Bài tập thực hành 04

**MSSV: B2110065**

**Họ và tên sinh viên: Phạm Hoàng Tuấn**

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.

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:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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

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

A screenshot of a computer

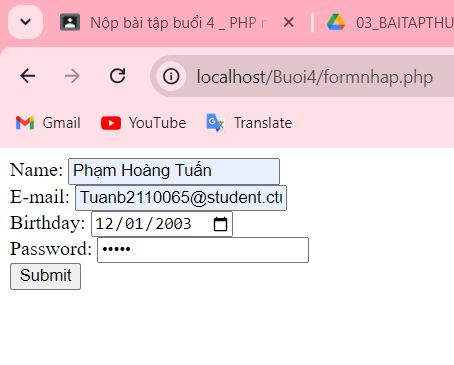
Description automatically generated

A screenshot of a computer

Description automatically generated

Tạo formnhap.php

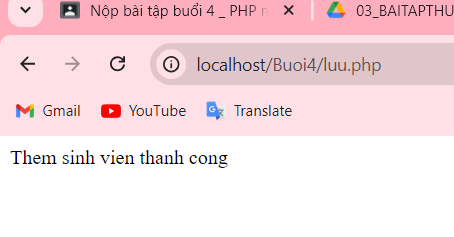




luu.php

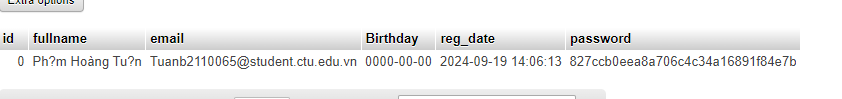
A screenshot of a computer program

Description automatically generated



Sau khi chúng ta nhập xong mật khẩu với 12345 nhưng khi quan sát trong CSDL ta sẽ thấy mật

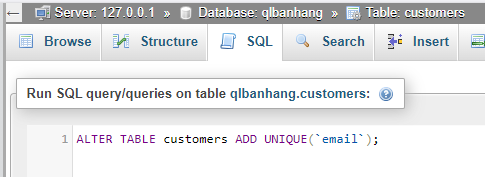
khẩu đã được mã hóa ra chuỗi 827ccb0eea8a706c4c34a16891f84e7b:



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)

