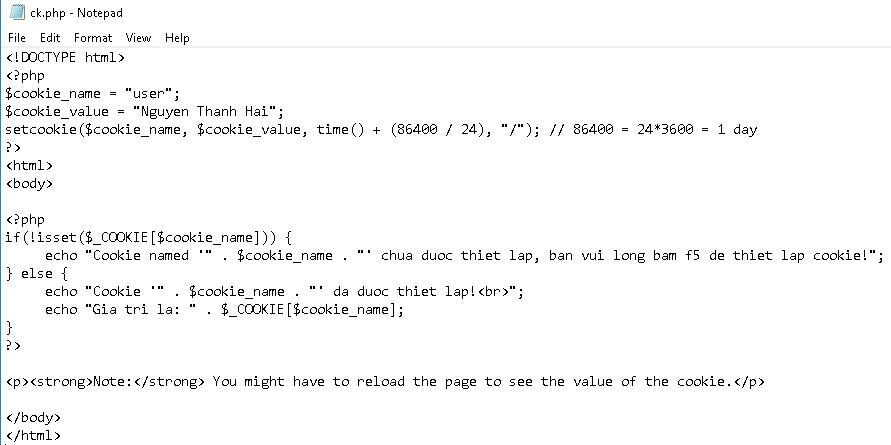
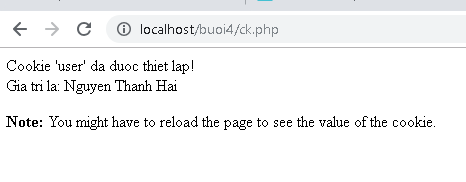
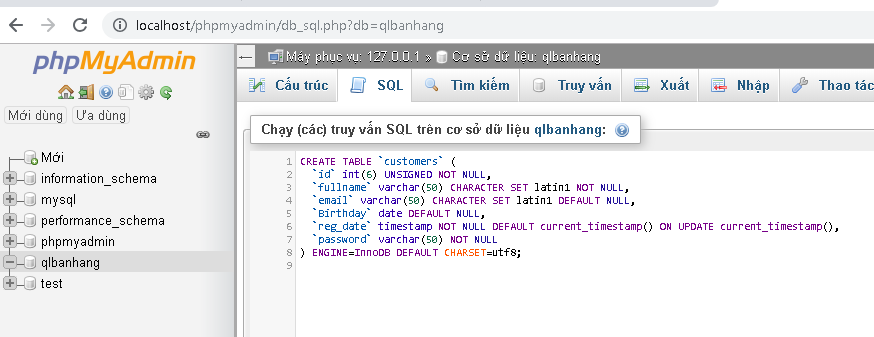
Câu 1.Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.

