Bài 1: SQL injection vulnerability in WHERE clause allowing retrieval of hidden data

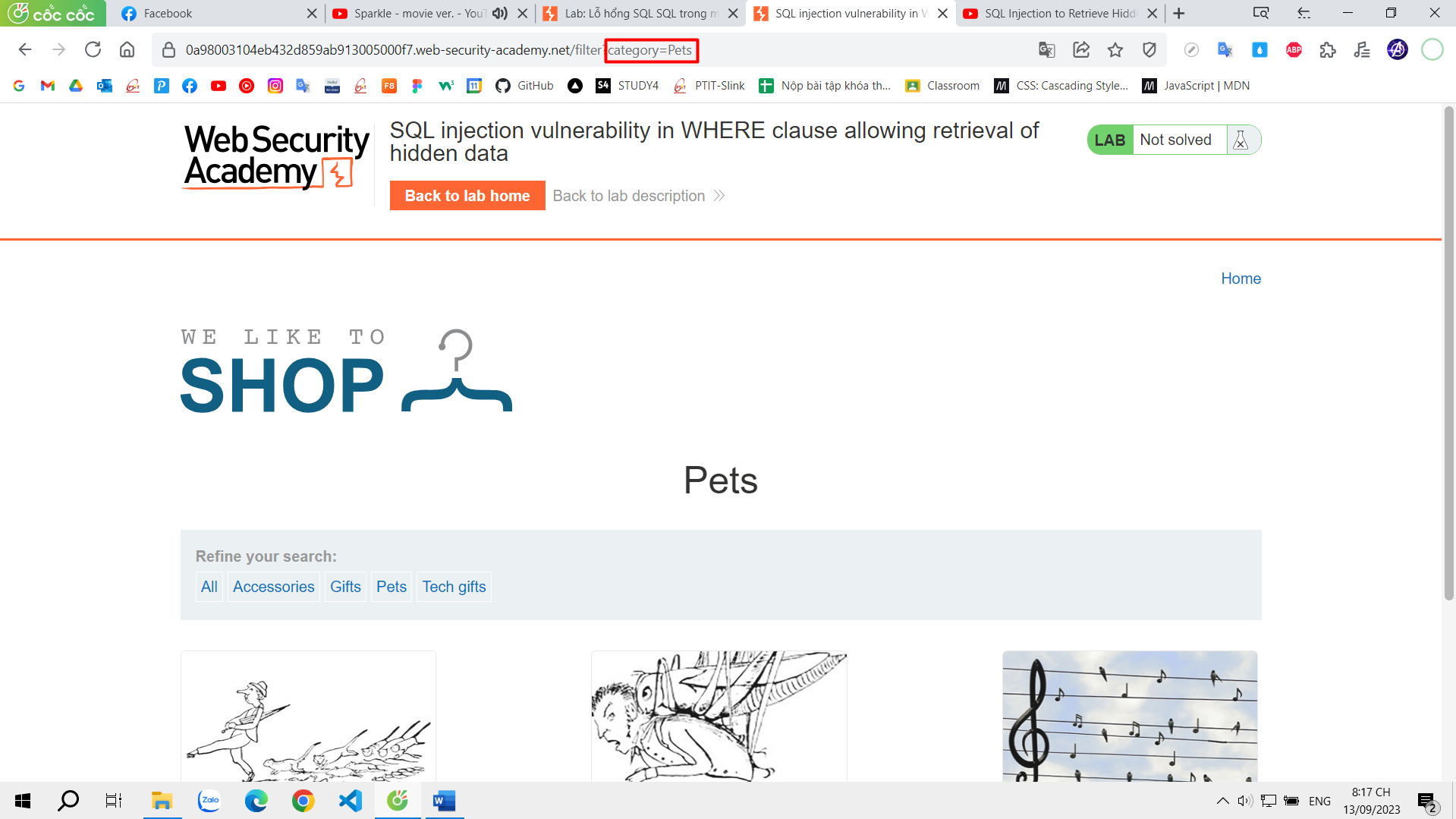
Trang web này chứa lỗ hổng [chèn SQL](https://portswigger.net/web-security/sql-injection) trong bộ lọc danh mục sản phẩm. Khi người dùng chọn một danh mục, ứng dụng sẽ thực hiện truy vấn SQL như sau:

SELECT \* FROM products WHERE category = 'Gifts' AND released = 1

- Câu truy vấn này là tất cả sản phẩm trong danh mục quà tặng và đã được phát hành

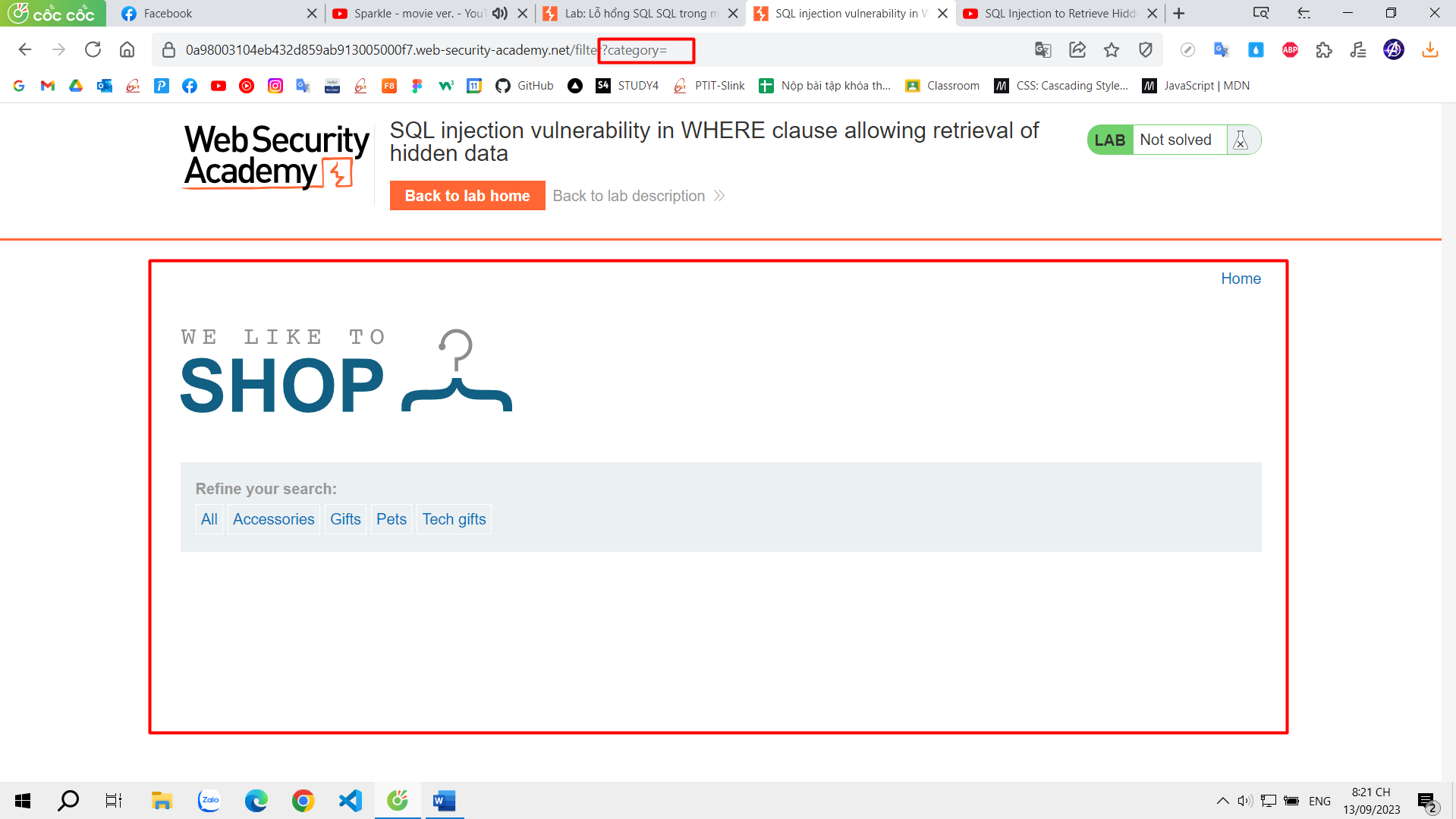
Ta vào danh mục Pets thì sẽ hiện ra câu truy vấn:

