

BEGINNING PHP5.APACHE.MYSQL WEB DEVELOPMENT

Programmer to Programmer™



Beginning

PHP5, Apache, MySQL® Web Development

Elizabeth Naramore, Jason Gerner, Yann Le Scouarnec, Jeremy Stolz, Michael K. Glass



Updates, source code, and Wrox technical support at www.wrox.com

Chương 1: SƠ LƯỢC VỀ APACHE, PHP, MYSQL VÀ CÁCH CÀI ĐẶT

1.1. Tóm tắt sơ lược về Apache, PHP, MySQL:

Apache, PHP, MySQL là các phần của chương trình phần mềm mã nguồn mở. Phần mềm này là sự hợp tác của những lập trình viên có đầu óc. Bằng việc thay đổi trên những mã nguồn có sẵn, các lập trình viên trên thế giới tiếp tục hoàn thiện và phát triển nó ngày càng trở nên mạnh hơn và hiệu quả hơn. Sự đóng góp của những người này được đưa ra công khai cho người khác sử dụng mã nguồn, tạo ra các phần mềm nổi tiếng trên thế giới.

Tại sao mã nguồn mở gây chấn động?

-*Nó miễn phí*: vì là mã nguồn mở nên các lập trình viên có thể dựa trên mã nguồn sẵn có để phát triển thêm, phần mềm này không có bản quyền nên mọi người có thể tự do sử dụng .

-*Nó dựa trên nền chữ thập (cross_platform) và kỹ thuật trung lập(technology-neutral)*. Nó không có một nền tảng cụ thể. Tập đoàn mã nguồn mở bảo đảm rằng nó không được phát triển bởi một cá nhân nào. Do đó định nghĩa mã nguồn mở được cung cấp bởi OSI (open source initiative), mã nguồn mở không phụ thuộc vào bất cứ một kỹ thuật cá nhân hay một loại giao diện, nó phải là một kỹ thuật trung lập.

-*Nó không bị giới hạn bởi những phần mềm khác*, điều này có nghĩa là nếu một chương trình mã nguồn mở phân phối cho nhiều chương trình thì những chương trình khác có thể cũng là mã nguồn mở và giao dịch trong tự nhiên, điều này làm cho phần mềm phát triển tối đa và linh hoạt.

-*Nó có tính đa dạng*: được phát triển bởi nhiều người thuộc nhiều nền văn hoá, nhiều lĩnh vực phong phú.

Những phần AMP (Apache, MySQL, PHP) làm việc với nhau như thế nào?

Tương tượng trang web động như một nhà hàng, khách đến ăn không cần quan tâm đến nó làm ra như thế nào coi như nó có sẵn và người bồi bàn giúp bạn chọn món ăn.

Tôi có thể mô tả 3 thành phần của AMP như sau:

- **Apache:** Cứ tưởng tượng như một đầu bếp . Khi khách hàng yêu cầu món ăn . Cô ta nhanh nhẹn linh hoạt và có thể chuẩn bị vô số những món ăn khác nhau. Hoạt động của Apache trong nhiều phương thức giống nhau là phân tích những file riêng lẻ và đưa ra kết quả.
- **PHP:** giống như người bồi bàn trong khách sạn . Anh ta nhận lời đề nghị của khách hàng và mang đến món ăn từ trong nhà bếp với những lời hướng dẫn cụ thể.
- **MySQL:** là những kho lưu trữ, những thành phần thông tin.

Khi khách đến ăn (người thăm trang web) đến nhà hàng Anh ta sẽ chú ý đến món ăn với nhu cầu của mình. Người bồi bàn (PHP) lấy thức ăn theo yêu cầu từ trong nhà bếp được làm bởi đầu bếp(Apache). Người đầu bếp này sẽ đến kho để lấy những nguyên liệu(dữ liệu) để chuẩn bị cho món ăn.

1.1.1. Apache :

Hoạt động của apache như là web server của bạn : công việc chính của nó là phân tích bất kỳ những file yêu cầu của trình duyệt web nào và đưa ra những kết quả chính xác.Apache tuyệt vời có thể hoàn thành hầu hết những yêu cầu của bạn.

1.1.2. PHP:

PHP là một ngôn ngữ mở rộng chính mà cho phép trang web của bạn hoạt động. Nó linh hoạt và tương đối nhỏ. Nó đã được áp dụng và phát triển trong kinh doanh. Nó cũng giống như ngôn ngữ Asp của Microsoft, Nhưng PHP thân thiện hơn,dễ sử dụng hơn.

1.1.3. MySQL :

MySQL là một cấu trúc dữ liệu có thể kết hợp với PHP, Apache để làm việc với nhau. Nó là ngôn ngữ truy vấn có cấu trúc. Nó đưa ra hiệu ứng và tốc độ cao.

Nó cập nhật dữ liệu, xóa dữ liệu, thêm dữ liệu mà không ảnh hưởng đến cấu trúc hệ thống.

1.2. Cài đặt:

Các bước cài đặt bộ ba Apache,PHP,MySQL

Bước 1: tải bộ ba Apache, PHP, MySQL tại địa chỉ :

<http://www.apachefriends.org/download.php?xampp-win32-1.4.16-installer.exe>

Bước 2 : Cài đặt bình thường như những ứng dụng khác

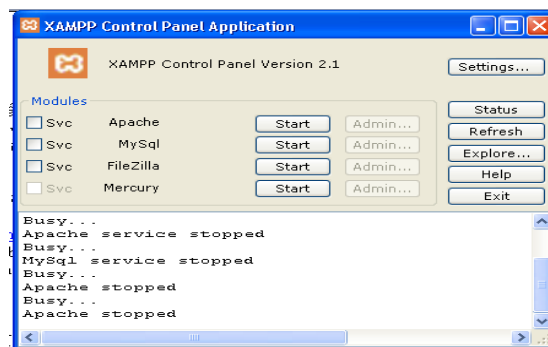
Bước 3 : Kiểm tra cài đặt thành công chưa:

- Sau khi cài đặt trên màn hình có biểu tượng như **Hình 1.2-1**



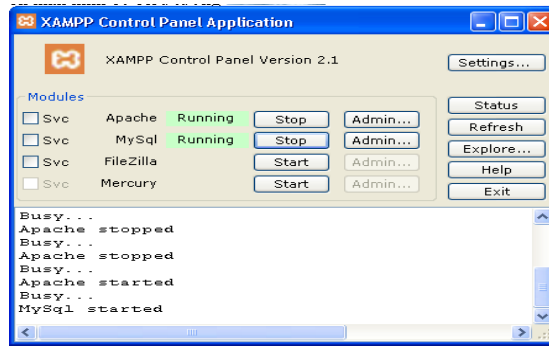
Hình 1.2.1

- Nhấp đúp vào biểu tượng sẽ xuất hiện như **Hình 1.2-2**



Hình 1.2.2

- Nhấp vào nút start của Apache và MySQL cho chương trình chạy **Hình 1.2.3**



Hình 1.2.3

Sau đó thực hiện các thao tác để viết lệnh trên PHP hoặc MySQL

Lưu ý: khi thực hiện lệnh trên PHP hoặc MySQL thì vẫn để chương trình XAMPP chạy song song với chương trình đang thực hiện.

Để kiểm tra việc cài đặt có thành công chưa ta thực hiện như sau.

Vào một trình soạn thảo chẳng hạn như Notepad gõ dòng lệnh:

```
<?
```

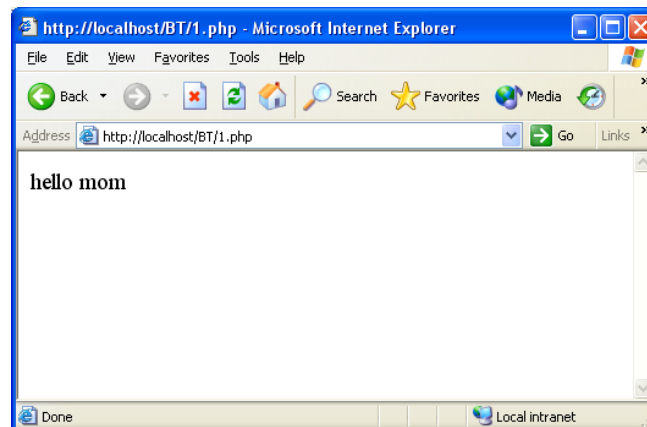
```
    echo "hello mom ";
```

```
?>
```

Lưu vào thư mục đã cài đặt XAMP: xamp\htdocs. Với đuôi *.php

Sau đó mở trình duyệt web gõ http://localhost/*.php

Nếu thành công trình duyệt sẽ xuất hiện như **Hình 1.2.4:**



Hình 1.2.4

Như vậy là việc cài đặt đã thành công.

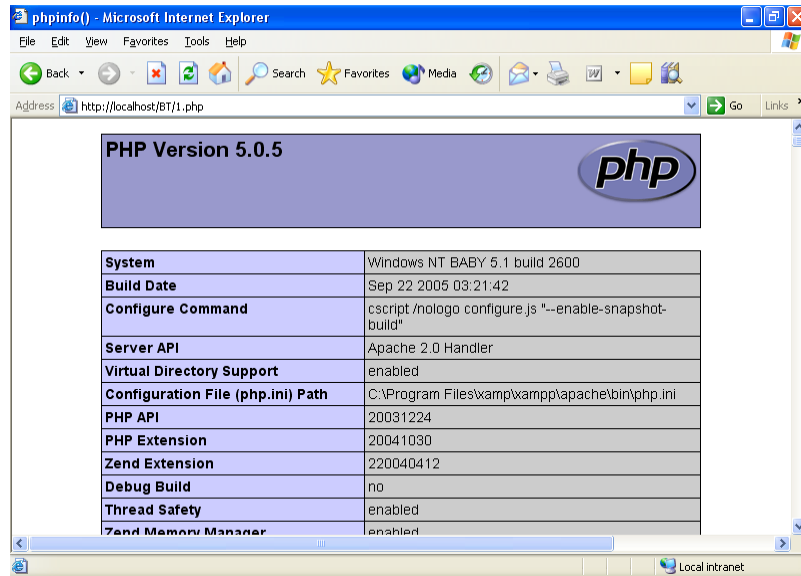
Hoặc gõ như sau:

<?

 Phpinfo();

?>

Bạn sẽ thấy kết quả như **Hình 1.2.5**:



Hình 1.2.5

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

Hai nguyên tắc cơ bản trong PHP

- PHP hiển thị trong trang Web với thẻ mở và thẻ đóng như sau:

```
<?php
    // First line of code goes here;
    // Second line of code goes here;
    // Third line of code goes here;
?>
```

-Những dòng PHP kết thúc bằng dấu “;”.Lời chú thích được viết sau “//” (trên mỗi dòng) hoặc “/*” ở đầu đoạn chú thích và “*/” ở cuối đoạn chú thích

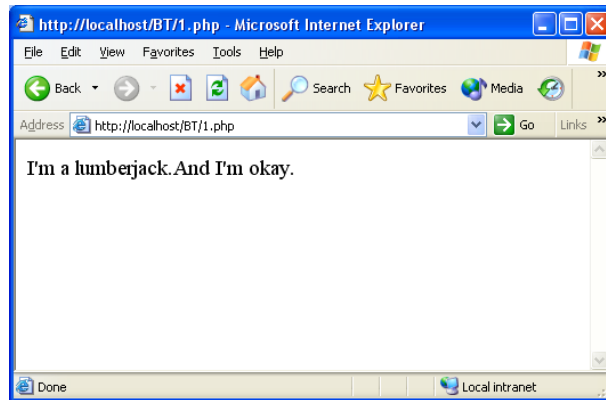
Sử dụng “echo” để trình bày text

Hàm “echo” là hàm xuất cơ bản trên trình duyệt Web.

Vd:

```
<?php
    echo "I'm a lumberjack.";
    echo "And I'm okay.";
?>
```

Kết quả Hình 2.1.1:



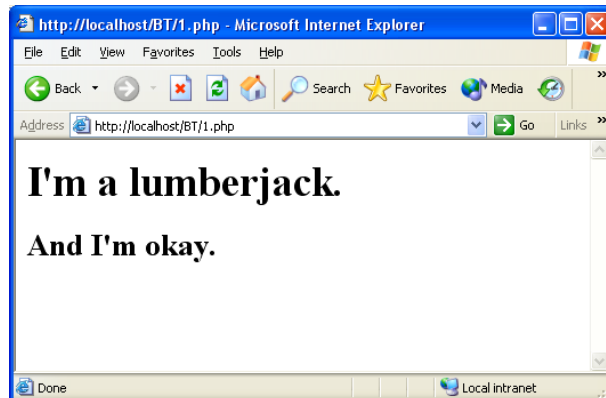
Hình 2.1.1

Chúng ta có thể lồng HTML với PHP.

Vd:

```
<?php
    echo "<h1>I'm a lumberjack.</h1>";
    echo "<h2>And I'm okay.</h2>";
?>
```

Kết quả như Hình 2.1.2:



Hình 2.1.2

Định dạng text bằng HTML và PHP

```
<html>
<head>
<title>My First PHP Program</title>
</head>
<body>
<?php
    echo "<h1>I'm a lumberjack.</h1>";
    echo "<h2>And I'm okay.</h2>";
?>
</body>
</html>
```

Việc kết hợp giữa HTML và PHP nhằm đạt tới hai mục tiêu :

- Cải tiến được hình thức trình bày của web site
- Có thể giữ xen kẽ những dòng mã HTML với các đoạn mã của PHP.

Bạn có thể so sánh hai đoạn mã sau:

Vd1:

```
<?php
echo "<table width='100%' border='2' bgcolor='#FFFFFF'>";
echo "<tr>";
echo "<td width='50%'>";
echo "<font face='Verdana, Arial' size='2'>";
echo "First Name:";
echo "</font></td>";
echo "<td width='50%'>";
echo "<font face='Verdana, Arial' size='2'>";
echo $_POST["fname"]
echo "</font></td>";
```

```
        echo "</tr>";
        echo "</table>";
    ?>
```

Kết quả hiển thị ra trình duyệt **Hình 2.2.1**:

A screenshot of a web browser displaying a form. The form consists of a single input field with the text "First Name:" to its left. The input field is empty and has a thin border.

Hình 2.2.1

Vd2:

```
<table width="100%" border="2" bgcolor="#FFFFFF">
<tr>
<td width="50%">
<font face="Verdana, Arial" size="2">
First Name:
</font>
</td>
<td width="50%">
<font face="Verdana, Arial" size="2">
<?
        echo $_POST["fname"];
    ?>
</font>
</td>
</tr>
</table>
```

Kết quả hiển thị như **Hình 2.2.2**:

A screenshot of a web browser displaying a form. The form consists of a single input field with the text "First Name:" to its left. The input field is empty and has a thin border.

Hình 2.2.2

Sử dụng hằng và biến

Tổng quan về hằng:

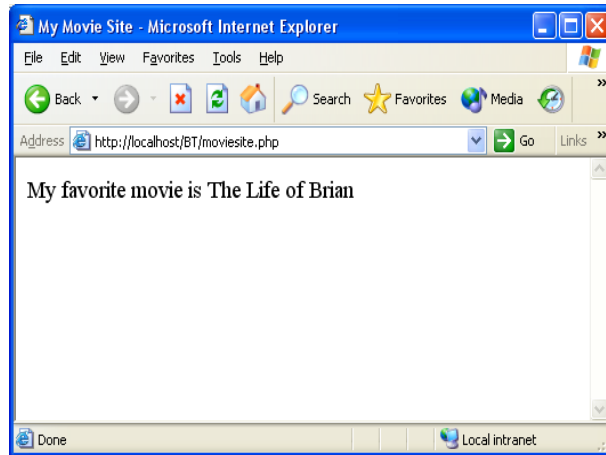
Tương tự như trong những ngôn ngữ lập trình khác, hằng trong PHP cũng là một giá trị được khai báo trước khi sử dụng. Nó phải bắt đầu bằng chữ cái hoặc dấu gạch dưới (_) và không được bắt đầu bằng số, nó có thể là một chuỗi.

Hằng trong PHP được định nghĩa bằng hàm define(). Nó không thay đổi và không được được định nghĩa lại.

Vd: tạo một file mới, lưu với tên moviesite.php

```
<html>
<head>
<title>My Movie Site</title>
</head>
<body>
<?php
    define ("FAVMOVIE", "The Life of Brian");
    echo "My favorite movie is ";
    echo FAVMOVIE;
?>
</body>
</html>
```

Kết quả: Trên trình duyệt xuất hiện như **Hình 2.3.1:**



Hình 2.3.1

Tổng quan về biến.

Không giống như hằng, nó không cần định nghĩa, có thể thay đổi trong chương trình. Cách đặt tên của biến cũng giống như cách đặt tên của Hằng. Biến phân biệt chữ hoa và chữ thường.

Tất cả biến trong PHP đều bắt đầu bằng dấu “\$” bất chấp đó là loại biến gì: chuỗi, số nguyên, số động hoặc mảng, tất cả chúng đều trông giống nhau. Nó tự động hiểu các kiểu biến người dùng không cần phải khai báo kiểu biến.

Nói chung, khi sử dụng biến trong PHP ta cần quan tâm đến 3 vị trí khác nhau của biến:

1. Khai báo ngay trong mã lệnh PHP
2. Chuyển tiếp từ một trang HTML.
3. Là biến sẵn có trong hệ thống PHP.

Vd:

Sử dụng lại ví dụ moviesite.php chỉ thêm vào biến movierate:

```
<html>
<head>
<title>My Movie Site</title>
</head>
<body>
<?php
```

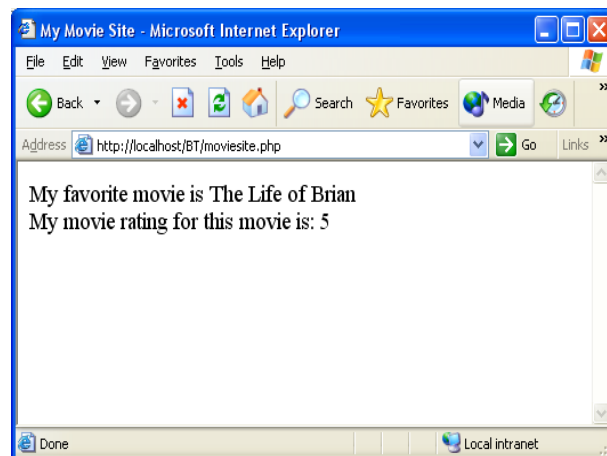
```

define ("FAVMOVIE", "The Life of Brian");
echo "My favorite movie is ";
echo FAVMOVIE;
echo "<br>";
$movierate = 5;
echo "My movie rating for this movie is: ";
echo $movierate;

?>
</body>
</html>

```

Kết quả hiển thị như **Hình 2.3.2.1**:



Hình 2.3.2.1

Chú ý:

- Trong PHP, phép gán là dấu “=”, so sánh là “==”:
- PHP có thể tự động chuyển đổi kiểu trong các phép tính.
- PHP Có thể thực hiện việc cộng một biến số với một chuỗi có số đứng đầu:

Vd:

```

$a = 2 ;
$b = “2 con heo con”;
$c = $a + $b;

```

```
Echo $c;
```

Kết quả : \$c = 4.

PHP có một số hàm toán học được xây dựng dựa trên số mà bạn có thể sử dụng những biến chứa con số như những hàm sau:

- rand(min),(max) phát sinh ra một số ngẫu nhiên.
- ceil(number) làm tròn số thập phân thành một số nguyên kế tiếp lớn hơn.
- floor(number) làm tròn một số thập phân xuống một số nguyên nhỏ hơn.
- number_format(number[,dec place] [,dec point] [,thousands]): Định dạng một số thập phân theo cách : chọn chữ số ở phần thập phân, dấu ngăn cách giữa phần thập phân và phần nguyên, dấu phân cách giữa 3 chữ số của phần nguyên.
- max(*argument1*, *argument2*, ...): Trả về giá trị lớn nhất trong các đối số.
- min(*argument1*, *argument2*, ...): Trả về giá trị nhỏ nhất.

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

Truyền biến qua URL

Phương thức đầu tiên để truyền biến giữa các trang là thông qua một URL.

Ví dụ một URL:

<http://www.mydomain.com/news/articles/showart.php?id=12345>

Những phần sau dấu ? gọi là chuỗi truy vấn (query string). Có thể nối giữa các biến trong URL bằng cách dùng dấu &

Như ví dụ sau:

<http://www.mydomain.com/news/articles/showart.php?id=12345&lang=en>

Ở đây có một vài bất lợi :

- Mọi người có thể nhìn thấy giá trị của biến, những thông tin nhạy cảm

thì không được bảo mật khi sử dụng cách thức này.

- Người dùng có thể đổi giá trị biến trong URL, mở những phần mà người lập trình không muốn hiển thị.
- Một người sử dụng có thể gặp phải những thông tin không chính xác hoặc đã cũ khi dùng URL đã được lưu với những biến cũ trong đó..

Ví dụ: sử dụng biến URL

Sửa file moviesite.php như sau.

```
<html>
<head>
<title>My Movie Site - <?php echo $favmovie; ?></title>
</head>
<body>
<?php
    //xóa dòng: define("FAVMOVIE", "The Life of
    Brian");
    echo "My favorite movie is ";
    echo $favmovie;
    echo "<br>";
    $movierate = 5;
    echo "My movie rating for this movie is: ";
    echo $movierate;
?>
</body>
</html>
```

Mở file mới nhập đoạn mã như sau và lưu với tên movie1.php

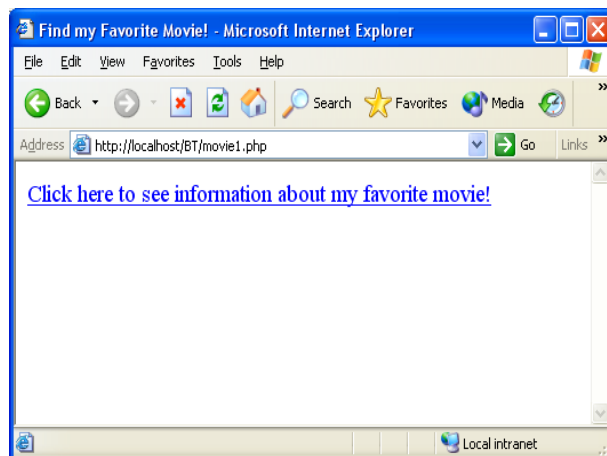
```
<html>
<head>
<title>Find my Favorite Movie!</title>
</head>
```



```
<body>
<?php
    echo "<a href='moviesite.php?favmovie=Stripes'>";
    echo "Click here to see information about my favorite
    movie!";
    echo "</a>";
?>
</body>
</html>
```

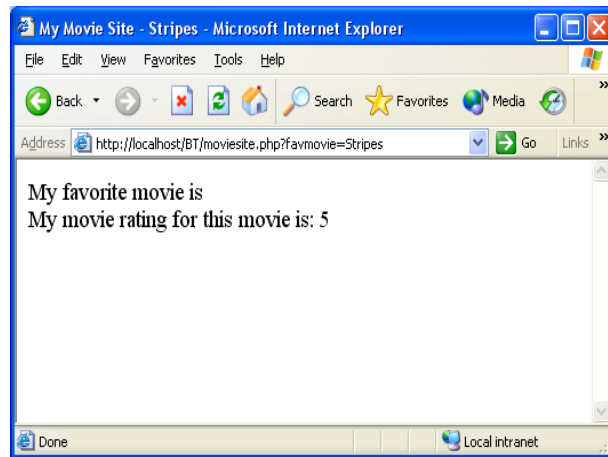
Cho trình duyệt chạy file movie1.php

Kết quả như **Hình 2.4.1.1.1**:



Hình 2.4.1.1.1

Khi click vào liên kết bạn sẽ thấy kết quả như **Hình 2.4.1.1.2**:



Hình 2.4.1.1.2:

Bạn thấy là giá trị của biến \$favmovie là “Stripes” trong URL, phải hiển thị trong **Hình 2.4.1.1.3**, Nhưng ở đây không hiển thị giá trị trong thân chương trình của bạn, nó chỉ là tương tượng. Nếu bạn có điều chỉnh E_ALL trong file php.ini, bạn sẽ thấy thông báo lỗi: “biến không được định nghĩa”.

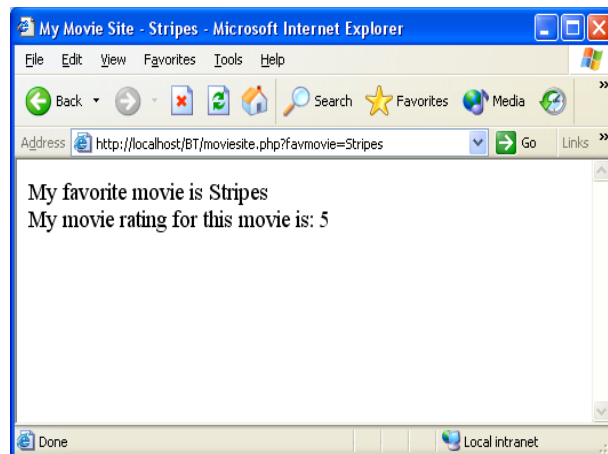
Có cái gì sai? Bạn đoán chính xác nếu bạn nói là biến toàn cục! Đây là một ví dụ đơn giản về việc không khôi phục lại biến trong cách hiệu chỉnh có thể làm cho trang web của bạn không làm việc và làm bạn khó hiểu.

Bây giờ sửa đổi file moviesite.php như sau.

```
<html>
<head>
<title>My Movie Site - <?php echo $_REQUEST['favmovie']; ?></title>
</head>
<body>
<?php
    echo “My favorite movie is “;
    echo $_REQUEST['favmovie'];
    echo “<br>”;
    $movierate = 5;
    echo “My movie rating for this movie is: “;
```

```
echo $movierate;  
?>  
</body>  
</html>
```

Bây giờ cho chạy lại movie1.php vào click vào liên kết sẽ cho kết quả như **Hình 2.4.1.1.3**



Hình 2.4.1.1.3

Chú ý: ở đây có một vài điều cần lưu ý:

- Mã PHP có thể chèn vào bất cứ đâu trong chương trình xen lẫn vào mã HTML (trong ví dụ này nó nằm ở phần “title”).
- Bạn thấy hiệu quả đầu tiên là không cần lấy biến toàn cục để truy cập vào một biến từ trang khác, nhưng cần chú ý rằng khi bạn đề cập đến \$movierate bạn không bao gồm cú pháp của biến toàn cục, bởi vì biến \$movierate nằm trong moviesite.php, bạn không thể lấy thông tin từ trang khác hoặc mã khác.
- \$_REQUEST đã được chọn trong cú pháp biến của bạn bởi vì nó thật sự không có ý nghĩa trong ví dụ này nơi mà giá trị của biến \$favmovie đưa đến. Bạn không có khó khăn gì khi công nhận mọi thứ hoặc ngăn chặn người dùng không được phép vào web site của bạn. Bạn đơn giản chỉ muốn chuyển qua giá trị .

1.1.1.1. *Ký tự đặc biệt của URL.*

Khi truyền biến qua URL , trong trường hợp biến đó chứa khoảng trắng, dấu & hay một ký tự đặc biệt nào đó, thì ta dùng hàm *urlencode()*.

Ví dụ:

```
<html>
<head>
<title>Find my Favorite Movie!</title>
</head>
<body>
<?php
    //thêm dòng:
    $myfavmovie = urlencode("Life of Brian");
    //thay đổi dòng:
    echo "<a href='moviesite.php?favmovie=$myfavmovie'>";
    //echo "<a href='moviesite.php?favmovie=Stripes'>";
    echo "Click here to see information about my favorite
movie!";
    echo "</a>";
?>
</body>
</html>
```

Khi click vào liên kết kết quả như **Hình 2.4.1.2.1**:



Hình 2.4.1.2.1

Session và Cooking

Session

Session là tập hợp những biến tạm tồn tại cho đến khi đóng trình duyệt, trừ khi ta có những chỉ định khác trong php.ini.

Mọi session có một session ID duy nhất, có thể được truyền qua cookie hoặc qua URL nếu phương thức trước không được phép.

Để bắt đầu một session, sử dụng hàm `session_start()` và không được dùng hàm `session_register`, trong trường hợp ta giả định `register_globals` trong php.ini là off.

Ví dụ:

Thêm vào ví dụ movie1.php

```
<?php
    session_start();
    $_SESSION['username'] = "Joe12345";
    $_SESSION['authuser'] = 1;
?>
```

<html>

Thêm vào ví dụ moviesite.php

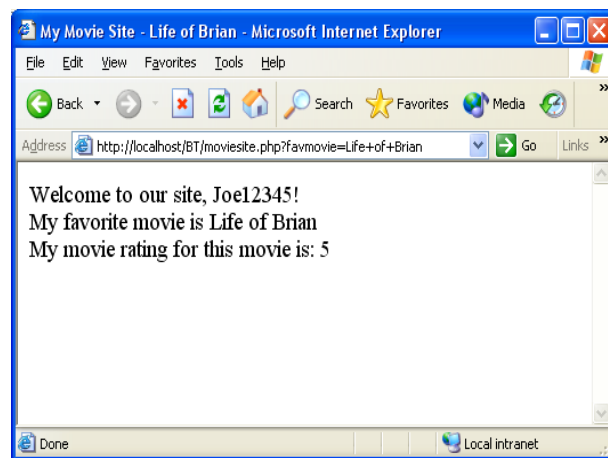
```
<?php
```

```

session_start();
//Kiểm tra để thấy nếu người dùng đăng nhập với một mật khẩu có
sẵn.
if ($_SESSION['authuser'] != 1)
{
    echo "Sorry, but you don't have permission to view this page, you
    loser!";
    exit();
}
?>
<html>
.....
<body>
<?php
    echo "Welcome to our site, ";
    echo $_SESSION['username'];
    echo "! <br>";
?>

```

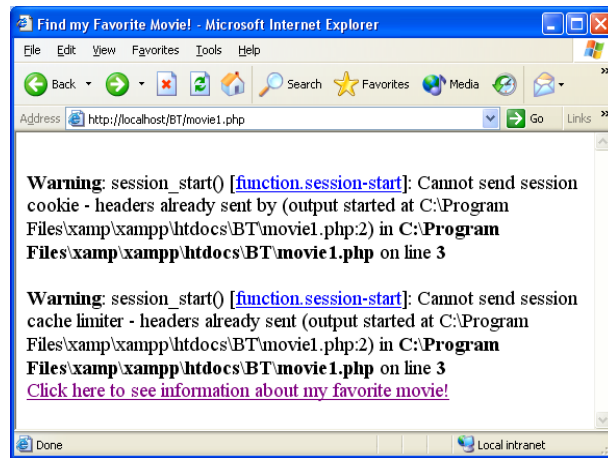
Khi click vào liên kết kết quả như **Hình 2.5.1.1**:



Hình 2.5.1.1

Các thông tin về session phải đặt ngay đầu trang, trước bất kỳ mã HTML

nào. Nếu phần trước đoạn mã PHP tại đầu trang có khoảng trắng thì sẽ có thông báo lỗi như **Hình 2.5.1.2**:



Hình 2.5.1.2:

Nếu sử dụng session variables thì phải dùng hàm `session_start()` ở đầu mỗi trang.

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

Cookies.

Cookies là những mẫu thông tin nhỏ được lưu trên máy người dùng web. Các Cookies do Webserver phát sinh, lưu trữ lại, sau đó sẽ được đọc ở lần truy cập sau.

Để có thể sử dụng một cookies bạn phải sử dụng hàm: `setcookie()` như sau:
`setcookie('tên cookie', 'giá trị', 'thời gian kết thúc', 'đường dẫn', 'vùng', 'kết nối an toàn');`

Ví dụ:

Thêm vào ví dụ movie1.php

```
<?php
setcookie('username', 'Joe', time()+60);
```

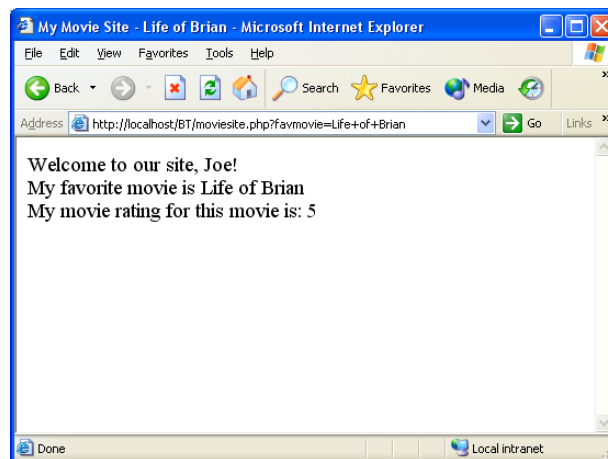


```
//$_SESSION['username']='Joe12345';
```

Thêm vào ví dụ moviesite.php

```
<?php  
echo "Welcome to our site, “;  
echo $_COOKIE['username'];  
//echo $_SESSION['username'];  
echo “! <br>”;
```

Kết quả như **Hình 2.5.2.**:



Hình 2.5.2.

Sau 60 giây thì chữ “Joe” sẽ biến mất(nếu cập nhật lại)

HTML FORMS

Lưu chuyển thông tin với form.

Form là mã trong HTML, nó được bao bởi một cặp thẻ mở và đóng: `<form> </form>`, nó có những thuộc tính sau:

Action: Thuộc tính này chỉ ra đường dẫn URL của trang mà Form sẽ được gửi đến để xử lý, nó chứa đường dẫn URL tương đối hoặc URL tuyệt đối.

Method: Thuộc tính này chỉ ra kiểu HTTP yêu cầu trình duyệt gửi thông tin đến server, nó phải được thiết lập là POST hoặc GET.

Name: Đây là thuộc tính hữu hiệu nhất trong việc quản lý các thành phần của form. Tên form không được gửi đến server khi form được gửi đi.

Các kiểu phần tử nhập trên form:

- **Text:** Đây là kiểu căn bản nó có những thuộc tính sau: Text có 3 thuộc tính:
Size: chiều rộng ô text tính bằng số ký tự.
Maxlength: Giới hạn số ký tự tối đa có thể nhập vào.
Value: chứa giá trị mặc định trong văn bản, người dùng có thể nhập giá trị khác
- **Checkbox:** Tên và giá trị của hộp kiểm này chỉ này chỉ được truyền nếu được chọn khi form được gửi. Nếu từ checked có trong thẻ thì hộp kiểm được chọn mặc định.
- **Radio:** Nút này cho phép người dùng chọn chỉ một trong số các lựa chọn có cùng tên.
- **Select:** Là hộp chọn sổ xuống, thường cho phép người dùng chọn một từ một danh sách, sự lựa chọn này có thể mô tả với thuộc tính value.
- **Password:** Giống với kiểu text, tuy nhiên văn bản nhập vào hộp được hiển thị bằng ký tự thay thế (ký tự “*”)

Những nút khác được miêu tả như nút submit, reset, hoặc những hình ảnh trên những nút do người dùng tạo ra.

Ví dụ :Sử dụng form để lấy thông tin:

Mở file movie1.php và sửa lỗi như sau:

```
<?php
    session_start();
    $_SESSION['username'] = $_POST['user'];
    $_SESSION['userpass'] = $_POST['pass'];
    $_SESSION['authuser'] = 0;
    //Kiểm tra thông tin username và password
    if (($_SESSION['username'] == 'Joe') and
        ($_SESSION['userpass'] == '12345'))
```

```
{
    $_SESSION['authuser'] = 1;
} else
{
    echo "Sorry, but you don't have permission to view this
page, you loser!";
    exit();
}
?>
```

Sau đó sửa trong moviesite.php như sau:

```
//xóa dòng này: echo $_COOKIE['username'];
echo $_SESSION['username'];
```

Tạo một file mới lưu với tên login.php

```
<?php
    session_unset();
?>
<html>
<head>
<title>Please Log In</title>
</head>
<body>
    <form method="post" action="movie1.php">
    <p>Enter your username:
    <input type="text" name="user">
    </p>
    <p>Enter your password:
    <input type="password" name="pass">
    </p>
    <p>
```

```
<input type="submit" name="Submit" value="Submit">
```

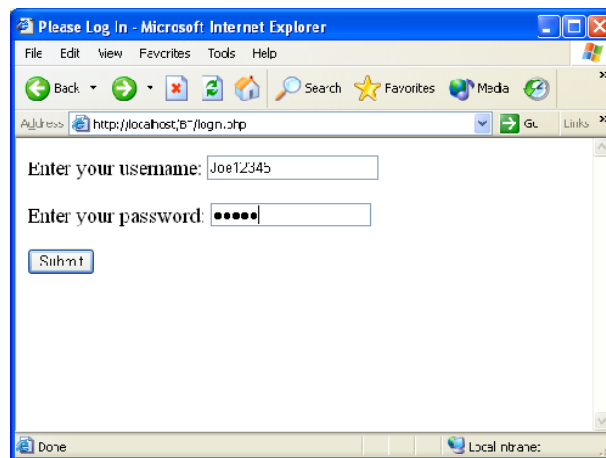
```
</p>
```

```
</form>
```

```
</body>
```

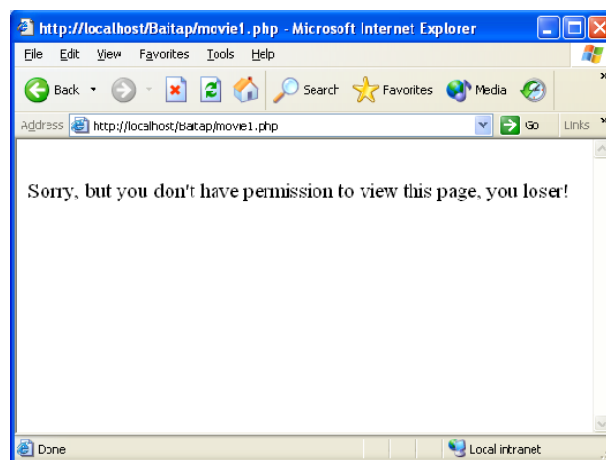
```
</html>
```

Mở trình duyệt chạy file login.php, nhập vào user name: Joe12345, password: 12345, bạn sẽ thấy kết quả hiển thị như **Hình 2.6.1.1**:



Hình 2.6.1.1

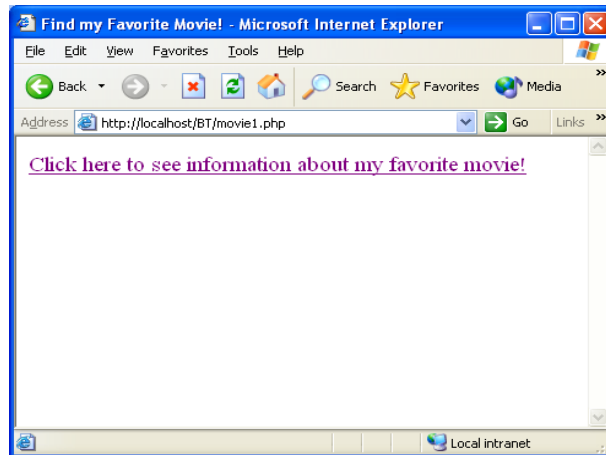
Khi click vào Submit, sẽ thấy kết quả như **Hình 2.6.1.2**:



Hình 2.6.1.2

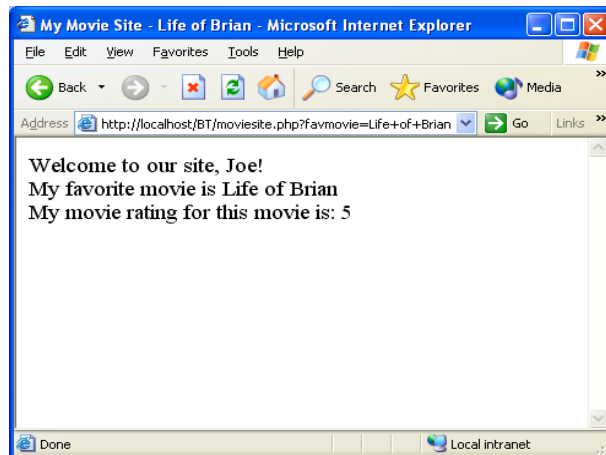
Nếu bây giờ ta nhập user name là Joe và password là 12345 thì kết quả như **Hình**

2.6.1.3:



Hình 2.6.1.3

Khi click vào liên kết thì kết quả như **Hình 2.6.1.4**:



Hình 2.6.1.4

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

CÂU LỆNH IF/ELSE

Mệnh đề if.

Khác với những ngôn ngữ lập trình khác, PHP sử dụng mệnh đề if riêng lẻ, cú pháp như sau:

if (điều kiện) mã lệnh được thi hành nếu điều kiện đúng;

Ví dụ:

```
if ($stockmarket >= 10000) echo "Hooray! Time to Party!";
```

Trong trường hợp mệnh đề if có nhiều câu lệnh thì ta dùng dấu “{}” để bao bọc nó.

Ví dụ

```

if ($stockmarket >= 10000)
{
    echo "Hooray! Time to Party!";
    $mood = "happy";
    $retirement = "potentially obtainable";
}

```

Các phép toán dùng để so sánh trong mệnh đề if

Operator	Appropriate Syntax
Bằng	: ==
Không bằng	: != or <>
Lớn hơn	: >
Nhỏ hơn	: <
Lớn hơn hoặc bằng	: >=
Nhỏ hơn hoặc bằng	: <=
Bằng, và kiểu dữ liệu thỏa kiểu dữ liệu giống nhau	: ===
Không bằng hoặc kiểu dữ liệu không giống nhau	: !==

Ví dụ:

```

<html>
<head>
<title>How many days in this month?</title>
</head>
<body>
<?php
    $month = date("n");
    if ($month == 1) echo "31";
    if ($month == 2) echo "28 (unless it's a leap year)";
    if ($month == 3) echo "31";
    if ($month == 4) echo "30";
    if ($month == 5) echo "31";

```

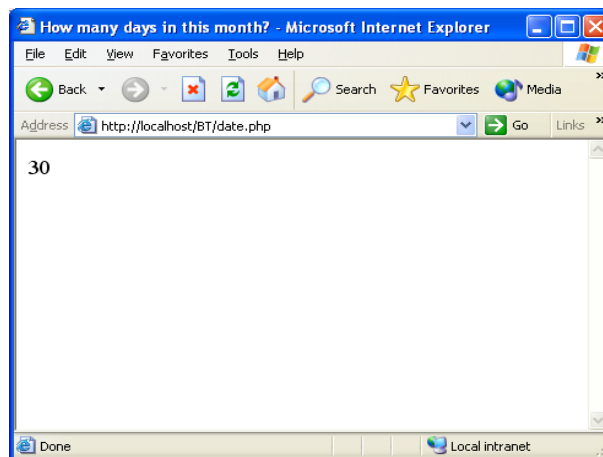
```

if ($month == 6) echo "30";
if ($month == 7) echo "31";
if ($month == 8) echo "31";
if ($month == 9) echo "30";
if ($month == 10) echo "31";
if ($month == 11) echo "30";
if ($month == 12) echo "31";

?>
</body>
</html>

```

Kết quả như **Hình 2.7.1.1** :



Hình 2.7.1.1

Hoạt động của mệnh đề **if** tương tự như những ngôn ngữ lập trình khác .

Điểm lưu ý trong ví dụ này là hàm *date("n")* sẽ cho giá trị tháng hiện hành khi người truy cập vào website.

Sử dụng if và else lồng nhau.

Trong một số trường hợp dùng mệnh đề “if” thì tốt, nhưng đôi khi kết hợp giữa “if” và “else” sẽ cho ta kết quả như mong muốn.

```

<html>
<head>

```

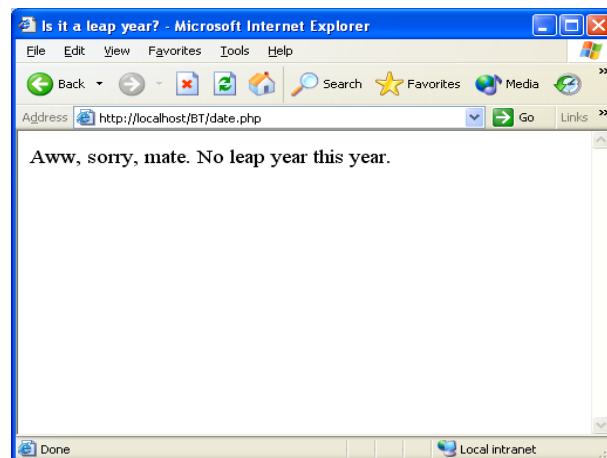


```

<title>Is it a leap year?</title>
</head>
<body>
    <?php
        $leapyear = date("L");
        if ($leapyear == 1) echo "Hooray! It's a leap year!";
        else echo "Aww, sorry, mate. No leap year this year.";
    ?>
</body>
</html>

```

Kết quả như **Hình 2.7.2**:



Hình 2.7.2

Sử dụng include cho đoạn mã có hiệu quả hơn

Khi thực hiện viết lệnh trong PHP để tránh sự lặp đi lặp lại không cần thiết ta dùng hàm *include*.

Để hiểu rõ vấn đề này chúng ta hãy xét ví dụ sau:

Tạo một file mới đặt với tên header.php

```

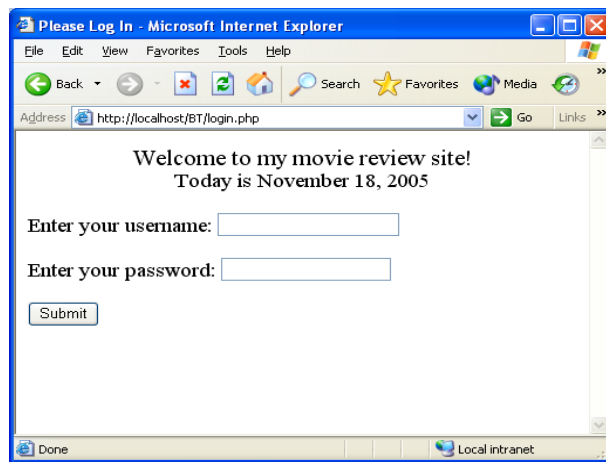
<div align="center"><font size="4">Welcome to my movie review site!</font>
<br>
<?php
    echo "Today is ";

```

```
echo date("F d");  
echo ", ";  
echo date("Y");  
?>  
</div>
```

Sau đó thêm vào các file `movie1.php`, `moviesite.php` và `login.php` câu lệnh sau ngay sau thẻ `<body>`: `<?php include "header.php"; ?>`

Mở trình duyệt với tên `login.php` bạn sẽ có kết quả như **Hình 2.8**:



Hình 2.8

Hoạt động: Khi PHP gặp một include trong đoạn mã script, nó ngừng hoạt động và ngay lập tức tìm đến file được chỉ dẫn bởi include. Server phân tích file này và trả kết quả trở lại file gốc và việc phân tích mã lại tiếp tục tại nơi đã dừng trước đó.

Sử dụng Hàm cho đoạn mã có hiệu quả hơn

Như với include, Hàm làm cho đoạn mã của bạn hiệu quả hơn và dễ dàng biên dịch hơn. Hàm là một khối lệnh có thể được gọi bất cứ nơi nào trong chương trình. Chúng thể sử dụng lại ở bất cứ khi nào. Nó có thể giúp ta đặt hoặc cập nhật biến và có thể xếp lồng vào nhau. Bạn cũng có thể tạo một hàm chỉ được thực thi khi điều kiện nào đó thỏa mãn.

Các hàm tự chúng là các chương trình nhỏ. Chúng không biết bất cứ biến nào xung

quanh nó trừ khi bạn khai báo biến toàn cục. Có thể dùng khai báo global \$tên_biến để tạo biến toàn cục. Điều này không cần với những biến toàn cục mặc định như POST, GET.

Ví dụ:

Mở movie1.php thêm vào đoạn mã như sau:

```
echo "<br>";
echo "<a href='moviesite.php?movienum=5'>";
echo "Click here to see my top 5 movies.";
echo "</a>";
echo "<br>";
echo "<a href='moviesite.php?movienum=10'>";
echo "Click here to see my top 10 movies.";
echo "</a>";
?>
</body>
</html>
```

Mở moviesite.php thêm vào phần sau:

```
<?php

function listmovies_1()
{
echo "1. Life of Brian<br>";
echo "2. Stripes<br>";
echo "3. Office Space<br>";
echo "4. The Holy Grail<br>";
echo "5. Matrix<br>";
}
}
```

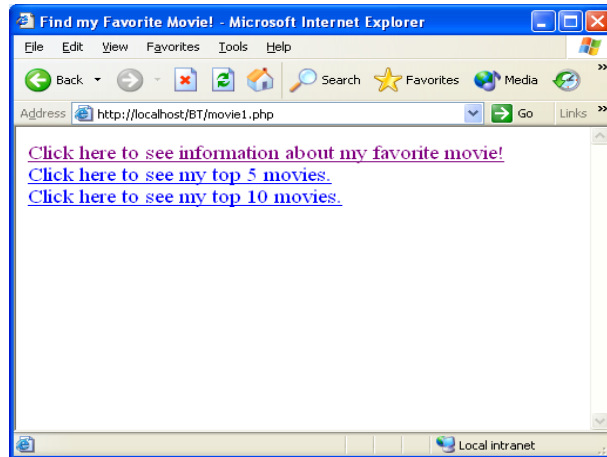
```

function listmovies_2()
{
echo "6. Terminator 2<br>";
echo "7. Star Wars<br>";
echo "8. Close Encounters of the Third Kind<br>";
echo "9. Sixteen Candles<br>";
echo "10. Caddyshack<br>";
}
if (isset($_REQUEST['favmovie']))
{
-----
} else {
echo "My top ";
echo $_REQUEST['movienum'];
echo " movies are:";
echo "<br>";
listmovies_1();
if ($_REQUEST['movienum'] == 10) listmovies_2();
}

```

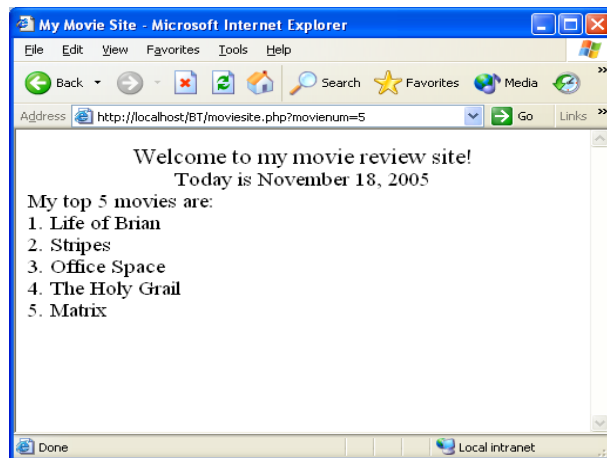
Bây giờ ta cho chạy file login.php : đăng nhập với tên username là Joe, password là 12345

Bạn sẽ thấy kết quả như sau **Hình 2.9.1**:



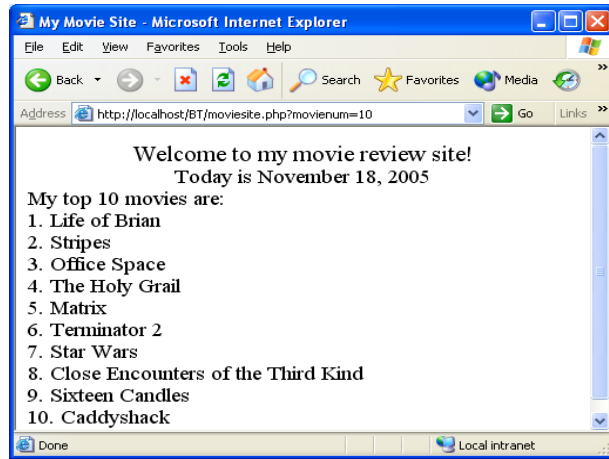
Hình 2.9.1

Khi click vào liên kết thứ hai ta sẽ thấy như **Hình 2.9.2**:



Hình 2.9.2

Khi click vào liên kết thứ ba ta sẽ thấy như **Hình 2.9.3**:



Hình 2.9.3

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

Mảng

Mảng là một danh sách chứa thông tin với khóa và được lưu trữ với một tên biến. Một mảng có dòng thông tin và khóa, khóa là cột đầu tiên trong bảng. Khóa phải đồng nhất để dễ tổ chức và sử dụng.

