

Bài 1: Giới thiệu, Cấu hình và cài đặt PHP



PHP (Hypertext Preprocessor) là ngôn ngữ script trên server được thiết kế để dễ dàng xây dựng các trang Web động. Mã PHP có thể thực thi trên Webserver để tạo ra mã HTML và xuất ra trình duyệt web theo yêu cầu của người sử dụng.

Ngôn ngữ PHP ra đời năm 1994 Rasmus Lerdorf sau đó được phát triển bởi nhiều người trải qua nhiều phiên bản. Phiên bản hiện tại là PHP 5 đã được công bố 7/2004.

Có nhiều lý do khiến cho việc sử dụng ngôn ngữ này chiếm ưu thế xin nêu ra đây một số lý do cơ bản :

- Mã nguồn mở (open source code)
- Miễn phí, download dễ dàng từ Internet.
- Ngôn ngữ rất dễ học, dễ viết.
- Mã nguồn không phải sửa lại nhiều khi viết chạy cho các hệ điều hành từ Windows, Linux, Unix
- Rất đơn giản trong việc kết nối với nhiều nguồn DBMS, ví dụ như : MySQL, Microsoft SQL Server 2000, Oracle, PostgreSQL, Adabas, dBase, Empress, FilePro, Informix, InterBase, mSQL, Solid, Sybase, Velocis và nhiều hệ thống CSDL thuộc Hệ Điều Hành Unix (Unix dbm) cùng bất cứ DBMS nào có sự hỗ trợ cơ chế ODBC (Open Database Connectivity) ví dụ như DB2 của IBM.

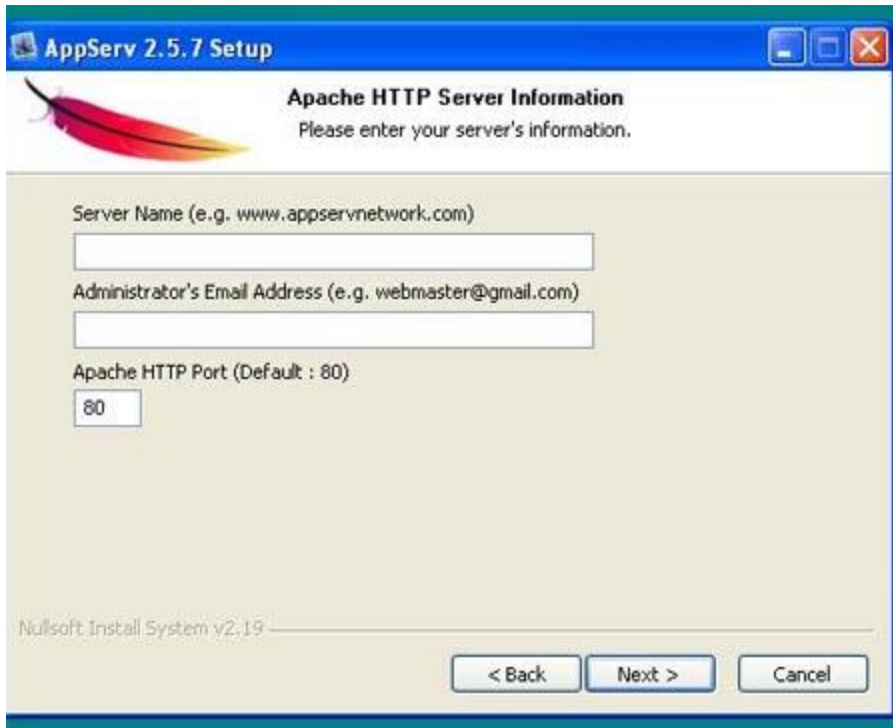
Cài đặt PHP, Apache, MYSQL

Để cài đặt PHP, các bạn có thể cài đặt từng gói riêng lẻ (PHP, Apache, Mysql). Tuy nhiên, tôi vẫn khuyến khích các bạn cài đặt dạng gói tích hợp. Sẽ tiện lợi cho việc sau này hơn rất nhiều.

Gói phần mềm mà tôi chọn là : Appserv - Win - 2.5.8 (Tên phần mềm là Appserv, dành cho phiên bản window, phiên bản 2.5.8).

Bước 1: Bạn cài đặt bình thường bằng cách nhấp vào file exe.

Bước 2: Phần mềm sẽ cho bạn chọn cần cài những module nào. Hãy giữ nguyên tất cả như hình dưới và nhấn next.



AppServ 2.5.7 Setup

Apache HTTP Server Information
Please enter your server's information.

Server Name (e.g. www.appservnetwork.com)

Administrator's Email Address (e.g. webmaster@gmail.com)

Apache HTTP Port (Default : 80)

Nullsoft Install System v2.19

< Back Next > Cancel

Bước 3: Trong giao diện dưới:



AppServ 2.5.7 Setup

MySQL Server Configuration
Configure the MySQL Server instance.

Please enter Root password for MySQL Server.

Enter root password

Re-enter root password

MySQL Server Setting

Character Sets and Collations

☒ Old Password Support (PHP MySQL API function.)

☐ Enable InnoDB

Nullsoft Install System v2.19

< Back Install Cancel

Server Name: bạn nhập vào: localhost

Email: Bạn nhập vào email của bạn:

Port: Mặc định là 80, tuy nhiên nếu máy bạn đã cài IIS thì có thể chỉnh thành 81 để chạy dịch vụ bên appserv.

Nhấn next để qua trang kế tiếp.

Bước 4: Trong giao diện bên dưới ta điền thông tin như sau:



Enter root password: Bạn gõ vào root

Re-enter root password: nhập lại 1 lần nữa root

Các cấu hình bên dưới giữ nguyên.

Ở phần: Enable InnoDB bạn đánh dấu vào. Để MYSQL sử lý được các ứng dụng có bật chế độ InnoDB.

Nhấn next để hoàn tất việc cài đặt.

Sau khi cài đặt xong bạn gõ vào trình duyệt : <http://localhost>

mà ra giao diện bên dưới, tức là bạn đã cài đặt thành công appserv.



The AppServ Open Project - 2.5.7 for Windows


 **phpMyAdmin Database Manager Version 2.9.0.2**
 **PHP Information Version 5.1.6**

About AppServ Version 2.5.7 for Windows

AppServ is a merging open source software installer package for Windows includes :

- **Apache Web Server** Version 2.2.3
- **PHP Script Language** Version 5.1.6
- **MySQL Database** Version 5.0.24a
- **phpMyAdmin Database Manager** Version 2.9.0.2
- ChangeLog
- README
- AUTHORS
- COPYING
- <http://www.AppServNetwork.com>

Change Language :  

 **Easy way to build Webserver, Database Server with AppServ :-)**

Như vậy là bạn đã cài đặt PHP thành công.

Bài 2: Kiến thức tổng quan về lập trình PHP



Ở bài trước chúng ta đã tìm hiểu về cách thức cấu hình và cài đặt PHP. Tiếp theo chúng ta sẽ nghiên cứu về cấu trúc cơ bản trong PHP. Về tổng quan PHP có cú pháp khá tương đồng với 1 số ngôn ngữ như C, java. Tuy nhiên, tự bản thân chúng cũng có những điểm rất riêng biệt.



1- Cấu trúc cơ bản:

PHP cũng có thể bắt đầu và kết thúc giống với ngôn ngữ HTML. Chỉ khác, đối với PHP chúng ta có nhiều cách để thể hiện.

Cách 1 : Cú pháp chính:

<?php Mã lệnh PHP ?>

Cách 2: Cú pháp ngắn gọn

<? Mã lệnh PHP ?>

Cách 3: Cú pháp giống với ASP.

<% Mã lệnh PHP %>

Cách 4: Cú pháp bắt đầu bằng script

<script language=php>

.....

</script>

Mặc dù có 4 cách thể hiện. Nhưng đối với 1 lập trình viên có kinh nghiệm thì việc sử dụng cách 1 vẫn là lựa chọn tối ưu.

Trong PHP để kết thúc 1 dòng lệnh chúng ta sử dụng dấu ";"

Để chú thích 1 đoạn dữ liệu nào đó trong PHP ta sử dụng dấu "//" cho từng dòng. Hoặc dùng cặp thẻ "/*.....*/" cho từng cụm mã lệnh.

```
1
2 <?
3 Echo "PHP is simple"; //đây là ví dụ về code PHP
4 /* Voi cu phap nay chung ta
5 Co the chu thích 1 cum ma lenh */
6 ?>
7
```

2- Xuất giá trị ra trình duyệt:

-

Để xuất dữ liệu ra trình duyệt chúng ta có những dòng cú pháp sau :

+ Echo "Thông tin";

+ Printf "Thông tin";

Thông tin bao gồm : biến, chuỗi, hoặc lệnh HTML

```
1 <?php
2 Echo "Hello word";
3 Printf"<br><font color=red>Who Are You ?</font>";
4 ?>
5
```

Nếu giữa hai chuỗi muốn liên kết với nhau ta sử dụng dấu "."

```
1 <?php
2 Echo "Hello"."who are you ?";
3 ?>
4
```

3- Khái niệm biến, hằng, chuỗi và các kiểu dữ liệu.

a) Biến trong PHP.

Biến được xem là vùng nhớ dữ liệu tạm thời. Và giá trị có thể thay đổi được. Biến được bắt đầu bằng ký hiệu "\$". Và theo sau chúng là 1 từ, 1 cụm từ nhưng phải viết liền hoặc có gạch dưới.

1 biến được xem là hợp lệ khi nó thỏa các yếu tố :

+ Tên của biến phải bắt đầu bằng dấu gạch dưới và theo sau là các ký tự, số hay dấu gạch dưới.

+ Tên của biến không được phép trùng với các từ khóa của PHP.

Trong PHP để sử dụng 1 biến chúng ta thường phải khai báo trước, tuy nhiên đối với các lập trình viên khi sử dụng họ thường xử lý cùng một lúc các công việc, nghĩa là vừa khai báo vừa gán dữ liệu cho biến.

Bản thân biến cũng có thể gán cho các kiểu dữ liệu khác. Và tùy theo ý định của người lập trình mong muốn trên chúng.

Một số ví dụ về biến :

```
1 <?
2 $a= 100 // biến a ở đây có giá trị là 100.
3 $a= "PHP is easy" // Biến a ở đây có giá trị "PHP Is easy".
4 Biena=123 //Có lỗi vì bắt đầu 1 biến phải có dấu "$"
5 $123a="PHE" //Có lỗi vì phần tên bắt đầu của biến là dạng số.
6 ?>
```

b) Khái niệm về hằng trong PHP.

Nếu biến là cái có thể thay đổi được thì ngược lại hằng là cái chúng ta không thể thay đổi được. Hằng trong PHP được định nghĩa bởi hàm define theo cú pháp: define (string tên_hằng, giá_trị_hằng).

Cũng giống với biến hằng được xem là hợp lệ thì chúng phải đáp ứng 1 số yếu tố :

- + Hằng không có dấu "\$" ở trước tên.
- + Hằng có thể truy cập bất cứ vị trí nào trong mã lệnh
- + Hằng chỉ được phép gán giá trị duy nhất 1 lần.
- + Hằng thường viết bằng chữ in để phân biệt với biến

Ví dụ :

```
1 <?
2 define ("C", "COMPANY");
3 define ("YELLOW", "#ffff00");
4 echo "Gia tri cua C la". C;
5 ?>
```

c) *Khái niệm về chuỗi:*

Chuỗi là một nhóm các kỹ tự, số, khoảng trắng, dấu ngắt được đặt trong các dấu nháy.

Ví dụ:

‘Huy’

"welcome to VietNam"

Để tạo 1 biến chuỗi, chúng ta phải gán giá trị chuỗi cho 1 biến hợp lệ.

Ví dụ:

\$fisrt_name= "Nguyen";

\$last_name= ‘Van A’;

Để liên kết 1 chuỗi và 1 biến chúng ta thường sử dụng dấu "."

Ví dụ:

```
1 <?php
2 $test="QHOnline.Info";
3 echo "welcome to".$test;
4 echo "<br><font color=red>welcome to".$test."</font><br>";
5 ?>
6
```

d) Kiểu dữ liệu trong PHP

Các kiểu dữ liệu khác nhau chiếm các lượng bộ nhớ khác nhau và có thể được xử lý theo cách khác nhau khi chúng được theo tác trong 1 script.

Trong PHP chúng ta có 6 kiểu dữ liệu chính như sau :

Kiểu Dữ Liệu	Ví dụ	Mô Tả
Integer	10	Một số nguyên
Double	5.208	Kiểu số thực
String	"How are you ?"	Một tập hợp các ký tự
Boolean	True or False	Giá trị true hoặc false
Object	Hướng đối tượng trong PHP	
Array	Mảng trong PHP, chứa các phần tử.	

Chúng ta có thể sử dụng hàm dựng sẵn `gettype()` của PHP4 để kiểm tra kiểu của bất kỳ biến.

Ví dụ:

```
1 <?php
2 $a= 5;
3 echo gettype($a); // Integer.
4 $a="qhonline.info";
5 echo gettype($a); //String
6 ?>
7 |
```

Sau bài này các bạn đã có những khái niệm đầu tiên về PHP, các cú pháp, các kiểu dữ liệu, và cách làm việc với môi trường PHP như thế nào. Ở bài sau, chúng ta sẽ tiếp tục tiếp cận với các thuật toán và cú pháp PHP một cách rõ ràng và quen thuộc trong các ngôn ngữ lập trình.

Bài 3: Toán tử và biểu thức trong PHP



Đối với mọi ngôn ngữ lập trình, toán tử và biểu thức luôn là những kiến thức cơ bản được sử dụng để xử lý các thao tác trong giai đoạn lập trình. Và PHP cũng không ngoại lệ, chúng vẫn có những kiến thức cơ bản như một ngôn ngữ lập trình thông thường.

A- Toán tử trong PHP:

1- Toán tử gán:

Chúng ta đã từng tiếp xúc với toán tử này bởi việc khởi tạo 1 biến. Nó gồm ký tự đơn `=`. Toán tử gán lấy giá trị của toán hạng bên phải gán nó vào toán hạng bên trái.

Ví dụ:

-

\$name = "Johny Nguyen";

2- Toán tử số học:

Là dạng phép tính giản đơn cộng, trừ, nhân, chia trong số học. Ngoài ra còn có phép chia lấy dư (%). Được sử dụng để lấy ra đơn vị dư của 1 phép toán.

Toán Tử	Lý giải	Ví dụ	Kết quả
+	Cộng hai số hạng	10+8	18
-	Trừ hai số hạng	10-8	2
*	Nhân hai số hạng	10*8	80
/	Chia hai số hạng	10/3	3.33333333
%	Trả về số dư	10%3	1

3- Toán tử so sánh:

Là toán tử được sử dụng để thực hiện các phép toán so sánh giữa hai số hạng. Chi tiết, xem bảng bên dưới.

Phép Toán	Tên	Giải Thích	Ví Dụ
==	Bằng	Hai số hạng bằng nhau	\$a==5\$
!=	Không bằng	Hai số hạng không bằng nhau	\$a!=5\$
===	Đồng nhất	Hai số bằng nhau và cùng kiểu	\$a===5\$
>	Lớn hơn	Vế trái lớn hơn vế phải	\$a>5\$
>=	Lớn hơn hoặc bằng	Vế trái lớn hơn hoặc bằng vế phải	\$a>=5\$
<	Nhỏ hơn	Vế trái nhỏ hơn vế phải	\$a<5\$
<=	Nhỏ hơn hoặc bằng	Vế trái nhỏ hơn hoặc bằng vế phải	\$a<=5\$

4- Toán tử logic:

Toán tử logic là các tổ hợp các giá trị boolean.

Ví dụ: toán tử or trở về true nếu toán tử trái hoặc toán tử phải là true.

True || false à true.

Ta có bảng các toán tử như sau:

Toán Tử	Tên	Trả về True Nếu	Ví dụ	Kết quả
	Or	Về trái hoặc về phải là True	True false	True
Or	Or	Về trái hoặc về phải là True	True false	True
Xor	Xor	Về trái hoặc về phải là True nhưng không phải cả hai	True True	False
&&	And	Về trái và về phải là true	True && false	False
And	And	Về trái và về phải là true	True && false	False
!	Not	không phải là true	!true	False

5- Toán tử kết hợp:

Khi tạo mã PHP, chúng ta sẽ thường nhận thấy cần phải tăng hoặc giảm lượng biến một số nguyên nào đó. Bạn sẽ thường thực hiện điều này khi chúng ta đếm 1 giá trị nào đó trong vòng lặp.

Phép Toán	Ví dụ	Lý Giải
++	\$a++	Bằng với \$a= \$a + 1
--	\$a--	Bằng với \$a= \$a - 1
+=	\$a+= \$b	Bằng với \$a= \$a + \$b
-=	\$a-= \$b	Bằng với \$a= \$a - \$b
=	\$a= \$b	Bằng với \$a= \$a * \$b
/=	\$a/= \$b	Bằng với \$a= \$a / \$b

B- Các biểu thức cơ bản trong PHP:

1- Biểu thức điều kiện:

Là biểu thức dùng kiểm tra 1 sự kiện. Nếu chúng thỏa điều kiện đó thì sẽ thực thi một hành động. Ngược lại sẽ là một hành động khác.

Cú pháp:

If(Điều kiện)

{

hành động

}

Ví dụ:

```

1  <?php
2  $a=5;
3  $b=7;
4  if($a < $b)
5  {
6      echo "Bien A co gia tri nho hon bien B";
7  }
8  else
9  {
10     echo "Bien A co gia tri lon hon bien B";
11 }
12 ?>

```

2- Vòng lặp trong PHP:

a- While()....

Phép lặp này yêu cầu phải thỏa mãn điều kiện thì mới thực thi được vòng lặp

Cú pháp:

While(điều kiện)

{

Hành động – thực thi

}

Ví dụ:

```

1  <?php
2  $a=1;
3  while($a < 10)
4  {
5      echo "gia tri $a la";
6      $a++;
7  }
8  ?>
9

```

b-Do....while():

Phép lặp này sẽ thực thi hành động ít nhất là một lần. Sau đó mới tiến hành kiểm tra điều kiện.

Cú pháp:

Do

{

Hành động thực thi

} while(điều kiện)

Ví dụ:

```
1  <?php
2  $a=5;
3  do{
4  echo "day la gia tri cua $a";
5  $a++;
6  }while($a >6)
7  ?>
```

c- For():

Phép lặp này là phép toán gộp các tham số. Giúp người lập trình giảm thiểu thời gian phải khai báo biến và các tham số khi thực thi việc lặp dữ liệu.