SELECT \* FROM products WHERE category = 'Pets' AND released = 1



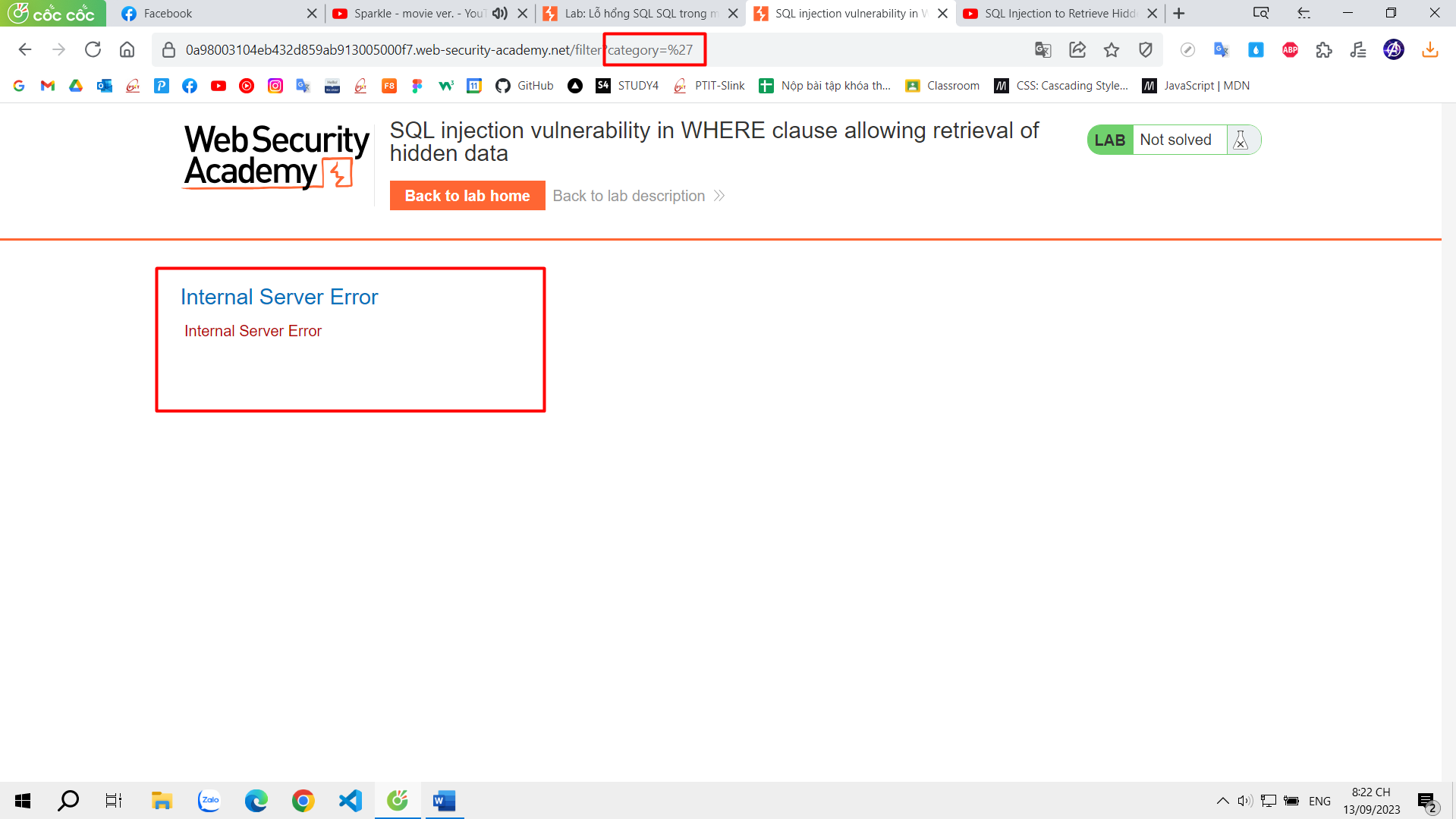
Khi ta sửa lại câu truy vấn thành:

SELECT \* FROM products WHERE category = AND released = 1 thì sẽ không hiện sản phẩm nào



Sửa thành:

SELECT \* FROM products WHERE category = ‘ AND released = 1 thì sẽ bị Lỗi máy chủ nội bộ

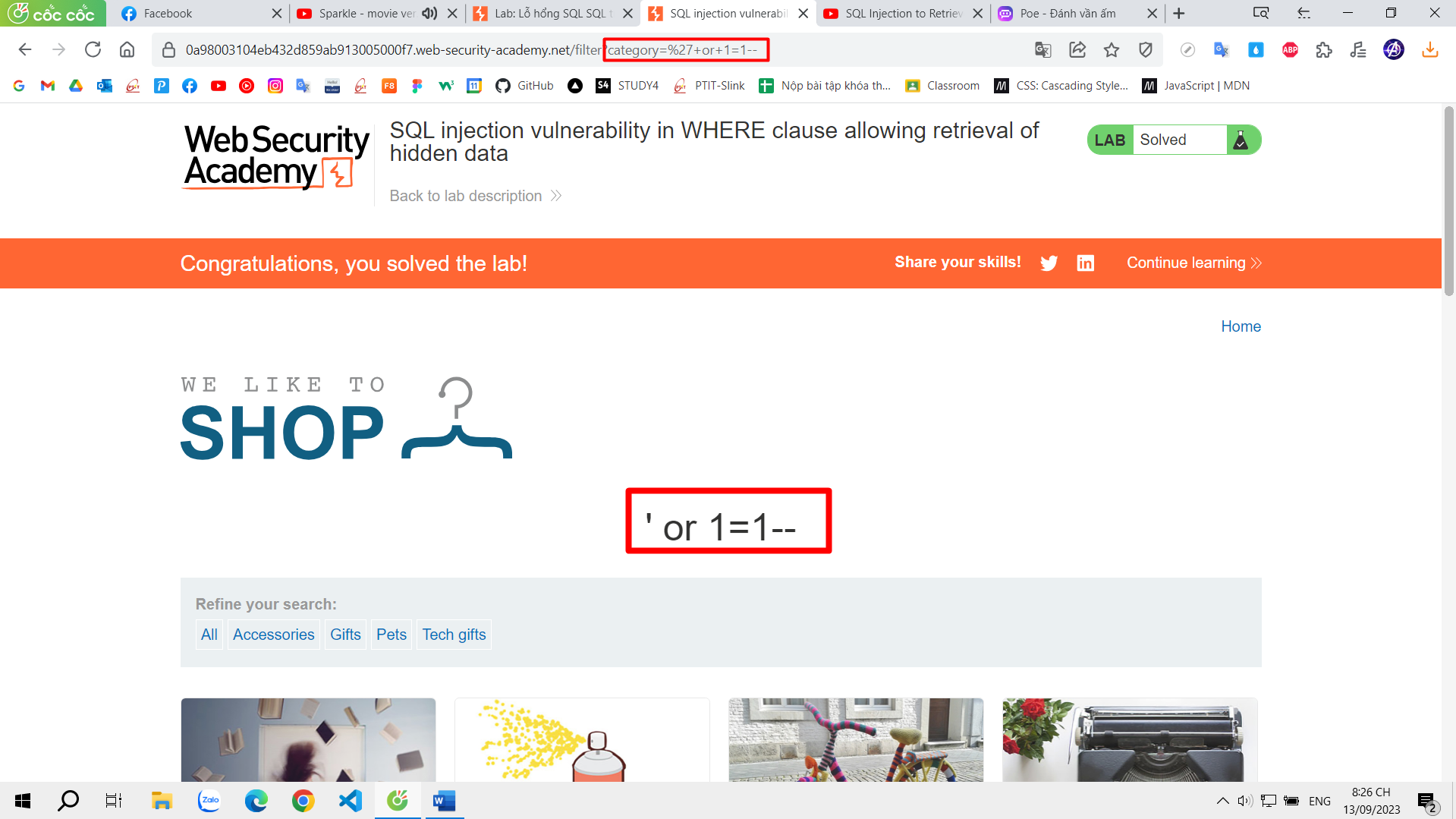


Khi sửa lại câu truy vấn thành

SELECT \* FROM products WHERE category = OR 1=1-- AND released = 1

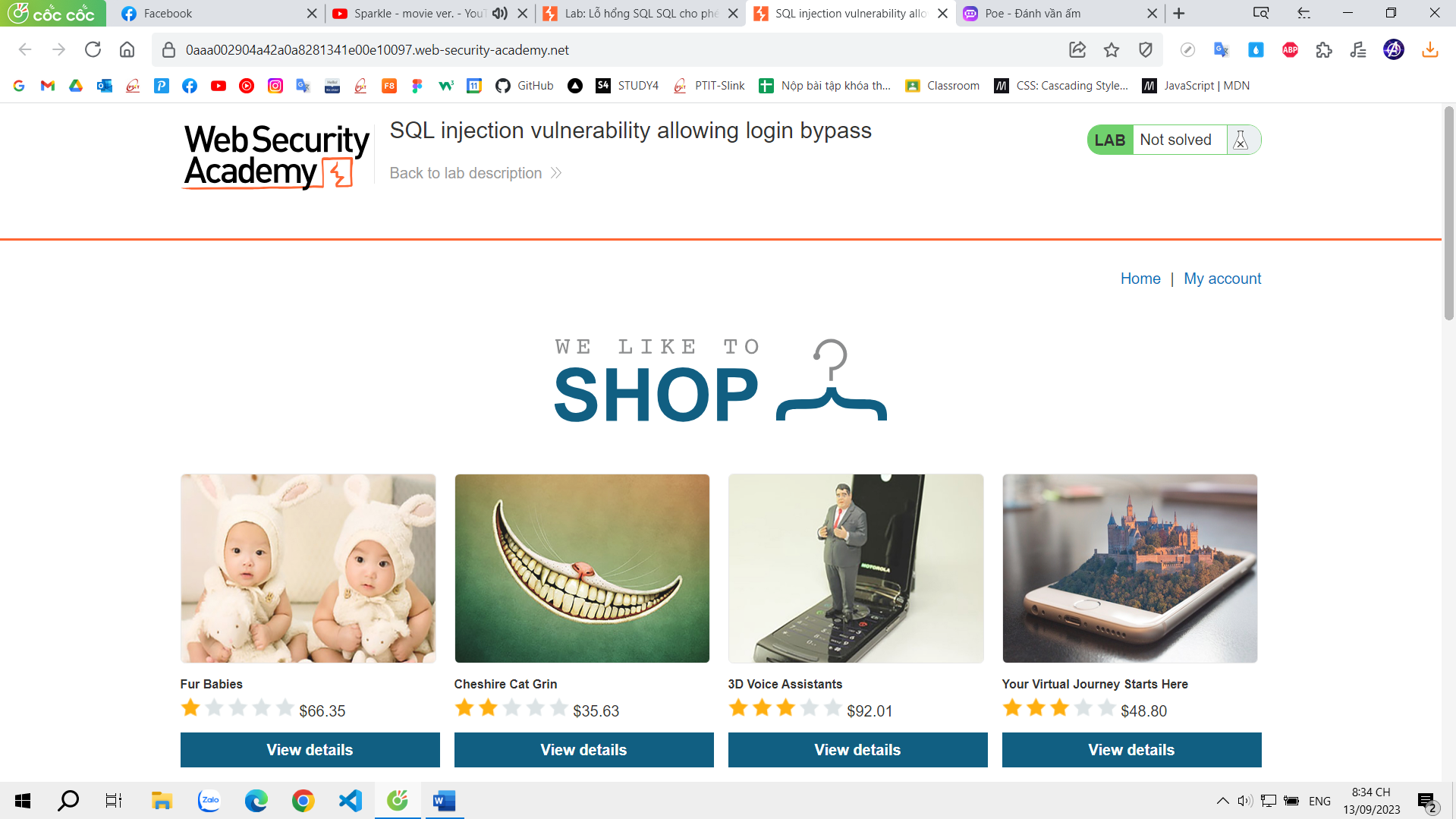
Phần "OR 1=1" được chèn vào để luôn trả về giá trị true (đúng), và "--" được sử dụng để chú thích (comment) phần còn lại của câu truy vấn, từ đó vô hiệu hóa điều kiện "AND released = 1".

