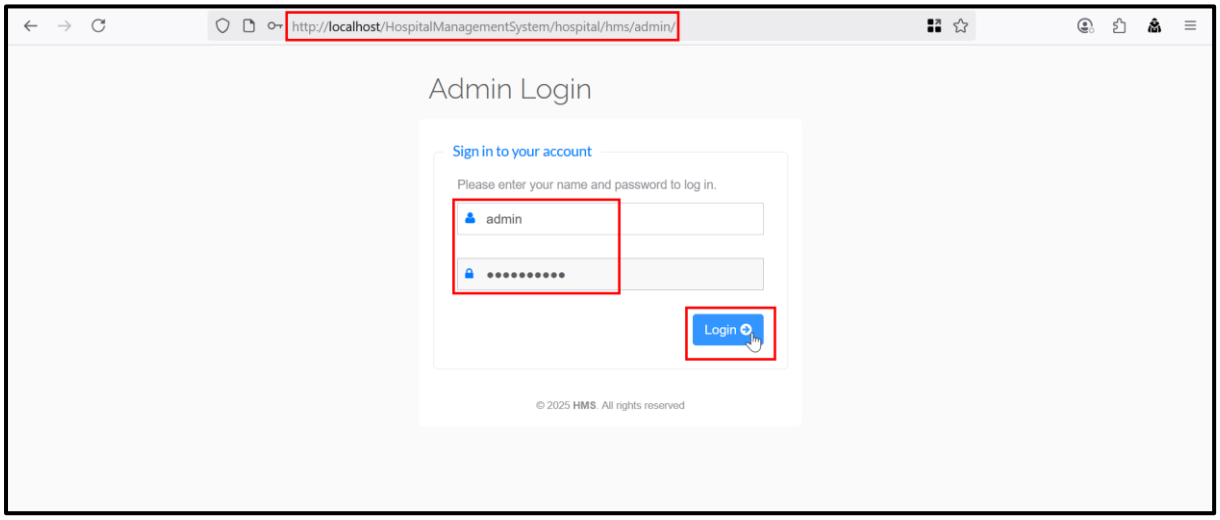
CVE-2025-9746: Edit Doctor Specialization XSS

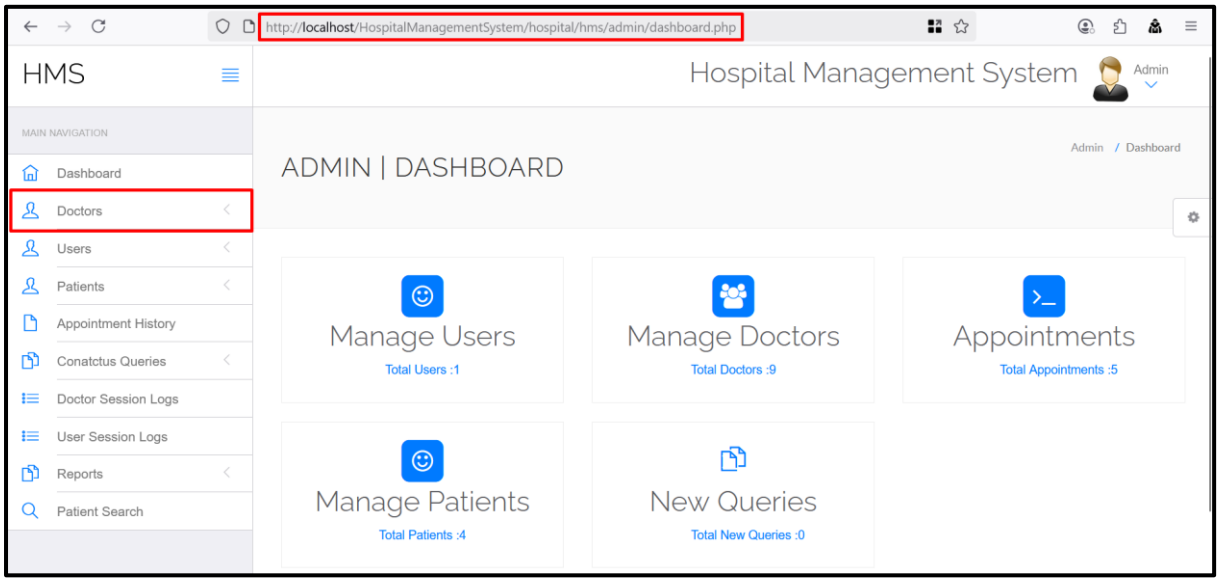
A vulnerability was detected in Campcodes Hospital Management System 1.0. This affects an unknown function of the file /admin/edit-doctor-specialization.php of the component Edit Doctor Specialization Page. The manipulation results in cross site scripting. The attack may be launched remotely. The exploit is now public and may be used.

