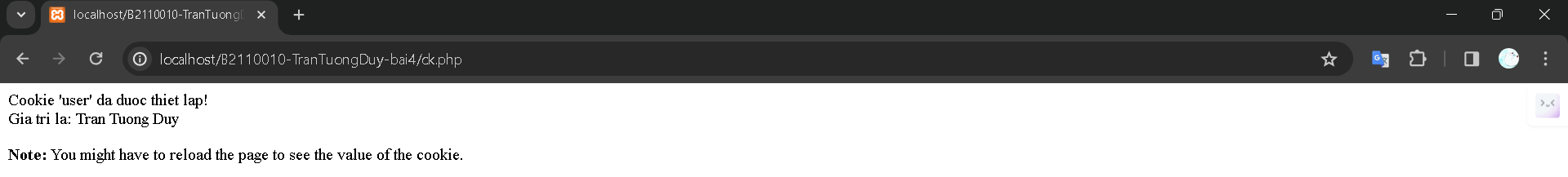
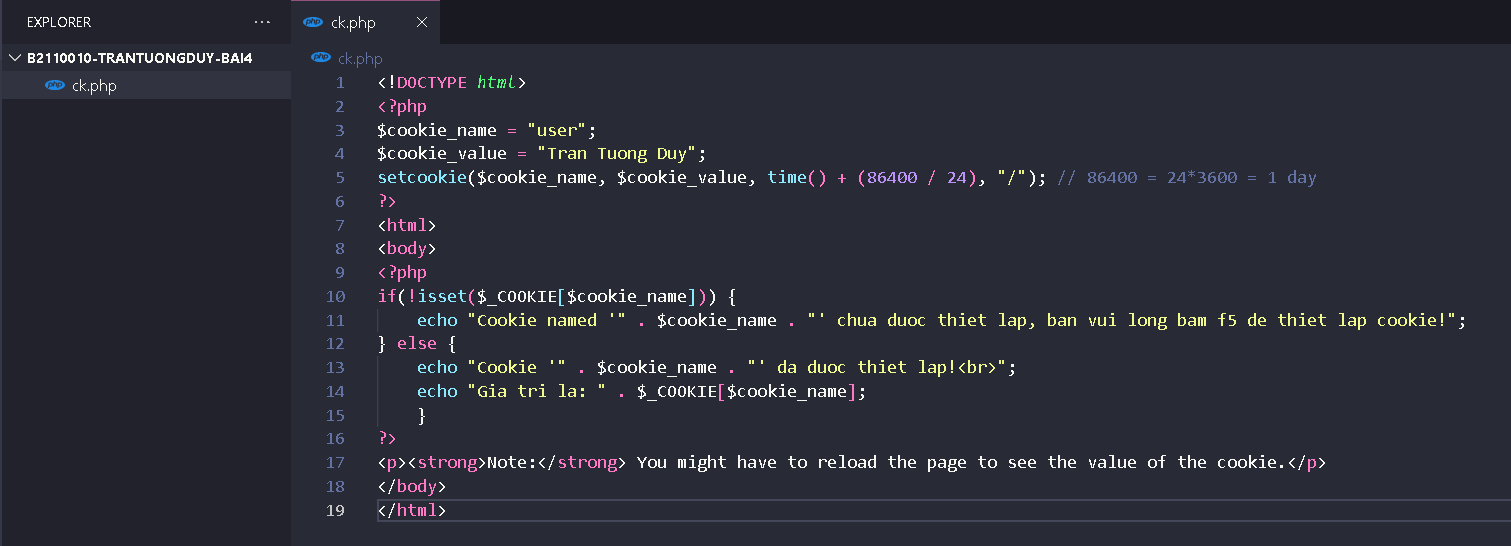
**BÀI THỰC HÀNH 4**

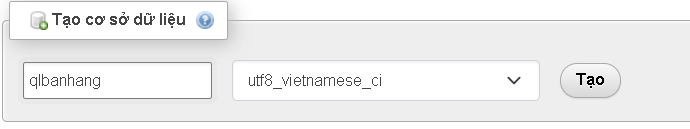
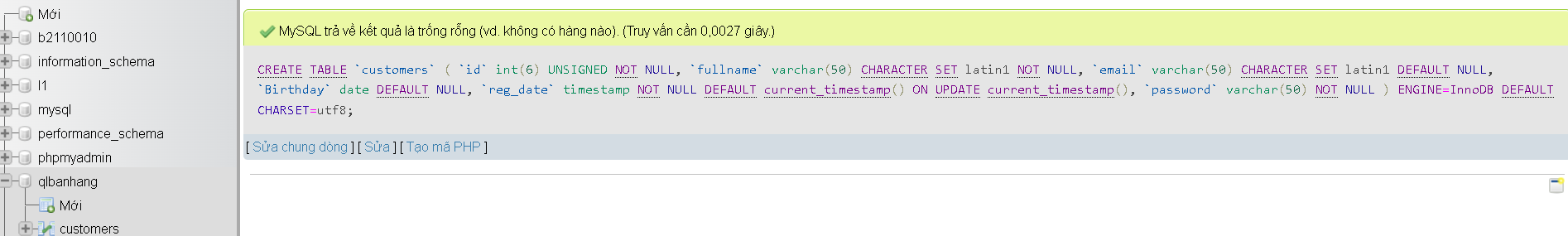
1. Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.

Tạo Cookies (Tập tin ck.php)

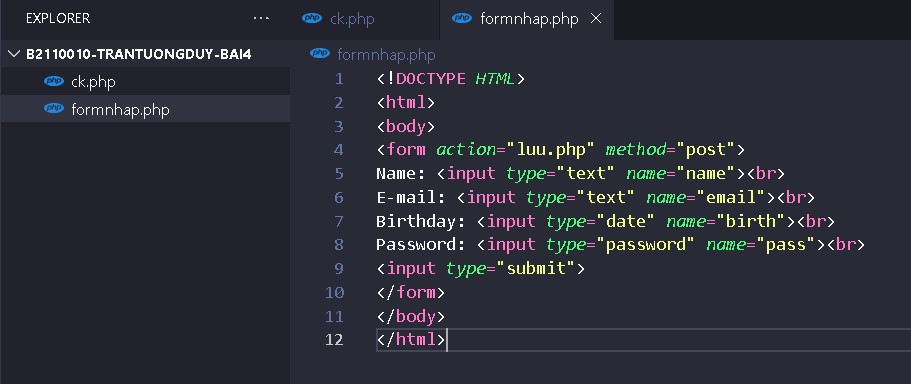


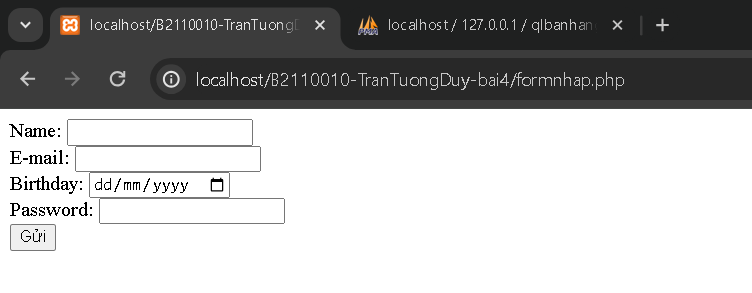
Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa

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

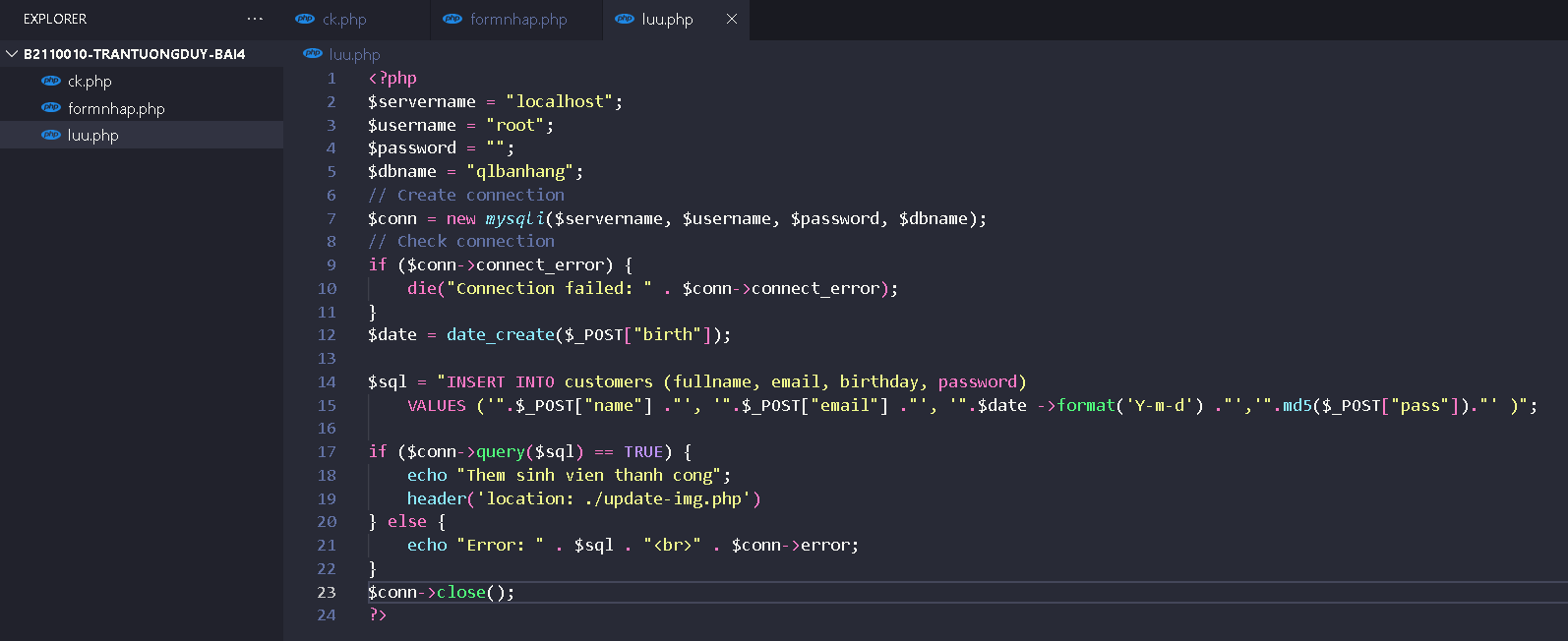
 

Tạo formnhap.php

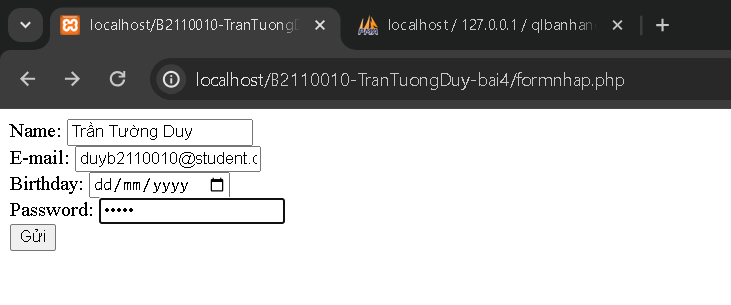


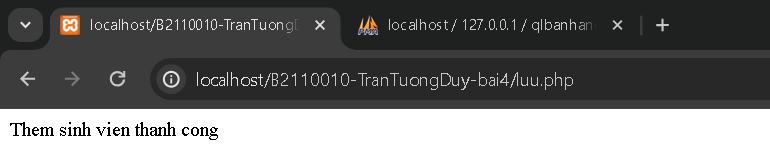
 Và luu.php, chú ý lúc này chúng ta sẽ mã hóa mật khẩu với giải thuật md5 để bảo mật mật khẩu:

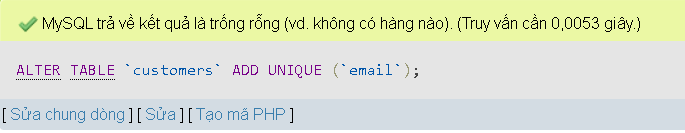
luu.php



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:



