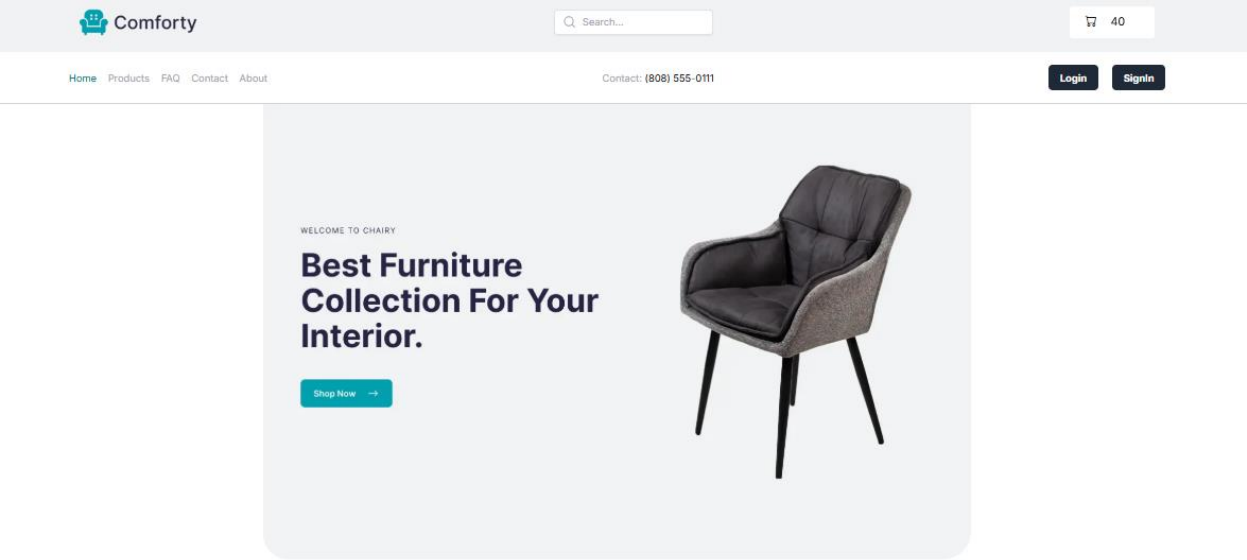
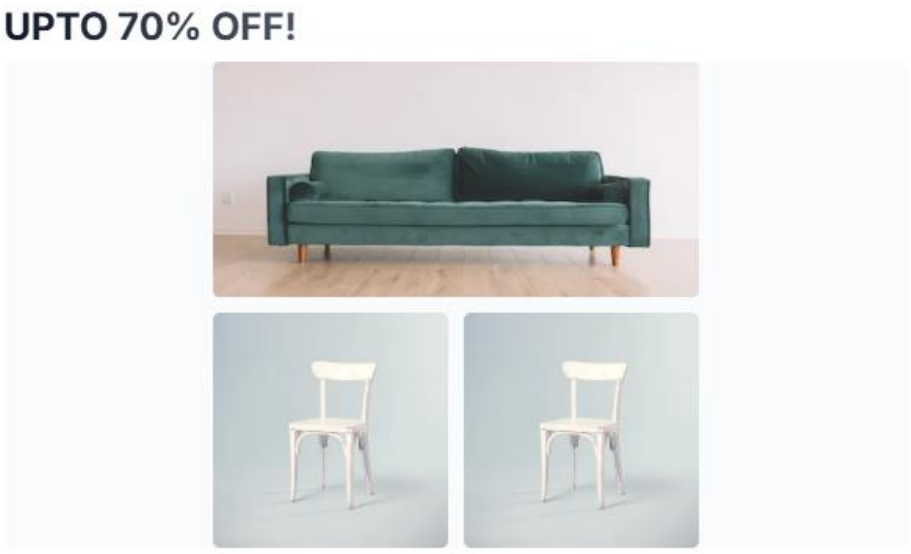
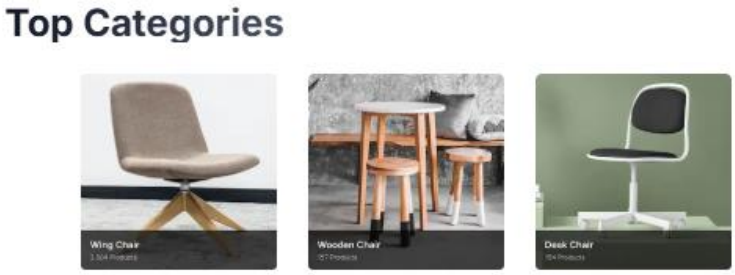
Day 4 Plan: Building Dynamic Frontend Components for My Marketplace

