

NAME	Chirag
UID	23BCS14318
CLASS	622-B

➤ Full Stack PRACTICE 8.3

- CODE

server.js :::

```
const express = require("express");
const jwt = require("jsonwebtoken");
const bodyParser = require("body-parser");

const app = express();
app.use(bodyParser.json());

const SECRET_KEY = "mysecretkey";

// Hardcoded sample users
const users = [
  { id: 1, username: "adminUser", password: "admin123", role: "Admin" },
  { id: 2, username: "normalUser", password: "user123", role: "User" },
  { id: 3, username: "modUser", password: "mod123", role: "Moderator" },
];

// LOGIN ROUTE – Issues token with role
app.post("/login", (req, res) => {
  const { username, password } = req.body;
  const user = users.find(
    (u) => u.username === username && u.password === password
  );
});
```

```

    if (!user) {
      return res.status(401).json({ message: "Invalid credentials" });
    }

    const token = jwt.sign(
      { id: user.id, username: user.username, role: user.role },
      SECRET_KEY,
      { expiresIn: "1h" }
    );

    res.json({ token });
  });

// Middleware – Verify JWT
function verifyToken(req, res, next) {
  const authHeader = req.headers["authorization"];
  if (!authHeader) return res.status(401).json({ message: "Token missing" });

  const token = authHeader.split(" ")[1];

  jwt.verify(token, SECRET_KEY, (err, decoded) => {
    if (err) return res.status(403).json({ message: "Invalid token" });
    req.user = decoded;
    next();
  });
}

// Middleware – Role-based Access Control
function authorizeRoles(...allowedRoles) {
  return (req, res, next) => {
    if (!allowedRoles.includes(req.user.role)) {
      return res
        .status(403)
        .json({ message: "Access denied: insufficient role" });
    }
    next();
  };
}

// ADMIN DASHBOARD (Only Admin)
app.get(
  "/admin-dashboard",
  verifyToken,
  authorizeRoles("Admin"),

```

```

    (req, res) => {
      res.json({
        message: "Welcome to the Admin dashboard",
        user: req.user,
      });
    }
  );

// MODERATOR MANAGEMENT (Only Moderator)
app.get(
  "/moderator-dashboard",
  verifyToken,
  authorizeRoles("Moderator"),
  (req, res) => {
    res.json({
      message: "Welcome to the Moderator dashboard",
      user: req.user,
    });
  }
);

// USER PROFILE (All authenticated roles)
app.get("/user-profile", verifyToken, (req, res) => {
  res.json({
    message: `Welcome to your profile, ${req.user.username}`,
    user: req.user,
  });
});

const PORT = 3000;
app.listen(PORT, () => {
  console.log(`Server running on http://localhost:${PORT}`)
});

```

OUTPUT:

POST http://localhost:3000/login Send

Params	Headers	Auth	Body
1	▼	{	
2		"id": 1,	
3		"username": "adminUser",	
4		"password": "admin123",	
5		"role": "Admin"	
6		}	

Request POST	Response 200
▶ HTTP/1.1 200 OK (6 headers)	
1	▼
2	"token":
	"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6Im5widXNlcm5hbWUiOiJhZG1pb1VzZXIiLCJyb2x1IjoiaWQRtaW4iLCJpYXQiOiJlE3NTI0NzU4MDgsImV4cCI6MTc1MjQ3OTQwOH0.yJ7ZV476tKbzKgmtTxcqUPF66ZMK87PHYhJS09dNys"
3	}

GET http://localhost:3000/admin-dashboard Send

Params Headers **Auth** Body

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJhZG1pb1VzZXIiLCJyb2x1IjoiaW4iLCJpYXQiOiJlbnR5cCI6MTc1MjQ3OTQwOH0.yJ7ZV476tKbzKgmtTxcqUpUF6A7MK87PHYh.IS09dNvs

Request GET Response 200

▶ HTTP/1.1 200 OK (6 headers)

```
1 {
2   "message": "Welcome to the Admin dashboard",
3   "user": {
4     "id": 1,
5     "username": "adminUser",
6     "role": "Admin",
7     "iat": 1752475808,
8     "exp": 1752479408
9   }
10 }
```

Params Headers **Auth** Body

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJhZG1pb1VzZXIiLCJyb2x1IjoiaW4iLCJpYXQiOiJlbnR5cCI6MTc1MjQ3OTQwOH0.yJ7ZV476tKbzKgmtTxcqUpUF6

Request GET Response 403

▶ HTTP/1.1 403 Forbidden (6 headers)

```
1 {
2   "message": "Access denied: insufficient role"
3 }
```

GET http://localhost:3000/user-profile Send

Params Headers **Auth** Body

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm5hbWUiOiJhZG1pb1VzZXIiLCJyb2x1IjoiaW4iLCJpYXQiOiJlbnR5cCI6MTc1MjQ3OTQwOH0.yJ7ZV476tKbzKgmtTxcqUpUF6

Request GET Response 200

▶ HTTP/1.1 200 OK (6 headers)

```
1 {
2   "message": "Welcome to your profile, adminUser",
3   "user": {
4     "id": 1,
5     "username": "adminUser",
6     "role": "Admin",
7     "iat": 1752475808,
8     "exp": 1752479408
9   }
10 }
```