

**Thực hành Phát triển phần mềm mã nguồn mở**

Ấn bản 2024

# MỤC LỤC

MỤC LỤC .....	I
HƯỚNG DẪN.....	III
<b>BÀI 1. CHUẨN BỊ MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG VỚI PHP VÀ CÁC THAO TÁC CƠ BẢN</b> .....	<b>1</b>
1.1 CÀI ĐẶT VISUAL CODE STUDIO.....	2
1.2 CÀI ĐẶT LARAGON .....	5
1.3 XÂY DỰNG ỨNG DỤNG WEB PHP ĐƠN GIẢN THEO CẤU TRÚC MVC .....	7
1.4 XÂY DỰNG TRANG WEB BÁN HÀNG CÓ CÁC CHỨC NĂNG THÊM/ XÓA/ SỬA/ HIỂN THỊ SẢN PHẨM: .....	11
1.5 YÊU CẦU BỔ SUNG .....	20
<b>BÀI 2. TẠO CƠ SỞ DỮ LIỆU CHO WEBSITE BÁN HÀNG, XÂY DỰNG CHỨC NĂNG HIỂN THỊ/ THÊM/ XÓA/ SỬA</b> .....	<b>21</b>
2.1 SỬ DỤNG LARAGON ĐỂ QUẢN LÝ CƠ SỞ DỮ LIỆU MYSQL.....	22
2.2 XÂY DỰNG WEBSITE BÁN HÀNG BẰNG PHP KẾT NỐI VỚI CƠ SỞ DỮ LIỆU MYSQL .....	24
2.2.1 Tạo dự án mới trong Laragon:.....	24
2.2.2 Tạo cơ sở dữ liệu .....	25
2.2.3 Xây dựng các Model tương ứng.....	26
2.2.4 Khởi tạo các Controller tương ứng.....	28
2.2.5 Xây dựng giao diện hiển thị của trang web .....	32
2.2.6 Tiến hay khởi chạy dự án và thực nghiệm .....	36
2.3 YÊU CẦU THÊM: .....	39
<b>BÀI 3. XÂY DỰNG CHỨC NĂNG GIỎ HÀNG. ĐẶT HÀNG, THANH TOÁN.....</b>	<b>52</b>
3.1 TẠO BẢNG ORDERS VÀ ORDER_DETAILS .....	52
3.2 CẬP NHẬT PRODUCTCONTROLLER.....	53
3.3 TẠO CÁC VIEWS TƯƠNG ỨNG .....	60
3.4 KHỞI TẠO SESSION .....	62
3.5 TIẾN HÀNH KHỞI CHẠY VÀ KIỂM TRA KẾT QUẢ: .....	63
<b>BÀI 4. XÂY DỰNG CHỨC NĂNG XÁC THỰC NGƯỜI DÙNG .....</b>	<b>67</b>
4.1 CẤU HÌNH CƠ SỞ DỮ LIỆU .....	67
4.2 TẠO CÁC MÔ HÌNH (MODELS) .....	67
4.3 TẠO CÁC ĐIỀU KHIỂN (CONTROLLERS).....	68
4.4 TẠO CÁC TRANG HIỂN THỊ (VIEWS).....	71
4.5 TẠO FILE SESSIONHELPER.....	75
4.6 CẬP NHẬT HỆ THỐNG ĐỊNH TUYẾN (ROUTING).....	75
<b>BÀI 5. XÂY DỰNG RESTFUL API .....</b>	<b>79</b>
5.1 CẬP NHẬT PRODUCTMODEL .....	79
5.2 XÂY DỰNG CÁC CONTROLLER TƯƠNG ỨNG.....	81
5.3 CẤU HÌNH ROUTER ĐỂ ĐỊNH TUYẾN CÁC YÊU CẦU API .....	84
5.4 CẬP NHẬT VIEWS ĐỂ QUẢN LÝ SẢN PHẨM.....	86

5.5	TIẾN HÀNH KHỞI CHẠY DỰ ÁN VÀ SỬ DỤNG POSTMAN ĐỂ KIỂM THỬ API .....	92
5.6	YÊU CẦU BỔ SUNG.....	95
<b>BÀI 6.</b>	<b>BẢO MẬT RESTFUL API VỚI JWT .....</b>	<b>96</b>
6.1	CÀI ĐẶT THƯ VIỆN JWT .....	96
6.2	TẠO LỚP XỬ LÝ JWT .....	99
6.3	CẬP NHẬT API ĐỂ SỬ DỤNG JWT.....	100
6.4	CẬP NHẬT CÁC TRANG HIỂN THỊ .....	105
6.5	TIẾN HÀNH KHỞI CHẠY DỰ ÁN VÀ THỰC NGHIỆM.....	111
	<b>TÀI LIỆU THAM KHẢO.....</b>	<b>112</b>

# HƯỚNG DẪN

## MÔ TẢ MÔN HỌC

Môn học giúp sinh viên nắm vững các khái niệm cơ bản và nâng cao về PHP, một ngôn ngữ kịch bản mạnh mẽ phổ biến trong phát triển web. Sinh viên sẽ học cách áp dụng mô hình MVC để tổ chức mã nguồn hiệu quả và duy trì dễ dàng, kết hợp với MySQL để quản lý cơ sở dữ liệu, phát triển API để giao tiếp giữa các hệ thống, và bảo mật API bằng JWT. Qua các bài học thực hành, Sinh viên sẽ xây dựng được các ứng dụng web từ cơ bản tới nâng cao bằng ngôn ngữ lập trình PHP.

## NỘI DUNG MÔN HỌC

Bài 1. Chuẩn Bị Môi Trường Phát Triển Ứng Dụng Với PHP Và Các Thao Tác Cơ Bản.

Bài 2. Tạo Cơ Sở Dữ Liệu Cho Website Bán Hàng, Xây Dựng Chức Năng Hiển Thị Thêm/ Xóa/ Sửa.

Bài 3: Xây Dựng Chức Năng Giỏ Hàng, Đặt Hàng, Thanh Toán

Bài 4: Xây Dựng Chức Năng Xác Thực Người Dùng.

Bài 5: RESTful API.

Bài 6: Bảo Mật RESTful API Với JWT.

## KIẾN THỨC TIỀN ĐỀ

Trước khi bắt đầu môn học này, sinh viên cần có những kiến thức tiền đề sau để có thể theo kịp chương trình và hiểu sâu về các khái niệm được giảng dạy:

### 1. Kiến thức lập trình hướng đối tượng:

Hiểu biết vững chắc về OOP (Lập trình hướng đối tượng), xử lý lỗi và các khái niệm cơ bản.

### 2. Cơ bản về mô hình MVC (Model-View-Controller):

Hiểu rõ mô hình MVC và cách nó được áp dụng trong phát triển web.

### 3. Cơ sở dữ liệu và SQL:

Kiến thức cơ bản về cơ sở dữ liệu quan hệ và ngôn ngữ truy vấn SQL.

#### **4. Kiến thức cơ bản về HTML/CSS và JavaScript:**

Hiểu biết về cách xây dựng giao diện người dùng web với HTML/CSS. Cơ bản về ngôn ngữ lập trình JavaScript.

### **YÊU CẦU MÔN HỌC**

Người học phải tham dự đầy đủ các buổi lên lớp và làm bài tập đầy đủ.

### **CÁCH TIẾP NHẬN NỘI DUNG MÔN HỌC**

Để học tốt môn này, người học cần ôn tập các bài đã học, trả lời các câu hỏi và làm đầy đủ bài tập; đọc trước bài mới và tìm thêm các thông tin liên quan đến bài học.

Đối với mỗi bài học, người học đọc trước mục tiêu và tóm tắt bài học, sau đó đọc nội dung bài học. Kết thúc mỗi ý của bài học, người đọc trả lời câu hỏi ôn tập và kết thúc toàn bộ bài học, người đọc làm các bài tập.

### **PHƯƠNG PHÁP ĐÁNH GIÁ MÔN HỌC**

Môn học được đánh giá gồm:

- Điểm chuyên cần (30%): Hình thức và cách đánh giá do giảng viên dạy thực hành quyết định được phê duyệt của bộ môn.
- Điểm bài tập (70%): Hình thức làm bài tập trong các buổi học thực hành và giảng viên đánh giá chấm điểm. Danh sách bài tập thực hành được bộ môn kiểm duyệt và cung cấp vào đầu khóa học.

# BÀI 1. CHUẨN BỊ MÔI TRƯỜNG PHÁT TRIỂN ỨNG DỤNG VỚI PHP VÀ CÁC THAO TÁC CƠ BẢN

Sau khi học xong bài này, sinh viên có thể:

- Hiểu về Visual Studio Code (VSC): Có kiến thức cơ bản về cách cài đặt và sử dụng Visual Studio Code làm môi trường phát triển tích hợp (IDE) cho việc phát triển ứng dụng PHP.
- Cài đặt Laragon cho môi trường phát triển ứng dụng với PHP và MySQL
- Tạo dự án PHP theo mô hình MVC: Biết cách tạo một dự án PHP mới trong Visual Studio Code, tạo cấu trúc thư mục phù hợp và cấu hình các thư viện cần thiết cho dự án.

Với những kiến thức này, học viên đã chuẩn bị môi trường cho việc xây dựng website với PHP theo mô hình MVC và có nền tảng để tiếp tục các bước tiếp theo trong quá trình phát triển ứng dụng.

## 1.1 Cài đặt Visual Code Studio

Visual Studio Code là một trình soạn thảo mã nguồn mạnh mẽ và miễn phí được phát triển bởi Microsoft, phù hợp cho việc phát triển web với nhiều ngôn ngữ lập trình khác nhau, bao gồm PHP.

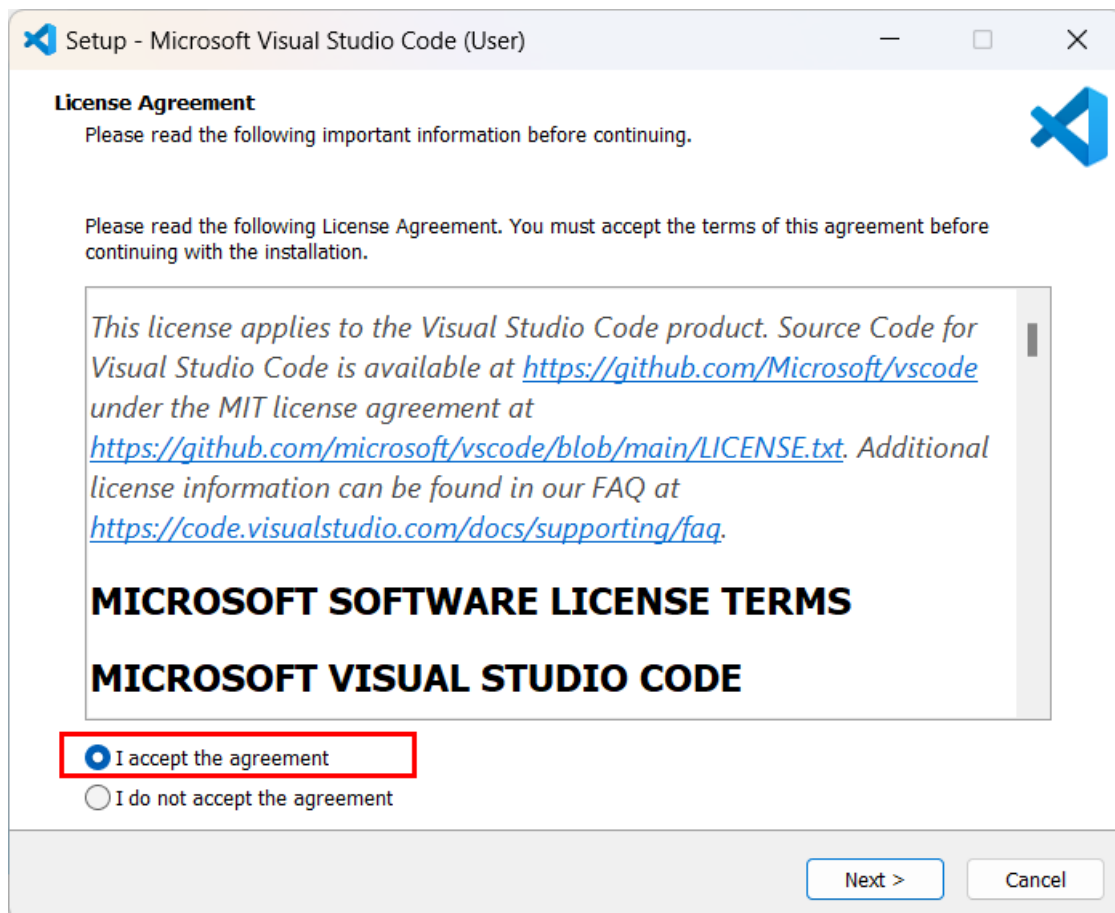
### Bước 1: Tải Visual Studio Code

1. Truy cập vào trang web chính thức của Visual Studio Code:  
<https://code.visualstudio.com/>.
2. Nhấn vào nút "Download" để tải về bản cài đặt phù hợp với hệ điều hành của bạn (Windows, macOS, hoặc Linux).

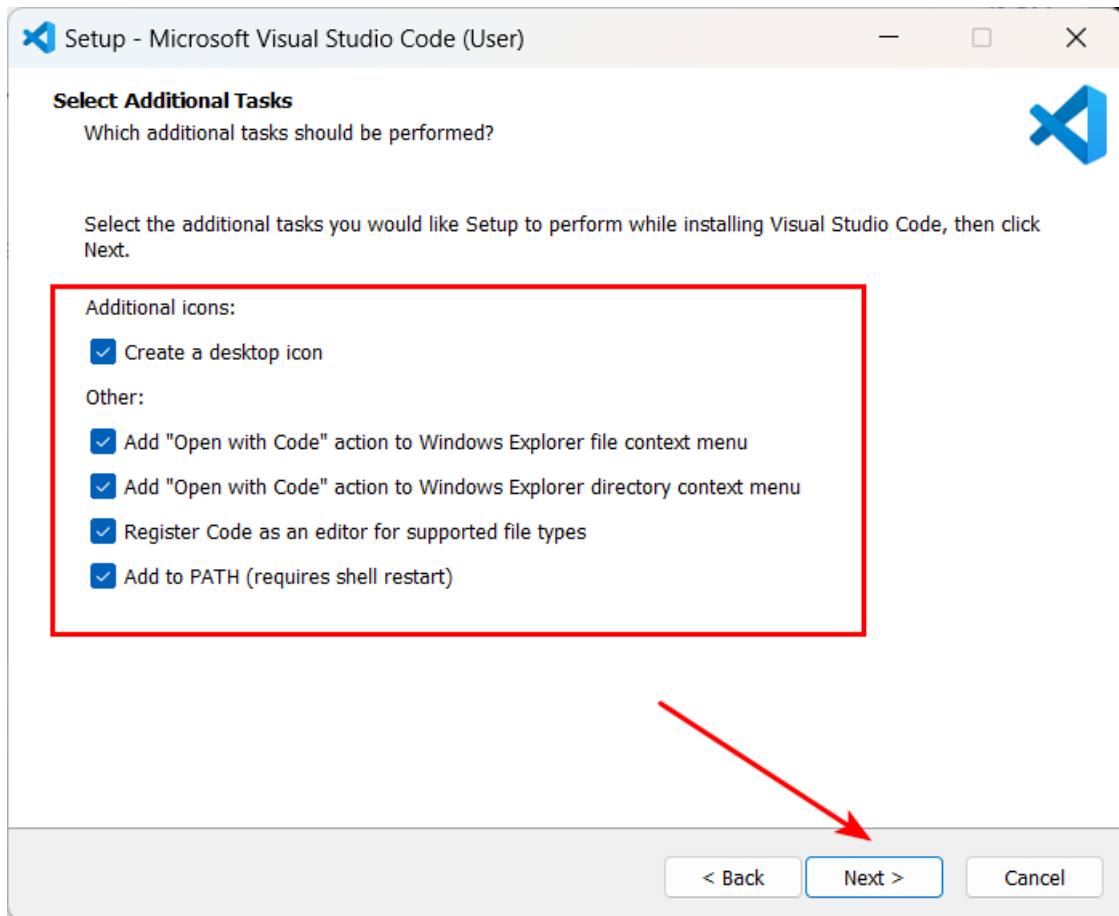
### Bước 2: Cài đặt Visual Studio Code trên Windows

Sau khi tải về file cài đặt, nhấn đúp vào file để bắt đầu quá trình cài đặt.

1. Trong cửa sổ cài đặt xuất hiện, chọn "I accept the agreement" để chấp nhận các điều khoản sử dụng và nhấn "Next".

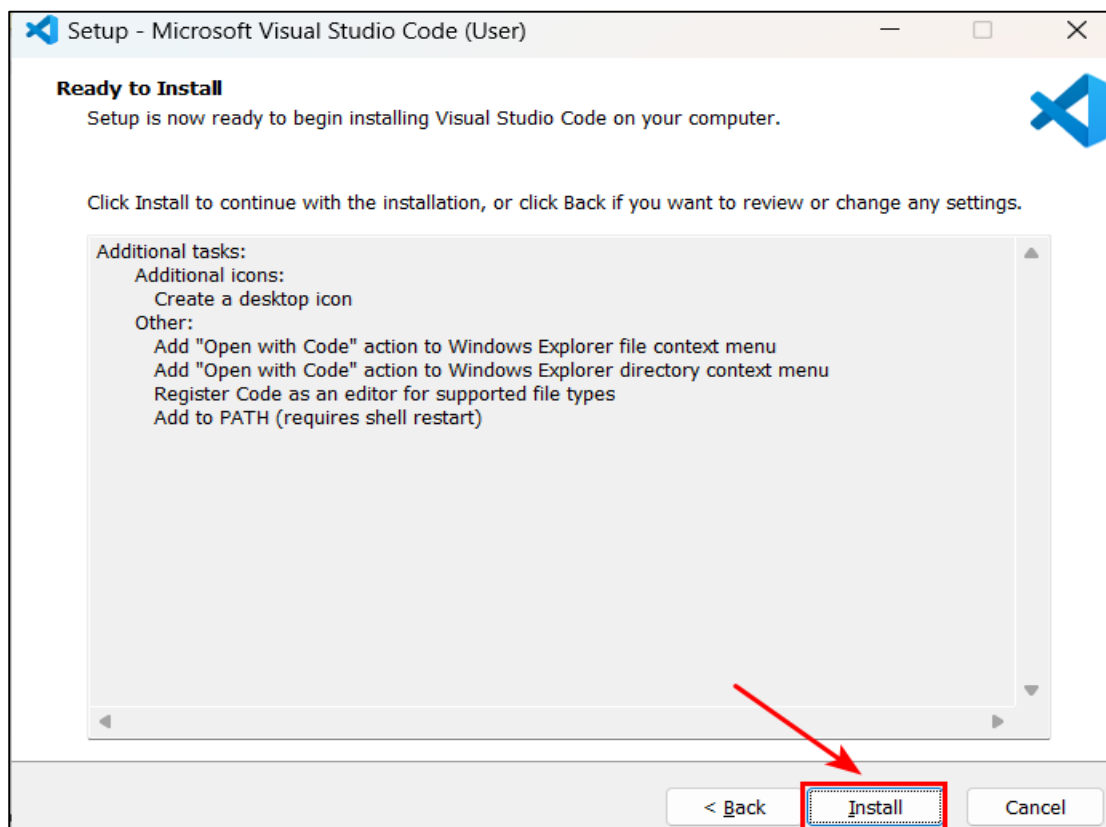


2. Chọn các tùy chọn bổ sung nếu cần, ví dụ như tạo shortcut trên desktop, và nhấn "Next".

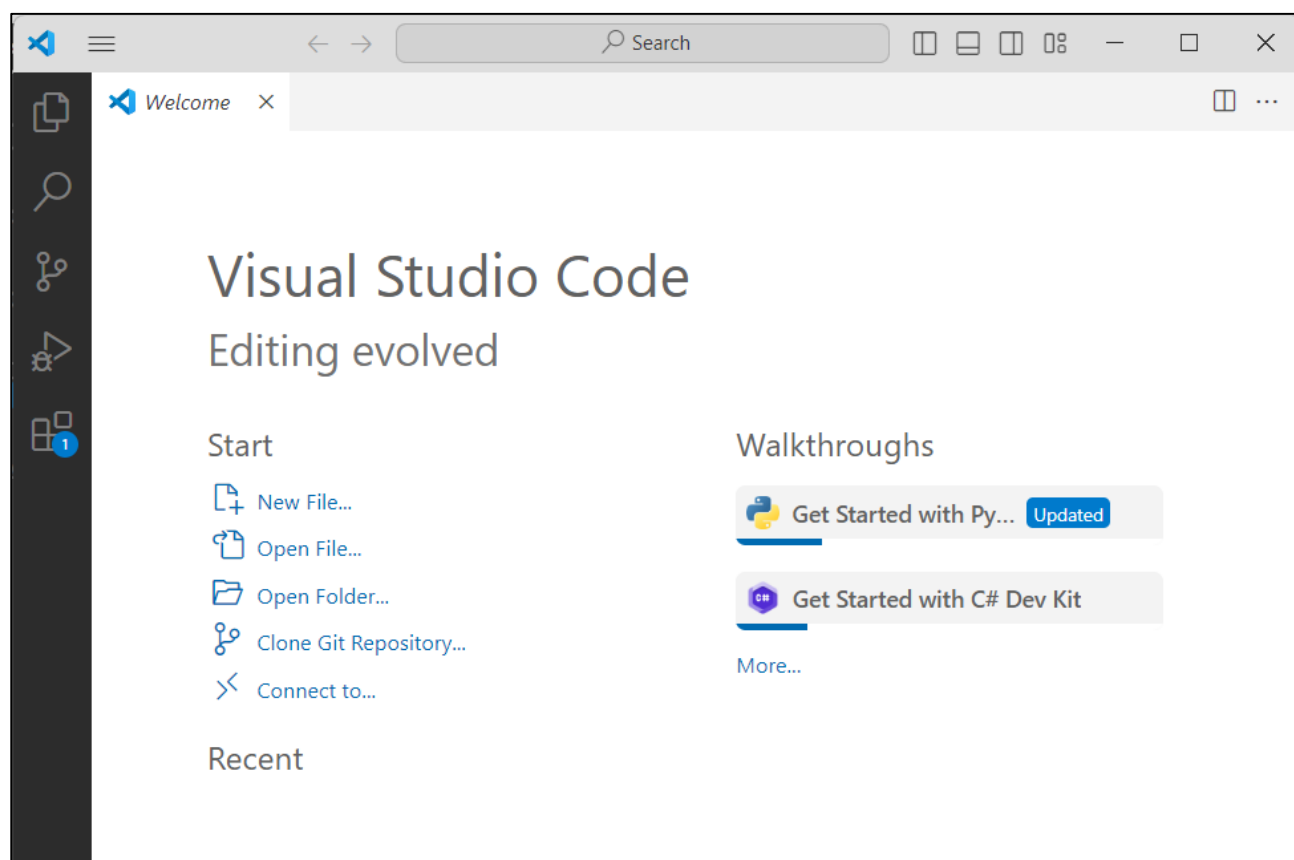


3. Nhấn "Install" để bắt đầu cài đặt.





4. Sau khi cài đặt xong, nhấn "Finish" để hoàn tất và mở Visual Studio Code.



### Bước 3: Cài đặt các tiện ích mở rộng (Extensions) cần thiết cho PHP

1. Mở Visual Studio Code.
2. Đi tới phần tiện ích mở rộng (Extensions) bằng cách nhấn tổ hợp phím Ctrl+Shift+X.
3. Tìm và cài đặt các tiện ích mở rộng sau:
  - **PHP Intelephense:** Hỗ trợ tự động hoàn thành và kiểm tra mã nguồn PHP.
  - **PHP Debug:** Hỗ trợ gỡ lỗi mã nguồn PHP.
  - **PHP CS Fixer:** Công cụ tự động sửa mã nguồn theo tiêu chuẩn.
  - **Code Runner:** Chạy PHP Script

## 1.2 Cài đặt Laragon

Laragon là một môi trường phát triển web trên Windows, giúp người dùng dễ dàng cài đặt và quản lý các công cụ cần thiết như Apache, PHP và MySQL. Laragon cung cấp một giao diện đồ họa thân thiện và khả năng tùy chỉnh linh hoạt, cho phép người dùng dễ dàng tạo và quản lý các môi trường phát triển web.

