

Bài 8: Cơ chế xác thực, Ajax, jQuery

Giảng viên: Trần Thị Thu Phương

Email: tttphuong2@hnmu.edu.vn

Mobile: 0966 224784

Nội dung

1. Cơ chế xác thực
2. Ajax
3. JQuery
4. Hoàn thiện Quản lý nhân viên

1. Cơ chế xác thực

1.1 http authentication

Khái niệm

- HTTP Authentication là một cơ chế trong giao thức HTTP (Hypertext Transfer Protocol) được sử dụng để xác định và xác thực người dùng hoặc ứng dụng trước khi họ có quyền truy cập vào tài nguyên trên máy chủ web. Nó giúp bảo vệ các tài nguyên trực tuyến khỏi truy cập trái phép bằng cách yêu cầu người dùng cung cấp thông tin đăng nhập hoặc mã thông báo xác thực.

Phân loại

- Basic Authentication: Đây là phương thức đơn giản nhất, trong đó thông tin đăng nhập (tên người dùng và mật khẩu) được mã hóa và gửi trong tiêu đề của yêu cầu HTTP. Tuy nhiên, cơ chế này không an toàn vì thông tin xác thực không được mã hóa mạnh và có thể bị dễ dàng nắm bắt bởi ai đó nghe trộm dữ liệu qua mạng.
- Digest Authentication: Đây là một hình thức an toàn hơn so với Basic Authentication. Nó sử dụng mã hóa để bảo vệ thông tin xác thực. Thay vì gửi mật khẩu nguyên thủy, máy khách và máy chủ **tạo mã băm của mật khẩu và các thông tin khác để xác thực người dùng**. Tuy nhiên, nó vẫn có một số hạn chế và yêu cầu tính toán nhiều hơn.
- Bearer Token Authentication: Trong phương thức này, máy khách cung cấp một mã thông báo (token) xác thực đã được cấp cho họ. Mã thông báo này thường được sử dụng trong các ứng dụng web API và được truyền trong tiêu đề "Authorization" của yêu cầu HTTP. Mã thông báo có thể được mã hóa và ký số để đảm bảo tính toàn vẹn và bảo mật của nó.
- OAuth: OAuth là một giao thức phức tạp được sử dụng cho việc ủy quyền truy cập vào tài nguyên giữa các ứng dụng. Nó cho phép người dùng cấp quyền truy cập tài nguyên của họ cho các ứng dụng khác mà họ tin tưởng mà không cần chia sẻ mật khẩu.

Lưu ý: sử dụng == hay ===

- == (So sánh bằng):
 - Toán tử == được sử dụng để so sánh giá trị của hai biểu thức mà không quan tâm đến kiểu dữ liệu của chúng. Nếu giá trị của hai biểu thức bằng nhau, thì toán tử == trả về true, ngược lại trả về false.
 - javascript
- ```
5 == "5" // true, vì giá trị là 5 ở cả hai bên
```
- ```
5 == 6    // false, vì giá trị không bằng nhau
```

=== (So sánh đồng nhất):

Toán tử === được sử dụng để so sánh giá trị và kiểu dữ liệu của hai biểu thức. Nếu cả giá trị và kiểu dữ liệu của hai biểu thức đều giống nhau, thì toán tử === trả về true, ngược lại trả về false.

Ví dụ:

```
javascript
```

```
5 === "5" // false, vì kiểu dữ liệu khác nhau
```

```
5 === 5    // true, vì cả giá trị và kiểu dữ liệu đều giống nhau
```

Lưu username và password

- MySQL → nếu lưu username và password vào CSDL thì gặp nguy cơ gì?
- nếu hacker truy cập vào và biết rõ thông tin này.
- Giải pháp là gì?
- Sử dụng cơ chế mã hóa hàm 1 chiều: thuật toán sử dụng hàm băm MD5, tuy nhiên hiện nay giải pháp này chưa thực sự an toàn (dễ bị tấn công)

Lưu username và password

- *salt* the password: thêm một số ký tự vào password

Ví dụ `echo hash('ripemd128', 'saltstringmypassword');` → kết quả là hiển thị chuỗi:

9eb8eb0584f82e5d505489e6928741e7

Tuy nhiên có một số vấn đề nếu không tuân thủ cũng khiến điều này cũng không an toàn do:

Sử dụng salt có ngẫu nhiên không? Có cập nhật thường xuyên không..?

