



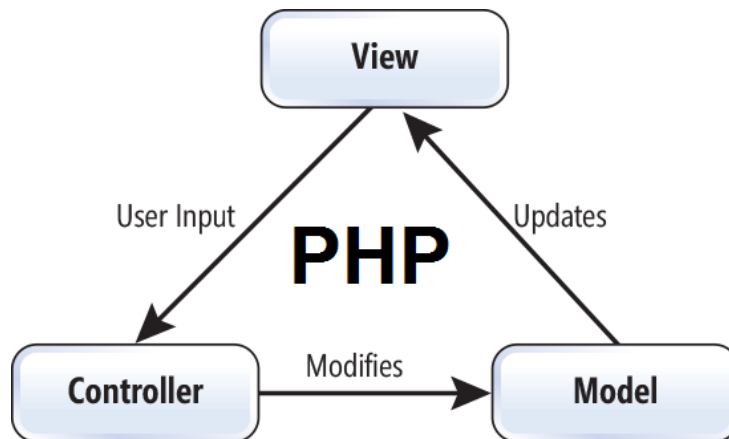
Kiến trúc Model-View-Controller (MVC) với PHP

Model-View-Controller (MVC) là một mẫu kiến trúc phần mềm được sử dụng rộng rãi trong phát triển web bằng **PHP** nhằm mục đích **tách biệt các thành phần** của ứng dụng thành ba thành phần riêng biệt, có tính tương tác cao. Sự phân tách này giúp mã nguồn **có tổ chức, dễ tái sử dụng, bảo trì** và mở rộng hơn.



Các Thành phần và Luồng hoạt động của MVC trong PHP

Ba thành phần cốt lõi của MVC, mỗi thành phần có một trách nhiệm cụ thể:



1. Model (Dữ liệu và Logic nghiệp vụ)

Model xử lý **dữ liệu và logic nghiệp vụ** của ứng dụng.

- **Trách nhiệm:** Tương tác với cơ sở dữ liệu (đọc, ghi, cập nhật), xác thực dữ liệu và thực thi các quy tắc nghiệp vụ.
- **Nguyên tắc chính:** Nó hoàn toàn **tách rời** khỏi lớp trình bày (View) và xử lý yêu cầu (Controller). Nó chỉ làm việc với dữ liệu và logic.
- **Trong PHP:** Thường được triển khai dưới dạng các lớp đại diện cho các thực thể dữ liệu (ví dụ: UserModel, ProductModel) và chứa các phương thức cho các thao tác CRUD (Create, Read, Update, Delete).

2. View (Giao diện người dùng)

View chịu trách nhiệm về **giao diện người dùng và hiển thị dữ liệu**.

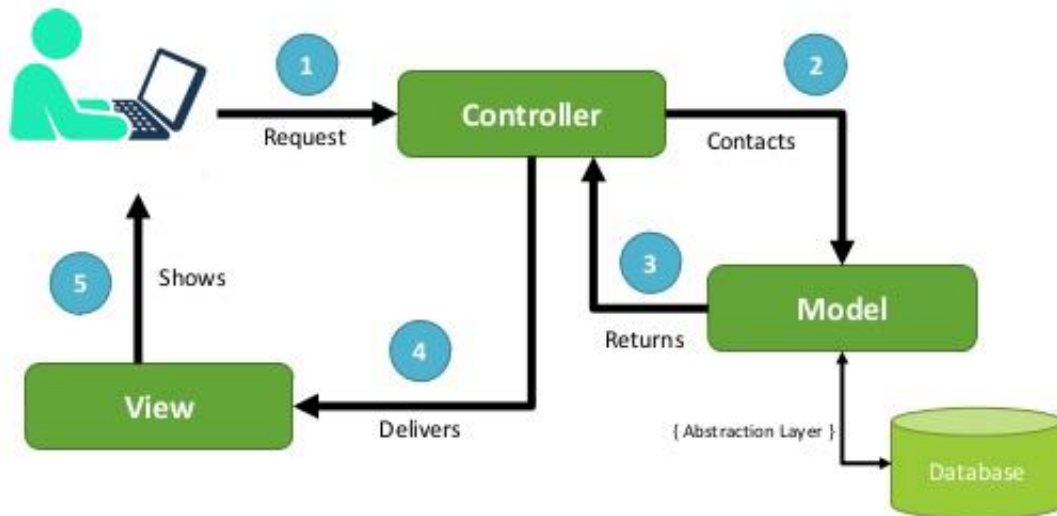
- **Trách nhiệm:** Tạo ra đầu ra mà người dùng nhìn thấy, thường là **HTML, CSS và JavaScript**. Nó hiển thị dữ liệu do Controller cung cấp.
- **Nguyên tắc chính:** Nó nên chứa **logic tối thiểu**—chủ yếu là logic trình bày (các vòng lặp để hiển thị danh sách, hiển thị có điều kiện các phần tử). Nó không truy cập trực tiếp vào cơ sở dữ liệu hoặc chứa các quy tắc nghiệp vụ.
- **Trong PHP:** Thường là sự kết hợp giữa HTML và mã PHP nhúng đơn giản (hoặc một công cụ template như Blade hoặc Twig) để lập và hiển thị dữ liệu được truyền tới.

