# Day 4 - Dynamic Front-End Components - Bandage

## Project Overview:

The Dynamic Frontend Components project centers around developing a responsive and feature-rich interface for Furniro, an online furniture marketplace. The project aimed to build dynamic and reusable components using modern web technologies such as Next.js, React.js, and Sanity CMS. It provided an opportunity to delve into advanced concepts like dynamic routing, state management, and API integrations.

**Technologies Employed:**

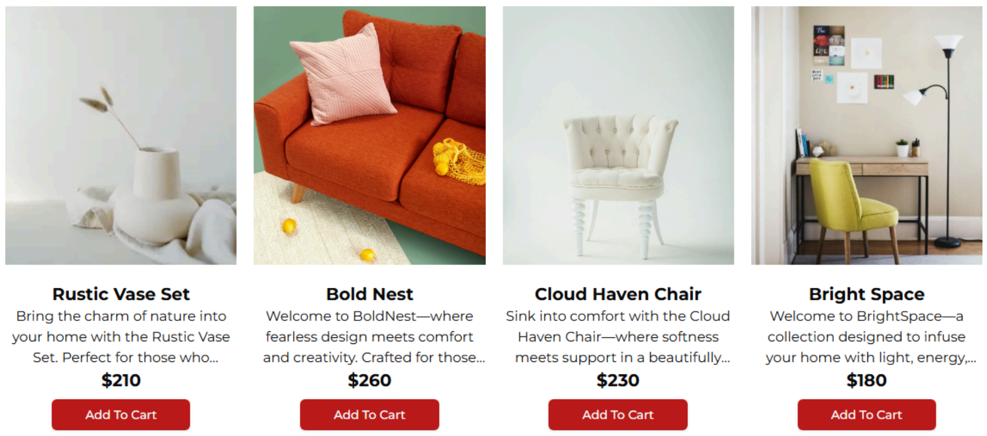
- **Next.js**: Used for server-side rendering, dynamic routing, and the overall structure of the frontend application.
- **React.js**: Applied for creating modular and reusable UI components, as well as managing state across the interface.
- **Sanity CMS**: Utilized for managing and storing dynamic content, such as product information.
- **Shadcn UI**: Used Shadcn UI's pre-designed, customizable components to enhance UI consistency and speed up development.

**Dynamic Frontend Components:**

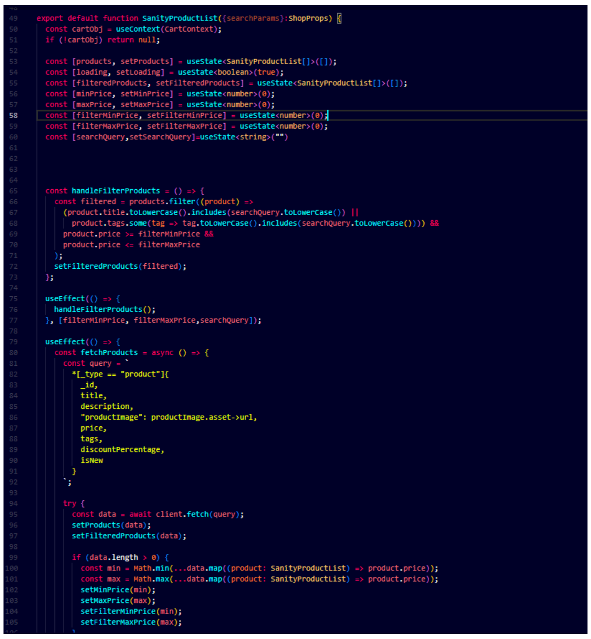
**1. Product Listing Component:** The product listing page dynamically fetches product data from Sanity CMS, presenting items in a responsive grid layout. Each product showcases key details such as:

  - Title of the product
  - A brief description
  - Product image
  - Price

The layout automatically adjusts for various screen sizes, allowing users to easily navigate through a selection of products, ensuring a smooth and engaging browsing experience


Rustic Vase Set
Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who...
$210
Add To Cart
Bold Nest
Welcome to BoldNest—where fearless design meets comfort and creativity. Crafted for those...
$260
Add To Cart
Cloud Haven Chair
Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully...
$230
Add To Cart
Bright Space
Welcome to BrightSpace—a collection designed to infuse your home with light, energy,...
$180
Add To Cart

## Code Snippet:

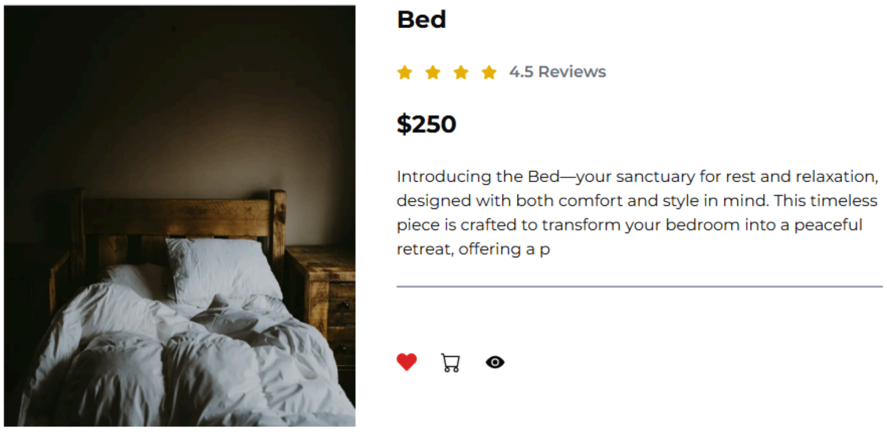


```
export default function SanityProductList({searchParams}:ShopProps) {
  const cartObj = useContext(CartContext);
  if (!cartObj) return null;

  const [products, setProducts] = useState<SanityProductList[]>([]);
  const [loading, setLoading] = useState<boolean>(true);
  const [filteredProducts, setFilteredProducts] = useState<SanityProductList[]>([]);
  const [minPrice, setMinPrice] = useState<number>(0);
  const [maxPrice, setMaxPrice] = useState<number>(0);
  const [filterMinPrice, setFilterMinPrice] = useState<number>(0);
  const [filterMaxPrice, setFilterMaxPrice] = useState<number>(0);
  const [searchQuery,setSearchQuery]=useState<string>("")




  const handleFilterProducts = () => {
    const filtered = products.filter((product) =>
      (product.title.toLowerCase().includes(searchQuery.toLowerCase()) ||
        product.tags.some(tag => tag.toLowerCase().includes(searchQuery.toLowerCase()))) &&
      product.price >= filterMinPrice &&
      product.price <= filterMaxPrice
    );
    setFilteredProducts(filtered);
  };

  useEffect(() => {
    handleFilterProducts();
  }, [filterMinPrice, filterMaxPrice,searchQuery]);

  useEffect(() => {
    const fetchProducts = async () => {
      const query = `
        *[_type == "product"]{
          _id,
          title,
          description,
          "productImage": productImage.asset->url,
          price,
          tags,
          discountPercentage,
          isNew
        }
      `;

      try {
        const data = await client.fetch(query);
        setProducts(data);
        setFilteredProducts(data);

        if (data.length > 0) {
          const min = Math.min(...data.map((product: SanityProductList) => product.price));
          const max = Math.max(...data.map((product: SanityProductList) => product.price));
          setMinPrice(min);
          setMaxPrice(max);
          setFilterMinPrice(min);
          setFilterMaxPrice(max);
```

**2. Product Details Component**: The product details page offers in-depth information about each item. By utilizing Next.js dynamic routing, we created unique pages for each product using its specific ID.

