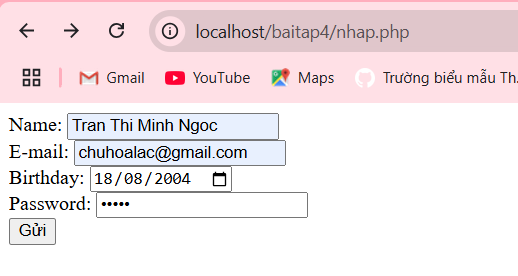
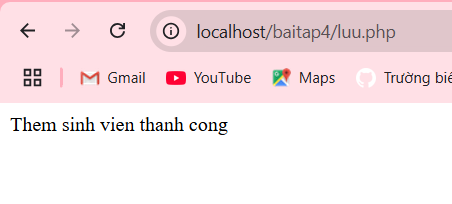
Câu 1:

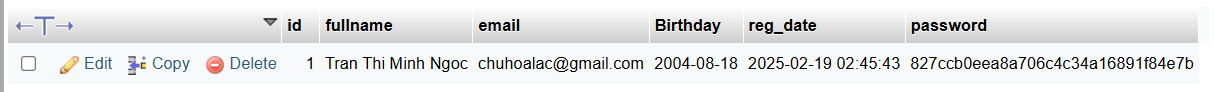
\_nhap.php



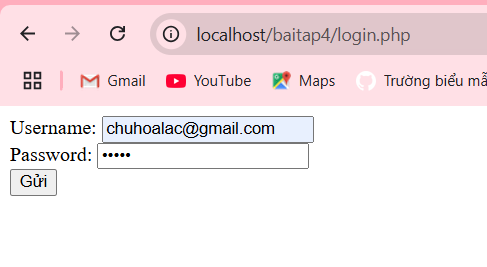
\_luu.php



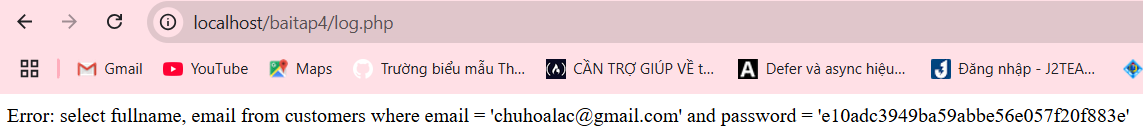
\_Lúc này trong cơ sở dữ liệu



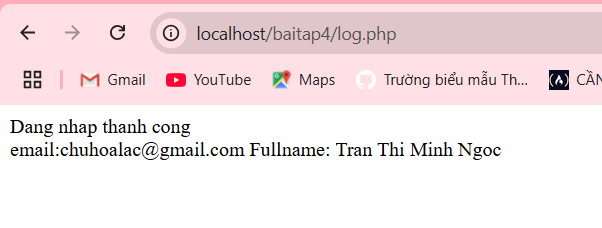
\_login.php



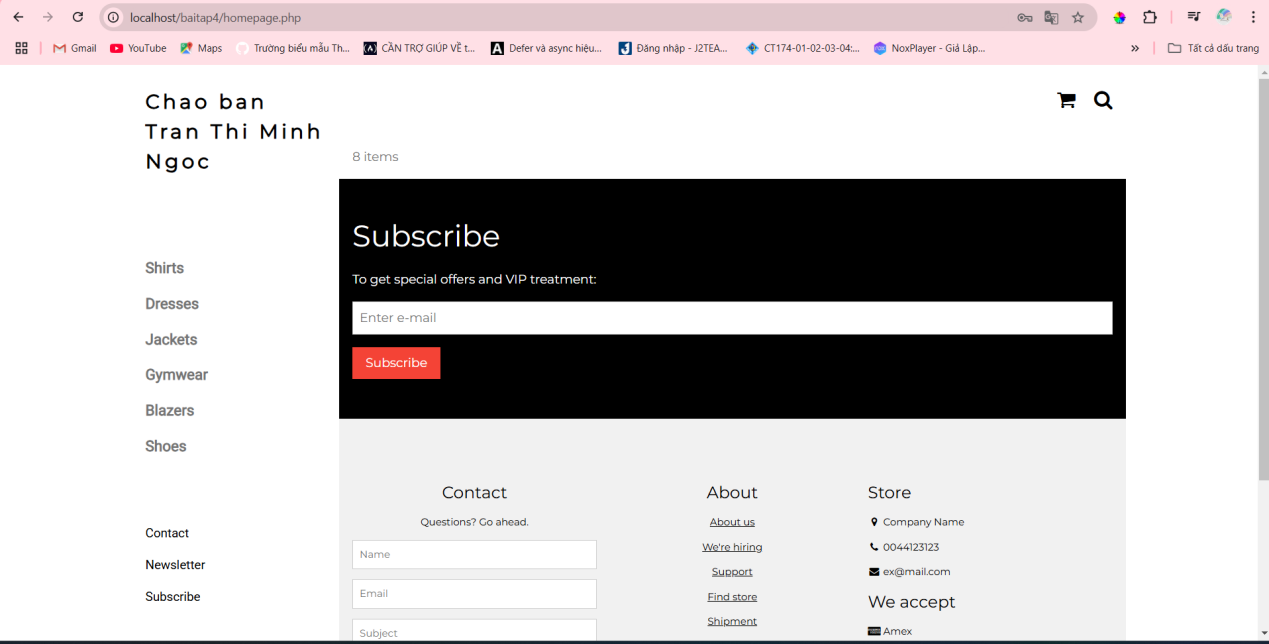
\_Nhập sai:



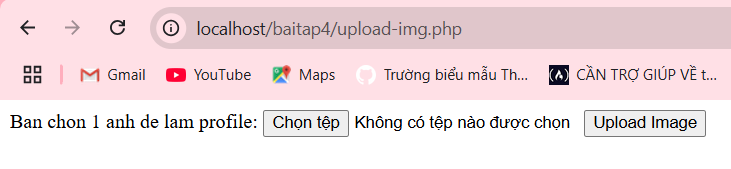
\_Nhập đúng:



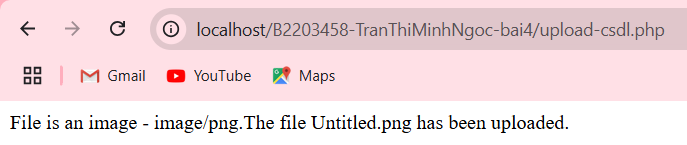
\_homepage.php



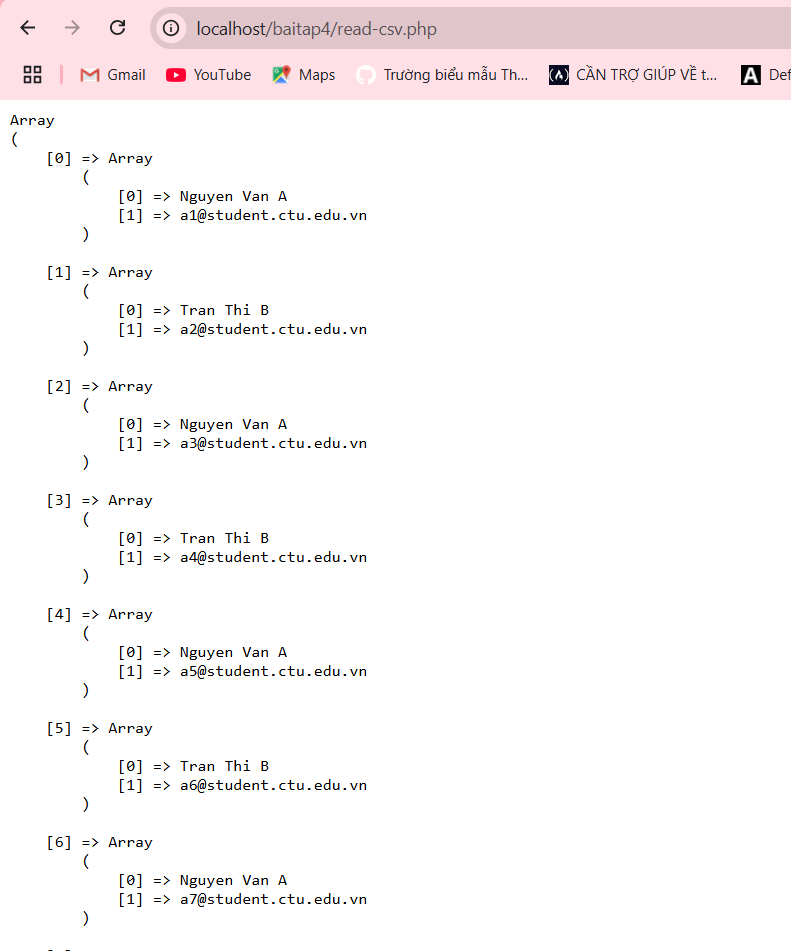
\_Upload-img.php



\_upload.php



\_read-csv.php

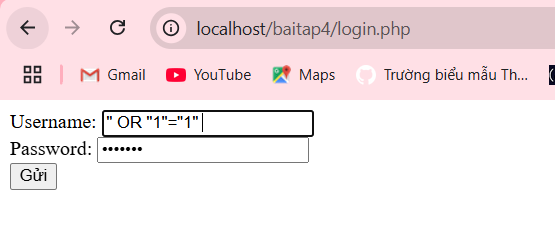


Câu 2: Trong vài trường hợp, hacker có thể sử dụng các kỹ thuật SQL Injection để hack hệ thống của bạn. Bạn hãy trình bày SQL Injection là gì và thử các ví dụ trình bày trong 1. Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.

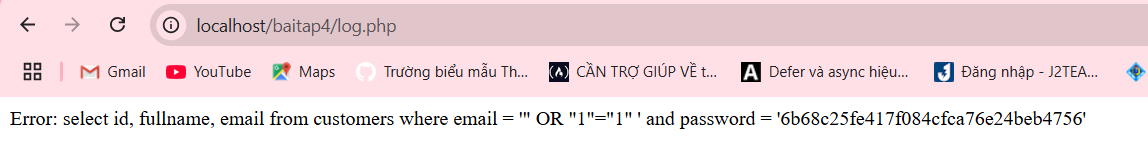
SQL Injection là kỹ thuật tấn công cho phép hacker chèn các câu lệnh SQL độc hại vào truy vấn của ứng dụng, từ đó có thể:

* Truy xuất dữ liệu trái phép
* Chỉnh sửa hoặc xóa dữ liệu
* Thậm chí chiếm quyền điều khiển hệ thống

Áp dụng với log.php



Thành công:



Câu 3: Dựa vào link 2, bạn hãy cho biết Cookie là gì, diễn giải ý nghĩa các tham số trong setcookie(). Cách lưu, lấy, xóa giá trị trong cookie.

Trả lời:

Cookie là một **tệp nhỏ** được trình duyệt lưu trữ trên máy tính của người dùng. Nó giúp **lưu thông tin tạm thời** như đăng nhập, giỏ hàng, tùy chỉnh trang web,…

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

Lý giải:

· **name**: Tên cookie (bắt buộc).

