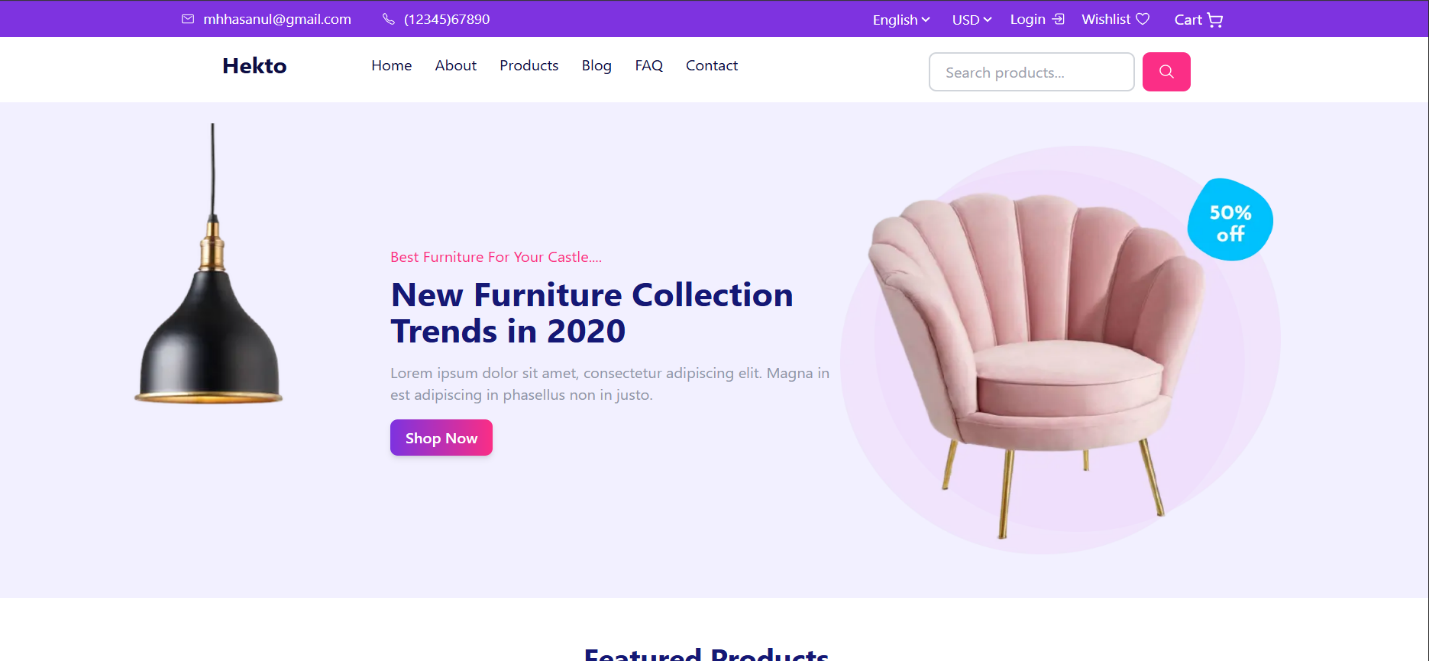
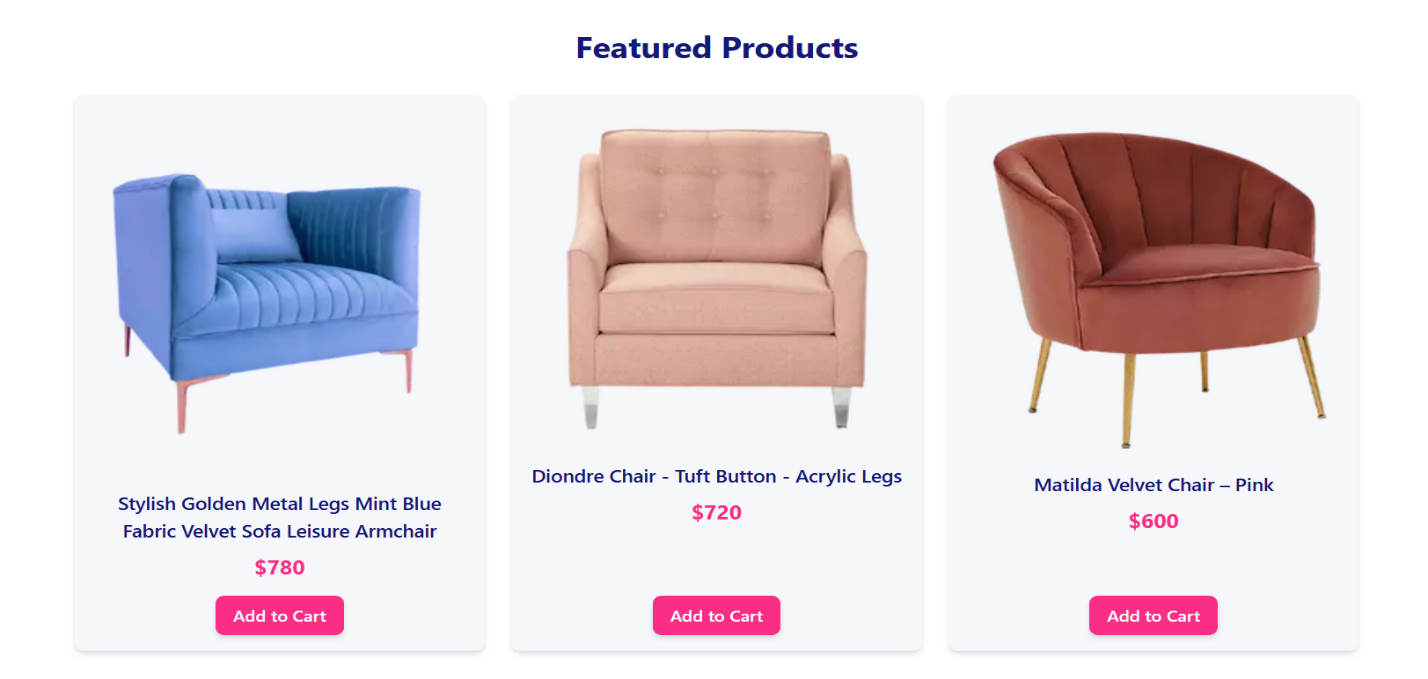
**Day 4 - Building Dynamic Frontend Components for Your Marketplace:**

