

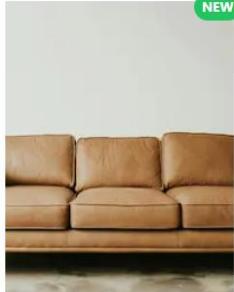
DAY 4 - DYNAMIC FRONTEND COMPONENTS

(Furniture)

Functional Deliverables

Screenshots of:

Product Listing Component:



Amber Haven
amber luxury cozy elegant furniture
Rs. 150



The Dandy chair
chair elegant vintage classic furniture
Rs. 150



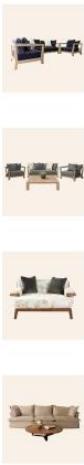
Syltherine
livingfancyelegance design
Rs. 200



Tropical Vibe
tropical vibe furniture exotic decor
Rs. 550

Product Detail Component:

Home > Shop > Reflective Haven



Reflective Haven

Rs.300

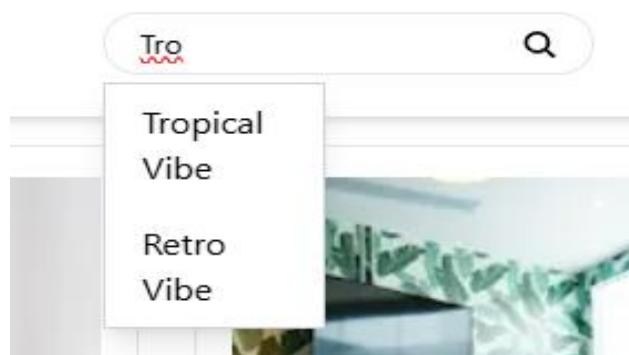
★★★★★ 5 Customer Reviews

Create a sanctuary of calm and introspection with ReflectiveHaven—a collection designed to bring peace, tranquility, and thoughtful elegance into your space. Inspired by serene environments and mindful living, ReflectiveHaven is perfect for those seeking a retreat from the hustle and bustle of everyday life. This collection invites you to slow down, reflect, and enjoy moments of stillness. The pieces in the ReflectiveHaven collection are designed with clean lines, soothing tones, and a harmonious blend of textures that foster a sense of balance and serenity. From minimalist furniture to subtle decorative accents, each item is thoughtfully crafted to create an atmosphere that encourages mindfulness and personal reflection. The calming neutral hues, natural materials, and understated beauty make it easy to turn any room into a peaceful haven. Perfect for meditation spaces, cozy reading nooks, or tranquil bedrooms, ReflectiveHaven brings a sense of clarity and inner peace to your home. It's for those who value simplicity, balance, and the ability to create an environment that nurtures both mental clarity and physical relaxation. Key Features: Minimalist design with a focus on calm, neutral tones and natural materials. High quality craftsmanship.

By Radiya Khan

Description**Additional Information****Review[5]**

Create a sanctuary of calm and introspection with ReflectiveHaven—a collection designed to bring peace, tranquility, and thoughtful elegance into your space. Inspired by serene environments and mindful living, ReflectiveHaven is perfect for those seeking a retreat from the hustle and bustle of everyday life. This collection invites you to slow down, reflect, and enjoy moments of stillness. The pieces in the ReflectiveHaven collection are designed with clean lines, soothing tones, and a harmonious blend of textures that foster a sense of balance and serenity. From minimalist furniture to subtle decorative accents, each item is thoughtfully crafted to create an atmosphere that encourages mindfulness and personal reflection. The calming neutral hues, natural materials, and understated beauty make it easy to turn any room into a peaceful haven. Perfect for meditation spaces, cozy reading nooks, or tranquil bedrooms, ReflectiveHaven brings a sense of clarity and inner peace to your home. It's for those who value simplicity, balance, and the ability to create an environment that nurtures both mental clarity and physical relaxation. Key Features: Minimalist design with a focus on calm, neutral tones and natural materials High-quality craftsmanship for lasting beauty and durability Perfect for creating peaceful, reflective spaces within your home Ideal for meditation, relaxation, or simply unwinding after a long day Invites mindfulness and inner peace with every detail Transform your living space into a peaceful retreat with ReflectiveHaven—where serene design meets mindful living, creating a sanctuary for reflection and relaxation.

**Search bar Component:**

By Radiya Khan

Search Bar Functionality:

- **Real-time Filtering:** As users type, their input updates the `searchQuery` state, and products are filtered to match the query.
- **Dropdown Results:** Matching products are displayed in a dropdown list below the search bar.
- **Click Action:** Users can click on a product from the dropdown to navigate or perform actions.
- **Responsive Design:** The search bar is styled to fit all screen sizes and provide a clean user experience.

