Route Parameters

INTERVIEW QUESTIONS-88

Follow on

@DUVVURU KISHORE

Interview question:

how do you add dynamic parameters to the url?

What are Route Parameters?

- Route parameters are placeholders in a URL pattern that allow you to create dynamic URLs.
- Instead of having separate URLs for each page or resource in your web application, you can use route parameters to handle different scenarios or user inputs without hardcoding every possible URL.

Why do we need Route Parameters?

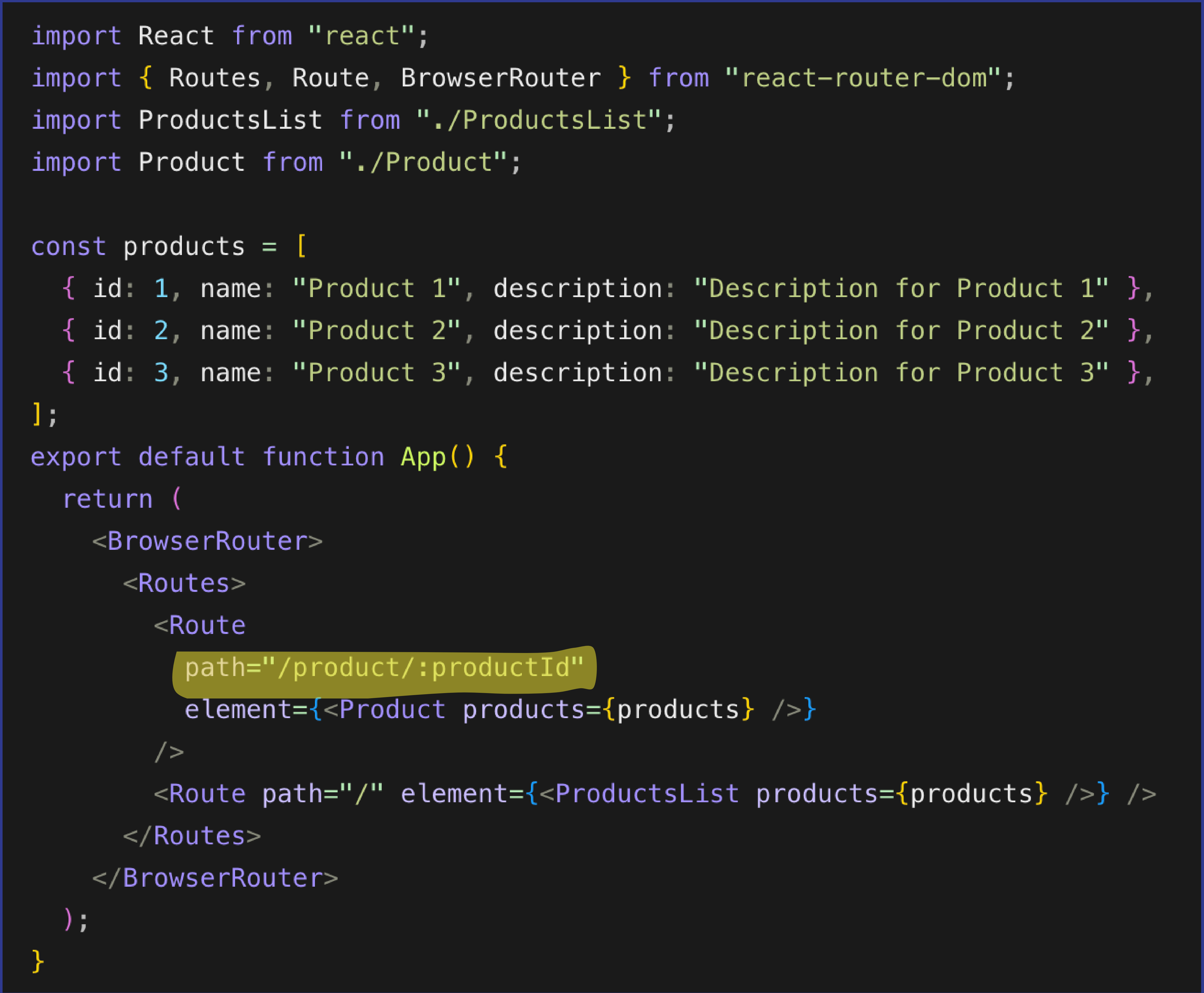
Dynamic Content: They help in displaying dynamic content based on user input or other factors. For example, in an e-commerce website, you can use route parameters to show different product details when users click on different product links.

