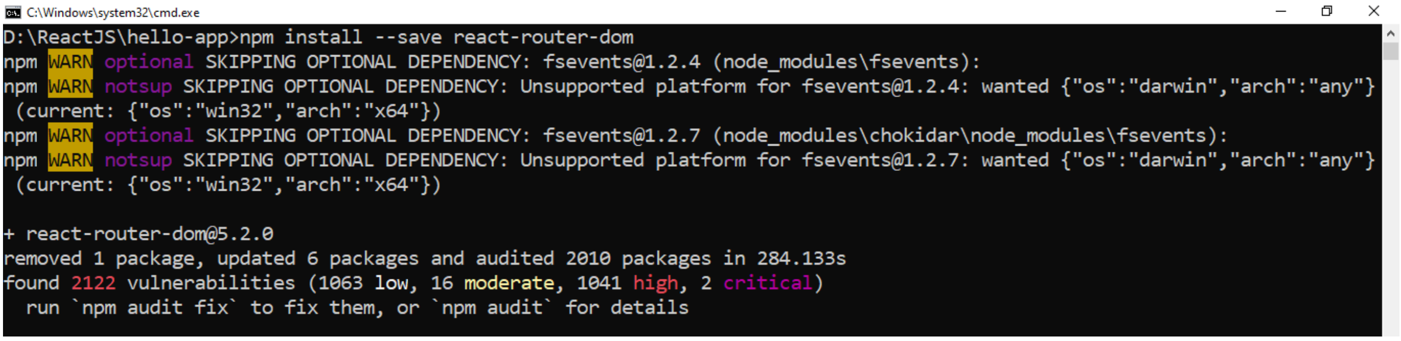
Working with Routing

- The “Routing” concept is used to create page navigation in a single page application (SPA).
- Routing enables navigation from one view to the next view (within the same page) as users perform different tasks.
- We use **React-Router** which is a Popular Library for implementing Routing.
- For a **web app**, we’ll use **react-router-dom**
- For a **Mobile app**, we’ll use **react-router-native**.
- Since ours is a Web App, lets install **react-router-dom**.

Reference:
https://reactrouter.com/web/guides/quick-start

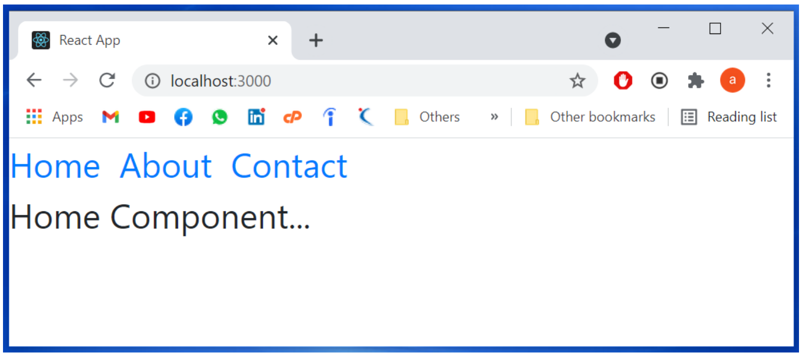
Installation of react-router-dom module:
Open the CMD and run the fallowing command from your project.

