### **Đề 01:**

Câu 1: Web Service là:

* ***Phần mềm thực hiện tương tác dữ liệu với phần mềm khác qua các giao thức web***

Câu 3: Thẻ (Tag) trong XML có phân biệt chữ hoa, chữ thường không?

* ***Có***

Câu 4: Khi thiết lập httponly=true thì:

* ***Trình duyệt không được truy cập cookie bằng javascript***

Câu 5: Lệnh jQuery ẩn phần tử có id=test:

* ***$("#test").hide();***

Câu 6: Bên trong phần tử HTML nào đặt JavaScript?

* **<script>**

Câu 8: HTML chính xác để tạo vùng văn bản gì?

* **<textarea>**

Câu 9: Làm thế nào thêm màu nền cho tất cả các phần tử <h1>?

* **h1 { background-color: #FFFFFF;}**

Câu 10: Dữ liệu REST API (RESTful API) trả về chỉ có dạng JSON

* **Sai**

Câu 11: Làm cách nào để viết câu lệnh IF trong JavaScript?

* **if (x == 5) {**

**}**

**Câu 12:** Mẫu an toàn (Security Patten) là:

**Mẫu an toàn là một mẫu mô tả cách thức ngăn ngừa một hiểm họa bằng việc sử dụng các cơ chế an toàn**

VD: RBAC

Câu 13: Các biện pháp đảm bảo an toàn phần mềm ở giai đoạn “Phát triển”

* **Mô hình hiểm họa > Mẫu thiết kế an toàn > Lập trình an toàn > Kiểm thử an toàn > Triển khai**

Câu 14: Đâu là cú pháp khai báo đúng của Bộ chọn nhóm trong CSS:

* **h1, h2, p{**

**text-align: center;**

**color: red;**

**}**

Câu 15: Các thành phần cơ bản của HTML

* **Các thẻ (Tags), các phần tử (Elements), các thuộc tính (Attributes)**

Câu 16: HTML chính xác để tạo danh sách thả xuống là gì

* **<select>**

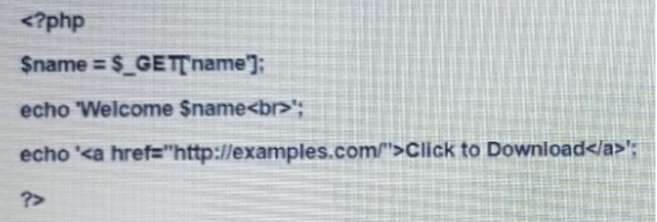
Câu 17: Đoạn mã sau kết nối cơ sở sử dụng

* MySQL

Câu 18: Theo OWASP TOP 10 2021, hiểm họa nào được xếp hạng thứ 9?

* **Security Logging and Monitoring Failures**

Câu 19: Trang web sau có thể bị tấn công bằng:



* **XSS - Cross site Scripting**

Câu 20: Lệnh HTML chính xác để chèn hình ảnh là gì

* **<img src="image.gif" alt="MyImage">**

Câu 21: PHP viết tắt của từ gì?

* **PHP: Hypertext Preprocessor**

Câu 22: Làm thế nào để khai báo một biến JavaScript?

* **var carName;**

Câu 23: Cú pháp chính xác của một cặp NAME/VALUE trong JSON

* **“teacher” {**

**“name” : “Trần Văn Bình”,**

**“department” : “An toàn thông tin”**

**}**

Câu 24: Cách chính xác sử dụng jQuery CDN

* **<head>**

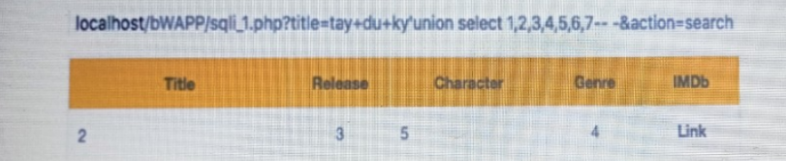
**<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>**

**</head>**

Câu 25: Phát biểu nào SAI

* **Có thể dùng kiểm tra hợp thức phía client và server thay thế cho nhau**

Câu 26: Trong hình sau các số 2,3,4,5 trong bảng có nghĩa là gì ?



