Bài 21
MVC Model

Module: **ADVANCED PROGRAMMING WITH PHP 2.0**

Mục tiêu

- Trình bày được kiến trúc phân tầng
- Trình bày được kiến trúc MVC
- Trình bày được các lợi ích của kiến trúc phân tầng và kiến trúc MVC
- Triển khai được các ứng dụng theo kiến trúc MVC

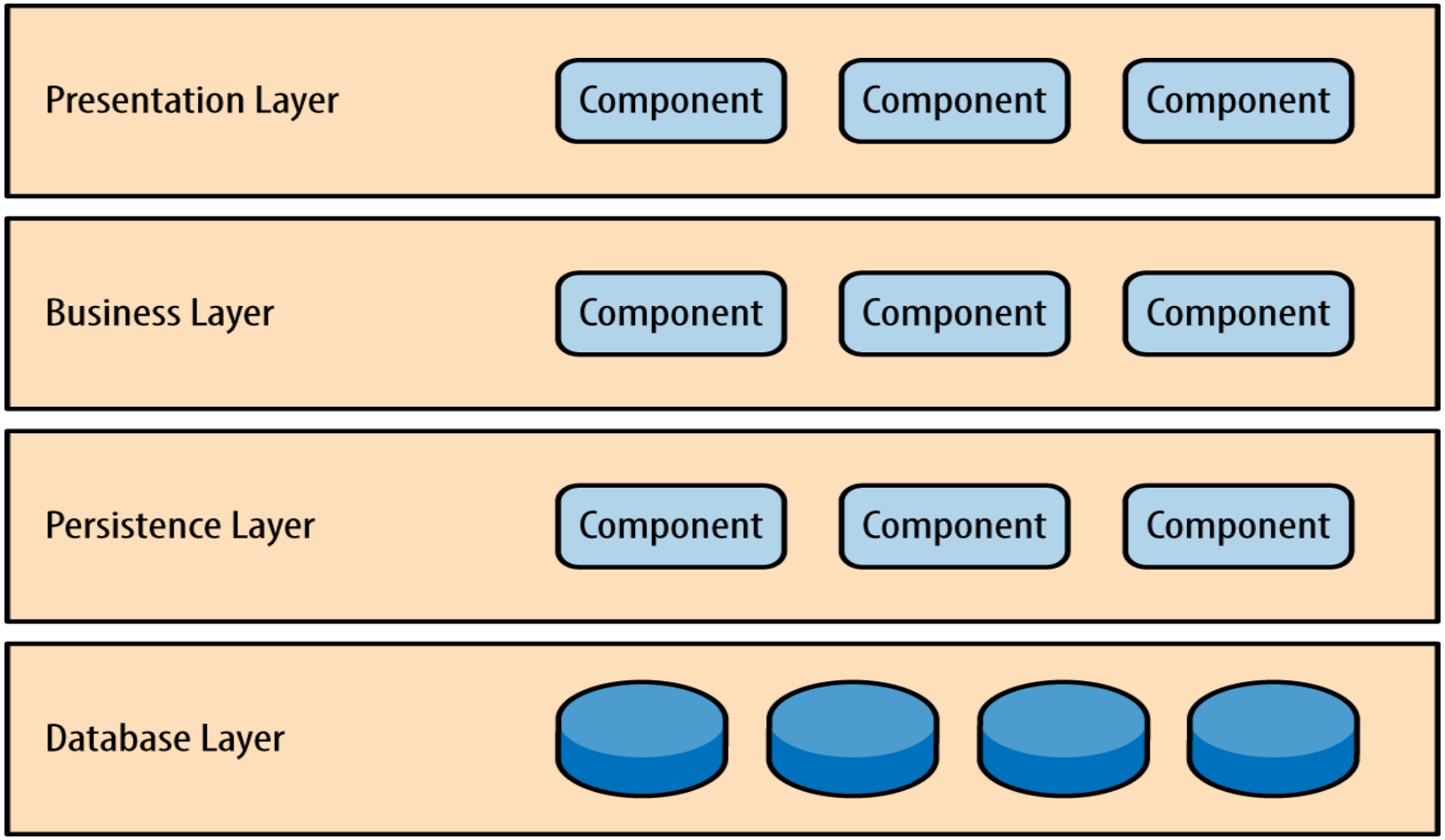
Thảo luận

Kiến trúc MVC

Kiến trúc phân tầng

- Kiến trúc phân tầng (multilayered/multitier/n-tier/architecture) là một kiến trúc phần mềm trong đó các thành phần của hệ thống được tổ chức theo các tầng theo chiều ngang, mỗi tầng thực hiện một nhóm nhiệm vụ cụ thể
- Chẳng hạn: Tầng giao diện, tầng điều khiển, tầng dịch vụ, tầng lưu trữ dữ liệu…
- Dạng kiến trúc n-tier phổ biến là 3 tầng (three-tier)

