

**Hackathon 3: Day\_4**

# **Dynamic Frontend Components**

---

## **Table of Contents**

1. Introduction
2. Development Approach
3. Implemented Components
4. Code Snippets
5. Screenshots
6. Challenges and Solutions
7. Conclusion

# 1. Introduction

## 1.1 Project Overview

The Comforty Furniture website includes dynamic frontend components to enhance user interaction and improve functionality. These components were built using Next.js and optimized for performance and responsiveness.

## 1.2 Objectives

- Develop reusable, scalable, and efficient frontend components.
  - Ensure a seamless and engaging user experience.
  - Implement interactive elements that adapt dynamically.
- 

# 2. Development Approach

## 2.1 Technologies Used

- **Next.js** – Framework for building the frontend.
- **Tailwind CSS** – For styling and responsiveness.
- **ShadCN UI** – For modern UI components.
- **Material UI** – For modern UI components.
- **Redux Toolkit** – For managing state, especially in Add to Cart functionality.

## 2.2 Best Practices Followed

- Modular component structure.
- Performance optimization techniques.
- Accessibility considerations.

## 3. Implemented Components

- **Product Card** – Displays product information dynamically, Customized for product listing and Featured, Category Sections.
  - **Product Details Card** – Displays product information dynamically for each product and individually with their respective details such as; Product Title, Description, Price, Inventory (quantity), Cart button.
  - **Search Bar** – Implements real-time search functionality.
  - **Filters** – Allows users to refine product selection listed below:
    - **Category** – Allows users to browse by category
    - **Sort By** – Allows users to browse by customized sorting as:
      - Price (Low to High) & Price (High to Low)
    - **All Prices** – Allows users to browse products according to their prices.
  - **Add to Cart** – Implements cart functionality using Redux Toolkit.
- 