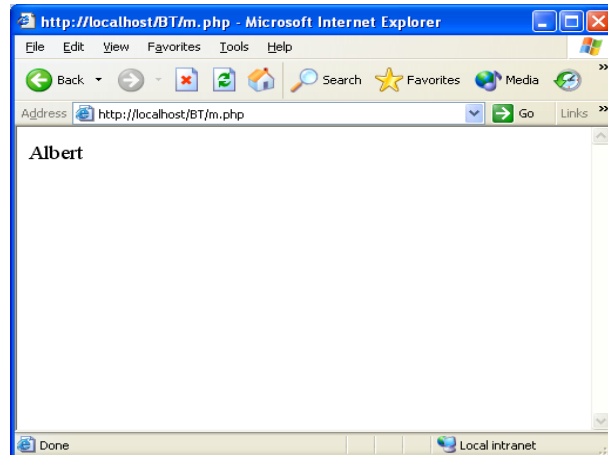
Cú pháp của mảng.

Mảng một chiều

```
<?php
$husband = array("firstname"=>"Albert",
```

```
"lastname"=>"instei",  
"age"=>"124");  
echo $husband[firstname];  
?>
```

Kết quả xuất ra màn hình **Hình 2.10.1.1**:



Hình 2.10.1.1

Chú ý: Sử dụng => để thay cho dấu bằng, khi ấn định những giá trị đến khóa của mảng. Điều này cho bạn xuất ra “Albert” và tất cả giá trị vẫn được lưu trữ trong biến \$husband.

Bạn cũng có thể đặt một mảng giá trị theo cách sau:

```
$husband[“firstname”] = “Albert”;  
$husband[“lastname”] = “Einstein”;  
$husband[“age”] = 124;
```

Mảng đa chiều:

Ví dụ:

```
<?php  
$table1 = array(“husband” => array(“firstname”=>”Albert”,  
“lastname”=>”Einstein”,
```



```

“age”=>124),
“wife” => array(“firstname”=>”Mileva”,
“lastname”=>”Einstein”,
“age”=>123));
//do the same for each table in your restaurant
?>

```

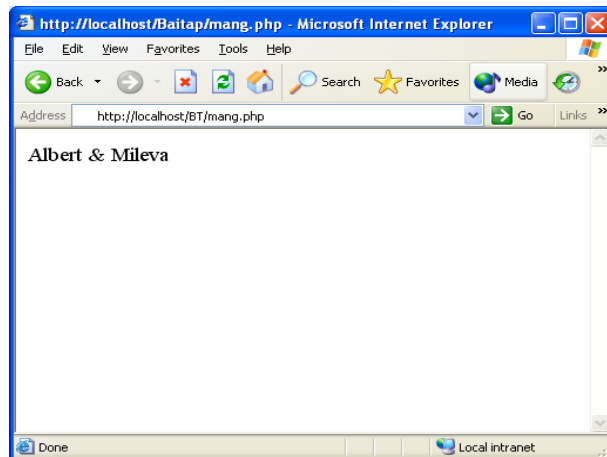
Nếu muốn xuất firstname của mọi người, ta dùng hàm xuất như sau:

```

<?php
echo $table1[“husband”][“firstname”];
echo “ & “;
echo $table1[“wife”][“firstname”];?>

```

Kết quả Hình 2.10.1.2:



Hình 2.10.1.2

Để đơn giản ta có thể dùng một cách khác để lưu trữ mảng như sau:

```

<?php
$flavor[] = “blue raspberry”;
$flavor[] = “root beer”;
$flavor[] = “pineapple”;
?>

```

Để xuất thông tin ra ta thực hiện câu lệnh như sau:

```

echo $flavor[0]; //xuất “blue raspberry”

```

```
echo $flavor[1]; // xuất "root beer"
```

```
echo $flavor[2]; // xuất "pineapple"
```

Sắp xếp mảng

PHP cung cấp nhiều cách để lưu trữ giá trị của mảng, sau đây là mô tả một vài hàm sắp xếp trong mảng.

arsort(array) :Sắp xếp một mảng giảm theo giá trị và duy trì quan hệ của khóa/ giá trị

asort(array): Sắp xếp một mảng tăng theo giá trị và duy trì quan hệ của khóa/ giá trị

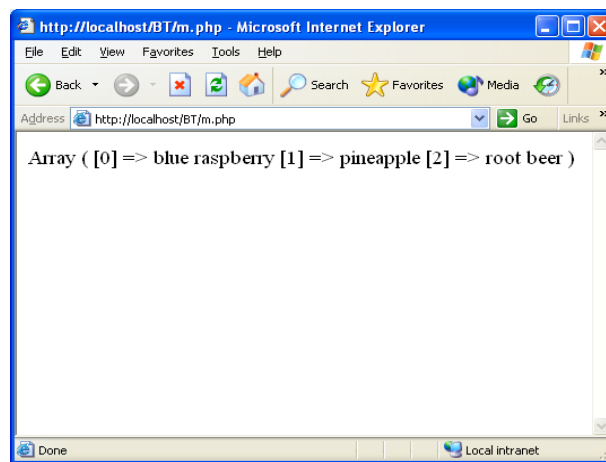
rsort(array): Sắp xếp một mảng giảm theo giá trị

sort(array): Sắp xếp một mảng tăng theo giá trị

Ví dụ:

```
<?php
    $flavor[] = "blue raspberry";
    $flavor[] = "root beer";
    $flavor[] = "pineapple";
    sort($flavor);
    print_r($flavor);
?>
```

Kết quả như **Hình 2.10.1.3**:



Hình 2.10.1.3

Chú ý:

Ở đây dùng hàm mới `printf_r`, chức năng của hàm này là xuất thông tin về một biến mà người ta có thể đọc nó dễ dàng. Nó thường được dùng để kiểm tra giá trị của mảng, một cách cụ thể.

Ở đây hàm `sort` dùng để sắp xếp theo thứ tự Alpha

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

Cấu trúc Foreach

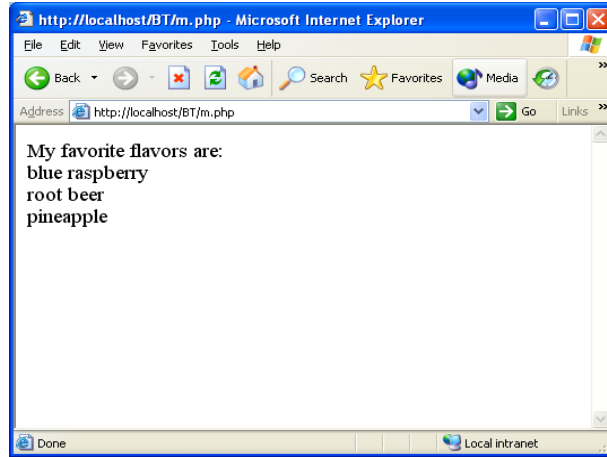
Ví dụ:

```
<?php
    $flavor[] = "blue raspberry";
    $flavor[] = "root beer";
    $flavor[] = "pineapple";
    echo "My favorite flavors are:<br>";
    foreach ($flavor as $currentvalue) {
        //these lines will execute as long as there is a value in $flavor
        echo $currentvalue . "<br>\n";
```

```
}
```

```
?>
```

Kết quả như **Hình 2.10.2.1**:



Hình 2.10.2.1

Khi PHP xử lý trong mảng. Nó giữ vị trí của các khóa bằng cách sử dụng con trỏ trong mảng.

Khi hàm foreach được gọi con trỏ chờ giá trị đầu tiên trong mảng. Cuối hàm con trỏ chuyển xuống khóa cuối cùng trong mảng. Vị trí của con trỏ có thể là công cụ trợ giúp.

Ví dụ:

1/ Thay đổi moviesite như sau:

```
<?php
    session_start();
    //kiểm tra username và password
    if ($_SESSION['authuser'] != 1) {
        echo "Sorry, but you don't have permission to view this
        page, you loser!";
        exit();
    }
?>
<html>
```

```
<head>
<title>My Movie Site</title>
</head>
<body>
<?php include "header.php"; ?>
<?php
    $favmovies = array("Life of Brian",
        "Stripes",
        "Office Space",
        "The Holy Grail",
        "Matrix",
        "Terminator 2",
        "Star Wars",
        "Close Encounters of the Third Kind",
        "Sixteen Candles",
        "Caddyshack");
    /*/delete these lines:
function listmovies_1()
{
    echo "1. Life of Brian<br>";
    echo "2. Stripes<br>";
    echo "3. Office Space<br>";
    echo "4. The Holy Grail<br>";
    echo "5. Matrix<br>";
}
function listmovies_2()
{
    echo "6. Terminator 2<br>";
    echo "7. Star Wars<br>";
```

```

        echo "8. Close Encounters of the Third Kind<br>";
        echo "9. Sixteen Candles<br>";
        echo "10. Caddyshack<br>";
    }
//kết thúc dòng xóa*/
if (isset($_REQUEST['favmovie']))
{
    echo "Welcome to our site, ";
    echo $_SESSION['username'];
    echo "! <br>";
    echo "My favorite movie is ";
    echo $_REQUEST['favmovie'];
    echo "<br>";
    $movierate = 5;
    echo "My movie rating for this movie is: ";
    echo $movierate;
} else
{
    echo "My top 10 movies are:<br>";
if (isset($_REQUEST['sorted']))
{
    sort($favmovies);
}

    /*/delete these lines
    echo $_REQUEST['movienum'];
    echo " movies are:";
    echo "<br>";
    listmovies_1();
if ($_REQUEST['movienum'] == 10) listmovies_2();

```

```
//end of deleted lines*/
foreach ($favmovies as $currentvalue)
{
    echo $currentvalue;
    echo "<br>\n";
}
}
?>
</body>
</html>
```

2/ Thay đổi movie1.php như sau:

```
<?php
    session_start();
    $_SESSION['username'] = $_POST['user'];
    $_SESSION['userpass'] = $_POST['pass'];
    $_SESSION['authuser'] = 0;
    // kiểm tra username và password
    if (($_SESSION['username'] == 'Joe') and
        ($_SESSION['userpass'] == '12345'))
    {
        $_SESSION['authuser'] = 1;
    } else {
        echo "Sorry, but you don't have permission to view this
        page, you loser!";
        exit();
    }
?>
<html>
```

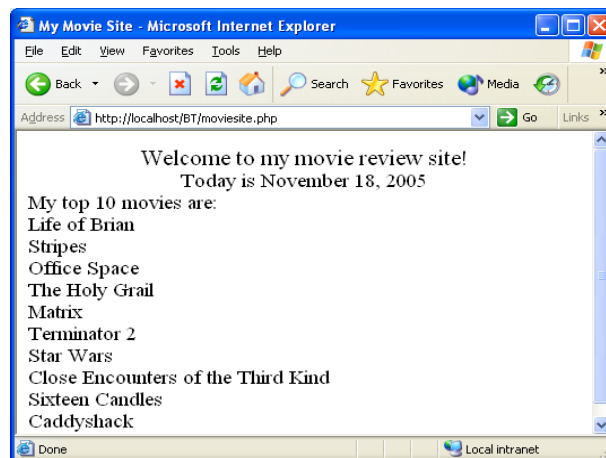


```
<head>
<title>Find my Favorite Movie!</title>
</head>
<body>
<?php include "header.php"; ?>
<?php
    $myfavmovie = urlencode("Life of Brian");
    echo "<a href='moviesite.php?favmovie=$myfavmovie'>";
    echo "Click here to see information about my favorite movie!";
    echo "</a>";
    echo "<br>";
    /*/delete these lines
    echo "<a href='moviesite.php?movienum=5'>";
    echo "Click here to see my top 5 movies.";
    echo "</a>";
    echo "<br>";
    //end of deleted lines*/
    //change the following line:
    echo "<a href='moviesite.php'>";
    echo "Click here to see my top 10 movies.";
    echo "</a>";
    echo "<br>";
    echo "<a href='moviesite.php?sorted=true'>";
    echo "Click here to see my top 10 movies, sorted alphabetically.";
    echo "</a>";
?>
</body>
</html>
```

3/ Cho chạy login.php và click vào dòng

[Click here to see my top 10 movies.](#)

Bạn sẽ thấy kết quả như **Hình 2.10.2.2** :

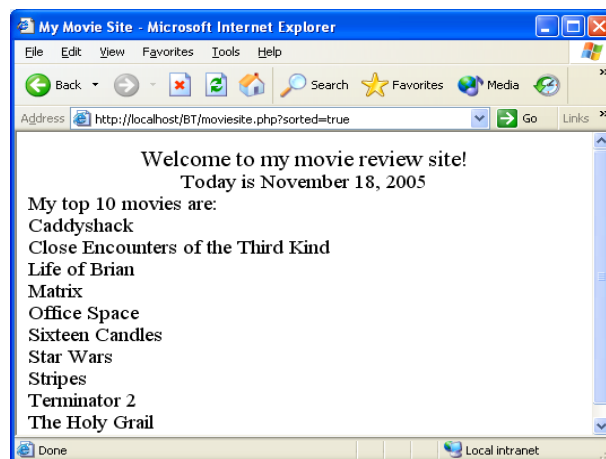


Hình 2.10.2.2

Quay lại file movie1.php click vào:

[Click here to see my top 10 movies, sorted alphabetically.](#)

Bạn sẽ thấy kết quả như sau như **Hình 2.10.2.3**:



Hình 2.10.2.3

Ở đây đã có sự sắp xếp theo thứ tự alpha

Cách thức hoạt động :

Trước tiên bạn đặt danh sách động trong biến, \$favmovies với mảng. Sau đó bạn có thể để danh sách movie từng cái một bằng cách sử dụng foreach trong file moviesite.php. Bạn cũng có thể thêm một liên kết để người sử dụng sắp xếp danh sách theo alphabel bằng cách dùng biến có tên \$_REQUEST[sorted].

Khi biến này là true thì hàm sort() được thực thi và bạn truyền biến true qua URL trong liên kết.

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

While và do.. while

Như ta đã thấy hàm foreach hoạt động trên mỗi phần tử của mảng. Ta cũng có thể dùng câu lệnh while để làm điều đó.

Ví dụ:

Sử dụng vòng lặp while để in ra dãy số từ 1→5

```
$num = 1;
while ($num <= 5)
{
    echo $num;
    echo “<br>”;
    $num = $num + 1;
}
```

Tương tự với do .. while cũng cho ra cùng kết quả.

```
$num = 1;
do
{
    echo $num;
    echo "<br>";
    $num = $num + 1
}
while ($num <= 5);
```

Ví dụ : về việc sử dụng hàm while:

1. Thay đổi movie1.php như sau:

```
<?php
    session_start();
    $_SESSION['username'] = $_POST['user'];
    $_SESSION['userpass'] = $_POST['pass'];
    $_SESSION['authuser'] = 0;
// kiểm tra username và password
    if (($_SESSION['username'] == 'Joe') and $_SESSION['userpass'] ==
    '12345')) {
        $_SESSION['authuser'] = 1;
    } else {
        echo "Sorry, but you don't have permission to view this
        page, you loser!";
        exit();
    }
?>
<html>
<head>
<title>Find my Favorite Movie!</title>
```

```

</head>
<body>
<?php include "header.php" ?>
<?php
    $myfavmovie=urlencode("Life of Brian");
    echo "<a href='moviesite.php?favmovie=$myfavmovie'>";
    echo "Click here to see information about my favorite movie!";
    echo "</a>";
    echo "<br>";
    /*
        echo "<a href='moviesite.php'>";
        echo "Click here to see my top 10 movies.";
        echo "</a>";
        echo "<br>";
        echo "<a href='moviesite.php?sorted=true'>";
        echo "Click here to see my top 10 movies, sorted alphabetically.";
        echo "</a>";
    */
    echo "Or choose how many movies you would like to see.";
    echo "</a>";
    echo "<br>";
?>
<form method="post" action="moviesite.php">
<p>Enter number of movies (up to 10):
<input type="text" name="num">
<br>
Check here if you want the list sorted alphabetically:
<input type="checkbox" name="sorted">
</p>

```

```
<input type="submit" name="Submit" value="Submit">
</form>
</body>
</html>
```

2. Thay đổi moviesite.php:

```
<?php
    session_start();
    //check to see if user has logged in with a valid password
    if ($_SESSION['authuser'] != 1) {
        echo "Sorry, but you don't have permission to view this
        page, you loser!";
        exit();
    }
?>
<html>
<head>
<title>My Movie Site</title>
</head>
<body>
<?php include "header.php"; ?>
<?php
    $favmovies = array("Life of Brian",
        "Stripes",
        "Office Space",
        "The Holy Grail",
        "Matrix",
        "Terminator 2",
        "Star Wars",
        "Close Encounters of the Third Kind",
```

```

“Sixteen Candles”,
“Caddyshack”);
if (isset($_REQUEST[‘favmovie’])) { //isset(xac dinh mot bien co trong
//tap hop khong)
    echo “Welcome to our site, “;
    echo $_SESSION[‘username’];
    echo “! <br>”;
    echo “My favorite movie is “;
    echo $_REQUEST[‘favmovie’];
    echo “<br>”;
    $movierate = 5;
    echo “My movie rating for this movie is: “;
    echo $movierate;
} else {
    echo “My top “. $_POST[“num”] . “ movies are:<br>”;
    if (isset($_REQUEST[‘sorted’])) {
        sort($favmovies);
    }
//list the movies
$numlist = 1;
while ($numlist <= $_POST[“num”])
{
    echo $numlist;
    echo “. “;
    echo pos($favmovies);
    next($favmovies);
    echo “<br>\n”;
    $numlist = $numlist + 1;
}

```

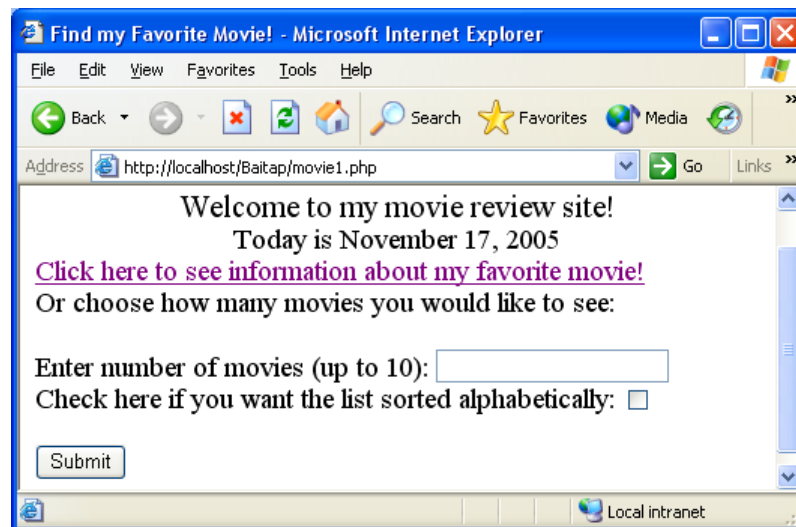


```

/*
    foreach ($favmovies as $currentvalue) {
    echo $currentvalue;
    echo "<br>\n";
    }
*/
}
?>
</body>
</html>

```

Kết quả Hình 2.11



Hình 2.11

Khi điền vào ô một số và đánh dấu check vào ô kiểm tra thì kết quả sẽ hiện ra danh sách có số phần tử bằng với con số nhập vào và sắp xếp theo thứ tự alphabet.

Chú ý:

- Chúng ta thêm vào echo một vài thủ thuật:

```
Echo" My top".$_POST["num"]." movie are:<br>";
```

Với phương thức này bạn có thể gặp vài lỗi và dấu nhảy kếp(“) không được nhận ra.

- Bạn đặt \$numlist là 1, và điều này kiểm tra con số bạn đặt.
- Bạn sử dụng biến \$_POST["num"] để đặt giới hạn danh sách bạn cho; con số này được cho bởi người sử dụng trong file movie1.php
- Hàm pos(\$favmovies) cũng là hàm mới. Hàm này trả về giá trị hiện hành nơi mà con trỏ được bắt đầu. Bạn muốn thấy giá trị hiện hành thì xuất ra hàm này.
- Hàm next(\$favmovies) cũng là hàm mảng khác mà con trỏ của mảng chỉ tới giá trị tiếp theo trong hàng. Điều này dễ dàng lặp lại trong mệnh đề tiếp theo.

Chương 2: TẠO NHỮNG TRANG PHP SỬ DỤNG PHP5

Chương 2 giới thiệu những phần sau:

1. Sử dụng “echo” để trình bày text.
2. Định dạng text bằng HTML và PHP.
3. Hằng và biến.
4. Sử dụng URL để truyền biến .
5. Sessions và cookies.
6. HTML forms.
7. Mệnh đề if/else.
8. Includes.
9. Functions.
10. Array và foreach.
11. While và do/while.
12. Sử dụng lớp và thứ tự với OOP.

Những cú pháp xen lẫn trong php.

Xen lẫn giữa <?php và ?>

Bạn có thể xem code php trong html trong cách khác

- <? và ?> Phải mở trong file php.ini với thẻ mở ngắn.
- <% và %> Phải mở trong file php.ini với thẻ **ASP**
- <script language="PHP"> and </script>. Đây là giá trị không chuyển đổi trong file php.ini

Xen lẫn trong mệnh đề echo

Bạn đã sử dụng print_r(), nhưng bạn cũng đã sử dụng print() để trình bày đoạn text hoặc giá trị biến trong trang web của bạn. Sự khác nhau giữa echo() và printf() là khi bạn sử dụng print(), một giá trị 0 hoặc 1 sẽ được trả lại cho sự thành

công hay thất bại của lệnh print(). Trong trường hợp này, bạn cũng có thể nói mọi thứ không in ra khi sử dụng lệnh print(). Trong khi echo() chỉ đưa ra những gì không có mà bạn biết là không có hoặc nó làm việc không chính xác.

Trong các trường hợp khác thì cả hai là như nhau.

Xen lẫn giữa các phép toán logic.

Các phép toán and(&&), or(||) là các phép toán logic.

Xen lẫn trong việc sử dụng dấu (“”).

Ngoài việc sử dụng dấu (“”) ta cũng có thể sử dụng dấu (= <<<)

Xen lẫn những giá trị tăng, giảm

- ++\$value: Tăng biến lên 1, trả về giá trị đã tăng.
- \$value++: Trả về giá trị, sau đó tăng lên 1.
- --\$value: Giảm 1, trả về giá trị đã giảm.
- \$value--: Trả về giá trị, sau đó giảm 1.
- \$value=\$value+1: Tăng giá trị lên 1.
- \$value+=1: Tăng giá trị lên 1.

Tổng quan về OOP

Bạn có hoặc không nghe những thứ vớ vẩn về PHP5 và việc sử dụng OOP. OOP nghĩa là lập trình hướng đối tượng và trong khi nó không là đoạn mã logic tốt nhất nhưng có thể cung cấp một vài tập lệnh tốt. Số lượng lớn về OOP trong PHP5 thông qua hệ phương pháp OOP có thể được chấp nhận trong PHP4. Với sự ra đời PHP5 nó trở nên dễ sử dụng và thực hiện. Khi một người bắt đầu, Bạn sẽ không cần nghiên cứu sâu về ngôn ngữ OOP, nhưng nó quang trọng để bạn hiểu những khái niệm đằng sau OOP.

Trong bảng tóm tắt OOP lấy những hàm cập nhật thông thường và thay vì đặt chúng trong include như bạn làm ở trước thì bạn đặt chúng trong một lớp. Một lớp là 1 tập hợp những biến và hàm mà xuất ra khi được gọi. Đối tượng là những kết quả từ lớp.

Tóm tắt những ví dụ về OOP

Sử dụng OOP giống như thứ tự tại một quầy pizza. Nó không làm bạn tăng cân và có một đôi chân đẹp, nhưng nó yêu cầu phải vận động:

Trước tiên người bồi bàn sẽ nhận yêu cầu của bạn và đến nhà bếp. Anh ta đề nghị một cái bánh pizza thích hợp với yêu cầu của bạn để nấu. Nhà bếp sẽ coi công thức làm cái bánh đó và cần thêm người phụ làm. Sau đó nhà bếp làm bề mặt của bánh pizza và nướng nó trong một thời gian. Cuối cùng họ sẽ mang bánh pizza cho người bồi bàn.

Trong ví dụ này những cách thức nhào bột, làm bề mặt của bánh, nấu và lấy ra từ lò. Những thành phần này là những đặt tử. Đối tượng của bạn là một bánh pizza.

Nếu chúng ta thể hiện kinh nghiệm làm bánh trong PHP/hệ phương pháp OOP thì nó như sau:

```
<?php
//đây là một lớp của chúng ta.
class Pizza
{
    public $dough;
    public $toppings;
    public function MakeDough($dough)
    {
        $this->dough = $dough;
        //lăn bột $this->dough
    }
    public function addToppings($toppings)
    {
        $this->toppings = $toppings;
        //chia nhỏ $this->toppings;
        //đặt $this->toppings on dough;
```

```

    }
    public function bake()
    {
        //nướng bánh
        return true;
    }
    public function make_pizza($dough, $toppings)
    {
        //làm bánh
        $step1 = $this->MakeDough($dough);
        if ($step1)
        {
            $step2 = $this->addToppings($toppings);
        }
        if ($step2)
        {
            $step3 = $this->bake();
        }
    }
}
?>

```

Sau đó bạn có thể tạo ra bánh pizza bất cứ lúc nào bạn cảm thấy thích và bạn có thể chắc rằng cái bánh được tạo đúng.

```

<?php
    //đây là tập lệnh PHP của chúng ta
    $table1 = new Pizza();
    $table1->make_pizza('hand-tossed', 'pepperoni');
    if ($table1->bake())
    {

```

```

        //phân phối $pizza cho bảng 1;
    }
    else echo "uh-oh, looks like you should have gone to eat fast food.";
?>

```

Rõ ràng, nếu bạn chạy tập lệnh này nó sẽ không làm việc, đây chỉ là một hiển thị đơn giản. Bây giờ bạn có thể thấy cách tạo một cái bánh pizza như thế nào bất cứ khi nào bạn muốn, bằng cách sử dụng những biến như: \$dough1, \$stoppings1, \$pizza1, \$dough2, \$stoppings2, \$pizza2, table1, table2. Bất cứ lúc nào một người yêu cầu một cái pizza bạn có thể gọi lớp pizza và một cái bánh sẽ được tạo ra. Và khi khác một người khác yêu cầu một cái bánh bạn cũng làm như trên.

Một vài chú ý:

- Đặt tên lớp là sử dụng trộn lẫn chữ hoa và chữ thường.
- Nếu muốn một hàm có thể sử dụng ở mọi nơi, ta cần khai báo function `__construct()`, khai báo như hàm đầu tiên trong lớp, được gọi là *constructor*

- Ví dụ

```

function __construct()
{
    $this->tray = $round;
}

```

- `$this->variable` tương tự như trong cú pháp mảng, `$this` có thể là một đối tượng đặt biệt được tạo ra.
- Bạn để ý rằng lớp của bạn bắt đầu với dòng biến ban đầu. Bạn cần khai báo một biến trong phạm vi một lớp. Bạn khai báo biến như “public,” “private,” hoặc “protected”. Biến public có thể thấy được ở bất kỳ lớp nào, biến private chỉ thấy trong lớp của nó, biến protected thấy trong lớp của nó và bất kỳ lớp nào mà có dòng mở rộng ở đầu. Nó có thể đồng ý để giữ hầu hết các biến của bạn như public, ngoại trừ những cái nào chứa thông tin cá nhân.
- Để tạo đối tượng, sử dụng từ khóa *new*

Ví dụ:

```
$table1 = new Pizza();
```

Điều này giữ tất cả các thông tin về bánh pizza trong biến \$table1

Để đơn giản, bạn tạo một hàm trong lớp của bạn mà nó gọi tất cả những hàm khác theo thứ tự bạn muốn. Nếu bạn được “carb-conscious” và tránh trộn bột lẫn vào nhau, quyết định không nướng bánh pizza. Bạn có thể vẫn sử dụng lớp pizza không? Tất nhiên là được. Đơn giản, bạn chỉ gọi cách thức addToppings thay vì gọi makePizza.

Vì sao sử dụng OOP

Sử dụng OOP có một vài lợi ích bao gồm một file đơn giản với những hàm trên nó. Đầu tiên, với OOP bạn có thể giữ những bit thông tin có liên quan với nhau và trình bày những nhiệm vụ phức tạp với dữ liệu đó. Thứ hai, bạn có thể xử lý dữ liệu không giới hạn thời gian mà không lo lắng về biến quá dài. Thứ ba, bạn có thể có nhiều ví dụ của những lớp chạy trong cùng thời gian mà những biến này không bị sửa hoặc đè lên.

Chương 3 : SỬ DỤNG PHP VỚI MYSQL

Ở chương 2, chúng ta đã tạo được trang Web với việc sử dụng những hàm. Nhưng chúng ta chưa thực sự hiểu mối quan hệ giữa MySQL và PHP. Trong chương này chúng ta sẽ xét kỹ điều này.

Trong chương này nói đến các vấn đề:

- Hiểu biết về cơ sở dữ liệu MYSQL.
- Những dữ liệu chứa trong MySQL.
- Tác động những thông tin đặc biệt, quyền từ Web site.
- Phần mềm quản lý bảng dễ dàng.
- Có thể sửa chữa code theo ý muốn.

1.1. Tổng quan về cấu trúc và cú pháp của MySQL.

MySQL là hệ thống cơ sở dữ liệu quan hệ. Ý nghĩa cơ bản của MySQL là nó có thể lưu trữ thông tin ở những vùng khác nhau và liên kết chúng lại với nhau. Chúng ta có thể chứa bất cứ thứ gì trong một cơ sở dữ liệu. Ví dụ như những thông tin liên quan đến một người: chẳng hạn như first name, last name, address, phone....

MySQL cho phép bạn tạo những thông tin riêng lẻ trên bảng hoặc những khu vực chứa thông tin thích hợp. Trong MySQL mỗi bảng bao gồm những trường dữ liệu (field) riêng lẻ.

1.1.1. Cấu trúc MySQL

Bởi vì MySQL là hệ quản lý dữ liệu quan hệ, nó cho phép chúng ta tạo những bảng thông tin riêng, hoặc những vùng thông tin thích hợp. Trong hệ thống cơ sở dữ liệu không quan hệ, tất cả những thông tin được lưu trữ trong một bảng lớn tạo nên những khó khăn trong việc sắp xếp và chỉ có thể chép dữ liệu mà bạn muốn. Trong SQL, mỗi bảng bao gồm những phần riêng biệt, biểu diễn mỗi thông tin.

Bạn có thể tạo ra cơ sở dựa trên những loại thông tin mà bạn lưu trữ. Những bảng riêng biệt của MySQL liên kết với nhau nơi mà giá trị của vùng phổ biến là như nhau.

Ví dụ: Cho rằng bảng bao gồm tên khách hàng, địa chỉ và số ID, bảng khác bao gồm số ID, nơi ở,Vùng chung là số ID, thông tin được lưu trữ trong hai bảng riêng biệt sẽ liên kết với nhau nơi mà số ID là như nhau. Điều này cho chúng ta thông tin về khách hàng cùng lúc.

1.1.2. Các kiểu dữ liệu

Khi bạn tạo một bảng ban đầu, bạn cần nói với MySQL server kiểu nào của thông tin sẽ lưu trữ trong mỗi bảng. Các kiểu khác nhau được cho trong bảng sau:

Loại MySQL	Diễn tả	Ví dụ
Char(length)	Mọi ký tự đều có thể là loại này, nhưng có chiều dài cố định	Trạng thái của khách hàng có 2 ký tự
Varchar(length)	Mỗi ký tự đều có thể trong loại này, dữ liệu có thể thay đổi chiều dài từ 0 đến 255 ký tự.	Địa chỉ của khách hàng có chữ và số, thay đổi trong chiều dài
Int(length)	Có chiều dài từ -2147483648 đến 2147483647.	Số sản phẩm trao tay
Int(length) unsigned	Lưu trữ số từ 0 đến 4294967295.	ID khách hàng
Text	Kích thước dữ liệu là 65536 ký tự.	Cho phép đoạn text dài hơn được lưu trữ, không có loại giới hạn đến 255 ký tự.
Decimal(length, dec)	Có thể lưu trữ số thập phân	Giá cả
Enum(“option1”, “option2”,..)	Lưu trữ giá trị chắc chắn như đúng hoặc sai	Giới tính của người dùng nam hoặc nữ

Date	Lưu trữ ngày như yyyy-mm-dd	Ngày sinh nhật,...
Time	Lưu trữ giờ như hh:mm:ss	Giờ một hoạt động mới được đưa vào trang Web
Datetime	Lưu trữ ngày và giờ như yyyy-mm-dd hh:mm:ss	Ngày và giờ sau khi người đến thăm trang Web

Mặc dù những loại trên đáp ứng được nhu cầu cần thiết, một bảng các kiểu cho dưới đây cũng thường gặp.

Loại MySQL	Diễn giải
tinyint(length)	Lưu trữ số nguyên từ -128 đến 127 (thêm tham số unsigned thì cho phép lưu trữ từ 0 đến 255)
smallint(length)	Lưu trữ số nguyên từ -32768 đến 32767 (thêm tham số unsigned thì cho phép lưu trữ từ 0 đến 65535)
loại MySQL	Diễn giải
mediumint(length)	Lưu trữ số từ -8388608 đến 8388607 (thêm tham số unsigned cho phép lưu trữ từ 0 đến 16777215)
bigint(length)	Lưu trữ số từ -9223372036854775808 đến -9223372036854775807 (nếu thêm tham số unsigned cho phép lưu trữ từ 0 đến 184467440709551615)
Tinytext	Cho phép lưu trữ trên 255 ký tự
mediumtext	Cho phép lưu trữ trên 1677215 ký tự
longtext	Cho phép lưu trữ trên 4294967295 ký tự
Blob	Bằng với kiểu text, trừ trường hợp dương khi sắp xếp và so sánh, cho phép lưu trữ trên 65535 ký tự
tinyblob	Bằng với kiểu tinytext, trừ trường hợp dương khi sắp xếp và so sánh
mediumblob	Bằng với kiểu mediumtext, trừ trường hợp dương khi sắp

	xếp và so sánh
longblob	Bằng với kiểu longtext, trừ trường hợp dương khi sắp xếp và so sánh
year(length)	Lưu trữ một năm trong 4 ký tự mặc định.

1.1.3. Lựa chọn kiểu cho đúng

Trước tiên, trường sẽ chứa cả chữ và số?

Nếu trả lời có như varchar, text, char, tinytext, mediumtext, longtext, blob, tinyblob, mediumblob, longblob. Sau đó cần quan tâm có bao nhiêu ký tự được lưu trữ? Nó sẽ biến đổi thế nào?

- Có bao nhiêu ký tự được lưu trữ? Thay đổi thế nào?
- 0 đến 255 ký tự, chiều dài thay đổi: sử dụng varchar nếu bạn muốn xóa tất cả những khoảng trắng, giá trị mặc định. Sử dụng longtext nếu bạn thay đổi về chiều dài. Sử dụng blob nếu bạn không giữ về khoảng cách.
- 256->65536 ký tự: Sử dụng text nếu bạn không cần trường hợp dương trong tìm kiếm, sắp xếp, so sánh. Dùng blob nếu cần trường hợp dương.
- 65537->1677215 ký tự: Sử dụng mediumtext nếu bạn không cần trường hợp dương trong tìm kiếm, sắp xếp, so sánh. Dùng mediumblob nếu cần trường hợp dương.
- 1677216->4294967295 ký tự: Sử dụng longtext nếu bạn không cần trường hợp dương trong tìm kiếm, sắp xếp, so sánh. Dùng longblob nếu cần trường hợp dương

Nếu trả lời có thì có lẽ chứa chữ cái và số, nhưng phải một trong số có giới hạn về giá trị enum

Nếu trả lời không, nó bao gồm thời gian, ngày tháng

1.1.4. NULL/NOT NULL

My SQL server của bạn muốn biết vùng dữ liệu của bạn có rỗng hay không, bạn dùng NULL hoặc NOT NULL. Nếu vùng dữ liệu được định nghĩa là NOT

NULL thì người dùng bắt buộc phải nhập dữ liệu vào. Nếu dùng NULL thì vùng dữ liệu sẽ không chứa giá trị gì hết.

1.1.5. INDEXES

MySQL sử dụng INDEXES để giải quyết việc nghiên cứu dòng thông tin. INDEXES làm việc thế nào? Bạn tưởng tượng bạn có một cái phòng đầy ắp, toàn những thứ bạn có được mà chưa bao giờ bạn đụng đến. Bạn muốn tìm lại thông tin gì đó, bạn phải mất khá nhiều thời gian để làm việc này. Bạn cảm thấy khó chịu và bạn bắt đầu sắp xếp chúng, tổ chức lại chúng.

Bạn thử tưởng tượng lượng thông tin được lưu trữ trong bảng, vào thời gian nào đó bạn cần tìm vài thứ, bạn bắt đầu tìm tất cả những dòng, bạn sẽ làm gì nếu có 10000 dòng? Chuyện gì sẽ xảy ra?

Bằng cách sử dụng hệ thống chọn lọc bên trong, MySQL sẽ giúp bạn tìm nhanh chóng và chính xác, nó làm được điều này nhờ sử dụng INDEXES, cũng được biết như một khóa.

MySQL yêu cầu INDEX trong mỗi bảng, vì thế có vài thứ được đến. Thông thường, bạn sử dụng khóa chính, hoặc tạo ra sự duy nhất để giữ dữ liệu riêng lẻ. Trường này phải “not null” và “unique”

1.1.6. UNIQUE

Chúng ta có thể sử dụng UNIQUE để thể hiện tính độc quyền, không thể chèn thêm dữ liệu, khi thêm vào chương trình sẽ báo lỗi.

1.1.7. Tăng tự động(auto Increment)

Bạn có thể thiết kế vùng tăng tự động bằng cách dùng lệnh Auto_Increment, ta có thể đặt lệnh cho bất cứ vùng nào trong bảng.

1.1.8. Những tham số khác

Bạn có thể làm những chỉ định khác khi tạo cơ sở dữ liệu, nhưng tốt hơn cho người sử dụng. Để biết thêm về danh sách những tham số, bạn có thể tham khảo thêm :www.mysql.com

1.1.9. Các kiểu bảng của My SQL và kỹ thuật lưu trữ

Những phiên bản mới gần đây của My SQL thường sử dụng 5 loại chính

- . MyISAM
- . MERGE
- . MEMORY
- . InnoDB
- . BDB

1.1.10. Lệnh và cú pháp trong My SQL

Các lệnh và cấu trúc của MySQL thường dùng:

- . CREATE: Tạo mới một cơ sở dữ liệu hoặc một bảng
- . ALTER: Thay đổi bảng
- . SELECT: Chọn dữ liệu
- . DELETE: Xóa dữ liệu từ bảng
- . DESCRIBE: Cho biết cấu trúc và chi tiết của bảng
- . INSERT INTO *tablename* VALUES: Đặt giá trị vào bảng
- . UPDATE: Sửa đổi dữ liệu trong bảng
- . DROP: Xóa toàn bộ bảng hoặc cơ sở dữ liệu

Chương 3 : SỬ DỤNG PHP VỚI MYSQL

Ở chương 2, chúng ta đã tạo được trang Web với việc sử dụng những hàm. Nhưng chúng ta chưa thực sự hiểu mối quan hệ giữa MySQL và PHP. Trong chương này chúng ta sẽ xét kỹ điều này.

Trong chương này nói đến các vấn đề:

- Hiểu biết về cơ sở dữ liệu MYSQL.
- Những dữ liệu chứa trong MySQL.
- Tác động những thông tin đặc biệt, quyền từ Web site.
- Phần mềm quản lý bảng dễ dàng.
- Có thể sửa chữa code theo ý muốn.

Bắt đầu với My SQL và PHP như thế nào?

Một vài hàm thường dùng là:

- `mysql_connect ("hostname", "user", "pass")`: Kết nối đến MY SQL server.
- `mysql_select_db("database name")`: Tương đương với lệnh USE trong My SQL, làm cho cơ sở dữ liệu được chọn hoạt động
- `mysql_query("query")`: Gửi bất kỳ câu lệnh của My SQL đến server
- `mysql_fetch_rows("biến kết quả từ truy vấn")`: Trả về một dòng kết quả trong một dữ liệu truy vấn
- `mysql_fetch_array("biến kết quả từ truy vấn")`: Trả về nhiều dòng kết quả của truy vấn
- `mysql_error()`: Đưa ra lỗi được trả về từ My SQL server

Bạn có thể gửi bất kỳ My SQL nào đến server thông qua câu lệnh PHP hoặc câu lệnh `Mysql_query`.

Ví dụ:

```
$query = "SELECT * from TABLE";
```

```
$results = mysql_query($query);
```

Bạn cũng có thể dùng như sau:

```
$results = mysql_query("SELECT * from TABLE");
```

Kết nối MY SQL Server

Lệnh kết nối với hàm PHP là `mysql_connect`, với cú pháp như sau:

```
$host = "localhost";
```

```
$user = "bp5am";
```

```
$pass = "bp5ampass";
```

```
$connect = mysql_connect($host, $user, $pass);
```

Trong đó :

\$host là tên của máy chủ

\$user:tên của máy người dùng

\$pass:mật khẩu truy cập của người dùng

Đọc, tạo cơ sở dữ liệu

Để tạo một cơ sở dữ liệu cho bạn cần 3 bảng:

- Một bảng movie: Ta sẽ lưu trữ tên và thông tin của movie.
- Một bảng movietype: Nơi lưu trữ những catalog của movie.
- Một bảng people: Ta sẽ lưu trữ tên của các diễn viên và đạo diễn.

Ví dụ tạo cơ sở dữ liệu và bảng

Tạo trang php với tên: createmovie.php

```
<?php
```

```
//connect to MySQL; note we've used our own parameters-  
you should use
```

```
//your own for hostname, user, and password
```

```
$connect = mysql_connect("localhost", "root", "") or
```

```
die ("Hey loser, check your server connection.");
```

```
//create the main database if it doesn't already exist
```



```
$create = mysql_query("CREATE DATABASE IF NOT
EXISTS moviesite") or die(mysql_error());
//make sure our recently created database is the active one
mysql_select_db("moviesite");
//create "movie" table
$movie = "CREATE TABLE movie (
    movie_id int(11) NOT NULL auto_increment,
    movie_name varchar(255) NOT NULL,
    movie_type tinyint(2) NOT NULL default 0,
    movie_year int(4) NOT NULL default 0,
    movie_leadactor int(11) NOT NULL default 0,
    movie_director int(11) NOT NULL default 0,
    PRIMARY KEY (movie_id),
    KEY movie_type (movie_type,movie_year)";
$results = mysql_query($movie)
    or die (mysql_error());
//create "movietype" table
$movietype = "CREATE TABLE movietype (
    movietype_id int(11) NOT NULL auto_increment,
    movietype_label varchar(100) NOT NULL,
    PRIMARY KEY (movietype_id)";
$results = mysql_query($movietype)
    or die(mysql_error());
//create "people" table
$people = "CREATE TABLE people (
    people_id int(11) NOT NULL auto_increment,
    people_fullname varchar(255) NOT NULL,
    people_isactor tinyint(1) NOT NULL default 0,
    people_isdirector tinyint(1) NOT NULL default 0,
```

```

        PRIMARY KEY (people_id));
$results = mysql_query($people)
        or die(mysql_error());
echo "Movie Database successfully created!";
?>

```

Tiếp theo ta tạo trang php với tên: moviedata.php

```

<?php
//connect to MySQL
$connect = mysql_connect("localhost", "root", "")
        or die ("Hey loser, check your server connection.");
//make sure we're using the right database
mysql_select_db("moviesite");
//insert data into "movie" table
$insert = "INSERT INTO movie (movie_id, movie_name,
movie_type, "
"movie_year, movie_leadactor, movie_director) "
"VALUES (1, 'Bruce Almighty', 5, 2003, 1, 2), "
"(2, 'Office Space', 5, 1999, 5, 6), "
"(3, 'Grand Canyon', 2, 1991, 4, 3)";
$results = mysql_query($insert)
        or die(mysql_error());
//insert data into "movietype" table
$type = "INSERT INTO movietype (movietype_id,
movietype_label) "
"VALUES (1, 'Sci Fi'), "
"(2, 'Drama'), "
"(3, 'Adventure'), "
"(4, 'War'), "
"(5, 'Comedy'), "

```

```

“(6, ‘Horror’), “.
“(7, ‘Action’), “.
“(8, ‘Kids’)” ;
$results = mysql_query($type)
    or die(mysql_error());
//insert data into “people” table
$people = “INSERT INTO people (people_id, people_fullname, “.
    .“people_isactor, people_isdirector) “.
    “VALUES (1, ‘Jim Carrey’, 1, 0), “.
    “(2, ‘Tom Shadyac’, 0, 1), “.
    “(3, ‘Lawrence Kasdan’, 0, 1), “.
    “(4, ‘Kevin Kline’, 1, 0), “.
    “(5, ‘Ron Livingston’, 1, 0), “.
    “(6, ‘Mike Judge’, 0, 1)”;
$results = mysql_query($people)
    or die(mysql_error());
echo “Data inserted successfully!”;
?>

```

Chúng làm việc như thế nào?

Đầu tiên ta thực hiện kết nối đến My SQL server, sau đó tạo lập cơ sở dữ liệu nếu không tạo được sẽ đưa ra thông báo lỗi.

Sau đó bắt đầu tạo từng bảng riêng lẻ

Đưa dữ liệu vào các bảng

Truy vấn cơ sở dữ liệu

Chúng ta đã tạo ra cơ sở dữ liệu, bây giờ muốn nhận lại thông tin từ dữ liệu đó, dùng công thức như sau:

```

SELECT [fieldnames]
AS [alias]

```

FROM [*tablename*]
WHERE [*criteria*]
ORDER BY [*fieldname to sort on*] [DESC]
LIMIT [*offset, maxrows*]

- SELECT [*fieldnames*]: Quyết định đầu tiên mà vùng chứa tên đặc biệt mà bạn muốn nhận lại. Nếu bạn muốn thấy tất cả thông tin bạn chỉ cần chọn dấu *
- AS: Bạn có thể sử dụng bí danh để nhóm hai hoặc nhiều hơn hai vùng khác nhau mà có thể chuyển đến chúng * như là một biến lớn

SELECT first_name, last_name AS full_name. . . ORDER BY full_name . . .

Bạn không thể sử dụng tham số AS với tham số WHERE, bởi vì nó bị giới hạn trong MY SQL. Khi WHERE được thực thi thì cột giá trị không biết

- FROM: Tên bảng cần truy xuất thông tin
- WHERE: Danh sách điều kiện chọn lọc dữ liệu
- ORDER BY: Để sắp xếp dữ liệu trong vùng
- LIMIT: Giới hạn kết quả trả về

Bắt đầu với My SQL và PHP như thế nào?

Một vài hàm thường dùng là:

- mysql_connect ("*hostname*", "*user*", "*pass*"): Kết nối đến MY SQL server.
- mysql_select_db("*database name*"): Tương đương với lệnh USE trong My SQL, làm cho cơ sở dữ liệu được chọn hoạt động
- mysql_query("*query*"): Gửi bất kỳ câu lệnh của My SQL đến server
- mysql_fetch_rows("*biến kết quả từ truy vấn*"): Trả về một dòng kết quả trong một dữ liệu truy vấn
- mysql_fetch_array("*biến kết quả từ truy vấn*"): Trả về nhiều dòng kết quả của truy vấn
- mysql_error(): Đưa ra lỗi được trả về từ My SQL server

Bạn có thể gửi bất kỳ My SQL nào đến server thông qua câu lệnh PHP hoặc câu lệnh Mysql_query.

Ví dụ:

```
$query = "SELECT * from TABLE";
```

```
$results = mysql_query($query);
```

Bạn cũng có thể dùng như sau:

```
$results = mysql_query("SELECT * from TABLE");
```

Kết nối MY SQL Server

Lệnh kết nối với hàm PHP là mysql_connect, với cú pháp như sau:

```
$host = "localhost";
```

```
$user = "bp5am";
```

```
$pass = "bp5ampass";
```

```
$connect = mysql_connect($host, $user, $pass);
```

Trong đó :

\$host là tên của máy chủ

\$user:tên của máy người dùng

\$pass:mật khẩu truy cập của người dùng

Chương 3 : SỬ DỤNG PHP VỚI MYSQL

Ở chương 2, chúng ta đã tạo được trang Web với việc sử dụng những hàm. Nhưng chúng ta chưa thực sự hiểu mối quan hệ giữa MySQL và PHP. Trong chương này chúng ta sẽ xét kỹ điều này.

Trong chương này nói đến các vấn đề:

- Hiểu biết về cơ sở dữ liệu MYSQL.
- Những dữ liệu chứa trong MySQL.
- Tác động những thông tin đặc biệt, quyền từ Web site.
- Phần mềm quản lý bảng dễ dàng.
- Có thể sửa chữa code theo ý muốn.

