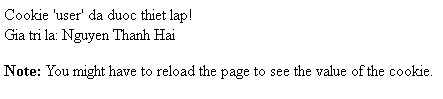
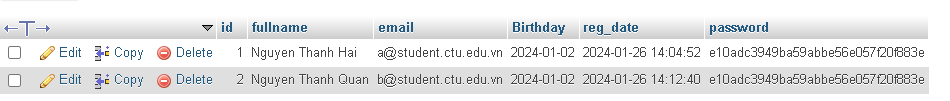
1. Bạn hãy chạy tất cả các lệnh hướng dẫn ở trên và chụp lại màn hình kết quả.

## Tạo Cookies



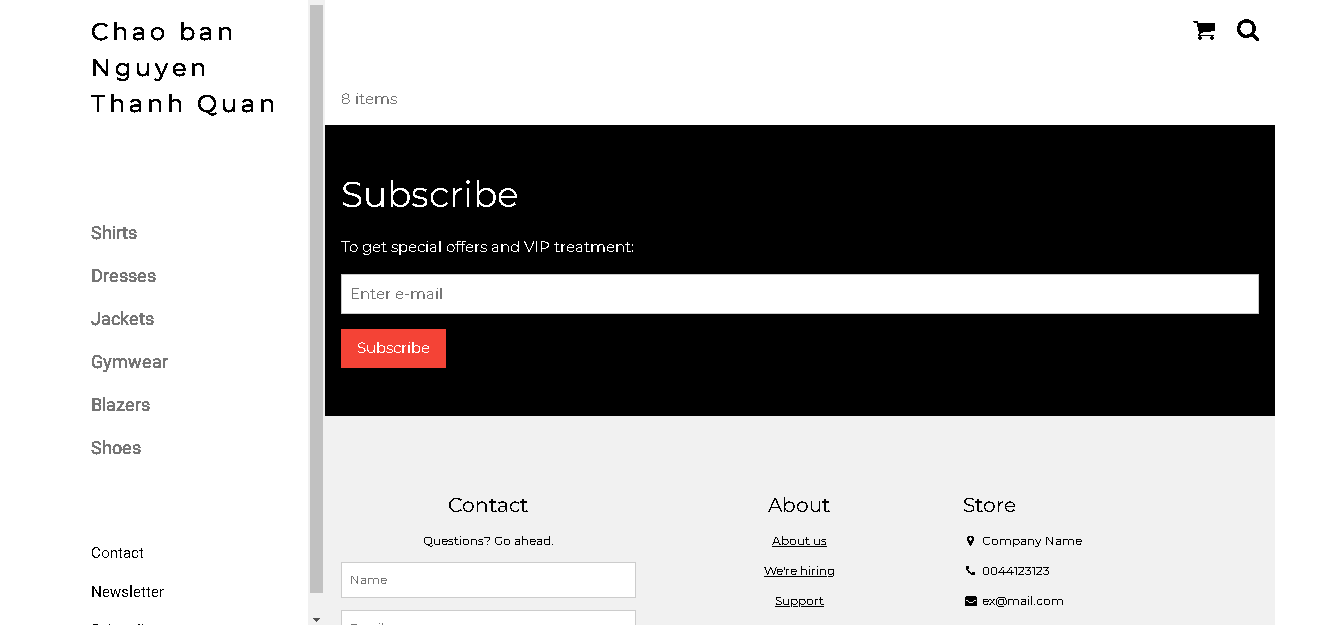
## Tạo 1 trang đăng ký người dùng với mật khẩu được mã hóa



