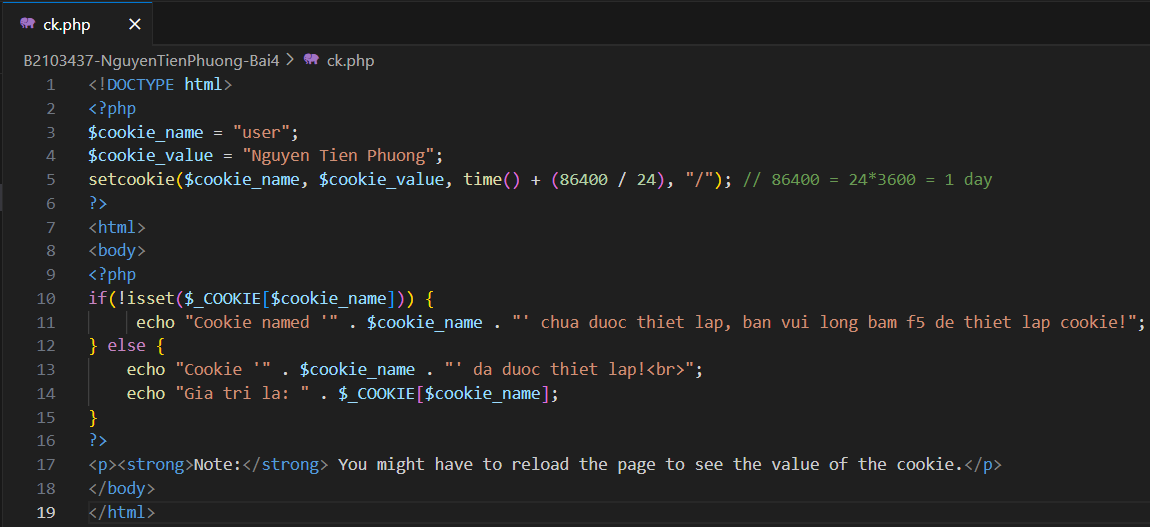
**BÀI THỰC HÀNH 4**

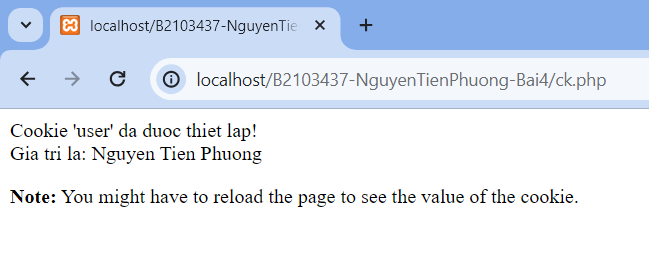
**YÊU CẦU BÀI THỰC HÀNH**

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

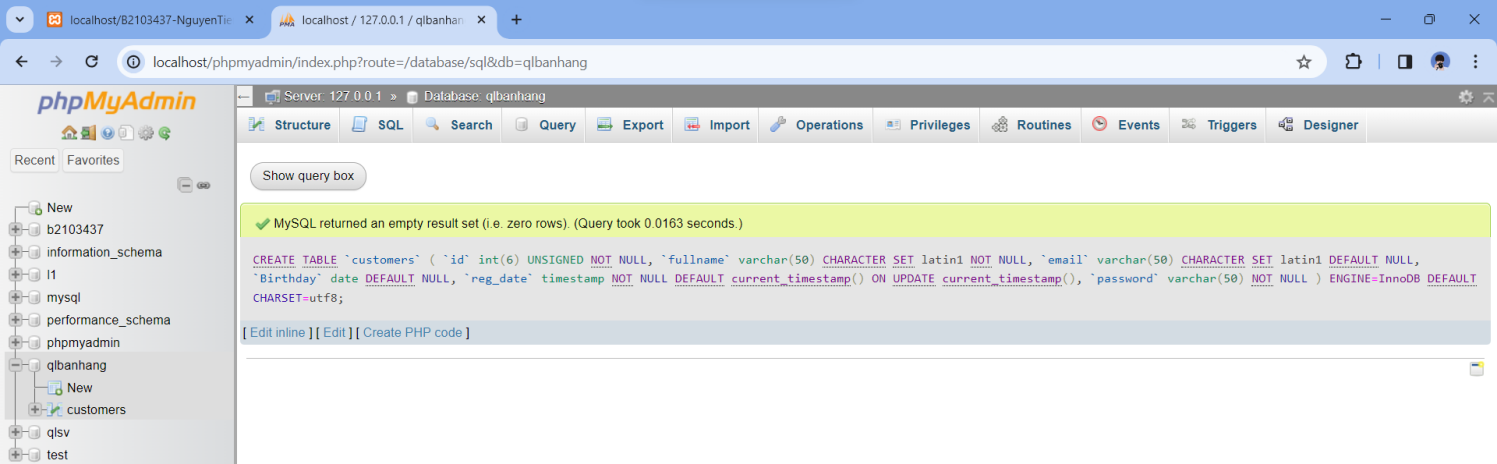
Tạo Cookies (Tập tin ck.php)



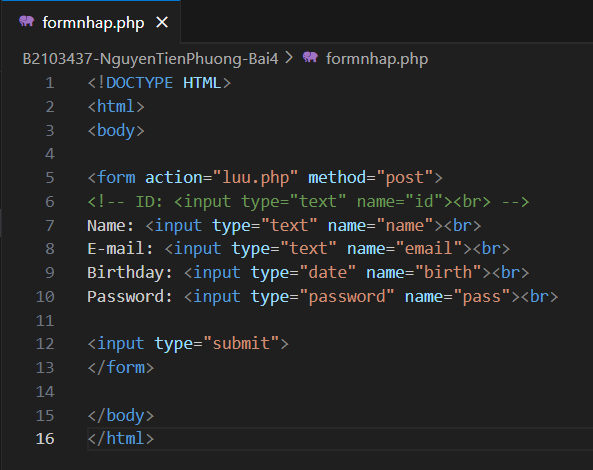
Kết quả:



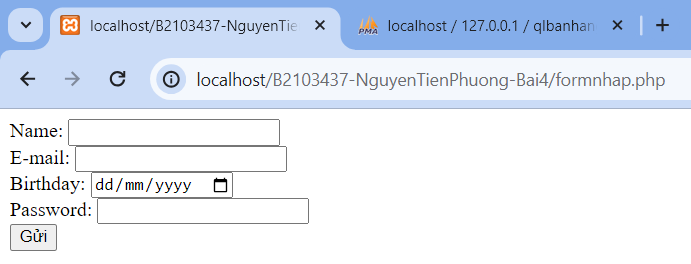
Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa

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

Tạo formnhap.php

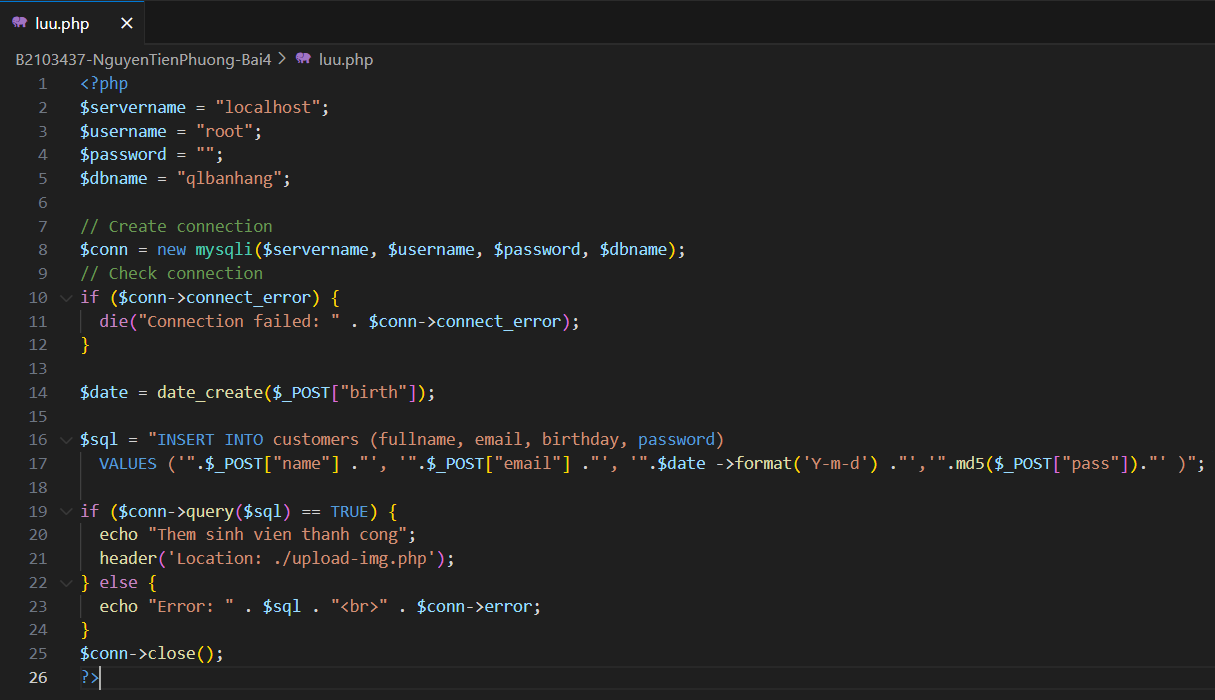


Kết quả:

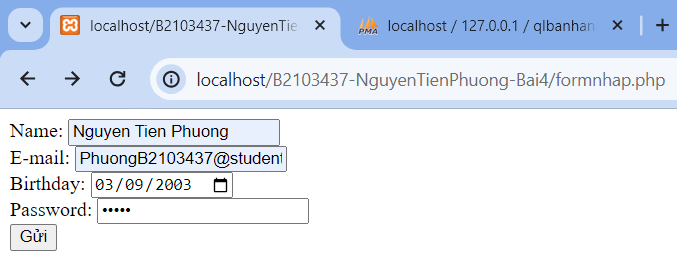


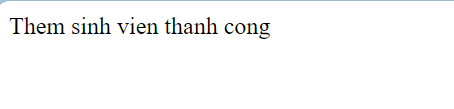
Và luu.php, chú ý lúc này chúng ta sẽ mã hóa mật khẩu với giải thuật md5 để bảo mật mật khẩu:

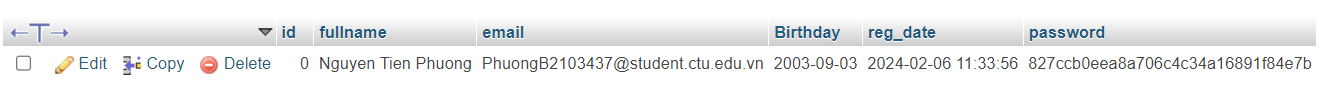
luu.php



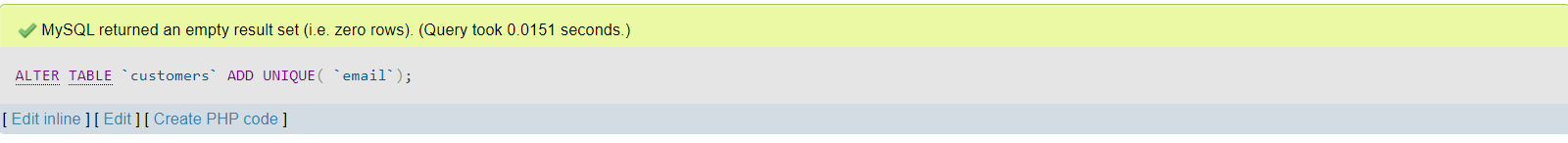
Sau khi chúng ta nhập xong mật khẩu với 12345 nhưng khi quan sát trong CSDL ta sẽ thấy mật khẩu đã được mã hóa ra chuỗi 827ccb0eea8a706c4c34a16891f84e7b:





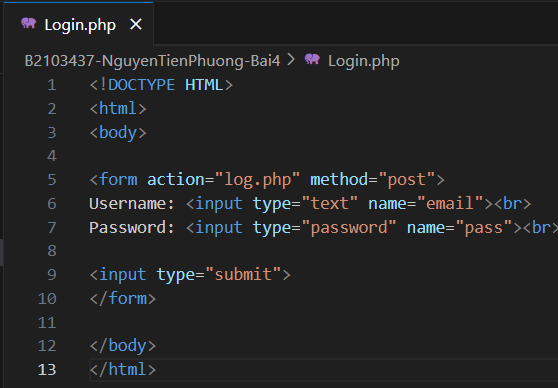


Trong trường hợp chúng ta muốn dùng email là username để đăng nhập ta nên đảm bảo 2 student không có email trùng nhau. Để đảm bảo điều này ta có thể chọn email là unique key (vì ta đã chọn id la primary key, và 1 bảng không thể có 2 khóa chính)

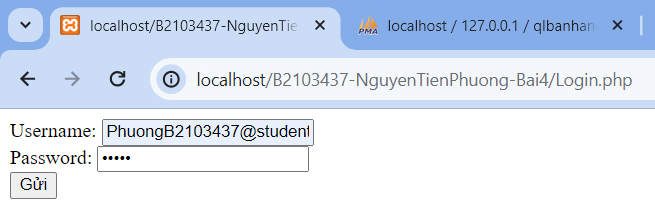


## Tạo form đăng nhập

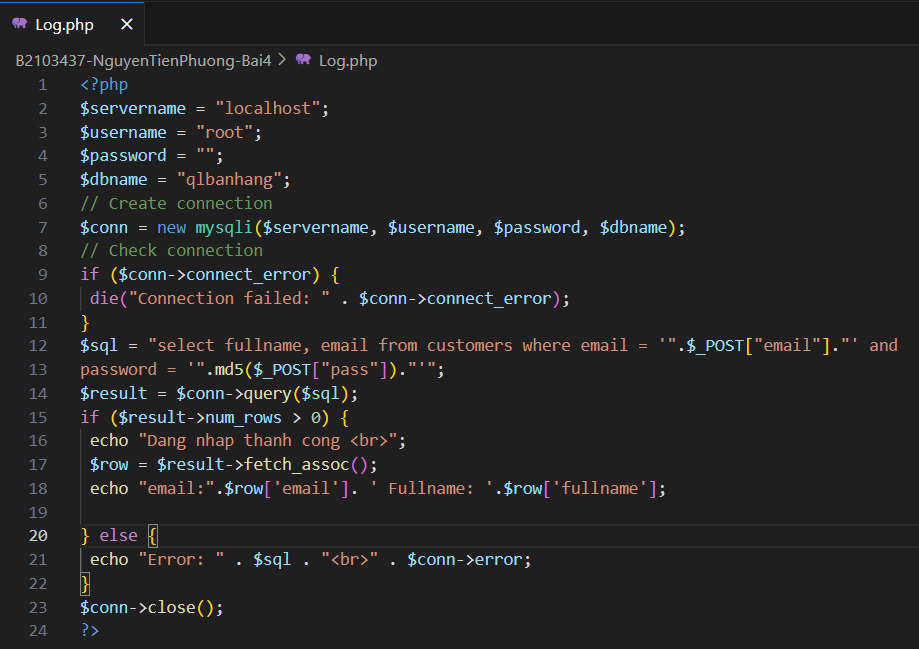
Sau khi tạo người dùng xong, ta có thể chạy form đăng nhập với username là email và password lấy từ CSDL



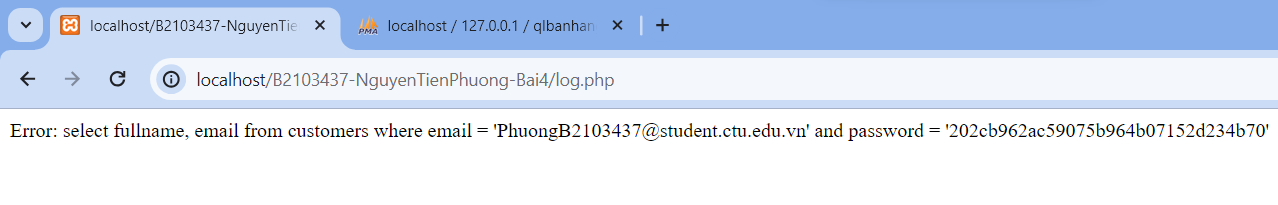
Kết quả:



Log.php

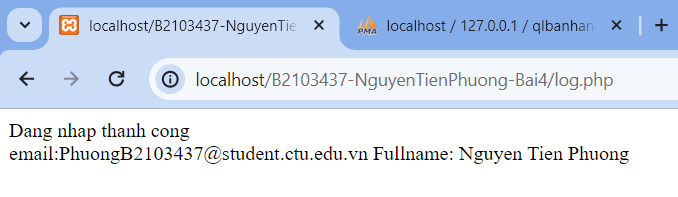


Khi chúng ta nhập sai, nhìn vào thông báo lỗi ta thấy



Mật khẩu đã cho không khớp với mã đã băm

Khi ta nhập đúng



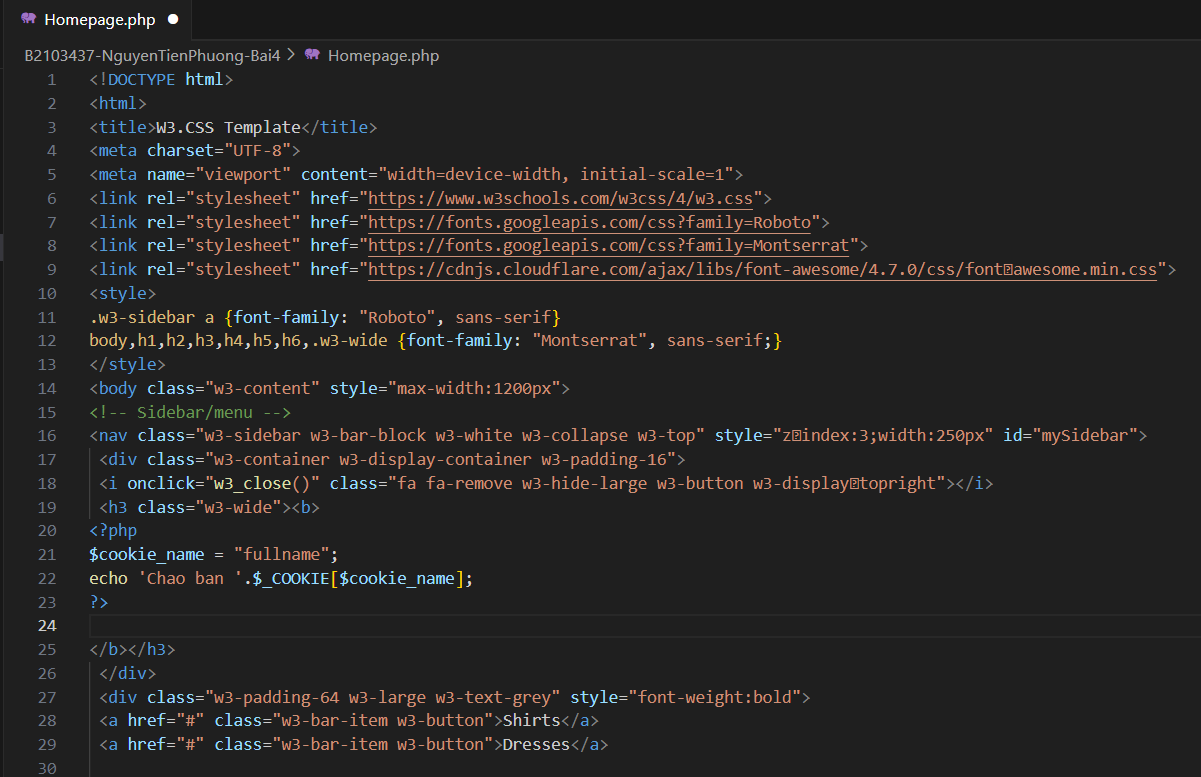
## Dùng Cookies lưu lại tên đăng nhập, email và điều hướng