Đọc, tạo cơ sở dữ liệu

Để tạo một cơ sở dữ liệu cho bạn cần 3 bảng:

- Một bảng movie: Ta sẽ lưu trữ tên và thông tin của movie.
- Một bảng movietype: Nơi lưu trữ những catalog của movie.
- Một bảng people: Ta sẽ lưu trữ tên của các diễn viên và đạo diễn.

Ví dụ tạo cơ sở dữ liệu và bảng

Tạo trang php với tên: createmovie.php

```
<?php
//connect to MySQL; note we've used our own parameters-
you should use
//your own for hostname, user, and password
$connect = mysql_connect("localhost", "root", "") or
die ("Hey loser, check your server connection.");
//create the main database if it doesn't already exist
$create = mysql_query("CREATE DATABASE IF NOT
EXISTS moviesite") or die(mysql_error());
```

```
//make sure our recently created database is the active one
mysql_select_db("moviesite");
//create "movie" table
$movie = "CREATE TABLE movie (
    movie_id int(11) NOT NULL auto_increment,
    movie_name varchar(255) NOT NULL,
    movie_type tinyint(2) NOT NULL default 0,
    movie_year int(4) NOT NULL default 0,
    movie_leadactor int(11) NOT NULL default 0,
    movie_director int(11) NOT NULL default 0,
    PRIMARY KEY (movie_id),
    KEY movie_type (movie_type,movie_year)";
$results = mysql_query($movie)
    or die (mysql_error());
//create "movietype" table
$movietype = "CREATE TABLE movietype (
    movietype_id int(11) NOT NULL auto_increment,
    movietype_label varchar(100) NOT NULL,
    PRIMARY KEY (movietype_id)";
$results = mysql_query($movietype)
    or die(mysql_error());
//create "people" table
$people = "CREATE TABLE people (
    people_id int(11) NOT NULL auto_increment,
    people_fullname varchar(255) NOT NULL,
    people_isactor tinyint(1) NOT NULL default 0,
    people_isdirector tinyint(1) NOT NULL default 0,
    PRIMARY KEY (people_id)";
$results = mysql_query($people)
```

```
        or die(mysql_error());
        echo "Movie Database successfully created!";
    ?>
```

Tiếp theo ta tạo trang php với tên: moviedata.php

```
<?php
//connect to MySQL
$connect = mysql_connect("localhost", "root", "")
        or die ("Hey loser, check your server connection.");
//make sure we're using the right database
mysql_select_db("moviesite");
//insert data into "movie" table
$insert = "INSERT INTO movie (movie_id, movie_name,
movie_type, "
"movie_year, movie_leadactor, movie_director) "
"VALUES (1, 'Bruce Almighty', 5, 2003, 1, 2), "
"(2, 'Office Space', 5, 1999, 5, 6), "
"(3, 'Grand Canyon', 2, 1991, 4, 3)";
$results = mysql_query($insert)
        or die(mysql_error());
//insert data into "movietype" table
$type = "INSERT INTO movietype (movietype_id,
movietype_label) "
"VALUES (1,'Sci Fi'), "
"(2, 'Drama'), "
"(3, 'Adventure'), "
"(4, 'War'), "
"(5, 'Comedy'), "
"(6, 'Horror'), "
"(7, 'Action'), "
```



```

        "(8, 'Kids')";
$results = mysql_query($type)
        or die(mysql_error());
//insert data into "people" table
$people = "INSERT INTO people (people_id, people_fullname, "
        ."people_isactor, people_isdirector) "
        ."VALUES (1, 'Jim Carrey', 1, 0), "
        ."(2, 'Tom Shadyac', 0, 1), "
        ."(3, 'Lawrence Kasdan', 0, 1), "
        ."(4, 'Kevin Kline', 1, 0), "
        ."(5, 'Ron Livingston', 1, 0), "
        ."(6, 'Mike Judge', 0, 1)";
$results = mysql_query($people)
        or die(mysql_error());
echo "Data inserted successfully!";
?>

```

Chúng làm việc như thế nào?

Đầu tiên ta thực hiện kết nối đến My SQL server, sau đó tạo lập cơ sở dữ liệu nếu không tạo được sẽ đưa ra thông báo lỗi.

Sau đó bắt đầu tạo từng bảng riêng lẻ

Đưa dữ liệu vào các bảng

Truy vấn cơ sở dữ liệu

Chúng ta đã tạo ra cơ sở dữ liệu, bây giờ muốn nhận lại thông tin từ dữ liệu đó, dùng công thức như sau:

```

SELECT [fieldnames]
AS [alias]
FROM [tablename]
WHERE [criteria]

```

ORDER BY [*fieldname to sort on*] [DESC]

LIMIT [*offset, maxrows*]

- SELECT [*fieldnames*]: Quyết định đầu tiên mà vùng chứa tên đặc biệt mà bạn muốn nhận lại. Nếu bạn muốn thấy tất cả thông tin bạn chỉ cần chọn dấu *
- AS: Bạn có thể sử dụng bí danh để nhóm hai hoặc nhiều hơn hai vùng khác nhau mà có thể chuyển đến chúng * như là một biến lớn

SELECT first_name, last_name AS full_name. . . ORDER BY full_name . . .

Bạn không thể sử dụng tham số AS với tham số WHERE, bởi vì nó bị giới hạn trong MYSQL. Khi WHERE được thực thi thì cột giá trị không biết

- FROM: Tên bảng cần truy xuất thông tin
- WHERE: Danh sách điều kiện chọn lọc dữ liệu
- ORDER BY: Để sắp xếp dữ liệu trong vùng
- LIMIT: Giới hạn kết quả trả về

Chương 3 : SỬ DỤNG PHP VỚI MYSQL

Ở chương 2, chúng ta đã tạo được trang Web với việc sử dụng những hàm. Nhưng chúng ta chưa thực sự hiểu mối quan hệ giữa MySQL và PHP. Trong chương này chúng ta sẽ xét kỹ điều này.

Trong chương này nói đến các vấn đề:

- Hiểu biết về cơ sở dữ liệu MYSQL.
- Những dữ liệu chứa trong MySQL.
- Tác động những thông tin đặc biệt, quyền từ Web site.
- Phần mềm quản lý bảng dễ dàng.
- Có thể sửa chữa code theo ý muốn.

WHERE , oh WHERE

Đưa ra điều kiện để hiển thị thông tin mà bạn cần

```
SELECT * FROM customers WHERE gender = "Male"
```

Trả về tất cả các thông tin khách hàng có giới tính "Nam"

Các phép toán so sánh trong câu lệnh WHERE:

- =,<,>,>=,<=,!=
- LIKE và %: So sánh một phần của đoạn code với đoạn chứa nó, xuất ra tất cả những mẫu tin có liên quan đến nó.

Ví dụ:

```
SELECT * FROM products WHERE description LIKE "%shirt%"
```

Ở đây sẽ đưa ra tất cả những phần có chứa 'shirt'.

Ví dụ về sử dụng truy vấn SELECT: Mở trình soạn thảo nhập đoạn mã và lưu lại với tên select.php.

```
<?php
```

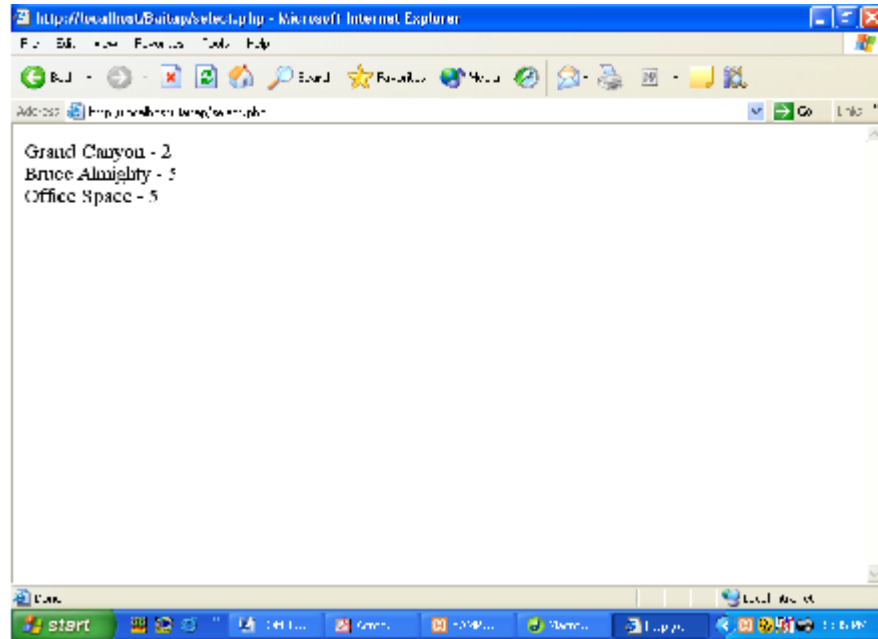
```
//connect to MySQL
```

```

$connect = mysql_connect("localhost", "bp5am", "bp5ampass")
    or die("Hey loser, check your server connection.");
//make sure we're using the right database
mysql_select_db("moviesite");
$query = "SELECT movie_name, movie_type " .
    "FROM movie " .
    "WHERE movie_year>1990 " .
    "ORDER BY movie_type";
$results = mysql_query($query)
or die(mysql_error());
while ($row = mysql_fetch_array($results))
{
    extract($row);
    echo $movie_name;
    echo " - ";
    echo $movie_type;
    echo "<br>";
}
?>

```

Sau khi chạy chương trình ta sẽ thấy giao diện màn hình như **Hình 3.5**.
 Error! No text of specified style in document.1



Hình 3.5. Error! No text of specified style in document..2

Chúng làm việc như thế nào?

Khi chúng ta muốn lấy thông tin tên, ta có:

```
SELECT *FROM
```

Nhưng thay vì bạn viết:

```
$query="SELECT movie_name, movie_type"
```

Kế đến, chúng ta muốn lấy thông tin từ bảng nào

```
"FROM movie"
```

Sau đó, ta đưa điều kiện muốn thấy thông tin từ năm 1990 trở lên:

```
WHERE movie_year>1990"
```

Bạn cần kết quả xuất ra được sắp xếp theo thứ tự bạn muốn:

```
ORDER BY movie_type;
```

Kế đó, chúng ta tập hợp tất cả những dòng theo yêu cầu đã đưa ra của bạn:

```
$results mysql_query($query)
```

```
Ordie(mysql_error());
```

Sau đó, chúng ta kết hợp đưa ra kết quả với những dòng:

```
while ($row = mysql_fetch_array($results))
```

```

    {
    extract($row);
    echo $movie_name;
    echo " - ";
    echo $movie_type;
    echo "<br>";
    }

```

Ghi chú với mỗi dòng tìm thấy, bạn lưu trữ bảng này trong một mảng \$row, sử dụng hàm `mysql_fetch_array()`, sau đó giải các biến trong \$row, sử dụng hàm *extract*, xuất ra những gì bạn cần.

Làm việc với PHP và mảng dữ liệu: foreach

Hàm *foreach* tương tự như hàm While, nếu sử dụng while kết hợp thông qua một danh sách kết quả từ truy vấn

```

foreach ($row as $value)
{
    echo $value;
    echo "<br>";
}

```

Ví dụ về việc sử dụng *foreach*. Mở tập `select.php` sửa đổi lại như sau.

```

<?php
//connect to MySQL
$connect = mysql_connect("localhost", "root", "")
    or die("Hey loser, check your server connection.");
//make sure we're using the right database
mysql_select_db("moviesite");
$query = "SELECT movie_name, movie_type " . "FROM movie ";
$results = mysql_query($query)
    or die(mysql_error());
//while ($row = mysql_fetch_array($results,MYSQL_ASSOC)){

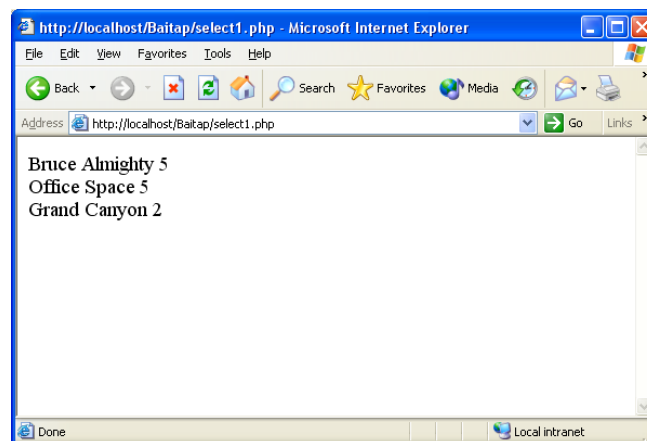
```

```

while ($row = mysql_fetch_assoc($results))
{
    foreach ($row as $val1)
    {
        echo $val1;
        echo " ";
    }
    echo "<br>";
    //extract($row);
    //echo $movietype_id;
    //echo " - ";
    //echo $movietype_label;
}
?>

```

Sau khi cho chạy chương trình, giao diện như **Hình 3.5.2.1**



Hình 3.5.2.1

Nó làm việc thế nào:

Ghi chú: Ở đây có sử dụng

```

while ($row = mysql_fetch_assoc($results))
{
    foreach ($row as $val1)
    {
        echo $val1;
        echo " ";
    }
}

```

So với phần trước, dùng hàm

```

while ($row = mysql_fetch_array($results))
{
    extract($row);
    echo $movie_name;
    echo " - ";
    echo $movie_type;
    echo "<br>";
}

```

Dùng hàm *foreach* ta thấy ngắn gọn hơn , nó xuất tất cả thông tin liên quan mà không cần dùng lệnh *echo* để chỉ định.

Một điều cần chú ý nữa là nếu ta sử dụng lệnh: `while ($row = mysql_fetch_array($results))` thay cho câu lệnh `:while ($row = mysql_fetch_assoc($results))` thì kết quả sẽ xuất ra 2 lần.

Để khắc phục tình trạng này ta dùng: `mysql_fetch_array($results,MYSQL_ASSOC)` hoặc `mysql_fetch_assoc`

- Sử dụng *foreach* để tạo một mảng

Ví dụ. `select2.php`

```

<?php
//connect to MySQL
$connect = mysql_connect("localhost", "root", "")

```



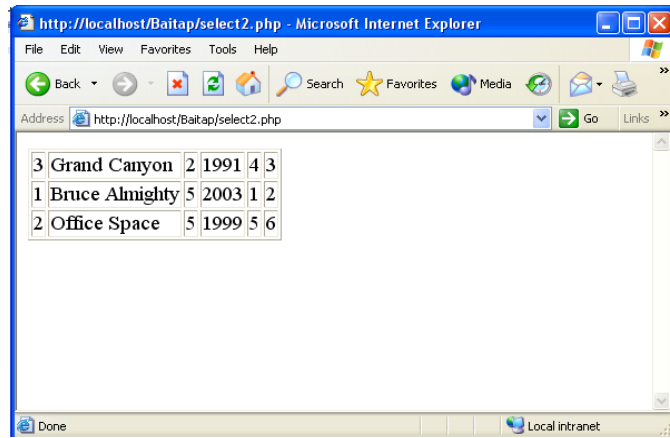
```

        or die("Hey loser, check your server connection.");
//make sure we're using the right database
        mysql_select_db("moviesite");
$query = "SELECT * " .
"FROM movie " .
"WHERE movie_year>1990 " .
        "ORDER BY movie_type";
$results = mysql_query($query)
or die(mysql_error());
echo "<table border=\"1\">\n";
while ($row = mysql_fetch_assoc($results))
{
        echo "<tr>\n";
        foreach($row as $value)
        {
                echo "<td>\n";
                echo $value;
                echo "</td>\n";
        }
        echo "</tr>\n";
}
echo "</table>\n";
?>

```

Kết quả sau khi chạy như **Hình 3.5.2.2**

:



Hình 3.5.2.2

Ở đây làm việc thế nào?

Bạn sử dụng mệnh `mysql_query` và `while` để phục hồi lại lĩnh vực của bạn. Sau đó, với mỗi giá trị bạn nhận lại, bạn đặt nó ở mỗi ô phân cách và kết chúng lại trong `html` và `echo`

Chương 3 : SỬ DỤNG PHP VỚI MYSQL

Ở chương 2, chúng ta đã tạo được trang Web với việc sử dụng những hàm. Nhưng chúng ta chưa thực sự hiểu mối quan hệ giữa MySQL và PHP. Trong chương này chúng ta sẽ xét kỹ điều này.

Trong chương này nói đến các vấn đề:

- Hiểu biết về cơ sở dữ liệu MYSQL.
- Những dữ liệu chứa trong MySQL.
- Tác động những thông tin đặc biệt, quyền từ Web site.
- Phần mềm quản lý bảng dễ dàng.
- Có thể sửa chữa code theo ý muốn.

Liên hệ giữa 2 bảng

Bạn có thể lấy thông tin từ nhiều bảng theo hai cách:

- Chuyển đến những bảng riêng biệt trong truy vấn của bảng và liên kết chúng trong một vùng chung tạm thời.
- Sử dụng công thức JOIN trong truy vấn của bạn đối với những bảng riêng biệt.

Dùng câu lệnh SELECT để phân biệt giữa hai bảng trong dữ liệu của bạn, kết nối chúng trong một bảng mới.

```
$query = "SELECT customers.name, orders.order_total  
FROM customers, orders  
WHERE customers.cust_ID = orders.cust_ID";
```

//trả về tên khách hàng từ bảng khách hàng và order_total từ bảng order nơi mà cust_ID trong bảng customers bằng với cust_ID trong bảng order.

Ví dụ : select21.php

```
<?php
```

```

//connect to MySQL
$connect = mysql_connect("localhost", "root", "")
    or die ("Hey loser, check your server connection.");
//make sure we're using the right database
mysql_select_db("moviesite");
$query      =      "SELECT      movie.movie_name,
movietype.movietype_label " .
"FROM movie, movietype " .
"WHERE movie.movie_type = movietype.movietype_id " .
"AND movie.movie_year>1990 " .
"ORDER BY movie_type";
$results = mysql_query($query)
    or die(mysql_error());
echo "<table border=\\"1\">\n";
while ($row = mysql_fetch_assoc($results))
{
    echo "<tr>\n";
    foreach($row as $value)
    {
        echo "<td>\n";
        echo $value;
        echo "</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";
?>

```

Kết quả như **Hình 3.5.3**



Hình 3.53

Ghi chú:

Ở đây có hai vùng được liên kết với nhau thông qua WHERE, những số id từ hai bảng khác nhau (movie_type trong bảng movie và movietype_id trong bảng movietype) được liên kết với nhau .

Kết nối hai bảng (Join two tables)

Cũng giống như phần kết hợp hai bảng ở trên, nhưng ở đây câu lệnh SELECT là:

```
SELECT movie_name, movietype_label.
```

Bạn muốn cập nhật dữ liệu loại nào dùng câu lệnh:

```
FROM movie
```

```
LEFT JOIN movietype
```

Dùng câu lệnh trên có nghĩa đơn giản là movietype phụ thuộc vào movie.

Sau đó bạn gửi đến server vùng nào cần kết nối như sau:

```
ON movie_type = movietype_id.
```

Ví dụ : select22.php

```
<?php
```

```
//connect to MySQL
```

```
$connect = mysql_connect("localhost", "root", "")
```

```
or die ("Hey loser, check your server connection.");
```

```
//make sure we're using the right database
```

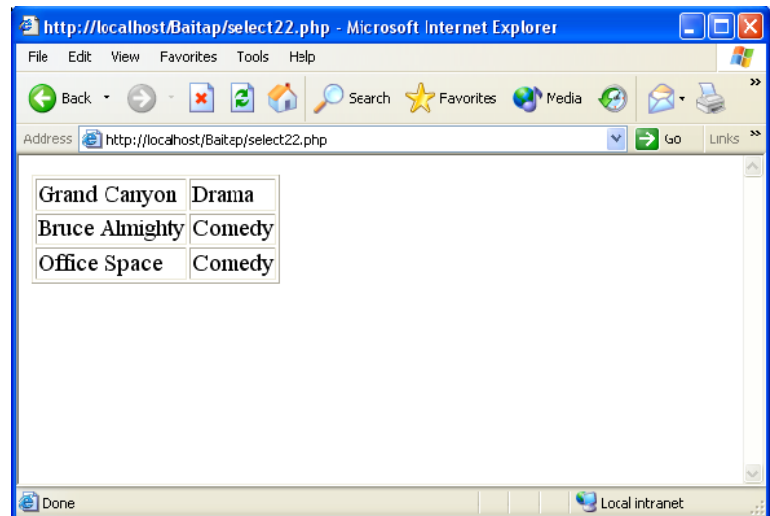
```
mysql_select_db("moviesite");
```

```

$query = "SELECT movie_name, movietype_label ".
        "FROM movie ".
        "LEFT JOIN movietype ".
        "ON movie_type = movietype_id ".
        "WHERE movie.movie_year>1990 ".
        "ORDER BY movie_type";
$results = mysql_query($query)
        or die(mysql_error());
echo "<table border='1'\>\n";
while ($row = mysql_fetch_assoc($results))
{
    echo "<tr>\n";
    foreach($row as $value)
    {
        echo "<td>\n";
        echo $value;
        echo "</td>\n";
    }
    echo "</tr>\n";
}
echo "</table>\n";
?>

```

Kết quả như **Hình 3.5.4** :



Hình 3.5.4

Bạn thấy kết quả giống như phần trên. Bạn làm điều này trong dòng đầu tiên của mệnh đề SELECT:

```
SELECT movie_name, movietpe_label
```

Sau đó bạn gọi đến Mysql những bảng nào bạn muốn cập nhật và loại nào bạn muốn kết hợp với nhau trong mệnh đề sau:

```
FROM movie
```

```
LEFT JOIN movietype
```

Bạn dùng LEFT kết hợp những mệnh đề trong trường hợp này. Bạn lấy thông tin chính từ movie và tham chiếu thông tin từ movietype

Help Tips và Suggestions

Việc cung cấp tài liệu

Việc sử dụng PHPMysqlAdmin

Nó dễ dàng cho làm những công việc sau:

- Cắt và tạo CSDL
- Tạo , xóa bảng.
- Tạo, xóa vùng.
- Nhập bất kỳ câu lệnh MySQL nào.
- Xem và in cấu trúc bảng.

- Mã PHP tổng quát.
- Xem dữ liệu trong bảng.

Chương 4 : SỬ DỤNG BẢNG ĐỂ TRÌNH BÀY DỮ LIỆU

Chúng ta có thể thành công trong việc kết hợp PHP và MySQL để tạo ra trang web động, khi có nhiều dòng dữ liệu cần trình bày, chúng ta cần phải có cơ chế để người xem dễ dàng đọc dữ liệu và cần phải đẹp, ngắn gọn, trình bày rõ ràng. Phương pháp dễ nhất là sử dụng bảng.

Trong chương này chúng ta sẽ biết:

- Tạo bảng để tổ chức dữ liệu từ cơ sở dữ liệu.
- Tạo tiêu đề cột tự động.
- Bảng thông dụng với kết quả truy vấn MySQL cơ bản.
- Bảng thông dụng với kết quả truy vấn MySQL phức tạp.
- Tạo trang thân thiện cho người dùng

Tạo bảng

Khi có một danh sách dữ liệu, cần phải đặt cấu trúc, tiêu đề cột, định dạng bảng.

Ví dụ: định nghĩa tiêu đề bảng: Mở trình soạn thảo nhập đoạn mã và lưu với tên table.php.

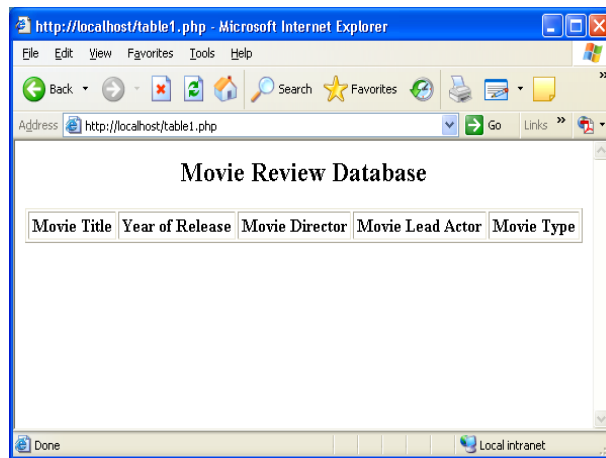
```
<?php
    $movie=<<<EOD
    <h2><center>Movie Review Database</center></h2>
    <table width="100%" border="1" cellpadding="2"
    cellspacing="2" align="center">
        <tr>
            <th>Movie Title</th>
            <th>Year of Release</th>
            <th>Movie Director</th>
            <th>Movie Lead Actor</th>
```

```

        <th>Movie Type</th>
    </tr>
</table>
EOD;
echo $movie;
?>

```

Kết quả như **Hình 4.1.1**



Hình Error! No text of specified style in document.-1.1

Tất cả đoạn mã nằm bên trong `<<<EOD` và `EOD` thì chứa trong biến `$table` vì thế, thay vì in mỗi thành phần của bảng HTML thì nên thêm biến `$table`. Nhân tiện nó thêm vào khung cho bảng để tiện việc trình bày.

Sau đó xuất nội dung đơn giản của biến `$table`. Cuối cùng chúng ta đóng PHP bằng thẻ đóng `?>`.

Bằng việc sử dụng 2 thẻ ta có thể sử dụng mã HTML đơn giản.

Như đã đề cập ở chương 2 về việc sử dụng heredoc ta có thể thay thế thẻ `<<<EOD` bằng bất cứ chuỗi nào mà ta muốn nhưng bắt đầu và kết thúc phải phù hợp.

Chú ý:

Ở đây không có khoảng cách sau `=<<<EOD` và thẻ `EOD`, không có khoảng cách nào giữa các dòng, thực lẽ hoặc bất kỳ ký tự nào trong dòng thẻ đóng heredoc. Nếu có bất kỳ khoảng trống nào thì nó sẽ báo lỗi. Do đó luôn luôn nhớ rằng phải xóa tất cả các khoảng cách sau các thẻ đó.

Bảng thường dùng

Ví dụ: Điền dữ liệu vào bảng:

Bởi vì đây là một đoạn mã lớn nên tất những thay đổi đều có được thể hiện rõ rệt. Một vài thứ được lấy ra từ tập lệnh gốc.

Mở file table1.php và sửa lại mã. Chúng ta sử dụng lại dữ liệu ở chương 3 cho ví dụ này. Phải nhớ tên server, tên người dùng, mật khẩu, tên cơ sở dữ liệu với những giá trị của bạn:

```
<?php
$link = mysql_connect("localhost","root","")
or die(mysql_error());
mysql_select_db("moviesite")
or die (mysql_error());
```

Bắt đầu kết nối cơ sở dữ liệu:

```
$query = "SELECT movie_name, movie_director, movie_leadactor " .
"FROM movie";
```

```
$result = mysql_query($query, $link)
or die(mysql_error());
```

```
$num_movies = mysql_num_rows($result);
```

Chạy một truy vấn SQL dựa vào cơ sở dữ liệu và lấy kết quả. Và trong lúc đó xem có bao nhiêu mẫu tin được trả về từ truy vấn.

Như đã thảo luận ở chương 3 ta đặt SQL là chữ hoa. Điều này dễ thực hiện bởi vì nó cho phép bạn dễ dàng nhận biết tên cột và từ khóa SQL. Nó cũng thực hiện tốt việc tạo truy vấn SQL dễ đọc hơn. Nó cũng giải thích tại sao chúng ta được viết truy vấn SQL trên dòng server.

```
$movie_header =<<<<EOD
<h2><center>Movie Review Database</center></h2>
<table width="70%" border="1" cellpadding="2">
```

```

cellspacing="2" align="center">
<tr>
<th>Movie Title</th>
<th>Movie Director</th>
<th>Movie Lead Actor</th>
</tr>

```

EOD;

Sau đó nhập đoạn mã vào góc(trừ mệnh đề echo)

Chú ý nó thật sự đã gọi là \$movie_header không phải là \$movie

```

$movie_details = '';
while ($row = mysql_fetch_array($result))
{
    $movie_name = $row['movie_name'];
    $movie_director = $row['movie_director'];
    $movie_leadactor = $row['movie_leadactor'];
    $movie_details .=<<<EOD
<tr>
<td>$movie_name</td>
<td>$movie_director</td>
<td>$movie_leadactor</td>
</tr>
    EOD;
}
$movie_details .=<<<EOD
<tr>
<td>&nbsp;</td>
</tr>
<tr>
<td>Total :$num_movies Movies</td>

```

</tr>

EOD;

Hoạt động:

Trong đoạn mã trước nó làm việc khá nhiều, vì vậy chúng ta xem xét chi tiết hơn.

Bạn biết rằng, dòng lặp của mệnh đề while thông qua những mẫu tin đã được trả về, đối với mỗi mẫu tin nó thực hiện một khối đoạn mã trong dấu ngoặc. Đừng lo lắng PHP đủ thông minh để biết có bao nhiêu mẫu tin và số mẫu tin hiện hành trong trường hợp này. Ở đây không có sự nguy hiểm cho việc ấn định giá trị sai cho mẫu tin.

Dòng đầu tiên của vòng lặp while, nói rằng các tập lệnh được viết ra là giá trị của cột movie_name trong mẫu tin hiện hành và đặt vào một biến là \$movie_name. Bốn dòng tiếp theo làm những việc tương tự, chúng đơn giản ấn định giá trị tên của cột khác cho những tên biến khác. Sau đó, bạn đến thẻ tương tự như cái bạn đã thấy khi bắt đầu chương này. Nó không hoàn toàn giống như trước bởi vì nó có .=<<<EOD thay vì =<<<EOD. Vì vậy, thay vì chỉ có một giá trị mẫu tin, \$movie_details chứa tất cả các giá trị mẫu tin được trả về, sau đó kết thúc là tổng số của những hình ảnh trong dữ liệu của bạn.

Bằng việc thêm vào dấu chấm (.) trước dấu =<<<EOD là bạn đang thêm giá trị 'existing' với giá trị 'current' của biến \$movie_details. Nếu bạn quên thêm dấu chấm (.), sau đó bạn thay thế giá trị 'existing' với giá trị 'current'. Bởi vì trong PHP \$var="1" có nghĩa là gán cho \$var có giá trị là 1 và \$var.="1" có nghĩa là "lấy giá trị hiện hành của \$var và thêm 1 cho nó". Trong ví dụ trước chú ý rằng bạn đã ấn định tên của phim cho \$movie_name và sau đó sử dụng \$movie_name thay vì làm như sau:

```
while ($row = mysql_fetch_row($result))
{
    $movie_details .=<<<EOD
<tr>
<td>$row['movie_name']</td>
</tr>
EOD;
```

}

Trong đoạn trước, mọi thứ chính xác như nhau nhưng nó sẽ giới hạn nếu bạn muốn định dạng bất kỳ giá trị biến nào.

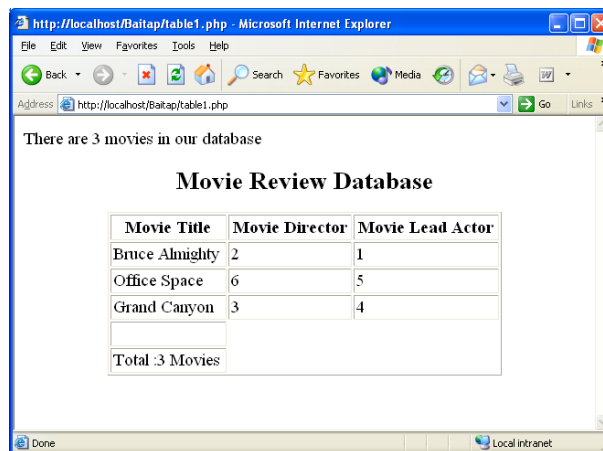
Thực hành, gồm nó tất cả lại với nhau.

Dữ liệu được nhận lại bây giờ nhưng bạn cần gửi đến tất cả các trình duyệt vì vậy nó sẽ được trình bày trong bảng.

Bạn ấn định \$movie_footer bằng cách nhập dòng sau:

```
$movie_footer = "</table>";  
$movie = <<<MOVIE  
$movie_header  
$movie_details  
$movie_footer  
MOVIE;  
echo "There are $num_movies movies in our database";  
echo $movie;  
?>
```

Trong đoạn mã trước bạn lưu file này với tên table1.php, sau đó load lên server Sau đó bạn sẽ thấy như **Hình 4.2.1**.



Hình 4.2.1.

Cách thức hoạt động:

Đầu tiên đoạn mã của bạn lấy thông tin lưu trữ trong \$movie_header, \$movie_footer, và cuộn tất cả lên và đặt nó trong \$movie với việc sử dụng heredoc. Sau đó có những dòng sau:

```
echo "There are $num_movies movies in our database";  
echo $movie;
```

Bạn in ra câu lệnh có bao nhiêu phim trong cơ sở dữ liệu và sau đó gửi tất cả các biến \$movie_header, \$movie_details, and \$movie_footer với dòng tiếp theo

Bảng này nhìn trông cũng đẹp, nhưng trong chương 3 nó không làm cho người dùng tốt nếu họ không có người giải mã bí mật để hướng dẫn cho họ biết mối liên hệ giữa diễn viên và đạo diễn trong phim. Bạn cần liên kết những bảng của bạn để điền đầy đủ thông tin.

Ví dụ: Phát triển bảng

Trong bài tập này bạn liên kết các bảng lại với nhau như bạn thấy trong chương 3, bạn có thể xuất đầy đủ thông tin dữ liệu.

Sửa file table1.php như sau lưu lại với tên table2.php

```
<?php  
$link = mysql_connect("localhost","root","")  
    or die(mysql_error());  
mysql_select_db("moviesite")  
    or die (mysql_error());  
$query = "SELECT movie_name, movie_director, movie_leadactor "  
"FROM movie";  
$result = mysql_query($query, $link)  
    or die(mysql_error());  
$num_movies = mysql_num_rows($result);  
$movie_header=<<<EOD  
<h2><center>Movie Review Database</center></h2>  
<table width="100%" border="1" cellpadding="2"  
cellspacing="2" align="center">
```

```
<tr>
<th>Movie Title</th>
<th>Movie Director</th>
<th>Movie Lead Actor</th>
</tr>
EOD;
function get_director()
{
    global $movie_director;
    global $director;
    $query_d = "SELECT people_fullname " .
    "FROM people " .
    "WHERE people_id='$movie_director'";
    $results_d = mysql_query($query_d)
    or die(mysql_error());
    $row_d = mysql_fetch_array($results_d);
    extract($row_d);
    $director = $people_fullname;
}
function get_leadactor()
{
    global $movie_leadactor;
    global $leadactor;
    $query_a = "SELECT people_fullname " .
    "FROM people " .
    "WHERE people_id='$movie_leadactor'";
    $results_a = mysql_query($query_a)
    or die(mysql_error());
    $row_a = mysql_fetch_array($results_a);
```



```

        extract($row_a);
        $leadactor = $people_fullname;
    }
while ($row = mysql_fetch_array($result))
{
    $movie_name = $row['movie_name'];
    $movie_director = $row['movie_director'];
    $movie_leadactor = $row['movie_leadactor'];
    //get director's name from people table
    get_director();
    //get lead actor's name from people table
    get_leadactor();
    $movie_details .=<<<EOD
    <tr>
    <td>$movie_name</td>
    <td>$director</td>
    <td>$leadactor</td>
    </tr>
    EOD;
}

$movie_details .=<<<EOD
<tr>
<td>Total :$num_movies Movies</td>
</tr>
EOD;
$movie_footer = "</table>";
$movie = <<<MOVIE
$movie_header
$movie_details

```

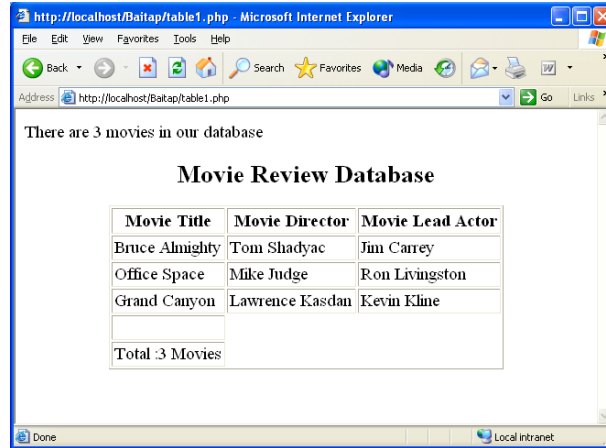
```

$movie_footer
MOVIE;
echo "There are $num_movies movies in our database";
echo $movie;

?>

```

Kết quả như **Hình 4.2.2.**



Hình 4.2.2.

Cách thức hoạt động:

Với việc thêm vào hàm `get_director` và `get_leadactor` tập lệnh yêu cầu những thông tin đặt biệt từ server cho mỗi dòng trên bảng. Điều này có thể cho bạn điền đầy đủ thông tin từ một đồng hồ độn trong truy vấn gốc, bạn có thể làm sạch những định dạng trong 2 dòng trước với sự thay đổi trong đoạn mã gần kết kết thúc của tập lệnh.

Cấu hình: Bạn đã thành công trong việc phát triển sức mạnh về tập lệnh, nó sẽ là cơ sở dữ liệu truy vấn và đặt nội dung vào bảng HTML. Cho chính bạn thấy ánh sáng trong bóng tối, nhưng giống tất cả những trình duyệt tốt, chúng ta phải đi về phía trước.

Chương 4 : SỬ DỤNG BẢNG ĐỂ TRÌNH BÀY DỮ LIỆU

Chúng ta có thể thành công trong việc kết hợp PHP và MySQL để tạo ra trang web động, khi có nhiều dòng dữ liệu cần trình bày, chúng ta cần phải có cơ chế để người xem dễ dàng đọc dữ liệu và cần phải đẹp, ngắn gọn, trình bày rõ ràng. Phương pháp dễ nhất là sử dụng bảng.

Trong chương này chúng ta sẽ biết:

- Tạo bảng để tổ chức dữ liệu từ cơ sở dữ liệu.
- Tạo tiêu đề cột tự động.
- Bảng thông dụng với kết quả truy vấn MySQL cơ bản.
- Bảng thông dụng với kết quả truy vấn MySQL phức tạp.
- Tạo trang thân thiện cho người dùng

Master là ai?

Bây giờ chúng ta hãy xây dựng một công việc tốt mà bạn đã làm nhiều và thêm nhiều thông tin và hàm hơn vào bảng của bạn. Phương tiện quan hệ giữa cha và con trong trang web của bạn cho phép người dùng có thể click vào tiêu đề của bộ phim để biết thông tin về bộ phim. Dĩ nhiên là ở đây tất cả sẽ trở thành trang web động chung, vì thế chúng ta tìm như thế nào để làm và quan hệ cha con có nghĩa.

Ví dụ: Thêm liên kết trong table

Mở trang table2.php trong bài và thêm vào những dòng mã mà xuất hiện nổi bật.

```
$query = "SELECT movie_id, movie_name, ".  
"movie_director, movie_leadactor "  
"FROM movie";  
$result = mysql_query($query, $link)  
or die(mysql_error());  
$num_movies = mysql_num_rows($result);
```

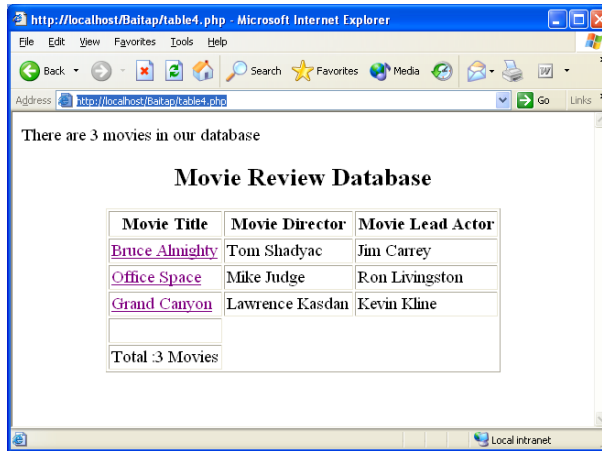
```

$movie_details = '';
while ($row = mysql_fetch_array($result))
{
    $movie_id = $row['movie_id'];
    $movie_name = $row['movie_name'];
    $movie_director = $row['movie_director'];
    $movie_leadactor = $row['movie_leadactor'];
    //get director's name from people table
    get_director();
    //get lead actor's name from people table
    get_leadactor();
    $movie_details .=<<<EOD
    <tr>
    <td><a href="movie_details.php?movie_id=$movie_id"
    title="Find out more about $movie_name">$movie_name</td>
    <td>$director</td>
    <td>$leadactor</td>
    </tr>
    EOD;
}

```

Lưu file với tên table3.php, sau đó load lên server, mở lại trình duyệt.

Bạn sẽ thấy như **Hình 4.3.1**



Hình 4.3.1

Cách thức hoạt động:

Bạn sẽ chú ý một sự thay đổi giữa hình table2.php và table3.php. Bạn có nhiều liên kết cho nhiều thông tin đến mỗi movie cho người viếng thăm trang web của bạn.

Thay đổi đầu tiên so với phần trước truy vấn MySQL là biến \$movie_id

Sau đó bạn thêm vùng mới cho kết quả trả về từ truy vấn.

Thay đổi cuối cùng tạo mã HTML mà đưa ra liên kết trong tên movie.

Bây giờ việc thay đổi đã hoàn thành, thực tế nó làm gì? Nơi bạn đặt con chuột trên dòng liên kết, bạn sẽ thấy mỗi liên kết là duy nhất và tạo ra trang web động. Trang này được coi như trang chủ, trang sẽ liên kết tới như trang con

Tốt, ồ? Không có nhiều loại liên kết khác nhau

Trước khi bạn đi xa hơn, bạn cần thêm một vài dữ liệu cho cơ sở hiện hành của bạn.

Nếu bạn gọi lại từ chương 3, cho mỗi phim, bạn có tên phim, người lãnh đạo và năm.

Vì dụ: Thêm dữ liệu vào bảng

Trong bài này, bạn sẽ thêm dữ liệu về mỗi movie đến cơ sở dữ liệu

Mở trình soạn thảo của bạn và gõ đoạn mã sau:

```
<?php
$link = mysql_connect("localhost","root","")
or die(mysql_error());
mysql_select_db("moviesite")
or die (mysql_error());
```

```
//alter "movie" table to include running time/cost/takings fields
$add = "ALTER TABLE movie ADD COLUMN ( " .
"movie_running_time int NULL, " .
"movie_cost int NULL, " .
"movie_takings int NULL)";
$results = mysql_query($add)
or die(mysql_error());
//insert new data into "movie" table for each movie
$update = "UPDATE movie SET " .
"movie_running_time=102, " .
"movie_cost=10, " .
"movie_takings=15 " .
"WHERE movie_id = 1";
$results = mysql_query($update)
or die(mysql_error());
$update = "UPDATE movie SET " .
"movie_running_time=90, " .
"movie_cost=3, " .
"movie_takings=90 " .
"WHERE movie_id = 2";
$results = mysql_query($update)
or die(mysql_error());
$update = "UPDATE movie SET " .
"movie_running_time=134, " .
"movie_cost=15, " .
"movie_takings=10 " .
"WHERE movie_id = 3";
$results = mysql_query($update)
or die(mysql_error());
```

?>

Bạn lưu với tên alter_movie.php, sau đó mở file trong trình duyệt. Không lo lắng, bạn sẽ thấy một màn hình trống, nhưng bảng của bạn đã được thay đổi và thông tin đã đưa vào tự động

Cách thức làm việc:

Trước tiên, tập lệnh dùng lệnh ALTER TABLE để thêm vào những vùng tương thích trong bảng hiện hành, sau đó sử dụng lệnh UPDATE để chèn dữ liệu mới vào những vùng đó.

Bây giờ bạn có dữ liệu, bạn cần tạo một trang mới mà bạn sẽ hiển thị thêm thông tin(movie_details.php)

Ví dụ: Trong bài tập này, bạn sẽ tạo một trang mới để hiển thị dữ liệu mà bạn thêm vào bài trước.

Mở trình soạn thảo và gõ vào chương trình sau:

```
<?php
$link = mysql_connect("localhost","root","")
or die(mysql_error());
mysql_select_db("moviesite")
or die (mysql_error());
/* Function to calculate if a movie made a profit,
loss or broke even */
function calculate_differences($takings, $cost)
{
    $difference = $takings - $cost;
    if ($difference < 0)
    {
        $difference = substr($difference, 1);
        $font_color = 'red';
        $profit_or_loss = "$" . $difference . "m";
    }
}
```

```

elseif ($difference > 0)
    {
        $font_color = 'green';
        $profit_or_loss = "$" . $difference . "m";
    }
else
    {
        $font_color = 'blue';
        $profit_or_loss = "Broke even";
    }
return "<font color=\">$font_color\ ">" . $profit_or_loss . "</font>";
}
?>

```

Lưu file này như movie_details.php

Cách thức làm việc:

Dòng chứa mã substr thì đặt trước dòng \$profit_or_loss bởi vì sẽ có sự trả về số âm và không có thực.

Điều quan trọng cần nhớ là trong PHP bạn có thể dễ dàng tạo biến mới bằng một hoạt động. Bạn không tổ chức thông tin trong dữ liệu không có nghĩa là bạn không tạo được nó.

Ví dụ: Trình bày thông tin mới

Trong bài này bạn sẽ thay đổi bảng trang chủ với dữ liệu mới và điều này tương ứng như bảng con mới của bạn

Thêm đoạn mã dưới đây vào movie_details.php:

```

/* Function to get the director's name from the people table */
function get_director()
{
    global $movie_director;
    global $director;

```



```

$query_d = "SELECT people_fullname " .
"FROM people " .
"WHERE people_id='$movie_director'";
$results_d = mysql_query($query_d)
or die(mysql_error());
$row_d = mysql_fetch_array($results_d);
extract($row_d);
$director = $people_fullname;
}
/* Function to get the lead actor's name from the people table */
function get_leadactor()
{
    global $movie_leadactor;
    global $leadactor;
    $query_a = "SELECT people_fullname " .
"FROM people " .
"WHERE people_id='$movie_leadactor'";
$results_a = mysql_query($query_a)
or die(mysql_error());
$row_a = mysql_fetch_array($results_a);
extract($row_a);
$leadactor = $people_fullname;
}

$query = "SELECT * FROM movie " .
"WHERE movie_id ='" . $_GET['movie_id'] . "'";
$result = mysql_query($query, $link)
or die(mysql_error());
$movie_table_headings=<<<EOD
<tr>

```

```

        <th>Movie Title</th>
        <th>Year of Release</th>
        <th>Movie Director</th>
        <th>Movie Lead Actor</th>
        <th>Movie Running Time</th>
        <th>Movie Health</th>
    </tr>
    EOD;
    while ($row = mysql_fetch_array($result))
    {
        $movie_name = $row['movie_name'];
        $movie_director = $row['movie_director'];
        $movie_leadactor = $row['movie_leadactor'];
        $movie_year = $row['movie_year'];
        $movie_running_time = $row['movie_running_time']. " mins";
        $movie_takings = $row['movie_takings'];

        $movie_cost = $row['movie_cost'];
        //get director's name from people table
        get_director();
        //get lead actor's name from people table
        get_leadactor();
    }

```

Cách thức hoạt động:

Bởi vì bạn đã được viết lại hàm để tạo tên đạo diễn và tên diễn viên, bạn “mượn” mã này từ bài table2.php. Sau đó bạn chuyển truy vấn để trả về mỗi thứ trong mỗi mẫu tin, như tương phản (opposed) một vài trường. Nó có nghĩa là bạn trả về một trường mà bạn không thực sự sử dụng. Câu truy vấn chứa mệnh đề WHERE. Mẫu tin bạn sẽ lấy lại dữ liệu từ quyết định này

Mệnh đề WHERE

- Bạn dùng \$_GET['movie_id'] trong mệnh đề WHERE. Đây là ID của movie mà được liên kết từ tabl3.php
- Bạn cũng tạo một biến khác *\$movie_table_headings* chứa phần đầu bạn sẽ sử dụng.
- Sự dừng lại của đoạn mã thì rất giống đoạn mã trong table3.php. Bạn thêm vào nhiều hơn 4 trường để WHILE kiểm soát vòng lặp.

Chúng tôi có nói phần trước là những vùng trả về mà bạn không cần thì hoạt động không tốt? Vâng, đúng như vậy. Tuy nhiên, trường hợp này bạn chỉ có nhiều hơn một trường bạn cần, như tương phản để trả về nhiều trường dư thừa. Vì thế, chúng tôi sẽ đưa ra lời khuyên? Chính xác 100%. Tuy nhiên, bởi vì bạn sử dụng nhiều trường trong mỗi mẫu tin, PHP sẽ không chứa từ cân bằng này, và nó có giá trị. Bạn sẽ không muốn làm điều này khi mà bạn muốn những giá trị của 5 trường và cấu trúc mẫu tin chứa 50 trường. Nếu bạn làm điều này, PHP sẽ hao mòn về tài nguyên để trả về 45 trường khác.

Ví dụ: Trình bày chi tiết phim ảnh

Trong bài tập này, bạn sẽ tăng cường trang movie_details với dữ liệu mới.

Thêm vào những dòng mã sau vào cuối movie_details.php

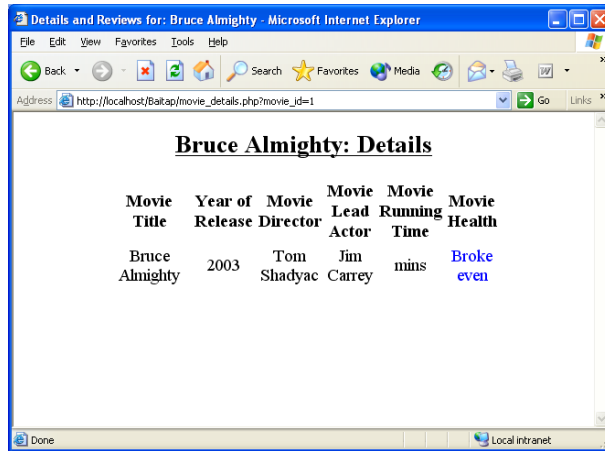
```
$movie_health = calculate_differences($movie_takings, $movie_cost);
$page_start =<<<<EOD
<html>
<head>
<title>Details and Reviews for: $movie_name</title>
</head>
<body>
EOD;
$movie_details =<<<<EOD
<table width="70%" border="0" cellspacing="2"
cellpadding="2" align="center">
<tr>
```

```

<th colspan="6"><u><h2>$movie_name: Details</h2></u></th>
</tr>
$movie_table_headings
<tr>
<td width="33%" align="center">$movie_name</td>
<td align="center">$movie_year</td>
<td align="center">$director</td>
<td align="center">$leadactor</td>
<td align="center">$movie_running_time</td>
<td align="center">$movie_health</td>
</tr>
</table>
<br>
<br>
EOD;
$page_end =<<<EOD
</body>
</html>
EOD;
    $detailed_movie_info =<<<EOD
    $page_start
    $movie_details
    $page_end
EOD;
echo $detailed_movie_info;
mysql_close();

```

Lưu file như `movie_details.php`, đưa lên server, mở trình duyệt `tbl3.php`, click vào tên movie, bạn sẽ thấy như **Hình 4.3.2**



Hình 4.3.2

Cách thức hoạt động:

Bạn nhớ hàm bạn tạo ở trên? Khi bạn thêm dòng trong bước 1 ở phần ví dụ trước, bạn gọi hàm và yêu cầu nó thi hành. Bất cứ giá trị nào được trả về từ hàm `calculate_difference` sẽ là nơi biến `$movie_health`. Thông qua biến `$movie_taking` và `$movie_costs` để hàm đưa ra kết quả chính xác.