· **value**: Giá trị cookie (có thể là chuỗi, số, JSON).

· **expire**: Thời gian hết hạn (tính bằng giây kể từ thời điểm hiện tại).

+time() + 3600 → Cookie tồn tại **1 giờ**.

+time() - 3600 → Xóa cookie ngay lập tức.

· **path**: Đường dẫn có thể truy cập cookie (mặc định là /).

· **domain**: Miền có thể sử dụng cookie (ví dụ: .example.com để dùng cho tất cả subdomain).

· **secure**: Chỉ gửi cookie qua kết nối HTTPS (true/false).

· **httponly**: Nếu true, cookie chỉ truy cập được bằng PHP, không thể lấy từ JavaScript.

Các cách:

\_Lưu cookie:

//Tạo cookie username = JohnDoe và tồn tại trong **1 giờ**.

setcookie("username", "JohnDoe", time() + 3600, "/");

\_Lấy giá trị cookie

//Nếu cookie username tồn tại, nó sẽ hiển thị.

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

echo "Welcome " . $\_COOKIE["username"];

} else {

echo "Cookie not found!";

}

\_Xóa cookie:

//Đặt thời gian hết hạn về quá khứ để xóa cookie

setcookie("username", "", time() - 3600, "/");

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

Session (phiên làm việc) là một cơ chế **lưu trữ dữ liệu tạm thời trên máy chủ** trong suốt quá trình người dùng truy cập website.

