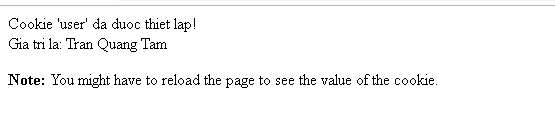
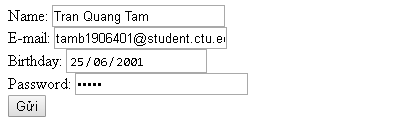
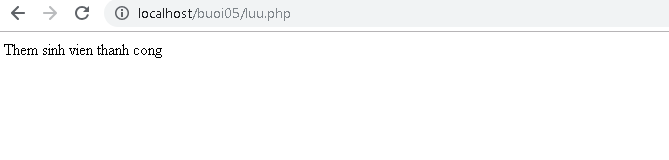
**TRẦN QUANG TÂM – B1906401**

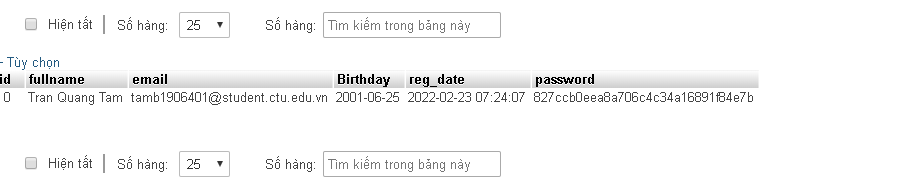
**Câu 1:**

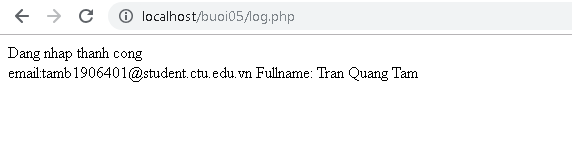
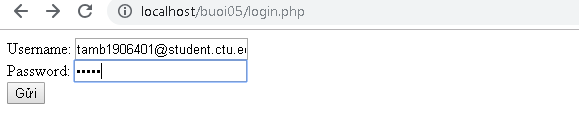


