Đại học Quốc gia thành phố Hồ Chí Minh
Trường Đại học Công nghệ Thông tin
Khoa mạng máy tính và truyền thông
Bộ môn An toàn thông tin

4

Lab

PHỤC VỤ MỤC ĐÍCH GIÁO DỤC
FOR EDUCATIONAL PURPOSE ONLY

Web Back-end

Thực hành môn Lập trình ứng dụng Web

Tháng 4/2025

Lưu hành nội bộ

<Nghiêm cấm đăng tải trên internet dưới mọi hình thức>

A. TỔNG QUAN

1. Mục tiêu

- Làm quen với các cấu trúc cơ bản của PHP.
- Sử dụng PHP để xử lý yêu cầu từ client.
- Xây dựng các chức năng đơn giản với AJAX, Session, Cookies và localStorage.

2. Thời gian thực hành

- Thực hành tại lớp: **5** tiết tại phòng thực hành.
- Hoàn thành báo cáo kết quả thực hành: tối đa **13** ngày.

3. Liên quan

- Sinh viên cần nắm các kiến thức nền tảng về HTML, CSS, JavaScript và PHP. Các kiến thức này đã được giới thiệu trong nội dung lý thuyết đã học do đó sẽ không được trình bày lại trong nội dung thực hành này.
- Tham khảo thêm tại website w3school.com để nắm vững kiến thức.

B. CHUẨN BỊ MÔI TRƯỜNG

- Cài đặt trình soạn thảo Visual Studio Code hoặc các trình soạn thảo tương tự.
- Cài đặt trình duyệt web (Chrome, Firefox, Edge,...) để quan sát kết quả.
- Cài đặt PHP: sử dụng gói tích hợp XAMPP hoặc cài đặt riêng PHP

  Lưu ý: Để kiểm tra cài đặt PHP, mở terminal và gõ lệnh php -v. Nếu hiển thị phiên bản PHP, cấu hình đã thành công.

```
C:\Users\NGAN>php -v
PHP 8.2.12 (cli) (built: Oct 24 2023 21:15:15) (ZTS Visual C++ 2019 x64)
Copyright (c) The PHP Group
Zend Engine v4.2.12, Copyright (c) Zend Technologies
```

  Nếu command không hoạt động, cần kiểm tra cài đặt biến môi trường (environment variable).

C. THỰC HÀNH

1. Giới thiệu PHP

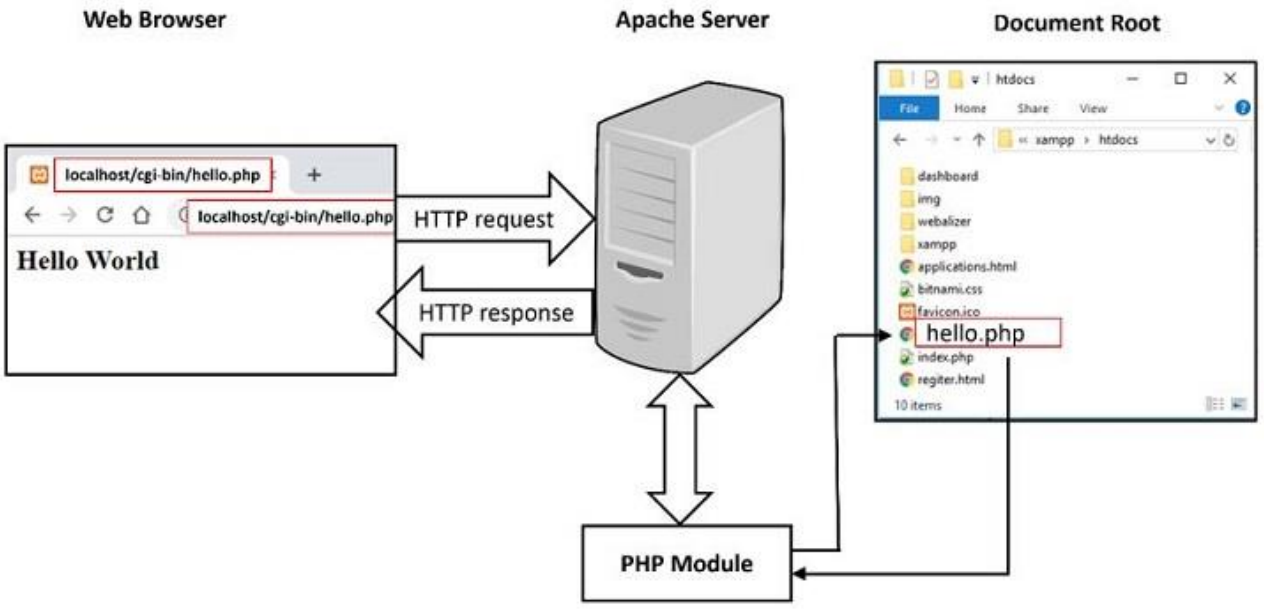
PHP là gì?

