Day 4 - Dynamic Frontend Components - COMFORTY

1. Introduction

Day 4 of the hackathon focused on building dynamic frontend components to display and interact with the data imported into Sanity CMS on Day 3. The aim was to create a scalable, responsive, and user-friendly interface for the furniture marketplace.

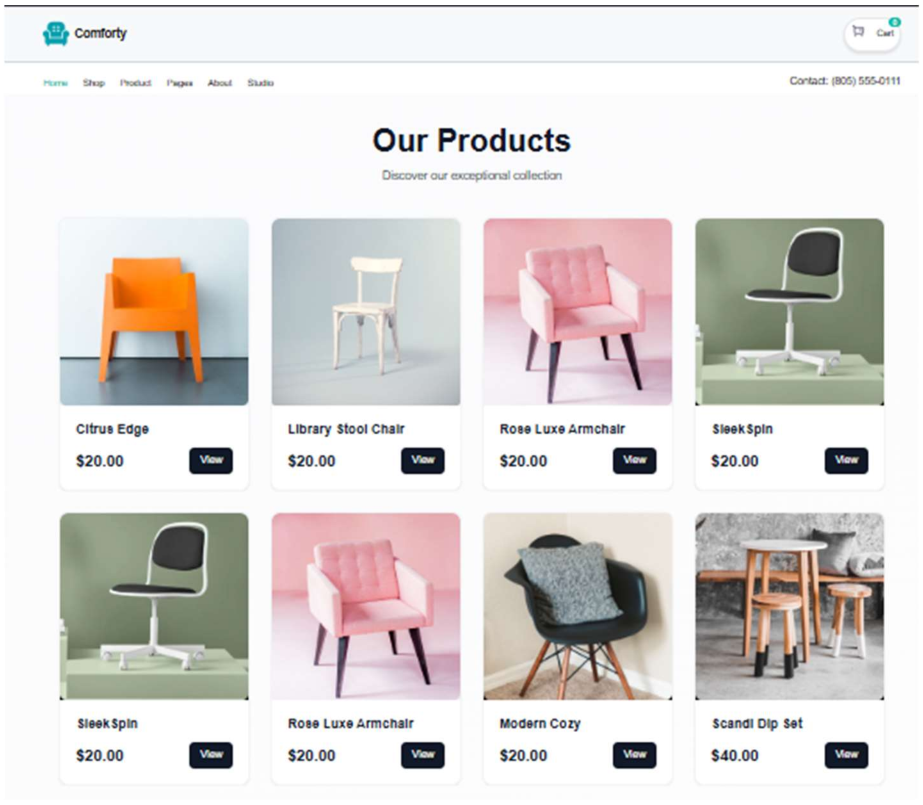
---

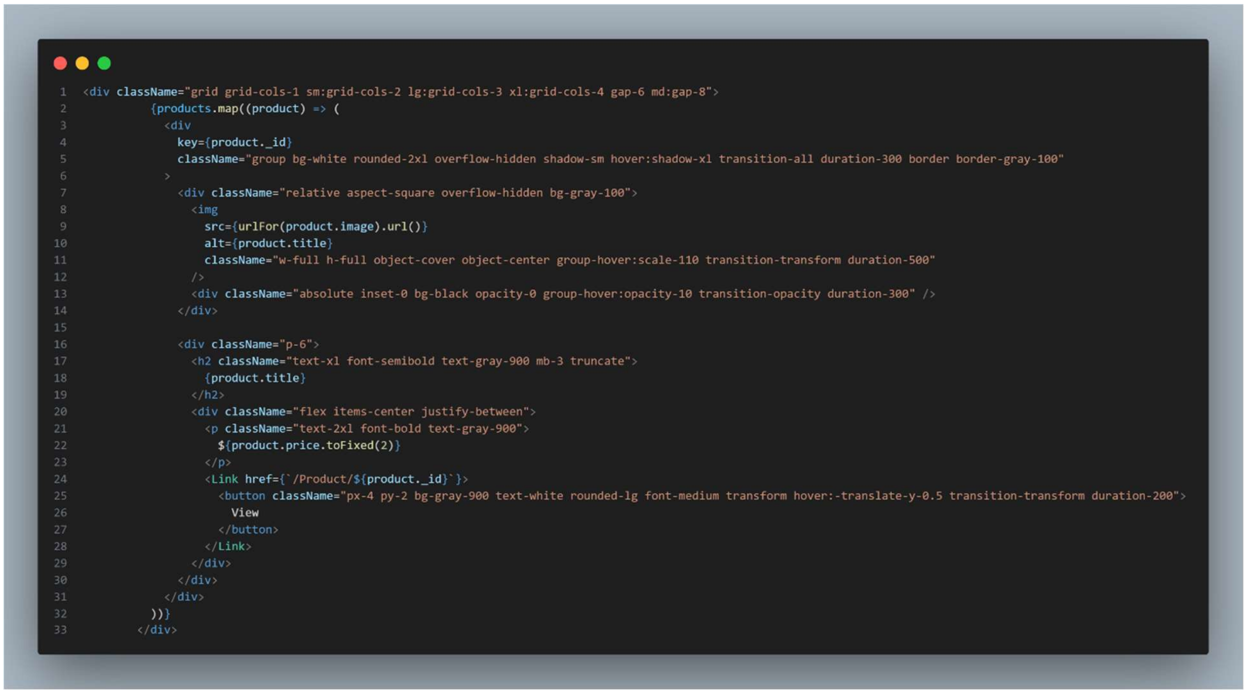
2. Key Components Implemented

Product Page

- **Purpose**: To display a dynamic list of products fetched from Sanity CMS.
- **Implementation**:
    - Fetched product data using Sanity's GROQ queries and displayed it in a grid layout.
    - Rendered product cards showing the name, price, image, and availability status.
    - Utilized reusable components for consistency and scalability.
- **Features**:
    - Responsive design for optimal viewing across devices.
    - Lazy loading for improved performance.

Snippets of Product Listing Page and its Code:



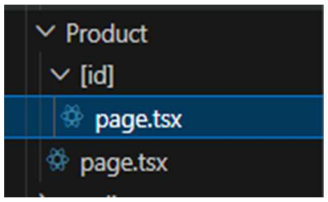


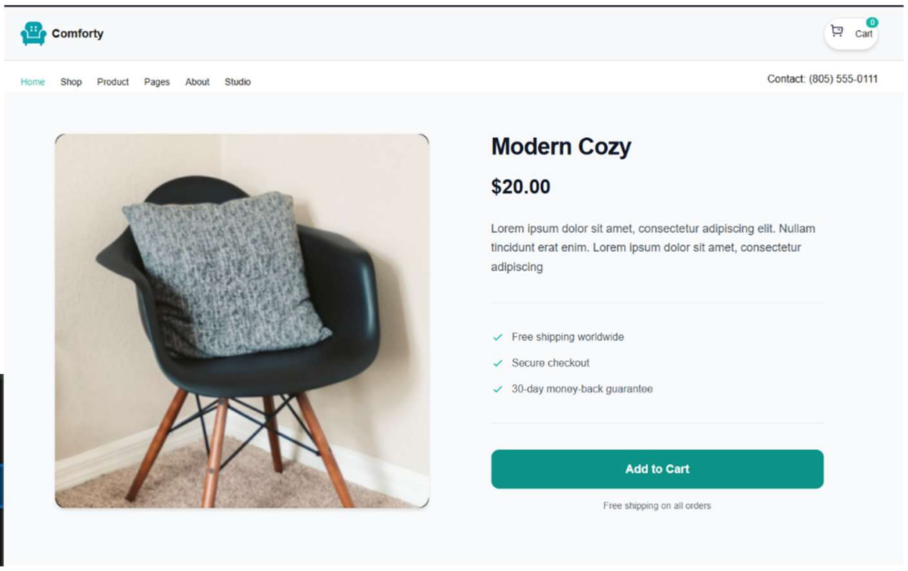
```
<div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-3 xl:grid-cols-4 gap-6 md:gap-8">
          {products.map((product) => (
            <div
              key={product._id}
              className="group bg-white rounded-2xl overflow-hidden shadow-sm hover:shadow-xl transition-all duration-300 border border-gray-100"
            >
              <div className="relative aspect-square overflow-hidden bg-gray-100">
                <img
                  src={urlFor(product.image).url()}
                  alt={product.title}
                  className="w-full h-full object-cover object-center group-hover:scale-110 transition-transform duration-500"
                />
                <div className="absolute inset-0 bg-black opacity-0 group-hover:opacity-10 transition-opacity duration-300" />
              </div>

              <div className="p-6">
                <h2 className="text-xl font-semibold text-gray-900 mb-3 truncate">
                  {product.title}
                </h2>
                <div className="flex items-center justify-between">
                  <p className="text-2xl font-bold text-gray-900">
                    ${product.price.toFixed(2)}
                  </p>
                  <Link href={`/Product/${product._id}`}>
                    <button className="px-4 py-2 bg-gray-900 text-white rounded-lg font-medium transform hover:-translate-y-0.5 transition-transform duration-200">
                      View
                    </button>
                  </Link>
                </div>
              </div>
            </div>
          ))}
        </div>
```

