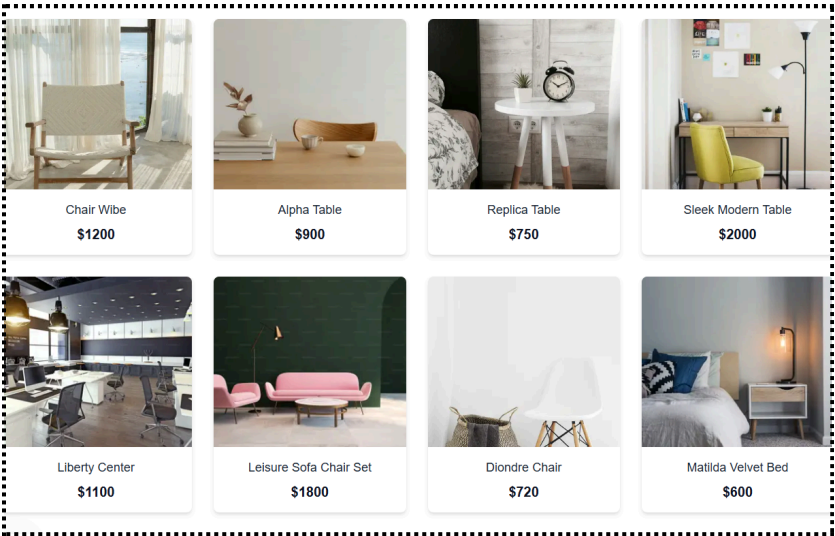


My Day 4 of Hackathone

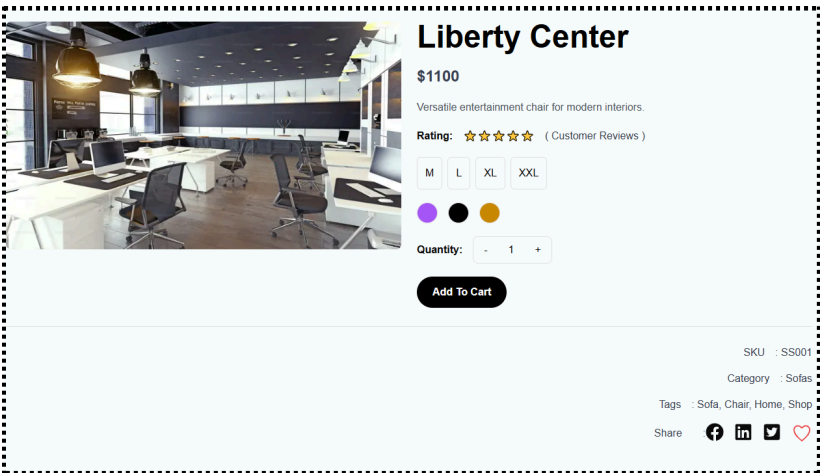
(Furniture website)

Product Listing Page with Dynamic Data:

The dynamic product listing page ensures all products are fetched and displayed in real-time, providing customers with up-to-date options. The individual product pages with dynamic routing personalize the shopping experience by allowing users to view detailed information tailored to their selected item.



Individual product Pge with Dynamic Routing:



```

34 }
35
36 // Fetch product data from Sanity to dynamic pages.....
37 async function getProductData(productId: string): Promise<Product | null> {
38   const query = `*_type == "product" && _id == $productId[0]`;
39   const product = await client.fetch(query, { productId });
40   return product;
41 }
42
43 function ProductPage() {
44   const [product, setProduct] = useState<Product | null>(null);
45   const [cart, setCart] = useState<CartItem[]>([]);
46   const [selectedSize] = useState<string | null>(null);
47   const [cartOpen, setCartOpen] = useState(false);
48
49   const routeParams = useParams();
50   const productId = routeParams.productId as string;
51
52   // Fetch data of product.....
53   useEffect(() => {
54     async function fetchData() {
55       if (productId) {
56         const productData = await getProductData(productId);
57         setProduct(productData);
58       }
59     }
60     fetchData();
61   }, [productId]);
62
63   // cart loading from local storage.....
64   useEffect(() => {

```

Add To Cart Functionality:

The cart functionality is user-friendly, enabling customers to effortlessly add items, view their selections, and make modifications before proceeding to checkout. This enhances the shopping journey by providing flexibility and control over purchases.

```

    if (storedCart) {
      setCart(JSON.parse(storedCart));
    }
  }, []);

  {
    // Cart save in local storage.....
  }
  useEffect(() => {
    localStorage.setItem("cart", JSON.stringify(cart));
  }, [cart]);


  // Add to Cart.....
  const addToCart = () => {
    if (!product) return;

    const existingItem = cart.find(
      (item) => item.id === product._id && item.size === selectedSize
    );

    if (existingItem) {
      setCart((prevCart) =>
        prevCart.map((item) =>
          item.id === product._id && item.size === selectedSize
            ? { ...item, quantity: item.quantity + 1 }
            : item
        )
      );
    } else {
      setCart((prevCart) => [...prevCart, { ...product, size: selectedSize, quantity: 1 }]);
    }
  };


```

Cart



Liberty Center

\$ 1100 x 1

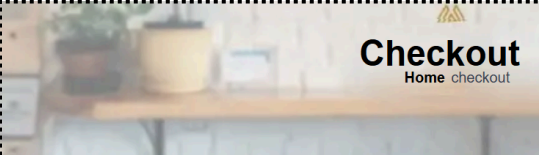


Total: \$ 1100

Checkout

CheckOut Page:

The checkout page is designed for efficiency, allowing users to review their cart and smoothly transition to payment. This ensures a hassle-free experience that boosts customer satisfaction.



