**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐH SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

****

**Báo Cáo Cuối Kỳ**

Môn học: Đồ án 2 – Xây dựng website

chứa lỗ hổng bảo mật

GVHD: Thầy Nguyễn Bá Tiến

SVTH: Phạm Hồng Cang - 15110014

Lê Minh Chương - 15110020

Phạm Gia Bảo Đại - 15110029

***TP.HCM, ngày 22 tháng 5 năm 2018***

MỤC LỤC

[LỜI MỞ ĐẦU 2](#_Toc515105332)

[I.Union based SQL injection 3](#_Toc515105333)

[1. Định nghĩa 3](#_Toc515105334)

[2. Demo tấn công 4](#_Toc515105335)

[3. Cách phòng chống 6](#_Toc515105336)

[II.File upload 6](#_Toc515105337)

[1. Định nghĩa 6](#_Toc515105338)

[2. Demo tấn công 7](#_Toc515105339)

[3. Cách phòng tránh 8](#_Toc515105340)

[III.DOM based XSS 10](#_Toc515105341)

[1. Định nghĩa 10](#_Toc515105342)

[2. Demo tấn công 11](#_Toc515105343)

[3. Cách phòng tránh 15](#_Toc515105344)

[IVBlind SQL injection 16](#_Toc515105345)

[1. Định nghĩa 16](#_Toc515105346)

[2. Demo tấn công 16](#_Toc515105347)

[3. Cách phòng tránh 17](#_Toc515105348)

[V.SQL Injection in INSERT Query 18](#_Toc515105349)

[1. Định nghĩa 18](#_Toc515105350)

[2. Demo tấn công 18](#_Toc515105351)

[3. Cách phòng chống 19](#_Toc515105352)

[VI.Cross-site Scripting – Reflected XSS 21](#_Toc515105353)

[1. Định nghĩa 21](#_Toc515105354)

[2. Demo tấn công 21](#_Toc515105355)

[3. Cách phòng chống 23](#_Toc515105356)

[VII.Cross-site Scripting – Stored XSS 24](#_Toc515105357)

[1. Định nghĩa 24](#_Toc515105358)

[2. Demo tấn công 24](#_Toc515105359)

[3. Cách phòng chống 27](#_Toc515105360)

[PHẦN KẾT LUẬN 28](#_Toc515105361)

# LỜI MỞ ĐẦU

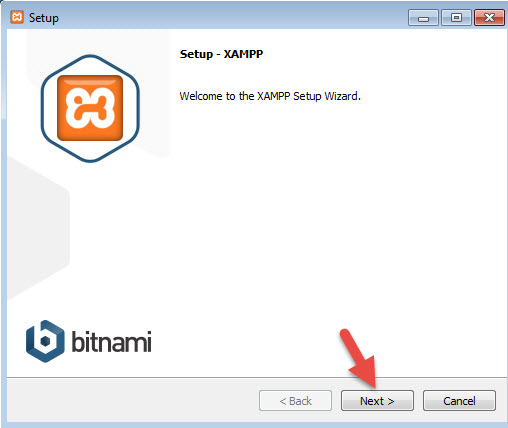
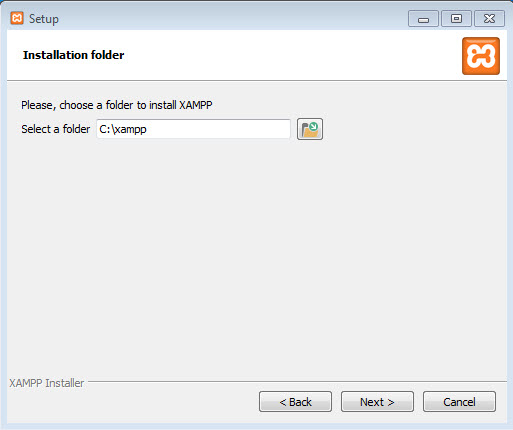
Trong thời đại ngày này nay internet đã rất trở nên quen thuộc và là một công cụ hữu ích để một đất nước giới thiệu hình ảnh hay đơn giản chỉ là một trang web cá nhân của một ai đó giới thiệu về mình. Tất cả đã kéo theo sự phát triển không ngừng của các ứng dụng web. Và dần dần khái niệm ứng dụng web đã trở nên phổ biến .Khi mà trên internet, ứng dụng web đã trở lên phổ biến ,ứng dụng một cách rộng rãi thì các cuộc tấn công ứng dụng web cũng phát triển hết sức phức tạp. Điều này đã đặt ra vấn đề cấp thiết cần làm như thế nào để bào đảm an toàn thông tin cho ứng dụng web, thông tin của người sử dụng. Các khái niệm chuyên môn về ứng dụng web và tấn công ứng dụng web cũng dần trở nên phổ biến hơn trong các tài liệu chuyên ngành. Các công cụ hỗ trợ người lập trình web, người quản trị mạng cũng xuất hiện giúp tìm kiếm lỗ hổng của ứng dụng web nhưng nó không theo kịp sự phát triển nhanh đến mức chóng mặt theo xu hướng nhanh hơn đẹp hơn của các ứng dụng web, và tất nhiên nó không thể ngăn chặn hoàn toàn các cuộc tấn công ứng dụng web, khi mà các cuộc tấn công ngày càng đa dạng khai thác triệt để những lỗi của ứng dụng web, của người quản trị, hay người lập trình ứng dụng web.

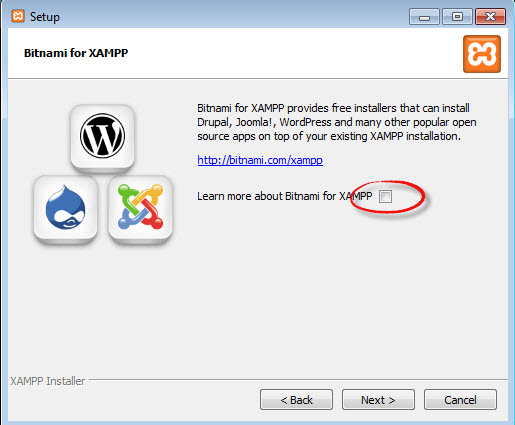
Thống kê cho thấy 75% cuộc tấn công internet là tấn công ứng dụng web ,nó gây ra những thiệt hại vô cùng to lớn, vì vậy việc tìm hiểu về tấn công ứng dụng web là rất cần thiết nhằm có cách phòng chống tấn công và bảo mật ứng dụng web hiệu quả trở thành một yêu cầu cấp thiết…

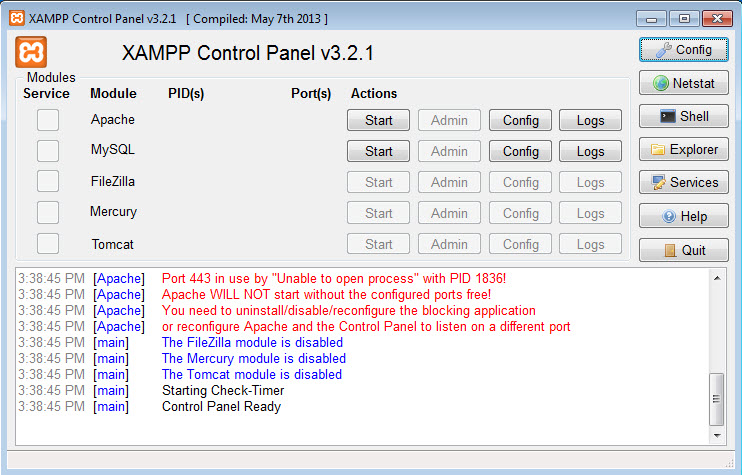
Do đây là một xu thế tất yếu của thời, nên việc tìm hiểu và nghiên cứu về bảo mật ứng dụng web sẽ giúp ích rất nhiều cho các nhà lập trình web mới, hay các quản trị viên mới còn ít kinh nghiệm trong việc quản trị hệ thống mạng của mình, phòng tránh, hay khắc phục những lỗi của ứng dụng web. Bài tập lớn này được thực hiện nhằm mục đích tìm hiểu về những tấn công ứng dụng web phổ dụng nhằm có cách phòng chống, bảo mật cho ứng dụng web hợp lý.

Hầu hết trong số chúng ta ai cũng đều đã từng để mất dữ liệu và điều này là không thể tránh khỏi. Không một ai có thể dám khẳng định được những dữ liệu này sẽ không bao giờ mất cả. Hầu hết người dùng đều tin rằng việc lưu trữ các dữ liệu quan trọng trong ổ cứng gắn ngoài hoặc ổ đĩa Flash di động sẽ an toàn hơn. Nhưng dù có phần cứng dự phòng đặt ở đó thì các dữ liệu ban đầu vẫn ở trong tình trạng nguy hiểm. Đây chính là lý do dịch vụ sao lưu trực tuyến rất được các chuyên gia và chủ doanh nghiệp ưa chuộng. Bằng cách này thì dữ liệu không chỉ an toàn mà còn rất tiện ích bởi tính di động được. Do đó bạn có thể quản lý, xem hoặc tải xuống từ bất cứ nơi nào trên thế giới. Trong bài tập lớn này, chúng em đã xây dựng hai trang web cung cấp dịch vụ chuyên nghiệp cung cấp sao lưu dữ liệu trực tuyến, một trang web chứa các lỗi bảo mật và trang web còn lại thì không.

# HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG TRANG WEB

* Bước 1: Vào trang <https://www.apachefriends.org/index.html> để tải bản phù hợp với hệ điều hành của bạn.
* Bước 2: Sau khi tải file cài đặt về xong, hãy chạy nó, sau đó chọn **Next**.
* Bước 3: Ở phần chọn đường dẫn, bạn hãy chọn đường dẫn cần lưu cài đặt của XAMPP. Lưu ý rằng đường dẫn này bạn phải nhớ vì khi cài đặt web lên localhost, bạn phải truy cập vào thư mục này. Bạn nên để mặc định là **c:\xampp**. Tiếp tục ấn Next.
* Bước 4: Ở trang kế tiếp, bạn bỏ chọn phần “Learn more about Bitnami for XAMPP“. Và ấn Next 2 lần nữa để bắt đầu quá trình cài đặt XAMPP.



* Bước 5: Sau khi cài xong, ấn nút **Finish** để kết thúc cài đặt và mở bảng điều khiển của XAMPP. Tuy nhiên, hãy khởi động lại máy sau khi cài đặt xong để tránh tình trạng không khởi động được localhost.
* KHỞI ĐỘNG LOCALHOST
* Bước 1: Bây giờ bạn hãy vào thư mục **c:\xampp** và mở file**xampp-panel.exe** lên để bật bảng điều khiển của XAMPP.
* 

# Union based SQL injection

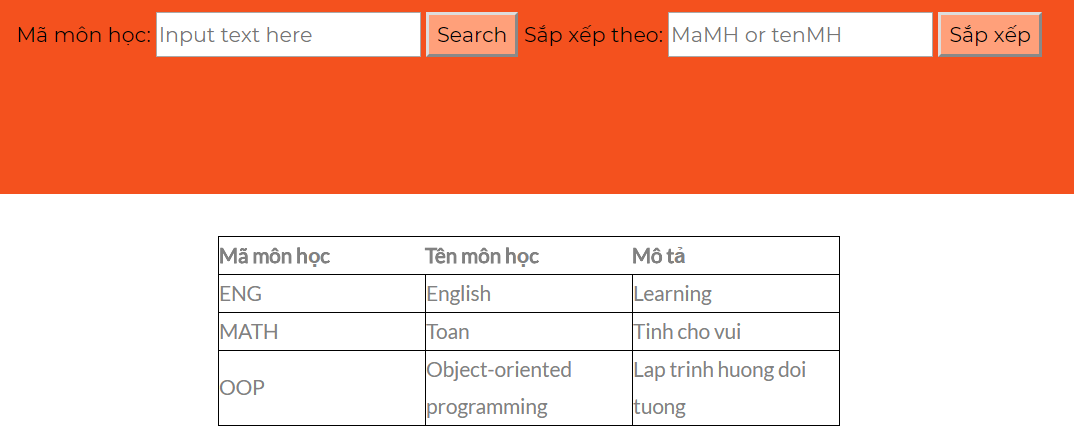
## Định nghĩa

Trước tiên, chúng ta cần phải hiểu về Sql Injection. Đó là một một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp, Sql Injection có thể cho phép những kẻ tấn công thực hiện các thao tác, thêm, sửa, xóa… trên cơ sở dữ liệu của ứng dụng. Lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase...

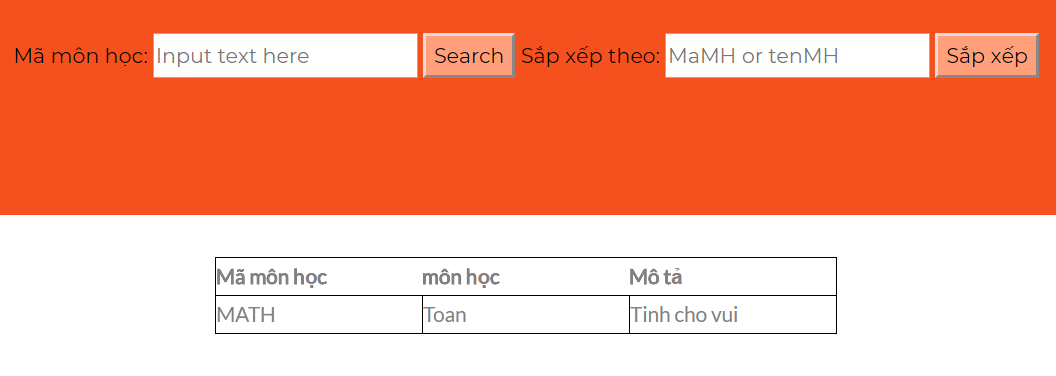
Union based là phương pháp phổ biến khi khai thác Sql injection. Cơ sở của nó là sử dụng từ khóa union để gộp các kết quả của các mệnh đề select, qua đó lấy được thông tin từ cơ sở dữ liệu.

UNION được sử dụng để nối thêm query SQL injection đến một truy vấn hợp pháp và kết hợp các thông tin chúng ta muốn lấy với các truy vấn hợp pháp. Lưu ý rằng bạn cần phải liệt kê các cột đầu tiên, điều này có thể đạt được bằng cách sử dụng các chức năng ORDER BY hoặc sử dụng UNION với các giá trị NULL.

