```
// A component that renders a login form
import { useState } from "react";
import axios from "axios";

const LoginForm = ({ setLogin }) => {
  // State variables for email and password inputs
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  // State variable for error message
  const [error, setError] = useState("");

  // Handler for submitting the form
  const handleSubmit = (e) => {
    e.preventDefault();
    // Check if the email and password match the fixed values
    if (email === "admin@gmail.com" && password === "admin") {
      // Send a post request to the backend with the login state
      axios({
        method: "post",
        url: "http://localhost:8000/admin",
        data: {
          logged: true,
        },
      })
        .then(({ data }) => {
          // If the response is successful, update the login state in the app component
          if (data.success) {
            setLogin(true);
          } else {
            // If the response is not successful, show the error message
            setError(data.message);
          }
        })
        .catch((error) => {
          // If there is an error in the request, show the error message
          setError(error.message);
        });
    } else {
      // If the email and password do not match, show an error message
      setError("Invalid email or password");
    }
  };

  return (
    <div className="login-form">
      <h1>Login</h1>
      <form onSubmit={handleSubmit}>
        <label htmlFor="email">Email:</label>
        <input
          type="email"
          id="email"
          value={email}
          onChange={(e) => setEmail(e.target.value)}
          required
        />
        <label htmlFor="password">Password:</label>
        <input
          type="password"
          id="password"
          value={password}
          onChange={(e) => setPassword(e.target.value)}
          required
        />
        <button type="submit">Login</button>
      </form>
      {error && <p className="error">{error}</p>}
    </div>
  );
};

export default LoginForm;

// A component that renders the admin page
const Admin = () => {
  return (
    <div className="admin-page">
      <h1>Welcome to the admin page</h1>
      <p>This page is only accessible to authorized users</p>
    </div>
  );
};

export default Admin;

// A component that renders the home page
const Home = () => {
  return (
    <div className="home-page">
      <h1>Welcome to the home page</h1>
      <p>This page is accessible to everyone</p>
    </div>
  );
};
```

```
export default Home;

// A component that renders the main app with routing and navigation
import { BrowserRouter as Router, Routes, Route, Link } from "react-router-dom";
import Home from "./pages/Home";
import Admin from "./pages/Admin";
import PrivateRoute from "./PrivateRoute";
import { useEffect, useState } from "react";
import LoginForm from "./LoginForm";

const App = () => {
  // useState hook to keep track of the login state
  const [login, setLogin] = useState(() => {
    // Used local storage to sustain the login state
    if (!localStorage.getItem("login")) {
      localStorage.setItem("login", "false");
      return false;
    }
    return JSON.parse(localStorage.getItem("login"));
  });

  // Updating the local storage whenever
  // the login state changes
  useEffect(() => {
    localStorage.setItem("login", JSON.stringify(login));
  }, [login]);

  // Click Handler updates the login state
  // when the button is clicked
  const click = () => {
    setLogin((prev) => {
      return !prev;
    });
  };

  return (
    <div className="App">
      <Router>
        <button onClick={() => click()}>
          {login ? "Logout" : "Login"}
        </button>
        <Link to={"/admin"}>Admin</Link>
        <Link to={"/"}>Home</Link>
        <Routes>
          <Route path="/" element={<Home />} />
          {/* Protecting the Admin Page */}
          <Route
            path="/admin"
            element={
              <PrivateRoute login={login}>
                <Admin />
              </PrivateRoute>
            }
          />
          {/* Adding a Login Page */}
          <Route path="/login" element={<LoginForm setLogin={setLogin} />} />
        </Routes>
      </Router>
    </div>
  );
};

export default App;

// A server that handles authentication using Express
const express = require("express");
const cors = require("cors");

// Creating an express server
const app = express();

// Middleware to parse the data into JSON
app.use(express.json());

// Middleware to accept requests from localhost:3000
app.use(
  cors({
    origin: "http://localhost:3000",
    credentials: true,
  })
);

// Middleware Function to authenticate the user
const auth = (req, res, next) => {
  console.log(req.body);
  if (req.body.logged) {
    next();
    return;
  }
  res.send({
    success: false,
    message: "Unauthorized Access",
  });
```

```
};

// Post request handler for the /admin route
app.post("/admin", auth, (req, res) => {
  res.send({
    success: true,
    message: "Successfully Authenticated",
  });
});

app.listen(8000, () => {
  console.log("Listening on port 8000");
});
```