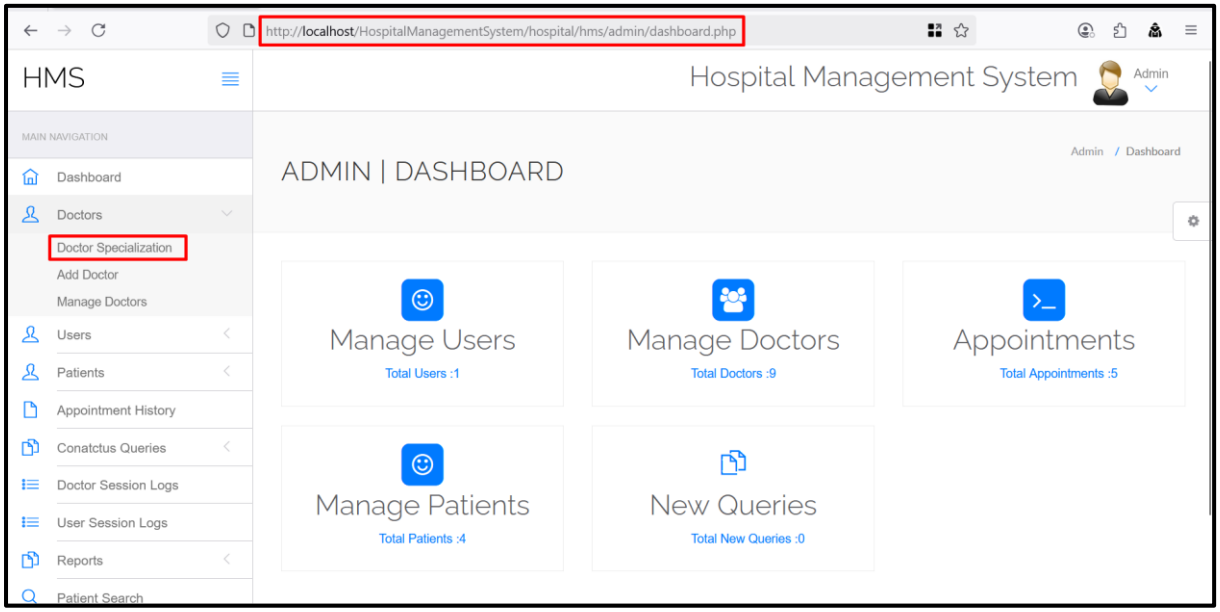
Proof of concept (POC):

Step 1: Access the URL and log in as an admin, as shown in the screenshot.

