

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG ĐIỆN – ĐIỆN TỬ



BÁO CÁO ĐỒ ÁN 1
Thiết kế website bán hàng máy tính

GVHD:

TS. Trần Thị Ngọc Lan

Sinh viên thực hiện:

Nguyễn Xuân Hoàng

MSSV:

20203433

Mục lục

Chương 1.	Tổng quan	2
1.1.	Giới thiệu về đề tài.....	2
1.2.	Lý do chọn đề tài	2
1.3.	Mục đích phát triển đề tài	2
1.4.	Ý nghĩa thực tiễn	3
Chương 2.	Tìm hiểu một số ngôn ngữ xây dựng website.....	4
2.1.	Ngôn ngữ HTML.....	4
2.1.1.	<i>Giới thiệu về HTML</i>	4
2.1.2.	<i>Bố cục HTML của một trang web</i>	4
2.1.3.	<i>Một số thẻ định dạng được dùng trong thẻ <body></i>	5
2.2.	Ngôn ngữ PHP.....	8
2.2.1.	<i>Giới thiệu về PHP</i>	8
2.2.2.	<i>Tại sao nên dùng PHP</i>	9
2.2.3.	<i>Tìm hiểu khái quát về PHP</i>	9
2.3.	Cơ sở dữ liệu từ MySQL.....	18
2.3.1.	<i>Giới thiệu về cơ sở dữ liệu</i>	18
2.3.2.	<i>Mục đích sử dụng cơ sở dữ liệu</i>	18
2.3.3.	<i>MySQL và PHP</i>	19
Chương 3.	Kết quả thu được	21
Kết luận		

Chương 1. Tổng quan

1.1. Giới thiệu về đề tài

Trong bối cảnh kinh tế thị trường hiện nay, việc mỗi công ty, cửa hàng hay cá nhân có một hay nhiều website là một việc hết sức bình thường, xong việc khai thác dữ liệu từ website biến những tác động lên website trở thành những dữ liệu có ích lại là một vấn đề khá còn mới mẻ. Vấn đề còn bỏ ngỏ, nhận thấy cơ hội, khả năng phát triển, em nhận đề tài “Thiết kế website bán hàng” nhằm mục đích đưa các giải pháp phát triển một website dễ dàng tiếp cận với người dùng.

1.2. Lý do chọn đề tài

Khi internet ra đời, không ai nghĩ rằng nó sẽ phát triển như ngày hôm nay, nó len lỏi đến mọi ngõ ngách trong cuộc sống và trở thành một phần không thể thiếu trong cuộc sống của chúng ta.

Chỉ cần ngồi ở nhà và cả thế giới đều ở trong tay bạn. Cũng như vậy, khi internet trở nên phổ biến việc mua bán hàng qua mạng internet phát triển cũng phát triển theo đó.

Xong bên cạnh đó, việc phát triển website đi kèm những tính năng nâng cao giúp tối ưu hóa việc trải nghiệm website của người dùng cũng là cách giúp website phát triển một cách bền vững và có hiệu quả.

1.3. Mục đích phát triển đề tài

Đề tài được phát triển nhằm mục đích thương mại điện tử, quản lý việc mua bán thông qua mạng internet, nâng cao trải nghiệm sử dụng website của người dùng qua đó đưa thương hiệu đến rộng rãi với người dùng hơn.

1.4. Ý nghĩa thực tiễn

Phát triển các sản phẩm công nghệ phù hợp với nhu cầu của xã hội, qua đó rèn luyện, trau dồi, phát triển kiến thức của bản thân nhằm đáp ứng nhu cầu của các nhà tuyển dụng sau khi ra trường, tạo ra các sản phẩm thực tế có giá trị sử dụng trong cuộc sống.

Chương 2. Tìm hiểu một số ngôn ngữ xây dựng website

2.1. Ngôn ngữ HTML

2.1.1. Giới thiệu về HTML

HTML là từ viết tắt của **HyperText Markup Language**, có nghĩa là **ngôn ngữ đánh dấu siêu văn bản**. Nó cùng với CSS được thiết kế ra để tạo nên các website tĩnh.

HTML dùng để định dạng bố cục, các thuộc tính liên quan đến cách hiển thị của một đoạn text và được hiển thị trên một chương trình đặc biệt gọi là trình duyệt (browser). Hiện nay có khá nhiều trình duyệt như Firefox, Chrome, Opera, Safari,... Tất cả trình duyệt đều có điểm chung là giúp người dùng thao tác với website đó và nó đều có khả năng biên dịch những đoạn mã HTML, CSS và JavaScript.

Chúng ta có thể sử dụng rất nhiều phần mềm soạn thảo để viết HTML, ví dụ như công cụ Notepad được tích hợp sẵn trong hệ điều hành Windows.

2.1.2. Bố cục HTML của một trang web

HTML là một định dạng đặc biệt của XML nên nó sẽ có thẻ mở và thẻ đóng với cú pháp như sau:

```
<tagname> </tagname>
```

Ngoài ra, mỗi thẻ HTML còn có một số thuộc tính riêng và danh sách các thuộc tính sẽ nằm bên trong thẻ mở như sau:

```
<tagname prop1="value1" prop2="value2"> </tagname>
```

Thông thường, bố cục HTML của một website sẽ có dạng như sau:

```
<html>
  <head>
    <title>Tiêu đề trang</title>
```

```
</head>
<body>
    Nội dung của trang
</body>
</html>
```

Trong đó:

`<html>` và `</html>` là cặp thẻ nằm ngoài cùng và nó có nhiệm vụ bao hết nội dung của website lại. Thẻ này là bắt buộc.

`<head>` và `</head>` là phần khai báo thông tin của website.

`<title>` và `</title>` nằm bên trong thẻ `<head>` và đây chính là khai báo tiêu đề cho website.

`<body>` và `</body>` là thành phần quan trọng nhất, nó chứa những đoạn mã HTML dùng để hiển thị trên website.

Các thẻ còn lại trong thẻ `<body>` chính là các thẻ định dạng dữ liệu.

Như vậy, trong một website chúng ta chia làm hai phần chính:

Phần 1: Là những khai báo thông tin cho website được đặt trong thẻ `<head>`

Phần 2: Là phần hiển thị định dạng nội dung của website được đặt trong thẻ `<body>`

2.1.3. Một số thẻ định dạng được dùng trong thẻ `<body>`

a. Thẻ xuống dòng

Khi bạn đánh máy một văn bản trong một chương trình soạn thảo văn bản nào đó, ta có thể chỉ cần bấm phím **Enter** để kết thúc một dòng hay một đoạn văn, nhưng ta sẽ cần phải làm nhiều thao tác hơn nếu muốn phân định riêng các đoạn văn khác nhau trong một trang Web. Trình duyệt tự động bỏ qua các mã xuống dòng bình thường. Vì vậy ta phải viết thêm thẻ `<p>` để phân đoạn hay viết thẻ `
` tại vị trí cụ thể cần xuống dòng.

