**Bài tập thực hành 04**

**Lập trình PHP nâng cao**

Yêu cầu bài thực hành:

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

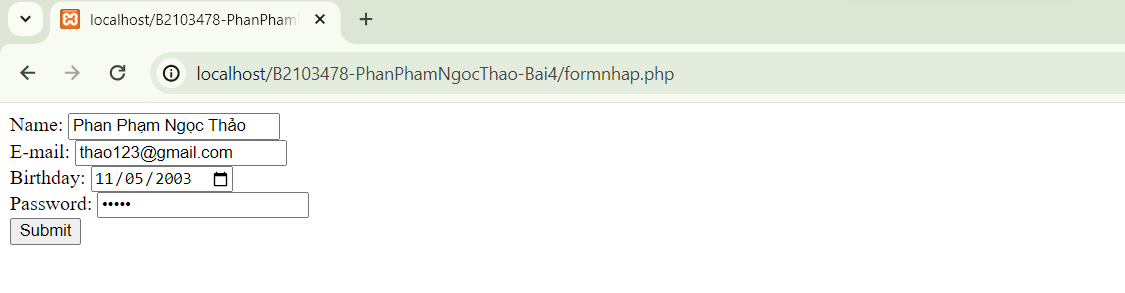
ck.php



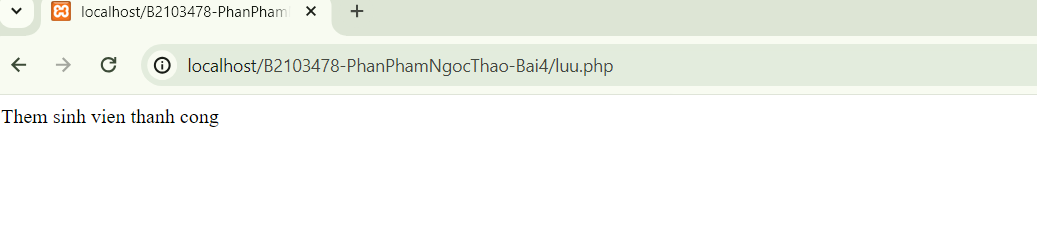
Tạo csdl qlbanhang, với bảng customers

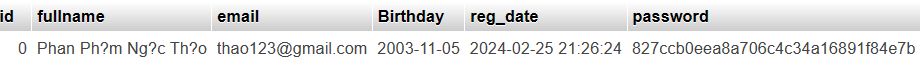


formnhap.php



luu.php

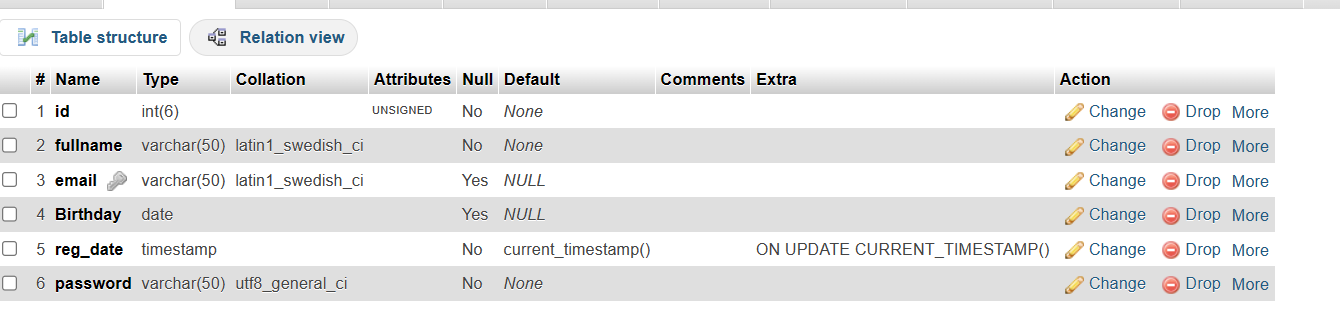




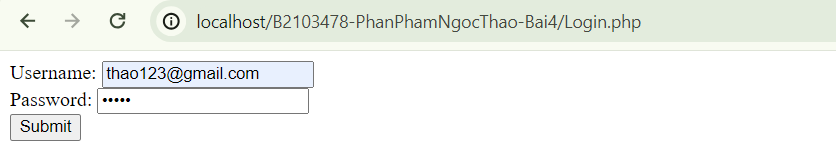
Trong trường hợp chúng ta muốn dùng email là username để đăng nhập ta nên đảm bảo 2

student không có email trùng nhau. Để đảm bảo điều này ta có thể chọn email là unique key (vì ta đã chọn id la primary key, và 1 bảng không thể có 2 khóa chính)

