
Day 4 – Hackathon Documentation

Kashaf Zeeshan (00461288)

Friday Morning Slot (Sir Hamzah Syed)

Template 08 (Comforty)

**Product Detail Page, Search
Functionality, Add to Cart &
Responsiveness for Comforty**

1. **Dynamic Routing:** Implemented dynamic routing for product detail pages.
2. **Search Functionality:** Integrated a search bar to find products.

3. **Add to Cart:** Enabled users to add products to their cart.
4. **Responsiveness:** Ensured a seamless experience across all screen sizes using Tailwind CSS.

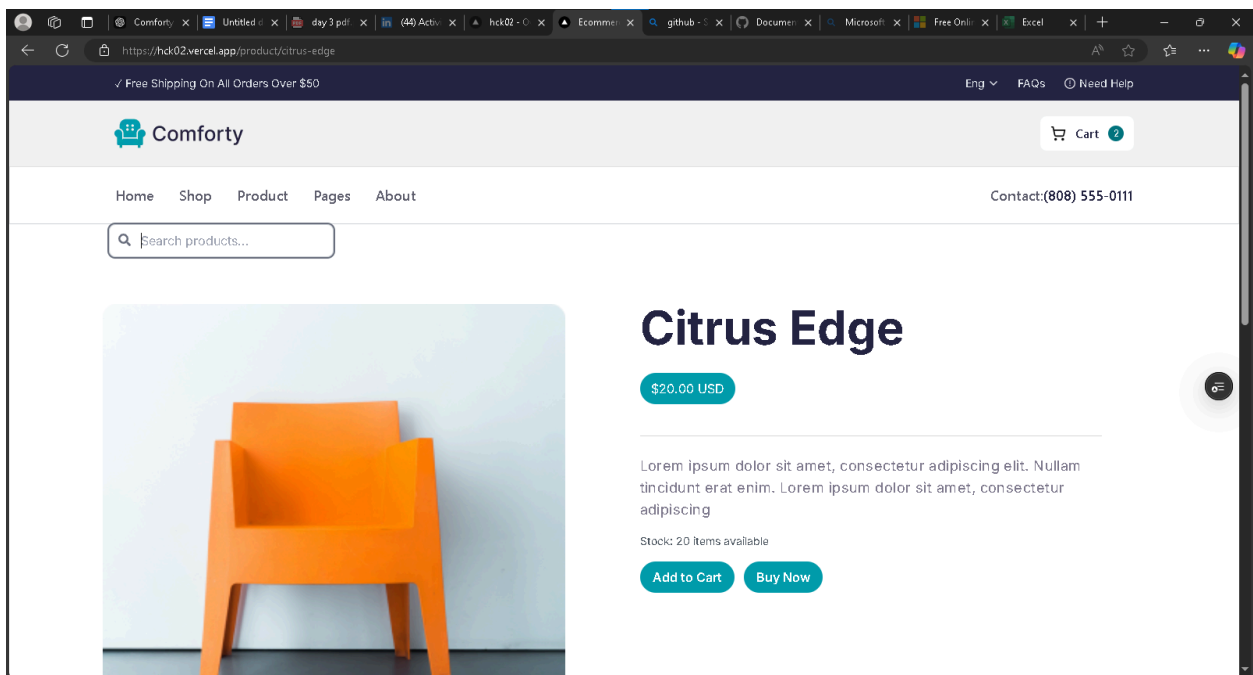
Product Detail Page & Dynamic Routing

For the Comforty Marketplace Plan, I developed a dynamic product detail page using **Next.js dynamic routing**. This allows each product to have a unique page where users can view complete product details.

Overview of the Implementation

- Created a **[slug].tsx** file to handle dynamic routing for product pages.

- Used **client-side fetching** with **fetch** to get product details from Sanity CMS using **GROQ queries**.
- Displayed product title, image, price, description, and category on the page.
- Ensured a **responsive layout** using Tailwind CSS for a seamless user experience on all devices.



