

**TRƯỜNG ĐẠI HỌC TRÀ VINH**  
**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**



**ISO 9001:2015**

**HUỖNH GIA BẢO**

**XÂY DỰNG**  
**SÀN GIAO DỊCH NÔNG SẢN**  
**THEO MÔ HÌNH C2C**

**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH CÔNG NGHỆ THÔNG TIN**

**TRÀ VINH, NĂM 2024**

TRƯỜNG ĐẠI HỌC TRÀ VINH  
KHOA KỸ THUẬT VÀ CÔNG NGHỆ

XÂY DỰNG  
SÀN GIAO DỊCH NÔNG SẢN  
THEO MÔ HÌNH C2C

ĐỒ ÁN TỐT NGHIỆP  
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên: Huỳnh Gia Bảo

Lớp: DA20TTA

MSSV: 110120128

GVHD: Ths. Nguyễn Khắc Quốc

TRÀ VINH, NĂM 2024

## LỜI MỞ ĐẦU

Trước sự phát triển mạnh mẽ của công nghệ thông tin, mạng máy tính đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày, không chỉ kết nối con người mà còn là nền tảng cho các hệ thống thông tin, quản lý dữ liệu và giao dịch tài chính. Xu hướng này đã mở ra không gian cho sự nổi lên của các nền tảng thương mại điện tử, đặc biệt là mô hình người dùng tới người dùng (C2C) trong lĩnh vực nông sản.

Đề tài này tập trung vào việc xây dựng hệ thống sàn giao dịch nông sản theo mô hình C2C, nhằm cung cấp một nền tảng an toàn và hiệu quả cho các nhà giao dịch. Mục tiêu là nghiên cứu và tối ưu hóa giao diện người dùng, đảm bảo tính thân thiện và dễ sử dụng, đồng thời triển khai các biện pháp bảo mật để bảo vệ thông tin cá nhân và giao dịch của người dùng để có thể đem lại những giá trị thiết thực và mang đến sự hài lòng cho người dùng.

## LỜI CẢM ƠN

Trước hết, em xin gửi lời cảm ơn chân thành đến các thầy cô tại trường, đặc biệt là các thầy cô thuộc Bộ môn Công nghệ Thông tin đã dành thời gian và tâm huyết để truyền đạt những kiến thức quý báu cho em.

Em xin chân thành cảm ơn thầy Nguyễn Khắc Quốc đã hỗ trợ và tư vấn cho em trong quá trình nghiên cứu đề tài "Xây dựng hệ thống sàn giao dịch nông sản theo mô hình C2C". Những đóng góp và sự chỉ dẫn tận tình của thầy đã giúp em hoàn thành đề tài này một cách tốt nhất và đúng thời hạn quy định.

Với sự hỗ trợ chân thành từ phía thầy Nguyễn Khắc Quốc, em hy vọng sẽ tiếp tục nhận được những góp ý và hướng dẫn quý báu để nâng cao kỹ năng và hiểu biết trong các công việc sau này.

Em xin một lần nữa bày tỏ lòng biết ơn sâu sắc đến thầy Nguyễn Khắc Quốc!

*Trà Vinh, ngày ..... tháng 6 năm 2024*

Sinh viên thực hiện

**Huỳnh Gia Bảo**

## This image shows a full page of a notebook or worksheet. It features approximately 20 horizontal rows of small, evenly spaced dots, designed to guide handwriting. The dots are arranged in straight lines across the width of the page, leaving a consistent margin at the top. The background is plain white, and there are no other markings, text, or illustrations present.

**Giảng viên hướng dẫn**  
(ký và ghi rõ họ tên)

.....

.....

.....

.....

.....

4. Điểm mới đề tài:

.....

.....

.....

.....

.....

5. Giá trị thực trên đề tài:

.....

.....

.....

.....

.....

.....

.....

7. Đề nghị sửa chữa bổ sung:

.....

.....

.....

.....

.....

.....

.....

8. Đánh giá:

.....

.....

.....

.....

Trà Vinh, ngày ..... tháng ..... năm 2024  
Giảng viên hướng dẫn  
(Ký & ghi rõ họ tên)

[illegible]

**Giảng viên chấm**  
(ký và ghi rõ họ tên)



**BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP**  
(Của cán bộ chấm đồ án, khóa luận)

Họ và tên người nhận xét: .....  
Chức danh: ..... Học vị: .....  
Chuyên ngành: .....  
Cơ quan công tác: .....  
Tên sinh viên: .....  
Tên đề tài đồ án, khóa luận tốt nghiệp: .....  
.....  
.....

**I. Ý KIẾN NHẬN XÉT**

1. Nội dung:  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....
2. Điểm mới các kết quả của đồ án, khóa luận:  
.....  
.....  
.....
3. Ứng dụng thực tế:  
.....  
.....  
.....  
.....  
.....  
.....

(Các câu hỏi của giáo viên phản biện)

### III. KẾT LUẬN

(Ghi rõ đồng ý hay không đồng ý cho bảo vệ đồ án khóa luận tốt nghiệp)

....., ngày ..... tháng ..... năm 2024

Người nhận xét  
(Ký & ghi rõ họ tên)

## MỤC LỤC

<b>CHƯƠNG 1. ĐẶT VẤN ĐỀ .....</b>	<b>1</b>
1.1. Lý do chọn đề tài.....	1
1.2. Mục tiêu.....	1
1.3. Nội dung .....	1
1.4. Đối tượng và phạm vi nghiên cứu.....	3
1.5. Phương pháp nghiên cứu .....	4
<b>CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....</b>	<b>5</b>
2.1. Mô tả đề tài .....	5
2.2. Laravel Framework .....	6
2.2.1. Giới thiệu về Laravel .....	6
2.3. Mô hình kinh doanh C2C .....	8
2.3.1 Giới thiệu về mô hình C2C .....	8
2.4 Hệ quản trị cơ sở dữ liệu MySQL.....	9
2.4.1 Giới thiệu về MySQL .....	9
2.5 Ngôn ngữ lập trình PHP .....	10
2.5.1 Giới thiệu về PHP .....	10
2.6 Artisan .....	12
2.6.1 Giới thiệu về Artisan.....	12
2.7 Mô hình MVC.....	13
2.7.1 Các thành phần trong mô hình MVC .....	13
2.7.1 Luồng xử lý trong MVC .....	14
2.7.2 Ưu và nhược điểm của MVC.....	14
2.8 Thư viện Bootstrap.....	15
2.8.1 Giới thiệu về Bootstrap.....	15
2.9 Nội dung nghiên cứu .....	17
2.9.1 Laravel.....	17
2.9.2 Mô hình kinh doanh C2C.....	19
2.9.3 Ngôn ngữ lập trình PHP.....	21
<b>CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU .....</b>	<b>23</b>
3.1. Cài đặt Laravel Framework .....	23
3.2 Thiết kế dữ liệu .....	25
3.2.1 Lược đồ use case tổng quan của hệ thống.....	25
3.2.2 Mô tả các use case.....	25
3.2.2 Mô hình dữ liệu .....	29
3.2.3 Chi tiết thực thể .....	30

<b>CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU .....</b>	<b>32</b>
4.1. Giao diện trang chủ .....	32
4.2. Giao diện chi tiết sản phẩm .....	33
4.3 Giao diện danh mục của từng loại hàng nông sản .....	33
4.3.1 Giao diện danh mục lúa gạo.....	33
4.3.2 Giao diện danh mục rau củ .....	34
4.3.3 Giao diện danh mục trái cây .....	34
4.3.4 Giao diện danh mục dưa muối .....	35
4.4 Giao diện quản lý gian hàng .....	35
4.5 Giao diện trang thông tin người dùng .....	36
4.6 Giao diện trang giỏ hàng .....	36
4.7 Giao diện trang thanh toán.....	37
4.8 Giao diện trang đặt hàng thành công.....	37
4.9 Giao diện trang đăng nhập.....	38
4.10 Giao diện trang đăng ký .....	39
4.11 Giao diện trang quản lý người dùng.....	39
4.12 Giao diện trang quản lý tất cả gian hàng .....	40
4.13 Giao diện trang duyệt bài đăng .....	40
4.14 Giao diện trang thanh toán Momo .....	41
<b>CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>42</b>
5.1. Kết luận.....	42
5.2. Hướng phát triển .....	42
<b>DANH MỤC TÀI LIỆU THAM KHẢO.....</b>	<b>43</b>

## DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH

Hình 2.1 Các thành phần của mô hình MVC .....	14
Hình 2.2 Luồng xử lý trong MVC .....	14
Hình 3.1 Trang tải Composer cho Windows.....	23
Hình 3.2 Khởi tạo dự án Laravel .....	23
Hình 3.3 Thư mục chứa dự án vừa tạo .....	24
Hình 1.4 Mô hình use case tổng quan của hệ thống .....	24
Hình 1.5 Use case xem sản phẩm .....	24
Hình 1.6 Use case đặt hàng .....	25
Hình 1.7 Use case đăng ký .....	25
Hình 3.8 Use case đăng nhập .....	26
Hình 3.9 Use case quản lý giỏ hàng .....	26
Hình 3.10 Use case đăng xuất .....	27
Hình 3.11 Use case Quản lý sản phẩm .....	27
Hình 3.12 Use case Quản lý người dùng .....	28
Hình 3.13 Mô hình cơ sở dữ liệu.....	28
Hình 4.1 Giao diện trang chủ .....	32
Hình 4.2 Giao diện chi tiết sản phẩm.....	33
Hình 4.3 Giao diện danh mục lúa gạo .....	33
Hình 4.4 Giao diện danh mục rau củ .....	34
Hình 4.5 Giao diện danh mục trái cây .....	34
Hình 4.6 Giao diện danh mục dưa muối .....	35
Hình 4.7 Giao diện quản lý gian hàng .....	34
Hình 4.8 Giao diện Thông tin người dùng.....	35
Hình 4.9 Giao diện giỏ hàng .....	35
Hình 4.10 Giao diện trang thanh toán.....	36
Hình 4.11 Giao diện trang Đặt hàng thành công.....	36
Hình 4.12 Giao diện trang Đăng nhập .....	37
Hình 4.13 Giao diện trang Đăng ký .....	37
Hình 4.14 Giao diện trang Quản lý người dùng.....	38
Hình 4.15 Giao diện trang Quản lý sản phẩm.....	38
Hình 4.16 Giao diện trang duyệt bài đăng .....	39

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
C2C	Customer to customer
COD	Cash On Delivery
CSS	Cascading Style Sheets
HTML	Hyper Text Markup Language
JS	JavaScript
UI	User Interface
UX	User Experience

## **CHƯƠNG 1. ĐẶT VẤN ĐỀ**

### **1.1. Lý do chọn đề tài**

Trong thời kỳ thương mại điện tử phát triển mạnh mẽ, việc xây dựng hệ thống sàn giao dịch nông sản theo mô hình C2C không chỉ là một chiến lược kinh doanh cần thiết mà còn là một bước tiến quan trọng giúp khai thác nhiều cơ hội mới. Nông sản là một ngành có tiềm năng lớn và nhu cầu về một nền tảng trực tuyến để kết nối trực tiếp giữa người nông dân và người tiêu dùng ngày càng tăng. Mô hình C2C (Consumer to Consumer) mang lại lợi ích to lớn bằng cách giảm bớt các khâu trung gian, từ đó giảm chi phí và nâng cao chất lượng sản phẩm đến tay người tiêu dùng.

Việc xây dựng một sàn giao dịch nông sản trực tuyến theo mô hình C2C không chỉ tạo điều kiện thuận lợi cho người nông dân bán sản phẩm trực tiếp mà còn giúp người tiêu dùng dễ dàng tiếp cận nguồn hàng chất lượng, an toàn với giá cả hợp lý. Trong bối cảnh thị trường nông sản truyền thống gặp nhiều thách thức về giá cả và kênh phân phối, một hệ thống trực tuyến có thể giúp giải quyết nhiều vấn đề tồn đọng, đồng thời tạo ra một môi trường giao dịch công bằng và minh bạch.

Do đó, việc chọn đề tài "Xây dựng hệ thống sàn giao dịch nông sản theo mô hình C2C" không chỉ nhằm mục đích học hỏi và nâng cao kiến thức về thương mại điện tử mà còn đóng góp vào việc phát triển bền vững ngành nông sản, đáp ứng nhu cầu ngày càng cao của thị trường và người tiêu dùng. Đây là cơ hội để tôi tìm hiểu và áp dụng các công nghệ tiên tiến vào thực tiễn, tạo ra một nền tảng hữu ích và thiết thực cho cộng đồng.

### **1.2. Mục tiêu**

- Tìm hiểu và nghiên cứu Laravel, mô hình MVC, mô hình C2C và MySQL.
- Xây dựng một nền tảng sàn giao dịch thân thiện, dễ sử dụng cho người dùng.
- Tối ưu hóa trải nghiệm người dùng.

### **1.3. Nội dung**

#### **1. Tìm hiểu công nghệ:**

- Tìm hiểu và sử dụng Laravel để xây dựng ứng dụng web.
- Tìm hiểu và cài đặt XAMPP để tạo môi trường phát triển local.
- Tìm hiểu và sử dụng MySQL để quản lý cơ sở dữ liệu trong ứng dụng Laravel.

## 2. Thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX):

- Xác định giao diện người dùng (UI) và trải nghiệm người dùng (UX) phù hợp với mục đích của trang web.
- Tạo ra giao diện thân thiện, dễ sử dụng.
- Sắp xếp các mặt hàng nông sản vào các danh mục và phân loại rõ ràng để người dùng dễ dàng tìm kiếm.

## 3. Chức năng cơ bản:

- Đăng nhập và đăng ký tài khoản cho người dùng.
- Hiện thị danh sách mặt hàng nông sản, kèm theo thông tin chi tiết và hình ảnh.
- Cho phép người dùng thêm sản phẩm vào giỏ hàng và thực hiện thanh toán.
- Cung cấp bộ lọc để người dùng dễ dàng tìm kiếm sản phẩm theo nhu cầu.