ALTER TABLE `customers` ADD UNIQUE( `email`);

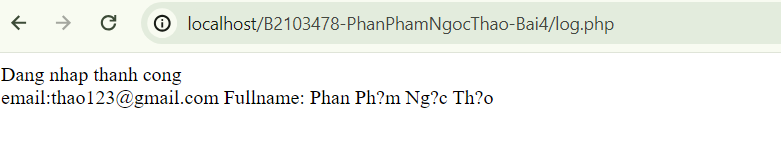


Login.php

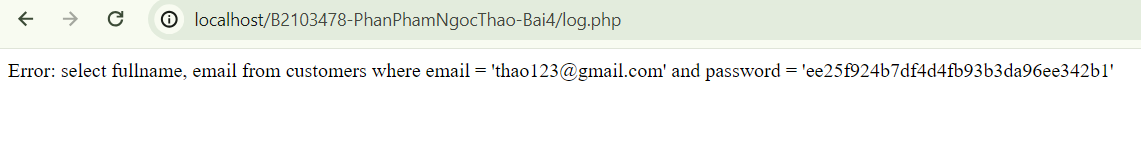


Log.php

- Đăng nhập thành công



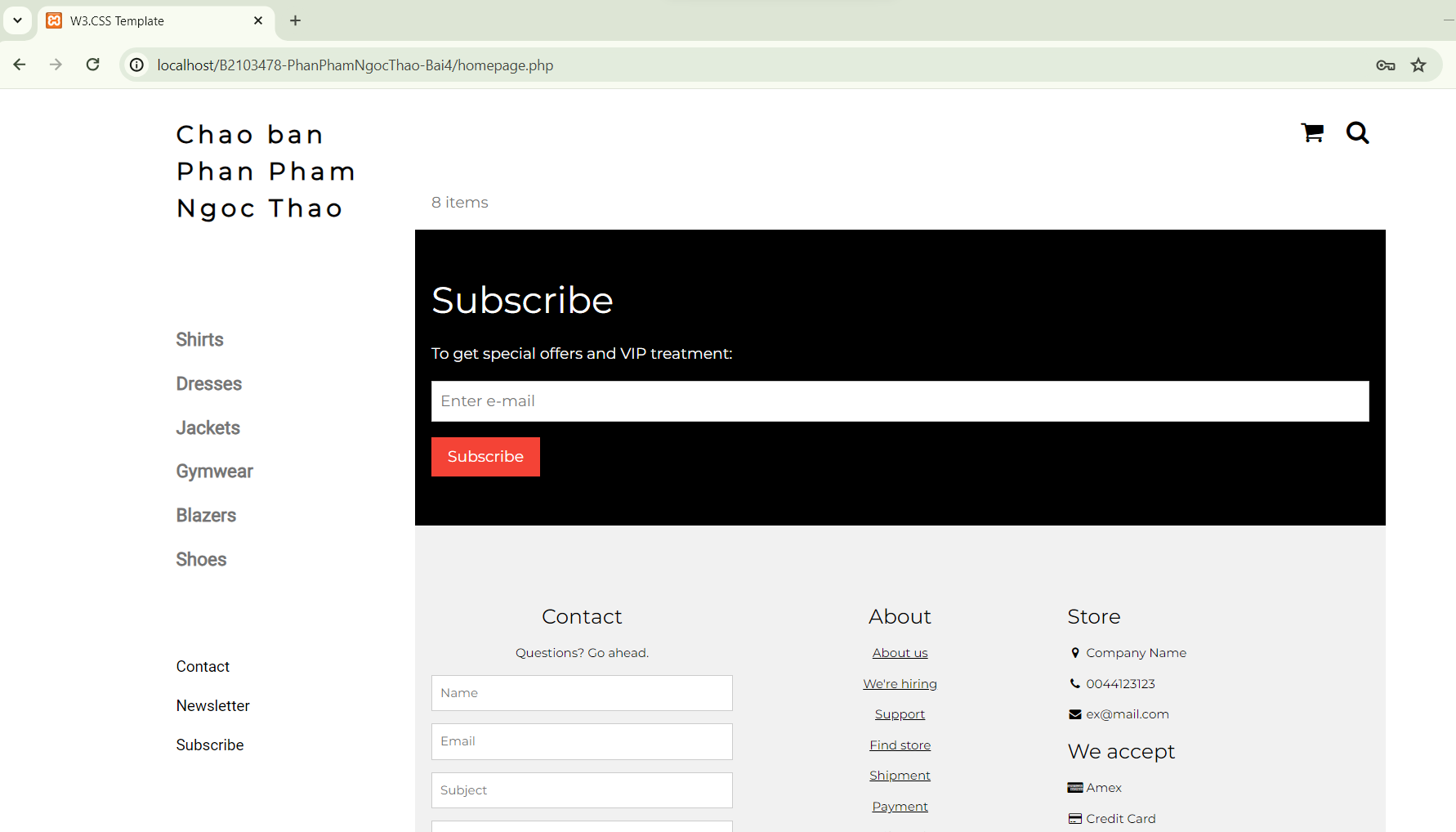
- Đăng nhập không thành công



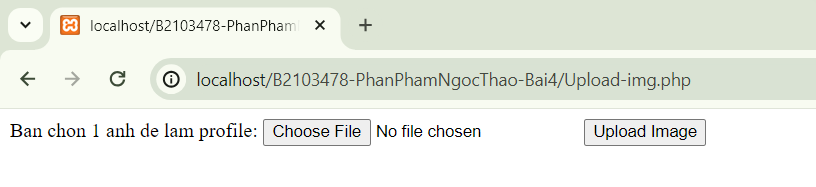
### Log.php (modified)