**[General E-commerce]**

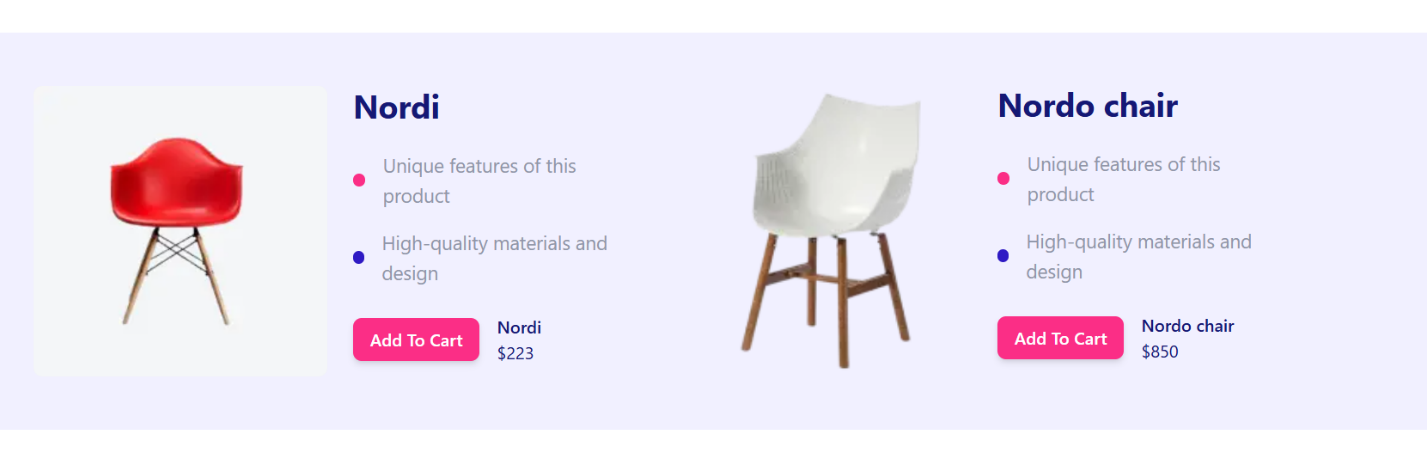
**Homepage:**

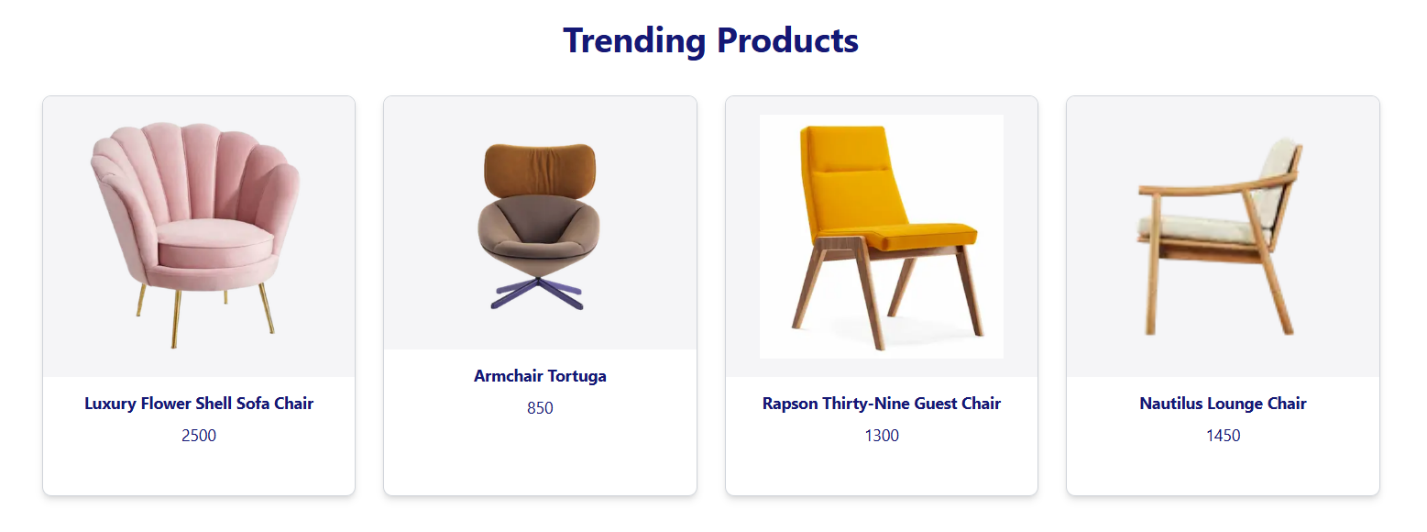
