

Name: Noor UI Sehar
Roll Number: 00349943
Class: Friday (9 am to 12 pm)

Day 4 - Building Dynamic Frontend Components for our Marketplace

Overview:

On Day 3, I focused on building dynamic frontend components to display marketplace data fetched from **Sanity CMS**. The goal was to create a modular, reusable, and responsive frontend that aligns with real-world professional workflows. Below is a detailed breakdown of the tasks completed.

Key Objectives:

1. Build dynamic frontend components to display data from Sanity CMS.
2. Implement reusable and modular components.
3. Apply state management techniques.
4. Ensure responsive design and UX/UI best practices.
5. Replicate professional workflows for real-world client projects.

Key Components Built:

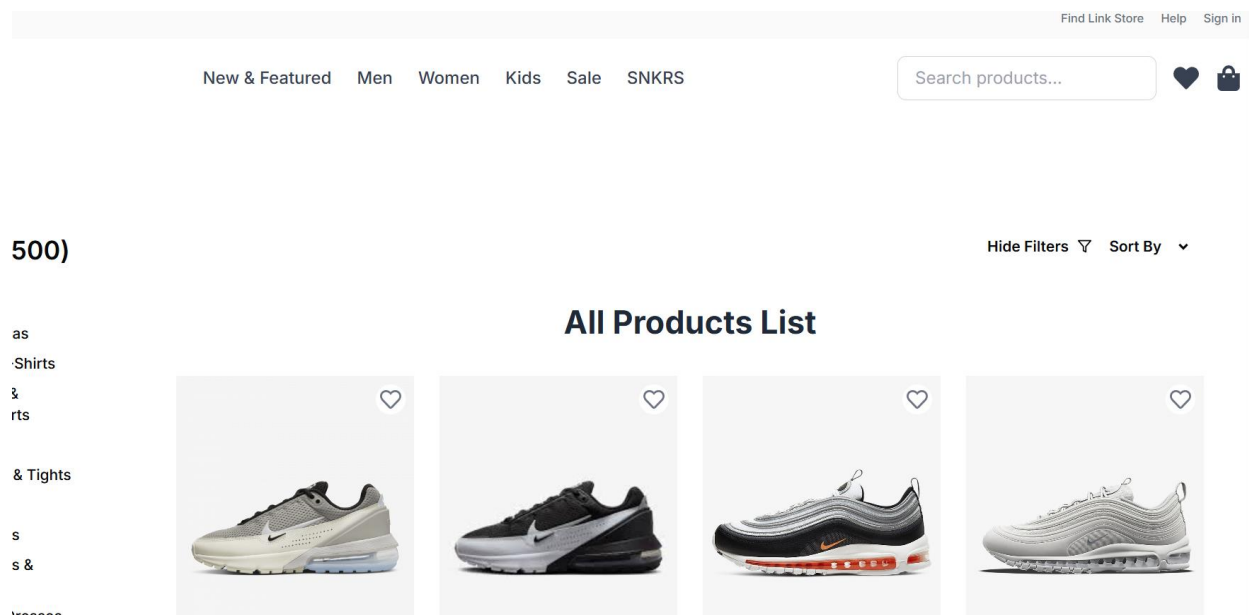
1. Product Listing Component:

- **Description:** Rendered product data dynamically in a grid layout.
- **Fields Displayed:** Product Name, Price, Image, Stock Status.
- **Implementation:** Make a ProductCard component to display each product. Fetched data from Sanity CMS using mockApi.

- **Screenshot:**

my-app > src > app > products >  page.tsx > ...

```
17
18 export default function AllProducts() {
19   const { wishlist, addToWishlist, removeFromWishlist } = useWishlist();
20   const [products, setProducts] = useState<Product[]>([]);
21   const [loading, setLoading] = useState(true);
22   const [error, setError] = useState<string | null>(null);
23
24   // Fetch products from API
25   useEffect(() => {
26     const fetchProducts = async () => {
27       try {
28         const response = await fetch('https://677ff2a50476123f76a8dd73.mockapi.io/product');
29         if (!response.ok) {
30           throw new Error("Failed to fetch products");
31         }
32         const data = await response.json();
33         setProducts(data);
34       } catch (error) {
35         setError(error instanceof Error ? error.message : "Failed to fetch products");
36       } finally {
37         setLoading(false);
38       }
39     };
40
41     fetchProducts();
42   }, []);
43
```