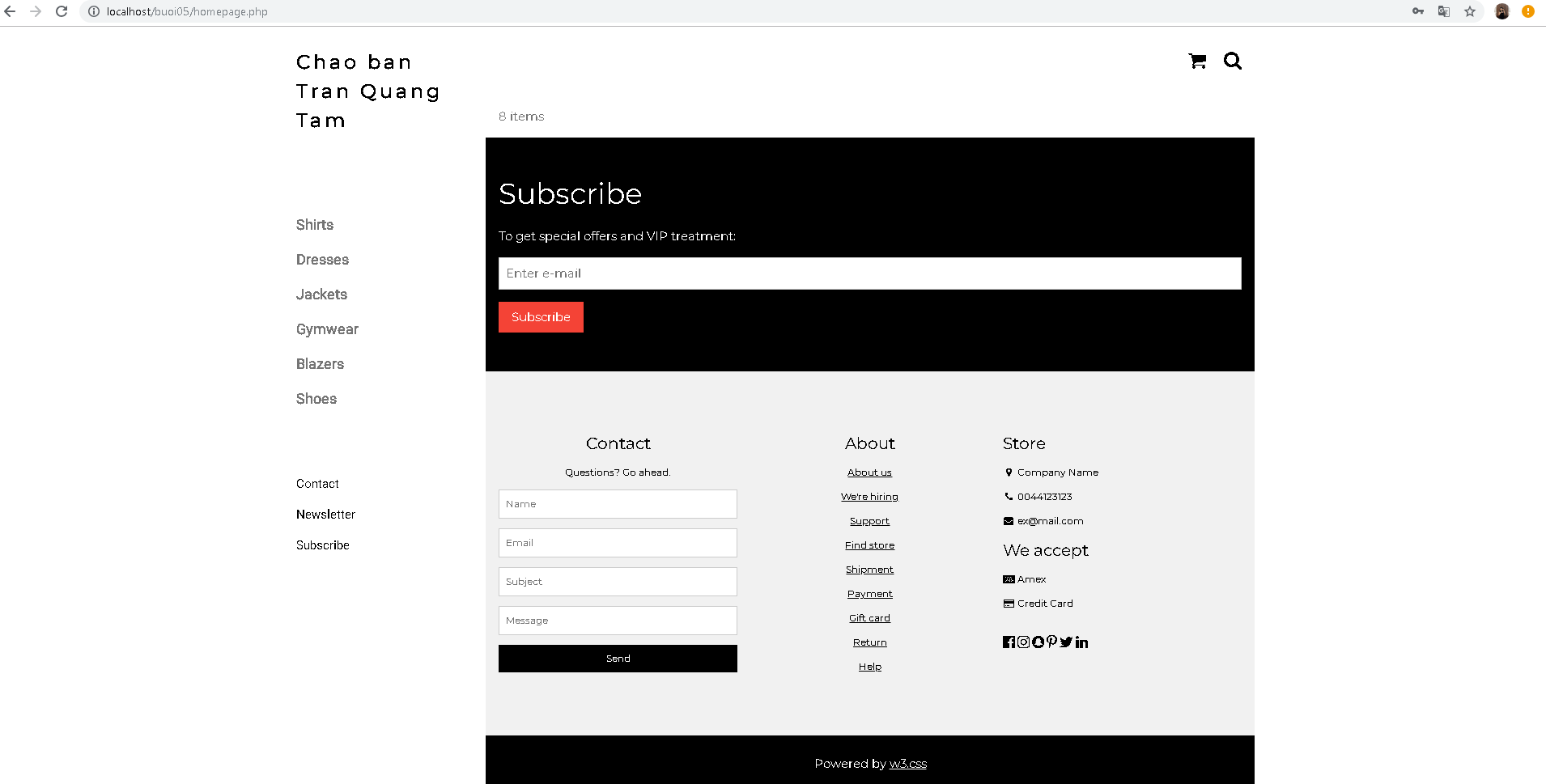


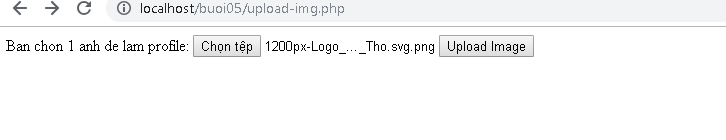


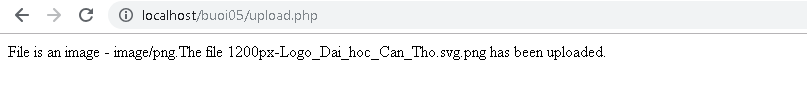


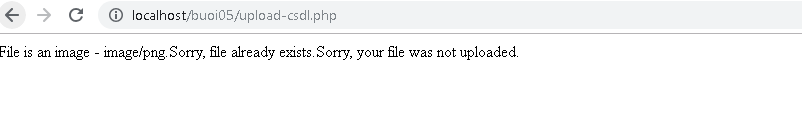


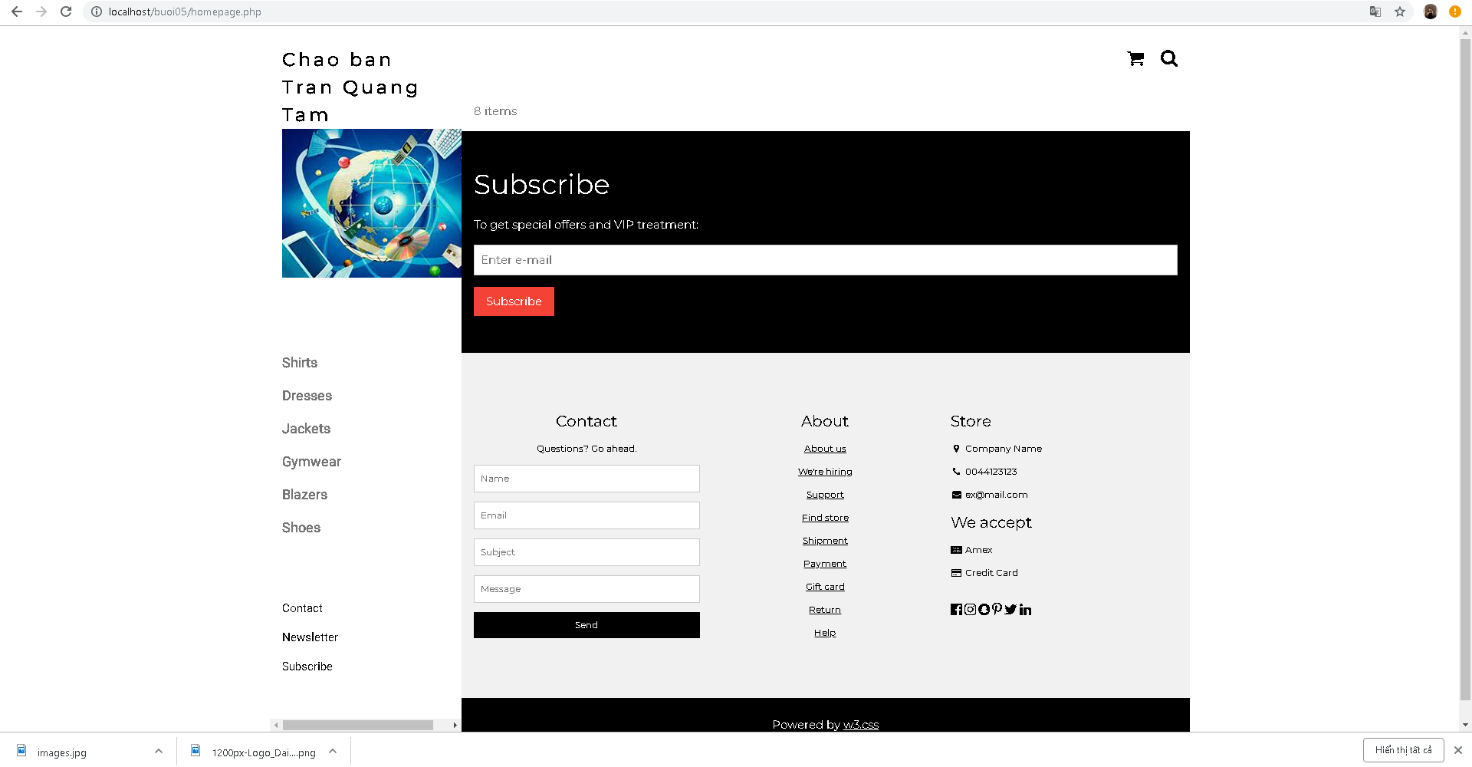


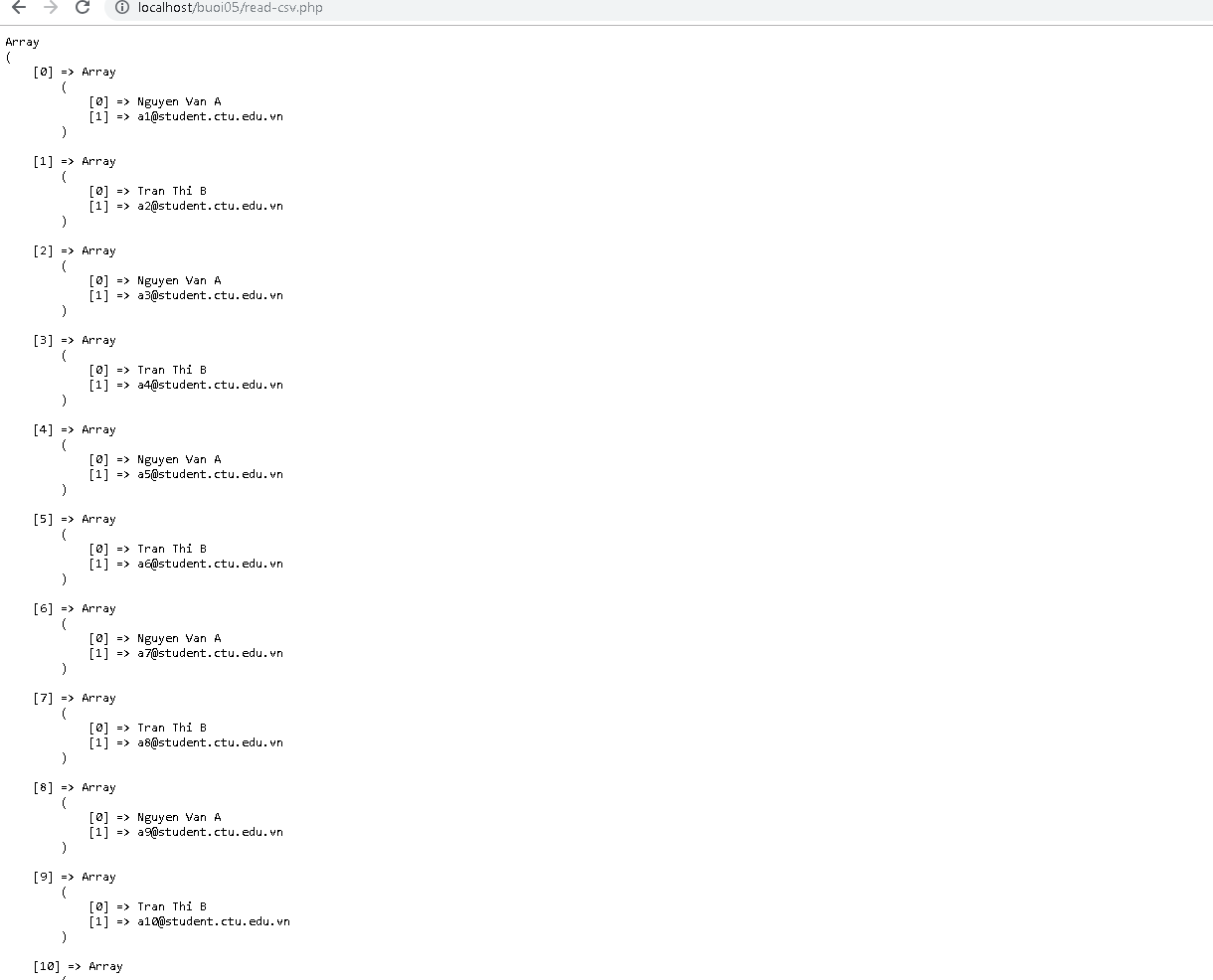












**Câu 2:**

- SQL Injection sử dụng những lỗ hổng trong các kênh đầu vào (input) của website để nhắm mục tiêu vào cơ sở dữ liệu nằm trong phần phụ trợ của ứng dụng web, nơi lưu giữ những thông tin nhạy cảm và có giá trị nhất. Chúng có thể được kẻ tấn công sử dụng để ăn cắp hoặc xáo trộn dữ liệu, cản trở sự hoạt động của các ứng dụng, và, trong trường hợp xấu nhất, nó có thể chiếm được quyền truy cập quản trị vào máy chủ cơ sở dữ liệu.

