Day 4 - Dynamic Frontend Components - EcoFurnish

Prepared by: *Zija Yaseen*

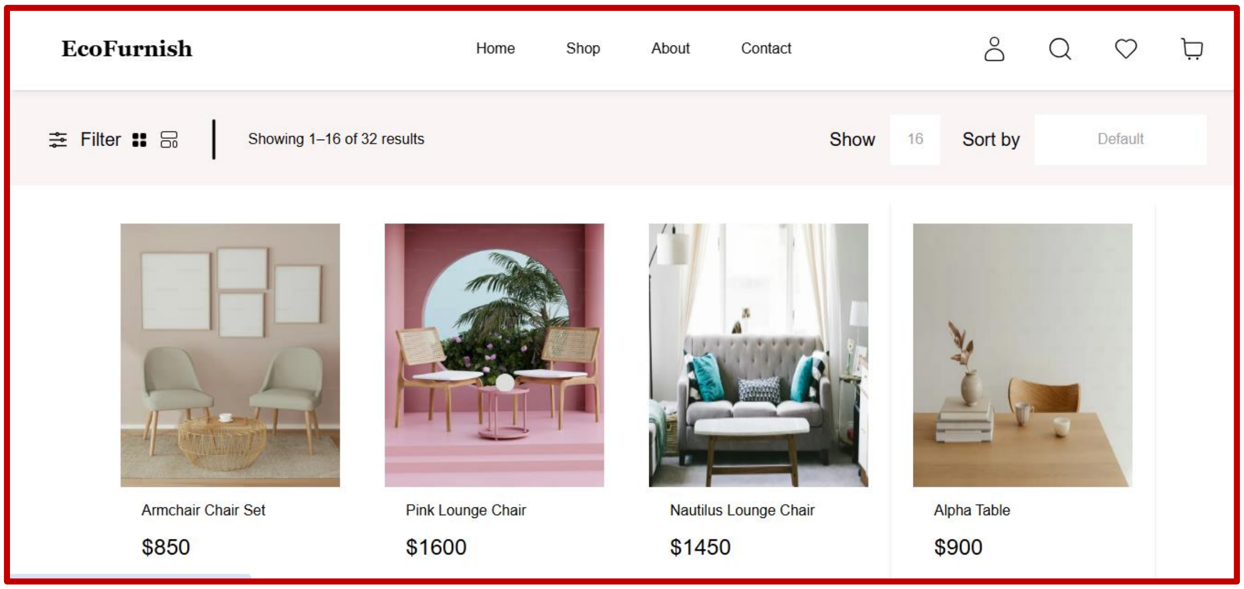
---

1. Functional Deliverables

Video Demonstration:
To see these features in action, watch the video demonstration:
[Watch Video]

Screenshots:
Below are the screenshots showcasing the implemented features:

- **Product Listing Page:** Displaying dynamically fetched product data.



