

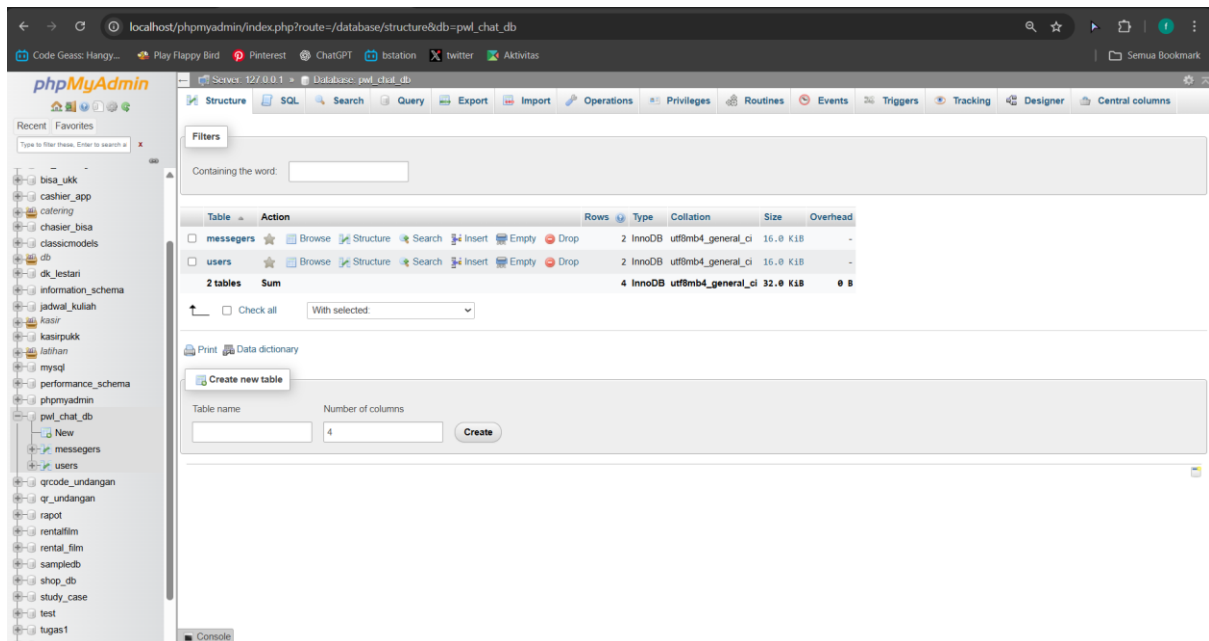
TUGAS 4 LAB. PWL

Nama: Faiz Faqih Haidar

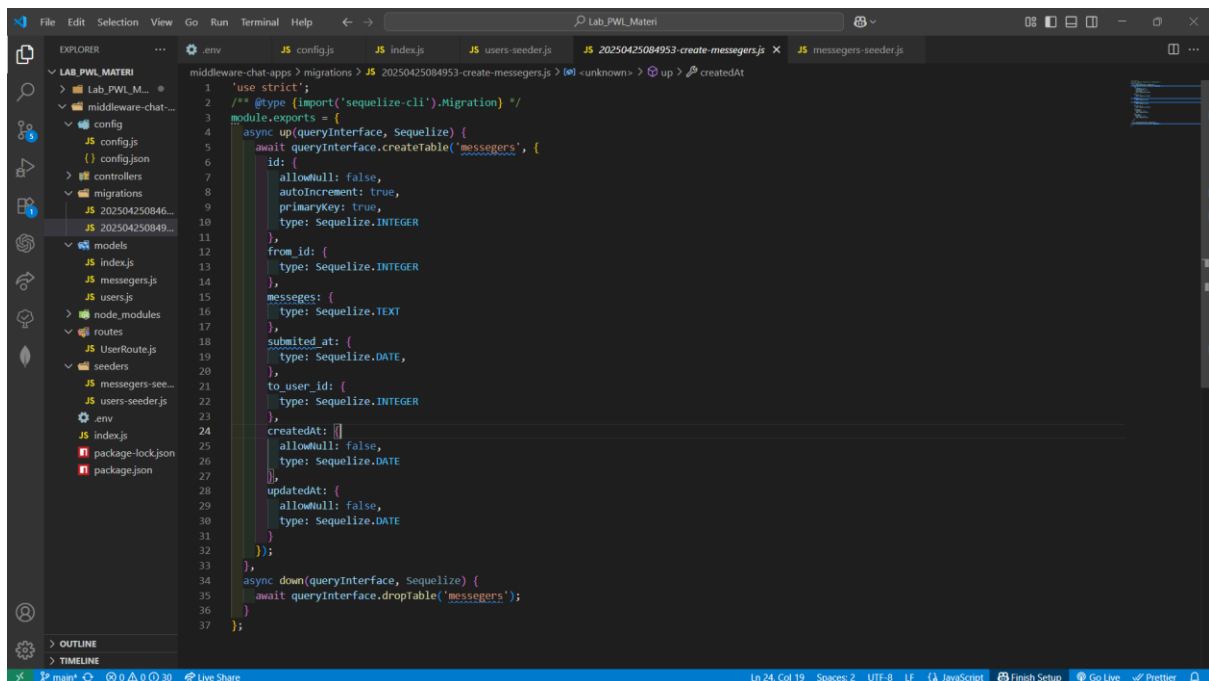
NPM: 222310047

Kelas: TI-22-PB

1.



2.



3.

The screenshot shows the VS Code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure for 'Lab_PWL_Materi'. The code editor displays the 'UserController.js' file, which contains the following code:

```
1 const { Users } = require('../models');
2 const bcrypt = require('bcryptjs');
3 const jwt = require('jsonwebtoken');
4
5 exports.createUser = async (req, res) => {
6   try {
7     const { username, password, fullname } = req.body;
8     const hashedPassword = await bcrypt.hash(password, 10);
9     const newUser = await Users.create({ username, password: hashedPassword, fullname });
10    res.status(201).json({ message: "User Created Successfully", code: 201, user: { id: newUser.id, username, fullname } });
11  } catch (error) {
12    res.status(400).json({ error: error.message });
13  }
14 };
15
16 exports.loginUser = async (req, res) => {
17   const { username, password } = req.body;
18
19   try {
20     const user = await Users.findOne({ where: { username } });
21
22     if (!user) return res.status(404).json({ error: 'User not found' });
23
24     const match = await bcrypt.compare(password, user.password);
25     if (!match) return res.status(401).json({ error: 'Invalid password' });
26
27     const token = jwt.sign(
28       { id: user.id, username: user.username, fullname: user.fullname },
29       '123456789',
30       { expiresIn: '1d' }
31     );
32
33     res.status(200).json({
34       message: 'Login successful',
35       code: 200,
36       user: {
37         id: user.id,
38         username: user.username,
39         fullname: user.fullname
```