🡪 sẽ hiện ra thêm những sản phẩm bị ẩn

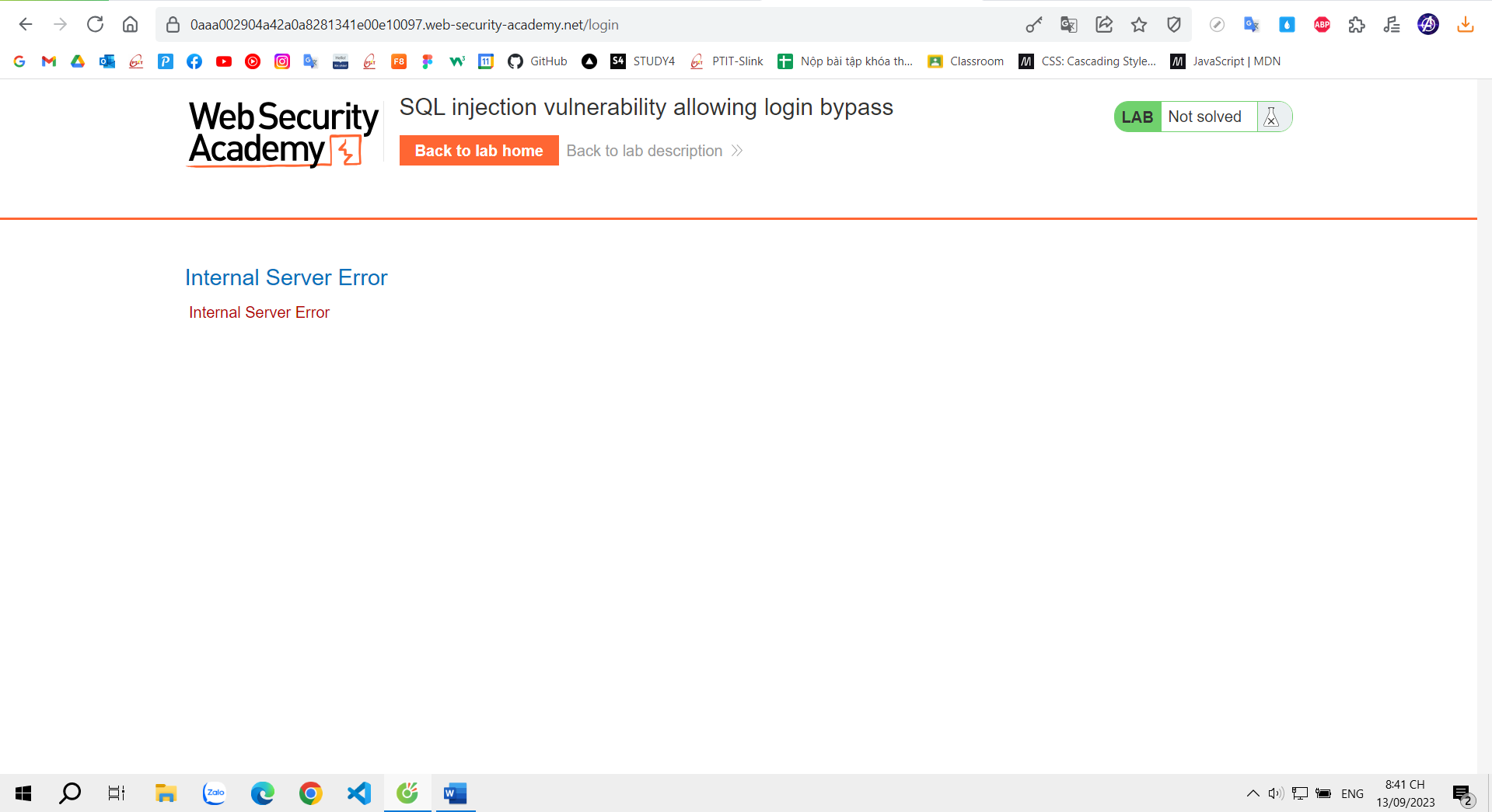


Bài 2: Lab: SQL injection vulnerability allowing login bypass

Web này chứa lỗ hổng [SQL SQL](https://portswigger.net/web-security/sql-injection) trong chức năng đăng nhập.

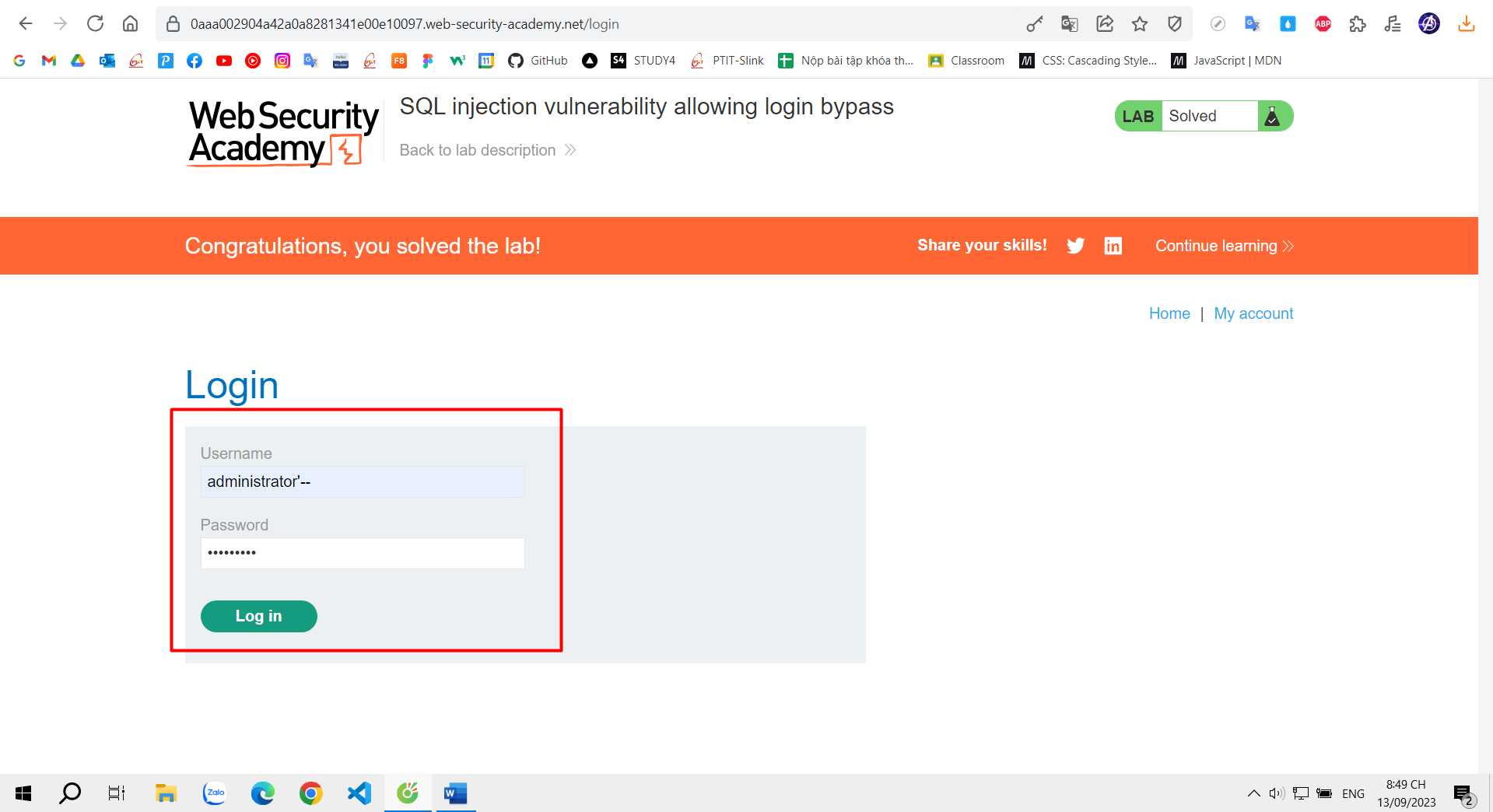


Đầu tiên là điền tên đăng nhập bằng dấu ‘

SELECT firtname From user WHERE username = ‘’ ’ AND password = ‘admin’ thì sẽ bị lỗi máy chủ nội bộ 

