Day 4 - Building Dynamic Frontend Components for Your Marketplace Assignment

Title:

Day 4 - Dynamic Frontend Components - Furniture Marketplace

Introduction

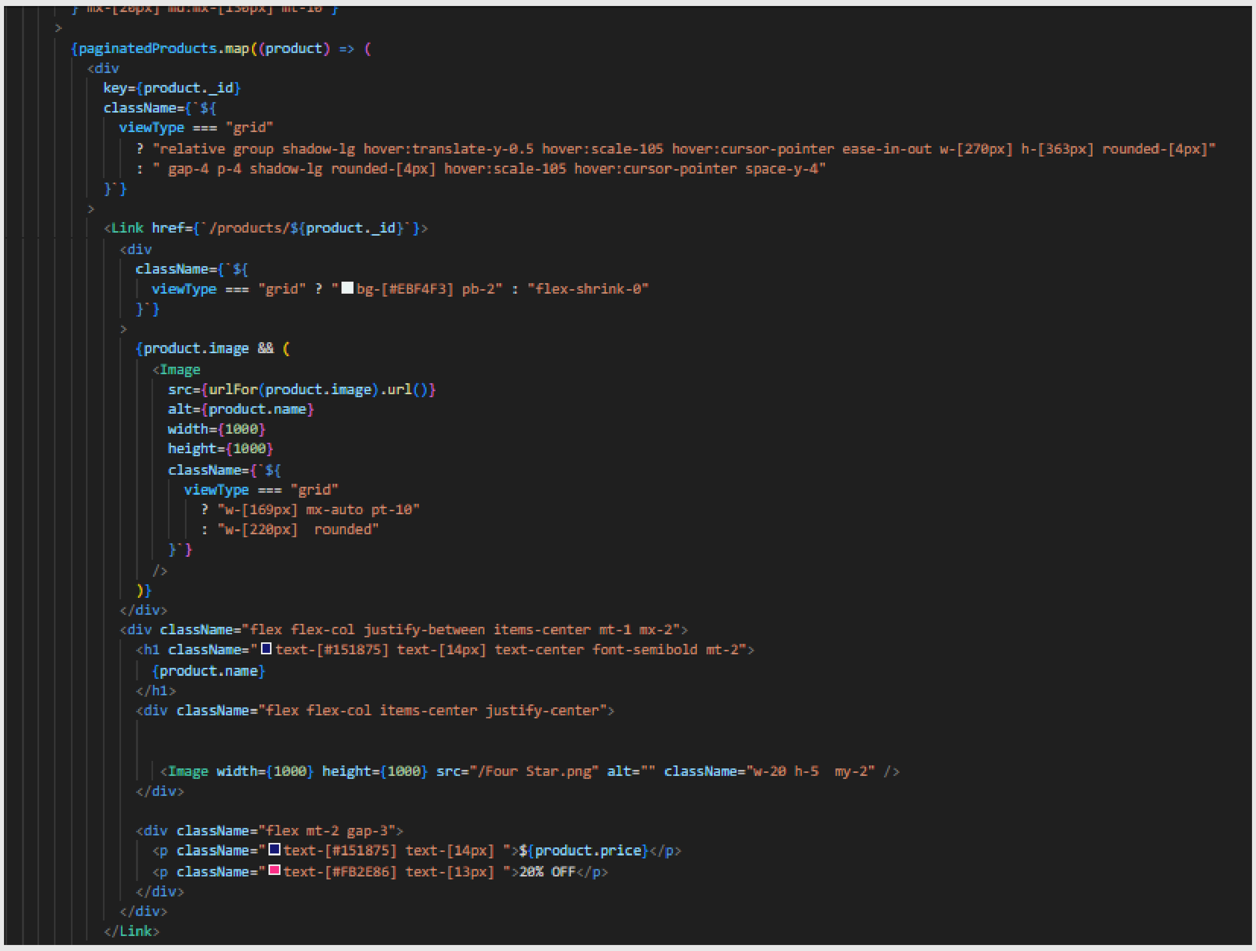
The objective of this assignment is to design and develop dynamic frontend components for a marketplace. The focus is on modular, reusable component design, responsive interfaces, and the integration of data fetched from Sanity CMS or APIs. This report will document the creation process, challenges faced, and solutions implemented.

1. Components I Built

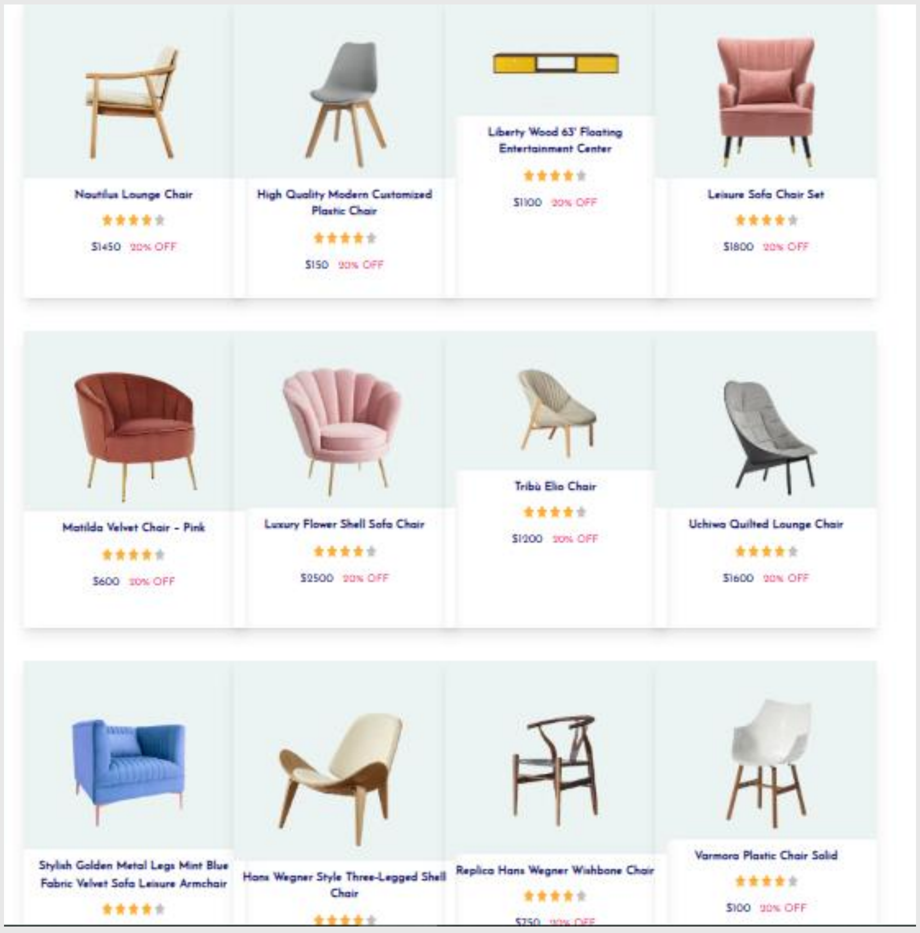
1. Product Listing Component:

- Rendered product data dynamically in a grid layout.
- Fields included:
  - Product Name
  - Price
  - Image
  - Review

- Layout: Cards displaying product detail



```
  } mx-[20px] md:mx-[150px] mt-10 }
  >
   {paginatedProducts.map((product) => (
     <div
       key={product._id}
       className={`${
         viewType === "grid"
           ? "relative group shadow-lg hover:translate-y-0.5 hover:scale-105 hover:cursor-pointer ease-in-out w-[270px] h-[363px] rounded-[4px]"
           : " gap-4 p-4 shadow-lg rounded-[4px] hover:scale-105 hover:cursor-pointer space-y-4"
       }`}
     >
       <Link href={`/products/${product._id}`}>
         <div
           className={`${
             viewType === "grid" ? "bg-[#EBF4F3] pb-2" : "flex-shrink-0"
           }`}
         >
           {product.image && (
             <Image
               src={urlFor(product.image).url()}
               alt={product.name}
               width={1000}
               height={1000}
               className={`${
                 viewType === "grid"
                   ? "w-[169px] mx-auto pt-10"
                   : "w-[220px]  rounded"
               }`}
             />
           )}
         </div>
         <div className="flex flex-col justify-between items-center mt-1 mx-2">
           <h1 className="text-[#151875] text-[14px] text-center font-semibold mt-2">
             {product.name}
           </h1>
           <div className="flex flex-col items-center justify-center">


             <Image width={1000} height={1000} src="/Four Star.png" alt="" className="w-20 h-5  my-2" />
           </div>

           <div className="flex mt-2 gap-3">
             <p className="text-[#151875] text-[14px] ">${product.price}</p>
             <p className="text-[#FB2E86] text-[13px] ">20% OFF</p>
           </div>
         </div>
       </Link>
```