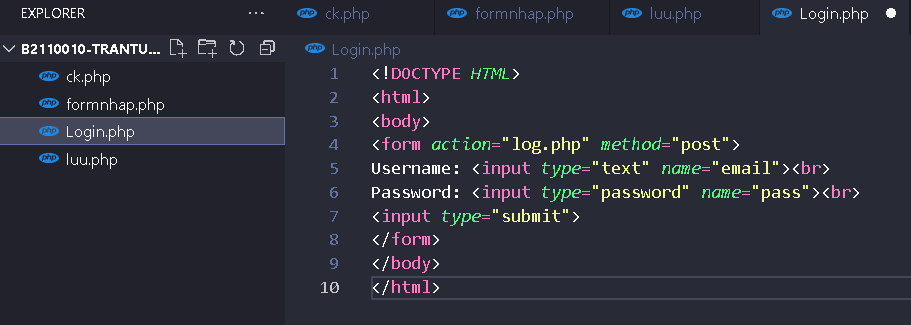
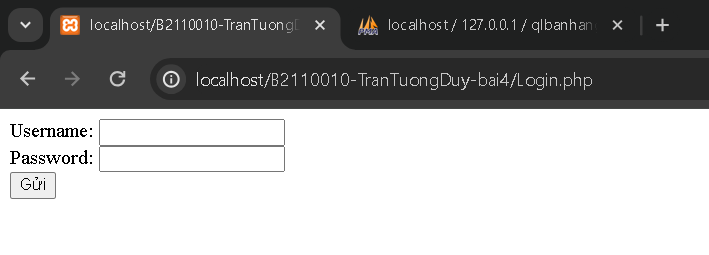
  Trong trường hợp chúng ta muốn dùng email là username để đăng nhập ta nên đảm bảo 2 student không có email trùng nhau. Để đảm bảo điều này ta có thể chọn email là unique key (vì ta đã chọn id la primary key, và 1 bảng không thể có 2 khóa chính)



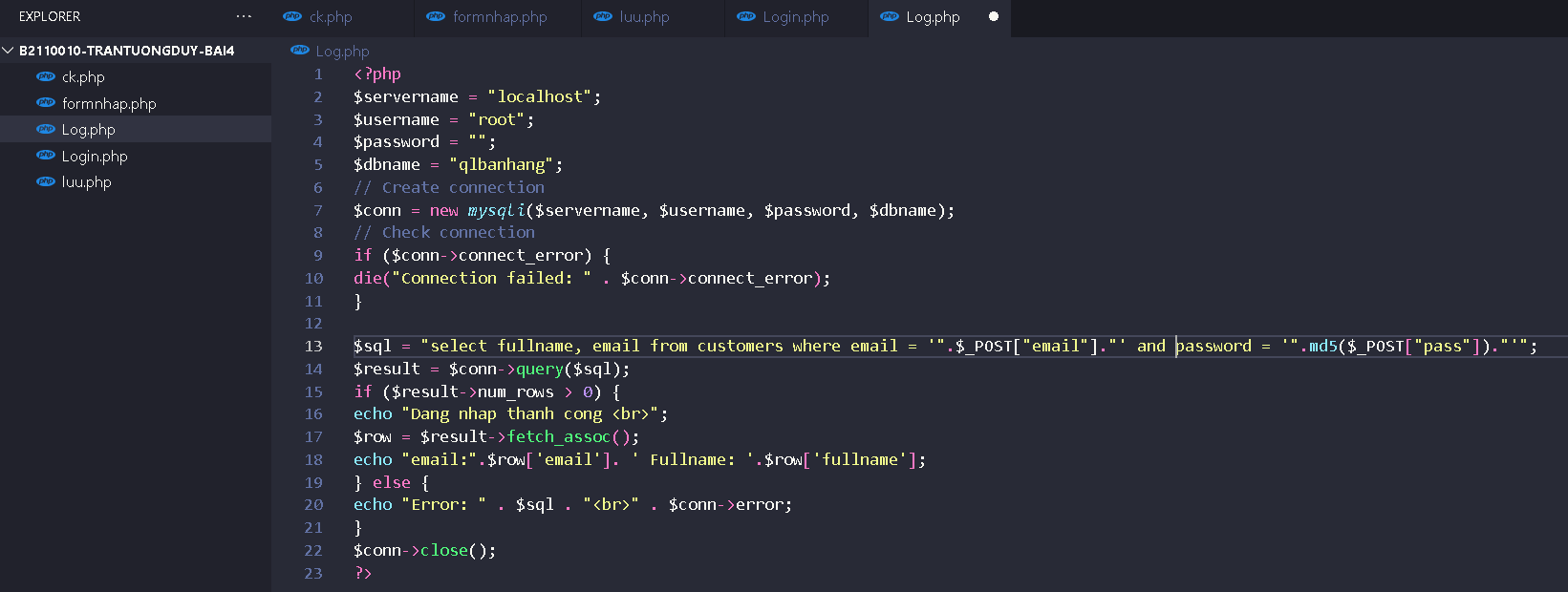
## Tạo form đăng nhập

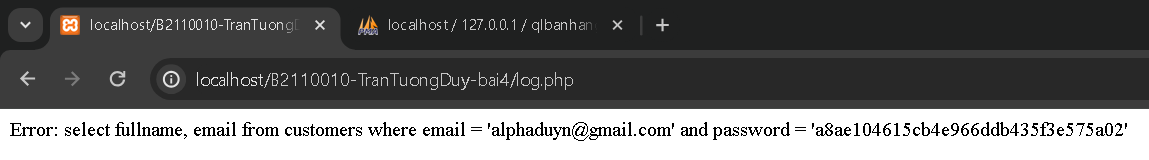
Sau khi tạo người dùng xong, ta có thể chạy form đăng nhập với username là email và password lấy từ CSDL

Login.php

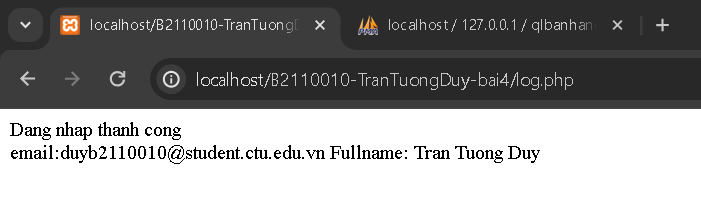
Log.php

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



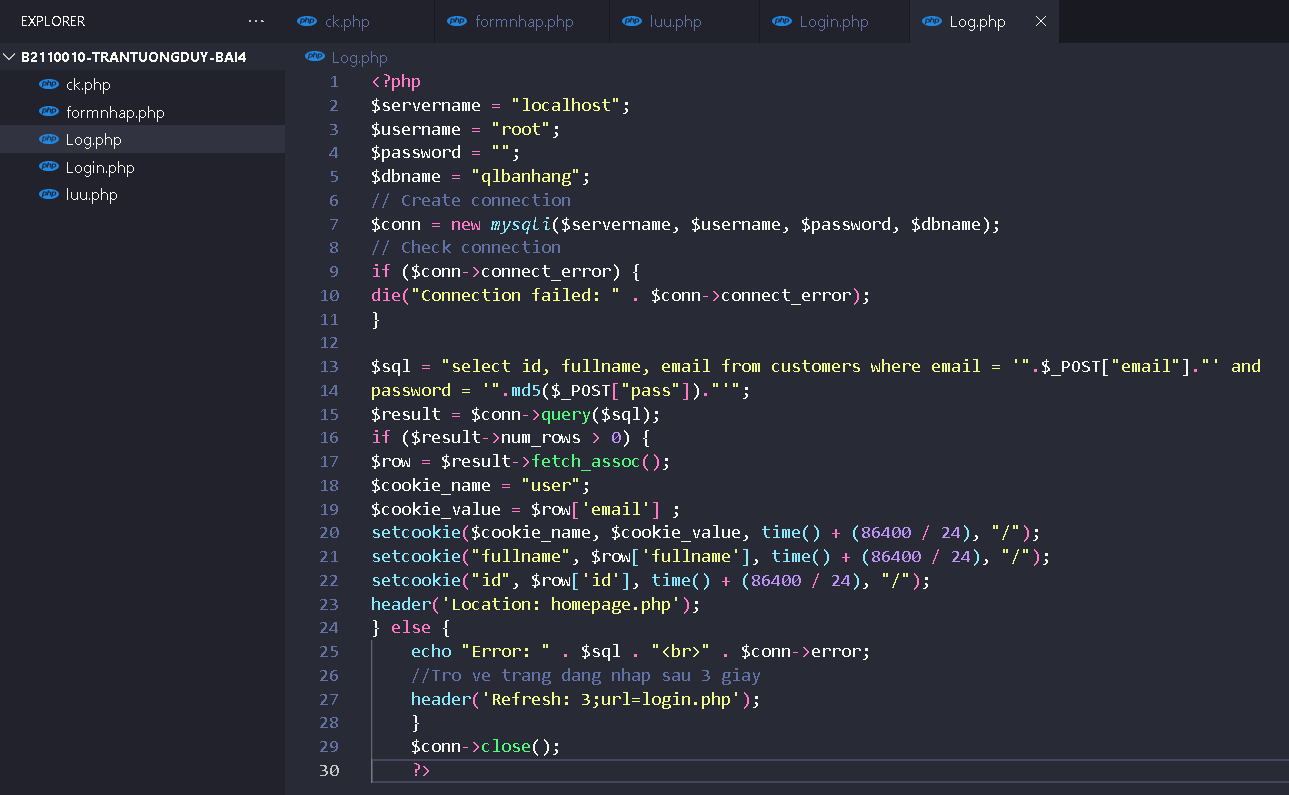
Mật khẩu đã cho không khớp với mã đã băm