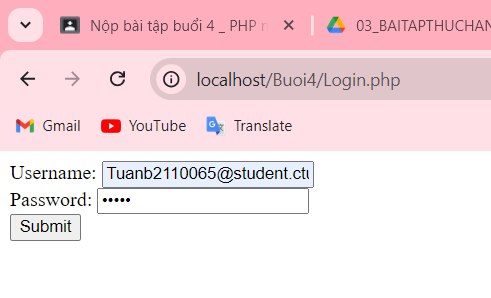


Tạo form đăng nhập

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

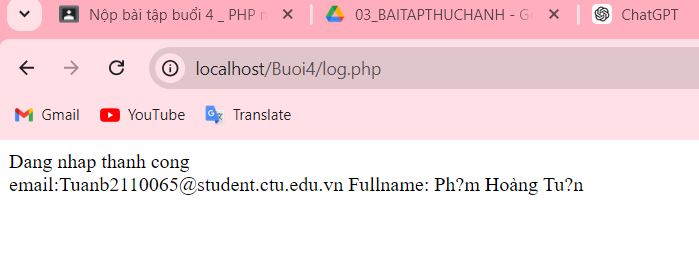
password lấy từ CSDL





Log.php



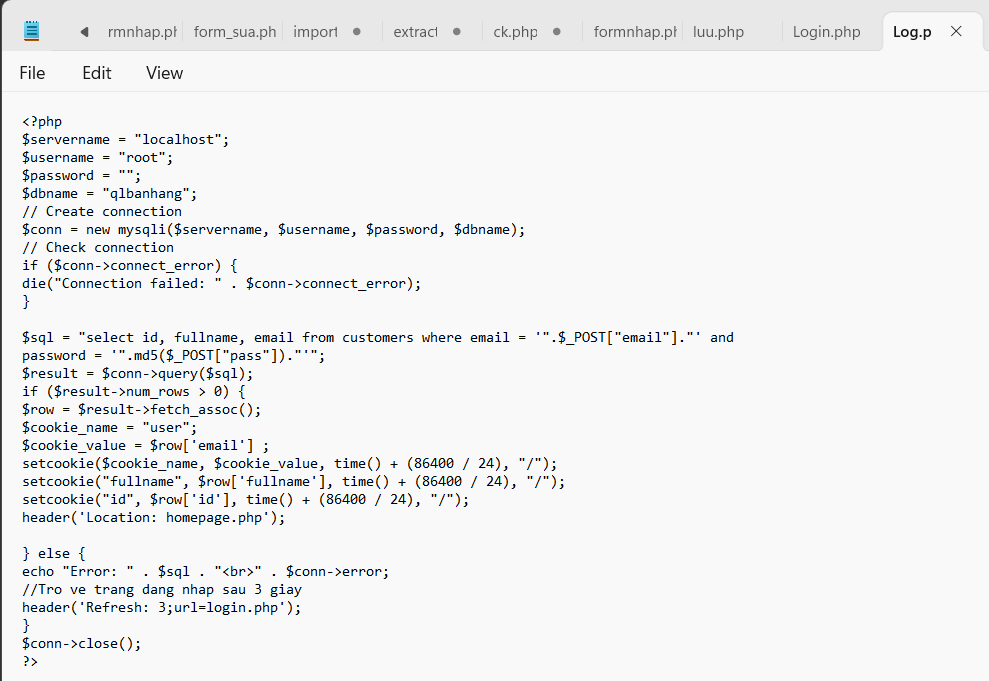


Dùng Cookies lưu lại tên đăng nhập, email và điều hướng

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

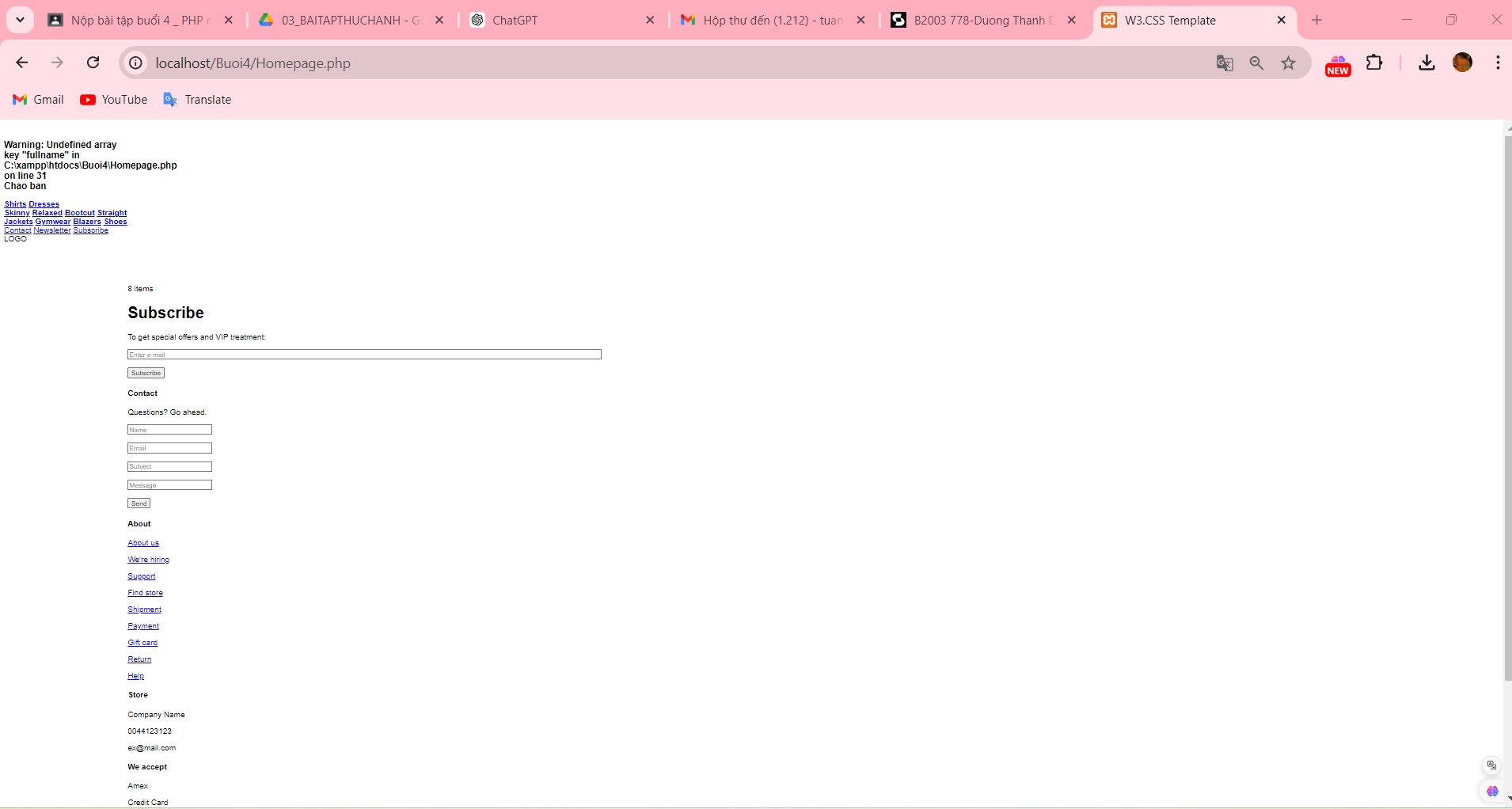
nếu không đúng thì trở lại trang đăng nhập

