Câu 1: Xây dựng cơ chế **Quản Lý trạng thái phiên làm việc** để đảm bảo an toàn cho ứng dụng web

Câu 2: Xây dựng cơ chế **quản lý tài khoản người dùng** để đảm bảo cho ứng dụng web

Câu 3: Xây dựng cơ **chế xác thực và phân quyền người dùng** để đảm bảo cho ứng dụng web

Câu 1: Quản lý trạng thái phiên làm việc

* Phiên (Session) là chuỗi các thông điệp HTTP Request và Http Response trao đổi giữa trình duyệt và website được lưu ở phía server
* Yêu cầu để đảm bảo an toàn:
  + Phiên làm việc chỉ khả thi người dùng đã xác thực
  + Thời gian phiên làm việc không quá dài
  + Khi đăng xuất thì phiên làm việc phải được kết thúc
  + Hạn chế nhất có thể trường hợp bị lộ thông tin về session hay cookie

Tại thời điểm đăng nhập, phiên sẽ được tạo

* **session\_start();**

Trước khi được Access Control thì tiến hành kiểm tra xác thực người dùng

* **if(!isset($\_SESSION[“id\_user”])** //Nếu biến $\_SESSION[“id\_user”] chưa tồn tại thì cần quay lại trang đăng nhập để tiến hành xác thực người dùng

**{**

**header(‘Location: dangnhap.php’);**

**}**

Set thời gian lưu thông tin người dùng trong phiên

* **$\_SESSION[“id\_user”]= IdUser;**
* **$\_SESSION[“start”]=time();**
* **If(time()-$\_SESSION[“start”] > 600)**

**{**

**session\_unset();**

**session\_destroy();**

**header(‘Location:dangnhap.php’);**

**}**

Khi đăng xuất thì phiên làm việc phải được kết thúc:

* **unset($\_SESSION[“id\_user”]);**
* **session\_destroy();**

Ngoài ra có thể dùng HTTPonly Flag và SECURITY flag để đảm bảo cookie lưu trên trình duyệt không bị lấy cắp dễ dàng

* **session.cookie\_httponly = true**
* **session\_cookie\_secure = true**

Câu 2: Xây dựng cơ chế **quản lý tài khoản người dùng** để đảm bảo cho ứng dụng web (câu ni chưa chắc)

Quản lý tài khoản người dùng giúp quản trị viên dễ dàng hơn trong việc quản lý, sắp xếp, phân loại hay thực thi các yêu cầu trên tài khoản người dùng. Để quản lý được tài khoản người dùng an toàn, đảm bảo trên ứng dụng web cần :

* Phân biệt rõ ràng, có các giá trị duy nhất nhằm xác định danh tính cụ thể
* Phân quyền cụ thể, đủ đáp ứng nhu cầu của người dùng

Phân biệt người dùng bằng cách gán mỗi tài khoản một giá trị duy nhất ( prime key ), cụ thể là ID\_user khi tạo database, tránh trường hợp bị trùng lặp.

Phân quyền hợp lý cho việc quản lý, cụ thể là những chức năng mà người dùng bình thường không được cấp như chỉnh sửa thông tin người khác, thay đổi một số trường dữ liệu. Chỉ có quản trị viên mới có quyền thay đổi.

* //thiết lập quyền “permission” cho tài khoản admin là 1, người dùng bình thường là 0

**If($\_SESSION[‘ID\_USER’]=”admin”){**

**$\_SESSION[‘permission’]=1;**

**)else{**

**$\_SESSION[‘permission’]=0;**

**)**

* **If(!isset($\_SESSION[‘id\_user’])){**

//kiểm tra người dùng chưa đăng nhập thì quay lại trang đăng nhập để xác thực người dùng

**Header(‘Location: dangnhap.php’);**

**}else{**

**If($\_SESSION[‘permission’]==0){**

**Echo”Bạn không có quyền truy cập vào trang này !”;**

**Exit();**

}

Câu 3: Xây dựng cơ **chế xác thực và phân quyền người dùng** để đảm bảo cho ứng dụng web

Phân quyền hợp lý cho việc quản lý, cụ thể là những chức năng mà người dùng bình thường không được cấp như chỉnh sửa thông tin người khác, thay đổi một số trường dữ liệu. Chỉ có quản trị viên mới có quyền thay đổi.

* //thiết lập quyền “permission” cho tài khoản admin là 1, người dùng bình thường là 0

**If($\_SESSION[‘ID\_USER’]=”admin”){**

**$\_SESSION[‘permission’]=1;**

**)else{**

**$\_SESSION[‘permission’]=0;**

**)**

* **If(!isset($\_SESSION[‘id\_user’])){**

//kiểm tra người dùng chưa đăng nhập thì quay lại trang đăng nhập để xác thực người dùng

**Header(‘Location: dangnhap.php’);**

**}else{**

**If($\_SESSION[‘permission’]==0){**

**Echo”Bạn không có quyền truy cập vào trang này !”;**

**Exit();**

}

Xác thực người dùng khi đăng nhập là bước quan trọng, đối chiếu dữ liệu người dùng nhập vào và dữ liệu đã được lưu trên server nhằm so sánh, nếu đúng thì access còn không thì thông báo đăng nhập không thành công. Để an toàn, thay vì cộng chuỗi ta nên dùng phương thức dùng biến parameter trong câu truy vấn SQL tránh khai thác SQL Injection

