

# Hackathon 3 Day 4

## Dynamic Frontend Components ~ [NIKE]

Prepared by: Sunaina Ismail

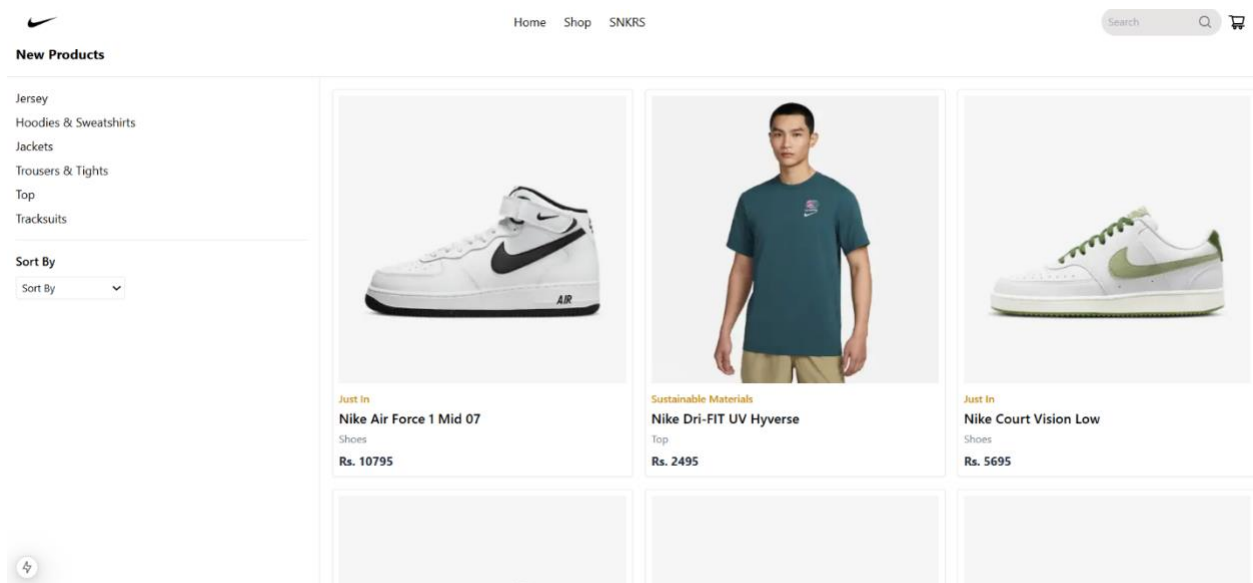
### 1. Functional Deliverables

#### ○ Product Listing page:

A layout showcasing the product listing page, including dynamic product details.

It includes:

- ProductName
- Price
- Image



#### ○ Product Details Page:

Dynamic product Detail Pages. It Includes:

- ProductName
- Price
- Image

- Description
- Stock
- Add to Cart button
- Colors



### Nike Air Force 1 Mid 07

The Nike Air Force 1 Mid '07 delivers timeless style with premium leather and a mid-cut design. Perfect for everyday wear, it provides exceptional comfort and durability. The iconic Air-Sole cushioning adds responsive support for long-lasting performance, ensuring all-day comfort. Whether on the court or the street, this sneaker combines classic design with modern functionality.

Color: ☐ [Stock 0](#)

Rs. 10795

Out of Stock

○ **Filters and Sort by:**

**New Products**

- Jersey
- Hoodies & Sweatshirts
- Jackets
- Trousers & Tights
- Top
- Tracksuits

Sort By  
Price: Low to High ▾



**Just In**  
**Nike Standard Issue Basketball Jersey**  
Jersey  
**Rs. 2895**



**Best Seller**  
**Nike Metcon 8**  
Jersey  
**Rs. 10295**

- **Add to Cart:**  
Products Successfully adds to the cart.