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

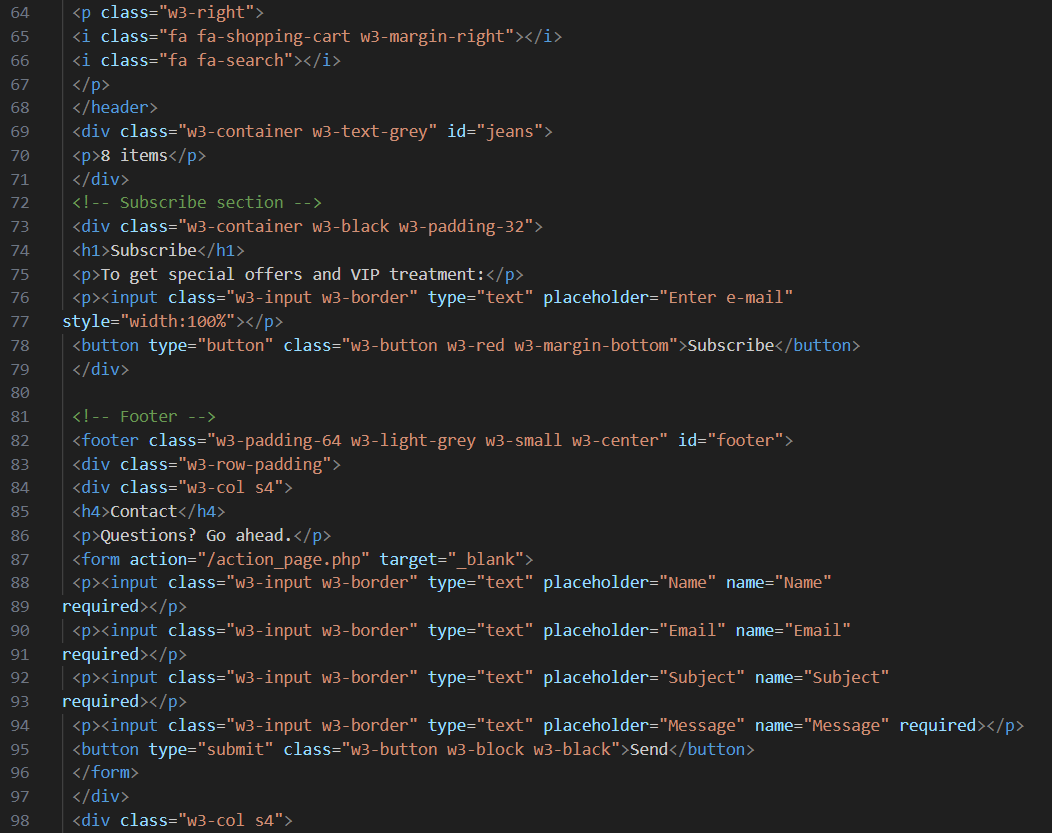


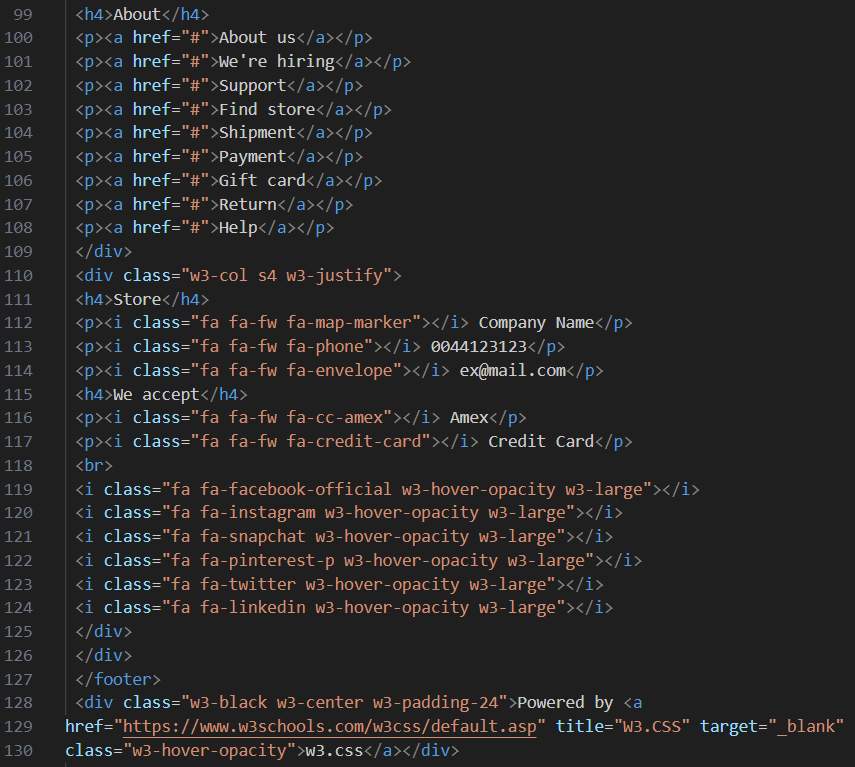
Tạo 1 trang chủ để khi đăng nhập thành công chuyển qua trang chủ này:

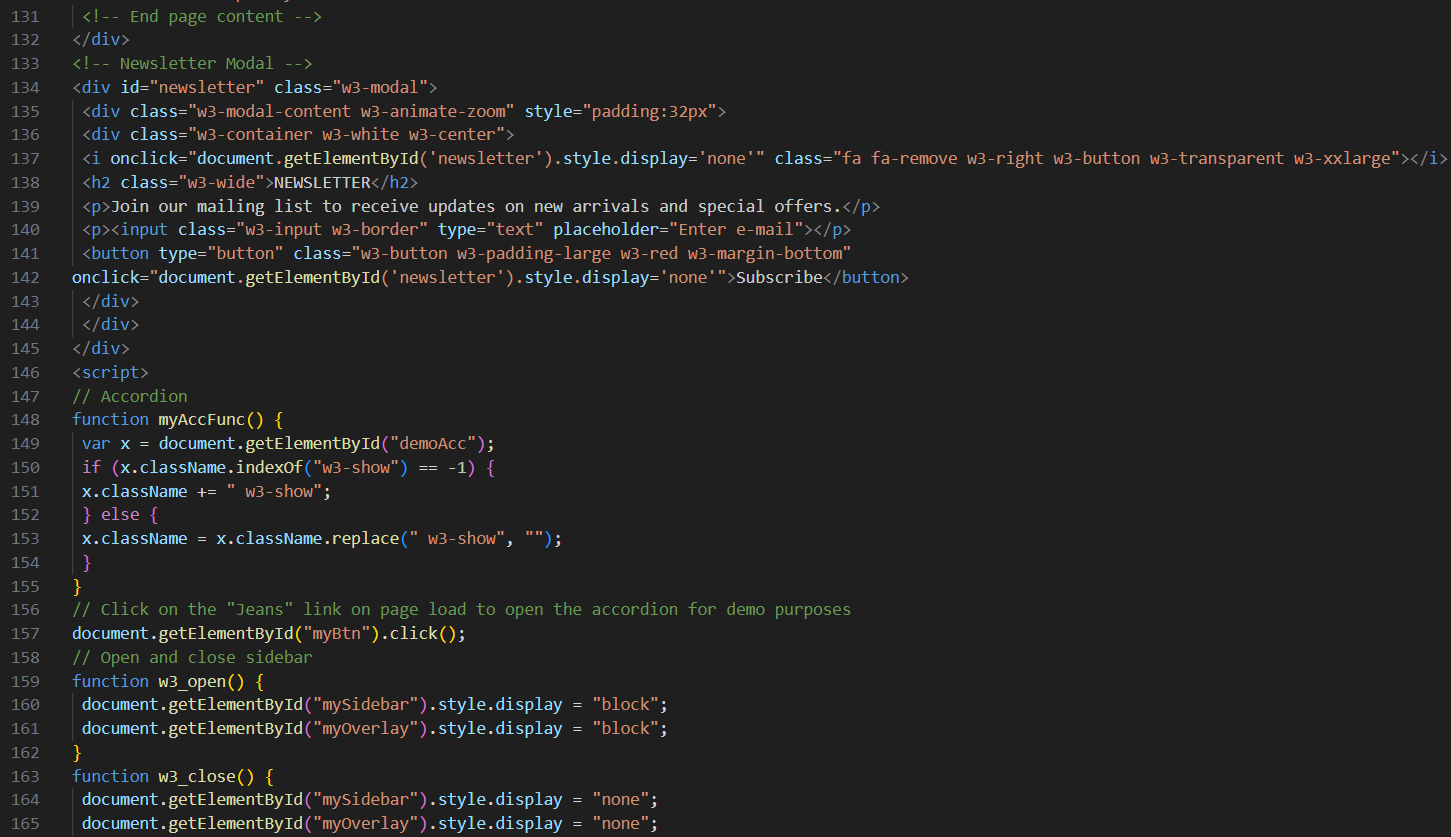
Homepage.php

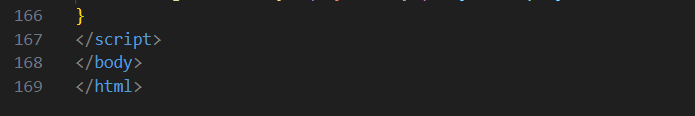




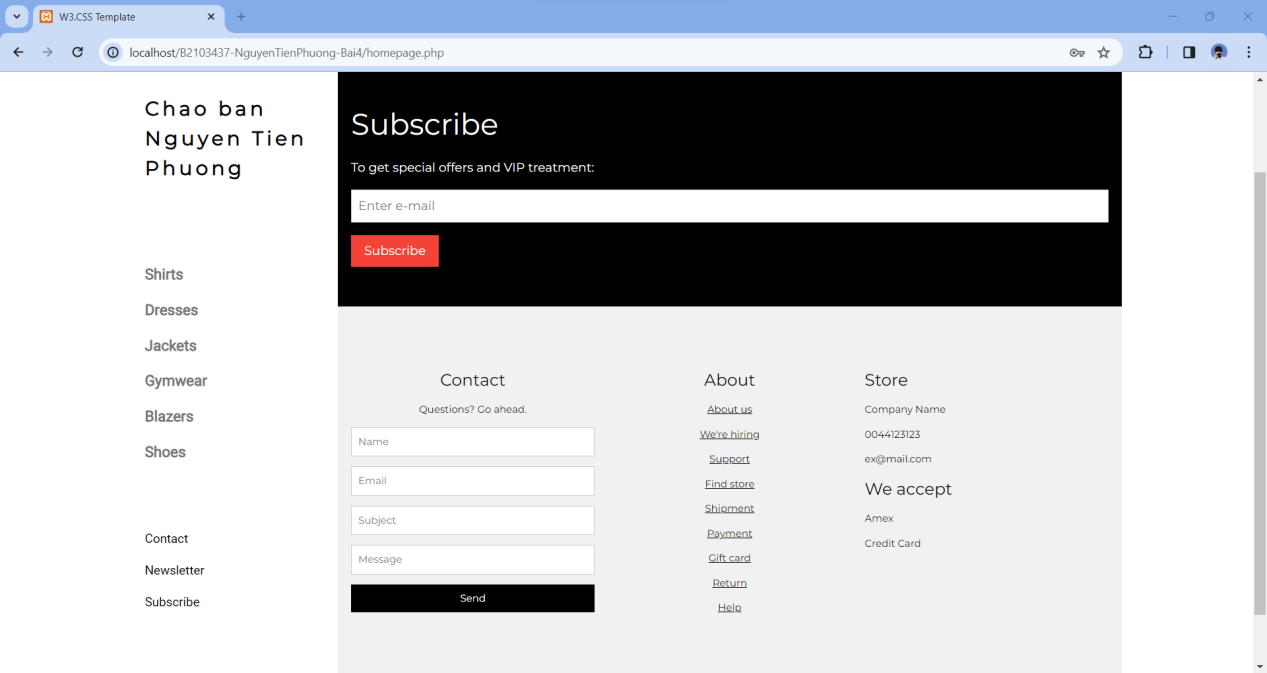








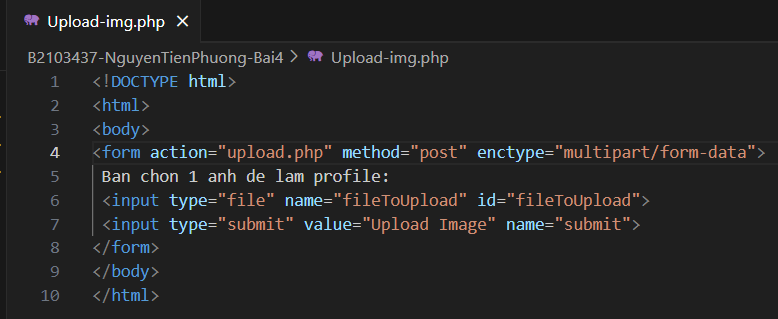
Kết quả:



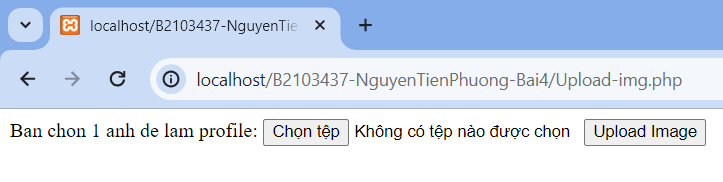
## Upload tập tin

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

### Upload-img.php

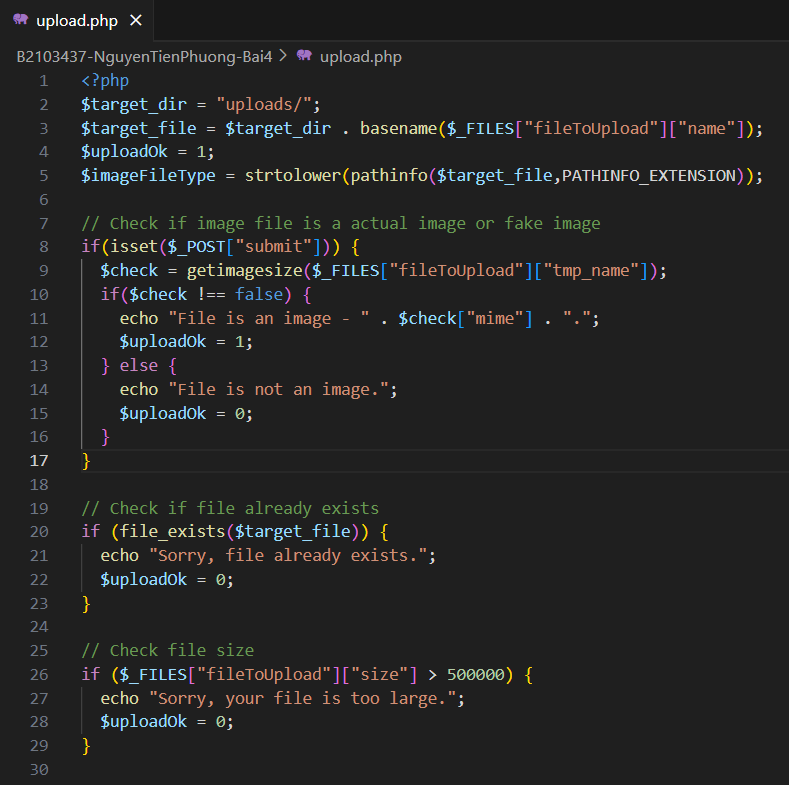


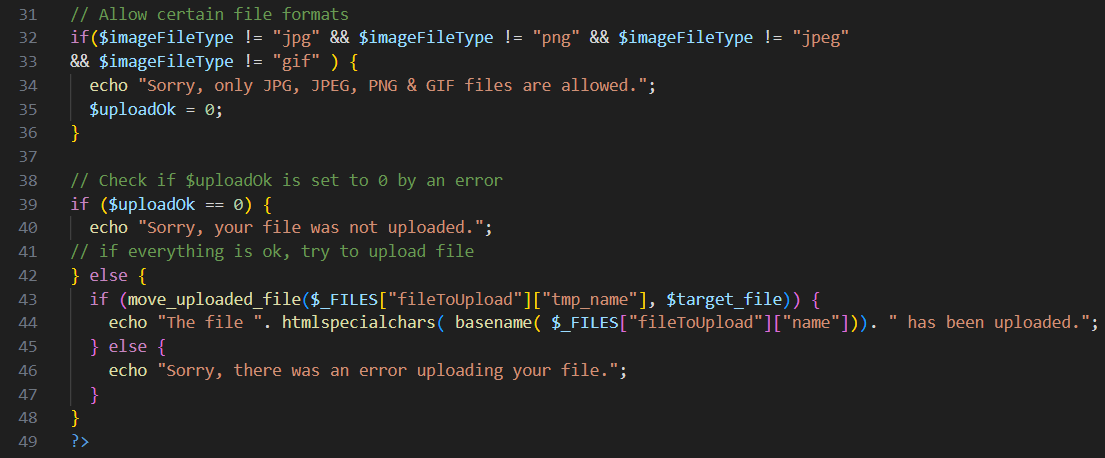
Kết quả:



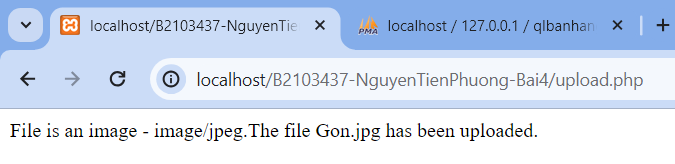
Tập tin upload.php xử lý hành động

upload.php





Kết quả: Sau khi chúng ta upload thành công giao diện sau sẽ hiển thị

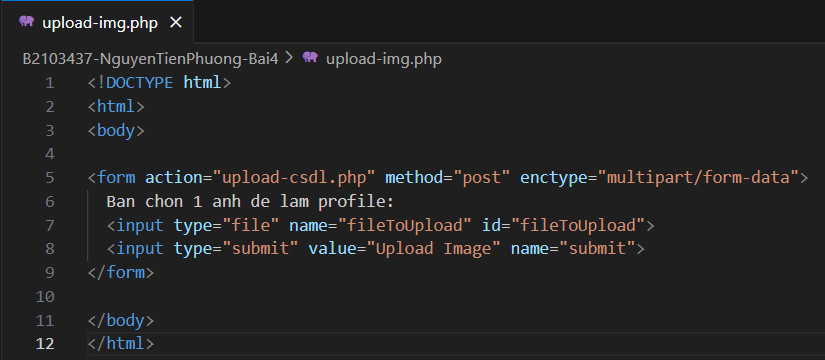


Upload ảnh và lưu dữ liệu về ảnh trong CSDL

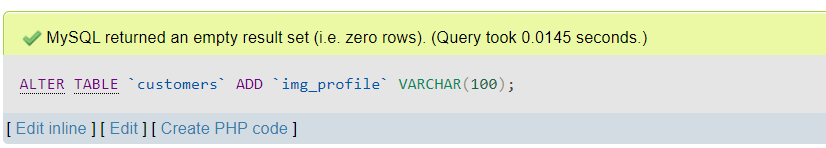
Trong phần này chúng ta minh họa tải tập tin ảnh làm ảnh profile

của người dùng

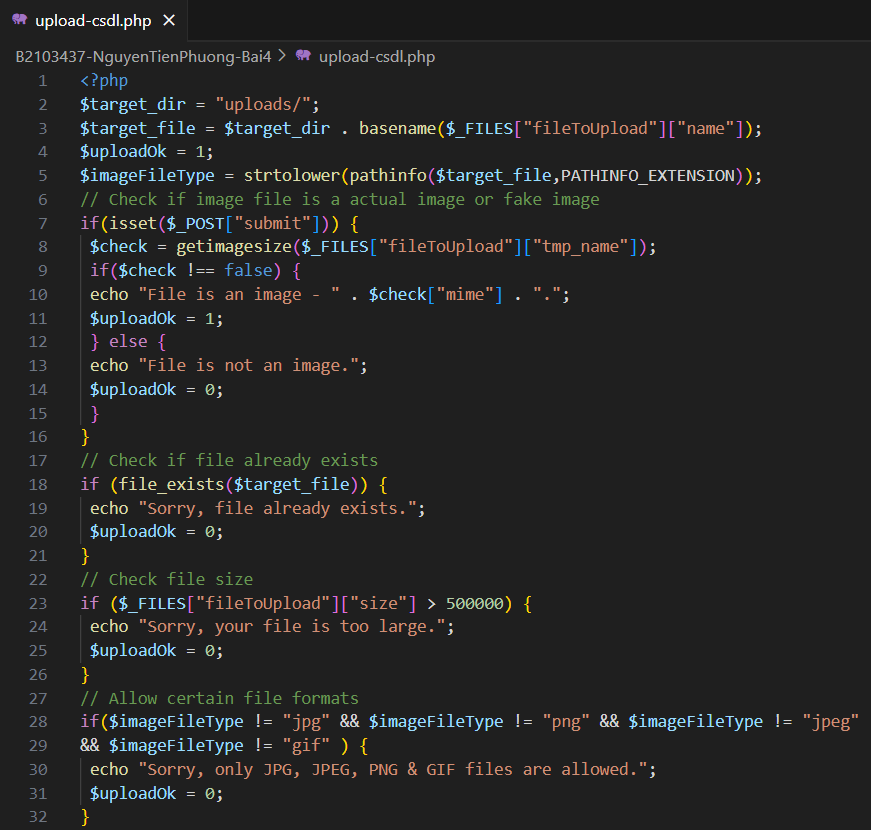
Upload-img.php (modified)

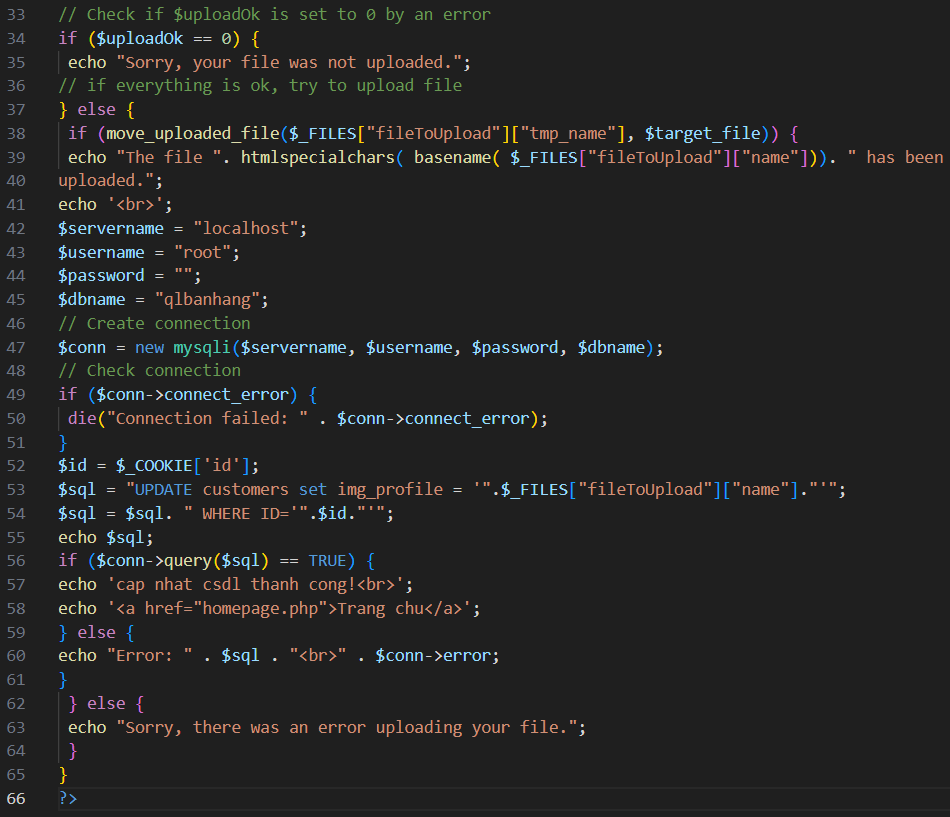


Thêm cột để lưu đường dẫn ảnh:

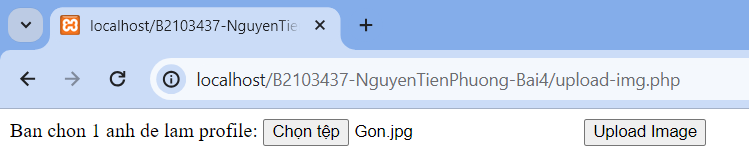


Upload-csdl.php



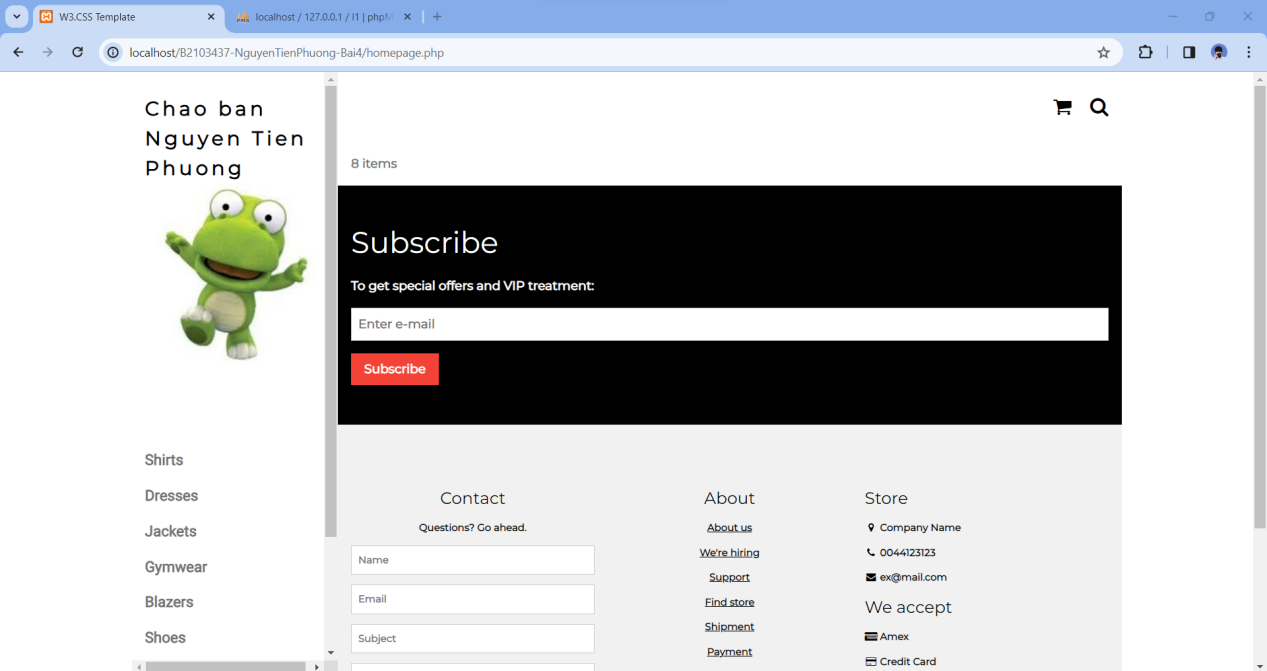


Kết quả:



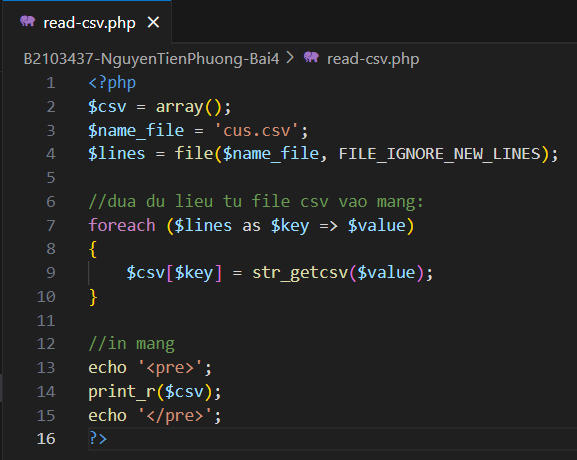


Homepage.php (modified với ảnh profile)

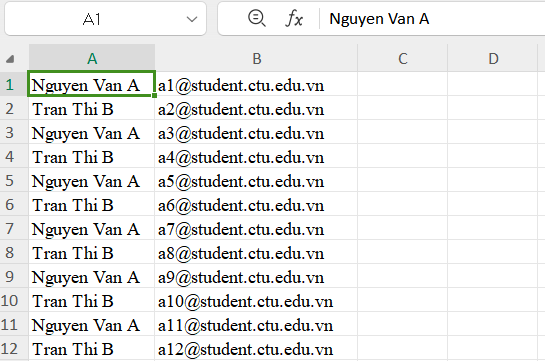


Upload file csv và đọc dữ liệu vào mảng

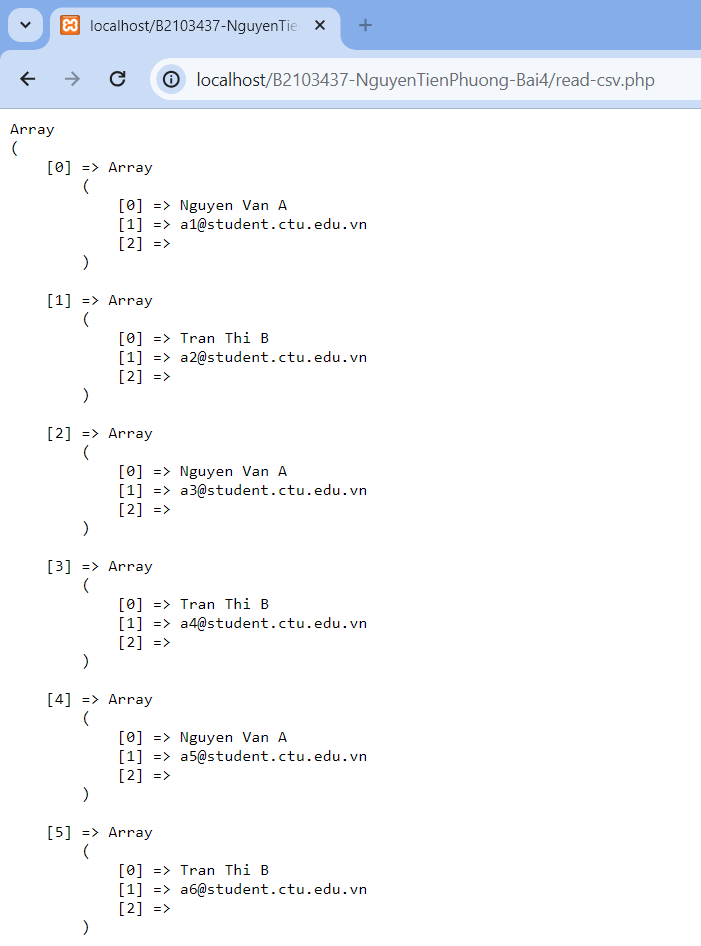
read-csv.php

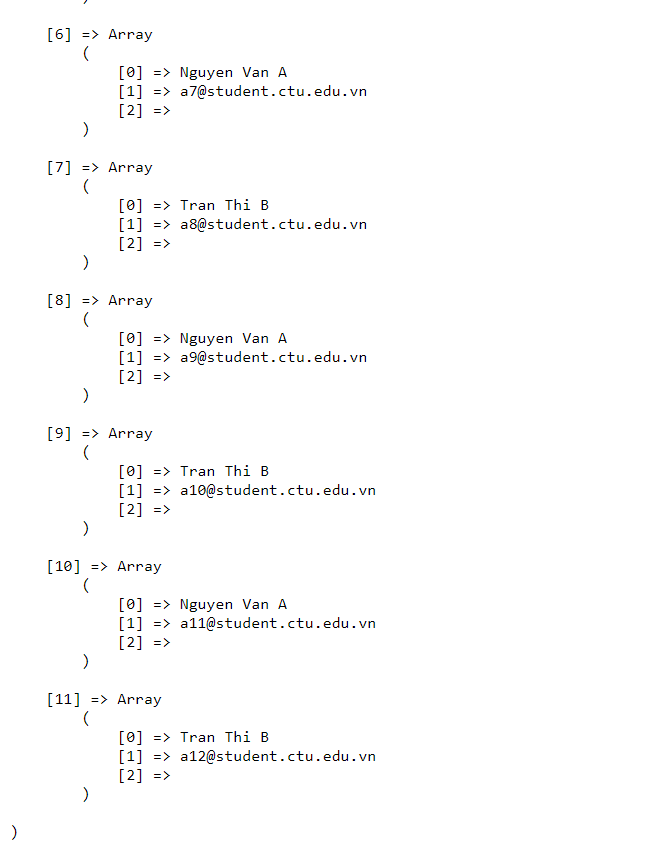


Tạo tập tin cus.csv với dữ liệu như sau:



Khi chạy code đọc trên ta sẽ có dữ liệu đã được đưa vào mảng và hiển thị





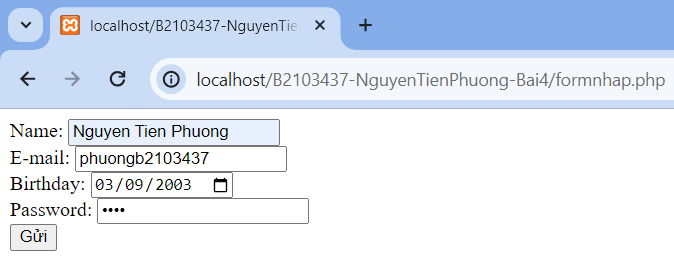
1. **Trong vài trường hợp, hacker có thể sử dụng các kỹ thuật SQL Injection để hack hệ thống của bạn. Bạn hãy trình bày SQL Injection là gì và thử các ví dụ trình bày trong1 . Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.**

- SQL injection là một kỹ thuật chèn mã có thể phá hủy cơ sở dữ liệu của bạn.

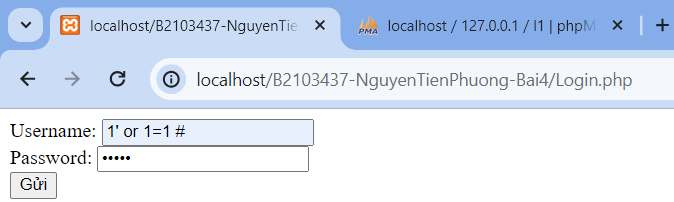
- SQL injection là một kỹ thuật chèn mã có thể phá hủy cơ sở dữ liệu của bạn.

- SQL injection là việc đặt mã độc trong các câu lệnh SQL, thông qua đầu vào trang web.