2. Product Detail Component:

- **Description:** Created individual product detail pages using dynamic routing in Next.js.
- **Fields Displayed:** Product Description, Price, Available Colors.
- **Implementation:** Used Next.js dynamic routing **(/product/[id])** to fetch and display product details based on the product ID.
- **Screenshot:**

```
export default function Product() {  
  return (  
    <div className="grid grid-cols-1 md:grid-cols-2 gap-8 p-6 rounded-lg">  
      <div className="flex justify-center items-center">  
        <img  
          className="w-full h-auto object-contain rounded-lg shadow-lg"  
          src={selectedProduct.img}  
          alt={selectedProduct.title}  
        />  
      </div>  
      <div className="flex flex-col justify-center space-y-6 px-4 md:px-8 lg:px-16">  
        <h1 className="font-bold text-2xl sm:text-3xl md:text-4xl text-gray-800 text-center md:text-left">  
          {selectedProduct.title}  
        </h1>  
        <h2 className="text-xs sm:text-sm md:text-base font-semibold text-gray-500 text-center md:text-left">  
          {selectedProduct.title2}  
        </h2>  
        <h4 className="text-xs sm:text-sm md:text-base text-[#9E3500] justify-between">Color: {selectedProduct.color.join(', ')}</h4>  
  
        <p className="text-sm sm:text-base md:text-lg text-gray-600 text-center md:text-left">  
          {selectedProduct.description}  
        </p>  
        <p className="text-lg sm:text-xl md:text-2xl font-semibold text-gray-800 text-center md:text-left">  
          {selectedProduct.price}  
        </p>  
        {successMessage && (  
          <div className="text-green-500 text-sm text-center md:text-left">  
            {successMessage}  
          </div>  
        )}  
        <button  
          onClick={() => addToCart(selectedProduct)}  
          className="mt-6 px-6 py-2 bg-blue-600 text-white font-semibold rounded-lg hover:bg-blue-700 focus:outline-none focus:ring-2 focus:ring-blue-600">  
          Add to Cart  
        </button>  
      </div>  
    )  
  )  
}
```



Nike Air Max 97 SE

Women's Shoes

Color: Red, White

A timeless silhouette meets contemporary flair in the Nike Air Max 97 SE Men's Shoes. With its sleek, futuristic design, premium materials, and responsive Air cushioning, this iconic model ensures all-day comfort while making a bold statement in both style and performance. Whether you're hitting the streets or heading to the gym, the Nike Air Max 97 SE combines versatility and modern design, making it the perfect addition to any wardrobe.

₹ 13,995

♡ Add to Wishlist

Add to Cart

3. Category Component:

- **Description:** Displayed categories dynamically fetched from **Sanity CMS**.
- **Functionality:** Enabled filtering of products by selected categories.
- **Implementation:** Used a dropdown menu to filter products based on categories.
- **Screenshot:**

```
    { /* sidebar */ }
    <section
      className={`flex flex-col max-w-[200px] mt-10 fixed lg:static left-0 top-24 □bg-white z-40 transform transition-transform duration-300
        ease-in-out shadow-lg ${
          isSidebarVisible ? "translate-x-0" : "-translate-x-full"
        } lg:translate-x-0`}
    >
      <h1 className="font-semibold text-sm md:text-2xl">New(500)</h1>
      <div className="font-medium text-[9px] lg:text-sm max-w-[150px] space-y-2 mt-6 flex flex-col">
        <Link href={"/shoes"} className="□ hover:text-red-500">Shoes</Link>
        <Link href={"/sports"} className="□ hover:text-blue-500">Sports Bras</Link>
        <Link href={"/top&tshirts"} className="□ hover:text-green-500">Tops & T-Shirts</Link>
        <Link href={"/hoodies"} className="□ hover:text-yellow-500">Hoodies & Sweatshirts</Link>
        <Link href={"/jackets"} className="□ hover:text-purple-500">Jackets</Link>
        <Link href={"/trousers"} className="□ hover:text-pink-500">Trousers & Tights</Link>
        <Link href={"/shorts"} className="□ hover:text-indigo-500">Shorts</Link>
        <Link href={"/tracksuits"} className="□ hover:text-teal-500">Tracksuits</Link>
        <Link href={"/rompers"} className="□ hover:text-orange-500">Jumpsuits & Rompers</Link>
        <Link href={"/skirts"} className="□ hover:text-cyan-500">Skirts & Dresses</Link>
        <Link href={"/socks"} className="□ hover:text-lime-500">Socks</Link>
        <Link href={"/accessories"} className="□ hover:text-amber-500">Accessories & Equipment</Link>
      </div>
      <div className="mt-10">
        <hr />
        <h1 className="font-semibold flex justify-between items-center text-sm mt-1">
          Gender
          <span>
            <IoIosArrowUp />
          </span>
        </h1>
        <div className="mt-2 text-[9px] lg:text-sm">
          <h1 className="flex gap-2 items-center">Men</h1>
          <h1 className="flex gap-2 items-center">Women</h1>
          <h1 className="flex gap-2 items-center">Unisex</h1>
        </div>
      </div>
    </section>
```

New(500)

- Shoes
- Sports Bras
- Tops & T-Shirts
- Hoodies & Sweatshirts
- Jackets
- Trousers & Tights
- Shorts
- Tracksuits
- Jumpsuits & Rompers
- Skirts & Dresses
- Socks
- Accessories & Equipment

