

DAY 4 - BUILDING DYNAMIC FRONTEND COMPONENTS FOR YOUR MARKETPLACE :

Key Components:

1. Product Listing Component
2. Product Detail Component
3. Cart Component
4. Wishlist Component
5. Checkout Flow Component
6. Footer and Header Components
7. FAQ and Help Center Component
8. Social Media Sharing Component
9. API Integration

Product Listing Component:

The Product Listing Component showcases a collection of products in a well-organized and responsive layout, making it easy for users to browse and explore. It dynamically fetches product data, such as names, prices, images, and descriptions, while offering features like filtering, sorting, and search functionality to refine the browsing experience. With support for pagination or infinite scrolling, it efficiently handles large product catalogs. Additional features like product labels, ratings, quick add-to-cart options, and hover effects enhance usability and engagement. Designed to be visually appealing and optimized for performance, this component ensures a seamless and user-friendly shopping experience.

- **Dynamic Display:** Fetches and displays product data like name, price, image, and description.
- **User-Friendly Layout:** Organized in grids or lists for better navigation and visibility.
- **Filter & Sort Options:** Allows users to refine their search by category, price, or relevance.
- **Pagination Support:** Handles large datasets by breaking them into manageable pages.
- **Responsive Design:** Adapts to different screen sizes for a seamless user experience.