- 

Nautilus Lounge Chair
$1450 20% OFF

High Quality Modern Customized Plastic Chair
$150 20% OFF

Liberty Wood 63' Floating Entertainment Center
$1100 20% OFF

Leisure Sofa Chair Set
$1800 20% OFF

Matilda Velvet Chair - Pink
$600 20% OFF

Luxury Flower Shell Sofa Chair
$2500 20% OFF

Tribù Elio Chair
$1200 20% OFF

Uchiwa Quilted Lounge Chair
$1600 20% OFF

Stylish Golden Metal Legs Mint Blue Fabric Velvet Sofa Leisure Armchair

Hans Wegner Style Three-Legged Shell Chair

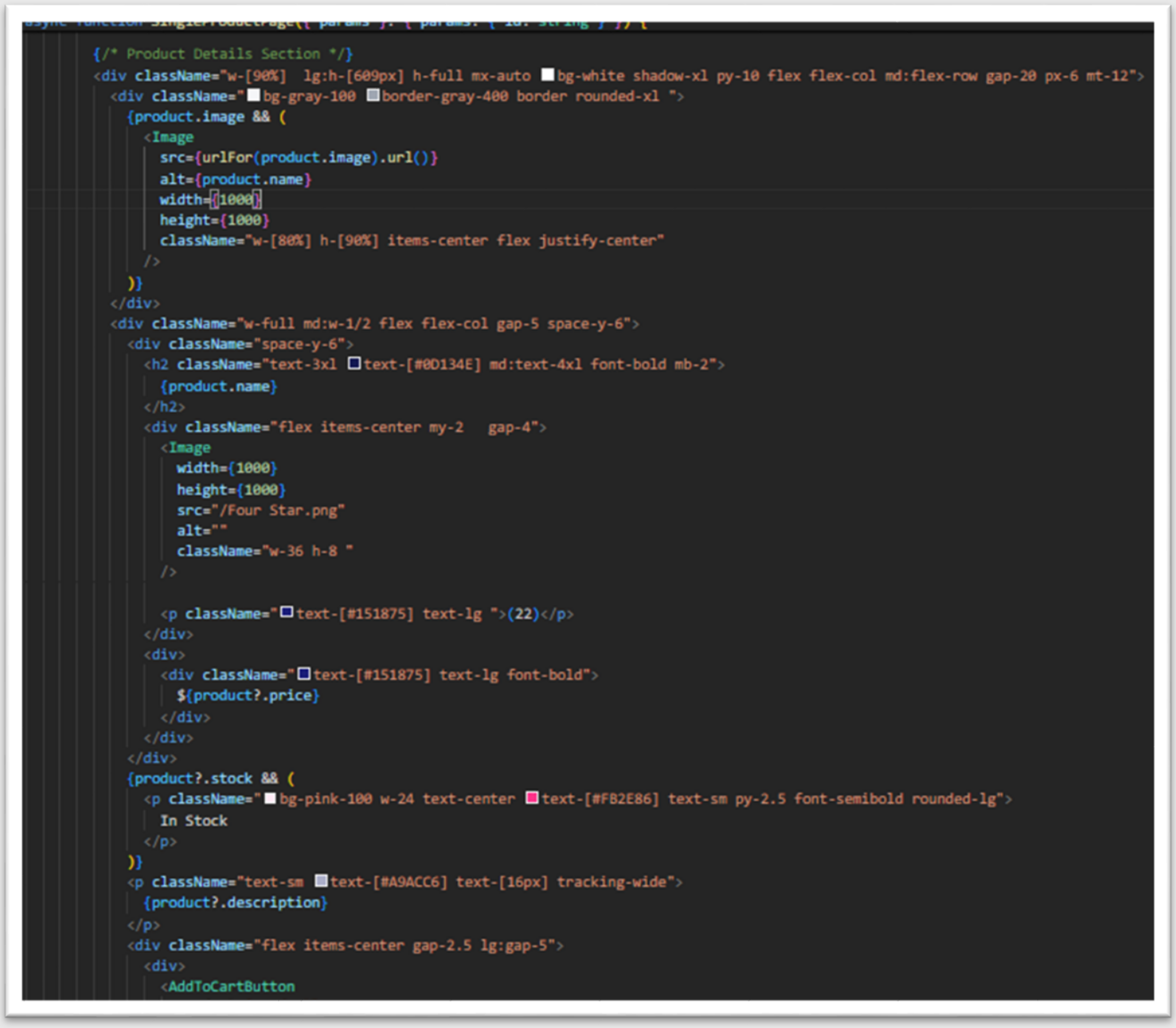
Replica Hans Wegner Wishbone Chair
$750 20% OFF

Varmora Plastic Chair Solid
$100 20% OFF

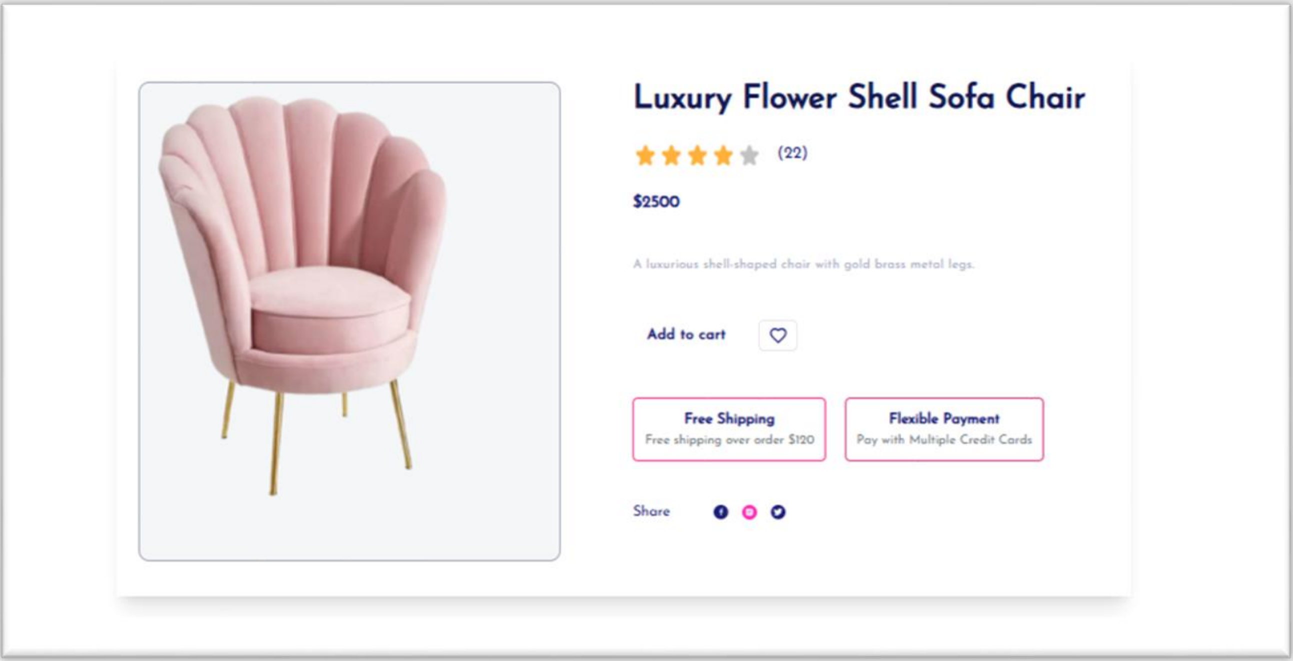
2. Product Detail Component:

- Created individual product detail pages using dynamic routing in Next.js.
- Fields included:
  - Product Description
  - Product Name
  - Price

- Review

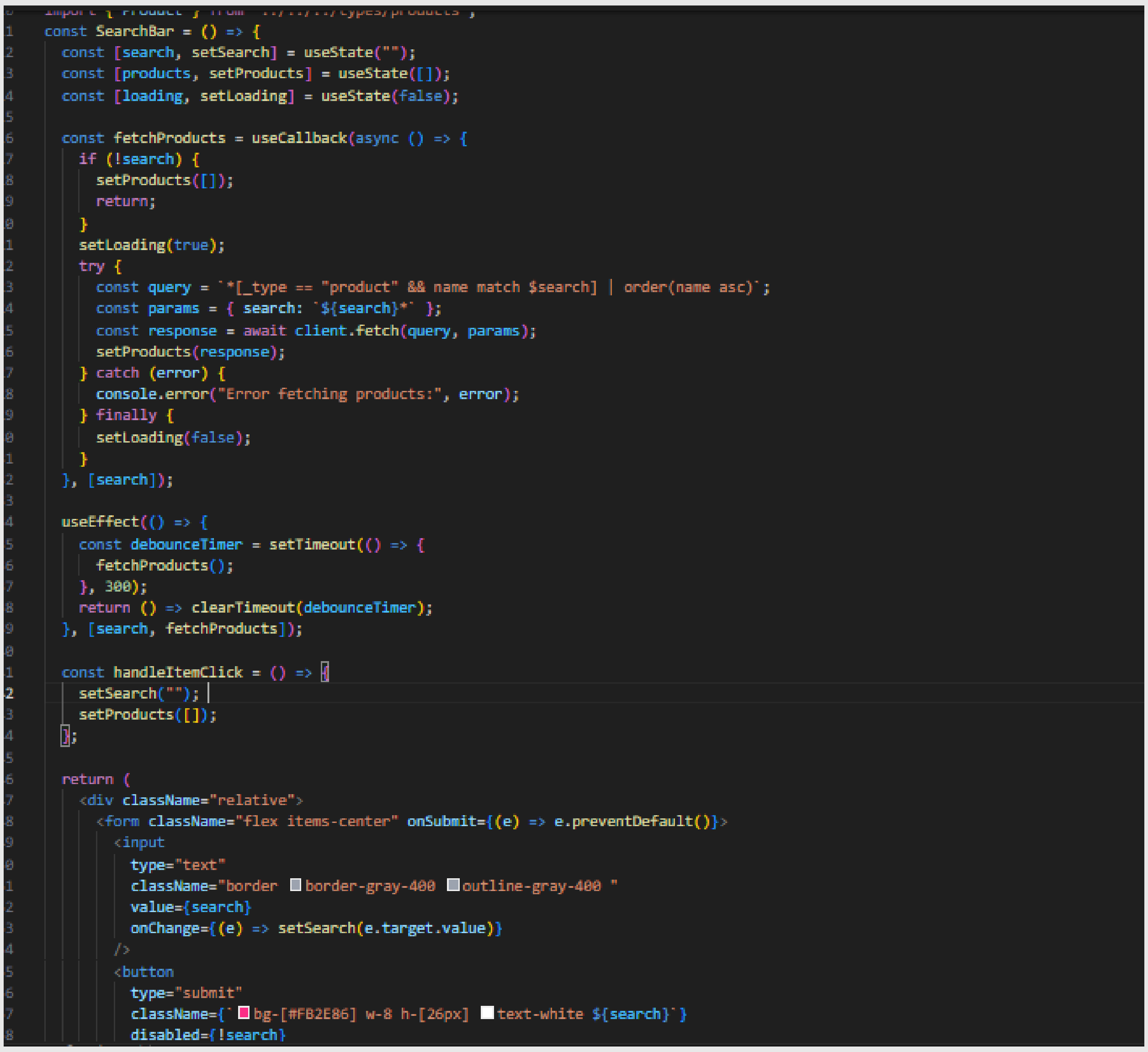


```
      {/* Product Details Section */}
      <div className="w-[90%]  lg:h-[609px] h-full mx-auto bg-white shadow-xl py-10 flex flex-col md:flex-row gap-20 px-6 mt-12">
        <div className="bg-gray-100 border-gray-400 border rounded-xl ">
          {product.image && (
            <Image
              src={urlFor(product.image).url()}
              alt={product.name}
              width={1000}
              height={1000}
              className="w-[80%] h-[90%] items-center flex justify-center"
            />
          )}
        </div>
        <div className="w-full md:w-1/2 flex flex-col gap-5 space-y-6">
          <div className="space-y-6">
            <h2 className="text-3xl text-[#0D134E] md:text-4xl font-bold mb-2">
              {product.name}
            </h2>
            <div className="flex items-center my-2   gap-4">
              <Image
                width={1000}
                height={1000}
                src="/Four Star.png"
                alt=""
                className="w-36 h-8 "
              />

              <p className="text-[#151875] text-lg ">(22)</p>
            </div>
            <div>
              <div className="text-[#151875] text-lg font-bold">
                ${product?.price}
              </div>
            </div>
          </div>
          {product?.stock && (
            <p className="bg-pink-100 w-24 text-center text-[#FB2E86] text-sm py-2.5 font-semibold rounded-lg">
              In Stock
            </p>
          )}
          <p className="text-sm text-[#A9ACC6] text-[16px] tracking-wide">
            {product?.description}
          </p>
          <div className="flex items-center gap-2.5 lg:gap-5">
            <div>
              <AddToCartButton
```