Lưu username và password

Giải pháp: password_hash function với tham số PASSWORD_DEFAULT: đảm bảo bảo mật và sinh ra chuỗi ngẫu nhiên.

Demo: test.php

Lưu ý: độ dài chuỗi mật khẩu sinh ra, do vậy có lưu ý khi tạo CSDL lưu trữ phải đủ lớn (tối thiểu 255)

Giải pháp: login.php: thiết lập kết nối CSDL

```
<?php // login.php
$host = 'localhost';
$data = 'b7_auth_http';
$user = 'root';    // Change as necessary
$pass = '';        // Change as necessary
$chrs = 'utf8mb4';
$attr = "mysql:host=$host;dbname=$data;charset=$chrs";
$opts =
[
    PDO::ATTR_ERRMODE          => PDO::ERRMODE_EXCEPTION,
    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
    PDO::ATTR_EMULATE_PREPARES  => false,
];
?>
```

Giải pháp: setupusers.php

```
<?php //setupusers.php
require_once 'login.php';

try
{
    $pdo = new PDO($attr, $user, $pass, $opts);
}
catch (\PDOException $e)
{
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}

$query = "CREATE TABLE users (
    forename VARCHAR(32) NOT NULL,
    surname VARCHAR(32) NOT NULL,
    username VARCHAR(32) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL
)";

$result = $pdo->query($query);

$forename = 'Bill';
$surname = 'Smith';
$username = 'bsmith';
$password = 'mysecret';
$hash = password_hash($password, PASSWORD_DEFAULT);

add_user($pdo, $forename, $surname, $username, $hash);

$forename = 'Pauline';
$surname = 'Jones';
```

```
$username = 'pjones';
$password = 'acrobat';
$hash = password_hash($password, PASSWORD_DEFAULT);

add_user($pdo, $forename, $surname, $username, $hash);

function add_user($pdo, $fn, $sn, $un, $pw)
{
    $stmt = $pdo->prepare('INSERT INTO users VALUES(?,?,?,?)');

    $stmt->bindParam(1, $fn, PDO::PARAM_STR, 32);
    $stmt->bindParam(2, $sn, PDO::PARAM_STR, 32);
    $stmt->bindParam(3, $un, PDO::PARAM_STR, 32);
    $stmt->bindParam(4, $pw, PDO::PARAM_STR, 255);

    $stmt->execute([$fn, $sn, $un, $pw]);
}
?>
```

Giải thích

- `password_hash($password, PASSWORD_DEFAULT)`: Đây là sử dụng hàm `password_hash()` để tạo giá trị băm từ mật khẩu `$password`. Hàm này thường nhận hai tham số:
 - `$password`: Đây là mật khẩu người dùng muốn tạo giá trị băm.
 - `PASSWORD_DEFAULT`: Đây là hằng số được sử dụng để xác định thuật toán băm mật khẩu. Trong trường hợp này, `PASSWORD_DEFAULT` sẽ sử dụng thuật toán băm mật khẩu được khuyến nghị mạnh nhất theo phiên bản PHP hiện tại. Hằng số này có thể thay đổi trong các phiên bản PHP mới để sử dụng các thuật toán băm mật khẩu mạnh hơn.