3. Controller (Bộ điều phối)

Controller hoạt động như **người trung gian giữa Model và View**.

- **Trách nhiệm:** Nhận đầu vào từ người dùng (thông qua các yêu cầu HTTP), xử lý, quyết định Model nào cần tương tác và sau đó chọn View thích hợp để hiển thị kết quả.
- **Nguyên tắc chính:** Nó phải **nhẹ nhàng**, ủy thác logic nghiệp vụ cho Model và chi tiết trình bày cho View.
- **Trong PHP:** Một lớp (ví dụ: UserController) với các phương thức (hành động, ví dụ: indexAction(), saveAction()) ánh xạ tới các yêu cầu của người dùng và điều phối luồng.

MVC Design Pattern



B1:

- ✓ **Tương tác người dùng:** Người dùng tương tác với ứng dụng (ví dụ: nhấp vào liên kết, gửi biểu mẫu), gửi một yêu cầu (thường thông qua một Router để định tuyến yêu cầu) đến điểm vào của ứng dụng.
- ✓ **Controller Nhận Yêu cầu:** Phương thức Controller tương ứng với URL yêu cầu được thực thi.

B2:

Controller Tương tác với Model: Controller xác định dữ liệu cần thiết và gọi (các) phương thức Model thích hợp để lấy, tạo hoặc cập nhật dữ liệu.

B3:

Model Xử lý Dữ liệu: Model thực hiện các thao tác cần thiết (ví dụ: truy vấn cơ sở dữ liệu) trả kết quả về cho Controller

B4:

Controller Chọn View: Controller lấy dữ liệu được trả về từ Model và chuyển nó cho View đã chọn.

B5:

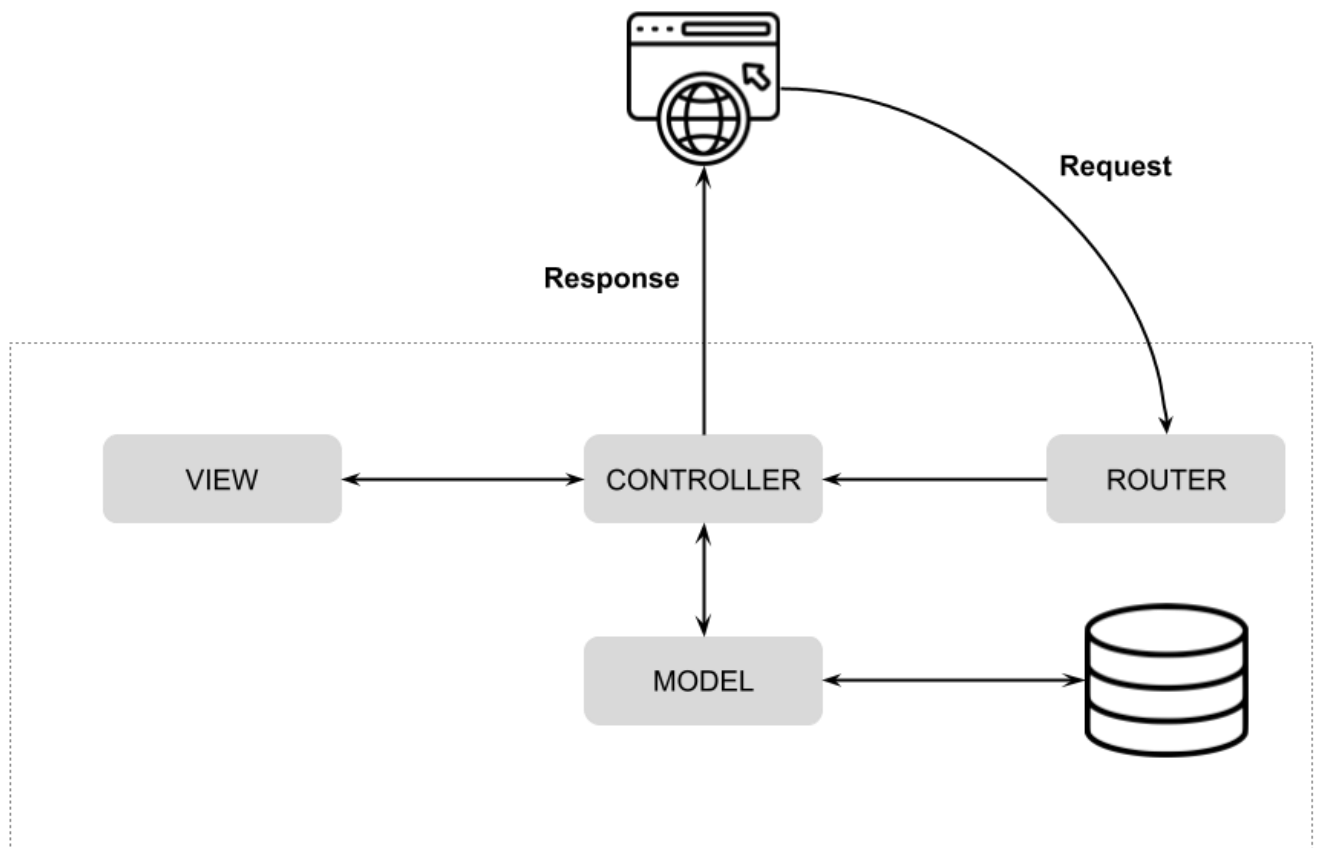
- ✓ **View Hiện thị Đầu ra:** View sử dụng dữ liệu để tạo ra đầu ra HTML cuối cùng.
- ✓ **Phản hồi đến người dùng:** HTML được tạo ra được gửi lại cho trình duyệt của người dùng.

Các Framework PHP sử dụng MVC

Mặc dù bạn có thể xây dựng một ứng dụng MVC thô bằng PHP thuần, hầu hết quá trình phát triển PHP hiện đại đều sử dụng các framework để thực thi mẫu này, cung cấp cấu trúc, công cụ và các phương pháp hay nhất:

- **Laravel:** Framework PHP phổ biến nhất, là một framework MVC full-stack.
- **Symfony:** Một framework trưởng thành, mạnh mẽ và có khả năng tùy biến cao, thường được sử dụng cho các ứng dụng cấp doanh nghiệp phức tạp. Nó dựa trên các thành phần nhưng tuân theo các nguyên tắc MVC.
- **CodeIgniter:** Một framework nhẹ được biết đến với thiết lập đơn giản rất phù hợp cho người mới bắt đầu và các dự án vừa và nhỏ.
- **Yii:** Một framework PHP dựa trên thành phần, hiệu suất cao, phù hợp để phát triển các ứng dụng web quy mô lớn.

Các bước triển khai MVC dùng PHP thuần: Xây dựng chức năng CRUD (Create Read Update Delete) cho bảng Nhaxb



1) Thiết lập Thư mục theo cấu trúc sau:

```
/bookstore-mvc
├── /app/
│   ├── /database/ # Lớp truy xuất cơ sở dữ liệu
│   ├── /controllers/ # Xử lý logic và nhận yêu cầu
│   ├── /models/ # Xử lý dữ liệu và logic nghiệp vụ
│   └── /views/ # Hiển thị HTML cho người dùng
├── /config/ # Các tập tin cấu hình cơ sở dữ liệu, ứng dụng
├── /public/ # Điểm truy cập duy nhất (Front Controller)
└── index.php # Tập tin chính, nơi mọi yêu cầu đi qua
```

2) Cấu hình ứng dụng:

- Tạo tập tin **.htaccess** trong thư mục gốc để **chuyển hướng** (rewrite) tất cả các yêu cầu đến public/index.php (trừ các tập tin tĩnh).

