**ỦY BAN NHÂN DÂN TP.HCM**

**TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC**



**BÁO CÁO CHUYÊN ĐỀ WEB 1**

**NĂM HỌC 2020 – 2021**

**GV : Phan Thanh Nhuần**

**Thành viên nhóm F:**

* Phạm Văn Hiệu ( Nhóm Trưởng )
* Huỳnh Đại Long ( Nhóm Phó )
* Nguyễn Đức Duy
* Lê Văn Tuyển
* Đinh Thành Đức

**BẢNG PHÂN CÔNG CÔNG VIỆC NHÓM F**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Tìm hiểu các kỹ thuật tấn công website | Ngày: 12/10/2020 Giờ:13h | Ngày:12/10/2020 Giờ:16h45 | 80% |
| 2 | Huỳnh Đại Long | Tìm hiểu website chia theo module | Ngày: 12/10/2020 Giờ:13h | Ngày:12/10/2020 Giờ:16h45 | 70% |
| 3 | Nguyễn Đức Duy | Tìm hiểu less sass | Ngày: 12/10/2020 Giờ:13h | Ngày:12/10/2020 Giờ:16h45 | 70% |
| 4 | Đinh Thành Đức | Tìm hiểu scrum và rup | Ngày: 12/10/2020 Giờ:13h | Ngày:12/10/2020 Giờ:16h45 | 60% |
| 5 | Lê Văn Tuyển | Tìm hiểu session và cookie | Ngày: 12/10/2020 Giờ:13h | Ngày:12/10/2020 Giờ:16h45 | 50% |

Bảng 1: Công việc ngày 12/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Tìm lỗi frontend và backend source trainning-php trang login.php và register.php | Ngày: 13/10/2020 Giờ:13h | Ngày:13/10/2020 Giờ:16h45 | 70% |
| 2 | Huỳnh Đại Long | Tìm lỗi frontend và backend source trainning-php trang admin.php | Ngày: 13/10/2020 Giờ:13h | Ngày:13/10/2020 Giờ:16h45 | 70% |
| 3 | Nguyễn Đức Duy | Tìm lỗi frontend và backend source trainning-php trang list.php | Ngày: 13/10/2020 Giờ:13h | Ngày:13/10/2020 Giờ:16h45 | 70% |
| 4 | Đinh Thành Đức | Tìm lỗi frontend và backend source trainning-php trang edit.php | Ngày: 13/10/2020 Giờ:13h | Ngày:13/10/2020 Giờ:16h45 | 70% |
| 5 | Lê Văn Tuyển | Tìm lỗi frontend và backend source trainning-php trang home.php | Ngày: 13/10/2020 Giờ:13h | Ngày:13/10/2020 Giờ:16h45 | 70% |

Bảng 2: Công việc ngày 13/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Tiếp tục tìm lỗi frontend và backend source trainning-php trang login.php và register.php | Ngày: 14/10/2020 Giờ:13h | Ngày:14/10/2020 Giờ:16h45 | 100% |
| 2 | Huỳnh Đại Long | Tiếp tục tìm lỗi frontend và backend source trainning-php trang admin.php | Ngày: 14/10/2020 Giờ:13h | Ngày:14/10/2020 Giờ:16h45 | 100% |
| 3 | Nguyễn Đức Duy | Tiếp tục tìm lỗi frontend và backend source trainning-php trang list.php | Ngày: 14/10/2020 Giờ:13h | Ngày:14/10/2020 Giờ:16h45 | 100% |
| 4 | Đinh Thành Đức | Tiếp tục tìm lỗi frontend và backend source trainning-php trang edit.php | Ngày: 14/10/2020 Giờ:13h | Ngày:14/10/2020 Giờ:16h45 | 100% |
| 5 | Lê Văn Tuyển | Tìếp tục tìm lỗi frontend và backend source trainning-php trang home.php | Ngày: 14/10/2020 Giờ:13h | Ngày:14/10/2020 Giờ:16h45 | 100% |

Bảng 3: Công việc ngày 14/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Tìm hiểu tấn công xss | Ngày: 19/10/2020 Giờ:13h | Ngày:19/10/2020 Giờ:16h45 | 50% |
| 2 | Huỳnh Đại Long | Tìm hiểu tấn công Command execution | Ngày: 19/10/2020 Giờ:13h | Ngày:19/10/2020 Giờ:16h45 | 50% |
| 3 | Nguyễn Đức Duy | Tìm hiểu tấn công user enumeration | Ngày: 19/10/2020 Giờ:13h | Ngày:19/10/2020 Giờ:16h45 | 50% |
| 4 | Đinh Thành Đức | Tìm hiểu tấn công csrf | Ngày: 19/10/2020 Giờ:13h | Ngày:19/10/2020 Giờ:16h45 | 50% |
| 5 | Lê Văn Tuyển | Tìm hiểu tấn công Sql injection | Ngày: 19/10/2020 Giờ:13h | Ngày:19/10/2020 Giờ:16h45 | 50% |

Bảng 4: Công việc ngày 19/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Làm chức năng phân quyền | Ngày: 20/10/2020 Giờ:13h | Ngày:20/10/2020 Giờ:16h45 | 90% |
| 2 | Huỳnh Đại Long | Làm chức năng đổi mật khẩu | Ngày: 20/10/2020 Giờ:13h | Ngày:20/10/2020 Giờ:16h45 | 85% |
| 3 | Nguyễn Đức Duy | Làm chức năng logout | Ngày: 20/10/2020 Giờ:13h | Ngày:20/10/2020 Giờ:16h45 | 85% |
| 4 | Đinh Thành Đức | Làm chức năng phân trang | Ngày: 20/10/2020 Giờ:13h | Ngày:20/10/2020 Giờ:16h45 | 75% |
| 5 | Lê Văn Tuyển | Làm chức năng login | Ngày: 20/10/2020 Giờ:13h | Ngày:20/10/2020 Giờ:16h45 | 80% |

Bảng 5: Công việc ngày 20/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Làm chức năng sửa | Ngày: 21/10/2020 Giờ:13h | Ngày:21/10/2020 Giờ:16h45 | 90% |
| 2 | Huỳnh Đại Long | Làm chức năng xóa | Ngày: 21/10/2020 Giờ:13h | Ngày:21/10/2020 Giờ:16h45 | 90% |
| 3 | Nguyễn Đức Duy | Làm chức năng remember | Ngày: 21/10/2020 Giờ:13h | Ngày:21/10/2020 Giờ:16h45 | 90% |
| 4 | Đinh Thành Đức | Làm chức năng phân trang | Ngày: 21/10/2020 Giờ:13h | Chưa hoàn thành | 50% |
| 5 | Lê Văn Tuyển | Làm chức năng thêm | Ngày: 21/10/2020 Giờ:13h | Chưa hoàn thành | 50% |

Bảng 6: Công việc ngày 21/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Làm chức năng upload file hình ảnh | Ngày: 26/10/2020 Giờ:13h | Ngày:26/10/2020 Giờ:16h45 | 100% |
| 2 | Huỳnh Đại Long | Làm chức năng tìm kiếm | Ngày: 26/10/2020 Giờ:13h | Ngày:26/10/2020 Giờ:16h45 | 100% |
| 3 | Nguyễn Đức Duy | Làm chức năng hiển thị | Ngày: 26/10/2020 Giờ:13h | Ngày:26/10/2020 Giờ:16h45 | 100% |
| 4 | Đinh Thành Đức | Tiếp tục làm chức năng phân trang | Ngày: 26/10/2020 Giờ:13h | Ngày:26/10/2020 Giờ:16h45 | 100% |
| 5 | Lê Văn Tuyển | Tiếp tục làm chức năng thêm | Ngày: 26/10/2020 Giờ:13h | Ngày:26/10/2020 Giờ:16h45 | 100% |

Bảng 7: Công việc ngày 26/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Tiếp tục cải thiện lại tấc cả chức năng sửa,upload file hình ảnh,phân quyền | Ngày: 27/10/2020 Giờ:13h | Ngày:27/10/2020 Giờ:16h45 | 100% |
| 2 | Huỳnh Đại Long | Tiếp tục cải thiện lại tấc cả chức năng tìm kiếm,xóa,đổi mật khẩu | Ngày: 27/10/2020 Giờ:13h | Ngày:27/10/2020 Giờ:16h45 | 100% |
| 3 | Nguyễn Đức Duy | Tiếp tục cải thiện lại tấc cả chức năng logout,remember,hiển thị | Ngày: 27/10/2020 Giờ:13h | Ngày:27/10/2020 Giờ:16h45 | 100% |
| 4 | Đinh Thành Đức | Tiếp tục cải thiện lại tấc cả chức năng phân trang,sắp xếp | Ngày: 27/10/2020 Giờ:13h | Ngày:27/10/2020 Giờ:16h45 | 100% |
| 5 | Lê Văn Tuyển | Tiếp tục cải thiện lại tấc cả chức năng thêm,login | Ngày: 27/10/2020 Giờ:13h | Ngày:27/10/2020 Giờ:16h45 | 100% |

Bảng 8: Công việc ngày 27/10/2020

|  |  |  |
| --- | --- | --- |
| **STT** | **Công Việc Chung Của Nhóm** | **Hoàn Thành** |
| 1 | Tìm hiểu SmartGit | 50% |
| 2 | Tìm hiểu và khắc phục lỗi tấn công từ id của user | 70% |

Bảng 9: Công việc chung của nhóm ngày 27/10/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Tổng hợp chức năng sửa,upload file hình ảnh,phân quyền và các chức năng của thành viên trong nhóm gọp vào project hoàn chỉnh. Tìm hiểu tấn công Sensitive data exposure. | Ngày: 02/11/2020 Giờ:13h | Ngày:02/11/2020 Giờ:16h45 | 100% |
| 2 | Huỳnh Đại Long | Tổng hợp chức năng tìm kiếm,xóa,đổi mật khẩu nộp cho nhóm trưởng. Tìm hiểu tấn công Insecure Direct Object References. | Ngày: 02/11/2020 Giờ:13h | Ngày:02/11/2020 Giờ:16h45 | 100% |
| 3 | Nguyễn Đức Duy | Tổng hợp chức năng logout,remember,hiển thị nộp cho nhóm trưởng. Tìm hiểu tấn công Security misconfiguration. | Ngày: 02/11/2020 Giờ:13h | Ngày:02/11/2020 Giờ:16h45 | 100% |
| 4 | Đinh Thành Đức | Tổng hợp chức năng phân trang,sắp xếp nộp cho nhóm trưởng. Tìm hiểu tấn công Missing function level access control. | Ngày: 02/11/2020 Giờ:13h | Ngày:02/11/2020 Giờ:16h45 | 100% |
| 5 | Lê Văn Tuyển | Tổng hợp chức năng thêm,login nộp cho nhóm trưởng. Tìm hiểu tấn công Broken Authentication. | Ngày: 02/11/2020 Giờ:13h | Ngày:02/11/2020 Giờ:16h45 | 100% |

Bảng 10: Công việc ngày 02/11/2020

|  |  |  |
| --- | --- | --- |
| **STT** | **Công Việc Chung Của Nhóm** | **Hoàn Thành** |
| 1 | Tìm hiểu SmartGit và tấc cả thành viên trong nhóm đều biết up và lấy source | 80% |
| 2 | Nộp file báo cáo topic 6 cho nhóm trưởng | 100% |

Bảng 11: Công việc chung của nhóm ngày 02/11/2020

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và Tên** | **Phân Công Công Việc** | **Bắt Đầu** | **Kết Thúc** | **Hoàn Thành** |
| 1 | Phạm Văn Hiệu | Đưa source nhóm lên github bằng smartgit, demo code bằng word chức năng sửa,upload file hình ảnh,phân quyền | Ngày: 03/11/2020 Giờ:13h | Ngày:03/11/2020 Giờ:16h45 | 90% |
| 2 | Huỳnh Đại Long | Demo code bằng word chức năng tìm kiếm,xóa,đổi mật khẩu | Ngày: 03/11/2020 Giờ:13h | Ngày:03/11/2020 Giờ:16h45 | 90% |
| 3 | Nguyễn Đức Duy | Demo code bằng word chức năng logout,remember,hiển thị | Ngày: 03/11/2020 Giờ:13h | Ngày:03/11/2020 Giờ:16h45 | 90% |
| 4 | Đinh Thành Đức | Demo code bằng word chức năng phân trang,sắp xếp | Ngày: 03/11/2020 Giờ:13h | Ngày:03/11/2020 Giờ:16h45 | 90% |
| 5 | Lê Văn Tuyển | Demo code bằng word chức năng thêm,login | Ngày: 03/11/2020 Giờ:13h | Ngày:03/11/2020 Giờ:16h45 | 90% |

Bảng 12: Công việc ngày 03/11/2020

|  |  |  |
| --- | --- | --- |
| **STT** | **Công Việc Chung Của Nhóm** | **Hoàn Thành** |
| 1 | - Chụp hinh màu in ra màu đen (nhận) hoặc đổi background rồi in  - Hình: có số và mô tả về hình  - Hình chụp đúng nội dung  - Không chụp hình tự vẽ bằng chuột.  - Font định dạng thống nhất nhau, - Kiểm tra lại lỗi chính tả. | 90% |
| 2 | Nộp file báo cáo topic 6 cho nhóm trưởng | 90% |

Bảng 13: Công việc chung của nhóm ngày 02/11/2020

**MỤC LỤC BẢNG PHÂN CÔNG CÔNG VIỆC**

[Bảng 1: Công việc ngày 12/10/2020 2](#_Toc55351755)

[Bảng 2: Công việc ngày 13/10/2020 3](#_Toc55351756)

[Bảng 3: Công việc ngày 14/10/2020 3](#_Toc55351757)

[Bảng 4: Công việc ngày 19/10/2020 4](#_Toc55351758)

[Bảng 5: Công việc ngày 20/10/2020 5](#_Toc55351759)

[Bảng 6: Công việc ngày 21/10/2020 5](#_Toc55351760)

[Bảng 7: Công việc ngày 26/10/2020 6](#_Toc55351761)

[Bảng 8: Công việc ngày 27/10/2020 6](#_Toc55351762)

[Bảng 9: Công việc chung của nhóm ngày 27/10/2020 6](#_Toc55351763)

[Bảng 10: Công việc ngày 02/11/2020 7](#_Toc55351764)

[Bảng 11: Công việc chung của nhóm ngày 02/11/2020 8](#_Toc55351765)

[Bảng 12: Công việc ngày 03/11/2020 8](#_Toc55351766)

[Bảng 13: Công việc chung của nhóm ngày 02/11/2020 9](#_Toc55351767)

**MỤC LỤC**

[**1.** **GIỚI THIỆU ĐỒ ÁN** 15](#_Toc55355533)

[**1.1.** **Trang Login + Remember** 16](#_Toc55355534)

[**1.1.1.** **Giao Diện** 16](#_Toc55355535)

[**1.1.2.** **Code chức năng login :** 16](#_Toc55355536)

[**1.2.** **Trang register + thêm user của admin** 18](#_Toc55355537)

[**1.2.1.** **Giao diện:** 18](#_Toc55355538)

[**1.2.2.** **Code chức năng :** 20](#_Toc55355539)

[**1.3.** **Trang Index** 22](#_Toc55355540)

[**1.3.1.** **Giao diện :** 22](#_Toc55355541)

[**1.3.2.** **Code chức năng :** 23](#_Toc55355542)

[**1.4.** **Trang đổi mật khẩu** 25](#_Toc55355543)

[**1.4.1.** **Giao diện :** 25](#_Toc55355544)

[**1.5.** **Trang ListUser + chức năng sắp xếp + tìm kiếm** 27](#_Toc55355545)

[**1.5.1.** **Giao diện :** 27](#_Toc55355546)

[**1.5.2.** **Code chức năng :** 27](#_Toc55355547)

[**1.6.** **Trang xem chi tiết user của admin** 31](#_Toc55355548)

[**1.6.1.** **Giao diện :** 31](#_Toc55355549)

[**1.6.2.** **Code chức năng :** 31](#_Toc55355550)

[**1.7.** **Trang sửa thông tin user của admin** 32](#_Toc55355551)

[**1.7.1.** **Giao diện :** 32](#_Toc55355552)

[**1.8.** **Chức năng xóa user của admin** 34](#_Toc55355553)

[**1.8.1.** **Giao diện :** 34](#_Toc55355554)

[**1.8.2.** **Code chức năng :** 34](#_Toc55355555)

[**2.** **Viết SCSS** 35](#_Toc55355556)

[**3.** **Lỗi Bảo Mật** 38](#_Toc55355557)

[**3.1.** **Tấn công xss** 38](#_Toc55355558)

[**3.1.1.** **XSS là gì:** 38](#_Toc55355559)