## 4. Code Snippets

**Product Card (Listing Page):**

```

<h1 className="text-2xl font-bold mb-6 text-center sm:text-left">All Products</h1>

<div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">

  {sortedProducts.length > 0 ? (
    sortedProducts.map(product => {
      <div key={product._id} className="relative bg-white p-4">
        <Link href={` /products/${product._id}`}>
          <div className="w-full h-72 relative mb-4 transition-transform transform hover:scale-105 duration-500">
            <Image
              src={urlFor(product.image.url())}
              alt={product.title}
              layout="fill"
              objectFit="contain"
              className="rounded-none rounded-t-lg"
            />
          </div>
        </Link>

        <h3 className="text-lg font-semibold">{product.title}</h3>

        <div className="flex justify-between items-center mt-2">

          <p className="text-gray-500 text-lg cursor-pointer">${product.price}</p>

          <Link href={` /products/${product._id}`} className="flex items-center space-x-1 text-gray-500 hover:text-gray-700 transition-colors">

            <span>View Details</span>
            <FiArrowRight className="text-xl" />
          </Link>

        </div>

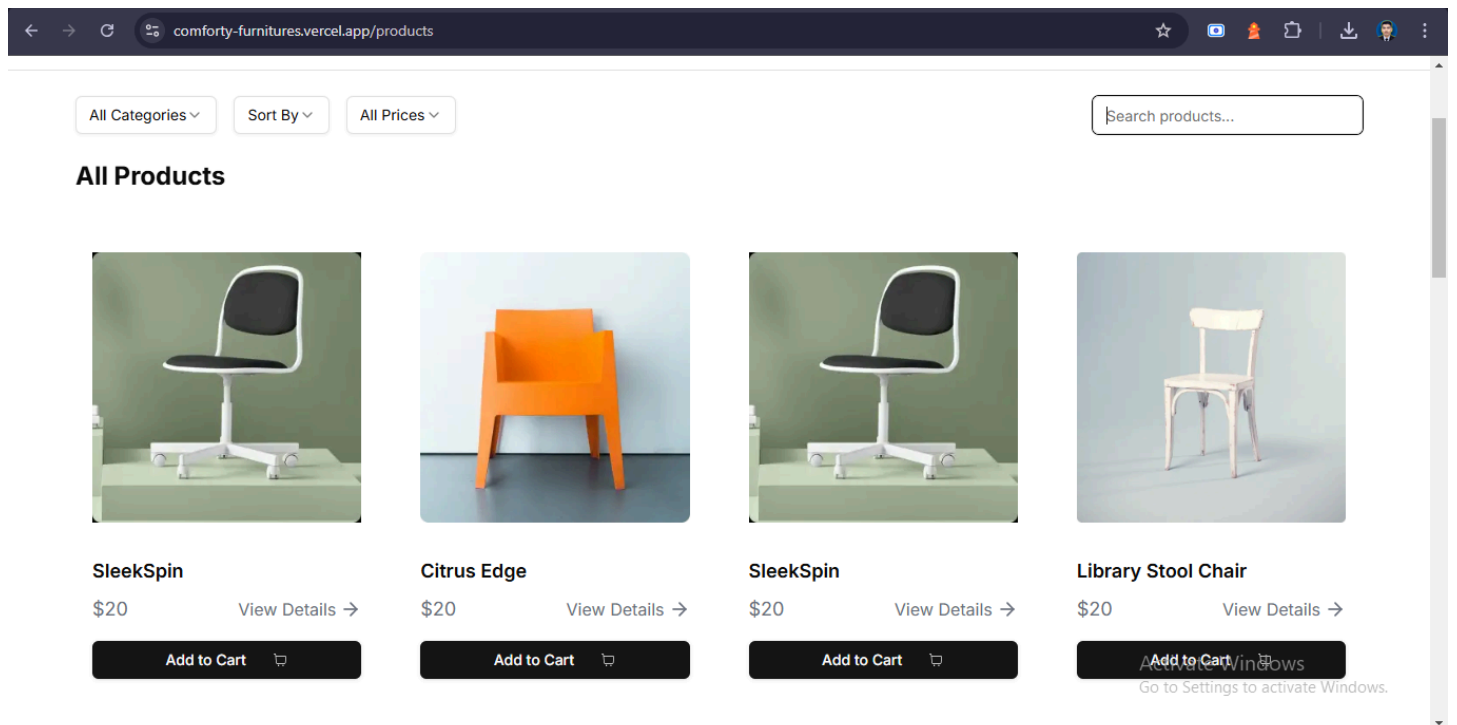
        <div className="flex justify-between items-center mt-4">

          <Button
            variant="default"
            className="w-full text-center"
            onClick={() => handleAddToCart(product)}
          >
            Add to Cart
            <CShoppingCart className="ml-4" />
          </Button>

        </div>
      </div>
    ) : (
      <p>No products found.</p>
    )
  )}
</div>

```

## Screenshot:



## Product Card (Detail Page):

```
<div className="flex flex-col md:flex-row items-center justify-center px-6 py-10 bg-white">
  /* Left: Product Image */
  <div className="w-full md:w-1/2 flex justify-center mb-6 md:mb-0">
    <Image
      src={urlFor(product.image).url()}
      alt={product.title}
      width={400}
      height={400}
      className="rounded-none object-contain"
    />
  </div>

  /* Right: Product Details */
  <div className="w-full md:w-1/2 max-w-md">
    <h1 className="text-3xl md:text-4xl font-semibold text-gray-900 mb-4">
      {product.title}
    </h1>

    <div className="flex items-center gap-3 mb-4">
      <span className="text-xl font-semibold text-white bg-cyan-600 rounded-full py-1 px-3">
        ${product.price} USD
      </span>
      {product.priceWithoutDiscount && (
        <span className="text-md text-gray-500 line-through ml-3">
          ${product.priceWithoutDiscount}
        </span>
      )}
    </div>

    <hr className="border-gray-300 my-4" />

    <p className="text-gray-700 text-sm mb-6">{product.description}</p>

    <p className={`text-base mb-6 font-semibold ${product.inventory === 0 ? "text-red-600" : product.inventory < 5 ? "text-red-600" : "text-green-600"}`}>
      {product.inventory === 0 ? "Out of Stock" : `In stock: ${product.inventory}`}
    </p>

    /* Quantity Section */
    <div className="flex items-center gap-4 mb-6">
      <p className="text-gray-700 text-lg font-semibold">Quantity:</p>

      /* Decrement Button */
      <button
        onClick={decreaseQuantity}
        disabled={quantity === 1} // Disable minus button if quantity is 1
        className={` ${quantity === 1
          ? "bg-gray-400 cursor-not-allowed"
          : "bg-cyan-600 hover:bg-cyan-700"
        } text-white text-lg px-4 py-2 rounded-3xl`}
      >
        -
      </button>
      /* Display Quantity */
      <span className="text-gray-700 text-lg font-semibold">{quantity}</span>
      /* Increment Button */
      <button
        onClick={increaseQuantity}
        className="bg-cyan-600 text-white text-lg px-4 py-2 rounded-3xl hover:bg-cyan-700"
      >
        +
      </button>
    </div>

    /* Add to Cart Button */
    <button
      onClick={handleAddToCart}
      className="flex items-center bg-cyan-600 text-white px-6 py-3 rounded-md hover:bg-cyan-700 transition-all"
    >
      <ShoppingCart className="text-white text-2xl mr-2" /> Add to Cart
    </button>
  </div>
</div>
```