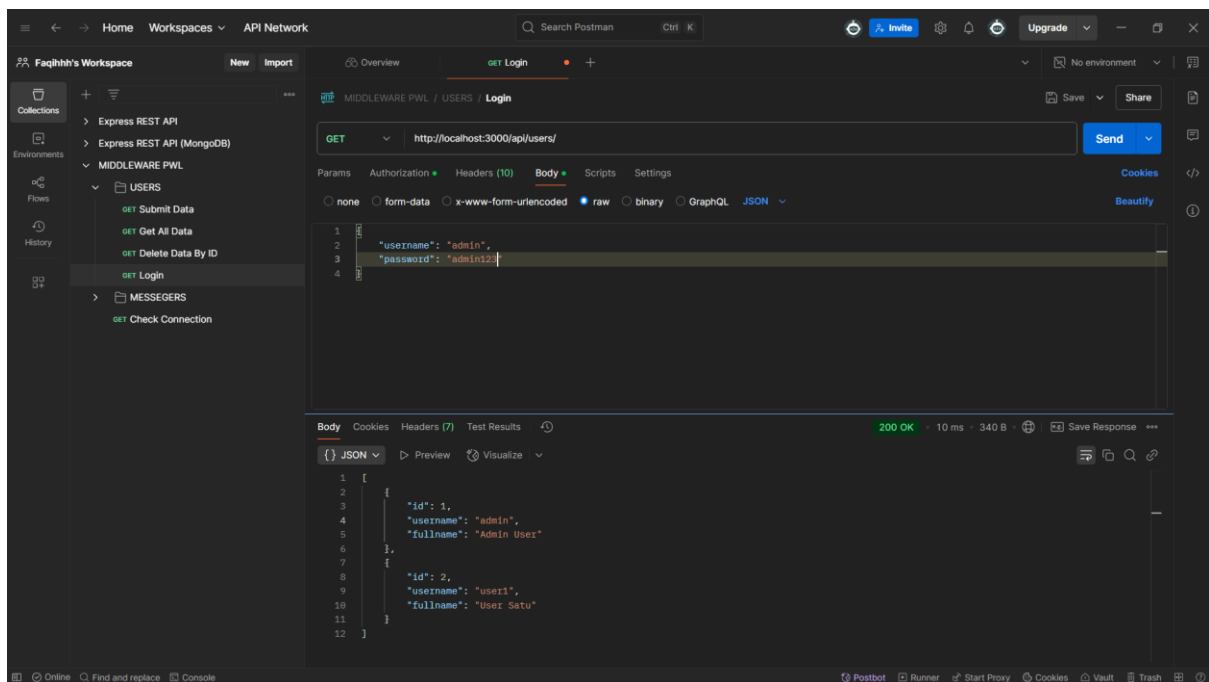
The screenshot shows the VS Code editor with the file explorer on the left and the code editor on the right. The file explorer shows the project structure for 'Lab_PWL_Materi'. The code editor displays the 'UserController.js' file, which contains the following code:

```
16 exports.loginUser = async (req, res) => {
36   user: {
40     },
41     token
42   });
43 } catch (err) {
44   res.status(500).json({ error: err.message });
45 }
46 };
47
48 exports.getAllUsers = async (req, res) => {
49   try {
50     const users = await Users.findAll({
51       attributes: ['id', 'username', 'fullname']
52     });
53     res.json(users);
54   } catch (error) {
55     res.status(500).json({ error: error.message });
56   }
57 };
58
59 exports.getUserById = async (req, res) => {
60   try {
61     const user = await Users.findById(req.params.id, {
62       attributes: ['id', 'username', 'fullname']
63     });
64     if (!user) return res.status(404).json({ message: 'User not found' });
65     res.json(user);
66   } catch (error) {
67     res.status(500).json({ error: error.message });
68   }
69 };
70
71
72 exports.updateUser = async (req, res) => {
73   const { id } = req.params;
74   const { username, password, fullname } = req.body;
```