Khi bạn định nghĩa biến `$page_start`, bạn bắt đầu sắp xếp cấu trúc trang thực sự. Bằng cách thêm vào biến `$movie_name`, bạn có thể thiết lập tựa đề trình duyệt. Bạn có thể thấy cú pháp `=<<<EOD` tiện lợi thế nào?

Kế đến, bạn định nghĩa biến `$movie_details`. Điều này hoàn toàn giải thích giải thích được. Nhớ biến `$movie_table_headings` bạn đã tạo trước. Tất cả bạn làm trong biến `$movie_details` và nó xuất hiện.

Cuối cùng, bạn định nghĩa biến `$page_end`.

Chương 4 : SỬ DỤNG BẢNG ĐỂ TRÌNH BÀY DỮ LIỆU

Chúng ta có thể thành công trong việc kết hợp PHP và MySQL để tạo ra trang web động, khi có nhiều dòng dữ liệu cần trình bày, chúng ta cần phải có cơ chế để người xem dễ dàng đọc dữ liệu và cần phải đẹp, ngắn gọn, trình bày rõ ràng. Phương pháp dễ nhất là sử dụng bảng.

Trong chương này chúng ta sẽ biết:

- Tạo bảng để tổ chức dữ liệu từ cơ sở dữ liệu.
- Tạo tiêu đề cột tự động.
- Bảng thông dụng với kết quả truy vấn MySQL cơ bản.
- Bảng thông dụng với kết quả truy vấn MySQL phức tạp.
- Tạo trang thân thiện cho người dùng

Quan hệ lâu dài

Bạn muốn tìm gì từ những cái được thấy lại? Bạn cần tạo một truy vấn SQL mới trong trang movie_details.php và thi hành nó khi nó được nạp, nó sẽ làm tổng 2 truy vấn trong một trang. Nó sẽ làm việc, nhưng nó thuận lợi.

Đó là thời gian để trả lời câu hỏi, quan hệ là gì?

Một quan hệ là một cách của việc kết hợp nhiều bảng vì thế bạn có thể cập nhật dữ liệu trong tất cả những bảng. Lợi ích của MySQL là cơ sở dữ liệu quan hệ, hỗ trợ cho việc tạo lập các bảng. Khi sử dụng những quan hệ chính xác có thể rất hữu ích và có thể dùng để lấy lại dữ liệu từ nhiều bảng trong truy vấn SQL.

Ví dụ: Tạo và điền một bảng hình ảnh xem lại

Trước tiên bạn có thể cập nhật những bảng xem lại, bạn cần tạo bảng và điền dữ liệu vào.

Mở trình soạn thảo và gõ vào đoạn mã sau:

```
<?php
```

```
//connect to MySQL
```

```

$connect = mysql_connect("localhost", "root", "")
    or die ("Hey loser, check your server connection.");
mysql_select_db("moviesite");
//create "reviews" table
$reviews = "CREATE TABLE reviews (
    review_movie_id int(11) NOT NULL,
    review_date date NOT NULL,
    review_name varchar(255) NOT NULL,
    review_reviewer_name varchar(255) NOT NULL,
    review_comment varchar(255) NOT NULL,
    review_rating int(11) NOT NULL default 0,
    KEY (review_movie_id))";
$results = mysql_query($reviews)
    or die (mysql_error());
//populate the "reviews" table
$insert = "INSERT INTO reviews
(review_movie_id, review_date, review_name,
review_reviewer_name, review_comment, review_rating)
VALUES
('1', '2003-08-02', 'This movie rocks!',
'John Doe', 'I thought this was a great movie even though
my girlfriend made me see it against my will.', '4'),
('1', '2003-08-01', 'An okay movie',
'Billy Bob', 'This was an okay movie. I liked Eraserhead
better.', '2'),
('1', '2003-08-10', 'Woo hoo!',
'Peppermint Patty', 'Wish I\'d have seen it sooner!', '5'),
('2', '2003-08-01', 'My favorite movie',
'Marvin Marian', 'I didn\'t wear my flair to the movie but

```

```

        I loved it anyway.', '5'),
        ('3', '2003-08-01', 'An awesome time',
        'George B.', 'I liked this movie, even though I thought it
        was an informational video from our travel agent.', '3')");
$insert_results = mysql_query($insert)
        or die(mysql_error());

```

?>

Lưu file như `createreviews.php`, đưa lên server, mở trình duyệt, chúng ta sẽ thấy bảng được tạo và điền đầy đủ.

Cách thức hoạt động:

Chúng ta làm như tạo bảng với PHP và MySQL

Ví dụ: Truy vấn những cái được xem lại

Trong ví dụ này, chúng ta sẽ liên kết 2 bảng (`movies` và `review`) để xem lại. Những yêu cầu này có nhiều thay đổi đến trong `movies_details.php`, vì thế cách tốt nhất là bạn copy file. Sau đó làm những bước sau:

Mở trình soạn thảo `movies_detail.php`

Thay đổi đoạn mã dưới đây:

```

$movie_query = "SELECT * FROM movie " .
"WHERE movie_id =" . $_GET['movie_id'] . """;
$movie_result = mysql_query($movie_query, $link)
        or die(mysql_error());

```

Và sau đoạn mã, thay đổi:

```

while ($row = mysql_fetch_array($movie_result)) {
    $movie_name = $row['movie_name'];
    $movie_director = $row['movie_director'];

```

Và thêm những dòng sau dấu ngoặc “đóng” trong lệnh While

```

$review_query = "SELECT * FROM reviews " .
"WHERE review_movie_id =" . $_GET['movie_id'] . "" .
"ORDER BY review_date DESC";

```



```
$review_result = mysql_query($review_query, $link)
or die(mysql_error());
```

Cách thức hoạt động:

Bạn thay đổi tên của biến `$query` cho biến `$movie_query`, thay `$result` cho `$movie_result`. Làm việc này bảo đảm không làm bạn cảm thấy rắc rối khi bạn cập nhật những kết quả liên quan đến sự trả về của truy vấn. Đây cũng là mệnh đề sắp xếp, mà chắc rằng những cái xem lại gần đây nhất ở trên đầu trang.

Một lỗi chủ yếu là nhiều người bắt đầu làm thì dùng tên biến như nhau khi tạo truy vấn SQL. Thừa nhận rằng bạn copy và dán đơn giản làm nhẹ bớt `movie query` và `movie result` khi `query` được gọi. Bạn có hai truy vấn SQL được gọi truy vấn, và hai kết quả gọi là `$result`. Khi kết quả đầu tiên chạy nó sẽ xuất ra những kết quả mong đợi. Tuy nhiên nếu bạn muốn chuyển đến những kết quả được trả về từ SQL đầu tiên bạn sẽ gặp vấn đề lớn.

Tại sao như vậy? Kết quả đầu tiên sẽ đè lên kết quả truy vấn lần thứ 2, với lý do này bạn phải luôn cẩn trọng khi dùng nhiều tên khác nhau cho truy vấn SQL và trả về kết quả từ truy vấn.

Ví dụ: Trình bày những review

Thêm dòng sau:

```
function generate_ratings($review_rating)
{
    $movie_rating = '';
    for($i=0; $i<$review_rating; $i++)
    {
        $movie_rating .= "<img src=\"thumbsup.gif\">&nbsp;";
    }
    return $movie_rating;
}
```

Thêm những đoạn mã trong những dòng sau:

```

$review_table_headings=<<

```

Bạn cần thêm vài dòng tiếp theo sau phần đầu bảng:

```

while($review_row = mysql_fetch_array($review_result))
{
    $review_flag =1;
    $review_title[] = $review_row['review_name'];
    $reviewer_name[] =
ucwords($review_row['review_reviewer_name']);
    $review[] = $review_row['review_comment'];
    $review_date[] = $review_row['review_date'];
    $review_rating[] =
generate_ratings($review_row['review_rating']);
}

```

Tiếp theo bạn thêm những dòng sau:

```

$i = 0;
$review_details = "";
while ($i<sizeof($review))
{
    $review_details .=<<

```

```

        <td width="15%" valign="top">$review_title[$i]</td>
        <td width="10%" valign="top">$reviewer_name[$i]</td>
        <td width="50%" valign="top">$review[$i]</td>
        <td width="10%" valign="top" align="center">$review_rating[$i]</td>
    </tr>
    EOD;
    $i++;
}

```

Thay đổi như dưới đây, làm cẩn thận và chính xác:

```

    <td>$movie_health</td>
</tr>
</table>
<br>
<br>
EOD;
if ($review_flag)
{
    $movie_details .=<<<EOD
    <table width="95%" border="0" cellpadding="2"
    cellspacing="20" align="center">
    $review_table_headings
    $review_details
    </table>
    EOD;
}

```

Lưu file movie_details.php

Đưa lên server, mở trình duyệt table3.php và click vào movie

Bạn sẽ thấy như **Hình 4.4**

The screenshot shows a web browser window titled "Details and Reviews for: Bruce Almighty - Microsoft Internet Explorer". The address bar shows a local host URL. The main content area displays the movie details for "Bruce Almighty" in a table format, followed by a table of reviews.

Bruce Almighty: Details					
Movie Title	Year of Release	Movie Director	Movie Lead Actor	Movie Running Time	Movie Health
Bruce Almighty	2003	Tom Shadyac	Jim Carrey	mins	Broke even

Date of Review	Review Title	Reviewer Name	Movie Review Comments	Rating
2003-08-10	Woo hoo!	Peppermint Patty	Wish I'd have seen it sooner!	✓✓✓✓
2003-08-02	This movie rocks!	John Doe	I thought this was a great movie even though my girlfriend made me see it against my will.	✓✓✓✓
2003-08-01	An okay movie	Billy Bob	This was an okay movie. I liked Eraserhead better.	✓✓

Hình 4.4

Cách thức hoạt động:

Hàm `generate_ratings` là hàm dễ hiểu. Bạn gọi nó cho biến là trường không tự chủ (rating) cho movie và nó tạo một hình "rating", trả về nó. Chú ý rằng bạn đang sử dụng `.=` (như `.=<<<<`). Đảm bảo rằng những hình ảnh với một "rating" nhiều hơn 1 sẽ tạo những hình thêm vào hình rating đơn giản.

Biến `$review_table_headings` chứa những phần đầu bảng cho việc xem lại mà bạn vừa lấp đầy qua truy vấn SQL trước.

Biến `$review_table_headings` chứa những phần đầu của bảng cho review mà bạn vừa điền đầy đủ trong truy vấn trước. Điều này sử dụng chính xác khái niệm như phần đầu bảng movie trong ví dụ trước. Vì thế bạn review tất cả những phần đầu bảng.

Tập lệnh `WHILE` là những dòng tổ chức của những review, nếu có bất kỳ những review nào cho movie, bạn thiết lập một cờ cho biết sử dụng biến `$review_flag`. Mã này tạo mảng tổ chức những giá trị sẽ trả về. Tại sao lại đặt chúng trong mảng và không hoàn toàn là biến bình thường? Điều này cho phép biến tổ chức dữ liệu nhiều hơn một review cho hình ảnh. Sau tất cả những điều này, bạn mong rằng sẽ có nhiều "review" cho mỗi hình ảnh. Nếu bạn không tạo biến review như mảng, sau đó bạn trả về review cuối cho hình ảnh. Trong phần thảo luận trước, chúng ta thấy là tại sao chúng ta ưu tiên đặt những giá trị vào hơn là xuất ra những giá trị. Nhìn vào dòng `review_name` bạn chú ý rằng

chúng ta đặt dòng `$review_row` bên trong hàm `ucwords` PHP. Điều này cho phép bạn thực hiện hàm `ucwords` tự động trong giá trị trả về từ trường đó.

Mã sau vòng lặp thông qua mảng và gán những giá trị cho mỗi trường mà bạn sẽ trình bày cho người xem. Bạn sử dụng hàm `sizeof` PHP để tính toán có bao nhiêu mẫu tin được trả về .

Cuối cùng, bạn chia biến `$movie_details` thành đoạn nhỏ và thêm chúng qua việc sử dụng `.=<<<`. Như bạn làm trước, bạn sử dụng định nghĩa biến và đặt nơi chính xác. Nếu còn review được thiết lập, bạn sẽ thấy những mục tạo thành những review

Bạn có thay đổi trong phần này nhưng bạn thấy là việc thay đổi được đánh giá cao. Bây giờ bạn biết sử dụng MySQL như thế nào để tạo quan hệ giữa các bảng. Bạn thành công trong việc lấy lại những review từ bảng review phụ thuộc vào biến `movie_id`. Bạn cũng sử dụng `$_GET` bao trùm qua giá trị từ một trang .

Chương 5 : NHỮNG THÀNH PHẦN CỦA FORM: CHO PHÉP NGƯỜI DÙNG LÀM VIỆC VỚI DỮ LIỆU

Trong ứng dụng Web, người sử dụng nhập nội dung, nhấn submit để xử lý. Một tiến trình thì không được hoàn thành bởi lệnh PHP, vì thế mã lệnh yêu cầu phải “thông minh”.

HTML sẽ gửi đến một vị trí đặc biệt và xử lý vì khi bạn điền thông tin vào form, để biết một địa chỉ email, mail,... bạn cần một phương tiện để đọc nội dung.

Form trong HTML thì đơn giản hơn, nó chỉ vị trí và cách thức gửi như thế nào. Tại điểm này PHP được ứng dụng hơn. Tập lệnh PHP nhận dữ liệu từ form và sử dụng nó để hoàn thành hoạt động, như cập nhật những nội dung của cơ sở dữ liệu, gửi một email, định dạng dữ liệu và hơn thế nữa.

PHP sử dụng một tập đơn giản mà mạnh, một sự kết hợp, cung cấp phương tiện để làm mọi thứ ảo khi chúng ta cần.

Trong chương này bạn bắt đầu xây dựng một ứng dụng đơn giản cho phép bạn thêm, bớt, xóa thành phần của dữ liệu. Chúng ta sẽ được đến với PHP/MySQL như:

- Tạo những form sử dụng nút, hộp soạn thảo và những thành phần khác.
- Tạo tập lệnh PHP để xử lý những form HTML.
- Nhận dữ liệu từ hai biến chính là: `$_POST` và `$_GET`.
- Qua thông tin ẩn để xử lý tập lệnh form với những điều khiển form ẩn và một chuỗi truy vấn URL.

Form đầu tiên

Bắt đầu là một form đơn giản chỉ có vùng soạn thảo và nút submit, tập lệnh xử lý sẽ hiển thị giá trị nhập vào vùng soạn thảo

Ví dụ: Trong bài tập này bạn sẽ điền tên vào form. Đây là một biến đơn giản trong chương trình “Hello Word”.

1/Mở trình soạn thảo nhập đoạn mã sau và lưu với tên form1.html

```

<html>
<head>
<title>Say My Name</title>
<style type="text/css">
TD{color:#353535;font-family:verdana}
TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
</style>
</head>
<body>
<form action="formprocess1.php" method="post">
<table border="0" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
  <tr>
    <td bgcolor="#FFFFFF" width="50%">Name</td>
    <td bgcolor="#FFFFFF" width="50%">
      <input type="text" name="Name"><br></td>
  </tr>
  <tr>
    <td bgcolor="#FFFFFF" colspan="2" align="center">
      <input type="submit" name="SUBMIT" value="Submit">
    </td>
  </tr>
</table>
</form>
</body>
</html>

```

2/ Mở trình soạn thảo mới nhập đoạn mã sau và lưu với tên formprocess1.php:

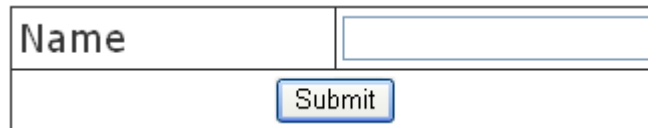
```

<html>
<head>

```

```
<title>Say My Name</title>
</head>
<body>
<?php
    echo "Hello " . $_POST['Name'];
    ?>
<pre>
    DEBUG :
<?php
    print_r($_POST);
?>
</pre>
</body>
</html>
```

3/Mở trình duyệt với file form1.html sẽ cho kết quả như **Hình 5.1.1**



The image shows a simple web form. It consists of a rectangular container. Inside, there is a label 'Name' on the left, followed by an empty text input field. Below the input field, centered, is a button labeled 'Submit'.

Hình 5.1

Nhập vào khung **Name** nội dung bất kỳ(ví dụ Test) và click nút Submit bạn sẽ thấy kết quả như sau:

```
Hello Test
DEBUG :
Array
(  
    [Name] => Test  
    [SUBMIT] => Submit
```


)

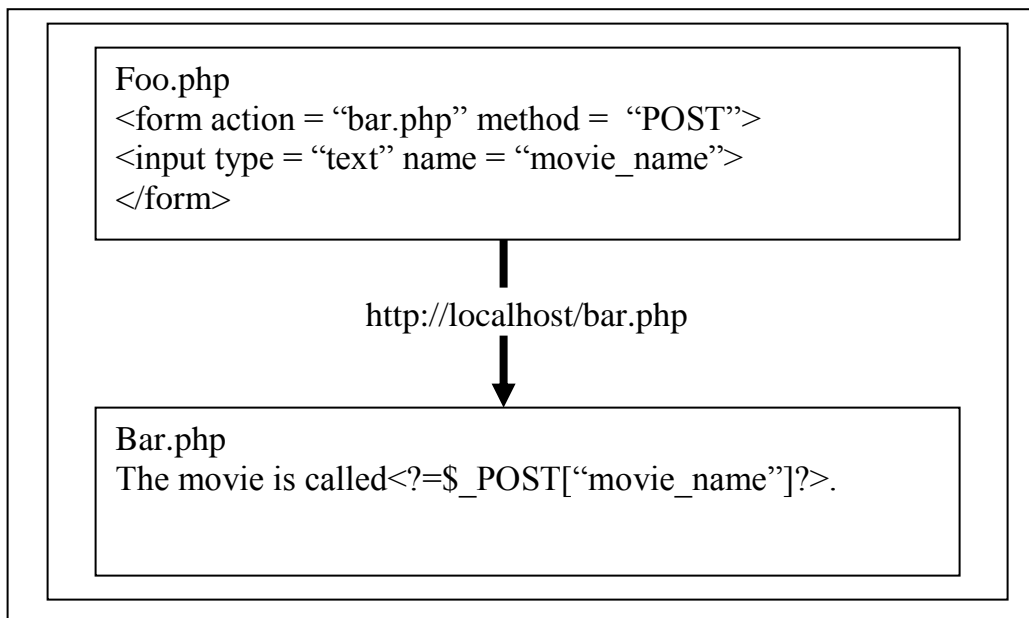
Cách thức hoạt động :

Để hiểu về cách thức hoạt động, chúng ta phải tìm hiểu về những thành phần của form và vài hàm PHP mới.

Thành phần form

Thành phần mà bạn cần biết trước tiên là FORM. Nó giới hạn vùng form trong trang và tổ chức những vùng mà bạn cần từ trang web:

```
<form action="formprocess1.php" method="post">  
<!--form controls here-->  
</form>
```



Chú ý: Thành phần Form có một thẻ kết thúc và hai thuộc tính.

1. Action: là cách thức chỉ đến đường dẫn.
2. Method: là cách thức mà bạn sẽ gửi dữ liệu đến người nhận. Gửi có hai phương pháp. Đó là POST và GET.

Phương pháp POST lấy dữ liệu từ trường form và gửi nó thông qua phần đầu HTTP. Trong trường hợp này dữ liệu không thể thấy qua URL

Phương thức GET lấy dữ liệu từ trường form, mã hóa nó, và thêm nó để gửi tới URL như trình bày dưới đây:

```
http://localhost/formprocess1.php?field1=valuea&field2=value%20b
```

Bạn có thể thấy, tên trường và giá trị của chúng thì dễ dàng đọc trong tập lệnh URL. Những tham số của tập lệnh trong URL cho phép người dùng có thể thay đổi, điều khiển chúng. Điều này có thể dẫn đến lỗi trong quá trình xử lý tập lệnh hoặc cập nhật dữ liệu(không phải là lần cập nhật đầu tiên.)

Phần tử nhập (INPUT)

Phần tử HTML mới thứ hai là INPUT. Đây là hình thức cơ bản và có thể sử dụng trong nhiều phương pháp khác nhau để tập hợp nhiều loại thông tin khác nhau. Trong trường hợp này bạn có thể sử dụng kiểu INPUT khác nhau: kiểu văn bản(text) và kiểu submit.

- Kiểu INPUT văn bản:

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

Kiểu INPUT văn bản là một tiêu chuẩn, hộp văn bản chỉ có những dòng đơn. Nó sẽ cập nhật nội dung thông qua cú pháp:

```
<?php  
    echo $_POST['Name']; // sẽ hiển thị một kiểu giá trị  
?>
```

- Kiểu INPUT submit:

```
<input type="submit" name="SUBMIT" value="Submit">
```

Tên của nó phải gợi ý khéo léo, phần tử submit hiển thị bằng một nút. Nút văn bản được đặt thông qua thuộc tính giá trị. Như phương pháp cho INPUT văn bản, điều khiển form này cần có tên cho tiến trình chuyển đến.

Xử lý Form

Trong lệnh, chúng ta phải chú ý đến hàm và cú pháp mới để tìm hiểu về chúng

Tập lệnh xử lý Form đầu tiên là một biến tương tác của “hello world” phổ biến, nhưng trong trường hợp này nó hiển thị “hello” và tên của bạn trong hộp text. Để điều này xảy ra, bạn cần in giá trị của trường văn bản mà bạn điền trong form.

Bạn biết lệnh xuất, vì thế chúng ta hãy di chuyển đến một bộ phận, `$_POST[‘tên’]`.

Mảng toàn cục `$_POST` chứa tất cả các form dữ liệu submit với phương pháp POST. Danh mục mảng của trường là tên của nó. Trong thời điểm này bạn sẽ thấy cách kiểm tra nội dung mảng `$_POST` sử dụng hàm `print_r()` như thế nào.

```
<?php
    echo “Hello “. $_POST[‘Name’];
?>
```

Trong ví dụ này, `$_POST[‘name’]` hiển thị cái gì bạn nhập vào trong hộp “Name.”Hello test”. Ở đây `print_r($_POST)` kết xuất đơn giản nội dung chung của mảng toàn cục `$_POST` để xuất ra. Đây là cách tốt nhất để biên dịch form. Nên sử dụng `$_POST` để kiểm tra, trình bày trạng thái đối tượng khi xây dựng tập lệnh.

Trong tập lệnh `formprocess1.php` xuất ra một vài thứ tương tự như sau

```
Hello test
DEBUG :
Array
(
    [Name] => test
    [SUBMIT] => Submit
)
```

Khi nhận form submit, PHP đặt mảng POST với dữ liệu mà form gửi tới. Như với bất kì mảng nào, bạn có thể cập nhật chính xác bất kì danh mục nào bằng tên. Trong ví dụ này, bạn có thể thấy rõ danh mục tên chứa giá trị test. Cách này làm việc với tất cả các form, ngay cả với những form phức tạp nhất.

Chúng ta hãy đi tiếp để thấy việc sử dụng những phần tử HTML trong suốt quá trình nhập form với giao diện người dùng.

Chương 5 : NHỮNG THÀNH PHẦN CỦA FORM: CHO PHÉP NGƯỜI DÙNG LÀM VIỆC VỚI DỮ LIỆU

Trong ứng dụng Web, người sử dụng nhập nội dung, nhấn submit để xử lý. Một tiến trình thì không được hoàn thành bởi lệnh PHP, vì thế mã lệnh yêu cầu phải “thông minh”.

HTML sẽ gửi đến một vị trí đặc biệt và xử lý vì khi bạn điền thông tin vào form, để biết một địa chỉ email, mail,... bạn cần một phương tiện để đọc nội dung.

Form trong HTML thì đơn giản hơn, nó chỉ vị trí và cách thức gửi như thế nào. Tại điểm này PHP được ứng dụng hơn. Tập lệnh PHP nhận dữ liệu từ form và sử dụng nó để hoàn thành hoạt động, như cập nhật những nội dung của cơ sở dữ liệu, gửi một email, định dạng dữ liệu và hơn thế nữa.

PHP sử dụng một tập đơn giản mà mạnh, một sự kết hợp, cung cấp phương tiện để làm mọi thứ ảo khi chúng ta cần.

Trong chương này bạn bắt đầu xây dựng một ứng dụng đơn giản cho phép bạn thêm, bớt, xóa thành phần của dữ liệu. Chúng ta sẽ được đến với PHP/MySQL như:

- Tạo những form sử dụng nút, hộp soạn thảo và những thành phần khác.
- Tạo tập lệnh PHP để xử lý những form HTML.
- Nhận dữ liệu từ hai biến chính là: `$_POST` và `$_GET`.
- Qua thông tin ẩn để xử lý tập lệnh form với những điều khiển form ẩn và một chuỗi truy vấn URL.

Driving the User Input

Form trong ví dụ này, hướng dẫn người dùng chọn những giá trị từ một tập hợp giá trị mà được cung cấp.

Tập hợp giá trị thông qua việc sử dụng những phần tử HTML đặc biệt: list boxes, radio buttons và checkboxes.

Có 2 loại trong form:

Loại cho phép người dùng chọn một mục từ những chọn lựa có sẵn.

Loại cho phép chọn nhiều mục.

Hộp danh sách xổ xuống và nút radio chỉ chọn một. Những checkbox và những hộp danh sách cung cấp nhiều lựa chọn

Ví dụ: Giới hạn của việc chọn lựa

Chúng ta hãy bắt đầu với một kiểu nhập đơn giản. Những bước sau là để tạo một danh sách đơn giản:

1 Tạo một file form2.html và mở nó trong trình soạn thảo của bạn

2. Nhập đoạn mã sau:

```
<html>
<head>
<title>Greetings Earthling</title>
<style type="text/css">
    TD{color:#353535;font-family:verdana}
    TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
</style>
</head>
<body>
    <form action="formprocess2.php" method="post">
    <table border="0" cellspacing="1" cellpadding="3"
    bgcolor="#353535" align="center">
<tr>
    <td bgcolor="#FFFFFF" width="50%">Name</td>
    <td bgcolor="#FFFFFF" width="50%">
    <input type="text" name="Name">
    </td>
</tr>
<tr>
```

```

        <td bgcolor="#FFFFFF">Greetings</td>
        <td bgcolor="#FFFFFF">
            <select name="Greeting">
                <option value="Hello">Hello</option>
                <option value="Hola">Hola</option>
                <option value="Bonjour">Bonjour</option>
            </select>
        </td>
    </tr>
    <tr>
        <td bgcolor="#FFFFFF" width="50%">Display Debug info</td>
        <td bgcolor="#FFFFFF" width="50%">
            <input type="checkbox" name="Debug" checked>
        </td>
    </tr>
    <tr>
        <td bgcolor="#FFFFFF" colspan=2 align="center">
            <input type="submit" name="SUBMIT" value="Submit">
        </td>
    </tr>
</table>
</form>
</body>
</html>

```

3. Tạo một file mới với tên formprocess2.php và nhập đoạn mã sau:

```

<html>
<head>
<title>Greetings Earthling</title>
<style type="text/css">

```

```

        TD{color:#353535;font-family:verdana}
        TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
</style>
</head>
<body>
<?php
    if (isset($_POST['Debug']) and $_POST['Debug'] == "on") {
?>
<pre>
<?php
    print_r($_POST);
?>
</pre>
<?php
}
?>
    <p align="center"><?php echo $_POST['Greeting']; ?>
    <?php echo $_POST['Name']; ?></p>
</body>
</html>


```

4. Lưu formprocess2.php và tải nó lên thư mục làm việc của bạn.
5. Gọi trang trình duyệt của bạn. Bạn có thể thấy kết quả như **hình 5.2.1**

Name	<input type="text" value="Chuong 5"/>
Greetings	<input type="text" value="Hello"/> ▾
Display Debug info	<input checked="" type="checkbox"/>
<input type="button" value="Submit"/>	

Hình 5.2.1

6. nhập tên của bạn và ấn nút submit .Kết quả như **hình 5.2.2**



```
Address http://localhost/Baitap/formprocess2.php

Array
(
    [Name] => Chuong 5
    [Greeting] => Hello
    [Debug] => on
    [SUBMIT] => Submit
)

Hello Chuong 5
```

Hình 5.2.2

Cách thức hoạt động:

Như bạn thấy đoạn mã này, trình tự giống như trong formprocess1.php. Hai trường có thêm một hộp xổ xuống và một checkbox. formprocess2.php giống như formprocess1.php nhưng thêm một hộp cuộn. Trình bày thông tin biên dịch chỉ khi hộp kiểm Debug được chọn và bạn dùng bất kì chọn lựa nào trong danh sách xổ xuống .

Kiểu nhập hộp checkbox

Checkbox có thể được trình bày ở hai vị trí: Khi được kiểm tra, nó thông qua giá trị trên mảng \$_POST, nhưng mặc khác nó không gửi đi mọi thứ. Đây là phương pháp hay để trình bày kiểu dữ liệu Boolean.

* SELECT element

```
<select name="Greeting">
<option value="Hello">Hello</option>
<option value="Hola">Hola</option>
<option value="Bonjour">Bonjour</option>
</select>
```

Phần tử SELECT cho phép bạn trình bày một danh sách chọn lựa cố định mà người dùng có thể chọn một phần tử. Mục chọn sẽ không được gửi như hiển thị nhưng sẽ gửi giá trị của nó. Trong ví dụ này, giá trị và hiển thị là giống nhau, nhưng trong một hệ thống ràng buộc cơ sở dữ liệu, bạn có thể thấy ID của mẫu tin như những giá trị và nhãn văn bản của chúng là danh sách chọn.

Khi sử dụng danh sách, phải thiết lập những phần giá trị mục chọn OPTION. Nếu ở đây không thiết lập tương tự như thấy trong danh sách, nhưng hoàn toàn không sử dụng bởi vì tất cả chọn lựa sẽ gửi như giá trị rỗng

Chương 5 : NHỮNG THÀNH PHẦN CỦA FORM: CHO PHÉP NGƯỜI DÙNG LÀM VIỆC VỚI DỮ LIỆU

Trong ứng dụng Web, người sử dụng nhập nội dung, nhấn submit để xử lý. Một tiến trình thì không được hoàn thành bởi lệnh PHP, vì thế mã lệnh yêu cầu phải “thông minh”.

HTML sẽ gửi đến một vị trí đặc biệt và xử lý vì khi bạn điền thông tin vào form, để biết một địa chỉ email, mail,... bạn cần một phương tiện để đọc nội dung.

Form trong HTML thì đơn giản hơn, nó chỉ vị trí và cách thức gửi như thế nào. Tại điểm này PHP được ứng dụng hơn. Tập lệnh PHP nhận dữ liệu từ form và sử dụng nó để hoàn thành hoạt động, như cập nhật những nội dung của cơ sở dữ liệu, gửi một email, định dạng dữ liệu và hơn thế nữa.

PHP sử dụng một tập đơn giản mà mạnh, một sự kết hợp, cung cấp phương tiện để làm mọi thứ ảo khi chúng ta cần.

Trong chương này bạn bắt đầu xây dựng một ứng dụng đơn giản cho phép bạn thêm, bớt, xóa thành phần của dữ liệu. Chúng ta sẽ được đến với PHP/MySQL như:

- Tạo những form sử dụng nút, hộp soạn thảo và những thành phần khác.
- Tạo tập lệnh PHP để xử lý những form HTML.
- Nhận dữ liệu từ hai biến chính là: `$_POST` và `$_GET`.
- Qua thông tin ẩn để xử lý tập lệnh form với những điều khiển form ẩn và một chuỗi truy vấn URL.

Một form đa tiến trình

Form luôn luôn tác động lại theo một phương thức xác định trước, dựa vào việc mã hóa tập lệnh để chỉ định dữ liệu mà người dùng gửi tới hệ thống như thế nào. Một form đơn giản có thể có nhiều định nghĩa hoạt động bằng việc sử dụng những nút submit khác nhau.

Ví dụ: Radio Button, Multiline List Boxes

Trong ví dụ sau, bạn tạo ra một form để chuẩn bị một nghiên cứu và tạo một giao diện film/điện viên/đạo diễn.

1. tạo một file form3.php và nhập đoạn mã sau:

```
<html>
<head>
<title>Add/Search Entry</title>
<style type="text/css">
    TD{color:#353535;font-family:verdana}
    TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
</style>
</head>
<body>
    <form action="formprocess3.php" method="post">
    <table border="0" cellspacing="1" cellpadding="3"
    bgcolor="#353535" align="center">
<tr>
    <td bgcolor="#FFFFFF" width="50%">Name</td>
    <td bgcolor="#FFFFFF" width="50%">
    <input type="text" name="Name">
    </td>
</tr>
<tr>
    <td bgcolor="#FFFFFF">What you are looking for</td>
    <td bgcolor="#FFFFFF">
    <select name="MovieType">
    <option value="" selected>Select a movie type...</option>
    <option value="Action">Action</option>
    <option value="Drama">Drama</option>
    <option value="Comedy">Comedy</option>
```

```

        <option value="Sci-Fi">Sci-Fi</option>
        <option value="War">War</option>
        <option value="Other">Other...</option>
    </select>
</td>
</tr>
<tr>
    <td bgcolor="#FFFFFF">Add what?</td>
    <td bgcolor="#FFFFFF">
        <input type="radio" name="type" value="Movie" checked>
        Movie<br>
        <input type="radio" name="type" value="Actor">
        Actor<br>
        <input type="radio" name="type" value="Director">
        Director<br>
    </td>
</tr>
<tr>
    <td bgcolor="#FFFFFF" width="50%">Display Debug info</td>
    <td bgcolor="#FFFFFF" width="50%">
        <input type="checkbox" name="Debug" checked>
    </td>
</tr>
<tr>
    <td bgcolor="#FFFFFF" colspan=2 align="center">
        <input type="submit" name="Submit" value="Search">
        <input type="submit" name="Submit" value="Add">
    </td>
</tr>

```

```
</table>
</form>
</body>
</html>
```

2. Tạo file khác tên formprocess3.php và nhập đoạn mã sau:

```
<?php
    if ($_POST['type'] == "Movie" && $_POST['MovieType'] == "")
    {
        header("Location:form3.php");
    }
    $title = $_POST['Submit'] . " " .
    $_POST['type'] . " : " .
    $_POST['Name'];
?>
<html>
<head>
<title><?php echo $title; ?></title>
</head>
<body>
<?php
    if ($_POST['Debug'] == "on") {
?>
<pre>
<?php
    print_r($_POST);
?>
</pre>
<?php
}
```

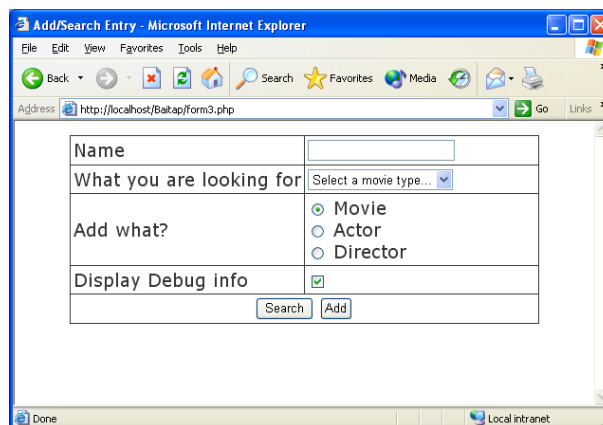
```

$name = $_POST['Name'];
$name[0] = strtoupper($name[0]);
if ($_POST['type'] == "Movie")
{
    $foo = $_POST['MovieType'] . " " . $_POST['type'];
} else
{
    $foo = $_POST['type'];
}
?>

<p align="center">
You are <?php echo $_POST['Submit']; ?>ing
<?php echo $_POST['Submit'] == "Search" ? "for " : ""; ?>
a <?php echo $foo ?>
named "<?php echo $name; ?>"
</p>
</body>
</html>

```

3. Mở <http://localhost/form3.php>. Kết quả như **Hình 5.2.2.1**



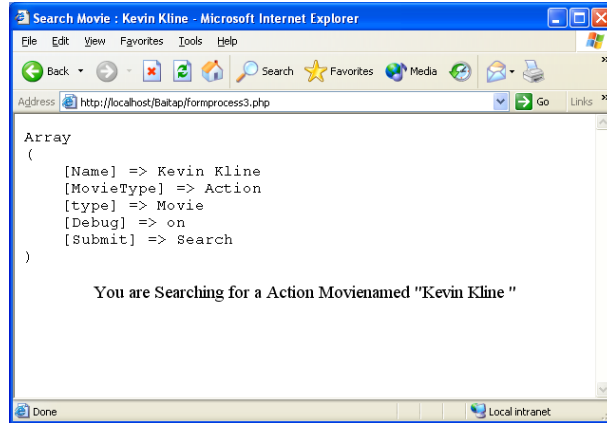
Hình 5.2.2.1

Chú ý ở đây có 2 nút Search và nút Add

4. Nhập **Kevin Kline** trong trường Name.

5. Chọn loại film.

6. Xóa hiển thị Debug Dump checkbox nếu bạn thích; sau đó nhấn nút Search .Kết quả như sau:



Hình 5.2.2.2

7. Bây giờ hãy thử thay đổi dữ liệu xem kết quả thế nào

Cách thức hoạt động

Bạn viết mã lệnh một form đơn giản với hai hoạt động. Phụ thuộc vào nút bạn nhấn và dữ liệu bạn chọn để enter, mã này xuất ra những thông tin khác nhau.

Một nhóm những nút radio và nút submit được thêm vào. Chúng ta hãy nhìn vào hình **Hình 5.2.2.1**

Chương 5 : NHỮNG THÀNH PHẦN CỦA FORM: CHO PHÉP NGƯỜI DÙNG LÀM VIỆC VỚI DỮ LIỆU

Trong ứng dụng Web, người sử dụng nhập nội dung, nhấn submit để xử lý. Một tiến trình thì không được hoàn thành bởi lệnh PHP, vì thế mã lệnh yêu cầu phải “thông minh”.

HTML sẽ gửi đến một vị trí đặc biệt và xử lý vì khi bạn điền thông tin vào form, để biết một địa chỉ email, mail,... bạn cần một phương tiện để đọc nội dung.

Form trong HTML thì đơn giản hơn, nó chỉ vị trí và cách thức gửi như thế nào. Tại điểm này PHP được ứng dụng hơn. Tập lệnh PHP nhận dữ liệu từ form và sử dụng nó để hoàn thành hoạt động, như cập nhật những nội dung của cơ sở dữ liệu, gửi một email, định dạng dữ liệu và hơn thế nữa.

PHP sử dụng một tập đơn giản mà mạnh, một sự kết hợp, cung cấp phương tiện để làm mọi thứ ảo khi chúng ta cần.

Trong chương này bạn bắt đầu xây dựng một ứng dụng đơn giản cho phép bạn thêm, bớt, xóa thành phần của dữ liệu. Chúng ta sẽ được đến với PHP/MySQL như:

- Tạo những form sử dụng nút, hộp soạn thảo và những thành phần khác.
- Tạo tập lệnh PHP để xử lý những form HTML.
- Nhận dữ liệu từ hai biến chính là: `$_POST` và `$_GET`.
- Qua thông tin ẩn để xử lý tập lệnh form với những điều khiển form ẩn và một chuỗi truy vấn URL.

Những phần tử nhập nút Radio

Nút radio là phần tử rất đơn giản. Bằng việc mặc định, nếu không có nút radio đặc biệt như CHECKED thì không có việc chọn

mặc định. Nên nhớ, việc chọn giá trị mặc định là phần quan trọng của việc thiết lập form.

Người sử dụng thường để mặc định.

```
<input type="radio" name="type" value="Movie" checked>
```

Movie


```
<input type="radio" name="type" value="Actor">
```

Actor


```
<input type="radio" name="type" value="Director">
```

Director

Cho nhiều nút radio phức tạp để liên kết các nhóm form với nhau và để xử lý một phần tử form đơn giản, chúng cần tên như nhau và những giá trị khác nhau, một cách hoàn toàn rõ ràng. Trong đoạn mã có trước, tên luôn là chữ in. Điều này nói với trình duyệt web rằng chỉ có một nút được chọn.

Những nút submit phức tạp

Như với nút radio, nút submit cũng có tên giống nhau với giá trị khác nhau. Nhấn vào một trong những nút đơn giản.

```
<input type="submit" name="Submit" value="Search">
```

```
<input type="submit" name="Submit" value="Add">
```

Khi bạn thấy biên dịch trở ngại, nút submit gửi thông tin đến tập lệnh. Bạn có thể cập nhật giá trị nút submit thông qua mảng `$_POST['Submit']`.

Việc kiểm tra nhập cơ bản

Tập lệnh xử lý là gì? Có gì mới?

Đoạn mã dưới đây kiểm tra loại mục là Movie, nó kiểm tra người sử dụng đã chọn một loại giá trị movie từ danh sách chưa. Nếu có hoặc không, anh ta hoặc cô ta được gửi một địa chỉ mới đến trang form.

Việc kiểm tra đơn giản nếu với một và toán tử. (Trong cách nói đơn giản Monopoly, nếu loại mục là movie và không đặc biệt, bạn trở lại một hình vuông và không thu được \$200.)

```
if($_POST['type'] == "Movie" && $_POST['MovieType'] ==  
"")  
{  
    header("Location:form3.php");  
}
```

Phần đầu hàm cho phép bạn gửi đến phần đầu HTTP thô. Nó thì hữu ích cho vấn đề bảo mật địa chỉ và hạn chế cập nhật. Trong ví dụ này, nó gửi một địa chỉ người dùng mới đến trang lý thuyết.

Chú ý: Một lỗi chung với người bắt đầu sử dụng PHP là hiểu sai một sự thật rất đơn giản. Một lần gửi những phần đầu không thể gửi một lần nữa. Điều này có nghĩa là bất kì việc xuất nào, bất kì khoảng trắng nào, bất kì sự trình bày bảng nào đúng, trước khi gọi phần đầu hàm nó sẽ sinh ra một cảnh báo trong việc thực thi tập lệnh. Ở đây là một vài kiểu lỗi:

```
<?php
    header("Location:form3.php");
?>
```

Đoạn mã này sẽ thất bại. Dòng bắt đầu tập lệnh rỗng sẽ gửi đến phần đầu với một điều khiển trả về và một đường vận chuyển (phụ thuộc vào hệ điều hành).

```
<?php
    echo "foobar";
    header("Location:form3.php");
?>
```

Đoạn mã này sẽ thất bại. Hàm xuất ra sẽ gửi phần đầu với text "foobar"

Tiêu đề trang động

Mã này thì dễ hiểu hơn: Bạn không bắt đầu xuất ra sớm như bắt đầu thực thi tập lệnh PHP. Điều gì sẽ xảy ra khi điểm bắt đầu tập lệnh sẽ kiểm tra sự xâm nhập và kiểm tra ngữ cảnh. Trong ví dụ này, bạn không sắp xếp mã ngữ cảnh phức tạp, nhưng bạn cần đặt tiêu đề trang động sử dụng kiểu action và kiểu item, bạn sẽ sử dụng handle của trang.

```
$title = $_POST['Submit'] . " " .
$_POST['type'] . " : " .
$_POST['Name'];
```

```
?>
<html>
<head>
<title><?php echo $title; ?></title>
```

Điều khiển một chuỗi bằng tay như một mảng để đổi ký tự đầu tiên

Chuỗi ký tự đơn giản có thể cập nhật thông qua cú pháp đơn giản giống như việc cập nhật chỉ mục của mảng. Chỉ mục đặc biệt của ký tự muốn cập nhật và voilà! Để đổi một ký tự hoặc toàn bộ chuỗi, sử dụng hàm: `strtoupper()`:

```
$name = $_POST['Name'];
$name[0] = strtoupper( $name[0]);
```

Bạn có thể dùng hàm `ucfirst()`

Toán tử tam nguyên

Dòng dưới đây chứa sự so sánh phép toán bậc ba. Toán tử tam nguyên không chỉ sử dụng trong PHP mà những ngôn ngữ khác cũng sử dụng (như C).

```
<?php echo $_POST['Submit'] == "Search" ? "for " : ""; ?>
```

Việc này có thể so sánh như cấu trúc if-else,

```
[biểu thức]?[thực hiện nếu TRUE]: [thực hiện nếu FALSE];
```

Toán tử tam nguyên này thì được biết như mỗi nguy hiểm lâu dài. Việc sử dụng nó sẽ làm cho đoạn mã của bạn dễ đọc hơn và có thể gây ra lỗi trong thời gian dài.

Chương 5 : NHỮNG THÀNH PHẦN CỦA FORM: CHO PHÉP NGƯỜI DÙNG LÀM VIỆC VỚI DỮ LIỆU

Trong ứng dụng Web, người sử dụng nhập nội dung, nhấn submit để xử lý. Một tiến trình thì không được hoàn thành bởi lệnh PHP, vì thế mã lệnh yêu cầu phải “thông minh”.

HTML sẽ gửi đến một vị trí đặc biệt và xử lý vì khi bạn điền thông tin vào form, để biết một địa chỉ email, mail,... bạn cần một phương tiện để đọc nội dung.

Form trong HTML thì đơn giản hơn, nó chỉ vị trí và cách thức gửi như thế nào. Tại điểm này PHP được ứng dụng hơn. Tập lệnh PHP nhận dữ liệu từ form và sử dụng nó để hoàn thành hoạt động, như cập nhật những nội dung của cơ sở dữ liệu, gửi một email, định dạng dữ liệu và hơn thế nữa.

PHP sử dụng một tập đơn giản mà mạnh, một sự kết hợp, cung cấp phương tiện để làm mọi thứ ảo khi chúng ta cần.

Trong chương này bạn bắt đầu xây dựng một ứng dụng đơn giản cho phép bạn thêm, bớt, xóa thành phần của dữ liệu. Chúng ta sẽ được đến với PHP/MySQL như:

- Tạo những form sử dụng nút, hộp soạn thảo và những thành phần khác.
- Tạo tập lệnh PHP để xử lý những form HTML.
- Nhận dữ liệu từ hai biến chính là: `$_POST` và `$_GET`.
- Qua thông tin ẩn để xử lý tập lệnh form với những điều khiển form ẩn và một chuỗi truy vấn URL.

Kết hợp những phần tử của form

Bây giờ chúng ta tạo ra một khung cho một phim ứng dụng. Hệ thống sẽ thêm những mục mới hoặc tìm kiếm những cái có sẵn.

Ví dụ: Liên kết tất cả chúng lại với nhau

Trong ví dụ này, bạn sẽ tạo ra nhiều tập lệnh mới làm việc với nhau để tái tạo cho phép người dùng thêm thông tin vào dữ liệu.