Code Snip:

```
'use client';
import { useState } from 'react';
import { Cart } from './Cart';
import { Image } from 'next/image';
import Link from 'next/link';
import { useGetProducts } from 'hooks/useGetProducts';
import { useGetCategory } from 'hooks/useGetCategory';
import { useGetProduct } from 'hooks/useGetProduct';
import { FaShoppingCart, FaUser, FaShuttle } from 'react-icons/fa';
import { BiList } from 'react-icons/bi';
import { Product } from '../types/Product';
import { useGetCategory } from 'hooks/useGetCategory';
import { useGetProducts } from 'hooks/useGetProducts';

const Products = () => {
  const [products, setProducts] = useState([]);
  const [category, setCategory] = useState('Food');
  const [productQuery, setProductQuery] = useState('');
  const [isCategorySelected, setIsCategorySelected] = useState(false);
  const [isProductSelected, setIsProductSelected] = useState(false);

  const handleCategoryChange = (category) => {
    setCategory(category);
    setProductQuery('');
    setIsCategorySelected(true);
    setIsProductSelected(false);
  };

  const handleProductSearch = (query) => {
    setProductQuery(query);
    setIsCategorySelected(false);
    setIsProductSelected(true);
  };

  const fetchProducts = async () => {
    try {
      const response = await fetch(`https://fakestoreapi.com/products`);
      const data = await response.json();
      setProducts(data);
    } catch (error) {
      console.error(error);
    }
  };

  useEffect(() => {
    if (!category) {
      return;
    }
    fetchProducts();
  }, [category]);
};

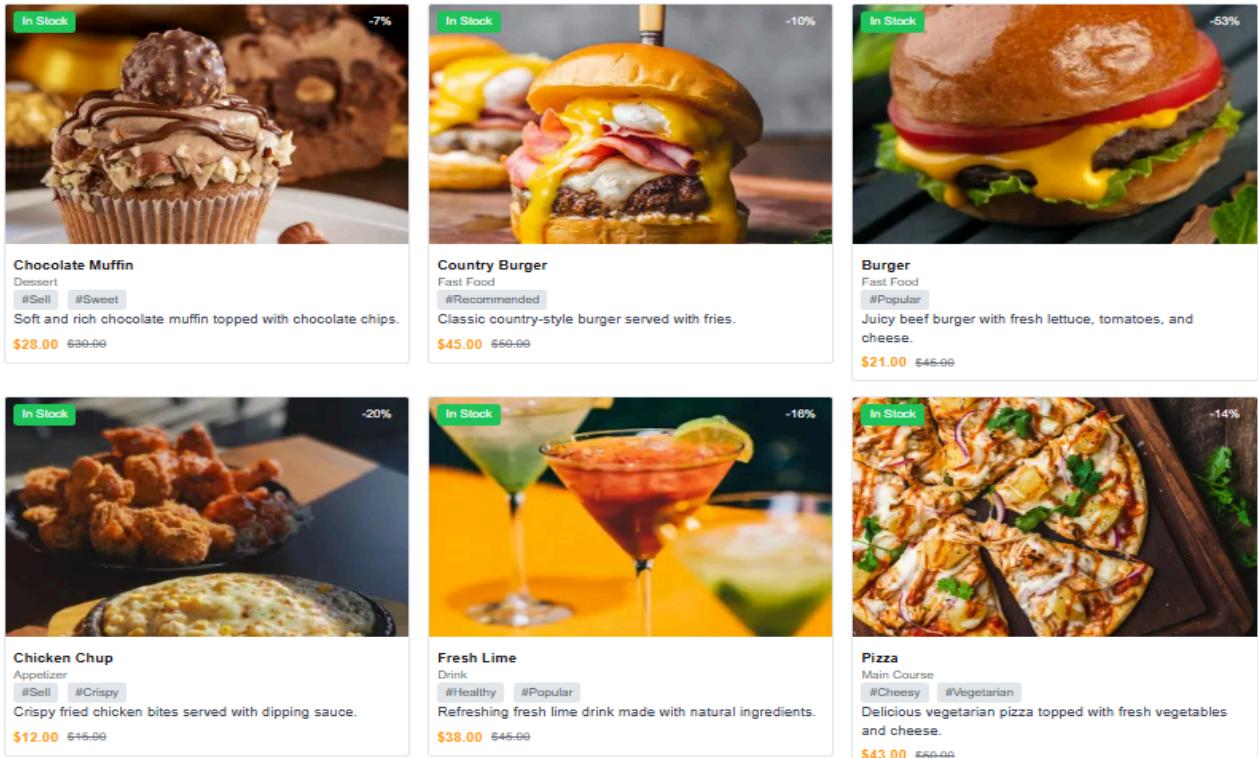
const ProductCard = ({ product }) => {
  const { id, title, description, image, price, originalPrice, category, rating, countInStock } = product;
  const discountPercentage = (originalPrice - price) / originalPrice * 100;

  return (
    <div style={{ border: '1px solid #ccc', padding: '10px', margin: '10px 0' }}>
      <div style={{ display: 'flex', align-items: 'center' }}>
        <img alt={title} style={{ width: '100px', height: '100px', objectFit: 'contain' }} />
        <div style={{ margin: '0 10px' }}>
          <span style={{ background-color: '#28a745', color: 'white', padding: '2px 5px', border-radius: '5px' }}>In Stock</span>
          <span style={{ background-color: '#f0f0f0', border: '1px solid #ccc', padding: '2px 5px', border-radius: '5px' }}>-{discountPercentage}%</span>
        </div>
        <div style={{ flex-grow: 1, display: 'flex', align-items: 'center' }}>
          <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{id}</span>
          <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{category}</span>
          <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{rating}</span>
          <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{countInStock}</span>
        </div>
      </div>
      <div style={{ margin: '10px 0' }}>
        <span style={{ font-weight: 'bold', font-size: '1.2em' }}>${price}</span>
        <span style={{ font-size: '0.8em', color: '#ccc' }}> ${originalPrice}</span>
      </div>
      <div style={{ margin: '10px 0' }}>
        <span style={{ font-weight: 'bold', font-size: '1.2em' }}>{title}</span>
        <span style={{ font-size: '0.8em', color: '#ccc' }}> {category}</span>
      </div>
      <div style={{ margin: '10px 0' }}>
        <span style={{ font-weight: 'bold', font-size: '1.2em' }}>{description}</span>
        <span style={{ font-size: '0.8em', color: '#ccc' }}> {image}</span>
      </div>
    </div>
  );
};

const ProductList = ({ products }) => {
  return (
    <div style={{ display: 'grid', grid-template-columns: '1fr 1fr 1fr', gap: '10px' }}>
      {products.map((product) => (
        <div style={{ border: '1px solid #ccc', padding: '10px', position: 'relative' }}>
          <img alt={product.image} style={{ width: '100%', height: '100%', objectFit: 'cover' }} />
          <div style={{ position: 'absolute', top: 0, left: 0, width: '100%', height: '100%' }}>
            <span style={{ background-color: '#28a745', color: 'white', padding: '2px 5px', border-radius: '5px' }}>In Stock</span>
            <span style={{ background-color: '#f0f0f0', border: '1px solid #ccc', padding: '2px 5px', border-radius: '5px' }}>-{product.discountPercentage}</span>
          </div>
          <div style={{ display: 'flex', align-items: 'center', margin-top: '10px' }}>
            <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{product.id}</span>
            <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{product.category}</span>
            <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{product.rating}</span>
            <span style={{ font-size: '1.2em', margin-right: '10px' }}>#{product.countInStock}</span>
          </div>
          <div style={{ margin: '10px 0' }}>
            <span style={{ font-weight: 'bold', font-size: '1.2em' }}>${product.price}</span>
            <span style={{ font-size: '0.8em', color: '#ccc' }}> ${product.originalPrice}</span>
          </div>
          <div style={{ margin: '10px 0' }}>
            <span style={{ font-weight: 'bold', font-size: '1.2em' }}>{product.title}</span>
            <span style={{ font-size: '0.8em', color: '#ccc' }}> {product.description}</span>
          </div>
          <div style={{ margin: '10px 0' }}>
            <span style={{ font-weight: 'bold', font-size: '1.2em' }}>{product.image}</span>
            <span style={{ font-size: '0.8em', color: '#ccc' }}> {product.image}</span>
          </div>
        </div>
      ))}
    </div>
  );
};

export default Products;
```

Outlook Snip:



Product Detail Component:

Code Snip:

```
// Fetch product data using Sanity client
const productFetchData: Dull[] = await client.fetch(
  groq`{type == "Food" && slug.current == ${slug}[
    name,
    category,
    price,
    originalPrice,
    tags,
    image,
    description,
    available,
    slug,
    _id
  ]`,
  { slug } // Pass id as a parameter to avoid string interpolation for security
);

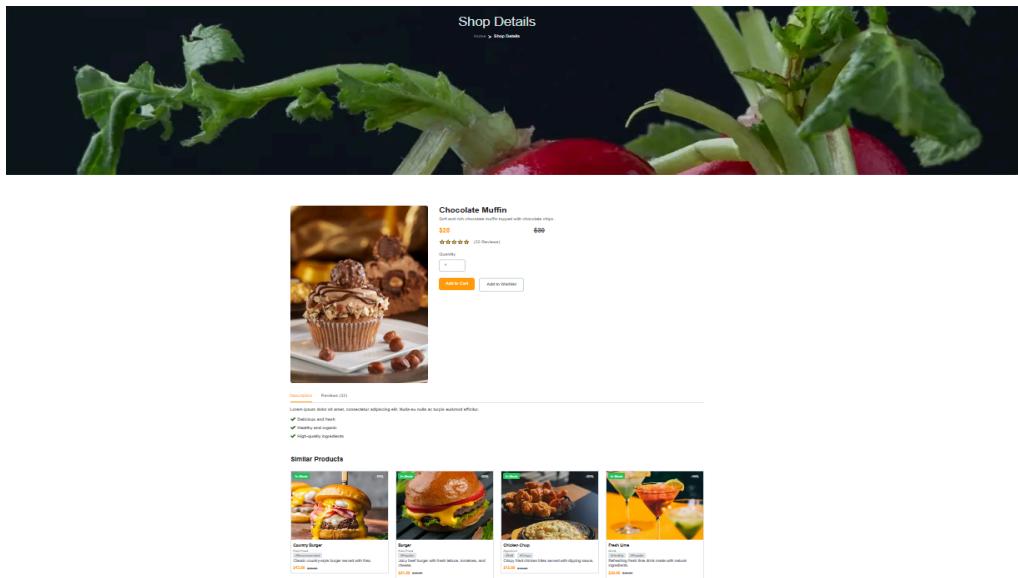
if (productFetchData.length > 0) {
  setProduct(productFetchData[0]); // Assuming only one product matches the ID
} else {
  console.error("No product found for the given product name.");
}

} catch (error) {
  console.error("Error fetching product data:", error);
}

};

fetchProductData();
}, [props.params]);
console.log(product);
```

Outlook Snip:



Cart Component:

The Cart Component is responsible for displaying items added to the cart, including their name, image, price, and quantity. It allows users to update item quantities, remove items, and view the total price. Integrated with state management (e.g., Redux), it dynamically updates the cart and provides a checkout button for proceeding to payment. Styled for responsiveness, it ensures a seamless shopping experience.

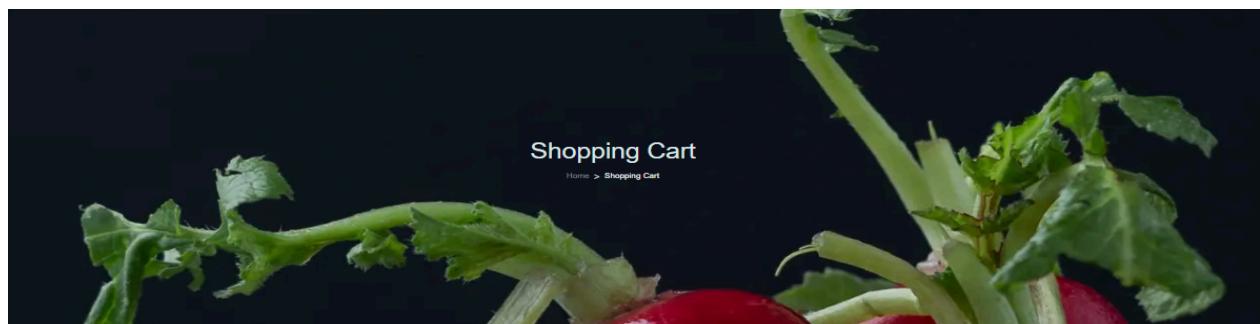
Code Snip:

```
const CartPage = () => {
  const dispatch = useAppDispatch()
  const cartSelector = useAppSelector(getCart)
  let totalPrice = 0
  cartSelector.forEach(item:Dish) => {
    totalPrice += item.price * item.quantity;
  }

  return (
    <>
    <MainBreadcum name='Shopping Cart' pageName=' Shopping Cart' />
    <div className="container mx-auto py-10">
      <div className="overflow-x-auto">
        <table className="w-full border border-gray-200">
          <thead>
            <tr>
              <th className="p-3 text-left">Product</th>
              <th className="p-3 text-left">Price</th>
              <th className="p-3 text-left">Quantity</th>
              <th className="p-3 text-left">Total</th>
              <th className="p-3 text-left">Remove</th>
            </tr>
          </thead>
          <tbody>
            {cartSelector.map(item:Dish) => (
              <tr key={item._id} className="border-b">
                <td className="p-3 flex items-center space-x-3">
                  <Image
                    src={urlFor(item.image).url()}
                    alt={item.name}
                    width={150}
                    height={150}
                    className="w-16 h-16 rounded object-cover"
                  />
                  <span>{item.name}</span>
                </td>
                <td className="p-3">${item.price.toFixed(2)}</td>
                <td className="p-3">
                  <div className="flex items-center">
                    <button
                      className="px-2 py-1 border border-gray-300 rounded"
                      onClick={() => {
                        if(item.quantity > 1){
                          dispatch(decrementQuantity(item))
                        }
                      }}
                    >
                      -
                    </button>
                    <span className="px-4">{item.quantity}</span>
                    <button
                      className="px-2 py-1 border border-gray-300 rounded"
                      onClick={() => {
                        dispatch(incrementQuantity(item))
                      }}
                    >
                      +
                    </button>
                  </div>
                </td>
                <td className="p-3">${(item.price * item.quantity).toFixed(2)}</td>
                <td className="p-3">
                  <button
                    onClick={() => {
                      dispatch(removeFromTheCart(item._id))
                    }}
                    className="text-red-500 hover:text-red-700"
                  >
                    <FaIcon className="text-lg" />
                  </button>
                </td>
              </tr>
            )}
          </tbody>
        </table>
      </div>
    </>
  )
}
```

```
  <div>
    <table>
      <thead>
        <tr>
          <th>Product</th>
          <th>Price</th>
          <th>Quantity</th>
          <th>Total</th>
          <th>Remove</th>
        </tr>
      </thead>
      <tbody>
        {cartSelector.map(item:Dish) => (
          <tr key={item._id} className="border-b">
            <td>
              <Image
                src={urlFor(item.image).url()}
                alt={item.name}
                width={150}
                height={150}
                className="w-16 h-16 rounded object-cover"
              />
              <span>{item.name}</span>
            </td>
            <td>${item.price.toFixed(2)}</td>
            <td>
              <div>
                <button
                  onClick={() => {
                    dispatch(decrementQuantity(item))
                  }}
                  className="px-2 py-1 border border-gray-300 rounded"
                >
                  If
                </button>
                <span>{item.quantity}</span>
                <button
                  onClick={() => {
                    dispatch(incrementQuantity(item))
                  }}
                  className="px-2 py-1 border border-gray-300 rounded"
                >
                  +
                </button>
              </div>
            </td>
            <td>${(item.price * item.quantity).toFixed(2)}</td>
            <td>
              <button
                onClick={() => {
                  dispatch(removeFromTheCart(item._id))
                }}
                className="text-red-500 hover:text-red-700"
              >
                <FaIcon className="text-lg" />
              </button>
            </td>
          </tr>
        )}
      </tbody>
    </table>
  </div>
```