Giải pháp: tiến hành validate: authenticate.php

```
<?php // authenticate.php
require_once 'login.php';

try
{
    $pdo = new PDO($attr, $user, $pass, $opts);
}
catch (\PDOException $e)
{
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}

if (isset($_SERVER['PHP_AUTH_USER']) &&
    isset($_SERVER['PHP_AUTH_PW']))
{
    $un_temp = sanitize($pdo, $_SERVER['PHP_AUTH_USER']);
    $pw_temp = sanitize($pdo, $_SERVER['PHP_AUTH_PW']);
    $query    = "SELECT * FROM users WHERE username=$un_temp";
    $result   = $pdo->query($query);
```

```
    if (!$result->rowCount()) die("User not found");

    $row = $result->fetch();
    $fn  = $row['forename'];
    $sn  = $row['surname'];
    $un  = $row['username'];
    $pw  = $row['password'];

    if (password_verify(str_replace("'", "", $pw_temp), $pw))
        echo htmlspecialchars("$fn $sn : Hi $fn,
            you are now logged in as '$un'");
    else die("Invalid username/password combination");
}
else
{
    header('WWW-Authenticate: Basic realm="Restricted Area"');
    header('HTTP/1.1 401 Unauthorized');
    die("Please enter your username and password");
}

function sanitize($pdo, $str)
{
    $str = htmlentities($str);
    return $pdo->quote($str);
}

?>
```

Giải thích

- Lý do chúng ta sử dụng `password_hash()` là để bảo mật mật khẩu của người dùng trong cơ sở dữ liệu. Thay vì lưu trữ mật khẩu thô (plain text) trong cơ sở dữ liệu, chúng ta lưu trữ giá trị băm này. Khi người dùng đăng nhập, chúng ta có thể so sánh giá trị băm của mật khẩu mà họ nhập với giá trị băm trong cơ sở dữ liệu bằng cách sử dụng hàm `password_verify()`. Điều này giúp đảm bảo rằng mật khẩu của người dùng được bảo vệ và không thể dễ dàng bị lộ khi có sự xâm nhập vào cơ sở dữ liệu.

localhost/b7_auth_http/authenticate.php

ui học... 9 Web Technologie... Đào tạo thiết kế_Fe...

Sign in to access this site

Authorization required by http://localhost

Username

Password

Sign in

Cancel

1.2 Kết hợp với sessions

Sessions (phiên)

- Phiên là một cách để lưu trữ thông tin (trong các biến) sẽ được sử dụng trên nhiều trang.
- Không giống như cookie, thông tin **không** được lưu trữ trên máy tính của người dùng.
- Biến phiên 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 phiên kéo dài cho đến khi người dùng đóng trình duyệt**.
- Ví thế; Các biến phiên chứa thông tin về một người dùng và có sẵn cho tất cả các trang trong một ứng dụng.

Bật session

- `$_SESSION['variable'] = $value;`

`//lưu trữ lại giá trị session`

`$variable = $_SESSION['variable'];`

Sử dụng session đảm bảo xác minh người dùng có quyền truy cập vào một số chức năng được gán quyền ngay khi mở tab khác trên cùng trình duyệt.

Nếu mở trình duyệt khác, yêu cầu đăng nhập lại.

Authenticate2.php

```
<?php // authenticate2.php
require_once 'login.php';

try
{
    $pdo = new PDO($attr, $user, $pass, $opts);
}
catch (\PDOException $e)
{
    throw new \PDOException($e->getMessage(), (int)$e->getCode());
}

if (isset($_SERVER['PHP_AUTH_USER']) &&
    isset($_SERVER['PHP_AUTH_PW']))
{
    $un_temp = sanitize($pdo, $_SERVER['PHP_AUTH_USER']);
    $pw_temp = sanitize($pdo, $_SERVER['PHP_AUTH_PW']);
    $query = "SELECT * FROM users WHERE username=$un_temp";
    $result = $pdo->query($query);

    if (!$result->rowCount()) die("User not found");

    $row = $result->fetch();
    $fn = $row['forename'];
    $sn = $row['surname'];
    $un = $row['username'];
    $pw = $row['password'];

    if (password_verify(str_replace("'", "", $pw_temp), $pw))
    {
        session_start();

        $_SESSION['forename'] = $fn;
        $_SESSION['surname'] = $sn;

        echo htmlspecialchars("$fn $sn : Hi $fn,
            you are now logged in as '$un'");
        die("<p><a href='continue.php'>Click here to continue</a></p>");
    }
    else die("Invalid username/password combination");
}
else
```

```
{
    header('WWW-Authenticate: Basic realm="Restricted Area"');
    header('HTTP/1.0 401 Unauthorized');
    die ("Please enter your username and password");
}

function sanitize($pdo, $str)
{
    $str = htmlentities($str);
    return $pdo->quote($str);
}
?>
```