****

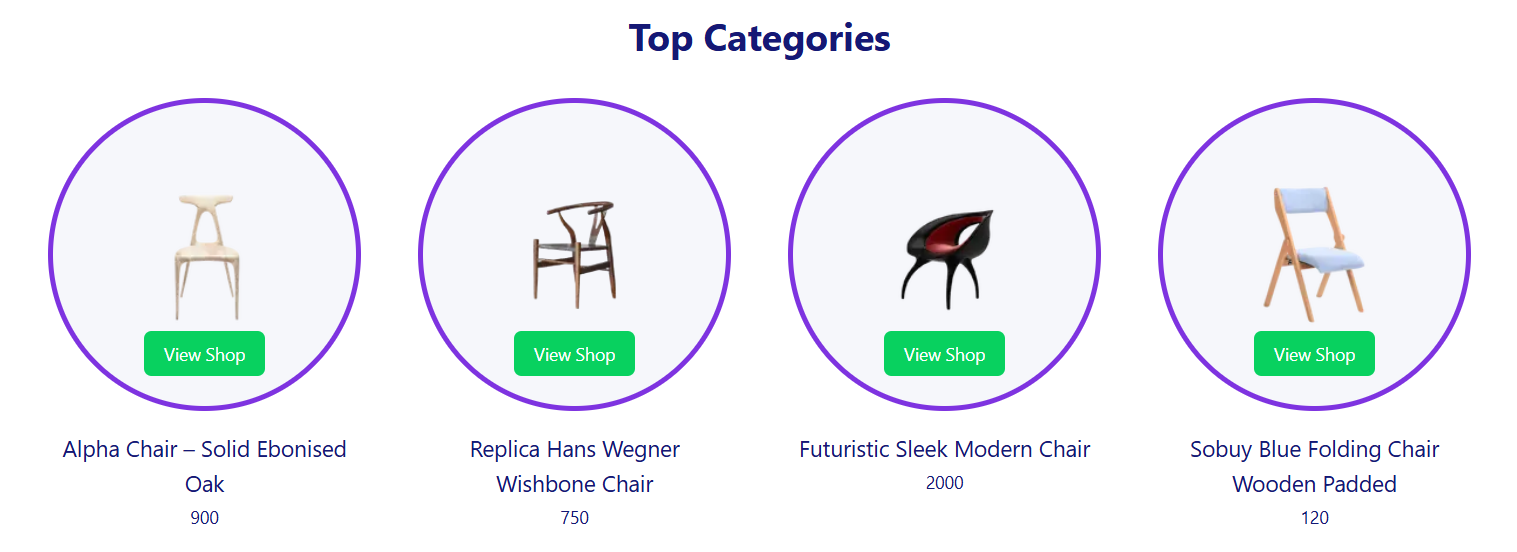
**Product Page:**

****

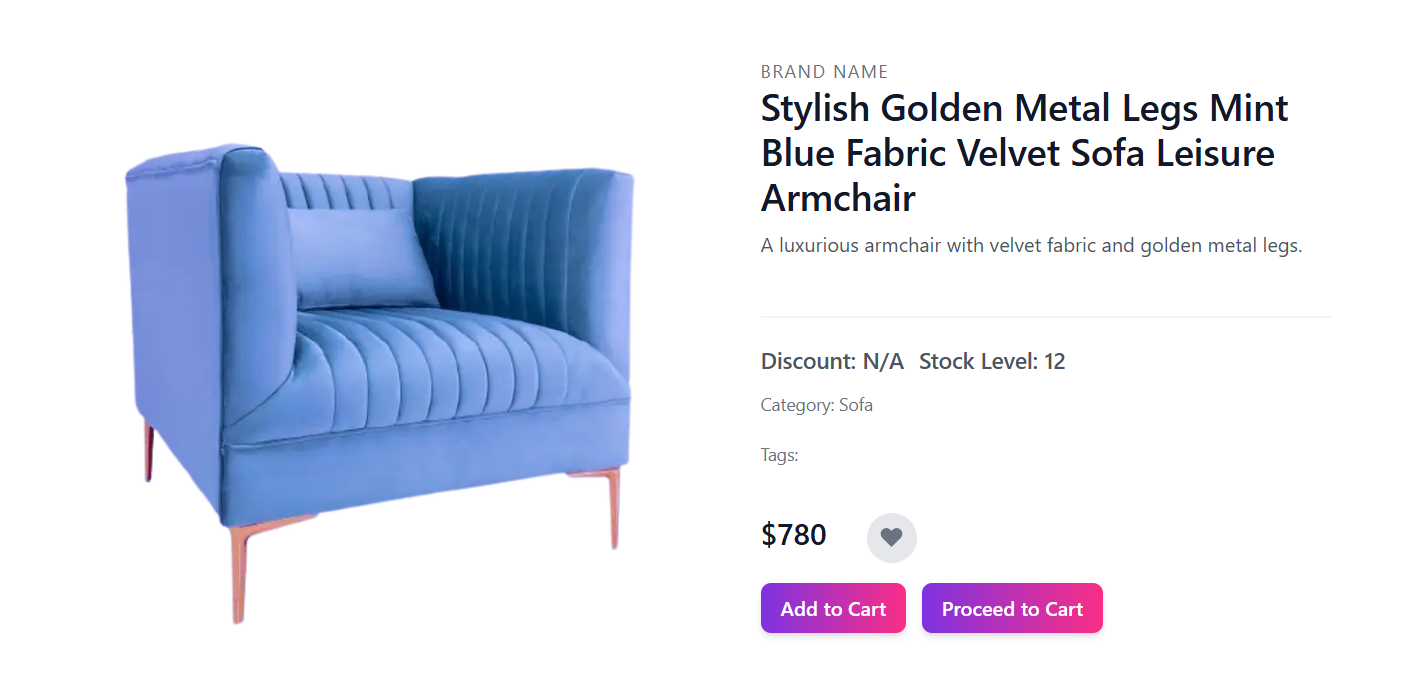