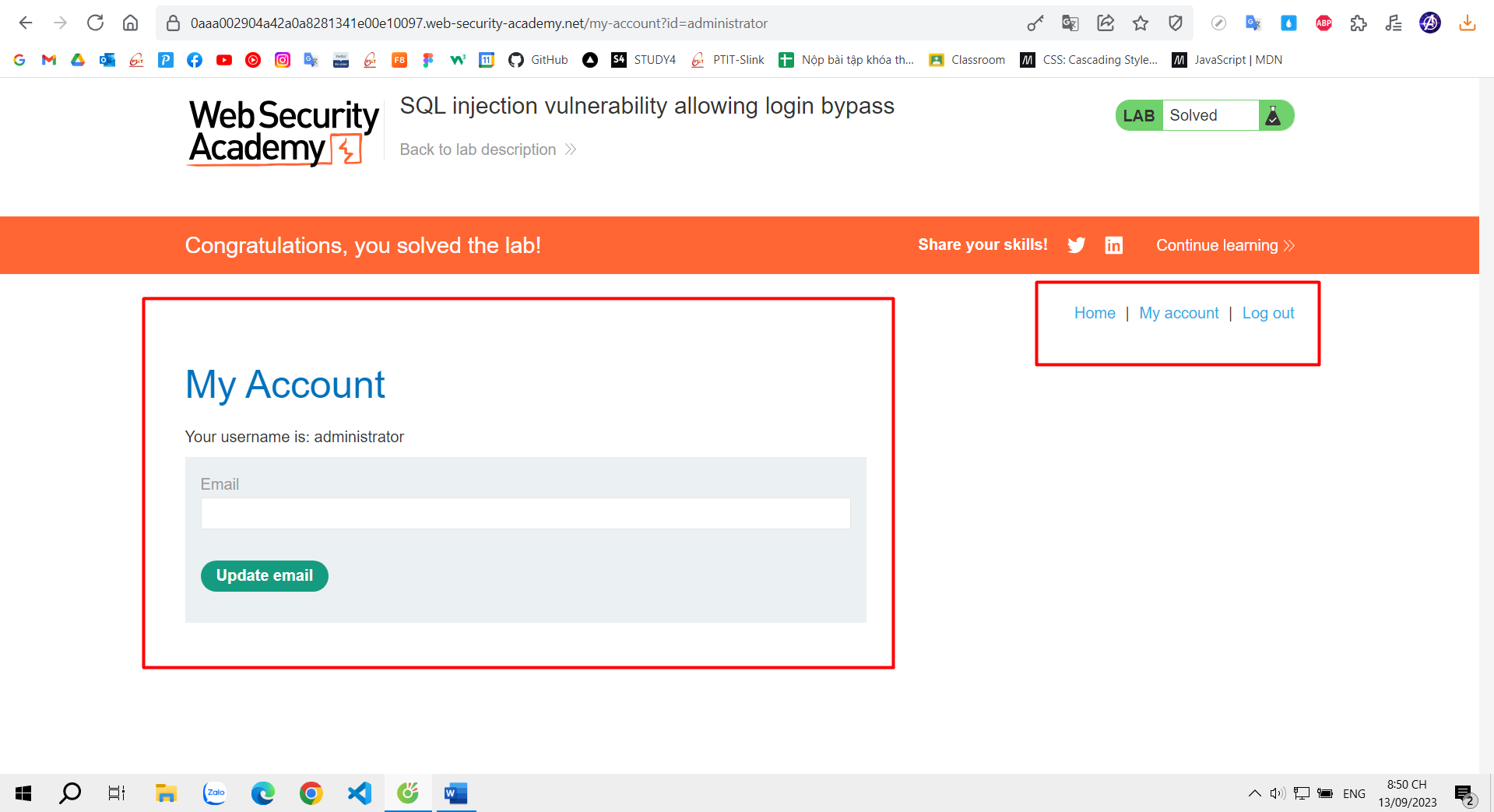
Sửa lại câu truy vấn thành:

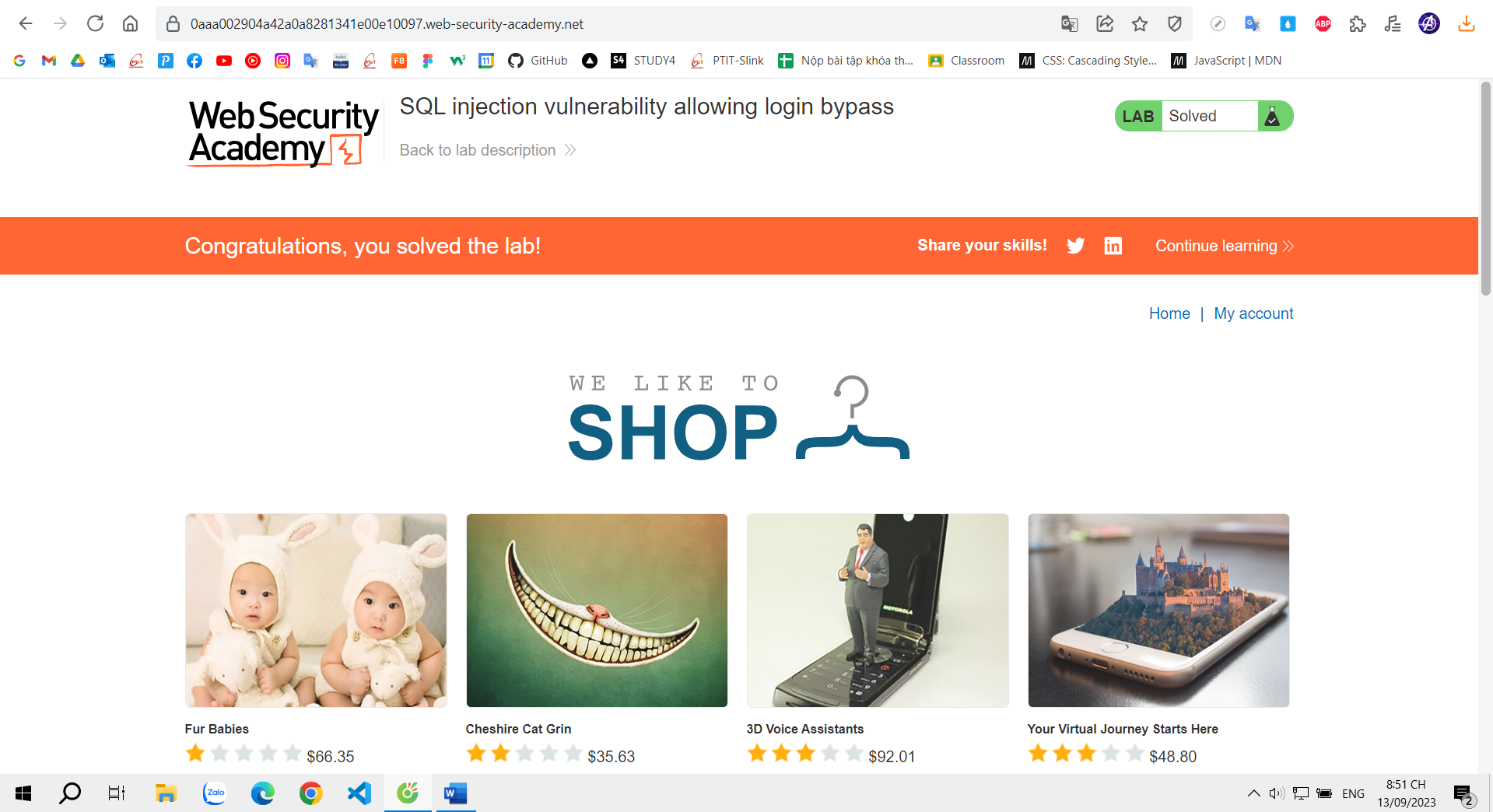
SELECT firtname From user WHERE username = ‘administator’-- ’ AND password = ‘admin’



🡪 Câu truy vấn này cố gắng lấy thông tin về trường "firtname" (tên) trong bảng "user" từ một người dùng có "username" là 'administrator’--'. Mã "--" được chèn vào cuối câu truy vấn để chú thích phần còn lại của câu truy vấn. Điều này có tác dụng vô hiệu hóa điều kiện "WHERE username = 'administrator'.

