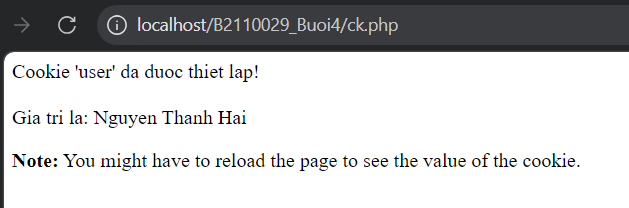
MSSV: B2110029

Họ và tên: Huỳnh Nguyễn Trường Thịnh

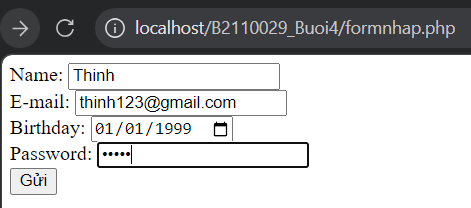
**BUỔI THỰC HÀNH 4**

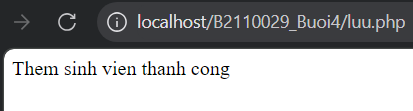
**1.**

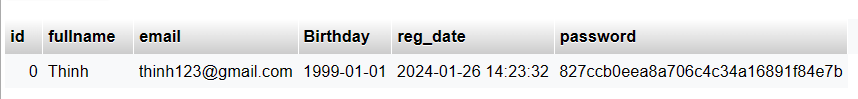
**Tạo Cookies**



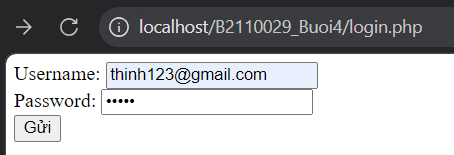
**Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa**