Outlook Snip:



The screenshot shows a shopping cart interface with a dark background featuring a close-up image of radishes. At the top, there's a navigation bar with links for Home, Cart, and Checkout. Below the navigation, the title "Shopping Cart" is displayed above a breadcrumb trail showing "Home > Shopping Cart".

The main content area contains a table with the following data:

Product	Price	Quantity	Total	Remove
Burger	\$21.00	4	\$84.00	X
Country Burger	\$45.00	6	\$270.00	X
Chocolate Muffin	\$28.00	1	\$28.00	X
Chicken Chup	\$12.00	1	\$12.00	X

Below the table, there are two input fields: "Coupon Code" with a placeholder "Enter your code" and an "Apply" button. To the right, there's a summary section titled "Total Bill" showing "Cart Subtotal" (\$394), "Shipping Charge" (\$0), and "Total Amount" (\$394). At the bottom right, there's a large orange button labeled "Proceed to Checkout".

Wishlist Component:

The Wishlist Component allows users to save and manage their favorite products for future reference. It displays a list of saved items with details such as name, image, and price. Users can remove items from the wishlist or move them directly to the cart for purchase. Integrated with state management (e.g., Redux), the component ensures dynamic updates and provides a visually appealing and responsive layout for a better user experience.

Code Snip:

```
const handleFollow = (productId: string) => {
  dispatch(unlikeProduct(productId))
}

return []
}



<div>
    <h3>WishList Page</h3>
    <div>
      <div>
        <h4>WishList Page</h4>
        <div>
          <h5>Grid View</h5>
          <table border="1">
            <thead>
              <tr>
                <th>Image</th>
                <th>Product Name</th>
                <th>Price</th>
                <th>Action</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <td><img alt="product image" href="#"/><br/>Alt product name<br/>height: 127px<br/>width: 111px<br/>classname: w-full h-127px object-cover</td>
                <td>Original Price <span>₹ 100</span><br/>Current Price <span>₹ 100</span><br/>Math.floor((product.originalPrice - product.price) / product.originalPrice * 100) % 100</td>
                <td>In Stock</td>
                <td><button>Add To Cart</button></td>
              </tr>
            </tbody>
          </table>
        </div>
      </div>
    </div>
  </div>


