

Exploring Authentication Flow with React Router

Objective:

Create a React application with an authentication flow comprising login and dashboard pages. Utilize React Router for routing, `useNavigate` for programmatic navigation, and `Navigate` for conditional navigation based on authentication status. Additionally, include a home page to provide a comprehensive user experience.

Requirements:

Setup:

Set up a new React application with React Router.

Create separate components for the home page, login page, and dashboard page.

Implement routing using `BrowserRouter`, `Route`, and `Routes`.

Home Page (Home.js):

Render a welcome message or introductory content.

Include a link to navigate to other pages.

Login Page (Login.js):

Create a login form with fields for username and password.

Implement a login button to simulate the authentication process.

Utilize the `useNavigate` hook to navigate to the dashboard page upon successful login.

Dashboard Page (Dashboard.js):

Display the dashboard content, such as user-specific data or features.

Implement conditional navigation using the `Navigate` component:

If the user is not authenticated, redirect them to the login page.

If the user is authenticated, allow access to the dashboard content.

Solution –

App.js

```
import React, { useState } from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import Home from "./Home";
import Login from "./Login";
import Dashboard from "./Dashboard"
import "./App.css";
```

```
function App() {
  const [isAuth, setIsAuth] = useState(false); // Track authentication state

  return (
    <Router>
      <div className="app">
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/login" element={<Login setAuth={setIsAuth} />} />
          <Route path="/dashboard" element={<Dashboard isAuth={isAuth} />} />
        </Routes>
      </div>
    </Router>
  );
}

export default App;
```

```
JS App.js basic-routing-app\... # App.css JS App.js auth-flow-app\... U X # index.css U JS Home.js U JS Login.js U JS Dashboard.js U
auth-flow-app > src > JS App.js > App
1 import React, { useState } from "react";
2 import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
3 import Home from "../Home";
4 import Login from "../Login";
5 import Dashboard from "../Dashboard";
6 import "../App.css";
7
8 function App() {
9   const [isAuth, setIsAuth] = useState(false); // Track authentication state
10
11   return (
12     <Router>
13       <div className="app">
14         <Routes>
15           <Route path="/" element={<Home />} />
16           <Route path="/login" element={<Login setAuth={setIsAuth} />} />
17           <Route path="/dashboard" element={<Dashboard isAuth={isAuth} />} />
18         </Routes>
19       </div>
20     </Router>
21   );
22 }
23
24 export default App;
25
```

Index.css

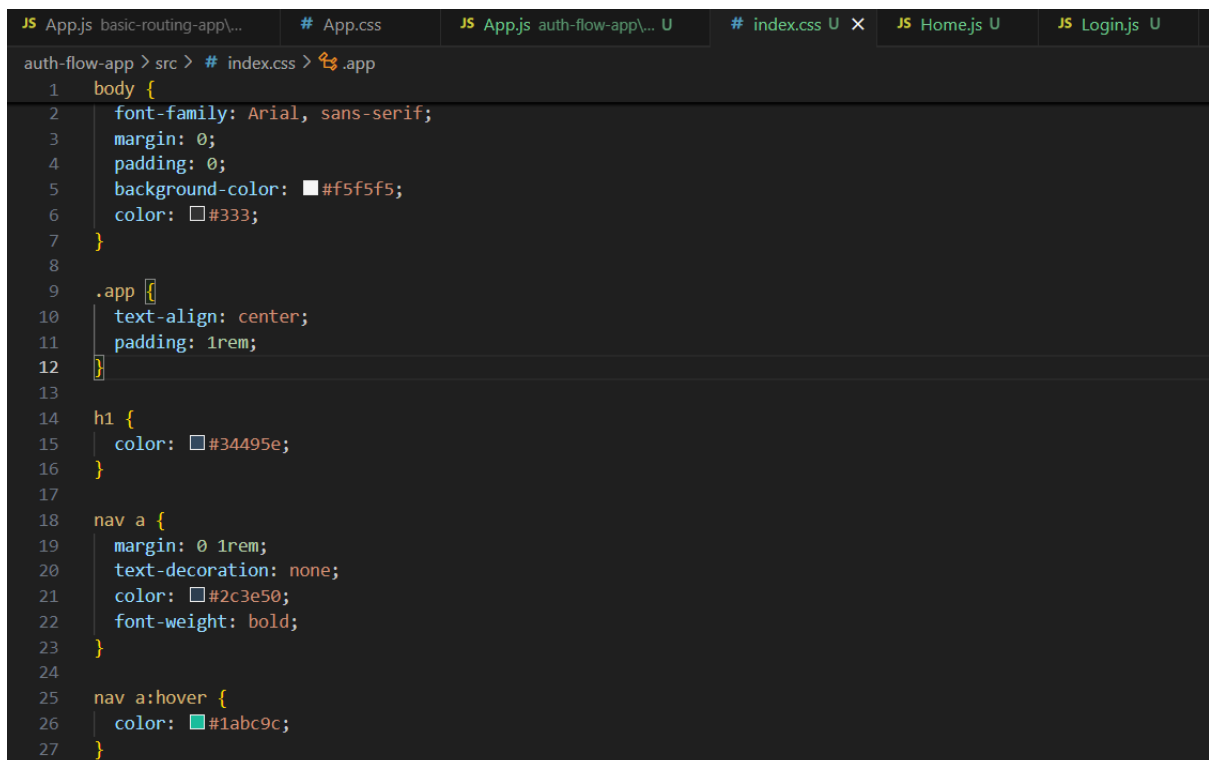
```
body {
  font-family: Arial, sans-serif;
  margin: 0;
  padding: 0;
  background-color: #f5f5f5;
  color: #333;
}

.app {
  text-align: center;
  padding: 1rem;
}

h1 {
  color: #34495e;
}

nav a {
  margin: 0 1rem;
  text-decoration: none;
```

```
    color: #2c3e50;
    font-weight: bold;
}
nav a:hover {
    color: #1abc9c;
}
form {
    margin-top: 2rem;
}
form div {
    margin-bottom: 1rem;
}
```