Khi ta nhập đúng

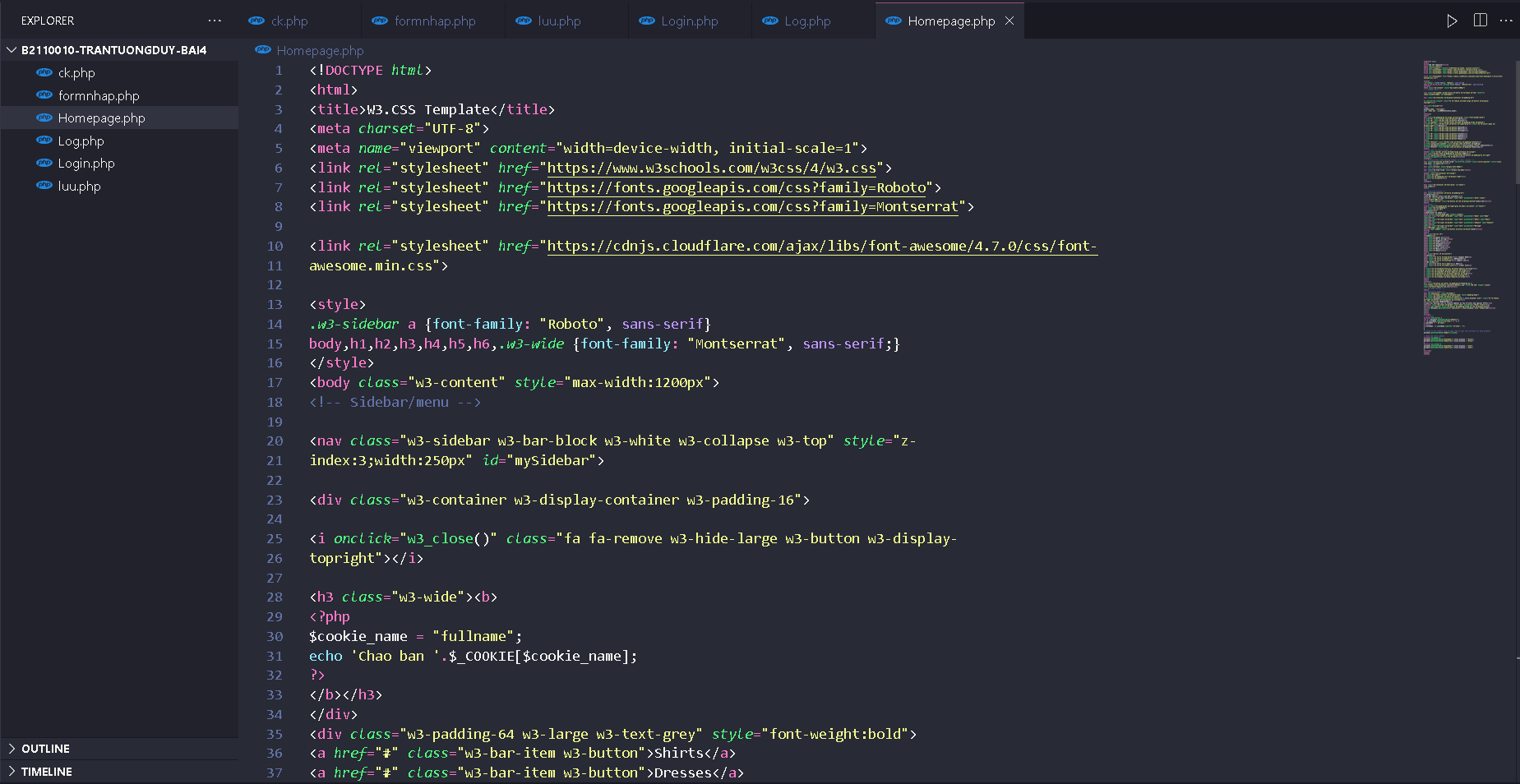


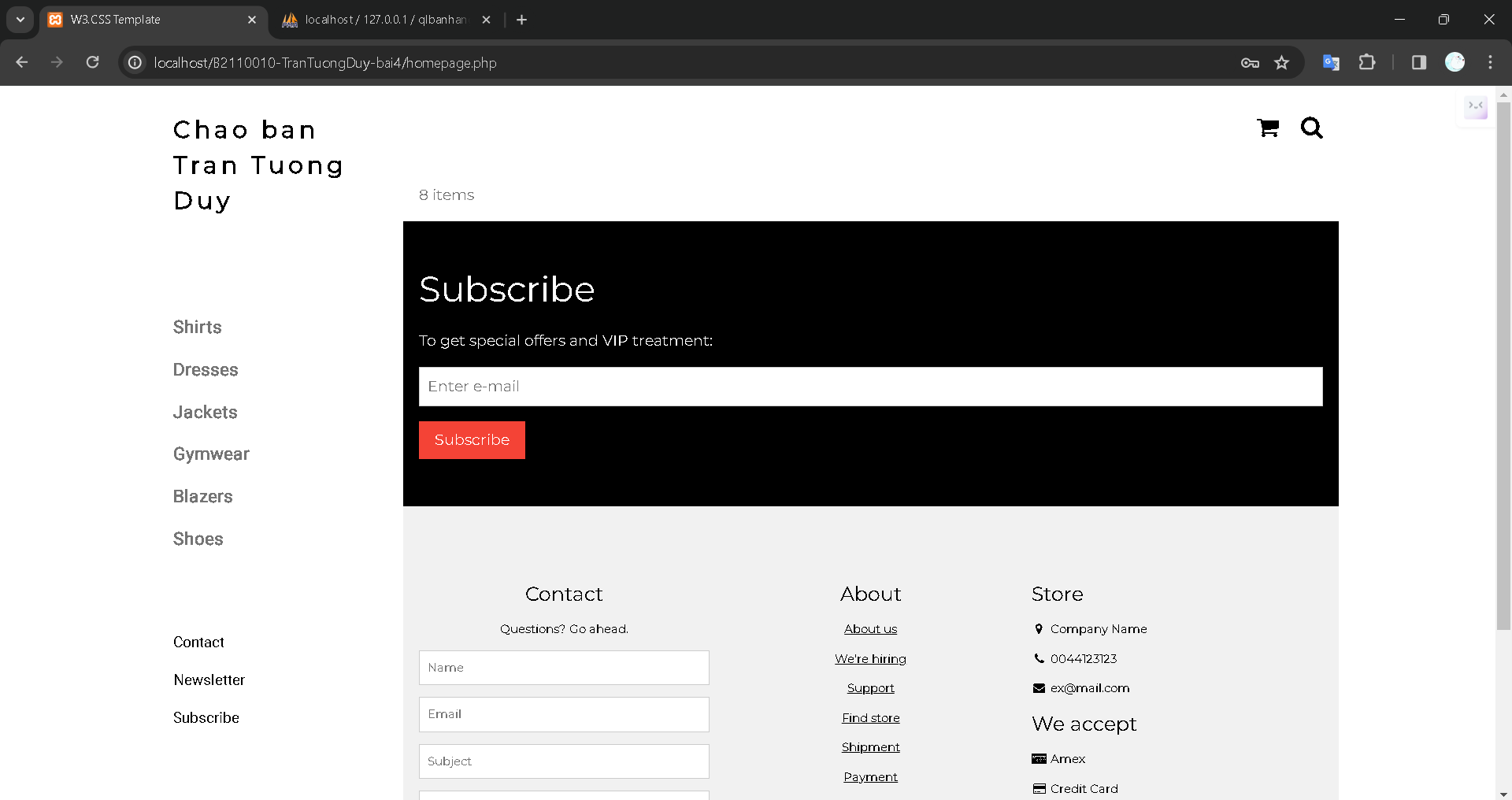
## Dùng Cookies lưu lại tên đăng nhập, email và điều hướng

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



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

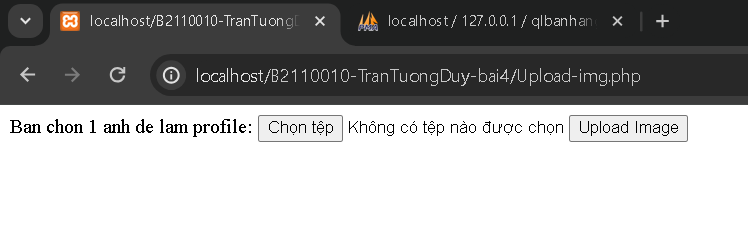




## Upload tập tin

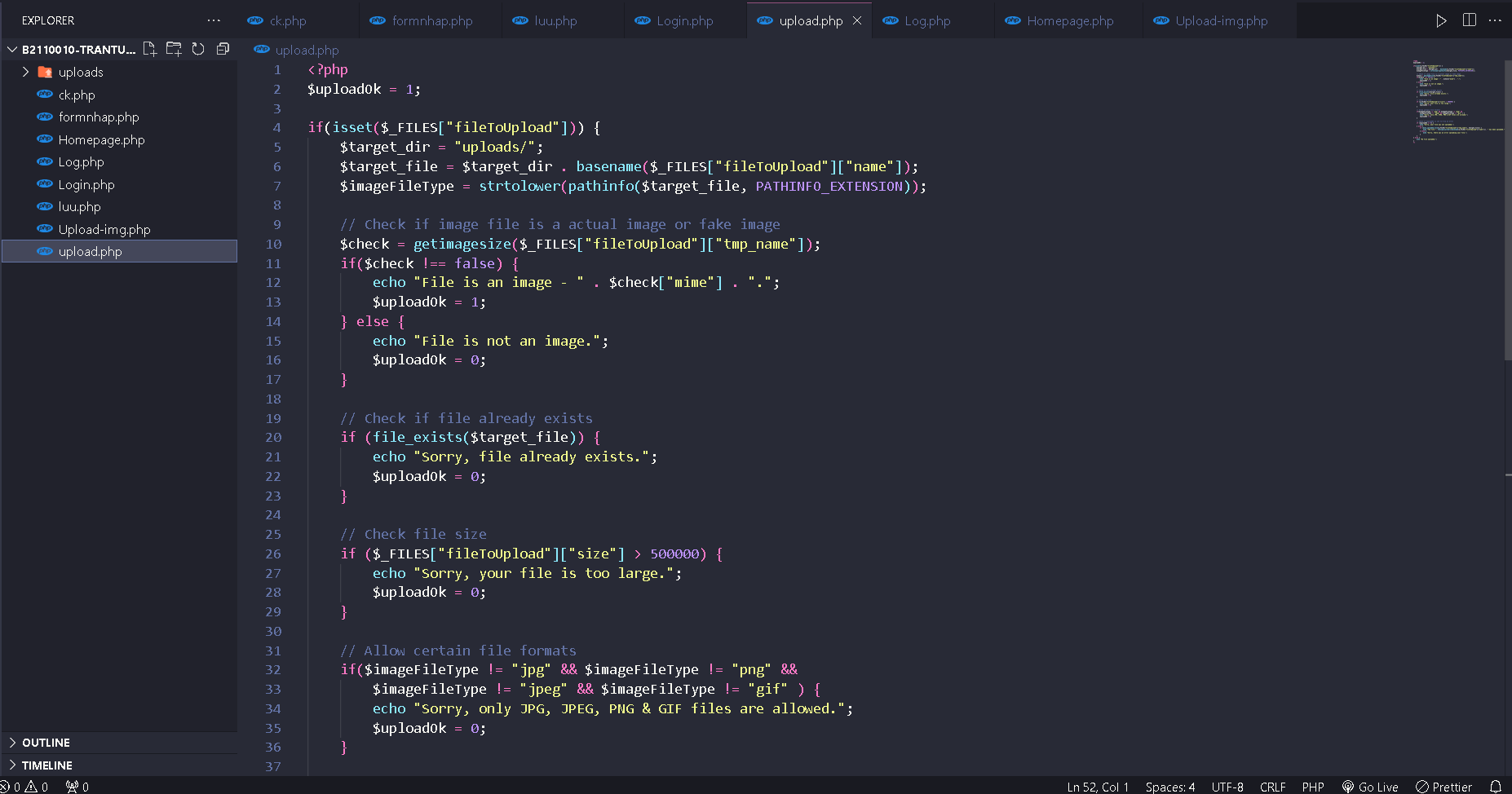
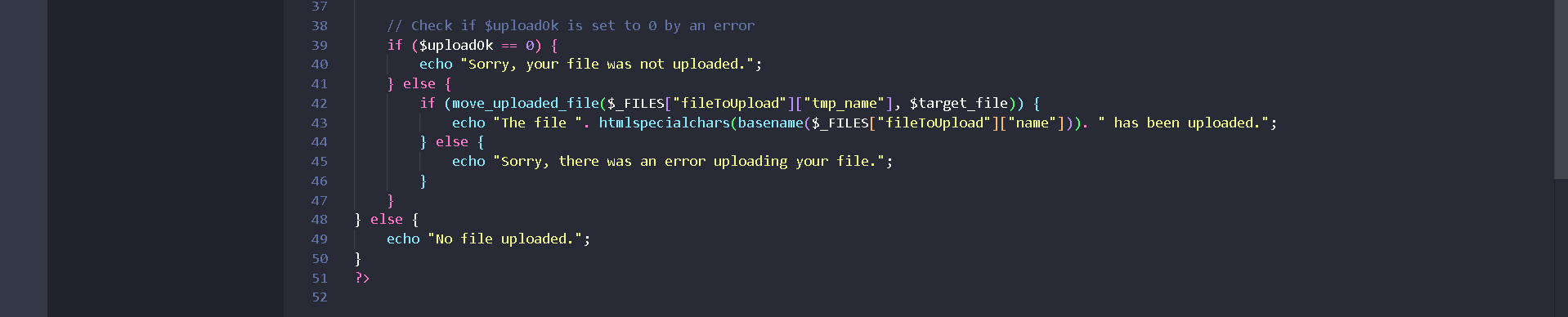
Upload file, chúng ta tạo form giao diện để upload file ảnh upload-img.php

