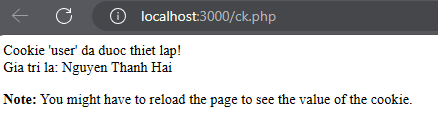
Bài Tập Thực Hành 04: Lập trình PHP nâng cao

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

**Trả lời:**

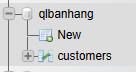
Tạo Cookies

+ ck.php

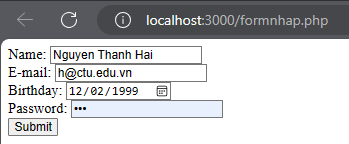


Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa

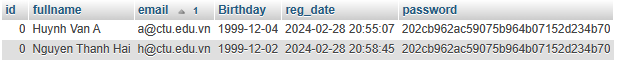
+Tạo csdl qlbanhang, với bảng customers đầu tiên



+Tạo formnhap.php



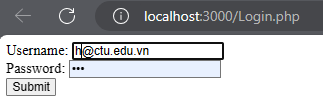
+luu.php





Tạo form đăng nhập

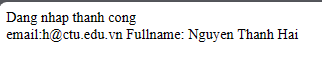
+Login.php



+Khi nhập sai sẽ nhận thông báo:



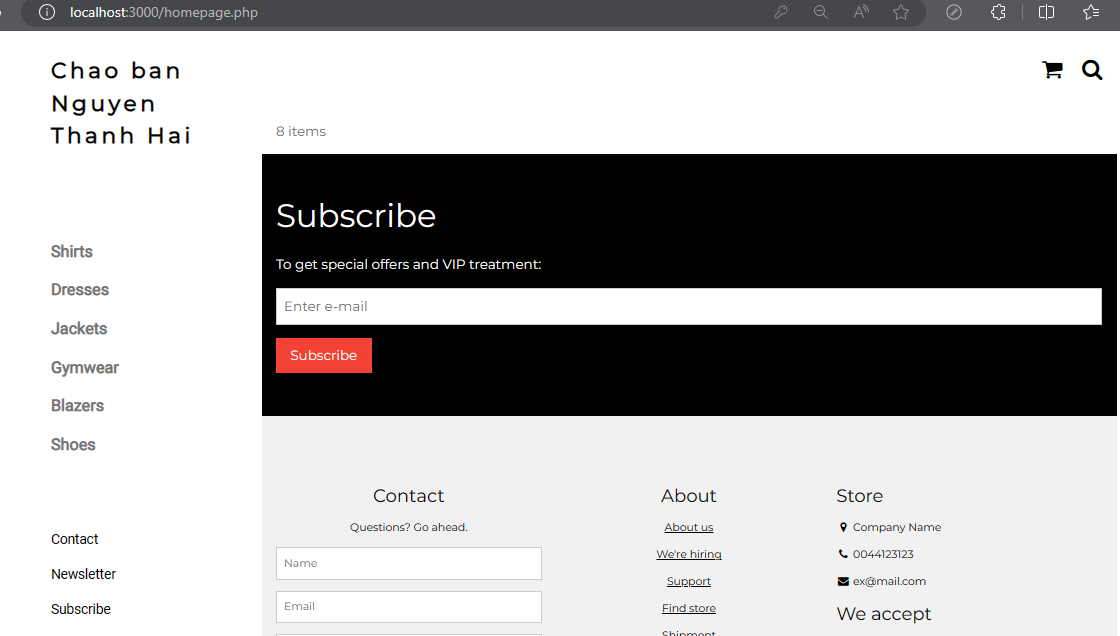
+Khi nhập đúng:



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

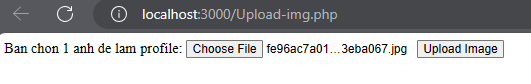
+Log.php (modified)

