

# MÔ HÌNH MVC & ĐỐI TƯỢNG PDO

# NỘI DUNG BÀI HỌC

- Tìm hiểu về mô hình MVC
- Đối tượng PDO
- Cách tổ chức code theo mô hình MVC

# Giới thiệu

- Mô hình MVC cơ bản ở đây là viết tắt của 3 từ **M – Model, V – View, C – Controller**.
- Mô hình MVC này được áp dụng trong hầu hết các dự án, trang web và có thể nói là rất quen thuộc với người lập trình.
- Mô hình MVC được tạo ra từ 3 lớp khác nhau nhưng lại tương tác với nhau, giúp chúng ta có thể nhìn vào mô hình này và thao tác xử lý nhanh chóng hơn.

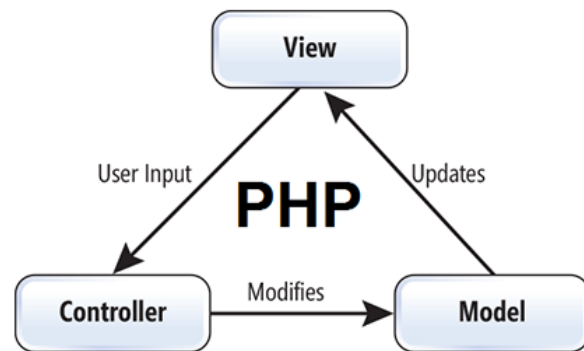
# Mô hình MVC

**1. Model:** có nhiệm vụ thao tác với cơ sở dữ liệu, nghĩa là nó sẽ chứa tất cả các hàm, các phương thức truy vấn trực tiếp với dữ liệu và controller sẽ lắng nghe những gì từ model đưa ra và sau đó trả về **View**

**2. View:** có nhiệm vụ tiếp nhận dữ liệu từ controller và hiển thị nội dung sang các đoạn mã HTML, nói dễ hiểu hơn thì View sẽ chứa giao diện website

**3. Controller:** đóng vai trò trung gian giữa Model và View. Nó có nhiệm vụ tiếp nhận yêu cầu từ client

sau đó xử lý request, load model tương ứng và gửi data qua view tương ứng rồi trả kết quả về cho client



# Ưu điểm và nhược điểm mô hình MVC

## Ưu điểm:

- Hệ thống phân ra từng phần riêng biệt nên dễ dàng phát triển
- Chia thành nhiều modun nhỏ nên nhiều người có thể làm chung dự án
- Dễ dàng nâng cấp và có khả năng bảo trì
- Dễ dàng debug trong quá trình tổ chức code

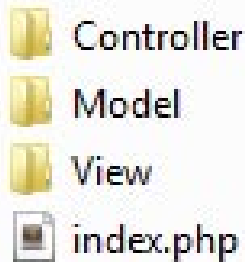
# Ưu điểm và nhược điểm mô hình MVC

## Nhược điểm:

- Kết quả cho ra và chạy chương trình không khác gì khi code “PHP thuần”.
- Tốc độ chương trình chạy chậm hơn PHP thuần, nhưng chậm không đáng kể và không phải là vấn đề quá lớn

# Tạo mô hình MVC

**Bước 1:** Tạo 3 thư mục với tên Model, View và Controller tương ứng. Bên ngoài cùng cấp với các folder sẽ là file: index.php



# Tạo mô hình MVC

**Bước 2:** Tạo thêm một folder: config, để viết kết nối theo hướng đối tượng.

```
myConfig.php
<?php
/**
 *
 */
class Connect
{
    private $localhost = "localhost";
    private $user = "root";
    private $pass = "";
    private $db = "db_main";
    protected $conn = null;

    function __construct()
    {
        $this->conn = mysqli_connect($this->localhost, $this->user, $this->pass, $this->db);
        if (!$this->conn) {
            echo "Connect Error";
            exit();
        }else{
            mysqli_set_charset($this->conn, "utf8");
        }
    }
}
```



# Tạo mô hình MVC

**Bước 3:** Trong model, tạo file Member\_m.php kế thừa lớp Connect và thực hiện viết các hàm truy vấn.

```
<?php
include_once 'config/myConfig.php';

/**
 *
 */
class Member_m extends Connect
{
    function __construct()
    {
        parent::__construct(); // Gọi hàm kết nối ở file myConfig, để sử dụng $conn
    }
}
```

# Tạo mô hình MVC

**Bước 4:** Trong controller, tạo file Member\_c.php kế thừa lớp Member\_m và thực hiện yêu cầu những gì model trả ra sau đó hiển thị giao diện về view

```
<?php
include_once 'model/Member_m.php';

/**
 *
 */
class Member_c extends Member_m
{
    private $mem; // khai báo biến với phạm vi không công khai

    function __construct()
    {
        $this->mem = new Member_m(); // Khởi tạo đối tượng sang class: Member_m
    }
}
```