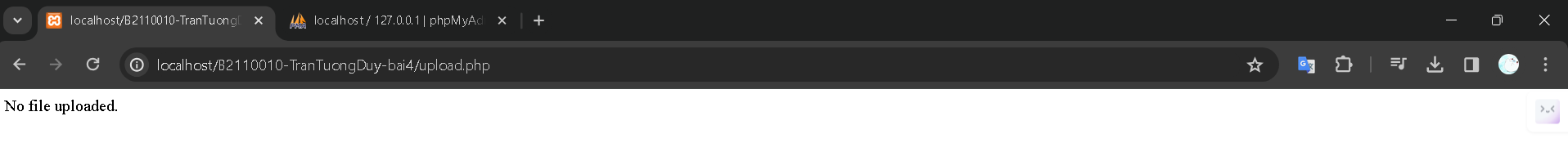
### Upload-img.php

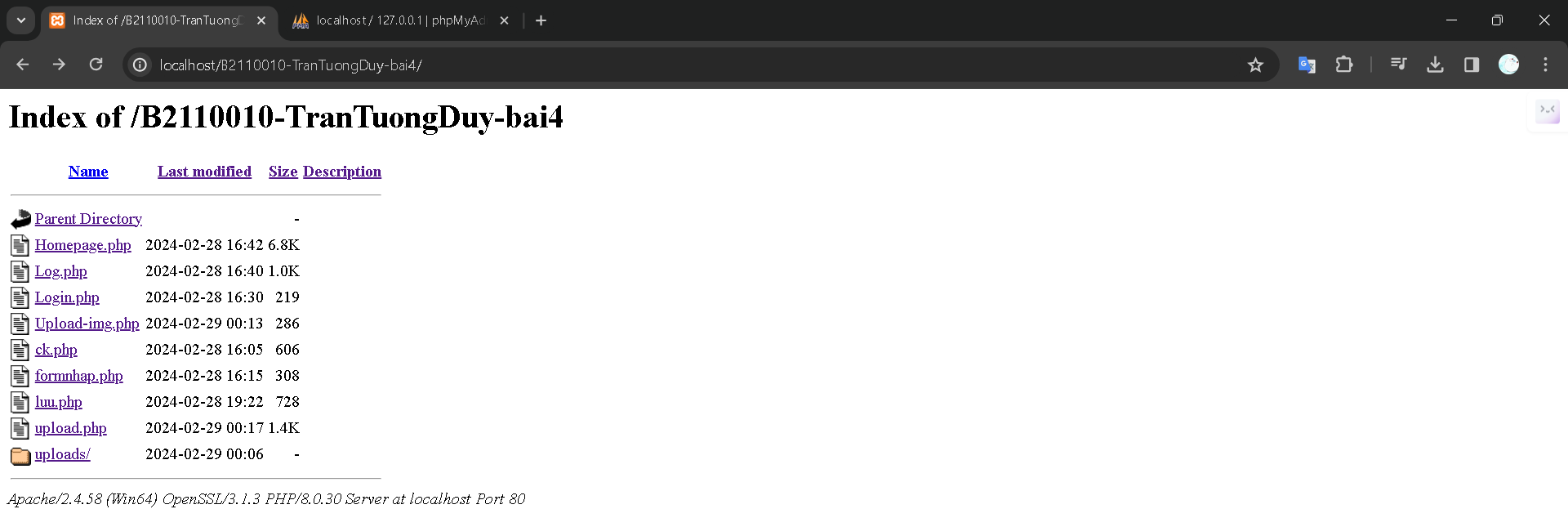
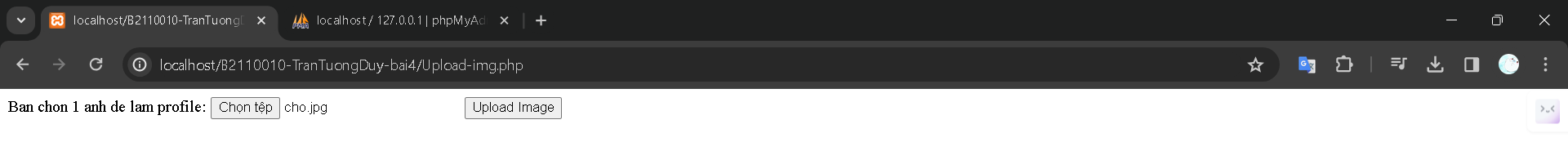
Tập tin upload.php xử lý hành động

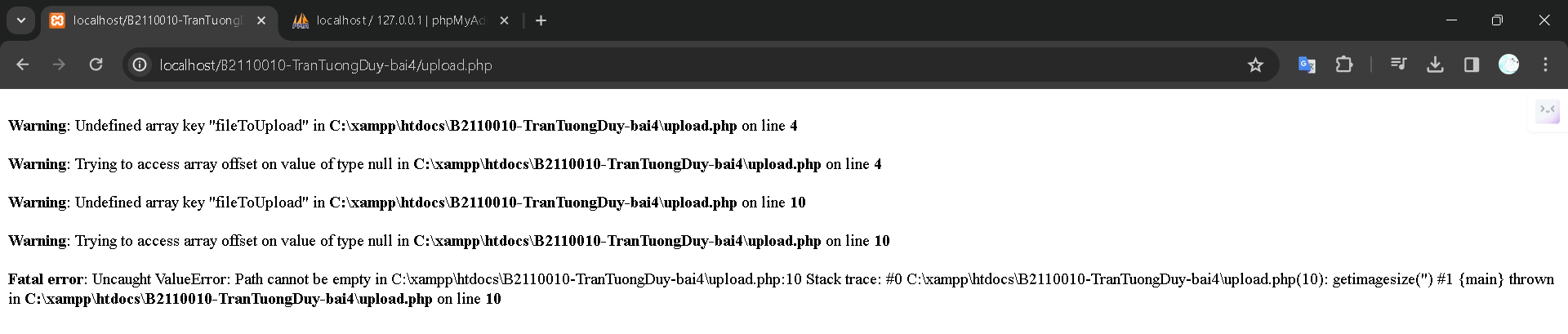
upload.php



>>Em sử dụng code của cô và đã gặp lỗi, khi bạn hỏi cô trong nhóm thì không thấy cô phản hồi nên em vẫn không sửa được và thay bằng code khác

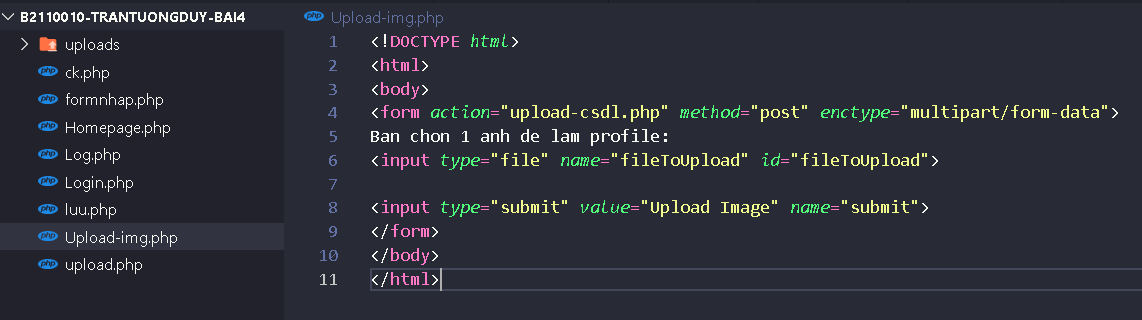


Upload ảnh và lưu dữ liệu về ảnh trong CSDL

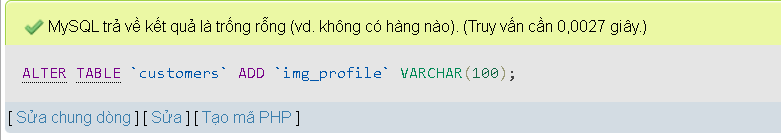
Trong phần này chúng ta minh họa tải tập tin ảnh làm ảnh profile

của người dùng

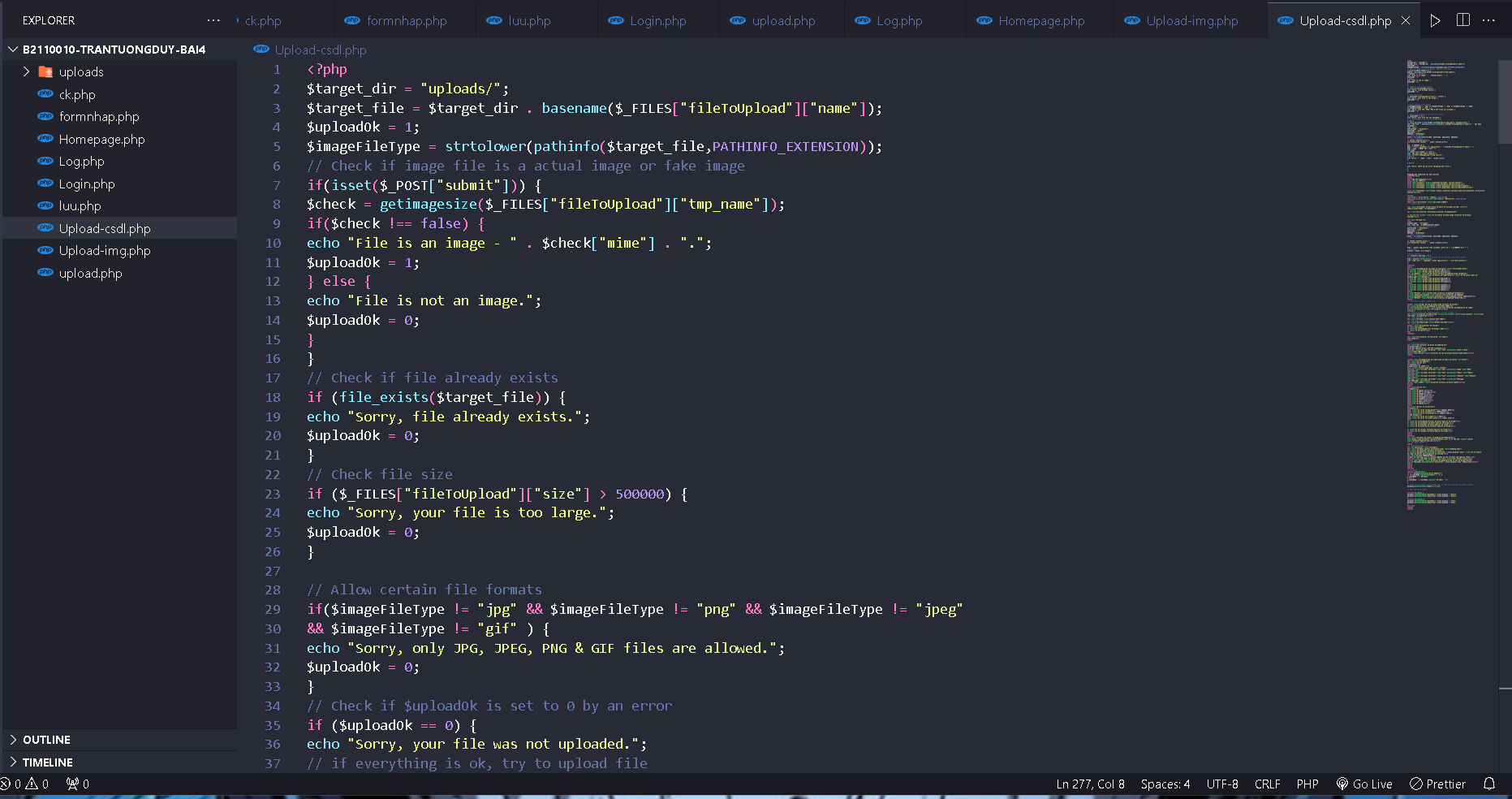
Upload-img.php (modified)



