Documentation Prepared By: Nadia Shaikh

Slot Timing: Saturday, 2:00 PM to 5:00 PM

Task Assigned By: Sir Ameen Alam

GIAIC

Quarter-2

DAY 4: BUILDING DYNAMIC FRONTEND COMPONENTS FOR THE MARKETPLACE

Today, I concentrated on designing and integrating **dynamic, reusable, and responsive frontend components** for the Marketplace. This work aimed to deliver a seamless and engaging user experience while adhering to best practices for performance optimization and state management. Through this process, I deepened my understanding of structuring components efficiently, managing complex application states, and enhancing scalability.

In this documentation, I'll provide an overview of the key components developed, explain their implementation, and share how I addressed the challenges encountered during the process.

The components built in this session include:

1. Product Listing Component

2. Product Detail Component

3. Cart Component

4. Wishlists Component

5. Checkout Flow Component

6. Footer Components

7. Header Components

8. Order Tracking Component

9. Search Bar Component

1. Product Listing Component

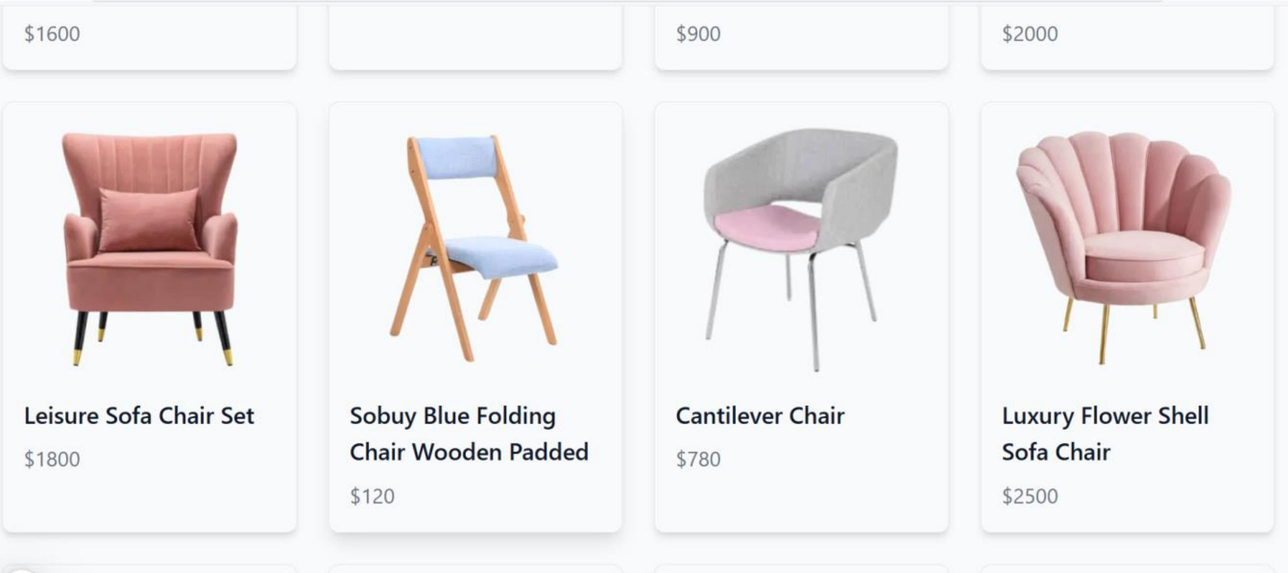
What I Learned:

The Product Listing component is essential for displaying a list of products dynamically. This task helped me understand how to handle data in React and render multiple items in a list. I also implemented filtering capabilities using categories and tags, allowing users to sort products based on their preferences.

How I Implemented It:

● I fetched product data and mapped it to render each product's name, image, and price.

● Added filters by categories and tags to refine the product list based on user selection.



