**TRƯỜNG ĐẠI HỌC LÂM NGHIỆP**

**KHOA CƠ ĐIỆN VÀ CÔNG TRÌNH**

****

BÁO CÁO ĐỒ ÁN

**Đề tài** : quản lý cửa hàng bán sách online

Họ và tên : Nguyễn Thị Huyền MSV : 2041070562 lớp k65 HTTT

Họ và tên : Lê Minh Tuyến MSV : 2041070416 lớp k65 HTTT

Họ và tên : Đinh Trung Nam MSV : 2041070562 lớp k65 HTTT

**Hà Nội, 2023**Phần I. Lý thuyết

1.Tóm tắt kiến thức HTML

HTML (HyperText Markup Language) là một ngôn ngữ đánh dấu được sử dụng để tạo ra các trang web. Nó xác định cách các phần tử và nội dung trong trang sẽ được hiển thị trên trình duyệt. Cú pháp của HTML dựa trên các thẻ (tag) và thuộc tính (attribute), được sắp xếp vào trong một tài liệu HTML.

Một trang HTML bắt đầu với thẻ đầu tiên là <!DOCTYPE html>, định nghĩa phiên bản của HTML mà tài liệu tuân thủ. Tiếp theo là thẻ <html>, định nghĩa vùng chứa toàn bộ nội dung trang web. Trong thẻ <html>, chúng ta sẽ tìm thấy thẻ <head> và thẻ <body>.

Thẻ <head> chứa các thông tin meta về trang web, chẳng hạn như tiêu đề, tài liệu css, các tập tin JavaScript và các thông tin khác không hiển thị trực tiếp trên trình duyệt. Thẻ <body> chứa nội dung hiển thị trên trình duyệt, bao gồm đoạn văn bản, hình ảnh, liên kết, bảng, video và các phần tử khác.

Các thẻ HTML có thể được nhóm thành các nhóm chức năng như sau:

1.1. Thẻ văn bản:

- <h1> đến <h6>: Định dạng tiêu đề.

- <p>: Định dạng đoạn văn bản.

- <b>: In đậm nội dung.

- <i>: In nghiêng nội dung.

1.2. Thẻ hình ảnh:

- <img>: Hiển thị hình ảnh trên trang web.

1.3. Thẻ liên kết:

- <a>: Liên kết đến trang web khác.

1.4. Thẻ danh sách:

- <ul>: Danh sách không có thứ tự.

- <ol>: Danh sách có thứ tự.

1.5. Thẻ bảng:

- <table>: Tạo bảng với các hàng và cột.

1.6. Thẻ biểu đồ:

- <canvas>: Vẽ biểu đồ và đồ họa trên trang web.

1.7. Thẻ đa phương tiện:

- <video>: Hiển thị video.

- <audio>: Hiển thị âm thanh.

1.8. Thẻ biểu mẫu:

- <form>: Tạo biểu mẫu để người dùng nhập liệu.

Ngoài ra, HTML cũng hỗ trợ sự tạo ra các khối và hình dạng phức tạp hơn bằng cách sử dụng CSS (Cascading Style Sheets) và JavaScript. CSS được sử dụng để xác định giao diện và kiểu dáng của các phần tử HTML, trong khi JavaScript được sử dụng để thêm các hiệu ứng động và tương tác vào trang web.

Tóm tắt kiến thức CSS

CSS (Cascading Style Sheets) là một ngôn ngữ đánh dấu được sử dụng để định dạng và trình bày nội dung của một trang web. Nó là một phần quan trọng của frontend web development và được sử dụng để tạo ra các trang web có giao diện đẹp, linh hoạt và tương tác.

CSS áp dụng nguyên tắc "Cascading" để xác định cách định dạng được áp dụng cho một phần tử HTML. Điều này có nghĩa là CSS cho phép các quy tắc định dạng được áp dụng từ nhiều nguồn khác nhau và ưu tiên quy tắc có độ ưu tiên cao hơn. CSS cũng giúp tách biệt khối HTML khỏi quy tắc định dạng, cho phép việc bảo trì và chỉnh sửa trang web trở nên dễ dàng hơn.

Các quy tắc CSS được áp dụng cho các phần tử HTML thông qua các bộ chọn. Bộ chọn cho phép chúng ta chọn các phần tử HTML mà chúng ta muốn áp dụng quy tắc CSS. Ví dụ, chúng ta có thể chọn phần tử bằng cách sử dụng tên lớp, ID, hoặc các yếu tố khác.

Quy tắc CSS bao gồm một bộ thuộc tính giá trị, trong đó mỗi thuộc tính xác định cách một phần tử HTML được định dạng. Ví dụ, thuộc tính "color" xác định màu chữ của một phần tử, "font-size" xác định kích thước chữ, và "background-color" xác định màu nền của một phần tử.

Cung cấp một ví dụ cụ thể, đoạn mã CSS dưới đây định dạng các đoạn văn bản có ID "paragraph" trong một trang web:

```

#paragraph {

color: blue;

font-size: 18px;

text-align: center;

}

```

Trong đoạn mã trên, "paragraph" là một ID và được chọn bằng cách sử dụng ký hiệu "#". Các thuộc tính được áp dụng vào đoạn văn bản với ID "paragraph" bao gồm màu chữ là màu xanh lá cây, kích thước chữ là 18 pixel, và canh giữa.

