**PHP - MVC**

1. **PHP**

**\* Global variable and scope :**

- Trong **function** thì biến **local**. Muốn thay đổi biến **global** thì truyền : **global $tên biến.**

**\* Constants :**

  define("tên biến",  'giá trị');

**\* Built-in :**

- **Hàm toán học :**

**1. pow(2, 7) :** 2^7

**2. rand(** số bắt đầu, số kết thúc **) :** random number

**3. pi( ) :** trả về giá trị là một hằng số 3,14

**4. abs( number ) :** Gía trị tuyệt đối của một số.

**5. sqrt( number ) :** Căn bậc hai.

**6. max( number ) :** Trả về giá trị lớn nhất.

**7. min( number ) :** Trả về giá trị nhỏ nhất.

- **Hàm sử lý chuỗi :**

**1. strlen( string ) :** Độ dài của chuỗi.

**2. strtoupper( string ) :** Viết hoa hết.

**3. strtolower( string ) :** Viết thường hết.

- **Hàm sử lý array :**

**1. max( array ) :** Giá trị lớn nhất trong mảng.

**2. min( array ) :** Gía trị nhỏ nhất trong mảng.

**3. sort( mảng ) :** Sắp xếp giá trị từ nhỏ đến lớn.

**\* Use from in PHP :**

**isset( $\_POST[‘submit’] ): isset** kiểm tra xem biến đó đã được lưu trong bộ nhớ máy chưa

**in\_array( phần tử, mảng ) :** Kiểm tra xem giá trị có tồn tại trong mảng hay không.

**\* Connect database :**

  $connection = mysqli\_connect("localhost", "root", "", "loginapp");

 $result = mysqli\_query($connection, $query);

$row = mysqli\_fetch\_assoc($result)

**Assoc** : in kết quả ra array có tên field.

**Row** : in kết quả ra một mảng 1, 2, 3...

1. **Kiến trúc MVC**

**What:**

Viết tắt của Model – View – Controller

Mô hình thiết kể sử dụng trong kĩ thuật viết phần mềm.

**Controller** : Là thành phần **load** đầu tiên trong trang **website**. Đóng vai trò là **boss**, nhận yêu cầu từ **url**, hoặc từ một **form** điền dữ liệu. Sau đó chỉ đạo các **model** và **view** làm các công việc còn lại.

**Model :** Đóng vai trò kĩ thuật. Thao tác với csdl tạo các chức năng. Model xây dựng nên một cái nhà thô, phần còn lại là view.

+ xử lý logic , dữ liệu.

**View :** Phần hiển thị, **model** gửi đến **view** thì phải qua **contronller**.

Nhận dữ liệu từ **controller**.

**Why:**

Thể hiện tính chuyên nghiệp, dễ nâng cấp, bảo trì.

Làm nhóm nhiều người tốt hơn.

1. **Ví dụ**

**Controller:**

**Defined(‘BASEPATH’):** phải truy cập đường dẫn bắt buộc phải qua controller.

**Hàm contruct**: hàm khởi tạo, mỗi lần tạo lớp kế thừa, chống ghi đè lớp có sẵn.

**Hàm index:** hàm đầu tiên dùng để load.

**Echo:** câu lệnh để in ra.

1. **Cơ sở dữ liệu Mysql:**

Cơ sử lữ liệu là nơi lư chữ dữ liệu. Được quản lý bởi hệ quản trị mysql.

Mysql chỉ là giao diện để quản lý cơ sở dữ liệu, có tốc độ cao, ổn định và dễ sử dụng.

1. **Các bước để link đến file thư viện trong mô hình MVC:**

**File: autoload.php**

| $autoload['helper'] = array('url', 'file');

\*/

$autoload['helper'] = array("url");

Sử dụng hàm **base\_url():** để load ra đường dẫn tới file thư viện.

|   $autoload['libraries'] = array('user\_agent' => 'ua');

\*/

$autoload['libraries'] = array('database');

**File: config.php**

$config['base\_url'] = 'http://localhost:8080/01/';

1. **Thao tác insert với Model**

**Hàm insert\_id :** lấy id của phần tử vừa được thêm vào ( chỉ đi kèm với hàm **insert** ).

**Model :**

 $dulieu = array(

      'ten' => $ten,

      'tuoi' => $tuoi,

      'sdt' => $sdt);

    $this->db->insert('nhan\_vien', $dulieu);

    return $this->db->insert\_id();

**Controller :**

1. **Thao tác select với Model**

**Hàm var\_dump( ) or print\_r( ) :** In ra kết quả, check lỗi. Dữ liệu và kiểu dữ liệu.

**Hàm die() :** Dừng lại để check lỗi.

   $ketqua = $ketqua->result\_array(); // biến đổi kết quả thành một mảng

1. **Thao tác xóa từng phần tử trong database**

Xóa theo id.