[**3.1.2.** **XSS hoạt động như thế nào:** 38](#_Toc55355560)

[**3.1.3.** **Phòng chống XXS:** 40](#_Toc55355561)

[**3.1.4.** **Nguồn tài liệu:** 42](#_Toc55355562)

[**3.1.5.** **Ví dụ về XXS:** 42](#_Toc55355563)

[**3.2.** **Sensitive Data Exposure – Phơi nhiễm dữ liệu nhạy cảm** 44](#_Toc55355564)

[**3.2.1.** **Phơi nhiễm dữ liệu “nhạy cảm” là gì?** 44](#_Toc55355565)

[**3.2.2.** **Cách thức hoạt đông:** 45](#_Toc55355566)

[**3.2.3.** **Cách khắc phục:** 46](#_Toc55355567)

[**3.2.4.** **Nguồn tài liệu:** 47](#_Toc55355568)

[**3.3.** **Tấn công insecure Direct Object References** 47](#_Toc55355569)

[**3.3.1.** **Thế nào là IDOR?** 47](#_Toc55355570)

[**3.3.2.** **Cách thức hoạt động IDOR:** 48](#_Toc55355571)

[**3.3.3.** **Ví dụ:** 49](#_Toc55355572)

[**3.3.4.** **Cách ngăn chặn và khắc phục:** 51](#_Toc55355573)

[**3.4.** **Tấn công command execution.** 55](#_Toc55355574)

[**3.4.1.** **Command execution là gì ?** 55](#_Toc55355575)

[**3.4.2.** **Cách hoạt động:** 56](#_Toc55355576)

[**3.4.3.** **Cách khắc phục:** 56](#_Toc55355577)

[**3.5.** **Tấn công User enumeration** 58](#_Toc55355578)

[**3.5.1.** **User enumeration là gì:** 58](#_Toc55355579)

[**3.5.2.** **Cách hoạt động:** 59](#_Toc55355580)

[**3.5.3.** **Cách khắc phục:** 61](#_Toc55355581)

[**3.6.** **Tấn công CSRF.** 61](#_Toc55355582)

[**3.6.1.** **Csrf là gì?** 61](#_Toc55355583)

[**3.6.2.** **Csrf hoạt động như thế nào?** 61](#_Toc55355584)

[**3.6.3.** **Phòng chống Csrf.** 62](#_Toc55355585)

[**3.6.4.** **Ví dụ về Csrf.** 62](#_Toc55355586)

[**3.7.** **Tấn công sql injction** 63](#_Toc55355587)

[**3.7.1.** **Tìm hiểu về lỗi SQL injection và cho ví dụ minh họa** 63](#_Toc55355588)

[**3.7.2.** **Nguyên nhân thường gặp dễ bị lỗi tấn công của SQL injection** 64](#_Toc55355589)

[**3.8.** **Tấn công Broken Authentication** 65](#_Toc55355590)

**MỤC LỤC HÌNH ẢNH DEMO WEBSITE**

[Hình 1. Giao diện Login 17](#_Toc55357061)

[Hình 2. Nếu đã login thì điều hướng vào trang Index.php 17](#_Toc55357062)

[Hình 3. Code chức năng Login 18](#_Toc55357063)

[Hình 4. Tổng quan trang Register 19](#_Toc55357064)

[Hình 5. Tổng quan trang thêm User của Admin 20](#_Toc55357065)

[Hình 6. Kiểm tra phân quyền register 21](#_Toc55357066)

[Hình 7. Kiểm tra tồn tại khi nhấn Register 21](#_Toc55357067)

[Hình 8. Kiểm tra thông tin file, đổi tên file và chức năng add user 22](#_Toc55357068)

[Hình 9. Function update 23](#_Toc55357069)

[Hình 10. Function register user 23](#_Toc55357070)

[Hình 11. Trang index của admin 23](#_Toc55357071)

[Hình 12. Trang Index của User 24](#_Toc55357072)

[Hình 13. Code cập nhập khi có chọn hình ảnh 25](#_Toc55357073)

[Hình 14. Code cập nhật khi không chọn hình ảnh 25](#_Toc55357074)

[Hình 15. Code function cập nhật thông tin user 26](#_Toc55357075)

[Hình 16. Giao diện trang đổi mật khẩu 26](#_Toc55357076)

[Hình 17. Code hiển thị form đổi mật khẩu trong trang index.php 27](#_Toc55357077)

[Hình 18. Code đổi mật khẩu 27](#_Toc55357078)

[Hình 19. Function đổi mật khẩu 28](#_Toc55357079)

[Hình 20. Giao diện trang danh sách user 28](#_Toc55357080)

[Hình 21. Giao diện sắp xếp 28](#_Toc55357081)

[Hình 22. Giao diện tìm kiếm 29](#_Toc55357082)

[Hình 23. Chức năng phân trang 29](#_Toc55357083)

[Hình 24. Code hiển thị danh sách user 30](#_Toc55357084)

[Hình 25. Function phân trang và sắp xếp 30](#_Toc55357085)

[Hình 26. Function tìm kiếm 30](#_Toc55357086)

[Hình 27. Chức năng tìm kiếm 31](#_Toc55357087)

[Hình 28. Chức năng sắp xếp 31](#_Toc55357088)

[Hình 29. Giao diện xem chi tiết user 32](#_Toc55357089)

[Hình 30. Chức năng xem user của admin 33](#_Toc55357090)

[Hình 31. Hiển thị thông tin user 33](#_Toc55357091)

[Hình 32. Function lấy thông tin user theo custom\_id 33](#_Toc55357092)

[Hình 33. Giao diện sửa user của admin 34](#_Toc55357093)

[Hình 34. Chức năng sửa user kèm file 35](#_Toc55357094)

[Hình 35. Chức năng sửa user không kèm file 35](#_Toc55357095)

[Hình 36. Giao diện nút xóa 36](#_Toc55357096)

[Hình 37. Giao diện hỏi trước khi xóa 36](#_Toc55357097)

[Hình 38. Code hiển thị nút xóa 36](#_Toc55357098)

[Hình 39. Code chức năng xóa user của admin 36](#_Toc55357099)

[Hình 40. Function xóa user 37](#_Toc55357100)

[Hình 41. Khai báo biến trong scss 37](#_Toc55357101)

[Hình 42. Impor scss và cấu trúc scss 38](#_Toc55357102)

[Hình 43. Extension convert scss to css (id = ritwickdey.live-sass) 39](#_Toc55357103)

**MỤC LỤC HÌNH ẢNH BẢO MẬT**

[Ảnh 1: Mô tả Xss là gì. 41](#_Toc55376382)

[Ảnh 2. Các trường dễ xảy ra tấn công XSS 42](#_Toc55376383)

[Ảnh 3: Mô tả cách tấn công XSS 43](#_Toc55376384)

[Ảnh 4. Sử dụng framework để tăng cường bảo mật 44](#_Toc55376385)

[Ảnh 5. Cách phòng chống tấn công XSS 45](#_Toc55376386)

[Ảnh 6: Thêm script vào trường fullname 45](#_Toc55376387)

[Ảnh 7: Thông báo lỗi XSS 46](#_Toc55376388)

[Ảnh 8. Đã loại bỏ thẻ tags script khi xuất ra thông tin fullname 46](#_Toc55376389)

[Ảnh 9. Tự động đổi các kí tự đặt biệt khi xuất ra thông tin fullname 47](#_Toc55376390)

[Ảnh 10. Mô tả tấn công phơi nhiễm dữ liệu nhạy cảm. 48](#_Toc55376391)

[Ảnh 11. Cách tấn công Sensitive Data Exposure 49](#_Toc55376392)

[Ảnh 12. Bảo mật SSL ( https) 50](#_Toc55376393)

[Ảnh 13. Mô tả cách nhận biết IDOR. 51](#_Toc55376394)

[Ảnh 14. Cách thức hoạt động của IDOR 52](#_Toc55376395)

[Ảnh 15. Mô tả cách hoạt động theo demo 53](#_Toc55376396)

[Ảnh 16. Các trường có thể bị tấn công 53](#_Toc55376397)

[Ảnh 17. Cách ngăn chặn và khắc phục 54](#_Toc55376398)

[Ảnh 18. Tạo column để chứa ID mã hóa trong table 56](#_Toc55376399)

[Ảnh 19. Viết function để mã hóa ID 57](#_Toc55376400)

[Ảnh 20. Xuất ra chuỗi ID đã được mã hóa 57](#_Toc55376401)

[Ảnh 21. Chuỗi ID đã được mã hóa trên thanh trình duyệt 57](#_Toc55376402)

[Ảnh 22. Chuyển đổi String thành mã hóa MD5 58](#_Toc55376403)

[Ảnh 23. Command execution 58](#_Toc55376404)

[Ảnh 24. Cách hoạt động của command execution 59](#_Toc55376405)

[Ảnh 25. Hình ảnh mô tá kết quả 59](#_Toc55376406)

[Ảnh 26. Hình ảnh mô tả Tạo command line trong php 60](#_Toc55376407)

[Ảnh 27. Sử dụng regex 60](#_Toc55376408)

[Ảnh 28. Sử dụng API để giảm thiểu rủi ro. 61](#_Toc55376409)

[Ảnh 29. Thông báo chi tiết khi nhập sai 62](#_Toc55376410)

[Ảnh 30. Lợi dụng thông báo như hình có thể xác định tồn tại 63](#_Toc55376411)

[Ảnh 31. Mô tả lỗi 404, 403 63](#_Toc55376412)

[Ảnh 32. Sử dụng Captcha để tăng cường bảo mật 63](#_Toc55376413)

[Ảnh 33. Khi điều kiện thất bại sẽ có khoảng thời gian đợi 64](#_Toc55376414)

[Ảnh 34. Sử dụng link giả mạo 65](#_Toc55376415)

[Ảnh 35. Code mô tả link giả mạo 66](#_Toc55376416)

[Ảnh 36. Mô tả ID số 6 bị xóa 66](#_Toc55376417)

[Ảnh 37. Mô tả SQL Injection 67](#_Toc55376418)

Chủ Đề

BẢO MẬT WEBSITE



# **GIỚI THIỆU ĐỒ ÁN**

Đồ án nhóm F xây dựng và sửa đổi theo source của giảng viên. Đồ án được nhóm thực hiện theo chia module ( TOPIC 3 ), các thành viên hoàn thành các module được nhóm trưởng phân công. Đồ án gồm có trang login, register, danh sách user, cập nhập thông tin, đổi mật khẩu, thêm user. Các chức năng có trên đồ án như :

* Login, Logout
* Register
* Xem, thêm, xóa, sửa user
* Đổi password
* Tìm kiếm phân trang
* Sắp xếp A-Z, Z-A
* Upload file, hình ( chỉ cho tải các file định dạng của office và hình ảnh có dung lượng dưới 2mb )

Đồ án áp dụng sử dụng SESSION, COOKIE ( TOPIC 4), thực hiện bảo mật website ( cụ thể như mã hóa id ) và biện pháp chống lỗi tấn công XSS.

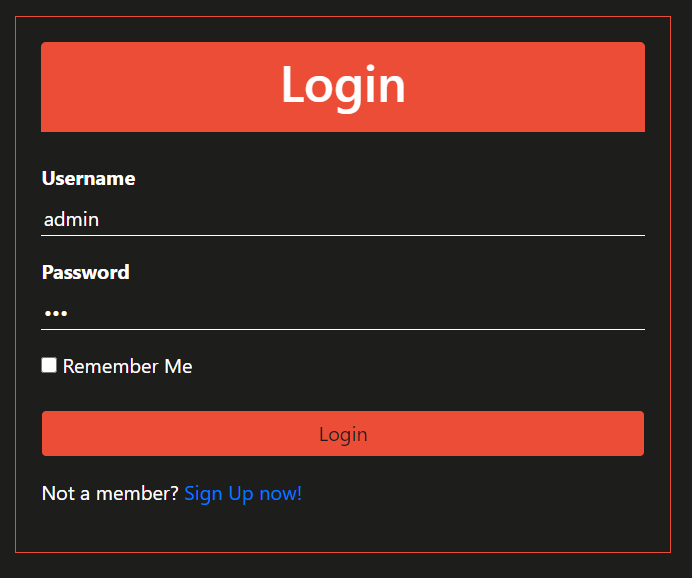
Đồ án của nhóm sử dụng bootstrap 4 để thực hiện dàn trang, các form, và sử dụng fontawesome 5 dùng icon làm website thân thiện với người dùng.

Ngoài ra, sử dụng SCSS ( TOPIC 2) để viết CSS theo cách viết ngôn ngữ lập trình. SCSS có cấu trúc rõ ràng, dễ phát triển, dễ sửa đổi code trong quá trình sử dụng.

Thông qua đồ án, nhóm tìm lỗi đồng thời tìm hiểu các cách tấn công website như SQL Injection, XSS,... và thông qua các lỗi đó tìm hiểu cách bảo mật website và fix các bug trong quá trình làm đồ án

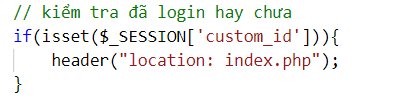
## **Trang Login + Remember**

### **Giao Diện**



Hình 1. Giao diện Login

### **Code chức năng login :**



Hình 2. Nếu đã login thì điều hướng vào trang Index.php

Hàm Strip\_tags() trong php dùng để loại các thẻ tags có nguy cơ làm cho website dính lỗi bảo mật.

Khi login thành công

=> Tạo ra 2 SESSION là custom\_id và user\_type ( custom\_id để biết user nào đang login, user\_type dùng để kiểm tra quyền truy cập của user đang login)

=> Nếu người dùng nhấn vào nút Remember tạo ra 2 cookie là username và password để lưu thông tin đăng nhập người dùng.

=> Điều hướng người dùng vào trang index

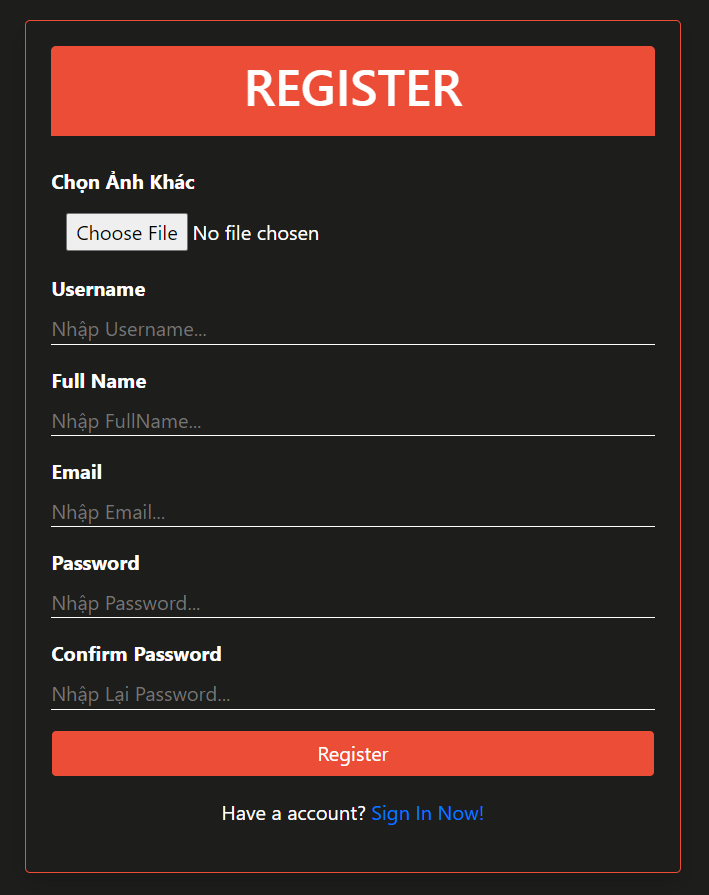
Khi login không thành công, thông báo lỗi “ wrong username or password!!!”