PHP (Hypertext Preprocessor) là ngôn ngữ lập trình kịch bản chạy trên máy server, được sử dụng chủ yếu để phát triển ứng dụng web. HTML có vai trò là Markup của các trang web "tĩnh", còn PHP được thiết kế cho các trang web "động". PHP có thể nhúng trực tiếp vào HTML và được sử dụng rộng rãi trên hầu hết các web server như Apache, Nginx.

PHP trong mô hình web

Ứng dụng Web được đặt trên máy chủ (HTTP Server) có cài đặt PHP. Khi đó, trình duyệt đóng vai trò là một HTTP client và giao tiếp với máy chủ thông qua giao thức HTTP. Quá trình hoạt động gồm:

1. Trình duyệt gửi yêu cầu (HTTP request) đến địa chỉ của file PHP.
2. Server xử lí mã PHP, sinh ra kết quả HTML
3. Server trả kết quả HTML về cho trình duyệt.
4. Trình duyệt hiển thị nội dung.



Thành phần	Công nghệ	Vai trò
Client-side	HTML, CSS, JavaScript, Bootstrap, React	Thiết kế giao diện và xử lí tương tác người dùng
Server-side	PHP, Python, Django, Node.js, Express.js	Xử lý logic, kết nối dữ liệu, tạo nội dung động
Cơ sở dữ liệu	MySQL, SQLite, MongoDB	Lưu trữ và xử lí truy vấn dữ liệu

Cú pháp PHP

PHP có thể nhúng trong file HTML, nằm giữa cặp thẻ <?php và ?>

Bước 1: File firstPHP.php có nội dung sau

```
<!DOCTYPE html>
<html lang="en">
<body>
<h1>My first PHP page</h1>

<?php
echo "Hello World!";
// Comment 1
# Comment 2
/*
   Comment 3 cho
   nhiều dòng
```

```
*/
?>
</body>
</html>
```

Bước 2: Khởi chạy web server bằng PHP trên localhost. Mở terminal và di chuyển đến thư mục chứa file PHP, chạy command:

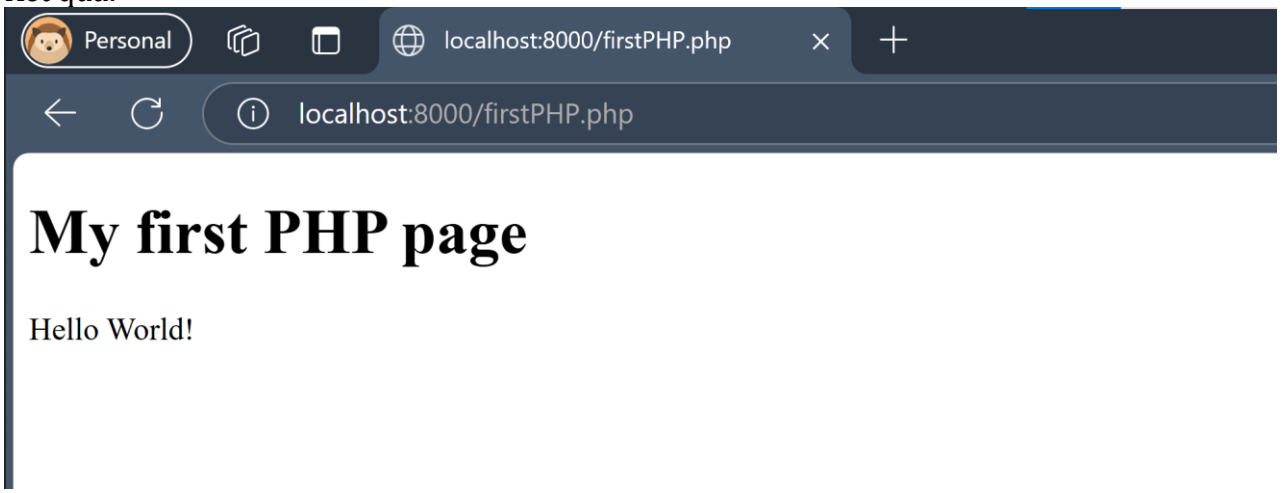
```
php -S localhost:8000
```

Ghi chú: `localhost:8000` là địa chỉ và cổng để chạy server

Bước 3: Mở trình duyệt và gõ đường dẫn tới file PHP.

```
http://localhost:8000/firstPHP.php
```

Kết quả:



Task 1: Setup thành công môi trường cho php, khởi chạy server tại localhost bằng web server tích hợp có sẵn trong PHP (ví dụ minh họa trên) hoặc bằng XAMPP (Apache) hoặc các extenstion tùy chọn.

2. Luyện tập với PHP

a) Xử lí form

Trong PHP, biến bắt đầu với kí hiệu **$**.

```
<?php
$hoTen = "Nguyễn Văn A";
echo "Chào bạn $hoTen";
$tuoi = 20;
$diem = 8.5;

$isPass = true;
if ($isPass) {
    echo "Bạn đã qua môn";
} else {
    echo "Bạn đã rớt môn";
}

$mang = array("NT208", "NT521", "NT513");
echo $mang[0]; //NT208
```

```php
/* Associative Array */
$sinhVien = array(
    "ten" => "A",
    "tuoi" => 20
);
echo $sinhVien["ten"]; //A

isset($sinhVien); //true
isset($sv); //false
?>
```

`isset($bien)` để kiểm tra biến có tồn tại và không phải **NULL**.

$_GET và $_POST: mảng siêu toàn cục của PHP

- $_GET là một mảng các biến nhận dữ liệu thông qua các tham số URL. Thông tin được gửi từ form bằng method GET có thể nhìn thấy được (tất cả tên biến và giá trị đều được hiển thị trong URL).
- $_POST là một mảng các biến nhận dữ liệu từ HTTP request có POST method, thường được sử dụng để xử lý dữ liệu từ form. Thông tin được gửi từ form bằng method POST sẽ không hiển thị lên URL (tất cả tên/giá trị đều được nhúng trong nội dung của HTTP request).

Ví dụ minh họa method GET:

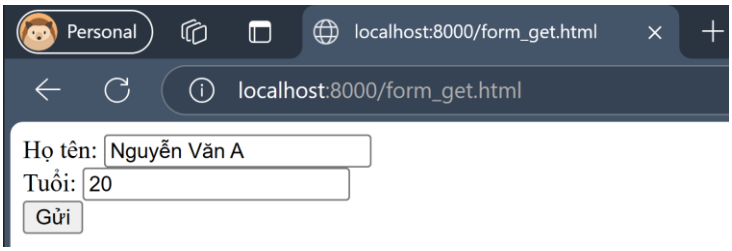
File *form_get.html*

```html
<form action="process_get.php" method="get">
  <label>Họ tên:</label>
  <input type="text" name="hoten"><br>
  <label>Tuổi:</label>
  <input type="number" name="tuoi"><br>
  <input type="submit" value="Gửi">
</form>
```