## Screenshot:

← → ↺


comforty-furnitures.vercel.app/products/znHQMgVDeUWHvH68ZXBqWb

☆

📺 🔔 📄 📥 👤 ⋮

✓ Free Shipping On All Orders Over \$50


More ▾

 Comforty

🛒 Cart

Home Products Category About Contact

Contact: (808) 555-0111



Ivory Charm

\$20 USD

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

In stock: 20

Quantity: 

- 1 +

🛒 Add to Cart

Activate Windows  
Go to Settings to activate Windows.

## Product Card (Feature & Category Section):

```

{ /* Product Grid */
  <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6 px-4 sm:px-16">
    {products.map(({product: IProducts}) => {
      <div key={product._id} className="relative rounded-lg cursor-pointer bg-white">
        { /* Badge */
          {product.badge} && (
            <span
              className={`absolute top-2 left-2 px-2 py-1 text-white text-sm font-semibold rounded-md ${product.badge === "New" ? "bg-green-500" : "bg-orange-500"}
            >
              {product.badge}
            </span>
          )
        }

        { /* Product Image */
        <div className="w-full h-72 mb-4">
          <Image
            src={urlFor(product.image).url()}
            alt={product.title}
            width={250}
            height={500}
            className="w-full h-full object-cover rounded-md"
          />
        </div>

        { /* Product Title */
        <h3 className="text-base text-gray-800 hover:text-cyan-600 mb-2">
          {product.title}
        </h3>

        { /* Prices */
        <div className="mb-4">
          <span className="text-base font-semibold text-gray-800">
            {product.price}
          </span>
          {product.priceWithoutDiscount} && (
            <span className="text-sm font-semibold text-gray-500 line-through ml-2">
              {product.priceWithoutDiscount}
            </span>
          )
        </div>

        { /* Add to Cart Button */
        <div
          className="absolute bottom-4 right-4 p-2 bg-gray-200 text-black rounded-lg transition-all hover:text-white hover:bg-cyan-600 cursor-pointer"
          onClick={(e) => {
            e.preventDefault(); // Prevents navigation
            handleAddToCart(product); // Add to Cart on click
          }}
        >
          <ShoppingCart className="w-6 h-6" />
        </div>

        { /* Product Detail Link */
        <Link href={` /products/${product._id}`} >
          { /* No need to modify Link, it's used for product detail page */
        </Link>
        </div>
      )}
    </div>
  )}
</div>
</div>

