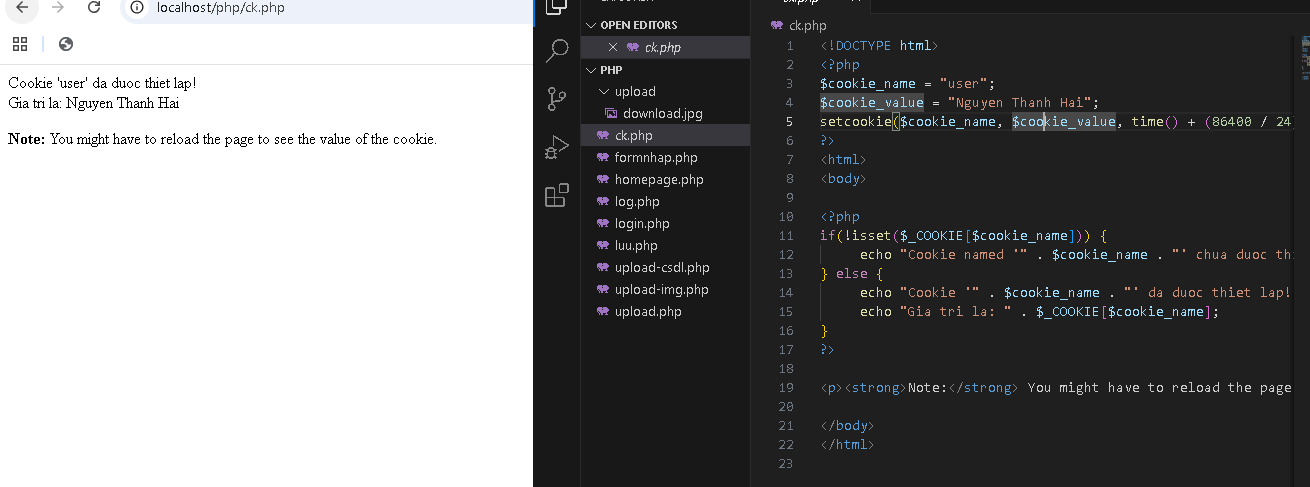
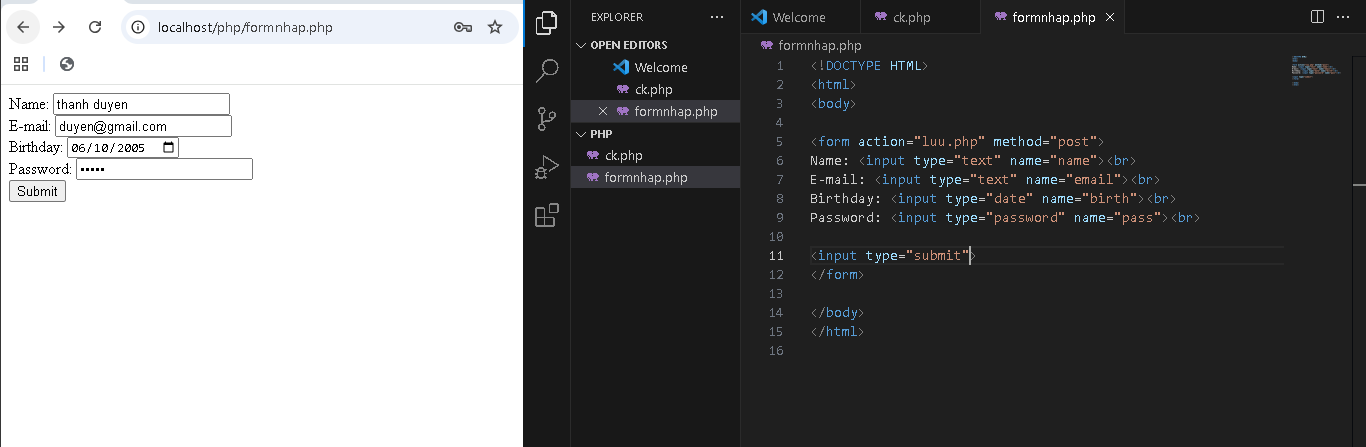
