



## CT275 – CÔNG NGHỆ WEB

# Mục tiêu

---

Trình bày kiến trúc MVC và  
ứng dụng kiến trúc này trong thiết kế ứng dụng web

# Nội dung

---

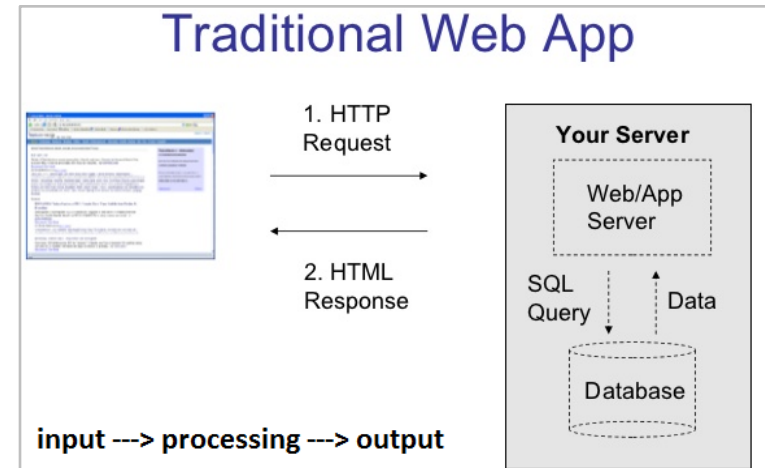
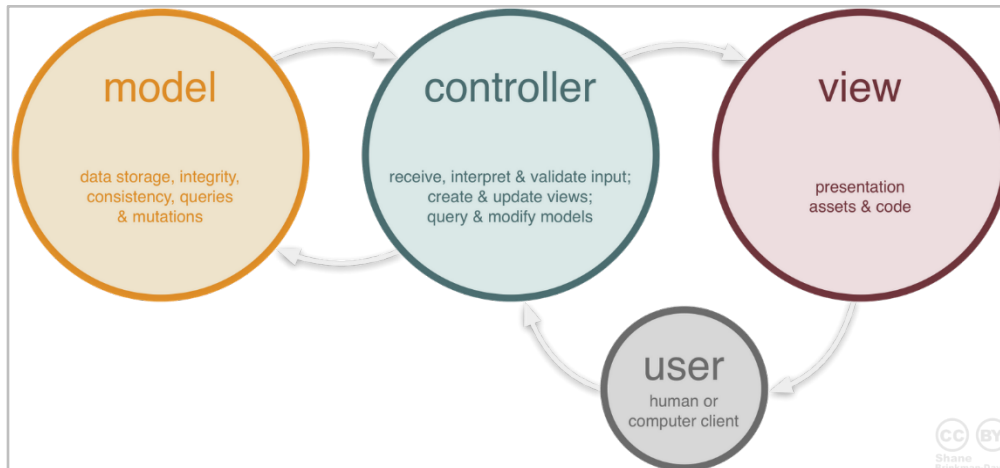
- Kiến trúc MVC
  - MVC là gì?
  - Các thành phần trong kiến trúc MVC
  - Tại sao cần MVC?
  - Luồng thực thi trong MVC
  - Front Controller
- Xây dựng ứng dụng Web với MVC
  - Cơ chế rewrite
  - Tổ chức thư mục dự án
  - Các router phổ biến

# Giới thiệu kiến trúc MVC

*MVC Architecture Introduction*

# Mô hình MVC là gì?

- **MVC** = **M**odel – **V**iew – **C**ontroller
- Là một mẫu thiết kế kiến trúc (architecture pattern) cho các **ứng dụng tương tác**
- Trong kiến trúc này, các thành phần của một ỨD được tổ chức thành 3 phần:



# Các thành phần trong kiến trúc MVC

---

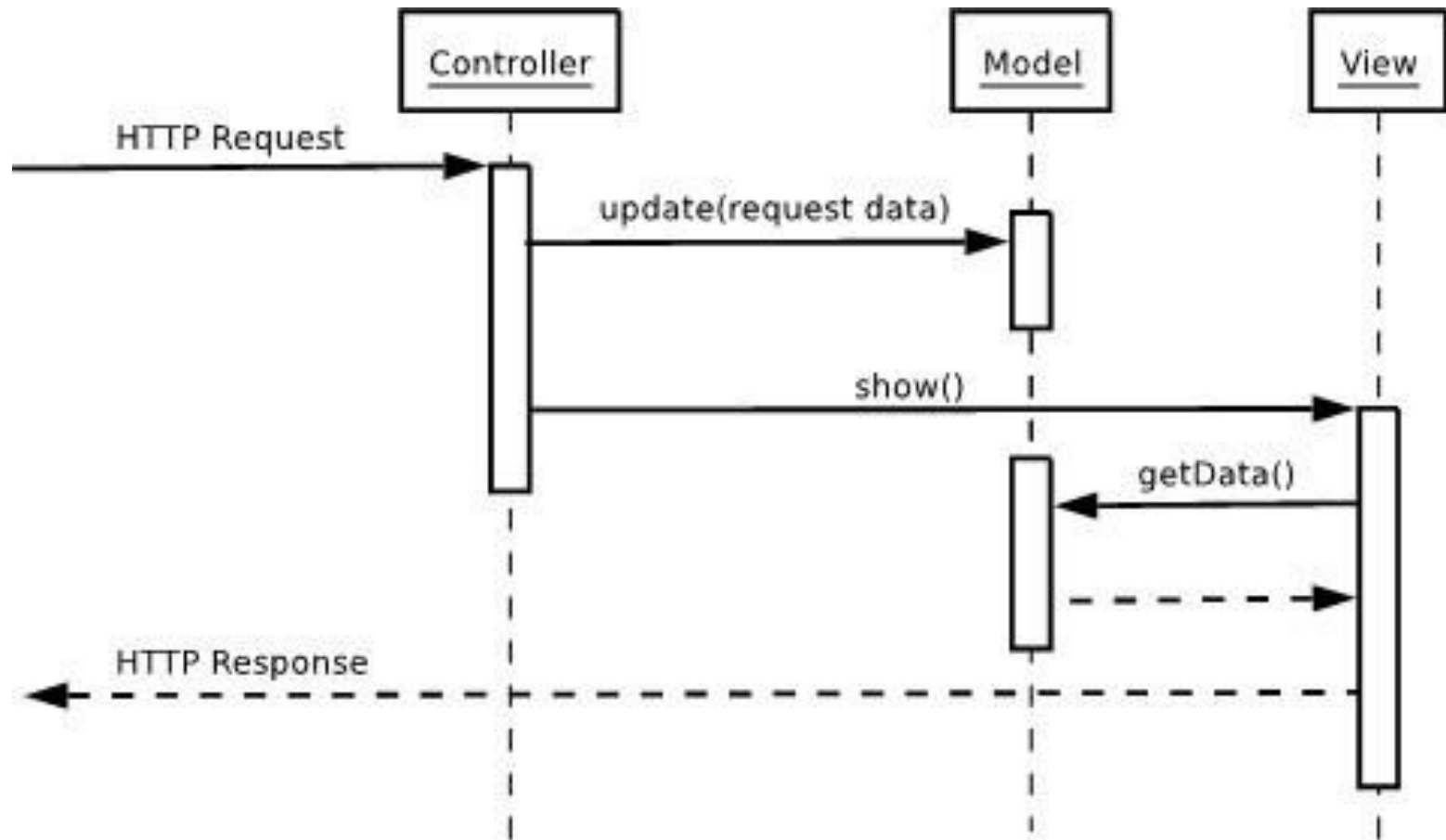
- Model:
  - Quản lý dữ liệu, thao tác trên dữ liệu, các quy luật, quy tắc trong lĩnh vực của ứng dụng (application domain)
  - Fat model: đa phần các xử lý dữ liệu nằm trong phần model
- View:
  - Trình bày, hiển thị dữ liệu, giao tiếp với người sử dụng
- Controller:
  - Điều khiển dữ liệu di chuyển giữa view và control
  - Là thành phần trung gian, liên kết giữa View và Model

# Tại sao cần MVC?

---

- Dựa trên sự **tách biệt trách nhiệm** (separation of concerns)
- Mỗi thành phần có một trách nhiệm riêng:
  - **SRP**: Single Responsibility Principle
  - **DRY**: Don't Repeat Yourself
- Dễ dàng kiểm thử, bảo trì hơn
- Những thay đổi ở một thành phần **không ảnh hưởng** đến các thành phần khác
- Các thành phần có thể được phát triển **song song**

