Day 4 Documentation - Building Dynamic Frontend Components for Your Marketplace

1. Introduction

In this phase of the project, the goal was to enhance the frontend of the marketplace by developing dynamic components such as filters and product lists. Additionally, we focused on implementing a search functionality that interacts seamlessly with the displayed products. The key challenge was integrating both the filter system and search mechanism, ensuring smooth interaction and a polished user experience.

The marketplace app is designed to provide a dynamic browsing experience, where users can search and filter through different categories of products. On Day 4, we focused on frontend components that allow users to efficiently search for products and apply various filters, such as category, price range, and discount availability.

2. Task Breakdown

FilterComponent and ProductList:

- **FilterComponent**: The filter section allows users to apply specific filters to narrow down the products shown. Filters included:
    - **Price Range**: A price slider is provided for users to select the price range of products they want to view.
    - **Discount Only**: Users can toggle a checkbox to view products that offer discounts.
    - **New Product Only**: A checkbox is provided to filter new products.
- **ProductList**: The products displayed are dynamically fetched based on the applied filters. The product cards contain:
    - Each product shows an image (with a placeholder in case of missing images).
    - Products show the price, and if there is a discount, the discounted price is shown along with the original price.
    - If a product has a discount, a badge shows the percentage off.
    - Products have an "Add to Cart" button that triggers an action when clicked.

