



# LẬP TRÌNH PHP NÂNG CAO MÔ HÌNH MVC + ZF2

# Bài 1: Mô hình MVC

1. Lịch sử MVC
2. Giới thiệu mô hình MVC
3. Ưu điểm và nhược điểm của mô hình MVC
4. Lý do nên làm việc theo mô hình MVC

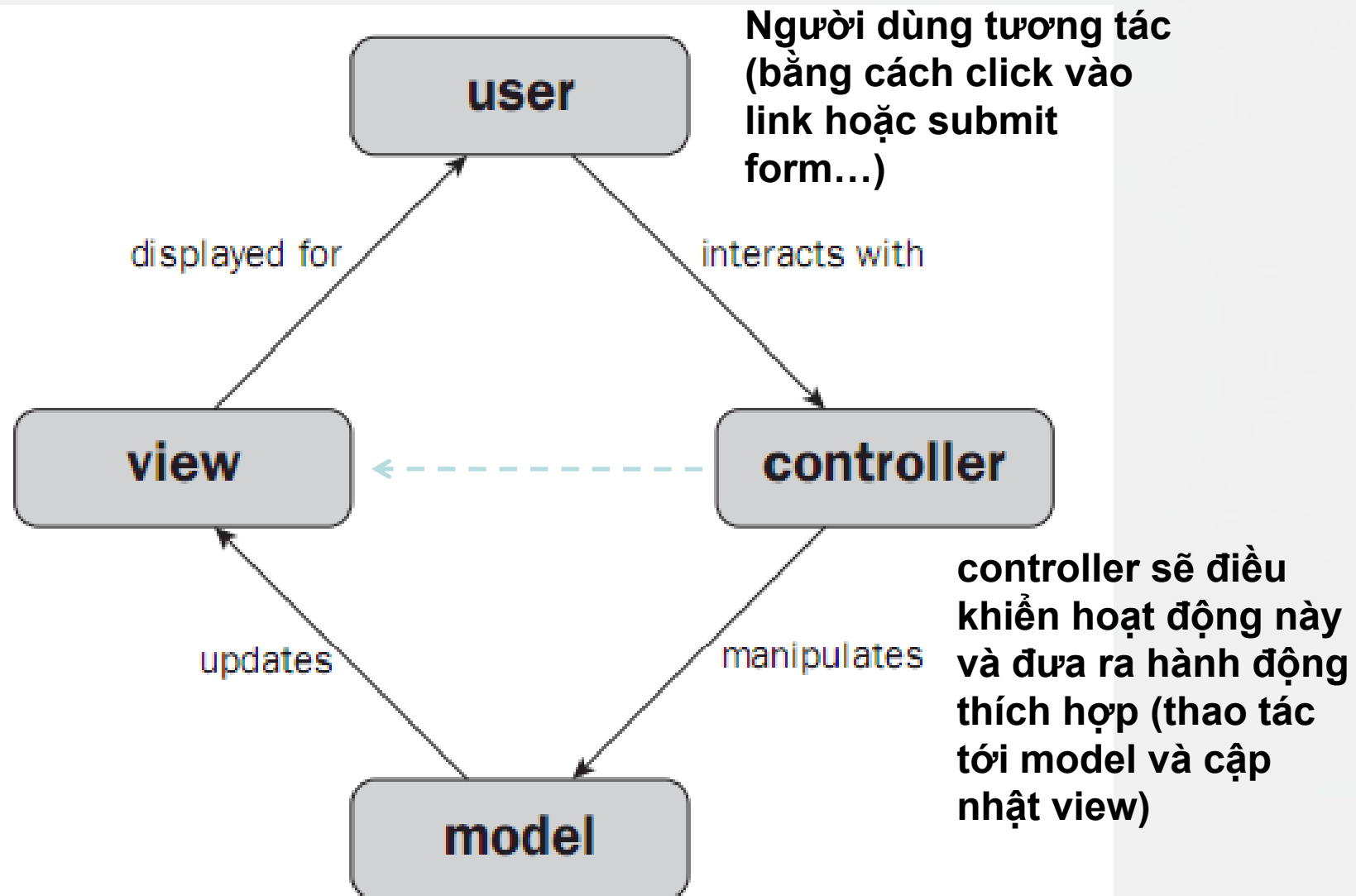
# 1. Lịch sử MVC

- Được đưa ra bởi các nhà nghiên cứu tại phòng thí nghiệm Xerox PARC Palo Alto, dựa trên Ngôn ngữ lập trình Smalltalk - lập trình theo hướng đối tượng và giao diện đồ họa GUI, công bố cuối những năm 70 đầu những năm 80.
- Thế hệ tiếp theo của MVC xuất hiện cùng với hệ điều hành NeXT và các phần mềm của nó.
- Kiến trúc này ngày càng được phát triển và hoàn thiện nhằm giải quyết các vấn đề phát sinh cũng như các giải pháp cho quá trình phát triển phần mềm. Vì vậy sau đó, lần lượt các MVC framework ra đời dựa trên mô hình MVC như: CodeIgniter, Zend...

## 2. Giới thiệu mô hình MVC

- MVC (Model – View - Controller): là một design pattern, theo hướng đối tượng, cho phép developer có thể chia nhỏ code của mình ra thành 3 phần:
  - Model: duy trì dữ liệu và các business rule
  - View: hiển thị dữ liệu và các thành phần trong giao diện người dùng
  - Controller: điều khiển các sự kiện của người dùng có ảnh hưởng đến Model và View

## 2. Giới thiệu mô hình MVC



## 2. Giới thiệu mô hình MVC

- Trong đó:
  - Controller: có thể gửi yêu cầu đến View liên kết của nó để thay đổi hiển thị trên View, cũng có thể gửi yêu cầu đến model để cập nhật trạng thái của model.
  - Model: thông báo đến các View và Controller có liên quan khi có thay đổi trạng thái. Thông báo này cho phép các View tạo ra các hiển thị được cập nhật và cho phép các Controller thay đổi các command
  - View: yêu cầu Model gửi các thông tin mà nó cần để tạo ra các hiển thị trên View.

### 3. Ưu điểm và nhược điểm của mô hình MVC

- Ưu điểm
