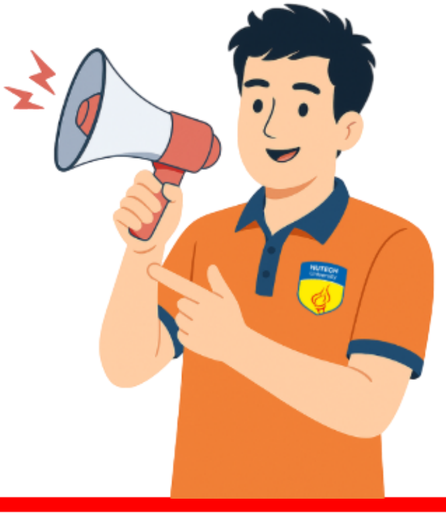
TÀI LIỆU THỰC HÀNH
PHÁT TRIỂN PHẦN MỀM MÃ NGUỒN MỞ

GV: DƯƠNG QUỐC NAM

BÀI 4: XÂY DỰNG CHỨC NĂNG XÁC THỰC NGƯỜI DÙNG

MỤC TIÊU CỦA BÀI THỰC HÀNH

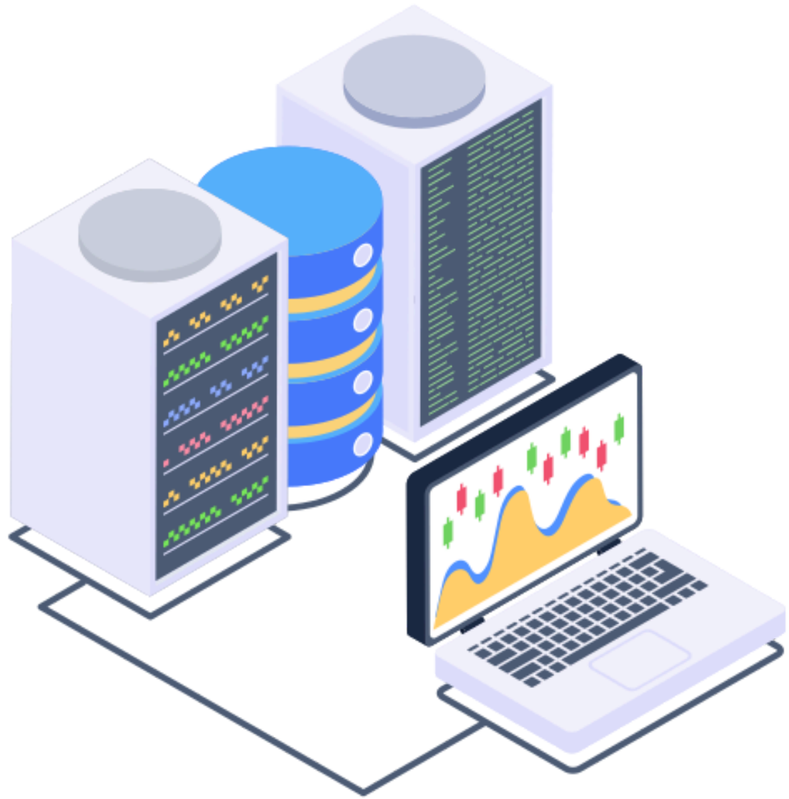
- Hiểu và xử lý cơ chế đăng nhập.
- Tạo phân quyền người dùng (Admin, Khách hàng).
- Thực hành bảo vệ chức năng quản trị bằng quyền hạn.



TỔNG QUAN VỀ SESSION

Session là gì?

- Là vùng lưu trữ tạm trên máy chủ cho từng người dùng.
- Dữ liệu vẫn còn giữ nguyên khi chuyển giữa các trang.
- Mỗi người dùng có một session riêng biệt.



Ưu điểm khi dùng Session

- Không cần cài đặt cơ sở dữ liệu.
- Lưu trữ đơn giản theo mảng.
- Tự động lưu trên server theo phiên làm việc.

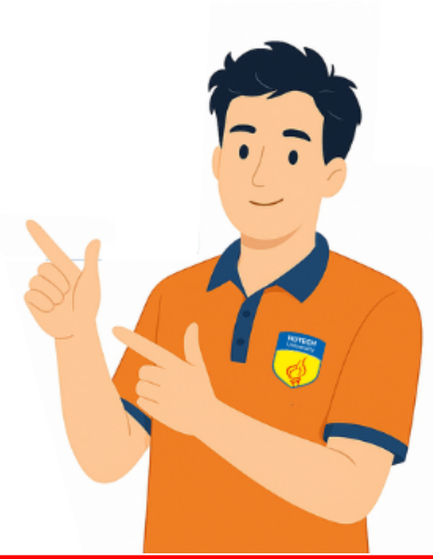
Lưu ý khi dùng Session

- Không phù hợp để lưu dữ liệu lâu dài.
- Dữ liệu sẽ mất khi đóng trình duyệt hoặc session hết hạn.
- Không dùng session để lưu dữ liệu lớn, nhạy cảm hoặc bảo mật cao.