+Homepage.php



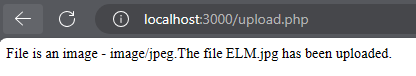
Upload tập tin

+Upload-img.php



Tập tin upload.php xử lý hành động

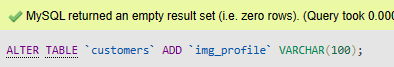
+upload.php



Upload ảnh và lưu dữ liệu về ảnh trong CSDL

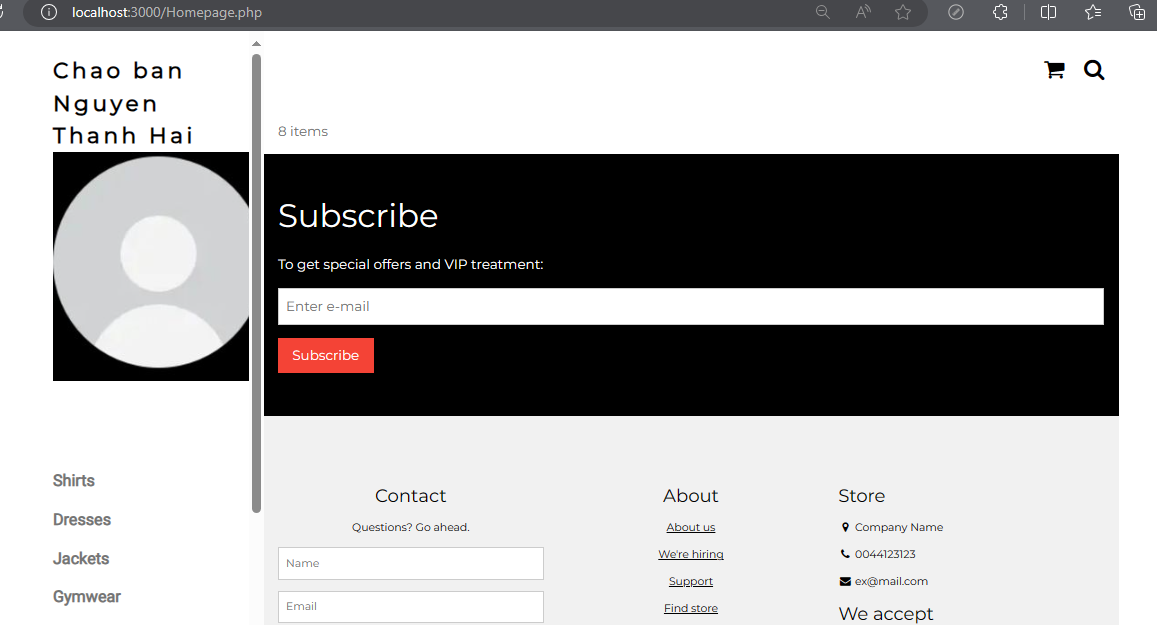
+Upload-img.php (modified)

Thêm cột để lưu đường dẫn ảnh:



+Upload-csdl.php

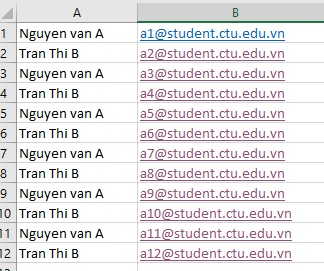
+Homepage.php (modified với ảnh profile)

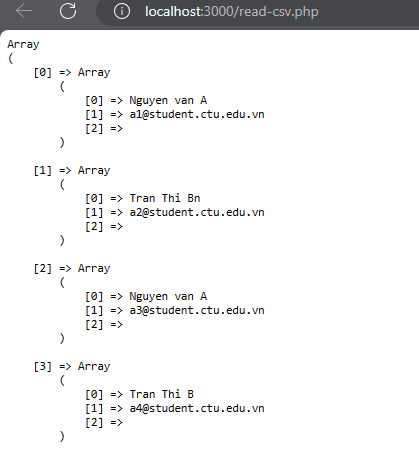


Upload file csv và đọc dữ liệu vào mảng

+read-csv.php

+ cus.csv





**2. 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 . Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.**

**Trả lời:**

SQL Injection là một kỹ thuật tấn công phổ biến được sử dụng bởi các hacker để xâm nhập vào hệ thống thông qua việc chèn các câu lệnh SQL độc hại vào các trường dữ liệu đầu vào của ứng dụng web hoặc các truy vấn SQL.