CSS cũng hỗ trợ hàng loạt thuộc tính cho các phần tử HTML khác nhau, cho phép chúng ta điều chỉnh bố cục, màu sắc, lề, độ trong suốt, hiệu ứng chuyển động và nhiều hơn nữa. Điều này giúp tạo ra các trang web hấp dẫn và cá nhân hóa.

Ngoài ra, CSS còn hỗ trợ cả việc tạo các hiệu ứng tương tác, chẳng hạn như hiển thị/huỷ bỏ hiển thị, hoạt động chuyển động, thay đổi kích thước, và thậm chí là việc tạo ra các hiệu ứng 3D. Điều này mở ra các khả năng thiết kế mạnh mẽ và sáng tạo cho các nhà phát triển web.

Cuối cùng, CSS cũng cho phép chúng ta sử dụng các kỹ thuật tiên tiến như responsive design để tạo ra các trang web linh hoạt và tương

Tóm tắt kiến thức JS

JavaScript (JS) là một ngôn ngữ lập trình phía client-side được sử dụng để tạo ra các hiệu ứng động và tương tác trên trang web. Ngôn ngữ này có thể được nhúng trực tiếp vào mã HTML hoặc được viết trong các tệp .js riêng biệt và được liên kết với mã HTML.

JS được phát triển bởi Netscape vào năm 1995 và ban đầu được thiết kế để tạo ra các kịch bản đơn giản cho việc xử lý sự kiện trên trình duyệt. Tuy nhiên, JS đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới và được sử dụng rộng rãi không chỉ trên web mà còn trên các máy tính cá nhân và các ứng dụng di động.

JS có cú pháp giống với những ngôn ngữ lập trình khác như C, C++, Java, với các loại biến, toán tử, hàm và cấu trúc điều khiển. Điểm đặc biệt của JS là nó là một ngôn ngữ dựa trên sự kiện (event-driven) và không đồng bộ (asynchronous). Điều này có nghĩa là JS có khả năng phản hồi và xử lý các sự kiện người dùng mà không cần tải lại toàn bộ trang web, tạo nên trải nghiệm mượt mà và tương tác trực tiếp với người dùng.

JS cung cấp nhiều tính năng mạnh mẽ cho phát triển web, bao gồm:

3.1. DOM manipulation: JS cho phép thay đổi nội dung, thuộc tính và kiểu dáng của các phần tử trên trang web. Điều này cho phép tạo ra các hiệu ứng động và thay đổi giao diện người dùng dễ dàng.

3.2. Xử lý sự kiện: JS có khả năng lắng nghe và xử lý các sự kiện người dùng như nhấp chuột, gõ phím, kéo thả và cuộn chuột. Điều này cho phép tạo ra các tương tác người dùng phong phú và phản hồi đáng tin cậy.

3.3. AJAX: JS hỗ trợ việc tải dữ liệu từ máy chủ mà không cần tải lại toàn bộ trang web. Điều này cho phép tạo ra các ứng dụng web động và liên tục với dữ liệu mới.

3.4. JSON: JavaScript Object Notation (JSON) là một định dạng dữ liệu rất phổ biến được sử dụng để truyền tải dữ liệu giữa trình duyệt và máy chủ. JS cung cấp các phương thức để phân tích và tạo ra JSON dễ dàng.

3.5. Thư viện và framework: JS có nhiều thư viện và framework phổ biến như jQuery, React, Angular, Vue.js,... giúp tăng cường hiệu suất phát triển và cung cấp các tính năng mạnh mẽ cho các ứng dụng web.

3.6. Một số khái niệm khác: JS hỗ trợ các khái niệm như closures, callbacks, promises và async/await để xử lý bất đồng bộ dễ dàng và hiệu quả.

Tóm lại, JavaScript là một ngôn ngữ lập trình phía client-side mạnh mẽ và linh hoạt, có thể sử dụng để xử lý sự kiện, tạo hiệu ứng động và tương tác trực tiếp với người dung

4.Tóm tắt kiến thức PHP

PHP là một ngôn ngữ lập trình phía máy chủ mạnh mẽ và phổ biến được sử dụng để xây dựng các ứng dụng web động. PHP là viết tắt của Personal Home Page Tools và hiện tại được coi là viết tắt của PHP: Hypertext Preprocessor.

4.1. Phát triển PHP:

- PHP được phát triển vào những năm 1994 bởi Rasmus Lerdorf. Ban đầu, PHP có nghĩa là "Personal Home Page", được tạo ra để quản lý và theo dõi cách mà viết lại CV của Rasmus Lerdorf. Trong quá trình phát triển, PHP đã trở thành một ngôn ngữ lập trình mạnh mẽ với cú pháp tương tự C và C++.

- Phiên bản đầu tiên của PHP (PHP/FI 1.0) chạy trên các máy chủ Unix và hỗ trợ nhúng mã HTML trong các tệp PHP. Sau đó, phiên bản 2.0 đã được phát hành với nhiều tính năng mới như hỗ trợ kết nối cơ sở dữ liệu MySQL. PHP 3 đã xuất hiện vào năm 1998 với một máy ảo Zend Engine mới, và phiên bản 4 được phát hành vào năm 2000 với nhiều cải tiến và khả năng quản lý tốt hơn.

