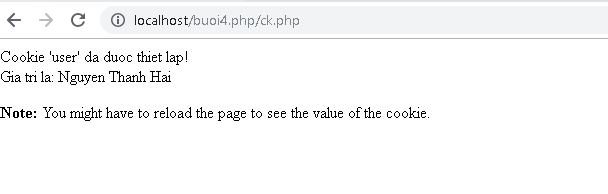
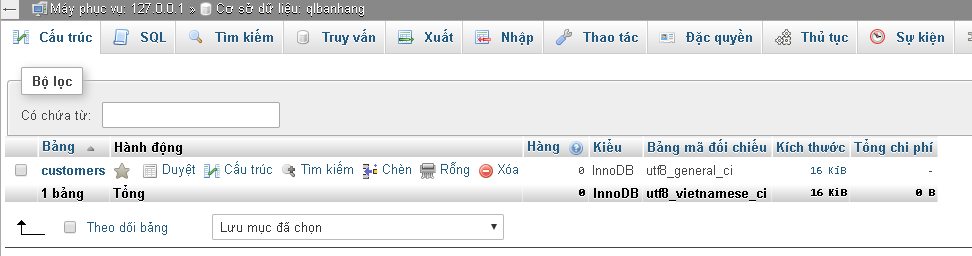
1. **Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.**

ck.php



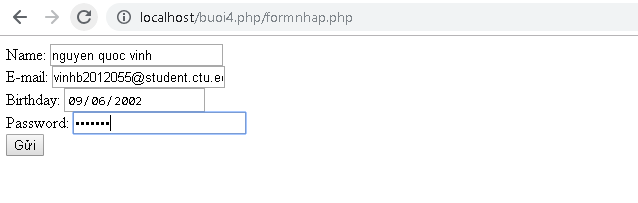
csdl qlbanhang:



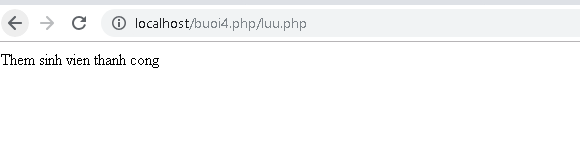
Bang customers:



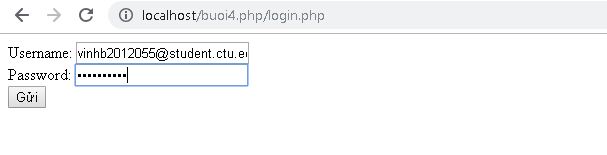
formnhap.php:



luu.php:

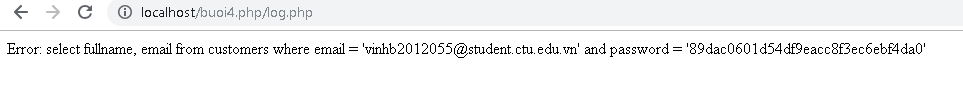


login.php:

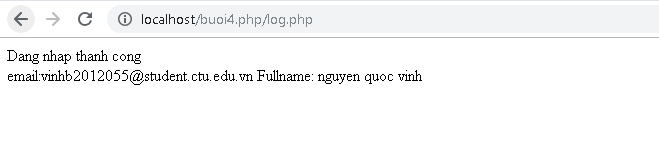


log.php:

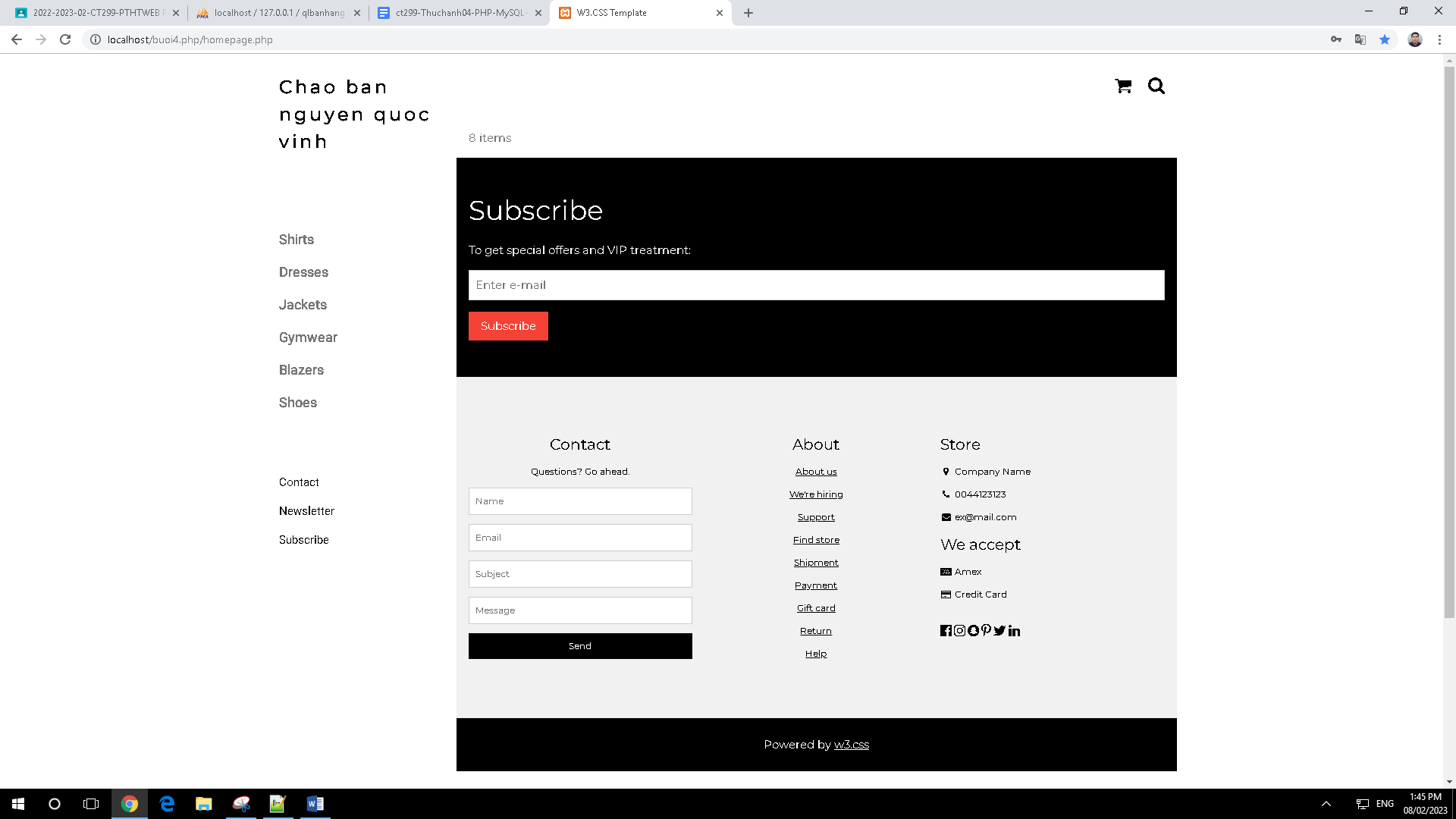
nếu nhập login.php sai:



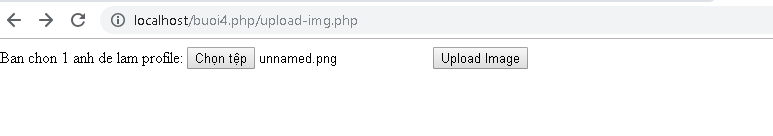
Nếu nhập login.php đúng:



homepage.php:

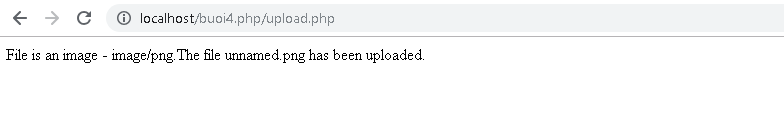


upload-img.php:

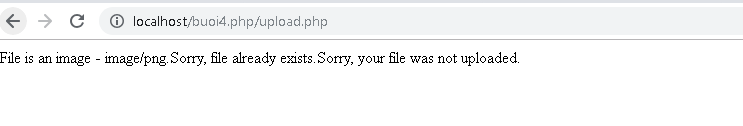


upload.php:

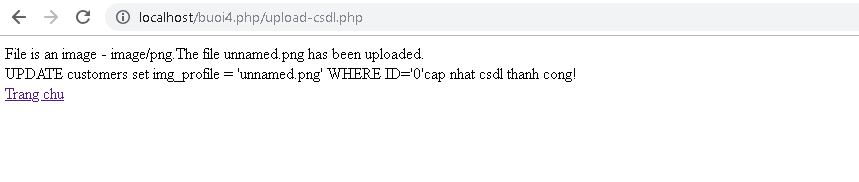
nếu chưa có img:



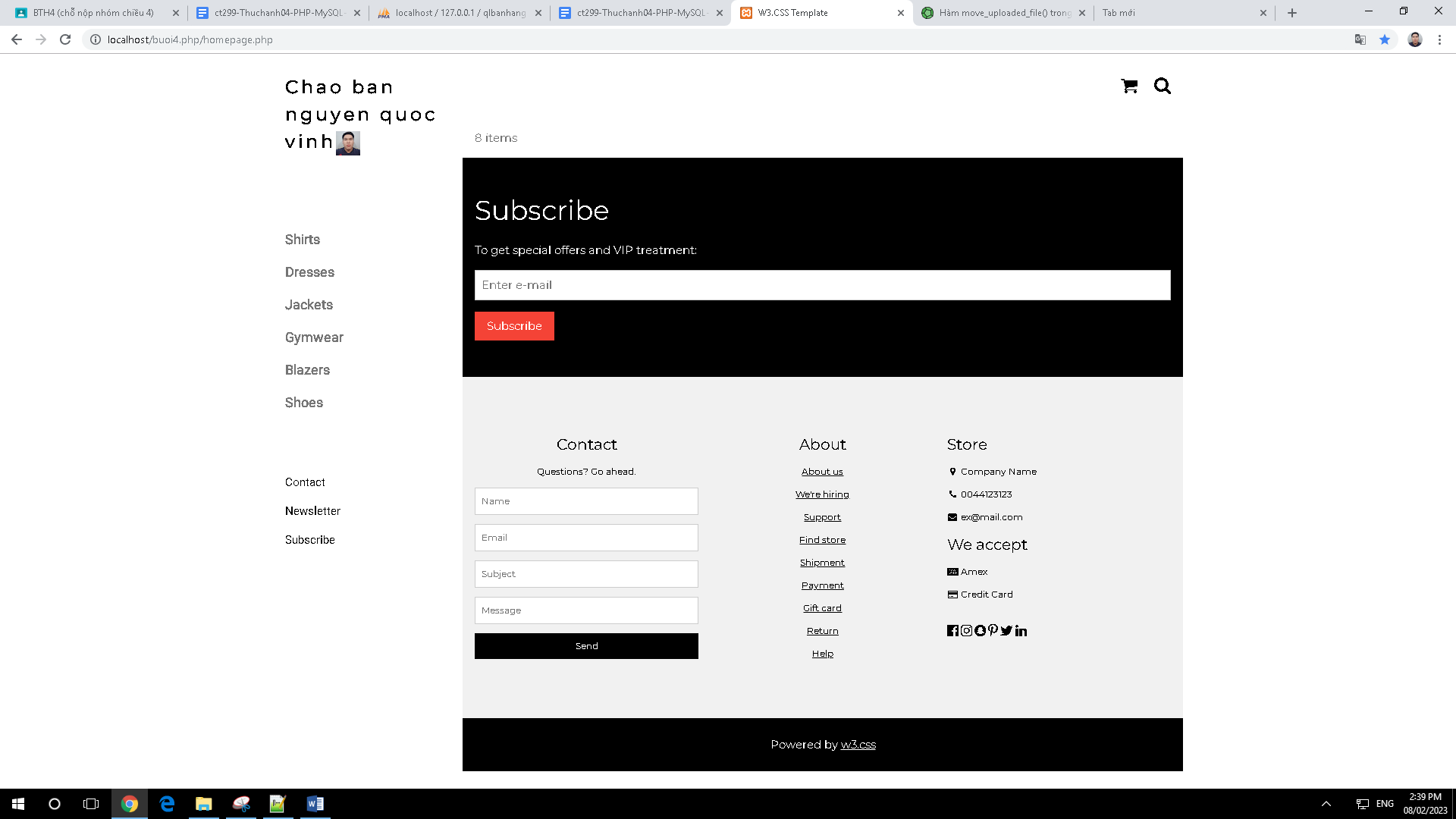
Nếu đã có img rồi:



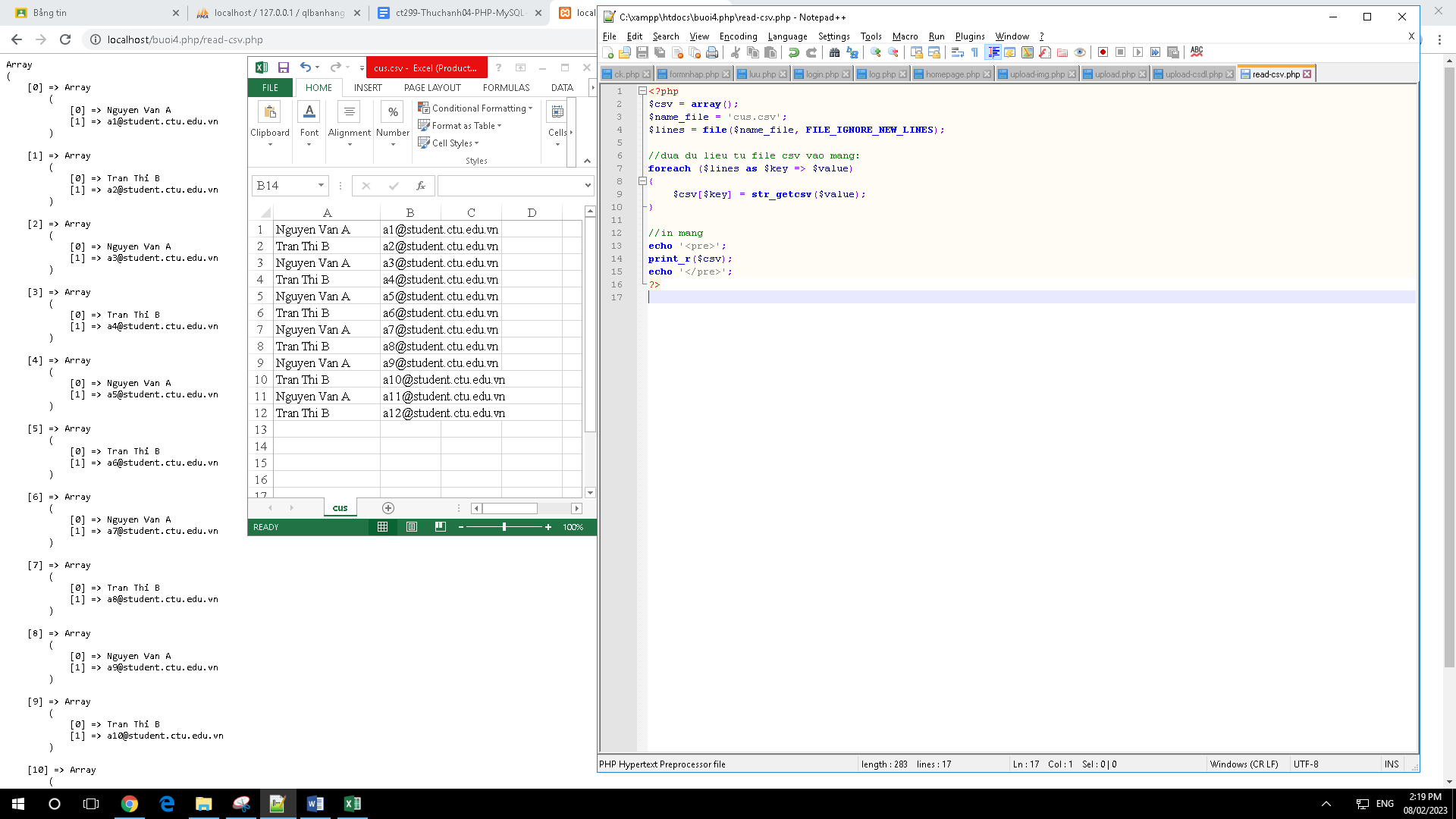
upload-csdl.php:



Homepage.php sau khi thêm ảnh:

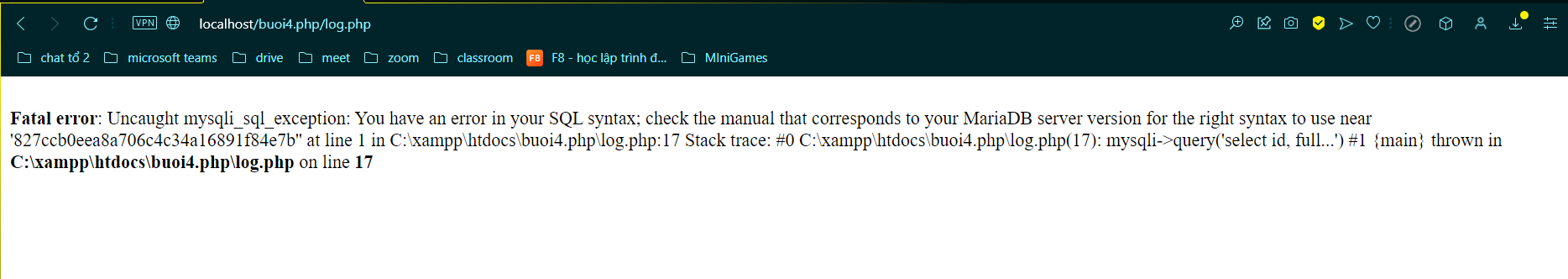


read-csv.php:



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 . Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.**

sql\_injection là một lỹ thuật lợi dụng những lỗ hổng về câu truy vấn của các ứng dụng bầng cách chèn thêm các câu lệnh sql để làm sai lệch đi câu truy vấn gốc, từ đó có thể khai thác dữ liệu từ database. Sql\_injection biến những người sữ dụng có quyền thành thao tác như một quản trị web trên csdl của họ.

vd: trang login.php khi ta nhập tên tk ' or "1"="1" và mật khẩu là ' or "1"="1" thì trang sẽ hiện thông báo là

điều này chứng tỏ hacker có thể dựa vào dữ liệu trên đưa ra code là $sql = "select id, fullname, email from customers where email = '".$\_POST["email"]."' and password = '".md5($\_POST["pass"])."'"; tuy nhiên do mật khẩu đã được mã hóa nên hacker sẽ rất khó khăn để bẻ khóa lớp bảo mật và đăng nhập vào tài khoản người dùng.

1. **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 là một tệp nhỏ mà máy chủ nhúng vào máy tính người dùng , thường được dùng để lưu trử tùy chọn riêng cho từng user, được luu trử trong client khi đóng trình duyệt cookie vẫn tồn tại cho đến khi hết thời gian sống.