* **Chỉ có những giá trị của trường tương ứng với những số này trong câu truy vấn mới hiển thị trong bảng**

Câu 27: Làm thế nào để chọn một phần tử có ìd “demo” trong CSS

* **#demo {**

**/\* các thuộc tính CSS \*/**

**}**

Câu 28: Làm thế nào để chọn các phần thử có tên lớp “test”

* **.test {**

**/\* các thuộc tính CSS \*/**

**}**

Câu 29: Cú pháp chính xác để tham chiếu đến tập lệnh bên ngoài có tên “xxx.js” là gì?

* **<script src="xxx.js"></script>**

Câu 30: Doctype nào phù hợp với HTML5

* **<!DOCTYPE html>**

Câu 31: Lệnh HTML chính xác để tham chiếu đến bảng định kiểu bên ngoài (External CSS) là gì?

* **<link rel="stylesheet" type="text/css" href="style.css">**

Câu 32: Biện pháp phòng chống tấn công CSRF - Cross site Request Forgery là?

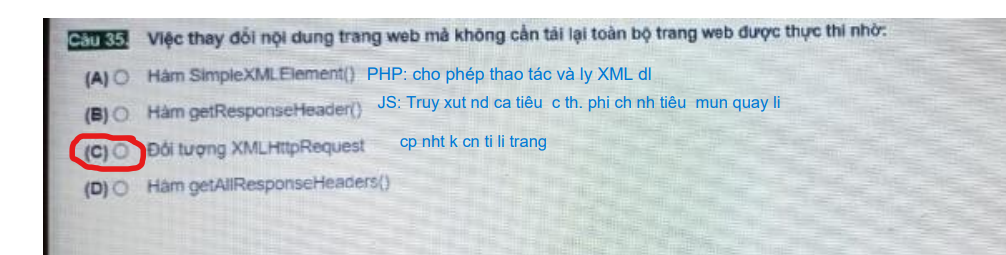
* **Sử dụng form token**

Câu 34: Trong PHP, cách duy nhất để xuất văn bản là sử dụng echo đúng không?

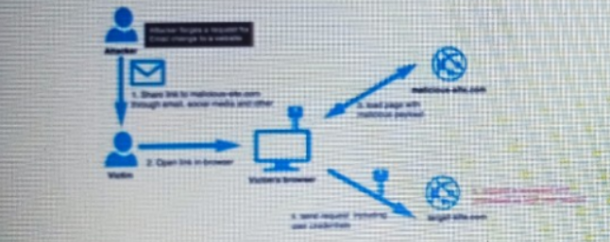
* **Sai**

Câu 35: Việc thay đổi nội dung trang web mà không cần tải lại toàn bộ trang web được thực thi nhờ hàm nào?

* **Đối tượng XMLHttpRequest**



**Câu 36:** Hình sau mô tả loại tấn công nào



**CSRF - Cross-site Request Forgery**

Câu 37: Làm thế nào để chọn tất cả các phần tử p bên trong phần tử div?

* **div.p**

Câu 38: HTML chính xác để chèn hình nền là gì?

* **<body style="background-image: url(background.gif)">**

Câu 39: Khi sử dụng phương thức POST, các biến được hiển thị trong URL

* **Sai**

Câu 40: Hàm PHP nào chỉ loại bỏ tất cả các biến phiên

* **session\_unset()**

Câu 41: Theo OWASP TOP 10 2021, hiểm họa nào được xếp hạng thứ 7?

* **Nhận dạng và xác thực không thành công (Identification and Authentication Failures).**

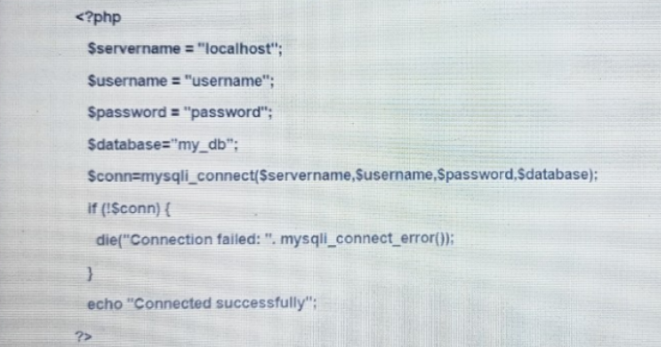
Câu 42: Các tệp bao gồm trong PHP phải có phần mở rộng tệp “.inc”

