

w12-p1 switch Register and Login, and log

The screenshot shows a web application with a Login form. The code on the left is as follows:

```
const initialState = {
  name: '',
  email: '',
  password: '',
  isMember: true,
  showAlert: false
};

const Register_85 = () => {
  const [values, setValues] = useState(initialState);

  const { showAlert, displayAlert } = useAppContext();

  const toggleMember = () => {
    setValues({ ...values, isMember: !values.isMember });
  };

  const handleChange = (e) => {
    console.log('e.target', e.target);
    setValues({ ...values, [e.target.name]: e.target.value });
  };

  const onSubmit = (e) => {
    e.preventDefault();
    //console.log('e.target', e.target);
    const { email, password } = values;
    if (!email || !password) {

```

The browser shows the Login form with a "Not a member yet? Register" link. The React DevTools component inspector shows the Register_85 component with the following state:

```
State: { email: '', isMember: true, name: '', password: '' }
isMember: true
```

The screenshot shows a web application with a Register form. The code on the left is as follows:

```
const Register_85 = () => {
  const [values, setValues] = useState(initialState);

  const { showAlert, displayAlert } = useAppContext();

  const toggleMember = () => {
    setValues({ ...values, isMember: !values.isMember });
  };

  const handleChange = (e) => {
    console.log('e.target', e.target);
    setValues({ ...values, [e.target.name]: e.target.value });
  };

  const onSubmit = (e) => {
    e.preventDefault();
    //console.log('e.target', e.target);
    const { email, password } = values;
    if (!email || !password) {

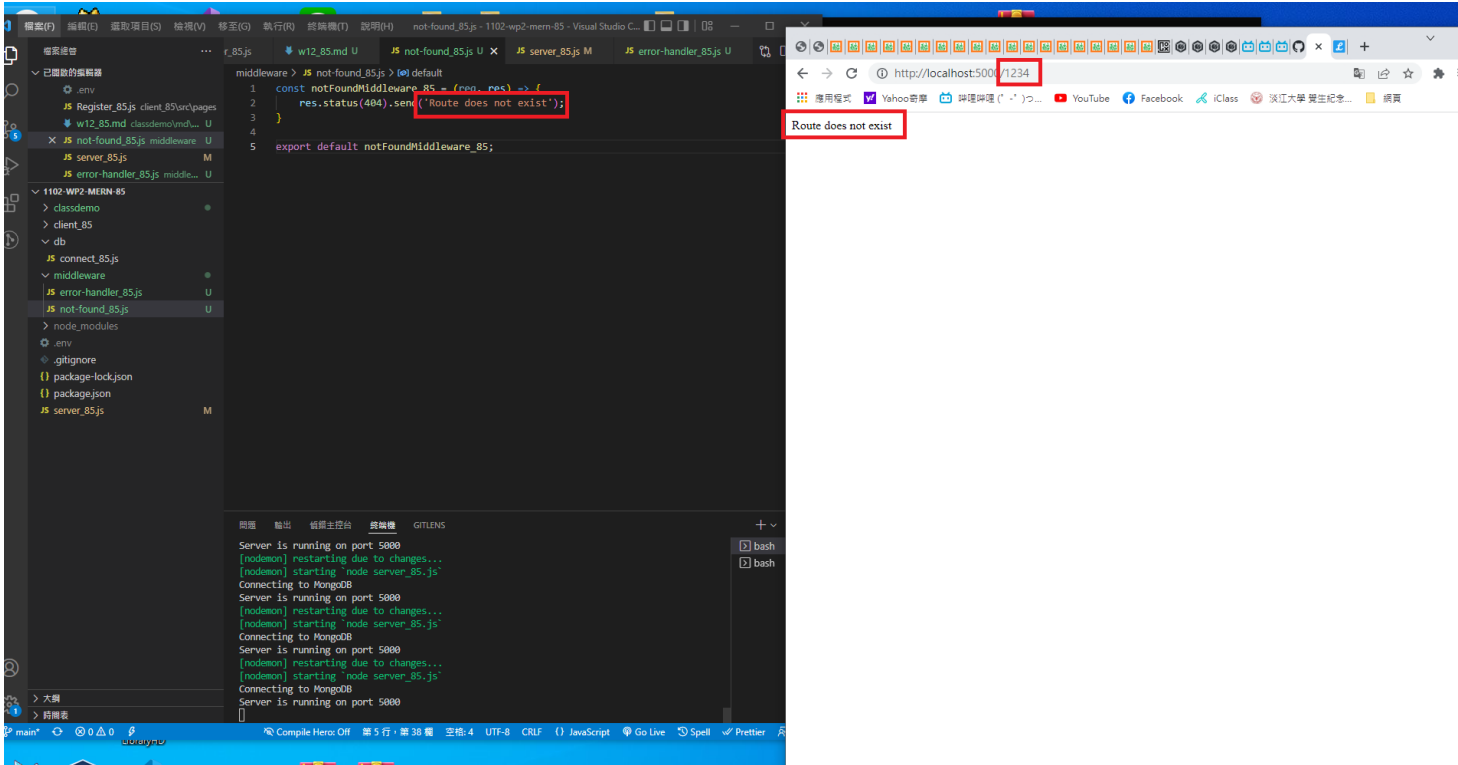
```