Objective:
To design and develop dynamic frontend components that display marketplace data fetched from Sanity CMS or APIs, ensuring modularity, reusability, and responsiveness.

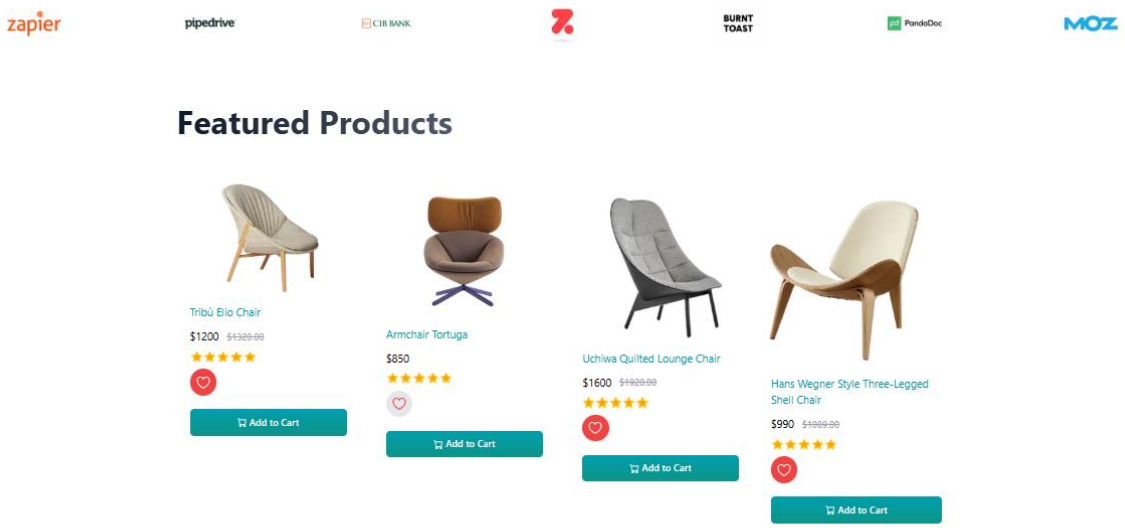






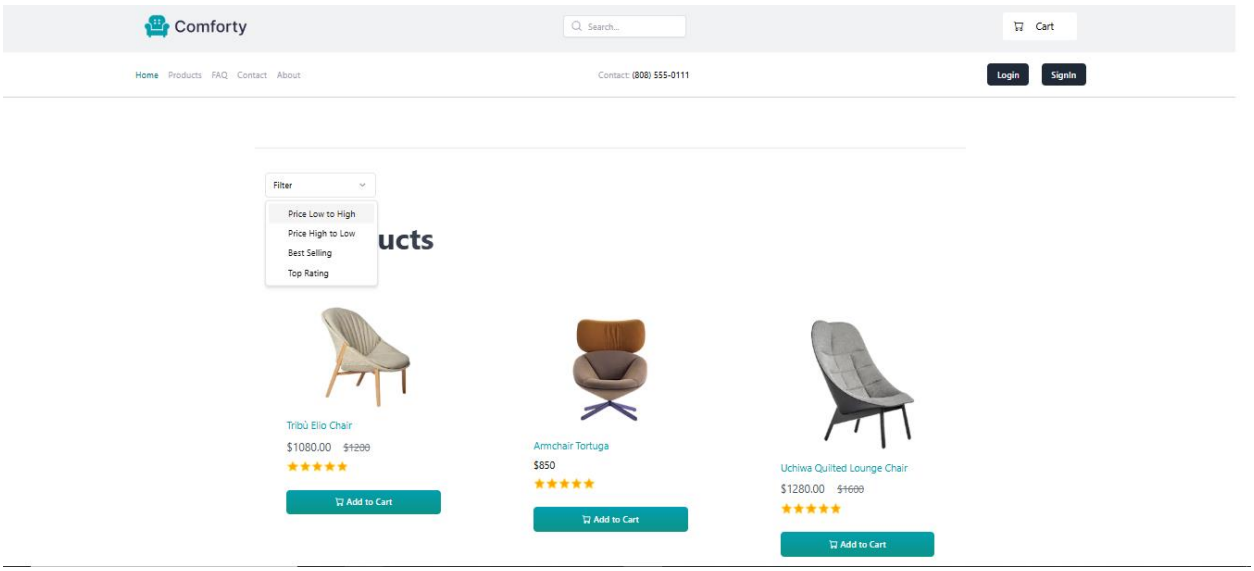
Step 1: Setup and Preparation

1. **Ensure API Integration:**
    a. Verify that my Next.js project is connected to Sanity CMS or the chosen API.
    b. Test data fetching to confirm that the data is available and correctly structured.
2. **Project Structure:**
    a. Organize my project files into a clear folder structure (e.g., `components`, `pages`, `styles`, `utils`).
    b. Set up a global state management system (e.g., React Context or Redux) if needed.
3. **Styling Setup:**
    a. Choose a styling library (e.g., Tailwind CSS, styled-components) and ensure it's properly configured.
    b. Set up responsive design breakpoints for mobile, tablet, and desktop views.