Showing 1–16 of 32 results
Show 16

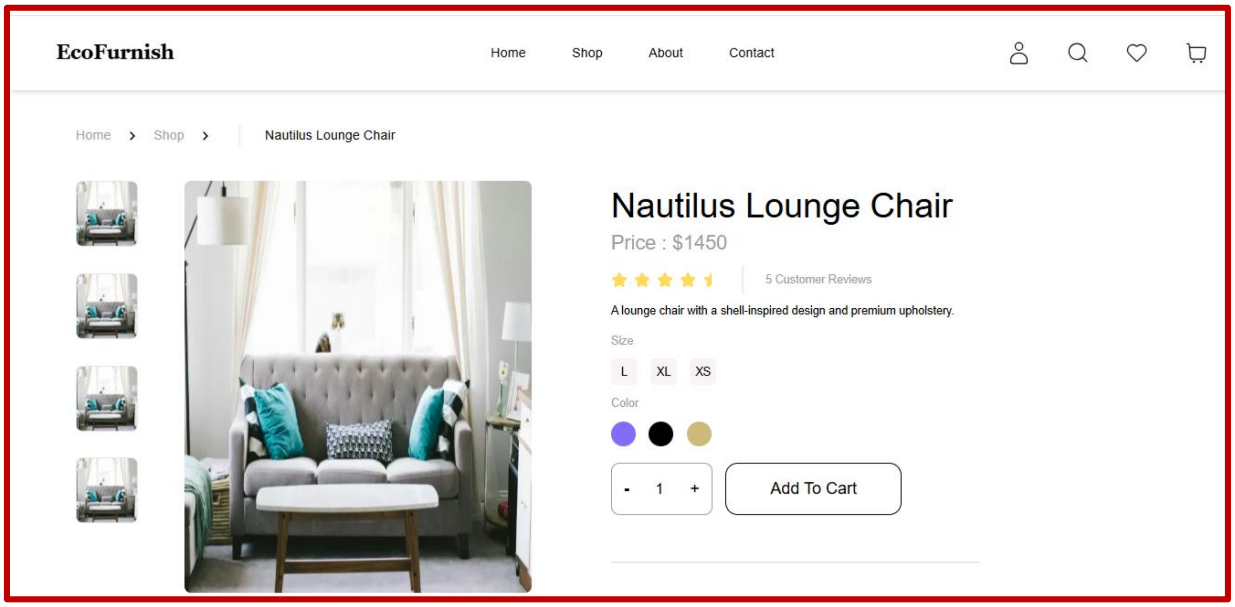
Armchair Chair Set
$850

Pink Lounge Chair
$1600

Nautilus Lounge Chair
$1450

Alpha Table
$900

- **Individual Product Detail Pages:** Proper routing and data rendering for selected products.



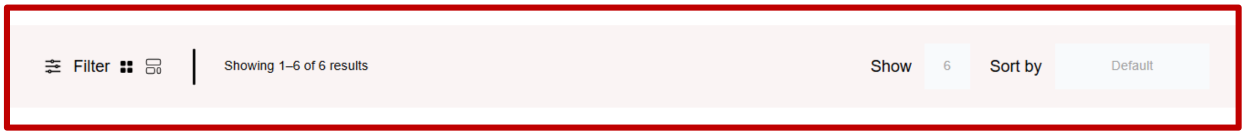
Nautilus Lounge Chair

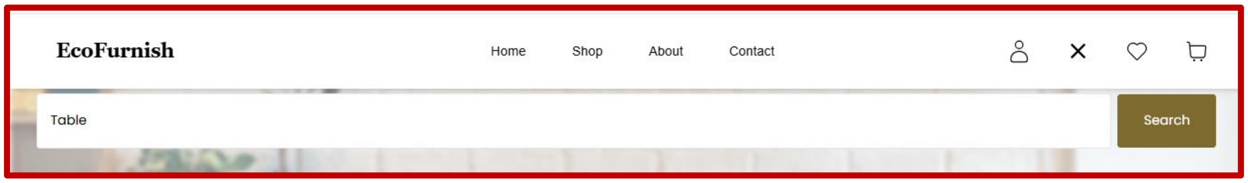
Price : $1450

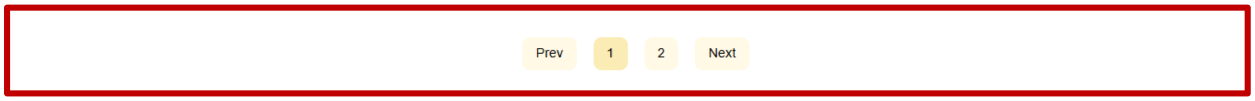
5 Customer Reviews

A lounge chair with a shell-inspired design and premium upholstery.