Cleaner Code: Route parameters make your code cleaner and more maintainable. You don't have to write separate code for each specific URL. Instead, you define a pattern with placeholders, making your code more flexible and easier to manage.

SEO Friendly URLs: They also help in creating more search engine-friendly URLs. Instead of using complex query strings or hash fragments, route parameters allow you to create cleaner and more descriptive URLs, which can improve your website's search engine optimization.

How do Route Parameters Work?

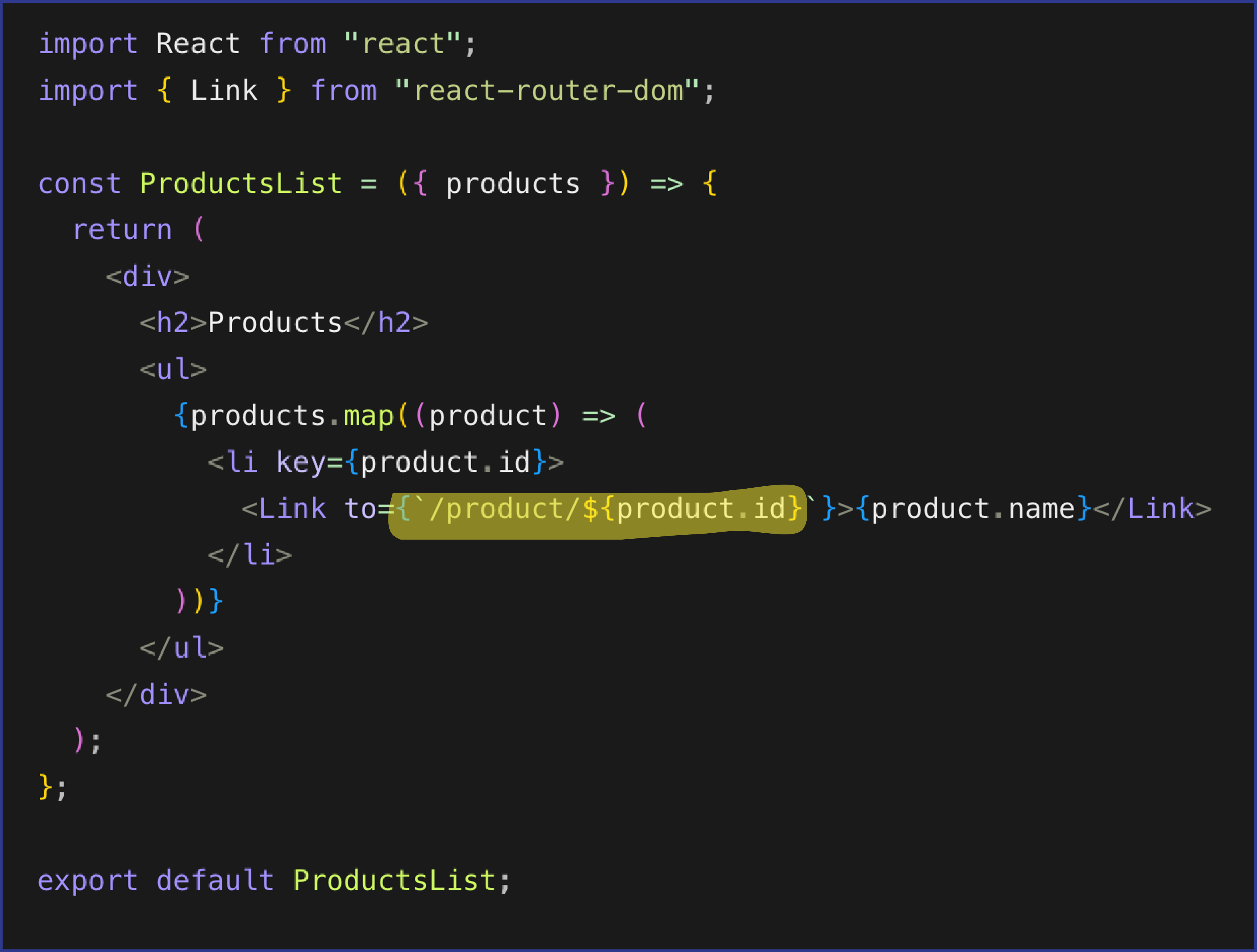
We define a route /product/:productId, where :productId is the route parameter.



```
import React from "react";
import { Routes, Route, BrowserRouter } from "react-router-dom";
import ProductsList from "./ProductsList";
import Product from "./Product";

const products = [
  { id: 1, name: "Product 1", description: "Description for Product 1" },
  { id: 2, name: "Product 2", description: "Description for Product 2" },
  { id: 3, name: "Product 3", description: "Description for Product 3" },
];
export default function App() {
  return (
    <BrowserRouter>
      <Routes>
        <Route
          path="/product/:productId"
          element={<Product products={products} />}
        />
        <Route path="/" element={<ProductsList products={products} />} />
      </Routes>
    </BrowserRouter>
  );
}
```

