ASRA FURNITURE'S MARKETPLACE E-COMMERCE WEBSITE

Day 4: Dynamic Frontend Components

Components Implemented for the E-Commerce Website:

1. Add to Cart Component

I've implemented an **Add to Cart** feature that allows users to add items like **chairs**, **sofas**, and **tables** to their shopping cart. This component dynamically updates the cart based on user interactions, and the cart state is preserved even when users reload the page.

How it works:

- When a user clicks the Add to Cart button, the product is dynamically added to the cart stored in localStorage.
- The cart updates in real-time, showing the number of items and total price.
- I used React state and localStorage to persist the cart items even after page reloads.

2. Wishlist Component

The **Wishlist** component allows users to save products they're interested in purchasing later, such as specific **sofas** or **lighting** fixtures.

How it works:

- Similar to the Add to Cart functionality, when a user clicks the Add to Wishlist button, the product gets saved to their wishlist, which is also stored in localStorage.
- Users can easily add/remove items from their wishlist, and it is preserved even after they reload the page.

3. Dynamic Product Detail Pages

I have created **dynamic product detail pages** for each furniture item, which provide users with more in-depth information such as **images**, **price**, **description**, and **product specifications**.

How it works:

- Each product is uniquely routed based on its slug (e.g., /product/sofa-1), and the product details are fetched dynamically from Sanity CMS.
- This ensures that when a user clicks on a product, they are directed to its unique page with real-time data pulled from Sanity.

4. Product Filters and Sorting

I've also implemented **product filters** and **sorting** options to enhance the product browsing experience for users. Filters allow users to narrow down their search by categories like **Sofas**, **Tables**, or **Chairs**, while sorting options help them sort products by **price** or **popularity**.

How it works:

- The filters and sorting options fetch product data from Sanity CMS, and the product list updates dynamically based on user selections.
- I created category filters and sorting dropdowns that call the relevant API endpoints to fetch filtered or sorted product data.

5. Blog Section for Furniture Tips

I've added a **Blog Section** to the site where users can find articles about **furniture care**, **styling tips**, and other useful content. This section is managed through **Sanity CMS**, making it easy to add and update blog posts without needing to touch the code.

How it works:

- Each blog post is created in Sanity CMS, and the blog listing page dynamically fetches and displays blog posts from the CMS.
- This keeps the content fresh and allows content creators to easily add new articles.

6. State Preservation Across Page Reloads

For a seamless user experience, I've ensured that the **cart** and **wishlist** components persist across page reloads. By saving the cart and wishlist in **localStorage**, users can reload the page and still have their data available.

How it works:

The Add to Cart and Wishlist components save data to localStorage.
When the page is reloaded, the saved cart and wishlist items are fetched from localStorage and displayed correctly.

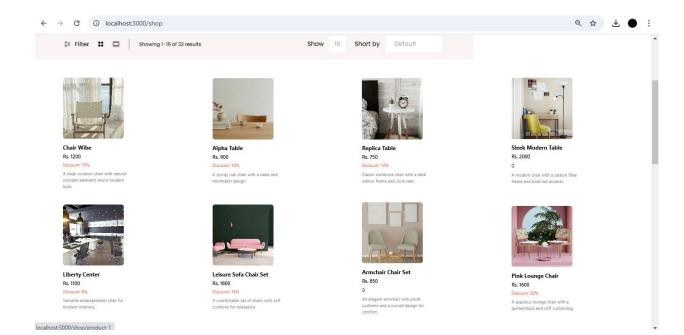
Final Ananlysis

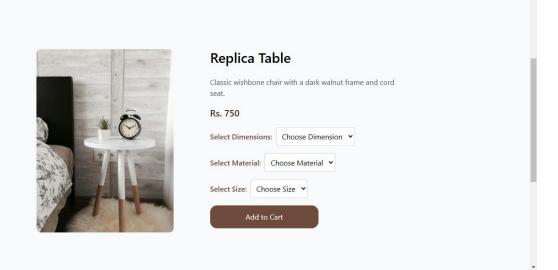
These dynamic features—Add to Cart, Wishlist, Product Filters, Blog, and State Preservation—were all integrated into my furniture e-commerce website. I utilized Sanity CMS to manage dynamic product content (product details, blog posts, etc.), ensuring a smooth and responsive user experience.

By leveraging **localStorage** for cart and wishlist persistence, and using dynamic routing for product pages, the site becomes highly interactive and user-friendly. This setup also allows for easy updates to products and content via **Sanity CMS**, without requiring code changes.

Self-Validation Checklist:

Task	Status
Frontend Component Development	✓
Styling and Responsiveness	✓
Code Quality	✓
Documentation and Submission	✓
Final Review	✓





Log In	Register
Username or email address	Email address
Password	A link to set a new password will be sent to your email address.
Remember me	Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our privacy policy.
Submit Lost Your Password?	Register