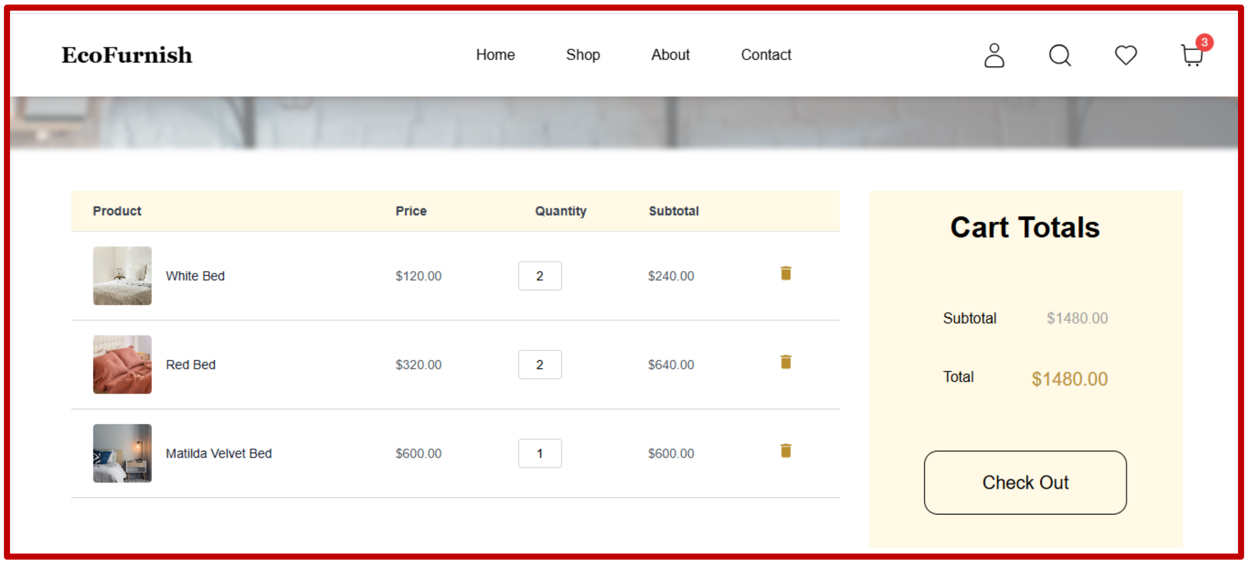
- **Category Filters, Search Bar, and Pagination:** Demonstrating functionality.







- **Cart Page:** Displaying items added to the cart dynamically with data persisted in **Sanity CMS**.



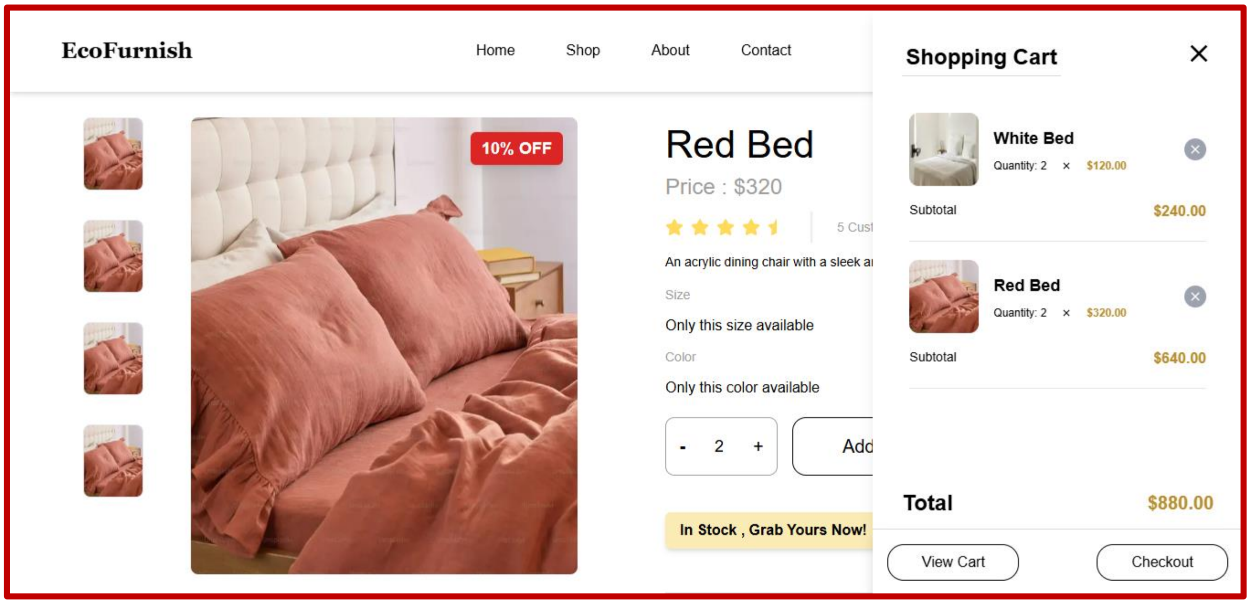
Product	Price	Quantity	Subtotal
White Bed	$120.00	2	$240.00
Red Bed	$320.00	2	$640.00
Matilda Velvet Bed	$600.00	1	$600.00

Cart Totals

Subtotal $1480.00

Total $1480.00

- **Sidebar Cart :** Displaying items added to the cart dynamically with data persisted in **Sanity CMS**.