1. **Mảng**

**Mảng 1 :** là một danh sách các phần tử cùng kiểu dữ liệu, nhưng mình lưu vào 1 biến.

**Mảng 2 :** Mảng dạng **key** **=>** **value**

Để duyệt vòng lặp dùng **foreach**.

Để nối chuỗi dùng dấu **chấm** **.**

1. **Xóa dữ liệu mySQL**

 <a href="showData/deleteData/<?php echo $value['id'] ?>

Truyền **id** theo đường dẫn trong **view,** truyền id vào hàm trong **model.**

   $this->db->where("id", $id);

    return $this->db->delete('so\_sim');

1. **Sử dụng hàm require**

**Require(‘tên file’) :** Thêm component vào.

**Include(‘tên file’) :** Thêm component vào.

1. **Sửa dữ liệu**

**Select :** Là lấy dữ liệu.

**Where :** Là điều kiện lấy dữ liệu.

**Get :** Trả về dữ liệu nhưng mà dạng phức tạp.

    $this->db->select('\*');

    $this->db->where('id', $i);

    $dulieu = $this->db->get('so\_sim');

    $dulieu = $dulieu->result\_array();

1. **Update dữ liệu**

**PROJECT QUẢN LÝ NHÂN SỰ**

1. **Sử lý trường upload file**

**Search keyword : upload file php w3school.**

1. **Tự động truy cập trang**

 <meta http-equiv="refresh" content="1;url=<?php echo base\_url() ?>index.php/Nhansu">

1. **Thêm dữ liệu**

**Model :**

public function insertDataToMysql($ten, $tuoi, $sdt){

    // xử lý thông tin nhận về và upload vào mysql

    $dulieu = array(

      'ten' => $ten,

      'tuoi' => $tuoi,

      'sdt' => $sdt);

    $this->db->insert('nhan\_vien', $dulieu);

    return $this->db->insert\_id();

  }

**Controller :**

$ten = $this->input->post('ten');

$tuoi = $this->input->post('tuoi');

$sdt = $this->input->post('sdt');

$this->load->model("nhansu\_model");

    $trangthai = $this->nhansu\_model->insertDataToMysql($ten, $tuoi, $sdt, $anhavatar, $linkfb, $sodonhang);

    if ($trangthai) {

      $this->load->view('insert\_success\_view');

    }

1. **Lấy dữ liệu**

**Model :**

public function getAllData(){

    $this->db->select("\*");

    $this->db->order\_by('id', 'asc');

    $dulieu = $this->db->get('nhav\_vien');

    $dulieu = $dulieu->result\_array();

    return $dulieu;}

**Controller :**

 public function index()

  {

    $this->load->model('nhansu\_model');

    $ketqua = $this->nhansu\_model->getAllData();

    $ketqua = array('mangketqua' => $ketqua);

    $this->load->view('nhansu\_view', $ketqua);

  }

**View :**

<?php foreach ($mangketqua as $value) : ?>

// code

<?php endforeach ?>

1. **Sửa dữ liệu.**

**Tạo : form** nhập dữ liệu người dùng, ấn nút edit.

**Model :**

  public function getDataById($id)

  {

    $this->db->select('\*');

    $this->db->where('id', $id);

    $dulieu = $this->db->get('nhan\_vien');

    $dulieu = $dulieu->result\_array(); // lấy về dữ liệu dạng mảng

    return $dulieu;

  }

**Controller :**

 public function sua\_nhansu($idnhanvao)

  {

    $this->load->model('nhansu\_model');

    $ketqua = $this->nhansu\_model->getDataById($idnhanvao);

    $ketqua = array("dulieukq" => $ketqua);

    $this->load->view('sua\_nhanvien\_view', $ketqua);

  }

**View :** Trong **view** phải biết được id của nội dung đó. Truyền **Id** vào **Controller** thông qua **url.**

<a href="<?php base\_url() ?>Nhansu/sua\_nhansu/<?php echo $value['id'] ?>

**Sửa và lưu dữ liệu:**

**View :** Trường **id hidden** . Thêm một trường **input type=”hidden” name = “AnhAvatar” -->** để lưu đường dẫn ảnh cũ.

**Model :**

$dulieucanupdate = array(

'id' => $id,

'so' => $so,

'gia' => $gia,

);

$this->db->where('id', $id);

return $this->db->update('so\_sim', $dulieucanupdate);

1. **Xóa dữ liệu**

**Xóa :** Lấy được **Id** của cái đấy.

**AJAX**

1. **Giới thiệu**

**Không cần thẻ form. Nút submit.**

**Ajax :**

**url :** Đường dẫn tới action.

**Data :** Dữ liệu cần gửi.

**Datatype: ‘json’.**

**.done() :** Nối vào thêm để hiển thị kết quả

**.fail() :** Khi hiện lỗi.

**Alway() :** Luôn luôn chạy.

1. **Xử lý upload file ajax,**

**Search google key word : jquery upload file.**

1. **ád**