my-express-app/

├── middleware/

│ └── authMiddleware.js

├── routes/

│ └── course.js

├── app.js

└── package.json

1. **authMiddleware.js**

// middleware/authMiddleware.js

const authMiddleware = (req, res, next) => {

  // Check for token in headers

  // const token = req.header('Authorization')?.split(' ')[1];

  const token = true;

  // Define a hardcoded token

  const validToken = "yourHardcodedToken";

  if (!token) {

    return res.status(401).json({ message: "No token, authorization denied" });

  }

  // If token is valid, proceed to the next middleware or route handler

  req.user = { username: "user" }; // Example user data

  next();

};

module.exports = authMiddleware;

1. **course.js**
2. const express = require("express");
3. const router = express.Router();
4. const authMiddleware = require("../middleware/authMiddleware.js");
5. router.get("/", authMiddleware, (req, res) => {
6. res.json({ message: "Welcome to the private course route", user: req.user });
7. });
8. module.exports = router;

**3) app.js**

const express = require("express");

const app = express();

const courseRoute = require("./router/course");

app.use(express.json());

// Define the /course route and apply the middleware

app.use("/course", courseRoute);

const PORT = process.env.PORT || 3000;

app.listen(PORT, () => {

  console.log(`Server running on port ${PORT}`);

});