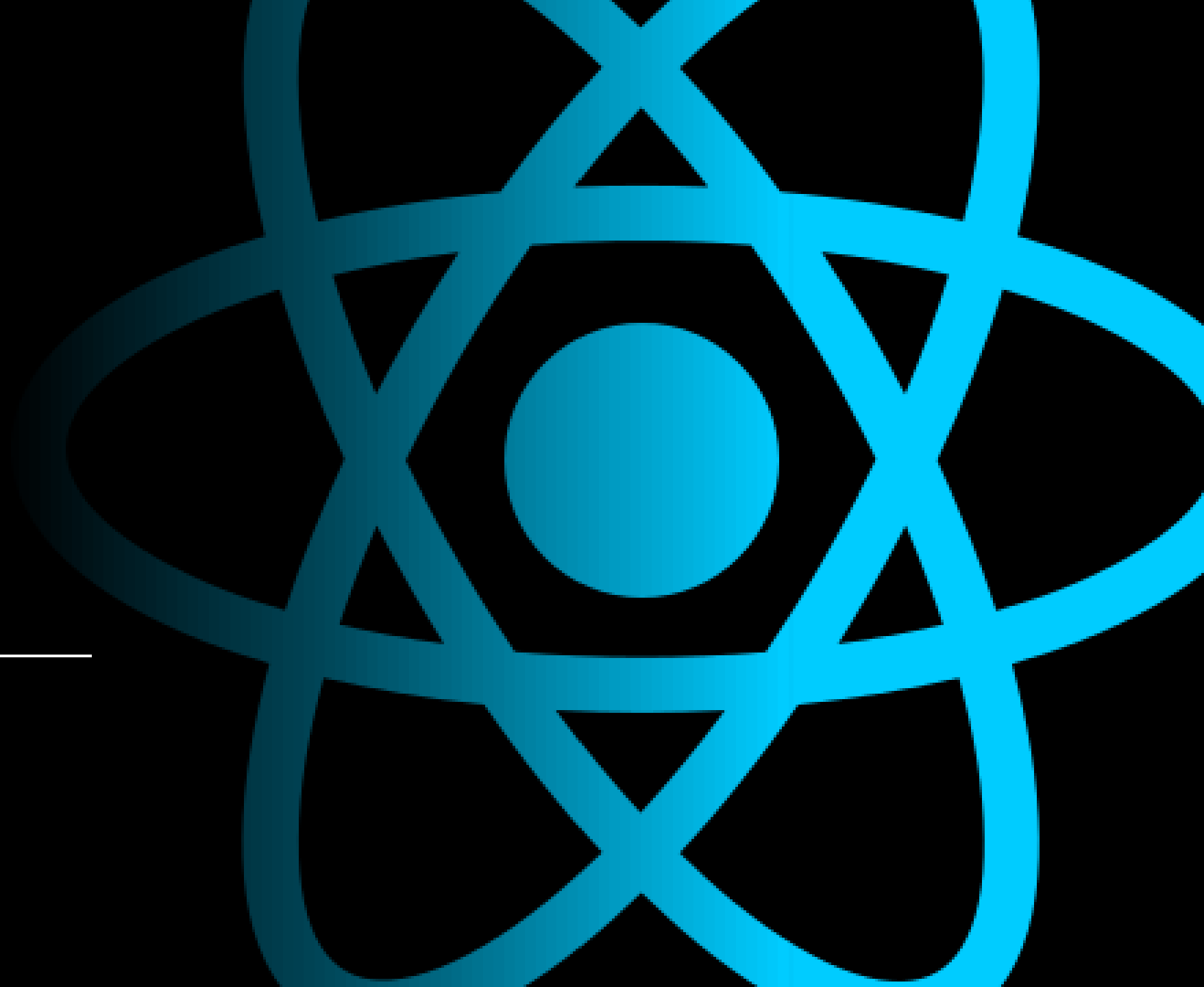


1. Basic routing
2. useHistory
3. useLocation
4. useParams
5. BrowserRouter
6. Link
7. NavLink
8. Nesting
9. 404



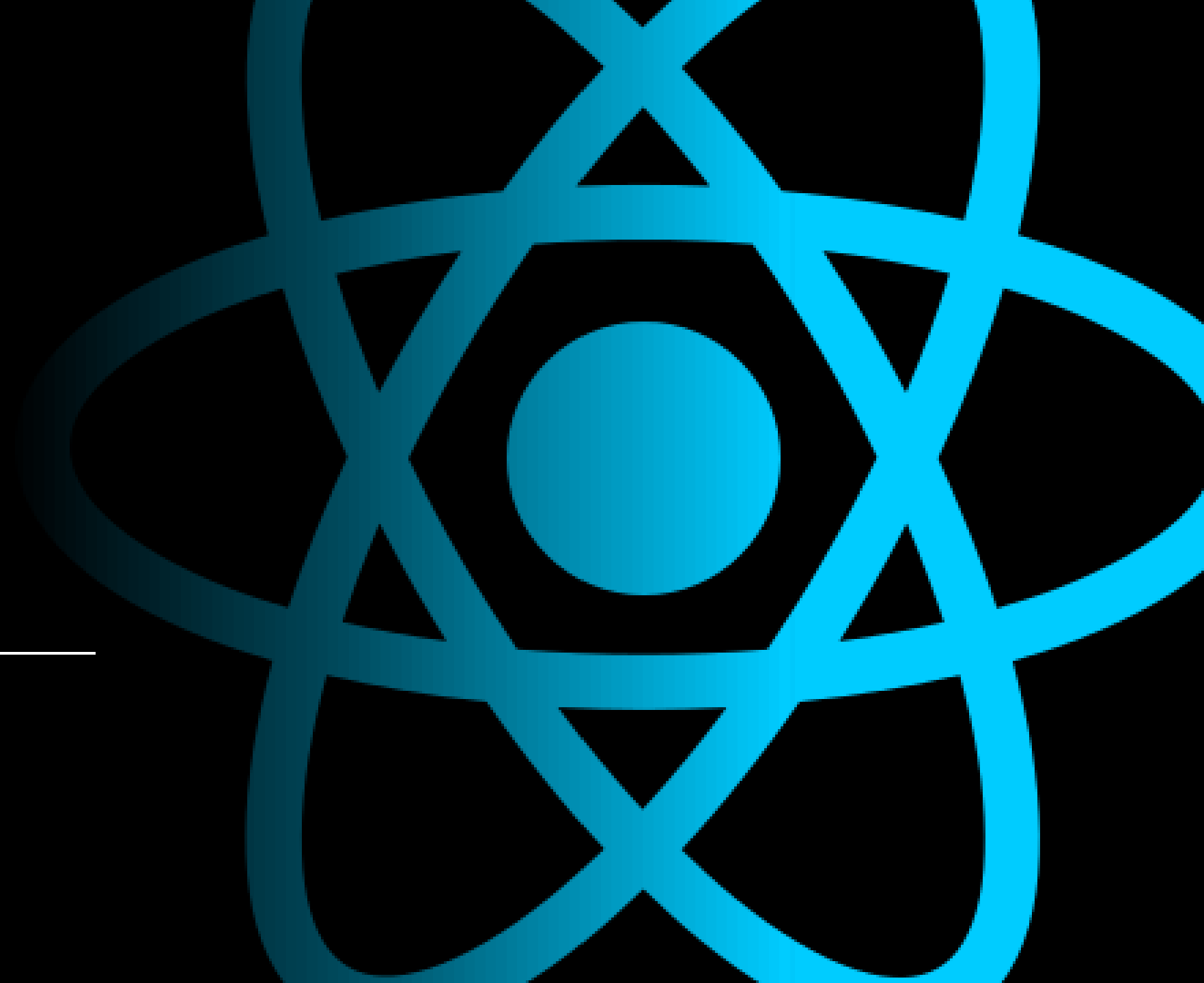


Basic routing

```
Yarn add @types/react-router-dom react-router-dom
```

```
import { BrowserRouter as Router, Switch, Route } from "react-router-dom";

return (
  <Router>
    <Wrapper>
      <Switch>
        <Route path="/entities" exact>
          <Entities />
        </Route>
        <Route path="/" exact>
          <div>strona główna</div>
          <button onClick={clickHandler}>Kliknij mnie</button>
        </Route>
      </Switch>
    </Wrapper>
  </Router>
);
```