← → ↻ ⓘ localhost/b7_auth_http/authenticate2.php

🕒 Trang chủ - LuxLac ➕ thu chi quy khoa K... ➕ thu chi quy khoa C... ➕ thu

Pauline Jones : Hi Pauline, you are now logged in as 'pjones'

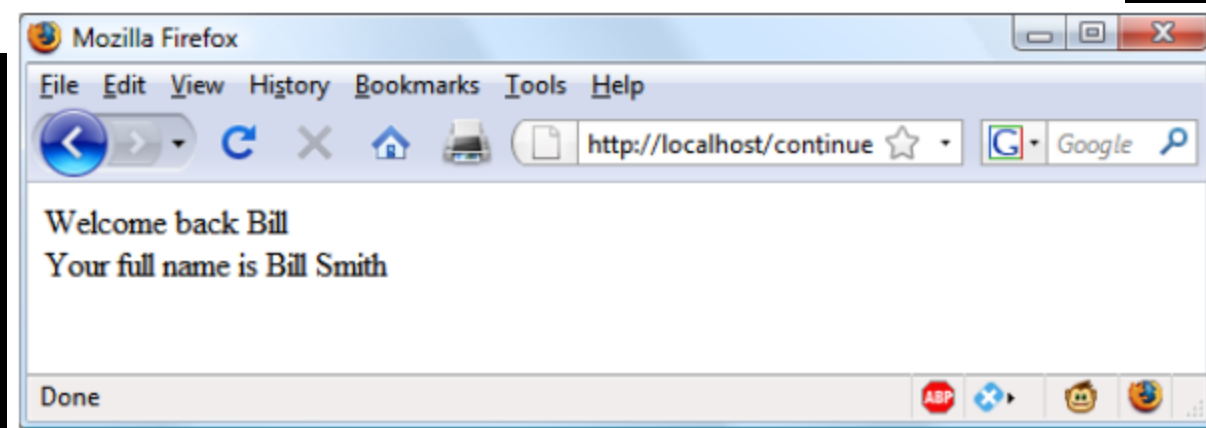
[Click here to continue](#)

Continue.php

```
<?php // continue.php
session_start();

if (isset($_SESSION['forename']))
{
    $forename = htmlspecialchars($_SESSION['forename']);
    $surname = htmlspecialchars($_SESSION['surname']);

    echo "Welcome back $forename.<br>
        Your full name is $forename $surname.<br>";
}
else echo "Please <a href='authenticate2.php'>click here</a> to log in.";
?>
```



- Sử dụng session để lưu thông tin đăng nhập. Thông tin này hiển thị lại khi chuyển trang (nhấp vào link)