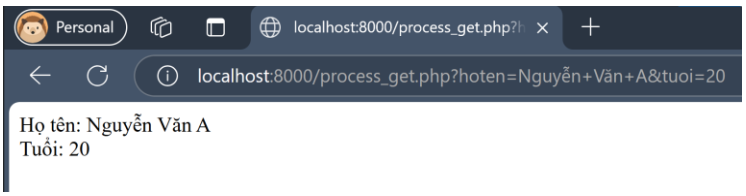
File *process_get.php*

```php
<?php
$hoten = $_GET['hoten'];
$tuoi = $_GET['tuoi'];

echo "Họ tên: " . $hoten . "<br>";
echo "Tuổi: " . $tuoi;
?>
```



Kết quả: sau khi submit, điều hướng với URL có chứa tham số và giá trị đã nhập từ form.



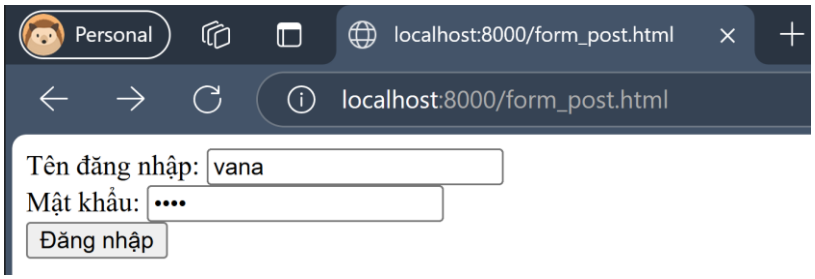
Ví dụ minh họa method POST:
File *form_post.html*

```
<form action="process_post.php" method="post">
    <label>Tên đăng nhập:</label>
    <input type="text" name="username"><br>
    <label>Mật khẩu:</label>
    <input type="password" name="password"><br>
    <input type="submit" value="Đăng nhập">
</form>
```