Hooks

useHistory jest hookiem dającym dostęp do historii, którą możemy używać do nawigacji

```
import { useHistory } from "react-router-dom";

function HomeButton() {
  let history = useHistory();

  function handleClick() {
    history.push("/home");
  }

  return (
    <button type="button" onClick={handleClick}>
      Go home
    </button>
  );
}
```

useLocation zwraca nam obiekt reprezentujący obecny adres. Za każdym razem gdy zmieni się adres useLocation zwraca nowe dane.

```
import React from "react";
import ReactDOM from "react-dom";
import {
  BrowserRouter as Router,
  Switch,
  useLocation
} from "react-router-dom";

function usePageViews() {
  let location = useLocation();
  React.useEffect(() => {
    ga.send(["pageview", location.pathname]);
  }, [location]);
}

function App() {
  usePageViews();
  return <Switch>...</Switch>;
}

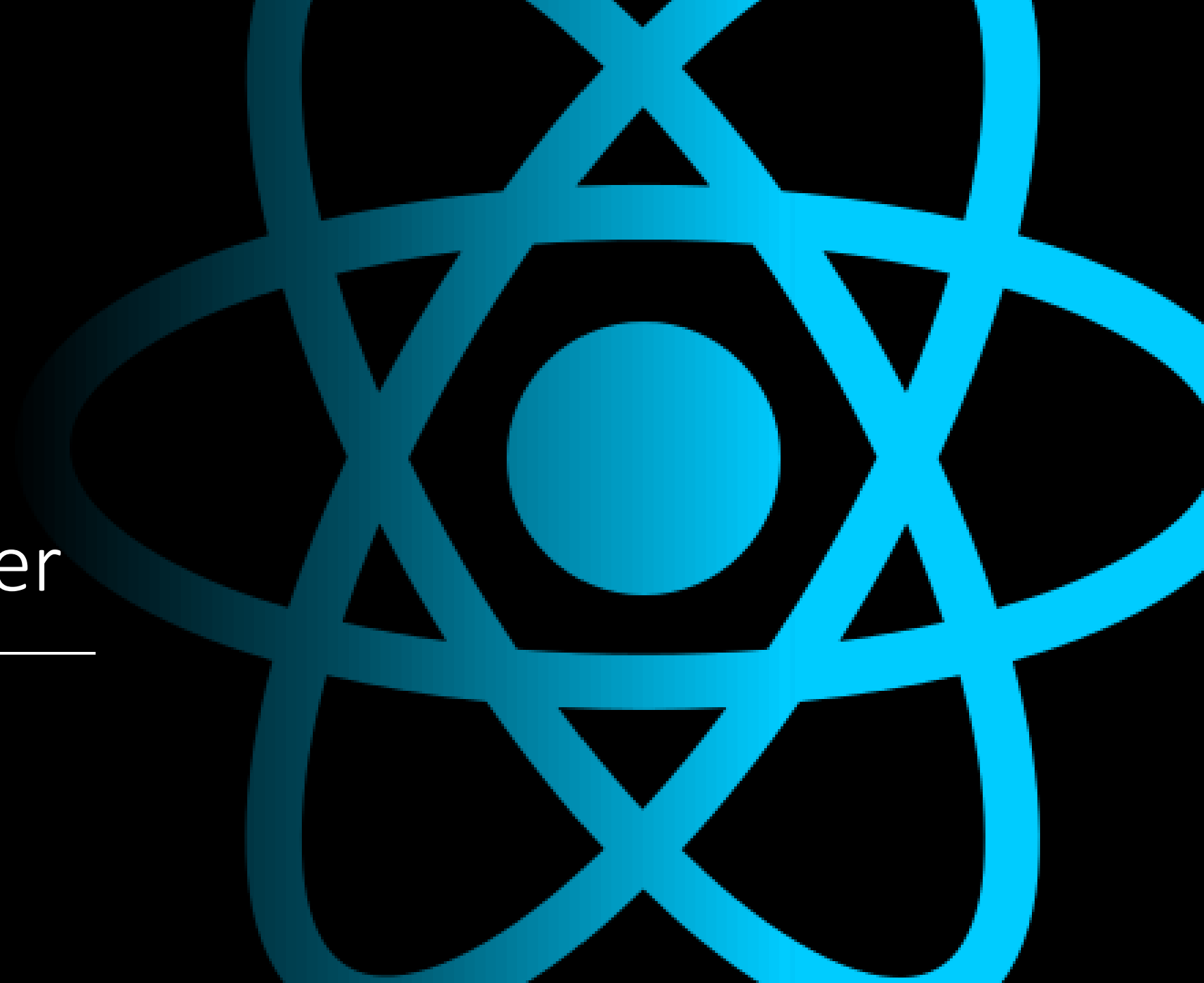
ReactDOM.render(
  <Router>
    <App />
  </Router>,
  node
);
```