Khi một ứng dụng web không kiểm tra hoặc xử lý đúng đắn dữ liệu đầu vào từ người dùng, hacker có thể chèn các đoạn mã SQL độc hại vào các ô nhập liệu. Khi dữ liệu này được truyền vào cơ sở dữ liệu và không được lọc bớt, hệ thống sẽ thực thi câu lệnh SQL do hacker chèn vào, điều này có thể dẫn đến các hậu quả nguy hiểm như:

1. Lấy thông tin không được ủy quyền: Hacker có thể sử dụng SQL Injection để lấy thông tin nhạy cảm từ cơ sở dữ liệu như tên người dùng, mật khẩu, thông tin tài khoản ngân hàng, v.v.

2. Thay đổi dữ liệu: Hacker có thể thực hiện các câu lệnh UPDATE hoặc DELETE để thay đổi hoặc xóa dữ liệu trong cơ sở dữ liệu của bạn.

3. Chèn dữ liệu độc hại: Hacker có thể chèn mã độc hoặc liên kết độc hại vào cơ sở dữ liệu, khiến cho các người dùng khác truy cập vào trang web của bạn bị tấn công.

Để ngăn chặn SQL Injection, các lập trình viên cần thực hiện các biện pháp bảo vệ như sử dụng câu lệnh thực thi tham số hóa (prepared statements), sử dụng ORM (Object-Relational Mapping) để tránh sử dụng trực tiếp câu lệnh SQL, kiểm tra và xử lý đầu vào từ người dùng trước khi truy vấn vào cơ sở dữ liệu.

**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.**

**Trả lời:**

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.

Trong PHP, hàm `setcookie()` được sử dụng để thiết lập cookie trong trình duyệt của người dùng. Cú pháp của hàm này là:

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

Dưới đây là ý nghĩa của từng tham số:

1. name: Tên của cookie. Đây là tham số bắt buộc. Tên này sẽ được sử dụng để truy cập giá trị của cookie sau này.

2. value: Giá trị của cookie. Đây cũng là một tham số bắt buộc. Điều này có thể là một chuỗi chứa thông tin bạn muốn lưu trữ.

3. expire: Thời gian hết hạn của cookie, được đo bằng số giây kể từ thời điểm hiện tại. Đây là một số nguyên tùy chọn. Nếu không cung cấp, cookie sẽ hết hạn khi trình duyệt đóng.

4. path: Đường dẫn tương đối trên máy chủ để cookie có thể truy cập. Mặc định là `'/'`, có nghĩa là cookie có thể truy cập từ tất cả các đường dẫn trong trang web.

5. domain: Tên miền của cookie. Mặc định là `''`, nghĩa là chỉ có thể truy cập cookie từ tên miền hiện tại.

6. secure: Nếu được đặt thành `true`, cookie chỉ được gửi qua kết nối an toàn HTTPS. Mặc định là `false`.

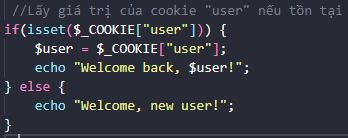
7. httponly: Nếu được đặt thành `true`, cookie chỉ có thể được truy cập thông qua HTTP (không thể sử dụng JavaScript để truy cập cookie). Điều này giúp ngăn chặn các cuộc tấn công Cross-Site Scripting (XSS). Mặc định là `false`.