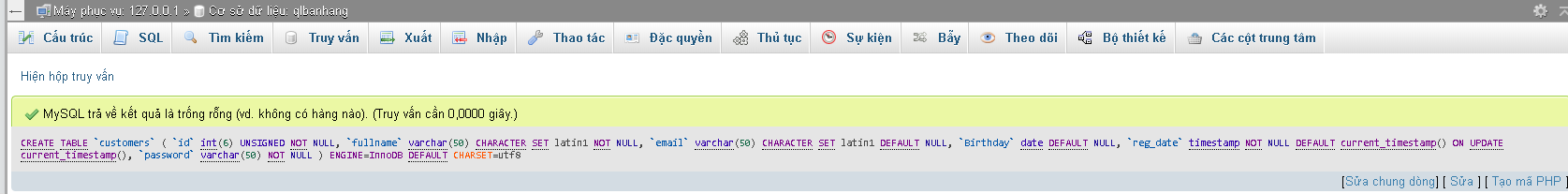
ck.php



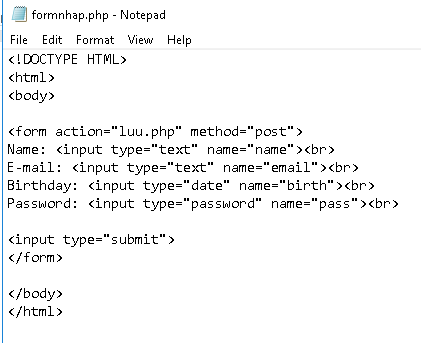


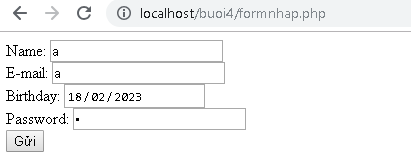
Tạo csdl qlbanhang



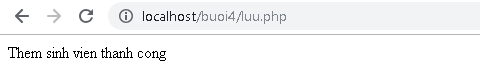
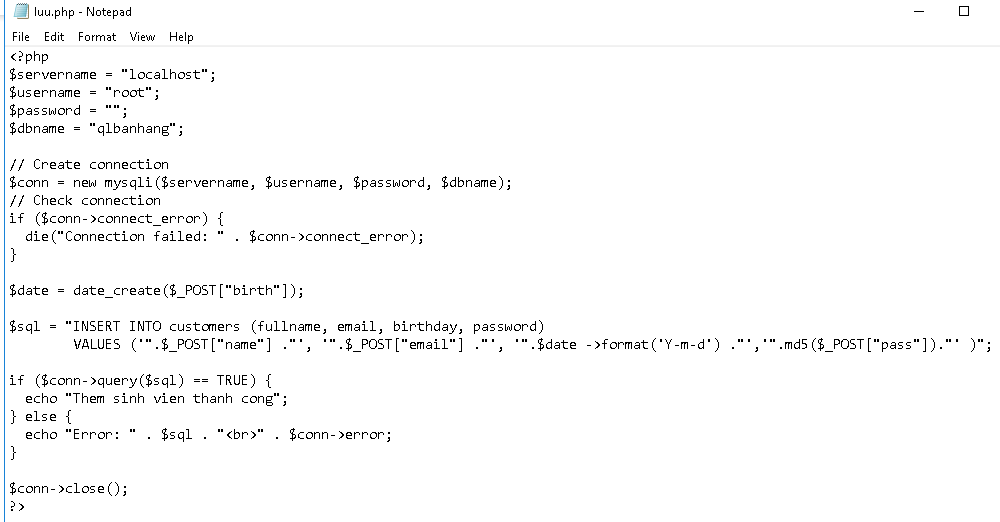


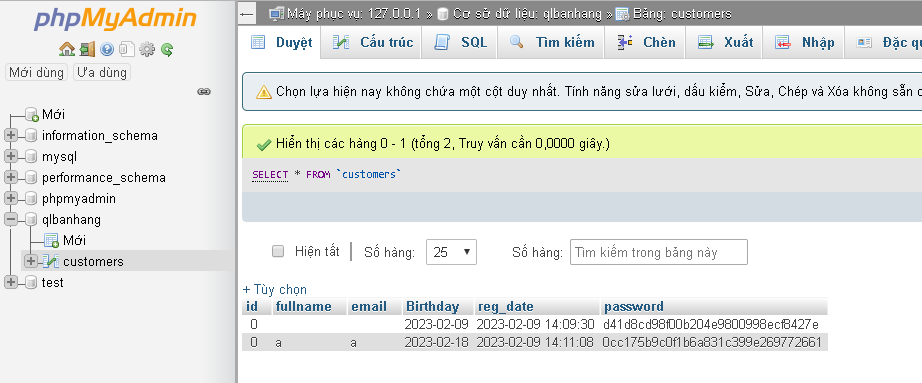
Formnhap.php



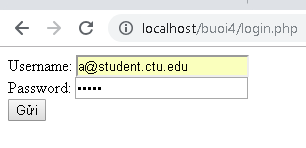
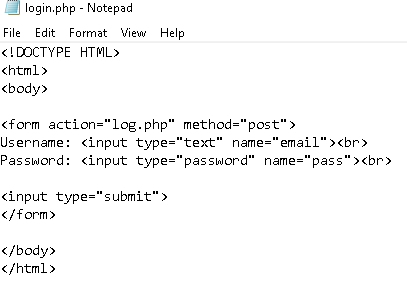


Luu.php

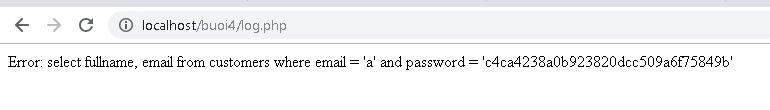
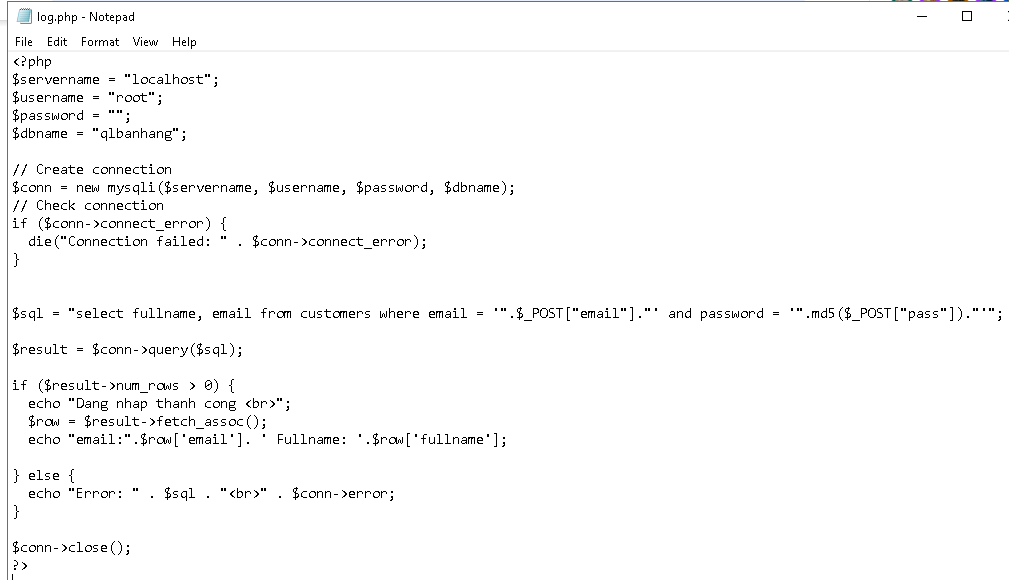