$1600

$900

$2000

Leisure Sofa Chair Set
$1800

Sobuy Blue Folding Chair Wooden Padded
$120

Cantilever Chair
$780

Luxury Flower Shell Sofa Chair
$2500

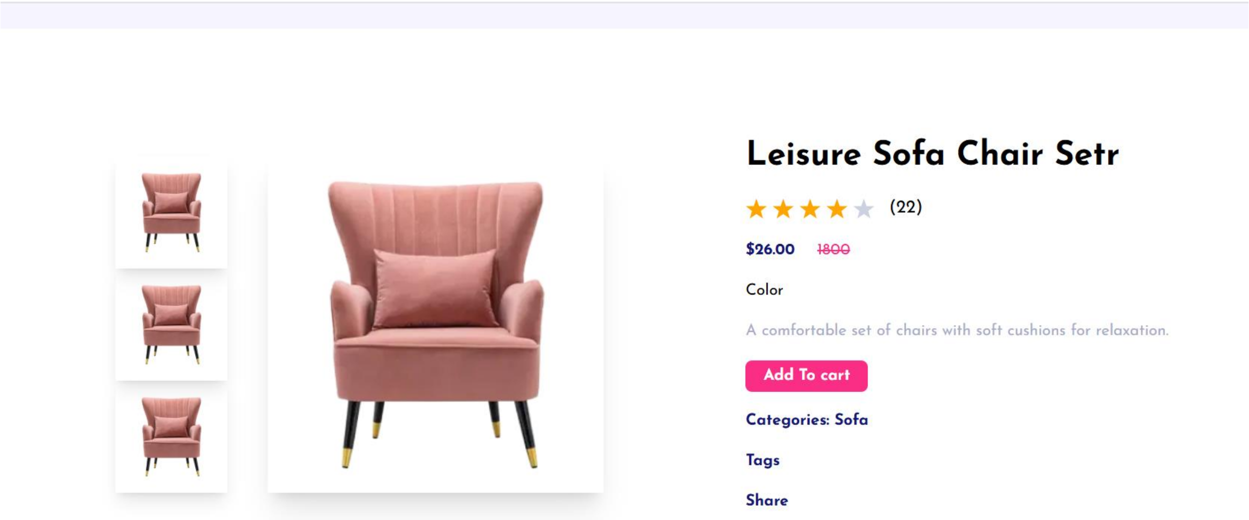
2. Product Detail Component

What I Learned:

I explored React's state management to dynamically update and display detailed product information, such as images, prices, and descriptions, based on user selection. This helped me better understand how to create interactive and responsive components.

How I Implemented It:

- Designed a dedicated product detail page that provides users with a comprehensive view of their selected item.
- Incorporated an image gallery, product description, ratings, and pricing details to create a visually appealing layout.
- Added an intuitive "Add to Cart" button for seamless shopping, ensuring a smooth user experience



Leisure Sofa Chair Setr

★★★★☆ (22)

$26.00 ~~1800~~

Color

A comfortable set of chairs with soft cushions for relaxation.

Categories: Sofa

Tags

Share

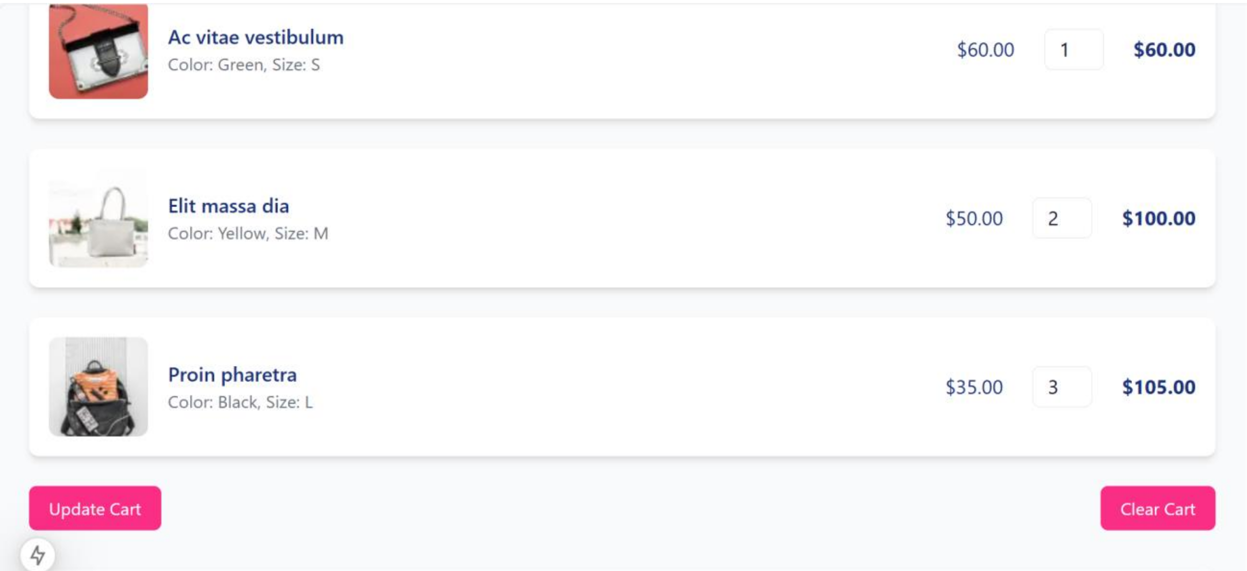
3. Cart Component :

What I Learned:

Building the Cart component helped me grasp how to manage the cart state and update it as users add or remove items. I learned how to dynamically adjust the cart's total price based on the products inside it. How I Implemented It:

● The cart component lists all the products added by the user, including quantity, price, and subtotal.

● Users can update the quantity or remove products, and the total price dynamically updates accordingly.



Ac vitae vestibulum Color: Green, Size: S	$60.00	1	**$60.00**
Elit massa dia Color: Yellow, Size: M	$50.00	2	**$100.00**
Proin pharetra Color: Black, Size: L	$35.00	3	**$105.00**