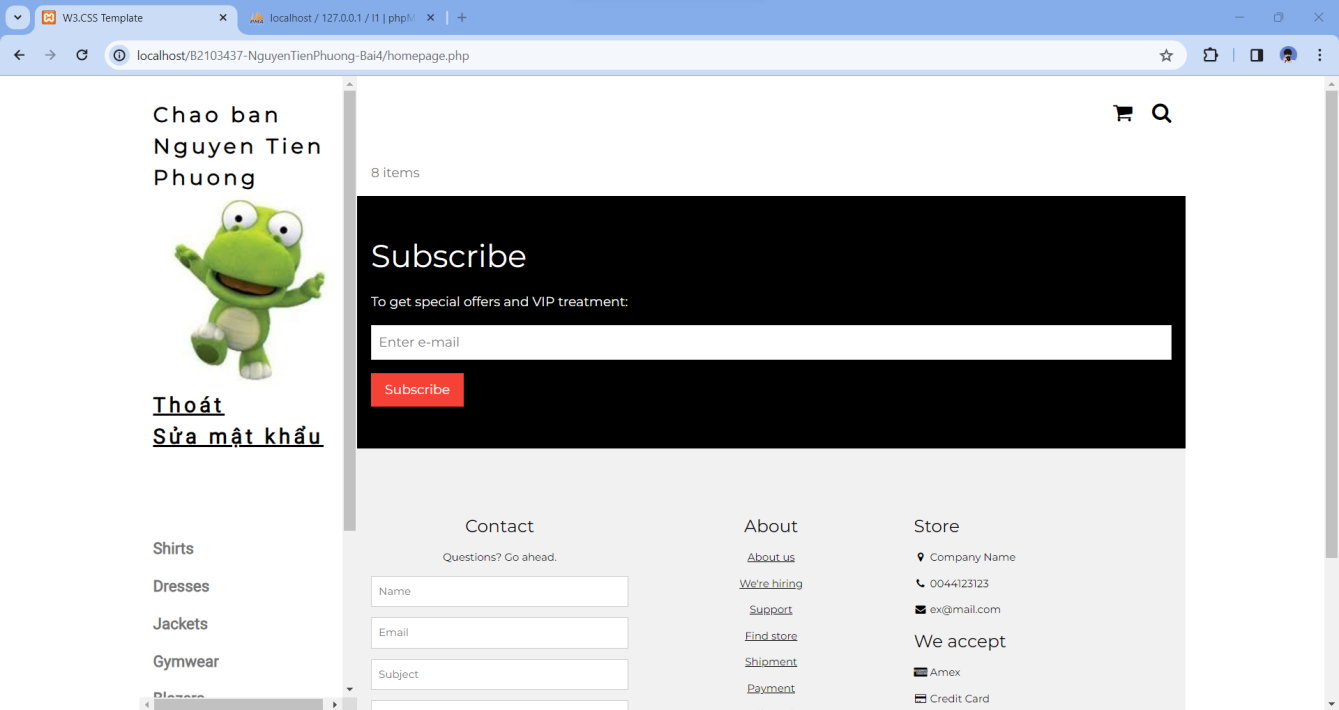
Ứng dụng vào trang đăng nhập (với email = ‘phuongb2103437’ , pass =’1234’)



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 #” không cần username, password



**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 thường được sử dụng để xác định người dùng. Cookie là một tệp nhỏ mà Máy chủ nhúng trên máy tính của người dùng. Mỗi lần cùng một máy tính yêu cầu một trang có trình duyệt, nó cũng sẽ gửi cookie. Với PHP, cả hai bạn có thể tạo và truy xuất các giá trị cookie.

- Hàm này có sáu tham số, là:

+ Name: Tên của cookie.

+ Value: Giá trị của cookie.

+ Options: Một mảng chứa các tùy chọn khác nhau cho cookie, như expires, path, domain, secure, httponly, samesite.

+ Expires: Thời gian hết hạn của cookie, tính bằng giây kể từ thời điểm hiện tại.

+ Path: Đường dẫn mà cookie sẽ được gửi cùng với yêu cầu HTTP.

+ Domain: Tên miền mà cookie sẽ được gửi cùng với yêu cầu HTTP.

- Đây là một ví dụ sử dụng hàm setcookie() để lưu một cookie được đặt tên username và gán giá trị giá trị Tien Phuong cho nó. Nó cũng chỉ định rằng cookie sẽ hết hạn sau 30 ngày ( 30 days \* 24 hours \* 60 min \* 60 sec).

**Ví dụ:**

**Lưu cookie**

<?php

setcookie("username", "Tien Phuong", time()+30\*24\*60\*60);

?>

**Lấy cookie**

<?php

echo $\_COOKIE["username"];

?>

**Xóa cookie**

<?php

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

?>

Nguồn : https://vzn.vn/lap-trinh-php/cookies-trong-php/

**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 cách để lưu trữ thông tin của người dùng trên nhiều trang web. Mỗi phiên làm việc của người dùng được gắn kết với một session duy nhất trên máy chủ. Để lưu trữ giá trị trong một session, bạn có thể sử dụng biến toàn cục $\_SESSION. Để bắt đầu một session, bạn phải sử dụng hàm session\_start() ở đầu trang web hoặc tập tin PHP. Bạn có thể truy cập giá trị trong Session bằng cách sử dụng cú pháp $ \_SESSION ['tên\_biến'], trong đó 'tên\_biến' là tên của biến bạn muốn lấy giá trị. Để xóa toàn bộ session và hủy phiên làm việc hiện tại, bạn có thể sử dụng hàm session\_destroy().

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

- Cookie và Session là hai cách để lưu trữ thông tin của người dùng trên web. Một số điểm khác biệt chính là:

+ Cookie là tệp nhỏ được lưu trữ trên máy khách, còn Session là biến toàn cục được lưu trữ trên máy chủ.

+ Session phụ thuộc vào Cookie, nhưng Cookie không phụ thuộc vào Session.

+ Session kết thúc khi người dùng đóng trình duyệt hoặc đăng xuất khỏi ứng dụng, còn Cookie hết hạn tùy thuộc vào thời gian bạn đặt cho nó.

+ Session có thể lưu trữ nhiều dữ liệu như người dùng muốn, còn Cookie có kích thước giới hạn là 4KB.

+ Session lưu trữ dữ liệu dưới dạng mã hóa, còn Cookie lưu trữ dữ liệu dưới dạng văn bản.

+ Session an toàn hơn Cookie.

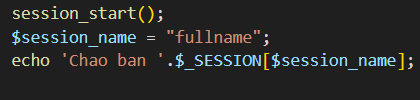
Nguồn : [Session vs Cookies: What's the Difference? - javatpoint](https://www.javatpoint.com/session-vs-cookies)