- Thẻ `<p>`: dùng để xuống dòng và định nghĩa một đoạn văn (paragraph). Trong thẻ này có thuộc tính **align** để xác định kiểu căn lề của đoạn văn, thuộc tính này có 3 giá trị đó là *left*, *right*, *center*.

- Thẻ `
`: dùng để xuống dòng trong một đoạn văn bản.

b. Đường kẻ ngang

Đường kẻ ngang chủ yếu được dùng để phân định một website thành những phần có tính logic và dễ nhìn. Có không nhiều công cụ đồ họa trong ngôn ngữ HTML, do đó đường kẻ ngang là một công cụ rất hữu ích. Để tạo đường kẻ ngang ta dùng thẻ `<hr>`. Thẻ `<hr>` gồm các thuộc tính sau:

- *size*: độ dày của đường kẻ, đơn vị tính thường là px
- *width*: độ dài của đường kẻ, đơn vị tính thường là px hoặc %
- *color*: màu của đường kẻ, giá trị có thể là mã màu hoặc tên màu
- *align*: vị trí của đường kẻ, giá trị có thể sử dụng đó là *left*, *right*, *center*

c. Danh sách

- Danh sách không có trật tự: không có trật tự đánh dấu (xuất hiện các nút tròn màu đen) cho từng yếu tố của danh sách. Để tạo danh sách không có trật tự, ta dùng đồng thời các cặp thẻ `` và ``.

```
<ul>
    <li> text1 </li>
    <li> text2 </li>
    <li> text3 </li>
</ul>
```

- Danh sách có trật tự: tự động đánh số từng thành tố của danh sách. Để tạo danh sách có trật tự, ta dùng đồng thời các cặp thẻ `` và ``.

```
<ol>
    <li> text1 </li>
    <li> text2 </li>
```

 text3

d. Ký tự định dạng, kiểu chữ

- Kiểu ký tự định dạng trước cho phép bảo toàn các dấu cách và dấu xuống dòng trong trang văn bản HTML y như khi gõ vào trong quá trình soạn thảo. Do đó, không nên sử dụng các thẻ <p> cũng như
 trong đoạn ký tự định dạng trước. Để sử dụng kiểu ký tự định dạng trước, ta dùng thẻ <pre> </pre>.

- Các kiểu chữ trong văn bản HTML: các thẻ kiểu chữ trong HTML cho phép điều khiển được vẻ bề ngoài của bản thân các ký tự, ví dụ ta có thể định dạng một từ là **đậm** hoặc *ngiên*. Các định dạng này có thể áp dụng từ phạm vi một đoạn hay toàn bộ văn bản cho tới từng ký tự đơn lẻ.

- + Thẻ : định dạng kiểu chữ đậm (**text**)
- + Thẻ <i> </i>: định dạng kiểu chữ nghiêng (*text*)
- + Thẻ <u> </u>: định dạng kiểu chữ gạch chân (text)
- + Thẻ <s> </s>: định dạng kiểu chữ gạch ngang qua ký tự (~~text~~)
- + Thẻ : đưa ký tự thụt xuống dưới 1/2 của dòng (H₂O)
- + Thẻ : đưa ký tự trôi lên 1/2 của dòng (x²)

e. Hình ảnh

Để giúp trang văn bản HTML sinh động và trực quan hơn, ta nên chèn thêm vào trang những hình ảnh minh họa (ví dụ hình ảnh về sản phẩm ...). Ngôn ngữ HTML cung cấp thẻ để giúp ta chèn được hình ảnh vào website. Thẻ này có 2 thuộc tính sau:

- *src*: xác định đường dẫn tham chiếu của hình ảnh
- *alt*: xác định văn bản thay thế cho hình ảnh trong trường hợp đường dẫn hình ảnh bị sai.

f. Địa chỉ liên kết

Để chèn địa chỉ liên kết trong một trang văn bản HTML, ta dùng thẻ `<a>`. Các thuộc tính trong thẻ `<a>` gồm:

- *href* : đường dẫn đến địa chỉ đích.
- *title* : tiêu đề cho đường dẫn liên kết.
- *target* : gồm các tùy chọn: *_blank* (chuyển link trên tab mới), *_self* (chuyển link trên tab hiện tại), *_parent* (chuyển link tới tab cha của tab hiện tại).

g. Bảng

Bảng gồm nhiều hàng và nhiều cột, giao của hàng và cột là một ô. Trong ngôn ngữ HTML, để tạo bảng ta dùng thẻ `<table>`.

- Để tạo một dòng mới trong bảng, ta dùng thẻ `<tr>` `</tr>`. Thẻ này được đặt trong thẻ `<table>`.

- Trong một dòng, để tạo một ô ta dùng thẻ `<td>` `</td>`. Thẻ này được đặt trong thẻ `<tr>`.

2.2. Ngôn ngữ PHP

2.2.1. Giới thiệu về PHP

PHP là viết tắt của “Personal Home Page” do Rasmus Lerdorf tạo ra năm 1994. Vì tính hữu dụng của nó và khả năng phát triển, PHP bắt đầu được sử dụng trong môi trường chuyên nghiệp và nó trở thành “PHP: Hypertext Preprocessor”.

Thực chất PHP là ngôn ngữ kịch bản nhúng trong HTML, nói một cách đơn giản đó là một trang HTML có nhúng mã PHP, PHP có thể được đặt rải rác trong HTML.

PHP là một ngôn ngữ lập trình được kết nối chặt chẽ với máy chủ, là một công nghệ phía máy chủ và không phụ thuộc vào môi trường. Đây là hai yếu tố rất quan trọng, thứ nhất khi nói công nghệ phía máy chủ tức là nói đến mọi thứ trong PHP đều xảy ra trên máy chủ, thứ hai, chính vì tính chất không phụ thuộc môi trường cho

phép PHP chạy trên hầu hết các hệ điều hành phổ biến hiện nay. Đặc biệt các mã PHP viết trên máy chủ này sẽ làm việc bình thường trên máy chủ khác mà không cần phải chỉnh sửa hoặc chỉnh sửa rất ít.

Khi một website muốn được dùng ngôn ngữ PHP thì phải đáp ứng được tất cả các quá trình xử lý thông tin trong website đó, sau đó đưa ra kết quả ngôn ngữ HTML.

Khác với ngôn ngữ lập trình, PHP được thiết kế để chỉ thực hiện điều gì đó sau khi một sự kiện xảy ra (ví dụ, khi người dùng gửi một biểu mẫu hoặc chuyển tới một URL).