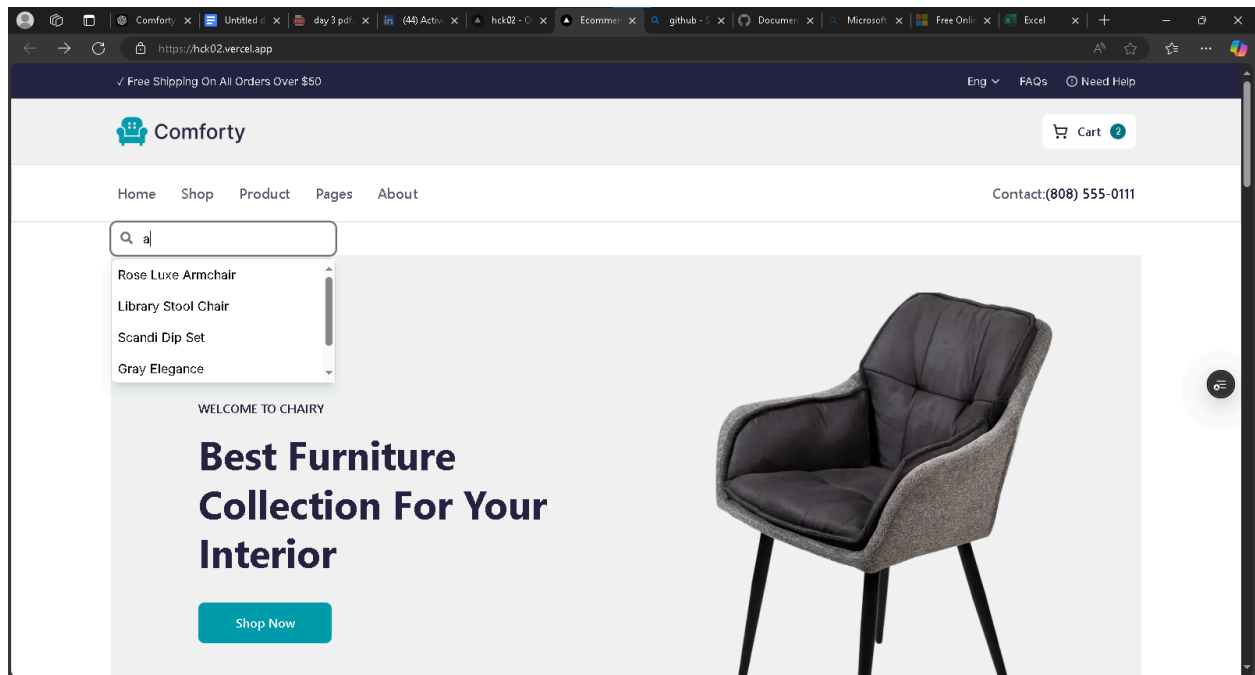
Search Functionality

To enhance user experience, I implemented a **search functionality** that allows users to find products by typing in the search bar.

When a user enters a query, relevant products appear, and clicking on a product directs them to its detail page.

Overview of the Search Implementation

- Used **useState** and **useEffect** to manage search input and filter results dynamically.
- Implemented a **debounced search function** for optimized performance.
- Filtered products based on the title and category.
- Integrated a **responsive search results dropdown** displaying matched products.
- Used **GROQ queries** to fetch product data from Sanity CMS dynamically.

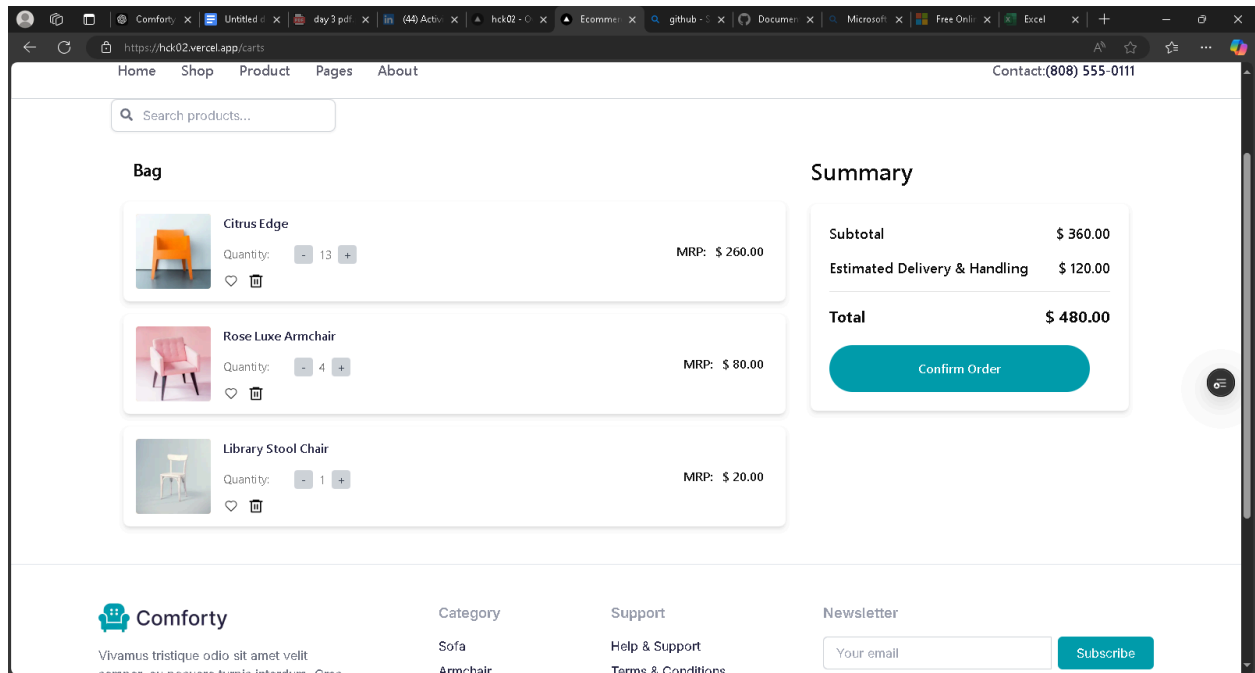


Add to Cart Functionality

I added an **Add to Cart** feature that allows users to select products and add them to their cart. The cart updates dynamically and maintains the selected items.