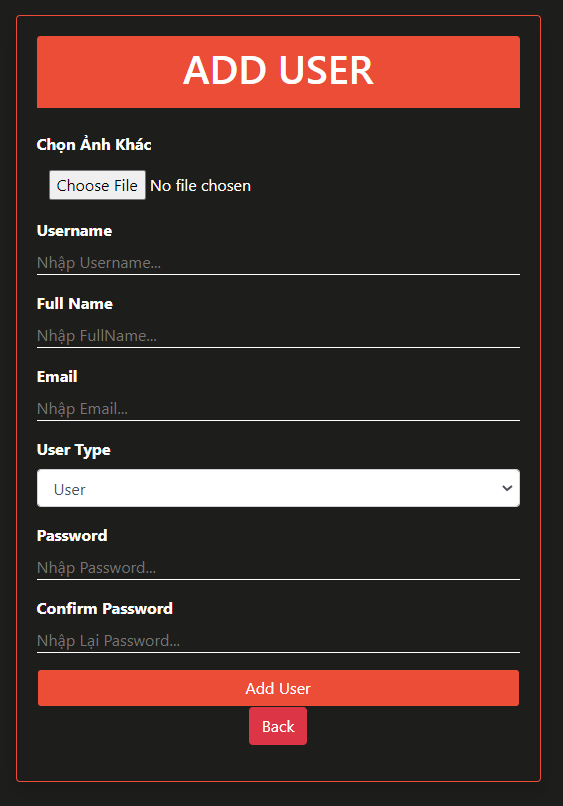
Hình 3. Code chức năng Login

## **Trang register + thêm user của admin**

### **Giao diện:**



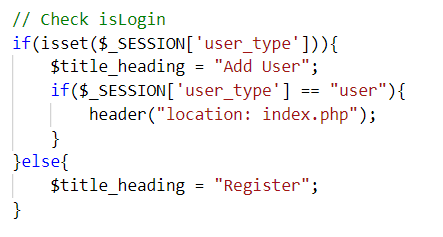
Hình 4. Tổng quan trang Register



Hình 5. Tổng quan trang thêm User của Admin

### **Code chức năng :**

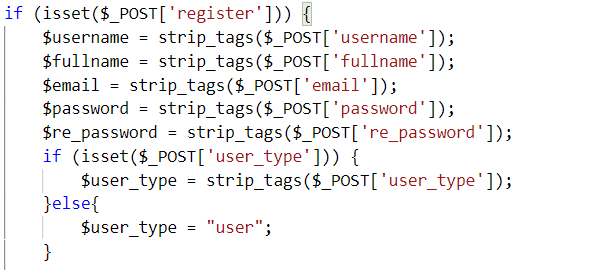
Nếu admin đã login thì thành trang thêm user, còn lại thành trang register



Hình 6. Kiểm tra phân quyền register

Nếu người dùng nhấn vào nút Register

=>Nếu tồn tại input user\_type ( admin ) thì lấy giá trị của user\_type, không thì mặc định là user



Hình 7. Kiểm tra tồn tại khi nhấn Register

Lấy thông tin file/hình ảnh được chọn ( tên file, path, kích thước, định dạng file, tmp ).

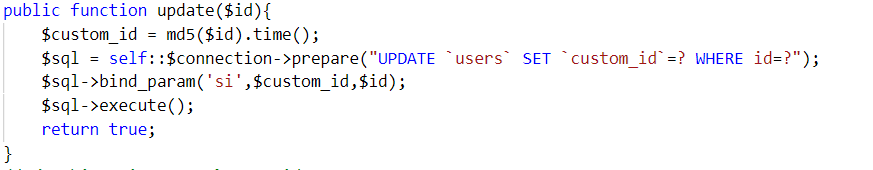
Đổi lại tên file để tránh bị trùng tên file, dùng hàm str\_shuffle và hàm random\_int để random tên không bị trùng lặp.

Kiểm tra kích thước ảnh, kiểm tra định dạng file để chặn những file nguy hiểm như \*.php, \*.js, \*...., cho website.

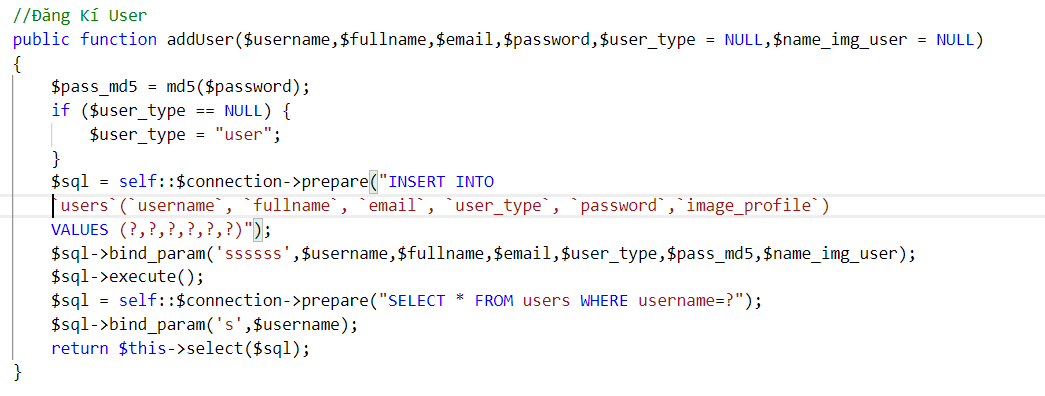
Thêm user và update lại custom\_id.



Hình 8. Kiểm tra thông tin file, đổi tên file và chức năng add user



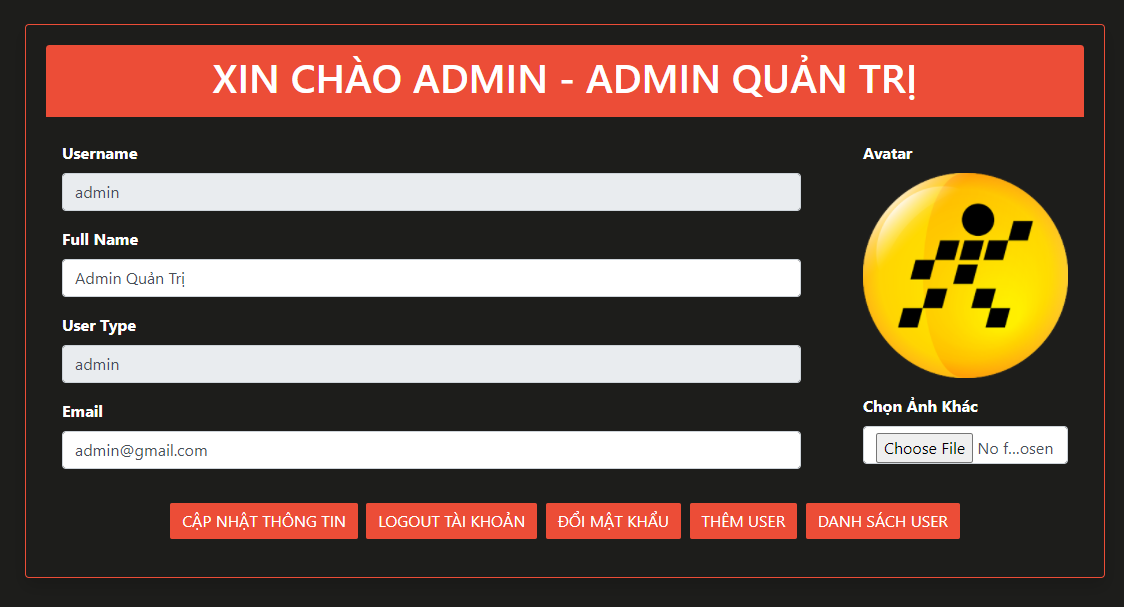
Hình 9. Function update



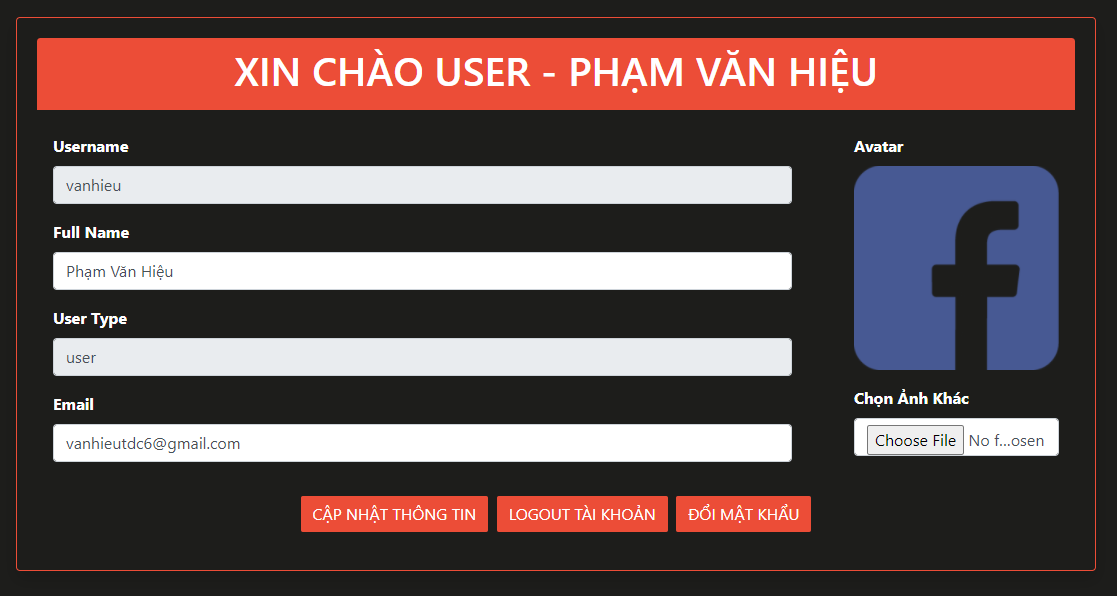
Hình 10. Function register user

## **Trang Index**

### **Giao diện :**



Hình 11. Trang index của admin



Hình 12. Trang Index của User

### **Code chức năng :**

* + Phân quyền admin và user

Có 2 trường hợp :

-Chọn hình ảnh : sẽ cập nhật thông tin user và file, hình ảnh.

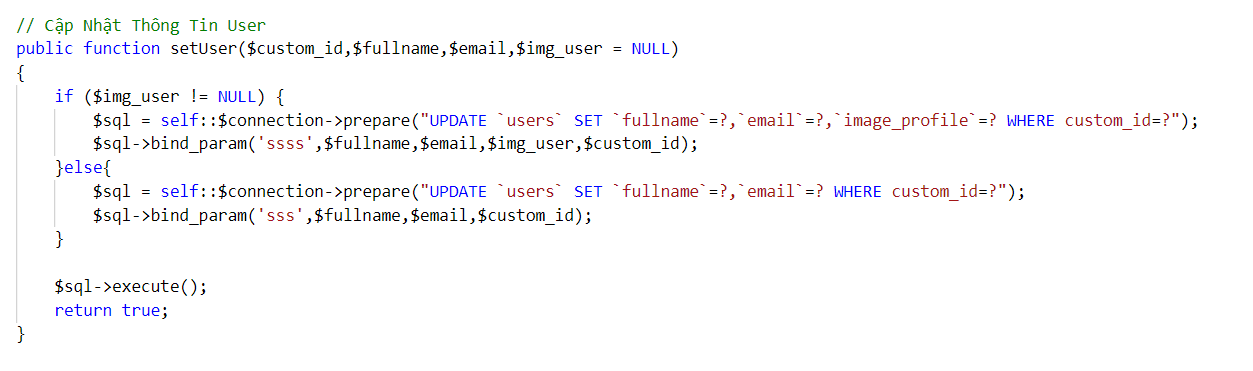


Hình 13. Code cập nhập khi có chọn hình ảnh

-Không chọn hình ảnh: sẽ cập nhật thông tin user mới, giữ nguyên hình ảnh cũ.



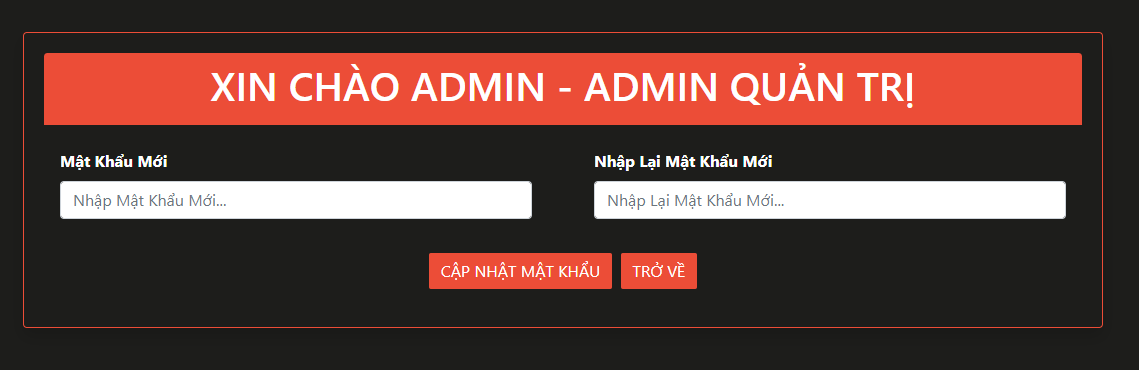
Hình 14. Code cập nhật khi không chọn hình ảnh



Hình 15. Code function cập nhật thông tin user

## **Trang đổi mật khẩu**

### **Giao diện :**



Hình 16. Giao diện trang đổi mật khẩu

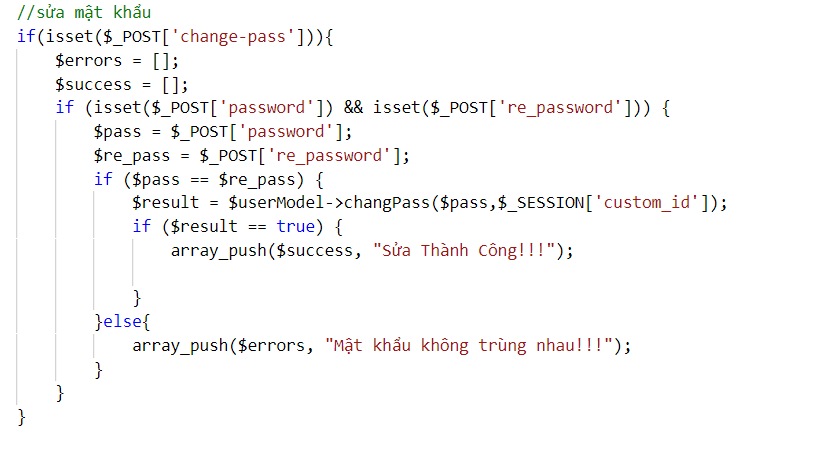
### **Code chức năng :**

Nếu người dùng nhấn vào nút đổi mật khẩu thì hiển thị ra form đổi mật khẩu.



Hình 17. Code hiển thị form đổi mật khẩu trong trang index.php

Nếu người dùng chọn vào đổi mật khẩu thì kiểm tra trường re\_password có trùng với trường password không, nếu trùng thì đổi mật khẩu, nếu không xuất thông báo “ mật khẩu không trùng nhau !!!”.



