Bug Report 1 – Discount Value Applied Without Backend Validation

Description

On the Cart page, a discount value of Ƀ250 is displayed and applied to the order total even though **no discount data is retrieved from the backend or database**.
Currently, the discount is **not validated, calculated, or controlled by backend business logic** and is directly reflected on the frontend.

This causes the Cart to show a reduced total price **without any Sales Manager–defined discount**, database record, or active discount rule.

Steps to Reproduce

1. Log in as a customer.
2. Add any product to the shopping cart (e.g., *Stone Rib Table*).
3. Navigate to the **Cart** page.
4. Observe the **Order Summary** section.

Expected Result

- Discounts should be displayed and applied **only if**:
  - They are defined by a **Sales Manager**
  - They are stored in the **database**
  - They are validated and calculated by the **backend**
- The frontend should display **only the discount value returned by the backend**.
- If no valid discount exists, the discount field should be hidden or shown as Ƀ0.

Actual Result

- A discount of Ƀ250 is displayed and applied automatically.
- The discount is not:
  - Retrieved from the database
  - Validated by the backend
  - Controlled by Sales Manager logic
- The order total is reduced incorrectly.

Impact

- Incorrect price calculation in the Cart
- Violates backend–frontend separation of concerns
- Creates inconsistency with Sales Manager discount responsibilities
- May confuse users and evaluators during demo

---

Severity

🟡 **Medium**

Proposed Fix

- Remove any hardcoded or frontend-based discount values.
- Connect the Cart discount section to **backend APIs** that:
    - Retrieve discount data from the database
    - Validate discount activity and eligibility
    - Calculate the correct discount amount
- Ensure the frontend:
    - Displays discounts **only when provided by the backend**
    - Reflects backend-calculated values in the Order Summary
- If no discount is returned by the backend, the discount should not be applied.

Attachment / Evidence
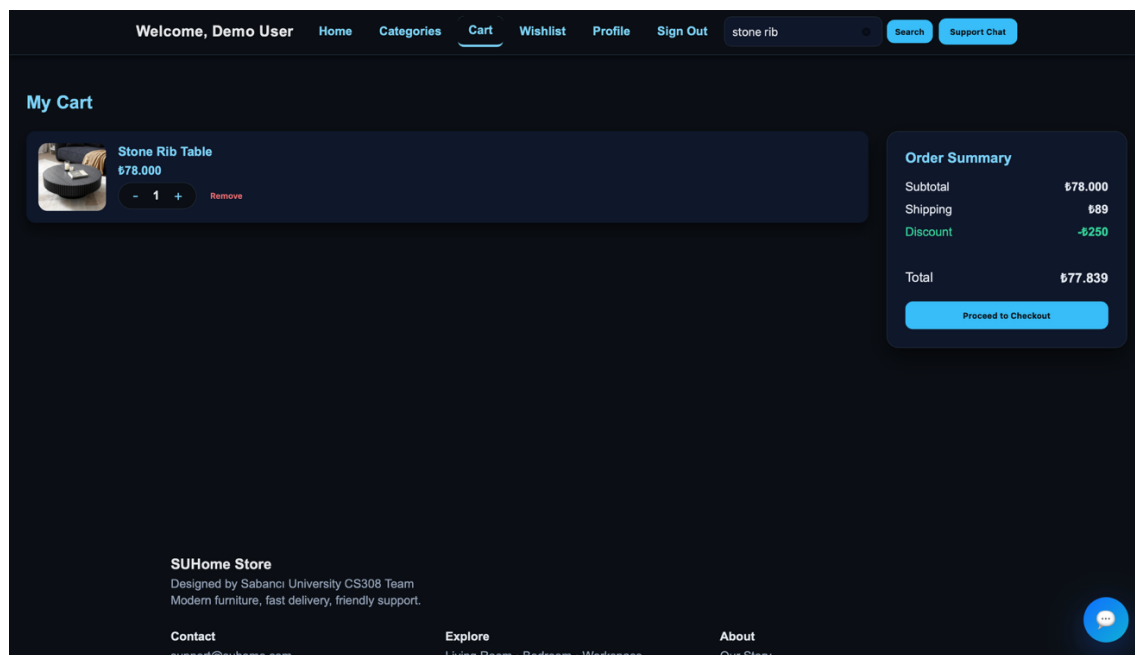
Bug Report 2 – Discount Not Reflected in Wishlist

Description

When a **Sales Manager applies a discount** to a product, the discounted price is correctly displayed on the **Categories** page.
However, the **same product does not display the discounted price in the Wishlist**, causing inconsistency between product views.

The Wishlist continues to show the **original (non-discounted) price**, even though the discount is active and visible elsewhere in the system.

Steps to Reproduce

1. Log in as a Sales Manager.
2. Apply a discount to a product (e.g., *Calm Studio Bed*).
3. Log in as a customer.
4. Navigate to the **Categories** page and observe the discounted price.
5. Add the same product to the **Wishlist** (or navigate to Wishlist if already added).
6. Observe the product price shown in the Wishlist.