# Luồng thực thi trong MVC

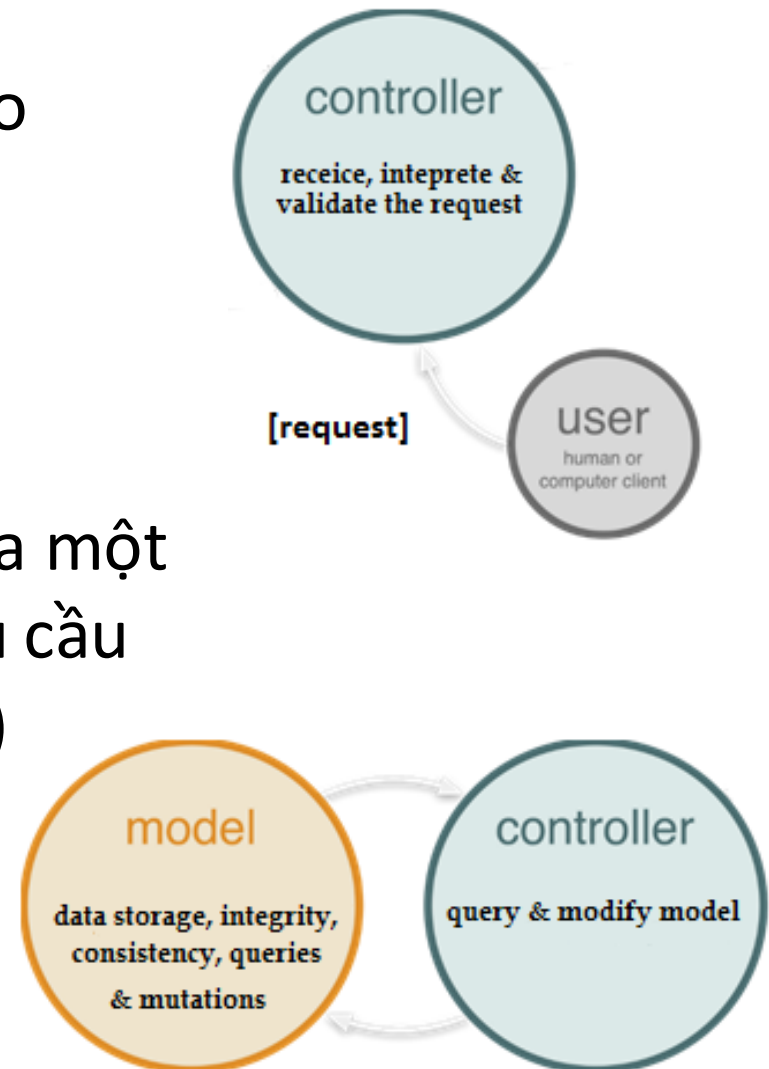




# Luồng thực thi trong MVC

1) Yêu cầu được chuyển đến cho **Controller**

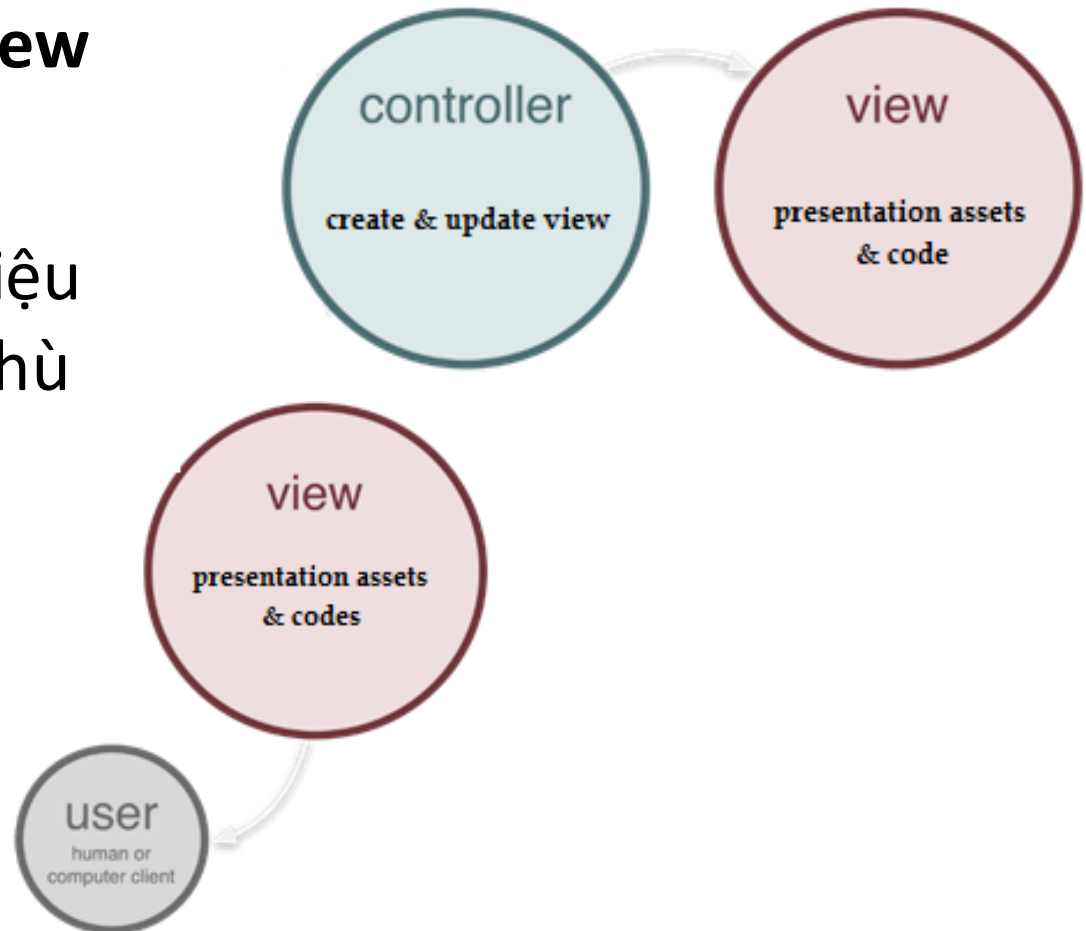
2) **Controller** nhờ sự giúp đỡ của một **Model** thích hợp để xử lý yêu cầu (truy vấn, cập nhật dữ liệu,...)



