

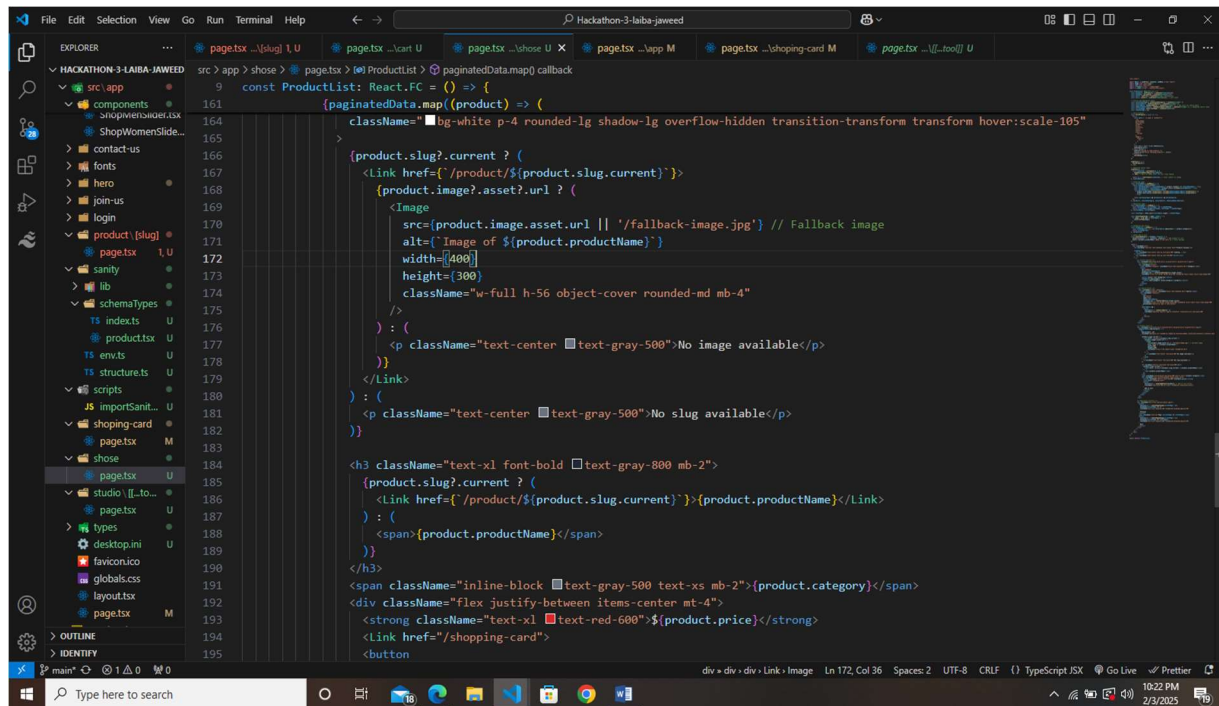
# Day 4 Building Dynamic Frontend Component for Marketplace

Laiba Jaweed

## Key Components :

### Product Listing Component

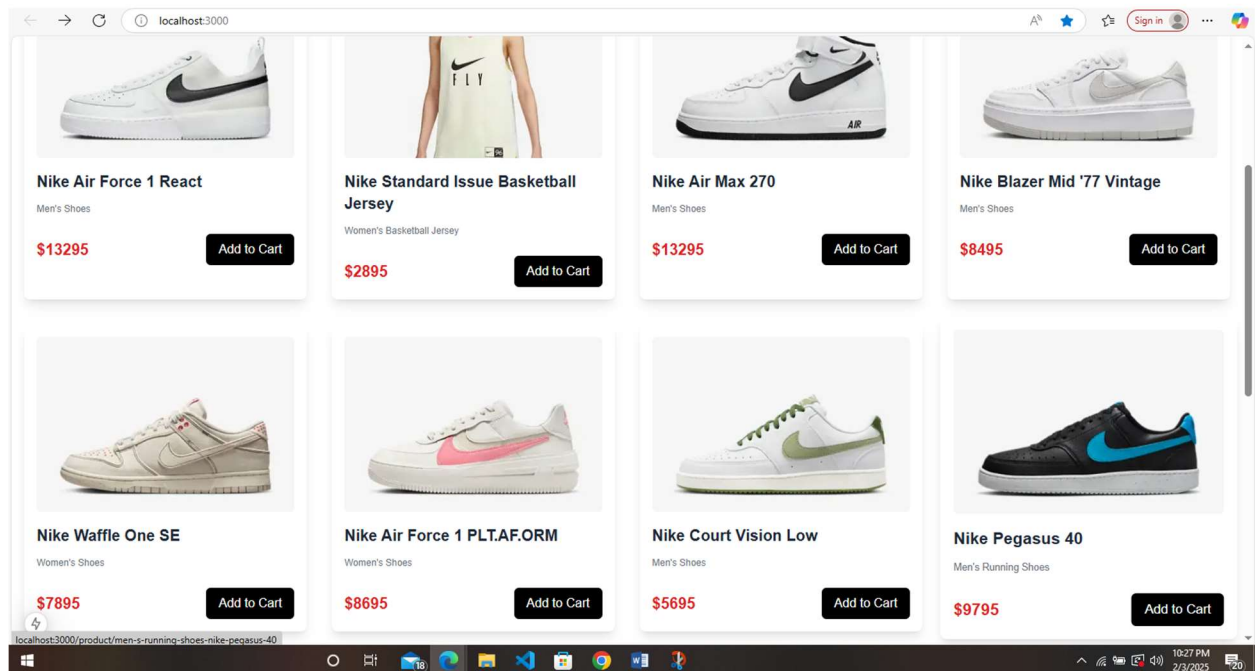
#### Code Snippet:

A screenshot of a code editor window titled 'Hackathon-3-laiba-jaweed'. The editor shows a React component for product listing. The code includes a constant 'ProductList' and a function 'ProductList: React.FC' that maps paginated data to a list of products. The component uses Tailwind CSS for styling and includes a fallback image for missing product images. The code is as follows:

```
const ProductList: React.FC = () => {
  {paginatedData.map((product) => {
    className="bg-white p-4 rounded-lg shadow-lg overflow-hidden transition-transform transform hover:scale-105"
  }
  {product.slug?.current ? (
    <Link href={`/product/${product.slug.current}`}>
      {product.image?.asset?.url ? (
        <Image
          src={product.image.asset.url || '/fallback-image.jpg'} // Fallback image
          alt={`Image of ${product.productName}`}
          width={400}
          height={300}
          className="w-full h-56 object-cover rounded-md mb-4"
        />
      ) : (
        <p className="text-center text-gray-500">No image available</p>
      )
    ) : (
      <p className="text-center text-gray-500">No slug available</p>
    )
  ) : (
    <h3 className="text-xl font-bold text-gray-800 mb-2">
      {product.slug?.current ? (
        <Link href={`/product/${product.slug.current}`}>{product.productName}</Link>
      ) : (
        <span>{product.productName}</span>
      )
    }</h3>
    <span className="inline-block text-gray-500 text-xs mb-2">{product.category}</span>
    <div className="flex justify-between items-center mt-4">
      <strong className="text-xl text-red-600">{product.price}</strong>
      <Link href="/shopping-card">
        <button
```

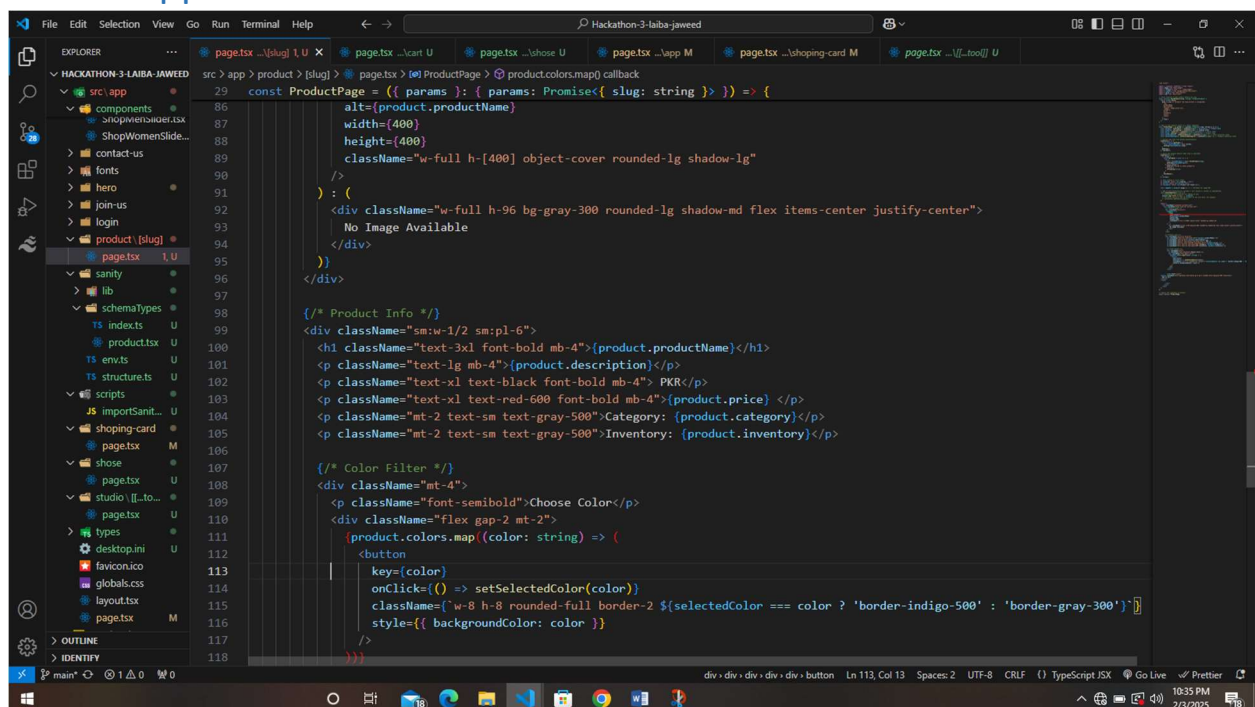
#### Output Snippet

## Product Display



## Dynamic Routing Component

### Code Snippet:



## Dynamic Routing in Next.js:

Dynamic routing allows you to create routes based on URL parameters.

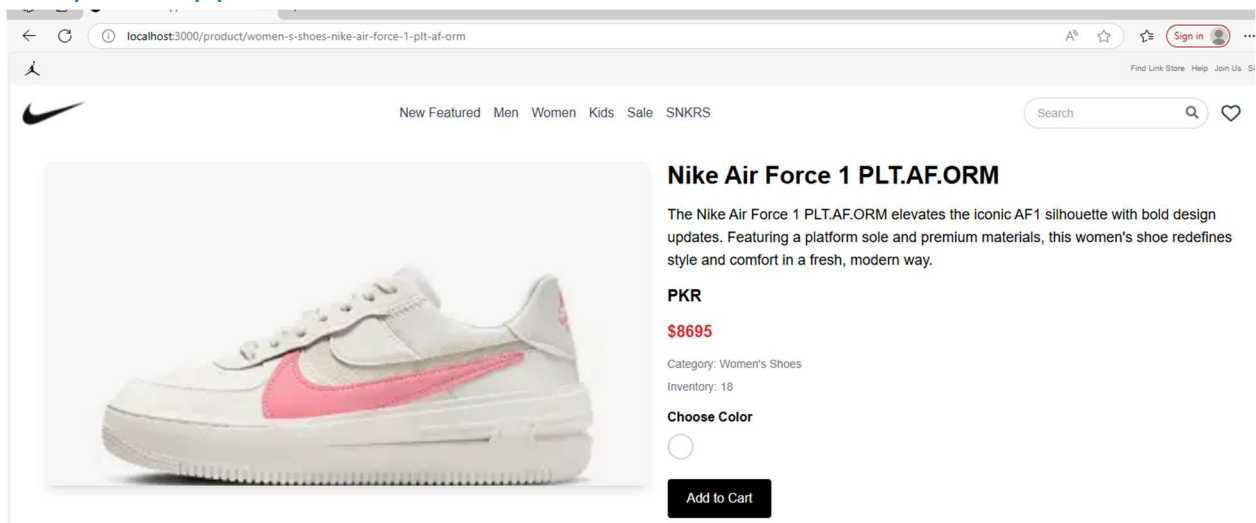
- File-based routing: Create dynamic routes using `[param].js` in the `pages` folder (e.g., `pages/[id].js`).
- `getStaticPaths`: Defines which dynamic routes need to be pre-rendered at build time.
- `getStaticProps`: Fetches data for each dynamic route.