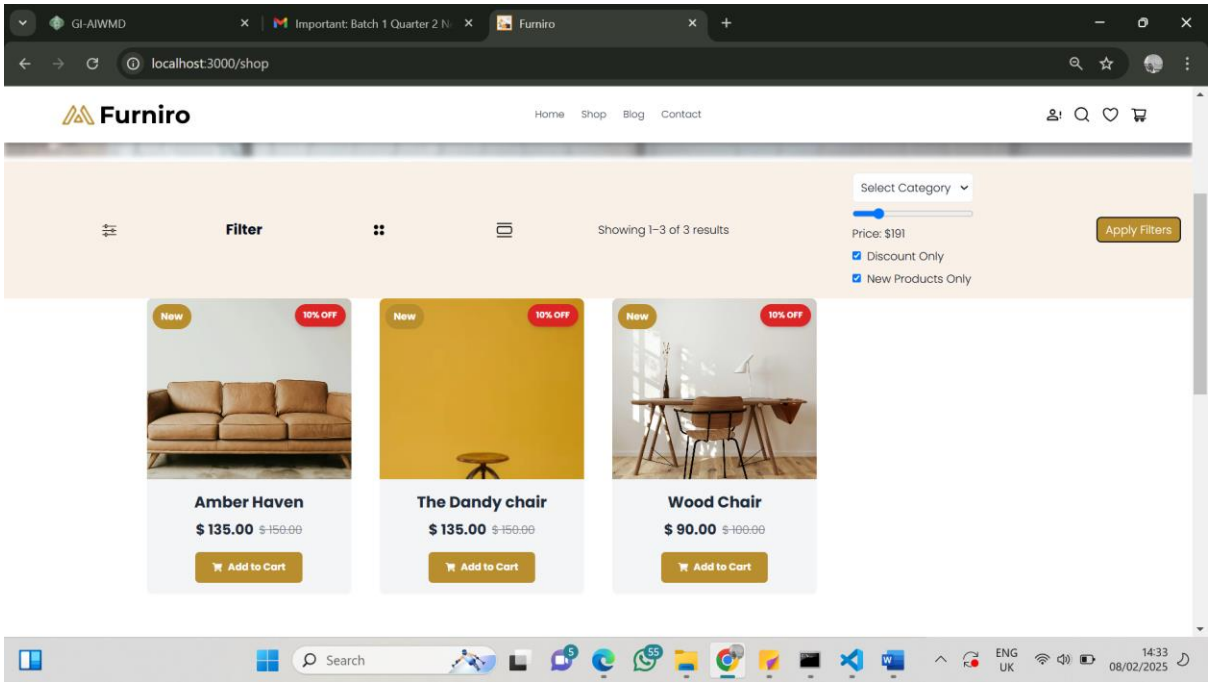

Figure 1: Filter Results

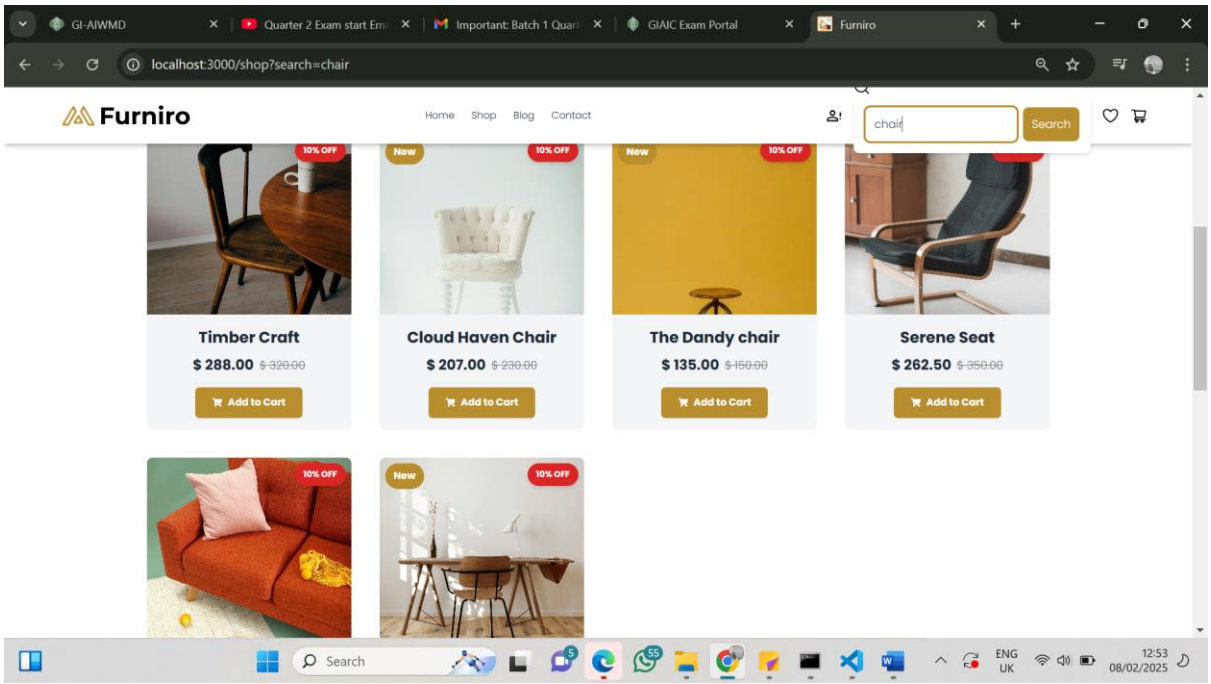

Figure 2: Search Results

3. Dynamic Frontend Development

To ensure the app is usable across different devices, both the **FilterComponent** and **ProductList** are designed responsively. The layout automatically adjusts based on the screen size:

- On mobile devices, the filter options are stacked vertically, and the product cards are displayed in a single column.
- On larger screens, filters appear side-by-side with the product cards in a grid layout.