4.2. Đặc điểm của PHP:

- Đa nền tảng: PHP có thể chạy trên hầu hết các hệ điều hành như Windows, Linux, macOS và còn nhiều nền tảng khác nhau.

- Dễ học và dễ sử dụng: PHP có cú pháp dễ hiểu, tương tự như C và C++. Nó cũng có nhiều tài liệu học hỏi và cộng đồng lớn, giúp người mới bắt đầu dễ dàng tiếp cận và nắm bắt ngôn ngữ này.

- Tích hợp tốt với HTML: PHP có khả năng nhúng mã PHP vào trong các tệp HTML, điều này giúp người phát triển tạo ra các trang web động dễ dàng.

4.3. Cú pháp PHP:

- Mã PHP được đặt trong cặp dấu "<?php ... ?>".

- Cú pháp PHP giống với các ngôn ngữ lập trình khác, bao gồm biểu thức toán học, câu lệnh điều kiện và vòng lặp.

4.4. Ví dụ về mã PHP:

- HelloWorld:

```php

<?php

echo "Hello World!";

?>

```

- Tạo biến:

```php

<?php

$name = "John Doe";

$age = 25;

echo "My name is " . $name . " and I am " . $age . " years old.";

?>

```

4.5. Các tính năng chính của PHP:

- Hỗ trợ kết nối cơ sở dữ liệu: PHP hỗ trợ kết nối và tương tác với các cơ sở dữ liệu như MySQL, PostgreSQL, SQLite và nhiều hơn nữa.

- Tạo các trang web động: PHP cho phép tạo ra các trang web có nội dung thay đổi dựa trên yêu cầu và dữ liệu người dùng.

- Xử lý biểu mẫu: PHP có thể xử lý dữ liệu được gửi từ biểu mẫu HTML và thực hiện các hành động như lưu dữ liệu vào cơ sở dữ liệu hoặc gửi dữ liệu qua email.

- Tạo và xử lý tệp: PHP cho phép tạo, đọc, ghi và xóa các tệp trên máy chủ.

Phần II: Ứng dụng

Phân tích CSDL

Thông tin về CSDL:

- Số lượng bảng: 5 bảng gồm:

+ Hệ quản trị CSDL sử dụng: Mysql v5.1.

Bảng book chứa thông tin về sách bao gồm các trường

- id: Mã quyển sách, mỗi quyển sách có mã duy nhất.

- title: Thông tin tiêu đề của quyển sách.

- author: Thông tin về tác giả của sách

- price: Thông tin về giá của quyển sách.

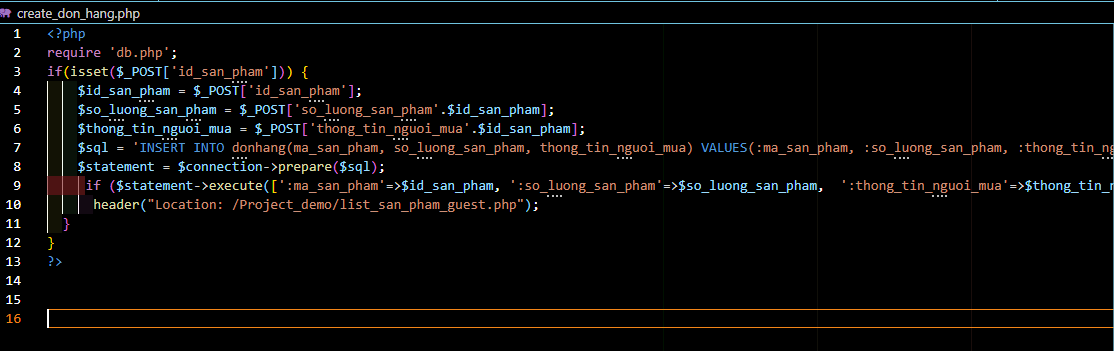
- tom tat: Nội dung tóm tắt của quyển sách.

- nxb: Tên nhà xuất bản cuốn sách.

- kichthuoc: Kích thước cuốn sách

- namxb: Năm cuốn sách xuất bản.

- Madm: Mã danh mục cuốn sách.



Đầu tiên, mã yêu cầu tệp ‘db.php’, tệp này có thể chứa thông tin về cơ sở dữ liệu và cách kết nối với nó.