Expected Result

- Discounted products should display **consistent pricing across all pages**, including:
  - Categories
  - Wishlist
  - Cart
- The Wishlist should reflect:
  - Discounted price
  - Discount rate (if applicable)
- Pricing data should be **retrieved from the backend** and shared consistently across all views.

Actual Result

- Discounted price is shown correctly in the **Categories** page.
- The **Wishlist displays the original price**, ignoring the active discount.
- No discount indication is shown in the Wishlist.

Impact

- Inconsistent user experience across pages
- Users may be misled about the actual product price
- Breaks pricing integrity of the system
- Indicates incomplete backend–frontend data synchronization

Severity: 🟡 Medium

Proposed Fix

- Ensure the Wishlist retrieves **up-to-date product pricing data from the backend**, including:
  - Active discounts
  - Discounted prices
- Unify product price calculation logic so that:
  - Categories, Wishlist, and Cart use the **same backend response**
- Avoid caching or storing outdated price values in the Wishlist.
- If a discount is active, the Wishlist should reflect it exactly as shown on the Categories page.

Attachment / Evidence
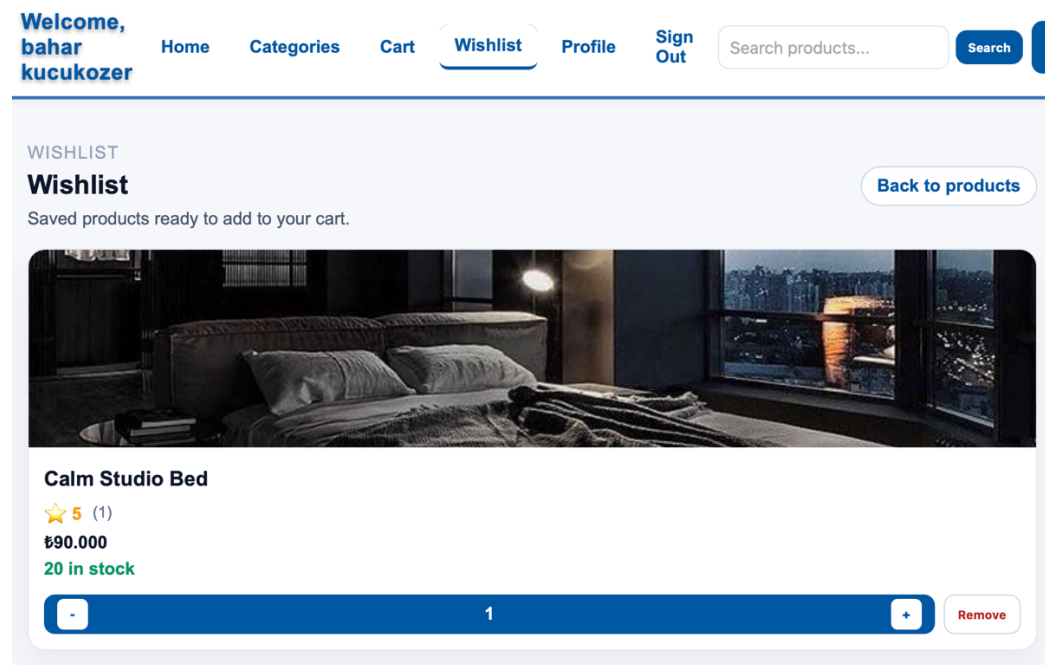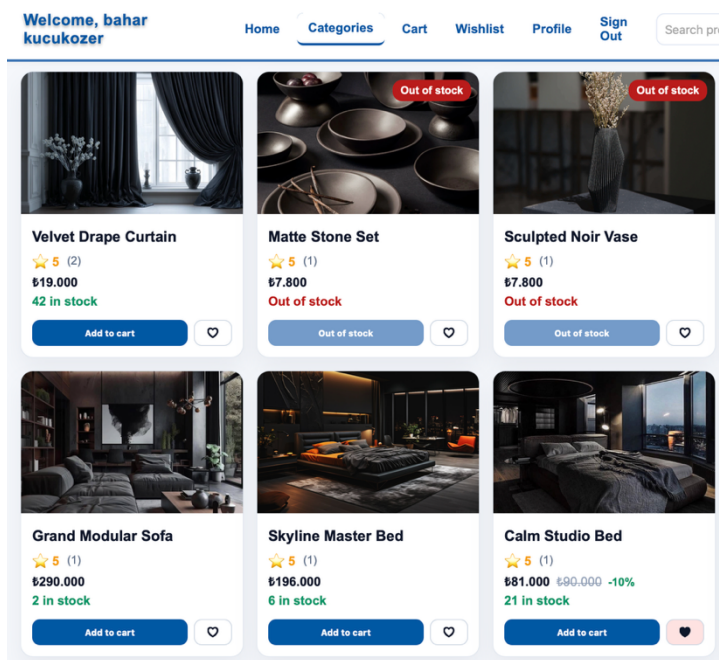
Bug Report 3– Sales Manager Page Causes Horizontal Overflow

Description

The **Sales Manager dashboard page** extends horizontally beyond the visible viewport, causing unwanted **horizontal scrolling**.
This layout issue breaks the intended responsive design and negatively affects the user experience, especially on smaller screens.