Product Detail Page

- **Purpose**: To provide detailed information about a specific product.
- **Implementation**:
    - Used Next.js dynamic routing (pages/product/[id].tsx) to create individual product pages.
    - Fetched and displayed detailed product data, including:
        - Name, price, description, images, and stock availability.
    - Integrated user-friendly navigation to return to the product listing.
- **Features**:
    - Dynamic URL-based navigation.
    - Clean and informative UI for an enhanced user experience.

Snippets of Product Detailed Page and Dynamic Routing:





Modern Cozy

$20.00

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing

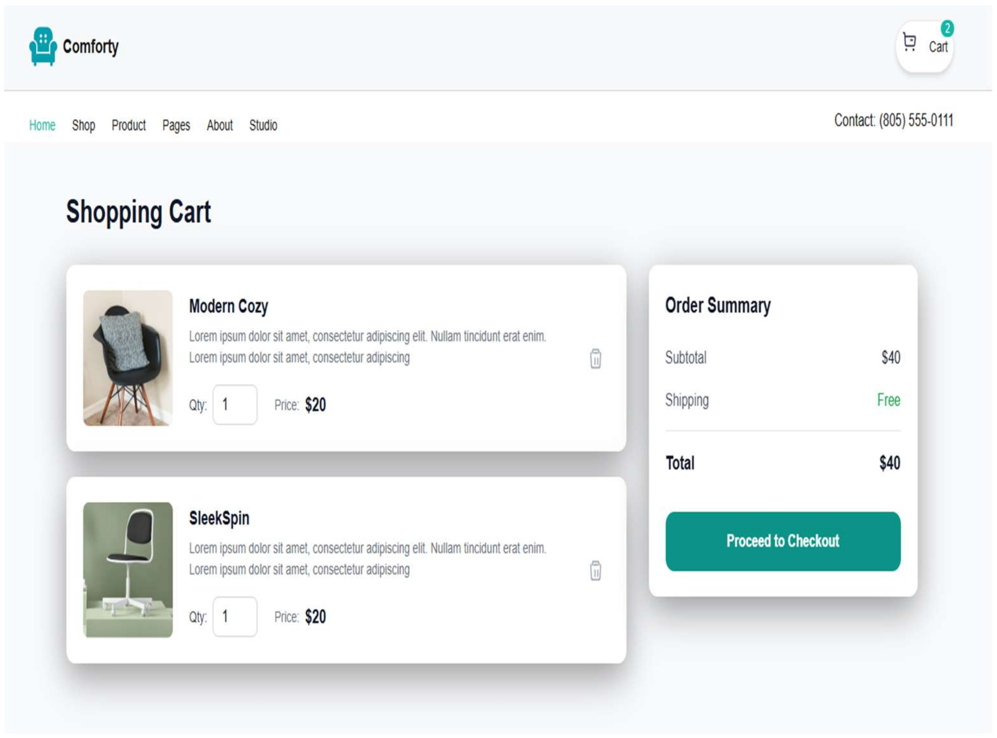
- Free shipping worldwide
- Secure checkout
- 30-day money-back guarantee

Free shipping on all orders

Cart Functionality

- **Purpose**: To allow users to add products to their cart and view selected items.
- **Implementation**:
  - Created a CartContext using React Context API for global state management.
  - Implemented "Add to Cart" functionality on the product detail page.
  - Cart features included:
    - Dynamic item count in the header.
    - View, update, and remove items from the cart.
  - Persisted cart state to local storage to retain data between sessions.