Đăng nhập thành công

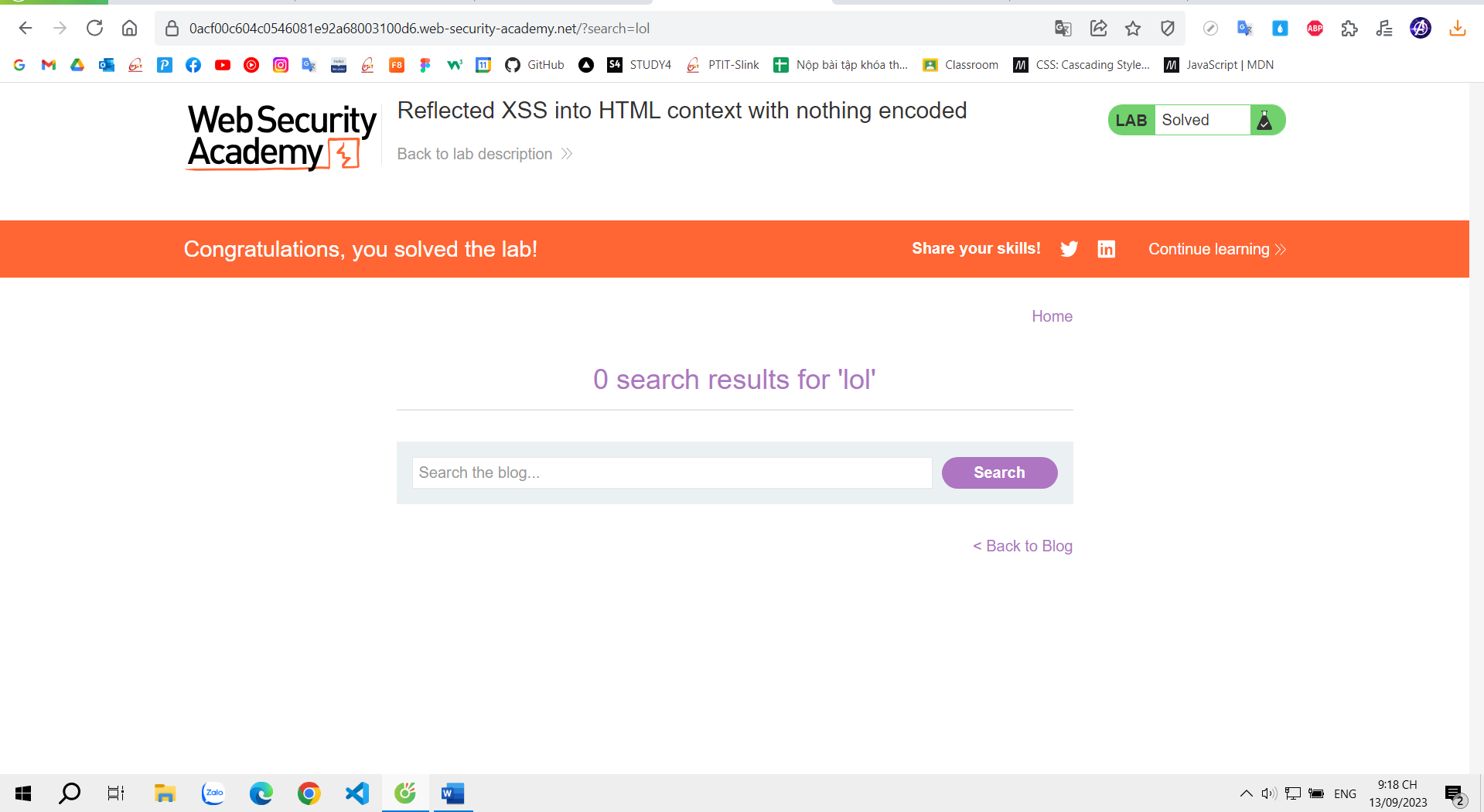


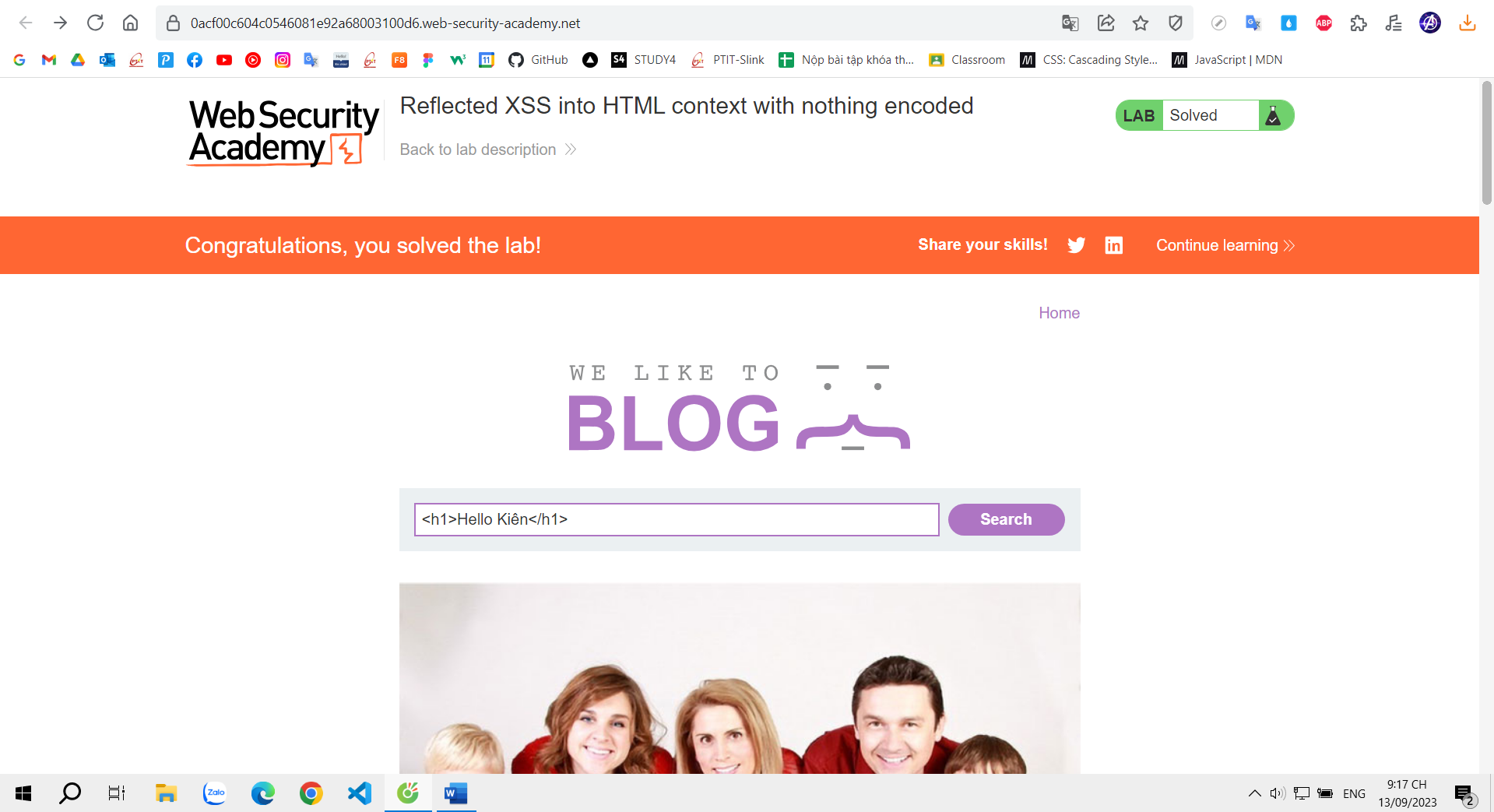


Bài 3:  [Reflected XSS](https://portswigger.net/web-security/cross-site-scripting/reflected) into HTML context with nothing encoded

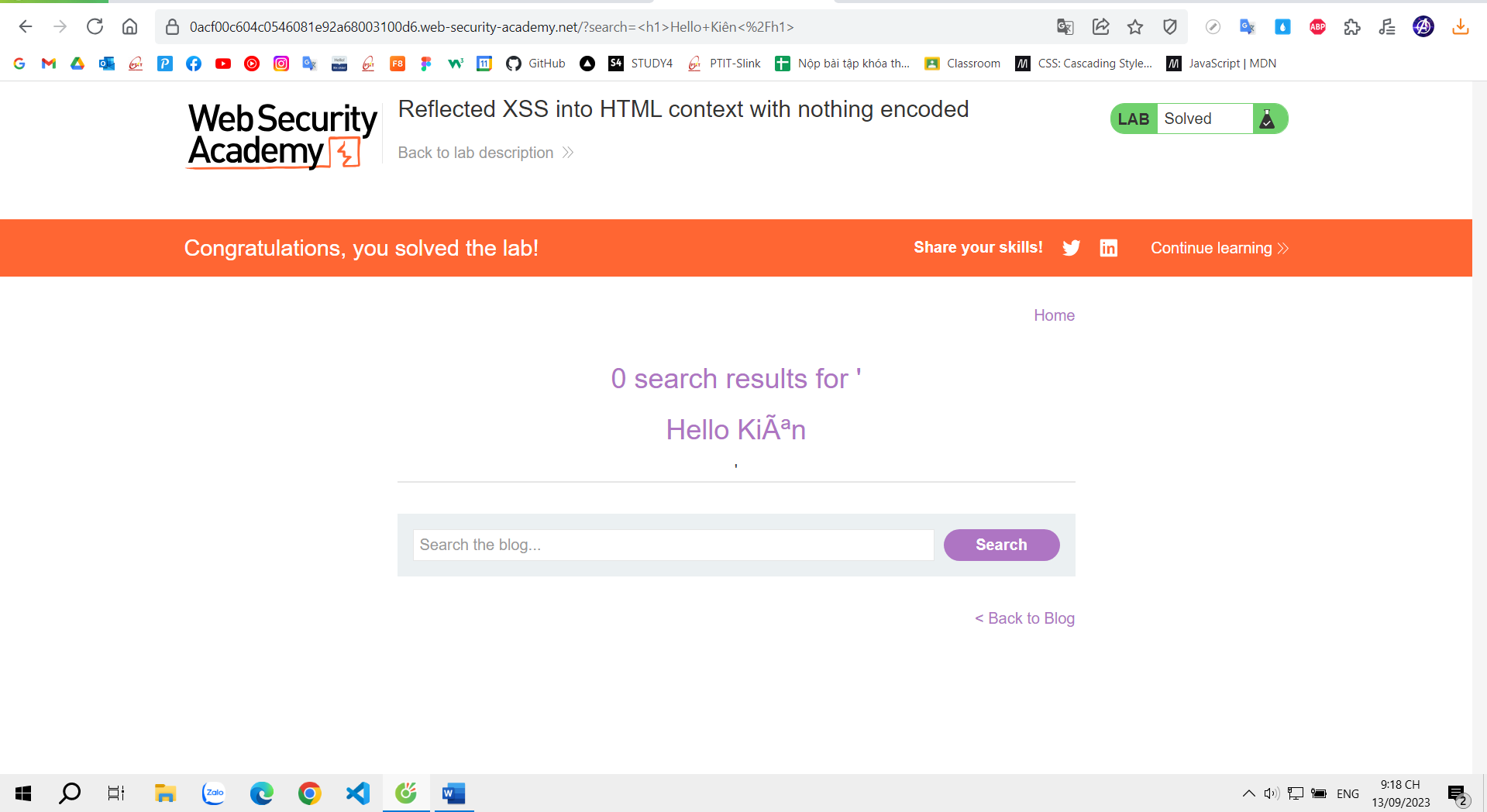
Web này chứa một lỗ hổng [kịch bản chéo trang đơn giản được phản ánh](https://portswigger.net/web-security/cross-site-scripting/reflected) trong chức năng tìm kiếm.

