**Name: Piyush Kasture**

**Module Name: Latest Industry Trends**

**PRN: 2509401147003**

## Assignment No. 5

1) Requirement Document 3: Online Shopping Cart (E-commerce)

Module: Shopping Cart

Description:

Users can add items to the cart and place an order.

Functional Requirements

User should be able to add products to the cart.

User should be able to remove products from the cart.

Cart should automatically recalculate total price when quantity changes.

Discount coupons should be applied before checkout.

If the cart is empty, checkout button should be disabled.

User must be logged in to proceed to checkout.

Business Rules

Maximum 10 units allowed per product.

Only one coupon can be applied per order.

**Prompt:**

You are a QA Test Case Generator with expertise in writing clear, structured, beginner-friendly test cases.

Intent:
Generate a complete set of positive, negative, validation, and boundary test cases for the specified module, and then validate functional coverage against the requirement document.

Context:
Use **"Requirement Document 3: Online Shopping Cart (E-commerce)"** as the single source of truth. The module includes:

Adding/removing products to/from cart

Updating quantities with auto-recalculation

Applying a single discount coupon before checkout

Checkout disabled when cart empty

Login required to proceed to checkout

Business rules: max 10 units per product, only one coupon per order

Data model: Cart and CartItem fields as defined in the requirement document

Output Format & Constraints:

1. Produce test cases in a table with these columns only:

   Test Case ID

   Test Scenario

   Test Steps (clear, numbered)

   Test Data (specific sample values where applicable)

   Expected Result (explicit messages or system state)

2. Generate 12–15 distinct test cases covering:

   Add item (happy path and adding existing item increments quantity)

   Add item beyond max ( >10 ) and beyond stock

   Remove item (decrease quantity and remove full line)

   Update quantity (valid, zero, negative, non-numeric, boundary 1 and 10)

   Auto-recalculation of totals (subtotal, discount, total)

   Apply coupon (valid, invalid/expired, second coupon attempt)

   Checkout behavior (disabled when empty, redirect to login for guest, cart preserved/merged on login)

   Persistence for authenticated users

   Edge cases (item out of stock after added, price change)

3. After the test-case table, validate functionality & requirement coverage:

   Provide a short mapping table/list that maps each requirement (FR-001…FR-006 and BR-001…BR-002) to one or more Test Case IDs that cover it.

   State any requirement not fully covered by the generated test cases and suggest 1–2 additional tests to fill the gap (if needed).

**Output:**

| Test Case ID | Test Scenario | Test Steps | Test Data | Expected Result |
|---|---|---|---|---|
| TC-CART-001 | Add single product to cart (happy path) | 1. Open product page for product_id=P1001 (price $20, stock 5). 2. Click Add to Cart. 3. Open Cart. | product_id: P1001quantity: 1 | Item P1001 appears in cart with quantity 1; subtotal = $20; toast: "Item added to cart" (or visible confirmation). Checkout enabled only if cart not empty. |
| TC-CART-002 | Add same product again increments quantity (within limit) | 1. Cart already has P1001 qty=1. 2. On product page click Add to Cart again. 3. Open Cart. | product_id: P1001initial qty: 1 → add 1 | Cart shows P1001 qty = 2; line_total = $40; subtotal updated accordingly. |
| TC-CART-003 | Add product exceeding maximum allowed per product (>10) | 1. On product page attempt to add quantity = 11 (or add until >10). 2. Submit add. | product_id: P2002attempt qty: 11 | Action blocked; inline error displayed: "Maximum 10 units allowed per product"; cart unchanged. |
| TC-CART-004 | Add product beyond available stock | 1. Product P3003 has stock = 3. 2. Try to add quantity = 4. 3. Submit. | product_id: P3003stock: 3attempt qty: 4 | Server/client validation fails; error message: "Only 3 units available" (or equivalent); item not added or quantity set to max available. |
| TC-CART-005 | Remove a single unit (decrease quantity) | 1. Cart has P1001 qty = 3. 2. Click "-" (decrease) once. 3. Observe cart. | product_id P1001initial qty: 3 | Quantity becomes 2; line_total and subtotal update accordingly. |
| TC-CART-006 | Remove entire line item | 1. Cart has P1001. 2. Click Remove for that | product_id P1001 | Item removed from cart; subtotal updated; if cart |

| Test Case ID | Test Scenario | Test Steps | Test Data | Expected Result |
|---|---|---|---|---|
| | (delete) with confirmation | line. 3. Confirm removal in modal. | | becomes empty, checkout button disabled and tooltip: "Add items to your cart to proceed". |
| TC-CART-007 | Update quantity to boundary values (1 and 10) | 1. Set qty to 1 → save. 2. Set qty to 10 → save. 3. Observe totals. | product_id P4004unit_price: $5 | qty=1 → line_total $5; qty=10 → line_total $50; subtotal updates accordingly. No errors. |
| TC-CART-008 | Update quantity to invalid values (0, negative, non-numeric) | 1. Edit quantity field to 0 → submit. 2. Edit to -1 → submit. 3. Edit to abc → submit. | product_id P4004 | For 0 or -1: show validation error "Quantity must be between 1 and 10" and revert. For abc: show "Enter a valid number" and revert. |
| TC-CART-009 | Automatic recalculation of totals on quantity change (debounced) | 1. Change qty from 1 → 3 in cart (type quickly). 2. Wait ~300ms. 3. Verify totals. | product_id P5005unit_price: $10 | After debounce, line_total = $30; subtotal and displayed total update. UI remains responsive; no duplicate API calls (debounce implied). |
| TC-CART-010 | Apply valid coupon (single coupon) | 1. Cart subtotal = $100. 2. Enter coupon code SAVE10 (valid: 10% off). 3. Click Apply. | coupon: SAVE10 (valid)subtotal: $100 | Coupon accepted: discount line shows $10; total = $90; success message "Coupon applied successfully." |
| TC-CART-011 | Apply invalid/expired coupon | 1. Enter coupon OLD50 (expired). 2. Click Apply. | coupon: OLD50 (expired) | Coupon rejected; error: "Invalid or expired coupon"; no discount applied. |

| Test Case ID | Test Scenario | Test Steps | Test Data | Expected Result |
|---|---|---|---|---|
| TC-CART-012 | Attempt to apply a second coupon (business rule) | 1. Apply valid coupon SAVE10. 2. Attempt to apply another coupon WELCOME5. | coupons: SAVE10, WELCOME5 | Second application blocked; error: "Only one coupon can be applied per order"; original coupon remains. |
| TC-CART-013 | Checkout disabled when cart empty (UI state) | 1. Ensure cart has 0 items. 2. Observe Checkout button. | cart: empty | Checkout button is disabled/greyed; tooltip on hover: "Add items to your cart to proceed". |
| TC-CART-014 | Guest clicks Checkout → redirect to Login & cart preserved (merge on login) | 1. As guest, add items (e.g., P1001 qty=2). 2. Click Checkout. 3. On Login page, complete login with user userA. 4. After login, navigate to Checkout. | guest cart: P1001 qty=2user creds: userA@example.com / pwd | After login, user's cart contains the previously added items (merged if user had existing cart). Checkout page shows preserved items and totals. |
| TC-CART-015 | Item goes out of stock after being added to cart (edge case) | 1. Add P6006 to cart (stock 2). 2. Another process reduces stock to 0. 3. User opens cart or proceeds to checkout. | product_id P6006initial stock:2 → 0 | System shows: "Item no longer available; removed from cart" (or similar). Item removed and totals updated; user notified. |