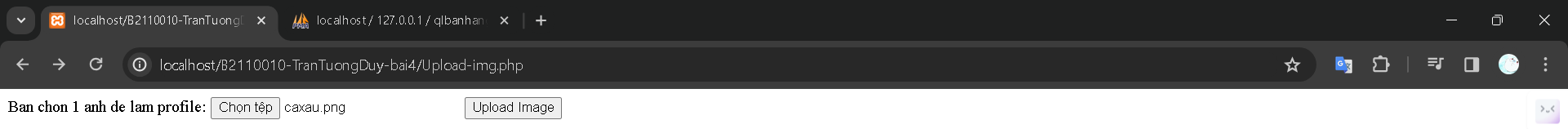
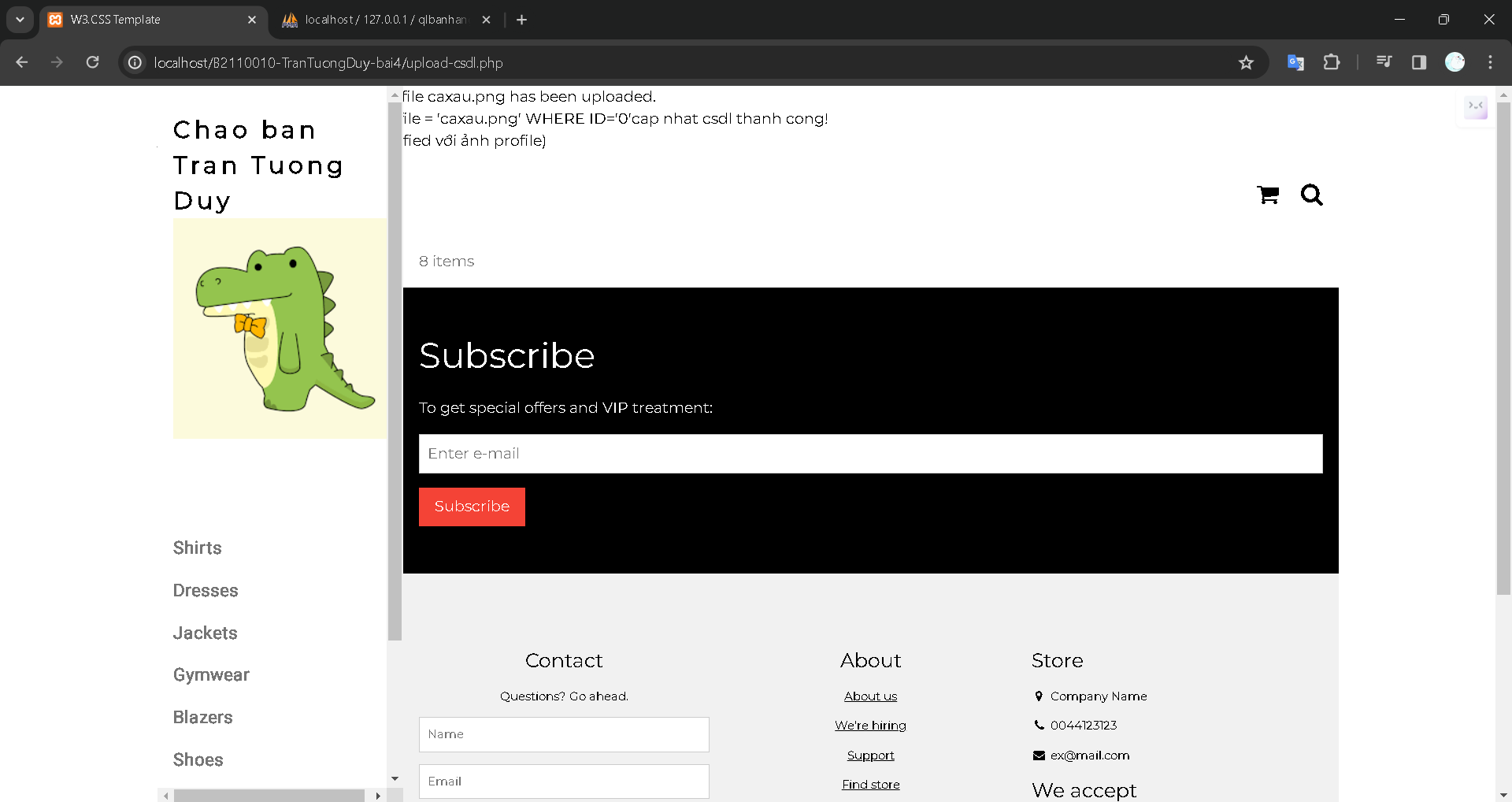
Thêm cột để lưu đường dẫn ảnh:



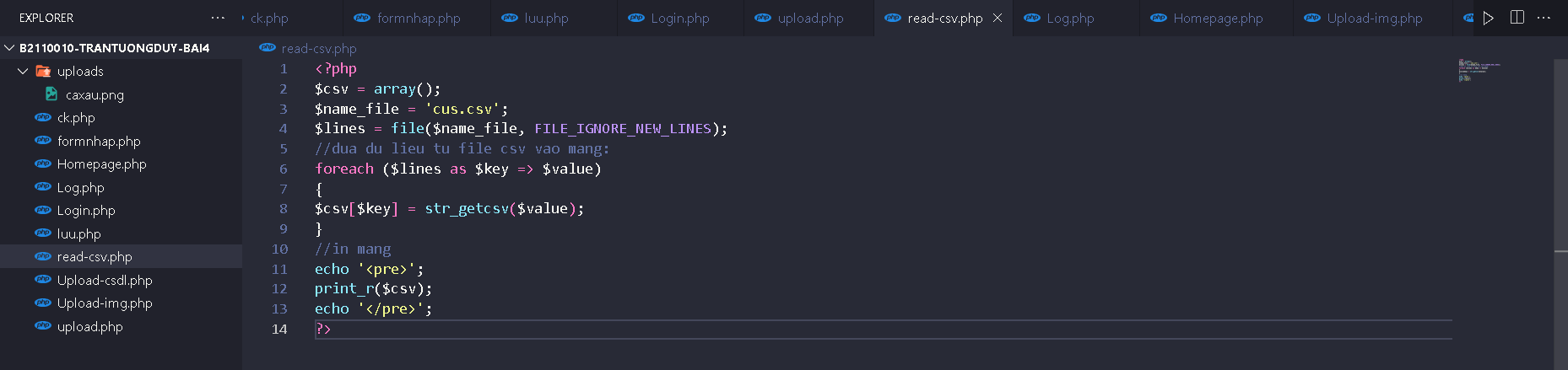
Upload-csdl.php



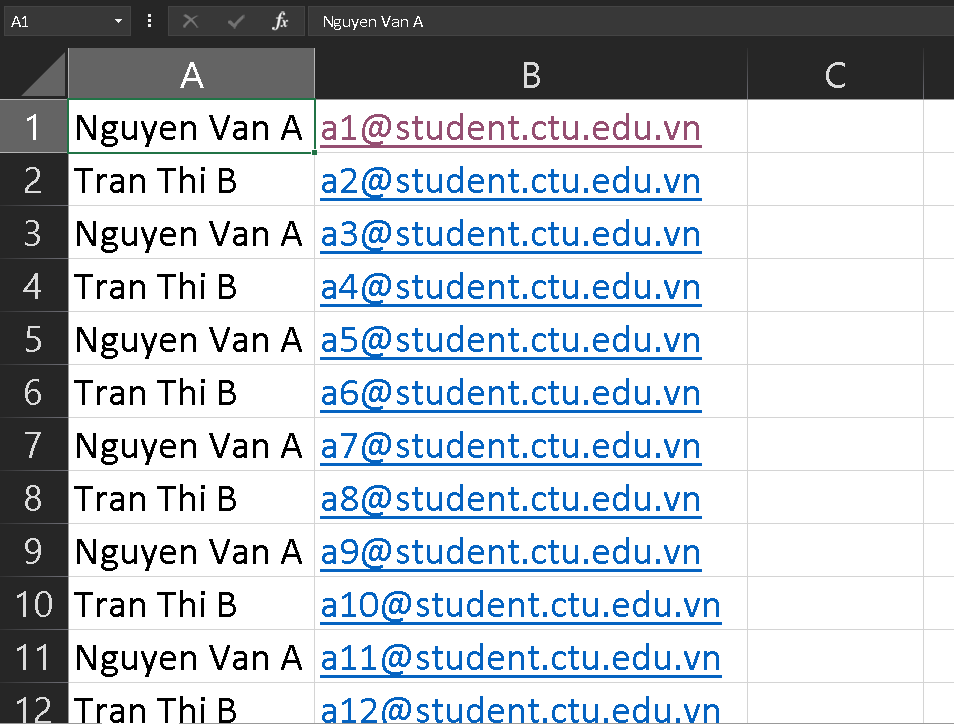
Kết quả:

  Upload file csv và đọc dữ liệu vào mảng

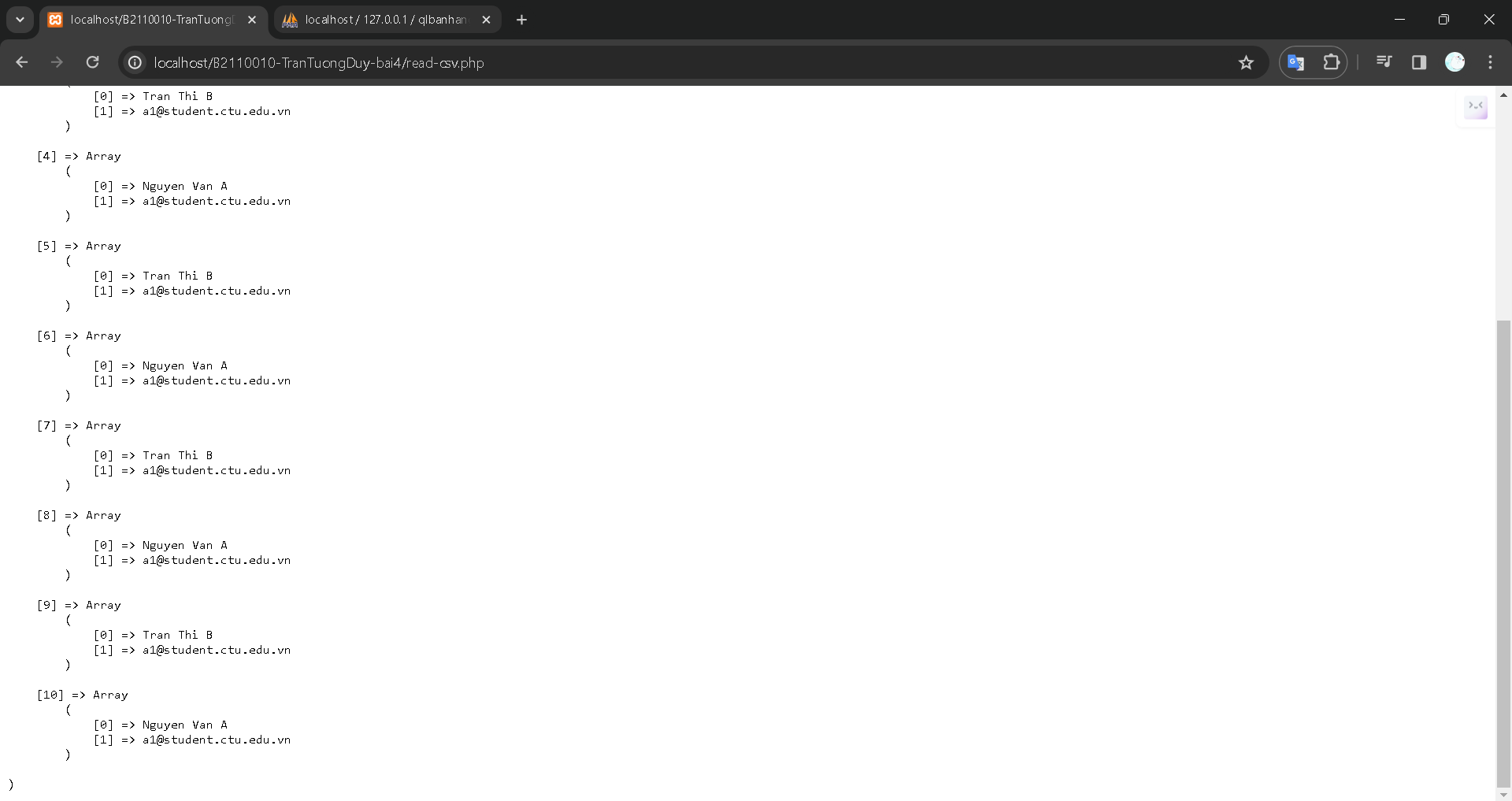
read-csv.php



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 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 tấn công phổ biến trong lập trình web, trong kẻ tấn công chèn các đoạn mã SQL bất hợp pháp vào các truy vấn SQL được tạo bởi ứng dụng web.

