Bài tập thực hành 04

**MSSV: B22034698**

**Họ và tên sinh viên: Trương Hữu Thái**

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

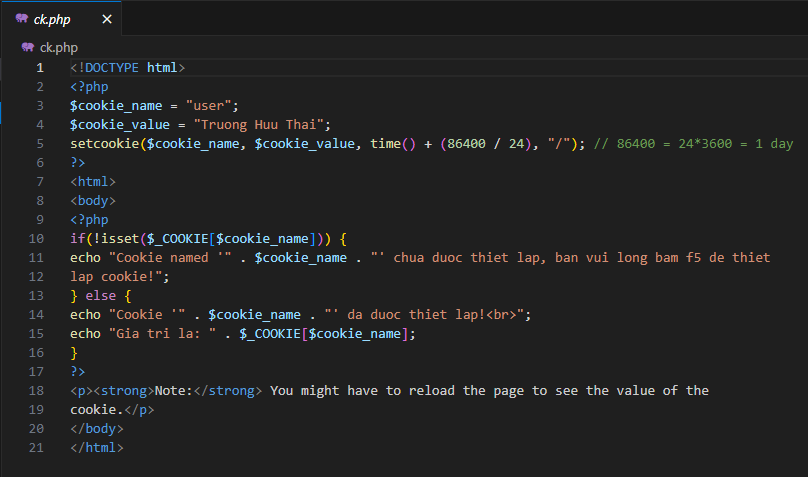
Các bạn đọc và thực hiện viết code như hướng dẫn để làm quen, sau đó đọc trả lời các **Yêu cầu** ở dưới bài hướng dẫn để thực hiện yêu cầu bài thực hành.

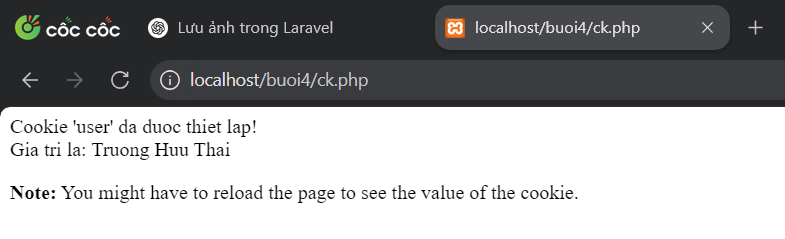
**Gợi ý Tham khảo:**

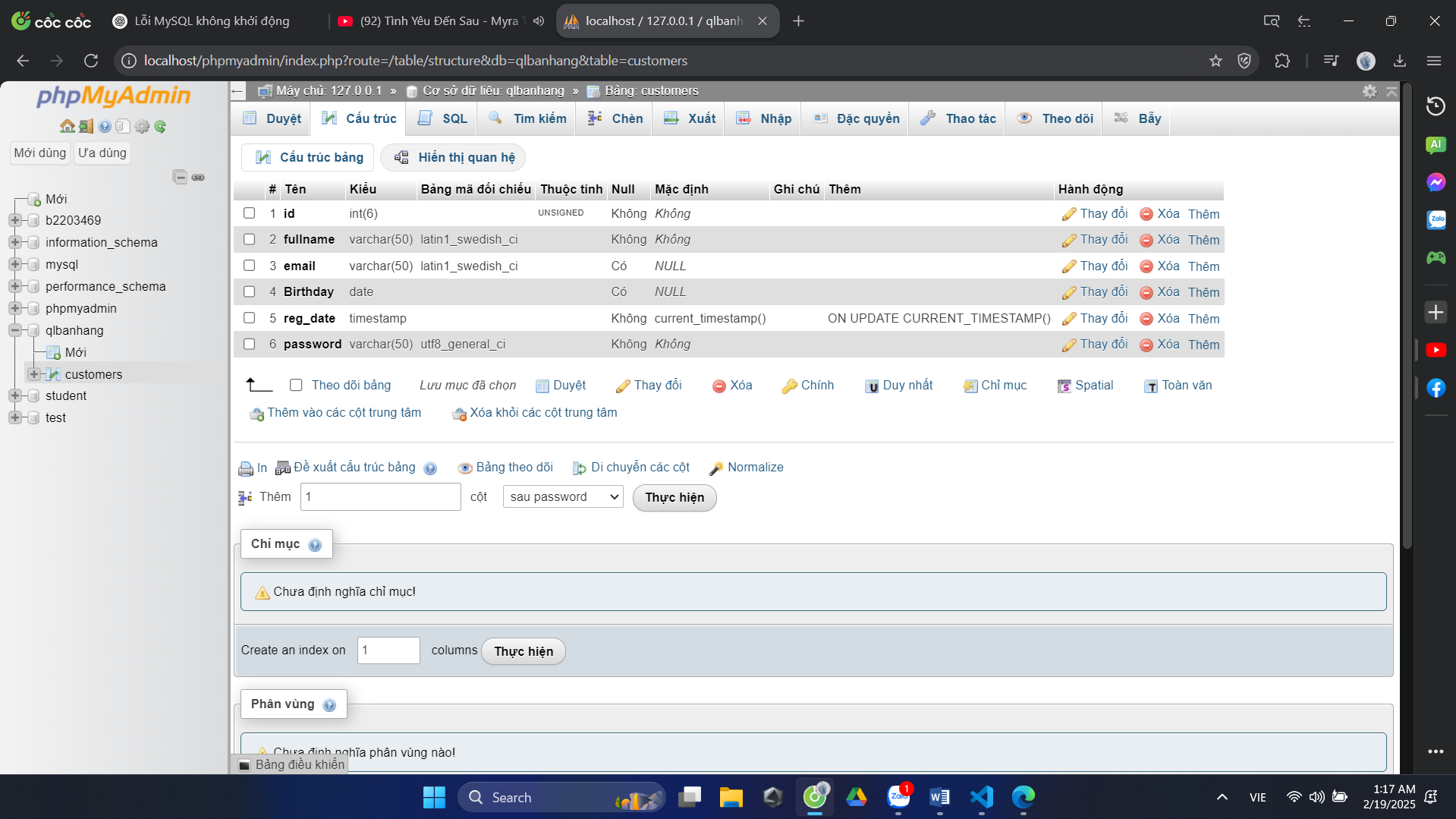
* <https://www.w3schools.com/php/>
* và các nguồn khác mà sinh viên tìm được

# Yêu cầu bài thực hành:

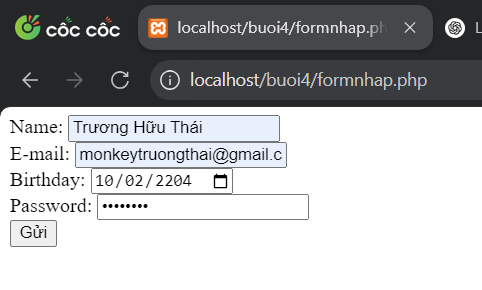
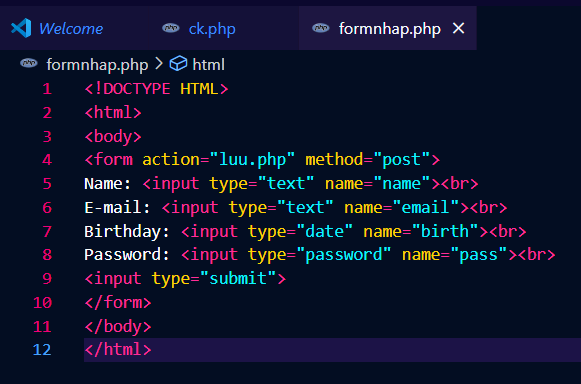
1. Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.

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:



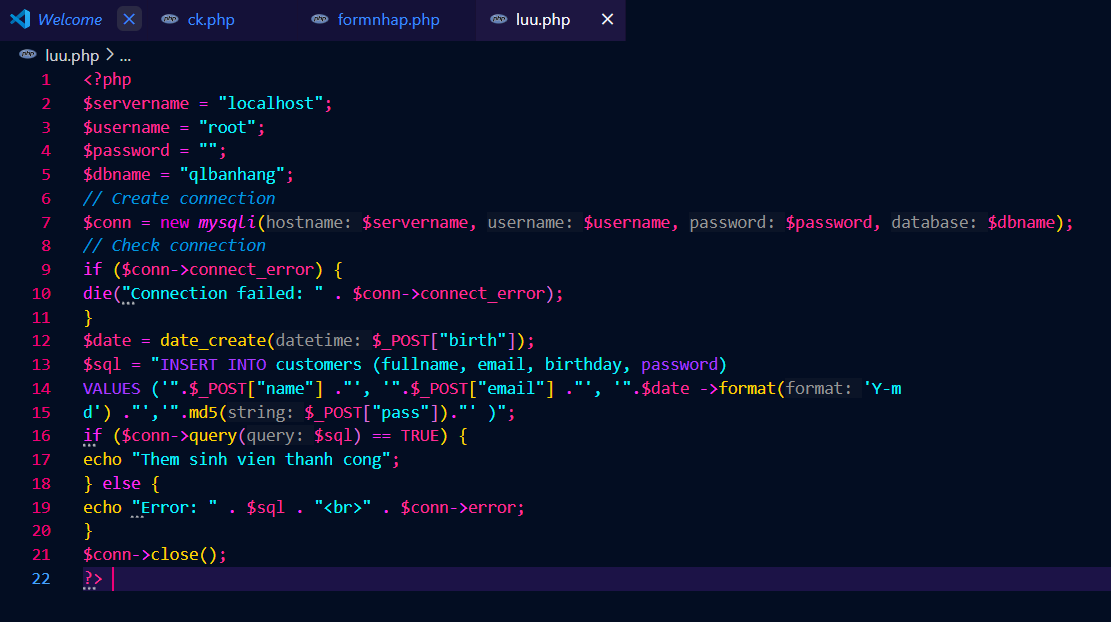
Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa  
Tạo csdl qlbanhang, với bảng customers đầu tiên như sau:

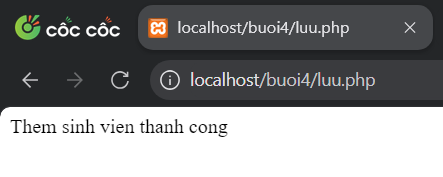
Tạo formnhap.php:



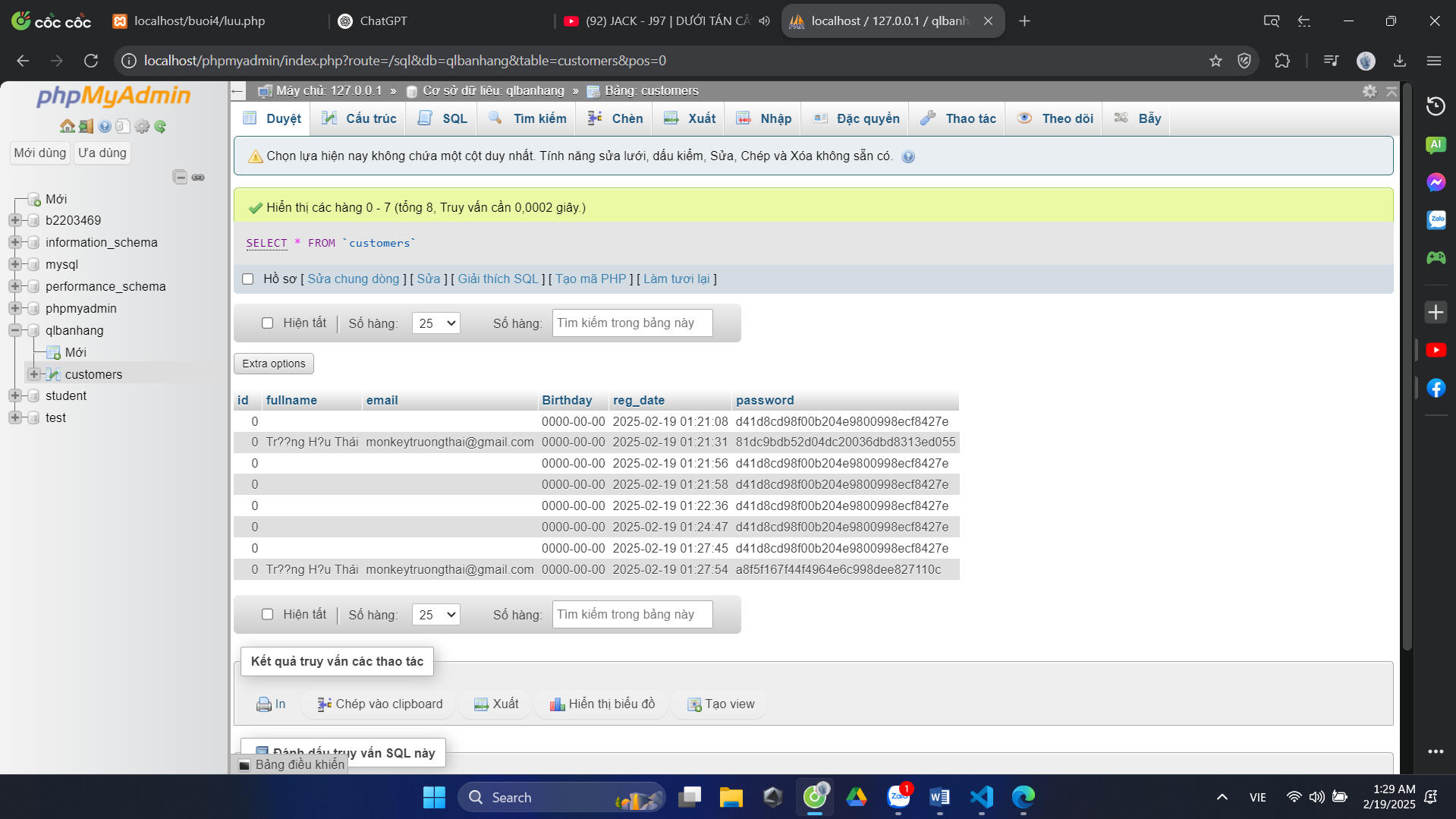
Và luu.php, chú ý lúc này chúng ta sẽ mã hóa mật khẩu với giải thuật md5 để bảo mật mật  
khẩu:

Luu.php

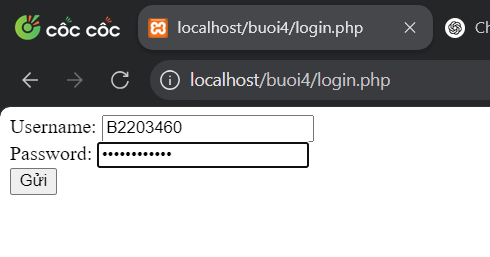
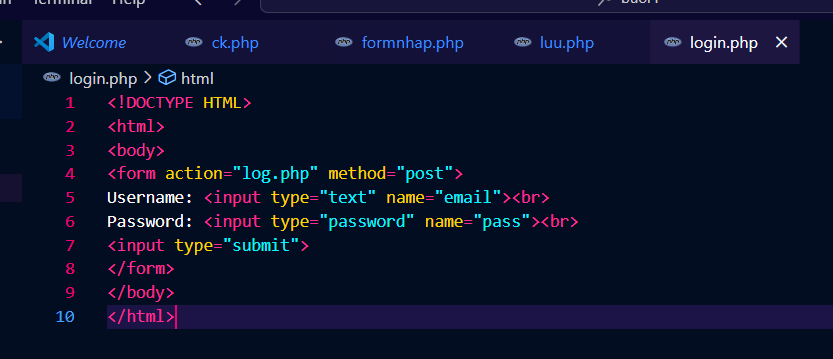




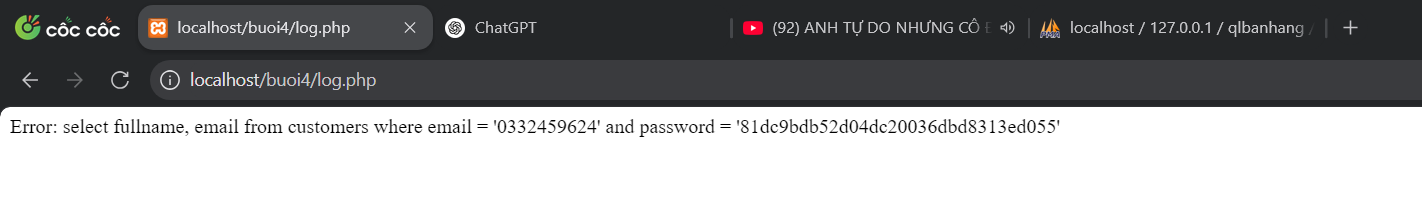
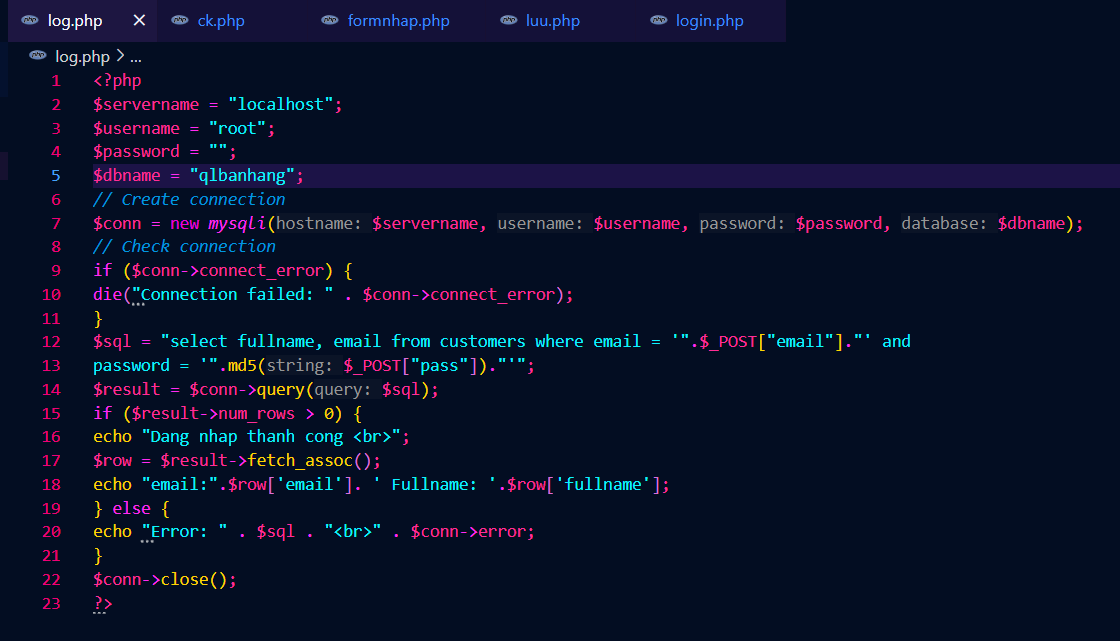
Sau khi chúng ta nhập xong mật khẩu với 12345 nhưng khi quan sát trong CSDL ta sẽ thấy mật khẩu đã được mã hóa ra chuỗi 827ccb0eea8a706c4c34a16891f84e7b:



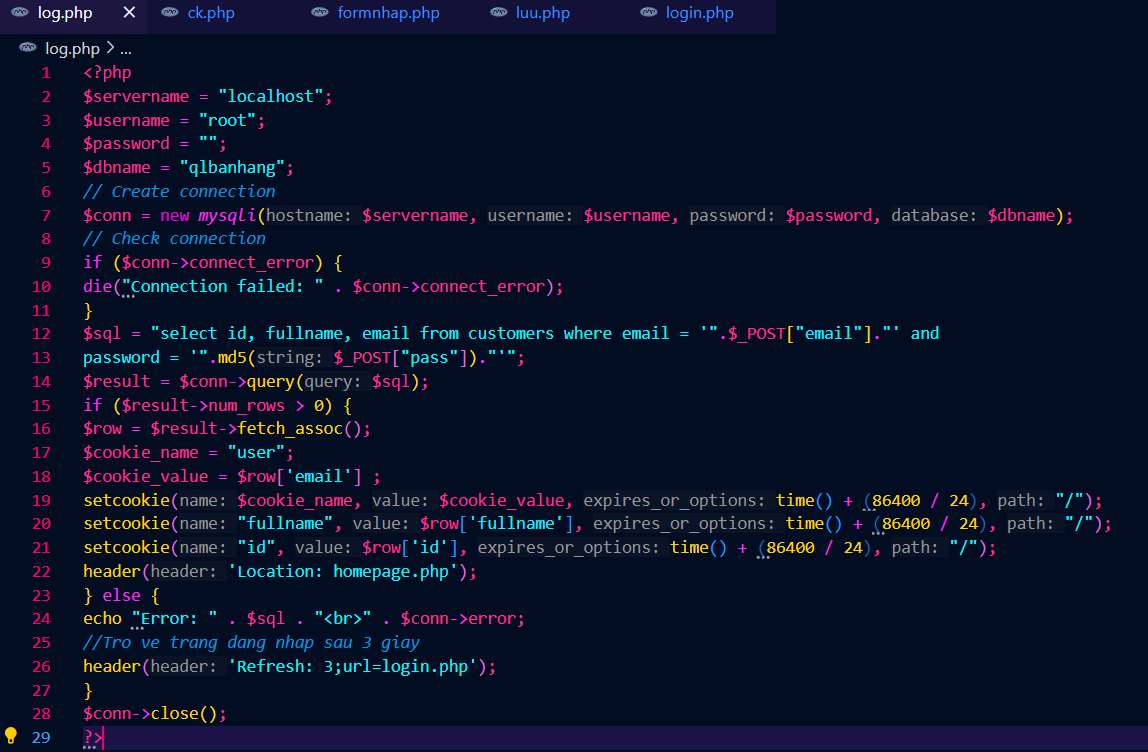
Tạo form đăng nhập  
Sau khi tạo người dùng xong, ta có thể chạy form đăng nhập với username là email và  
password lấy từ CSDL  
Login.php

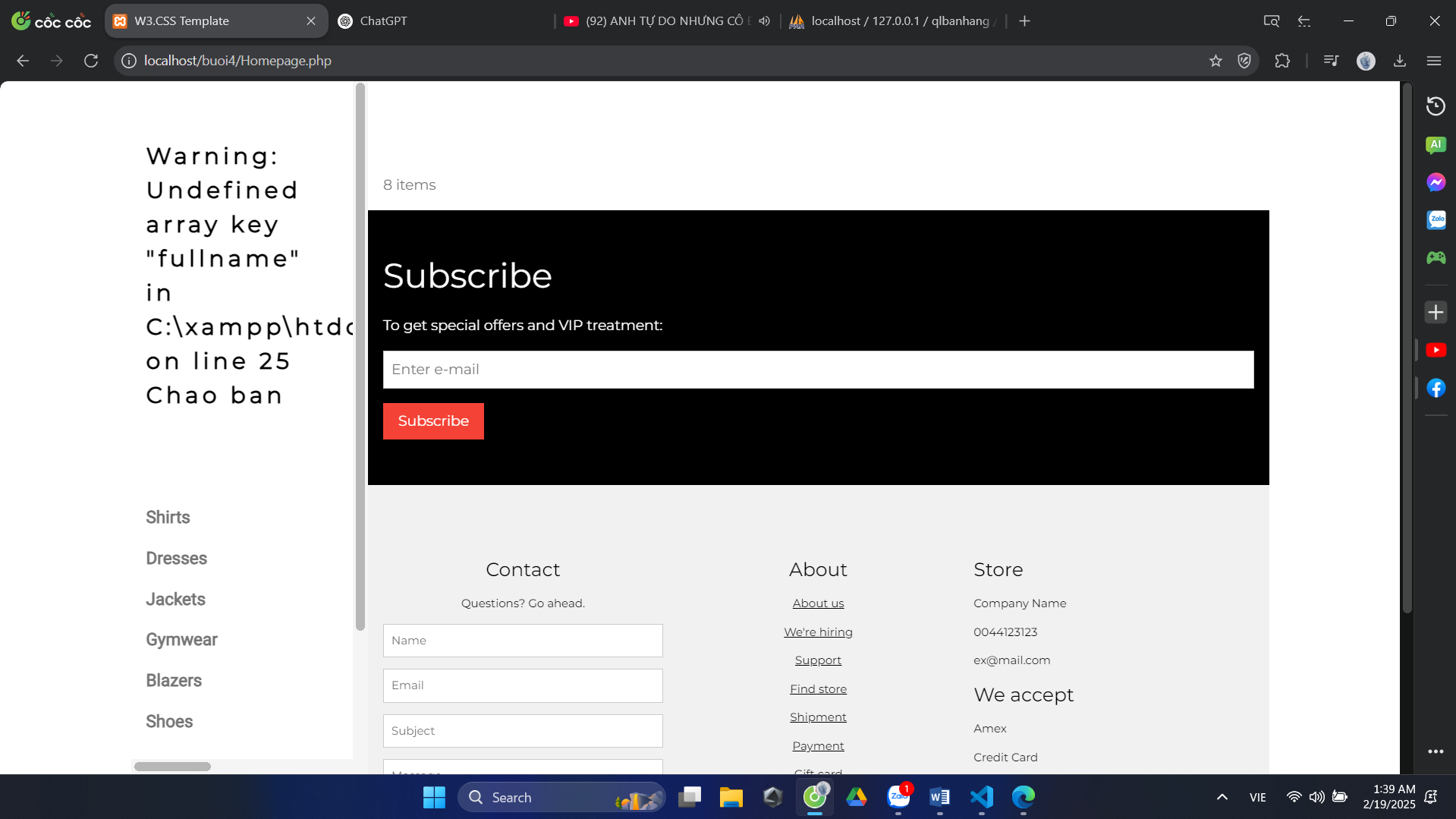


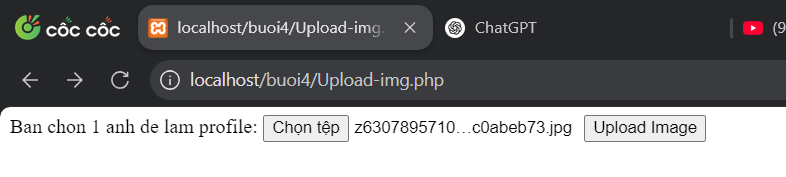
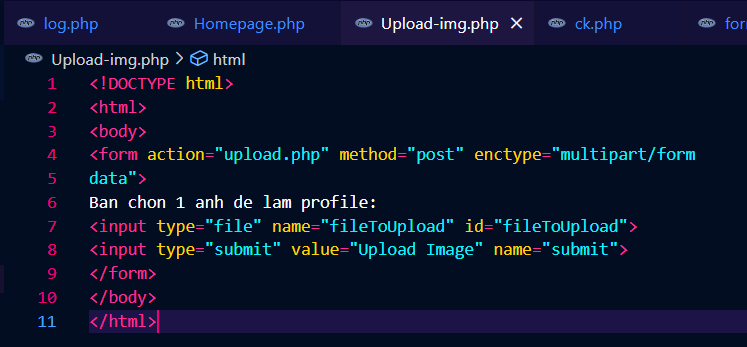
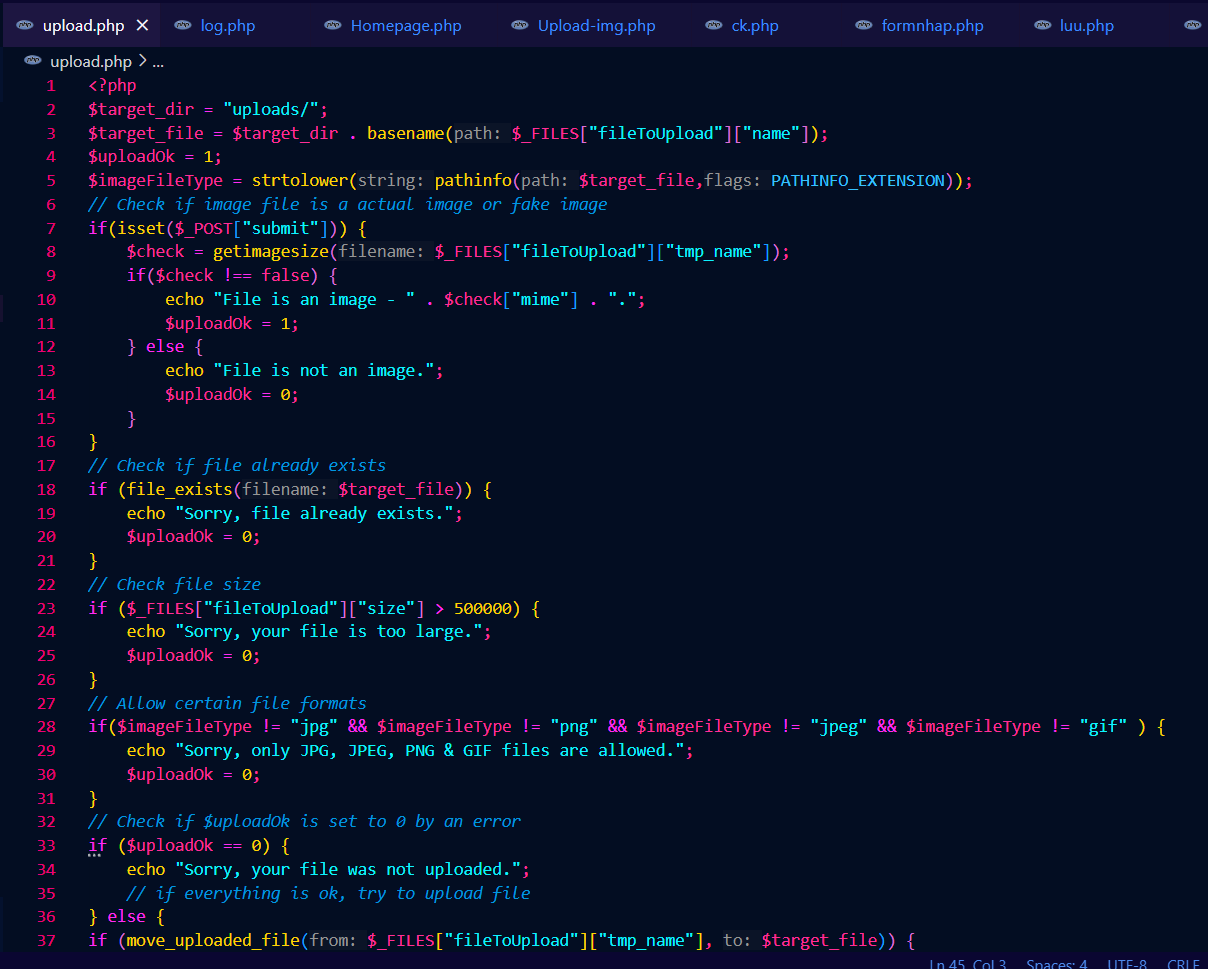
Log.php