APACHE

Thư mục gốc: /bookstore-mvc/.htaccess

```
<IfModule mod_rewrite.c>
```

```
    RewriteEngine On
```

```
    RewriteCond %{REQUEST_FILENAME} !-d
```

```
    RewriteCond %{REQUEST_FILENAME} !-f
```

```
    RewriteRule ^(.*)$ public/index.php/$1 [L]
```

```
</IfModule>
```

Giải thích: Các dòng này đảm bảo rằng trừ khi URL yêu cầu là một thư mục (-d) hoặc một tập tin (-f) thực sự tồn tại, mọi yêu cầu khác sẽ được chuyển hướng (Rewrite) đến public/index.php.

- Tạo tập tin **config/database.php**

PHP

```
<?php
```

```
// /config/database.php
```

```
define('DB_HOST', 'localhost');
```

```
define('DB_NAME', 'bookstore_vn');
```

```
define('DB_USER', 'root');
```

```
define('DB_PASS', '');
```

- Tạo tập tin **config/autoload.php**

PHP

```
<?php
```

```
spl_autoload_register(function ($classname) {
```

```
    //danh sach thu muc
```

```
    $classmap = [
```

```
        "database",
```

```
        "models"
```

```
    ];
```

```
    foreach ($classmap as $dir)
```

```
        if (file_exists(APP . '/' . $dir . '/' . $classname . '.class.php')) {
```

```
            require_once APP . '/' . $dir . '/' . $classname . '.class.php';
```

```
            return;
```

```
        }
```

```
});
```

3) Tạo Front Controller (public/index.php) có nhiệm vụ sau:

- Làm điểm truy cập duy nhất.
- Khởi tạo cấu hình.
- Thiết lập autoloading.
- Hiện thực hoặc gọi Router để xử lý URL.

PHP

```
<?php
// Define constants for path
define('ROOT', dirname(__DIR__));
define('APP', ROOT . '/app');

//load config;
require_once ROOT . '/config/database.php';

//auto class loading
require ROOT . '/config/autoload.php';

// A simple router to determine the controller and action
$url = $_GET['url'] ?? 'nhaxb/index'; // Default route
$url = explode('/', filter_var(rtrim($url, '/'), FILTER_SANITIZE_URL));

$controllerName = ucfirst($url[0]) . 'Controller'; //HomeController
$action = $url[1] ?? 'index';
$controllerPath = APP . '/controllers/' . $controllerName . '.class.php';

if (file_exists($controllerPath)) {
    require_once $controllerPath;
    $controller = new $controllerName();
    if (method_exists($controller, $action)) {
        $controller->$action();
    } else {
        // Handle 404 action not found
        echo "Action not found";
    }
} else {
    // Handle 404 controller not found
    echo "Controller not found";
}
```

Bảng định tuyến

URL Người dùng	Controller được gọi	Phương thức (Action)
index.php?url=nhaxb/index	NhaxbController	index()
index.php?url=nhaxb/edit	NhaxbController	edit()
index.php?url=nhaxb/del	NhaxbController	del()
index.php?url=nhaxb/upd	NhaxbController	upd()
index.php?url=nhaxb/add	NhaxbController	add()

4) Xây dựng các lớp **Controller, Model, và View**

- Tạo lớp **Db** (app/database/**Db.class.php**) kết nối CSDL

PHP

```
<?php
// app/database/Db.class.php
class Db {
    private $dbh; // Database Handler
    private $_numRow = 0;

    function __construct() {
        // Set DSN (Data Source Name)
        $dsn = 'mysql:host=' . DB_HOST . ';dbname=' . DB_NAME;
        // Create new PDO instance
        try {
            $this->dbh = new PDO($dsn, DB_USER, DB_PASS);
            $this->query(" set names 'utf8' ");
        } catch (PDOException $e) {
            echo $e->getMessage();
            exit;
        }
    }
    function getRowCount() {
        return $this->_numRow;
    }
    function __destruct() {
        $this->dbh = null;
    }
    // Prepare statement with query
    function query($sql, $params = array(), $mode = PDO::FETCH_ASSOC) {
        $stm = $this->dbh->prepare($sql);
        if (!$stm->execute($params)) {
            echo "Loi SQL!";
            exit;
        }
        $this->_numRow = $stm->rowCount();
        return $stm->fetchAll($mode);
    }
    // Execute the prepared statement
    public function execute($sql, $params = array()) {
        $this->query($sql, $params);
        return $this->getRowCount();
    }
}
```

Model (/app/models/)

Model là nơi chứa **logic nghiệp vụ** và tương tác với **cơ sở dữ liệu** (DB).

