**Bài tập thực hành 04**

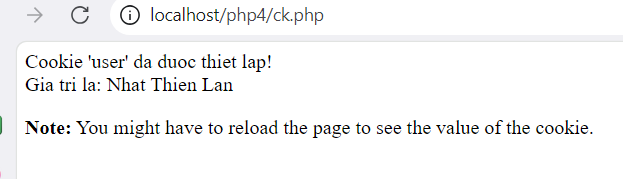
Lập trình PHP nâng cao

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

## Tạo Cookies

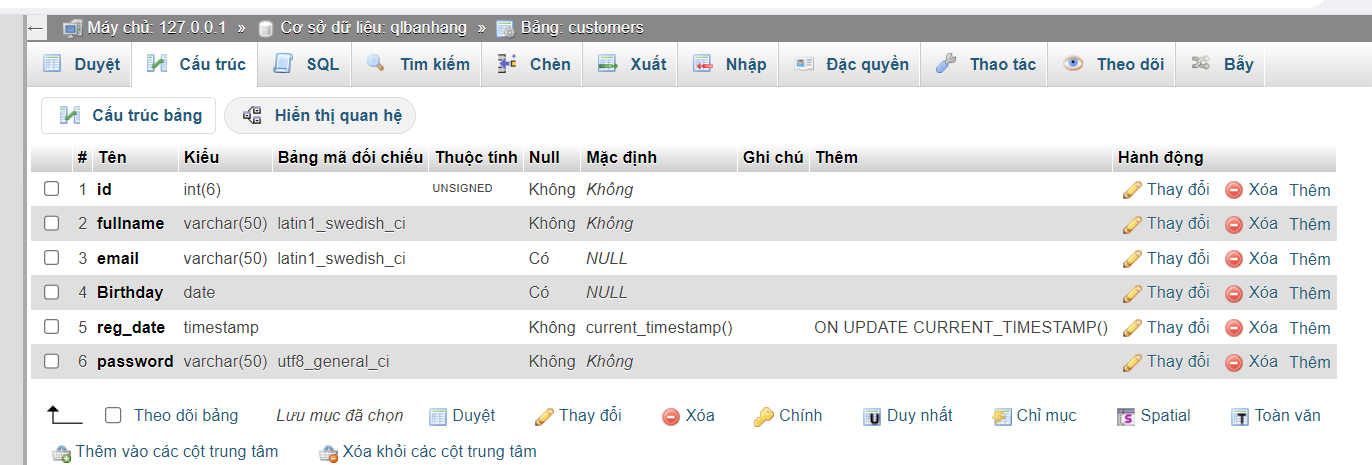
- Cookies và Session hỗ trợ chúng ta quản lý các phiên đăng nhập/đăng nhập, hoạt động của người dùng trên các website.

Tạo tập tin sử dụng khởi tạo cookies ck.php sau:



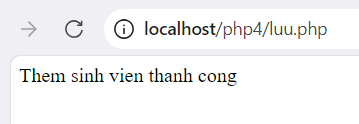
## Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa

Tạo csdl qlbanhang, với bảng customers đầu tiên như sau:

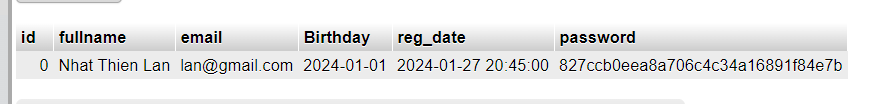


Tạo formnhap.php

Và luu.php, chú ý lúc này chúng ta sẽ mã hóa mật khẩu với giải thuật md5 để bảo mật mật khẩu:

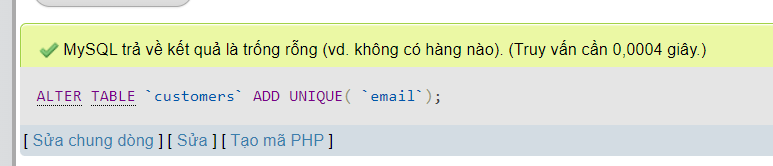


Sau khi chúng ta nhập xong mật khẩu với 12345 nhưng khi quan sát trong CSDL ta sẽ thấy mật khẩu đã được mã hóa ra chuỗi 827ccb0eea8a706c4c34a16891f84e7b:



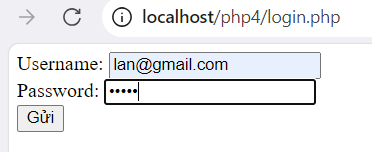
Trong trường hợp chúng ta muốn dùng email là username để đăng nhập ta nên đảm bảo 2 student không có email trùng nhau. Để đảm bảo điều này ta có thể chọn email là unique key (vì ta đã chọn id la primary key, và 1 bảng không thể có 2 khóa chính)

Sau khi thêm dòng này: [ALTER](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) [TABLE](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/alter-table.html) `customers` ADD UNIQUE( `email`);



## Tạo form đăng nhập

Sau khi tạo người dùng xong, ta có thể chạy form đăng nhập với username là email và password lấy từ CSDL



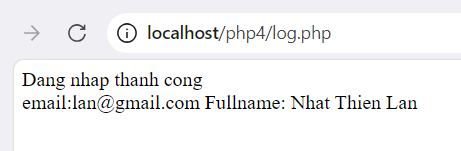
### Log.php

Khi chúng ta nhập sai, nhìn vào thông báo lỗi ta thấy



Mật khẩu đã cho không khớp với mã đã băm

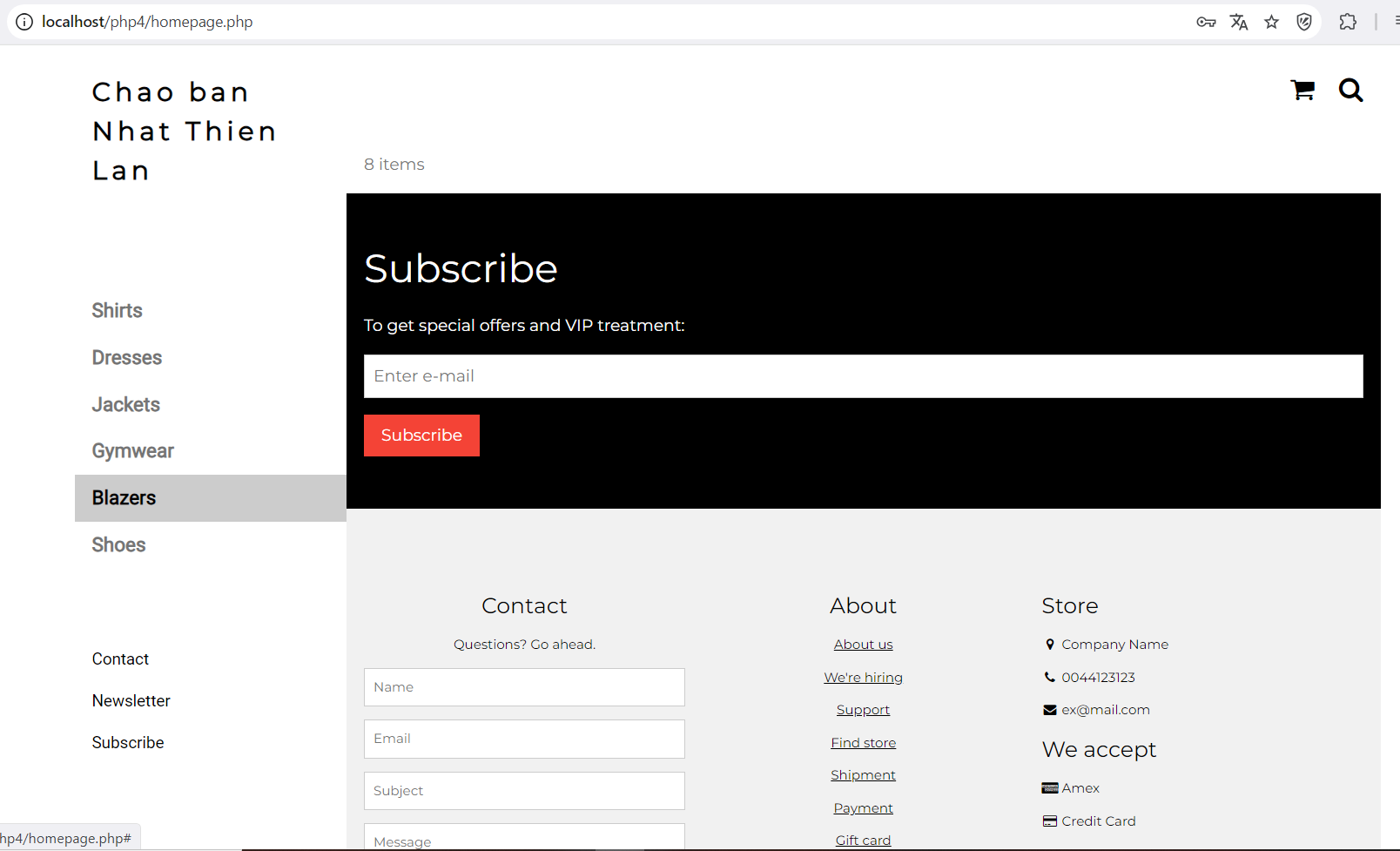
Khi ta nhập đúng



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

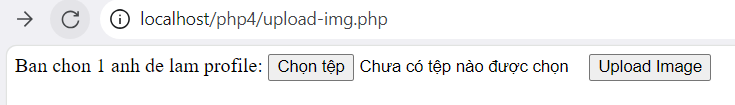
Chỉnh sửa lại log.php với code sau để khi đăng nhập thành công chuyển sang 1 trang chủ, còn nếu không đúng thì trở lại trang đăng nhập

### Log.php (modified) + Homepage.php



Upload tập tin

Upload file, chúng ta tạo form giao diện để upload file ảnh upload-img.php

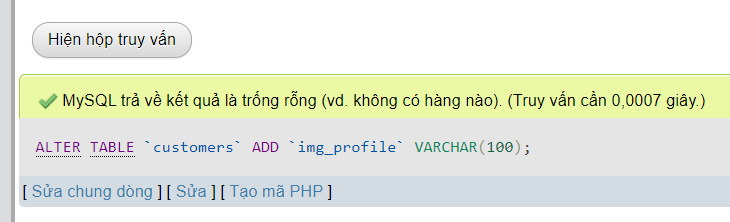


Upload ảnh và lưu dữ liệu về ảnh trong CSDL

Trong phần này chúng ta minh họa tải tập tin ảnh làm ảnh profile

của người dùng

Thêm cột để lưu đường dẫn ảnh: ALTER TABLE `customers` ADD `img\_profile` VARCHAR(100);



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 trong1 . 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.