**Different Components of Products:**

****

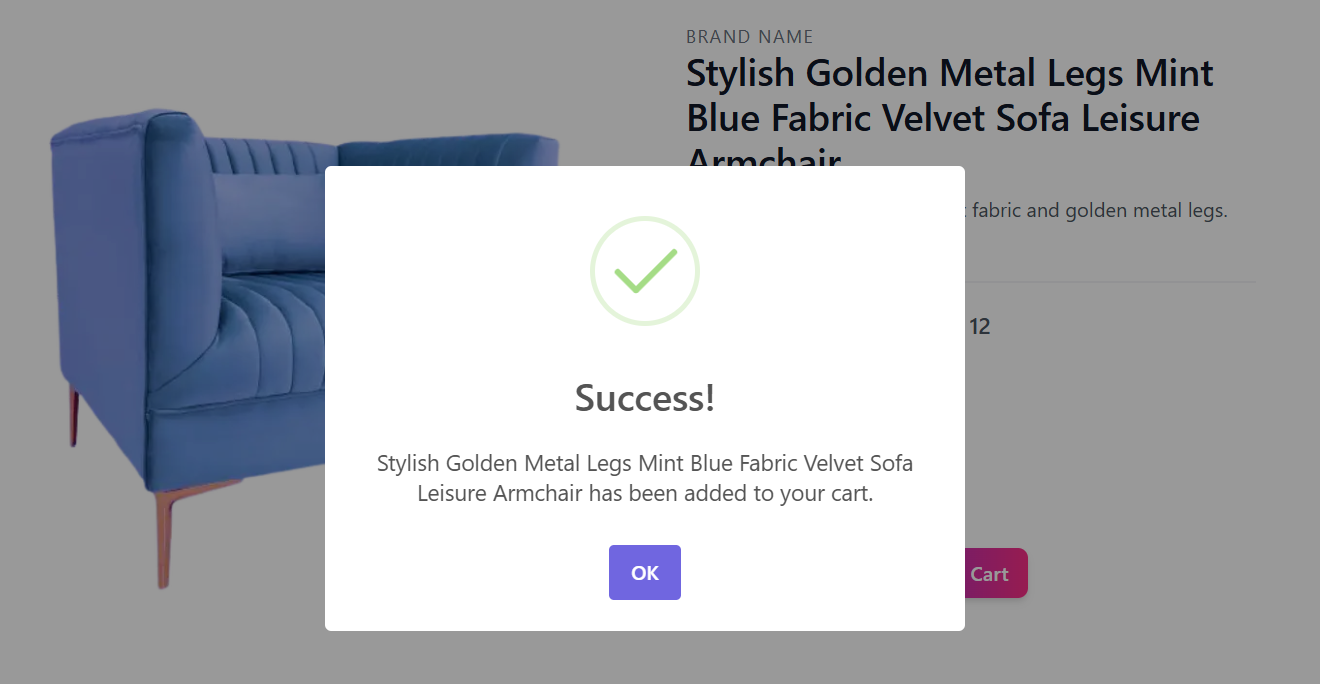
****

****

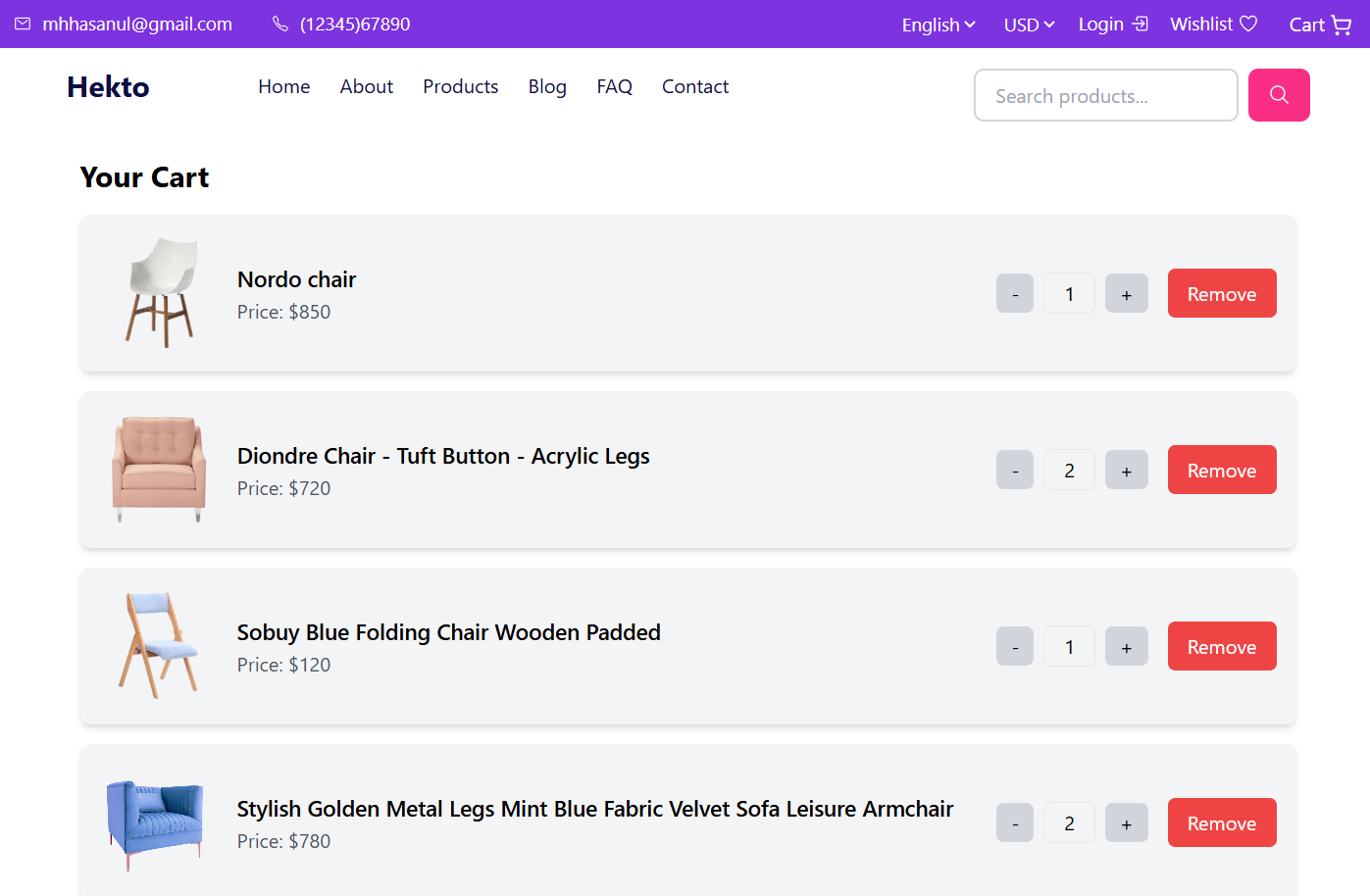