- Không tương tác trực tiếp với View.
- Trả dữ liệu thô (mảng hoặc đối tượng) cho Controller.

Tạo lớp Nhaxb (/app/models/Nhaxb.class.php)

PHP

<?php

```
class Nhaxb extends Db {
    function getAll() {
        return $this->query("select * from nhaxb");
    }
    function insert($manxb, $tennxb) {
        return $this->execute(
            "insert into nhaxb(manxb, tennxb) values(?,?)", [$manxb, $tennxb]);
    }
    function delete($manxb) {
        return $this->execute("delete from nhaxb where manxb=?", [$manxb]);
    }
    function getById($manxb) {
        return $this->query("select * from nhaxb where manxb=?", [$manxb]);
    }
    function update($manxb, $tennxb) {
        return $this->execute(
            "update nhaxb set tennxb=? where manxb=?", [$tennxb, $manxb]);
    }
}
```

Controller (/app/controllers/)

Controller nhận yêu cầu đã được Router định tuyến, thực hiện:

- **Gọi Model** để lấy dữ liệu.
- **Truyền dữ liệu** đó cho View.

Tạo lớp NhaxbController (/app/controllers/NhaxbController.class.php)

PHP

<?php

```
class NhaxbController {
    private $model;
    function __construct() {
        $this->model = new Nhaxb;
    }
    function add() {
        if (isset($_POST['manxb']) && isset($_POST['tennxb']))
            $this->model->insert($_POST['manxb'], $_POST['tennxb']);
        $this->redirect();
    }
    protected function render($view, $data=[]) {
        require_once APP . '/views/Nhaxb/' . $view . '.php';
    }
    function index() {
        $data = $this->model->getAll();
        $this->render('list', $data);
    }
}
```

```

function edit() {
    $data['action'] = "index.php?url=nhaxb/add";
    $data['isUpdate'] = false;
    if (isset($_GET['manxb'])) {
        $data['action'] = "index.php?url=nhaxb/upd";
        $data['isUpdate'] = true;
        $data['row'] = $this->model->getById($_GET['manxb'])[0];
    }
    $this->render('edit',$data);
}
function del() {
    if (isset($_GET['manxb']))
        $this->model->delete($_GET['manxb']);
    $this->redirect();
}
function upd() {
    if (isset($_POST['manxb'])&&isset($_POST['tennxb']))
        $this->model->update($_POST['manxb'], $_POST['tennxb']);
    $this->redirect();
}
private function redirect($url = 'index.php?url=nhaxb/index' ) {
    header('location:' . $url);
}
}

```

View (/app/views/)

View là tập tin **HTML** có nhiệm vụ hiển thị dữ liệu đã được Controller truyền cho.

- View chỉ chứa **logic trình bày** (vòng lặp foreach hoặc câu lệnh if đơn giản).
- View **không** được chứa logic truy vấn DB.

Tạo file **/app/views/Nhaxb/list.php**

HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>
<div>
    <a href="index.php?url=nhaxb/edit">Them nha xuat ban</a>
</div>&nbsp;
<table border="1" cellspacing="0">
    <tr>
        <th>Ma NXB</th>
        <th>Ten NXB</th>
        <th></th>
    </tr>
    <?php foreach ($data as $row): ?>
        <tr>
            <td><?php echo $row['manxb']; ?></td>

```



```

        <td><?php echo $row['tennxb']; ?></td>
        <td>
            <a href="index.php?url=nhaxb/del&manxb=<?php echo $row['manxb'];
?>">Delete</a> |
            <a href="index.php?url=nhaxb/edit&manxb=<?php echo $row['manxb'];
?>">Update</a>
        </td>
    </tr>
<?php endforeach; ?>
</table>
</body>
</html>

```

Tạo file `/app/views/Nhaxb/edit.php`

