

# React Role-Based Access Control (RBAC) with JWT - Frontend Example

## Project Structure

```
react-rbac/
  └── src/
    ├── App.js
    ├── Login.js
    ├── Dashboard.js
    ├── ProtectedRoute.js
    └── index.js
  └── package.json
```

## Install Dependencies

```
npx create-react-app react-rbac
cd react-rbac
npm install axios react-router-dom
```

## App.js

```
import React, { useState, useEffect } from "react";
import { BrowserRouter as Router, Route, Routes, Navigate } from "react-router-dom";
import Login from "./Login";
import Dashboard from "./Dashboard";
import ProtectedRoute from "./ProtectedRoute";

function App() {
  const [token, setToken] = useState(localStorage.getItem("token") || null);

  useEffect(() => {
    if (token) localStorage.setItem("token", token);
    else localStorage.removeItem("token");
  }, [token]);

  return (
    <Router>
      <Routes>
        <Route path="/login" element={<Login setToken={setToken} />} />
        <Route
          path="/dashboard"
          element={
            <ProtectedRoute token={token}>
              <Dashboard token={token} />
            </ProtectedRoute>
          }
        />
        <Route path="*" element={<Navigate to="/login" />} />
      </Routes>
    </Router>
  );
}

export default App;
```

## ProtectedRoute.js

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

const ProtectedRoute = ({ token, children }) => {
  if (!token) {
```

```

        return <Navigate to="/login" replace />;
    }
    return children;
};

export default ProtectedRoute;

```

## Login.js

```

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

function Login({ setToken }) {
    const [formData, setFormData] = useState({ username: "", password: "" });
    const [error, setError] = useState("");
    const navigate = useNavigate();

    const handleChange = (e) => {
        setFormData({ ...formData, [e.target.name]: e.target.value });
    };

    const handleSubmit = async (e) => {
        e.preventDefault();
        setError("");
        try {
            const res = await axios.post("http://localhost:3000/login", formData);
            setToken(res.data.token);
            navigate("/dashboard");
        } catch {
            setError("Invalid username or password");
        }
    };

    return (
        <div style={{ maxWidth: "300px", margin: "80px auto", textAlign: "center" }}>
            <h2>Login</h2>
            <form onSubmit={handleSubmit}>
                <input
                    name="username"
                    placeholder="Username"
                    value={formData.username}
                    onChange={handleChange}
                    required
                /><br/><br/>
                <input
                    name="password"
                    type="password"
                    placeholder="Password"
                    value={formData.password}
                    onChange={handleChange}
                    required
                /><br/><br/>
                <button type="submit">Login</button>
            </form>
            {error && <p style={{ color: "red" }}>{error}</p>}
        </div>
    );
}

export default Login;

```

## Dashboard.js

```

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

function Dashboard({ token }) {
    const [profile, setProfile] = useState({});
    const navigate = useNavigate();

    useEffect(() => {

```

```
const fetchProfile = async () => {
  try {
    const res = await axios.get("http://localhost:3000/profile", {
      headers: { Authorization: `Bearer ${token}` },
    });
    setProfile(res.data);
  } catch {
    alert("Session expired. Please login again.");
    navigate("/login");
  }
};

fetchProfile();
, [token, navigate]);

const handleLogout = () => {
  localStorage.removeItem("token");
  navigate("/login");
};

return (
  <div style={{ textAlign: "center", marginTop: "50px" }}>
    <h2>Dashboard</h2>
    <p>Welcome, <b>{profile.message}</b></p>
    <p>Role: <b>{profile.role}</b></p>
    <button onClick={handleLogout}>Logout</button>
  </div>
);
}

export default Dashboard;
```