Truy cập trang website chính thức của Laragon để tải bản cài đặt về máy, sau đó tiến hành cài đặt.

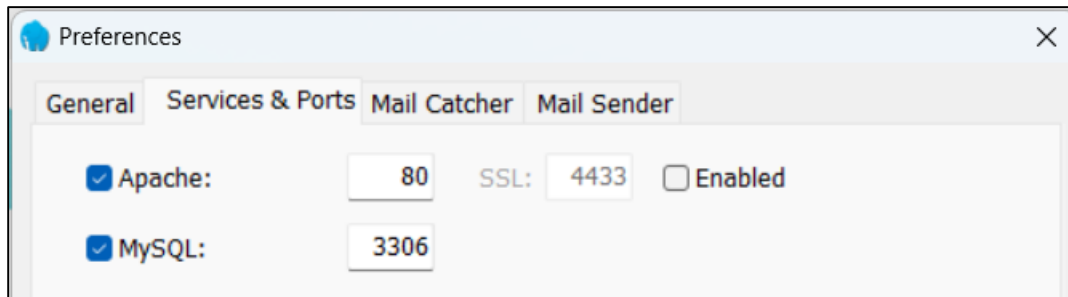
Link: <https://laragon.org/download/>

Tải xong bấm vào khởi động chương trình cài đặt phần mềm Laragon:



Các dịch vụ cần thiết liên quan tới một Website PHP:

- **Apache:** một trong những máy chủ web phổ biến nhất hiện nay.
- **MySQL:** hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến, thường được sử dụng với PHP để lưu trữ và truy xuất dữ liệu.



Trong trường hợp gặp lỗi do trùng cổng với các dịch vụ khác, hãy thử thay đổi cổng sử dụng.

## 1.3 Xây dựng ứng dụng web PHP đơn giản theo cấu trúc MVC

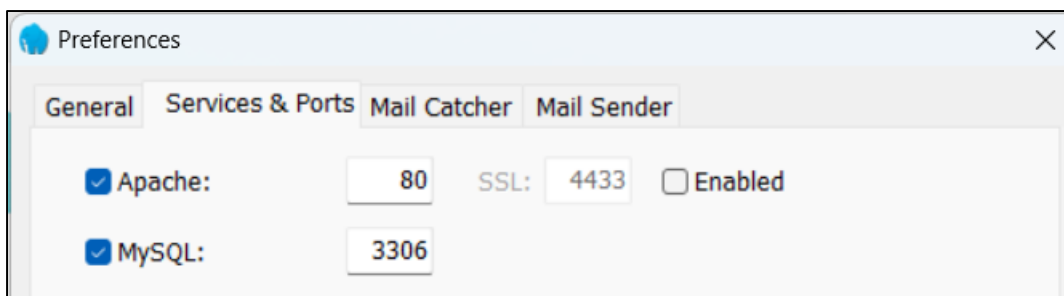
### Mục tiêu:

- Hiểu về kiến trúc MVC (Model-View-Controller).
- Khởi tạo dự án PHP theo kiến trúc MVC.
- Tạo các tệp tin cơ bản cho MVC: Model, View, Controller.

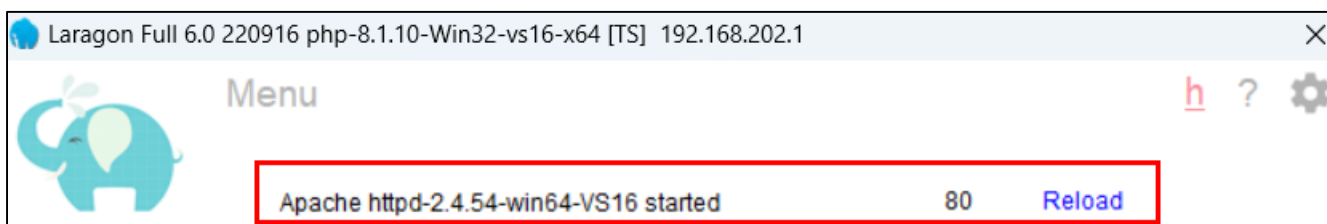
### Xây dựng trang Web hiện ra màn hình dòng chữ "HELLO HUTECH"

#### 1. Khởi động Laragon

Mở **Laragon** từ menu Start hoặc biểu tượng trên Desktop.



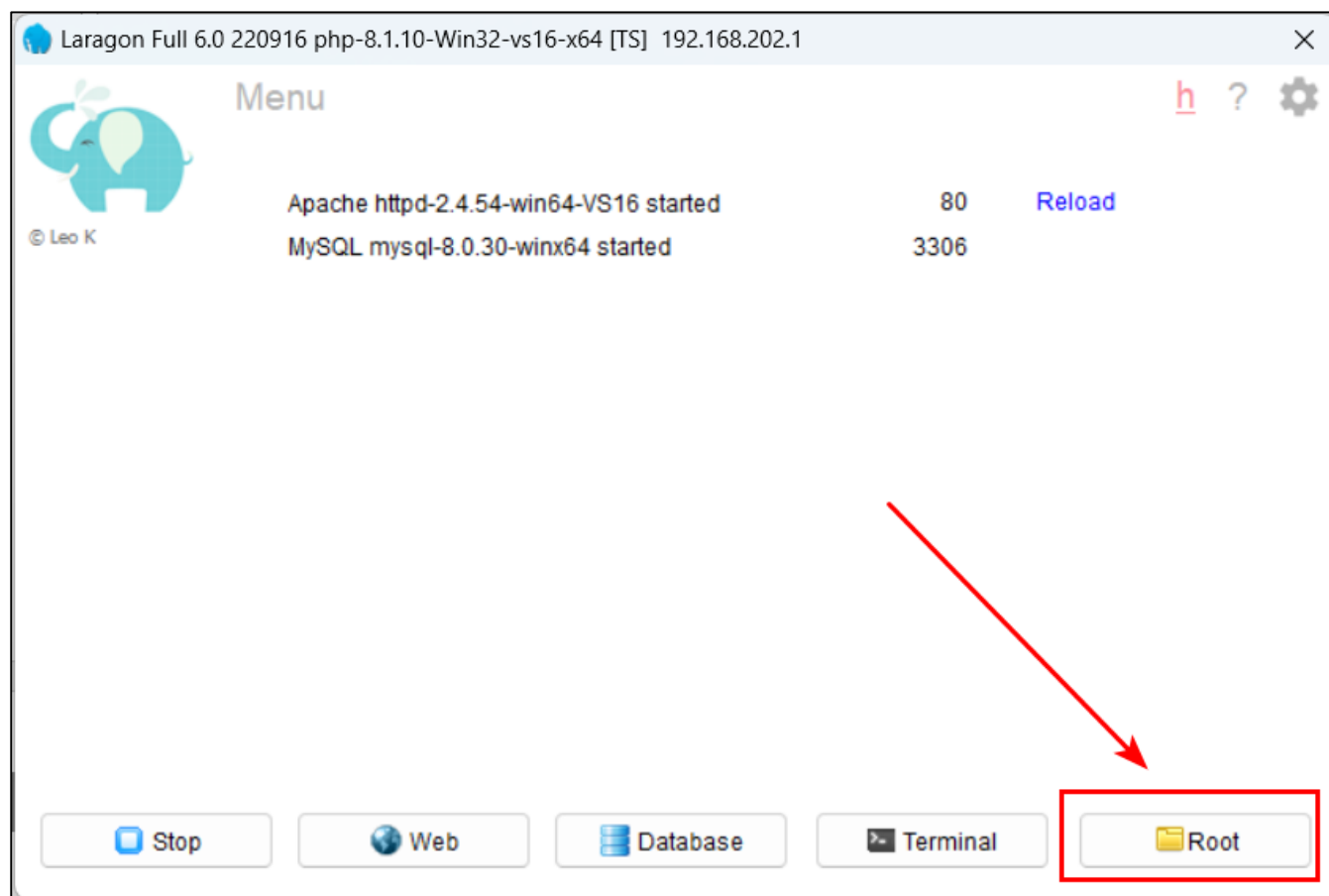
Nhấn nút **Start All** để khởi động dịch vụ **Apache**



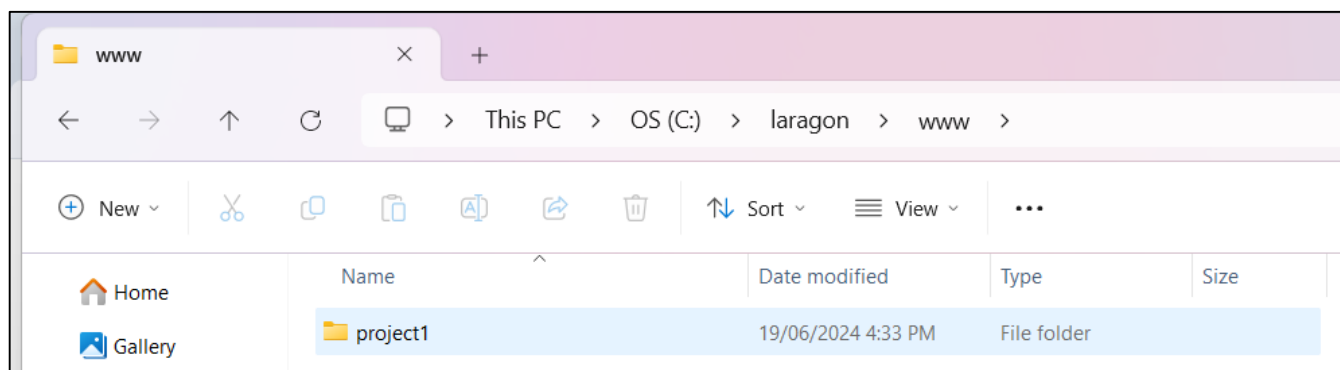
#### 2. Tạo dự án mới trong Laragon:

Chọn **Root** để mở thư mục gốc nơi Laragon lưu trữ các dự án web của bạn. Thông thường, thư mục này sẽ là **C:\laragon\www**.

Trong thư mục gốc (**www**), tạo một thư mục mới cho dự án.



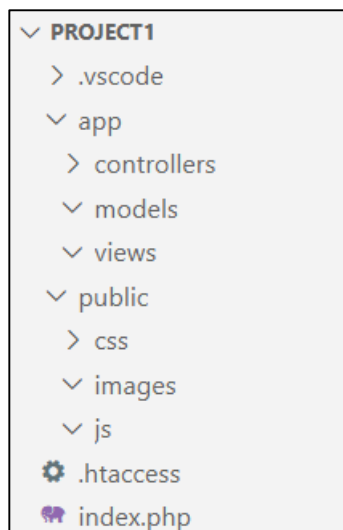
Ví dụ, tạo thư mục có tên **project1**. Vào đường dẫn **C:\laragon\www**.



### 3. Tạo cấu trúc thư mục dự án PHP theo kiến trúc MVC:

Thiết lập cấu trúc thư mục theo mô hình MVC (Model-View-Controller) cho dự án PHP.

1. Mở Visual Studio Code.
2. Mở thư mục dự án vừa tạo: File > Open Folder và chọn **C:\laragon\www\project1**.
3. Tạo cấu trúc thư mục như sau:



### 4. Cấu hình tệp tin '.htaccess':

```
RewriteEngine On
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ index.php?url=$1 [QSA,L]
```

### 5. Xây dựng tệp tin 'index.php':

```
<?php

$url = $_GET['url'] ?? '';
$url = rtrim($url, '/');
$url = filter_var($url, FILTER_SANITIZE_URL);
$url = explode('/', $url);
```

```
// Kiểm tra phần đầu tiên của URL để xác định controller
$controllerName = isset($url[0]) && $url[0] != '' ? ucfirst($url[0]) . 'Controller' :
'DefaultController';

// Kiểm tra phần thứ hai của URL để xác định action
$action = isset($url[1]) && $url[1] != '' ? $url[1] : 'index';

// Kiểm tra xem controller và action có tồn tại không
if (!file_exists('app/controllers/' . $controllerName . '.php')) {
    // Xử lý không tìm thấy controller
    die('Controller not found');
}

require_once 'app/controllers/' . $controllerName . '.php';

$controller = new $controllerName();

if (!method_exists($controller, $action)) {
    // Xử lý không tìm thấy action
    die('Action not found');
}

// Gọi action với các tham số còn lại (nếu có)
call_user_func_array([$controller, $action], array_slice($url, 2));
```

## 6. Tạo 'DefaultController.php' trong thư mục 'controllers'

```
<?php
class DefaultController
{
    public function index(){
        echo "HELLO HUTECH ";
    }
}
```

## 7. Chạy ứng dụng

Khởi động Laragon và đảm bảo Apache đang chạy.

Mở trình duyệt và truy cập vào URL: <http://localhost/project1/>

Dòng chữ "**HELLO HUTECH**" sẽ xuất hiện nếu mọi thứ đã được cấu hình đúng cách.

HELLO HUTECH

## 1.4 Xây dựng trang web bán hàng có các chức năng thêm/ xóa/ sửa/ hiển thị sản phẩm:

Các thuộc tính sản phẩm và ràng buộc nhập liệu như sau:

- Tên (Name): bắt buộc nhập, tối đa 100 ký tự, tối thiểu 10 ký tự
- Mô tả (Description)
- Giá: Lớn hơn 0

Bước 1: Xây dựng class ProductModel.php trong thư mục models:

```
<?php

class ProductModel
{
    // Thuộc tính của lớp ProductModel
    private $ID;
    private $Name;
    private $Description;
    private $Price;

    // Constructor để khởi tạo đối tượng ProductModel
    public function __construct($ID, $Name, $Description, $Price)
    {
        $this->ID = $ID;
        $this->Name = $Name;
        $this->Description = $Description;
        $this->Price = $Price;
    }

    // Getter và Setter cho thuộc tính ID
    public function getID()
    {
        return $this->ID;
    }

    public function setID($ID)
    {
        $this->ID = $ID;
    }
}
```



```
// Getter và Setter cho thuộc tính Name
public function getName()
{
    return $this->Name;
}

public function setName($Name)
{
    $this->Name = $Name;
}

// Getter và Setter cho thuộc tính Description
public function getDescription()
{
    return $this->Description;
}

public function setDescription($Description)
{
    $this->Description = $Description;
}

// Getter và Setter cho thuộc tính Price
public function getPrice()
{
    return $this->Price;
}

public function setPrice($Price)
{
    $this->Price = $Price;
}
}

?>
```

Bước 2: Xây dựng ProductController.php trong thư mục controllers

```
<?php

require_once 'app/models/ProductModel.php';

class ProductController
{
    private $products = [];
```

```
public function __construct()
{
    // Giả sử chúng ta lưu trữ sản phẩm trong session để giữ lại khi làm mới trang
    session_start();
    if (isset($_SESSION['products'])) {
        $this->products = $_SESSION['products'];
    }
}

public function index()
{
    $this->list();
}

public function list()
{
    // Hiển thị danh sách sản phẩm
    $products = $this->products;
    include 'app/views/product/list.php';
}

public function add()
{
    $errors = [];

    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $name = $_POST['name'];
        $description = $_POST['description'];
        $price = $_POST['price'];

        // Kiểm tra tên sản phẩm
        if (empty($name)) {
            $errors[] = 'Tên sản phẩm là bắt buộc.';
        } elseif (strlen($name) < 10 || strlen($name) > 100) {
            $errors[] = 'Tên sản phẩm phải có từ 10 đến 100 ký tự.';
        }

        // Kiểm tra giá
        if (!is_numeric($price) || $price <= 0) {
            $errors[] = 'Giá phải là một số dương lớn hơn 0.';
        }

        if (empty($errors)) {
            $id = count($this->products) + 1;

            $product = new ProductModel($id, $name, $description, $price);
            $this->products[] = $product;
        }
    }
}
```

```
        $_SESSION['products'] = $this->products;

        header('Location: /project1/Product/list');
        exit();
    }
}

include 'app/views/product/add.php';
}

public function edit($id)
{
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        foreach ($this->products as $key => $product) {
            if ($product->getID() == $id) {
                $this->products[$key]->setName($_POST['name']);
                $this->products[$key]->setDescription($_POST['description']);
                $this->products[$key]->setPrice($_POST['price']);
                break;
            }
        }

        $_SESSION['products'] = $this->products;

        header('Location: /project1/Product/list');
        exit();
    }

    foreach ($this->products as $product) {
        if ($product->getID() == $id) {
            include 'app/views/product/edit.php';
            return;
        }
    }

    die('Product not found');
}

public function delete($id)
{
    foreach ($this->products as $key => $product) {
        if ($product->getID() == $id) {
            unset($this->products[$key]);
            break;
        }
    }
}
```

```
$this->products = array_values($this->products);
$_SESSION['products'] = $this->products;

header('Location: /project1/Product/list');
exit();
}
}
?>
```

Bước 3: Xây dựng các trang hiển thị tương ứng:

Trong thư mục views tạo thư mục product

**app/views/product/add.php**

```
<!DOCTYPE html>
<html>
<head>
  <title>Thêm sản phẩm</title>
  <script>
    function validateForm() {
      let name = document.getElementById('name').value;
      let price = document.getElementById('price').value;
      let errors = [];

      if (name.length < 10 || name.length > 100) {
        errors.push('Tên sản phẩm phải có từ 10 đến 100 ký tự.');
      }

      if (price <= 0 || isNaN(price)) {
        errors.push('Giá phải là một số dương lớn hơn 0.');
      }

      if (errors.length > 0) {
        alert(errors.join('\n'));
        return false;
      }

      return true;
    }
  </script>
</head>
<body>
  <h1>Thêm sản phẩm mới</h1>
```

```

<?php if (!empty($errors)): ?>
    <ul>
        <?php foreach ($errors as $error): ?>
            <li><?php echo htmlspecialchars($error, ENT_QUOTES, 'UTF-8'); ?></li>
        <?php endforeach; ?>
    </ul>
<?php endif; ?>
<form method="POST" action="/project1/Product/add" onsubmit="return
validateForm();">
    <label for="name">Tên sản phẩm:</label>
    <input type="text" id="name" name="name" required><br><br>

    <label for="description">Mô tả:</label>
    <textarea id="description" name="description" required></textarea><br><br>

    <label for="price">Giá:</label>
    <input type="number" id="price" name="price" step="0.01" required><br><br>

    <button type="submit">Thêm sản phẩm</button>
</form>
<a href="/project1/Product/list">Quay lại danh sách sản phẩm</a>
</body>
</html>