useParams zwraca obiekt (key/value) na podstawie parametrów z URL.

```
import React from "react";
import ReactDOM from "react-dom";
import {
  BrowserRouter as Router,
  Switch,
  Route,
  useParams
} from "react-router-dom";

function BlogPost() {
  let { slug } = useParams();
  return <div>Now showing post {slug}</div>;
}

ReactDOM.render(
  <Router>
    <Switch>
      <Route exact path="/">
        <HomePage />
      </Route>
      <Route path="/blog/:slug">
        <BlogPost />
      </Route>
    </Switch>
  </Router>,
  node
);
```



BrowserRouter

Komponent który korzysta z interface API historii HTML5.

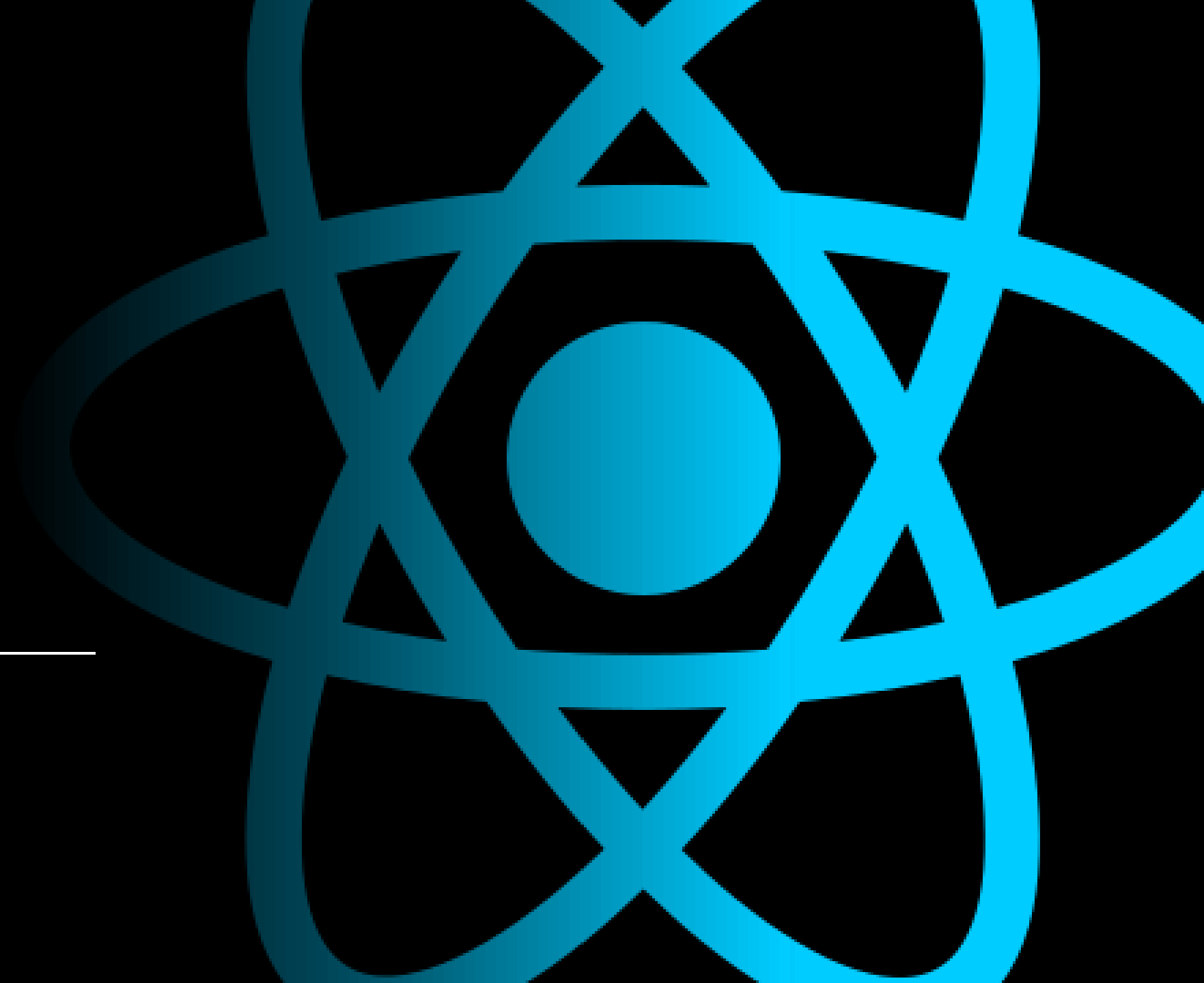
Wszystkie komponenty w których chcemy korzystać z routingu muszą znajdować się wewnątrz komponentu

`<Router>`

```
return (  
  <Router>  
    <Wrapper>  
      <Switch>  
        <Route path="/entities" exact>  
          <Entities />  
        </Route>  
        <Route path="/" exact>  
          <div>strona główna</div>  
          <button onClick={clickHandler}>Kliknij mnie</button>  
        </Route>  
      </Switch>  
    </Wrapper>  
  </Router>  
)  
);
```