* **Sai**

Câu 43: Một <iframe> được sử dụng để hiển thị một trang web trong một trang web

* **True**

Câu 44: Đoạn mã sau kết nối cơ sở sử dụng



* **MySqli ( Thủ tục )**

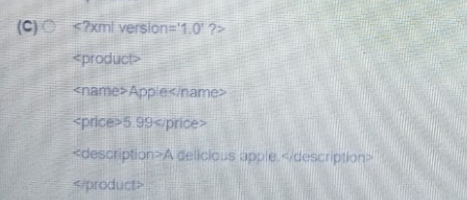
Câu 45: Thẻ HTML nào được sử dụng để xác định Bảng định kiểu bên trong Internal CSS

* **<style>**

Câu 46: Làm thế nào để viết “Hello World” trong một hộp cảnh báo

* **alert("Hello World");**

Câu 47: Khai báo nào sau đây khai báo chính xác một tài liệu XML



Câu 48: Lệnh HTML chính xác để tạo siêu liên kết là gì

* **<a href="url">text</a>**

Câu 49: Làm thế nào để hiển thị một đường viền như thế này: đường viền trên cùng = 10 pixel, đường viền dưới dùng = 5 pixel, đường viền bên trái = 20 pixel, đường viền bên phải = 1 pixel

* **border-width: 10px 1px 5px 20px**

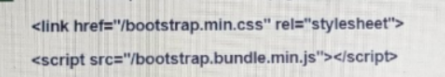
Câu 50:Đâu không phải là cách phòng chống tấn công CSRF trong framework Laravel?



Câu 52: Làm cách nào để tạo hàm trong Js?

* **function myFunction()**

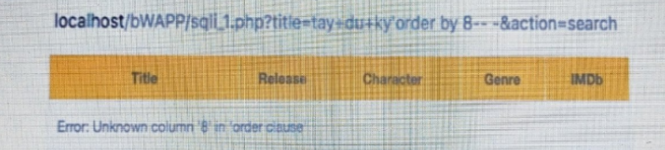
Câu 53: Mã khai báo phía dưới để sử dụng Bootstrap CDN Đúng hay Sai: **Đúng**



Câu 54: Đâu là vị trí chính xác để chèn JS?

* **Cả phần <head> và <body>**

**Câu 55:** Kết quả thu được như hình vẽ sau, cho thấy



* **Câu truy vấn CSDL tương ứng có kết quả có số cột <8**

Câu 56: Hàm nào trong PHP thực hiện việc loại bỏ các ký tự điều khiển có trong đầu vào

* **filter\_var(), filter\_var\_array(),filter\_input() và filter\_input\_array()**

Câu 57: Kết quả của việc mô hình hóa hiểm họa là

* **Danh sách hiểm họa và cách thức xử lý hiểm họa**

Câu 58: Để đảm bảo an toàn cho ứng dụng web, cần thực hiện mô hình hóa hiểm họa ở giai đoạn nào trong quá trình phát triển phần mềm theo mô hình thác nước

* **Thiết kế**

Câu 59: HTML là viết tắt của từ gì

* **Hypertext Markup Language**

Câu 60: Lệnh HTML chính xác để thêm màu nền là

* **<body style="background-color: yellow;">**

### **Đề 02:**

1. Cú pháp JS chính xác thay đổi nội dung của p.tử HTML bên dưới :

<p id = “demo” > Hello World ! </p>

**document.getElementById ("demo"). innerHTML = "This is show";**

1. Đâu là vị trí chính xác để chèn JS: **giữa cặp thẻ <script> và </script>**
2. Khai báo chính xác XML