## 4. Giỏ hàng và thanh toán

- Cho phép người dùng thêm và xóa sản phẩm từ giỏ hàng.
- Hiện thị tổng giá trị của giỏ hàng và chi phí vận chuyển (nếu có).
- Cung cấp nhiều phương thức thanh toán như thanh toán trực tuyến qua cổng thanh toán an toàn, thanh toán khi nhận hàng (COD), và các phương thức thanh toán điện tử khác.

## 5. Quản trị:

- Quản lý các gian hàng: người quản trị có thể xem, xóa các gian hàng người dùng đã đăng, duyệt bài đăng.
- Quản lý người dùng: người quản trị có thể xem, xóa, người dùng.



## 1.4. Đối tượng và phạm vi nghiên cứu

### **Đối tượng nghiên cứu:**

Các công nghệ liên quan đến phát triển ứng dụng web, cụ thể là Laravel và MySQL.

Mô hình kinh doanh C2C.

Người dùng của website bao gồm người nông dân cần bán nông sản, người tiêu dùng cần mua nông sản và quản trị viên quản lý các bài đăng và người dùng.

### **Phạm vi nghiên cứu:**

#### 1. Về công nghệ:

- Frontend: Tìm hiểu và áp dụng Blade Templates để phát triển giao diện người dùng.
- Backend: Nghiên cứu ngôn ngữ PHP và cách thức hoạt động của Laravel Framework để xây dựng các tính năng của dự án.
- Database: Sử dụng MySQL để lưu trữ và quản lý dữ liệu. Nghiên cứu cách thiết kế mô hình dữ liệu phù hợp với yêu cầu của hệ thống.

#### 2. Về tính năng và giao diện:

- Thiết kế giao diện người dùng (UI) và trải nghiệm người dùng (UX) nhằm tạo ra một trang web thân thiện và dễ sử dụng.
- Phát triển các tính năng cơ bản cho người dùng như đăng ký, đăng nhập, xem danh sách sản phẩm, lọc sản phẩm, thêm vào giỏ hàng, và thanh toán.
- Cung cấp các tính năng quản trị như quản lý sản phẩm, quản lý người dùng, và duyệt bài đăng.

#### 3. Về người dùng:

- Nghiên cứu nhu cầu và hành vi của người dùng mục tiêu (người nông dân và người tiêu dùng) để thiết kế giao diện và chức năng phù hợp.

#### 4. Phạm vi kỹ thuật

- Xây dựng và triển khai ứng dụng web trên môi trường phát triển Visual Studio Code.
- Đảm bảo tính bảo mật và hiệu suất của hệ thống trong quá trình phát triển và triển khai.

- Tích hợp các phương thức thanh toán an toàn và đáng tin cậy.

### **1.5. Phương pháp nghiên cứu**

*Phương pháp lý thuyết:* Đọc hiểu, tìm hiểu các tài liệu về mô hình kinh doanh C2C, Laravel Framework.

*Phương pháp thực nghiệm:* Xây dựng website sản giao dịch nông sản trên môi trường Visual Studio Code sử dụng Laravel Framework để thiết kế website và cơ sở dữ liệu là MySQL để lưu trữ thông tin.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Mô tả đề tài

Đề tài này nhằm xây dựng một hệ thống sàn giao dịch nông sản theo mô hình C2C (Consumer to Consumer), tạo ra một nền tảng trực tuyến tiện lợi và tối ưu cho người dùng. Mục tiêu chính là phát triển một hệ thống website thân thiện, dễ sử dụng và cung cấp đầy đủ các chức năng cần thiết cho việc mua bán và quản lý sản phẩm nông sản. Các công nghệ được sử dụng để xây dựng hệ thống này bao gồm Laravel cho phần backend và MySQL để lưu trữ và quản lý cơ sở dữ liệu, với môi trường phát triển local được thiết lập bằng XAMPP.

Hệ thống sàn giao dịch sẽ bao gồm các chức năng cơ bản như đăng ký và đăng nhập người dùng, hiển thị danh sách sản phẩm nông sản với thông tin chi tiết và hình ảnh, cho phép người dùng đăng tải sản phẩm, thêm sản phẩm vào giỏ hàng và thực hiện các giao dịch mua bán. Ngoài ra, hệ thống cũng sẽ cung cấp tính năng tìm kiếm và bộ lọc để người dùng dễ dàng tìm kiếm và lựa chọn sản phẩm theo các tiêu chí như loại sản phẩm, giá cả, địa điểm và tình trạng sản phẩm.

Để tối ưu hóa trải nghiệm người dùng, hệ thống sẽ có giao diện thân thiện, dễ sử dụng, cung cấp nhiều phương thức thanh toán an toàn và tiện lợi. Hệ thống cũng bao gồm các chức năng quản trị như quản lý gian hàng, quản lý người dùng.

Với mục tiêu tạo ra một nền tảng giao dịch nông sản hiệu quả và an toàn, đề tài này hy vọng sẽ góp phần thúc đẩy sự phát triển của ngành nông nghiệp và mang lại lợi ích thiết thực cho cộng đồng người dùng.

## 2.2. Laravel Framework

### 2.2.1. Giới thiệu về Laravel

Laravel là một framework PHP mã nguồn mở được phát triển bởi Taylor Otwell, ra mắt vào năm 2011. Laravel được thiết kế để hỗ trợ các nhà phát triển xây dựng các ứng dụng web mạnh mẽ và linh hoạt theo mô hình MVC (Model-View-Controller). Một trong những điểm nổi bật của Laravel là cú pháp rõ ràng và dễ hiểu, giúp các lập trình viên dễ dàng tiếp cận và phát triển ứng dụng một cách nhanh chóng.

Laravel cung cấp một hệ sinh thái phong phú với nhiều gói mở rộng và thư viện hỗ trợ, giúp việc phát triển các tính năng trở nên dễ dàng hơn. Eloquent ORM là một công cụ quản lý cơ sở dữ liệu mạnh mẽ của Laravel, cho phép tương tác với cơ sở dữ liệu một cách trực quan và hiệu quả. Blade, công cụ template của Laravel, giúp tạo ra các giao diện web dễ dàng với các tính năng như kế thừa layout và các directive tùy chỉnh.

Trước khi có Laravel, việc phát triển các ứng dụng web phức tạp với PHP thường gặp nhiều khó khăn và mất nhiều thời gian. Laravel ra đời với mục đích cải thiện quá trình phát triển ứng dụng web, giúp các lập trình viên xây dựng các ứng dụng nhanh hơn, hiệu quả hơn và ít lỗi hơn. Với hệ thống routing linh hoạt, middleware mạnh mẽ, và hệ thống migration và seeding tiện lợi, Laravel đã trở thành một công cụ không thể thiếu đối với nhiều nhà phát triển web.

Laravel không chỉ mang lại hiệu quả trong việc phát triển ứng dụng mà còn đảm bảo tính bảo mật với các tính năng như mã hóa, bảo vệ CSRF và XSS, cùng với hệ thống xác thực người dùng mạnh mẽ. Cộng đồng lớn và tích cực cùng tài liệu phong phú của Laravel cũng là một điểm mạnh, giúp các nhà phát triển dễ dàng học tập và giải quyết các vấn đề trong quá trình làm việc.

#### **Ưu điểm:**

Laravel cung cấp một hệ sinh thái phong phú với nhiều gói mở rộng và thư viện hỗ trợ. Điều này giúp bạn không cần phải tự xây dựng từ đầu mọi thứ mà có thể tận dụng các công cụ đã có sẵn như Laravel Cashier, Laravel Passport, và Laravel Socialite.

Laravel cho phép bạn xây dựng các thành phần có thể tái sử dụng qua nhiều dự án khác nhau. Điều này giúp tiết kiệm thời gian và công sức khi phát triển các ứng dụng mới, đồng thời đảm bảo tính nhất quán và giảm thiểu lỗi.

Eloquent ORM của Laravel giúp việc tương tác với cơ sở dữ liệu trở nên dễ dàng và trực quan hơn. Bạn có thể định nghĩa các quan hệ giữa các bảng dữ liệu một cách dễ dàng và sử dụng chúng trong mã nguồn mà không cần viết quá nhiều SQL.

Laravel hỗ trợ tốt cho SEO với các công cụ và phương pháp tối ưu hóa. Bạn có thể dễ dàng tạo ra các URL thân thiện với SEO và quản lý meta tags để cải thiện thứ hạng tìm kiếm trên các công cụ như Google.

Laravel cung cấp các công cụ mạnh mẽ để debug và phát triển ứng dụng, như Laravel Tinker và Laravel Debugbar. Điều này giúp bạn nhanh chóng phát hiện và sửa lỗi, tăng tốc quá trình phát triển và triển khai sản phẩm.

Laravel đi kèm với các công cụ phát triển hiện đại như Laravel Mix để quản lý tài nguyên front-end, và Laravel Horizon để giám sát các hàng đợi. Những công cụ này giúp tăng cường hiệu quả và khả năng mở rộng của ứng dụng.

Laravel có nhiều tính năng bảo mật tích hợp như mã hóa, bảo vệ CSRF, và bảo vệ XSS. Điều này giúp bảo vệ ứng dụng của bạn khỏi các mối đe dọa bảo mật phổ biến và đảm bảo an toàn cho dữ liệu người dùng.

Laravel không chỉ là một framework PHP mạnh mẽ mà còn là một công cụ phát triển toàn diện, giúp bạn xây dựng các ứng dụng web chất lượng cao một cách nhanh chóng và hiệu quả.

### **Nhược điểm:**

Đối với những người mới bắt đầu, việc học Laravel có thể khá phức tạp. Nó yêu cầu hiểu biết về PHP cơ bản và một số khái niệm về lập trình hướng đối tượng, MVC và các công nghệ web liên quan.

Laravel không cung cấp một cách tiếp cận kiến trúc cứng nhắc, điều này có thể làm cho việc quản lý các phần của ứng dụng, như routing, middleware và services, trở nên phức tạp nếu không được tổ chức một cách cẩn thận.

Mặc dù Laravel cung cấp một hệ sinh thái mạnh mẽ với các gói và thư viện hỗ trợ, nhưng việc tích hợp với các thư viện hoặc API bên ngoài đôi khi có thể gặp phải một số thách thức, đặc biệt là khi phải duy trì tính tương thích giữa các phiên bản.

Laravel có các cơ chế tối ưu hóa hiệu suất, nhưng việc quản lý không tốt các truy vấn database, middleware và caching có thể dẫn đến hiệu suất kém. Điều này đòi hỏi lập trình viên phải có kiến thức sâu rộng và cẩn thận trong việc tối ưu hóa ứng dụng.

Với việc Laravel thường xuyên cập nhật và phát hành phiên bản mới, việc duy trì và cập nhật ứng dụng có thể đòi hỏi nhiều thời gian và công sức, đặc biệt là khi có các thay đổi lớn trong framework.

Laravel có một hệ sinh thái phong phú, nhưng điều này cũng có nghĩa là bạn có thể trở nên phụ thuộc vào các công cụ và gói của Laravel. Điều này có thể gây khó khăn nếu bạn muốn chuyển sang một framework khác hoặc cần tích hợp với các công cụ không thuộc hệ sinh thái của Laravel.

Nhìn chung, mặc dù Laravel là một framework mạnh mẽ và linh hoạt, nhưng nó cũng đòi hỏi một mức độ hiểu biết và quản lý nhất định để phát triển các ứng dụng hiệu quả và duy trì lâu dài.

## 2.3. Mô hình kinh doanh C2C

### 2.3.1 Giới thiệu về mô hình C2C

Mô hình C2C (Consumer to Consumer) là một dạng mô hình kinh doanh mà ở đó người tiêu dùng có thể trao đổi, mua bán hàng hóa hoặc dịch vụ trực tiếp với nhau thông qua một nền tảng trung gian. Dưới đây là một giới thiệu chi tiết về mô hình C2C:

**C2C là mô hình kinh doanh mà người tiêu dùng giao dịch với nhau:** Trái ngược với mô hình B2C (Business to Consumer) hay B2B (Business to Business), mô hình C2C cho phép các cá nhân tự do mua bán và trao đổi hàng hóa hoặc dịch vụ mà không cần phải thông qua các doanh nghiệp lớn. Các nền tảng C2C phổ biến hiện nay bao gồm các trang web đấu giá, thị trường trực tuyến và các nền tảng mạng xã hội.

**Nền tảng trung gian đóng vai trò quan trọng:** Để thực hiện các giao dịch C2C, cần có một nền tảng trung gian đáng tin cậy để kết nối người mua và người bán. Các nền tảng này không chỉ cung cấp không gian để đăng bán sản phẩm mà còn hỗ trợ các chức năng thanh toán, vận chuyển và quản lý tranh chấp. Ví dụ điển hình của nền tảng C2C là

eBay, Amazon Marketplace, Etsy, và các ứng dụng mua bán trực tuyến như Shopee hay Chợ Tốt.

#### **Lợi ích của mô hình C2C:**

**Tiết kiệm chi phí:** Người tiêu dùng có thể mua hàng trực tiếp từ người bán mà không phải chịu chi phí trung gian như trong mô hình B2C.

**Đa dạng sản phẩm:** Người tiêu dùng có thể tìm thấy một loạt các sản phẩm, bao gồm cả những sản phẩm độc đáo, hiếm có hoặc đã qua sử dụng mà thường không có sẵn trong các cửa hàng truyền thống.

**Khả năng kiếm tiền:** Mô hình C2C mở ra cơ hội cho người tiêu dùng bán lại các sản phẩm không còn sử dụng, từ đó tạo ra thu nhập bổ sung.

#### **Thách thức của mô hình C2C:**

**Chất lượng và độ tin cậy:** Do không có sự kiểm soát chặt chẽ như trong các mô hình kinh doanh khác, người mua có thể gặp phải rủi ro về chất lượng sản phẩm hoặc dịch vụ không như mong đợi.

**An toàn và bảo mật:** Giao dịch trực tuyến có thể tiềm ẩn nguy cơ về lừa đảo và an ninh mạng. Các nền tảng C2C cần có các biện pháp bảo vệ người dùng, bao gồm hệ thống đánh giá, bảo đảm thanh toán và quy trình giải quyết tranh chấp.

## **2.4 Hệ quản trị cơ sở dữ liệu MySQL**

### **2.4.1 Giới thiệu về MySQL**

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở rất phổ biến, được phát triển và duy trì bởi Oracle Corporation. Ra đời vào năm 1995, MySQL đã trở thành lựa chọn hàng đầu cho việc lưu trữ và quản lý dữ liệu trong các ứng dụng web và phần mềm doanh nghiệp.