Gender

Men

Women

Unisex

Kids

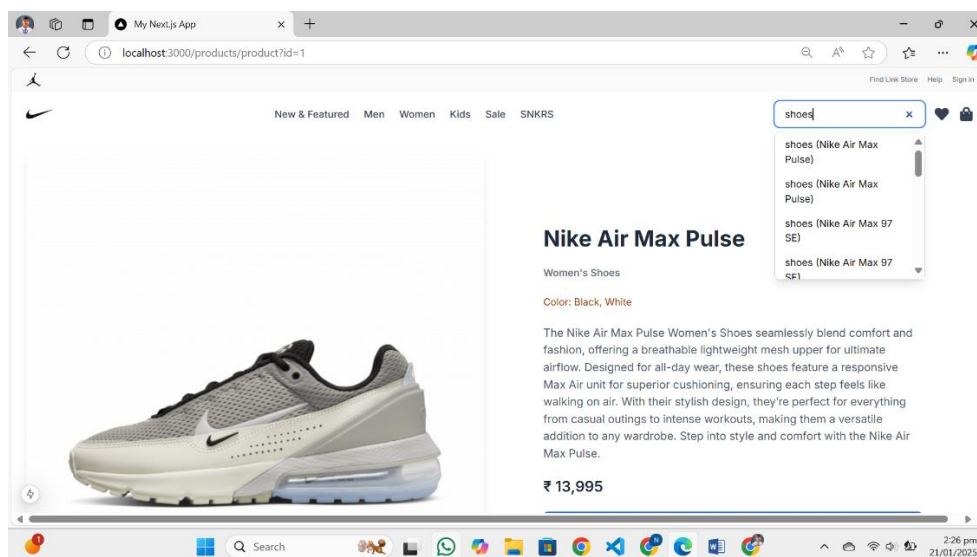
Boys

Girls

4. Search Bar:

- **Description:** Implemented search functionality to filter products by name or tags.
- **Implementation:** Used a search input field to filter products dynamically as the user types.
- **Screenshot:**

```
const SearchBar: React.FC = () => {  
  <input  
    type="search"  
    placeholder="Search products..."  
    onChange={handleChange}  
    onFocus={handleFocus}  
    onBlur={handleBlur}  
    value={searchInput}  
    className="w-full px-4 py-2 border border-gray-300 rounded-lg focus:outline-none focus:ring-2 focus:ring-blue-500 focus:border-transparent"/>  
  />  
  
  /* Dropdown List */  
  {isFocused && (  
    <div className="absolute z-10 w-full mt-2 bg-white border border-gray-300 rounded-lg shadow-lg max-h-60 overflow-y-auto">  
      {loading ? (  
        <div className="px-4 py-2 text-gray-500">Loading...</div>  
      ) : error ? (  
        <div className="px-4 py-2 text-red-500">{error}</div>  
      ) : filteredProducts.length === 0 ? (  
        <div className="px-4 py-2 text-gray-500">No products found.</div>  
      ) : (  
        <ul>  
          {filteredProducts.map((product) => (  
            <li  
              key={product.id}  
              className="px-4 py-2 hover:bg-gray-50 cursor-pointer"  
              onClick={() => handleProductClick(product.title)} // Update search input with product title  
            >  
              <Link  
                href={` /products/product?id=${product.id}`}  
                className="block w-full h-full"  
              >  
                {product.type} {product.title}  
              </Link>  
            </li>  
          )}  
        </ul>  
      )  
    )  
  )  
}
```







5. Cart Component:

- **Description:** Displayed added items, quantity, and total price.
- **State Management:** Used React Context API to manage cart items globally.
- **Implementation:** Added functionality to add/remove items and update quantities.
- **Screenshot:**

```
my-app > s C:\Users\Dell\Desktop\sehar hakaton2\my-app\src\app\Cart\page.tsx • 1 problem in this file •
31    co Untracked
140    {cart.map((item) => (
153
154        {/* Title */}
155        <p className="font-semibold truncate">{item.title}</p>
156
157        {/* Quantity Section */}
158        <div className="flex items-center justify-center space-x-2">
159            <button
160                onClick={() => handleQuantityChange(item.id, "decrease")}
161                className="px-3 py-1 bg-gray-200 text-sm rounded"
162            >
163                -
164            </button>
165            <p className="w-8 text-center">{item.quantity}</p>
166            <button
167                onClick={() => handleQuantityChange(item.id, "increase")}
168                className="px-3 py-1 bg-gray-200 text-sm rounded"
169            >
170                +
171            </button>
172        </div>
173
174        {/* Price Section */}
175        <p className="font-semibold text-lg">${cleanPrice(item.price)}</p>
176
177        {/* Remove Button */}
178        <button
179            onClick={() => handleDeleteItem(item.id)}
180            className="text-red-600 text-sm underline"
181        >
182            Remove
183        </button>
184    </div>
185  )]}
186 </div>
```

Shopping Cart

	Nike Air Max 97 SE Add to Favorites	- 1 +	\$13995 (Original: \$13995) Total: \$13995.00	
	Nike Air Max Pulse Add to Favorites	- 2 +	\$13995 (Original: \$13995) Total: \$27990.00	

Order Summary



Nike Air Max 97 SE (x1)	₹ 13995.00
Nike Air Max Pulse (x2)	₹ 27990.00
Subtotal	₹ 41985.00
Delivery Charges	₹ 200.00
Total to Pay	₹ 42185.00
Proceed to Checkout	
Continue Shopping →	

