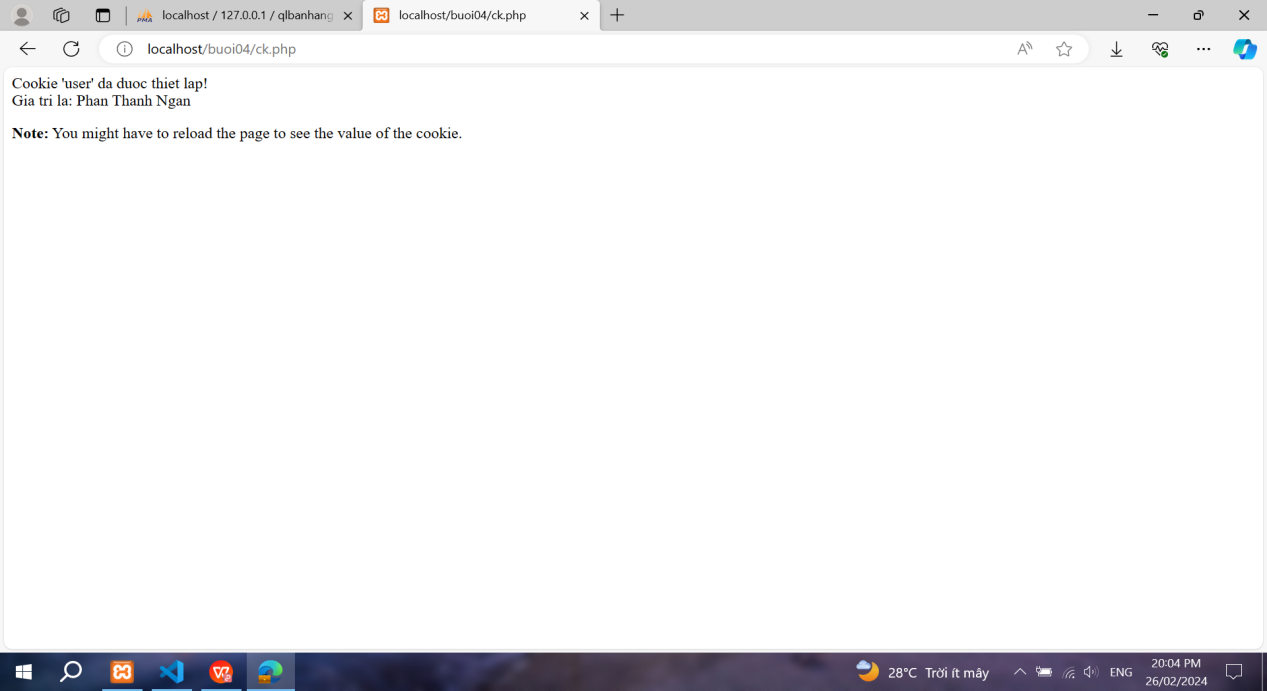
**Bài tập thực hành 04:**

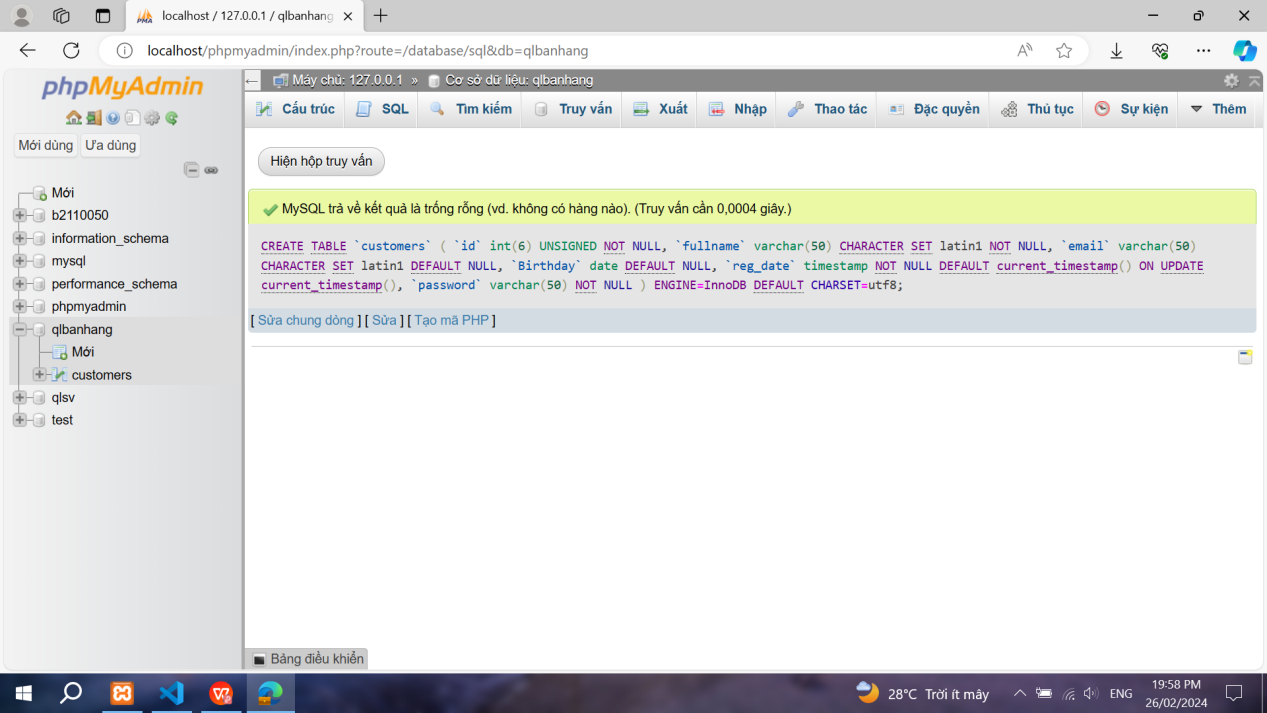
**LẬP TRÌNH PHP NÂNG CAO**

**Câu 1:**

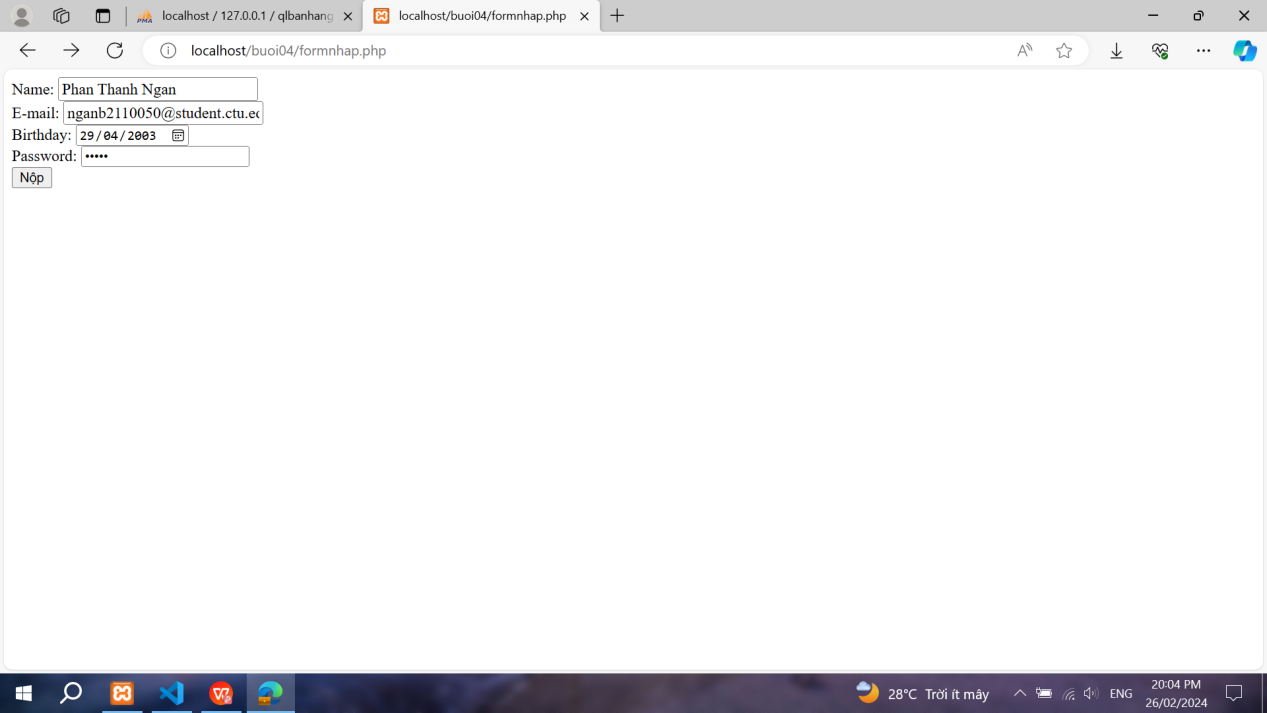
- Tạo ck.php:



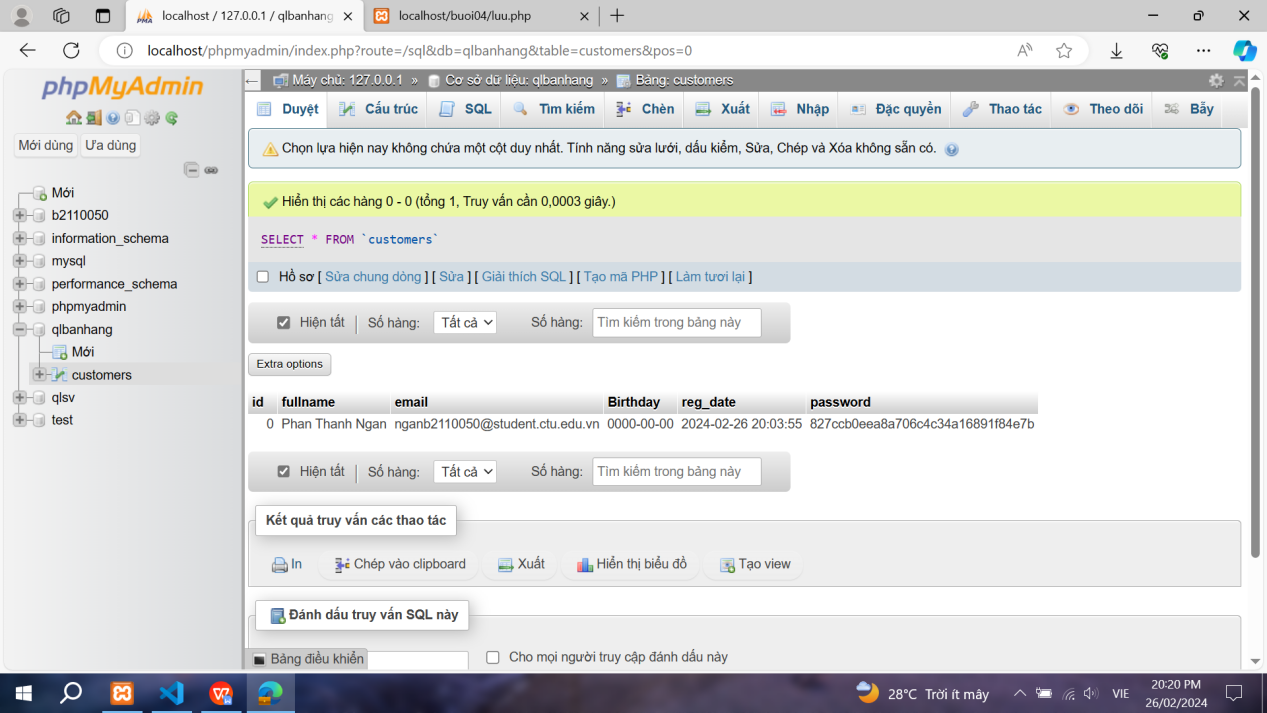
- Tạo csdl qlbanhang với bảng customers:



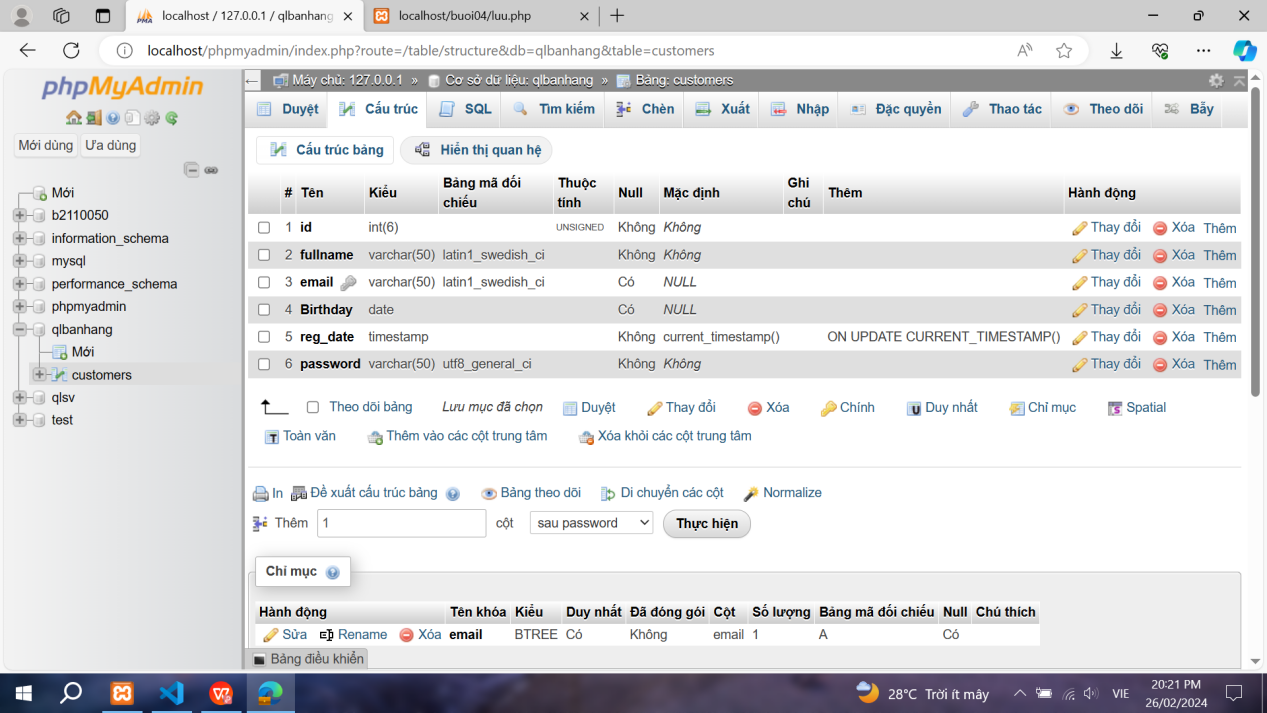
- Tạo formnhap.php:



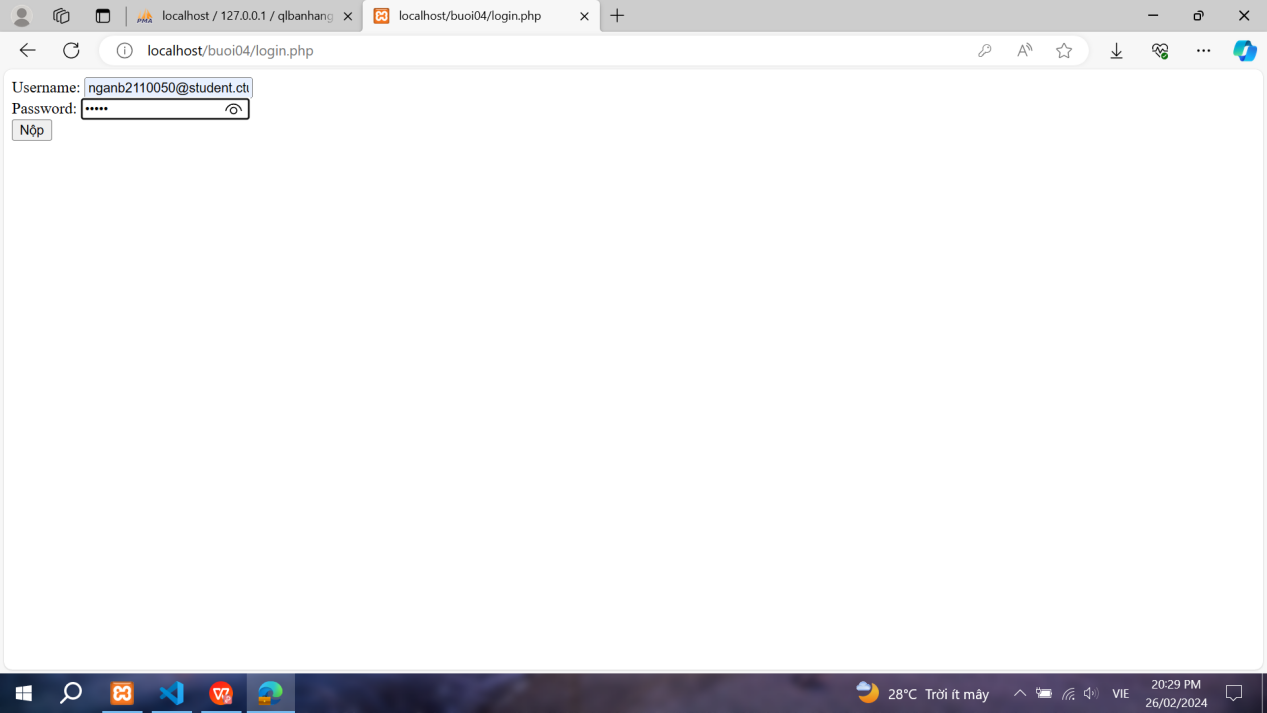
- Tạo luu.php:



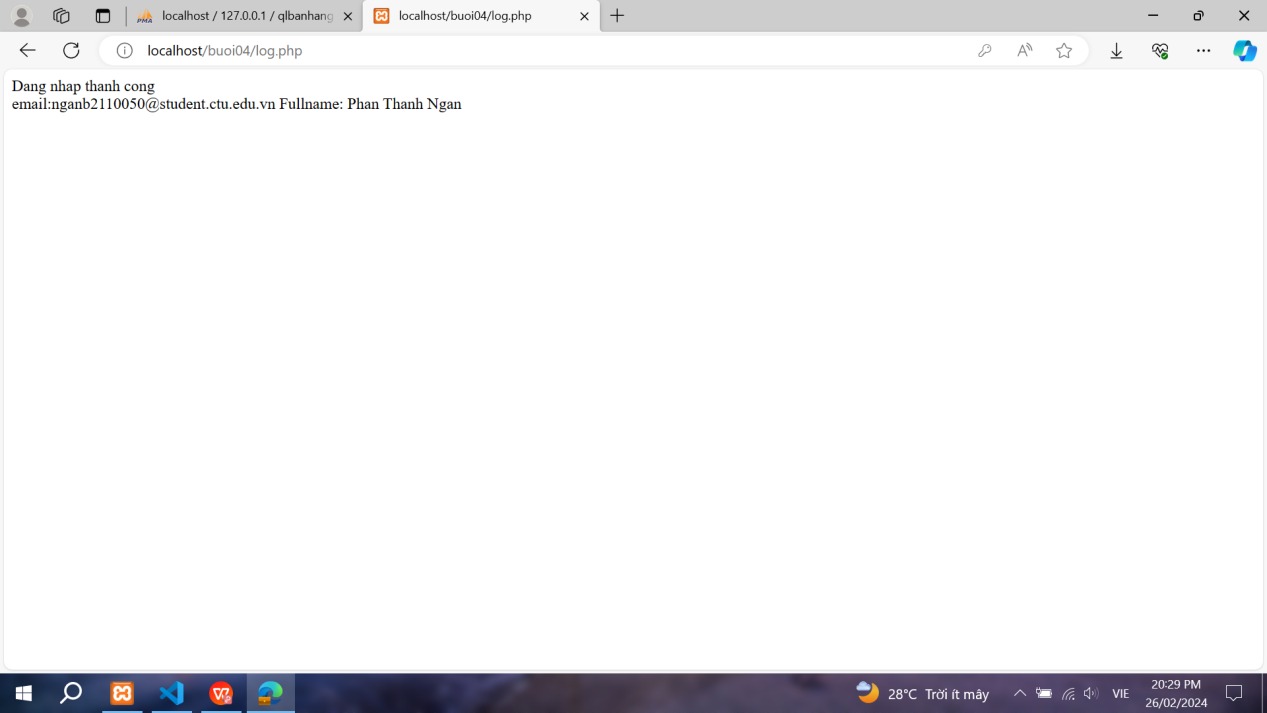
- chọn email là unique key:



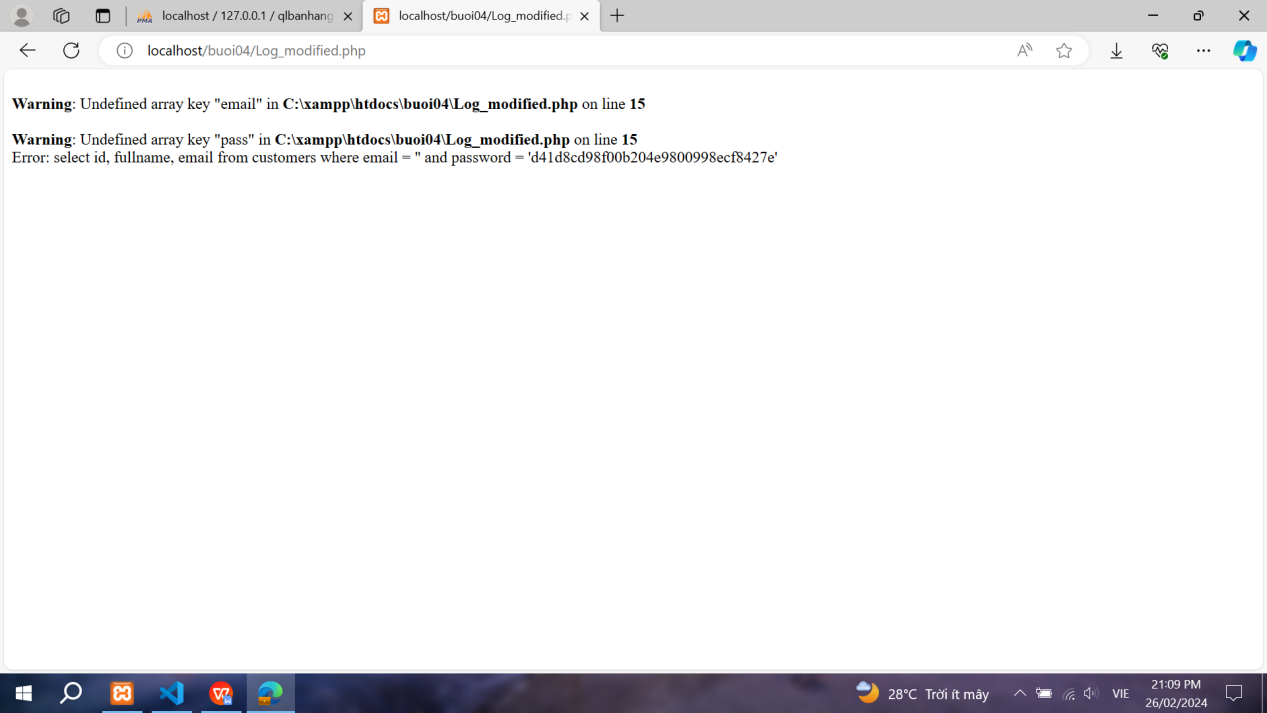
- Tạo login.php:

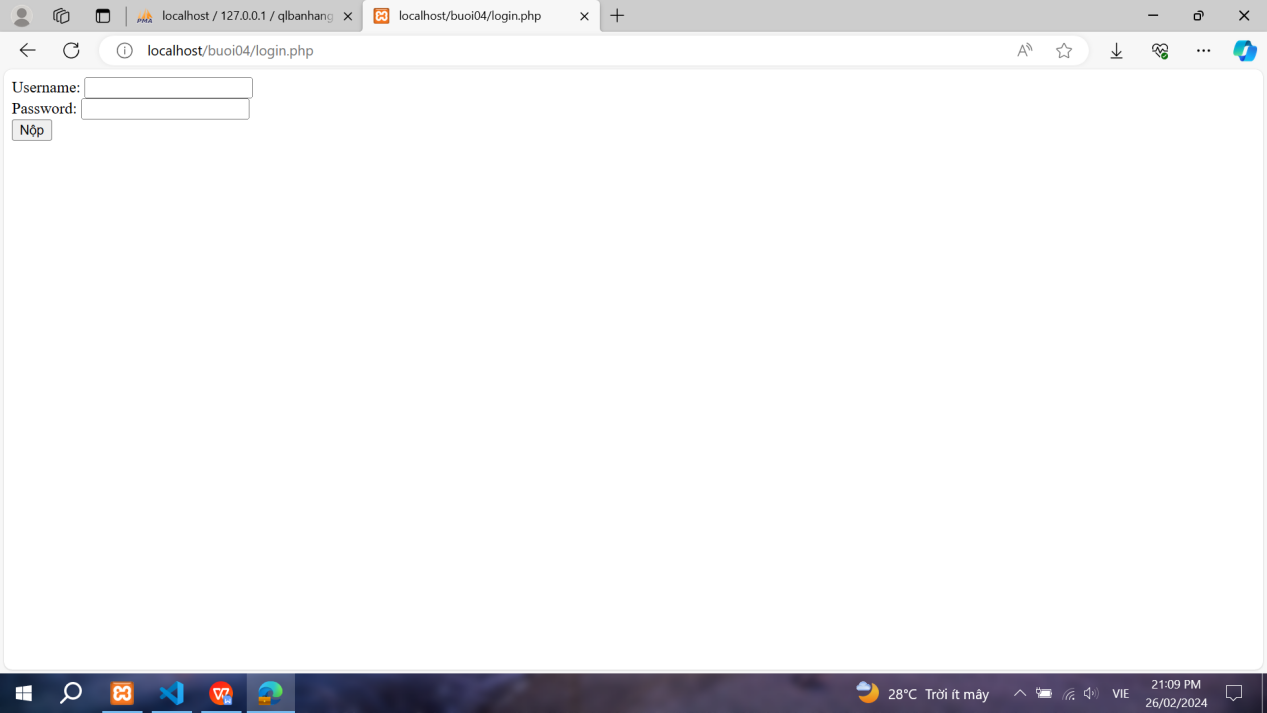


- Tạo log.php

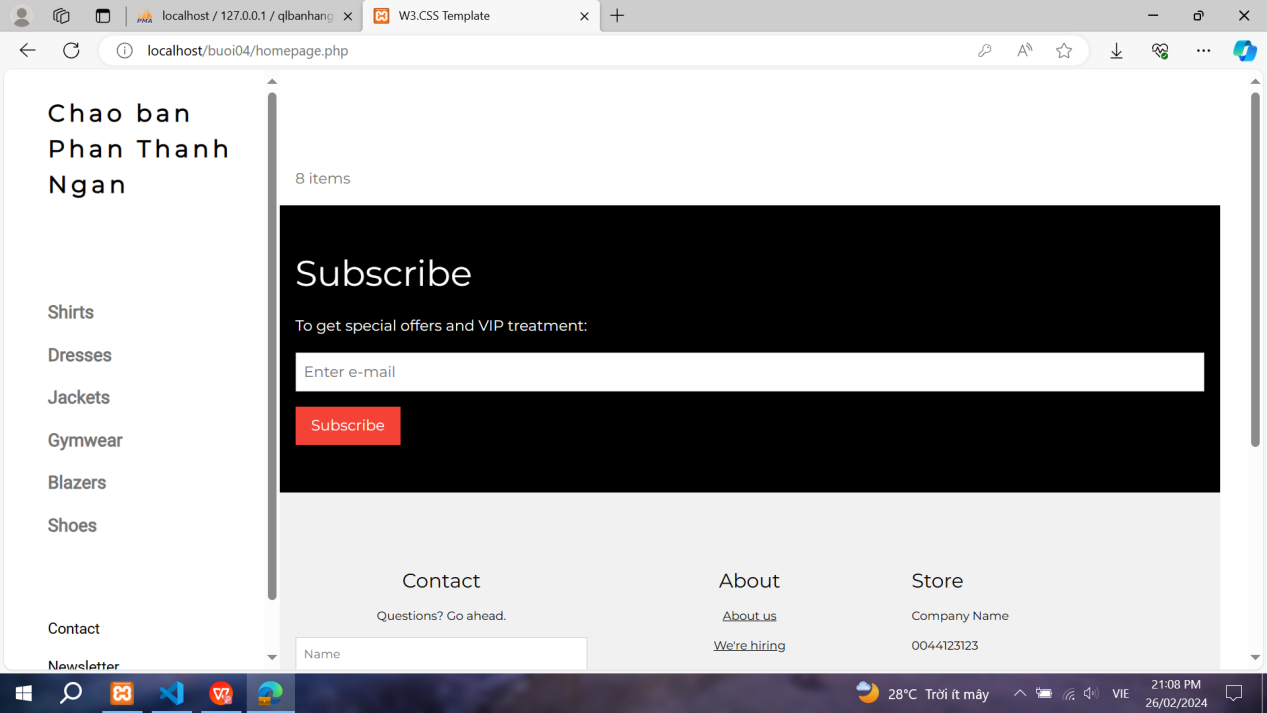


- Tạo log\_modified.php:

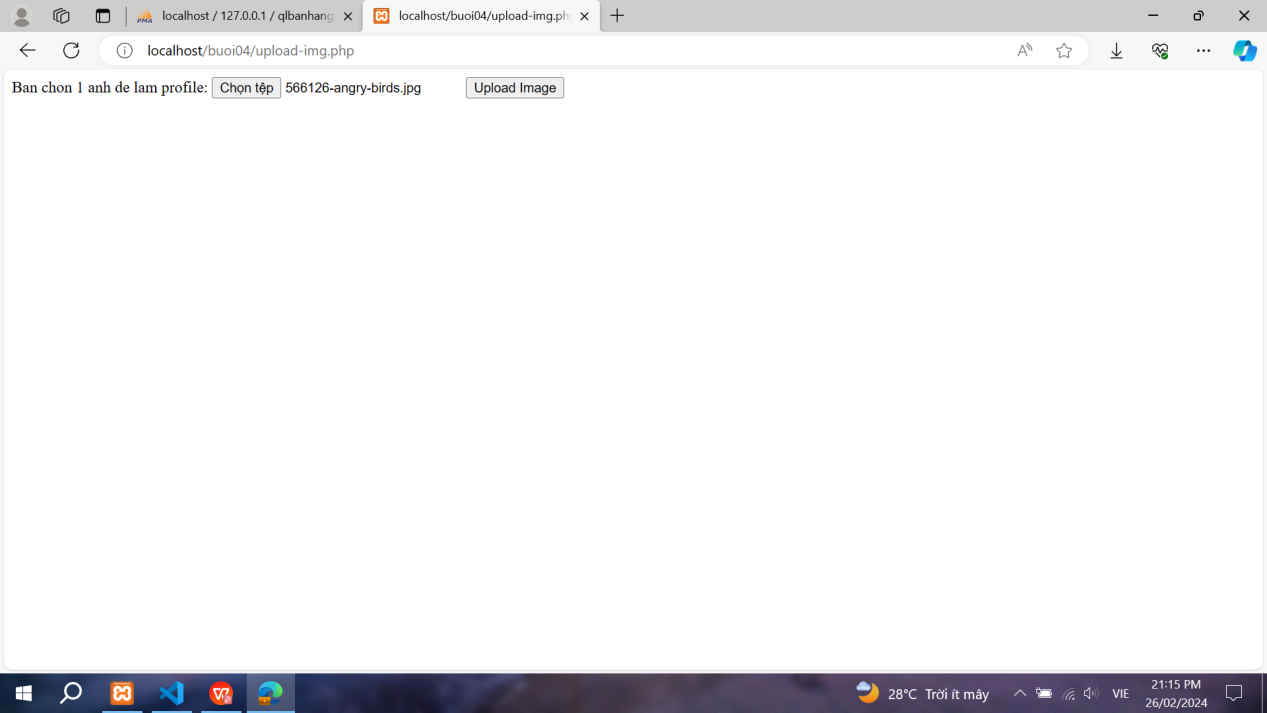




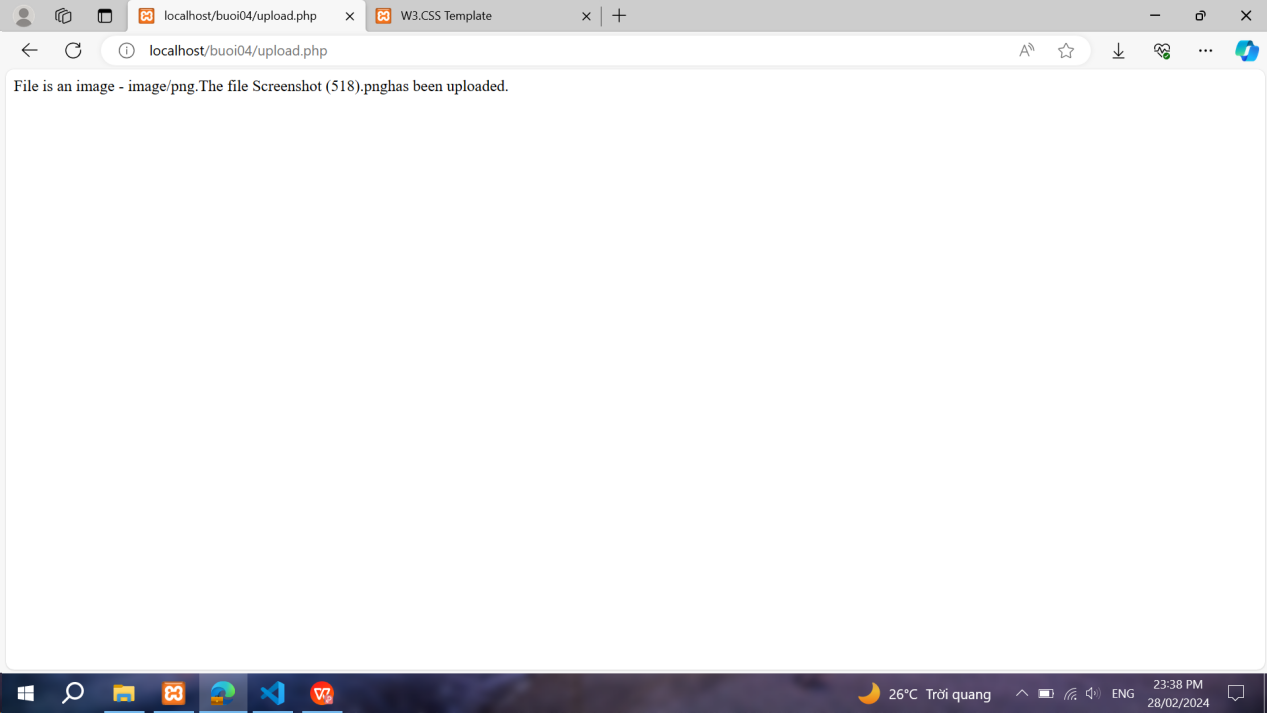
- Tạo homepage.php:



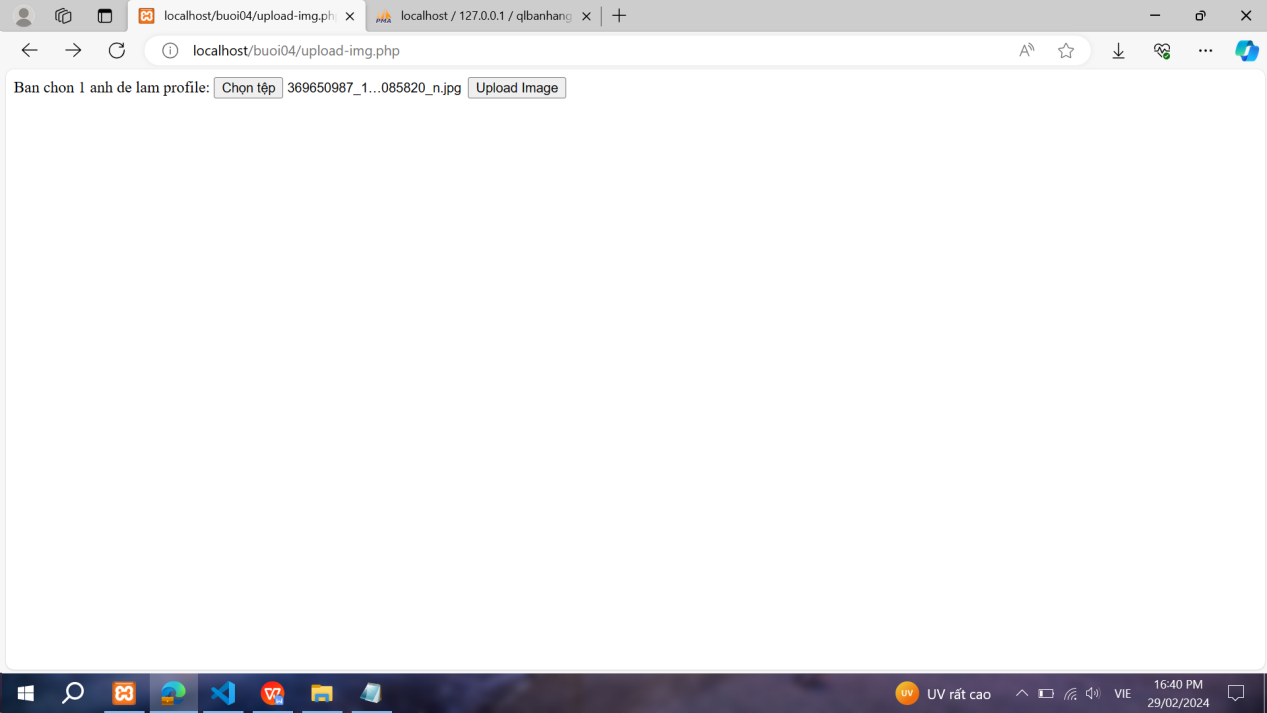
- Tạo upload-img.php:



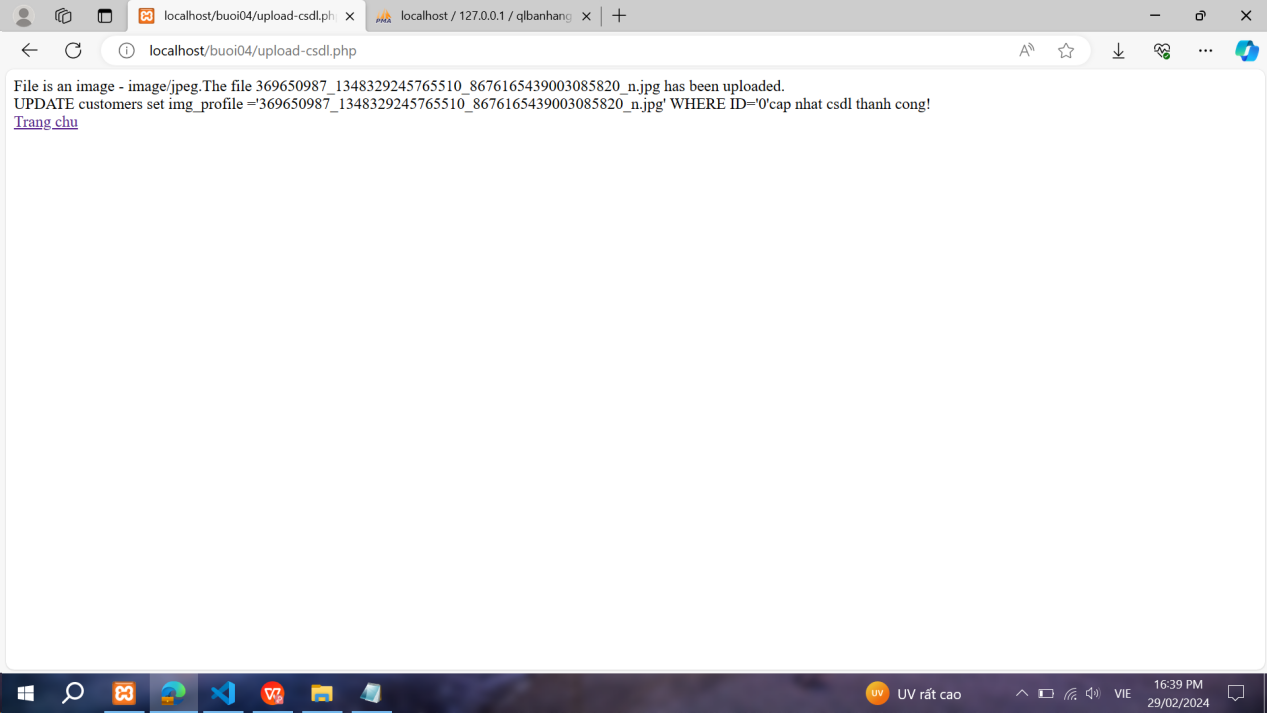
- Tạo upload.php:



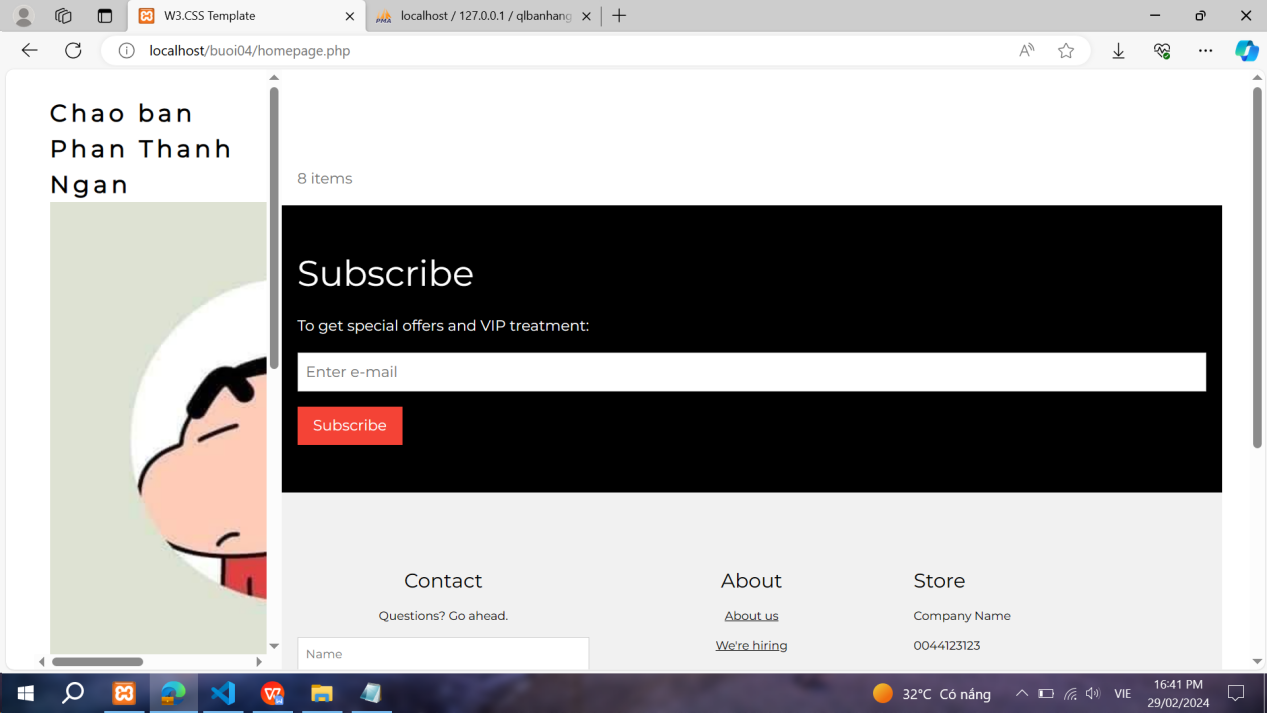
- Tạo Upload-img.php (modified):



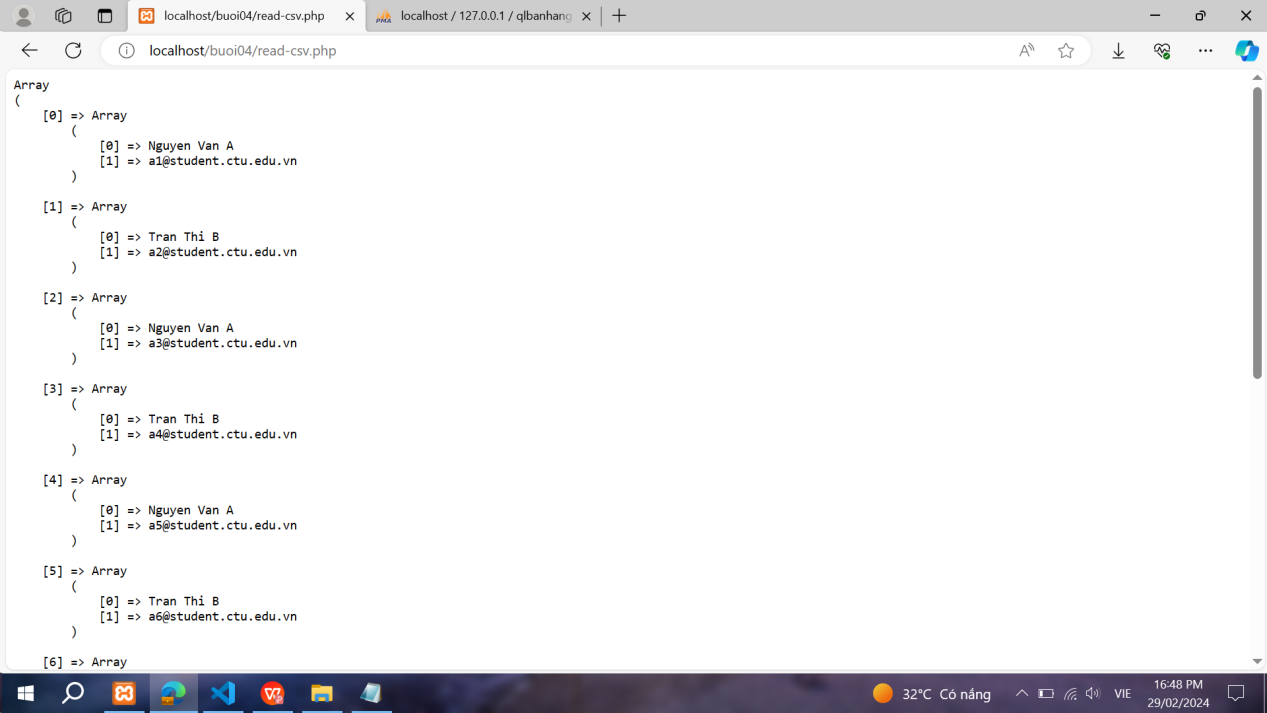
- Tạo Upload-csdl.php:



- Homepage.php (modified với ảnh profile):



- read-csv.php:

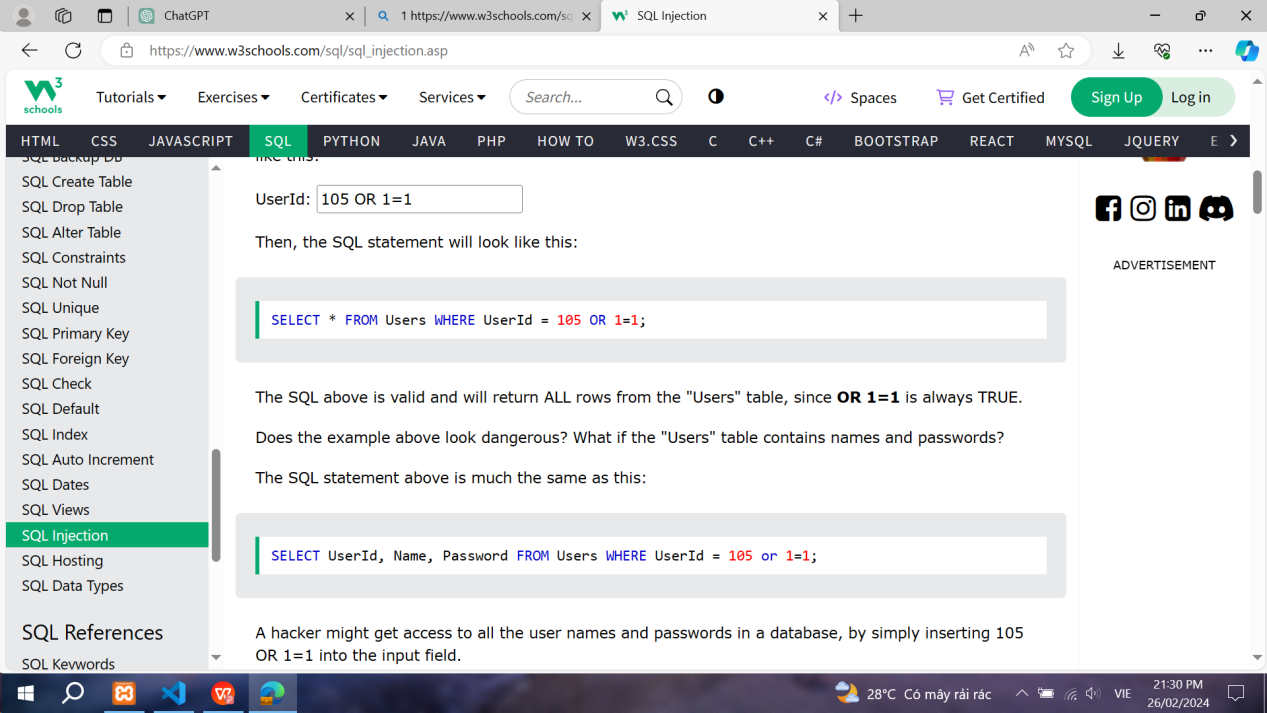


**Câu 2:**

- SQL Injection là một kỹ thuật tấn công mà hacker sử dụng để chèn các câu lệnh SQL bất hợp pháp vào các truy vấn được thiết kế để tương tác với cơ sở dữ liệu của ứng dụng. Khi thành công, tấn công SQL Injection có thể gây ra nhiều vấn đề bảo mật nghiêm trọng, bao gồm lợi dụng thông tin người dùng, thực hiện thay đổi không phù hợp trong cơ sở dữ liệu, hoặc thậm chí là kiểm soát toàn bộ hệ thống.

