

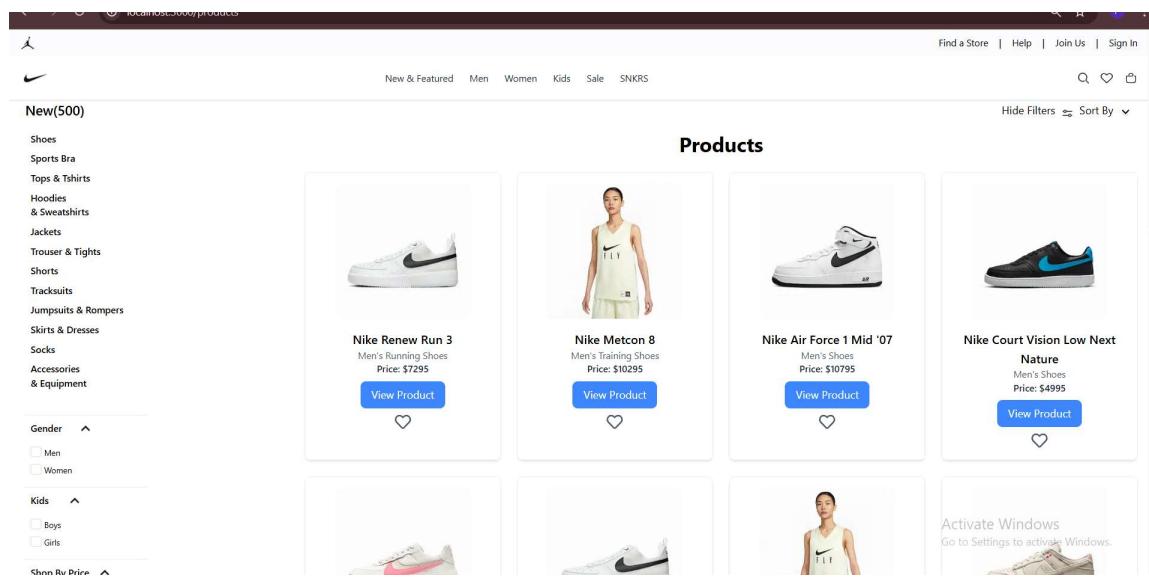
Day 4 - Dynamic Frontend

Components - [Nike E-Commerce]

1. Functional Deliverables:

Product Listing Page with Dynamic Data:

The product listing page dynamically displays all products fetched from an API .



Individual Product Detail Page with Dynamic Routing:

The product detail page dynamically fetches data based on the product's ID from the URL and displays detailed information.



Find a Store | Help | Join Us | Sign In

New & Featured Men Women Kids Sale SNKRS

Q |



Nike Dri-FIT UV Hyverse

The Nike Dri-FIT UV Hyverse graphic top offers comfort and UV protection in style. Made from soft, breathable fabric, this fitness top keeps you cool and dry during workouts or casual outings.

MRP: ₹2495

Available Colors:



Add to Cart

FIND A STORE

Reserve. Order. Pickup.

GET HELP

Order Status

ABOUT NIKE

News

FOLLOW US

```

src > app > product > [id] > page.tsx > ...
7
8  const ProductPage = ({ params }: { params: { id: string } }) => {
9    const { addItem } = useCart();
10   const [product, setProduct] = useState<any>(null);
11   const router = useRouter();
12
13   useEffect(() => {
14     const fetchProduct = async () => {
15       const query = `*[_type == "product" && _id == ${id}[0] {
16         _id,
17         productName,
18         category,
19         price,
20         inventory,
21         colors,
22         status,
23         description,
24         "imageUrl": image.asset->url
25       }`;
26
27       try {
28         const productData = await client.fetch(query, { id: params.id });
29         setProduct(productData);
30       } catch (error) {
31         console.error('Error fetching product details:', error);
32       }
33     };
34
35     fetchProduct();
36   }, [params.id]);
37
38   if (!product) {
39     return <div className="text-center py-10">Loading product details...</div>;
40   }
41
42   return (
43     <div className="container mx-auto px-6 py-10">
44       <div className="flex flex-col md:flex-row items-start gap-8">
45         {/* Product Image */}
46         <div className="flex-1 max-w-sm">
47           {product.imageUrl ? (
48             <Image
49               src={product.imageUrl}
50               alt={product.productName}
51               width={400} // Smaller image width
52               height={400}
53               className="w-full rounded-lg object-contain"
54             />
55           ) : (
56             <div className="w-full h-60 bg-gray-200 flex items-center justify-center rounded-lg">
57               No Image Available
58             </div>
59           )
60         </div>
61       {/* Product Details */}
```

Add to Cart Functionality:

The cart component allows users to add items to their shopping cart, view, and modify the cart.