### Example:

- `pages/[id].js` matches routes like `/product/1`, `/product/2`.

This enables flexible, dynamic content rendering based on URL parameters.

## Output Snippet:



## Laiba Jaweed

## Search Bar

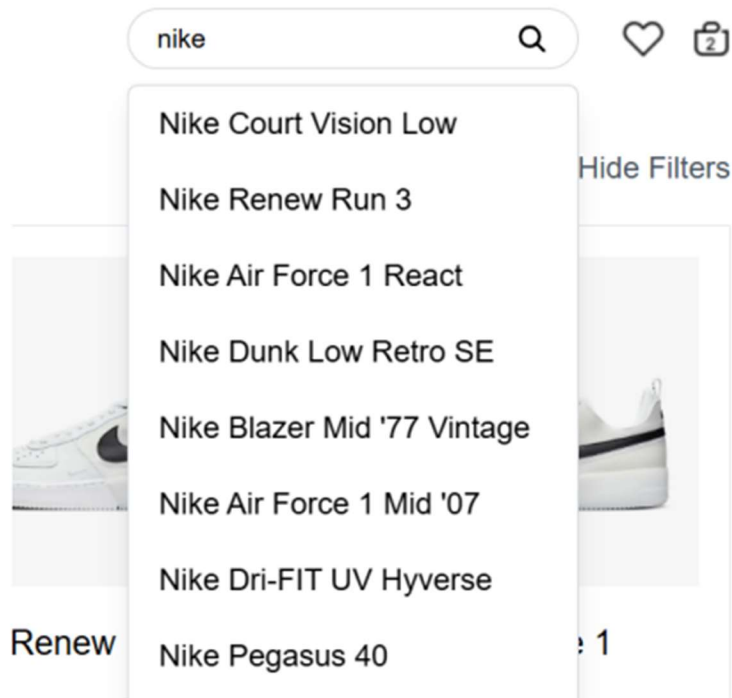
## Code Snippet:

```

src > components > Header.jsx > Header
9 export default function Header() {
10   const [isMenuOpen, setIsMenuOpen] = useState(false);
11   const [searchQuery, setSearchQuery] = useState('');
12   const [products, setProducts] = useState<any[]>([]);
13   const [filteredProducts, setFilteredProducts] = useState<any[]>([]);
14
15   // Fetch products from Sanity
16   useEffect(() => {
17     const fetchProducts = async () => {
18       const data = await client.fetch<*_type == "product">[
19         _id,
20         productName,
21         category,
22         price,
23         image,
24         description,
25       ];
26       setProducts(data);
27       setFilteredProducts(data); // Initialize with all products
28     };
29
30     fetchProducts();
31   }, []);
32
33   // Handle search query change
34   const handleSearch = (event: React.ChangeEvent<HTMLInputElement>) => {
35     const query = event.target.value.toLowerCase();
36     setSearchQuery(query);
37
38     // Filter products based on search query
39     const filtered = products.filter((product) =>
40       product.productName.toLowerCase().includes(query)
41     );
42     setFilteredProducts(filtered);
43   };

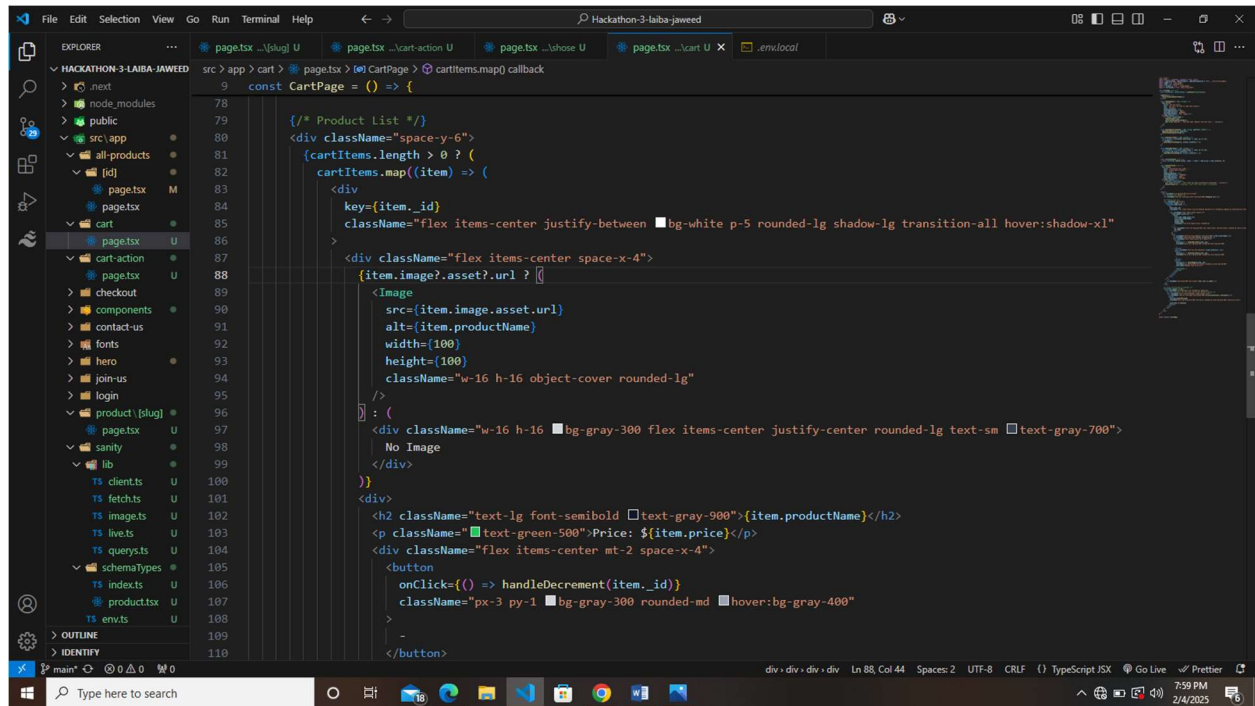
```

