**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**

A logo with blue and purple text

Description automatically generated

**TIỂU LUẬN MÔN HỌC  
AN TOÀN BẢO MẬT THÔNG TIN**

**XÂY DỰNG GIẢI PHÁP BẢO MẬT CHO WEBSITE BÁN HÀNG DHND**

**Giảng Viên Hướng Dẫn: ThS. Lê Từ Minh Trí**

**Nhóm 4 Mã số sinh viên Lớp**

**Nguyễn Kim Hoàng 2124802010093 D21CNTT02**

**Trần Nguyễn Vũ Nam** **2124802010401 D21CNTT02**

**Châu Minh Đương** **2124802010036 D21CNTT02**

**Trịnh Xuân Đức 2124802010687 D21CNTT03**

**BÌNH DƯƠNG - 05/2024**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**VIỆN KỸ THUẬT CÔNG NGHỆ**

A logo with blue and purple text

Description automatically generated

**TIỂU LUẬN MÔN HỌC  
AN TOÀN BẢO MẬT THÔNG TIN**

**XÂY DỰNG GIẢI PHÁP BẢO MẬT CHO WEBSITE BÁN HÀNG DHND**

**Giảng Viên Hướng Dẫn: ThS. Lê Từ Minh Trí**

**Nhóm 4 Mã số sinh viên Lớp**

**Nguyễn Kim Hoàng 2124802010093 D21CNTT02**

**Trần Nguyễn Vũ Nam 2124802010401 D21CNTT02**

**Châu Minh Đương 2124802010036 D21CNTT02**

**Trịnh Xuân Đức 2124802010687 D21CNTT03**

**BÌNH DƯƠNG - 05/2024**

MỤC LỤC

[MỤC LỤC i](#_Toc166783202)

[BẢNG PHÂN CÔNG CỦA NHÓM ix](#_Toc166783203)

[DANH MỤC HÌNH x](#_Toc166783204)

[MỞ ĐẦU 1](#_Toc166783205)

[CHƯƠNG 1. GIỚI THIỆU CHUNG 2](#_Toc166783206)

[1.1. Lý do thực hiện đề tài 2](#_Toc166783207)

[1.1.1. Hiện trạng 2](#_Toc166783208)

[1.1.2. Yêu cầu bảo mật 2](#_Toc166783209)

[1.2. Tổng quan về vấn đề nghiên cứu 3](#_Toc166783210)

[1.3. Phạm vi nghiên cứu 3](#_Toc166783211)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 4](#_Toc166783212)

[2.1. Khái niệm về bảo mật 4](#_Toc166783213)

[2.1.1. Khái niệm an toàn thông tin 4](#_Toc166783214)

[2.1.2. Khái niệm về khai thác lỗ hổng bảo mật 4](#_Toc166783215)

[2.1.3. Vấn đề an toàn thông tin của Việt Nam hiện nay 5](#_Toc166783216)

[2.2. Giới thiệu về Postman 6](#_Toc166783217)

[2.2.2. Lỗ hổng khi phát triển API 7](#_Toc166783218)

[2.3. Giải pháp vá lổ hổng 7](#_Toc166783219)

[2.4. Giới thiệu về bảo mật mã hóa 8](#_Toc166783220)

[CHƯƠNG 3. KHAI THÁC LỖ HỔNG WEBSITE VÀ GIẢI PHÁP BẢO MẬT MÃ HÓA 9](#_Toc166783221)

[3.1. Giới thiệu lỗ hống 9](#_Toc166783222)

[3.1.1. Khái niệm về lỗ hổng 9](#_Toc166783223)

[3.1.2. Phân loại lỗ hổng 9](#_Toc166783224)

[3.2. Thực hiện khai thác lỗ hổng 10](#_Toc166783225)

[3.2.1. Truy cập vào website QLHOCSINH 10](#_Toc166783226)

[3.2.2. Kiểm tra điều kiên ràng buộc của website QLHOCSINH 10](#_Toc166783228)

[3.2.3. Nhập liệu và lấy giá trị Request của website QLHOCSINH 11](#_Toc166783230)

[3.2.4. Sử dụng Postman để kiểm tra Request của website 12](#_Toc166783234)

[3.2.5. Thay đổi thông tin để kiểm tra website có bảo mật hay không 12](#_Toc166783236)

[3.2.6. Giải pháp khắc phục nhằm tăng cường bảo mật 13](#_Toc166783240)

[3.3. Thực hiện lỗ hổng XSS và đưa ra giải pháp bảo mật mã hóa 14](#_Toc166783244)

[3.3.1. Triển khai lỗ hổng 14](#_Toc166783245)

[3.3.2. Giải pháp bảo mật mã hóa 15](#_Toc166783248)

[3.3.3. Sau khi bảo mật mã hóa được áp dụng 15](#_Toc166783251)

[KẾT LUẬN 17](#_Toc166783255)

[TÀI LIỆU THAM KHẢO 18](#_Toc166783256)

BẢNG PHÂN CÔNG CỦA NHÓM

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên thành viên** | **Nội dung công việc** | **Kết quả** |
| 1 | Nguyễn Kim Hoàng | Nghiên cứu lỗ hổng | 100% |
| Xây dựng website | 100% |
| Thực hiện vá lỗ hổng | 100% |
| 2 | Trần Nguyễn Vũ Nam | Nghiên cứu lỗ hổng | 100% |
| Thực hiện vá lỗ hổng | 100% |
| 3 | Châu Minh Đương | Xây dựng Website | 100% |
| Viết báo cáo | 100% |
| 4 | Trịnh Xuân Đức | Nghiên cứu lỗ hổng | 100% |
| Viết báo cáo | 100% |

DANH MỤC HÌNH

[**Hình 2.1:** POSTMAN 6](#_Toc166783140)

[**Hình 3.1:** Truy cập vào website 10](#_Toc166783141)

[**Hình 3.2:** Kiểm tra điều kiên ràng buộc của website QLHOCSINH 10](#_Toc166783143)

[**Hình 3.3:** Nhập liệu và lấy giá trị Request của website QLHOCSINH 11](#_Toc166783145)

[**Hình 3.4:** Sử dụng Postman để kiểm tra Request của website 12](#_Toc166783146)

[**Hình 3.5:** Thay đổi thông tin để kiểm tra website có bảo mật hay không 13](#_Toc166783148)

[**Hình 3.6:** Giải pháp khắc phục nhằm tăng cường bảo mật 14](#_Toc166783149)

[**Hình 3.7:** Triển khai lỗ hổng 14](#_Toc166783150)

[**Hình 3.8:** Giải pháp bảo mật mã hóa 15](#_Toc166783151)

[**Hình 3.9:** Sau khi bảo mật mã hóa được áp dụng 16](#_Toc166783152)

MỞ ĐẦU

Trong bối cảnh công nghệ thông tin đang phát triển vượt bậc, việc bảo vệ thông tin trở thành một nhiệm vụ thiết yếu và không thể bỏ qua, đặc biệt là trong lĩnh vực bán hàng trực tuyến. Một trang web bán hàng trực tuyến không chỉ mang lại sự tiện lợi và trải nghiệm mua sắm nhanh chóng cho người tiêu dùng, mà còn lưu trữ nhiều thông tin nhạy cảm về khách hàng và các giao dịch. Từ đó, đặt ra một thách thức lớn về bảo mật thông tin, khi mà nguy cơ tấn công mạng và xâm phạm dữ liệu ngày càng tăng cao.

Xuất phát từ vấn đề này, chúng em quyết định chọn đề tài “Xây dựng giải pháp bảo mật cho website bán hàng DHND” cho bài báo cáo. Trong bài báo cáo, chúng em sẽ trình bày, phân tích các lỗ hổng bảo mật phổ biến mà một trang web bán hàng trực tuyến có thể gặp phải, từ đó xây dựng các giải pháp bảo mật hiệu quả nhằm ngăn chặn các cuộc tấn công tiềm ẩn đó.

Mong rằng, qua các nội dung trong báo cáo này, chúng em có thể mang lại một cái nhìn tổng quan và sâu sắc về vấn đề bảo mật thông tin trong lĩnh vực bán hàng trực tuyến, đồng thời cung cấp những giải pháp thực tiễn để bảo vệ thông tin một cách an toàn và hiệu quả, góp phần nâng cao độ tin cậy và uy tín của trang web bán hàng trực tuyến nói chung và website bán hàng DHND nói riêng.

1. GIỚI THIỆU CHUNG
   1. Lý do thực hiện đề tài

Với sự bùng nổ của các website bán hàng trực tuyến trong bối cảnh công nghệ và kinh tế thị trường ngày càng phát triển, việc tạo ra các hệ thống này không chỉ tăng cường hiệu quả quản lý mà còn cung cấp một cơ sở thuận lợi cho việc lưu trữ và xử lý thông tin của khách hàng. Tuy nhiên, cùng với những lợi ích đó, các hệ thống này cũng phải đối mặt với nhiều thách thức về an toàn và bảo mật thông tin. Trong tình hình này, việc chọn đề tài "Xây dựng giải pháp bảo mật cho website bán hàng DHND" là điều hợp lý và cần thiết. Bảo vệ thông tin nhạy cảm của khách hàng, như dữ liệu cá nhân và giao dịch, không chỉ đảm bảo sự riêng tư mà còn làm tăng độ tin cậy và uy tín của doanh nghiệp. Việc nghiên cứu và phát triển các giải pháp bảo mật hiệu quả sẽ giúp ngăn chặn các cuộc tấn công mạng và bảo vệ dữ liệu, đồng thời duy trì hoạt động ổn định của trang web.

* + 1. Hiện trạng

Các website bán hàng trực tuyến ngày nay đang phải đối mặt với một thách thức lớn từ các cuộc tấn công mạng, đặc biệt là khi giá trị mà chúng mang lại ngày càng tăng cao. Việc thiếu các biện pháp bảo mật cần thiết có thể dẫn đến những rủi ro không lường trước và tổn thất không kiểm soát khi bị tấn công. Trong số các lỗ hổng bảo mật phổ biến, Cross-Site Scripting (XSS) là một trong những mối đe dọa đáng lo ngại nhất.

Tấn công XSS mở ra cánh cửa cho tin tặc để chèn các đoạn mã độc hại vào trang web, từ đó có thể thu thập thông tin cá nhân của khách hàng hoặc chiếm quyền điều khiển phiên làm việc của họ. Hậu quả của việc này không chỉ dừng lại ở việc đánh cắp thông tin, mà còn ảnh hưởng đến uy tín của website và tạo ra sự lo lắng không cần thiết cho các khách hàng. Với mỗi cuộc tấn công thành công, không chỉ dữ liệu mà còn uy tín của doanh nghiệp đều có thể bị tổn thất nghiêm trọng, gây ra những hậu quả khó lường và khó khăn trong việc khôi phục lại niềm tin từ phía khách hàng. Do đó, việc đầu tư và triển khai các biện pháp bảo mật mạnh mẽ trở thành điều cần thiết để bảo vệ cả dữ liệu và uy tín của các website bán hàng trực tuyến trong môi trường mạng ngày càng phức tạp và nguy hiểm này.

* + 1. Yêu cầu bảo mật

Trước những nguy cơ hiện hữu, việc bảo đảm an toàn thông tin cho website bán hang trực tuyến trở nên cấp thiết hơn bao giờ hết. Các yêu cầu bảo mật cơ bản bao gồm:

* **Ngăn chặn các hình thức tấn công:** Đặc biệt là tấn công XSS, SQL Injection, CSRF, và các loại tấn công khác.
* **Bảo vệ dữ liệu nhạy cảm:** Đảm bảo thông tin cá nhân của học sinh không bị truy cập trái phép.
* **Đảm bảo tính toàn vẹn và bảo mật của dữ liệu:** Dữ liệu phải được bảo vệ trong suốt quá trình nhập liệu, lưu trữ, và truy xuất.
* **Tăng cường các và đưa ra các biện pháp bảo mật đầu vào:** Xử lý và mã hóa các đầu vào từ người dùng để ngăn chặn việc chèn mã độc.
* **Kiểm soát và giám sát các hoạt động trên hệ thống:** Theo dõi và kiểm tra các hoạt động trên hệ thống để phát hiện và phản ứng kịp thời với các hành vi đáng ngờ hoặc không hợp lệ. Sử dụng các công cụ giám sát và cảnh báo để bảo vệ hệ thống khỏi các mối đe dọa tiềm tàng và đảm bảo hệ thống luôn ở trạng thái an toàn.

Việc nghiên cứu và áp dụng các biện pháp mã hóa để bảo vệ hệ thống khỏi tấn công XSS là một trong những giải pháp hiệu quả nhằm nâng cao tính bảo mật cho website bán hàng. Điều này không chỉ bảo vệ dữ liệu khách hàng mà còn giúp website duy trì uy tín và sự tin tưởng của khách hàng.

* 1. Tổng quan về vấn đề nghiên cứu

Đề tài "Xây dựng giải pháp bảo mật cho website bán hàng DHND" nhằm mục đích:

* Phân tích và xác định các lỗ hổng bảo mật hiện có trên website bán hàng.
* Thử nghiệm tấn công XSS để minh họa mức độ nguy hiểm của lỗ hổng.
* Ngăn chặn các hành vi đánh cắp mã nguồn.
* Đảm bảo lưu lượng truy cập, tránh tình trạng quá tải hệ thống.
* Đề xuất và triển khai giải pháp vá lỗ hổng để xử lý các đầu vào từ người dùng, ngăn chặn tấn công XSS.
* Đánh giá hiệu quả của giải pháp đã đề xuất thông qua các thử nghiệm thực tế.
  1. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài bao gồm:

* Kiểm tra và xác định các lỗ hổng bảo mật trên website DHND hiện tại.
* Đề xuất các biện pháp bảo mật để bảo vệ mã nguồn, quyền truy cập và hệ thống khỏi các mối đe dọa tiềm ẩn. Đồng thời, đảm bảo lưu lượng truy cập trên hệ thống, tránh tình trạng quá tải diễn ra.
* Thực hiện các thử nghiệm tấn công XSS để minh chứng các lỗ hổng.
* Đề xuất và triển khai giải pháp mã hóa để xử lý các đầu vào từ người dùng nhằm ngăn chặn tấn công XSS.
* Đánh giá hiệu quả của giải pháp thông qua các thử nghiệm thực tế sau khi áp dụng.

Phạm vi nghiên cứu tập trung chủ yếu vào việc giải quyết vấn đề XSS, nhưng cũng xem xét các biện pháp bảo mật bổ sung để tăng cường toàn diện an toàn thông tin cho hệ thống.

1. CƠ SỞ LÝ THUYẾT
   1. Khái niệm về bảo mật
      1. Khái niệm an toàn thông tin

An toàn thông tin (ATTT) là một lĩnh vực quan trọng trong công nghệ thông tin, tập trung vào việc bảo vệ thông tin khỏi các mối đe dọa, rủi ro và tấn công. ATTT đảm bảo thông tin được bảo vệ và duy trì tính bảo mật, toàn vẹn và sẵn sàng.

* **Tính bảo mật (Confidentiality)**: Đảm bảo thông tin chỉ được truy cập bởi những người được ủy quyền và không bị lộ ra ngoài một cách trái phép.
* **Tính toàn vẹn (Integrity)**: Đảm bảo thông tin không bị sửa đổi một cách trái phép, và chỉ những người được ủy quyền mới có thể thay đổi dữ liệu.
* **Tính sẵn sàng (Availability)**: Đảm bảo thông tin luôn có sẵn cho những người dùng được ủy quyền khi cần thiết, tránh các sự cố gây gián đoạn dịch vụ hoặc truy cập thông tin.
* **Tính chống thoái thác (Non-repudiation)**: Khả năng ngăn chặn việc từ chối một hành vi đã thực hiện trên hệ thống thông tin

ATTT đóng vai trò quan trọng trong việc bảo vệ tài sản thông tin, bảo vệ quyền riêng tư của cá nhân và tổ chức, và duy trì sự tin cậy trong hoạt động truyền thông và giao tiếp. Các biện pháp và kỹ thuật ATTT cung cấp cơ sở cho việc phát triển và vận hành các hệ thống thông tin an toàn và đáng tin cậy.

* + 1. Khái niệm về khai thác lỗ hổng bảo mật

Khai thác lỗ hổng bảo mật là quá trình tìm kiếm và lợi dụng các điểm yếu hoặc lỗ hổng trong hệ thống thông tin để thực hiện các hành vi trái phép. Kẻ tấn công có thể sử dụng các lỗ hổng này để truy cập trái phép vào hệ thống, đánh cắp dữ liệu, phá hoại hệ thống hoặc thực hiện các hành vi gian lận khác. Các loại lỗ hổng bảo mật phổ biến có thể kể đến như sau:

* **Lỗ hổng phần mềm:** Những lỗi hoặc sai sót trong mã nguồn của phần mềm có thể bị lợi dụng để tấn công. Ví dụ: lỗi tràn bộ đệm (Buffer Overflow), lỗi định dạng chuỗi (Format String Vulnerability), và các lỗi logic.
* **Lỗ hổng cấu hình:** Các cấu hình sai hoặc không an toàn trong hệ thống có thể tạo ra các lỗ hổng bảo mật. Ví dụ: sử dụng mật khẩu mặc định, mở cổng không cần thiết, quyền truy cập không cần thiết và không cập nhật các bản vá bảo mật.
* **Lỗ hổng trong giao thức mạng:** Các lỗi hoặc yếu kém trong các giao thức truyền thông mạng có thể bị khai thác để tấn công. Ví dụ: các lỗ hổng trong giao thức HTTP, SSL/TLS, DNS và các loại tấn công như ARP Spoofing, "Man in the Middle".