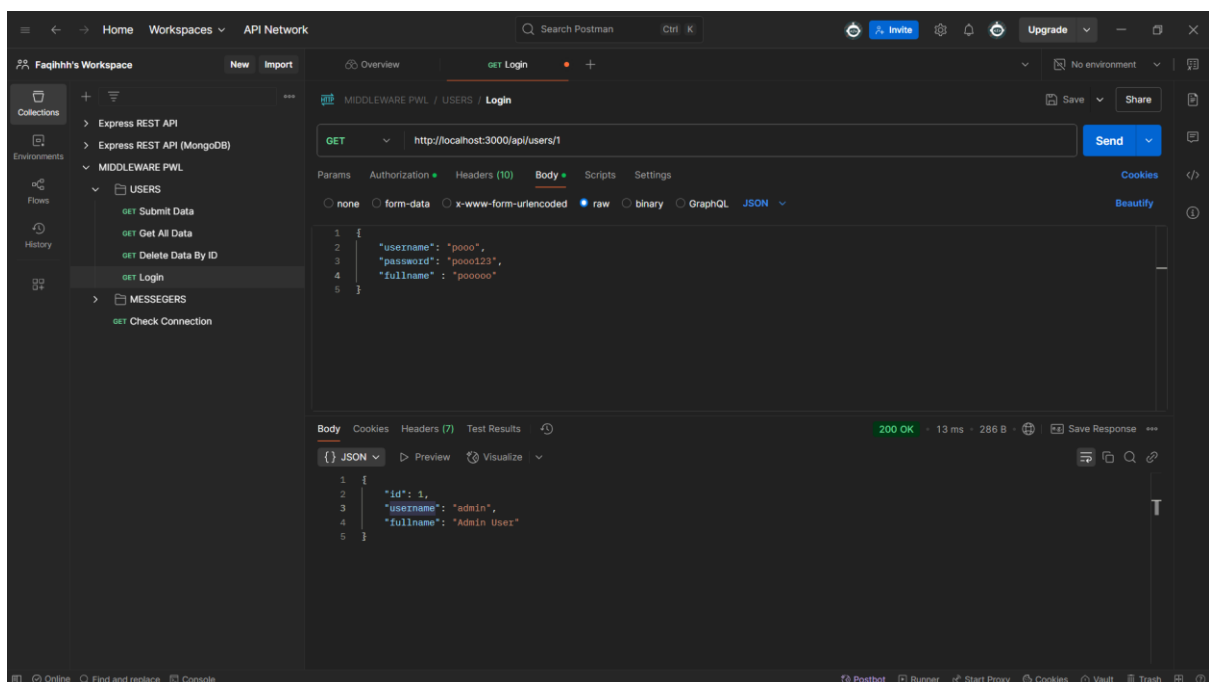
```
74 const { username, password, fullname } = req.body;
75
76 try {
77   const user = await Users.findById(id);
78   if (!user) return res.status(404).json({ error: 'User not found' });
79
80   if (password) {
81     const bcrypt = require('bcryptjs');
82     user.password = await bcrypt.hash(password, 10);
83   }
84
85   user.username = username || user.username;
86   user.fullname = fullname || user.fullname;
87
88   await user.save();
89   res.json({ message: 'User updated successfully', user });
90 } catch (error) {
91   res.status(400).json({ error: error.message });
92 }
93
94 exports.deleteUser = async (req, res) => {
95   const { id } = req.params;
96
97   try {
98     const user = await Users.findById(id);
99     if (!user) return res.status(404).json({ error: 'User not found' });
100
101     await user.destroy();
102     res.json({ message: 'User deleted successfully' });
103   } catch (error) {
104     res.status(500).json({ error: error.message });
105   }
106 }
107
108
```

```
1 const express = require('express');
2 const router = express.Router();
3 const messageController = require('../controllers/MessageController');
4 const { authenticateUser } = require('../middlewares/authMiddleware');
5
6 router.post('/', authenticateUser, messageController.sendMessage);
7 router.get('/:user1_id/:user2_id', authenticateUser, messageController.getMessagesBetweenUsers);
8 router.put('/:id', authenticateUser, messageController.updateMessage);
9 router.delete('/:id', authenticateUser, messageController.deleteMessage);
10
11 module.exports = router;
```

