

## 1. React Router

Step 1 : npm i react-router-dom

Step 2: react-router-dom ke import kore nite hobe, from **App.jsx**, import snippet,

```
import {
  Route,
  createBrowserRouter,
  createRoutesFromElements,
  RouterProvider,
} from "react-router-dom";
```

Step 3: After importing, implement router,

```
const router = createBrowserRouter(
  createRoutesFromElements(<Route index element={<h1>My App</h1>} />)
);
```

Step 4: Then etake const App er moddhe return korate hobe,

```
const App = () => {
  return <RouterProvider router={router} />;
};
```

From **App.jsx** full boiler plate snippet,

```
import React from "react";
import {
  Route,
  createBrowserRouter,
  createRoutesFromElements,
  RouterProvider,
} from "react-router-dom";

const router = createBrowserRouter(
  createRoutesFromElements(<Route index element={<h1>My App</h1>} />)
);

const App = () => {
  return <RouterProvider router={router} />;
};

export default App;
```

## 2. React Router pages

Suppose a user route link has <http://localhost:3000/about> then evaluate point has, from App.jsx,

```
const router = createBrowserRouter(  
  createRoutesFromElements(<Route path="/about" element={<h1>My App</h1>} />  
)  
);
```

Just index page like <http://localhost:3000/> e nite hole path="" er jaygay just index likhlei hobe, from App.jsx,

```
const router = createBrowserRouter(  
  createRoutesFromElements(<Route index element={<h1>My App</h1>} />  
)  
);
```

#importing components and adding on routes,

```
import HomePage from "../pages/HomePage";  
  
const router = createBrowserRouter(  
  createRoutesFromElements(<Route index element={<HomePage></HomePage>} />  
)  
);
```

## 3. React Outlet

Mainly concept ta as a wrapper hishebe kaj kore, like **MainLayout.jsx** as a wrapper and tar vitorer **jsx** file child element hishebe thakbe, like components from Laravel,

Step 1: **MainLayout.jsx** naam e ekta component create kora holo

Step 2: From **App.jsx**,

```
const router = createBrowserRouter(  
  createRoutesFromElements(  
    <Route path="/" element={<MainLayout></MainLayout>}>  
      <Route index element={<HomePage></HomePage>} />  
    </Route>  
  )  
);
```

- MainLayout er under e baki gula wrapper

Step 3: From **MainLayout.jsx**,

- Firstly import Outlet,

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

Then use it as a component, Full code snippet,

```
import React from "react";
import { Outlet } from "react-router-dom";

const MainLayout = () => {
  return (
    <>
      <Outlet></Outlet>
    </>
  );
};

export default MainLayout;
```

#### 4. React Outlet with components example,

Suppose Amader main <Outlet> component er upor kono ekta component thakbe, tokhon amra evabe Components add korte parbo,

```
<NavBar></NavBar>
<Outlet></Outlet>
```

Full code snippet,

```
import React from "react";
import { Outlet } from "react-router-dom";
import NavBar from "../components/NavBar";

const MainLayout = () => {
  return (
    <>
      <NavBar></NavBar>
      <Outlet></Outlet>
    </>
  );
};

export default MainLayout;
```

It acts like <x-admin-master> as an example from Laravel

## 5. React Router – Links

Step 1: from jsx files, import **Link**,

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

Step 2: Instead of anchor tags, we use links,

```
<Link to="/jobs" className="text-white hover:bg-gray-900 hover:text-white rounded-md px-3 py-2">
```

## 6. React Router – Not Found Route

If any of the routes are not found, we can use this as a not found route,

Its from **App.jsx**,

```
<Route path="*" exact={true} element={<NotFoundPage />} />
```

Full code snippet from App.jsx,

```
const router = createBrowserRouter(  
  createRoutesFromElements(  
    <Route path="/" element={<MainLayout></MainLayout>}>  
      <Route index element={<HomePage></HomePage>} />  
      <Route path="/jobs" element={<JobsPage></JobsPage>} />  
      <Route path="*" exact={true} element={<NotFoundPage />} />  
    </Route>  
  )  
);
```

## 7. React Router redirection suggestion from Stackoverflow



If you want to display a 404 and **keep the path** (Same functionality as NotFoundRoute)



```
<Route path='*' exact={true} component={My404Component} />
```

If you want to display a 404 page but **change the url** (Same functionality as DefaultRoute)

```
<Route path='/404' component={My404Component} />  
<Redirect from='*' to='/404' />
```

Example with multiple levels:

```
<Route path='/' component={Layout} />  
  <IndexRoute component={MyComponent} />  
  <Route path='/users' component={MyComponent}>  
    <Route path='user/:id' component={MyComponent} />  
    <Route path='*' component={UsersNotFound} />  
  </Route>  
  <Route path='/settings' component={MyComponent} />  
  <Route path='*' exact={true} component={GenericNotFound} />  
</Route>
```