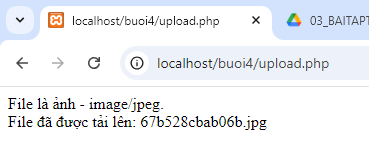


Dùng Cookies lưu lại tên đăng nhập, email và điều hướng  
Chỉnh sửa lại log.php với code sau để khi đăng nhập thành công chuyển sang 1 trang chủ, còn  
nếu không đúng thì trở lại trang đăng nhập  
Log.php (modified)

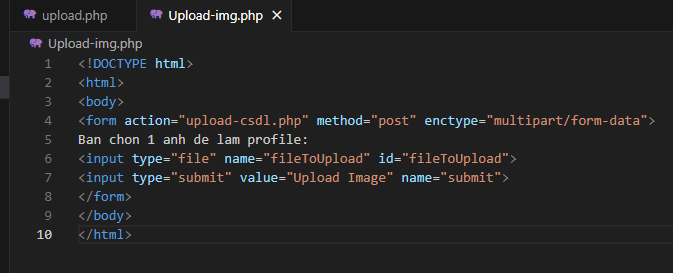


Tạo 1 trang chủ để khi đăng nhập thành công chuyển qua trang chủ này:  
Homepage.php

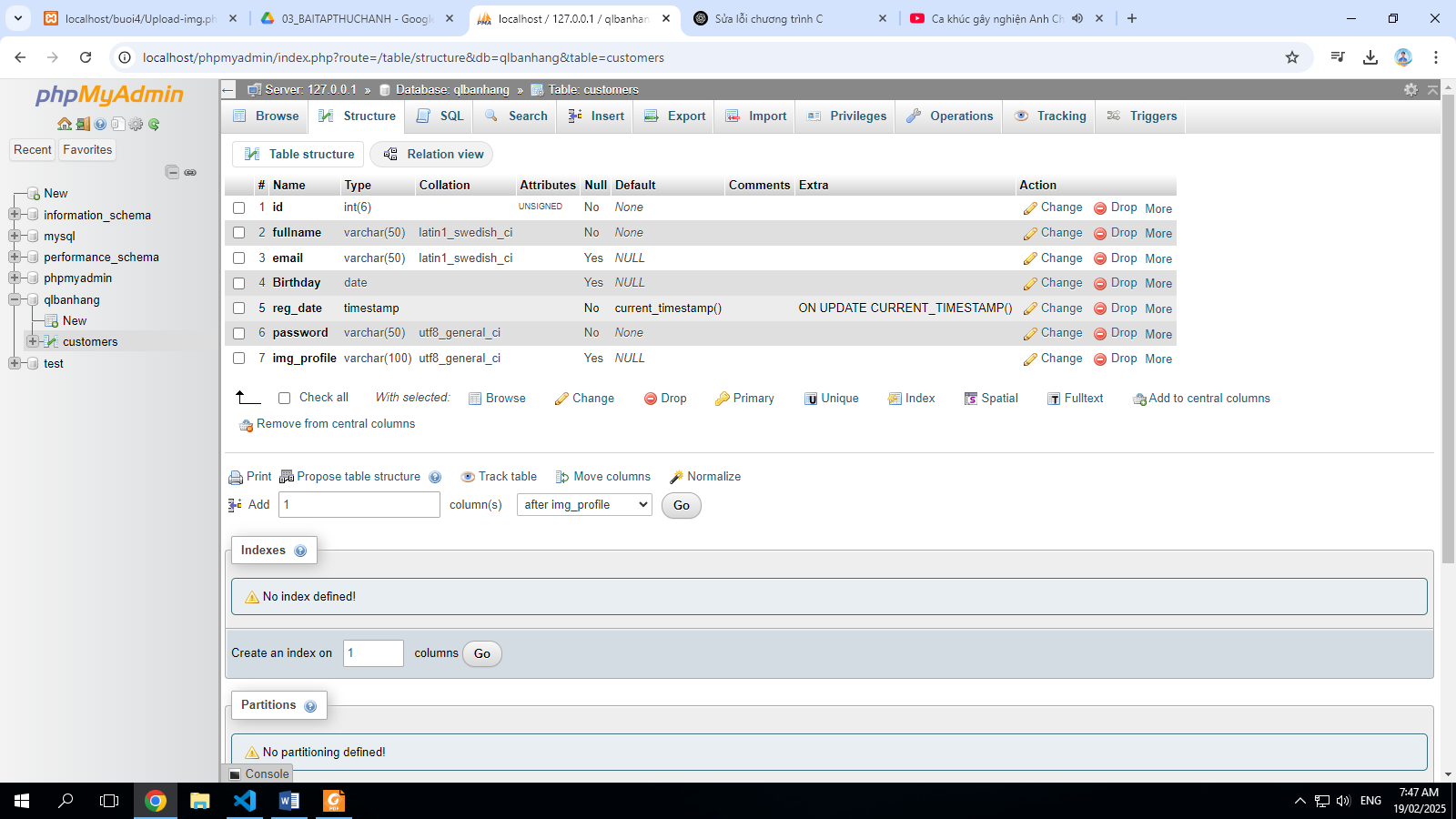
Upload tập tin  
Upload file, chúng ta tạo form giao diện để upload file ảnh upload-img.php  
Upload-img.phpTập tin upload.php xử lý hành động  
upload.php



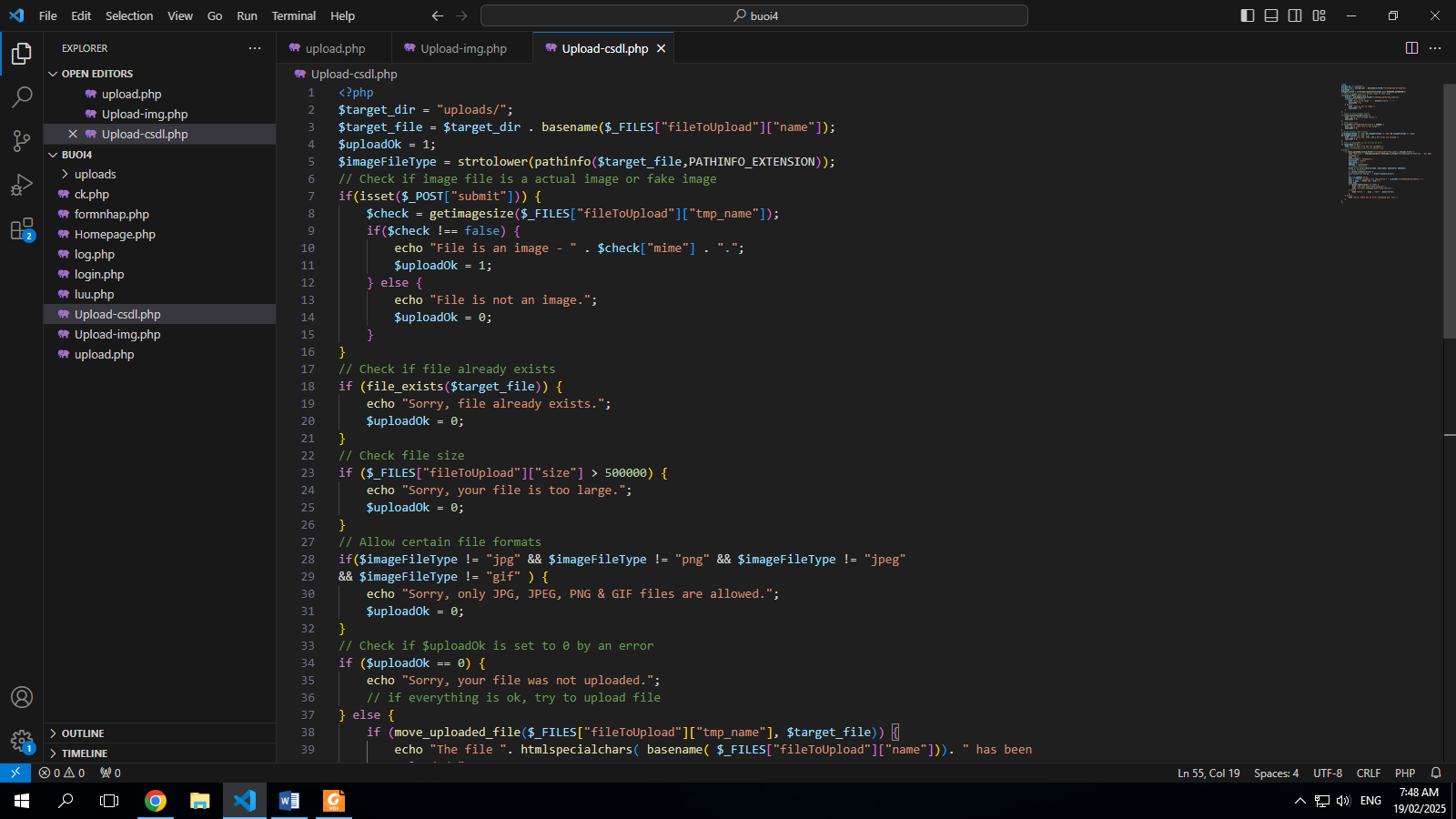
Upload ảnh và lưu dữ liệu về ảnh trong CSDL  
Trong phần này chúng ta minh họa tải tập tin ảnh làm ảnh profile  
của người dùng  
Upload-img.php (modified)

