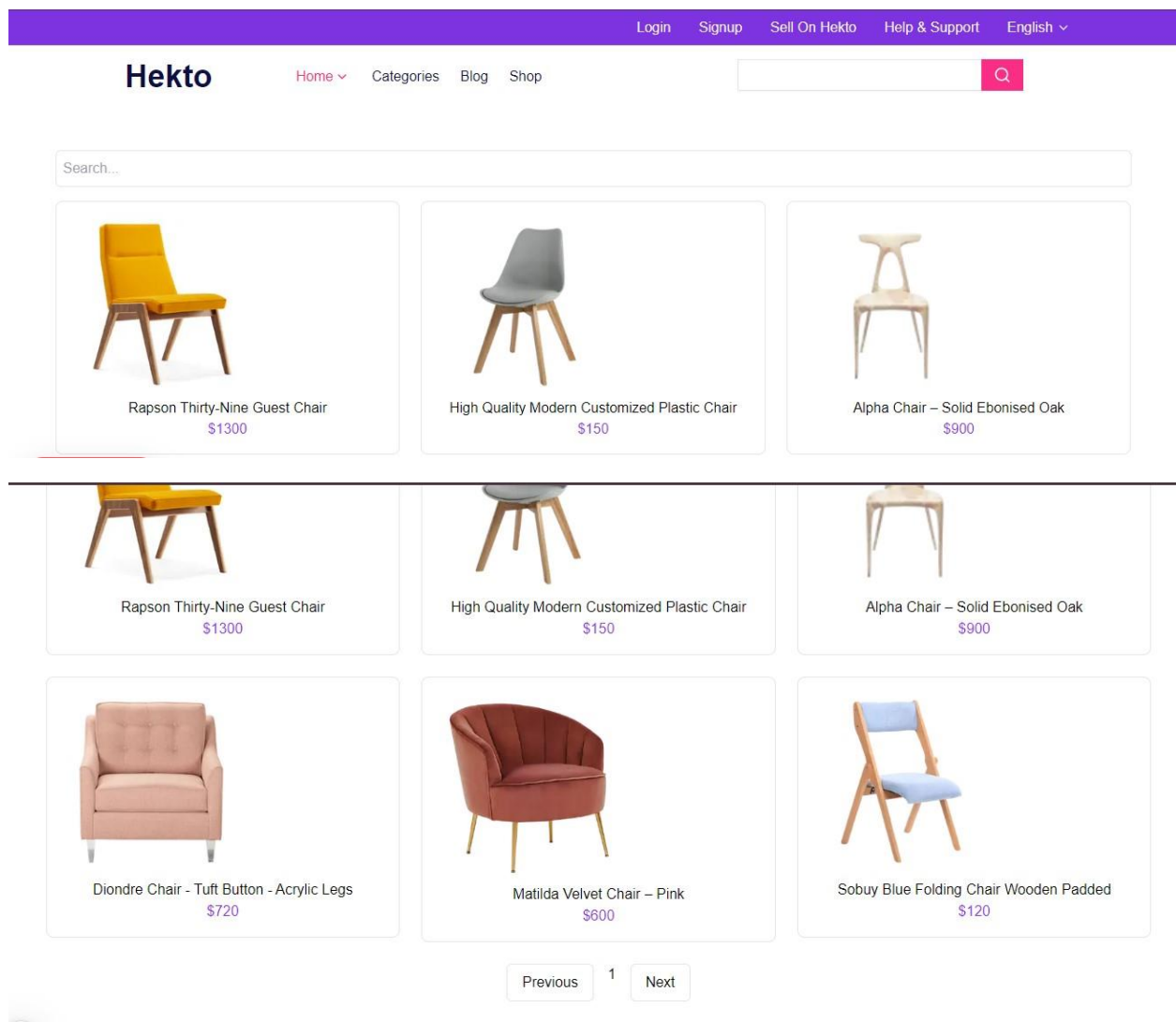


# Dynamic Frontend Components

## 1. Product Listing Page

### Key Features:

- ✓ Fetches and displays all products from Sanity CMS dynamically.
- ✓ Implements \*\*search functionality\*\* to filter products by name.
- ✓ Uses \*\*pagination\*\* to optimize performance and prevent long product lists.
- ✓ Provides a responsive, user-friendly grid layout.



