**Bài tập thực hành 04**

Lập trình PHP nâng cao

**Mục tiêu cần đạt**:

Sau buổi các bạn cần nắm thao tác sử dụng PHP nâng cao

● Hiểu và biết cách dùng Cookies và Session

● Tạo form đăng nhập và dùng cookies, session để quản lý các đăng nhập, thực hiện chức năng log out (thoát) khỏi hệ thống

● Biết về SQL Injection.

● Upload file, lưu thông tin về tập tin trong CSDL

● Đọc, và xử lý nội dung trong tập tin

Các bạn đọc và thực hiện viết code như hướng dẫn để làm quen, sau đó đọc trả lời các **Yêu cầu** ở dưới bài hướng dẫn để thực hiện yêu cầu bài thực hành.

**Gợi ý Tham khảo:**

● https://www.w3schools.com/php/

● và các nguồn khác mà sinh viên tìm được

Yêu cầu bài thực hành:

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

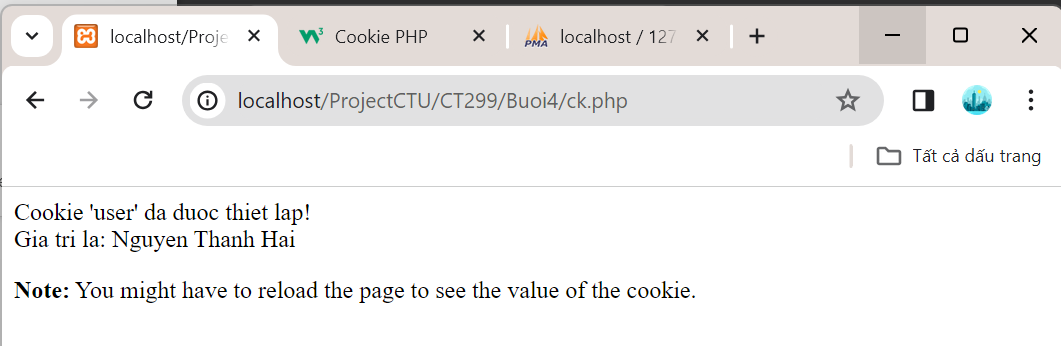
Tạo Cookies

Cookies và Session hỗ trợ chúng ta quản lý các phiên đăng nhập/đăng nhập, hoạt động của người dùng trên các website

Tạo tập tin sử dụng khởi tạo cookies ck.php sau:

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:

CREATE TABLE `customers` (

`id` int(6) UNSIGNED NOT NULL,

`fullname` varchar(50) CHARACTER SET latin1 NOT NULL,

`email` varchar(50) CHARACTER SET latin1 DEFAULT NULL,

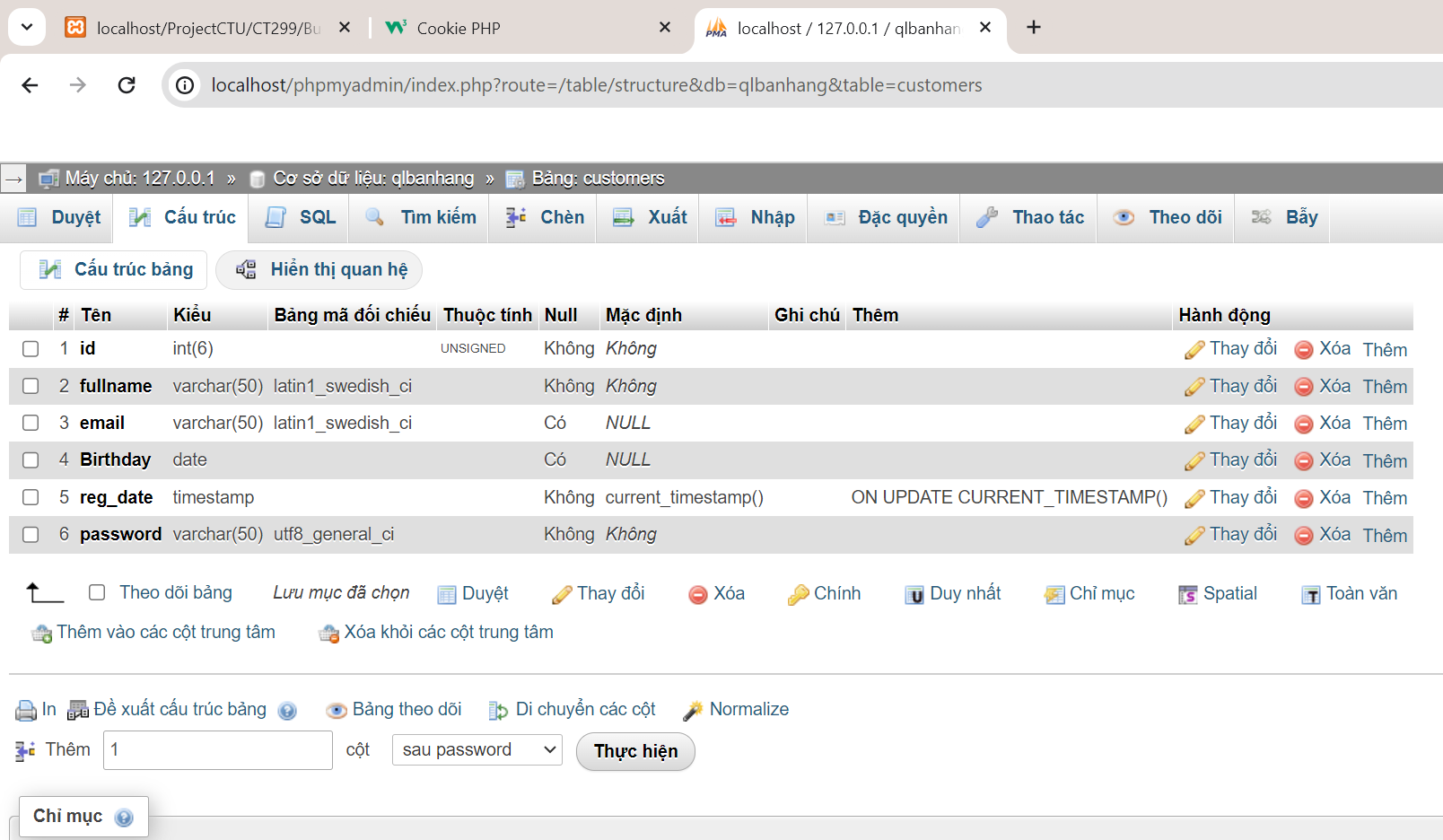
`Birthday` date DEFAULT NULL,

`reg\_date` timestamp NOT NULL DEFAULT current\_timestamp() ON UPDATE

current\_timestamp(),

`password` varchar(50) NOT NULL

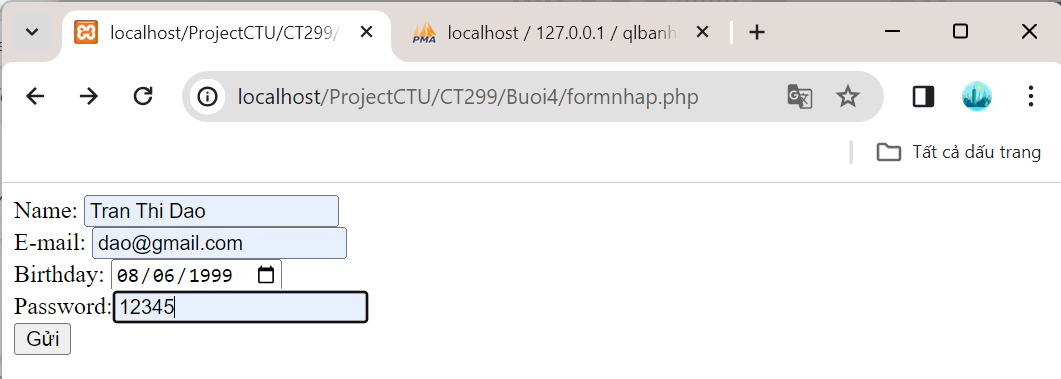
) ENGINE=InnoDB DEFAULT CHARSET=utf8;



Tạo formnhap.php

formnhap.php

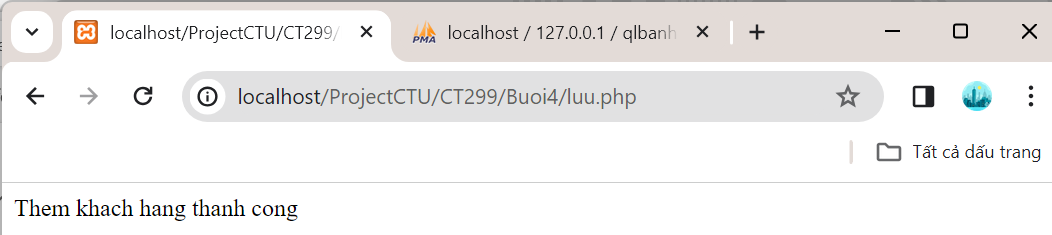
Kết quả:



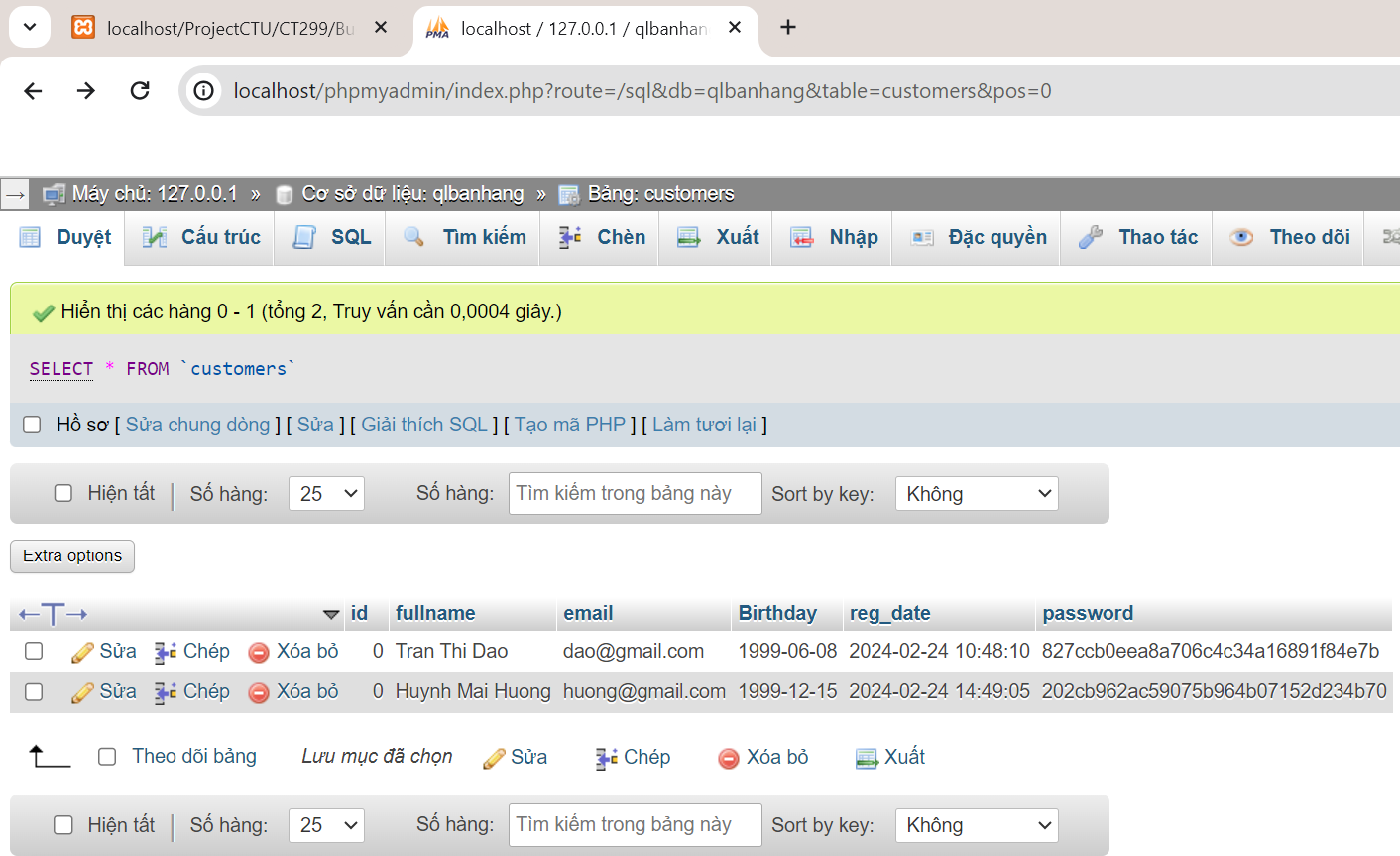
Và luu.php, chú ý lúc này chúng ta sẽ mã hóa mật khẩu với giải thuật md5 để bảo mật mật khẩu:

luu.php

Kết quả:



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:



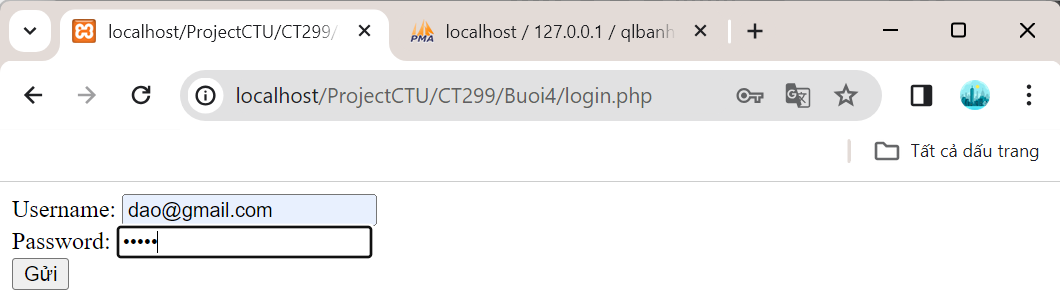
Trong trường hợp chúng ta muốn dùng email là username để đăng nhập ta nên đảm bảo 2 Customer không có email trùng nhau. Để đảm bảo điều này ta có thể chọn email là unique key (vì ta đã chọn id la primary key, và 1 bảng không thể có 2 khóa chính)

ALTER TABLE `customers` ADD UNIQUE( `email`);

Tạo form đăng nhập

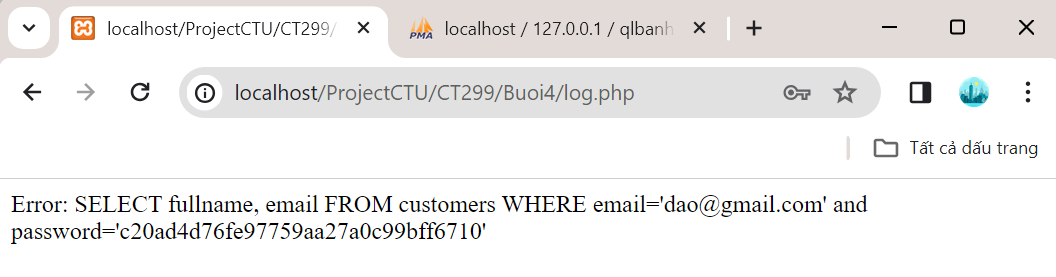
Sau khi tạo người dùng xong, ta có thể chạy form đăng nhập với username là email và password lấy từ CSDL

Login.php

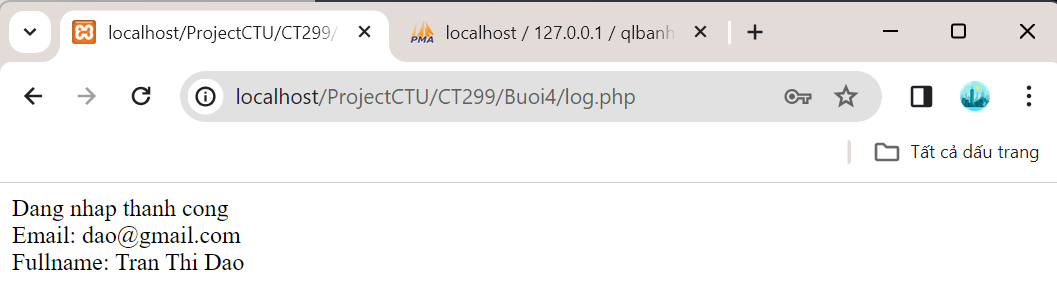


Log.php

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: đăng nhập thành công



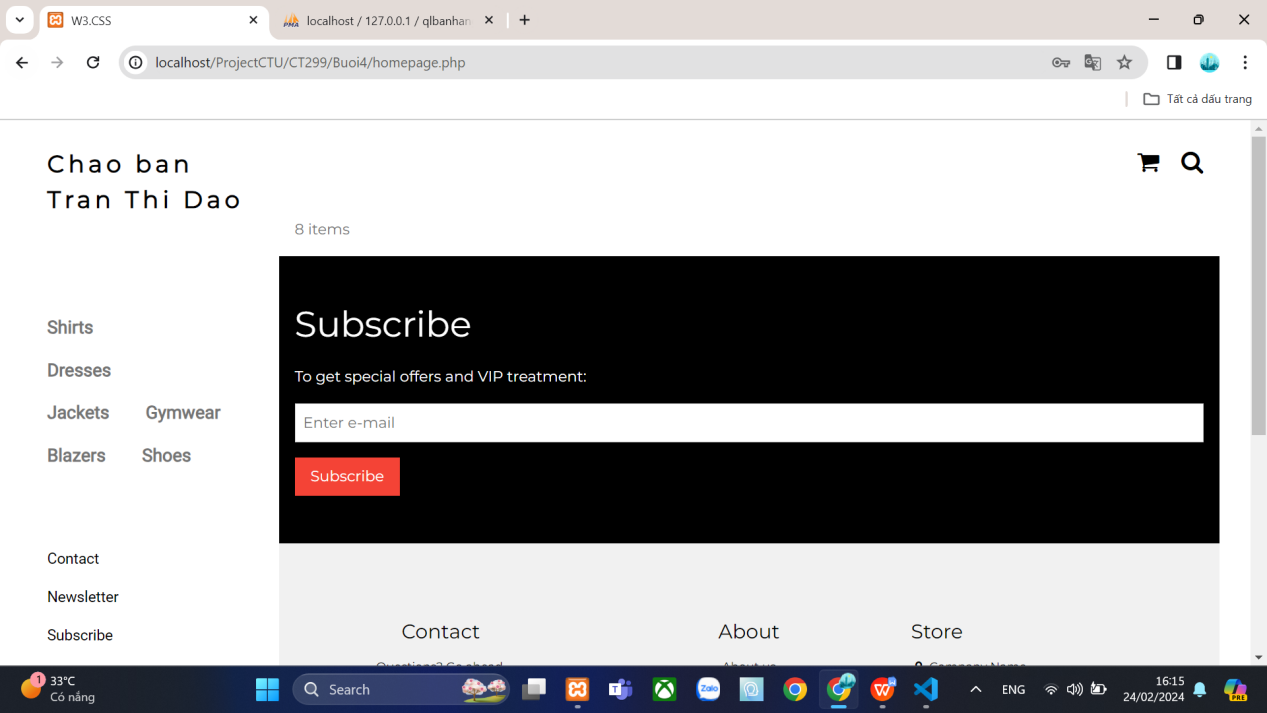
Dùng Cookies lưu lại tên đăng nhập, email và điều hướng

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

Log.php (modified)