**<?xml**

**version="version\_number"**

**encoding="encoding\_declaration"**

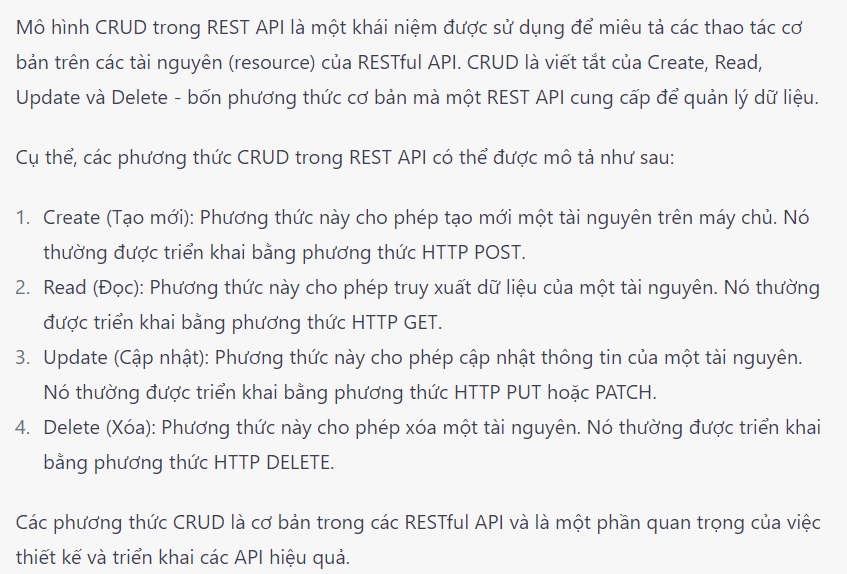
**standalone="standalone\_status"**

**?>**

1. Hàm nào trong PHP thực hiện việc loại bỏ các kí tự điều khiển có trong đầu vào: **Trim()**
2. Hàm trong PHP thực hiện việc chuyển đổi các ký tự điều khiển thành “ escape sequence “ tương ứng : addslashes()
3. Lệnh HTML chính xác để tạo hộp kiểm: **<input type="checkbox">**
4. Phần tử HTML cho tiêu đề lớn nhất: **<h1> </h1>**
5. CSS là viết tắt của **"Cascading Style Sheets"**
6. Cách chính xác để viết một mảng JavaScript là: **var myArray = [element1, element2, ..., elementN];**
7. Phần tử nào trong số những phần tử này đều là phần tử <table>

**<tr><td>**

1. Phần tử HTML sử dụng để chỉ định tiêu đề cho tài liệu ? **<title>**
2. Mô hình CRUD trong REST API ?



1. PHP có thể chạy trên Microsoft Windows IIS ? **Đúng**
2. Thuộc tính HTML nào sử dụng để xác định kiểu nội tuyến? **style** ( inline css)
3. Trong framework Laravel, View welcome đc định nghĩa với tên: **welcome.blade.php**
4. Mô hình MVC , có nhiều ph.án khác nhau về luồng tương tác giữa 3 thành phần ? **Đúng**

### **Note:**

#### **Danh sách top 10 OWASP trong năm 2021:**

1. Kiểm soát truy cập bị hỏng (Broken Access Control).
2. Lỗi mật mã (Cryptographic Failures).
3. Tiêm nhiễm (Injection).
4. Insecure Design: danh mục mới, tin tặc lợi dụng sai sót trong thiết kế để tấn công.
5. Cấu hình sai bảo mật (Security Misconfiguration).
6. Các thành phần dễ bị tổn hại và lỗi thời (Vulnerable and Outdated Components).
7. Nhận dạng và xác thực không thành công (Identification and Authentication Failures).
8. Software and Data Integrity Failures: lỗi đưa ra các giả định liên quan đến cập nhật phần mềm, dữ liệu quan trọng mà không xác minh tính toàn vẹn.
9. Security Logging and Monitoring Failures.
10. Yêu cầu phía máy chủ giả mạo (Server-Side Request Forgery).

WATERFALL MODEL (MÔ HÌNH THÁC NƯỚC)

1. Requirements

2. Design

2.1 Threat Modeling : Mô hình hóa hiểm họa

- 4 stages:

+ Diagramming : What are we working on?

+ Threat enumeration : What can go wrong?

+ Mitigation : What are we going to do about it?

+ Verification: Did we do a good job?