## Demo tấn công

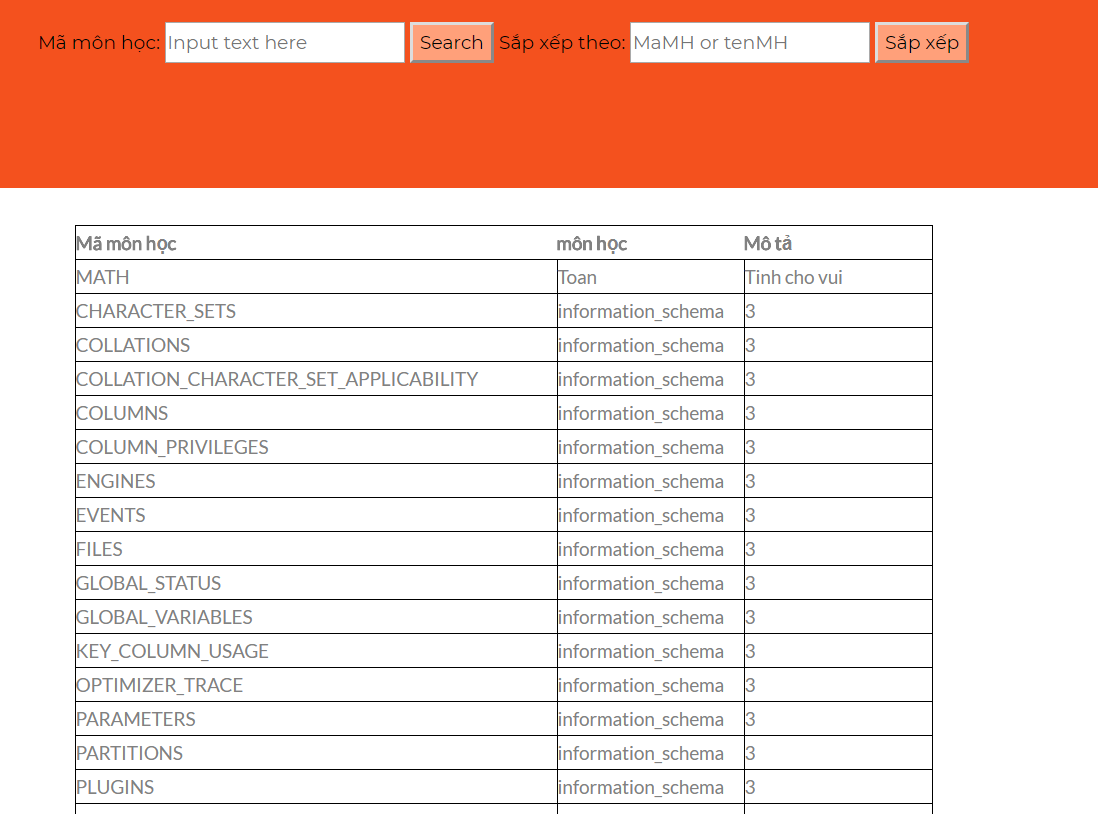
Tìm môn học theo mã môn học:

Source file: C:\wamp64\www\doan\errorweb\monhoc.php

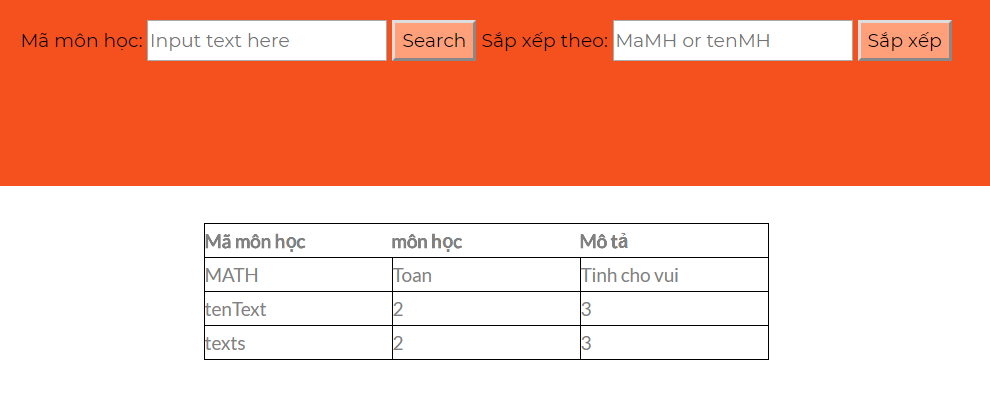
Tìm kiếm với “MATH”:

Hacker sẽ lợi dụng điểm này để khai thác với lệnh:

MATH' UNION (SELECT TABLE\_NAME, TABLE\_SCHEMA, 3 FROM information\_schema.tables);-- -

Để kiểm tra thông tin Table và Tên cơ sở dữ liệu

Kiểm tra các cột trong Table mới tìm được ở trên:

MATH' UNION (SELECT COLUMN\_NAME, 2, 3 FROM information\_schema.columns WHERE TABLE\_NAME='ghichu');-- -

Và nhiều đoạn mã khai thác khác…

## Cách phòng chống

Source file: C:\wamp64\www\doan\normalweb\monhoc.php

Sử dụng đoạn lệnh để mã hóa chuỗi trong ô input thành kiểu String giúp cơ sở dữ liệu hiểu đúng về câu lệnh truy vấn:

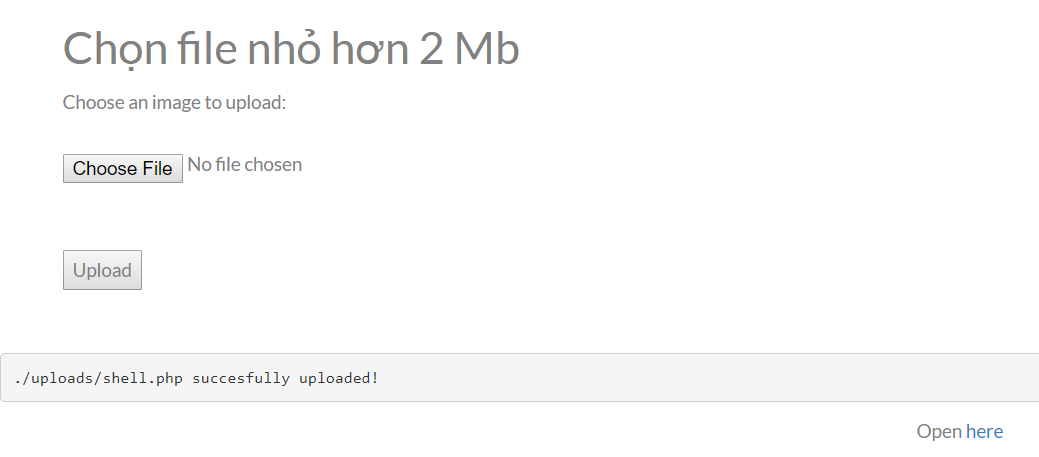
# File upload

## Định nghĩa

**Các form upload cho phép người dùng đầu cuối upload file lên trang web của bạn cũng giống như việc mở một của hậu cho tin tặc xâm nhập máy chủ của bạn. Mặc dù vậy, trong ứng dụng web hiện nay, upload là một nhu cầu phổ biến bởi nó giúp tương tác trở nên hiệu quả hơn. Upload file được cho phép trong các ứng dụng mạng xã hội như Facebook và Twitter. Các diễn đàn, blog, Youtube, trang web thương mại điện tử hay cổng hỗ trợ trực tuyến cũng sử dụng tính năng này nhằm cho phép người dùng đầu cuối chia sẻ file với nhân viên công ty. Người dùng được phép tải lên hình ảnh, video, ảnh đại diện và nhiều khiểu file khác.**

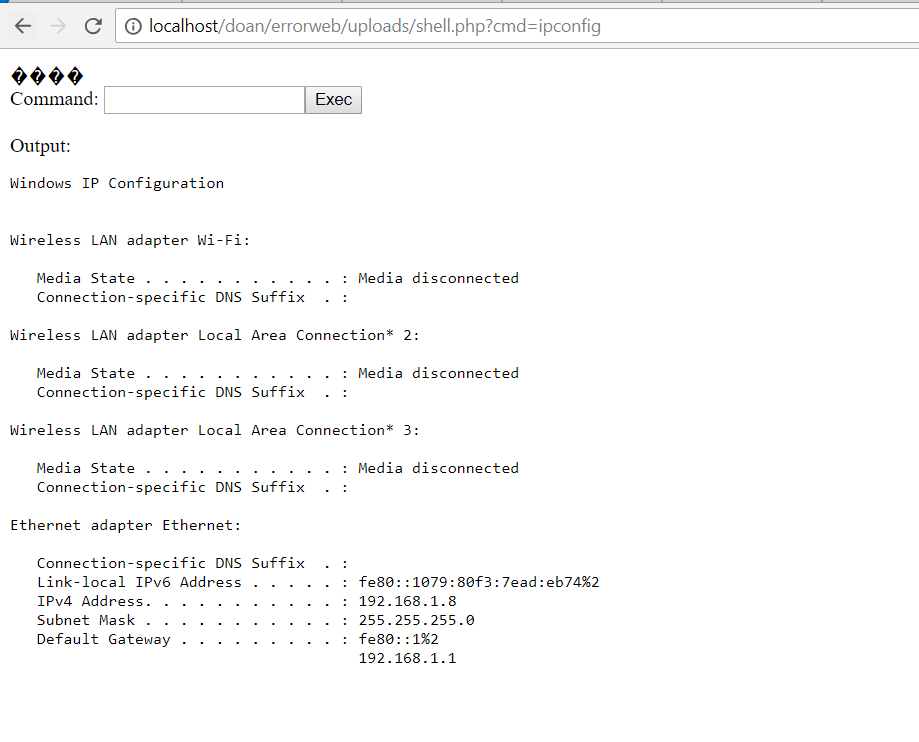
Nhiều chức năng cung cấp cho người dùng sẽ càng ẩn chứa nhiều nguy cơ tồn tại lỗ hổng và những lỗ hổng này sẽ bị tin tặc lợi dụng nhằm đoạt quyền truy cập vào trang web hay xâm nhập vào máy chủ.

## Demo tấn công

Ở đây ta tải lên file php và truy suất file này:

Source file : C:\wamp64\www\doan\errorweb\fileupload.php

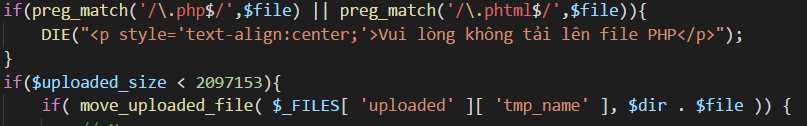
Sau khi mở file ra tại button here, ta sẽ có web:



Tại web này Hacker có thể làm rất nhiều tác vụ gây hại cho server.

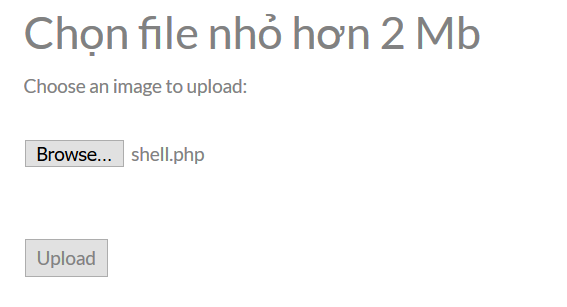
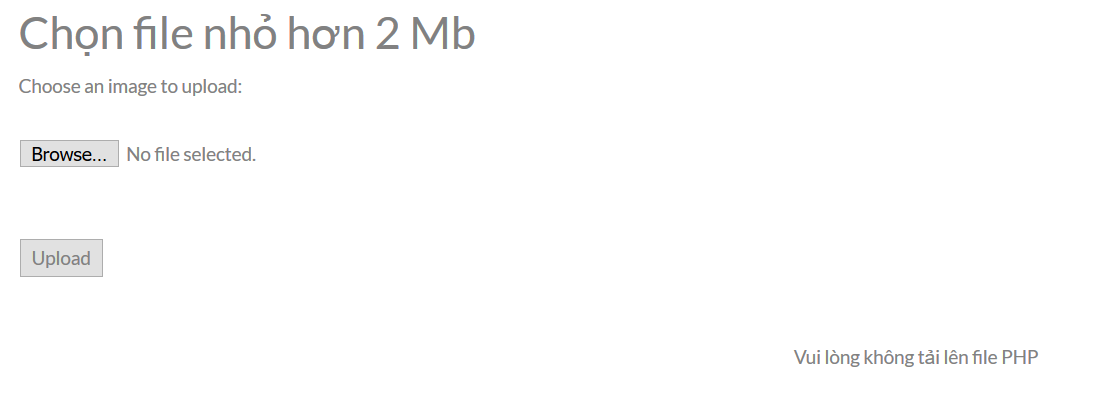
## Cách phòng tránh

* Xác định một file .htaccess chỉ cho phép truy cập file với phần mở rộng được phép.
* Không đặt file .htaccess trong cùng địa chỉ với file upload.
* File .htaccess chỉ cho phép tải gif, jpg, jpeg, và png nên chứa lệnh dưới đây để đề phòng phần mở rộng kép.
* Nếu có thể, tải file lên một địa chỉ nằm ngoài gốc của máy chủ web.
* Ngăn ngừa ghi đè những file đã tồn tại.
* Tạo một danh sách mime-type được chấp nhận.
* Tạo ra tên file ngẫu nhiên và thêm vào sau đó phần mở rộng.
* Không nên phụ thuộc vào kiểm duyệt phía client, điều đó là không an toàn. Cần kiểm duyệt cả từ phía máy chủ.

Code phòng chống của nhóm:

Souce file: C:\wamp64\www\doan\normalweb\fileupload.php

Chúng ta kiểm tra định dạng file và dung lượng của file, nếu đúng yêu cầu thì file sẽ được tải lên còn không thì ngược lại.

Ở đây chúng ta tải lại file shell.php như demo trước, sẽ có thông báo:

# DOM based XSS

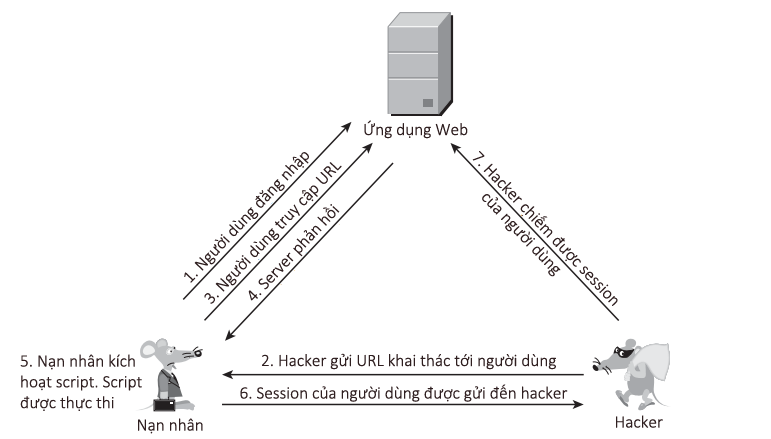
## Định nghĩa

Cross-Site Scripting hay còn được gọi tắt là XSS (thay vì gọi tắt là CSS để tránh nhầm lẫn với CSS-Cascading Style Sheet của HTML) là một kĩ thuật tấn công bằng cách chèn vào các website động (ASP, PHP, CGI, JSP ...) những thẻ HTML hay những đoạn mã script nguy hiểm có thể gây nguy hại cho những nạn nhân sử dụng.

