

Training Day 23 Report

Date: 22 July 2025

Project: MERN Notes App

Topic Covered: Authentication UI/UX Enhancements

On the twenty-third day, we improved the Login and Signup forms by adding validation, error handling, and visual feedback. This made authentication more user-friendly and secure.

Concepts Learned

- Form validation in React (required fields, password length, email format).
- Displaying error messages from backend (e.g., wrong credentials).
- Adding loading states for better user experience.

Signup Form with Validation

Listing 1: Signup.js with validation

```
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   const [error, setError] = useState('');
8   const [loading, setLoading] = useState(false);
9
10  const handleChange = (e) => {
11    setFormData({ ...formData, [e.target.name]: e.target.value });
12  };
13
14  const handleSubmit = async (e) => {
15    e.preventDefault();
16    if (!formData.name || !formData.email || formData.password.length <
17      6) {
18      return setError('All fields are required and password must be 6+
19        chars');
20    }
21    try {
22      setLoading(true);
23      const res = await axios.post('/api/users/signup', formData);
24      localStorage.setItem('userInfo', JSON.stringify(res.data));
25      setLoading(false);
26    } catch (err) {
27      setLoading(false);
28      setError(err.response?.data?.message || 'Signup failed');
29    }
30  }
31 }
```

```

27   };
28
29   return (
30     <form onSubmit={handleSubmit}>
31       {error && <p style={{ color: 'red' }}>{error}</p>}
32       <input name="name" placeholder="Name" onChange={handleChange} />
33       <input name="email" placeholder="Email" onChange={handleChange}
34         />
35       <input type="password" name="password" placeholder="Password"
36         onChange={handleChange} />
37       <button type="submit" disabled={loading}>
38         {loading ? 'Signing up...' : 'Sign Up'}
39       </button>
40     </form>
41   );
42 }
43
44 export default Signup;

```

Login Form Improvements

Listing 2: Login.js with error handling

```

1 function Login() {
2   const [formData, setFormData] = useState({ email: '', password: '' })
3   ;
4   const [error, setError] = useState('');
5
6   const handleSubmit = async (e) => {
7     e.preventDefault();
8     try {
9       const res = await axios.post('/api/users/login', formData);
10      localStorage.setItem('userInfo', JSON.stringify(res.data));
11    } catch (err) {
12      setError('Invalid email or password');
13    }
14  };
15
16  return (
17    <form onSubmit={handleSubmit}>
18      {error && <p style={{ color: 'red' }}>{error}</p>}
19      <input name="email" placeholder="Email" onChange={(e) =>
20        setFormData({ ...formData, email: e.target.value })} />
21      <input type="password" name="password" placeholder="Password"
22        onChange={(e) => setFormData({ ...formData, password: e.target
23          .value })} />
24      <button type="submit">Login</button>
25    </form>

```

```
22     );  
23 }
```

Testing the UI/UX

- Tried signing up with a short password → validation error displayed.
- Attempted login with wrong credentials → “Invalid email or password” shown.
- Observed loading state → button disabled while request in progress.

Outcome of the Day

- Added proper validation and error messages to authentication forms.
- Improved user feedback with loading indicators.
- Authentication flow is now polished, responsive, and user-friendly.