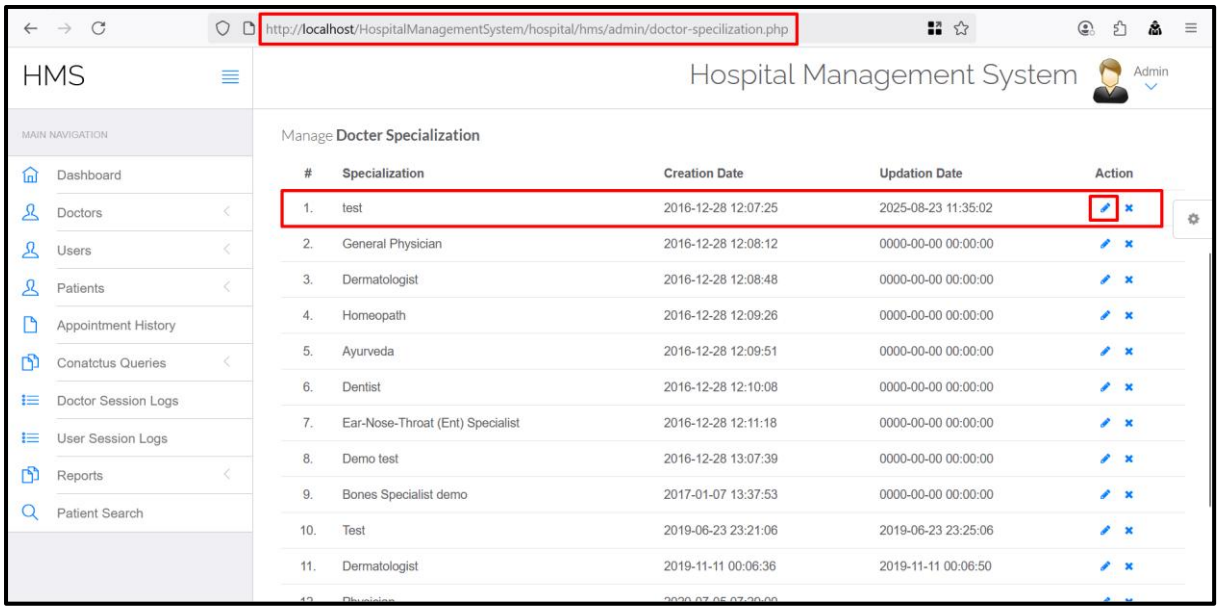


Step 2: After successfully logging in, click on “Doctors” and then select “Doctor Specialization,” as shown in the screenshot below.



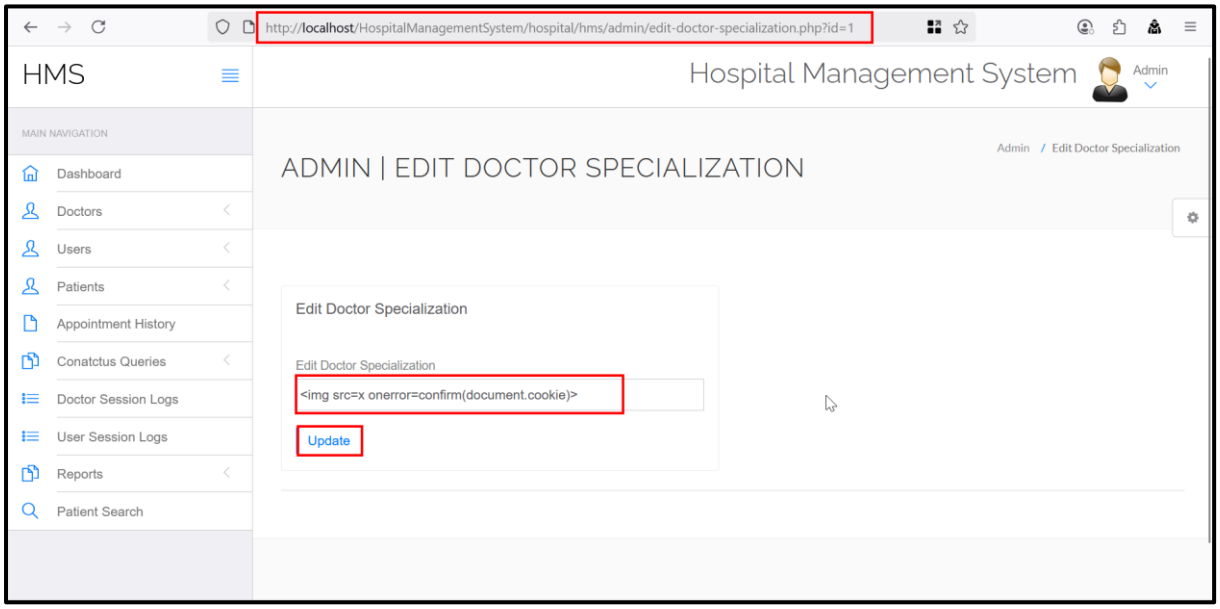


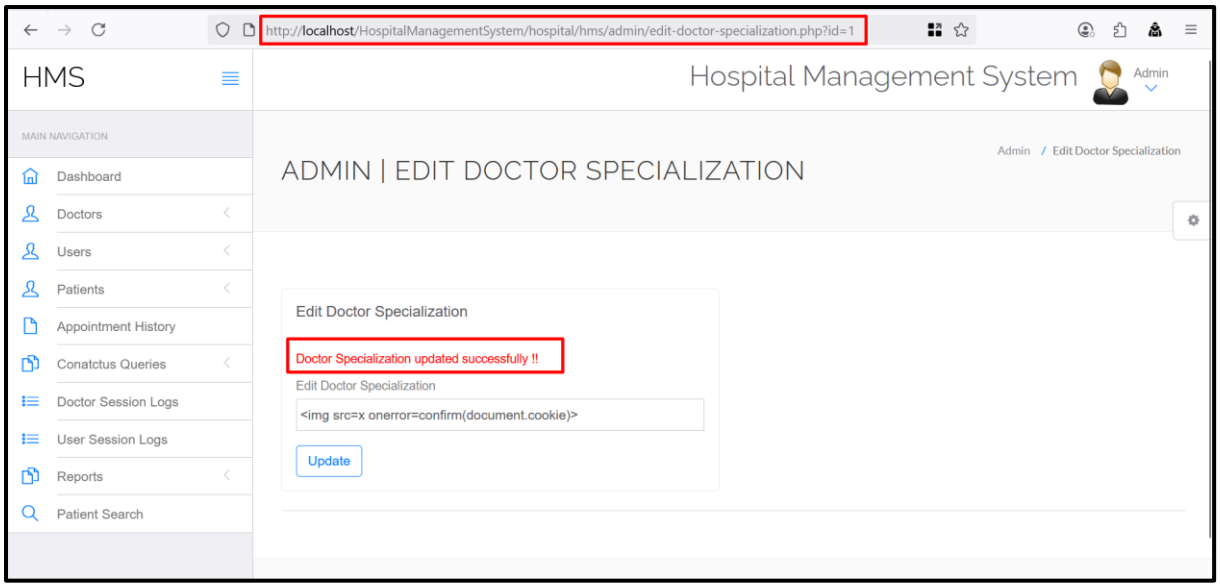
Step 3: Click the edit icon under the "Action" column, as shown in the screenshot below.