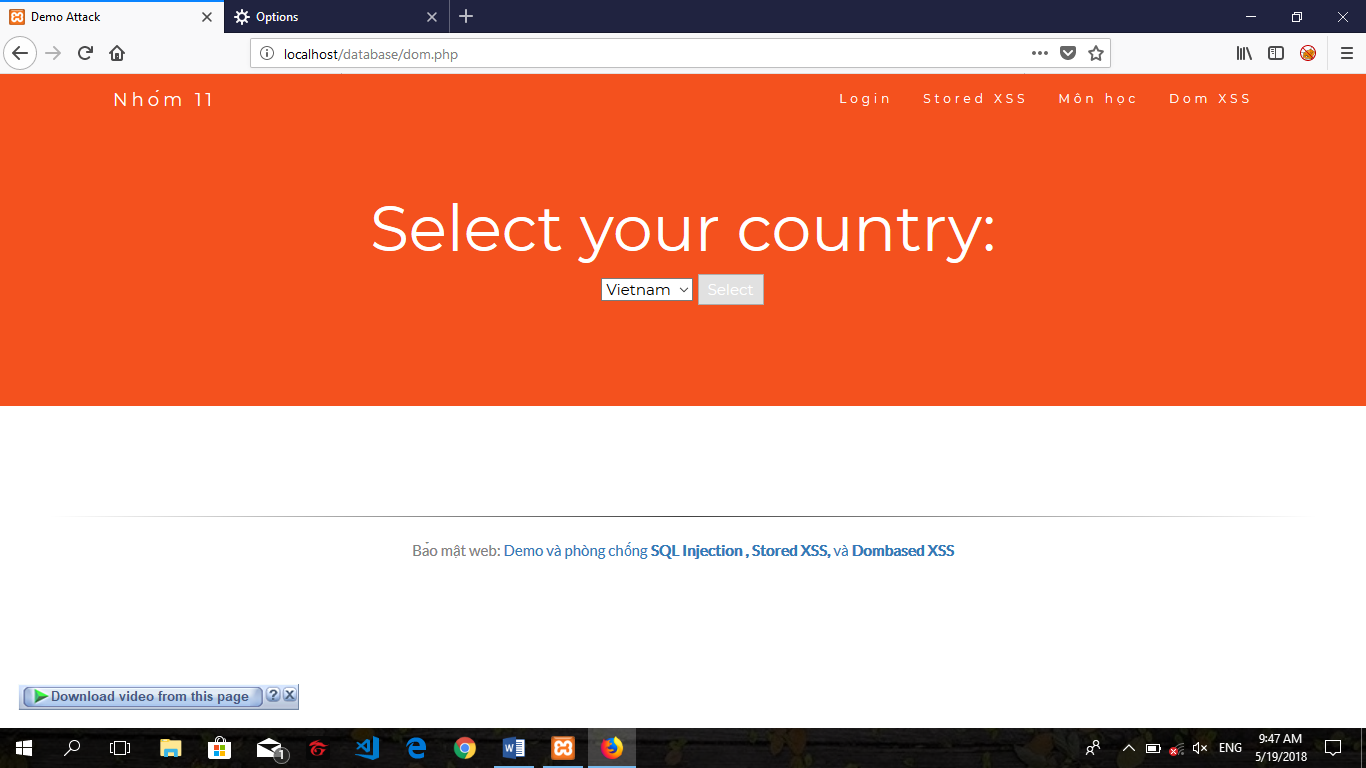
DOM viết tắt của Document Object Model là 1 dạng chuẩn của [W3C](http://www.w3.org/DOM/) đưa ra nhằm để truy xuất và thao tác dữ liệu của tài liệu có cấu trúc như HTML, XML. Mô hình này thể hiện tài liệu dưới dạng cấu trúc cây phân cấp. Tất cả các thành phần trong HTML, XML đều được xem như một node.

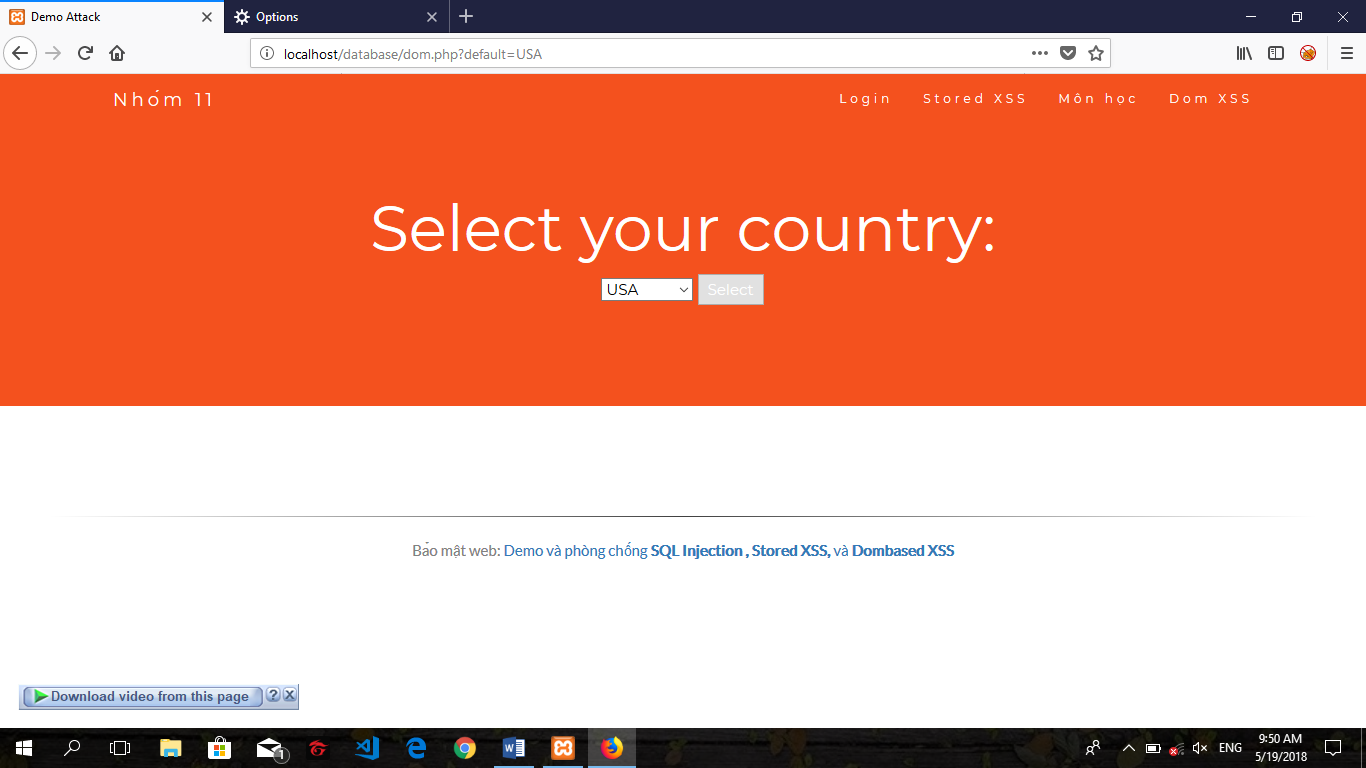
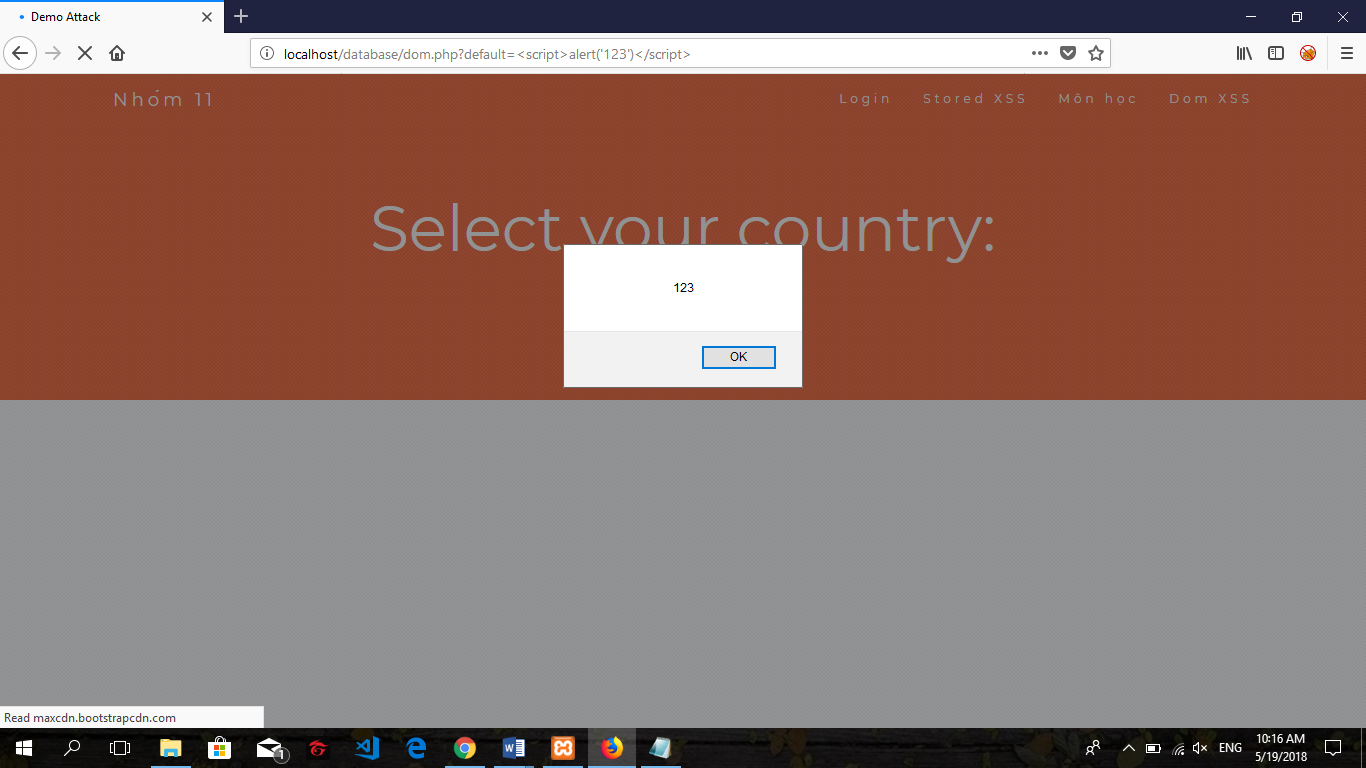
DOM Based XSS là kỹ thuật khai thác XSS dựa trên việc thay đổi cấu trúc DOM của tài liệu, cụ thể là HTML.

Kịch bản tấn công như sau:

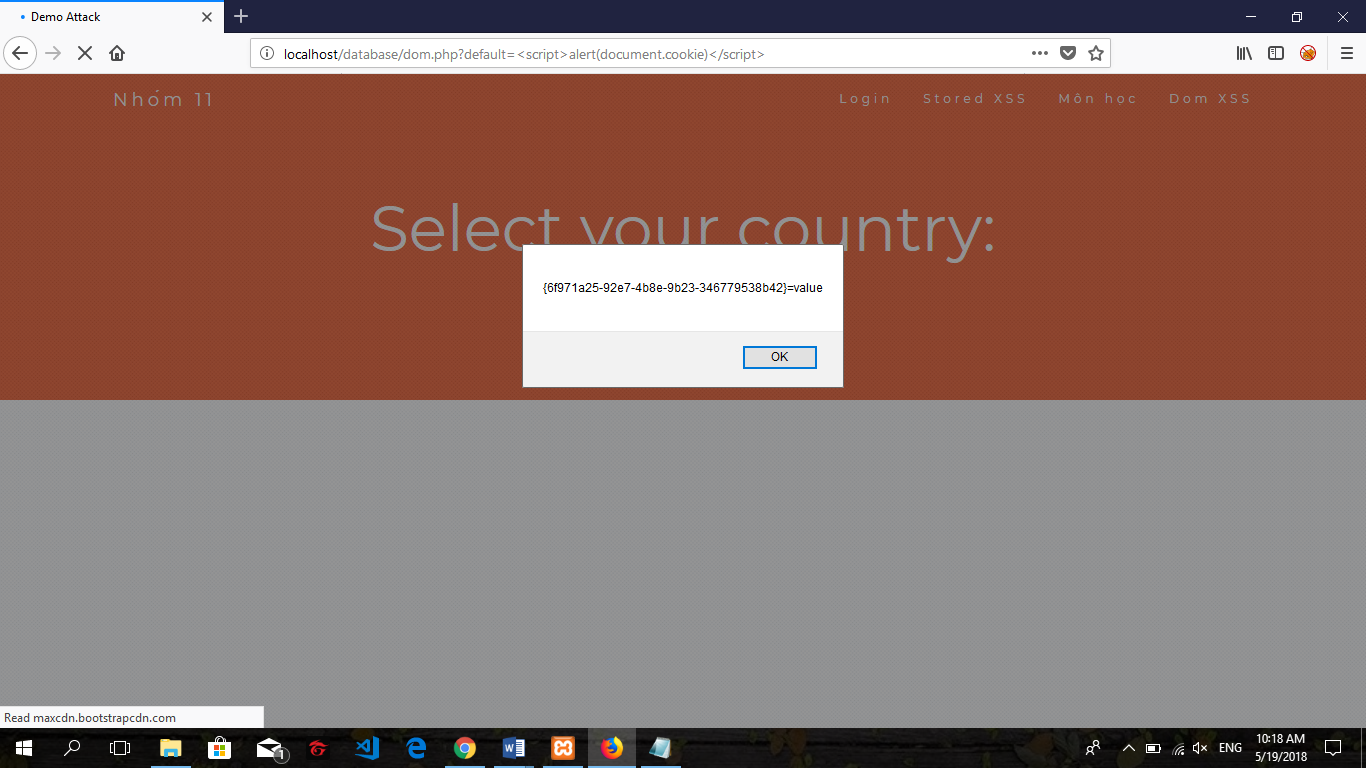
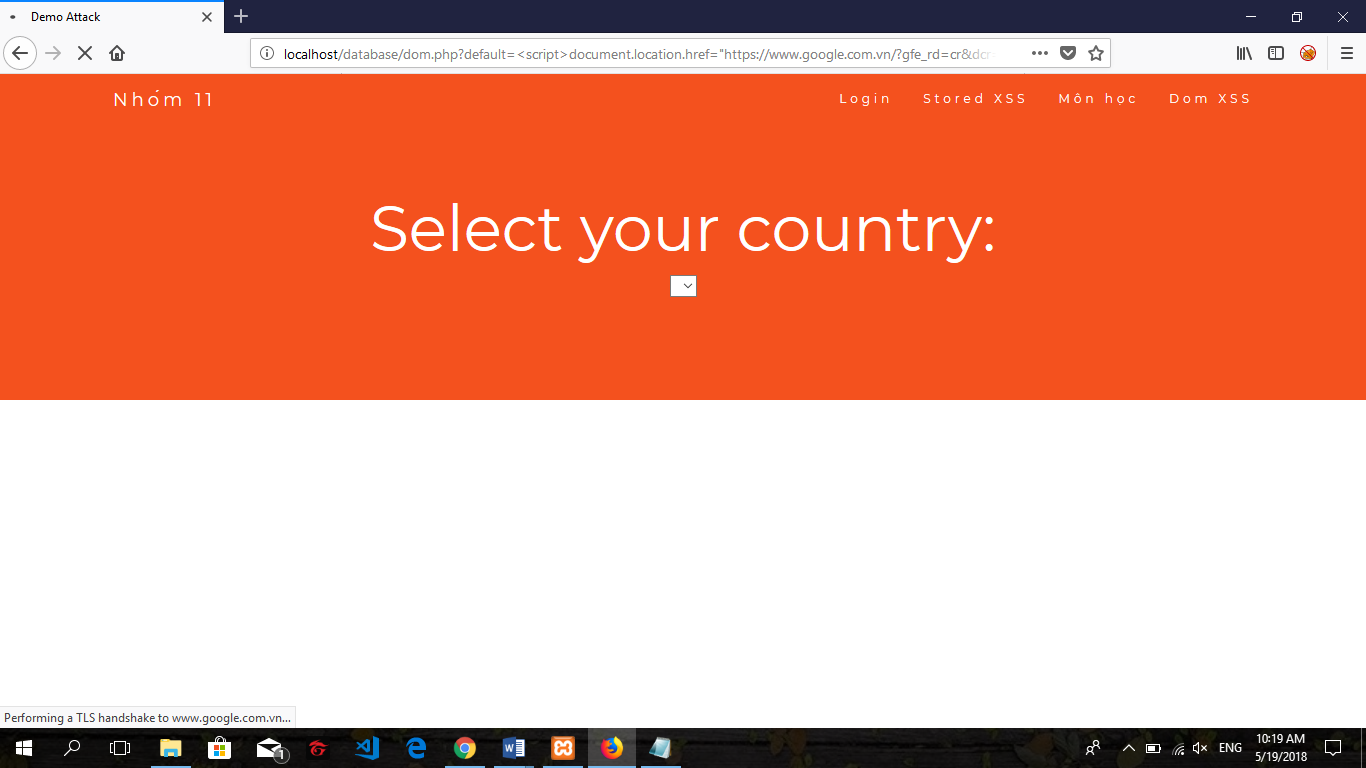