6. Wishlist Component:



- **Description:** Allowed users to save products for future reference.
- **State Management:** Used local storage to persist wishlist data.
- **Implementation:** Added a "Add to Wishlist" button on each product card.
- **Screenshot:**

```
);
<\qī\>
})
<\qī\>
)})
<\qī\>
<\qī\>
<\pñfrou>
    86w0v6 frow mīzējīzēf
    <f9h69lf cJ922ñ9w6="wū-J" \>
    >
    cJ922ñ9w6="wf-5 ■ f6xf-l6q-e00 f6xf-2w nuqelJīn6 fJ6x if6w2-cenfer,"
    onCīīcK={() => μ9uqJ686w0v6frowmīzējīzēf(if6w,īq)}
    <pñfrou
    <b cJ922ñ9w6="■ f6xf-8l9l-e00">?{cJ69u6līc6(if6w,blīc6)}<\b>
    <b cJ922ñ9w6="fouf-2ewīpōJq flnuC9f6">{if6w,fīfJ6}<\b>
    <qī\ cJ922ñ9w6="wf-4">
    \>
    cJ922ñ9w6="m-fñJJ μ-48 opJ6cf-c0v6l lonuq6q-J8,"
    9Jf={if6w,fīfJ6}
    2lc={if6w,īw8}
    <īw8
    <qī\ k6l={if6w,īq} cJ922ñ9w6="pouqel lonuq6q-J8 b-4 2μ9qom-wq">
    {mīzējīzēf.w9b((if6w) => (
    <qī\ cJ922ñ9w6="8līq 8līq-cōJ2-J wq:8līq-cōJ2-5 J8:8līq-cōJ2-3 89b-e">
    ) : (
    <b>λonl mīzējīzēf īz 6wbfλī<\b>
    {mīzējīzēf.J6u8fμ === 0 } (
    <μJ cJ922ñ9w6="f6xf-5XJ fōuf-pōJq w9-e">λonl mīzējīzēf<\μJ>
    <qī\ cJ922ñ9w6="b-e wf-JJ">
    l6fñlū (
```



Your Wishlist



Women's Shoes
Nike Air Max 97 SE
color: Red, White
MRP: ₹ 13,995



Women's Shoes
Nike Air Max Pulse
color: Pink, White
MRP: ₹ 13,995



Women's Sports
Nike Zenvy Tie-Dye
color: Purple, White
MRP: ₹ 13,995

7. Checkout Flow Component:

- **Description:** Created a multi-step form for checkout.
- **Fields Included:** Billing and shipping address, payment details (mock implementation).
- **Implementation:** Used a multi-step form with validation for each step.
- **Screenshot:**

How would you like to get your order?

Customs regulation for India require a copy of the recipient's KYC. The address on the KYC needs to match the shipping address. Our courier will contact you via SMS/email to obtain a copy of your KYC. The KYC will be stored securely and used solely for the purpose of clearing customs (including sharing it with customs officials) for all orders and returns. If your KYC does not match your shipping address, please click the link for more information.[Learn More](#)



Deliver It

Enter your name and address:

First Name *

First Name

Last Name *

Last Name

Address Line 1 *

Address Line 1

We do not ship to P.O. boxes

Address Line 2

Address Line 2

Free Delivery

Applies to orders of ₹ 14,000.00 or more. [View Details](#)

Bag



Nike Air Max 97 SE

Women's Shoes

Size:

Quantity: 3

MRP: ₹ 13995.00



Nike Air Max Pulse

Men's Shoes

Size:

Quantity: 2

MRP: ₹ 13995.00



State/Territory *

Country *

Save this address to my profile

Make this my preferred address

What's your contact information? *

Email

A confirmation email will be sent after checkout.

Phone Number *

Phone Number

A carrier might contact you to confirm delivery.

What's your PAN?

PAN

Enter your PAN to enable payment with UPI, Net Banking, or local card methods.

Save PAN details to Nike Profile *

I have read and consent to eShopWorld processing my information in accordance with the Privacy Statement and Cookie Policy. eShopWorld is a trusted Nike partner. *

Continue

Summary

Item-wise Subtotal	
Nike Air Max 97 SE (x3)	₹ 41985.00
Nike Air Max Pulse (x2)	₹ 27990.00
Subtotal	
Total Price of Items	₹ 69975.00
Delivery Charges	
Standard Delivery	₹ 200.00
Total to Pay	₹ 70175.00

8. User Profile Component:

- **Description:** Displayed user-specific details like name, email & order etc.
- **Implementation:** Fetched user data from the backend and displayed it in a profile page which is mine dashboard but I used clerk.

- **Screenshot:**

Account

Manage your account info.

Profile

Security

Profile details

Profile

N

Noorul Sehar

Update profile

Username

Set username

Email addresses

nh7220869@gmail.com

Primary

...

+ Add email address

Connected accounts

Google

• nh7220869@gmail.com

...