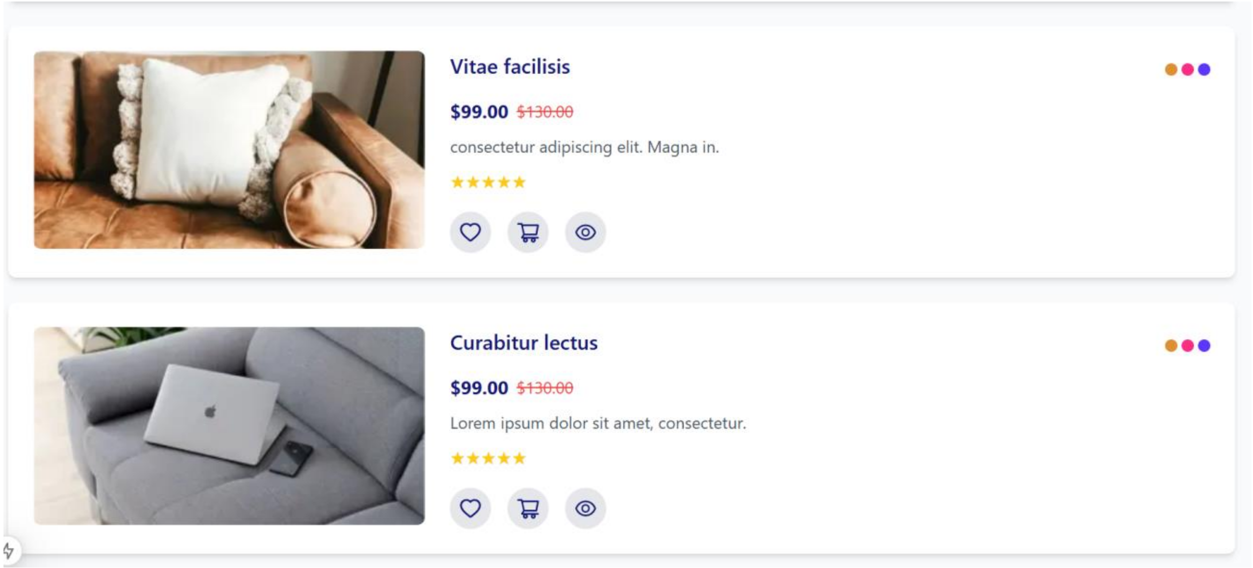
4. Wishlist Component

What I Learned:

Building the Wishlist component helped me understand how to enable users to save products for future consideration. I also learned how to implement seamless functionality for transferring items from the wishlist to the cart.

How I Implemented It:

- Created a feature that allows users to save their favorite products to a wishlist for easy access later.
- Added a user-friendly button that moves items from the wishlist to the cart, streamlining the purchasing process when the user decides to buy.



Vitae facilisis

$99.00 ~~$130.00~~

consectetur adipiscing elit. Magna in.