```

## app/views/product/list.php

```

<!DOCTYPE html>
<html>
<head>
    <title>Danh sách sản phẩm</title>
</head>
<body>
    <h1>Danh sách sản phẩm</h1>
    <a href="/project1/Product/add">Thêm sản phẩm mới</a>
    <ul>
        <?php foreach ($products as $product): ?>
            <li>
                <h2><?php echo htmlspecialchars($product->getName(), ENT_QUOTES, 'UTF-8'); ?></h2>
                <p><?php echo htmlspecialchars($product->getDescription(), ENT_QUOTES, 'UTF-8'); ?></p>
                <p>Giá: <?php echo htmlspecialchars($product->getPrice(), ENT_QUOTES, 'UTF-8'); ?></p>
                <a href="/project1/Product/edit/<?php echo $product->getID(); ?>">Sửa</a>
                <a href="/project1/Product/delete/<?php echo $product->getID(); ?>"
onclick="return confirm('Bạn có chắc chắn muốn xóa sản phẩm này?');">Xóa</a>
            </li>
        <?php endforeach; ?>
    </ul>

```

```
        </li>
    <?php endforeach; ?>
</ul>
</body>
</html>
```

## app/views/product/edit.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Sửa sản phẩm</title>
</head>
<body>
    <h1>Sửa sản phẩm</h1>
    <form method="POST" action="/project1/Product/edit/<?php echo $product->getID();
?>">
        <label for="name">Tên sản phẩm:</label>
        <input type="text" id="name" name="name" value="<?php echo
htmlspecialchars($product->getName(), ENT_QUOTES, 'UTF-8'); ?>" required><br><br>

        <label for="description">Mô tả:</label>
        <textarea id="description" name="description" required><?php echo
htmlspecialchars($product->getDescription(), ENT_QUOTES, 'UTF-8');
?></textarea><br><br>

        <label for="price">Giá:</label>
        <input type="number" id="price" name="price" value="<?php echo
htmlspecialchars($product->getPrice(), ENT_QUOTES, 'UTF-8'); ?>" required><br><br>

        <button type="submit">Lưu thay đổi</button>
    </form>
    <a href="/project1/Product/list">Quay lại danh sách sản phẩm</a>
</body>
</html>
```

## Cập nhật file index.php

```
<?php

require_once 'app/models/ProductModel.php';

$url = $_GET['url'] ?? '';
$url = rtrim($url, '/');
$url = filter_var($url, FILTER_SANITIZE_URL);
$url = explode('/', $url);

// Kiểm tra phần đầu tiên của URL để xác định controller
$controllerName = isset($url[0]) && $url[0] != '' ? ucfirst($url[0]) . 'Controller' :
'DefaultController';

// Kiểm tra phần thứ hai của URL để xác định action
$action = isset($url[1]) && $url[1] != '' ? $url[1] : 'index';

// die ("controller=$controllerName - action=$action");

// Kiểm tra xem controller và action có tồn tại không
if (!file_exists('app/controllers/' . $controllerName . '.php')) {
    // Xử lý không tìm thấy controller
    die('Controller not found');
}

require_once 'app/controllers/' . $controllerName . '.php';

$controller = new $controllerName();

if (!method_exists($controller, $action)) {
    // Xử lý không tìm thấy action
    die('Action not found');
}

// Gọi action với các tham số còn lại (nếu có)
call_user_func_array([$controller, $action], array_slice($url, 2));
```

## Khởi chạy và kiểm tra kết quả

### Trang thêm sản phẩm:

## Thêm sản phẩm mới

Tên sản phẩm:

Mô tả:

Giá:

[Quay lại danh sách sản phẩm](#)

### Trang hiển thị sản phẩm:

## Danh sách sản phẩm

[Thêm sản phẩm mới](#)

### • Laptop Hutech

Laptop Hutech

Giá: 12300

[Sửa](#) [Xóa](#)



## Trang sửa sản phẩm:

### Sửa sản phẩm

Tên sản phẩm:

Mô tả:

Giá:

[Quay lại danh sách sản phẩm](#)

## Kết quả sau khi sửa

### Danh sách sản phẩm

[Thêm sản phẩm mới](#)

- Laptop Hutech 2024**

Laptop Hutech 2024

Giá: 12300

[Sửa](#) [Xóa](#)

## 1.5 YÊU CẦU BỔ SUNG

Xây dựng các trang views bằng Bootstrap

# BÀI 2. TẠO CƠ SỞ DỮ LIỆU CHO WEBSITE BÁN HÀNG, XÂY DỰNG CHỨC NĂNG HIỂN THỊ/ THÊM/ XÓA/ SỬA

Sau khi hoàn thành bài học, sinh viên sẽ nắm vững các kiến thức và kỹ năng sau:

## 1. Quản lý cơ sở dữ liệu MySQL:

Hiểu và biết cách sử dụng Laragon để quản lý cơ sở dữ liệu MySQL.

Tạo cơ sở dữ liệu và các bảng dữ liệu cần thiết cho một website bán hàng.

## 2. Xây dựng mô hình dữ liệu (Models):

Thiết kế các mô hình dữ liệu trong PHP tương ứng với các bảng trong cơ sở dữ liệu.

Hiểu cách ánh xạ các đối tượng PHP (Objects) với các bảng trong cơ sở dữ liệu MySQL.

## 3. Tạo và quản lý Controller:

Xây dựng các controller để xử lý các yêu cầu từ người dùng (HTTP requests).

Triển khai các phương thức CRUD (Create, Read, Update, Delete) cho các đối tượng dữ liệu.

## 4. Xây dựng giao diện người dùng (Views):

Tạo các file giao diện HTML để hiển thị danh sách sản phẩm, thêm mới, chỉnh sửa và xóa sản phẩm.

Sử dụng các mẫu giao diện để quản lý sản phẩm và danh mục sản phẩm.

## 5. Kết nối ứng dụng PHP với MySQL:

Hiểu và thực hiện kết nối giữa ứng dụng PHP và cơ sở dữ liệu MySQL.

Sử dụng PDO (PHP Data Objects) để truy vấn và thao tác với cơ sở dữ liệu.

## 6. Khởi chạy và kiểm tra ứng dụng:

Cấu hình và khởi chạy dự án PHP trên máy chủ local.

Kiểm tra và đảm bảo các chức năng thêm, xóa, sửa và hiển thị sản phẩm hoạt động đúng.

## **7. Xử lý và quản lý dữ liệu:**

Hiểu cách xử lý dữ liệu từ form và lưu trữ vào cơ sở dữ liệu.

Quản lý dữ liệu đầu vào và đầu ra, đảm bảo tính toàn vẹn và bảo mật dữ liệu.

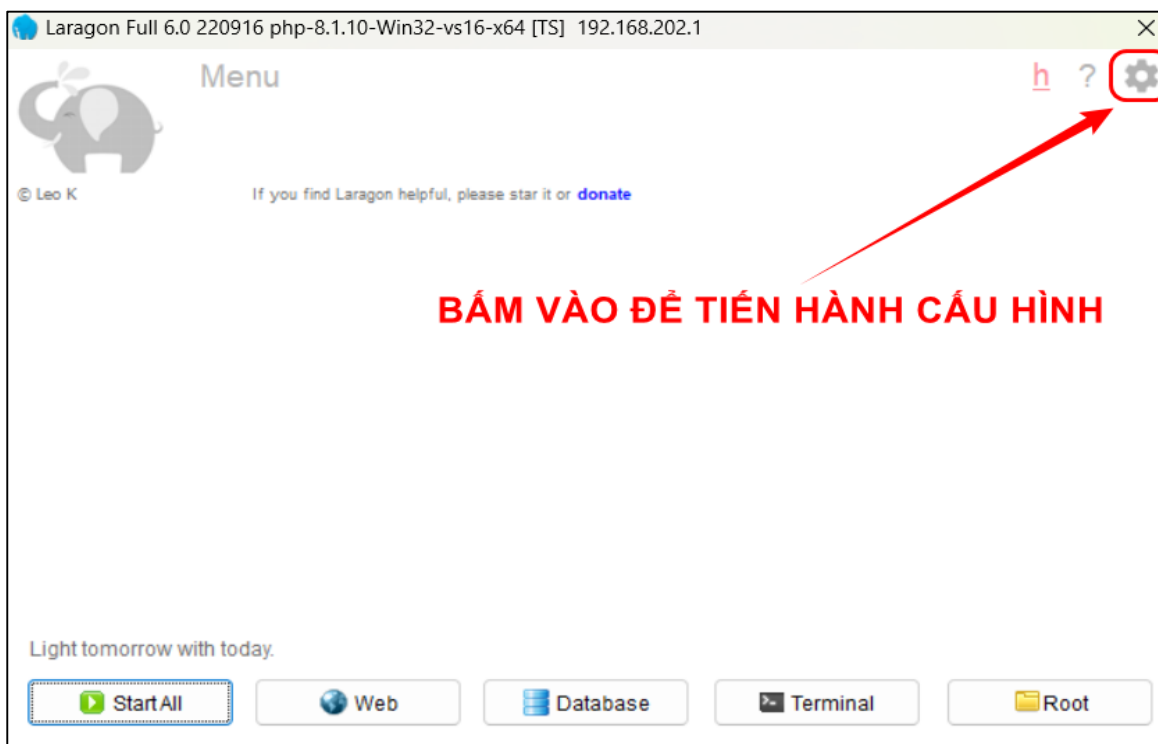
Bài học này cung cấp một nền tảng vững chắc cho sinh viên trong việc xây dựng các ứng dụng web động sử dụng PHP và MySQL. Sau bài học này, sinh viên sẽ có đủ kiến thức và kỹ năng để tiếp tục xây dựng các chức năng phức tạp hơn cho ứng dụng web của mình.

## **2.1 Sử dụng Laragon để quản lý cơ sở dữ liệu MySQL**

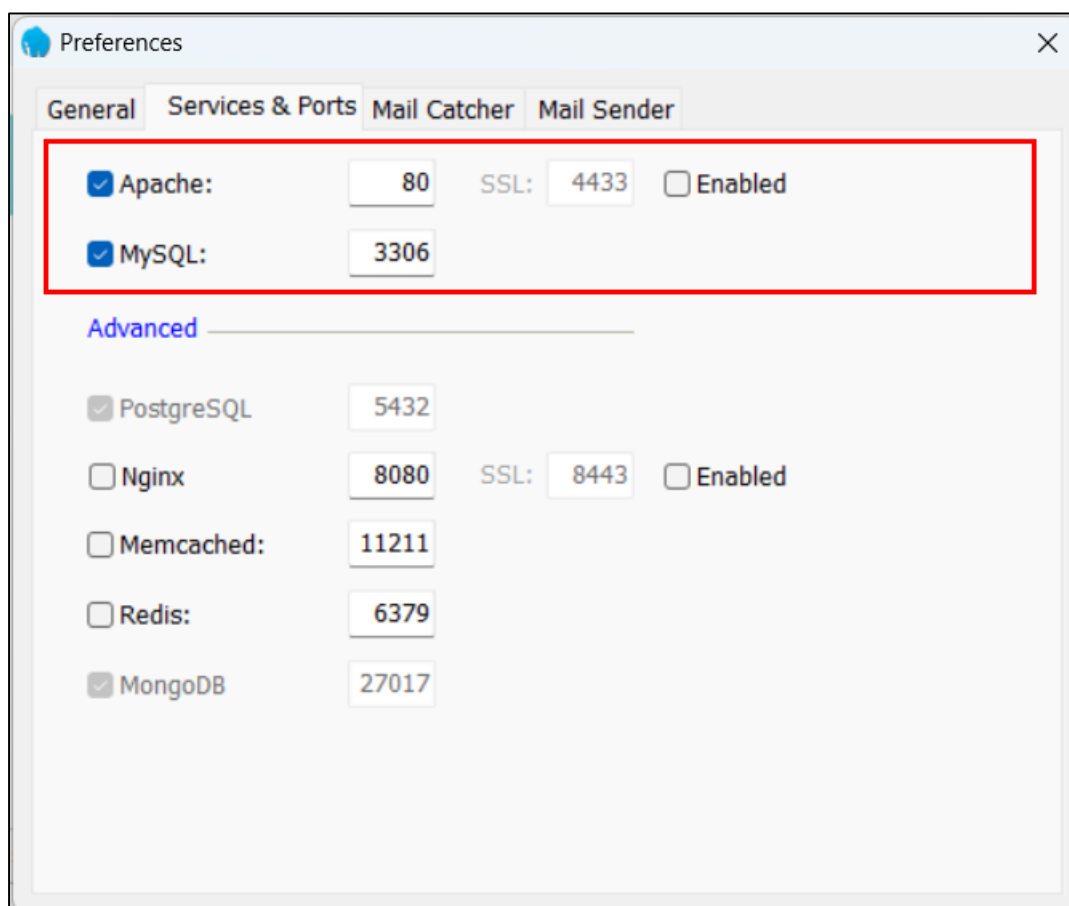
---

Laragon là một môi trường phát triển web dựa trên Windows, giúp người dùng dễ dàng cài đặt và quản lý các công cụ cần thiết như Apache, PHP và MySQL. Laragon cung cấp một giao diện đồ họa thân thiện và khả năng tùy chỉnh linh hoạt, cho phép người dùng dễ dàng tạo và quản lý các môi trường phát triển web.

Khởi động chương trình cài đặt phần mềm Laragon và tiến hành cấu hình



Khi lựa chọn cấu hình, chỉ tích chọn MySQL. Trong trường hợp gặp lỗi do trùng cổng với các dịch vụ khác, hãy thử thay đổi cổng sử dụng.



Sau khi thiết lập xong ấn **Start All** để khởi động. Kết quả như hình bên dưới:

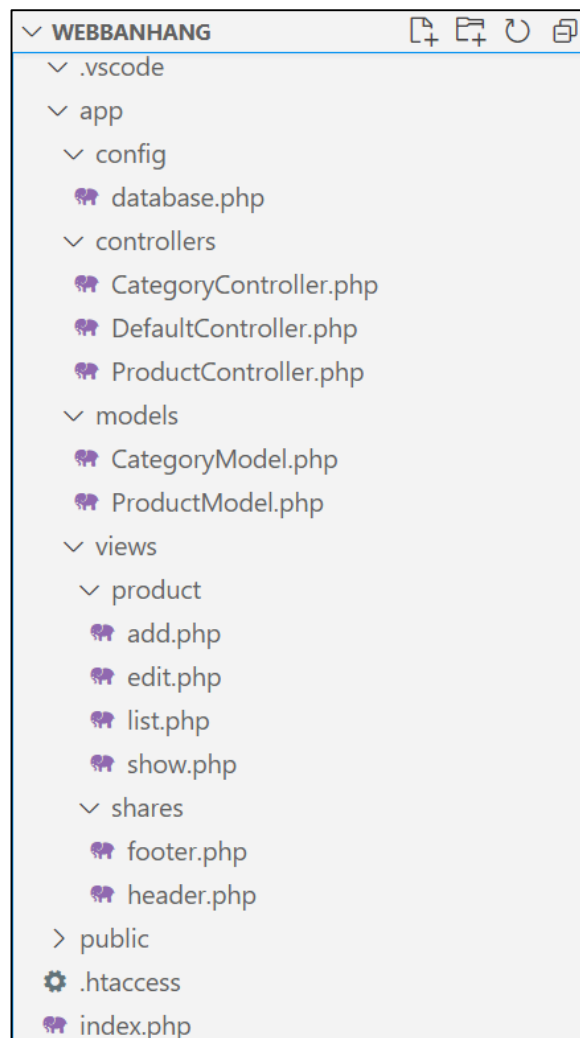
## 2.2 Xây dựng Website bán hàng bằng PHP kết nối với cơ sở dữ liệu MySQL

Xây dựng trang Web bán hàng với chức năng Hiển thị/Thêm/Xóa/Sửa sản phẩm.

### 2.2.1 Tạo dự án mới trong Laragon:

Chọn Root để mở thư mục gốc nơi Laragon lưu trữ các dự án web của bạn. Thông thường, thư mục này sẽ là C:\laragon\www.

Tạo dự án mang tên 'webbanhang' với cấu trúc thư mục như sau:



## 2.2.2 Tạo cơ sở dữ liệu

Tạo cơ sở dữ liệu có 2 bảng product và category

```
CREATE DATABASE my_store; USE my_store;
CREATE TABLE category ( id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, description TEXT
);
CREATE TABLE product (
    id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    description TEXT,
    price DECIMAL(10, 2) NOT NULL,
    image VARCHAR(255) DEFAULT NULL;

    category_id INT,
    FOREIGN KEY (category_id) REFERENCES category(id)
);
```

Thêm file '**database.php**' để kết nối database đã tạo trước đó trong thư mục có

**'/webbanhang/config/'**

```
<?php
class Database {
    private $host = "localhost";
    private $db_name = "my_store";
    private $username = "root";
    private $password = "";
    public $conn;

    public function getConnection() {
        $this->conn = null;

        try {
            $this->conn = new PDO("mysql:host=" . $this->host . ";dbname=" . $this->db_name, $this->username, $this->password);
            $this->conn->exec("set names utf8");
        } catch(PDOException $exception) {
            echo "Connection error: " . $exception->getMessage();
        }

        return $this->conn;
    }
}
```

### 2.2.3 Xây dựng các Model tương ứng

Trong thư mục models, tạo các file ProductModel.php và CategoryModel.php để đại diện cho các bảng trong cơ sở dữ liệu.

#### ProductModel.php

```
<?php
class ProductModel
{
    private $conn;
    private $table_name = "product";

    public function __construct($db)
    {
        $this->conn = $db;
    }

    public function getProducts()
    {
        $query = "SELECT p.id, p.name, p.description, p.price, c.name as category_name
        FROM " . $this->table_name . " p
        LEFT JOIN category c ON p.category_id = c.id";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);
        return $result;
    }

    public function getProductById($id)
    {
        $query = "SELECT * FROM " . $this->table_name . " WHERE id = :id";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(':id', $id);
        $stmt->execute();
        $result = $stmt->fetch(PDO::FETCH_OBJ);
        return $result;
    }

    public function addProduct($name, $description, $price, $category_id)
    {
        $errors = [];
        if (empty($name)) {
            $errors['name'] = 'Tên sản phẩm không được để trống';
        }
        if (empty($description)) {
            $errors['description'] = 'Mô tả không được để trống';
        }
    }
}
```

```
}
if (!is_numeric($price) || $price < 0) {
    $errors['price'] = 'Giá sản phẩm không hợp lệ';
}
if (count($errors) > 0) {
    return $errors;
}

$query = "INSERT INTO " . $this->table_name . " (name, description, price,
category_id) VALUES (:name, :description, :price, :category_id)";
$stmt = $this->conn->prepare($query);

$name = htmlspecialchars(strip_tags($name));
$description = htmlspecialchars(strip_tags($description));
$price = htmlspecialchars(strip_tags($price));
$category_id = htmlspecialchars(strip_tags($category_id));

$stmt->bindParam(':name', $name);
$stmt->bindParam(':description', $description);
$stmt->bindParam(':price', $price);
$stmt->bindParam(':category_id', $category_id);

if ($stmt->execute()) {
    return true;
}

return false;
}

public function updateProduct($id, $name, $description, $price, $category_id)
{
    $query = "UPDATE " . $this->table_name . " SET name=:name,
description=:description, price=:price, category_id=:category_id WHERE id=:id";
    $stmt = $this->conn->prepare($query);

    $name = htmlspecialchars(strip_tags($name));
    $description = htmlspecialchars(strip_tags($description));
    $price = htmlspecialchars(strip_tags($price));
    $category_id = htmlspecialchars(strip_tags($category_id));

    $stmt->bindParam(':id', $id);
    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':description', $description);
    $stmt->bindParam(':price', $price);
    $stmt->bindParam(':category_id', $category_id);

    if ($stmt->execute()) {
```



```

        return true;
    }
    return false;
}

public function deleteProduct($id)
{
    $query = "DELETE FROM " . $this->table_name . " WHERE id=:id";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(':id', $id);
    if ($stmt->execute()) {
        return true;
    }
    return false;
}
}
?>

```

### CategoryModel.php

```

<?php
class CategoryModel
{
    private $conn;
    private $table_name = "category";

    public function __construct($db)
    {
        $this->conn = $db;
    }

    public function getCategories()
    {
        $query = "SELECT id, name, description FROM " . $this->table_name;
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);
        return $result;
    }
}
?>

```

### 2.2.4 Khởi tạo các Controller tương ứng

Trong thư mục controllers, tạo các file ProductsController.php và CategoryController.php.

## ProductController.php

```
<?php
// Require SessionHelper and other necessary files
require_once('app/config/database.php');
require_once('app/models/ProductModel.php');
require_once('app/models/CategoryModel.php');

class ProductController
{
    private $productModel;
    private $db;

    public function __construct()
    {
        $this->db = (new Database())->getConnection();
        $this->productModel = new ProductModel($this->db);
    }

    public function index()
    {
        $products = $this->productModel->getProducts();
        include 'app/views/product/list.php';
    }

    public function show($id)
    {
        $product = $this->productModel->getProductById($id);

        if ($product) {
            include 'app/views/product/show.php';
        } else {
            echo "Không thấy sản phẩm.";
        }
    }

    public function add()
    {
        $categories = (new CategoryModel($this->db))->getCategories();
        include_once 'app/views/product/add.php';
    }

    public function save()
    {
        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            $name = $_POST['name'] ?? '';
            $description = $_POST['description'] ?? '';
            $price = $_POST['price'] ?? '';
        }
    }
}
```