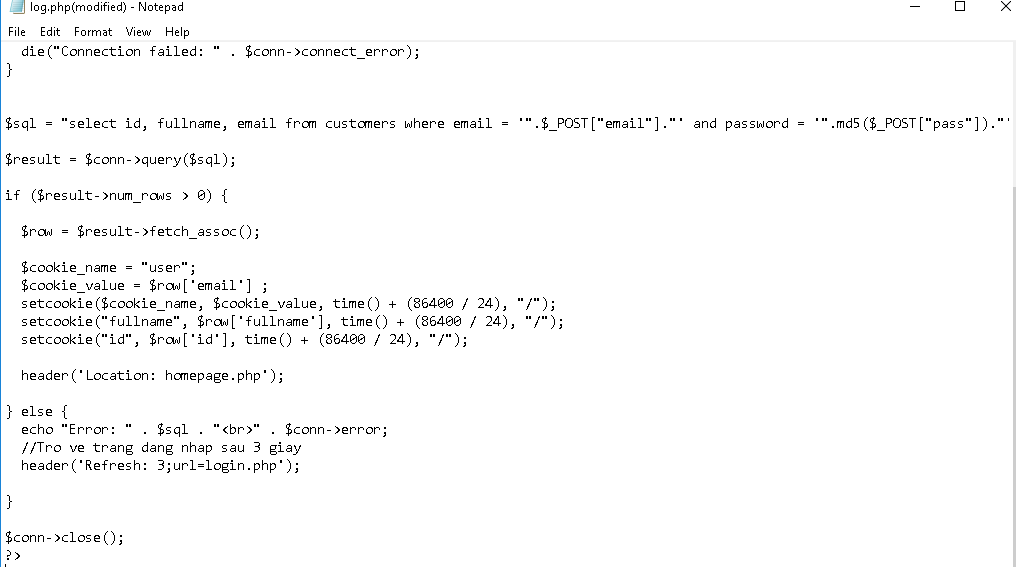
Login.php



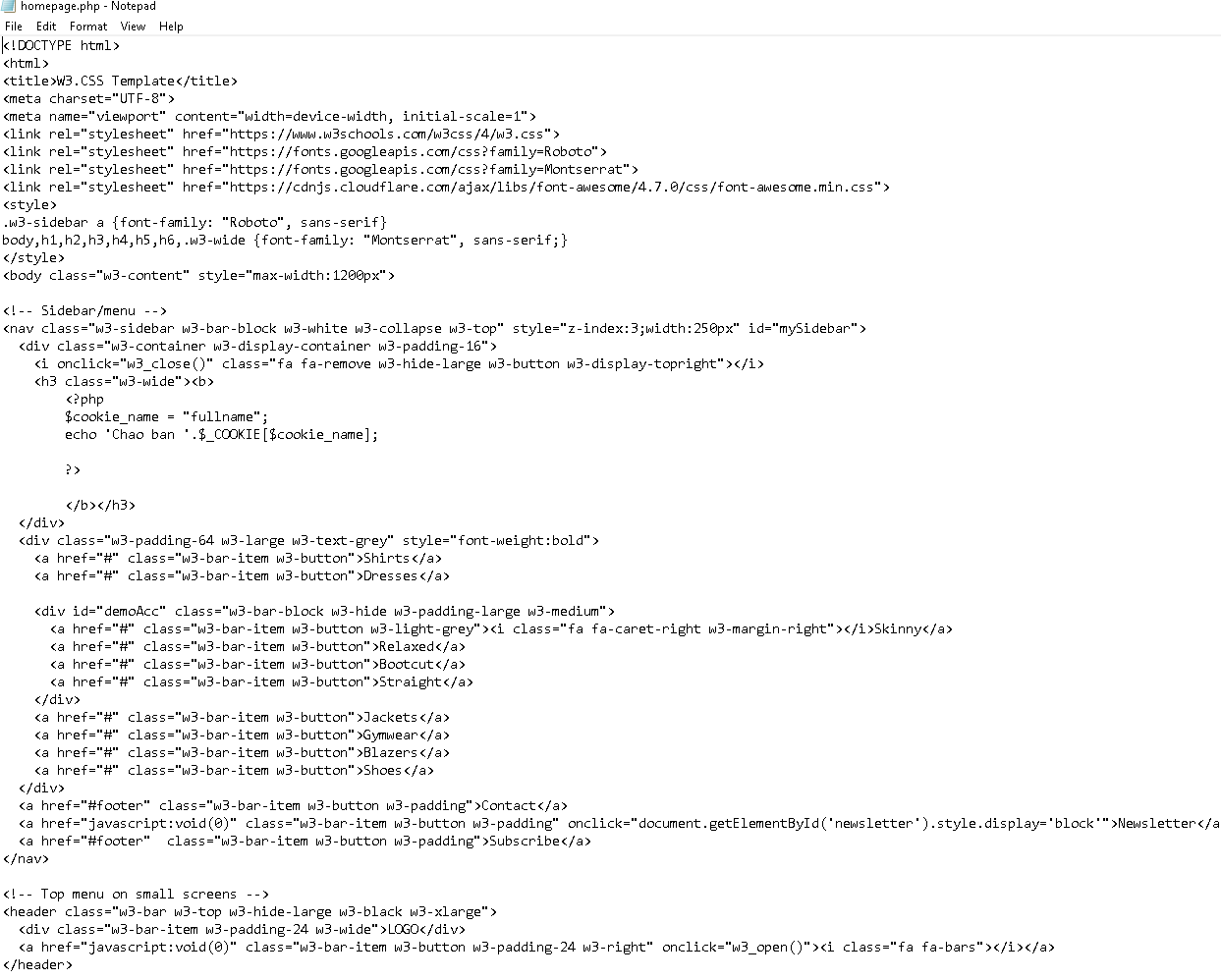
