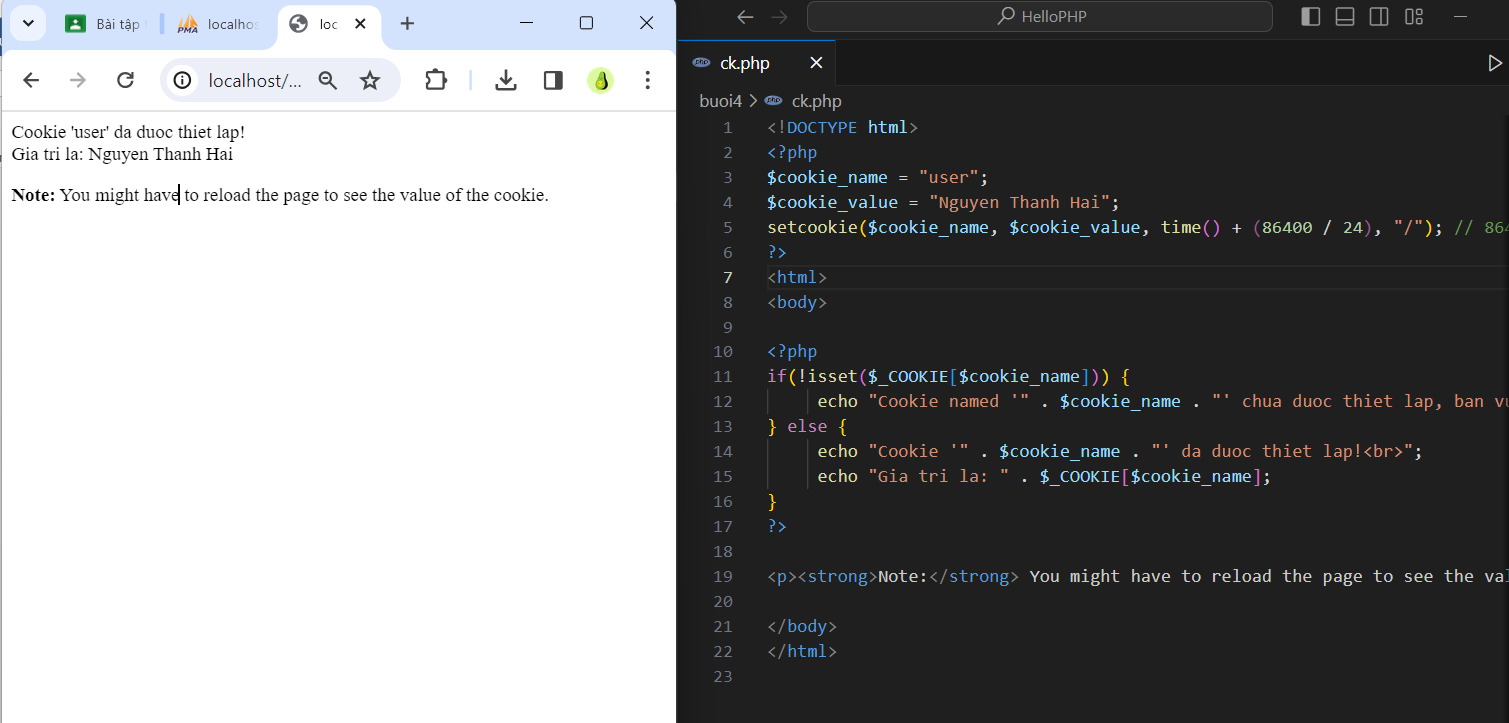
**PHÁT TRIỂN HỆ THỐNG WEB**

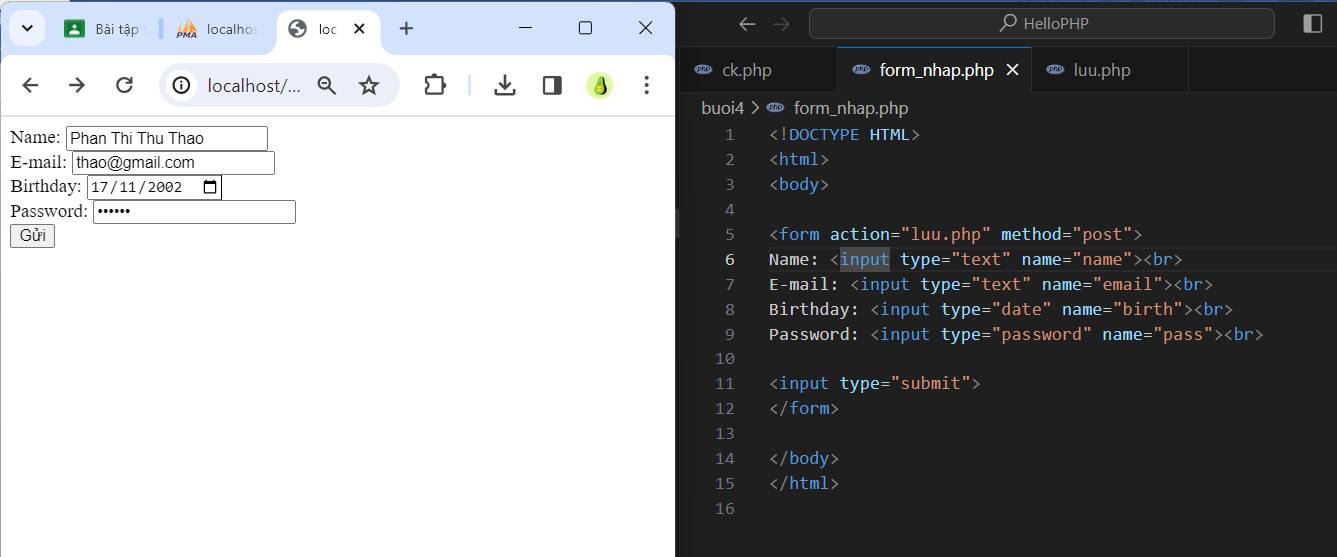
**Thực hành buổi 4**

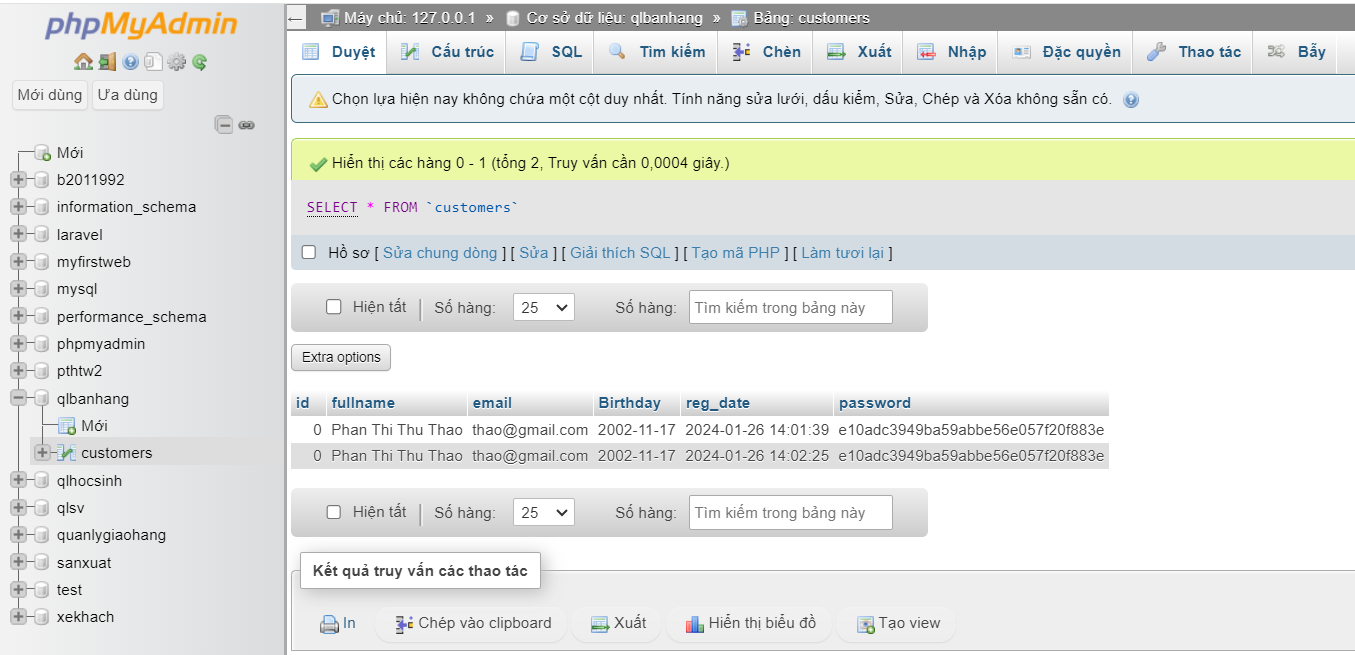
1. Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.

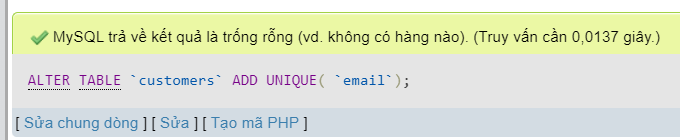
* Tạo Cookies



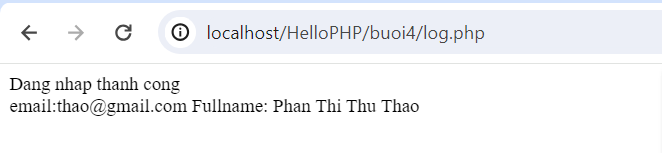
* Tạo một trang đăng ký người dùng với mật khẩu được mã hóa



