1, Cấu hình file config.php

Phần 7:

[**#PHP-1**](https://plus.google.com/s/%23PHP-1): TỪNG BƯỚC HỌC LẬP TRÌNH PHP CƠ BẢN QUA DỰ ÁN WEBSITE GIỚI THIỆU SẢN PHẨM - KẾT NỐI DB (PHẦN 7)  
  
Nội dung kiến thức phần này gồm:  
  
- Liên kết file (Require, include).  
- Biến.  
- Kiểu dữ liệu.  
  
Website giới thiệu sản phẩm là một website động (Có lưu trữ dữ liệu trong DB) nên để PHP có thể kết nối, tương tác với DB (Ở đây là MySQL) thì PHP đã cũng cấp sẳn các hàm để làm việc đó.  
  
Bây giờ, các bạn tạo một file dùng để viết mã lệnh PHP kết nối tới DB (Tôi gọi là file connect.php) trong thư mục libraries.  
  
Nội dung file libraries/connect.php như sau:  
  
<?php  
//Require  
require 'configs/config.php';  
  
//Kết nối  
$connect = mysql\_pconnect(DB\_SERVER, DB\_USERNAME, DB\_PASSWORD) or die('Not connected DB!');  
$db = mysql\_select\_db(DB\_DATABASE, $connect) or die('Not selected DB!');  
  
//Yêu cầu lưu trữ UTF8 (Tiếng Việt)  
mysql\_query('SET NAMES UTF8', $connect);  
?>  
  
Khối lệnh trên dùng để thực hiện kết nối với MySQL. Trong đó:  
  
- Hàm mysql\_pconnect dùng để kết nối tới DB với ba tham số lần lượt là tên server chứa DB (Đã khai báo ở hằng DB\_SERVER), tài khoản kết nối DB (Đã khai báo ở hằng DB\_USERNAME) và mật khẩu kết nối DB (Đã khai báo ở hằng DB\_PASSWORD). Nếu kết nối thành công thì PHP sẽ tiếp tục thực hiện các câu lệnh bên dưới, ngược lại nếu kết nối không thành công (Có thể thông tin khai báo bị sai) thì phần lệnh phía sau "die('Not connected DB!')" sẽ được thực thi in ra màn hình câu "Not connected DB!" và PHP sẽ dừng lại tại đây (Tức những câu lệnh bên dưới sẽ không được thực thi). Nếu kết nối thành công thì hàm mysql\_pconnect sẽ trả về dữ liệu kết nối và gán vào biến $connect.  
- Hàm mysql\_select\_db với hai tham số lần lượt là tên DB kết nối (Đã khai báo ở hằng DB\_DATABASE) và dữ liệu kết nối DB (Biến $connect).  
- Hàm mysql\_query dùng để truy vấn câu lệnh SQL "SET NAMES UTF8" với mục đích yêu cầu MySQL lưu trữ dữ liệu theo kiểu UTF8 (Tức là lưu được tiếng Việt).  
  
\* Liên kết file (Require, include):  
  
Phần trước, tôi có tạo một file config.php để khai báo các thông tin cấu hình website. Ở file connect.php này tôi cần sử dụng các thông tin đó dưới dạng các hằng đã được khai báo. Vì vậy, tôi cần liên kết với file config.php đã tạo bằng lệnh require như trên.  
  
Vậy, lệnh require các bạn có thể hiểu là dùng để liên kết một file PHP đã tồn tại vào một file PHP khác (Hoặc cũng có thể hiểu là với câu lệnh require như trên thì toàn bộ nội dung file configs/config.php đã nằm ngay tại vị trí câu lệnh).  
  
Ngoài lệnh require, chúng ta còn có một lệnh tương tự là require\_once với cách sử dụng như require chỉ khác một điều là lệnh require\_once có kiểm tra xem file được liên kết đã được liên kết trước đó chưa. Nếu file chưa được liên kết thì lệnh này sẽ liên kết mới, ngược lại thì lệnh này sẽ không liên kết file đó nữa.  
  
Có ý nghĩa và tác dụng tương tự lệnh require, require\_once tương ứng là lệnh include, include\_once. Hai lệnh require và include (Tương tự là require\_once và include\_once) về cơ bản là giống nhau nhưng có một sự khác biệt nhỏ các bạn có thể tìm hiểu thêm ở liên kết <http://php.net/manual/en/function.require.php>và <http://php.net/manual/en/function.include.php>.  
  