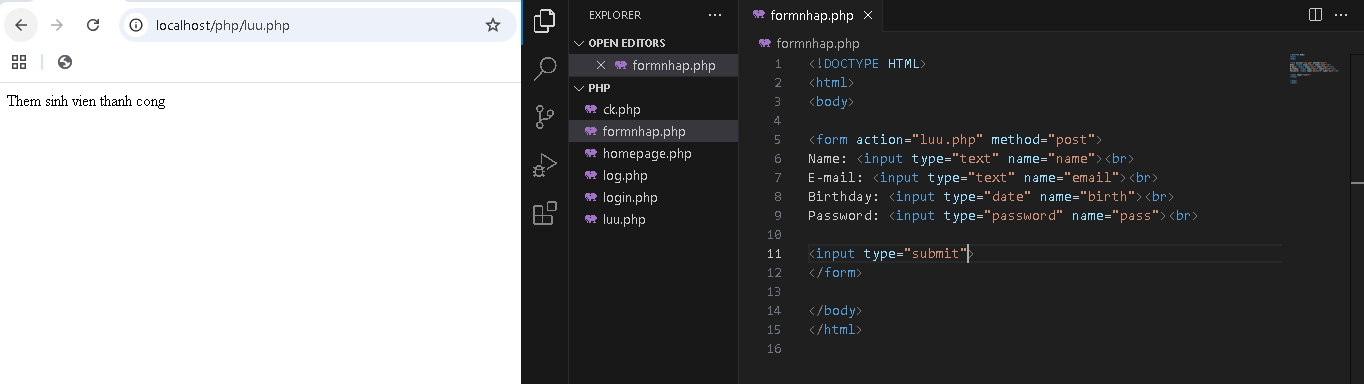
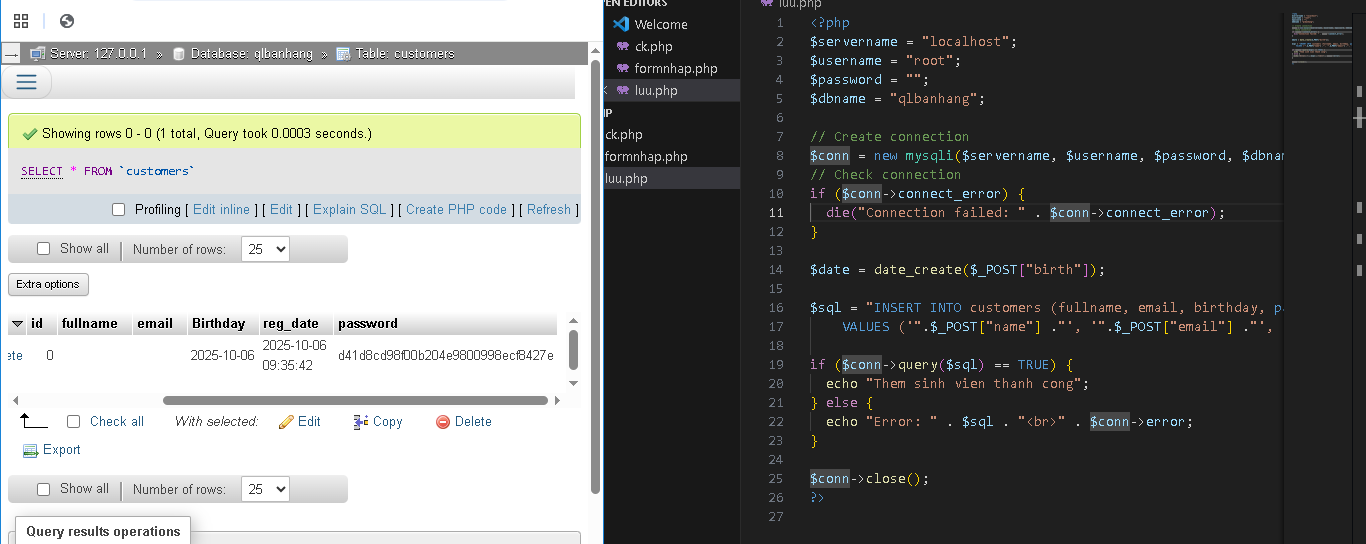
