**BÀI THỰC HÀNH 04**

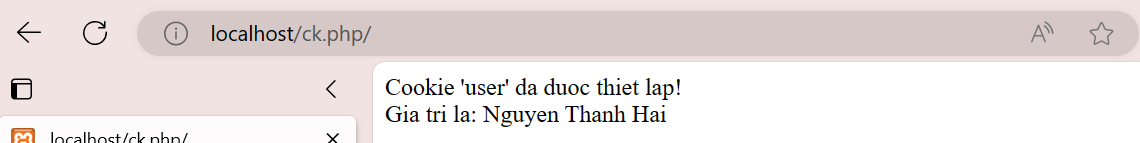
1. *Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.*

**Tạo tập tin sử dụng khởi tạo cookies ck.php**

A white background with black text

Description automatically generated

**Sau khi nhấn f5:**



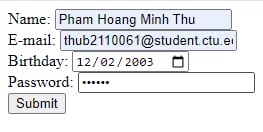
**Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa**

* Tạo csdl qlbanhang, với bảng customers đầu tiên

A screenshot of a computer

Description automatically generated

* Tạo formnhap.php

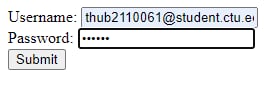


* luu.php

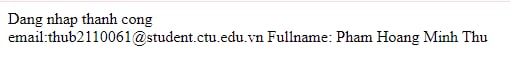


**Tạo form đăng nhập**

* Login.php

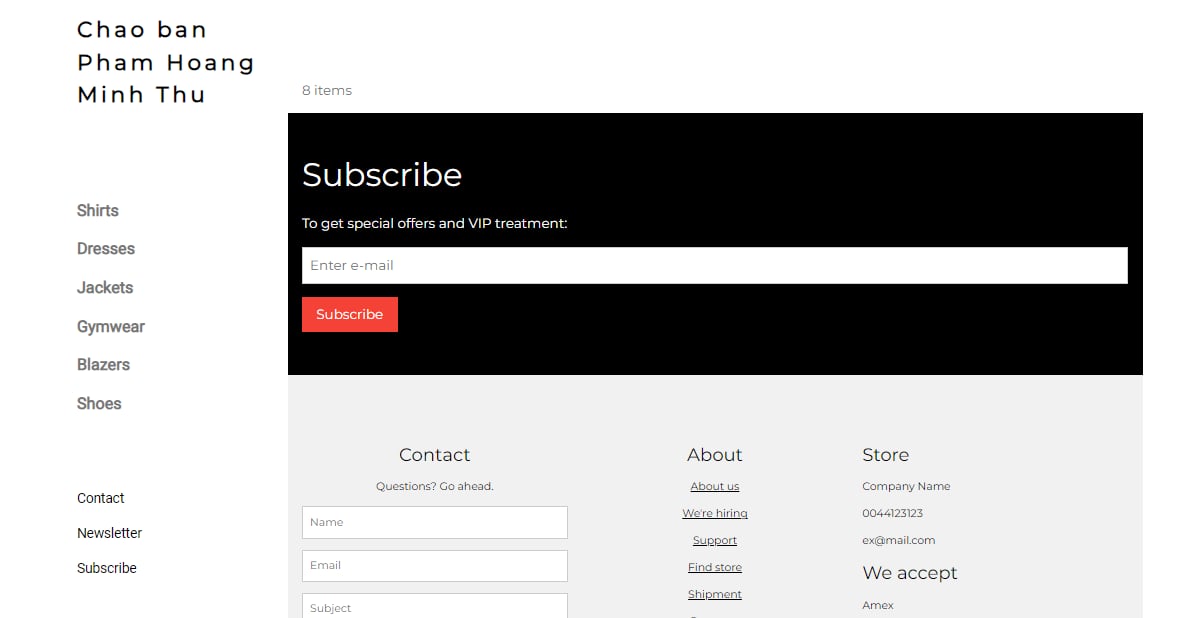


* Log.php



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

* Sau để khi đăng nhập thành công chuyển sang 1 trang chủ



**Upload tập tin**

* Upload-img.php



* Tập tin upload.php xử lý hành động

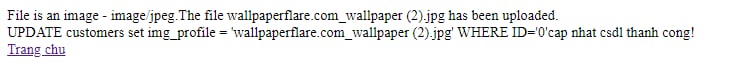


**Upload ảnh và lưu dữ liệu về ảnh trong CSDL**

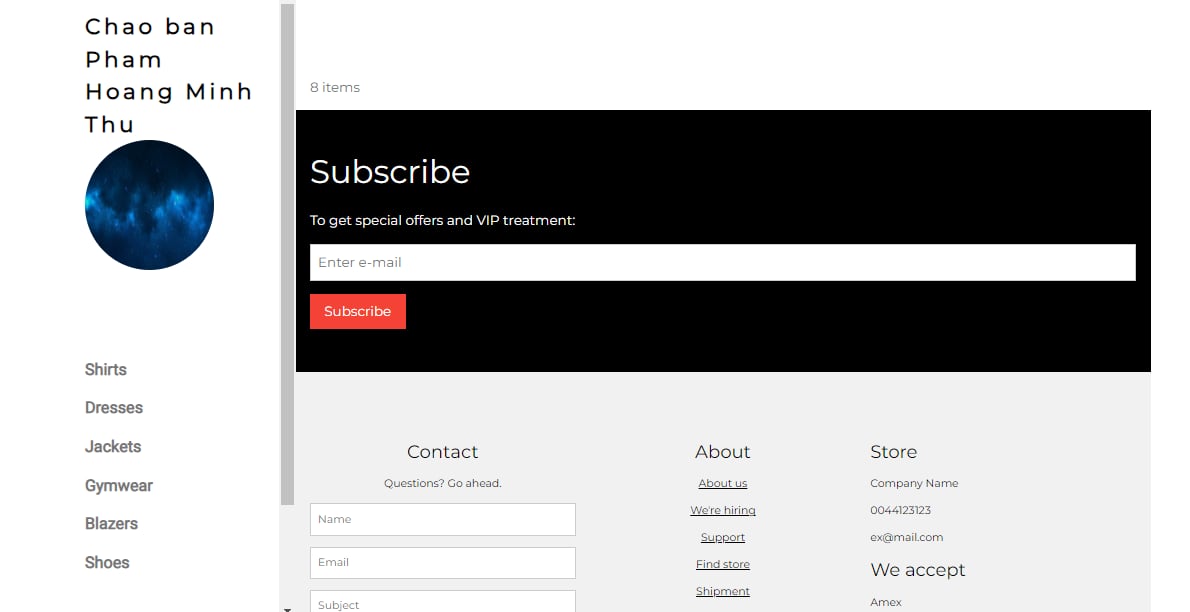
* Upload-img.php (modified)



* Upload-csdl.php



* Homepage.php (modified với ảnh profile)