```

<!DOCTYPE html>
<html lang="en">

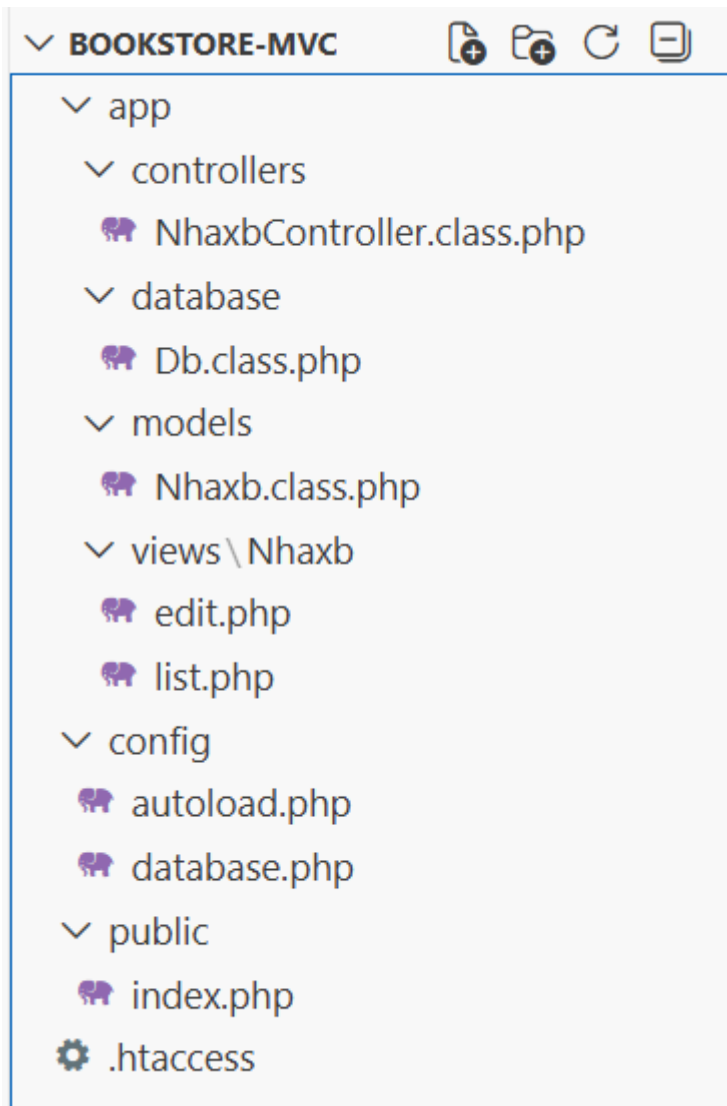
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <form action="<?php echo $data['action']; ?>" method="post">
        <table>
            <tr>
                <th colspan="2">THONG TIN NXB</th>
            </tr>
            <tr>
                <td>Ma NXB:</td>
                <td><input type="text" name="manxb" <?php echo ($data['isUpdate']) ?
"readonly" : "">; ?> value="<?php echo $data['row']['manxb'] ?? "">; ?>"></td>
            </tr>
            <tr>
                <td>Ten NXB:</td>
                <td><input type="text" name="tennxb" value="<?php echo
$data['row']['tennxb'] ?? "">; ?>"></td>
            </tr>
            <tr>
                <td></td>
                <td><input type="submit" value="Save"> <a
href="index.php?url=nhaxb/index">Cancel</a></td>
            </tr>
        </table>
    </form>
</body>

</html>

```

CẤU TRÚC THƯ MỤC HOÀN CHỈNH



| manxb | tennxb | SoLuongSach |
|-------|----------------------|-------------|
| gd | Giáo dục | 10 |
| hcm | Tổng Hợp Hồ Chí Minh | 9 |
| tn | Thanh Niên | 4 |
| vhth | Văn Hóa Thông Tin | 2 |

| maloai | tenloai | SoLuongSach |
|--------|-----------|-------------|
| gk | Giáo Khoa | 1 |
| td | Từ Điển | 6 |
| th | Tin Học | 18 |

Bài tập:

- 1) Kiểm tra trùng manxb trước khi thêm
- 2) Chỉ hiển thị link Delete khi nhà xuất bản không có quyển sách nào.
- 3) Cài đặt CRUD cho bảng Loai.
- 4) Tạo Trang chủ cho người dung chọn Quản lý Loai hay Quản lý Nhaxb