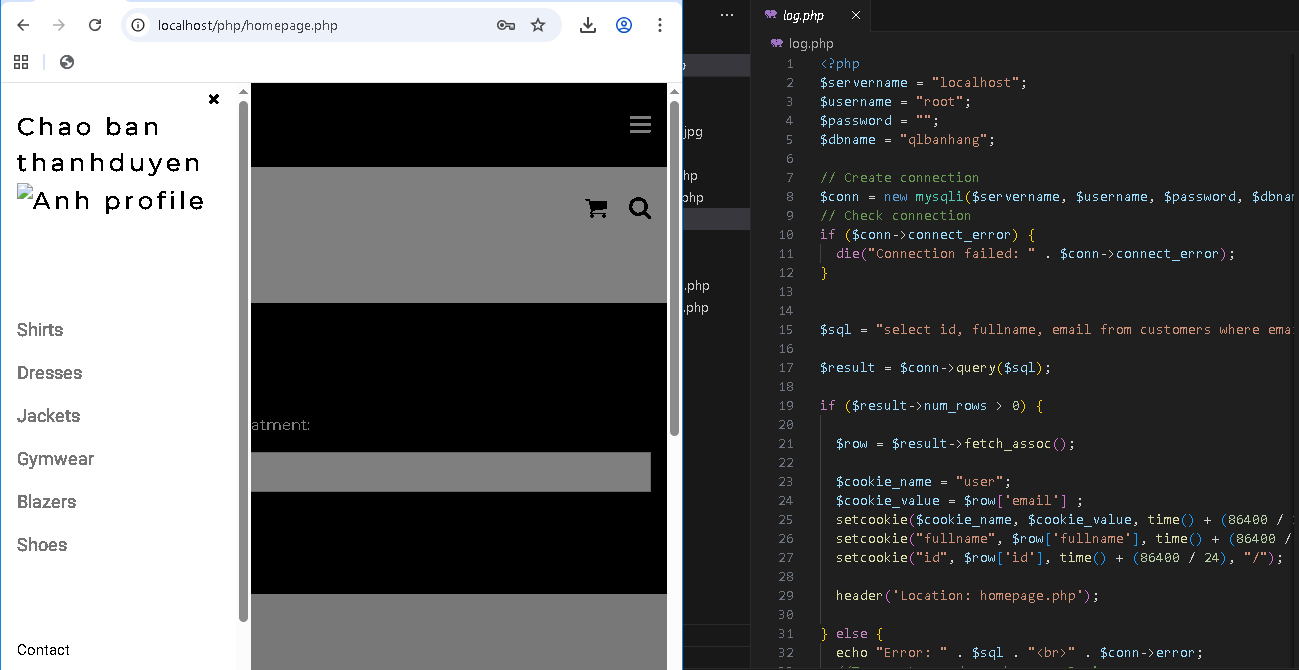
1. **Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.**