## Implementation Steps:

### 1. Fetching Products & Categories:

- Calls `client.fetch(allProducts)` to retrieve all products from Sanity CMS.
- Calls `client.fetch(allCategories)` to retrieve available categories.
- Stores the data in state using `useState()`.

### 2. Filtering Products by Category & Attributes:

- Implements a category selector that allows filtering by category.
- Implements filters for discounted products, high/low price sorting, and availability.
- Uses `useState()` to track the selected filters and updates the product list dynamically.

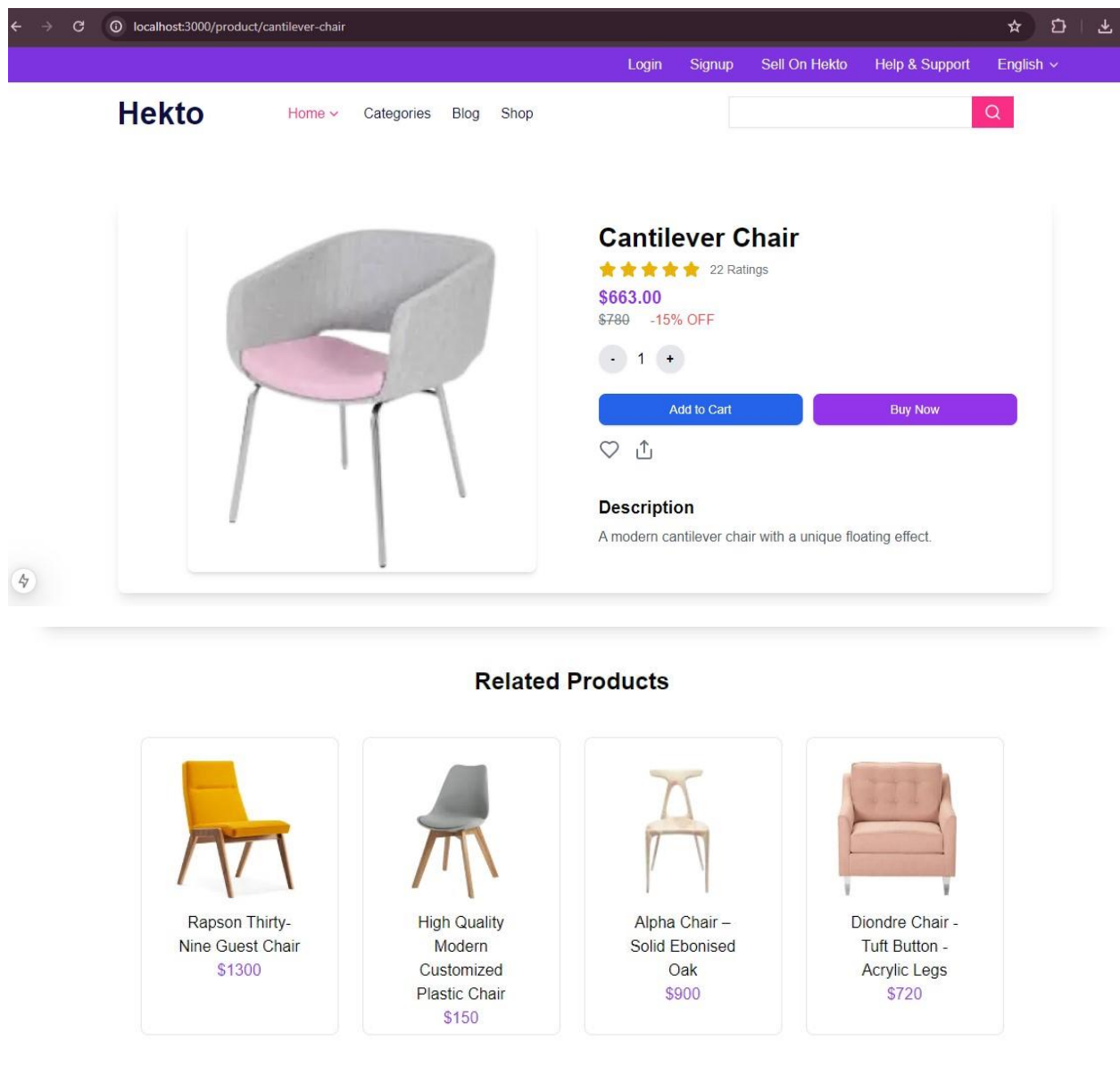
### 3. Pagination:

- Displays only a limited number of products per page (6 per page).
- Implements Previous and Next buttons to navigate between pages.