>> Khi ứng dụng không kiểm tra hoặc xác nhận đầu vào của người dùng đúng cách, hacker có thể chèn các câu lệnh SQL độc hại vào các trường dữ liệu của form và thực hiện các hành động không được phép như truy xuất, sửa đổi, xóa dữ liệu hoặc thậm chí là kiểm soát toàn bộ hệ thống.

- Các ví dụ:

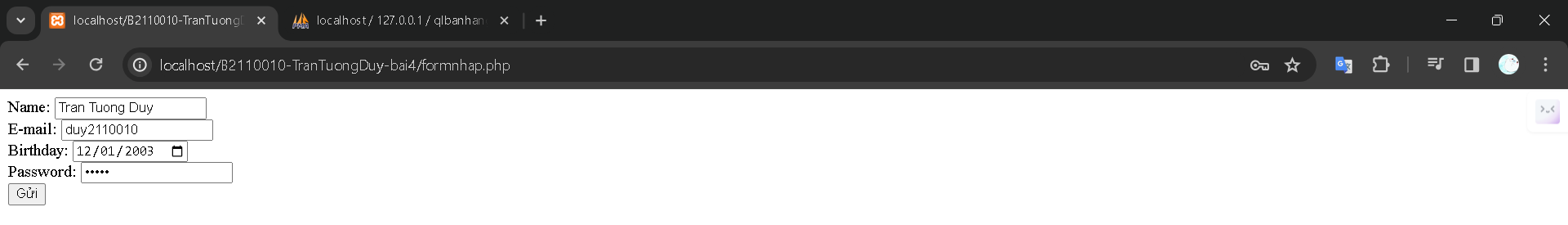
•Sử dụng câu lệnh SQL Injection để xóa bảng:

$sql = "SELECT id, firstname, lastname FROM MyGuests WHERE lastname = 'Doe'";

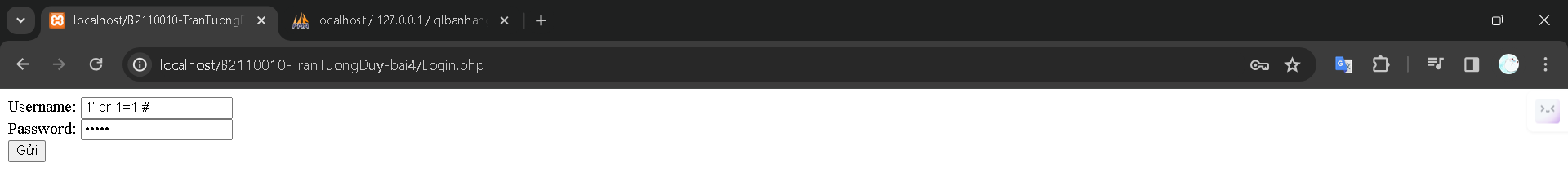
•Sử dụng câu lệnh SQL Injection để truy xuất dữ liệu bảng nhật ký đăng nhập:

$sql = "SELECT id, username FROM users WHERE username = '$username' AND password = '$password'";

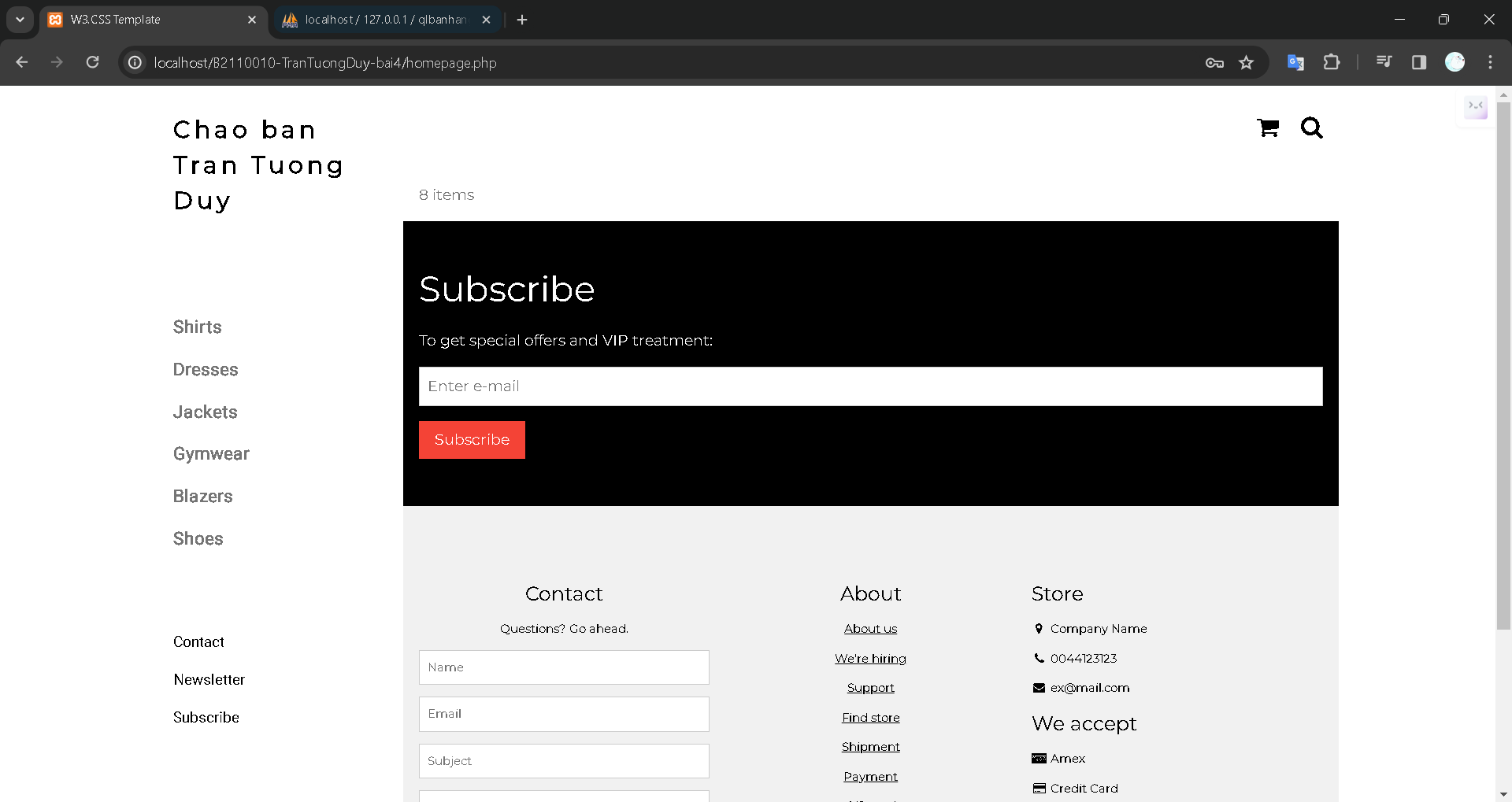
- Ứng dụng vào trang đăng nhập (với email = ‘duy2110010’ , pass =’11111’)



- Đăng nhập vào trang login với mật khẩu bất kỳ cùng key “1’ or 1=1 #”



Vào trang chủ với key “1’ or 1=1 #”



**3. 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 phần nhỏ chứa dữ liệu, được tự động lưu trữ trên máy tính của người dùng qua trình duyệt web. Thông tin này được sử dụng để nhận dạng và theo dõi người dùng trong các phiên truy cập sau của họ trên trang web.

>>Ý nghĩa của các tham số trong setcookie():

- Tên (name): Tên đặt cho cookie.

- Giá trị (value): Nội dung của cookie.

- Hết hạn (expire): Thời điểm khi cookie hết hiệu lực, được biểu diễn dưới dạng timestamp. Nếu không chỉ định, cookie sẽ tự động hết hiệu lực khi phiên làm việc kết thúc.

- Đường dẫn (path): Phạm vi của cookie, chỉ có các trang trong đường dẫn này mới có thể truy cập được cookie.

- Miền (domain): Phạm vi miền của cookie, chỉ có các trang thuộc miền này mới có thể truy cập được cookie.

- An toàn (secure): Xác định liệu cookie chỉ được gửi qua kết nối an toàn (HTTPS) hay không.

- Chế độ HTTP only (httponly): Đặt cookie như một cookie chỉ sử dụng trong HTTP, không thể truy cập được bằng JavaScript, nhằm mục đích ngăn chặn các cuộc tấn công Cross-Site Scripting (XSS).

- Cách thực hiện lưu, lấy và xóa giá trị trong cookie:

Lưu giá trị vào cookie: setcookie("user", "", time() + 3600, "/");

- Lấy giá trị từ cookie: $user = $\_COOKIE['user'];

echo "Chào mừng $user!";

- Xóa giá trị từ cookie: setcookie("user", "", time() - 3600, "/");

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

>>Sessions trong PHP là một cách để lưu trữ thông tin của người dùng qua nhiều trang web khác nhau. Mỗi phiên làm việc của người dùng sẽ được liên kết với một session duy nhất trên máy chủ. Để lưu trữ dữ liệu trong một session, bạn có thể sử dụng biến toàn cục $\_SESSION. Để bắt đầu một session, bạn cần sử dụng hàm session\_start() ở đầu của trang web hoặc tập tin PHP. Để truy cập giá trị trong session, bạn có thể sử dụng cú pháp $\_SESSION['tên\_biến'], trong đó 'tên\_biến' là tên của biến mà bạn muốn truy xuất giá trị. Để xóa toàn bộ session và kết thúc phiên làm việc hiện tại, bạn có thể sử dụng hàm session\_destroy().

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

- Bản chất:

• Cookie: là một dạng dữ liệu được lưu trữ trên máy tính của người dùng dưới dạng văn bản thông qua trình duyệt web. Cookie được gửi từ máy chủ đến trình duyệt và được lưu trữ tại đó.

•Session: là một cách để lưu trữ thông tin trên máy chủ. Mỗi phiên làm việc của người dùng được gắn với một ID phiên duy nhất, và thông tin trong session được lưu trữ trên máy chủ.

- Thời gian lưu trữ

• Cookie: có thể được lưu trữ trong một khoảng thời gian cụ thể, xác định bởi thời gian sống của cookie hoặc cho đến khi người dùng xóa cookie hoặc hết hạn.

• Session: thông tin trong session thường chỉ tồn tại trong suốt phiên làm việc của người dùng. Khi phiên làm việc kết thúc, thông tin trong session sẽ bị xóa.

- Bảo mật:

• Cookie: dữ liệu trong cookie được lưu trữ trên máy tính của người dùng và có thể bị tiết lộ nếu bị hack. Tuy nhiên, có thể sử dụng các tùy chọn bảo mật như mã hóa để bảo vệ dữ liệu.

• Session: thông tin session được lưu trữ trên máy chủ và không bị tiết lộ cho người dùng, điều này làm cho session an toàn hơn so với cookie trong một số trường hợp.

- Sử dụng:

• Cookie: thường được sử dụng để lưu trữ thông tin như thông tin đăng nhập, tùy chọn người dùng, và các dữ liệu như giỏ hàng trong các trang mua sắm trực tuyến.