### Output Snippet:



## Cart Component

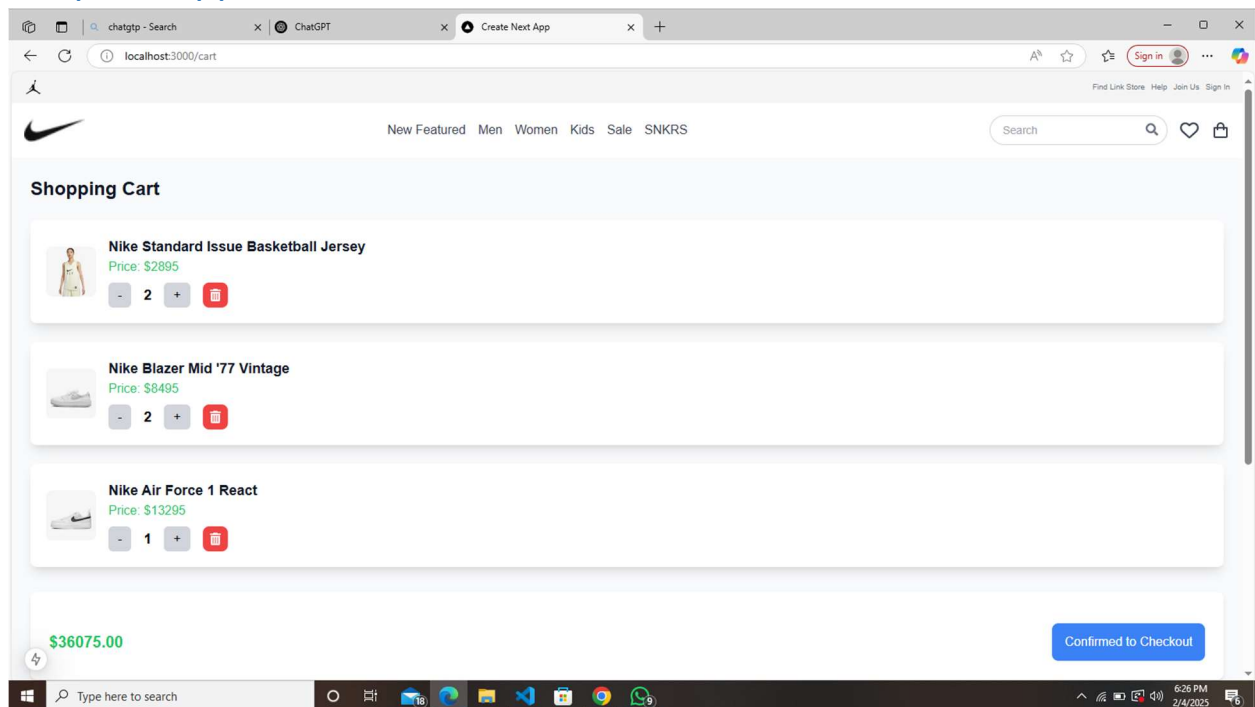
## Code Snippet:



```
9  const CartPage = () => {
78
79   /* Product List */
80   <div className="space-y-6">
81     {cartItems.length > 0 ? (
82       cartItems.map((item) => {
83         <div
84           key={item._id}
85           className="flex items-center justify-between bg-white p-5 rounded-lg shadow-lg transition-all hover:shadow-xl"
86         >
87           <div className="flex items-center space-x-4">
88             {item.image?.asset?.url ? (
89               <Image
90                 src={item.image.asset.url}
91                 alt={item.productName}
92                 width={100}
93                 height={100}
94                 className="w-16 h-16 object-cover rounded-lg"
95               />
96             ) : (
97               <div className="w-16 h-16 bg-gray-300 flex items-center justify-center rounded-lg text-sm text-gray-700">
98                 No Image
99               </div>
100             )}
101           <div>
102             <h2 className="text-lg font-semibold text-gray-900">{item.productName}</h2>
103             <p className="text-green-500">Price: ${item.price}</p>
104             <div className="flex items-center mt-2 space-x-4">
105               <button
106                 onClick={() => handleDecrement(item._id)}
107                 className="px-3 py-1 bg-gray-300 rounded-md hover:bg-gray-400"
108               >
109               -
110             </button>

```

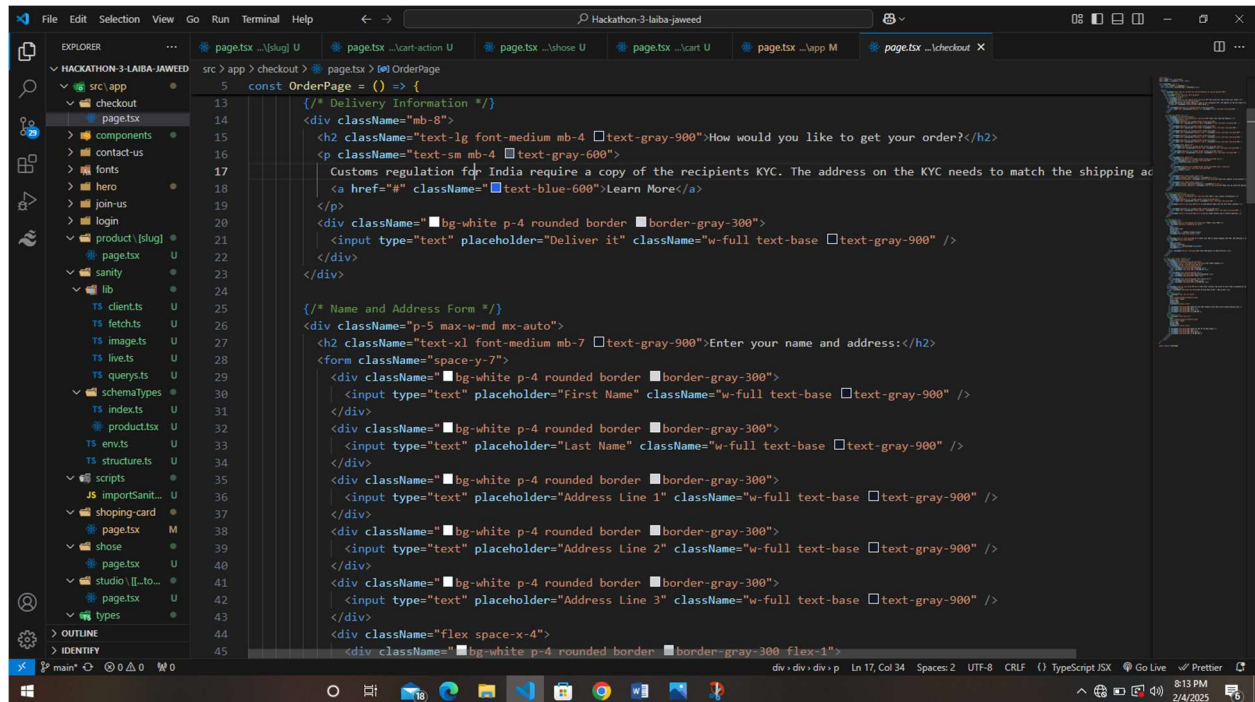
## Output Snippet:



## Checkout Component

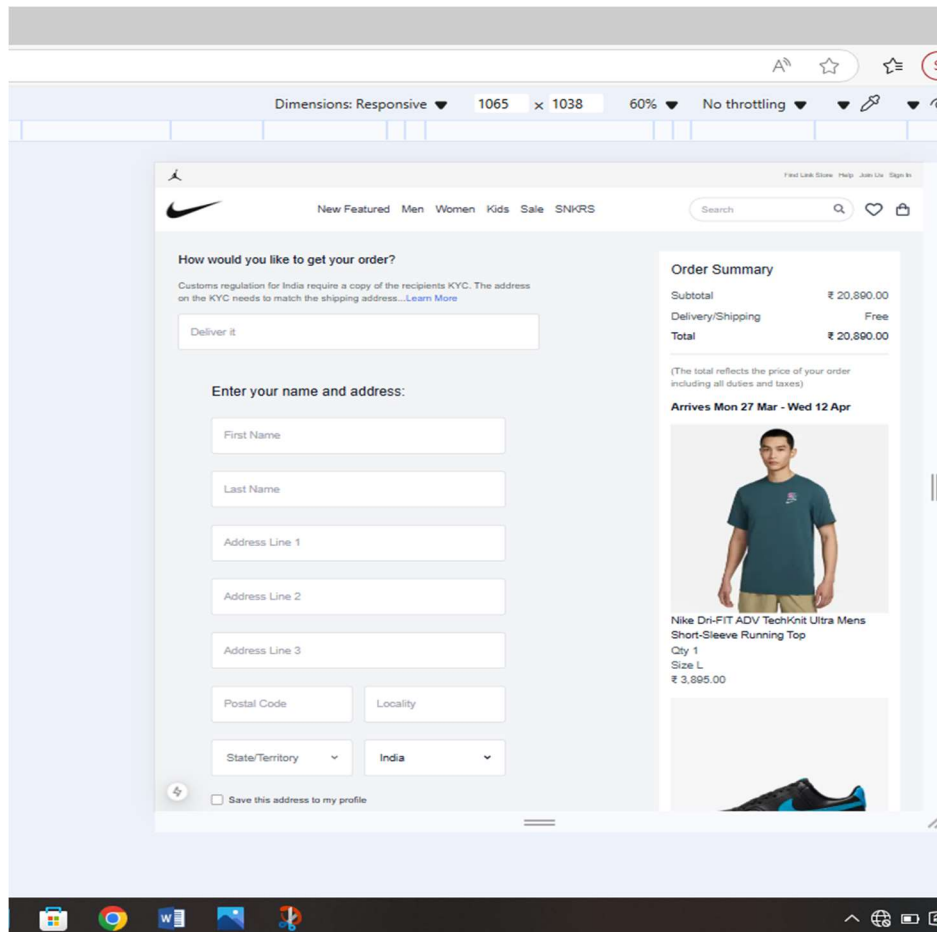


## Code Snippet:



```
5  const OrderPage = () => {
13    /* Delivery Information */
14    <div className="mb-8">
15      <h2 className="text-lg font-medium mb-4 text-gray-900">How would you like to get your order?</h2>
16      <p className="text-sm mb-4 text-gray-600">Customs regulation for India require a copy of the recipients KYC. The address on the KYC needs to match the shipping address</p>
17      <a href="#" className="text-blue-600">Learn More</a>
18    </div>
19
20    <div className="bg-white p-4 rounded border border-gray-300">
21      <input type="text" placeholder="Deliver it" className="w-full text-base text-gray-900" />
22    </div>
23
24    /* Name and Address Form */
25    <div className="p-5 max-w-md mx-auto">
26      <h2 className="text-xl font-medium mb-7 text-gray-900">Enter your name and address:</h2>
27      <form className="space-y-7">
28        <div className="bg-white p-4 rounded border border-gray-300">
29          <input type="text" placeholder="First Name" className="w-full text-base text-gray-900" />
30        </div>
31        <div className="bg-white p-4 rounded border border-gray-300">
32          <input type="text" placeholder="Last Name" className="w-full text-base text-gray-900" />
33        </div>
34        <div className="bg-white p-4 rounded border border-gray-300">
35          <input type="text" placeholder="Address Line 1" className="w-full text-base text-gray-900" />
36        </div>
37        <div className="bg-white p-4 rounded border border-gray-300">
38          <input type="text" placeholder="Address Line 2" className="w-full text-base text-gray-900" />
39        </div>
40        <div className="bg-white p-4 rounded border border-gray-300">
41          <input type="text" placeholder="Address Line 3" className="w-full text-base text-gray-900" />
42        </div>
43        <div className="flex space-x-4">
44          <div className="bg-white p-4 rounded border border-gray-300 flex-1">
45
```