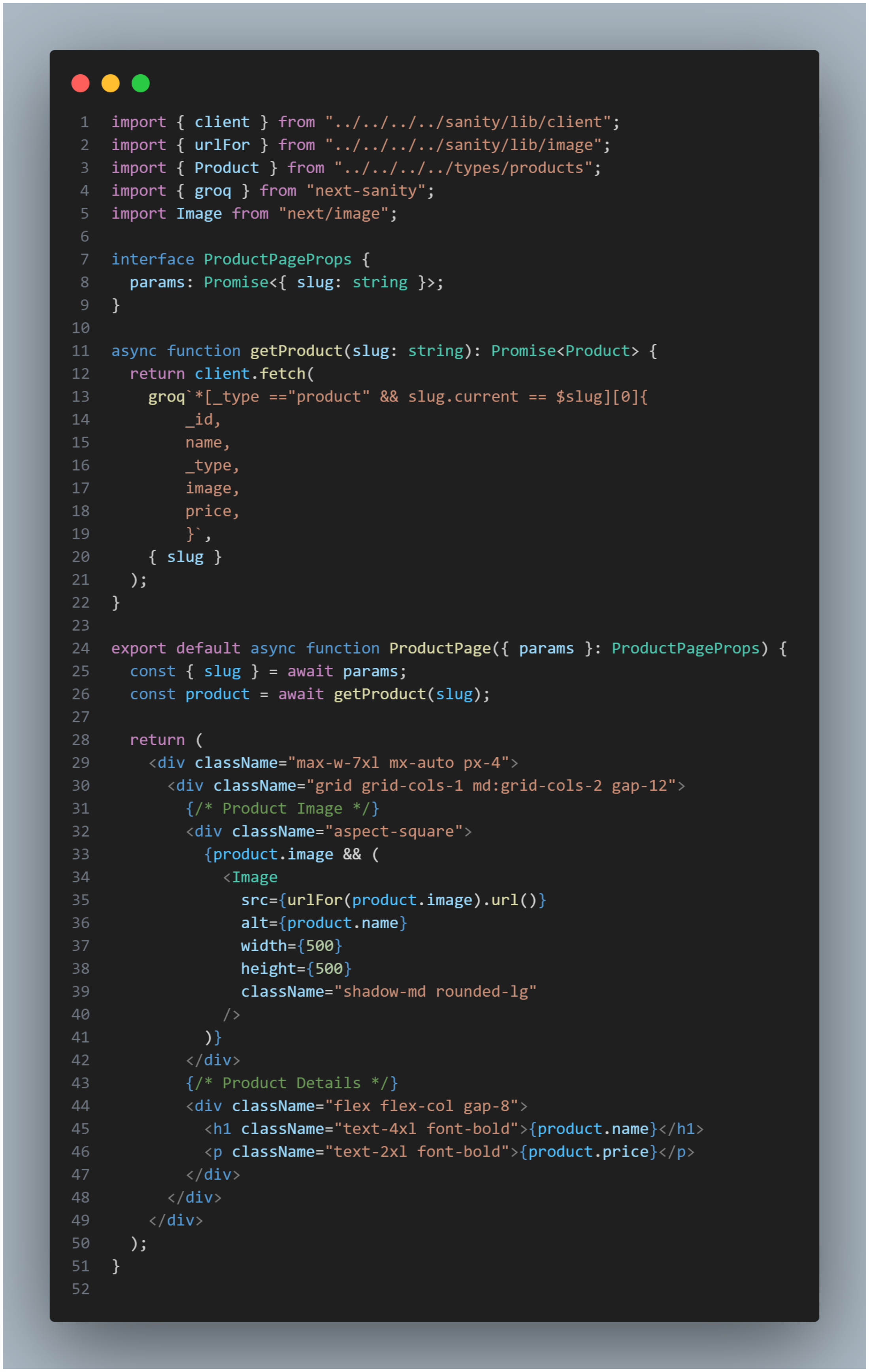
★★★★★

Curabitur lectus

$99.00 ~~$130.00~~

Lorem ipsum dolor sit amet, consectetur.

★★★★★



```tsx
import { client } from "../../../../sanity/lib/client";
import { urlFor } from "../../../../sanity/lib/image";
import { Product } from "../../../../types/products";
import { groq } from "next-sanity";
import Image from "next/image";

interface ProductPageProps {
  params: Promise<{ slug: string }>;
}

async function getProduct(slug: string): Promise<Product> {
  return client.fetch(
    groq`*[_type =="product" && slug.current == $slug][0]{
        _id,
        name,
        _type,
        image,
        price,
        }`,
    { slug }
  );
}

export default async function ProductPage({ params }: ProductPageProps) {
  const { slug } = await params;
  const product = await getProduct(slug);

  return (
    <div className="max-w-7xl mx-auto px-4">
      <div className="grid grid-cols-1 md:grid-cols-2 gap-12">
        {/* Product Image */}
        <div className="aspect-square">
          {product.image && (
            <Image
              src={urlFor(product.image).url()}
              alt={product.name}
              width={500}
              height={500}
              className="shadow-md rounded-lg"
            />
          )}
        </div>
        {/* Product Details */}
        <div className="flex flex-col gap-8">
          <h1 className="text-4xl font-bold">{product.name}</h1>
          <p className="text-2xl font-bold">{product.price}</p>
        </div>
      </div>
    </div>
  );
}

```

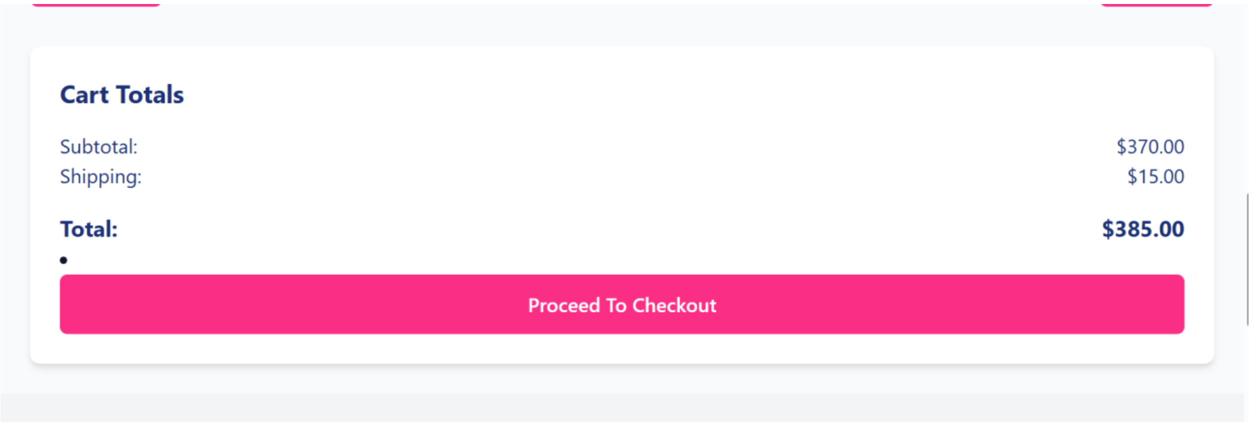
5. Checkout Flow Component

What I Learned:

Developing the Checkout Flow was a challenging yet rewarding experience. It involved managing user inputs for shipping, billing, and payment details across multiple steps. I gained valuable insights into structuring complex processes while ensuring a seamless user experience.