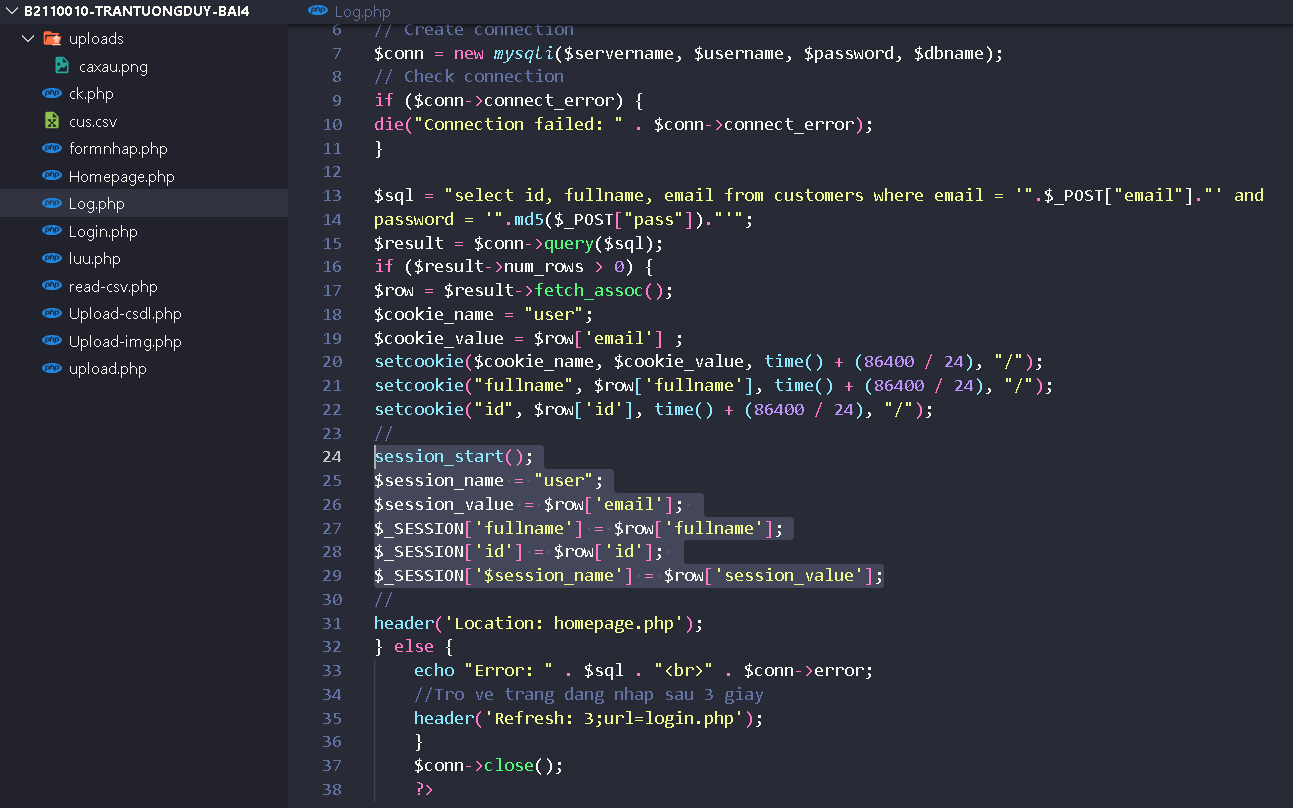
• Session: thường được sử dụng để lưu trữ thông tin phiên làm việc của người dùng, như thông tin đăng nhập, danh sách sản phẩm được chọn trong một phiên mua sắm, và các thông tin tạm thời khác.

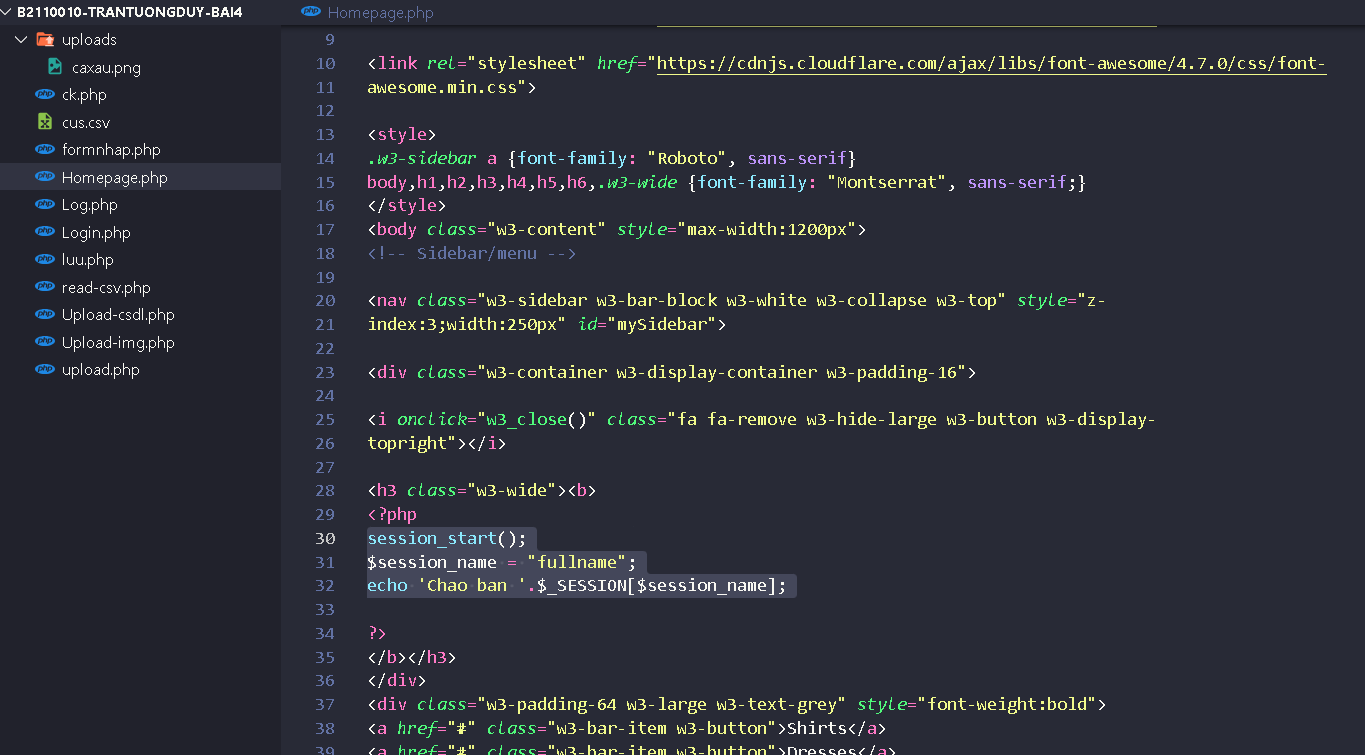
- Dung lượng lưu trữ:

• Cookie: có giới hạn về dung lượng ( thường khoảng vài KB dến vài MB).

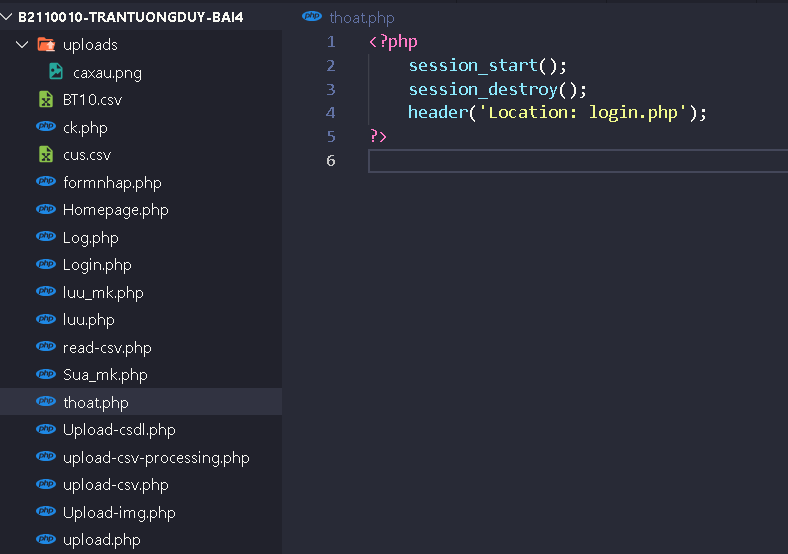
• Session: dung lượng lưu trữ phụ thuộc vào cấu hình máy chủ và không có giới hạn cụ thể, nhưng quản trị viên cần cân nhắc để tránh quá tải máy chủ với nhiều thông tin session.