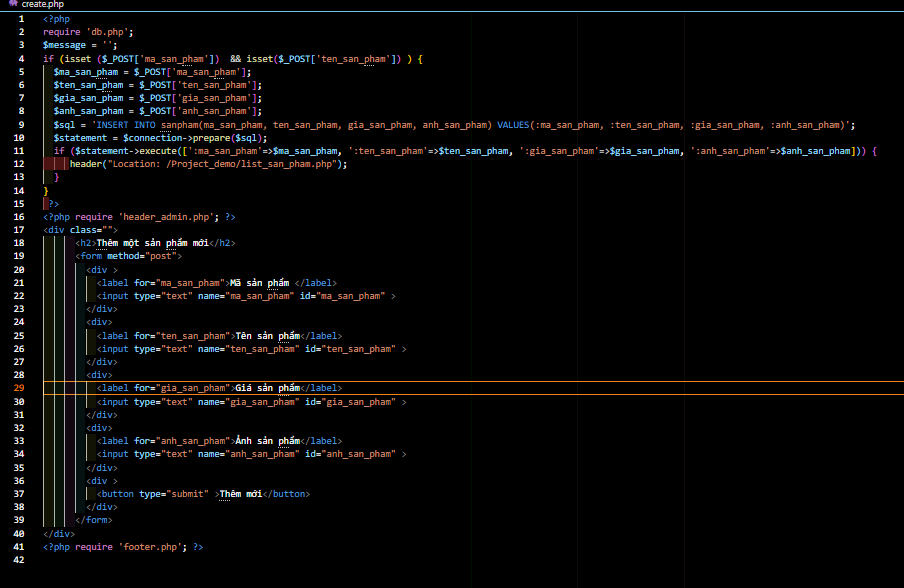
Mã sau đó kiểm tra xem biến $\_POST['id\_san\_pham'] có tồn tại hay không. Biến $\_POST là một mảng chứa các biến được gửi đến tập lệnh hiện tại thông qua phương thức HTTP POST.

Nếu biến $\_POST['id\_san\_pham'] tồn tại, mã lưu giá trị của nó vào biến $id\_san\_pham. Mã cũng lưu giá trị của $\_POST['so\_luong\_san\_pham'.$id\_san\_pham] và $\_POST['thong\_tin\_nguoi\_mua'.$id\_san\_pham] vào các biến $so\_luong\_san\_pham và $thong\_tin\_nguoi\_mua.

Mã sau đó tạo một câu lệnh SQL để thêm một hàng mới vào bảng ‘donhang’ trong cơ sở dữ liệu. Câu lệnh SQL này sẽ thêm giá trị của $id\_san\_pham, $so\_luong\_san\_pham và $thong\_tin\_nguoi\_mua vào các cột ‘ma\_san\_pham’, ‘so\_luong\_san\_pham’ và ‘thong\_tin\_nguoi\_mua’ của bảng ‘donhang’.

Mã sau đó chuẩn bị câu lệnh SQL này để thực thi bằng cách sử dụng phương thức prepare() của đối tượng $connection.

Nếu câu lệnh SQL được thực thi thành công (tức là hàng mới được thêm vào bảng ‘donhang’), mã sẽ chuyển hướng người dùng đến trang ‘list\_san\_pham\_guest.php’ bằng cách sử dụng hàm header().



Đầu tiên, mã yêu cầu tệp ‘db.php’, tệp này có thể chứa thông tin về cơ sở dữ liệu và cách kết nối với nó.

Mã sau đó kiểm tra xem các biến $\_POST['ma\_san\_pham'] và $\_POST['ten\_san\_pham'] có tồn tại hay không. Biến $\_POST là một mảng chứa các biến được gửi đến tập lệnh hiện tại thông qua phương thức HTTP POST.

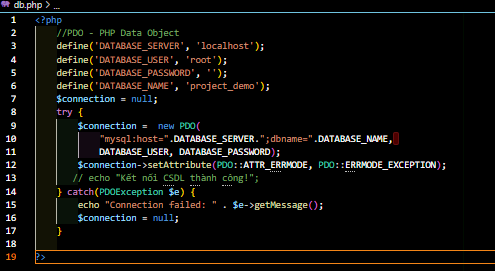
Nếu cả hai biến này đều tồn tại, mã lưu giá trị của chúng cũng như giá trị của $\_POST['gia\_san\_pham'] và $\_POST['anh\_san\_pham'] vào các biến $ma\_san\_pham, $ten\_san\_pham, $gia\_san\_pham và $anh\_san\_pham.

Mã sau đó tạo một câu lệnh SQL để thêm một hàng mới vào bảng ‘sanpham’ trong cơ sở dữ liệu. Câu lệnh SQL này sẽ thêm giá trị của $ma\_san\_pham, $ten\_san\_pham, $gia\_san\_pham và $anh\_san\_pham vào các cột ‘ma\_san\_pham’, ‘ten\_san\_pham’, ‘gia\_san\_pham’ và ‘anh\_san\_pham’ của bảng ‘sanpham’.

Mã sau đó chuẩn bị câu lệnh SQL này để thực thi bằng cách sử dụng phương thức prepare() của đối tượng $connection.

Nếu câu lệnh SQL được thực thi thành công (tức là hàng mới được thêm vào bảng ‘sanpham’), mã sẽ chuyển hướng người dùng đến trang ‘list\_san\_pham.php’ bằng cách sử dụng hàm header().

Cuối cùng, mã hiển thị một biểu mẫu cho phép người dùng nhập thông tin về sản phẩm mới. Khi người dùng nhấn nút ‘Thêm mới’, thông tin họ nhập vào biểu mẫu sẽ được gửi đến tập lệnh PHP hiện tại thông qua phương thức POST.



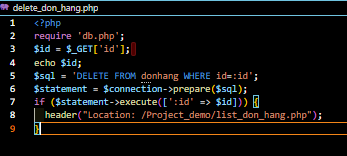
Đầu tiên, mã định nghĩa một số hằng số chứa thông tin về cơ sở dữ liệu, bao gồm tên máy chủ cơ sở dữ liệu (DATABASE\_SERVER), tên người dùng (DATABASE\_USER), mật khẩu (DATABASE\_PASSWORD) và tên cơ sở dữ liệu (DATABASE\_NAME).

