🔁 Routing Structure Overview

You are defining a **nested route structure** using `createBrowserRouter`. The key parts of the routing tree are:

```bash
/
├── RootLayout (common layout)
│   ├── /                    → HomePage
│   ├── /products            → ProductsPage
│   └── /products/:productId → ProductDetailPage
```

🧱 Components and Their Roles

1. `RootLayout` (Layout Component)

```jsx
function RootLayout() { return ( <> <MainNavigation /> <main> <Outlet /> </main> </> ); }
```

- **Purpose**: Serves as a layout wrapper for all child routes.
- **Contains**:
  - `MainNavigation`: Always shown at the top.
  - `<Outlet />`: Placeholder where nested child routes render.

So, any route nested inside `/` will appear **inside** this layout, below the navigation bar.

---

2. Route Configuration in `App.js`

```js
const router = createBrowserRouter([ { path: "/", element: <RootLayout />, // Wraps all children errorElement: <ErrorPage />, // Shown if route fails children: [ { path: "/", element: <HomePage /> }, // Matches root "/" { path: "/products", element: <ProductsPage /> }, { path: "/products/:productId", element: <ProductDetailPage /> }, ], }, ]);
```

This defines:

- **Root Path (`/`):**
  - Uses `<RootLayout />` as the wrapper.
  - Renders `<HomePage />` when URL is exactly `/`.
- `/products`:
  - Shows `<ProductsPage />` inside `RootLayout`.
- `/products/:productId`:
  - Shows `<ProductDetailPage />` based on the dynamic `productId`.

🧩 Dynamic Routing

```js
{ path: "/products/:productId", element: <ProductDetailPage /> }
```

- This is a **parameterized route.**
- You can access `productId` inside `ProductDetailPage` using:

  ```js
  import { useParams } from 'react-router-dom'; const { productId } = useParams();
  ```

⚠️ Error Handling

```js
errorElement: <ErrorPage />
```

- If any error occurs while rendering a route (e.g., 404 or fetch failure), `ErrorPage` will be shown.

🧭 Rendering the Router

Finally, you wrap everything in:

```jsx
<RouterProvider router={router} />
```

This enables React Router to handle navigation throughout the app.

🖼️ Page Flow

If you open:

- `/` → Renders `MainNavigation` + `HomePage`
- `/products` → `MainNavigation` + `ProductsPage`
- `/products/123` → `MainNavigation` + `ProductDetailPage` (for product with ID 123)
- Any invalid route → `ErrorPage`