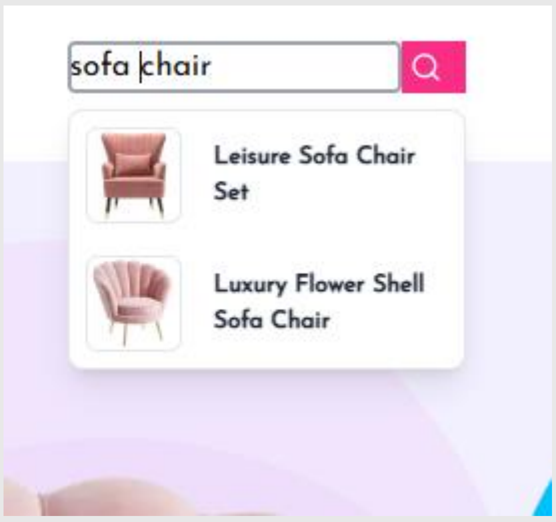

Luxury Flower Shell Sofa Chair
(22)
$2500
A luxurious shell-shaped chair with gold brass metal legs.
Add to cart
Free Shipping
Free shipping over order $120
Flexible Payment
Pay with Multiple Credit Cards
Share

3. Search Bar:

Implemented search functionality to filter products by name.



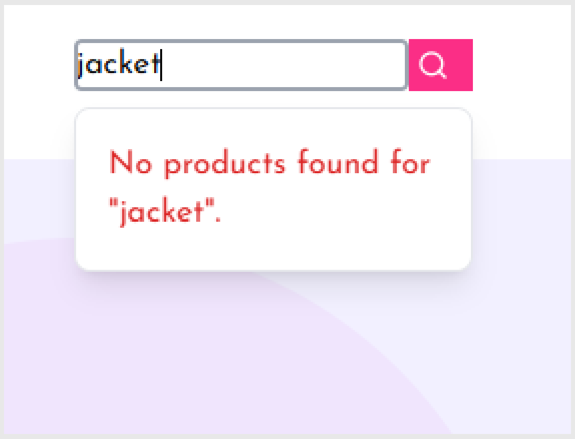
```
const SearchBar = () => {
  const [search, setSearch] = useState("");
  const [products, setProducts] = useState([]);
  const [loading, setLoading] = useState(false);

  const fetchProducts = useCallback(async () => {
    if (!search) {
      setProducts([]);
      return;
    }
    setLoading(true);
    try {
      const query = `*[_type == "product" && name match $search] | order(name asc)`;
      const params = { search: `${search}*` };
      const response = await client.fetch(query, params);
      setProducts(response);
    } catch (error) {
      console.error("Error fetching products:", error);
    } finally {
      setLoading(false);
    }
  }, [search]);

  useEffect(() => {
    const debounceTimer = setTimeout(() => {
      fetchProducts();
    }, 300);
    return () => clearTimeout(debounceTimer);
  }, [search, fetchProducts]);

  const handleItemClick = () => {
    setSearch("");
    setProducts([]);
  };

  return (
    <div className="relative">
      <form className="flex items-center" onSubmit={(e) => e.preventDefault()}>
        <input
          type="text"
          className="border border-gray-400 outline-gray-400 "
          value={search}
          onChange={(e) => setSearch(e.target.value)}
        />
        <button
          type="submit"
          className={` bg-[#FB2E86] w-8 h-[26px] text-white ${search}`}
          disabled={!search}
```



sofa chair

Leisure Sofa Chair Set

Luxury Flower Shell Sofa Chair

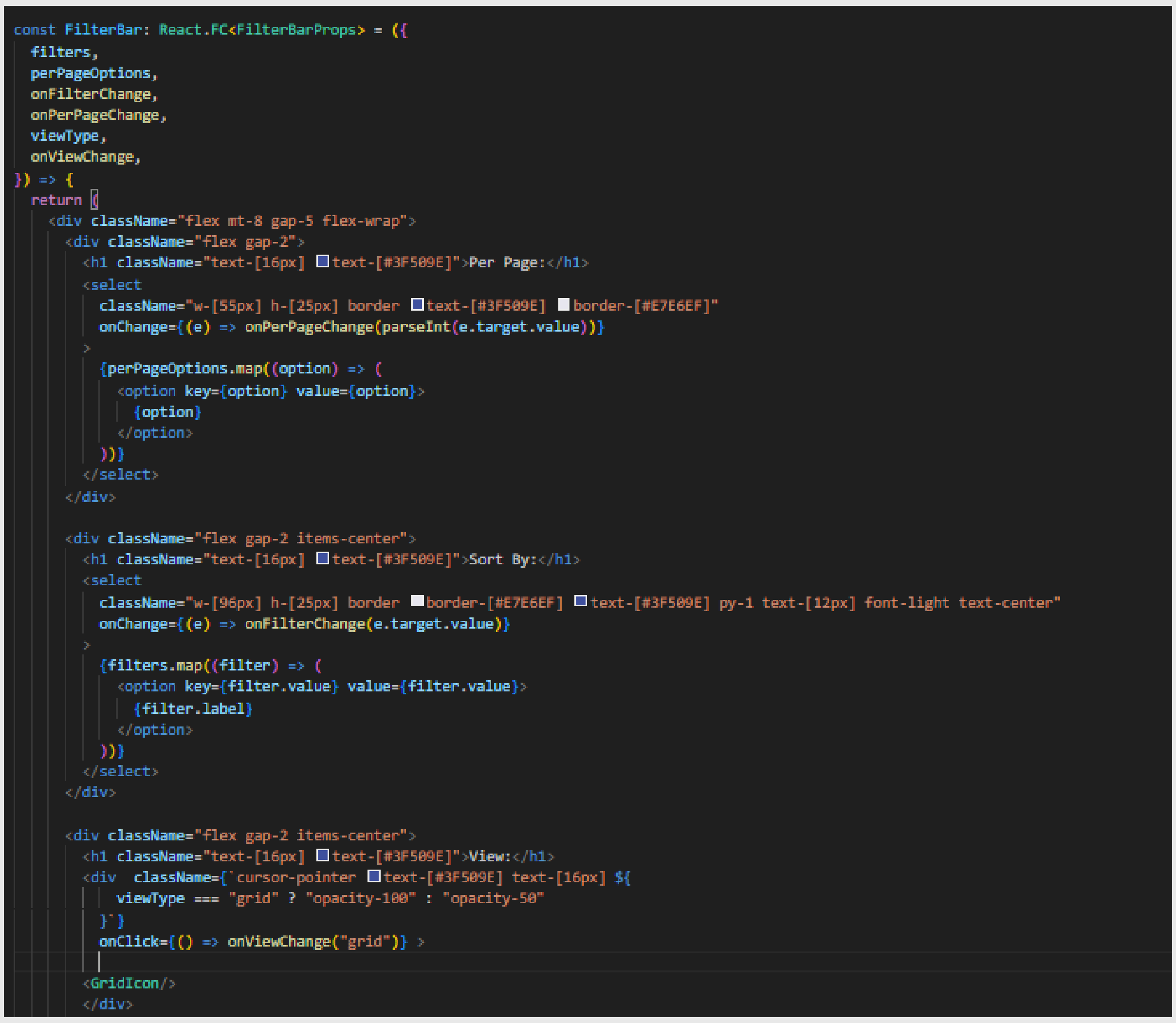


jacket

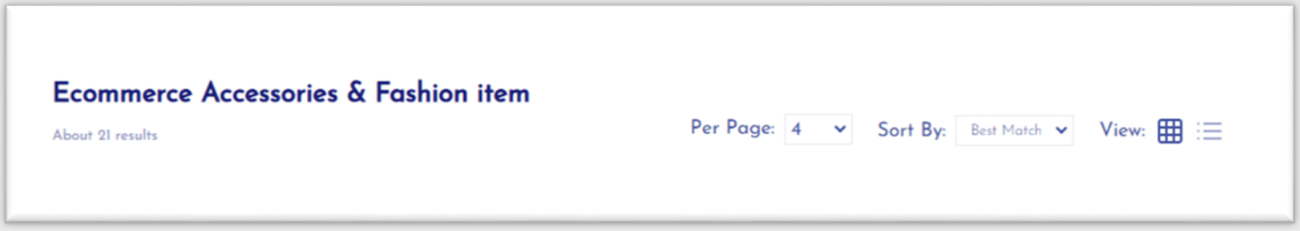
No products found for "jacket".