**Upload file csv và đọc dữ liệu vào mảng**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

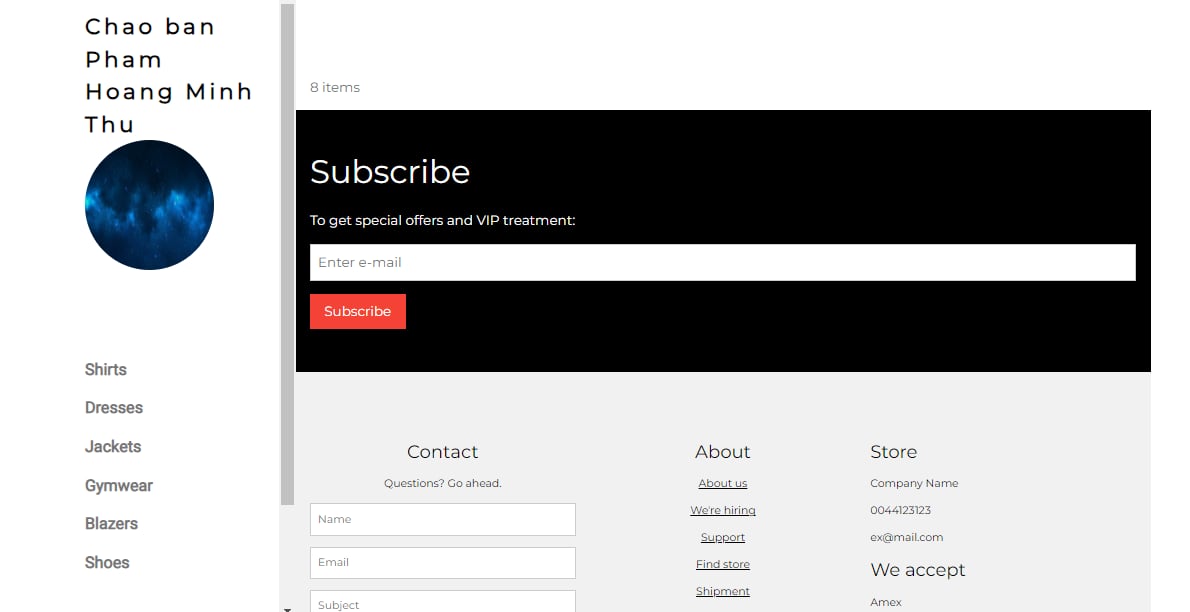
Description automatically generated

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 phổ biến trong lĩnh vực bảo mật web. Nó lợi dụng lỗ hổng trong việc kiểm tra dữ liệu đầu vào của ứng dụng web để chèn thêm các câu lệnh SQL bất hợp pháp.
* Kết quả:

A computer screen shot of a computer screen

Description automatically generated



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** là một tập hợp các tệp thông tin do chính người dùng tạo ra mỗi lần truy cập vào một website nào đó và để lại dữ liệu trình duyệt web.
* Ý nghĩa các tham số trong setcookie().
* name: Tên của Cookie.
* value: Giá trị của Cookie.
* expire: Thời gian hết hạn của Cookie (thường là số giây tính từ thời điểm hiện tại).
* path: Đường dẫn trên máy chủ mà Cookie có hiệu lực.
* domain: Tên miền mà Cookie áp dụng.
* secure: Xác định liệu Cookie có được gửi qua kết nối an toàn (HTTPS) hay không.
* httponly: Xác định liệu Cookie có thể truy cập bằng JavaScript hay không.

**Lưu giá trị trong Cookie:**

Để lưu giá trị trong Cookie, bạn sử dụng hàm **setcookie()**

**Lấy giá trị từ Cookie:**

Để lấy giá trị từ Cookie, bạn sử dụng biến **$\_COOKIE**

* Ví dụ: $\_COOKIE['username'] sẽ trả về giá trị của Cookie có tên là “username”.

**Xóa giá trị trong Cookie:**

* Để xóa giá trị trong Cookie, bạn đặt thời gian hết hạn của Cookie là một thời điểm trong quá khứ.
* Ví dụ: setcookie('username', '', time() - 3600); sẽ xóa giá trị của Cookie “username” bằng cách đặt thời gian hết hạn là 1 giờ trước thời điểm hiện tại.

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

* **Session** trong PHP là một cách để lưu trữ thông tin (trong các biến) được sử dụng trên nhiều trang.
* **Lưu giá trị trong Session**
* Để lưu một giá trị mới vào Session ta dùng cú pháp như sau:

$\_SESSION['session\_name'] = $session\_value

* **Lấy giá trị trong Session**
* Để lấy giá trị Session ta dùng cú pháp sau: $tenbien = $\_SESSION['session\_name'].
* **Xóa giá trị trong Session**
* Để xóa giá trị trong Session ta dùng cú pháp sau: unset($\_SESSION['session\_name'])