### Your Cart

2 items



Nike Standard Issue Basketball Jersey  
Color: Red  
Quantity: 1

Rs.2895



Nike Dri-FIT UV Hyverse  
Color: Teal  
Quantity: 2

Rs.4990



### Summary

Subtotal Rs.7885

Total Rs.7885

Member Checkout

7

- **Billing:**

### Billing details

First Name

Last Name

Country / Region

Province

City

Town / Area

Street Address

ZIP Code

Phone

### Product

### Subtotal

Nike Standard Issue Basketball Jersey * 1	Rs. 2895
Nike Dri-FIT UV Hyverse * 2	Rs. 4990
Subtotal	Rs. 7885.00
Total	<b>Rs. 7885.00</b>

### Select Payment Method

- ☐ Easypaisa
- ☐ JazzCash
- ☐ Bank Transfer

The selected method will guide you to securely complete the payment. Please use your Order ID as the payment reference.

Your personal data will be used to support your experience throughout this website, to manage access to your account, and for other purposes described in our [privacy policy](#).

Place Order

- **Order Tracking:**

Display updates on Order status.

## Track Your Order

Enter your phone number to track your order status.

TRACK ORDER

## 2. Code Snippets

- ProductCard Component Code

```
"use client";
import Image from "next/image";
import Link from "next/link";
import { AllProducts } from "@sanity/types/type";

interface ProductCardProps {
  product: AllProducts;
}

export default function ProductCard({ product }: ProductCardProps) {
  const isSoldOut = product.inventory === 0;
  return (
    <div className="p-2 border rounded shadow-sm">
      <Link href={` /product/${product.slug}`}>
        <Image
          src={product.image}
          alt={product.productName}
          width={700}
          height={700}
          className="mb-2 object-cover"
        />
      </Link>
      {isSoldOut ? (
        <span className="text-sm font-medium text-red-600">Sold Out</span>
      ) : (
        <span className="text-sm font-medium text-yellow-600">
          {product.status}
        </span>
      )}
      <h3 className="font-semibold text-lg mb-1">{product.productName}</h3>
      <span className="block text-sm text-gray-500 mb-2">
        {product.category}
      </span>
      <p className="font-bold text-gray-800">Rs. {product.price}</p>
    </div>
  );
}
```

- Dynamic product Details Component Code

```

import { fetchProductBySlug } from "@sanity/schemaTypes/data-fetch-utils";
import ProductDetailCard from "@components/Home/ProductDetailCard";
import { notFound } from "next/navigation";

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

const Product = async ({ params }: ProductDetailsProps) => {
  const { slug } = await params;
  const product = await fetchProductBySlug(slug);

  if (!product) {
    notFound();
  }

  return (
    <div>
      <ProductDetailCard
        productName={product.productName}
        description={product.description}
        price={product.price}
        imageUrl={product.imageUrl}
        colors={product.colors}
        inventory={product.inventory}
        productId={product._id}
      />
    </div>
  );
};

export default Product;

```

- Cart Code

```

"use client"
import Link from "next/link";
import Image from "next/image";
import { useCart } from "../context/cartContext";

export default function Cart() {
  const { cartItems, addToCart, removeFromCart } = useCart();

  const handleIncrement = (id: string) => {
    const item = cartItems.find((cartItem) => cartItem.id === id);
    if (item) {
      addToCart({ ...item, quantity: 1 });
    }
  };

  const handleDecrement = (id: string) => {
    const item = cartItems.find((cartItem) => cartItem.id === id);
    if (item && item.quantity > 1) {
      addToCart({ ...item, quantity: -1 });
    }
  };

  const handleRemove = (id: string) => {
    removeFromCart(id);
  };

  const calculateSubtotal = () => {
    cartItems.reduce((total, item) => total + item.price * item.quantity, 0);
  };

  return (
    <div className="min-h-screen py-24 px-3 sm:px-6 md:px-9 lg:px-32">
      <div className="max-w-[1152px] mx-auto py-4 flex flex-col lg:flex-row">
        {/* Cart Items Section */}
        <div className="w-full lg:w-2/3">
          <h5 className="text-basse sm:text-lg md:text-2xl font-semibold mb-3">Your Cart </h5>

          <h5 className="text-sm sm:text-base md:text-lg font-semibold mb-3">
            {cartItems.length === 0 ? "Your cart is empty" : `${cartItems.length} items`}
          </h5>

          {cartItems.map((item) => (
            <div
              key={item.id}
              className="border-b border-gray-300 py-2 flex items-start sm:py-3 md:py-4"
            >

```