## 2. Product Detail Page

### Key Features:

- ✓ Fetches individual product details dynamically from Sanity CMS using `slug`.
- ✓ Displays product image, name, price, discount, description, and stock level.
- ✓ Provides Add to Cart and Buy Now functionalities with SweetAlert notifications.
- ✓ Shows Related Products from the same category for enhanced user engagement.



## **Implementation Steps:**

### 1. Fetching Product Data:

- Uses `useParams()` to extract the `slug` from the URL.
- Calls `getProduct(slug)` to fetch product details from Sanity CMS.
- Calls `getRelatedProducts(category, currentProductId)` to fetch related products.

### 2. Displaying Product Details:

- Uses Next.js Image component for optimized image rendering.
- Displays product name, price, discount, and stock information.
- Implements **quantity selection** with `+` and `-` buttons.

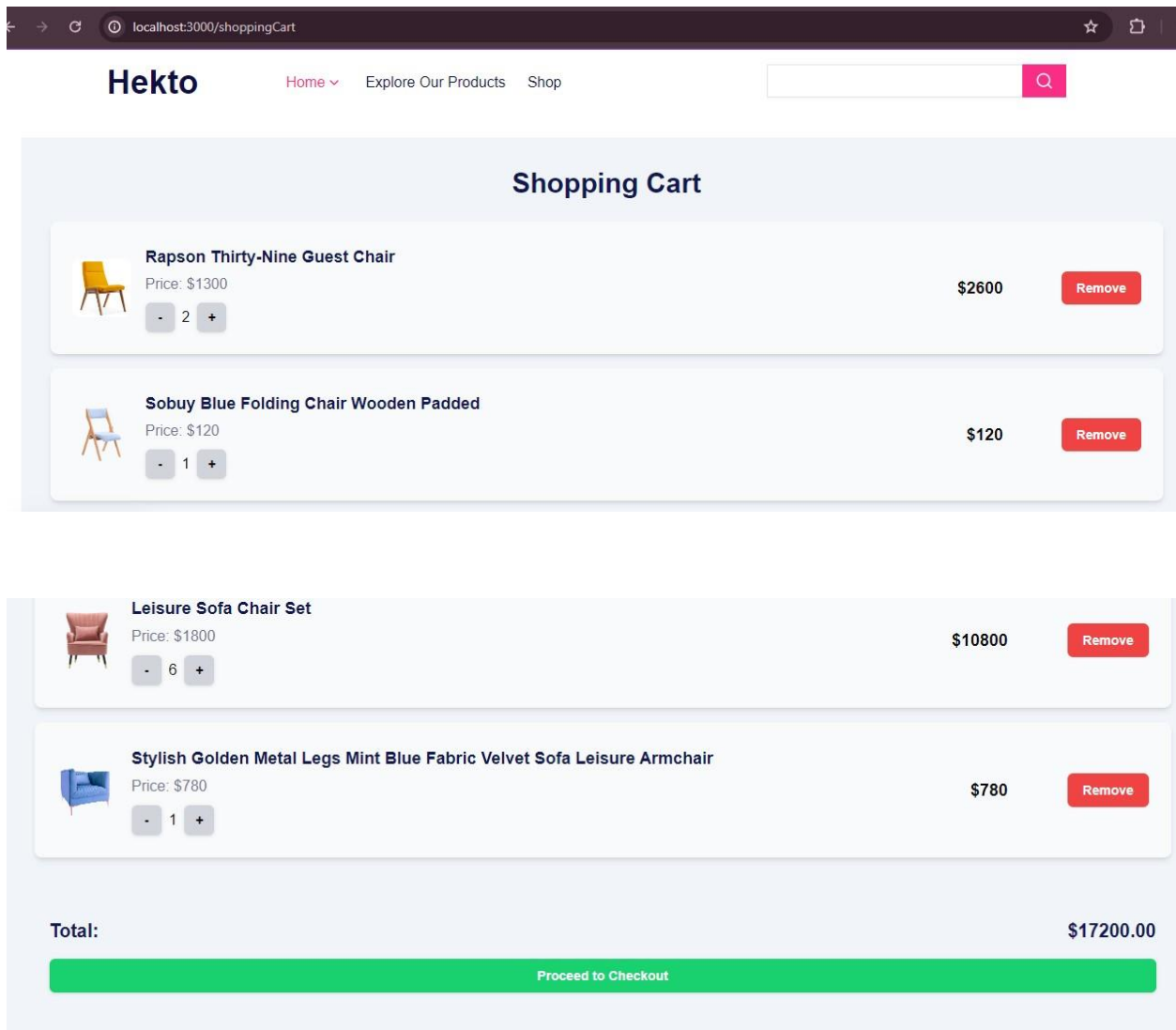
### 3. Related Products Section:

- Fetches and displays up to 4 related products based on the same category.
- Provides clickable links to navigate to individual related product pages.

### 3. Cart Functionality

#### Key Features:

- ✓ Users can add, remove, and update items in the cart.
- ✓ Displays cart total and quantity summary.
- ✓ Persists cart data using local storage.
- ✓ Provides a user-friendly interface for cart management.



## **Implementation Steps:**

### **1. Managing Cart Data:**

- Stores cart items in `localStorage` for persistence.
- Uses `getCartItems()` to retrieve stored cart items.
- Implements `addToCart()`, `removeFromCart()`, and `updateCart()` for cart management.

### **2. Displaying Cart Items:**

- Loops through cart items and shows name, price, quantity, and total.
- Implements `+` and `-` buttons to update item quantity dynamically.
- Uses `SweetAlert2` for user notifications when adding or removing items.

### **3. Proceed to Checkout:**





- Includes a checkout button that confirms the order and redirects users to the checkout page.
- Displays total cart value before proceeding.

## 4. Checkout Page

### Key Features:

- ✓ Collects user details and shipping information.
- ✓ Applies discounts stored in local storage.
- ✓ Validates form input before placing an order.
- ✓ Saves order data to Sanity CMS.
- ✓ Provides real-time order status updates.
- ✓ Protected using Clerk Authentication (User must be logged in to access checkout).

The screenshot displays the checkout page of a web application named 'Hekto'. The browser's address bar shows 'localhost:3000/checkout'. The page features a navigation bar with links to 'Home', 'Explore Our Products', and 'Shop', along with a search bar. The main content area is divided into two columns. The left column, titled 'Order Summary', lists four items: 'Rapson Thirty-Nine Guest Chair' (2 units, \$2600), 'Sobuy Blue Folding Chair Wooden Padded' (1 unit, \$120), 'Nautilus Lounge Chair' (2 units, \$2900), and 'Leisure Sofa Chair Set' (6 units, \$10800). The right column, titled 'Billing Information', contains form fields for 'Name', 'Email', 'Phone', 'Address', 'City', and 'Zip Code'. A blue 'Place Order' button is positioned at the bottom of the right column.

Order Summary	
	Rapson Thirty-Nine Guest Chair Quantity: 2 \$2600
	Sobuy Blue Folding Chair Wooden Padded Quantity: 1 \$120
	Nautilus Lounge Chair Quantity: 2 \$2900
	Leisure Sofa Chair Set Quantity: 6 \$10800

Billing Information	
<b>Name</b> <input type="text" value="Enter your name"/>	<b>Email</b> <input type="text" value="Enter your email"/>
<b>Phone</b> <input type="text" value="Enter your phone"/>	<b>Address</b> <input type="text" value="Enter your address"/>
<b>City</b> <input type="text" value="Enter your city"/>	<b>Zip Code</b> <input type="text" value="Enter your zip code"/>
<button>Place Order</button>	

## **Implementation Steps:**

### **1. Fetching Cart Items & Discounts:**

- Calls ``getCartItems()`` to retrieve cart data.
- Retrieves applied discounts from ``localStorage``.

### **2. Handling User Input & Validation:**

- Uses ``useState()`` to track form values.
- Validates required fields before allowing order placement.
- Uses input styling improvements for a better user experience.