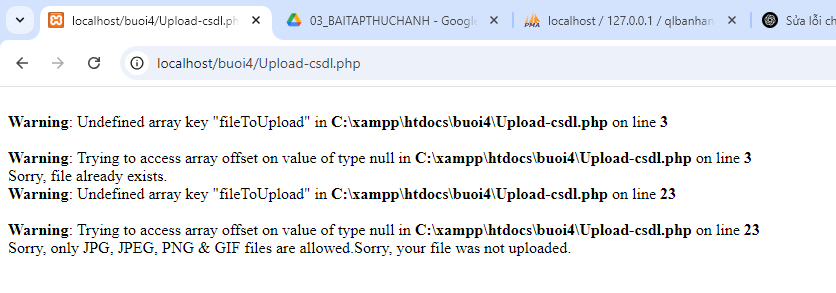


Thêm cột để lưu đường dẫn ảnh:

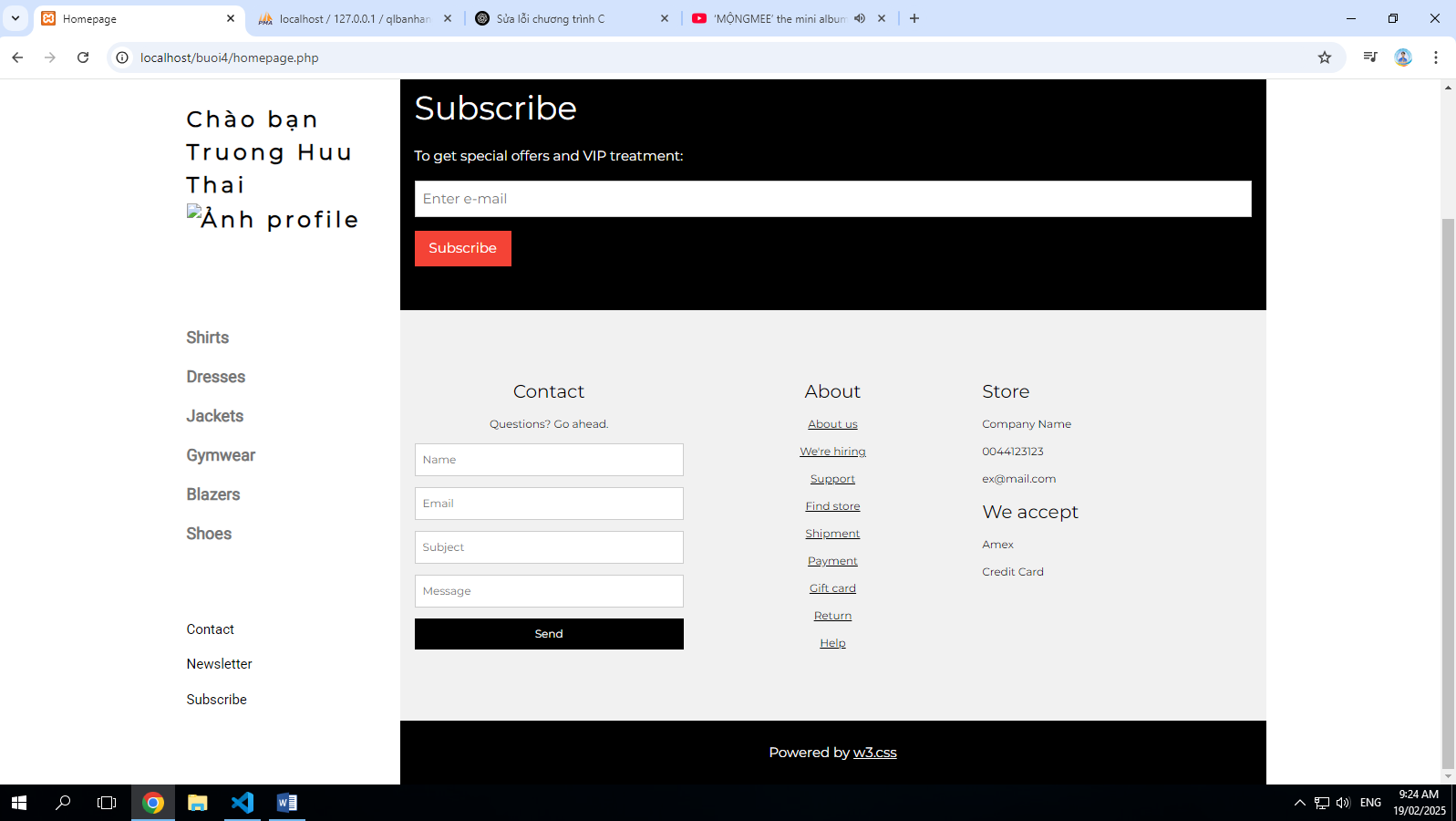


Upload-csdl.php

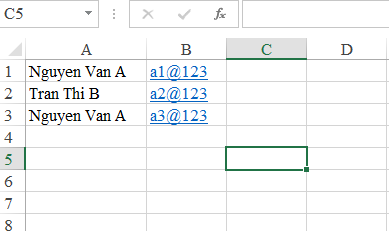




Homepage.php (modified với ảnh profile)



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.



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 [[1]](#footnote-0). 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 trong những kỹ thuật hack web phổ biến nhất.

SQL Injection là việc đặt mã độc vào câu lệnh SQL thông qua đầu vào trang web.

1. Dựa vào link[[2]](#footnote-1), 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 nhận dạng người dùng. Cookie là một tệp nhỏ mà máy chủ nhúng vào máy tính của người dùng. Mỗi lần máy tính đó yêu cầu một trang bằng trình duyệt, nó cũng sẽ gửi cookie. Với PHP, bạn có thể vừa tạo vừa truy xuất các giá trị cookie
* Hàm setcookie() được sử dụng để tạo và gửi một cookie từ máy chủ đến trình duyệt. Cú pháp của hàm như sau:

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

* Giải thích các tham số:

1. name:
   * Tên của cookie. Đây là chỉ số dùng để lưu và truy xuất giá trị của cookie (ví dụ: "user").
2. value:
   * Giá trị của cookie. Thông tin bạn muốn lưu, thường được mã hóa (ví dụ: "John Doe").
   * Lưu ý: Giá trị này có thể là một chuỗi văn bản.
3. expire:
   * Thời gian hết hạn của cookie, tính theo thời gian Unix (số giây kể từ thời điểm 1/1/1970).
   * Ví dụ: time() + 3600 sẽ làm cookie hết hạn sau 1 giờ.
   * Nếu không đặt hoặc đặt bằng 0, cookie sẽ chỉ tồn tại trong phiên làm việc của trình duyệt (khi trình duyệt đóng, cookie sẽ mất).
4. path:
   * Đường dẫn trên máy chủ mà cookie có hiệu lực.
   * Ví dụ: "/" có nghĩa là cookie có hiệu lực trên toàn bộ website, còn "/subfolder/" chỉ có hiệu lực trên thư mục đó.
5. domain:
   * Tên miền mà cookie sẽ có hiệu lực.
   * Ví dụ: .example.com có nghĩa cookie sẽ có hiệu lực cho tất cả các subdomain của example.com.
6. secure:
   * Nếu được đặt là true, cookie chỉ được truyền qua kết nối HTTPS an toàn.
   * Nếu là false hoặc không được thiết lập, cookie có thể được truyền qua kết nối HTTP không an toàn.
7. httponly:
   * Nếu được đặt là true, cookie sẽ không thể truy cập thông qua JavaScript (hỗ trợ bảo vệ chống XSS).
   * Nếu là false, cookie có thể được truy cập từ JavaScript.

* Cách lưu, lấy, và xóa giá trị trong cookie

1. Lưu (Tạo) Cookie:

Sử dụng hàm setcookie(). Ví dụ:

setcookie("user", "John Doe", time() + (86400 \* 30), "/");

1. Lấy (Đọc) Cookie:

Sử dụng biến siêu toàn cục $\_COOKIE. Ví dụ:

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

echo "Cookie 'user' có giá trị: " . $\_COOKIE["user"];

} else {

echo "Cookie 'user' chưa được thiết lập.";

}

1. Xóa Cookie:

Để xóa cookie, bạn đặt thời gian hết hạn của nó về quá khứ. Ví dụ:

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

1. Dựa vào link[[3]](#footnote-2), bạn hãy cho biết Session dùng để làm gì. Cách lưu, lấy, xóa giá trị trong Session.

* Session trong PHP là một cơ chế cho phép lưu trữ thông tin của người dùng trên máy chủ trong quá trình họ duyệt web.
* Nó giúp duy trì trạng thái và dữ liệu giữa các trang (ví dụ: thông tin đăng nhập, giỏ hàng, tùy chọn người dùng, v.v.)
* Dữ liệu session được lưu trên server, và phiên làm việc của người dùng được nhận diện thông qua một cookie (thường là PHPSESSID) được gửi từ trình duyệt.
* Cách sử dụng Session trong PHP

1. Khởi tạo Session: Để sử dụng session, bạn cần gọi hàm session\_start() ở đầu mỗi file PHP, trước bất kỳ đầu ra nào.

<?php

session\_start();

?>

1. Lưu giá trị vào Session

* Bạn có thể lưu trữ dữ liệu vào mảng siêu toàn cục $\_SESSION.
* Ví dụ: Lưu thông tin tên người dùng:

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

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

1. Lấy giá trị từ Session: Để truy xuất dữ liệu đã lưu trong session, bạn đọc các giá trị từ $\_SESSION.

<?php

session\_start();

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

echo "Chào mừng, " . $\_SESSION['username'];

} else {

echo "Bạn chưa đăng nhập.";