Link



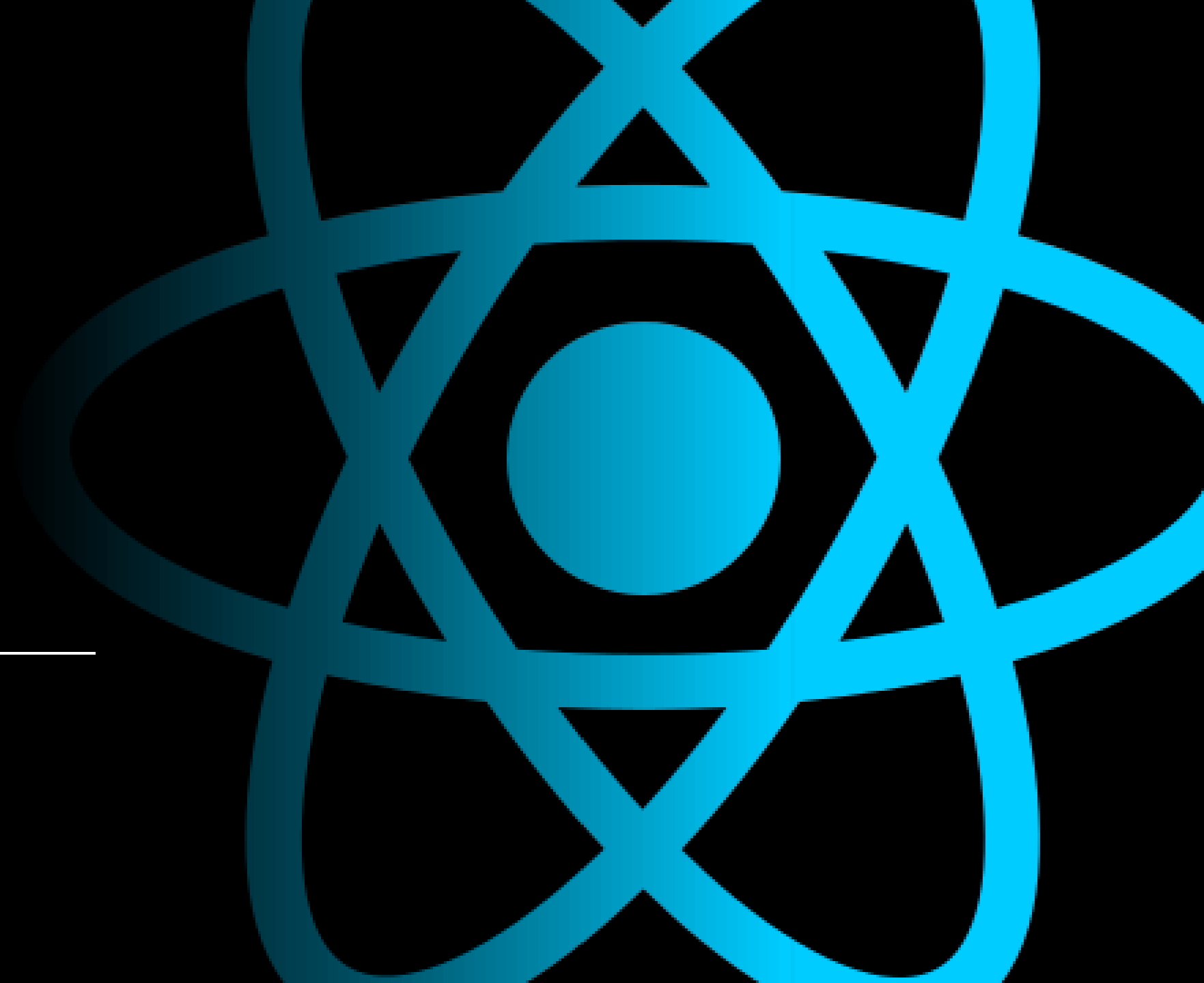
```
<Link to="/about">About</Link>
```

```
<Link  
  to={{  
    pathname: "/courses",  
    search: "?sort=name",  
    hash: "#the-hash",  
    state: { fromDashboard: true }  
  }}  
>
```

```
<Link to={location => ({ ...location, pathname: "/courses" })} /><Link to={location => `${location.pathname}?sort=name`} />
```

```
const FancyLink = React.forwardRef((props, ref) => (  
  <a ref={ref} {...props}>👉 {props.children}</a>  
))
```

```
<Link to="/" component={FancyLink} />
```