- SQL Injection là một trong những kỹ thuật hack web phổ biến nhất.

- SQL Injection là việc đặt mã độc vào câu lệnh SQL thông qua đầu vào trang web.

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.**

- Cookie thường được sử dụng để nhận dạng người dùng. Cookie 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.

- setcookie(name, value, expire, path, domain, secure, httponly);

**Trong đó:**

1. **name (Bắt buộc):**

Đây là tên của cookie. Cookie được xác định thông qua tên này khi nó được gửi từ trình duyệt đến máy chủ.

1. **value (Bắt buộc):**

Đây là giá trị của cookie, được liên kết với tên cookie. Dữ liệu này thường được mã hóa để đảm bảo tính toàn vẹn và bảo mật.

1. **expire (Tùy chọn):**

Xác định thời gian sống của cookie, tính bằng giây kể từ thời điểm hiện tại. Nếu giá trị này không được đặt, cookie sẽ tồn tại trong suốt phiên làm việc của trình duyệt.

1. **path (Tùy chọn):**

Đường dẫn trên máy chủ mà cookie có thể được truy cập. Nếu đường dẫn không được xác định, cookie sẽ có thể truy cập từ tất cả các đường dẫn.

1. **domain (Tùy chọn):**

Miền mà cookie có thể áp dụng. Nếu không xác định, cookie chỉ sẽ áp dụng cho miền của trang web hiện tại.

1. **secure (Tùy chọn):**

Nếu được đặt thành true, cookie chỉ sẽ được gửi nếu kết nối là an toàn (https). Mặc định là false.

1. **httponly (Tùy chọn):**

Nếu được đặt thành true, cookie chỉ sẽ được truy cập thông qua HTTP và JavaScript sẽ không thể truy cập nó. Điều này giúp bảo vệ cookie khỏi các tấn công Cross-Site Scripting (XSS). Mặc định là false.