Diễn giải cú pháp :setcookie(name, value, expire, path, domain, secure, httponly)

Trong đó:

* $name là tên của cookie các bạn muốn tạo.
* $value là giá trị của cookie đó.
* $expire là thời gian sống(hạn sữ dụng) của cookie.
* $path là đường dẫn lưu trữ cookie.
* $domain là tên của domain.

\*Lưu ý: hàm setcookie() được đạt trước tag <html></html> và trong hàm setcookie() $name là dữ ;iệu bắt buộc các dữ liệu còn lại là tùy chọn.

* Lưu cookie:

Để lưu cookie ta dùng hàm setcookie()

Vd: setcookie(‘user’, ‘Nguyen Quoc Vinh’, time() +(86400 \*7) ); \*lệnh này sẽ tạo một cookie tên ‘user’ giá trị là ‘Nguyen Quoc Vinh’ ,tồn tại trong 1 tuần dữ liệu sau khi nnhập sẽ được lưu trong $\_COOKIE.

* Lấy cookie:

Để lấy cookie ta dùng lệnh $\_COOKIE()

Vd: lấy $name ta gõ $\_COOKIE(‘user’); vì $name mà ta vừa viết là ‘user’.

* Xóa cookie:

Để xóa cookie ta chỉ việc giảm thời gian sống của cookie đó.

Vd: setcookie(‘user’, ‘Nguyen Quoc Vinh’, time() -(86400 \*7) ); \*cookie sẽ hết hạn và tự động bị xóa.

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

Session dùng để lưu trử thống tin người dùng thông tin cấu hình... Sesssion tồn tại như một phiên nên khi đóng trình diệt thì sesstion sẽ kết thúc.mỗi client có một idsession khac nhau.Session thường dùng để đăng nhập, tạo giỏ hàng…

* Lưu giá trị:

sau khi thêm,sửa session sẽ được lưu trong $\_SESSION.

Vd: $\_SESSION[‘user’]=’admin’; session sẽ được lưu $name là ‘user’ và $value là ‘admin’.

* Lấy giá trị:

Ta dùng $\_SESSION[] dể lấy giá trị đẫ lưu

Vd: $\_SESSION[‘user’] sẽ lấy giá trị ‘user’ là ‘admin’.

* Xóa giâ trị:

Để xóa giá trị ta dùng:lệnh unset();

Vd unset($\_SESSION[‘user’]);

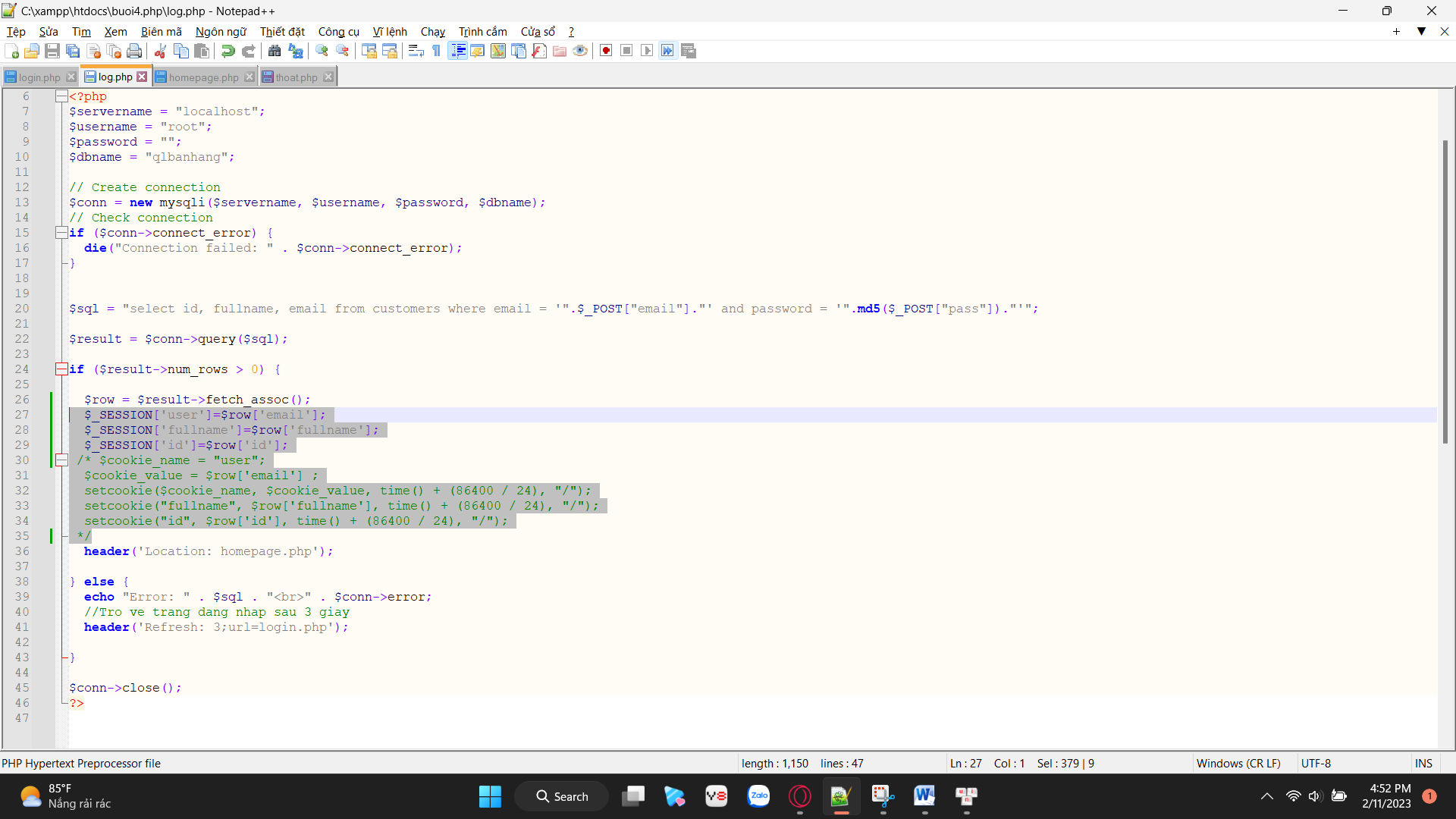
1. **Bạn hãy so sánh Cookie và Session.**