Quá trình khai thác lỗ hổng thường bắt đầu bằng việc phát hiện lỗ hổng (thông qua kiểm tra bảo mật hoặc các công cụ quét lỗ hổng), sau đó sử dụng các kỹ thuật và công cụ để lợi dụng lỗ hổng đó. Một số kỹ thuật khai thác phổ biến bao gồm:

* **Cross-Site Scripting (XSS)**: Kẻ tấn công chèn mã JavaScript độc hại vào trang web, cho phép họ thực thi mã độc trên trình duyệt của người dùng. Ví dụ: Năm 2017, một lỗ hổng XSS trong Disqus cho phép kẻ tấn công đánh cắp cookie của người dùng và chiếm quyền truy cập vào các tài khoản của họ.
* **SQL Injection:** Kẻ tấn công chèn các câu lệnh SQL độc hại vào các truy vấn cơ sở dữ liệu, cho phép họ truy cập hoặc thay đổi dữ liệu trái phép. Ví dụ: Năm 2013, một lỗ hổng SQL Injection trong trang web của Target đã dẫn đến việc rò rỉ dữ liệu cá nhân của hơn 40 triệu khách hàng.
* **Phishing:** Kẻ tấn công tạo ra các trang web hoặc email giả mạo để đánh lừa người dùng tiết lộ thông tin cá nhân như mật khẩu hoặc thông tin thẻ tín dụng. Ví dụ: Năm 2016, một chiến dịch phishing nhắm vào nhân viên của Google đã đánh cắp thông tin đăng nhập của họ và truy cập vào các hệ thống nội bộ của công ty.
  + 1. Vấn đề an toàn thông tin của Việt Nam hiện nay

An toàn thông tin đang trở thành một vấn đề ngày càng quan trọng không chỉ trên thế giới mà còn tại Việt Nam. Trong thời đại số hóa hiện nay, việc bảo vệ thông tin cá nhân và dữ liệu quan trọng trở nên cấp bách hơn bao giờ hết. Tại Việt Nam, những vấn đề liên quan đến an toàn thông tin không chỉ đặt ra những thách thức mà còn cung cấp những cơ hội mới cho sự phát triển kinh tế và xã hội.

Việt Nam đang trải qua một giai đoạn chuyển đổi số mạnh mẽ, với việc phát triển các lĩnh vực công nghệ thông tin, viễn thông, và internet. Tuy nhiên, điều này cũng đồng nghĩa với việc tăng cường rủi ro về an ninh mạng và an toàn thông tin. Các cuộc tấn công mạng, vi phạm dữ liệu và lừa đảo trực tuyến đang trở nên phổ biến hơn, đe dọa sự ổn định và sự tin cậy của hệ thống thông tin tại Việt Nam.

Theo tổng hợp của NCS, năm 2023 đã ghi nhận 13.900 vụ tấn công mạng vào các tổ chức tại Việt Nam, trung bình mỗi tháng xảy ra 1.160 vụ, tăng 9,5% so với năm 2022. Các loại tấn công phổ biến bao gồm: lừa đảo, mã độc, zero-day,.. nhắm vào các cơ quan chính phủ, doanh nghiệp, tổ chức tài chính và người dùng cá nhân. Nguyên nhân chủ yếu là do hệ thống bảo mật yếu kém, thiếu hụt nhân lực an ninh mạng có trình độ chuyên môn cao và ý thức bảo mật của người dùng còn thấp.

Đặc biệt, trong 3 tháng cuối năm 2023 số vụ tấn công mạng tăng mạnh, lên tới 1.614 vụ trong 1 tháng, gấp rưỡi so với trung bình. Nguyên nhân vì thời điểm cuối năm, các cơ quan, doanh nghiệp, tổ chức có nhiều dự án CNTT cần hoàn thành, nhân sự thường phải hoạt động trên 100% năng suất nên khả năng xảy ra nhiều sai sót, đây cũng là cơ hội để hacker có thể tấn công, phá hoại.

Việt Nam đã thực hiện nhiều biện pháp pháp lý và chính sách nhằm tăng cường an ninh mạng và bảo vệ thông tin cá nhân cũng như dữ liệu. Trong đó, Luật An ninh mạng được ban hành vào năm 2018 đã tạo ra một khuôn khổ pháp lý cụ thể cho công tác này. Chính phủ cũng đã đưa ra nhiều chiến lược, đề án và chương trình nhằm mục tiêu tăng cường an ninh mạng và bảo vệ thông tin cá nhân cũng như dữ liệu quan trọng. Các cơ quan chức năng của Việt Nam chịu trách nhiệm quan trọng trong việc thực hiện công tác an ninh mạng. Trong đó, Cục An ninh mạng và Phòng chống tội phạm sử dụng công nghệ cao (Bộ Công an) là cơ quan đầu mối thực hiện công tác an ninh mạng. Bên cạnh đó, Bộ Thông tin và Truyền thông, cùng với Bộ Khoa học và Công nghệ, cũng đóng vai trò quan trọng trong việc bảo đảm an toàn thông tin.

Mặc dù năng lực kỹ thuật về an ninh mạng tại Việt Nam đang được nâng cao, nhưng vẫn tồn tại nhiều hạn chế. Sự thiếu hụt nhân lực an ninh mạng có trình độ chuyên môn cao và kinh nghiệm thực tiễn là một trong những thách thức đáng chú ý. Tuy nhiên, Việt Nam đã tích cực tham gia vào các hoạt động hợp tác quốc tế trong lĩnh vực an ninh mạng, bao gồm chia sẻ thông tin và kinh nghiệm với các quốc gia khác, cũng như tham gia vào các tổ chức quốc tế như Tổ chức An ninh mạng Quốc tế (OIC) và Liên minh An ninh mạng Châu Á - Thái Bình Dương (APAC CERT).

Để đáp ứng các thách thức trong bối cảnh ngày càng phức tạp của an ninh mạng, Việt Nam đang tăng cường nhận thức về an ninh mạng cho người dân và doanh nghiệp, đồng thời đầu tư vào hệ thống an ninh mạng và phát triển nguồn nhân lực có trình độ chuyên môn cao. Hơn nữa, việc tăng cường hợp tác quốc tế cũng được xem là một phần không thể thiếu trong chiến lược bảo vệ an ninh mạng của Việt Nam...

* 1. Giới thiệu về Postman

Postman là một công cụ phổ biến dùng để phát triển và kiểm thử API.

A logo with a person in a circle

Description automatically generated

* + - 1. POSTMAN
* **Giao diện người dùng trực quan:** Giúp tạo, sửa đổi và quản lý các yêu cầu API dễ dàng.
* **Hỗ trợ đa phương thức HTTP:** GET, POST, PUT, DELETE, PATCH, OPTIONS, HEAD.
* **Quản lý bộ sưu tập (Collections):** Tổ chức các yêu cầu thành các bộ sưu tập, giúp quản lý và tái sử dụng các yêu cầu API.
* **Môi trường (Environments):** Tạo và quản lý các môi trường khác nhau (phát triển, thử nghiệm, sản xuất), giúp dễ dàng chuyển đổi giữa các cấu hình.
* **Kiểm tra tự động (Automated Testing):** Viết các kịch bản kiểm tra tự động bằng JavaScript, giúp xác thực tính đúng đắn của các API phản hồi.
* **Tích hợp CI/CD:** Tích hợp với các công cụ CI/CD như Jenkins, CircleCI, Travis CI, giúp tự động hóa quá trình kiểm thử API.
* **Ghi lại yêu cầu (Request Capture):** Ghi lại các yêu cầu HTTP từ trình duyệt hoặc ứng dụng khác để phân tích và kiểm tra.
  + 1. Lỗ hổng khi phát triển API

Trong quá trình phát triển API, có một số lỗ hổng bảo mật phổ biến cần được nhà phát triển chú ý và giải quyết để đảm bảo tính bảo mật của hệ thống.

* **Cross-Site Scripting (XSS):** Lỗ hổng XSS cho phép kẻ tấn công chèn các đoạn mã JavaScript độc hại vào trang web hoặc ứng dụng, từ đó có thể thực thi các hành động không mong muốn trên trình duyệt của người dùng, như đánh cắp cookie, thông tin đăng nhập, hoặc thậm chí kiểm soát trang web. Điều này có thể gây ra các hậu quả nghiêm trọng như đánh cắp thông tin cá nhân hoặc chiếm quyền kiểm soát toàn bộ trang web.
* **Không xác thực hoặc phân quyền không an toàn:** Việc thiếu kiểm tra xác thực hoặc phân quyền đúng đắn có thể dẫn đến việc kẻ tấn công truy cập trái phép vào các tài nguyên hoặc chức năng của ứng dụng, gây ra sự đe dọa đến tính toàn vẹn và bảo mật của dữ liệu.
* **Lỗ hổng không kiểm tra đầu vào đúng cách:** Nếu không kiểm tra và xử lý đầu vào từ người dùng một cách cẩn thận, ứng dụng có thể trở nên dễ bị tấn công, như chèn mã độc, đầu vào không hợp lệ hoặc các kỹ thuật tấn công khác.
  1. Giải pháp vá lổ hổng

Để đảm bảo tính bảo mật của hệ thống API, cần áp dụng các giải pháp sau để vá lỗ hổng:

* **Giải pháp cho lỗ hổng Cross-Site Scripting (XSS):**
* Sử dụng các hàm mã hóa đúng cách: Áp dụng các hàm mã hóa như encodeURIComponent() để mã hóa dữ liệu trước khi hiển thị trên trang web, từ đó ngăn chặn việc thực thi các đoạn mã JavaScript độc hại.
* Sử dụng Content Security Policy (CSP): Thiết lập CSP để hạn chế các nguồn tài nguyên được tải và thực thi, giúp ngăn chặn việc chèn mã JavaScript từ các nguồn không tin cậy.
* **Giải pháp cho lỗ hổng không xác thực hoặc phân quyền không an toàn:**
* Thiết lập cơ chế xác thực mạnh mẽ: Sử dụng các phương pháp xác thực như OAuth, JWT để đảm bảo tính bảo mật của hệ thống.
* Kiểm tra phân quyền đúng đắn: Đảm bảo rằng mỗi yêu cầu được kiểm tra phân quyền trước khi được xử lý, và chỉ cho phép truy cập vào các tài nguyên được ủy quyền.
  1. Giới thiệu về bảo mật mã hóa

Mã hóa là một trong những phương pháp quan trọng nhất để đảm bảo tính bảo mật của dữ liệu trong các hệ thống thông tin.

* **Mã hóa:** Mã hóa là quá trình biến đổi dữ liệu từ dạng rõ ràng thành dạng không đọc được (ciphertext) bằng cách sử dụng một thuật toán mã hóa và một khóa. Mục đích chính của mã hóa là ngăn chặn việc truy cập không ủy quyền vào dữ liệu bằng cách biến dữ liệu trở nên không đọc được cho bất kỳ ai không có khóa.
* **Khóa mã hóa:** Khóa mã hóa là một chuỗi dữ liệu hoặc thông tin được sử dụng cùng với thuật toán mã hóa để biến đổi dữ liệu. Mỗi khóa có thể được sử dụng để mã hóa hoặc giải mã dữ liệu, tùy thuộc vào thuật toán mã hóa được sử dụng.
* **Thuật toán mã hóa:** Là một quy trình cụ thể được áp dụng để thực hiện mã hóa hoặc giải mã dữ liệu. Các thuật toán mã hóa có thể được chia thành hai loại chính: mã hóa đối xứng (symmetric encryption) và mã hóa không đối xứng (asymmetric encryption).
* **Mã hóa đối xứng:** Trong mã hóa đối xứng, cùng một khóa được sử dụng cả cho quá trình mã hóa và giải mã dữ liệu. Các thuật toán phổ biến của mã hóa đối xứng bao gồm AES (Advanced Encryption Standard), DES (Data Encryption Standard) và 3DES (Triple DES).
* **Mã hóa không đối xứng:** Trong mã hóa không đối xứng, có hai khóa liên quan với nhau: khóa công khai (public key) và khóa bí mật (private key). Dữ liệu được mã hóa bằng khóa công khai và chỉ có thể được giải mã bằng khóa bí mật tương ứng, và ngược lại. RSA (Rivest-Shamir-Adleman) là một trong những thuật toán mã hóa không đối xứng phổ biến nhất.
* **Sử dụng trong bảo mật thông tin:** Mã hóa được sử dụng trong nhiều lĩnh vực của bảo mật thông tin, bao gồm bảo mật dữ liệu truyền qua mạng, bảo mật dữ liệu lưu trữ, xác thực và chứng thực, và nhiều ứng dụng khác.
  1. Khái niệm Validation website:

Validation website (hay còn gọi là xác thực dữ liệu website) là quá trình kiểm tra xem dữ liệu người dùng nhập vào website có hợp lệ hay không. Dữ liệu này có thể được nhập vào thông qua các form, input, textarea,...

Validation website thường được thực hiện bằng cách sử dụng các quy tắc validation được định nghĩa trước. Các quy tắc này có thể dựa trên nhiều tiêu chí khác nhau như:

* **Kiểu dữ liệu:** Kiểm tra xem dữ liệu nhập vào có đúng kiểu dữ liệu hay không (ví dụ: số nguyên, số thực, email,...)
* **Độ dài:** Kiểm tra xem độ dài của dữ liệu nhập vào có nằm trong phạm vi cho phép hay không
* **Định dạng:** Kiểm tra xem định dạng của dữ liệu nhập vào có đúng hay không (ví dụ: định dạng ngày tháng, định dạng email,...)
* **Giá trị:** Kiểm tra xem giá trị của dữ liệu nhập vào có nằm trong phạm vi cho phép hay không (ví dụ: giá trị tuổi phải lớn hơn 18)

Việc sử dụng Validation trong website nhằm nhiều mục đích như:

* **Cải thiện trải nghiệm người dùng:** Validation giúp đảm bảo rằng người dùng nhập dữ liệu chính xác và đầy đủ, từ đó giúp họ hoàn thành các thao tác trên website một cách nhanh chóng và dễ dàng hơn.
* **Giảm thiểu lỗi:** Validation giúp ngăn chặn người dùng nhập dữ liệu sai, từ đó giúp giảm thiểu lỗi hệ thống và nâng cao hiệu quả hoạt động của website.
* **Tăng cường bảo mật:** Validation giúp ngăn chặn các hành vi tấn công website, ví dụ như tấn công SQL injection hoặc cross-site scripting.
* **Thu thập dữ liệu chính xác:** Validation giúp đảm bảo rằng dữ liệu người dùng thu thập được là chính xác và có thể sử dụng cho các mục đích khác nhau như phân tích dữ liệu, marketing,...
  1. Khái niệm mã hóa MD5:

MD5 (viết tắt của Message Digest Algorithm 5) là một thuật toán băm (hashing function) được sử dụng để mã hóa dữ liệu thành một chuỗi ký tự có độ dài cố định (128 bit). Thuật toán này được phát triển bởi Rivest (R), MD (Mandelbrot) và Colin (C) vào năm 1991.

Mã hóa MD5 được sử dụng cho nhiều mục đích khác nhau, bao gồm:

* **Kiểm tra tính toàn vẹn dữ liệu:** MD5 có thể được sử dụng để kiểm tra xem dữ liệu có bị thay đổi hay không trong quá trình truyền tải hoặc lưu trữ.
* **Bảo mật mật khẩu:** MD5 có thể được sử dụng để mã hóa mật khẩu người dùng, giúp bảo vệ mật khẩu khỏi bị đánh cắp.
* **Xác thực tập tin:** MD5 có thể được sử dụng để xác minh tính toàn vẹn của tập tin, đảm bảo rằng tập tin không bị thay đổi hoặc bị hỏng.
* **Theo dõi tệp tin:** MD5 có thể được sử dụng để theo dõi các thay đổi đối với tệp tin, giúp xác định ai đã thay đổi tệp tin và khi nào.

Mã hóa MD5 mang 1 số ưu điểm nổi bật như:

* Dễ dàng sử dụng và triển khai.
* Tốc độ tính toán nhanh.
* Kích thước giá trị băm nhỏ gọn.
* Được hỗ trợ rộng rãi bởi nhiều ngôn ngữ lập trình và phần mềm.

1. KHAI THÁC LỖ HỔNG WEBSITE VÀ GIẢI PHÁP BẢO MẬT MÃ HÓA
   1. Giới thiệu lỗ hống
      1. Khái niệm về lỗ hổng

Lỗ hổng bảo mật (tiếng Anh: vulnerability) là một điểm yếu, thiếu sót hoặc sai sót trong hệ thống, ứng dụng, quy trình hoặc giao thức mà tin tặc có thể lợi dụng để xâm nhập, tấn công hoặc gây hại cho hệ thống mà không cần có quyền truy cập hợp lệ. Lỗ hổng có thể tồn tại ở mọi cấp độ của hệ thống, từ phần mềm, phần cứng, mạng, hệ điều hành, giao thức mạng đến con người.