2.2.2. Tại sao nên dùng PHP

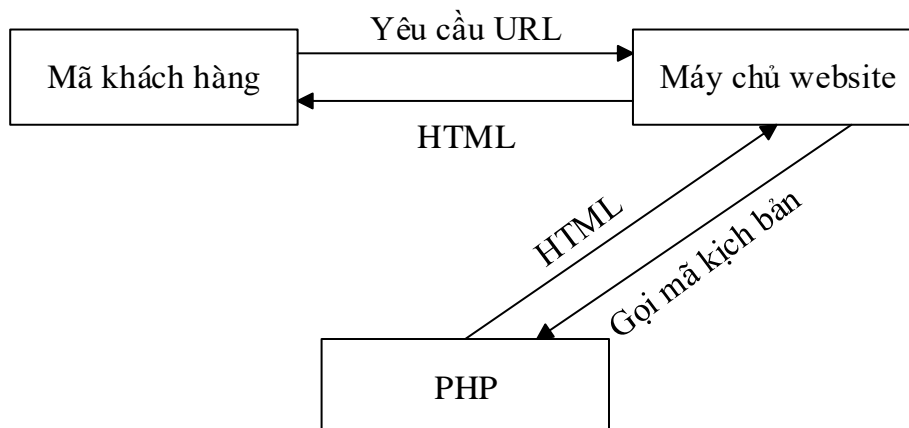
Để thiết kế website động có rất nhiều ngôn ngữ lập trình khác nhau như ASP, PHP, Java, Perl,... Tuy nhiên, hiện nay PHP là ngôn ngữ đang được sử dụng khá rộng rãi vì những lí do sau:

- PHP được sử dụng làm website động vì nó nhanh, dễ dàng, tốt hơn so với các giải pháp khác.
- PHP có khả năng thực hiện và tích hợp chặt chẽ với hầu hết các cơ sở dữ liệu có sẵn, tính linh động, bền vững và khả năng phát triển không giới hạn.
- PHP là mã nguồn mở, do đó tất cả các đặc tính đều được miễn phí, và được cộng đồng các nhà phát triển web luôn có ý thức cải tiến, nâng cao để khắc phục các lỗi trong các chương trình.
- PHP vừa dễ với người mới sử dụng vừa có thể đáp ứng được mọi yêu cầu của các lập trình viên chuyên nghiệp.

2.2.3. Tìm hiểu khái quát về PHP

Vì PHP là ngôn ngữ của máy chủ nên mã lệnh của PHP sẽ tập trung trên máy chủ để phục vụ các website theo yêu cầu của người dùng thông qua trình duyệt.

Sơ đồ hoạt động:



Khi người dùng truy cập website viết bằng PHP, máy chủ đọc mã lệnh PHP và xử lý chúng theo các hướng dẫn được mã hóa. Mã lệnh PHP yêu cầu máy chủ gửi một dữ liệu thích hợp đến trình duyệt web. PHP cũng là một trang HTML nhưng có nhúng mã PHP và có phần mở rộng là HTML. Do đó, trình duyệt xem nó như là một trang HTML tiêu chuẩn. Khi trình duyệt truy cập vào một trang PHP, máy chủ sẽ đọc nội dung file PHP và lọc ra các đoạn mã PHP và thực thi các đoạn mã đó, lấy kết quả nhận được của đoạn mã PHP thay thế vào chỗ ban đầu của chúng trong file PHP, cuối cùng máy chủ trả về kết quả cuối cùng là một trang nội dung HTML về cho trình duyệt.

a. Các loại thẻ PHP

Có 3 loại thẻ khác nhau có thể sử dụng khi thiết kế trang PHP:

- Kiểu Short: thẻ mặc định mà các nhà lập trình PHP thường sử dụng

```
<?php
    echo "Hello World";
?>
```

- Kiểu Script: trong trường hợp sử dụng PHP như một script thì khai báo tương tự khai báo Javascript:

```
<script language= "php">  
    Echo "PHP Script";  
</script>
```

- Kiểu ASP: trong trường hợp khai báo thẻ PHP như một phần trong trang ASP

```
<% echo "PHP ASP"; %>
```

PHP và HTML là các ngôn ngữ không “nhạy cảm” với khoảng trắng, do đó ta có thể đặt các khoảng trắng xung quanh để các mã lệnh rõ ràng hơn. Chỉ có khoảng trắng đơn mới có ảnh hưởng đến sự thể hiện của website, nhiều khoảng trắng liên tiếp sẽ chỉ thể hiện dưới dạng một khoảng trắng đơn.

b. Các kiểu dữ liệu

- Số nguyên: được khai báo và sử dụng giá trị giống với C.

```
$a = 12345;  
$b = -456;
```

- Số thực:

```
$c = 2.141;  
$d = 2.4e4;
```

- Xâu ký tự: có hai cách để xác định một xâu: đặt giữa hai dấu nháy kép (" ") hoặc giữa hai dấu nháy đơn (' ').

```
$s1 = "Bach khoa";  
$s2 = 'HUST in my heart';
```

- Mảng: mảng thực chất gồm hai bảng: bảng chỉ số và bảng liên kết.

```
$arr1 = array(1, 2, 3);  
$arr2 = array('John' => 10, 'Sam' => 9, 'Peter' => 11);
```

c. Biên – giá trị

PHP quy định một biến được bắt đầu bằng ký tự \$, sau đó là một chữ cái hoặc dấu gạch dưới.

Một số biến đã được khai báo sẵn:

+ *HTTP_GET_VARS*: Mảng các giá trị nguyên truyền tới Script thông qua phương thức HTTP GET.

+ *HTTP_POST_VARS*: Mảng các giá trị nguyên truyền tới Script thông qua phương thức HTTP POST.

+ *HTTP_COOKIE_VARS*: Mảng các giá trị được truyền tới Script hiện tại bằng HTTP Cookie.

Phạm vi giá trị:

PHP coi một biến có một giới hạn. Để xác định một biến toàn cục có tác dụng trong một hàm ta cần khai báo lại bằng từ khóa *global*. Nếu không thì giá trị của biến sẽ được coi như là cục bộ trong hàm.

Tên biến: mỗi biến có thể gắn với một cái tên

Ví dụ: \$a = "chao";

\$chao = "cacban";

echo \$\$a;

Kết quả sẽ là: "chaocacban"

d. Các giá trị bên ngoài phạm vi PHP

HTML Form: Khi một giá trị gắn với một file PHP thông qua phương thức POST

```
<form action = "process.php" method = "post">
```

```
    Name: <input type = "text" name = "name"><br>
```

```
    <input type = "submit">
```

```
</form>
```

PHP sẽ tạo ra một biến \$name bao gồm mọi giá trị trong trường name của form và PHP có thể hiểu được một mảng một chiều gồm các giá trị trong một form.