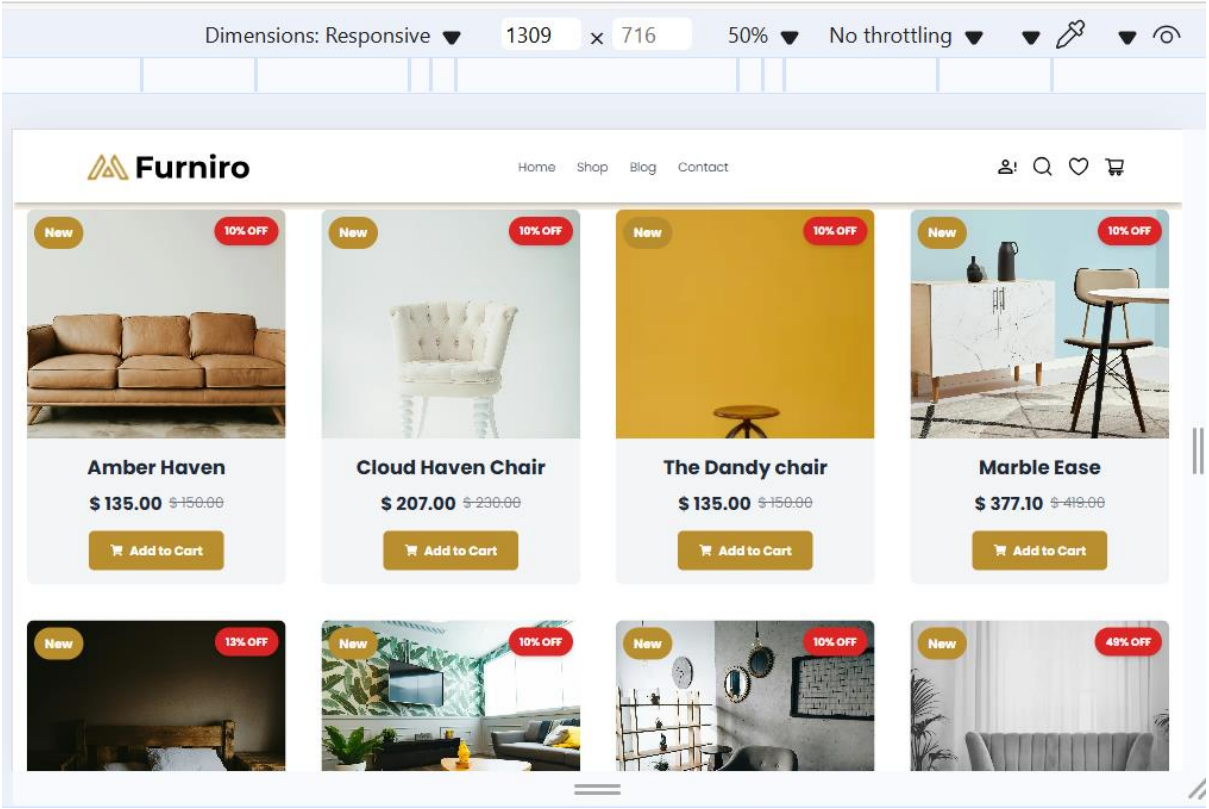

Figure 3: Desktop Screen

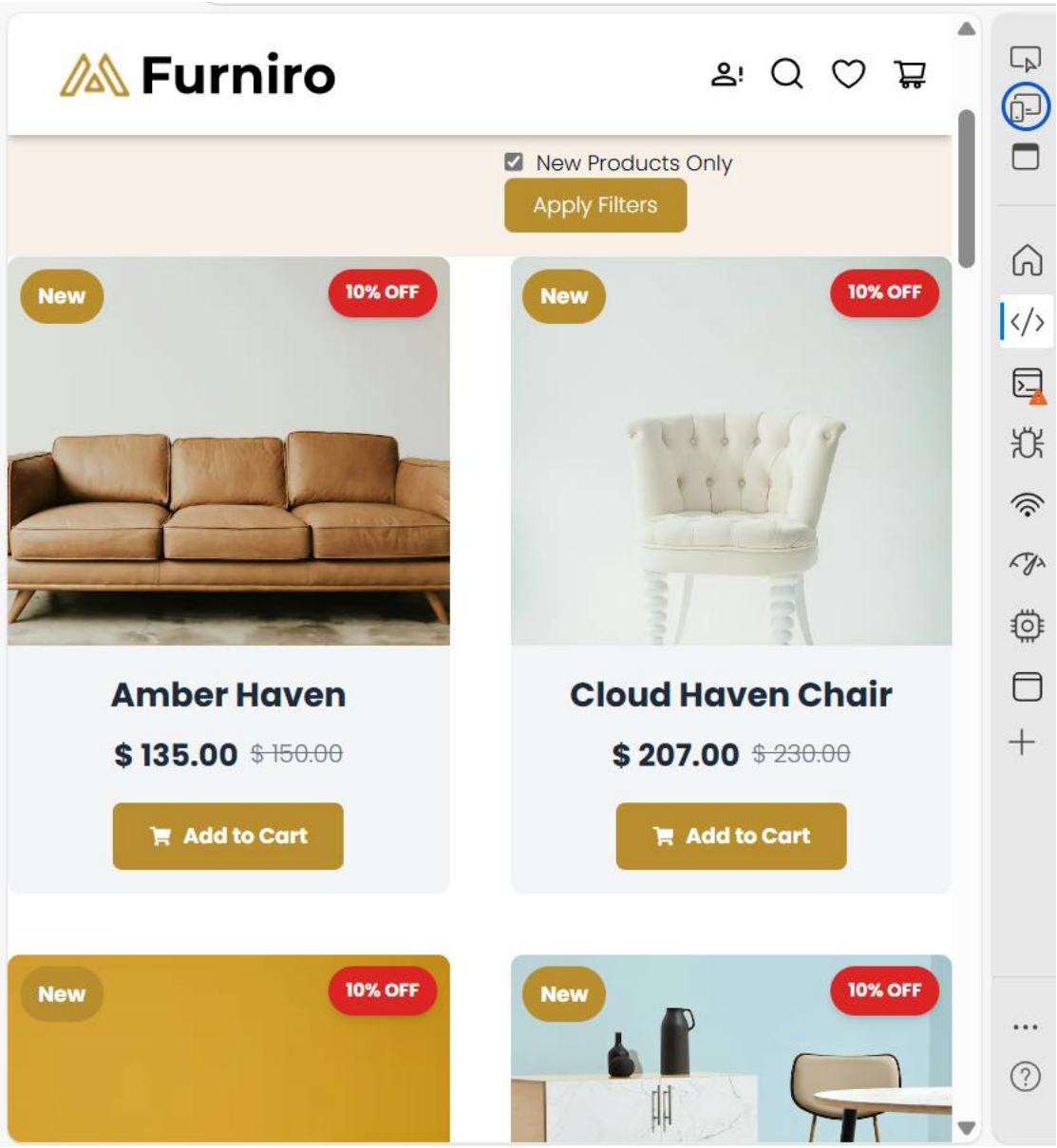

Figure 4: Mobile Screen

4. Functionality and Performance

Handling Data Fetching:

The product data is fetched dynamically from the database using **Sanity**. A key part of the functionality is dynamically updating the list of products based on the filters applied or the search term entered.

- **Fetching Product Data**: We used groq queries to fetch products based on the user's search criteria or applied filters. This ensures that only relevant products are shown, improving both the user experience and performance.

Loading States:

To enhance user experience, a loading spinner is displayed when products are being fetched from the database. This ensures the user is aware that data is being loaded and prevents confusion when the page is still updating.

5. Interactivity and User Flow

User Interaction with Filters/Search:

The main task was to create an intuitive interface for filtering and searching products:

- **Filters**: As users select categories, price ranges, or toggle checkboxes, the product list updates dynamically.
- **Search Bar**: The search bar listens for the user's input and filters the products accordingly.

Button and Click Interactions:

The **Add to Cart** button was implemented with a hover effect to indicate interactivity. When clicked, the user is given feedback through a success notification powered by **SweetAlert2**.

6. Accessibility and Usability

Accessibility Features:

- Keyboard navigation was supported throughout the application, especially in the **FilterComponent** and **ProductList**.
- We also made sure that interactive elements (e.g., buttons and checkboxes) have clear focus styles and that the page is navigable without a mouse.

Usability Testing:

To ensure the usability of the filters and search function, we conducted informal testing by simulating real-world use cases, such as:

- Applying multiple filters and searching for specific product names or categories.
- Testing responsiveness on different screen sizes.

7. Final Reflection and Challenges

Key Learnings:

- Handling dynamic updates to the product list based on filter changes and search input taught me a lot about managing state in React, especially with asynchronous data fetching.
- Ensuring a smooth flow between the filter and search functionalities was challenging but ultimately rewarding when everything came together.