D:\ReactJS\hello-app>npm install --save react-router-dom



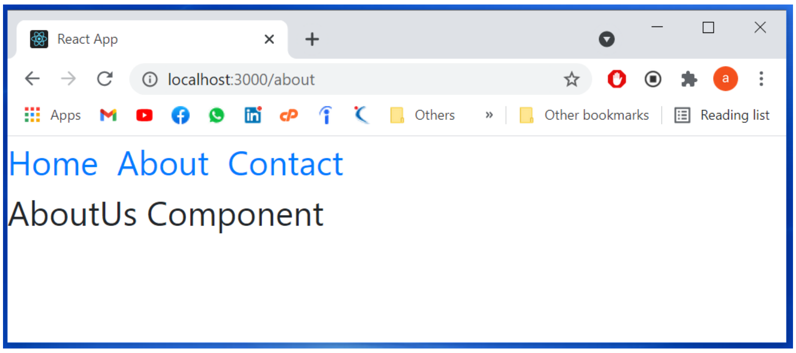
```
D:\ReactJS\hello-app>npm install --save react-router-dom
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.4 (node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.4: wanted {"os":"darwin","arch":"any"}
 (current: {"os":"win32","arch":"x64"})
npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@1.2.7 (node_modules\chokidar\node_modules\fsevents):
npm WARN notsup SKIPPING OPTIONAL DEPENDENCY: Unsupported platform for fsevents@1.2.7: wanted {"os":"darwin","arch":"any"}
 (current: {"os":"win32","arch":"x64"})

+ react-router-dom@5.2.0
removed 1 package, updated 6 packages and audited 2010 packages in 284.133s
found 2122 vulnerabilities (1063 low, 16 moderate, 1041 high, 2 critical)
  run `npm audit fix` to fix them, or `npm audit` for details
```

Example:



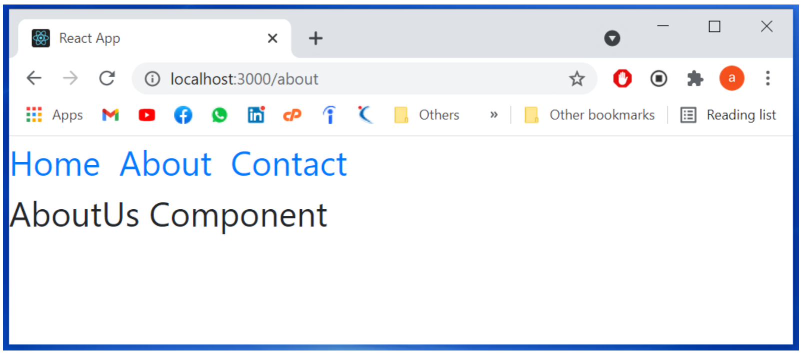
Home About Contact

Home Component...



Home About Contact

AboutUs Component



Home About Contact

AboutUs Component

src\components\Home.js

```
function Home(){
    return <p>Home Component...</p>
}

export default Home
```

src\components\About.js

```
function About(){
    return <p>AboutUs Component</p>
}

export default About
```

hello-app\src\components\Contact.js

```
function Contact(){
    return <p>Contact Us Component...</p>
}

export default Contact
```

hello-app\src\App.js

```
import "./App.css";
import { BrowserRouter, Link, NavLink, Route, Switch } from "react-router-dom";
import Home from "./components/Home";
import About from "./components/About";
import Contact from "./components/Contact";

function App() {
  return (
    <BrowserRouter>
      <div className="customClass">
        <Link to="/" >Home</Link>&nbsp;&nbsp;
        <NavLink to="/about" >About</NavLink>&nbsp;&nbsp;
        <NavLink to="/contact">Contact</NavLink>&nbsp;&nbsp;
        <Switch>
          <Route exact path="/" component={Home}></Route>
          <Route exact path="/about" component={About}></Route>
          <Route exact path="/contact" component={Contact}></Route>
        </Switch>
      </div>
    </BrowserRouter>
  );
}
export default App;
```

hello-app\src\index.js

```
import ReactDOM from "react-dom";
import React, { Component } from "react";
import App from './App';

ReactDOM.render(<App/>,document.getElementById("root"));
```

BrowserRouter:

- It wraps all the other components
- It is a parent component and can have only single child.

Link:

- It links to the components.
- Link component uses the **'to'** prop to describe the location where the links should navigate to.

Route:

- It used to establish the link between component's UI and the URL.
- Route component Props.
  1. **exact**: It is used to match the exact value with the URL
  2. **path**: Path specifies a pathname we assign to our component
  3. **component**: It refers to the component which will render on matching the path.