**Product Details:**

****

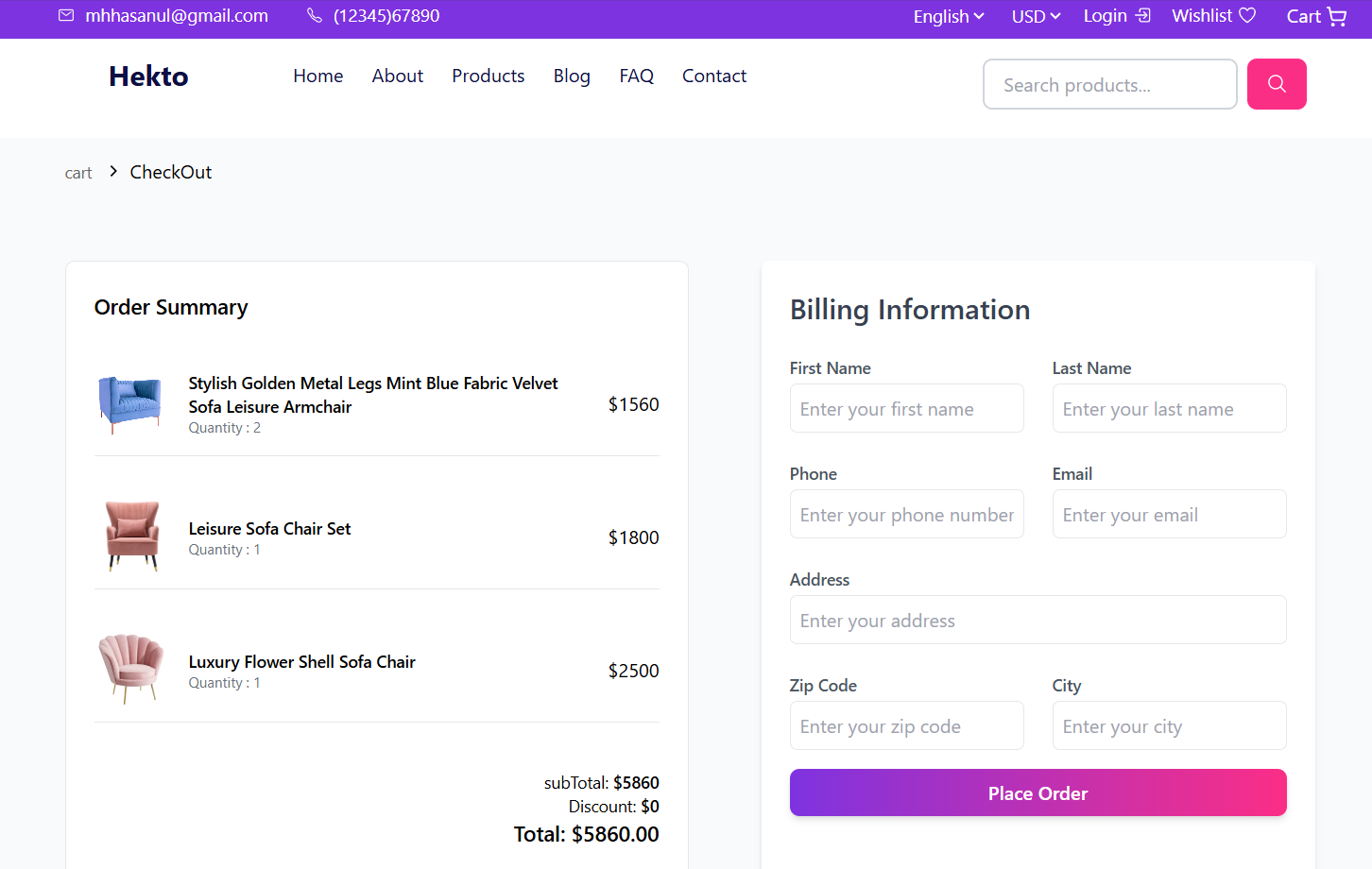
**On clicking Add to Cart Functionality:**