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

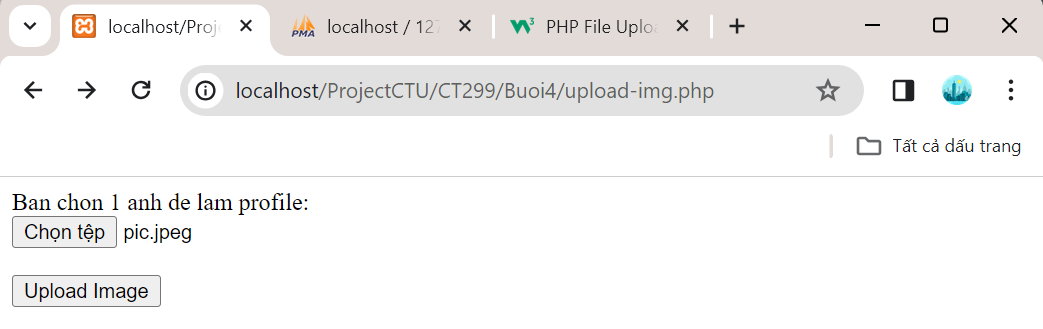
Homepage.php



Upload tập tin

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

Upload-img.php

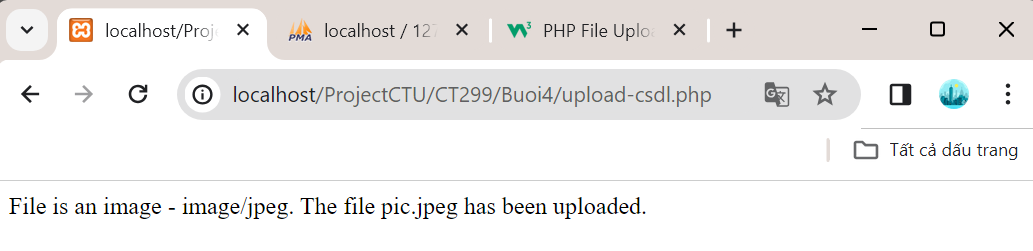


Tập tin upload.php xử lý hành động

upload.php

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

Upload vào **thư mục “upload/”**:



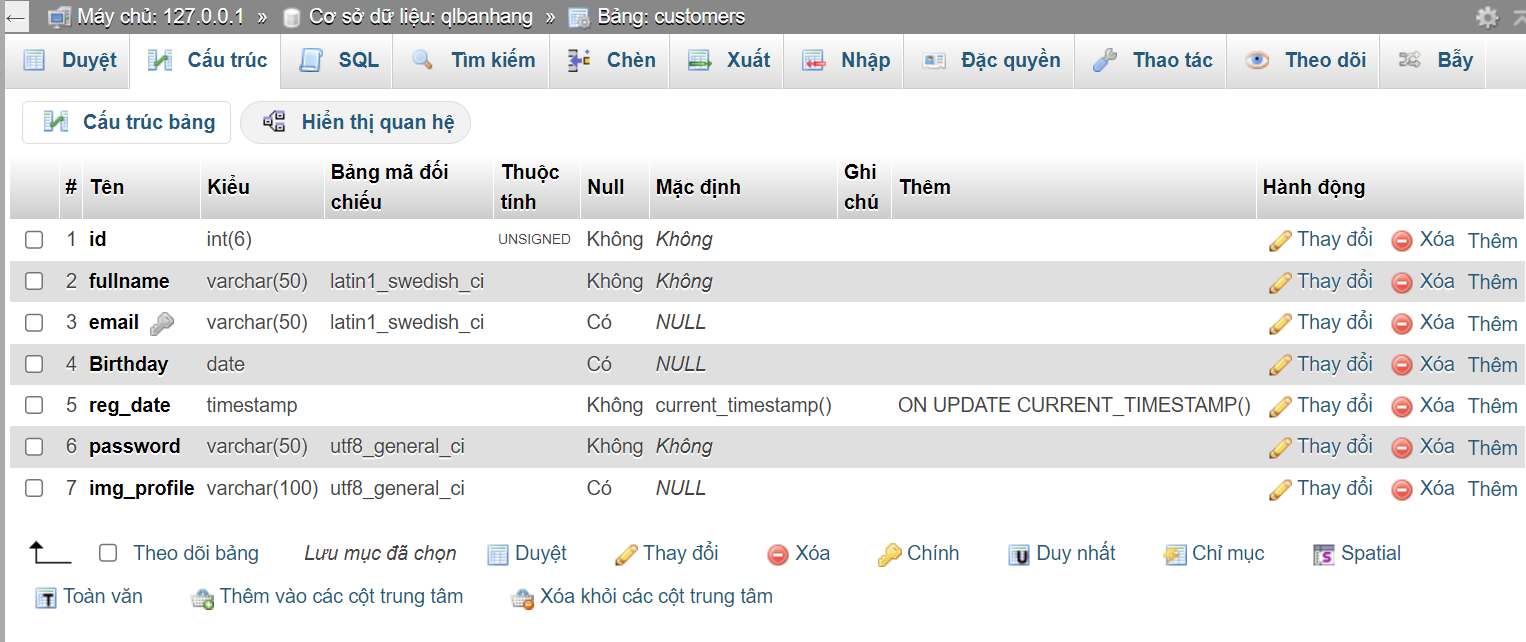
Upload ảnh và lưu dữ liệu về ảnh trong CSDL

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

Upload-img.php (modified)

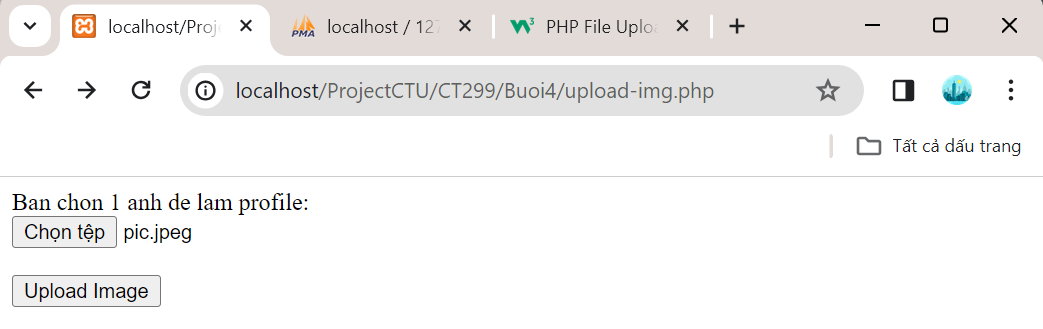
Thêm cột để lưu đường dẫn ảnh:

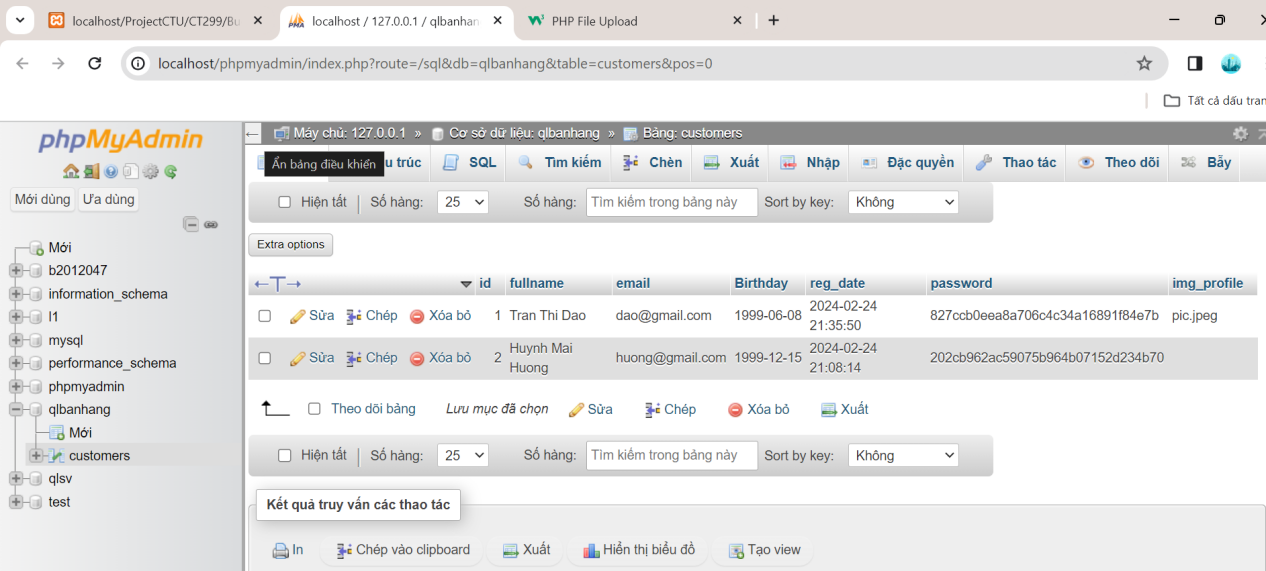
ALTER TABLE `customers` ADD `img\_profile` VARCHAR(100);



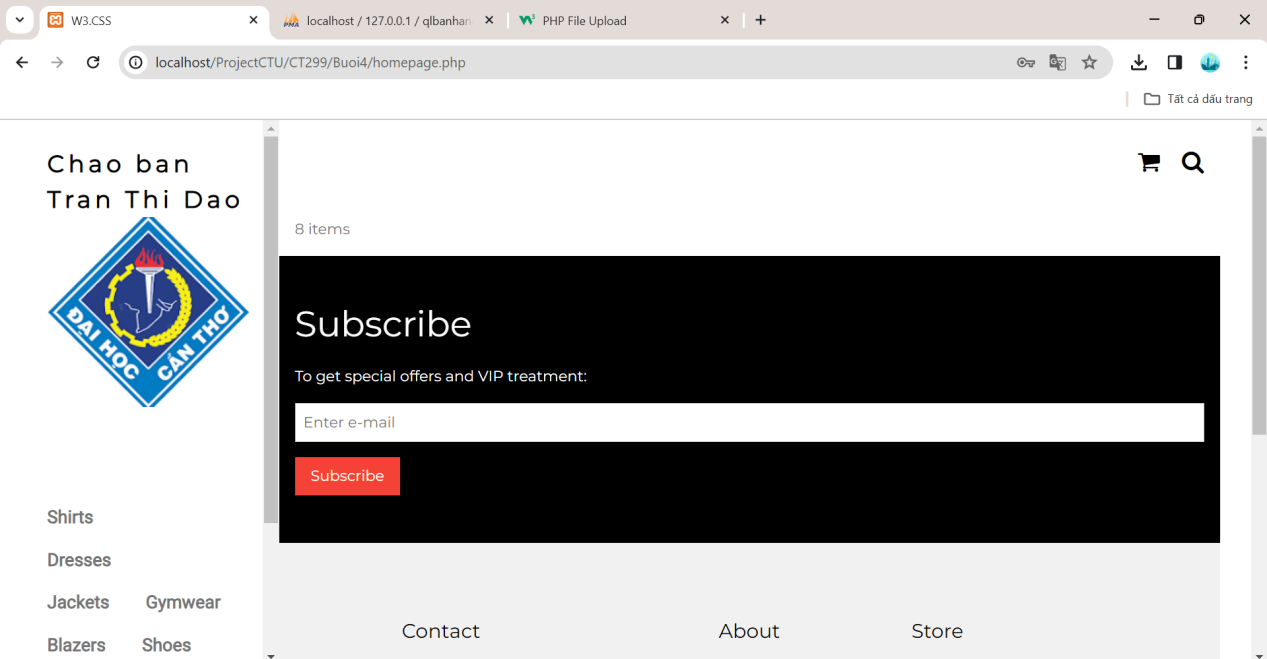
Upload-csdl.php

Upload vào **thư mục “upload/”**, đồng thời cập nhật vào **cơ sở dữ liệu**:





Homepage.php (modified với ảnh profile)



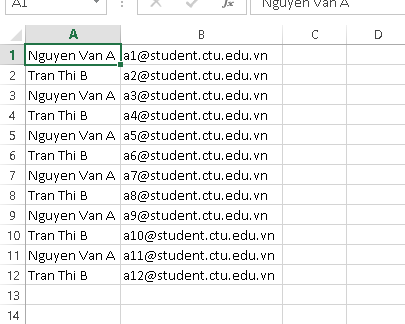
\*\*\*Ở đây do biểu diễn từng bước nên đoạn này chúng ta lấy dữ liệu link ảnh profile từ id của người dùng. Trong thực tế, các bạn có thể lưu link ảnh profile vào cookie/session ngay từ khi đăng nhập.