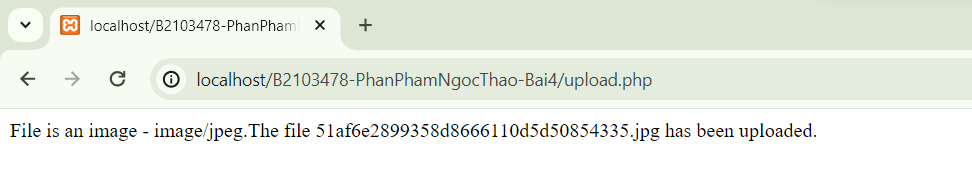
homepage.php



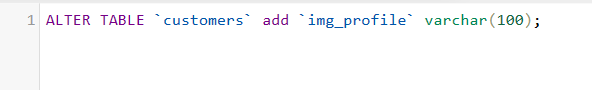
Upload-img.php



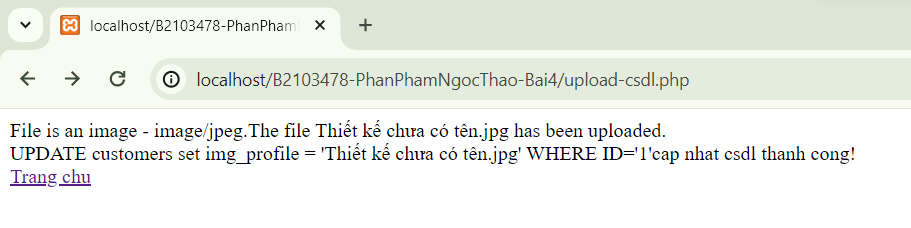
upload.php



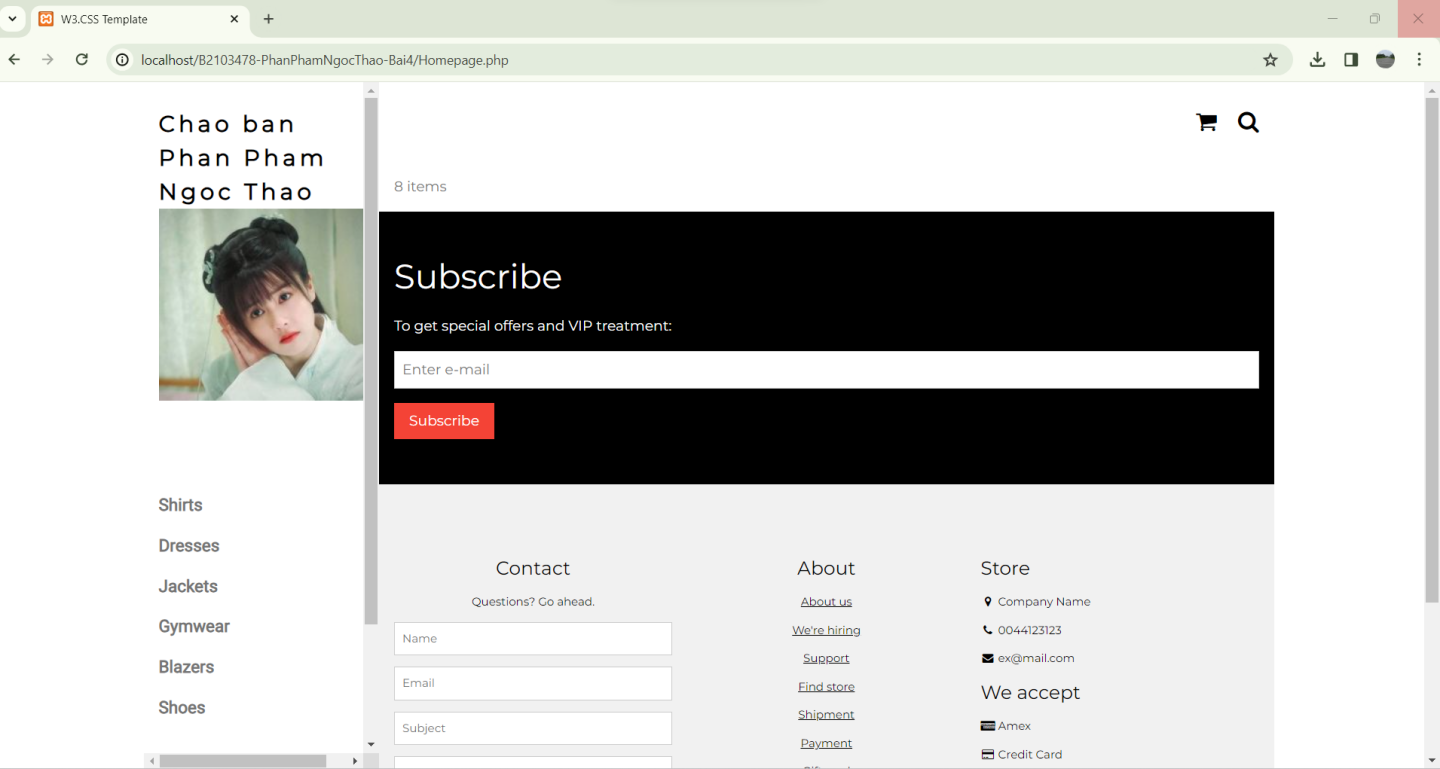
Thêm cột để lưu đường dẫn ảnh:



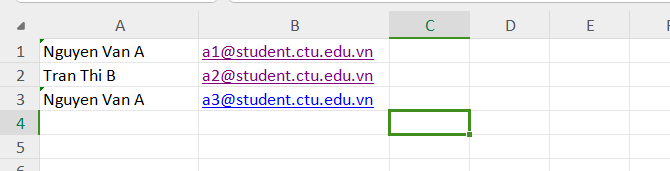
Upload-csdl.php



Homepage.php (modified với ảnh profile)



Tạo tập tin cus.csv với dữ liệu như sau



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

- **SQL Injection** là một kỹ thuật tấn công mạng được sử dụng bởi các hacker để xâm nhập vào hệ thống cơ sở dữ liệu bằng cách chèn các câu lệnh SQL bất hợp pháp vào các trường nhập liệu của ứng dụng web. Khi thành công, kỹ thuật này cho phép hacker thực hiện các hành động không được phép như truy cập, sửa đổi hoặc xóa dữ liệu, thậm chí kiểm soát toàn bộ hệ thống.

- Trong trường hợp này, nếu người dùng nhập một chuỗi như ' OR '1'='1 vào trường Username và Password, câu lệnh SQL sẽ trở thành:

SELECT \* FROM users WHERE username='' OR '1'='1' AND password='' OR '1'='1';

Do '1'='1' luôn đúng trong điều kiện SQL, điều này sẽ trả về tất cả các bản ghi từ bảng users, bỏ qua việc kiểm tra tên đăng nhập và mật khẩu. Kết quả là người dùng có thể đăng nhập mà không cần biết tên đăng nhập hoặc mật khẩu.

1. **Dựa vào link2, bạn hãy cho biết Cookie là gì, diễn giải ý nghĩa các tham số trong setcookie(). Cách lưu, lấy, xóa giá trị trong cookie.**

- Trong PHP, cookie là một cách để lưu trữ thông tin trên máy tính của người dùng khi họ truy cập vào một trang web cụ thể. Cookie được gửi từ máy chủ web đến trình duyệt của người dùng và được lưu trữ dưới dạng các cặp tên/giá trị. Cookie có thể được sử dụng để nhớ thông tin về người dùng như tên đăng nhập, cài đặt cá nhân, hoặc các dữ liệu khác.