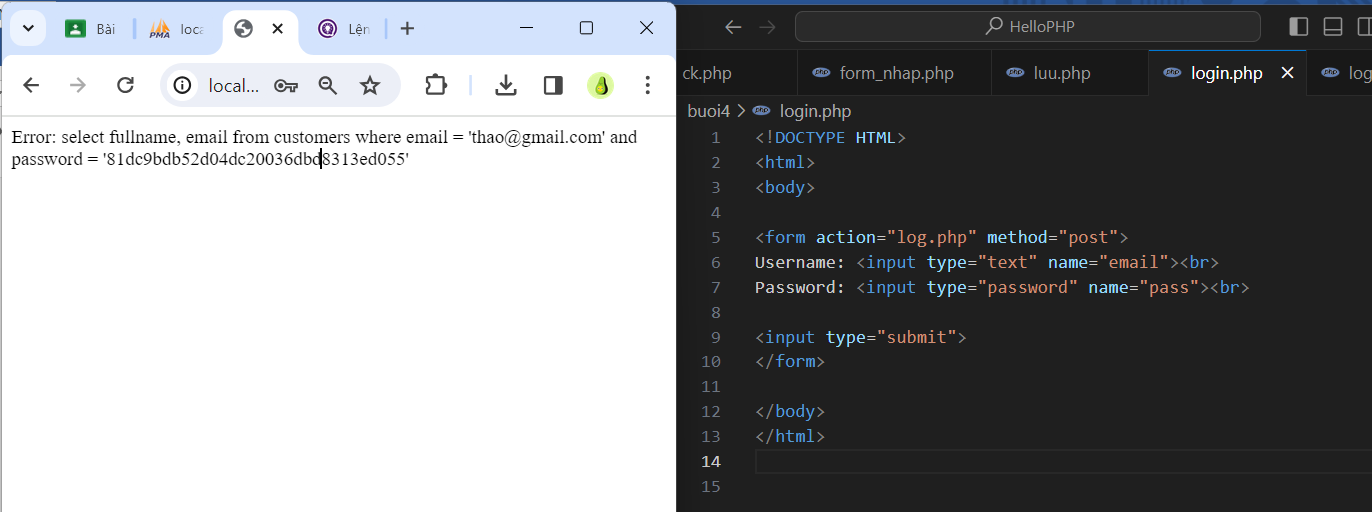




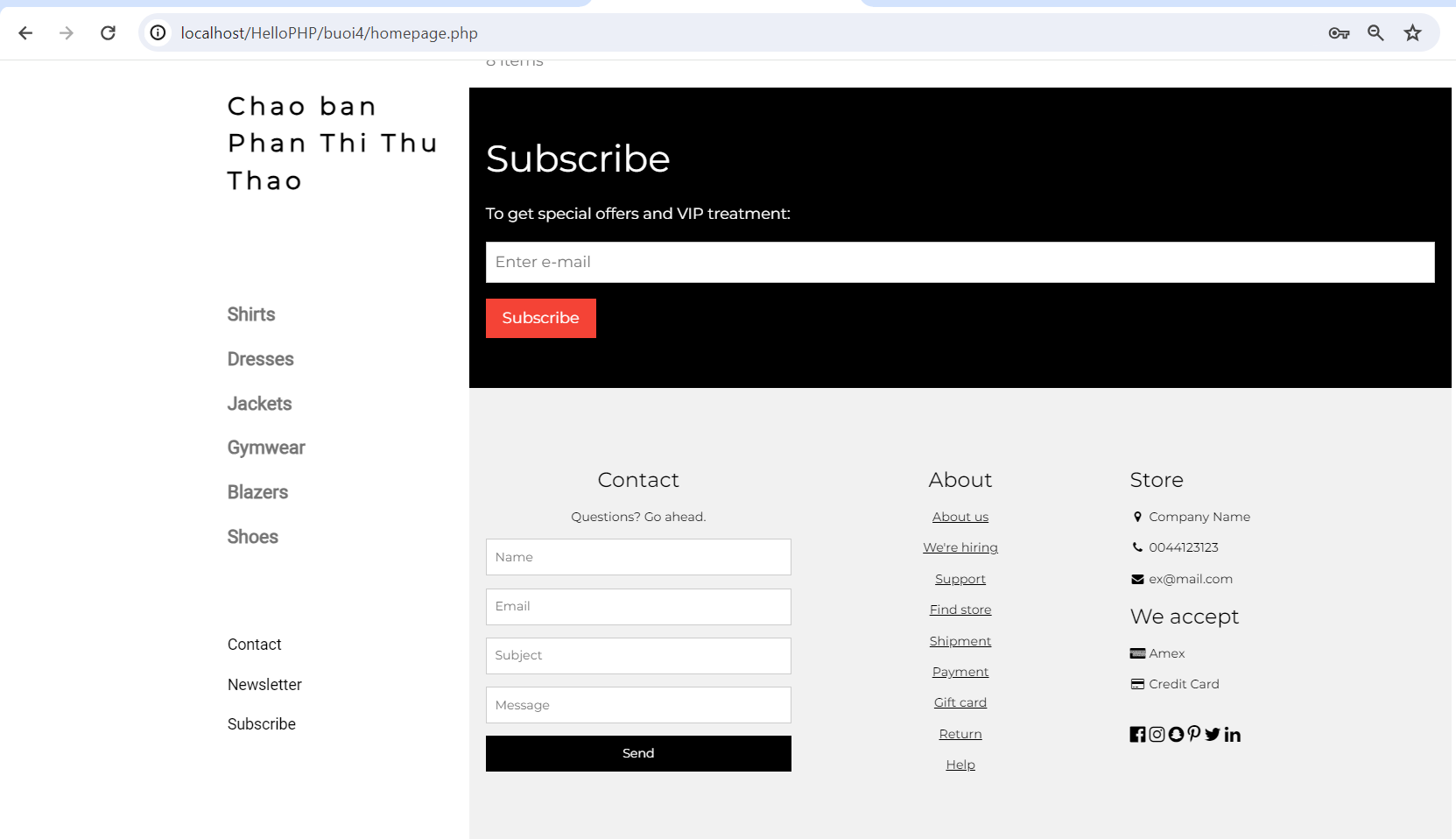
* Tạo Form đăng nhập



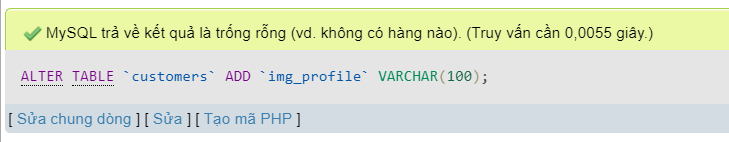
* Khi nhập sai

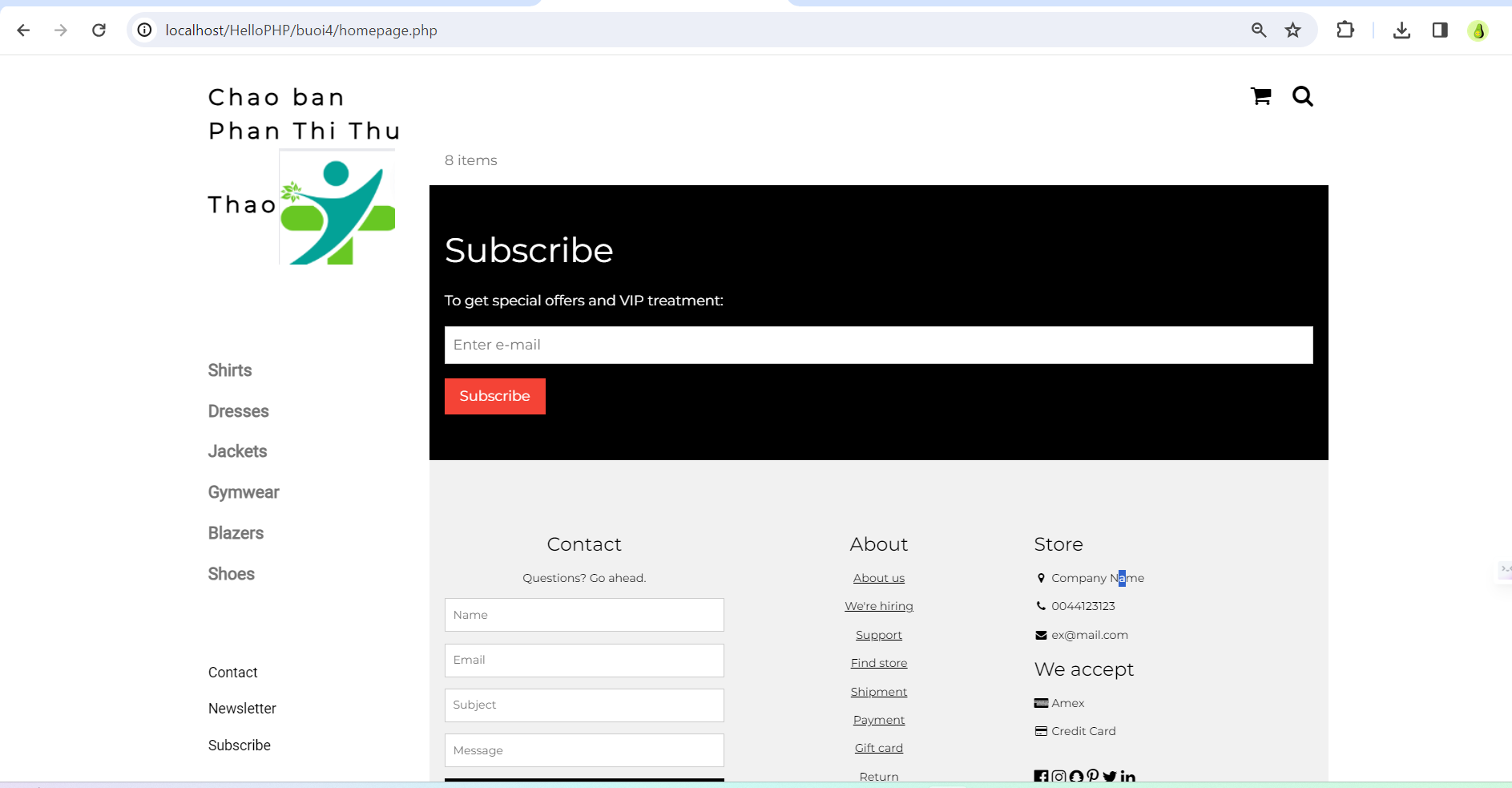


* Dùng Cookies lưu lại tên đăng nhập, email và điều hướng

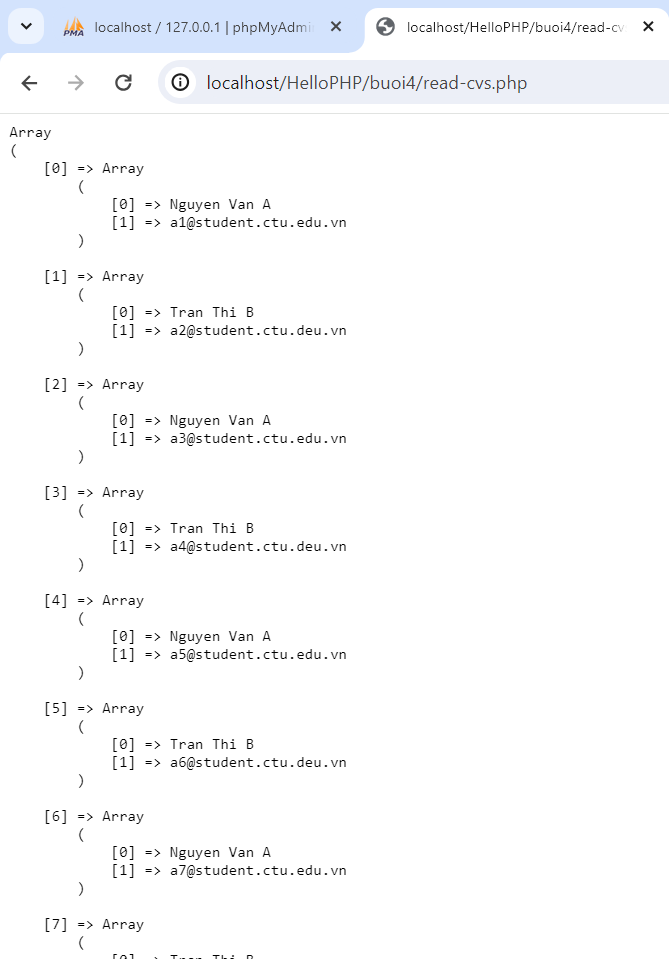


* Upload tập tin



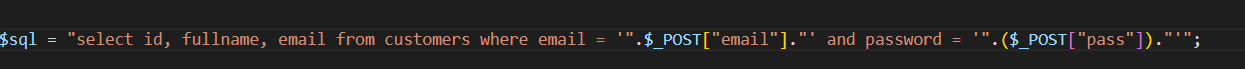


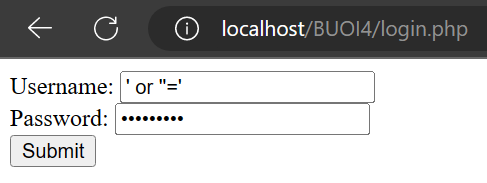
* Upload file csv và đọc dữ liệu vào mảng



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-1). Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.

* SQL Injection là:
* Một kỹ thuật chèn mã có thể phá hủy cơ sở dữ liệu của bạn
* Một trong những kỹ thuất hack web phổ biến nhất
* Việc đặt mã độc vào lệnh SQL thông qua đầu vào trang web
* Ứng dụng vào trang đăng nhập





* Nguồn tham khảo:

<https://www.w3schools.com/sql/sql_injection.asp>

1. Dựa vào link[[2]](#footnote-2), 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 thường được sử dụng để nhận dạng người dùng; là một tệp nhỏ mà máy chủ nhúng vào máy tính của người dùng. Mỗi lần máy tính đó yêu cầu một trang bằng trình duyệt, nó cũng sẽ gửi cookie. Với PHP, bạn có thể vừa tạo vừa truy xuất các giá trị cookie