  - Phát triển phần mềm: chuyên nghiệp hóa, có thể chia công việc cho nhiều nhóm chuyên môn khác nhau: nhóm thiết kế, nhóm lập trình, nhóm tổ chức database... Giúp phát triển ứng dụng nhanh, đơn giản, dễ nâng cấp.
  - Bảo trì: Với các lớp được phân chia, các thành phần của một hệ thống dễ được thay đổi, nhưng sự thay đổi có thể được cô lập trong từng lớp, hoặc chỉ ảnh hưởng đến lớp ngay gần kề của nó, chứ không làm ảnh hưởng cả chương trình.
  - Mở rộng: Việc thêm chức năng vào cho từng lớp sẽ dễ dàng hơn là phân chia theo cách khác.

### 3. Ưu điểm và nhược điểm của mô hình MVC

- **Nhược điểm**

- Thích hợp với dự án vừa và lớn còn đối với dự án nhỏ việc áp dụng mô hình MVC sẽ gây cồng kềnh, tốn thời gian trong quá trình phát triển.
- Tốn thời gian trung chuyển dữ liệu giữa các lớp



## 4. Lý do nên làm việc theo mô hình MVC

- Vì mô hình MVC đã chia ứng dụng thành M, V và C nên developer có thể tạo ra nhiều view và nhiều controller cho các model mà không phải đối mặt với việc thay đổi trong thiết kế model.
- => Giúp cho việc duy trì, di chuyển và tổ chức ứng dụng dễ dàng hơn.

## 4. Lý do nên làm việc theo mô hình MVC

- Đối với những người mới thì việc xây dựng ứng dụng dựa trên mô hình MVC phức tạp và lãng phí vì công việc này như là xây dựng một dự án lớn, tuy nhiên “bí mật” của MVC pattern không nằm ở chỗ viết code mà ở chỗ duy trì nó.
- Cho phép sửa đổi code mà không ảnh hưởng nhiều đến các thành phần khác.

## 4. Lý do nên làm việc theo mô hình MVC

- Cho phép việc làm việc nhóm trở nên dễ dàng hơn vì nhóm nào sẽ làm việc của nhóm đó dựa trên thế mạnh của mình.
  - Nhóm View Team sẽ chịu trách nhiệm về việc thực hiện các View
  - Nhóm Model Team sẽ lo về mặt dữ liệu
  - Nhóm Controller Team sẽ có cái nhìn tổng thể về các luồng của ứng dụng, quản lý các yêu cầu, làm việc với các model và lựa chọn View hiển thị cho khách hàng.