* + 1. Phân loại lỗ hổng

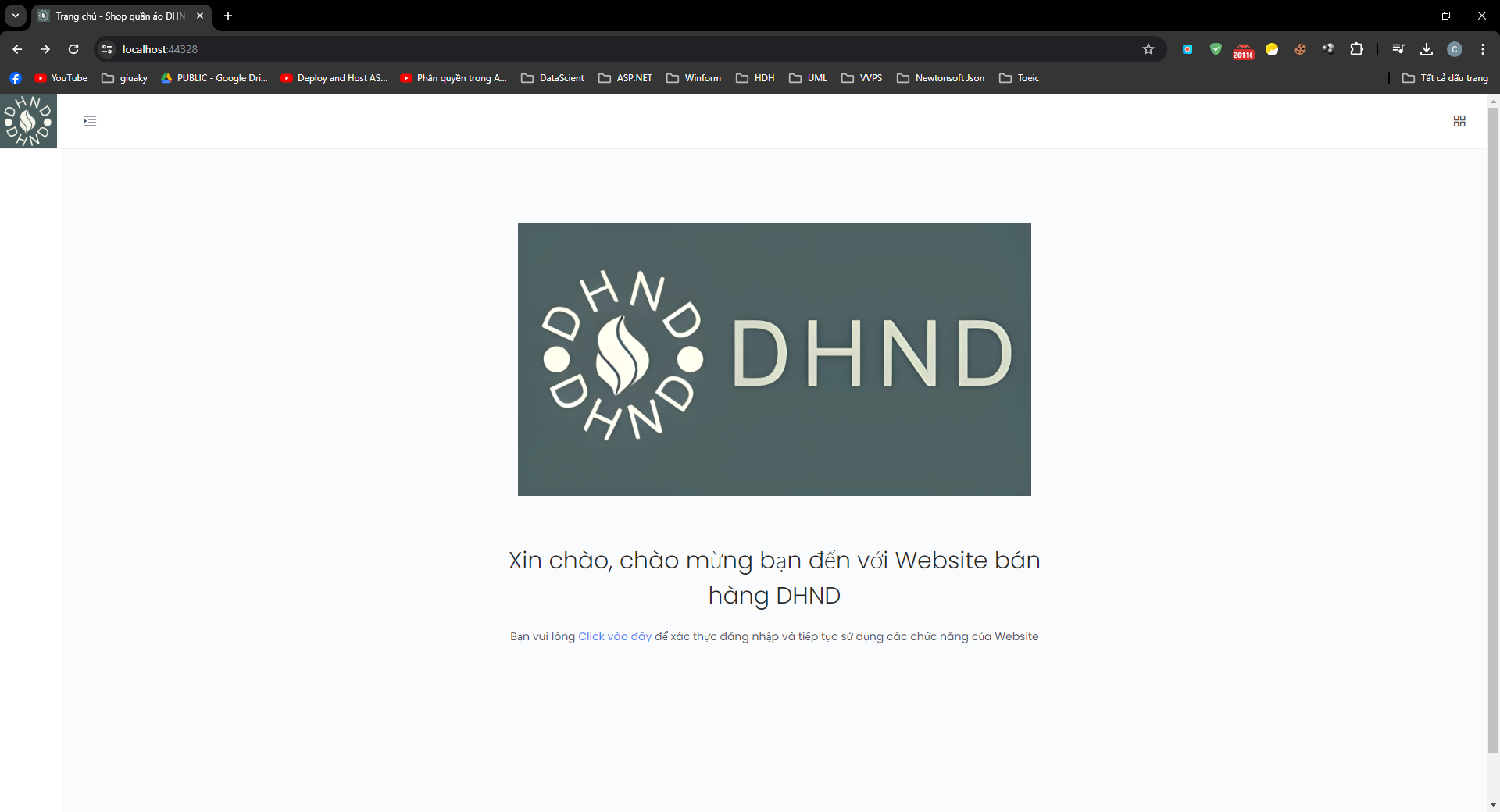
Lỗ hổng bảo mật có thể được phân loại theo nhiều tiêu chí khác nhau, nhưng hai cách phân loại chính thường được sử dụng là:

1. **Theo cách thức tấn công:**

* **Loại 1:** Lỗ hổng cho phép thực thi mã độc
* **Lỗ hổng tràn bộ đệm (Buffer Overflow)**: Xảy ra khi hacker gửi quá nhiều dữ liệu vào bộ đệm bộ nhớ, khiến dữ liệu tràn sang các vùng bộ nhớ khác và có thể thực thi mã độc.
* **Lỗi định dạng chuỗi (Format String Vulnerability)**: Xảy ra khi phần mềm không kiểm tra đầu vào dữ liệu một cách cẩn thận, cho phép hacker chèn mã độc vào chương trình.
* **Lỗ hổng SQL Injection**: Xảy ra khi hacker chèn các câu lệnh SQL độc hại vào các truy vấn cơ sở dữ liệu, cho phép họ truy cập hoặc thay đổi dữ liệu trái phép.
* **Lỗ hổng Command Injection**: Xảy ra khi hacker chèn các lệnh hệ thống độc hại vào các ứng dụng, cho phép họ thực thi các lệnh trên hệ thống.
* **Loại 2:** Lỗ hổng cho phép thao túng dữ liệu
* **Lỗ hổng Cross-Site Scripting (XSS)**: Cho phép hacker chèn mã JavaScript độc hại vào trang web hoặc ứng dụng để thực hiện các hành động không mong muốn, ví dụ như đánh cắp cookie, thông tin đăng nhập hoặc thực hiện các hành động thay người dùng.
* **Lỗ hổng Cross-Site Request Forgery (CSRF)**: Cho phép hacker thực hiện các yêu cầu HTTP giả mạo từ người dùng hợp lệ đến ứng dụng, đe dọa tính toàn vẹn và bảo mật của dữ liệu.
* **Lỗ hổng Insecure Direct Object References (IDOR)**: Xảy ra khi ứng dụng cho phép hacker truy cập trực tiếp vào các đối tượng mà không cần kiểm tra quyền truy cập hợp lệ.
* **Loại 3:** Lỗ hổng cho phép truy cập trái phép
* **Lỗ hổng cấu hình sai**: Xảy ra khi hệ thống hoặc ứng dụng được cấu hình sai, tạo ra các điểm yếu mà hacker có thể khai thác.
* **Lỗ hổng Zero-Day (Zero-Day Vulnerability)**: Là những lỗ hổng chưa được nhà cung cấp phần mềm biết đến và chưa có bản vá lỗi. Hacker có thể khai thác những lỗ hổng này để tấn công hệ thống trước khi nhà cung cấp phần mềm có thể khắc phục.

1. **Theo cấp độ nghiêm trọng:**

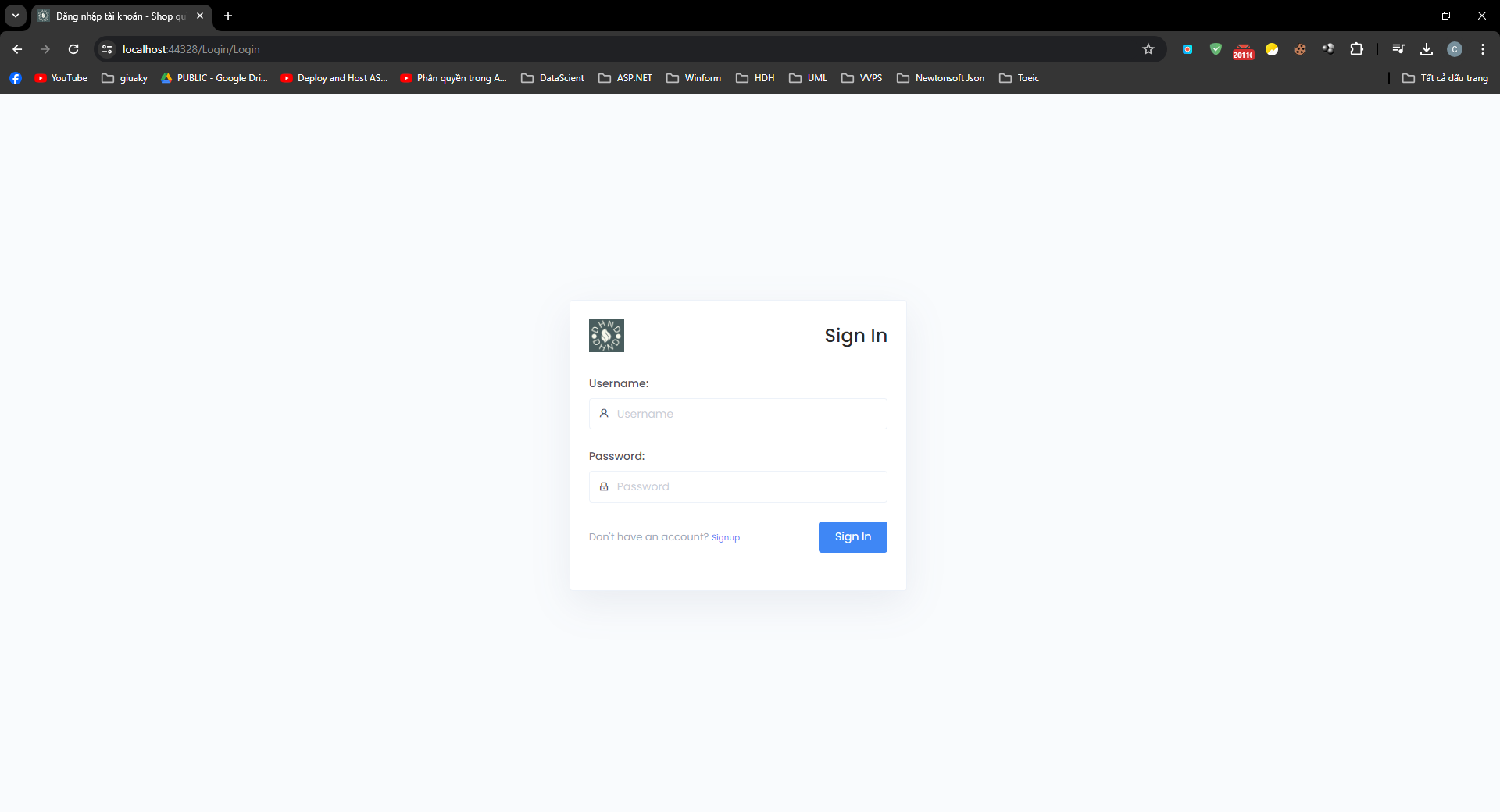
* **Lỗ hổng cấp độ thấp:** Có ít tác động đến hệ thống hoặc có thể khắc phục một cách dễ dàng.
* **Lỗ hổng cấp độ trung bình:** Có tác động vừa phải đến hệ thống và cần được khắc phục ngay khi có thể.
* **Lỗ hổng cấp độ cao:** Có tác động nghiêm trọng đến hệ thống và cần được ưu tiên khắc phục ngay lập tức để tránh các hậu quả không mong muốn.
  1. Thực hiện các giải pháp bảo mật
     1. Truy cập vào website Shop quần áo DHND



* + - 1. Truy cập vào website

Chúng ta sẽ truy cập vào website và nhập liệu các mục trong phần thêm mới học sinh để kiểm tra xem website đã có các điều kiện ràng buộc chưa

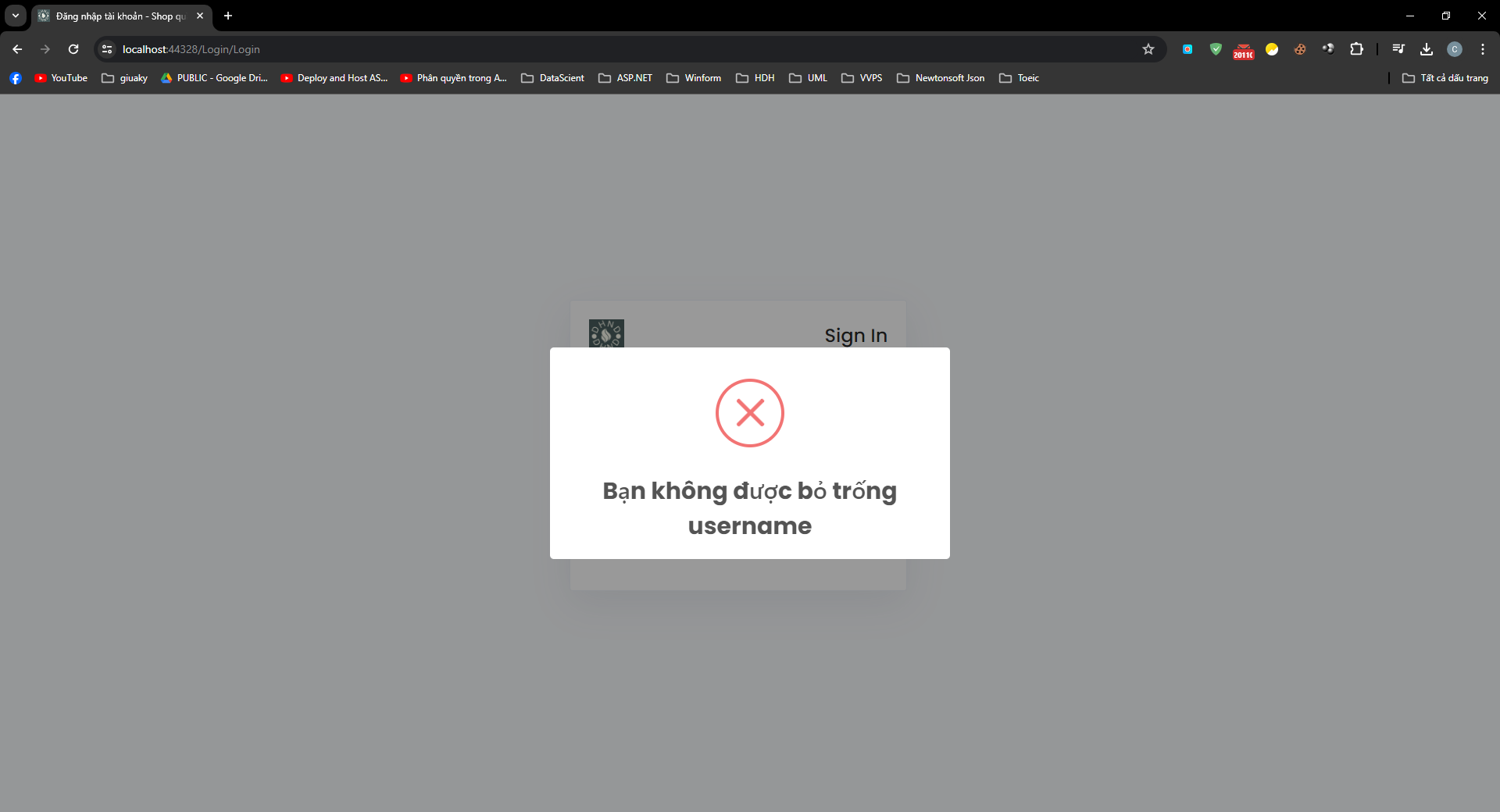
* + 1. Truy cập vào Login

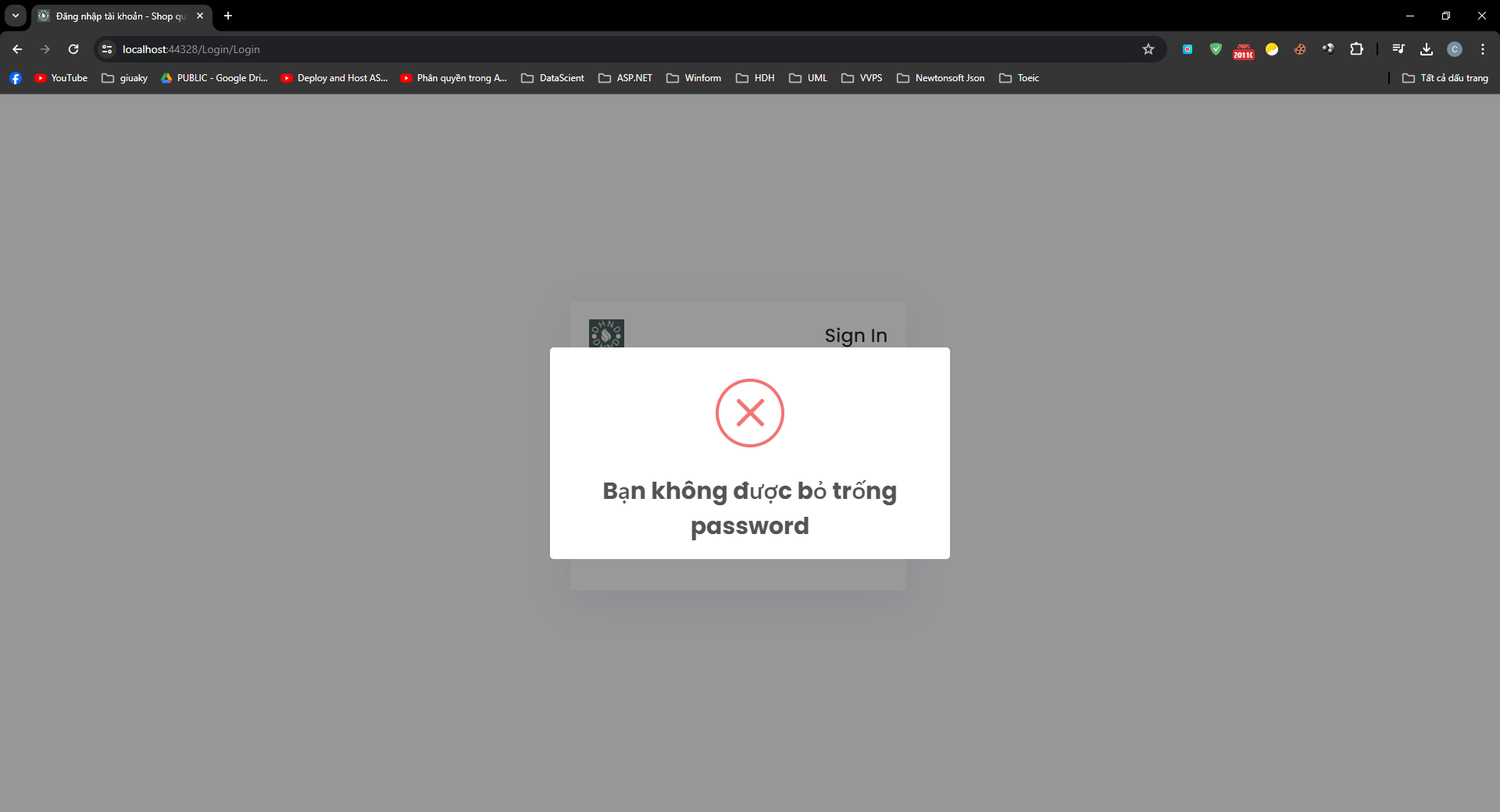


* + - 1. Kiểm tra điều kiên ràng buộc của website QLHOCSINH

Qua đây chúng ta có thể thấy được website đã có sự ràng buộc trong các ô nhập liệu của hệ thống