**Steps taken to build**:

- **Dynamic Routing**: Employed Next.js's [id].js file to dynamically generate pages based on each product's ID.
- **Reusable UI Components**: Elements such as buttons, rating stars, and badges were reused across the page to maintain consistency and improve development efficiency.



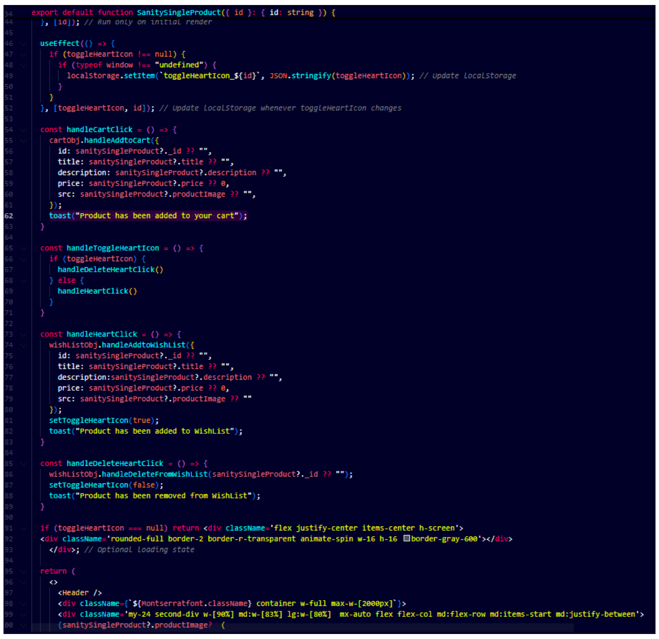
Bed

★★★★ 4.5 Reviews

$250

Introducing the Bed—your sanctuary for rest and relaxation, designed with both comfort and style in mind. This timeless piece is crafted to transform your bedroom into a peaceful retreat, offering a p

## Code Snippet:

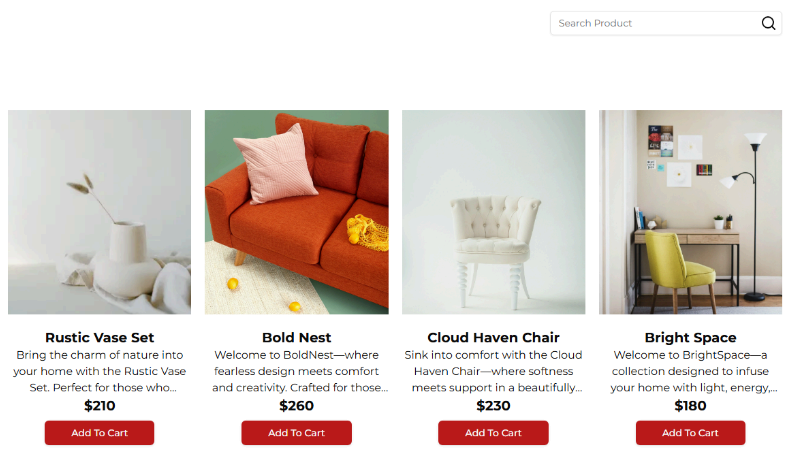


```tsx
export default function SanitySingleProduct({ id }: { id: string }) {
  }, [id]); // Run only on initial render

  useEffect(() => {
    if (toggleHeartIcon !== null) {
      if (typeof window !== "undefined") {
        localStorage.setItem(`toggleHeartIcon_${id}`, JSON.stringify(toggleHeartIcon)); // Update localStorage
      }
    }
  }, [toggleHeartIcon, id]); // Update localStorage whenever toggleHeartIcon changes

  const handleCartClick = () => {
    cartObj.handleAddtoCart({
      id: sanitySingleProduct?._id ?? "",
      title: sanitySingleProduct?.title ?? "",
      description: sanitySingleProduct?.description ?? "",
      price: sanitySingleProduct?.price ?? 0,
      src: sanitySingleProduct?.productImage ?? "",
    });
    toast("Product has been added to your cart");
  }

  const handleToggleHeartIcon = () => {
    if (toggleHeartIcon) {
      handleDeleteHeartClick()
    } else {
      handleHeartClick()
    }
  }

  const handleHeartClick = () => {
    wishListObj.handleAddtoWishList({
      id: sanitySingleProduct?._id ?? "",
      title: sanitySingleProduct?.title ?? "",
      description:sanitySingleProduct?.description ?? "",
      price: sanitySingleProduct?.price ?? 0,
      src: sanitySingleProduct?.productImage ?? ""
    });
    setToggleHeartIcon(true);
    toast("Product has been added to WishList");
  }

  const handleDeleteHeartClick = () => {
    wishListObj.handleDeleteFromWishList(sanitySingleProduct?._id ?? "");
    setToggleHeartIcon(false);
    toast("Product has been removed from WishList");
  }

  if (toggleHeartIcon === null) return <div className='flex justify-center items-center h-screen'>
  <div className='rounded-full border-2 border-r-transparent animate-spin w-16 h-16 border-gray-600'></div>
    </div>; // Optional loading state

  return (
    <>
      <Header />
      <div className={`${Montserratfont.className} container w-full max-w-[2000px]`}>
      <div className='my-24 second-div w-[90%] md:w-[83%] lg:w-[80%]  mx-auto flex flex-col md:flex-row md:items-start md:justify-between'>
      {sanitySingleProduct?.productImage?  (
```

## 3. Search Bar:

A search bar was incorporated to allow users to filter products by name or tags instantly. Positioned on the product listing page, it updates the visible products in real-time as the user types their query.

Before Search:



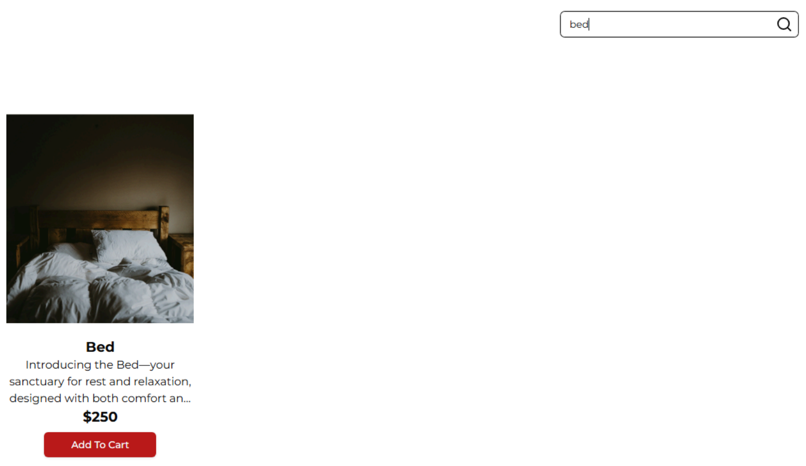
**Rustic Vase Set**
Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who...
$210

