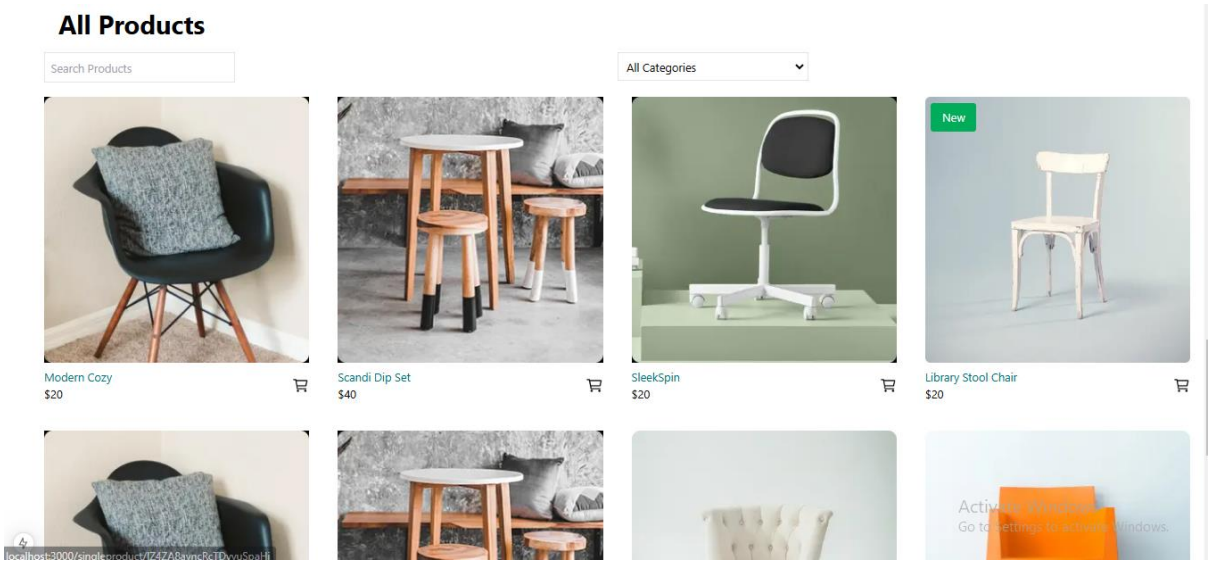
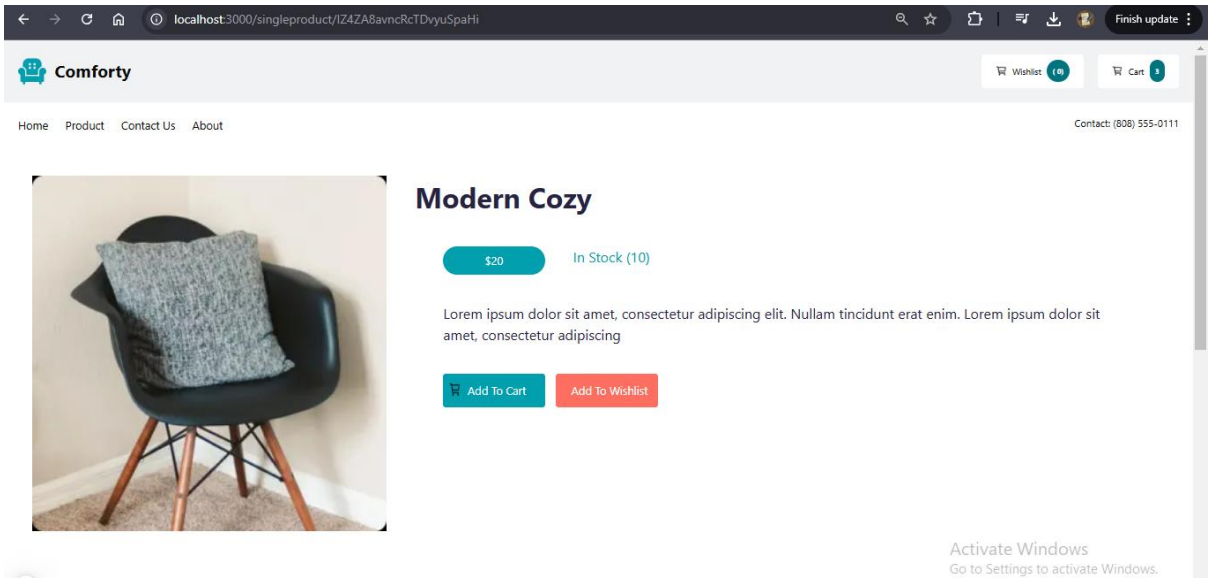
Day 4