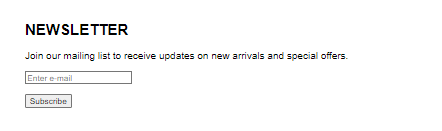
Log.php (modified)



Tạo 1 trang chủ để khi đăng nhập thành công chuyển qua trang chủ này:

Homepage.php

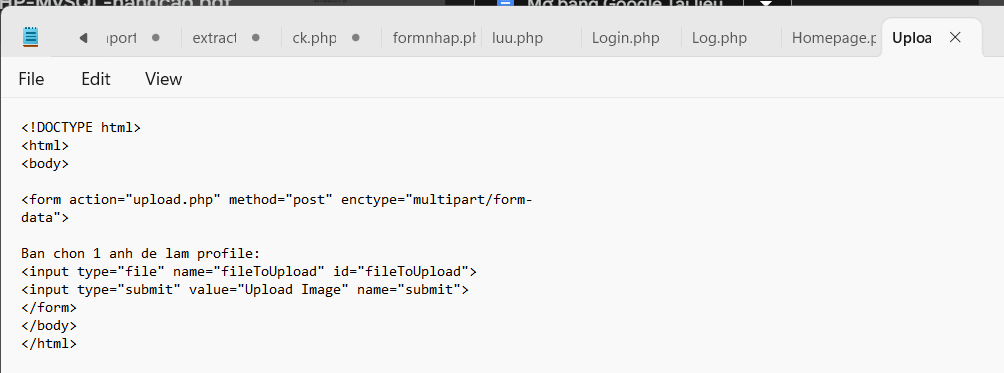


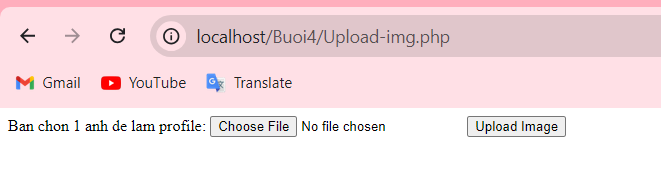


Upload tập tin

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

Upload-img.php





Tập tin upload.php xử lý hành động

upload.php

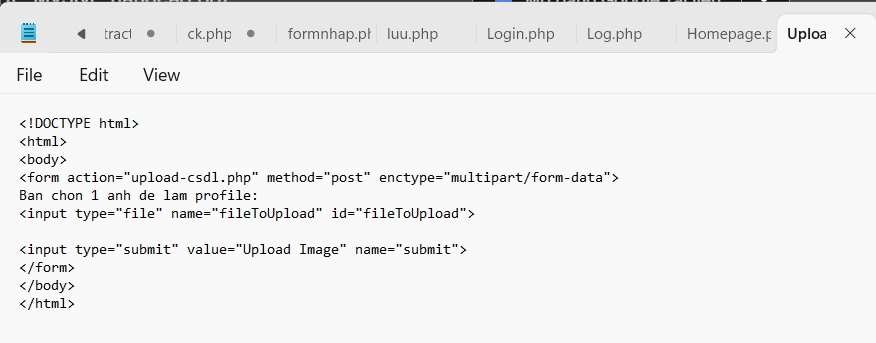


Upload ảnh và lưu dữ liệu về ảnh trong CSDL

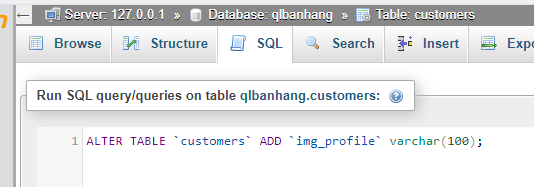
Trong phần này chúng ta minh họa tải tập tin ảnh làm ảnh profile