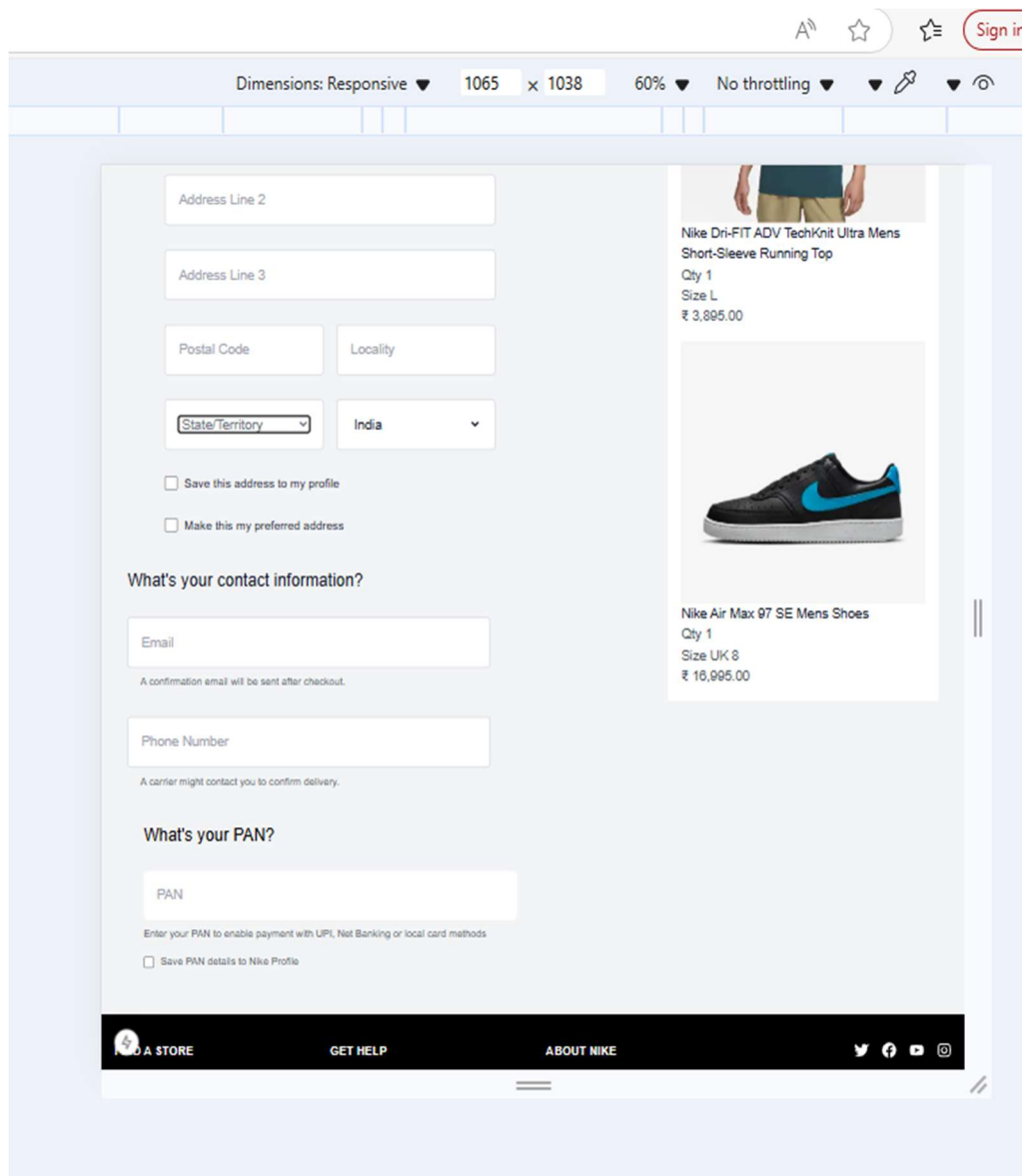
## Output Snippet



[Laiba Jaweed](#)

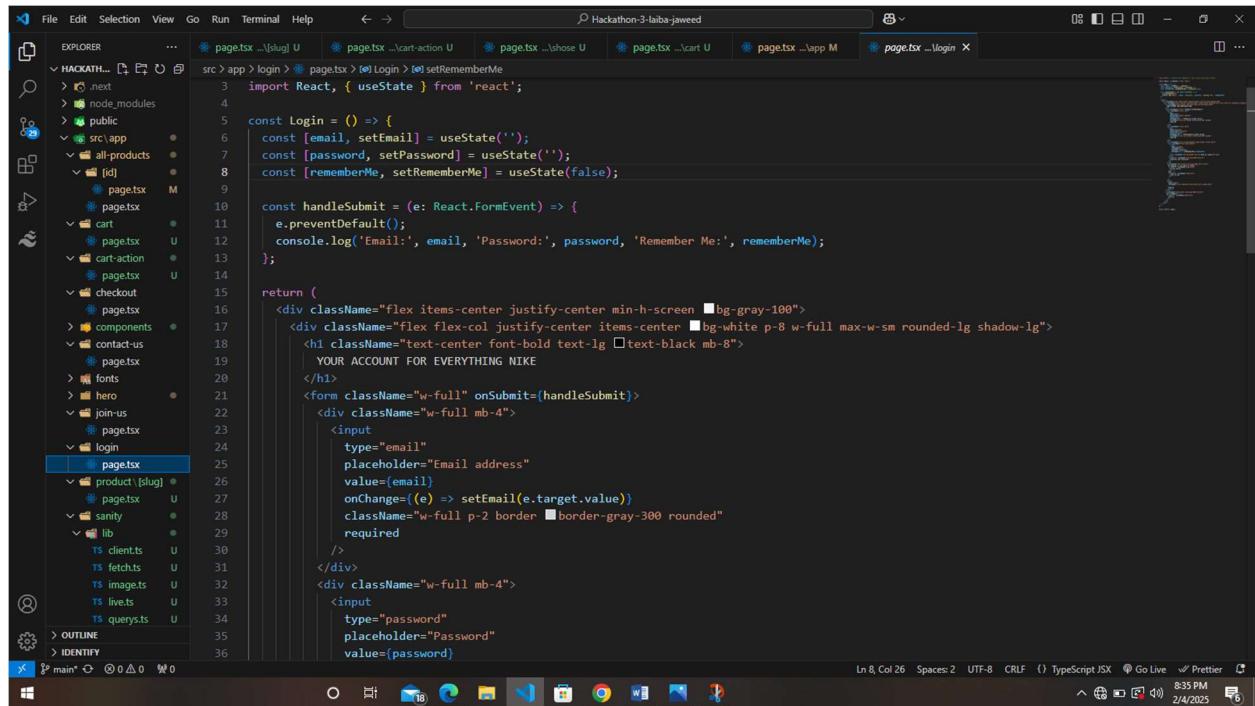
Output Snippet:





## Login Component

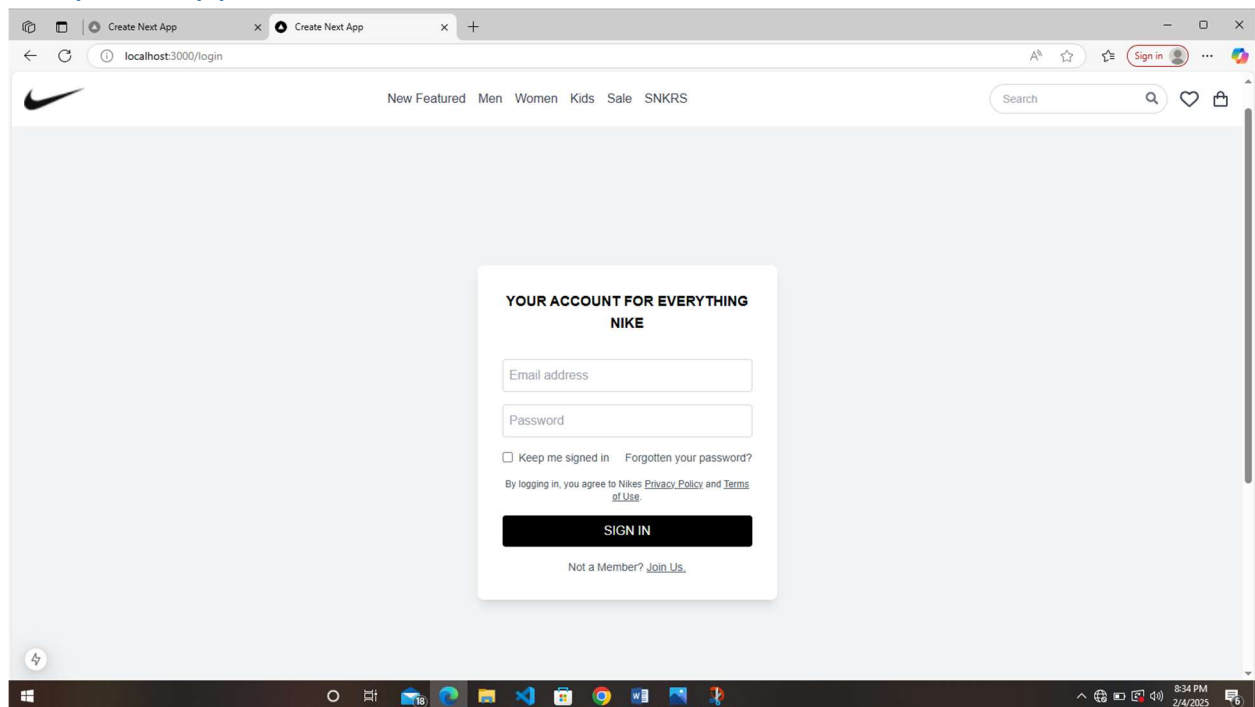
## Code Snippet:



The screenshot shows a VS Code editor window with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like 'src', 'public', 'components', and 'pages'. The code editor shows the 'login' page code, which is a React component using useState for email, password, and rememberMe. It includes a handleSubmit function and JSX for the login form.

```
src > app > login > page.jsx > setRememberMe
3 import React, { useState } from 'react';
4
5 const Login = () => {
6   const [email, setEmail] = useState('');
7   const [password, setPassword] = useState('');
8   const [rememberMe, setRememberMe] = useState(false);
9
10  const handleSubmit = (e: React.FormEvent) => {
11    e.preventDefault();
12    console.log('Email:', email, 'Password:', password, 'Remember Me:', rememberMe);
13  };
14
15  return (
16    <div className="flex items-center justify-center min-h-screen bg-gray-100">
17      <div className="flex flex-col justify-center items-center bg-white p-8 w-full max-w-sm rounded-lg shadow-lg">
18        <h1 className="text-center font-bold text-lg text-black mb-8">
19          YOUR ACCOUNT FOR EVERYTHING NIKE
20        </h1>
21        <form className="w-full" onSubmit={handleSubmit}>
22          <div className="w-full mb-4">
23            <input
24              type="email"
25              placeholder="Email address"
26              value={email}
27              onChange={(e) => setEmail(e.target.value)}
28              className="w-full p-2 border border-gray-300 rounded"
29              required
30            />
31          </div>
32          <div className="w-full mb-4">
33            <input
34              type="password"
35              placeholder="Password"
36              value={password}
```

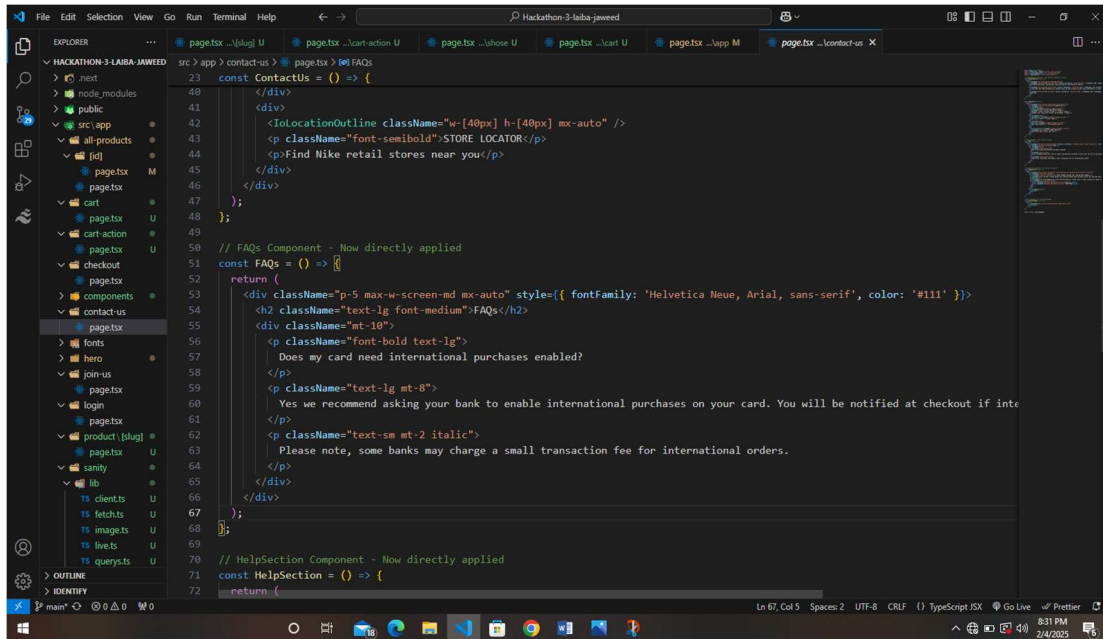
## Output Snippet:



Laiba Jaweed

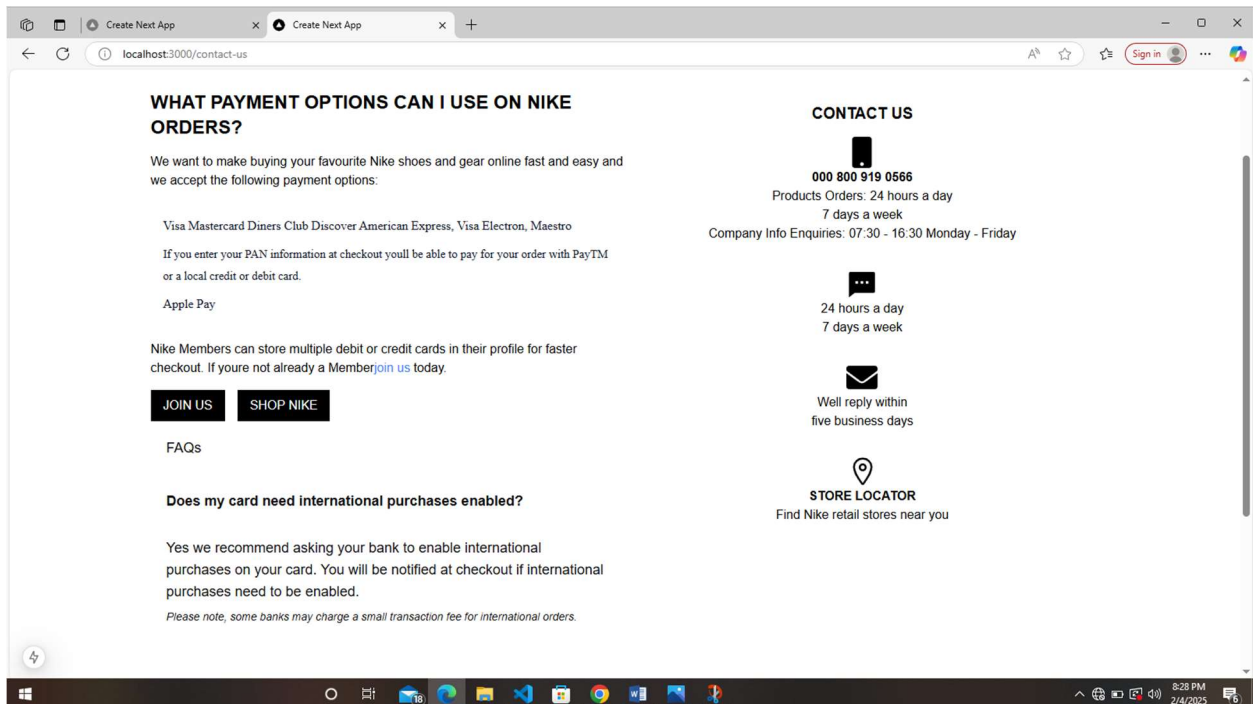
# Contact and FAQs Component

## Code Snippet:



```
23 const ContactUs = () => {
40   </div>
41   <div>
42     <ToLocationOutline className="w-[40px] h-[40px] mx-auto" />
43     <p className="font-semibold">STORE LOCATOR</p>
44     <p>Find Nike retail stores near you</p>
45   </div>
46 </div>
47 };
48
49 // FAQs Component - Now directly applied
50 const FAQs = () => {
51   return (
52     <div className="p-5 max-w-screen-md mx-auto" style={{ fontFamily: 'Helvetica Neue, Arial, sans-serif', color: '#111' }}>
53       <h2 className="text-lg font-medium">FAQs</h2>
54       <div className="mt-10">
55         <p className="font-bold text-lg">Does my card need international purchases enabled?</p>
56         <p></p>
57         <p className="text-lg mt-8">Yes we recommend asking your bank to enable international purchases on your card. You will be notified at checkout if international purchases need to be enabled.</p>
58         <p></p>
59         <p className="text-sm mt-2 italic">Please note, some banks may charge a small transaction fee for international orders.</p>
60       </div>
61     </div>
62   );
63 }
64
65 // HelpSection Component - Now directly applied
66 const HelpSection = () => {
67   return (
68     <div>
69       <h3>WHAT PAYMENT OPTIONS CAN I USE ON NIKE ORDERS?</h3>
70       <p>We want to make buying your favourite Nike shoes and gear online fast and easy and we accept the following payment options:</p>
71       <p>Visa Mastercard Diners Club Discover American Express, Visa Electron, Maestro</p>
72       <p>If you enter your PAN information at checkout you'll be able to pay for your order with PayTM or a local credit or debit card.</p>
73       <p>Apple Pay</p>
74       <p>Nike Members can store multiple debit or credit cards in their profile for faster checkout. If you're not already a Member<a href="#">join us today.</p>
75       <div>
76         <button>JOIN US</button>
77         <button>SHOP NIKE</button>
78       </div>
79       <h3>FAQs</h3>
80       <p></p>
81       <h4>Does my card need international purchases enabled?</h4>
82       <p>Yes we recommend asking your bank to enable international purchases on your card. You will be notified at checkout if international purchases need to be enabled.</p>
83       <p>Please note, some banks may charge a small transaction fee for international orders.</p>
84     </div>
85   );
86 }
```


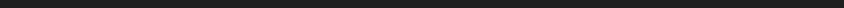
## Output Snippet:



## API Integration:

1. **Sanity API Fetching Logic:** Since you're using Sanity CMS, you need to ensure data fetching is properly managed.
2. **Sanity Client Setup (if not already set up):** In your lib folder, configure the Sanity client.

### Code Snippet:

```
.env.local  
1 NEXT_PUBLIC_SANITY_PROJECT_ID="  
2 NEXT_PUBLIC_SANITY_DATASET="production"  
3 SANITY_API_TOKEN=sk-  
4  
5
```

### Final CheckList

Frontend Component Development	Styling and Responsiveness	Code Quality	Documentation and Submission	Final Review
✓	✓	✓	✓	✓

Laiba Jaweed