Mã sau đó khởi tạo biến $connection là null.

Mã sau đó thử tạo một đối tượng PDO mới, sử dụng thông tin cơ sở dữ liệu đã được định nghĩa trước đó. Nếu kết nối thành công, đối tượng PDO này sẽ được gán cho biến $connection.

Mã sau đó đặt thuộc tính PDO::ATTR\_ERRMODE của đối tượng PDO thành PDO::ERRMODE\_EXCEPTION. Điều này có nghĩa là nếu có bất kỳ lỗi nào xảy ra khi tương tác với cơ sở dữ liệu, một ngoại lệ sẽ được ném ra.

Nếu không thể kết nối với cơ sở dữ liệu (ví dụ, do thông tin cơ sở dữ liệu không chính xác), mã sẽ bắt ngoại lệ PDOException và in ra thông báo lỗi. Biến $connection sau đó sẽ được đặt lại thành null.



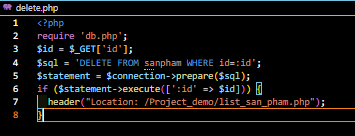
Đầu tiên, mã yêu cầu tệp ‘db.php’, tệp này có thể chứa thông tin về cơ sở dữ liệu và cách kết nối với nó.

Mã sau đó lấy giá trị của biến $\_GET['id'] và lưu nó vào biến $id. Biến $\_GET là một mảng chứa các biến được gửi đến tập lệnh hiện tại thông qua phương thức HTTP GET.

Mã sau đó tạo một câu lệnh SQL để xóa hàng từ bảng ‘donhang’ trong cơ sở dữ liệu có ‘id’ tương ứng với giá trị của $id.

Mã sau đó chuẩn bị câu lệnh SQL này để thực thi bằng cách sử dụng phương thức prepare() của đối tượng $connection.

Nếu câu lệnh SQL được thực thi thành công (tức là hàng với ‘id’ tương ứng đã được xóa khỏi bảng ‘donhang’), mã sẽ chuyển hướng người dùng đến trang ‘list\_don\_hang.php’ bằng cách sử dụng hàm header().



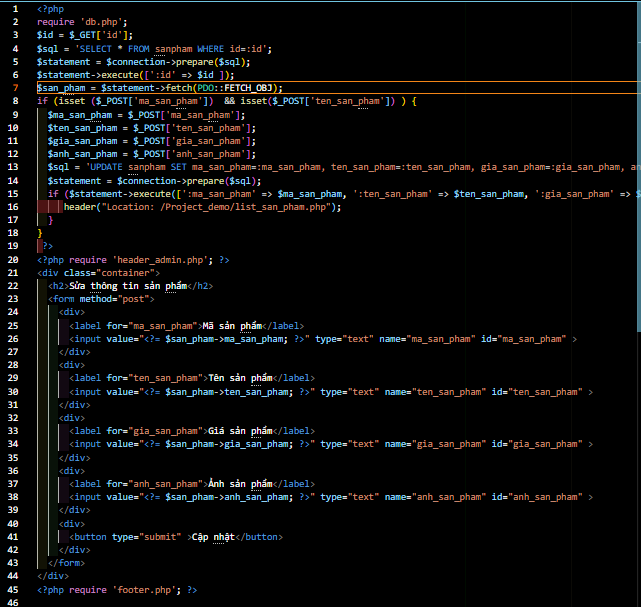
Đầu tiên, mã yêu cầu tệp ‘db.php’, tệp này có thể chứa thông tin về cơ sở dữ liệu và cách kết nối với nó.

Mã sau đó lấy giá trị của biến $\_GET['id'] và lưu nó vào biến $id. Biến $\_GET là một mảng chứa các biến được gửi đến tập lệnh hiện tại thông qua phương thức HTTP GET.

Mã sau đó tạo một câu lệnh SQL để xóa hàng từ bảng ‘sanpham’ trong cơ sở dữ liệu có ‘id’ tương ứng với giá trị của $id.

Mã sau đó chuẩn bị câu lệnh SQL này để thực thi bằng cách sử dụng phương thức prepare() của đối tượng $connection.

Nếu câu lệnh SQL được thực thi thành công (tức là hàng với ‘id’ tương ứng đã được xóa khỏi bảng ‘sanpham’), mã sẽ chuyển hướng người dùng đến trang ‘list\_san\_pham.php’ bằng cách sử dụng hàm header().



Đầu tiên, mã yêu cầu tệp ‘db.php’, tệp này có thể chứa thông tin về cơ sở dữ liệu và cách kết nối với nó.

Mã sau đó lấy giá trị của biến $\_GET['id'] và lưu nó vào biến $id. Biến $\_GET là một mảng chứa các biến được gửi đến tập lệnh hiện tại thông qua phương thức HTTP GET.

Mã sau đó tạo một câu lệnh SQL để lấy tất cả thông tin từ bảng ‘sanpham’ trong cơ sở dữ liệu có ‘id’ tương ứng với giá trị của $id.

Mã sau đó chuẩn bị câu lệnh SQL này để thực thi bằng cách sử dụng phương thức prepare() của đối tượng $connection.