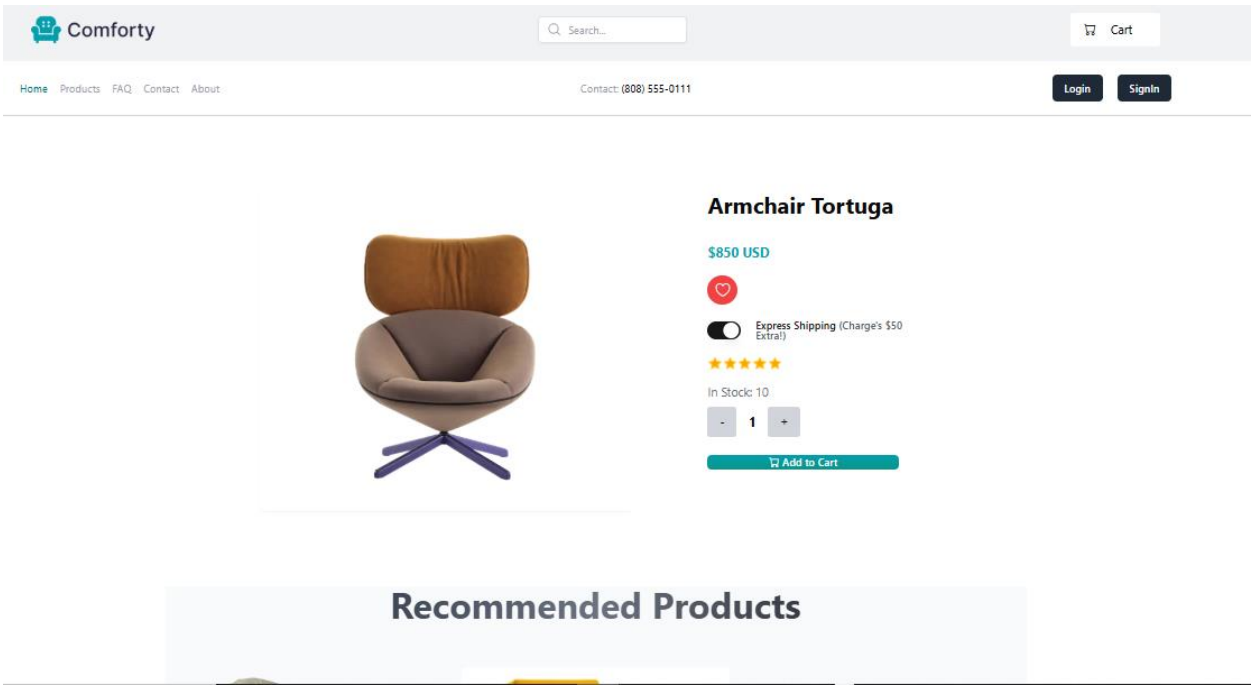
Step 2: Build Core Components

1. **Product Listing Component:**
    a. Create a `ProductCard` component to display product details (name, price, image, stock status).
    b. Implement a `ProductList` component to render multiple `ProductCard` components in a grid layout.
    c. Fetch and display product data dynamically from Sanity CMS or API.
2. **Product Detail Component:**
    a. Set up dynamic routing in Next.js for individual product pages (e.g., `/product/[id]`).
    b. Create a `ProductDetail` component to display detailed product information (description, price, sizes, colors).
    c. Fetch product data based on the dynamic route parameter (`id`).
3. **Category Component:**
    a. Build a `CategoryFilter` component to display categories fetched from the data source.