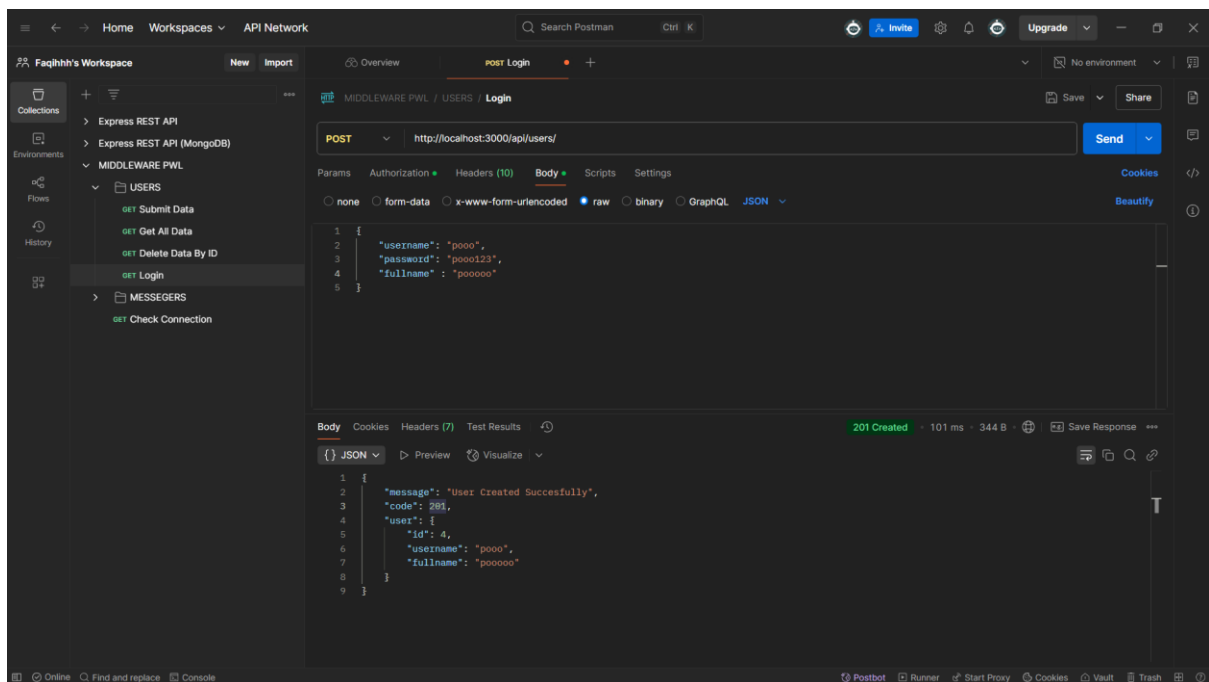

B. Get All Users (get)



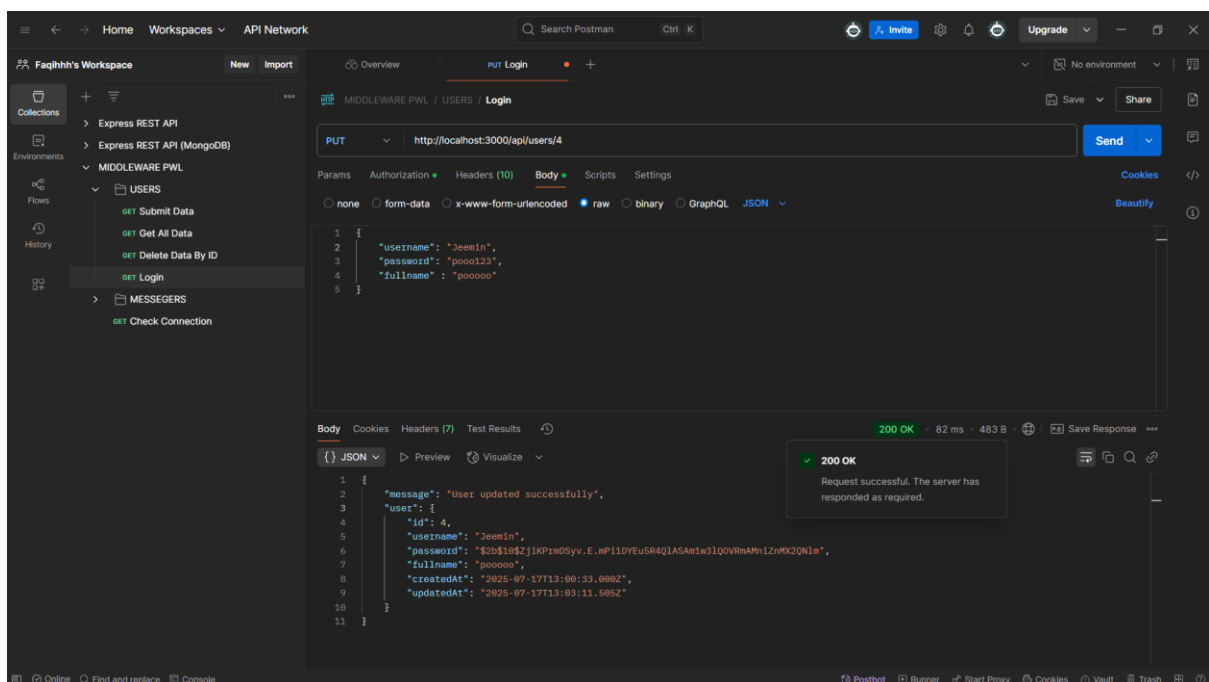
C. Get user by id (get)