Free Delivery
Applies to orders of ₹ 14 000.00 or more. [View details](#)

Bag

Nike Dri-FIT UV Hyverse
The Nike Dri-FIT UV Hyverse graphic top offers comfort and UV protection in style. Made from soft, breathable fabric, this fitness top keeps you cool and dry during workouts or casual outings.

Nike Renew Run 3
The Nike Renew Run 3 offers plush cushioning and a breathable design for smooth, comfortable runs. Perfect for beginners or seasoned athletes, it delivers support and style for every step.

Summary

Subtotal	₹ 24380.00
Estimated Delivery & Handling	Free
Total	₹ 24380.00

[Member Checkout](#)

```

: > app > cart > ✎ page.tsx > ⚡ Bag > ⚡ items.map() callback
1  'use client';
2  import Image from "next/image";
3  import { useCart } from "../context/CartContext";
4  import { PlusCircle, MinusCircle } from 'lucide-react';
5  import Link from "next/link";
6
7  export default function Bag() {
8    const { items, removeItem, updateQuantity } = useCart();
9
10   const subtotal = items.reduce((total, item) => {
11     const price = parseFloat(item.price.replace(/\^0\.\d/g, ''));
12     return total + (price * item.quantity);
13   }, 0);
14
15   return (
16     <div className="flex flex-col lg:flex-row">
17       <div className="flex-1 bg-white p-4 lg:w-2/3">
18         <div className="bg-[#f0f0f0] w-auto mx-2 p-3">
19           <p className="text-[13px] font-medium">Free Delivery</p>
20           <div className="flex gap-2">
21             <p className="text-[10px] xl:text-[12px] lg:text-[12px]">
22               Applies to orders of ₹ 14 000.00 or more.
23             </p>
24             <p className="text-[10px] font-medium underline xl:text-[12px] lg:text-[12px]">
25               View details
26             </p>
27           </div>
28         </div>
29         <h1 className="text-[18px] font-medium ml-2 mt-2">Bag</h1>
30       </div>
31       {items.map((product) => (
32         <div
33           key={product.id}
34           className="flex justify-between items-center p-2 border-b border-gray-100"
35         >
36           <Image
37             src={product.image}
38             alt={product.productName}
39             width={70}
40             height={70}
41             className="w-[70px] h-[70px]"
42           />
43           <div className="flex-1 ml-4">
44             <h3 className="text-[14px] font-semibold">{product.productName}</h3>
45             <p className="text-[12px] text-[#757575]">{product.detail}</p>
46           </div>
47         </div>
48       )}
49     </div>
50   )
51 }
52 
```

Checkout Page:

The checkout page allows users to review their cart and proceed

with payment.

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](#)

Enter your name and address:

We do not ship to P.O. boxes

- Save this address to my profile
 Make this my preferred address

Order Summary

Subtotal ₹ 24380.00

Delivery/Shipping Free

Total ₹ 24380.00

(The total reflects the price of your order, including all taxes and duties)