**Bold Nest**
Welcome to BoldNest—where fearless design meets comfort and creativity. Crafted for those...
$260

**Cloud Haven Chair**
Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully...
$230

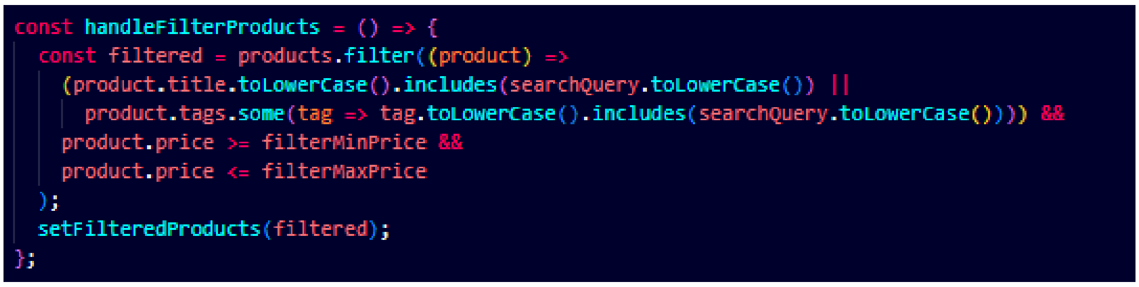
**Bright Space**
Welcome to BrightSpace—a collection designed to infuse your home with light, energy,...
$180

After Search:



**Bed**
Introducing the Bed—your sanctuary for rest and relaxation, designed with both comfort an...
$250

## Code Snippet:



```
const handleFilterProducts = () => {
  const filtered = products.filter((product) =>
    (product.title.toLowerCase().includes(searchQuery.toLowerCase()) ||
      product.tags.some(tag => tag.toLowerCase().includes(searchQuery.toLowerCase()))) &&
    product.price >= filterMinPrice &&
    product.price <= filterMaxPrice
  );
  setFilteredProducts(filtered);
};
```

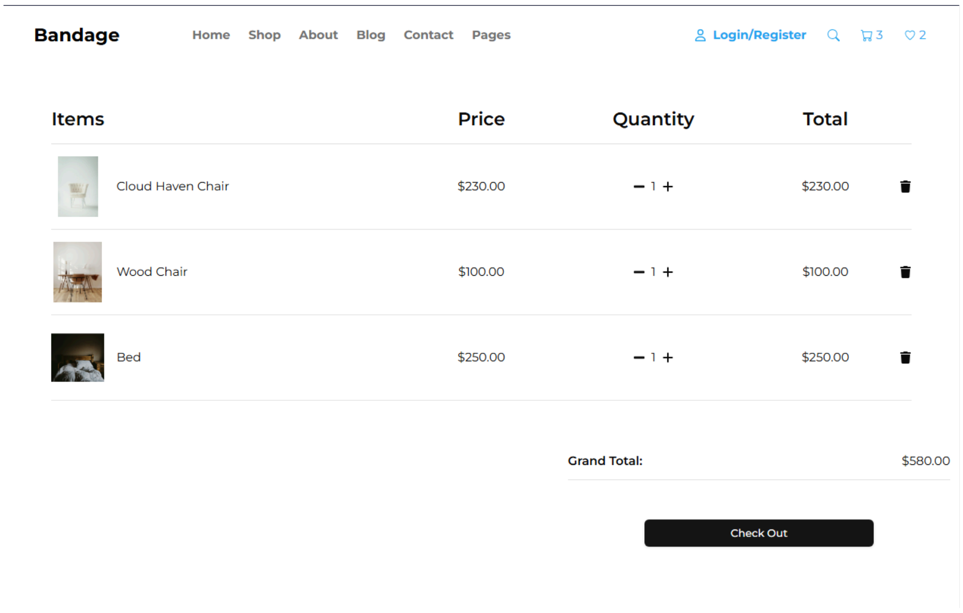
## 4. Add To Cart Feature:

The Shopping Cart feature allows users to view and manage the products they have selected.

### Steps taken to build:

- **Context API & Local Storage:** We utilized React's Context API for global state management, allowing the cart data to be accessible across components. Additionally, local storage is used to persist cart items, ensuring they are retained even after the user refreshes or reopens the page.

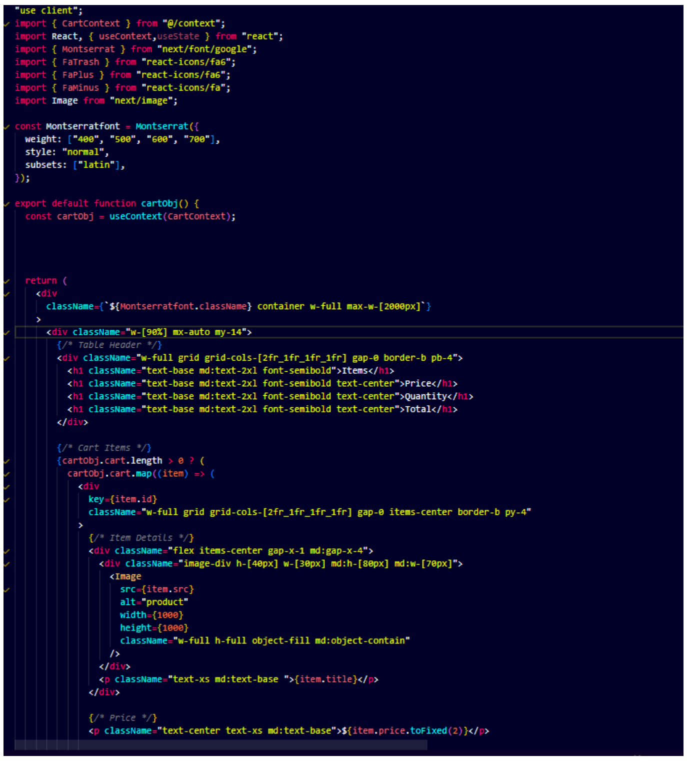
- **Features:**
  - Displays a list of added products with details such as name, price, and quantity.
  - Provides an option for users to modify the quantity of each product.
  - Automatically computes and shows the total price and grand total.
  - Allows users to remove items from the cart.
  - Displays a toast notification when an item is successfully added .
  - The cart icon in the header dynamically shows the number of items currently in the cart.



| Items | Price | Quantity | Total |
|---|---|---|---|
| Cloud Haven Chair | $230.00 | − 1 + | $230.00 |
| Wood Chair | $100.00 | − 1 + | $100.00 |
| Bed | $250.00 | − 1 + | $250.00 |