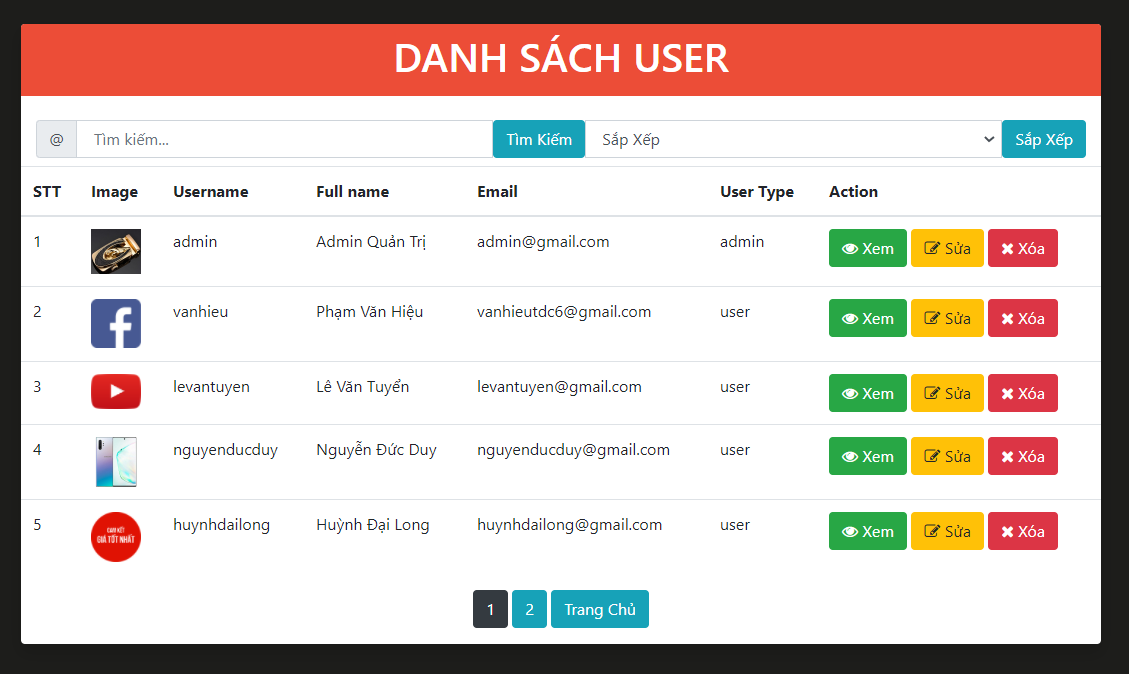
Hình 18. Code đổi mật khẩu



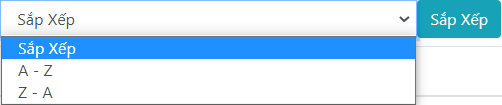
Hình 19. Function đổi mật khẩu

## **Trang ListUser + chức năng sắp xếp + tìm kiếm**

### **Giao diện :**



Hình 20. Giao diện trang danh sách user



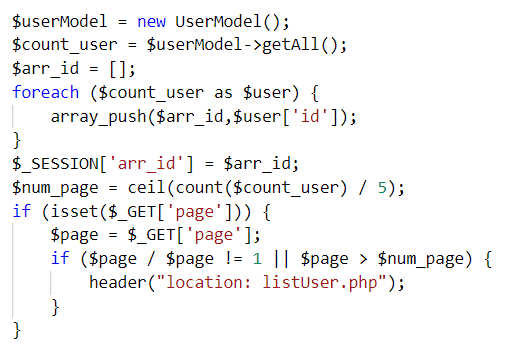
Hình 21. Giao diện sắp xếp



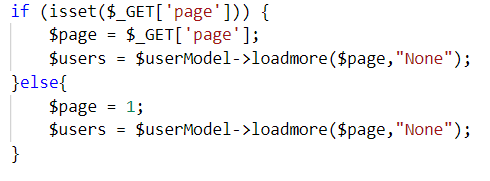
Hình 22. Giao diện tìm kiếm

### **Code chức năng :**

Nếu page không phải là số nguyên hoặc page lớn số trang user thì sẽ quay về trang listuser.php



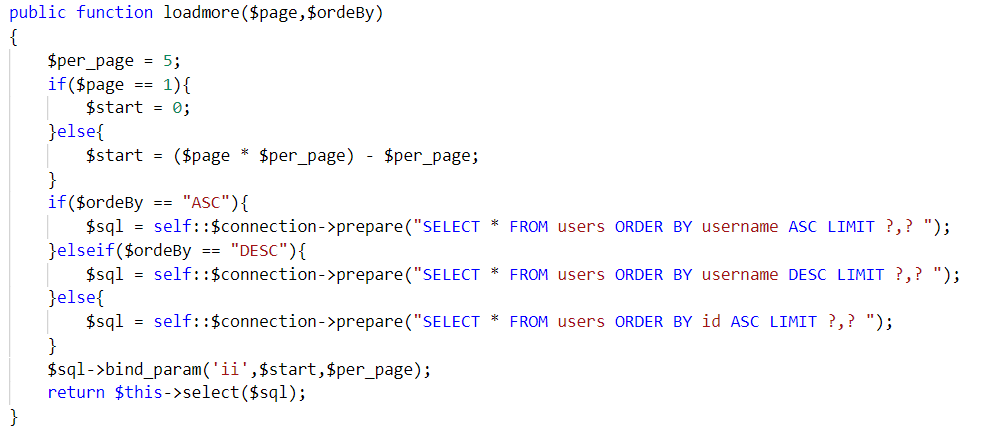
Hình 23. Code kiểm tra phân trang của danh sách user



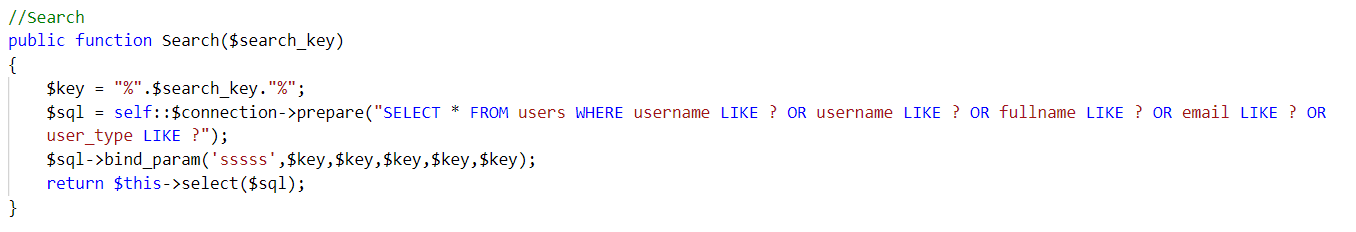
Hình 23. Chức năng phân trang



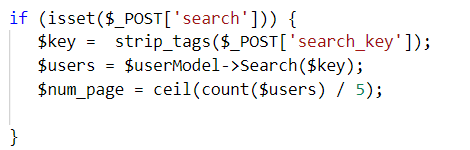
Hình 24. Code hiển thị danh sách user



Hình 25. Function phân trang và sắp xếp



Hình 26. Function tìm kiếm



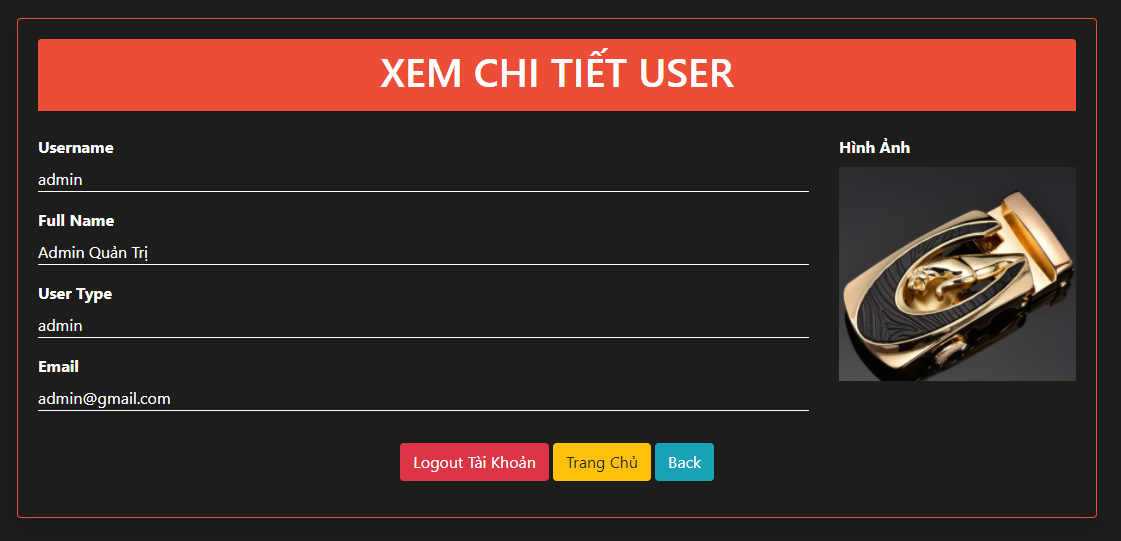
Hình 27. Chức năng tìm kiếm



Hình 28. Chức năng sắp xếp

## **Trang xem chi tiết user của admin**

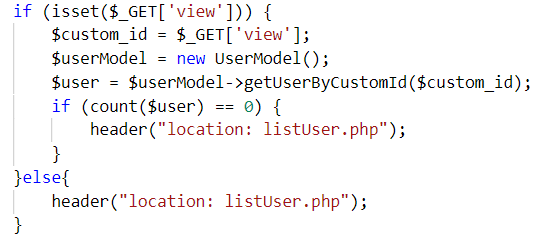
### **Giao diện :**



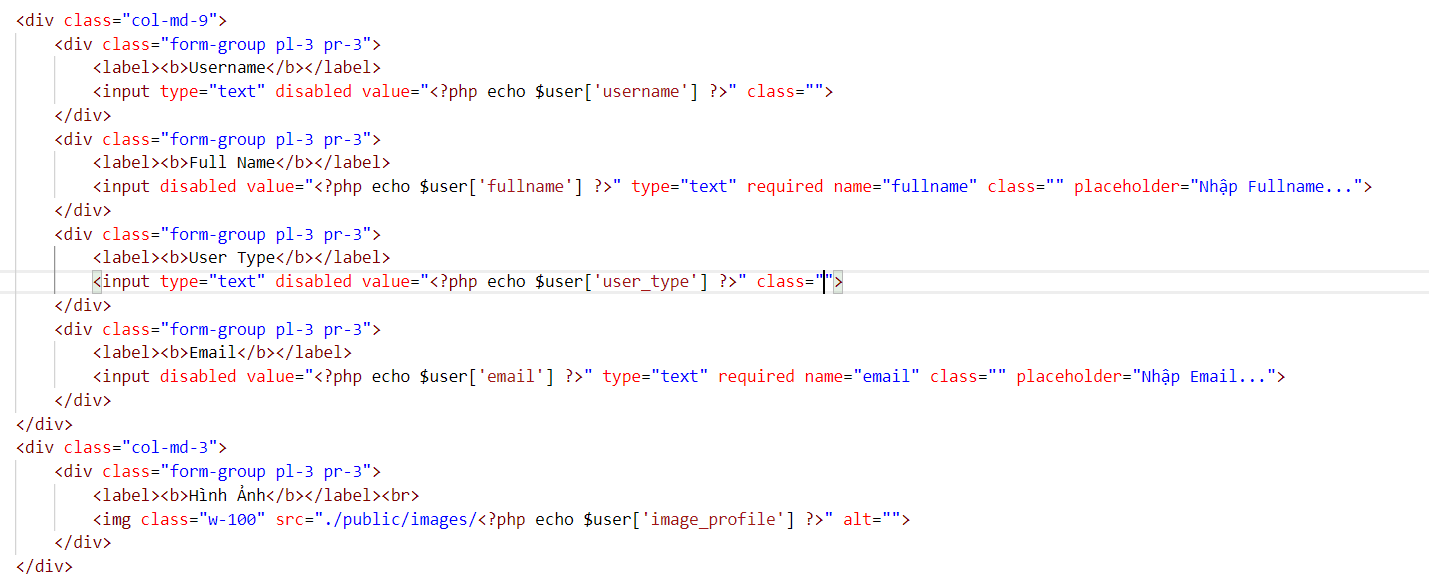
Hình 29. Giao diện xem chi tiết user

### **Code chức năng :**

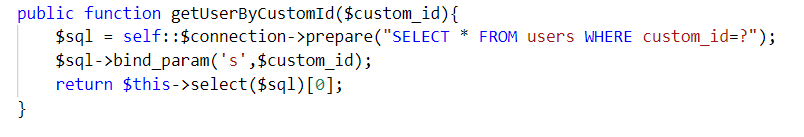
Chỉ xem không được sửa



Hình 30. Chức năng xem user của admin



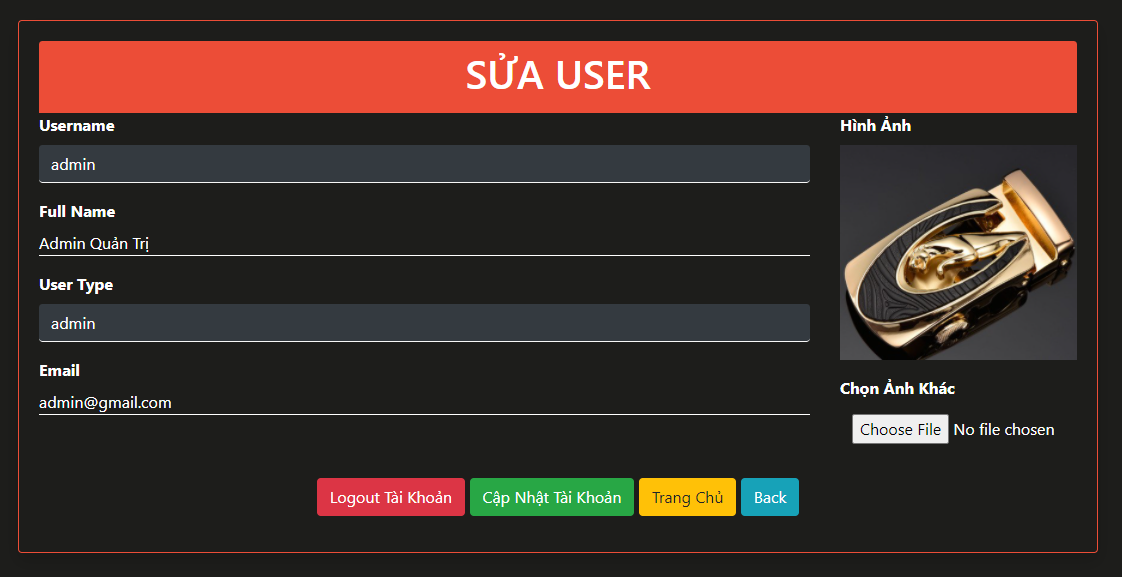
Hình 31. Hiển thị thông tin user



Hình 32. Function lấy thông tin user theo custom\_id

## **Trang sửa thông tin user của admin**

### **Giao diện :**



Hình 33. Giao diện sửa user của admin

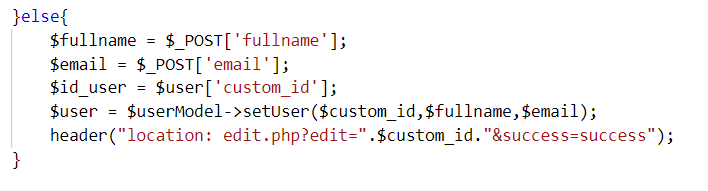
### **Code chức năng**

Sửa thông tin user kèm file, kiểm tra file.



Hình 34. Chức năng sửa user kèm file

Sửa thông tin user không kèm file.



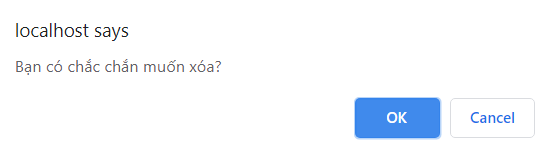
Hình 35. Chức năng sửa user không kèm file

## **Chức năng xóa user của admin**

### **Giao diện :**



Hình 36. Giao diện nút xóa

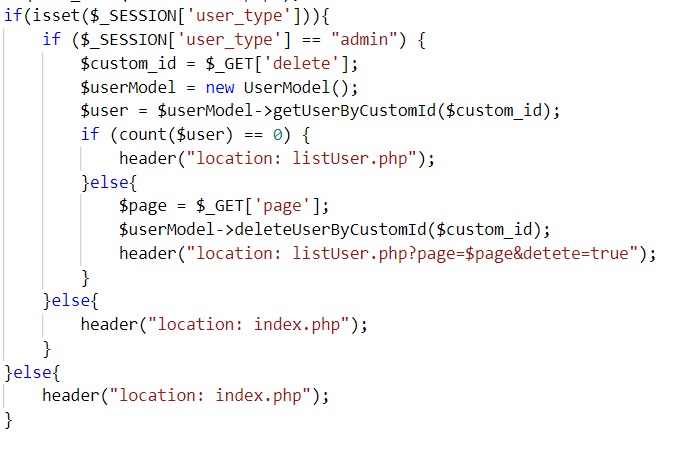


Hình 37. Giao diện hỏi trước khi xóa

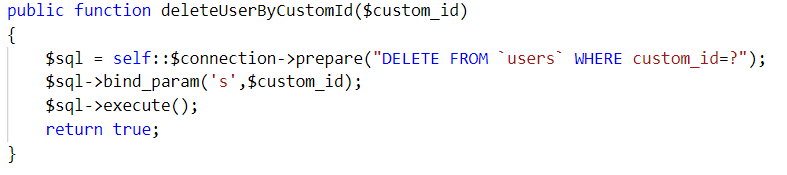
### **Code chức năng :**



Hình 38. Code hiển thị nút xóa

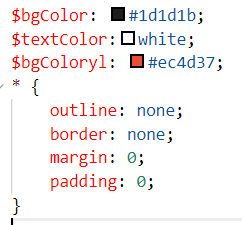


Hình 39. Code chức năng xóa user của admin

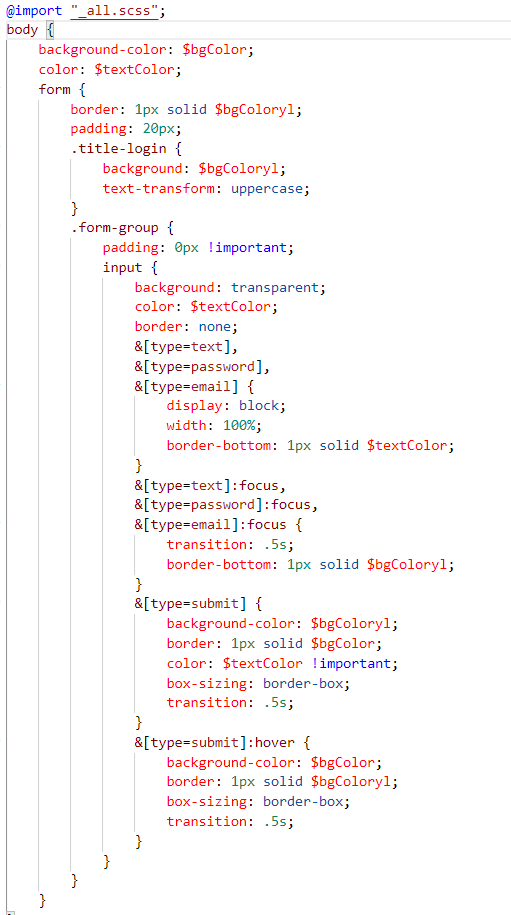


Hình 40. Function xóa user

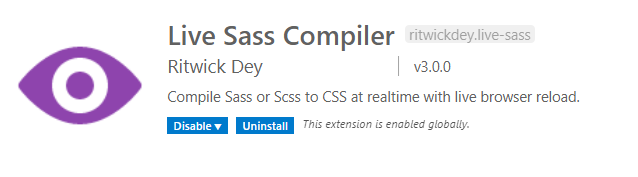
# **Viết SCSS**



Hình 41. Khai báo biến trong scss



Hình 42. Impor scss và cấu trúc scss



Hình 43. Extension convert scss to css (id = ritwickdey.live-sass)