```
        $category_id = $_POST['category_id'] ?? null;

        $result = $this->productModel->addProduct($name, $description, $price,
$category_id);

        if (is_array($result)) {
            $errors = $result;
            $categories = (new CategoryModel($this->db))->getCategories();
            include 'app/views/product/add.php';
        } else {

            header('Location: /webbanhang/Product');
        }
    }
}

public function edit($id)
{
    $product = $this->productModel->getProductById($id);
    $categories = (new CategoryModel($this->db))->getCategories();

    if ($product) {
        include 'app/views/product/edit.php';
    } else {
        echo "Không thấy sản phẩm.";
    }
}

public function update()
{
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $id = $_POST['id'];
        $name = $_POST['name'];
        $description = $_POST['description'];
        $price = $_POST['price'];
        $category_id = $_POST['category_id'];

        $edit = $this->productModel->updateProduct($id, $name, $description,
$price, $category_id);

        if ($edit) {
            header('Location: /webbanhang/Product');
        } else {
            echo "Đã xảy ra lỗi khi lưu sản phẩm.";
        }
    }
}
```

```
}

public function delete($id)
{
    if ($this->productModel->deleteProduct($id)) {
        header('Location: /webbanhang/Product');
    } else {
        echo "Đã xảy ra lỗi khi xóa sản phẩm.";
    }
}

}

?>
```

CategoryController.php

```
<?php
// Require SessionHelper and other necessary files
require_once('app/config/database.php');
require_once('app/models/CategoryModel.php');

class CategoryController
{
    private $categoryModel;
    private $db;

    public function __construct()
    {
        $this->db = (new Database())->getConnection();
        $this->categoryModel = new CategoryModel($this->db);
    }

    public function list()
    {
        $categories = $this->categoryModel->getCategories();
        include 'app/views/category/list.php';
    }
}

?>
```

## 2.2.5 Xây dựng giao diện hiển thị của trang web

### 2.2.5.1 Tạo các giao diện dùng chung:

Trong thư mục views tạo thêm thư mục shares, trong thư mục shares tạo thêm 2 file header.php và footer.php

File header.php

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Quản lý sản phẩm</title>
  <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-light bg-light">
    <a class="navbar-brand" href="#">Quản lý sản phẩm</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNav">
      <ul class="navbar-nav">
        <li class="nav-item">
          <a class="nav-link" href="/webbanhang/Product/">Danh sách sản
phẩm</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="/webbanhang/Product/add">Thêm sản
phẩm</a>
        </li>
      </ul>
    </div>
  </nav>
  <div class="container mt-4">
```

File footer.php

```
</div>
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
```

```

    <script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.5.2/dist/umd/popper.min.js"></scrip
t>
    <script
src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>

```

## 2.2.5.2 Tạo các views tương ứng cho product

Trong thư mục views, tạo các thư mục con products và categories, mỗi thư mục chứa các file giao diện tương ứng. Trong product tạo thêm các file sau:

### File 'list.php'

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Danh sách sản phẩm</h1>
<a href="/webbanhang/Product/add" class="btn btn-success mb-2">Thêm sản phẩm mới</a>
<ul class="list-group">
    <?php foreach ($products as $product): ?>
        <li class="list-group-item">
            <h2> <a href="/webbanhang/Product/show/<?php echo $product->id; ?>" >
                <?php echo htmlspecialchars($product->name, ENT_QUOTES, 'UTF-8'); ?>
            </a>
            </h2>
            <p><?php echo htmlspecialchars($product->description, ENT_QUOTES, 'UTF-
8'); ?></p>
            <p>Giá: <?php echo htmlspecialchars($product->price, ENT_QUOTES, 'UTF-8');
?></p>
            <a href="/webbanhang/Product/edit/<?php echo $product->id; ?>" class="btn
btn-warning">Sửa</a>
            <a href="/webbanhang/Product/delete/<?php echo $product->id; ?>"
class="btn btn-danger" onclick="return confirm('Bạn có chắc chắn muốn xóa sản phẩm
này?');">Xóa</a>
        </li>
    <?php endforeach; ?>
</ul>

<?php include 'app/views/shares/footer.php'; ?>

```

### File add.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Thêm sản phẩm mới</h1>

```

```

<?php if (!empty($errors)): ?>
    <div class="alert alert-danger">
        <ul>
            <?php foreach ($errors as $error): ?>
                <li><?php echo htmlspecialchars($error, ENT_QUOTES, 'UTF-8'); ?></li>
            <?php endforeach; ?>
        </ul>
    </div>
<?php endif; ?>
<form method="POST" action="/webbanhang/Product/save" onsubmit="return
validateForm();">
    <div class="form-group">
        <label for="name">Tên sản phẩm:</label>
        <input type="text" id="name" name="name" class="form-control" required>
    </div>
    <div class="form-group">
        <label for="description">Mô tả:</label>
        <textarea id="description" name="description" class="form-control"
required></textarea>
    </div>
    <div class="form-group">
        <label for="price">Giá:</label>
        <input type="number" id="price" name="price" class="form-control" step="0.01"
required>
    </div>
    <div class="form-group">
        <label for="category_id">Danh mục:</label>
        <select id="category_id" name="category_id" class="form-control" required>
            <?php foreach ($categories as $category): ?>
                <option value="<?php echo $category->id; ?>"><?php echo
htmlspecialchars($category->name, ENT_QUOTES, 'UTF-8'); ?></option>
            <?php endforeach; ?>
        </select>
    </div>
    <button type="submit" class="btn btn-primary">Thêm sản phẩm</button>
</form>
<a href="/webbanhang/Product/list" class="btn btn-secondary mt-2">Quay lại danh sách
sản phẩm</a>

<?php include 'app/views/shares/footer.php'; ?>

```

## File edit.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Sửa sản phẩm</h1>
<?php if (!empty($errors)): ?>

```

```

<div class="alert alert-danger">
    <ul>
        <?php foreach ($errors as $error): ?>
            <li><?php echo htmlspecialchars($error, ENT_QUOTES, 'UTF-8'); ?></li>
        <?php endforeach; ?>
    </ul>
</div>
<?php endif; ?>
<form method="POST" action="/webbanhang/Product/update" onsubmit="return
validateForm();">
    <input type="hidden" name="id" value="<?php echo $product->id; ?>">
    <div class="form-group">
        <label for="name">Tên sản phẩm:</label>
        <input type="text" id="name" name="name" class="form-control" value="<?php
echo htmlspecialchars($product->name, ENT_QUOTES, 'UTF-8'); ?>" required>
    </div>
    <div class="form-group">
        <label for="description">Mô tả:</label>
        <textarea id="description" name="description" class="form-control"
required><?php echo htmlspecialchars($product->description, ENT_QUOTES, 'UTF-8');
?></textarea>
    </div>
    <div class="form-group">
        <label for="price">Giá:</label>
        <input type="number" id="price" name="price" class="form-control" step="0.01"
value="<?php echo htmlspecialchars($product->price, ENT_QUOTES, 'UTF-8'); ?>"
required>
    </div>
    <div class="form-group">
        <label for="category_id">Danh mục:</label>
        <select id="category_id" name="category_id" class="form-control" required>
            <?php foreach ($categories as $category): ?>
                <option value="<?php echo $category->id; ?>" <?php echo $category->id
== $product->category_id ? 'selected' : ''; ?>>
                    <?php echo htmlspecialchars($category->name, ENT_QUOTES, 'UTF-8');
?>
                </option>
            <?php endforeach; ?>
        </select>
    </div>
    <button type="submit" class="btn btn-primary">Lưu thay đổi</button>
</form>
<a href="/webbanhang/Product/list" class="btn btn-secondary mt-2">Quay lại danh sách
sản phẩm</a>

<?php include 'app/views/shares/footer.php'; ?>

```



## 2.2.6 Tiến hay khởi chạy dự án và thực nghiệm

Trang thêm sản phẩm:

[Quản lý sản phẩm](#) [Danh sách sản phẩm](#) [Thêm sản phẩm](#)

## Thêm sản phẩm mới

Tên sản phẩm:

Mô tả:

Giá:

Danh mục:

Thêm sản phẩmQuay lại danh sách sản phẩm

Trang hiển thị sản phẩm:

[Quản lý sản phẩm](#) [Danh sách sản phẩm](#) [Thêm sản phẩm](#)

## Danh sách sản phẩm

Thêm sản phẩm mới

### Laptop

A high-performance laptop

Giá: 999.99

[Sửa](#) [Xóa](#)

### Smartphone

A latest model smartphone

Giá: 699.99

[Sửa](#) [Xóa](#)

Trang sửa sản phẩm:

Quản lý sản phẩm   Danh sách sản phẩm   Thêm sản phẩm

## Sửa sản phẩm

Tên sản phẩm:

Mô tả:

Giá:

Danh mục:

Lưu thay đổiQuay lại danh sách sản phẩm

Kiểm tra Database:

Laragon.MySQL\my\_store\product\ - HeidiSQL Portable 12.1.0.6537

File Edit Search Query Tools Go to Help

Database filter Table filter Host: 127.0.0.1 Database: my\_store Table: product Data Query

my\_store.product: 5 rows total (approximately)

id	name	description	price	category_id
56	Laptop	A high-performance laptop	999.99	1
57	Smartphone	A latest model smartphone	699.99	1
58	Novel	A captivating novel	19.99	2
59	T-shirt	A comfortable cotton t-shirt	9.99	3
60	Laptop 2024	Laptop 2024	12,300.0	1

## 2.3 YÊU CẦU BỔ SUNG

Xây dựng thêm các chức năng chèn hình và hiển thị hình ảnh cho Product, xây dựng các chức năng thêm xóa sửa của Category.

**Hướng dẫn chèn hình và hiển thị hình ảnh:**

### 1. Cập nhật cơ sở dữ liệu

Thêm một cột image vào bảng product để lưu đường dẫn hình ảnh:

```
ALTER TABLE product ADD COLUMN image VARCHAR(255) DEFAULT NULL;
```

### 2. Cập nhật model

Cập nhật phương thức **addProduct** và **updateProduct** để xử lý lưu đường dẫn hình ảnh:

```
<?php
class ProductModel
{
    private $conn;
    private $table_name = "product";

    public function __construct($db)
    {
        $this->conn = $db;
    }

    public function getProducts()
    {
        $query = "SELECT p.id, p.name, p.description, p.price, p.image, c.name as
category_name
                FROM " . $this->table_name . " p
                LEFT JOIN category c ON p.category_id = c.id";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);
        return $result;
    }

    public function getProductById($id)
    {
        $query = "SELECT * FROM " . $this->table_name . " WHERE id = :id";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(':id', $id);
        $stmt->execute();
    }
}
```

```
$result = $stmt->fetch(PDO::FETCH_OBJ);
return $result;
}

public function addProduct($name, $description, $price, $category_id, $image)
{
    $errors = [];
    if (empty($name)) {
        $errors['name'] = 'Tên sản phẩm không được để trống';
    }
    if (empty($description)) {
        $errors['description'] = 'Mô tả không được để trống';
    }
    if (!is_numeric($price) || $price < 0) {
        $errors['price'] = 'Giá sản phẩm không hợp lệ';
    }
    if (count($errors) > 0) {
        return $errors;
    }

    $query = "INSERT INTO " . $this->table_name . " (name, description, price,
category_id, image) VALUES (:name, :description, :price, :category_id, :image)";
    $stmt = $this->conn->prepare($query);

    $name = htmlspecialchars(strip_tags($name));
    $description = htmlspecialchars(strip_tags($description));
    $price = htmlspecialchars(strip_tags($price));
    $category_id = htmlspecialchars(strip_tags($category_id));
    $image = htmlspecialchars(strip_tags($image));

    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':description', $description);
    $stmt->bindParam(':price', $price);
    $stmt->bindParam(':category_id', $category_id);
    $stmt->bindParam(':image', $image);

    if ($stmt->execute()) {
        return true;
    }

    return false;
}

public function updateProduct($id, $name, $description, $price, $category_id,
$image)
{

```

```

        $query = "UPDATE " . $this->table_name . " SET name=:name,
description=:description, price=:price, category_id=:category_id, image=:image WHERE
id=:id";
        $stmt = $this->conn->prepare($query);

        $name = htmlspecialchars(strip_tags($name));
        $description = htmlspecialchars(strip_tags($description));
        $price = htmlspecialchars(strip_tags($price));
        $category_id = htmlspecialchars(strip_tags($category_id));
        $image = htmlspecialchars(strip_tags($image));

        $stmt->bindParam(':id', $id);
        $stmt->bindParam(':name', $name);
        $stmt->bindParam(':description', $description);
        $stmt->bindParam(':price', $price);
        $stmt->bindParam(':category_id', $category_id);
        $stmt->bindParam(':image', $image);

        if ($stmt->execute()) {
            return true;
        }
        return false;
    }

    public function deleteProduct($id)
    {
        $query = "DELETE FROM " . $this->table_name . " WHERE id=:id";
        $stmt = $this->conn->prepare($query);
        $stmt->bindParam(':id', $id);
        if ($stmt->execute()) {
            return true;
        }
        return false;
    }
}
?>

```

### 3. Cập nhật controller

ProductController.php

Cập nhật phương thức save và update để xử lý upload hình ảnh:

```

<?php
// Require SessionHelper and other necessary files
require_once('app/config/database.php');
require_once('app/models/ProductModel.php');

```

```
require_once('app/models/CategoryModel.php');

class ProductController
{
    private $productModel;
    private $db;

    public function __construct()
    {
        $this->db = (new Database())->getConnection();
        $this->productModel = new ProductModel($this->db);
    }

    public function index()
    {
        $products = $this->productModel->getProducts();
        include 'app/views/product/list.php';
    }

    public function show($id)
    {
        $product = $this->productModel->getProductById($id);

        if ($product) {
            include 'app/views/product/show.php';
        } else {
            echo "Không thấy sản phẩm.";
        }
    }

    public function add()
    {
        $categories = (new CategoryModel($this->db))->getCategories();
        include_once 'app/views/product/add.php';
    }

    public function save()
    {
        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            $name = $_POST['name'] ?? '';
            $description = $_POST['description'] ?? '';
            $price = $_POST['price'] ?? '';
            $category_id = $_POST['category_id'] ?? null;

            if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
                $image = $this->uploadImage($_FILES['image']);
            } else {
```

```
        $image = "";
    }

    $result = $this->productModel->addProduct($name, $description, $price,
$category_id, $image);

    if (is_array($result)) {
        $errors = $result;
        $categories = (new CategoryModel($this->db))->getCategories();
        include 'app/views/product/add.php';
    } else {

        header('Location: /webbanhang/Product');
    }
}

}

public function edit($id)
{
    $product = $this->productModel->getProductById($id);
    $categories = (new CategoryModel($this->db))->getCategories();

    if ($product) {
        include 'app/views/product/edit.php';
    } else {
        echo "Không thấy sản phẩm.";
    }
}

public function update()
{
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $id = $_POST['id'];
        $name = $_POST['name'];
        $description = $_POST['description'];
        $price = $_POST['price'];
        $category_id = $_POST['category_id'];

        if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
            $image = $this->uploadImage($_FILES['image']);
        } else {
            $image = $_POST['existing_image'];
        }

        $edit = $this->productModel->updateProduct($id, $name, $description,
$price, $category_id, $image);
    }
}
```



```
        if ($edit) {
            header('Location: /webbanhang/Product');
        } else {
            echo "Đã xảy ra lỗi khi lưu sản phẩm.";
        }
    }
}

public function delete($id)
{
    if ($this->productModel->deleteProduct($id)) {
        header('Location: /webbanhang/Product');
    } else {
        echo "Đã xảy ra lỗi khi xóa sản phẩm.";
    }
}

private function uploadImage($file)
{
    $target_dir = "uploads/";

    // Kiểm tra và tạo thư mục nếu chưa tồn tại
    if (!is_dir($target_dir)) {
        mkdir($target_dir, 0777, true);
    }

    $target_file = $target_dir . basename($file["name"]);
    $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));

    // Kiểm tra xem file có phải là hình ảnh không
    $check = getimagesize($file["tmp_name"]);
    if ($check === false) {
        throw new Exception("File không phải là hình ảnh.");
    }

    // Kiểm tra kích thước file (10 MB = 10 * 1024 * 1024 bytes)
    if ($file["size"] > 10 * 1024 * 1024) {
        throw new Exception("Hình ảnh có kích thước quá lớn.");
    }

    // Chỉ cho phép một số định dạng hình ảnh nhất định
    if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType !=
"jpeg" && $imageFileType != "gif") {
        throw new Exception("Chỉ cho phép các định dạng JPG, JPEG, PNG và GIF.");
    }

    // Lưu file
```

```

        if (!move_uploaded_file($file["tmp_name"], $target_file)) {
            throw new Exception("Có lỗi xảy ra khi tải lên hình ảnh.");
        }

        return $target_file;
    }

    public function addToCart($id)
    {
        $product = $this->productModel->getProductById($id);
        if (!$product) {
            echo "Không tìm thấy sản phẩm.";
            return;
        }

        if (!isset($_SESSION['cart'])) {
            $_SESSION['cart'] = [];
        }

        if (isset($_SESSION['cart'][$id])) {
            $_SESSION['cart'][$id]['quantity']++;
        } else {
            $_SESSION['cart'][$id] = [
                'name' => $product->name,
                'price' => $product->price,
                'quantity' => 1,
                'image' => $product->image
            ];
        }

        header('Location: /webbanhang/Product/cart');
    }
}
?>

```

#### 4. Cập nhật views:

app/views/product/add.php

Thêm trường upload hình ảnh vào form:

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Thêm sản phẩm mới</h1>
<?php if (!empty($errors)): ?>

```

```

<div class="alert alert-danger">
    <ul>
        <?php foreach ($errors as $error): ?>
            <li><?php echo htmlspecialchars($error, ENT_QUOTES, 'UTF-8'); ?></li>
        <?php endforeach; ?>
    </ul>
</div>
<?php endif; ?>
<form method="POST" action="/webbanhang/Product/save" enctype="multipart/form-data"
onsubmit="return validateForm();">
    <div class="form-group">
        <label for="name">Tên sản phẩm:</label>
        <input type="text" id="name" name="name" class="form-control" required>
    </div>
    <div class="form-group">
        <label for="description">Mô tả:</label>
        <textarea id="description" name="description" class="form-control"
required></textarea>
    </div>
    <div class="form-group">
        <label for="price">Giá:</label>
        <input type="number" id="price" name="price" class="form-control" step="0.01"
required>
    </div>
    <div class="form-group">
        <label for="category_id">Danh mục:</label>
        <select id="category_id" name="category_id" class="form-control" required>
            <?php foreach ($categories as $category): ?>
                <option value="<?php echo $category->id; ?>"><?php echo
htmlspecialchars($category->name, ENT_QUOTES, 'UTF-8'); ?></option>
            <?php endforeach; ?>
        </select>
    </div>
    <div class="form-group">
        <label for="image">Hình ảnh:</label>
        <input type="file" id="image" name="image" class="form-control">
    </div>
    <button type="submit" class="btn btn-primary">Thêm sản phẩm</button>
</form>
<a href="/webbanhang/Product/list" class="btn btn-secondary mt-2">Quay lại danh sách
sản phẩm</a>

<?php include 'app/views/shares/footer.php'; ?>

```

app/views/product/edit.php

Thêm trường upload hình ảnh vào form và hiển thị hình ảnh hiện tại:

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Sửa sản phẩm</h1>
<?php if (!empty($errors)): ?>
    <div class="alert alert-danger">
        <ul>
            <?php foreach ($errors as $error): ?>
                <li><?php echo htmlspecialchars($error, ENT_QUOTES, 'UTF-8'); ?></li>
            <?php endforeach; ?>
        </ul>
    </div>
<?php endif; ?>
<form method="POST" action="/webbanhang/Product/update" enctype="multipart/form-data"
onsubmit="return validateForm();">
    <input type="hidden" name="id" value="<?php echo $product->id; ?>">
    <div class="form-group">
        <label for="name">Tên sản phẩm:</label>
        <input type="text" id="name" name="name" class="form-control" value="<?php
echo htmlspecialchars($product->name, ENT_QUOTES, 'UTF-8'); ?>" required>
    </div>
    <div class="form-group">
        <label for="description">Mô tả:</label>
        <textarea id="description" name="description" class="form-control"
required><?php echo htmlspecialchars($product->description, ENT_QUOTES, 'UTF-8');
?></textarea>
    </div>
    <div class="form-group">
        <label for="price">Giá:</label>
        <input type="number" id="price" name="price" class="form-control" step="0.01"
value="<?php echo htmlspecialchars($product->price, ENT_QUOTES, 'UTF-8'); ?>"
required>
    </div>
    <div class="form-group">
        <label for="category_id">Danh mục:</label>
        <select id="category_id" name="category_id" class="form-control" required>
            <?php foreach ($categories as $category): ?>
                <option value="<?php echo $category->id; ?>" <?php echo $category->id
== $product->category_id ? 'selected' : ''; ?>>
                    <?php echo htmlspecialchars($category->name, ENT_QUOTES, 'UTF-8');
?>
            </option>
            <?php endforeach; ?>
        </select>
    </div>
    <div class="form-group">
        <label for="image">Hình ảnh:</label>
        <input type="file" id="image" name="image" class="form-control">

```