1. Tạo một file tên form4.php

Nhập đoạn code như sau:

```
<?php
// Hiển thị thông tin biên dịch
function debugDisplay()
{
    ?>
    <pre>
    $_POST
    <?php
    print_r($_POST);
    ?>
    $_GET
    <?php
    print_r($_GET);
    ?>
    </pre>
    <?php
}
if (!isset($_GET['step']))
{
    require('startform.php');
}
else
{
    // Switch on search/add wizard step
    switch ($_GET['step'])
    {
        // #####
```

```

// Search/Add form
// #####
case "1":
$type = explode(":", $_POST['type']);
if ($_POST['Submit'] == "Add")
{
    require($_POST['Submit'] . $type[0] . '.php');
} else
{
    if ($_POST['type'] == "Movie:Movie" &&
        $_POST['MovieType'] == "")
    {
        header("Location:form4.php");
    }
    ?>
    <h1>Search Results</h1>
    <p>You are looking for a "<?php echo $type[1]; ?>"
    named
    "<?php echo $_POST['Name']; ?>"</p>
    <?php
}
if ($_POST['Debug'] == "on")
{
    debugDisplay();
}
break;
// #####
// Add Summary
// #####

```



```

case "2":
$type = explode(":", $_POST['type']);
?>
<h1>New <?php echo $type[1]; ?> : <?php echo
$_POST['Name']; ?></h1>
<?php
switch ($type[0])
{
    case "Movie":
        ?>
        <p>Released in <?php echo $_POST['MovieYear'];
        ?></p>
        <p><?php echo nl2br(stripslashes($_POST['Desc']));
        ?></p>
        <?php
        break;
    default:
        ?>
        <h2>Quick Bio</h2>
        <p><?php echo nl2br(stripslashes($_POST['Bio']));
        ?></p>
        <?php
        break;
}
break;
// #####
// Starting form
// #####
default:

```

```

        require('startform.php');
        break;
    }
}

```

?>

2. Tạo một file với tên startform.php và nhập mã sau

```

<html>
<head>
<title>Multipurpose Form</title>
<style type="text/css">
TD{color:#353535;font-family:verdana}
TH{color:#FFFFFF;font-family:verdana;background-
color:#336699}
</style>
</head>
<body>
<form action="form4.php?step=1" method="post">
<table border="0" width="750" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
<tr>
<td bgcolor="#FFFFFF" width="30%">Name</td>
<td bgcolor="#FFFFFF" width="70%">
<input type="TEXT" name="Name">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Item Type</td>
<td bgcolor="#FFFFFF">
<input type="radio" name="type" value="Movie:Movie" checked>

```

```

Movie<br>
<input type="radio" name="type" value="Person:Actor">
Actor<br>
<input type="radio" name="type" value="Person:Director">
Director<br>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie type (if applicable)</td>
<td bgcolor="#FFFFFF">
<select name="MovieType">
<option value="" selected>Movie type...</option>
<option value="Action">Action</option>
<option value="Drama">Drama</option>
<option value="Comedy">Comedy</option>
<option value="Sci-Fi">Sci-Fi</option>
<option value="War">War</option>
<option value="Other">Other...</option>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" width="50%">Display Debug Dump</td>
<td bgcolor="#FFFFFF" width="50%">
<input type="checkbox" name="Debug" checked>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" colspan=2 align="center">

```

```

<input type="submit" name="Submit" value="Search">
<input type="submit" name="Submit" value="Add">
</td>
</tr>
</table>
</form>
</body>
</html>

```

4. Tạo một file mới với tên AddMovie.php:

```

<?php
if ($_POST['type'] == "Movie:Movie" &&
$_POST['MovieType'] == "") {
header("Location:form4.php");
}
$title = $_POST['Submit'] . " " .
$_POST['type'] . " : " .
$_POST['Name'];
$name = $_POST['Name'];
$name[0] = strtoupper($name[0]);
?>
<html>
<head>
<title><?php echo $title; ?></title>
<style type="text/css">
TD{color:#353535;font-family:verdana}
TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
</style>
</head>
<body>

```

```

<form action="form4.php?step=2" method="post">
<input type="hidden" name="type" value="<?php echo $type[1]; ?>">
<input type="hidden" name="action"
value="<?php echo $_POST['Submit']; ?>">
<table border="0" width="750" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
<tr>
<td bgcolor="#FFFFFF" width="30%">Movie Name</td>
<td bgcolor="#FFFFFF" width="70%">
<?php echo $name; ?>
<input type="hidden" name="Name" value="<?php echo $name; ?>">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Type</td>
<td bgcolor="#FFFFFF">
<?php echo $_POST['MovieType']; ?><br>
<input type="hidden" name="type"
value="Movie: <?php echo $_POST['MovieType']; ?>">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Year</td>
<td bgcolor="#FFFFFF">
<select name="MovieYear">
<option value="" selected>Select a year...</option>
<?php
for ($year=date("Y"); $year >= 1970 ;$year--) {
?>

```

```

<option value="<?php echo $year; ?>"><?php
echo $year; ?></option>
<?php
}
?>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Description</td>
<td bgcolor="#FFFFFF">
<textarea name="Desc" rows="5" cols="60"></textarea>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" colspan="2" align="center">
<input type="submit" name="SUBMIT" value="Add">
</td>
</tr>
</table>
</form>
</body>
</html>

```

5. Tạo một file mới với tên AddPerson.php và nhập đoạn mã sau

```

<?php
$title = $_POST['Submit'] . " " .
$_POST['type'] . " : " .
$_POST['Name'];
$name = $_POST['Name'];

```

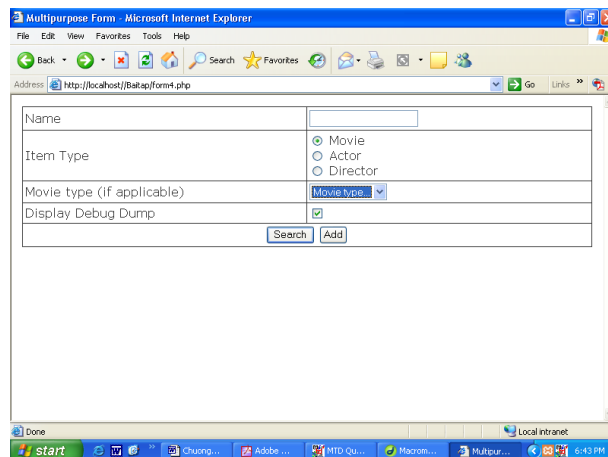
```
$name[0] = strtoupper($name[0]);
?>
<html>
<head>
<title><?php echo $title; ?></title>
<style type="text/css">
TD{color:#353535;font-family:verdana}
TH{color:#FFFFFF;font-family:verdana;background-
color:#336699}
</style>
</head>
<body>
<form action="form4.php?step=2" method="post">
<input type="hidden" name="type"
value="Person: <?php echo $type[1]; ?>">
<input type="hidden" name="action"
value="<?php echo $_POST['Submit']; ?>">
<table border="0" width="750" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
<tr>
<td bgcolor="#FFFFFF" width="30%">
<?php echo $type[1]; ?> Name
</td>
<td bgcolor="#FFFFFF" width="70%">
<?php echo $name?>
<input type="hidden" name="Name" value="<?php echo $name;
?>">
</td>
</tr>
```

```

<tr>
<td bgcolor="#FFFFFF">Quick Bio</td>
<td bgcolor="#FFFFFF">
<textarea name="Bio" rows="5" cols="60"></textarea>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" colspan="2" align="center">
<input type="submit" name="SUBMIT" value="Add">
</td>
</tr>
</table>
</form>
</body>
</html>

```

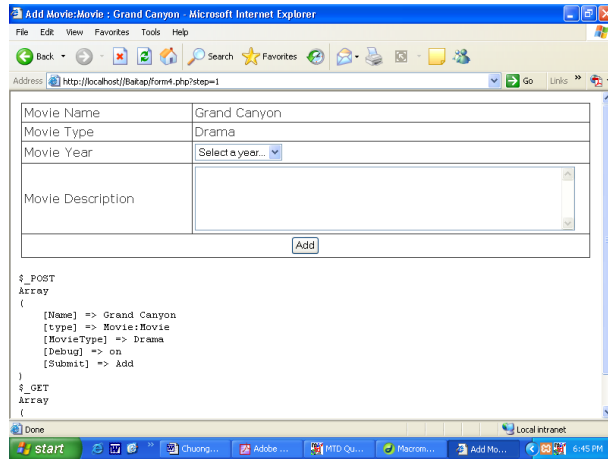
6. Cho chạy form4.php trên trình duyệt. Kết quả như sau:



Hình 5.3.1

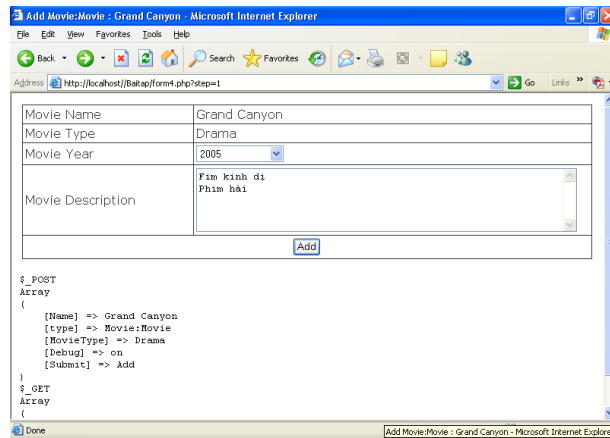
7. Nhập tên phim: “Grand Canyon.”

8. Click vào nút Add, cho bạn hình như sau:



Hình 5.3.2

9. Chọn một ngày trong năm sản xuất phim
10. Chọn Drama trong danh sách loại phim.
11. Hiện thị nhanh phim, nếu muốn nhập vào đâu, enter tại đó
12. Click vào nút Add và thấy thông tin hiển thị như sau:



Hình 5.3.3

Khi nhấn nút Add màn hình hiển thị:



Hình 5.3.4

Cách thức hoạt động

Tập lệnh này thiết kế xung quanh một ý tưởng đơn giản: một tập lệnh khung (form4.php) và nhiều tập lệnh **flesh-andmuscle** trên URL với chuỗi truy vấn.

Chương 5 : NHỮNG THÀNH PHẦN CỦA FORM: CHO PHÉP NGƯỜI DÙNG LÀM VIỆC VỚI DỮ LIỆU

Trong ứng dụng Web, người sử dụng nhập nội dung, nhấn submit để xử lý. Một tiến trình thì không được hoàn thành bởi lệnh PHP, vì thế mã lệnh yêu cầu phải “thông minh”.

HTML sẽ gửi đến một vị trí đặc biệt và xử lý vì khi bạn điền thông tin vào form, để biết một địa chỉ email, mail,... bạn cần một phương tiện để đọc nội dung.

Form trong HTML thì đơn giản hơn, nó chỉ vị trí và cách thức gửi như thế nào. Tại điểm này PHP được ứng dụng hơn. Tập lệnh PHP nhận dữ liệu từ form và sử dụng nó để hoàn thành hoạt động, như cập nhật những nội dung của cơ sở dữ liệu, gửi một email, định dạng dữ liệu và hơn thế nữa.

PHP sử dụng một tập đơn giản mà mạnh, một sự kết hợp, cung cấp phương tiện để làm mọi thứ ảo khi chúng ta cần.

Trong chương này bạn bắt đầu xây dựng một ứng dụng đơn giản cho phép bạn thêm, bớt, xóa thành phần của dữ liệu. Chúng ta sẽ được đến với PHP/MySQL như:

- Tạo những form sử dụng nút, hộp soạn thảo và những thành phần khác.
- Tạo tập lệnh PHP để xử lý những form HTML.
- Nhận dữ liệu từ hai biến chính là: `$_POST` và `$_GET`.

- Qua thông tin ẩn để xử lý tập lệnh form với những điều khiển form ẩn và một chuỗi truy vấn URL.

Tập lệnh khung

Khung ở đây là tập lệnh form4.php. Tất cả nó quay xung quanh việc sử dụng cấu trúc switch case. Nó bắt đầu với việc định nghĩa hàm cho việc hiển thị trình biên dịch(cái mà chứa hiển thị củamng toàn cục \$_GET).

Trong thực tế form sử dụng phương thức POST và vì vậy truyền thông tin đến mảng \$_POST, phần nội dung chuyển đổi của trang sẽ qua chuỗi truy vấn của mảng \$GET

Mỗi bước trong quá trình xây dựng thì được hướng dẫn bởi giá trị chỉ mục \$GET[‘bước’]. Nó chứa thông tin qua phần ?step=1 của URL.

Mỗi giá trị của tham số GET có một tập lệnh đặc biệt kèm theo. Tham số này cho tập lệnh chính(index.php) nơi mà một nhánh tiến trình dữ liệu được nhận.

Câu trả lời mặc định

Điều gì sẽ xảy ra khi gọi trang đầu tiên và bước tham số URL không được thiết lập? tập lệnh đánh giá điều kiện

switch và tìm các trường hợp. Nếu không có sẽ thực thi mặc định.

```
switch ($_GET['step']) {  
    ...  
    default:  
        require('startform.php');  
        break;  
}
```

Hàm `require()` lấy nội dung của file lý thuyết kể cả tập lệnh biên dịch cùng lúc. Những khác biệt của hàm `require()` với hàm `include()` là nó khởi tạo một lỗi lớn thay vì một cảnh báo nếu file không tìm thấy. Trong ví dụ này, tập lệnh `startform.php` sẽ ít quy tụ trong hàm, vì thế mà bạn muốn biết nếu không tìm thấy file.

Thêm mục con

Bạn cần hai form khác nhau để thêm vào mục con, vì thế cần nhánh thứ hai để quyết định form hiển thị.

Bây giờ chúng ta gặp phải một phần lệnh mà có một ít mưu mẹo. Danh sách giá trị mục con sử dụng lưu trữ 2 giá trị thay vì một. Cách này dùng dấu phân cách và đưa giá trị

explode vào mảng, cập nhật phần bạn cần.(Hàm explode() lấy mỗi đơn vị văn bản bằng dấu ngăn cách và chèn nó như phần tử mới của mảng.

Trong trường hợp bạn có 3 kiểu mục(Actors, Directors, and Movies), mỗi kiểu yêu cầu một form. Nhưng bạn muốn mục Actor và mục Director cùng thông tin .

Do đó, bạn không cần hai form khác nhau. Bạn làm điều này bằng việc thêm vào một cấu trúc cây bên trên mục Person hay Movie. Dưới Person là Actor và Director . Toàn bộ điểm có thể sử dụng mức độ tên của hệ đăng cấp mới đến tên file và bạn có thể thêm một mức độ mới su đó

Trong startform.php bạn có:

```
<input type="radio" name="type" value="Person:Actor">
Actor<br>
```

Chú ý: phần giá trị của kiểu phần tử bao gồm 2 giá trị khác nhau ngăn cách bởi dấu chấm phẩy(;).

Trong form4.php bạn có:

```
...
$type = explode(":", $_POST['type']);
if ($_POST['Submit'] == "Add") {
```

```
require($_POST['Submit'] . $type[0] . '.php');  
}  
...
```

Trong tập lệnh này, bạn lấy lại giá trị kiểu phân tử sử dụng chỉ mục mảng `$_POST['kiểu']` và sau đó sử dụng hàm `explode()` trên nội dung của nó. Hàm `explode()` thì khá dễ sử dụng. Nó cần chuỗi giới hạn đặc biệt và chuỗi chứa văn bản để `explode`.

Ví dụ, bạn có “Person:Actor” như giá trị để `explode` và dấu hai chấm (:) như giới hạn. Kết quả của biến `$type` sẽ là một mảng chứa một phần của chuỗi cắt tại dấu(;) trong ví dụ Nếu bạn trình bày nó trong định dạng `print_r`, bạn có:

```
Array  
(  
  [0] => Person  
  [1] => Actor  
)
```

Mục tiêu của tên file đơn giản được hoàn thành. Bạn có hai tập lệnh `Add` và một tên:

```
AddPerson.php and AddMovie.php.  
require($_POST['Submit'] . $type[0] . '.php');
```

Dòng này tạo tên tự động.

Chương 6 : CHO PHÉP NGƯỜI SỬ DỤNG SOẠN THẢO CƠ SỞ DỮ LIỆU

Nhận dữ liệu từ cơ sở dữ liệu khi cung cấp một vài dữ liệu cho cơ sở dữ liệu. Nhưng cơ sở dữ liệu thì không tạo ra nội dung mà chỉ lấy dữ liệu từ hệ thống khác. Điều này có nghĩa là bạn phải tạo hệ thống dữ liệu từ PHP.

Tất cả cơ sở dữ liệu tương tác dựa trên SQL. Bạn biết là cú pháp SQL lấy dữ liệu từ mảng.

Hầu hết mọi người dùng SQL để chèn dữ liệu mà PHP sửa đổi hoặc tạo ra.

Chương này đề cập đến soạn thảo cơ sở dữ liệu, bao gồm:

- Thêm: điều này thì đơn giản, nhưng bạn sẽ tìm để thêm vào trong một cơ sở dữ liệu
- Xóa: không sửa đổi lại cấu trúc cơ sở dữ liệu và có liên quan đến tính toàn vẹn.
- Sửa đổi: thay thế một vài vùng soạn thảo với nội dung mới trong một hồ sơ có sẵn.

Chuẩn bị Battlefield

Điều này đầy đủ và hoàn chỉnh như 1 đơn vị Vulcan, nhưng nếu muốn quản lý cơ sở dữ liệu thì biến toàn cục phải được tạo ra. Để lưu trong cùng một thời gian thì chúng ta sử dụng cơ sở dữ liệu hiện có để tránh bất kì vấn đề gì xảy ra khi làm bài tập. Tạo ra một cơ sở dữ liệu mới rỗng trong phpMyAdmin với tên moviesite. Trong cơ sở dữ liệu mới được tạo ra, thực thi tập lệnh chap6.mysql script, chứa cơ sở dữ liệu và dữ liệu của nó được bắt đầu.

Ví dụ:

Trước tiên, bạn cần một trang bắt đầu. Sau đây là những bước tạo

1. Tạo thư mục mới tên chap6 trong htdocs
2. Tạo 1 file index.php và nhập đoạn mã như sau

```
<?php
```

```
$link = mysql_connect("localhost", "root", "")  
or die("Could not connect: " . mysql_error());
```

```

mysql_select_db('moviesite', $link)
or die(mysql_error());
?>
<html>
<head>
<title>Movie database</title>
<style type="text/css">
TD{color:#353535;font-family:verdana}
TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
</style>
</head>
<body>
<table border="0" width="600" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
<tr>
<td bgcolor="#FFFFFF" colspan="2" align="center">
Movies <a href="movie.php?action=add&id=">[ADD]</a>
</td>
</tr>
<?php
$moviesql = "SELECT * FROM movie";
$result = mysql_query($moviesql)
or die("Invalid query: " . mysql_error());
while ($row = mysql_fetch_array($result)) {
?>
<tr>
<td bgcolor="#FFFFFF" width="50%">
<?php echo $row['movie_name']; ?>
</td>

```

```

        <td bgcolor=""#FFFFFF" width=""50%" align=""right">
        <a href=""movie.php?action=edit&id=<?php
        echo $row['movie_id']; ?>">[EDIT]</a>
        <a href=""delete.php?type=movie&id=<?php
        echo $row['movie_id']; ?>">[DELETE]</a>
        </td>
    </tr>
    <?php
    }
    ?>
    <tr>
        <td bgcolor=""#FFFFFF" colspan=""2" align=""center">
        People <a href=""people.php?action=add&id="">[ADD]</a>
        </td>
    </tr>
    <?php
        $moviesql = "SELECT * FROM people";
        $result = mysql_query($moviesql)
        or die("Invalid query: " . mysql_error());
        while ($row = mysql_fetch_array($result)) {
    ?>
    <tr>
        <td bgcolor=""#FFFFFF" width=""50%">
        <?php echo $row['people_fullname']; ?>
        </td>
        <td bgcolor=""#FFFFFF" width=""50%" align=""right">
        <a href=""people.php?action=edit&id=<?php
        echo $row['people_id']; ?>">[EDIT]</a>
        <a href=""delete.php?type=people&id=<?php

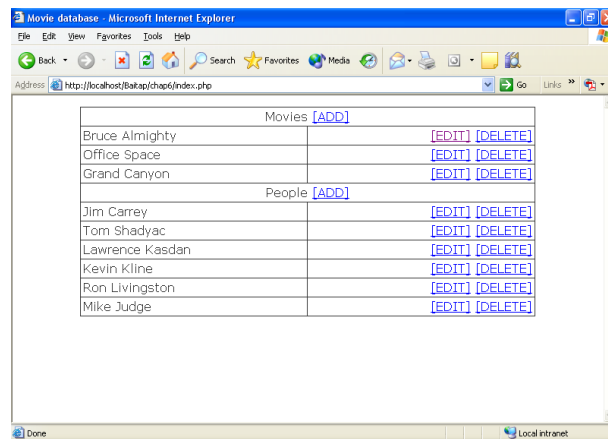
```

```

        echo $row['people_id']; ?>]>[DELETE]</a>
    </td>
</tr>
<?php
}
?>
</table>
</body>
</html>

```

3. Mở trình duyệt <http://localhost/chapter6/index.php> kết quả như sau:



Hình 6.1-1

Tất cả các liên kết tại thời điểm này đều không được thực hiện, chuyện này là bình thường vì chưa có các trang để liên kết.

Cách thức làm việc

Bạn phải có một giao diện quản lý trung tâm cho phép bạn thi hành những hoạt động trong cơ sở dữ liệu và dễ dàng thấy nội dung. Tập lệnh này là giao diện quản lý. Nó hiển thị và cho phép quản lý mọi thứ.

Chèn một mẫu tin đơn giản từ phpMyAdmin

Chú ý rằng tập lệnh sử dụng là một quy tắc đơn giản có liên quan đến SQL: Phải cố gắng truy vấn trong MySQL trước khi thử chèn nó trong đoạn mã. Lý do đơn giản là bạn có thể biên dịch ngôn ngữ tốt hơn trong mọi thời gian

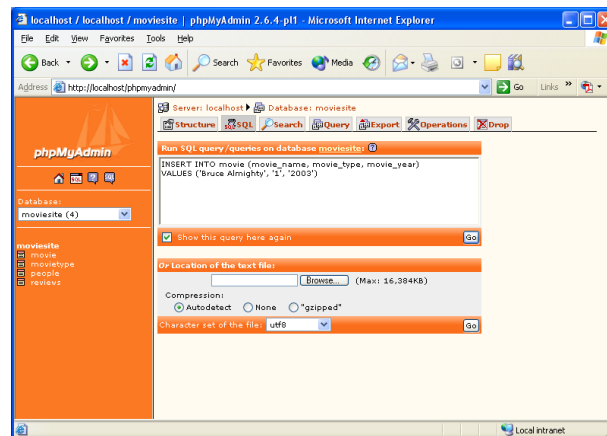
Ví dụ về chèn dữ liệu

Trong bài tập này bạn sẽ chèn một vài dữ liệu vào bảng

1. Mở cơ sở dữ liệu của bạn trong MyAdmin hoặc trong máy khách MySQL và nhập mã SQL sau:

```
INSERT INTO movie (movie_name, movie_type, movie_year)
VALUES ('Bruce Almighty', '1', '2003')
```

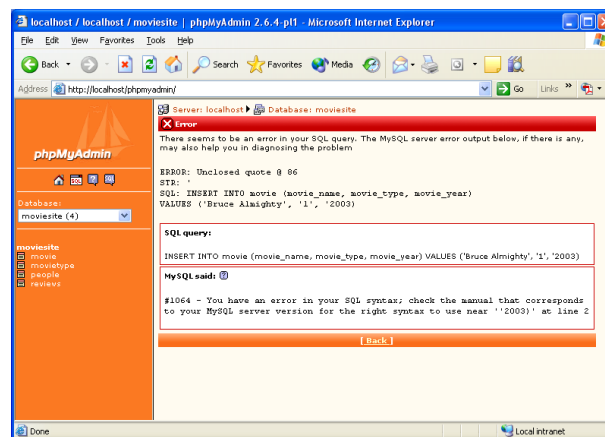
Như Hình 6.2.1



Hình 6.2.1

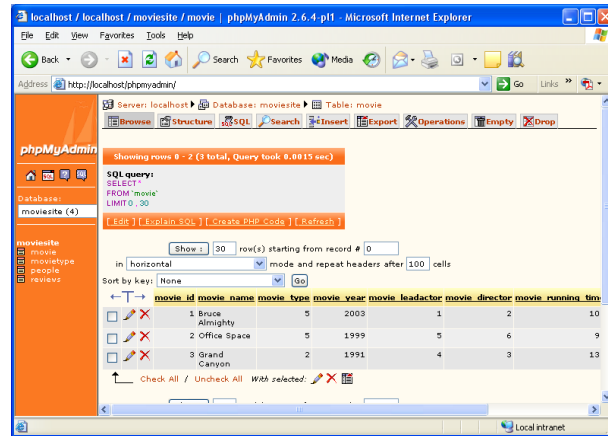
2. Thông báo sau xuất hiện như Hình 6.2.2

You have an error in your SQL syntax. Check the manual that corresponds to your MySQL server version for the right syntax to use near '2003)' at line 2



Hình 6.2.2

3. Đưa lỗi như đề nghị này (hoàn toàn đơn giản để làm với một số ký tự, phải nhảy đơn sau 2003) và chạy phpMyAdmin sau đó hiển thị thực thi SQL và hướng dẫn bạn về hiển thị nội dung của bảng như trong Hình 6.3.3



Hình 6.2.3

Trước khi truy vấn SQL trong PHP, bạn nên kiểm tra câu lệnh SQL trong phpMyAdmin. Nó có thể biên dịch SQL trước khi chèn vào đoạn mã và ngăn chặn việc biên dịch hai ngôn ngữ khác nhau trong cùng thời gian.

Cách thức làm việc

Khi chèn một mẫu tin trong bảng, không cần chèn ID nếu bạn đặt ở vùng khóa chính tự động giảm. SQL sẽ dễ dàng. Điều này chắc chắn rằng không có những khóa giống nhau trong bảng.

Để lấy id tự động của mẫu tin chèn trong PHP phải sử dụng hàm `mysql_insert_id()` sau đó gọi hàm `mysql_query()`. Hàm này trả về khóa chính khi chèn 1 mẫu tin mới.

Bởi vì bạn đã tạo truy vấn SQL nhiều hơn một dòng, bạn có thể thấy thông báo sau:

You have an error in your SQL syntax. Check the manual that corresponds to your MySQL server version for the right syntax to use near '2003)' at line 2

Dòng này tương ứng với một phần giá trị của câu lệnh SQL, như hiển thị ở đây: `VALUES ('Bruce Almighty', '1', '2003)`

Nếu truy vấn SQL ở trên một dòng thì chỉ có một thông báo “error in line 1”. Bạn có thể thấy điểm sai là bạn quên đóng dấu nháy trong giá trị movie_year value.

Bây giờ bạn có thể thấy chúng ta đã bỏ sót trong vùng movie. movie_id. Chúng ta làm điều này có mục đích. (Không có một sức mạnh đặc biệt của giá trị khóa chính nào trong phương tiện của MySQL để tự động quyết định giảm giá trị tự động). Bạn không cần biết khóa tiếp theo là gì

Chương 6 : CHO PHÉP NGƯỜI SỬ DỤNG SOẠN THẢO CƠ SỞ DỮ LIỆU

Nhận dữ liệu từ cơ sở dữ liệu khi cung cấp một vài dữ liệu cho cơ sở dữ liệu. Nhưng cơ sở dữ liệu thì không tạo ra nội dung mà chỉ lấy dữ liệu từ hệ thống khác. Điều này có nghĩa là bạn phải tạo hệ thống dữ liệu từ PHP.

Tất cả cơ sở dữ liệu tương tác dựa trên SQL. Bạn biết là cú pháp SQL lấy dữ liệu từ mảng.

Hầu hết mọi người dùng SQL để chèn dữ liệu mà PHP sửa đổi hoặc tạo ra.

Chương này đề cập đến soạn thảo cơ sở dữ liệu, bao gồm:

- Thêm: điều này thì đơn giản, nhưng bạn sẽ tìm để thêm vào trong một cơ sở dữ liệu
- Xóa: không sửa đổi lại cấu trúc cơ sở dữ liệu và có liên quan đến tính toàn vẹn.
- Sửa đổi: thay thế một vài vùng soạn thảo với nội dung mới trong một hồ sơ có sẵn.

Chèn một mẫu tin trong cơ sở dữ liệu quan hệ

Trong cơ sở dữ liệu thường có nhiều bảng. Tất cả các bảng này có thể hoàn toàn độc lập, nhưng sẽ giống như việc sử dụng xe hơi của bạn để lưu trữ một vài thứ trong thùng xe nhưng không bao giờ lái xe.

Trong hệ thống cũ của cơ sở dữ liệu quan hệ không tồn tại, mỗi dòng chứa tất cả các thông tin. Hệ thống đang chạy chỉ với một bảng chứa tất cả các thông tin. Bảng movie sẽ chứa tất cả các dữ liệu về diễn viên và đạo diễn và những loại khác trong phim. Giả sử rằng một ngày nào đó bạn quyết định một loại film nào đó sẽ được chuyển từ hành động sang thám hiểm (thay đổi mọi thứ). Bạn sẽ thông qua tất cả mẫu tin để thay đổi các nhãn film.

Trong hệ quản trị cơ sở dữ liệu mới(RDBMS), nó không hơn trường hợp nào, bạn sẽ tạo ra một bảng movietype lưu trữ tất cả các loại film được chuyển đến và bạn sẽ liên kết những loại film đó với loại film có liên quan.

Để liên kết những bảng đó, bạn sử dụng một loại khóa chính/khóa ngoại. Khóa chính của bảng movietype là một số xác định cho mỗi loại film. Ví dụ trong cơ sở dữ liệu của bạn id 1 có liên quan đến film hài. Khóa ngoại là quan hệ giữa bảng movie đến khóa chính của movietype.

Trong bài tập sau bạn sử dụng PHP và MySQL để chèn một bộ film vào cơ sở dữ liệu . Film này là một loại film từ bảng movietype.

Ví dụ : Chèn một bộ film với người và loại film đã biết.

Trong thời gian này chúng ta sẽ làm một vài thứ phức tạp hơn. Bạn sẽ thêm một bộ film vào hệ thống, trong khi ghi rõ có sẵn loại film, diễn viên và đạo diễn.

1. Tạo một file mới tên movie.php và nhập đoạn mã sau:

```
<?php
    $link = mysql_connect("localhost", "bp5am", "bp5ampass")
    or die("Could not connect: " . mysql_error());
    mysql_select_db('moviesite', $link)
    or die ( mysql_error());
    $peoplesql = "SELECT * FROM people";
    $result = mysql_query($peoplesql)
    or die("Invalid query: " . mysql_error());
    while ($row = mysql_fetch_array($result))
    {
        $people[$row['people_id']] = $row['people_fullname'];
    }
?>
<html>
<head>
<title>Add movie</title>
```

```

<style type="text/css">
TD{color:#353535;font-family:verdana}
TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
</style>
</head>
<body>
<form action="commit.php?action=add&type=movie" method="post">
<table border="0" width="750" cellspacing="1" cellpadding="3"
bgcolor="#353535" align="center">
<tr>
<td bgcolor="#FFFFFF" width="30%">Movie Name</td>
<td bgcolor="#FFFFFF" width="70%">
<input type="text" name="movie_name">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Type</td>
<td bgcolor="#FFFFFF">
<select id="game" name="movie_type" style="width:150px">
<?php
    $sql = "SELECT movietype_id, movietype_label " .
    "FROM movietype ORDER BY movietype_label";
    $result = mysql_query($sql)
    or die("<font color='\#FF0000\''>Query Error</font>") .
    mysql_error();
    while ($row = mysql_fetch_array($result)) {
    echo '<option value="' . $row['movietype_id'] . "'>' .
    $row['movietype_label'] . '</option>' . "\r\n";
    }

```

```

        ?>
    </select>
</td>
</tr>
<tr>
    <td bgcolor="#FFFFFF">Movie Year</td>
    <td bgcolor="#FFFFFF">
        <select name="movie_year">
            <option value="" selected>Select a year...</option>
            <?php
                for ($year = date("Y"); $year >= 1970; $year--) {
                    ?>
                    <option value="<?php echo $year; ?>"><?php
                        echo $year; ?></option>
                }
            ?>
        </select>
    </td>
</tr>
<tr>
    <td bgcolor="#FFFFFF">Lead Actor</td>
    <td bgcolor="#FFFFFF">
        <select name="movie_leadactor">
            <option value="" selected>Select an actor...</option>
            <?php
                foreach ($people as $people_id => $people_fullname) {
                    ?>
                    <option value="<?php echo $people_id; ?>" >

```

```

        <?php
            echo $people_fullname; ?></option>
        <?php
            }
        ?>
    </select>
</td>
</tr>
<tr>
    <td bgcolor=""#FFFFFF">Director</td>
    <td bgcolor=""#FFFFFF">
        <select name=""movie_director">
            <option value="" selected>Select a director...</option>
            <?php
                foreach ($people as $people_id => $people_fullname) {
                    ?>
                    <option value=""<?php echo $people_id; ?>" >
                        <?php
                            echo $people_fullname; ?></option>
                        <?php
                            }
                    ?>
                </select>
            </td>
        </tr>
        <tr>
            <td bgcolor=""#FFFFFF" colspan=""2" align=""center">
                <input type=""submit" name=""SUBMIT" value=""Add">
            </td>

```

```
</tr>
</table>
</form>
</body>
</html>
```

2. Lưu file và đưa lên server trong thư mục chap6.

3. Tạo một file mới tên commit.php và nhập đoạn mã:

```
<?php
// COMMIT ADD
$link = mysql_connect("localhost", "bp5am", "bp5ampass")
or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
or die ( mysql_error());
switch ($_GET['action'])
{
case "add":
switch ($_GET['type'])
{
case "movie":
$sql = "INSERT INTO movie
(movie_name,
movie_year,
movie_type,
movie_leadactor,
movie_director)
VALUES
('" . $_POST['movie_name'] . "',
'" . $_POST['movie_year'] . "',
'" . $_POST['movie_type'] . "',
```

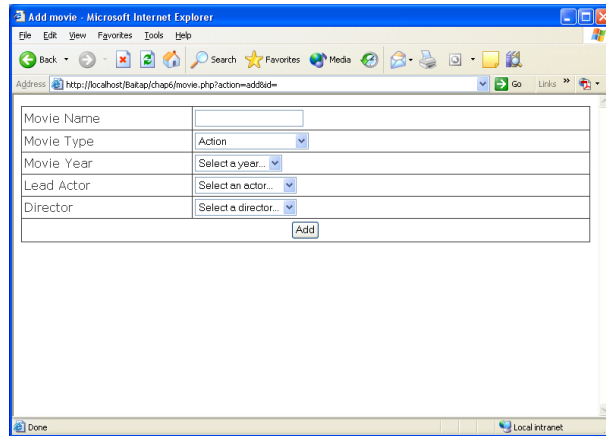
```

    “ . $_POST[‘movie_leadactor’] . “ ,
    “ . $_POST[‘movie_director’] . “ ”);
break;
}
break;
}
if (isset($sql) && !empty($sql))
{
echo “<!--” . $sql . “-->”;
$result = mysql_query($sql)
or die(“Invalid query: “ . mysql_error());
?>
<p align=“center” style=“color:#FF0000”>
Done. <a href=“index.php”>Index</a>
</p>
<?php
    }
?>

```

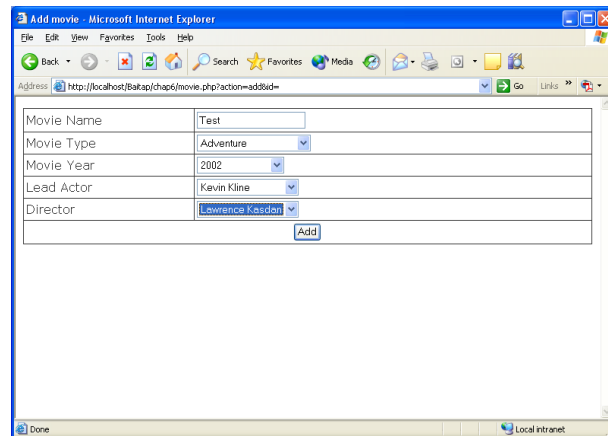
4. Lưu file vào thư mục chap6.

5. Mở trình duyệt trang index.php và click ADD tiếp theo phần đầu của bảng movie. Kết quả như **Hình 6.3.1**



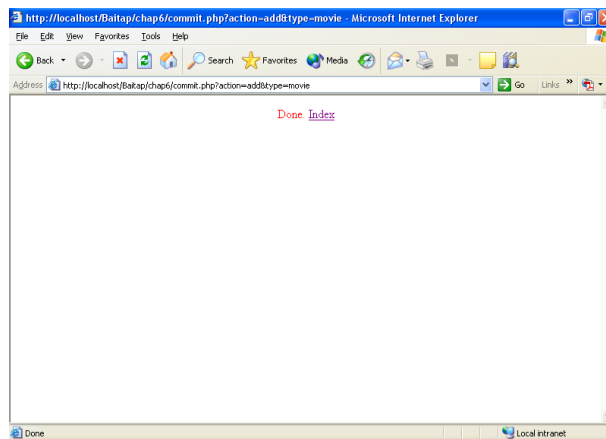
Hình 6.3.1

6. Thêm 1 movie named “Test” với kiểu tự do, diễn viên, và đạo diễn như Hình 6.3.2



Hình 6.3.2

7. Click vào nút “add” bạn sẽ thấy như Hình 6.3.3



Hình 6.3.3

Cách thức làm việc

Form HTML cho phép bạn điều khiển cách người dùng nhập dữ liệu. Một nút submit, form gửi thông tin đến server mà PHP có thể sử dụng chung và chạy câu lệnh SQL INSERT.

Như bạn thấy trong việc chèn film trong movie.php, bạn có 4 combo boxes và một vùng Text. Nội dung vùng text được nhập tự do, nhưng combobox thì hoàn toàn theo chỉ dẫn. Chúng ta hãy xem lại nội dung của combobox từ nội dung cơ sở dữ liệu.

Trước tiên, chúng ta hãy thử trong combobox. Mỗi danh sách combobox biểu diễn một người.

```
<?php
$link = mysql_connect("localhost", "root", "")
    or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
    or die ( mysql_error());
$peoplesql = "SELECT * FROM people";
$result = mysql_query($peoplesql)
    or die("Invalid query: " . mysql_error());
while ($row = mysql_fetch_array($result))
{
    $people[$row['people_id']] = $row['people_fullname'];
}
?>
```

Vào lúc bắt đầu tập lệnh, bạn truy vấn bảng người và đặt nội dung của nó trong mảng. Dữ liệu người lưu trữ trong bảng người. Danh sách chung của người, bạn truy vấn cơ sở dữ liệu đơn giản, lấy lại tất cả những gì liên quan đến người trong hệ thống và hiển thị tên trong combobox và chuyển đến khóa chính như giá trị mục.

Mỗi người được biết sẽ có một mục trong combobox.

```
<select name="movie_director">
    <option value="" selected>Select a director...</option>
<?php
```



```

        foreach ($people as $people_id => $people_fullname)
        {
?>
        <option value="<?php echo $people_id; ?>" ><?php
        echo $people_fullname; ?></option>
        <?php
        }
?>
</select>

```

Ở đây bạn sử dụng cú pháp foreach để đi đến tất cả các chọn lựa trong mảng. Loại phim tổng quan trong combobox, điều này thuận tiện hơn cho việc sử dụng nội dung SQL chung. Bạn dùng lại đoạn mã này để tạo ra một form có đặc điểm chung để thêm và chỉnh sửa, vì thế bạn cần phải hiểu chi tiết về việc này.

```

<select id="game" name="movie_type" style="width:150px">
<?php
        $sql = "SELECT movietype_id, movietype_label " .
        "FROM movietype ORDER BY movietype_label";
        $result = mysql_query($sql)
        or die("<font color='\"#FF0000\"'>Query Error</font>") .
        mysql_error());
        while ($row = mysql_fetch_array($result))
        {
                echo '<option value="' . $row['movietype_id'] . "' >' .
                $row['movietype_label'] . '</option>' . "\r\n";
        }
?>
</select>

```

Đoạn mã này tạo ra những lựa chọn trong combobox bằng bảng truy vấn movietype để trích ra các loại phim có sẵn.

Mỗi lựa chọn sẽ có một kiểu id như một giá trị và loại phim mà tự nó là một nhãn.

Form đã sẵn sàng, bạn cần có một tập lệnh để sử dụng cơ sở dữ liệu tạo ra các mẫu tin. Như bạn có thể thấy switch case on `$_GET['action']` thì lúc này hoàn toàn không sử dụng. Trong bài tập tiếp theo bạn thêm vào nhiều đoạn mã trong tập lệnh `movie.php`, vì vậy bạn có thể sử dụng nó để chỉnh sửa phim .

Chương 6 : CHO PHÉP NGƯỜI SỬ DỤNG SOẠN THẢO CƠ SỞ DỮ LIỆU

Nhận dữ liệu từ cơ sở dữ liệu khi cung cấp một vài dữ liệu cho cơ sở dữ liệu. Nhưng cơ sở dữ liệu thì không tạo ra nội dung mà chỉ lấy dữ liệu từ hệ thống khác. Điều này có nghĩa là bạn phải tạo hệ thống dữ liệu từ PHP.

Tất cả cơ sở dữ liệu tương tác dựa trên SQL. Bạn biết là cú pháp SQL lấy dữ liệu từ mảng.

Hầu hết mọi người dùng SQL để chèn dữ liệu mà PHP sửa đổi hoặc tạo ra.

Chương này đề cập đến soạn thảo cơ sở dữ liệu, bao gồm:

- Thêm: điều này thì đơn giản, nhưng bạn sẽ tìm để thêm vào trong một cơ sở dữ liệu
- Xóa: không sửa đổi lại cấu trúc cơ sở dữ liệu và có liên quan đến tính toàn vẹn.
- Sửa đổi: thay thế một vài vùng soạn thảo với nội dung mới trong một hồ sơ có sẵn.

Xóa một mẫu tin

Xóa một mẫu tin thì dễ dàng. Như đã đề cập từ trước, luôn phải kiểm tra truy vấn trong cơ sở dữ liệu thử nghiệm. Việc xóa một mẫu tin trong cơ sở dữ liệu thử nghiệm không bao giờ là mối đe dọa cho hệ thống của bạn. Và việc kiểm tra truy vấn giúp bạn tìm ra những lỗi SQL trước khi xóa tất cả các mẫu tin trong cơ sở dữ liệu bởi vì bạn quên một vài thứ như câu lệnh WHERE. MySQL xóa những thứ thỏa câu lệnh SQL.

Xóa có nghĩa là mất dữ liệu. Để xóa một mẫu tin bạn cần biết vị trí của mẫu tin trong cơ sở dữ liệu thông qua một tập hợp các câu lệnh điều kiện WHERE. Mỗi lần câu lệnh được thực thi mẫu tin sẽ bị xóa, không thể quay lại. Mẫu tin bị xóa không hy vọng trả lại. Đây là điều cần lưu ý khi sử dụng câu lệnh DELETE.

Ví dụ: Xóa mẫu tin đơn giản

Trước khi bảo PHP xóa mọi thứ, bạn nên thử xóa một mẫu tin từ phpMyAdmin với câu lệnh DELETE.

1. Mở phpMyAdmin và nhập đoạn mã sau:

```
DELETE FROM movie
WHERE movie_id = 12
LIMIT 1
```

2. phpMyAdmin trả lại thông báo rằng bạn xóa một mẫu tin từ bảng

Cách thức hoạt động

Câu lệnh DELETE SQL sử dụng rất đơn giản, bạn sử dụng câu lệnh LIMIT 1 để giới hạn xóa một mẫu tin.

Một cơ sở dữ liệu thường chứa những mẫu tin liên quan với nhau trong những bảng khác nhau. Xóa một vài mẫu tin không xem xét quan hệ của chúng, làm cho bạn khó hiểu.

Không may là MySQL không duy trì tính toàn vẹn.

Để tránh vấn đề này bạn có thể dùng form phức tạp hơn của câu lệnh DELETE, *Cascade Delete*, như đề cập trong phần sau

Ví dụ: *Cascade Delete*

Bây giờ bạn đã biết cách sử dụng DELETE, bạn sẽ thêm nó vào hệ thống để xóa thông tin một người từ hệ thống. Khi bạn lưu trữ quan hệ một người trong bảng movie, bạn cần cập nhật nội dung của bảng, vì vậy bạn không cần xóa những gì liên quan tới người.

Xóa một người giống như cách mở khóa xe và mong đợi bãi giữ xe trống. Bạn cần để chắc chắn không liên quan đến việc xóa các mẫu tin còn lại.

Sau đây là những bước thực thi *Cascade Delete*:

1. Tạo một file mới với tên delete.php và nhập đoạn mã sau:

```
<?php
$link = mysql_connect("localhost", "bp5am", "bp5ampass")
or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
```

```

or die ( mysql_error());
// DELETE SCRIPT
if (isset($_GET['do']) || $_GET['do'] != 1) {
?>
<p align="center" style="color:#FF0000">
Are you sure you want to delete this
<?php
    echo $_GET['type']; ?><br>
    <a href="<?php echo $_SERVER['REQUEST_URI'];
?>&do=1">yes</a>
or <a href="index.php">Index</a>
</p>
<?php
} else
{
if ($_GET['type'] == "people") {
    // Xóa những gì liên quan đến người trong bảng movie
    // Xóa những gì liên quan đến lãnh đạo diễn viên
    $actor = "UPDATE movie
SET movie_leadactor = '0'
WHERE movie_leadactor = '" . $_GET['id'] . "'";
$result = mysql_query($actor)
or die("Invalid query: " . mysql_error());
    // Xóa những gì liên quan đến đạo diễn
    $director = "UPDATE movie
SET movie_director = '0'
WHERE movie_director = '" . $_GET['id'] . "'";
$result = mysql_query($director)
or die("Invalid query: " . mysql_error());

```

```

}

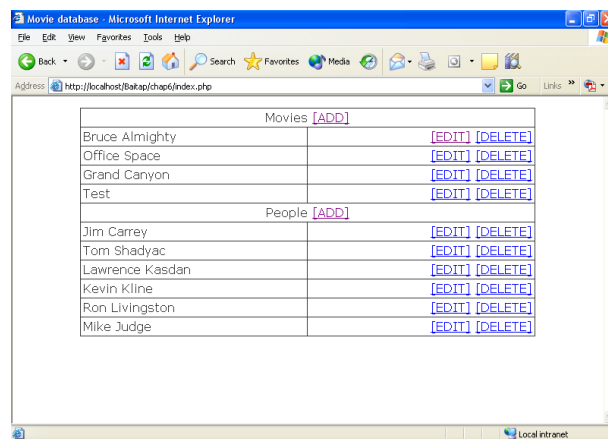
// SQL chung
$sql = "DELETE FROM " . $_GET['type'] . "
WHERE " . $_GET['type'] . "_id = " . $_GET['id'] . "
LIMIT 1";
// xuất SQL cho mục đích
echo "<!--" . $sql . "-->";
$result = mysql_query($sql)
or die("Invalid query: " . mysql_error());
?>

<p align="center" style="color:#FF0000">
Your <?php echo $_GET['type']; ?> has been deleted.
<a href="index.php">Index</a>
</p>
<?php
}
?>

```

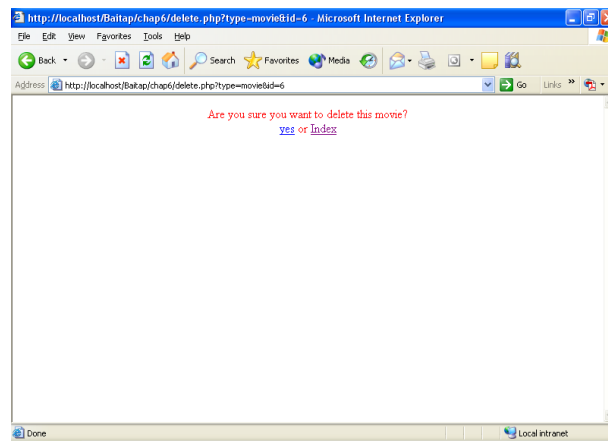
2. Lưu delete.php vào thư mục chap6.

3. Mở index.php trên trình duyệt. bạn sẽ thấy DELETE những liên kết tiếp sau mỗi film hoặc người như **Hình 6.4.1**



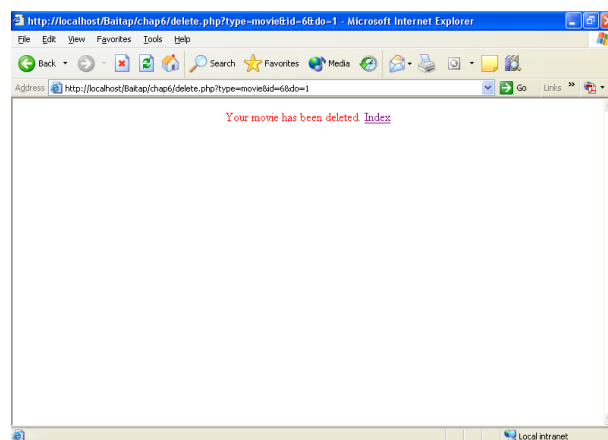
Hình 6.4.1

4. Thử xóa film mà bạn đã thêm vào ở bài tập trước bởi việc click vào liên kết DELETE trên hàng “Test” movie name. Bạn sẽ thấy như **Hình 6.4.2** .



Hình 6.4.2

5. Click vào liên kết “yes” để xác nhận việc xóa và đợi thông báo xác nhận như **Hình 6.4.3**



Hình 6.4.3

Cách thức hoạt động

Ở đây là kế hoạch tiêu diệt 1 tập hợp vô tội của dữ liệu. Đặt nó sang một bên, chúng ta hãy xem tập lệnh này hoạt động như thế nào.

Đầu tiên, bạn cần hiểu rằng bạn không thể xóa mẫu tin trong cơ sở dữ liệu và phải quên chúng đi. Việc xóa được xem xét cẩn thận. Ví dụ nếu xóa một người từ bảng people, điều này ngăn cản bạn từ việc tìm ra một khả năng có liên quan đến người trong bảng movie. Nếu bạn xóa Jim Carrey từ bảng people, ai sẽ lãnh đạo diễn viên *Bruce Almighty*? Nếu bạn không làm mọi thứ, id của Jim Carrey’s sẽ còn lại trong mẫu tin và bạn

sẽ có một cơ sở dữ liệu bị sửa đổi. Bạn không muốn điều này đúng không (câu trả lời là không)

Để giải quyết vấn đề này thì để chắc rằng bạn luôn có một khóa ngoại trong mẫu tin. Trong đoạn mã sau bạn cập nhật bảng film với giá trị 0 trước khi xóa mẫu tin people. Điều này cũng cho phép bạn kiểm tra cách hoạt động của câu lệnh UPDATE SQL.

```
// xóa phần liên quan đến lãnh đạo diễn viên
$actor = "UPDATE movie
SET movie_leadactor = '0'
WHERE movie_leadactor = '" . $_GET['id'] . "'";
$result = mysql_query($actor)
or die("Invalid query: " . mysql_error());
// xóa phần liên quan đến đạo diễn
$director = "UPDATE movie
SET movie_director = '0'
WHERE movie_director = '" . $_GET['id'] . "'";
$result = mysql_query($director)
or die("Invalid query: " . mysql_error());
```

Trong đoạn mã trước, đặt bất kỳ một vùng nào trong bảng movie mà có thể chứa một điều rủi ro cho bạn, khi bạn xóa một người. Câu lệnh UPDATE làm việc một cách đơn giản. Nó đặt những vùng lý thuyết với những giá trị lý thuyết mới trong tất cả các mẫu tin theo nhu cầu của câu lệnh WHERE.

Bạn có thể hỏi điều gì sẽ xảy ra nếu một người quên đi phần WHERE. Điều này sẽ cập nhật tất cả những mẫu tin trong bảng, mà có thể không có một thứ gì bạn muốn làm trong thực tế .

Thứ tự xóa

```
// phát sinh SQL
$sql = "DELETE FROM " . $_GET['type'] . "
WHERE " . $_GET['type'] . "_id = '" . $_GET['id'] . "'
LIMIT 1";
```



```
// Xuất SQL cho mục đích biên dịch
echo "<!--" . $sql . "-->";
$result = mysql_query($sql)
or die("Invalid query: " . mysql_error());
```

Truy vấn DELETE là một đơn vị động, nhưng nó khá dễ hiểu. Bạn không muốn mã hóa câu lệnh SQL cho mỗi loại. Vì vậy bạn sử dụng thông tin thông qua việc truyền biến trong URL để phát sinh ra câu lệnh SQL. Bảng và vùng khóa chính được phát sinh linh hoạt từ loại mục để xóa.

Chương 6 : CHO PHÉP NGƯỜI SỬ DỤNG SOẠN THẢO CƠ SỞ DỮ LIỆU

Nhận dữ liệu từ cơ sở dữ liệu khi cung cấp một vài dữ liệu cho cơ sở dữ liệu. Nhưng cơ sở dữ liệu thì không tạo ra nội dung mà chỉ lấy dữ liệu từ hệ thống khác. Điều này có nghĩa là bạn phải tạo hệ thống dữ liệu từ PHP.