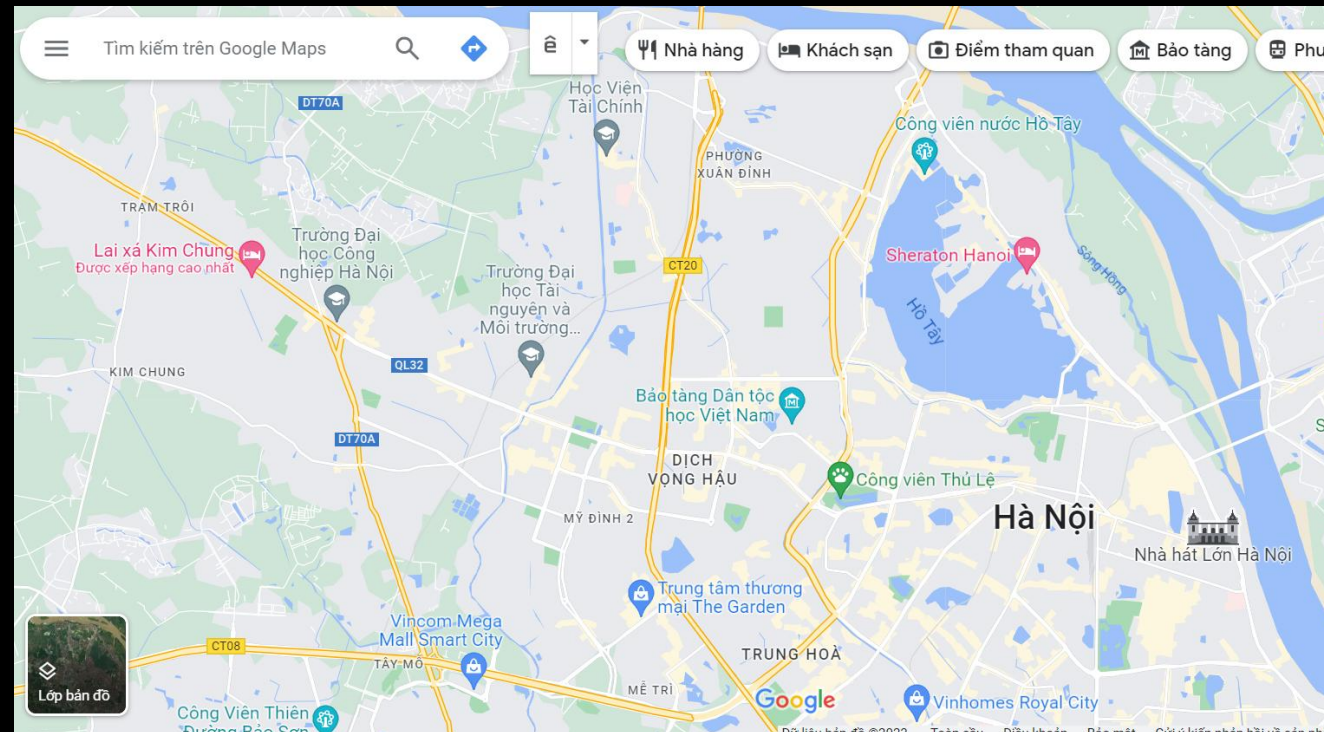
2. Ajax (Asynchronous JavaScript and XML)

Hiểu về ajax

- Ajax (Asynchronous JavaScript and XML) là một kỹ thuật trong lập trình web cho phép truyền dữ liệu giữa trình duyệt và máy chủ mà không cần phải tải lại trang web. Thay vì yêu cầu một trang web mới từ máy chủ, Ajax cho phép trình duyệt gửi yêu cầu HTTP và nhận dữ liệu từ máy chủ một cách bất đồng bộ (asynchronously), sau đó cập nhật nội dung trang web mà không cần làm mới trang.

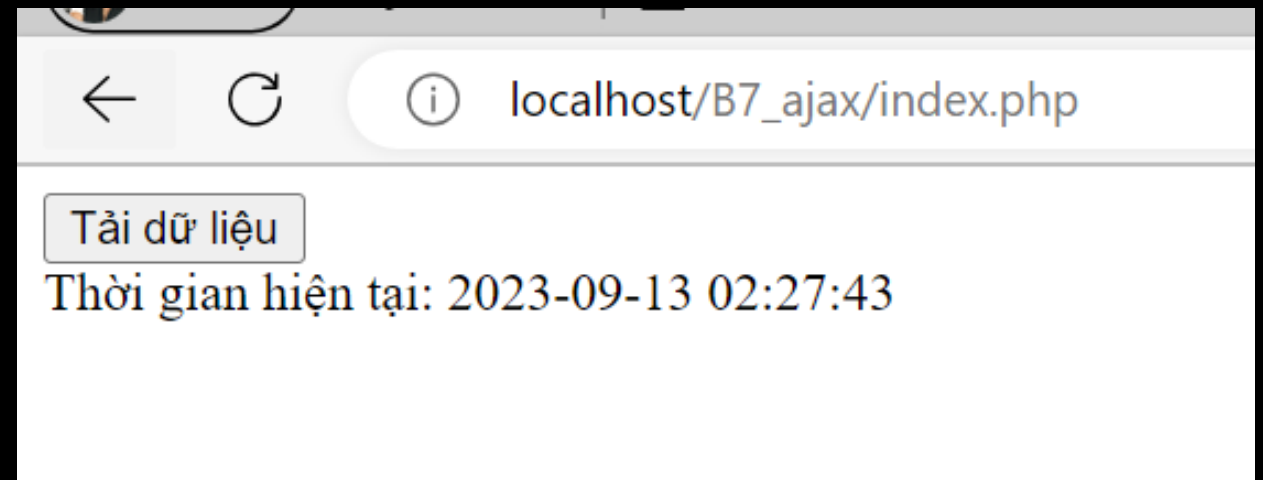
Ví dụ

- Một ví dụ điển hình của công nghệ này là Google Maps trong đó các phần mới của bản đồ được tải xuống từ máy chủ khi cần mà không cần làm mới trang.