của người dùng

Upload-img.php (modified)



Thêm cột để lưu đường dẫn ảnh:



A screenshot of a computer

Description automatically generated

Upload file csv và đọc dữ liệu vào mảng

Thao tác đọc dữ liệu từ các tập tin là quan trọng, giúp chúng ta backup dữ liệu và phục hồi hay

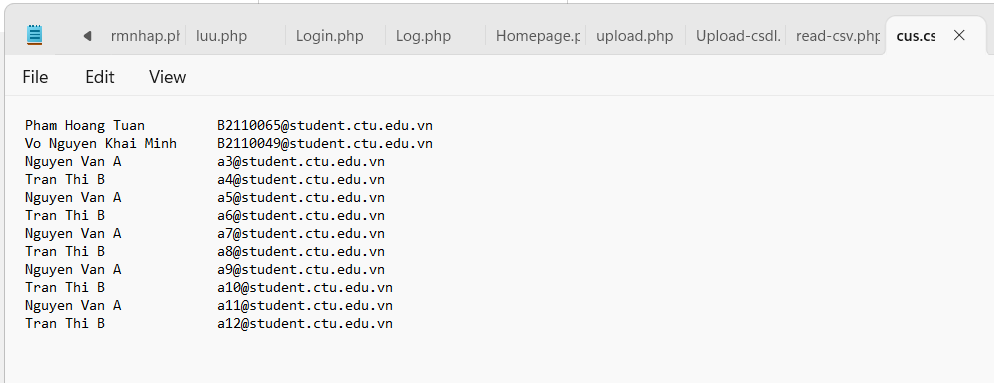
import dữ liệu từ các hệ thống khác. Ví dụ dưới đây giúp minh họa cho các bạn thao tác đọc dữ

liệu từ tập tin csv và từ đó đưa vào mảng để phục vụ cho việc xử lý.

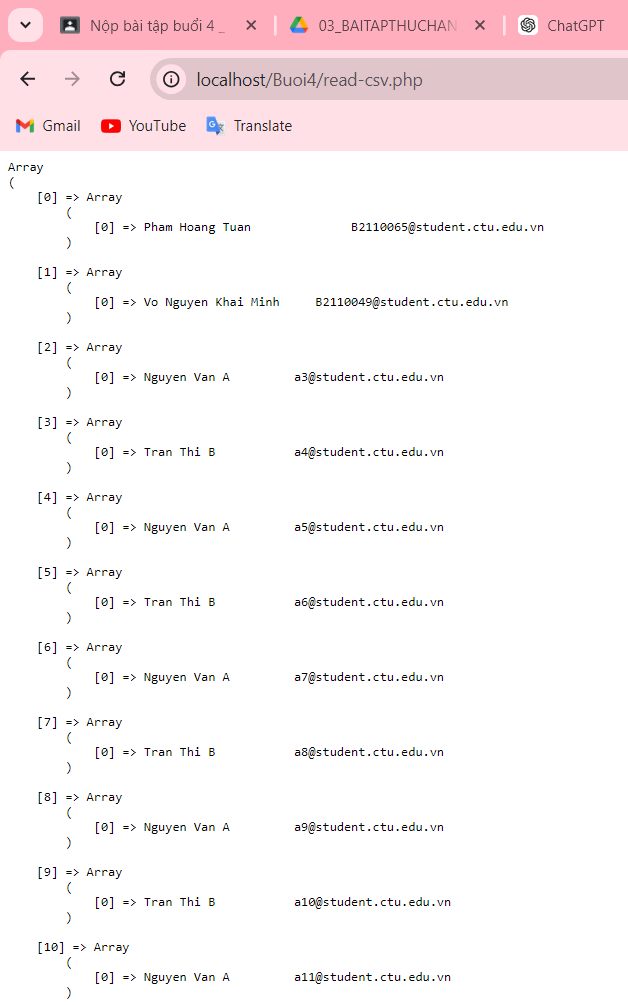
read-csv.php:



**Tạo tập tin cus.csv với dữ liệu như sau**

****

**Khi chạy code đọc trên ta sẽ có dữ liệu đã được đưa vào mảng và hiển thị**

****

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.

SQL Injection là gì?

-SQL Injection (SQLi) là một kỹ thuật tấn công an ninh mà trong đó, kẻ tấn công sử dụng các câu lệnh SQL độc hại để can thiệp hoặc thao tác vào cơ sở dữ liệu của một ứng dụng web. Kỹ thuật này thường được sử dụng cho mục đích xác thực (như trang đăng nhập) để lấy quyền truy cập không hợp pháp vào thông tin nhạy cảm, hoặc để thực hiện các thao tác không mong muốn khác trên cơ sở dữ liệu.

Cách thức hoạt động:

* Kẻ tấn công thực hiện việc nhập mã SQL độc hại vào các trường đầu vào của ứng dụng, ví dụ như trường tên đăng nhập hoặc mật khẩu.
* Nếu ứng dụng không kiểm tra và làm sạch dữ liệu đầu vào trước khi thực thi câu lệnh SQL, mã SQL độc hại sẽ được chạy trực tiếp trên cơ sở dữ liệu.

Ví dụ từ W3Schools:

**Ví dụ 1: Đăng nhập với SQL Injection**

Giả sử bạn có một form đăng nhập với hai trường: **username** và **password**. Câu lệnh SQL đơn giản có thể trông như sau:

SELECT \* FROM Users WHERE username = 'user' AND password = 'pass';

Kẻ tấn công có thể nhập vào trường username giá trị sau:

user' OR '1'='1

Và mật khẩu gì đó (hoặc để trống). Câu lệnh SQL sẽ trở thành:

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

Câu lệnh trên luôn trả về đúng và cho phép kẻ tấn công đăng nhập vào với bất kỳ mật khẩu nào.

**Ví dụ 2: Xóa tất cả dữ liệu**

Một kẻ tấn công có thể sử dụng câu lệnh SQL để xóa dữ liệu bằng cách thực hiện nhập vào trường:

'; DROP TABLE Users; --

Câu lệnh cuối cùng đưa ra sẽ trông như thế này:

SELECT \* FROM Users WHERE username = ''; DROP TABLE Users; -- ' AND password = '';

Câu lệnh này sẽ làm ứng dụng xóa bảng Users trong cơ sở dữ liệu.

Ứng dụng Kỹ thuật SQL Injection vào Trang Đăng Nhập

Để minh họa SQL Injection, dưới đây là ví dụ mã PHP của trang đăng nhập có lỗ hổng SQL Injection. Tuy nhiên, tôi không thể thực hiện hành động tấn công thực sự hoặc chụp ảnh màn hình, nhưng tôi sẽ đưa ra mô tả của cách mà bạn có thể thực hiện để thử nghiệm.