Vì vậy, ta có thể nhóm những giá trị liên quan lại với nhau hoặc sử dụng đặc tính này để nhận các giá trị từ một đầu vào tùy chọn.

Các giá trị được Submit sẽ lấy ra qua phương thức GET và POST hoặc có thể lấy ra từ hai mảng toàn cục `$HTTP_GET_VARS` và `$HTTP_POST_VARS`.

e. Hằng

PHP định nghĩa sẵn các hằng số:

`_FILE_`: tên của script file đang thực hiện

`_LINE_`: số dòng của mã script đang được thực hiện trong script file hiện tại

`_PHP_VERSION_`: phiên bản hiện tại của PHP đang chạy

`TRUE`

`FALSE`

`E_ERROR`: báo hiệu có lỗi

`E_PARSE`: báo lỗi sai khi biên dịch

`E_NOTICE`: một vài sự kiện có thể là lỗi hoặc không

Ta cũng có thể định nghĩa một hằng số bằng hàm *define()*.

f. Biểu thức

Biểu thức là một thành phần quan trọng trong PHP. Một dạng cơ bản nhất của biểu thức bao gồm các biến và hằng số. PHP hỗ trợ ba kiểu giá trị cơ bản nhất: số nguyên, số thực và xâu ký tự. Ngoài ra còn có mảng và đối tượng. Mỗi kiểu giá trị này có thể gán cho các biến hay làm giá trị của các hàm.

g. Các cấu trúc lệnh

Các lệnh điều kiện và toán tử

Mỗi câu lệnh điều kiện bao gồm một mệnh đề if:

```
if(<condition>){
```

```
    // Thực hiện một việc gì đó
```

```
}
```

Điều kiện này cũng có thể được mở rộng thành:

```
if(<condition>){  
    // Thực hiện một việc gì đó  
} else {  
    // Thực hiện một việc khác  
}
```

Hay:

```
if(<condition1>){  
    // Thực hiện một việc gì đó  
} elseif(<condition2>) {  
    // Thực hiện một việc khác  
} else{  
    // Thực hiện một điều khác nữa  
}
```

Các toán tử thường được dùng với câu lệnh điều kiện

Ký hiệu	Ý nghĩa	Dạng	Ví dụ
=	Được gán giá trị của	Gán	\$a = 2
==	Bằng	So sánh	\$a == \$b
!=	Khác	So sánh	\$a != \$b
<	Nhỏ hơn	So sánh	\$a < \$b
>	Lớn hơn	So sánh	\$a > \$b
<=	Nhỏ hơn hoặc bằng	So sánh	\$a <= \$b
>=	Lớn hơn hoặc bằng	So sánh	\$a >= \$b
!	Phủ định	Logic	!\$a

&&	AND	Logic	\$a>0 && \$b<2
	OR	Logic	\$a<2 \$b>1

PHP còn có một dạng câu lệnh điều kiện nữa là switch. Câu lệnh này rất phù hợp cho việc thay thế nhiều câu lệnh if-elseif-else quá dài. Cú pháp của câu lệnh switch như sau:

```
switch($var){
    case 'value1':
        // Thực hiện lệnh
        break;
    case 'value2':
        // Thực hiện lệnh
        break;
    default:
        // Thực hiện lệnh
}
```

Vòng lặp for

```
for(<expression1>;<condition>;<expression2>){
    // Thực hiện lệnh
}
```

Trong đó:

condition: điều kiện giới hạn của vòng lặp for

expression1: giá trị khởi đầu của vòng lặp for

expression2: giá trị lặp của vòng lặp for

Vòng lặp while

```
while(<condition>){
```



```
// Thực hiện lệnh  
}
```

Khối lệnh trong vòng lặp while được thực thi chừng nào điều kiện trong while còn đúng.

Vòng lặp do...while

```
do{  
    // Thực hiện lệnh  
}  
while(<condition>);
```

Sau mỗi lần thực hiện khối lệnh trong vòng lặp, điều kiện trong while được kiểm tra, nếu điều kiện đúng thì khối lệnh tiếp tục được thực hiện, nếu điều kiện sai thì kết thúc vòng lặp.

h. Hàm

Hàm trong PHP được dùng giống với trong C++. Ngoại trừ việc không cần phải khai báo kiểu cho tham số của hàm:

- Tham trị

```
function print($str){  
    echo $str;  
}
```

- Tham biến

```
function increasing(&$a){  
    $a++;  
}
```

- Tham số có giá trị mặc định

```
function makecoffee($type = "cappucino"){
```

```

        return "Making a cup of $type.";
    }

```

Chú ý: khi sử dụng hàm có đối số có giá trị mặc định, các biến này sẽ phải nằm về phía phải ngoài cùng trong danh sách đối số.

- Giá trị trả lại của hàm: có thể là bất kỳ giá trị nào. Tuy vậy, không thể trả lại nhiều giá trị riêng lẻ nhưng có thể trả lại một mảng các giá trị.

- Hàm biến: PHP cho phép sử dụng hàm giá trị nghĩa là khi một biến được gọi có kèm theo dấu ngoặc đơn, PHP sẽ tìm hàm có cùng tên với giá trị biến đó thực hiện.

i. Các toán tử

Các phép số học: +, -, *, /, %

Các toán tử logic: &&, ||, !

Toán tử thao tác với bit: &, |, ^, ~, <<, >>

Toán tử so sánh: ==, !=, <, >, <=, >=, ===

Toán tử điều khiển lỗi: @ - khi đứng trước một biểu thức thì các lỗi của biểu thức sẽ bị bỏ qua và lưu trong \$PHP_errormsg

j. Lớp và đối tượng

Class: là tập hợp các biến và hàm làm việc với các biến này. Một lớp có định dạng như sau:

```

<?php
class Cart{
    var $item;
    function add_item($a){
        echo $a;
    }
}

```

}

?>

Lớp Cart ở đây là một kiểu dữ liệu.

2.3. Cơ sở dữ liệu từ MySQL

2.3.1. Giới thiệu về cơ sở dữ liệu

MySQL là ứng dụng cơ sở dữ liệu mã nguồn mở phổ biến nhất hiện nay và được sử dụng phối hợp với PHP. Trước khi làm việc với MySQL cần xác định các nhu cầu cho ứng dụng.

MySQL là cơ sở dữ liệu có trình giao diện, cho phép người dùng có thể thao tác các hành động liên quan đến nó. Việc tìm hiểu từng công nghệ trước khi bắt tay vào việc viết mã kịch bản PHP và tích hợp hai công nghệ PHP và MySQL là một công việc cần thiết và rất quan trọng.