\* Threat enumeration:

- S : Spoofing

- T : Tampering

- R : Repudiation

- I : Information Disclosure

- D : Denial of service

- E : Elevation of Privilege

==> STRIDE là một mô hình hiểm họa, được sử dụng để xác định các hiểm họa đối với một hệ thống

- Kết quả của việc threat modeling là danh mục các hiểm họa và cách thức xử lý các hiểm họa

2.2 Security Design Pattern : Mẫu thiết kế an toàn

Mẫu an toàn là một mẫu mô tả cách thức ngăn ngừa một hiểm họa bằng việc sử dụng các cơ chế an toàn

VD: RBAC

3. Development

+ Secure Coding : Lập trình an toàn

4. Testing

+ Testing for Security : Kiểm thử an toàn

5. Deployment : Triển khai an toàn

6. Maintenance

Inline : Style

Internal : <Style> ở head

External : <link> ở head

ID : #tenid

Class: .tenclass

HTML

Gạch ngang: <s>

Mũ trên: <sup>

Mũ dưới: <sub>

Ngắt dòng: <br>

Đoạn văn: <p>

blank &nbsp; &#160;

< &lt; &#60;

> &gt; &#62;

& &amp; &#38;

hiển thị ảnh: <img src="...">

audio/video: <embed src="..." >

đường kẻ ngang: <hr>

Danh sách có thứ tự số: <ol><li>

Ds k thứ tự: <ul><li>

Ds định nghĩa: <dl><dt><dd> ---k có - hoặc chấm đen, dt mới có ------

Siêu liên kết: <a href=...></a>

Liên kết:

- Tạo điểm đánh dấu: <a name="bookmark1"><a>

- Tạo liên kết đến điểm đánh dấu: <a href="#name">Chuyển đến điểm đánh dấu</a?

Bảng:

<table>

<tr>---bản ghi---

<td> tên cột </td>

</tr>

</table>

Đường viền: ---đường viền 2 nét----

<table border="1" borderColor="red">...</table>

Đường viền 1 nét:

thêm style="BORDER-COLLAPSE: collapse; align="center"

Trộn cột:

thêm <td colspan="số cột gộp">tên cột lớn</td>

Trộn hàng: <td rowspan=...>...</td>

Tạo điều khiển nút bấm: --button-- <input type="button"...>

Tạo checkbox: <input type="checkbutton"...>

Tạo radio: <input type="radio"...>tên

Tạo textbox: <input type="text"...>tên

Mật khẩu: <input type="password"...>

ô văn bản: <textarea ...>text</textarea>

dropdown đơn:

<select>

<option> 1 </option>

<option> 2 </option>

</select>

Dropdown đa:

<select name="tinh"multiple>

<option> 1 </option>

<option> 2 </option>

</select>

Label: <span>

Upload tệp:

File: <input type="file">

Phân đoạn: <div>

Khoanh vùng có nhãn:

<fieldset>

<legend>Thông tin cá nhân</legend>

</fieldset>

Sd frameset để xây dựng cấu trúc trang

1, chia màn hình thành 3 frames theo chiều ngang

2, Chia frame thứ 2 thành 2 frames theo chiều dọc

3, Đặt các trang vào các frames

HTML5:

- Giảm các plugin

- Xử lý lỗi tốt hơn

- Nhiều đánh dấu thay cho kịch bản

- Độc lập thiết bị

\*\*các đặc trưng mới:

- Đối tượng <canvas> cho đồ họa 2D

- <video><audio> cho media

- các điều khiển form mới như calendar, date, time, email, url, search

- hỗ trợ lưu trữ cục bộ

\*\*

+ Nhập màu: <input type = "color" ...>

+ Nhập ngày: date

+ Nhập giờ: time

+ Nhập ngày giờ: datetime

+ Nhập ngày giờ địa phương: datetime-local

+ Nhập tuần: week

+ Nhập tháng: month

+ Nhập email: email

+ Nhập số: number

+ Nhập giá trị trong khoảng: range

+ Nhập nội dung tìm kiếm: search

+ Nhập điện thoại: tel

+ Nhập URL: url

# **Câu 1: REGISTER**

<html>

<head>

<title>REGISTER</title>

<?php

$conn = new mysqli("localhost", "thieuhoa", "123", "quanly\_nv");

if ($conn->connect\_error) {

echo 'connect false!';

die("Connect fail ".$conn->connect\_error);