Billing details

First Name

Last Name

Company Name (Optional)

Country / Reigion

Street address

Town / City

Province

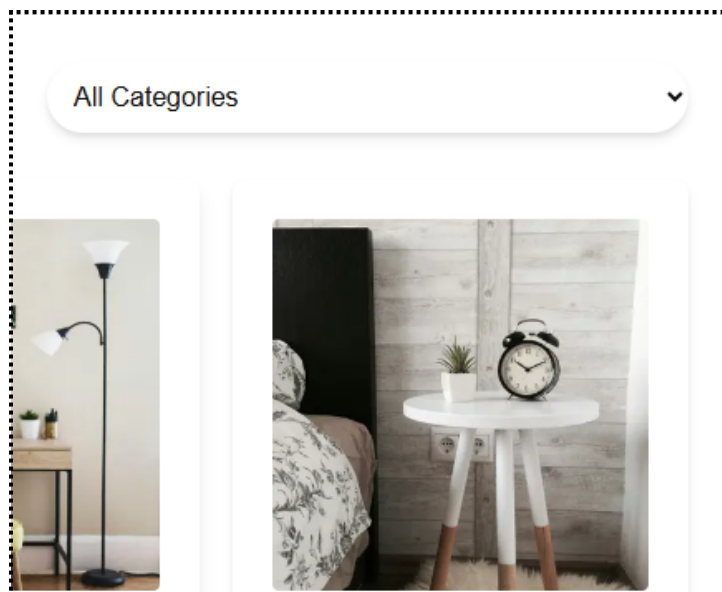
```

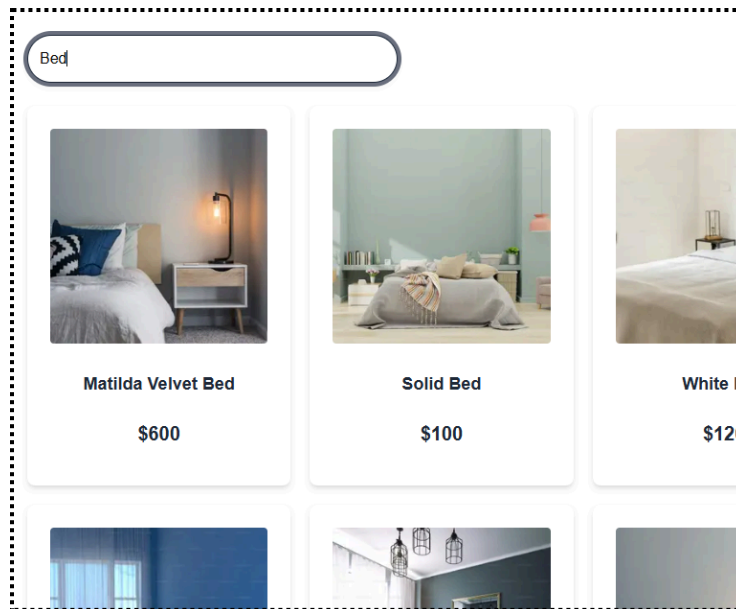
1 import React from 'react'
2 import Link from 'next/link'
3 import Image from 'next/image'
4 import Navbar from '../components/Navbar'
5 import Center from '../components/Center'
6 import Footer from '../components/footer'
7
8
9
10
11 const Checkout = () => {
12   return (
13     <>
14       <div className="max-w-screen-xl container mx-auto pb-8 px-4">
15         <div className="bg-[#faf4f4]">
16           <Navbar />
17         </div>
18
19         <div className="relative text-black">
20           
23
24           <h1 className="absolute top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2 text-2xl md:text-
25             Checkout
26           </h1>
27
28           <div className="absolute top-28 left-1/2 transform -translate-x-1/3 -translate-y-1/3 mt-10">
29             <p className="text-gray-700 text-xs md:text-xl flex items-center gap-2 mt-5">
30               <Link href="/" className="font-bold hover:underline text-black">Home</Link>
31               <Link href="/shop" className="hover:underline">checkout</Link>
32             </p>
33           </div>
34         </div>
35
36         <div className="flex flex-col gap-7 mx-10 mt-12 lg:flex-row sm:flex-col">
37           <div className="w-full md:mx-10">
38             <h1 className="text-black font-semibold text-4xl px-9">Billing details</h1>
39             <div className="grid grid-cols-1 gap-6 sm:grid-cols-2">
40               <div>
41                 <label htmlFor="firstname" className="block my-6 text-gray-800 font-semibold">First Name</lab
42                 <input type="text" id="firstname" className="w-full border rounded border-gray-400" />
43               </div>

```

Categories & Search Page:

The categories and search features make product discovery easy and efficient. By allowing users to browse through specific categories or search for items, the website ensures a quick and convenient navigation experience.





Conclusion:

Your e-commerce furniture website is a well-designed platform that provides users with a smooth and engaging shopping experience. With dynamic product listing and detail pages, efficient add-to-cart functionality, and a streamlined checkout process, it ensures convenience and ease for customers. The inclusion of categories and search features further enhances usability by simplifying product discovery. This comprehensive approach makes your website a reliable and user-friendly destination for furniture shopping.