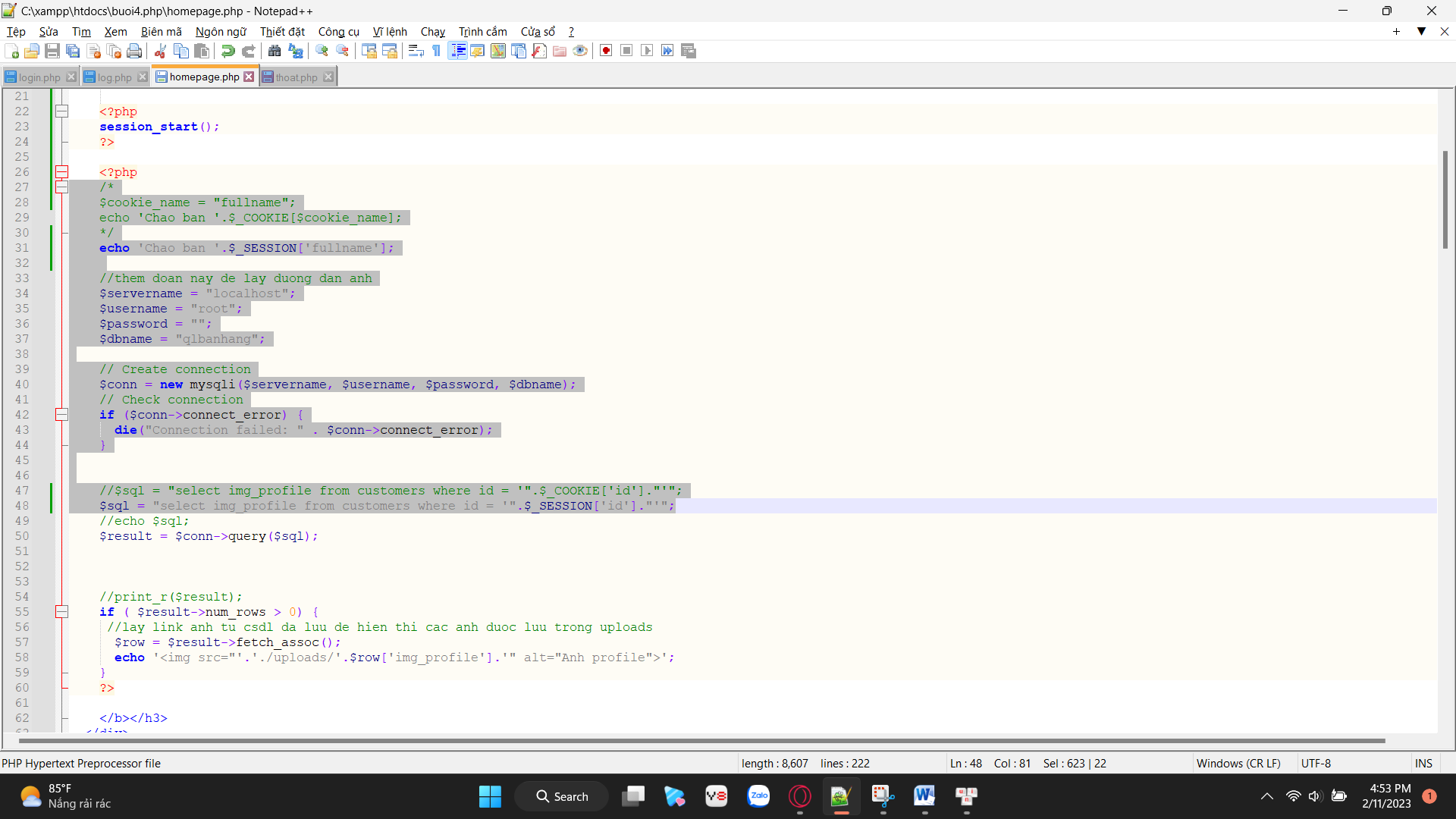
Cả cookie và session đều dùng để lưu dữ liệu của người dùng tuy nhiên điểm khác biệt giữa cookie và session chính là:

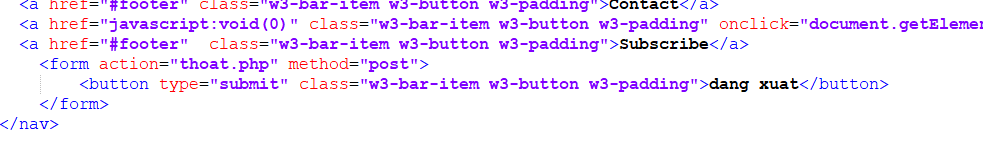
* Cookie sẽ lưu trử trên trình duyệt của người dùng còn session thì không
* Dữ liệu lưu trử của cookie thì nằm trên máy khách hàng còn session là nằm trong máy chủ
* Dữ liệu cookie có thể dể dàng sữa đổi vì chúng nằm trong máy khách còn session nằm trong máy chủ nên việc sữa đổi là rất khó khăn
* Dữ liệu cookie có thời hạn nhất định và ko bị mất nếu chưa hết thời hạn đó còn sesion thì sẽ mất nếu ta đóng trình duyệt.