    b. Implement functionality to filter products based on the selected category.
4. **Search Bar:**
    a. Create a `SearchBar` component with a search input field.
    b. Implement search functionality to filter products by name or tags.



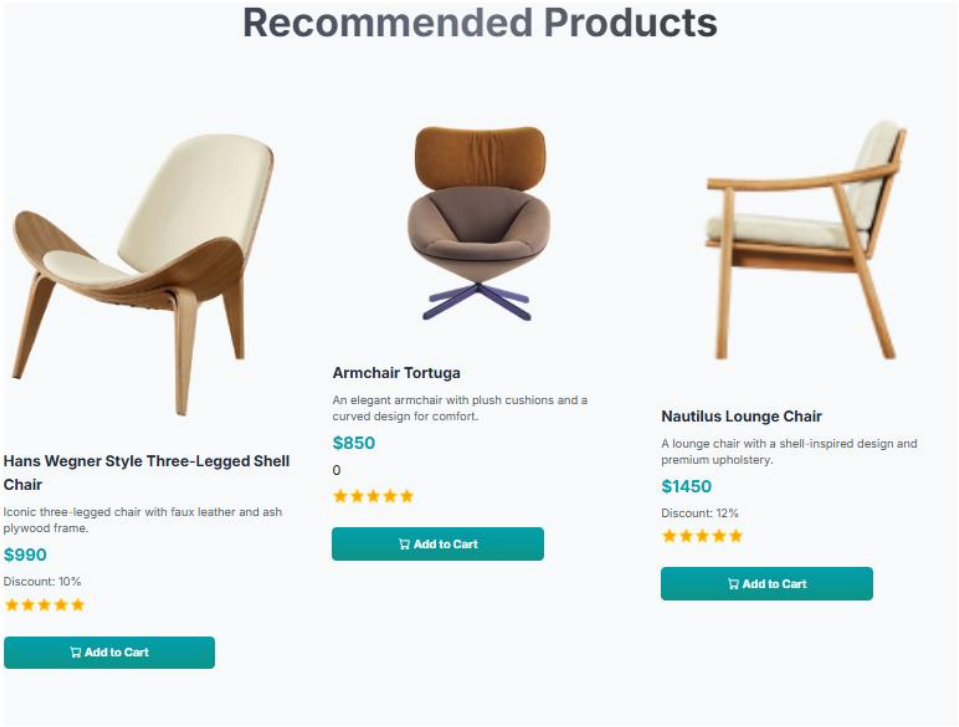
Step 3: Implement Additional Features

1. **Cart Component:**
    a. Build a `Cart` component to display added items, quantity, and total price.
    b. Use state management
2. **Wishlist Component:**
    a. Create a `Wishlist` component to allow users to save products.
    b. Use local storage or a global state management tool to persist wishlist data.
3. **Pagination Component:**
    a. Implement a `Pagination` component to break down large product lists into manageable pages.
    b. Add previous and next buttons or numbered pagination.