**Mô tả mã giả của Trang Đăng Nhập:**

<?php

// Kết nối tới cơ sở dữ liệu

$conn = new mysqli($servername, $username, $password, $dbname);

// Nhập thông tin đăng nhập từ người dùng

$user = $\_POST['username'];

$pass = $\_POST['password'];

// Tạo câu lệnh SQL không an toàn

$sql = "SELECT \* FROM Users WHERE username='$user' AND password='$pass'";

$result = $conn->query($sql);

if ($result->num\_rows > 0) {

echo "Đăng nhập thành công!";

} else {

echo "Sai tên đăng nhập hoặc mật khẩu.";

}

?>

**Thực hiện Kiểm Tra SQL Injection**:

**Gửi yêu cầu đăng nhập** với trường username như sau:

admin' OR '1'='1

và mật khẩu là bất kỳ (hoặc để trống).

**kết quả**:

* Nếu ứng dụng không có cơ chế bảo mật, bạn sẽ thấy thông báo "Đăng nhập thành công!", bất kể mật khẩu thực tế là gì.
* **Kiểm tra các câu lệnh khác**:

Bạn cũng có thể thử nghiệm các câu lệnh SQL Injection khác như:

admin' -- (để bỏ qua phần kiểm tra mật khẩu)

'; DROP TABLE Users; -- (để thử nghiệm việc xóa bảng).

**Cách Phòng Ngừa SQL Injection**

Để bảo mật ứng dụng của bạn chống lại SQL Injection, bạn nên áp dụng các biện pháp sau:

* Sử dụng câu lệnh chuẩn bị (prepared statements).
* Chạy các biến vào câu lệnh SQL an toàn.
* Sử dụng ORM (Object Relational Mapping) như PDO trong PHP.
* Kiểm tra và làm sạch dữ liệu đầu vào của người dùng.
* Sử dụng các công cụ phát hiện lỗ hổng bảo mật.

Đừng bao giờ thực hiện các thử nghiệm trên hệ thống thật mà không có sự cho phép rõ ràng từ quản trị viên hoặc chủ sở hữu hệ thống. Các thử nghiệm này phải được thực hiện trong môi trường an toàn và có sự giám sát.

1. 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 gì

Cookie là một đoạn dữ liệu lưu trữ nhỏ mà một máy chủ gửi đến trình duyệt của người dùng. Cookie được lưu trên máy tính của người dùng và có thể được sử dụng để lưu trữ thông tin viên (như thông tin đăng nhập, tùy chọn người dùng, nội dung giỏ hàng, v.v.) nhằm cải thiện trải nghiệm người dùng trên website. Cookie giúp website ghi nhớ các thông tin giữa các phiên truy cập.

### Các tham số trong setcookie()

Hàm setcookie() trong PHP được sử dụng để tạo cookie. Cú pháp cơ bản là:

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

#### Diễn giải các tham số:

1. **name**: Tên của cookie (bắt buộc).
2. **value**: Giá trị của cookie (bắt buộc). Đây có thể là bất kỳ chuỗi nào mà bạn muốn lưu.
3. **expire**: Thời gian sống của cookie (tính bằng giây). Nếu không được thiết lập, cookie sẽ hết hạn khi trình duyệt đóng.
4. **path**: Đường dẫn trên máy chủ mà cookie sẽ có sẵn. Mặc định là "/" (tất cả các đường dẫn trên trang).
5. **domain**: Tên miền mà cookie sẽ được gửi đến. Nếu không chỉ định, cookie sẽ chỉ được gửi đến 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 chỉ có thể được truy cập thông qua giao thức HTTP và không thể bị truy cập bởi JavaScript, tăng cường bảo mật.

### Cách lưu, lấy, xóa giá trị trong cookie

#### 1. Lưu giá trị trong cookie:

setcookie("username", "JohnDoe", time() + 3600, "/"); // Thời gian sống 1 giờ

#### 2. Lấy giá trị từ cookie:

Sau khi cookie được thiết lập, bạn có thể lấy giá trị của nó bằng cách sử dụng biến $\_COOKIE:

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

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

} else {

echo "Cookie không tồn tại.";

}

#### 3. Xóa cookie:

Để xóa một cookie, bạn có thể thiết lập thời gian hết hạn của cookie về một thời điểm 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 gì?

