

DAY 4

Building Dynamic Frontend

Components For Marketplace:-

Built Components:

1. Product Listing Component:

- Dynamically fetch and display product data from Sanity CMS or API

```

interface Product {
  _id: string;
  name: string;
  category: string;
  brand: string;
  slug: { current: string };
  image: string;
  price: number;
  quantity: number;
  tags: string[];
  description: string;
  features: string[];
  dimensions: {
    height: number;
    width: number;
    depth: number;
  };
  dateAdded: string;
}

async function fetchProducts(
  selectedCategory?: string,
  minPrice?: number,
  maxPrice?: number,
  selectedBrand?: string,
  sortOrder?: string // Add sorting parameter
) {
  const query = `
    *[_type == "product"


```

- Display product names and prices in a clean, responsive grid layout.


All products

Category ▾Price ▾Brands ▾


Sort by: Date Added ▾




Pure Wood Round Center Table
Price: \$320
[View Details](#)



Saint Tropez Tall Vase Plant Pot
Price: \$280
[View Details](#)



Snacks Candy Serving Bowls With Wooden Tray
Price: \$300
[View Details](#)



Snack Serving Tray
Price: \$100
[View Details](#)

2. Product Detail Component:

- Use dynamic routing to render individual product details, including name, price, description, and additional data.

```

"use client";

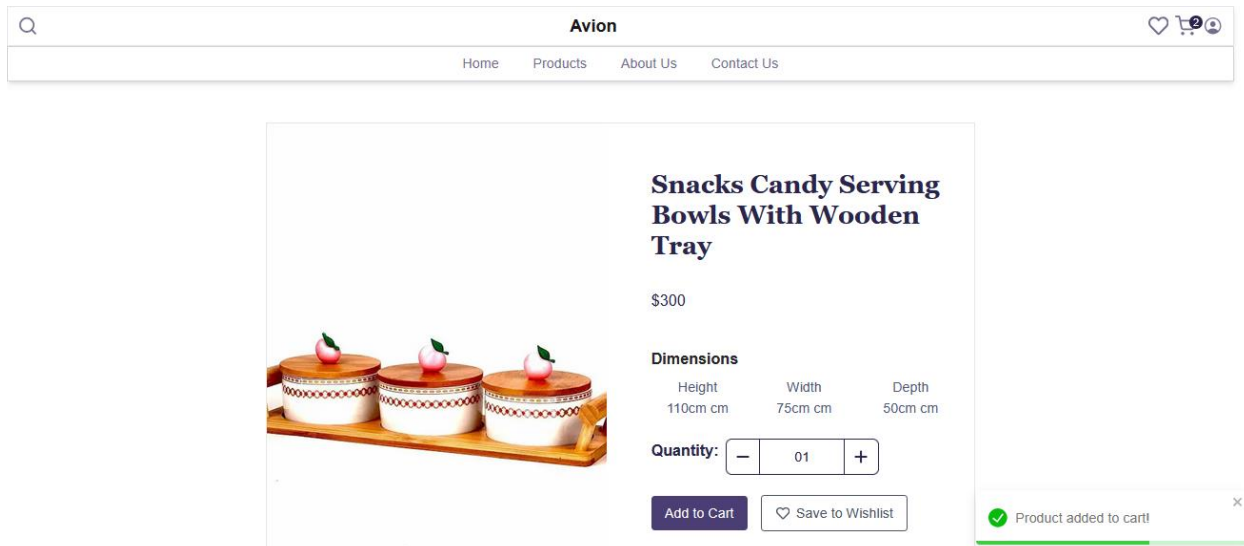
import { useState, useEffect } from "react";
import Image from "next/image";
import { FaPlus, FaMinus } from "react-icons/fa6";
import { FaRegHeart, FaHeart } from "react-icons/fa";
import { toast, ToastContainer } from "react-toastify";
import "react-toastify/dist/ReactToastify.css";

interface Product {
  _id: string;
  name: string;
  category: string;
  brand: string;
  slug: { current: string };
  image: string;
  price: number;
  quantity: number;
  tags: string[];
  description: string;
  features: string[];
  dimensions: {
    height: number;
    width: number;
    depth: number;
  };
  dateAdded: string;
}

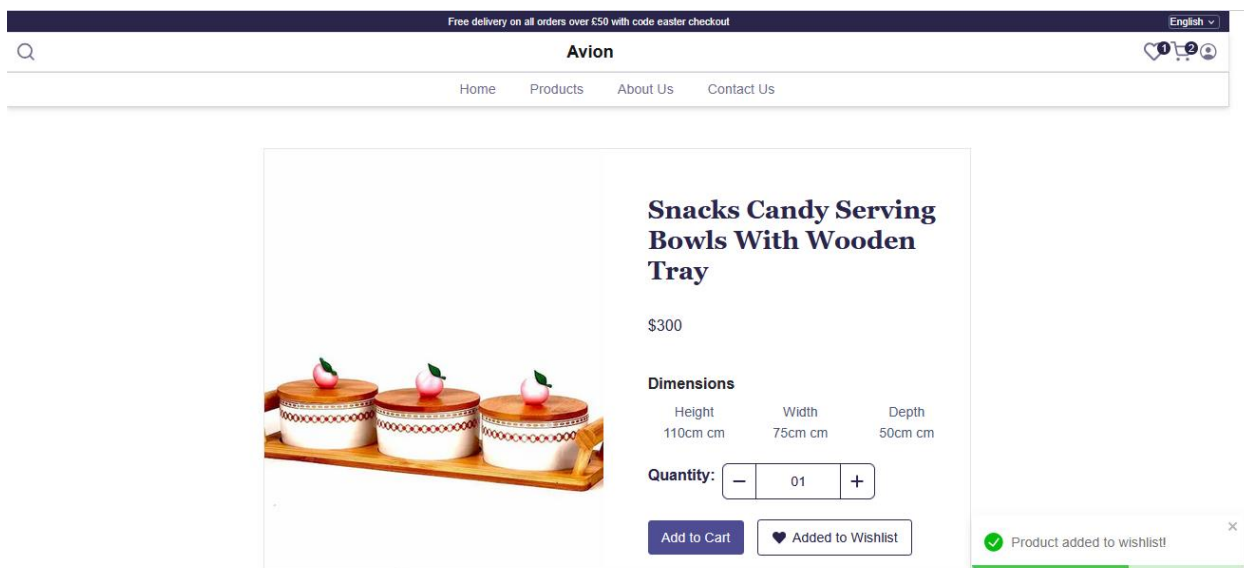
interface ProductDetailClientProps {
  product: Product;
}