- Trong PHP, để tạo một cookie, ta sử dụng hàm setcookie(). Các tham số của hàm này có ý nghĩa như sau:

1. name: Tên của cookie.
2. value: Giá trị của cookie.
3. expire: Thời gian sống của cookie, được biểu diễn dưới dạng timestamp. Mặc định là 0, nghĩa là cookie sẽ hết hiệu lực khi trình duyệt đóng.
4. path: Đường dẫn trên server mà cookie có hiệu lực. Mặc định là '/'.
5. domain: Tên miền mà cookie có hiệu lực. Mặc định là trang web hiện tại.
6. secure: Xác định liệu cookie có chỉ được gửi qua HTTPS không. Mặc định là false.
7. httponly: Nếu được đặt thành true, cookie sẽ chỉ được truy cập thông qua HTTP và không thể truy cập bằng JavaScript. Mặc định là false.

- Để lưu trữ giá trị vào cookie, bạn có thể sử dụng hàm setcookie() như sau:

setcookie("username", "John", time() + 3600, "/");

- Để lấy giá trị của cookie, ta sử dụng biến siêu toàn cục $\_COOKIE:

$username = $\_COOKIE["username"];

- Để xóa giá trị của cookie, ta có thể sử dụng hàm setcookie() với thời gian hết hiệu lực là thời điểm trong quá khứ:

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

1. **Dựa vào link3, bạn hãy cho biết Session dùng để làm gì. Cách lưu, lấy, xóa giá trị trong Session.**

- Session là một cách để lưu trữ dữ liệu trên máy chủ thay vì trên máy tính của người dùng, giúp duy trì trạng thái của người dùng qua các trang hoặc các yêu cầu gửi đến máy chủ.

- Lưu trữ giá trị trong session: Để lưu trữ giá trị trong session, ta sử dụng mảng siêu toàn cục $\_SESSION.

- Lấy giá trị từ session: Để lấy giá trị từ session, ta truy cập vào mảng $\_SESSION với khóa tương ứng.

- Xóa giá trị từ session: Để xóa một giá trị từ session, ta sử dụng lệnh unset() với khóa tương ứng trong mảng $\_SESSION.

1. **Bạn hãy so sánh Cookie và Session.**

***- Cookie*:**

+ Cookie là một cách lưu trữ thông tin trên máy tính của người dùng.

+ Dữ liệu trong cookie được gửi từ máy chủ đến trình duyệt của người dùng và được lưu trữ trên máy tính của họ trong một thời gian nhất định (do thời gian sống của cookie được chỉ định).

+ Cookie có thể được sử dụng để theo dõi và nhớ thông tin của người dùng trên nhiều trang web.

+ Dữ liệu trong cookie có thể được chỉnh sửa hoặc xóa bởi người dùng.

***- Session:***

+ Session là một cách lưu trữ thông tin trên máy chủ.

+ Mỗi phiên làm việc của người dùng sẽ có một session tương ứng trên máy chủ.

+ Thông tin session không được lưu trữ trên máy tính của người dùng, mà thay vào đó được lưu trữ trên máy chủ và được quản lý thông qua một ID session được gửi đến trình duyệt của người dùng thông qua cookie hoặc URL.

+ Dữ liệu trong session không thể được chỉnh sửa hoặc xóa một cách dễ dàng từ phía người dùng mà chỉ có thể được thay đổi từ phía máy chủ.

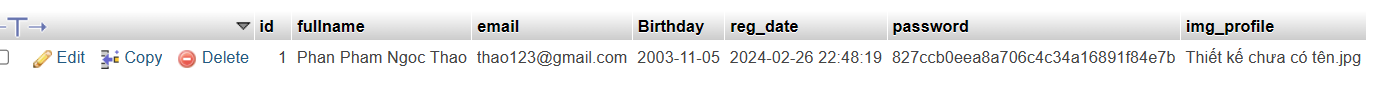
- **Bảo mật**: Do session dữ liệu được lưu trữ trên máy chủ, nên an toàn hơn so với cookie, nơi dữ liệu được lưu trữ trên máy tính của người dùng và có thể bị sửa đổi.