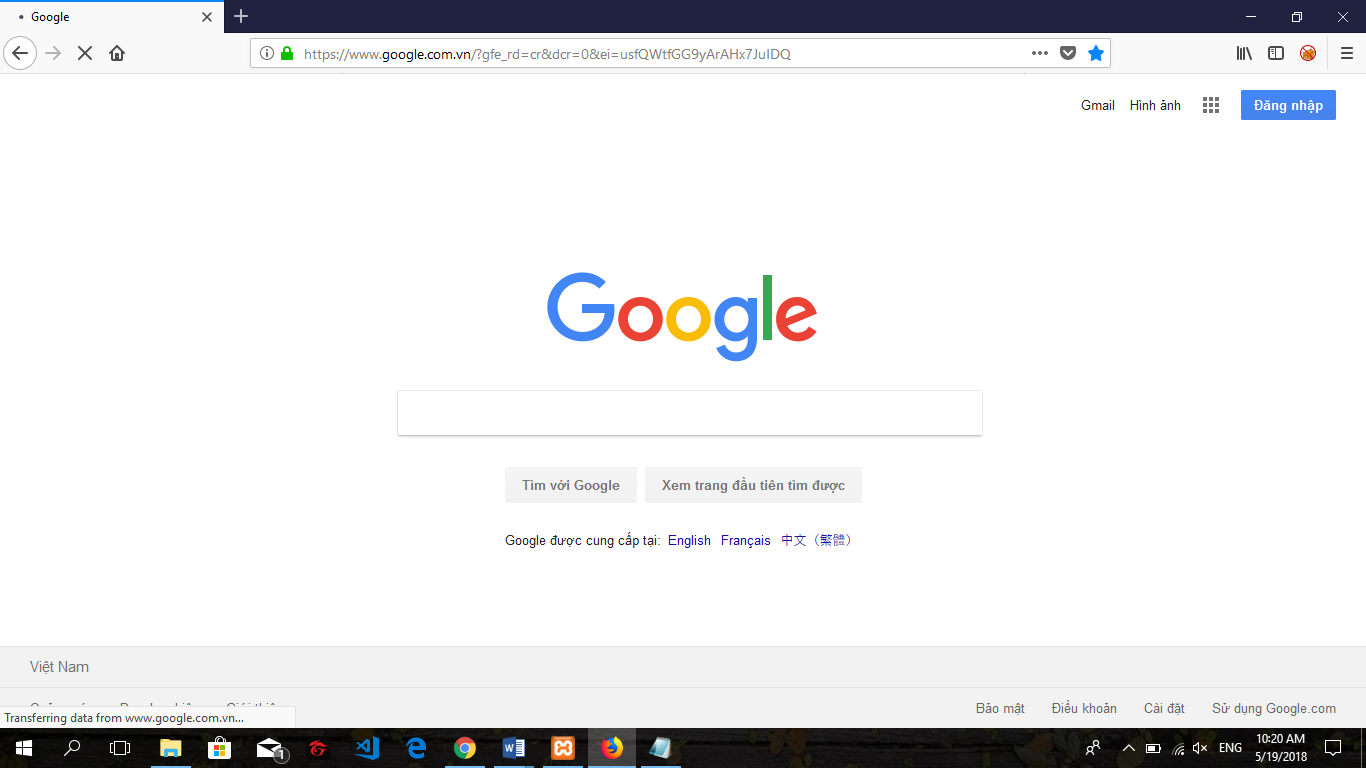
## Demo tấn công

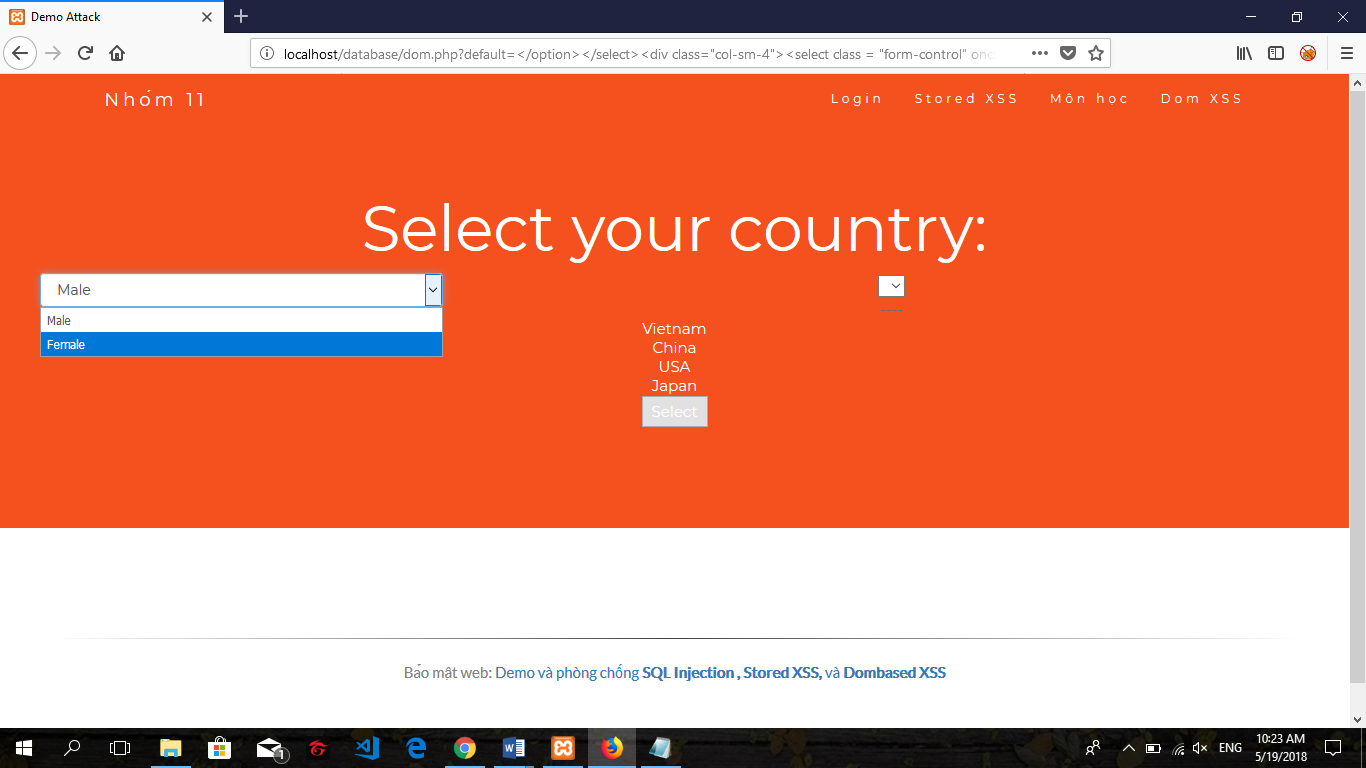
* B1: Truy cập đường dẫn http://localhost/database/dom.php

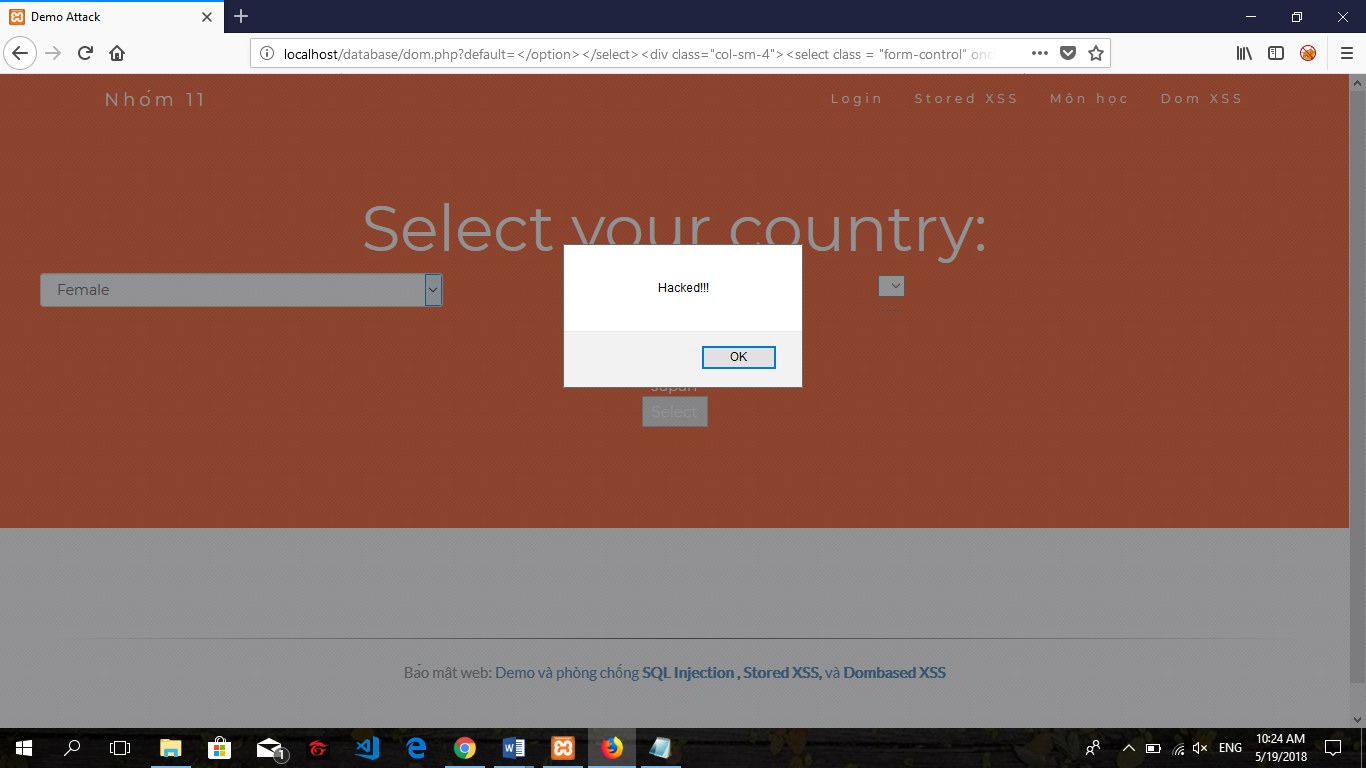
* B2: Sau khi select USA
* B2: Thử thay đổi giá trị sau Default

Vậy là trang web này dính lỗi Dom based XSS.

* B3: Lấy cookie của phiên đăng nhập
* B4: Chuyển hướng trang (redirect)



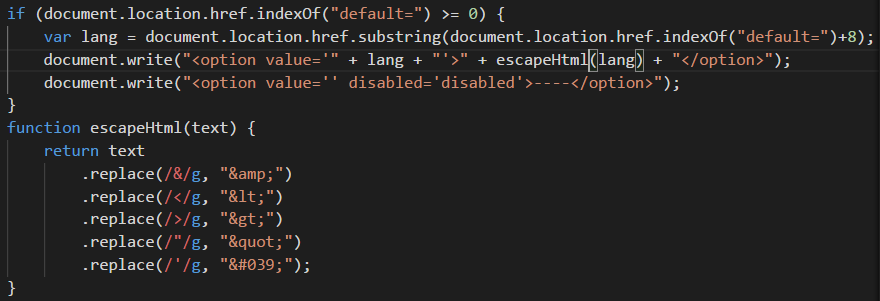
* B5: Tạo form giả đánh lừa người dùng

Ngay khi người dùng select một lựa chọn trong form, sẽ hiện thông báo bị hack:

## Cách phòng tránh

Khi viết chương trình, ta nên viết một hàm để mã hóa những ký tự đặc biệt ở đầu vào mà có thể gây hiểu lầm cho chương trình. Ví dụ như những ký tự: nháy đơn, nháy kép, dấu bé hơn, dấu lớn hơn,...