\* Biến:  
  
Ở khối lệnh trên tôi có sử dụng hai biến là $connect và $db. Như vậy, để tạo ra một biến trong PHP các bạn chỉ việc khai báo bằng cách đặt dấu "$" ngay đầu tên biến.  
  
Một số lưu ý khi khao báo biến:  
  
- Tên biến chỉ chấp nhận các số từ 0 đến 9, ký tự từ a đến z (Kể cả chữ in hoa) và dấu gạch dưới "\_".  
- Tên biến không được bắt đầu bằng số.  
- Không đặt tên biến là $this vì biến $this là một biến đặc biệt mặc định của PHP.  
  
Ví dụ một số cách đặt tên biến như sau:  
  
$a;  
$\_a;  
$a1;  
  
Khi khai báo biến các bạn có thể gán giá trị cho biến bằng toán tử gán "=". Ví dụ:  
  
$a = 1;  
$b = 'Hello';  
  
Các bạn có thể tìm hiểu thêm về biến trong PHP ở liên kết <http://php.net/manual/en/language.variables.php>.  
  
\* Kiểu dữ liệu:  
  
Kiểu dữ liệu trong PHP có thể được chia thành ba nhóm sau:  
  
- Nhóm scalar (Vô hướng).  
- Nhóm compound (Hỗn hợp).  
- Nhóm special (Đặc biệt).  
  
Nhóm scalar gồm:  
  
- Boolean: Gồm hai giá trị là "true" và "false".  
- Integer: Kiểu số nguyên.  
- Float: Kiểu số thực.  
- String: Kiểu chuổi.  
  
Đối với kiểu chuổi (String) thì các bạn phải đặt trong dấu nháy đơn hoặc dấu nháy đôi. Dấu nháy đơn hay dấu nháy đôi cũng có một chút khác biệt (Các bạn có thể tìm hiểu thêm).  
  
Nhóm compound gồm:  
  
- Array: Mãng (Sẽ có một bài giải thích rõ kiểu mãng).  
- Object: Đối tượng.  
  
Nhóm special gồm:  
  
- NULL: Kiểu null (Tức là không có gì hết. Lưu ý là khác với chuổi '').  
- Resource: Kiểu tài nguyên do PHP tạo ra.  
  
Các bạn có thể tìm hiểu thêm về kiểu dữ liệu trong PHP ở liên kết<http://php.net/manual/en/language.types.php>.