The overflow is caused by page elements (such as tables or containers) exceeding the viewport width without proper overflow handling or responsive constraints.

Steps to Reproduce

1. Log in as a **Sales Manager**.
2. Navigate to the **Admin workspace / Sales Manager dashboard**.
3. Observe the page layout.
4. Scroll horizontally or resize the browser window.

Expected Result

- The Sales Manager page should:
  - Fit entirely within the viewport width
  - Not cause horizontal scrolling
  - Maintain a responsive and clean layout across screen sizes
- Tables and containers should adapt to screen width or handle overflow properly.

Actual Result

- Page content extends to the right beyond the viewport.
- Horizontal scrolling appears.
- Layout looks broken and unpolished.

Impact

- Poor user experience for Sales Managers
- Breaks responsive design principles
- Makes the admin interface look unprofessional
- Can affect usability during demos

Severity: 🟡 Medium

Proposed Fix

- Review container and table widths in the Sales Manager dashboard.
- Ensure no element exceeds `100vw` or parent container width.
- Apply proper CSS rules such as:
    - `overflow-x: hidden` on the main layout container
    - Responsive table handling (e.g., fixed layout or controlled horizontal scroll inside tables)
- Align the admin layout with responsive design best practices.

Attachment / Evidence

Bug Report 4– Discount Notification Not Sent to Wishlist Users

Description

When a **Sales Manager applies a discount** to a product, users who have that product in their **Wishlist do not receive any notification**, even though the product is eligible for notification according to the system requirements.

The discount is applied successfully and reflected on the Categories page; however, **no notification is triggered or delivered** to users whose Wishlist contains the discounted product.

Steps to Reproduce

1. Log in as a customer.
2. Add a product to the **Wishlist**.
3. Log out.
4. Log in as a **Sales Manager**.
5. Apply a discount to the same product.
6. Log back in as the customer.
7. Observe notifications (or lack thereof).

Expected Result

- When a Sales Manager applies a discount:
  - All users who have the discounted product in their Wishlist should receive a notification.
- Notifications should be:
  - Automatically triggered by the backend
  - Delivered consistently (e.g., notification panel, badge, or message)
- This behavior should align with the project requirement that users are notified about discounts on Wishlist items.

Actual Result

- No notification is sent to users.
- Users are unaware that a product in their Wishlist has been discounted.
- Discount information is not propagated through the notification system.

Impact

- Breaks expected Wishlist–discount interaction
- Users may miss active discounts

- Reduces usability and feature completeness
- Does not fully satisfy Sales Manager discount notification responsibility

---

Severity

🟡 **Medium**

Proposed Fix

- Connect the discount application logic to the **notification system** on the backend.
- When a discount is applied:
  - Query users whose Wishlist contains the discounted product
  - Automatically generate notifications for those users
- Ensure notifications are:
  - Stored in the database
  - Displayed in the frontend notification panel
  - Consistent across sessions

Bug Report 5– Login Card Theme Inconsistent with Global Dark Mode

Description

On the **Login page**, the **Sign In card (login form container)** initially appears in **light mode**, while the rest of the page follows the global **dark theme**.

When the user starts typing into any input field (email or password), the **Sign In card suddenly switches to dark mode**, aligning with the global theme. This transition happens without an explicit user action and creates a noticeable visual inconsistency.

The issue appears to be caused by the Sign In card not correctly inheriting the global theme state on initial render.

Steps to Reproduce

1. Navigate to the **Login** page.
2. Observe the **Sign In card** appearance.
3. Click on the email or password input field.
4. Start typing any character.
5. Observe the Sign In card styling.

Expected Result

- The Sign In card should:
  - Render in **dark mode immediately**, consistent with the rest of the page
  - Maintain the same theme throughout all interactions
- UI components should correctly inherit the **global theme state** on initial render.

Actual Result

- Sign In card initially appears in **light mode**.
- Upon user input:
  - The card switches to **dark mode**
- The theme change is abrupt and visually noticeable.

Impact

- Visual inconsistency on a critical entry page
- Creates a perception of unstable UI behavior
- Reduces overall polish and professionalism of the application
- May negatively affect first-time user experience

Severity

🟡 **Medium**

Proposed Fix

- Ensure the Sign In card:
    - Correctly inherits the global dark theme on initial render
- Avoid applying default (light mode) styles to the login card before theme initialization.
- Centralize theme control so that:
    - UI components do not change theme based on input focus or form state updates
- Verify that theme-related classes or styles are not conditionally applied during re-renders triggered by input events.

Attachment / Evidence