4. Filter Bar Component:

- Added a filter bar to refine product listings based on price ranges and availability.

- 

```
const FilterBar: React.FC<FilterBarProps> = ({
  filters,
  perPageOptions,
  onFilterChange,
  onPerPageChange,
  viewType,
  onViewChange,
}) => {
  return (
    <div className="flex mt-8 gap-5 flex-wrap">
      <div className="flex gap-2">
        <h1 className="text-[16px] text-[#3F509E]">Per Page:</h1>
        <select
          className="w-[55px] h-[25px] border text-[#3F509E] border-[#E7E6EF]"
          onChange={(e) => onPerPageChange(parseInt(e.target.value))}
        >
          {perPageOptions.map((option) => (
            <option key={option} value={option}>
              {option}
            </option>
          ))}
        </select>
      </div>

      <div className="flex gap-2 items-center">
        <h1 className="text-[16px] text-[#3F509E]">Sort By:</h1>
        <select
          className="w-[96px] h-[25px] border border-[#E7E6EF] text-[#3F509E] py-1 text-[12px] font-light text-center"
          onChange={(e) => onFilterChange(e.target.value)}
        >
          {filters.map((filter) => (
            <option key={filter.value} value={filter.value}>
              {filter.label}
            </option>
          ))}
        </select>
      </div>

      <div className="flex gap-2 items-center">
        <h1 className="text-[16px] text-[#3F509E]">View:</h1>
        <div  className={`cursor-pointer text-[#3F509E] text-[16px] ${
            viewType === "grid" ? "opacity-100" : "opacity-50"
          }`}
          onClick={() => onViewChange("grid")} >
          
        <GridIcon/>
        </div>
```



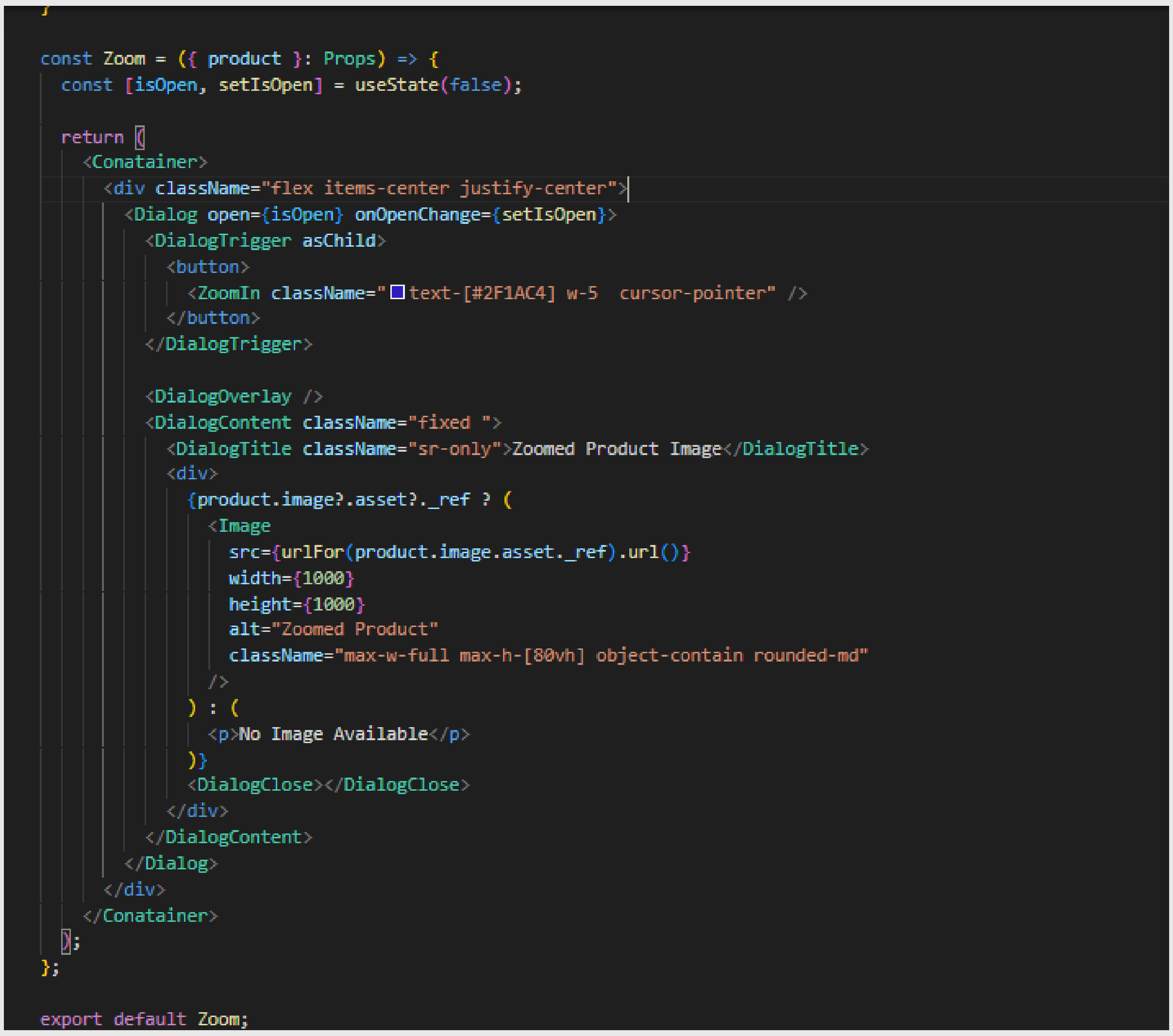
Ecommerce Accessories & Fashion item

About 21 results

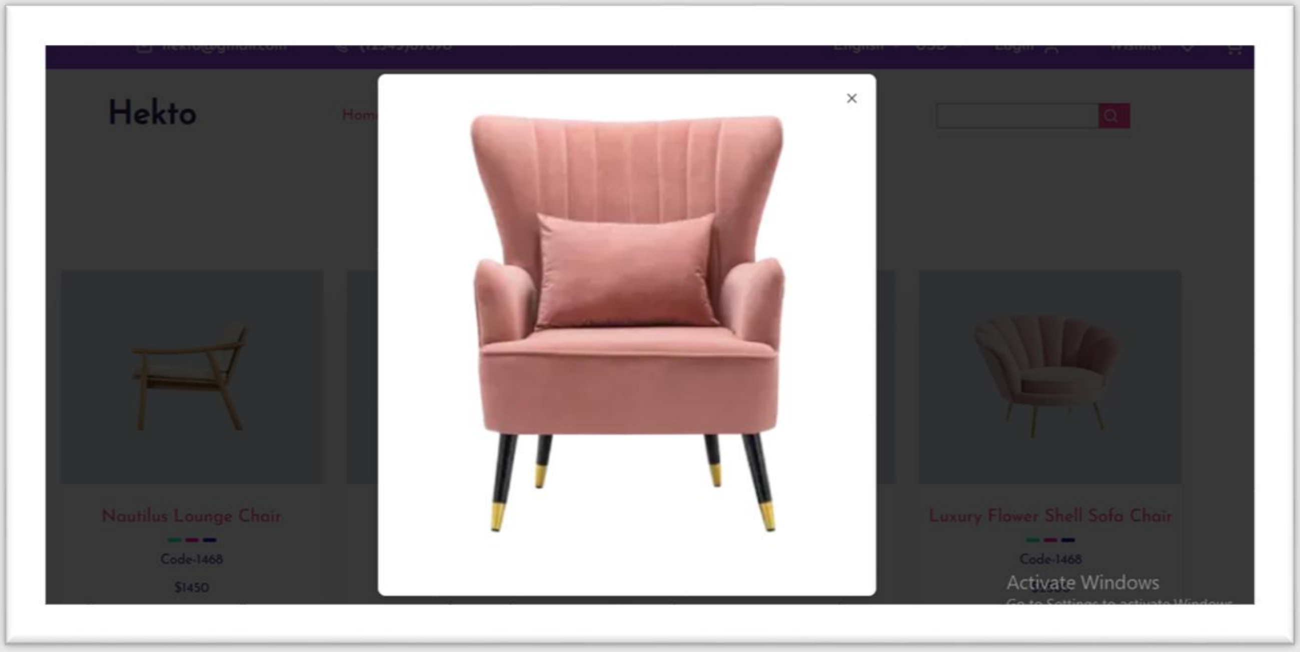
Per Page: 4 Sort By: Best Match View:

5. Zoom Product Feature:

- Implemented a zoom feature on the product image for enhanced user experience.