Switch:

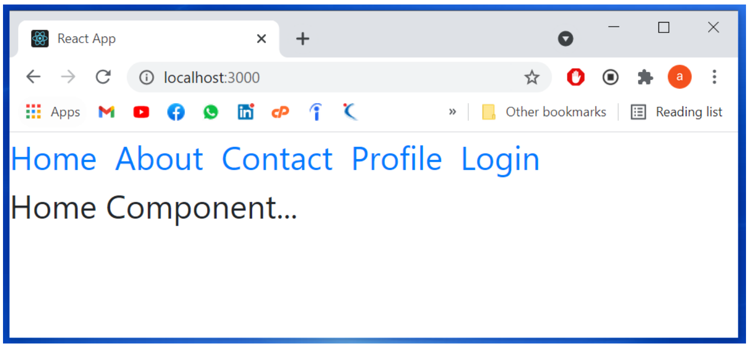
- By default, routes are inclusive which means more than one Route component can match the URL path and render at the same time.
- To render a single component, wrap all the routes inside the **Switch** Component.

Handling Secured Routes

- Based on the programing requirement sometimes we have to secure the routes from unauthorized users.

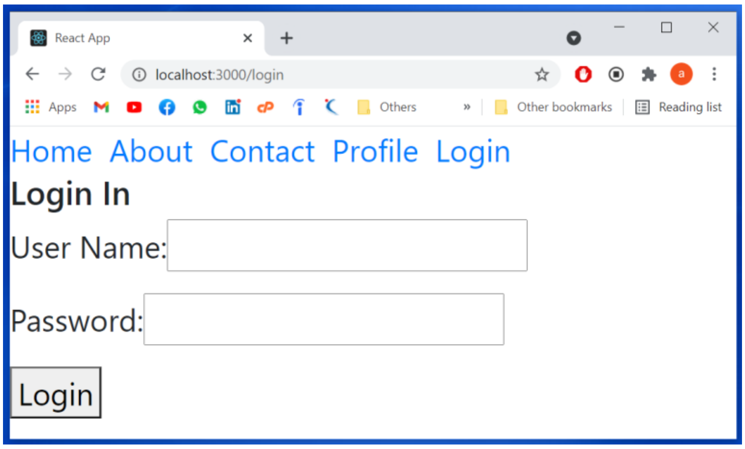
Use Case:

- Create 3 components as **Home, About, Contact, Login and Profile**.
- Provide access to **Home, About, Contact and Login** to all users.
- Whenever user try to access **Profile route**, redirect to Login component and user needs to enter credentials.



Home About Contact Profile Login

Home Component...



Home About Contact Profile Login

Login In

User Name:

Password:

Login

src\components\Home.js

```
function Home(){
    return <p>Home Component...</p>
}

export default Home
```

src\components\Contact.js

```
function Contact(){
    return <p>Contact Us Component...</p>
}

export default Contact
```

src\components\About.js

```
function About(){
    return <p>AboutUs Component</p>
}

export default About
```

src\components\Profile.js

```
function Profile(){
    return <p>Profile Component...</p>
}

export default Profile
```

src\components\Authentication.js

```
const authentication = {
    isLoggedIn: false,
    onAuthentication(){
        this.isLoggedIn = true
    },
    getLoginStatus(){
        return this.isLoggedIn
    }
}

export default authentication
```

src\components\Login.js

```
import { useState } from "react";
import authentication from "./Authentication";

function LogIn(props) {
  const initialState = { userName: "", password: "" };
  const [loginData, setLoginData] = useState(initialState);

  function changeLoginData(e) {
    setLoginData({ ...loginData, [e.target.name]: e.target.value });
  }
  function onLogin() {
    if (loginData.userName === "admin" && loginData.password === "admin123") {
      //success
      authentication.onAuthentication();
      props.history.push("/profile");
    } else {
      //failed
      alert("Invalid Login");
    }
  }
  return (
    <div>
      <h2>Login In</h2>
      <p>
        <label>User Name:</label>
        <input
          type="text"
          value={loginData.userName}
          name="userName"
          onChange={changeLoginData}
        ></input>
      </p>
      <p>
        <label>Password:</label>
        <input
          type="password"
          value={loginData.password}
          name="password"
          onChange={changeLoginData}
        ></input>
      </p>
      <button onClick={onLogin}>Login</button>
    </div>
  );
}
export default LogIn;
```