Grand Total: $580.00

## Code Snippet:



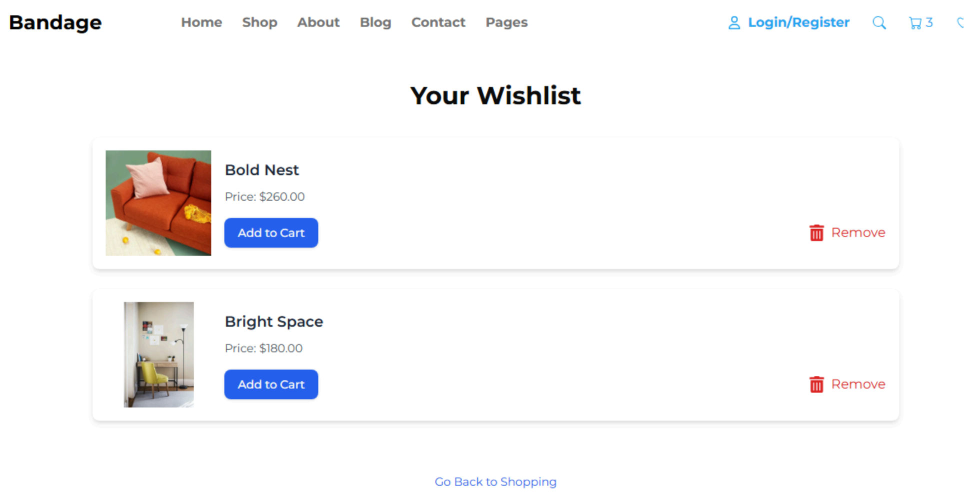
```
"use client";
import { CartContext } from "@/context";
import React, { useContext,useState } from "react";
import { Montserrat } from "next/font/google";
import { FaTrash } from "react-icons/fa6";
import { FaPlus } from "react-icons/fa6";
import { FaMinus } from "react-icons/fa";
import Image from "next/image";

const Montserratfont = Montserrat({
  weight: ["400", "500", "600", "700"],
  style: "normal",
  subsets: ["latin"],
});

export default function cartObj() {
  const cartObj = useContext(CartContext);



  return (
    <div
      className={`${Montserratfont.className} container w-full max-w-[2000px]`}
    >
      <div className="w-[90%] mx-auto my-14">
        {/* Table Header */}
        <div className="w-full grid grid-cols-[2fr_1fr_1fr_1fr] gap-0 border-b pb-4">
          <h1 className="text-base md:text-2xl font-semibold">Items</h1>
          <h1 className="text-base md:text-2xl font-semibold text-center">Price</h1>
          <h1 className="text-base md:text-2xl font-semibold text-center">Quantity</h1>
          <h1 className="text-base md:text-2xl font-semibold text-center">Total</h1>
        </div>

        {/* Cart Items */}
        {cartObj.cart.length > 0 ? (
          cartObj.cart.map((item) => (
            <div
              key={item.id}
              className="w-full grid grid-cols-[2fr_1fr_1fr_1fr] gap-0 items-center border-b py-4"
            >
              {/* Item Details */}
              <div className="flex items-center gap-x-1 md:gap-x-4">
                <div className="image-div h-[40px] w-[30px] md:h-[80px] md:w-[70px]">
                  <Image
                    src={item.src}
                    alt="product"
                    width={1000}
                    height={1000}
                    className="w-full h-full object-fill md:object-contain"
                  />
                </div>
                <p className="text-xs md:text-base ">{item.title}</p>
              </div>

              {/* Price */}
              <p className="text-center text-xs md:text-base">${item.price.toFixed(2)}</p>
```

**5. Wishlist Component:**

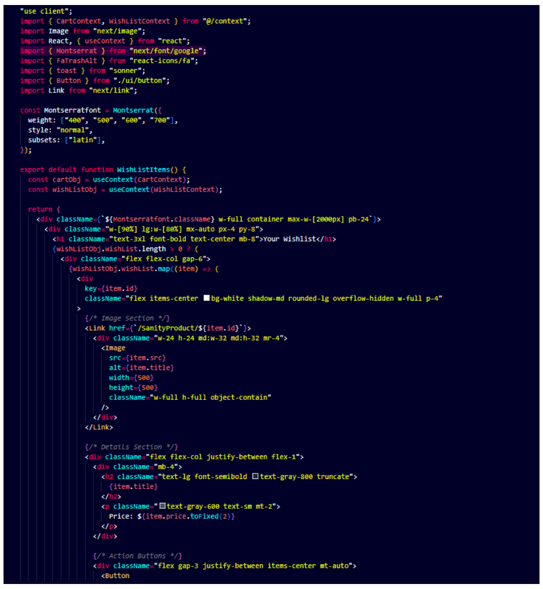
The Wishlist component allows users to save products they are interested in, without adding them to their cart, for future reference or purchase.

**Steps taken to build:**

- **Context API & Local Storage:** Similar to the cart component, we utilized React's Context API to manage the wishlist state globally. Local storage is also used to ensure the wishlist persists across sessions, even after the user refreshes or revisits the page.
- **Features:**
  - Displays a list of products that the user has added to their wishlist.
  - Users can remove products from the wishlist when they are no longer interested.
  - Users can add items from the wishlist to the cart with a single click.
  - A success toast notification appears when items are added to or removed from the wishlist.
  - The header includes a dynamic wishlist icon that shows the number of items currently in the wishlist.
  - Allows users to toggle between viewing products in the wishlist or the product details page.



## Code Snippet:



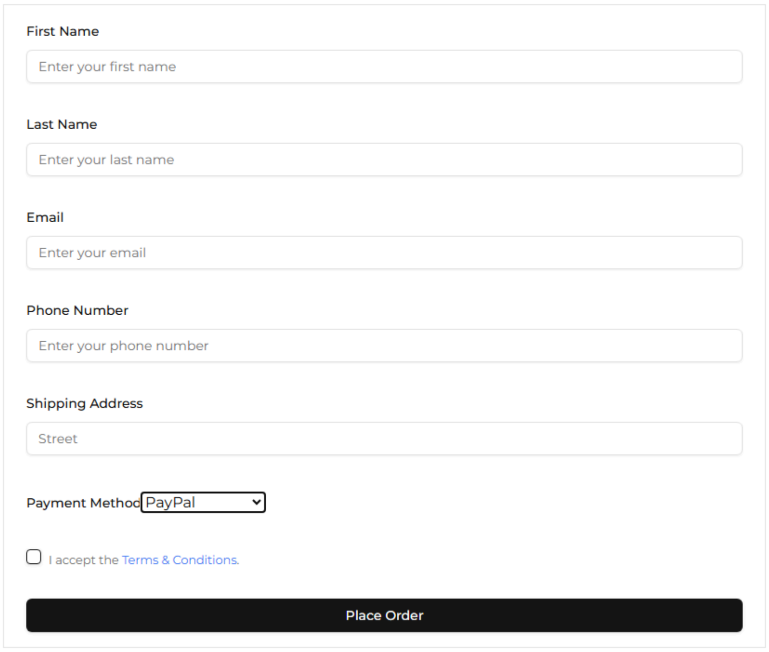
```jsx
"use client";
import { CartContext, WishListContext } from "@/context";
import Image from "next/image";
import React, { useContext } from "react";
import { Montserrat } from "next/font/google";
import { FaTrashAlt } from "react-icons/fa";
import { toast } from "sonner";
import { Button } from "./ui/button";
import Link from "next/link";

const Montserratfont = Montserrat({
  weight: ["400", "500", "600", "700"],
  style: "normal",
  subsets: ["latin"],
});

export default function WishListItems() {
  const cartObj = useContext(CartContext);
  const wishListObj = useContext(WishListContext);

  return (
    <div className={`${Montserratfont.className} w-full container max-w-[2000px] pb-24`}>
      <div className="w-[90%] lg:w-[80%] mx-auto px-4 py-8">
        <h1 className="text-3xl font-bold text-center mb-8">Your Wishlist</h1>
        {wishListObj.wishList.length > 0 ? (
          <div className="flex flex-col gap-6">
            {wishListObj.wishList.map((item) => (
              <div
                key={item.id}
                className="flex items-center bg-white shadow-md rounded-lg overflow-hidden w-full p-4"
              >
                {/* Image Section */}
                <Link href={`/SanityProduct/${item.id}`}>
                  <div className="w-24 h-24 md:w-32 md:h-32 mr-4">
                    <Image
                      src={item.src}
                      alt={item.title}
                      width={500}
                      height={500}
                      className="w-full h-full object-contain"
                    />
                  </div>
                </Link>

                {/* Details Section */}
                <div className="flex flex-col justify-between flex-1">
                  <div className="mb-4">
                    <h2 className="text-lg font-semibold text-gray-800 truncate">
                      {item.title}
                    </h2>
                    <p className="text-gray-600 text-sm mt-2">
                      Price: ${item.price.toFixed(2)}
                    </p>
                  </div>

                  {/* Action Buttons */}
                  <div className="flex gap-3 justify-between items-center mt-auto">
                    <Button
```

**6. Checkout Flow Component:**

The Checkout Flow component provides a multi-step form that guides users through the process of completing their purchase, collecting necessary details for billing, shipping, and payment.