* Tạo cookie với PHP bằng hàm:



* Hàm setcookie(): xác định một cookie sẽ được gửi cùng với phần còn lại của tiên đề HTTP và phải xuất hiện trước thẻ <html>
* Giá trị của cookie được tự động mã hóa URL khi gửi cookie và tự động giải mã khi nhận được (để ngăn chặn mã háo URL, thay vào đó hãy sử dụng setrawcookie() )

|  |  |
| --- | --- |
| Tham số | Mô tả |
| name (tên) | Yêu cầu, chỉ định tên của cookie |
| value (giá trị) | Không bắt buộc, chỉ định giá trị của cookie |
| Expire (hết hạn) | Không bắt buộc, chỉ định thời điểm cookie hết hạn. Gái trị: time()+86400\*30, sẽ đặt cookie hết hạn sau 30 ngày. Nếu tham số này bị bỉ qua hoặc được đặt thành 0, cookie sẽ hết hạn vào cuối phiên (khi đóng trình duyệt). Mặc định là 0. |
| path (đường dẫn) | Tùy chọn. chỉ định đường dẫn máy chủ của cookie. Nếu được đặt thành ”/”, cookie sẽ khả dụng trong toàn bộ miền. Nếu được đặt thành “/php/”, cookie sẽ chỉ khả dụng trong thưu mục php và tất cả các thư mục con của php. Giá trị mặc định là thư mục hiện tại mà cookie đang được đặt |