**MySQL giúp cho các dự án mới tránh phải viết lại các thành phần cơ bản:** Với MySQL, bạn không cần phải tự xây dựng các thành phần cơ bản cho hệ thống quản lý cơ sở dữ liệu của mình. Thay vào đó, MySQL cung cấp một tập hợp phong phú các tính năng

đã được thử nghiệm và tối ưu hóa, giúp bạn tập trung vào phát triển các chức năng chính của ứng dụng mà không cần lo lắng về việc xây dựng và quản lý cơ sở dữ liệu từ đầu.

## **Công dụng của MySQL**

**Quản lý dữ liệu dễ dàng:** MySQL cung cấp các công cụ mạnh mẽ để quản lý và thao tác với cơ sở dữ liệu. Bạn có thể dễ dàng tạo, đọc, cập nhật và xóa dữ liệu (CRUD) thông qua các câu lệnh SQL.

**Hiệu suất cao và khả năng mở rộng:** MySQL được thiết kế để xử lý lượng dữ liệu lớn và hỗ trợ nhiều người dùng đồng thời mà vẫn đảm bảo hiệu suất cao. Điều này rất quan trọng đối với các ứng dụng web và doanh nghiệp có lưu lượng truy cập lớn.

**Bảo mật mạnh mẽ:** MySQL cung cấp các tính năng bảo mật tiên tiến, bao gồm kiểm soát truy cập người dùng, mã hóa dữ liệu và hỗ trợ SSL/TLS để bảo vệ dữ liệu khỏi các mối đe dọa bảo mật.

**Tích hợp dễ dàng:** MySQL có thể dễ dàng tích hợp với nhiều ngôn ngữ lập trình và công cụ phát triển, chẳng hạn như PHP, Python, Java và .NET. Điều này giúp các lập trình viên dễ dàng xây dựng và triển khai các ứng dụng sử dụng MySQL.

## **2.5 Ngôn ngữ lập trình PHP**

### **2.5.1 Giới thiệu về PHP**

PHP, viết tắt của "Hypertext Preprocessor", là một ngôn ngữ lập trình kịch bản phía server được thiết kế chủ yếu cho phát triển web nhưng cũng được sử dụng như một ngôn ngữ lập trình mục đích chung. PHP được phát triển vào năm 1994 bởi Rasmus Lerdorf và đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới.

PHP giúp tạo ra các trang web động và tương tác bằng cách nhúng mã PHP vào mã HTML. Khi một trang PHP được truy cập, mã PHP được thực thi trên server và kết quả (thường là HTML) được trả về trình duyệt của người dùng.

PHP có cú pháp đơn giản, dễ hiểu, và tương tự như các ngôn ngữ lập trình khác như C, Java và Perl, làm cho nó dễ học đối với các lập trình viên mới.



PHP có thể dễ dàng nhúng vào HTML, cho phép tạo ra các trang web động một cách nhanh chóng. PHP xử lý các yêu cầu HTTP và có thể kết nối với nhiều loại cơ sở dữ liệu, từ MySQL, PostgreSQL đến SQLite và MongoDB.

PHP có một cộng đồng người dùng và các nhà phát triển rất lớn, cung cấp nhiều tài liệu, tài nguyên và thư viện hỗ trợ. Cộng đồng này không ngừng phát triển và cải tiến PHP để phù hợp với nhu cầu thay đổi của lập trình web.

#### **Ưu điểm của PHP:**

**Dễ học và sử dụng:** PHP có cú pháp đơn giản và tài liệu phong phú, làm cho việc học và sử dụng trở nên dễ dàng đối với các lập trình viên. Chương trình rất dễ học.

**Đa nền tảng:** PHP có thể chạy trên nhiều hệ điều hành khác nhau như Windows, Linux, macOS và Unix.

**Tích hợp dễ dàng với các cơ sở dữ liệu:** PHP hỗ trợ kết nối với nhiều loại cơ sở dữ liệu, cho phép phát triển các ứng dụng web mạnh mẽ và linh hoạt.

**Hiệu suất cao:** PHP có hiệu suất cao trong xử lý các trang web động và tương tác với cơ sở dữ liệu.

**Cộng đồng lớn và nhiều tài nguyên:** PHP có một cộng đồng người dùng lớn, cung cấp nhiều tài liệu, hướng dẫn, và thư viện mã nguồn mở.

#### **Nhược điểm của PHP:**

**Bảo mật:** PHP có thể dễ dàng bị tấn công nếu không được lập trình cẩn thận và không tuân thủ các nguyên tắc bảo mật.

**Khó bảo trì mã nguồn lớn:** Khi các dự án PHP trở nên lớn và phức tạp, việc quản lý và bảo trì mã nguồn có thể trở nên khó khăn.

**Hiệu suất kém đối với các ứng dụng lớn:** PHP có thể không phù hợp cho các ứng dụng lớn đòi hỏi hiệu suất rất cao và thời gian phản hồi nhanh.

**Sự không đồng nhất của các phiên bản:** Do sự phát triển và cải tiến liên tục, có thể có sự không đồng nhất giữa các phiên bản PHP, gây khó khăn cho việc duy trì và cập nhật ứng dụng.

## 2.6 Artisan

### 2.6.1 Giới thiệu về Artisan

Artisan là công cụ dòng lệnh được cung cấp sẵn trong Laravel, giúp lập trình viên thực hiện các tác vụ quản lý và tự động hóa trong quá trình phát triển ứng dụng web. Các tác vụ mà Artisan hỗ trợ bao gồm tạo mô hình (model), controller, migration, seeding dữ liệu, tạo các câu lệnh command tùy chỉnh, và nhiều tác vụ khác nữa. Điều này giúp tối ưu hóa quá trình phát triển bằng cách loại bỏ các thao tác lặp lại thủ công và giảm thiểu lỗi do con người.

#### Ưu điểm của Artisan:

**Tăng tốc độ phát triển:** Artisan cung cấp các lệnh mẫu cho các thành phần chính của ứng dụng như model, controller, migration, giúp lập trình viên tạo các thành phần cơ bản một cách nhanh chóng.

**Giảm thiểu lỗi nhân lực:** Bằng cách sử dụng Artisan, các công việc lặp lại và dễ mắc lỗi như tạo bảng cơ sở dữ liệu, seeding dữ liệu, và chạy migration có thể được thực hiện một cách tự động, giúp giảm thiểu sai sót từ con người.

**Tích hợp dễ dàng:** Artisan được tích hợp sẵn trong Laravel và có cấu trúc rõ ràng, dễ dàng mở rộng với các command tùy chỉnh để phù hợp với nhu cầu cụ thể của dự án.

**Được cộng đồng hỗ trợ và phát triển:** Artisan được sử dụng rộng rãi trong cộng đồng Laravel, do đó có rất nhiều tài liệu và hướng dẫn hỗ trợ, giúp người dùng nhanh chóng tìm hiểu và sử dụng.

#### Nhược điểm của Artisan:

**Khó khăn trong tùy chỉnh nâng cao:** Mặc dù Artisan cung cấp các lệnh mẫu tiêu chuẩn, nhưng tùy chỉnh nâng cao và việc tạo ra các command phức tạp hơn có thể đòi hỏi người dùng có hiểu biết sâu về Laravel và PHP.

**Phụ thuộc vào dòng lệnh:** Việc sử dụng Artisan yêu cầu người dùng phải có kiến thức cơ bản về dòng lệnh và một số kỹ năng về quản lý hệ thống, điều này có thể làm tăng ngưỡng học và sử dụng đối với một số lập trình viên mới.

**Có thể không phù hợp với dự án nhỏ:** Đối với các dự án nhỏ hoặc không có yêu cầu phức tạp, việc sử dụng Artisan có thể được coi là quá mức hoặc không cần thiết.

Tóm lại, Artisan là một công cụ quan trọng và mạnh mẽ trong framework Laravel,

mang lại nhiều lợi ích trong việc tăng tốc quá trình phát triển và giảm thiểu lỗi từ con người, mặc dù cũng có những nhược điểm cần cân nhắc khi sử dụng.

## 2.7 Mô hình MVC

MVC là viết tắt của cụm từ Model-View-Controller. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia thành ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác. Tên gọi 3 thành phần:

- Model (dữ liệu): Quản lý xử lý các dữ liệu.
- View (giao diện): Nơi hiển thị dữ liệu cho người dùng.
- Controller (bộ điều khiển): Điều khiển sự tương tác của hai thành phần

Model và View.

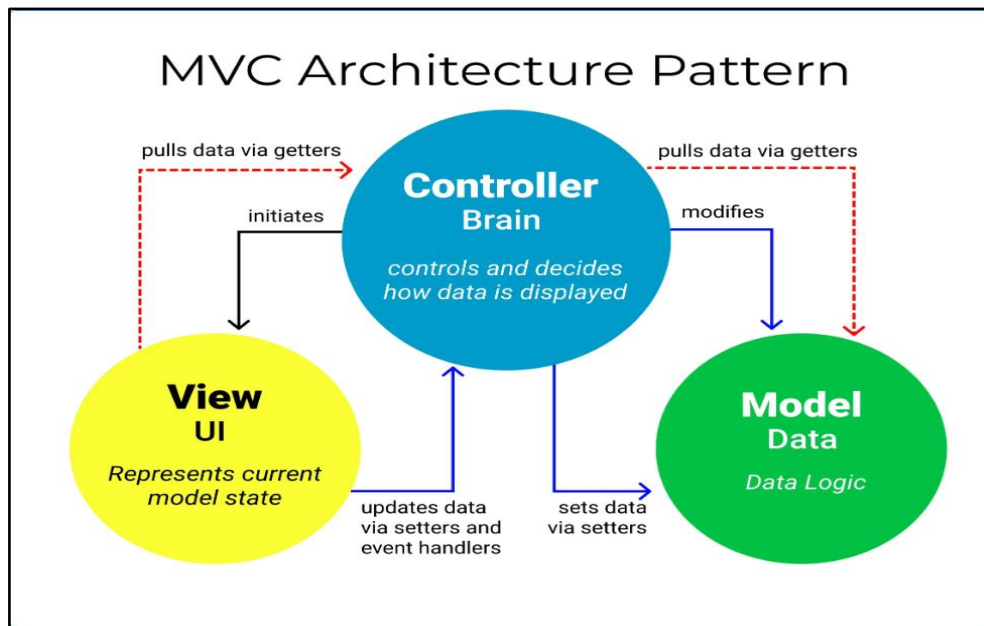
Mô hình MVC (MVC pattern) thường được dùng để phát triển giao diện người dùng. Nó cung cấp các thành phần cơ bản để thiết kế một chương trình cho máy tính hoặc điện thoại di động, cũng như là các ứng dụng web.

### 2.7.1 Các thành phần trong mô hình MVC

Mô hình MVC gồm ba loại chính là thành phần bên trong không thể thiếu khi áp dụng mô hình này:

- Model: Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng và là cầu nối giữa hai thành phần bên dưới là View và Controller. Một model là dữ liệu được sử dụng bởi chương trình. Đây có thể là cơ sở dữ liệu, hoặc file XML bình thường hay một đối tượng đơn giản.
- View: Đây là phần giao diện (theme) dành cho người sử dụng. View là phương tiện hiển thị các đối tượng trong một ứng dụng. Chẳng hạn như hiển thị một cửa sổ, nút hay văn bản trong một cửa sổ khác. Nó bao gồm mọi thứ mà người dùng có thể nhìn thấy được.

Controller: Là bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua View. Một controller bao gồm cả Model lẫn View. Nó nhận input và thực hiện các update tương ứng.



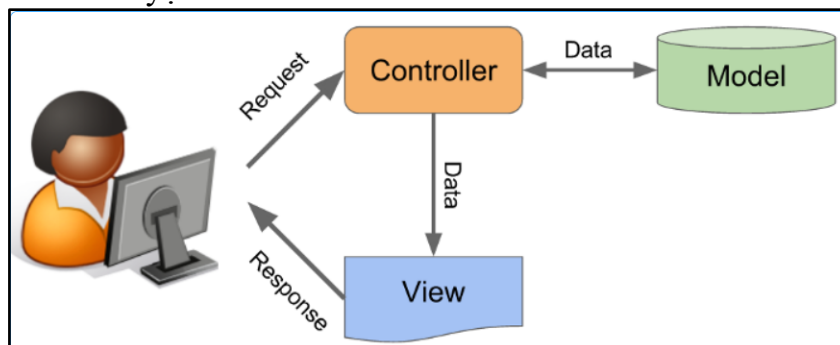
Hình 2.1 Các thành phần của mô hình MVC

### 2.7.1 Luồng xử lý trong MVC

Luồng xử lý trong của mô hình MVC, có thể hình dung cụ thể và chi tiết qua từng bước dưới đây:

- Khi một yêu cầu của từ máy khách (Client) gửi đến Server. Controller trong MVC chặn lại để xem đó là URL request hay sự kiện.
- Sau đó, Controller xử lý input của user rồi giao tiếp với Model trong MVC.
- Model chuẩn bị data và gửi lại cho Controller.

Cuối cùng, khi xử lý xong yêu cầu thì Controller gửi dữ liệu trở lại View và hiển thị cho người dùng trên trình duyệt.



Hình 2.2 Luồng xử lý trong MVC

### 2.7.2 Ưu và nhược điểm của MVC

#### Ưu điểm:

- Đầu tiên, nhắc tới ưu điểm mô hình MVC thì đó là băng thông (Bandwidth) nhẹ vì không sử dụng viewstate nên khá tiết kiệm băng thông. Việc giảm băng thông giúp website hoạt động ổn định hơn.

- Kiểm tra đơn giản và dễ dàng, kiểm tra lỗi phần mềm trước khi bàn giao lại cho người dùng.
- Một lợi thế chính của MVC là nó tách biệt các phần Model, Controller và View với nhau.
- Sử dụng mô hình MVC chức năng Controller có vai trò quan trọng và tối ưu trên các nền tảng ngôn ngữ khác nhau
- Ta có thể dễ dàng duy trì ứng dụng vì chúng được tách biệt với nhau.
- Có thể chia nhiều developer làm việc cùng một lúc. Công việc của các developer sẽ không ảnh hưởng đến nhau.
- Hỗ trợ TTD (test-driven development). Chúng ta có thể tạo một ứng dụng với unit test và viết các won test case.
- Phiên bản mới nhất của MVC hỗ trợ trợ thiết kế responsive website mặc định và các mẫu cho mobile. Chúng ta có thể tạo công cụ View của riêng mình với cú pháp đơn giản hơn nhiều so với công cụ truyền thống.