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

+ Để lưu trữ giá trị vào cookie, ta sử dụng hàm **`setcookie()`**



+ Lấy giá trị từ cookie, ta sử dụng biến toàn cục **`$\_COOKIE`**



+ Để xóa cookie, hãy sử dụng **setcookie()**hàm có ngày hết hạn trong quá khứ:



**4. Dựa vào link3 , bạn hãy cho biết Session dùng để làm gì. Cách lưu, lấy, xóa giá trị trong Session.**

**Trả lời:**

Khi bạn làm việc với một ứng dụng, bạn mở nó, thực hiện một số thay đổi rồi đóng nó lại. Điều này giống như một Session. Máy tính biết bạn là ai. Nó biết khi nào bạn khởi động ứng dụng và khi nào bạn kết thúc. Nhưng trên internet có một vấn đề: máy chủ web không biết bạn là ai hoặc bạn làm gì, vì địa chỉ HTTP không duy trì trạng thái.

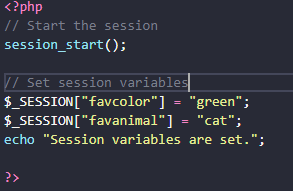
Biến Session giải quyết vấn đề này bằng cách lưu trữ thông tin người dùng sẽ được sử dụng trên nhiều trang (ví dụ: tên người dùng, màu yêu thích, v.v.). Theo mặc định, các biến Session kéo dài cho đến khi người dùng đóng trình duyệt.

Vì thế; Các biến Session chứa thông tin về một người dùng và có sẵn cho tất cả các trang trong một ứng dụng.

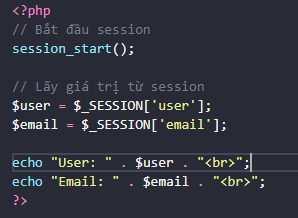
**Cách lưu, lấy, xóa giá trị trong Session**

+ Lưu giá trị vào session, ta sử dụng biến toàn cục `$\_SESSION`.

Ví dụ:

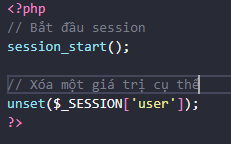


+ Lấy giá trị đã lưu trong session, ta truy cập vào biến `$\_SESSION`.

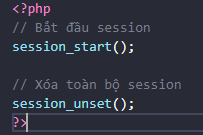


+ Xóa giá trị khỏi session, ta sử dụng `unset()` hoặc `session\_unset()`.

.Xóa một giá trị cụ thể



.Xóa toàn bộ session:



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

**Trả lời:**

Cookie và session là cách để lưu trữ thông tin trên máy chủ và truy cập thông tin đó khi người dùng truy cập trang web. Dưới đây là một so sánh giữa Cookie và session:

**Cookie:**

1. Lưu trữ trên máy khách (client-side):

- Cookie được lưu trữ trên máy tính của người dùng trong trình duyệt web của họ.

2. Thời gian sống:

- Có thể được thiết lập với thời gian sống cụ thể (ví dụ: 1 ngày, 1 tuần, 1 tháng).

- Có thể là "session cookie" tồn tại chỉ trong phiên làm việc của người dùng.

3. Dữ liệu lưu trữ:

- Thông tin được lưu trữ dưới dạng cặp key-value.

- Dữ liệu được lưu trữ trong cookie có thể được truy cập và chỉnh sửa bởi cả máy chủ và máy khách.

4. Sử dụng:

- Sử dụng để lưu trữ thông tin như tên người dùng, cài đặt ngôn ngữ, thông tin giỏ hàng trong mua sắm trực tuyến.

- Có thể sử dụng để theo dõi người dùng và mục đích quảng cáo.

**Session:**

1. Lưu trữ trên máy chủ (server-side):

- Session được lưu trữ trên máy chủ và chỉ có một "session ID" được gửi đến máy khách để xác định phiên làm việc.

2. Thời gian sống:

- Thường tồn tại trong suốt phiên làm việc của người dùng trên trang web.

- Session sẽ kết thúc khi người dùng đóng trình duyệt hoặc thoát khỏi trang web.

3. Dữ liệu lưu trữ:

- Thông tin được lưu trữ trên máy chủ và không nằm trực tiếp trên máy khách.

- An toàn hơn so với cookie vì dữ liệu không bị truy cập hoặc chỉnh sửa bởi người dùng.