**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.**



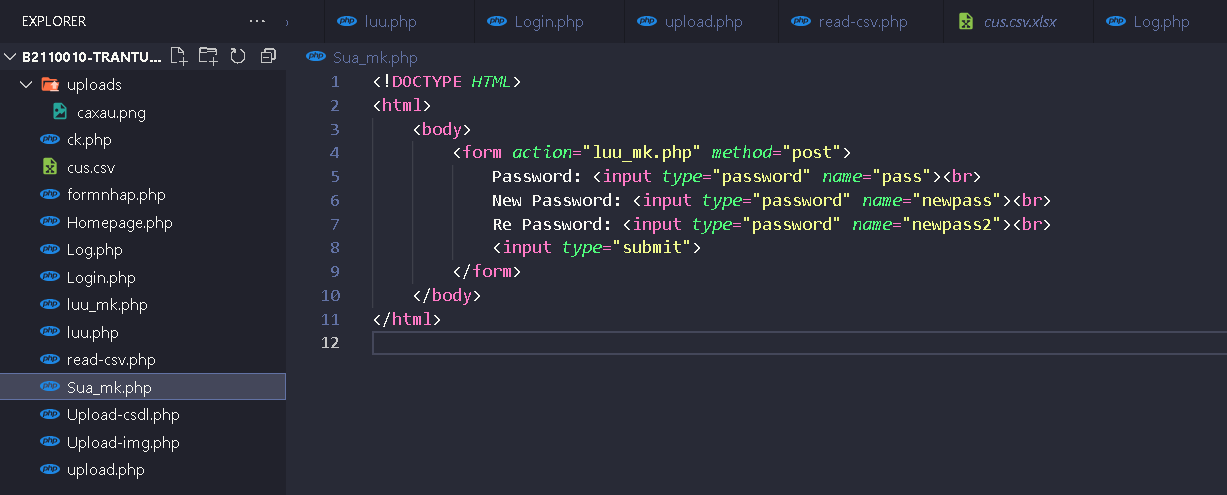


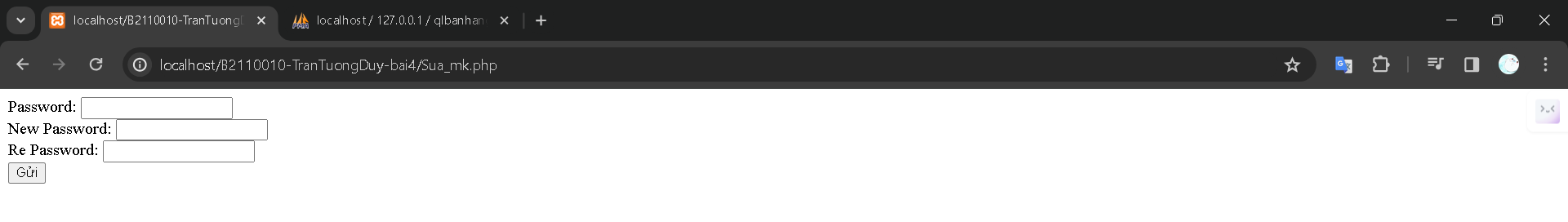
thoat.php



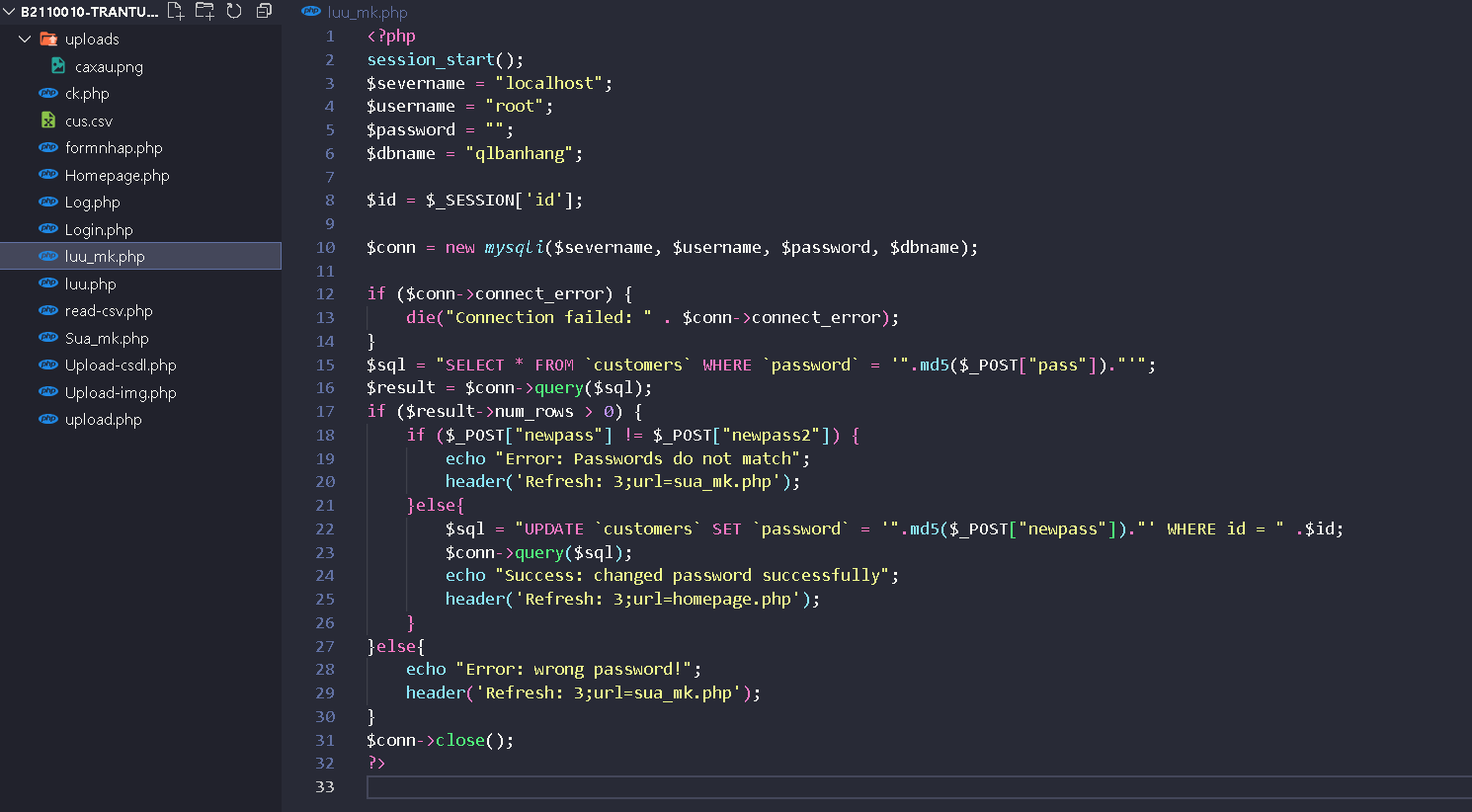
1. **Bạn hãy tạo form sua\_mk.php cho phép người dùng chỉnh sửa mật khẩu sau khi đăng nhập. Yêu cầu gồm 3 input: ô để nhập mật khẩu cũ, 1 ô để nhập mật khẩu mới, 1 ô cho phép nhập mật lại mật khẩu mới. Khi nhập xong cần đảm bảo: mật khẩu cũ là khớp với CSDL đang lưu, 2 ô nhập mật khẩu mới phải khớp với 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.**

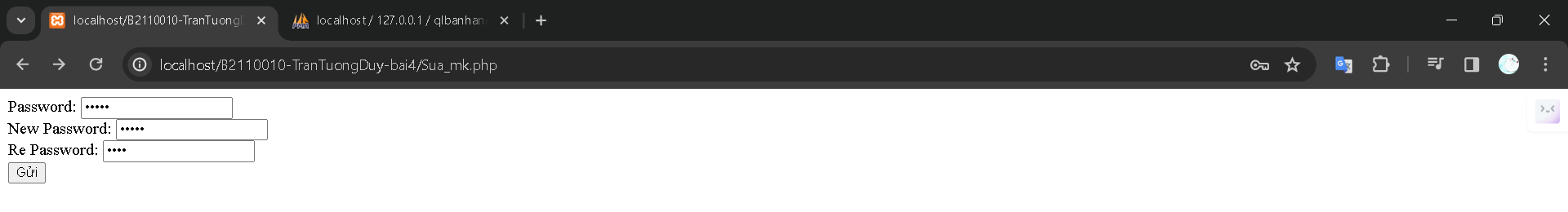
Sua\_mk.php

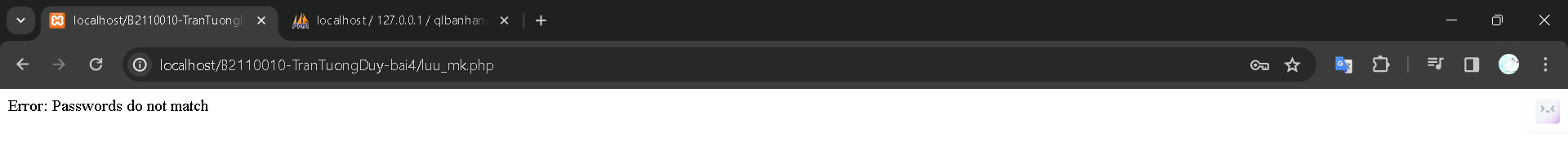




Luu\_mk.php





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

>>Trong HTML, để cho phép người dùng tải tệp lên, cần sử dụng một mẫu HTML <form> với thuộc tính enctype="multipart/form-data" để cho phép tải lên tệp. Ví dụ:

<form action="upload.php" method="post" enctype="multipart/form-data">

Chọn tệp để tải lên:

<input type="file" name="fileToUpload" id="fileToUpload">

<input type="submit" value="Tải lên tệp" name="submit">

</form>

>>PHP Upload File: trang web cung cấp một ví dụ về cách xử lý tệp được tải lên trong PHP. Đoạn mã này bao gồm các bước sau:

- Kiểm tra xem tệp đã được chọn và tải lên thành công chưa.

- Kiểm tra kích thước của tệp để đảm bảo rằng nó không quá lớn.

- Kiểm tra loại tệp để đảm bảo rằng nó là loại tệp hợp lệ.

- Di chuyển tệp từ thư mục tạm thời đến thư mục mục tiêu.

>>Hàm PHP cụ thể:

- move\_uploaded\_file(): được sử dụng để di chuyển tệp tải lên từ thư mục tạm thời sang một vị trí mới. Ví dụ:

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

>>Biến siêu toàn cục $\_FILES: trong PHP, các tệp được tải lên thông qua một mảng gọi là $\_FILES. Trong ví dụ, $\_FILES["fileToUpload"]["tmp\_name"] được sử dụng để tham chiếu đến tệp tạm thời đã được tải lên.

>>Xử lý lỗi: trang web cũng cung cấp một số mã để xử lý các lỗi có thể xảy ra trong quá trình tải tệp lên, chẳng hạn như kiểm tra kích thước tệp quá lớn hoặc loại tệp không được phép.

>>Bảo mật: ngoài các hướng dẫn cơ bản, trang web cũng cung cấp một số gợi ý về bảo mật, chẳng hạn như kiểm tra loại tệp và mô phỏng việc tải lên tệp trong một thư mục không công khai.

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

- Hàm file() đọc một tập tin vào một mảng. Mỗi phần tử mảng chứa một dòng từ tệp, với ký tự dòng mới vẫn được đính kèm.

- Cú pháp của hàm file() như sau: file(filename, flag, context)

- Các tham số trong hàm file():

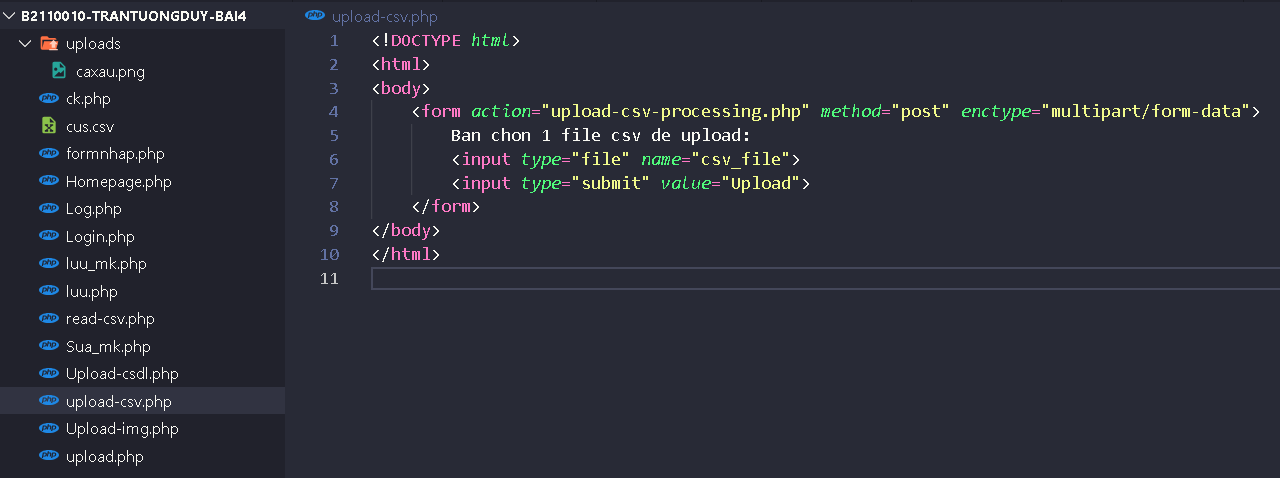
+ $filename (bắt buộc): Đường dẫn của tệp tin cần đọc. Đây có thể là một đường dẫn tệp tin tương đối hoặc tuyệt đối trên hệ thống tệp tin.

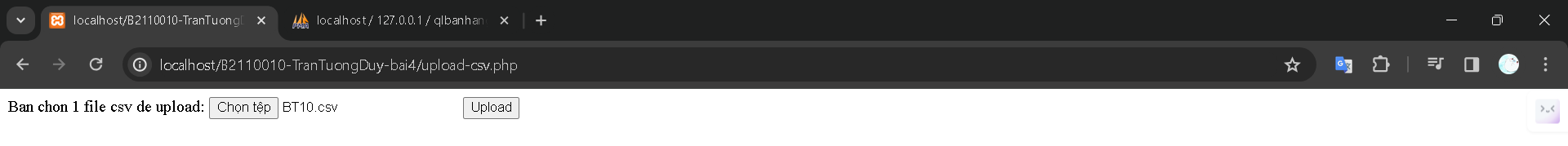
+ $flags (tùy chọn): Các cờ điều khiển đọc tệp tin. Các giá trị phổ biến cho tham số này bao gồm: FILE\_USE\_INCLUDE\_PATH: Tìm kiếm tệp tin trong danh sách đường dẫn include\_path. FILE\_IGNORE\_NEW\_LINES: Loại bỏ ký tự xuống dòng (\n) khỏi các phần tử trong mảng kết quả. FILE\_SKIP\_EMPTY\_LINES: Bỏ qua các dòng trống trong tệp tin.

**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 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.**

Tạo file csv10 dòng dữ liệu

upload-csv.php





Tập tin upload-csv-processing.php