8. Additional Screenshots

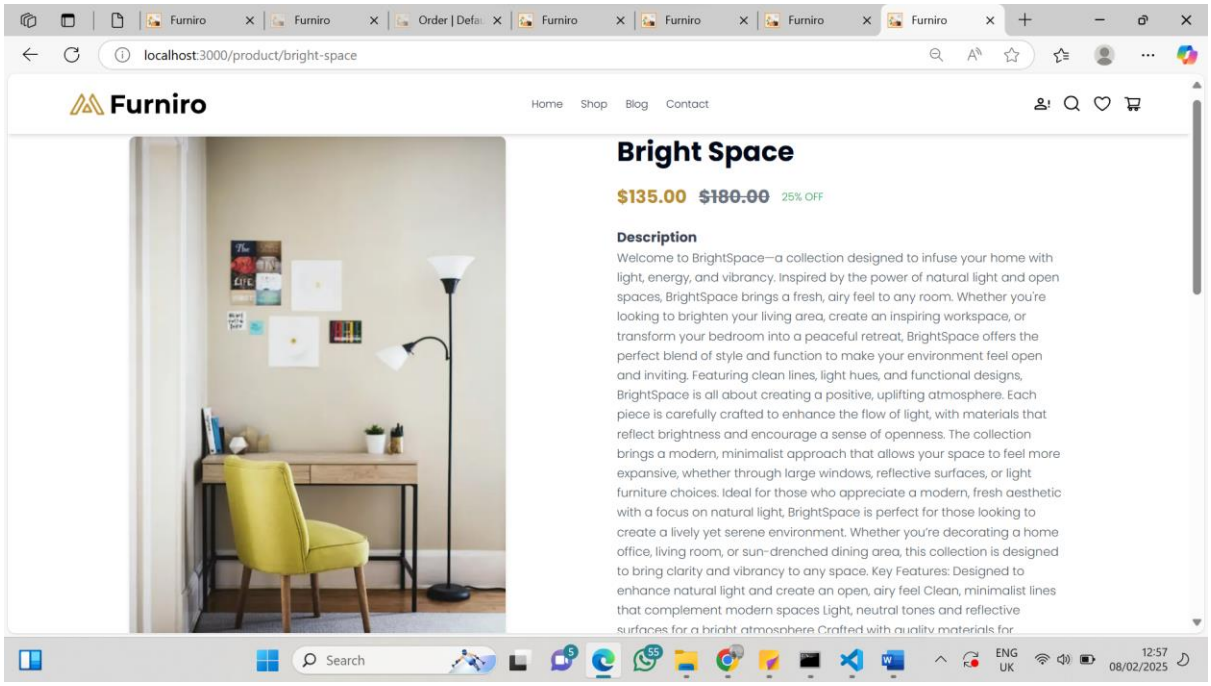

Figure 5: Dynamic Product Page

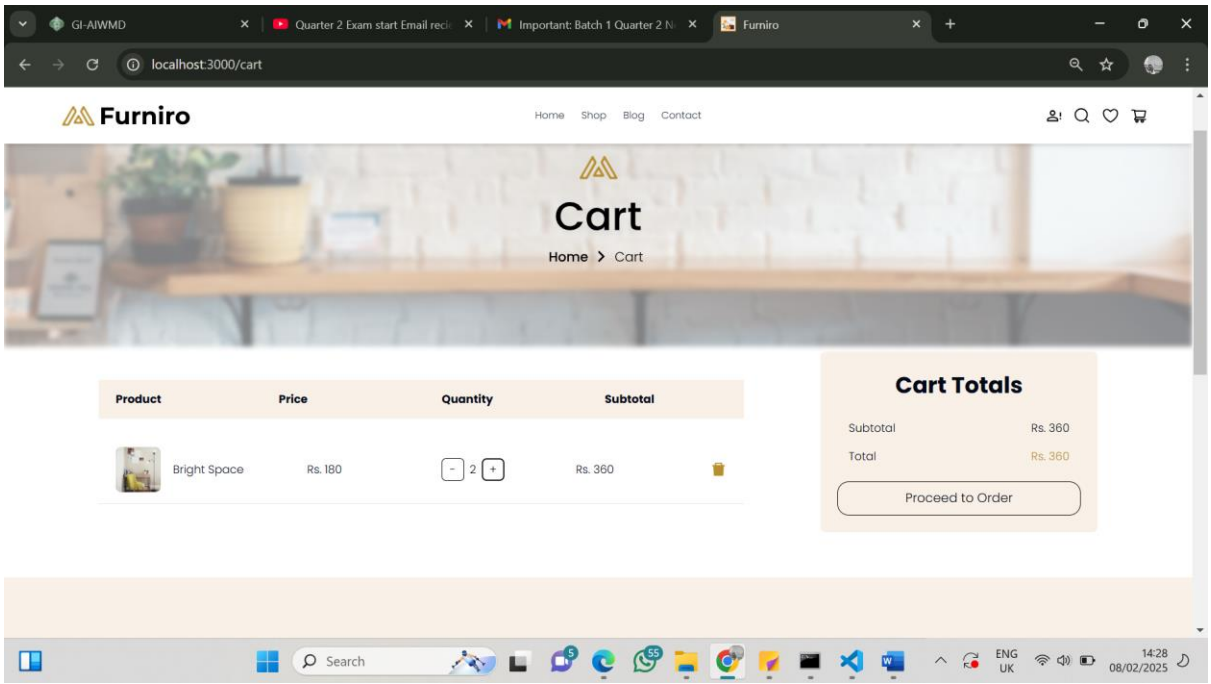