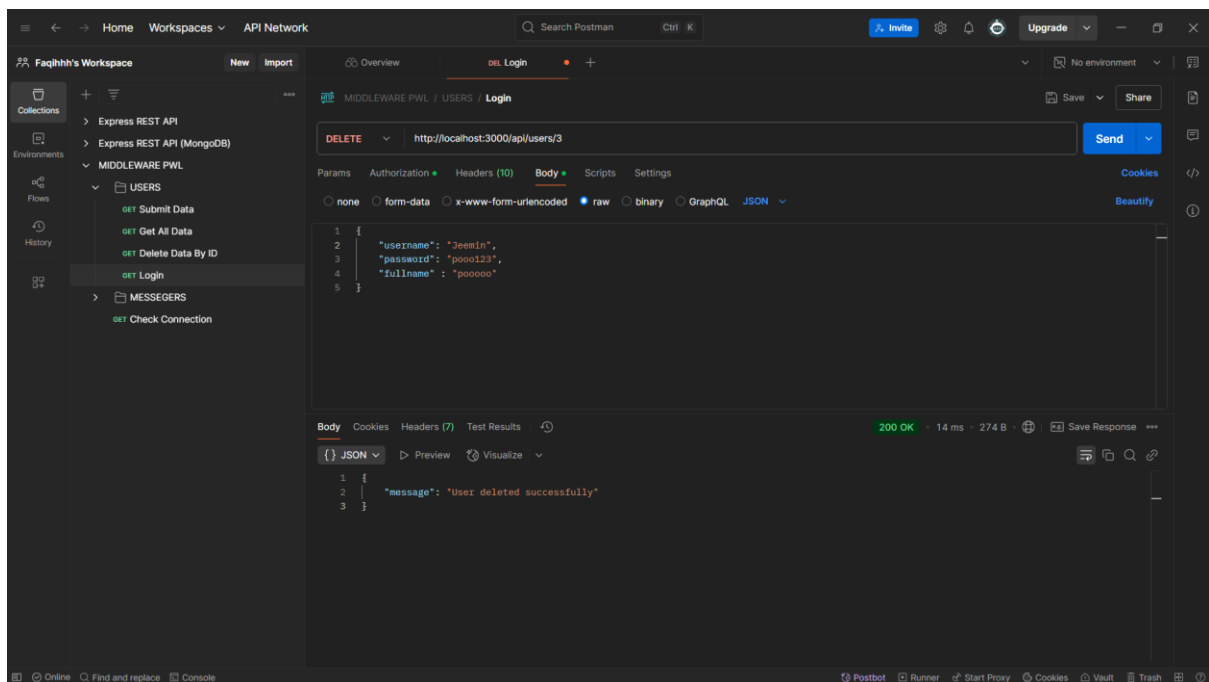
D. Create User (post)