# Hết

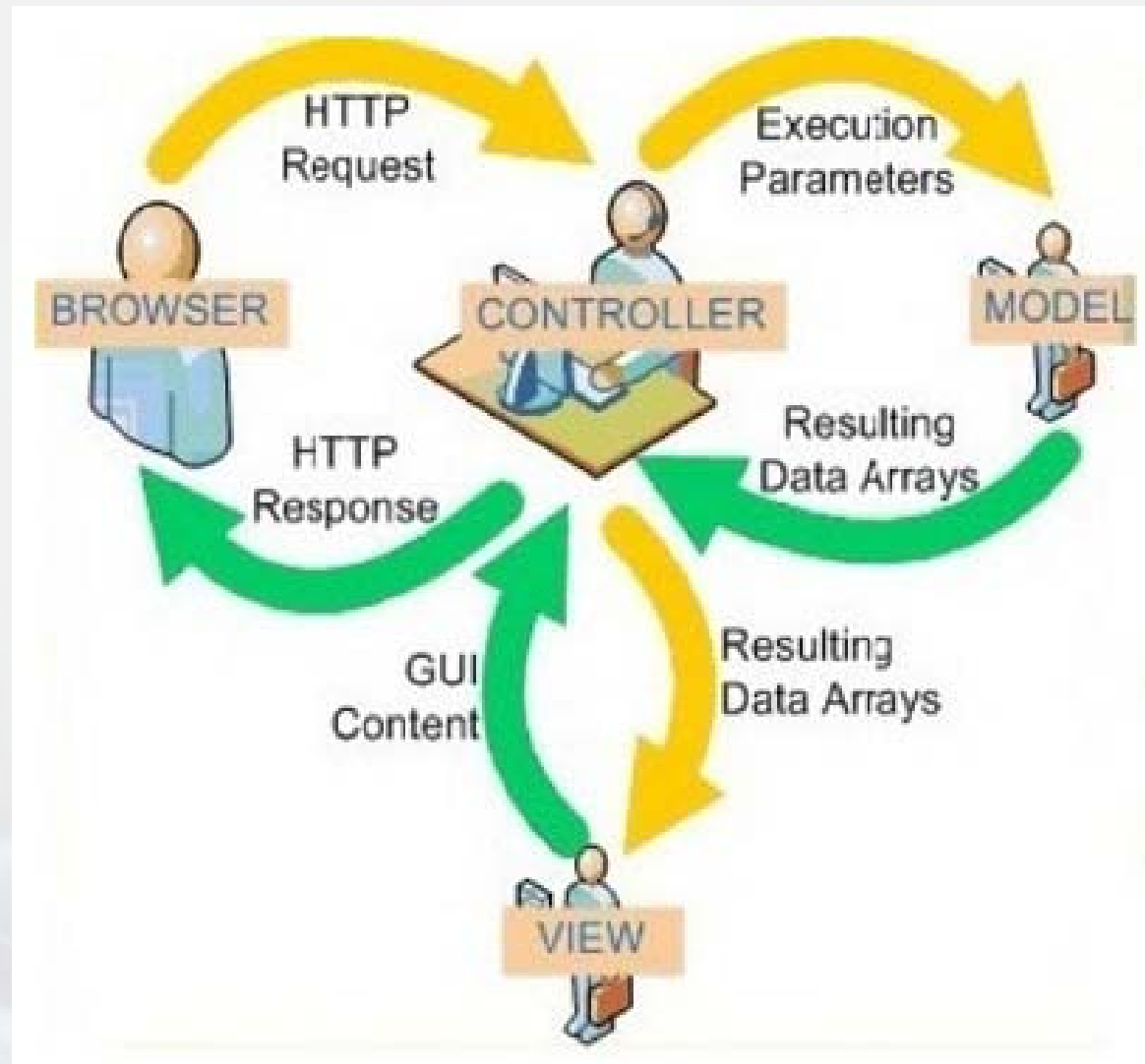


# LẬP TRÌNH PHP NÂNG CAO MÔ HÌNH MVC + ZEND FRAMEWORK

# **Bài 2: Xây dựng ứng dụng theo mô hình MVC**

- 1. Xây dựng Model**
- 2. Xây dựng Controller**
- 3. Xây dựng View**
- 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC**