```

- Provide an "Add to Cart" button with functionality.



- Provide an "Save to Wishlist" button with functionality.



3. Category Component:

- Fetch categories dynamically from the data source.

```

"use client";
import { useState, useEffect } from "react";
import { MdArrowDropDown } from "react-icons/md";
import { client } from "@sanity/lib/client";

interface Category {
  name: string;
  slug: { current: string };
}

interface CategoryProps {
  onCategorySelect: (categorySlug?: string) => void; // Callback for category selection
}

const Category = ({ onCategorySelect }: CategoryProps) => {
  const [categories, setCategories] = useState<Category[]>([]);
  const [isDropdownOpen, setIsDropdownOpen] = useState(false);

  useEffect(() => {
    const fetchCategories = async () => {
      const query = `*_type == "category"{ name, slug }`;
      const result = await client.fetch(query);
      setCategories(result);
    };

    fetchCategories();
  }, []);

  const toggleDropdown = () => {
    setIsDropdownOpen((prev) => !prev);
  };

```

- Allow filtering of products by selected categories.

All products

Category ▾ Price ▾ Brands ▾

All Categories

Ceramics

Chairs


Crockery

Cutlery


Plant Pots

Tables


Tableware



Wa Tableware **a Royal**
Platinum
Price: \$980
[View Details](#)



Bone Dinner Set
Price: \$250
[View Details](#)



Bowl Serving Regent Brilliant
3Tier With Stand
Price: \$480
[View Details](#)

4. Brand Component:

- Fetch brand dynamically from the data source.

```

"use client";
import { useState, useEffect } from "react";
import { MdArrowDropDown } from "react-icons/md";
import { client } from "@sanity/lib/client";

interface Brand {
  name: string;
  slug: { current: string };
}

interface BrandProps {
  onBrandSelect: (BrandSlug?: string) => void; // Callback for brand selection
}

const BrandFilter = ({ onBrandSelect }: BrandProps) => {
  const [brands, setBrands] = useState<Brand[]>([]);
  const [isDropdownOpen, setIsDropdownOpen] = useState(false);

  useEffect(() => {
    const fetchBrands = async () => {
      const query = `*[_type == "brand"]{ name, slug }`;
      const result = await client.fetch(query);
      setBrands(result);
    };

    fetchBrands();
  }, []);


  const toggleDropdown = () => {
    setIsDropdownOpen((prev) => !prev);
  };

```

- Allow filtering of products by selected brand.


