

Training Day 16 Report

Date: 14 July 2025

Project: MERN Notes App

Topic Covered: Frontend Authentication (React Login & Signup)

On the sixteenth day, we connected the frontend React application to the backend authentication system. Users can now register, log in, and persist their session with JWT tokens.

Concepts Learned

- Creating authentication forms (Signup and Login) with controlled inputs.
- Sending API requests from React using `fetch` or `axios`.
- Storing JWT tokens securely in `localStorage`.
- Updating global state to reflect whether a user is logged in.

Signup Component

Listing 1: Signup form in React

```
1 import React, { useState } from 'react';
2 import axios from 'axios';
3
4 function Signup() {
5   const [formData, setFormData] = useState({ name: '', email: '',
6     password: '' });
7
8   const handleChange = (e) => {
9     setFormData({ ...formData, [e.target.name]: e.target.value });
10   };
11
12   const handleSubmit = async (e) => {
13     e.preventDefault();
14     const res = await axios.post('/api/users/signup', formData);
15     localStorage.setItem('userInfo', JSON.stringify(res.data));
16     alert('Signup successful');
17   };
18
19   return (
20     <form onSubmit={handleSubmit}>
21       <input name="name" placeholder="Name" onChange={handleChange} />
22       <input name="email" placeholder="Email" onChange={handleChange}
23         />
24       <input type="password" name="password" placeholder="Password"
25         onChange={handleChange} />
26       <button type="submit">Sign Up</button>
27     </form>
```

```

25     );
26 }
27
28 export default Signup;

```

Login Component

Listing 2: Login form in React

```

1  import React, { useState } from 'react';
2  import axios from 'axios';
3
4  function Login() {
5      const [formData, setFormData] = useState({ email: '', password: '' })
6          ;
7
7      const handleChange = (e) => {
8          setFormData({ ...formData, [e.target.name]: e.target.value });
9      };
10
11     const handleSubmit = async (e) => {
12         e.preventDefault();
13         const res = await axios.post('/api/users/login', formData);
14         localStorage.setItem('userInfo', JSON.stringify(res.data));
15         alert('Login successful');
16     };
17
18     return (
19         <form onSubmit={handleSubmit}>
20             <input name="email" placeholder="Email" onChange={handleChange}
21                 />
22             <input type="password" name="password" placeholder="Password"
23                 onChange={handleChange} />
24             <button type="submit">Login</button>
25         </form>
26     );
27 }
28
29 export default Login;

```

Persisting User State

- After signup/login, the JWT token is stored in `localStorage`.
- On app load, we check for `userInfo` in storage and update the state.
- The token will later be included in headers for protected API requests.

Testing the Authentication Flow

- Registered a new user through the Signup form → verified user was added in the database.
- Logged in with correct credentials → token returned and stored in `localStorage`.
- Attempted login with incorrect credentials → received 401 `Unauthorized`.

Outcome of the Day

- Successfully built Signup and Login forms in React.
- Connected frontend with backend authentication APIs.
- Stored tokens in `localStorage` for persistent login sessions.