# **Lỗi Bảo Mật**

## **Tấn công xss**

### **XSS là gì:**

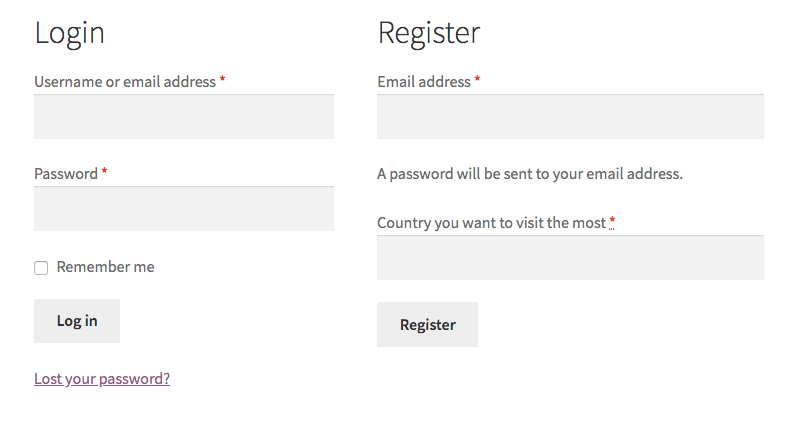
* + Xss là viết tắt của cụm từ Cross-Site Scripting.
  + Là kĩ thuật tấn công khá phổ biến hiện nay dễ dàng sử dụng nhưng lại mang đến hậu quả vô cùng nghiêm trọng cho website.
  + Kể cả những website không dùng database vẫn sẽ sử dụng được loại tấn công này.



Ảnh 1: Mô tả Xss là gì.

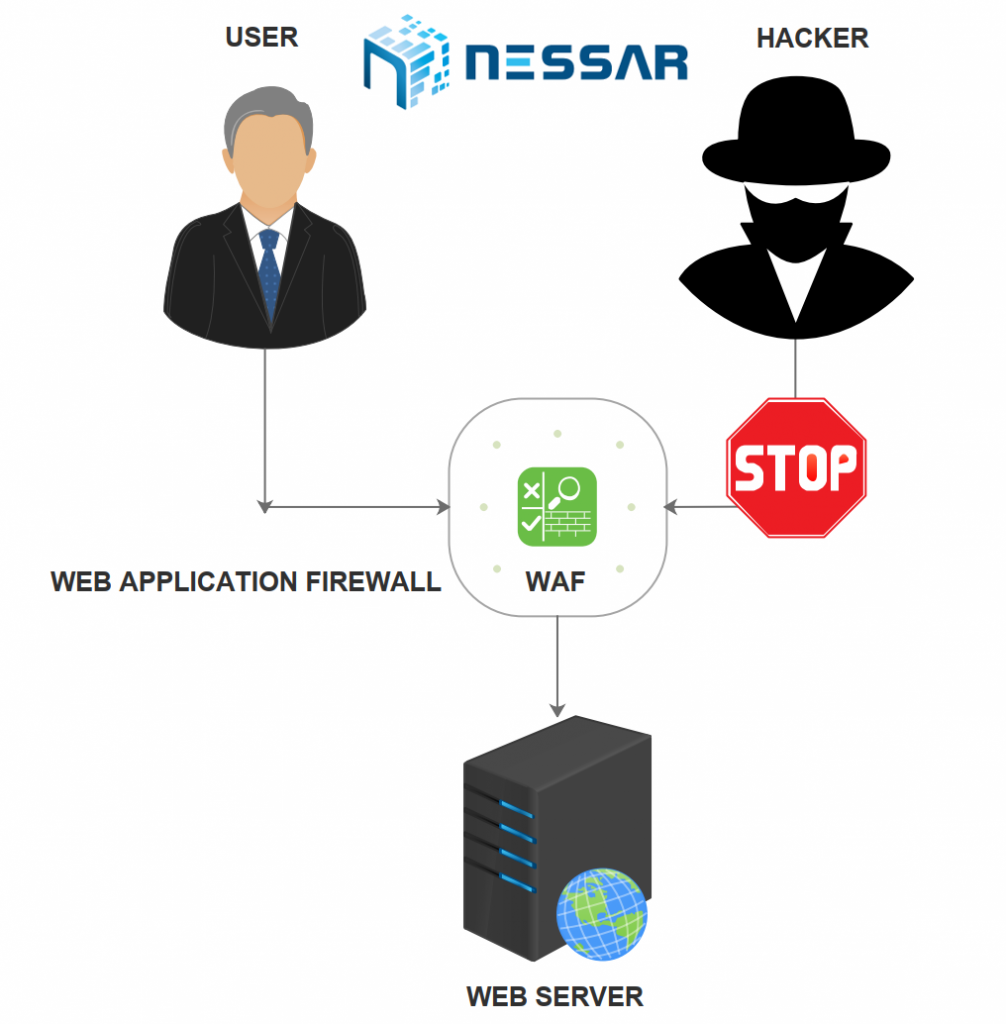
### **XSS hoạt động như thế nào:**

* + Xss hoạt động trong các trường hợp nhập dữ liệu như: login,chat, comment, sửa thông tin và search ...



Ảnh 2. Các trường dễ xảy ra tấn công XSS

* + Khi người dùng nhập vào các trường đó nhưng chúng ta lại không dùng các kĩ thuật kiểm tra dữ liệu trước khi thực hiện thì khi user nhập những đoạn code nhúng vào trong website sẽ rất nguy hiểm đối với website của chúng ta và gây hậu quả nặng nề như mất thông tin người dùng ...



Ảnh 3: Mô tả cách tấn công XSS

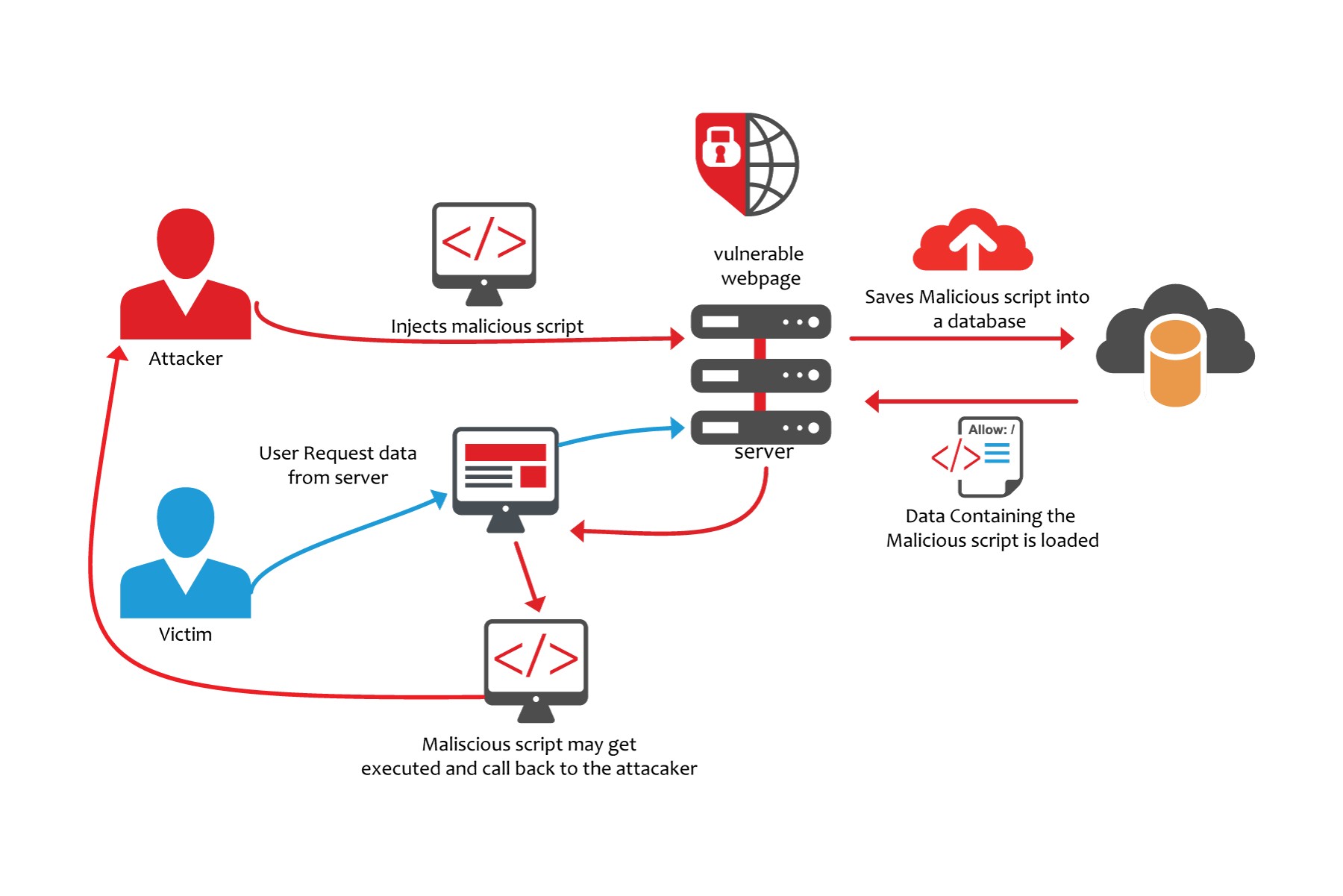
### **Phòng chống XXS:**

* + Sử dụng framework.



Ảnh 4. Sử dụng framework để tăng cường bảo mật

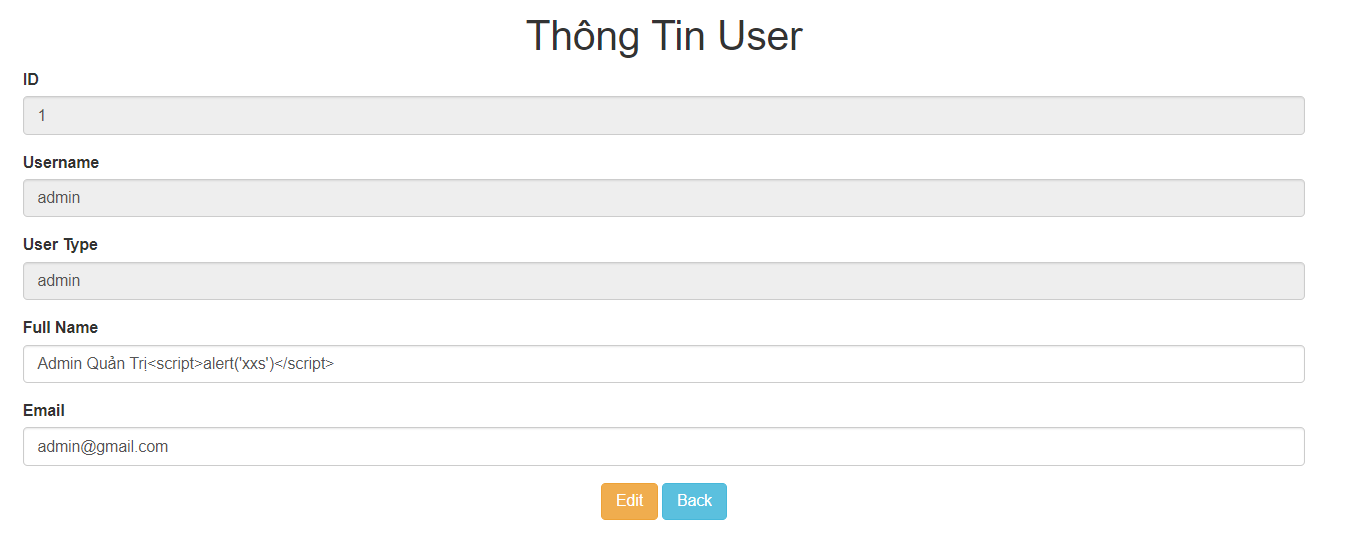
* + Lọc dữ liệu trước khi thực hiện một chức năng nào đó khi có người dùng nhập vào.
  + Sử dụng hàm strip\_tags() trong php để lọc các thẻ một cách an toàn.
  + Sử dụng hàm mysqli\_real\_escape\_string() trong php để thay đổi các kí tự lạ thành chuỗi khác để không làm ảnh hưởng đến website.

Ảnh 5. Cách phòng chống tấn công XSS

### **Nguồn tài liệu:**

* + <https://www.hacksplaining.com/prevention/xss-stored>

### **Ví dụ về XXS:**

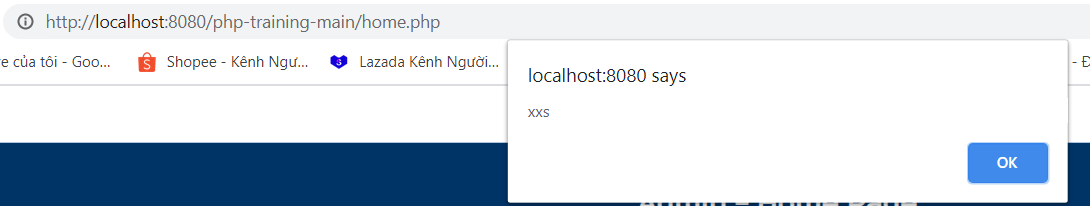
* + form edit không kiểm tra dữ liệu nhập vào và xuất ra:
  + thêm phần script: <script>alert('xxs')</script> vào trường fullname dữ liệu được thêm vào database. 

Ảnh 6: Thêm script vào trường fullname

* + Kết quả khi xuất ra ngoài:

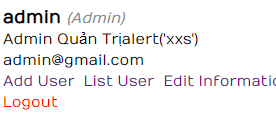


Hình 7



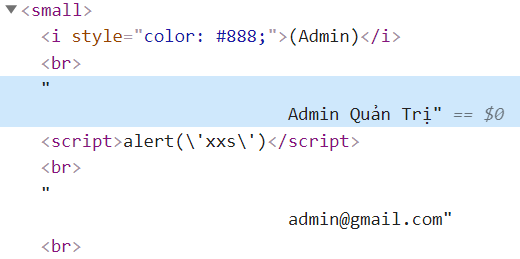
Ảnh 7: Thông báo lỗi XSS

* + Vậy là website đã bị tấn công bởi xxs.
  + Cách để vô hiệu hóa đơn giản là:
  + 1: thêm hàm strip\_tags() để xuất các trường thì sẽ vô hiệu hóa thẻ script này.



Ảnh 8. Đã loại bỏ thẻ tags script khi xuất ra thông tin fullname

* Hoặc sử dụng mysqli\_real\_escape\_string() trong php để thay đổi các kí tự đặc biệt.