All products

Category - Price - Brands - Sort by: Date Added -




Snacks Candy Serving Bowl With Wooden Tray
Price: \$300
[View Details](#)


All Brands
Corelle
Wedgwood
Versace
Kajaria
Habitt
Lenox
IKEA



Snack Serving Tray
Price: \$100
[View Details](#)



Bone Dinner Set
Price: \$250
[View Details](#)



Bowl Serving Regent Brilliant 3Tier With Stand
Price: \$480
[View Details](#)

[View Collection](#)

5. PriceFilter Component:

- Fetch pricefilter dynamically from the data source.

```

"use client";

import { useState } from "react";
import { useRouter, useSearchParams } from "next/navigation";
import { MdArrowDropDown } from "react-icons/md";

export default function PriceFilter() {
  const [showPriceFilter, setShowPriceFilter] = useState(false);
  const [minPrice, setMinPrice] = useState<number | undefined>(undefined);
  const [maxPrice, setMaxPrice] = useState<number | undefined>(undefined);

  const router = useRouter();
  const searchParams = useSearchParams();
  const selectedCategory = searchParams.get("category") || undefined;

  const handlePriceFilter = () => {
    const params = new URLSearchParams();
    if (selectedCategory) params.set("category", selectedCategory);
    if (minPrice) params.set("minPrice", String(minPrice));
    if (maxPrice) params.set("maxPrice", String(maxPrice));

    router.push(`?${params.toString()}`);

    // ▼ Dropdown close karne ke liye state update karein
    setShowPriceFilter(false);
  };


  return (
    <div className="relative">
      <button
        onClick={() => setShowPriceFilter(!showPriceFilter)}
        className="flex items-center text-[#2A254B] font-semibold pl-10"

```


- Allow filtering of products by selected pricefilter.

Category
Price
Brands


300
400
Apply




Pure Wood Round Center Table
Price: \$320
[View Details](#)




Snacks Candy Serving Bowls With Wooden Tray
Price: \$300
[View Details](#)




Infante Console Table
Price: \$300
[View Details](#)



Golden Duck Tea Spoon Set
Price: \$340
[View Details](#)