```

## Screenshot: (Featured)

CIB BANK



BURNT  
TOAST

pdf PandaDoc

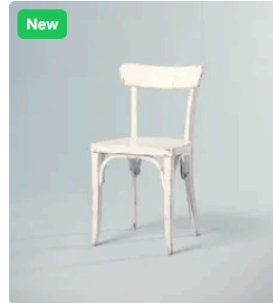
zapier

pipedrive

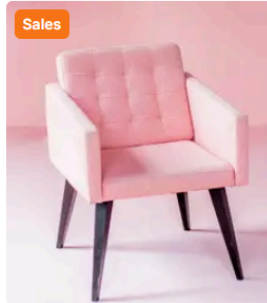
## Featured Products



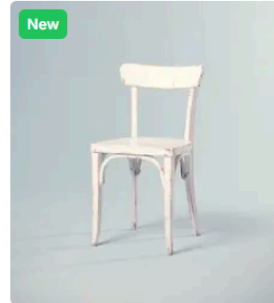
Citrus Edge  
\$20



Library Stool Chair  
\$20



Rose Luxe Armchair  
\$20 ~~\$39~~



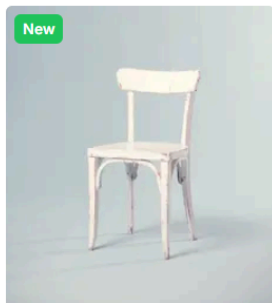
Library Stool Chair  
\$20



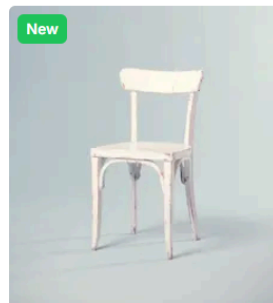
Activate Windows  
Go to Settings to activate Windows.

## Screenshot: (Category)

### Wooden Chair



Library Stool Chair  
\$20



Library Stool Chair  
\$20



Modern Cozy  
\$20



Scandi Dip Set  
\$40



Activate Windows  
Go to Settings to activate Windows.



## Search Bar – Implements real-time search functionality.

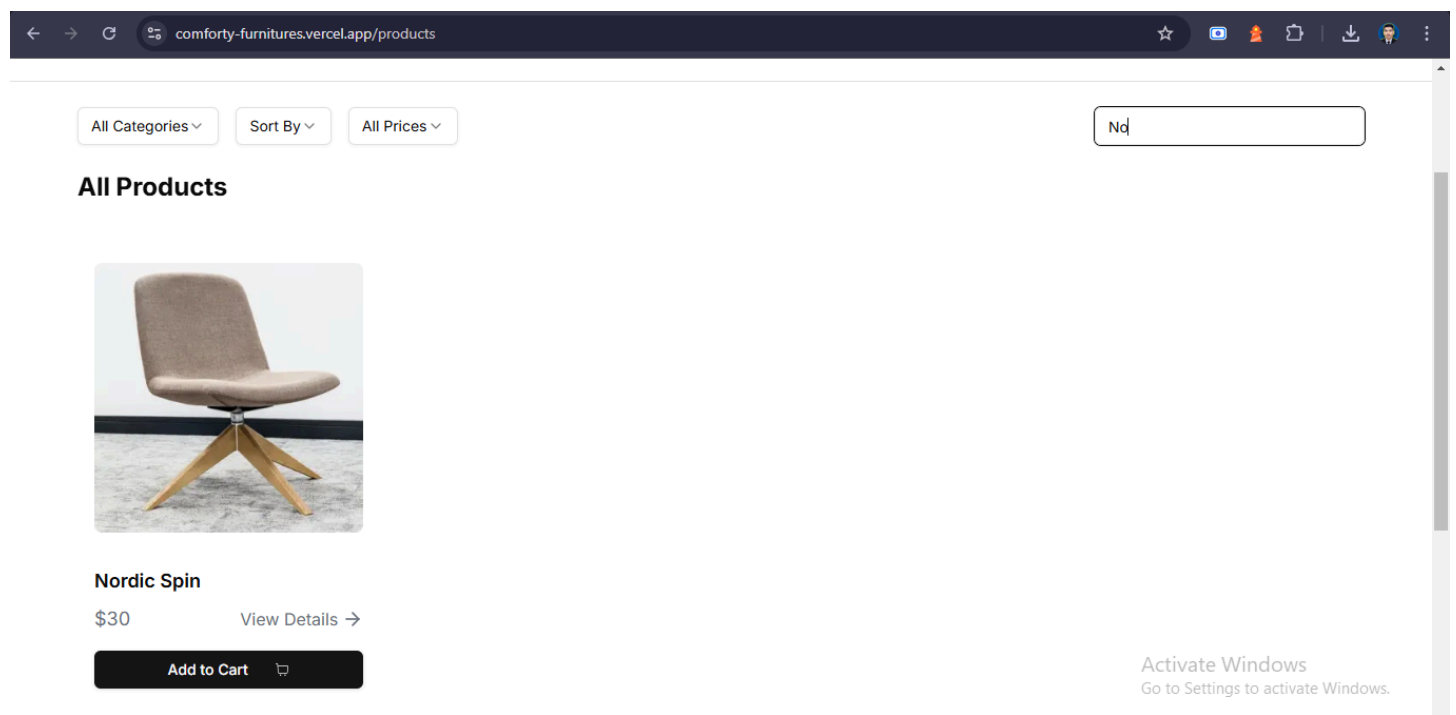
### Logic:

```
const { min, max } = getPriceRange(priceRange);
const filteredProducts = products.filter((product) =>
  product.title.toLowerCase().includes(debouncedSearch.toLowerCase()) &&
  (filter === "all" || product.category === filter) &&
  product.price >= min && product.price <= max
);
```