Welcome to Your Dashboard

Profile Information

N

Full Name

Noorul Sehar

Email Address

nh7220869@gmail.com

Joined On

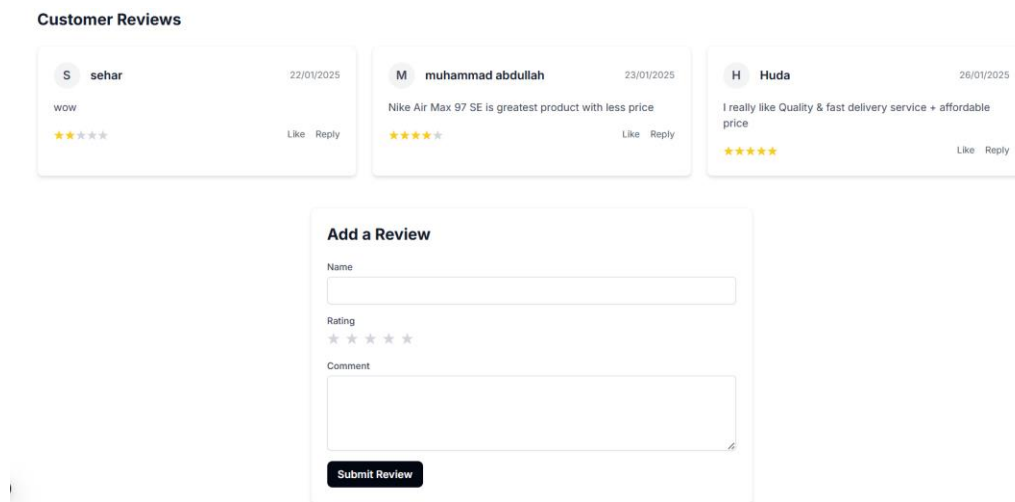
20/01/2025

Actions

Sign Out

9. Reviews and Ratings Component:

- **Description:** Allowed users to view and submit reviews for products.
- **Implementation:** Displayed average ratings and individual reviews dynamically fetched from Sanity CMS.
- **Screenshot:**



10. Pagination Component:

- **Description:** Broke down large product lists into manageable pages.
- **Implementation:** Added previous and next buttons for navigation between pages. Make a separate components.
- **Screenshot:**



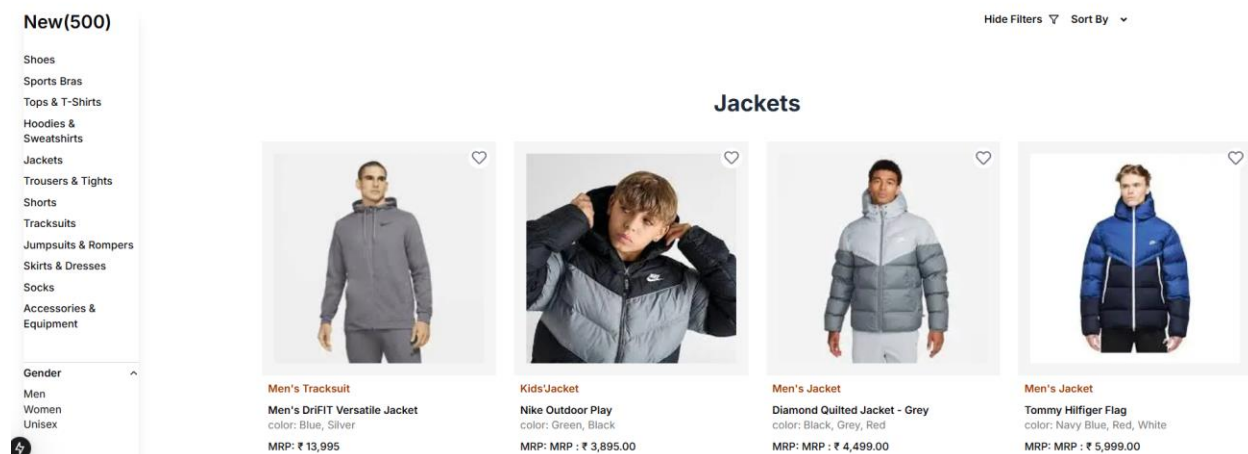
11. Filter Panel Component:

- **Description:** Provided advanced filtering options like price range sliders, brand selection, and availability toggles.
- **Implementation:** Used sliders and checkboxes to filter products dynamically.
- **Screenshot:**

```
// Fetch products from API
useEffect(() => {
  const fetchProducts = async () => {
    try {
      const response = await fetch(
        "https://677ff2a50476123f76a8dd73.mockapi.io/product"
      );
      if (!response.ok) {
        throw new Error("Failed to fetch products");
      }
      const data = await response.json();
      setProducts(data);
    } catch (error) {
      setError(
        error instanceof Error ? error.message : "Failed to fetch products"
      );
    } finally {
      setLoading(false);
    }
  };

  fetchProducts();
}, []);

// Filter products for jackets
const filteredData = products.filter(
  (product) => product.type?.toLowerCase() === "jacket"
);
```