NavLink

```
<NavLink to="/about">About</NavLink>
```

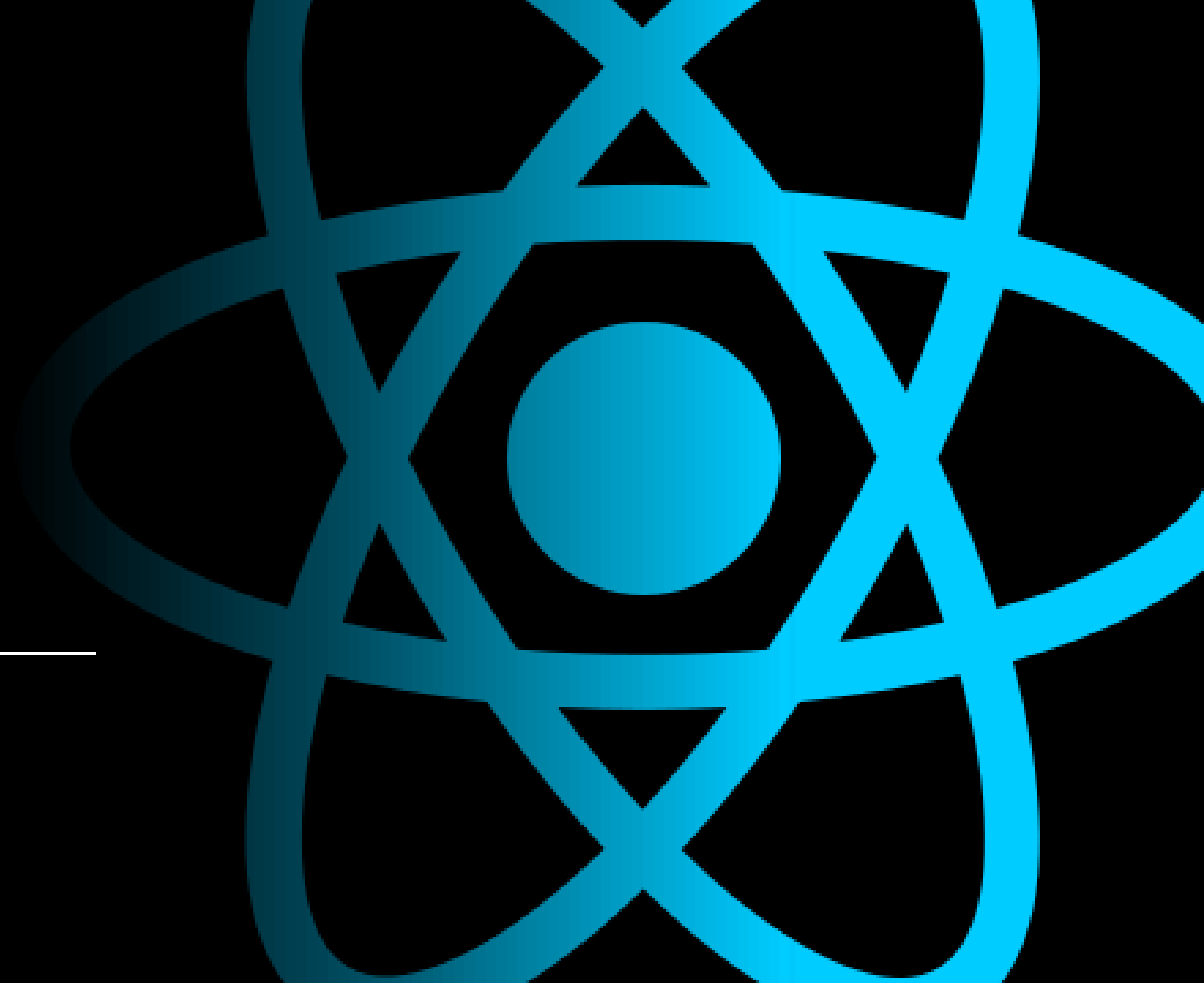
```
<NavLink to="/faq" activeClassName="selected">  
  FAQs  
</NavLink>
```

```
<NavLink  
  to="/events/123"  
  isActive={(match, location) => {  
    if (!match) {  
      return false;  
    }  
  
    // only consider an event active if its event id is an odd number  
    const eventId = parseInt(match.params.eventID);  
    return !isNaN(eventID) && eventID % 2 === 1;  
  }}  
>  
  Event 123  
</NavLink>
```