src\components\SecuredRoute.js

```
import { Redirect, Route } from "react-router";
import authentication from "./Authentication"

function SecuredRoute(props) {
  return (
    <Route
      path={props.path}
      render={(data) =>
        authentication.getLoginStatus() ? (
          <props.component {...data}></props.component>
        ) : (
          <Redirect to={{ pathname: "/login" }}></Redirect>
        )
      }
    ></Route>
  );
}
export default SecuredRoute
```

<Redirect>

Rendering a <Redirect> will navigate to a new location.

The new location will override the current location in the history stack, like server-side redirects (HTTP 3xx) do.

src\App.js

```
import "./App.css";
import { BrowserRouter, Link, NavLink, Route, Switch } from "react-router-dom";
import Home from "./components/Home";
import About from "./components/About";
import Contact from "./components/Contact";
import Profile from "./components/Profile";
import LogIn from "./components/Login"
import SecuredRoute from "./components/SecuredRoute"

function App() {
  return (
    <BrowserRouter>
      <div className="customClass">
        <Link to="/" >Home</Link>&nbsp;&nbsp;
        <NavLink to="/about" >About</NavLink>&nbsp;&nbsp;
        <NavLink to="/contact">Contact</NavLink>&nbsp;&nbsp;
        <NavLink to="/profile">Profile</NavLink>&nbsp;&nbsp;
        <NavLink to="/login">Login</NavLink>

        <Switch>
          <Route exact path="/" component={Home}></Route>
          <Route exact path="/about" component={About}></Route>
          <Route exact path="/contact" component={Contact}></Route>
          <SecuredRoute path="/profile" component={Profile}></SecuredRoute>
          <Route exact path="/login" component={LogIn}></Route>
        </Switch>
      </div>
    </BrowserRouter>
  );
}
export default App;
```

src\index.js

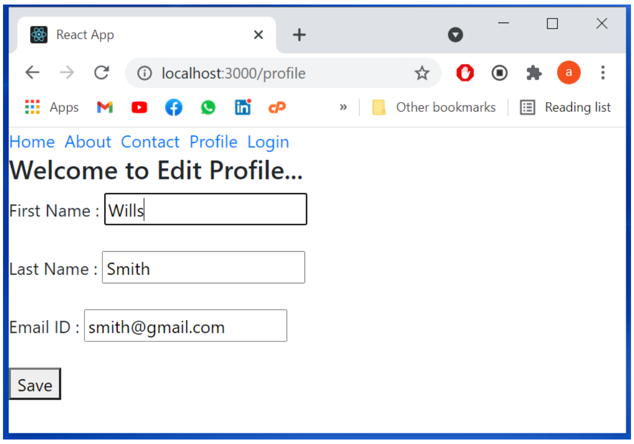
```
import React, { useState } from "react";
import ReactDOM from "react-dom";

const element = <App/>
ReactDOM.render(element, document.getElementById("root"));
```

Prevent Navigation in Routing

- ➢ Based on the programing requirement sometimes we have to prevent navigations on routes.

Use Case:

- ➢ When user is performing any modification on **Profile**, prevent user to navigate to other navigations without saving any changes.