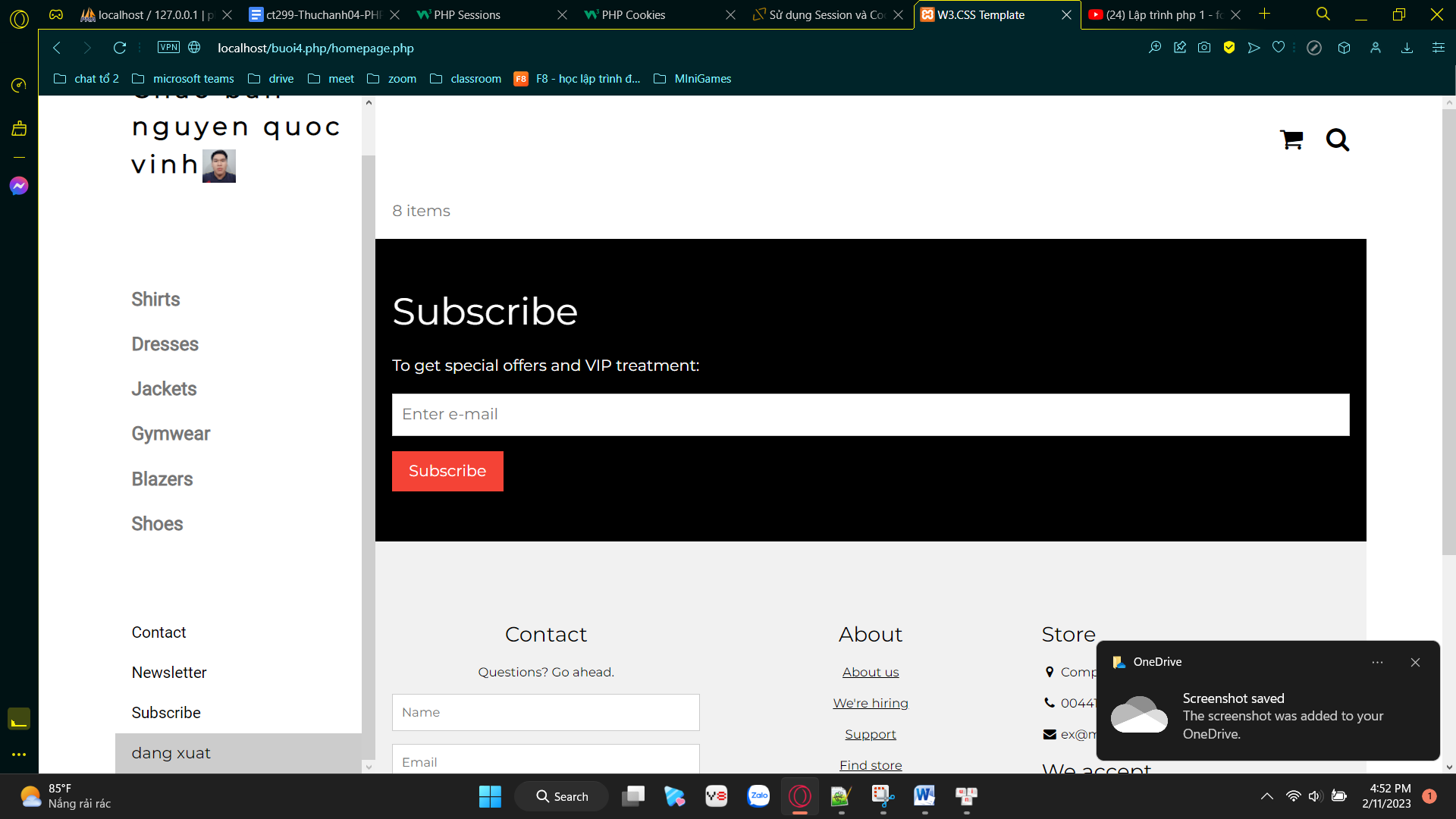
1. **Chỉnh sửa các đoạn gán, khởi tạo, lấy giá trị cookie trong log.php và homepage.php thay bằng cách dùng Session. Tạo tập tin thoat.php để xóa các giá trị trong session, cookie thực hiện chức năng log out khỏi hệ thống.**

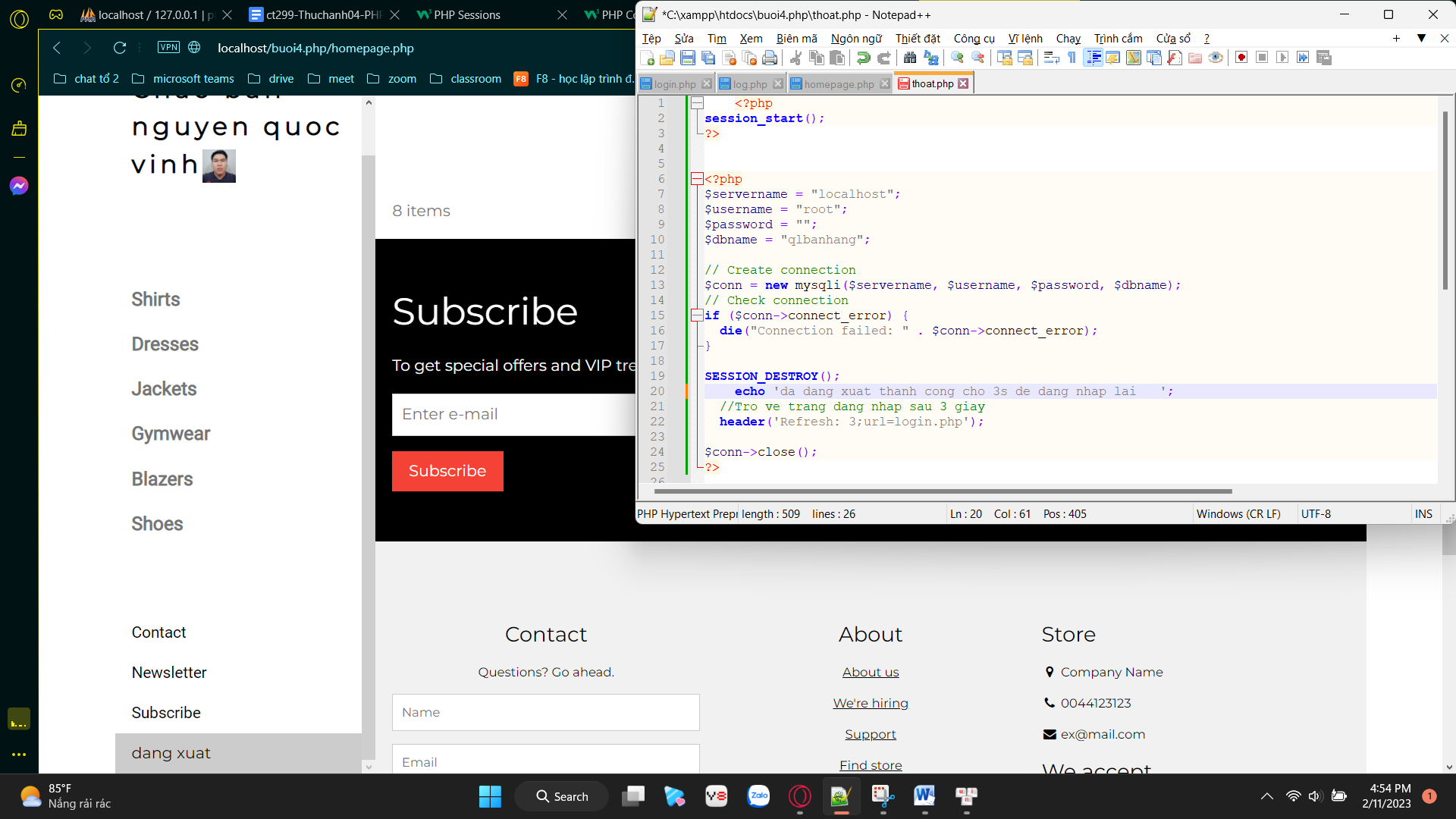
Chuyển cookie sang session:

Log.php: 