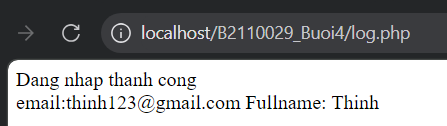




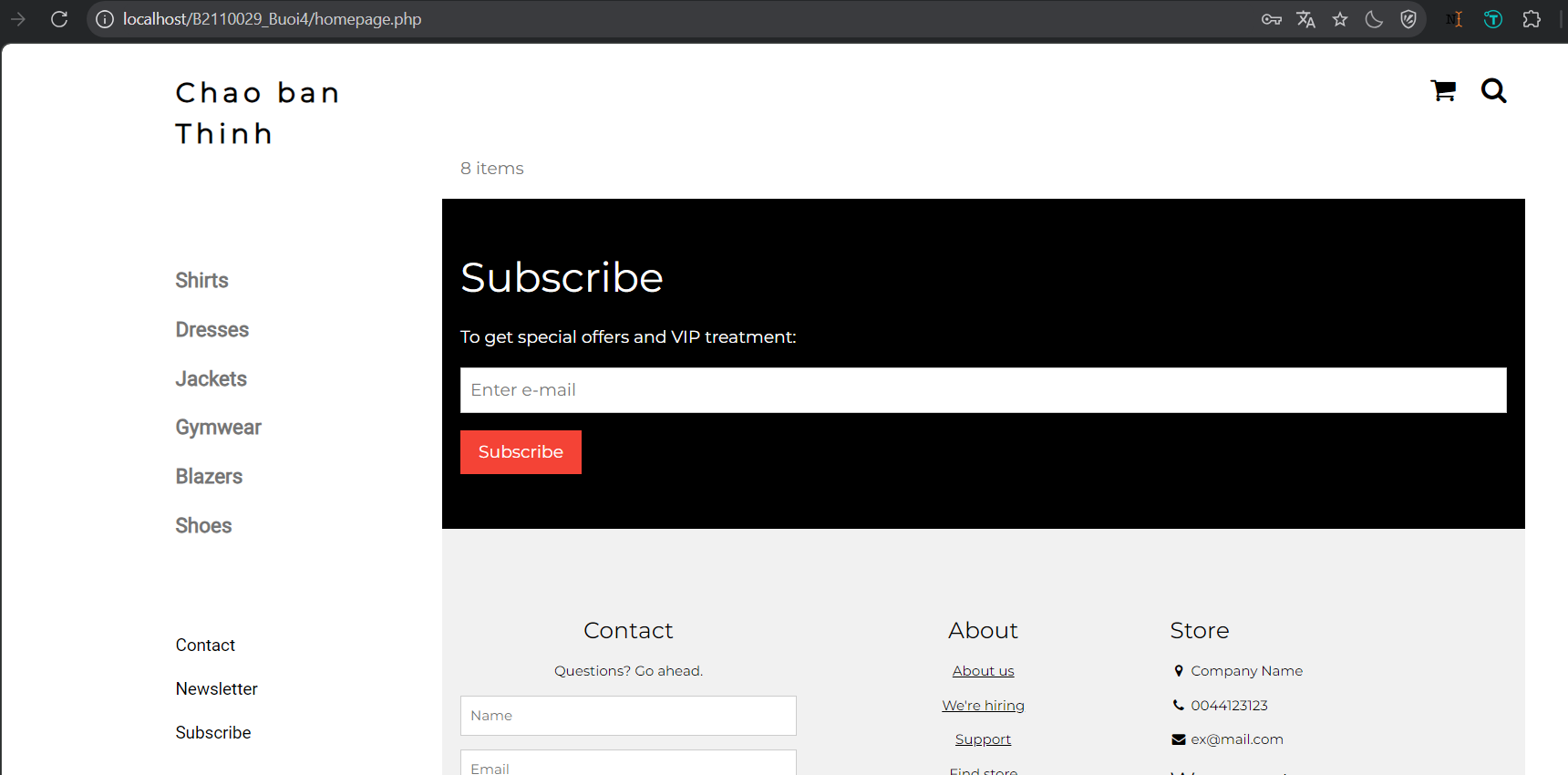


**Tạo form đăng nhập**

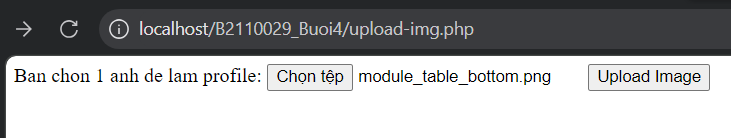


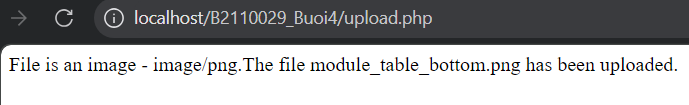


