MODULE 2: BUILDING THE PRODUCT STORE

LESSON 5: Dynamic Routes for Products

Learning Objective

By the end of this lesson, you will be able to create individual product pages with dynamic URLs like `/product/1`, `/product/2`, etc.

5.1 What are Dynamic Routes?

In Next.js, dynamic routes allow you to create pages based on data. Instead of creating a separate file for each product, you create one template that works for all products.

Example:

- `ceylonmart.com/product/1` → Shows product with ID 1
- `ceylonmart.com/product/2` → Shows product with ID 2

- `ceylonmart.com/product/3` → Shows product with ID 3

All using the same code!

5.2 Creating the Dynamic Route

Step 1: Create a new folder structure in `app`

```
text
app/
├── product/
|   └── [id]/
|       └── page.js
```

In Next.js, folders with square brackets `[id]` create dynamic routes.

Step 2: Create `app/product/[id]/page.js`

```javascript
"use client";

import { useParams } from 'next/navigation';
import Link from 'next/link';
import Header from '@/app/components/Header';
import Footer from '@/app/components/Footer';
import products from '@/app/data/products';

export default function ProductPage() {
  // Get the product ID from the URL
  const params = useParams();
  const productId = parseInt(params.id);

  // Find the product with this ID
```

```
const product = products.find(p => p.id === productId);

// If product not found, show 404 message
if (!product) {
  return (
    <div>
      <Header />
      <main style={{ padding: '4rem 2rem', textAlign: 'center' }}>
        <h1 style={{ fontSize: '2rem', color: '#dc2626' }}>Product Not Found</h1>
        <p style={{ margin: '1rem 0 2rem' }}>The product you're looking for doesn't exist.</p>
        <Link href="/" style={{
          backgroundColor: '#1e40af',
          color: 'white',
          padding: '0.75rem 1.5rem',
          textDecoration: 'none',
          borderRadius: '4px'
        }}>
          Back to Home
        </Link>
      </main>
      <Footer />
    </div>
  );
}

return (
  <div>
    <Header />

    <main style={{ padding: '2rem' }}>
      {/* Breadcrumb navigation */}
      <div style={{ marginBottom: '2rem', color: '#64748b' }}>
        <Link href="/" style={{ color: '#1e40af', textDecoration: 'none' }}>Home</Link>
        <span style={{ margin: '0 0.5rem' }}>/</span>
        <Link href="/" style={{ color: '#1e40af', textDecoration: 'none' }}>Products</Link>
        <span style={{ margin: '0 0.5rem' }}>/</span>
        <span>{product.name}</span>
```

```
</div>

{/* Product Details Grid */}
<div style={{
  display: 'grid',
  gridTemplateColumns: '1fr 1fr',
  gap: '3rem',
  maxWidth: '1200px',
  margin: '0 auto'
}}>
  {/* Left Column - Image */}
  <div>
    <div style={{
      width: '100%',
      height: '400px',
      backgroundColor: '#f1f5f9',
      borderRadius: '8px',
      display: 'flex',
      alignItems: 'center',
      justifyContent: 'center',
      fontSize: '1.2rem',
      color: '#64748b'
    }}>
      🖼️ {product.name} Image
    </div>
  </div>

  {/* Right Column - Details */}
  <div>
    <h1 style={{ fontSize: '2.5rem', margin: '0 0 1rem 0' }}>
      {product.name}
    </h1>

    <p style={{ fontSize: '1.1rem', color: '#475569', lineHeight: '1.6', marginBottom: '2rem' }}>
      {product.description}
    </p>
```

```
          <div style={{ marginBottom: '2rem' }}>
            <span style={{ fontSize: '2rem', fontWeight: 'bold', color: '#1e40af' }}>
              Rs. {product.price.toLocaleString()}
            </span>
            <span style={{ marginLeft: '1rem', color: '#059669', backgroundColor: '#d1fae5',
padding: '0.25rem 0.75rem', borderRadius: '9999px', fontSize: '0.875rem' }}>
              In Stock
            </span>
          </div>

          <div style={{ marginBottom: '2rem' }}>
            <label style={{ display: 'block', marginBottom: '0.5rem', fontWeight: 'bold' }}>
              Quantity:
            </label>
            <select style={{
              padding: '0.5rem',
              border: '1px solid #cbd5e1',
              borderRadius: '4px',
              width: '100px'
            }}>
              {[1,2,3,4,5].map(num => (
                <option key={num}>{num}</option>
              ))}
            </select>
          </div>

          <div style={{ display: 'flex', gap: '1rem' }}>
            <button style={{
              flex: 2,
              backgroundColor: '#1e40af',
              color: 'white',
              border: 'none',
              padding: '1rem',
              borderRadius: '4px',
              fontSize: '1.1rem',
              fontWeight: 'bold',
              cursor: 'pointer'
```

```
  }}>
    Add to Cart
  </button>

  <button style={{
    flex: 1,
    backgroundColor: 'white',
    color: '#1e40af',
    border: '2px solid #1e40af',
    padding: '1rem',
    borderRadius: '4px',
    fontSize: '1.1rem',
    cursor: 'pointer'
  }}>
    Buy Now
  </button>
</div>

{/* Product Details Table */}
<div style={{ marginTop: '3rem', borderTop: '1px solid #e2e8f0', paddingTop: '2rem' }}>
  <h3 style={{ marginBottom: '1rem' }}>Product Details</h3>
  <table style={{ width: '100%', borderCollapse: 'collapse' }}>
    <tbody>
      <tr style={{ borderBottom: '1px solid #e2e8f0' }}>
        <td style={{ padding: '0.75rem 0', fontWeight: 'bold' }}>Category</td>
        <td style={{ padding: '0.75rem 0' }}>{product.category}</td>
      </tr>
      <tr style={{ borderBottom: '1px solid #e2e8f0' }}>
        <td style={{ padding: '0.75rem 0', fontWeight: 'bold' }}>Origin</td>
        <td style={{ padding: '0.75rem 0' }}>Sri Lanka</td>
      </tr>
      <tr style={{ borderBottom: '1px solid #e2e8f0' }}>
        <td style={{ padding: '0.75rem 0', fontWeight: 'bold' }}>Shipping</td>
        <td style={{ padding: '0.75rem 0' }}>Free within Colombo, Rs. 250 nationwide</td>
      </tr>
    </tbody>
  </table>
```

```
        </div>
      </div>
    </div>
  </main>

  <Footer />
 </div>
 );
}
```