```

        <input type="hidden" name="existing_image" value="<?php echo $product->image;
?>">
        <?php if ($product->image): ?>
            
        <?php endif; ?>
    </div>
    <button type="submit" class="btn btn-primary">Lưu thay đổi</button>
</form>
<a href="/webbanhang/Product/list" class="btn btn-secondary mt-2">Quay lại danh sách
sản phẩm</a>

<?php include 'app/views/shares/footer.php'; ?>

```

Hiển thị hình ảnh trong danh sách sản phẩm

app/views/product/list.php

Thêm hiển thị hình ảnh vào danh sách sản phẩm:

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Danh sách sản phẩm</h1>
<a href="/webbanhang/Product/add" class="btn btn-success mb-2">Thêm sản phẩm mới</a>
<ul class="list-group">
    <?php foreach ($products as $product): ?>
        <li class="list-group-item">
            <h2><a href="/webbanhang/Product/show/<?php echo $product->id; ?>"><?php
echo htmlspecialchars($product->name, ENT_QUOTES, 'UTF-8'); ?></a></h2>
            <?php if ($product->image): ?>
                
            <?php endif; ?>
            <p><?php echo htmlspecialchars($product->description, ENT_QUOTES, 'UTF-
8'); ?></p>
            <p>Giá: <?php echo htmlspecialchars($product->price, ENT_QUOTES, 'UTF-8');
?> VND</p>
            <p>Danh mục: <?php echo htmlspecialchars($product->category_name,
ENT_QUOTES, 'UTF-8'); ?></p>
            <a href="/webbanhang/Product/edit/<?php echo $product->id; ?>" class="btn
btn-warning">Sửa</a>
            <a href="/webbanhang/Product/delete/<?php echo $product->id; ?>"
class="btn btn-danger" onclick="return confirm('Bạn có chắc chắn muốn xóa sản phẩm
này?');">Xóa</a>
        </li>
    <?php endforeach; ?>
</ul>

```

```
<?php include 'app/views/shares/footer.php'; ?>
```

## 5. Tiến khởi chạy dự án và kiểm tra kết quả:

Trang thêm sản phẩm:

[Quản lý sản phẩm](#) [Danh sách sản phẩm](#) [Thêm sản phẩm](#)

## Thêm sản phẩm mới

Tên sản phẩm:

Mô tả:

Giá:

Danh mục:

Hình ảnh:

laptop2.png

## Trang hiển thị danh sách sản phẩm

[Quản lý sản phẩm](#)[Danh sách sản phẩm](#)[Thêm sản phẩm](#)

# Danh sách sản phẩm

[Thêm sản phẩm mới](#)

## Laptop



A high-performance laptop

Giá: 999.99 VND

Danh mục: Electronics

[Sửa](#)[Xóa](#)

## Smartphone



A latest model smartphone

Giá: 699.99 VND

## Trang sửa sản phẩm

Quản lý sản phẩm
Danh sách sản phẩm
Thêm sản phẩm

# Sửa sản phẩm

Tên sản phẩm:


Mô tả:

Giá:

Danh mục:

Hình ảnh:

No file chosen



## Kiểm tra Database:

Laragon.MySQL\my\_store\product\ - HeidiSQL Portable 12.1.0.6537

File Edit Search Query Tools Go to Help

Database filter Table filter Host: 127.0.0.1 Database: my\_store Table: product Data Query

my\_store.product: 4 rows total (approximately)

id	name	description	price	category_id	image
56	Laptop	A high-performance laptop	999.99	1	uploads/laptop.jpg
57	Smartphone	A latest model smartphone	699.99	1	uploads/hinh smartphone.jpg
58	Novel	A captivating novel	19.99	2	uploads/hinh novel.jpg
59	T-shirt	A comfortable cotton t-shirt	9.99	3	uploads/hinh tshirt.jpg

**THÊM SẢN PHẨM VÀ LƯU HÌNH ẢNH THÀNH CÔNG**



## BÀI 3. XÂY DỰNG CHỨC NĂNG GIỎ HÀNG. ĐẶT HÀNG, THANH TOÁN

Sau khi học xong bài này, học viên có thể:

- Thiết kế cơ sở dữ liệu cho giỏ hàng và đặt hàng, tạo các bảng orders và order\_details trong MySQL.
- Cập nhật ProductController để xử lý việc thêm sản phẩm vào giỏ hàng, quản lý giỏ hàng và cập nhật số lượng sản phẩm.
- Tạo các giao diện HTML cho giỏ hàng, cho phép người dùng thêm, xóa, cập nhật sản phẩm và tiến hành đặt hàng.
- Sử dụng session trong PHP để lưu trữ và quản lý thông tin giỏ hàng của người dùng.
- Cấu hình và khởi chạy dự án PHP trên máy chủ local, kiểm tra chức năng giỏ hàng, đặt hàng và thanh toán.

Với các kỹ năng này, sinh viên sẽ có khả năng xây dựng và triển khai các chức năng giỏ hàng, đặt hàng và thanh toán cho một website bán hàng.

### 3.1 Tạo bảng orders và order\_details

Tạo bảng orders để lưu thông tin về các đơn hàng và bảng order\_details để lưu chi tiết các sản phẩm trong từng đơn hàng. Hai bảng này có mối quan hệ 1-nhiều.

SQL để tạo bảng orders

```
CREATE TABLE orders (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(255) NOT NULL,  
    phone VARCHAR(20) NOT NULL,  
    address TEXT NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

SQL để tạo bảng order\_details

```
CREATE TABLE order_details (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    order_id INT NOT NULL,  
    product_id INT NOT NULL,  
    quantity INT NOT NULL,  
    price DECIMAL(10, 2) NOT NULL,  
    FOREIGN KEY (order_id) REFERENCES orders(id)  
);
```

## 3.2 Cập nhật ProductController

Cập nhật ProductController để xử lý các yêu cầu liên quan đến giỏ hàng, bao gồm thêm sản phẩm vào giỏ hàng, xóa sản phẩm khỏi giỏ hàng và cập nhật số lượng sản phẩm.

Thêm phương thức để xử lý việc thêm sản phẩm vào giỏ hàng và đặt hàng

ProductController.php

Thêm phương thức cart, addToCart, checkout và processCheckout.

```
<?php  
// Require SessionHelper and other necessary files  
require_once('app/config/database.php');  
require_once('app/models/ProductModel.php');  
require_once('app/models/CategoryModel.php');  
  
class ProductController  
{  
    private $productModel;  
    private $db;  
  
    public function __construct()  
    {  
        $this->db = (new Database())->getConnection();  
        $this->productModel = new ProductModel($this->db);  
    }  
}
```

```
}

public function index()
{
    $products = $this->productModel->getProducts();
    include 'app/views/product/list.php';
}

public function show($id)
{
    $product = $this->productModel->getProductById($id);

    if ($product) {
        include 'app/views/product/show.php';
    } else {
        echo "Không thấy sản phẩm.";
    }
}

public function add()
{
    $categories = (new CategoryModel($this->db))->getCategories();
    include_once 'app/views/product/add.php';
}

public function save()
{
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $name = $_POST['name'] ?? '';
        $description = $_POST['description'] ?? '';
        $price = $_POST['price'] ?? '';
        $category_id = $_POST['category_id'] ?? null;

        if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
            $image = $this->uploadImage($_FILES['image']);
        } else {
            $image = "";
        }

        $result = $this->productModel->addProduct($name, $description, $price,
$category_id, $image);

        if (is_array($result)) {
            $errors = $result;
            $categories = (new CategoryModel($this->db))->getCategories();
            include 'app/views/product/add.php';
        } else {
```

```
        header('Location: /webbanhang/Product');
    }
}

public function edit($id)
{
    $product = $this->productModel->getProductById($id);
    $categories = (new CategoryModel($this->db))->getCategories();

    if ($product) {
        include 'app/views/product/edit.php';
    } else {
        echo "Không thấy sản phẩm.";
    }
}

public function update()
{
    if ($_SERVER['REQUEST_METHOD'] === 'POST') {
        $id = $_POST['id'];
        $name = $_POST['name'];
        $description = $_POST['description'];
        $price = $_POST['price'];
        $category_id = $_POST['category_id'];

        if (isset($_FILES['image']) && $_FILES['image']['error'] == 0) {
            $image = $this->uploadImage($_FILES['image']);
        } else {
            $image = $_POST['existing_image'];
        }

        $edit = $this->productModel->updateProduct($id, $name, $description,
$price, $category_id, $image);

        if ($edit) {
            header('Location: /webbanhang/Product');
        } else {
            echo "Đã xảy ra lỗi khi lưu sản phẩm.";
        }
    }
}

public function delete($id)
{
    if ($this->productModel->deleteProduct($id)) {
```

```
        header('Location: /webbanhang/Product');
    } else {
        echo "Đã xảy ra lỗi khi xóa sản phẩm.";
    }
}

private function uploadImage($file)
{
    $target_dir = "uploads/";

    // Kiểm tra và tạo thư mục nếu chưa tồn tại
    if (!is_dir($target_dir)) {
        mkdir($target_dir, 0777, true);
    }

    $target_file = $target_dir . basename($file["name"]);
    $imageFileType = strtolower(pathinfo($target_file, PATHINFO_EXTENSION));

    // Kiểm tra xem file có phải là hình ảnh không
    $check = getimagesize($file["tmp_name"]);
    if ($check === false) {
        throw new Exception("File không phải là hình ảnh.");
    }

    // Kiểm tra kích thước file (10 MB = 10 * 1024 * 1024 bytes)
    if ($file["size"] > 10 * 1024 * 1024) {
        throw new Exception("Hình ảnh có kích thước quá lớn.");
    }

    // Chỉ cho phép một số định dạng hình ảnh nhất định
    if ($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType !=
"jpeg" && $imageFileType != "gif") {
        throw new Exception("Chỉ cho phép các định dạng JPG, JPEG, PNG và GIF.");
    }

    // Lưu file
    if (!move_uploaded_file($file["tmp_name"], $target_file)) {
        throw new Exception("Có lỗi xảy ra khi tải lên hình ảnh.");
    }

    return $target_file;
}

public function addToCart($id)
{
    $product = $this->productModel->getProductById($id);
    if (!$product) {
```

```
        echo "Không tìm thấy sản phẩm.";
        return;
    }

    if (!isset($_SESSION['cart'])) {
        $_SESSION['cart'] = [];
    }

    if (isset($_SESSION['cart'][$id])) {
        $_SESSION['cart'][$id]['quantity']++;
    } else {
        $_SESSION['cart'][$id] = [
            'name' => $product->name,
            'price' => $product->price,
            'quantity' => 1,
            'image' => $product->image
        ];
    }

    header('Location: /webbanhang/Product/cart');
}

public function cart()
{
    $cart = isset($_SESSION['cart']) ? $_SESSION['cart'] : [];
    include 'app/views/product/cart.php';
}

public function checkout()
{
    include 'app/views/product/checkout.php';
}

public function processCheckout()
{
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $name = $_POST['name'];
        $phone = $_POST['phone'];
        $address = $_POST['address'];

        // Kiểm tra giỏ hàng
        if (!isset($_SESSION['cart']) || empty($_SESSION['cart'])) {
            echo "Giỏ hàng trống.";
            return;
        }

        // Bắt đầu giao dịch
```

```
$this->db->beginTransaction();

try {
    // Lưu thông tin đơn hàng vào bảng orders
    $query = "INSERT INTO orders (name, phone, address) VALUES (:name,
:phone, :address)";
    $stmt = $this->db->prepare($query);
    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':phone', $phone);
    $stmt->bindParam(':address', $address);
    $stmt->execute();
    $order_id = $this->db->lastInsertId();

    // Lưu chi tiết đơn hàng vào bảng order_details
    $cart = $_SESSION['cart'];
    foreach ($cart as $product_id => $item) {
        $query = "INSERT INTO order_details (order_id, product_id,
quantity, price) VALUES (:order_id, :product_id, :quantity, :price)";
        $stmt = $this->db->prepare($query);
        $stmt->bindParam(':order_id', $order_id);
        $stmt->bindParam(':product_id', $product_id);
        $stmt->bindParam(':quantity', $item['quantity']);
        $stmt->bindParam(':price', $item['price']);
        $stmt->execute();
    }

    // Xóa giỏ hàng sau khi đặt hàng thành công
    unset($_SESSION['cart']);

    // Commit giao dịch
    $this->db->commit();

    // Chuyển hướng đến trang xác nhận đơn hàng
    header('Location: /webbanhang/Product/orderConfirmation');
} catch (Exception $e) {
    // Rollback giao dịch nếu có lỗi
    $this->db->rollBack();
    echo "Đã xảy ra lỗi khi xử lý đơn hàng: " . $e->getMessage();
}
}

public function orderConfirmation()
{
    include 'app/views/product/orderConfirmation.php';
}
}
```

?>



## 3.3 Tạo các views tương ứng

File Cart.php

```
<?php include 'app/views/shares/header.php'; ?>

<h1>Giỏ hàng</h1>

<?php if (!empty($cart)): ?>
    <ul class="list-group">
        <?php foreach ($cart as $id => $item): ?>
            <li class="list-group-item">
                <h2><?php echo htmlspecialchars($item['name'], ENT_QUOTES, 'UTF-8');
?></h2>

                <?php if ($item['image']): ?>
                    
                <?php endif; ?>
                <p>Giá: <?php echo htmlspecialchars($item['price'], ENT_QUOTES, 'UTF-
8'); ?> VND</p>
                <p>Số lượng: <?php echo htmlspecialchars($item['quantity'],
ENT_QUOTES, 'UTF-8'); ?></p>
            </li>
        <?php endforeach; ?>
    </ul>
<?php else: ?>
    <p>Giỏ hàng của bạn đang trống.</p>
<?php endif; ?>

<a href="/webbanhang/Product" class="btn btn-secondary mt-2">Tiếp tục mua sắm</a>
<a href="/webbanhang/Product/checkout" class="btn btn-secondary mt-2">Thanh Toán</a>

<?php include 'app/views/shares/footer.php'; ?>
```

Checkout.php

```
<?php include 'app/views/shares/header.php'; ?>

<h1>Thanh toán</h1>

<form method="POST" action="/webbanhang/Product/processCheckout">
    <div class="form-group">
        <label for="name">Họ tên:</label>
        <input type="text" id="name" name="name" class="form-control" required>
    </div>
    <div class="form-group">
```

```

        <label for="phone">Số điện thoại:</label>
        <input type="text" id="phone" name="phone" class="form-control" required>
    </div>
    <div class="form-group">
        <label for="address">Địa chỉ:</label>
        <textarea id="address" name="address" class="form-control"
required></textarea>
    </div>
    <button type="submit" class="btn btn-primary">Thanh toán</button>
</form>

<a href="/webbanhang/Product/cart" class="btn btn-secondary mt-2">Quay lại giỏ
hàng</a>

<?php include 'app/views/shares/footer.php'; ?>

```

### orderConfirmation.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Xác nhận đơn hàng</h1>
<p>Cảm ơn bạn đã đặt hàng. Đơn hàng của bạn đã được xử lý thành công.</p>
<a href="/webbanhang/Product/list" class="btn btn-primary mt-2">Tiếp tục mua sắm</a>

<?php include 'app/views/shares/footer.php'; ?>

```

### Cập nhật list.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Danh sách sản phẩm</h1>
<a href="/webbanhang/Product/add" class="btn btn-success mb-2">Thêm sản phẩm mới</a>
<ul class="list-group">
    <?php foreach ($products as $product): ?>
        <li class="list-group-item">
            <h2><a href="/webbanhang/Product/show/<?php echo $product->id; ?>"><?php
echo htmlspecialchars($product->name, ENT_QUOTES, 'UTF-8'); ?></a></h2>
            <?php if ($product->image): ?>
                
            <?php endif; ?>
            <p><?php echo htmlspecialchars($product->description, ENT_QUOTES, 'UTF-
8'); ?></p>
            <p>Giá: <?php echo htmlspecialchars($product->price, ENT_QUOTES, 'UTF-8');
?> VND</p>
        </li>
    </?php ?>
</ul>

```

```

        <p>Danh mục: <?php echo htmlspecialchars($product->category_name,
ENT_QUOTES, 'UTF-8'); ?></p>
        <a href="/webbanhang/Product/edit/<?php echo $product->id; ?>" class="btn
btn-warning">Sửa</a>
        <a href="/webbanhang/Product/delete/<?php echo $product->id; ?>"
class="btn btn-danger" onclick="return confirm('Bạn có chắc chắn muốn xóa sản phẩm
này?');">Xóa</a>
        <a href="/webbanhang/Product/addToCart/<?php echo $product->id; ?>"
class="btn btn-primary">Thêm vào giỏ hàng</a>
    </li>
    <?php endforeach; ?>
</ul>

<?php include 'app/views/shares/footer.php'; ?>

```

### 3.4 Khởi tạo session

Đảm bảo rằng đã khởi tạo session ở đầu mỗi request:

Cập nhật file index.php

```
// Bắt đầu session ở đầu file index.php
```

```
session_start();
```

```

<?php
session_start();
require_once 'app/models/ProductModel.php';
// Product/add
$url = $_GET['url'] ?? '';
$url = rtrim($url, '/');
$url = filter_var($url, FILTER_SANITIZE_URL);
$url = explode('/', $url);

// Kiểm tra phần đầu tiên của URL để xác định controller
$controllerName = isset($url[0]) && $url[0] != '' ? ucfirst($url[0]) . 'Controller' :
'DefaultController';

// Kiểm tra phần thứ hai của URL để xác định action
$action = isset($url[1]) && $url[1] != '' ? $url[1] : 'index';

// die ("controller=$controllerName - action=$action");

// Kiểm tra xem controller và action có tồn tại không

```

```
if (!file_exists('app/controllers/' . $controllerName . '.php')) {  
    // Xử lý không tìm thấy controller  
    die('Controller not found');  
}  
  
require_once 'app/controllers/' . $controllerName . '.php';  
  
$controller = new $controllerName();  
  
if (!method_exists($controller, $action)) {  
    // Xử lý không tìm thấy action  
    die('Action not found');  
}  
  
// Gọi action với các tham số còn lại (nếu có)  
call_user_func_array([$controller, $action], array_slice($url, 2));
```

## 3.5 Tiến hành khởi chạy và kiểm tra kết quả:


Trang danh sách sản phẩm:

[Quản lý sản phẩm](#) [Danh sách sản phẩm](#) [Thêm sản phẩm](#)

# Danh sách sản phẩm

Thêm sản phẩm mới

## Laptop



A high-performance laptop

Giá: 999.99 VND

Danh mục: Electronics

Sửa

Xóa


Thêm vào giỏ hàng

Trang giỏ hàng:

[Quản lý sản phẩm](#) [Danh sách sản phẩm](#) [Thêm sản phẩm](#)

## Giỏ hàng

### Laptop



Giá: 999.99 VND

Số lượng: 2

[Tiếp tục mua sắm](#) [Thanh Toán](#)

Trang thanh toán:

[Quản lý sản phẩm](#) [Danh sách sản phẩm](#) [Thêm sản phẩm](#)

## Thanh toán

Họ tên:

Số điện thoại:

Địa chỉ:

Thanh toánQuay lại giỏ hàng

Xác nhận đơn hàng

[Quản lý sản phẩm](#) [Danh sách sản phẩm](#) [Thêm sản phẩm](#)

## Xác nhận đơn hàng

Cảm ơn bạn đã đặt hàng. Đơn hàng của bạn đã được xử lý thành công.

Tiếp tục mua sắm

Kiểm tra database:

Bảng order\_detail:

The screenshot shows the Laragon MySQL interface. On the left, the database structure is visible, with 'my\_store' selected. The 'order\_details' table is highlighted. On the right, the table's data is displayed. A red box highlights the first row of data, and a red arrow points to it with the text 'Chi tiết đơn hàng'.

id	order_id	product_id	quantity	price
1	3	56	2	999.99

Bảng order

Thông tin đơn hàng đã đặt thành công

The screenshot shows the Laragon MySQL interface. On the left, the database structure is visible, with 'my\_store' selected. The 'orders' table is highlighted. On the right, the table's data is displayed. A red box highlights the third row of data.

id	name	phone	address	created_at
1	Laptop Hutech 11231	0933929347	An Binh	2024-06-17 22:22:06
2	Hung Tran	098832131	Thủ Đức	2024-06-25 15:48:42
3	Nguyen Van A	0988321312	Thủ Đức	2024-06-25 16:27:31

# BÀI 4. XÂY DỰNG CHỨC NĂNG XÁC THỰC NGƯỜI DÙNG

Để xây dựng chức năng đăng ký, đăng nhập và đăng xuất, cần thực hiện các bước sau:

- Cấu hình cơ sở dữ liệu: Tạo bảng users để lưu trữ thông tin người dùng.
- Tạo các mô hình (models): Xử lý dữ liệu người dùng.
- Tạo các bộ điều khiển (controllers): Xử lý logic đăng ký, đăng nhập và đăng xuất.
- Tạo các trang hiển thị (views): Giao diện người dùng để đăng ký, đăng nhập và đăng xuất.

## 4.1 Cấu hình cơ sở dữ liệu

Tạo bảng users để lưu trữ thông tin người dùng.

SQL Script để tạo bảng users