for home page path ('/') has all the products list and products list is loaded from below component

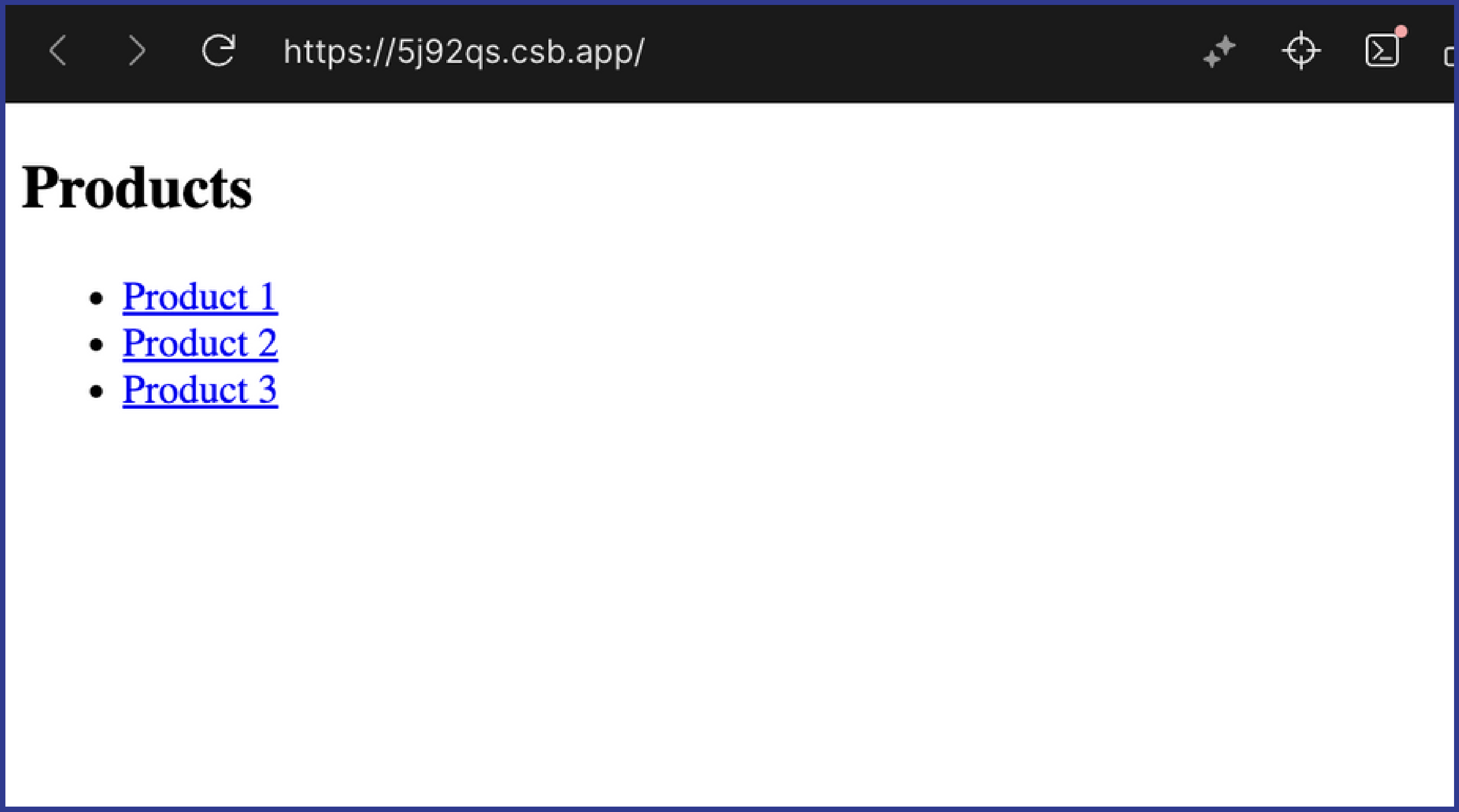


```
import React from "react";
import { Link } from "react-router-dom";

const ProductsList = ({ products }) => {
  return (
    <div>
      <h2>Products</h2>
      <ul>
        {products.map((product) => (
          <li key={product.id}>
            <Link to={`/product/${product.id}`}>{product.name}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default ProductsList;
```