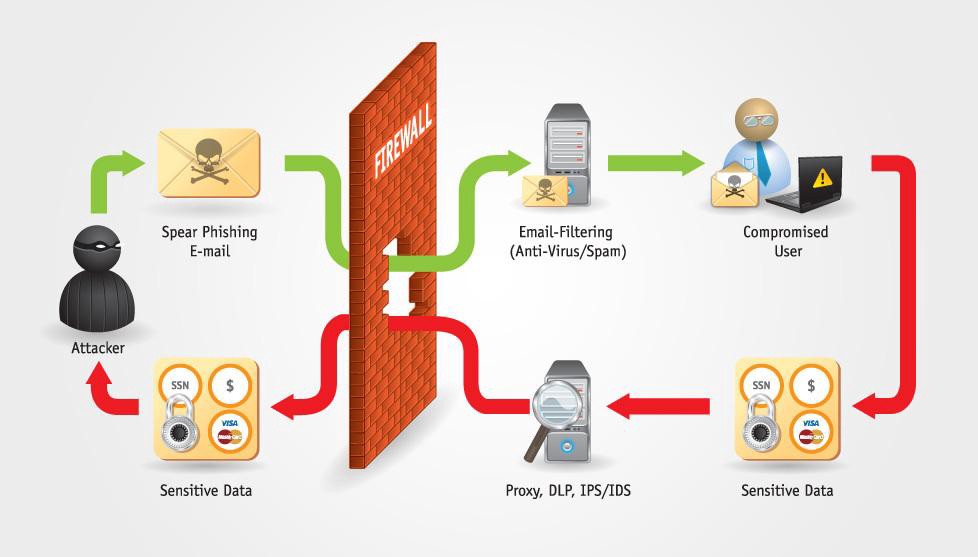


Ảnh 9. Tự động đổi các kí tự đặt biệt khi xuất ra thông tin fullname

## **Sensitive Data Exposure – Phơi nhiễm dữ liệu nhạy cảm**

### **Phơi nhiễm dữ liệu “nhạy cảm” là gì?**

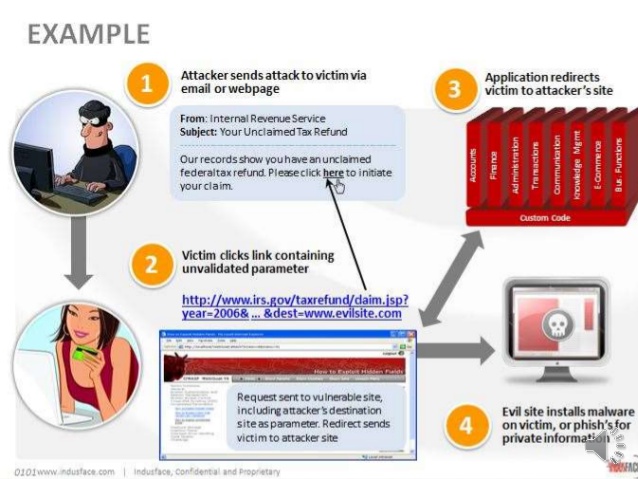
* + - là chúng liên quan đến việc vô tình tiết lộ thông tin nhạy cảm mà lẽ ra phải được bảo mật bằng mật mã.
    - Và nó dùng để chiếm quyền làm chủ user, lấy thông tin cá nhân, tài khoản ngân hàng và nhiều thông tin quan trọng khác....



Ảnh 10. Mô tả tấn công phơi nhiễm dữ liệu nhạy cảm.

### **Cách thức hoạt đông:**

* + - Một hacker đã tìm thấy một lỗ hổng bảo mật trong website của bạn? Hoặc có thể một nhân viên IT quên không bảo mật dữ liệu? Nếu như thông tin người dùng và doanh nghiệp của bạn không được bảo mật đúng cách, tin tặc và các bên trái phép khác có thể truy cập. Và việc công bố dữ liệu này thể hiện rủi ro đáng kể cho các công ty và cá nhân bạn.



Ảnh 11. Cách tấn công Sensitive Data Exposure

### **Cách khắc phục:**

* + - Cài đặt và gia hạn chứng chỉ SSL cho website của bạn.
    - Sử dụng mật khẩu có độ phức tạp cao.
    - Sử dụng các thuật toán mã hóa mới nhất.



Ảnh 12. Bảo mật SSL ( https)

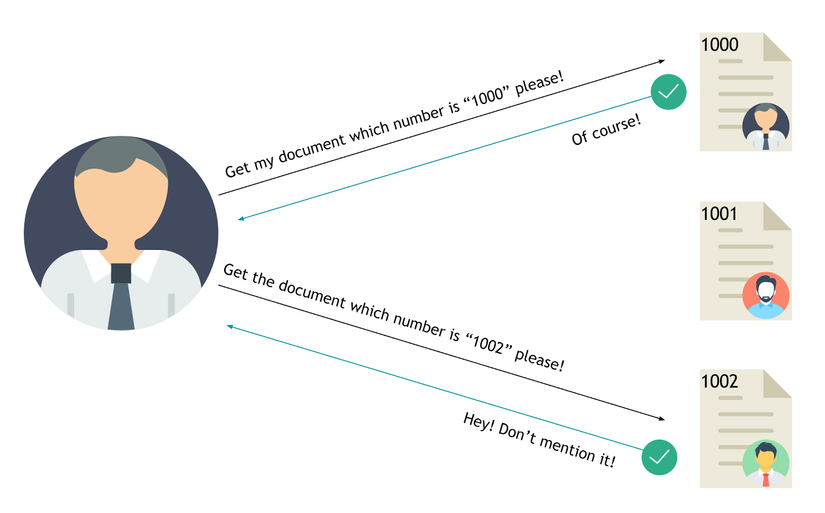
### **Nguồn tài liệu:**

* + - <https://hdivsecurity.com/owasp-sensitive-data-exposure>
    - <https://ropesec.com/articles/sensitive-data-exposure/>

## **Tấn công insecure Direct Object References**

### **Thế nào là IDOR?**

- Lỗi Insecure Direct Object References(IDOR) là một lỗ hỏng bảo mật rất nghiêm trọng và một là một lập trình viên điều đơn giản nhất phải biết lỗi này.



Ảnh 13. Mô tả cách nhận biết IDOR.

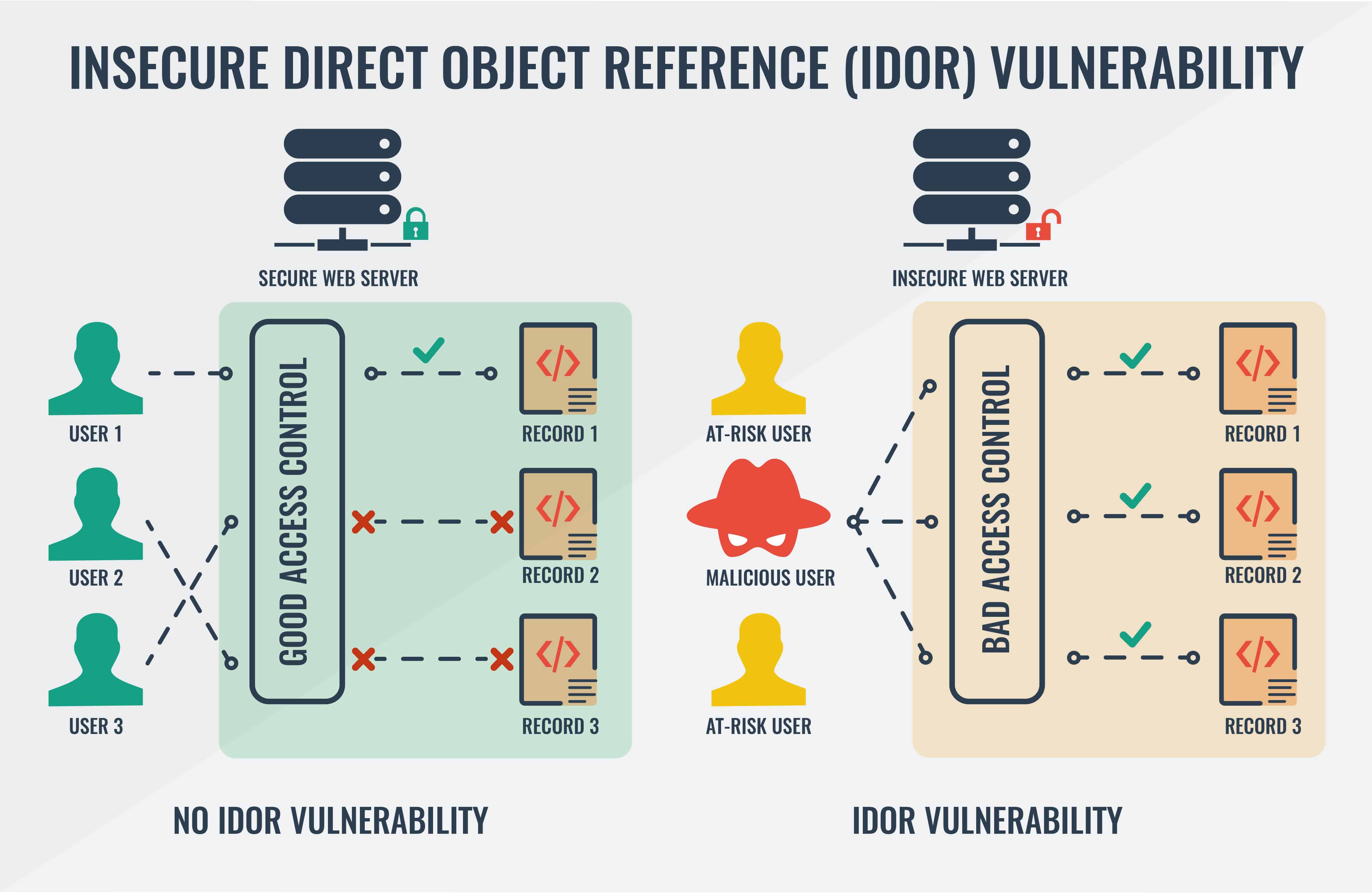
- Lỗi này là một lỗi căn bản nằm ở cấp độ một trong mười hai cấp độ bảo mật. Nói như vậy chắc các bạn thấy loại này tuy đơn giản nhưng chúng ta không biết cách khắc phục thì nó sẽ trở thành một lỗi rất nghiêm trọng ,nó sẽ gián tiếp là công cụ để các hacker khai thác lỗ hỏng của trang web của mình.

### **Cách thức hoạt động IDOR:**

- Lỗ hỏng IDOR này chúng ta có thể sử dụng được ủy quyền để lấy thông tin người dùng khác .

- Và có thể thay đổi giá trị id để có thể xem được thông tin của người dùng khác.

- Hoặc chứng ta có thể áp dụng IDOR thay đổi giá trị id lấy được từ database để có thể làm các chức năng như THÊM, SỬA, XÓA thông tin người dùng mặt dù chúng ta không đăng nhập vào tài khoản người dùng đó.



Ảnh 14. Cách thức hoạt động của IDOR

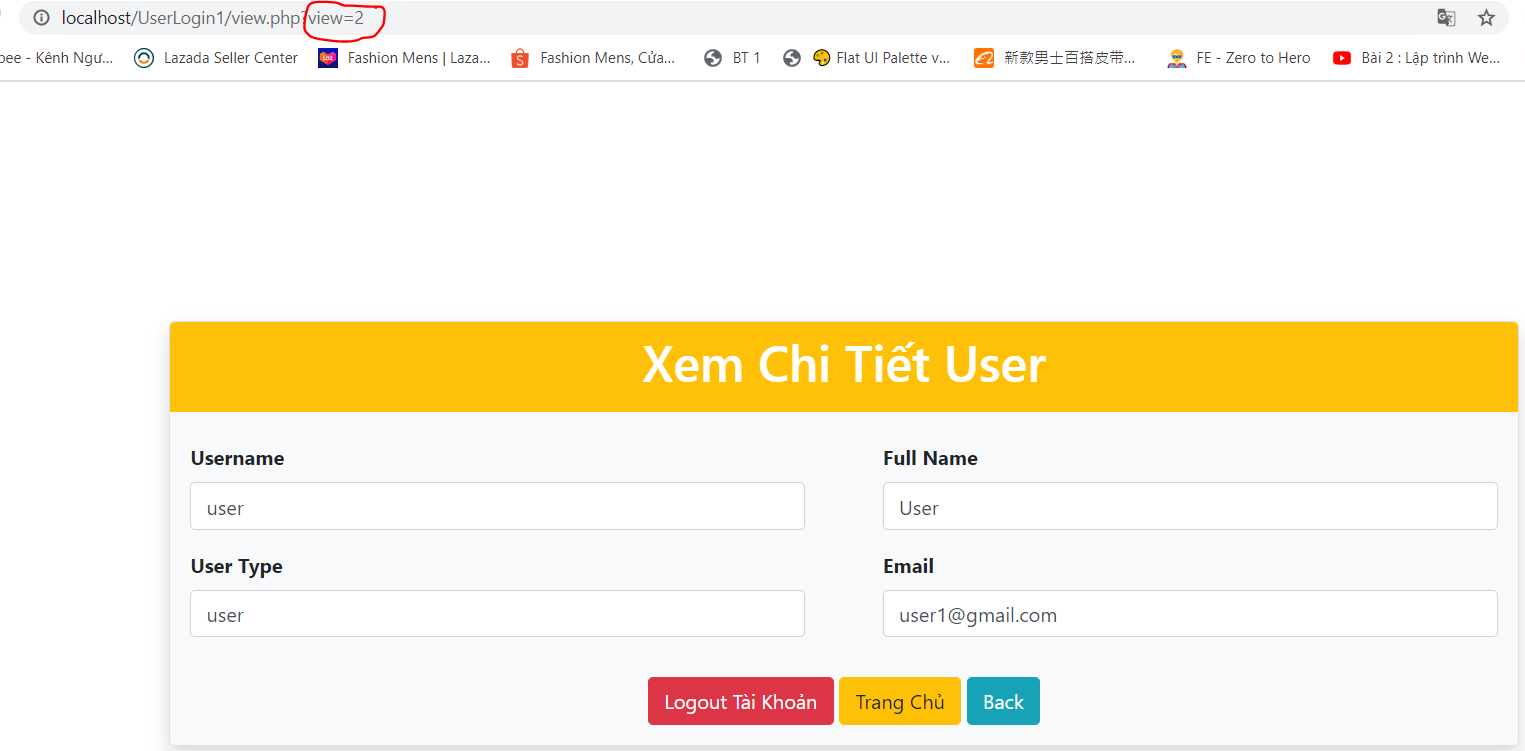
- Nghe thì chúng ta ai cũng muốn biết được cách này để hack vô tài khoản hoặc xóa tài khoản Facebook của những người mình không thích.

- Tuy nhiên như mình đã nói là lỗi này mỗi lập trình viên đều phải biết nên chúng ta sẽ không bao giờ thực hiện ở những trang web lớn. Lý do tại sao thì các bạn chắc đã biết rồi.

- Để biết rõ ràng lỗi này mình sẽ có những hình ảnh minh họa để các bạn hiểu rõ về lỗi này.

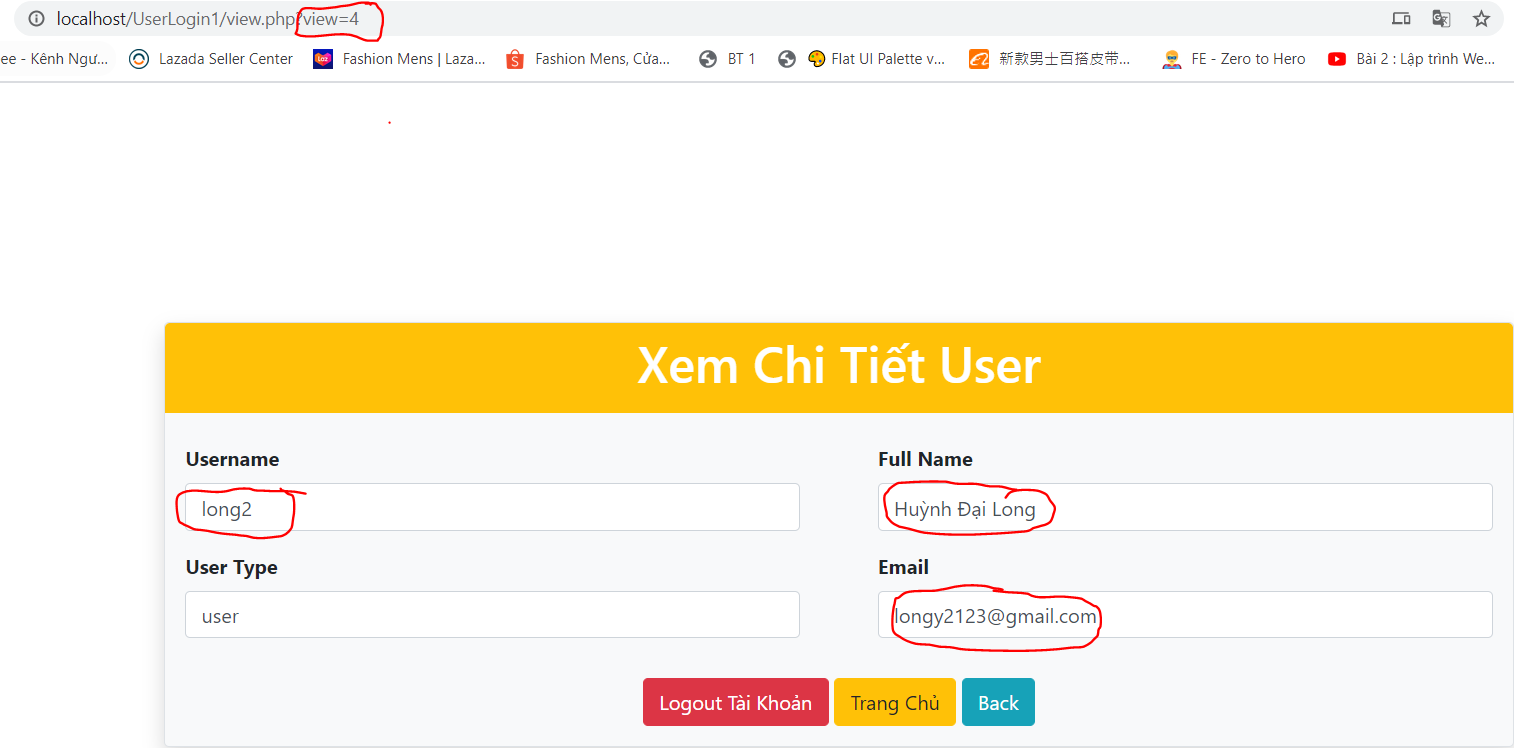
### **Ví dụ:**

* Chúng ta vào xem chi tiết của một thành viên mang id=2.
* http://localhost/UserLogin1/view.php?view=2.