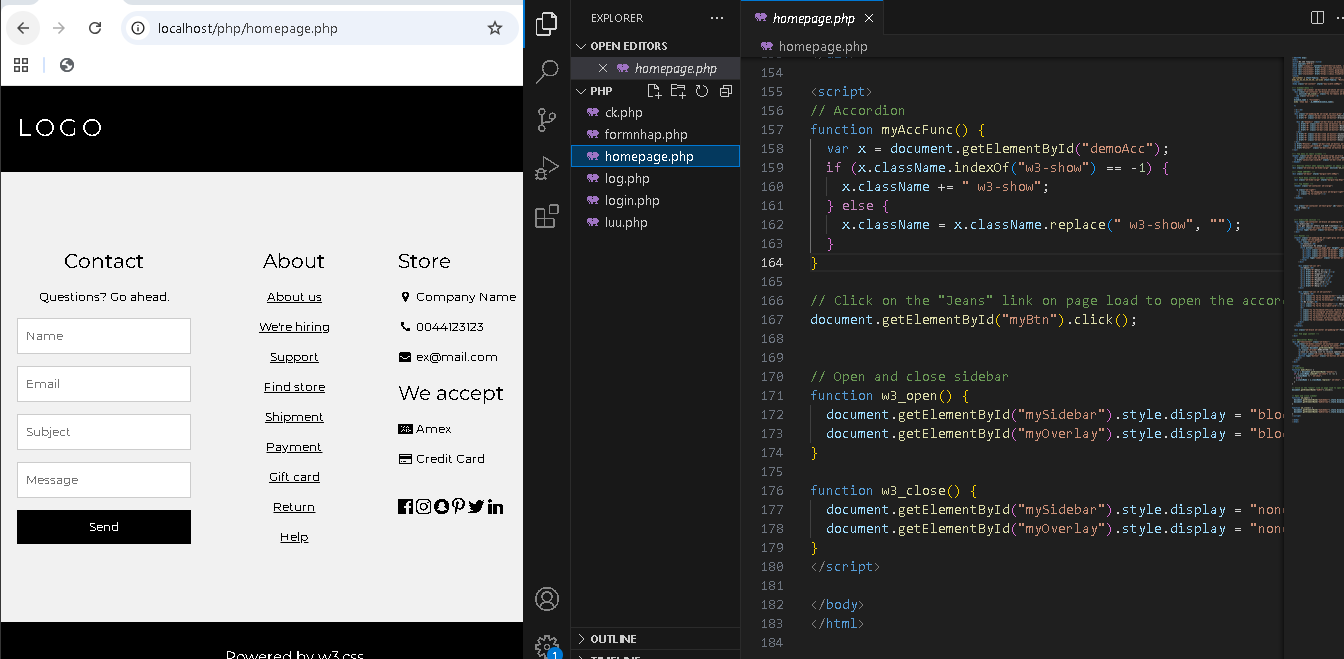


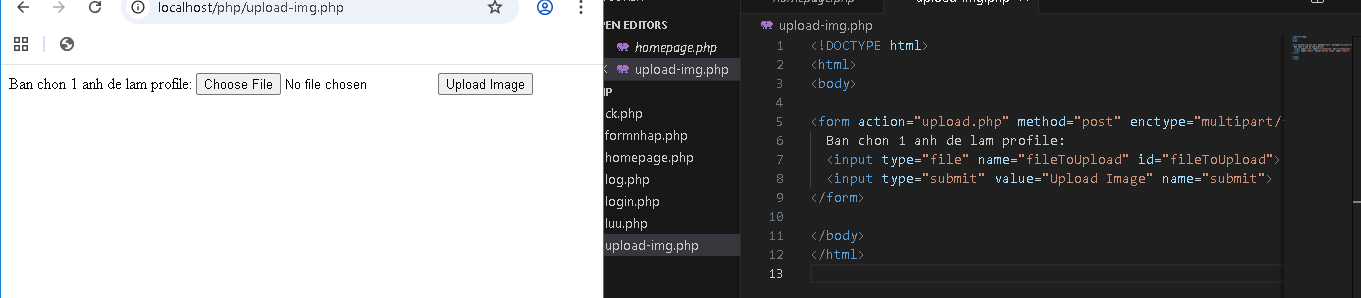


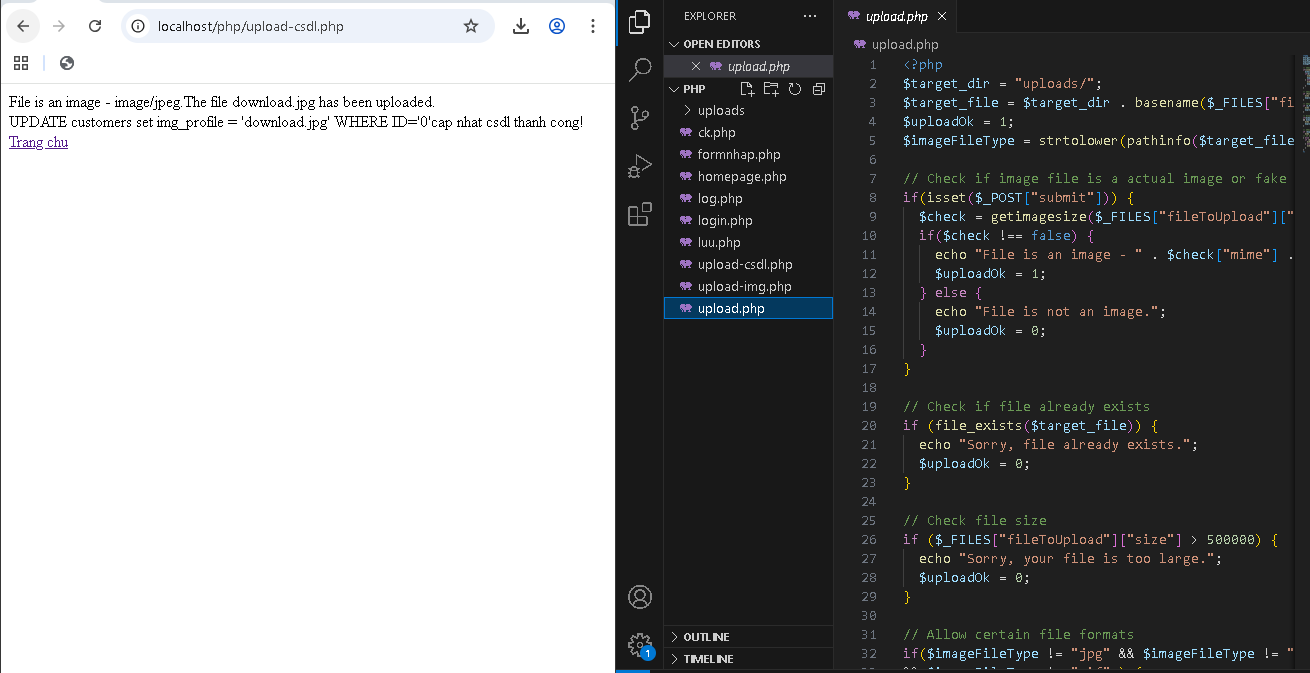


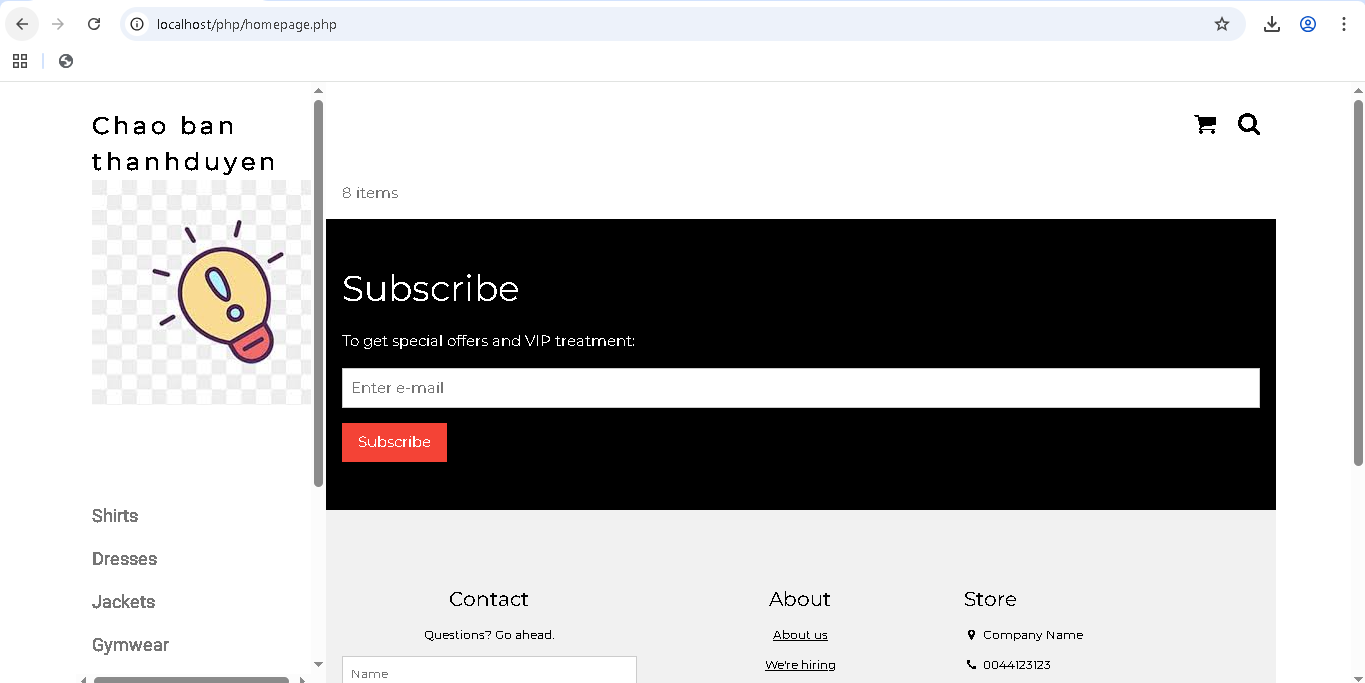


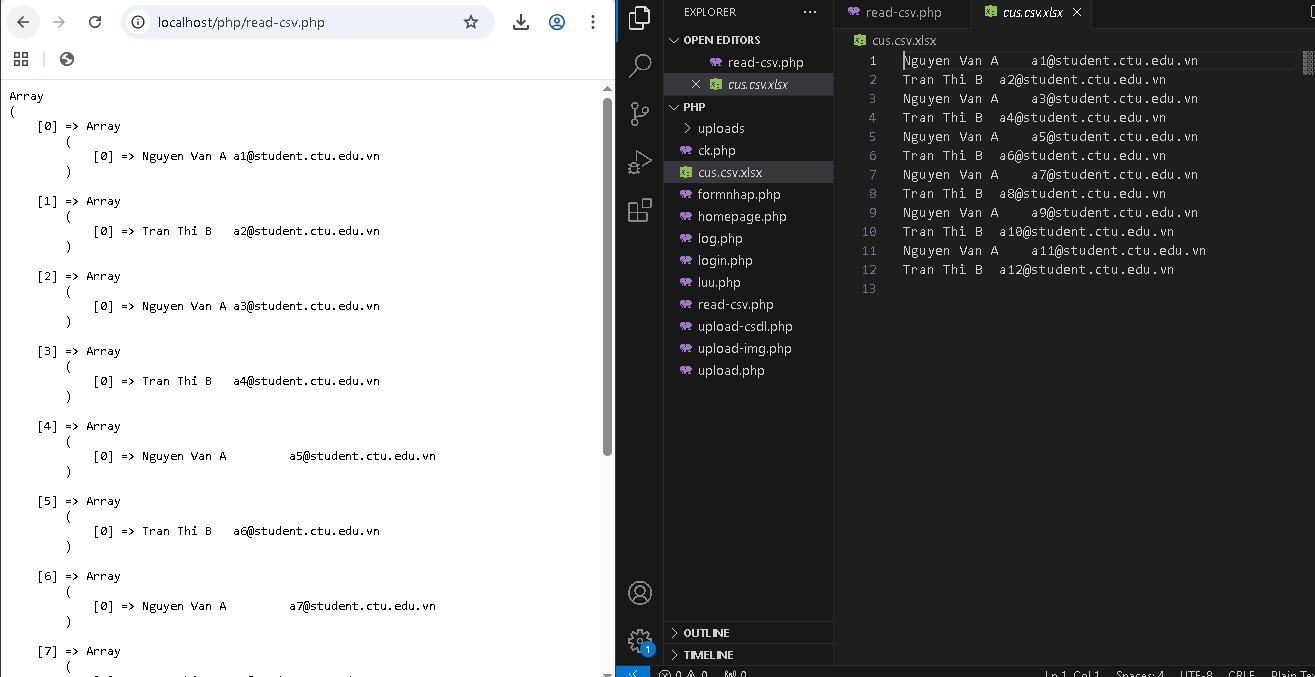












1. **Trong vài trường hợp, hacker có thể sử dụng các kỹ thuật SQL Injection để hack hệ thống của bạn. Bạn hãy trình bày SQL Injection là gì và thử các ví dụ trình bày trong . Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.**

SQL Injection là lỗ hổng khi ứng dụng chèn trực tiếp dữ liệu đầu vào (do người dùng cung cấp) vào câu lệnh SQL mà không kiểm soát/đối xử an toàn. Kẻ tấn công có thể chèn chuỗi dữ liệu đặc biệt để thay đổi ý nghĩa câu lệnh SQL — dẫn đến rò rỉ dữ liệu, truy cập trái phép, hoặc sửa/xóa dữ liệu.