Cú pháp:

For(giá trị ; điều kiện ; biến tăng hoặc giảm)

{ Hành động }

Ví dụ:

```
1  <?php
2  $a=2;
3  for ($i=1; $i<=10; $i++)
4  {
5      echo"$a x $i =" . $a*$i . "<br>";
6  }
7  ?>
8
```

3- Biểu thức switch case:

Là biểu thức sử dụng để giảm thiểu quá trình xử lý dữ liệu nếu có quá nhiều phép toán if else.

Cú pháp:

Switch(biến)

{

Case giá trị 1: Hành động; Break;

.....

Case giá trị N: Hành động; Break;

Default: Hành động; Break;

```
}
```

Ví dụ:

```
1  <?php
2  $a=5;
3  switch ($a)
4  {
5      case 1: echo "day la gia tri $a"; break;
6      case 2: echo "day la gia tri $a"; break;
7      case 3: echo "day la gia tri $a"; break;
8      case 4: echo "day la gia tri $a"; break;
9      case 5: echo "<font color=red>day la gia tri $a</font>"; break;
10     default: echo "Khong co gia tri phu hop"; break;
11 }
12
13 ?>
```

C- Tổng kết:

Kết thúc bài học này, các bạn ít nhiều đã nắm được những thuộc tính cơ bản của các phép toán học trong PHP, ngoài ra chúng ta cũng từng bước hiểu được cú pháp của từng biểu thức.

Đối với các vòng lặp, chúng ta cần hiểu và vận dụng một cách linh hoạt và mềm dẻo, phân biệt được while và do...while. Sử dụng hàm switch trong trường hợp có quá nhiều giá trị if...else trả về.

D- Bài tập áp dụng:

Bài tập 1:

Viết 1 trang web có giá trị từ 1->20. Hãy xuất ra trình duyệt những số chẵn nằm trong khoảng 1->20 đó.

Bài tập 2:

Xây dựng 1 website thỏa yêu cầu xuất ra bảng cửu chương từ 2 ->10.

Bài 4: Xử lý giá trị form trong PHP



Một trong những ứng dụng quan trọng của PHP đó là giúp tương tác xử lý dữ liệu trên form của người sử dụng. Nhằm mục đích giúp cập nhật thông tin một cách linh động và dễ dàng quản lý chung hơn bởi sự kết hợp tuyệt vời của cơ sở dữ liệu. Tuy nhiên để làm được điều ấy PHP yêu cầu form phải đáp ứng 1 số quy định chung đặt ra.

Chúng ta cùng phân tích thẻ form trong **HTML** sau:

```

1
2 <form name="Tên form" action="link xử lý" method="Phương thức">
3

```

Chúng ta thấy rằng 1 form phải bao gồm:

Tên form để dễ dàng tách biệt với giá trị của chúng.

Action: hành động chuyển tiếp đến link xử lý.

Method: Là phương thức truyền bao gồm POST và GET.

Ví dụ:

-

```

1 <form action = "check.php" method = post name=reg>
2 Please type your name here : <BR>
3 <input type = text name = "username"><BR><BR>
4 <input type = submit value = "Submit Data" name="add">
5 </form>
6

```

Như vậy ta thấy rằng. Đoạn code trên làm những việc sau.

Đầu tiên khi khách nhập liệu username vào thì chúng sẽ chuyển tới trang check.php để tiến hành xử lý thông tin. Trên phương thức POST, với tên form là reg. Giá trị mà chúng ta gửi là username vừa nhập liệu.

Vậy làm cách nào để chúng ta lấy được giá trị vừa nhập liệu nào ?.

PHP cho phép ta lấy giá trị dựa vào 2 phương thức POST và GET.

Đối với **POST** ta có : **`$_POST['Giá trị']`**

Đối với **GET** ta có : **`$_GET['Giá trị']`**

Vậy với đoạn code trên có thể lấy được biến xử lý là : `$_POST['username']`;

Username là tên của field mà người sử dụng nhập liệu vào.

1 Phương thức GET:

Phương thức này cũng được dùng để lấy dữ liệu từ form nhập liệu. Tuy nhiên nhiệm vụ chính của nó vẫn là lấy nội dung trang dữ liệu từ web server.

Ví dụ:

-

Với url sau: shownews.php?id=50

Vậy với trang shownews ta dùng hàm `$_GET['id']` sẽ được giá trị là 50.

2- Phương thức POST:

-

Phương thức này được sử dụng để lấy dữ liệu từ form nhập liệu. Và chuyển chúng lên trình chủ webserver.

Ví dụ:

Xây dựng 1 trang HTML với nội dung gồm form nhập liệu họ và tên. Sau đó dùng 1 file php để xuất ra thông tin họ và tên mà người sử dụng vừa nhập liệu.

Đáp Án:

Tạo file userform.htm với nội dung sau:

```
2 <html>
3 <form action = "processform.php" method = post>
4 Please type your name here : <BR>
5 <input type = text name = "username"><BR><BR>
6 <input type = submit value = "Submit Data">
7 </form>
8 </html>
9
```

Tạo file processform.php để xuất ra dữ liệu

```
2 <?
3 echo "welcome, " . $_POST['username'] . "!";
4 ?>
5
```

Tổng Kết:

Kết thúc bài này các bạn đã nắm được kỹ thuật kiểm tra thông tin dựa trên PHP. Một trong những vấn đề không thể thiếu đối với bất kỳ một website động nào. Chúng cho ta kiểm tra tính hợp lệ của người sử dụng 1 cách dễ dàng bởi sự tùy biến trong các biểu thức của PHP.

Bài tập áp dụng :

Tạo 1 trang web với hộp thoại nhập liệu username và password. Nếu người sử dụng nhập thông tin username/password là admin/12345 thì xuất ra thông báo "welcome, admin" với kiểu chữ Tahoma, màu đỏ. Ngược lại nếu nhập sai thì xuất thông báo "Username hoặc password sai. Vui lòng nhập lại".

Đáp án:

Tạo trang login.html với nội dung sau:

```

2  <html>
3  <head>
4  <title>login page</title>
5  </head>
6  <body>
7  <form action="checklogin.php" method=POST>
8  Username <input type=text name="user" size=15>
9  <br>Password <input type=password name="pass" size=15>
10 <br><input type=submit name=submit value="Login">
11 </form>
12 </body>
13 </html>

```

Tiếp tục tạo trang checklogin.php với nội dung sau:

```

2  <?php
3  $username=$_POST['user'];
4  $password=$_POST['pass'];
5  if($username == "admin" && $password="12345")
6  {
7      echo "<font color=red>Welcome to, ".$username."<font>";
8  }
9  else
10 {
11     echo "<font color=red>Username hoặc password không chính xác,
12 vui lòng đăng nhập lại<font>";
13 }
14 ?>

```

Bài 5: Tìm hiểu quy trình làm việc trên file trong PHP



Một trong những tác vụ đặc biệt của PHP đó là cho phép xử lý dữ liệu trực tiếp thông qua quá trình nhận và đọc nội dung trên 1 file dữ liệu. Điều này giúp cho PHP trở nên tinh tế và dễ tùy biến hơn khi xử lý 1 lượng dữ liệu có quy mô lớn. Việc thao tác mở, đọc, ghi, đóng file này cũng có ý nghĩa tuần tự như bạn đang làm việc trực tiếp trên 1 file dữ liệu thực thụ.

1- Đóng, mở 1 file trong PHP:

-

Để mở 1 file ta sử dụng cú pháp sau: fopen("Đường dẫn", thuộc tính).

Trong đó Đường dẫn chính là đường dẫn tới file cần mở.

Thuộc tính bao gồm các quyền hạn cho phép thao tác trên file đó như thế nào.

Các thuộc tính cơ bản :

Mode	Diễn giải
r	Read only
r+	Read_Write
w	Write Only
w+	Write_Read. Nếu file này tồn tại, nội dung cũ sẽ bị xoá đi, còn nếu không tồn tại nó sẽ được tạo mới
a	Mở dưới dạng append dữ liệu, chỉ có write, nếu file tồn tại, sẽ ghi tiếp vào phần dưới của nội dung có sẵn, nếu file không tồn tại sẽ được tạo mới.
a+	Mở dưới dạng append dữ liệu (write và read), nếu file tồn tại, dữ liệu sẽ ghi tiếp vào phần bên dưới của nội dung cũ, nếu file không tồn tại sẽ được tạo mới
b	Mở dưới chế độ file binary

Ví dụ:

-

```
<?php
```

```
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
```

```
?>
```

Tương tự như thế, để đóng 1 file ta có cú pháp như sau: fclose(file vừa mở)

Ví dụ:

-

```
<?php
```

```
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
```

```
fclose($fp);
```

?>

Việc mở và đóng này không có ý nghĩa là chúng đã được đọc. Muốn đọc được nội dung của file chúng ta lại tiếp tục với thao tác lấy dữ liệu từ file nữa.

2- Đọc và ghi file trong PHP.

a) Đọc 1 file trong PHP

PHP cho ta nhiều sự lựa chọn trong việc đọc 1 file. Có nhiều hình thức hỗ trợ nhưng hiện nay 2 hình thức phổ biến nhất vẫn là đọc file theo từng dòng và đọc file theo từng ký tự.

- Đọc file theo từng dòng:

Cú pháp : fgets(file vừa mở).

Ví dụ:

-

<?php

```
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
```

```
echo fgets($fp);
```

```
fclose($fp);
```

?>

- Đọc file theo từng ký tự:

Cú pháp : fgetc(file vừa mở).

Ví dụ:

-

<?php

```
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
```

```
echo fgetc($fp);
```

```
fclose($fp);
```

```
?>
```

Quy trình đọc sẽ diễn ra theo từng yêu cầu của cú pháp sử dụng. Nhưng sẽ có sự ràng buộc bởi việc kiểm tra đã đến cuối file chưa ?.

Ở đây chúng ta dùng cú pháp sau: **feof(file vừa mở)**

Ví dụ:

-

```
<?php
```

```
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
```

```
while(!feof($fp))
```

```
{
```

```
echo fgets($fp);
```

```
}
```

```
fclose($fp);
```

```
?>
```

b) Ghi 1 file trong PHP

PHP cung cấp cho ta 1 cú pháp nhỏ để ghi dữ liệu vào 1 file

Cú pháp cơ bản : fwrite("file vừa mở", "Nội dung cần ghi vào file")

-

Ví dụ:

```
<?php

$fp=fopen("test.txt",a)or exit("khong tim thay file can mo");

$news="Trung Tam Tin hoc Viet Chuyenn";

fwrite($fp,$news);

fclose($fp);

?>
```

Tổng kết:

Việc sử dụng file một cách thành thạo sẽ giúp bạn dễ dàng vận hành các ứng dụng mang quy mô vừa và nhỏ như: website nhiều ngôn ngữ, bộ đếm,... và cả những công nghệ web mới như XML một cách dễ dàng. Qua bài học này chúng ta cũng hiểu được nguyên lý hoạt động, trình tự xử lý 1 file dữ liệu khi chúng được triệu gọi trong tài liệu PHP.

Bài tập áp dụng:

Xây dựng bộ đếm cho website, mỗi khi khách truy cập thì sẽ tự động tăng lên 1 giá trị.

Bài 6: Tổng quan về mảng và các hàm hỗ trợ trong PHP



Cũng như bao ngôn ngữ lập trình khác. PHP cũng tồn tại một loại dữ liệu được gọi là mảng. Cụ thể hơn Mảng là 1 biến danh sách. Nghĩa là, một biến vốn chứa nhiều phần tử được tạo bởi các số hoặc chuỗi. Nó cho phép chúng ta lưu trữ, sắp xếp thứ tự và truy cập nhiều giá trị với một tên. Và cần lưu ý rằng một phần tử mảng luôn bắt đầu với giá trị là 0.

1- Định nghĩa mảng trong PHP:

Với cú pháp giống với java ta có cách gọi 1 mảng như sau:

```
$tên_biến= array("Kenny","Maria","Julia","Kenvin")
```

Như vậy khi gọi giá trị trong mảng ta có:

```
Echo $tên_biến[2]; // Giá trị sẽ cho là Julia.
```

Ngoài cách khai báo ở trên chúng ta cũng có thể khai báo bằng cách khác như sau:

```
$tên_biến[] = "Kenny";
```

```
$tên_biến[] = "Gillian";
```

```
$tên_biến[] = "Charlene";
```

```
$tên_biến[] = "Calvin"
```

Chúng ta không cần thêm ký tự vào. Vì PHP đã tự xử lý cho chúng ta phần đó.

Để thêm 1 phần tử vào mảng chúng ta cũng có thể sử dụng như sau:

```
$tên_biến[] = "Jiro";
```

Như vậy nếu chúng ta gọi giá trị: `echo $tên_biến[4]` thì giá trị sẽ là jiro. Vì nó được PHP thêm vào sau cùng và ngầm hiểu là giá trị tiếp theo.

Ví dụ:

```
<?php
```

```
$a=array("Kenny","Maria","Julia","Kenvin");
```

```
echo $a[2]; //Xuất ra giá trị Julia
```

```
?>
```

2- Khái niệm mảng kết hợp trong PHP

Là các mảng được tạo index bằng các chuỗi , chúng được gọi là các mảng kết hợp.

Ví dụ:

```
<?
```

```
$a= array (name => "Kenny", job => "Teacher", age=>"45", email =>  
"webmaster@vietchuyen.com.vn")
```

```
?>
```

Như vậy để truy cập giá trị của mảng trên chúng ta không thể sử dụng cách cũ được.

Ví dụ: Để lấy giá trị của tuổi tôi không thể sử dụng cách trên : **echo \$a[2]**.

Vì chúng ta đã gán giá trị của index cho 1 tên gọi khác. Do vậy để lấy giá trị của tuổi ta sử dụng như sau: **echo \$a[age]** // Kết quả sẽ cho ra 45.

```
<?
```

```
$tên_biến= array (name => "Kenny", job => "Teacher", age=>"45", email =>  
"webmaster@vietchuyen.com.vn")
```

```
echo $a[age];
```

```
?>
```

3- Phép lặp trong mảng:

Cú pháp:

```
foreach($array as $temp)
```

```
{ Hành Động }
```

Trong đó \$array là mảng mà ta muốn thực thi việc lặp dữ liệu, và \$temp là một biến trong đó ta sẽ tạm thời lưu trữ mỗi phần tử.

Ví dụ:

```
<?php  
  
$name= array("Kenny", "Jiro", "Calvin", "Gillian");  
  
foreach ($name as $test)  
  
{ echo "$test<br>"; }  
  
?>
```

b) Lặp lại qua một mảng kết hợp:

Cú pháp:

```
Foreach($array as $key=>$value)  
  
{ Hành Động }
```

Trong đó \$array là mảng mà ta muốn thực thi việc lặp dữ liệu, \$key là một biến vốn tạm thời chứa mỗi khóa, và \$value là một biến vốn tạm thời chứa mỗi giá trị.

Ví dụ:

```
<?php  
  
$person= array(name=>"Kenny", job=>"Teacher", email=>"whiletionvn@gmail.com",  
age=>"38");  
  
foreach($person as $key=>$test)  
  
{  
  
echo "Key: $key. Gia Tri: $test<br>";  
  
}  
  
?>
```

4- Các hàm hỗ trợ trong PHP:

+ Hàm gộp mảng:

Cú pháp: `array_merge($mang1, $mang2);`

+ Hàm tách mảng:

Cú pháp: `array_slice($mang, vị trí tách, số lượng tách);`

+ Hàm sắp xếp mảng:

Cú pháp: `sort($mảng);`

+ Hàm sắp xếp mảng theo chỉ mục:

Cú pháp: `ksort($mảng);`

+ Hàm tìm phần tử có tồn tại trong mảng hay không ?.

Cú pháp: `in_array(giá trị tìm, $mảng);`

5 - Tổng kết:

Qua bài học này, chúng ta tìm hiểu kỹ về các thành phần của mảng một chiều, nhiều chiều và các hàm cơ bản về xử lý dữ liệu. Hiểu rõ cấu trúc mảng, sự sắp xếp đồng bộ sẽ giúp ích cho các bạn rất nhiều trong những ứng dụng lớn.

Bài tập áp dụng :

- 1- Xây dựng website có tính năng upload cho phép upload cùng lúc 10 file dữ liệu.
- 2- Xây dựng website có tính năng upload, và chỉ cho phép upload file có định dạng hình ảnh như gif, jpg. Ngoài ra các tính năng khác đều có thông báo lỗi không cho upload.

Bài 7: Khái niệm cơ bản về Cookie và session trong PHP



Cookie và session là hai phương pháp sử dụng để quản lý các phiên làm việc giữa người sử dụng và hệ thống. Việc quản lý phiên làm việc này sẽ giúp bạn tạo ra sự chứng thực hiệu quả bởi việc xác nhận thông tin trước khi truy cập vào một phân vùng cố định. Ngoài ra, việc quản lý tốt phiên làm việc cũng giúp người truy cập cảm thấy dễ dàng sử dụng dịch vụ của trang web cho những lần truy cập sau. Bởi cơ chế quản lý phiên làm việc ghi nhận lại quá trình truy cập của người sử dụng khi họ thăm viếng trang web của bạn lần đầu.

1- Tổng quan về cookie:

Cookie là 1 đoạn dữ liệu được ghi vào đĩa cứng hoặc bộ nhớ của máy người sử dụng. Nó được trình duyệt gửi ngược lên lại server mỗi khi browser tải 1 trang web từ server.

Những thông tin được lưu trữ trong cookie hoàn toàn phụ thuộc vào website trên server. Mỗi website có thể lưu trữ những thông tin khác nhau trong cookie, ví dụ thời điểm lần cuối ta ghé thăm website, đánh dấu ta đã login hay chưa, v.v...

Cookie được tạo ra bởi website và gửi tới browser, do vậy 2 website khác nhau (cho dù cùng host trên 1 server) sẽ có 2 cookie khác nhau gửi tới browser. Ngoài ra, mỗi browser quản lý và lưu trữ cookie theo cách riêng của mình, cho nên 2 browser cùng truy cập vào 1 website sẽ nhận được 2 cookie khác nhau.

a -Thiết lập cookie:

Để thiết lập cookie ta sử dụng cú pháp:

```
Setcookie("tên cookie","giá trị", thời gian sống)
```

Tên cookie là tên mà chúng ta đặt cho phiên làm việc.

Giá trị là thông số của tên cookie.

Ví dụ:

```
Setcookie("username","admin", time() +3600)
```

Như ví dụ trên ta thấy với tên là username và giá trị là admin, có thời gian sống là 1 giờ tính từ thời điểm thiết lập.

Chú ý: Kịch bản cookie phải đặt trên mọi giá trị trả về bao gồm thẻ HTML và lệnh echo.