Trước khi làm việc với session, cần gọi session\_start() ở đầu mỗi trang: session\_start();

\_Lưu

//Sau khi đặt session, nó sẽ tồn tại trên server cho đến khi người dùng đóng trình duyệt hoặc bị hủy.

session\_start();

$\_SESSION["username"] = "JohnDoe";

$\_SESSION["role"] = "admin";

\_Lấy:

//Kiểm tra xem session có tồn tại không trước khi truy xuất.

session\_start();

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

echo "Welcome " . $\_SESSION["username"];

} else {

echo "Session not found!";

}

\_Xóa:

//Chỉ xóa session username, các session khác vẫn còn.

session\_start();

unset($\_SESSION["username"]);

//hủy toàn bộ

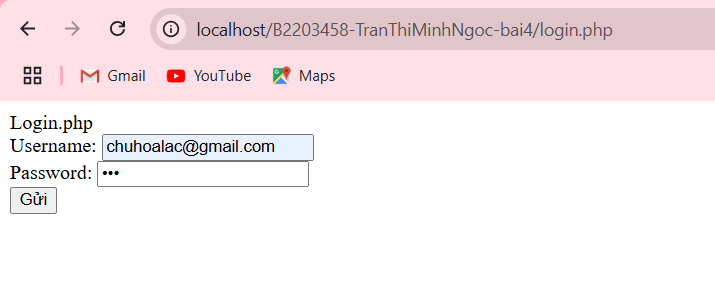
session\_start();

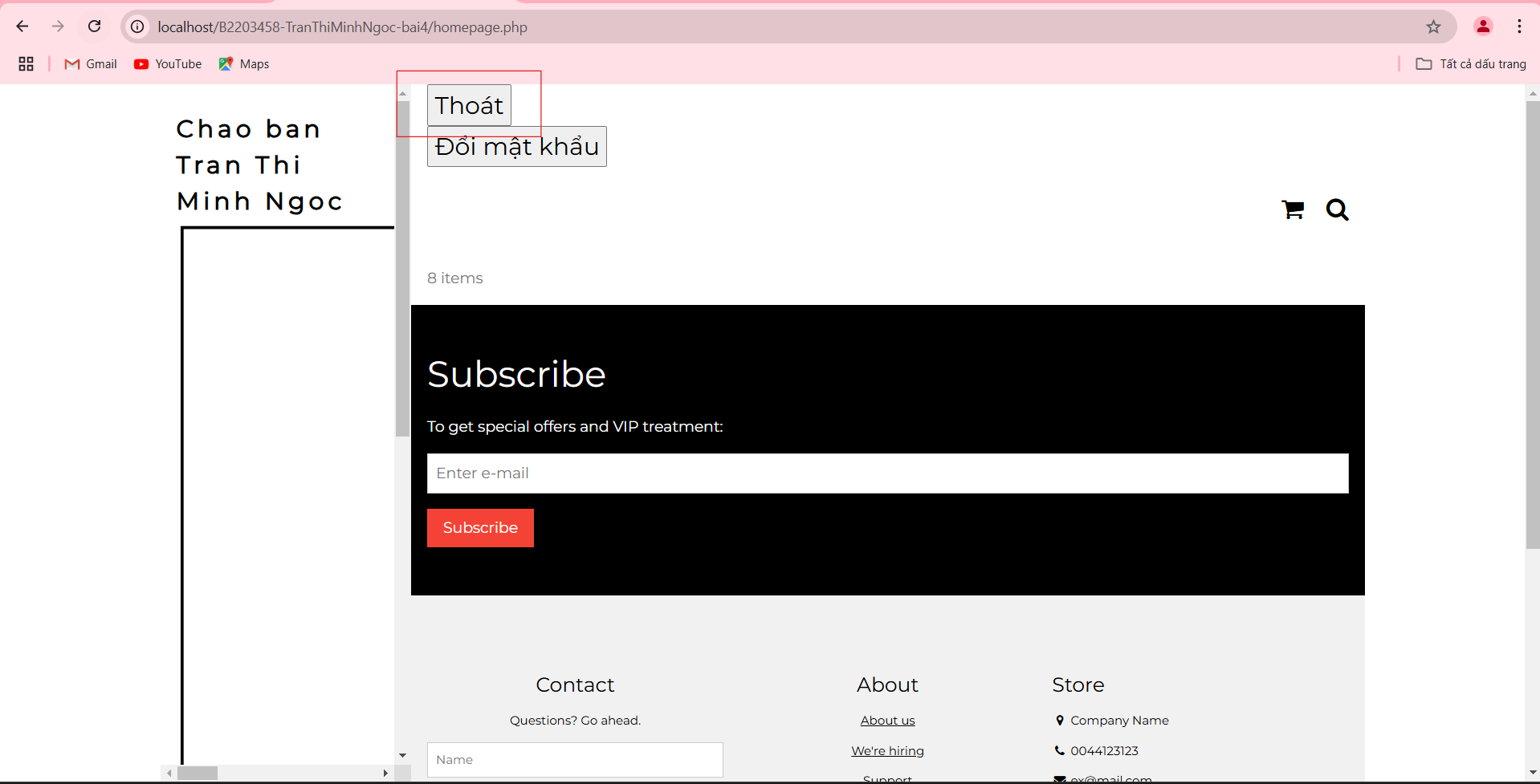
session\_destroy();