**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.**

Chỉnh sửa trong log.php



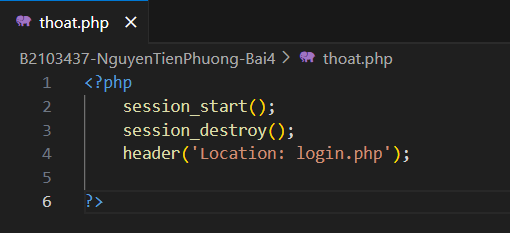
Chỉnh sửa trong homepage.php

k

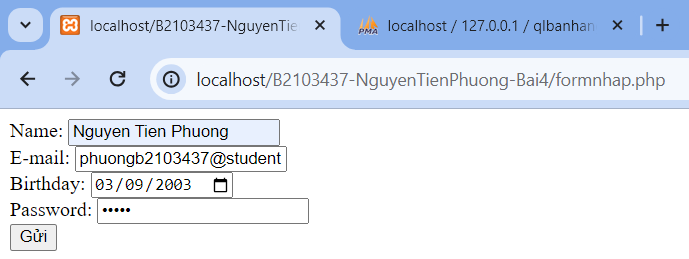


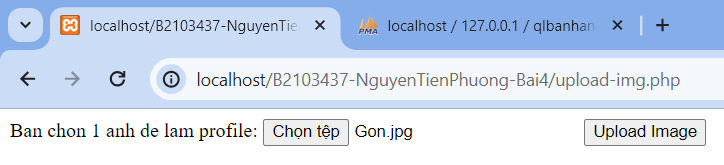


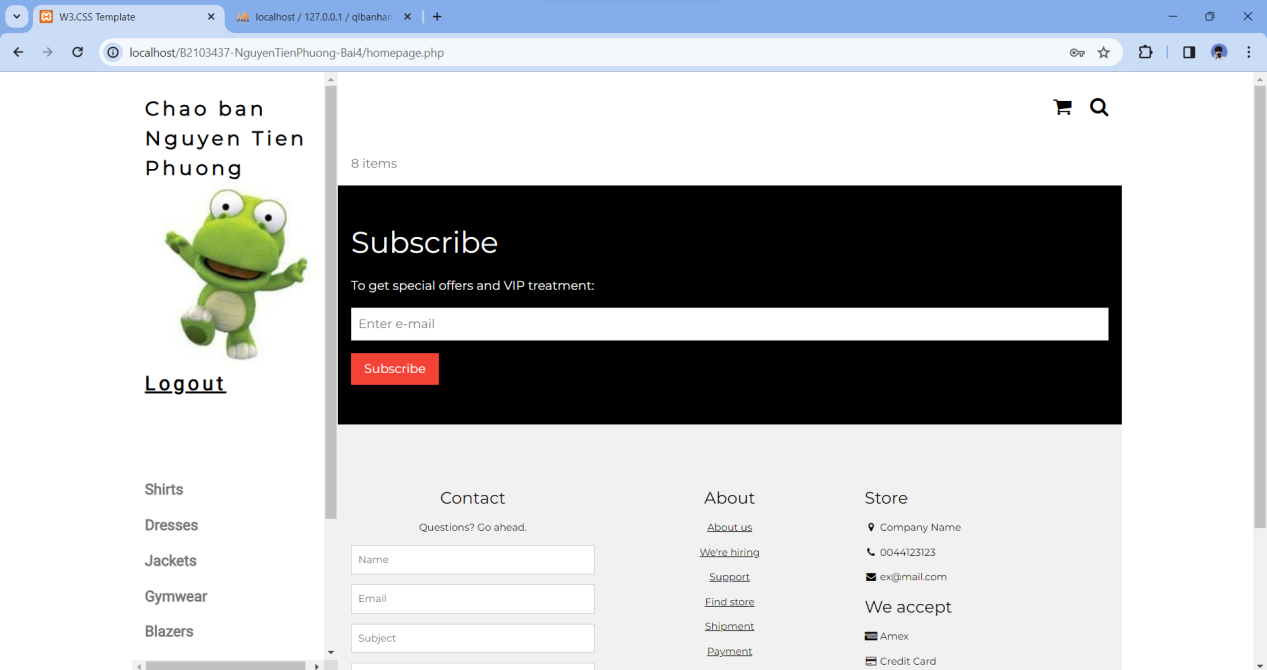
Tạo tập tin thoat.php



Chương trình sau chỉnh sửa

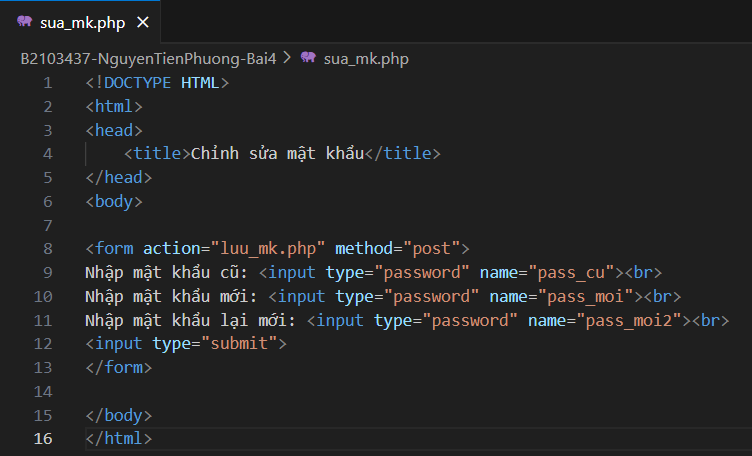






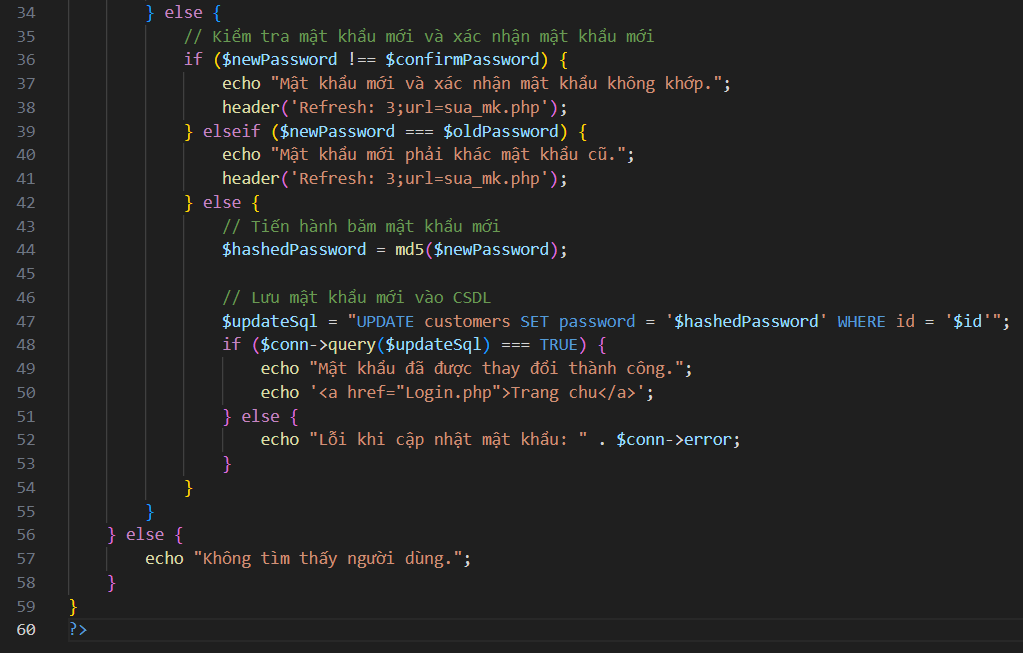
1. **Bạn hãy tạo form sua\_mk.php cho phép người dùng chỉnh sửa mật khẩu sau khi đăng nhập. Yêu cầu gồm 3 input: ô để nhập mật khẩu cũ, 1 ô để nhập mật khẩu mới, 1 ô cho phép nhập mật lại mật khẩu mới. Khi nhập xong cần đảm bảo: mật khẩu cũ là khớp với CSDL đang lưu, 2 ô nhập mật khẩu mới phải khớp với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

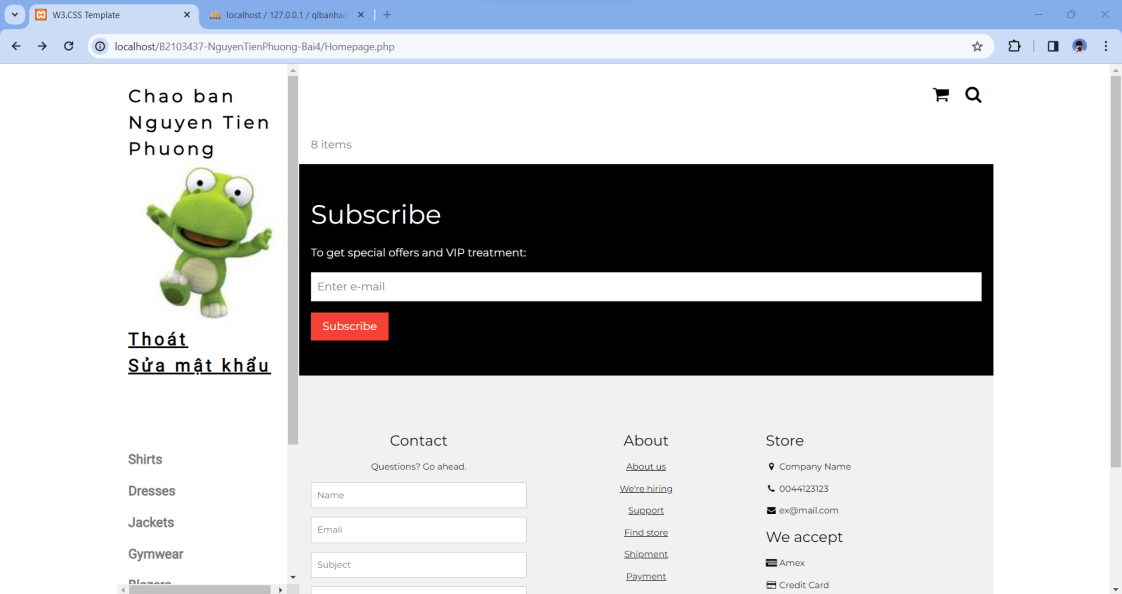
Sua\_mk.php



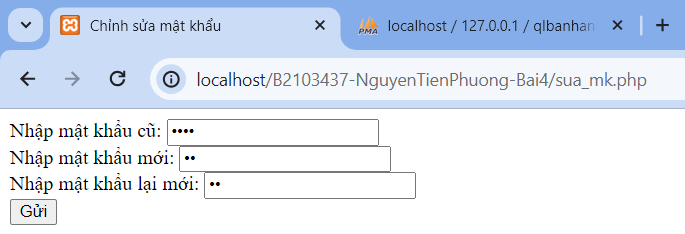
Luu\_mk.php



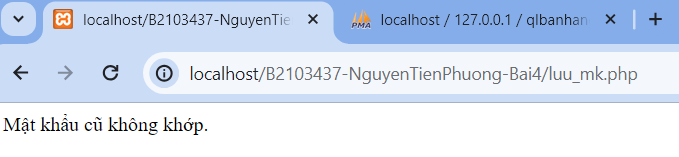


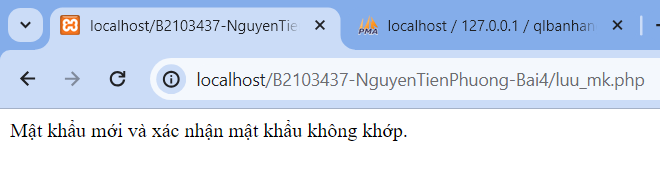


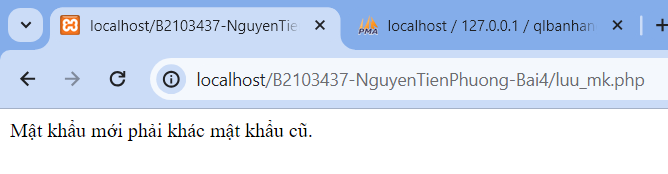
Vào sửa mật khẩu



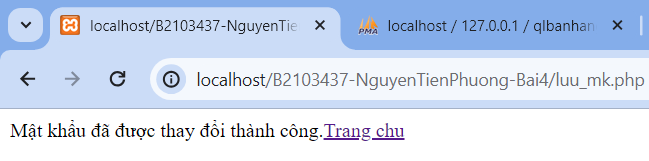
Các trường hợp lỗi







Trường hợp đổi mật khẩu thành công



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

- Dưới đây là mô tả về các cơ chế, hàm và thủ tục được sử dụng để thực hiện việc upload:

+ HTML Form: Để cho phép người dùng chọn và tải lên tệp tin, bạn cần sử dụng một biểu mẫu HTML và đảm bảo rằng thuộc tính enctype của biểu mẫu được đặt thành "multipart/form-data". Điều này cho phép dữ liệu tải lên có thể chứa các tệp tin.

+ $\_FILES Super Global: Khi người dùng gửi biểu mẫu, thông tin về tệp tin tải lên sẽ được lưu trữ trong mảng siêu toàn cục $\_FILES. Mảng này chứa các khóa và giá trị tương ứng với thông tin về tệp tin như tên, loại MIME, đường dẫn tạm thời và kích thước. move\_uploaded\_file(): Để lưu trữ tệp tin tải lên vào vị trí mong muốn trên máy chủ, bạn cần sử dụng hàm.

+ move\_uploaded\_file(). Hàm này nhận hai tham số: đường dẫn tạm thời của tệp tin và đường dẫn đích mà bạn muốn lưu trữ tệp tin đó.

+ Kiểm tra lỗi: Trong quá trình upload, có thể xảy ra các lỗi như kích thước tệp tin vượt quá giới hạn cho phép hoặc tệp tin không hợp lệ. Để xử lý các lỗi này, bạn có thể sử dụng các điều kiện và thông báo lỗi tương ứng.

**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():

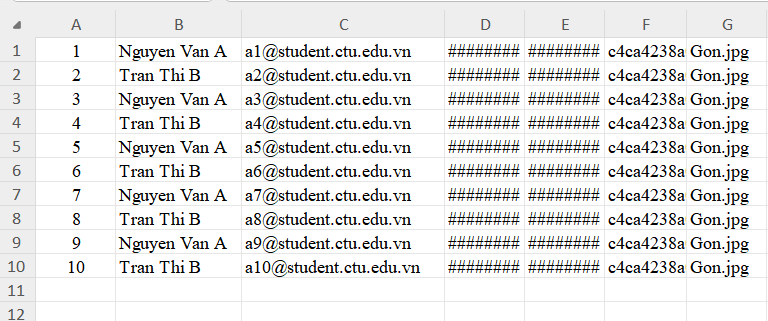
+ $filename (bắt buộc): Đường dẫn của tệp tin cần đọc. Đây có thể là một đường dẫn tệp tin tương đối hoặc tuyệt đối trên hệ thống tệp tin.

+ $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.

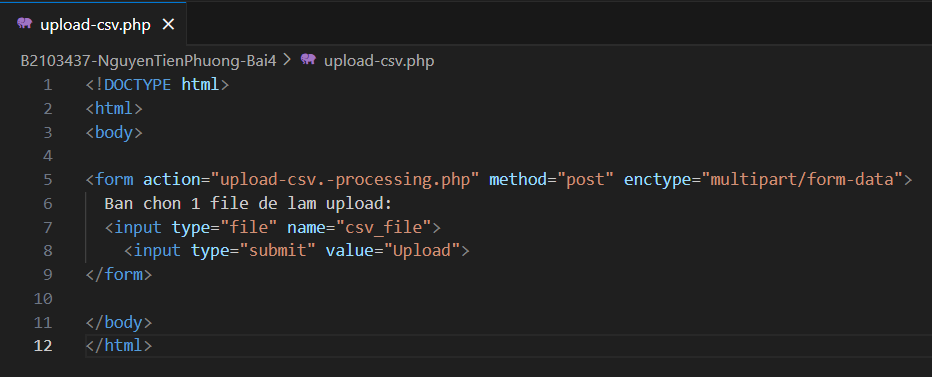
+ $context (tùy chọn): Một ngữ cảnh tệp tin được sử dụng để thiết lập các tùy chọn bổ sung cho việc đọc tệp tin, ví dụ như thông tin xác thực.