} else {

// echo 'connect successfully!';

}

function getRandomString($n)

{

$characters = '0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ';

$randomString = '';

for ($i = 0; $i < $n; $i++) {

$index = rand(0, strlen($characters) - 1);

$randomString .= $characters[$index];

}

return $randomString;

}

if(isset($\_POST['register']))

{

if($\_POST['username'] == null)

{

$check = false;

echo("Please enter your username!<br />");

}

else

{

$u = $\_POST['username'];

$check = true;

}

if($\_POST['email'] == null)

{

$check = false;

echo("Please enter your email!<br />");

}

else

{

$email = $\_POST['email'];

$check = true;

}

if($\_POST['phone'] == null)

{

$check = false;

echo("Please enter your phone!<br />");

}

else

{

$phone = $\_POST['phone'];

$check = true;

}

if($\_POST['password'] == null)

{

$check = false;

echo("Please enter your password!<br />");

}

else

{

$check = true;

$p = $\_POST['password'];

}

// Validate password strength

$uppercase = preg\_match('@[A-Z]@', $p);

$lowercase = preg\_match('@[a-z]@', $p);

$number = preg\_match('@[0-9]@', $p);

$specialChars = preg\_match('@[^\w]@', $p);

if(!$uppercase || !$lowercase || !$number || !$specialChars || strlen($p) < 8) {

$check = false;

echo ("Password should be at least 8 characters in length and should include at least one upper case letter, one number, and one special character");

}

if($check)

{

$salt = getRandomString(32);

$crypt = sha1($salt.$p);

$sql = "INSERT INTO `user`(`id`, `username`, `password`, `salt`, `email`, `phone`) VALUES ('','".$u."','".$crypt."','".$salt."','".$email."','".$phone."')";

if ($conn->query($sql) === TRUE) {

header("Location: login.php?registerSuccessfully");

} else {

echo $conn->error;

}

$conn->close();

}

}

?>

</head>

<body>

<form action="register.php" method="post">

username: <input type="text" name="username" size="25" /><br />

email: <input type="text" name="email" size="25" /><br />

phone: <input type="text" name="phone" size="25" /><br />

password: <input type="text" name="password" size="25" /><br />

<input type="submit" name="register" />

</form>

</body>

</html>

# **Câu 2: LOGIN**

<html>

<head>

<title>LOGIN</title>

<?php

if(isset($\_POST['login']))

{

if($\_POST['username'] == null)

{

echo("Please enter your username!<br />");

}

else

{

$u = $\_POST['username'];

}

if($\_POST['password'] == null)

{

echo("Please enter your password!<br />");

}

else

{

$p = $\_POST['username'];

}

if($u && $p)

{

$conn = mysql\_connect("localhost", "root", "123");

mysql\_select\_db("login", $conn);

$sql = "select \* from user where username= '" .$u. " ' and password=' " .$p." ' ";

$query = mysql\_query($sql);

if(mysql\_num\_rows($query) == 0)

{

echo("Username or password is not correct, please try again");

}

else

{

echo("Hello $u");

}

}

</head>

<body>

<form action="login.php" method="post">

username: <input type="text" name="Ten dang nhap" size="25" /><br />

password: <input type="text" name="Mat khau" size="25" /><br />

<input type="submit" name="Dang nhap">

</form>

</body>

</html>

# **Câu 3: QUẢN LÝ PHÂN QUYỀN**

//xác thực và phân quyền người dùng

<?php

$email = $\_POST['email']; //lấy email từ form đăng nhập

$password = $\_POST['password']; //lấy pass từ form đăng nhập

//hàm kết nối đb

$connect = mysqli\_connect('local','tk','mk','db');

$sql = "lấy email và pass trong bảng người dùng";

$result = mysqli\_query($connect,$sql);//trả kết quả

$number\_rows = mysqli\_num\_rows($result);//kiểm tra sự tồn tại của người dùng

//nếu có tk và mk

if ($number\_rows == 1) {

$permission = $\_SESSION['permision'];

// Kiểm tra quyền của người đó có phải là admin hay không

if ($permission == '0') {

// Nếu không phải admin thì xuất thông báo

echo "Bạn không đủ quyền truy cập vào trang này<br>";

echo "<a href='http://localhost/website/index.php'> Click để về lại trang chủ</a>";

exit();