5.3 Making Product Cards Clickable

Now let's update our ProductCard component to link to these detail pages.

Step 1: Update `app/components/ProductCard.js`

javascript

```
import Link from 'next/link';

function ProductCard({ product }) {
 return (
  <Link href={`/product/${product.id}`} style={{ textDecoration: 'none', color: 'inherit' }}>
   <div style={{
    border: '1px solid #e2e8f0',
    borderRadius: '8px',
    padding: '1rem',
    backgroundColor: 'white',
    boxShadow: '0 2px 4px rgba(0,0,0,0.1)',
    transition: 'transform 0.2s, box-shadow 0.2s',
    cursor: 'pointer',
    height: '100%',
    display: 'flex',
    flexDirection: 'column'
```

```
}}
onMouseEnter={e => {
 e.currentTarget.style.transform = 'scale(1.02)';
 e.currentTarget.style.boxShadow = '0 10px 15px -3px rgba(0,0,0,0.1)';
}}
onMouseLeave={e => {
 e.currentTarget.style.transform = 'scale(1)';
 e.currentTarget.style.boxShadow = '0 2px 4px rgba(0,0,0,0.1)';
}}
>
 {/* Product Image Placeholder */}
 <div style={{
  width: '100%',
  height: '200px',
  backgroundColor: '#f1f5f9',
  borderRadius: '4px',
  marginBottom: '1rem',
  display: 'flex',
  alignItems: 'center',
  justifyContent: 'center',
  color: '#64748b'
 }}>
  🖼️ {product.name}
 </div>

 {/* Product Details */}
 <h3 style={{ margin: '0 0 0.5rem 0', fontSize: '1.2rem' }}>
  {product.name}
 </h3>

 <p style={{ color: '#64748b', margin: '0 0 1rem 0', fontSize: '0.9rem', flex: 1 }}>
  {product.description.substring(0, 60)}...
 </p>

 <div style={{
  display: 'flex',
  justifyContent: 'space-between',
```

```
          alignItems: 'center',
          marginTop: 'auto'
        }}>
          <span style={{
            fontSize: '1.3rem',
            fontWeight: 'bold',
            color: '#1e40af'
          }}>
            Rs. {product.price.toLocaleString()}
          </span>

          <button
            onClick={(e) => {
              e.preventDefault(); // Prevent navigation when clicking button
              console.log('Added to cart:', product.name);
            }}
            style={{
              backgroundColor: '#1e40af',
              color: 'white',
              border: 'none',
              padding: '0.5rem 1rem',
              borderRadius: '4px',
              cursor: 'pointer',
              fontSize: '0.9rem'
            }}
          >
            Add to Cart
          </button>
        </div>
      </div>
    </Link>
  );
}

export default ProductCard;
```

Important: Notice the `onClick` on the button has `e.preventDefault()`. This stops the click from also triggering the link navigation.

5.4 Understanding the Link Component

Next.js provides a `Link` component for client-side navigation. Unlike regular `<a>` tags, `Link` doesn't reload the page - it's faster and smoother.

```javascript
// Regular HTML - full page reload
<a href="/product/1">View Product</a>

// Next.js Link - no page reload
<Link href="/product/1">View Product</Link>
```

Link Props:

- `href`: Where to navigate to
- `replace`: Replace current history instead of adding new entry
- `scroll`: Scroll to top after navigation (default true)

5.5 Adding Related Products

Let's enhance the product page by showing related products from the same category.

Step 1: Update `app/product/[id]/page.js` - add this after the main product section

Add this import at the top:

```javascript
import ProductCard from '@/app/components/ProductCard';
```

Add this section inside the main tag, after the product details grid:

```javascript
{/* Related Products */}
<div style={{ marginTop: '4rem' }}>
  <h2 style={{ fontSize: '1.8rem', marginBottom: '2rem' }}>
    You Might Also Like
  </h2>

  <div style={{
    display: 'grid',
    gridTemplateColumns: 'repeat(auto-fill, minmax(250px, 1fr))',
    gap: '2rem'
  }}>
    {products
      .filter(p => p.category === product.category && p.id !== product.id)
      .slice(0, 4)
      .map(relatedProduct => (
        <ProductCard key={relatedProduct.id} product={relatedProduct} />
      ))
    }
  </div>

</div>
```

Lesson 5 Summary

- Dynamic routes use folder names with `[id]` syntax

- `useParams()` gets URL parameters
- `Link` component provides fast client-side navigation
- Always handle the "not found" case
- Related products improve user experience

Practice Exercises

1. Add previous/next navigation buttons to move between products
2. Create a reviews section on the product page
3. Add a "Share" button that copies the product URL
4. Challenge: Add a zoom feature for product images (hint: use CSS transform)