**Steps taken to build:**

- **Multi-Step Form:** We created a step-by-step process, allowing users to easily navigate through the checkout flow .
- **Shadcn Form Components:** Utilized Shadcn's form components for managing input fields, validation, and user interactions, ensuring a consistent UI/UX across all steps.
- **Features:**
  - **Step 1:** Collects billing and shipping address information. Users are prompted to enter their name, address, contact details, and preferred shipping method.
  - **Step 2:** Gathers payment details, including card information .
  - **Dynamic Validation:** Real-time validation ensures that all required fields are correctly filled before progressing to the next step.
  - **Order Confirmation:** After completing the payment step, users receive an order confirmation and a success message.



First Name

Enter your first name

Last Name

Enter your last name

Email

Enter your email

Phone Number

Enter your phone number

Shipping Address

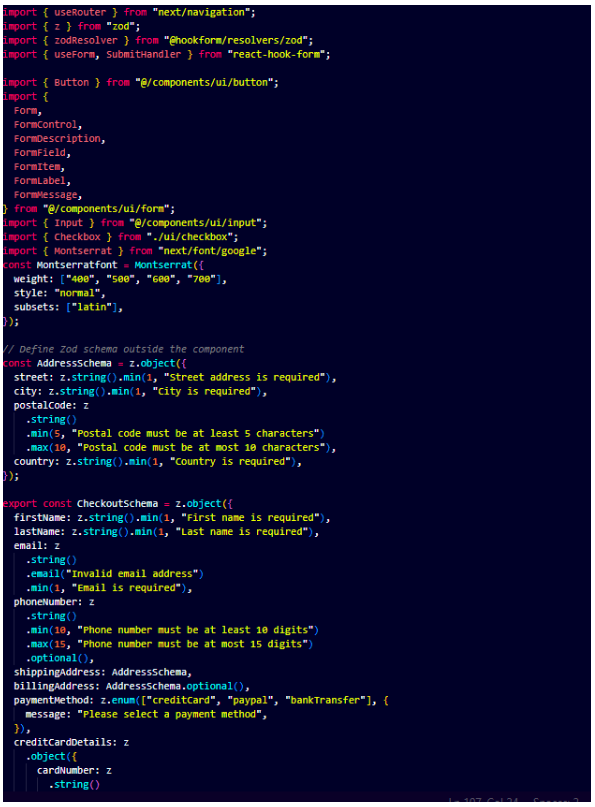
Street

Payment Method PayPal

☐ I accept the Terms & Conditions.

Place Order

## Code Snippet:



```
import { useRouter } from "next/navigation";
import { z } from "zod";
import { zodResolver } from "@hookform/resolvers/zod";
import { useForm, SubmitHandler } from "react-hook-form";

import { Button } from "@/components/ui/button";
import {
  Form,
  FormControl,
  FormDescription,
  FormField,
  FormItem,
  FormLabel,
  FormMessage,
} from "@/components/ui/form";
import { Input } from "@/components/ui/input";
import { Checkbox } from "./ui/checkbox";
import { Montserrat } from "next/font/google";
const Montserratfont = Montserrat({
  weight: ["400", "500", "600", "700"],
  style: "normal",
  subsets: ["latin"],
});

// Define Zod schema outside the component
const AddressSchema = z.object({
  street: z.string().min(1, "Street address is required"),
  city: z.string().min(1, "City is required"),
  postalCode: z
    .string()
    .min(5, "Postal code must be at least 5 characters")
    .max(10, "Postal code must be at most 10 characters"),
  country: z.string().min(1, "Country is required"),
});

export const CheckoutSchema = z.object({
  firstName: z.string().min(1, "First name is required"),
  lastName: z.string().min(1, "Last name is required"),
  email: z
    .string()
    .email("Invalid email address")
    .min(1, "Email is required"),
  phoneNumber: z
    .string()
    .min(10, "Phone number must be at least 10 digits")
    .max(15, "Phone number must be at most 15 digits")
    .optional(),
  shippingAddress: AddressSchema,
  billingAddress: AddressSchema.optional(),
  paymentMethod: z.enum(["creditCard", "paypal", "bankTransfer"], {
    message: "Please select a payment method",
  }),
  creditCardDetails: z
    .object({
      cardNumber: z
        .string()
```

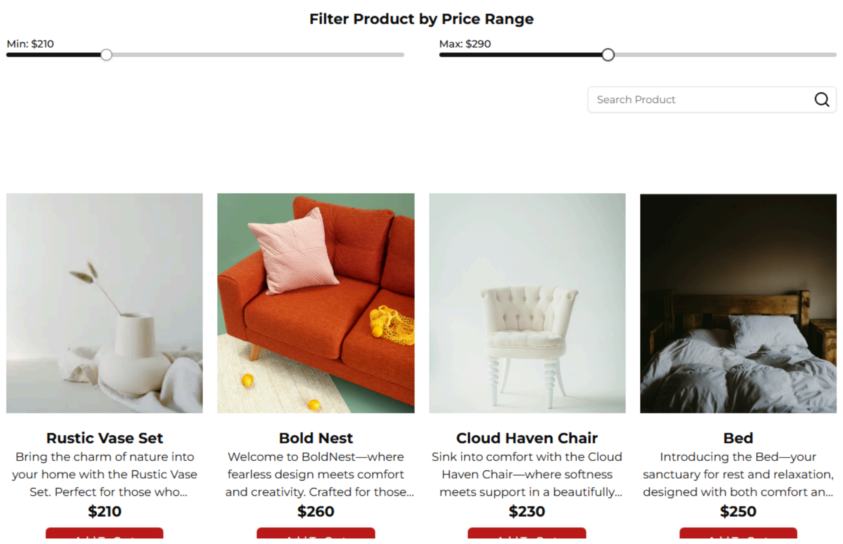
## 7. Filter Panel Component:

The Filter Panel component allows users to narrow down product listings by adjusting the price range through an interactive slider.

### Steps taken to build:

- **Price Range Slider:** Implemented a price range slider that lets users filter products based on a minimum and maximum price range.

- **State Management:** Utilized React's useState hook to dynamically update the displayed products as the price range is adjusted.
- **Real-Time Filtering:** As users slide the range, the product listings are instantly updated, providing a seamless and interactive experience.
- **Responsive Design:** Ensured that the filter panel is fully responsive, adapting to different screen sizes for both desktop and mobile users.