Red Bed

Price : $320

5 Cust

An acrylic dining chair with a sleek a

Size

Only this size available

Color

Only this color available

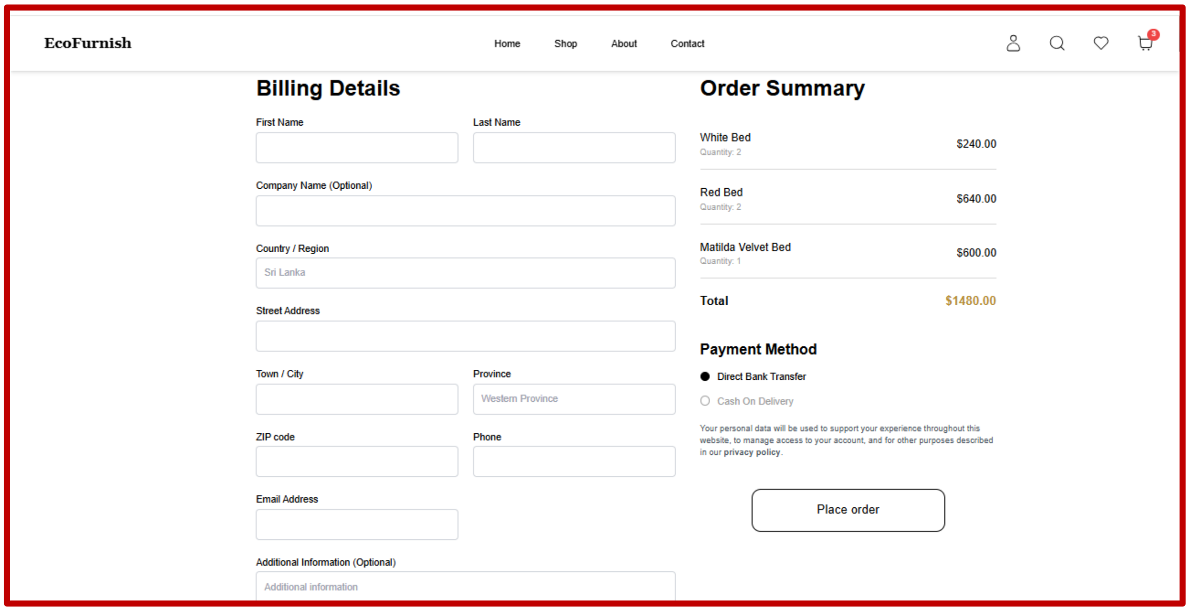
Shopping Cart

White Bed
Quantity: 2 × $120.00
Subtotal $240.00

Red Bed
Quantity: 2 × $320.00
Subtotal $640.00

Total $880.00

- **Checkout Page:** Reviewing items and processing orders with checkout data saved in **Sanity** to retain information after page refreshes.



EcoFurnish

Home Shop About Contact

Billing Details

First Name

Last Name

Company Name (Optional)

Country / Region

Sri Lanka

Street Address

Town / City

Province

Western Province

ZIP code

Phone

Email Address

Additional Information (Optional)

Additional information

Order Summary

White Bed Quantity: 2	$240.00
Red Bed Quantity: 2	$640.00
Matilda Velvet Bed Quantity: 1	$600.00
Total	$1480.00