The browser shows the Register form with an "Already a member? Login" link. The React DevTools component inspector shows the Register_85 component with the following state:

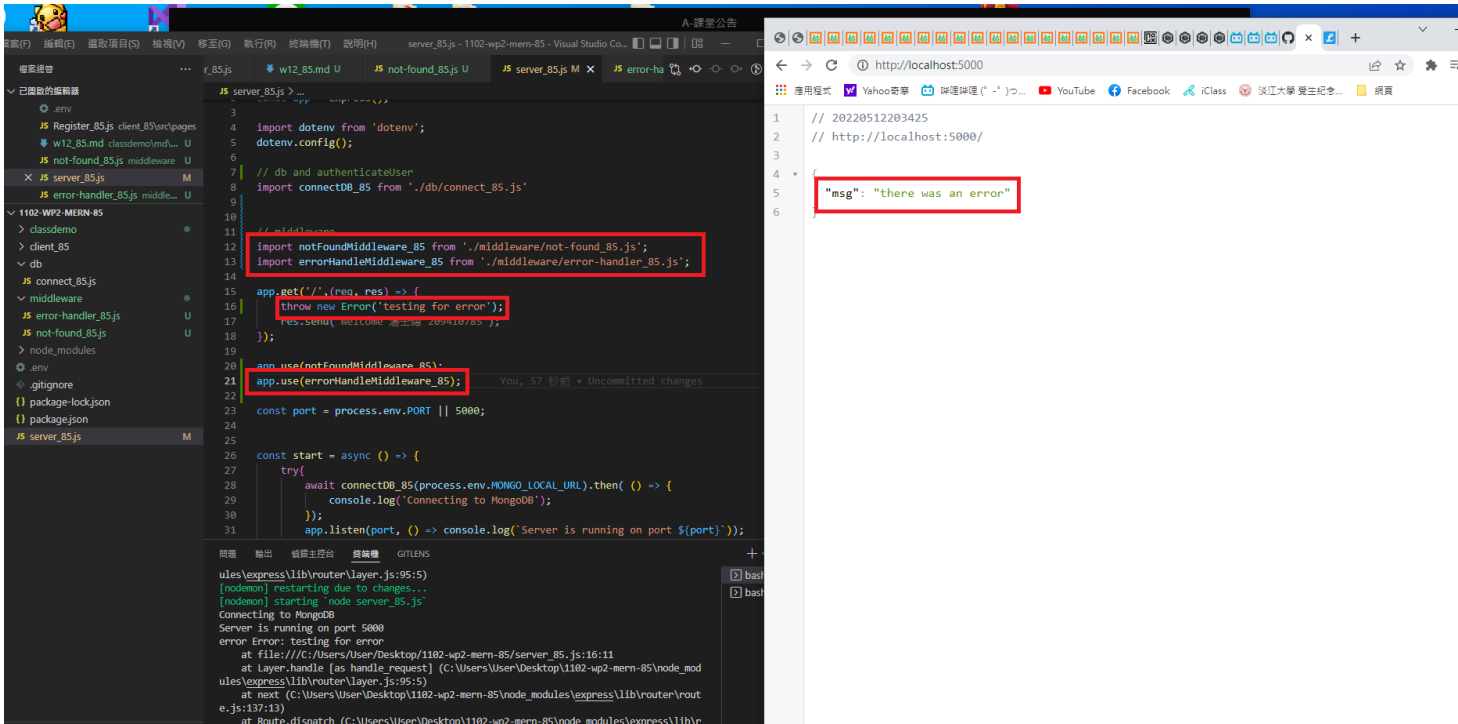
```
State: { email: '', isMember: false, name: '', password: '' }
isMember: false
```

```
User@DESKTOP-102TF2A MINGW64 ~/Desktop/1102-wp2-mern-85 (main)
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%" --after="2022-05-11"
50ba110 tsw0603 Thu May 12 20:09:30 2022 +0800 w12-p1 switch Register and Login
```