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

* Cookie:
* **Khái niệm**: Cookie là một tệp tin tạm thời lưu trữ trên máy tính của người dùng khi truy cập vào ứng dụng.
* **Lưu trữ**: Dữ liệu của cookie được lưu trữ trực tiếp trên trình duyệt của người dùng (client).
* **Định danh**: Cookie được gửi từ server tới browser và lưu trữ trên máy tính của người dùng.
* **Sử dụng**: Thường được tạo ra khi người dùng truy cập một website để ghi nhớ thông tin như tên đăng nhập, mật khẩu, và các tuỳ chọn.
* **An toàn**: Dễ bị tấn công do lưu trữ trên máy tính người dùng.
* Session:
* **Khái niệm**: Session là một phiên làm việc, tồn tại từ khi client gửi request đến server cho đến khi hết thời gian timeout hoặc khi bạn đóng ứng dụng.
* **Lưu trữ**: Dữ liệu session được lưu trên máy chủ (server).
* **Định danh**: Mỗi session được cấp phát một định danh duy nhất (SessionID).
* **Sử dụng**: Thường được sử dụng cho các chức năng như đăng nhập, đăng xuất, và lưu trữ thông tin tạm thời.
* **An toàn**: Khó bị sửa đổi hoặc tấn công vì lưu trữ trên server.

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.*

**Homepage.php**

A screen shot of a computer program

Description automatically generated

**Log.php**

A black background with white text and colorful symbols

Description automatically generated

**Thoat.php**

A screen shot of a computer code

Description automatically generated

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

*Định nghĩa một biểu mẫu HTML để cho phép người dùng chọn tệp tin để tải lên:*

* Sử dụng thẻ `input` với phương thức `POST` và thuộc tính`enctype="multipart/form-data"`.
* Sử dụng thẻ với thuộc tính `type="file"` để người dùng có thể chọn tệp tin

*Trên phía máy chủ, sử dụng các hàm và thủ tục PHP để xử lý việc tải lên:*

* Sử dụng hàm `move\_uploaded\_file()` để di chuyển tệp tin được tải lên vào vị trí lưu trữ trên máy chủ.
* Sử dụng biến siêu toàn cục `$\_FILES` để truy cập thông tin về tệp tin đã tải lên, bao gồm tên tệp tin, loại tệp tin, đường dẫn tạm thời và kích thước tệp tin.

*Kiểm tra và xử lý lỗi khi tải lên tệp tin:*

* Sử dụng thuộc tính `$\_FILES['file']['error']` để kiểm tra mã lỗi liên quan đến tải lên tệp tin.
* Sử dụng các câu lệnh điều kiện để kiểm tra lỗi và xử lý mỗi trường hợp lỗi.

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

* `file\_exists($filename)`: Kiểm tra xem một tệp tin có tồn tại hay không
* `$filename`: Tên hoặc đường dẫn tới tệp tin cần kiểm tra.
* Trả về `true` nếu tệp tin tồn tại, ngược lại trả về `false`
* `is\_file($filename)`: Kiểm tra xem một tệp tin có phải là một tệp tin hay không
* `$filename`: Tên hoặc đường dẫn tới tệp tin cần kiểm tra.
* Trả về `true` nếu tệp tin là một tệp tin, ngược lại trả về `false`.
* `file\_get\_contents($filename)`: Đọc nội dung của một tệp tin và trả về dưới dạng một chuỗi.
* `$filename`: Tên hoặc đường dẫn tới tệp tin cần đọc
* Trả về nội dung của tệp tin dưới dạng chuỗi, hoặc `false` nếu có lỗi xảy ra.
* `fopen($filename, $mode)`: Mở một tệp tin hoặc URL.
* `$filename`: Tên hoặc đường dẫn tới tệp tin cần mở.
* `$mode`: Chế độ mở tệp tin, ví dụ: "r" (mở tệp tin chỉ để đọc), "w" (mở tệp tin để ghi, xóa nội dung cũ), "a" (mở tệp tin để ghi, giữ nội dung cũ).
* Trả về một con trỏ tệp tin (file pointer) hoặc `false` nếu có lỗi xảy ra.
* `fgets($file)` và `fgetcsv($file)`: Đọc một dòng từ tệp tin.
* `$file`: Con trỏ tệp tin trả về từ hàm `fopen()`.
* `fgets($file)`: Đọc một dòng từ tệp tin và trả về dưới dạng một chuỗi.
* `fgetcsv($file)`: Đọc một dòng từ tệp tin và trả về dưới dạng một mảng chứa các giá trị được phân tách bằng dấu phẩy (dùng cho đọc tệp tin 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*