**Dùng Cookies lưu lại tên đăng nhập, email và điều hướng**

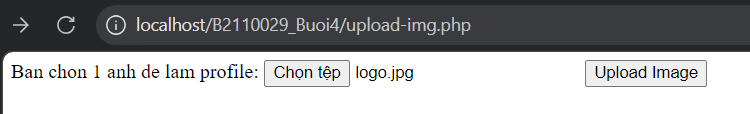


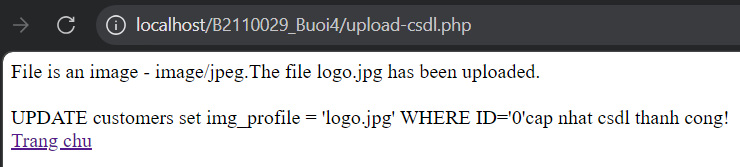
**Upload tập tin**

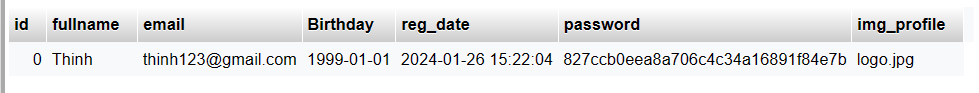


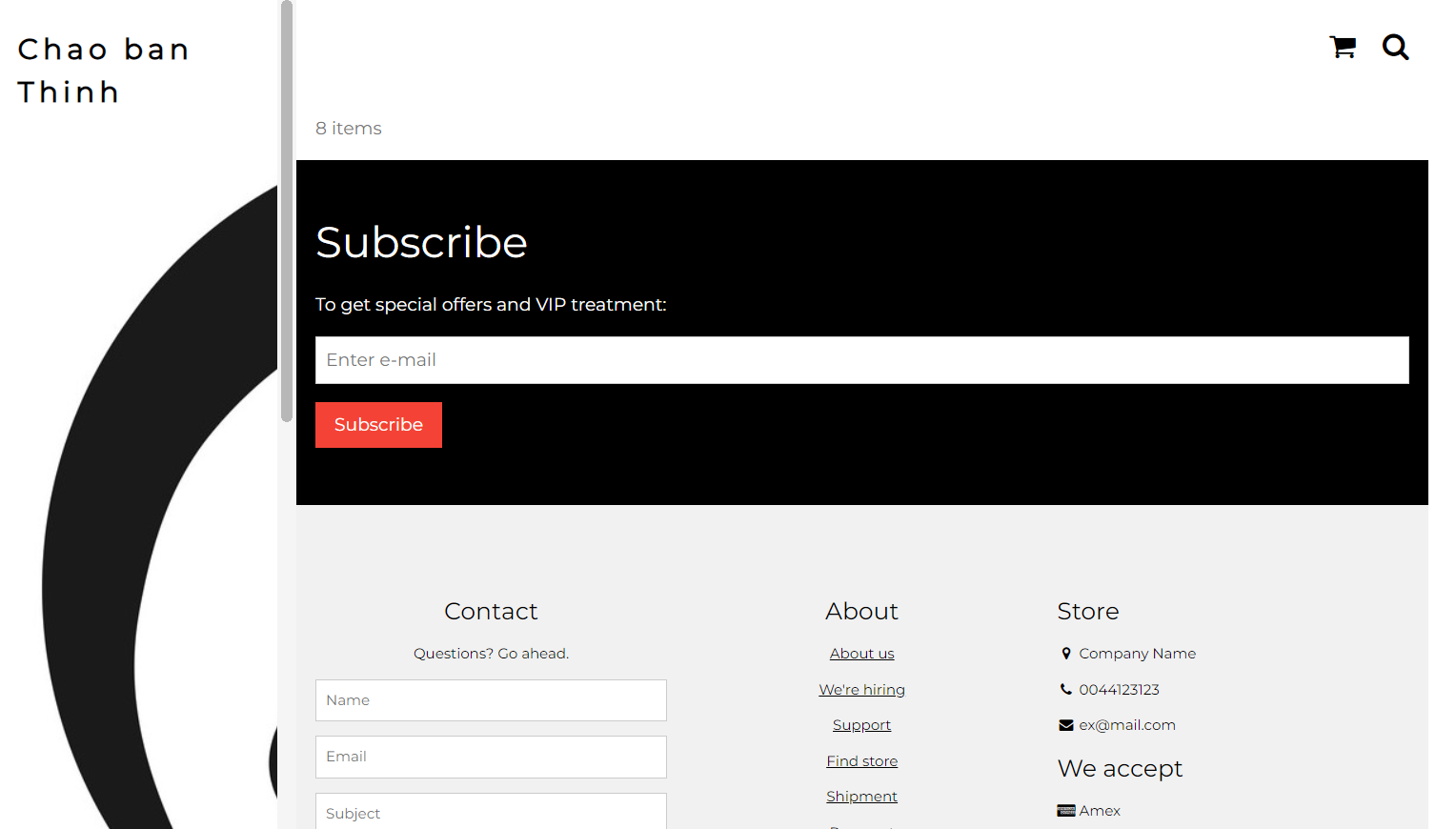