Overview of the Add to Cart Implementation

- **Stored cart data in localStorage** to persist across sessions.
- Added an **Add to Cart** button on the product detail page.
- Updated the cart item count dynamically using **localStorage**.
- **Ensured a mobile-friendly cart layout** using Tailwind CSS utilities.
- **Note:** Cart item count is not showing in the header. The count only updates on the button where products are shown after being added to the cart.

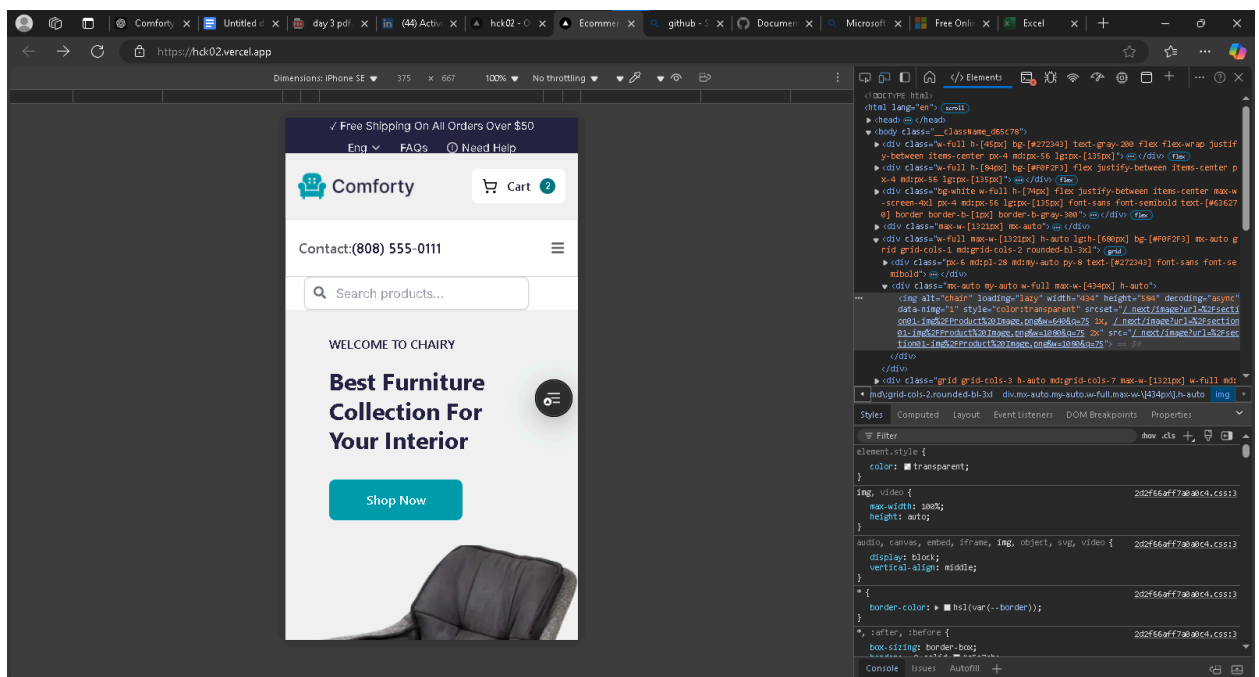


Responsiveness Implementation

To provide a smooth user experience, I ensured that all pages, including the product detail page, search results, and cart, are fully responsive using **Tailwind CSS**.

Overview of the Responsiveness Implementation

- Used Tailwind CSS breakpoints (**sm**, **md**, **lg**, **xl**) for adaptive layouts.
- Ensured flexible **grid and column structures** for mobile and tablet views.
- Used **object-fit: cover** and **max-w-full** for responsive images.
- Verified **cross-device compatibility** for a seamless shopping experience.



Conclusion:

- Implemented **dynamic routing** for product detail pages, ensuring each product has a unique URL
- Added **search functionality**, allowing users to quickly find their desired products.
- Integrated an **Add to Cart** feature, providing a seamless shopping experience.
- Successfully ensured **responsiveness** using Tailwind CSS, making the UI user-friendly across all devices.
- Successfully tested all features and ensured smooth user interaction with the system.