* + 1. Check Validation trước khi đăng nhập

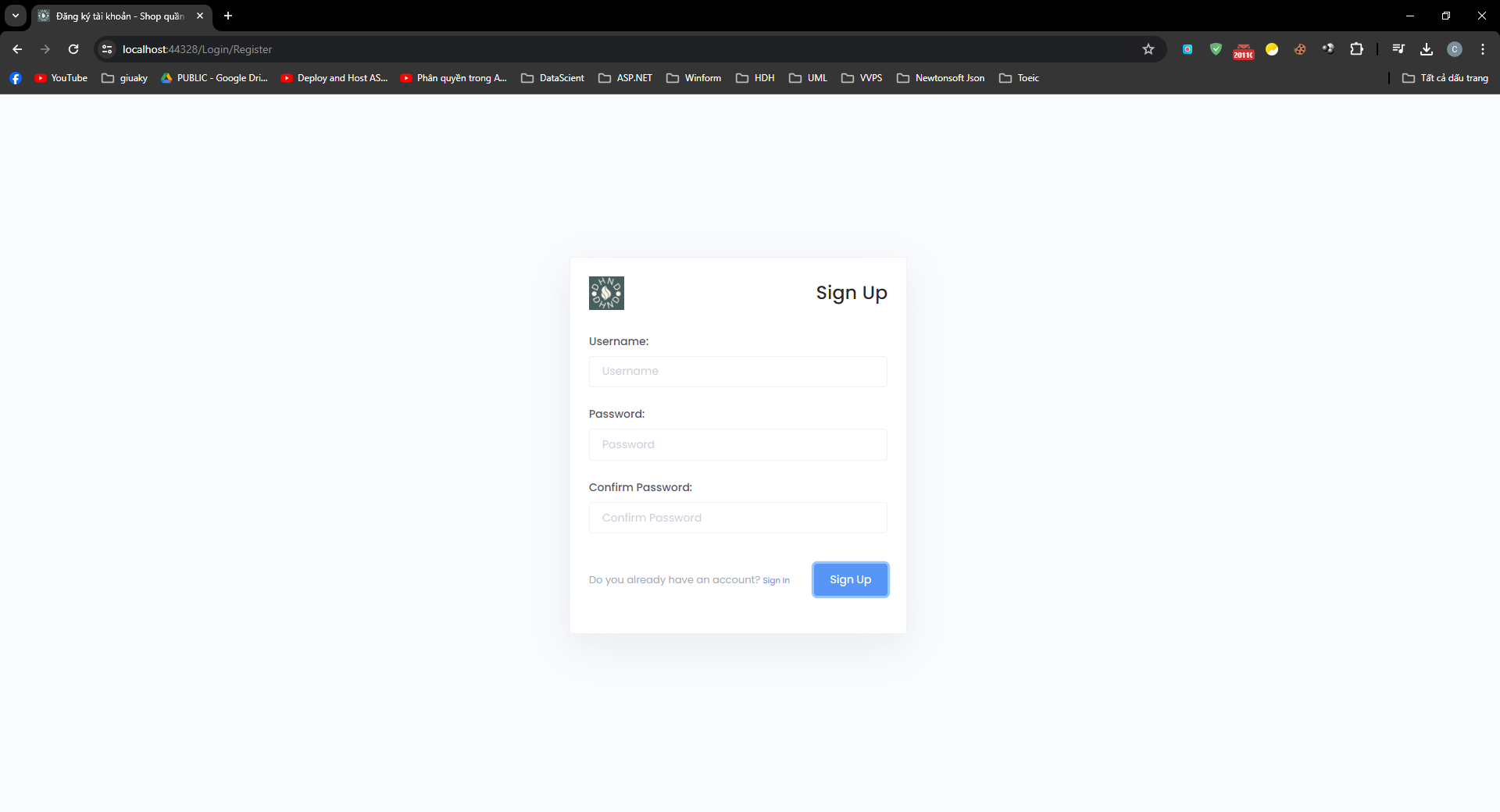


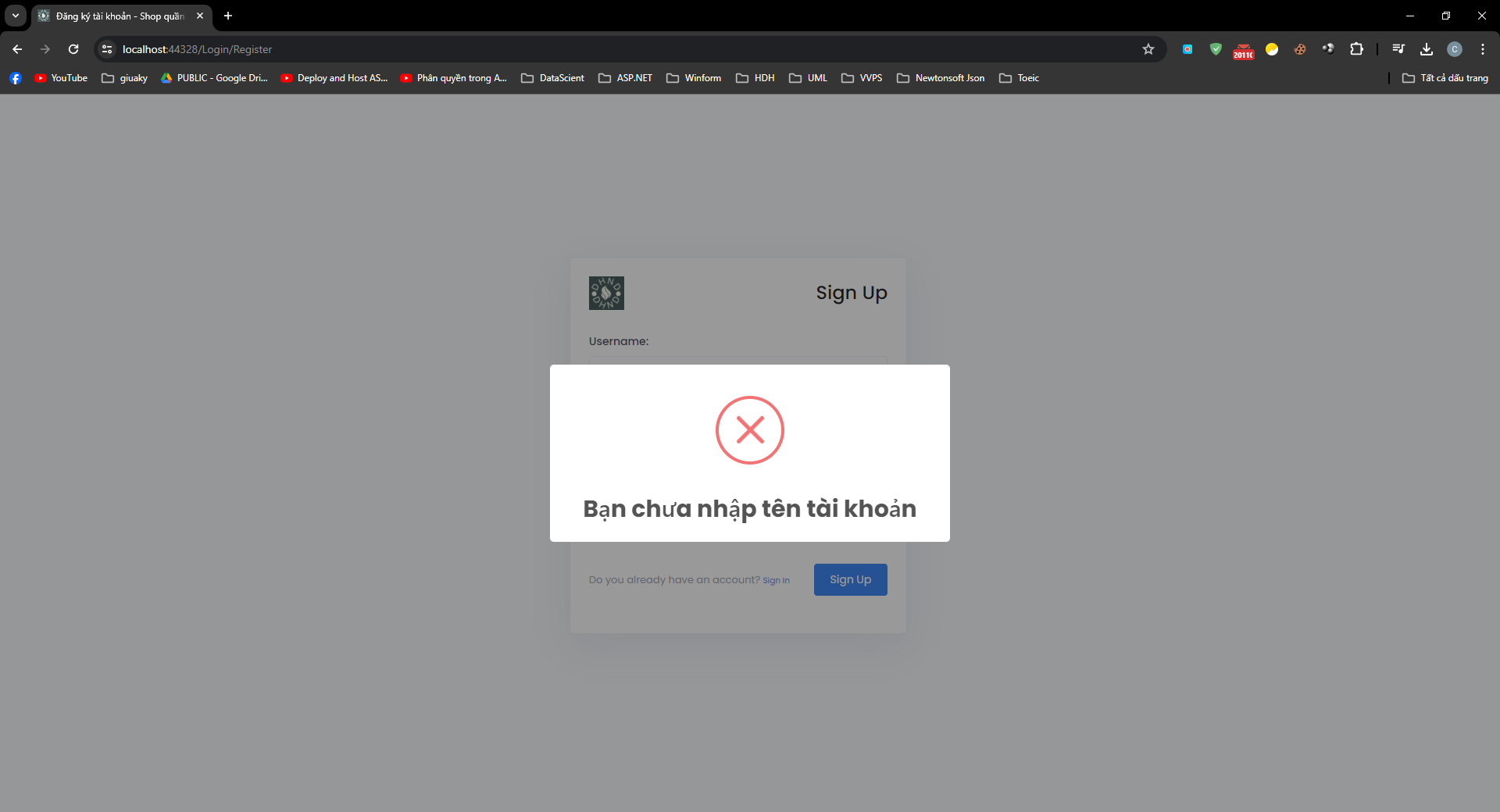


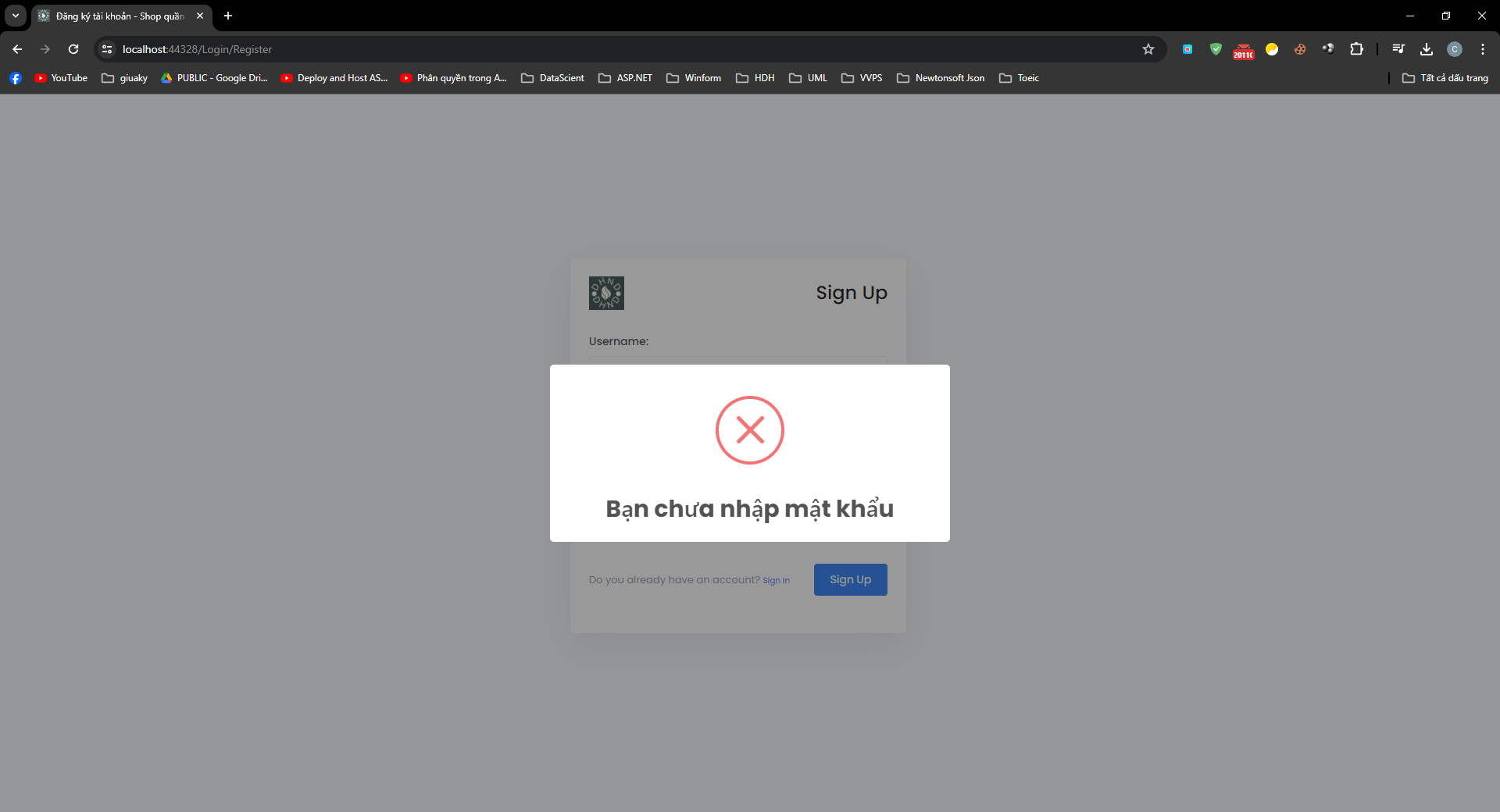
* + - 1. Nhập liệu và lấy giá trị Request của website QLHOCSINH

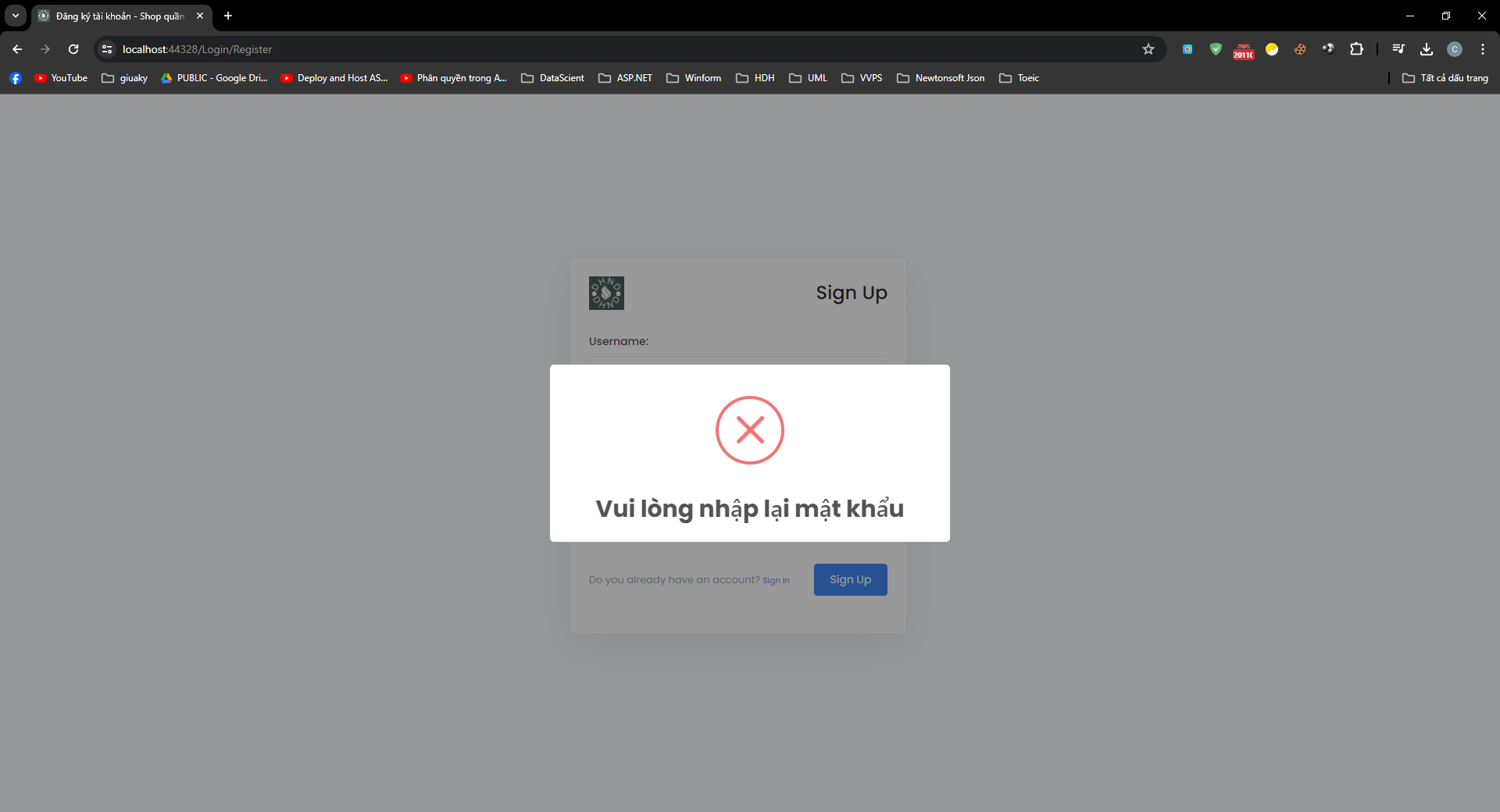
Như chúng ta thấy khi nhập các thông tin vào trong form nhập liệu của website và nhấn “Thêm mới” chúng ta có thể thấy được Request URL của website.