#### **Nhược điểm:**

- Khó khăn khi bắt đầu: Mô hình MVC có thể khá phức tạp khi bắt đầu, đòi hỏi người lập trình phải có kiến thức và kinh nghiệm nhất định.
- Tốn thời gian và công sức: Mô hình MVC đòi hỏi nhiều thời gian và công sức để thiết kế và triển khai, đặc biệt là đối với các dự án lớn.
- Không phù hợp với các dự án nhỏ: Mô hình MVC có thể trở nên cồng kềnh và tốn kém đối với các dự án nhỏ.

## **2.8 Thư viện Bootstrap**

### **2.8.1 Giới thiệu về Bootstrap**

Bootstrap là một framework front-end mạnh mẽ và phổ biến, được thiết kế để giúp lập trình viên xây dựng các giao diện web nhanh chóng và dễ dàng. Bootstrap cung cấp một bộ công cụ toàn diện bao gồm CSS, JavaScript, và các thành phần UI được thiết kế sẵn như nút, biểu mẫu, thẻ, thanh điều hướng, và nhiều thành phần khác. Điều này giúp đảm bảo rằng các ứng dụng web có giao diện nhất quán và chuyên nghiệp trên các trình duyệt và thiết bị khác nhau.

### **Ưu điểm của Bootstrap:**

- Tăng tốc độ phát triển: Bootstrap cung cấp nhiều thành phần UI và mẫu thiết kế sẵn, giúp lập trình viên nhanh chóng xây dựng và tùy chỉnh giao diện mà không cần viết quá nhiều mã CSS hay JavaScript từ đầu.
- Đáp ứng (Responsive): Với hệ thống lưới (grid system) mạnh mẽ và các tiện ích CSS, Bootstrap giúp các ứng dụng web tự động điều chỉnh và hiển thị tốt trên mọi kích thước màn hình, từ điện thoại di động đến máy tính bàn.
- Dễ sử dụng: Bootstrap có tài liệu chi tiết và dễ hiểu, cùng với cộng đồng lớn, giúp người dùng dễ dàng học và áp dụng vào các dự án của mình. Việc sử dụng các lớp CSS có sẵn và các thành phần được thiết kế sẵn giúp giảm thiểu thời gian và công sức thiết kế giao diện.
- Tính nhất quán: Sử dụng Bootstrap giúp đảm bảo rằng các phần khác nhau của ứng dụng web có giao diện và hành vi nhất quán, giúp cải thiện trải nghiệm người dùng và dễ bảo trì hơn.
- Hỗ trợ từ cộng đồng: Bootstrap có một cộng đồng phát triển lớn và tích cực, cung cấp nhiều tài liệu, hướng dẫn, và các gói mở rộng (plugins) hữu ích giúp người dùng dễ dàng mở rộng và tùy chỉnh giao diện.

### **Nhược điểm của Bootstrap:**

- Giới hạn trong thiết kế: Mặc dù Bootstrap cung cấp nhiều thành phần và tiện ích, nhưng việc sử dụng quá nhiều các lớp và thành phần mặc định của Bootstrap có thể làm cho trang web của bạn trông giống như các trang web khác sử dụng cùng framework. Điều này có thể hạn chế sự sáng tạo và làm giảm tính độc đáo của giao diện.
- Kích thước tệp lớn: Sử dụng toàn bộ Bootstrap có thể làm tăng kích thước tệp CSS và JavaScript của trang web, ảnh hưởng đến tốc độ tải trang và hiệu suất của ứng dụng. Tuy nhiên, điều này có thể được giảm thiểu bằng cách tùy chỉnh và chỉ sử dụng các thành phần cần thiết.
- Phụ thuộc vào lớp CSS: Việc sử dụng nhiều lớp CSS trong HTML có thể làm cho mã nguồn trở nên phức tạp và khó quản lý, đặc biệt là khi dự án phát triển lớn hơn.
- Không tối ưu cho tất cả các dự án: Đối với các dự án nhỏ hoặc đơn giản, việc sử dụng Bootstrap có thể không cần thiết và làm tăng độ phức tạp không cần thiết.

## 2.9 Nội dung nghiên cứu

### 2.9.1 Laravel

Cần nghiên cứu các thành phần sau:

*Controllers:*

- **Basic Controllers:** Điều khiển các logic xử lý trong ứng dụng Laravel, từ việc xử lý dữ liệu đến gửi dữ liệu cho view.
- **Resource Controllers:** Tận dụng các controller tài nguyên để tự động hóa các phương thức CRUD (Create, Read, Update, Delete) cho một tài nguyên nhất định.

Ví dụ về tạo 1 Controller:

```
<?php

namespace App\Http\Controllers;

use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController
{
    //
}
```

*Views:*

- **Blade Templating Engine:** Sử dụng Blade để tạo giao diện người dùng linh hoạt và hiệu quả hơn, bao gồm layout, sections, và components.

```
<div>
    <!-- Smile, breathe, and go slowly. - Thich Nhat Hanh -->
</div>
```

*Models:*

- ORM (Object-Relational Mapping): Sử dụng Eloquent ORM để tương tác với cơ sở dữ liệu, bao gồm các phương thức quan hệ như belongsTo, hasMany, và nâng cao về eager loading.

```
<?php

namespace App\Models;

use Illuminate\Database\Eloquent\Factories\HasFactory;
use Illuminate\Database\Eloquent\Model;

class test extends Model
{
    use HasFactory;
}
```

#### *Routes:*

- Route Definitions: Định nghĩa và quản lý các tuyến đường (routes) trong ứng dụng Laravel, bao gồm route naming, route parameters, và middleware.

#### *Middleware:*

- Custom Middleware: Xây dựng và áp dụng các middleware tùy chỉnh để xử lý các yêu cầu HTTP trước khi chúng được xử lý bởi ứng dụng.

#### *Database Migrations:*

- Migration Files: Sử dụng migration để quản lý và duy trì cấu trúc cơ sở dữ liệu, bao gồm việc tạo, chỉnh sửa và xóa các bảng và cột.

#### *Authentication:*

- Laravel Passport: Cài đặt và cấu hình Laravel Passport để cung cấp xác thực API bảo mật cho ứng dụng.

#### *Testing:*



- Unit Testing và Feature Testing: Thực hiện các bài kiểm tra đơn vị (unit tests) và kiểm tra tính năng (feature tests) để đảm bảo tính đúng đắn và hoạt động như mong đợi của ứng dụng.

*Performance Optimization:*

- Caching và Query Optimization: Tối ưu hóa hiệu suất bằng cách sử dụng caching và tối ưu hóa các câu truy vấn để giảm thiểu thời gian phản hồi của ứng dụng.

*Security:*

- Middleware và Xác thực: Bảo vệ ứng dụng bằng cách sử dụng middleware và xác thực để ngăn chặn các cuộc tấn công và bảo vệ dữ liệu quan trọng.

*Deployment:*

- Continuous Integration và Continuous Deployment (CI/CD): Triển khai ứng dụng Laravel một cách tự động và liên tục để đảm bảo sự ổn định và sẵn sàng cho người dùng cuối.

Nghiên cứu sâu về các thành phần trên sẽ giúp hiểu rõ hơn về cách Laravel hoạt động và cách sử dụng hiệu quả framework này để phát triển các ứng dụng web chuyên nghiệp và bảo mật.

## **2.9.2 Mô hình kinh doanh C2C**

*Định nghĩa và khái niệm:*

- Mô hình C2C (Consumer to Consumer): Hiểu về mô hình kinh doanh C2C, nơi các cá nhân bán sản phẩm hoặc dịch vụ trực tiếp cho người tiêu dùng khác thông qua một nền tảng trung gian.

*Các nền tảng C2C phổ biến:*

- E-commerce Platforms: Tìm hiểu về các nền tảng thương mại điện tử như eBay, Etsy, Amazon Marketplace, và các trang web đấu giá trực tuyến khác.
- Social Media Marketplaces: Nghiên cứu các nền tảng bán hàng qua mạng xã hội như Facebook Marketplace, Instagram Shopping.

*Quy trình hoạt động:*

- Tạo danh sách sản phẩm: Tìm hiểu cách người bán tạo danh sách sản phẩm, bao gồm mô tả, giá cả, và hình ảnh.
- Tạo danh sách sản phẩm: Tìm hiểu cách người bán tạo danh sách sản phẩm, bao gồm mô tả, giá cả, và hình ảnh.
- Đánh giá và phản hồi: Khám phá vai trò của hệ thống đánh giá và phản hồi trong việc xây dựng uy tín và tin cậy giữa người mua và người bán.

#### *Lợi ích của mô hình C2C:*

- Giảm chi phí trung gian: Người bán và người mua có thể giao dịch trực tiếp mà không cần qua trung gian, giúp giảm chi phí.
- Tiếp cận thị trường rộng lớn: Người bán có thể tiếp cận một lượng lớn khách hàng tiềm năng trên toàn cầu.
- Khả năng tái sử dụng và tái chế: Mô hình này khuyến khích việc tái sử dụng và tái chế các sản phẩm, góp phần bảo vệ môi trường.

#### *Thách thức của mô hình C2C:*

- Quản lý chất lượng sản phẩm: Khó kiểm soát chất lượng sản phẩm do không có sự can thiệp của bên thứ ba.
- An ninh và bảo mật: Nguy cơ gian lận và lừa đảo cao hơn do thiếu các biện pháp bảo vệ như trong các mô hình B2C (Business to Consumer).
- Dịch vụ khách hàng: Dịch vụ hỗ trợ khách hàng có thể không được đảm bảo như khi mua hàng từ các doanh nghiệp lớn.

#### *Các yếu tố thành công:*

- Hệ thống đánh giá và phản hồi mạnh mẽ: Tạo dựng lòng tin giữa người mua và người bán thông qua hệ thống đánh giá và phản hồi minh bạch.
- Chính sách bảo vệ người tiêu dùng: Đảm bảo có các chính sách bảo vệ người tiêu dùng để giảm thiểu rủi ro cho người mua.
- Quảng cáo và tiếp thị hiệu quả: Sử dụng các chiến lược quảng cáo và tiếp thị để thu hút người dùng và tạo dựng thương hiệu cho nền tảng C2C.

#### *Các công nghệ hỗ trợ:*

- Payment Gateways: Tìm hiểu về các cổng thanh toán điện tử hỗ trợ giao dịch trong mô hình C2C.

- AI và Machine Learning: Nghiên cứu cách sử dụng AI và Machine Learning để cải thiện trải nghiệm người dùng, tối ưu hóa tìm kiếm sản phẩm và cá nhân hóa đề xuất.

Nghiên cứu sâu về các thành phần trên sẽ giúp hiểu rõ hơn về cách mô hình kinh doanh C2C hoạt động và cách áp dụng hiệu quả mô hình này trong thực tế kinh doanh.

### **2.9.3 Ngôn ngữ lập trình PHP**

Nghiên cứu về ngôn ngữ lập trình PHP để sử dụng trong phát triển web với Laravel gồm các nội dung sau:

#### *Tổng quan về PHP*

- Giới thiệu về PHP: Hiểu về lịch sử phát triển và ứng dụng của PHP trong lập trình web.
- Cú pháp và cấu trúc cơ bản của PHP: Tìm hiểu về cú pháp, biến, hàm, cấu trúc điều kiện và vòng lặp cơ bản của PHP.

#### *Cấu trúc và cú pháp cơ bản*

- Biến và kiểu dữ liệu: Tìm hiểu về các kiểu dữ liệu trong PHP và cách khai báo, sử dụng biến.
- Toán tử và biểu thức: Hiểu về các loại toán tử trong PHP và cách sử dụng chúng trong các biểu thức.
- Cấu trúc điều kiện và vòng lặp: Nắm vững các cấu trúc điều kiện như 'if', 'else', 'switch' và các vòng lặp như 'for', 'while', 'foreach'.

#### *Hàm và lập trình hướng đối tượng (OOP)*

- Kết nối cơ sở dữ liệu: Tìm hiểu cách kết nối PHP với cơ sở dữ liệu MySQL bằng cách sử dụng MySQLi hoặc PDO.
- Thao tác cơ sở dữ liệu: Hiểu về các thao tác cơ bản như tạo, đọc, cập nhật, và xóa (CRUD) trong cơ sở dữ liệu bằng PHP.
- Sử dụng ORM (Object-Relational Mapping): Nghiên cứu về Eloquent ORM trong Laravel để quản lý cơ sở dữ liệu một cách hiệu quả.

### *Quản lý tệp và xử lý biểu mẫu*

- Quản lý tệp: Tìm hiểu cách đọc, ghi và xử lý tệp trong PHP.
- Xử lý biểu mẫu: Hiểu cách thu thập và xử lý dữ liệu từ các biểu mẫu HTML bằng PHP.

### *Bảo mật trong PHP*

- Bảo mật dữ liệu: Nắm vững các kỹ thuật bảo mật dữ liệu như bảo vệ chống tấn công SQL injection, XSS (Cross-site Scripting), và CSRF (Cross-Site Request Forgery).
- Quản lý phiên và cookie: Tìm hiểu cách sử dụng phiên (session) và cookie để quản lý trạng thái người dùng.

### *API và tích hợp frontend*

- Tạo và sử dụng API: Nghiên cứu cách tạo và quản lý API trong Laravel để cung cấp dữ liệu cho ứng dụng frontend.
- Tích hợp với JavaScript: Hiểu về cách giao tiếp với API từ phía frontend bằng cách sử dụng các phương thức HTTP như GET, POST, PUT, và DELETE.

## CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Cài đặt Laravel Framework

Bước 1: Vào trang <https://getcomposer.org/download/> sau đó tải tệp Composer-Setup.exe và tiến hành cài đặt

[Home](#) | [Getting Started](#) | [Download](#) | [Documentation](#) | [Browse Packages](#)

Download Composer Latest: v2.7.7

#### Windows Installer

The installer - which requires that you have PHP already installed - will download Composer for you and set up your PATH environment variable so you can simply call `composer` from any directory.