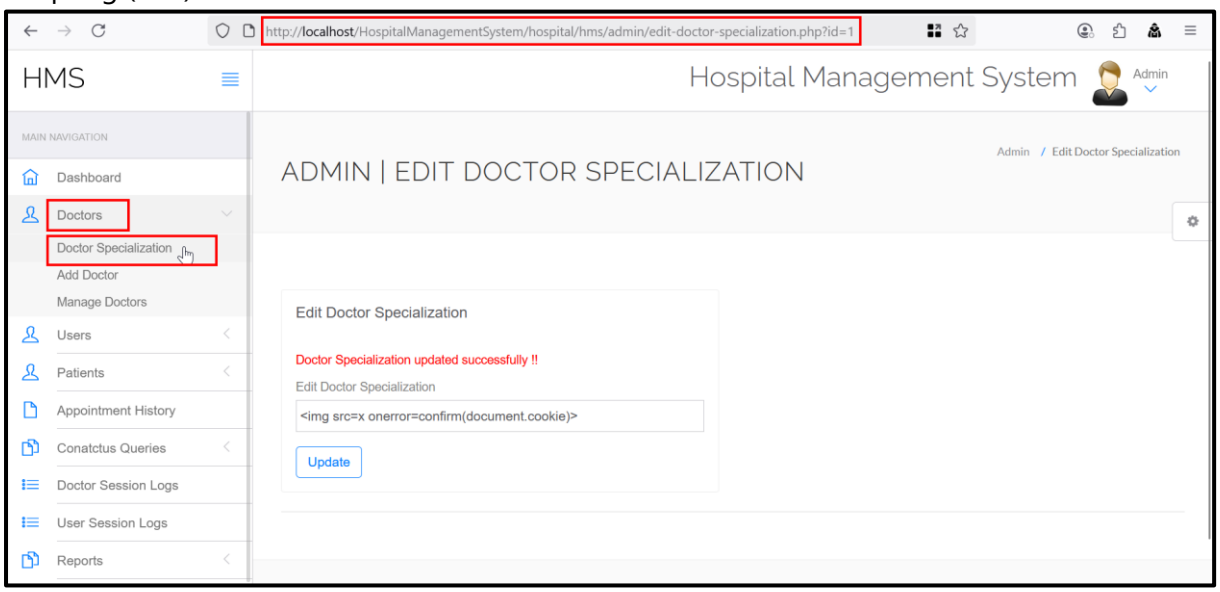
Step 4: Enter the following payload into the "Edit Doctor Specialization" field and click the "Update" button, as shown in the screenshot below.

