Summary

This is a cheat sheet. You can find here all the necessary rules and code templates that will help you complete home assignments in this lesson. If you find it useful, feel free to copy the code blocks and change them however you need.

Useful Components

Component	When to Use
<BrowserRouter>	To use the HTML5 history API and sync the UI with the URL in the browser.
<Route>	To define a route in the app. It takes a path and component and renders the component when the URL matches the path.
<Route> with a render prop	To use a render prop instead of a component prop. It's useful when you need more control over what is rendered based on the route.
<Route> with a children prop	To use a children prop instead of a component prop. It's useful when you need to render multiple components based on the route.
<Switch>	To group the <Route> components. The component renders only the first <Route> that matches the URL. It's useful when you have multiple routes with similar paths.
<Link>	To navigate between pages in the app. The component creates an anchor tag with a specified URL and prevents page refreshing.
<NavLink>	To create a navigation link. <NavLink> is similar to the <Link> component, but it adds styling to indicate the active link.
<Redirect>	To redirect a user to another URL.
<Prompt>	To display a message to a user before leaving the current page.
<Router>	To create a custom router. The component takes a history prop, which specifies the type of history object to use. It's useful when you need to use a custom history object. For example, to integrate a third-party routing library.

Useful Hooks

Hook	When to Use
useHistory()	To access the history object. The history object contains methods to manipulate the browser's history. This helps a computer navigate different routes in response to user actions or app state changes.
useParams()	To access the current route’s parameters. The hook returns an object with key-value pairs of the parameters. This helps you dynamically render the content based on the current route. For example, display information about a particular item in a list.
useLocation()	To access the current location object. The location object contains information about the current URL. This helps you conditionally render components based on the current URL. For example, highlight the active link in a navigation menu.