# Luồng thực thi trong MVC

3) Dữ liệu từ **Model** được **Controller** chuyển đến cho **View**

4) **View** biến đổi dữ liệu thành định dạng phù hợp và trình bày



# Front Controller

---

- Trong thực tế, thành phần Controller có thể bổ sung 1 **Front Controller**:
  - Là điểm đầu vào tập trung cho việc xử lý các yêu cầu đến ứng dụng web (thường là tập tin **index.php**)
- Chức năng của Front Controller:
  - Nạp các phụ thuộc
  - Xử lý yêu cầu
  - Gửi trả lời cho trình duyệt web

# Xây dựng ứng dụng Web với MVC

*Build MVC Web Applications*

# Xây dựng ứng dụng web với MVC

---

- Sử dụng cơ chế “**rewrite**” của web server để chuyển tất cả các yêu cầu HTTP đến **Front Controller** (index.php)
- Front Controller sử dụng 1 **Router** để thực hiện **định tuyến** yêu cầu vào **đoạn mã xử lý** tương ứng
  - Router phân tích yêu cầu đến (HTTP method + URL) để xác định đoạn mã xử lý tương ứng
- Xây dựng ỨD web MVC: định nghĩa tập **rule** cho Router
  - **Rule** = HTTP + URL pattern + Handle
  - URL trong các rule của Router là các URL luận lý (không tương ứng với 1 tài nguyên vật lý nào)