Upload file csv và đọc dữ liệu vào mảng

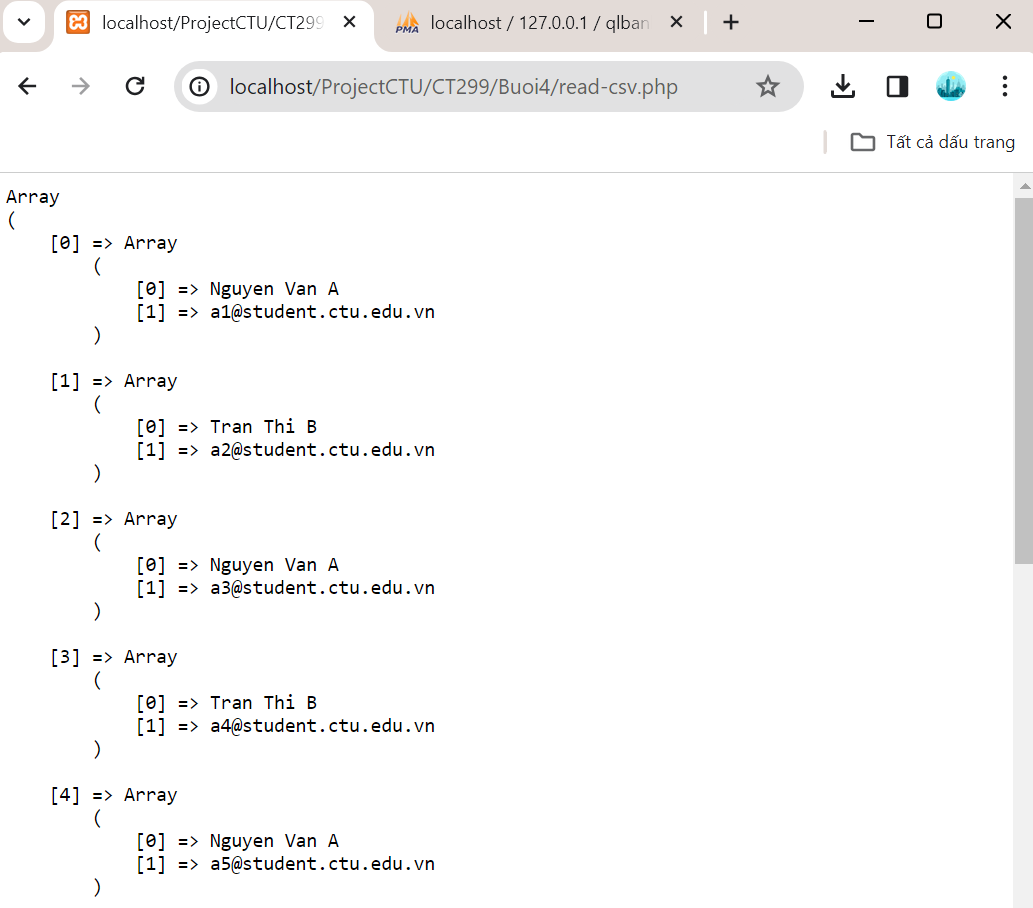
Thao tác đọc dữ liệu từ các tập tin là quan trọng, giúp chúng ta backup dữ liệu và phục hồi hay import dữ liệu từ các hệ thống khác. Ví dụ dưới đây giúp minh họa cho các bạn thao tác đọc dữ liệu từ tập tin csv và từ đó đưa vào mảng để phục vụ cho việc xử lý.

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:

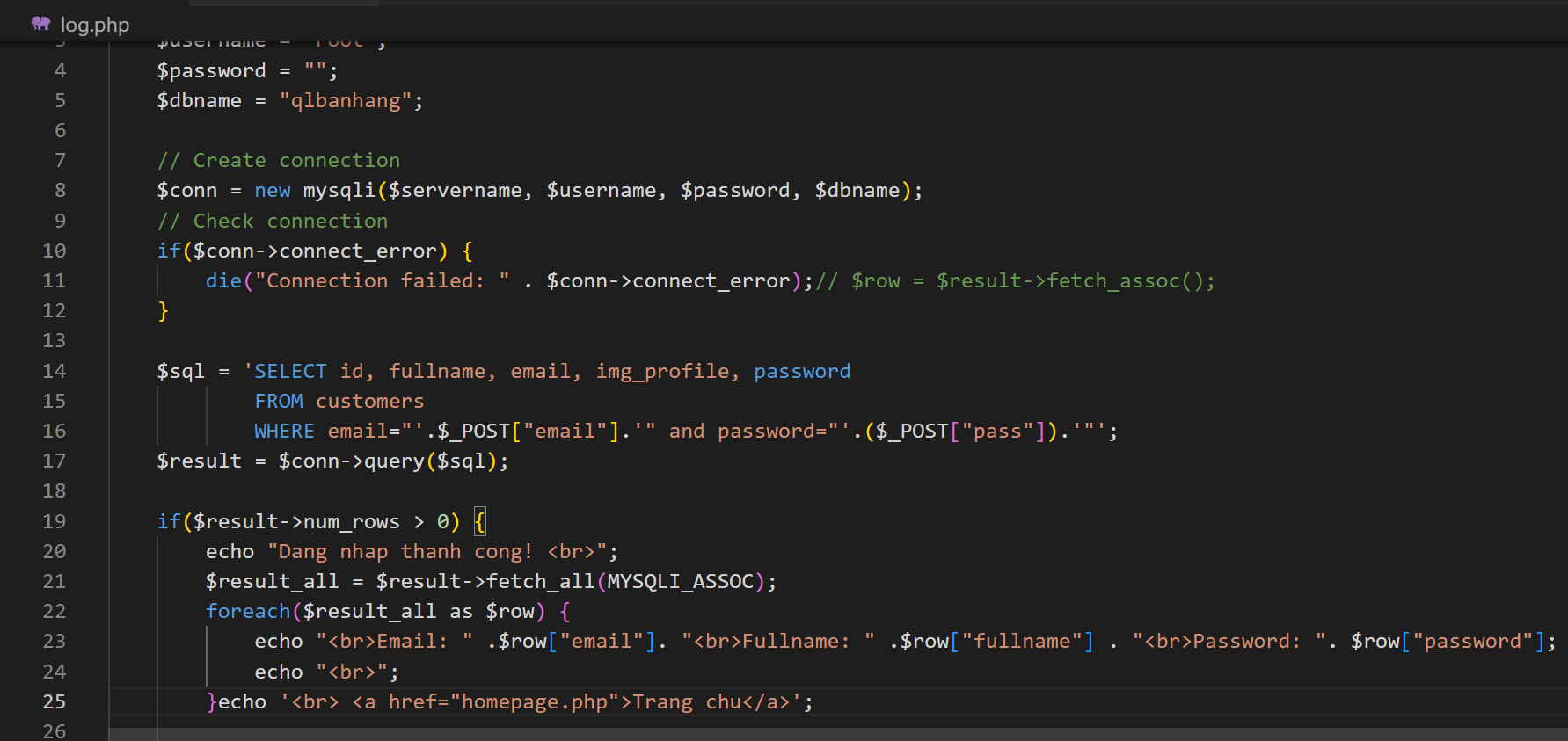


1. **Trong vài trường hợp, hacker có thể sử dụng các kỹ thuật SQL Injection để hack hệ thống của bạn. Bạn hãy trình bày SQL Injection là gì và thử các ví dụ trình bày trong https://www.w3schools.com/sql/sql\_injection.asp . Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.**

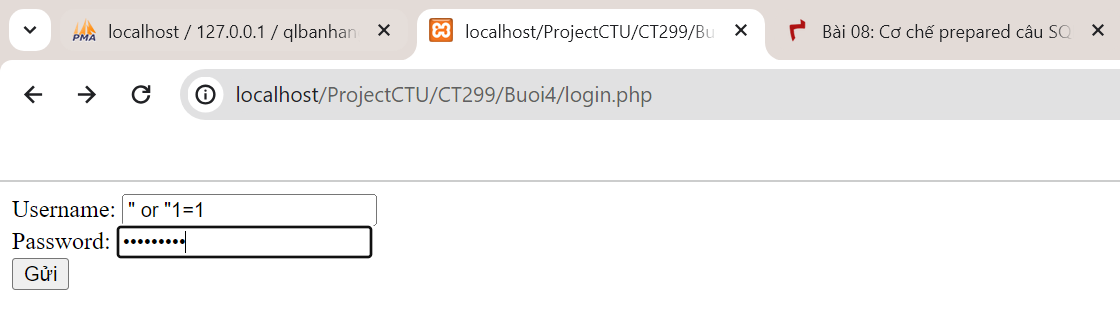
- SQL Injection là một kỹ thuật chèn mã nhằm phá hủy cơ sở dữ liệu, là một trong những kỹ thuật hack web phổ biến nhất. Việc đặt mã độc vào câu lệnh SQL thông qua đầu vào trang web (khi yêu cầu người dùng nhập dữ liệu người dùng sẽ cung cấp 1 câu lệnh SQL mà ta sẽ vô tình chạy trên cơ sở dữ liệu của mình).

