# User Authentication: Login with valid/invalid credentials

## User Authentication

**+ Prerequisites:**

- The E-Commerce Mobile Application is downloaded.
- The internet is enabled.

---

**+ Acceptance Criteria:**

- As a client user, I will open the app and attempt to log in by entering my registered credentials.
- If the credentials are valid, the user will be authenticated and redirected to the home screen.
- If the credentials are invalid:
    - An appropriate error message will be displayed.
    - The user will be prompted to try again.
- The system must validate the entered username and password format before submission.

---

**+ Expected Results:**

- Valid users are successfully logged in and redirected to the home screen.
- Invalid login attempts display relevant error messages and keep the user on the login screen.

# **Product Listing:** Displaying products, sorting options

## Prerequisites :

- The website is accessible and working.
- The user is either logged in or allowed to browse as a guest.
- Internet connection is enabled and stable.
- Products are activated and available in the backend database.

## Product listing display :

As a client user, I can navigate to the products screen from the main menu or home screen.

The product listing screen should display products in a grid or list view, showing the following details per item:

- Product image
- Product name
- Short description (optional)
- Price
- Availability status (In Stock / Out of Stock)
- Action button (e.g., “View Details” or “Add to Cart”)

## Sorting options:

As a client user, I can sort the displayed products based on:

- Price (Low to High / High to Low)
- Product Name (A–Z / Z–A)
- Newly arrived products
- Top Rated / Best Sellers product

## Navigation and browsing :

- Users should be able to navigate through products using:
  - Pagination
  - “Load More” button (if pagination is not used)
- The product screen must be fully responsive across devices (mobile, tablet, desktop).

Expected results :

- Products should be displayed clearly and consistently with updated data.
- Sorting actions should update the view immediately (preferably without reloading the page).
- Filtering should narrow down product results based on user selection.
- Tapping or clicking a product redirect to its detailed view screen.
- Performance should remain smooth even with a large number of items.
- If no results are found, a proper message (e.g., “No matching products found”) should be shown.

# Shopping Cart User Story

Prerequisites:
1- Desktop or Mobile Device
2-The internet is enabled
3-Open the web site by browser
4-Valid account for login
5-The user interface (UI) for product listings, product detail pages, and cart must be fully implemented and functional.
6-A backend product catalog should be available and synchronized with real-time: Product names /Prices/Images/Inventory/stock availability.

## Shopping Cart – Add, Remove, and Update Items

### 1. Add Items to Cart:

As a client user should be able to add products to the shopping cart from:
The product listing page or the product detail page and when the user add an item:
The cart icon in the header should update in real time to reflect the new total number of items in the cart. If the user add the same item (same variant) again the system should increase the quantity in the cart rather than adding a new row. A tooltip or message should inform that the item is already in the cart and the quantity has been updated.

#### System validations:

If the item is out of stock, an error message should appear, and the Add to Cart button should be disabled.

### 2. Remove Items from Cart:

As a client user, I should be able to:
Remove individual items directly from the cart page using a clearly labeled "Remove"button.
See the cart content update instantly, reflecting the decreased total item count and updated subtotal and also Updated total price
Additional system behaviors:
If I remove the last remaining item, the cart should show a message such as: "Your cart is currently empty."

## 3. Update Quantity in Cart:

As a client user, I should be able to: Change the quantity of any item directly in the cart using a text input field or increment/decrement buttons ("+ / –") on updating the quantity .The line item total (price × quantity) should update automatically.and the cart subtotal and total price should refresh in real time.

### Validation & behavior rules:

The quantity field should only accept numeric values having a minimum value of 1

### Additional notes:

System should save updated quantities if the user navigates away or refreshes the page.
For mobile users, quantity controls must be tap-friendly and responsive.

### Validation:

Quantity must be a positive number greater than zero.
If set to zero, item should be removed automatically.
Quantity updates must not exceed the available stock, as defined in the product inventory settings.

## Checkout & Payment.

+ Prerequisites:

- The application or website is accessible and working.
- The user is either logged in or allowed to proceed as a guest.
- The user has added at least one product to the shopping cart.
- Internet connection is enabled and stable.

——————————-

+ Acceptance Criteria:

1. Checkout Flow:

- As a user, I can access the cart and click on the “Proceed to Checkout” button.
- The checkout screen should display:
- Order Summary
- Shipping Information
- Payment Methods
- I should be able to:
- Enter or select a valid shipping address.
- Choose one of the available payment options (e.g. Credit Card, PayPal, Cash on Delivery).
- When I click “Place Order”:
- The system should validate all required fields.
- If payment is online, it should securely process the transaction.
- If successful, a confirmation message should appear on-screen.
- A confirmation email or in-app notification should also be sent to the user.

2. Order Summary Validation:

- The order summary should clearly show:
- List of items with name, quantity, and unit price.
- Subtotal (before tax and shipping).
- Shipping fee (if applicable).
- Taxes (based on selected country/region).
- Discounts or promo codes (if applied).
- Final total.
- If a user applies a valid promo code, the discount should reflect instantly in the summary.
- If the promo code is invalid or expired, the user should see an error message.
- Users should be able to edit quantities or remove items directly from the summary before confirming the order.

---

+ Expected Results:

- The order should be successfully submitted when all data is valid.
- The calculated totals (subtotal, tax, shipping, discount, final total) must be accurate and clearly displayed.
- A successful order submission should trigger:
- Order record saved in the database.
- Payment confirmation (for online payments).
- Confirmation message and email/notification sent to the user.
- In case of payment failure, the user should be informed with a proper message and given the option to try again or change the payment method.

---

+ Dependencies:

- Payment Gateways (e.g., PayPal, Stripe, etc.) must be properly integrated and configured.
- Shipping API or service should return available methods and fees correctly.
- Promo Code system should be managed via the admin dashboard and linked to the checkout.
- The application backend should handle order storage and trigger notifications.
- Product and cart systems should stay in sync throughout the checkout process.

## Menu Options

**+ Prerequisites:**

- The user is logged in to the application.
- The internet is enabled.

---

**+ Acceptance Criteria:**

- As a client user, I will click the menu icon to access:
    - All Items: Redirects to product listing.
    - About: Displays app version, company info, and contact.
    - Logout: Signs out the user and returns to login screen.
    - Reset App State: Clears local storage and app session data.
- All options must be clearly labeled and responsive to tap.
- Confirmation prompt is displayed before logging out or resetting the app.

---

**+ Expected Results:**

- All Items redirects to the main product listing.
- About screen loads with correct static information.
- Logout clears session and returns user to login screen.
- Reset App State fully clears app data and restarts the app session.

---