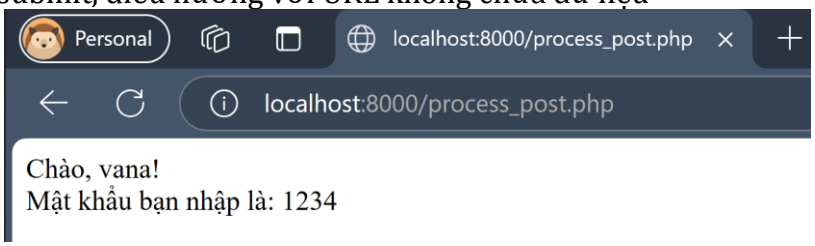
File *process_post.php*

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

echo "Chào, " . $username . "!<br>";
echo "Mật khẩu bạn nhập là: " . $password;
?>
```



Kết quả: sau khi submit, điều hướng với URL không chứa dữ liệu



Sanitizing và Validating

Làm sạch và xác thực dữ liệu đầu vào là bước quan trọng để bảo vệ ứng dụng web khỏi các lỗi và lỗ hổng bảo mật như XSS, SQL Injection, v.v. PHP cung cấp nhiều hàm đơn giản và hiệu quả để hỗ trợ xử lí.

- **Sanitizing**

Hàm	Vai trò
htmlspecialchars()	Chuyển các ký tự HTML thành dạng an toàn (chống XSS), xem thêm
strip_tags()	Loại bỏ toàn bộ thẻ HTML và PHP, xem thêm

trim()	Xóa khoảng trắng ở đầu và cuối chuỗi
filter_var($data, FILTER_SANITIZE_*)	Làm sạch dữ liệu theo loại cụ thể (email, URL, string...), xem thêm

- **Validating**

Hàm	Vai trò
filter_var($data, FILTER_VALIDATE_*)	Kiểm tra định dạng dữ liệu theo loại cụ thể (email, URL, string...), xem thêm
is_numeric()	Kiểm tra có phải số hay không
ctype_alpha()	Kiểm tra chuỗi chỉ chứa ký tự chữ không
ctype_digit()	Kiểm tra chuỗi chỉ chứa ký tự số không

Task 2: Tạo form với method là POST để đăng kí mua vé xem film với các trường thông tin sau:

1) Input Họ và tên.
2) Input Ngày sinh, tính tuổi hiện tại để thông báo người dùng có đủ tuổi tham gia không.
3) Chọn film theo danh sách film.
4) Chọn hạng vé và giá tiền.
5) Chọn số lượng vé.
6) Nút submit.
7) Sau khi submit, điều hướng tới trang mới Thông tin vé đã đặt gồm Lời chào + tên người đặt, nếu người dùng đủ tuổi, hiển thị tên phim, loại vé, số lượng đã đặt và tổng tiền. Dùng method GET để lấy dữ liệu.
8) Có làm sạch và xác thực dữ liệu đầu vào bằng các hàm được giới thiệu ở trên.

b) Cookies, sessions

Cookie là một file nhỏ được lưu trữ trên trình duyệt của user nhằm nhận biết user khi truy cập vào một trang web. Nó ghi nhớ những thông tin như tên đăng nhập, mật khẩu, các tuỳ chọn do user lựa chọn.
Khác với dữ liệu gửi từ form (POST hay GET) thì cookies sẽ được trình duyệt tự động gửi đi theo mỗi lần truy cập lên server và có thể bị thay đổi giá trị. Theo mặc định, cookie tồn tại đến khi hết phiên (đóng trình duyệt), tuy nhiên ta có thể thiết lập thời gian tồn tại cho nó. Ví dụ như chế độ ghi nhớ ID và Password.

Lưu ý: Hàm setcookie() phải đặt TRƯỚC thẻ <html>.

```php
<?php
/* Tạo cookie */
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + 3600); // tồn tại 1 tiếng

/* Đọc cookie */
echo $_COOKIE['username'];