Kịch bản cookie phải đặt trên mọi giá trị trả về bao gồm thẻ HTML và lệnh echo.

b - Sử dụng cookie:

Để sử dụng lại cookie vừa thiết lập, chúng ta sử dụng cú pháp:

Cú pháp: \$_COOKIE["tên cookies"]

Tên cookie là tên mà chúng ta thiết lập phía trên.

Ví dụ:

Tạo trang cookie.php với nội dung sau:

```
<?php
setcookie("name","Kenny Huy",time() + 3600);
?>

<html>

<head>

<title>Test page 1</title></head>

<body>

<b><a href=cookie2.php>Click here</a></b>
```

```
</body>
```

```
</html>
```

Tiếp tục tạo trang cookie2.php với nội dung sau:

```
<html>
```

```
<head><title>Result Page</title></head>
```

```
<body>
```

```
<?php
```

```
echo "Ten cua ban la <b>".$_COOKIE['name']. "</b>";
```

```
?>
```

```
</body>
```

```
</html>
```

c- Hủy Cookie:

Để hủy 1 cookie đã được tạo ta có thể dùng 1 trong 2 cách sau:

+ Cú pháp: setcookie("Tên cookie")

Gọi hàm setcookie với chỉ duy nhất tên cookie mà thôi

+ Dùng thời gian hết hạn cookie là thời điểm trong quá khứ.

Ví dụ: setcookie("name", "Kenny Huy", time()-3600);

Ví dụ:

Tiếp tục tạo trang cookie3.php với nội dung sau:

```
<?php
setcookie("name","Kenny Huy",time()-360);
?>

<html>

<head>

<title>Test page 1</title></head>

<body>

<b><a href=cookie2.php>Click here</a></b>

</body>

</html>
```

2- Tổng quan về session:

Một cách khác quản lý người sử dụng là session. Session được hiểu là khoảng thời gian người sử dụng giao tiếp với 1 ứng dụng. Một session được bắt đầu khi người sử dụng truy cập vào ứng dụng lần đầu tiên, và kết thúc khi người sử dụng thoát khỏi ứng dụng. Mỗi session sẽ có được cấp một định danh (ID) khác nhau và nội dung được lưu trong thư mục thiết lập trong file php.ini (tham số session.save_path).

a- Thiết lập session:

Để thiết lập 1 session ta sử dụng cú pháp: session_start()

Đoạn code này phải được nằm trên các kịch bản HTML. Hoặc những lệnh echo, printf.

Để thiết lập 1 giá trị session, ngoài việc cho phép bắt đầu thực thi session. Chúng ta còn phải đăng ký 1 giá trị session. Để tiện cho việc gán giá trị cho session đó.

Ta có cú pháp sau: session_register("Name")

Ví dụ:

```
<?php  
Session_start();  
Session_register("username");  
?>
```

b- Sử dụng giá trị của session:

Giống với cookie. Để sử dụng giá trị của session ta sử dụng mã lệnh sau:

Cú pháp: \$_SESSION["name"]

\$_SESSION["name"]

Với Name là tên mà chúng ta sử dụng hàm session_register("name") để khai báo.

Ví dụ:

Tạo trang session.php với nội dung sau:

```
<?php  
session_start();  
session_register("name");  
$_SESSION["name"] = "Kenny Huy";
```

```
?>

<html>

<head>

<title>Test page 1</title></head>

<body>

<b><a href=session2.php>Click here</a></b>

</body>

</html>
```

Tạo trang session2.php với nội dung sau:

```
<?

session_start();

?>

<html>

<head><title>Result Page</title></head>

<body>

<?php

echo "Ten của bạn là <b>".$_SESSION["name"]."</b>";

?>

</body>

</html>
```

c- Hủy bỏ session:

Để hủy bỏ giá trị của session ta có những cách sau:

`session_destroy()` // Cho phép hủy bỏ toàn bộ giá trị của session

`session_unset()` // Cho phép hủy bỏ session .

Ví dụ:

Tạo trang session3.php với nội dung sau:

```
<?php
session_start();
session_destroy();

?>

<html>

<head>

<title>Test page 1</title></head>

<body>

<b><a href=session2.php>Click here</a></b>

</body>

</html>
```

3- Tổng Kết:

Sau bài học này chúng ta đã nắm được cách điều khiển phiên làm việc giữa cookie và session. Sử dụng chúng trong từng trường hợp cụ thể. Từ đó có thể áp dụng để viết những ứng dụng nhỏ như kiểm soát người đăng nhập, làm giỏ hàng online,.....

Bài tập áp dụng:

Xây dựng website sử dụng tính năng session thỏa mãn những yêu cầu sau:

- 1- Tạo trang đăng nhập với username/password là : admin/vietchuyen. Nếu user đăng nhập thành công thì sẽ xuất ra dòng welcome, admin.
- 2- Tạo trang đăng xuất thoát khỏi phần quản trị

Bài 8: Khái niệm về hàm và gọi lại hàm trong PHP



Để giảm thời gian lặp lại 1 thao tác code nhiều lần, PHP hỗ trợ người lập trình việc tự định nghĩa cho mình những hàm có khả năng lặp lại nhiều lần trong website. Việc này cũng giúp cho người lập trình kiểm soát mã nguồn một cách mạch lạc. Đồng thời có thể tùy biến ở mọi trang. Mà không cần phải khởi tạo hay viết lại mã lệnh như HTML thuần.

1- Hàm tự định nghĩa:

Cú pháp:

```
function functiono_name()
```

```
{
```

```
//Lệnh thực thi
```



```
}
```

Tên hàm có thể là một tổ hợp bất kỳ những chữ cái, con số và dấu gạch dưới, nhưng phải bắt đầu từ chữ cái và dấu gạch dưới.

Ví dụ:

```
<?php  
  
function name()  
{  
  
$name= array("Kenny", "Jiro", "Calvin", "Gillian");  
  
foreach ($name as $test)  
{ echo "$test<br>"; }  
  
}  
  
name();  
  
?>
```

2- Hàm tự định nghĩa với các tham số:

Cú pháp:

```
function function_name($gt1,$gt2)  
{  
  
//Hành động  
  
}
```

Ví dụ:

```
<?php
function indulieu($text)
{
    echo "$text<br>";
}
indulieu("welcome");
indulieu("who are you ?");
?>
```

3- Hàm tự định nghĩa với giá trị trả về:

Cú pháp:

```
function function_name(Có hoặc không có đối số)
{
    //Thực thi
    return giatri;
}
```

Ví dụ:

```
<?php  
  
function tinh tong($a,$b)  
  
{  
  
$total=$a+$b;  
  
return $total;  
  
}  
  
echo tinh tong(19,31)  
  
?>
```

4-Gọi lại hàm trong PHP:

PHP cung cấp nhiều hàm cho phép triệu gọi lại file. Như hàm include("URL đến file"), require("URL Đến file").

Ngoài hai cú pháp trên còn có include_once(), require_once(). Hai hàm này cũng có trách nhiệm gọi lại hàm. Những chúng sẽ chỉ gọi lại duy nhất 1 lần mà thôi.

Ví dụ:

Tạo file top.html với nội dung:

```
<html><head><title>Welcome to you</title></head>  
  
<body>  
  
<table border=1 width=700>
```

```
<tr><td colspan=5 align=center>Banner</td></tr>
```

```
<tr>
```

```
<td align=center> Home </td>
```

```
<td align=center> News </td>
```

```
<td align=center> Music </td>
```

```
<td align=center> Download</td>
```

```
<td align=center> Contact</td></tr>
```

Tạo file body.html với nội dung:

```
<tr><td colspan=5 align=center>
```

```
Noi dung website
```

```
</td>
```

```
</tr>
```

Tạo file bottom.html với nội dung:

```
<tr>
```

```
<td colspan=5 align=center> Copyright@ abc.com</td>
```

```
</tr>
```

```
</table>
```

```
</body>
```

```
</html>
```

Tạo trang index.php với nội dung:

```
<?php  
include("top.html");  
include("body.html");  
include("bottom.html");  
?>
```

Tổng kết:

Kết thúc bài học này, hẳn các bạn đã có những khái niệm cơ bản về hàm và cách sử dụng lại mã PHP. Trên ứng dụng thực tế việc dùng các hàm để triệu gọi đóng vai trò khá quan trọng, nó giúp mã nguồn của người sử dụng logic và dễ dàng chỉnh sửa hơn, việc ứng dụng nó trên website như thế nào cũng sẽ tùy biến hơn so với cách viết trực tiếp trên từng file.

Bài tập áp dụng:

- 1- Thiết kế website hoàn chỉnh bằng việc phân tách các file.
- 2- Xây dựng website với yêu cầu sau. Cho phép người sử dụng chọn mua số lượng hàng. Với giá tương ứng với từng mặt hàng. Sau đó xuất tổng tiền(đã bao gồm thuế VAT 10%).

Bài 9: Cơ bản về ngôn ngữ SQL và Mysql



Mysql là hệ quản trị dữ liệu miễn phí, được tích hợp sử dụng chung với apache, PHP. Chính yếu tố phát triển trong cộng đồng mã nguồn mở nên

mysql đã qua rất nhiều sự hỗ trợ của những lập trình viên yêu thích mã nguồn mở. Mysql cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. Nhưng Mysql không bao quát toàn bộ những câu truy vấn cao cấp như SQL. Về bản chất Mysql chỉ đáp ứng việc truy xuất đơn giản trong quá trình vận hành của website nhưng hầu hết có thể giải quyết các bài toán trong PHP.

1- Cách khởi động và sử dụng MYSQL.

Chúng ta sử dụng command như sau:

mysql -hname -uuser -ppass

Để truy cập vào cơ sở dữ liệu.

Hoặc sử dụng bộ appserv để vào nhanh hơn theo đường dẫn sau:

Start/ Appserv/ Mysql command Line client

Sau đó nhập password mà chúng ta đã đặt vào.

2- Những định nghĩa cơ bản:

a) Định nghĩa cơ sở dữ liệu, bảng, cột:

Cơ sở dữ liệu: là tên của cơ sở dữ liệu chúng ta muốn sử dụng

Bảng: Là 1 bảng giá trị nằm trong cơ sở dữ liệu.

Cột là 1 giá trị nằm trong bảng. Dùng để lưu trữ các trường dữ liệu.

Thuộc tính

Ví dụ:

Bảng user	
User_id	1234
First_name	Nguyen
Last_name	Hai Anh
Username	Kenny
Password	12345
Email	kenny@qhonline.info
Date	2007-12-12 16:25:30

Như vậy ta có thể hiểu như sau:

1 cơ sở dữ liệu có thể bao gồm nhiều bảng.

1 bảng có thể bao gồm nhiều cột

1 cột có thể có hoặc không có những thuộc tính.

b) Định nghĩa 1 số thuật ngữ:

NULL : Giá trị cho phép rỗng.

AUTO_INCREMENT : Cho phép giá trị tăng dần (tự động).

UNSIGNED : Phải là số nguyên dương

PRIMARY KEY : Cho phép nó là khóa chính trong bảng.

c)Loại dữ liệu trong Mysql:

Ở đây chúng tôi chỉ giới thiệu 1 số loại thông dụng: 1 số dữ liệu khác có thể tham khảo trên trang chủ của mysql.

Kiểu dữ liệu	Mô Tả
Char	Định dạng text có chiều dài từ 0->255
Varchar	Định dạng text có chiều dài từ 0->255
Text	Định dạng text có chiều dài 0->65535
Longtext	Định dạng text có chiều dài 0->4294967215
INT	Định dạng số có chiều dài từ 0->4294967215
Float	Định dạng số thập phân có chiều dài nhỏ
Double	Định dạng số thập phân có chiều dài lớn
Date	Định dạng thời gian theo định dạng: YYYY-MM-DD
DateTime	Định dạng thời gian theo định dạng: YYYY-MM-DD HH:MM:SS

3- Những cú pháp cơ bản:

Cú pháp tạo 1 cơ sở dữ liệu:

```
CREATE DATABASE tên_cơ_sở_dữ_liệu;
```

Cú pháp sử dụng cơ sở dữ liệu: Use tên_database;

Cú pháp thoát khỏi cơ sở dữ liệu: Exit

Cú pháp tạo 1 bảng trong cơ sở dữ liệu:

```
CREATE TABLE user (<tên_cột> <mô_tả>,...,<tên_cột_n>.....<mô_tả_n>)
```

Ví dụ:


```
mysql> create table user(user_id INT(15) UNSIGNED NOT NULL AUTO_INCREMENT,  
username VARCHAR(255) NOT NULL, password CHAR(50) NOT NULL, email  
VARCHAR(200) NOT NULL, PRIMARY KEY (user_id));
```

Hiển thị có bao nhiêu bảng: show tables;

Hiển thị có bao nhiêu cột trong bảng: show columns from table;

Thêm 1 cột vào bảng :

```
ALTER TABLE tên_bảng ADD <tên_cột> <thuộc_tính> AFTER <tên_cột>
```

Ví dụ:

```
mysql> alter table user add sex varchar(200) NOT NULL after email;
```

4- Thêm giá trị vào bảng:

Cú pháp:

```
INSERT INTO Tên_bảng(tên_cột) VALUES(Giá_trị_tương_ứng);
```

Ví dụ:

```
mysql> insert into user(username,password,email,sex,home)  
values("Lanna","12345","lanna@yahoo.com","F","www.abc.com");
```

5- Truy xuất dữ liệu:

Cú pháp:

```
SELECT tên_cột FROM Tên_bảng;
```

Ví dụ:

```
mysql> select user_id,username from user;
```

6- Truy xuất dữ liệu với điều kiện:

Cú pháp:

SELECT tên_cột FROM Tên_bảng WHERE điều kiện;

Ví dụ:

```
mysql> select user_id,username from user where user_id=2;
```

7- Truy cập dữ liệu và sắp xếp theo trình tự

Cú pháp:

SELECT tên_cột FROM Tên_bảng

WHERE điều kiện (có thể có where hoặc không)

ORDER BY Theo quy ước sắp xếp.

Trong đó quy ước sắp xếp bao gồm hai thông số là ASC (từ trên xuống dưới), DESC (từ dưới lên trên).

```
mysql> select user_id,username from user order by username ASC ;
```

8- Truy cập dữ liệu có giới hạn :

Cú pháp:

SELECT tên_cột FROM Tên_bảng

WHERE điều kiện (có thể có where hoặc không)

LIMIT vị trí bắt đầu, số record muốn lấy ra

Ví dụ:

```
mysql> select user_id,username from user order by username ASC limit 0,10 ;
```

9- Cập nhật dữ liệu trong bảng:

Cú pháp:

Update tên_bảng set tên_cột=Giá trị mới

WHERE (điều kiện).

Nếu không có ràng buộc điều kiện, chúng sẽ cập nhật toàn bộ giá trị mới của các record trong bảng.

Ví dụ:

```
mysql> update user set email="admin@qhonline.info" where user_id=1 ;
```

10- Xóa dữ liệu trong bảng:

Cú pháp:

DELETE FROM tên_bảng WHERE (điều kiện).

Nếu không có ràng buộc điều kiện, chúng sẽ xóa toàn bộ giá trị của các record trong bảng.

Ví dụ

```
mysql>delete from user where user_id=1 ;
```

Bài 10: Kết hợp PHP và MYSQL trong ứng dụng website



Ở bài trước, chúng ta đã cùng nghiên cứu về các [cú pháp sql](#) và [Mysql cơ bản](#) bao gồm việc tạo bảng, tạo kết nối, thêm, sửa, xóa các dòng dữ liệu trong cơ sở dữ liệu. Và tiếp theo bài này, chúng ta sẽ cùng tìm hiểu về cách sử dụng mysql kết hợp với PHP để xuất ra dữ liệu như thế nào.

Để làm việc với mysql và PHP chúng ta cần nắm 6 hàm cơ bản:

1- Kết nối cơ sở dữ liệu:

Cú pháp:

```
mysql_connect("hostname","user","pass")
```

2- Lựa chọn cơ sở dữ liệu:

Cú pháp:

```
mysql_select_db("tên_CSDL")
```

Ví dụ:

```
$conn=mysql_connect("localhost","root","root") or die(" không thể kết nối");  
mysql_select_db("demo");
```

3- Thực thi câu lệnh truy vấn:

Cú pháp:

```
mysql_query("Câu truy vấn ở đây");
```

4- Đếm số dòng dữ liệu trong bảng:

Cú pháp:

```
mysql_num_rows();
```

5- Lấy dữ liệu từ bảng đưa vào mảng:

Cú pháp:

```
mysql_fetch_array();
```

6- Đóng kết nối cơ sở dữ liệu:

Cú pháp:

```
mysql_close();
```

Ví dụ áp dụng:

Tạo cơ sở dữ liệu dựa trên từng đoạn code sau:

```
mysql -hlocalhost -uroot -proot
```

```
mysql>create database demo_mysql;
```

```
mysql> use demo_mysql;
```

```
mysql>create table user(id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, username  
VARCHAR(50) NOT NULL, password CHAR(50) NOT NULL, PRIMARY KEY(id));
```

Tạo trang test.php. Đầu tiên chúng ta sẽ kết nối cơ sở dữ liệu.

```
<?
```

```
$conn=mysql_connect("localhost", "root", "root") or die("can't connect database");
```

```
mysql_select_db("demo_mysql",$conn);
```

```
?>
```

Tiếp đến viết câu truy vấn lấy ra tất cả user từ database.

```
<?
```

```
$sql="select * from user";
```

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

```
?>
```

Kiểm tra xem trong bảng dữ liệu đã tồn tại user nào chưa ?. Nếu chưa thì xuất ra thông báo lỗi, ngược lại thì đưa chúng vào mảng và lặp cho đến hết bảng dữ liệu.

```
<?
```

```
if(mysql_num_rows($query) == 0)

{

echo "Chua co du lieu";

}

else

{

while($row=mysql_fetch_array($query))

{

echo $row[username] ." - ".$row[password]."<br />";

}

}

?>
```

Và cuối cùng chúng ta đóng kết nối và kết thúc thao tác xử lý.

```
<?
```

```
mysql_close($conn);
```

```
?>
```

Và cuối cùng là file hoàn chỉnh của ứng dụng trên:

```
<?
$conn=mysql_connect("localhost", "root", "root") or die("can't connect database");

mysql_select_db("demo_mysql",$conn);

$sql="select * from user";

$query=mysql_query($sql);

if(mysql_num_rows($query) == 0)
{
echo "Chua co du lieu";
}

else
{
while($row=mysql_fetch_array($query))
{
echo $row[username] ." - ".$row[password]."<br />";
}
}

mysql_close($conn);

?>
```

Tổng kết:

Ở bài này chúng ta đã làm việc với sql và các kỹ thuật kết nối với database của PHP. Các hàm PHP hỗ trợ mysql hiện nay có rất nhiều. Tuy nhiên về tính năng cũng như phương thức sử dụng thì chúng cũng giống nhau. Do vậy các bạn không nên lo lắng khi làm việc với những hàm mà chúng ta đang nghiên cứu.