- Cơ sở của SQL Injection là việc không kiểm tra hoặc xử lý đúng đắn dữ liệu đầu vào từ người dùng trước khi thực hiện các truy vấn SQL. Thông thường, các ứng dụng web chấp nhận dữ liệu từ người dùng thông qua các biểu mẫu trực tuyến, thẻ cookie, thậm chí là các tham số trên URL. Khi dữ liệu này không được kiểm tra cẩn thận, hacker có thể chèn các đoạn mã SQL độc hại vào các truy vấn, điều này dẫn đến thực thi các hành động không mong muốn.

- Theo ví dụ ở dưới đây thì hacker đã sử dụng đoạn mã SQL có độc “UserId = 105 or 1=1”, có thêm “105” là mã độc được hacker thêm vào.



**Câu 3:**

- Cookie là một cách để lưu trữ thông tin trên máy tính của người dùng. Thông thường, cookie được sử dụng để lưu trữ thông tin về người dùng hoặc trạng thái của trang web. Cookie được gửi từ máy chủ và được lưu trữ trên máy tính của người dùng, sau đó được gửi lại đến máy chủ mỗi lần trình duyệt yêu cầu một trang từ máy chủ.

- “setcookie(name, value, expire, path, domain, secure, httponly); ” có:

+ name: Tên của cookie.

+ value: Giá trị của cookie.

+ expire: Thời điểm cookie sẽ hết hạn, được biểu diễn dưới dạng timestamp. Nếu không được đặt, cookie sẽ hết hạn khi trình duyệt đóng.

+ path: Đường dẫn trên máy tính của người dùng mà cookie có sẵn. Mặc định là '/' nghĩa là có sẵn cho tất cả các đường dẫn trong trang web.

+ domain: Tên miền mà cookie có sẵn. Nếu được đặt, cookie chỉ có sẵn cho tên miền đó và các subdomain của nó.

+ secure: Nếu đặt là true, cookie chỉ được gửi nếu kết nối là an toàn (HTTPS).

+ httponly: Nếu đặt là true, cookie chỉ có thể được truy cập thông qua HTTP, không thể được truy cập thông qua JavaScript.

- Lưu giá trị vào cookie:

setcookie("user", "Thanh Ngan", time() + 3600, "/");

+ time() + 3600: đặt thời điểm hết hạn của cookie là một giờ sau thời điểm hiện tại.

- Lấy giá trị từ cookie:

$user = $\_COOKIE["user"];

echo "Welcome $user!";

- Xóa giá trị từ cookie:

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

+ time() - 3600: đặt thời điểm hết hạn trong quá khứ, khiến cho cookie không còn hiệu lực.

+ / : cho biết cookie sẽ có sẵn cho tất cả các đường dẫn.

**Câu 4:**

- Session trong PHP là một cách để lưu trữ thông tin phiên làm việc (session data) trên máy chủ. Session giữ thông tin giữa các trang, giúp lưu trữ dữ liệu liên quan đến người dùng trong suốt thời gian họ duyệt trang web.

- Lưu giá trị vào session:

session\_start();

$\_SESSION["user"] = "John Doe";

- Lấy giá trị từ session:

session\_start();

$user = $\_SESSION["user"];

- Xóa giá trị của session:

session\_start();

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

- Xóa toàn bộ session:

session\_start();

session\_destroy(); -> xóa toàn bộ có thể dẫn đến người dùng đăng xuất.

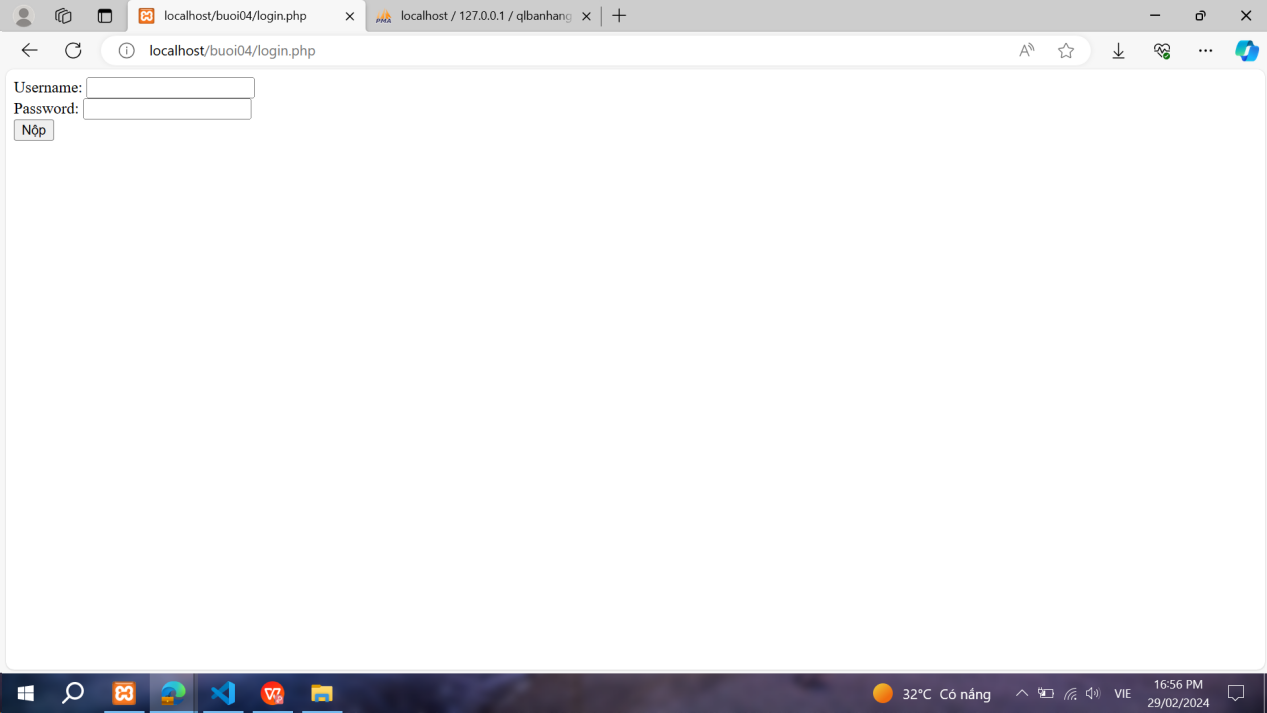
**Câu 5:**

|  |  |
| --- | --- |
| **Cookie** | **Session** |
| Lưu trữ thông tin trên máy tính người dùng, thường được sử dụng để theo dõi trạng thái và giữ lại thông tin giữa các lần truy cập. | Lưu trữ thông tin trên máy chủ, giữ lại thông tin giữa các trang của một phiên làm việc. |
| Có thể bị mạo danh nếu bị đánh cắp, vì thông tin được lưu trữ trên máy tính người dùng. | Thường được coi là an toàn hơn vì thông tin lưu trữ trên máy chủ và không thể dễ dàng truy cập từ phía người dùng. |
| Có thể có thời gian sống cố định, tức là nó sẽ tồn tại cho đến khi hết thời gian hoặc khi người dùng xóa nó. | Thường tồn tại trong suốt phiên làm việc của người dùng và bị xóa khi phiên làm việc kết thúc. |
| Lưu trữ thông tin trên máy tính người dùng và được gửi qua mỗi yêu cầu HTTP. | Lưu trữ thông tin trên máy chủ và chỉ gửi một ID phiên (session ID) tới máy tính người dùng. |
| Có thể được truy cập từ cả JavaScript và server-side scripts. | Thường được truy cập từ server-side scripts. |
| Cần lưu trữ thông tin ở phía máy tính người dùng, có thể tăng tải cho máy tính người dùng nếu có quá nhiều cookie. | Thông tin lưu trữ trên máy chủ, giảm gánh nặng cho máy tính người dùng. |
| Thường được sử dụng cho các dữ liệu nhỏ, như trạng thái đăng nhập, cài đặt ngôn ngữ. | Thường được sử dụng cho các thông tin lớn hoặc những thông tin đặc biệt như thông tin đăng nhập chi tiết, giỏ hàng. |

- Tài liệu tham khảo: ChatGPT.