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

**Cách lưu**

setcookie("user", "John Doe", time() + 3600, "/");

**Cách lấy**

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

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

echo "Hello, $user!";

} else {

echo "Cookie 'user' không tồn tại.";

}

**Cách xóa**

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

1. **Dựa vào link3, 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.

**\*Cách lưu, lấy, xóa giá trị trong Session.**

**Cách lưu**

session\_start();

$\_SESSION["user"] = "John Doe";

**Cách lấy**

session\_start();

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

$user = $\_SESSION["user"];

echo "Hello, $user!";

} else {

echo "Giá trị 'user' không tồn tại trong Session.";

}

**Cách xóa**

session\_start();

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

1. **Bạn hãy so sánh Cookie và Session.**
2. **Nơi lưu trữ dữ liệu:**

Cookie: Lưu trữ trên máy khách (client-side), thông qua trình duyệt.

Session: Lưu trữ trên máy chủ (server-side).

1. **Bảo mật:**

Cookie: Có thể bị mạo danh nếu bị đánh cắp hoặc chỉnh sửa trên máy khách.

Session: An toàn hơn vì thông tin lưu trữ trên máy chủ, không thể bị truy cập hoặc chỉnh sửa trực tiếp từ phía người dùng.

1. **Thời gian sống:**

Cookie: Có thể có thời gian sống cố định hoặc thời gian sống chỉ trong phiên làm việc của trình duyệt.

Session: Thường tồn tại trong suốt phiên làm việc của người dùng và sẽ bị xóa khi người dùng đăng xuất hoặc phiên làm việc kết thúc.

1. **Lưu trữ khả năng:**

Cookie: Có giới hạn về kích thước (thường khoảng 4KB).

Session: Có thể lưu trữ lượng lớn dữ liệu vì thông tin được lưu trữ trên máy chủ.

1. **Hiệu suất:**

Cookie: Dữ liệu được truyền giữa máy khách và máy chủ trong mỗi yêu cầu, làm tăng kích thước của yêu cầu.

Session: Chỉ có một ID session được truyền giữa máy khách và máy chủ, giảm lượng dữ liệu truyền đi so với sử dụng cookie để lưu trữ thông tin trạng thái.

1. **Quản lý:**

Cookie: Lưu trữ trên máy khách và có thể bị quản lý bởi trình duyệt người dùng.

Session: Lưu trữ trên máy chủ và được quản lý bởi ứng dụng web.

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

thuộc tính enctype="multipart/form-data" để cho phép tải lên tệp tin.

1. move\_uploaded\_file() - Di chuyển tệp tin đã tải lên:

Hàm này được sử dụng để di chuyển tệp tin từ vị trí tạm thời (thư mục tmp\_name) vào vị trí đích trên máy chủ.

1. $\_FILES - Biến toàn cục chứa thông tin về tệp tin đã tải lên:

$\_FILES là một mảng chứa thông tin về tệp tin đã được tải lên, bao gồm tên, loại, tên tệp tạm thời, kích thước, và trạng thái của quá trình tải lên.

1. pathinfo() - Trích xuất thông tin về đường dẫn của tệp tin:

Hàm này trích xuất thông tin về đường dẫn của tệp tin, chẳng hạn như phần mở rộng (extension).

1. getimagesize() - Kiểm tra xem tệp tin có phải là hình ảnh không:

Hàm này kiểm tra xem tệp tin có phải là hình ảnh hợp lệ hay không.

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

$csv: Một mảng trống để lưu trữ dữ liệu CSV.

$name\_file: Tên của tệp CSV (cus.csv trong trường hợp này).

$lines: Hàm file() đọc nội dung của tệp CSV vào một mảng ($lines). Cờ FILE\_IGNORE\_NEW\_LINES được sử dụng để bỏ qua ký tự xuống dòng ở cuối mỗi dòng.

Vòng lặp foreach duyệt qua từng dòng trong mảng $lines.

str\_getcsv($value): Hàm str\_getcsv() được sử dụng để phân tích mỗi dòng thành một mảng giá trị. Hàm này được thiết kế đặc biệt để phân tích chuỗi CSV.

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)