| domain (tên miền) | Tùy chọn. Chỉ định tên miền của cookie. Để cung cấp cookie trên tất cả các tên miền phụ của example.com, hãy đặt tên miền thành “example.com”. Đặt nó thành [www.example.com](http://www.example.com) sẽ làm cho cookie chỉ khả dụng trong tên miền phụ www |
| secure (an toàn) | Tùy chọn. Chỉ định xem cookie chỉ được truyền qua kết nối HTTPS an toàn hay không. TRUE chỉ ra rằng cookie sẽ chỉ được đặt nếu tồn tại kết nối an toàn. Mặc định là FALSE |
| httponly | Tùy chọn. Nếu được đặt thành TRUE, cookie sẽ chỉ có thể truy cập được thông báo qua giao thức HTTP (cookie sẽ không thể truy cập được bằng ngôn ngữ tập lệnh). Cài đặt này có thể giúp giảm hành vi trộm cắp danh tính thông qua các cuộc tấn XSS. Mặc định là FALSE |

* Cách lưu, lấy, xóa giá trị trong cookie.
* Lưu, lấy giá trị cookie: Ví dụ sau tạo cookie có tên "người dùng" với giá trị "John Doe". Cookie sẽ hết hạn sau 30 ngày (86400 \* 30). "/" có nghĩa là cookie có sẵn trên toàn bộ trang web (nếu không, hãy chọn thư mục bạn thích).