6. DateAdded Component:

- Fetch dateadded dynamically from the data source.

```

"use client";

import { useState } from "react";
import { useRouter, useSearchParams } from "next/navigation";
import { MdArrowDropDown } from "react-icons/md";

export default function DateAddedFilter() {
  const [showDateFilter, setShowDateFilter] = useState(false);
  const [sortOrder, setSortOrder] = useState<string>("desc");

  const router = useRouter();
  const searchParams = useSearchParams();
  const selectedCategory = searchParams.get("category") || undefined;

  const handleDateFilter = () => {
    const params = new URLSearchParams();
    if (selectedCategory) params.set("category", selectedCategory);
    if (sortOrder) params.set("sortOrder", sortOrder); // Set the sortOrder

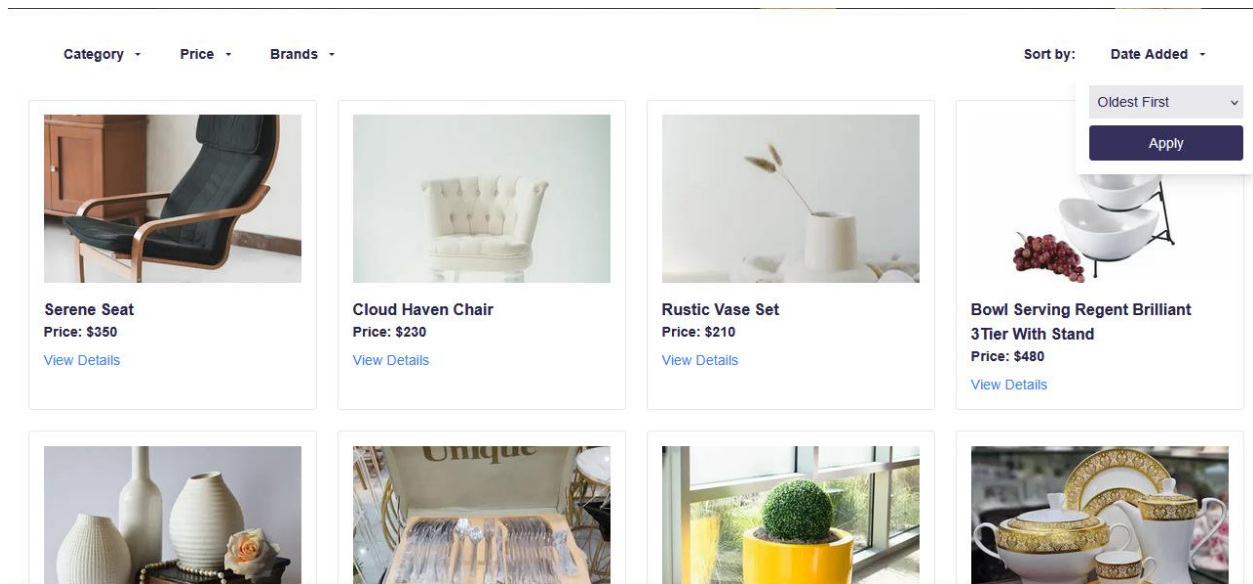
    router.push(`?${params.toString()}`);

    // 📌 Dropdown close karne ke liye state update karein
    setShowDateFilter(false);
  };

  return (
    <div className="relative">
      <button
        onClick={() => setShowDateFilter(!showDateFilter)}
        className="flex items-center text-[#2A254B] font-semibold pl-10"
      >

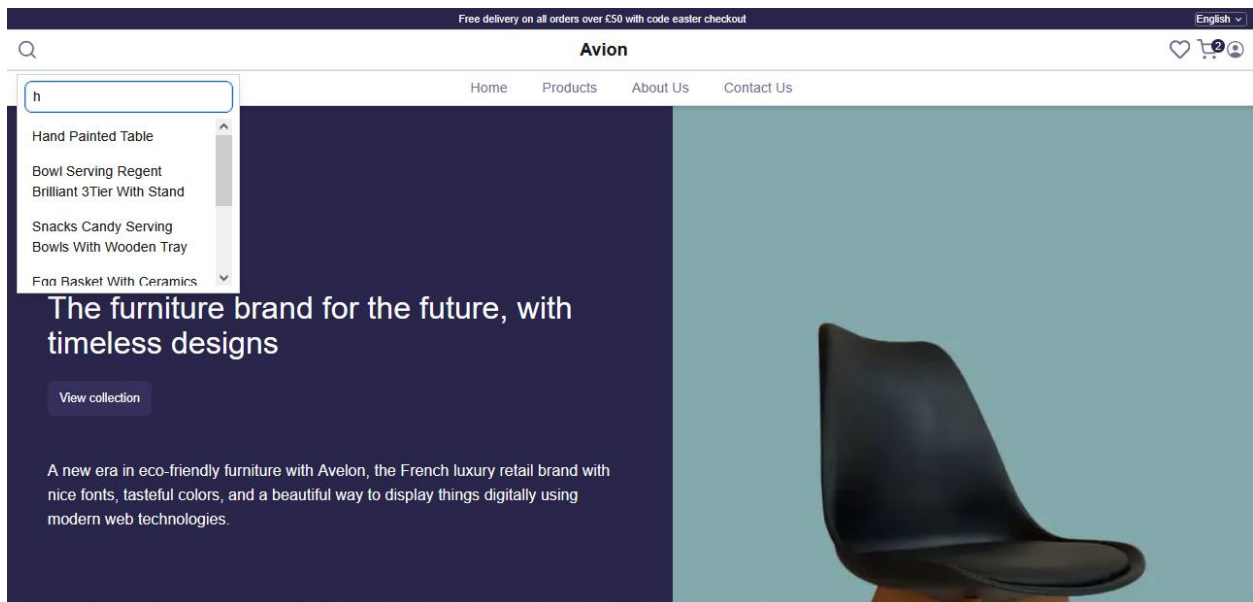
```

- Allow filtering of products by selected dateadded.