}

?>

1. Xóa giá trị khỏi Session: Để xóa một biến cụ thể trong session, sử dụng hàm unset().

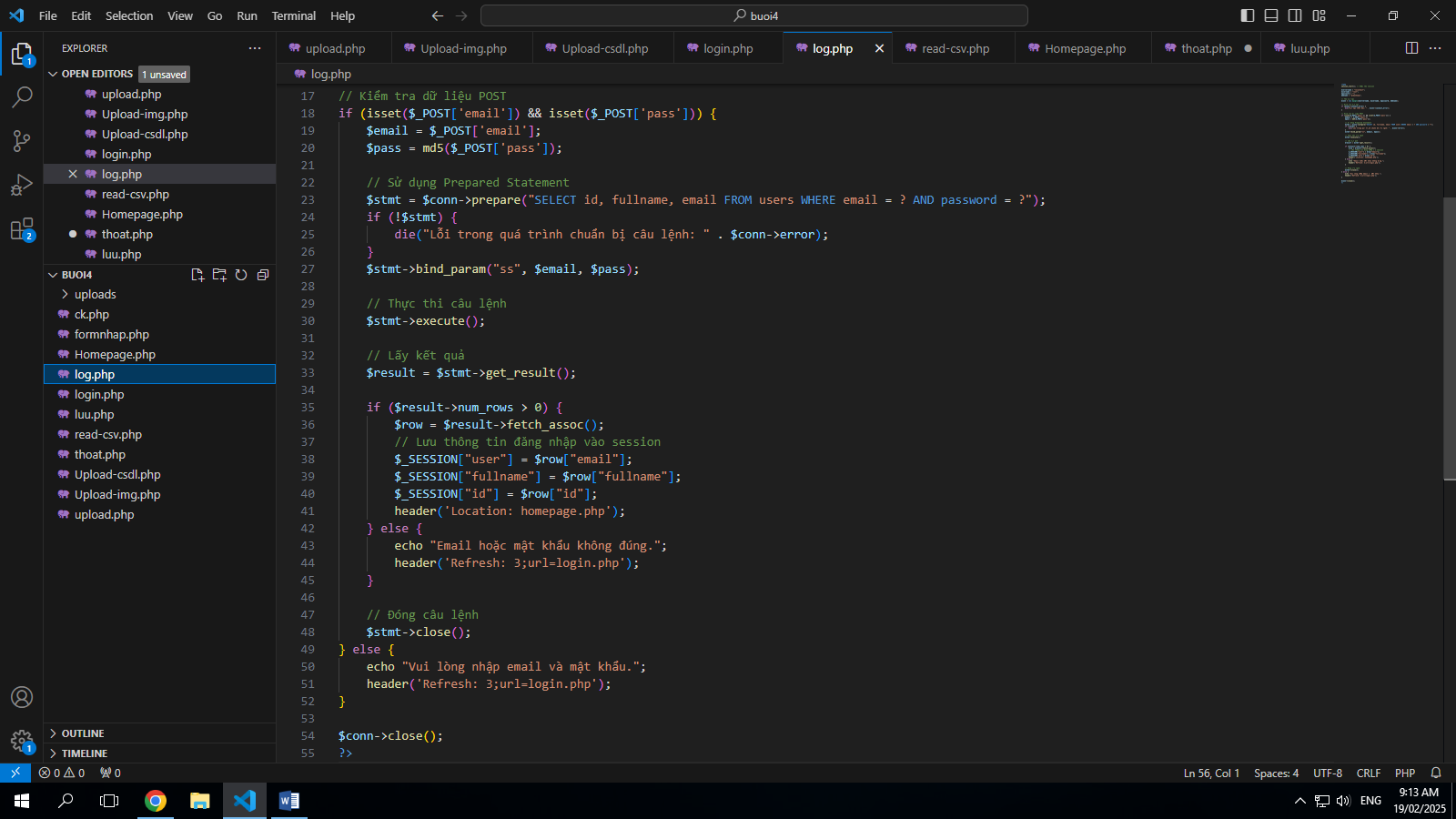
unset($\_SESSION['username']);

1. Bạn hãy so sánh Cookie và Session.

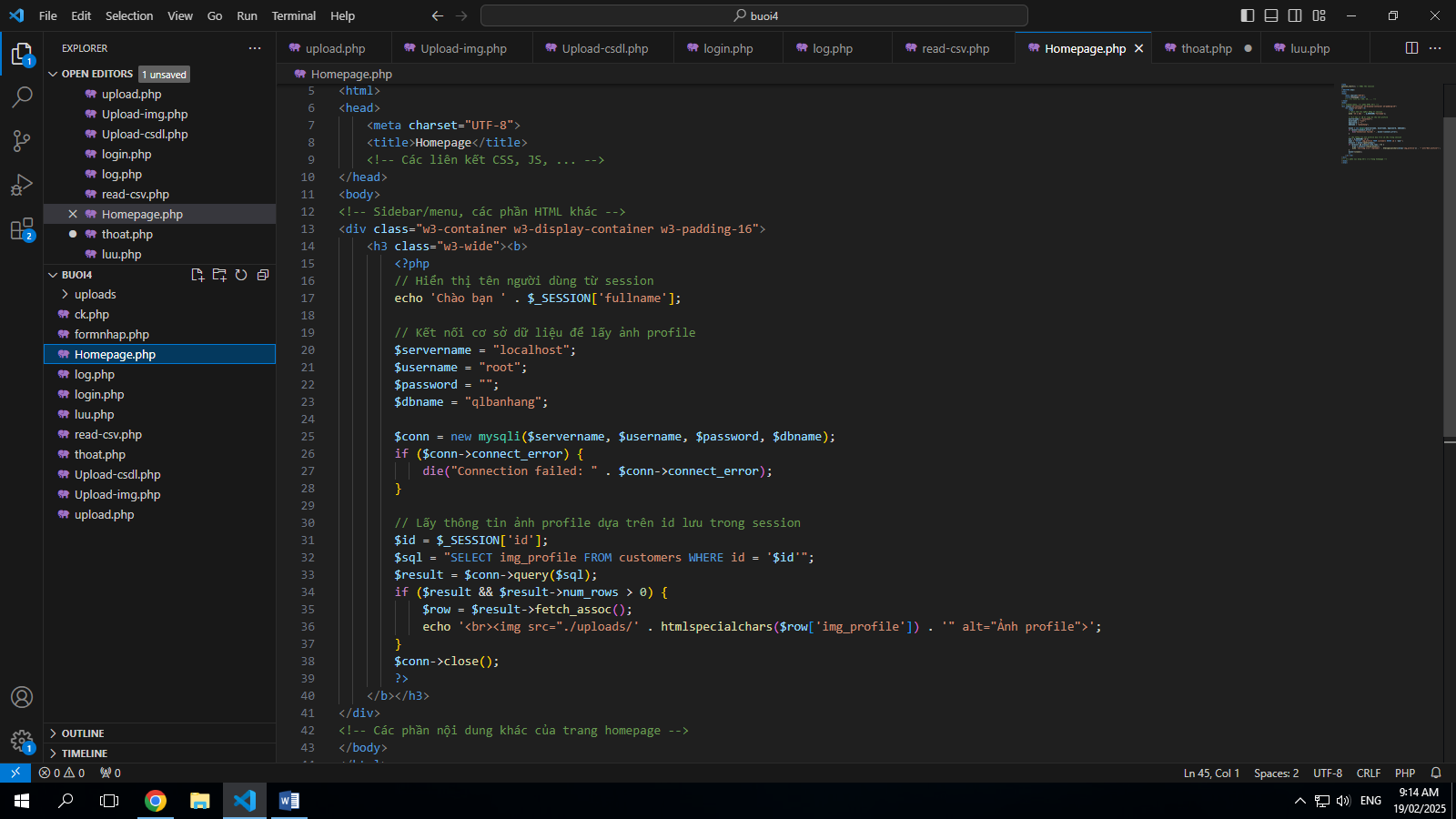
|  | Cookie | Session |
| --- | --- | --- |
| Nơi lưu trữ | Được lưu trữ trên máy tính (trình duyệt) của người dùng. | Được lưu trữ trên máy chủ, dữ liệu của người dùng chỉ tồn tại tạm thời trên server. |
| Bảo mật | Dễ bị sửa đổi hoặc đánh cắp nếu không được bảo vệ (ví dụ: thông qua HTTPS, httponly). | An toàn hơn vì dữ liệu được lưu trên server, chỉ giữ một mã nhận dạng (session ID) trên trình duyệt. |
| Dung lương lưu trữ | Có giới hạn kích thước (thường khoảng 4KB). | Có thể lưu trữ nhiều dữ liệu hơn, phụ thuộc vào cấu hình của server. |
| Thời gian sống | Có thể được đặt thời gian hết hạn cụ thể, tồn tại lâu hơn (ngay cả sau khi trình duyệt đóng) nếu được cấu hình. | Thường tồn tại trong suốt phiên làm việc của trình duyệt và bị hủy khi trình duyệt đóng (trừ khi có cấu hình đặc biệt hoặc session lưu trên server ngoài). |
| Hiệu suất và giao tiếp mạng | Dữ liệu cookie được gửi kèm với mỗi yêu cầu HTTP đến server, có thể làm tăng lưu lượng mạng. | Chỉ gửi session ID qua cookie (hoặc URL), dữ liệu session thực sự không được gửi đi cùng mỗi yêu cầu. |
| Ứng dụng | Thường dùng để lưu trữ thông tin ít nhạy cảm, như tùy chọn người dùng, lưu đăng nhập (remember me), hoặc theo dõi hành vi người dùng. | Thường dùng để lưu trữ thông tin nhạy cảm và tạm thời của người dùng như trạng thái đăng nhập, giỏ hàng, hoặc các dữ liệu cần bảo mật cao. |

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.