Code:

```
const Header = () => {  
  const [searchQuery, setSearchQuery] = useState('');  
  const [products, setProducts] = useState<any[]>([]);  
  const filteredProducts = products.filter(product =>  
    product.title.toLowerCase().includes(searchQuery.toLowerCase())  
  );  
  const handleSearch = (event: React.ChangeEvent<HTMLInputElement>) => {  
    setSearchQuery(event.target.value);  
  };
```

```
<div>  
  <div className="flex items-center gap-[10px] px-[15px] py-[5px] border border-color-yellow rounded-2xl">  
    <input  
      type="text"  
      placeholder="Search..."  
      value={searchQuery}  
      onChange={handleSearch}  
      className="bg-transparent outline-none text-white text-[14px] placeholder:text-white w-full" />  
    <IoSearch className="text-white text-[20px] h-[20px]" />  
  </div>  
  
  /* Dropdown for filtered products */  
  {searchQuery && filteredProducts.length > 0 && (  
    <div className="absolute bg-white w-[100px] mt-1 border border-gray-300 rounded-md shadow-lg z-10">  
      <ul>  
        {filteredProducts.map((product: any) => (  
          <li  
            key={product._id}  
            className="px-4 py-2 text-black hover:bg-gray-200 cursor-pointer"  
          >  
            {product.title}  
          </li>  
        ))}  
      </ul>  
    </div>  
  )}  
</div>
```

Wishlist Functionality:

By Radiya Khan

Wishlist

Home > Wishlist

Product	Price
	Asgard sofa Rs. 250,000.00

Activate Windc
Go to Settings to ac

Code:

```

"use client";

import React, { useState, useEffect } from "react";
import Image from "next/image";
import { MdDelete } from "react-icons/md";
import { client } from "@/sanity/lib/client";

const Wishlist = async () => [
  const [wishlist, setWishlist] = useState([]);

  // Fetch wishlist from localStorage on component mount
  useEffect(() => {
    const savedWishlist = JSON.parse(localStorage.getItem("wishlist") || "[]");
    setWishlist(savedWishlist);
  }, []);

  // Remove product from wishlist
  const removeFromWishlist = (productId :any) => {
    const updatedWishlist = wishlist.filter((item) => item !== productId);
    setWishlist(updatedWishlist);
    localStorage.setItem("wishlist", JSON.stringify(updatedWishlist));
    alert("Item removed from wishlist!");
  };
  const data = await client.fetch(`*[_type == "product"]{
    _id,
    title,
    description,
    productImage {
      asset->{
        url
      },
    },
    price,
    tags,
    discountPercentage,
    isNew
  }`).

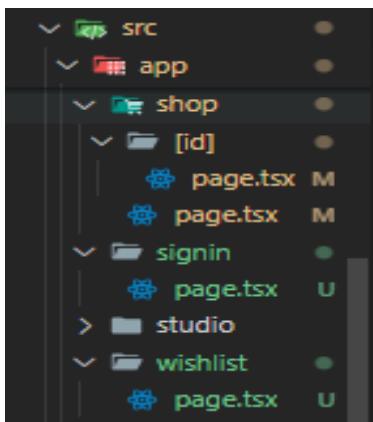
```

Code Deliverables

Product Details (Dynamic Routes):

- Shows a product image, name, description, price (original and discounted), and a discount percentage for real-time.
- Includes hover on the image and linking to the product's detailed page using its _id .

File Structure:



Code:

```
src > app > shop > [id] > page.tsx > Page
1  import Image from "next/image";
2  import Link from "next/link";
3  import { AiFillTwitterCircle } from "react-icons/ai";
4  import { FaChevronRight, FaFacebook, FaLinkedin } from "react-icons/fa";
5  import { client } from "@sanity/lib/client";
6  import { urlFor } from "@sanity/lib/image";
7
8
9  const Page = async ({params}:{params:{id:string}}) => [
10    const data = await client.fetch(`*[${_type == "product"} && ${_id == "${params.id}"}]{
11      _id,
12      title,
13      description,
14      productImage {
15        asset->{
16          url
17        },
18      },
19      price,
20      tags,
21      dicountPercentage,
22      isNew
23    });
24    console.log("Fetched Data:", data);
25    return (
26      <div>
27        <div className="bg-[#F9F1E7]">
28          <div className="h-[100px] max-w-[1400px] pl-2 gap-4 md:gap-8 flex items-center xl:pl-12 mx-auto">
29            <div className="flex gap-4 items-center">
30              <p className="font-poppins text-base font-normal text-[#9F9F9F]">
31                Home
32                </p>
33                <FaChevronRight />
34              </div>
35            <div className="flex gap-4 items-center">
36              <p className="font-poppins text-base font-normal text-[#9F9F9F]">
37                Shop
38                </p>
39                <FaChevronRight />

```

Product List:

- Fetches data from a Sanity CMS backend for items of type "product".
- Displays each product in a card format, showing its name, description, price, discount, and an image.

```
src > app > shop > 🌐 page.tsx > [⌚] Shop
1  import React from "react";
2  import Image from "next/image";
3  import { HiMiniAdjustmentsHorizontal } from "react-icons/hi2";
4  import { BsFillGridFill, BsViewList } from "react-icons/bs";
5  import Link from "next/link";
6  import CustomerCare from "@/components/customer-care";
7  import Banner from "@/components/banner";
8  import { client } from "@/sanity/lib/client";
9  import { urlFor } from "@/sanity/lib/image";
10 | import { FaHeart } from "react-icons/fa";
11 | import { GoArrowSwitch } from "react-icons/go";
12 | import { IoShareSocialSharp } from "react-icons/io5";
13 |
14 const Shop = async () => {
15   const data = await client.fetch(`*[_type == "product"]{
16     _id,
17     title,
18     description,
19     productImage {
20       asset->{
21         url
22       },
23     },
24     price,
25     tags,
26     dicountPercentage,
27     isNew
28   }`);
29   console.log("Fetched Data:", data);
30   const fatalimit = data.slice(0 ,47)
31   const formatPrice = (price: number) => {
32     // Convert number to string and format it as X.XX.XXX
33     const priceStr = price.toString();
34     const formatted = priceStr.replace(/(\d)(?=(\d{3})+(?!\\d))/g, "$1.");
35     return formatted;
36   };
37 |
38   return (
39     <div className="min-h-screen">
```

API Integration:

1. **Sanity API Fetching Logic:** Since you're using Sanity CMS, you need to ensure data fetching is properly managed.

- **Sanity Client Setup** (if not already set up): In your `lib` folder, configure the Sanity client:
- Fetching Data in `useEffect`:

```

src > app > shop > 🌐 page.tsx > [Shop]
1 import React from "react";
2 import Image from "next/image";
3 import { HiMiniAdjustmentsHorizontal } from "react-icons/hi2";
4 import { BsFillGridFill, BsViewList } from "react-icons/bs";
5 import Link from "next/link";
6 import CustomerCare from "@/components/customer-care";
7 import Banner from "@/components/banner";
8 import { client } from "@/sanity/lib/client";
9 import { urlFor } from "@/sanity/lib/image";
10 import { FaHeart } from "react-icons/fa";
11 import { GoArrowSwitch } from "react-icons/go";
12 import { IoShareSocialSharp } from "react-icons/io5";
13
14 const Shop = async () => {
15   const data = await client.fetch(`*[_type == "product"]{
16     _id,
17     title,
18     description,
19     productImage {
20       asset->{
21         url
22       },
23     },
24     price,
25     tags,
26     discountPercentage,
27     isNew
28   }`);
29   console.log("Fetched Data:", data);
30   const fatalimit = data.slice(0 ,47)
31   const formatPrice = (price: number) => {
32     // Convert number to string and format it as X.XX.XXX
33     const priceStr = price.toString();
34     const formatted = priceStr.replace(/(\d)(?=(\d{3})+(?!\d))/g, "$1.");
35     return formatted;
36   };
37
38   return (
39     <div className="min-h-screen">

```

Dynamic Routing:

Dynamic routing is used to handle routes like /shop/:id for individual product pages.

1. Create Dynamic Route File:

In the pages directory, create a [id].tsx file inside the shop folder:
pages/shop/[id].tsx

2. Get Product Data Dynamically:

```
src > app > shop > [id] > page.tsx > Page
  1 < import Image from "next/image";
  2 < import Link from "next/link";
  3 < import { AiFillTwitterCircle } from "react-icons/ai";
  4 < import { FaChevronRight, FaFacebook, FaLinkedin } from "react-icons/fa";
  5 < import { client } from "@/sanity/lib/client";
  6 < import { urlFor } from "@/sanity/lib/image";
  7
  8
  9 < const Page = async ({params}:{params:{id:string}}) => [
10 <   const data = await client.fetch(`*[_type == "product" && _id == "${params.id}"]{
11     _id,
12     title,
13     description,
14     productImage {
15       asset->{
16         url
17       },
18     },
19     price,
20     tags,
21     discountPercentage,
22     isNew
23   });
24   console.log("Fetched Data:", data);
25   return (
26     <div>
27       <div className="bg-[#F9F1E7]">
28         <div className="h-[100px] max-w-[1400px] pl-2 gap-4 md:gap-8 flex items-center xl:pl-12 mx-auto">
29           <div className="flex gap-4 items-center">
30             <p className="font-poppins text-base font-normal text-[#9F9F9F]">
31               Home
32             </p>
33             <FaChevronRight />
34           </div>
35           <div className="flex gap-4 items-center">
36             <p className="font-poppins text-base font-normal text-[#9F9F9F]">
37               Shop
38             </p>
39             <FaChevronRight />

```

Final CheckList:

Frontend Component Development	Styling and Responsiveness	Code Quality	Documentation and Submission	Final Review
✓	✓	✓	✓	✓