2.3.2. Mục đích sử dụng cơ sở dữ liệu

Mục đích sử dụng cơ sở dữ liệu bao gồm các chức năng như: lưu trữ, truy cập, tổ chức và xử lý.

- Lưu trữ: lưu trữ ở trên đĩa và có thể chuyển đổi dữ liệu từ cơ sở dữ liệu này sang cơ sở dữ liệu khác. Nếu sử dụng cho quy mô nhỏ có thể chọn các cơ sở dữ liệu nhỏ như Excel, Access, MySQL, Visual FoxPro,... Nếu ứng dụng có quy mô lớn có thể chọn cơ sở dữ liệu có quy mô lớn hơn như Oracle, SQL Server,...

- Truy cập: Truy cập dữ liệu phụ thuộc vào mục đích và yêu cầu của người sử dụng, ở mức độ mang tính cục bộ, truy cập cơ sở dữ liệu ngay trong cơ sở dữ liệu với nhau, nhằm trao đổi hay xử lý dữ liệu ngay bên trong chính nó, nhưng do mục đích và yêu cầu người dùng vượt ra ngoài cơ sở dữ liệu nên ta cần có các phương thức truy cập dữ liệu giữa các cơ sở dữ liệu với nhau như Access với SQL Server, hay SQL Server với Oracle,...

- Tổ chức: Tổ chức cơ sở dữ liệu phụ thuộc vào mô hình cơ sở dữ liệu, phân tích và thiết kế cơ sở dữ liệu tức là tổ chức cơ sở dữ liệu phụ thuộc vào đặc điểm

riêng của từng ứng dụng. Tuy nhiên khi tổ chức cơ sở dữ liệu cần phải tuân theo một số tiêu chuẩn của hệ thống cơ sở dữ liệu nhằm tăng tính tối ưu khi truy cập và xử lý.

- Xử lý: Tùy vào nhu cầu tính toán và truy vấn cơ sở dữ liệu với các mục đích khác nhau, cần phải sử dụng các phát biểu truy vấn cùng các phép toán, phát biểu của cơ sở dữ liệu để xuất ra kết quả như yêu cầu. Để thao tác hay xử lý dữ liệu bên trong chính cơ sở dữ liệu ta sử dụng các ngôn ngữ lập trình như PHP, C++, Java,...

2.3.3. **MySQL và PHP**

Để kết nối tới một cơ sở dữ liệu trên MySQL rất đơn giản. Chỉ cần sử dụng hàm `mysql_connect(host, username, password)` để mở kết nối tới MySQL. Sau đó, dùng hàm `mysql_select_db(dbname, $link)` để kết nối tới cơ sở dữ liệu có tên `dbname`.

Để thực hiện một câu lệnh truy vấn tới cơ sở dữ liệu, ta dùng hàm `mysql_query(query, link)`. Giá trị trả lại của hàm là kết quả của câu truy vấn. Nếu bỏ qua link thì kết nối cuối cùng tới MySQL sẽ được thực hiện.

Để lấy kết quả thực hiện, ta có thể dùng hàm `mysql_result()`, `mysql_fetch_row()`, `mysql_fetch_array()`, `mysql_fetch_object()`.

- `mysql_fetch_row()`: trả về một mảng các dữ liệu lấy từ một dòng trong cơ sở dữ liệu. Nếu đã ở cuối cơ sở dữ liệu thì giá trị trả về là false. Dữ liệu được lấy ra thông qua chỉ số của các trường trong cơ sở dữ liệu.

- `mysql_fetch_array(int res[], int res_type)`: trả về một mảng chứa dữ liệu lấy từ một dòng trong cơ sở dữ liệu. Tương tự như `mysql_fetch_row()` nhưng ta có thể lấy thêm trường để lấy dữ liệu.

- `mysql_fetch_object()`: chuyển kết quả thành một đối tượng với các trường là các trường trong cơ sở dữ liệu.

Một số câu lệnh truy vấn MySQL thường dùng:

"SELECT * FROM Table_name": câu lệnh này dùng để lấy ra các dòng trong bảng cơ sở dữ liệu.

"INSERT INTO Table_name (Prop1, Prop2,...) VALUES (Value1, Value2,...)": câu lệnh này dùng để chèn một dòng dữ liệu mới vào bảng trong cơ sở dữ liệu.

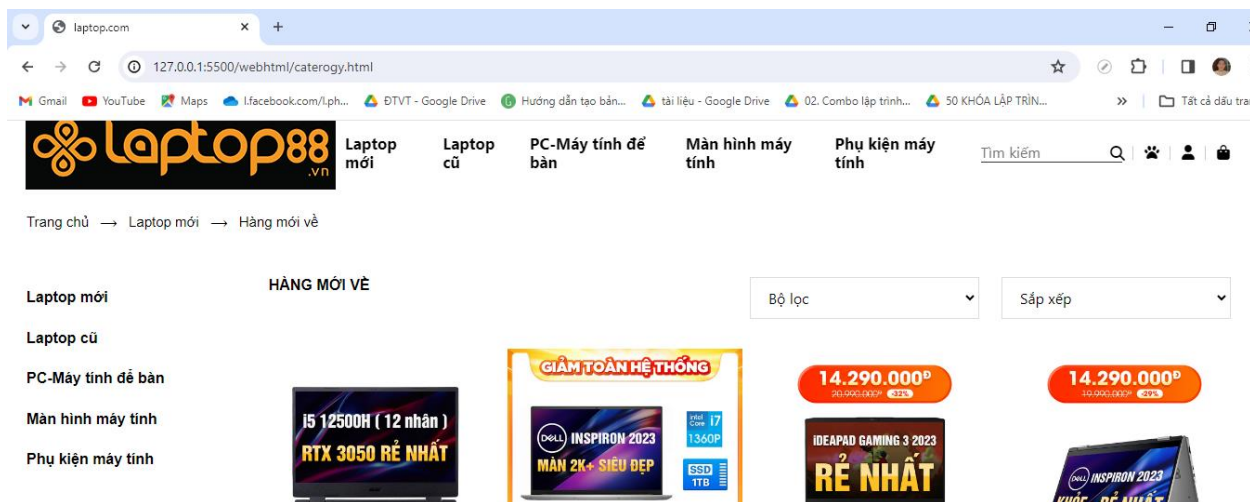
"UPDATE Table_name SET Prop1=Value1, Prop2=Value2,... WHERE ... ": câu lệnh này dùng để cập nhật, thay đổi một dòng dữ liệu đã có trong bảng cơ sở dữ liệu.