Source file: C:\wamp64\www\doan\normalweb\dombased.php

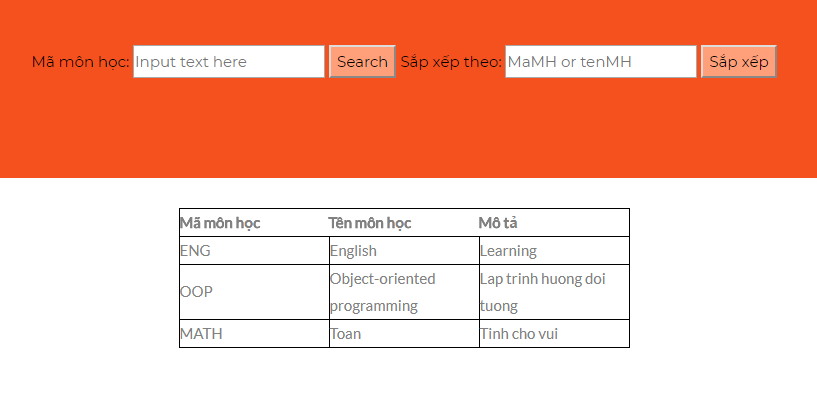


# Blind SQL injection

## Định nghĩa

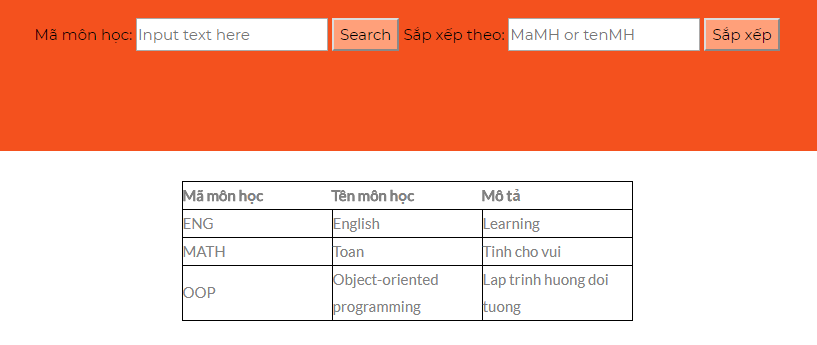
Lỗi SQL injection dạng này là dạng lỗi tồn tại ngay trong ứng dụng web nhưng hậu quả của chúng lại không hiển thị trực quan cho những kẻ tấn công. Nó có thể gây ra sự sai khác khi hiển thị nội dung của một trang chứa lỗi bảo mật này, hậu quả của sự tấn công SQL injection dạng này khiến cho lập trình viên hay người dùng phải mất rất nhiều thời gian để phục hồi chính xác từng bit dữ liệu. Những kẻ tấn công còn có thể sử dụng một số công cụ để dò tìm lỗi dạng này và tấn công với những thông tin đã được thiết lập sẵn.

## Demo tấn công

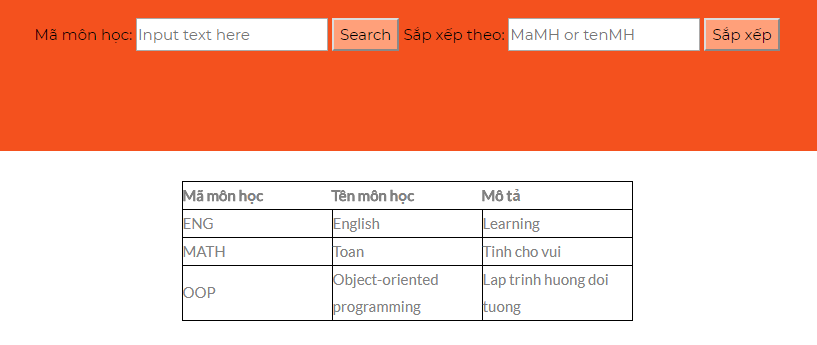
Ta sẽ sử dụng câu lệnh điều kiện sau: if((substr(database(),1,1)='A'),MaMH,tenMH). Mục đích là dò tìm từng kí tự trong tên cơ sở dữ liệu. Ở đây, ta sẽ thử ký tự đầu tiên là ‘A’, nếu đúng thì sẽ sắp xếp trong cột MaMH, sai thì sắp xếp trong cột tenMH.

Source file : C:\wamp64\www\doan\errorweb\monhoc.php

Thử lại bằng dòng lệnh thay chữ ‘A’ bằng ‘d’:

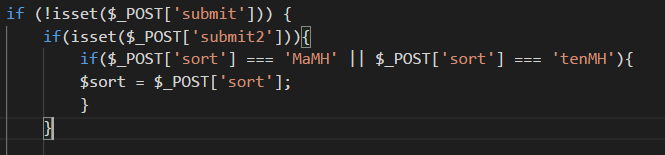


Vậy kí tự đầu tiên trong tên database là ‘d’, ta sẽ thử tiếp với ‘e’ bằng dòng lệnh if((substr(database(),2,1)='e'),MaMH,tenMH).



Quá trình cứ lặp đi lặp lại cho đến khi không còn phép thử nào đúng. Hacker sẽ dò ra được tên database.

## Cách phòng tránh

Lỗi SQL injection xuất phát từ việc lập trình không an toàn, do đó giải pháp tốt nhất để khắc phục điều này là lập trình viên cần thận trọng trong việc phát triển ứng dụng web. Đối với lỗi SQL injection, có thể dùng phương pháp sử dụng Prepare Statement để khắc phục, khi đó dữ liệu đầu vào từ người dùng sẽ không được thực thi trong câu truy vấn. Trong bài tập lớn này, chúng em sẽ sử dụng phương pháp “white list” để khắc phục tấn công. Code khắc phục như sau:

Source file: C:\wamp64\www\doan\normalweb\monhoc.php

Ở đây, chúng ta sẽ xét các giá trị truyền vào, chỉ chấp nhận người dùng nhập vào Mã môn học hoặc Tên môn học.

# SQL Injection in INSERT Query

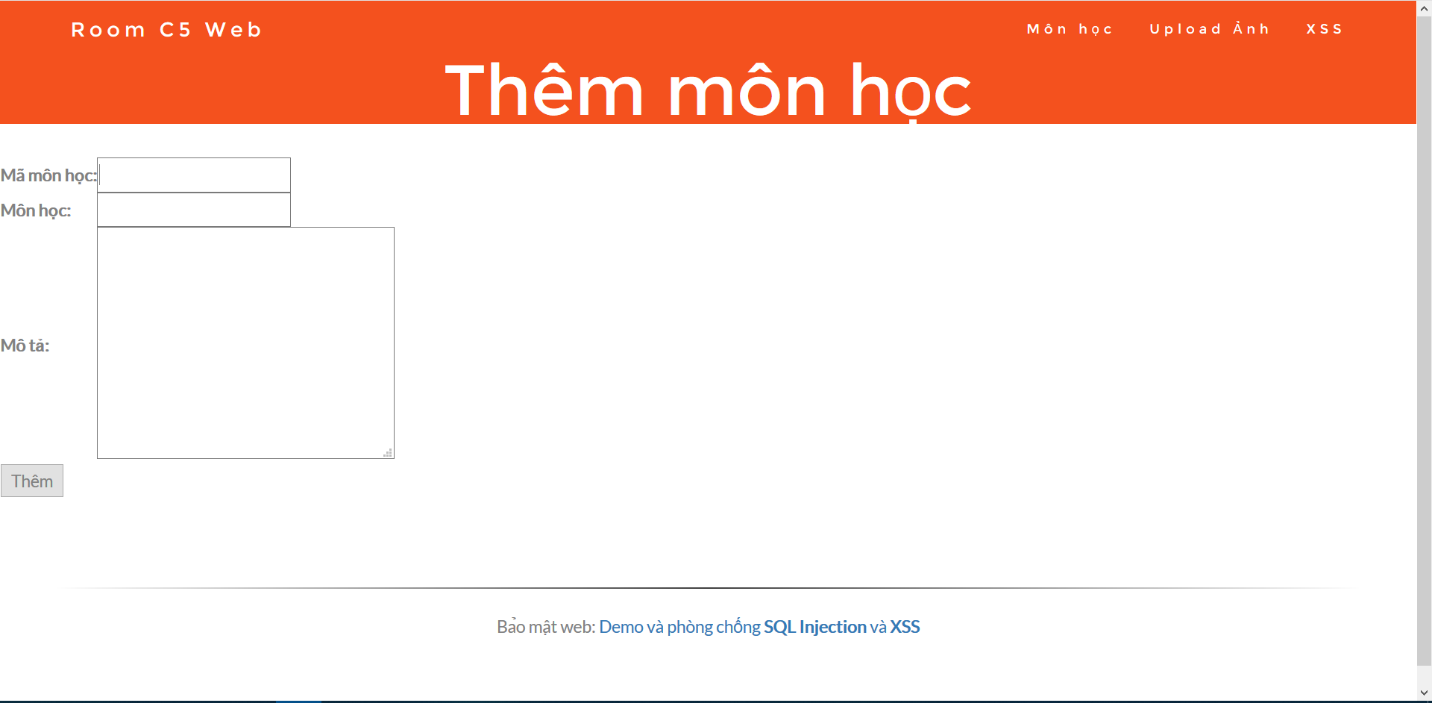
## Định nghĩa

Trước tiên, chúng ta cần phải hiểu về SQL Injection. Đó là một một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào trong các ứng dụng web và các thông báo lỗi của hệ quản trị cơ sở dữ liệu trả về để inject (tiêm vào) và thi hành các câu lệnh SQL bất hợp pháp, Sql Injection có thể cho phép những kẻ tấn công thực hiện các thao tác, thêm, sửa, xóa… trên cơ sở dữ liệu của ứng dụng. Lỗi này thường xảy ra trên các ứng dụng web có dữ liệu được quản lý bằng các hệ quản trị cơ sở dữ liệu như SQL Server, MySQL, Oracle, DB2, Sysbase...

Thông thường các ứng dụng web cho phép người dùng đăng ký một tài khoản để tham gia. Chức năng không thể thiếu là sau khi đăng ký thành công, người dùng có thể xem và hiệu chỉnh thông tin của mình. SQL INSERT Injection có thể được dùng khi hệ thống không kiểm tra tính hợp lệ của thông tin nhập vào.

## Demo tấn công

B1: Ta có một web Thêm môn học bị lỗi SQL INSERT, hacker sẽ lợi dụng thông tin nhập vào để thi hành các câu lệnh SQL bất hợp pháp.

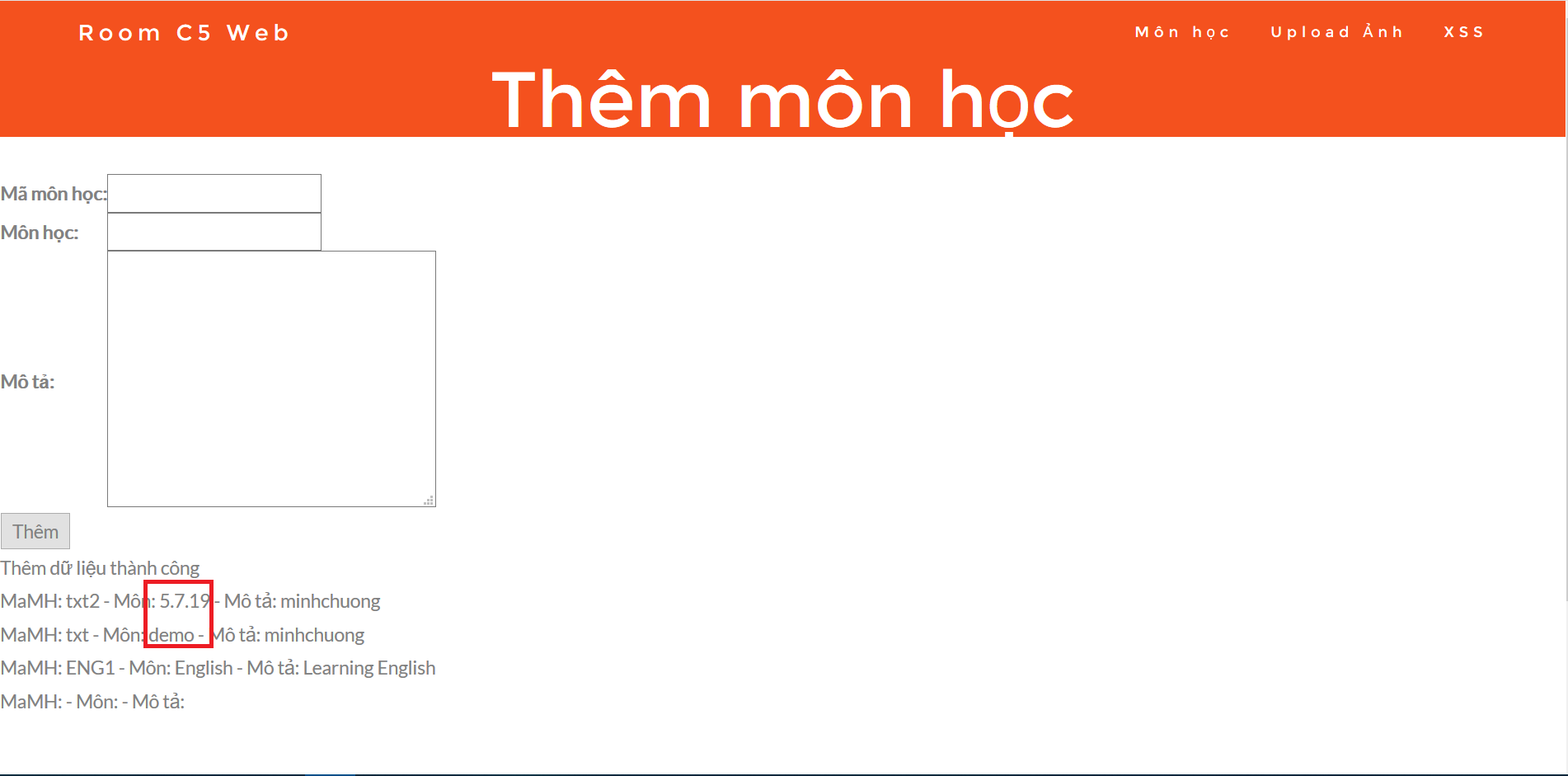
Source file: C:\wamp64\www\doan\errorweb\insertsql.php