****

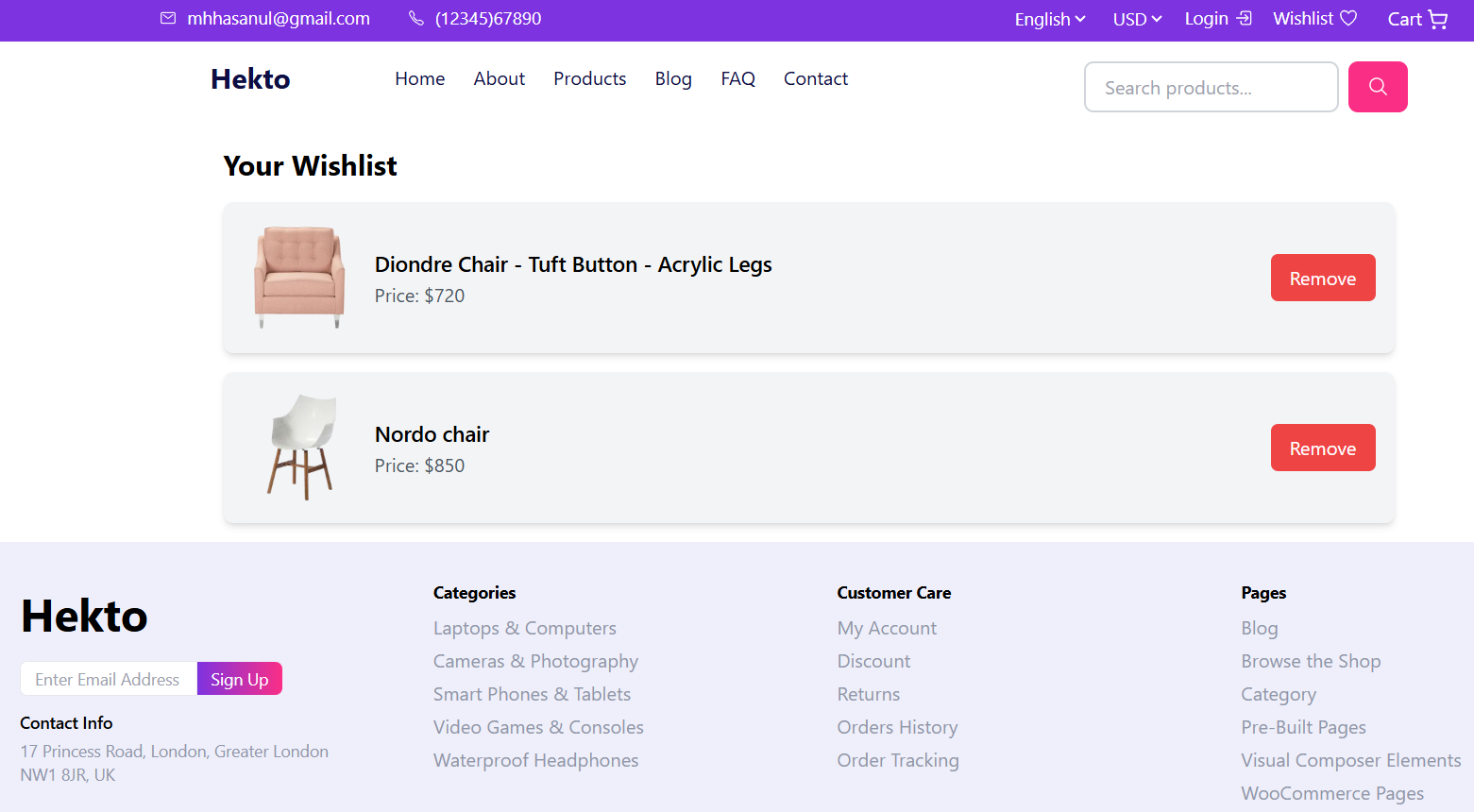
**Proceed to Add to Cart page Dynamic Routing:**