Ảnh 15. Mô tả cách hoạt động theo demo

* + Sau đó mình đổi lại id trên thanh công bằng một số bất kì ví dụ như id=4.
  + <http://localhost/UserLogin1/view.php?view=4>.



Ảnh 16. Các trường có thể bị tấn công

* Như vậy chúng ta thấy chỉ cần chúng ta chỉ đăng nhập một tài khoản của mình vừa đăng kí xong chúng ta vào chỉ dùng một cách như vậy thì chúng ta đã xem được tấc cả các dữ liệu của các tài khoản ở trong database.
* Đây chỉ là một demo nhỏ của mình để cho các bạn hiểu về IDOR và các bạn hình dùng khi một lập trình viên muốn xây dựng một trang web:

+ Mua bán hàng : Chúng ta cần chú ý trang THANH TOÁN SẢN PHẨM . Để tránh trường hợp mua sản phẩm này và dùng thủ thuật này để sửa thành sản phẩm khác.

+ Trang trường mình TDC: Chúng ta cần chú ý trang ĐÓNG HỌC PHÍ.

+ Hacker sẽ dùng thủ thuật này để lấy dữ liệu cá nhân.Hoặc có thể phá banh cái databa của mình.

+ Chắc mình đưa ví dụ thực tiển như vậy vào chắc các bạn đã hiểu cách hoạt động của IDOR rồi chứ gì.

* Và tương tự như trang thêm sửa xóa các tài khoản trong database đều thực hiện như vậy được hết.Vì vậy lỗ hỏng IDOR này nhìn sơ qua thì thấy đơn giản nhưng thực chất nó rất là quan trọng.

### **Cách ngăn chặn và khắc phục:**

- Theo mình được biết, học hỏi và phân tích của thầy Phan Thanh Nhuần.

- Thì mình gồm biết được 3 cách để khắc phục.

#### **Ví dụ:**

Header

body

Ảnh 17. Cách ngăn chặn và khắc phục

- Quyền admin ( được cung cấp tất cả các quyền trên hệ thống ).

- Quyền User ( chỉ được phép sửa đổi chính nó ).

- Quyền guest ( chỉ được phép xem ).

- Trong một trang website, User chỉ được phép thay đổi thông tin

của chính nó ( cụ thể mỗi user hoặc sản phẩm có mỗi id đặc trưng)

- Khi User thay đổi thông tin của chính nó, từ client sẽ gửi thông tin

về Server, khi gửi thông tin về Server sẽ gửi 2 cái chính bao gồm :

header và body.

- Body bao gồm những thông yêu tin cầu chỉnh sửa của User.

- Header bao gồm những thông tin của User như trình duyệt, hệ điều

hành, địa chỉ, IP address,...

- Khi gửi thông tin về, phần header sẽ gửi tất cả thông tin của người

dùng, còn body gửi về yêu cầu. Khi gửi từ Client về Server, thông

tin từ Client sẽ đi xuống Server. Khi đến Server sẽ yêu cầu kiểm

tra xem có có tồn tại SESSION người dùng không.

- Server lúc này sẽ kiểm tra một mục riêng lẻ chỉ dùng để chứa

SESSION, nếu SESSION người dùng chưa tồn tại sẽ gửi thông báo

về phía Client và bắt đăng nhập để sửa thông tin. Khi Server kiểm

tra người dùng tồn tại thì phía Server sẽ kiểm tra ID của người

dùng ( sản phẩm) cần sửa.

- Vì lí do này, sẽ tạo ra lỗ hỏng của hệ thống. Cụ thể, khi muốn sửa

đổi thông tin của người dùng hoặc sản phẩm có ID = 2, người

dùng sửa đổi thông tin của ID thứ nhất nhưng trước khi nhất Save,

người dùng sẽ đổi ID thành một ID khác ví dụ như ID =4. Lúc này,

ở phía Server sẽ hiểu là sửa thông tin của ID = 4.

#### **Khi đó hướng giải quyết như thế nào ? Đề xuất biện pháp giải pháp như sau :**

- Chúng ta sẽ tạo thêm một cột là customer\_id liên kết với id.



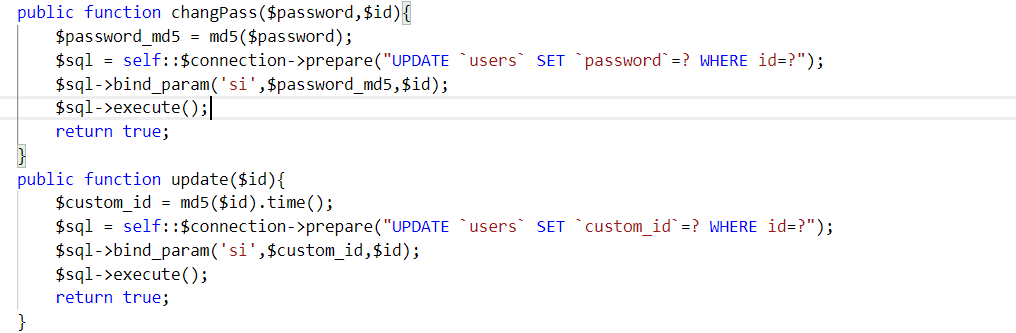
Ảnh 18. Tạo column để chứa ID mã hóa trong table

- Khi người dùng tạo một tài khoản sẽ mã hóa ID phía Client, và khi gửi chuỗi mã hóa ID đó về phía Server so sánh với ID được lưu ở phía Server, ID phía server khi tạo ở phía database cũng sẽ được mã hóa. Phía Server sẽ so sánh chuỗi ID ở phía Server với chuỗi ID ở phía Client. Khi 2 chuỗi mã hóa giống nhau sẽ cho phép sửa thông tin. Khi làm biện pháp này, sẽ giảm nguy cơ hacker sẽ tìm kiếm ra ID giống nhau nhưng sẽ tồn tại phát sinh như nếu hacker biết ID được mã hóa ở dạng nào.

#### **Vậy biện pháp thay thế là gì ? biện pháp đề xuất như sau:**

- Khi mã hóa chúng ta sẽ mã hóa ID kèm với 1 đoạn mã ngẫu nhiên

để hacker không dò ra đưọc như hình ở bên dưới.



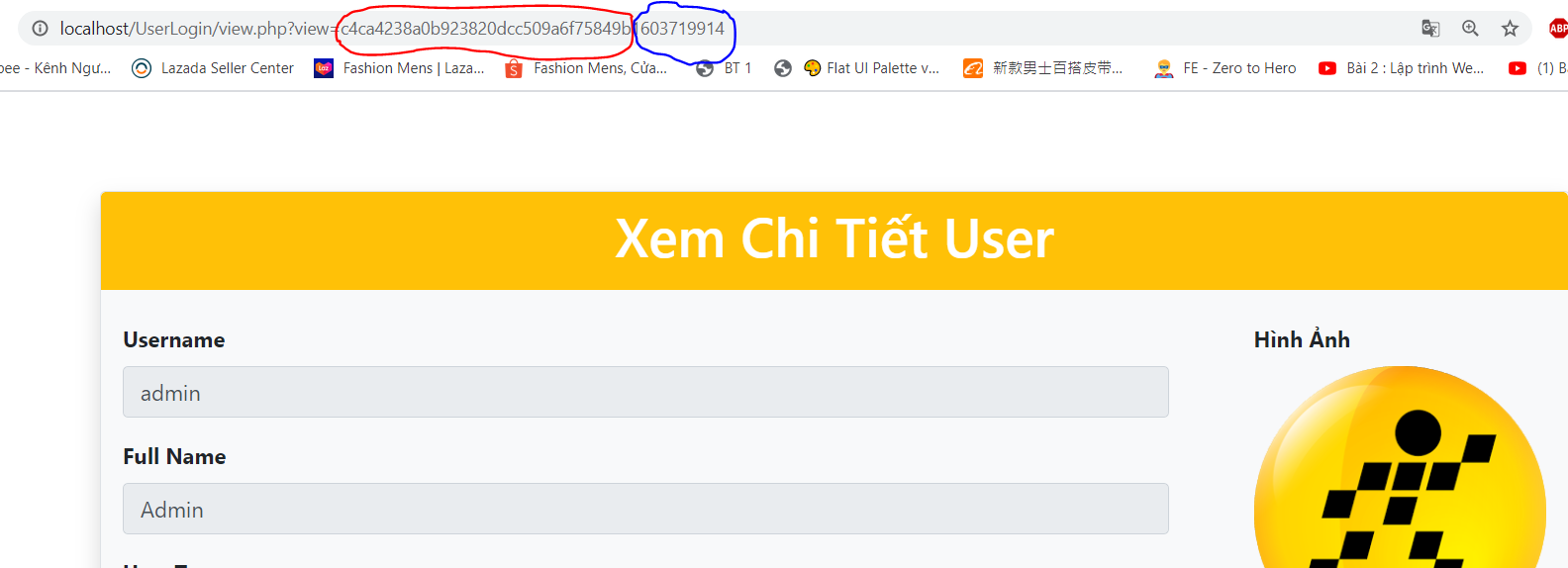
Ảnh 19. Viết function để mã hóa ID

- Để bảo mật thêm chúng ta sẽ cho thêm một hàm **time()** vào để khi người dùng bắt đầu tạo một tài khoản thì chúng ta sẽ mã hóa id bằng cách mã hóa md5 và lấy theo customer\_id và cộng thêm time().

- Hàm time() này mỗi khi chúng ta tạo ra một tài khoản thì nhiệm vụ vào hàm này sẽ tự quy định một chuỗi các kí tự số trả về cho chúng ta . Vì vậy hi hacker có biết chúng ta sửa dụng mã hóa md5 như mỗi hàm time() sẽ tự trả về một kí tự khác nhau nên hacker sẽ k biết chuỗi mã hóa của id nó như thế nào.

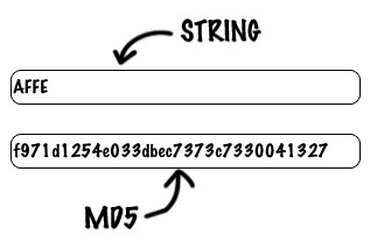


Ảnh 20. Xuất ra chuỗi ID đã được mã hóa



Ảnh 21. Chuỗi ID đã được mã hóa trên thanh trình duyệt

- Để cho dễ hình dung các bạn hãy nhìn vào hình trên chuỗi kí tự nằm trong vùng màu đỏ đó chính là chuỗi mã hóa md5 của id, và cái chính là vùng nằm trong vùng màu xanh đó chính là kết quả chúng ta nối chuỗi với hàm time() và nó sẽ tạo ra một chuỗi kí tự như vậy.Thì lúc đó hacker sẽ không khai thác được lỗ hỏng này nữa.



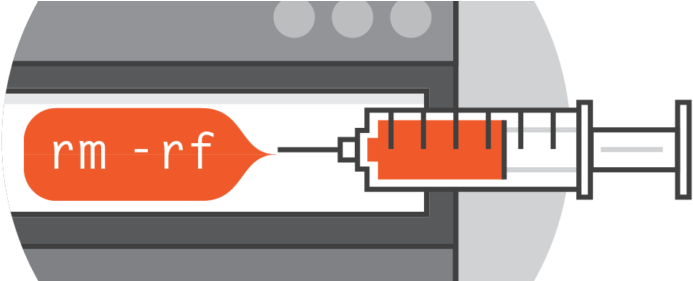
Ảnh 22. Chuyển đổi String thành mã hóa MD5

## **Tấn công command execution.**

### **Command execution là gì ?**

- Các trang web hiện nay thường gọi operating system processes thông qua command line, nên thao tác này sẽ tìm ẩn rất nhiều về lỗ hỏng bảo mật.

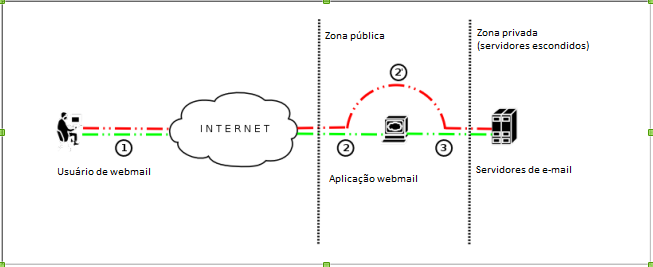
- Chúng ta thường sử dụng command line để đọc files, đọc images, kiểm tra gmail,...



Ảnh 23. Command execution

### **Cách hoạt động:**

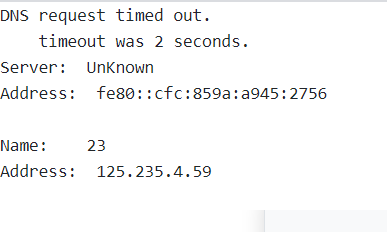
-Chúng ta đang sử dụng một trang wed đơn giản thực hiện edit một tài khoản. Wedsite sẽ thực hiện command tra cứu và cho chúng ta thấy hiển thị một kết quả ra một form như đoạn code trên.



Ảnh 24. Cách hoạt động của command execution

-Từ một biến $edit chưa được lọc, thì chúng ta đã tạo ra một lỗ hỏng và lỗ hỏng đó được gọi là command injection.

- Kêt quả cho thấy



Ảnh 25. Hình ảnh mô tá kết quả

### **Cách khắc phục:**

Tạo command line trong php:

shell\_exec "ls -l"

exec "ls -l"

passthru "ls -l"

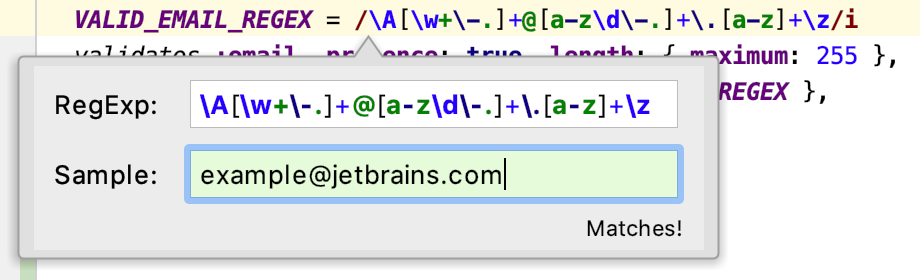
system "ls -l"

`ls -l`

Ảnh 26. Hình ảnh mô tả Tạo command line trong php

Espace inputs

Khi các kí tự như : ;, &, |. ` không được lọc kỹ sẽ xảy ra lỗi. Ngoài ra có thể dùng regex



Ảnh 27. Sử dụng regex

Hạn chế các command được cho phếp

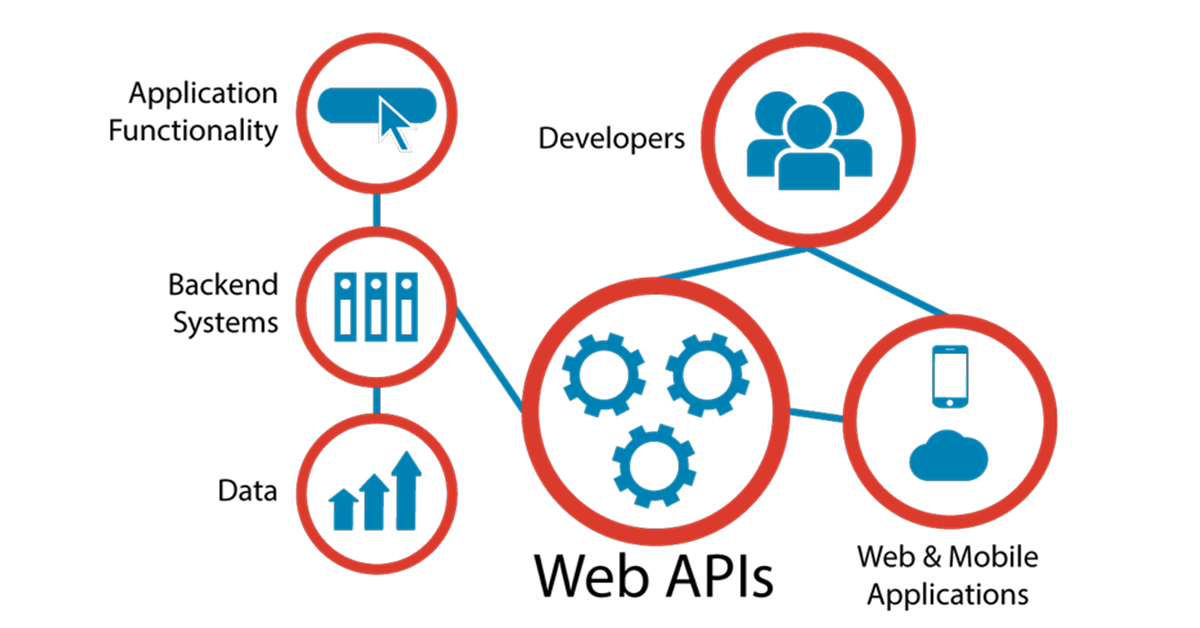
Tự xây dụng các Shell command thay vì người dùng tự nhập.

