Bài tập thực hành 03

**MSSV: B2203450**

**Họ và tên sinh viên: Mai Văn Khánh**

Lập trình PHP nâng cao

**Mục tiêu cần đạt**:

Sau buổi các bạn cần nắm thao tác sử dụng PHP nâng cao

* Hiểu và biết cách dùng Cookies và Session
* Tạo form đăng nhập và dùng cookies, session để quản lý các đăng nhập, thực hiện chức năng log out (thoát) khỏi hệ thống
* Biết về SQL Injection.
* Upload file, lưu thông tin về tập tin trong CSDL
* Đọc, và xử lý nội dung trong tập tin

Các bạn đọc và thực hiện viết code như hướng dẫn để làm quen, sau đó đọc trả lời các **Yêu cầu** ở dưới bài hướng dẫn để thực hiện yêu cầu bài thực hành.

**Gợi ý Tham khảo:**

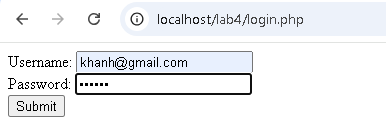
* <https://www.w3schools.com/php/>
* và các nguồn khác mà sinh viên tìm được

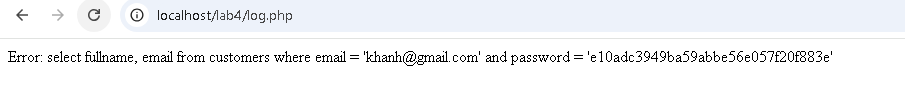
# Yêu cầu bài thực hành:

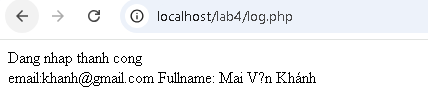
1. Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.