* Xóa cookie: sử dụng setcookie() hàm có ngày hết hạn trong quá khứ

VD:



* Nguồn tham khảo:

<https://xuanthulab.net/su-dung-session-va-cookie-trong-php.html#luu-sookie>

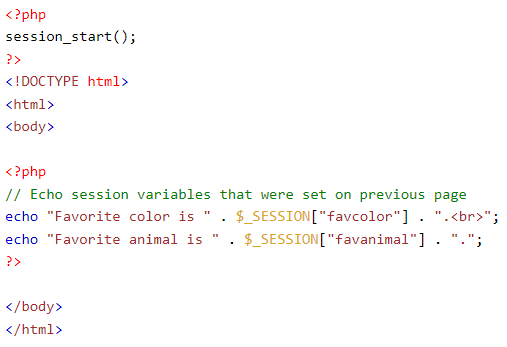
<https://www.w3schools.com/php/php_cookies.asp>

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

* Phiên là một cách lưu trữ thông tin (theo biến) để sử dụng trên nhiều trang
* Không giống như cookie, thông tin không được lưu trữ trên máy tính của người dùng.
* Khi bạn làm việc với một ứng dụng, bạn mở nó, thực hiện một số thay đổi rồi đóng nó lại. Điều này giống như một Phiên. Máy tính biết bạn là ai. Nó biết khi nào bạn khởi động ứng dụng và khi nào bạn kết thúc. Nhưng trên internet có một vấn đề: máy chủ web không biết bạn là ai hoặc bạn làm gì, vì địa chỉ HTTP không duy trì trạng thái.
* Biến phiên giải quyết vấn đề này bằng cách lưu trữ thông tin người dùng sẽ được sử dụng trên nhiều trang (ví dụ: tên người dùng, màu yêu thích, v.v.). Theo mặc định, các biến phiên kéo dài cho đến khi người dùng đóng trình duyệt.
* Vì thế; Các biến phiên chứa thông tin về một người dùng và có sẵn cho tất cả các trang trong một ứng dụng.
* Cách lưu, lấy, xóa giá trị trong Session.
* Một phiên được bắt đầu với session\_start()chức năng.
* Các biến phiên được đặt bằng biến toàn cục PHP: $\_SESSION.
* Bây giờ, hãy tạo một trang mới có tên "demo\_session1.php". Trong trang này, chúng tôi bắt đầu một phiên PHP mới và đặt một số biến phiên:



* Tiếp theo, chúng tôi tạo một trang khác có tên "demo\_session2.php". Từ trang này, chúng tôi sẽ truy cập thông tin phiên mà chúng tôi đã đặt trên trang đầu tiên ("demo\_session1.php").
* Lưu ý rằng các biến phiên không được chuyển riêng lẻ đến từng trang mới, thay vào đó chúng được truy xuất từ ​​phiên chúng ta mở ở đầu mỗi trang ( session\_start()).
* Cũng lưu ý rằng tất cả các giá trị biến phiên được lưu trữ trong biến $\_SESSION toàn cục:



* Để xóa tất cả các biến phiên toàn cục và hủy phiên, hãy sử dụng session\_unset() and session\_destroy():



* Nguồn tham khảo:

<https://www.w3schools.com/php/php_sessions.asp>

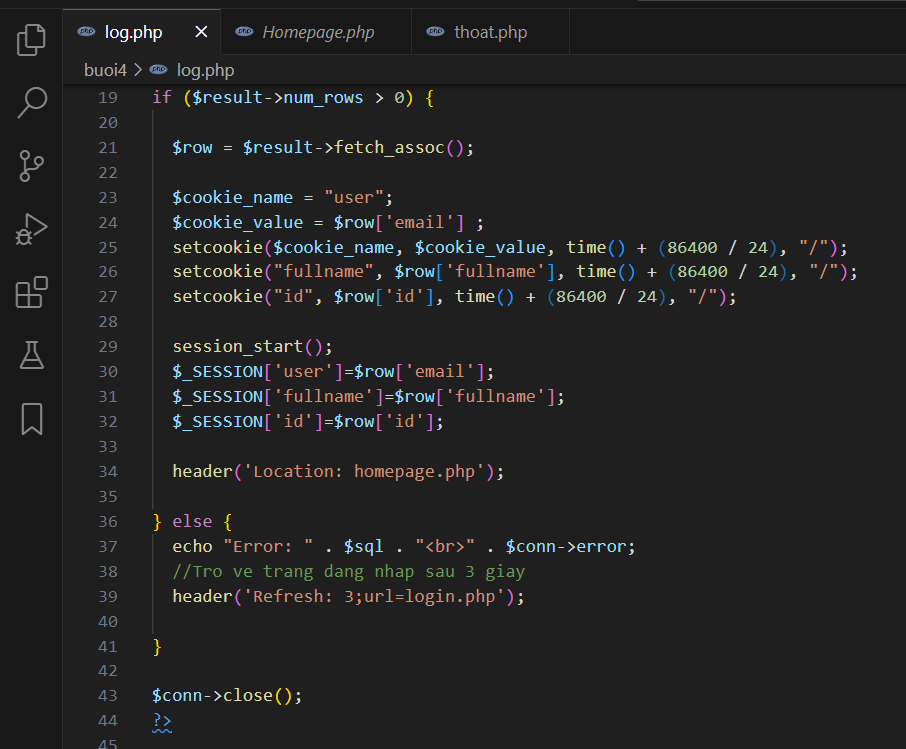
1. Bạn hãy so sánh Cookie và Session.