7. Advanced Search Bar:

- Implement faceted search to combine multiple filters and provide precise results.





Hand Painted Table

\$550

Dimensions

Height	Width	Depth
110cm cm	75cm cm	50cm cm

Quantity: - 01 +

Add to Cart

Save to Wishlist

8. Cart Component:

Your Shopping Cart

Your cart is empty.

Order Summary

Subtotal	\$0.00
Total	\$0.00

Checkout

Continue Shopping

- Enable users to view and manage their cart.

Your Shopping Cart



Snack Serving Tray

\$100.00

- 1 +

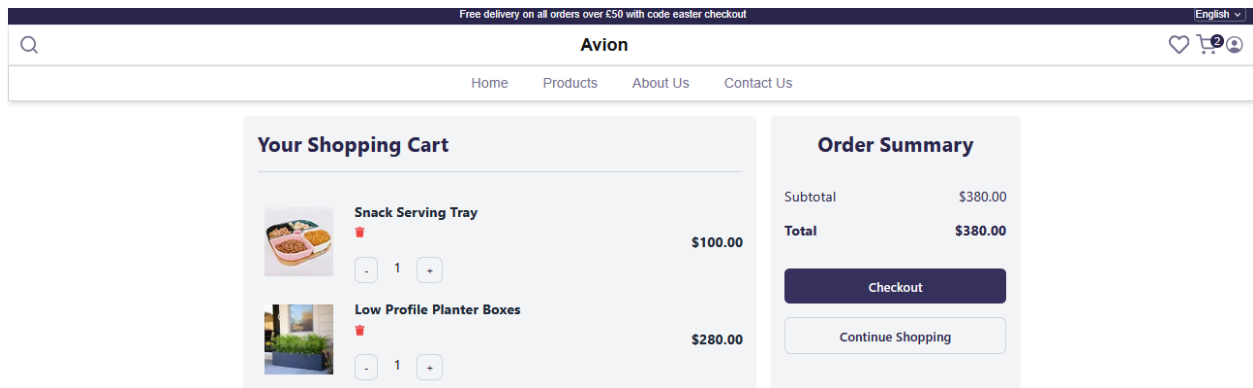
Order Summary

Subtotal	\$100.00
Total	\$100.00

Checkout

Continue Shopping

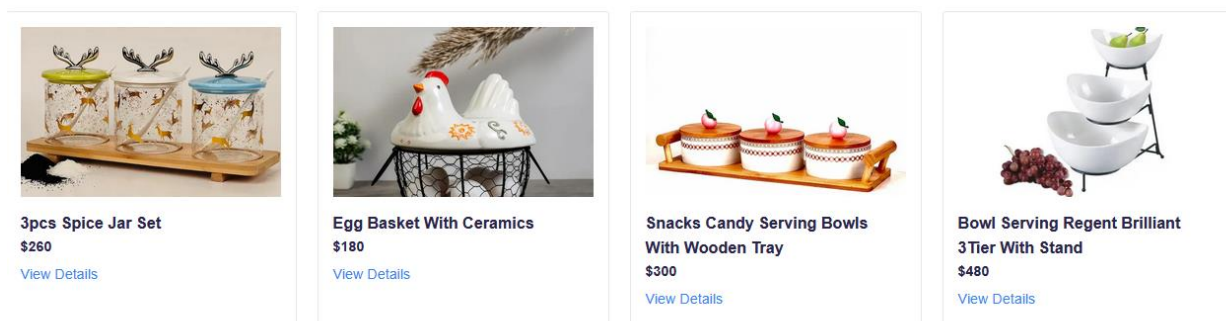
- Include features like updating quantities and removing items.



9. Related Products Component:

- Display similar products based on tags or categories on product detail pages.

You might also like



10. Footer and Header Components:

- Consistent navigation and branding elements.



- Include links to Home, About, Contact, and other key pages.
- Ensure responsiveness and accessibility.

Menu

New arrivals
Best sellers
Recently viewed
Popular this week
All products

Categories

Crockery
Furniture
Homeware
Plant pots
Chairs
Crockery

Our company

About us
Vacancies
Contact us
Privacy
Return policy

Join our mailing list

Sign up