**Upload ảnh và lưu dữ liệu về ảnh trong CSDL**

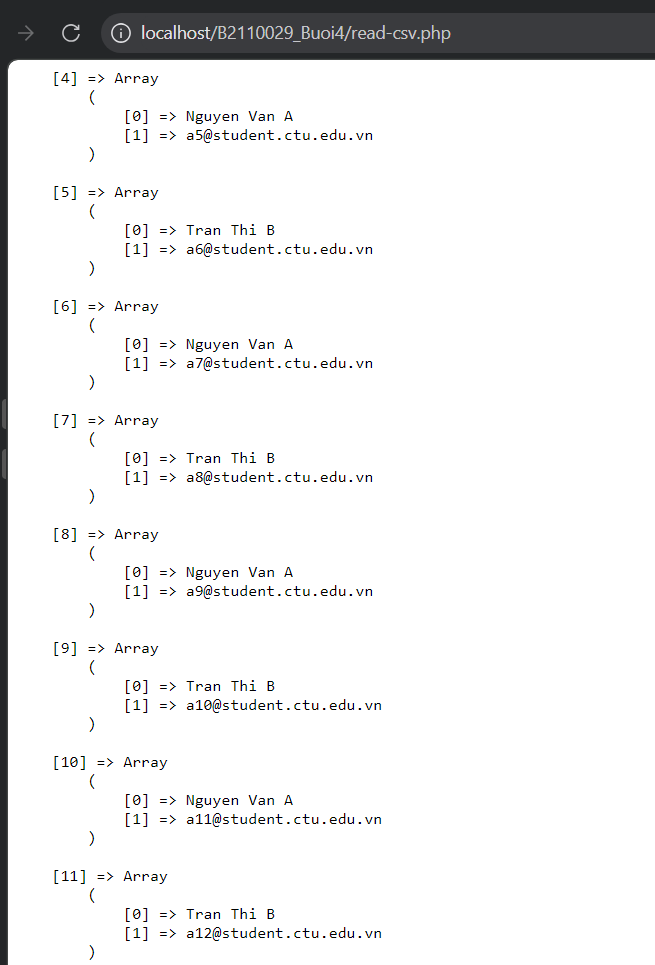
****

****

****

****

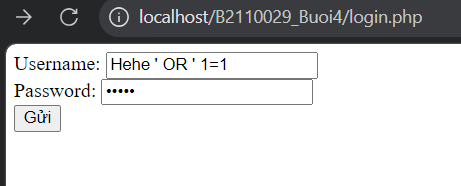
**Upload file csv và đọc dữ liệu vào mảng**

