Training Day 8 Report

Date: 4 July 2025

Topic Covered: Creating the POST Route to Add a Note

On the eighth day of MERN stack training, we implemented the API endpoint for adding new notes to the database. This was the first real CRUD operation for the Notes App backend.

Understanding the POST Method

The **POST** HTTP method is used to send data from the client to the server to create a new resource. In our case:

- The client sends a note's title and content.

- The backend processes the request and stores the note in MongoDB.

Implementing the Route

Inside `backend/routes/noteRoutes.js`:

```
const express = require('express');
const router = express.Router();
const Note = require('../models/Note');

router.post('/', async (req, res) => {
    try {
        const { title, content } = req.body;
        const newNote = new Note({ title, content });
        const savedNote = await newNote.save();
        res.status(201).json(savedNote);
    } catch (error) {
        res.status(500).json({ error: error.message });
    }
});

module.exports = router;
```

Key points:

- Used `async/await` for asynchronous database operations.

- Returned the saved note in the response for confirmation.

Testing the Endpoint

- Used Postman to send a POST request to `http://localhost:5000/api/notes`.

- Sent JSON data with `title` and `content`.

- Verified that the note was saved in MongoDB and returned in the response.

Outcome of the Day

- Learned how to handle POST requests in Express.

- Successfully created and tested the API endpoint for adding notes.

- Confirmed database integration with Mongoose is working as expected.