## Mô hình MVC



## 1. Xây dựng Model

- Thường thì nên xây dựng Model trước bởi vì Model phục vụ như là ***nền tảng dữ liệu*** cho toàn bộ ứng dụng web;
- Nhiều chức năng của Controller phụ thuộc vào model.
- Trong Model sẽ có các phương thức xử lý liên quan đến dữ liệu: dữ liệu trực tiếp hoặc dữ liệu lấy từ CSDL



## 1. Xây dựng Model

- Ví dụ:
  - Ví dụ: Hiển thị sách có trong danh mục sách: dạng danh mục sách và dạng hiển thị từng sách theo lựa chọn của người dùng.

### Danh mục sách

| List of Books                   |                |
|---------------------------------|----------------|
| <i>Title</i>                    | <i>Author</i>  |
| <u>Jungle Book</u>              | R. Kipling     |
| <u>Professional CodeIgniter</u> | Thomas Mayer   |
| <u>PHP for Dummies</u>          | Some Smart Guy |

Hiện thị chi tiết sách theo lựa chọn của người dùng

*Detail of book*  
 Title: **Professional CodeIgniter**  
 Author: **Thomas Mayer**  
 Description: **A book about how to program with CodeIgniter Framework**

# 1. Xây dựng Model

- Phân tích yêu cầu:
  - Theo yêu cầu: các món ăn sẽ có các thông tin: title, author, description
  - ⇒ Xây dựng class Book để tổ chức thông tin này
  - ⇒ Class này sẽ được gọi sử dụng tại model của ví dụ

# 1. Xây dựng Model

```
class Book {  
    public $title;  
    public $author;  
    public $description;  
    public function __construct($title, $author,  
        $description)  
    {  
        $this->title = $title;  
        $this->author = $author;  
        $this->description = $description;  
    }  
}
```

# 1. Xây dựng Model

- Phân tích yêu cầu:
  - Cần hiển thị các thông tin về sách: Xây dựng Model là nơi xử lý thông tin về các sách
  - Với yêu cầu của người dùng là hiển thị danh sách các sách, và sau khi chọn một sách bằng cách click vào link thì sẽ hiển thị thông tin của sách được chọn => cần phải có hai function: lấy danh mục sách và lấy một sách theo tiêu đề được chọn

### 3. Xây dựng View

```
class bookModel {
    public function getBookList()
    {
        // here goes some hardcoded values to simulate the
        database
        return array(
            "Jungle Book" => new Book("Jungle Book", "R.
            Kipling", "A classic book."),
            "Professional CodeIgniter" => new Book("Professional
            CodeIgniter", "Thomas Mayer", "A book about how to
            program with CodeIgniter Framework"),
            "PHP for Dummies" => new Book("PHP for Dummies",
            "Some Smart Guy", "A tutorial book about programming
            language")
        );
    }
}
```

# 1. Xây dựng Model

```
public function getBook($title)
{
    // we use the previous function to get all the
    books and then we return the requested one.
    // in a real life scenario this will be done
    through a db select command
    $allBooks = $this->getBookList();
    return $allBooks[$title];
}
}
```

## 2. Xây dựng Controller

- Controller được xem là “cơ bắp” của ứng dụng.
- Mọi thứ mà người dùng ***có thể thực hiện*** trên trang web đều được thể hiện trong Controller.
- Điều khiển các sự kiện của người dùng có ảnh hưởng đến Model và View

## 2. Xây dựng Controller

- Ví dụ: cũng với ví dụ trên
  - Phân tích: Ta thấy cần phải xây dựng một Controller tương tác đến Model để lấy dữ liệu theo yêu cầu và tương tác đến View hiển thị dữ liệu có được từ Model .