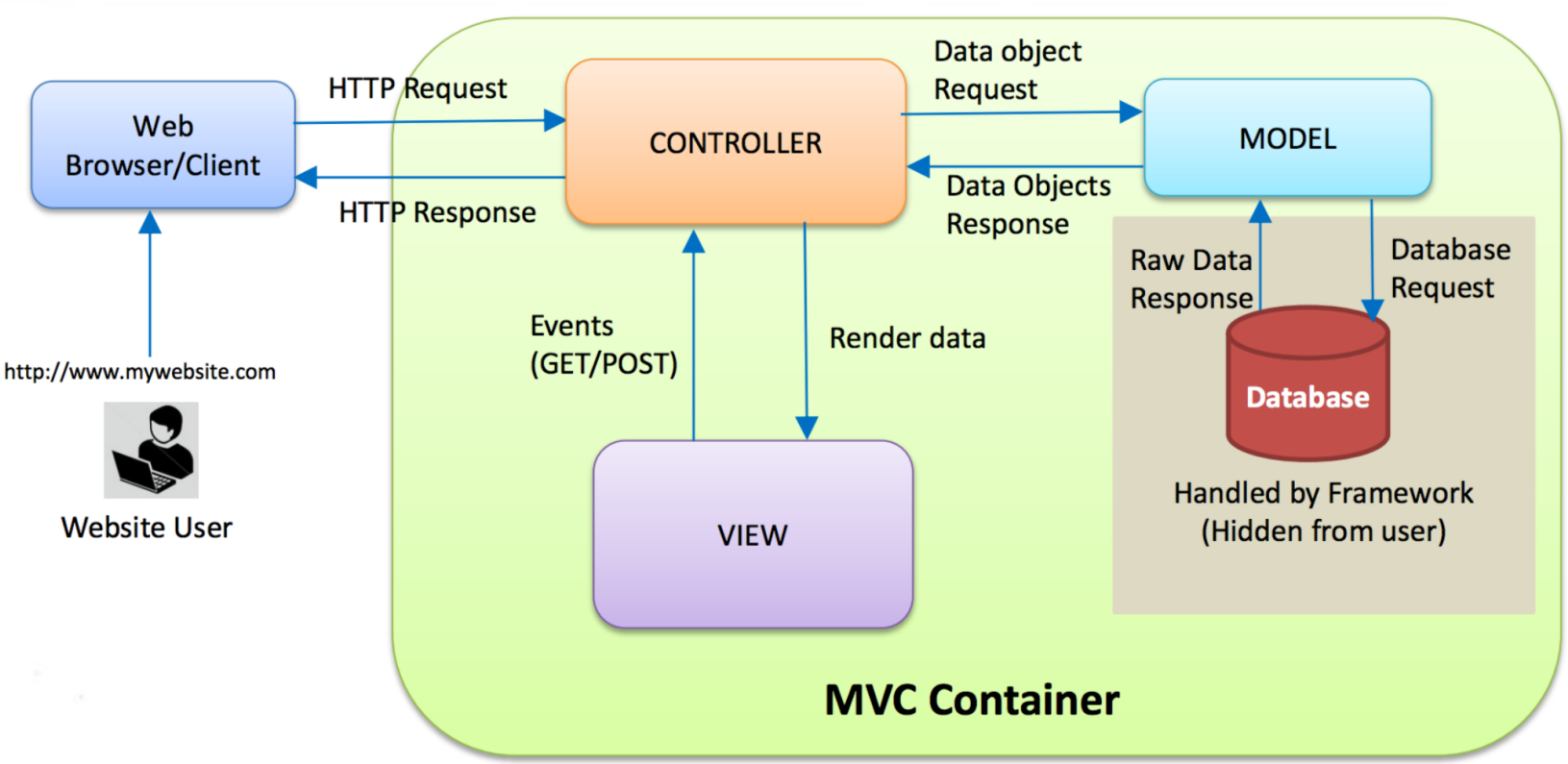
Kiến trúc phân tầng: Ví dụ



Kiến trúc MVC

- MVC là viết tắt của Model-View-Controller
- Kiến trúc MVC tổ chức các thành phần của hệ thống vào 3 tầng riêng biệt và có kết nối đến nhau:
  - Tầng Model: Biểu diễn dữ liệu và các logic nghiệp vụ
  - Tầng View: Hiển thị dữ liệu và là giao diện tương tác với người dùng
  - Tầng Controller: Xử lý các thao tác từ người dùng, kết nối giữa Model và View

MVC là gì


Web Browser/Client
HTTP Request
HTTP Response
CONTROLLER
Data object Request
Data Objects Response
MODEL
Raw Data Response
Database Request
Database
Handled by Framework (Hidden from user)
http://www.mywebsite.com
Website User
Events (GET/POST)
Render data
VIEW
MVC Container

Lợi ích của kiến trúc MVC

- Dễ tái sử dụng
- Dễ bảo trì và mở rộng
- Tách phần view và phần nghiệp vụ riêng biệt
- Cho phép các Lập trình viên làm việc trên các thành phần khác nhau trong cùng một thời điểm