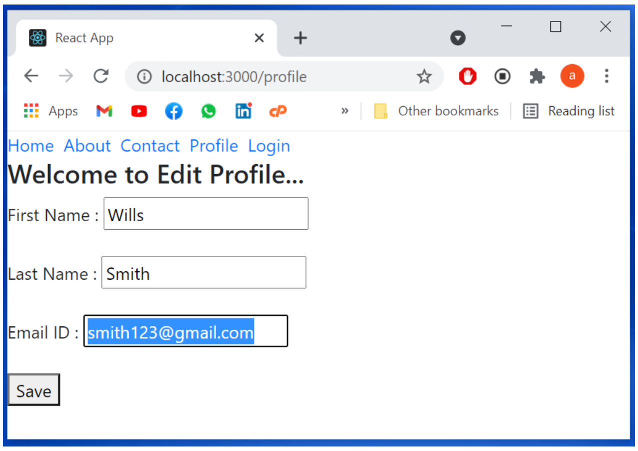


Welcome to Edit Profile...

First Name : Wills

Last Name : Smith

Email ID : smith@gmail.com

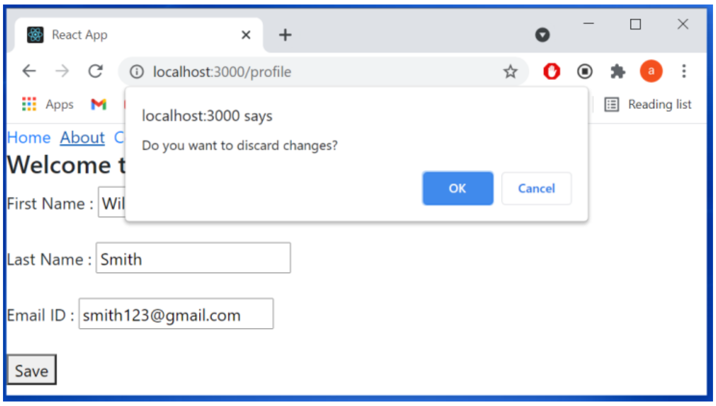


Welcome to Edit Profile...

First Name : Wills

Last Name : Smith

Email ID : smith123@gmail.com



localhost:3000 says

Do you want to discard changes?

Welcome t

First Name : Wil

Last Name : Smith

Email ID : smith123@gmail.com

Example:

Update Profile.js file as fallows.

hello-app\src\components\Profile.js

```
import { useState } from "react";
import { Prompt } from 'react-router-dom';

function Profile() {

  let [isDataChanged, setDataChanged] = useState(false);

  const initialData = {firstName: 'Wills', lastName: 'Smith', emailId: 'smith@gmail.com'}

  let [userInfo, setUserInfo] = useState(initialData);

  let [updateMsg, setUpdateMsg] = useState()

  function saveChanges() {
      setDataChanged(false);
      setUpdateMsg("Data Updated")
  }

  function onChangeHandler(e) {
      setUserInfo({ ...userInfo, [e.target.name]: e.target.value });
      setDataChanged(true);
  }
  return (
      <div>
          <p style={{color:"red"}}>{updateMsg}</p>
          <Prompt when={isDataChanged}
              message={() => "Do you want to discard changes?"}></Prompt>
          <h4>Welcome to Edit Profile...</h4>
          <p>
              <label>First Name : <input type="text" name="firstName"
                  onChange={onChangeHandler}
                  value={userInfo.firstName}></input></label>
          </p>
          <p>
              <label>Last Name : <input type="text" name="lastName"
                  onChange={onChangeHandler} value={userInfo.lastName}></input></label>
          </p>
          <p>
              <label>Email ID : <input type="text" name="emailId"
                  onChange={onChangeHandler}
                  value={userInfo.emailId}></input></label>
          </p>
          <button onClick={saveChanges}>Save</button>
      </div>
  );
}
export default Profile;
```

<Prompt>

Used to prompt the user before navigating away from a page.

When your application enters a state that should prevent the user from navigating away (like a form is half-filled out), render a <Prompt>.