Now UI looks like below where home page is displaying the products.



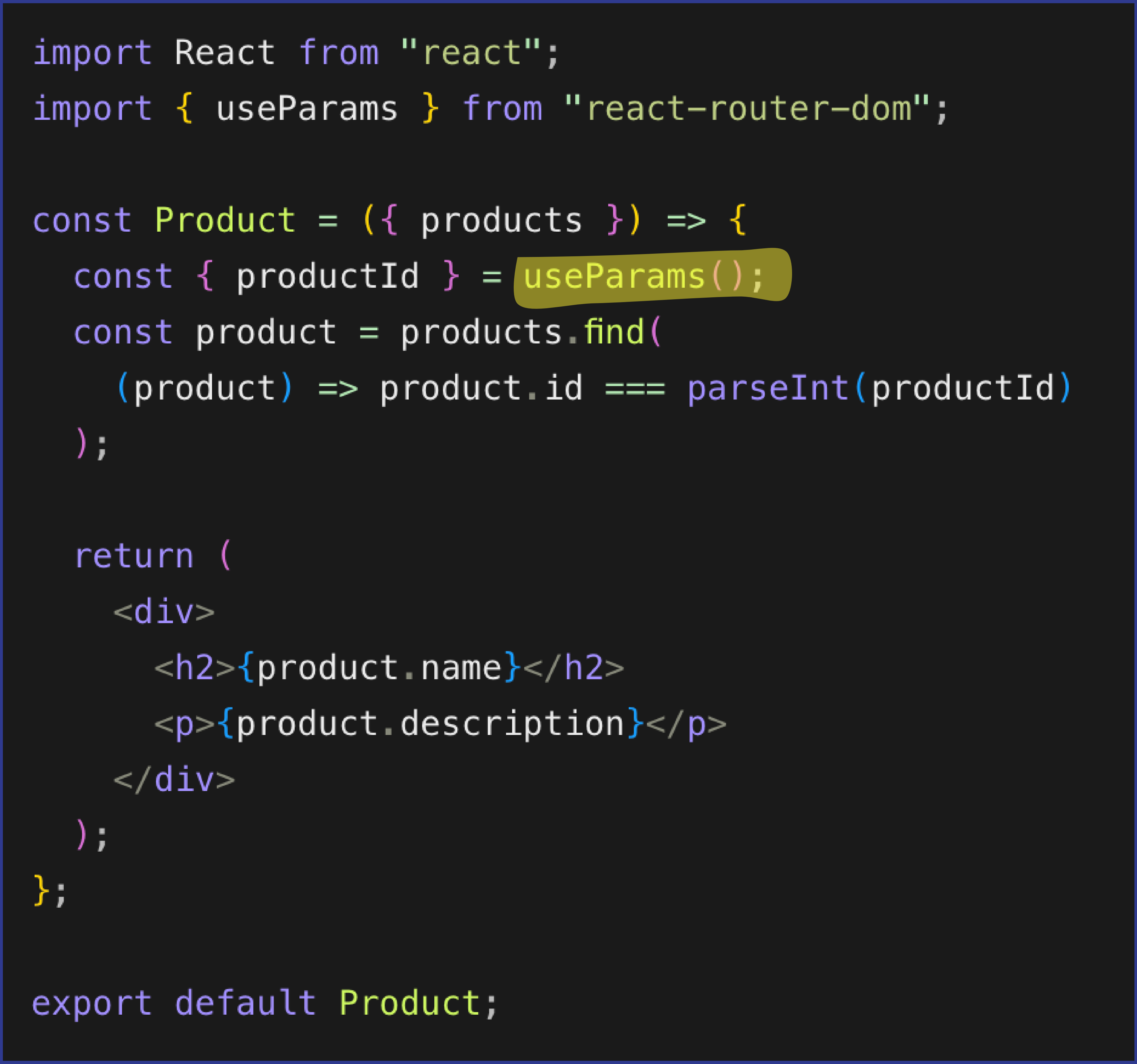
https://5j92qs.csb.app/

Products

- Product 1
- Product 2
- Product 3

Now when we click on each product ,it should take to the url /product/${product.id}

When the URL matches this pattern (e.g., /product/1), React Router extracts the value of :productId from the URL and passes it to the Product component via useParams() hook.