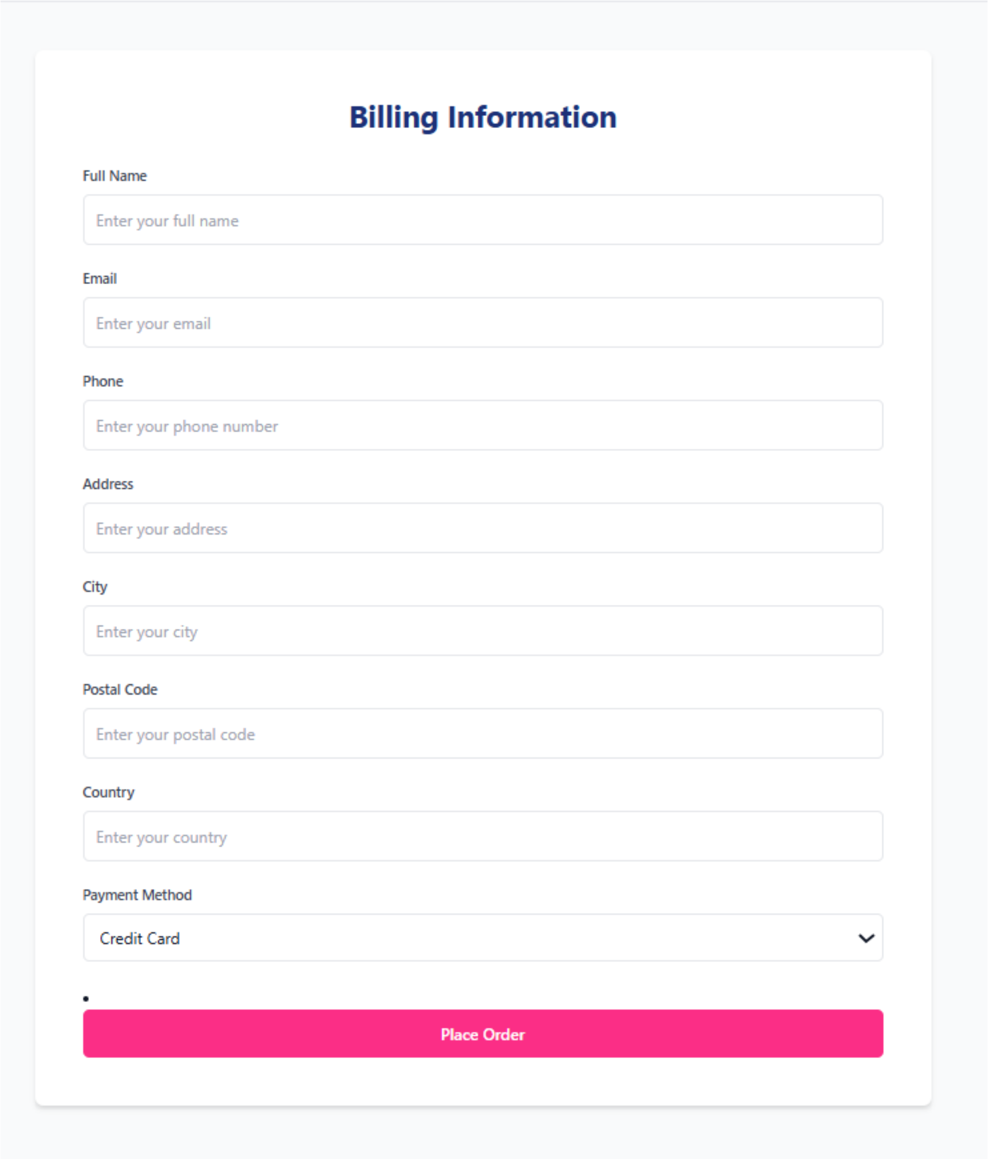
How I Implemented It:

- Designed the checkout process as a multi-step flow, including sections for shipping details, payment information, and a final order summary.
- Each step collected and validated the required information, guiding users through the process intuitively.
- At the final stage, the order was reviewed and confirmed before submission, ensuring all details were accurate.



Cart Totals

Subtotal:	$370.00
Shipping:	$15.00
Total:	**$385.00**



Billing Information

Full Name

Enter your full name

Email

Enter your email

Phone

Enter your phone number

Address

Enter your address

City

Enter your city

Postal Code

Enter your postal code

Country

Enter your country

Payment Method

Credit Card

6. Footer and Header Components

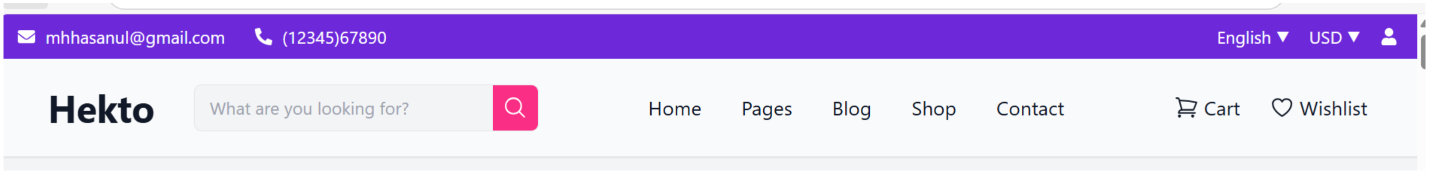
What I Learned:

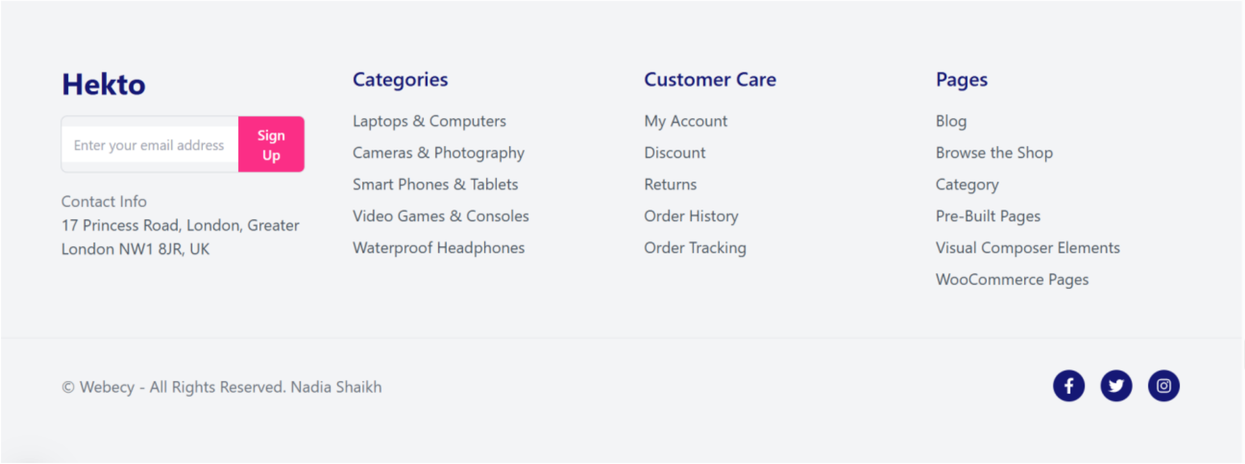
Building the Header and Footer components highlighted the significance of consistent navigation and accessibility across the website. These elements play a vital role in improving the user experience by providing quick access to essential links, contact details, and the shopping cart.

How I Implemented It:

- **Header**: Designed a visually appealing header featuring the website logo, intuitive navigation links, and a functional search bar to help users find products effortlessly.

- **Footer**: Developed an informative footer with links to key sections like About Us, Contact, and social media platforms. Added social media icons to foster engagement and keep users connected.





Hekto

Enter your email address | Sign Up

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

Categories

Laptops & Computers
Cameras & Photography
Smart Phones & Tablets
Video Games & Consoles
Waterproof Headphones

