

Build a multi-page application using React Router (e.g., a simple E-commerce site).

### Step 1: Setup the Project

**1. Install Node.js and npm:**

- Download and install Node.js from [here](#).
- npm (Node Package Manager) comes bundled with Node.js.

**2. Create a New React Project:**

- Open your terminal/command prompt.
- Run the following command to create a new React app (replace `ecommerce-site` with your desired project name):

```
npx create-react-app ecommerce-site
```

**3. Navigate to the Project Directory:**

```
cd ecommerce-site
```

**4. Install React Router:**

```
npm install react-router-dom
```

**5. Start the Development Server:**

```
npm start
```

- This will start the development server and open your new React app in the browser.

### Step 2: Create Basic Pages

**1. Open the Project in Your Code Editor:**

- Use your preferred code editor (e.g., VSCode) to open the `ecommerce-site` directory.

**2. Create Basic Pages (Home, Products, Product Details, Cart):**

- Inside the `src` folder, create a new folder called `pages`.
- Inside the `pages` folder, create the following files: `Home.js`, `Products.js`, `ProductDetails.js`, and `Cart.js`.

**3. Edit `Home.js` to Add the Home Page Component:**

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

function Home() {
  return (
    <div>
      <h1>Welcome to Our E-commerce Site</h1>
      <Link to="/products">View Products</Link>
    </div>
  );
}

export default Home;
```

#### 4. Edit `Products.js` to Add the Products Page Component:

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

const products = [
  { id: 1, name: 'Product 1' },
  { id: 2, name: 'Product 2' },
  { id: 3, name: 'Product 3' },
];

function Products() {
  return (
    <div>
      <h1>Products</h1>
      <ul>
        {products.map((product) => (
          <li key={product.id}>
            <Link
to={`/${products}/${product.id}`}>{product.name}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
}

export default Products;
```

#### 5. Edit `ProductDetails.js` to Add the Product Details Page Component:

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

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' },
];

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

  if (!product) {
    return <div>Product not found</div>;
  }

  return (
    <div>
      <h1>{product.name}</h1>
      <p>{product.description}</p>
    </div>
  );
}
```

```
export default ProductDetails;
```

## 6. Edit `Cart.js` to Add the Cart Page Component:

```
import React from 'react';

function Cart() {
  return (
    <div>
      <h1>Cart</h1>
      <p>Your cart is empty.</p>
    </div>
  );
}

export default Cart;
```

### Step 3: Setup React Router

## 1. Edit `App.js` to Include React Router and Define Routes:

```
import React from 'react';
import { BrowserRouter as Router, Route, Routes, Link } from 'react-router-dom';
import Home from './pages/Home';
import Products from './pages/Products';
import ProductDetails from './pages/ProductDetails';
import Cart from './pages/Cart';
import './App.css';

function App() {
  return (
    <Router>
      <div className="App">
        <nav>
          <Link to="/">Home</Link>
          <Link to="/products">Products</Link>
          <Link to="/cart">Cart</Link>
        </nav>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/products" element={<Products />} />
          <Route path="/products/:productId" element={<ProductDetails
/>} />
          <Route path="/cart" element={<Cart />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

## 2. Add Basic Styling (Optional):

- Open `src/App.css` and add some basic styling:

```
.App {
```

```
    text-align: center;
  }

  nav {
    display: flex;
    justify-content: center;
    gap: 20px;
    margin: 20px 0;
  }

  a {
    text-decoration: none;
    color: #61dafb;
  }

  a:hover {
    text-decoration: underline;
  }
}
```

#### Step 4: Run Your App

##### 1. Start the Development Server Again (if it's not running):

```
npm start
```

##### 2. Open Your Browser:

- Navigate to `http://localhost:3000/` to see your E-commerce site in action.

#### Step 5: Additional Enhancements (Optional)

##### 1. Add Functionality to Add Products to the Cart:

- Implement a global state management solution like Context API or Redux to handle the cart state.

##### 2. Add More Pages:

- Create additional pages like a Checkout page, User Profile page, etc.

##### 3. Add More Styling:

- Use CSS frameworks like Bootstrap or Material-UI to enhance the look and feel of the site.

By following these steps, you should have a basic multi-page E-commerce site using React Router.