React + Vite Course (Kids 13–14)

Module 8 — React Router & Navigation (80 minutes)

> **Audience:** Kids 13–14
> This module builds directly on Modules 1–7.
> All code stays in the **same project**.
> Today's work goes into `src/day8/`.

Why This Module Is SUPER Important

Until now, students built **single-screen apps**.

Real websites have:

- Pages (Home, About, Profile, Contact)
- URLs (`/`, `/about`, `/profile`)
- Navigation without page reloads

This module introduces **React Router**, which lets React apps behave like **real websites**.

> Teaching line: *"React Router gives your app pages without reloading."*

Learning Objectives

By the end of this module, students will:

1. Understand what **routing** means
2. Know the difference between **normal links and React routing**
3. Install and use **React Router**
4. Create multiple pages in a React app
5. Navigate using `Link`
6. Understand **URLs & paths**
7. Use **Route parameters** (simple version)
8. Build a small multi-page app
9. See full homework solutions

80-Minute Agenda

- **0–10 min** — Recap (Context API)

- **10–20 min** — What is routing? (concepts)
- **20–30 min** — Installing React Router
- **30–50 min** — Creating pages & routes
- **50–65 min** — Navigation with Link
- **65–75 min** — Route parameters (intro)
- **75–80 min** — Homework + recap

Recap from Module 7 (0–10 min)

Ask students:

- What problem does Context solve?
- What is prop drilling?
- What is global state?

> **Reminder:** Context shares data, Router shares pages.

What Is Routing? (10–20 min)

Simple Explanation

Routing means:

> Showing **different components** based on the **URL**.

Examples:

- `/` → Home page
- `/about` → About page
- `/profile` → Profile page

Real-Life Analogy (VERY IMPORTANT)

Think of a **school building**:

- Room 1 → Math class
- Room 2 → English class
- Room 3 → Computer class

The **room number (URL)** decides what happens inside.

Normal Website vs React App

Normal Website

- Clicking a link reloads the page
- Browser asks server again

React App

- Page does **not reload**
- Only components change
- Faster and smoother

Teaching line: *"React changes screens, not websites."*

Installing React Router (20–30 min)

In terminal:

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

Explain slowly:

- This adds routing power to our app
- We install **once per project**

Setting Up the Router (30–40 min)

Update `main.jsx`:

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter } from 'react-router-dom';
import App from './App';

ReactDOM.createRoot(document.getElementById('root')).render(
  <BrowserRouter>
    <App />
  </BrowserRouter>
);
```

Explanation

- `BrowserRouter` watches the URL
- Everything inside can use routing

Creating Pages (40–50 min)

Create folder:

```
src/day8/pages/
```

Home.jsx

```jsx
function Home() {
  return <h2>Home Page</h2>;
}

export default Home;
```

About.jsx

```jsx
function About() {
  return <h2>About Page</h2>;
}

export default About;
```

Profile.jsx

```jsx
function Profile() {
  return <h2>Profile Page</h2>;
}

export default Profile;
```

Defining Routes (50–60 min)

Update `App.jsx`:

```jsx
import { Routes, Route } from 'react-router-dom';
import Home from './day8/pages/Home';
import About from './day8/pages/About';
import Profile from './day8/pages/Profile';

function App() {
  return (
    <Routes>
      <Route path="/" element={<Home />} />
      <Route path="/about" element={<About />} />
      <Route path="/profile" element={<Profile />} />
    </Routes>
  );
}

export default App;
```

Explanation (Very Important)

- `path` = URL
- `element` = component
- Only one route shows at a time

Navigation with Link (60–65 min)

Create `src/day8/NavBar.jsx`:

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

function NavBar() {
  return (
    <nav>
      <Link to="/">Home</Link> |
      <Link to="/about">About</Link> |
      <Link to="/profile">Profile</Link>
    </nav>
  );
}

export default NavBar;
```

Use it in `App.jsx` **above Routes**.

Teaching Point

- `Link` replaces `<a>`
- Prevents page reload

Route Parameters (65–75 min)

Why Route Parameters?

Dynamic pages like:

- `/user/1`
- `/user/2`

Example

Add route:

```
<Route path="/user/:id" element={<User />} />
```

User.jsx:

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

function User() {
  const { id } = useParams();

  return <h2>User ID: {id}</h2>;
}

export default User;
```

> Teaching line: *"The URL can send data too."*

Homework (Module 8)

Task 1

Create pages:

- Home
- About
- Contact

Task 2

Add navigation using `Link`

Task 3 (Bonus)

Create `/product/:id` page

Homework — FULL SOLUTIONS

Contact.jsx

```jsx
function Contact() {
  return <h2>Contact Page</h2>;
}

export default Contact;
```

Routes

```jsx
<Route path="/contact" element={<Contact />} />
```

Product.jsx

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

function Product() {
  const { id } = useParams();

  return <h2>Product ID: {id}</h2>;
}
```

```
export default Product;
```

Final Recap (VERY IMPORTANT)

- Router creates pages
- Routes connect URLs to components
- Link navigates safely
- Params make dynamic pages

Instructor Teaching Lines

- “URLs decide what the user sees.”
- “Link replaces anchor tags.”
- “Routing makes apps feel real.”

Next Module Preview

👉**Module 9 — Practice Project: Multi-Page App**