w12-p2-1 test for notFoundMiddleware_85.js



w12-p2-2 test for errorHandlerMiddleware_85.js



```
User@DESKTOP-102TF2A MINGW64 ~/Desktop/1102-wp2-mern-85 (main)
$ git log --pretty=format:"%h%x09%an%x09%ad%x09%s" --after="2022-05-11"
7745f22 tsw0603 Thu May 12 20:39:34 2022 +0800 w12-p2-1 test for notFoundMiddleware_85.
js, and w12-p2-2 test for errorHandlerMiddleware_85.js
50ba110 tsw0603 Thu May 12 20:09:30 2022 +0800 w12-p1 switch Register and Login
User@DESKTOP-102TF2A MINGW64 ~/Desktop/1102-wp2-mern-85 (main)
```

w12-p3 Use postman to test route /api/v1/auth_85x/register_85

The screenshot shows the Postman application interface. The main workspace displays a POST request configuration for the route `/api/v1/auth_85x/register_85`. The request body is a JSON object: `{ "register user -- 湯士健, 209410785" }`. The interface includes a sidebar with collections, a main workspace, and a right-hand panel with tabs for Params, Authorization, Headers, Body, Pre-request Script, Tests, and Settings.

Key elements highlighted in the image:

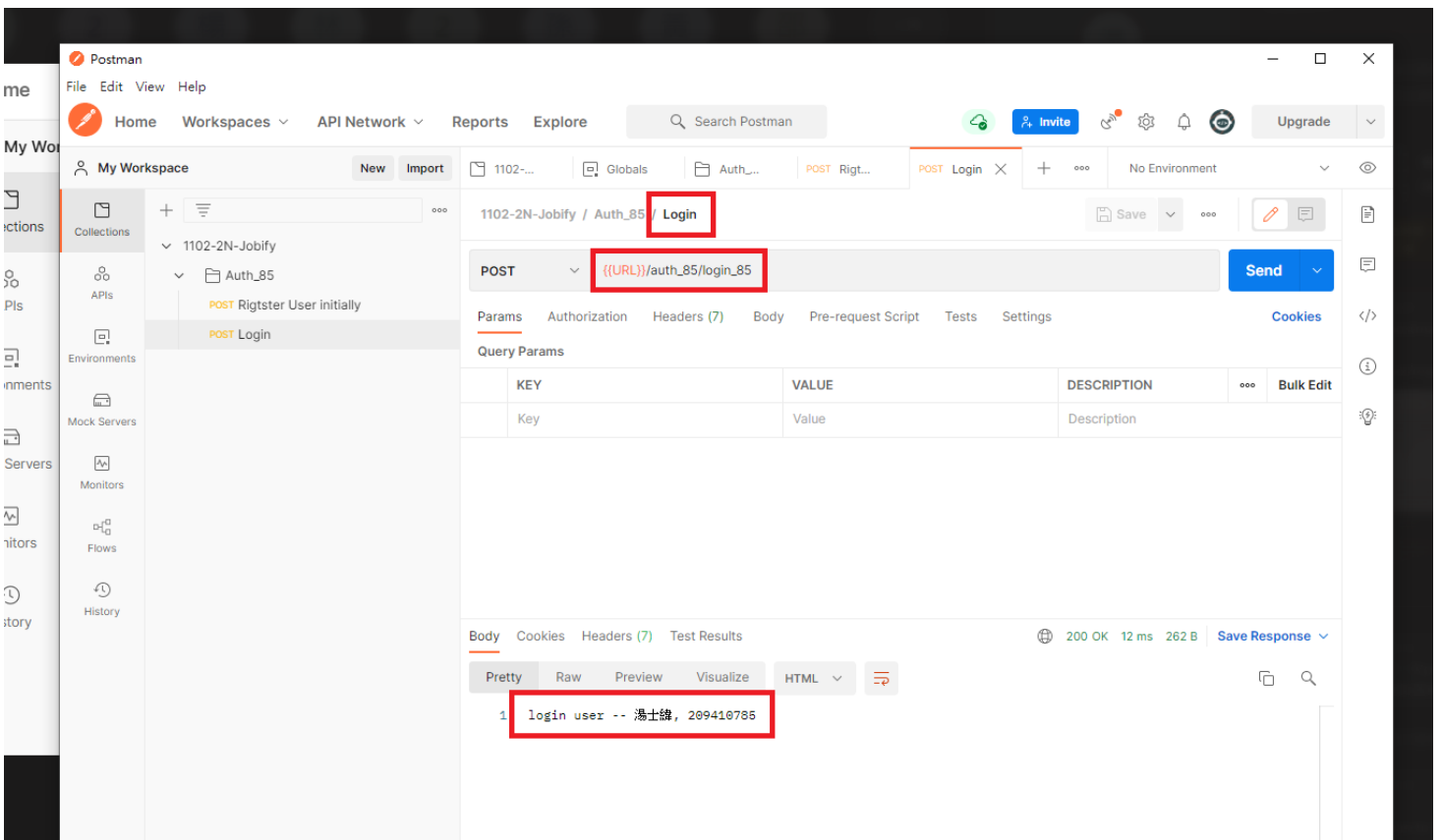
- The **POST** method is selected.
- The URL is `{{URL}}/auth_85/register_85`.
- The request body is a JSON object: `{ "register user -- 湯士健, 209410785" }`.