Payment Method

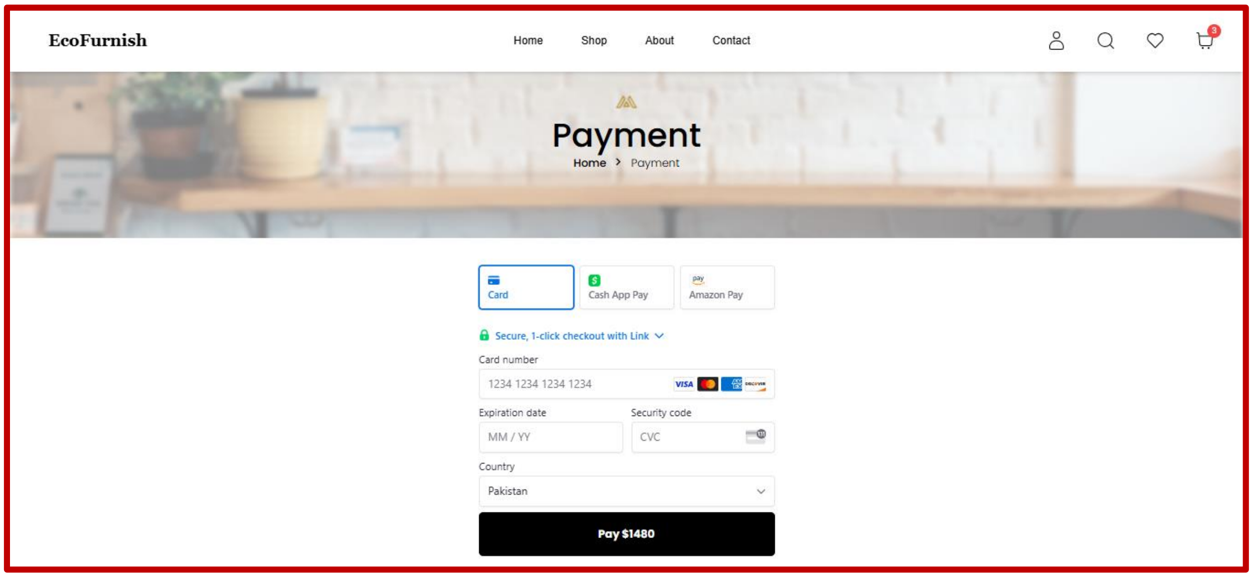
● Direct Bank Transfer

○ Cash On Delivery

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.

Place order

- **Payment Page:** Payment processing integrated with the checkout flow.



EcoFurnish

Home Shop About Contact

Payment

Home > Payment

Card | Cash App Pay | Amazon Pay

Secure, 1-click checkout with Link

Card number

1234 1234 1234 1234

Expiration date

MM / YY

Security code

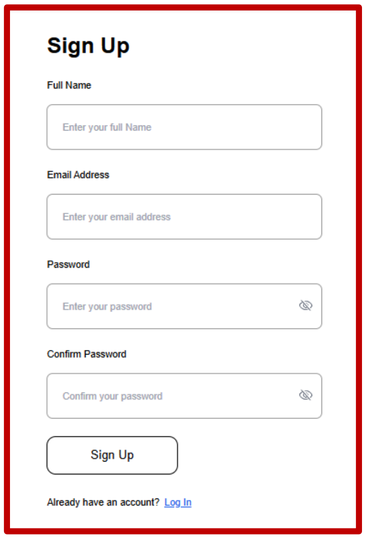
CVC

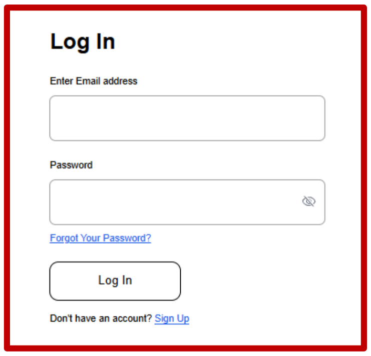
Country

Pakistan

Pay $1480

- **Account Page:** User profile, order history, and account settings.





---

2. Code Deliverables

Here are key component code snippets from the project:

GitHub Repo

- **ProductList.tsx:** Displays the list of products fetched from Sanity CMS.
- **Pagination.tsx:** Manages pagination logic and UI.
- **SearchBar.tsx:** Enables search functionality to filter products.
- **CartPage.tsx:** Displays products added to the cart and syncs with **Sanity CMS** for persistent cart data, even after page refresh.
- **CheckoutPage.tsx:** Handles order review and submits checkout data to **Sanity** to simulate real-world order persistence.
- **Sidebar.tsx:** Provides navigation links for the application.
- **PaymentPage.tsx:** Integrates payment options for completing purchases.
- **AccountPage.tsx:** Displays user information, order history, and allows profile updates.