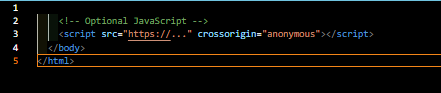
Mã sau đó thực thi câu lệnh SQL và lưu kết quả vào biến $san\_pham.

Nếu biến $\_POST['ma\_san\_pham'] và $\_POST['ten\_san\_pham'] tồn tại, mã sẽ lấy giá trị của chúng cũng như giá trị của $\_POST['gia\_san\_pham'] và $\_POST['anh\_san\_pham'] và lưu vào các biến tương ứng.

Mã sau đó tạo một câu lệnh SQL để cập nhật thông tin sản phẩm trong bảng ‘sanpham’ có ‘id’ tương ứng với giá trị của $id.

Nếu câu lệnh SQL được thực thi thành công (tức là thông tin sản phẩm đã được cập nhật), mã sẽ chuyển hướng người dùng đến trang ‘list\_san\_pham.php’ bằng cách sử dụng hàm header().

Cuối cùng, mã hiển thị một biểu mẫu cho phép người dùng cập nhật thông tin về sản phẩm. Khi người dùng nhấn nút ‘Cập nhật’, thông tin họ nhập vào biểu mẫu sẽ được gửi đến tập lệnh PHP hiện tại thông qua phương thức POST.



Đoạn mã HTML trên là phần cuối của một tệp HTML. Nó đóng thẻ <body> và thẻ <html>, đánh dấu kết thúc của tài liệu HTML.

Ngoài ra, nó còn bao gồm một thẻ <script> để thêm JavaScript vào trang. Thuộc tính src của thẻ <script> được đặt thành URL của tệp JavaScript muốn thêm vào trang. Thuộc tính crossorigin được đặt thành “anonymous”, điều này có nghĩa là trình duyệt sẽ không gửi bất kỳ thông tin xác thực nào (như cookie) cùng với yêu cầu tải tệp JavaScript.



Đầu tiên, mã khai báo loại tài liệu là HTML5 bằng cách sử dụng <!doctype html>.

Mã sau đó mở thẻ <html> với thuộc tính lang được đặt thành “en”, chỉ ra rằng trang web được viết bằng tiếng Anh.

Trong thẻ <head>, mã đặt tiêu đề của trang web là “Hello, world!” bằng cách sử dụng thẻ <title>.

Mã sau đó thêm một số thẻ <meta> để đặt bộ mã ký tự của trang là UTF-8 và để thiết lập viewport cho thiết bị di động.

Mã sau đó thêm một tệp CSS vào trang bằng cách sử dụng thẻ <style> và @import url("CSS/TH06.css");.

Mã sau đó thêm một tệp JavaScript vào trang bằng cách sử dụng thẻ <script> với thuộc tính src được đặt thành URL của tệp JavaScript.

Mã sau đó tạo một thanh điều hướng bằng cách sử dụng thẻ <nav> và <ul>. Thanh điều hướng này chứa ba liên kết: một đến trang ‘list\_san\_pham.php’, một đến trang ‘list\_don\_hang.php’, và một đến trang ‘logout.php’.

Cuối cùng, mã mở thẻ <body> để bắt đầu phần nội dung của trang web.



Đầu tiên, mã khai báo loại tài liệu là HTML5 bằng cách sử dụng <!doctype html>.

Mã sau đó mở thẻ <html> với thuộc tính lang được đặt thành “en”, chỉ ra rằng trang web được viết bằng tiếng Anh.

Trong thẻ <head>, mã đặt tiêu đề của trang web là “Hello, world!” bằng cách sử dụng thẻ <title>.

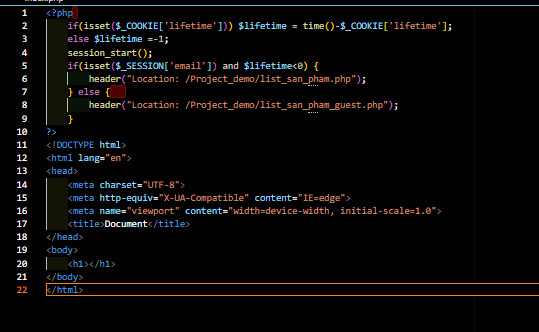
Mã sau đó thêm một số thẻ <meta> để đặt bộ mã ký tự của trang là UTF-8 và để thiết lập viewport cho thiết bị di động.

Mã sau đó thêm một tệp CSS vào trang bằng cách sử dụng thẻ <style> và @import url("CSS/TH06.css");.

Mã sau đó thêm một tệp JavaScript vào trang bằng cách sử dụng thẻ <script> với thuộc tính src được đặt thành URL của tệp JavaScript.

Mã sau đó tạo một thanh điều hướng bằng cách sử dụng thẻ <nav> và <ul>. Thanh điều hướng này chứa một liên kết đến trang ‘login.php’.

Cuối cùng, mã mở thẻ <body> để bắt đầu phần nội dung của trang web.