Ví dụ

- Khi người dùng nhấn vào tải dữ liệu, thời gian hiện tại sẽ hiển thị mà không cần load lại trang



Index.php

- Thư viện jQuery để dễ dàng thực hiện Ajax

```
<title>Ajax Demo</title>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
    $(document).ready(function() {
        $("#load-data-button").click(function() {
            $.ajax({
                url: "getData.php", // Tên tập tin PHP xử lý dữ liệu
                type: "GET",
                dataType: "text",
                success: function(response) {
                    $("#result").html(response); // Cập nhật div với dữ liệu từ máy chủ
                },
                error: function() {
                    alert("Đã xảy ra lỗi trong quá trình tải dữ liệu.");
                }
            });
        });
    });
</script>
</head>
<body>
    <button id="load-data-button">Tải dữ liệu</button>
    <div id="result">Kết quả sẽ xuất hiện ở đây</div>
</body>
</html>
```

Giải thích

1. Thư viện jQuery để thực hiện Ajax..
2. Khi người dùng nhấn vào nút "Tải Dữ Liệu," sự kiện click sẽ được kích hoạt.
3. Trong hàm xử lý sự kiện, `$.ajax()` để gửi một yêu cầu Ajax đến máy chủ. Các thông số chính trong hàm `$.ajax()` là:
 - url: Đây là đường dẫn đến tệp xử lý máy chủ, trong ví dụ này là "getData.php".
 - type: Phương thức HTTP được sử dụng, ở đây là "GET"
 - success: Hàm này sẽ được gọi khi yêu cầu Ajax thành công. Chúng ta cập nhật nội dung của thẻ div có id "data" với dữ liệu trả về từ máy chủ.
 - error: Hàm này sẽ được gọi khi có lỗi xảy ra trong quá trình yêu cầu Ajax.

3. JQuery

Hiểu về JQuery

- Mục đích của jquery để sử dụng javascript dễ dàng và nhanh chóng.
- JQuery bao gồm:
 - HTML/DOM manipulation
 - CSS manipulation
 - HTML event methods
 - Effects and animations
 - AJAX
 - Utilities
- Học jquery: <https://www.w3schools.com/jquery/default.asp>

Ví dụ

- Khi người dùng nhấn add classes to elements, hệ thống thêm class cho text

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

This is some important text!

Add classes to elements

Heading 1

Heading 2

This is a paragraph.

This is another paragraph.

This is some important text!