**Câu 3:**

- Cookies là các tệp được trang web người dùng truy cập tạo ra. Cookie giúp trải nghiệm trực tuyến của bạn dễ dàng hơn bằng cách lưu thông tin duyệt web. Với Cookies, các trang web có thể duy trì trạng thái đăng nhập của bạn, ghi nhớ tùy chọn trang web và cung cấp nội dung phù hợp với vị trí của người dùng.

- Khởi tạo Cookie theo cú pháp : setcookie($name, $value, $expire, $path, $domain);

Trong đó:

+$name là tên của cookie các bạn muốn tạo.

+$value là giá trị của cookie đó.

+$expire là thời gian sống của cookie.

+$path là đường dẫn lưu trữ cookie.

+$domain là tên của domain.

Ví dụ: setcookie(‘user’,’TranQuang Tam’, time() + 600);

- Lấy giá trị của Cookie: $name là tên của cookie các bạn muốn lấy.

Ví dụ: $\_COOKIE['user'];

- Xóa Cookie: Để xóa cookie thì các bạn chỉ cần xét cho thời gia sống của nó nhỏ hơn thời điểm hiện tại.

Ví dụ: setcookie('user', 'TranQuangTam', time() - 100);

**Câu 4:**

- Session trong PHP dùng để lưu trữ thông tin như thông tin người dùng, thông tin cấu hình,... Mà nó chỉ có thể tồn tại đến khi chúng ta đóng trình duyệt. Và mỗi client sẽ có một SessionID khác nhau nên hoàn toàn không có thể xảy ra việc xung đột session. Ứng dụng thực tế trong việc sử dụng session thường làm chức năng đăng nhập, giỏ hàng,...