### **3. Placing an Order:**

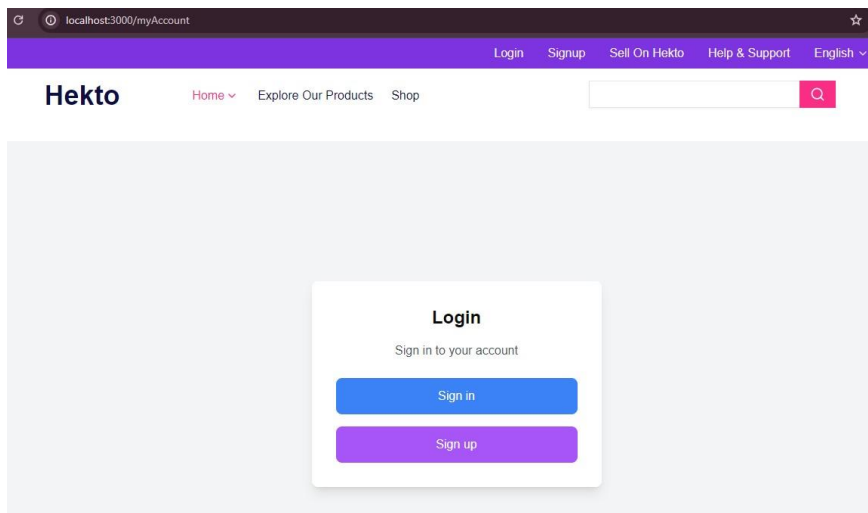
- Saves order data in Sanity CMS using ``client.create()``.
- Uses ``SweetAlert2`` for confirmation dialogs.
- Protects checkout process using AuthGuard & Clerk Middleware.



## 5. Authentication (Login & Signup using Clerk)

### Key Features:

- ✓ Uses **Clerk Authentication** for Google Sign-in.
- ✓ Protects ``Cart`` & ``Checkout`` using middleware (Login required).
- ✓ Redirects users back to ``Cart`` or ``Checkout`` after login.



### Implementation Steps:

1. Setup Clerk Auth in ``middleware.ts``:
  - Ensures ``cart`` & ``checkout`` pages are protected (requires login).
  - Redirects users to login page
2. **Login Page Enhancements:**
  - Uses ``SignedIn``, ``SignedOut``, ``SignInButton``, ``UserButton`` components.
  - Improves UI & ensures a smooth user experience.
3. Profile Page:
  - Provides Logout Button.

## 6. Dashboard & Order Tracking

### Key Features:

- ✓ Manage Orders: View, update, and delete orders.
- ✓ Filter Orders: Sort by Pending, Processing, Shipped, and Delivered.
- ✓ Update Order Status: Modify order status dynamically.
- ✓ Protected Access: Only authorized admin users can access the panel.



Admin Dashboard

Orders

Order ID	Customer	Email	Phone	Address	Order Date	Total	Status	Action
Mn3RZG5zPAWOop6Yg36GUL	Aanmol	aanmol187@gmail.com	03029056088	Karachi, Pakistan	2/11/2025	\$1800	Pending	Delete

### Implementation Steps:

#### 1. Fetching Orders:

- Calls `client.fetch(ordersQuery)` to retrieve all orders from Sanity CMS.
- Filters orders dynamically based on their status.

#### 2. Order Management:

- Allows **updating order status** (Pending → Processing → Shipped → Delivered).
- Implements **order deletion** with confirmation dialogs.

#### 3. Access Control:

- Uses Clerk authentication to verify admin credentials.
- Only authorized users (e.g., store owner) can modify orders.

## **7. Technologies Used**

- Next.js – Handles routing and server-side rendering.
- Sanity CMS – Stores and manages product data.
- Tailwind CSS – Ensures a fully responsive design.
- SweetAlert2 – Displays user-friendly notifications.
- GROQ Queries – Fetches optimized product data from Sanity CMS.

---

## **8. Future Enhancements**

- Wishlist Component – Allow users to save favorite products for later.
- Order Cancellation & Returns – Implement request-based returns and refunds.
- Real-Time Chat with Sellers\*\* – Enable direct communication between buyers and sellers.
- Profile Page Improvements– Enhance user profile with additional customization.
- Admin Dashboard Enhancements – Provide analytics and order management tools.
- Advanced Order Tracking – Implement live tracking with estimated delivery time.
- Seller Registration – Allow sellers to register and manage their products.
- Payment Methods Integration – Support multiple payment gateways for better user convenience.
- Customization Option – Enable users to request custom modifications for products.

---