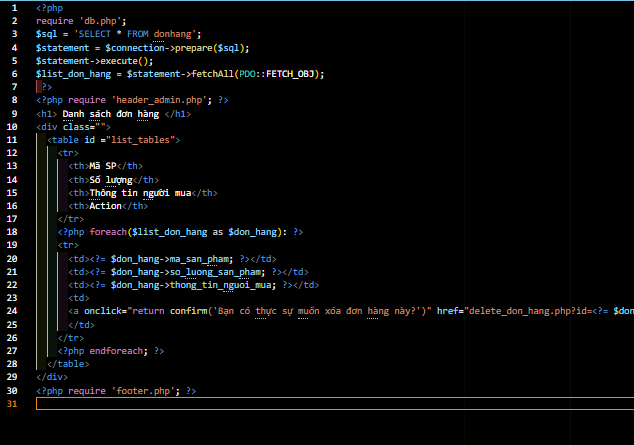
Đoạn mã PHP: Đoạn mã PHP ở đầu tệp sẽ được thực thi ở phía máy chủ trước khi HTML được gửi đến trình duyệt của người dùng.

Đầu tiên, nó kiểm tra xem cookie có tên ‘lifetime’ đã được đặt chưa. Nếu có, nó sẽ tính thời gian tồn tại của cookie là thời gian hiện tại trừ thời gian cookie được đặt. Nếu cookie chưa được đặt, nó sẽ gán -1 cho biến $lifetime.

Sau đó, nó khởi động một phiên bằng session\_start(). Phiên là cách để lưu trữ thông tin (trong các biến) để sử dụng trên nhiều trang.

Tiếp theo, nó kiểm tra xem một biến phiên có tên ‘email’ đã được đặt chưa và thời gian tồn tại của cookie có nhỏ hơn 0 không. Nếu cả hai điều kiện đều đúng, nó sẽ chuyển hướng người dùng đến trang /Project\_demo/list\_san\_pham.php. Ngược lại, nó sẽ chuyển hướng người dùng đến trang /Project\_demo/list\_san\_pham\_guest.php.

Tài liệu HTML: Phần còn lại của tệp là một tài liệu HTML đơn giản với một tiêu đề trống. Nó không liên quan đến đoạn mã PHP ở trên và không hiển thị bất kỳ nội dung nào cho người dùng.



Đoạn mã PHP: Đoạn mã PHP ở đầu tệp sẽ được thực thi ở phía máy chủ trước khi HTML được gửi đến trình duyệt của người dùng.

Đầu tiên, nó yêu cầu tệp ‘db.php’ để kết nối với cơ sở dữ liệu.

Sau đó, nó tạo một câu lệnh SQL để lấy tất cả các bản ghi từ bảng ‘donhang’.

Câu lệnh SQL sau đó được chuẩn bị và thực thi, và kết quả được lấy ra dưới dạng một mảng các đối tượng, mỗi đối tượng tương ứng với một bản ghi trong bảng ‘donhang’.

Tài liệu HTML: Phần còn lại của tệp là một tài liệu HTML hiển thị danh sách các đơn hàng.

Nó bao gồm một bảng với các tiêu đề cột là ‘Mã SP’, ‘Số lượng’, ‘Thông tin người mua’, và ‘Action’.

Mỗi hàng trong bảng tương ứng với một đơn hàng, với các thông tin về mã sản phẩm, số lượng sản phẩm, thông tin người mua, và một liên kết để xóa đơn hàng.

Liên kết xóa sẽ chuyển hướng người dùng đến trang ‘delete\_don\_hang.php’ với id của đơn hàng được chuyển qua URL. Trước khi chuyển hướng, nó sẽ hiển thị một hộp thoại xác nhận yêu cầu người dùng xác nhận họ có thực sự muốn xóa đơn hàng hay không.

3của30



Đoạn mã PHP: Đoạn mã PHP ở đầu tệp sẽ được thực thi ở phía máy chủ trước khi HTML được gửi đến trình duyệt của người dùng.

Đầu tiên, nó yêu cầu tệp ‘db.php’ để kết nối với cơ sở dữ liệu.

Sau đó, nó tạo một câu lệnh SQL để lấy tất cả các bản ghi từ bảng ‘sanpham’.

Câu lệnh SQL sau đó được chuẩn bị và thực thi, và kết quả được lấy ra dưới dạng một mảng các đối tượng, mỗi đối tượng tương ứng với một bản ghi trong bảng ‘sanpham’.

Tài liệu HTML: Phần còn lại của tệp là một tài liệu HTML hiển thị danh sách các sản phẩm.

Nó bao gồm một danh sách các sản phẩm, mỗi sản phẩm được hiển thị trong một div với thông tin về tên sản phẩm, hình ảnh sản phẩm, giá sản phẩm, và một nút để mua hàng.

Khi người dùng nhấp vào nút mua hàng, họ sẽ được chuyển hướng đến trang ‘create\_don\_hang.php’ với thông tin về sản phẩm được chuyển qua biểu mẫu.

Trang này cũng bao gồm các chức năng JavaScript để thay đổi số lượng sản phẩm được mua và tính toán tổng tiền cho sản phẩm đó.



Đoạn mã PHP: Đoạn mã PHP ở đầu tệp sẽ được thực thi ở phía máy chủ trước khi HTML được gửi đến trình duyệt của người dùng.

Đầu tiên, nó yêu cầu tệp ‘db.php’ để kết nối với cơ sở dữ liệu.

Sau đó, nó tạo một câu lệnh SQL để lấy tất cả các bản ghi từ bảng ‘sanpham’.

Câu lệnh SQL sau đó được chuẩn bị và thực thi, và kết quả được lấy ra dưới dạng một mảng các đối tượng, mỗi đối tượng tương ứng với một bản ghi trong bảng ‘sanpham’.