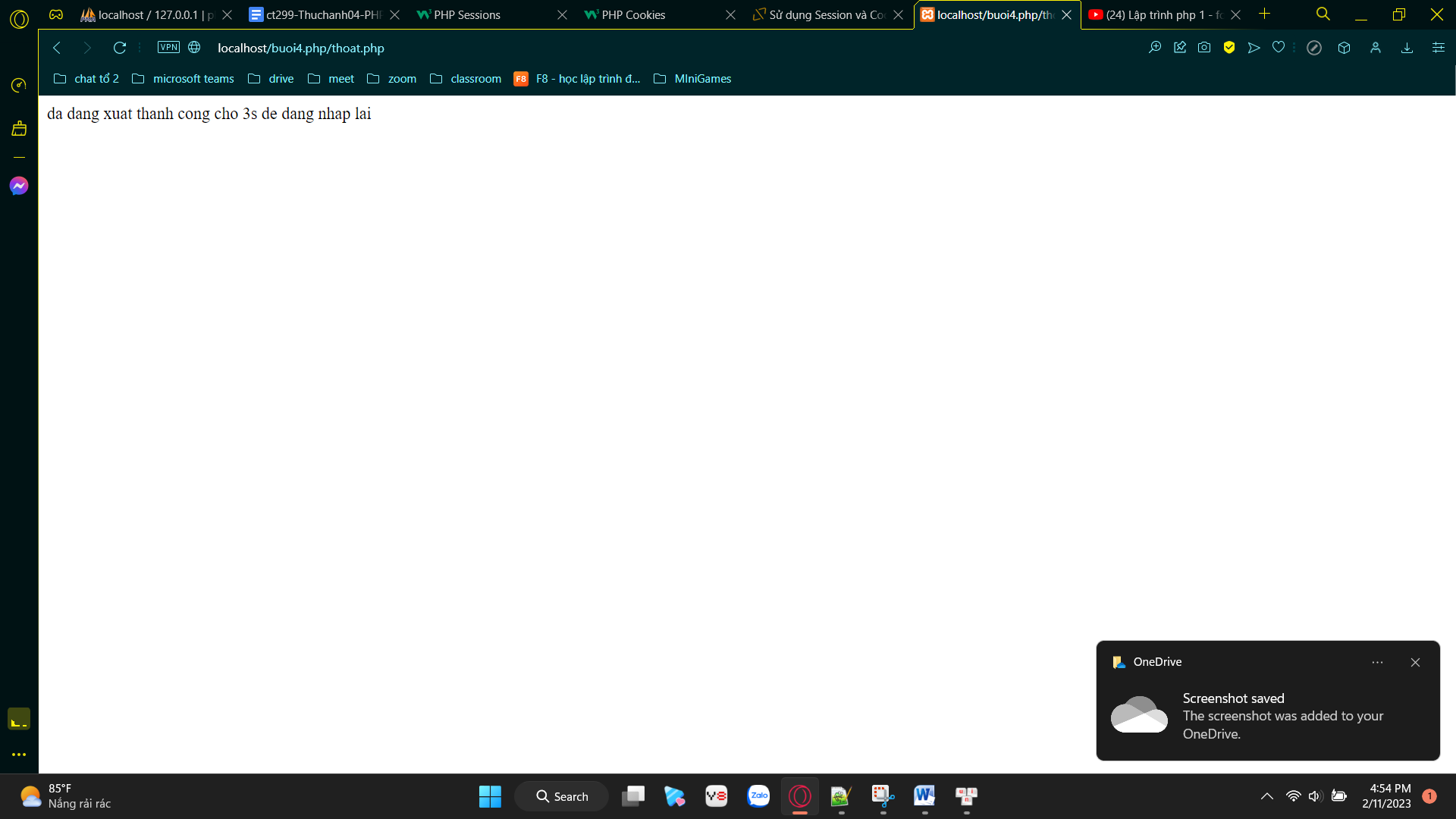
Homepage.php: 

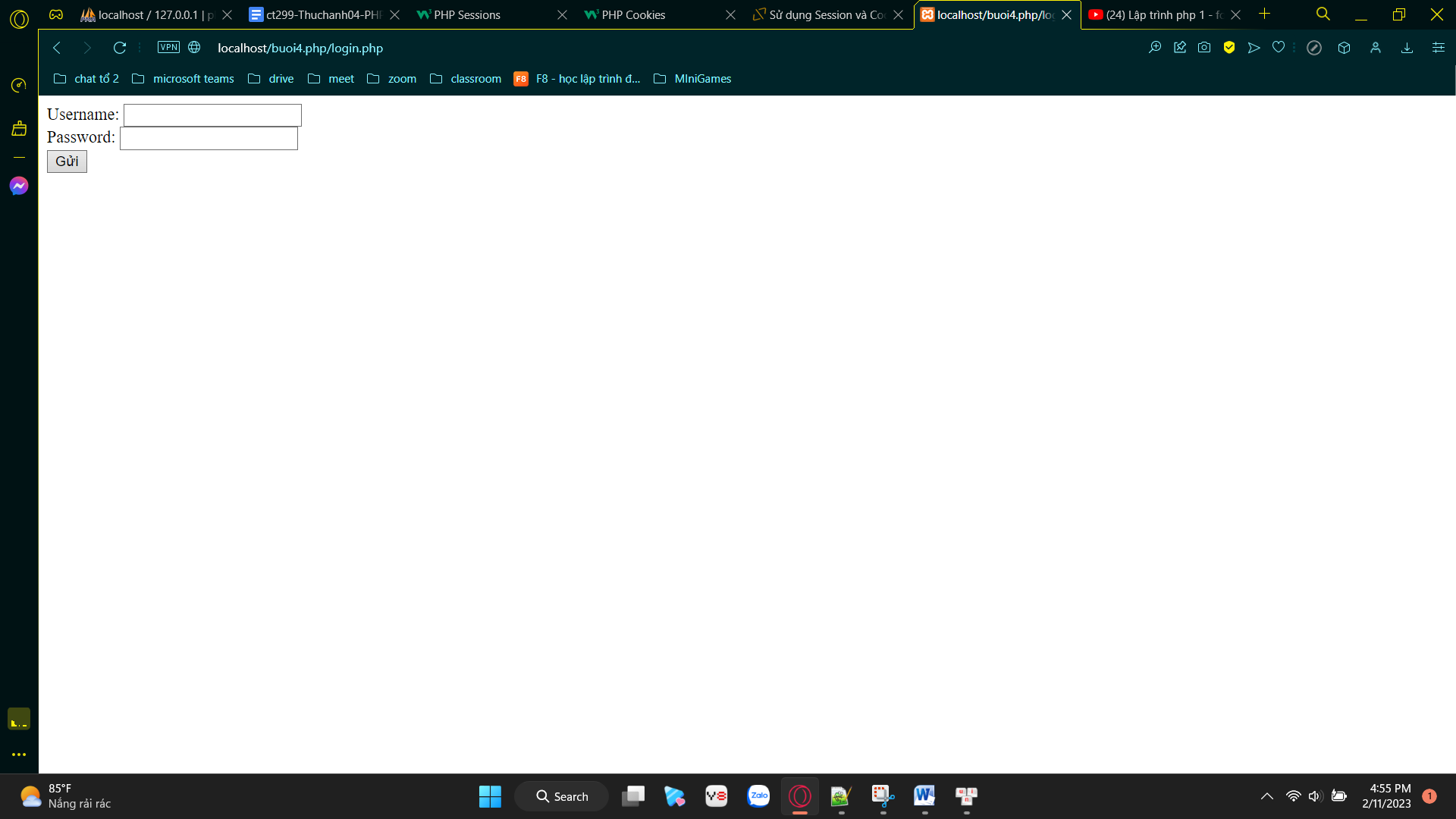
Thêm nút đăng xuất vào trang homepage.php 