### Search bar using ShadCN UI element:

```
<Input
  type="text"
  placeholder="Search products..."
  className="w-full sm:w-64"
  value={searchQuery}
  onChange={(e) => setSearchQuery(e.target.value)}
/>
```

### Screenshot:





# Filters:

## Using ShadCN UI components:

```
<div className="hidden lg:flex space-x-4 items-center">
  <Select onValueChange={setFilter} defaultValue={filter}>
    <SelectTrigger><SelectValue placeholder="All Categories" /></SelectTrigger>
    <SelectContent>
      <SelectItem value="all">All Categories</SelectItem>
      <SelectItem value="wing-chair">Wing Chair</SelectItem>
      <SelectItem value="wooden-chair">Wooden Chair</SelectItem>
      <SelectItem value="office-chair">Office Chair</SelectItem>
    </SelectContent>
  </Select>

  <Select onValueChange={setSort} defaultValue={sort}>
    <SelectTrigger><SelectValue placeholder="Sort by" /></SelectTrigger>
    <SelectContent>
      <SelectItem value="default">Sort By</SelectItem>
      <SelectItem value="priceLow">Price (Low to High)</SelectItem>
      <SelectItem value="priceHigh">Price (High to Low)</SelectItem>
    </SelectContent>
  </Select>

  <Select onValueChange={setPriceRange} defaultValue={priceRange}>
    <SelectTrigger><SelectValue placeholder="All Prices" /></SelectTrigger>
    <SelectContent>
      <SelectItem value="all">All Prices</SelectItem>
      <SelectItem value="under10">Under $10</SelectItem>
      <SelectItem value="10to30">$10 - $30</SelectItem>
      <SelectItem value="30to50">$30 - $50</SelectItem>
      <SelectItem value="50to100">$50 - $100</SelectItem>
      <SelectItem value="100plus">$100+</SelectItem>
    </SelectContent>
  </Select>
</div>
```