Download and run [Composer-Setup.exe](#) - it will install the latest composer version whenever it is executed.

#### Command-line installation

To quickly install Composer in the current directory, run the following script in your terminal. To automate the installation, use [the guide on installing Composer programmatically](#).

```
php -r "copy('https://getcomposer.org/installer', 'composer-setup.php');"
php -r "if (hash_file('sha384', 'composer-setup.php') === 'dac665fdc30fdd8ec78b38b9800061b4150413ff2e3b6f88543c
php composer-setup.php
php -r "unlink('composer-setup.php');"
```

This installer script will simply check some `php.ini` settings, warn you if they are set incorrectly, and then download the latest `composer.phar` in the current directory. The 4 lines above will, in order:

- Download the installer to the current directory
- Verify the installer SHA-384, which you can also [cross-check here](#)
- Run the installer
- Remove the installer

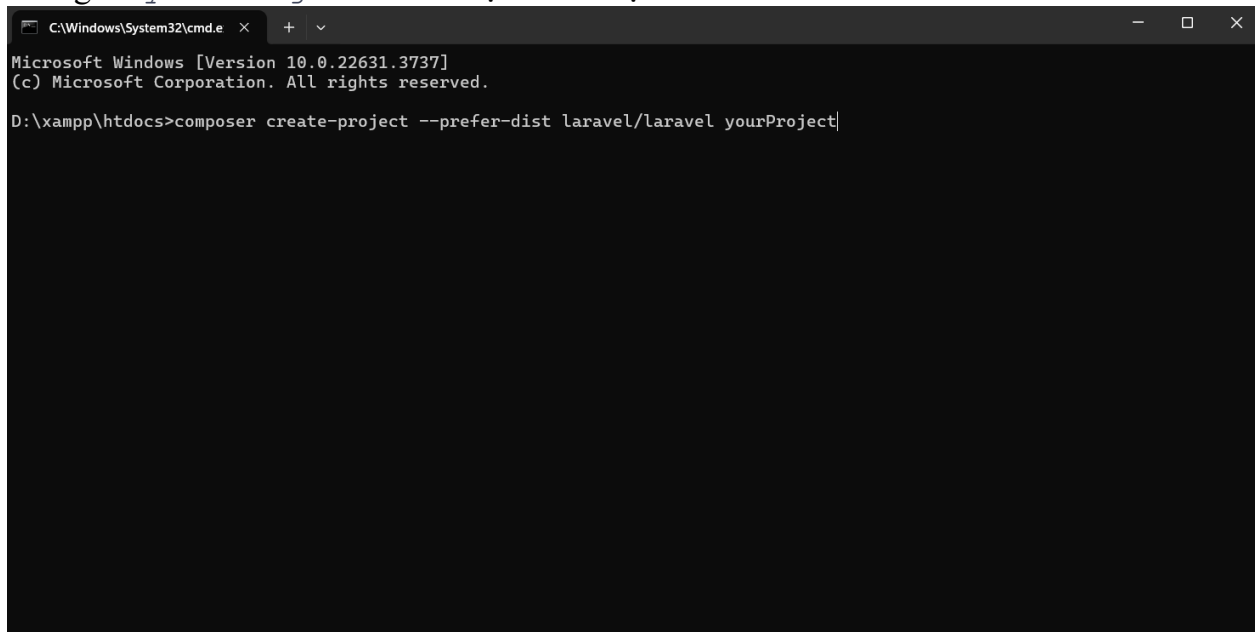
Most likely, you want to put the `composer.phar` into a directory on your PATH, so you can simply call `composer` from any directory

Hình 3.1 Trang tải Composer cho Windows

Bước 2: Chạy mã lệnh cài đặt laravel

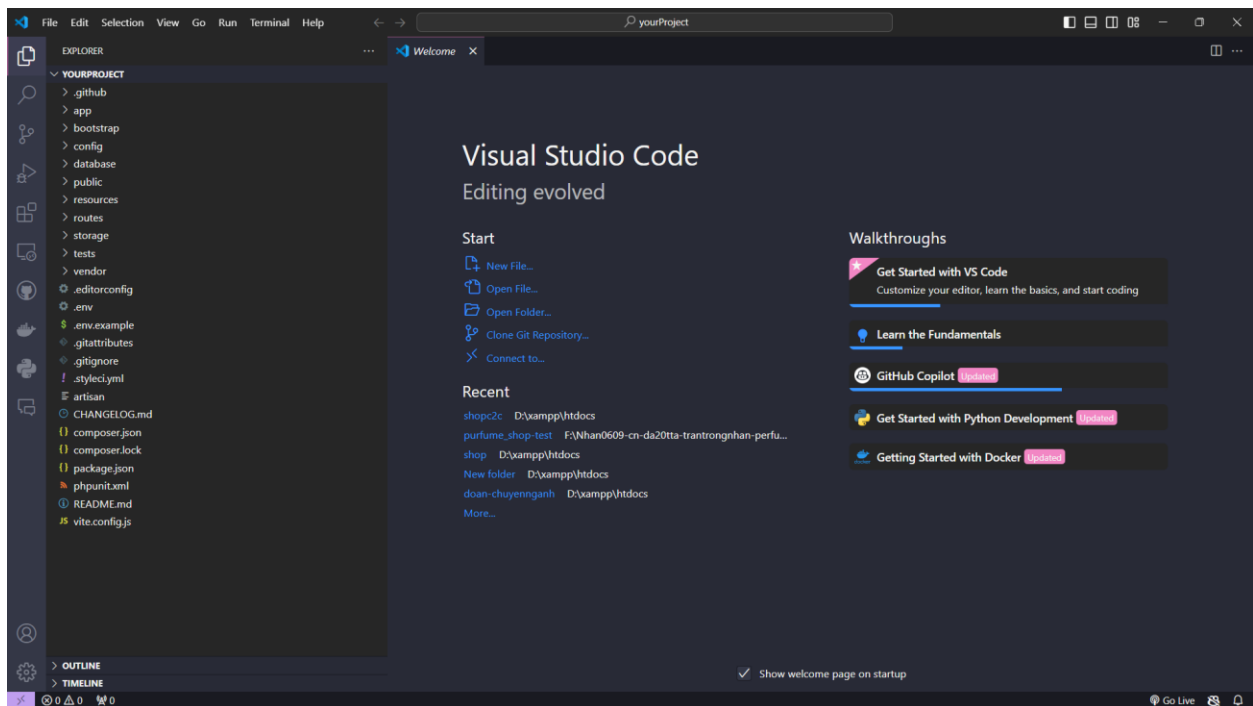
Mở cmd tại thư mục muốn cài đặt dự án và chạy lệnh: `composer create-project --prefer-dist laravel/laravel yourProject`

Trong đó `yourProject` là tên dự án của bạn.



Hình 3.2 Khởi tạo dự án Laravel

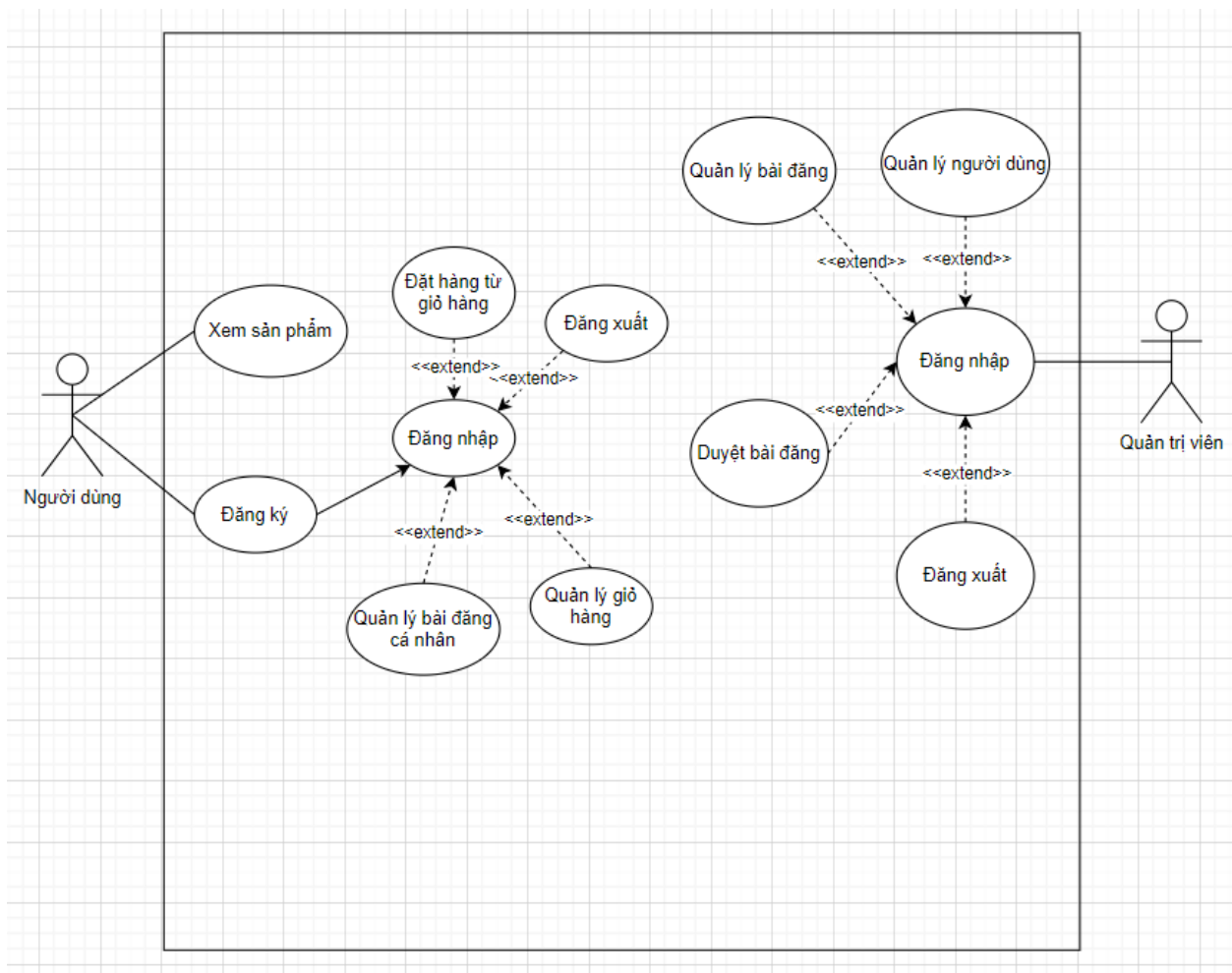
### Bước 3: Mở VS Code lên và mở thư mục dự án vừa tạo



Hình 3.3 Thư mục chứa dự án vừa tạo

## 3.2 Thiết kế dữ liệu

### 3.2.1 Lược đồ use case tổng quan của hệ thống

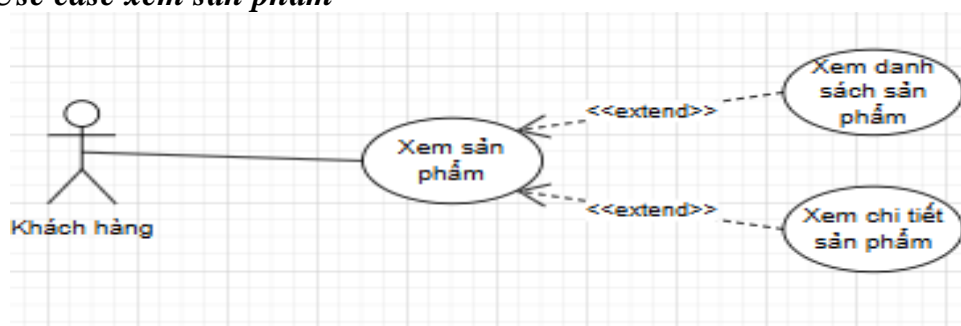


Hình 3.4 Mô hình use case tổng quan của hệ thống

Mô tả: Khách hàng có quyền đăng ký tài khoản, xem bài đăng, ngoài ra sau khi đăng nhập, khách hàng có thể quản lý bài đăng của bản thân, quản lý giỏ hàng của bản thân và đặt hàng từ giỏ hàng. Quản trị viên có quyền quản lý bài đăng, quản lý tài khoản, duyệt bài đăng.

### 3.2.2 Mô tả các use case

#### 3.2.2.1 Use case xem sản phẩm



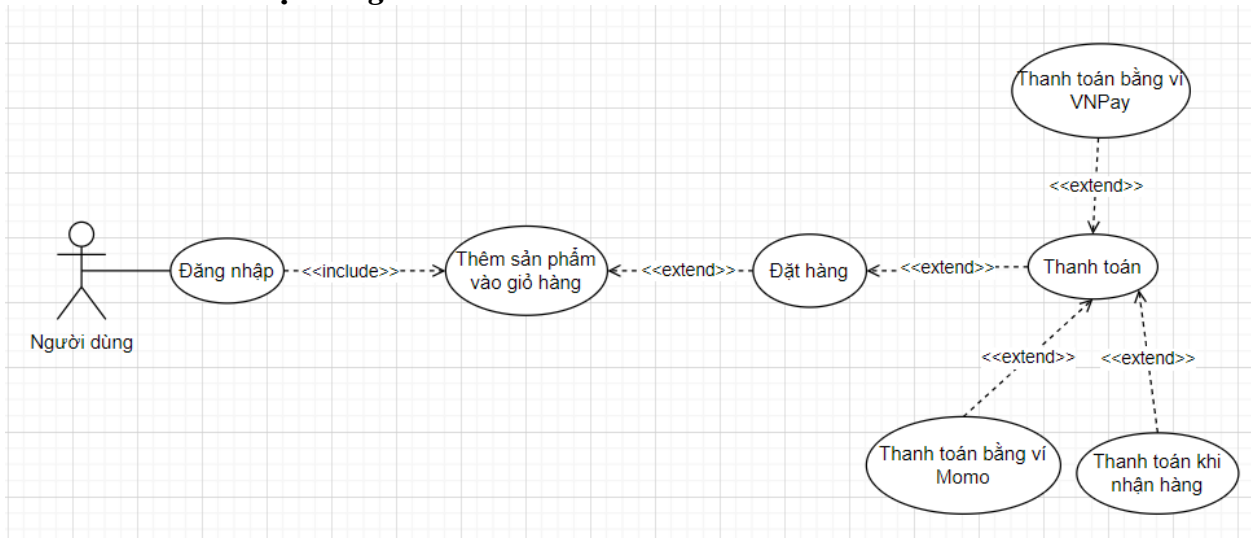
Hình 3.5 Use case xem sản phẩm

Tên use case: xem sản phẩm

Actor: khách hàng

Mô tả use case: use case này cho phép khách hàng có thể xem danh sách sản phẩm và chi tiết sản phẩm

### 3.2.2.2 Use case đặt hàng



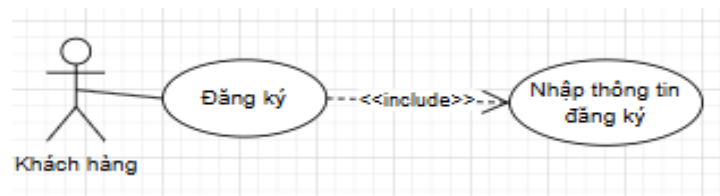
Hình 3.6 Use case đặt hàng

Tên use case: đặt hàng

Actor: khách hàng

Mô tả use case: use case này cho phép khách hàng có thể đặt hàng (có sản phẩm trong giỏ hàng) và thanh toán theo 3 phương thức thanh toán khi nhận hàng, thanh toán bằng ví Momo, thanh toán bằng ví VNPay.