---

3. Scripts and Logic for API Integration & Dynamic Routing

Global State Management (Redux) + Sanity Integration:

- **Redux Toolkit** is used for fast, efficient global state management.
- **productSlice** stores fetched product data for fast rendering.
- **cartSlice** manages cart state locally and syncs with **Sanity CMS** to persist cart data across sessions.
- **checkoutSlice** handles checkout information, storing it in **Redux** for quick UI updates and in **Sanity** for persistent storage.
- **authSlice** manages user authentication state.
- **createAsyncThunk** is used for asynchronous API calls, making data fetching and state updates seamless.

Cart & Checkout Data Persistence (Sanity + Redux):

- Cart and checkout data are **saved in Sanity CMS** to ensure persistence even after page reloads or browser refreshes.
- **Redux** is used for real-time updates and faster page load times, while **Sanity** ensures that data is always backed up and retrievable, simulating a real-world e-commerce setup.
- When users add or remove items from the cart:
    - The **Redux state** updates instantly for a fast, responsive UI.
    - The cart data is also **pushed to Sanity**, ensuring it's saved in the backend.
    - On page refresh, cart data is **fetched from Sanity** and synced with Redux for consistency.
- During checkout:
    - Order details are saved both in **Redux** (for immediate feedback) and **Sanity** (to simulate order processing and tracking).
    - This ensures that users can refresh the page or revisit later, and their order history remains intact.

---

4. Documentation

Steps Taken:

1. **Designed Components:** Created reusable components like `ProductCard`, `ProductList`, `SearchBar`, and `Sidebar`.
2. **Implemented Routing:** Used Next.js dynamic routes for individual product pages and account-related pages.
3. **Integrated API:** Connected frontend to **Sanity CMS** for fetching product, user, cart, and order data.
4. **Added Filters & Pagination:** Implemented search, category filters, and pagination for better UX.
5. **Built Cart & Checkout:** Developed a cart system with checkout functionality, syncing data between **Redux** and **Sanity** for both speed and persistence.
6. **User Authentication:** Created login and sign-up pages, storing user data in **Sanity** for secure, persistent account management.

Challenges & Solutions:

- **Challenge:** Ensuring cart data persists after page refresh (real-world scenario).
  Solution: Synced cart data between **Redux** (for fast UI) and **Sanity CMS** (for persistent storage).
- **Challenge:** Handling asynchronous data fetching and state updates without performance issues.
  Solution: Used **Redux Toolkit's createAsyncThunk** for efficient API handling and **lazy loading** techniques for optimization.
- **Challenge:** Managing multiple data sources (Sanity + Redux) without data conflicts.
  Solution: Implemented a robust synchronization strategy to keep local state (Redux) and backend data (Sanity) consistent.
- **Challenge:** Building a responsive UI that adapts to different devices.
  Solution: Leveraged **Tailwind CSS** for responsive design and mobile-first development.

Best Practices Followed:

- **Data Persistence:** Used **Sanity CMS** to ensure cart, checkout, and user data remain intact even after page refreshes, mimicking real-world e-commerce functionality.
- **Fast UI Updates:** Utilized **Redux** for quick state management, ensuring fast rendering and a responsive user experience.
- **Component Reusability:** Modular, reusable components for efficient development and maintenance.
- **State Management:** Efficiently combined `useState`, `useEffect`, and **Redux Toolkit** for seamless data flow.
- **Error Handling:** Implemented comprehensive error handling for API requests and user input validation.
- **Performance Optimization:** Optimized rendering using lazy loading, efficient state management, and memoization techniques.
- **Secure Data Handling:** Stored sensitive data like user credentials securely in **Sanity**, with proper authentication mechanisms in place.

---

5. Submission Format

- **Document Title:** *Day 4 - Dynamic Frontend Components - EcoFurnish*
- **Format:** PDF
- **Contents:**
    1. Screenshots/Recordings
    2. Code Snippets
    3. Technical Documentation