Thoat.php: 

Bấm vào sẽ xóa hết dữ liệu session và trở về trang login.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.**
2. **Bạn đọc trong để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.**

Đầu tiên tạo một html để người dùng có thể chọn ảnh và upload:



Sau khi tạo html để lấy file ta tiến hành xác định đường dẫn cho file kiểm tra file có thật hay không:



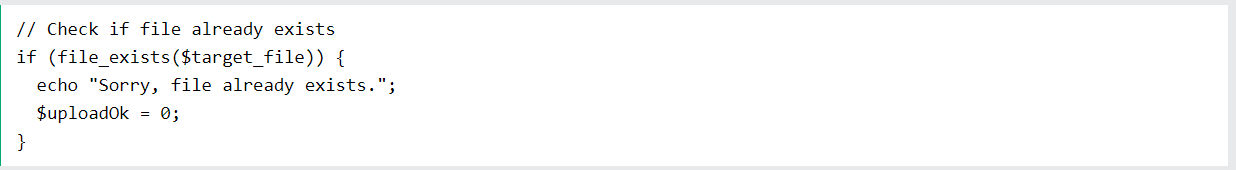
$target\_dir : dùng để xác định vị trí của file để chuyển tải dữ liệu vào.

$target\_file : chỉ định đường dẩn mà file đã được chuyển đến.

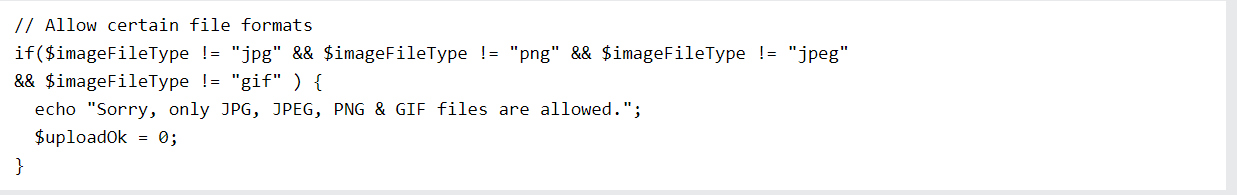
$uploadOk = 1 : dữ liệu chưa được upload(có thể được upload).

$imageFileType : giữ phần mở rộng của đường dẫn bằng chữ thường.

$check = getimagesize() nếu $check đúng thì đó là file thật còn sai thì đó là file giả.

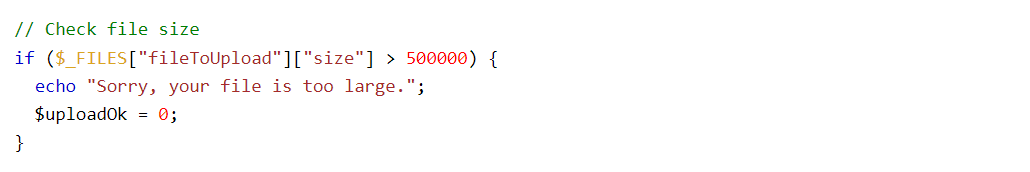
Nếu file là thật ta kiểm tra xem file đã tồn tại trong tệp uploads hay chưa: 

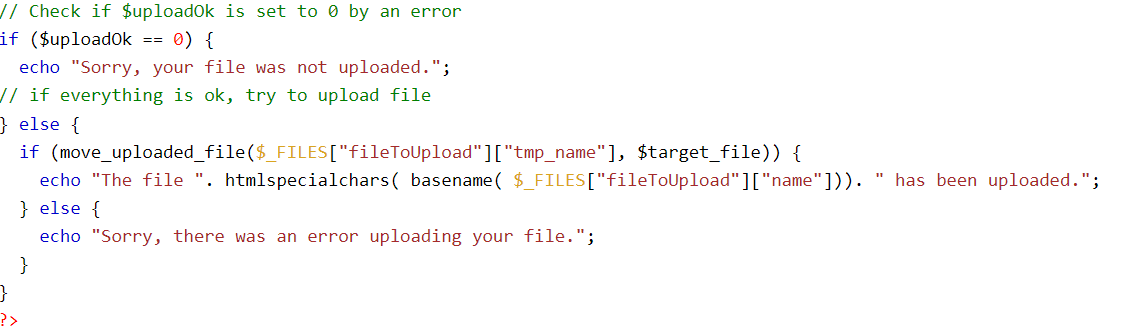
File\_exists(): file đã tồn tại hay chưa.

Nếu chưa ta sẽ kiểm tra file muốn có thuộc loại file ảnh hay không: 

$imageFileType : loại đuôi của tập tin ảnh(.jpg,.png,.jpeg,.gif)

Cuối cùng ta kiểm tra xem dung lương file muốn upload có quá lớn hay không:



Nếu tất cả điều ổn thì tiến hành chuyển file uploads vào tệp uploads: 

Move\_upload\_file(): chuyển file upload vào tệp chỉ định.

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

Filename:tên của file(bắt buộc phải có).

Flag: dùng để mô tả một hoặc nhiều hằng số.

FILE\_IGNORE\_NEW\_LINE: Bỏ qua dòng mới ở cuối mỗi phần tử mảng

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

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

Content: chỉ định bối cảnh xữ lý của tệp.

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 tậptin 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.
   * Tham khảo upload-img.php để thiết kế giao diện upload file
   * 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: ,. 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.