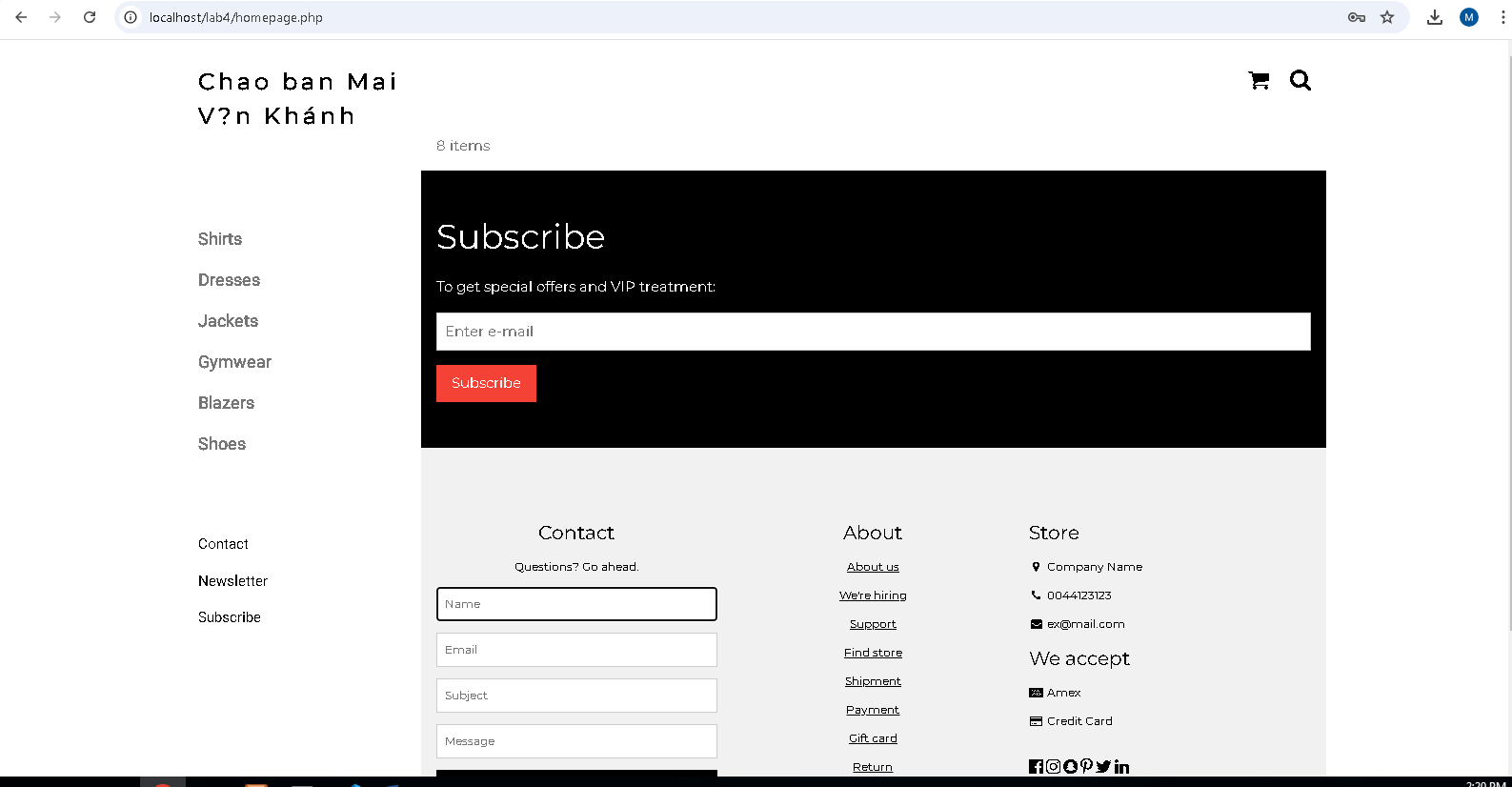


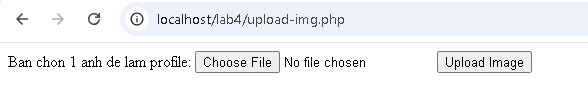


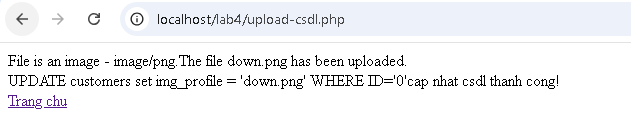


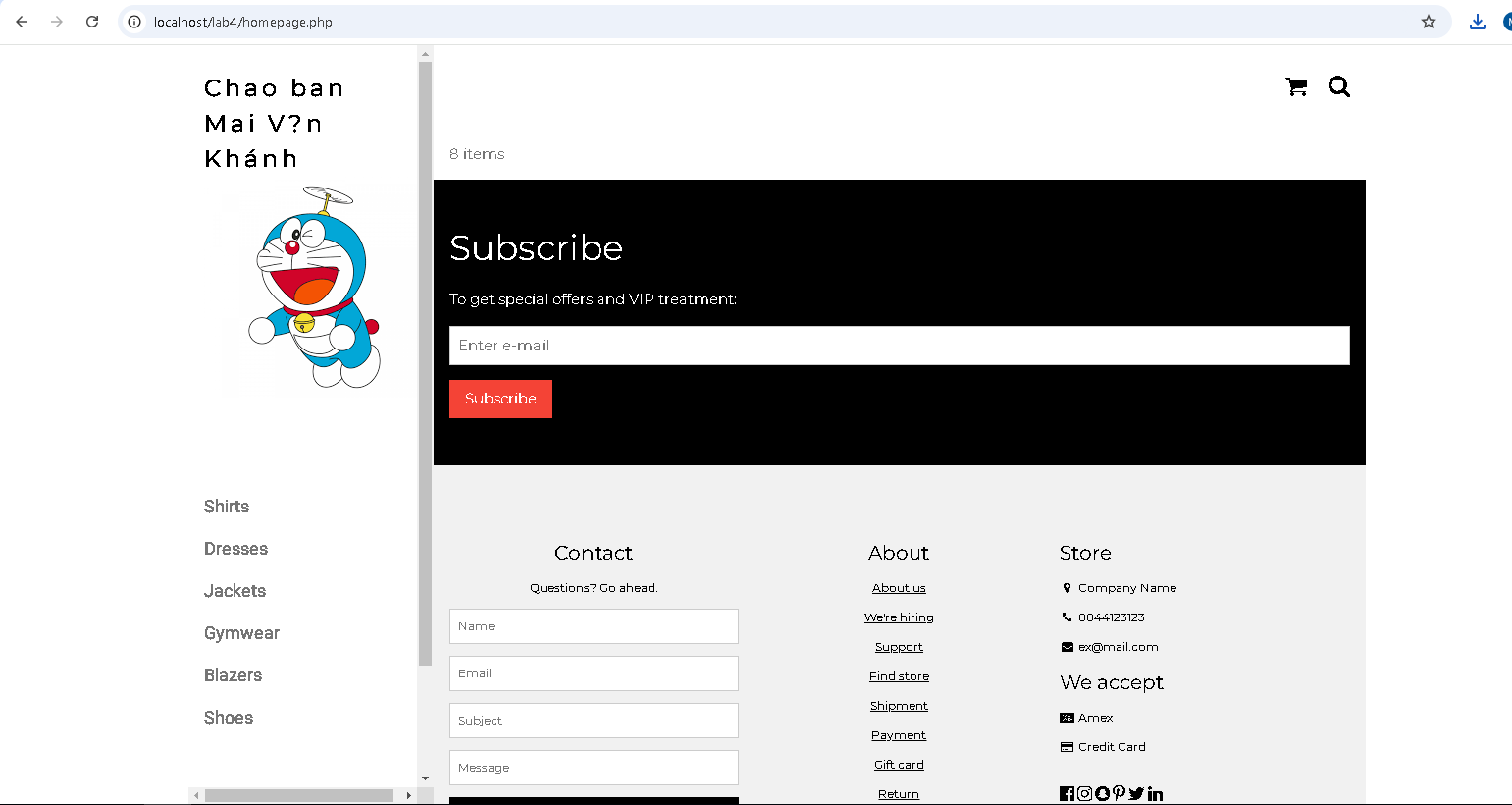


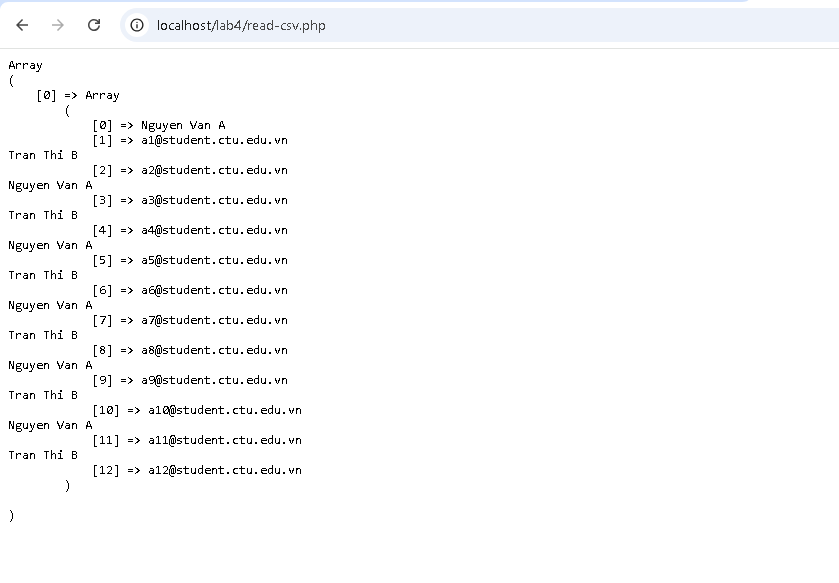












1. Trong vài trường hợp, hacker có thể sử dụng các kỹ thuật SQL Injection để hack hệ thống của bạn. Bạn hãy trình bày SQL Injection là gì và thử các ví dụ trình bày trong [[1]](#footnote-0). Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.
2. Dựa vào link[[2]](#footnote-1), bạn hãy cho biết Cookie là gì, diễn giải ý nghĩa các tham số trong setcookie(). Cách lưu, lấy, xóa giá trị trong cookie.

Cookie là một tệp nhỏ mà máy chủ gửi đến trình duyệt web của người dùng, lưu trữ thông tin để theo dõi các hoạt động của người dùng trên trang web. Cookie thường được sử dụng để ghi nhớ thông tin phiên làm việc, sở thích, hoặc các thông tin xác thực để cải thiện trải nghiệm người dùng.