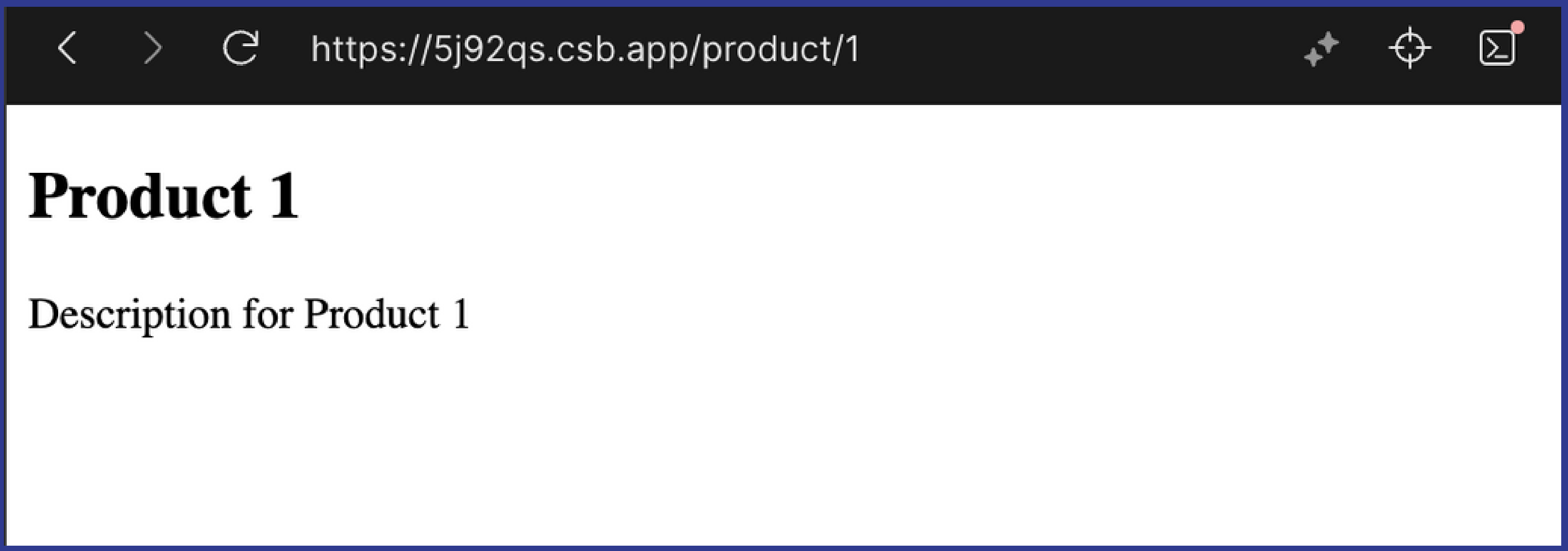
```
import React from "react";
import { useParams } from "react-router-dom";

const Product = ({ products }) => {
  const { productId } = useParams();
  const product = products.find(
    (product) => product.id === parseInt(productId)
  );

  return (
    <div>
      <h2>{product.name}</h2>
      <p>{product.description}</p>
    </div>
  );
};

export default Product;
```