B2: Nhập vào ô đầu tiên:

txt',database(),'minhchuong')-- -

txt2',version(),'minhchuong')-- -

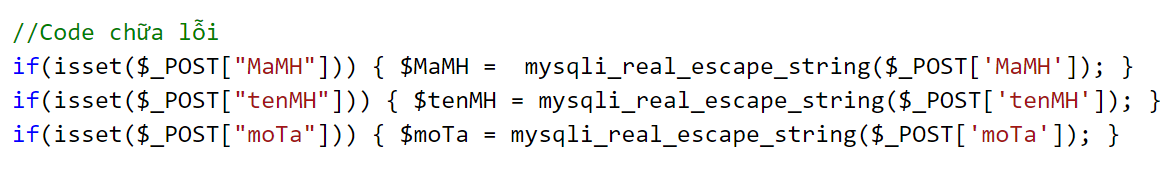
Hacker lấy được tên database và version của database



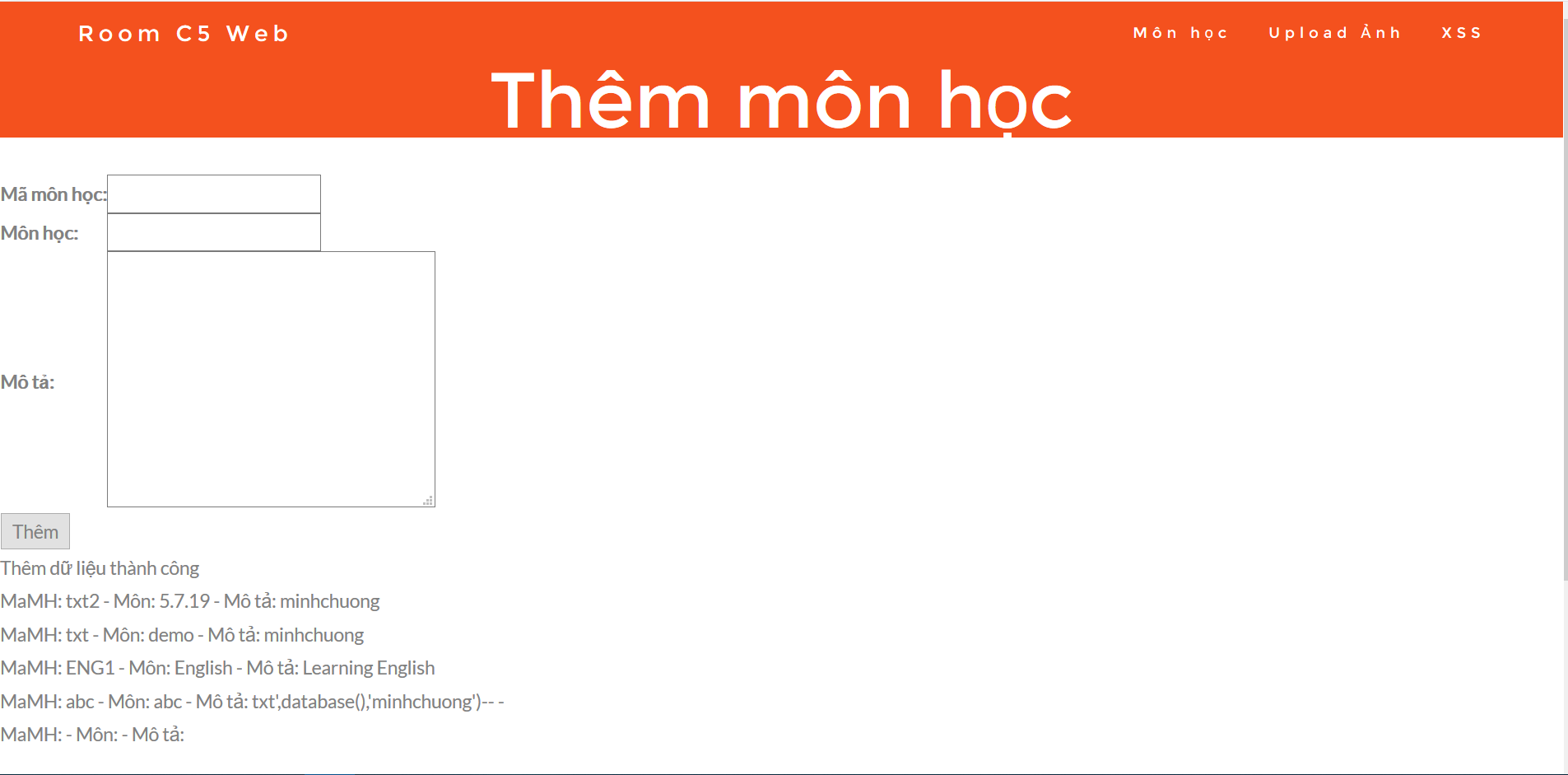
## Cách phòng chống

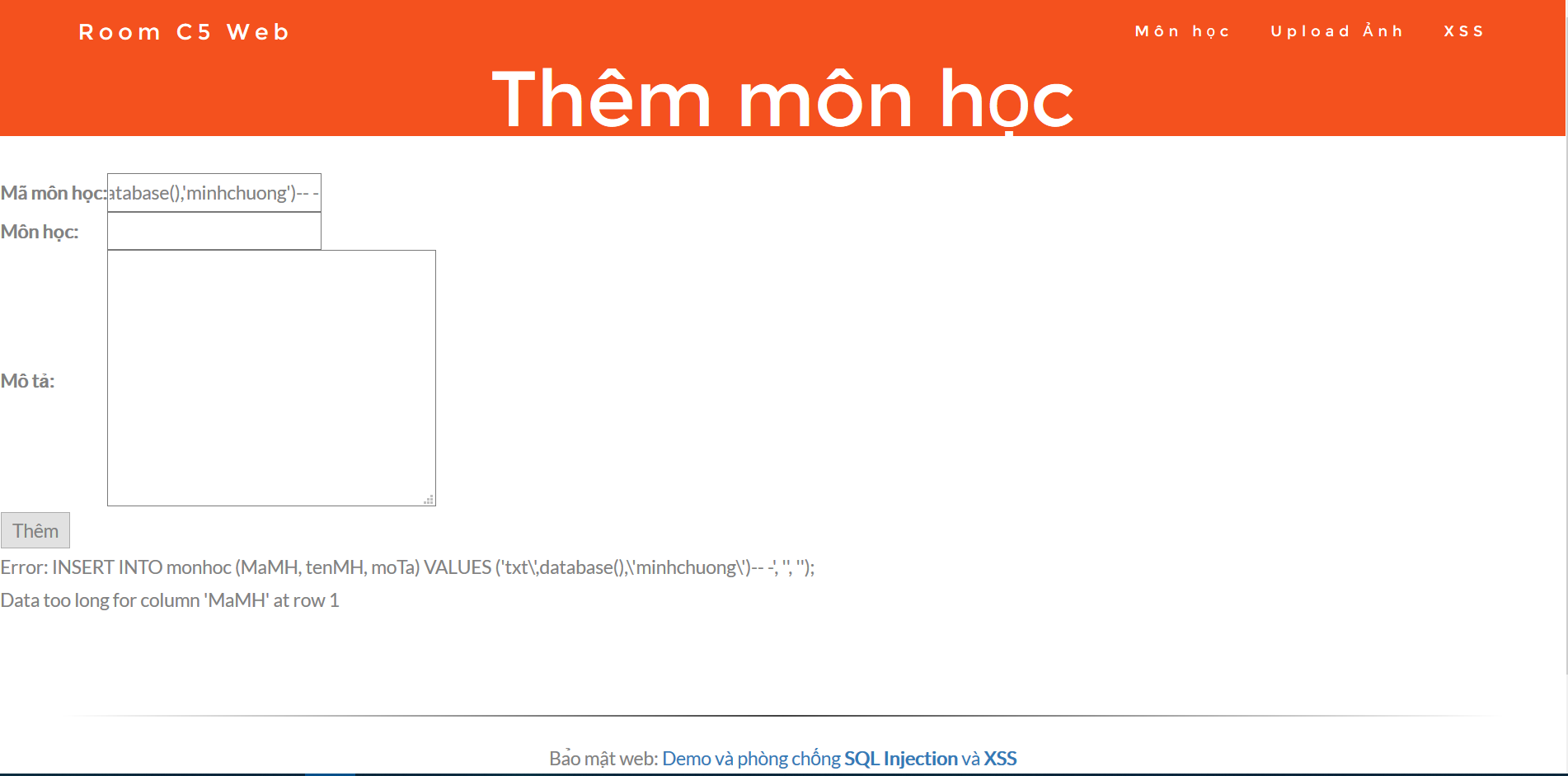
Dùng mysqli\_real\_escape\_string để kiểm tra dữ liệu đầu vào:

Source file: C:\wamp64\www\doan\normalweb\insertsql.php



Kết quả:



Quy định độ dài cho các trường thông tin, hacker sẽ không nhập được truy vấn quá dài:

# Cross-site Scripting – Reflected XSS

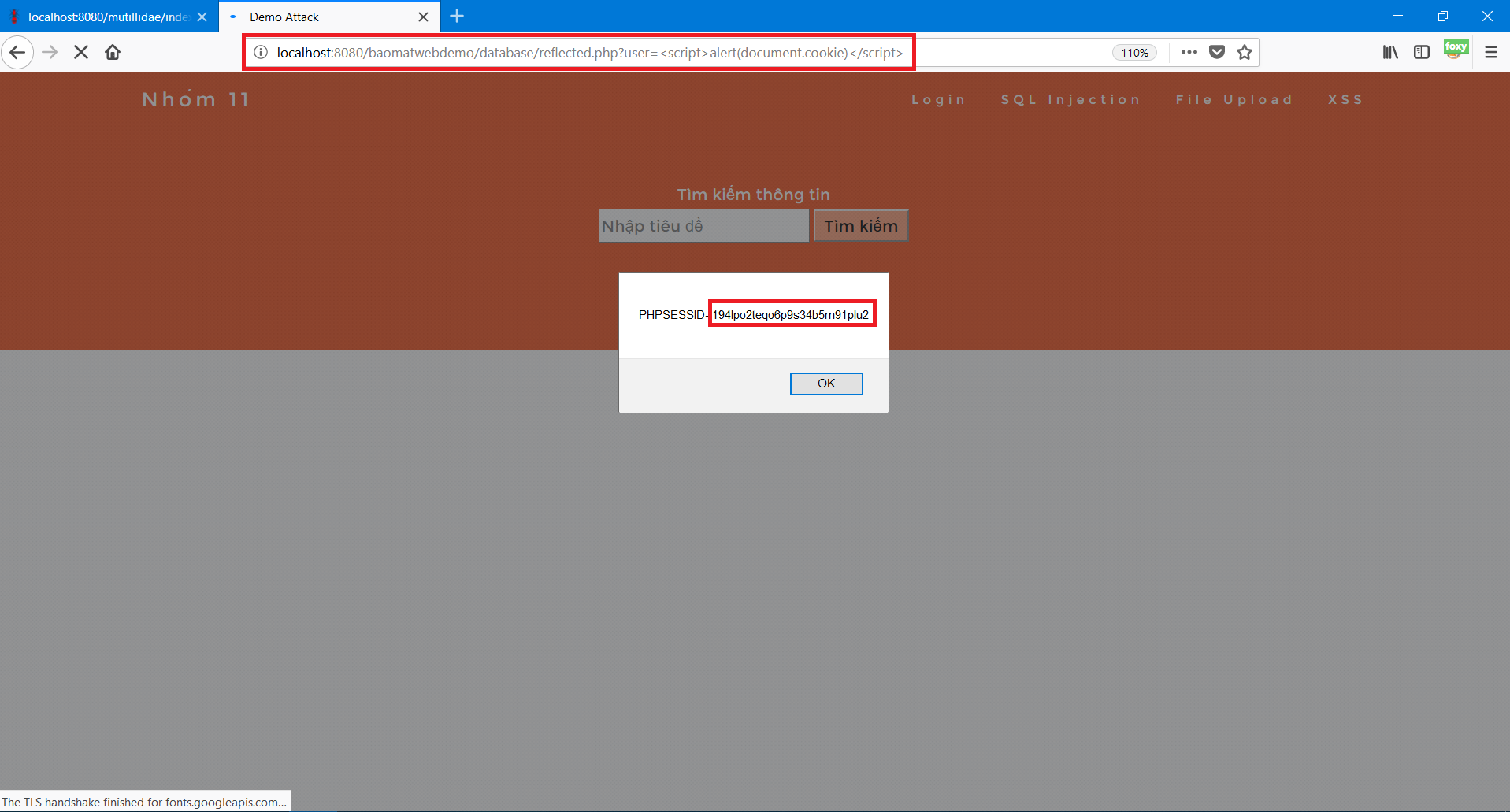
## Định nghĩa

* **Cross-site scripting** là một lỗ hổng phổ biến trong ứng dụng web. Để khai thác một lỗ hổng XSS, hacker sẽ chèn mã độc thông qua các đoạn script để thực thi chúng ở phía client. Thông thường, các cuộc tấn công XSS được sử dụng để vượt qua các kiểm soát truy cập và mạo danh người dùng.
* Với **Reflected XSS**, hacker sẽ gửi cho nạn nhân một URL có chứa đoạn mã nguy hiểm. Nạn nhân chỉ cần gửi request đến URL này thì hacker sẽ có được kết quả mong muốn.