1. **Dựa vào link, 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 một đoạn dữ liệu nhỏ được server gửi đến trình duyệt (client) và được trình duyệt lưu trữ lại. Mỗi khi trình duyệt gửi request đến server cùng domain, cookie sẽ được gửi kèm theo, giúp server nhận biết người dùng (như phiên làm việc, trạng thái đăng nhập, tùy chọn cá nhân, v.v).

Cookies giúp duy trì trạng thái giữa các request HTTP vốn là stateless (không trạng thái).

**Ý nghĩa các tham số:**

| **Tham số** | **Ý nghĩa** |
| --- | --- |
| $name | Tên của cookie, là khóa để nhận giá trị |
| $value | Giá trị cookie (nên mã hóa/encode nếu chứa ký tự đặc biệt) |
| $expires | Thời điểm hết hạn cookie (Unix timestamp). Nếu = 0 hoặc không set, cookie hết hạn khi đóng trình duyệt (session cookie) |
| $path | Đường dẫn trên server mà cookie có hiệu lực (mặc định: toàn bộ website /) |
| $domain | Domain mà cookie có hiệu lực (ví dụ: .example.com để áp dụng cho cả subdomain) |
| $secure | Nếu true, cookie chỉ gửi khi kết nối HTTPS |
| $httponly | Nếu true, cookie không thể truy cập bằng JavaScript (chỉ gửi qua HTTP header) |

**Cách lưu giá trị vào Cookie**

**setcookie('user', 'Alice', time() + 3600, "/");** Lưu cookie tên "user" với giá trị "Alice" trong 1 giờ

* Cookie sẽ được gửi kèm trong HTTP response header.
* Trình duyệt sẽ lưu lại cookie này.

**Cách lấy giá trị Cookie**

Dữ liệu cookie được PHP tự động đưa vào biến siêu toàn cục $\_COOKIE.

Ví dụ:

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

echo "Hello, " . htmlspecialchars($\_COOKIE['user']);

} else {

echo "Chưa có cookie user.";

}

**Cách xóa cookie**

Để xóa cookie, bạn set lại cookie đó với thời gian hết hạn đã qua (thường là thời điểm trong quá khứ).

Ví dụ:

// Xóa cookie 'user'

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

Lưu ý: Để xóa thành công, các tham số path và domain phải giống với khi tạo cookie.

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

**Session** trong PHP là một cơ chế lưu trữ dữ liệu tạm thời trên **máy chủ** (server) để duy trì trạng thái người dùng trong suốt phiên làm việc với ứng dụng web. Khác với **cookie**, dữ liệu session không được lưu trữ trên trình duyệt của người dùng, giúp bảo mật hơn và không bị giới hạn dung lượng.

