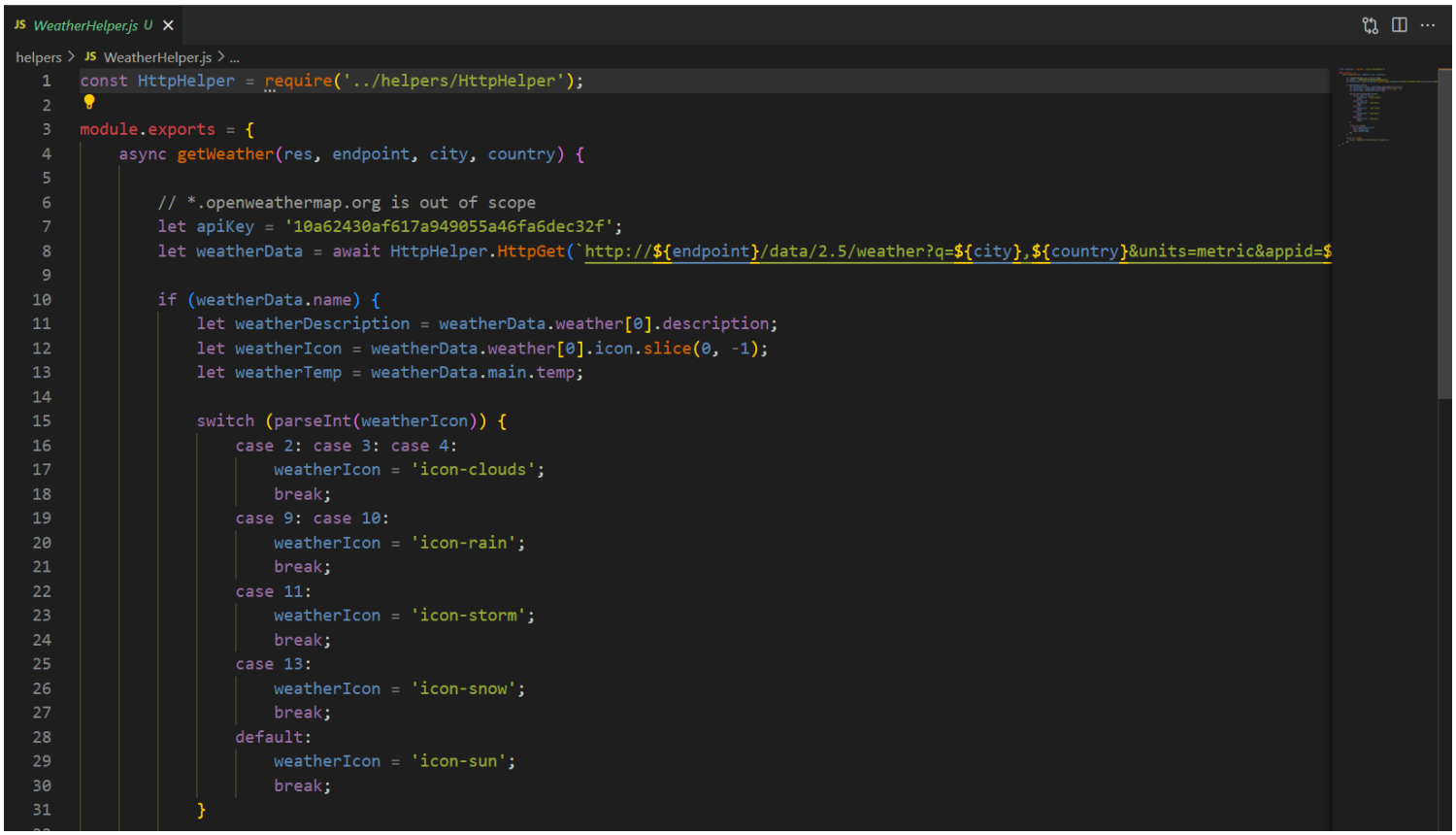
Weather App

1) Checked the source code

```
router.get('/login', (req, res) => {
    return res.sendFile(path.resolve('views/login.html'));
});

router.post('/login', (req, res) => {
    let { username, password } = req.body;

    if (username && password) {
        return db.isAdmin(username, password)
            .then(admin => {
                if (admin) return res.send(fs.readFileSync('/app/flag').toString());
                return res.send(response('You are not admin'));
            })
            .catch(() => res.send(response('Something went wrong')));
    }

    return re.send(response('Missing parameters'));
});
```

```
router.post('/register', (req, res) => {

    if (req.socket.remoteAddress.replace(/^.*:/, '') != '127.0.0.1') {
        return res.status(401).end();
    }

    let { username, password } = req.body;

    if (username && password) {
        return db.register(username, password)
            .then(()  => res.send(response('Successfully registered')))
            .catch(() => res.send(response('Something went wrong')));
    }

    return res.send(response('Missing parameters'));
});
```

```
router.post('/api/weather', (req, res) => {
    let { endpoint, city, country } = req.body;

    if (endpoint && city && country) {
        return WeatherHelper.getWeather(res, endpoint, city, country);
    }

    return res.send(response('Missing parameters'));
});
```



```
JS WeatherHelper.js U
helpers > JS WeatherHelper.js > ...
1  const HttpHelper = require('../helpers/HttpHelper');
2
3  module.exports = {
4      async getWeather(res, endpoint, city, country) {
5
6          // *.openweathermap.org is out of scope
7          let apiKey = '10a62430af617a949055a46fa6dec32f';
8          let weatherData = await HttpHelper.HttpGet(`http://${endpoint}/data/2.5/weather?q=${city},${country}&units=metric&appid=$
9
10         if (weatherData.name) {
11             let weatherDescription = weatherData.weather[0].description;
12             let weatherIcon = weatherData.weather[0].icon.slice(0, -1);
13             let weatherTemp = weatherData.main.temp;
14
15             switch (parseInt(weatherIcon)) {
16                 case 2: case 3: case 4:
17                     weatherIcon = 'icon-clouds';
18                     break;
19                 case 9: case 10:
20                     weatherIcon = 'icon-rain';
21                     break;
22                 case 11:
23                     weatherIcon = 'icon-storm';
24                     break;
25                 case 13:
26                     weatherIcon = 'icon-snow';
27                     break;
28                 default:
29                     weatherIcon = 'icon-sun';
30                     break;
31             }
```

We have to find a way to SSRF and register a user, we can do that by injecting on endpoint variable