```
CREATE TABLE users (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(255) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
);
```

## 4.2 Tạo các mô hình (models)

Tạo mô hình UserModel để xử lý dữ liệu người dùng.

app/models/AccountModel.php

```
<?php  
class AccountModel  
{  
    private $conn;  
  
    private $table_name = "account";  
  
    public function __construct($db)  
    {  
        $this->conn = $db;  
    }  
}
```



```
public function getAccountByUsername($username)
{
    $query = "SELECT * FROM account WHERE username = :username";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(':username', $username, PDO::PARAM_STR);
    $stmt->execute();
    $result = $stmt->fetch(PDO::FETCH_OBJ);
    return $result;
}

function save($username, $name, $password, $role="user"){

    $query = "INSERT INTO " . $this->table_name . "(username, password, role)
VALUES (:username,:password, :role)";

    $stmt = $this->conn->prepare($query);

    // Làm sạch dữ liệu
    $name = htmlspecialchars(strip_tags($name));
    $username = htmlspecialchars(strip_tags($username));

    // Gán dữ liệu vào câu lệnh

    $stmt->bindParam(':username', $username);
    $stmt->bindParam(':password', $password);
    $stmt->bindParam(':role', $role);

    // Thực thi câu lệnh
    if ($stmt->execute()) {
        return true;
    }

    return false;
}
}
```

## 4.3 Tạo các điều khiển (controllers)

---

Tạo bộ điều khiển AccountController để xử lý logic đăng ký, đăng nhập và đăng xuất.

app/controllers/AccountController.php

```
<?php
require_once('app/config/database.php');
require_once('app/models/AccountModel.php');
```

```
class AccountController {
    private $accountModel;
    private $db;
    public function __construct() {
        $this->db = (new Database())->getConnection();
        $this->accountModel = new AccountModel($this->db);
    }

    function register(){
        include_once 'app/views/account/register.php';
    }
    public function login() {
        include_once 'app/views/account/login.php';
    }

    function save(){

        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            $username = $_POST['username'] ?? '';
            $fullName = $_POST['fullname'] ?? '';
            $password = $_POST['password'] ?? '';
            $confirmPassword = $_POST['confirmpassword'] ?? '';

            $errors = [];
            if(empty($username)){
                $errors['username'] = "Vui long nhap userName!";
            }
            if(empty($fullName)){
                $errors['fullname'] = "Vui long nhap fullName!";
            }
            if(empty($password)){
                $errors['password'] = "Vui long nhap password!";
            }
            if($password != $confirmPassword){
                $errors['confirmPass'] = "Mat khau va xac nhan chua dung";
            }
            //kiểm tra username đã được đăng ký chưa?
            $account = $this->accountModel->getAccountByUsername($username);

            if($account){
                $errors['account'] = "Tai khoan nay da co nguoi dang ky!";
            }

            if(count($errors) > 0){
                include_once 'app/views/account/register.php';
            }else{
                $password = password_hash($password, PASSWORD_BCRYPT, ['cost' => 12]);
            }
        }
    }
}
```

```

        $result = $this->accountModel->save($username, $fullName, $password);

        if($result){
            header('Location: /webbanhang/account/login');
        }
    }
}

function logout(){

    unset($_SESSION['username']);
    unset($_SESSION['role']);

    header('Location: /webbanhang/product');
}

public function checkLogin(){
    // Kiểm tra xem liệu form đã được submit
    if ($_SERVER['REQUEST_METHOD'] == 'POST') {
        $username = $_POST['username'] ?? '';
        $password = $_POST['password'] ?? '';

        $account = $this->accountModel->getAccountByUserName($username);
        if ($account) {
            $pwd_hashed = $account->password;
            //check mật khẩu
            if (password_verify($password, $pwd_hashed)) {

                session_start();

                // $_SESSION['user_id'] = $account->id;
                // $_SESSION['user_role'] = $account->role;
                $_SESSION['username'] = $account->username;

                header('Location: /webbanhang/product');
                exit;
            }
            else {
                echo "Password incorrect.";
            }
        } else {
            echo "Bao loi khong tim thay tai khoan";
        }
    }
}
}

```

## 4.4 Tạo các trang hiển thị (views)

Tạo các trang giao diện người dùng cho đăng ký, đăng nhập và đăng xuất.

Trong thư mục views tạo một thư mục account để chứa các file view tương ứng:

app/views/account/register.php

```
<?php include 'app/views/shares/header.php'; ?>
<?php

if (isset($errors)) {
    echo "<ul>";
    foreach ($errors as $err) {
        echo "<li class='text-danger'>$err</li>";
    }
    echo "</ul>";
}

?>

<div class="card-body p-5 text-center">
    <form class="user" action="/webbanhang/account/save" method="post">
        <div class="form-group row">
            <div class="col-sm-6 mb-3 mb-sm-0">
                <input type="text" class="form-control form-control-user"
id="username" name="username" placeholder="username">
            </div>
            <div class="col-sm-6">
                <input type="text" class="form-control form-control-user"
id="fullname" name="fullname" placeholder="fullname">
            </div>
        </div>
        <div class="form-group row">
            <div class="col-sm-6 mb-3 mb-sm-0">
                <input type="password" class="form-control form-control-user"
id="password" name="password" placeholder="password">
            </div>
            <div class="col-sm-6">
                <input type="password" class="form-control form-control-user"
id="confirmpassword" name="confirmpassword" placeholder="confirmpassword">
            </div>
        </div>
        <div class="form-group text-center">
            <button class="btn btn-primary btn-icon-split p-3">
                Register
            </button>
        </div>
    </form>
</div>
```

```

        </button>
    </div>
</form>

</div>
<?php include 'app/views/shares/footer.php'; ?>

```

app/views/account/login.php

```

<?php include 'app/views/shares/header.php'; ?>

<section class="vh-100 gradient-custom">
    <div class="container py-5 h-100">
        <div class="row d-flex justify-content-center align-items-center h-100">
            <div class="col-12 col-md-8 col-lg-6 col-xl-5">
                <div class="card bg-dark text-white" style="border-radius: 1rem;">
                    <div class="card-body p-5 text-center">

                        <form action="/webbanhang/account/checklogin" method="post">

                            <div class="mb-md-5 mt-md-4 pb-5">

                                <h2 class="fw-bold mb-2 text-uppercase">Login</h2>
                                <p class="text-white-50 mb-5">Please enter your login and password!</p>

                                <div class="form-outline form-white mb-4">
                                    <input type="text" name="username" class="form-control form-control-
lg" />
                                    <label class="form-label" for="typeEmailX">UserName</label>
                                </div>

                                <div class="form-outline form-white mb-4">
                                    <input type="password" name="password" class="form-control form-
control-lg" />
                                    <label class="form-label" for="typePasswordX">Password</label>
                                </div>

                                <p class="small mb-5 pb-lg-2"><a class="text-white-50" href="#">Forgot
password?</a></p>

                                <button class="btn btn-outline-light btn-lg px-5"
type="submit">Login</button>

                                <div class="d-flex justify-content-center text-center mt-4 pt-1">

```

```

        <a href="#" class="text-white"><i class="fab fa-facebook-f fa-
lg"></i></a>
        <a href="#" class="text-white"><i class="fab fa-twitter fa-lg mx-4
px-2"></i></a>
        <a href="#" class="text-white"><i class="fab fa-google fa-
lg"></i></a>
    </div>

</div>

<div>
    <p class="mb-0">Don't have an account? <a href="
/webbanhang/account/register " class="text-white-50 fw-bold">Sign Up</a>
    </p>
</div>
</form>

</div>
</div>
</div>
</div>
</div>
</section>

<?php include 'app/views/shares/footer.php'; ?>

```

Cập nhật file header.php

Trong header.php, sử dụng SessionHelper để hiển thị trạng thái đăng nhập/đăng xuất.

app/views/shares/header.php

```

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

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Quản lý sản phẩm</title>
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        .product-image {
            max-width: 100px;
            height: auto;

```

```

    }
    </style>
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Quản lý sản phẩm</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav"
            aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="/webbanhang/Product/">Danh sách sản
phẩm</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/webbanhang/Product/add">Thêm sản
phẩm</a>
                </li>
                <li class="nav-item">
                    <?php
                        if(SessionHelper::isLoggedIn()){
                            echo "<a class='nav-
link'>".$_SESSION['username']."'</a>";
                        }
                        else{
                            echo "<a class='nav-link'
href='/webbanhang/account/login'>Login</a>";
                        }
                    ?>
                </li>
                <li class="nav-item">
                    <a>
                        <?php
                            if(SessionHelper::isLoggedIn()){
                                echo "<a class='nav-link'
href='/webbanhang/account/logout'>Logout</a>";
                            }
                        ?>
                    </a>
                </li>
            </ul>
        </div>
    </nav>

```

```
        </ul>
    </div>
</nav>
<div class="container mt-4">
```

## 4.5 Tạo file SessionHelper

Sử dụng SessionHelper trong hệ thống để quản lý trạng thái đăng nhập và quyền của người dùng

Trong thư mục app tạo thư mục helpers để chứa file SessionHelper:

```
<?php

class SessionHelper {
    public static function isLoggedIn() {
        return isset($_SESSION['username']);
    }

    public static function isAdmin() {
        return isset($_SESSION['username']) && $_SESSION['user_role'] === 'admin';
    }
}
```

## 4.6 Cập nhật hệ thống định tuyến (routing)

Cập nhật index.php để định tuyến các yêu cầu đến các bộ điều khiển tương ứng.

index.php

```
<?php
session_start();
require_once 'app/models/ProductModel.php';
require_once 'app/helpers/SessionHelper.php';
// Product/add
$url = $_GET['url'] ?? '';
$url = rtrim($url, '/');
$url = filter_var($url, FILTER_SANITIZE_URL);
$url = explode('/', $url);

// Kiểm tra phần đầu tiên của URL để xác định controller
$controllerName = isset($url[0]) && $url[0] != '' ? ucfirst($url[0]) . 'Controller' :
'DefaultController';

// Kiểm tra phần thứ hai của URL để xác định action
```



```
$action = isset($url[1]) && $url[1] != '' ? $url[1] : 'index';

// die ("controller=$controllerName - action=$action");

// Kiểm tra xem controller và action có tồn tại không
if (!file_exists('app/controllers/' . $controllerName . '.php')) {
    // Xử lý không tìm thấy controller
    die('Controller not found');
}

require_once 'app/controllers/' . $controllerName . '.php';

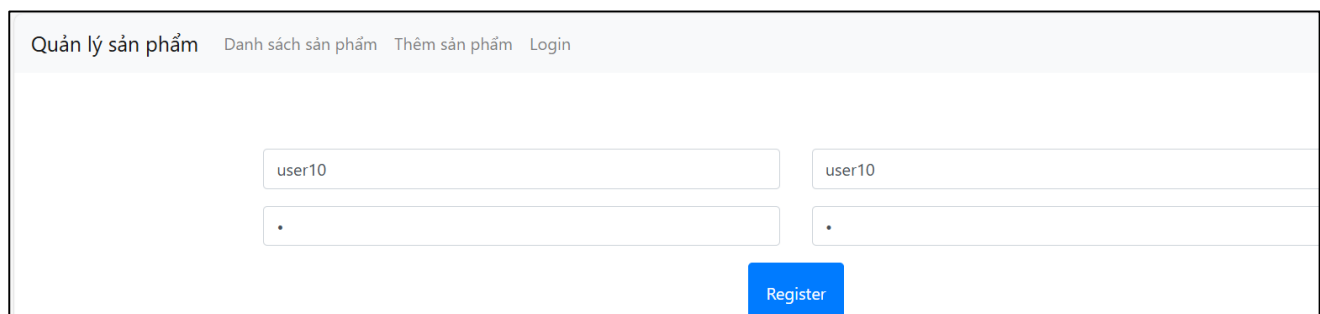
$controller = new $controllerName();

if (!method_exists($controller, $action)) {
    // Xử lý không tìm thấy action
    die('Action not found');
}

// Gọi action với các tham số còn lại (nếu có)
call_user_func_array([$controller, $action], array_slice($url, 2));
```

### Khởi chạy trang web và kiểm tra kết quả:

Trang đăng ký:



Trang đăng nhập:

Quản lý sản phẩm   Danh sách sản phẩm   Thêm sản phẩm   Login

### LOGIN

Please enter your login and password!

UserName

Password

[Forgot password?](#)

Login

[Don't have an account? Sign Up](#)

Kết quả đăng nhập thành công:


← ↻ ⓘ localhost:90/webbanhang/product

Quản lý sản phẩm   Danh sách sản phẩm   Thêm sản phẩm   **user10**   Logout

## Danh sách sản phẩm

Thêm sản phẩm mới

### Laptop



A high-performance laptop

Giá: 999.99 VND

Danh mục: Electronics

Sửa

Xóa

Thêm vào giỏ hàng

**KẾT QUẢ ĐĂNG NHẬP THÀNH CÔNG**

Kiểm tra database:

Laragon.MySQL		my_store.account: 4 rows total (approximately)			
>	information_s...				
▼	my_store	112.0 KiB			
	account	16.0 KiB			
	category	16.0 KiB			
	orders	16.0 KiB			
	order_details	32.0 KiB			
	product	32.0 KiB			

id	username	password	role
1	1	1	1
2	2	2	2
3	3	\$2y\$12\$2SIEloxQJgKyFkRzJ22H9uExSUIINKz0ILf...	user
4	user10	\$2y\$12\$Sx41M98dc5FEdpNQtiMJWe20gDSiqV...	user

**Tài khoản tạo thành công**

# BÀI 5. XÂY DỰNG RESTful API

Sau khi học xong bài này, sinh viên có thể nắm được:

- **Khái niệm về RESTful API:** Hiểu các nguyên tắc cơ bản của REST và cách thức hoạt động của RESTful API.
- **Cấu hình API trong PHP:** Biết cách cấu hình và triển khai API sử dụng PHP.
- **Xây dựng các phương thức CRUD cho API:** Tạo các phương thức GET, POST, PUT, DELETE để quản lý tài nguyên.
- **Tạo và quản lý endpoints:** Thiết kế và triển khai các endpoints API cho các chức năng khác nhau.
- **Kiểm thử API:** Sử dụng Postman hoặc công cụ tương tự để kiểm thử các API đã triển khai.

## 5.1 Cập nhật ProductModel

Bỏ thuộc tính image của product

```
<?php
class ProductModel
{
    private $conn;
    private $table_name = "product";

    public function __construct($db)
    {
        $this->conn = $db;
    }

    public function getProducts()
    {
        $query = "SELECT p.id, p.name, p.description, p.price, c.name as category_name
                  FROM " . $this->table_name . " p
                  LEFT JOIN category c ON p.category_id = c.id";
        $stmt = $this->conn->prepare($query);
        $stmt->execute();
        $result = $stmt->fetchAll(PDO::FETCH_OBJ);
        return $result;
    }

    public function getProductById($id)
```

```
{
    $query = "SELECT * FROM " . $this->table_name . " WHERE id = :id";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(':id', $id);
    $stmt->execute();
    $result = $stmt->fetch(PDO::FETCH_OBJ);
    return $result;
}

public function addProduct($name, $description, $price, $category_id)
{
    $errors = [];
    if (empty($name)) {
        $errors['name'] = 'Tên sản phẩm không được để trống';
    }
    if (empty($description)) {
        $errors['description'] = 'Mô tả không được để trống';
    }
    if (!is_numeric($price) || $price < 0) {
        $errors['price'] = 'Giá sản phẩm không hợp lệ';
    }
    if (count($errors) > 0) {
        return $errors;
    }

    $query = "INSERT INTO " . $this->table_name . " (name, description, price,
category_id) VALUES (:name, :description, :price, :category_id)";
    $stmt = $this->conn->prepare($query);

    $name = htmlspecialchars(strip_tags($name));
    $description = htmlspecialchars(strip_tags($description));
    $price = htmlspecialchars(strip_tags($price));
    $category_id = htmlspecialchars(strip_tags($category_id));

    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':description', $description);
    $stmt->bindParam(':price', $price);
    $stmt->bindParam(':category_id', $category_id);

    if ($stmt->execute()) {
        return true;
    }

    return false;
}
```

```
public function updateProduct($id, $name, $description, $price, $category_id)
{
    $query = "UPDATE " . $this->table_name . " SET name=:name,
description=:description, price=:price, category_id=:category_id WHERE id=:id";
    $stmt = $this->conn->prepare($query);

    $name = htmlspecialchars(strip_tags($name));
    $description = htmlspecialchars(strip_tags($description));
    $price = htmlspecialchars(strip_tags($price));
    $category_id = htmlspecialchars(strip_tags($category_id));

    $stmt->bindParam(':id', $id);
    $stmt->bindParam(':name', $name);
    $stmt->bindParam(':description', $description);
    $stmt->bindParam(':price', $price);
    $stmt->bindParam(':category_id', $category_id);

    if ($stmt->execute()) {
        return true;
    }
    return false;
}

public function deleteProduct($id)
{
    $query = "DELETE FROM " . $this->table_name . " WHERE id=:id";
    $stmt = $this->conn->prepare($query);
    $stmt->bindParam(':id', $id);
    if ($stmt->execute()) {
        return true;
    }
    return false;
}
}
?>
```

## 5.2 Xây dựng các Controller tương ứng

Xây dựng file ProductApiController.php

```
<?php
require_once('app/config/database.php');
require_once('app/models/ProductModel.php');
require_once('app/models/CategoryModel.php');
```

```
class ProductApiController
{
    private $productModel;
    private $db;

    public function __construct()
    {
        $this->db = (new Database())->getConnection();
        $this->productModel = new ProductModel($this->db);
    }

    // Lấy danh sách sản phẩm
    public function index()
    {
        header('Content-Type: application/json');
        $products = $this->productModel->getProducts();
        echo json_encode($products);
    }

    // Lấy thông tin sản phẩm theo ID
    public function show($id)
    {
        header('Content-Type: application/json');
        $product = $this->productModel->getProductById($id);
        if ($product) {
            echo json_encode($product);
        } else {
            http_response_code(404);
            echo json_encode(['message' => 'Product not found']);
        }
    }

    // Thêm sản phẩm mới
    public function store()
    {
        header('Content-Type: application/json');
        $data = json_decode(file_get_contents("php://input"), true);

        $name = $data['name'] ?? '';
        $description = $data['description'] ?? '';
        $price = $data['price'] ?? '';
        $category_id = $data['category_id'] ?? null;

        $result = $this->productModel->addProduct($name, $description, $price,
        $category_id, null);
    }
}
```

```
        if (is_array($result)) {
            http_response_code(400);
            echo json_encode(['errors' => $result]);
        } else {
            http_response_code(201);
            echo json_encode(['message' => 'Product created successfully']);
        }
    }

    // Cập nhật sản phẩm theo ID
    public function update($id)
    {
        header('Content-Type: application/json');
        $data = json_decode(file_get_contents("php://input"), true);

        $name = $data['name'] ?? '';
        $description = $data['description'] ?? '';
        $price = $data['price'] ?? '';
        $category_id = $data['category_id'] ?? null;

        $result = $this->productModel->updateProduct($id, $name, $description, $price,
        $category_id, null);

        if ($result) {
            echo json_encode(['message' => 'Product updated successfully']);
        } else {
            http_response_code(400);
            echo json_encode(['message' => 'Product update failed']);
        }
    }

    // Xóa sản phẩm theo ID
    public function destroy($id)
    {
        header('Content-Type: application/json');
        $result = $this->productModel->deleteProduct($id);

        if ($result) {
            echo json_encode(['message' => 'Product deleted successfully']);
        } else {
            http_response_code(400);
            echo json_encode(['message' => 'Product deletion failed']);
        }
    }
}
?>
```



### Xây dựng file CategoryApiController.php

```
<?php
require_once('app/config/database.php');
require_once('app/models/CategoryModel.php');

class CategoryApiController
{
    private $categoryModel;
    private $db;

    public function __construct()
    {
        $this->db = (new Database())->getConnection();
        $this->categoryModel = new CategoryModel($this->db);
    }

    // Lấy danh sách danh mục
    public function index()
    {
        header('Content-Type: application/json');
        $categories = $this->categoryModel->getCategories();
        echo json_encode($categories);
    }
}
?>
```

## 5.3 Cấu hình router để định tuyến các yêu cầu API

```
<?php
session_start();
require_once 'app/models/ProductModel.php';
require_once 'app/helpers/SessionHelper.php';

require_once 'app/controllers/ProductApiController.php';
require_once 'app/controllers/CategoryApiController.php';
// Start session

$url = $_GET['url'] ?? '';
$url = rtrim($url, '/');
$url = filter_var($url, FILTER_SANITIZE_URL);
$url = explode('/', $url);
```

```
// Kiểm tra phần đầu tiên của URL để xác định controller
$controllerName = isset($url[0]) && $url[0] != '' ? ucfirst($url[0]) . 'Controller' :
'DefaultController';

// Kiểm tra phần thứ hai của URL để xác định action
$action = isset($url[1]) && $url[1] != '' ? $url[1] : 'index';

// Định tuyến các yêu cầu API
if ($controllerName === 'ApiController' && isset($url[1])) {
    $apiControllerName = ucfirst($url[1]) . 'ApiController';
    if (file_exists('app/controllers/' . $apiControllerName . '.php')) {
        require_once 'app/controllers/' . $apiControllerName . '.php';
        $controller = new $apiControllerName();