****

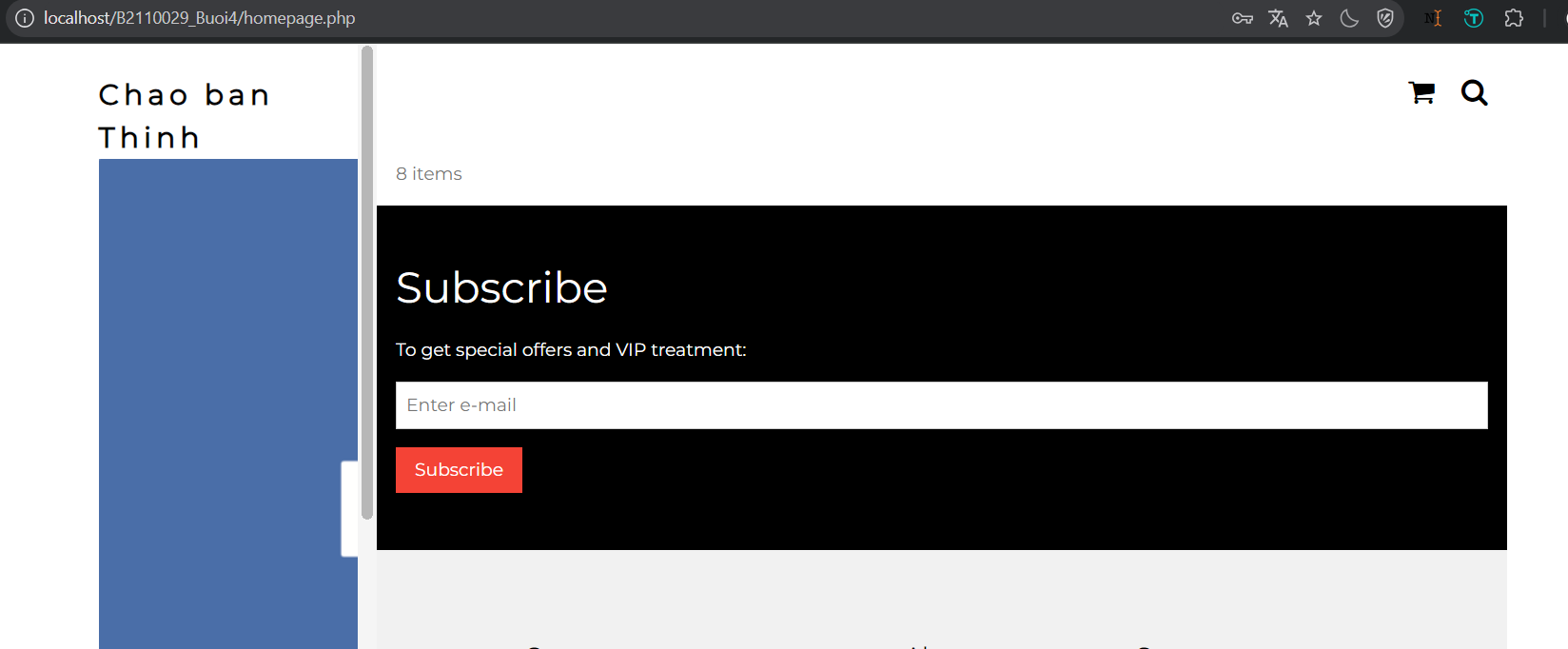
**2.**

- SQL injection là một kỹ thuật cho phép những kẻ tấn công lợi dụng lỗ hổng của việc kiểm tra dữ liệu đầu vào để inject và thi hành các câu lệnh SQL bất hợp pháp.

Ví dụ: Khi ta đăng nhập vào tài khoản [thinh123@gmail.com](mailto:thinh123@gmail.com) (có mật khẩu là 12345)

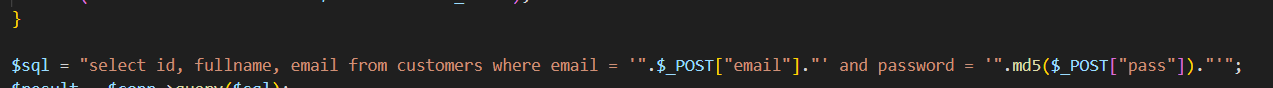


Ta sẽ để username là Hehe ' OR ' 1=1, kết quả chạy như sau :



->Giải thích cơ chế hack :

Thực chất, khi ta đăng nhập ở trang login.php, khi đăng nhập xong sẽ chuyển qua trang log.php. Tại đây sẽ xác nhận tài khoản và mật khẩu nhập vào rồi mới quyết định việc có cho người dùng vào hệ thống hay không. Tuy nhiên, trong code của log.php có đoạn như sau :



Hacker có thể lợi dụng kẻ hở nơi username nhập vào và chèn đoạn « Hehe ' OR ' 1=1 »

Lúc này, câu lệnh truy vấn sẽ bị đổi thành

« select id, fullname, email from customers where email = *Hehe OR 1=1* and password = 12345.

Vì 1=1 luôn là kết quả TRUE nên câu lệnh truy vấn trên luôn hợp lệ=>Đăng nhập vào hệ thống thành công.

**3.**

- 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.

setcookie(name, value, expire, path, domain, secure, httponly): name là bắt buộc, các tham số khác là tùy chọn

- Diễn giải các tham số trong hàm setcookie:

+ name: tên.

+ value: giá trị.

+ expire: thời hạn.

+ path: đường dẫn.

+ 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.