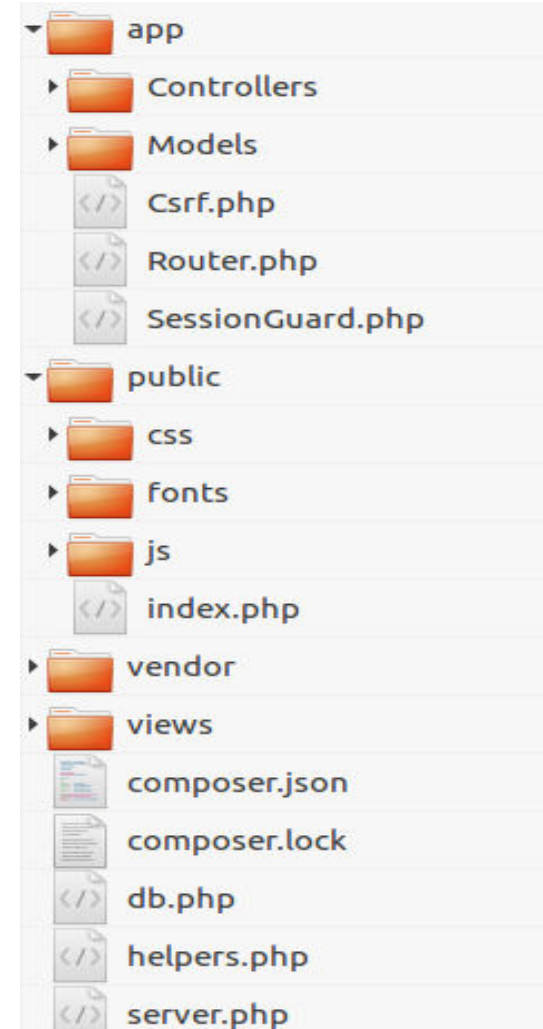
# Tổ chức thư mục dự án

## URL Rewrite trên Apache

```
<VirtualHost *:8080>
    DocumentRoot "C:/xampp/apps/phonebook/public"
    ServerName localhost

    #Set access permission
    <Directory "C:/xampp/apps/phonebook/public">
        AllowOverride None
        Require all granted

        RewriteEngine On
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule ^.*$ index.php [L]
    </Directory>
</VirtualHost>
```



# Các router phổ biến

---

- slim/slim
- nikic/fast-route
- klein/klein
- altorouter/altorouter
- aura/router
- noahbuscher/macaw

# noahbuscher/macaw

---

```
// public/index.php
```

```
<?php
require 'vendor/autoload.php';
use \NoahBuscher\Macaw\Macaw;

Macaw::get('/',
    'Controllers\Demo@index');
Macaw::get('page',
    'Controllers\Demo@page');
Macaw::get('view/(:num)',
    'Controllers\Demo@view');

Macaw::dispatch();
?>
```