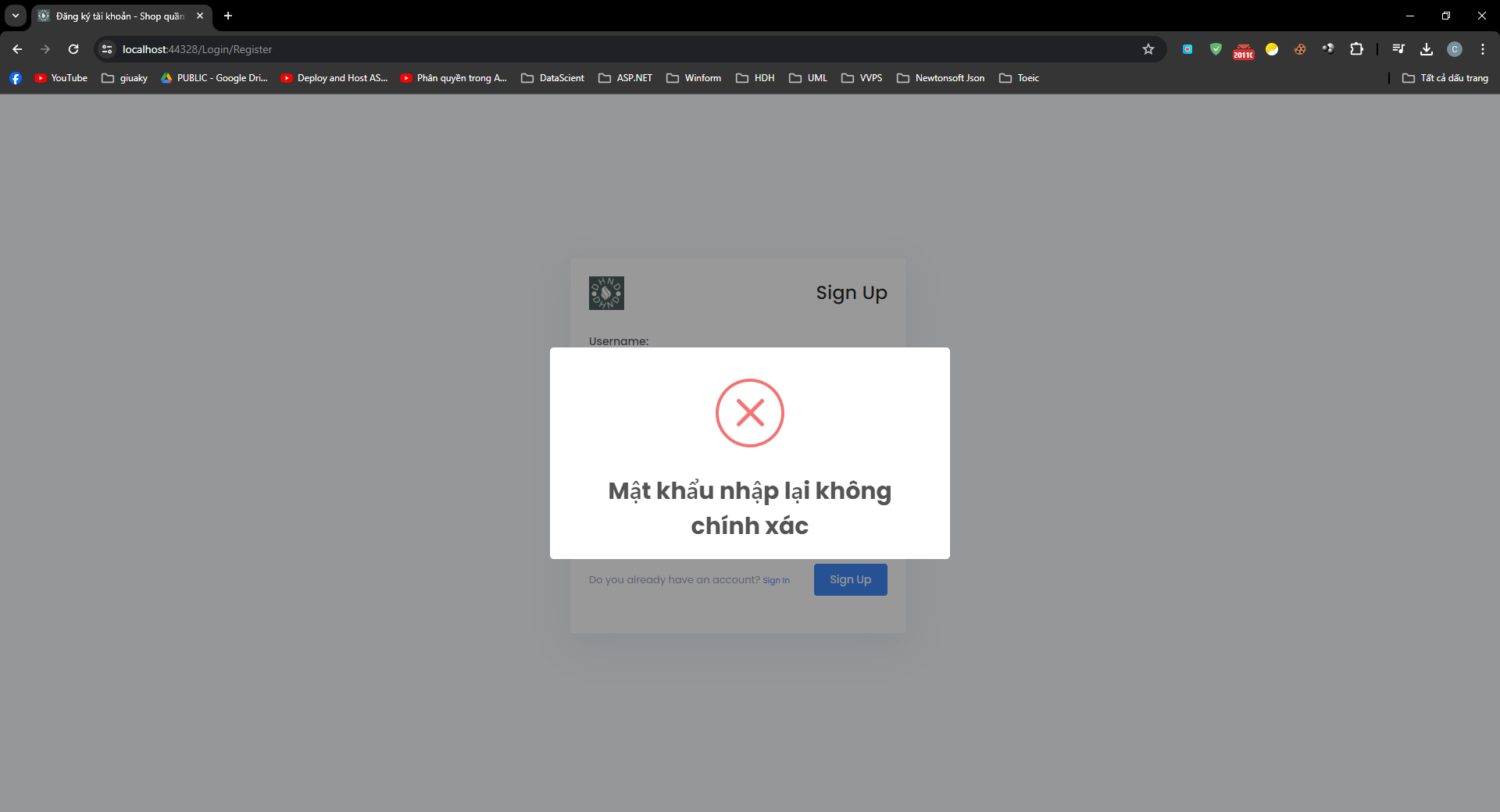
* + 1. Truy cập và check Validation form Registry







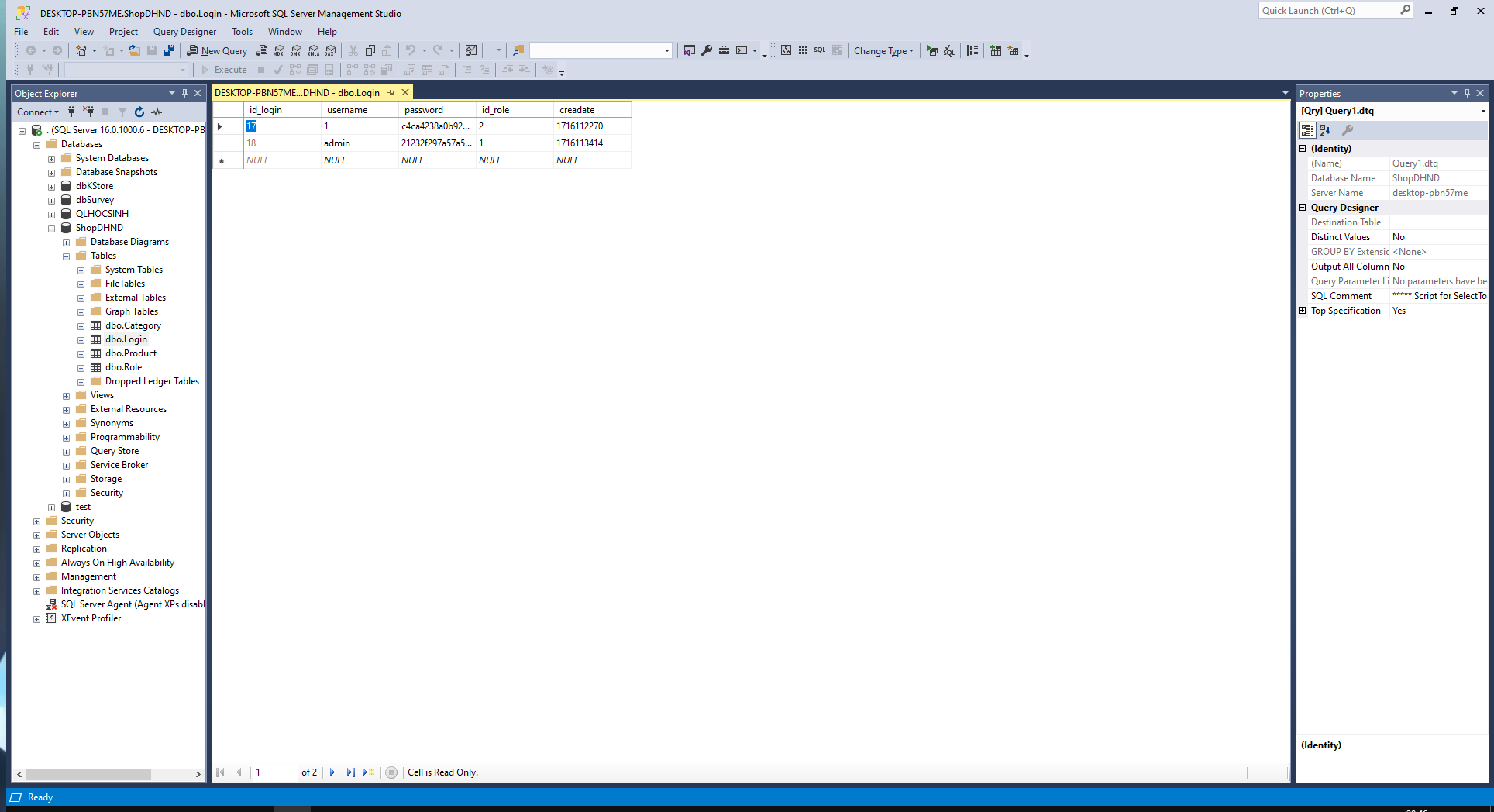


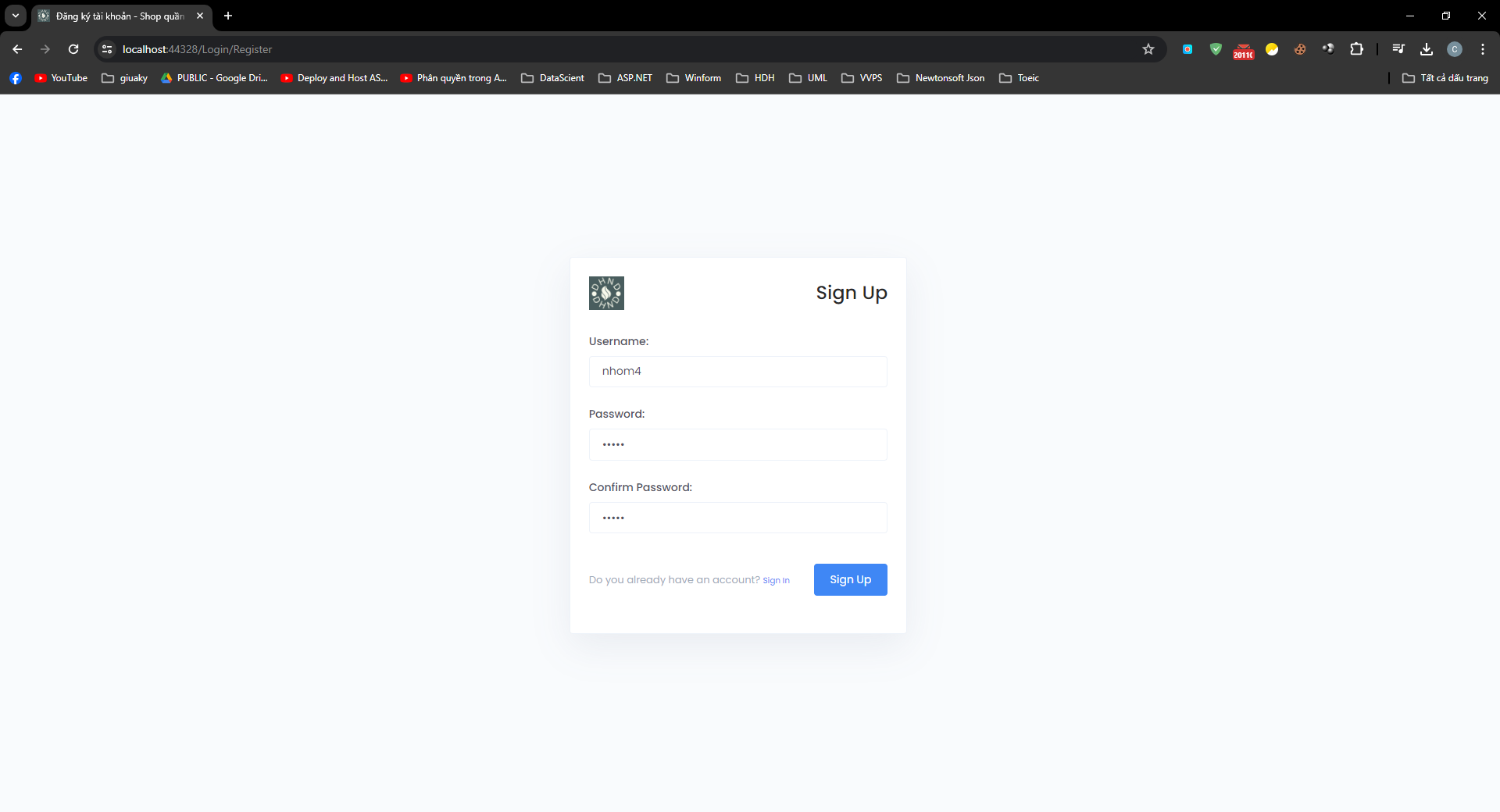


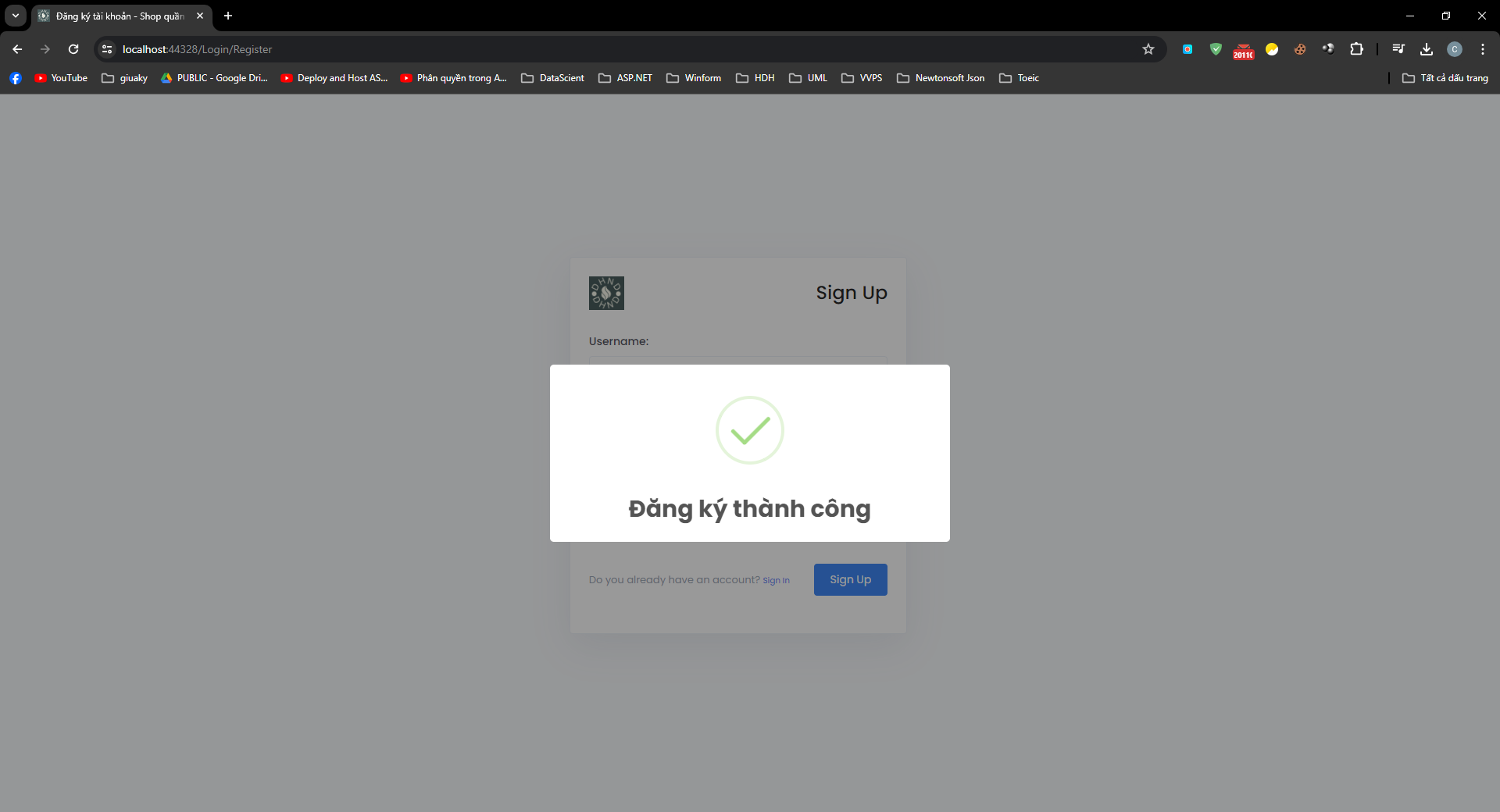
* + - 1. Sử dụng Postman để kiểm tra Request của website

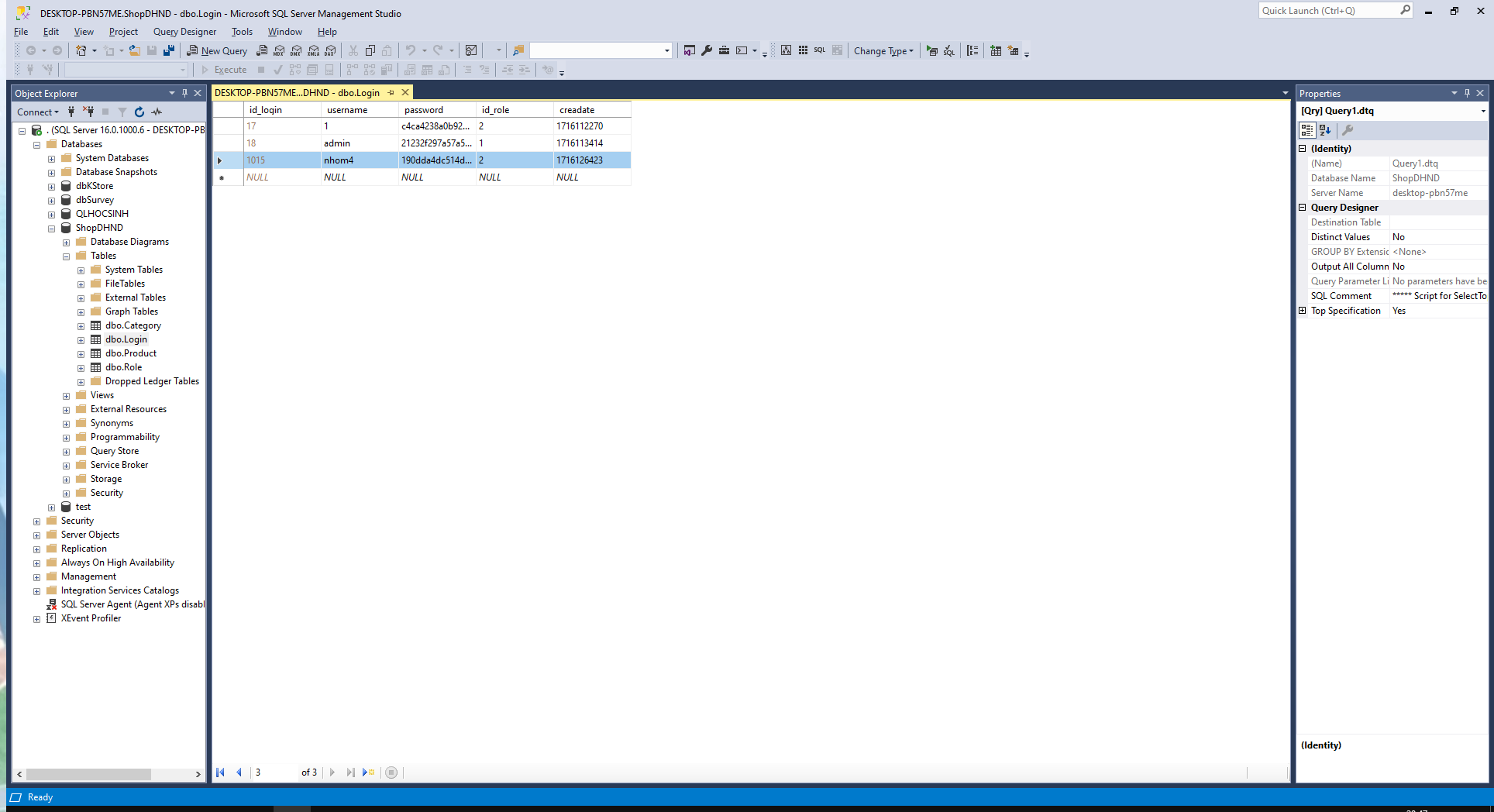
Qua đây chúng ta có thể thấy được khi gửi Request về cho website chúng ta có thể thấy được các yêu cầu trả về bên Server

