

Training Day 29 Report

Date: 29 July 2025

Project: MERN Notes App

Topic Covered: Final App Enhancements (Refactor & Tags Feature)

On the twenty-ninth day, we focused on final improvements to the app. We refactored the code for clarity and added a new feature: note categories/tags for better organization.

Concepts Learned

- Code refactoring to improve readability and maintainability.
- Adding new schema fields in MongoDB for categories/tags.
- Updating frontend forms and displays to support tags.

Updated Note Model

In backend/models/noteModel.js:

Listing 1: Added tags field to Note schema

```
1 const noteSchema = new mongoose.Schema({
2   title: { type: String, required: true },
3   content: { type: String, required: true },
4   user: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required:
      true },
5   tags: [{ type: String }]
6 }, { timestamps: true });
```

Adding Tags in Create Note Form

In CreateNote.js:

Listing 2: Added tags input

```
1 const [tags, setTags] = useState('');
2
3 const handleSubmit = (e) => {
4   e.preventDefault();
5   onCreate({
6     title,
7     content,
8     tags: tags.split(',').map(tag => tag.trim())
9   });
10  setTitle('');
11  setContent('');
12  setTags('');
13 };
```

```

14
15 return (
16   <form onSubmit={handleSubmit}>
17     <input value={title} onChange={(e) => setTitle(e.target.value)}
18       placeholder="Title" />
19     <textarea value={content} onChange={(e) => setContent(e.target.
20       value)} placeholder="Content" />
21     <input value={tags} onChange={(e) => setTags(e.target.value)}
22       placeholder="Tags (comma separated)" />
23     <button type="submit">Add Note</button>
24   </form>
25 );

```

Displaying Tags in Notes

In `NoteItem.js`:

Listing 3: Show tags with note

```

1 <div className="flex flex-wrap gap-2 mt-2">
2   {note.tags.map((tag, idx) => (
3     <span key={idx} className="px-2 py-1 bg-gray-200 rounded text-sm">
4       #{tag}
5     </span>
6   ))}
7 </div>

```

Code Refactoring

- Moved repeated axios logic into a centralized `api.js`.
- Organized backend routes and controllers into separate files.
- Cleaned unused imports and optimized React components.

Testing Enhancements

- Created notes with tags like “work, urgent” → displayed as tag chips.
- Verified tags persisted in MongoDB and appeared in UI.
- Confirmed app still worked after refactoring.

Outcome of the Day

- Improved code structure with refactoring.
- Successfully added categories/tags for notes.
- Application is feature-complete and user-friendly.