Kiểm tra code kỹ lưỡng

Chạy với quyền hạn chế

Set quyền cụ thể cho từng user, mỗi user chỉ nên cho 1 số quyền nhất định

Tránh gọi các lệnh trùng nhau ngôn ngữ lập trình hiện đại cho phép bạn đọc các tập tin, gửi email, và thực hiện các chức năng khác trên hệ thống. Sử dụng API bất cứ khi nào có thể, chỉ sử dụng shell khi nào cần thiết, điều này giúp giảm thiểu rủi do và cũng đơn giản hóa codebase của bạn



Ảnh 28. Sử dụng API để giảm thiểu rủi ro.

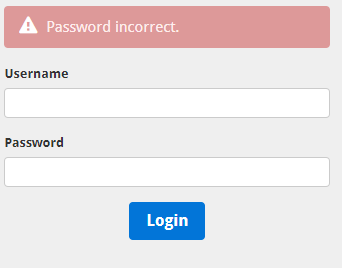
## **Tấn công User enumeration**

### **User enumeration là gì:**

User enumeration được hiểu như hacker lợi dụng những thông báo chi tiết, lỗ hỏng của web để liệt kê ra danh sách người dùng tồn tại và sử dụng các đoạn mã script, ứng dụng để dò các loại password yếu của người dùng.

Các kỹ thuật được sử dụng trong liệt kê có thể kể ra như:

* Kỹ thuật Win2k Enumeration : dùng để trích xuất thông tin tài khoản người dùng (user name).
* Kỹ thuật SNMP (Simple Network Management Protocol) để liệt kê thông tin người dùng.
* Kỹ thuật Active Directory Enumeration dùng trong liệt kê hệ thống Active Directory.
* Sử dụng Email IDs để tìm kiếm thông tin.

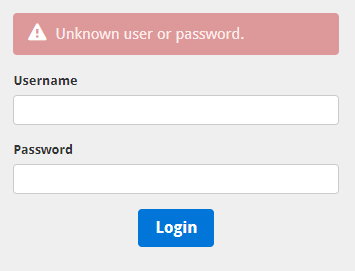
Ví dụ như :

Ảnh 29. Thông báo chi tiết khi nhập sai

### **Cách hoạt động:**

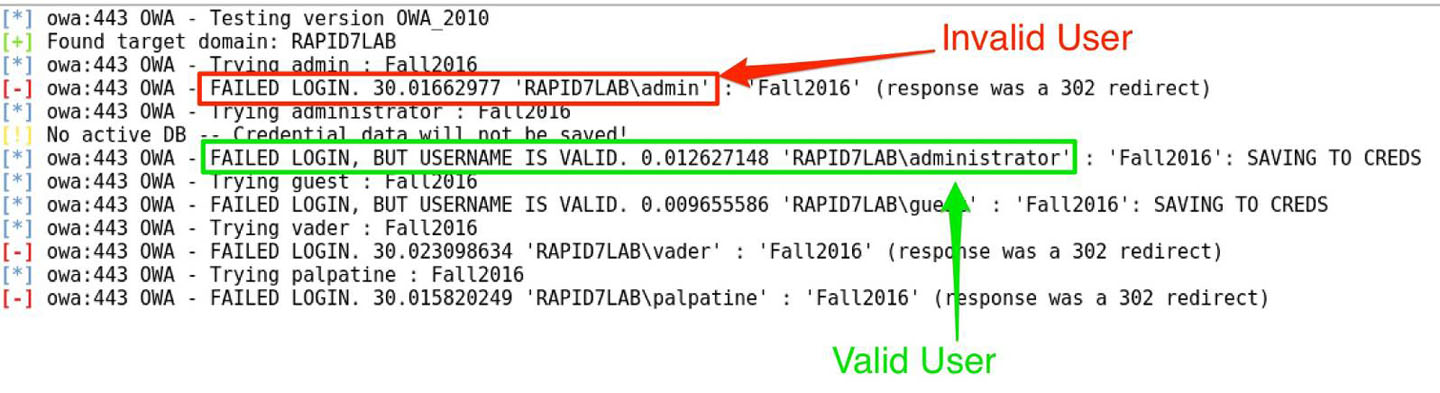
Hacker có thể lợi dụng những thông báo chi tiết như user không tồn tại, user hoặc password không đúng, password sai,… để xác định tài khoản người dùng đó có tồn tại trên dữ liệu web hay không.

Phương pháp tấn công user enumeration là dò tìm user, password yếu trên hệ thống bằng các phần mềm dò tự động vì vậy những biện pháp đề xuất để ngăn chặn hình thức tấn công này như sau :



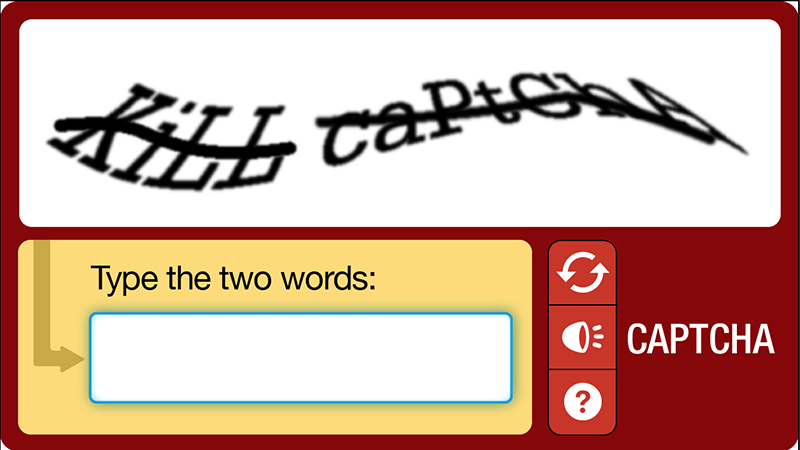
Ảnh 30. Lợi dụng thông báo như hình có thể xác định tồn tại

* Không hiển thị chi tiết thông báo trả về, chỉ hiển thị thông báo chung như không xác định được user hoặc password, người dùng hoặc mật khẩu không chính xác,..



Ảnh 31. Mô tả lỗi 404, 403

* Hacker dùng máy để quét dò user, password vì vậy nên chèn các xác nhận cấp 2 mà máy tính chưa đủ nhận biết để vượt qua được ( mã captcha ) vào các form login, register.



Ảnh 32. Sử dụng Captcha để tăng cường bảo mật

### **Cách khắc phục:**

* Sử dụng thuật toán Exponential backoff ( thuật toán được hiểu như là cách 1 khoảng thời gian khi đăng nhập thất bại )



Ảnh 33. Khi điều kiện thất bại sẽ có khoảng thời gian đợi

## **Tấn công CSRF.**

### **Csrf là gì?**

CSRF được viết tắt bởi từ “Cross-site Request Forgery” có nghĩa phương thức tấn công mà ở đó có yêu cầu truy cập các trang web một cách gian lận. Đối với cuộc tấn công này, thì các hacker thường dùng cách dụ dỗ người dùng thực hiện các hành động mà người đó không có ý định thực hiện.

### **Csrf hoạt động như thế nào?**

Khi đăng nhập vào một trình duyệt web thì sẽ lưu một Cookie.Các cooki thường được lưu ở một sesion. Và người dùng có thể đăng nhập lại bằng cookie đó mà không cần mật khẩu.

Một khi cookie đó vẫn còn hiệu lực, thì có rất nhiều cách để hack cơ lấy cookie đó. Ví dụ như tạo một trang web giả danh.

Sau đó , gửi link đó dụ cho người dùng click vào trang web đó. Thì người dùng đó sẽ bị hacker lấy được cookie.

### **Phòng chống Csrf.**

Để phòng chống cuộc tấn công CSRF nên thực hiện:

Đăng xuất khỏi các website.

Login vào một máy riêng.

Không nên click vào các website không quan trọng, những web sai có khả nghi.

Luôn xóa cookie thường xuyên

### **Ví dụ về Csrf.**

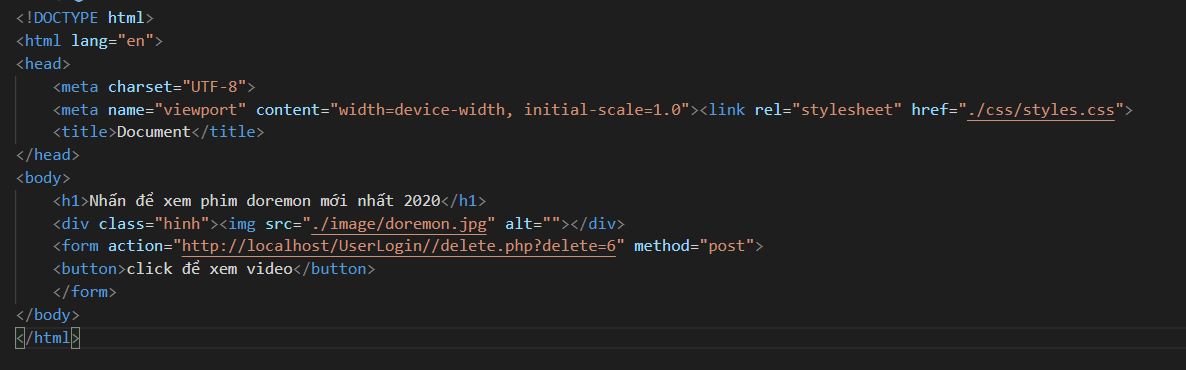
Người dùng có thể tạo một trang web giả mạo để lợi dụng tài khoản admin để xóa user bất kỳ.

* Giao diện:



Ảnh 34. Sử dụng link giả mạo

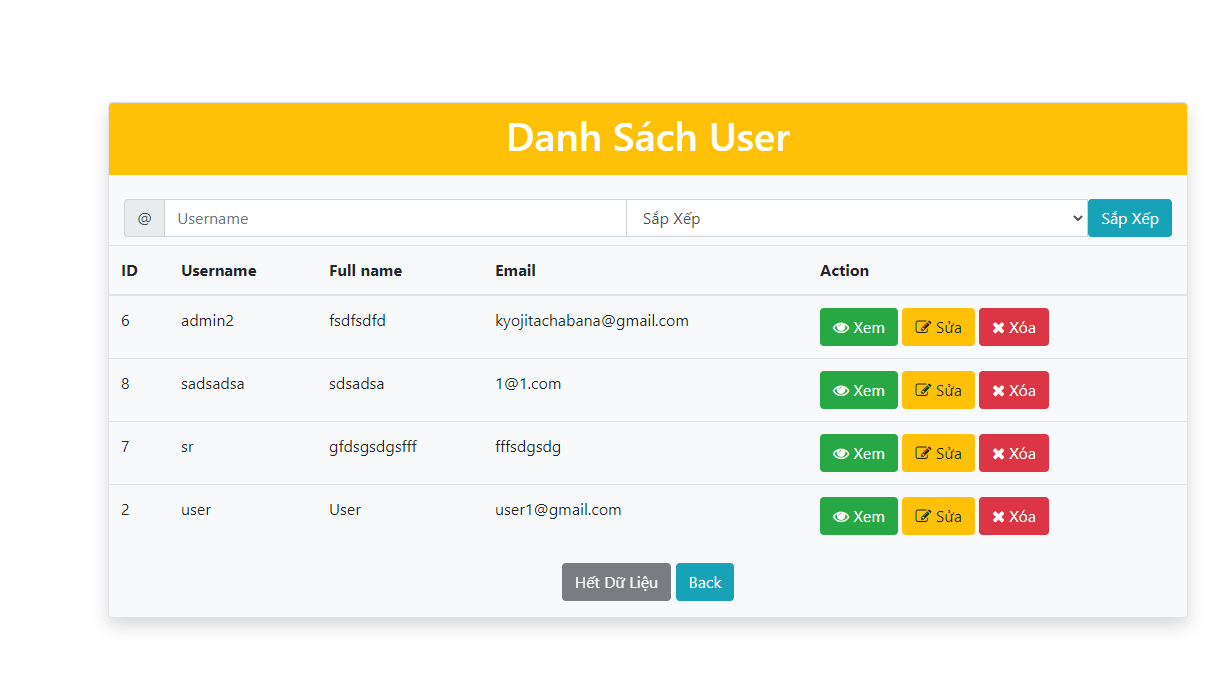
* Code:



Ảnh 35. Code mô tả link giả mạo

* Kết quả:

Id số 6 sẽ bị xóa khỏi danh sách



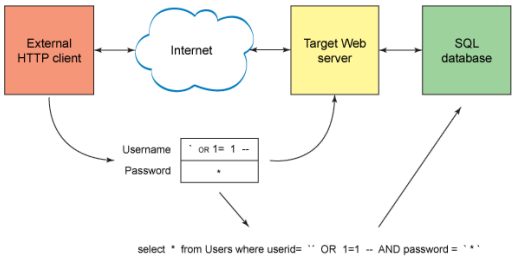
Ảnh 36. Mô tả ID số 6 bị xóa

## **Tấn công sql injction**

### **Tìm hiểu về lỗi SQL injection và cho ví dụ minh họa**

# Khái niệm về SQL injection:

=> Là lỗi do những kẻ tấn công nhằm vào các dữ liệu đầu vào để chèn thêm các câu lệnh vào SQL



Ảnh 37. Mô tả SQL Injection

### **Nguyên nhân thường gặp dễ bị lỗi tấn công của SQL injection**

- Không kiểm tra ký tự thoát truy vấn:

Thiếu mã kiểm tra dữ liệu đầu vào trong câu lệnh SQL

- Lỗi bảo mật bên trong máy chủ của CSDL

- Blind SQL injection: là dạng lỗi tồn tại trong ứng dụng wed gây ra sự sai hoặc khác khi hiển thị nội dung

- Dạng tấn công vượt qua kiểm tra lúc đăng nhập:

Tin tặc sử dụng SQL thao tác với CSDL trên trình duyệt Wed

- Tấn công bằng stored-procedures

Là loại tấn công sẽ gây tác hại rất lớn nếu ứng dụng được thực thi với quyền quản trị hệ thống 'sa'.

- Xử lý không đúng kiểu:

Do lập trình viên định nghĩa dữ liệu đầu vào không rõ ràng thiếu bước kiểm tra và lọc dữ liệu đầu vào.

## **Tấn công Broken Authentication**

- Là nhóm vấn đề có thể xảy ra trong quá trình xác thực:

+ URL có thể chứa Session ID và rò rỉ nó trong Referer Header của người dùng khác

+ Mật khẩu không được mã hóa hoặc dễ giải mã trong khi lưu trữ.

+ Lỗ hổng Session Fixation.

+ Tấn công Session Hijacking có thể xảy ra khi thời gian hét hạn của session không được triển khai đúng hoặc sử dụng HTTP (không bảo mật SSL)…

# Cách khắc phục lỗi:

- để tránh lỗ hổng bảo mật web này nên sử dụng một framework.

- Hãy dự đoán trước một số trường hợp có thể xảy ra với trang đăng nhập của mình