## Demo tấn công

Source file: C:\wamp64\www\doan\errorweb\reflected.php

* Lấy thông tin session (phiên làm việc) của người dùng và thực hiện các quyền mà nạn nhân được gán. Hacker sẽ gửi cho nạn nhân một URL có dạng tương tự như:

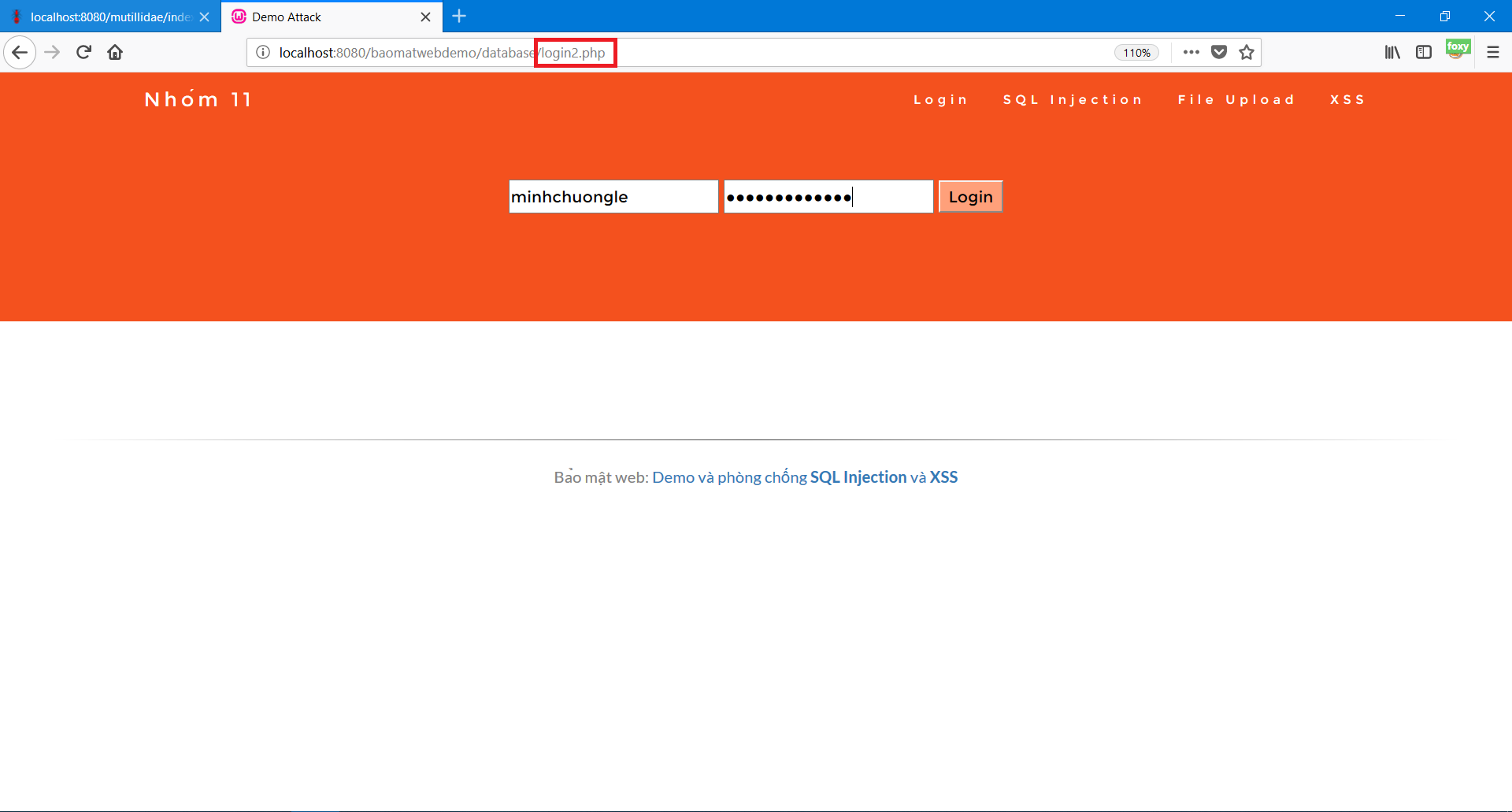
[http://localhost:8080/doan/normalweb/reflected.php?user=<script>alert(document.cookie)</script](http://localhost:8080/doan/normalweb/reflected.php?user=%3cscript%3ealert(document.cookie)%3c/script)>

Người dùng dễ dàng bị hacker chiếm session

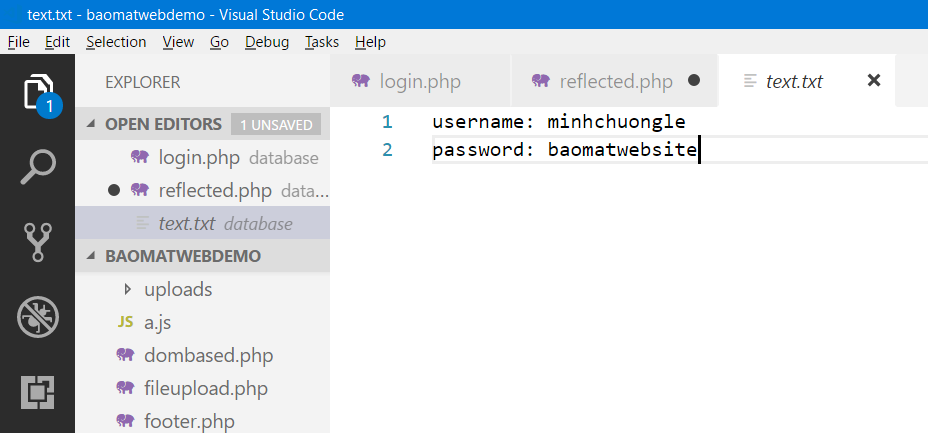
* Đưa người dùng đến một trang web giả mạo để lấy cắp thông tin. Hacker sẽ gửi đến người dùng một URL có dạng:

http://localhost:8080/doan/normalweb/reflected.php?user= <script> window.location.href = "login2.php"</script>

Trong đó: login2.php là một trang web giả mạo trang login để người dùng nhập thông tin vào và gửi đến hacker.

Source file: C:\wamp64\www\doan\errorweb\login2.php

Người dùng vẫn vô tư nhập thông tin username và password mà không để ý mình đã bị chuyển hướng sang một trang web giả mạo.

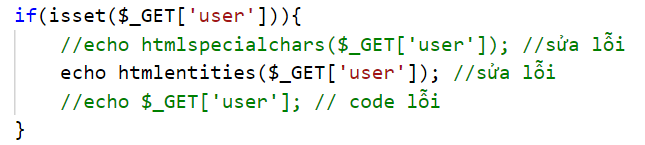


Thông tin nạn nhân được gửi đến hacker.

## Cách phòng chống

* Dùng hàm htmlspecialchars( $string) hoặc htmlentities($string) để xử lý các chuỗi đưa vào. Hàm này chuyển các thẻ html trong chuỗi $string sang dạng thực thể của chúng (html sẽ không còn tác dụng nên có thể echo ra bên ngoài).

Source file : C:\wamp64\www\doan\normalweb\reflected.php



Chuỗi đưa vào đã được xử lý nên không thể thực thi đoạn mã của hacker.

# Cross-site Scripting – Stored XSS

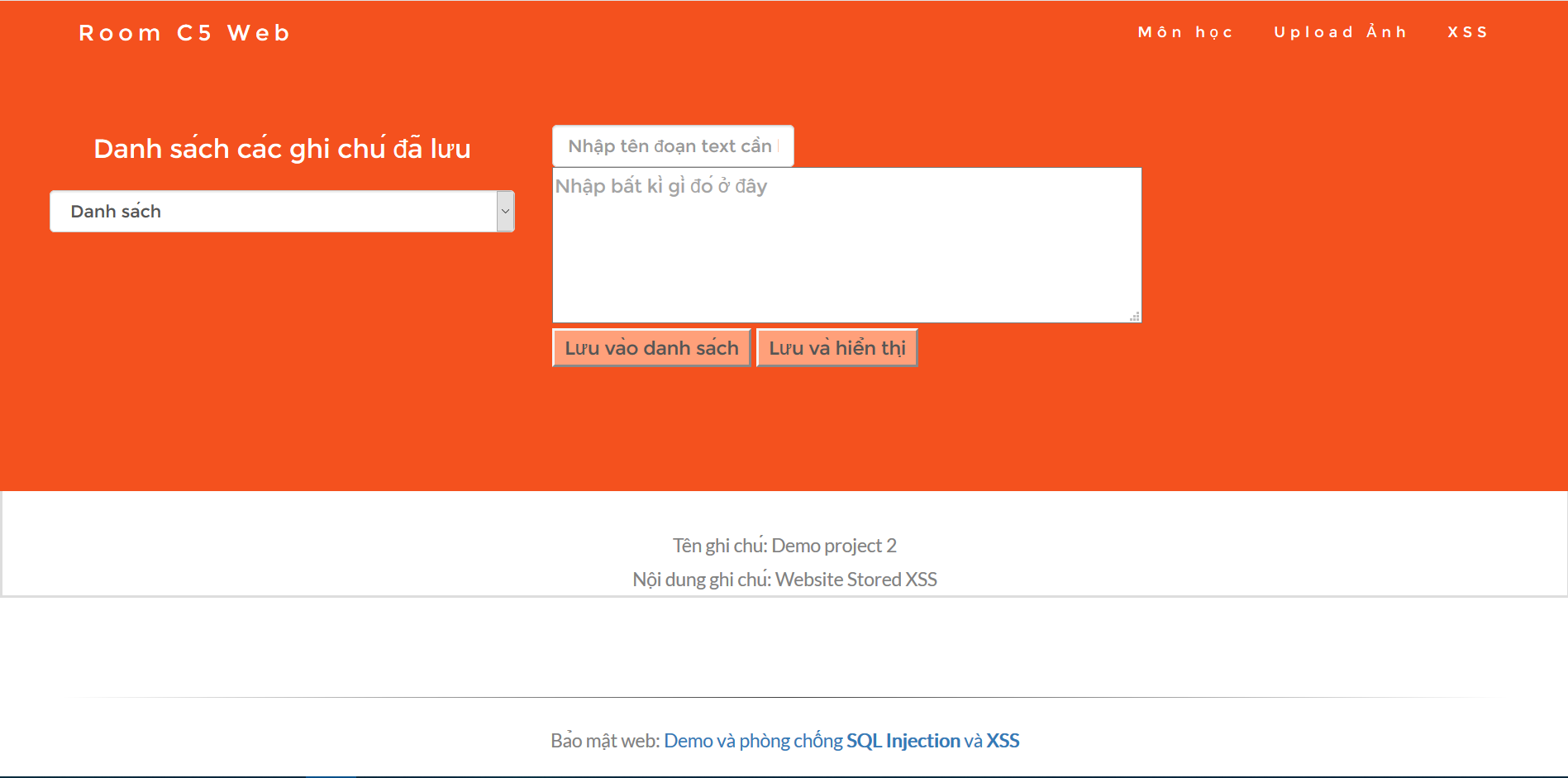
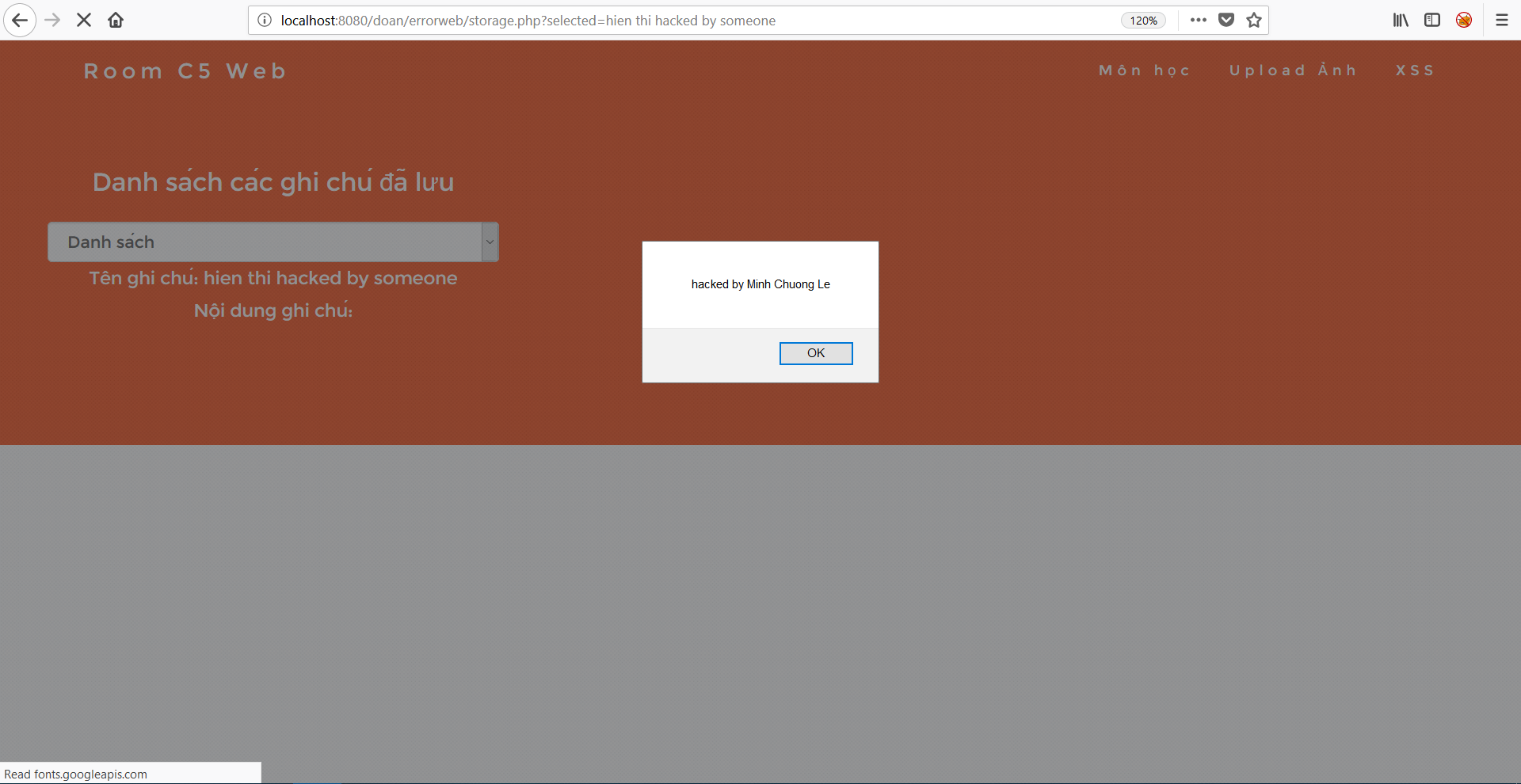
## Định nghĩa

* **Cross-site scripting** là một lỗ hổng phổ biến trong ứng dụng web. Để khai thác một lỗ hổng XSS, hacker sẽ chèn mã độc thông qua các đoạn script để thực thi chúng ở phía client. Thông thường, các cuộc tấn công XSS được sử dụng để vượt qua các kiểm soát truy cập và mạo danh người dùng.
* Khác với Reflected tấn công trực tiếp vào một số nạn nhân mà hacker nhắm đến, Stored XSS hướng đến nhiều nạn  nhân hơn. Lỗi này xảy ra khi ứng dụng web không kiểm tra kỹ các dữ liệu đầu vào trước khi lưu vào cơ sở dữ liệu (ở đây tôi dùng khái niệm này để chỉ database, file hay những khu vực khác nhằm lưu trữ dữ liệu của ứng dụng web). Ví dụ như các form góp ý, các comment … trên các trang web.