        $method = $_SERVER['REQUEST_METHOD'];
        $id = $url[2] ?? null;

        switch ($method) {
            case 'GET':
                if ($id) {
                    $action = 'show';
                } else {
                    $action = 'index';
                }
                break;
            case 'POST':
                $action = 'store';
                break;
            case 'PUT':
                if ($id) {
                    $action = 'update';
                }
                break;
            case 'DELETE':
                if ($id) {
                    $action = 'destroy';
                }
                break;
            default:
                http_response_code(405);
                echo json_encode(['message' => 'Method Not Allowed']);
                exit;
        }

        if (method_exists($controller, $action)) {
            if ($id) {
                call_user_func_array([$controller, $action], [$id]);
            }
        }
    }
}
```

```

        } else {
            call_user_func_array([$controller, $action], []);
        }
    } else {
        http_response_code(404);
        echo json_encode(['message' => 'Action not found']);
    }
    exit;
} else {
    http_response_code(404);
    echo json_encode(['message' => 'Controller not found']);
    exit;
}
}

// Tạo đối tượng controller tương ứng cho các yêu cầu không phải API
if (file_exists('app/controllers/' . $controllerName . '.php')) {
    require_once 'app/controllers/' . $controllerName . '.php';
    $controller = new $controllerName();
} else {
    die('Controller not found');
}

// Kiểm tra và gọi action
if (method_exists($controller, $action)) {
    call_user_func_array([$controller, $action], array_slice($url, 2));
} else {
    die('Action not found');
}
?>

```

## 5.4 Cập nhật views để quản lý sản phẩm

app/views/product/list.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Danh sách sản phẩm</h1>
<a href="/webbanhang/Product/add" class="btn btn-success mb-2">Thêm sản phẩm mới</a>
<ul class="list-group" id="product-list">
    <!-- Danh sách sản phẩm sẽ được tải từ API và hiển thị tại đây -->
</ul>

<?php include 'app/views/shares/footer.php'; ?>

<script>
document.addEventListener("DOMContentLoaded", function() {
    fetch('/webbanhang/api/product')

```

```

        .then(response => response.json())
        .then(data => {
            const productList = document.getElementById('product-list');
            data.forEach(product => {
                const productItem = document.createElement('li');
                productItem.className = 'list-group-item';
                productItem.innerHTML = `
                    <h2><a
href="/webbanhang/Product/show/${product.id}">${product.name}</a></h2>
                    <p>${product.description}</p>
                    <p>Giá: ${product.price} VND</p>
                    <p>Danh mục: ${product.category_name}</p>
                    <a href="/webbanhang/Product/edit/${product.id}" class="btn btn-
warning">Sửa</a>
                    <button class="btn btn-danger"
onclick="deleteProduct(${product.id})">Xóa</button>
                `;
                productList.appendChild(productItem);
            });
        });
    });

function deleteProduct(id) {
    if (confirm('Bạn có chắc chắn muốn xóa sản phẩm này?')) {
        fetch(`/webbanhang/api/product/${id}`, {
            method: 'DELETE'
        })
        .then(response => response.json())
        .then(data => {
            if (data.message === 'Product deleted successfully') {
                location.reload();
            } else {
                alert('Xóa sản phẩm thất bại');
            }
        });
    }
}
</script>

```

app/views/product/add.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Thêm sản phẩm mới</h1>
<form id="add-product-form">
    <div class="form-group">
        <label for="name">Tên sản phẩm:</label>

```

```

        <input type="text" id="name" name="name" class="form-control" required>
    </div>
    <div class="form-group">
        <label for="description">Mô tả:</label>
        <textarea id="description" name="description" class="form-control"
required></textarea>
    </div>
    <div class="form-group">
        <label for="price">Giá:</label>
        <input type="number" id="price" name="price" class="form-control" step="0.01"
required>
    </div>
    <div class="form-group">
        <label for="category_id">Danh mục:</label>
        <select id="category_id" name="category_id" class="form-control" required>
            <!-- Các danh mục sẽ được tải từ API và hiển thị tại đây -->
        </select>
    </div>
    <button type="submit" class="btn btn-primary">Thêm sản phẩm</button>
</form>

<a href="/webbanhang/Product/list" class="btn btn-secondary mt-2">Quay lại danh sách
sản phẩm</a>

<?php include 'app/views/shares/footer.php'; ?>

<script>
document.addEventListener("DOMContentLoaded", function() {
    fetch('/webbanhang/api/category')
        .then(response => response.json())
        .then(data => {
            const categorySelect = document.getElementById('category_id');
            data.forEach(category => {
                const option = document.createElement('option');
                option.value = category.id;
                option.textContent = category.name;
                categorySelect.appendChild(option);
            });
        });

    document.getElementById('add-product-form').addEventListener('submit',
function(event) {
    event.preventDefault();

    const formData = new FormData(this);
    const jsonData = {};
    formData.forEach((value, key) => {

```

```

        jsonData[key] = value;
    });

    fetch('/webbanhang/api/product', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(jsonData)
    })
    .then(response => response.json())
    .then(text => {
        console.log('Raw response:', text); // Log the raw response text
        try {
            const data = text;
            if (data.message === 'Product created successfully') {
                location.href = '/webbanhang/Product';
            } else {
                alert('Thêm sản phẩm thất bại');
            }
        } catch (error) {
            console.error('Error parsing JSON:', error);
            alert('Lỗi: Không thể phân tích JSON từ phản hồi của máy chủ.');
```

app/views/product/edit.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Sửa sản phẩm</h1>
<form id="edit-product-form">
    <input type="hidden" id="id" name="id">
    <div class="form-group">
        <label for="name">Tên sản phẩm:</label>
        <input type="text" id="name" name="name" class="form-control" required>
    </div>
    <div class="form-group">
        <label for="description">Mô tả:</label>
        <textarea id="description" name="description" class="form-control"
required></textarea>
    </div>

```

```
<div class="form-group">
  <label for="price">Giá:</label>
  <input type="number" id="price" name="price" class="form-control" step="0.01"
required>
</div>
<div class="form-group">
  <label for="category_id">Danh mục:</label>
  <select id="category_id" name="category_id" class="form-control" required>
    <!-- Các danh mục sẽ được tải từ API và hiển thị tại đây -->
  </select>
</div>
<button type="submit" class="btn btn-primary">Lưu thay đổi</button>
</form>

<a href="/webbanhang/Product/list" class="btn btn-secondary mt-2">Quay lại danh sách
sản phẩm</a>

<?php include 'app/views/shares/footer.php'; ?>

<script>
document.addEventListener("DOMContentLoaded", function() {
  // const urlParams = new URLSearchParams(window.location.search);
  const productId = <?=$editId ?>;

  fetch(`/webbanhang/api/product/${productId}`)
    .then(response => response.json())
    .then(data => {
      document.getElementById('id').value = data.id;
      document.getElementById('name').value = data.name;
      document.getElementById('description').value = data.description;
      document.getElementById('price').value = data.price;
      document.getElementById('category_id').value = data.category_id;
    });

  fetch('/webbanhang/api/category')
    .then(response => response.json())
    .then(data => {
      const categorySelect = document.getElementById('category_id');
      data.forEach(category => {
        const option = document.createElement('option');
        option.value = category.id;
        option.textContent = category.name;
        categorySelect.appendChild(option);
      });
    });
});
```

```
document.getElementById('edit-product-form').addEventListener('submit',
function(event) {
    event.preventDefault();

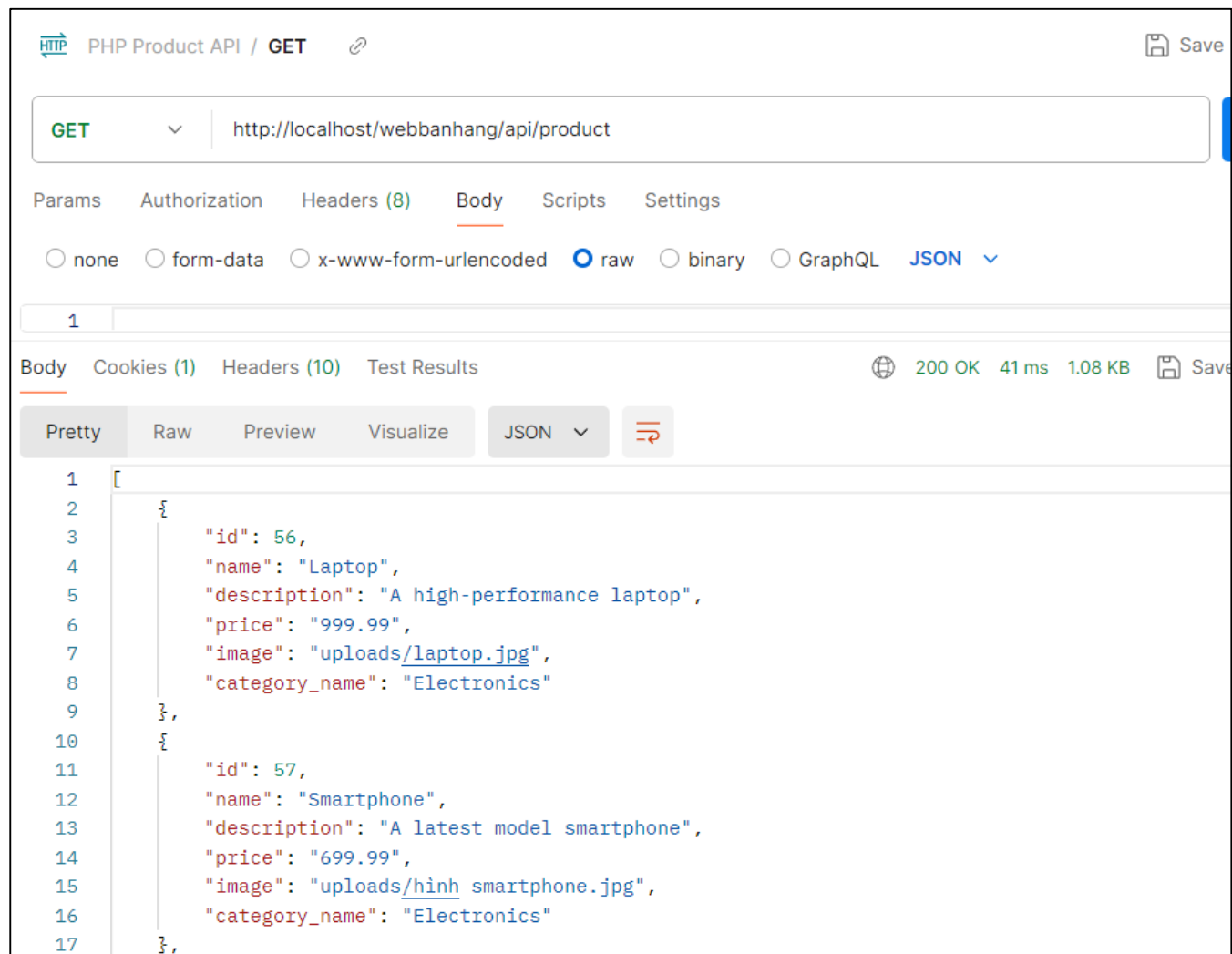
    const formData = new FormData(this);
    const jsonData = {};
    formData.forEach((value, key) => {
        jsonData[key] = value;
    });

    fetch(`/webbanhang/api/product/${jsonData.id}`, {
        method: 'PUT',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(jsonData)
    })
    .then(response => response.json())
    .then(data => {
        if (data.message === 'Product updated successfully') {
            location.href = '/webbanhang/Product';
        } else {
            alert('Cập nhật sản phẩm thất bại');
        }
    });
});
});
</script>
```



## 5.5 Tiến hành khởi chạy dự án và sử dụng postman để kiểm thử API

Phương thức GET



## Phương thức POST:

The screenshot displays a REST client interface for a "PHP Product API". The method is set to "POST" and the URL is "http://localhost/webbanhang/api/product". The "Body" tab is selected, showing a JSON payload with product details: name, description, price, and category\_id. The response status is 201 (Created), and the response body is a JSON object with a success message.

HTTP PHP Product API / POST Save

POST ▼ http://localhost/webbanhang/api/product

Params Authorization Headers (10) **Body ●** Scripts Settings

☐ none ☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary ☐ GraphQL **JSON ▼**

```
1 {
2   "name": "Laptop Hutech",
3   "description": "A high-performance laptop",
4   "price": "1999.99",
5   "category_id": 1
6 }
```

Body Cookies (1) Headers (10) Test Results 201 Created 15 ms 404 B Save

Pretty Raw Preview Visualize **JSON ▼** ↺

```
1 {
2   "message": "Product created successfully"
3 }
```

## Phương thức PUT:

The screenshot displays a REST client interface for a PUT request. The top bar shows the URL `http://localhost/webbanhang/api/product/73` and the method `PUT`. Below the URL bar, tabs for `Params`, `Authorization`, `Headers (10)`, `Body`, `Scripts`, and `Settings` are visible. The `Body` tab is active, showing a JSON payload with the following structure:

```
1 {
2   ... "name": "Laptop Hutech PUT",
3   ... "description": "A high-performance laptop",
4   ... "price": "1999.99",
5   ... "category_id": 1
6   ... }
```

Below the body tab, there are radio buttons for `none`, `form-data`, `x-www-form-urlencoded`, `raw` (selected), `binary`, and `GraphQL`. A dropdown menu shows `JSON` as the selected format. The response section at the bottom shows a `200 OK` status with a response time of `32 ms` and a size of `399 B`. The response body is displayed in the `Pretty` format, showing a JSON object:

```
1 {
2   "message": "Product updated successfully"
3 }
```

## Phương thức DELETE:

HTTP PHP Product API / DELETE Save

**DELETE** ▼ `http://localhost/webbanhang/api/product/75`

Params Authorization Headers (8) Body Scripts Settings

Query Params

	Key	Value	Description
	Key	Value	Description

Body Cookies (1) Headers (10) Test Results 200 OK 41 ms 399 B Save

Pretty Raw Preview Visualize JSON ▼ ≡

```
1 {  
2   "message": "Product deleted successfully"  
3 }
```

## 5.6 Yêu cầu bổ sung

Xây dựng trang front-end cho API bằng jquery.

## BÀI 6. Bảo mật RESTful API với JWT

Sau khi học xong bài này, sinh viên có thể nắm được:

- **Khái niệm về JWT (JSON Web Token):** Hiểu cơ bản về JWT, cách thức hoạt động và ứng dụng của nó trong bảo mật API.
- **Cài đặt thư viện JWT:** Biết cách cài đặt và cấu hình thư viện JWT trong dự án PHP.
- **Tạo và xác thực JWT:** Học cách tạo token JWT khi người dùng đăng nhập và xác thực token này trong các yêu cầu API.
- **Bảo vệ các endpoints API:** Triển khai bảo vệ các endpoints quan trọng bằng cách yêu cầu JWT hợp lệ cho mỗi yêu cầu.
- **Thực hành bảo mật API:** Hiểu và triển khai các biện pháp bảo mật nâng cao cho RESTful API sử dụng JWT.

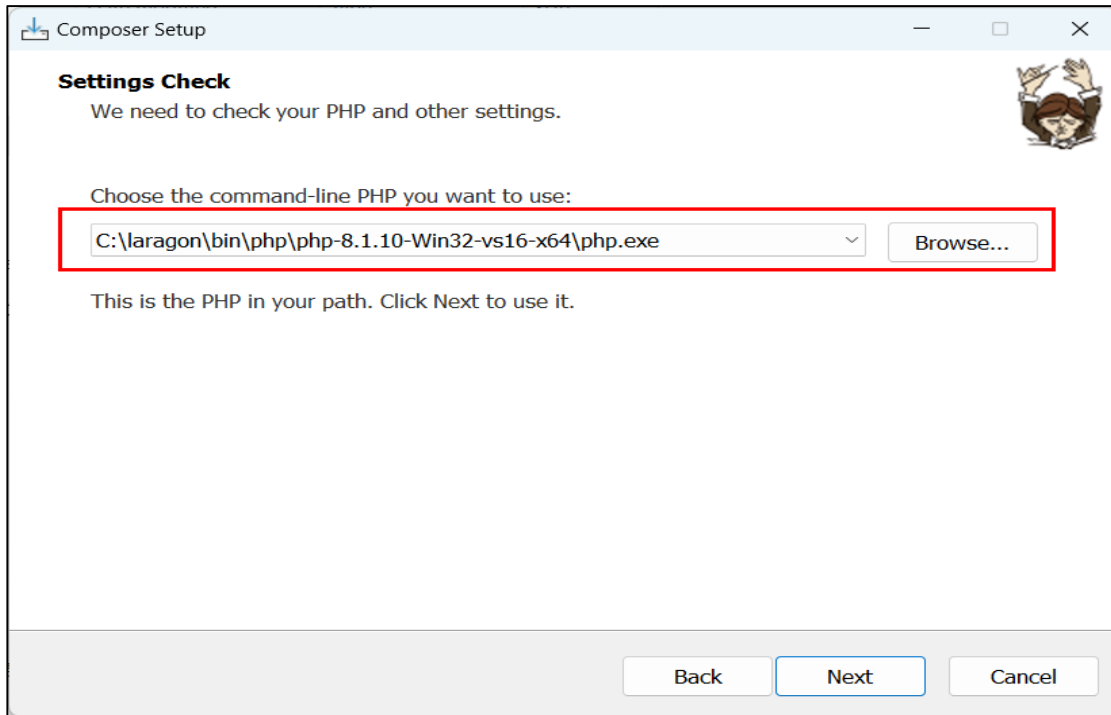
### 6.1 Cài đặt thư viện JWT

---

JSON Web Tokens (JWT) là một cách phổ biến để bảo mật API và xác thực người dùng. Trong hướng dẫn này, chúng ta sẽ sử dụng PHP để bảo mật RESTful API bằng JWT. Chúng ta sẽ sử dụng thư viện firebase/php-jwt để tạo và xác thực JWT.

Truy cập trang web: <https://getcomposer.org/download/> để tiến hành tải bản cài đặt

Các bước cài đặt tiến hành cài mặc định.



Vào terminal gõ composer để kiểm tra cài đặt thành công:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\winda> composer

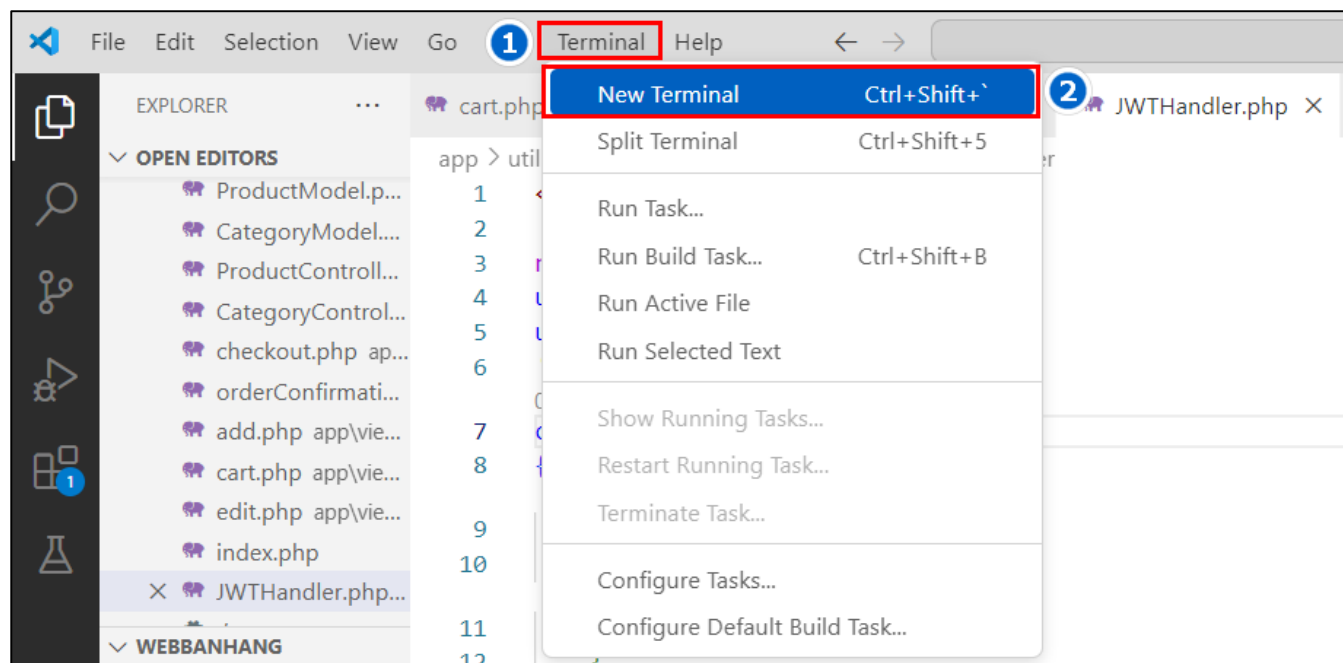
Composer version 2.7.7 2024-06-10 22:11:12

Usage:
  command [options] [arguments]