/* Xóa cookie */
setcookie("username", "", time() - 3600);
?>
```

Session - phiên làm việc, là cách đơn giản để lưu trữ 1 biến và khiến biến đó có thể tồn tại từ trang này sang trang khác. Do đó, session là khu vực lưu trữ dữ liệu trên server, được liên kết với người dùng thông qua một ID duy nhất (session ID), dùng để quản lý đăng nhập, quyền truy cập, hoặc lưu thông tin tạm thời.
Về chi tiết hoạt động:

1) Dữ liệu session (như tên, email, trạng thái đăng nhập...) được lưu trên server và mỗi user được gán một Session ID duy nhất.
2) PHP tự động tạo một cookie có tên là PHPSESSID và gửi về trình duyệt.
3) Trình duyệt lưu cookie `PHPSESSID=abc123xyz`
4) Mỗi khi gửi request lên server, trình duyệt tự đính kèm cookie này trong HTTP header.
5) Lúc này, PHP kiểm tra cookie PHPSESSID, nếu có thì tìm session tương ứng trên server. Nếu không, tạo mới session.

Task 3: Viết trang web đăng nhập cơ bản với 2 loại tài khoản admin và user, nếu đúng tài khoản thì chuyển hướng đến dashboard, nếu sai thì thông báo lỗi. Yêu cầu tính năng:

1) Session-based login: Tự động đăng nhập lại nếu người dùng đã login (session vẫn còn).
2) Remember me: Nếu người dùng tích chọn Remember me thì tạo cookie để ghi nhớ người dùng. Khi quay lại trang sau khi đóng trình duyệt (session kết thúc) → tự động đăng nhập lại bằng cookie.
3) Session timeout: session hết hạn sau 60s không hoạt động, nếu không Remember me thì phải đăng nhập lại sau timeout.
4) Giao diện Light/Dark: Người dùng chọn theme và ghi nhớ lựa chọn đó vào cookie.
5) Phân quyền: Nếu người dùng là admin thì dashboard hiện thị nội dung đặc biệt (Secret content). Nếu người dùng là user thì giao diện thường và hiển thị số lượt truy cập.
6) Đếm lượt truy cập: Tạo cookie để lưu số lần user truy cập trang dashboard.
7) Dashboard: Hiển thị lời chào tương ứng với tài khoản, nút chọn giao diện Light/Dark, nút đăng xuất (không còn tự động đăng nhập lại).
8) Đổi tên hiện thị: Sau khi đăng nhập thành công, người dùng có thể đổi tên của mình, tên này được lưu trong session và thay đổi lại lời chào trên dashboard với tên mới

Cộng điểm: Thiết kế giao diện với css hoặc Boostrap

c) AJAX - Asynchronous JavaScript and XML

AJAX là kĩ thuật về việc cập nhật các phần của trang web mà không cần tải lại toàn bộ trang, qua đó giúp các trang web trở nên tương tác hơn, phản hồi nhanh hơn và có trải nghiệm người dùng mượt mà hơn.

Gửi và nhận dữ liệu với `fetch()` (AJAX đơn giản). File .html:

```
<body>
  <h2>Nhập nội dung bất kỳ:</h2>
  <input type="text" id="inputText">
  <button id="sendBtn">Gửi</button>
  <div id="result"></div>
  <script>
    // Gán sự kiện khi click vào nút "Gửi"
    document.getElementById("sendBtn").addEventListener("click", () => {

      // Lấy nội dung người dùng nhập
      const text = document.getElementById("inputText").value.trim();
```

```
      // Gửi yêu cầu GET đến ajax.php, kèm tham số q=
      fetch("ajax.php?q=" + encodeURIComponent(text))
        .then(res => res.text()) // Lấy phản hồi dạng text từ server
        .then(data => {
          document.getElementById("result").textContent = data;
        });
    });
  </script>
</body>
```

File .php

```
<?php
// Lấy nội dung gửi từ client qua phương thức GET
$q = $_GET["q"] ?? "";

echo "Bạn vừa nhập: " . htmlspecialchars($q);
?>
```

Task 4: Viết chương trình Tự động gợi ý (auto-suggestion) khi người dùng nhập liệu. Khi người dùng gõ từng kí tự của tên nghệ sĩ yêu thích vào ô tìm kiếm, AJAX gửi chuỗi đã nhập đến suggest.php để tìm những nghệ sĩ phù hợp (tìm theo từ khoá) và trả về danh sách. Sau đó, giao diện hiển thị danh sách gợi ý bên dưới ô tìm kiếm. Cụ thể:

a) Sử dụng JavaScript (`fetch`) để gửi yêu cầu tới server trong khi người dùng đang gõ.
b) PHP đọc danh sách nghệ sĩ định sẵn từ file, hàm `file()`
c) Trả về danh sách phù hợp dưới dạng danh sách HTML.
d) Không reload trang.

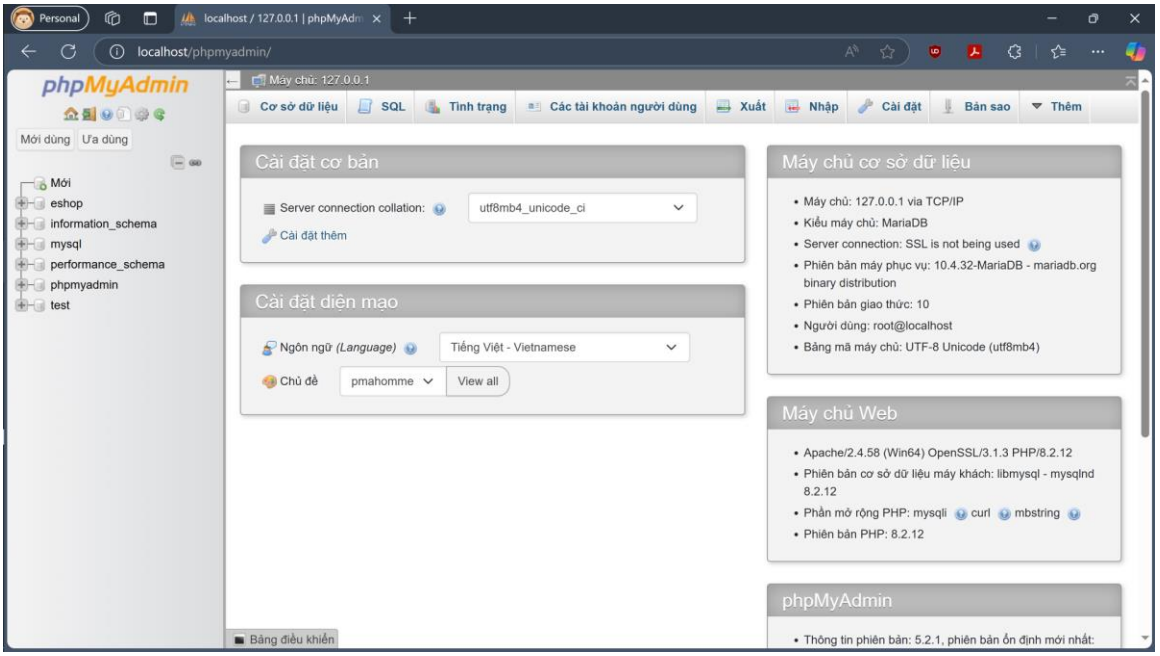
3. Bài tập về nhà

d) Cơ sở dữ liệu

Cơ sở dữ liệu (Database) là hệ thống lưu trữ thông tin có tổ chức, cho phép truy vấn, thêm, sửa, xóa và quản lý dữ liệu. Một số cơ sở dữ liệu phổ biến gồm:

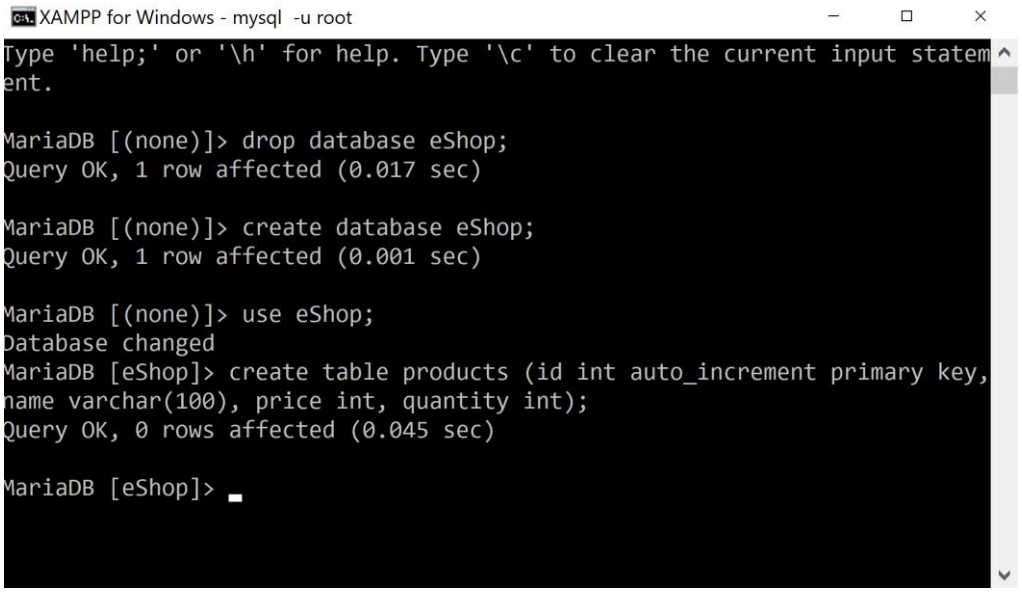
- SQL: MySQL, PostgreSQL, SQLite
- NoSQL: MongoDB, Redis

Trong XAMPP, cơ sở dữ liệu được tích hợp sẵn là MySQL / MariaDB, với phpMyAdmin là giao diện web để quản lý CSDL.



Luyện tập với CSDL: Cải tiến cho trang web Thương mại điện tử ở các lab trước, liên kết với CSDL để lưu dữ liệu và bổ sung các tính năng sau:

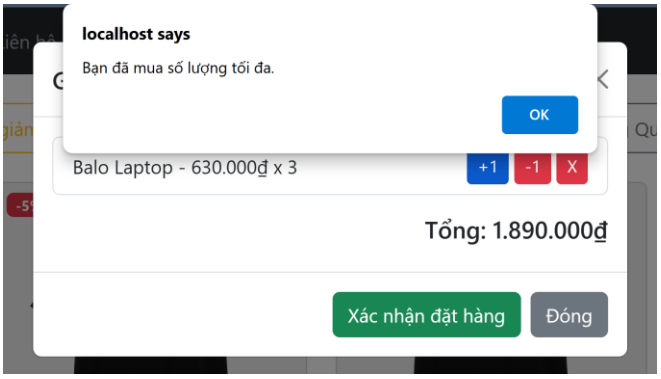
1. **Tạo bảng** và lưu dữ liệu về sản phẩm(tên, giá tiền, số lượng,...)



Hình: Minh họa tạo bảng trong MariaDB XAMPP

- Tại giao diện website, hiển thị số lượng sản phẩm còn lại ở mỗi mặt hàng, báo hết hàng nếu số lượng về 0 và không cho người dùng đặt hàng.

- Tại giỏ hàng, giới hạn số lượng sản phẩm người dùng được chọn mua theo số lượng còn lại trong database. Cảnh báo nếu người dùng mua vượt quá số lượng.



- Sau khi người dùng đặt hàng thành công, trừ số lượng tương ứng vào database và hiển thị lại đúng số lượng trên thẻ sản phẩm.

2. **Tạo bảng** lưu tài khoản người dùng và triển khai các chức năng Đăng kí, đăng nhập, đăng xuất cho website. Mật khẩu được lưu trong database phải dùng hàm băm.

	id	username	password_hash
bỏ	1	user1	$2y$10$PvwPGOw5Xy..CcxJeca.E.HgiWK3dWB4Lu1bzx6R/ix...
bỏ	2	user2	$2y$10$iLAvN7xhOzmlNm.2k5KOZOo6WUusNbKvPlx432/p1Jy...

3. Cập nhật thông tin đơn hàng (user, mã đơn hàng, tên mặt hàng, số lượng,...) vào database sau khi mua hàng thành công.

Task 5: Hoàn thành các yêu cầu nêu trên cho website Thương mại điện tử, sử dụng CSDL tích hợp sẵn trong XAMPP. Cần giải thích các bước thực hiện, các đoạn code quan trọng để hoàn thành yêu cầu. Chụp kết quả minh chứng của giao diện website và database khi thử nghiệm các tính năng. Cộng điểm: sinh viên tự triển khai các CSDL tùy chọn khác, kết nối đến web server.

D. YÊU CẦU & ĐÁNH GIÁ

- Sinh viên tìm hiểu và thực hành theo hướng dẫn, thực hiện theo nhóm đã đăng ký.
- Nộp báo cáo kết quả gồm chi tiết những việc (Report) mà nhóm đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Báo cáo:
  - File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
  - Đặt tên theo định dạng: [Mã lớp]-LabX_NhomY.
  - Ví dụ: [NT208.P12.ANTT.1]-Lab1_Nhom1.
  - Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
  - Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, ... sẽ được xử lý tùy mức độ vi phạm.

HẾT

Chúc các bạn hoàn thành tốt!