"DELETE FROM Table_name WHERE ...": câu lệnh này dùng để xóa một dòng dữ liệu trong bảng cơ sở dữ liệu.

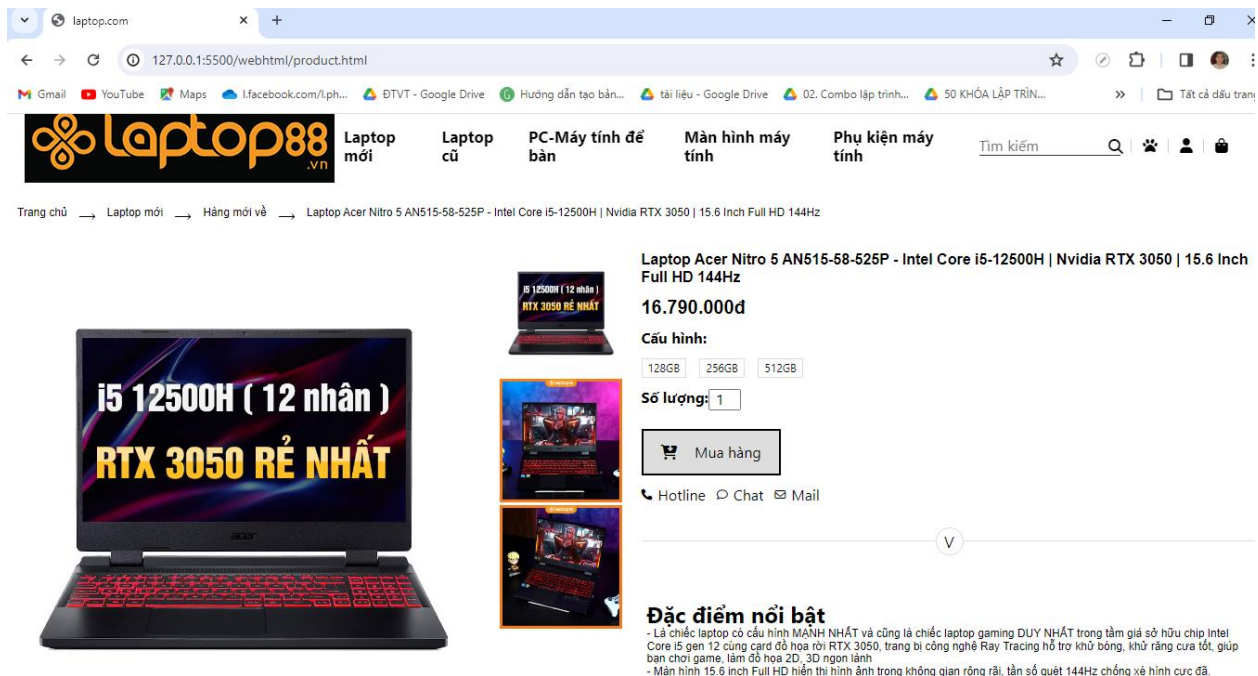
Chương 3. Kết quả thu được



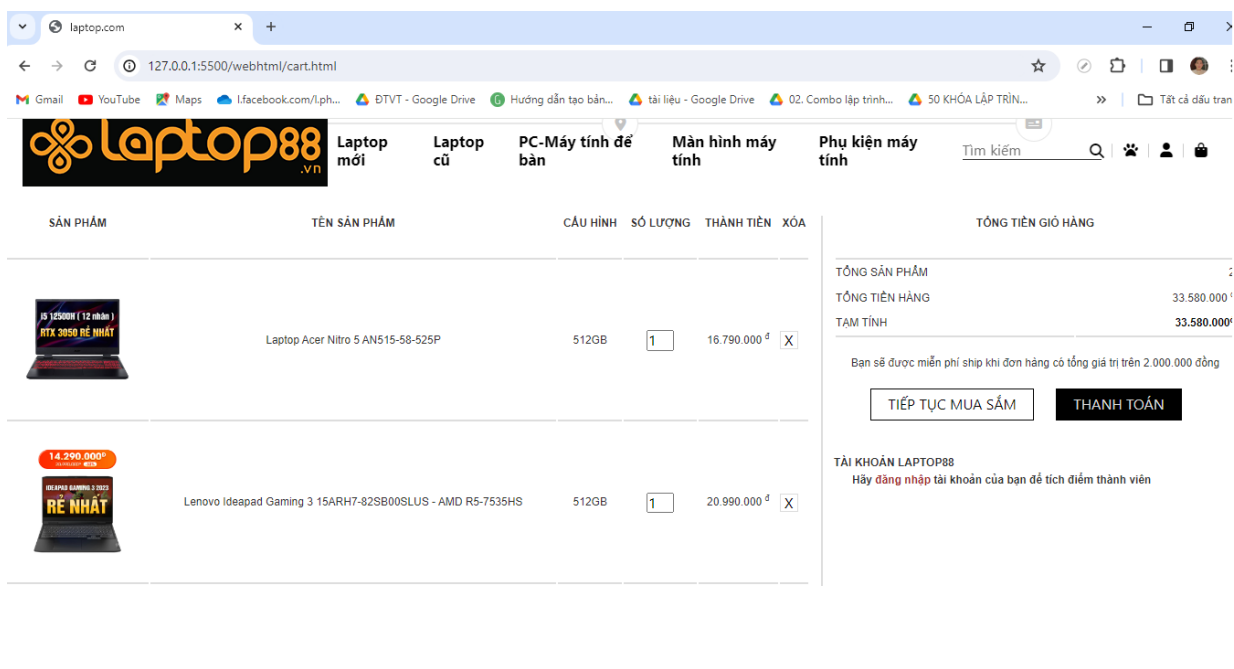
Hình 1: Giao diện trang website chính



Hình 2: Giao diện phần trưng bày sản phẩm



Hình 3: Giao diện chi tiết một sản phẩm



Hình 4: Giao diện giỏ hàng

Vui lòng chọn địa chỉ giao hàng

☒ Đăng nhập (Nếu bạn đã có tài khoản của IVY)
☒ Khách lẻ (Nếu bạn không muốn lưu lại thông tin)
☐ Đăng ký (Tạo tài khoản với thông tin bên dưới)

Họ tên *

Điện thoại *

Tỉnh/TP *

Quận/Huyện *

Địa chỉ *

« Quay lại giỏ hàng

THANH TOÁN VÀ GIAO HÀNG

Tên sản phẩm	Giảm giá	Số lượng	Thành tiền
Laptop Acer Nitro 5 AN515-58-525P	15%	1	15.000.000 đ
Tổng tiền hàng			15.000.000 đ