Session trong PHP là một phương pháp lưu trữ dữ liệu tạm thời trên máy chủ, cho phép bạn lưu trữ thông tin người dùng trong suốt phiên làm việc với ứng dụng web. Điều này có nghĩa là dữ liệu sẽ được giữ lại giữa các trang mà người dùng truy cập, tạo ra trải nghiệm liền mạch hơn. Sessions thích hợp cho việc lưu trữ thông tin nhạy cảm, như thông tin đăng nhập, mà không cần phải lưu trên thiết bị của người dùng (như một cookie).

### Cách sử dụng Session trong PHP

#### 1. Bắt đầu một Session

Trước khi bạn có thể lưu dữ liệu vào một session, bạn cần phải khởi động session bằng cách sử dụng hàm session\_start().

session\_start();

#### 2. Lưu giá trị trong Session

Sau khi khởi động session, bạn có thể lưu dữ liệu bằng cách sử dụng biến siêu toàn cục $\_SESSION. Ví dụ:

session\_start(); // Khởi động session

$\_SESSION["username"] = "JohnDoe"; // Lưu giá trị vào session

$\_SESSION["role"] = "admin"; // Lưu giá trị khác vào session

#### 3. Lấy giá trị từ Session

Để lấy giá trị đã lưu trong session, bạn cũng sử dụng biến $\_SESSION:

session\_start(); // Khởi động session

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

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

} else {

echo "Session không tồn tại.";

}

#### 4. Xóa giá trị trong Session

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

session\_start(); // Khởi động session

unset($\_SESSION["username"]); // Xóa giá trị username khỏi session

#### 5. Kết thúc một Session

Để kết thúc và xóa mọi thông tin liên quan đến session, bạn sử dụng session\_destroy():

session\_start(); // Khởi động session

session\_destroy(); // Kết thúc session

1. Bạn hãy so sánh Cookie và Session.

| **Tiêu chí** | **Cookie** | **Session** |
| --- | --- | --- |
| **Lưu trữ** | Lưu trữ trên máy khách (trình duyệt của người dùng). | Lưu trữ trên máy chủ. |
| **Kích thước** | Giới hạn kích thước thường khoảng 4KB trên mỗi cookie. | Kích thước lớn hơn, tùy thuộc vào khả năng của máy chủ. |
| **Thời gian sống** | Thời gian sống có thể được chỉ định; có thể tồn tại lâu dài cho đến khi người dùng xóa. | Thời gian sống thường chỉ tồn tại trong suốt phiên truy cập, tự động kết thúc khi trình duyệt đóng hoặc timeout. |
| **An ninh** | Dễ bị tấn công (chẳng hạn như đánh cắp thông tin). | An toàn hơn cho thông tin nhạy cảm, vì dữ liệu không được lưu trên máy khách. |
| **Truy cập** | Có thể truy cập qua JavaScript. | Không thể truy cập trực tiếp qua JavaScript (để bảo mật). |
| **Tình trạng người dùng** | Thích hợp cho việc lưu trữ thông tin không nhạy cảm, chẳng hạn như tùy chọn người dùng. | Thích hợp cho việc lưu trữ thông tin nhạy cảm hơn, như thông tin đăng nhập. |
| **Sử dụng** | Thích hợp cho việc nhớ thông tin giữa các lần truy cập (ví dụ: thông tin giỏ hàng). | Thích hợp cho việc theo dõi thông tin trong suốt một phiên làm việc (ví dụ: thông tin người dùng đã đăng nhập). |
| **Quản lý** | Dễ dàng thiết lập nhưng có thể yêu cầu người dùng chấp nhận cookie. | Dễ dàng quản lý trên máy chủ, mà không cần sự chấp nhận từ người dùng. |

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.

Log.php

A screenshot of a computer

Description automatically generated

homepage.php

Thoat.php

A screenshot of a computer

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. sua\_mk.php

sua\_mk.php:

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

Luu\_mk.php:

A screenshot of a computer program

Description automatically generated

8.Bạn đọc trong[[4]](#footnote-3) 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 tinh type="file" để người dùng có thể chọn tệp tin.

+ Trên phía máy chủ, sử dụng các hàm và thủ tục PHP để xữ 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

tài 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 tài lên tệp tin.