- Khi tìm bất kỳ từ khóa trên thanh tìm kiếm

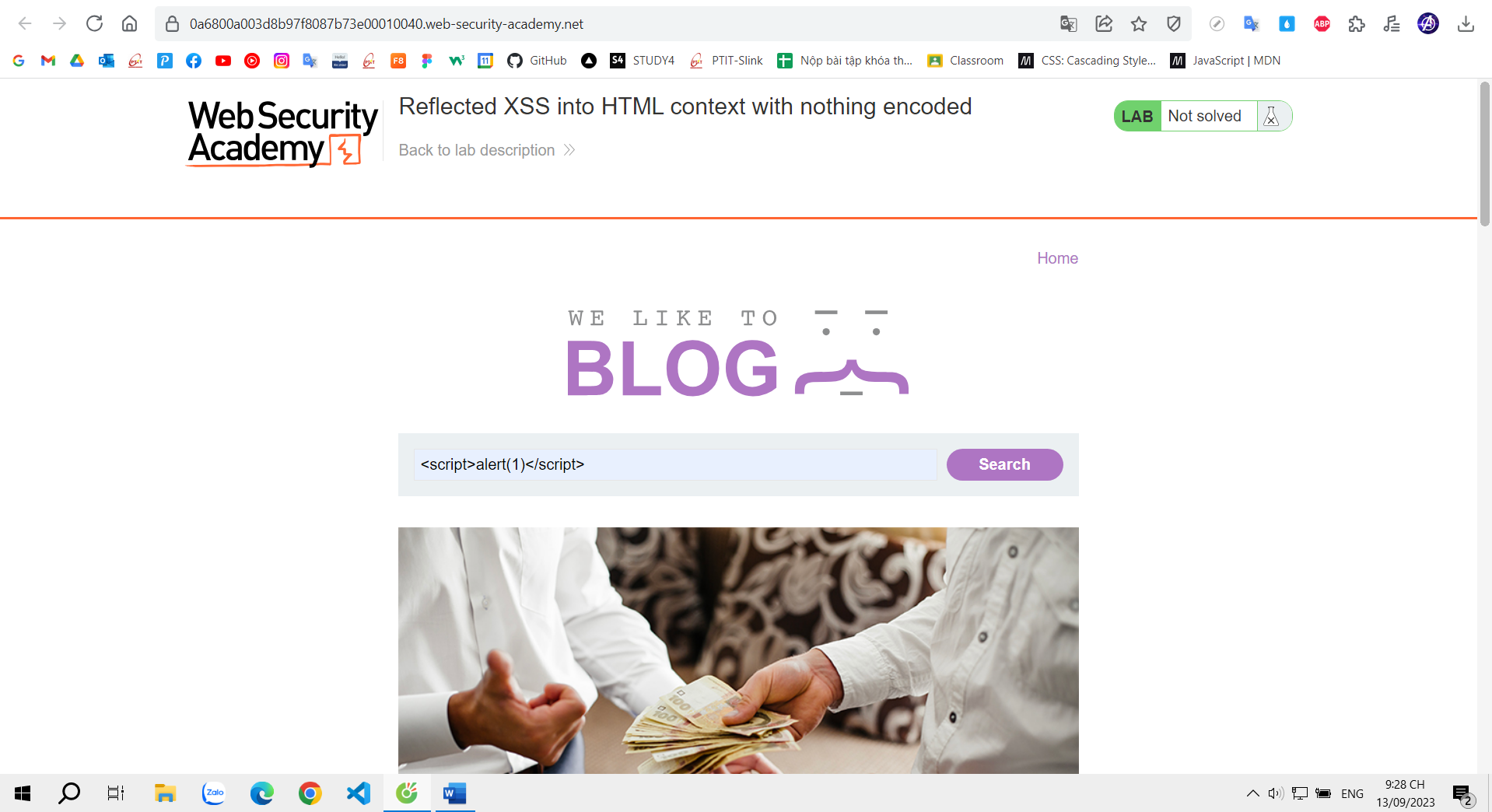


Gõ <h1>Hello Kiên</h1> trên thanh tìm kiếm

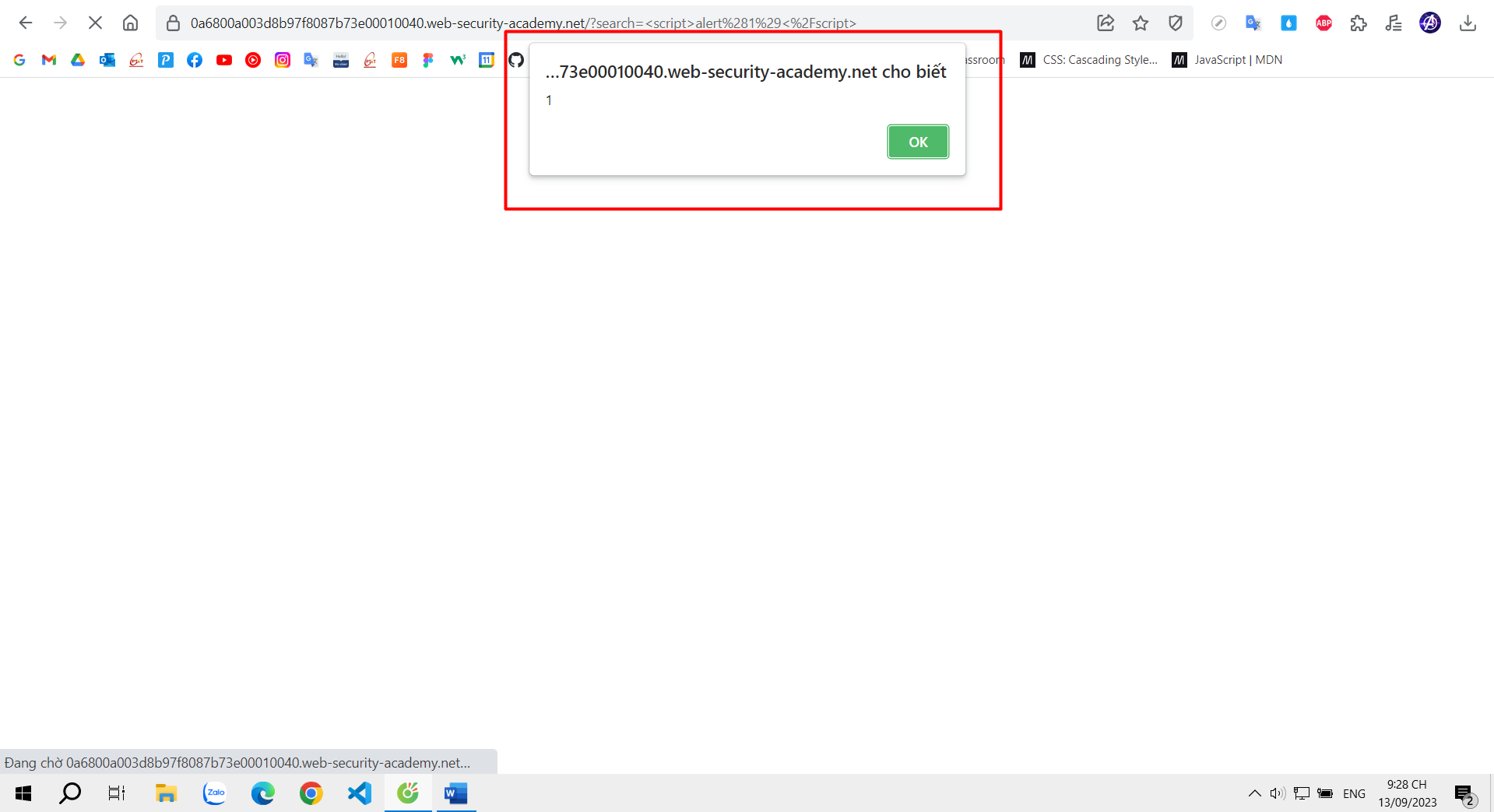
🡪 Thì dòng hello Kiên sẽ xuống dòng vì thẻ h1 là thẻ block