Arrives in 5-7 business days



Nike Dri-FIT UV Hyverse
Qty 1
₹2495



Nike Renew Run 3
Qty 3
₹7295

Activate Windows

For an additional activation fee of ₹100/-

```
  ↵ nextconfig.mjs w      ↵ page.tsx .../products 0      ↵ pages.tsx .../wishlist 0      ↵ layout.tsx M      ↵ pages.tsx .../cart
: app > checkout > page.tsx > Checkout
4 import { useCart } from '../context/cartContext';
5
6 export default function Checkout() {
7   const [isConfirmed, setIsConfirmed] = useState(false);
8   const { items, clearCart } = useCart();
9
10  const subtotal = items.reduce((total, item) => {
11    const price = parseFloat(item.price.replace(/[^0-9.]/g, ''));  

12    return total + (price * item.quantity);
13  }, 0);
14
15  const handleSubmit = (e: React.FormEvent<HTMLFormElement>) => {
16    e.preventDefault();
17    setIsConfirmed(true);
18    clearCart();
19  };
20
21  if (isConfirmed) {
22    return (
23      <div className="flex items-center justify-center min-h-[50vh] mt-6">
24        <div className="text-center">
25          <div className="flex flex-col justify-center items-center mb-4">
26            <Image src="/order.png" alt="check" className="w-[40px] h-[40px]" />
27          </div>
28          <h1 className="text-2xl font-bold mb-4">Your order is confirmed!</h1>
29          <p className="text-gray-600">Thank you for shopping with us.</p>
30        </div>
31      </div>
32    );
33  }
34
35  return (
36    <div className="p-4">
37      <form onSubmit={handleSubmit}>
38        <div className="flex flex-col lg:flex-row lg:justify-between lg:gap-8">
39          <div className="lg:w-2/3 xl:w-3/5">
40            <div>
41              <h1 className="font-medium text-[16px]">
42                How would you like to get your order?
43              </h1>
44              <p className="font-light text-[14px]">
45                Customs regulation for India require a copy of the recipient's
46                KYC. The address on the KYC needs to match the shipping address.
47                Our courier will contact you via SMS/email to obtain a copy of
48                your KYC. The KYC will be stored securely and used solely for the
49                purpose of clearing customs (including sharing it with customs).
50            </div>
51          </div>
52        </div>
53      </form>
54    </div>
55  );
56}
57
```

Wishlist:

The wishlist component allows users to add items to their wishlist.

The screenshot shows the Nike website's 'My Wishlist' page. At the top, there's a navigation bar with links for 'Find a Store', 'Help', 'Join Us', and 'Sign In'. Below the navigation is a search bar and a user icon. The main title 'My Wishlist' is centered above a grid of three product cards. Each card displays a product image, its name, a brief description, and a 'View Product' button.

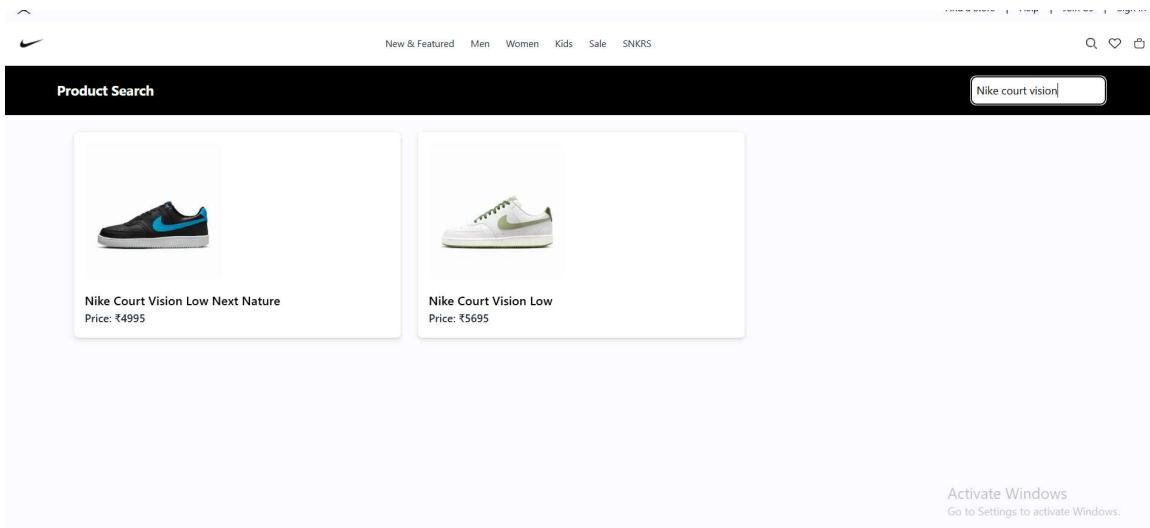
Product	Description	Action
Nike Renew Run 3	Men's Running Shoes Price: \$7295	View Product
Nike Court Vision Low Next Nature	Men's Shoes Price: \$4995	View Product
Nike Waffle One SE	Women's Shoes Price: \$7895	View Product