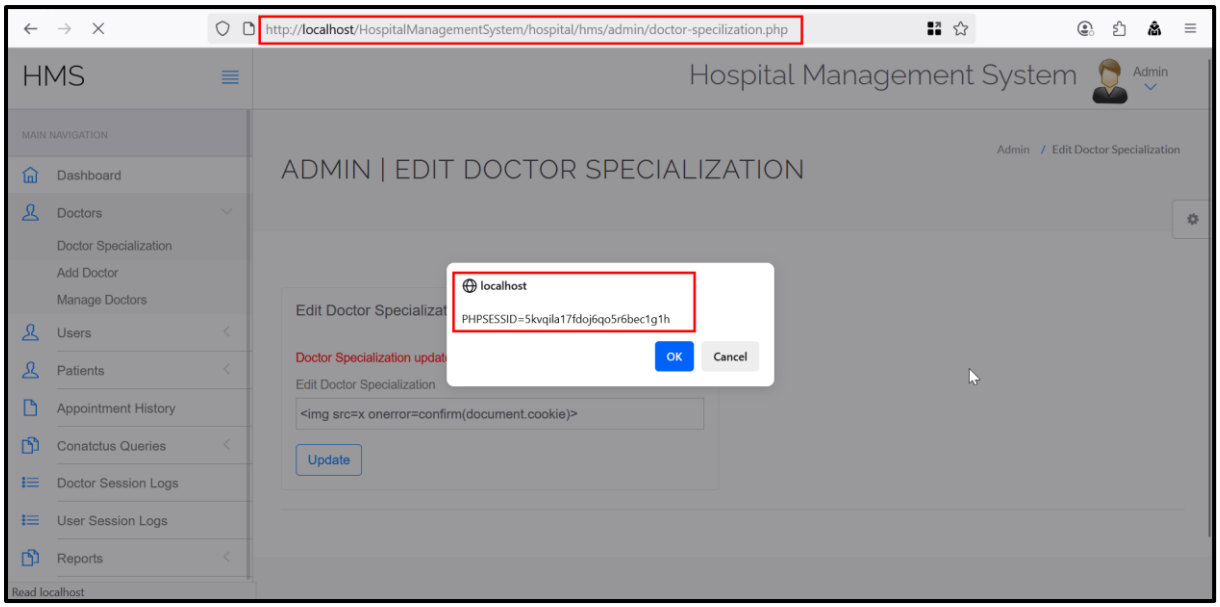
Payload: "*<img src=x onerror=confirm(document.cookie)>*"





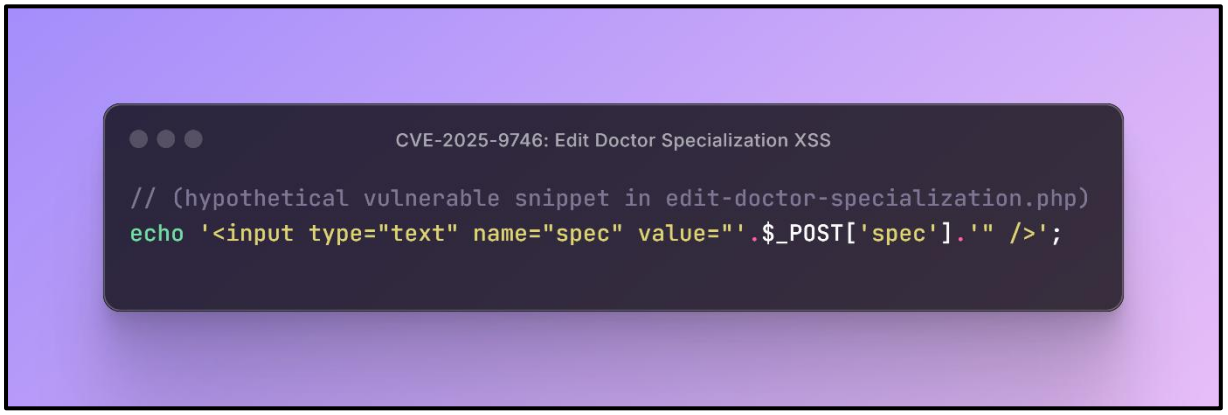
Step 5: After successfully updating the doctor specialization, click on "Doctors" and then select "Doctor Specialization." It can be observed that the payload executes successfully, indicating that the application does not properly sanitize user input, resulting in Stored Cross-Site Scripting (XSS).





Observation:

In Campcodes Hospital Management System 1.0, the "Edit Doctor Specialization" page (*/admin/edit-doctor-specialization.php*) is vulnerable to XSS. The CVE summary states that user manipulation causes cross-site scripting. Although the exact code isn't in the provided files, this typically means a user-input field (e.g. a specialization name) is output without escaping. For example, if the code contained:



without using htmlspecialchars, then entering a value like "><script>malicious()</script> would inject script into the page. The description "manipulation results in cross site scripting" cvedetails.com confirms this.

- **Unsanitized field:** Likely the specialization name ($_POST['spec']) is not sanitized before being placed in HTML.
- **Consequence:** Injected scripts execute in the context of any user viewing that page (stored XSS).
- **Mitigation:** Always escape output (e.g. htmlspecialchars($_POST['spec'], ENT_QUOTES)) and validate input. Use parameterized queries for any database updates as well.

References:

https://nvd.nist.gov/vuln/detail/CVE-2025-9746
https://www.cvedetails.com/cve/CVE-2025-9746/
https://www.cve.org/CVERecord?id=CVE-2025-9746
https://www.tenable.com/cve/CVE-2025-9746