DEMO MVC

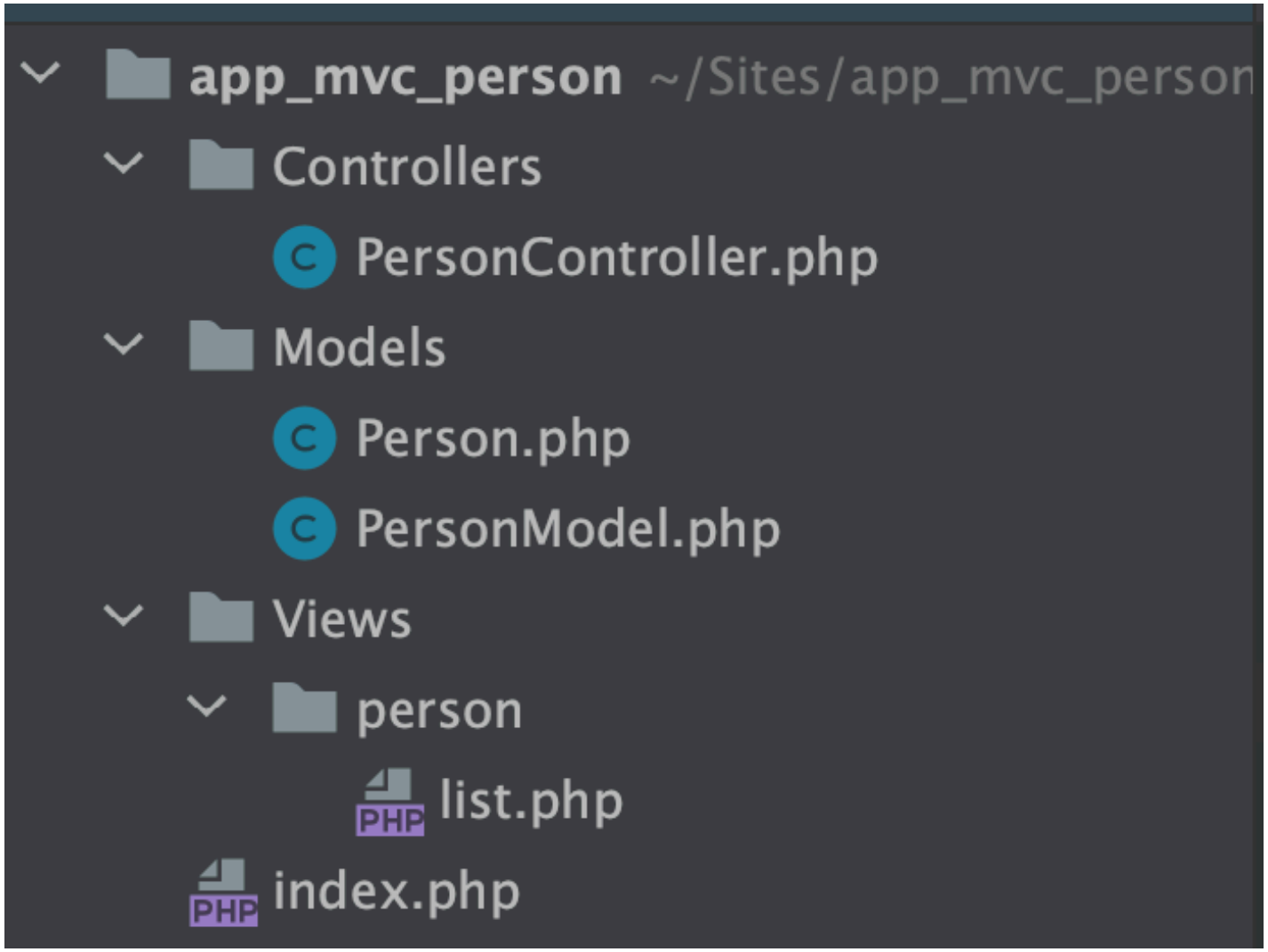
Ví dụ demo Model-View-Controller

Bài toán quản lý danh sách người dùng

Person Manager

#	Name	Email	Phone
1	Nguyen Van Teo	teo@gmail.com	149289829
2	Nguyen Van Tun	tun@gmail.com	87289829
3	Pham Van Nam	nam@gmail.com	87289829

Cấu trúc project



Tạo file Models/Person.php

```php
<?php


class Person
{
    //khai bao property
    protected string $name;
    protected string $email;
    protected int $phone;

    public function __construct(string $name,
                                string $email,
                                int $phone)
    {
        $this->name = $name;
        $this->email = $email;
        $this->phone = $phone;
    }

    /**
     * @return string
     */
    public function getName(): string
    {
        return $this->name;
    }

    /**
     * @return string
     */
```

Tạo file Models/PersonModel.php

```php
<?php


class PersonModel
{
    function getList(): array
    {
        $person1 = new Person( name: 'Nguyen Van Teo', email: 'teo@gmail.com', phone: 149289829);
        $person2 = new Person( name: 'Nguyen Van Tun', email: 'tun@gmail.com', phone: 87289829);
        $person3 = new Person( name: 'Pham Van Nam', email: 'nam@gmail.com', phone: 87289829);
        return [$person1, $person2, $person3];
    }
}
```

Tạo View/person/list.php

```
<table class="table">
    <thead class="thead-light">
    <tr>
        <th scope="col">#</th>
        <th scope="col">Name</th>
        <th scope="col">Email</th>