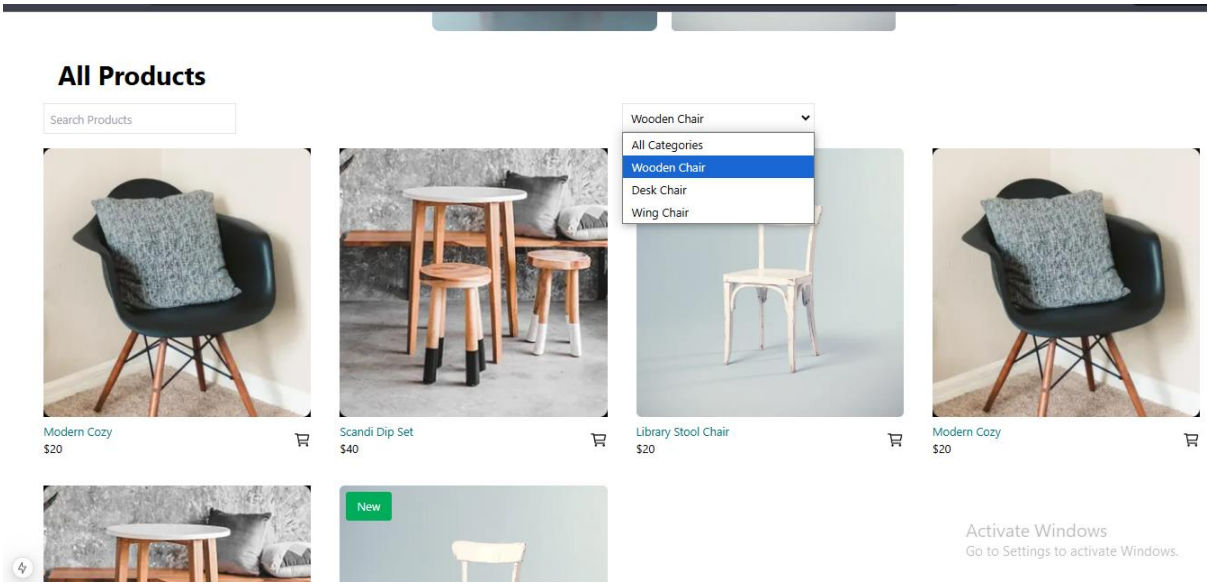
1. The product listing page with dynamic data.