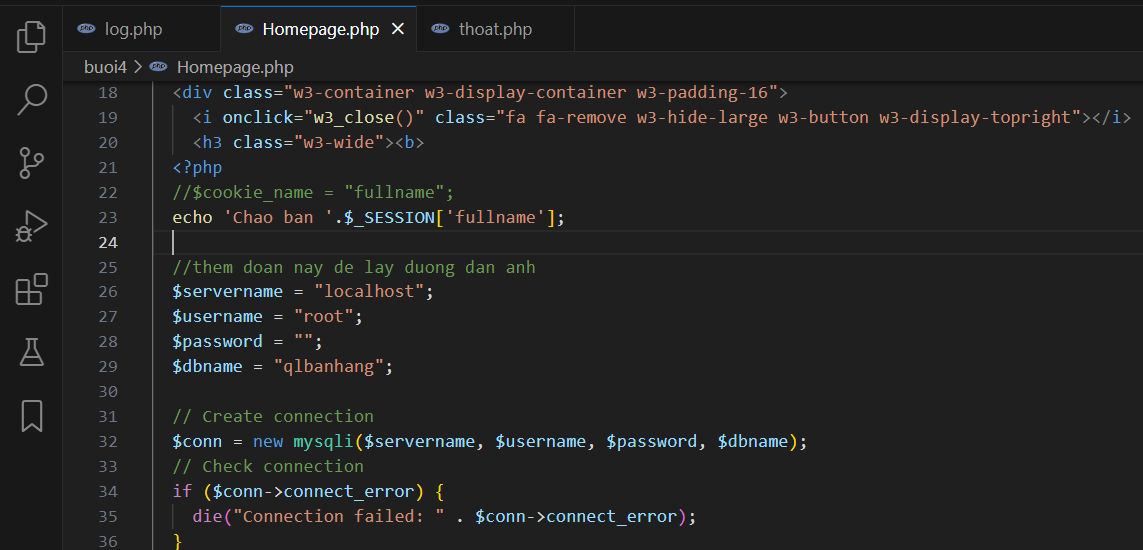
|  |  |
| --- | --- |
| **COOKIE** | **SESSION** |
| Được lưu trữ trên trình duyệt của người dùng | Không được lưu trữ trên trình duyệt |
| Dữ liệu: được lưu ở phía client (máy khách), dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client, có sẵn trong trình duyệt đến khi expired | Dữ liệu: được lưu ở server (máy chủ), không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ nên sẽ bảo mật tốt hơn cookie |
| Kết thúc tùy vào thời gian Developer cài đặt | Một session sẽ kết thúc khi bị đóng trình duyệt |
| Không bị phụ thuộc vào session | Session phụ thuộc vào Cookie (SessionID được lưu ở cookie) |
| Dễ bị tấn công do lưu ở phía client | Dữ liệu session không dễ bị sửa đổi, tấn công |
|  | Sau khi đóng trình duyệt sẽ hết phiên làm việc (session |

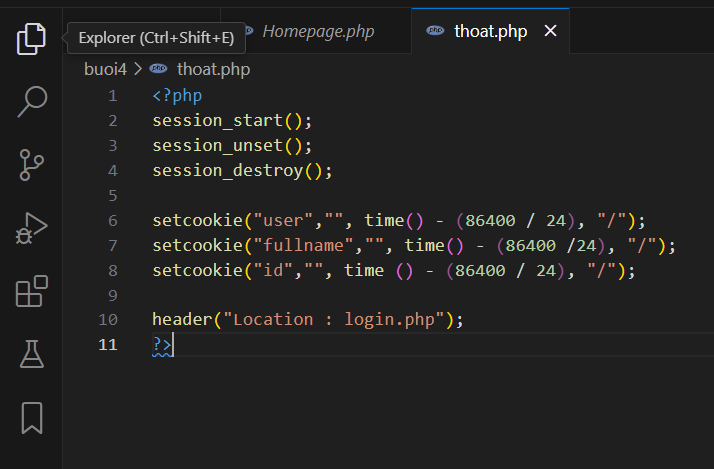
* Nguồn tham khảo:

<https://hocjavascript.net/tong-quan/so-sanh-session-va-cookie/#So_sanh_Session_va_Cookie>

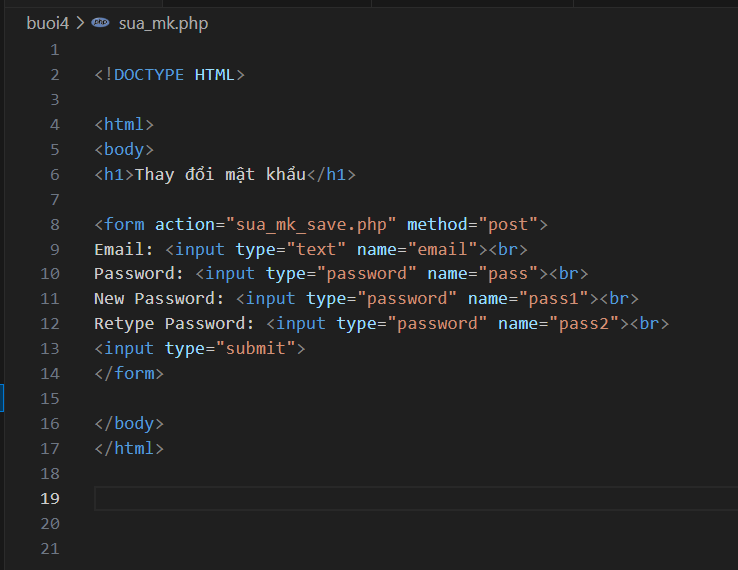
1. Chỉnh sửa các đoạn gán, khởi tạo, lấy giá trị cookie trong log.php và homepage.php thay bằng cách dùng Session. Tạo tập tin thoat.php để xóa các giá trị trong session, cookie thực hiện chức năng log out khỏi hệ thống.