Tất cả cơ sở dữ liệu tương tác dựa trên SQL. Bạn biết là cú pháp SQL lấy dữ liệu từ mảng.

Hầu hết mọi người dùng SQL để chèn dữ liệu mà PHP sửa đổi hoặc tạo ra.

Chương này đề cập đến soạn thảo cơ sở dữ liệu, bao gồm:

- Thêm: điều này thì đơn giản, nhưng bạn sẽ tìm để thêm vào trong một cơ sở dữ liệu
- Xóa: không sửa đổi lại cấu trúc cơ sở dữ liệu và có liên quan đến tính toàn vẹn.
- Sửa đổi: thay thế một vài vùng soạn thảo với nội dung mới trong một hồ sơ có sẵn.

Sửa dữ liệu trong mẫu tin

Có dữ liệu trong cơ sở dữ liệu là một việc tốt, nhưng dữ liệu có khả năng nhận rõ chính nó và có khuynh hướng cập nhật. Để cập nhật dữ liệu bạn cần nhận biết dữ liệu cập nhật và hệ thống người sử dụng hiện tại với giao diện tốt. Việc sử dụng giao diện tương tự như việc tạo dữ liệu thường là một thực tiễn tốt.

Ví dụ: Sửa đổi một Movie

Trong ví dụ này, bạn tạo ra một tập lệnh mà có thể sửa đổi một bộ phim. Bạn sẽ xây dựng trên movie.php .

1. Mở movie.php và nhập mã lệnh

```
<?php
$link = mysql_connect("localhost", "root", "")
    or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
    or die ( mysql_error());
```

```

$peoplesql = "SELECT * FROM people";
$result = mysql_query($peoplesql)
    or die("Invalid query: " . mysql_error());
while ($row = mysql_fetch_array($result)) {
    $people[$row['people_id']] = $row['people_fullname'];
}
switch ($_GET['action'])
{
    case "edit":
        $moviesql = "SELECT * FROM movie
            WHERE movie_id = " . $_GET['id'] . """;
        $result = mysql_query($moviesql)
            or die("Invalid query: " . mysql_error());
        $row = mysql_fetch_array($result);
        $movie_name = $row['movie_name'];
        $movie_type = $row['movie_type'];
        $movie_year = $row['movie_year'];
        $movie_leadactor = $row['movie_leadactor'];
        $movie_director = $row['movie_director'];
        break;
    default:
        $movie_name = "";
        $movie_type = "";
        $movie_year = "";
        $movie_leadactor = "";
        $movie_director = "";
        break;
}
?>

```

```

<html>
<head>
    <title><?php echo $_GET['action']; ?> movie</title>
    <style type="text/css">
        TD{color:#353535;font-family:verdana}
        TH{color:#FFFFFF;font-family:verdana;background-color:#336699}
    </style>
</head>
<body>
    <form action="commit.php?action=<?php
    echo $_GET['action']; ?>&type=movie&id=<?php
    echo $_GET['id']; ?>" method="post">
    <table border="0" width="750" cellspacing="1" cellpadding="3"
    bgcolor="#353535" align="center">
    <tr>
        <td bgcolor="#FFFFFF" width="30%">Movie Name</td>
        <td bgcolor="#FFFFFF" width="70%">
            <input type="text" name="movie_name"
            value="<?php echo $movie_name; ?>">
        </td>
    </tr>
    <tr>
        <td bgcolor="#FFFFFF">Movie Type</td>
        <td bgcolor="#FFFFFF">
            <select          id="game"          name="movie_type"
            style="width:150px">
            <?php
            $sql = "SELECT movietype_id, movietype_label " .
            "FROM movietype ORDER BY movietype_label";

```

```

        $result = mysql_query($sql)
        or die("<font color='\">#FF0000\`>Query Error</font>".
        mysql_error());
        while ($row = mysql_fetch_array($result))
        {
            if ($row['movietype_id'] == $movie_type)
            {
                $selected = " selected";
            }
            else {
                $selected = "";
            }
            echo '<option value="' . $row['movietype_id'] . "' .
            $selected.'>' . $row['movietype_label'] . '</option>' .
            "\r\n";
        }
    ?>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Year</td>
<td bgcolor="#FFFFFF">
<select name="movie_year">
<option value="" selected>Select a year...</option>
<?php
    for ($year = date("Y"); $year >= 1970; $year--)
    {
        if ($year == $movie_year)

```

```

        {
            $selected = " selected";
        } else {
            $selected = "";
        }
    ?>
    <option value="<?php echo $year; ?>"<?php
        echo $selected; ?>><?php echo $year; ?></option>
    <?php
        }
    ?>
</select>
    </td>
</tr>
<tr>
    <td bgcolor="#FFFFFF">Lead Actor</td>
    <td bgcolor="#FFFFFF">
        <select name="movie_leadactor">
            <option value="" selected>Select an actor...</option>
        <?php
            foreach ($people as $people_id => $people_fullname) {
                if ($people_id == $movie_leadactor)
                {
                    $selected = " selected";
                } else {
                    $selected = "";
                }
            }
        ?>
        <option value="<?php echo $people_id; ?>"<?php

```

```

        echo $selected; ?><?php echo $people_fullname;
        ?></option>
    <?php
        }
    ?>
</select>
</td>
</tr>
<tr>
    <td bgcolor="#FFFFFF">Director</td>
    <td bgcolor="#FFFFFF">
        <select name="movie_director">
            <option value="" selected>Select a director...</option>
            <?php
                foreach ($people as $people_id => $people_fullname) {
                    if ($people_id == $movie_director) {
                        $selected = " selected";
                    } else {
                        $selected = "";
                    }
                }
            ?>
            <option value="<?php echo $people_id; ?>"<?php
                echo $selected; ?><?php echo $people_fullname;
                ?></option>
            <?php
                }
            ?>
        </select>
    </td>

```

```

</tr>
<tr>
  <td bgcolor="#FFFFFF" colspan="2" align="center">
    <input type="submit" name="SUBMIT" value="<?php
    echo $_GET['action']; ?>">
  </td>
</tr>
</table>
</form>
</body>
</html>

```

2. Mở tập lệnh commit.php và thay đổi nội dung như sau:

```

<?php
// COMMIT ADD AND EDITS
$link = mysql_connect("localhost", "bp5am", "bp5ampass")
    or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
    or die ( mysql_error());
switch ($_GET['action'])
{
case "edit":
    switch ($_GET['type'])
    {
        case "movie":
            $sql = "UPDATE movie SET
            movie_name = " . $_POST['movie_name'] . ",
            movie_year = " . $_POST['movie_year'] . ",
            movie_type = " . $_POST['movie_type'] . ",
            movie_leadactor = " . $_POST['movie_leadactor'] . ",

```



```

        movie_director = “. $_POST[‘movie_director’] . “
        WHERE movie_id = “. $_GET[‘id’] . “”;
        break;
    }
    break;
case “add”:
    switch ($_GET[‘type’])
    {
        case “movie”:
            $sql = “INSERT INTO movie
            (movie_name,
            movie_year,
            movie_type,
            movie_leadactor,
            movie_director)
            VALUES
            (“. $_POST[‘movie_name’] . “,
            “. $_POST[‘movie_year’] . “,
            “. $_POST[‘movie_type’] . “,
            “. $_POST[‘movie_leadactor’] . “,
            “. $_POST[‘movie_director’] . “)”;
            break;
        }
    break;
}
if (isset($sql) && !empty($sql))
{
    echo “<!--” . $sql . “-->”;
    $result = mysql_query($sql)

```

```
or die("Invalid query: " . mysql_error());
```

```
?>
```

```
<p align="center" style="color:#FF0000">
```

```
Done. <a href="index.php">Index</a>
```

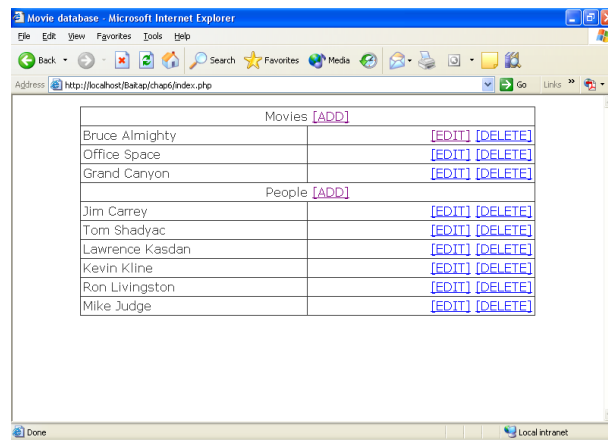
```
</p>
```

```
<?php
```

```
}
```

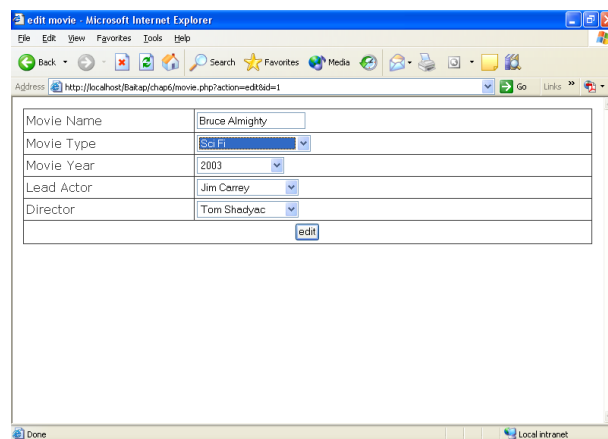
```
?>
```

3. Bây giờ mở <http://localhost/chapter6/index.php> hiển thị như **Hình 6.6.1**



Hình 6.6.1

4. Thử click liên kết EDIT ở “Bruce Almighty” movie, thay đổi một vài thứ trong hộp và tên film, ấn nút “edit” sẽ hiển thị như **Hình 6.6.2**



Hình 6.6.2

5. Edit “Bruce Almighty” nhận lại với thủ tục trong bước 4, và ấn định lại như cũ.

Bây giờ liên kết EDIT cho film, bạn thấy rằng tập lệnh lưu trữ giá trị và cho phép sửa đổi dữ liệu dễ dàng

Cách thức làm việc

Đoạn mã commit.php giống với những gì bạn thấy. Nhưng có một phát triển cần quan tâm trong movie.php. Chúng ta hãy nhìn vào một vài chi tiết

Đầu tiên hãy nhìn vào switch ở phần đầu của tập lệnh. Bạn định nghĩa một switch trên một chuỗi truy vấn tham số tên action. Nếu action là edit, bạn truy vấn cơ sở dữ liệu cho một mẫu tin tương ứng đến id lý thuyết trong chuỗi truy vấn tham số id và đặt một vài biến. Những biến này được đặt để tránh action không là edit..

```
switch ($_GET['action'])
{
case “edit”:
    $moviesql = “SELECT * FROM movie
    WHERE movie_id = “ . $_GET['id'] . “”;
    $result = mysql_query($moviesql)
        or die(“Invalid query: “ . mysql_error());
    $row = mysql_fetch_array($result);
    $movie_name = $row['movie_name'];
    $movie_type = $row['movie_type'];
    $movie_year = $row['movie_year'];
    $movie_leadactor = $row['movie_leadactor'];
    $movie_director = $row['movie_director'];
break;
default:
    $movie_name = “”;
    $movie_type = “”;
    $movie_year = “”;
    $movie_leadactor = “”;
```

```
$movie_director = "";  
break;  
}  
?>
```

Những biến đặt trong đoạn mã trước thường đặt giá trị mặc định của vùng form. Mỗi vùng có một giá trị nếu như bạn chỉnh sửa một mẫu tin và có một giá trị trống nếu như bạn tạo một mẫu tin.

```
<tr>  
    <td bgcolor="#FFFFFF" width="30%">Movie Name</td>  
    <td bgcolor="#FFFFFF" width="70%">  
        <input type="text" name="movie_name"  
        value="<?php echo $movie_name; ?>">  
    </td>  
</tr>
```

Trong ví dụ này, vùng movie_name lấy nội dung biến \$movie_name như một giá trị mặc. Điều này cho phép bạn nạp lại với mẫu tin đã được chỉnh sửa.

Việc sửa đổi văn bản thì dễ dàng. Sửa đổi giá trị trong một danh sách là một chuyện khác. Bạn không thể hiển thị danh sách và hi vọng người dùng sẽ đặt lại giá trị gốc khi anh ta hoặc cô ta sửa mẫu tin.

Bạn cần phải nạp lại hoàn toàn danh sách và đặt giá trị trước như mặc định trong danh sách, vì vậy người dùng có thể bỏ qua nó nếu không muốn thay đổi nó.

Làm điều này như thế nào?

```
<tr>  
    <td bgcolor="#FFFFFF">Movie Type</td>  
    <td bgcolor="#FFFFFF">  
        <select id="game" name="movie_type" style="width:150px">  
<?php  
    $sql = "SELECT movietype_id, movietype_label "  
        "FROM movietype ORDER BY movietype_label";
```

```

$result = mysql_query($sql)
    or die("<font color='\"#FF0000\"'>Query Error</FONT>".
mysql_error());
while ($row = mysql_fetch_array($result))
{
    if ($row['movietype_id'] == $movie_type)
    {
        $selected = " selected";
    } else {
        $selected = "";
    }
    echo "<option value='\"' . $row['movietype_id'] . \"\" .
    $selected . '>' . $row['movietype_label'] . '</option>' .
    "\r\n";
}
?>
</select>
</td>
</tr>

```

Bạn nạp vào danh sách như bạn sẽ làm nếu bạn thêm một mẫu tin, nhưng bạn so sánh giá trị hiện hành với giá trị mặc định. Nếu chúng đúng thì thêm cờ đơn giản SELECTED để lựa chọn. Việc này đặt danh sách giá trị mặc định đến giá trị hiện hành trong bảng.

```

if ($row['movietype_id'] == $movie_type)
{
    $selected = " selected";
} else {
    $selected = "";
}

```


Chương 7 : THAO TÁC VÀ TẠO HÌNH ẢNH VỚI PHP

Chương này bao gồm những phần sau:

- Có thể tải đặt PHP bao gồm cả thư viện GD.
- Cho phép người dùng đưa hình ảnh của họ lên.
- Nhận lại thông tin về hình ảnh như loại, kích thước.
- Tạo một hình ảnh mới.
- Sao chép một ảnh hoặc một phần của ảnh.
- Tạo thumbnails(phiên bản thu nhỏ của ảnh).
- Tạo một phiên bản ảnh trắng đen.
- Thêm hoa văn và tên cho hình.

1.1. Làm việc với thư viện GD

GD được viết bằng C++ và cho phép thao tác với hình ảnh hiện tại. Bởi vì PHP không thể tự động xử lý hình ảnh với việc xây dựng hàm. Cần chắc chắn rằng bạn có thư viện GD. May thay, Trong tất cả các phiên bản của PHP gần đây có hỗ trợ một gói thư viện này. Nếu bạn không có gói này bạn có thể tìm nó tại <http://www.boutell.com/gd/>. Tuy nhiên chúng tôi khuyên rằng : bạn sử dụng gói này tương phản với những phiên bản có sẵn bên ngoài được tải về nếu có thể.

1.1.1. Tôi có thể sử dụng loại files gì với GD và PHP?

GD có thể làm việc với vô số hình ảnh, nhưng khi bạn sử dụng nó với PHP, bạn có thể định dạng hình ảnh với bất kỳ loại nào như GIF, JPG, PNG, SWF, SWC, PSD, TIFF, BMP, IFF, JP2, JPX, JB2, JPC, XBM hoặc WBMP.

Bạn có thể thao tác và tạo hình ảnh với định dạng GIF, JPG, PNG, WBMP, và XBM. GD cũng cho phép PHP tạo hình dạng như hình vuông, hình đa giác, hình elíp, tốt như những hộp văn bản sử dụng đúng kiểu Fonts.

Phụ thuộc vào phiên bản của GD, GIF có thể hoặc là không thể hỗ trợ. Nếu GIF có thể hỗ trợ với việc sử dụng hàm `gd_info` diễn tả trong phần ví dụ “kiểm tra GD”.

1.1.2. Biên dịch PHP với GD

Nếu bạn sử dụng một Web chủ, tình cờ chúng đã có sẵn GD trong cài đặt PHP. Nếu bạn chạy trong máy của bạn, có thể GD không sẵn sàng. Trong Window, dễ dàng tìm thấy dòng sau trong file php.ini;extension=php_gd2.dll

Không chú ý giống với dòng:extension=php_gd2.dll

Bạn cần khởi động lại Apache để làm cho thay đổi có hiệu lực.

Trong Linux, bạn cần chọn cấu hình --with-gd cho GD. Mặc khác, bởi vì gói phiên bản của GD yêu cầu dùng với PHP, bạn không cần nhận dạng thư mục cài đặt GD. Nó được mặc định

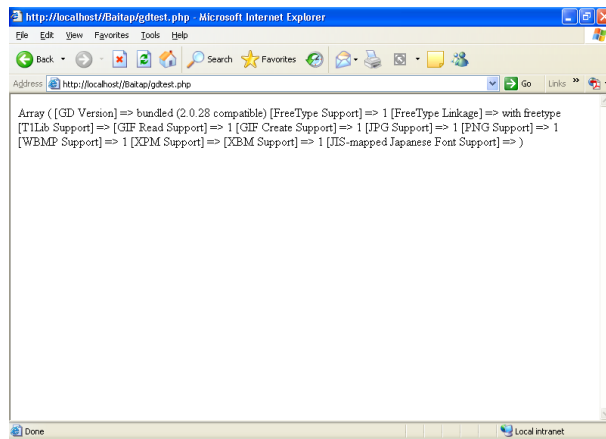
Ví dụ: Kiểm tra GD

Mở trình soạn thảo và nhập đoạn code

```
<?php
print_r(gd_info());
?>
```

Lưu với tên gdtest.php (đưa nó lên web server nếu cần).

Mở trình duyệt bạn sẽ thấy kết quả như sau:**Hình 7.1**



Hình 7.1

Cách thức hoạt động:

Hàm gd_info thì hoàn toàn hữu ích, bởi vì phiên bản GD được đóng gói với PHP. Mục đích của nó là đặt tất cả các thông tin về phiên bản GD vào trong một mảng mà bạn có thể xem lại sau đó. Việc này không chỉ đáp ứng như kiểm tra để chắc chắn rằng PHP

với GD thì vận hành tốt với mỗi thứ khác. Nhưng nó cho phép bạn thấy giới hạn của bạn cho việc sử dụng GD trong PHP. Để đáp ứng mục đích trong ví dụ của chương này bạn cần có hỗ trợ JPG,GIF và PNG. Nếu phiên bản của Gd không hỗ trợ bất cứ loại hình ảnh nào thì bạn cần nâng cấp. Bạn có thể tìm hướng dẫn nâng cấp đầy đủ và tập lệnh mã nguồn tại <http://www.boutell.com/gd>.

Hàm print_r() lấy tất cả các thông tin lưu trữ trong một biến(bao gồm mảng) và xuất nó lên trình duyệt, do đó bạn có thể thấy nó.

Bây giờ bạn biết GD thì làm việc tốt và loại hình ảnh nào sẽ được hỗ trợ

Cho phép người dùng đưa hình ảnh lên

PHP cho phép người dùng đưa hình ảnh lên server. Bạn cần sự giúp đỡ của MySQL để lưu trữ hình ảnh và tất cả các thông tin về hình ảnh.

Ví dụ: Tạo một bảng hình ảnh:

Trước tiên bạn cần tạo một bảng lưu trữ thông tin về những hình ảnh. Bạn sẽ lưu trữ thông tin cơ bản về mỗi hình như tên, tựa đề của hình. Sau đó, cho người sử dụng một form mà người dùng có thể submit một hình để hiển trên website. Bạn sẽ hỏi một vài thông tin cơ bản về hình. Sau đó bạn cho phép người sử dụng đưa file trực tiếp từ tiện nghi trình duyệt hiếm có của họ, không cần sự viện trợ của bất cứ phần mềm FPT nào.

Nếu bạn không có một thư mục cho hình ảnh nhà của bạn. Bạn cần tạo một thư mục. trong bài tập này, hình ảnh sẽ được lưu trữ.

Mở trình soạn thảo nhập đoạn mã sau:

```
<?php
//Kết nối cơ sở dữ liệu
$link = mysql_connect("localhost", "root", "")
or die("Could not connect: " . mysql_error());
mysql_select_db("moviesite", $link)
or die (mysql_error());
//Tạo bảng images
$sql = "CREATE TABLE IF NOT EXISTS images (
```

```

image_id INT(11) NOT NULL AUTO_INCREMENT,
image_caption VARCHAR(255) NOT NULL,
image_username VARCHAR(255) NOT NULL,
image_date DATE NOT NULL,
PRIMARY KEY (image_id)
)”;
$results = mysql_query($sql)
or die(mysql_error());
echo “Image table successfully created.”;
?>

```

Lưu với tên create_images_table.php. Mở trình duyệt và bạn sẽ thấy thông báo “Image table successfully created.”

Mở trình soạn thảo nhập đoạn mã sau:

```

<html>
<head>
<title>Upload your pic to our site!</title>
</head>
<body>
<form name="form1" method="post" action="check_image.php"
enctype="multipart/form-data">
<table border="0" cellpadding="5">
<tr>
<td>Image Title or Caption<br>
<em>Example: You talkin' to me?</em></td>
<td><input name="image_caption" type="text" id="item_caption"
size="55"
length="255"></td>
</tr>
<tr>

```

```

        <td>Your Username</td>
        <td><input          name="image_username"          type="text"
        id="image_username" size="15"
        maxlength="255"></td>
    </tr>
    <td>Upload Image:</td>
    <td><input          name="image_filename"          type="file"
    id="image_filename"></td>
</tr>
</table>
    <br>
    <em>Acceptable image formats include: GIF, JPG/JPEG, and
    PNG.</em>
    <p align="center"><input type="submit" name="Submit"
    value="Submit">
    &nbsp;
    <input type="reset" name="Submit2" value="Clear Form">
    </p>
</form>
</body>
</html>

```

Lưu file upload_image.htm. Trong ví dụ đơn giản này bạn không có một mã lệnh PHP nào trong form, vì vậy bạn không cần phần mở rộng là .php

Tạo một file mới và nhập đoạn mã sau:

```

<?php
    //Kết nối cơ sở dữ liệu
    $link = mysql_connect("localhost", "root", "")
    or die("Could not connect: " . mysql_error());
    mysql_select_db("moviesite", $link)

```

```

or die (mysql_error());
//làm việc với biến có sẵn
$image_caption = $_POST['image_caption'];
$image_username = $_POST['image_username'];
$image_tempname = $_FILES['image_filename']['name'];
$today = date("Y-m-d");
//Đưa hình ảnh lên và kiểm tra loại hình ảnh
//Thay đổi đường dẫn đến thư mục hình ảnh
$imageDir = "c:/Program Files/Apache
Group/Apache2/test/images/";
$imageName = $imageDir . $image_tempname;
if(move_uploaded_file($_FILES['image_filename']['tmp_name'],
$imageName))
{
    //lấy thông tin về hình ảnh đang đưa lên
    list($width, $height, $type, $attr) =
    getimagesize($imageName);
    switch ($type)
    {
    case 1:
        $ext = ".gif";
        break;
    case 2:
        $ext = ".jpg";
        break;
    case 3:
        $ext = ".png";
        break;
    default:

```

```

        echo "Sorry, but the file you uploaded was not a GIF, JPG, or
        “.
        “PNG file.<br>”;
        echo "Please hit your browser’s ‘back’ button and try again.”;
    }
//insert info into image table
$insert = “INSERT INTO images
(image_caption, image_username, image_date)
VALUES
(‘$image_caption’, ‘$image_username’, ‘$today’)”;
$insertresults = mysql_query($insert)
or die(mysql_error());
$lastpicid = mysql_insert_id();
$newfilename = $ImageDir . $lastpicid . $ext;
rename($ImageName, $newfilename);
}
?>
<html>
<head>
<title>Here is your pic!</title>
</head>
<body>
    <h1>So how does it feel to be famous?</h1><br><br>
    <p>Here is the picture you just uploaded to our servers:</p>
    <img src=”images/<?php echo $lastpicid . $ext; ?>” align=”left”>
    <strong><?php echo $image_name; ?></strong><br>
    This image is a <?php echo $ext; ?> image.<br>
    It is <?php echo $width; ?> pixels wide
    and <?php echo $height; ?> pixels high.<br>

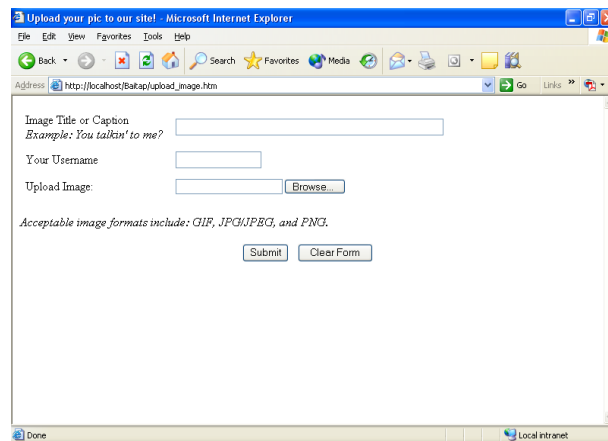
```

It was uploaded on <?php echo \$today; ?>.

</body>

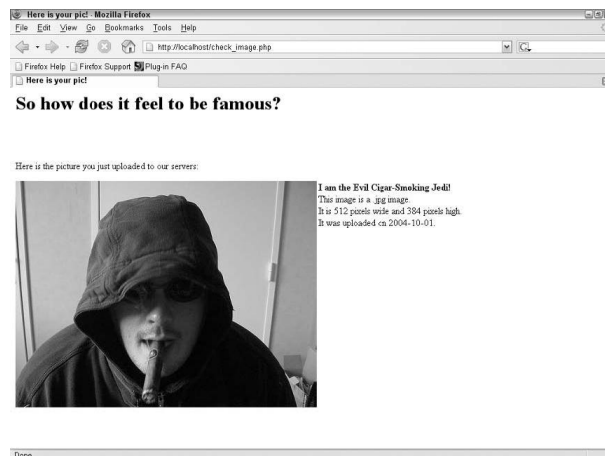
</html>

Lưu file với tên check_image.php. Mở upload_image.htm trên trình duyệt. Kết quả như hình 7.2.2



Hình 7.2.2

Đưa hình ảnh lên, bạn sẽ thấy như Hình 7.2.3



Hình 7.2.3

Cách thức làm việc

Trong upload_image.htm, bạn cho HTML một sức mạnh để tìm vùng đĩa của người dùng với nút “Browse”, đơn giản bằng cách thêm enctype đến thuộc tính form:

```
<form name="form1" method="post" action="check_image.php"
      enctype="multipart/form-data">
```

Sau đó bạn có một vài vùng nhập, bao gồm vùng nhập “file”, lấy file và gửi nó đến server, trong một vùng tạm thời.

Sau đó, trong `check_image.php`, bạn có nhiều thứ khác nhau để tiếp tục. Đầu tiên bạn kết nối cơ sở dữ liệu và tạo những biến dễ dàng cập nhật trong tập lệnh của bạn. Tiếp theo bạn định nghĩa thư mục chứa tất cả các hình và tên hình. Chúng nhìn vào dòng đặc biệt này

```
$image_tempname = $_FILES['image_filename']['name'];
```

Bạn có thể sử dụng nhiều phương pháp khác nhau khi xử lý hình ảnh. Nếu bạn nghĩ bạn sẽ có một số file cho mỗi người dùng. Bạn có thể tạo thư mục cho mỗi người, sau đó chuyển hình ảnh cho mỗi thư mục. Trong ví dụ này, bạn giữ tất cả hình ảnh trong một thư mục lớn. Không quan tâm đến cấu trúc thư mục bạn chọn, bạn nên áp dụng một vài kiểm tra cho những tên file giống nhau.

Trong trường hợp này, bạn đổi tên những tên file giống nhau như gán cho nó một ID duy nhất. bảo đảm là mỗi file sẽ có một tên duy nhất, không có vấn đề gì nếu hai người đưa vào một tên file `photo1.jpg`. Do đó bạn sẽ lấy tên file tạm thời mà nó được đưa lên bằng cách sử dụng biến `$image_tempname`, và đổi tên nó một lần nữa, nó đã được đưa lên thành công và được chèn vào bảng của bạn, sử dụng biến `$newfilename` mà bạn thấy trong tập lệnh.

Kế tiếp bạn kiểm tra để chắc rằng file đã được đưa lên thành công với dòng sau.

```
if (move_uploaded_file($_FILES['image_filename']['tmp_name'],  
$imageName))
```

Hàm `move_uploaded_file` chuyển một file uploaded từ hình gốc được cung cấp bởi người dùng, gán ‘`tmp_name`’ bởi máy chủ để đến đích cuối cùng, `$imageName`. Nó thì quan trọng bao gồm bước ‘`tmp_name`’ trong tập lệnh của bạn, chú ý rằng bạn không làm bất kì một cái gì với giá trị trong biến này, nó được đưa ra bởi máy chủ. Giá trị thì ẩn từ bạn nhưng nó quan trọng, bạn không đặt tên nó và sau đó biến được chuyển đến.

Bước tiếp theo là lấy thông tin về file được đưa lên. Trong ví dụ bạn chỉ cho phép nhiều file hoạt động với phiên bản hiện hành của PHP/GD. Bao gồm file GIF, JPG, và PNG. Tất cả các file này thì dễ dàng thao tác trong PHP cũng như WPMP.

WBMP không giống như một file trong Window Bitmap, Nó là file Wireless Bitmap, sử dụng trong Palm Pilots. Vào lúc viết PHP/GD không được thao tác trên file BMP. Bạn cần một ứng dụng khác như ImageMagick để đổi file BMP thành GIF, JPG, or PNG. Nếu bạn muốn làm việc với chúng mà dùng PHP/GD.

Hàm getimagesize cho bạn thông tin về chiều dài, chiều rộng và loại hình ảnh và cho những file JPG, nó có thể cho bạn số kênh và số đơn vị. Nó đưa ra thông tin trong một mảng mà bạn cập nhật dùng hàm list:

```
list($width, $height, $type, $attr) = getimagesize($ImageName);
```

Chiều rộng và chiều cao của hình ảnh thì trả về số nguyên. Loại file thì trả về một số nguyên với khóa dưới đây

1 GIF	9 JPC
2 JPG	10 JP2
3 PNG	11 JPX
4 SWF	12 JB2
5 PSD	13 SWC
6 BMP	14 IFF
7 TIFF (Intel byte order)	15 WBMP
8 TIFF (Motorola byte order)	16 XBM

Biến \$attr chứa chiều rộng và chiều cao mà bạn sử dụng trong thẻ hình HTML

Ví dụ

```
width="640" height="480
```

Trở về tập lệnh sử dụng switch để lọc ra những loại hình ảnh không sử dụng

```
switch ($type)
```

```
{
```

```
case 1:
```

```
    $ext = ".gif";
```

```
    break;
```

```
case 2:
```



```

        $ext = “.jpg”;
        break;
    case 3:
        $ext = “.png”;
        break;
    default:
        echo “Sorry, but the file you uploaded was not a GIF, JPG, or “.
        “PNG file.<br>”;
        echo “Please hit your browser’s ‘back’ button and try again.”;
    }

```

Bạn gán file mở rộng dựa vào loại file và bạn sẽ cần có những thông tin có sẵn khi bạn đổi tên file. Nếu file đưa lên không thích hợp với bất kì trường hợp nào của bạn thì áp dụng mặc định. Và người đọc sẽ thấy câu “Sorry, but the file you uploaded was not a GIF, JPG or PNG file”. Với cách này bạn có thể lọc ra những file không được chấp nhận mà vẫn đưa lên.

Sau đó bạn chèn thông tin vào bảng như đoạn mã dưới đây;

```

//chèn thông tin vào bảng
$insert = “INSERT INTO images
(image_caption, image_username, image_date)
VALUES
(‘$image_caption’, ‘$image_username’, ‘$today’)”;
$insertresults = mysql_query($insert)
or die(mysql_error());
$lastpicid = mysql_insert_id();

```

Sau đó bạn đặt lại tên file để tránh xung đột trong tương lai sử dụng ID tăng tự động của hình ảnh;

```

$newfilename = $ImageDir . $lastpicid . $ext;
rename($ImageName, $newfilename);

```

Chương 7 : THAO TÁC VÀ TẠO HÌNH ẢNH VỚI PHP

Chương này bao gồm những phần sau:

- Có thể tải đặt PHP bao gồm cả thư viện GD.
- Cho phép người dùng đưa hình ảnh của họ lên.
- Nhận lại thông tin về hình ảnh như loại, kích thước.
- Tạo một hình ảnh mới.
- Sao chép một ảnh hoặc một phần của ảnh.
- Tạo thumbnails(phiên bản thu nhỏ của ảnh).
- Tạo một phiên bản ảnh trắng đen.
- Thêm hoa văn và tên cho hình.

Chuyển đổi loại file hình ảnh

Có một vài chú ý về cách chèn hình ảnh của bạn khi đưa nó đến với người dùng. Hãy nhìn vào dòng sau:

```

```

bạn sử dụng hai biến `$lastpicid` và `$ext` để điều chỉnh tên hình ảnh của bạn. Bạn có chú ý rằng, thông tin không được lưu trữ ở bất cứ nơi nào trong bảng hình ảnh không? Bạn sẽ cập nhật lại hình ảnh như thế nào khi thông tin trong biến đã hết hiệu lực. Bạn có thể cập nhật phần đầu tiên của tên file, bởi vì nó giống như `image_id`. Bạn biết về phần mở rộng của file như thế nào, nếu nó khác nhau cho mỗi hình. Bạn có thể làm một trong 3 biện pháp cứu chữa này:

1. Thêm vào một vùng trong bảng hình ảnh cho phép lưu trữ đầy đủ tên hình ảnh.
2. Thêm vào một vùng trong bảng hình ảnh cho phép lưu trữ phần mở rộng.
3. Chuyển đổi tất cả hình ảnh đưa vào thành những loại file giống nhau và có phần mở rộng giống nhau.

Chúng ta sẽ chọn cách thứ 3. Ở đây, chúng ta sẽ thay đổi file check_image.php một cách hợp lý. Bằng cách tạo ra một bản sao loại .jpg(trong 3 loại GIF, JPG, PNG) chứ không thay đổi bản.

Để chuyển đổi loại file bạn làm 4 bước sau:

1. Tạo một hình ảnh GD-thân thiện mới từ hình ảnh gốc để làm một hình ảnh nguồn tạm.

2. Tạo một hình ảnh GD-thân thiện rộng để làm một hình ảnh đích tạm.

3. Sao chép hình ảnh nguồn mới đến hình ảnh đích mới.

4. Lưu hoặc xuất ra hình ảnh đích đã được thay đổi.

5. (Tùy ý, nhưng khuyên dùng) xóa hình ảnh nguồn tạm và hình ảnh đích tạm.

PHP có những hàm cho các file loại đặc biệt cho bước 1 và 4(ví dụ imagecreatefromgif, imagecreatefromjpg), vì vậy, điều quan trọng là bạn biết loại file nào mà bạn đang làm việc với nó.

Ví dụ: sắp xếp hợp lý tiến trình

Để xử lý file của bạn cần thay đổi check_image.php như sau:

```
<?php
//kết nối cơ sở dữ liệu
$link = mysql_connect("localhost", "root", "")
    or die("Could not connect: " . mysql_error());
mysql_select_db("moviesite", $link)
    or die (mysql_error());
//thêm biến có sẵn
$image_caption = $_POST['image_caption'];
$image_username = $_POST['image_username'];
$image_tempname = $_FILES['image_filename']['name'];
$today = date("Y-m-d");
//đưa hình ảnh và kiểm tra loại hình ảnh
$imageDir ="c:/Program Files/Apache
Group/Apache2/test/images/";
```

```

$ImageName = $ImageDir . $image_tempname;
if (move_uploaded_file($_FILES['image_filename']['tmp_name'],
$ImageName))
{
//Lấy thông tin về hình ảnh đưa lên
    list($width, $height, $type, $attr) =
        getimagesize($ImageName);
/**Xóa từ dòng này
switch ($type)
{
    case 1:
        $ext = “.gif”;
        break;
    case 2:
        $ext = “.jpg”;
        break;
    case 3:
        $ext = “.png”;
        break;
    default:
        echo “Sorry, but the file you uploaded was not a GIF, JPG, or “.
        “PNG file.<br>”;
        echo “Please hit your browser’s ‘back’ button and try again.”;
    }
/**kết thúc xóa
/**thêm những dòng mới này
if ($type > 3)
{
    echo “Sorry, but the file you uploaded was not a GIF, JPG, or “.

```

```

        "PNG file.<br>";
        echo "Please hit your browser's 'back' button and try again.";
    } else
    {
        //Hình ảnh có thể chấp nhận
        /**kết thúc việc chèn
        //chèn thông tin về bảng hình ảnh
            $insert = "INSERT INTO images
            (image_caption, image_username, image_date)
            VALUES
            ('$image_caption', '$image_username', '$today')";
            $insertresults = mysql_query($insert)
            or die(mysql_error());
            $lastpicid = mysql_insert_id();
        //thay đổi dòng này:
        $newfilename = $ImageDir . $lastpicid . ".jpg";
        /**chèn dòng này
        if ($type == 2)
        {
            rename($ImageName, $newfilename);
        } else
        {
            if ($type == 1)
            {
                $image_old = imagecreatefromgif($ImageName);
            } elseif ($type == 3)
            {
                $image_old = imagecreatefrompng($ImageName);
            }
        }
    }

```

```

//chuyển đổi hình ảnh thành jpg
    $image_jpg = imagecreatetruecolor($width, $height);
    imagecopyresampled($image_jpg, $image_old, 0, 0, 0, 0,
    $width, $height, $width, $height);
    imagejpeg($image_jpg, $newfilename);
    imagedestroy($image_old);
    imagedestroy($image_jpg);
    }
    $url = "location: showimage.php?id=" . $lastpicid;
    header($url);
/**kết thúc dòng chèn
}
?>
<!-- DELETE THESE LINES
<html>
<head>
<title>Here is your pic!</title>
</head>
<body>
<h1>So how does it feel to be famous?</h1><br><br>
<p>Here is the picture you just uploaded to our servers:</p>

<strong><?php echo $image_caption; ?></strong><br>
This image is a <?php echo $ext; ?> image.<br>
It is <?php echo $width; ?> pixels wide
and <?php echo $height; ?> pixels high.<br>
It was uploaded on <?php echo $today; ?>.
</body>
</html>

```

END OF DELETED LINES-->

Mở trình soạn thảo và nhập đoạn mã lưu với tên Showimage.php

```
<?php
    //kết nối cơ sở dữ liệu
    $link = mysql_connect("localhost", "root", "")
        or die("Could not connect: " . mysql_error());
    mysql_select_db("moviesite", $link)
        or die (mysql_error());
    //make variables available
    $id = $_REQUEST['id'];
    //get info on the pic we want
    $getpic = mysql_query("SELECT * FROM images WHERE image_id =
    '$id'")
        or die(mysql_error());
    $rows = mysql_fetch_array($getpic);
    extract($rows);
    $image_filename = "images/" . $image_id . ".jpg";
    list($width, $height, $type, $attr) = getimagesize($image_filename);
?>
<html>
<head>
<title>Here is your pic!</title>
</head>
    <body>
        <h1>So how does it feel to be famous?</h1><br><br>
        <p>Here is the picture you just uploaded to our servers:</p>
         >
        <strong><?php echo $image_caption; ?></strong><br>
```

```

        It is <?php echo $width; ?> pixels wide a
        nd <?php echo $height; ?> pixels high.<br>
        It was uploaded on <?php echo $image_date; ?>
        by <?php echo $image_username; ?>.
    </body>
</html>

```

3. Nếu bạn lưu file, đưa lên trình duyệt, và đưa hình ảnh của bạn, bạn sẽ chú ý rằng về cơ bản giống như màn hình trước.

Cách thức làm việc

Hãy nhìn vào phần chính mà bạn thêm trong chương trình của bạn. Đầu tiên chúng ta sẽ xử lý với file JPG bởi vì chúng sẵn sàng trong định dạng mà bạn muốn.

```

if ($type == 2) {
    rename($ImageName, $newfilename);

```

Ở đây, nếu file là file JPG, bạn sẽ đổi tên nó thành ID hình ảnh của bạn cộng với phần mở rộng ".jpg".

Mặc khác, nếu file là GIF hoặc PNG bạn sử dụng hàm thích hợp để xử lý chúng. Bạn kiểm tra để thấy nếu file là JPG

```

    } else {
        if ($type == 1) {
            $image_old = imagecreatefromgif($ImageName);

```

imagecreatefromgif() là hàm hợp lý, bởi vì hình gốc có đuôi GIF. Nếu hình là PNG thì bạn sử dụng hàm imagecreatefrompng :

```

        } elseif ($type == 3) {
            $image_old = imagecreatefrompng($ImageName);
        }

```

Bây giờ bạn có hình gốc GD thân thiện, bạn cần đến bước 2 để tạo hình ảnh đích tạm. Bạn làm điều này trong dòng tiếp theo

```

$image_jpg = imagecreatetruecolor($width, $height);

```


Bạn sử dụng hàm `imagecreatetruecolor()` để tạo màu cho hình. Bạn muốn kích thước của hình thay đổi bạn sử dụng biến `$width` và `$height` được chứa trong hàm `getimagesize`.

Bây giờ bạn chuyển sang bước 3 trong dòng tiếp theo:

```
imagecopyresampled($image_jpg, $image_old, 0, 0, 0, 0,  
$width, $height, $width, $height);
```

Bạn sử dụng hàm `imagecopyresampled` để quyết định thuộc tính cho hình. Ở đây hình gốc và hình đích đều bắt đầu với tọa độ `x,y` với chiều dài và chiều rộng của mỗi hình. Nếu bạn chỉ muốn sao chép vị trí của hình gốc vào hình đích bạn có thể dùng tọa độ và các biến `$width/$height`.

Tiếp theo, bạn lưu hình ảnh bằng dòng sau:

```
imagejpeg($image_jpg, $newfilename);
```

Đây là nơi thật sự chuyển đổi, trước dòng này hình tạm có đặc điểm chung.

Bạn quyết định làm file đích JPG với hàm `imagejpeg`. Bạn cũng có thể sử dụng `imagepng` hoặc `imagegif`, nhưng một lần nữa, bạn muốn làm việc với JPG vì phần lớn những file bạn load lên sẽ được sao chép. Trong hàm này bạn đặt tên file nguồn tạm và file đích lâu dài. Quan trọng bạn cần có phần tên dùng biến `$newfilename`, bạn cũng có thể xóa những hình tạm như dưới đây

```
imagedestroy($image_old);  
imagedestroy($image_jpg);
```

Trong bảng tóm tắt, bạn đổi tên file JPG thành `image_id.jpg`, Hoặc tạo một bản sao và lưu với và lưu file JPG với tên `image_id.jpg`.

Tập lệnh cũng xây dựng lại một file mới `showimage.php` mà chủ yếu là điền thông tin mà bạn đã nhập vào cơ sở dữ liệu. Điều khác biệt là bạn có thể cập nhật mọi lúc mọi nơi, và cho phép người dùng có thể sửa đổi.

Chương 7 : THAO TÁC VÀ TẠO HÌNH ẢNH VỚI PHP

Chương này bao gồm những phần sau:

- Có thể tải đặt PHP bao gồm cả thư viện GD.
- Cho phép người dùng đưa hình ảnh của họ lên.
- Nhận lại thông tin về hình ảnh như loại, kích thước.
- Tạo một hình ảnh mới.
- Sao chép một ảnh hoặc một phần của ảnh.
- Tạo thumbnails(phiên bản thu nhỏ của ảnh).
- Tạo một phiên bản ảnh trắng đen.
- Thêm hoa văn và tên cho hình.

Màu trắng và đen

Bây giờ bạn đã có thư mục với nhiều hình ảnh. Nếu muốn cho phép người dùng làm những hình ảnh thành màu trắng và đen. Chúng ta hãy thêm tùy chọn đó vào trang showimage, như vậy người dùng có thể chọn là thấy hay không thấy hình ảnh trong “grayscale”.Bạn sẽ sử dụng hàm imagefilter(), để chuyển hình ảnh tới grayscale. Hàm này cũng có thể làm ảnh ngược, độ chói, làm nổi, mờ, làm nhẵn và màu của hình ảnh...Có thể dùng hàm này để làm sạch hoặc tạo ra những phiên bản mới nhất của những bức ảnh uploaded.

Ví dụ:Thêm Grayscale

Trong bài tập này, chúng ta chỉ thêm những đặc tính imagefilter() vào vị trí, cho phép người dùng tùy chọn để thấy hình ảnh trong grayscale

1.Mở file showimage.php và thay đổi như sau:

```
<?php
//Kết nối cơ sở dữ liệu
$link = mysql_connect(“localhost”, “root”, “”)
```

```

or die("Could not connect: " . mysql_error());
mysql_select_db("moviesite", $link)
or die (mysql_error());
//Làm những biến có sẵn
$id = $_REQUEST['id'];
/**Những dòng chèn thêm
if (isset($_REQUEST['mode']))
{
    $mode = $_REQUEST['mode'];
} else
{
    $mode = '';
}
/**Kết thúc dòng chèn
//Tạo thông tin chúng ta muốn
$getpic = mysql_query("SELECT * FROM images WHERE image_id =
'$id'")
or die(mysql_error());
$rows = mysql_fetch_array($getpic);
extract($rows);
$image_filename = "images/" . $image_id . ".jpg";
list($width, $height, $type, $attr) = getimagesize($image_filename);
?>
<html>
<head>
<title>Here is your pic!</title>
</head>
<body>
<h1>So how does it feel to be famous?</h1><br><br>

```

```

<!--INSERT THE FOLLOWING LINES: -->
<?php
if ($mode == 'change') {
echo "<font color='CC0000'><em><strong>Your image has been
modified.</strong></em></font>";
echo "<img src='\"' . $image_filename . \"' align='left'\" \" .
$attr . \">\";
} else {
?>
<!--END OF INSERTED LINES-->
<p>Here is the picture you just uploaded to our servers:</p>
 >
<strong><?php echo $image_caption; ?></strong><br>
It is <?php echo $width; ?> pixels wide
and <?php echo $height; ?> pixels high.<br>
It was uploaded on <?php echo $image_date; ?>
by <?php echo $image_username; ?>.
<!--INSERT THE FOLLOWING LINES:-->
<?php
//end the else
}
?>
<hr>
<p><em><strong>Modifying Your Image</strong></em></p>
<form action="modifyimage.php" method="post">
<p>
Please choose if you would like to modify your image with any of
the following options. If you would like to preview the image

```

before saving, you will need to hit your browser's 'back' button to return to this page. Saving an image with any of the modifications listed below *cannot be undone.*

```
</p>
<input name="id" type="hidden" value="<?php echo $image_id;
?>">
<input name="bw" type="checkbox">black & white<br>
<p align="center">
<input type="submit" name="action" value="preview">
<input type="submit" name="action" value="save">
</p>
</form>
<!--END OF INSERTED LINES-->
</body>
</html>
```

2. Kế đến, ta tạo hồ sơ mới để sửa đổi hình ảnh, lưu với file modify.php

```
<?php
//kết nối cơ sở dữ liệu
$link = mysql_connect("localhost", "root", "");
or die("Could not connect: " . mysql_error());
mysql_select_db("moviesite", $link)
or die (mysql_error());
//làm những biến có sẵn
$id = $_POST['id'];
if (isset($_POST['bw']))
{
    $bw = $_POST['bw'];
} else
```

```

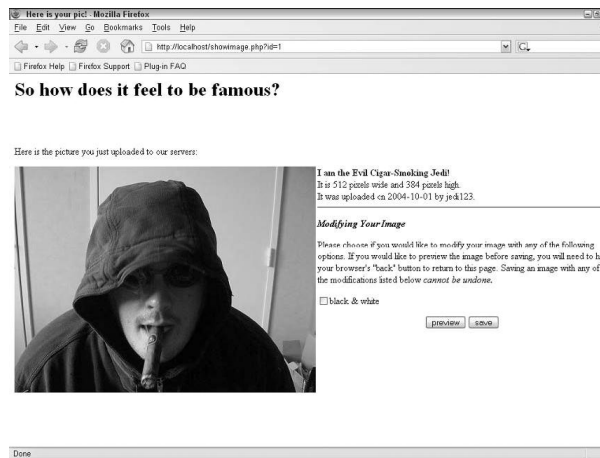
{
    $bw = '';
}
$action = $_POST['action'];
//tạo thông tin bạn muốn
$getpic = mysql_query("SELECT * FROM images WHERE
image_id = '$id'")
or die(mysql_error());
$rows = mysql_fetch_array($getpic);
extract($rows);
$image_filename = "images/" . $image_id . ".jpg";
list($width, $height, $type, $attr) =
getimagesize($image_filename);
$image = imagecreatefromjpeg("$image_filename");
if ($bw == 'on')
{
    imagefilter($image, IMG_FILTER_GRAYSCALE);
}
if ($action == "preview")
{
    header("Content-type:image/jpeg");
    imagejpeg($image);
}
if ($action == "save")
{
    imagejpeg($image, $image_filename);
    $url = "location:showimage.php?id=" . $id .
"&mode=change";
    header($url);
}

```

}

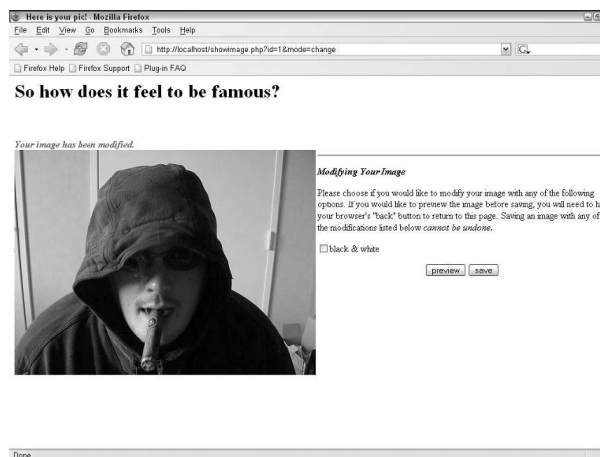
?>

1. Chúng ta không cần nạp dữ liệu vào hình khác, bởi vì chúng ta không thay đổi gì trong các bước của tiến trình. Chúng ta hãy vào `http://localhost/showimage.php?id=1`, bạn sẽ thấy như hình dưới đây



Hình 7.4.1

Nếu xem trước hình ảnh trắng đen, bạn sẽ thấy màn hình không có gì. Cố gắng quay lại và lưu file trong màu trắng đen cùng lúc.