Filter Product by Price Range

Min: $210

Max: $290

Search Product

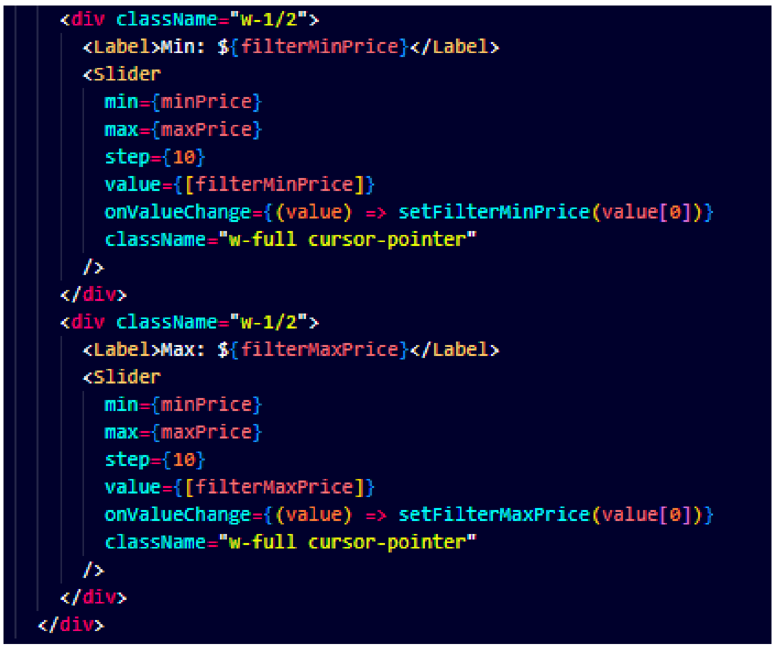
**Rustic Vase Set**
Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who...
**$210**

**Bold Nest**
Welcome to BoldNest—where fearless design meets comfort and creativity. Crafted for those...
**$260**

**Cloud Haven Chair**
Sink into comfort with the Cloud Haven Chair—where softness meets support in a beautifully...
**$230**

**Bed**
Introducing the Bed—your sanctuary for rest and relaxation, designed with both comfort an...
**$250**

## Code Snippet:



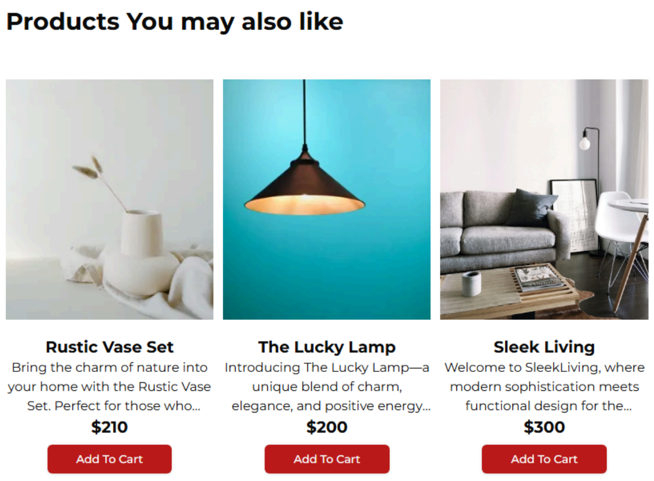
```
  <div className="w-1/2">
    <Label>Min: ${filterMinPrice}</Label>
    <Slider
      min={minPrice}
      max={maxPrice}
      step={10}
      value={[filterMinPrice]}
      onValueChange={(value) => setFilterMinPrice(value[0])}
      className="w-full cursor-pointer"
    />
  </div>
  <div className="w-1/2">
    <Label>Max: ${filterMaxPrice}</Label>
    <Slider
      min={minPrice}
      max={maxPrice}
      step={10}
      value={[filterMaxPrice]}
      onValueChange={(value) => setFilterMaxPrice(value[0])}
      className="w-full cursor-pointer"
    />
  </div>
</div>
```

## 8. Related Products Component:

The Related Products component displays suggestions for similar or complementary items on the product detail page.

### Steps taken to build:

- **Product Suggestions:** Based on the current product's tags, the component fetches and displays related products.
- **Dynamic Data Fetching:** Utilized Next.js's dynamic data fetching to retrieve products that share common tags, ensuring that the recommendations are relevant to the user's interests.
- **Carousel Layout:** Displayed related products in a horizontally scrollable carousel to enhance user engagement while keeping the page layout clean and organized.
- **Responsive Design:** Ensured the component adapts to different screen sizes, maintaining usability across devices.



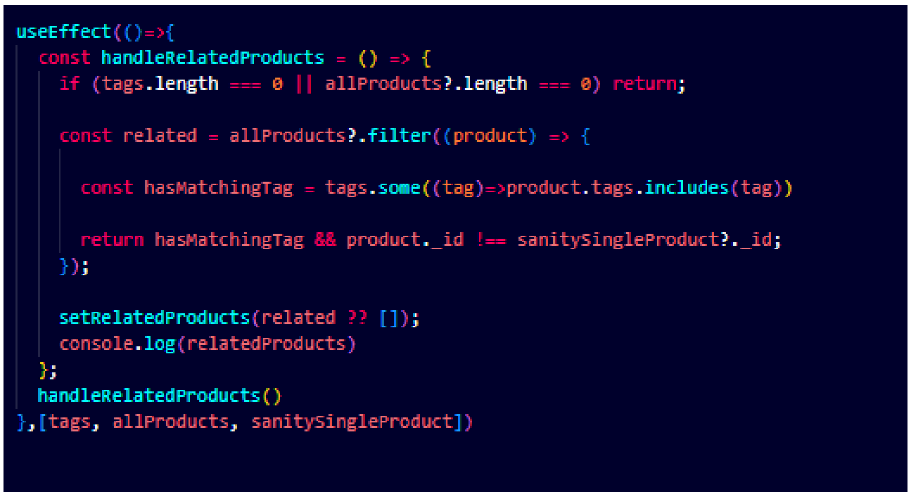
Products You may also like

Rustic Vase Set
Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who...
$210
Add To Cart

The Lucky Lamp
Introducing The Lucky Lamp—a unique blend of charm, elegance, and positive energy...
$200
Add To Cart

Sleek Living
Welcome to SleekLiving, where modern sophistication meets functional design for the...
$300
Add To Cart

## Code Snippet:



```
useEffect(()=>{
  const handleRelatedProducts = () => {
    if (tags.length === 0 || allProducts?.length === 0) return;

    const related = allProducts?.filter((product) => {

      const hasMatchingTag = tags.some((tag)=>product.tags.includes(tag))

      return hasMatchingTag && product._id !== sanitySingleProduct?._id;
    });

    setRelatedProducts(related ?? []);
    console.log(relatedProducts)
  };
  handleRelatedProducts()
},[tags, allProducts, sanitySingleProduct])
```