- Bỏ giải thuật mã hoá password md5 để đăng nhập bằng kỹ thuật SQL Injection:

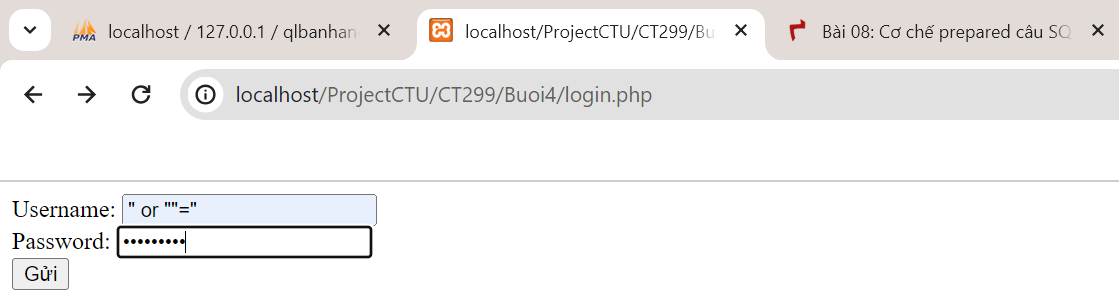
\*\*\* sửa lại 1 chút trong log.php để hiển thị danh sách khách hàng có trong csdl:



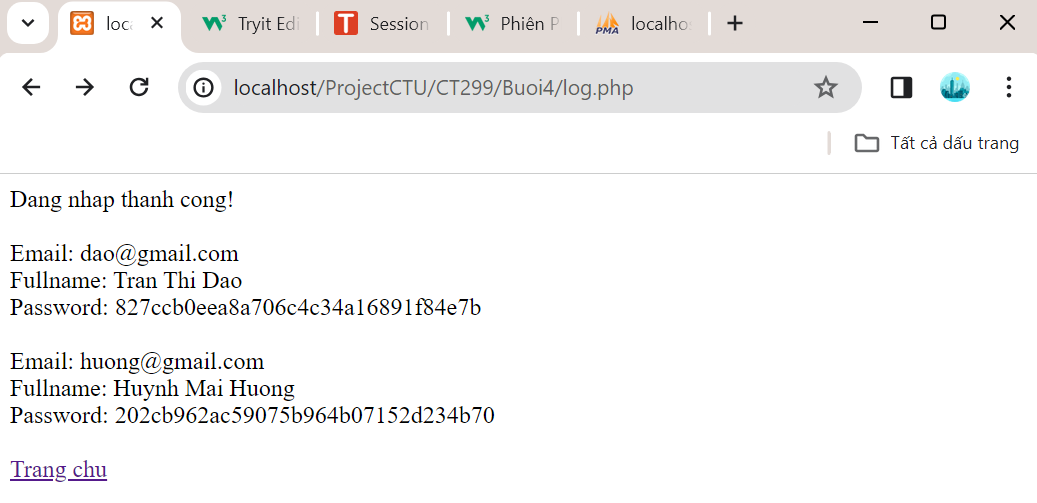
Cách 1:

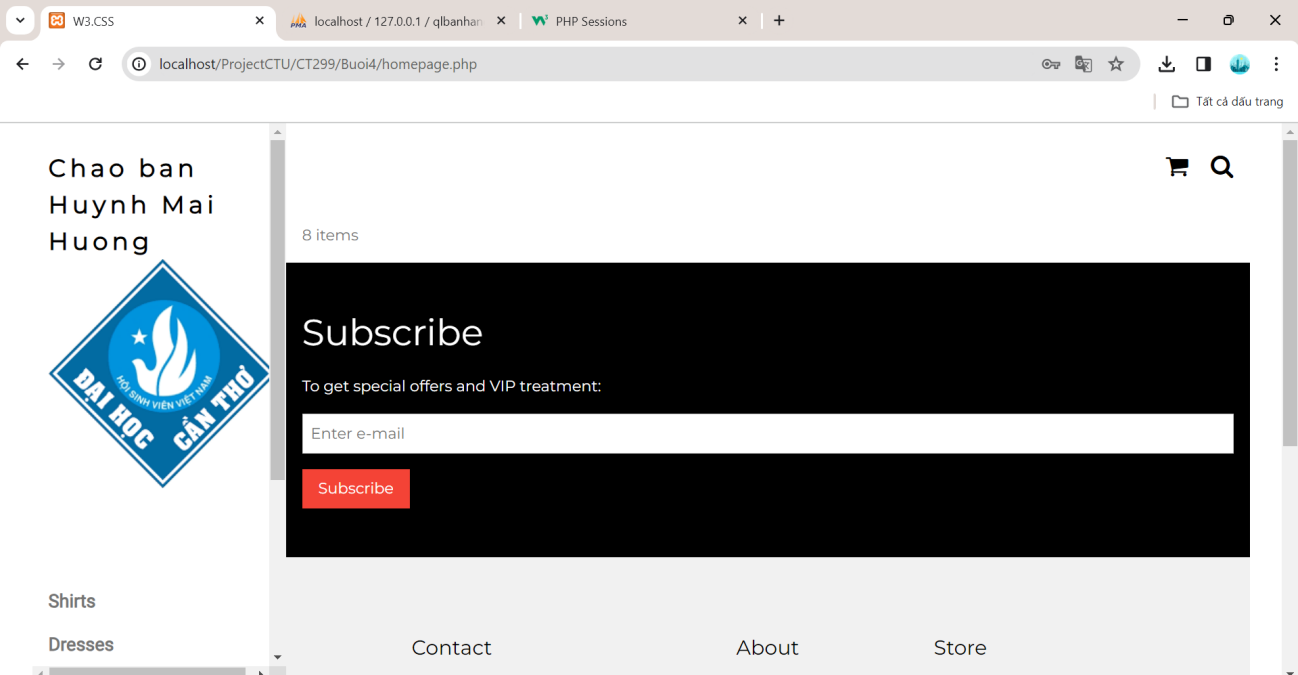


Cách 2:



=> Cả 2 cách đều có thể lấy hết dữ liệu khách hàng trong csdl và cho phép đăng nhập vào hệ thống:





1. **Dựa vào link https://www.w3schools.com/php/php\_cookies.asp, 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 1 tệp nhỏ mà máy chủ nhúng vào máy tính của người dùng. Cookie thường được sử dụng để nhận dạng người dùng. Mỗi lần máy tính đó yêu cầu một trang bằng trình duyệt, nó cũng sẽ gửi cookie. Với PHP, có thể vừa tạo vừa truy xuất các giá trị cookie.

- Diễn giải ý nghĩa các tham số trong setcookie():

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

|  |  |
| --- | --- |
| **Tham số** | **Mô tả** |
| *name (bắt buộc)* | Tên cookie |
| *value* | Giá trị cookie |
| *expire* | - Thời điểm cookie hết hạn  - Mặc định là 0 (cookie sẽ hết hạn vào cuối phiên - khi đóng trình duyệt)  - Vd: time()+86400\*30, sẽ đặt cookie hết hạn sau 30 ngày |
| *path* | - Đường dẫn máy chủ của cookie  - Nếu được đặt thành "/", cookie sẽ khả dụng trong toàn bộ miền  - Nếu được đặt thành "/php/", cookie sẽ chỉ khả dụng trong thư mục php và tất cả các thư mục con của php  - Giá trị mặc định là thư mục hiện tại mà cookie đang được đặt |
| *domain* | - Tên miền của cookie |
| *secure* | - Chỉ định cookie chỉ được truyền qua kết nối HTTPS an toàn.  + TRUE, cookie sẽ chỉ được đặt nếu tồn tại kết nối an toàn.  + Mặc định là FALSE |
| *httponly* | - TRUE, cookie sẽ chỉ có thể truy cập được thông qua giao thức HTTP (cookie sẽ không thể truy cập được bằng ngôn ngữ tập lệnh). Cài đặt này có thể giúp giảm hành vi trộm cắp danh tính thông qua các cuộc tấn công XSS.  - Mặc định là FALSE |

- Cách lưu, lấy, xóa giá trị trong cookie:

+ Lưu: sử dụng hàm setcookie():

Ví dụ: $cookie\_name = "user";  
 $cookie\_value = "John Doe";  
 setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 30), "/");

Lưu ý: Hàm setcookie() phải xuất hiện TRƯỚC thẻ <html>.

+ Lấy: sử dụng biến toàn cục: $\_COOKIE[$cookie\_name]

Ví dụ: $cookie\_name = "user";  
 $cookie\_value = "John Doe";  
 echo $\_COOKIE[$cookie\_name];

Hoặc echo $\_COOKIE["user"];

+ Xoá: sử dụng hàm setcookie() có ngày hết hạn trong quá khứ

Ví dụ: setcookie("user", "", time() - 3600);

1. **Dựa vào link https://www.w3schools.com/php/php\_sessions.asp, bạn hãy cho biết Session dùng để làm gì. Cách lưu, lấy, xóa giá trị trong Session.**

- Session là một cách lưu trữ thông tin (theo biến) để sử dụng trên nhiều trang. Một session bắt đầu khi client gửi request đến server, nó tồn tại xuyên suốt từ trang này đến trang khác trong ứng dụng web. Theo mặc định, các biến session kéo dài cho đến khi người dùng đóng trình duyệt.