Bài 11: Viết ứng dụng đăng nhập bằng PHP và MYSQL



Ở những bài trước chúng ta đã học qua các kiến thức trọng yếu về **PHP**. Tuy nhiên, để có thể xây dựng một website hoàn chỉnh bằng những kiến thức đó thì quả thật không đơn giản. Bởi vì các kiến thức qua sách vở và tài liệu dù sao cũng chỉ là những kiến thức tổng quát và thiếu tính khách quan thực tiễn. Nên khi người học lập trình bước vào giai đoạn viết ứng dụng thì thường rất lúng túng. Đó cũng là lý do tôi viết bài này để hướng dẫn các bạn dần làm quen với cách tiếp cận một ứng dụng **PHP và MYSQL** như thế nào.

Trước hết chúng ta phải thiết kế và xây dựng mô hình cơ sở dữ liệu ứng với từng lệnh bên dưới (xem lại bài **ngôn ngữ SQL và MYSQL Căn Bản**) sau đó mới tiến hành triển khai lập trình.

```
1 mysql -hlocalhost -uroot -proot
2 mysql>create database project;
3 mysql>use project;
   mysql>create table user( id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT, username
4 VARCHAR(50) NOT NULL, password CHAR(50) NOT NULL, level CHAR(1) NOT NULL,
   PRIMARY KEY(id));
5 mysql>insert into user(username,password,level) values ("admin","12345","2")
   ("abc","12345","1");
```

Vậy là chúng ta đã có cơ sở dữ liệu như mô hình trên. Tiếp đến chúng ta tạo file **login.php** và thiết kế Form HTML để có màn hình đăng nhập khi người dùng truy cập.

Username:

Password:

<http://www.qhonline.info>

```
1 <form action='login.php' method='post'>
2 Username: <input type='text' name='username' size='25' /><br />
3 Password: <input type='password' name='password' size='25' /><br />
```

```
4 <input type='submit' name='ok' value='Dang Nhap' />
5 </form>
```

Tiếp đến chúng ta tiến hành kiểm tra dữ liệu từ form

```
01 <?php
02 if(isset($_POST['ok']))
03 {
04 $u=$p="";
05 if($_POST['username'] == NULL)
06 {
07 echo "Please enter your username<br />";
08 }
09 else
10 {
11 $u=$_POST['username'];
12 }
13 if($_POST['password'] == NULL)
14 {
15 echo "Please enter your password<br />";
16 }
17 else
18 {
19 $p=$_POST['password'];
20 }
21 }
22 ?>
```

Đoạn code ở trên sẽ kiểm tra xem người dùng có tiến hành nhấn nút đăng nhập hay không. Và nếu có thì chúng ta sẽ xét tiếp tình trạng người dùng có để trống các ô username và password hay không. Nếu có chúng ta sẽ thông báo lỗi ở bên trên form, để người sử dụng tiến hành nhập liệu. Vì phiên bản PHP 5.3 trở lên sẽ yêu cầu ta khai báo biến trước khi sử dụng. Vì thế mặc định ban đầu ta khởi tạo 2 biến \$u và \$p mang giá trị là rỗng.

Kế đến chúng ta kiểm tra xem có tồn tại hai biến \$u và \$p (chỉ khi người dùng đăng nhập thành công thì mới có thể tạo ra 2 biến đó). Tiếp đến chúng ta tiến hành kết nối cơ sở dữ liệu (xem lại bài [**kết hợp PHP & MYSQL trong ứng dụng web**](#)).

```
1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can't connect this
  database");
3 mysql_select_db("project",$conn);
```

4 ?>

Và tiến hành kiểm tra xem username và password người sử dụng vừa nhập có trùng khớp với thông tin có trong cơ sở dữ liệu hay không ?. Nếu không thì chúng ta sẽ báo lỗi ngay. Ngược lại sẽ tiến hành lấy dữ liệu từ bảng và gán vào **session**. Để có thể quản lý phiên làm việc một cách hiệu quả trên mọi trang của khu vực admin (xem lại bài [khái niệm cơ bản về cookie và session](#)).

```
01 <?php
02 if($u && $p)
03 {
04     $conn=mysql_connect("localhost","root","root") or die("can't connect this
database");
05     mysql_select_db("project",$conn);
06     $sql="select * from user where username='".$u.'" and password='".$p.'";
07     $query=mysql_query($sql);
08     if(mysql_num_rows($query) == 0)
09     {
10         echo "Username or password is not correct, please try again";
11     }
12     else
13     {
14         $row=mysql_fetch_array($query);
15         session_start();
16         $_SESSION['userid'] = $row['id'];
17         $_SESSION['level'] = $row['level'];
18     }
19 }
20 ?>
```

Như vậy code hoàn chỉnh cho ứng dụng (file login.php) này là như sau:

[view source](#)
[print?](#)

```
01 <?php
02 if(isset($_POST['ok']))
03 {
04     $u=$p="";
05     if($_POST['username'] == NULL)
06     {
07         echo "Please enter your username<br />";
08     }
09     else
10     {
```

```

11  $u=$_POST['username'];
12  }
13  if($_POST['password'] == NULL)
14  {
15      echo "Please enter your password<br />";
16  }
17  else
18  {
19      $p=$_POST['password'];
20  }
21  if($u && $p)
22  {
23      $conn=mysql_connect("localhost","root","root") or die("can't connect this
database");
24      mysql_select_db("project",$conn);
25      $sql="select * from user where username='".$u."' and password='".$p."'";
26      $query=mysql_query($sql);
27      if(mysql_num_rows($query) == 0)
28      {
29          echo "Username or password is not correct, please try again";
30      }
31      else
32      {
33          $row=mysql_fetch_array($query);
34          session_start();
35          $_SESSION['userid'] = $row[id];
36          $_SESSION['level'] = $row[level];
37
38      }
39  }
40 }
41 ?>
42 <form action='login.php' method='post'>
43 Username: <input type='text' name='username' size='25' /><br />
44 Password: <input type='password' name='password' size='25' /><br />
45 <input type='submit' name='ok' value='Dang Nhap' />
46 </form>

```

Bài 12: Viết ứng dụng thêm thành viên bằng PHP và MYSQL



Ở bài trước, chúng ta đã xây dựng ứng dụng thành viên. Cho phép [người sử dụng thực thi công việc đăng nhập vào hệ thống](#). Trong bài này chúng ta sẽ tiếp tục công việc phát triển ứng dụng trên, để người sử dụng sau khi đăng nhập có những thao tác đặc quyền như thêm, quản lý, sửa, xóa thành viên của trang web. (xem lại [Bài 11: Viết ứng dụng đăng nhập bằng PHP & MYSQL](#)).

Sau khi đăng nhập vào trang admin người dùng sẽ được đẩy sang trang quản lý admin, tại đây chỉ những ai đã đăng nhập đúng với username và password trong cơ sở dữ liệu và có quyền hạn level là 2 thì mới có thể truy xuất vào trang này. Vì vậy chúng ta cần thao tác [kiểm tra session để quản lý phiên làm việc của người sử dụng](#). Đặt đoạn code sau ở những trang bạn muốn kiểm tra quyền truy cập của người sử dụng.

```
01 <?php
02 session_start();
03 if(isset($_SESSION['userid']) && $_SESSION['level'] == 2)
04 {
05 // tại đây thực thi các hoạt động khi đăng nhập thành công.
06 }
07 else
08 {
09 header("location: login.php");
10 exit();
11 }
12 ?>
```

Như vậy, ta tạo trang add_user.php và đặt đoạn code trên, đồng thời viết thêm 1 form thực thi công việc thêm thành viên.

Level: ▼

Username:

Password:

Re-Password:

<http://www.qhonline.info>

```
01 <form action='add_user.php' method='POST'>
02 Level: <select name='level'>
```

```

03 <option value='1'>Member</option>
04 <option value='2'>Admin </option>
05 </select><br />
06 Username: <input type='text' name='username' size='25' /><br />
07 Password: <input type='password' name='password' size='25' /> <br />
08 Re-Password: <input type='password' name='re-password' size='25' /><br />
09 <input type='submit' name='adduser' value='Add New User' />
10 </form>

```

Tương tự như những bài trước, chúng ta cũng lần lượt kiểm tra từng đối tượng. Trước hết cần xem thử người dùng đã nhấn nút Add New user chưa ?, sau đó lại khai báo 2 biến \$u và \$p, mặc định ban đầu ta cho 2 biến này là rỗng. Kế tới ta kiểm tra xem người dùng đã nhập đầy đủ thông tin chưa. Cuối cùng là so sánh giữa password và re-password có giống nhau hay không. Tiếp theo chúng ta gán biến \$l sẽ bằng với giá trị mà khi người dùng chọn level cho user đó. (xem lại **Bài 4: Xử lý giá trị form trong PHP**)

```

01 <?php
02 if(isset($_POST['adduser']))
03 {
04     $u=$p="";
05     if($_POST['username'] == NULL)
06     {
07         echo "Vui long nhap username<br />";
08     }
09     else
10     {
11         $u=$_POST['username'];
12     }
13     if($_POST['password'] != $_POST['re-password'])
14     {
15         echo "Password va re-password khong chinh xac<br />";
16     }
17     else
18     {
19         if($_POST['password'] == NULL )
20         {
21             echo "Vui long nhap password<br />";
22         }
23         else
24         {
25             $p=$_POST['password'];
26         }
27     }

```

```

28 $l=$_POST['level'];
29 }
30 ?>

```

Kế đến chúng ta lại tiếp tục kiểm tra xem có tồn tại biến \$u, \$p và \$l hay không (chỉ khi người dùng nhập liệu thì mới tồn tại cùng lúc 3 biến này). sau khi đã kiểm tra sự tồn tại của 3 biến này chúng ta tiếp tục **dùng PHP kết nối cơ sở dữ liệu** để kiểm tra xem username mà người dùng vừa chọn có trùng với username khác trong cơ sở dữ liệu chưa. nếu trùng thì phải xuất thông báo lỗi yêu cầu họ chọn username khác còn nếu không thì ta lại tiến hành thêm thành viên mới này vào CSDL.

```

01 <?php
02 if($u & $p & $l)
03 {
04     $conn=mysql_connect("localhost","root","root") or die("can't connect this
database");
05     mysql_select_db("project",$conn);
06     $sql="select * from user where username='".$u."'";
07     $query=mysql_query($sql);
08     if(mysql_num_rows($query) != "" )
09     {
10         echo "Username nay da ton tai roi<br />";
11     }
12     else
13     {
14         $sql2="insert into user(username,password,level)
values('".$u."','".$p."','".$l."' )";
15         $query2=mysql_query($sql2);
16         echo "Da them thanh vien moi thanh cong";
17     }
18 }
19 ?>

```

Tới đây ta đã kết thúc thao tác thêm một thành viên mới vào cơ sở dữ liệu. Sau đây là toàn bộ code trong ứng dụng này.

```

01 <?php
02 session_start();
03 if(isset($_SESSION['userid']) && $_SESSION['level'] == 2)
04 {
05     if(isset($_POST['adduser']))
06     {
07         $u = $p = "";
08         if($_POST['username'] == NULL)

```

```

09  {
10    echo "Vui long nhap username<br />";
11  }
12  else
13  {
14    $u=$_POST['username'];
15  }
16  if($_POST['password'] != $_POST['re-password'])
17  {
18    echo "Password va re-password khong chinh xac<br />";
19  }
20  else
21  {
22    if($_POST['password'] == NULL)
23    {
24      echo "Vui long nhap password<br />";
25    }
26    else
27    {
28      $p=$_POST['password'];
29    }
30  }
31  $l=$_POST['level'];
32  if($u & $p & $l)
33  {
34    $conn=mysql_connect("localhost","root","root") or die("can't connect
    this database");
35    mysql_select_db("project",$conn);
36    $sql="select * from user where username='".$u."'";
37    $query=mysql_query($sql);
38    if(mysql_num_rows($query) != "" )
39    {
40      echo "Username nay da ton tai roi<br />";
41    }
42    else
43    {
44      $sql2="insert into user(username,password,level)
      values('".$u."','".$p."','".$l."')";
45      $query2=mysql_query($sql2);
46      echo "Da them thanh vien moi thanh cong";
47    }
48  }
49 }

```

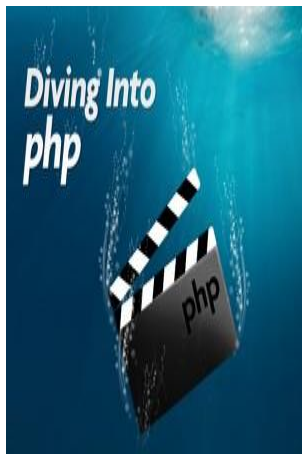


```

50 }
51 ?>
52 <form action='add_user.php' method='POST'>
53 Level: <select name='level'>
54 <option value='1'>Member</option>
55 <option value='2'>Admin </option>
56 </select><br />
57 Username: <input type='text' name='username' size='25' /><br />
58 Password: <input type='password' name='password' size='25' /> <br />
59 Re-Password: <input type='password' name='re-password' size='25' /><br />
60 <input type='submit' name='adduser' value='Add New User' />
61 </form>
62 <?php
63 }
64 else
65 {
66     header("location: login.php");
67     exit();
68 }
69 ?>

```

Bài 13: Viết ứng dụng quản lý thành viên bằng PHP và MYSQL



Ở các bài trước, chúng ta đã cùng tìm hiểu về việc **thêm một thành viên** và **thiết lập trang đăng nhập vào hệ thống administrator**. Ở bài này, chúng ta sẽ cùng nghiên cứu về ứng dụng quản trị các tài khoản. Hay nói cách khác là hệ thống lại toàn bộ danh sách của những thành viên trên website của chúng ta.

Sau khi đăng nhập vào trang admin người dùng sẽ được đẩy sang trang quản lý admin, tại đây chỉ những ai đã đăng nhập đúng với username và password trong cơ sở dữ liệu và có quyền hạn level là 2 thì mới có thể truy xuất vào trang này. Vì vậy chúng ta cần thao tác **kiểm tra session để quản lý phiên làm việc của người sử dụng**. Đặt đoạn code sau ở những trang bạn muốn kiểm tra quyền truy cập của người sử dụng.

```

01 <?php
02 session_start();

```

```

03 if(isset($_SESSION['userid']) && $_SESSION['level'] == 2)
04 {
05 // tại đây thực thi các hoạt động khi đăng nhập thành công.
06 }
07 else
08 {
09 header("location: login.php");
10 exit();
11 }
12 ?>

```

Như vậy, chúng ta sẽ đặt đoạn code này ở trang quản lý user. Cụ thể tôi đặt trang này có tên file là `manage_user.php`. Vì dữ liệu sẽ lặp lại toàn bộ user và ứng với từng user sẽ là 1 dòng dữ liệu được lặp lại. Chúng ta sẽ xây dựng 1 bảng gồm có STT là số thứ tự của từng user được đếm trên mỗi user khi lặp, username là tên truy cập của họ, level là cấp bậc của user (1 là member và 2 là admin), edit là cột chứa các link chỉnh sửa user, del là cột xóa các user.

```

1 <table align='center' width='400' border='1'>
2 <tr>
3 <td>STT</td>
4 <td>Username</td>
5 <td>Level</td>
6 <td>Edit</td>
7 <td>Del</td>
8 </tr>

```

Tiếp theo, chúng ta sẽ kết nối CSDL để tiến hành lập các user.

```

01 <?php
02 $conn=mysql_connect("localhost","root","root") or die("can't connect this
    database");
03 mysql_select_db("project",$conn);
04 $sql="select * from user order by id DESC";
05 $query=mysql_query($sql);
06 if(mysql_num_rows($query) == "")
07 {
08 echo "<tr><td colspan='5' align='center'>Chua co username nao</td></tr>";
09 }
10 ?>

```

Đoạn code ở trên tiến hành lựa chọn tất cả user có trong database. Đồng thời kiểm tra xem trong database có tồn tại user nào không. Nếu không sẽ xuất ra thông báo "chưa có username nào". Ngược lại khi có tồn tại user chúng ta sẽ xử lý tiếp như sau:

```

01 <?php
02 $stt=0;
03 while($row=mysql_fetch_array($query))
04 {
05 $stt++;
06 echo "<tr>";
07 echo "<td>$stt</td>";
08 echo "<td>$row['username']</td>";
09 if($row['level'] == "1")
10 {
11 echo "<td>Member</td>";
12 }
13 else
14 {
15 echo "<td>Admin</td>";
16 }
17 echo "<td><a href='edit_user.php?userid=$row[id]'>Edit</a></td>";
18 echo "<td><a href='del_user.php?userid=$row[id]'>Del</a></td>";
19 echo "</tr>";
20 ?>

```

Ở đây ta dùng biến \$stt để làm bộ đếm cho số lượng user. Như vậy, cứ mỗi lần lặp 1 user thì bộ đếm này sẽ tự động tăng lên 1 đơn vị. Sau đó, chúng ta đưa dữ liệu vào mảng và truyền các giá trị trên vào các cột dữ liệu trong bảng.

STT	Username	Level	Edit	Del
1	jacky	Admin	Edit	Del
2	test	Member	Edit	Del
3	abc	Member	Edit	Del
4	admin	Admin	Edit	Del

<http://www.qhonline.info>

Tới đây ta đã kết thúc thao tác quản lý các user đang tồn tại trong cơ sở dữ liệu. Sau đây là toàn bộ code trong ứng dụng này.

```

01 <?php
02 session_start();
03 if(isset($_SESSION['userid']) && $_SESSION['level'] == 2)

```

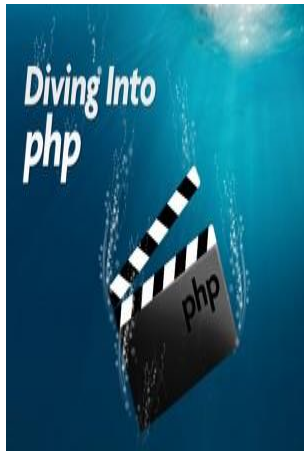
```

04 {
05 ?>
06 <table align='center' width='400' border='1'>
07 <tr>
08 <td>STT</td>
09 <td>Username</td>
10 <td>Level</td>
11 <td>Edit</td>
12 <td>Del</td>
13 </tr>
14 <?
15 $conn=mysql_connect("localhost","root","root") or die("can't connect this
    database");
16 mysql_select_db("project",$conn);
17 $sql="select * from user order by id DESC";
18 $query=mysql_query($sql);
19 if(mysql_num_rows($query) == "")
20 {
21 echo "<tr><td colspan=5 align=center>Chua co username nao</td></tr>";
22 }
23 else
24 {
25 $stt=0;
26 while($row=mysql_fetch_array($query))
27 {
28 $stt++;
29 echo "<tr>";
30 echo "<td>$stt</td>";
31 echo "<td>$row['username']</td>";
32 if($row['level'] == "1")
33 {
34 echo "<td>Member</td>";
35 }
36 else
37 {
38 echo "<td>Admin</td>";
39 }
40 echo "<td><a href='edit_user.php?userid=$row[id] ' >Edit</a></td>";
41 echo "<td><a href='del_user.php?userid=$row[id] ' >Del</a></td>";
42 echo "</tr>";
43 }
44
45 }

```

```
46 ?>
47 </table>
```

Bài 14: viết ứng dụng sửa xóa thành viên bằng PHP và MYSQL



Trong bài trước chúng ta đã nói về cách thêm và quản lý thành viên kết hợp PHP và MYSQL. Tiếp theo trong bài này, chúng ta sẽ viết tiếp 2 ứng dụng sửa và xóa thành viên để hoàn tất module user.