Handling 404 error page or Page Not Found.

hello-app\src\components\PageNotFound.js

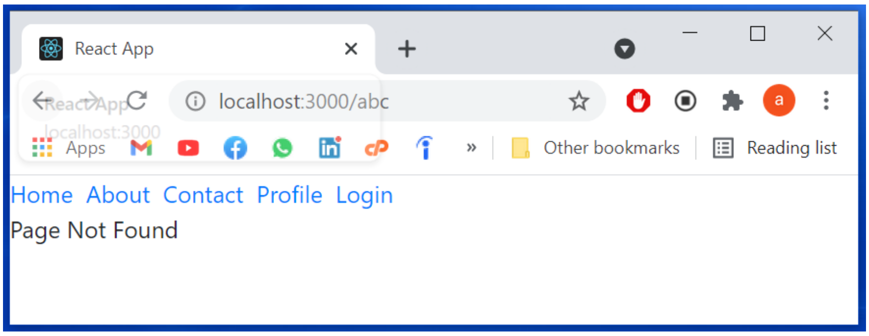
```
function PageNotFound() {
    return <p>Page Not Found</p>;
}
export default PageNotFound
```

hello-app\src\App.js

```
import "./App.css";
import { BrowserRouter, Link, NavLink, Route, Switch } from "react-router-dom";
import Home from "./components/Home";
import About from "./components/About";
import Contact from "./components/Contact";
import Profile from "./components/Profile";
import LogIn from "./components/Login"
import SecuredRoute from "./components/SecuredRoute"
import PageNotFound from "./components/PageNotFound";

function App() {
  return (
    <BrowserRouter>
      <div className="customClass">
        <Link to="/" >Home</Link>&nbsp;&nbsp;
        <NavLink to="/about" >About</NavLink>&nbsp;&nbsp;
        <NavLink to="/contact">Contact</NavLink>&nbsp;&nbsp;
        <NavLink to="/profile">Profile</NavLink>&nbsp;&nbsp;
        <NavLink to="/login">Login</NavLink>

        <Switch>
          <Route exact path="/" component={Home}></Route>
          <Route exact path="/about" component={About}></Route>
          <Route exact path="/contact" component={Contact}></Route>
          <SecuredRoute path="/profile" component={Profile}></SecuredRoute>
          <Route exact path="/login" component={LogIn}></Route>
          <Route path="*" component={PageNotFound}></Route>
        </Switch>
      </div>
    </BrowserRouter>
  );
}
export default App;
```

Test the fallowing url

http://localhost:3000/abc



Home About Contact Profile Login

Page Not Found

Route Configurations

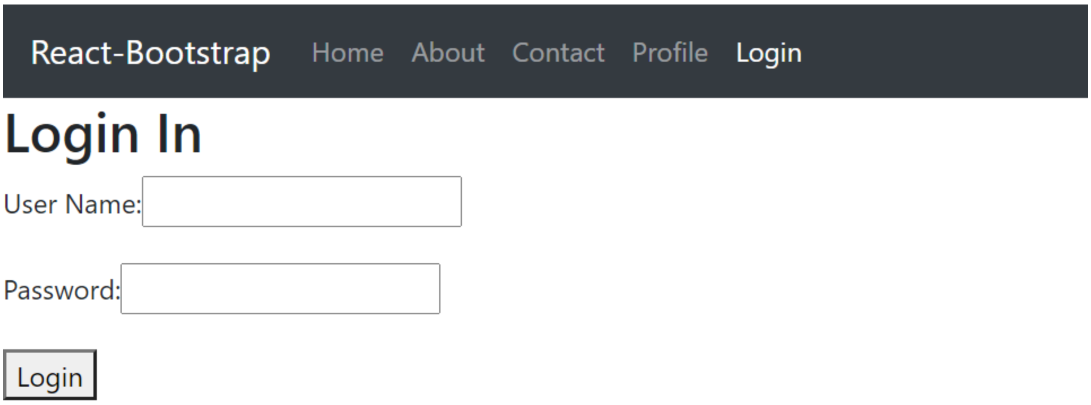
hello-app\src\configs\routes.js

```
import Home from "../components/Home";
import About from "../components/About";
import Contact from "../components/Contact";
import Profile from "../components/Profile";
import LogIn from "../components/Login";
import PageNotFound from "../components/PageNotFound";
import { BrowserRouter, NavLink, Route, Switch } from "react-router-dom";

const routes = [
    {
        path: "/home",
        component: Home,
    },
    {
        path: "/about",
        component: About,
    },
    {
        path: "/contact",
        component: Contact,
    },
    {
        path: "/profile",
        component: Profile,
    },
    {
        path: "/login",
        component: LogIn,
    },
    {
        path: "*",
        component: PageNotFound,
    },
];

const CreateRoutes = () => {
    return (
        <BrowserRouter>
            <NavLink to="/home">Home</NavLink>&nbsp;&nbsp;
            <NavLink to="/about">About</NavLink>&nbsp;&nbsp;
            <NavLink to="/contact">Contact</NavLink>&nbsp;&nbsp;
            <NavLink to="/profile">Profile</NavLink>&nbsp;&nbsp;
            <NavLink to="/login">Login</NavLink>
            <Switch>
                {routes.map((route) => (
                    <Route
                        key={route.path}
                        path={route.path}
                        component={route.component}
                    />
                ))}
            </Switch>
        </BrowserRouter>
    );
};
export default CreateRoutes;
```