Tài liệu HTML: Phần còn lại của tệp là một tài liệu HTML hiển thị danh sách các sản phẩm.

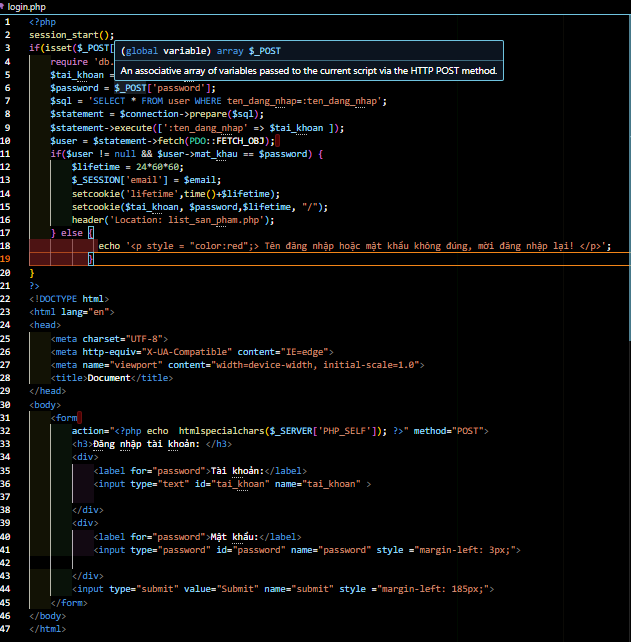
Nó bao gồm một bảng với các tiêu đề cột là ‘Mã SP’, ‘Tên sản phẩm’, ‘Giá sản phẩm’, ‘Ảnh’, và ‘Action’.

Mỗi hàng trong bảng tương ứng với một sản phẩm, với các thông tin về mã sản phẩm, tên sản phẩm, giá sản phẩm, hình ảnh sản phẩm, và hai liên kết để chỉnh sửa và xóa sản phẩm.

Liên kết chỉnh sửa sẽ chuyển hướng người dùng đến trang ‘edit.php’ với id của sản phẩm được chuyển qua URL.

Liên kết xóa sẽ chuyển hướng người dùng đến trang ‘delete.php’ với id của sản phẩm được chuyển qua URL. Trước khi chuyển hướng, nó sẽ hiển thị một hộp thoại xác nhận yêu cầu người dùng xác nhận họ có thực sự muốn xóa sản phẩm hay không.

Cuối cùng, có một liên kết để thêm một sản phẩm mới, chuyển hướng người dùng đến trang ‘create.php’.



Bắt đầu bằng hàm **session\_start()**.

Nếu người dùng nhấn nút "Submit" trên form đăng nhập, mã sẽ được thực thi.

Tiến hành kết nối đến cơ sở dữ liệu MySQL thông qua hàm **require 'db.php';**.

Lấy dữ liệu người dùng nhập từ các input tên đăng nhập (**tai\_khoan**) và mật khẩu (**password**).

Chuẩn bị và thực thi câu truy vấn SQL để lấy thông tin người dùng từ cơ sở dữ liệu.

Sử dụng hàm **fetch()** để lấy kết quả trả về từ câu truy vấn (kết quả là một đối tượng).

Kiểm tra xem người dùng có tồn tại và mật khẩu có khớp không. Nếu có, tiếp tục xử lý.

Thực hiện các bước sau đăng nhập thành công, bao gồm:

Thiết lập biến session **$\_SESSION['email']** để lưu trữ thông tin người dùng.

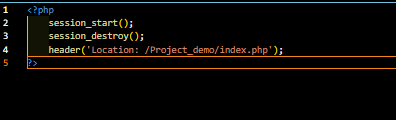
Thiết lập cookie **lifetime** để xác định thời gian tồn tại của cookie.

Thiết lập cookie **tai\_khoan** và **password** để lưu trữ thông tin đăng nhập của người dùng.

Chuyển hướng người dùng đến trang **list\_san\_pham.php**.

Nếu thông tin đăng nhập không hợp lệ, hiển thị thông báo lỗi.

Sau đó, mã tiếp tục với một đoạn mã HTML để hiển thị giao diện đăng nhập đơn giản. Form này sẽ gửi dữ liệu người dùng nhập đến cùng một trang (**$\_SERVER['PHP\_SELF']**) để xử lý.

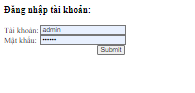


Bắt đầu phiên làm việc (session\_start())

Huỷ phiên làm việc hiện tại (session\_destroy())

Chuyển hướng trình duyệt đến trang /Project\_demo/index.php (header('Location: /Project\_demo/index.php'))

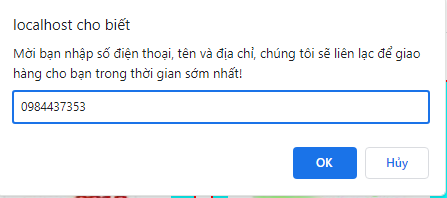
**+ Đăng nhập, đăng xuất**

****

+ Quản lý sản phẩm (sách sản phẩm, thêm, sửa, xóa sản phẩm)

****

+ Đặt hàng (cho khách hàng)





+ Xử lý đơn hàng