Để theo kịp bài này, các bạn vui lòng xem lại bài 11,12,13. Vì các bài được hướng dẫn liên mạch từ đầu tới cuối.

Cũng như những bài trước, hệ thống sửa và xóa thành viên này cũng chỉ có thể thực hiện được khi người sử dụng đăng nhập thành công với quyền hạn của 1 administrator. Do vậy, bạn đừng quên kiểm tra session ở đầu khi bắt đầu viết ứng dụng này nhé.

```
01 <?php
02 session_start();
03 if(isset($_SESSION['userid']) && $_SESSION['level'] == 2)
04 {
05     // tại đây thực thi các hoạt động khi đăng nhập thành công.
06 }
07 else
08 {
09     header("location: login.php");
10     exit();
11 }
12 ?>
```

A- Xây dựng trang sửa thành viên:

Vì là trang chỉnh sửa thành viên, nên nội dung của chúng có phần sẽ giống với phần thêm thành viên, chỉ khác là các ô nhập liệu giờ đây đã có dữ liệu. Dữ liệu này chúng ta tiến hành lấy từ cơ sở dữ liệu thông qua biến truyền mà ở trang quản lý đã gửi edit_user.php?userid=\$row[id]

Level:

Username:

Password:

Re-password:

<http://www.qhonline.info>

Như vậy để lấy được giá trị từ liên kết này chúng ta sẽ sử dụng biến `$_GET['userid']`. Sau khi đã có được giá trị này, việc còn lại của bạn là lấy thông tin của id này từ cơ sở dữ liệu và đưa vào form để người dùng có thể chỉnh sửa.

Vậy ta có.

```
01 <?php
02 $sql="select * from user where id='".$id."'";
03 $query=mysql_query($sql);
04 $row=mysql_fetch_array($query);
05 ?>
06 <form action="edit_user.php?userid=<?=$id?>" method=post>
07 Level: <select name=level>
08 <option value=1 <? if($row[level] == 1) echo "selected"; ?>
09 >Member</option>
10 <option value=2 <? if($row[level] == 2) echo "selected";
11 ?>>Administrator</option>
12 </select><br />
13 Username: <input type=text name=user size=20 value="<?=$row[username]?>"
14 /><br />
15 Password: <input type=password name=pass size=20 /> <br />
16 Re-password: <input type=password name=repass size=20 /><br />
17 <input type=submit name=ok value="Edit User" />
18 </form>
```

Theo như đoạn code ở trên phần value chúng ta muốn đưa giá trị vào thì buộc phải sử dụng cú pháp gọn `<?=$ten?>` hoặc cũng có thể sử dụng bằng cách `<? echo $ten; ?>`. Ở đây chúng ta chỉ đưa ra giá trị trên text box username, còn lại mật khẩu do có thể bị mã hóa nên không nên hiển thị ra. Phần level do lưu trong cơ sở dữ liệu là những con số 1,2 nên khi đưa ra chúng ta phải so sánh, nếu trong bảng lưu là 1 thì ở chỗ có giá trị bằng 1 sẽ thêm chữ selected. Để mặc định lựa chọn khi người dùng nhấn vào nút chỉnh sửa.

Tiếp tới người dùng sẽ nhấn nút edit. Việc còn lại của chúng ta là kiểm tra dữ liệu khi họ tiến hành chỉnh sửa dữ liệu. Điểm khác biệt ở đây là chúng ta phải cân nhắc khi viết trang chỉnh sửa. Bởi không phải ai vào chỉnh sửa cũng đều sửa cả thông tin username, password, level. Có khi họ chỉ sửa mỗi level hoặc có khi họ chỉ sửa mỗi password.

Vì thế ta phải xét các trường hợp để giải quyết.

Trường hợp 1: Người dùng chỉ chỉnh sửa username, level mà không chỉnh sửa password. Khi đó username, level bắt buộc đều đã có dữ liệu nên ta không cần kiểm tra chúng rỗng hay không. Nhưng với password chúng ta không thể bắt người lập trình nhập liệu. Và họ hoàn toàn có thể để rỗng.

Trường hợp 2: Người dùng chỉnh sửa username, level ,password. Khi đó ta xem password có dữ liệu và ta tiến hành cập nhật.

Để giải quyết bài toán trên thỏa 2 trường hợp ta sẽ xét phương pháp sau:

Ta kiểm tra password và repassword có giống nhau hay không ?. Nếu chúng khác nhau nghĩa là người dùng nhập liệu nhưng nhập sai. Vậy phải cảnh báo lỗi cho họ biết là họ đã nhập sai. Còn ngược lại nếu password và repassword giống nhau thì sẽ nảy sinh 2 trường hợp con.

+ Một là password và repassword đều có dữ liệu

+ Hai là password và repassword không có dữ liệu.

Khi đó ở trường hợp con 1 chúng ta sẽ khởi tạo biến \$p và ghi nhận thông tin họ nhập liệu. Và ở trường hợp 2 chúng ta không ghi nhận thông tin người nhập liệu vì họ không có ý định chỉnh sửa mật khẩu. Có thể viết đoạn code đơn giản như sau:

```

01 <?php
02 if($_POST['pass'] != $_POST['repass'])
03 {
04     echo "Password and re-password is not correct";
05 }
06 else
07 {
08     if($_POST['pass'] != NULL)
09     {
10         $p=$_POST['pass'];
11     }
12 }
13 ?>

```

Tiếp tới chúng ta sẽ phân loại dữ liệu cập nhật. Khi có username, password, level thì ta cập nhật tất cả thông tin của họ bằng cú pháp SQL. (xem lại [bài 9: ngôn ngữ SQL và MYSQL cơ bản](#)). Ngược lại khi không tồn tại password thì ta chỉ cập nhật username và level.

Một điều nữa chúng ta cần quan tâm đó là đối với lệnh update và delete, ta phải truyền 1 id cụ thể để tránh việc chúng xóa hoặc sửa dữ liệu toàn bộ trong bảng user. Với id tôi truyền nhận từ bên ngoài ta sẽ có.

```

01 <?php
02 if($u && $p && $l )
03 {
04     $sql="update user set username='".$u."', password='".$p."', level='".$l."'
    where id='".$id."'";
05     mysql_query($sql);
06     header("location:mana_user.php");
07     exit();
08 }
09 else
10 {
11     if($u && $l)
12     {
13         $sql="update user set username='".$u."', level='".$l."' where
        id='".$id."'";
14         mysql_query($sql);
15         header("location:mana_user.php");
16         exit();
17     }
18 }

```


19 ?>

Sau khi sửa thành công một thành viên chúng ta sẽ đưa người dùng trở về với trang quản lý (xem bài 13 để biết cách **xây dựng trang quản lý thành viên** như thế nào)

Nội dung của toàn bộ code ở trên như sau:

```
01 <?php
02 $conn=mysql_connect("localhost","root","root") or die("can't connect this
    database");
03 mysql_select_db("project",$conn);
04 $id=$_GET['userid'];
05 if(isset($_POST['ok']))
06 {
07     if($_POST['user'] == NULL)
08     {
09         echo "Please enter your username";
10     }
11     else
12     {
13         $u=$_POST['user'];
14     }
15     if($_POST['pass'] != $_POST['repass'])
16     {
17         echo "Password and re-password is not correct";
18     }
19     else
20     {
21         if($_POST['pass'] != NULL)
22         {
23             $p=$_POST['pass'];
24         }
25     }
26     $l = $_POST['level'];
27     if($u && $p && $l )
28     {
29         $sql="update user set username='".$u."', password='".$p."',
    level='".$l."' where id='".$id."'";
30         mysql_query($sql);
31         header("location:mana_user.php");
32         exit();
33     }
```

```

34 else
35 {
36   if($u && $l)
37   {
38     $sql="update user set username='".$u."', level='".$l."' where
id='".$id."'";
39     mysql_query($sql);
40     header("location:mana_user.php");
41     exit();
42   }
43 }
44 }
45 $sql="select * from user where id='".$id."'";
46 $query=mysql_query($sql);
47 $row=mysql_fetch_array($query);
48 ?>
49 <form action="edit_user.php?userid=<?=$id?>" method=post>
50 Level: <select name=level>
51 <option value=1 <? if($row[level] == 1) echo "selected"; ?>
>Member</option>
52 <option value=2 <? if($row[level] == 2) echo "selected";
?>>Administrator</option>
53 </select><br />
54 Username: <input type=text name=user size=20 value="<?=$row[username]?>"
/><br />
55 Password: <input type=password name=pass size=20 /> <br />
56 Re-password: <input type=password name=repass size=20 /><br />
57 <input type=submit name=ok value="Edit User" />
58 </form>

```

B- Xây dựng trang xóa thành viên:

Đối với trang xóa dữ liệu, chúng ta cũng không cần phải xử lý quá nhiều. Bởi nhiệm vụ của chúng chỉ đơn giản là xóa đi những dòng trong bảng.

Như vậy cũng như trang edit chúng ta nhận giá trị từ nội dung liên kết ở trang quản lý đã gửi là `del_user.php?userid=$row[id]`.

Và thực thi lệnh kết nối csdl để xử lý nội dung này.

```

1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can't connect this
  database");
3 mysql_select_db("project",$conn);
4 $sql="delete from user where id='".$id."'";
5 mysql_query($sql);
6 header("location:mana_user.php");
7 exit();
8 ?>

```

Sau khi xóa xong thành viên thì hệ thống sẽ đưa chúng ta trở về trang quản lý user lại.

C- Tổng kết:

Như vậy chúng ta đã kết thúc các thao tác thêm, sửa, xóa, quản lý thành viên. Qua những chuỗi bài học này, chắc hẳn các bạn đã có những khái niệm hình thành từ việc ứng dụng lập trình PHP để xây dựng các hệ thống cơ bản rồi. Việc lập trình 1 ứng dụng đòi hỏi bạn ngoài việc vận dụng nội dung đã học ra, bạn còn phải biết linh động xử lý các tình huống một cách hiệu quả. Tránh việc rập khuôn một cách máy móc và học thuộc các nội dung code. Vì ở những mô hình khác, bài toán khác bạn sẽ có những cách thức xử lý và giải quyết khác nhau.

Bài 15: Viết ứng dụng tạo mã xác nhận bằng PHP



Ở những bài trước, chúng ta đã tìm hiểu về cách làm việc với PHP kết hợp MYSQL trong việc xây dựng hệ thống thêm, sửa, xóa và quản lý user. Trong bài này, chúng ta tiếp tục tìm hiểu phương pháp tạo dãy số ngẫu nhiên để chống tấn công flood dữ liệu. Hoặc ai đó cố tình spam khiến cơ sở dữ liệu của chúng ta không thể xử lý tiếp được.

Trước tiên, chúng ta tiến hành tạo form HTML đơn giản để thực thi thao tác nhập liệu như sau:

```

01 <form action="form.php" method=post>
02 <table>
03 <tr>

```

```

04  <td align="left">
05    <label for="captcha">Captcha</label>
06  </td>
07  <td>
08    <input type="text" name="txtCaptcha" maxlength="10" size="32" />
09  </td>
10  <td>
11    
12  </td>
13 </tr>
14 <tr>
15  <td> </td>
16  <td>
17    <input type="submit" name="ok" value="Check" />
18  </td>
19 </tr>
20 </table>
21 </form>

```

Chúng ta chú ý phần hình ảnh, tại đây tôi truyền đường dẫn chính là liên kết tới trang `random_image.php`. Trang này sẽ thực thi công việc tạo ra những bức hình có dãy số ngẫu nhiên để phân nhập liệu của người sử dụng tham chiếu tới.

Tiếp theo, chúng ta khởi tạo file `random_image.php` để lấy ra dãy số ngẫu nhiên và phát sinh chúng ngay trên tấm hình cho người truy cập nhập liệu.

Để làm được điều đó. Chúng ta sẽ khởi tạo session và lưu dãy số nhập liệu đó vào session của mình, sau đó so khớp với trang form bên kia.(xem lại [bài 7: Khái niệm cơ bản về cookie và session trong PHP](#))

Captcha 

<http://www.qhonline.info>

Đầu tiên chúng ta sử dụng hàm `md5` và ran để mã hóa các ký tự bao gồm số và chữ cái. Khi sử dụng hàm `md5()` ký tự phát sinh sẽ lên tới 32 ký tự. Và chúng ta chỉ lấy duy nhất 5 ký tự từ chuỗi mã hóa đó bằng hàm `substr`. Tiếp tục ta lưu đoạn mã hóa này trong session cụ thể

`$_SESSION['security_code']`], để tại trang form ta sẽ sử dụng so sánh với phần nhập liệu của người sử dụng.

Vậy nên đoạn code dưới sẽ giải quyết các tình huống này.

```
1 <?php
2 $md5_hash = md5(rand(0,999));
3 $security_code = substr($md5_hash, 15, 5);
4 $_SESSION["security_code"] = $security_code;
5 ?>
```

Kế đến ta khởi tạo tấm hình với chiều rộng và chiều cao mà ta thiết lập thông qua hàm `ImageCreate()` . Và tiếp tục khai báo 2 màu chính là trắng và đen bằng hàm `ImageColorAllocate($image, red, green, blue)` . Hàm này sẽ tạo ra một màu sắc từ hệ màu RGB trên tấm hình mà chúng ta vừa khởi tạo.

Tiếp tục ta đổ background của tấm hình sẽ là màu đen và chữ xuất hiện trên tấm hình sẽ là màu trắng bằng hàm `ImageFill()` và hàm `ImageString()` . Hàm `ImageString()` có một số đối số cơ bản như sau: `ImageString($image, 5, 30, 6, $security_code, $white)`; Trong đó:

+ `$image` là hình mà chúng ta khởi tạo.

+ `5`: là font-size mà chúng ta quy ước cho ký tự xuất hiện trên hình.

+ `30`: là khoảng cách bên trái của tấm hình.

+ `6` : là khoảng cách từ trên của tấm hình.

+ `$security_code`: là đoạn code sau khi chúng ta cắt ra ở trên.

+ \$white: là màu sắc mà chúng ta đã sử dụng hàm ImageColorAllocate() ở trên để khởi tạo ra màu trắng.

```
1 <?php
2 $width = 100;
3 $height = 30;
4 $image = ImageCreate($width, $height);
5 $white = ImageColorAllocate($image, 255, 255, 255);
6 $black = ImageColorAllocate($image, 0, 0, 0);
7 ImageFill($image, 0, 0, $black);
8 ImageString($image, 5, 30, 6, $security_code, $white);
9 ?>
```

Kế tới chúng ta sẽ sử dụng lệnh header để trả nội dung này trở về dữ liệu dạng hình. Và tiến hành khởi tạo định dạng cho file hình mà chúng ta vừa tạo là JPG đồng thời giải phóng hình tạm được sử dụng để khởi tạo ra tấm hình này bằng đoạn code sau:

```
1 <?php
2 header("Content-Type: image/jpeg");
3 ImageJpeg($image);
4 ImageDestroy($image);
5 ?>
```

Để dễ quản lý chúng ta sẽ đặt tất cả đoạn code trên vào trong 1 hàm để dễ sử dụng và quản lý chúng tốt hơn. Vậy toàn bộ đoạn code trong trang random_image.php này như sau.

```
01 <?php
02 session_start();
03 function create_image()
04 {
05     $md5_hash = md5(rand(0,999));
06     $security_code = substr($md5_hash, 15, 5);
07     $_SESSION["security_code"] = $security_code;
08     $width = 100;
09     $height = 30;
10     $image = ImageCreate($width, $height);
11     $white = ImageColorAllocate($image, 255, 255, 255);
12     $black = ImageColorAllocate($image, 0, 0, 0);
13     ImageFill($image, 0, 0, $black);
14     ImageString($image, 5, 30, 6, $security_code, $white);
15     header("Content-Type: image/jpeg");
16     ImageJpeg($image);
17     ImageDestroy($image);
```

```
18 }
19 create_image() ;
20 exit() ;
21 ?>
```

Tiếp tục, trong trang form chúng ta phải khởi tạo session để chúng nhận được các giá trị mà ta đã khởi tạo và sử dụng ở trang random_image, kế tới ta kiểm tra xem người dùng có nhấn submit chưa. Nếu có ta sẽ tiếp tục kiểm tra xem người dùng nhập vào textbox có đúng là dãy số hiển thị trên hình ảnh hay không. Và xuất ra thông báo tương ứng với nội dung của người nhập liệu. (xem lại **[bài 4: Xử lý giá trị form trong PHP](#)**)

Sau đây là toàn bộ code của trang form.php.

```
01 <?php
02 session_start();
03 if(isset($_POST['ok']))
04 {
05     if($_POST['txtCaptcha'] == NULL)
06     {
07         echo "Please enter your code";
08     }
09     else
10     {
11         if($_POST['txtCaptcha'] == $_SESSION['security_code'])
12         {
13             echo "ma lenh hop le";
14         }
15         else
16         {
17             echo "Ma lenh khong hop le";
18         }
19     }
20 }
21 ?>
22 <form action="form.php" method=post>
23 <table>
24 <tr>
25 <td align="left">
26 <label for="captcha">Captcha</label>
27 </td>
28 <td>
```

```

29     <input type="text" name="txtCaptcha" maxlength="10" size="32" />
30 </td>
31 <td>
32     
33 </td>
34 </tr>
35 <tr>
36 <td> </td>
37 <td>
38     <input type="submit" name="ok" value="Check" />
39 </td>
40 </tr>
41 </table>
42 </form>

```

Tổng kết:

Qua bài học này, chúng ta đã hiểu được nguyên tắc xử lý và những hàm cơ bản mà 1 hệ thống khi sử dụng captcha cần phải vận dụng như thế nào. Và tất nhiên hàm xử lý captcha này cũng chỉ mang kiến trúc tương đối. Để tăng cường tính bảo mật hơn, các bạn nên sử dụng 1 số hàm chuyên dụng nhằm mã hóa hoặc gây nội dung khó nhìn hơn để chặn các tools hoặc các hệ thống bot có thể qua mặt thông số này.