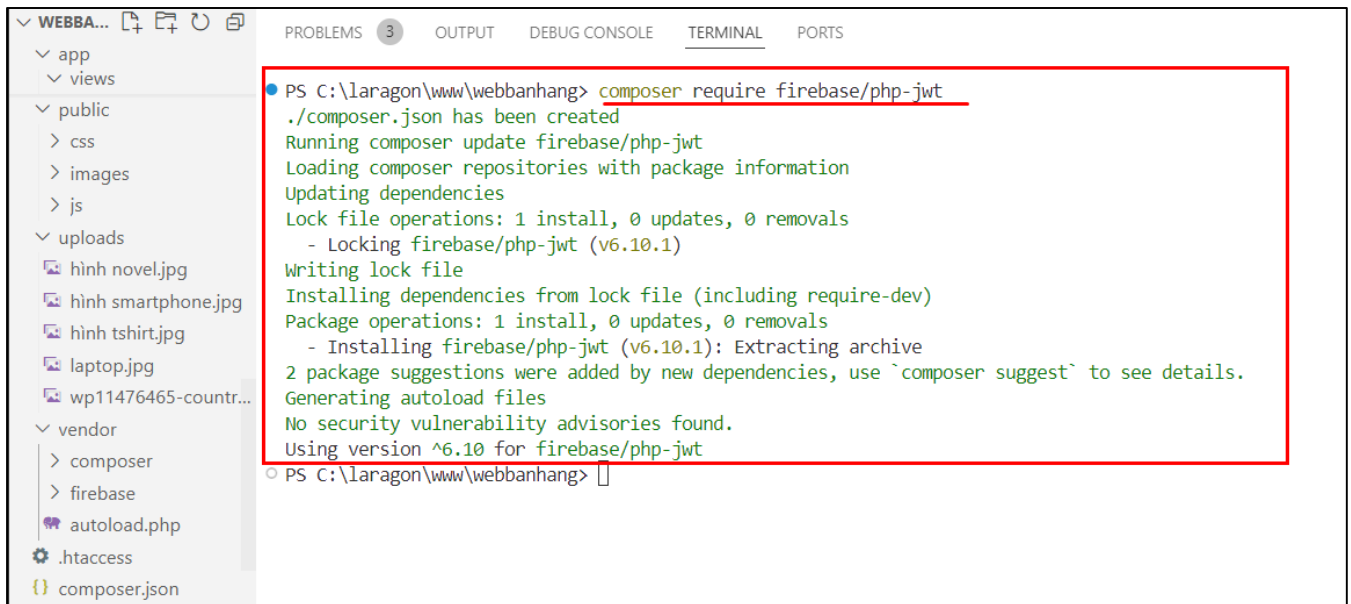
Options:
  -h, --help                Display help for the given command. When no command is given display help for the list
command
  -q, --quiet               Do not output any message
  -V, --version              Display this application version
  --ansi|--no-ansi          Force (or disable --no-ansi) ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                 Display timing and memory usage information
  --no-plugins              Whether to disable plugins.
  --no-scripts              Skips the execution of all scripts defined in composer.json file.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache                Prevent use of the cache
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and
```

Cài đặt composer vào dự án:

Trong cửa sổ visual code studio của dự án chọn thẻ Terminal và click New Terminal:



Nhập câu lệnh sau: `composer require firebase/php-jwt`



```
PS C:\laragon\www\webbanhang> composer require firebase/php-jwt
./composer.json has been created
Running composer update firebase/php-jwt
Loading composer repositories with package information
Updating dependencies
Lock file operations: 1 install, 0 updates, 0 removals
  - Locking firebase/php-jwt (v6.10.1)
Writing lock file
Installing dependencies from lock file (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
  - Installing firebase/php-jwt (v6.10.1): Extracting archive
2 package suggestions were added by new dependencies, use `composer suggest` to see details.
Generating autoload files
No security vulnerability advisories found.
Using version ^6.10 for firebase/php-jwt
PS C:\laragon\www\webbanhang>
```

## 6.2 Tạo lớp xử lý JWT

Tạo một lớp JWTHandler để tạo và xác thực JWT.

Trong thư mục app tạo thêm thư mục utils để chứa file JWTHandler.php

app/utils/JWTHandler.php

```
<?php

require_once 'vendor/autoload.php';
use \Firebase\JWT\JWT;
use \Firebase\JWT\Key;

class JWTHandler
{
    private $secret_key;

    public function __construct()
    {
        $this->secret_key = "HUTECH"; // Thay thế bằng khóa bí mật của bạn
    }

    // Tạo JWT
    public function encode($data)
    {
        $issuedAt = time();
        $expirationTime = $issuedAt + 3600; // jwt valid for 1 hour from the issued
time
```



```

        $payload = array(
            'iat' => $issuedAt,
            'exp' => $expirationTime,
            'data' => $data
        );

        return JWT::encode($payload, $this->secret_key, 'HS256');
    }

    // Giải mã JWT
    public function decode($jwt)
    {
        try {
            $decoded = JWT::decode($jwt, new Key($this->secret_key, 'HS256'));
            return (array) $decoded->data;
        } catch (Exception $e) {
            return null;
        }
    }
}
?>

```

## 6.3 Cập nhật API để sử dụng JWT

Cập nhật **ProductApiController** để bảo vệ các endpoint bằng JWT.

```

<?php
require_once('app/config/database.php');
require_once('app/models/ProductModel.php');
require_once('app/models/CategoryModel.php');

require_once('app/utils/JWTHandler.php'); //
class ProductApiController
{
    private $productModel;
    private $db;
    private $jwtHandler; //
    public function __construct()
    {
        $this->db = (new Database())->getConnection();
        $this->productModel = new ProductModel($this->db);

        $this->jwtHandler = new JWTHandler(); //
    }

    private function authenticate()
    {
        $headers = apache_request_headers();
    }
}

```

```
if (isset($headers['Authorization'])) {
    $authHeader = $headers['Authorization'];
    $arr = explode(" ", $authHeader);
    $jwt = $arr[1] ?? null;
    if ($jwt) {
        $decoded = $this->jwtHandler->decode($jwt);
        return $decoded ? true : false;
    }
}
return false;
}

// Lấy danh sách sản phẩm
public function index()
{
    if ($this->authenticate()) {
        header('Content-Type: application/json');
        $products = $this->productModel->getProducts();
        echo json_encode($products);
    } else {
        http_response_code(401);
        echo json_encode(['message' => 'Unauthorized']);
    }
}

// Lấy thông tin sản phẩm theo ID
public function show($id)
{
    header('Content-Type: application/json');
    $product = $this->productModel->getProductById($id);
    if ($product) {
        echo json_encode($product);
    } else {
        http_response_code(404);
        echo json_encode(['message' => 'Product not found']);
    }
}

// Thêm sản phẩm mới
public function store()
{
    header('Content-Type: application/json');
    $data = json_decode(file_get_contents("php://input"), true);
```

```
$name = $data['name'] ?? '';
$description = $data['description'] ?? '';
$price = $data['price'] ?? '';
$category_id = $data['category_id'] ?? null;

$result = $this->productModel->addProduct($name, $description, $price,
$category_id, null);

if (is_array($result)) {
    http_response_code(400);
    echo json_encode(['errors' => $result]);
} else {
    http_response_code(201);
    echo json_encode(['message' => 'Product created successfully']);
}
}

// Cập nhật sản phẩm theo ID
public function update($id)
{
    header('Content-Type: application/json');
    $data = json_decode(file_get_contents("php://input"), true);

    $name = $data['name'] ?? '';
    $description = $data['description'] ?? '';
    $price = $data['price'] ?? '';
    $category_id = $data['category_id'] ?? null;

    $result = $this->productModel->updateProduct($id, $name, $description, $price,
$category_id, null);

    if ($result) {
        echo json_encode(['message' => 'Product updated successfully']);
    } else {
        http_response_code(400);
        echo json_encode(['message' => 'Product update failed']);
    }
}

// Xóa sản phẩm theo ID
public function destroy($id)
{
    header('Content-Type: application/json');
    $result = $this->productModel->deleteProduct($id);

    if ($result) {
        echo json_encode(['message' => 'Product deleted successfully']);
    }
}
```

```

    } else {
        http_response_code(400);
        echo json_encode(['message' => 'Product deletion failed']);
    }
}
}
?>

```

### Cập nhật AccountController.php

```

<?php
require_once('app/config/database.php');
require_once('app/models/AccountModel.php');
require_once('app/utils/JWTHandler.php');
class AccountController {
    private $accountModel;
    private $db;

    private $jwtHandler;
    public function __construct() {
        $this->db = (new Database())->getConnection();
        $this->accountModel = new AccountModel($this->db);
        $this->jwtHandler = new JWTHandler();
    }

    function register(){
        include_once 'app/views/account/register.php';
    }
    public function login() {
        include_once 'app/views/account/login.php';
    }

    function save(){

        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            $username = $_POST['username'] ?? '';
            $fullName = $_POST['fullname'] ?? '';
            $password = $_POST['password'] ?? '';
            $confirmPassword = $_POST['confirmpassword'] ?? '';

            $errors = [];
            if(empty($username)){
                $errors['username'] = "Vui long nhap userName!";
            }
            if(empty($fullName)){
                $errors['fullname'] = "Vui long nhap fullName!";
            }

```

```
}
if(empty($password)){
    $errors['password'] = "Vui long nhap password!";
}
if($password != $confirmPassword){
    $errors['confirmPass'] = "Mat khau va xac nhan chua dung";
}
//kiểm tra username đã được đăng ký chưa?
$account = $this->accountModel->getAccountByUsername($username);

if($account){
    $errors['account'] = "Tai khoan nay da co nguoi dang ky!";
}

if(count($errors) > 0){
    include_once 'app/views/account/register.php';
}else{
    $password = password_hash($password, PASSWORD_BCRYPT, ['cost' => 12]);

    $result = $this->accountModel->save($username, $fullName, $password);

    if($result){
        header('Location: /webbanhang/account/login');
    }
}
}

function logout(){

    unset($_SESSION['username']);
    unset($_SESSION['role']);

    header('Location: /webbanhang/product');
}
public function checkLogin()
{
    header('Content-Type: application/json');
    $data = json_decode(file_get_contents("php://input"), true);

    $username = $data['username'] ?? '';
    $password = $data['password'] ?? '';

    $user = $this->accountModel->getAccountByUserName($username);
    if ($user && password_verify($password, $user->password)) {
        $token = $this->jwtHandler->encode(['id' => $user->id, 'username' =>
$user->username]);
    }
```

```

        echo json_encode(['token' => $token]);
    } else {
        http_response_code(401);
        echo json_encode(['message' => 'Invalid credentials']);
    }
}
}

```

## 6.4 Cập nhật các trang hiển thị

Cập nhật views app/views/account/login.php

```

<?php include 'app/views/shares/header.php'; ?>

<section class="vh-100 gradient-custom">
    <div class="container py-5 h-100">
        <div class="row d-flex justify-content-center align-items-center h-100">
            <div class="col-12 col-md-8 col-lg-6 col-xl-5">
                <div class="card bg-dark text-white" style="border-radius: 1rem;">
                    <div class="card-body p-5 text-center">

                        <form id="login-form">
                            <div class="mb-md-5 mt-md-4 pb-5">

                                <h2 class="fw-bold mb-2 text-uppercase">Login</h2>
                                <p class="text-white-50 mb-5">Please enter your login and password!</p>

                                <div class="form-outline form-white mb-4">
                                    <input type="text" name="username" class="form-control form-control-
lg" />

                                    <label class="form-label" for="typeEmailX">UserName</label>
                                </div>

                                <div class="form-outline form-white mb-4">
                                    <input type="password" name="password" class="form-control form-
control-lg" />

                                    <label class="form-label" for="typePasswordX">Password</label>
                                </div>

                                <p class="small mb-5 pb-lg-2"><a class="text-white-50" href="#">Forgot
password?</a></p>

                                <button class="btn btn-outline-light btn-lg px-5"
type="submit">Login</button>

```

```

        <div class="d-flex justify-content-center text-center mt-4 pt-1">
            <a href="#" class="text-white"><i class="fab fa-facebook-f fa-
lg"></i></a>
            <a href="#" class="text-white"><i class="fab fa-twitter fa-lg mx-4
px-2"></i></a>
            <a href="#" class="text-white"><i class="fab fa-google fa-
lg"></i></a>
        </div>

    </div>

    <div>
        <p class="mb-0">Don't have an account? <a href="#" class="text-white-50
fw-bold">Sign Up</a>
        </p>
    </div>
</form>

</div>
</div>
</div>
</div>
</div>
</section>

<?php include 'app/views/shares/footer.php'; ?>
<script>
document.getElementById('login-form').addEventListener('submit', function(event) {
    event.preventDefault();

    const formData = new FormData(this);
    const jsonData = {};
    formData.forEach((value, key) => {
        jsonData[key] = value;
    });

    fetch('/webbanhang/account/checkLogin', {
        method: 'POST',
        headers: {
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(jsonData)
    })
    .then(response => response.json())
    .then(data => {

        if (data.token) {

```

```

        localStorage.setItem('jwtToken', data.token);
        location.href = '/webbanhang/Product';
    } else {
        alert('Đăng nhập thất bại');
    }
    });
});
</script>

```

Cập nhật file header.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Quản lý sản phẩm</title>
    <link
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"
rel="stylesheet">
    <style>
        .product-image {
            max-width: 100px;
            height: auto;
        }
    </style>
</head>

<body>
    <nav class="navbar navbar-expand-lg navbar-light bg-light">
        <a class="navbar-brand" href="#">Quản lý sản phẩm</a>
        <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNav"
        aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle
navigation">
            <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNav">
            <ul class="navbar-nav">
                <li class="nav-item">
                    <a class="nav-link" href="/webbanhang/Product/">Danh sách sản
phẩm</a>
                </li>
                <li class="nav-item">
                    <a class="nav-link" href="/webbanhang/Product/add">Thêm sản
phẩm</a>

```



```

        </li>
        <li class="nav-item" id="nav-login">
            <a class="nav-link" href="/webbanhang/account/login">Login</a>
        </li>
        <li class="nav-item" id="nav-logout" style="display: none;">
            <a class="nav-link" href="#" onclick="logout()">Logout</a>
        </li>
    </ul>
</div>
</nav>

<script>
function logout() {
    localStorage.removeItem('jwtToken');
    location.href = '/webbanhang/account/login';
}

document.addEventListener("DOMContentLoaded", function() {
    const token = localStorage.getItem('jwtToken');
    if (token) {
        document.getElementById('nav-login').style.display = 'none';
        document.getElementById('nav-logout').style.display = 'block';
    } else {
        document.getElementById('nav-login').style.display = 'block';
        document.getElementById('nav-logout').style.display = 'none';
    }
});
</script>
<div class="container mt-4">

```

### Cập nhật trang list.php

```

<?php include 'app/views/shares/header.php'; ?>

<h1>Danh sách sản phẩm</h1>
<a href="/webbanhang/Product/add" class="btn btn-success mb-2">Thêm sản phẩm mới</a>
<ul class="list-group" id="product-list">
    <!-- Danh sách sản phẩm sẽ được tải từ API và hiển thị tại đây -->
</ul>

<?php include 'app/views/shares/footer.php'; ?>

<script>
document.addEventListener("DOMContentLoaded", function() {
    const token = localStorage.getItem('jwtToken');
    if (!token) {
        alert('Vui lòng đăng nhập');
        location.href = '/webbanhang/account/login'; // Điều hướng đến trang đăng nhập
    }
});

```

```

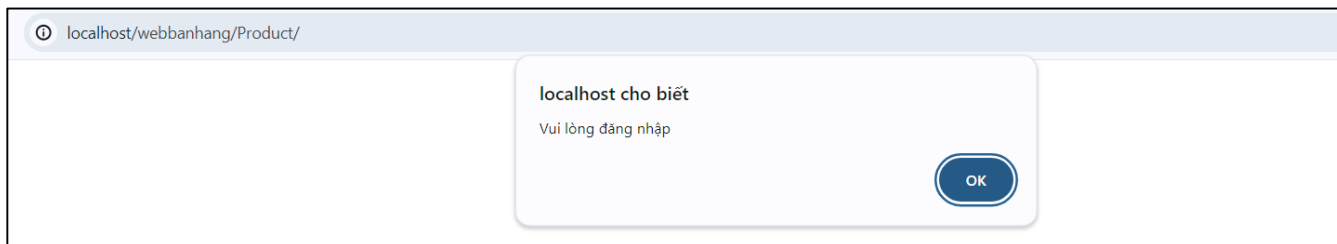
        return;
    }
    fetch('/webbanhang/api/product', {
        method: 'GET',
        headers: {
            'Content-Type': 'application/json',
            'Authorization': 'Bearer ' + token
        }
    })
    .then(response => response.json())
    .then(data => {
        const productList = document.getElementById('product-list');
        data.forEach(product => {
            const productItem = document.createElement('li');
            productItem.className = 'list-group-item';
            productItem.innerHTML = `
                <h2><a
href="/webbanhang/Product/show/${product.id}">${product.name}</a></h2>
                <p>${product.description}</p>
                <p>Giá: ${product.price} VND</p>
                <p>Danh mục: ${product.category_name}</p>
                <a href="/webbanhang/Product/edit/${product.id}" class="btn btn-
warning">Sửa</a>
                <button class="btn btn-danger"
onclick="deleteProduct(${product.id})">Xóa</button>
            `;
            productList.appendChild(productItem);
        });
    });
});
function deleteProduct(id) {
    if (confirm('Bạn có chắc chắn muốn xóa sản phẩm này?')) {
        fetch(`/webbanhang/api/product/${id}`, {
            method: 'DELETE'
        })
        .then(response => response.json())
        .then(data => {
            if (data.message === 'Product deleted successfully') {
                location.reload();
            } else {
                alert('Xóa sản phẩm thất bại');
            }
        });
    }
}
</script>

```



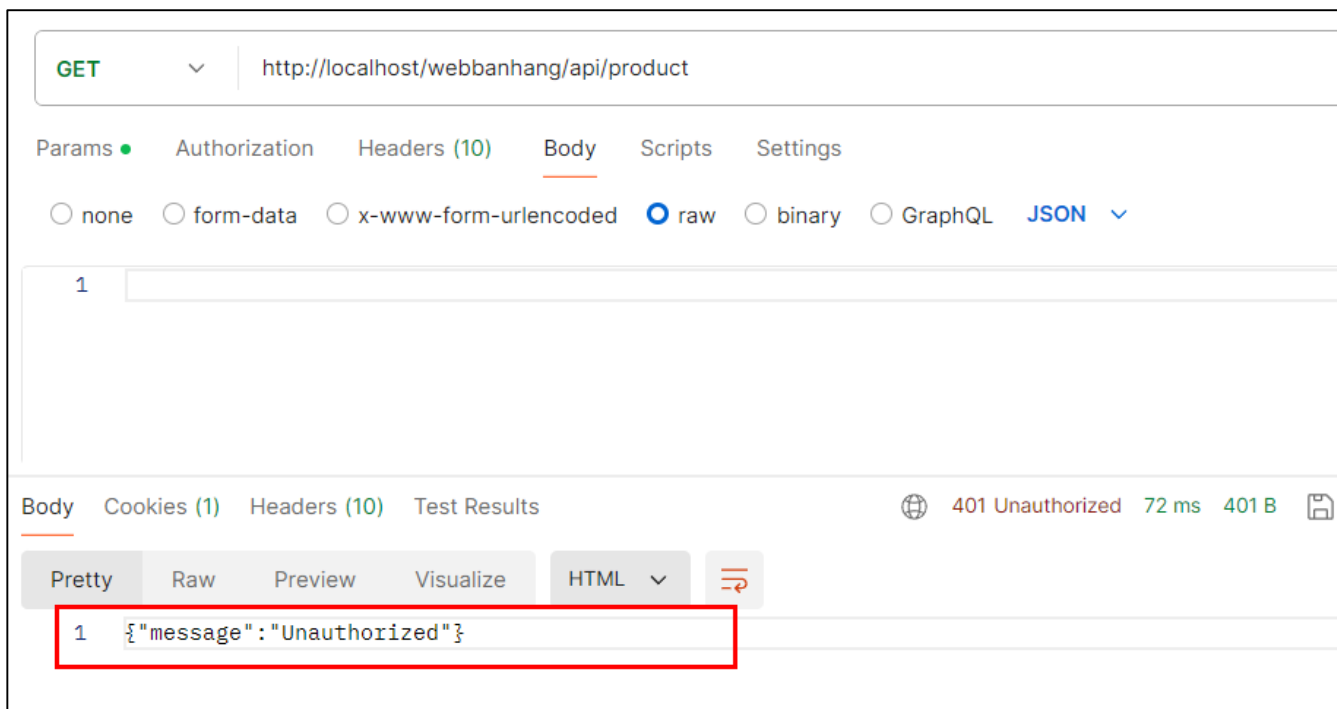
## 6.5 Tiến hành khởi chạy dự án và thực nghiệm

Yêu cầu đăng nhập mới xem được trang danh sách sản phẩm:



Kiểm thử trong Postman:

API đã bị chặn khi không đăng nhập tài khoản



## TÀI LIỆU THAM KHẢO

1. Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin, 2008
2. Eloquent JavaScript: A Modern Introduction to Programming, Marijn Haverbeke, 2018 (3rd edition)
3. You Don't Know JS (series), Kyle Simpson, 2015
4. MDN Web Docs (<https://developer.mozilla.org>)
5. Stack Overflow (<https://stackoverflow.com>)
6. GitHub (<https://github.com>)
7. W3Schools (<https://www.w3schools.com>)
8. Smashing Magazine (<https://www.smashingmagazine.com>)