### 3.2.2.3 Use case đăng ký



Hình 3.7 Use case đăng ký

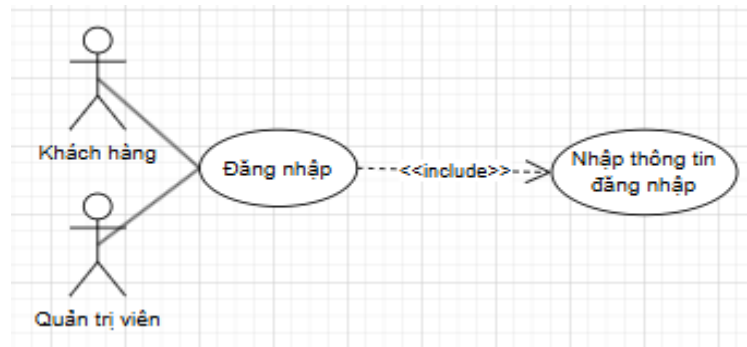
Tên use case: đăng ký

Actor: khách hàng

Mô tả use case: use case này cho phép khách hàng có thể đăng ký tài khoản



#### 3.2.2.4 Use case đăng nhập



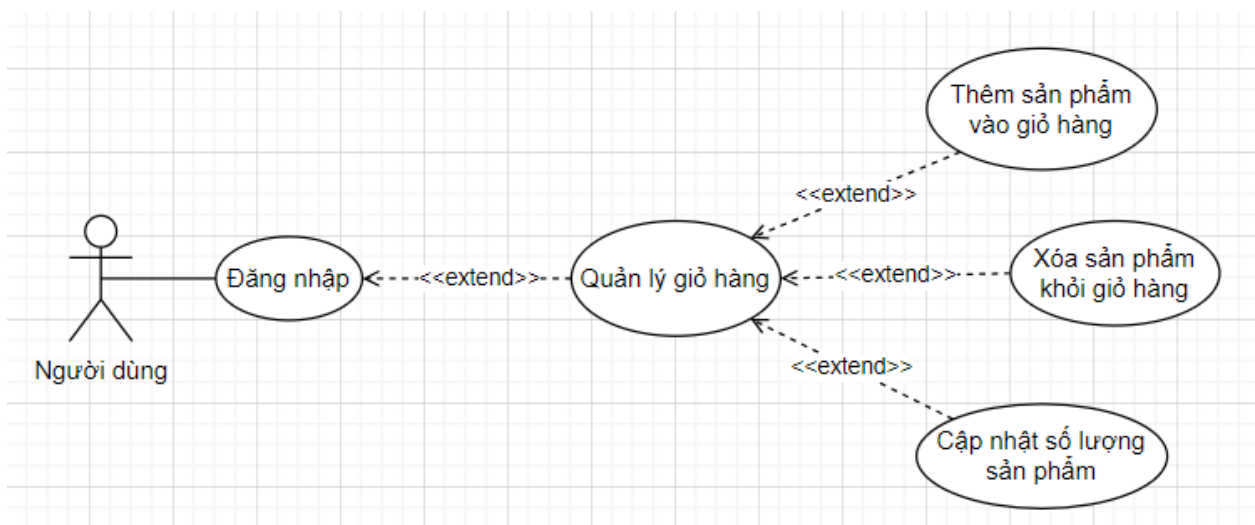
Hình 3.8 Use case đăng nhập

Tên use case: đăng nhập

Actor: khách hàng, quản trị viên

Mô tả use case: use case này cho phép khách hàng và quản trị viên có thể đăng nhập vào hệ thống.

#### 3.2.2.5 Use case quản lý giỏ hàng



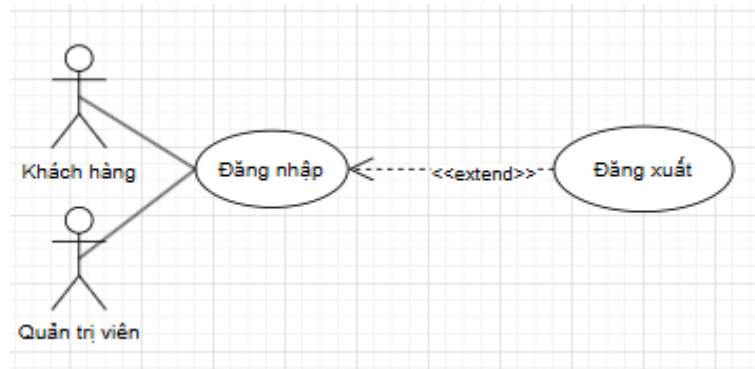
Hình 3.9 Use case quản lý giỏ hàng

Tên use case: quản lý giỏ hàng

Actor: khách hàng

Mô tả use case: use case này cho phép khách hàng có thể thêm sản phẩm vào giỏ hàng, cập nhật số lượng và xóa sản phẩm ra khỏi giỏ hàng.

### 3.2.2.8 Use case đăng xuất



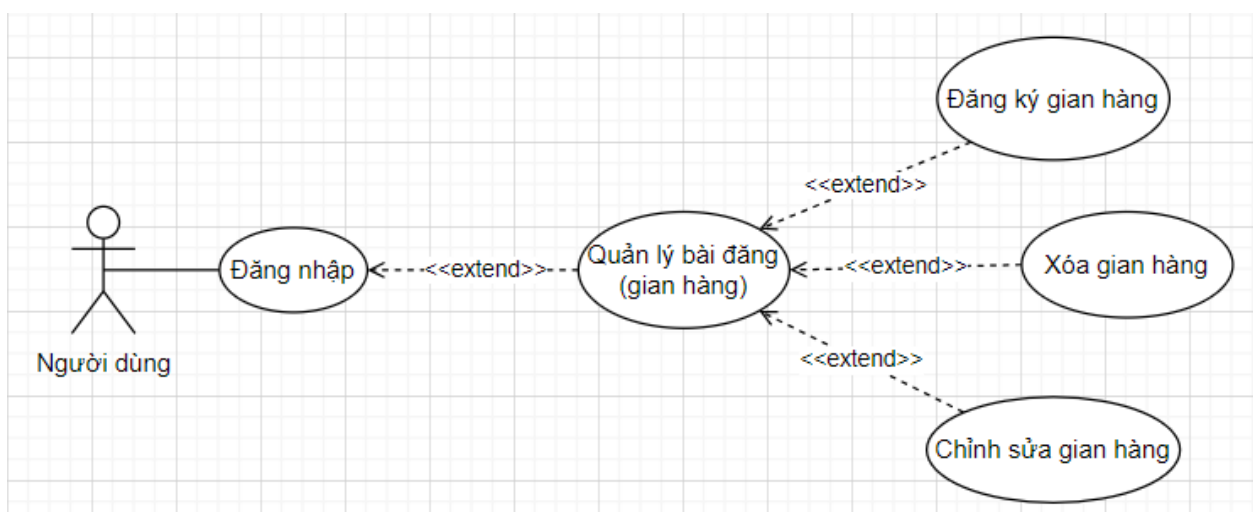
Hình 3.10 Use case đăng xuất

Tên use case: đăng xuất

Actor: khách hàng, quản trị viên

Mô tả use case: use case này cho phép khách hàng và quản trị viên có thể thoát khỏi hệ thống. Use case này chỉ thực hiện được sau khi khách hàng, quản trị viên đăng nhập.

### 3.2.2.9 Use case quản lý bài đăng



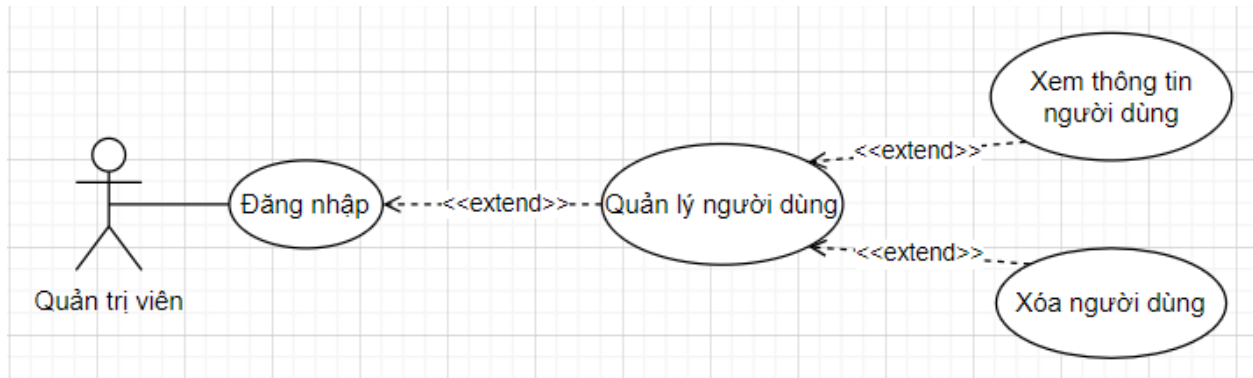
Hình 3.11 Use case Quản lý sản phẩm

Tên use case: quản lý sản phẩm

Actor: Người dùng

Mô tả use case: use case này cho phép người dùng có thể thêm, sửa, xóa bài đăng (gian hàng). Use case này chỉ thực hiện được sau khi người dùng đăng nhập.

### 3.2.2.10 Use case quản lý người dùng



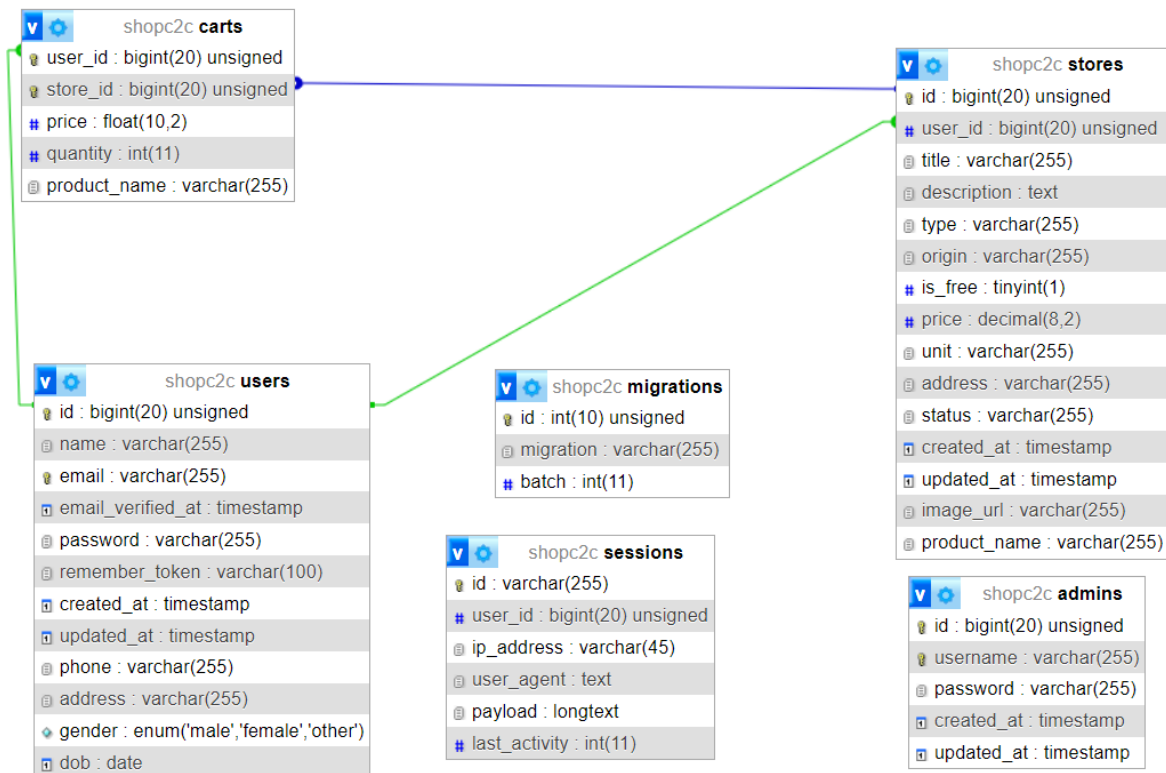
Hình 3.12 Use case Quản lý người dùng

Tên use case: quản lý người dùng

Actor: quản trị viên

Mô tả use case: use case này cho phép quản trị viên có thể xem và xóa người dùng. Use case này chỉ thực hiện được sau khi quản trị viên đăng nhập.