* + 1. Mã hóa mật khẩu nhập vào khi tạo

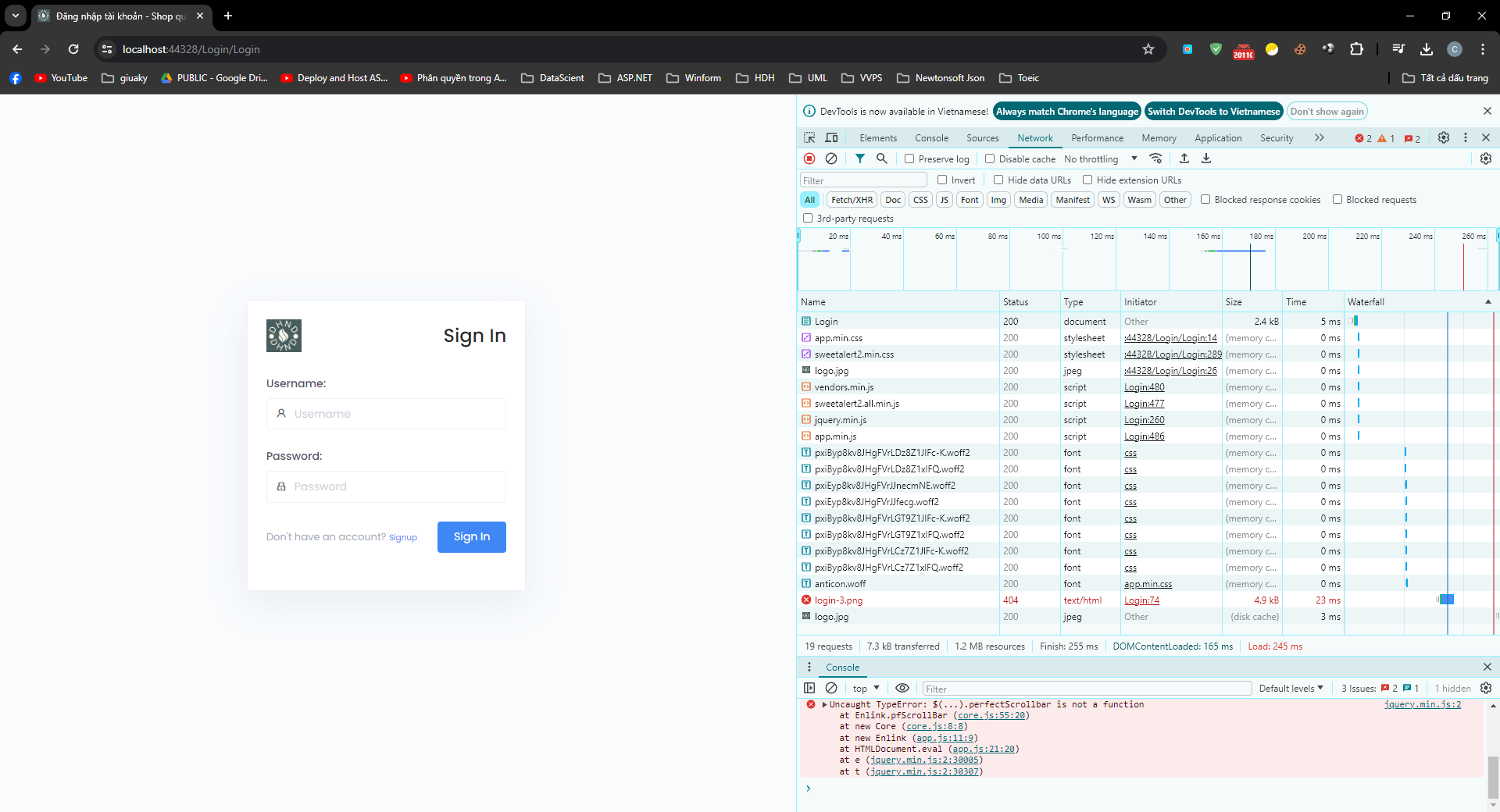


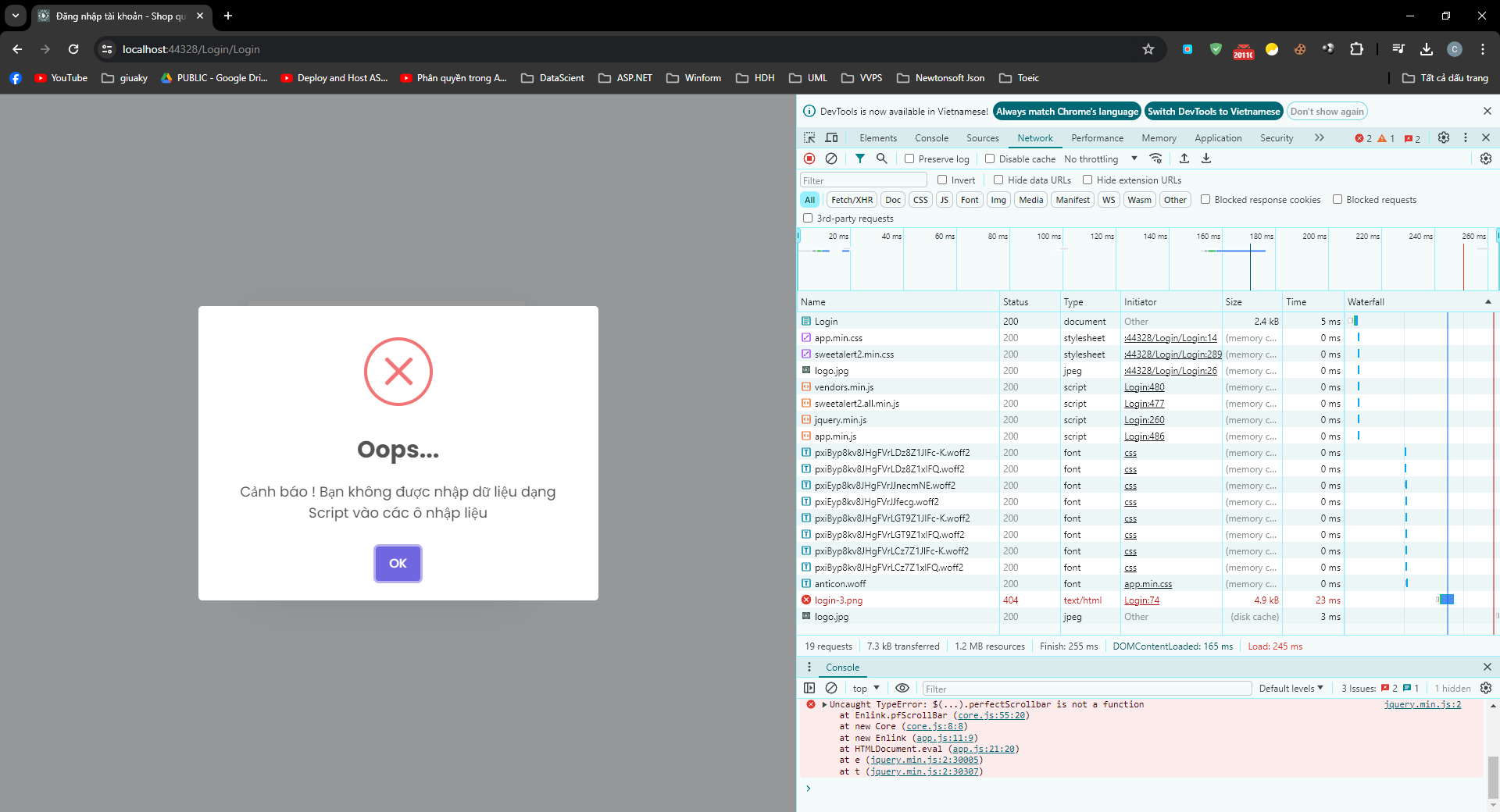




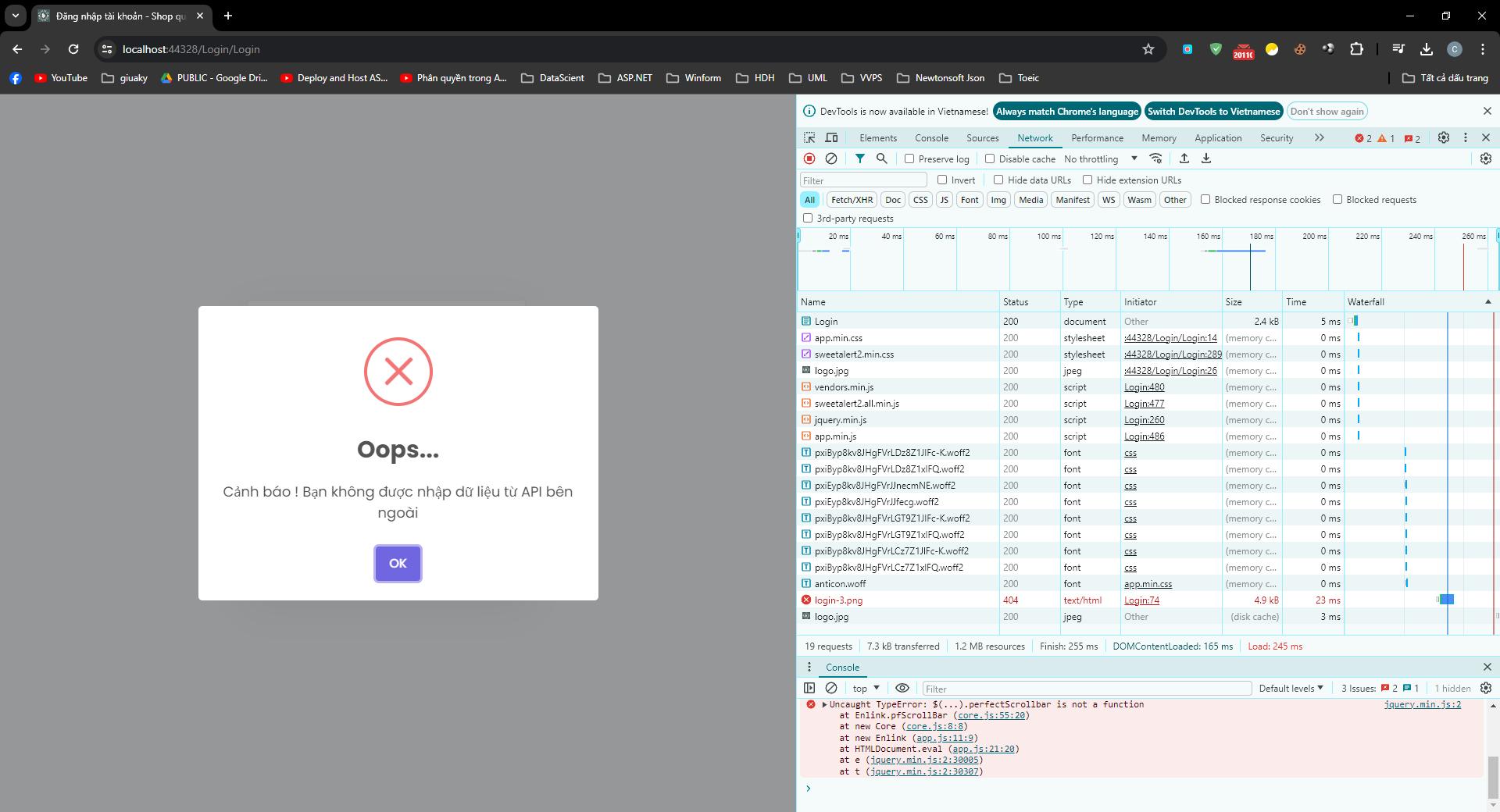


* + 1. Thực hiện việc bảo mật các lỗ hổng XSS của Website

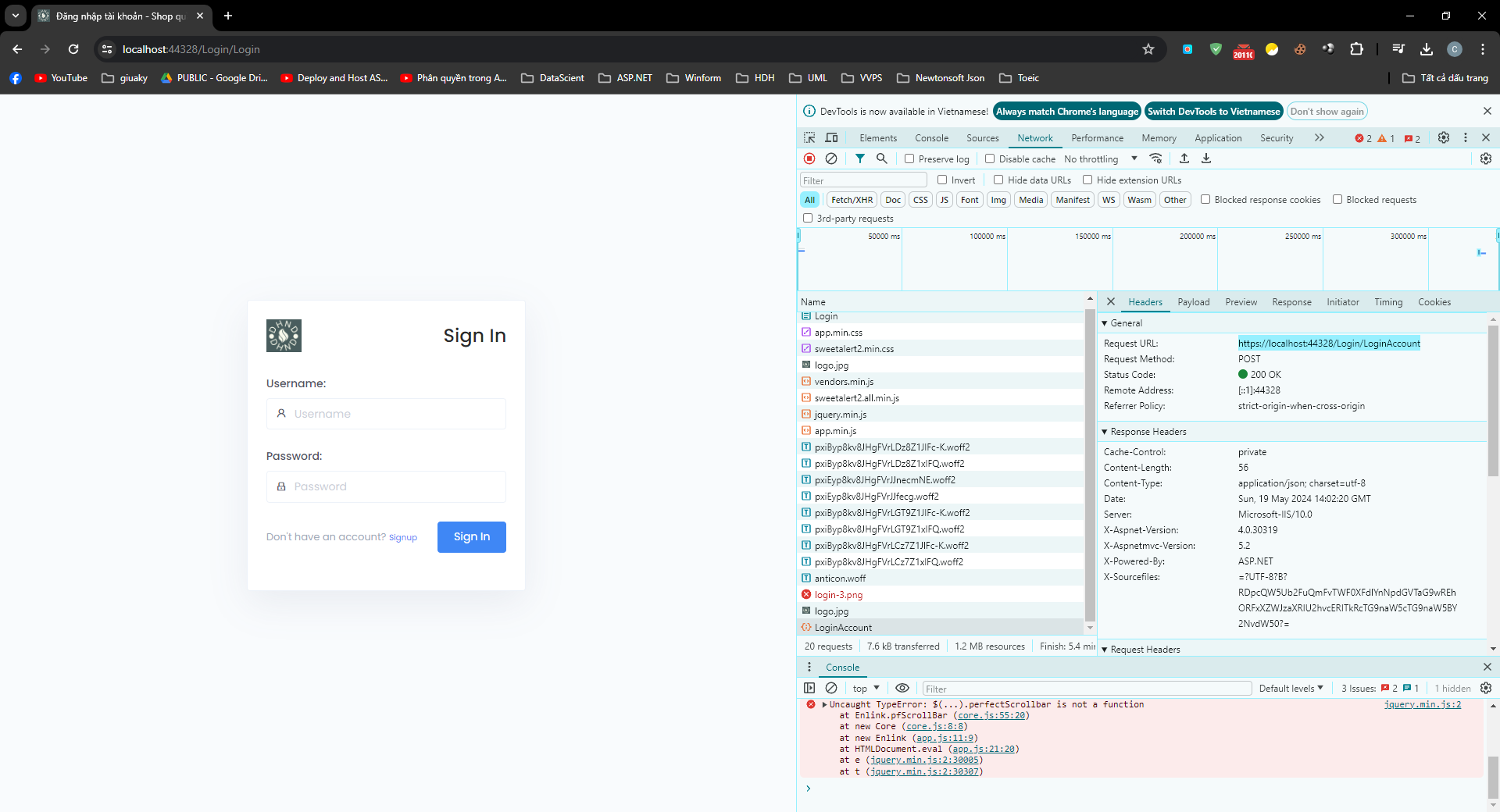


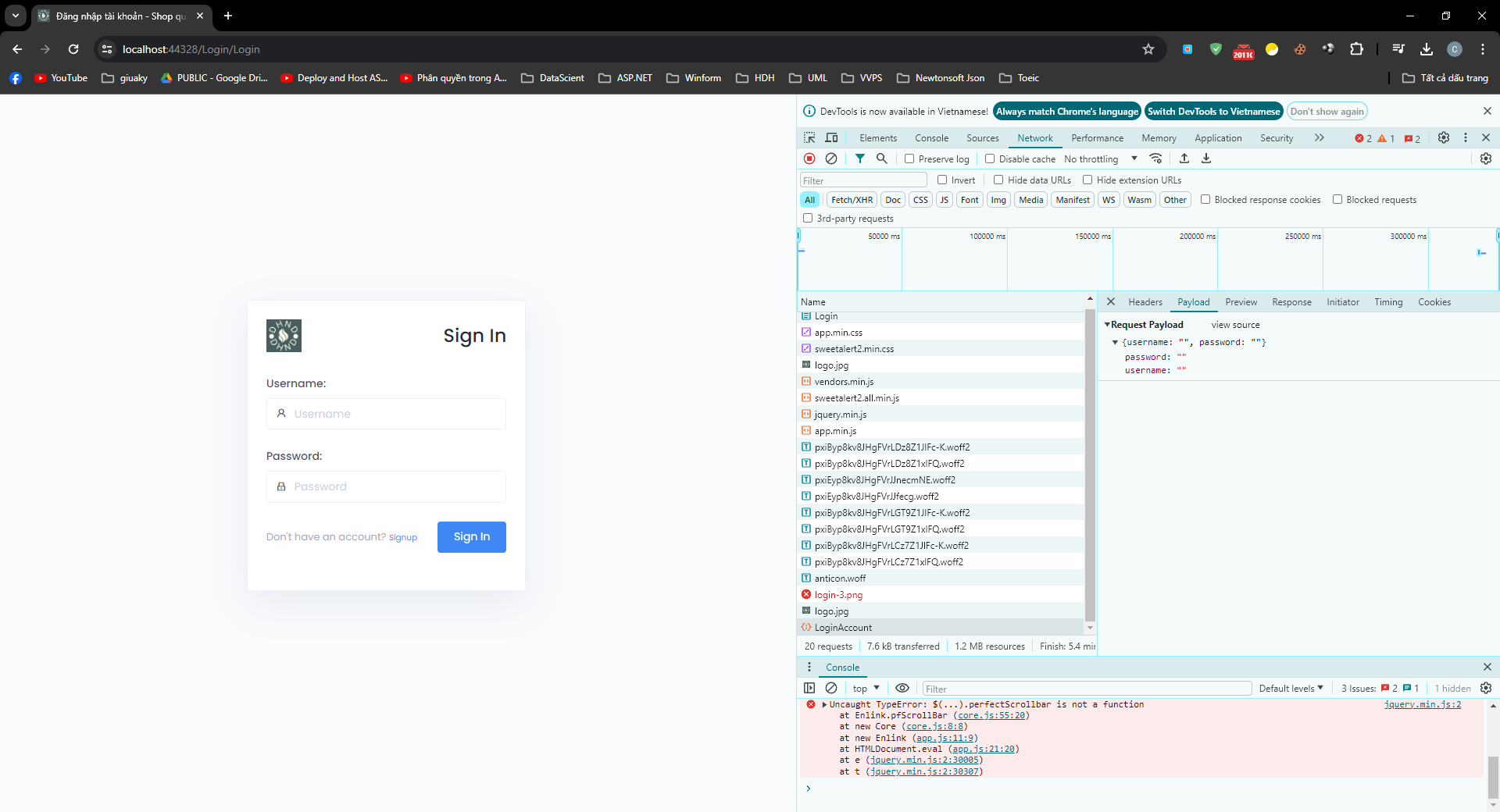


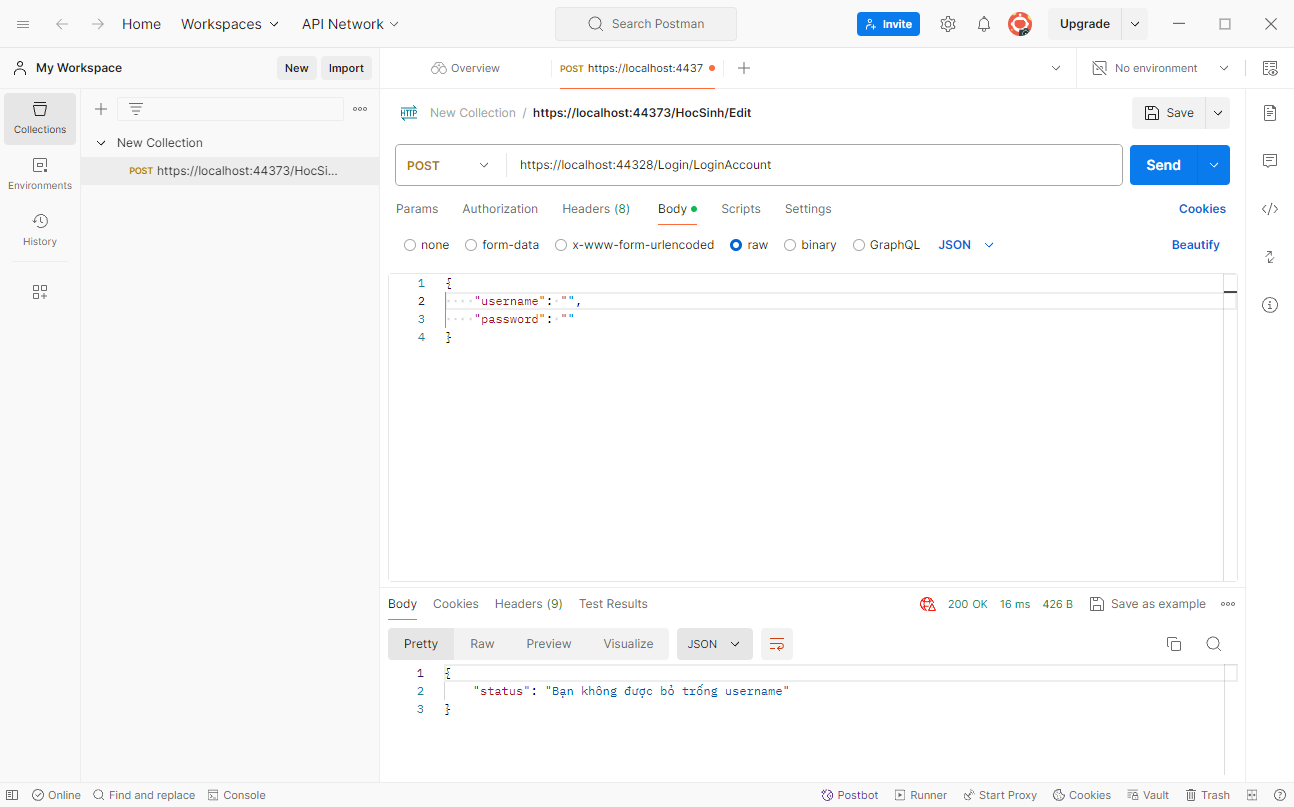
<script>alert("Tấn công - Nhóm 4")</script>