```

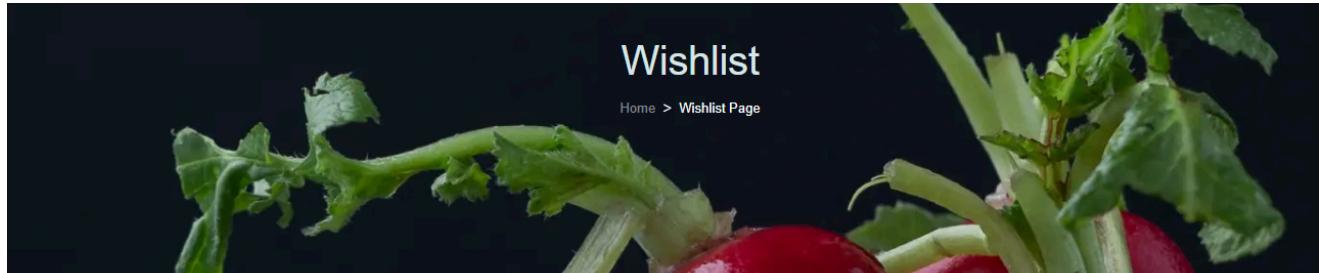
```
<Link href={""}>
  <button type="button"
    onClick={() => [
      dispatch(likeProduct(product))
    ]}>
    <span>{product.likes}</span>
    <span>Like</span>
  </button>
</Link>



<div>
    <div>
      <div>
        <div>
          <div>
            <div><Link href={product.id}><span>View Details</span></Link></div>
            <div><Link href={product.id}><span>Edit</span></Link></div>
            <div><Link href={product.id}><span>Delete</span></Link></div>
          </div>
        </div>
      </div>
    </div>
  </div>


```

Outlook Snip:



WishList Page



Country Burger
Fast Food
#Recommended
Classic country-style burger served with fries.
\$45.00 \$50.00

[Remove](#)



Chocolate Muffin
Dessert
#Sell #Sweet
Soft and rich chocolate muffin topped with chocolate chips
\$28.00 ~~\$30.00~~

[Remove](#)

Checkout Flow Component:

The Checkout Flow Component streamlines the purchasing process by guiding users through key steps like reviewing their cart, entering shipping details, selecting payment methods, and confirming their order. It ensures a smooth and secure transaction experience by validating user inputs and integrating with payment gateways. Designed to be intuitive and responsive, this component provides a clear and efficient path to complete purchases seamlessly.