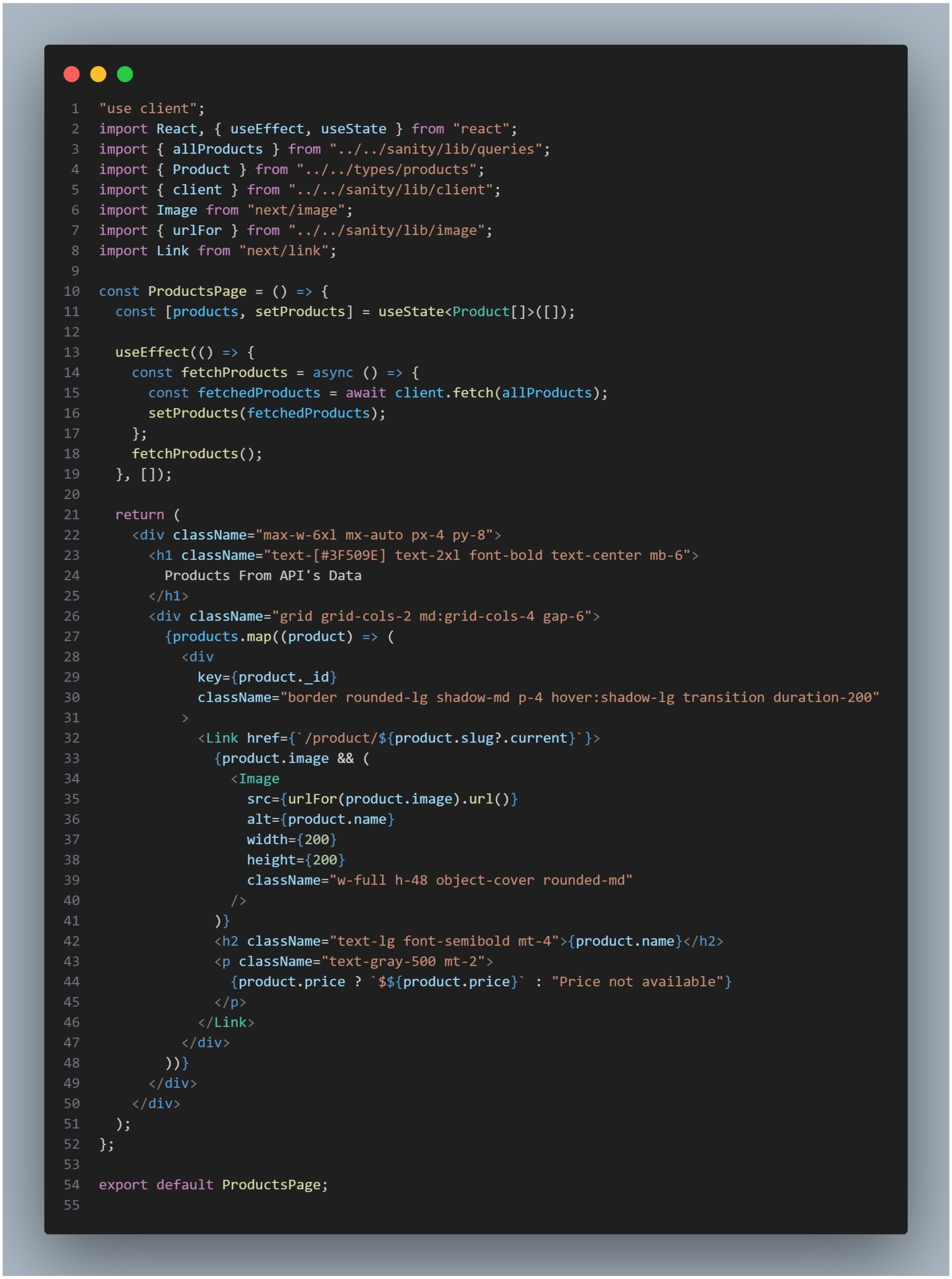
Customer Care

My Account
Discount
Returns
Order History
Order Tracking

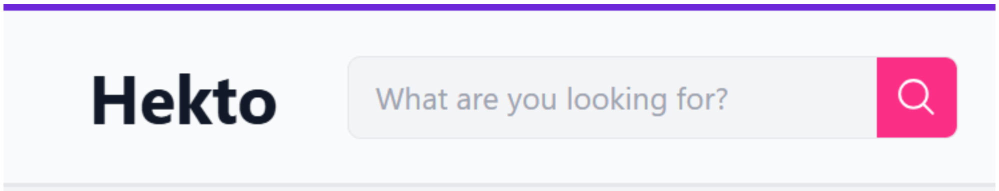
Pages

Blog
Browse the Shop
Category
Pre-Built Pages
Visual Composer Elements
WooCommerce Pages

© Webecy - All Rights Reserved. Nadia Shaikh



```
"use client";
import React, { useEffect, useState } from "react";
import { allProducts } from "../../sanity/lib/queries";
import { Product } from "../../types/products";
import { client } from "../../sanity/lib/client";
import Image from "next/image";
import { urlFor } from "../../sanity/lib/image";
import Link from "next/link";

const ProductsPage = () => {
  const [products, setProducts] = useState<Product[]>([]);

  useEffect(() => {
    const fetchProducts = async () => {
      const fetchedProducts = await client.fetch(allProducts);
      setProducts(fetchedProducts);
    };
    fetchProducts();
  }, []);

  return (
    <div className="max-w-6xl mx-auto px-4 py-8">
      <h1 className="text-[#3F509E] text-2xl font-bold text-center mb-6">
        Products From API's Data
      </h1>
      <div className="grid grid-cols-2 md:grid-cols-4 gap-6">
        {products.map((product) => (
          <div
            key={product._id}
            className="border rounded-lg shadow-md p-4 hover:shadow-lg transition duration-200"
          >
            <Link href={`/product/${product.slug?.current}`}>
              {product.image && (
                <Image
                  src={urlFor(product.image).url()}
                  alt={product.name}
                  width={200}
                  height={200}
                  className="w-full h-48 object-cover rounded-md"
                />
              )}
              <h2 className="text-lg font-semibold mt-4">{product.name}</h2>
              <p className="text-gray-500 mt-2">
                {product.price ? `$${product.price}` : "Price not available"}
              </p>
            </Link>
          </div>
        ))}
      </div>
    </div>
  );
};

export default ProductsPage;

```



Technical Report: Integration of Dynamic Components for Marketplace

Overview:

Marketplace. These components included **Product Listing**, **Cart Management**, **Checkout Flow**, and **Order Tracking**. The goal was to create an intuitive, seamless shopping experience while addressing challenge
Today's focus was on designing and implementing dynamic, user-centric components for the s through best practices in web development. Below is a breakdown of the tasks accomplished and the methodologies applied?

---

1. Product Listing Component

Implementation Highlights:

- Dynamically rendered products by mapping over an array of product data, ensuring scalability and flexibility for future updates.
- Added interactive filters for category, price range, and availability, enabling users to quickly refine their search results.
- Incorporated responsive design principles, ensuring the product grid adapts gracefully to various screen sizes for a better user experience.

Key Challenges:

- **Challenge:** Handling large product datasets while maintaining fast load times.
- **Solution:** Leveraged lazy loading techniques and optimized product images to enhance performance.

---

2. Cart Management Component

Implementation Highlights:

- Built a robust cart system allowing users to add, update, or remove products with real-time updates to the cart summary (item count and total cost).

- Integrated features like quantity adjustments and product removal for greater flexibility.

- Displayed visual indicators and confirmations for actions like “Product added to cart,” enhancing user feedback.

Key Challenges:

- **Challenge:** Managing cart state efficiently across the application.

- **Solution:** Utilized React’s Context API to share cart data globally and maintain state consistency.

---

3. Multi-Step Checkout Flow

Implementation Highlights:

- Developed a step-by-step process guiding users through the checkout journey, with the following stages:
    1. **User Information:** Collected essential details such as name, phone, and shipping address.

    2. **Payment Details:** Implemented secure forms for credit card information (card number, CVV, expiry date).