**Các tham số:**

1. **name**: Tên của cookie (bắt buộc).
2. **value**: Giá trị của cookie (bắt buộc).
3. **expire**: Thời điểm hết hạn của cookie (tính bằng giây kể từ thời điểm hiện tại). Nếu không chỉ định, cookie sẽ hết hạn khi phiên làm việc kết thúc.
4. **path**: Đường dẫn mà cookie có thể sử dụng. Nếu không chỉ định, cookie sẽ chỉ khả dụng trong thư mục hiện tại.
5. **domain**: Tên miền mà cookie có thể truy cập. Nếu không chỉ định, cookie chỉ khả dụng cho tên miền hiện tại.
6. **secure**: Nếu được đặt thành true, cookie chỉ được gửi qua kết nối HTTPS.
7. **httponly**: Nếu được đặt thành true, cookie sẽ không thể truy cập từ JavaScript, giúp bảo vệ chống lại một số loại tấn công.

**Cách lưu, lấy và xóa giá trị trong cookie**

**1. Lưu cookie:**

Để lưu giá trị trong cookie, bạn sử dụng setcookie():

setcookie("username", "JohnDoe", time() + 3600); // Hết hạn sau 1 giờ

**2. Lấy giá trị của cookie:**

Để lấy giá trị của cookie, bạn sử dụng biến $\_COOKIE:

if (isset($\_COOKIE["username"])) {

echo "Username: " . $\_COOKIE["username"];

} else {

echo "Cookie not set!";

}

**3. Xóa cookie:**

Để xóa cookie, bạn chỉ cần gọi setcookie() với thời gian hết hạn trong quá khứ:

setcookie("username", "", time() - 3600); // Xóa cookie

1. Dựa vào link[[3]](#footnote-2), bạn hãy cho biết Session dùng để làm gì. Cách lưu, lấy, xóa giá trị trong Session.

Session là một cơ chế lưu trữ thông tin tạm thời trên máy chủ, giúp theo dõi trạng thái người dùng trong quá trình truy cập trang web. Khác với cookie, thông tin của session không được lưu trên máy khách, mà được lưu trữ trên máy chủ. Session thường được sử dụng để quản lý thông tin người dùng, chẳng hạn như xác thực, giỏ hàng trong thương mại điện tử, hoặc các dữ liệu tạm thời khác.

**1. Lưu giá trị vào session:**

Bạn có thể lưu giá trị vào session bằng cách gán giá trị cho biến $\_SESSION:

$\_SESSION["username"] = "JohnDoe";

**2. Lấy giá trị từ session:**

Để lấy giá trị từ session, bạn sử dụng biến $\_SESSION:

if (isset($\_SESSION["username"])) {

echo "Username: " . $\_SESSION["username"];

} else {

echo "Session not set!";

}

**3. Xóa giá trị trong session:**

Để xóa một giá trị cụ thể trong session, bạn có thể sử dụng unset():

unset($\_SESSION["username"]);

1. Bạn hãy so sánh Cookie và Session.

**1. Định nghĩa**

* **Cookie**: Là một tệp nhỏ lưu trữ trên máy khách (trình duyệt) của người dùng, chứa thông tin mà máy chủ gửi đến. Cookie có thể được truy cập bởi cả máy chủ và trình duyệt.
* **Session**: Là một cơ chế lưu trữ thông tin tạm thời trên máy chủ, theo dõi trạng thái của người dùng trong suốt phiên làm việc. Thông tin session không được lưu trữ trên máy khách.

**2. Kích thước**

* **Cookie**: Giới hạn kích thước thường là khoảng 4 KB cho mỗi cookie.
* **Session**: Kích thước thông tin lưu trữ trong session phụ thuộc vào dung lượng bộ nhớ của máy chủ, không bị giới hạn như cookie.

**3. Thời gian tồn tại**

* **Cookie**: Có thể được thiết lập để hết hạn sau một khoảng thời gian cụ thể hoặc khi trình duyệt đóng lại. Cookie có thể tồn tại lâu dài.
* **Session**: Thường chỉ tồn tại trong suốt phiên làm việc của người dùng. Khi người dùng đóng trình duyệt hoặc session hết hạn (thường là sau một khoảng thời gian không hoạt động), thông tin trong session sẽ bị xóa.