- Lưu: từ PHP có thể thiết lập Cookie bằng hàm setcookie (phải xuất hiện trước thẻ html).

- Lấy: dùng biến $\_COOKIE để truy cập thông tin cookie (có thể dùng isset() để kiểm tra sự tồn tại của COOKIE).

- Xóa: sử dụng chính hàm setcookie với thủ thuật đặt thời gian đã hết hạn.

VD: setcookie( "$cookie\_name", "$cookie\_value", time() - 60, "/","", 0);

**4.**

- Session dùng để lưu trữ thông tin (trong các biến) sẽ được sử dụng trên nhiều trang. Không giống như cookie, thông tin không được lưu trữ trên máy tính của người dùng.

- Lưu: một phiên được bắt đầu với session\_start() chức năng. Các biến session được đặt bằng biến toàn cục PHP: $\_SESSION.

- Lấy: dùng biến $\_SESSION để truy cập thông tin session.

- Xóa: để xóa session sử dụng hàm session\_unset() và session\_destroy() .

**5.**

- Về Cookie:

+ Cookie được lưu trữ trên trình duyệt người dùng.

+ Cookie dược lưu trữ ở phía máy khách.

+ Cookie dễ dàng sửa đổi vì được lưu trữ ở phía máy khách.

+ Dữ liệu có sẵn trong trình duyệt chạy đến khi hết hạn.

- Về Session:

+ Session không được lưu trữ trên trình duyệt người dùng.

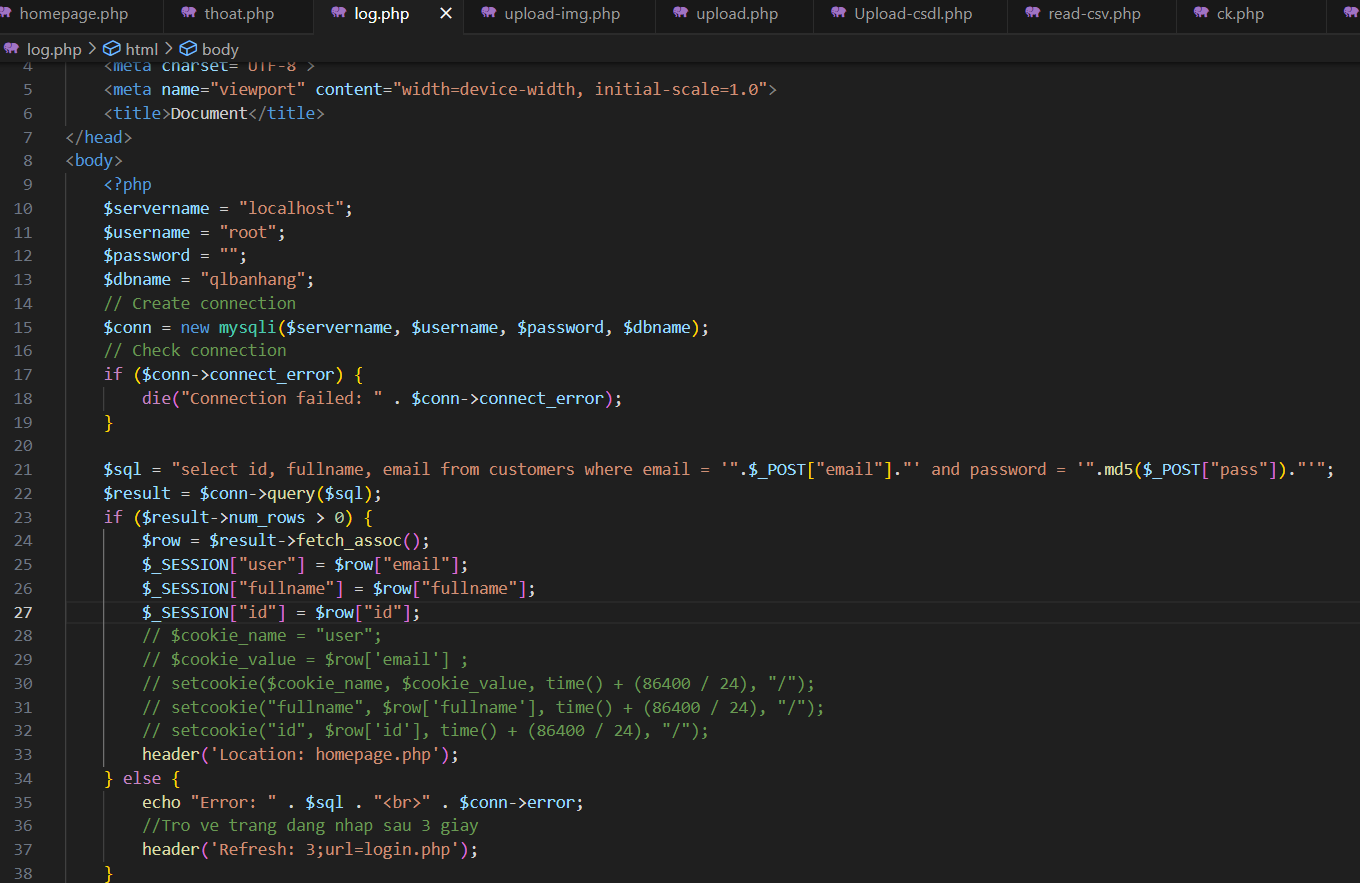
+ Session được lưu trữ ở phía máy chủ.