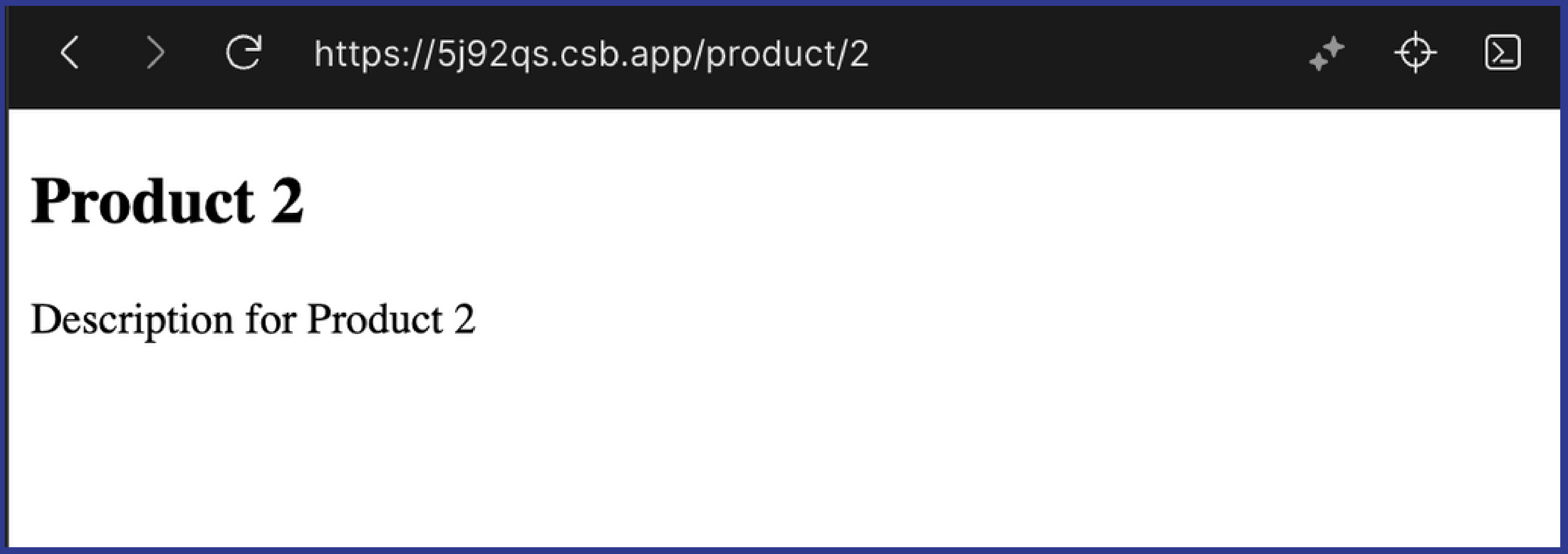
The **Product** component can then use this **productId** to fetch and display the details of the corresponding product.



https://5j92qs.csb.app/product/1

Product 1

Description for Product 1



https://5j92qs.csb.app/product/2

Product 2

Description for Product 2

Overall, route parameters are a powerful feature of React Router that enable you to build dynamic and interactive web applications with complex routing requirements. By leveraging route parameters effectively, you can create intuitive and user-friendly navigation experiences.

Follow on in

@Duvvuru Kishore