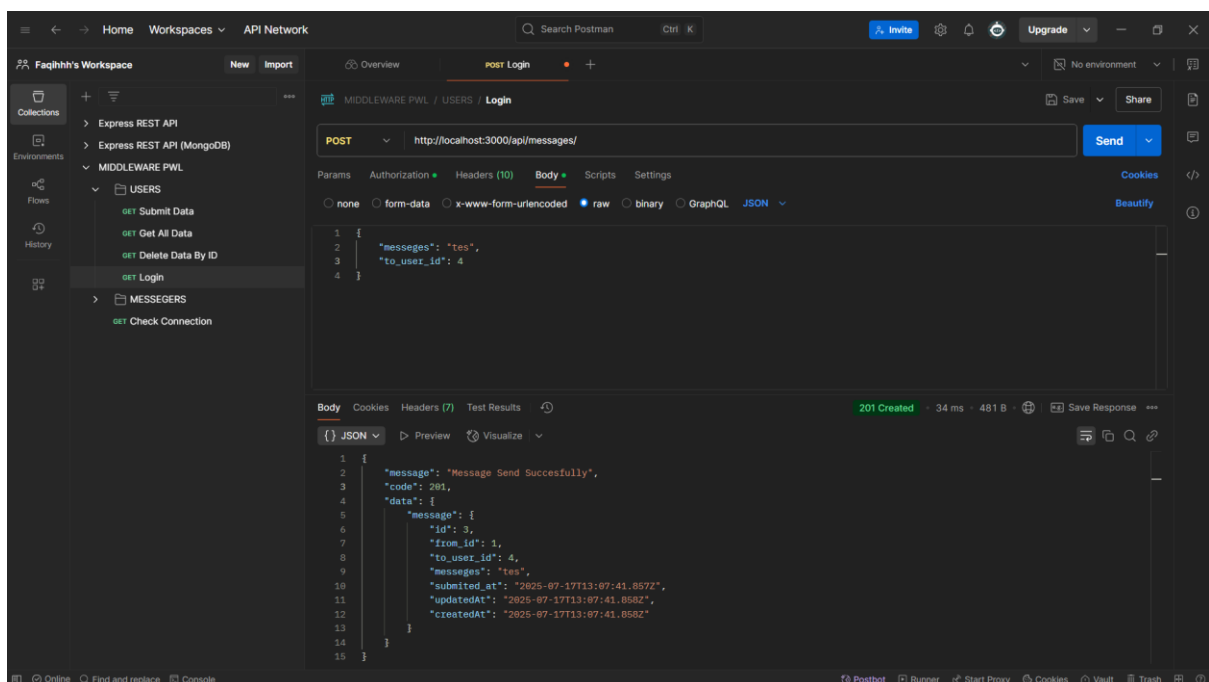
E. Edit User (put)