Câu 5: Bạn hãy so sánh Cookie và Session.

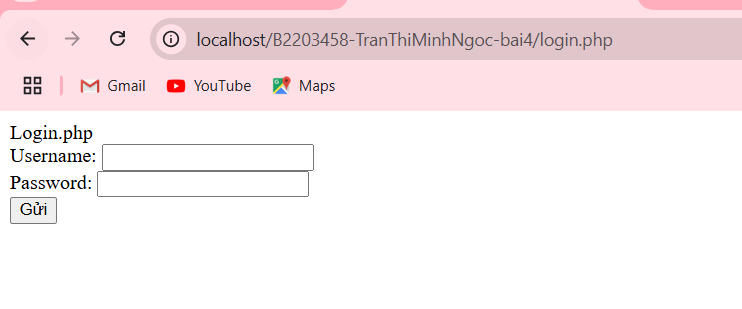
|  |  |  |
| --- | --- | --- |
| Tiêu chí | Cookie | Session |
| Vị trí lưu trữ | Trình duyệt (Client-side) | Máy chủ (server-side) |
| Bảo mật | Thấp(dễ bị đánh cắ, chỉnh sửa) | Cao(dữ liệu chỉ có trên server) |
| Dung lượng | Giới hạn 4KB | Cao (dữ liệu chỉ có trên server) |
| Thời gian tồn tại | Có thể đặt thời gian hết hạn | Hết khi đóng trình duyệt (hoặc session timeout) |
| Tốc độ xử lí | Nhanh hơn (vì lưu trên client) | Chậm hơn (vì xử lý trên server) |
| Ứng dụng | Ghi nhớ thông tin lâu dài (đăng nhập, cài đặt người dùng) | Lưu dữ liệu tạm thời (giỏ hàng, đăng nhập phiên làm việc) |
| Cách lấy dữ liệu | $\_COOKIE["name"] | $\_SESSION["name"] |
| Cách xóa | setcookie("name", "", time() - 3600, "/"); | unset($\_SESSION["name"]); hoặc session\_destroy(); |

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.