Log.php

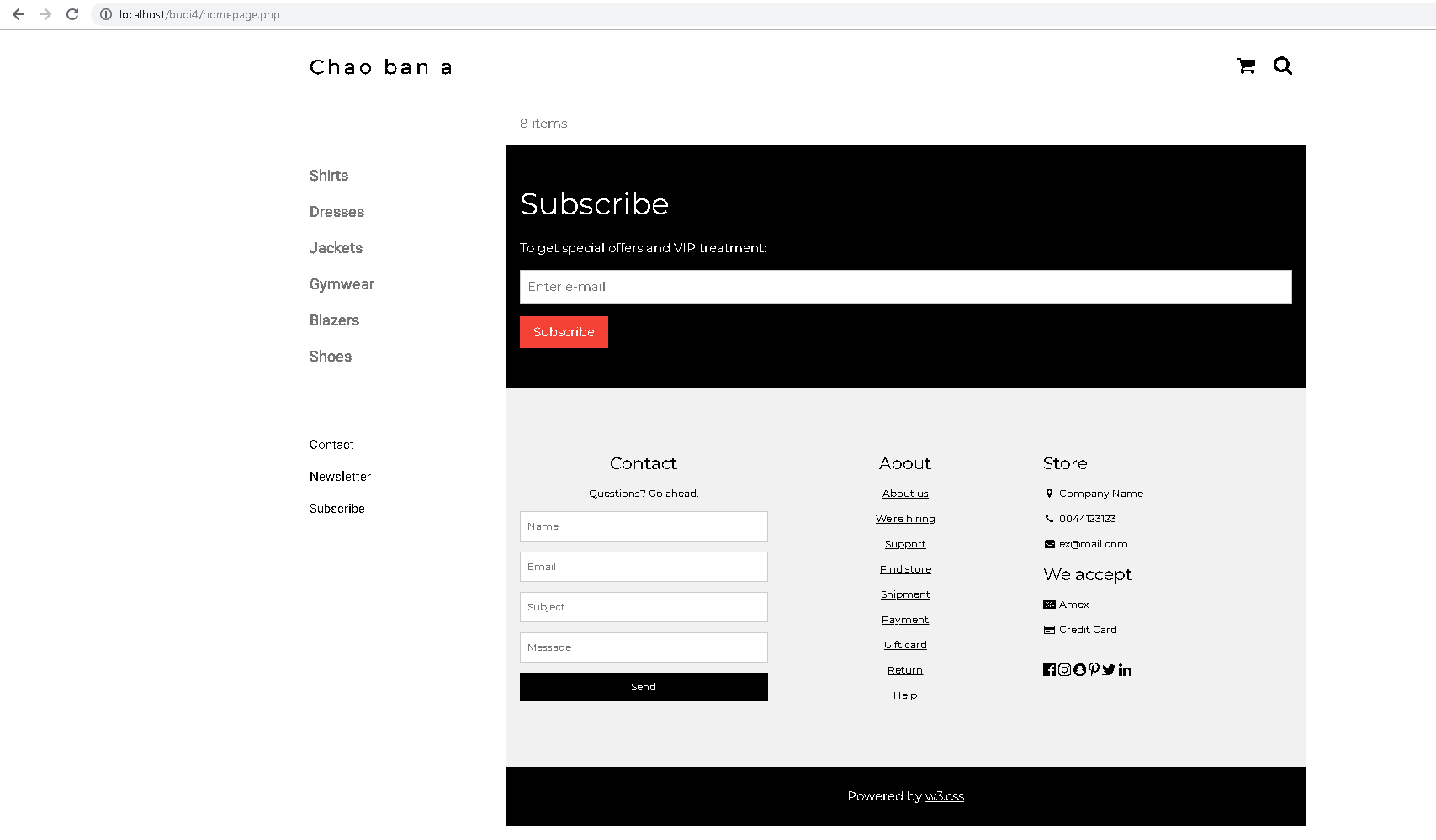


Log.php(modified)