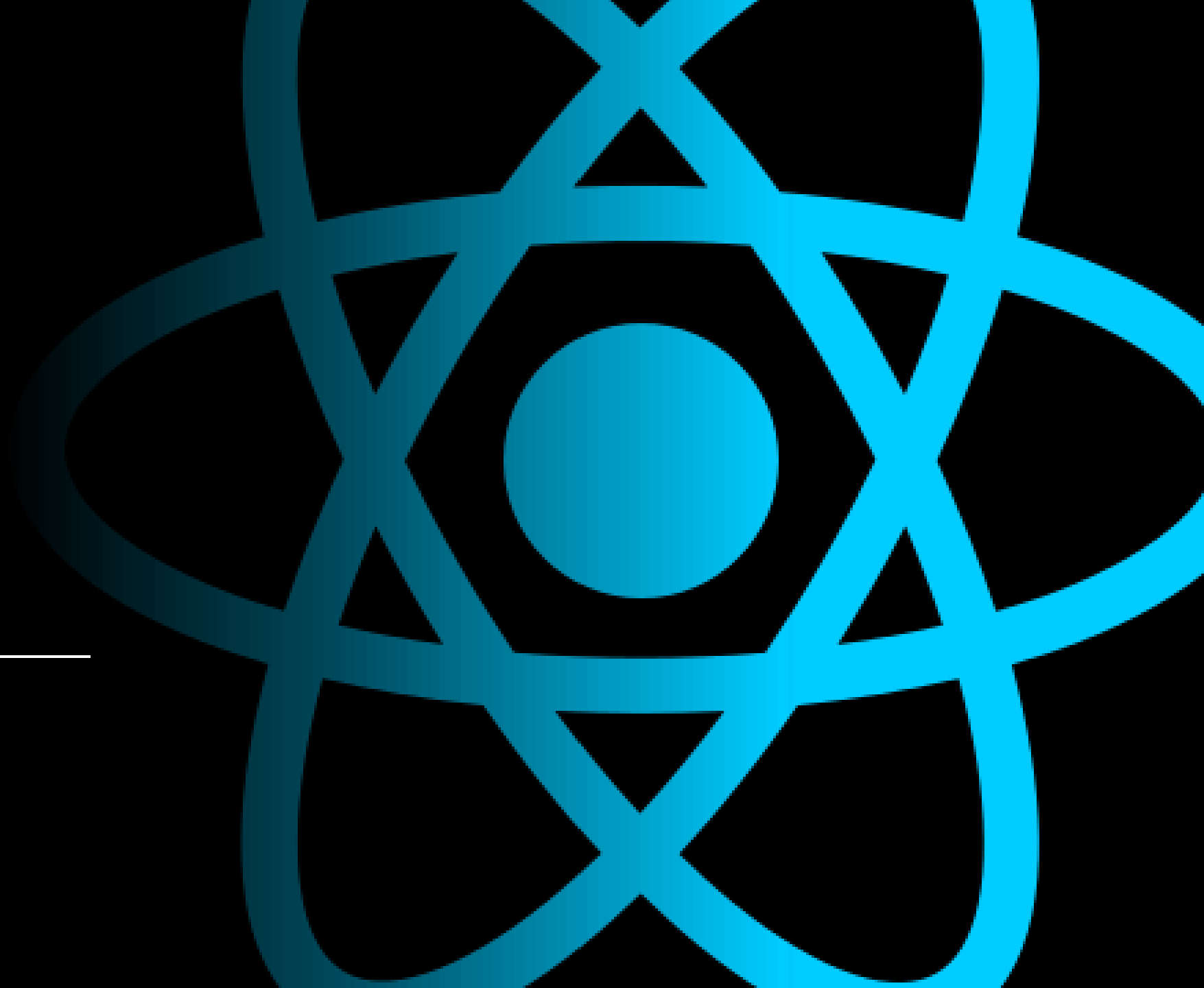
```
<NavLink  
  to="/faq"  
  activeStyle={{  
    fontWeight: "bold",  
    color: "red"  
  }}  
>  
  FAQs  
</NavLink>
```




Nesting



<https://reactrouter.com/web/example/nesting>



404

<https://reactrouter.com/web/example/no-match>