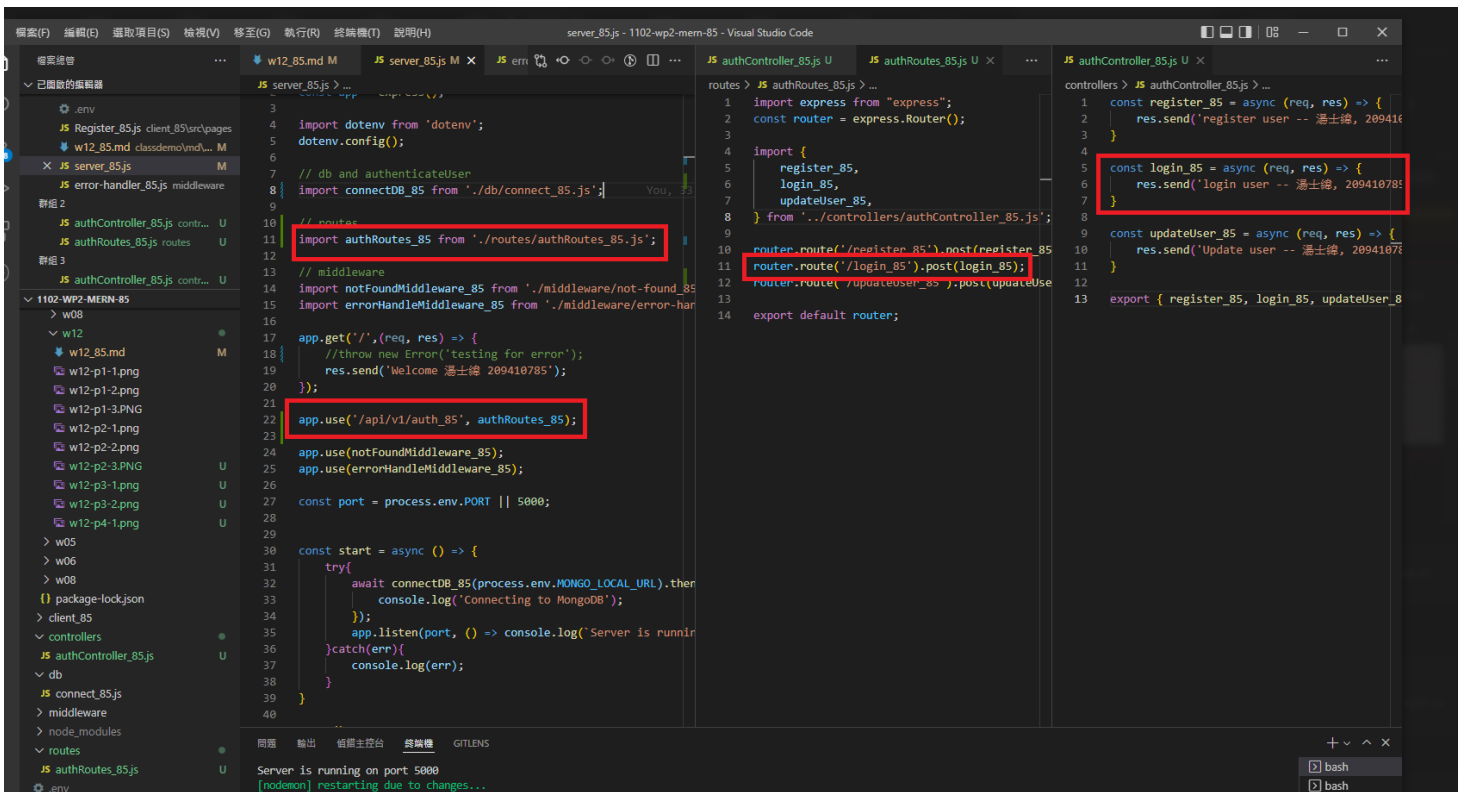
```
server_85.js
1 import express from 'express';
2 const router = express.Router();
3
4 import {
5   register_85,
6   login_85,
7   updateUser_85,
8 } from './controllers/authController_85.js';
9
10 router.route('/register_85').post(register_85);
11 router.route('/login_85').post(login_85);
12 router.route('/updateUser_85').post(updateUser_85);
13
14 export default router;

authRoutes_85.js
1 import express from 'express';
2 const router = express.Router();
3
4 import {
5   register_85,
6   login_85,
7   updateUser_85,
8 } from './controllers/authController_85.js';