4. Sử dụng:

- Sử dụng để lưu trữ thông tin nhạy cảm hoặc quan trọng như thông tin đăng nhập, thông tin phiên làm việc.

- Thích hợp cho các trang web yêu cầu bảo mật cao hơn, tránh lưu trữ thông tin quan trọng trên máy khách.

Tóm tắt:

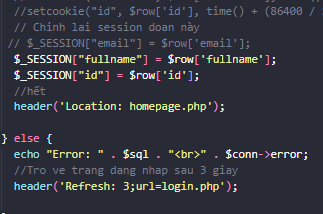
- Cookie: là cách lưu trữ thông tin trên máy khách, có thể thiết lập thời gian sống và được sử dụng để lưu trữ thông tin không nhạy cảm, dễ truy cập và chỉnh sửa.

- Session: là cách lưu trữ thông tin trên máy chủ, chỉ gửi "session ID" đến máy khách, an toàn hơn cho dữ liệu quan trọng và nhạy cảm, tồn tại trong phiên làm việc của người dùng trên trang web.

**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.**

**Trả lời:**

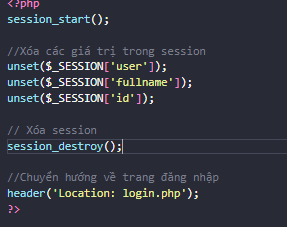
**+** Sửa file Log.php



+Sửa file Homepage.php



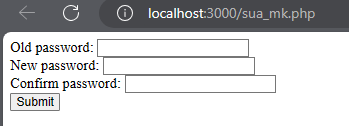
+Thêm file thoat.php



**7. 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.**

**Trả lời:**

+sua\_mk.php





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

**Trả lời:**

- Định nghĩa một biểu mẫu HTML để cho phép người dùng chọn tệp tin để tải lên:

+ Sử dụng thẻ `input` với phương thức POST và thuộc tính `enctype`=”multipart/form-data”

+ Sử dụng thẻ `` với thuộc tính type=”file” để người dùng có thể chọn tệp tin

- Trên phía máy chủ, sử dụng các hàm và thủ tục PHP để xử lí việc tải lên.

+ Sử dụng hàm `move\_uploaded\_file()` để di chuyển tệp tin được tải lên vào vị trí lưu trữ trên máy chủ.

+ Sử dụng biến siêu toàn cục `$\_FILES` để truy cập thông tin về tệp tin đã tải lên, bao gồm tên tệp tin, loại tệp tin, đường dẫn tạm thời và kích thước tệp tin

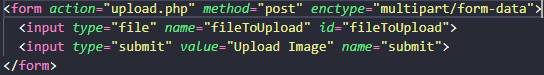
* Kiểm tra và xử lí lỗi khi tải lên tệp tin

+ Sử dụng thuộc tính `$\_FILE[‘file’][‘error’]` để kiểm tra mã lỗi liên quan đến tải lên tệp tin

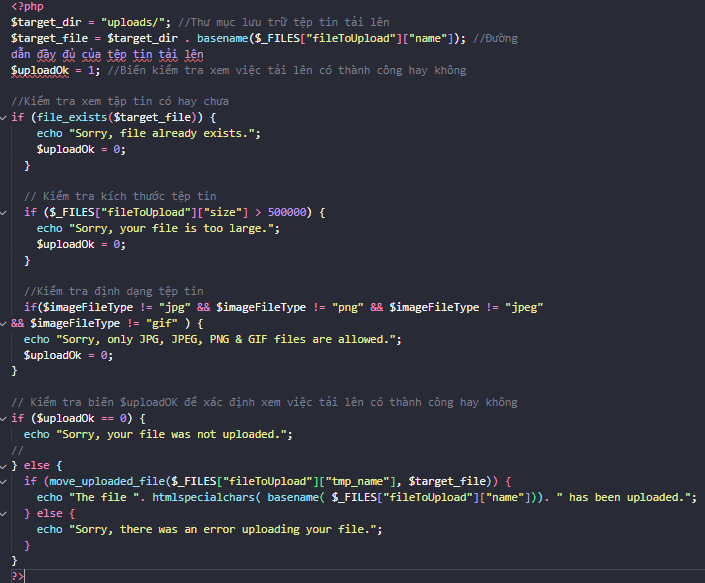
+ Sử dụng các câu lệnh điều kiện để kiểm tra lỗi và xử lí mỗi trường hợp lỗi.