Code Snip:

```
"use client";
import MainBreadcum from "B/app/components/MainBreadcum";
import Link from "next/link";
import { useState } from "react";
import Image from "next/image";
import { useAppDispatch, useAppSelector } from "../../../../hooks/redux";
import { cartSlice } from "../../../../redux/cartSlice";
import { Dull } from "../../../../types/Product";
import { urlFor } from "gsaanity/lib/image";

const CheckoutPage = () =>
  const [shippingAddress, setShippingAddress] = useState(true);
  const dispatch = useAppDispatch();
  const cartSelector = useAppSelector(getCart);
  let totalPrice = 0;
  cartSelector.forEach((item: Dull) => [
    totalPrice += item.price * item.quantity;
  ]);
  return (
    <>
      <MainBreadcum name="Checkout Page" pageName="Checkout" />

      <div className="min-h-screen bg-gray-100 py-10">
        <div className="container mx-auto px-4 lg:px-8">
          </> Shipping Address
          <div className="flex flex-col lg:flex-row justify-between space-y-10 lg:space-y-0 lg:space-x-8">
            <div className="flex flex-col lg:rounded-lg shadow-md">
              <h2 className="text-lg font-semibold mb-4">Shipping Address</h2>
              <form className="space-y-4">
                <div className="grid grid-cols-1 sm:grid-cols-2 gap-4">
                  <div>
                    <label>First name</label>
                    <input type="text" placeholder="First name" />
                  </div>
                  <div>
                    <label>Last name</label>
                    <input type="text" placeholder="Last name" />
                  </div>
                  <div>
                    <label>Email</label>
                    <input type="text" placeholder="Email" />
                  </div>
                  <div>
                    <label>Phone number</label>
                    <input type="text" placeholder="Phone number" />
                  </div>
                  <div>
                    <label>Company</label>
                    <input type="text" placeholder="Company" />
                  </div>
                  <div>
                    <label>City</label>
                    <input type="text" placeholder="Choose city" />
                  </div>
                  <div>
                    <label>Address 1</label>
                    <input type="text" placeholder="Address 1" />
                  </div>
                  <div>
                    <label>Address 2</label>
                    <input type="text" placeholder="Address 2" />
                  </div>
                  <div>
                    <input checked="" type="checkbox" /> Same as shipping address
                  </div>
                </div>
              </form>
            </div>
          </div>
        </div>
      </div>
    </>
  );

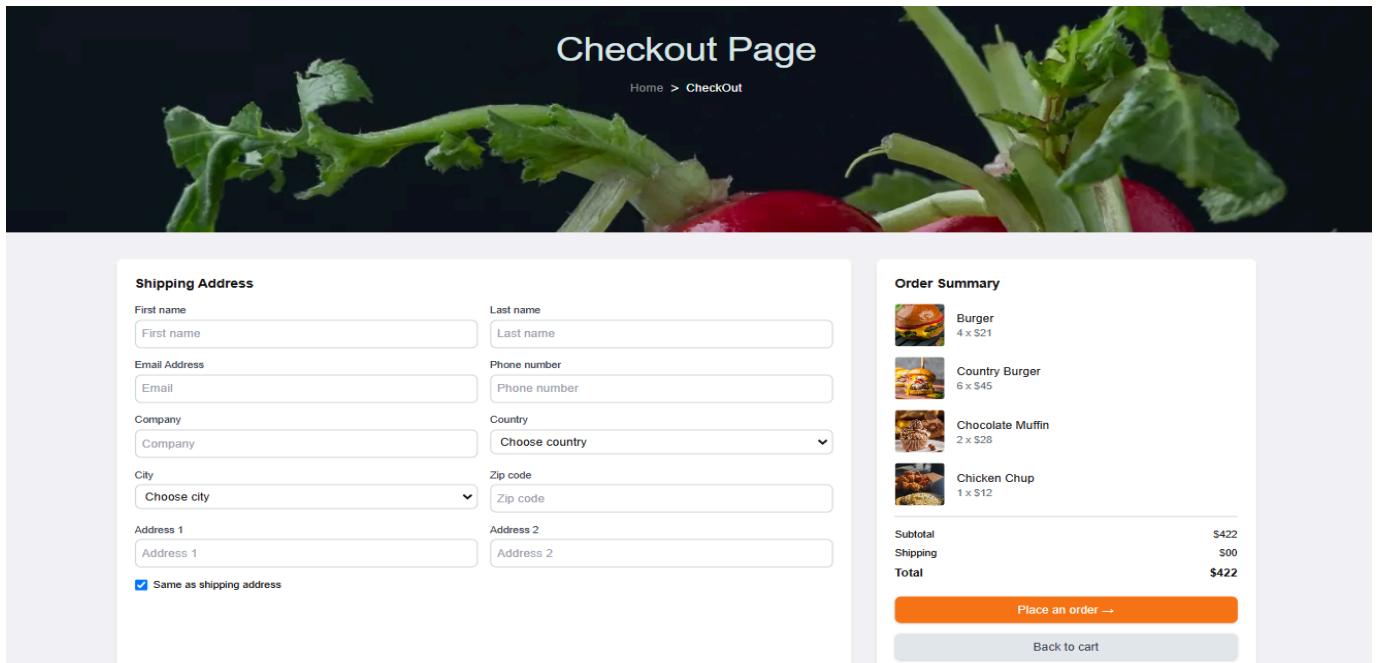
```

```
/* Order Summary */


<h2>Order Summary</h2>
  <div className="space-y-4">
    <cartSelector.map(product: Dull) => (
      <div key={product.id} className="flex items-center space-x-4">
        <img alt={product.name} width={200} height={200} className="w-16 h-16 rounded object-cover" />
        <div>
          <img alt={urlFor(product.image).url()} />
          <p>{product.name}</p>
          <p>{product.quantity} x ${product.price}</p>
        </div>
      </div>
    )
  </div>
  <div className="border-t border-gray-300 mt-4 pt-4 space-y-2">
    <div>
      <span>Subtotal</span>
      <span>${totalPrice}</span>
    </div>
    <div>
      <span>Shipping</span>
      <span>$0</span>
    </div>
    <div>
      <span>Total</span>
      <span>${totalPrice}</span>
    </div>
  </div>


```

Outlook Snip:



Footer and Header Component:

Code Snip:

```
1  /*use client*/
2  import { FaRegHeart } from "react-icons/fa";
3
4  import { useState } from "react";
5  import { FiSearch, FiShoppingCart } from "react-icons/fi";
6  import Link from "next/link";
7  import { FaShoppingCart, FaStarAlt, FaBars, FaTimes, FaSearch } from "react-icons/fa";
8  import { useAppSelector } from "../../../../../hooks/redux";
9  import { getCart, getLikedProducts } from "../../../../../redux/cartSlice";
10
11
12  export default function Header() {
13    const [menuOpen, setMenuOpen] = useState(false);
14
15    const cartSelector = useAppSelector(getCart);
16    const likeSelector = useAppSelector(getLikedProducts);
17
18    return (
19      <div>
20        </div>
21        <header className="bg-transparent text-white relative">
22          <!-- Logo Section -->
23          <div className="flex justify-center mx-auto">
24            <a href="/">
25              <h1 className="text-2xl font-bold border-[#FF9F00]">Food</h1>
26              <span className="text-white">luck</span>
27            </a>
28          </div>
29        </div>
30
31        <!-- Navigation Bar -->
32        <div className="container mx-auto flex items-center justify-between px-4 py-4">
33          <!-- Desktop Navigation -->
34          <div className="hidden md:flex space-x-7 text-sm">
35            <a href="#">Home</a>
36            <a href="#">Menu</a>
37            <a href="#">Blog</a>
38          </div>
39
40          <div className="relative group">
41            <a href="#">About</a>
42            <button className="hover:text-[#FF9F00] flex items-center">
43              About <span className="ml-1"></span>
44            </button>
45          </div>
46        </div>
47      </div>
48    );
49  }
50
```

```
import Image from "next/image";
import Link from "next/link";
import { FaFacebook, FaTwitter, FaInstagram, FaYoutube } from "react-icons/fa";
import { AiOutlineClockCircle } from "react-icons/ai";

const Footer = () => {
  return (
    <footer className="bg-black text-gray-100 relative bg-[url('/footerLeaves.png')] bg-no-repeat bg-right-bottom -z-0">
      <div className="mx-10 py-10 flex flex-wrap justify-between border-b-2 border-[#FF9F00] px-6 lg:px-24 py-10">
        <!-- Top Section -->
        <section className="flex flex-wrap justify-between border-b-2 border-[#FF9F00] px-6 lg:px-24 py-10">
          <div className="mb-6 lg:mb-0">
            <h3 className="text-2xl font-bold">
              <span>Still</span> need our support?
            </h3>
            <p>Don't wait, make a smart & logical quote here. It's pretty easy.</p>
          </div>
        </div>
        <div className="flex flex-wrap items-center">
          <input type="email" placeholder="Enter your email" className="bg-gray-800 text-gray-100 px-4 py-2 rounded-l-md outline-none" />
          <button className="bg-[#FF9F00] text-black px-4 py-2 rounded-r-md hover:bg-orange-500" onClick={Subscribe now}>Subscribe now</button>
        </div>
      </section>
      <!-- Main Footer -->
      <section className="flex flex-wrap justify-between mt-10 px-6 lg:px-24">
        <!-- About Us -->
        <div className="mb-8 lg:mb-0 lg:w-1/4">
          <h3>About Us</h3>
          <p>We serve delicious foods and dishes to give you quality meal and make your tastebuds feel a lot pleasure</p>
        </div>
      </section>
    </div>
  );
}

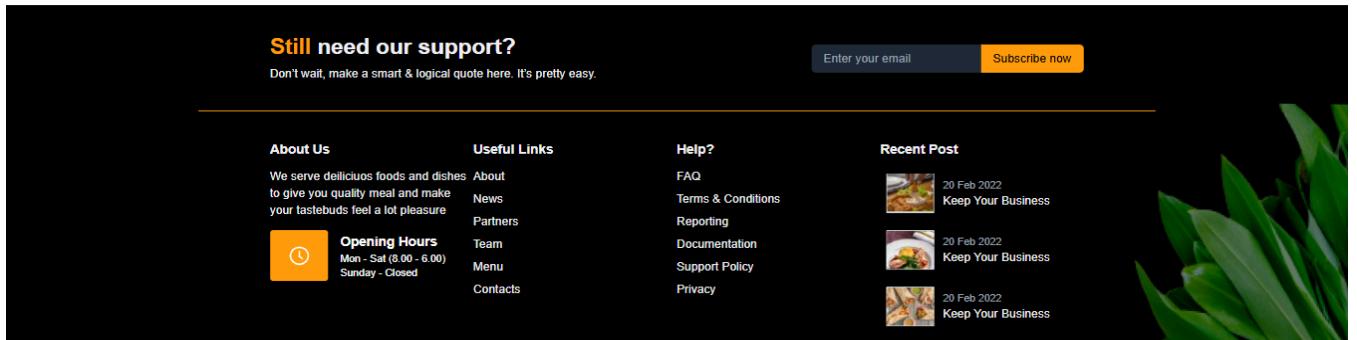
// Still need our support?
```

Outlook Snip:

Header



Footer



FAQ and Help Center Component:

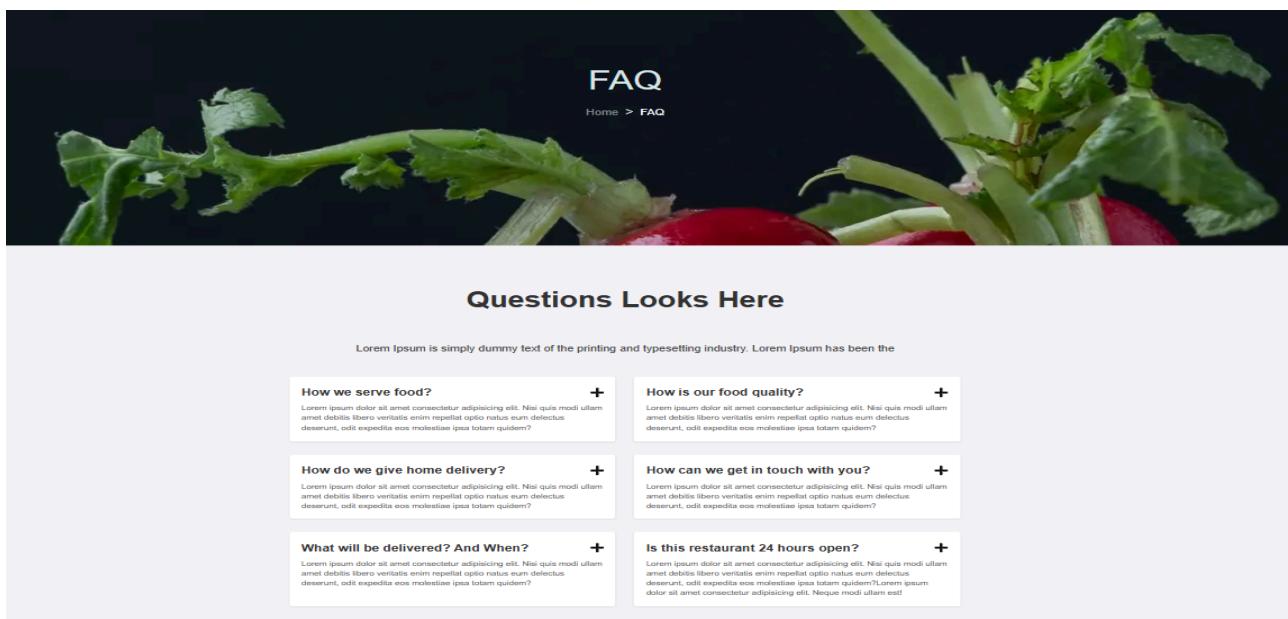
The FAQ and Help Center Component provides users with quick access to common questions and helpful resources. It features a searchable list of FAQs, categorized topics, and step-by-step guides to address user queries. Integrated with a clean and responsive design, this component ensures users can easily find support, enhancing their overall experience and reducing the need for direct customer service.