    3. **Order Summary:** Provided a detailed review of the order before submission.
- Integrated robust form validation, ensuring accurate and secure data input for fields like phone numbers and payment details.
- Generated a unique order ID upon form submission, which was seamlessly passed to the Order Tracking component.

Key Challenges:

- **Challenge:** Ensuring data security and user trust during the payment process.
- **Solution:** Simulated encryption and ensured form validation for critical data fields.

---

4. Order Tracking Component

Implementation Highlights:

- Developed an Order Tracking page where users can view the real-time status of their order (e.g., "Processing," "Shipped," "Delivered").

- Fetched and displayed relevant order details such as items purchased, delivery date, and total cost using the unique order ID generated during checkout.

- Added visual status indicators and a progress bar to enhance the user experience.

Key Challenges:

- **Challenge:** Maintaining a seamless data flow from the checkout process to the order tracking system.
- **Solution:** Implemented API integration and session storage to fetch and display order details efficiently.

Overall Challenges and Resolutions:

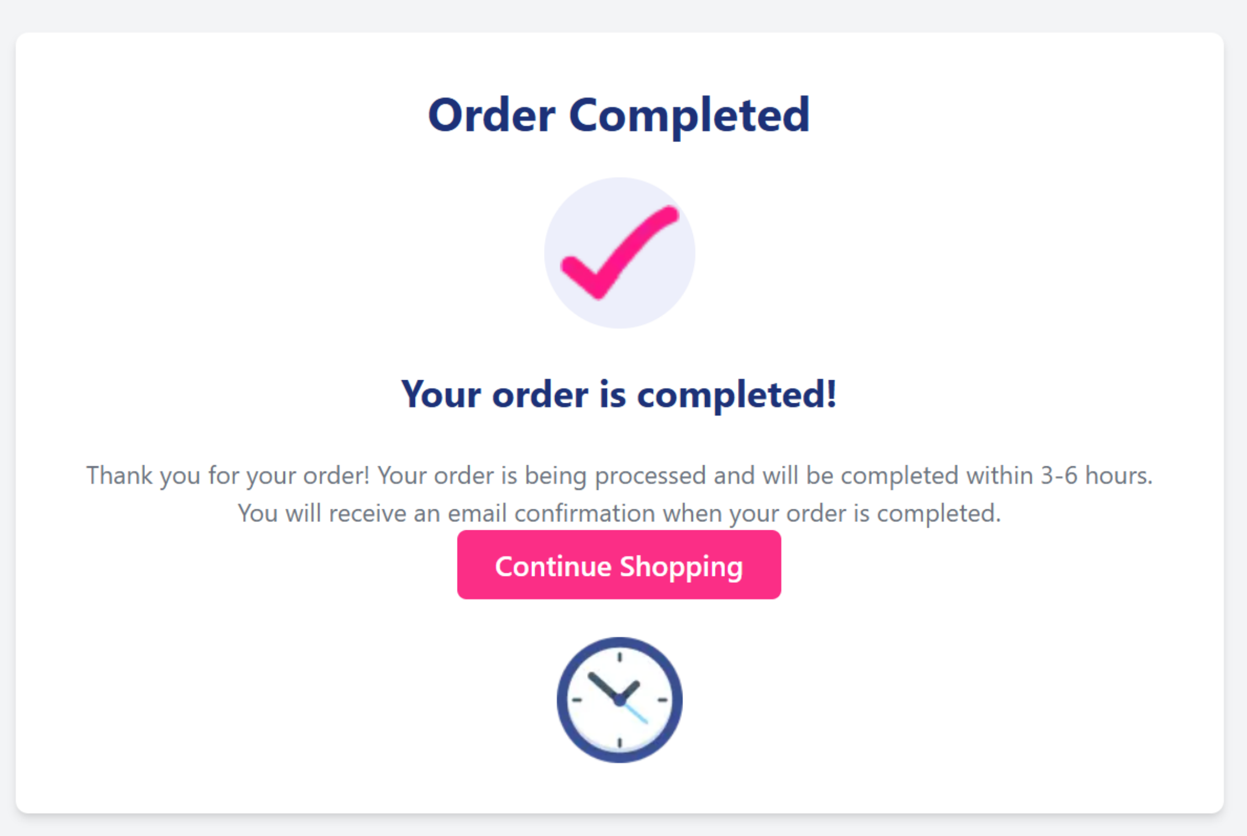
1. **State Management:** Ensuring consistent state across components.
   - **Resolution:** Used React's Context API and hooks like useReducer to streamline state management.

2. **Performance Optimization:** Handling large datasets and ensuring smooth transitions.
   - **Resolution:** Employed techniques like code splitting, lazy loading, and optimized media files.

3. **User Experience:** Simplifying complex processes like checkout and order tracking.
    - **Resolution:** Focused on clear UI/UX design and interactive feedback mechanisms.

Conclusion:

The components developed today significantly enhance the functionality and user experience of the Marketplace. By adhering to best practices and addressing technical challenges effectively, the platform is now more dynamic, responsive, and user-friendly. Future enhancements will include further optimization of API calls and the addition of features like user reviews and wishlists.



Order Completed

Your order is completed!

Thank you for your order! Your order is being processed and will be completed within 3-6 hours.
You will receive an email confirmation when your order is completed.