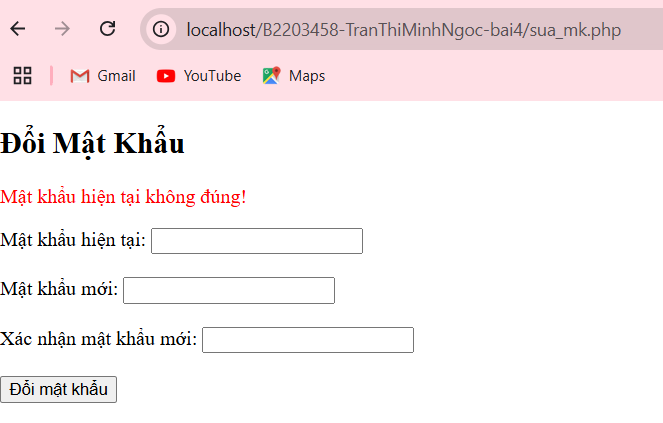


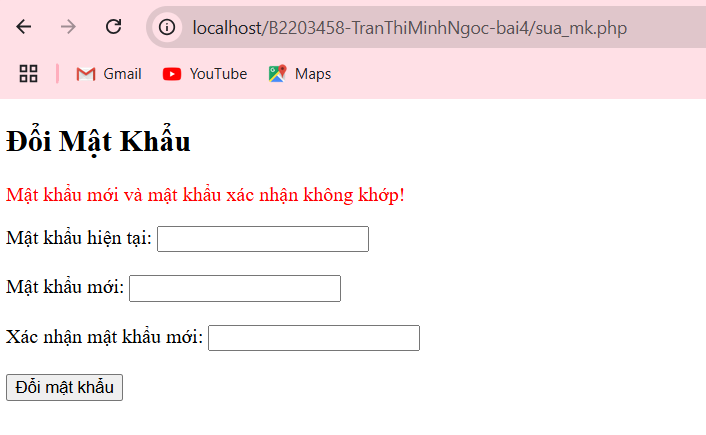


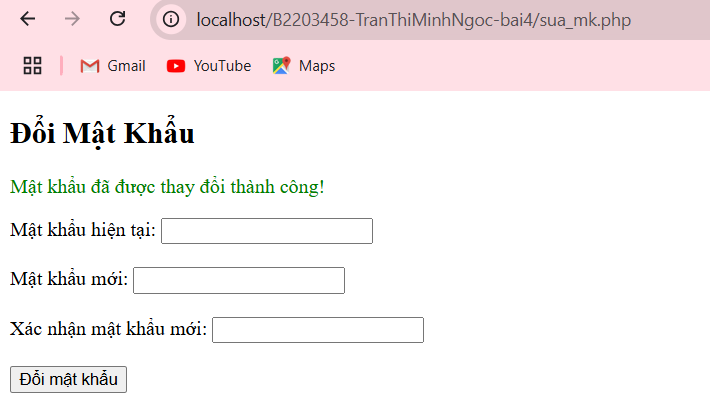
\_Khi nhấn nút thoát quay về trang login.php



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 4 để tìm hiểu và mô tả các cơ chế, các hàm thủ tục để thực hiện việc upload.

· **Người dùng chọn file từ máy tính** và gửi lên server qua <input type="file">.

· **File tạm thời được lưu trên server** trong thư mục tạm ($\_FILES chứa thông tin file).

· **Kiểm tra file hợp lệ** (dung lượng, loại file, lỗi...).

· **Di chuyển file vào thư mục đích** bằng move\_uploaded\_file().

Các hàm thủ tục để thực hiện việc upload:

move\_uploaded\_file($tmp\_name, $target\_file): Di chuyển file từ thư mục tạm sang thư mục đích

basename($path): Lấy tên file từ đường dẫn

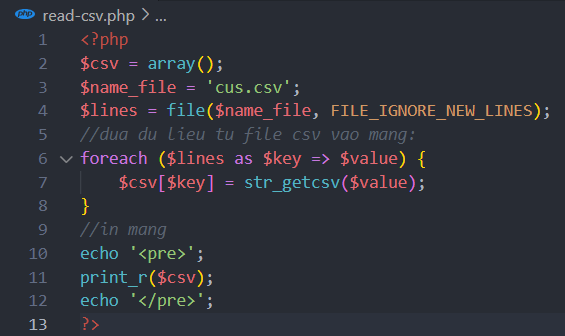
pathinfo($path, PATHINFO\_EXTENSION): Lấy phần mở rộng file

$\_FILES["fileToUpload"]["size"]: Kiểm tra dung lượng file

$\_FILES["fileToUpload"]["error"]: Kiểm tra lỗi khi upload

Câu 9: Bạn hãy đọc 5, mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).

Mô tả chức năng của hàm này và các tham số trong hàm:



Hàm file() đọc một tệp vào một mảng. Cú pháp IMG_257

· $filename: Tên hoặc đường dẫn của tệp cần đọc.

· $flags (tùy chọn): Điều chỉnh cách thức xử lý tệp. Một số giá trị phổ biến:

* FILE\_USE\_INCLUDE\_PATH: Tìm kiếm tệp trong các đường dẫn được chỉ định trong include\_path của php.ini.
* FILE\_IGNORE\_NEW\_LINES: Loại bỏ ký tự xuống dòng (\n, \r\n) khi lưu dữ liệu vào mảng.
* FILE\_SKIP\_EMPTY\_LINES: Bỏ qua các dòng trống.

· $context (tùy chọn): Xác định cách thức xử lý luồng tệp. Nếu không cần, có thể truyền NULL.

str\_getcsv($value) là một hàm trong PHP dùng để phân tách một chuỗi (dòng văn bản) thành một mảng con theo định dạng CSV (Comma-Separated Values).

print\_r($csv) sẽ in ra nội dung của mảng $csv. Sau khi xử lý xong, mảng này sẽ chứa các mảng con (mỗi mảng con đại diện cho một dòng trong tệp CSV).

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 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.

○ 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. 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.

cus.csv

