

Training Day 15 Report

Date: 12 July 2025

Project: MERN Notes App

Topic Covered: Making Notes User-Specific

On the fifteenth day, we modified the Notes model and routes so each note is tied to the user who created it. This ensures privacy and that users only access their own notes.

Updating the Note Model

In `backend/models/noteModel.js`, we added a reference to the user:

Listing 1: Note Schema with user field

```
1 const mongoose = require('mongoose');
2
3 const noteSchema = new mongoose.Schema({
4   title: { type: String, required: true },
5   content: { type: String, required: true },
6   user: {
7     type: mongoose.Schema.Types.ObjectId,
8     ref: 'User',
9     required: true
10  }
11 }, { timestamps: true });
12
13 module.exports = mongoose.model('Note', noteSchema);
```

Creating Notes Linked to User

In the note controller (`backend/controllers/noteController.js`):

Listing 2: Creating a note with user association

```
1 const createNote = async (req, res) => {
2   const { title, content } = req.body;
3
4   try {
5     const note = await Note.create({
6       title,
7       content,
8       user: req.user._id // link note to logged-in user
9     });
10    res.status(201).json(note);
11  } catch (error) {
12    res.status(500).json({ error: error.message });
13  }
14};
```

Fetching Notes for Logged-in User

Listing 3: Fetching notes specific to logged-in user

```
1 const getNotes = async (req, res) => {  
2   try {  
3     const notes = await Note.find({ user: req.user._id });  
4     res.json(notes);  
5   } catch (error) {  
6     res.status(500).json({ error: error.message });  
7   }  
8 };
```

Authorization for Update/Delete

We restricted updates and deletions to only the note owner:

Listing 4: Checking note ownership before update/delete

```
1 if (note.user.toString() !== req.user._id.toString()) {  
2   return res.status(401).json({ message: 'Not authorized' });  
3 }
```

Testing the Functionality

- Logged in with User A and created notes → only User A's notes were visible.
- Logged in with User B → only User B's notes appeared.
- Verified that attempts to update/delete another user's note returned **401 Not authorized**.

Outcome of the Day

- Successfully linked notes to specific users.
- Restricted read, update, and delete operations to the note owner.
- Backend is now fully multi-user secure, with user-specific note handling.