Homepage.php





Upload-img.php

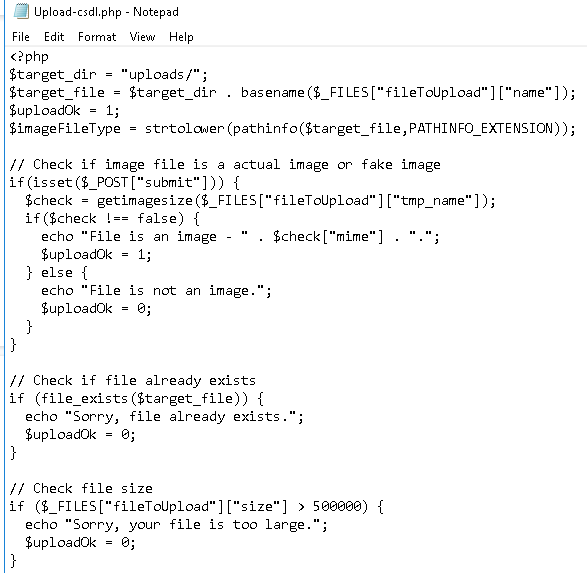


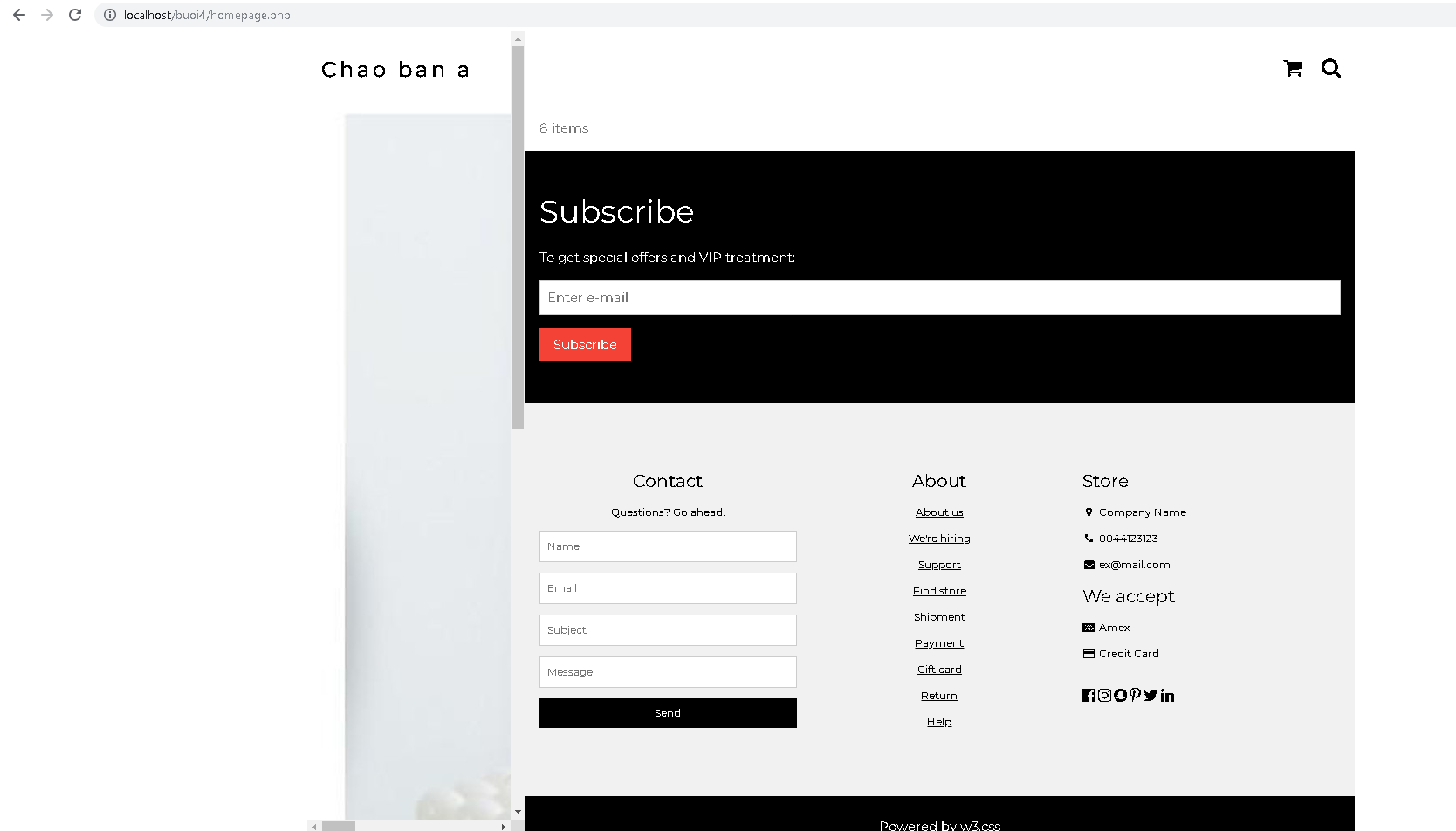


upload.php

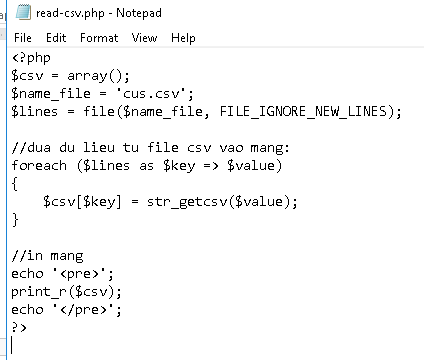


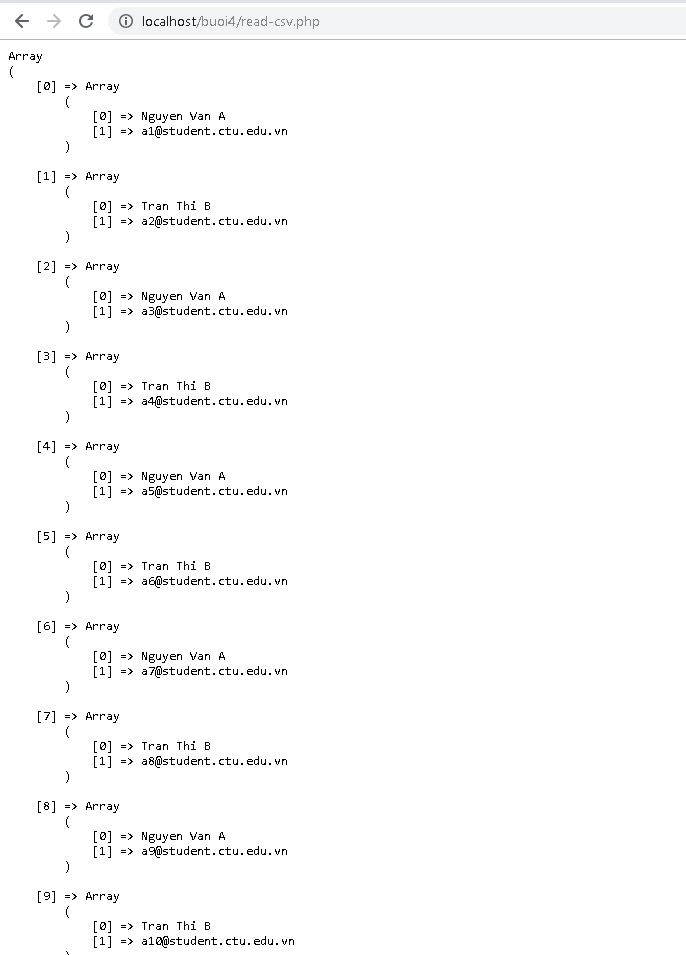
Upload-csdl.php





Read-csv.php





Câu 2.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 . 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, một trong những kỹ thuật hack web phổ biến, vị trí của mã độc hại trong câu lệnh SQL thông qua đầu vào trang web.

Nếu không có gì ngăn người dùng nhập đầu vào "sai", người dùng có thể nhập một số đầu vào "thông minh" như sau:

Tên người dùng:

105 HOẶC 1=1

Sau đó, câu lệnh SQL sẽ trông như thế này: SELECT \* FROM Users WHERE UserId = 105 OR 1=1;

Câu SQL ở trên hợp lệ và sẽ trả về TẤT CẢ các hàng từ bảng "Người dùng", vì OR 1=1 luôn là TRUE.

Ví dụ trên có nguy hiểm không? Nếu bảng "Người dùng" chứa tên và mật khẩu thì sao?