        <th scope="col">Phone</th>
    </tr>
    </thead>
    <tbody>
    <?php foreach ($persons as $key => $person): ?>
        <tr>
            <th scope="row"><?php echo $key + 1 ?></th>
            <td><?php echo $person->getName() ?></td>
            <td><?php echo $person->getEmail() ?></td>
            <td><?php echo $person->getPhone() ?></td>
        </tr>
    <?php endforeach; ?>
    </tbody>
</table>
```

Tạo Controllers/PersonController.php

```php
<?php


use JetBrains\PhpStorm\Pure;

class PersonController
{
    protected PersonModel $personModel;

    #[Pure] public function __construct()
    {
        $this->personModel = new PersonModel();
    }

    function index() {
        $persons = $this->personModel->getList();
        include_once "Views/person/list.php";
    }
}
```

Tạo trang index.php

```
<div class="container">
    <div class="card mt-5">
        <div class="card-header">
            Person Manager
        </div>
        <div class="card-body">
            <?php
            $personController->index();
            ?>
        </div>
    </div>

</div>
```

Tóm tắt bài học

- Mô hình MVC là một trong những mô hình quan trọng khi triển khai xây dựng hệ thống:
  - Controller tiếp nhận request và trả về response
  - Model thao tác với Database
  - View hiển thị giao diện cho người dùng
- Sử dung MVC giúp dễ tái sử dụng code, dễ bảo trì, mở rộng…

Hướng dẫn

- Hướng dẫn làm bài thực hành và bài tập
- Chuẩn bị bài tiếp: **Cascading Style Sheet**