## 2. Xây dựng Controller

```
class Controller {  
    public $model;  
    public function __construct()  
    {  
        $this->model = new bookModel();  
    }  
    public function invoke()  
    {  
        if (!isset($_GET['book']))  
        {  
            // khi người dùng chưa lựa chọn thì sẽ hiển  
            thị danh mục sách  
            $books = $this->model->getBookList();  
            include 'view/booklist.php';  
        }  
    }  
}
```

## 2. Xây dựng Controller

```
else
{
    // khi người dùng lựa chọn thì sẽ hiển thị
    thông tin của sách được lựa chọn
    $book = $this->model->getBook($_GET['book']);
    include 'view/viewbook.php';
}
}
}
```

### 3. Xây dựng View

- Xây dựng View là bước đơn giản nhất trong quá trình xây dựng ứng dụng theo mô hình MVC.
- Mỗi View là một kết quả hiển thị theo yêu cầu.

## 3. Xây dựng View

- Ví dụ: cũng với ví dụ trên
  - Phân tích: dựa vào yêu cầu ta thấy cần phải hiển thị:
    - Danh mục sách: view hiển thị danh mục sách
    - Một sách được chọn: view hiển thị một sách được chọn.

### 3. Xây dựng View

```
<table>
  <tr><td>Title</td><td>Author</td><td>Description</td></tr>
  <?php
    // hiển thị danh mục sách
    foreach ($books as $title => $book)
    {
      echo '<tr><td> <a
href="index.php?book=' . $book->title . '">' . $book->title . '</a> </td><td>' . $book->author . '</td>tr>';
    }
  ?>
</table>
```

## 3. Xây dựng View

```
<?php
    // hiển thị một sách cụ thể
    echo 'Title:' . $book->title . '<br/>';
    echo 'Author:' . $book->author . '<br/>';
    echo 'Description:' . $book->description .
        '<br/>';
?>
```

## Bài tập

- Hãy thực hiện cài đặt ví dụ nêu trên.

### Danh mục sách

| List of Books                                   |                |
|-------------------------------------------------|----------------|
| <i>Title</i>                                    | <i>Author</i>  |
| <a href="#"><u>Jungle Book</u></a>              | R. Kipling     |
| <a href="#"><u>Professional CodeIgniter</u></a> | Thomas Mayer   |
| <a href="#"><u>PHP for Dummies</u></a>          | Some Smart Guy |

Hiện thị chi tiết sách theo lựa chọn của người dùng

#### *Detail of book*

Title: **Professional CodeIgniter**

Author: **Thomas Mayer**

Description: **A book about how to program with CodeIgniter Framework**

## 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC

- Yêu cầu: Xây dựng ứng dụng hiển thị món ăn ba miền trong đó hiển thị danh mục món ăn ba miền kèm hình ảnh, khi người dùng chọn một món ăn bất kỳ thì hiển thị thông tin chi tiết của món ăn này.



## 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC



Khi người dùng  
chọn một món ăn

| MÓN NGON BA MIỀN |                                                                            |          |
|------------------|----------------------------------------------------------------------------|----------|
| Tên món          | Thông tin                                                                  | Hình ảnh |
| Phở              | Món ăn truyền thống với nguyên liệu chính là thịt bò, bánh phở và rau thơm |          |

## 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC

### ■ Phân tích yêu cầu

- Theo yêu cầu: các món ăn sẽ có các thông tin: tên món, mô tả, hình ảnh
- ⇒ Xây dựng class Mon\_an để tổ chức thông tin này
- ⇒ Class này sẽ được gọi sử dụng tại model của ứng dụng

## 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC

- Phân tích yêu cầu
  - Cần hiển thị các thông tin về món ăn: Xây dựng Model là nơi xử lý thông tin về các món ăn, theo từng miền
  - Với yêu cầu của người dùng là hiển thị món ăn theo miền, và sau khi chọn một món ăn bằng cách click vào link thì sẽ hiển thị thông tin chi tiết của món ăn được chọn => cần phải có hai function:
    - Lấy danh sách các món ăn: khởi tạo mảng các món ăn theo ba miền Bắc – Trung - Nam
    - Lấy một món ăn theo tên món được chọn

## 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC

- Phân tích yêu cầu
  - Cần phải xây dựng một Controller tương tác đến Model để lấy dữ liệu theo yêu cầu và tương tác đến View hiển thị dữ liệu có được từ Model.