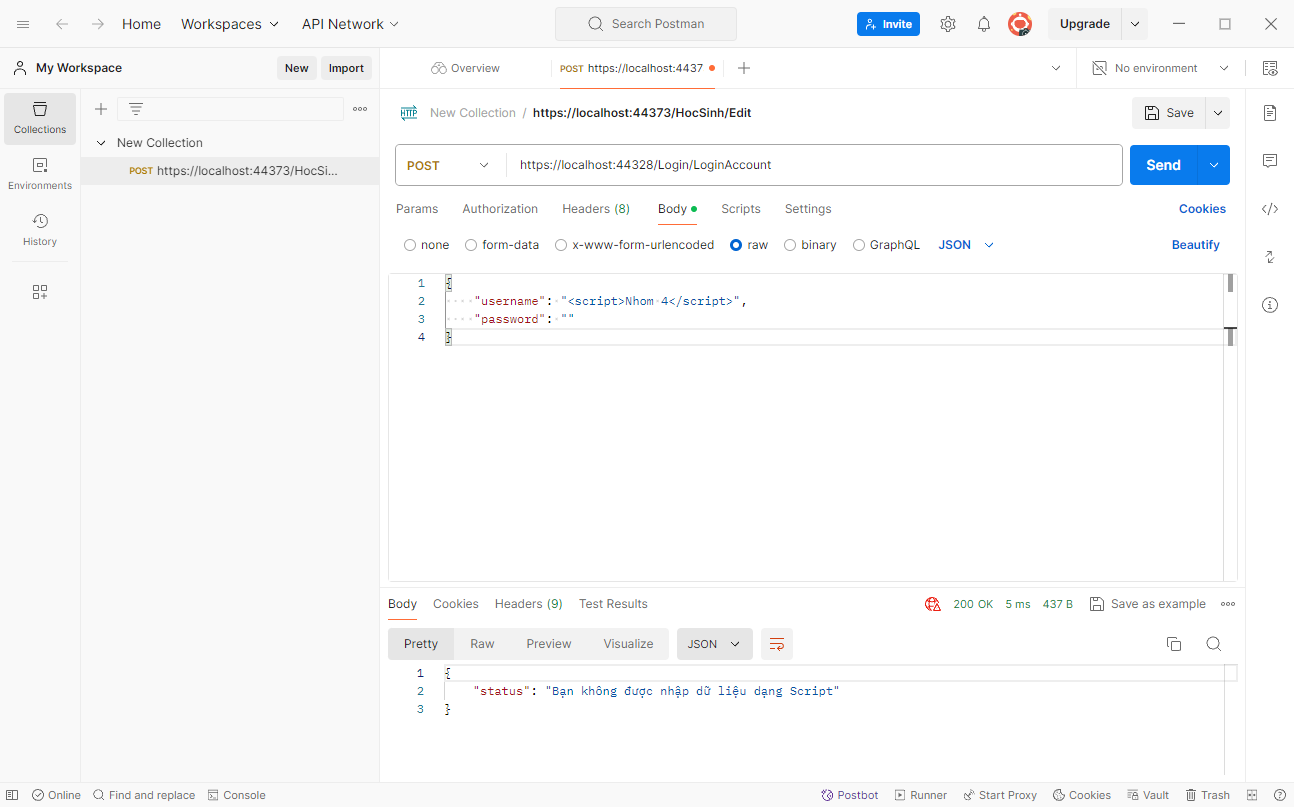


<https://cdn.jsdelivr.net/npm/sweetalert2@11.11.0/dist/sweetalert2.all.min.js>

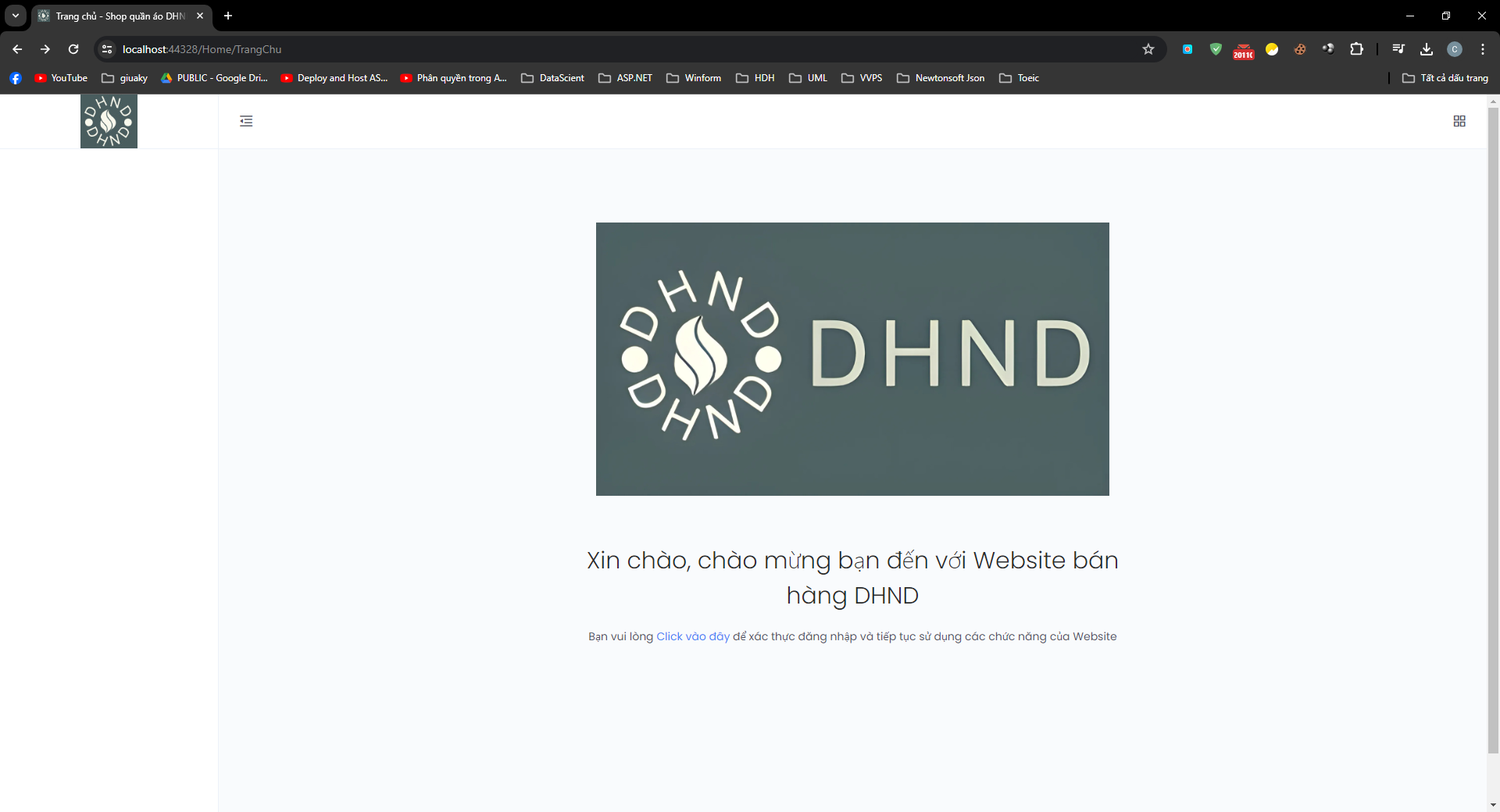


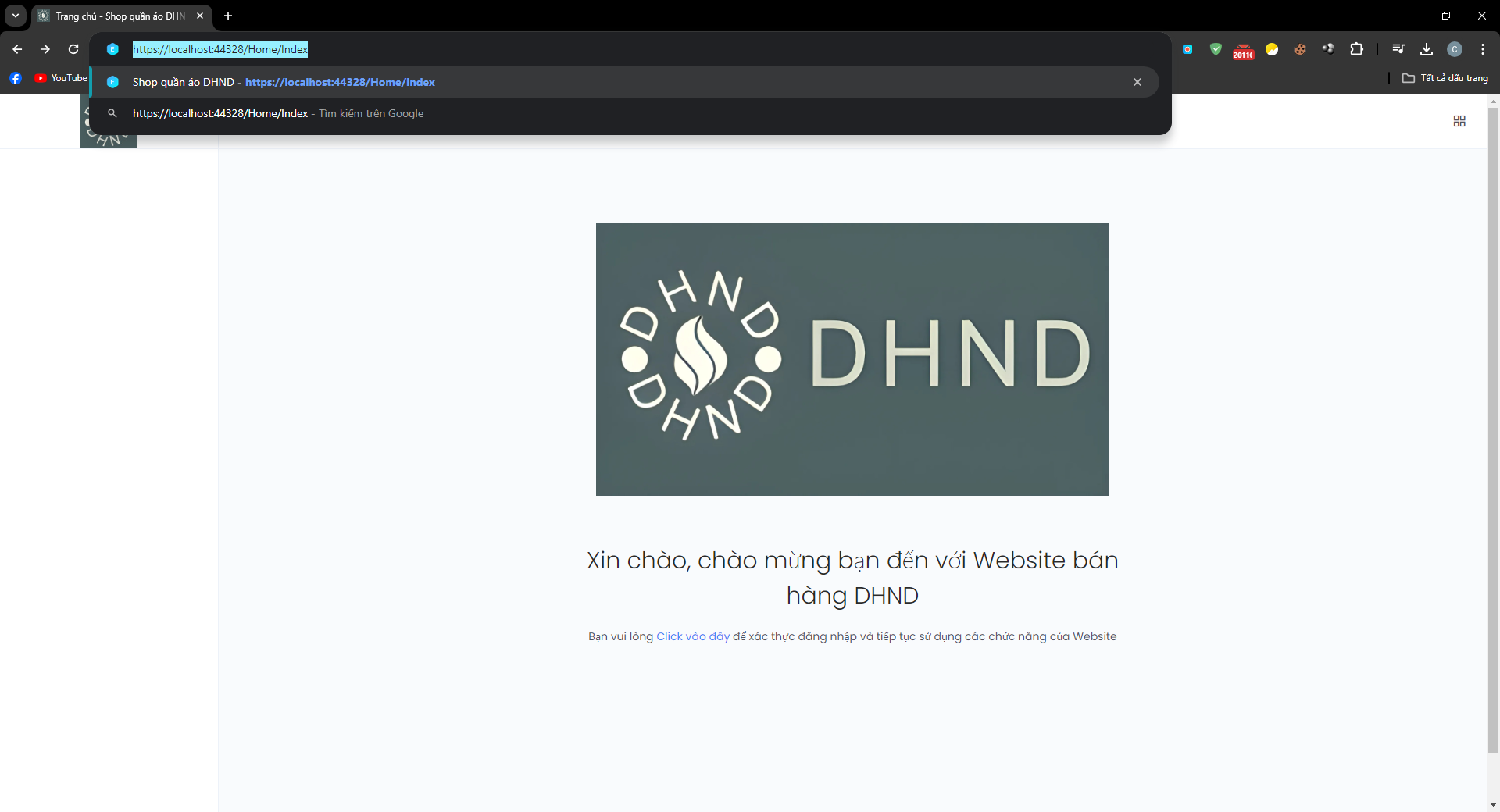




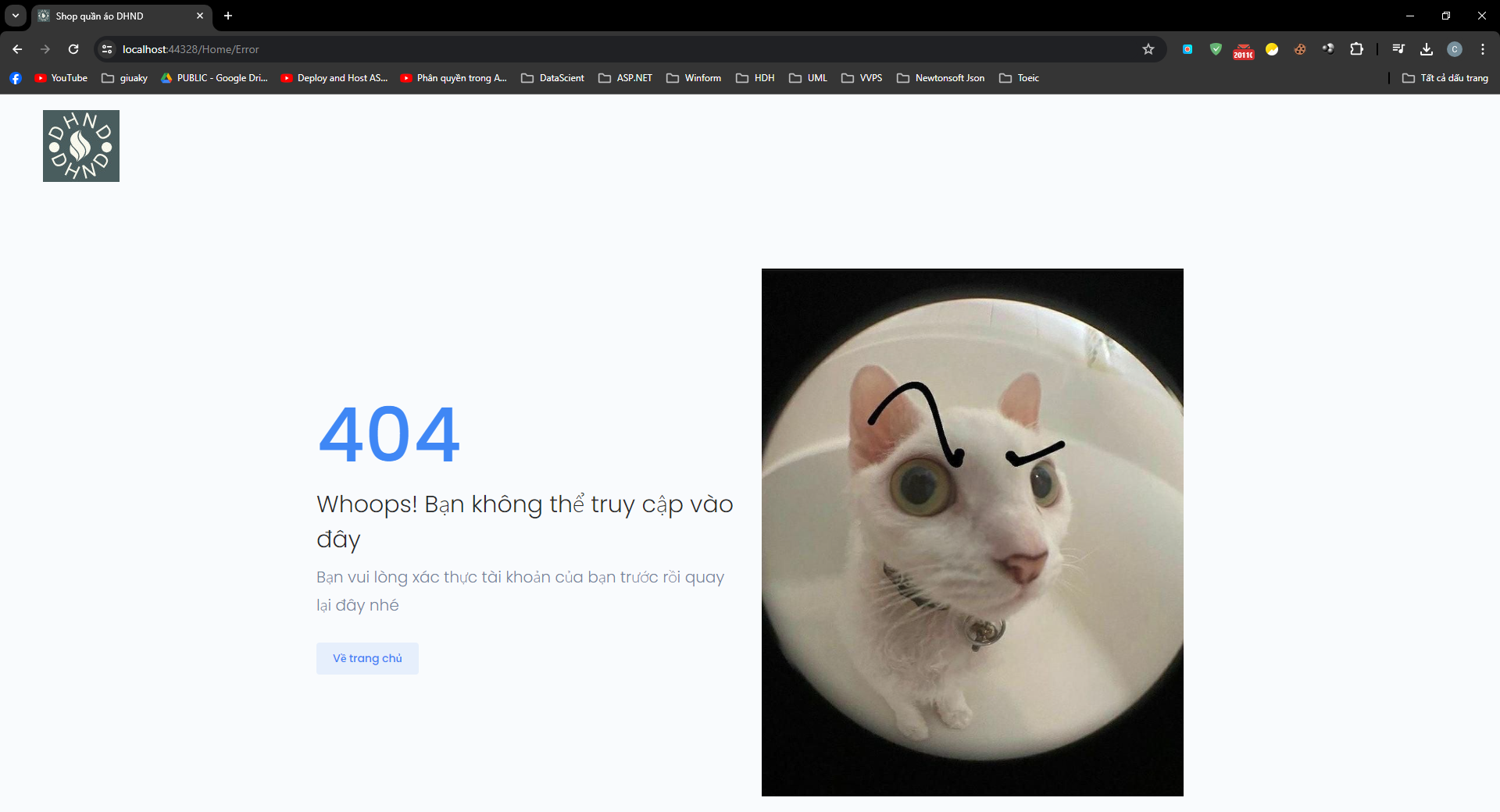


* + 1. Chống hành vi truy cập trái phép của người dùng vào trang quản trị

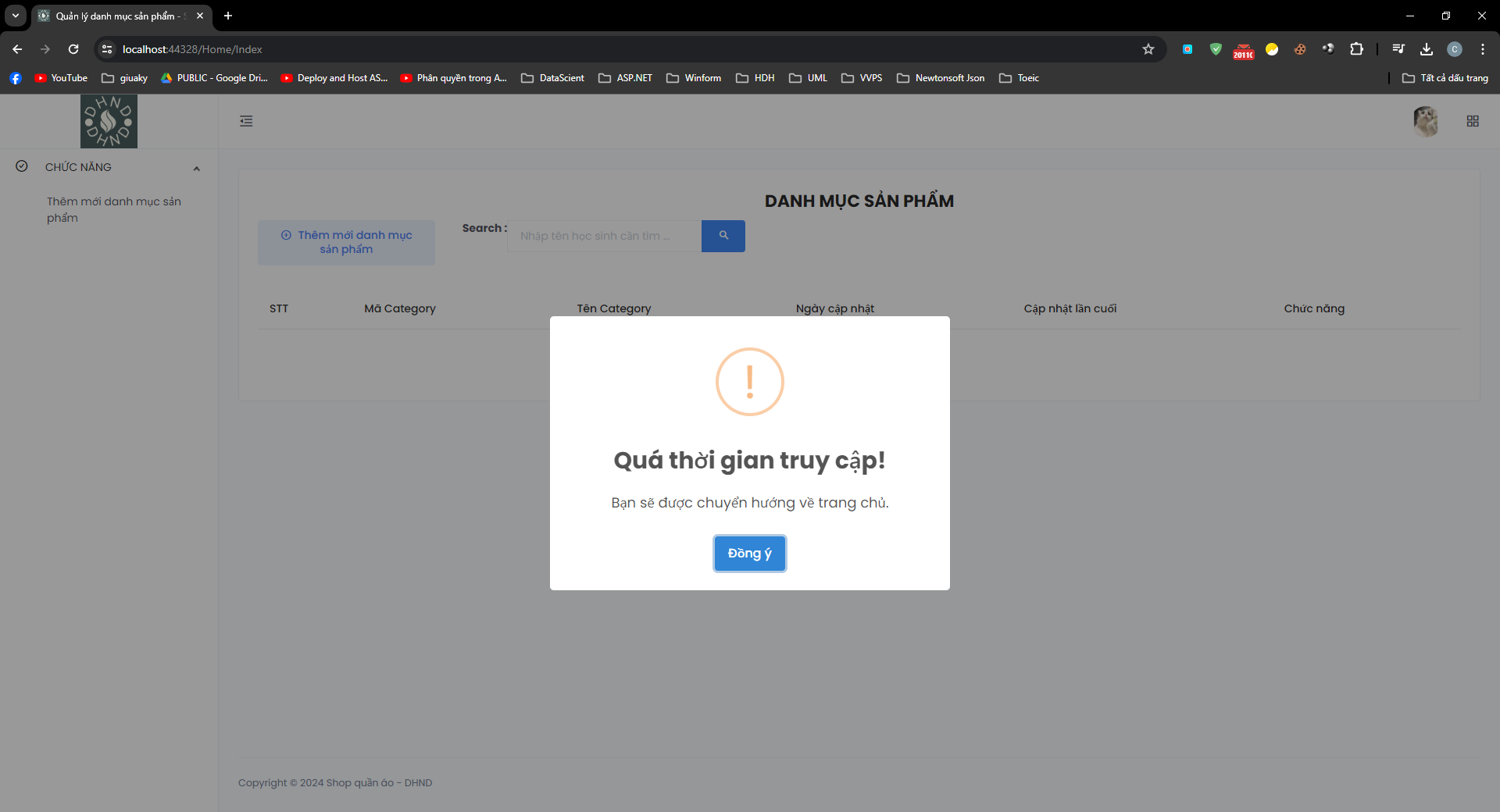


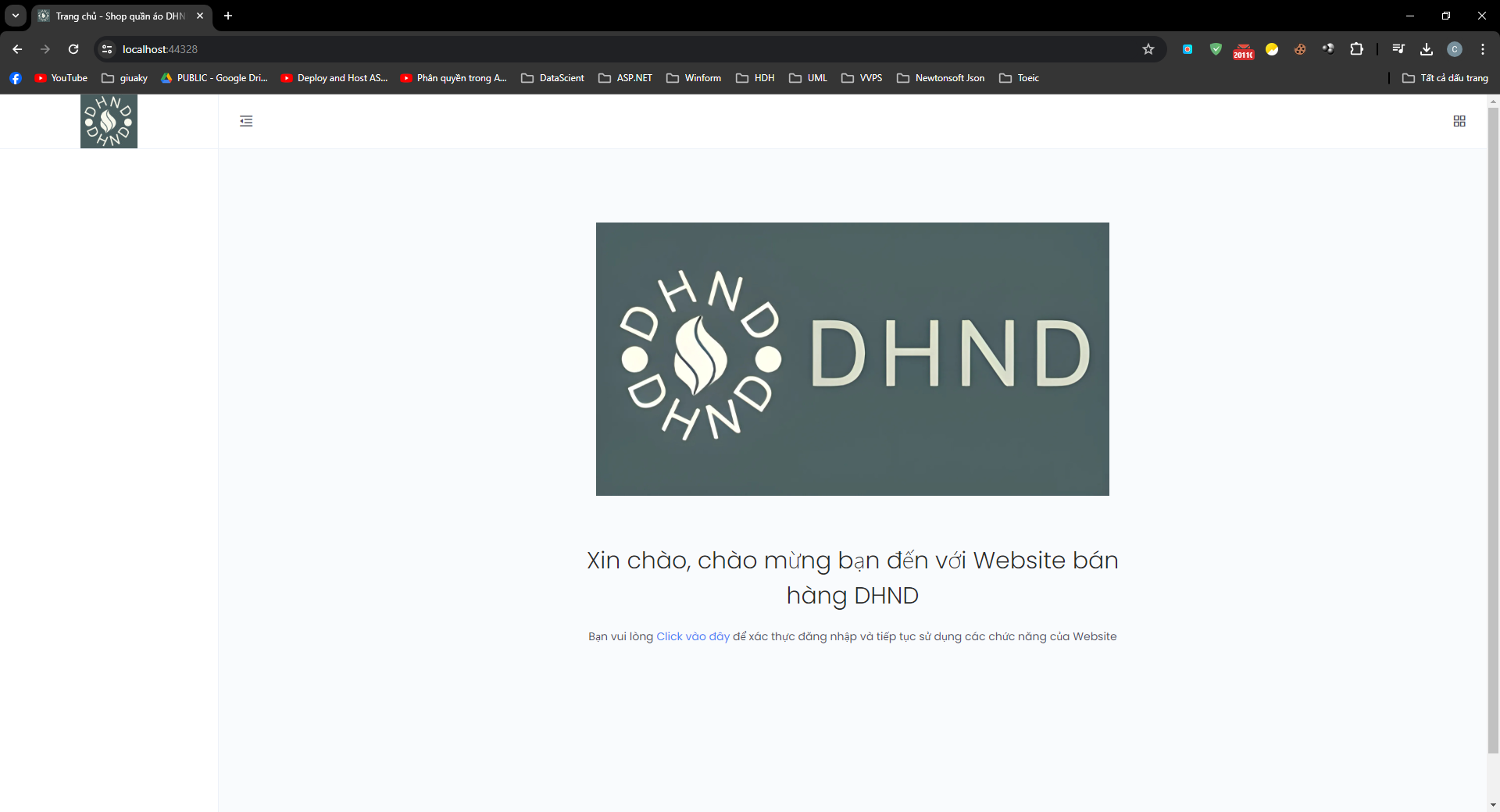


<https://localhost:44328/Home/Index>

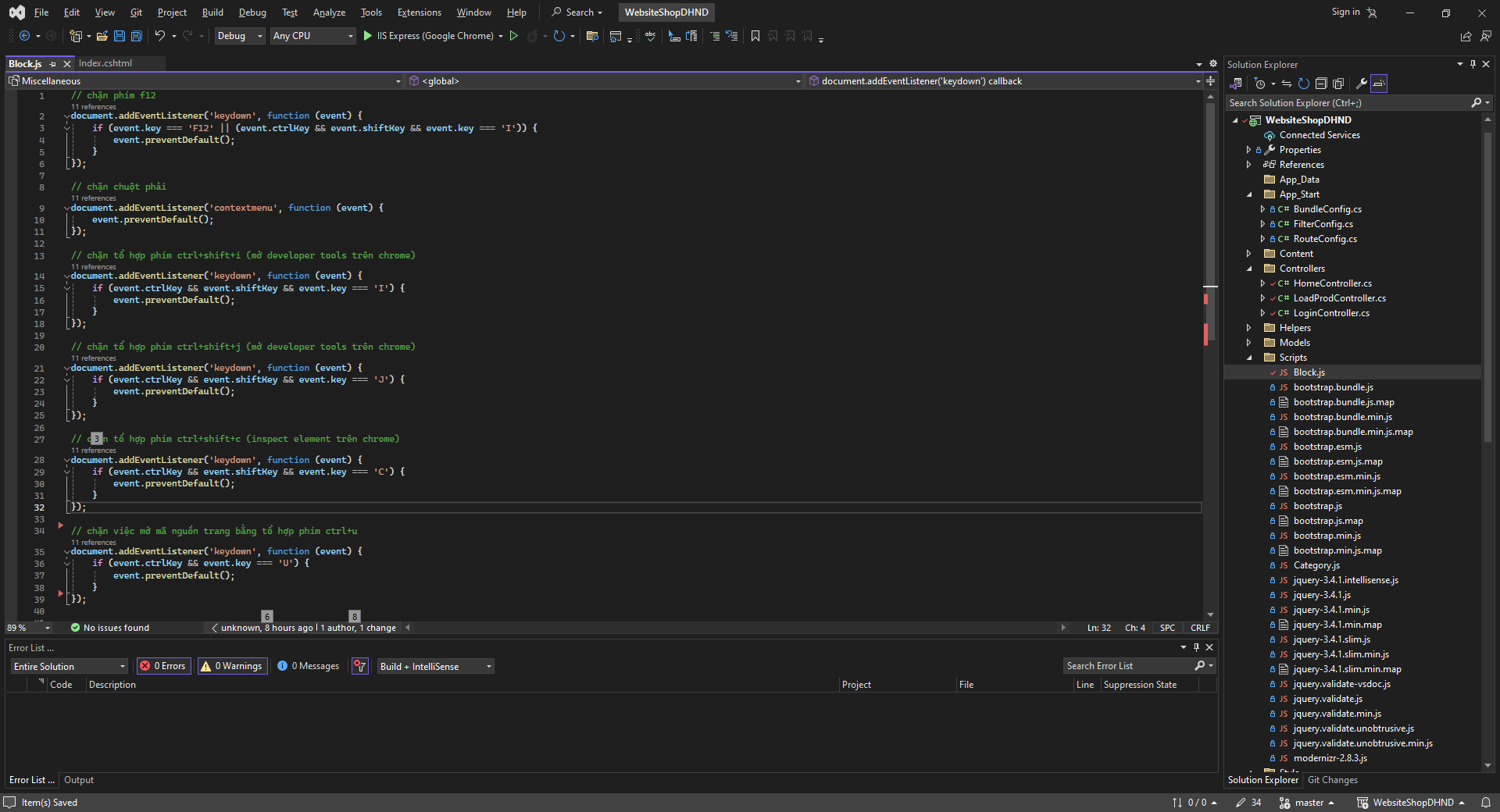


* + 1. Chống quá tải hệ thống truy cập





* + 1. Bảo vệ mã nguồn bằng cách chống người dùng truy cập vào DevTool



KẾT LUẬN

**1. Kết quả đạt được**

Trong quá trình nghiên cứu và thực hiện đề tài "Xây dựng giải pháp bảo mật cho website bán hàng DHND", chúng em đã đạt được những kết quả sau:

* Phân tích và hiểu rõ về các lỗ hổng bảo mật phổ biến trên website.
* Tìm hiểu về công cụ Postman và cách sử dụng để kiểm thử lỗ hổng trên website.
* Áp dụng các giải pháp bảo mật như kiểm tra Validation, ngăn chặn chèn mã độc dưới dạng scripts, ngăn chặn các hành vi đánh cắp mã nguồn, chống quá tải hệ thống truy cập, chống hành vi truy cập trái phép của người dùng vào trang quản trị, mã hóa mật khẩu người dùng.

**2. Hướng phát triển của đề tài**

Mặc dù đã đạt được một số kết quả tích cực, nhưng đề tài vẫn còn nhiều điểm có thể phát triển trong tương lai.

* Nghiên cứu và áp dụng các biện pháp bảo mật tiên tiến hơn như mã hóa quantum để ngăn chặn các cuộc tấn công từ máy tính cực kỳ mạnh mẽ.
* Phát triển các phương pháp kiểm thử tự động và kiểm thử dựa trên AI để phát hiện và loại bỏ các lỗ hổng bảo mật một cách hiệu quả và nhanh chóng.
* Mở rộng phạm vi nghiên cứu để bao gồm các loại lỗ hổng bảo mật khác như Insecure Direct Object References, XML External Entity Injection, và Server-Side Request Forgery.