A screenshot of a code editor with a dark theme. The editor has several tabs at the top: 'JS App.js basic-routing-app\...', '# App.css', 'JS App.js auth-flow-app\... U', '# index.css U' (which is the active tab), 'JS Home.js U', and 'JS Login.js U'. The active tab shows the following CSS code:

```
1  body {
2    font-family: Arial, sans-serif;
3    margin: 0;
4    padding: 0;
5    background-color: #f5f5f5;
6    color: #333;
7  }
8
9  .app {
10   text-align: center;
11   padding: 1rem;
12 }
13
14 h1 {
15   color: #34495e;
16 }
17
18 nav a {
19   margin: 0 1rem;
20   text-decoration: none;
21   color: #2c3e50;
22   font-weight: bold;
23 }
24
25 nav a:hover {
26   color: #1abc9c;
27 }
```

```
25   nav a:hover {
26     | color: #1abc9c;
27   }
28
29   form {
30     | margin-top: 2rem;
31   }
32
33   form div {
34     | margin-bottom: 1rem;
35   }
36
```

Home.js

```
import React from "react";
```

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

```
function Home() {
```

```
  return (
```

```
    <div>
```

```
      <h1>Welcome to the Home Page</h1>
```

```
      <p>Explore the app using the navigation links below:</p>
```

```
      <nav>
```

```
        <Link to="/login">Login</Link> | <Link
to="/dashboard">Dashboard</Link>
```

```
      </nav>
```

```
    </div>
```

```
  );
```

```
}
```

```
export default Home;
```

```
JS App.js basic-routing-app\... # App.css JS App.js auth-flow-app\... U # index.css U JS Home.js U X JS Login.js U
auth-flow-app > src > JS Home.js > ...
1 import React from "react";
2 import { Link } from "react-router-dom";
3
4 function Home() {
5   return (
6     <div>
7       <h1>Welcome to the Home Page</h1>
8       <p>Explore the app using the navigation links below:</p>
9       <nav>
10        <Link to="/login">Login</Link> | <Link to="/dashboard">Dashboard</Link>
11      </nav>
12    </div>
13  );
14 }
15
16 export default Home;
17
```

Login.js

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

```
function Login({ setAuth }) {
  const [username, setUsername] = useState("");
  const [password, setPassword] = useState("");
  const navigate = useNavigate();

  const handleLogin = () => {
    // Simulate authentication logic
    if (username && password) {
      setAuth(true); // Set the user as authenticated
      navigate("/dashboard"); // Navigate to the dashboard
    } else {
      alert("Please enter valid credentials.");
    }
  };
}
```

```
return (  
  <div>  
    <h1>Login Page</h1>  
    <form>  
      <div>  
        <label>Username:</label>  
        <input  
          type="text"  
          value={username}  
          onChange={(e) => setUsername(e.target.value)}  
        />  
      </div>  
      <div>  
        <label>Password:</label>  
        <input  
          type="password"  
          value={password}  
          onChange={(e) => setPassword(e.target.value)}  
        />  
      </div>  
      <button type="button" onClick={handleLogin}>  
        Login  
      </button>  
    </form>  
  </div>  
}
```

export default Login;

```
auth-flow-app > src > JS Login.js > Login
1  import React, { useState } from "react";
2  import { useNavigate } from "react-router-dom";
3
4  function Login({ setAuth }) {
5    const [username, setUsername] = useState("");
6    const [password, setPassword] = useState("");
7    const navigate = useNavigate();
8
9    const handleLogin = () => {
10     // Simulate authentication logic
11     if (username && password) {
12       setAuth(true); // Set the user as authenticated
13       navigate("/dashboard"); // Navigate to the dashboard
14     } else {
15       alert("Please enter valid credentials.");
16     }
17   };
18
19   return (
20     <div>
21       <h1>Login Page</h1>
22       <form>
23         <div>
24           <label>Username:</label>
25           <input
26             type="text"
27             value={username}
28             onChange={e => setUsername(e.target.value)}