Để thực hiện việc tải lên tệp tin trong PHP, ta cần làm theo các bước sau:

**Bước 1:** Định nghĩa biểu mẫu HTML cho việc tải lên tệp tin



**Bước 2:** Tạo một tập tin PHP( ví dụ: upload.php) để xử lí việc tải lên:



Trong ví dụ trên, chúng ta sử dụng biến siêu toàn cục `$\_FILES` để truy cập thông tin về tệp tin đã tải lên. `$\_FILES[“fileToUpload”][“name”] là tệp tin gốc, `$\_FILES[“fileToUpload”][“tmp\_name”] là kích thước tệp tin, `$\_FILES[“fileToUpload”][“tmp\_name”] là đường dẫn tạm thời của tệp tin tải lên.

**Bước 3:** Kiểm tra và xử lí lỗi khi tải tệp tin

Sử dụng các câu lệnh điều kineej để kiểm tra và xử lí lỗi trong tập tin upload.php.

Ví dụ: Kiểm tra xem tệp tin đã tinof tại, kiểm tra kích thước và đinh dạng tệp tin, kiểm tra xem việc di chuyển tệp tin đã thành công hay không.

**9. Bạn hãy đọc 5 , mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).**

**Trả lời:**

Dưới đây là mô tả về chức năng của một số hàm quan trọng và các tham số tương ứng trong hàm đó, dựa trên tài liệu từ trang web W3Schools và ví dụ về đọc tệp tin CSV (read-csv.php).

`file\_exists($filename): Kiểm tra xem một tệp tin có tồn tại hay không

* `$filename`: Tên hoặc đường dẫn tới tệp tin cần kiểm tra
* Trả về `true` nếu tệp tin tồn tại, ngược lại trả về `false`.

`is\_file($filename)`: Kiểm tra xem một tệp tin có phải là một tệp tin hay không

* `$filename`: Tên hoặc đường dẫn tới tệp tin cần kiểm tra.
* Trả về `true` nếu tệp tin tồn tại, ngược lại trả về `false`.

`file\_get\_contents($filename)`: Đọc nội dung của một tệp tin và trả về dưới dạng một chuỗi.

* `$filename`:Tên hoặc đường dẫn tới tệp tin cần đọc
* Trả về nội dung của tệp tin dưới dạng chuỗi, hoặc `false` nếu có lỗi xảy ra.

`fopen($filename, $mode)` :Mở một tệp tin hoặc URL

* `$filename`: Tên hoặc đường dẫn tới tệp tin cần mở.
* `$mode`: Chế độ mở tệp tin, ví dụ: “r” (mở tệp tin chỉ để đọc), “w’ (mở tệp tin để ghi, xóa nội dung cũ), “a” (mở tệp tin để ghi, giữ nội dung cũ).
* Trả về một con trỏ tệp tin(file pointer) hoặc `false` nếu có lỗi xảy ra.

`fgets($file)` và fgetcsv($file): Đọc một dòng từ tệp tin

* `$file`: Con trỏ tệp tin trả về từ hàm `fopen()`.
* `fgets($file)`: Đọc một dòng từ tệp tin và trả về dưới dạng một chuỗi.
* `fgetcsv($file)`: Đọc một dòng từ tệp tin và trả về dưới dạng một mảng chứa các giá trị được phân tách bằng dấu phẩy (dùng cho đọc tệp tin CSV).

Ví dụ về đọc tệp tin CSV (read-csv.php) sử dụng các hàm trên:



**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**

**Trả lời:**