hello-app\src\App.js

```
import "./App.css";
import CreateRoutes from "./configs/routes";

function App() {
  return (
      <CreateRoutes></CreateRoutes>
    );
}
export default App;
```

Adding Bootstrap Navbar to routes



Login In

User Name:

Password:

Install the fallowing package;

hello-app>npm install react-bootstrap bootstrap

hello-app>npm install react-router-bootstrap --save

hello-app\src\configs\routes.js

```
import Home from "../components/Home";
import About from "../components/About";
import Contact from "../components/Contact";
import Profile from "../components/Profile";
import LogIn from "../components/Login";
import PageNotFound from "../components/PageNotFound";
import { BrowserRouter, Route, Switch } from "react-router-dom";
import { Navbar, Nav } from "react-bootstrap";
import { LinkContainer } from "react-router-bootstrap";

import "bootstrap/dist/css/bootstrap.min.css";

const routes = [
  {
    path: "/home",
    component: Home,
  },
  {
    path: "/about",
    component: About,
  },
  {
    path: "/contact",
    component: Contact,
  },
  {
    path: "/profile",
    component: Profile,
  },
  {
    path: "/login",
    component: LogIn,
  },
  {
    path: "*",
    component: PageNotFound,
  },
];

const CreateRoutes = () => {
  return (
    <BrowserRouter>
      <Navbar bg="dark" variant="dark" expand="lg">
        <LinkContainer to="/home">
          <Navbar.Brand>React-Bootstrap</Navbar.Brand>
        </LinkContainer>
        <Navbar.Toggle aria-controls="basic-navbar-nav" />
        <Navbar.Collapse id="basic-navbar-nav">
          <Nav className="mr-auto">
            <LinkContainer to="/home">
              <Nav.Link>Home</Nav.Link>
            </LinkContainer>
            <LinkContainer to="/about">
              <Nav.Link>About</Nav.Link>
            </LinkContainer>
            <LinkContainer to="/contact">
              <Nav.Link>Contact</Nav.Link>
            </LinkContainer>
            <LinkContainer to="/profile">
              <Nav.Link>Profile</Nav.Link>
            </LinkContainer>
            <LinkContainer to="/login">
              <Nav.Link>Login</Nav.Link>
            </LinkContainer>
          </Nav>
        </Navbar.Collapse>
      </Navbar>

      <Switch>
        {routes.map((route) => (
          <Route
            key={route.path}
            path={route.path}
            component={route.component}
          />
        ))}
      </Switch>
    </BrowserRouter>
  );
};
export default CreateRoutes;
```