### 3.2.2 Mô hình dữ liệu



Hình 3.13 Mô hình cơ sở dữ liệu

### 3.2.3 Chi tiết thực thể

**Tên thực thể:** users

- Mô tả: lưu trữ thông tin người dùng, bao gồm: id, họ tên, email, mật khẩu, số điện thoại, địa chỉ, giới tính, ngày sinh.
- Chi tiết thực thể:

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	id	Mã người dùng	bigint	Khóa chính
2	name	Họ tên người dùng	varchar	Không thể rỗng
3	email	Địa chỉ email	varchar	Không thể rỗng
4	password	Mật khẩu	varchar	Không thể rỗng
5	phone	Số điện thoại	varchar	Không thể rỗng
6	address	Địa chỉ	varchar	Không thể rỗng
7	gender	Giới tính	enum	Không thể rỗng
8	dob	Ngày sinh	date	Không thể rỗng

**Tên thực thể:** stores

- Mô tả: lưu trữ thông tin gian hàng, bao gồm: id, id người dùng, tiêu đề, mô tả chi tiết, tên sản phẩm, loại sản phẩm, nguồn gốc, có miễn phí không, giá, đơn vị tính, địa chỉ bán, tình trạng, đường dẫn ảnh.
- Chi tiết thực thể:

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	id	Mã gian hàng	bigint	Khóa chính
2	user_id	Mã người dùng	bigint	Khóa ngoại
3	title	Tiêu đề	varchar	Không thể rỗng
4	description	Mô tả	text	Không thể rỗng
5	type	Loại sản phẩm	varchar	Không thể rỗng
6	origin	Nguồn gốc xuất xứ	varchar	Không thể rỗng
7	is_free	Có miễn phí không	inyint	Không thể rỗng
8	price	Giá	decimal	Có thể rỗng
9	unit	Đơn vị tính	varchar	Không thể rỗng
10	address	Địa chỉ bán	varchar	Không thể rỗng
11	status	Tình trạng	varchar	Không thể rỗng
12	image_url	Đường dẫn hình ảnh	varchar	Có thể rỗng
13	product_name	Tên sản phẩm	varchar	Không thể rỗng

**Tên thực thể:** carts

- Mô tả: lưu trữ giỏ hàng của người dùng, bao gồm: id người dùng, id gian hàng, giá, số lượng, tên sản phẩm.
- Chi tiết thực thể:

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	user_id	Mã người dùng	bigint	Khóa chính
2	store_id	Mã gian hàng	bigint	Khóa chính
3	price	Giá	float	Không thể rỗng
4	quantity	Số lượng	int	Không thể rỗng
5	product_name	Tên sản phẩm	varchar	Không thể rỗng

**Tên thực thể:** admins

- Mô tả: lưu trữ tài khoản quản trị viên, bao gồm: id, tên đăng nhập, mật khẩu.
- Chi tiết thực thể:

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	id	Mã số	bigint	Khóa chính
2	username	Tên đăng nhập	varchar	Không thể rỗng
3	password	Mật khẩu	varchar	Không thể rỗng

**Tên thực thể:** provinces

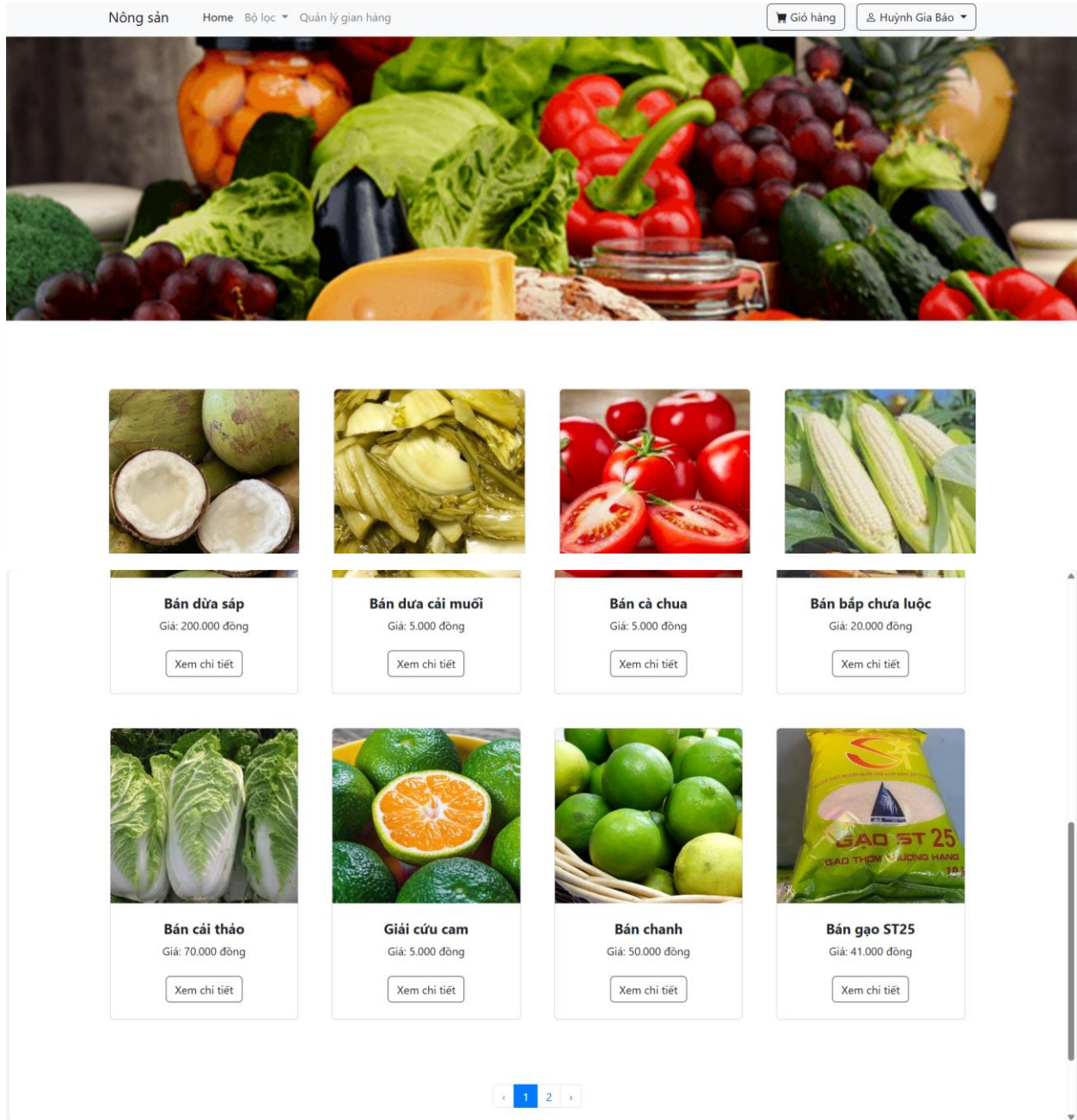
- Mô tả: lưu trữ danh sách các tỉnh thành, bao gồm: id, tên.
- Chi tiết thực thể:

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	id	Mã số	bigint	Khóa chính
2	name	Tên tỉnh thành	varchar	Không thể rỗng

## CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

### 4.1. Giao diện trang chủ

Đây là giao diện trang chủ ở đây người dùng có thể xem các sản phẩm và thêm sản phẩm vào giỏ hàng, ngoài ra còn có thể cập nhật thông tin cá nhân.

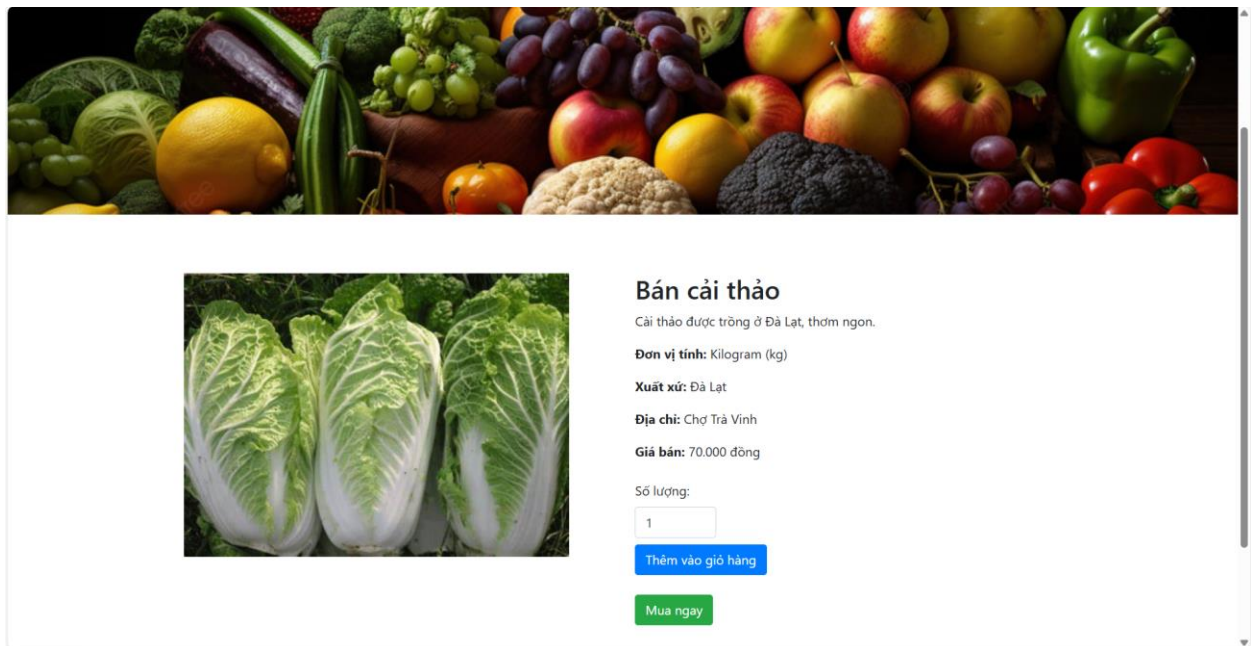


Hình 4.1 Giao diện trang chủ



## 4.2. Giao diện chi tiết sản phẩm

Đây là giao diện của trang chi tiết sản phẩm ở đây người dùng có thể xem chi tiết thông tin sản phẩm, tăng và giảm số lượng sản phẩm để thêm vào giỏ hàng

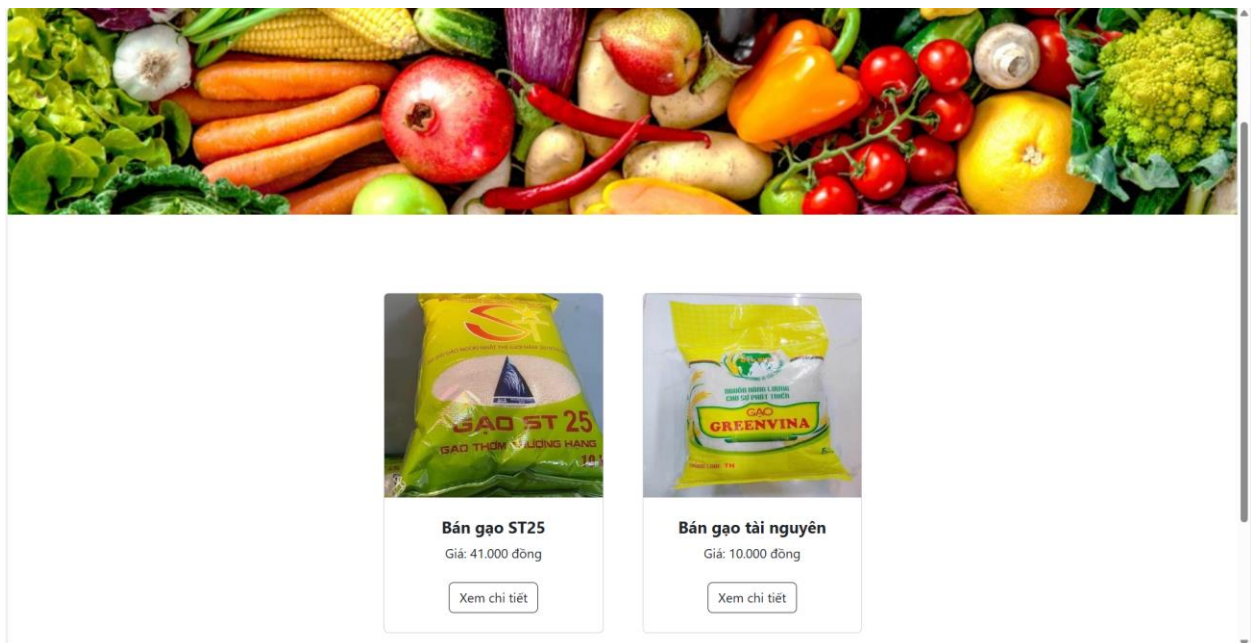


Hình 4.2 Giao diện chi tiết sản phẩm

## 4.3 Giao diện danh mục của từng loại hàng nông sản

### 4.3.1 Giao diện danh mục lúa gạo

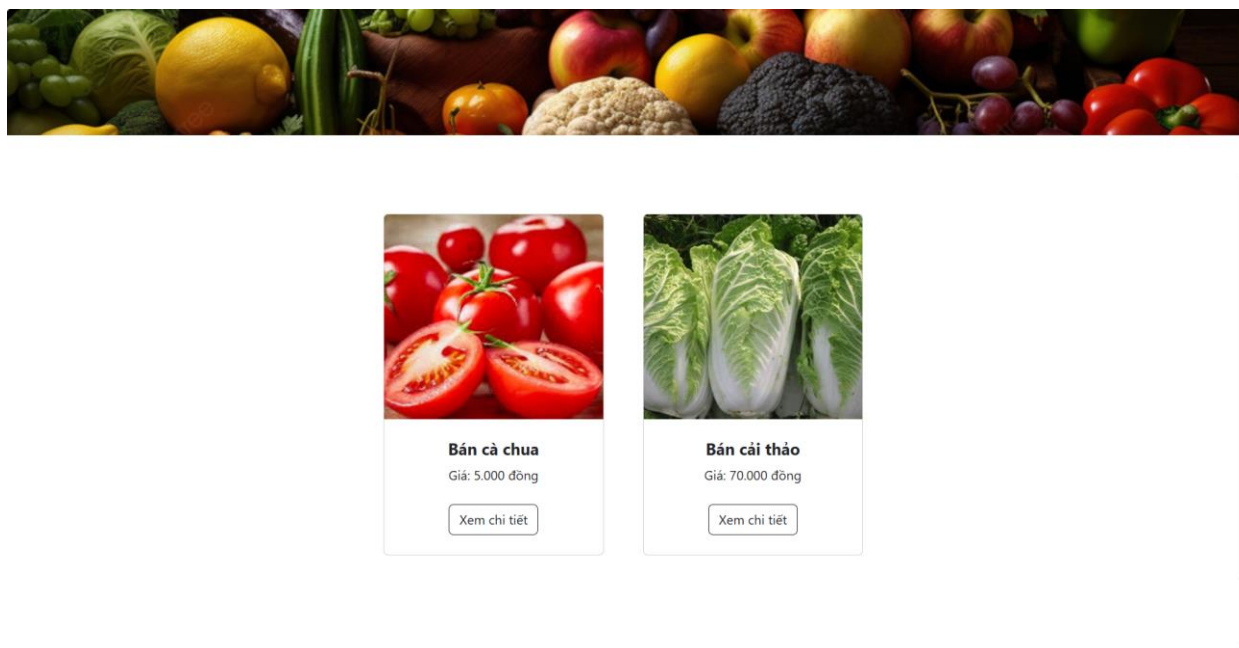
Đây là giao diện của trang danh mục lúa gạo