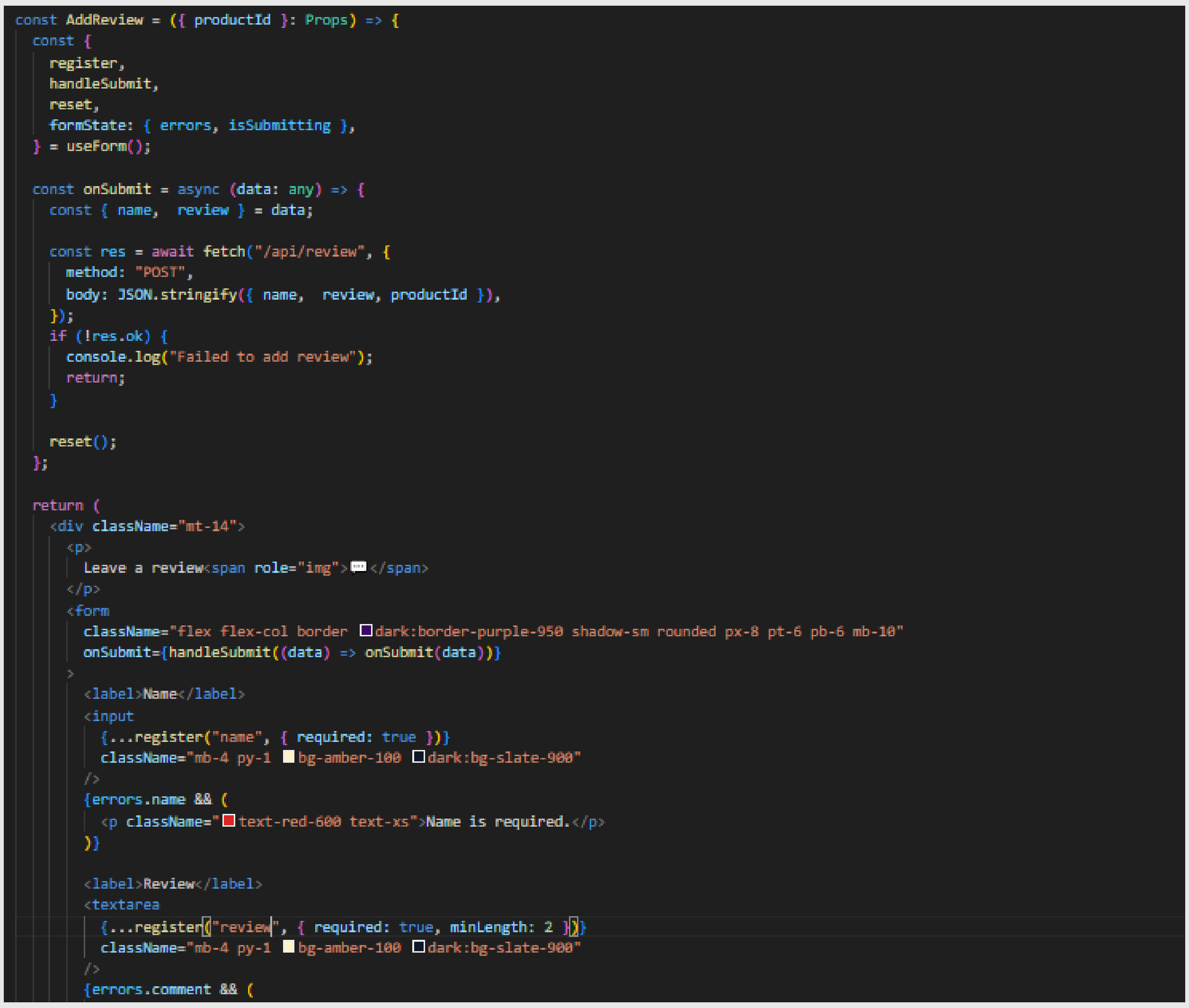
- 

```
const Zoom = ({ product }: Props) => {
  const [isOpen, setIsOpen] = useState(false);

  return (
    <Conatainer>
      <div className="flex items-center justify-center">
        <Dialog open={isOpen} onOpenChange={setIsOpen}>
          <DialogTrigger asChild>
            <button>
              <ZoomIn className="text-[#2F1AC4] w-5  cursor-pointer" />
            </button>
          </DialogTrigger>

          <DialogOverlay />
          <DialogContent className="fixed ">
            <DialogTitle className="sr-only">Zoomed Product Image</DialogTitle>
            <div>
              {product.image?.asset?._ref ? (
                <Image
                  src={urlFor(product.image.asset._ref).url()}
                  width={1000}
                  height={1000}
                  alt="Zoomed Product"
                  className="max-w-full max-h-[80vh] object-contain rounded-md"
                />
              ) : (
                <p>No Image Available</p>
              )}
              <DialogClose></DialogClose>
            </div>
          </DialogContent>
        </Dialog>
      </div>
    </Conatainer>
  );
};

export default Zoom;
```

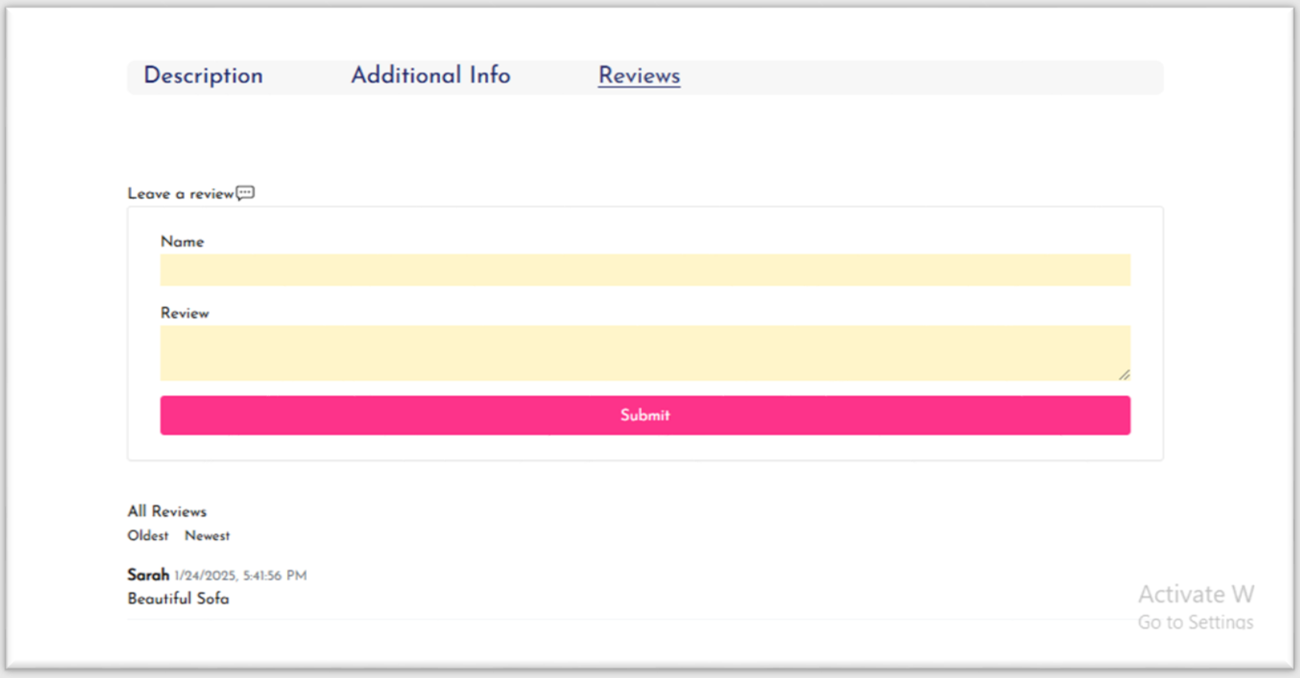


6. Add Reviews Component:

- Reviews for each product, allowing users to view and submit feedback.