## Tạo form đăng nhập



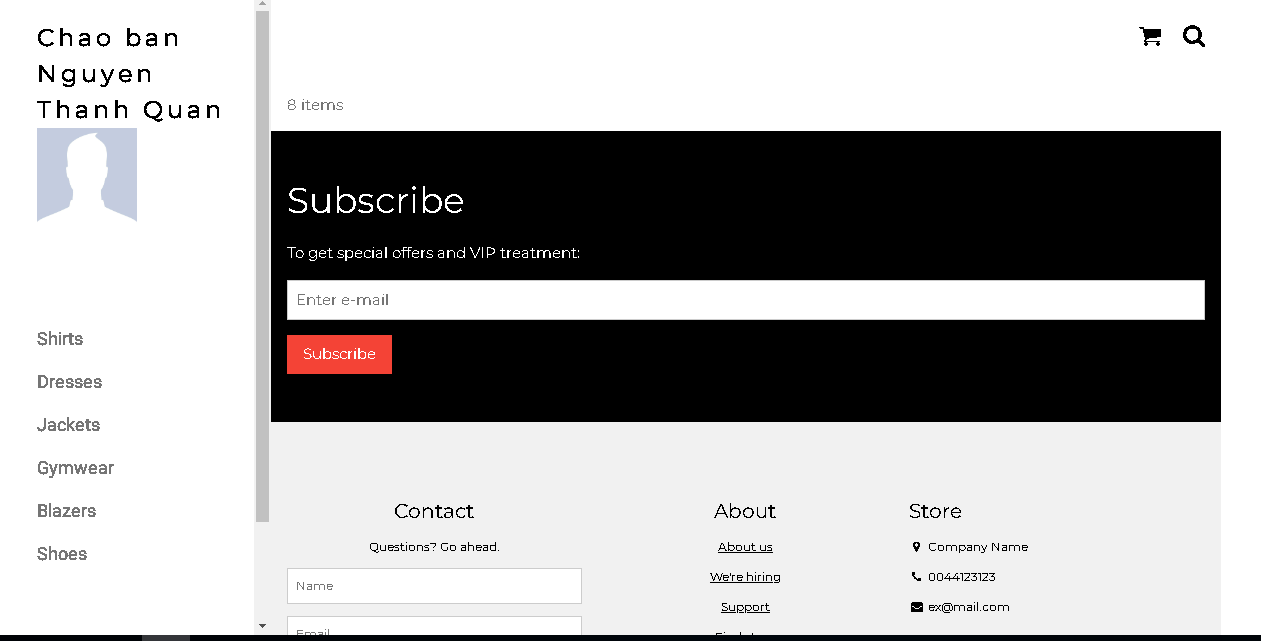




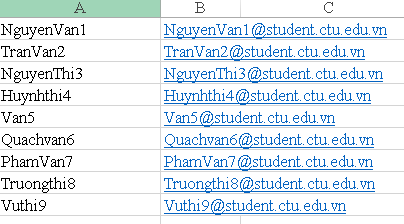
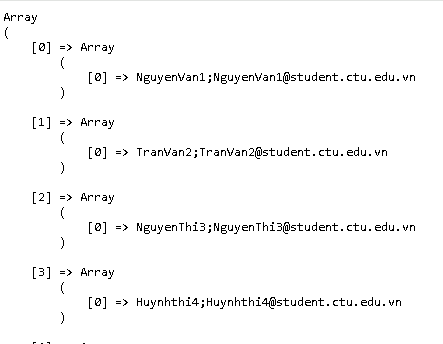
## Upload tập tin



## Upload ảnh và lưu dữ liệu về ảnh trong CSDL



Upload file csv và đọc dữ liệu vào mảng



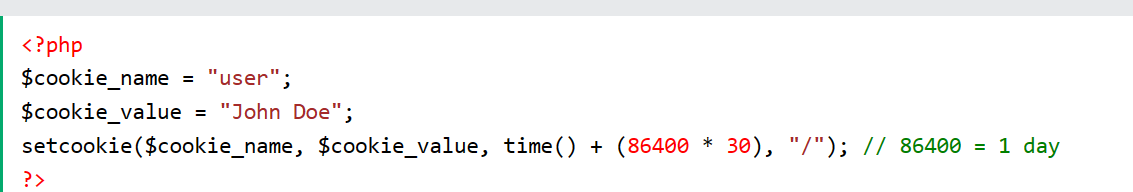
1. Trong vài trường hợp, hacker có thể sử dụng các kỹ thuật SQL Injection để hack hệ thống của bạn. Bạn hãy trình bày SQL Injection là gì và thử các ví dụ trình bày trong [[1]](#footnote-1). Ứng dụng kỹ thuật đó vào trang đăng nhập mà bạn đã tạo, chụp lại các kết quả.
2. Dựa vào link[[2]](#footnote-2), 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 lưu trữ trên máy tính của người dùng từ trang web mà họ đang truy cập.

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

* name: Tên của cookie.
* value: Giá trị của cookie.
* expire: thời gian hết hạn của cookie, . Nếu không chỉ định, cookie sẽ hết hạn khi trình duyệt đóng.
* path: Đường dẫn trên server mà cookie có thể được truy cập.
* domain: Miền của cookie, chỉ cookie nào thuộc miền này mới được gửi đi.
* secure: Nếu được đặt là true, cookie sẽ chỉ được gửi qua kết nối an toàn (HTTPS).
* httponly: Nếu được đặt là true, cookie sẽ chỉ có thể được truy cập thông qua HTTP và không thể bị JavaScript truy cập.

Lưu giá trị



Lấy giá trị

$value = $\_COOKIE["key"];

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

**Dùng để duy trì trạng thái:** Session được sử dụng để duy trì trạng thái thông tin giữa các request và response của người dùng trên trang web.

**Lưu giá trị:**

**$\_SESSION['key'] = 'value';**

**Lấy giá trị**

**$value = $\_SESSION['key'];**

**Xóa giá trị**

**unset($\_SESSION['key']);**

1. Bạn hãy so sánh Cookie và Session.

|  |  |
| --- | --- |
| **Cookie** | **Session** |
| **Cookie** được lưu trữ trên trình duyệt của người dùng. | **Session** không được lưu trữ trong trình duyệt của người dùng. |
| Dữ liệu **cookie** được lưu trữ ở phía máy khách. | Dữ liệu **session** được lưu trữ ở phía máy chủ. |
| Dữ liệu **cookie** dễ dàng sửa đổi khi chúng được lưu trữ ở phía khách hàng. | Dữ liệu **session** không dễ dàng sửa đổi vì chúng được lưu trữ ở phía máy chủ. |
| Dữ liệu **cookie** có sẵn trong trình duyệt của chúng ta đến khi hết hạn. | Dữ liệu **session** có sẵn cho trình duyệt chạy. Sau khi đóng trình du |

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.
2. 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.
3. Bạn đọc trong[[4]](#footnote-4) để tìm hiểu và mô tả các cơ chế, các hàm/thủ tục để thực hiện việc upload.
4. Bạn hãy đọc[[5]](#footnote-5), mô tả chức năng của hàm này và các tham số trong hàm (tham khảo thêm từ read-csv.php).

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

1. <https://www.w3schools.com/sql/sql_injection.asp> [↑](#footnote-ref-1)
2. <https://www.w3schools.com/php/php_cookies.asp> [↑](#footnote-ref-2)
3. <https://www.w3schools.com/php/php_sessions.asp> [↑](#footnote-ref-3)
4. <https://www.w3schools.com/php/php_file_upload.asp> [↑](#footnote-ref-4)
5. <https://www.w3schools.com/php/func_filesystem_file.asp> [↑](#footnote-ref-5)