

## ✓ Day 24 – Authentication Basics (Modern 2025)

### 🚫 What is Authentication?

Authentication means verifying **who the user is**.

Example: *Login system* → user enters email/password → we verify identity.

### ✓ Why Authentication?

- Secure user data
- Personalization (dashboards, profiles)
- Restrict access to certain pages
- Real-world apps cannot exist without login

### ✓ Authentication vs Authorization

| Concept        | Meaning          | Example              |
|----------------|------------------|----------------------|
| Authentication | Who are you?     | Login                |
| Authorization  | What can you do? | Admin vs Normal user |

### ✓ Modern Authentication Methods (2025)

| Method                            | Used For               | Secure | Notes                     |
|-----------------------------------|------------------------|--------|---------------------------|
| JWT + HttpOnly Cookies            | Full-stack apps        | ✓✓✓    | Recommended best practice |
| Auth services like Firebase/Auth0 | Fast login integration | ✓✓✓    | No backend required       |
| Tokens in LocalStorage            | Dev/testing            | ✗      | Vulnerable to XSS         |

✓ We start with **Mock Auth + Context + Protected Routes** → THEN real backend.

### ✓ Authentication Flow in React

1 User enters login info 2 Validate → authenticate 3 Store user data in **Context** (global state) 4 Protect private pages → redirect if not logged in

---

## Tools We Use

✓ React Router → protected navigation   ✓ Context API → global auth state   ✓ useNavigate() → redirect after login

---

### ✓ AuthContext Structure

- Holds user data
  - Login function
  - Logout function
  - Shared across app
- 

### ✓ Protected Routes (PrivateRoute Component)

Block pages unless user logged in.

Example pages protected: ✓ Dashboard   ✓ Profile   ✓ Settings

---

### ✓ Project Structure

```
src/
├── AuthContext/
│   └── AuthProvider.jsx ✓
├── Pages/
│   ├── Login.jsx
│   ├── Dashboard.jsx
│   └── Profile.jsx
└── PrivateRoute.jsx ✓
```

---

## ✓ Day 24 – Mini Project Requirements

### 🔒 Secure Dashboard App

✓ Login Page with validation   ✓ Dashboard (ONLY if logged-in)   ✓ Logout button   ✓ Save user in context   ✓ useNavigate for redirection   ✓ Tailwind/MUI styling recommended

**Credentials for mock test:**

email: admin@gmail.com  
password: 123456

---

### **Bonus Challenges**

☒ Add Profile Page ☒ Show logged-in user's email ☒ Redirect user to login after logout ☒ Prevent back-button access after logout

---

### **What You Will Learn**

- Real authentication architecture
  - Secure route protection
  - Global state for auth
  - Redirect behavior in React
- 

 After you finish ☒ Share your code → I will review like a senior dev 