Bài 16: Viết ứng dụng đếm số người online bằng php



Trong bài này, chúng ta sẽ nói về cách thức xây dựng tính năng thống kê số người đang viếng thăm website của bạn. Vì là chức năng nhỏ, nên tôi không đi nhiều về những tính năng lớn như bao nhiêu khách, bao nhiêu thành viên,...Chỉ đơn thuần là quá trình hiển thị hiện tại có bao nhiêu người đang thăm viếng website của bạn.

Trước hết, chúng ta khởi tạo cơ sở dữ liệu như sau. (xem lại bài [bài 9: ngôn ngữ SQL và MYSQL cơ bản](#))

```

1 mysql> create table useronline (tgtmp INT(15) DEFAULT "0" NOT NULL ,
2 ip VARCHAR(50) NOT NULL ,

```



```
3 local VARCHAR(100) NOT NULL,  
4 PRIMARY KEY(tgtmp) ,  
5 KEY ip(ip) ,  
6 KEY local(local)) ;
```

tgtmp là thời gian mà họ truy cập được tính khi họ truy vào trang web đó.

IP là dãy số lưu thông tin IP của họ khi viếng thăm website của chúng ta.

local là nơi lưu đường dẫn mà họ đang truy cập.

Khi người dùng truy cập vào trang web, chúng ta sẽ tiến hành insert thông tin của họ vào cơ sở dữ liệu với các tham số cơ bản nhưng trong database.

tgtmp được tính ra bằng hàm time(). Hàm này sẽ lấy ra thời gian hiện tại của người truy cập. Tiếp tục ta lại tính thời gian mới của họ, được tính bằng việc quy ước thời gian quy định. Cụ thể ở đây tôi cho là 900 giây tương đương với 15 phút truy cập của họ.

```
1 <?php  
2 $tg=time() ;  
3 $tgout=900;  
4 $tgnew=$tg - $tgout;  
5 ?>
```

như vậy nếu thời gian lưu trong database mà nhỏ hơn thời gian new này thì chúng ta có thể hiểu rằng vị khách ấy đã rời khỏi website của chúng ta. Cụ thể hơn.

Ví dụ: tôi viếng thăm website đó là 7h.

như vậy hệ thống sẽ ghi thông tin lúc đó là 7h.

Nếu sau 1 thời gian tôi không làm gì, hoặc tôi không truy cập website đó nữa thì hệ thống sẽ không ghi nhận thông tin mới. Như thế nếu bây giờ 8h và trừ đi 15 phút tôi quy ước, rõ ràng là thời gian mới hiện tại là 7h45, Thời gian này vẫn lớn hơn thời điểm lưu thông tin 7h ($7h45 > 7h$). Do vậy, nếu chúng thỏa điều kiện đó thì chúng ta chỉ việc xóa đi các record trong cơ sở dữ liệu là xong.

Vậy ta có code kết nối CSDL như sau:(xem lại [bài 10: kết hợp PHP và MYSQL trong ứng dụng](#))

```
1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can't connect");
3 mysql_select_db("online",$conn);
4 ?>
```

Sau đó ta tiến hành ghi nhận thông tin người dùng vào CSDL.

```
1 <?php
2 $sql="insert into useronline(tgtmp,ip,local)
3 values('$tg','$REMOTE_ADDR','$PHP_SELF')";
4 $query=mysql_query($sql);
5 ?>
```

\$REMOTE_ADDR là biến môi trường dùng để lấy ra IP của người truy cập.

\$PHP_SELF là biến môi trường dùng để lấy ra đường dẫn mà người dùng đang truy cập.

Tiếp đến ta tiến hành xóa record khi thời gian thực lớn hơn thời gian trong cơ sở dữ liệu.

```
1 <?php
2 $sql="delete from useronline where tgtmp < $tgnew";
3 $query=mysql_query($sql);
4 ?>
```

Tiếp tục là công việc hiển thị thông tin ra bên ngoài bằng cách liệt kê các record có trong database.

```
1 <?php
2 $sql="SELECT DISTINCT ip FROM useronline WHERE local='$PHP_SELF'";
3 $query=mysql_query($sql);
```

```

4 $user = mysql_num_rows($query);
5 ?>

```

DISTINCT là cú pháp cho phép liệt kê các dòng record mà không cho phép chúng có dữ liệu trùng lặp như cú pháp select bình thường.

Vậy chúng ta sẽ liệt kê tất cả những ip của những ai đang truy cập trên trang useronline.php.

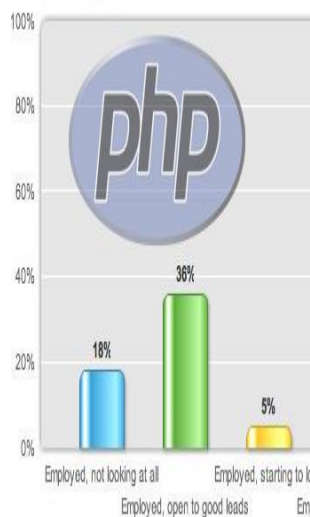
Phần việc cuối cùng còn lại là chúng ta sẽ xuất thông tin ấy ra trình duyệt.

Và sau đây là toàn bộ nội dung code của trang useronline.php

```

01 <?php
02 $tg=time();
03 $tgout=900;
04 $tgnew=$tg - $tgout;
05 $conn=mysql_connect("localhost","root","root") or die("can't connect");
06 mysql_select_db("online",$conn);
07 $sql="insert into useronline(tgtmp,ip,local)
values('$tg','$REMOTE_ADDR','$PHP_SELF')";
08 $query=mysql_query($sql);
09 $sql="delete from useronline where tgtmp < $tgnew";
10 $query=mysql_query($sql);
11 $sql="SELECT DISTINCT ip FROM useronline WHERE local='$PHP_SELF'";
12 $query=mysql_query($sql);
13 $user = mysql_num_rows($query);
14 echo "user online :$user";
15 ?>

```



Bài 17: Xây dựng hệ thống bình chọn bằng PHP và MYSQL

Ở loạt bài trước, chúng ta đã bàn tới các ứng dụng quản lý người sử dụng, xây dựng bộ đếm, tạo mã bảo mật để phòng chống những người có dụng ý xấu trên website. Tiếp theo, trong bài này chúng ta sẽ bàn về cách thức xây dựng hệ thống cho phép người dùng bình chọn khi họ tiến hành thăm

viếng website của chúng ta. Để xây dựng hệ thống bình chọn chúng ta cần phân tích xem hệ thống này cần có bao nhiêu bảng quan hệ.

A- Phân tích và thiết kế cơ sở dữ liệu:

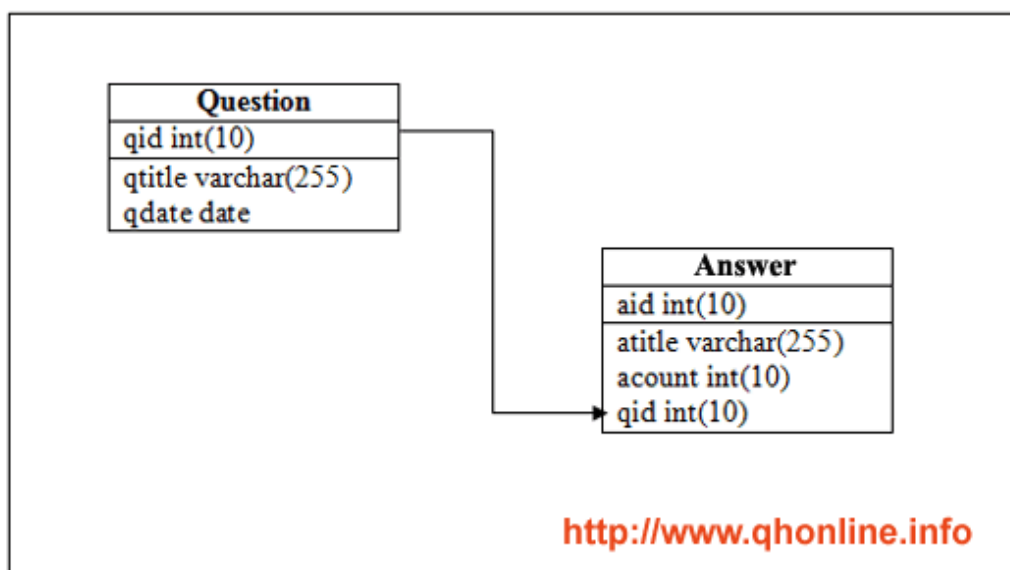
Trong tình huống này chúng ta cần 1 bảng câu hỏi và 1 bảng câu trả lời.

Phân tích quan hệ giữa chúng ta có:

1 Câu hỏi có nhiều câu trả lời.

1 câu trả lời chỉ dành cho 1 câu hỏi.

Vậy giữa 2 bảng này sẽ phát sinh một khóa ngoại, là khóa dùng để liên kết giữa hai bảng này.



Vậy ta có cú pháp tạo 2 bảng như sau.

Bảng câu hỏi (question)

```
1 mysql> create table question(qid INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
2 -> qtitle VARCHAR(255) NOT NULL,  
3 -> qdate DATE NOT NULL DEFAULT '0000-00-00',  
4 -> PRIMARY KEY(qid));
```

Bảng câu trả lời (answer)

```
1 mysql> create table answer(aid INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,  
2 -> qid INT(10) UNSIGNED NOT NULL,  
3 -> atitle VARCHAR(255) NOT NULL,  
4 -> account INT(10) NOT NULL DEFAULT '0',  
5 -> PRIMARY KEY(aid));
```

Nhập liệu bằng CSDL ta có:

```
1 mysql> Insert into question(qtitle,qdate) values("Ban thay qhonline the nao  
?","  
2 2009-10-10");  
3 mysql> Insert into answer(qid,atitle,account) values("1","Nhin rat dep",0);  
4 mysql> Insert into answer(qid,atitle,account) values("1","Nhin Dep",0);  
5 mysql> Insert into answer(qid,atitle,account) values("1","Nhin Cung duoc",0);  
6 mysql> Insert into answer(qid,atitle,account) values("1","Nhin qua xau",0);
```

B- Xây dựng trang bình chọn poll.php

Kết nối cơ sở dữ liệu:

```
1 <?php  
2 $conn=mysql_connect("localhost","root","root") or die("can not connect  
database");  
3 mysql_select_db("poll_exam",$conn);  
4 ?>
```

Lựa chọn câu hỏi có trong cơ sở dữ liệu để liệt kê ra trên website, sau đó ta lại tiếp tục liệt kê các câu trả lời của câu hỏi đó bằng cách sử dụng dấu chọn lựa (radio). Như vậy chúng ta sẽ phải chạy cùng lúc 2 câu truy vấn lồng nhau trong suốt quá trình truy xuất.

```
01 <?php  
02 $sql="select * from question order by qid desc";  
03 $query=mysql_query($sql);  
04 if(mysql_num_rows($query) > 0)
```

```

05 {
06 $row=mysql_fetch_array($query);
07 $qid=$row[qid];
08 echo "<form action='poll.php?questionid=$qid' method='post' >";
09 echo "<h2>$row[qttitle]</h2>";
10 $sql2="select * from answer where qid='$qid' order by aid";
11 $query2=mysql_query($sql2);
12 if(mysql_num_rows($query2) > 0)
13 {
14 while($row2=mysql_fetch_array($query2)) {
15 echo "<input type='radio' name='answer' value=$row2[aid]>$row2[atitle]<br
16 />";
17 }
18 echo "<input type='submit' name='ok' value='Bình Chọn'>";
19 echo "<a href='result.php?questionid=$qid'>Xem Ket Qua</a>";
20 echo "</form>";
21 }
22 ?>

```

Và màn hình sẽ xuất ra như hình bên dưới

Ban thay qhonline the nao ?

- ☐ Nhìn rat dep
 - ☐ Nhìn Dep
 - ☐ Nhìn Cung duoc
 - ☐ Nhìn qua xau
- [Xem Ket Qua](#)

<http://www.qhonline.info>

Và khi người dùng nhất nút Bình chọn, chúng ta sẽ gọi lại chính trang đó để xử lý dữ liệu mà người dùng vừa lựa chọn.

Vậy ta phải sử dụng lệnh isset() để kiểm tra xem người dùng có nhấn nút bình chọn không, tiếp đến ta lấy ra id mà người dùng vừa tiến hành chọn ở form bên dưới.

Cuối cùng ta cập nhật dữ liệu bằng cách lấy số trong cơ sở dữ liệu cộng tiếp cho 1 đơn vị nữa. Và đưa người dùng sang trang kết quả.

Vì trong đoạn code này có lệnh header nên chúng ta phải đặt quá trình xử lý này ở trên form. Nếu không sẽ bị lỗi dữ liệu khi gọi.

```
01 <?php
02 if(isset($_POST['ok']))
03 {
04     $id=$_POST['answer'];
05     $qid=$_GET['questionid'];
06     $sql3="update answer set account=account + 1 where aid='".$id."'";
07     mysql_query($sql3);
08     header("location: result.php?questionid=$qid");
09     exit();
10 }
11 ?>
```

Vậy toàn bộ code của trang poll.php này như sau.

```
01 <?php
02 $conn=mysql_connect("localhost","root","root") or die("can not connect
03 database");
04 mysql_select_db("poll_exam",$conn);
05 if(isset($_POST['ok']))
06 {
07     $id=$_POST['answer'];
08     $qid=$_GET['questionid'];
09     $sql3="update answer set account=account + 1 where aid='".$id."'";
10     mysql_query($sql3);
11     header("location: result.php?questionid=$qid");
12     exit();
13 }
14 $sql="select * from question order by qid desc";
15 $query=mysql_query($sql);
16 if(mysql_num_rows($query) > 0)
17 {
18     $row=mysql_fetch_array($query);
19     $qid=$row[qid];
20     echo "<form action='poll.php?questionid=$qid' method='post' >";
```

```

20 echo "<h2>$row[qttitle]</h2>";
21 $sql2="select * from answer where qid='$qid' order by aid";
22 $query2=mysql_query($sql2);
23 if(mysql_num_rows($query2) > 0)
24 {
25     while($row2=mysql_fetch_array($query2)){
26         echo "<input type=radio name=answer value=$row2[aid]>$row2[atitle]<br
27     />";
28     }
29     echo "<input type='submit' name='ok' value='Bình Chọn'>";
30     echo "<a href='result.php?questionid=$qid'>Xem Ket Qua</a>";
31     echo "</form>";
32 }
33 ?>

```

C- Xây dựng trang kết quả bình chọn result.php

Ở trang này, ta sẽ lấy giá trị question id của người bình chọn từ trang poll.php và tính toán xem tỷ lệ phần trăm của các câu hỏi ấy như thế nào.

Đầu tiên, ta lấy giá trị từ poll.php?questionid=1. vậy 1 là giá trị ta sẽ có được từ \$_GET['questionid']. Tiếp đến ta lại dùng giá trị vừa lấy được này để liệt kê thông tin câu hỏi và tính toán giá trị.

Để tính toán giá trị, ta cần biết tổng số bình chọn trong toàn bộ các câu trả lời hiện nay là bao nhiêu. Bằng câu lệnh SUM ta có thể làm được điều đó.

```

1 $sql2="select qid, SUM(acount) as total from answer group by qid having
   qid='". $qid. "'";

```

Câu lệnh này cho phép ta lấy ra tổng số bình chọn trong cơ sở dữ liệu của câu hỏi mà ta truyền vào. Để SUM được account ta cần phải GROUP(gom nhóm) chúng ta lại theo mã câu hỏi mà chúng ta cần tìm. having là mệnh đề theo sau có ý nghĩa tương đương với where, nó thường được sử dụng theo GROUP.

Như vậy để lấy ra tổng số phiếu bình chọn ta chỉ việc thực thi câu truy vấn này, và lấy tên cột tạm là total.

```
1 <?php
2 $sql2="select qid, SUM(acount) as total from answer group by qid having
3 qid='".$qid."'";
4 $query2=mysql_query($sql2);
5 $row2=mysql_fetch_array($query2);
6 $total=$row2[total];
7 ?>
```

Kế tới, ta lại tính toán số phiếu của từng câu hỏi trong cơ sở dữ liệu, dựa vào tổng số câu hỏi này.

Ví dụ: Tổng số lượt bình chọn là 10

câu 1 là 3

câu 2 là 4

câu 3 là 2

câu 4 là 1

Vậy suy ra phần trăm trên tổng số 10 của 4 câu này được tính bằng cách lấy số liệu của từng câu chia cho tổng số câu hỏi và nhân cho 100. Ví dụ câu 1 là: $(3/10)*100 = 30\%$

Để làm tròn kết quả ta sử dụng hàm round(). Giúp dữ liệu đưa về sự đồng bộ và rõ ràng nhất.

Vậy code xử lý của chúng ta như sau:

```
01 <?php
02 $sql3="select * from answer where qid='".$qid."' order by aid";
03 $query3=mysql_query($sql3);
04 if(mysql_num_rows($query3) > 0)
05 {
06   while($row3=mysql_fetch_array($query3)) {
07     $percent=round(($row3[acount]/$total)*100,2);
08     echo "<h4 style='color:red; font:12px verdana; '>$row3[atitle] :
    $row3[acount] ($percent %)</h4>";
```

```
09 }  
10 }  
11 ?>
```

Kết quả khi xuất ra sẽ có hình như sau:

Ban thay qhonline the nao ?

Nhin rat dep : 1 (14.29 %)

Nhin Dep : 4 (57.14 %)

Nhin Cung duoc : 1 (14.29 %)

Nhin qua xau : 1 (14.29 %)

<http://www.qhonline.info>

Vậy toàn bộ mã nguồn của trang result.php này như sau:

```
01 <?php  
02 $conn=mysql_connect("localhost","root","root") or die("can not connect  
03 database");  
04 mysql_select_db("poll_exam",$conn);  
05 if(isset($_GET['questionid']))  
06 {  
07     $qid=$_GET['questionid'];  
08     $sql="select * from question where qid='".$qid."'";  
09     $query=mysql_query($sql);  
10     $row=mysql_fetch_array($query);  
11     echo "<h2>$row[qttitle]</h2>";  
12     $sql2="select qid, SUM(acount) as total from answer group by qid having  
13 qid='".$qid."'";  
14     $query2=mysql_query($sql2);  
15     $row2=mysql_fetch_array($query2);  
16     $total=$row2[total];  
17     $sql3="select * from answer where qid='".$qid.'" order by aid";  
18     $query3=mysql_query($sql3);  
19     if(mysql_num_rows($query3) > 0){  
20         while($row3=mysql_fetch_array($query3)){  
21             $percent=round(($row3[acount]/$total)*100,2);
```

```

20     echo "<h4 style='color:red; font:12px verdana; '>$row3[atitle] :
    $row3[account] ($percent %)</h4>";
21 }
22 }
23 }
24 ?>

```

Bài 18: Xây dựng ứng dụng tạo trang upload nhiều hình ảnh



Ở những bài trước, chúng ta đã có dịp nghiên cứu từng ứng dụng nhỏ bởi việc kết hợp PHP và Mysql. Trong bài này, chúng ta sẽ tiếp tục tìm hiểu phương pháp xây dựng một trang upload nhiều hình ảnh như thế nào. Và lưu chúng vào cơ sở dữ liệu ra sao trong lập trình php.