Figure 6: Cart

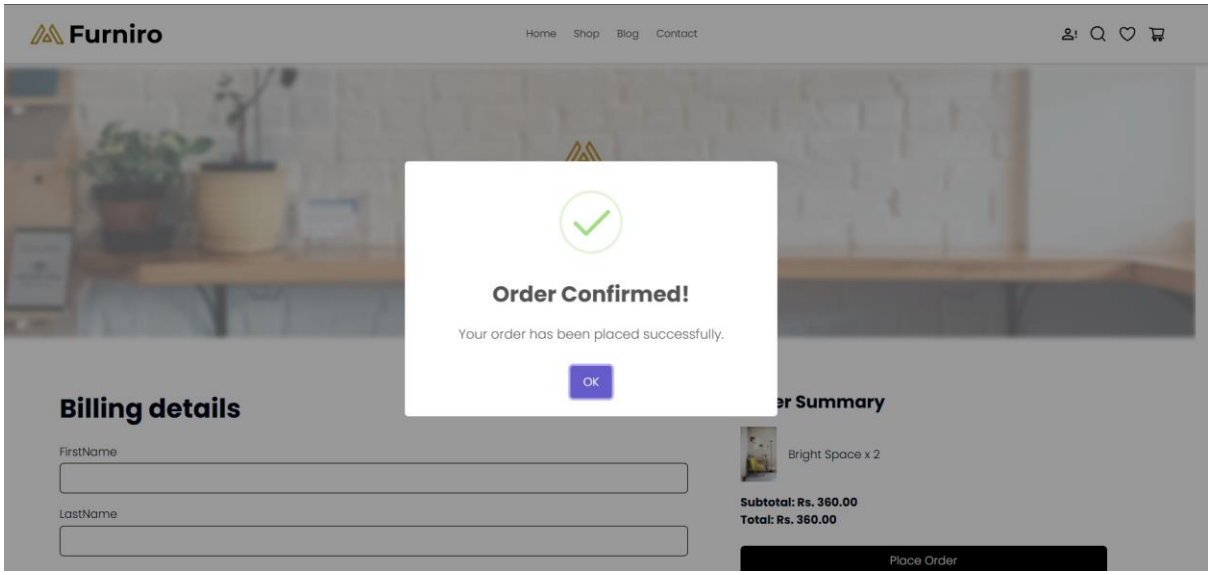

*Figure 7: Proceed to Order **to** Checkout Page*