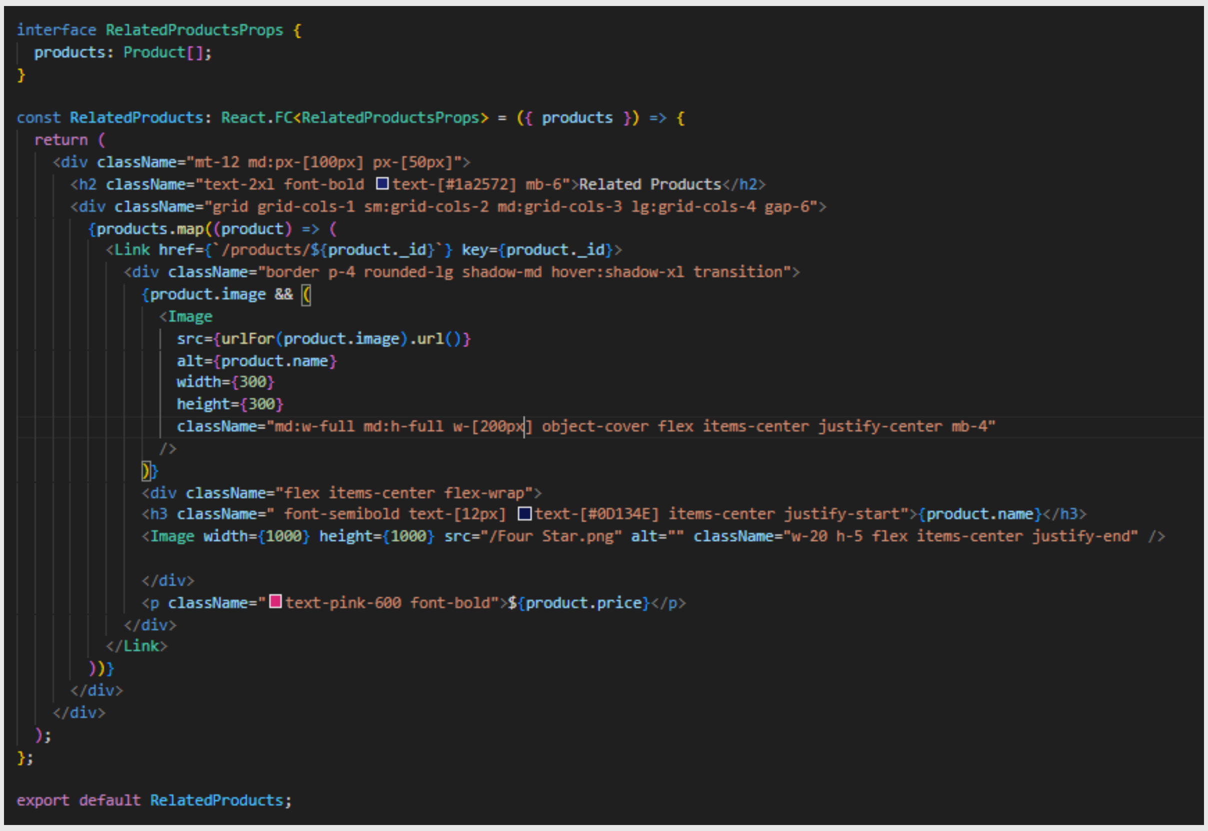
```
const AddReview = ({ productId }: Props) => {
  const {
    register,
    handleSubmit,
    reset,
    formState: { errors, isSubmitting },
  } = useForm();

  const onSubmit = async (data: any) => {
    const { name,  review } = data;

    const res = await fetch("/api/review", {
      method: "POST",
      body: JSON.stringify({ name,  review, productId }),
    });
    if (!res.ok) {
      console.log("Failed to add review");
      return;
    }

    reset();
  };

  return (
    <div className="mt-14">
      <p>
        Leave a review<span role="img">💬</span>
      </p>
      <form
        className="flex flex-col border dark:border-purple-950 shadow-sm rounded px-8 pt-6 pb-6 mb-10"
        onSubmit={handleSubmit((data) => onSubmit(data))}
      >
        <label>Name</label>
        <input
          {...register("name", { required: true })}
          className="mb-4 py-1 bg-amber-100 dark:bg-slate-900"
        />
        {errors.name && (
          <p className="text-red-600 text-xs">Name is required.</p>
        )}

        <label>Review</label>
        <textarea
          {...register("review", { required: true, minLength: 2 })}
          className="mb-4 py-1 bg-amber-100 dark:bg-slate-900"
        />
        {errors.comment && (
```



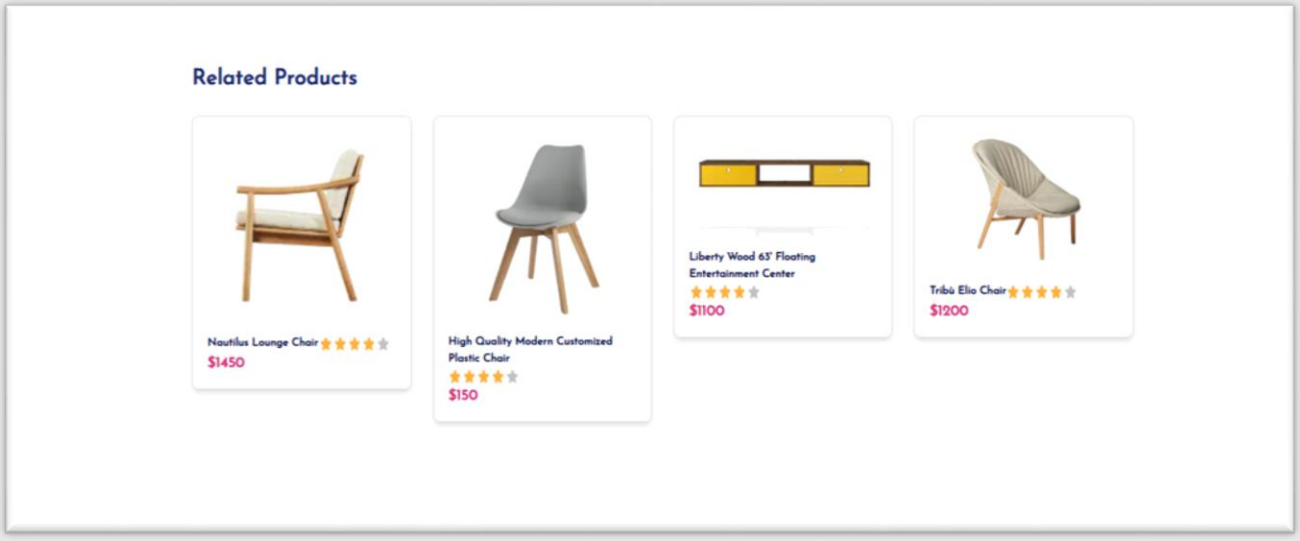
Sarah 1/24/2025, 5:41:56 PM
Beautiful Sofa

7. Related Products Component:

- Suggested similar or complementary products on the product detail page.

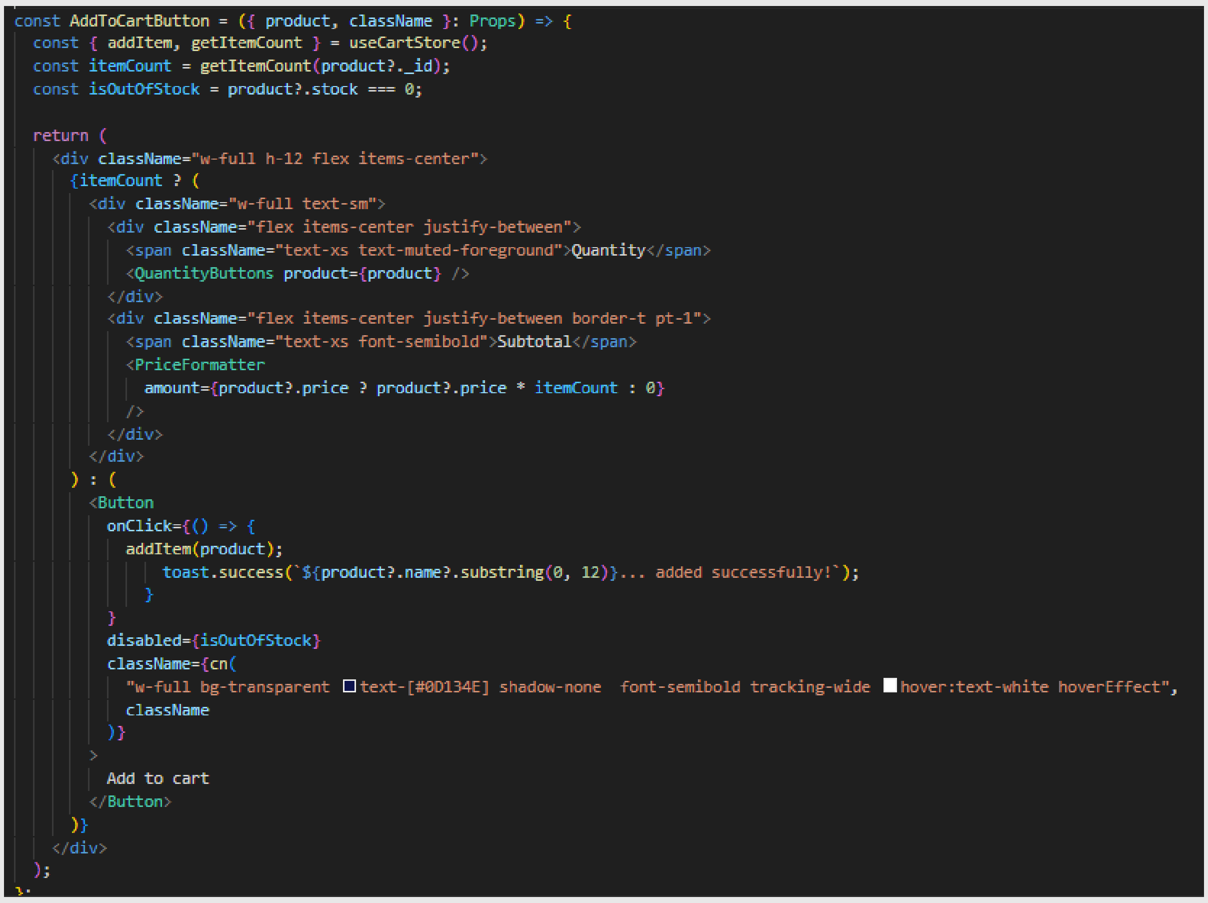
- 

```tsx
interface RelatedProductsProps {
  products: Product[];
}

const RelatedProducts: React.FC<RelatedProductsProps> = ({ products }) => {
  return (
    <div className="mt-12 md:px-[100px] px-[50px]">
      <h2 className="text-2xl font-bold text-[#1a2572] mb-6">Related Products</h2>
      <div className="grid grid-cols-1 sm:grid-cols-2 md:grid-cols-3 lg:grid-cols-4 gap-6">
        {products.map((product) => (
          <Link href={`/products/${product._id}`} key={product._id}>
            <div className="border p-4 rounded-lg shadow-md hover:shadow-xl transition">
              {product.image && (
                <Image
                  src={urlFor(product.image).url()}
                  alt={product.name}
                  width={300}
                  height={300}
                  className="md:w-full md:h-full w-[200px] object-cover flex items-center justify-center mb-4"
                />
              )}
              <div className="flex items-center flex-wrap">
              <h3 className=" font-semibold text-[12px] text-[#0D134E] items-center justify-start">{product.name}</h3>
              <Image width={1000} height={1000} src="/Four Star.png" alt="" className="w-20 h-5 flex items-center justify-end" />

              </div>
              <p className="text-pink-600 font-bold">${product.price}</p>
            </div>
          </Link>
        ))}
      </div>
    </div>
  );
};

export default RelatedProducts;
```