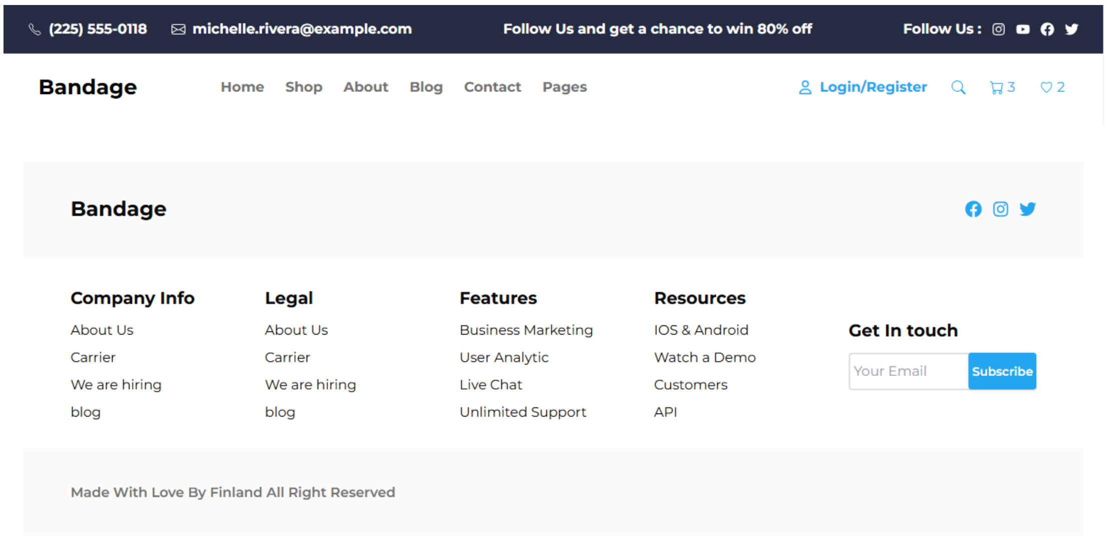
## 9. Footer and Header Components:

The Footer and Header components provide essential navigation and branding elements for the application.

### Steps taken to build:

- **Navigation Links:** Both components include links to key pages such as Home, About, and Contact for easy access throughout the site.

- **Branding:** The header displays the company logo and serves as the primary branding element.
- **Responsiveness:** Ensured the components adapt to various screen sizes, offering a seamless experience across devices.
- **Accessibility:** Incorporated accessibility features, such as proper ARIA labels and keyboard navigation support, to make the site more inclusive.
- **Sticky Header:** Implemented a sticky header that remains visible while scrolling for easier navigation.
- **Footer Information:** The footer also includes additional information like social media links, terms of service, and privacy policy for quick reference.


(225) 555-0118 michelle.rivera@example.com Follow Us and get a chance to win 80% off Follow Us :

Bandage Home Shop About Blog Contact Pages Login/Register 3 2

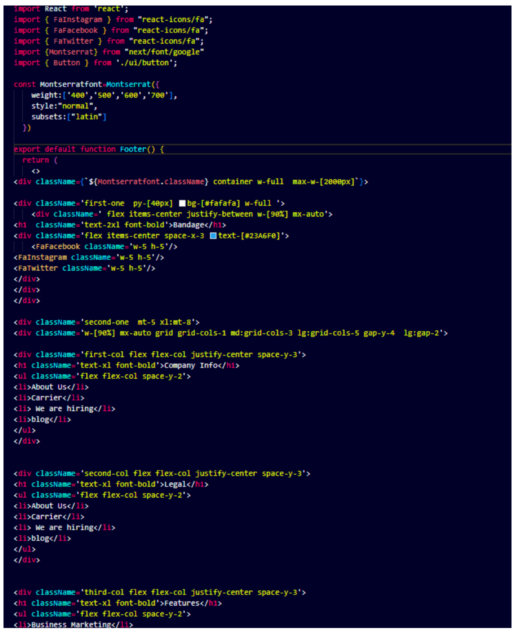
Bandage

| Company Info | Legal | Features | Resources | |
|---|---|---|---|---|
| About Us | About Us | Business Marketing | IOS & Android | Get In touch |
| Carrier | Carrier | User Analytic | Watch a Demo | Your Email Subscribe |
| We are hiring | We are hiring | Live Chat | Customers | |
| blog | blog | Unlimited Support | API | |

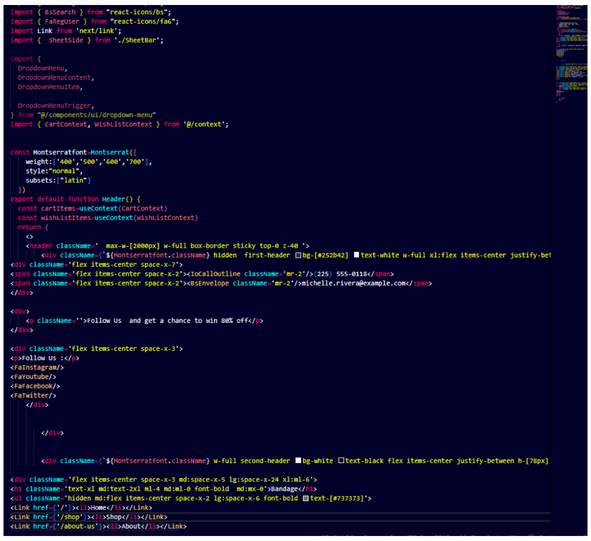
Made With Love By Finland All Right Reserved

## Code Snippet:

## Footer:


```
import React from 'react';
import { FaInstagram } from "react-icons/fa";
import { FaFacebook } from "react-icons/fa";
import { FaTwitter } from "react-icons/fa";
import {Montserrat} from "next/font/google"
import { Button } from './ui/button';

const Montserratfont=Montserrat({
    weight:['400','500','600','700'],
    style:"normal",
    subsets:["latin"]
  })

export default function Footer() {
  return (
    <>
<div className={`${Montserratfont.className} container w-full  max-w-[2000px]`}>

<div className='first-one  py-[40px] bg-[#fafafa] w-full '>
    <div className=' flex items-center justify-between w-[90%] mx-auto'>
<h1  className='text-2xl font-bold'>Bandage</h1>
<div className='flex items-center space-x-3 text-[#23A6F0]'>
    <FaFacebook className='w-5 h-5'/>
<FaInstagram className='w-5 h-5'/>
<FaTwitter className='w-5 h-5'/>
</div>
</div>
</div>

<div className='second-one  mt-5 xl:mt-8'>
<div className='w-[90%] mx-auto grid grid-cols-1 md:grid-cols-3 lg:grid-cols-5 gap-y-4  lg:gap-2'>

<div className='first-col flex flex-col justify-center space-y-3'>
<h1 className='text-xl font-bold'>Company Info</h1>
<ul className='flex flex-col space-y-2'>
<li>About Us</li>
<li>Carrier</li>
<li> We are hiring</li>
<li>blog</li>
</ul>
</div>


<div className='second-col flex flex-col justify-center space-y-3'>
<h1 className='text-xl font-bold'>Legal</h1>
<ul className='flex flex-col space-y-2'>
<li>About Us</li>
<li>Carrier</li>
<li> We are hiring</li>
<li>blog</li>
</ul>
</div>


<div className='third-col flex flex-col justify-center space-y-3'>
<h1 className='text-xl font-bold'>Features</h1>
<ul className='flex flex-col space-y-2'>
<li>Business Marketing</li>
```