**4. Bảo mật**

* **Cookie**: Dữ liệu cookie có thể bị giả mạo hoặc đọc bởi người dùng, đặc biệt nếu không được mã hóa. Không nên lưu trữ thông tin nhạy cảm trong cookie.
* **Session**: Thông tin lưu trữ trong session an toàn hơn, vì nó được giữ trên máy chủ và chỉ có thể được truy cập qua một ID phiên. Tuy nhiên, vẫn cần phải bảo vệ chống lại các cuộc tấn công như session hijacking.

**5. Sử dụng**

* **Cookie**: Thích hợp cho việc lưu trữ thông tin không nhạy cảm và thông tin có thể dùng lại, như tùy chọn người dùng, lịch sử truy cập, hoặc thông tin đăng nhập (nếu được mã hóa).
* **Session**: Thích hợp cho việc lưu trữ thông tin nhạy cảm hơn, như thông tin người dùng trong quá trình đăng nhập, giỏ hàng, hoặc dữ liệu cần duy trì trong suốt phiên làm việc.

1. Chỉnh sửa các đoạn gán, khởi tạo, lấy giá trị cookie trong log.php và homepage.php thay bằng cách dùng Session. Tạo tập tin thoat.php để xóa các giá trị trong session, cookie thực hiện chức năng log out khỏi hệ thống.
2. Bạn hãy tạo form sua\_mk.php cho phép người dùng chỉnh sửa mật khẩu sau khi đăng nhập. Yêu cầu gồm 3 input: ô để nhập mật khẩu cũ, 1 ô để nhập mật khẩu mới, 1 ô cho phép nhập mật lại mật khẩu mới. Khi nhập xong cần đảm bảo: mật khẩu cũ là khớp với CSDL đang lưu, 2 ô nhập mật khẩu mới phải khớp với nhau và không giống với mật khẩu cũ, nếu đáp ứng điều kiện thì tiến hành băm mật khẩu với md5 và lưu mật khẩu mới vào CSDL.
3. Bạn đọc trong[[4]](#footnote-3) để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.

**Cơ chế Upload File trong PHP**

Upload file trong PHP là một quá trình cho phép người dùng gửi tệp từ máy tính của họ lên máy chủ web. Để thực hiện upload file, bạn cần cấu hình một số yếu tố trong PHP và HTML.

### **1. HTML Form**

Để upload file, bạn cần một biểu mẫu HTML với thuộc tính enctype được đặt là multipart/form-data. Đây là một ví dụ về biểu mẫu upload:

<form action="upload.php" method="post" enctype="multipart/form-data">

Chọn file để upload:

<input type="file" name="fileToUpload" id="fileToUpload">

<input type="submit" value="Upload File" name="submit">

</form>

**2. Xử lý Upload trong PHP**

Khi người dùng gửi biểu mẫu, file sẽ được gửi đến một tệp PHP để xử lý (trong ví dụ trên là upload.php). Dưới đây là các bước cơ bản để xử lý upload file:

**a. Kiểm tra nếu có file được upload**

Bạn có thể kiểm tra xem file có được gửi lên hay không bằng cách kiểm tra biến $\_FILES:

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

if (isset($\_FILES["fileToUpload"])) {

// Tiến hành xử lý upload

}

}

**b. Xử lý thông tin file**

Các thông tin liên quan đến file upload được lưu trữ trong mảng $\_FILES, bao gồm:

* $\_FILES['fileToUpload']['name']: Tên file gốc.
* $\_FILES['fileToUpload']['type']: Loại file (MIME type).
* $\_FILES['fileToUpload']['tmp\_name']: Tên tạm thời của file trên server.
* $\_FILES['fileToUpload']['error']: Mã lỗi (nếu có).
* $\_FILES['fileToUpload']['size']: Kích thước file (tính bằng byte).

**c. Kiểm tra lỗi upload**

Bạn cần kiểm tra mã lỗi để xác định xem file đã được upload thành công hay chưa:

if ($\_FILES["fileToUpload"]["error"] == UPLOAD\_ERR\_OK) {

// Upload thành công, xử lý file

} else {

// Xử lý lỗi

}

**d. Di chuyển file**

Sau khi upload, bạn cần di chuyển file từ thư mục tạm thời đến vị trí mong muốn trên server bằng cách sử dụng hàm move\_uploaded\_file():

$target\_dir = "uploads/"; // Thư mục lưu trữ file

$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);

if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file)) {

echo "File ". htmlspecialchars(basename($\_FILES["fileToUpload"]["name"])) . " đã được upload thành công.";

} else {

echo "Có lỗi xảy ra khi upload file.";

}

**3. Các Hàm và Thủ Tục Khác**

* **is\_uploaded\_file()**: Kiểm tra xem file có phải là file đã được upload hay không.
* **file\_exists()**: Kiểm tra xem file đã tồn tại trong thư mục đích hay chưa.
* **pathinfo()**: Lấy thông tin về đường dẫn file (ví dụ: mở rộng file).