## Screenshot: (Filters Implmented)

All Categories, Price (High to Low), and Price range as \$10 to \$30

All Products



Nordic Spin

\$30

View Details →

Add to Cart



SleekSpin

\$20

View Details →

Add to Cart



Citrus Edge

\$20

View Details →

Add to Cart



SleekSpin

\$20

View Details →

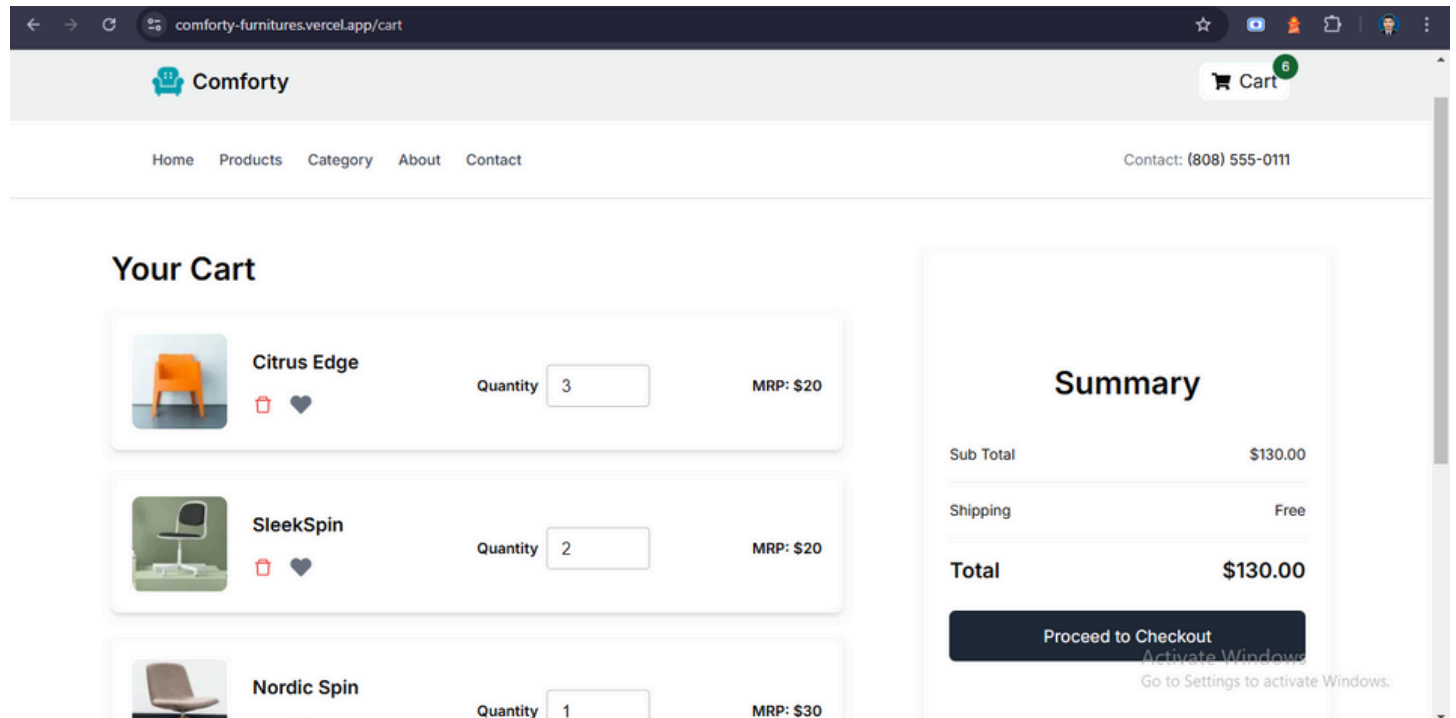
Add to Cart

Activate Windows  
Go to Settings to activate Windows.

# Cart:

Used Redux-Toolkit for state management

## Screenshot:



## 6. Challenges and Solutions

### 6.1 Challenges Encountered

- **Performance Issues:** Heavy UI elements caused lag.
- **State Management:** Handling complex component interactions.
- **Complex UI Integration:** Combining ShadCN UI and Material UI elements.
- **Cart Functionality:** Managing state across components efficiently.

### 6.2 Solutions Implemented

- Used **Lazy Loading** for images and components.
- Implemented **useContext & useReducer** for better state management.
- Used **ShadCN UI** for modern styling and **Material UI** for advanced UI components.
- Integrated **Redux Toolkit** to handle Add to Cart state effectively.

## 7. Conclusion

The development of dynamic frontend components for The Comforty Furnitures significantly improved user engagement and interactivity. These components are scalable, efficient, and adhere to best development practices.