```

```
28             onChange={e => setUsername(e.target.value)}
29           />
30         </div>
31         <div>
32           <label>Password:</label>
33           <input
34             type="password"
35             value={password}
36             onChange={e => setPassword(e.target.value)}
37           />
38         </div>
39         <button type="button" onClick={handleLogin}>
40           Login
41         </button>
42       </form>
43     </div>
44   );
45 }
46
47 export default Login;
48
```


Dashboard.js

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

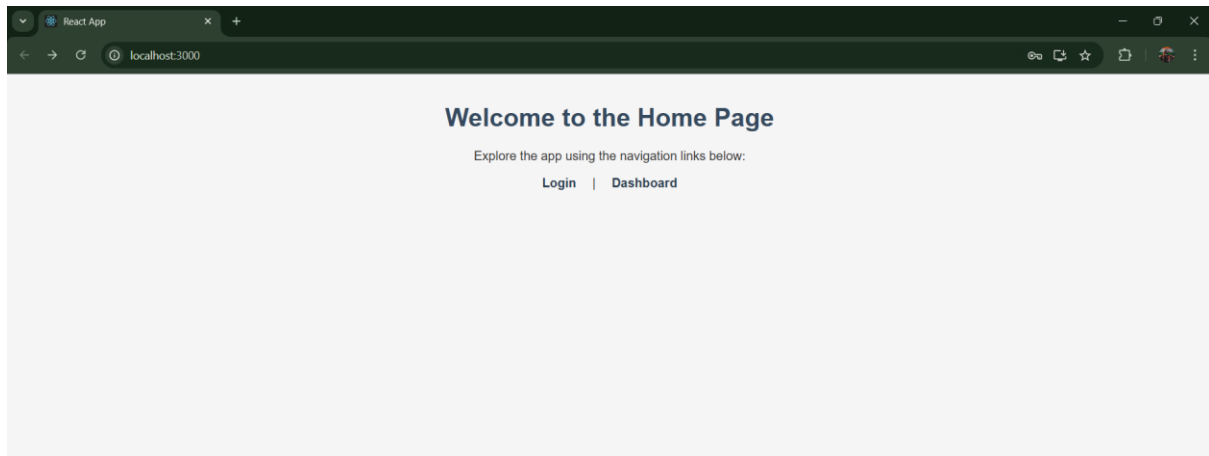
function Dashboard({ isAuth }) {
  if (!isAuth) {
    return <Navigate to="/login" replace />;
  }

  return (
    <div>
      <h1>Dashboard</h1>
      <p>Welcome to your dashboard!</p>
    </div>
  );
}

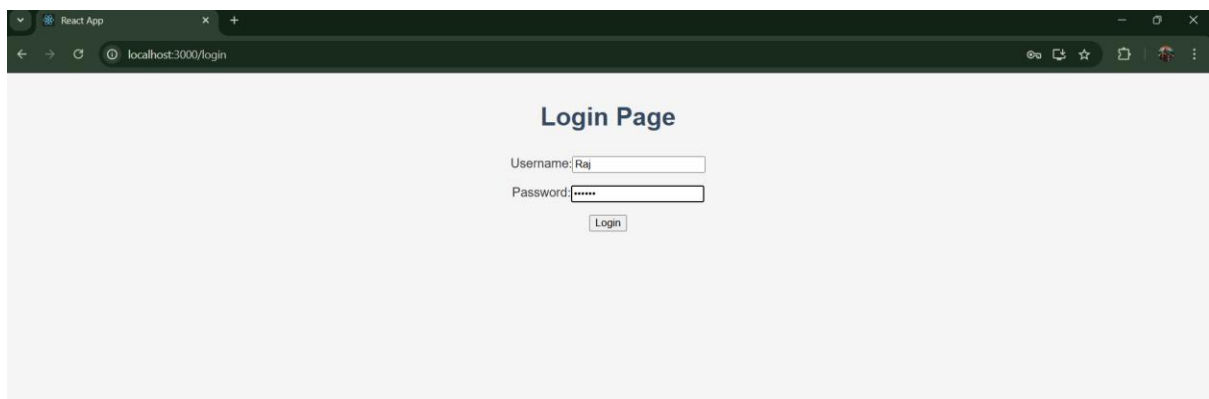
export default Dashboard;
```

```
auth-flow-app > src > JS Dashboard.js > ...
1  import React from "react";
2  import { Navigate } from "react-router-dom";
3
4  function Dashboard({ isAuth }) {
5    if (!isAuth) {
6      return <Navigate to="/login" replace />;
7    }
8
9    return (
10     <div>
11       <h1>Dashboard</h1>
12       <p>Welcome to your dashboard!</p>
13     </div>
14   );
15 }
16
17 export default Dashboard;
18
```

OUTPUT-



After Clicking on the Login Button



After Clicking on the Dashboard Button