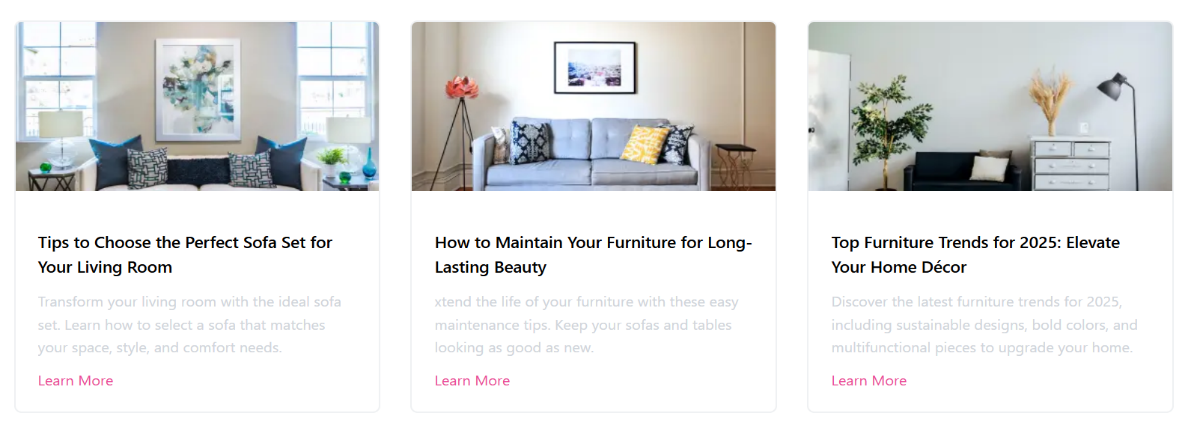
****

**Proceed to Checkout:**

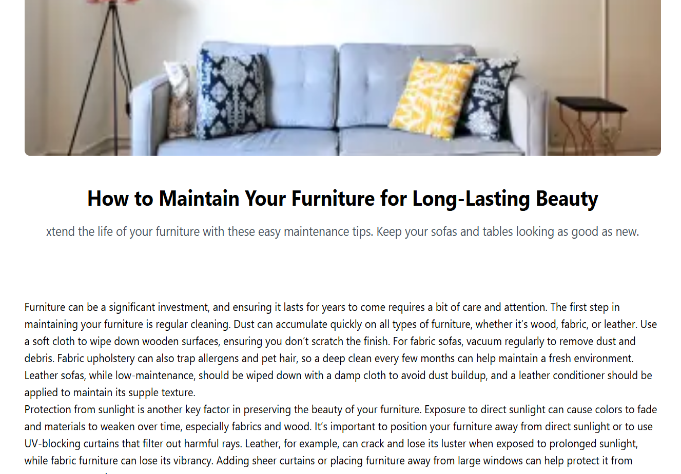
****

**Wishlist Page:**

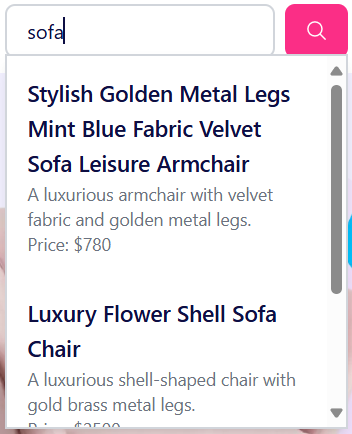
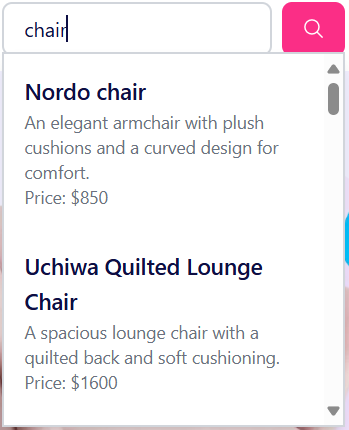
****

**Blogs:  
**

**Blogs Dynamic Routing Page:**

** **

**Searchbar Funtionality:**

** **

**Project Overview**

This phase focuses on developing dynamic and reusable frontend components for an interactive e-commerce experience. The implementation ensures efficiency, responsiveness, and seamless integration with Sanity CMS while maintaining a structured and modular codebase.

**Core Components & Implementation**

**1. Product Card Component**

* **Reusable Design:** Displays individual product information in a visually appealing card.
* **Elements Included:**
  + Product image, name, and price.
  + Wishlist toggle button.
  + Clickable link to navigate to the detailed product page.
* **Wishlist Functionality:** Uses Local Storage to store user preferences.
* **Performance Optimizations:** Implements optimized image rendering.

**2. Product List Component**

* **Two-Part Implementation for Performance:**
  + **Server-side Fetching:** Retrieves product data from Sanity CMS using GROQ queries.
  + **Client-side Rendering:** Handles filtering, display updates, and user interactions.
* **Grid Layout:** Products are displayed in a responsive grid format.

**API Integration & Routing**

**1. Sanity CMS Integration**

* **Data Management:**
  + Fetches product details such as images, prices, and categories.
  + Uses GROQ queries for optimized data retrieval.
* **Image Optimization:** Leverages Sanity's CDN for fast and responsive image loading.

**2. Dynamic Routing System**

* **SEO-Friendly URLs:** Unique URLs generated using product slugs.
* **Error Handling:** Implements fallback mechanisms for missing or non-existent product pages.

**3. Data Flow**

* **Server Components:** Fetch and serve initial product data.
* **Client Components:** Handle user interactions and dynamic updates.
* **URL Parameters:** Maintain the filter state across page reloads and navigation.
* **Local Storage:** Manages user preferences for wishlists.