## Phần 9: Trang danh sách thành viên: Nội dung kiến thức phần này gồm: - Hàm. - Tham số. Trong thư mục admin/user, tạo mới file list.php để viết mã lệnh trang danh sách thành viên. Nội dung file admin/user/list.php như sau: <?php //Khởi động session session\_start(); //Kiểm tra nếu chưa đăng nhập thì quay về trang đăng nhập if(!isset($\_SESSION['user'])){ header('location:login.php'); } //Require các file cần thiết require '../../configs/config.php'; require '../../libraries/connect.php'; require '../../models/user.php'; //Lấy danh sách thành viên $user\_list = get\_user\_list(); //Require file giao diện (View) require '../../views/admin/user/list.tpl.php'; ?> Mở file models/user.php và thêm vào khối lệnh mới dưới đây: function get\_user\_list(){ //SQL $sql = "SELECT \* FROM tbl\_user ORDER BY user\_id DESC"; //Query và return return mysql\_query($sql); } Trong thư mục views/admin/user, tạo mới file list.tpl.php để trình bày giao diện trang danh sách thành viên. Nội dung file views/admin/user/list.tpl.php như sau: <html> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> <title>Quản trị - Thành viên - Danh sách</title> </head> <body> <table width="100%" cellpadding="10"> <tr> <th>ID</th> <th>Tài khoản</th> <th>Họ tên</th> <th>Trạng thái</th> <th>Ngày tạo</th> <th>Ngày chỉnh sửa</th> <th>Tác vụ</th> </tr> <?php while($user = mysql\_fetch\_assoc($user\_list)): ?> <tr> <td><?php echo $user['user\_id']; ?></td> <td><a href="<?php echo SITE\_URL . 'admin/user/edit.php?user\_id=' . $user['user\_id']; ?>"><?php echo $user['username']; ?></a></td> <td><?php echo $user['fullname']; ?></td> <td><?php echo ($user['status'] == 1) ? 'Kích hoạt' : 'Không kích hoạt'; ?></td> <td><?php echo date('d-m-Y H:i:s', $user['created']); ?></td> <td><?php echo date('d-m-Y H:i:s', $user['modified']); ?></td> <td><a href="<?php echo SITE\_URL . 'admin/user/delete.php?user\_id=' . $user['user\_id']; ?>">Xóa</a></td> </tr> <?php endwhile; ?> </table> </body> </html> Các bạn truy cập trang danh sách thành viên theo địa chỉ <http://localhost/admin/user/list.php> để kiểm tra. \* Hàm: Ở phần trước trong file models/user.php, tôi có khởi tạo một hàm tên là get\_user\_by\_username và ở phần này tôi cũng tạo một hàm là get\_user\_list. Theo như những gì các bạn thấy thì hàm là một khối lệnh làm một nhiệm vụ gì đó theo ý đồ của người lập trình nhằm mục đích "đóng gói" và "tái sử dụng" ở nhiều nơi. Để định nghĩa một hàm, các bạn thực hiện theo mẫu sau: function Tên\_hàm(){ //Khối lệnh thực hiện nhiệm vụ nào đó } Hoặc function Tên\_hàm(Tham\_số\_1, Tham\_số\_2, Tham\_số\_n){ //Khối lệnh thực hiện nhiệm vụ nào đó } Hoặc function Tên\_hàm(Tham\_số\_1 = 'Giá\_trị\_mặc\_định\_1', Tham\_số\_2 = 'Giá\_trị\_mặc\_định\_2', Tham\_số\_n = 'Giá\_trị\_mặc\_định\_n'){ //Khối lệnh thực hiện nhiệm vụ nào đó } Trong đó: - Tên hàm phải tuân theo quy tắc đặt tên mà PHP quy định là chỉ chấp nhận các số từ 0 đến 9, chữ cái từ a đến z, dấu gạch dưới "\_". Tên hàm không được bắt đầu bằng số và không được trùng lặp. - Một hàm có thể có hoặc không có tham số truyền vào. Nếu có nhiều hơn một tham số thì các tham số phân cách nhau mởi dấu phẩy ",". Các tham số có thể gán giá trị mặc định cho nó. Ví dụ 1: function get\_user\_list(){ //Những dòng lệnh các bạn đã biết ở trên } Tôi đã tạo ra một hàm tên là get\_user\_list và hàm này không có tham số truyền vào. Nhiệm vụ của hàm get\_user\_list là lấy danh sách thành viên từ DB và trả về lại kết quả. Ví dụ 2: function get\_user\_by\_username($username){ //Những dòng lệnh các bạn đã biết ở phần trước } Hàm ở trên tên là get\_user\_by\_username và hàm này có một tham số là $username. Nhiệm vụ của hàm get\_user\_by\_username là lấy thông tin thành viên từ DB và trả về lại kết quả. Ở hai ví dụ trên, tôi có nhắc đến "trả về lại kết quả" vậy đó là gì? Câu này được thể hiện bằng từ khóa "return" ở cuối mỗi hàm mà các bạn thấy. Ví dụ: return mysql\_query($sql); Dòng lệnh này ở hàm get\_user\_list mà tôi đã định nghĩa ở trên dùng để trả về lại kết quả mà người lập trình mong muốn (Ở ví dụ này đó là kết quả truy vấn từ DB). Một hàm có thể có giá trị trả về hoặc cũng có thể không có giá trị trả về tùy vào mục đích của người lập trình. Các bạn có thể tìm hiểu thêm về hàm ở liên kết<http://php.net/manual/en/language.functions.php>. \* Tham số: Ở hàm get\_user\_by\_username có khai báo một tham số là $username. Vậy tham số là những biến nhận giá trị truyền vào nhằm mục đích nào đó theo ý đồ của người lập trình. Các tham số và biến được khởi tạo bên trong hàm thì chỉ tồn tại bên trong hàm đó. Tham số bao gồm hai loại là: - Tham trị. - Tham biến. Các bạn có thể tìm hiểu thêm về tham số ở liên kết<http://php.net/manual/en/functions.arguments.php>.﻿