- **Hiệu suất**: Cookie thường tốn ít tài nguyên hơn so với session vì dữ liệu không cần lưu trữ trên máy chủ. Tuy nhiên, việc sử dụng cookie để lưu trữ thông tin lớn có thể làm chậm tốc độ tải trang.

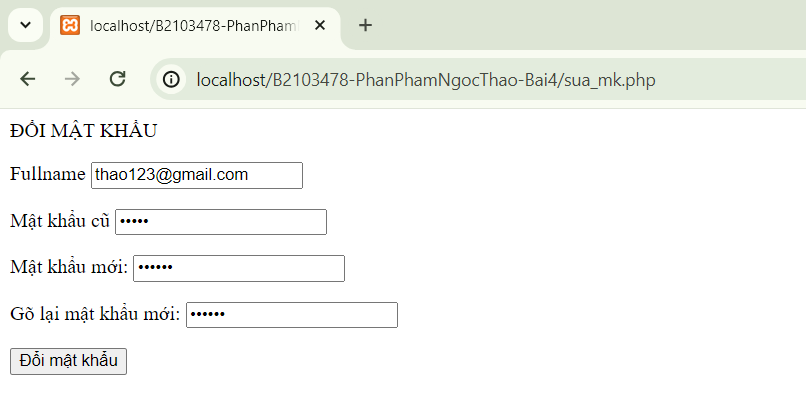
- **Khả năng chia sẻ dữ liệu**: Cookie thường được sử dụng để chia sẻ thông tin giữa các trang web hoặc các domain khác nhau. Trong khi đó, session thường được sử dụng để lưu trữ thông tin chỉ trong phạm vi của một ứng dụng cụ thể.

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.**

Mật khẩu cũ



Form đổi mật khẩu



Mật khẩu được đổi



1. **Bạn đọc trong4 để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.**

**- Form HTML:**

Để cho phép người dùng tải tệp lên máy chủ, bạn cần tạo một biểu mẫu HTML có thuộc tính enctype="multipart/form-data" để cho phép tải lên tệp:

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

Select image to upload:

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

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

</form>

- **Hàm move\_uploaded\_file():**

Trong script PHP, sau khi người dùng gửi tệp lên máy chủ, bạn cần sử dụng hàm move\_uploaded\_file() để di chuyển tệp từ vị trí tạm thời đã được lưu trữ (do PHP tạo ra) đến vị trí mà bạn muốn lưu trữ trên máy chủ.

$target\_dir = "uploads/";

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

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

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

if ($\_FILES["fileToUpload"]["size"] > 500000) {

echo "Sorry, your file is too large.";

$uploadOk = 0;

}

- **Chỉ định vị trí lưu trữ:**

Cần xác định vị trí trên máy chủ mà tệp sẽ được lưu trữ. Điều này thường được thực hiện thông qua biến $target\_dir

1. **Bạn hãy đọc5, mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).**

- **Hàm file\_exists()**

Chức năng:

Hàm file\_exists() kiểm tra xem một tệp hoặc thư mục có tồn tại hay không

file\_exists($file)

**- Tham số:**

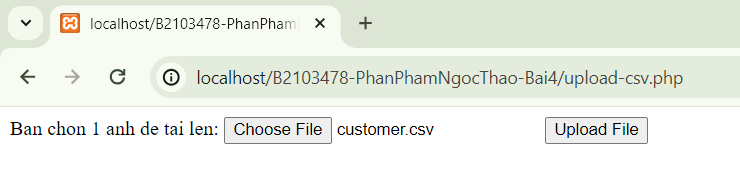
**$file**: Đường dẫn đến tệp hoặc thư mục mà bạn muốn kiểm tra.

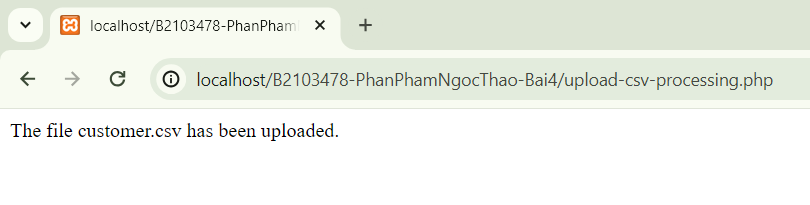
Giá trị trả về:

**true** nếu tệp hoặc thư mục tồn tại.

**false** nếu tệp hoặc thư mục không tồn tại hoặc nếu không thể truy cập được.

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: [[1]](#footnote-0),[[2]](#footnote-1). 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. [↑](#footnote-ref-0)
2. [↑](#footnote-ref-1)