}

//nếu không tồn tại tk chuyển hướng lại form login

header('location:signin.php');

# **Câu 4: QUẢN LÝ PHIÊN LÀM VIỆC**

<?php

$email = $\_POST['email']; //lấy email từ form đăng nhập

$password = $\_POST['password']; //lấy pass từ form đăng nhập

//hàm kiểm tra người dùng có ấn ghi nhớ đăng nhập hay không

if(isset(remember))

remember = true

else

remember = false

//hàm kết nối đb

$connect = mysqli\_connect('local','tk','mk','db');

$sql = "lấy email và pass trong bảng người dùng";

$result = mysqli\_query($connect,$sql);//trả kết quả

$number\_rows = mysqli\_num\_rows($result);//kiểm tra sự tồn tại của người dùng

//nếu có tk và mk

if ($number\_rows == 1) {

session\_start();//khởi tạo phiên

//từ kết quả lấy id người dùng

//phiên lưu lại id và tên người dùng

if(remember = true)

//tạo token & lưu token ghi nhớ đăng nhập

}

header('location:user.php');//chuyển hướng đến trang người dùng

exit;

//nếu không tồn tại tk chuyển hướng lại form login

header('location:signin.php')

# **Câu 5: CHANGE PASSWORD**

<?php

// kiểm tra xem đã đăng nhập chưa

if (!isset($\_SESSION["username"])) {

header("location:login.php");

}

//thông tin để kết nối tới database

$host = "localhost";

$user = "ntq";

$pass = '1009';

$db\_name = "db\_test";

// tạo PDO để chống tấn công SQL Injection

$dsn = "mysql:host=$host;dbname=$db\_name";

$con = new PDO($dsn, $user, $pass);

if (isset($\_POST['change\_password'])) {

$currentPass = $\_POST['currentPass'];

$newPass = $\_POST['newPass'];

$passwordConfirm = $\_POST['passwordConfirm'];

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

// kiểm tra hai dòng mật khẩu mới người dùng nhập có khớp hay không

if ($newPass != $passwordConfirm) {

header("location:changePass.php?status=not\_match");

}

//kiểm tra độ mạnh của mật khẩu mới để chống tấn công dò quét mật khẩu

if (preg\_match('@[A-Z]@', $newPass) || preg\_match('@[a-z]@', $newPass) || preg\_match('@[0-9]@', $newPass) || preg\_match('@[^\w]@', $newPass)|| strlen($password) < 8) {

header("location:changePass.php?status=weak\_pass");

}

// mã hoá mật khẩu trước khi lưu vào DB để nếu dữ liệu bị lộ thì mật khẩu của người dùng vẫn được bảo vệ

$options = array("cost" => 4);

$newPass = password\_hash($newPass, PASSWORD\_BCRYPT, $options);

$currentPass = password\_hash($currentPass, PASSWORD\_BCRYPT, $options);

$stm = $con->prepare("SELECT password FROM user WHERE username=?");

$stm->execute([$username]);

$row = $stm->fetch(PDO::FETCH\_ASSOC);

// kiểm tra độ chính xác của mật khẩu cũ rồi đổi mật khẩu.

if ($currentPass === $row['password']) {

$stm = $con->prepare("UPDATE user SET password = ? where username = ?;");

$stm->execute([$newPass, $username]);

header("location:changePass.php?status=success");

}else{

header("location:changePass.php?status=fail");

}

}

?>

# **Câu 6: Hàm tạo mã CSRF**

function generate\_csrf\_token() {

// Nếu mã CSRF chưa được tạo, tạo mới mã CSRF

if (!isset($\_SESSION['csrf\_token'])) {

// Sử dụng hàm random\_bytes() để tạo mã CSRF ngẫu nhiên

$csrf\_token = bin2hex(random\_bytes(32));

// Lưu trữ mã CSRF trong biến session

$\_SESSION['csrf\_token'] = $csrf\_token;

}

// Trả về mã CSRF

return $\_SESSION['csrf\_token'];

}

?>

# **Câu 7: Quản lí phiên đăng nhập mã hoá cookie**