At the bottom of the page is a footer with sections for 'FIND A STORE', 'GET HELP', 'ABOUT NIKE', and 'FOLLOW US'. It also includes links for 'Become Link Member', 'Sign Up for Email', 'Student Discounts', 'Order Status', 'Delivery', '>Returns', 'News', 'Careers', and 'Investors'. Social media icons for Twitter, Facebook, YouTube, and Instagram are present in the 'FOLLOW US' section.

```
> app > wishlist > 🌐 page.tsx > 🏷 WishlistPage
4   import { useState, useEffect } from 'react';
5   import Image from 'next/image';
6   import Link from 'next/link';
7
8   const WishlistPage = () => {
9     const [wishlist, setWishlist] = useState<any>([]);
10
11   useEffect(() => {
12     // Load wishlist from localStorage
13     const storedWishlist = JSON.parse(localStorage.getItem('wishlist') || '[]');
14     setWishlist(storedWishlist);
15   }, []);
16
17   // Function to remove item from wishlist
18   const removeFromWishlist = (productId: string) => {
19     const updatedWishlist = wishlist.filter((product) => product._id !== productId);
20     setWishlist(updatedWishlist);
21     localStorage.setItem('wishlist', JSON.stringify(updatedWishlist));
22   };
23
24   if (!wishlist || wishlist.length === 0) {
25     return (
26       <div>
27         <p className="text-center mt-10">Your wishlist is empty.</p>
28       </div>
29     );
30   }
31
32   return [
33     <div className="flex flex-col">
34
35       /* Wishlist Items */
36       <div className="container mx-auto p-4">
37         <h1 className="text-3xl font-bold mb-6 text-center">My Wishlist</h1>
38         <div className="grid grid-cols-1 sm:grid-cols-2 lg:grid-cols-4 gap-6">
39           {wishlist.map((product: any) => (
40             <div
41               key={product._id}
42               className="border rounded-lg p-4 shadow-md flex flex-col items-center relative"
43             >
44               /* Cross icon to remove */
45               <button
46                 onClick={() => removeFromWishlist(product._id)}
47                 className="absolute top-2 right-2 text-xl text-gray-600 hover:text-red-500 transition-colors"
48               >
49             </div>
50           )
51         )
52       </div>
53     </div>
54   ];
55 }
```

Search Bar:

The search bar allows users to filter products based on keywords.



```
src > app > SearchPage > page.tsx > ProductSearchPage > useEffect() callback
7
8  const ProductSearchPage = () => {
9    const [products, setProducts] = useState<any>([]);
10   const [searchResults, setSearchResults] = useState<any>([]);
11   const [searchQuery, setSearchQuery] = useState('');
12   const router = useRouter();
13
14   useEffect(() => [
15     const fetchProducts = async () => {
16       const query = `*[_type == "product"] {
17         _id,
18         productName,
19         category,
20         price,
21         inventory,
22         colors,
23         status,
24         description,
25         "imageUrl": image.asset->url
26       }`;
27
28       try {
29         const productsData = await client.fetch(query);
30         setProducts(productsData);
31         setSearchResults(productsData);
32       } catch (error) {
33         console.error('Error fetching products:', error);
34       }
35     };
36
37     fetchProducts();
38   ], []);
39
40   const handleSearch = (query: string) => {
41     setSearchQuery(query);
42     if (!query.trim()) {
43       setSearchResults(products);
44     } else {
45       const filteredResults = products.filter((product) =>
46         product.productName.toLowerCase().includes(query.toLowerCase())
47       );
48       setSearchResults(filteredResults);
49     }
50   };
51 }
```

Steps Taken to Build and Integrate Components:

Product Listing: Dynamically fetches and displays products in a grid layout.

Product Detail Page: Fetches product details based on dynamic routing.

Add to Cart: Users can add/remove items from the cart.

Checkout Page: Users can proceed with payment for the products in their cart.

Wishlist: Users can add/remove products to/from their wishlist.

Search Bar: Filters products dynamically based on user input.

Challenges Faced and Solutions Implemented:

Data Handling: Handling dynamic fetching and displaying of data from local storage.

State Management: Efficient state updates for cart, wishlist, and search filtering.

CheckList:

Code Quality ✓

Styling ✓

Responsiveness ✓

Functionality ✓