- Sử dụng các câu lệnh điều kiện để kiểm tra lỗi và xử lý mỗi trường hợp

lỗi.

Để thực hiện việc tải lên tệp tin trong PHP, ta cần làm theo các bước sau:

Bước 1: Định nghĩa biểu mẫu HTML cho việc tải lên tệp tin:

A screenshot of a computer program

Description automatically generated

Bước 2: tạo tập tin PHP uploads.php để xử lí việc tải lên:

A screenshot of a computer program

Description automatically generated

Trong vi dụ trên, chúng ta sử dụng biến siêu toàn cục $\_FILES' để truy cập

thông tin về tep tin đã tai lên. $\_FILES["fileToUpload"]["name"] là tên tệp tin

gốc, `$\_FILES["fileToUpload"]["size"]` là kích thước tệp tin,

`$\_FILES["fileToUpload"]["tmp\_name"]` là đường dẫn tạm thời của tệp tin tải lên.

Bước 3: Kiểm tra và xữ lý lỗi khi tải lên tệp tin:

Sử dụng các câu lệnh điều kiện để kiểm tra và xữ lý lỗi trong tập tin

upload.php. Ví dụ: kiểm tra xem tệp tin đã tồn tại, kiểm tra kích thước và định

dạng tệp tin, kiểm tra xem việc di chuyển tệp tin đã thành công hay không.

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).

Dưới đây là mô tà về chức năng của một số hàm quan trọng và các tham số

tương ứng trong hàm đó, dựa trên tài liệu từ trang web W3Schools và ví dụ về đọc

tệp tin CSV (read-csv.php):

file\_exists(Sfilename)': 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(Sfilename) : Đọc nội dung của một tệp tin và tra 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.

- 'Smode': 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 trõ tệp tin (file pointer) hoặc false nếu có lỗi xảy ra.

fgets($file) và fgetcsv($file) : Đọc một dòng từ tệp tin.

- 'Sfile': Con trõ tệp tin trả về từ hàm fopen()'.

- fgets(Sfile): 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 màng chứa

các giá trị được phân tách bằng dấu phẩy (dùng cho đọc tệp tin CSV).

Ví dụ về đọc tệp tin CSV (read-csv.php) sử dụng các hàm trên:

A screenshot of a computer program

Description automatically generated

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.

customer\_data.csv:

A screen shot of a computer

Description automatically generated

* + Tham khảo upload-img.php để thiết kế giao diện upload file

upload-csv.php:

A screenshot of a computer program

Description automatically generated

* + 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.

upload-csv-processing.php

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Chú ý:**

* Các bạn nộp file word: Quy tắc đặt tên file: **<mssv>-<hoten>-<bai><stt\_bai thực hành>.docx** nộp lên Classroom (VD: **B123456-NguyenVanA-bai1.docx**), kèm với các file khác được yêu cầu như phần câu hỏi đã nêu. **Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file zip**. File zip đặt tên như file word.
* Mỗi câu các bạn trả lời bằng hình hoặc dạng text tùy vào yêu cầu của câu hỏi và **TRẢ LỜI THEO ĐÚNG THỨ TỰ CÂU HỎI**. Nếu câu nào không trả lời được các bạn cứ để số thứ tự câu hỏi và bỏ trống phần trả lời.
* Các câu trả lời có tham khảo trên Internet phải trích dẫn link/nguồn.
* *Vi phạm 1 trong các điều sau đây bài thực hành sẽ bị 0 điểm:*
  + Đặt tên KHÔNG ĐÚNG quy tắc được yêu cầu.
  + Bài không đủ các thành phần (word, code+data (nếu có),...) đã được yêu cầu.
  + Bài không thực hiện đúng yêu cầu “**Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file .zip”**
  + Bị phát hiện copy, sao chép từ các bạn khác
  + Phần trả lời không ghi rõ trả lời cho câu nào
  + Thứ tự câu trả lời không đúng thứ tự câu hỏi

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. <https://www.w3schools.com/php/php_sessions.asp> [↑](#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. https://stackoverflow.com/questions/6654351/check-file-uploaded-is-in-csv-format [↑](#footnote-ref-6)