Mô hình này là rất cần thiết đối với các hệ thống. Bởi qua đó người sử dụng có thể thao tác và upload những hình ảnh đi kèm với hệ thống của họ như tin tức, sản phẩm,...

Đầu tiên, ta tiến hành xây dựng cơ sở dữ liệu với bảng images như sau (xem lại [bài 9: cơ bản về ngôn ngữ sql và mysql](#)).

```

1 CREATE TABLE `images` (
2   `img_id` INT( 10 ) UNSIGNED NOT NULL AUTO_INCREMENT
  PRIMARY KEY ,
3   `img_url` VARCHAR( 255 ) NOT NULL ,
4   `img_name` VARCHAR( 150 ) NOT NULL
5 );

```

Như vậy, tôi có bảng images với 3 cột thuộc tính là img_id dùng để lưu khóa của từng hình, img_url dùng để lưu đường dẫn vào hệ thống, img_name dùng để lưu tên hình ảnh.

Ứng dụng của chúng ta cũng rất cần sự tùy biến và dễ sử dụng. Sao cho người dùng có thể hoàn toàn thao tác. Ví dụ: nếu có 2 hình, họ chỉ muốn tạo ra 2 chế độ chọn file thì sẽ chỉ khai báo là 2. Và ngược lại nếu là n hình, họ sẽ tạo được n file lựa chọn. Để làm được điều này, trước tiên ta cần tạo 1 form nhỏ cho phép người dùng nhập vào số lượng file mà họ muốn thao tác. Vậy tôi khởi tạo file có tên upload.php với mã lệnh form như sau:

```

1 <form action="upload.php" method="post">
2 Enter your Image: <input type="text" name="txtnum" value="<?php echo
  $_POST['txtnum']; ?>" size="10" />
3 <input type="submit" name="ok_num" value="Accept" />
4 </form>

```

Sau khi người dùng nhấn nút accept thì cũng là lúc chúng ta tạo ra số field upload theo yêu cầu của họ. Vậy, ta có code xử lý bên dưới như sau: (xem lại [bài 4: xử lý giá trị trên form](#)).

```

01 <?php

```

```

02 if(isset($_POST['ok_num']))
03 {
04     $num=$_POST['txtnum'];
05     echo "<hr />";
06     echo "Ban dang chon $num file upload<br />";
07     echo "<form action='doupload.php?file=$num' method='post'
    enctype='multipart/form-data'>";
08     for($i=1; $i <= $num; $i++)
09     {
10         echo "<input type='file' name='img[]' /><br />";
11     }
12     echo "<input type='submit' name='ok_upload' value='Upload' />";
13     echo "</form>";
14 }
15 ?>

```

Đoạn code này sử dụng vòng lặp for với điều kiện là biến \$i sẽ nhỏ hơn số file mà họ nhập vào. Đồng thời, để việc upload thành công tại form ta cũng cần thêm đoạn code :

`enctype='multipart/form-data'.`

Số lượng field sẽ lặp theo yêu cầu của người dùng. Do vậy. tại name của thẻ input, ta sẽ sử dụng mảng để lưu thông tin. Sử dụng name='img[]', có nghĩa là mỗi lần lặp và mỗi lần người dùng chọn file thì mảng img sẽ thêm 1 khóa tương ứng. (xem lại [bài 6: tổng quan về mảng và các hàm hỗ trợ](#))

Và cuối cùng là nút submit thực thi công việc upload và gọi tới trang doupload.php?file=\$num (chỉ rõ số file cụ thể do người dùng nhập tại trang doupload).

Vậy toàn bộ code của trang upload này sẽ như sau:

```

01 <form action="upload.php" method="post">
02 Enter your Image: <input type="text" name="txtnum" value="<?php echo

```

```

    $_POST['txtnum']; ?>" size="10" />
03 <input type="submit" name="ok_num" value="Accept" />
04 </form>
05 <?php
06 if(isset($_POST['ok_num']))
07 {
08     $num=$_POST['txtnum'];
09     echo "<hr />";
10     echo "Ban dang chon $num file upload<br />";
11     echo "<form action='doupload.php?file=$num' method='post'
enctype='multipart/form-data'>";
12     for($i=1; $i <= $num; $i++)
13     {
14         echo "<input type='file' name='img[]' /><br />";
15     }
16     echo "<input type='submit' name='ok_upload' value='Upload' />";
17     echo "</form>";
18 }
19 ?>

```

Tại trang thực hiện nhiệm vụ doupload.php ta cần kiểm tra xem, người dùng có nhấn nút upload từ trang bên kia hay không ?. Nếu có ta sẽ xử lý công đoạn upload ấy, nếu không ta cần thông báo lỗi cho người dùng và yêu cầu họ chọn file.

```

01 <<?php
02 if(isset($_POST['ok_upload']))
03 {
04     //thực thi upload
05 }
06 else
07 {
08     echo "Vui long chon hinh truoac khi truy cap vao trang nay";
09 }
10 ?>

```

Như vậy, khi người dùng nhấn upload, ta sẽ lấy được tham số là biến \$num từ liên kết \$_GET['file'] mà ta đã thiết lập trên phần form action của file upload ở trang trước.

Biến \$num này dùng để thực thi việc upload và tuần tự lấy các thông tin của file.

Để upload file từ máy lên hệ thống ta cần sử dụng hàm:

move_uploaded_file(tên_file_tạm,đường dẫn tới hệ thống).

Có thể diễn giải nôm na công việc của hàm này như sau: Ngay sau khi người dùng nhấn upload, hệ thống sẽ tạo ra 5 tham số ứng với từng file. Cụ thể là tên tạm (tmp_name), tên gốc (name), kích thước (size), định dạng mime (type) và lỗi (error) nếu có. Và khác với kiểu nhập liệu thông

thường, khi chúng ta sử dụng `<input type=file name=ten>` thì lúc này sẽ phát sinh một biến môi trường mới là `$_FILES['ten']` và dĩ nhiên nó sẽ đi kèm với 5 tham số cụ thể ở trên.

Ví dụ: `$_FILES['ten']['name']` //Lấy ra tên gốc của file.

Vậy ứng với cú pháp upload ở trên ta sẽ có:

```
move_uploaded_file($_FILES['img']['tmp_name'], "data/".$_FILES['img']['name']);
```

Cú pháp này sẽ tự động upload hình ảnh lên thư mục data trên hệ thống. Vậy ta cần phải tạo thư mục data ngang cấp với 2 file upload và doupload. Nếu là host thật, bạn cần CHMOD thư mục là 777 (cho phép thư mục có toàn quyền) thì mới thực thi upload file được.

Trong tình huống của chúng ta, vì là upload một lúc nhiều file, nên lúc này ta phải sử dụng tham số mảng tại trang upload ta đã khởi tạo.

Vậy ứng với biến \$num ta nhận từ upload ở trên ta sẽ triển khai code đơn giản như sau:

```
1 <?php
2 for($i=0; $i< $num; $i++)
3 {
4     move_uploaded_file($_FILES['img']['tmp_name'][$i], "data/".$_FILES['img']['name'][$i]);
5     $url="data/".$_FILES['img']['name'][$i];
6     $name=$_FILES['img']['name'][$i];
7 }
8 ?>
```

Với biến \$i là số lượng tăng lên theo vòng lặp. Vì mảng luôn bắt đầu bởi khóa là 0 nên ta cần phải thiết lập khởi tạo biến \$i với giá trị là 0. Biến \$url và \$name là đường dẫn hình ảnh và tên gốc của tấm hình được sử dụng để lưu vào CSDL.

Kế tới, ta cần lưu thông tin hình ảnh tương ứng vào cơ sở dữ liệu. Để làm được điều này, ta cần kết nối CSDL với cú pháp sau: (xem lại [**bài 10: kết hợp php & mysql trong ứng dụng website**](#)).

```
1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can't connect your
  database");
3 mysql_select_db("images",$conn);
4 ?>
```

Như vậy mỗi khi upload thành công một hình ảnh, ta phải thêm hình ảnh đó vào CSDL như sau:

```
1 <?php
2 $sql="insert into images(img_url,img_name) values('$url','$name)";
3 mysql_query($sql);
4 ?>
```

Sau khi thêm vào CSDL ta cần hiển thị thông tin và hình ảnh vừa upload để cho người dùng biết họ đã upload thành công.

Khi đó ta sử dụng một biến \$site để lưu đường dẫn gốc hiển thị trên textbox.

Demo Images Script - Copyright by QHOnline.Info

Upload Thành công file **akira.jpg**



www.qhonline.info

Images URL: <http://www.qhonline.info/data/akira.jpg>

Upload Thành công file **fullhouse7.jpg**



www.qhonline.info

Images URL: <http://www.qhonline.info/data/fullhouse7.jp>

Upload Thành công file **fullhouse8.jpg**



www.qhonline.info

Images URL: <http://www.qhonline.info/data/fullhouse8.jp>

Vậy hoàn chỉnh code cho file doupload.php này như sau:

```
01 <?php
02 $site="http://www.qhonline.info";
03 if(isset($_POST['ok_upload']))
04 {
05     $num=$_GET['file'];
06     echo "<h3>Demo Images Script - Copyright by QHOnline.Info</h3>";
07     $conn=mysql_connect("localhost","root","root") or die("can't connect your
database");
08     mysql_select_db("images",$conn);
09     for($i=0; $i< $num; $i++)
10     {
11         move_uploaded_file($_FILES['img']['tmp_name'][$i],"data/".$_FILES['img']['
12         'name'][$i]);
13         $url="data/".$_FILES['img']['name'][$i];
14         $name=$_FILES['img']['name'][$i];
15         $sql="insert into images(img_url,img_name) values('$url','$name)";
16         mysql_query($sql);
17         echo "Upload Thành công file <b>$name</b><br />";
```

```

17 echo "<img src='$url' width='120' /><br />";
18 echo "Images URL: <input type='text' name='link' value='$site/$url'
    size='35' /><br />";
19
20 }
21 mysql_close($conn);
22 }
23 else
24 {
25 echo "Vui long chon hinh truoc khi truy cap vao trang nay";
26 }
27 ?>

```

Bài 19: Viết ứng dụng tạo giỏ hàng - shopping cart - phần 1



Ở những bài trước, chúng ta đã đề cập tới các ứng dụng cơ bản, gần gũi với môi trường làm việc của chúng ta. Trong bài này, tôi sẽ tiếp tục hướng dẫn các bạn xây dựng hệ thống giỏ hàng (**shopping cart**) một ứng dụng phổ biến rất thường gặp trên các website cung cấp sản phẩm hiện nay. Nó gần như được thay thế cho việc mua sắm, việc lựa chọn thủ công mà chúng ta vẫn thường làm.

Do nội dung và kiến thức của bài này khá dài, nên tôi sẽ trình bày bài viết này ở 2 phần, để giúp các bạn dễ theo dõi và dễ nắm bắt các kiến thức mà tôi đề cập về ứng dụng.

Phần 1: Xây dựng trang hiển thị sản phẩm.

Đầu tiên, ta khởi tạo bảng CSDL đơn giản như sau (xem lại [bài 9: cơ bản về ngôn ngữ sql và mysql](#))

```

1 CREATE TABLE `books` (
2   `id` int(10) unsigned NOT NULL auto_increment,
3   `title` varchar(255) NOT NULL,
4   `author` varchar(100) NOT NULL,
5   `price` int(30) NOT NULL,
6   PRIMARY KEY (`id`)
7 );

```

Ở trên là một bảng lưu thông tin của các quyển sách với tên, tác giả, giá tiền của các quyển sách.

```

1 INSERT INTO `books` VALUES (1, 'PHP Can Ban', 'Kenny', 115);
2 INSERT INTO `books` VALUES (2, 'PHP Nang Cao', 'Kenny', 150);
3 INSERT INTO `books` VALUES (3, 'PHP Framework', 'Kenny', 300);

```



```
4 INSERT INTO `books` VALUES (4, 'Joomla Can Ban', 'Kenny', 100);
```

Tiếp tục ta tạo file style.css để trình bày dữ liệu đẹp mắt

```
01 body{
02 font:12px verdana;
03 }
04 .pro{
05 border:1px solid #999999;
06 margin:5px;
07 padding:5px;
08 width:400px;
09 }
10 a{
11 color:#666666;
12 text-decoration:none;
13 font-weight:900;
14 }
15 #cart{
16 border:1px solid #999999;
17 margin:5px;
18 padding:5px;
19 width:400px;
20 text-align:center;
21 }
```

Sau khi đã hoàn tất việc chuẩn bị cơ sở dữ liệu và giao diện, tiếp theo ta sẽ xây dựng trang hiển thị các món hàng trên website, cho phép người sử dụng lựa chọn sách khi truy cập.

Để liệt kê danh sách các quyển sách đang có trong database, ta cần kết nối CSDL với thao tác code như sau. (xem lại [bài 10: kết hợp php & mysql trong ứng dụng website](#)).

```
1 <?php
2 $connect=mysql_connect("localhost","root","root")
3 or die("Can not connect database");
4 mysql_select_db("shop",$connect);
5 ?>
```

Lúc này, ta có thể liệt kê các quyển sách bằng cú pháp sau:

```
01 <?php
02 $sql="select * from books order by id desc";
03 $query=mysql_query($sql);
04 if(mysql_num_rows($query) > 0)
05 {
06 while($row=mysql_fetch_array($query))
07 {
08 echo "<div class='pro'>";
```

```

09  echo "<h3>$row[title]</h3>";
10  echo "Tac Gia: $row[author] - Gia: ".number_format($row[price],3).
    VND<br />";
11  echo "<p align='right'><a href='addcart.php?item=$row[id]'>Mua Sach
    Nay</a></p>";
12  echo "</div>";
13  }
14  }
15  ?>

```

Đoạn code ở trên thực thi việc hiển thị sách nếu trong CSDL ít nhất 1 record. Và chúng sẽ liệt kê tiêu đề sách, tác giả, giá tiền. Ở đây, tôi sử dụng number_format() để lấy ra 3 số 000 cuối, ứng với đơn vị tiền tệ của Việt Nam là VNĐ.

Sau cùng tôi tạo một liên kết cho phép thêm quyền sách đó vào giỏ hàng nếu người mua cảm thấy ưng ý. (addcart.php sẽ là trang thêm vào giỏ hàng với tham số là mã của quyền sách).

Demo Shopping Cart

Ban khong co mon hang nao trong gio hang	
Joomla Layout Tac Gia: Bui Quoc Huy - Gia: 115.000 VND	www.qhonline.info Mua Sach Nay
Zend Framework Tac Gia: Bui Quoc Huy - Gia: 200.000 VND	www.qhonline.info Mua Sach Nay

Nói đến ứng dụng shopping cart thì hiện nay, trên thị trường tồn tại nhiều phương pháp code. Có thể sử dụng chuỗi để lưu giỏ hàng, cũng có thể lưu vào CSDL nháp giỏ hàng của người dùng và cũng có thể sử dụng mảng và session để lưu thông tin giỏ hàng. Trong khuôn khổ bài viết này, tôi sẽ sử dụng session và mảng để lưu thông tin giỏ hàng. (xem lại [Bài 7: Khái niệm cơ bản về Cookie và session trong PHP](#))

Tại trang addcart.php tôi cần khởi tạo một session và lưu mã quyền sách vào một mảng. Cụ thể là: \$_SESSION['cart'][\$id] (xem lại [bài 6: tổng quan về mảng và các hàm hỗ trợ](#)). Với \$id là mã quyền sách mà người dùng đã chọn ở trang xem hàng hóa (index.php). Mục đích chính của trang addcart này là lưu trữ hoặc tính toán lại số lượng sản phẩm khi mà họ lựa chọn. (lưu ý là số lượng các món hàng).

Một vấn đề đặt ra trong trang này, là làm thế nào để nhận biết món hàng người đó đã chọn hay chưa. Chẳng hạn. Lần đầu tôi chọn mua quyển A, sau đó tôi quay lại chọn mua tiếp quyển A. vậy trong giỏ hàng phải ghi nhận số lượng quyển A này là 2. Chứ không thể chỉ lưu là 1 được.

Vậy, lúc này ta sẽ kiểm tra xem. Quyển sách mà ta vừa chọn có tồn tại trong giỏ hàng hay chưa. Nếu có, ta phải tiến hành lấy số lượng đang có tăng lên 1 đơn vị. Còn nếu không, ta phải gán số lượng của chúng là 1.

Code xử lý hoàn chỉnh trang addcart.php này sẽ như sau:

```
01 <?php
02 session_start();
03 $id=$_GET['item'];
04 if(isset($_SESSION['cart'][$id]))
05 {
06     $qty = $_SESSION['cart'][$id] + 1;
07 }
08 else
09 {
10     $qty=1;
11 }
12 $_SESSION['cart'][$id]=$qty;
13 header("location:cart.php");
14 exit();
15 ?>
```

Nôm na, chúng ta có thể hiểu addcart chỉ đơn giản là xử lý số lượng hàng hóa và lưu chúng ở dạng mảng mà thôi.

Như vậy tại trang mua sách, ta cũng cần cho khách hàng biết rằng trong giỏ hàng của họ hiện đang có bao nhiêu món hàng. Hoặc nếu chưa có món nào, ta cũng phải báo cho họ biết về việc đó.

Vậy khi nào thì giỏ hàng rỗng ?. Đó là khi session của giỏ hàng không tồn tại Id của quyển sách nào. Cụ thể, \$_SESSION['cart'][\$id]. Khi id không tồn tại trong session này thì cũng là lúc giỏ hàng không tồn tại.

Vậy trước khi cho hiển thị giỏ hàng, ta cần kiểm tra xem có tồn tại id nào trong giỏ hàng hay không. Và vì id lưu ở dạng mảng đa chiều, nên ta cần dùng vòng lặp duyệt mảng foreach.

