

Training Day 14 Report

Date: 11 July 2025

Project: MERN Notes App

Topic Covered: JWT Middleware for Route Protection

On the fourteenth day, we implemented middleware to protect private routes using JSON Web Tokens (JWT). This ensures that only authenticated users with valid tokens can access note-related endpoints.

Concepts Learned

- Middleware in Express: functions that run between the request and the route handler.
- Extracting and verifying JWTs from request headers.
- Attaching the authenticated user to `req.user` for further use.

Middleware Implementation

In `backend/middleware/authMiddleware.js`:

Listing 1: JWT Authentication Middleware

```
1 const jwt = require('jsonwebtoken');
2 const User = require('../models/userModel');
3
4 const protect = async (req, res, next) => {
5   let token;
6
7   if (req.headers.authorization &&
8       req.headers.authorization.startsWith('Bearer ')) {
9     try {
10      token = req.headers.authorization.split(' ')[1];
11      const decoded = jwt.verify(token, process.env.JWT_SECRET);
12
13      req.user = await User.findById(decoded.id).select('-password');
14      return next();
15    } catch (error) {
16      return res.status(401).json({ message: 'Not authorized, token
17        failed' });
18    }
19  }
20
21  if (!token) {
22    return res.status(401).json({ message: 'Not authorized, no token'
23      });
24  }
25  module.exports = { protect };
```

Applying Middleware to Routes

In backend/routes/noteRoutes.js:

Listing 2: Protecting Note Routes with Middleware

```
1 const { protect } = require('../middleware/authMiddleware');
2
3 router.get('/', protect, getNotes);
4 router.post('/', protect, createNote);
5 router.get('/:id', protect, getNoteById);
6 router.put('/:id', protect, updateNote);
7 router.delete('/:id', protect, deleteNote);
```

Key points:

- Middleware checks for a token in the **Authorization** header.
- Decoded token provides the user ID, which is used to fetch the user from the DB.
- If token is invalid or missing, response is **401 Unauthorized**.

Testing the Middleware

- Sent requests to protected routes without a token → received **401 Unauthorized**.
- Logged in to obtain a token, then retried with **Authorization: Bearer <token>** → request succeeded.
- Verified that the authenticated user's data was available in **req.user**.

Outcome of the Day

- Learned how to create and apply Express middleware.
- Successfully protected all note routes with JWT authentication.
- Backend is now secure and only accessible by authenticated users.