Câu lệnh SQL ở trên giống như thế này:

SELECT UserId, Name, Password FROM Users WHERE UserId = 105 or 1=1;

Câu 3. Dựa vào link, 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 xác định người dùng. Cookie là một tệp nhỏ mà máy chủ nhúng vào máy tính của người dùng. Mỗi khi cùng một máy tính yêu cầu một trang bằng trình duyệt, nó cũng sẽ gửi cookie. Với PHP, bạn có thể tạo và truy xuất các giá trị cookie.

Một cookie được tạo bằng setcookie()chức năng.

cú pháp:setcookie(*name, value, expire, path, domain, secure, httponly*);

+name:là tên cookie, tên này được sử dụng để truy cập vào biến $cookie để lấy thông tin.

+value:là giá trị gán cho cookie

+expire:thời điểm hết hạn(hủy) của cookie, thời gian dạng unix timestamp, có thể dùng hàm time() để lấy unix timestamp.Sau thời điểm này cookie bị hủy lưu tại trình duyệt.Nếu thiết lập lại là 0 thì cookie hủy khi trình đống duyệt.

+path:đường dẫn mà cookie có hiệu lực.Sử dụng / cho biết cookie có hiệu lực ở cấp độ domain - mọi URL thuộc domain, nếu thiết lập là /foo/ thì Cookie chỉ có hiệu lực với các URL dạng /foo/ và các con của nó như /foo/abc/... Nếu để giá trị mặc định "" thì có hiệu lực theo đường dẫn hiện tại

+domain:tên miền

+secure:= true nếu sử dụng HTTPS để truyền dữ liệu (mặc định false)

+httponly: mặc định là false, nếu true thì chỉ cho phép cookie với giao thức HTTP

Cách lưu cookie:

<!DOCTYPE html>

<?php

setcookie('username', 'thehalfheart', time() + 3600);

?>

<html>

<head>

<title></title>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

</head>

<body>

</body>

</html>

Cách lấy giá trị cookie:

if (isset($\_COOKIE['username']))

{

echo $\_COOKIE['username'];

}

Cách xóa: để xóa cookie, hãy sử dụng setcookie()chức năng có ngày hết hạn trong quá khứ setcookie("username", "", time()-3600);

Câu 4.Dựa vào link, bạn hãy cho biết Session dùng để làm gì. Cách lưu, lấy, xóa giá trị trong Session

Session trong PHP được dùng để lưu trữ thông tin của người dùng hoặc là lưu trữ tùy chọn cấu hình hệ thống cho người dùng. Đặc biệt mỗi client sẽ có một ID session khác nhau nên việc thông tin Session ở Client A bị ảnh hưởng qua Client B là điều không thể. Thông thường chúng ta sử dụng Session để lưu thông tin đăng nhập, giỏ hàng hoặc những dữ liệu mang tính chất tạm thời và mỗi client sẽ có dữ liệu khác nhau.

**Cách lưu**:Session được lưu trữ trong biến toàn cục $\_SESSION, vì thể để lưu thêm dữ liệu Session hay là thay đổi dữ liệu của Session thì ta sẽ thao tác trên biến đó. Lưu ý là trước khi dùng phép lấy giá trị Session phải kiểm tra session đó có tồn tại không rồi hãy lấy.Để lưu một giá trị mới vào Session ta dùng cú pháp như sau:$\_SESSION['session\_name'] = $session\_value

**Cách lấy**:để lấy giá trị Session ta dùng cú pháp sau:$tenbien$\_SESSION['session\_name'].Và lưu ý trước khi bạn lấy giá trị Session bạn nên kiểm tra nó có tồn tại không rồi hãy lấy.

**Cách xóa:**Session đều lưu trữ trong biến $\_SESSION nên để xóa nó chỉ việc dùng hàm unset($\_SESSION['session\_name']), trong đó hàm unset dùng để giải phóng một biến ra khỏi bộ nhớ.Nếu bạn muốn xóa hết tất cả các Session thì ta dùng hàm session\_destroy().

Câu 5.Bạn hãy so sánh Cookie và Session.

Session và Cookie dùng để lưu trữ dữ liệu tạm thời và từ đó hệ thống có thể đưa ra những quyết định về quyền hạn hoặc là những tùy chọn riêng tư. Cả hai tuy có chung chức năng là lưu trữ dữ liệu nhưng bên trong nó lại khác nhau. Session dùng để lưu trữ dữ liệu trên Server và đồng thời nó sẽ có một đoạn code dữ liệu được lưu trữ ở client (cookie). Còn Cookie thì lưu trữ dữ liệu trên máy Client.

Câu 6.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.

Câu 7. 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.

Câu 8. Bạn đọc trong link4 để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.

Trước tiên, hãy đảm bảo rằng PHP được định cấu hình để cho phép tải tệp lên.

