React Router

By TECH JASHWANTH

What is SPA

What is React-router-DOM

Why React-router-DOM

Installing React-router-DOM

What is BrowserRouter

Creating Different Pages for Routing

Using <Routes> & <Route>

Navigation Using <Link>

Creating a simple NAVBAR

Programmatic Navigation with useNavigate

Nested Routes

Catch-All Route for 404 Page

What is SPA?

What is SPA?

SPA stands for **Single Page Application.**

- It loads only one HTML page (index.html)
- The page never reloads during navigation
- Content updates using JavaScript and React Router
- Makes the app feel fast and smooth

⚙️ Key Features:

- 🚫 No full page reloads
- ⚡ Fast user experience
- 🔗 URL changes using JavaScript
- ✅ CDNs like Bootstrap load only once

Creating Website Pages

What is React Router DOM?

What is React Router DOM?

- React Router DOM is a library used in React applications to handle navigation.
- It allows users to move from one page to another without reloading the entire page.
- React is a Single Page Application (SPA) — it loads one HTML file and updates the view dynamically.

Why Use React Router?

Why Use React Router?

Without React Router:

- Clicking a link reloads the page.
- React components are lost and reloaded.

With React Router:

- Page doesn't reload.
- Only the view changes.
- Navigation is faster and smoother.
- ✅ Used for multi-page apps like blogs, dashboards, ecommerce sites, etc.

Installing React Router

BrowserRouter – The Main Wrapper

BrowserRouter – The Main Wrapper

- BrowserRouter is a component that wraps your entire app and enables routing features.
- Use it in the main file (like main.jsx)

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


<BrowserRouter>
  <App />
</BrowserRouter>
```

Using <Routes> & <Route> Components

Using <Routes> and <Route> Components

- <Routes> and <Route> are the building blocks for routing in React apps.

Routes:

- A wrapper component from react-router-dom
- It holds all the <Route> components
- It checks the current URL and shows the matching component

Route:

- Defines a single route
- Has two main props:
- path: the URL path
- element: the component to show

Using <Routes> and <Route> Components

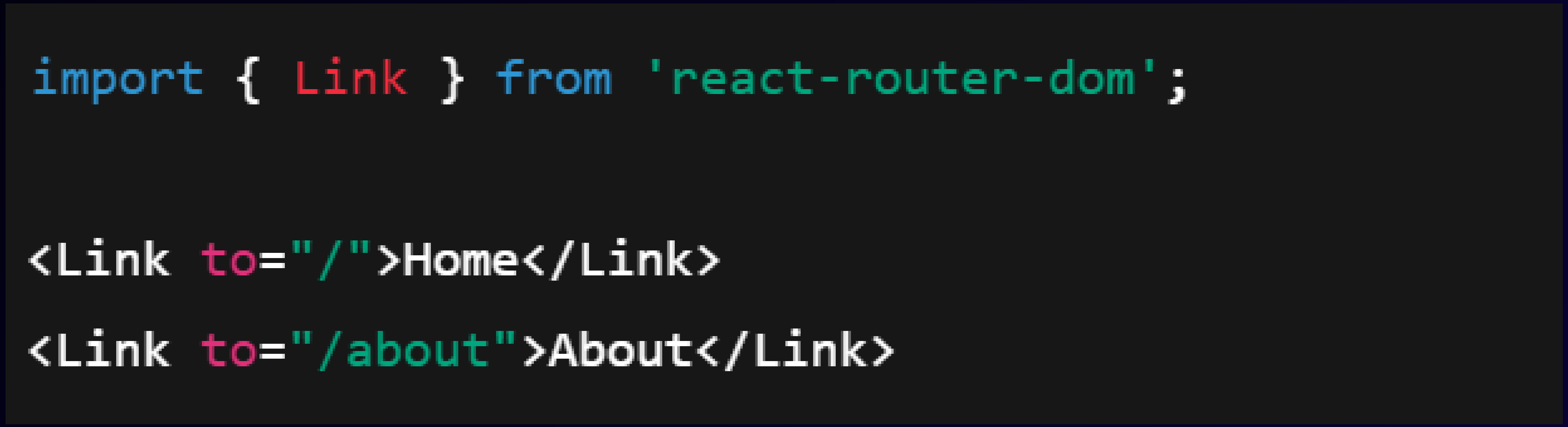
```
import { Routes, Route } from 'react-router-dom';
import Home from './pages/Home';
import About from './pages/About';

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

Navigation Using <Link>

Navigation Using <Link>

- Use Link instead of the <a> tag to move between pages without reloading.



