React Form Handling + Validation + LocalStorage + Login/Register

==========

1. FORM HANDLING (Single State Object)

- Use one object to store all input fields.
- Use dynamic key updating: [e.target.name]: e.target.value
- Spread operator (...) keeps old values.

Example:

setForm({ ...form, [e.target.name]: e.target.value });

Why?

- Prevents data loss.
- Ensures controlled components.
- React re-renders properly.

2. FORM VALIDATION (Frontend)

Basic checks before submitting:

- Check empty fields.
- Check email format.
- Check password length.
- Show error messages.

Example:

if (!form.email.includes("@")) { alert("Invalid Email"); }

3. STORING DATA LOCALLY (localStorage)

- Save data permanently in the browser.
- JSON.stringify() when saving.
- JSON.parse() when reading.

Example:

localStorage.setItem("users", JSON.stringify(array));

const users = JSON.parse(localStorage.getItem("users"));

4. REGISTER PROCESS

- Take data from form.
- Validate fields.
- Get old users from localStorage.

- Push new user.
- Save again.
- Redirect to login.

5. LOGIN PROCESS

- User enters email/password.
- Fetch user array from localStorage.
- Compare login credentials.
- If match → save 'currentUser'
- Redirect to dashboard.

6. DASHBOARD PROTECTION

- Check currentUser exists.
- If not → redirect to login.

7. INTERVIEW QUESTIONS

---

1. What are controlled components?
2. Why is React state used for input fields?
3. Why is spread operator needed in form handling?
4. Difference between controlled and uncontrolled inputs.
5. How localStorage works?
6. How to implement login without backend?
7. What is dynamic key in JavaScript?
8. How does useEffect help in loading stored data?
9. Explain why React needs immutable state.
10. Why JSON.stringify and JSON.parse are required?

8. FULL PROJECT CODE

--- App.jsx ---

```
import { BrowserRouter, Routes, Route } from "react-router-dom";
import Register from "./Register";
import Login from "./Login";
import Dashboard from "./Dashboard";

function App() {
return (
```

```
} />
} />
} />

);
}
export default App;

--- Register.jsx ---
import { useState } from "react";
import { useNavigate } from "react-router-dom";

function Register() {
const navigate = useNavigate();
const [form, setForm] = useState({ name:"", email:"", password:"" });

function handleChange(e) {
setForm({ ...form, [e.target.name]: e.target.value });
}

function handleSubmit(e) {
e.preventDefault();
const oldData = JSON.parse(localStorage.getItem("users")) || [];
oldData.push(form);
localStorage.setItem("users", JSON.stringify(oldData));
navigate("/login");
}

return (...)
}
export default Register;

--- Login.jsx ---
import { useState } from "react";
import { useNavigate } from "react-router-dom";

function Login() {
const navigate = useNavigate();
const [email, setEmail] = useState("");
const [password, setPassword] = useState("");

function handleLogin(e) {
e.preventDefault();
const users = JSON.parse(localStorage.getItem("users")) || [];
```

```
const found = users.find(u => u.email === email && u.password === password);

if (found) {
localStorage.setItem("currentUser", JSON.stringify(found));
navigate("/dashboard");
} else {
alert("Invalid credentials");
}
}

return (...)
}
export default Login;

--- Dashboard.jsx ---
import { useEffect, useState } from "react";
import { useNavigate } from "react-router-dom";

function Dashboard() {
const navigate = useNavigate();
const [user, setUser] = useState(null);

useEffect(() => {
const data = localStorage.getItem("currentUser");
if (!data) navigate("/login");
else setUser(JSON.parse(data));
}, []);

function logout() {
localStorage.removeItem("currentUser");
navigate("/login");
}

return (...)
}
export default Dashboard;
```