* Tiến hành thử nghiệm thực tế trên một số website thực tế và đánh giá hiệu quả của giải pháp bảo mật được đề xuất.

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Dương Quốc Việt (2018), *An toàn dữ liệu: mã hóa bảo mật thông tin, an ninh cơ sở dữ liệu và an ninh mạng*, NXB Đại học Quốc gia Hà Nội. [Ngày truy cập: 11/05/2024]
2. Thái Thanh Tùng (2011), *Giáo trình Mật mã học và Hệ thống Thông tin an toàn*, NXB Thông tin và Truyền thông. [Ngày truy cập: 11/05/2024]
3. Nguyễn Quang Huy (2021), *Giáo trình Bảo mật hệ thống thông tin,* NXB Đại học Sư phạm TP.HCM. [Ngày truy cập: 12/05/2024]

**Tiếng Anh**

1. W. Stallings and L. Brown (2018), *Computer Security: Principles & Practice*, Pearson. [Ngày truy cập: 12/05/2024]
2. Jason Andress (2014)*, The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice,* Syngress. [Ngày truy cập: 13/05/2024]
3. Alan G. Konheim (2007), *Computer Security and Cryptography*, Wiley. [Ngày truy cập: 12/05/2024]

**Website**

1. <https://ncsgroup.vn/tong-ket-an-ninh-mang-viet-nam-nam-2023-va-du-bao-2024>, *Tổng kết An ninh mạng Việt Nam năm 2023 và dự báo 2024.* [Truy cập ngày 12/05/2024]
2. <https://viblo.asia/p/postman-cong-cu-danh-cho-nguoi-moi-bat-dau-kiem-thu-api-gAm5ya0w5db>, *Postman - Công cụ dành cho người mới bắt đầu kiểm thử API.*  [Truy cập ngày 13/05/2024]
3. <https://viblo.asia/p/md5-sha-va-sm3-la-cac-thuat-toan-ma-hoa-mot-chieu-pho-bien-EoW4omQxLml>, *MD5, SHA và SM3 là các thuật toán mã hóa một chiều phổ biến.* [Truy cập ngày 13/05/2024]
4. <https://viblo.asia/p/cac-cach-validate-form-trong-thiet-ke-website-uu-nhuoc-diem-Az45bPQOZxY>, *Các cách validate form trong thiết kế website - Ưu nhược điểm.* [Truy cập ngày 13/05/2024]