9
10 router.route('/register_85').post(register_85);
11 router.route('/login_85').post(login_85);
12 router.route('/updateUser_85').post(updateUser_85);
13
14 export default router;

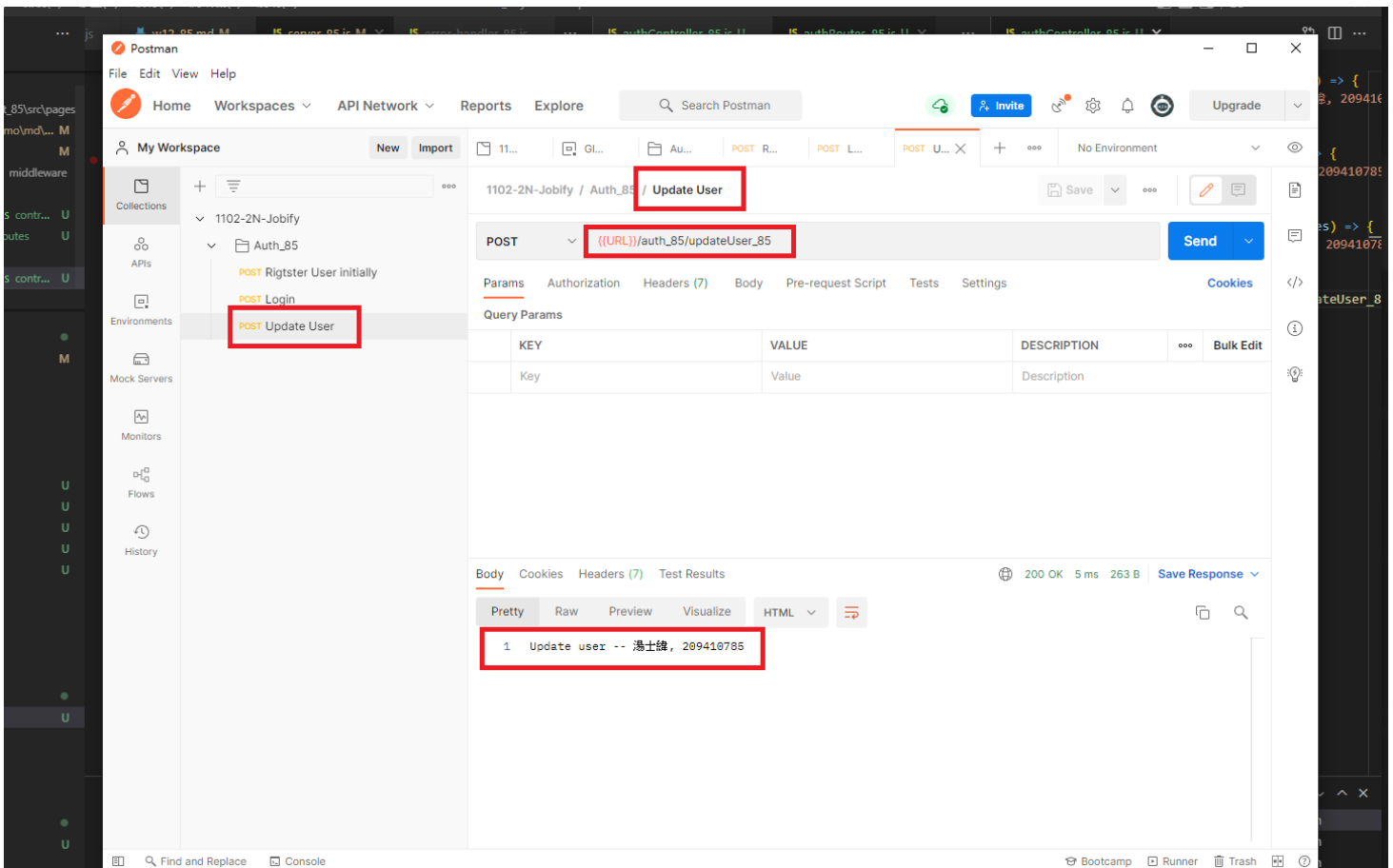
authController_85.js
1 const register_85 = async (req, res) => {
2   res.send('register user -- 湯士維, 209410785');
3 }
4
5 const login_85 = async (req, res) => {
6   res.send('login user -- 湯士維, 209410785');
7 }
8
9 const updateUser_85 = async (req, res) => {
10   res.send('Update user -- 湯士維, 209410785');
11 }
12
13 export { register_85, login_85, updateUser_85 };
```

