

Training Day 18 Report

Date: 16 July 2025

Project: MERN Notes App

Topic Covered: Connecting Frontend CRUD with JWT Authorization

On the eighteenth day, we integrated the frontend Notes components with the backend APIs. All requests now include the JWT token in headers, ensuring only authenticated users can perform CRUD operations on their notes.

Concepts Learned

- Extracting token from `localStorage`.
- Sending token in `Authorization` header with API requests.
- Handling CRUD operations (Create, Read, Update, Delete) from React.

Axios Instance with Token

We created a reusable axios instance in `src/api.js`:

Listing 1: Axios instance with JWT header

```
1 import axios from 'axios';
2
3 const userInfo = JSON.parse(localStorage.getItem('userInfo'));
4 const token = userInfo?.token;
5
6 const api = axios.create({
7   baseURL: '/api',
8   headers: {
9     Authorization: token ? `Bearer ${token}` : ''
10  }
11 });
12
13 export default api;
```

Fetching Notes

In `src/screens/Notes.js`:

Listing 2: Fetch user notes

```
1 import React, { useEffect, useState } from 'react';
2 import api from '../api';
3
4 function Notes() {
5   const [notes, setNotes] = useState([]);
6 }
```

```

7   useEffect(() => {
8     const fetchNotes = async () => {
9       const res = await api.get('/notes');
10      setNotes(res.data);
11    };
12    fetchNotes();
13  }, []);
14
15  return (
16    <div>
17      <h2>Your Notes</h2>
18      <ul>
19        {notes.map(note => (
20          <li key={note._id}>{note.title} - {note.content}</li>
21        ))}
22      </ul>
23    </div>
24  );
25 }
26
27 export default Notes;

```

Creating a Note

Listing 3: Create new note

```

1  const createNote = async () => {
2    const res = await api.post('/notes', {
3      title: 'New Note',
4      content: 'This is a new note'
5    });
6    setNotes([...notes, res.data]);
7  };

```

Updating and Deleting Notes

Listing 4: Update and delete note

```

1  const updateNote = async (id) => {
2    const res = await api.put(`/notes/${id}`, {
3      title: 'Updated Title',
4      content: 'Updated content'
5    });
6    setNotes(notes.map(n => n._id === id ? res.data : n));
7  };
8
9  const deleteNote = async (id) => {
10   await api.delete(`/notes/${id}`);
11   setNotes(notes.filter(n => n._id !== id));

```

Testing the Functionality

- Verified fetching notes only works with a valid token.
- Created a note → immediately reflected in the Notes list.
- Updated a note → changes persisted in database and UI.
- Deleted a note → removed from UI and DB.

Outcome of the Day

- Successfully connected frontend CRUD operations to backend APIs.
- Implemented JWT-based authorization for all requests.
- Application now provides a full, secure note management system.