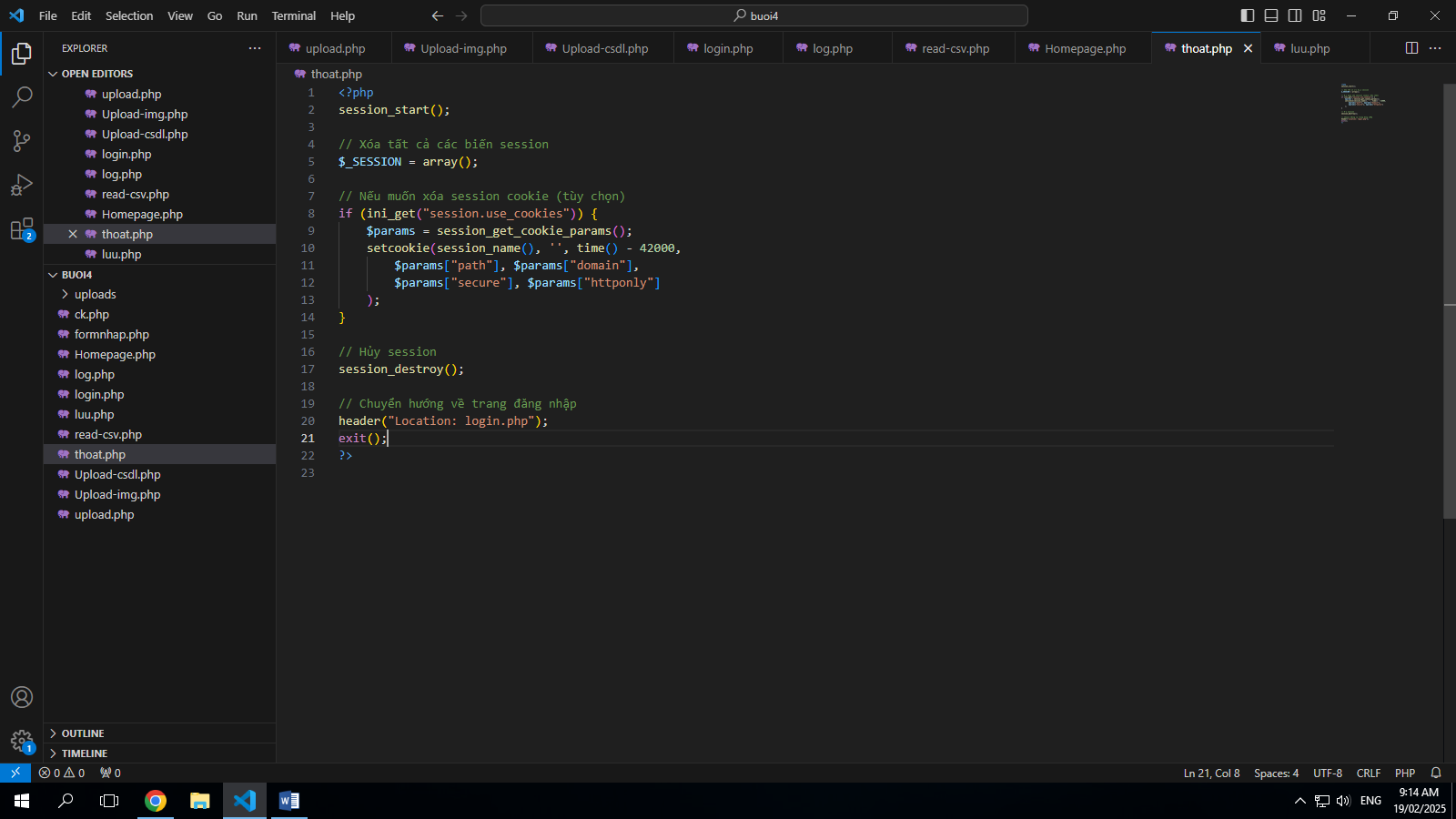
Log.php



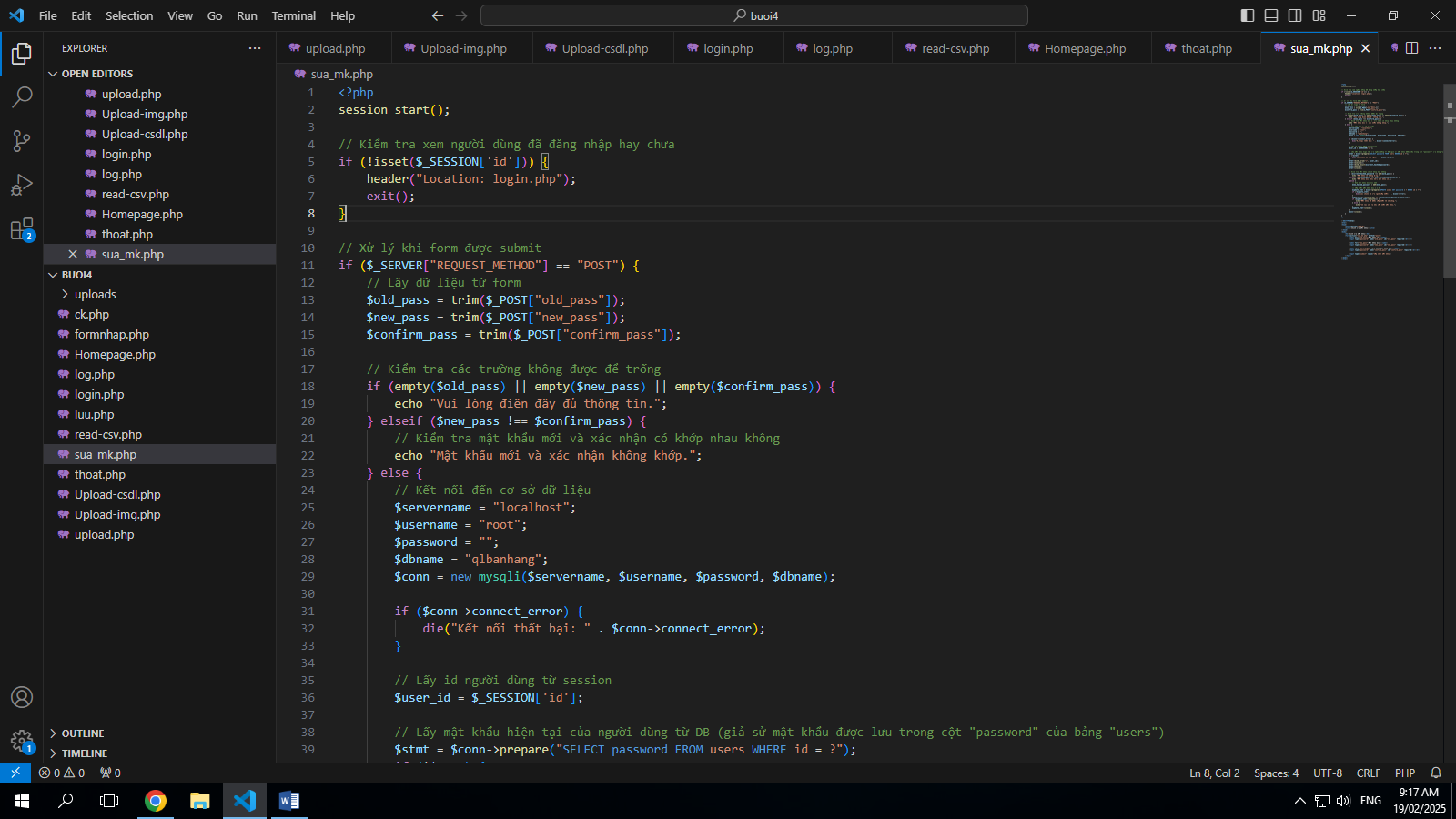
Homepage.php

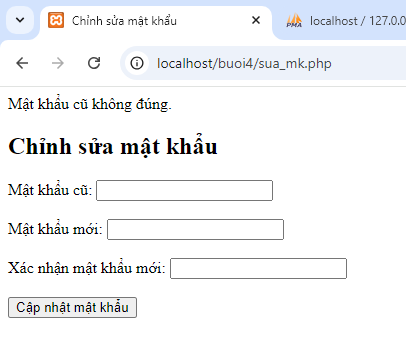


Thoat.php



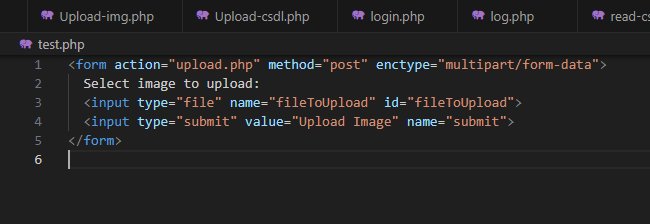
1. Bạn hãy tạo form sua\_mk.php cho phép người dùng chỉnh sửa mật khẩu sau khi đăng nhập. Yêu cầu gồm 3 input: ô để nhập mật khẩu cũ, 1 ô để nhập mật khẩu mới, 1 ô cho phép nhập mật lại mật khẩu mới. Khi nhập xong cần đảm bảo: mật khẩu cũ là khớp với CSDL đang lưu, 2 ô nhập mật khẩu mới phải khớp với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



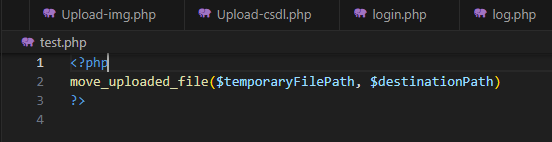


1. Bạn đọc trong[[4]](#footnote-3) để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.

* Cơ chế upload file trong PHP:
  + HTML Form:
    - Để upload file, bạn cần một form HTML với thuộc tính method="post" và đặc biệt cần đặt enctype="multipart/form-data". Thuộc tính này cho phép form gửi các dữ liệu nhị phân (chẳng hạn file) tới server.



* + **Quá trình upload:**
    - Khi người dùng gửi form, file được upload sẽ được lưu tạm thời trên máy chủ (thông qua thư mục tạm thời được cấu hình trong PHP, thường là tmp).
    - PHP thu thập thông tin file upload trong mảng siêu toàn cục $\_FILES, chứa các thông tin như tên file, kiểu file, kích thước file, đường dẫn file tạm thời và mã lỗi (nếu có).
* **Các hàm và thủ tục chính để upload file:**
  + **$\_FILES:**
    - Là một mảng chứa các thông tin về file được upload. Các key thường dùng:
      * $\_FILES["fileToUpload"]["name"]: Tên gốc của file.
      * $\_FILES["fileToUpload"]["type"]: Kiểu file (ví dụ: "image/jpeg").
      * $\_FILES["fileToUpload"]["size"]: Kích thước file tính bằng byte.
      * $\_FILES["fileToUpload"]["tmp\_name"]: Đường dẫn tạm thời nơi file được lưu trên server.
      * $\_FILES["fileToUpload"]["error"]: Mã lỗi upload (nếu có lỗi, giá trị khác 0).
    - **move\_uploaded\_file():**
      * Hàm này có nhiệm vụ di chuyển file từ vị trí tạm thời (tmp\_name) đến một vị trí lưu trữ cố định mà bạn mong muốn.