**Câu 6:**

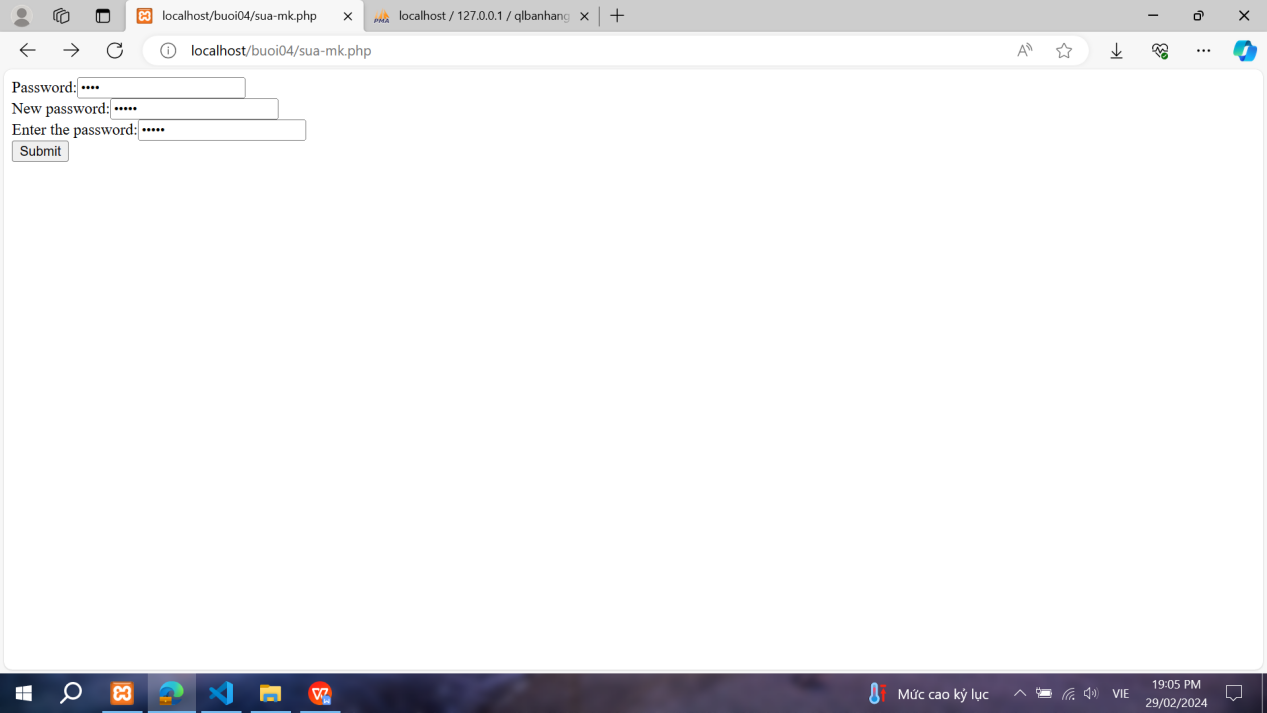
**-** Thay đổi trong tệp log.php:



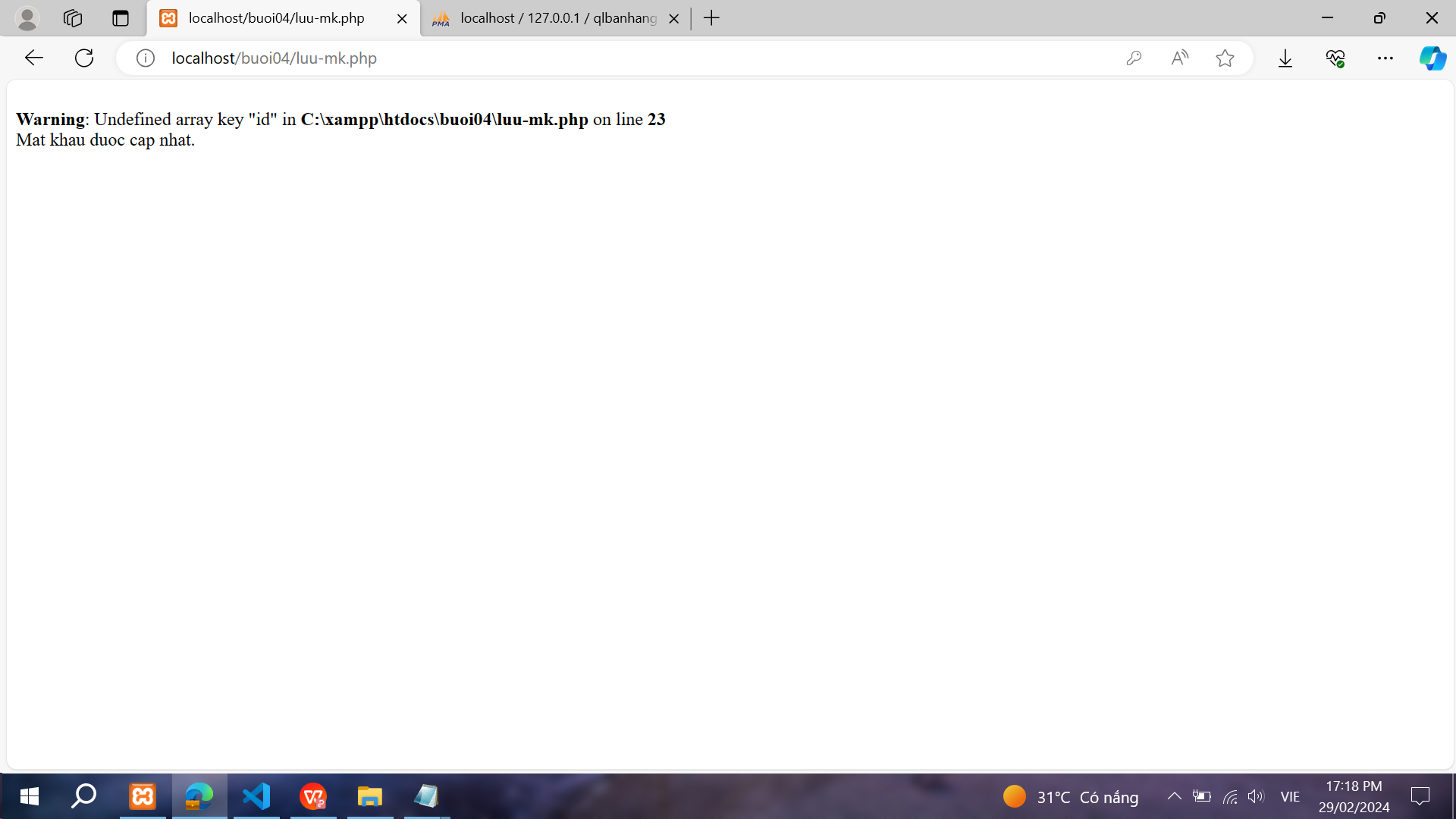
- Đã thay đổi tệp homepage.php và tạo tệp thoat.php(phần code).

**Câu 7:**

**-** Sửa mk:



-Tệp luu-mk.php:



**Câu 8:**

- Tính năng upload file trong PHP thường được thực hiện bằng cách sử dụng biểu mẫu HTML kết hợp với các hàm và thủ tục PHP.

- Biểu mẫu HTML:

+ Sử dụng thẻ <form> để tạo biểu mẫu upload file.

+ Sử dụng thuộc tính enctype="multipart/form-data" để báo hiệu rằng biểu mẫu này chứa dữ liệu file.

- PHP Upload File:

+ Sử dụng biến siêu toàn cục $\_FILES để lấy thông tin về file được upload.

+ Sử dụng hàm move\_uploaded\_file() để di chuyển file từ thư mục tạm thời đến thư mục lưu trữ.

+ $target\_dir là thư mục lưu trữ file đã upload.

+ $target\_file là đường dẫn đầy đủ đến file được upload.

+ $uploadOk được sử dụng để kiểm tra xem việc upload có thành công hay không.

- Để sử dụng chức năng upload file, cần đảm bảo rằng thư mục lưu trữ (uploads/ trong trường hợp này) có quyền ghi.

- Luôn kiểm tra các điều kiện bảo mật như kiểm tra kiểu file, kích thước file, và ngăn chặn việc upload các file độc hại.

**Câu 9:**

- Hàm file() trong PHP được sử dụng để đọc nội dung của một file và trả về nó dưới dạng một mảng. Mỗi phần tử trong mảng tương ứng với một dòng trong file.

- Cú pháp:

file(filename, flags , context )

- Trong đó:

+ filename: Tên của file cần đọc.

+ flags: Các cờ tùy chọn, có thể là một hoặc kết hợp của các hằng số như FILE\_USE\_INCLUDE\_PATH, FILE\_IGNORE\_NEW\_LINES, FILE\_SKIP\_EMPTY\_LINES.

+ context: Một nguồn tài nguyên có thể được sử dụng để thiết lập context stream.

**Câu 10:**