* $query = "SELECT id FROM member WHERE username= ? AND pass= ? ";

$stmt = mysqli\_prepare($conn, $query);

        mysqli\_stmt\_bind\_param($stmt, "ss", $username, $pass);

        $stmt->execute();

Hoặc có thể dùng filter để lọc các ký tự lạ nhập từ bàn phím

$username = mysqli\_real\_escape\_string($conn, $username);

$pass = mysqli\_real\_escape\_string($conn, $pass);

Câu 4: Phân biệt phương thức POST và GET

Câu 5:Những phương thức quản trạng thái phiên làm việc . Nêu ưu và nhược điểm

Câu 6: Phân biệt cookie và session

Câu 7: Trình bày các kỹ thuật Lọc dữ liệu người dùng

Câu 8: xây dựng chức năng đăng nhập

Câu 9: xây dựng chức năng thêm sản phẩm

Câu 10: xây dựng chức năng đổi mật khẩu người dùng

Câu 8:

Để đăng nhập được an toàn cần một số tiêu chí sau:

* Form đăng nhập phải được lọc ký tự đầu vào
* Dùng truy vấn tham số thay cho truy vấn cộng chuỗi
* Không hiện thông báo lỗi của Cơ sở dữ liệu
* Điền đầy đủ các trường cần thiết

Để lọc ký tự đầu vào ta dùng các hàm có sẵn như escape\_string

$username = mysqli\_real\_escape\_string($conn, $username);

$pass = mysqli\_real\_escape\_string($conn, $pass);

Dùng tuy vấn tham số parameter

* $query = "SELECT id FROM member WHERE username= ? AND pass= ? ";

$stmt = mysqli\_prepare($conn, $query);

        mysqli\_stmt\_bind\_param($stmt, "ss", $username, $pass);

        $stmt->execute();

Không hiện thông báo lỗi khi có vấn đề về cơ sở dữ liệu

die($message);

Yêu cầu người dùng nhập đủ các trường cần thiết

**If(!$username||!$password){**

**Echo “vui lòng nhập đầy đủ tên đăng nhập và mật khẩu !”;**

**Exit();**

**}**

Câu 9: xây dựng chức năng thêm sản phẩm

Thêm sản phẩm có thể là chức năng của mỗi người dùng hoặc chức năng của chỉ quản trị viên. Nếu là chỉ có quản trị viên mới được thêm sản phẩm thì chúng ta phải thực hiện phân quyền người dùng

$permission = 0 //Nếu là người dùng bình thường

$permission = 1// Nếu là quản trị viên

**If($permission =0){**

**Echo”bạn không có quyền vào trang này!”;**

**Exit();**

**}**

Nếu thêm sản phẩm là chức năng của mỗi người dùng trong hệ thống, chúng ta cần đảm bảo một số tiêu chí sau:

* Lọc ký tự đầu vào
* Sử dụng blacklist để tránh inject một số script độc hại
* Lựa chọn hóa các mục để người dùng lựa chọn hơn là nhập từ bàn phím

Lọc ký tự dùng các hàm có sẵn như escape\_string

$product1 = mysqli\_real\_escape\_string($conn,$product1);

Sử dụng blacklist

**Function check\_blacklist($query)**

**{**

**$blacklist=array(“UNION”,”SCHEMA”,”DELETE”) //vi du**

**$strToUpper = strtouper($qr); // Chuyển câu truy vấn thành chữ hoa**

**For($i=0 ; $i < count($blacklist) ; $i++){**

**$blackkey=$blacklist[$i];**

**$pos = strpos($strToUpper, $blackkey); //nếu từ khóa được phát hiện thì sẽ trả về vị trí của nó**

**If( $pos >0 ) {**

**Return false;**

**}**

**} return true; }**

Lựa chọn hóa các mục để người dùng lựa chọn và phân loại

**<select name = “loại hàng”>**

**<option value= “”></option>**

**<option value= “lua chon 1”></option>**

**<option value= “lua chon 2”></option>**

**</select>**

Câu 10: xây dựng chức năng đổi mật khẩu người dùng

Chức năng đổi mật khẩu cho người dùng cần một số tiêu chí an toàn sau

* Lọc ký tự đầu vào
* Yêu cầu nhập mật khẩu cũ
* Dùng hàm băm mật khẩu trước khi lưu vào database

Lọc ký tự dùng hàm escape\_string có sẵn

**$pass = mysqli\_real\_escape\_string($conn,$pass);**

Sau khi nhập mật khẩu cũ cần kiểm tra bằng cách so sánh chuỗi lấy từ dtb

**$pass\_old = md5($\_POST[‘txtPassOld]);**

**$pass\_old\_real[‘pass’] = ($conn->query(“Select pass from member where username = $\_SESSION[‘username’]”)->fech\_assoc();**

**If( $pass\_old != $pass\_old\_real[‘pass’]){**

**Echo”Nhap sai mat khau cu !”;**

**exit();**

**}**

Nếu sau khi check xác thực, trước khi lưu mật khẩu mới, cần dùng hàm băm để bảo mật

**$pass = md5($pass);**