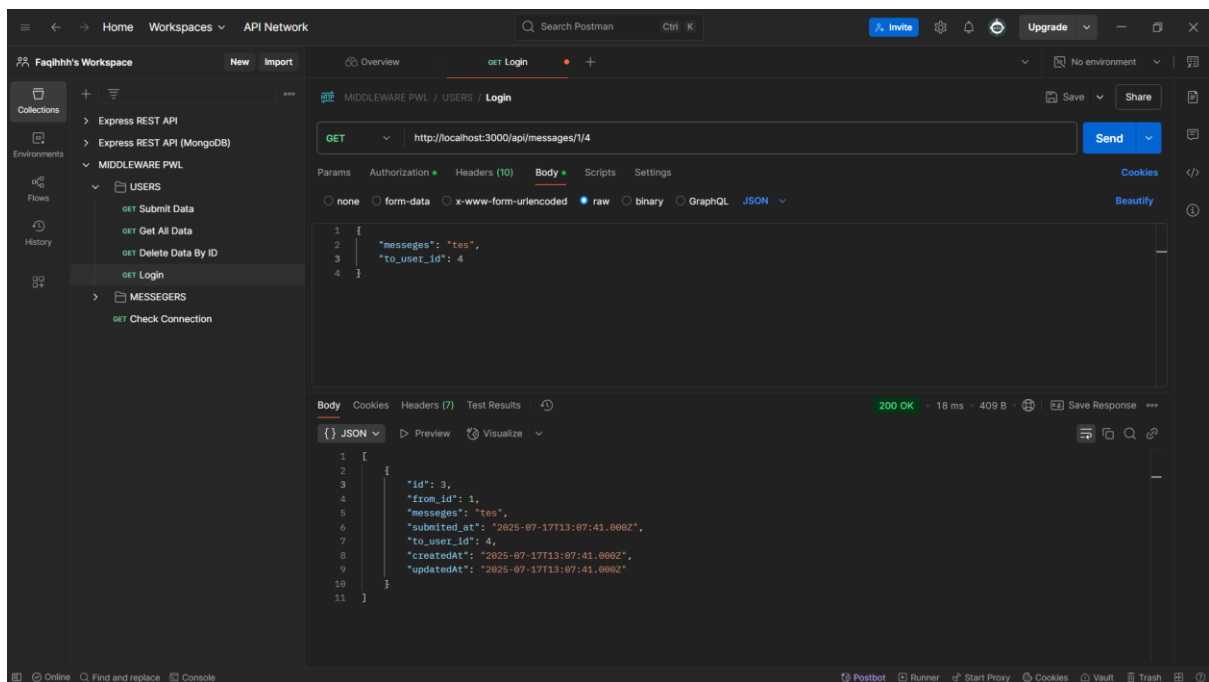
F. Delete User



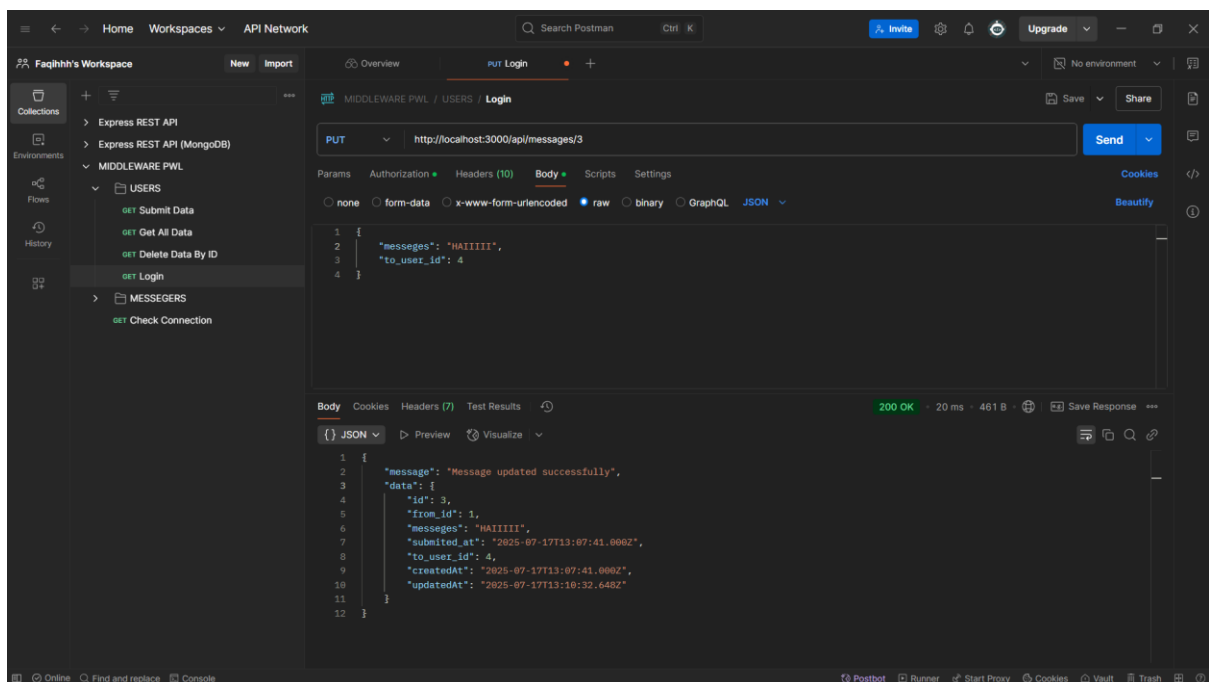
G. Send Message (Post)



H. Check History antara User (get)



I. Update Message (put)



J. Delete Message (delete)