## 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC

- Phân tích yêu cầu: Dựa vào yêu cầu ta thấy cần phải hiển thị:
  - Danh mục món ăn: view hiển thị danh mục món ăn theo miền
  - Một món ăn được chọn: view hiển thị chi tiết món ăn được chọn.

### 4. Xây dựng ứng dụng đầu tiên theo mô hình MVC

- Thực hiện
  - Triển khai xây dựng ứng dụng sau khi đã phân tích.



Khi người dùng chọn một món ăn

| MÓN NGON BA MIỀN |                                                                            |                                                                                       |
|------------------|----------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
| Tên món          | Thông tin                                                                  | Hình ảnh                                                                              |
| Phở              | Món ăn truyền thống với nguyên liệu chính là thịt bò, bánh phở và rau thơm |  |

# Hết



# LẬP TRÌNH PHP NÂNG CAO MÔ HÌNH MVC + ZEND FRAMEWORK



# Bài 3: Xây dựng ứng dụng có sử dụng CSDL theo mô hình MVC

1. Xây dựng Model
2. Xây dựng View
3. Xây dựng Controller

# 1. Xây dựng Model

- Các bước thực hiện
  - Bước 1: Vì có sử dụng CSDL nên trong Model ta cần phải xây dựng ***lớp Database*** để tương tác đến CSDL và xây dựng các phương thức cần thiết có liên quan như đến CSDL như: kết nối CSDL, tạo truy vấn, thực hiện truy vấn, lấy dữ liệu trả về dưới dạng mảng, dạng đối tượng...

# 1. Xây dựng Model

```
class Database {  
    var $_sql = '';  
    var $_connection = '';  
    var $_cursor = null;  
    // các phương thức  
    // phương thức khởi tạo lớp, kết nối tới CSDL  
    // tạo và gán câu lệnh truy vấn  
    // thực thi câu lệnh truy vấn  
    // lấy các dòng trong trong CSDL và gán vào cho  
    // mảng các đối tượng  
    // lấy một dòng thỏa điều kiện trong CSDL và gán  
    // cho đối tượng  
    // ngắt kết nối  
}
```

# 1. Xây dựng Model

```
//phương thức khởi tạo lớp, kết nối tới CSDL
function Database($host,$user,$pass,$db)
{
    $this->_connection = @mysql_connect( $host, $user,
    $pass);
    if (!$this->_connection )
    {die( "Cannot connect to MySQL" );
    }
    if ($db != " && !mysql_select_db( $db, $this-
    >_connection))
    {die ( "Cannot open database $db: ".mysql_error() );
    }
}
```

# 1. Xây dựng Model

```
// tạo và gán câu lệnh truy vấn
function setQuery( $sql)
{
    $this->_sql = $sql;
}

// thực thi câu lệnh truy vấn
function query()
{
    $this->_cursor = mysql_query( $this->_sql, $this->
    _connection );
    return $this->_cursor;
}
```

# 1. Xây dựng Model

```
// lấy các dòng trong trong CSDL và gán vào cho mảng các đối tượng
function loadAllObject() {
    if (!($cur = $this->query())) {
        return null;
    }
    while ($object = mysql_fetch_object($cur))
    {
        $array[] = $object;
    }
    mysql_free_result( $cur );
    return $array;
}
```

# 1. Xây dựng Model

```
// lấy một dòng thỏa điều kiện trong CSDL và gán cho  
đối tượng  
function loadAnObject()  
{  
    if (!($cur = $this->query())) {  
        return null;  
    }  
    $ret = null;  
    if ($object = mysql_fetch_object( $cur )) {  
        $ret = $object;  
    }  
    mysql_free_result( $cur );  
    return $ret;  
}
```

## 1. Xây dựng Model

```
// ngắt kết nối  
function disconnect()  
{  
    mysql_close( $this->_connection );  
}
```



# 1. Xây dựng Model

- Các bước thực hiện:
  - Bước 2: Xây dựng các lớp Model cần thiết cho ứng dụng. trong các lớp Model này thì đối tượng Database là một thuộc tính của lớp. Trong một lớp Model sẽ có phương thức khởi tạo Model và các phương thức cần thiết như lấy và hiển thị thông tin, cập nhật thông tin, thêm mới thông tin hay xóa thông tin...