**4. Bảo Mật**

* Kiểm tra loại file trước khi upload để tránh upload các loại file không mong muốn.
* Giới hạn kích thước file upload bằng cách cấu hình trong php.ini (ví dụ: upload\_max\_filesize).
* Sử dụng các biện pháp mã hóa tên file để tránh trùng lặp và bảo vệ thông tin nhạy cảm.

1. Bạn hãy đọc[[5]](#footnote-4), mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).

Hàm file() trong PHP được sử dụng để đọc nội dung của một tệp và trả về một mảng, trong đó mỗi phần tử là một dòng trong tệp. Đây là một cách đơn giản và nhanh chóng để xử lý tệp văn bản.

**Tham số**

1. **$filename**:
   * **Loại**: string
   * **Mô tả**: Đường dẫn đến tệp bạn muốn đọc.
2. **$flags** (Tùy chọn):
   * **Loại**: int
   * **Mô tả**: Các cờ để điều chỉnh hành vi của hàm. Một số cờ thường dùng là:
     + FILE\_USE\_INCLUDE\_PATH: Tìm tệp trong các thư mục đã được định nghĩa trong include\_path.
     + FILE\_IGNORE\_NEW\_LINES: Bỏ qua ký tự xuống dòng (newline) ở cuối mỗi dòng.
     + FILE\_SKIP\_EMPTY\_LINES: Bỏ qua các dòng trống.
3. **$context** (Tùy chọn):
   * **Loại**: resource
   * **Mô tả**: Ngữ cảnh của tệp, cho phép bạn điều chỉnh cách đọc tệp, chẳng hạn như thiết lập thời gian chờ hoặc thông tin xác thực.

**Ví dụ Sử Dụng**

<?php

$lines = file('example.txt'); // Đọc tệp và lưu vào mảng

foreach ($lines as $line) {

echo $line . "<br>"; // Hiển thị mỗi dòng

}

?>

1. Bạn hãy tạo 1 tập tin csv ít nhất 10 dòng dữ liệu với các cột dữ liệu như bảng customer trong csdl qlbanhang. Tạo tập tin upload-csv.php với giao diện cho phép upload các tập tin csv, và tập tin upload-csv-processing để xử lý nút xử lý sự kiện upload file csv và đưa dữ liệu vào bảng customers trong CSDL. Gợi ý:
   * Xem cấu trúc của bảng customer, mở excel và nhập liệu lưu lại với định dạng CSV.
   * Tham khảo upload-img.php để thiết kế giao diện upload file
   * Tham khảo upload-csdl.php để thiết kế action xử lý việc upload file, chú ý chỉnh sửa loại tập tin chấp nhận các file csv. Tham khảo: [[6]](#footnote-5),[[7]](#footnote-6). Ở đoạn sau khi upload thành công, bạn lấy tên file vừa upload đưa vào hàm đọc tập tin. Bạn tham khảo read-csv.php chỉnh sửa đọc dữ liệu từ csv đưa vào mảng để lần lượt thực hiện insert từng dòng dữ liệu trong csv vào CSDL.

1. <https://www.w3schools.com/sql/sql_injection.asp> [↑](#footnote-ref-0)
2. <https://www.w3schools.com/php/php_cookies.asp> [↑](#footnote-ref-1)
3. [↑](#footnote-ref-2)
4. <https://www.w3schools.com/php/php_file_upload.asp> [↑](#footnote-ref-3)
5. <https://www.w3schools.com/php/func_filesystem_file.asp> [↑](#footnote-ref-4)
6. <https://www.php.net/manual/en/features.file-upload.post-method.php>, [↑](#footnote-ref-5)
7. [↑](#footnote-ref-6)