**Ứng dụng phổ biến của session:**

* **Quản lý đăng nhập người dùng**: Lưu trữ thông tin người dùng sau khi đăng nhập thành công.
* **Giỏ hàng trong thương mại điện tử**: Lưu trữ thông tin giỏ hàng của người dùng.
* **Lưu trữ dữ liệu tạm thời giữa các trang**: Giữ thông tin người dùng khi điều hướng giữa các trang web.

**Lưu trữ dữ liệu vào session**

Sau khi khởi tạo session, bạn có thể lưu trữ dữ liệu vào biến siêu toàn cục $\_SESSION.

<?php

session\_start();

$\_SESSION['username'] = 'Alice';

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

?>

**Truy xuất dữ liệu từ session**

Để lấy dữ liệu từ session, bạn sử dụng cú pháp $\_SESSION['key'].

<?php

session\_start();

echo 'Username: ' . $\_SESSION['username'];

echo 'Role: ' . $\_SESSION['role'];

?>

**Xóa dữ liệu khỏi session**

Để xóa một biến cụ thể khỏi session, bạn sử dụng hàm unset().

<?php

session\_start();

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

?>

**Hủy toàn bộ session**

Để hủy toàn bộ session và xóa tất cả dữ liệu liên quan, bạn sử dụng hàm session\_destroy(). Lưu ý rằng hàm này chỉ hủy session trên server; nếu bạn muốn xóa tất cả cookie liên quan đến session, bạn cần gọi session\_unset() trước đó.

<?php

session\_start();

session\_unset(); // Xóa tất cả biến session

session\_destroy(); // Hủy session

?>

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

| **Tiêu chí** | **Cookie** | **Session** |
| --- | --- | --- |
| **Nơi lưu trữ** | Trình duyệt (client-side) | Máy chủ (server-side) |
| **Dung lượng lưu trữ** | Giới hạn khoảng 4KB | Không giới hạn hoặc rất lớn (tuỳ server) |
| **Thời gian tồn tại** | Có thể đặt thời gian hết hạn hoặc khi đóng trình duyệt | Thường tồn tại trong phiên làm việc, có thể cấu hình timeout |
| **Bảo mật** | Dễ bị sửa đổi hoặc đánh cắp nếu không mã hóa và cấu hình đúng | An toàn hơn vì lưu trên server, không dễ bị thay đổi trực tiếp |
| **Truy cập** | Được gửi tự động trong mọi request đến server | Truy cập thông qua session ID (thường qua cookie) |
| **Khả năng lưu trữ dữ liệu lớn** | Giới hạn nhỏ (4KB) | Có thể lưu dữ liệu lớn, tuỳ thuộc bộ nhớ server |
| **Phụ thuộc trình duyệt** | Có thể bị xóa, chặn hoặc tắt bởi người dùng | Không phụ thuộc trình duyệt, nhưng session sẽ mất khi hết thời gian hoặc bị xoá server |
| **Sử dụng phổ biến** | Lưu trữ thông tin nhẹ như tùy chọn người dùng, tracking | Quản lý trạng thái đăng nhập, dữ liệu tạm thời, giỏ hàng |
| **Khả năng kiểm soát** | Người dùng có thể tự xóa hoặc chỉnh sửa cookie | Server kiểm soát hoàn toàn dữ liệu session |
| **Tác động hiệu suất** | Gửi cookie đi kèm trong mọi request, có thể làm tăng dung lượng request | Dữ liệu không gửi đi kèm request, giảm tải dữ liệu truyền |

* **Cookie** dùng để lưu trữ dữ liệu nhẹ trên trình duyệt, thích hợp cho lưu tùy chọn người dùng, tracking, hay dữ liệu không nhạy cảm.
* **Session** lưu dữ liệu trên server, an toàn hơn, dùng để lưu trạng thái người dùng, bảo mật thông tin đăng nhập hoặc giỏ hàng.