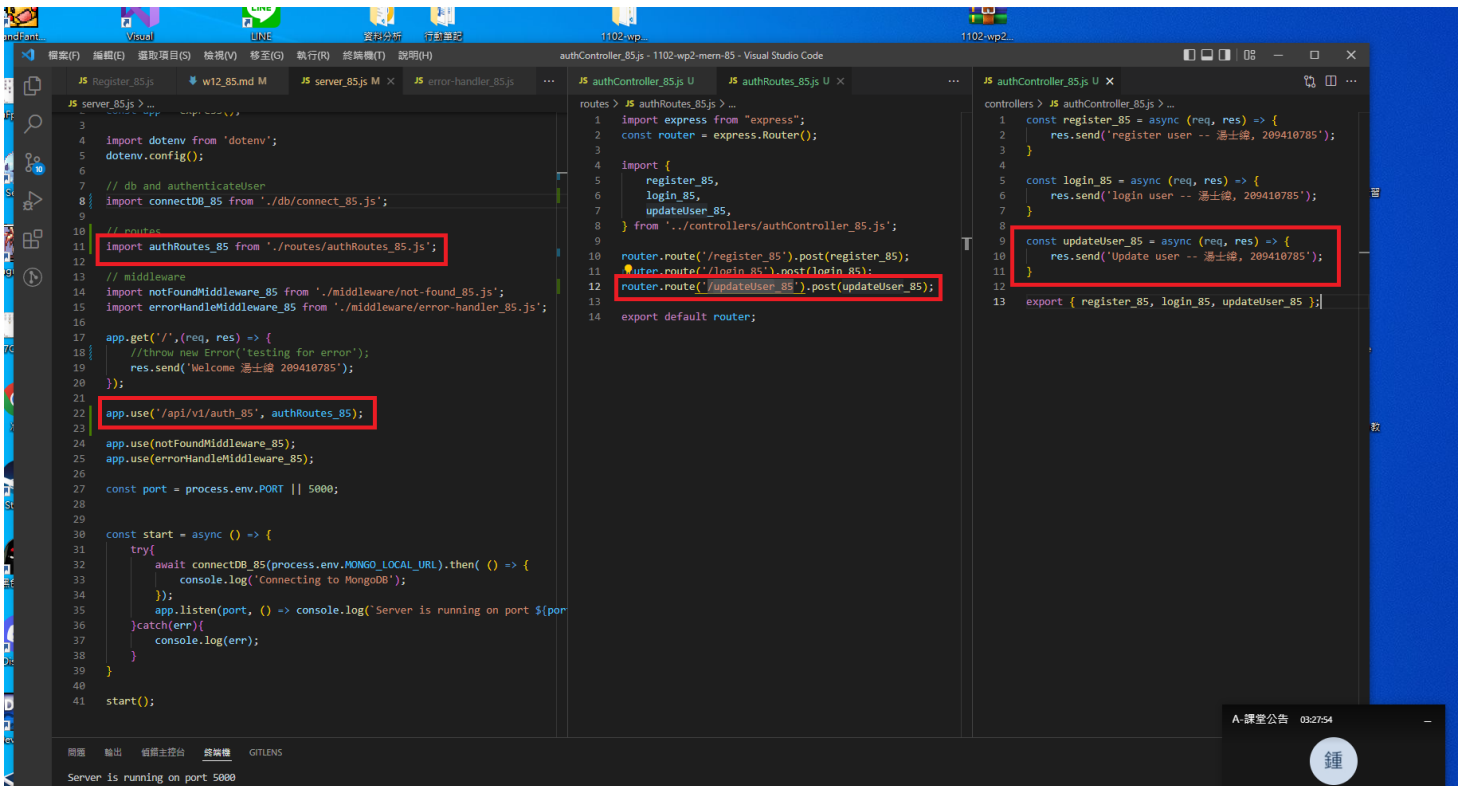
w12-p4 Use postman to test route /api/v1/auth_xx/login_xx