Trong tệp "php.ini" của bạn, tìm kiếm lệnh file\_uploads và đặt thành Bật:file\_uploads = On

Tiếp theo, tạo một biểu mẫu HTML cho phép người dùng chọn tệp hình ảnh họ muốn tải lên:

<!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 đối với biểu mẫu HTML ở trên:

* Đảm bảo rằng biểu mẫu sử dụng method="post"
* Biểu mẫu cũng cần thuộc tính sau: enctype="multipart/form-data". Nó chỉ định loại nội dung nào sẽ được sử dụng khi gửi biểu mẫu

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

Những điều khác cần chú ý:

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

Biểu mẫu trên gửi dữ liệu đến một tệp có tên "upload.php", mà chúng ta sẽ tạo tiếp theo.

Tệp "upload.php" chứa mã để tải tệp lên:

<?php

$target\_dir = "uploads/";

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

$uploadOk = 1;

$imageFileType = strtolower(pathinfo($target\_file,PATHINFO\_EXTENSION));

// Check if image file is a actual image or fake image

if(isset($\_POST["submit"])) {

$check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);

if($check !== false) {

echo "File is an image - " . $check["mime"] . ".";

$uploadOk = 1;

} else {

echo "File is not an image.";

$uploadOk = 0;

}

}

?>

Tập lệnh PHP đã giải thích:

* $ 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 sử dụng sau)
* $imageFileType giữ phần mở rộng tệp của tệp (ở dạng chữ thường)
* Tiếp theo, kiểm tra xem tệp hình ảnh là hình ảnh thật hay hình ảnh giả mạo

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.

Kiểm tra xem tệp có tồn tại chưa đầu tiên sẽ kiểm tra xem tệp đã tồn tại trong thư mục "tải lên" chưa. Nếu có, một thông báo lỗi sẽ hiển thị và $uploadOk được đặt thành 0:

// Check if file already exists

if (file\_exists($target\_file)) {

echo "Sorry, file already exists.";

$uploadOk = 0;

}

Trường nhập tệp trong biểu mẫu HTML của chúng tôi ở trên được đặt tên là "fileToUpload".

Bây giờ, chúng tôi muốn kiểm tra kích thước của tệp. Nếu tệp lớn hơn 500KB, thông báo lỗi sẽ hiển thị và $uploadOk được đặt thành 0:

// Check file size

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

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

$uploadOk = 0;

}

Mã bên dưới chỉ cho phép người dùng tải lên các tệp JPG, JPEG, PNG và GIF. Tất cả các loại tệp khác đưa ra thông báo lỗi trước khi đặt $uploadOk thành 0:

// Allow certain file formats

if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"

&& $imageFileType != "gif" ) {

echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";

$uploadOk = 0;

}

Tệp "upload.php" hoàn chỉnh bây giờ trông như thế này:

<?php

$target\_dir = "uploads/";

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

$uploadOk = 1;

$imageFileType = strtolower(pathinfo($target\_file,PATHINFO\_EXTENSION));

// Check if image file is a actual image or fake image

if(isset($\_POST["submit"])) {

$check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);

if($check !== false) {

echo "File is an image - " . $check["mime"] . ".";

$uploadOk = 1;

} else {

echo "File is not an image.";

$uploadOk = 0;

}

}

// Check if file already exists

if (file\_exists($target\_file)) {

echo "Sorry, file already exists.";

$uploadOk = 0;

}

// Check file size

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

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

$uploadOk = 0;

}

// Allow certain file formats

if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"

&& $imageFileType != "gif" ) {

echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";

$uploadOk = 0;

}

// Check if $uploadOk is set to 0 by an error

if ($uploadOk == 0) {

echo "Sorry, your file was not uploaded.";

// if everything is ok, try to upload file

} else {

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

echo "The file ". htmlspecialchars( basename( $\_FILES["fileToUpload"]["name"])). " has been uploaded.";

} else {

echo "Sorry, there was an error uploading your file.";

}

}

?>

trích dẫn link: <https://www.w3schools.com/php/php_file_upload.asp>

Câu 9. Bạn hãy đọc link5, mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).

Hệ thống tập tin PHP: các chức năng của hệ thống tập tin cho phép bạn truy cập và thao tác với hệ thống tập tin.

File() đọc một tập tin 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*)

Giá trị tham số:

| **Parameter** | **Description** |
| --- | --- |
| *filename* | Yêu cầu. Chỉ định đường dẫn đến tệp để đọc |
| *flag* | Không bắt buộc. có thể 1 hoặc nhiều constants:   * FILE\_USE\_INCLUDE\_PATH - Search for the file in the include\_path (in php.ini) * FILE\_IGNORE\_NEW\_LINES - Skip the newline at the end of each array element * FILE\_SKIP\_EMPTY\_LINES - Skip empty lines in the file |
| *context* | Optional. Specifies the context of the file handle. Context is a set of options that can modify the behavior of a stream. Can be skipped by using NULL. |

trích dẫn link: <https://www.w3schools.com/php/func_filesystem_file.asp>

Câu 10. 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