Snippets of Cart Page:



Contact: (805) 555-0111

Shopping Cart

Modern Cozy

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing

Qty: 1 Price: $20

SleekSpin

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam tincidunt erat enim. Lorem ipsum dolor sit amet, consectetur adipiscing

Qty: 1 Price: $20

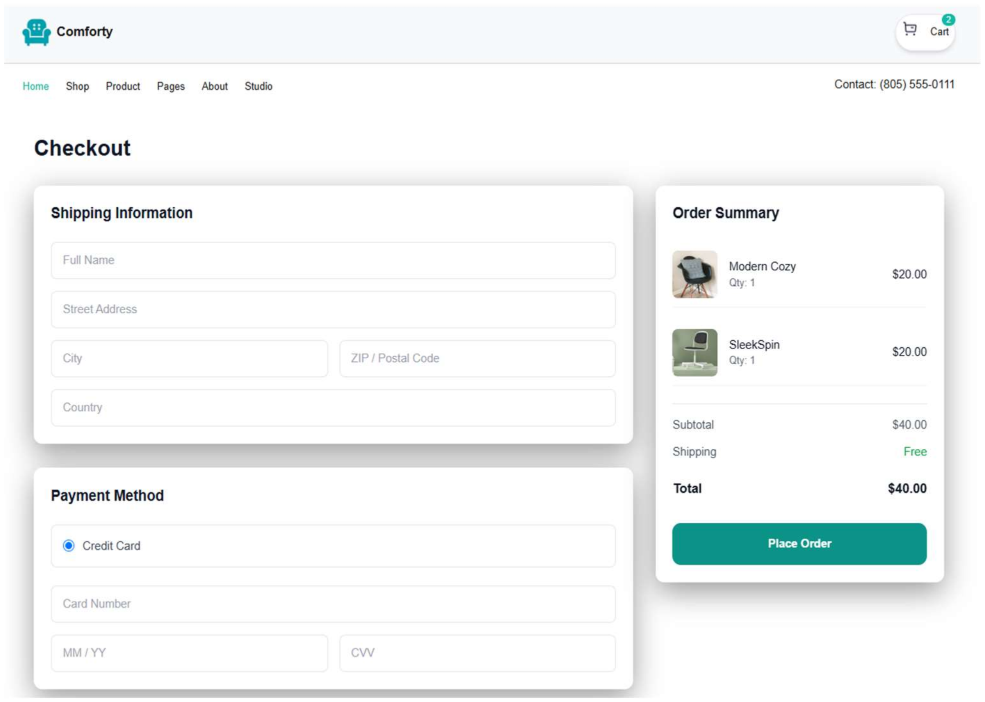
Order Summary

Subtotal	$40
Shipping	Free
Total	$40

Checkout Page

- **Purpose**: To finalize the purchase process.
- **Implementation**:
  - Designed a multi-step checkout form to collect:
    - Billing and shipping details.
    - Payment information (mock implementation).
  - Displayed a summary of the cart with the total price.
- **Features**:
  - Clean and intuitive UI for user convenience.
  - Error handling for incomplete or invalid inputs.

Snippets of CheckOut Page:



3. Challenges and Solutions

- **API Integration with Sanity CMS**:
  - **Challenge**: Ensuring accurate data fetching and rendering for the product pages.
  - **Solution**: Verified the GROQ queries and implemented error handling to manage API response issues.
- **State Management**:
  - **Challenge**: Managing global cart state efficiently.
  - **Solution**: Utilized Context API for simplicity and local storage for persistence.
- **Dynamic Routing**:
  - **Challenge**: Dynamically generating pages for each product using Next.js.
  - **Solution**: Leveraged the getStaticPaths and getStaticProps functions to pre-render pages at build time.

---

4. Best Practices Followed

- Modular and reusable component design for scalability.
- Responsive design using Tailwind CSS to ensure compatibility across devices.
- Optimized data fetching with Sanity GROQ and React query hooks.
- Proper error handling for robust user interactions.

---

6. Project Link

https://giaic-market-place-e-commerce-hackathon.vercel.app/

---

7. Conclusion

Day 4 was focused on transforming static data into an interactive and functional user interface. By completing the product pages, cart functionality, and checkout flow, the project demonstrates a scalable approach to building an eCommerce platform.