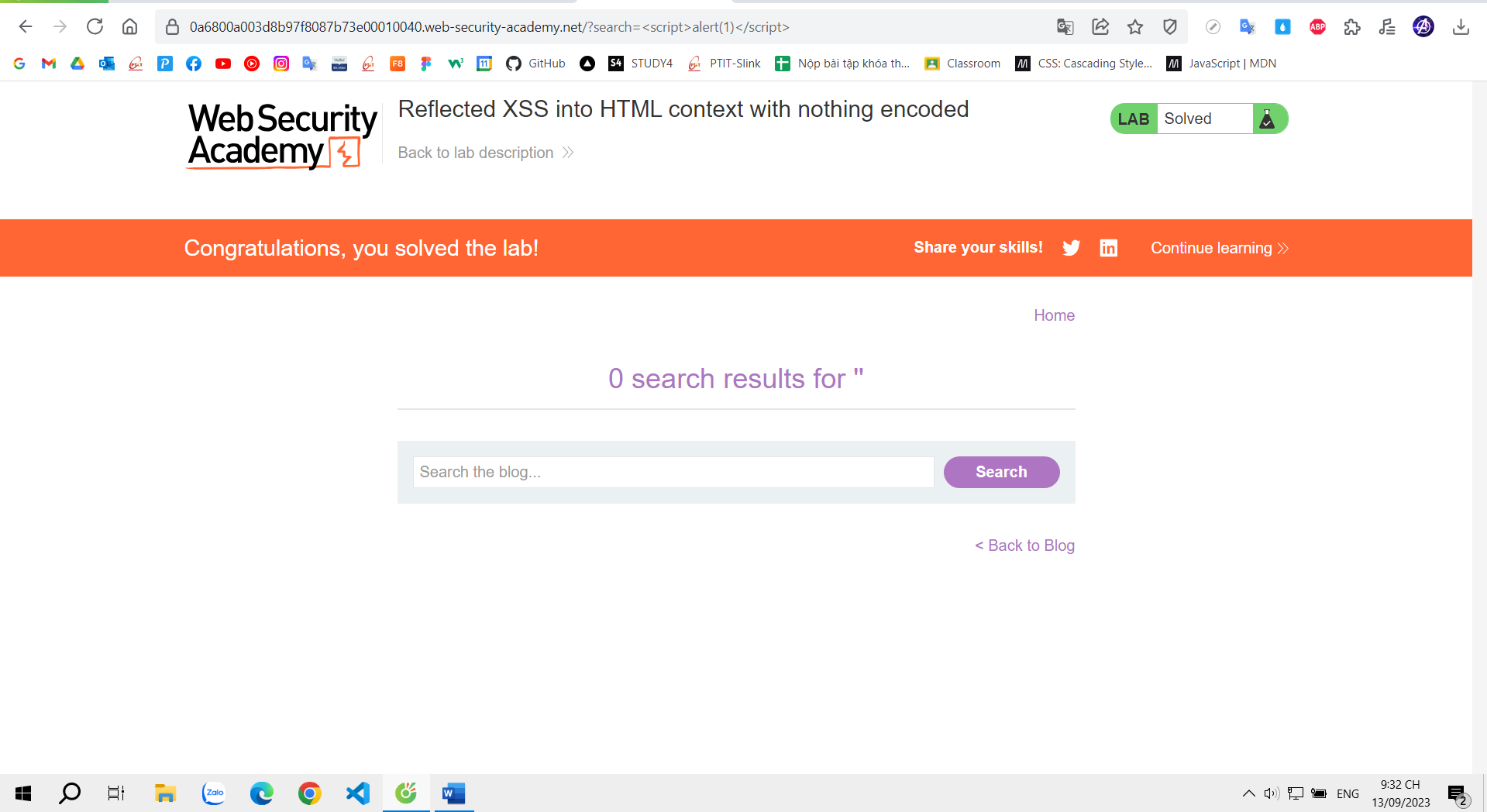
Gõ <script>alert(1)</script> vào thanh tìm kiếm



Khi được chèn vào trong thanh tìm kiếm của một trang web và được thực thi, nó sẽ tạo ra một hộp thoại cảnh báo trên trình duyệt với nội dung là số 1.



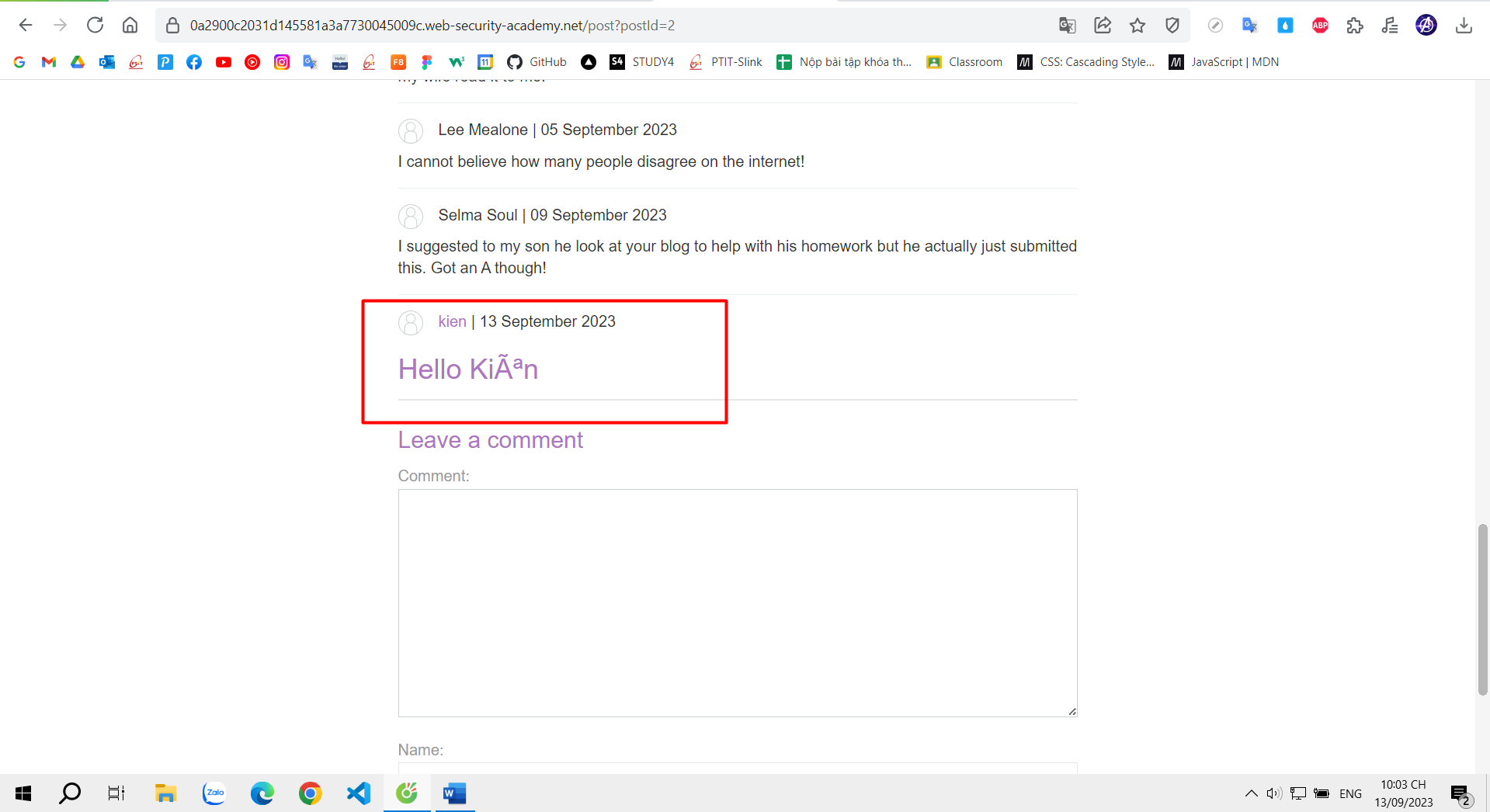
🡪 Khi người dùng truy cập vào trang đó, mã độc sẽ được thực thi trên trình duyệt của họ để đánh cắp thông tin người dùng, thay đổi nội dung trang web, hoặc thực hiện các hành động không mong muốn khác trên trình duyệt của người dùng.