2. Individual product detail pages with accurate routing and data rendering.

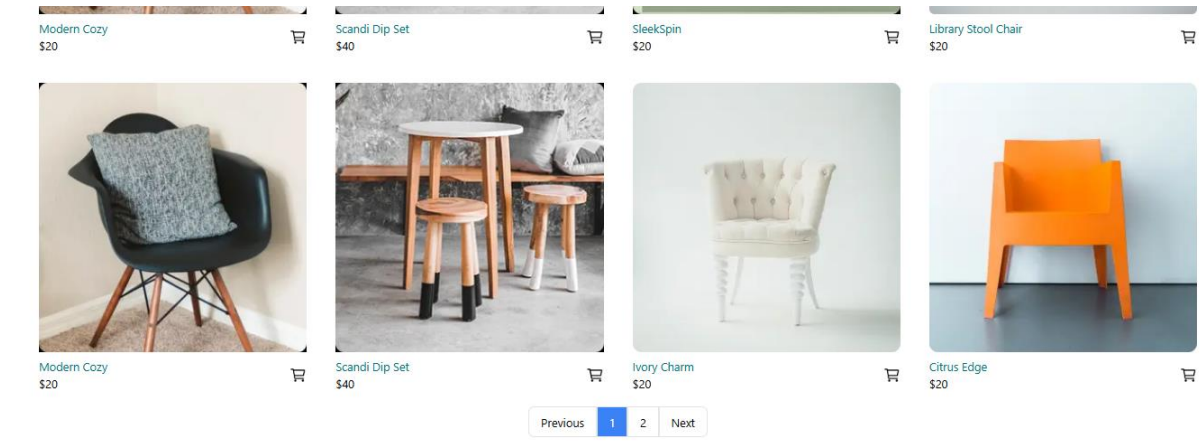


3. Working category filters.

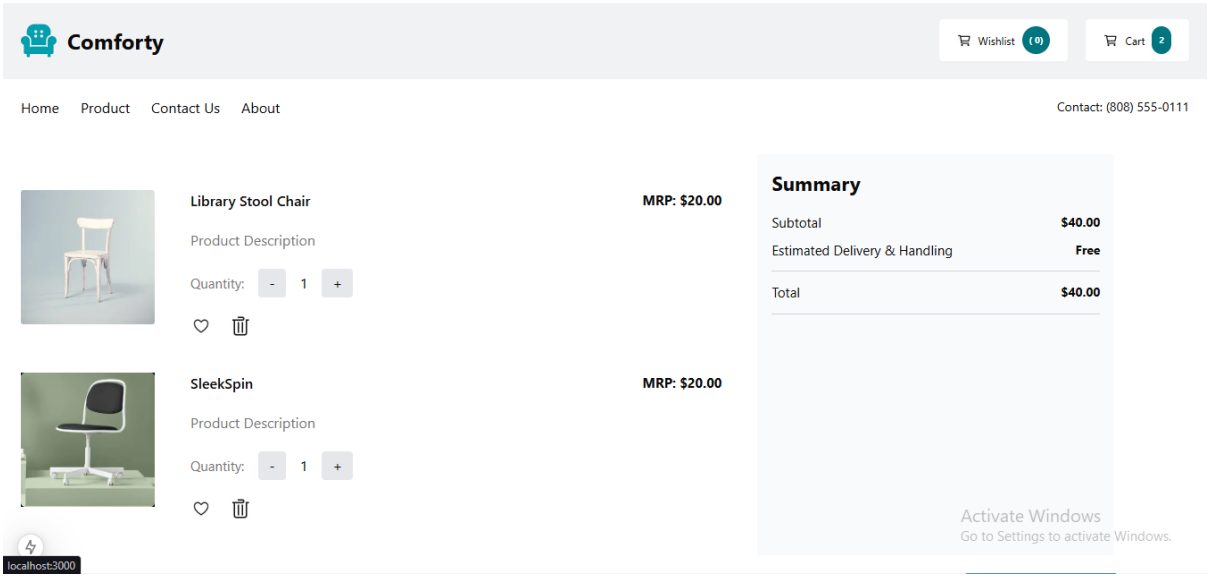


3. Working search bar

3. Working Pagination



// Add to card



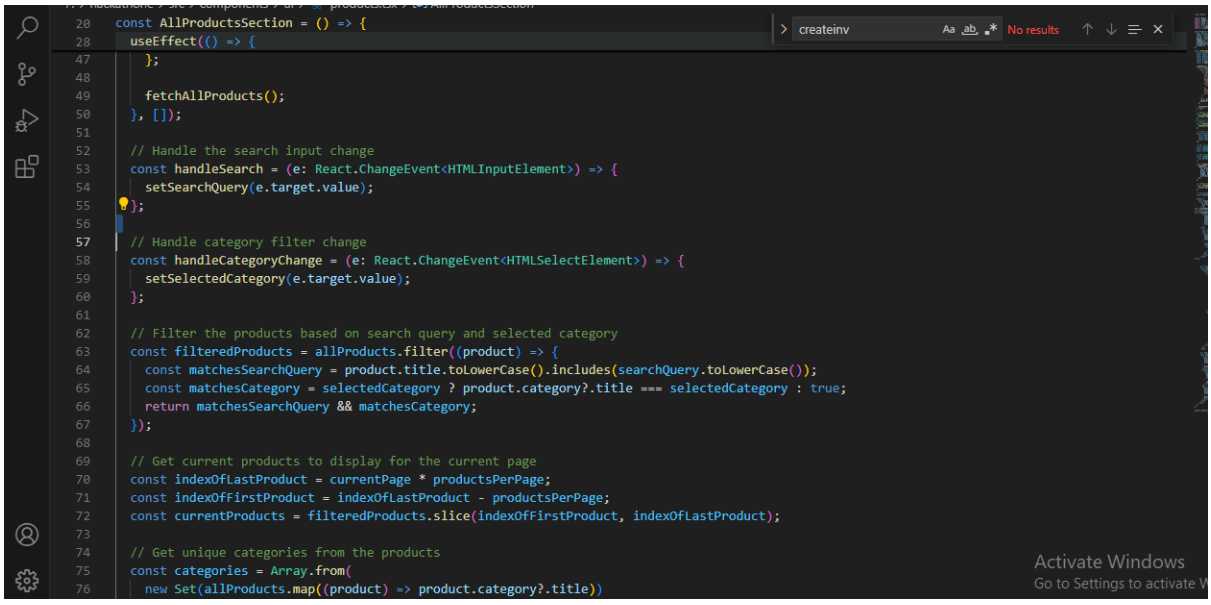
❖ **Code Deliverables:**

1. Code snippets for key components.

For dynamic data:

```tsx
export default function Shop({ params }: { params: { id: string } }) {
  const [product, setProduct] = useState<Product | null>(null);
  const [wishlist, setWishlist] = useState<Product[]>([]);

  const { addItem } = useShoppingCart();



  // Fetch product data
  useEffect(() => {
    const fetchProduct = async () => {
      const productData = await client.fetch(
        `*[_type == "products" && _id == $id][0]{
          title,
          price,
          description,
          "imageUrl": image.asset->url,
          inventory
        }`,
        { id: params.id }
      );
      setProduct(productData);
    };

    fetchProduct();
```

For filter code snippet:



Steps Taken to Build and Integrate Components

1. **Backend Configuration:**
    - Set up **Sanity CMS** to manage product data efficiently.
2. **Dynamic Data Retrieval:**
    - Integrated API calls in **Next.js** to fetch product data.
3. **Reusable Component Development:**
    - Built key reusable components:
        - **ProductCard:** Displays product details.
        - **ProductList:** Dynamically renders product collections.
        - **Filter:** Enables category-based filtering.
4. **Filter Integration:**
    - Implemented category filters for a seamless and intuitive user experience

Challenges and Solutions

1. **Filter Functionality:**
    - **Challenge:** Product lists didn’t update dynamically with selected categories.
    - **Solution:** Optimized filter logic to handle category changes smoothly and efficiently.
2. **Cart Management:**
    - **Challenge:** Managing the "Add to Cart" feature across the app.
    - **Solution:** Leveraged **“USE-SHOPPING-CART”** for streamlined state management and consistent functionality.
3. **Responsive Design:**
    - **Challenge:** Achieving a responsive interface across different devices.
    - **Solution:** Used Tailwind Css and conducted extensive testing to ensure compatibility on all screen sizes.