+ Session không dễ dàng sửa đổi vì được lưu trữ ở phía máy chủ.

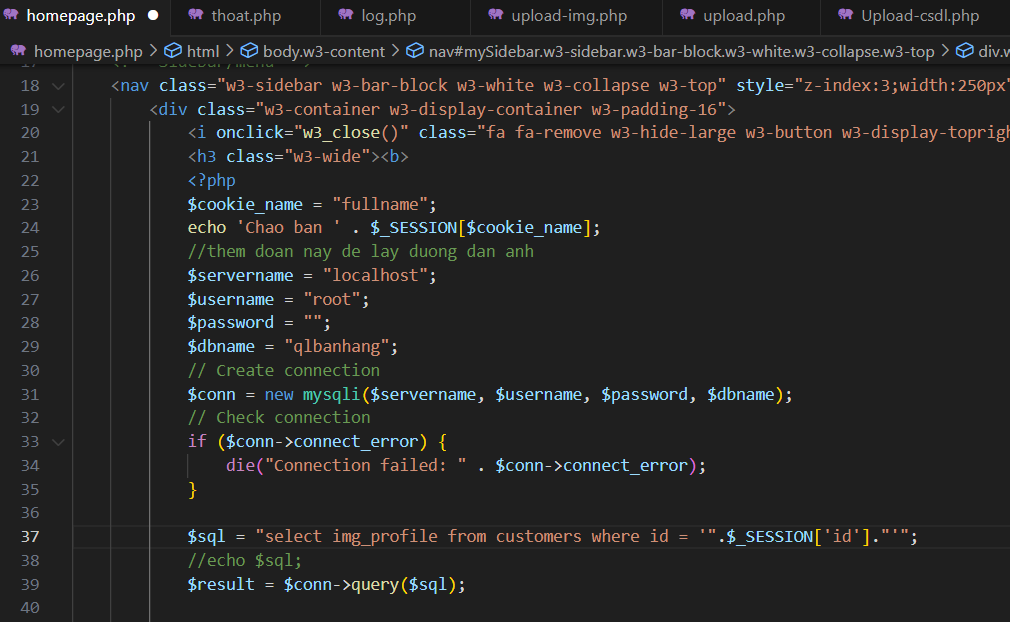
+ Dữ liệu có sẵn cho trình duyệt chạy, sẽ mất thông tin session khi đóng trình duyệt.

**6.**

- Sửa trong log.php:



- Sửa trong homepage.php:



- Tạo thoat.php:



**7.**

**8.**

**9.**

**10.**