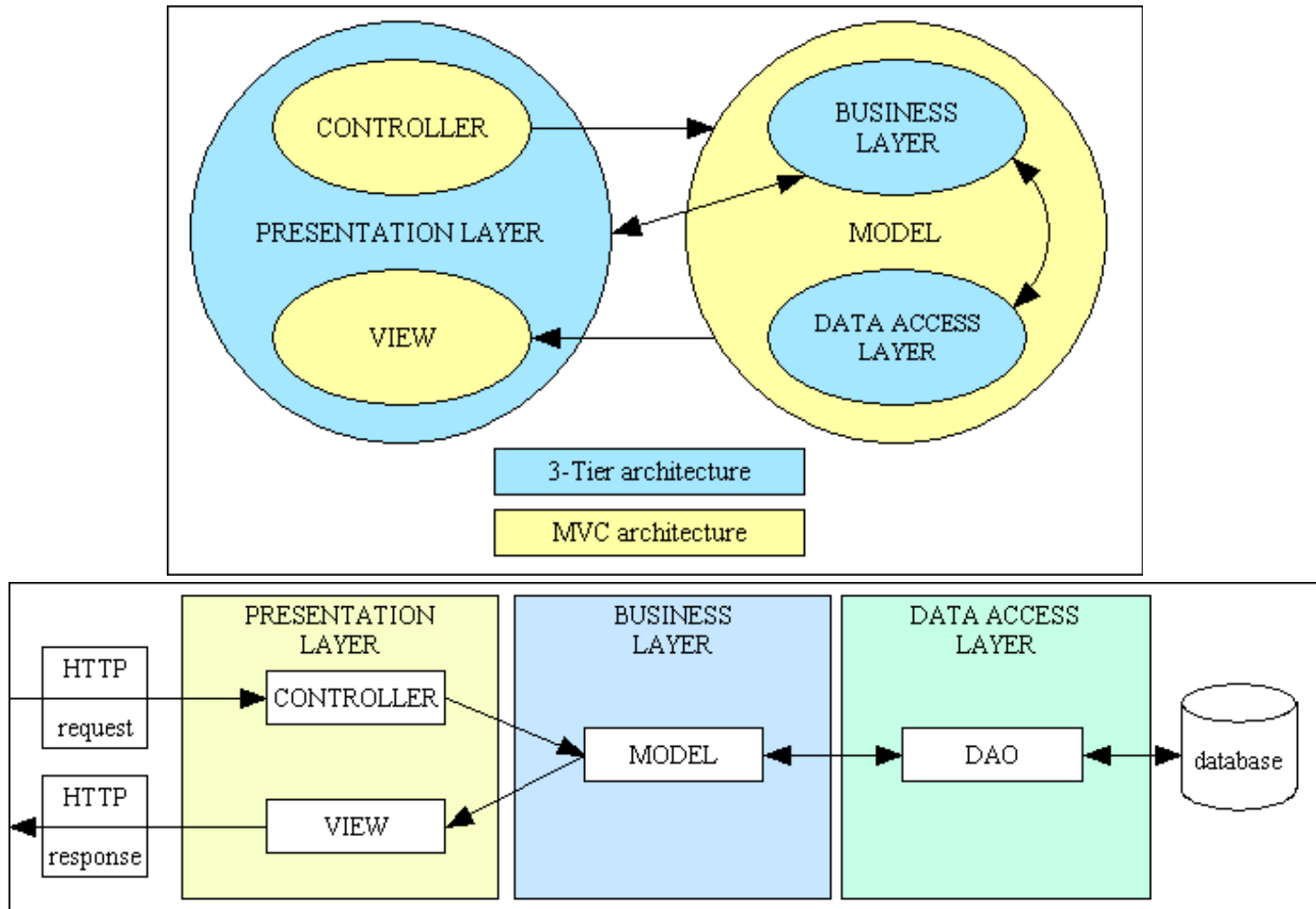
```
// app/Controllers/Demo.php
```

```
<?php
namespace Controllers;

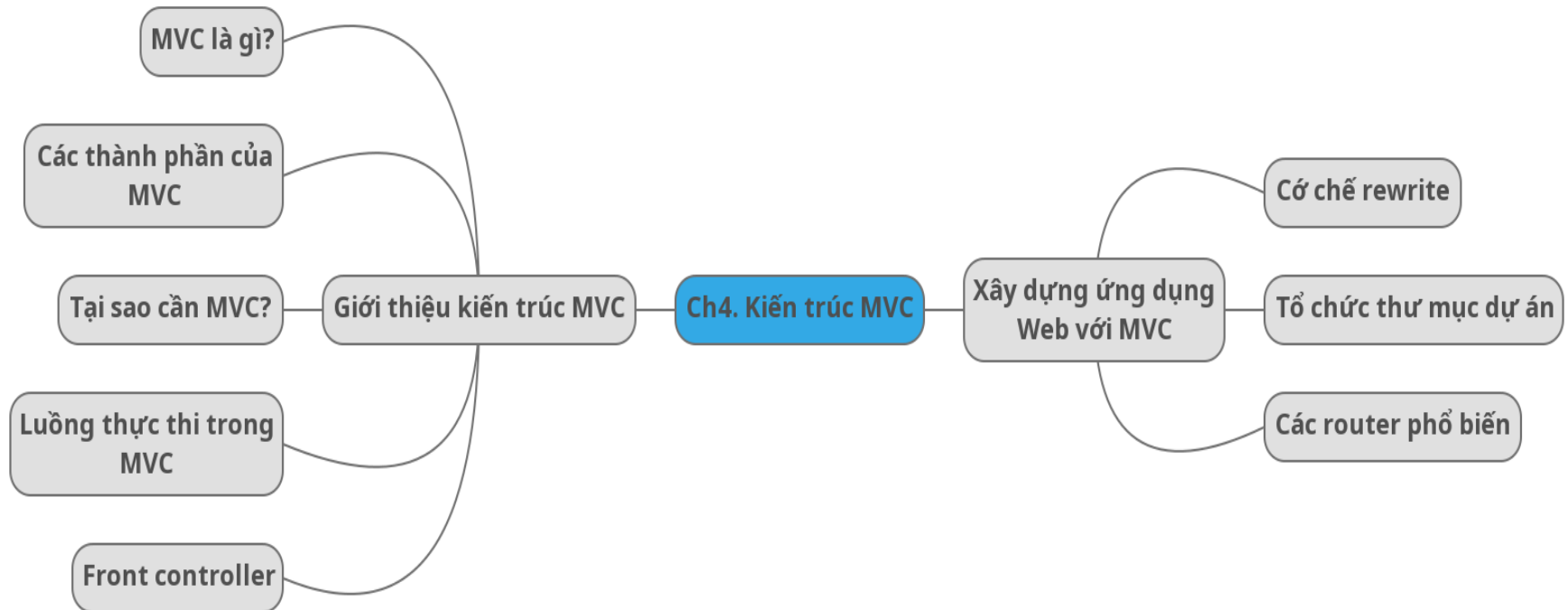
class Demo {
    public function index() {
        echo 'home';
    }
    public function page() {
        echo 'page';
    }
    public function view($id) {
        echo $id;
    }
}
?>
```



# MVC và kiến trúc 3-tầng



# Tóm tắt





Question?

CT275 – CÔNG NGHỆ WEB