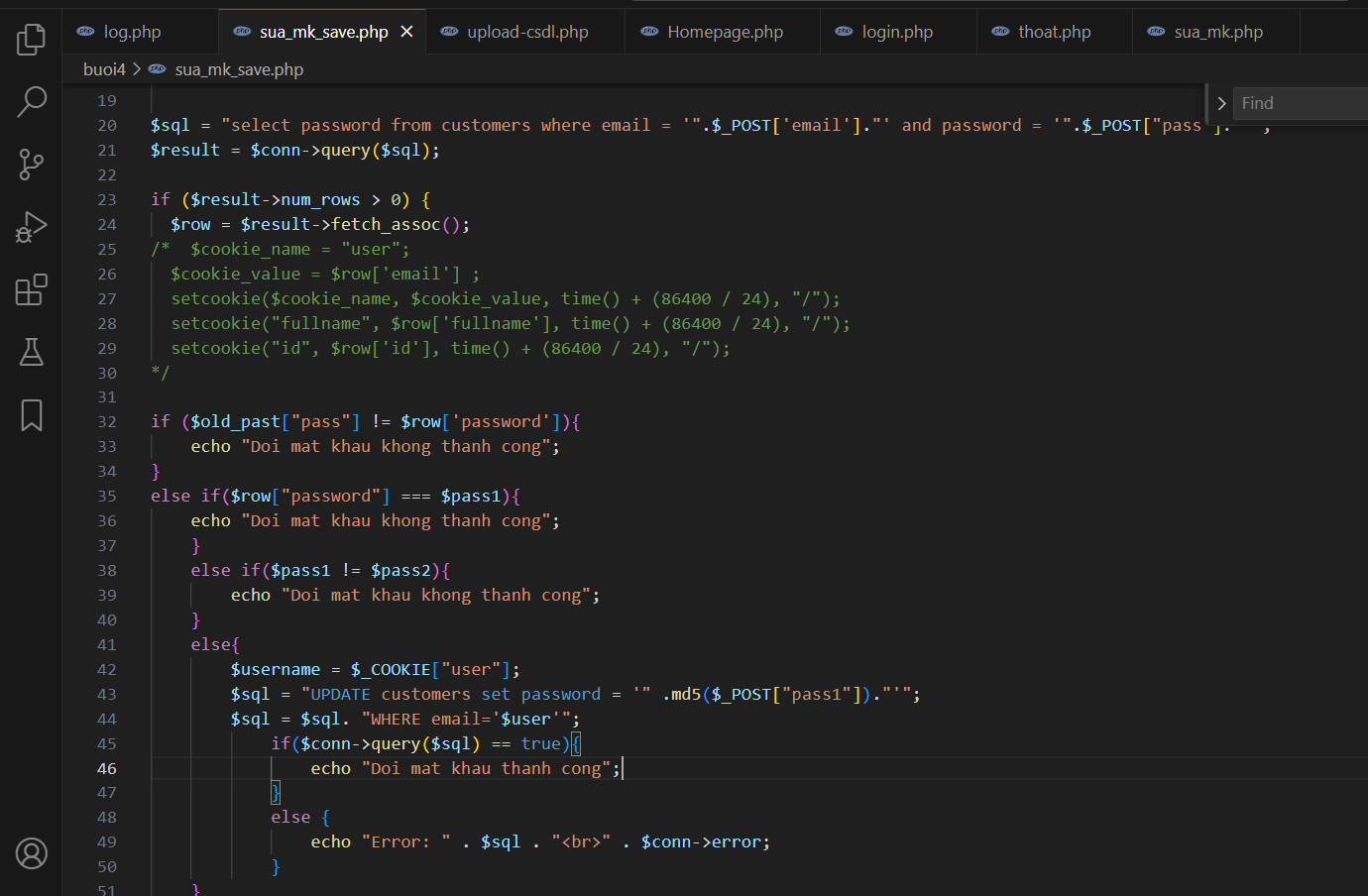




1. Bạn hãy tạo form sua\_mk.php cho phép người dùng chỉnh sửa mật khẩu sau khi đăng nhập. Yêu cầu gồm 3 input: ô để nhập mật khẩu cũ, 1 ô để nhập mật khẩu mới, 1 ô cho phép nhập mật lại mật khẩu mới. Khi nhập xong cần đảm bảo: mật khẩu cũ là khớp với CSDL đang lưu, 2 ô nhập mật khẩu mới phải khớp với 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.







1. Bạn đọc trong[[4]](#footnote-4) để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.

## **Định cấu hình tệp "php.ini"**

* Trước tiên, hãy đảm bảo rằng PHP được cấu hình để cho phép tải tệp lên.
* Trong tệp "php.ini" của bạn, tìm kiếm lệnh file\_uploadsvà đặt thành Bật:



## **Tạo biểu mẫu HTML**

* Tiếp theo, tạo một biểu mẫu HTML cho phép người dùng chọn tệp hình ảnh họ muốn tải lên:



Một số quy tắc cần tuân theo đối với biểu mẫu HTML ở trên:

* Đảm bảo rằng biểu mẫu sử dụng phương thức = "post"
* Biểu mẫu cũng cần thuộc tính sau: enctype="multipart/form-data". Nó chỉ định loại nội dung nào sẽ sử dụng khi gửi biểu mẫu

Nếu không có các yêu cầu trên thì việc tải file lên sẽ không thành công.

Những điều khác cần chú ý:

* Thuộc tính type="file" của thẻ <input> hiển thị trường đầu vào dưới dạng điều khiển chọn tệp, với nút "Duyệt" bên cạnh điều khiển đầu vào

Biểu mẫu ở trên gửi dữ liệu đến một tệp có tên "upload.php", tệp này chúng ta sẽ tạo tiếp theo.

## **\*Tạo tập lệnh PHP tệp tải lên**

\*Tệp "upload.php" chứa mã để tải tệp lên:



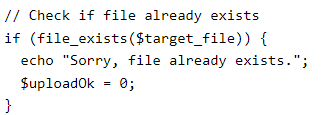
Tập lệnh PHP đã giải thích:

* $target\_dir = "uploads/" - chỉ định thư mục nơi tệp sẽ được đặt
* $target\_file chỉ định đường dẫn của tệp sẽ được tải lên
* $uploadOk=1 chưa được sử dụng (sẽ được sử dụng sau)
* $imageFileType chứa phần mở rộng tệp của tệp (viết thường)
* Tiếp theo kiểm tra file ảnh là ảnh thật hay ảnh giả

**Lưu ý:** Bạn sẽ cần tạo một thư mục mới có tên "uploads" trong thư mục chứa tệp "upload.php". Các tập tin được tải lên sẽ được lưu ở đó.

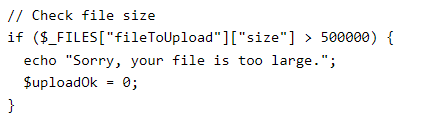
## **Kiểm tra xem tệp đã tồn tại chưa**

* Bây giờ chúng ta có thể thêm một số hạn chế.
* Đầu tiên, chúng tôi sẽ kiểm tra xem tệp đã tồn tại trong thư mục "tải lên" hay chưa. Nếu đúng như vậy, một thông báo lỗi sẽ hiển thị và $uploadOk được đặt thành 0:



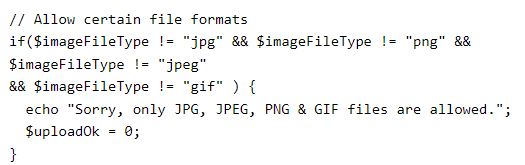
## **Giới hạn kích thước tệp**

* Trường nhập tệp trong biểu mẫu HTML của chúng tôi ở trên có tên là "fileToUpload".
* Bây giờ, chúng tôi muốn kiểm tra kích thước của tập tin. Nếu tệp lớn hơn 500KB, thông báo lỗi sẽ hiển thị và $uploadOk được đặt thành 0:



## **Giới hạn loại tệp**

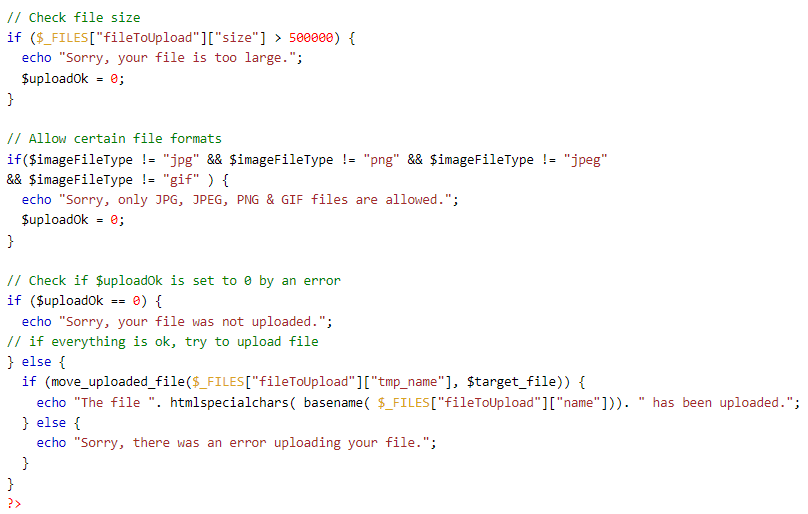
* Mã bên dưới chỉ cho phép người dùng tải lên các tệp JPG, JPEG, PNG và GIF. Tất cả các loại tệp khác đều đưa ra thông báo lỗi trước khi đặt $uploadOk thành 0:



## **Hoàn tất tải lên tập lệnh PHP**

* Tệp "upload.php" hoàn chỉnh bây giờ trông như thế này:





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

## **Định nghĩa và cách sử dụng**

* file() đọc một tập tin vào một mảng.
* Mỗi phần tử mảng chứa một dòng từ tệp, với ký tự dòng mới vẫn được đính kèm.

## **Cú pháp**



* Giá trị tham số

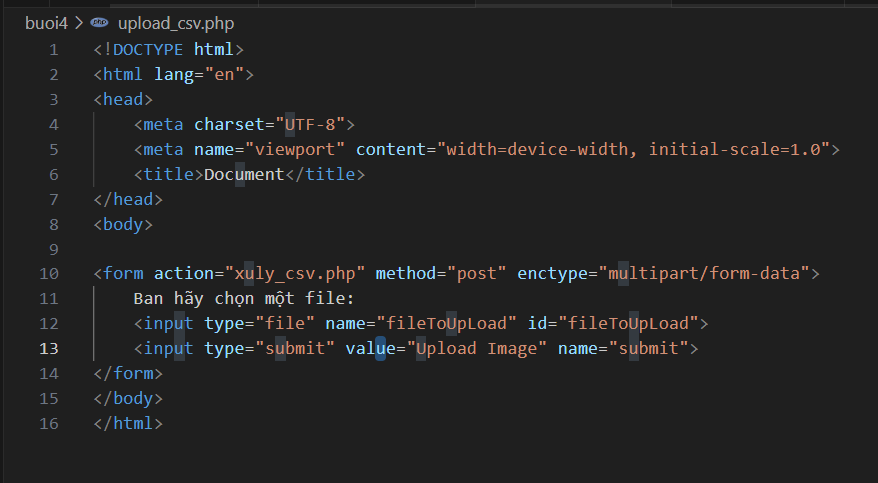
|  |  |
| --- | --- |
| **Tham số** | **Mô tả** |
| Filename (tên tập tin) | Bắt buộc, chỉ định đường dẫn tới file cần đọc |
| Flag (cờ) | Tùy chọn, có thể là một hoặc nhiều hằng số sau:   * FILE\_USE\_INCLUDE\_PATH – Tìm kiếm tệp trong include\_path (trong php.ini) * FILE\_IGNORE\_NEW\_LINES – Bỏ qua dòn mới ở cuối mỗi phần từ mảng * FILE\_SKIP\_EMPTY\_LINES – Bỏ qua các dòng trống trong tệp |
| Context (ngữ cảnh) | Tùy chọn, chỉ định ngữ cảnh của việc xử lý tập tin. Ngữ cảnh là một tập hợp các tùy chọn có thể sửa đổi hành vi của luồng. Có thể bỏ qua bằng cách sử dụng NULL |

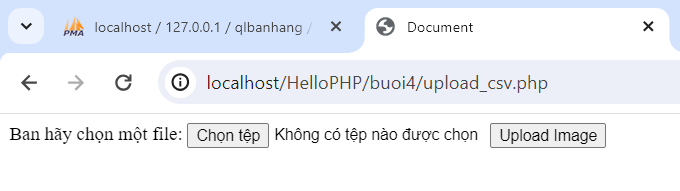
* Nguồn tham khảo:

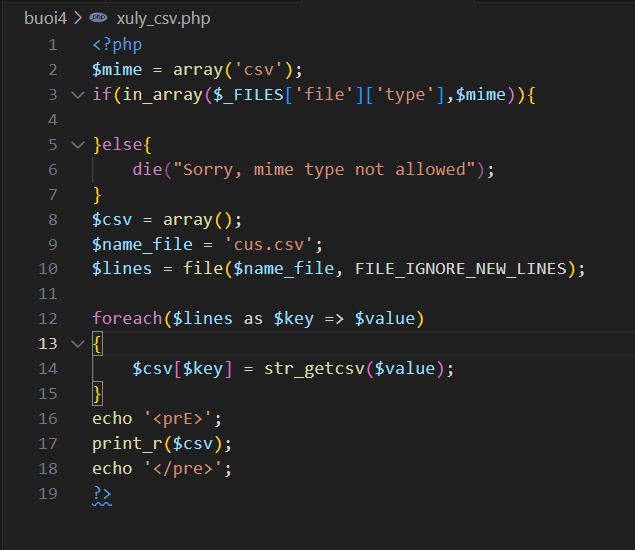
<https://www.w3schools.com/php/func_filesystem_file.asp>

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

* 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-6),[[7]](#footnote-7). 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.







*--Hết--*

1. <https://www.w3schools.com/sql/sql_injection.asp> [↑](#footnote-ref-1)
2. <https://www.w3schools.com/php/php_cookies.asp> [↑](#footnote-ref-2)
3. <https://www.w3schools.com/php/php_sessions.asp> [↑](#footnote-ref-3)
4. <https://www.w3schools.com/php/php_file_upload.asp> [↑](#footnote-ref-4)
5. <https://www.w3schools.com/php/func_filesystem_file.asp> [↑](#footnote-ref-5)
6. <https://www.php.net/manual/en/features.file-upload.post-method.php>, [↑](#footnote-ref-6)
7. https://stackoverflow.com/questions/6654351/check-file-uploaded-is-in-csv-format [↑](#footnote-ref-7)