## Demo tấn công

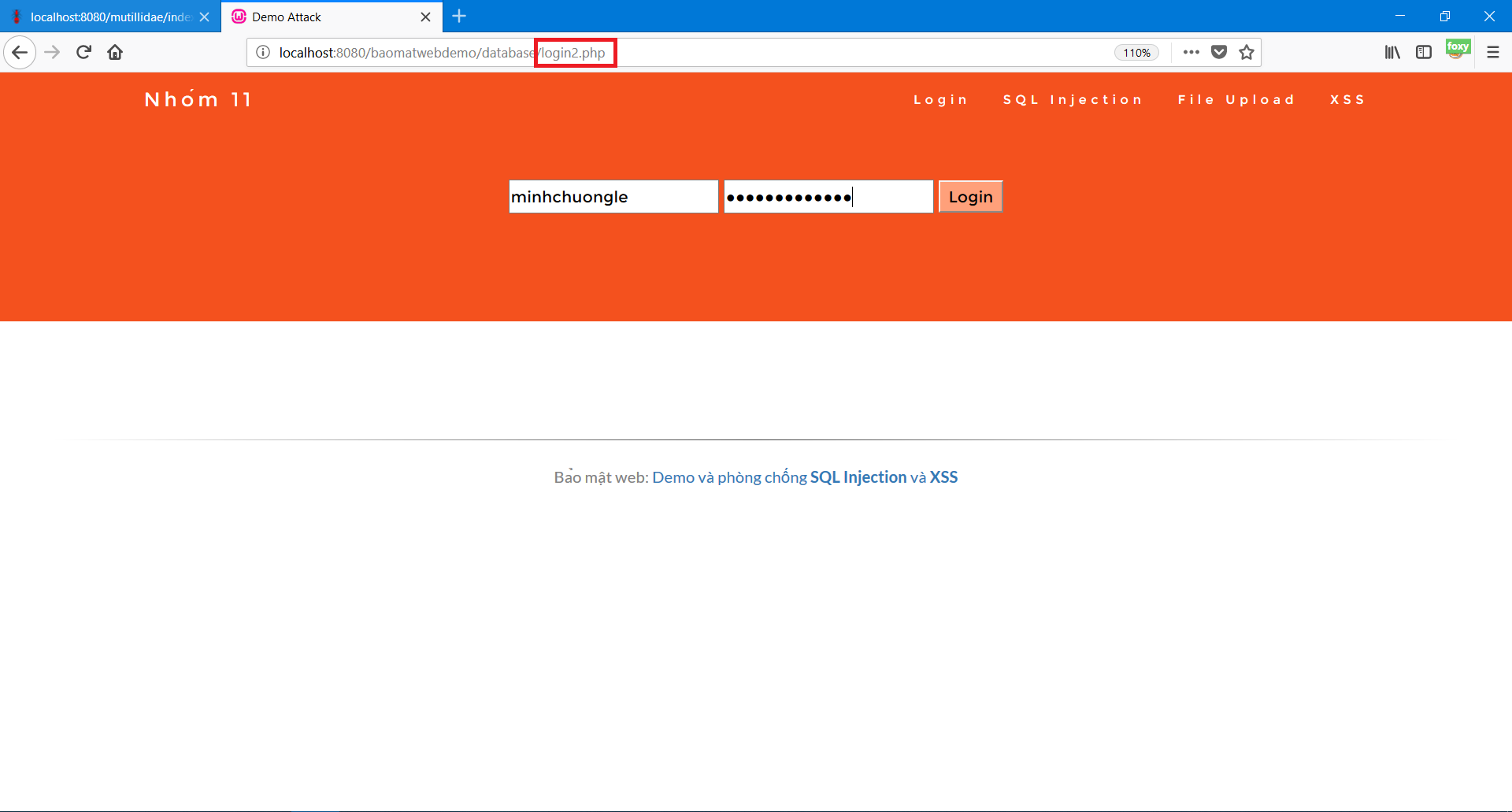
Ta có một trang web Lưu ghi chú bị dính lỗi Stored XSS:

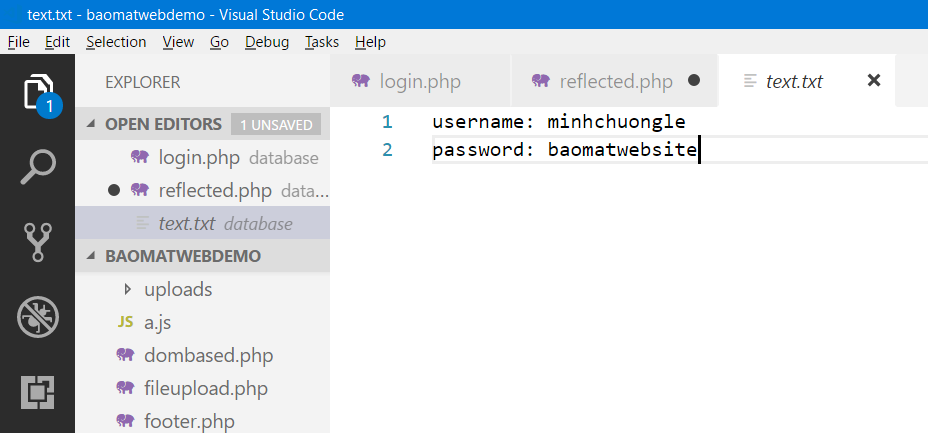
Source file: C:\wamp64\www\doan\errorweb\storage.php

Hacker lợi dụng cơ sở dữ liệu để lưu các câu query vào đó, khi người dùng ấn vào một chức năng nào đó sẽ bị dính lỗi:

* Đưa người dùng đến một trang web giả mạo để lấy cắp thông tin. Trong đó: login2.php là một trang web giả mạo trang login để người dùng nhập thông tin vào và gửi đến hacker.

Source file: C:\wamp64\www\doan\errorweb\login2.php

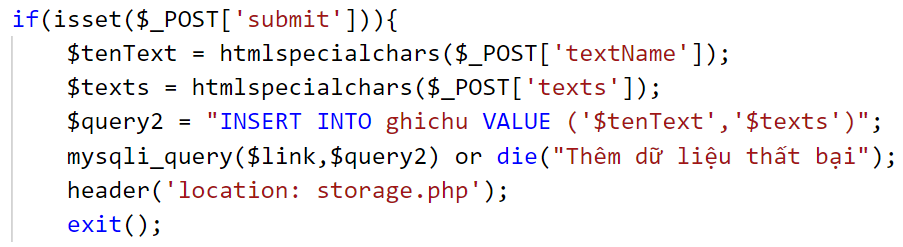


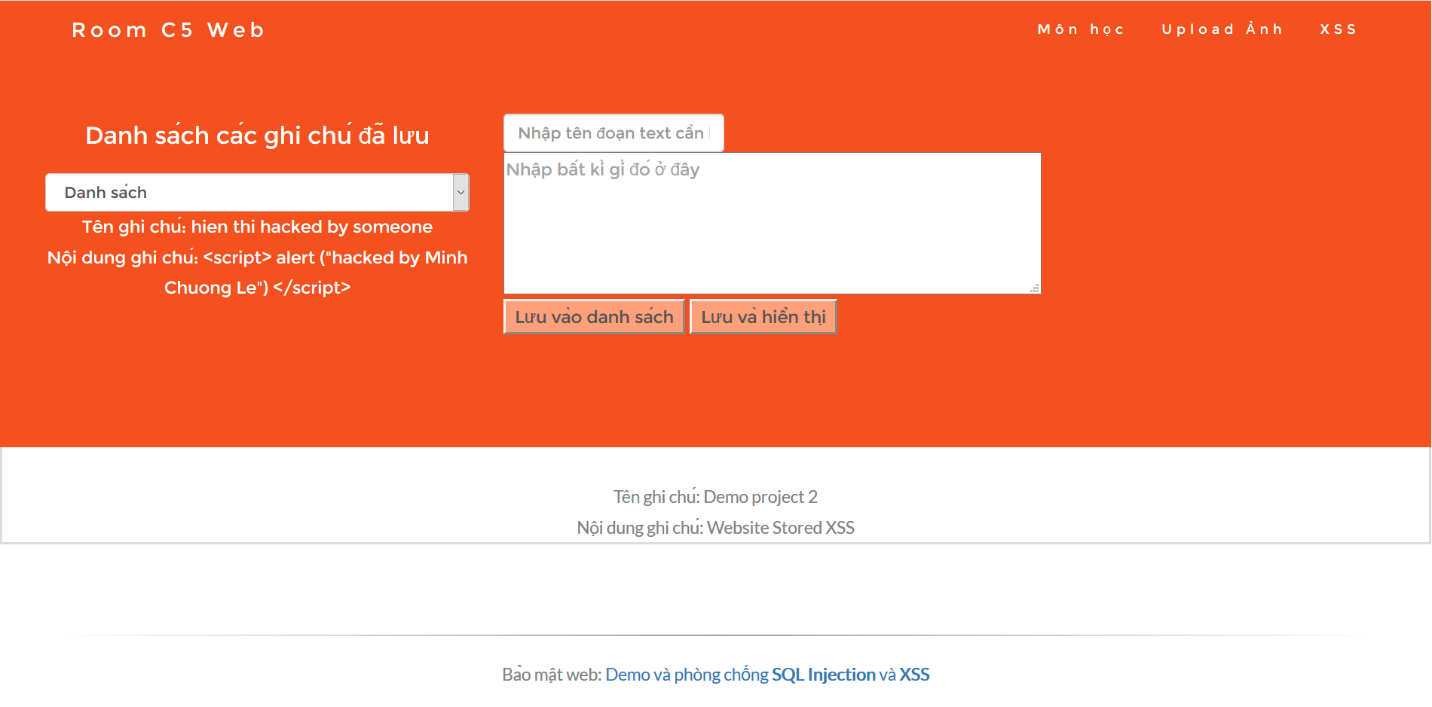
Người dùng vẫn vô tư nhập thông tin username và password mà không để ý mình đã bị chuyển hướng sang một trang web giả mạo

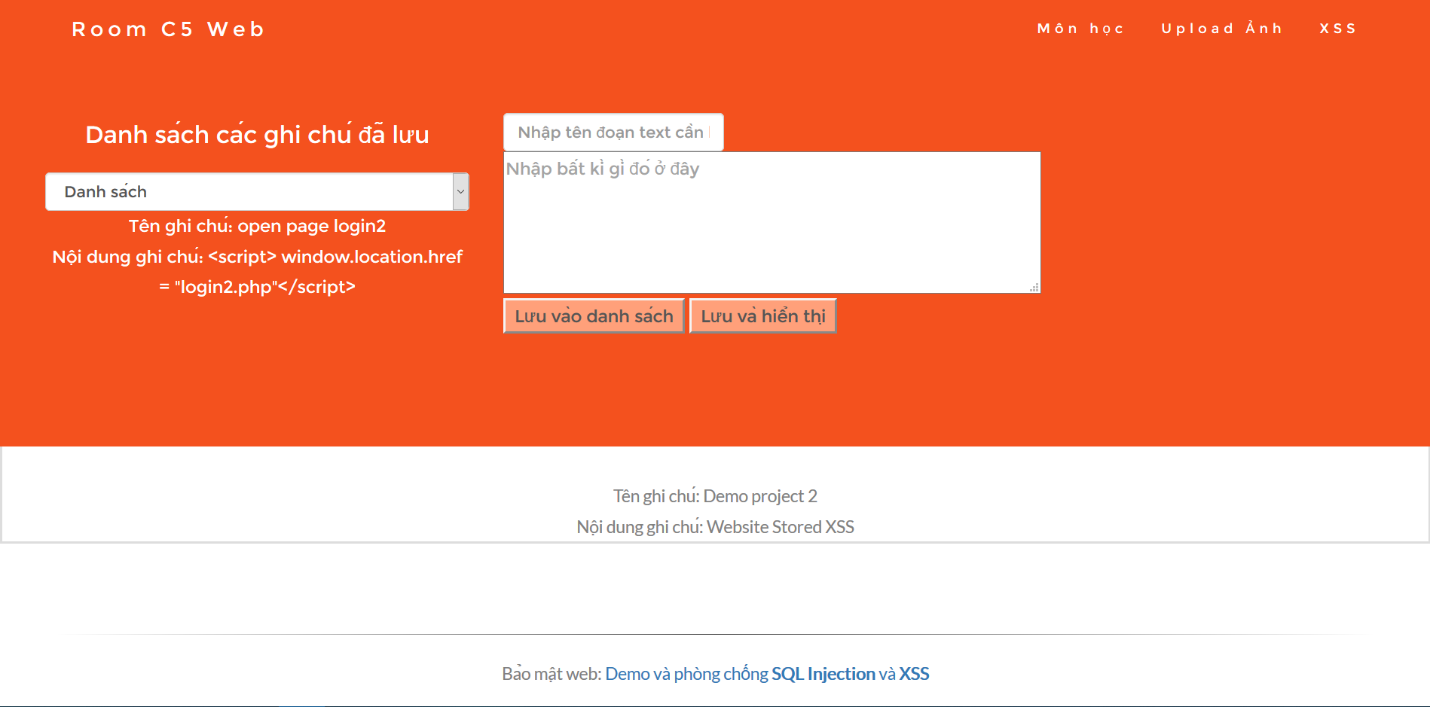
Thông tin nạn nhân được gửi đến hacker.

## Cách phòng chống

* Dùng hàm htmlspecialchars( $string) hoặc htmlentities($string) để xử lý các thông tin đưa vào database. Hàm này chuyển các thẻ html trong chuỗi $string sang dạng thực thể của chúng (html sẽ không còn tác dụng nên có thể echo ra bên ngoài).

Source file: C:\wamp64\www\doan\normalweb\storage.php

Thông tin nhập vào được mã hóa trước khi lưu vào database nên vẫn hiển thị lên bình thường, không thực thi đoạn script độc.



# PHẦN KẾT LUẬN

Ứng dụng web phát triển với tốc độ chóng mặt vì vậy kĩ thuật tấn công và bảo mật ứng dụng web cũng phát triển với trình độ cao. Tấn công và bảo mật ứng dụng web là một đề tài rất rộng đòi hỏi sự am hiểu và phải có kĩ thuật chuyên môn cao. Do thời gian làm bài tập lớn và kĩ thuật của chúng em còn có hạn nên những phần chúng em trình bày ở bài tập lớn này mới là những kiến thức cơ bản và còn nhiều hạn chế. Những được sự giúp đỡ tận tình của thầy Nguyễn Bá Tiến. Chúng em cũng hiểu rõ hơn về những kĩ thuật tấn công ứng dụng cơ bản là nền tảng cho những phương pháp tấn công hiện đại đồng thời chúng em cũng biết được một số phương pháp phòng tránh tấn công và bảo mật ứng dụng web.

Những kiến thức này sẽ giúp ích rất nhiều cho chúng em sau này. Những kiến thức này sẽ giúp chúng em có nền tảng vững chắc hơn về kĩ thuật chuyên môn liên quan đến kĩ thuật bảo mật ứng dụng web. Sau khi thực hiện xong bài tập lớn này chúng em sẽ sử dụng những kiến thức đã tìm hiểu và cố gắng học thêm, tìm hiểu thêm những kiến thức chuyên môn liên quan nhăm xây dụng một công cụ hay một phương pháp bảo mật ứng dụng web hiệu quả trước các cuộc tấn công.