`foreach($_SESSION['cart'] as $k=>$v)`

Với \$k có ý nghĩa tương đương \$id quyển sách và \$v tương đương là số lượng của quyển sách trong giỏ hàng. Vậy nếu tồn tại biến \$k, thì tức có nghĩa là trong giỏ hàng có sách. Khi đó ta sử dụng một biến đã đề báo hiệu rằng sách có tồn tại trong giỏ hàng hay không.

```
01 <?php
02 $ok=1;
```

```

03 if(isset($_SESSION['cart']))
04 {
05     foreach($_SESSION['cart'] as $k=>$v)
06     {
07         if(isset($k))
08         {
09             $ok=2;
10         }
11     }
12 }
13 if ($ok != 2)
14 {
15     echo '<p>Ban khong co mon hang nao trong gio hang</p>';
16 } else {
17     $items = $_SESSION['cart'];
18     echo '<p>Ban dang co <a href="cart.php">'.count($items).' mon hang trong
gio hang</a></p>';
19 }
20 ?>

```

Trong đoạn code trên ta sử dụng hàm count nhằm đếm xem trong mảng hiện tại (giỏ hàng) hiện đang có bao nhiêu quyển sách. Và thực thi việc báo cho người dùng biết họ đang có bao nhiêu quyển sách trong giỏ hàng.

Toàn bộ code xử lý của trang mua sách (index.php) này như sau:

```

01 <?php
02 session_start();
03 ?>
04 <html>
05 <head>
06 <title>Demo Shopping Cart - Created By My Kenny</title>
07 <link rel="stylesheet" href="style.css" />
08 </head>
09 <body>
10 <h1>Demo Shopping Cart</h1>
11 <div id='cart'>
12 <?php
13 $ok=1;
14 if(isset($_SESSION['cart']))
15 {
16     foreach($_SESSION['cart'] as $k=>$v)
17     {
18         if(isset($v))
19         {

```

```

20     $ok=2;
21     }
22     }
23     }
24     if ($ok != 2)
25     {
26         echo '<p>Ban khong co mon hang nao trong gio hang</p>';
27     } else {
28         $items = $_SESSION['cart'];
29         echo '<p>Ban dang co <a href="cart.php">'.count($items).' mon hang trong
gio hang</a></p>';
30     }
31 ?>
32 </div>
33 <?php
34 $connect=mysql_connect("localhost","root","root")
35 or die("Can not connect database");
36 mysql_select_db("shop",$connect);
37 $sql="select * from books order by id desc";
38 $query=mysql_query($sql);
39 if(mysql_num_rows($query) > 0)
40 {
41     while($row=mysql_fetch_array($query))
42     {
43         echo "<div class=pro>";
44         echo "<h3>$row[title]</h3>";
45         echo "Tac Gia: $row[author] - Gia: ".number_format($row[price],3)."
VND<br />";
46         echo "<p align='right'><a href='addcart.php?item=$row[id]'>Mua Sach
Nay</a></p>";
47         echo "</div>";
48     }
49 }
50
51 ?>
52 </body>
53 </html>

```

Như vậy, trong bài này chúng ta đã hoàn tất trang xem sách và trang xử lý việc thêm sản phẩm mới vào giỏ hàng như thế nào. Mời các bạn đọc tiếp bài sau với tính năng quản lý, sửa, xóa từng món hàng hoặc cả giỏ hàng trong hệ thống.

Bài 20: Viết ứng dụng tạo giỏ hàng - shopping cart - phần 2



Ở phần 1 tôi đã trình bày phần hiển thị thông tin sách có trong cơ sở dữ liệu, chu trình thực thi của việc thêm một quyển sách vào giỏ hàng. Và cơ chế quản lý giỏ hàng của chúng ta như thế nào. Tiếp tục với phần 2, tôi sẽ trình bày cách quản lý giỏ hàng như chỉnh sửa món hàng, liệt kê hàng hóa trong giỏ hàng và xóa bỏ từng món hàng hay cả giỏ hàng như thế nào. Để có thể hiểu được phần 2. Các bạn vui lòng xem lại [bài 19: Viết ứng dụng tạo giỏ hàng – shopping cart – phần 1](#).

Phần 2: Xây dựng hệ thống quản lý giỏ hàng.

Sau khi đã thêm một món hàng, tại giỏ hàng ta cũng cần phải xử lý để hiển thị các món hàng đã có trong giỏ. Và như phương pháp ở phần 1, chúng ta tạm lưu mã sách trong session. (xem lại [Bài 7: Khái niệm cơ bản về Cookie và session trong PHP](#)). Vậy ở trang giỏ hàng này, ta sẽ khởi tạo session và sử dụng vòng lặp duyệt mảng foreach để lặp toàn bộ mã sách đang lưu trong session. Tiếp tục, ta lại chuyển chúng sang dạng chuỗi bằng hàm implode().

```
1 <?php
2 session_start();
3 foreach($_SESSION['cart'] as $key=>$value)
4 {
5     $item[]=$key;
6 }
7 $str=implode(",",$item);
8 ?>
```

Giả sử lúc này chuỗi của chúng ta sẽ có dạng 7,8,9. Công việc tiếp theo là kết nối CSDL để liệt kê các sản phẩm có mã như ở trên. (xem lại [bài 10: kết hợp php & mysql trong ứng dụng website](#)). Thay vì sử dụng select * from tên_bảng where id= ?? . Thì để tối ưu hơn, tôi sẽ sử dụng phép in trong SQL. Lúc này câu truy vấn sẽ tương đương:

```
1 <?php
2 $sql="select * from books where id in ('$str') order by id desc";
3 $query=mysql_query($sql);
4 while($row=mysql_fetch_array($query))
5 ?>
```

Tiếp tục, ta lặp toàn bộ thông tin sách bao gồm tên, tác giả, giá tiền và cả số lượng mà ta đã lưu trong session là \$_SESSION['cart'][\$ID_Món_Hàng]. ID_Món_hàng chính là thông tin ta lặp ra từ CSDL (\$row[id]).

Bên cạnh đó tại phần số lượng, ta sẽ đưa giá trị lưu ở session ra textbox (vì tại giỏ hàng, người dùng được phép điều chỉnh số lượng, nên lúc này ta cần tạo textbox cho họ điều chỉnh). Vì lặp toàn bộ các quyển sách nên tại tên của textbox số lượng ta cũng cần truyền id để nhận biết số lượng đó thuộc ID của quyển sách nào.

```

1 <?php
2 echo "<p align='right'>So Luong: <input type='text' name='qty[$row[id]]'
size='5' value='{$_SESSION['cart'][$row[id]]}' /> - ";
3 ?>

```

Chúng ta cũng cho người dùng được phép xóa 1 món hàng nào đó ra khỏi giỏ hàng của họ. Bằng cách truyền mã quyền sách của từng quyền vào liên kết delcart.php.

```

1 <?php
2 echo "<a href='delcart.php?productid=$row[id]'>Xoa Sach Nay</a></p>";
3 ?>

```

Tại đây, ta cũng cần tính luôn giá tiền của từng quyền sách tương ứng với số lượng mà họ đã chọn. Như vậy, số lượng là phần ta lưu ở session, còn giá tiền là phần ta lấy ra từ CSDL ứng với mảng \$row (\$row['price']).

```

1 <?php
2 echo "<p align='right'> Gia tien cho mon hang: ".
number_format($_SESSION['cart'][$row[id]]*$row[price],3) ." VND</p>";
3 ?>

```

Sau cùng, ta cần tính tổng tiền của toàn bộ sản phẩm có trong giỏ hàng. Bằng cách cộng dồn tổng giá tiền của từng món.

```

1 <?php
2 $total+=$_SESSION['cart'][$row[id]]*$row[price];
3 ?>

```

Phần còn lại, là chúng ta hiển thị giá tiền với đúng định dạng VND của Việt Nam.

```

1 <?php
2 echo "<b>Tong tien cho cac mon hang: <font color='red'>".
number_format($total,3) ." VND</font></b>";
3 ?>

```

Đồng thời, ta cũng tạo nút cho phép người dùng cập nhật. Và cho phép người dùng xóa toàn bộ giỏ hàng.

```

1 <?php
2 echo "<input type='submit' name='submit' value='Cap Nhat Gio Hang' />";
3 echo "<div class='pro' align='center'>";
4 echo "<b><a href='index.php'>Mua Sach Tiep</a> - <a
href='delcart.php?productid=0'>Xoa Bo Gio Hang</a></b>";
5 ?>

```

Như vậy, code đầy đủ sẽ là:

```

01 <?php
02 echo "<form action='cart.php' method='post'>";
03 foreach($_SESSION['cart'] as $key=>$value)
04 {
05   $item[]=$key;
06 }
07 $str=implode(",",$item);

```

```

08 $connect=mysql_connect("localhost","root","root") or die("Can not connect
    database");
09 mysql_select_db("shop",$connect);
10 $sql="select * from books where id in ($str)";
11 $query=mysql_query($sql);
12 while($row=mysql_fetch_array($query))
13 {
14     echo "<div class='pro'>";
15     echo "<h3>$row[title]</h3>";
16     echo "Tac gia: $row[author] - Gia: ".number_format($row[price],3)." VND<br
    />";
17     echo "<p align='right'>So Luong: <input type='text' name='qty[$row[id]]'
    size='5' value='{$_SESSION['cart'][$row[id]]}'> - ";
18     echo "<a href='delcart.php?productid=$row[id]'>Xoa Sach Nay</a></p>";
19     echo "<p align='right'> Gia tien cho mon hang: ".
    number_format($_SESSION['cart'][$row[id]]*$row[price],3) ." VND</p>";
20     echo "</div>";
21     $total+=$_SESSION['cart'][$row[id]]*$row[price];
22 }
23 echo "<div class='pro' align='right'>";
24 echo "<b>Tong tien cho cac mon hang: <font color='red'>".
    number_format($total,3)." VND</font></b>";
25 echo "</div>";
26 echo "<input type='submit' name='submit' value='Cap Nhat Gio Hang' />";
27 echo "<div class='pro' align='center'>";
28 echo "<b><a href='index.php'>Mua Sach Tiep</a> - <a
    href='delcart.php?productid=0'>Xoa Bo Gio Hang</a></b>";
29 echo "</div>";
30 ?>

```

Sau khi thiết lập thành công trang giỏ hàng cơ bản, lúc này ta đã có thể thêm sách một cách dễ dàng. Tuy nhiên, giả sử trong trường hợp không có sách thì sao ?. Chúng ta vẫn chưa xét đến trường hợp giỏ hàng rỗng thì sẽ như thế nào. Vậy khi nào thì giỏ hàng rỗng ?. Đó là khi session của giỏ hàng không tồn tại Id của quyển sách nào. Cụ thể, \$_SESSION['cart'][\$id]. Khi id không tồn tại trong session này thì cũng là lúc giỏ hàng không tồn tại.

Demo Shopping Cart

Zend Framework	www.qhonline.info
Tác giả: Bui Quoc Huy - Giá: 200.000 VND	
So Luong: <input type="text" value="4"/>	- Xóa Sách Nay
Giá tiền cho món hàng: 800.000 VND	

Joomla Layout	www.qhonline.info
Tác giả: Bui Quoc Huy - Giá: 115.000 VND	
So Luong: <input type="text" value="3"/>	- Xóa Sách Nay
Giá tiền cho món hàng: 345.000 VND	

Tong tien cho cac mon hang: 1,145.000 VND

Cập Nhật Giỏ Hàng	www.qhonline.info
-----------------------------------	----------------------------------------------------------

Mua Sách Tiếp - Xóa Bỏ Giỏ Hàng

Vậy trước khi cho hiển thị giỏ hàng, ta cần kiểm tra xem có tồn tại id nào trong giỏ hàng hay không. Và vì id lưu ở dạng mảng đa chiều, nên ta cần dùng vòng lặp duyệt mảng foreach.

```
foreach($_SESSION['cart'] as $k=>$v)
```

Với \$k có ý nghĩa tương đương \$id quyền sách và \$v tương đương là số lượng của quyền sách trong giỏ hàng. Vậy nếu tồn tại biến \$k, thì tức có nghĩa là trong giỏ hàng có sách.

```
01 <?php
02 $ok=1;
03 if(isset($_SESSION['cart']))
04 {
05     foreach($_SESSION['cart'] as $k=> $v)
06     {
07         if(isset($k))
08         {
09             $ok=2;
10         }
11     }
12 }
13 if($ok == 2)
```

```

14 {
15 // code xử lý giỏ hàng ở trên.
16 }
17 ?>

```

Đoạn code này, ta sử dụng biến \$ok để làm biến kiểm tra, mặc định khi load dữ liệu biến \$ok sẽ bằng 1. Và khi trong giỏ hàng tồn tại sách thì chúng ta sẽ thay đổi biến \$ok thành 2. Và gọi giỏ hàng như code ở trên.

Ở phần trên, ta cũng có đề cập khi người dùng tiến hành chỉnh sửa số lượng từng món hàng đơn lẻ qua textbox và nhấn cập nhật thì hệ thống sẽ tiến hành chỉnh sửa lại thông tin giỏ hàng. Vậy chúng ta sẽ xử lý như thế nào cho tương hợp đó.

Nếu chú ý, các bạn sẽ thấy dòng code chứa textbox cho phép người dùng nhập số lượng có một tham số đặc biệt là name=qty[\$row[id]]. Vậy tham số này được dùng để làm gì ?.

Tham số này, nói cho chúng ta biết số lượng đang hiển thị là thuộc mã sản phẩm nào. Vậy khi tiến hành cập nhật giỏ hàng ta sẽ kiểm tra, nếu \$qty[\$row[id]] mà có giá trị là 0. Tức là người đó muốn xóa bỏ giỏ hàng. Ngược lại, ta chỉ việc cập nhật giỏ hàng \$id tương ứng với số lượng nhập ở textbox.

```

01 <?php
02 if(isset($_POST['submit']))
03 {
04     foreach($_POST['qty'] as $key=>$value)
05     {
06         if( ($value == 0) and (is_numeric($value)))
07         {
08             unset($_SESSION['cart'][$key]);
09         }
10         elseif(($value > 0) and (is_numeric($value)))
11         {
12             $_SESSION['cart'][$key]=$value;
13         }
14     }
15     header("location:cart.php");
16 }
17 ?>

```

Việc xóa món hàng ở trên chỉ đơn giản là hủy bỏ session của id đó. Việc cập nhật số lượng chỉ đơn giản là gán đè số lượng người nhập (thẻ value trong textbox) vào số lượng đang lưu trong session.

Vậy code hoàn chỉnh của trang cart.php này sẽ như sau:

```

01 <?php
02 session_start();
03 if(isset($_POST['submit']))

```

```

04 {
05  foreach($_POST['qty'] as $key=>$value)
06  {
07      if( ($value == 0) and (is_numeric($value)))
08      {
09          unset ($_SESSION['cart'][$key]);
10      }
11      elseif(($value > 0) and (is_numeric($value)))
12      {
13          $_SESSION['cart'][$key]=$value;
14      }
15  }
16  header("location:cart.php");
17 }
18 ?>
19 <html>
20 <head>
21  <title>Demo Shopping Cart - Created By My Kenny</title>
22  <link rel="stylesheet" href="style.css" />
23 </head>
24 <body>
25 <h1>Demo Shopping Cart</h1>
26 <?
27 $ok=1;
28 if(isset($_SESSION['cart']))
29 {
30  foreach($_SESSION['cart'] as $k=> $v)
31  {
32      if(isset($k))
33      {
34          $ok=2;
35      }
36  }
37 }
38 if($ok == 2)
39 {
40
41     echo "<form action='cart.php' method='post'>";
42     foreach($_SESSION['cart'] as $key=>$value)
43     {
44         $item[]=$key;
45     }

```

```

46     $str=implode(",",$item);
47     $connect=mysql_connect("localhost","root","root") or die("Can not
connect database");
48     mysql_select_db("shop",$connect);
49     $sql="select * from books where id in ($str)";
50     $query=mysql_query($sql);
51     while($row=mysql_fetch_array($query))
52     {
53         echo "<div class='pro'>";
54         echo "<h3>$row[title]</h3>";
55         echo "Tac gia: $row[author] - Gia: ".number_format($row[price],3)."
VND<br />";
56         echo "<p align='right'>So Luong: <input type='text' name='qty[$row[id]]'
size='5' value='{$_SESSION['cart'][$row[id]]}'> - ";
57         echo "<a href='delcart.php?productid=$row[id]'>Xoa Sach Nay</a></p>";
58         echo "<p align='right'> Gia tien cho mon hang: ".
number_format($_SESSION['cart'][$row[id]]*$row[price],3) ." VND</p>";
59         echo "</div>";
60         $total+=$_SESSION['cart'][$row[id]]*$row[price];
61     }
62     echo "<div class='pro' align='right'>";
63     echo "<b>Tong tien cho cac mon hang: <font color='red'>".
number_format($total,3)." VND</font></b>";
64     echo "</div>";
65     echo "<input type='submit' name='submit' value='Cap Nhat Gio Hang'>";
66     echo "<div class='pro' align='center'>";
67     echo "<b><a href='index.php'>Mua Sach Tiep</a> - <a
href='delcart.php?productid=0'>Xoa Bo Gio Hang</a></b>";
68     echo "</div>";
69 }
70 else
71 {
72     echo "<div class='pro'>";
73     echo "<p align='center'>Ban khong co mon hang nao trong gio hang<br /><a
href='index.php'>Buy Ebook</a></p>";
74     echo "</div>";
75 }
76 ?>
77 </body>
78 </html>

```

Và cuối cùng, khi người dùng nhấn xóa toàn bộ giỏ hàng hay chỉ xóa một món hàng, ta sẽ gọi tới trang delcart.php. Vậy trang này sẽ xử lý như thế nào ?.

Tại đây, ta có thể nhận tham số là: \$_GET['productid'];

Tham số này sẽ là \$id mà chúng truyền qua liên kết. Nếu xóa toàn bộ giỏ hàng, tức ta sẽ truyền cho nó giá trị bằng 0. Lúc này, ta sẽ hủy toàn bộ \$_SESSION['cart']. Ngược lại, nếu là một \$id cụ thể, thì ta chỉ xóa món hàng đó mà thôi. \$_SESSION['cart'][\$id].

Code hoàn chỉnh của file delcart.php như sau:

```
01 <?php
02 session_start();
03 $cart=$_SESSION['cart'];
04 $id=$_GET['productid'];
05 if($id== 0)
06 {
07     unset($_SESSION['cart']);
08 }
09 else
10 {
11     unset($_SESSION['cart'][$id]);
12 }
13 header("location:cart.php");
14 exit();
15 ?>
```

Như vậy, chúng ta đã hoàn tất việc xây dựng hoàn chỉnh một hệ thống **shopping cart** đơn giản. Tuy rằng, đây không phải là một bài viết hoàn chỉnh trong việc xây dựng mô hình **thương mại điện tử**. Nhưng quá đó, phần nào giúp các bạn hiểu và dễ dàng phát triển hệ thống của mình có tổ chức hơn.