* + - * Nếu việc di chuyển thành công, hàm trả về true; ngược lại, trả về false.
      * Đây là bước bắt buộc để "lưu" file upload vào thư mục bạn mong muốn trên server.
    - **getimagesize():**
      * Nếu bạn upload file ảnh, hàm này có thể được dùng để kiểm tra xem file đó có phải là ảnh thật hay không. Hàm sẽ trả về thông tin kích thước và kiểu ảnh nếu file hợp lệ, ngược lại trả về false.
  + **Xử lý lỗi và kiểm tra thông tin:**
    - Bạn nên kiểm tra các điều kiện sau:
      * **Mã lỗi của file upload:** Sử dụng $\_FILES["fileToUpload"]["error"] để kiểm tra nếu có lỗi.
      * **Kích thước file:** Để đảm bảo file không vượt quá giới hạn cho phép (thiết lập trong php.ini như upload\_max\_filesize).
      * **Kiểu file:** Để cho phép chỉ các định dạng file nhất định (ví dụ chỉ cho phép JPG, PNG, GIF).

1. Bạn hãy đọc[[5]](#footnote-4), 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()
* Mục đích:  
  Hàm file() dùng để đọc toàn bộ nội dung của một file và trả về dưới dạng một mảng, trong đó mỗi phần tử của mảng tương ứng với một dòng trong file đó. Nếu đọc thành công, hàm trả về một mảng; nếu có lỗi (ví dụ file không tồn tại), hàm trả về false.
* Ứng dụng:  
  Hàm này hữu ích khi bạn cần thao tác trên từng dòng của file, ví dụ như khi xử lý dữ liệu từ file CSV (mỗi dòng chứa một hàng dữ liệu) hoặc file văn bản. Từ kết quả của hàm file(), bạn có thể dùng các hàm khác (như explode(), preg\_split() hoặc fgetcsv()) để phân tích dữ liệu chi tiết.

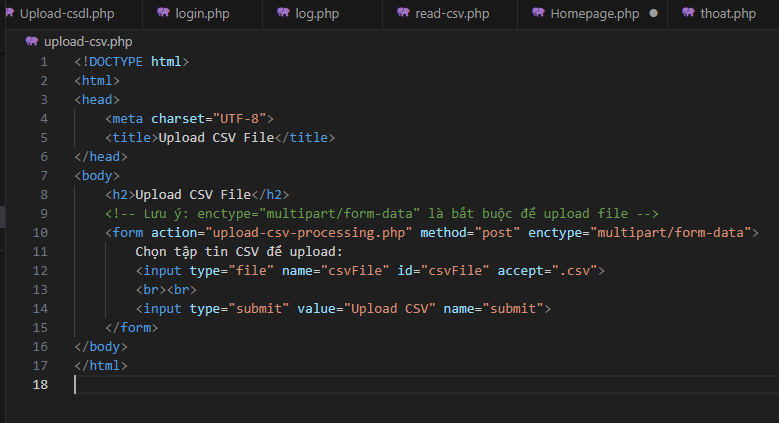
Cú pháp:

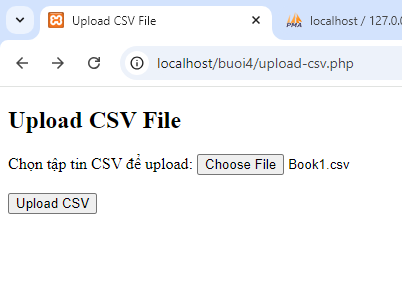
array file ( string $filename [, int $flags = 0 [, resource $context ]] )

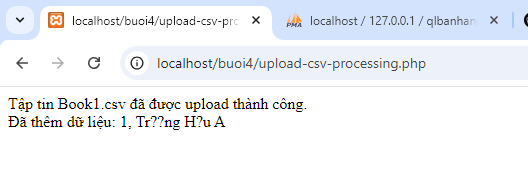
* $filename:
* Là đường dẫn hoặc tên file bạn muốn đọc.
* Ví dụ: "data.csv" hoặc "uploads/info.txt".
* $flags (tùy chọn):
* Là một số giá trị bit để chỉ định cách thức đọc file. Một số flag thông dụng:
  + FILE\_IGNORE\_NEW\_LINES: Khi đặt flag này, các ký tự xuống dòng (newline) sẽ không được thêm vào cuối mỗi phần tử của mảng.
  + FILE\_SKIP\_EMPTY\_LINES: Bỏ qua các dòng trống trong file.
* Bạn có thể kết hợp các flag bằng toán tử OR (|).
* Ví dụ: FILE\_IGNORE\_NEW\_LINES | FILE\_SKIP\_EMPTY\_LINES.
* **$context** (tùy chọn):
* Là một resource tạo bởi hàm **stream\_context\_create()**. Tham số này cho phép bạn thiết lập các tùy chọn cho stream (ví dụ: timeout, proxy, v.v.) khi đọc file.
* Nếu không cần thiết, bạn có thể bỏ qua tham số này.

1. 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: [[6]](#footnote-5),[[7]](#footnote-6). 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.

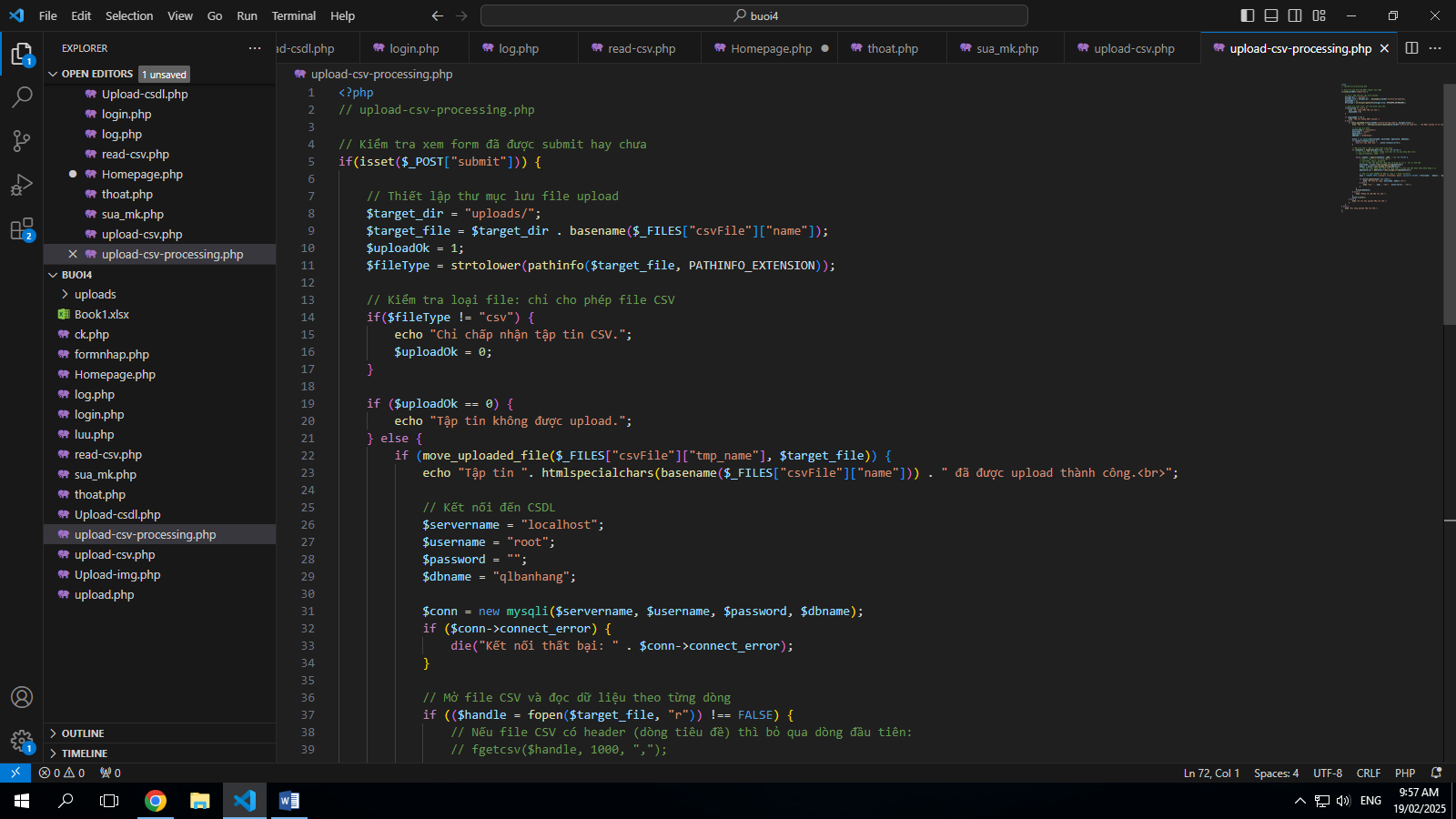
Upload-csv.php







upload-csv-processing.php



**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

1. <https://www.w3schools.com/sql/sql_injection.asp> [↑](#footnote-ref-0)
2. <https://www.w3schools.com/php/php_cookies.asp> [↑](#footnote-ref-1)
3. <https://www.w3schools.com/php/php_sessions.asp> [↑](#footnote-ref-2)
4. <https://www.w3schools.com/php/php_file_upload.asp> [↑](#footnote-ref-3)
5. <https://www.w3schools.com/php/func_filesystem_file.asp> [↑](#footnote-ref-4)
6. <https://www.php.net/manual/en/features.file-upload.post-method.php>, [↑](#footnote-ref-5)
7. https://stackoverflow.com/questions/6654351/check-file-uploaded-is-in-csv-format [↑](#footnote-ref-6)