# App.js

**Abkürzung über einen kleinen Wrapper (Tipp von ChatGPT)**

Wenn du trotzdem pro Route verschiedene Fallbacks haben willst, aber keine Lust auf Copy-Paste hast, kannst du dir eine Hilfskomponente bauen:

function Loadable(Component, fallback = "Loading...") {

return (

<Suspense fallback={<div>{fallback}</div>}>

<Component />

</Suspense>

);

}

Und dann deine Routen so schreiben:

<Route index element={Loadable(<LitterOverview />, "Loading Overview…")} />

<Route path="whosNext" element={Loadable(<WhosNext />, "Loading Who’s Next…")} />

# Root.js

**Erweitertes Styling von nav (Tipp von ChatGPT)**

Stell dir vor, du willst nicht nur color: deeppink;, sondern zusätzlich ein Icon neben aktiven Links anzeigen. Mit der Callback-Variante könntest du isActive && "navlink active with-icon" zurückgeben und so gezielt mehr Styling oder sogar konditionalen Inhalt anhängen.

// NavigationMain.jsx

import { NavLink } from "react-router-dom";

export default function NavigationMain() {

return (

<nav>

<NavLink

to="/info"

className={({ isActive }) =>

isActive ? "navlink navlink--selected" : "navlink"

}

>

Info {({ isActive }) => isActive && "⭐"}

</NavLink>

<NavLink

to="/about"

className={({ isActive }) =>

isActive ? "navlink navlink--selected" : "navlink"

}

>

About {({ isActive }) => isActive && "⭐"}

</NavLink>

</nav>

);

}

/\* styles.css \*/

.navlink {

color: black;

text-decoration: none;

}

.navlink--selected {

color: deeppink;

font-weight: bold;

border-bottom: 2px solid deeppink;

}

Hier bekommst du:

Einen anderen Klassennamen (navlink--selected) statt active.

Ein ⭐-Icon direkt neben dem aktiven Link, weil du isActive auch im Inhalt nutzen kannst.

Volle Freiheit für komplexere Bedingungen (z. B. "navlink disabled" wenn eine Route gesperrt ist).

Soll ich dir noch ein Beispiel zeigen, wo man mit der Callback-Variante gleich mehrere Zustände unterscheiden kann? (z. B. isActive, isPending → während ein Link gerade geladen wird). Das zeigt ziemlich schön, warum man die Variante manchmal braucht