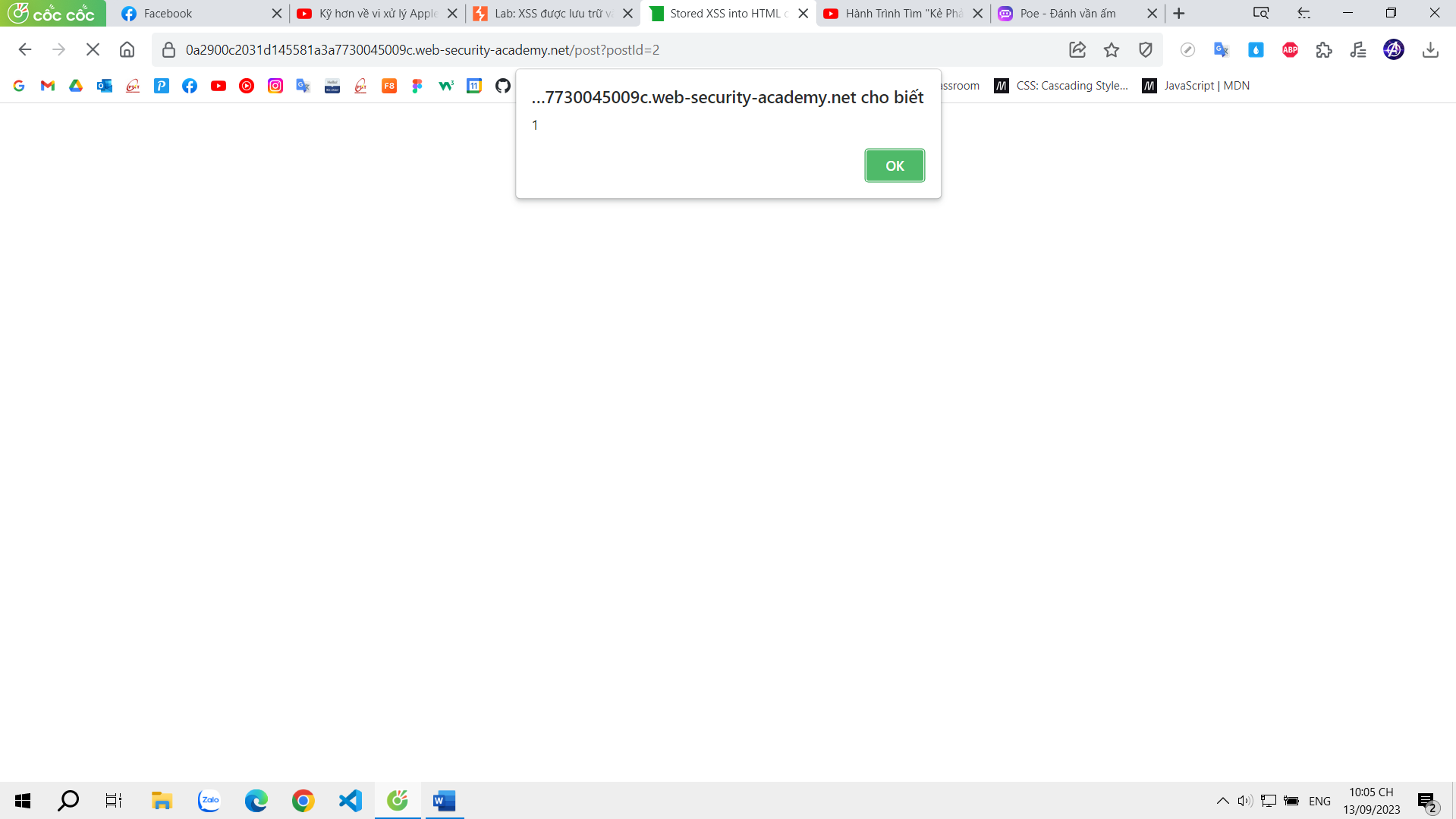
Bài 4: [Stored XSS](https://portswigger.net/web-security/cross-site-scripting/stored) into HTML context with nothing encoded

Web này chứa lỗ hổng [kịch bản chéo trang được lưu trữ](https://portswigger.net/web-security/cross-site-scripting/stored) trong chức năng comment.

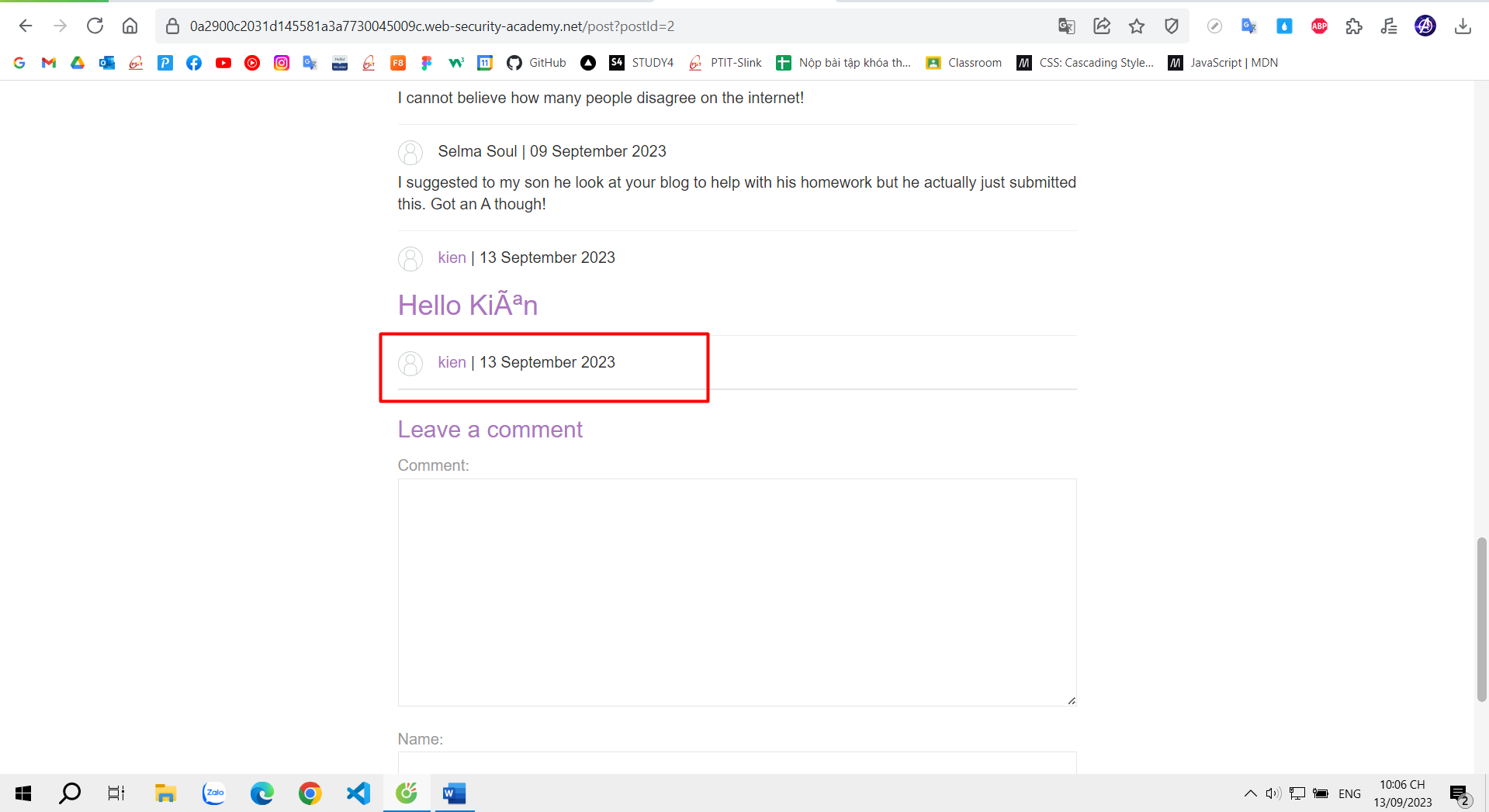
Khi gõ <h1>Hello Kiên</h1> vào comment thì sẽ hiện ra chữ khác biệt với các dòng trên vì thẻ h1 chèn vào



Khi gõ <script> alert (1) </script> vào comment, post bài và quay lại thì sẽ hiện lên thông bso



🡪 Kẻ tấn công chèn mã độc vào phần bình luận của một bài đăng trên diễn đàn, và khi người dùng xem bài đăng đó, mã độc sẽ thực thi trên trình duyệt của họ.



Bài 5: [DOM XSS](https://portswigger.net/web-security/cross-site-scripting/dom-based) in **innerHTML** sink using source **location.search**

Web này chứa lỗ hổng kịch bản chéo trang dựa trên DOM trong chức năng blog tìm kiếm. Nó sử dụng một innerHTML phép gán để thay đổi nội dung HTML của một divphần tử, sử dụng dữ liệu từ các tệp location.search.

DOM XSS (Cross-Site Scripting) là một lỗ hổng bảo mật thường gặp trong ứng dụng web, trong đó tấn công viên có thể chèn mã độc hại vào trang web và thực thi nó trên trình duyệt của người dùng khác. Lỗ hổng DOM XSS xảy ra khi ứng dụng web không kiểm tra hoặc xử lý đúng cách dữ liệu người dùng nhập vào và sau đó sử dụng dữ liệu này để tạo HTML hoặc JavaScript.

innerHTML là một thuộc tính trong DOM (Document Object Model) cho phép truy cập và chỉnh sửa nội dung HTML bên trong một phần tử. Khi không kiểm tra dữ liệu người dùng trước khi gán vào innerHTML, có thể xảy ra lỗ hổng DOM XSS.



Nhập <img src=1 onerror=alert(1)> vào thanh tìm kiếm khi đó:

-Thẻ <img> là chỉ định đường dẫn tới một hình ảnh để hiển thị. Tuy nhiên, trong trường hợp này, giá trị của src là 1, không phải là một đường dẫn hợp lệ, do đó, việc tải hình ảnh sẽ thất bại.

-Khi việc tải hình ảnh thất bại (onerror), trình duyệt sẽ thực thi mã JavaScript được đặt trong thuộc tính onerror. Trong trường hợp này, mã alert(1) được thực thi, và một hộp thoại cảnh báo sẽ xuất hiện trên trình duyệt với nội dung là số 1.

🡪 Kết quả là tải trọng được thực thi bất cứ khi nào trình duyệt của người dùng cố tải trang chứa bài đăng độc hại của bạn.