- Cách lưu, lấy, xóa giá trị trong Session: Trước tiên, 1 phiên được bắt đầu với session\_start() trước thẻ <html>

+ Lưu: sử dụng hàm biến toàn cục $\_COOKIE[""]

Ví dụ: $\_SESSION["favcolor"] = "green";  
 $\_SESSION["favanimal"] = "cat";

+ Lấy: sử dụng hàm biến toàn cục $\_COOKIE[""]:

Ví dụ: echo $\_SESSION["favcolor"];

echo $\_SESSION["favanimal"];

+ Xoá: để xóa tất cả các biến phiên toàn cục và hủy phiên, sử dụng session\_unset() và session\_destroy()

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

|  |  |
| --- | --- |
| **Cookie** | **Session** |
| Được lưu trữ trên trình duyệt của người dùng | Không được lưu trữ trên trình duyệt |
| Dữ liệu được lưu trữ ở phía client | Dữ liệu được lưu trữ ở phía server |
| Dữ liệu dễ dàng sửa đổi hoặc đánh cắp khi chúng được lưu trữ ở phía client. | Dữ liệu không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ |
| Dữ liệu có sẵn trong trình duyệt đến khi expired. | Sau khi đóng trình duyệt sẽ hết phiên làm việc (session) |

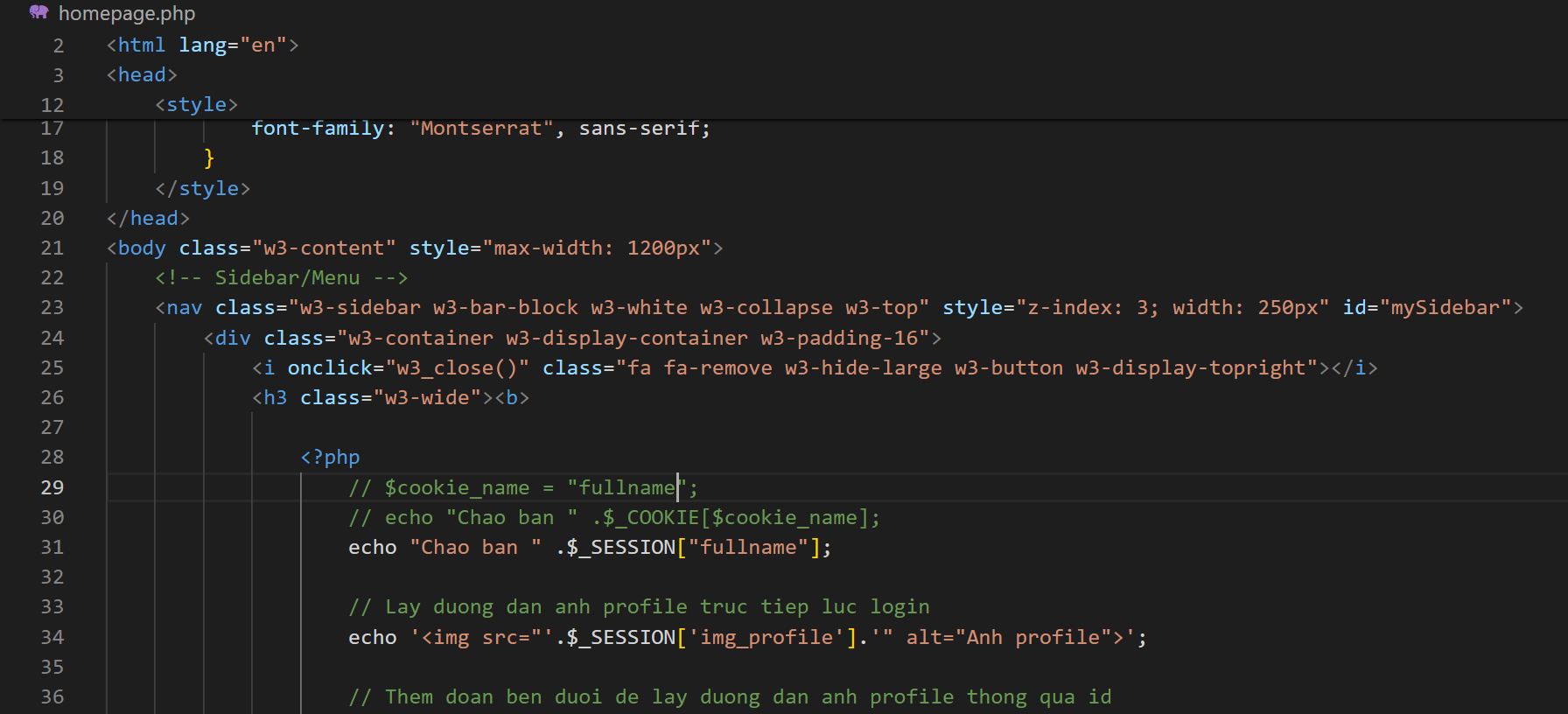
1. **Chỉnh sửa các đoạn gán, khởi tạo, lấy giá trị cookie trong log.php và homepage.php thay bằng cách dùng Session. Tạo tập tin thoat.php để xóa các giá trị trong session, cookie thực hiện chức năng log out khỏi hệ thống.**

- Chỉnh cookie sang session:

log.php



homepage.php



- Tạo tập tin thoat.php:

Trước tiên thêm nút “LOG-OUT” ở homepage.php:



thoat.php



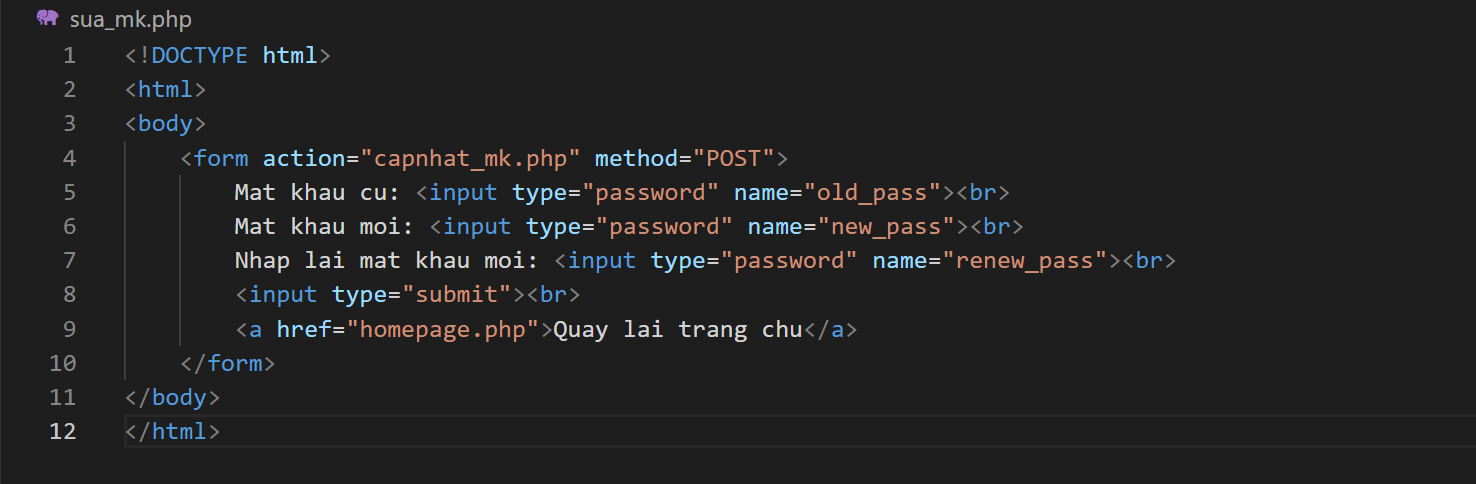
1. **Bạn hãy tạo form sua\_mk.php cho phép người dùng chỉnh sửa mật khẩu sau khi đăng nhập. Yêu cầu gồm 3 input: ô để nhập mật khẩu cũ, 1 ô để nhập mật khẩu mới, 1 ô cho phép nhập mật lại mật khẩu mới. Khi nhập xong cần đảm bảo: mật khẩu cũ là khớp với CSDL đang lưu, 2 ô nhập mật khẩu mới phải khớp với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

- Trước tiên thêm nút “Change your pasword” ở homepage.php:

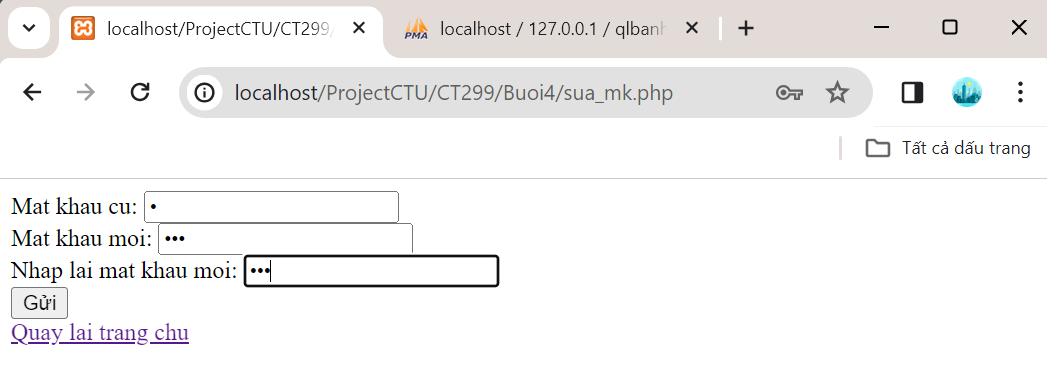


- Form sua\_mk.php

Code:

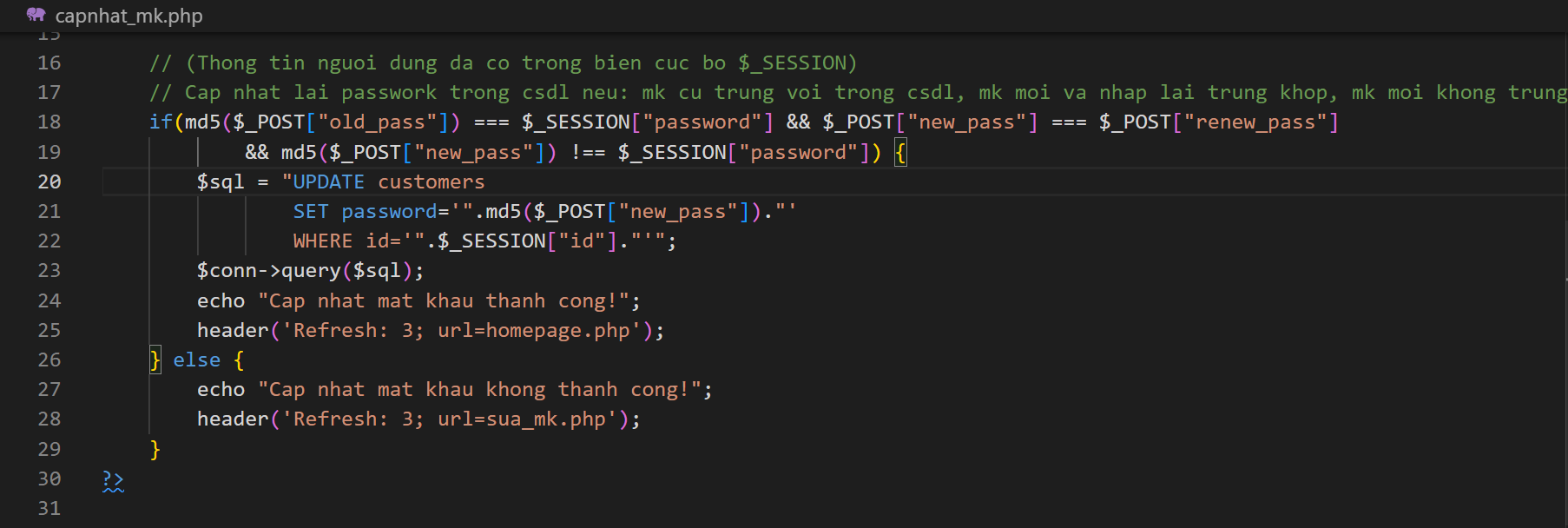


Kết quả:

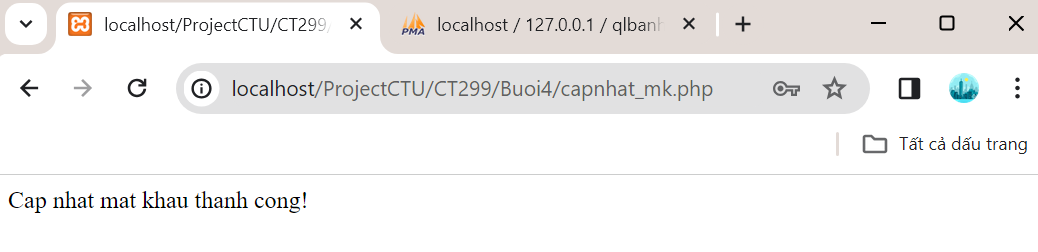


- capnhat\_mk.php

Code:



Kết quả:



1. **Bạn đọc trong <https://www.w3schools.com/php/php_file_upload.asp> để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.**

- Cơ chế, các hàm/thủ tục để thực hiện việc upload:

+ Khai báo các biến sau:

* $target\_dir = "uploads/" - chỉ định thư mục nơi tệp sẽ được đặt
* $target\_file chỉ định đường dẫn của tệp sẽ được tải lên
* $imageFileType chứa phần mở rộng tệp của tệp (viết thường)
* $uploadOk=1 để xác định tải tệp lên thành công không

+ Tạo một thư mục mới có tên "uploads" trong thư mục chứa tệp "upload.php". Các tập tin được tải lên sẽ được lưu ở đó.

+ Tiến hành kiểm tra tính hợp lệ của tệp, cập nhật $uploadOk=0 nếu:

* Tệp đã tồn tại, kiểm tra bằng hàm file\_exists()
* Vượt quá giới hạn kích thước, sai loại tệp

+ Nếu tệp hợp lệ ($uploadOk=1), tiến hành upload bằng hàm move\_uploaded\_file() để di chuyển tệp được tải lên vào một nơi được chỉ định.

1. **Bạn hãy đọc https://www.w3schools.com/php/func\_filesystem\_file.asp, mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).**

- Chức năng của 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.

- Các tham số trong hàm: file*(filename, flag, context)*

|  |  |
| --- | --- |
| **Tham số** | **Mô tả** |
| *filename (bắt buộc)* | Chỉ định đường dẫn tới file cần đọc |
| *flag* | Có thể là một hoặc nhiều hằng số sau:   * FILE\_USE\_INCLUDE\_PATH - Tìm kiếm tệp trong include\_path (trong php.ini) * FILE\_IGNORE\_NEW\_LINES - Bỏ qua dòng mới ở cuối mỗi phần tử mảng * FILE\_SKIP\_EMPTY\_LINES - Bỏ qua các dòng trống trong tệp |
| *context* | Chỉ định ngữ cảnh của việc xử lý tập tin. Ngữ cảnh là một tập hợp các tùy chọn có thể sửa đổi hành vi của luồng. Có thể bỏ qua bằng cách sử dụng NULL |

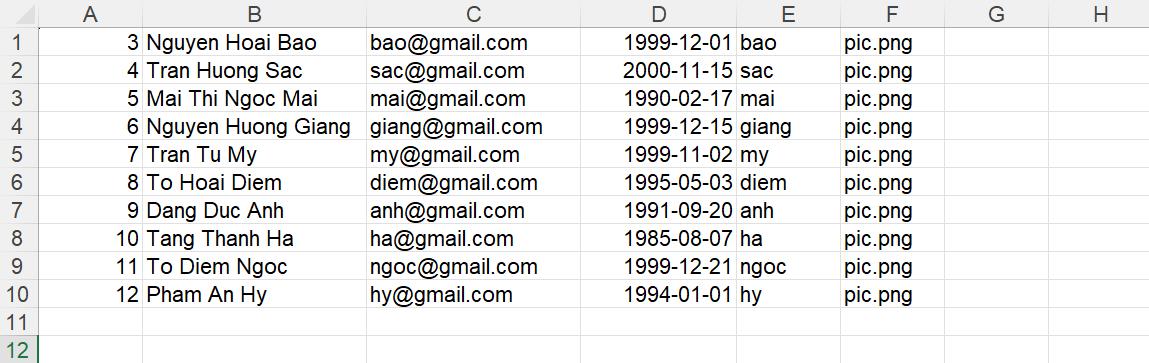
**10. Bạn hãy tạo 1 tập tin csv ít nhất 10 dòng dữ liệu với các cột dữ liệu như bảng customer trong csdl qlbanhang. Tạo tập tin upload-csv.php với giao diện cho phép upload các tập tin csv, và tập tin upload-csv-processing để xử lý nút xử lý sự kiện upload file csv và đưa dữ liệu vào bảng customers trong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

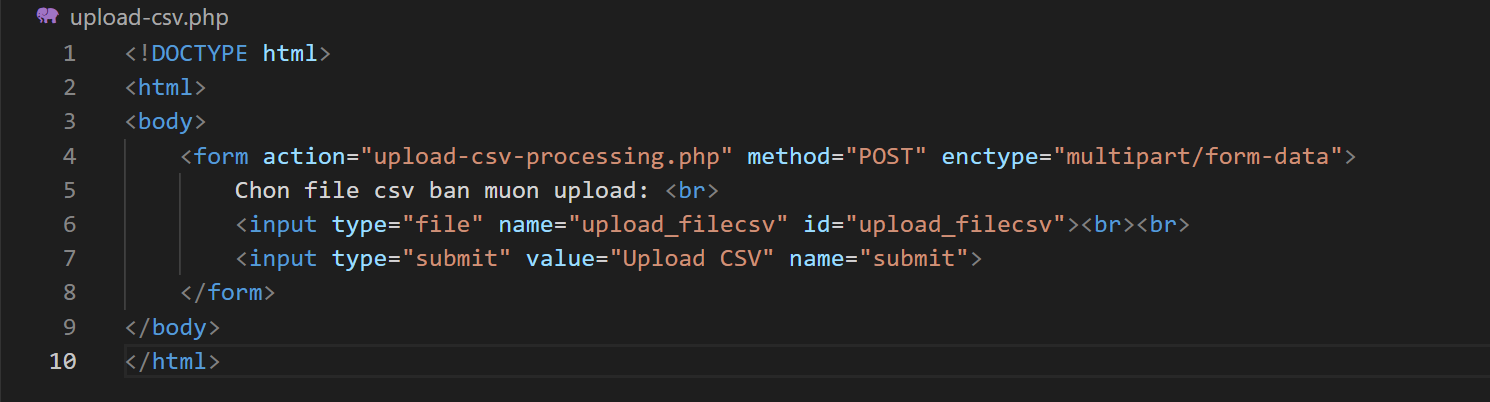
**○ Tham khảo upload-csdl.php để thiết kế action xử lý việc upload file, chú ý chỉnh sửa loại tập tin chấp nhận các file csv. Tham khảo link https://www.php.net/manual/en/features.file-upload.post-method.php , https://stackoverflow.com/questions/6654351/check-file-uploaded-is-in-csv-format . Ở đoạn sau 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.**

- Tạo tập tin Infocustomers.csv chứa dữ liệu customers:

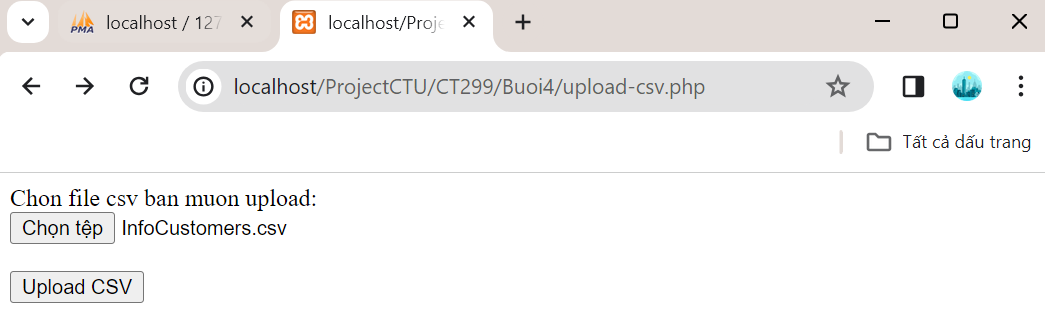


- Tạo tập tin upload-csv.php với giao diện cho phép upload các tập tin csv

Code:

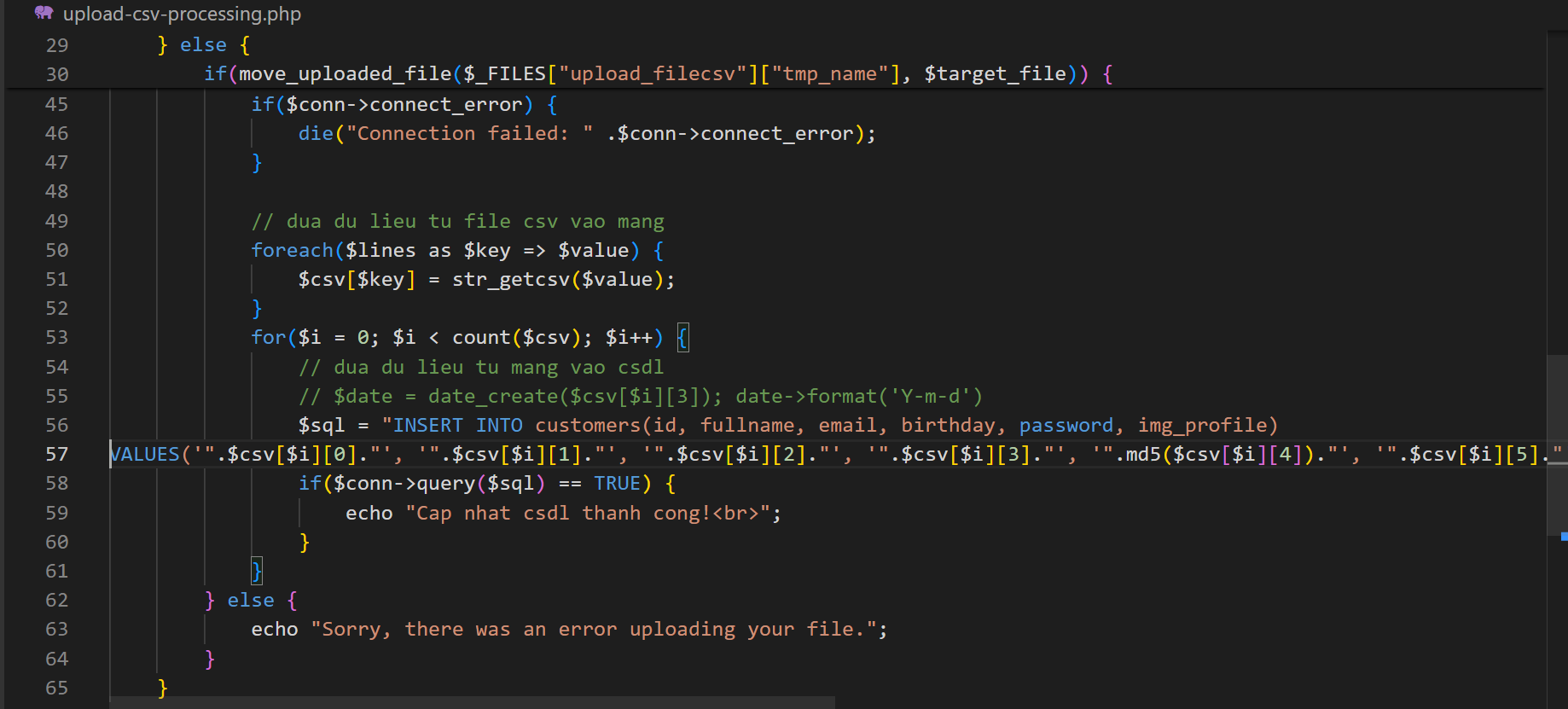


Kết quả:

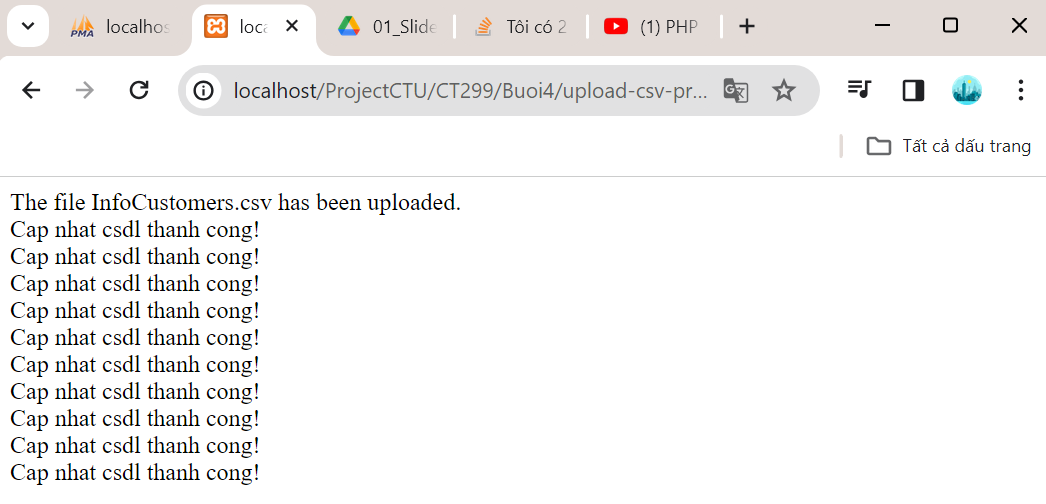


- Tập tin upload-csv-processing để xử lý nút xử lý sự kiện upload file csv và đưa dữ liệu vào bảng customers trong CSDL

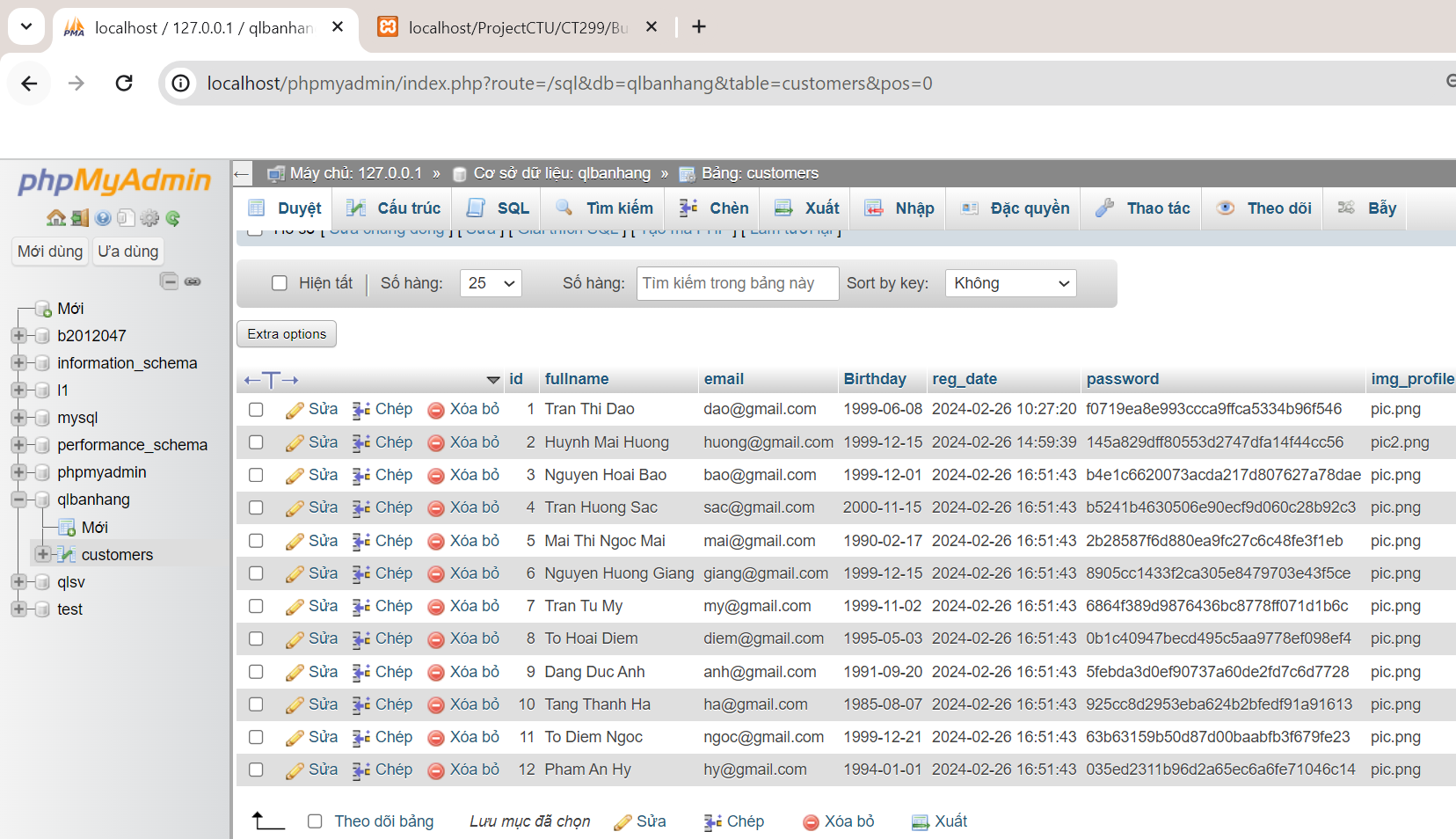
Code:



Kết quả:



Dữ liệu mới đã được đưa vào bảng customers trong CSDL



**Chú ý:**

● Các bạn nộp file word: Quy tắc đặt tên file: **<mssv>-<hoten>-<bai><stt\_bai thực**

**hành>.docx** nộp lên Classroom (VD: **B123456-NguyenVanA-bai1.docx**), kèm với các file khác được yêu cầu như phần câu hỏi đã nêu. **Ngoại trừ file word trả lời câu hỏi, các file còn lại các bạn nén vào 1 file zip**. File zip đặt tên như file word.

● Mỗi câu các bạn trả lời bằng hình hoặc dạng text tùy vào yêu cầu của câu hỏi và **TRẢ LỜI THEO ĐÚNG THỨ TỰ CÂU HỎI**. Nếu câu nào không trả lời được các

bạn cứ để số thứ tự câu hỏi và bỏ trống phần trả lời.

● Các câu trả lời có tham khảo trên Internet phải trích dẫn link/nguồn.

● *Vi phạm 1 trong các điều sau đây bài thực hành sẽ bị 0 điểm:*

○ Đặt tên KHÔNG ĐÚNG quy tắc được yêu cầu.

○ Bài không đủ các thành phần (word, code+data (nếu có),...) đã được yêu

cầu.

○ Bài không thực hiện đúng yêu cầu “**Ngoại trừ file word trả lời câu hỏi, các**

**file còn lại các bạn nén vào 1 file .zip”**

○ Bị phát hiện copy, sao chép từ các bạn khác

○ Phần trả lời không ghi rõ trả lời cho câu nào

○ Thứ tự câu trả lời không đúng thứ tự câu hỏi