# THỰC HÀNH



Tạo và tổ chức code theo mô hình PDO-MVC, quản lý thành viên trong lớp học

# Đối tượng PDO

- **PDO là gì?**
  - Là đối tượng có sẵn được cung cấp từ phiên bản PHP > 5.1, mục đích ra đời giúp bạn thao tác với database thông qua các Object (đối tượng) làm cho công việc trở nên hiệu quả, dễ dàng hơn.
  - PDO cung cấp các cơ chế Prepared Statements, Stored Procedures



## So sánh giữa PDO và mysqli

	PDO	MySQLi
Database hỗ trợ	Hơn 12 loại	Chỉ hỗ trợ MySQL
API	Hướng đối tượng (OOP)	Hướng đối tượng (OOP) - Hướng thủ tục (Procedural)
Kết nối Database	Dễ dàng	Dễ dàng
Đặt tên tham số	Có	Không
Object Mapping	Có	Có
Prepared Statements	Có	Không
Hiệu năng	Cao	Cao
Stored Procedures	Có	Có

# Ưu điểm của PDO

- Là một lớp truy xuất cơ sở dữ liệu cung cấp một phương pháp thống nhất để làm việc với nhiều loại cơ sở dữ liệu khác nhau
- Khi làm việc với PDO bạn sẽ không cần phải viết các câu lệnh SQL cụ thể mà chỉ sử dụng các phương thức mà PDO cung cấp

# Sử dụng PDO

## 1. Kết nối CSDL

```
$conn = new PDO('mysql:host=localhost;dbname=ql_hocvien', $username, $password);
```

# Truy vấn dữ liệu

```
$sql = "SELECT *FROM tbl_hocvien"; //return $this->pdo->query($sql);  
$pre = $this->pdo->prepare($sql);  
$pre->execute();  
$result = $pre->fetchAll(PDO::FETCH_ASSOC);  
return $result;
```

## Trong đó:

- prepare: Chuẩn bị một câu lệnh SQL. (với câu lệnh sql không cần phải khai báo và viết cụ thể).
- execute(): Dùng để thực thi và chạy câu lệnh sql
- PDO::FETCH\_ASSOC là lấy ra dữ liệu sau khi truy vấn dưới dạng key value, key và các trường trong db, value là giá trị tương ứng



# "Đọc" dữ liệu từ database

- Thông qua phương `fetch()` hoặc `fetchAll` để đọc dữ liệu từ db với nhiều lựa chọn khác nhau như:

`PDO::FETCH_ASSOC`: Trả về dữ liệu dạng mảng với key là tên của column (column của các table trong database)

`PDO::FETCH_BOTH` (default): Trả về dữ liệu dạng mảng với key là tên của column và cả số thứ tự của column

`PDO::FETCH_CLASS`: Gán giá trị của từng column cho từng thuộc tính (property/attribute) của một lớp Class theo tên column và tên thuộc tính.

`PDO::FETCH_INT`: Gán giá trị của từng column cho từng thuộc tính của một Class Instance (thể hiện của một lớp)..v..v...

# Sử dụng PDO

## 2. Insert

```
$sql = "INSERT INTO tbl_hocvien(id_khoa, tenHV, phoneHV, email, address) VALUES(:id_khoa, :tenHV, :phoneHV, :email, :address)";  
$pre = $this->pdo->prepare($sql);  
$pre->bindParam(':id_khoa', $idKhoa);  
$pre->bindParam(':tenHV', $tenHV);  
$pre->bindParam(':phoneHV', $phoneHV);  
$pre->bindParam(':email', $email);  
$pre->bindParam(':address', $address);  
$pre->execute();
```

# Sử dụng PDO

## 3. Delete

```
$pre = $this->pdo->prepare("DELETE FROM tbl_hocvien WHERE id_hocvien = :id");  
$pre->bindParam(':id', $id);  
$pre->execute();
```

# AJAX TRONG PHP

# Giới thiệu Ajax

- Ajax là bộ máy trung gian, xử lý tương tác dữ liệu giữa client và server, với dữ liệu trả về từ server có thể là json, xml, text... tùy theo cách chúng ta muốn lấy ra dữ liệu.
- Hiểu đơn giản là ajax giúp cho website load nhanh hơn, thao tác được mượt mà hơn và không phải load lại trang

# Phương thức trong Ajax

- Để gửi dữ liệu đi và lấy dữ liệu về trong Ajax chúng ta có 3 phương thức:

1. **Load**
2. **Post**
3. **Get**