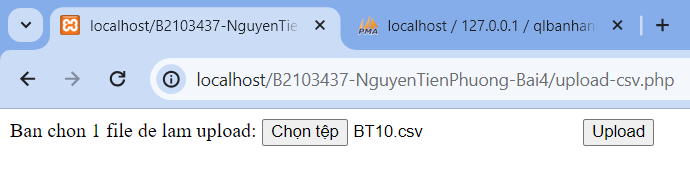
**10. Bạn hãy tạo 1 tập tin csv ít nhất 10 dòng dữ liệu với các cột dữ liệu như bảng customer trong csdl qlbanhang. Tạo tập tin upload-csv.php với giao diện cho phép upload các tập tin csv, và tập tin upload-csv-processing để xử lý nút xử lý sự kiện upload file csv và đưa dữ liệu vào bảng customers trong CSDL.**

Tập tin CSV(BT10.csv)

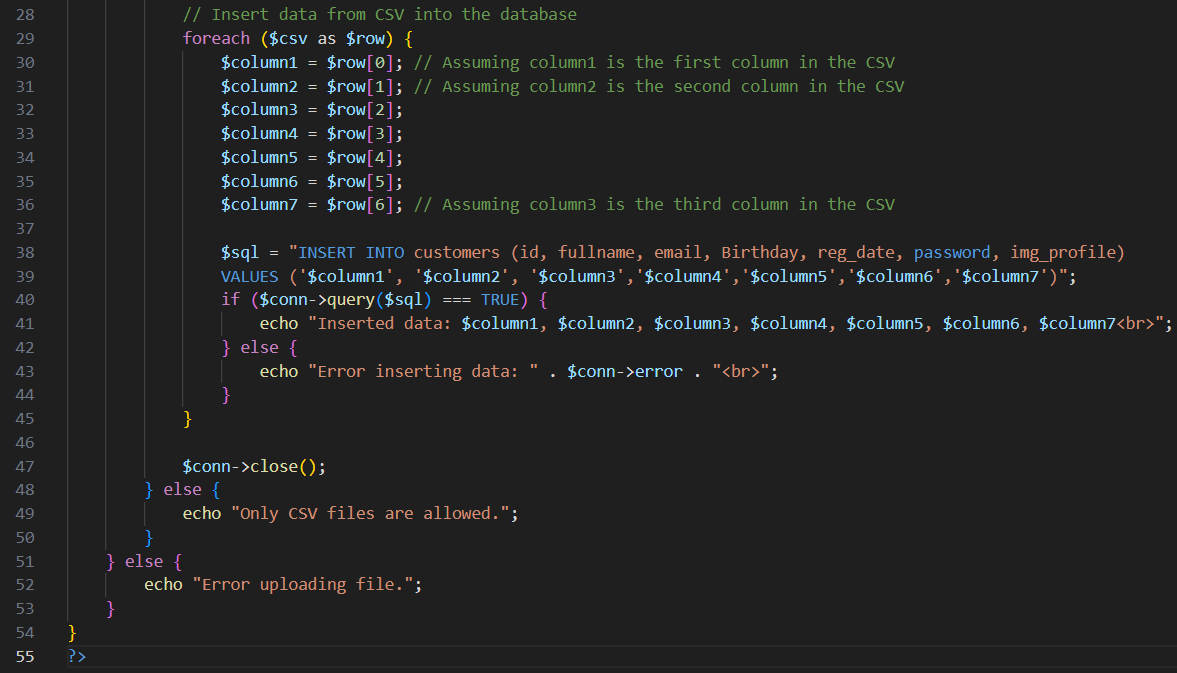
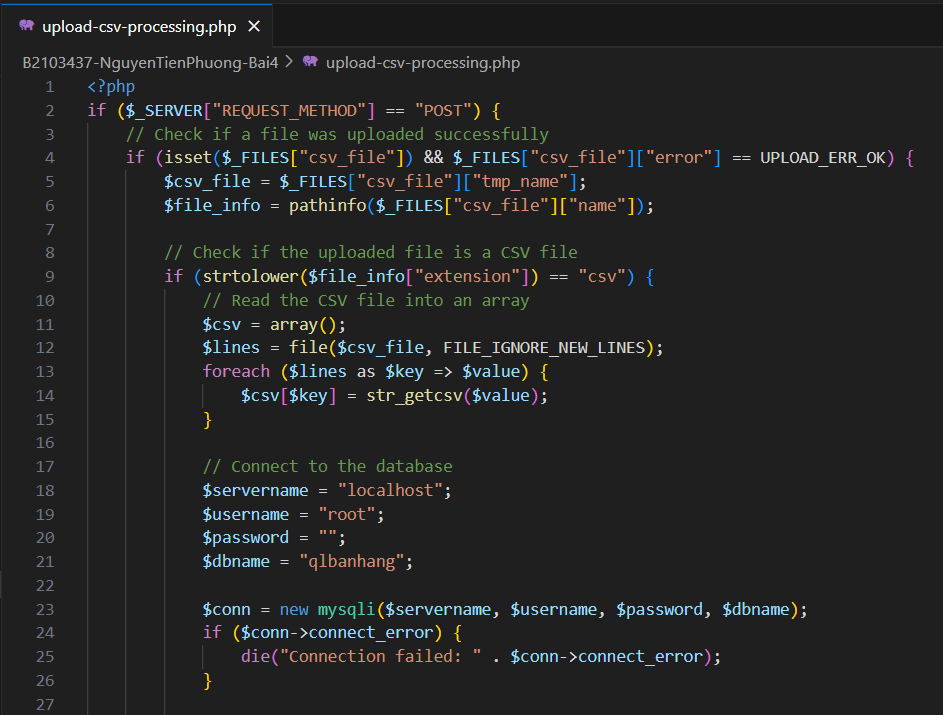


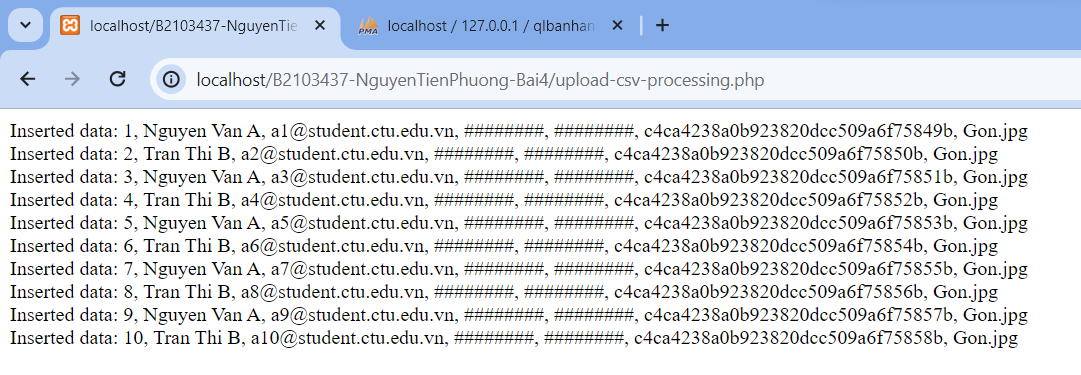
Tập tin upload-csv.php

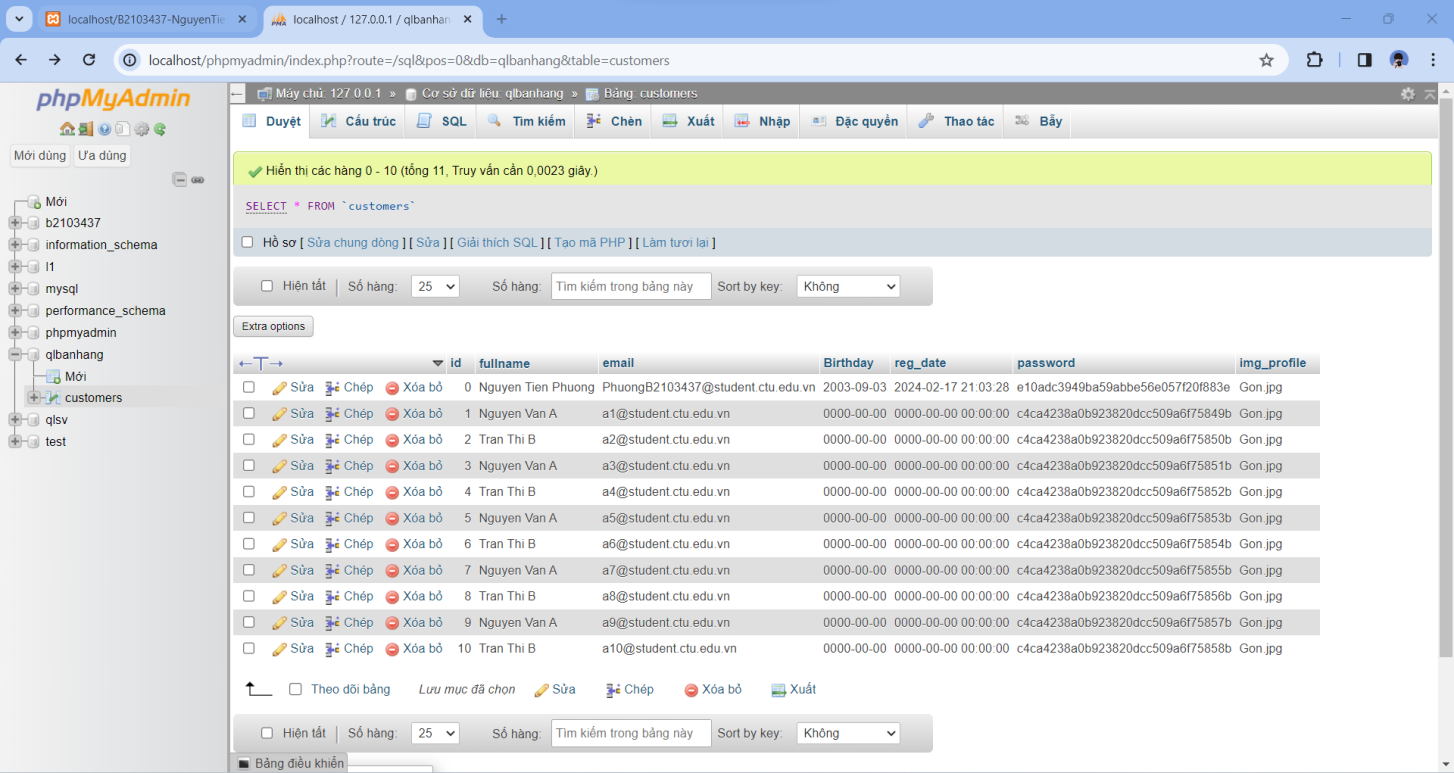




Tập tin upload-csv-processing.php







---Hết---