# 1. Xây dựng Model

- Ví dụ: Xây dựng một ứng dụng để hiển thị thông tin sản phẩm quần áo của Shop Bibi như sau: Khi người dùng đến thăm Website thì sẽ hiển thị thông tin của tất cả các sản phẩm quần áo, còn khi người dùng chọn cụ thể một sản phẩm thì chỉ hiển thị thông tin của sản phẩm được chọn.

## Danh mục sản phẩm

| Danh mục sản phẩm                                                                                                                          |                                                                                                                                       |                                                                                                                                          |                                                                                                                                     |
|--------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <br><b>Áo bé gái hoa Tulip</b><br>Giá: 520.000 VNĐ        | <br><b>Váy thời trang hồng</b><br>Giá: 580.000 VNĐ   | <br><b>Áo thun chấm nâu</b><br>Giá: 70.000 VNĐ          | <br><b>Váy BURBERRY</b><br>Giá: 160.000 VNĐ       |
| <br><b>Quần lửng hồng</b><br>Giá: 75.000 VNĐ              | <br><b>Đầm dạ hội cho bé</b><br>Giá: 115.000 VNĐ     | <br><b>Áo thun bé trai tay ngắn</b><br>Giá: 110.000 VNĐ | <br><b>Áo thun xanh trắng</b><br>Giá: 130.000 VNĐ |
| <br><b>Áo sát nách vàng ô vuông</b><br>Giá: 157.000 VNĐ | <br><b>Áo thun bé trai Pig</b><br>Giá: 110.000 VNĐ | <br><b>Đồ kiểu bé trai</b><br>Giá: 150.000 VNĐ        |                                                                                                                                     |

## Sản phẩm được chọn



# 1. Xây dựng Model

- Phân tích yêu cầu:

Theo như yêu cầu trên thì cần phải xây dựng ProductModel có:

- Phương thức khởi tạo
- Phương thức lấy danh sách các sản phẩm
- Phương thức lấy một sản phẩm theo khóa được chọn

# 1. Xây dựng Model

```
// khai báo và khởi tạo lớp ProductModel
class ProductModel {
    var $dao;
    function ProductModel()
    {
        $this->dao = new Database('localhost', 'root',
        '', 'ql_quan_ao');
    }
    ...
}
```

# 1. Xây dựng Model

```
// lấy tất cả các sản phẩm
function listOfProducts()
{
    $this->dao->setQuery("Select * from products,
    type where products.type_id = type.type_id");
    $result = $this->dao->loadAllObject();
    return $result;
}
```

# 1. Xây dựng Model

```
// lấy một sản phẩm dựa trên id của sản phẩm được  
chọn  
function loadAProduct($id)  
{  
    $this->dao->setQuery("Select * from products  
where id = ". $id);  
    $result = $this->dao->loadAnObject();  
    return $result;  
}
```

## 2. Xây dựng View

- Xây dựng các View cần thiết cho ứng dụng, tương tự như cách xây dựng View đã giới thiệu ở bài trước.



## 2. Xây dựng View

- Ví dụ: như đã nêu trong phần Xây dựng Model
- Phân tích yêu cầu: với yêu cầu trên thì cần phải có 2 View:
  - View hiển thị các sản phẩm
  - View hiển thị chi tiết một sản phẩm

## 3. Xây dựng Controller

- Xây dựng các Controller cần thiết cho ứng dụng, tương tự như cách xây dựng Controller đã giới thiệu ở bài trước.

### 3. Xây dựng Controller

- Ví dụ: như đã nêu trong phần Xây dựng Model
- Phân tích yêu cầu:
  - Với yêu cầu như trên thì cần xây dựng một lớp ProductController gồm các phương thức:
    - Gọi sử dụng model và view đã xây dựng

## 3. Xây dựng Controller

```
class ProductController {  
    public $product_model;  
    public function __construct()  
    {  
        $this->product_model = new ProductModel();  
    }  
}
```

## 3. Xây dựng Controller

```
public function invoke()
{
    if (!isset($_GET['id']))
    {
        $products = $this->product_model->
listOfProducts();
        include 'view/list_Product.php';
    }
    if (isset($_GET['id']))
    {
        $product = $this->product_model->
loadAProduct($_GET['id']);
        include 'view/productView.php';
    }
}
}
```



# Hết