w12-p5 Use postman to test route /api/v1/auth_xx/updateUser_xx





W12-LAST-LOG show all W12 log records in Github

```
User@DESKTOP-102TF2A MINGW64 ~/Desktop/1102-wp2-mern-85 (main)
$ git log --pretty=format:"%h%x09%an%x09%ad%x09s" --after="2022-05-11"
1818697 tsw0603 Thu May 12 21:49:03 2022 +0800 w12-p3 Use postman to test route /api/v1/auth_85x/register_85, w12-p4 login, w12-p5 u
pdateuser
7745f22 tsw0603 Thu May 12 20:39:34 2022 +0800 w12-p2-1 test for notFoundMiddleware_85.js, and w12-p2-2 test for errorHandlerMiddlew
are_85.js
50ba110 tsw0603 Thu May 12 20:09:30 2022 +0800 w12-p1 switch Register and Login
```

```
$ git log --pretty=format:"%h%x09%an%x09%ad%x09s" --after="2022-05-11"
1818697 tsw0603 Thu May 12 21:49:03 2022 +0800 w12-p3 Use postman to test route /api/v1/auth_85x/register_85, w1
7745f22 tsw0603 Thu May 12 20:39:34 2022 +0800 w12-p2-1 test for notFoundMiddleware_85.js, and w12-p2-2 test for
50ba110 tsw0603 Thu May 12 20:09:30 2022 +0800 w12-p1 switch Register and Login
```