TỔNG QUAN VỀ SESSION



Criteria	Local Storage	Session Storage	Cookies
Storage Capacity	5-10 mb	5-10 mb	4 kb
Auto Expiry	No	Yes	Yes
Server Side Accessibility	No	No	Yes
Data Transfer HTTP Request	No	No	Yes
Data Persistence	Till manually deleted	Till browser tab is closed	As per expiry TTL set



MÃ HÓA MẬT KHẨU KHI ĐĂNG NHẬP

Tại sao cần mã hóa mật khẩu?

- Bảo vệ dữ liệu người dùng nếu bị lộ database.
- Ngăn chặn rò rỉ thông tin khi bị tấn công.
- Không bao giờ lưu mật khẩu "*thô*" trong cơ sở dữ liệu.

Ví dụ sử dụng password_hash() trong PHP

```php
<?php
$password = $_POST['password'];
$hash = password_hash($password, PASSWORD_DEFAULT);
// Lưu $hash vào DB

```

So sánh mật khẩu khi đăng nhập

```php
<?php
$input = $_POST['password'];
if (password_verify($input, $user['password'])) {
  // Mật khẩu đúng
}

```

CƠ CHẾ PHÂN QUYỀN NGƯỜI DÙNG

Phân quyền là gì?

- Kiểm soát ai được truy cập vào tính năng nào.
- Tránh người dùng không hợp lệ truy cập trang quản trị.
- Bảo vệ dữ liệu và chức năng quan trọng.

Các nhóm quyền

- Khách hàng: Xem sản phẩm, thêm giỏ hàng, mua hàng
- Quản trị viên: Quản lý sản phẩm, người dùng, đơn hàng

Lưu ý quan trọng

- Luôn kiểm tra session và role ở đầu trang cần bảo vệ.
- Tách giao diện và chức năng riêng cho mỗi loại người dùng.

Phân quyền khi đăng nhập

```php
<?php
$_SESSION['user'] = [
  'id' => $row['id'],
  'username' => $row['username'],
  'role' => $row['role']
];

```

```sql
-- Tạo bảng account
CREATE TABLE account (
   id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(255) NOT NULL UNIQUE,
   password VARCHAR(255) NOT NULL,
   role ENUM('admin', 'user') DEFAULT 'user',
   created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

CHUYỂN HƯỚNG (REDIRECT) SAU ĐĂNG NHẬP

Tại sao cần redirect sau đăng nhập?

- Tránh ở lại trang đăng nhập sau khi đăng nhập thành công.
- Đưa người dùng đến đúng nơi phù hợp với vai trò:
  - Người dùng → trang chủ.
  - Admin → trang quản trị.

```php
<?php
header("Location: /trang_dich.php");
exit;

// Lưu ý: exit; là bắt buộc để dừng script.
```

Cách redirect trong PHP

```php
<?php
if ($_SESSION['user']['role'] == 1) {
  header("Location: /admin/dashboard.php");
} else {
  header("Location: /index.php");
}
exit;
```

Ví dụ chuyển hướng theo role

THIẾT KẾ CSDL – BẢNG USERS

```sql
-- Tạo bảng account
CREATE TABLE account (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(255) NOT NULL UNIQUE,
    password VARCHAR(255) NOT NULL,
    role ENUM('admin', 'user') DEFAULT 'user',
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Dùng để lưu trữ thông tin khách hàng:

- id: khoá chính.
- name: tên đăng nhập.
- password: mật khẩu đã mã hoá.
- role: quyền người dùng (user = khách, admin = quản trị viên).

FORM ĐĂNG KÝ NGƯỜI DÙNG

Chức năng chính của form đăng ký

- Thu thập thông tin người dùng:
  - username
  - password
  - confirm password (*xác nhận lại mật khẩu*)
- Kiểm tra hợp lệ dữ liệu.
- Lưu thông tin vào database.

Bảo mật cần lưu ý

- Sử dụng password_hash() thay vì md5.
- Kiểm tra đầu vào để tránh SQL injection.

```php
<?php
if ($_POST['password'] == $_POST['confirm_password']) {
  $hash = password_hash($_POST['password'], PASSWORD_DEFAULT);
  // INSERT INTO users (username, password) ...
} else {
  echo "Mật khẩu không khớp!";
}

```

❌ Lỗi thường gặp

- Không kiểm tra xác nhận mật khẩu.
- Không mã hóa mật khẩu trước khi lưu.
- Không kiểm tra username đã tồn tại.

FORM ĐĂNG NHẬP NGƯỜI DÙNG

Chức năng chính của form đăng nhập

- Thu thập username và password từ người dùng.
- So sánh với thông tin trong database.
- Nếu đúng → lưu session và chuyển hướng.
- Nếu sai → hiển thị thông báo lỗi.