12. Related Products Component:

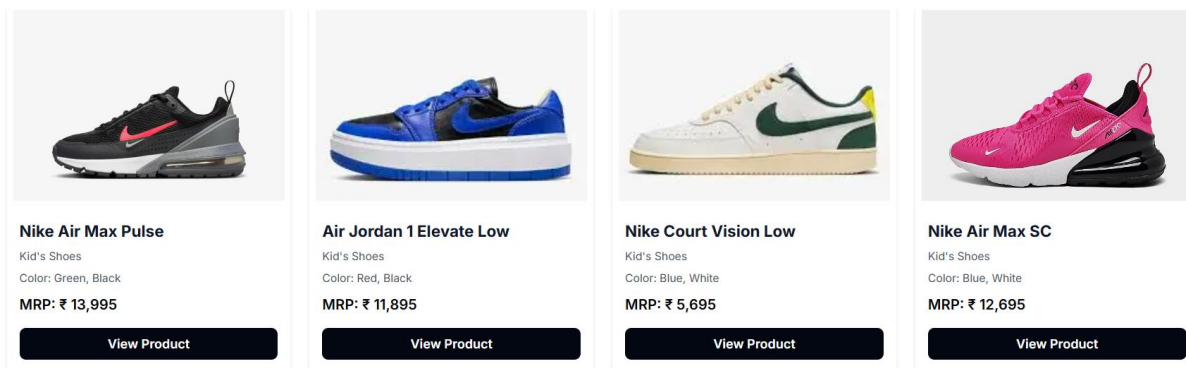
- **Description:** Suggested similar or complementary products on the product detail page.
- **Implementation:** Fetched related products based on tags or categories.
- **Screenshot:**

```

    { /* Related Products Section */}
    <div className="mt-12">
      <h2 className="text-2xl font-bold text-gray-900 mb-6">Related Products</h2>
      <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6">
        {relatedProducts.map((item) => (
          <div
            key={item.id}
            className="bg-white rounded-lg shadow-md overflow-hidden hover:shadow-lg transition-shadow duration-300"
          >
            <div className="relative w-full h-64">
              <Image
                src={item.img}
                alt={item.title}
                fill
                className="object-cover h-72 p-2 hover:scale-105 transition-transform duration-300"
              />
            </div>
            <div className="p-4">
              <h3 className="text-xl font-bold text-gray-900 mb-2">{item.title}</h3>
              <p className="text-sm text-gray-600 mb-2">{item.title2}</p>
              <p className="text-sm text-gray-600 mb-2">Color: {item.color.join(", ")}</p>
              <p className="text-lg font-semibold mb-4">MRP: {item.price}</p>
              <Link href={`/products/product?id=${item.id}`}>
                <button
                  onClick={() => router.push(`/product?id=${item.id}`)}
                  className="w-full px-4 py-2 bg-gray-900 text-white font-semibold rounded-lg hover:bg-gray-200 transition"
                >
                  View Product
                </button>
              </Link>
            </div>
          </div>
        ))}
      </div>
    </div>
  </div>
</div>

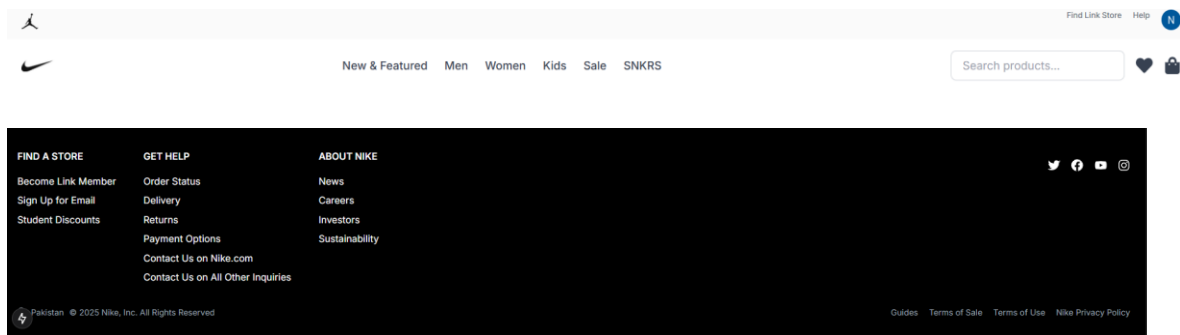
```

Related Products



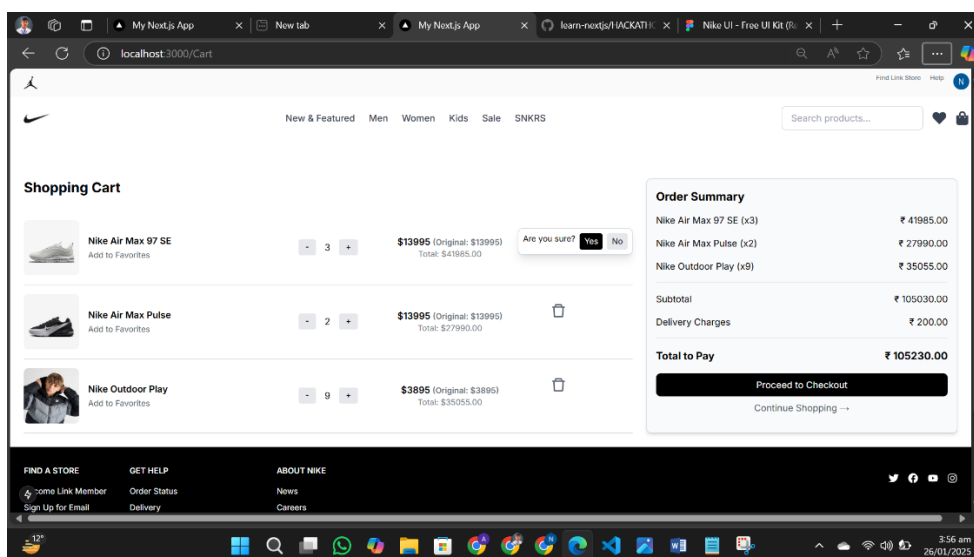
13. Footer and Header Components:

- **Description:** Built consistent navigation and branding elements.
- **Implementation:** Included links to key pages (e.g., Home, About, Contact) and ensured responsiveness.
- **Screenshot:**



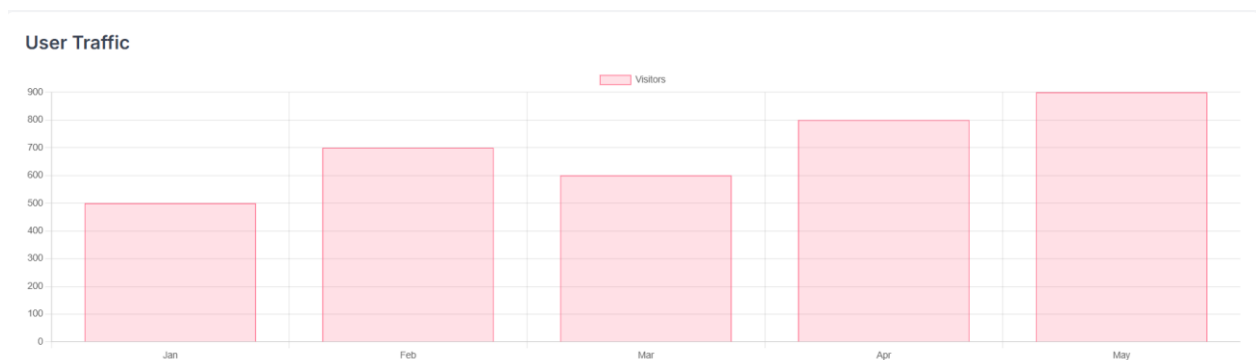
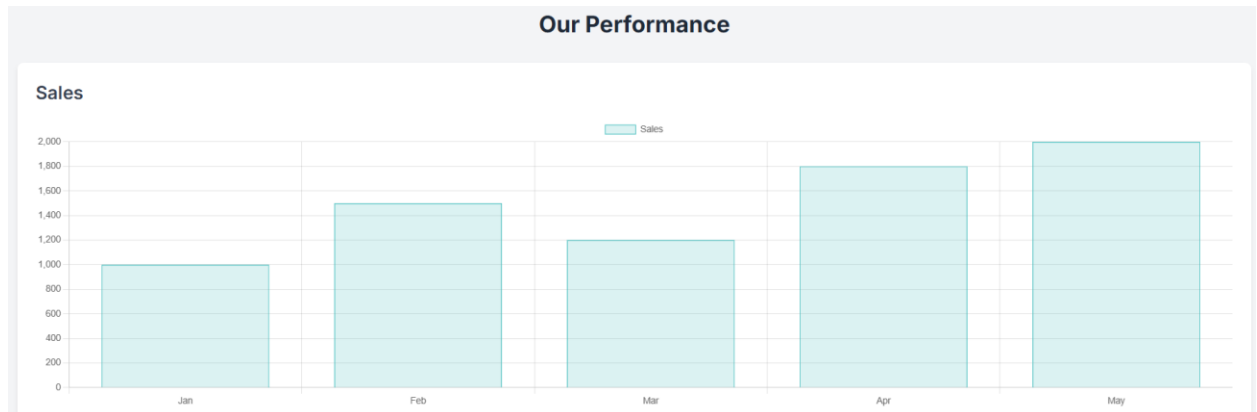
14. Notifications Component:

- **Description:** Showed real-time alerts for actions like adding to cart, errors, or successful purchases.
- **Implementation:** Used email notifications to display alerts.
- **Screenshot:**



15. Analytics Dashboard Component:

- **Description:** Showed our success story , performance or successful purchases.
- **Implementation:** Used graph to display progress.
- **Screenshot:**



16. Order Tracking Component::

- **Description:** Track order like delivered, hold, shipped etc.
- **Implementation:** Used shipengine to implement this..
- **Screenshot:**

