Practice 2 - Authentication System

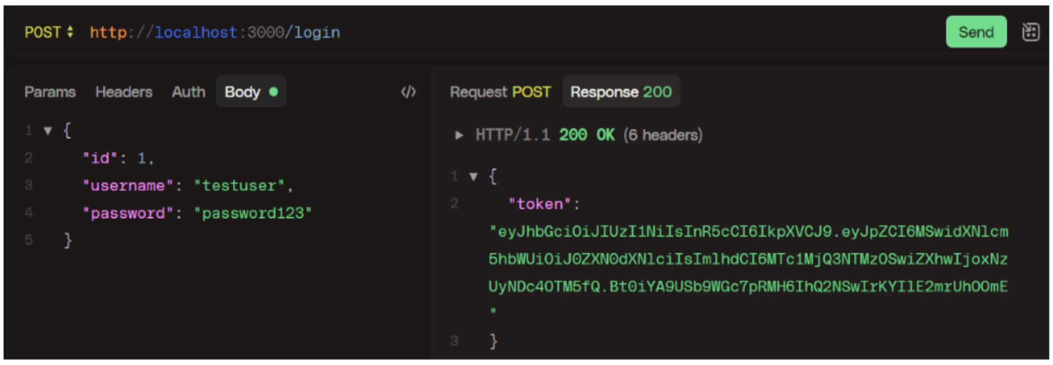
Instruction:
Perform the following tasks in the Nimbus.

Title: Implement Protected Routes with JWT Verification

Objective:
Learn how to secure backend API routes using JSON Web Tokens (JWT) to ensure that only authenticated users can access certain resources. This task helps you understand token-based authentication, how to verify JWTs on the server side, and how to protect specific routes from unauthorized access.

Task Description:
Create a Node.js and Express.js backend that uses JWT to protect certain API routes. Implement a login route that issues a JWT token when valid user credentials are provided (you can hardcode a sample user for simplicity). Create a middleware function that verifies the JWT token sent in the Authorization header as a Bearer token. Apply this middleware to one or more protected routes so that these routes can only be accessed if the token is valid. Test your implementation by accessing the protected route with and without a valid token to confirm that unauthorized requests are blocked and authorized requests succeed.

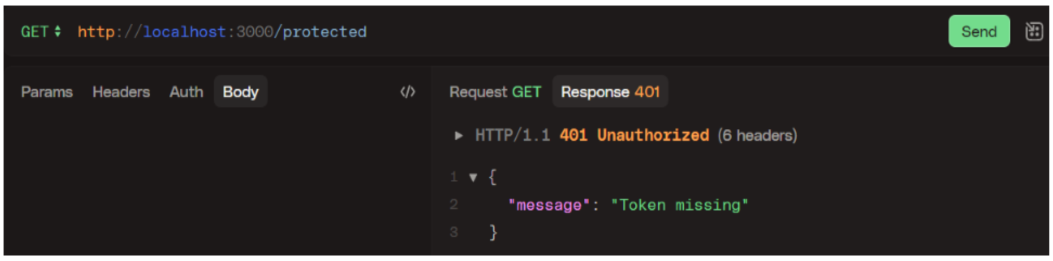
Expected Output:



```
http://localhost:3000/login
```

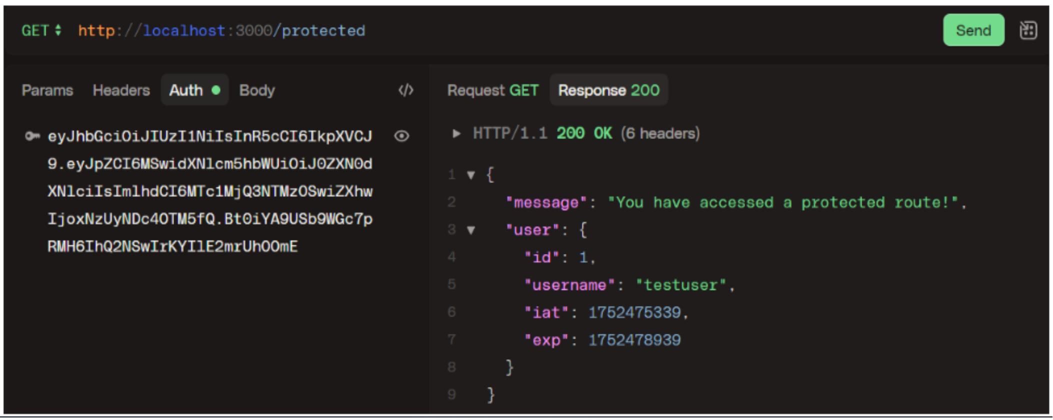
```
{
  "id": 1,
  "username": "testuser",
  "password": "password123"
}
```

```
HTTP/1.1 200 OK (6 headers)
{
  "token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MSwidXNlcm
5hbWUiOiJ0ZXN0dXNlciIsImlhdCI6MTc1MjQ3NTMzOSwiZXhwIjoxNz
UyNDc4OTM5fQ.Bt0iYA9USb9WGc7pRMH6IhQ2NSwIrKYIlE2mrUhOOmE
"
}
```



```
http://localhost:3000/protected
```

```
HTTP/1.1 401 Unauthorized (6 headers)
{
  "message": "Token missing"
}
```



```
http://localhost:3000/protected
```

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

```
HTTP/1.1 200 OK (6 headers)
{
  "message": "You have accessed a protected route!",
  "user": {
    "id": 1,
    "username": "testuser",
    "iat": 1752475339,
    "exp": 1752478939
  }
}
```

Node.js Code:

```
// Import required modules
const express = require('express');
const jwt = require('jsonwebtoken');
const bodyParser = require('body-parser');

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

// Secret key for JWT
const SECRET_KEY = 'mysecretkey';

// Sample user
const user = {
  id: 1,
  username: 'testuser',
  password: 'password123'
};

// Login route
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, { expiresIn: '1h' }
    res.json({ token });
  } else {
    res.status(401).json({ message: 'Invalid credentials' });
  }
});

// Middleware to verify token
function verifyToken(req, res, next) {
  const authHeader = req.headers['authorization'];
  const token = authHeader && authHeader.split(' ')[1];

  if (!token) return res.status(401).json({ message: 'Token missing' });

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

// Protected route
app.get('/protected', verifyToken, (req, res) => {
  res.json({
    message: 'You have accessed a protected route!',
    user: req.user
  });
});

// Start server
app.listen(3000, () => console.log('Server running on http://localhost:3000'));
```