**Best Practices Followed**

**1. Code Organization**

* **Component-Based Architecture:** Modular approach for better scalability.
* **Type Safety:** Implemented TypeScript for defining strict types.
* **Separation of Concerns:** Clear distinction between API calls, UI components, and business logic.

**2. Performance Optimization**

* **Next.js Image Optimization:** Improves page load speed with automatic resizing.
* **Server-Side Rendering (SSR):** Enhances SEO and performance.

**3. Security Measures**

* **Environment Variables:** Secures sensitive API credentials.
* **API Route Protection:** Restricts unauthorized access.

**Challenges & Solutions**

**1. Handling Dynamic Routing with Sanity**

* **Problem:** Displaying product pages dynamically based on data from Sanity CMS.
* **Solution:** Implemented dynamic routing in Next.js to generate product pages using slug-based URLs, ensuring each product has a unique and SEO-friendly link.

**2. Optimizing Real-Time Search**

* **Problem:** Implementing an efficient product search functionality.
* **Solution:** Integrated a client-side search bar that filters product data in real time, updating displayed products dynamically as the user types.

**3. Ensuring Responsive Design**

* **Problem:** Maintaining a seamless experience across various devices.
* **Solution:** Utilized Tailwind CSS for adaptive layouts and breakpoints.

**Additional Features Implemented**

1. **Related Products Section** – Displays similar products dynamically.
2. **User Wishlist Management** – Allows users to save and manage favorite products.
3. **Dynamic Category-Based Navigation** – Enables users to explore different product types efficiently.
4. **Real-Time Search Functionality** – Enhances product discovery.
5. **Fully Responsive Design** – Ensures a smooth user experience across devices.

**Summary**

This phase successfully established the foundation for dynamic, scalable, and performance-optimized frontend components in the e-commerce project. By integrating Next.js server and client-side rendering, Sanity CMS for content management, and Tailwind CSS for styling, the platform ensures an engaging and seamless shopping experience for users.