Shipping Rates Calculator

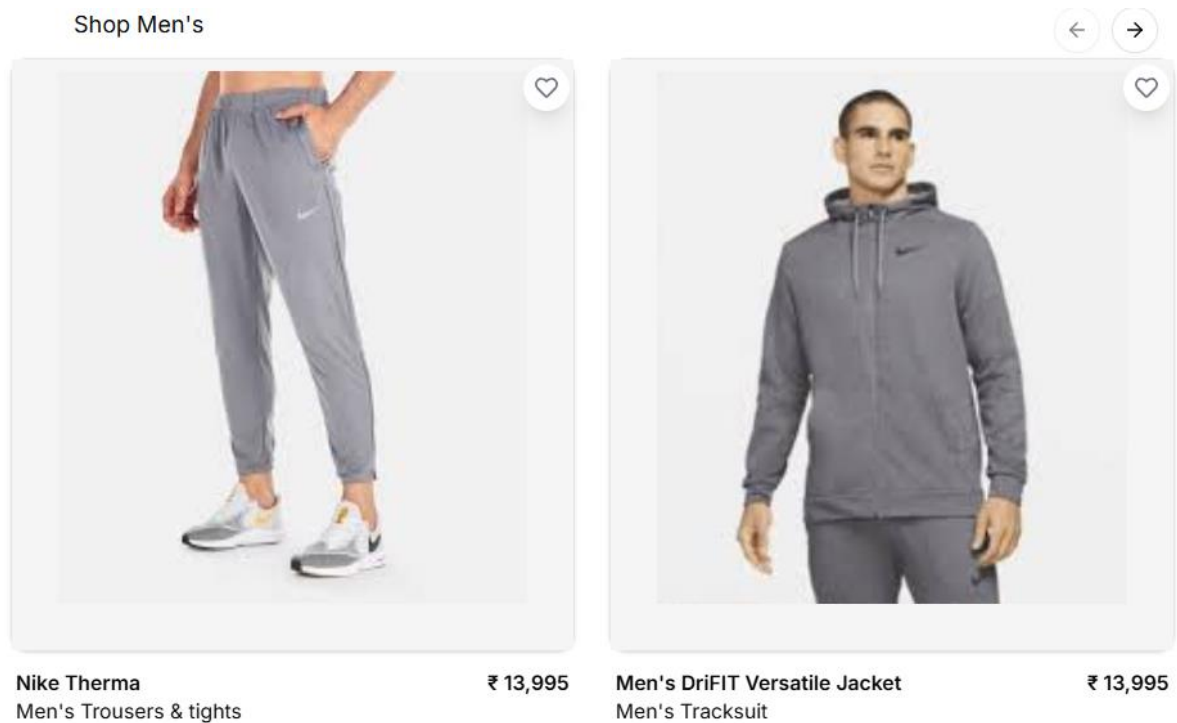
Ship To Address

Track Your Shipment

Enter Label ID or Tracking Number:

17. Carousel Component:

- **Description:** Move on next product using arrows .
- **Implementation:** Used Carousel from components/ui/carousel to implement this.
- **Screenshot:**



18. Contact Us Component:

- **Description:** Contact page for users .
- **Implementation:** Used Email Api to receive users messages .
- **Screenshot:**

The screenshot shows a 'Contact Us' form. It has a title 'Contact Us' at the top. Below the title are three input fields: 'Name' with a placeholder 'Your Name', 'Email' with a placeholder 'Your Email', and 'Message' with a placeholder 'Your Message'. At the bottom of the form is a black button labeled 'Submit Form'.

Contact Us

Name
Your Name

Email
Your Email

Message
Your Message

Submit Form


```
async function handleSubmit(event: React.FormEvent<HTMLFormElement>) {
  event.preventDefault();
  const form = event.currentTarget; // Get the form element
  const formData = new FormData(form);

  formData.append("access_key", "c2fb6e2e-1a04-4190-9aab-3e853b4604e1");

  const object = Object.fromEntries(formData);
  const json = JSON.stringify(object);

  try {
    const response = await fetch("https://api.web3forms.com/submit", {
      method: "POST",
      headers: {
        "Content-Type": "application/json",
        Accept: "application/json",
      },
    });
  }
}
```

Conclusion:

This documentation marks the culmination of Day 4, where I successfully built and implemented **18 dynamic frontend components** for our marketplace application. Each component was designed with modularity, reusability, and responsiveness in mind, ensuring a seamless user experience across devices. By leveraging **Sanity CMS** for data management, **Next.js** for dynamic routing, and **React Context API** for state management, I was able to replicate real-world professional workflows and deliver a robust frontend solution.

From **product listings** and **dynamic filtering** to **user profiles** and **order tracking**, every component was meticulously crafted to align with modern UX/UI best practices. The integration of features like **search functionality**, **cart management**, **wishlists**, and **checkout flows** ensures that the marketplace is not only functional but also user-friendly and intuitive.

The use of **Clerk** for authentication, **ShipEngine** for order tracking, and **email APIs** for notifications further enhances the application's professionalism and readiness for real-world deployment. Additionally, the inclusion of **analytics dashboards** and **carousel components** adds a layer of interactivity and engagement for users.

This project has been an incredible learning experience, allowing me to apply theoretical knowledge to practical, real-world scenarios. I am proud of the progress made and look forward to refining these components further as I continue to grow as a developer.

Thank you for following along with this journey.