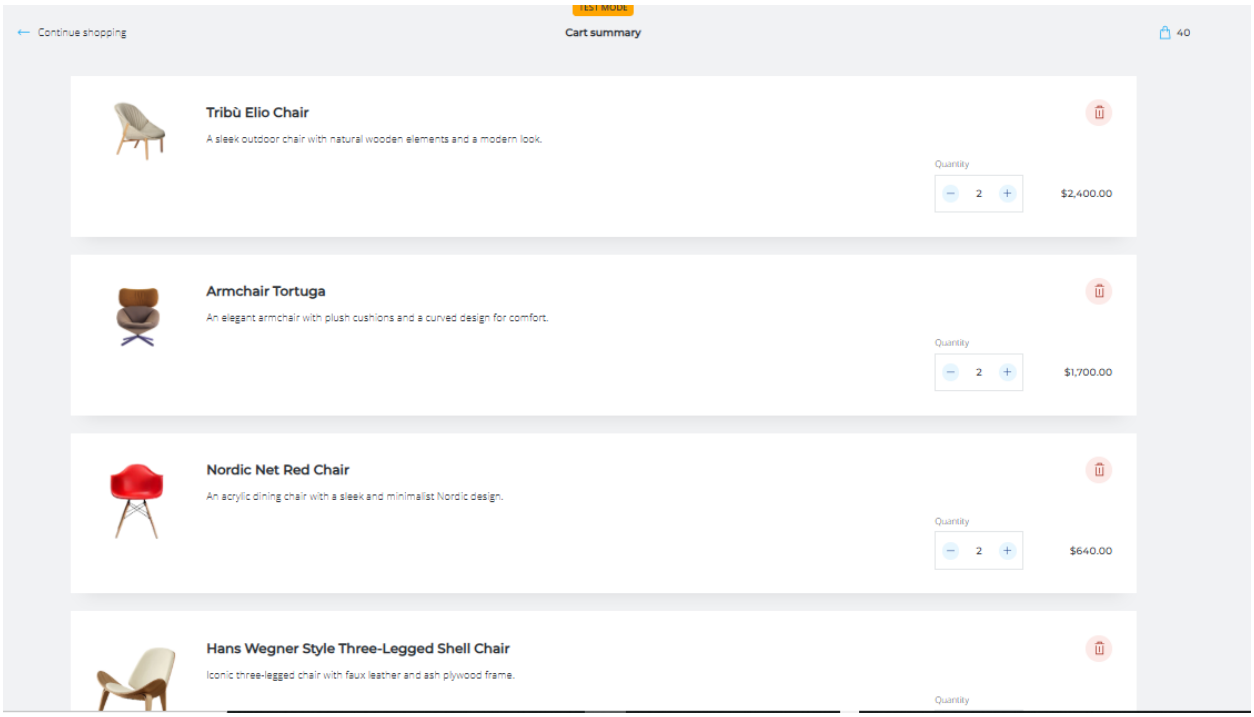
Step 4: Enhance User Experience

1. **Footer and Header Components:**
    a. Create consistent `Header` and `Footer` components with navigation links
    b. Ensure responsiveness and accessibility.
2. **Notifications Component:**
    a. Implement a `Notifications` component to show real-time alerts (e.g., adding to cart, errors, successful purchases).
    b. Use toast notifications or modal windows.
3. **Responsive Design:**
    a. Ensure all components are responsive and look professional across different devices (mobile, tablet, desktop).



Step 5: Advanced Components

1. **Reviews and Ratings Component:**
    a. Build a `Reviews` component to allow users to view and submit reviews for products.
    b. Display average ratings and individual reviews dynamically.
2. **Filter Panel Component:**
    a. Create a `FilterPanel` component with advanced filtering options (e.g., price range, brand selection, availability).
3. **Checkout Flow Component:**
    a. Implement a multi-step checkout form with fields for billing/shipping address and payment details (mock implementation)



Conclusion of My Plan:

By following this structured plan for Day 4, I will have successfully built a dynamic and responsive frontend for my marketplace. The core components, such as the **Product Listing**, **Product Detail**, **Category Filter**, and **Search Bar**, will be fully functional and integrated with data from Sanity CMS or APIs. Additionally, I will have implemented advanced features like the **Cart**, **Wishlist**, **Pagination**, and **Related Products** components to enhance the user experience.