Lưu ý bảo mật

- Dùng password_verify() để so sánh mật khẩu.
- Không echo lỗi cụ thể như: "*Sai mật khẩu*".
- Đảm bảo đã có session_start() ở đầu file.

```php
<?php
$username = $_POST['username'];
$password = $_POST['password'];

$sql = "SELECT * FROM users WHERE username = '$username'";
$result = mysqli_query($conn, $sql);
$user = mysqli_fetch_assoc($result);

if (password_verify($password, $user['password'])) {
  $_SESSION['user'] = $user;
  header("Location: index.php");
} else {
  echo "Sai tài khoản hoặc mật khẩu!";
}

```

Chuyển hướng sau khi đăng nhập →

Admin → /admin/dashboard

User → trang chủ /index.php

PHÂN QUYỀN SAU ĐĂNG NHẬP

Mục tiêu của phân quyền

- Giao quyền truy cập khác nhau cho từng loại người dùng:
  - Người dùng bình thường (User).
  - Quản trị viên (Admin).
- Đảm bảo bảo mật và đúng luồng chức năng.

```php
<?php
if ($_SESSION['user']['role'] == 1) {
  header("Location: /admin/dashboard.php");
} else {
  header("Location: /index.php");
}
exit;
```

Chuyển hướng theo vai trò

```php
<?php
if (!isset($_SESSION['user']) || $_SESSION['user']['role'] != 1) {
  header("Location: /unauthorized.php");
  exit;
}
```

Kiểm tra phân quyền trước khi truy cập

CHẶN TRUY CẬP TRÁI PHÉP

Vì sao cần chặn truy cập trái phép?

- Ngăn người dùng chưa đăng nhập truy cập tài nguyên riêng.
- Bảo vệ chức năng chỉ dành cho Admin.
- Tránh lộ thông tin hoặc thao tác sai đối tượng.

```php
<?php
session_start();
// Đảm bảo người dùng đã đăng nhập mới được tiếp tục
if (!isset($_SESSION['user'])) {
  header("Location: /Auth/login.php");
  exit;
}

```

Kiểm tra đăng nhập trước khi vào trang riêng

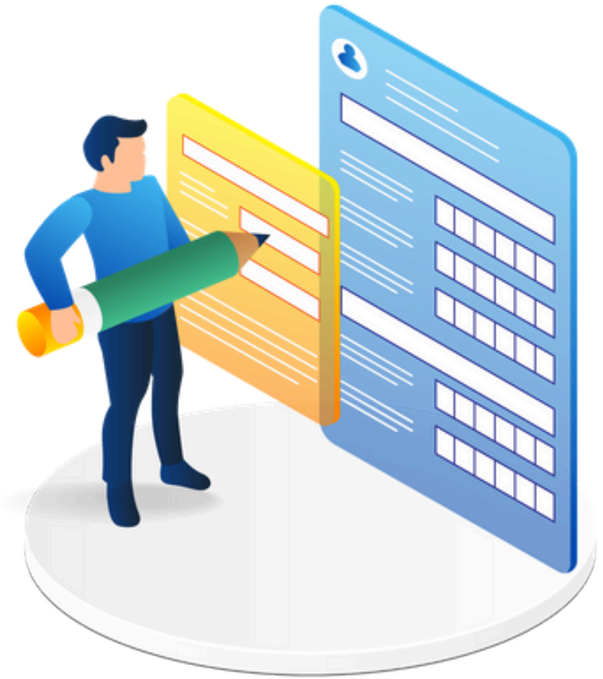
```php
<?php
// Chỉ cho phép role = 1 (admin) truy cập vào các trang quản trị

if ($_SESSION['user']['role'] != 1) {
  header("Location: /unauthorized.php");
  exit;
}

```

Chặn quyền truy cập nếu không phải Admin

QUÊN SESSION_START() - KHÔNG THỂ LƯU TRẠNG THÁI ĐĂNG NHẬP

Tại sao cần session_start()?

- PHP cần session_start() để bắt đầu quản lý phiên làm việc
- Nếu không gọi hàm này:
  - Không thể dùng $_SESSION
  - Không lưu được thông tin người dùng đăng nhập



Lưu ý khi sử dụng session:

- Gọi session_start() ở mọi file cần dùng $_SESSION
- Chỉ gọi một lần duy nhất trên mỗi file
- Không echo, print, HTML trước nó

KHÔNG KIỂM TRA ISSET($_SESSION['USER'])

Vấn đề là gì?

- Nếu không kiểm tra $_SESSION['user'] trước khi truy cập trang riêng.
- Người chưa đăng nhập vẫn có thể truy cập chức năng hạn chế.
- Gây rò rỉ quyền → lộ thông tin, thao tác sai người.

```php
<?php

session_start();

if (!isset($_SESSION['user'])) {
  header("Location: /Auth/login.php");
  exit;
}
```

Sau đó kiểm tra quyền

```php
<?php

if ($_SESSION['user']['role'] != 1) {
  header("Location: /unauthorized.php");
}

```

CÁCH HIỂN THỊ TỪNG GIAO DIỆN

Chức năng	Đường dẫn
Đăng ký	http://localhost:<port>/<ten_du_an>/account/register
Đăng nhập	http://localhost:<port>/<ten_du_an>/account/login
Đăng xuất	http://localhost:<port>/<ten_du_an>/account/logout
Đổi mật khẩu	http://localhost:<port>/<ten_du_an>/account/change-password