NAME	Chirag
UID	23BCS14318
CLASS	622-B

➢ Full Stack PRACTICE 8.2
- CODE

  server.js :::

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

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

const SECRET_KEY = "mysecretkey";

const user = {
  id: 1,
  username: "testuser",
  password: "password123",
};

//  LOGIN ROUTE — issues a JWT token
app.post("/login", (req, res) => {
  const { username, password } = req.body;

  if (username === user.username && password === user.password) {
    const token = jwt.sign({ id: user.id, username: user.username }, SECRET_KEY,
{
```

```javascript
      expiresIn: "1h",
    });
    res.json({ token });
  } else {
    res.status(401).json({ message: "Invalid credentials" });
  }
});

//  JWT VERIFICATION MIDDLEWARE
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();
  });
}

// PROTECTED ROUTE — accessible only with valid JWT
app.get("/protected", verifyToken, (req, res) => {
  res.json({
    message: "You have accessed a protected route!",
    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": "testuser",
4     "password": "password123"
5 }
```

Request **POST** Response **200**

▶ HTTP/1.1 **200 OK** (6 headers)

```
1 ▼ {
2     "token":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm
    5hbWUiOiJ0ZXN0dXNlciIsImlhdCI6MTc1MjQ3NTMzOSwiZXhwIjoxNz
    UyNDc4OTM5fQ.Bt0iYA9USb9WGc7pRMH6IhQ2NSwIrKYIlE2mrUhOOmE
    "
3 }
```

GET `http://localhost:3000/protected` — Send

Params Headers Auth **Body** `</>`

Request **GET** Response **401**

▶ HTTP/1.1 **401 Unauthorized** (6 headers)

```
1 ▼ {
2     "message": "Token missing"
3 }
```

GET `http://localhost:3000/protected` — Send

Params Headers **Auth** • Body `</>`

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ
9.eyJpZCI6MSwidXNlcm5hbWUiOiJ0ZXN0d
XNlciIsImlhdCI6MTc1MjQ3NTMzOSwiZXhw
IjoxNzUyNDc4OTM5fQ.Bt0iYA9USb9WGc7p
RMH6IhQ2NSwIrKYIlE2mrUhOOmE
```

Request **GET** Response **200**

▶ HTTP/1.1 **200 OK** (6 headers)

```
1 ▼ {
2     "message": "You have accessed a protected route!",
3 ▼     "user": {
4         "id": 1,
5         "username": "testuser",
6         "iat": 1752475339,
7         "exp": 1752478939
8     }
9 }
```