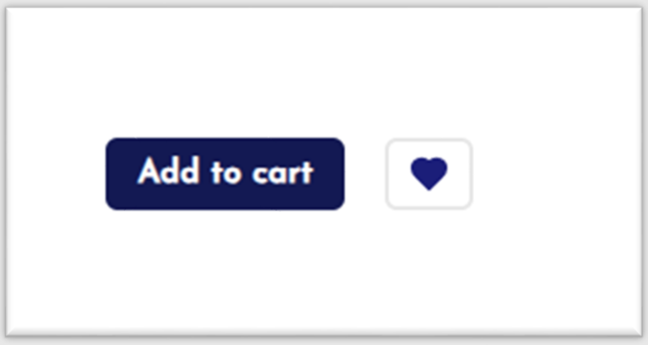

Related Products
Nautilus Lounge Chair
$1450
High Quality Modern Customized Plastic Chair
$150
Liberty Wood 63' Floating Entertainment Center
$1100
Tribù Elio Chair
$1200

8. Add to Cart Component:

- Enabled users to add products to their cart.

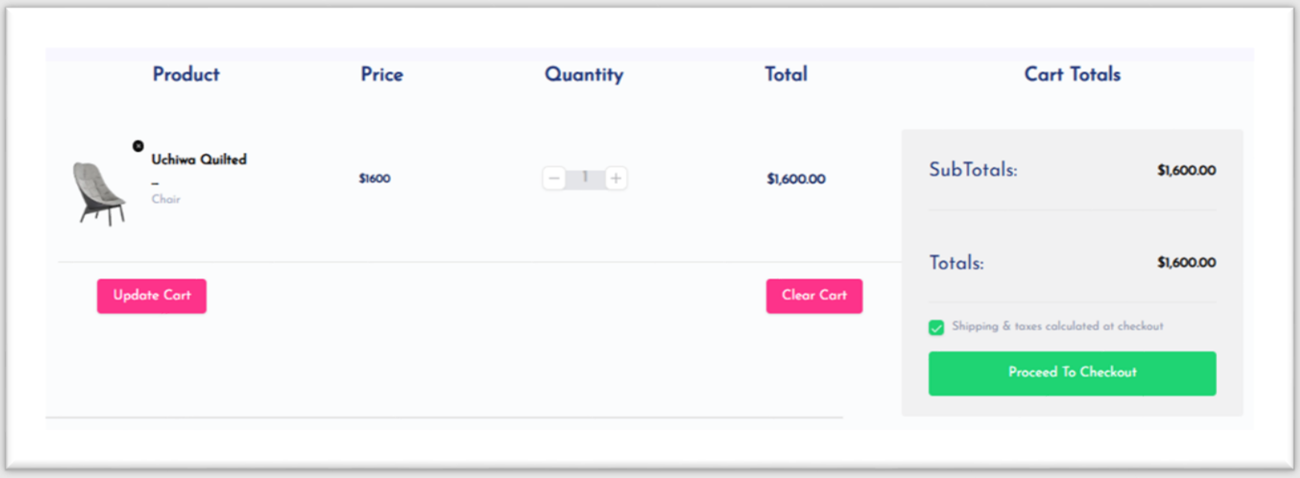
- 

```
const AddToCartButton = ({ product, className }: Props) => {
  const { addItem, getItemCount } = useCartStore();
  const itemCount = getItemCount(product?._id);
  const isOutOfStock = product?.stock === 0;

  return (
    <div className="w-full h-12 flex items-center">
      {itemCount ? (
        <div className="w-full text-sm">
          <div className="flex items-center justify-between">
            <span className="text-xs text-muted-foreground">Quantity</span>
            <QuantityButtons product={product} />
          </div>
          <div className="flex items-center justify-between border-t pt-1">
            <span className="text-xs font-semibold">Subtotal</span>
            <PriceFormatter
              amount={product?.price ? product?.price * itemCount : 0}
            />
          </div>
        </div>
      ) : (
        <Button
          onClick={() => {
            addItem(product);
              toast.success(`${product?.name?.substring(0, 12)}... added successfully!`);
            }
          }
          disabled={isOutOfStock}
          className={cn(
            "w-full bg-transparent text-[#0D134E] shadow-none  font-semibold tracking-wide hover:text-white hoverEffect",
            className
          )}
        >
          Add to cart
        </Button>
      )}
    </div>
  );
};
```



9. Delete from Cart Component:

- Added functionality to remove items from the cart.



Product	Price	Quantity	Total
Uchiwa Quilted – Chair	$1600	1	$1,600.00

Cart Totals

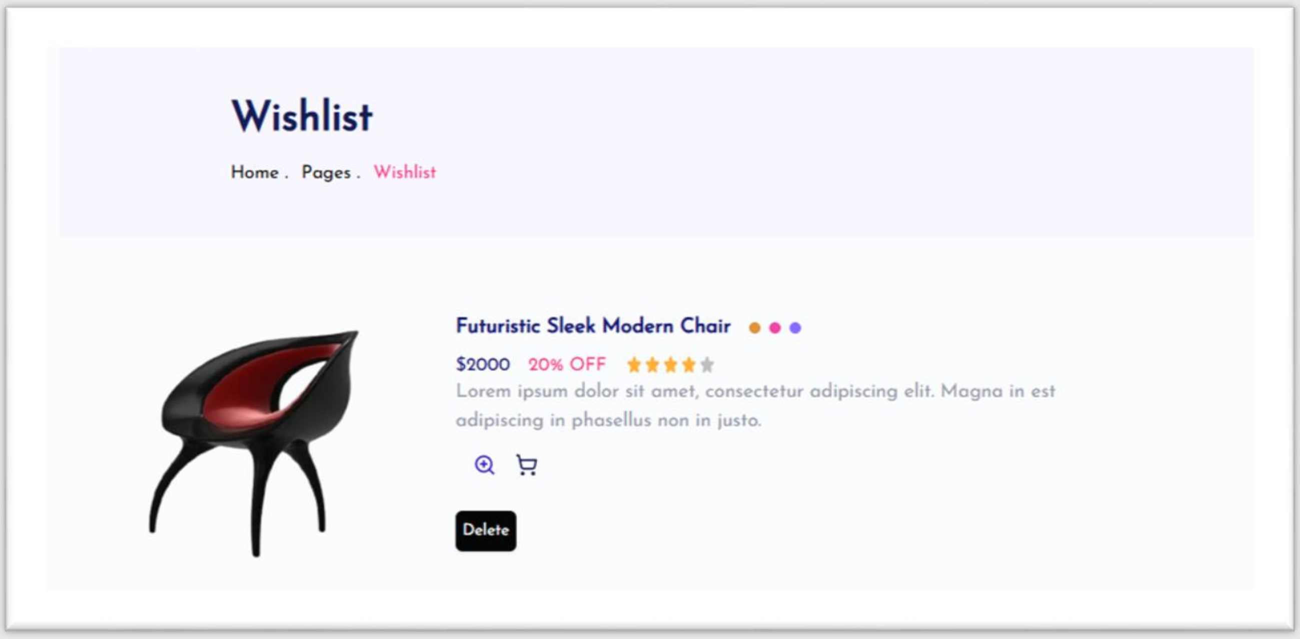
SubTotals: $1,600.00

Totals: $1,600.00

Shipping & taxes calculated at checkout

10. Add to Wishlist Component:

- Enabled users to add their favorite products in wishlist.