Code Snip:

```
import MainBreadcum from "@/app/components/MainBreadcum";
import { FaPlus } from "react-icons/fa";

const FAQ = () => {
  const faqs = [
    {
      question: "How we serve food?",
      answer: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?",
    },
    {
      question: "How is our food quality?",
      answer: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?",
    },
    {
      question: "How do we give home delivery?",
      answer: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?",
    },
    {
      question: "How can we get in touch with you?",
      answer: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?",
    },
    {
      question: "What will be delivered? And When?",
      answer: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?",
    },
    {
      question: "Is this restaurant 24 hours open?",
      answer: "Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?Lorem ipsum dolor sit amet consectetur adipisicing elit. Neque modi ullam est!",
    }
  ];
  return (
    <>
    <MainBreadcum name="FAQ" pageName="FAQ"/>
    <div className="bg-gray-100 min-h-screen p-6">
      <div className="max-w-4xl mx-auto">
        <div className="text-center mb-8">
          <p>Questions Looks Here:</p>
          <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the</p>
        </div>
        <div className="grid md:grid-cols-2 gap-6">
          <div>
            <h3>How we serve food?</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?</p>
          </div>
          <div>
            <h3>How is our food quality?</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?</p>
          </div>
          <div>
            <h3>How do we give home delivery?</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?</p>
          </div>
          <div>
            <h3>How can we get in touch with you?</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?</p>
          </div>
          <div>
            <h3>What will be delivered? And When?</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?</p>
          </div>
          <div>
            <h3>Is this restaurant 24 hours open?</h3>
            <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Nisi quis modi ullam amet debitis libero veritatis enim repellat optio natus eum delectus deserunt, odit expedita eos molestiae ipsa totam quidem?Lorem ipsum dolor sit amet consectetur adipisicing elit. Neque modi ullam est!</p>
          </div>
        </div>
      </div>
    </div>
  );
}
```

Outlook Snip:



Social Media Sharing Component:

Code Snip:

```
<section className="bg-gray-500 md:h-16 h-auto mt-2">
  <div className="max-w-7xl mx-auto flex justify-between items-center px-6 lg:px-24 h-full">
    <p>Copyright © 2024 by Raaef. All Rights Reserved.</p>
    <div>
      <a href="https://www.facebook.com" target="_blank" rel="noopener noreferrer" className="w-6 h-6 bg-gray-100 text-gray-600 flex items-center justify-center cursor-pointer hover:text-gray-800 hover:bg-gray-200">
        <FaFacebookF />
      </a>
      <a href="https://www.twitter.com" target="_blank" rel="noopener noreferrer" className="w-6 h-6 bg-gray-100 text-gray-600 flex items-center justify-center cursor-pointer hover:text-gray-800 hover:bg-gray-200">
        <FaTwitter />
      </a>
      <!-- Instagram -->
      <a href="https://www.instagram.com" target="_blank" rel="noopener noreferrer" className="w-6 h-6 bg-gray-100 text-gray-600 flex items-center justify-center cursor-pointer hover:text-gray-800 hover:bg-gray-200">
        <FaInstagram />
      </a>
      <!-- YouTube -->
      <a href="https://www.youtube.com" target="_blank" rel="noopener noreferrer" className="w-6 h-6 bg-gray-100 text-gray-600 flex items-center justify-center cursor-pointer hover:text-gray-800 hover:bg-gray-200">
        <FaYouTube />
      </a>
    </div>
  </div>
</section>
```

Outlook Snip:



API Integration:

API integration enables seamless communication between different applications by allowing them to exchange data and functionality. It connects front-end components with back-end services, databases, or third-party platforms, ensuring dynamic and real-time updates. This integration enhances functionality, scalability, and user experience by automating processes, reducing manual work, and enabling features like data fetching, user authentication, and payment processing.

Code Snip:

```
$ .env.local
1  NEXT_PUBLIC_SANITY_PROJECT_ID="REDACTED"
2  NEXT_PUBLIC_SANITY_DATASET="REDACTED"
3  SANITY_API_TOKEN="REDACTED"
4  |
```