Hình 5: Giao diện điền thông tin địa chỉ giao hàng

Phương thức giao hàng

☐ Giao hàng chuyển phát nhanh
☒ Phương thức thanh toán

Mọi giao dịch đều được bảo mật và mã hóa. Thông tin thẻ tín dụng sẽ không bao giờ được lưu lại

☐ Thanh toán bằng thẻ tín dụng(OnePay)
☐ Thanh toán bằng thẻ ATM(OnePay)
☐ Thanh toán MOMO
☐ Thanh toán khi nhận hàng

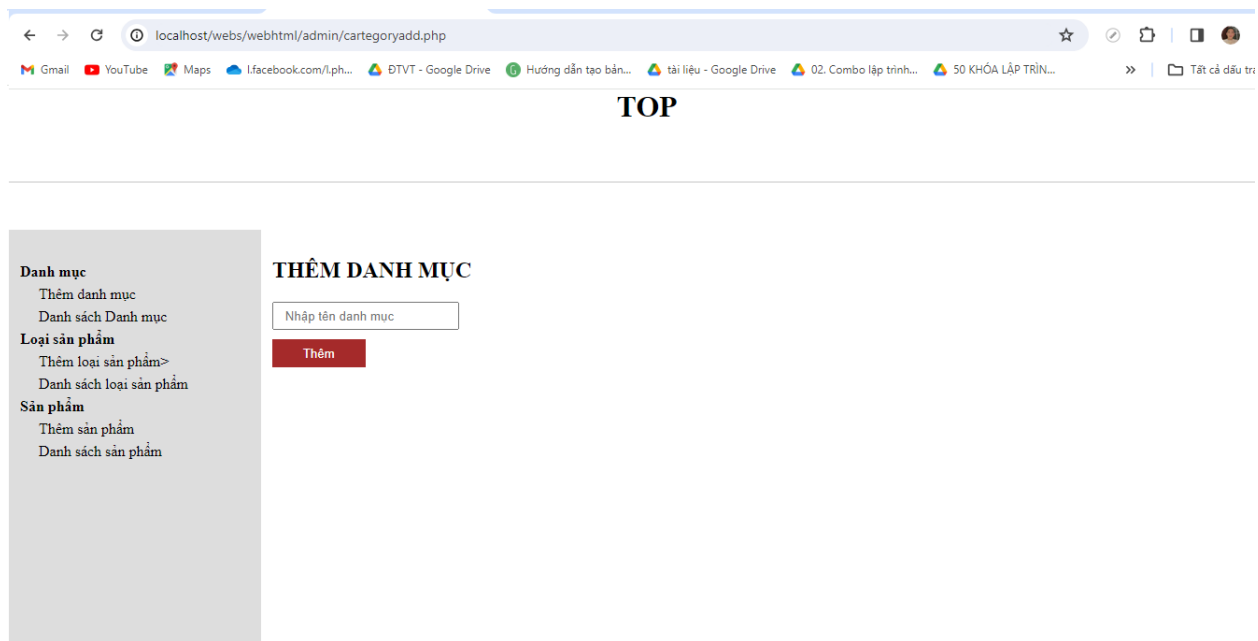
Mã giảm giá/Quà tặng

Mã cộng tác viên

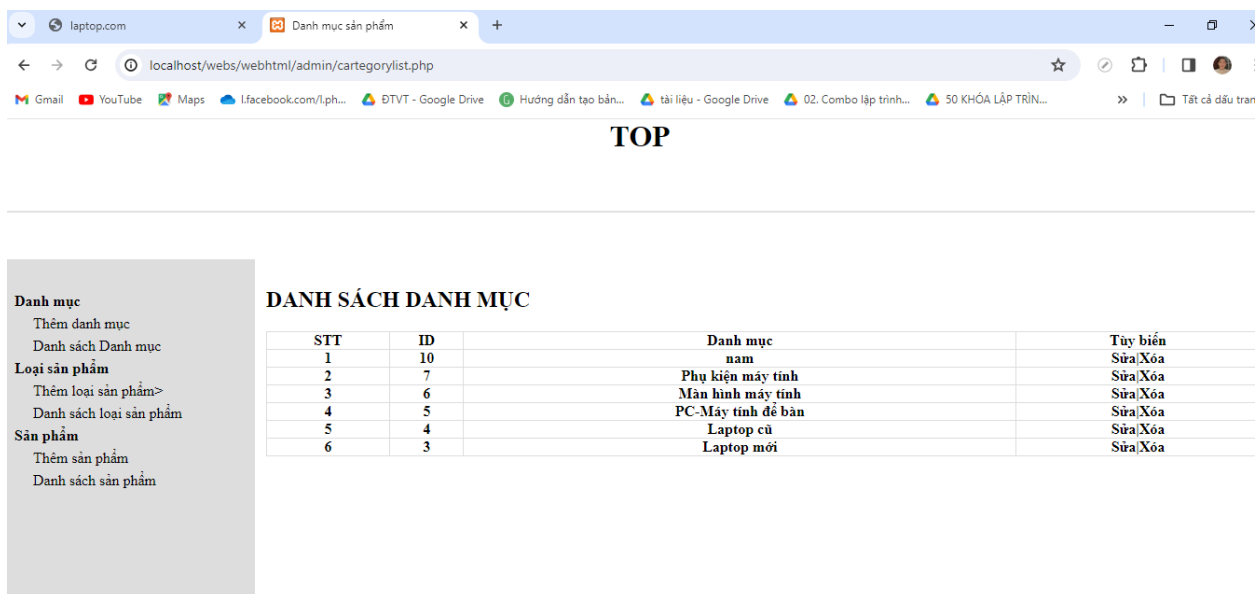
« Quay lại

TIẾP TỤC THANH TOÁN

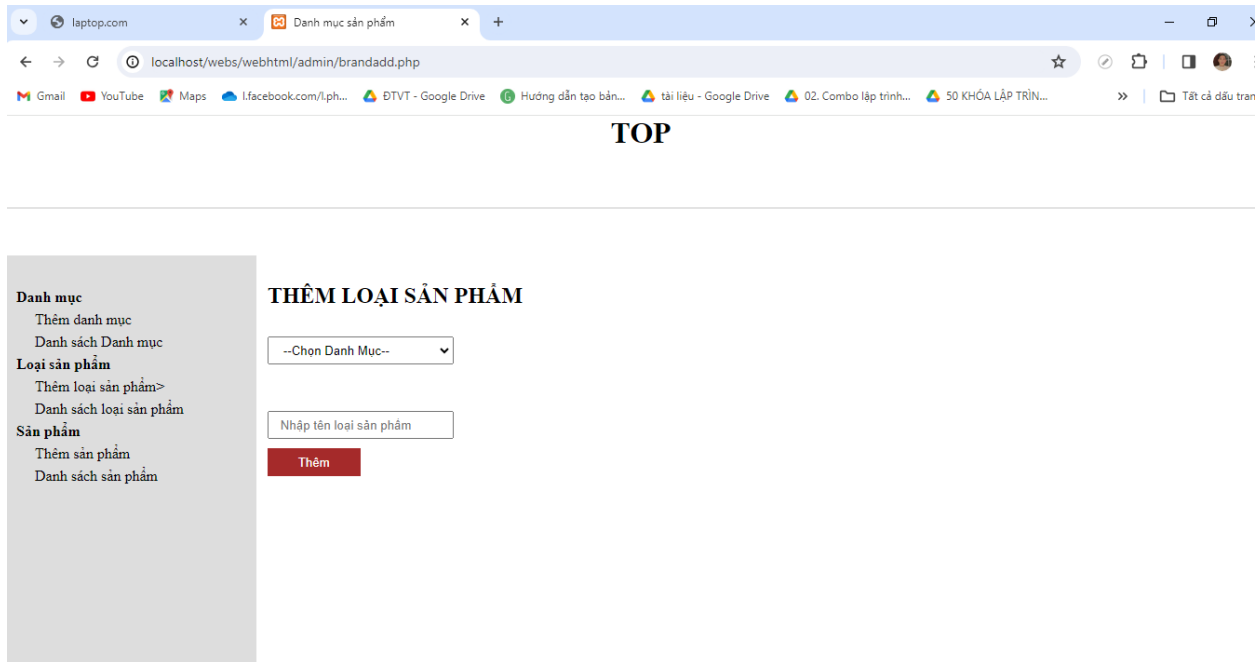
Hình 6: Giao diện trang thanh toán



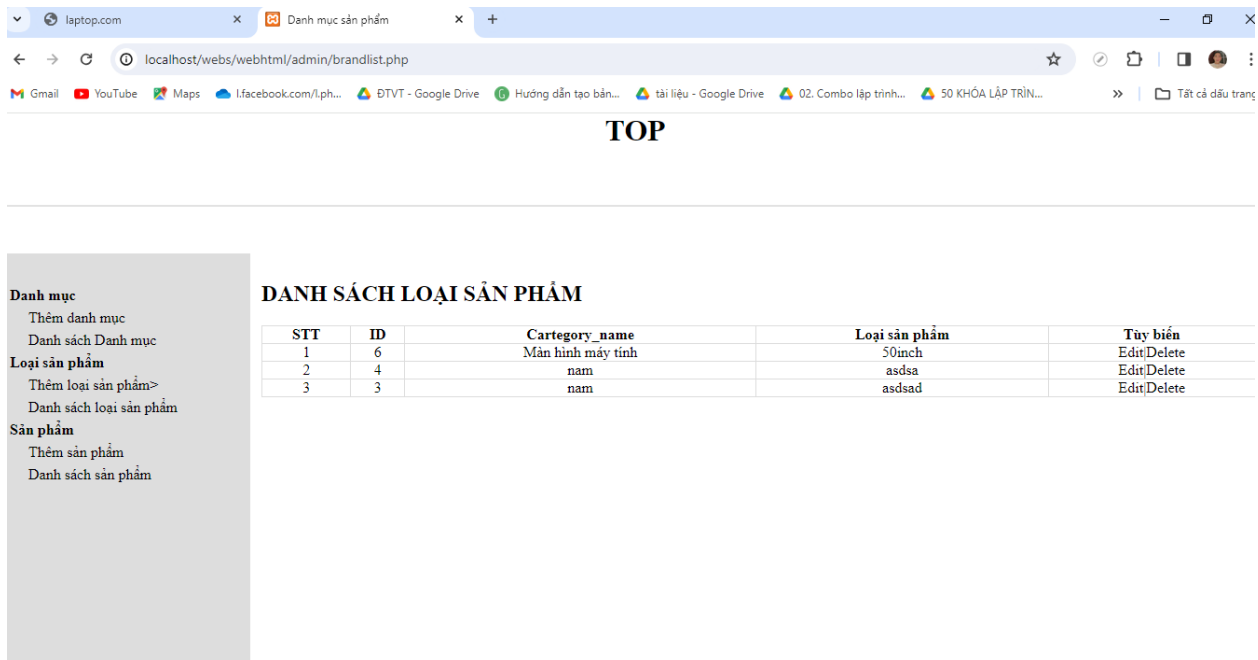
Hình 7: Giao diện thêm danh mục



Hình 8: Hiện thị danh sách danh mục được thêm



Hình 9: Giao diện thêm loại sản phẩm



Hình 10: Hiện thị danh sách loại sản phẩm được thêm

The screenshot displays a web browser window with two tabs: 'laptop.com' and 'Danh mục sản phẩm'. The address bar shows the URL 'localhost/webs/webhtml/admin/productadd.php'. The browser's toolbar includes icons for back, forward, refresh, and search, along with a list of bookmarks such as 'Gmail', 'YouTube', 'Maps', and 'ĐTVT - Google Drive'. The main content area is titled 'THÊM SẢN PHẨM' and contains the following elements:

- Sidebar (Left):**
 - Danh mục**
 - Thêm danh mục
 - Danh sách Danh mục
 - Loại sản phẩm**
 - Thêm loại sản phẩm>
 - Danh sách loại sản phẩm
 - Sản phẩm**
 - Thêm sản phẩm
 - Danh sách sản phẩm
- Main Content Area:**
 - THÊM SẢN PHẨM**
 - Input field: Nhập tên sản phẩm
 - Dropdown menu: --Chọn--
 - Dropdown menu: --Chọn loại sản phẩm--
 - Input field: Giá sản phẩm
 - Input field: Giá khuyến mãi
 - Rich text editor toolbar: Includes icons for image, link, paragraph, bold, italic, and undo/redo.
 - Text area: Mô tả sản phẩm
 - File upload buttons: Two buttons labeled 'Chọn tệp' with the text 'Không có tệp nào được chọn'.
 - Submit button: A red button labeled 'Thêm'.

Hình 11: Giao diện thêm sản phẩm

Kết luận

Sau khi làm xong đồ án 1. Em xin cảm ơn cô Trần Thị Ngọc Lan đã hướng dẫn em trong quá trình làm đồ án. Mặc dù đồ án lần này chưa được hoàn thiện lắm và không tránh khỏi những sai sót, nhưng sau khi hoàn thiện đồ án 1 đã giúp em tiếp thu được rất nhiều về cách để xây dựng một website. Bên cạnh đó cũng giúp em hiểu hơn về ngôn ngữ HTML, PHP, MySQL, tiếp cho em thêm những động lực nỗ lực để hoàn thành tốt chương trình học tập của mình.