## Header:



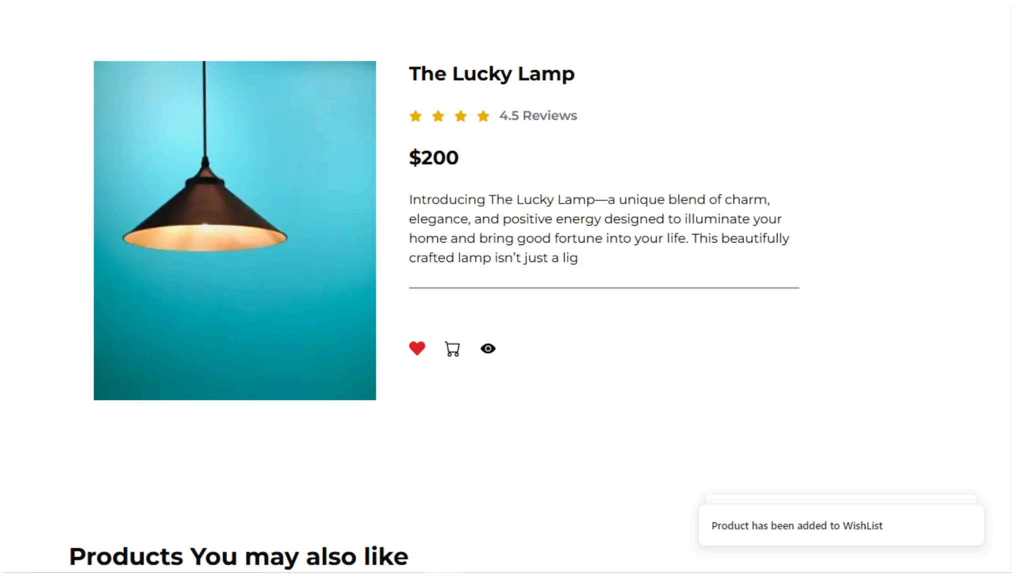
```
import { BsSearch } from "react-icons/bs";
import { FaRegUser } from "react-icons/fa6";
import Link from 'next/link';
import {  SheetSide } from './SheetBar';

import {
  DropdownMenu,
  DropdownMenuContent,
  DropdownMenuItem,

  DropdownMenuTrigger,
} from "@/components/ui/dropdown-menu"
import { CartContext, WishListContext } from '@/context';


const Montserratfont=Montserrat({
    weight:['400','500','600','700'],
    style:"normal",
    subsets:["latin"]
  })
export default function Header() {
  const cartItems=useContext(CartContext)
  const wishListItems=useContext(WishListContext)
  return (
    <>
    <header className='  max-w-[2000px] w-full box-border sticky top-0 z-40 '>
        <div className={`${Montserratfont.className} hidden  first-header bg-[#252b42] text-white w-full xl:flex items-center justify-bet
<div className='flex items-center space-x-7'>
<span className='flex items-center space-x-2'><IoCallOutline className='mr-2'/>(225) 555-0118</span>
<span className='flex items-center space-x-2'><BsEnvelope className='mr-2'/>michelle.rivera@example.com</span>
</div>

<div>
    <p className=''>Follow Us  and get a chance to win 80% off</p>
</div>

<div className='flex items-center space-x-3'>
<p>Follow Us :</p>
<FaInstagram/>
<FaYoutube/>
<FaFacebook/>
<FaTwitter/>
    </div>


        </div>


        <div className={`${Montserratfont.className} w-full second-header bg-white text-black flex items-center justify-between h-[78px]

<div className='flex items-center space-x-3 md:space-x-5 lg:space-x-24 xl:ml-6'>
<h1 className='text-xl md:text-2xl ml-4 md:ml-0 font-bold  md:mx-0'>Bandage</h1>
<ul className='hidden md:flex items-center space-x-2 lg:space-x-6 font-bold text-[#737373]'>
<Link href={'/'}><li>Home</li></Link>
<Link href={'/shop'}><li>Shop</li></Link>
<Link href={'/about-us'}><li>About</li></Link>
```

### 10. Notifications Component:

The Notifications component enhances user experience by providing real-time feedback for various actions within the application.

**Steps taken to build:**

- **Real-time Alerts:** Displays alerts for actions such as adding items to the cart, errors, or successful purchases to keep users informed.
- **Toast Notifications:** Used React Toastify for showing non-intrusive toast notifications that appear briefly on the screen.
- **Modal Windows:** Implemented modal windows for more critical or in-depth alerts, such as confirming a successful order or an error message.
- **Customizable Notifications:** Enabled customization of notification types (success, error, info) and duration, giving users clear, actionable feedback.
- **Accessibility:** Ensured notifications are accessible by providing screen reader support and making sure they don't interrupt the user experience.



The Lucky Lamp

★ ★ ★ ★ 4.5 Reviews

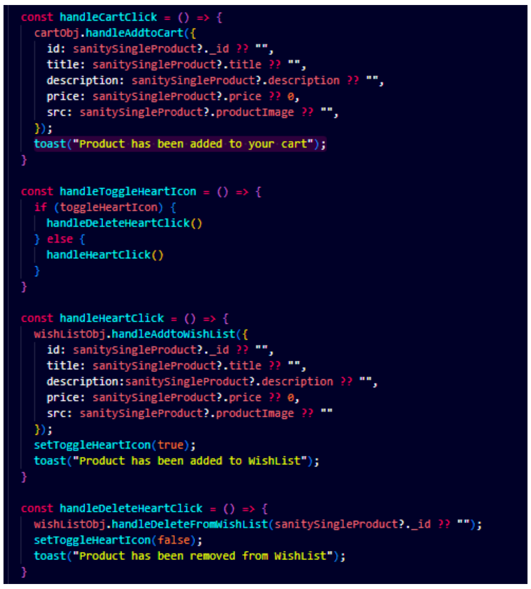
$200

Introducing The Lucky Lamp—a unique blend of charm, elegance, and positive energy designed to illuminate your home and bring good fortune into your life. This beautifully crafted lamp isn't just a lig

Product has been added to WishList

Products You may also like

**Code Snippet:**



```
const handleCartClick = () => {
  cartObj.handleAddtoCart({
    id: sanitySingleProduct?._id ?? "",
    title: sanitySingleProduct?.title ?? "",
    description: sanitySingleProduct?.description ?? "",
    price: sanitySingleProduct?.price ?? 0,
    src: sanitySingleProduct?.productImage ?? "",
  });
  toast("Product has been added to your cart");
}

const handleToggleHeartIcon = () => {
  if (toggleHeartIcon) {
    handleDeleteHeartClick()
  } else {
    handleHeartClick()
  }
}

const handleHeartClick = () => {
  wishListObj.handleAddtoWishList({
    id: sanitySingleProduct?._id ?? "",
    title: sanitySingleProduct?.title ?? "",
    description:sanitySingleProduct?.description ?? "",
    price: sanitySingleProduct?.price ?? 0,
    src: sanitySingleProduct?.productImage ?? ""
  });
  setToggleHeartIcon(true);
  toast("Product has been added to WishList");
}

const handleDeleteHeartClick = () => {
  wishListObj.handleDeleteFromWishList(sanitySingleProduct?._id ?? "");
  setToggleHeartIcon(false);
  toast("Product has been removed from WishList");
}
```

## Self-Validation Checklist:

| Task | Status |
|---|---|
| FrontEnd components | ✓ |
| Styling and responsiveness | ✓ |
| Code Quality | ✓ |
| Documentation and submission | ✓ |
| Final Review | ✓ |