Add classes to elements

Index.php

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>
5 <script>
6 $(document).ready(function(){
7     $("button").click(function(){
8         $("h1, h2, p").addClass("blue");
9         $("div").addClass("important");
10    });
11 });
12 </script>
13 <style>
14 .important {
15     font-weight: bold;
16     font-size: xx-large;
17 }
18
19 .blue {
20     color: blue;
21 }
22 </style>
23 </head>
24
25 <body>
26
27 <h1>Heading 1</h1>
28 <h2>Heading 2</h2>
29
30 <p>This is a paragraph.</p>
31 <p>This is another paragraph.</p>
32
33 <div>This is some important text!</div><br>
34
35 <button>Add classes to elements</button>
36
37 </body>
38 </html>
```

Project: Hoàn thiện quản lý nhân viên

Nội dung

- Đăng nhập
- Phân quyền người dùng
- Quản trị người dùng và nhân viên
- Thoát

Tiến trình

- Hệ thống yêu cầu đăng nhập để bắt đầu sử dụng
- Hệ thống xác thực người dùng
 - Nếu quyền admin thì hiển thị trang quản trị
 - Nếu quyền employees, hiển thị trang cho nhân viên
 - Hệ thống từ chối tất cả các phương thức nhập địa chỉ trực tiếp để truy cập các chức năng dành riêng (phải đăng nhập)
- Logout

Đăng Nhập

Tên đăng nhập :	<input type="text"/>
Mật khẩu :	<input type="password"/>
<input type="button" value="Đăng Nhập"/>	

Bạn đăng nhập chưa đúng Click vào đây để đăng nhập [Login](#)

- Nếu nhập sai tên đăng nhập hoặc mật khẩu, yêu cầu đăng nhập lại
- Nếu đăng nhập thành công và vai trò quản trị

Đây là trang dành cho quản trị

xin chào phuongttt

[Thoát](#)
[Home](#) [Quản trị nhân viên](#)
[Quản trị người dùng](#)

Đăng Nhập

Tên đăng nhập :	<input type="text"/>
Mật khẩu :	<input type="password"/>
<input type="button" value="Đăng Nhập"/>	

Bạn đăng nhập chưa đúngClick vào đây để đăng nhập [Login](#)

- Nếu nhập sai tên đăng nhập hoặc mật khẩu, yêu cầu đăng nhập lại
- Nếu đăng nhập thành công và vai trò nhân viên

xin chào phuongnhi123Đây là trang cho nhân viên [Thoát](#) [Home](#)

Danh sách người dùng (admin- bổ sung, sửa, xóa)

Danh sách người dùng

[Thêm người dùng](#)

[Admin Home](#)

User ID	Tên đăng nhập	Mật khẩu	Vai trò
4	ductrung243	\$2y\$10\$Zz1Hb1ddweKYCPA7n1v3kOoMHCIerldjimlx170MALyXHQVPBOySO	employees
1	phuongttt	\$2y\$10\$uOkCURnQDwqfSIg3aDE5geeRQHcSCoc89b/s4sXGliQIdO0yvQBku	admin
14	phuongnhi281	\$2y\$10\$gywz3v1OrVtbbr1GAhkVyOWuFW2q11.VvjMdI5u9Gj7PUBBP0Y6P.	employees
5	phuongnhi123	\$2y\$10\$i569WKIoAc0GJ30h7JSSp.x1dcYW3vu.FPZd8eOi1..9JMbOAXgPq	employees

Thêm người dùng










Thêm người dùng

[Trở về](#)

User name	<input type="text"/>
password	<input type="password"/>
Vai trò	<input type="text" value="admin"/> ▼
Nhân viên	<input type="text" value="1"/> ▼
<input type="button" value="Lưu"/>	

Giải pháp

- Bổ sung bảng user lưu trữ thông tin người dùng, username, password, vai trò; bảng roles: quyền người dùng
- Xây dựng các hàm (lấy thông tin người dùng, thêm người dùng, sửa, xóa)
- Login
- Cơ chế xác thực (http authentication)
- Phân quyền: session
- Logout: session

Options				user_id	username	password	role_name
<input type="checkbox"/>	 Edit	 Copy	 Delete	4	duc trung243	\$2y\$10\$Zz1Hb1ddweKYCPA7n1v3kOoMHCIerldjimlxl70MALy...	employees
<input type="checkbox"/>	 Edit	 Copy	 Delete	1	phuongttt	\$2y\$10\$uOkCURnQDwqfSlg3aDE5geeRQHcSCoc89b/s4sXGliQ...	admin
<input type="checkbox"/>	 Edit	 Copy	 Delete	14	phuongnhi281	\$2y\$10\$gywz3v1OrVtbbr1GAhkVyOWuFW2q11.VvjMdl5u9Gj7...	employees
<input type="checkbox"/>	 Edit	 Copy	 Delete	5	phuongnhi123	\$2y\$10\$i569WKloAc0GJ30h7JSSp.x1dcYW3vu.FPZd8eOi1.....	employees

- Thực tế: username: tự sinh hoặc địa chỉ email (cần validate là duy nhất)
- Role name: lấy từ CSDL (thêm bảng roles)
- Mã hóa mật khẩu
- Thiết lập mật khẩu ban đầu, cho phép người dùng đổi mật khẩu