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


<Link to="/">Home</Link>
<Link to="/about">About</Link>
```

- **<Link> works without refreshing the page.**
- **to** defines where to go.

Creating a Simple Navbar

Dynamic Routes Using URL Parameters

Dynamic Routes Using URL Parameters

- You can create paths that accept dynamic values (like an ID):

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

- You can access the id in the User component:

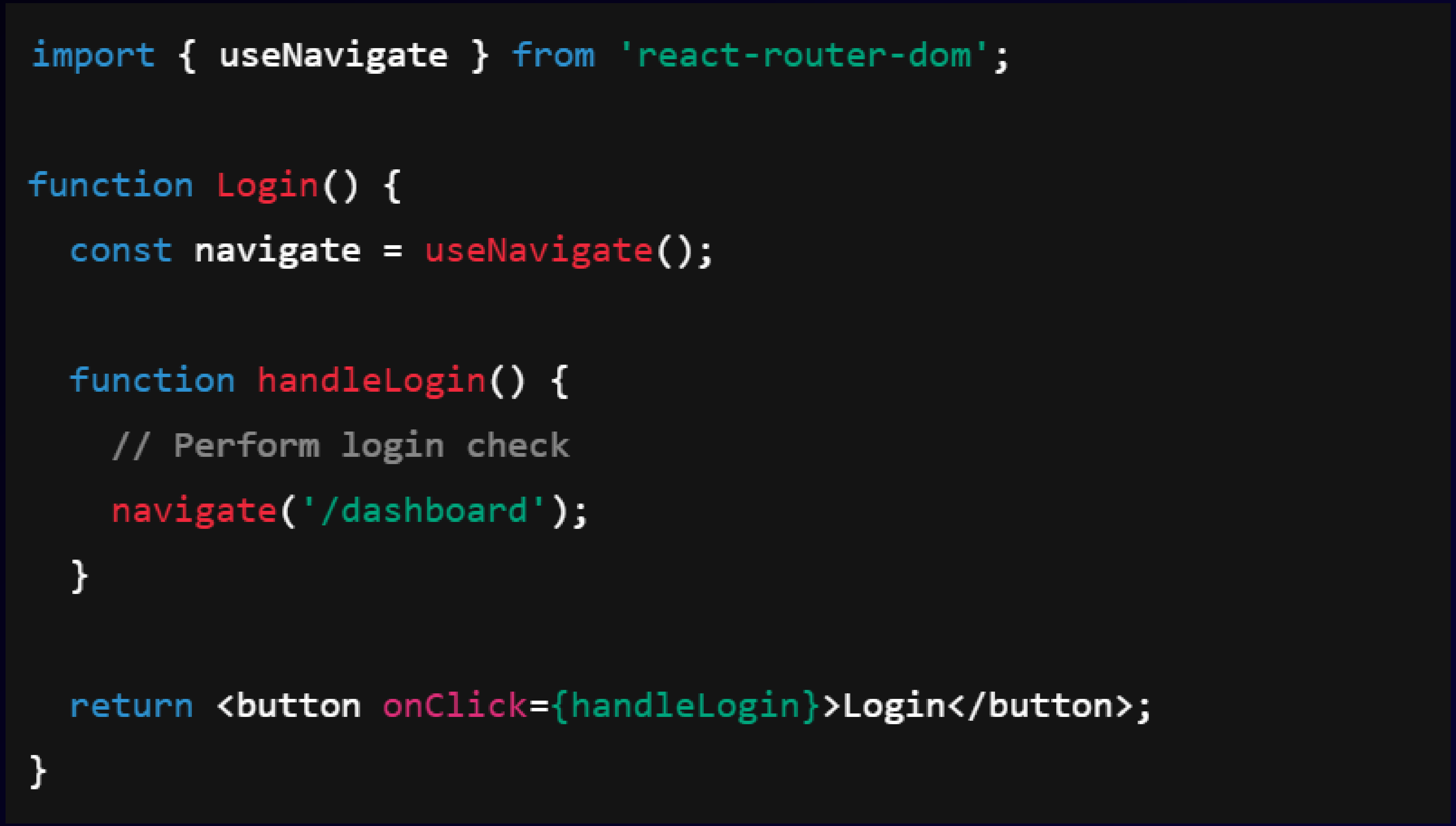
```
import { useParams } from 'react-router-dom';


function User() {
  const { id } = useParams();
  return <h2>User ID is {id}</h2>;
}
```

Programmatic Navigation with useNavigate

Programmatic Navigation with useNavigate

- Sometimes you need to navigate from code (like after login)
- useNavigate() gives you a function to go to another route using code.



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

function Login() {
  const navigate = useNavigate();

  function handleLogin() {
    // Perform login check
    navigate('/dashboard');
  }

  return <button onClick={handleLogin}>Login</button>;
}
```

Nested Routes (Child Routes)

Nested Routes (Child Routes)

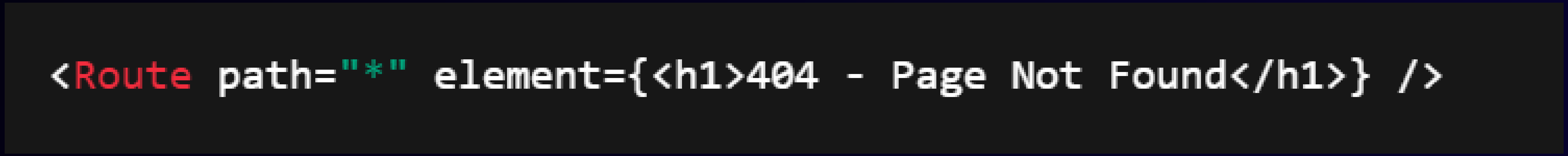
- Nested routes let you show pages inside pages.

```
<Route path="/dashboard" element={<Dashboard />}>
  <Route path="profile" element={<Profile />} />
  <Route path="settings" element={<Settings />} />
</Route>
```

Catch-All Route for 404 Page

Catch-All Route for 404 Page

- This route shows when no other route matches.



```
<Route path="*" element={<h1>404 - Page Not Found</h1>} />
```

Subscribe to

Tech Jashwanth

To Learn Full-stack Development

Youtube Link **Instagram Link** **Telegram Link**

-