Wishlist

Home . Pages . Wishlist

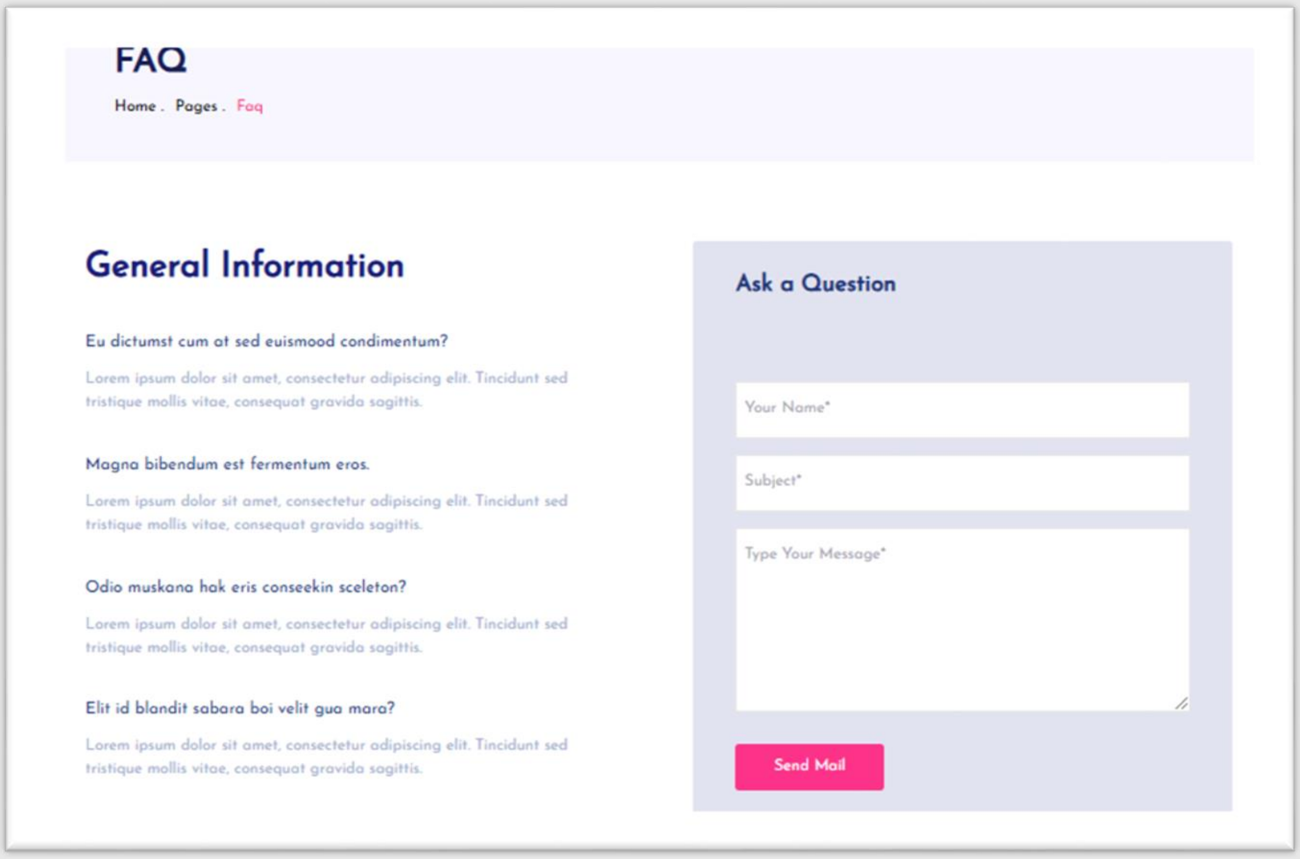
Futuristic Sleek Modern Chair

$2000 20% OFF

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Magna in est adipiscing in phasellus non in justo.

11. FAQ and Help Center Component:

- FAQ and Contact form.


FAQ

Home . Pages . Faq

General Information

Eu dictumst cum at sed euismood condimentum?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt sed tristique mollis vitae, consequat gravida sagittis.

Magna bibendum est fermentum eros.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt sed tristique mollis vitae, consequat gravida sagittis.

Odio muskana hak eris conseekin sceleton?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt sed tristique mollis vitae, consequat gravida sagittis.

Elit id blandit sabara boi velit gua mara?

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Tincidunt sed tristique mollis vitae, consequat gravida sagittis.

Ask a Question

Your Name*

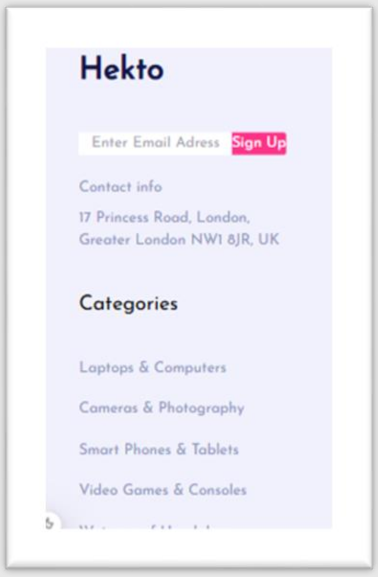
Subject*

Type Your Message*

Send Mail

12. Header and Footer Component:

Reusable and responsive header and footer component.


Hekto

Enter Email Adress Sign Up

Contact info

17 Princess Road, London, Greater London NW1 8JR, UK

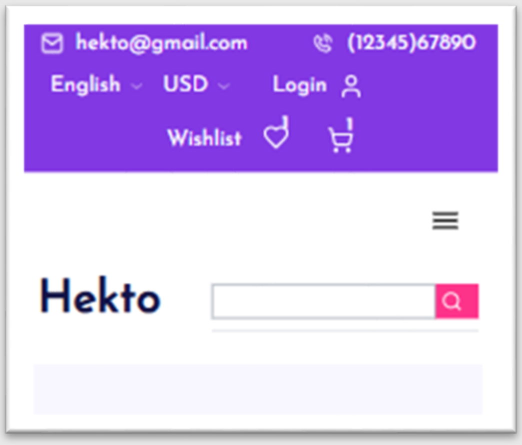
Categories

Laptops & Computers

Cameras & Photography

Smart Phones & Tablets

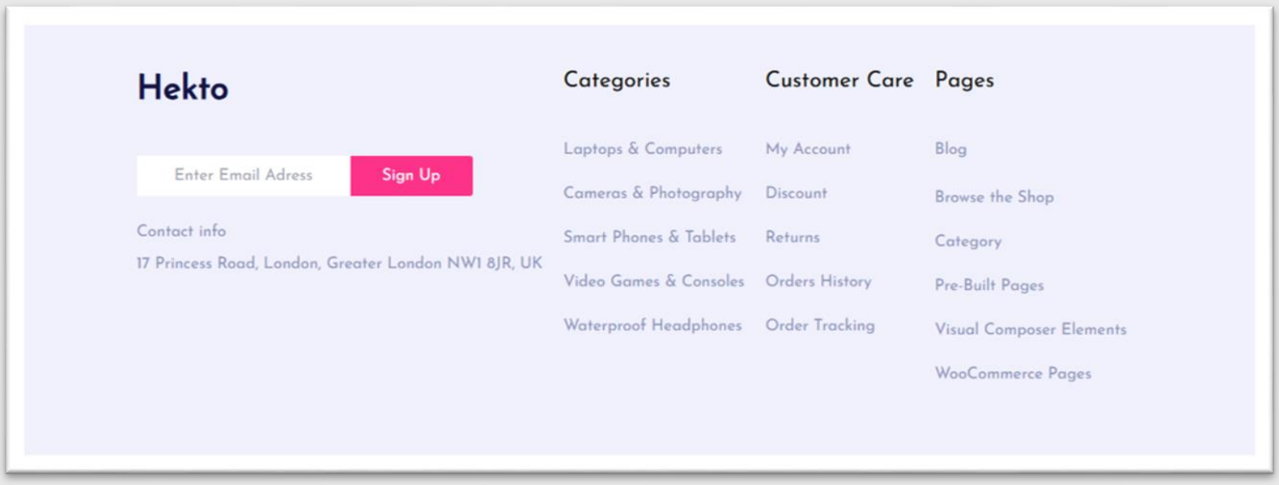
Video Games & Consoles


hekto@gmail.com (12345)67890

English USD Login

Wishlist

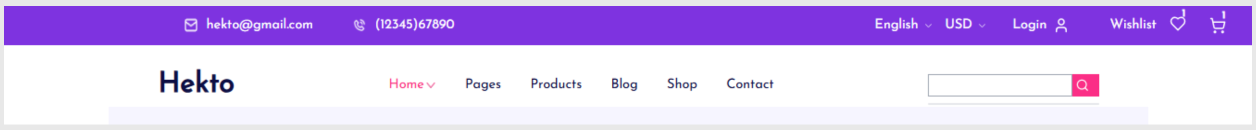
Hekto


Hekto

Enter Email Adress | Sign Up

Contact info
17 Princess Road, London, Greater London NW1 8JR, UK

Categories	Customer Care	Pages
Laptops & Computers	My Account	Blog
Cameras & Photography	Discount	Browse the Shop
Smart Phones & Tablets	Returns	Category
Video Games & Consoles	Orders History	Pre-Built Pages
Waterproof Headphones	Order Tracking	Visual Composer Elements
		WooCommerce Pages


hekto@gmail.com (12345)67890 English USD Login Wishlist

Hekto Home Pages Products Blog Shop Contact

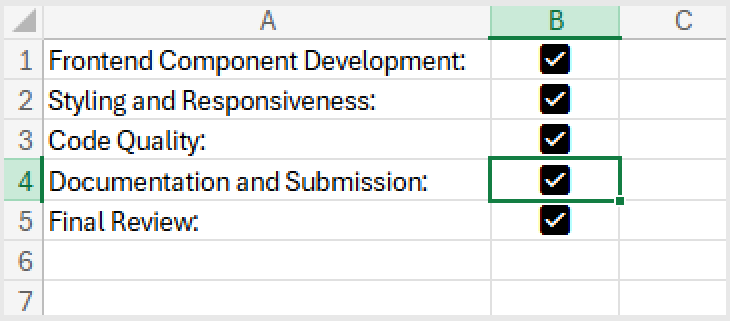
Best Practices Followed

1. **Reusable Components:**
    a. Created modular components like Header,Footer,ProductGrid, FilterBar etc for flexibility.
    b. Used props to pass data and maintain reusability.

2. **State Management:**
    a. Used Zustand for global state management to handle cart and wishlist features effectively.

3. **Styling:**
    a. Used Tailwind CSS for clean and responsive designs.
    b. Ensured mobile and desktop compatibility.

Checklist:



	A	B	C
1	Frontend Component Development:	✓	
2	Styling and Responsiveness:	✓	
3	Code Quality:	✓	
4	Documentation and Submission:	✓	
5	Final Review:	✓	
6			
7			

Conclusion

This assignment provided hands-on experience in building dynamic frontend components for a marketplace. The focus on modular design, responsive interfaces, and real-world integration prepared me for professional workflows.