-Đầu tiên các bạn muốn khởi tạo được session trong PHP thì bắt buộc các bạn phải khai báo session\_start(); ở đầu mỗi file. Hàm này có tác dụng khai báo cho server biết phiên làm việc có sử dụng session.

Ví dụ:  
<?php

session\_start();

// code

?>

- Lưu trữ Session: Cũng giống như GET và POST, session cũng được lưu trong một biến toàn cục $\_SESSION, vì vậy khi muốn thêm sửa xóa session chúng ta cũng phải thao tác trên biến này.

+ Thêm hoặc xóa Session với cú pháp: $\_SESSION['name'] = 'value';

+Trong đó:

name là tên của session.

value là giá trị của session, giá trị này có thể là một chuỗi,mảng,số,...

Ví dụ thêm Session:  
<?php

$\_SESSION['user'] = 'admin';

// hoặc đối với mảng

$array = [5, 8, 6, 4, 7, 5];

$\_SESSION['user'] = $array;

Ví dụ sửa Session:

<?php

// khởi tạo session user có giá trị là TranQuangTam

$\_SESSION['user'] = 'TranQuangTam';

// sửa giá trị của session user thành Toidicode-er

$\_SESSION['user'] = 'Toidicode-er';

+ Lấy giá trị Session theo cú pháp: $\_SESSION['name'];

Trong đó: name là tên của session muốn lấy.( Chú ý kiểm tra chặt chẽ xem session đó có tồn tại hay không trước khi lấy)

Ví dụ:

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

echo $\_SESSION['user'];

}

+ Xóa Session theo cú pháp: unset($bien);

Trong đó: $bien là thư muốn xóa

Ví dụ: Xóa Session user ở ví dụ trên  
unset($\_SESSION['user']);

+Xóa tất cả Session dùng theo cú pháp:

session\_destroy();

//hoac

unset($\_SESSION);

**Câu 5:**

Sự khác nhau giữa COOKIE và SESSION

- Cookie:

Cookie được lưu trữ trên trình duyệt của người dùng.

Dữ liệu cookie được lưu trữ ở phía client.

Dữ liệu cookie dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client.

Dữ liệu cookie có sẵn trong trình duyệt đến khi expired.

- Session:

Session không được lưu trữ trên trình duyệt  
Dữ liệu Session được lưu trữ ở phía server.

Dữ liệu Session không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ.  
Sau khi đóng trình duyệt sẽ hết phiên làm việc (session).

**Câu 8:**

**Code tạo biểu mẫu html**

<!DOCTYPE html>

<html>

<body>

<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>

</body>

</html>

- Một số quy tắc cần tuân theo cho biểu mẫu HTML ở trên:

+Đảm bảo rằng biểu mẫu sử dụng method = "post"

+Biểu mẫu cũng cần thuộc tính sau: enctype = "multiart / form-data". Nó chỉ định loại nội +dung sẽ sử dụng khi gửi biểu mẫu

+Nếu không có các yêu cầu trên, quá trình tải lên tệp sẽ không hoạt động.

- Những điều khác cần lưu ý:

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

- Các hàm và thủ tục

$ 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 giữ phần mở rộng tệp của tệp (chữ thường)

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

**Câu 9:**

Tệp () đọc một tệp thành một mảng.

Mỗi phần tử mảng chứa một dòng từ tệp, với ký tự dòng mới vẫn được đính kèm.

Cú pháp:

file(filename, flag, context)

Trong đó:  
filename: Tên tệp bắt buộc

flag: FILE\_USE\_INCLUDE\_PATH - Tìm kiếm tệp trong đường dẫn bao gồm (trong php.ini)

FILE\_IGNORE\_NEW\_LINES - Bỏ qua dòng mới ở cuối mỗi phần tử mảng

FILE\_SKIP\_EMPTY\_LINES - Bỏ qua các dòng trống trong tệp

Context: Chỉ định ngữ cảnh của xử lý tệp. Ngữ cảnh là một tập hợp các tùy chọn có thể sửa đổi hành vi của một luồng. Có thể được bỏ qua bằng cách sử dụng NULL.