sign-in. |
| Log-out | 1. Users should be able log-out from the application. | High | 1. Users can log-out from the menu section. |
| View product details | 1. Users should be able to view the product details. | Medium | 1. Users can view the product details by clicking the product image or the product name. |
| Sort items | 1. Users should be able to sort the items by name and by price. | Low | 1. Users can sort the items by pressing the sort button. |
| Add item to cart | 1. Users should be able to add the item to their cart. | High | 1. Users can add the item to their cart by clicking the “**Add to cart**” button. |
| Remove item from cart | 1. Users should be able to remove an item in their cart. | High | 1. Users can remove an item by either:    1. Clicking the “**Remove**” button inside the card.    2. By going to their shopping cart and by clicking the “**Remove**” button |
| View Shopping cart | 1. Users should be able to view their shopping cart. | High | 1. Users can view their shopping cart by clicking the shopping cart icon. |
| Checkout | 1. Users should be able to check out the items in their shopping cart. | High | 1. Users can fill out the required details in each text field. 2. Users can complete the checkout process by clicking the “**Finish**” button |

## **Legend**

### **Test cases**

Test cases will have a status column where each test case will be identified as **PASS** or **FAIL**.

If the main function of the test case **PASSED** but has some additional issues such as performance and UI.

It will be considered as **PASSED** but it should be declared in the **Remarks** column of the test case document.

In the **Remarks** column, the tester will specify the details of the additional issues that need to be addressed such as problems in UI, responsiveness and performance.

| **Status** | **Criteria** |
| --- | --- |
| **PASS** | 1. Main function must execute properly 2. Main function executed properly but has UI issues: misplaced icon, unresponsive button etc. 3. Main function executed properly but has performance issues: slow response time. |
| **FAIL** | 1. Main function failed to execute properly 2. Main function executed properly but does not meet the actual result criteria. |