Hình 7.4.2

Nó làm việc thế nào?

Nếu bạn nhìn những dòng bạn thêm trong `showiamge.php`, bạn thấy những dòng này trước tiên:

```
/**Những dòng chèn thêm
```


to return to this page. Saving an image with any of the modifications listed below *cannot be undone.*

</p>

```
<input name="id" type="hidden" value="<?php echo $image_id; ?>">
```

```
<input name="bw" type="checkbox">black & white<br>
```

```
<p align="center">
```

```
<input type="submit" name="action" value="preview">
```

```
<input type="submit" name="action" value="save">
```

</p>

</form>

<!--END OF INSERTED LINES-->

Chúng ta mới tạo file modifileimage.php, cho đến hàng này thì mọi thứ đã khá chuẩn

```
$image = imagecreatefromjpeg("$image_filename");
```

Bạn đã có quá trình 5 bước chuyển đổi, bạn tạo ra một GD_ bản sao hình ảnh thân thiện của hình ảnh nguồn, tiếp theo bạn muốn làm tất cả những gì tạo ra hình ảnh thân thiện như thấy dưới đây

```
if ($bw == 'on') {  
    imagefilter($image, IMG_FILTER_GRAYSCALE);  
}
```

Bạn quyết định là giữ hay không giữ những thay đổi, bằng những dòng code sau:

```
if ($action == "preview") {  
    header("Content-type:image/jpeg");  
    imagejpeg($image);  
}
```

Nếu bạn chỉ xem trước hình ảnh, bạn gọi trực tiếp đến trình duyệt(được sử dụng bởi hàm iamgejpeg()). Nhưng tại sao lại gọi đầu mục cho trình duyệt và tại sao lại không có HTML? Một lần nữa vì bạn gọi hình ảnh trực tiếp, bạn không cần(và thật sự không thể gọi) bất kỳ văn bản nào với hình ảnh. Bạn chỉ gọi đơn thuần hình ảnh. Nếu bạn chỉ rõ

kiểu nội dung trong trang thì trình duyệt sẽ hiểu là văn bản và bạn có một trang đầy rác. Bạn phải thêm vào đầu mục để trình duyệt giải thích dòng hình ảnh chính xác.

Vì trình duyệt đang chờ một hình ảnh, nếu gửi bất kỳ một văn bản kèm theo hình ảnh nào, cũng gửi thêm đầu mục thì bạn sẽ nhận được lỗi.

Nếu người dùng muốn lưu những thay đổi hình ảnh và click vào “save” khi đang submit, bạn phải thực hiện những dòng sau:

```
if ($action == “save”) {  
    imagejpeg($image, $image_filename);  
    $url = “location:showimage.php?id=”. $id . “&mode=change”;  
    header($url);  
}
```

Chú ý việc sử dụng kiểu biến(người sử dụng sẽ thấy thông tin văn bản khi trang đưa lên) và filename chỉ rõ trong hàm imagejpeg(). Chỉ rõ filename đến lưu hình ảnh tạm trong thời gian dài. Trong trường hợp này, nó ghi đè lên file có sẵn, làm cho sự chuyển đổi lâu.

Chương 7 : THAO TÁC VÀ TẠO HÌNH ẢNH VỚI PHP

Chương này bao gồm những phần sau:

- Có thể tải đặt PHP bao gồm cả thư viện GD.
- Cho phép người dùng đưa hình ảnh của họ lên.
- Nhận lại thông tin về hình ảnh như loại, kích thước.
- Tạo một hình ảnh mới.
- Sao chép một ảnh hoặc một phần của ảnh.
- Tạo thumbnails(phiên bản thu nhỏ của ảnh).
- Tạo một phiên bản ảnh trắng đen.
- Thêm hoa văn và tên cho hình.

Thêm những dấu đề

Một nhóm hàm đặc biệt cho phép bạn thêm những dấu đề(hoặc chú thích một bản quyền , một văn bản khác). PHP/GD cho phép bạn thay đổi kiểu phông và kích thước sử dụng, thậm chí cho phép bạn tải phông theo yêu cầu.

Ví dụ về việc thêm văn bản trong hình ảnh

Bạn thay đổi file showimage.php và file modifyimage.php để thấy những dấu đề kèm theo hình ảnh.

1. Xác định vị trí mục sau trong file showimage.php và thêm những dòng sau:

```
<hr>
<p><em><strong>Modifying Your Image</strong></em></p>
<form action="modifyimage.php" method="post">
<p>
```

Hãy chọn nếu bạn muốn thay đổi hình ảnh với bất kỳ lựa chọn nào dưới đây. Nếu muốn xem hình ảnh trước khi cất giữ, bạn cần đánh nút “back” để trả về trang này. Lưu hình ảnh với danh sách:Không thể hủy.

```

</p>
<input name="id" type="hidden" value="<?php echo $image_id; ?>">
<input name="bw" type="checkbox">black & white<br>
<!--INSERT THE FOLLOWING LINE-->
<input name="text" type="checkbox">embedded caption<br>
<!--END OF INSERTED LINES-->
<p align="center">
<input type="submit" name="action" value="preview">
<input type="submit" name="action" value="save">
</p>
</form>

```

2. Phông “arial.ttf” được dùng trong bài tập này, nhưng bạn nên dùng phông được cài đặt trong máy chủ. Nếu thử chạy phiên bản mà không cài đặt phông trên máy chủ, sẽ báo lỗi. Trong file modifyimage.php, thêm những dòng sau:

```

<?php
//Kết nối cơ sở dữ liệu
$link = mysql_connect("localhost", "bp5am", "bp5ampass")
or die("Could not connect: " . mysql_error());
mysql_select_db("moviesite", $link)
or die (mysql_error());
//Làm những biến có sẵn
$id = $_POST['id'];
if (isset($_POST['bw'])) {
    $bw = $_POST['bw'];
} else {
    $bw = '';
}
$action = $_POST['action'];
/**INSERT THE FOLLOWING LINES:

```

```

if (isset($_POST['text'])) {
$text = $_POST['text'];
} else {
$text = '';
}
/**END OF INSERT
//get info on the pic we want
$getpic = mysql_query("SELECT * FROM images WHERE image_id =
'$id'")
or die(mysql_error());
$rows = mysql_fetch_array($getpic);
extract($rows);
$image_filename = "images/" . $image_id . ".jpg";
list($width, $height, $type, $attr) = getimagesize($image_filename);
$image = imagecreatefromjpeg("$image_filename");
if ($bw == 'on') {
imagefilter($image, IMG_FILTER_GRAYSCALE);
}
/**INSERT THE FOLLOWING LINES:
if ($text == 'on') {
imagefttext($image, 12, 0, 20, 20, 0, "arial.ttf", $image_caption);
}
/**END OF INSERT
if ($action == "preview") {
header("Content-type:image/jpeg");
imagejpeg($image);
}
if ($action == "save") {
imagejpeg($image, $image_filename);
}

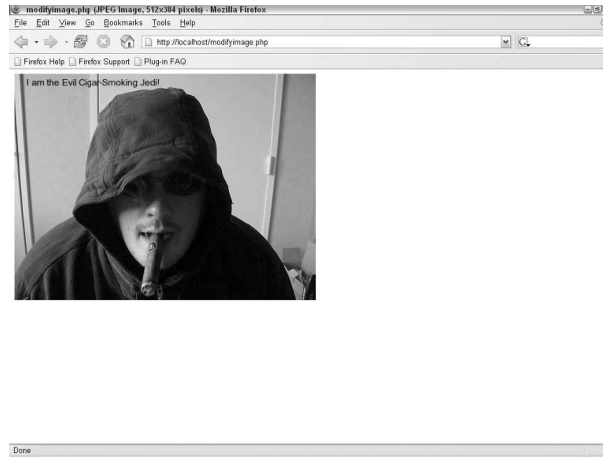
```

```

$url = "location:showimage.php?id=". $id . "&mode=change";
header($url);
}
?>

```

1. Quay lại trang <http://localhost/showimage.php?id=1> và xem trước ảnh với việc”nhúng dấu đề” có đặc tính kiểm tra. Sẽ thấy như Hình 7.5.



Hình 7.5

2. Bạn có thể dễ dàng thêm tự động những chú thích bản quyền hay bất kỳ một văn bản nào trong ảnh.

Cách thức hoạt động.

Trước tiên, thêm lựa chọn”nhúng dấu đề” trong showimage.php. Sau đó, thêm hàm imagettftext trong file modifyimage như sau:

```

/**Bắt đầu chèn:
if ($text == 'on') {
imagettftext($image, 12, 0, 20, 20, 0, "arial.ttf", $image_caption);
}

```

/**Kết thúc chèn

Hàm imagettftext() là một trong nhiều hàm văn bản/chuỗi có sẵn trong PHP/GD. Hàm tạo 8 giá trị:

1. Hình ảnh nơi chèn văn bản (\$image trong ví dụ)

2. Cỡ phông văn bản(trong ví dụ là 12)
3. Sự luân phiên văn bản(trong ví dụ là 0)
4. Tọa độ X là vị trí bắt đầu của văn bản, với 0 là ranh giới(trong ví dụ là 20)
5. Tọa độ y là vị trí bắt đầu của văn bản, với 0 là ranh giới trên của hình ảnh(trong ví dụ này là 20)
6. Màu sử dụng (trong ví dụ là 0, màu đen)
7. Tên phông, được định vị tự động trong thư mục phông mặc định(Trong ví dụ này là phông arial.ttf)
8. Chuỗi văn bản được cho thấy(Trong ví dụ là biến image_option)

Trong quá trình làm nhớ phải điền đầy đủ 8 giá trị, nếu không sẽ gặp lỗi)

Chương 7 : THAO TÁC VÀ TẠO HÌNH ẢNH VỚI PHP

Chương này bao gồm những phần sau:

- Có thể tải ảnh PHP bao gồm cả thư viện GD.
- Cho phép người dùng đưa hình ảnh của họ lên.
- Nhận lại thông tin về hình ảnh như loại, kích thước.
- Tạo một hình ảnh mới.
- Sao chép một ảnh hoặc một phần của ảnh.
- Tạo thumbnails(phiên bản thu nhỏ của ảnh).
- Tạo một phiên bản ảnh trắng đen.
- Thêm hoa văn và tên cho hình.

Thêm hình mờ và kết hợp hình ảnh

Bởi vì bạn đang thấy những hình ảnh trong Movie review Site, bạn muốn làm biểu tượng đăng sau mỗi hình, như một loại hình mờ. Chúng ta có thể làm điều này với biểu tượng của mình để bảo vệ bất kỳ hình ảnh copy nào. Trong ví dụ này, chúng ta sẽ hòa trộn 2 hình (hình nguồn và hình biểu tượng) để tạo hiệu ứng. Chúng ta thay đổi như sau:

Ví dụ : Trộn 2 hình

Trộn 2 hình, bạn cần thay đổi trong showimage.php và trong modifyimage.php

1. Thêm vào những dòng sau trong showimage.php, trong phần như trước

```
<input name="watermark" type="checkbox">include Movie Review Site  
watermark<br>
```

2. Thêm vào những dòng sau trong modifyimage.php, như trước

```
/**Chèn gần đỉnh của file  
if (isset($_POST['watermark']))
```



```

{
    $watermark = $_POST['watermark'];
} else
{
    $watermark = '';
}

```

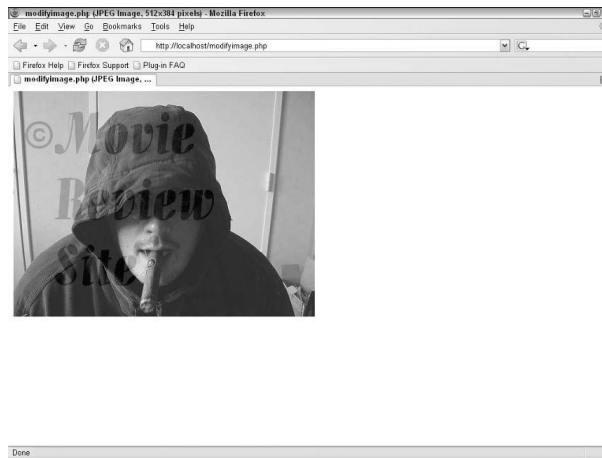
Sau đó thêm vào những dòng dưới đây vào sau tập lệnh

```

if ($watermark == 'on') {
    $image2 = imagecreatefromgif("images/logo.gif");
    imagecopymerge($image, $image2, 0,0,0, 0, $width, $height, 15);
}

```

3. Mặc dù không có file logo.gif nhưng bạn có thể sử dụng bất kỳ file nào bạn thích, chắc chắn rằng tập lệnh có thể tìm thấy. Bạn sẽ thấy như hình 7.6 sau:



Hình 7.6

Cách thức hoạt động:

Bạn thêm tùy chọn khác cho người dùng, bạn sử dụng hàm `imagecopymerge()` trong `modifyimage.php`. Chú ý là trước khi hòa trộn hai hình, phải làm hình “GD thân thiện” thứ hai bằng việc tạo một bản sao chép. Vì hình ảnh của bạn là hình GIF nên sử dụng hàm `imagecreatefromgif()`

Nguyên bản của hàm:

`imagecopymerge($image, $image2, 0,0,0, 0, $width, $height, 15);`

Những tham số trong hàm `imagecopymerge` như sau:

1. Tên của hình ảnh tới(Trong ví dụ này, dùng biến `$image`, file `$image` thay đổi tất cả và sẽ thấy ở cuối của nguyên bản)
2. Tên của hình “lần hai” hoặc hình nguồn(Trong ví dụ này là `$image2`)
3. Tọa độ X của hình đến(Trong ví dụ này là 0, đại diện cho ranh giới dưới)
4. Tọa độ Y của hình ảnh đến (Trong ví dụ này là 0, đại diện cho ranh giới trên)
5. Tọa độ X của hình thứ hai(Trong ví dụ này là 0)
6. Tọa độ Y của hình thứ hai (Trong ví dụ này là 0)
7. Chiều rộng của phần hình thứ hai được trộn(Trong ví dụ này dùng biến `$width`, đại diện cho hình thứ hai phù hợp với hình đến)
8. Chiều cao của phần hình ảnh thứ hai được hòa trộn(Trong ví dụ này, dùng biến `$height`, đại diện cho hình thứ hai phù hợp với hình đến)
9. Phần trăm của hai hình, với 100 nghĩa là hình thứ hai hoàn toàn đè lên hình thứ nhất (Trong ví dụ này là 15)

Chúng ta hãy nói về số 7 và số 8. Vì hàm `imagecopymerge()` có thể hòa trộn một phần hình với hình khác, bạn phải chỉ rõ bao nhiêu hình muốn hòa trộn. Biểu tượng CBA thì không lờ, lớn hơn bức ảnh người dùng. Bạn chỉ muốn trộn phần ảnh lớn bằng bức ảnh người dùng, mà tại sao bạn sử dụng `$height` và `$width`. Nếu logo nhỏ, bạn chỉ rõ chiều rộng và chiều cao của chúng, nếu muốn hòa trộn toàn bộ những thứ với hình ảnh đầu tiên.

Tạo Thumbnails

Tất nhiên, việc cho thấy những hình ảnh của người dùng thì tốt. Tuy nhiên, nó không tốt cho cuộc triển lãm hay một danh sách hình trong một trang. Bạn cần làm sao để tạo ra thumbnail tự động của mỗi file upload mà bạn sử dụng cho mục đích là một phòng triển lãm tranh.

Ví dụ về việc tạo Thumbnails

Bạn muốn tạo tự động một phiên bản thumbnail tất cả các hình mà người dùng upload, bạn cần sửa đổi `check_image.php` và bao gồm hàm:

1. Tạo một danh mục hình ảnh con cho hình ảnh. Ví dụ, bạn tạo c:\Program Files\Apache Group\Apache2\images\thumbs, chắc rằng thư mục được cho phép.
2. Thay đổi check_image.php như sau:

```
<?php
//Kết nối cơ sở dữ liệu
$link = mysql_connect("localhost", "bp5am", "bp5ampass")
or die("Could not connect: " . mysql_error());
mysql_select_db("moviesite", $link)
or die (mysql_error());
//Tạo những biến có sẵn
$image_caption = $_POST['image_caption'];
$image_username = $_POST['image_username'];
$image_tempname = $_FILES['image_filename']['name'];
$today = date("Y-m-d");
//đưa hình ảnh lên và kiểm tra loại hình ảnh
$imageDir = "c:/Program Files/Apache Group/Apache2/test/images/";
/**Chèn
$imageThumb = $imageDir . "thumbs/";
/**Kết thúc chèn
$imageName = $imageDir . $image_tempname;
if (move_uploaded_file($_FILES['image_filename']['tmp_name'],
$imageName))
{
    //get info about the image being uploaded
    list($width, $height, $type, $attr) =
    getimagesize($imageName);
    if ($type > 3)
    {
```

```

        echo "Sorry, but the file you uploaded was not a GIF, JPG, or
        “.
        “PNG file.<br>”;
        echo "Please hit your browser’s ‘back’ button and try again.”;
    } else
    {
//image is acceptable; ok to proceed
//Chèn thông tin vào bảng hình ảnh
$insert = "INSERT INTO images
(image_caption, image_username, image_date)
VALUES
('$image_caption', '$image_username', '$today')";
$insertresults = mysql_query($insert)
or die(mysql_error());
$lastpicid = mysql_insert_id();
$newfilename = $ImageDir . $lastpicid . ".jpg";
if ($type == 2)
{
        rename($ImageName, $newfilename);
} else
{
        if ($type == 1)
        {
                $image_old = imagecreatefromgif($ImageName);
        }
} elseif ($type == 3)
{
        $image_old = imagecreatefrompng($ImageName);
        //”convert” the image to jpg
        $image_jpg = imagecreatetruecolor($width, $height);

```

```

        imagecopyresampled($image_jpg, $image_old, 0, 0, 0, 0,
        $width, $height, $width, $height);
        imagejpeg($image_jpg, $newfilename);
        imagedestroy($image_old);
        imagedestroy($image_jpg);
    }

    /**Bắt đầu chèn
    $newthumbname = $ImageThumb . $lastpicid . ".jpg";
    //thiết lập chiều cho thumbnail
    $thumb_width = $width * 0.10;
    $thumb_height = $height * 0.10;
    //Tạo thumbnail
    $largeimage = imagecreatefromjpeg($newfilename);
    $thumb      =      imagecreatetruecolor($thumb_width,
    $thumb_height);
    imagecopyresampled($thumb, $largeimage, 0, 0, 0, 0,
    $thumb_width, $thumb_height, $width, $height);
    imagejpeg($thumb, $newthumbname);
    imagedestroy($largeimage);
    imagedestroy($thumb);
    /**Kết thúc chèn
    $url = "location: showimage.php?id=" . $lastpicid;
    header ($url);
    }
}
?>

```

3. Bây giờ chúng ta sẽ tạo gallery.php như phòng triển lãm của bạn, nhập đoạn mã sau:

```
<?php
```

```

//Kết nối cơ sở dữ liệu
$link = mysql_connect("localhost", "root", "")
or die("Could not connect: " . mysql_error());
mysql_select_db("moviesite", $link)
or die (mysql_error());
$imageDir = "images";
$imageThumb = $imageDir . "/thumbs/";
?>
<html>
<head>
<title>Welcome to our Photo Gallery</title>
</head>
<body>
<p align="center">Click on any image to see it full sized.</p>
<table align="center">
<tr>
<td align="center">Image</td>
<td align="center">Caption</td>
<td align="center">Uploaded By</td>
<td align="center">Date Uploaded</td>
</tr>
<?php
    //get the thumbs
    $getpic = mysql_query("SELECT * FROM images")
or die(mysql_error());
while ($rows = mysql_fetch_array($getpic)) {
extract($rows);
echo "<tr>\n";
echo "<td><a href=\`" . $imageDir . $image_id . ".jpg\`">";

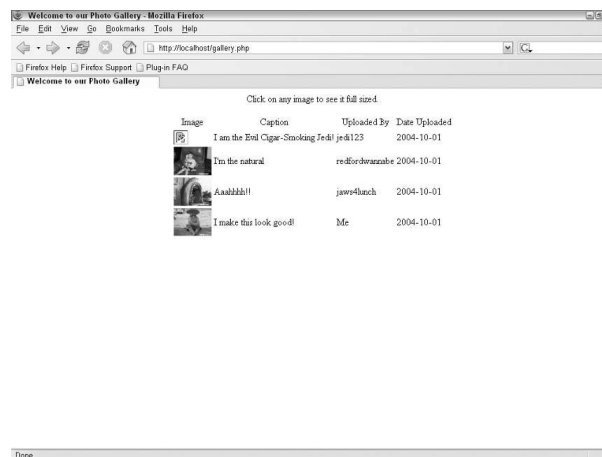
```

```

        echo "<img src=\"\" . $ImageThumb . $image_id . \".jpg\"
        border=\"0\">";
        echo "</a></td>\n";
        echo "<td>" . $image_caption . "</td>\n";
        echo "<td>" . $image_username . "</td>\n";
        echo "<td>" . $image_date . "</td>\n";
        echo "</tr>\n";
    }
?>
</table>
</body>
</html>

```

4. Bây giờ bạn đưa một vài hình sử dụng cho trang upload_image.php. Bạn sẽ thấy như hình 7-8, bạn có thể thêm âm thanh tùy ý.



Hình 7.8

Cách thức hoạt động:

Thực sự tự thumbnails tạo ra trong trang check_image.php, bạn thêm những đoạn sau đây để hoàn thành

```

/**Chèn từ dòng này
$newthumbnail = $ImageThumb . $lastpicid . ".jpg";

```

```
//Tạo kích thước cho thumbnails
$thumb_width = $width * 0.10;
$thumb_height = $height * 0.10;
//Tạo thumbnail
$largeimage = imagecreatefromjpeg($newfilename);
$thumb = imagecreatetruecolor($thumb_width, $thumb_height);
imagecopyresampled($thumb, $largeimage, 0, 0, 0, 0,
$thumb_width, $thumb_height, $width, $height);
imagejpeg($thumb, $newthumbname);
imagedestroy($largeimage);
imagedestroy($thumb);
/**Kết thúc dòng chèn
```

Bạn tạo một thumbnail quá trình gồm 5 bước như trước:

1. Tạo một GD thân thiện từ hình nguồn
2. Tạo một GD thân thiện là một hình trống, với kích thước nhỏ
3. Sao chép hình nguồn vào hình trống
4. cất giữ hình mới đó vào một thư mục thích hợp
5. Xóa những hình tạm thời

Chương 8 : NGƯỜI DÙNG NHẬP CÓ HIỆU LỰC

Nếu bạn chấp nhận cho người dùng được nhập vào trang web, bạn phải chuẩn bị gặp những lỗi. Có thể chỉ là lỗi đơn giản hoặc những lỗi ảnh hưởng trang web của bạn. Lỗi chung nhất thường gặp là lỗi thuật in cơ bản, lỗi định dạng (ví dụ cho một năm trong một ngày). Những lỗi như người dùng không cung cấp địa chỉ email, hoặc người dùng xâm nhập vào cơ sở dữ liệu của bạn. Tập lệnh cần có phần điều khiển nhập, bằng việc xác định dữ liệu xấu và trả về trang thông báo lỗi cho người dùng. Bao gồm:

- ▶ Làm cho có hiệu lực những giá trị chuỗi đơn giản
- ▶ Làm cho có hiệu lực những giá trị nguyên
- ▶ Làm cho có hiệu lực văn bản nhập vào đã định dạng

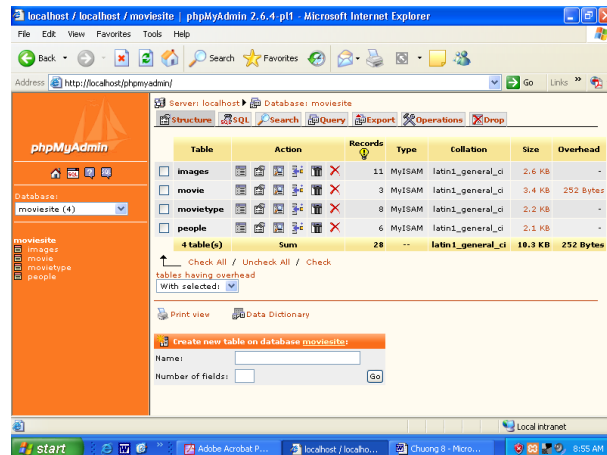
Những người dùng chỉ là những người dùng ...

Ví dụ bây giờ bạn đang làm việc trong ngân hàng, bạn thiết lập hệ thống quản lý tài khoản cập nhật của khách hàng. Bạn sử dụng định dạng MM-DD-YYYY để định dạng ngày tháng. Khi đưa vào hoạt động thì nó không làm việc. Tại sao? Vì tất cả hệ thống của bạn sử dụng ISO 8601 định dạng ngày tháng YYYY-MM-DD. Người sử dụng sẽ nhầm lẫn giữa 2 dạng này nên nhập thông tin không có hiệu lực. Bạn có thể khắc phục điều này bằng việc sử dụng khuôn mẫu chung và làm cho người sử dụng nhập có hiệu lực.

Kết hợp có hiệu lực vào Movie Site

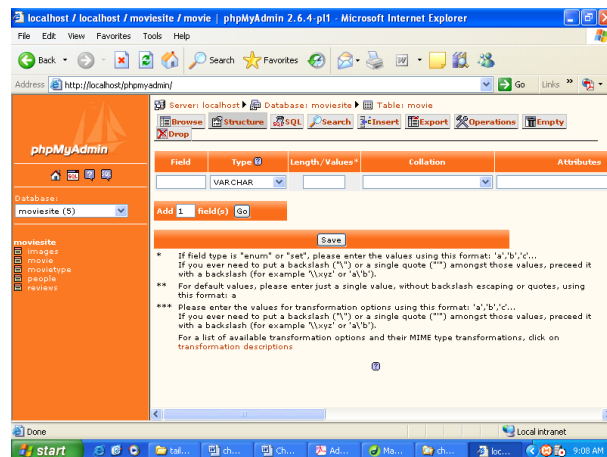
Để thật sự hiểu vai trò và sự kết hợp của người dùng nhập, bạn muốn thấy nó hoạt động. Như vậy, bạn cần thêm vài vùng trong cơ sở dữ liệu của bạn, cung cấp nhiều phương tiện để kiểm tra người dùng nhập vào. Bạn cần thêm một vài đặc tính trong ứng dụng, giúp ta xem lại những phần trước đây.

Thêm vào một vùng movie_release INT(11) với giá trị mặc định là 0 sau vùng movie_year, như Hình 8.1



Hình 8.1

Cho phép bạn lưu trữ một timestamp cho ngày tháng trong phiên bản movie. Thêm vùng movie_rating vào cuối kiểu TINYINT(2). Thông tin đó chứa đánh giá khi bạn xem lại.(xem Hình 8.2)Cái này bắt buộc đi từ 0 đến 10



Quên điều gì?

Đôi lúc người dùng nhập dữ liệu trong form, họ quên điền đầy đủ thông tin. Nếu điều này xảy ra, hệ thống phải được lồng những dữ liệu sai hoặc không đầy đủ để không

làm ảnh hưởng đến cơ sở dữ liệu. Bạn phải làm thế nào để hệ thống phản ứng lại những lỗi như vậy mà không làm hỏng cơ sở dữ liệu.

Ví dụ: Thêm vào tập lệnh để người dùng nhập

Trong ví dụ này bạn chắc rằng tập lệnh có thể sửa lại khi người dùng nhập sai.

1. Copy đoạn mã trong chương 6 vào thư mục mới, mở tập lệnh movie.php và sửa như những dòng hướng dẫn sau:

```
<?php
$link = mysql_connect("localhost", "root", "");
or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
or die ( mysql_error());
$peoplesql = "SELECT * FROM people";
$result = mysql_query($peoplesql)
or die("Invalid query: " . mysql_error());
while ($row = mysql_fetch_array($result))
{
    $people[$row['people_id']] = $row['people_fullname'];
}
switch ($_GET['action'])
{
    case "edit":
        $moviesql = "SELECT * FROM movie " .
        "WHERE movie_id = " . $_GET['id'] .
        $result = mysql_query($moviesql)
or die("Invalid query: " . mysql_error());
        $row = mysql_fetch_array($result);
        $movie_name = $row['movie_name'];
        $movie_type = $row['movie_type'];
        $movie_year = $row['movie_year'];
```

```

        $movie_leadactor = $row['movie_leadactor'];
        $movie_director = $row['movie_director'];
        break;
    default:
        $movie_name = "";
        $movie_type = "";
        $movie_year = "";
        $movie_leadactor = "";
        $movie_director = "";
        break;
    }
?>
<html>
<head>
<title><?php echo $_GET['action']; ?> movie</title>
<style type="text/css">
TD{color:#353535;font-family:verdana}
TH{color:#FFFFFF;font-family:verdana;background-
color:#336699}
</style>
</head>
<body>
<form action="commit.php?action=<?php
echo $_GET['action']; ?>&type=movie&id=<?php
if (isset($_GET['id'])) { echo $_GET['id']; } ?>"
method="post">
<?php
    if (!empty($_GET['error']))
    {

```

```

        echo "<div align=\"center\" “ .
        “style=\"color:#FFFFFF;background-color:#FF0000;
        “font-weight:bold\">”
        nl2br(urldecode($_GET['error'])) .
        “</div><br />”;
    }
?>

```

```

<table border="0" width="750" cellspacing="1"
cellpadding="3" bgcolor="#353535"
align="center">
<tr>
<td bgcolor="#FFFFFF" width="30%">Movie
Name</td>
<td bgcolor="#FFFFFF" width="70%">
<input type="text" name="movie_name"
value="<?php echo $movie_name?>">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Type</td>
<td bgcolor="#FFFFFF">
<select id="game" name="movie_type"
style="width:150px">
<option value="" selected>Select a type...</option>

```

```

<?php
$sql = "SELECT movietype_id, movietype_label “ .
“FROM movietype ORDER BY movietype_label”;
$result = mysql_query($sql)
or die("<font color=\"#FF0000\">Query

```

```

Error" . mysql_error());
while ($row = mysql_fetch_array($result))
{
    if ($row['movietype_id'] == $movie_type) {
        $selected = " selected";
    } else
    {
        $selected = "";
    }
    echo '<option value="' . $row['movietype_id'] . "' . "' .
    $selected .
    '>' . $row['movietype_label'] . "</option>\r\n";
}
?>

</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Year</td>
<td bgcolor="#FFFFFF">
<select name="movie_year">
<option value="" selected>Select a year...</option>
<?php
for ($year=date("Y"); $year >= 1970 ; $year--)
{
    if ($year == $movie_year)
    {
        $selected = " selected";
    } else

```

```

        {
            $selected = "";
        }
    ?>
<option value="<?php echo $year; ?>"
<?php echo $selected; ?><?php echo $year; ?></option>
<?php
    }
?>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Lead Actor</td>
<td bgcolor="#FFFFFF">
<select name="movie_leadactor">
<option value="" selected>Select an actor...</option>
<?php
    foreach ($people as $people_id =>
        $people_fullname)
    {
        if ($people_id == $movie_leadactor)
        {
            $selected = " selected";
        } else {
            $selected = "";
        }
    }
?>
<option value="<?php echo $people_id; ?>"

```



```

<?php echo $selected; ?><?php echo $people_fullname;
?></option>
<?php
    }
?>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Director</td>
<td bgcolor="#FFFFFF">
<select name="movie_director">
<option value="" selected>Select a director...</option>
<?php
    foreach ($people as $people_id =>
        $people_fullname)
    {
        if ($people_id == $movie_director)
        {
            $selected = " selected";
        } else {
            $selected = "";
        }
    }
?>
<option value="<?php echo $people_id; ?>"
<?php echo $selected; ?><?php echo $people_fullname;
?></option>
<?php
    }

```

```
?>
</select>
</td>
</tr>
<tr>
<td bgcolor=""#FFFFFF" colspan=""2" align=""center">
<input type=""submit" name=""submit"
value=""<?php echo $_GET['action']; ?>">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Chương 8 : NGƯỜI DÙNG NHẬP CÓ HIỆU LỰC

Nếu bạn chấp nhận cho người dùng được nhập vào trang web, bạn phải chuẩn bị gặp những lỗi. Có thể chỉ là lỗi đơn giản hoặc những lỗi ảnh hưởng trang web của bạn. Lỗi chung nhất thường gặp là lỗi thuật in cơ bản, lỗi định dạng (ví dụ cho một năm trong một ngày). Những lỗi như người dùng không cung cấp địa chỉ email, hoặc người dùng xâm nhập vào cơ sở dữ liệu của bạn. Tập lệnh cần có phần điều khiển nhập, bằng việc xác định dữ liệu xấu và trả về trang thông báo lỗi cho người dùng. Bao gồm:

- ▶ Làm cho có hiệu lực những giá trị chuỗi đơn giản
- ▶ Làm cho có hiệu lực những giá trị nguyên
- ▶ Làm cho có hiệu lực văn bản nhập vào đã định dạng

Quên điều gì (Tiếp theo)

2. Lưu file với tên movie.php và đưa đoạn mã mới vào thư mục làm việc của bạn
3. Mở tập lệnh commit.php và chỉnh sửa như những dòng hướng dẫn sau:

```
<?php
// COMMIT ADD AND EDITS
$error = "";
$link = mysql_connect("localhost", "root", "")
or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
or die ( mysql_error());
switch ($_GET['action'])
{
    case "edit":
        switch ($_GET['type'])
```

```

{
    case "people":
        $sql = "UPDATE people SET " .
            "people_fullname = " .
            $_POST['people_fullname'] .
            " WHERE people_id = " . $_GET['id'] . """;
        break;
    case "movie":
        $movie_name = trim($_POST['movie_name']);
        if (empty($movie_name))
        {
            $error .=
                "Please+enter+a+movie+name%21%0D%0A";
        }
        if (empty($_POST['movie_type']))
        {
            $error .=
                "Please+select+a+movie+type%21%0D%0A";
        }
        if (empty($_POST['movie_year']))
        {
            $error .=
                "Please+select+a+movie+year%21%0D%0A";
        }
        if (empty($error))
        {
            $sql = "UPDATE movie SET " .
                "movie_name = " . $_POST['movie_name'] .
                " , " .

```

```

        "movie_year = " . $_POST['movie_year'] .
        " " .
        "movie_type = " . $_POST['movie_type'] .
        " " .
        "movie_leadactor = " .
        $_POST['movie_leadactor'] . " " .
        "movie_director = " .
        $_POST['movie_director'] . " " .
        "WHERE movie_id = " . $_GET['id'] . """;
    } else {
        header("location:movie.php?action=edit&error
        =");
        $error .= "&id=" . $_GET['id'];
    }
    break;
}
break;
case "add":
switch ($_GET['type'])
{
    case "people":
        $sql = "INSERT INTO people (people_fullname) " .
        "VALUES (" . $_POST['people_fullname'] . ")";
        break;
    case "movie":
        $movie_name = trim($_POST['movie_name']);
        if (empty($movie_name))
        {
            $error .=

```

```

        "Please+enter+a+movie+name%21%0D%0A";
    }
    if (empty($_POST['movie_type']))
    {
        $error .=
            "Please+select+a+movie+type%21%0D%0A";
    }
    if (empty($_POST['movie_year']))
    {
        $error .=
            "Please+select+a+movie+year%21%0D%0A";
    }
    if (empty($error))
    {
        $sql = "INSERT INTO movie
        (movie_name,movie_year," .
        "movie_type,movie_leadactor,movie_director)
        " .
        "VALUES (" . $_POST['movie_name'] . "," .
        " " . $_POST['movie_year'] . "," .
        " " . $_POST['movie_type'] . "," .
        " " . $_POST['movie_leadactor'] . "," .
        " " . $_POST['movie_director'] . ")";
    } else {
        header("location:movie.php?action=add&error="
        " " . $error);
    }
    break;
}

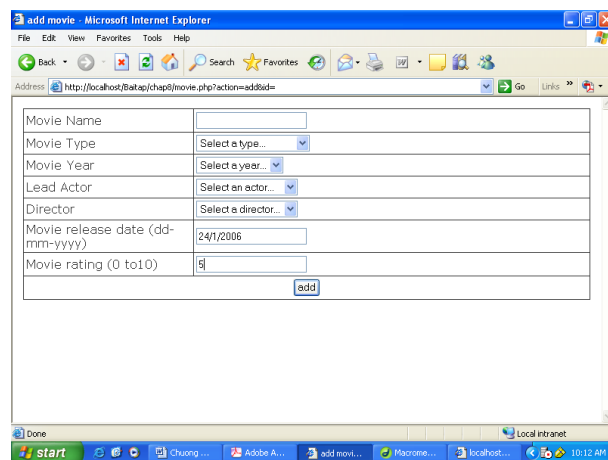
```

```

        break;
    }
    if (isset($sql) && !empty($sql))
    {
        echo "<!--". $sql. "-->";
        $result = mysql_query($sql)
        or die("Invalid query: " . mysql_error());
    }
?>
<p align="center" style="color:#FF0000">
Done. <a href="index.php">Index</a>
</p>
<?php
}
?>

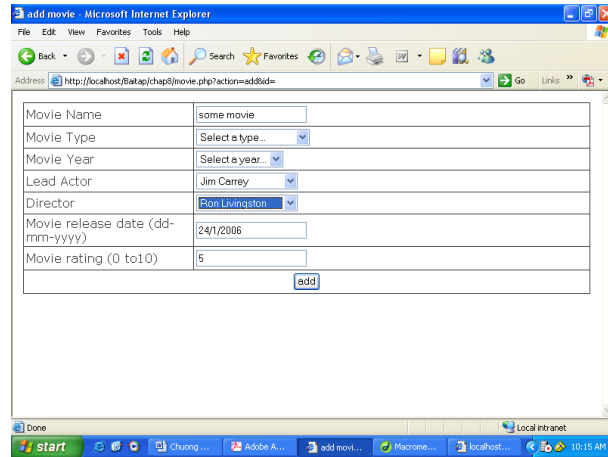
```

4. Lưu với tên commit.php và đưa lên server
5. Mở trình soạn thảo và vào trang <http://localhost/chapter8/index.php> và thêm vào một movie không tên, như **Hình 8.3**.



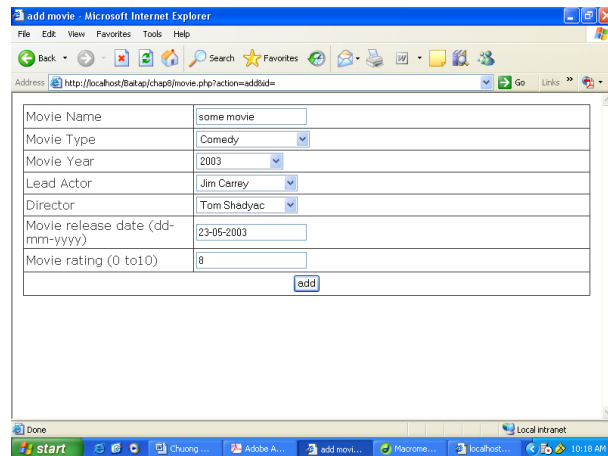
Hình 8.3

6. Nhập vào một movie mới mà không đặt năm và kiểu movie (Như **Hình 8.4**)



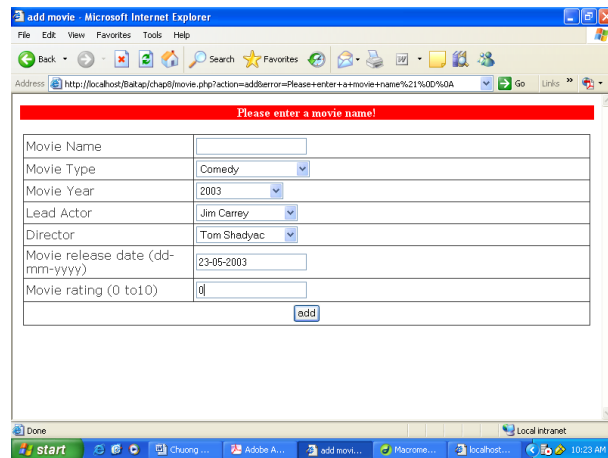
Hình 8.4

7. Soạn thảo một movie từ chỉ số và xóa form tên, submit(Như Hình 8.5)



Hình 8.5

8. Chú ý thông báo lỗi trong việc điền form(Như Hình 8.6)



Hình 8.6

Cách thức hoạt động

Khi form chuyển thông tin đến tập lệnh, dữ liệu sẽ được xác nhận. Bạn sử dụng một phương pháp đơn giản: hàm empty() trả về true nếu chuỗi rỗng, trả về false nếu không. Để đảm bảo rằng người dùng không trình bày form với một không gian đơn giản trong vùng tên movie, bạn dùng trim() trong nội dung của vùng để loại trừ bất kỳ dẫn đến nào hoặc kéo dài chuỗi(một vài người làm cho công việc trở nên khó khăn).

Cùng lúc đó mà có lỗi được phát hiện thì bạn đưa ra thông báo lỗi, dùng biến \$error(tập hợp tất cả các lỗi)

```
if (empty($movie_name))
{
    $error = "Please+enter+a+movie+name%21%0D%0A";
}
```

Một lần nữa bạn thấy rằng có một lỗi xuất hiện, bạn gửi đến người dùng một thông báo lỗi. Thông báo lỗi được URL mã hóa để đảm bảo nó được chuyển đến tập lệnh move.php mà không bị hỏng

```
if (empty($error))
{
    ...
} else
{
    header("location:movie.php?action=add&error=" . $error);
}
```

Một lần gửi đến form, hệ thống cần hiển thị giải mã thông báo lỗi

```
<?
if (!empty($_GET['error']))
{
    echo "<div align=\"center\" \" .
        \"style=\"color:#FFFFFF;background-color:#FF0000;\" .
        \"font-weight:bold\">\" . nl2br(urldecode($_GET['error'])).
```

```
“</div><br />”;  
}  
?>
```

Điều này trình bày thông báo quá màu mè mà người dùng sẽ không nhớ. Chính sự cập nhật được thực hiện vào lúc cuối của đoạn mã, cùng với tất cả các điều khiển và đưa ra thông báo bạn cần

```
if (isset($sql) && !empty($sql))  
{  
    echo “<!--”.$sql.”-->”;  
    $result = mysql_query($sql)  
    or die(“Invalid query: “ . mysql_error());  
    ?>  
    <p align=“center” style=“color:#FF0000”>  
    Done. <a href=“index.php”>Index</a>  
    </p>  
    <?php  
}
```

Nếu biến \$sql không được thiết lập trước (mà có thể xảy ra nếu được gọi ra khỏi nội dung), đoạn mã sẽ không thực thi và không làm gì cả. (Chú ý nó sẽ là bài tập tốt cho bạn để lập một phản hồi cho biến cố này, như một thông điệp hoặc một đăng ký lỗi trong cơ sở dữ liệu)

Chương 8 : NGƯỜI DÙNG NHẬP CÓ HIỆU LỰC

Nếu bạn chấp nhận cho người dùng được nhập vào trang web, bạn phải chuẩn bị gặp những lỗi. Có thể chỉ là lỗi đơn giản hoặc những lỗi ảnh hưởng trang web của bạn. Lỗi chung nhất thường gặp là lỗi thuật in cơ bản, lỗi định dạng (ví dụ cho một năm trong một ngày). Những lỗi như người dùng không cung cấp địa chỉ email, hoặc người dùng xâm nhập vào cơ sở dữ liệu của bạn. Tập lệnh cần có phần điều khiển nhập, bằng việc xác định dữ liệu xấu và trả về trang thông báo lỗi cho người dùng. Bao gồm:

- ▶ Làm cho có hiệu lực những giá trị chuỗi đơn giản
- ▶ Làm cho có hiệu lực những giá trị nguyên
- ▶ Làm cho có hiệu lực văn bản nhập vào đã định dạng

Việc kiểm tra những kiểu lỗi

Việc kiểm tra lỗi ngày tháng hoặc những dữ liệu định dạng khác là một yêu cầu cần thiết trong hệ thống bởi vì người dùng không phải lúc nào cũng được hướng dẫn nhập dữ liệu. Bạn phải luôn kiểm tra dữ liệu mà người dùng nhập vào nếu bạn định dạng hoặc thiết lập giá trị đặc biệt.

Tại điểm này, bạn cần những biểu thức chung, những biểu thức này cho phép bạn định nghĩa một mẫu và kiểm tra xem có thể áp dụng vào dữ liệu hay không. Nó giúp bạn kiểm tra ngày tháng, số bảo mật và bất kỳ một dữ liệu nào mà được chú ý việc định dạng (nó giúp chắc chắn trong việc định dạng vùng nguồn).

Ví dụ về kiểm tra ngày tháng và số

Trong bài tập này bạn sẽ thay đổi vài trang để bạn kiểm tra việc định dạng ngày tháng của người dùng.

1. Mở file movie.php và thay đổi nó như sau (thay đổi được làm nổi bật)

```

<?php
$link = mysql_connect("localhost", "bp5am", "bp5ampass")
or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
or die(mysql_error());
$peoplesql = "SELECT * FROM people";
$result = mysql_query($peoplesql)
or die("Invalid query: " . mysql_error());
while ($row = mysql_fetch_array($result))
{
    $people[$row['people_id']] = $row['people_fullname'];
}
switch ($_GET['action'])
{
    case "edit":
        $moviesql = "SELECT * FROM movie " .
        "WHERE movie_id = " . $_GET['id'] . " ";
        $result = mysql_query($moviesql)
or die("Invalid query: " . mysql_error());
        $row = mysql_fetch_array($result);
        $movie_name = $row['movie_name'];
        $movie_type = $row['movie_type'];
        $movie_year = $row['movie_year'];
        $movie_release = $row['movie_release'];
        $movie_leadactor = $row['movie_leadactor'];
        $movie_director = $row['movie_director'];
        $movie_rating = $row['movie_rating'];
        break;
    default:

```

```
$movie_name = "";
$movie_type = "";
$movie_year = "";
$movie_release = time();
$movie_leadactor = "";
$movie_director = "";
$movie_rating = "5";
break;
}
?>
<html>
<head>
<title><?php echo $_GET['action']; ?> movie</title>
<style type="text/css">
TD
{
    color:#353535;font-family:verdana
}
TH
{
    color:#FFFFFF;font-family:verdana;background-color:#336699
}
</style>
</head>
<body>
<form action="commit.php?action=<?php
echo $_GET['action']; ?>&type=movie&id=<?php
if (isset($_GET['id']))
{
```

```

        echo $_GET['id'];
    }
    ?>" method="post">
<?php
if (!empty($_GET['error']))
{
    echo "<div align=\"center\" “ .
        “style=\"color:#FFFFFF;background-color:#FF0000;” .
        “font-weight:bold\">” . nl2br(urldecode($_GET['error'])) .
        “</div><br />”;
}
?>
<table border="0" width="750" cellspacing="1"
cellpadding="3" bgcolor="#353535" align="center">
<tr>
<td bgcolor="#FFFFFF" width="30%">Movie Name</td>
<td bgcolor="#FFFFFF" width="70%">
<input type="text" name="movie_name"
value="<?php echo $movie_name?>">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Movie Type</td>
<td bgcolor="#FFFFFF">
<select id="game" name="movie_type" style="width:150px">
<option value="" selected>Select a type...</option>
<?php
$sql = "SELECT movietype_id, movietype_label “ .
“FROM movietype ORDER BY movietype_label”;

```

```

$result = mysql_query($sql)
or die("<font color=#FF0000>Query Error</font>". mysql_error());
while ($row = mysql_fetch_array($result))
{
    if ($row['movietype_id'] == $movie_type)
    {
        $selected = " selected";
    }
    else
    {
        $selected = "";
    }
    echo '<option value="' . $row['movietype_id'] . "' . $selected .
    '>' . $row['movietype_label'] . "</option>\r\n";
}
?>
</select>
</td>
</tr>
<tr>
<td bgcolor=#FFFFFF>Movie Year</td>
<td bgcolor=#FFFFFF>
<select name="movie_year">
<option value="" selected>Select a year...</option>
<?php
for ($year=date("Y"); $year >= 1970 ;$year--)
{
    if ($year == $movie_year)
    {

```

```

        $selected = " selected";
    }
    else
    {
        $selected = "";
    }
?>
<option value="<?php echo $year; ?>"
<?php echo $selected; ?><?php echo $year; ?></option>
<?php
}
?>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Lead Actor</td>
<td bgcolor="#FFFFFF">
<select name="movie_leadactor">
<option value="" selected>Select an actor...</option>
<?php
foreach ($people as $people_id => $people_fullname)
{
    if ($people_id == $movie_leadactor)
    {
        $selected = " selected";
    }
    else
    {

```



```

                $selected = "";
            }
        ?>
        <option value="<?php echo $people_id; ?>"
        <?php echo $selected; ?><?php echo $people_fullname;
        ?></option>
    <?php
}
?>
</selected>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF">Director</td>
<td bgcolor="#FFFFFF">
<select name="movie_director">
<option value="" selected>Select a director...</option>
<?php
foreach ($people as $people_id => $people_fullname)
{
    if ($people_id == $movie_director)
    {
        $selected = " selected";
    }
    else
    {
        $selected = "";
    }
}
?>

```

```

        <option value="<?php echo $people_id; ?>"
        <?php echo $selected; ?><?php echo $people_fullname;
        ?></option>
        <?php
    }
    ?>
</select>
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" width="30%">
Movie release date (dd-mm-yyyy)
</td>
<td bgcolor="#FFFFFF" width="70%">
<input type="text" name="movie_release"
value="<?php echo date("d-m-Y", $movie_release); ?>">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" width="30%">
Movie rating (0 to 10)
</td>
<td bgcolor="#FFFFFF" width="70%">
<input type="text" name="movie_rating"
value="<?php echo $movie_rating; ?>">
</td>
</tr>
<tr>
<td bgcolor="#FFFFFF" colspan="" align="center">

```

```
<input type="submit" name="submit"
value="<?php echo $_GET['action']; ?>">
</td>
</tr>
</table>
</form>
</body>
</html>
```

Chương 8 : NGƯỜI DÙNG NHẬP CÓ HIỆU LỰC

Nếu bạn chấp nhận cho người dùng được nhập vào trang web, bạn phải chuẩn bị gặp những lỗi. Có thể chỉ là lỗi đơn giản hoặc những lỗi ảnh hưởng trang web của bạn. Lỗi chung nhất thường gặp là lỗi thuật in cơ bản, lỗi định dạng (ví dụ cho một năm trong một ngày). Những lỗi như người dùng không cung cấp địa chỉ email, hoặc người dùng xâm nhập vào cơ sở dữ liệu của bạn. Tập lệnh cần có phần điều khiển nhập, bằng việc xác định dữ liệu xấu và trả về trang thông báo lỗi cho người dùng. Bao gồm:

- ▶ Làm cho có hiệu lực những giá trị chuỗi đơn giản
- ▶ Làm cho có hiệu lực những giá trị nguyên
- ▶ Làm cho có hiệu lực văn bản nhập vào đã định dạng

Việc kiểm tra những kiểu lỗi (tiếp theo)

2. Bây giờ file commit.php và thay đổi như sau:

```
<?php
// COMMIT ADD AND EDITS
$error = '';
$link = mysql_connect("localhost", "root", "")
or die("Could not connect: " . mysql_error());
mysql_select_db('moviesite', $link)
or die ( mysql_error());
switch ($_GET['action'])
{
    case "edit":
        switch ($_GET['type'])
```

```

{
case "people":
$ssql = "UPDATE people SET " .
"people_fullname = " . $_POST['people_fullname'] .
" WHERE people_id = " . $_GET['id'] . """;
break;
case "movie":
$movie_rating = trim($_POST['movie_rating']);
if (!is_numeric($movie_rating))
{
$error .= "Please+enter+a+numeric+rating+%21%0D%0A";
}
else
{
if ($movie_rating < 0 || $movie_rating > 10)
{
$error .= "Please+enter+a+rating+" .
"between+0+and+10+%21%0D%0A";
}
}
if (!ereg("[0-9]{2})-[0-9]{2}-[0-9]{4}"),
$_POST['movie_release'] ,
$reldatepart))
{
$error .= "Please+enter+a+date+" .
"with+the+dd-mm-yyyy+format+%21%0D%0A";
}
else
{

```

```

$movie_release = @mktime(0, 0, 0, $reldatepart['2'],
$reldatepart['1'],
$reldatepart['3']);
if ($movie_release == '-1')
{
    $error .= "Please+enter+a+real+date+" .
    "with+the+dd-mm-yyyy+format%21%0D%0A";
}
}
$movie_name = trim($_POST['movie_name']);
if (empty($movie_name))
{
    $error .= "Please+enter+a+movie+name%21%0D%0A";
}
if (empty($_POST['movie_type']))
{
    $error .= "Please+select+a+movie+type%21%0D%0A";
}
if (empty($_POST['movie_year']))
{
    $error .= "Please+select+a+movie+year%21%0D%0A";
}
if (empty($error) )
{
    $sql = "UPDATE movie SET " .
    "movie_name = " . $_POST['movie_name'] . " , " .
    "movie_year = " . $_POST['movie_year'] . " , " .
    "movie_release = '$movie_release' , " .
    "movie_type = " . $_POST['movie_type'] . " , " .

```

```

        "movie_leadactor = "" . $_POST['movie_leadactor'] . "" , " .
        "movie_director = "" . $_POST['movie_director'] . "" , " .
        "movie_rating = '$movie_rating'" .
        "WHERE movie_id = "" . $_GET['id'] . """;
    }
else
{
    header("location:movie.php?action=edit&error=" .
    $error . "&id=" . $_GET['id']);
}
break;
}
break;
case "add":
switch ($_GET['type'])
{
    case "people":
        $sql = "INSERT INTO people (people_fullname) " .
        "VALUES ("" . $_POST['people_fullname'] . "")";
        break;
    case "movie":
        $movie_rating = trim($_POST['movie_rating']);
        if (!is_numeric($movie_rating))
        {
            $error .= "Please+enter+a+numeric+rating+%21%0D%0A";
        }
        else
        {
            if ($movie_rating < 0 || $movie_rating > 10)

```

```

        {
            $error .= "Please+enter+a+rating+" .
                "between+0+and+10%21%0D%0A";
        }
    }
    $movie_release = trim($_POST['movie_release']);
    if (!ereg("[0-9]{2})-([0-9]{2})-([0-9]{4})",
    $movie_release,
    $reldatepart) || empty($movie_release))
    {
        $error .= "Please+enter+a+date+" .
            "with+the+dd-mm-yyyy+format%21%0D%0A";
    }
    else
    {
        $movie_release = @mktime(0, 0, 0, $reldatepart['2'],
        $reldatepart['1'],
        $reldatepart['3']);
        if ($movie_release == '-1')
        {
            $error .= "Please+enter+a+real+date+" .
                "with+the+dd-mm-yyyy+format%21%0D%0A";
        }
    }
    $movie_name = trim($row['movie_name']);
    if (empty($movie_name))
    {
        $error .= "Please+enter+a+movie+name%21%0D%0A";
    }
}

```



```

if (empty($_POST['movie_type']))
{
    $error .= "Please select a movie type%21%0D%0A";
}
if (empty($_POST['movie_year']))
{
    $error .= "Please select a movie year%21%0D%0A";
}
if (empty($error))
{
    $sql = "INSERT INTO movie (movie_name,movie_year," .
    "movie_release,movie_type,movie_leadactor," .
    "movie_director,movie_rating) " .
    "VALUES (" . $_POST['movie_name'] . "," .
    " . $_POST['movie_year'] . "," .
    "$movie_release'," .
    " . $_POST['movie_type'] . "," .
    " . $_POST['movie_leadactor'] . "," .
    " . $_POST['movie_director'] . "," .
    "$movie_rating)";
}
else
{
    header("location:movie.php?action=add&error=" . $error);
}
break;
}
break;
}

```

```

if (isset($sql) && !empty($sql))
{
    echo "<!--". $sql. "-->";
    $result = mysql_query($sql)
    or die("Invalid query: " . mysql_error());
    ?>
    <p align="center" style="color:#FF0000">
    Done. <a href="index.php">Index</a>
    </p>
    <?php
}
?>

```

3. Hãy lưu những file này lại, mở trình duyệt bạn sẽ thấy index.

4. Click bất kỳ một movie nào, thử nhập 2003-10-10 vào phiên bản ngày tháng. Bạn sẽ thấy một form đẹp, rõ ràng. (Như Hình 8.7)

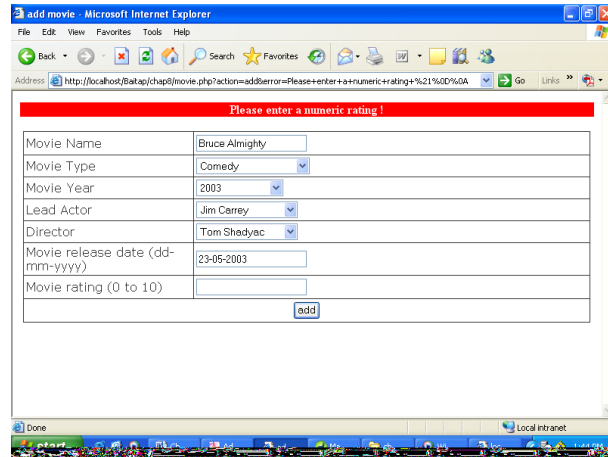
The screenshot shows a web browser window with a form for adding a movie. The form contains the following fields and values:

Movie Name	Bruce Almighty
Movie Type	Comedy
Movie Year	2003
Lead Actor	Jim Carrey
Director	Tom Shadyac
Movie release date (dd-mm-yyyy)	2003-10-10
Movie rating (0 to 10)	

There is an 'add' button at the bottom right of the form. A red error message is displayed at the top of the form area: "Please enter a numeric rating! Please enter a date with the dd-mm-yyyy format!".

Hình 8.7

5. Thử nhập vào giá trị chữ số bên trong (như Hình 8.8) để dàng có được hộp kéo xuống nhưng là vùng văn bản cho mục đích trong bài tập.



Hình 8.8

Nếu nhập vào giá trị mà không có trong phạm vi từ 0 đến 10, nó sẽ từ chối. (chú ý những phần thập phân thì không có trong mã này nên cũng bị mất)

Nó hoạt động thế nào?

Trước hết, hãy nhìn vào những hàm làm cho có hiệu lực. Trong commit.php, bạn sử dụng hàm `is_numeric()`. Hàm này trả về giá trị BOOL, nếu đúng trả về TRUE, nếu sai trả về FALSE. Có nhiều hàm có sẵn, bạn gồm:

`is_string` kiểm tra nếu nó dạng chuỗi

`is_bool` kiểm tra nếu dạng BOOL(TRUE, FALSE, 0, 1)

`is_array` cho ta biết nếu biến chứa mảng

`is_object` xác định nếu biến chứa một đối tượng (khi bạn cố gắng dùng code hướng đối tượng trong PHP, nó rất có ích)

Những hàm này là tài liệu PHP trong www.php.net/variable Trong trường hợp này sử dụng `is_numeric` cho phép bạn biết chắc rằng người dùng đã nhập đúng giá trị số.

```

$movie_rating = trim($_POST['movie_rating']);
if (!is_numeric($movie_rating))
{
    $error .= "Please+enter+a+numeric+rating+%21%0D%0A";
}
else
{

```

```

if ($movie_rating < 0 || $movie_rating > 10)
{
    $error .= "Please+enter+a+rating+" .
    "between+0+and+10%21%0D%0A";
}
}

```

Đoạn mã dùng hàm trim() để kiểm tra giá trị. Nếu giá trị là số trong khoảng từ 0 đến 10 thì cho thấy, nếu không thì đưa ra thông báo lỗi.

Làm có hiệu lực ngày tháng thì hầu như đơn giản để hiểu, nếu bạn biết những biểu thức chung. Ở đây là:

```

$movie_release = trim($_POST['movie_release']);
if (!ereg("[0-9]{2})-(0-9){2}-(0-9){4}"),
$movie_release,
$reldatepart) || empty($movie_release))
{
    $error .= "Please+enter+a+date+" .
    "with+the+dd-mm-yyyy+format%21%0D%0A";
}
else
{
    $movie_release = @mktime(0, 0, 0, $reldatepart['2'],
    $reldatepart['1'],
    $reldatepart['3']);
    if ($movie_release == '-1')
    {
        $error .= "Please+enter+a+real+date+" .
        "with+the+dd-mm-yyyy+format%21%0D%0A";
    }
}

```

}

Như bạn thấy trong bài tập đầu tiên của chương này sử dụng hàm trim() để kiểm tra tất cả những không gian trong chuỗi để chắc rằng người dùng nhập chỉ một không gian.

Bạn có thể tìm những hàm tại www.php.net/strings . Bạn có thể tìm những hàm có ích khác trong trang web này.

Những mệnh đề tiếp theo chứa 2 điều kiện. Một là những biểu thức chung như ([0-9] { 2 })-([0-9] { 2 })-([0-9] { 4 }). Điều này bạn kiểm tra những số trong khoảng từ 0 đến 9 với 2 biến cố. Ví dụ: nếu không là 2. Nếu [0-9] { 4 } chỉ khác nhau ở chỗ là có 4 ký số trong số, xác định phân năm trong ngày.(hai phần đầu là ngày và tháng)

Vì thế, trong Tiếng Anh có nghĩa là “Tôi muốn chuỗi bắt đầu với một số hai ký số, đi theo bởi một dấu nối và sau đó nhóm với 2 ký số, rồi đến dấu nối, và cuối cùng nhóm với 4 ký số.”

```
if (!ereg("([0-9]{2})-([0-9]{2})-([0-9]{4})",
$movie_release,
$reldatepart) || empty( $movie_release ))
{
...
}
```

Đây là xác định của biểu thức bình thường. Nếu chuỗi thỏa điều kiện của bạn, bạn sẽ chia 3 phần khác nhau, mỗi phần giới hạn bởi dấu ngoặc.

Việc cắt này được thực hiện bởi hàm ereg(). Nếu biến \$movie_release thỏa với mẫu, ereg sẽ cắt chuỗi thành từng phần, lưu trữ mỗi phần như từng phần tử của mảng \$reldatepart.

Để đọc tài liệu PHP về những biểu thức bình thường chúng ta tham khảo trang web www.php.net/regex và hỏi ý kiến thầy để hiểu hơn về những biểu thức bình thường. Nếu người dùng nhập 02-03-2004, sẽ được mảng như sau:

Array

(

```
[0] => 02-03-2004
[1] => 02
[2] => 03
[3] => 2004
)
```

Như bạn thấy, chỉ số đầu tiên giữ toàn bộ chuỗi, những số còn lại lưu giữ mỗi bộ phận của chuỗi, giới hạn bởi dấu ngoặc bên ngoài.

Bây giờ bạn có ngày được định dạng dễ hiểu, bạn có thể chuyển nó thành thời gian tạm bằng việc sử dụng hàm mktime(), cho phép bạn tạo ra thời gian tạm từ những phần ngày, tháng. Nó cũng là một hàm rất có ích cho việc vận dụng ngày.

```
$movie_release = mktime(0, 0, 0, $reldatepart['2'],
    $reldatepart['1'],
    $reldatepart['3']);
```

Mã này lưu giữ thời gian tạm từ thông tin ngày, tháng, năm đến hệ thống bên trong biến \$movie_release. Định dạng này là int mktime(int hour, int minute, int second, int month, int day, int year). Giá trị trả về là số giây giữa January 1, 1970 và ngày chỉ định.

Nếu mktime thất bại khi tạo thời gian tạm từ ngày, tháng bạn cho, nó sẽ trả về -1. Điều này xảy ra khi nhập sai, mặc dù nó phù hợp với biểu thức bình thường ví dụ như:99-99 9999 sẽ qua sự kiểm tra của biểu thức bình thường nhưng đây không là ngày tháng hợp lệ. Để chắc rằng phải là ngày hợp lệ, bạn kiểm tra giá trị trả về từ mktime và trả lời tương ứng

```
if ($movie_release == '-1')
{
    $error .= "Please+enter+a+real+date+" .
        "with+the+dd-mm-yyyy+format%21%0D%0A";
}
```

Trong trường hợp này, một ngày không hợp lệ sẽ đưa ra một lỗi hỏi về ngày tháng hợp lệ. Ở đây có một kỹ xảo khác: bạn có thể thi hành thời gian tạm sử dụng SQL. Nhiều thứ mà PHP thao tác trực tiếp từ SQL, như:

```
if (!ereg("[0-9]{2})-([0-9]{2})-([0-9]{4})",
    $movie_release,
    $reldatepart) || empty($movie_release))
{
    ...
}
$reldate = $reldatepart['3'] . "-" .
    $reldatepart['2'] . "-" .
    $reldatepart['1'] . " 00:00:00";
$sql = "INSERT INTO movie (movie_release) " .
    "VALUES (UNIX_TIMESTAMP('$reldate'))";
```

Trong đoạn mã này, SQL phát sinh thời gian tạm. Hàm UNIX_TIMESTAMP() SQL định dạng YYYY-MM-DD HH:MM:SS (2004-12-05 02:05:00) và tạo thời gian tạm từ đó. Trong mã, bạn bắt buộc tạo thời gian tạm tại thời điểm 00:00

Tóm tắt

Việc làm cho người dùng khi nhập vào có hiệu lực là ta sẵn sàng xử lý những lỗi gặp phải, những lỗi đó có thể cố ý hoặc vô tình, nó xảy ra hằng ngày. Những biểu thức bình thường giúp bạn rất nhiều

Chương 9 : ĐIỀU KHIỂN VÀ TRÁNH NHỮNG LỖI

Sau khi viết mã lệnh, người mới lập trình sẽ gặp không ít khó khăn trong việc tìm ra lỗi và chỉnh sửa lỗi. PHP hỗ trợ đầy đủ tính năng lập trình giao diện API (Applications Programming Interface) có thể tìm ra lỗi và khắc phục những lỗi ấy. Ngoài PHP thì Apache Web Server cũng hỗ trợ người lập trình tìm ra lỗi và khắc phục lỗi.

Tiếp xúc với lỗi Apache Web Server như thế nào?

Apache có một thư mục Lỗi tài liệu, bạn có thể định cấu hình trong file httpd.conf để tạo ra những trang thông báo lỗi với PHP, vì thế người thăm trang web không thấy buồn chán. PHP tạo ra trang web thông báo lỗi thân thiện cho người dùng.

Không giống những trang thông báo lỗi như PHP, trang của Apache được sử dụng nhiều cho những trang đã mất.

Chỉ thị lỗi của Apache

Ví dụ về việc sử dụng phương pháp Apache.

Trước hết cần thay đổi trong thư mục httpd.conf để cho phép tạo ra một trang thông báo lỗi, Apache thường được mặc định bên trong những trang lỗi. Nhưng bạn muốn có một thông báo như ý muốn khi có lỗi xuất hiện.

Bạn cần có những thay đổi sau:

1. Mở file httpd.conf, những dòng 750 hoặc xa hơn, tìm vài dòng giống như sau:

```
# Customizable error responses come in three flavors:
```

```
# 1) plain text 2) local redirects 3) external redirects
```

```
#
```

```
# Some examples:
```

```
#ErrorDocument 500 "The server made a boo boo."
```

```
#ErrorDocument 404 /missing.html
```

```
#ErrorDocument 404 "/cgi-bin/missing_handler.pl"
```

```
#ErrorDocument 402 http://www.example.com/subscription\_info.html
```


2. Thay đổi thông tin cho phép sau đó khởi động lại Apache

Customizable error responses come in three flavors:

1) plain text 2) local redirects 3) external redirects

#

Some examples:

```
ErrorDocument 400 /error.php?400
```

```
ErrorDocument 401 /error.php?401
```

```
ErrorDocument 403 /error.php?403
```

```
ErrorDocument 404 /error.php?404
```

```
1ErrorDocument 500 /error.php?500
```

Nó hoạt động thế nào?

Bạn vừa soạn thảo cấu hình Apache để giúp bạn thi hành lỗi, bạn có thể gửi đến người dùng những trang lỗi phụ thuộc vào những lỗi mà người dùng gặp phải

Có nhiều lỗi trong tài liệu nhưng chúng ta sẽ tập trung vào lỗi mà thường gặp hằng ngày.

❑ **400:** Lời thỉnh cầu không tốt

❑ **401:** Yêu cầu được cho phép

❑ **403:** Không được phép

❑ **404:** Không tìm thấy

❑ **500:** Lỗi từ server

Ví dụ: Hiển thị lỗi

Để hiển thị thông báo lỗi, bạn làm những bước sau:

1. Mở trình duyệt và lưu với tên error.php

2. Nhập những dòng sau:

```
<?php
```

```
$error_no = $_SERVER['QUERY_STRING'];
```

```

switch ($error_no)
{
    case 400:
        $error_output = "<h1>&quot;Bad Request&quot;
Error Page - " .
        "(Error Code 400)</h1>";
        $error_output .= "The browser has made a Bad
Request<br>";
        $error_output      .=      "<a
href=\"mailto:sysadmin@localhost.com\">" .
        "Contact</a> the system administrator";
        $error_output .= " if you feel this to be in error";
        break;

    case 401:
        $error_output = "<h1>&quot;Authorization
Required&quot; " .
        "Error Page - (Error Code 401)</h1>";
        $error_output .= "You have supplied the wrong
information to " .
        "access a secure area<br>";
        $error_output      .=      "<a
href=\"mailto:sysadmin@localhost.com\">" .
        "Contact</a> the system administrator";
        $error_output .= " if you feel this to be in error";
        break;

    case 403:
        $error_output = "<h1>&quot;Forbidden Access&quot;
Error Page - " .
        "(Error Code 403)</h1>";

```

```

$error_output .= "You are denied access to this
area<br>";
$error_output .= "<a
href='\"mailto:sysadmin@localhost.com\">'>" .
"Contact</a> the system administrator";
$error_output .= " if you feel this to be in error";
break;

```

case 404:

```

$error_output = "<h1>&quot;Page Not Found&quot;
Error Page - " .
"(Error Code 404)</h1>";
$error_output .= "The page you are looking for cannot
" .
"be found<br>";
$error_output .= "<a
href='\"mailto:sysadmin@localhost.com\">'>" .
"Contact</a> the system administrator";
$error_output .= " if you feel this to be in error";
break;

```

case 500:

```

$error_output = "<h1>&quot;Internal Server
Error&quot; " .
"Error Page – (Error Code 500)</h1>";
$error_output .= "The server has encountered an
internal " .
"error<br>";
$error_output .= "<a
href='\"mailto:sysadmin@localhost.com\">'>" .
"Contact</a> the system administrator";

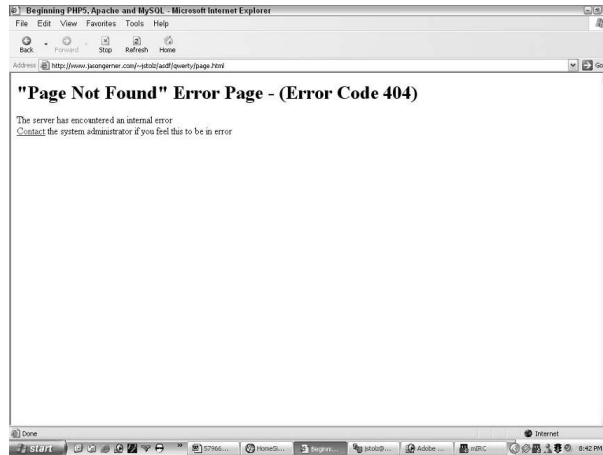
```

```

        $error_output .= " if you feel this to be in error";
        break;
    default:
        $error_output = "<h1>Error Page</h1>";
        $error_output .= "This is the custom error Page<br>";
        $error_output .= "You should be <a
        href=\"index.php\">here</a>";
    }
?>
<html>
<head>
<title>Beginning PHP5, Apache, MySQL Web Development</title>
</head>
<body>
<?php
        echo $error_output;
?>
</body>
</html>

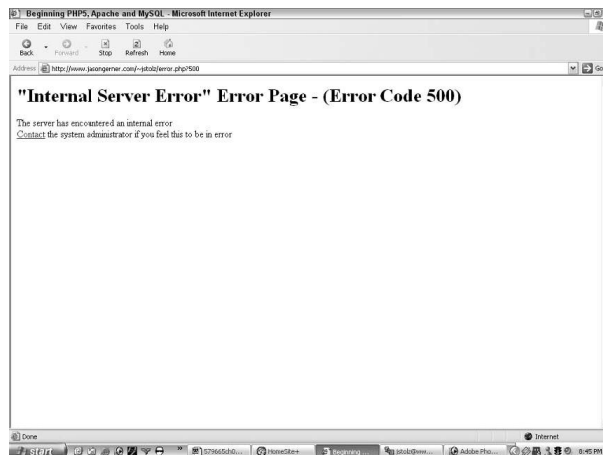
```

3. Mở trình duyệt và vào <http://localhost/asdf/qwerty/page.html> hoặc bất kỳ trang web nào mà bạn biết là không thể vào thăm trang web, bạn sẽ nhìn thấy như **hình 9.1**



Hình 9-1

1. Cách khác để kiểm tra rằng đoạn mã của bạn được ứng dụng vào trong trang web với chuỗi thông tin truy vấn cho người dùng. Ví dụ mở trình duyệt với tên <http://localhost/error.php?500> , trang sẽ sử dụng truy vấn và chạy đoạn mã nếu có lỗi nó sẽ xuất hiện như hình sau (giống hình trước).



Hình 9.2

Nó hoạt động thế nào?

Bạn vừa tạo ra những thông báo lỗi thường gặp, bằng việc sử dụng switch(), bạn có thể hiển thị thích hợp những thông báo lỗi cần thiết cho người dùng.

Lỗi tài liệu của Apache: Trang báo lỗi thông thường

Bạn chỉ cho người dùng nhìn thấy những lỗi thông thường, mặc dù bạn có thể làm nhiều hơn thế, đây là cách tốt để bạn theo dõi những trang mà bạn không thể kiểm tra định kỳ được.

Ví dụ tạo một E_mail lỗi

Trong bài tập này, bạn tạo ra một email phát sinh tự động gửi đến người quản trị thời gian xuất hiện lỗi, vào ngày nào, lỗi gì, trang nào phát sinh lỗi và thông báo lỗi nào được hiển thị cho người dùng định hướng.

1. Mở file error.php, thay đổi như sau:

2. Nhập vào đoạn mã sau:

```
<?php
function email_admin($error_no,
$error_output,
$full_date,
$full_time,
$request_page)
{
    $to = "Administrator <admin@yourdomain.com>";
    $subject = "Apache Error Generation";
    $body = "<html>";
    $body .= "<head>";
    $body .= "<title>Apache Error</title>";
    $body .= "</head>";
    $body .= "<body>";
    $body .= "Error occurred on <b>" . $full_date . "</b> " .
    "at <b>" . $full_time . "</b><br>";
    $body .= "Error received was a <b>" . $error_no . "</b>
error.<br>";
    $body .= "The page that generated the error was: <b>" .
```

```

$request_page . “</b><br>”;
$body .= “The generated error message was:” . $error_output;
$body .= “</body>”;
$body .= “</html>”;
$headers = “MIME-Version: 1.0\r\n”;
$headers .= “Content-type: text/html; charset=iso-8859-1\r\n”;
$headers .= “From: Apache Error <host@yourdomain.com>\r\n”;
$headers .= “Cc: webmaster@yourdomain.com\r\n”;
mail($to, $subject, $body, $headers);
}
$date = getdate();
$full_date = $date[‘weekday’] . “, “ .
$date[‘month’] . “ “ .
$date[‘mday’] . “, “ .
$date[‘year’];
$full_time = $date[‘hours’] . “:” .
$date[‘minutes’] . “:” .
$date[‘seconds’] . “:” .
$date[‘year’];
$error_no = $_SERVER[‘QUERY_STRING’];
$request_page = $_SERVER[‘REQUEST_URI’];
switch ($error_no)
{
case 400:
    $error_output = “<h1>\”Bad Request\” Error Page - “ .
    “(Error Code 400)</h1>”;
    $error_output .= “The browser has made a Bad Request<br>”;
    $error_output .= “<a href=\”mailto:sysadmin@localhost.com\”>” .
    “Contact</a> the system administrator”;

```

```
$error_output .= " if you feel this to be in error";  
email_admin($error_no,  
$error_output,  
$full_date,  
$full_time,  
$request_page);  
break;
```

case 401:

```
$error_output = "<h1>\`Authorization Required\` Error Page - "  
"(Error Code 401)</h1>";  
$error_output .= "You have supplied the wrong information to "  
"access a secure area<br>";  
$error_output .= "<a href=\`mailto:sysadmin@localhost.com\`>" .  
"Contact</a> the system administrator";  
$error_output .= " if you feel this to be in error";  
email_admin($error_no,  
$error_output,  
$full_date,  
$full_time,  
$request_page);  
break;
```

case 403:

```
$error_output = "<h1>\`Forbidden Access\` Error Page - "  
"(Error Code 403)</h1>";  
$error_output .= "You are denied access to this area<br>";  
$error_output .= "<a href=\`mailto:sysadmin@localhost.com\`>" .  
"Contact</a> the system administrator";  
$error_output .= " if you feel this to be in error";  
email_admin($error_no,
```



```
$error_output,  
$full_date,  
$full_time,  
$request_page);  
break;
```

case 404:

```
$error_output = "<h1>\\"Page Not Found\\" Error Page - "  
"(Error Code 404)</h1>";  
$error_output .= "The page you are looking for "  
"cannot be found<br>";  
$error_output .= "<a href=\\\"mailto:sysadmin@localhost.com\\\">".  
"Contact</a> the system administrator";  
$error_output .= " if you feel this to be in error";  
email_admin($error_no,  
$error_output,  
$full_date,  
$full_time,  
$request_page);  
break;
```

case 500:

```
$error_output = "<h1>\\"Internal Server Error\\" Error Page - "  
"(Error Code 500)</h1>";  
$error_output .= "The server has encountered "  
"an internal error<br>";  
$error_output .= "<a href=\\\"mailto:sysadmin@localhost.com\\\">".  
"Contact</a> the system administrator";  
$error_output .= " if you feel this to be in error";  
email_admin($error_no,  
$error_output,
```

```

        $full_date,
        $full_time,
        $request_page);
    break;
default:
    $error_output = "<h1>Error Page</h1>";
    $error_output .= "This is the custom error Page<br>";
    $error_output .= "You should be <a href=\"index.php\">here</a>";
}
?>
<html>
<head>
<title>Beginning PHP5, Apache, MySQL Web Development</title>
</head>
<body>
<?php
echo $error_output;
?>
</body>
</html>

```

Nó hoạt động thế nào?

Hàm mail() được sử dụng để gửi email cho người quản trị, getdate() ghi nhớ thời gian và ngày xuất hiện lỗi. Hàm mail() cho phép bạn gửi mail đến bất cứ ai khi xuất hiện lỗi.

Chương 9 : ĐIỀU KHIỂN VÀ TRÁNH NHỮNG LỖI

Sau khi viết mã lệnh, người mới lập trình sẽ gặp không ít khó khăn trong việc tìm ra lỗi và chỉnh sửa lỗi. PHP hỗ trợ đầy đủ tính năng lập trình giao diện API (Applications Programming Interface) có thể tìm ra lỗi và khắc phục những lỗi ấy. Ngoài PHP thì Apache Web Server cũng hỗ trợ người lập trình tìm ra lỗi và khắc phục lỗi.

Tiếp xúc với lỗi Apache Web Server như thế nào?

Apache có một thư mục Lỗi tài liệu, bạn có thể định cấu hình trong file httpd.conf để tạo ra những trang thông báo lỗi với PHP, vì thế người thăm trang web không thấy buồn chán. PHP tạo ra trang web thông báo lỗi thân thiện cho người dùng.

Không giống những trang thông báo lỗi như PHP, trang của Apache được sử dụng nhiều cho những trang đã mất.

Thi hành lỗi và tạo ra việc sửa trang lỗi với PHP

Phụ thuộc vào kiểu lỗi xuất hiện mà đưa ra những thông báo lỗi thích hợp, phải đưa ra những lỗi để người nhận thông báo lỗi cảm thấy dễ chịu, không cảm thấy xấu hổ và có hướng giải quyết lỗi tốt.

Các loại lỗi trong PHP

Có 12 kiểu lỗi trong PHP, được liệt kê dưới đây

Lỗi	Giá trị nguyên	Tên hằng
E_ERROR	1	Lỗi nặng
E_WARNING	2	Lỗi nhẹ(cảnh báo)
E_PARSE	4	Thời gian biên dịch lỗi

E_NOTICE	8	Chú ý
E_CORE_ERROR	16	Lỗi nặng xuất hiện lúc khởi động
E_CORE_WARNINGS	32	Lỗi không nặng ngay lúc khởi động
E_COMPILE_WARNING	128	Lỗi thời gian biên dịch không nặng lắm(cảnh báo)
E_USER_ERROR	256	Lỗi phát sinh người dùng bởi hàm PHP trigger_error()
E_USER_WARNING	512	Cảnh báo người dùng khi dùng hàm trigger_error()
E_USER_NOTICE	1024	Chú ý khi dùng hàm trigger_error()
E_ALL	2047	Tất cả những lỗi và cảnh báo
E_STRICT	2048	Chú ý để thời gian thực hiện cần có thay đổi tương thích

Bạn không phải lo lắng về lỗi đánh máy, hãy quan tâm đến những cảnh báo, những lỗi mà bạn gặp phải, để có sự giải quyết kịp thời, để sử dụng hơn cho người dùng.

Ở đây có 3 loại lỗi chính

Fatal error: Lỗi nặng khi thực hiện, thông báo lỗi không thể sửa đổi, thực hiện theo nguyên bản

Warning: Cảnh báo, không được dùng tập lệnh vừa rồi.

Notices: Chú ý, khi chạy vẫn đúng nhưng có một lỗi nhỏ.

Việc phát sinh lỗi PHP

Chúng ta tạo vài lỗi để có thể kiểm tra cần giải quyết thế nào? Như ví dụ sau:

```
<?php
//set string with "Wrox" spelled wrong
$string_variable = "Worx books are great!";
//try to use str_replace to replace Worx with Wrox
//this will generate an E_WARNING
//because of wrong parameter count
str_replace("Worx", "Wrox");
?>
```

Nếu bạn chạy đoạn mã này, bạn sẽ thấy lỗi như sau:

```
Warning: Wrong parameter count for str_replace() in
C:\FoxServ\www\errorhandling\error1.php on line 8
```

Lỗi này xuất hiện vì hàm `str_replace` yêu cầu tham số thứ 3 cho hàm. Tham số này là biến, `$string_variable` hoặc chuỗi văn bản mà bạn muốn nghiên cứu cho tham số thứ nhất, "Worx" và thay thế nó với "Wrox". Bởi điều này không tạo ra lỗi lớn mà không thi hành tập lệnh, bạn có thể vẫn chạy đoạn code sau khi lỗi xuất hiện, nếu bạn làm như sau:

```
<?php
//set string with "Wrox" spelled wrong
$string_variable = "Worx books are great!";
//try to use str_replace to replace Worx with Wrox
//this will generate an E_WARNING
//because of wrong parameter count
str_replace("Worx", "Wrox");
```

```
//this is a non-fatal error, so the original
//variable should still show up after the warning
echo $string_variable;
?>
```

Chuỗi vẫn tiếp tục được thi hành sau lỗi, nó sẽ sinh ra như sau:

```
Warning: Wrong parameter count for str_replace() in
c:\FoxServ\www\errorhandling\error1.php on line 8
Worx books are great!
```

Kế đến, chúng ta sẽ đưa ra những lỗi nặng để cho bạn thấy nó sinh ra những kết quả khác nhau khi lỗi xuất hiện như thế nào.

```
<?php
//beginning of page
echo "Beginning";
//we are going to make a call to
//a function that doesn't exist
//this will generate an E_ERROR
//and will halt script execution
//after the call of the function
fatalerror();
//end of page
echo "End";
//won't be output due to the fatal error
?>
```

Sẽ xuất hiện như sau:

```
Beginning
Fatal error: Call to undefined function: fatalerror() in
c:\FoxServ\www\errorhandling\error2.php on line 10.
```

Chú ý là “Beginning” được đưa ra vì hàm trước nó được gọi, nhưng “End” thì không đưa ra vì lỗi thực hiện quá lớn. Bạn có thể ngăn lỗi lại bằng việc dùng một ký hiệu như sau: @fatalerror (ngăn chặn lỗi nhưng vẫn gặp lỗi khi thi hành)

Khi PHP mặc định lỗi không thấy E_NOTICE xuất hiện, tuy nhiên bạn có thể cho phép người dùng gỡ những lỗi như khi dùng \$row[variable] vì PHP coi [variable] như một hằng, nếu không là hằng nó giả thiết là chuỗi trong mảng. Có thể đưa ra thông báo lỗi khi dùng error_reporting(*number*) mà số không thay đổi.

Nếu bạn không biết cấp độ lỗi được thiết lập, bạn có thể chạy hàm error_reporting() đơn giản với bất kỳ chủ đề nào:

```
<?php
echo error_reporting();
?>
```

Mặc định, tất cả những việc thi hành lỗi được điều khiển bởi người xây dựng nên lỗi, cho bạn những thông báo về liệu lỗi, tên file, số dòng lỗi xuất hiện

Bạn chú ý một lỗi như sau:

```
Warning: Wrong parameter count for str_replace() in
c:\FoxServ\www\errorhandling\error1.php on line 8
```

Thông thường việc phát sinh lỗi thì tốt vì có thể thông báo cho người quản trị biết là anh ta có thể biết nhiều hơn từ những lỗi ấy.

Ví dụ về việc một người tạo lỗi thông thường

Bạn sẽ tạo một lỗi thông thường nhưng thân thiện với người dùng và hiển thị như sau:

1. Trình soạn thảo trong ví dụ trước như sau:

```
<?php
//create your error handler function
function handler($error_type,
$error_message,
$error_file,
$error_line)
```

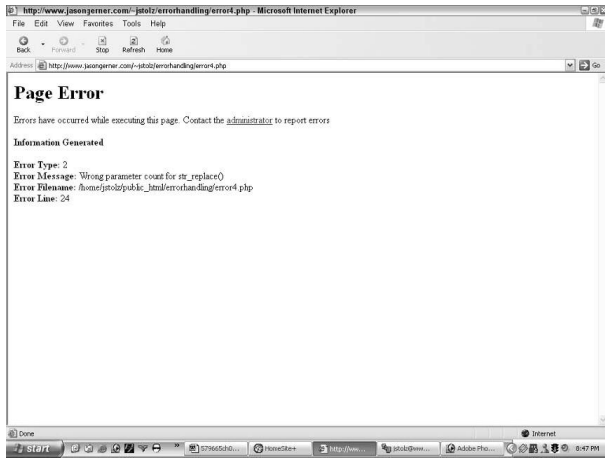
```

{
    echo "<h1>Page Error</h1>";
    echo "Errors have occurred while executing this page.
    Contact the ";
    echo
    href="mailto:admin@yourdomain.com">administrator</a>
    ".
    "to report errors<br><br>";
    echo "<b>Information Generated</b><br><br>";
    echo "<b>Error Type:</b> $error_type<br>";
    echo "<b>Error Message:</b> $error_message<br>";
    echo "<b>Error Filename:</b> $error_file<br>";
    echo "<b>Error Line:</b> $error_line";
}

//set the error handler to be used
set_error_handler("handler");
//set string with "Wrox" spelled wrong
$string_variable = "Worx books are great!";
//try to use str_replace to replace Worx with Wrox
//this will generate an E_WARNING
//because of wrong parameter count
str_replace("Worx", "Wrox");
?>

```

2. Lưu file với tên custom_error.php và mở trình duyệt, bạn sẽ thấy như



Chương 9 : ĐIỀU KHIỂN VÀ TRÁNH NHỮNG LỖI

Sau khi viết mã lệnh, người mới lập trình sẽ gặp không ít khó khăn trong việc tìm ra lỗi và chỉnh sửa lỗi. PHP hỗ trợ đầy đủ tính năng lập trình giao diện API (Applications Programming Interface) có thể tìm ra lỗi và khắc phục những lỗi ấy. Ngoài PHP thì Apache Web Server cũng hỗ trợ người lập trình tìm ra lỗi và khắc phục lỗi.

Tiếp xúc với lỗi Apache Web Server như thế nào?

Apache có một thư mục Lỗi tài liệu, bạn có thể định cấu hình trong file httpd.conf để tạo ra những trang thông báo lỗi với PHP, vì thế người thăm trang web không thấy buồn chán. PHP tạo ra trang web thông báo lỗi thân thiện cho người dùng.

Không giống những trang thông báo lỗi như PHP, trang của Apache được sử dụng nhiều cho những trang đã mất.

Việc phát sinh lỗi PHP (tiếp theo)

3. Vì lỗi của bạn là do người dùng định nghĩa, bạn có thể bắt lỗi và có thể tạo lại thông báo lỗi dựa vào kiểu lỗi. Tạo file custom_error.php để sắp xếp những lỗi như sau:

```
<?php
//create your error handler function
function handler($error_type,
$error_message,
$error_file,
$error_line)
{
    switch ($error_type)
    {
        //fatal error
        case E_ERROR:
```

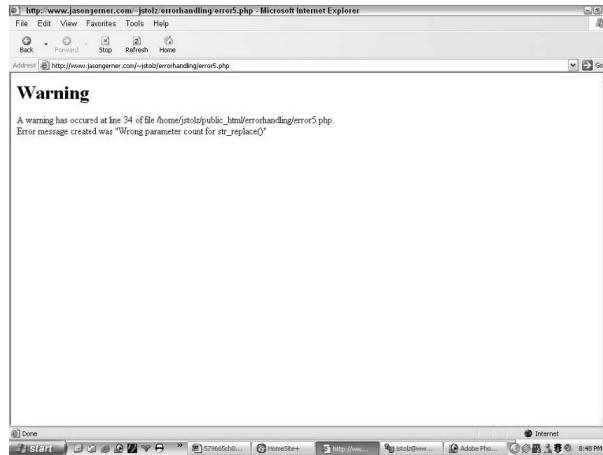
```

        echo "<h1>Fatal Error</h1>";
        die("A fatal error has occurred at line $error_line of file
        "
        "$error_file.<br>"
        "Error message created was
        &quot;$error_message&quot;");
        break;
//warnings
case E_WARNING:
    echo "<h1>Warning</h1>";
    echo "A warning has occurred at line $error_line of file
    "
    "$error_file.<br>";
    echo "Error message created was
    &quot;$error_message&quot;";
//notices
case E_NOTICE:
//don't show notice errors
    break;
    }
}
//set the error handler to be used
set_error_handler("handler");
//set string with "Wrox" spelled wrong
$string_variable = "Worx books are great!";
//try to use str_replace to replace Worx with Wrox
//this will generate an E_WARNING
//because of wrong parameter count
str_replace("Worx", "Wrox");

```

?>

4. Lưu file và đưa lên trình duyệt. Kết quả như Hình 9.4. E_ERROR được gọi trong mệnh đề switch có thể sử dụng bất kỳ loại lỗi nào và thực hiện những hoạt động khác nhau tùy theo lỗi.



Hình 9.4.

Nó hoạt động thế nào?

Tạo ra thông báo lỗi cho bạn điều khiển đầy đủ những trang web của bạn, bất chấp thành công hay thất bại khi thi hành. Bạn sẽ bắt lỗi, đưa ra thông báo lỗi, hồ sơ thông báo lỗi và lỗi nào xuất hiện. Trang web mà bạn đưa ra thông báo lỗi, trình bày hoặc gửi thông báo nhất định đến người dùng phụ thuộc vào lỗi mà người dùng mắc phải.

Bạn đã tạo ra thông báo lỗi nhưng không muốn người dùng nhìn thấy thông báo ấy, bạn có thể lịch sự xin lỗi người dùng hãy qua một trang khác vì trang này đang bị lỗi như thế người dùng không thấy lỗi của mình. Dùng phương pháp email có thể cho người dùng không trực tiếp xem xét cơ sở dữ liệu mà là có bao nhiêu yêu cầu tới trang lỗi xuất hiện.

Ví dụ về việc tạo một trang lỗi rõ ràng

Trong bài này, bạn sẽ thiết lập việc thi hành một lỗi rõ ràng như bạn muốn. Bạn có thể tìm tất cả những lỗi mà không cần sử dụng những hướng dẫn có sẵn trong PHP

1. Trình soạn thảo feature_error.php như sau:

```
<?php
//create your error handler function
```

```

function handler($error_type,
$error_message,
$error_file,
$error_line) {
switch($error_type)
{
//fatal error
case E_ERROR:
$to = "Administrator <admin@yourdomain.com>";
$subject = "Custom Error Handling";
$body = "<html>";
$body .= "<head>";
$body .= "<title>Website error</title>";
$body .= "</head>";
$body .= "<body>";
$body .= "<h1>Fatal Error</h1>";
$body .= "Error received was a <b>" . $error_type .
"</b> error.<br>";
$body .= "The page that generated the error was: <b>" .
$error_file . "</b>";
$body .= " and was generated on line: <b>" . $error_line .
"</b><br>";
$body .= "The generated error message was:" .
$error_message;
$body .= "</body>";
$body .= "</html>";
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-
1\r\n";

```

```

$headers .= "From: Apache Error
<host@yourdomain.com>\r\n";
$headers .= "Cc: webmaster@yourdomain.com\r\n";
mail($to, $subject, $body, $headers);
die(); //kill the script
break;
//warnings
case E_WARNING:
$to = "Administrator <admin@yourdomain.com>";
$subject = "Custom Error Handling";
$body = "<html>";
$body .= "<head>";
$body .= "<title></title>";
$body .= "</head>";
$body .= "<body>";
$body .= "<h1>Warning</h1>";
$body .= "Error received was a <b>" . $error_type .
"</b> error.<br>";
$body .= "The page that generated the error was: <b>" .
$error_file . "</b>";
$body .= " and was generated on line: <b>" . $error_line .
"</b><br>";
$body .= "The generated error message was:" .
$error_message;
$body .= "</body>";
$body .= "</html>";
$headers = "MIME-Version: 1.0\r\n";
$headers .= "Content-type: text/html; charset=iso-8859-
1\r\n";

```

```

$headers .= "From: Apache Error
<host@yourdomain.com>\r\n";
$headers .= "Cc: webmaster@yourdomain.com\r\n";
mail($to, $subject, $body, $headers);
break;
//script will continue
//notices
case E_NOTICE:
//don't show notice errors
break;
}
}
/*
set error handling to 0
we will handle all error reporting
only notifying admin on warnings and fatal errors
don't bother with notices as they are trivial errors
really only meant for debugging
*/
error_reporting(0);
//set the error handler to be used
set_error_handler("handler");
/*
Create the rest of your page here.
We will not be displaying any errors
We will be e-mailing the admin an error message
Keep in mind that fatal errors will still halt the
execution, but they will still notify the admin
*/

```

?>

Nó hoạt động thế nào?

Với cách này đưa ra thông báo đến người dùng những thông báo lỗi thông thường nhưng thân thiện đối với người dùng bằng cách dùng hàm email .

Vì vậy, với cách đơn giản bạn chạy trang này và bạn nhận một lỗi, đôi khi đưa ra thông tin có ích cho người dùng.

Chương 9 : ĐIỀU KHIỂN VÀ TRÁNH NHỮNG LỖI

Sau khi viết mã lệnh, người mới lập trình sẽ gặp không ít khó khăn trong việc tìm ra lỗi và chỉnh sửa lỗi. PHP hỗ trợ đầy đủ tính năng lập trình giao diện API (Applications Programming Interface) có thể tìm ra lỗi và khắc phục những lỗi ấy. Ngoài PHP thì Apache Web Server cũng hỗ trợ người lập trình tìm ra lỗi và khắc phục lỗi.

Tiếp xúc với lỗi Apache Web Server như thế nào?

Apache có một thư mục Lỗi tài liệu, bạn có thể định cấu hình trong file httpd.conf để tạo ra những trang thông báo lỗi với PHP, vì thế người thăm trang web không thấy buồn chán. PHP tạo ra trang web thông báo lỗi thân thiện cho người dùng.

Không giống những trang thông báo lỗi như PHP, trang của Apache được sử dụng nhiều cho những trang đã mất.

Một phương pháp khác của việc thi hành lỗi

Bạn vừa đưa ra những thông báo lỗi thông thường, có một phương pháp khác tự nhiên hơn chèn mệnh đề echo để kiểm tra giá trị của biến và xem mệnh đề điều kiện đúng hay chưa. PHP cũng cung cấp vài thông báo lỗi cho những lỗi đơn giản.

Những ngoại lệ

PHP5 giới thiệu đặc tính mới gọi là những ngoại lệ. Những điều đó tương tự như ngôn ngữ Java. Những điều kiện điều khiển ngoại lệ trong ứng dụng Web và cho bạn một cách hiệu quả để điều khiển những lỗi gặp phải. PHP5 sử dụng phương pháp tìm kiếm, nắm bắt để điều khiển những ngoại lệ.

Ví dụ về việc thử nghiệm với những ngoại lệ

Trong bài này, bạn sẽ tạo một phiên bản đưa ra vài ngoài lệ để thấy chúng làm việc thế nào

1. Tạo một trang PHP với đoạn code sau:

```
<?php
    //$x = ""; //Throws null Exception
    //$x = "500"; //Throws less than Exception
    $x = "1000"; //Throws NO Exception
    try
    {
        if ($x == "")
        {
            throw new Exception("Variable cannot be null");
        }
        if ($x < 1000)
        {
            throw new Exception("Variable cannot be less
than 1000");
        }
        echo "Validation Accepted!";
    }
    catch (Exception $exception)
    {
        echo $exception->getMessage();
    }
}
```

```
echo " - Validation Denied!";
```

```
}
```

?>

2. Lưu đoạn code với tên exceptions.php và chạy nó trong trình duyệt. Bạn không thấy bất cứ lỗi nào

3. Thay $x= "1000"$ bằng $x= ""$

4. Lưu file và chạy một lần nữa, bạn sẽ thấy một thông báo rỗng

5. Thay $x= ""$ bằng $x= "500"$

6. Lưu file lại và bạn sẽ thấy thông báo "less than 1000"

Nó hoạt động thế nào?

Sự hữu ích của khối này là tất cả những điều kiện phải gặp hoặc thấy. Đây là hữu ích khi bạn cần kiểm tra nhiều biến hay không muốn thông qua nhiều mệnh đề if/else cho việc tìm ra kết quả. Tất cả yêu cầu trong mệnh đề if mà đúng thì sẽ thực hiện, sẽ đưa ra thông báo lỗi thích hợp phụ thuộc vào ngoại lệ gặp phải.

Trong vùng thấy lỗi, bạn có thể thi hành lỗi trong bất cứ phương pháp nào bạn thích. Bạn muốn gọi đến người dùng là bạn muốn thiết lập vài biến mặc định, kết nối cả hai hoặc bất cứ cái gì mà bạn cần.

Hữu ích của việc sử dụng những ngoại lệ là cách chúng đưa ra những hàm và code. Ví dụ, nếu bạn có một hàm gọi là hàm B mà bên trong thì gọi hàm C và một ngoại lệ trong hàm C thông qua việc sử dụng `try{} ,` ngoại lệ sẽ dừng việc xử lý phiên bản nếu khối được tìm thấy

Nếu không tìm thấy try{}...catch{} khi thi hành code, lỗi sẽ xuất hiện trên màn hình để thông báo ngoại lệ đã xuất hiện.

Ngoại lệ cũng có thể thông qua như sau:

```
<?php
    try
    {
        throw new Exception("This will be rethrown");
    }
    catch (Exception $e)
    {
        throw $e;
    }
?>
```

Không gặp những điều kiện

Không phải trong bất cứ đoạn mã nào cũng gặp lỗi, nó chỉ xuất hiện khi liên quan đến PHP, bằng việc sử dụng vài phương pháp khác nhau trong đoạn mã để kiểm tra lỗi xuất hiện.

```
<input type="submit" name="submit" value="Submit">
```

Bạn đang kiểm tra để thấy nút nhấn khi nào hoặc sẽ không thi hành thông tin khi nào. Bạn có thể làm điều này như sau:

```
if ($_POST['submit'] == "submit")
{
```

```
        //form has been submitted
    } else
    {
        //form has not been submitted
    }
```

Phân tích lỗi

Phân tích lỗi là kiểu lỗi chính, nó xuất hiện khi bạn quên một dấu chấm phẩy,...Những lỗi này không làm với mệnh đề điều kiện, chúng là cú pháp chính là nguyên nhân dừng việc thực thi. Phân tích lỗi xấu hơn những lỗi nặng vì chúng không thể chạy tất cả những phiên bản, chúng cho bạn thông tin về lỗi.

Tóm lại.

Bạn đọc được nhiều thứ trong chương này, từ việc biết được những lỗi cho đến việc phòng ngừa những lỗi, giúp người lập trình làm thế nào để phòng ngừa lỗi, tìm ra lỗi.