Hình 4.3 Giao diện danh mục lúa gạo

### 4.3.2 Giao diện danh mục rau củ

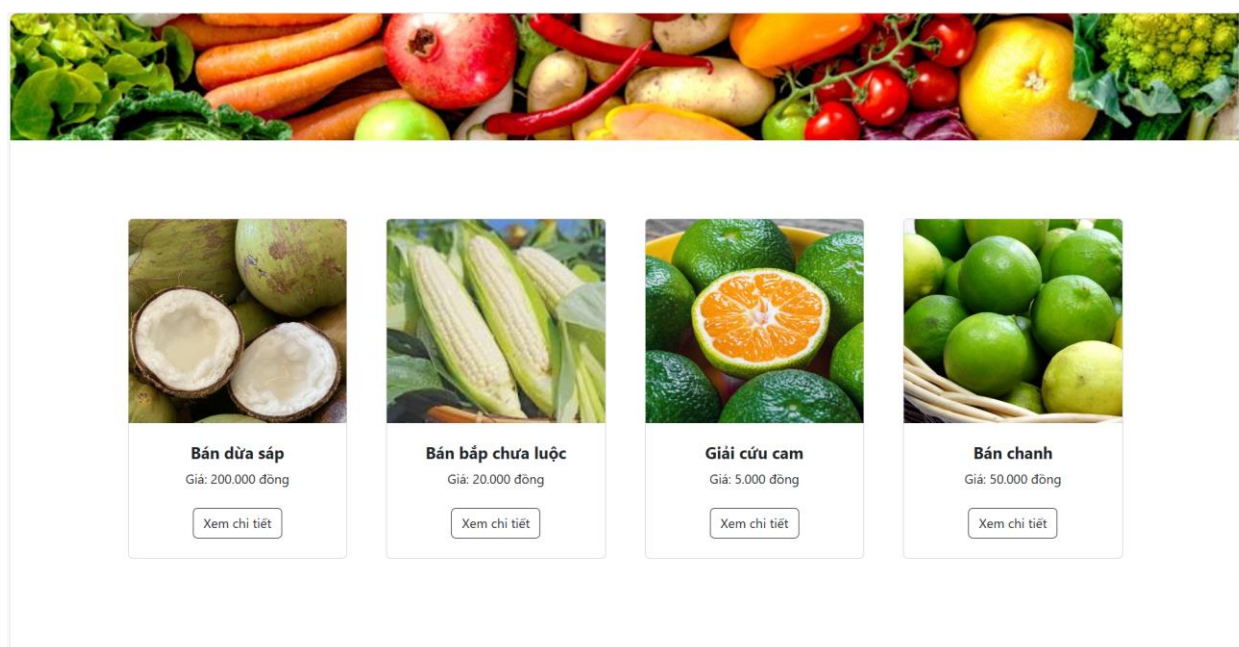
Đây là giao diện của trang danh mục rau củ



Hình 4.4 Giao diện danh mục rau củ

### 4.3.3 Giao diện danh mục trái cây

Đây là giao diện của trang danh mục trái cây

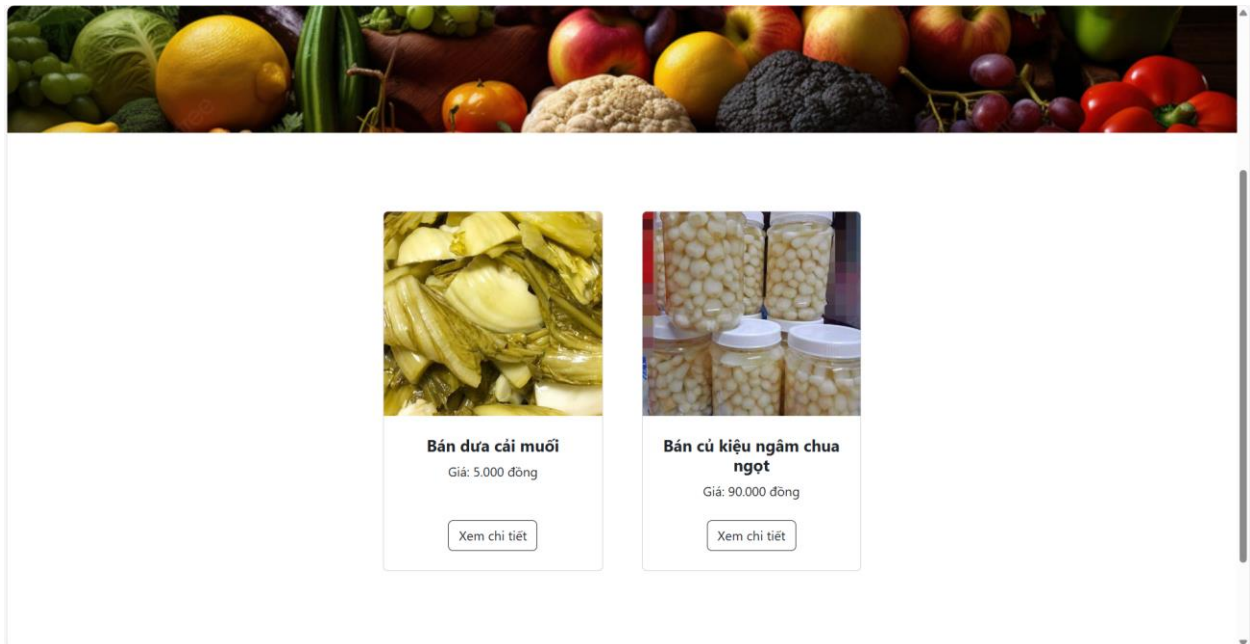


Hình 4.5 Giao diện danh mục trái cây



#### 4.3.4 Giao diện danh mục dưa muối

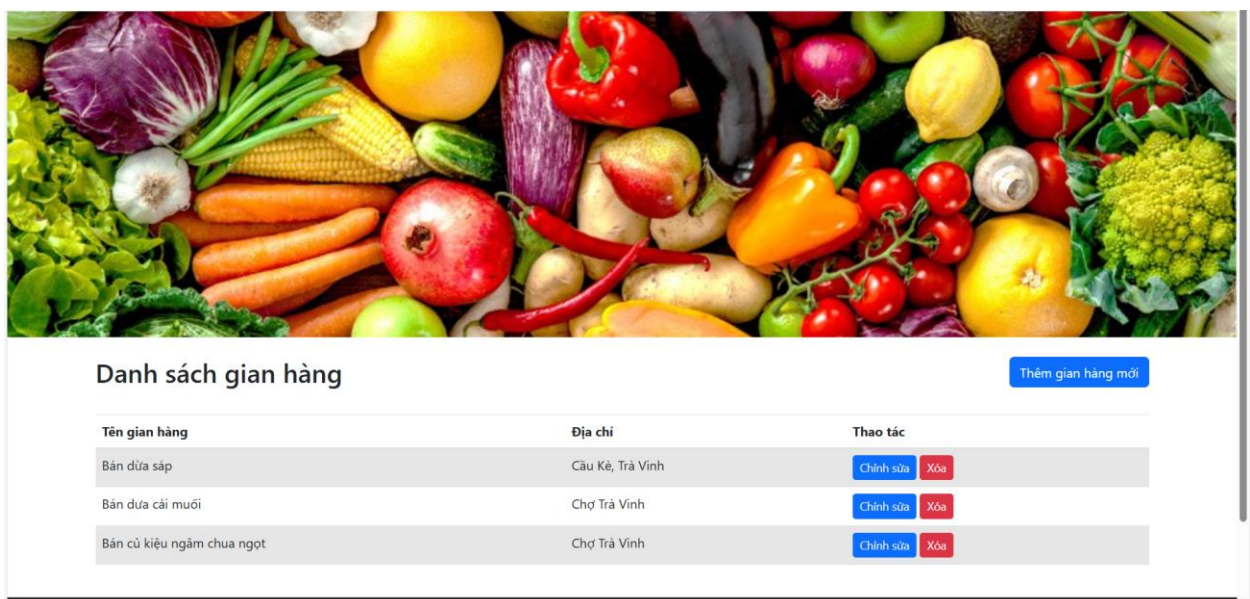
Đây là giao diện của trang danh mục dưa muối



Hình 4.6 Giao diện danh mục dưa muối

#### 4.4 Giao diện quản lý gian hàng

Đây là giao diện quản lý gian hàng ở đây người dùng có thể xem, sửa, xóa gian hàng đã đăng



Hình 4.7 Giao diện quản lý gian hàng

#### 4.5 Giao diện trang thông tin người dùng

Đây là trang thông tin người dùng ở đây người dùng có thể thay đổi mật khẩu

**Thông tin người dùng**

Tên  
Huỳnh Gia Bảo

Email  
bao@gmail.com

Số điện thoại  
0123456789

Địa chỉ  
Trà Vinh

[Đổi mật khẩu](#)

Mật khẩu  
\*\*\*\*\*

Xác nhận mật khẩu

[Lưu thay đổi](#)

**Gián hàng đã đăng**

- **Bán dừa sáp**  
Dừa sáp được trồng tại Cầu Kè, Trà Vinh
- **Bán dưa cải muối**  
Dưa cải muối nhà làm ngâm 6 tháng, đậm vị thơm ngon. Ăn liền chấm nước thịt, cá hay đem đi kho, nấu canh đều ngon.
- **Bán củ kiệu ngâm chua ngọt**  
Củ kiệu ngâm chua ngọt, vừa ăn.

Hình 4.8 Giao diện Thông tin người dùng

#### 4.6 Giao diện trang giỏ hàng

Đây là giao diện danh mục giỏ ở đây người dùng có thể xem, cập nhật số lượng, xóa sản phẩm khỏi giỏ hàng

**Giỏ hàng của bạn**

Sản phẩm	Giá	Số lượng	Thành tiền	Hành động
Dưa cải muối	5000	6	30000	<input type="text" value="6"/> <a href="#">Cập nhật</a> <a href="#">Xóa</a>
Củ kiệu	90000	2	180000	<input type="text" value="2"/> <a href="#">Cập nhật</a> <a href="#">Xóa</a>

[Xóa toàn bộ giỏ hàng](#)

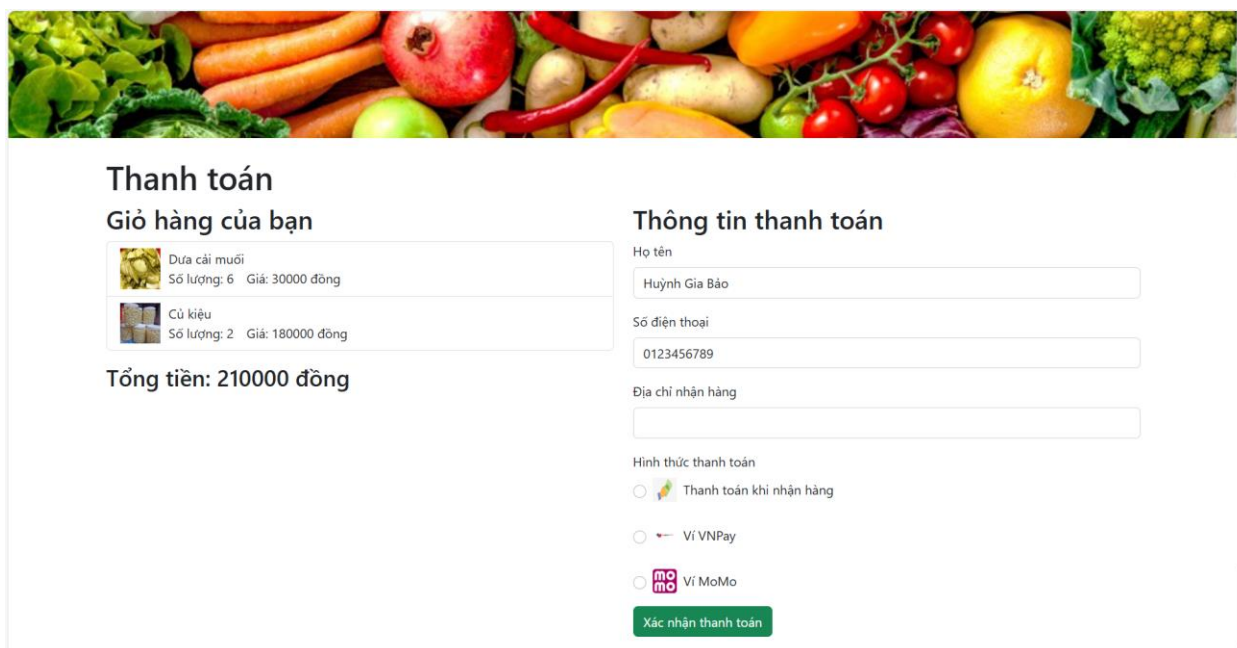
Tổng tiền: 210000 đồng

[Thanh toán](#)

Hình 4.9 Giao diện giỏ hàng

#### 4.7 Giao diện trang thanh toán

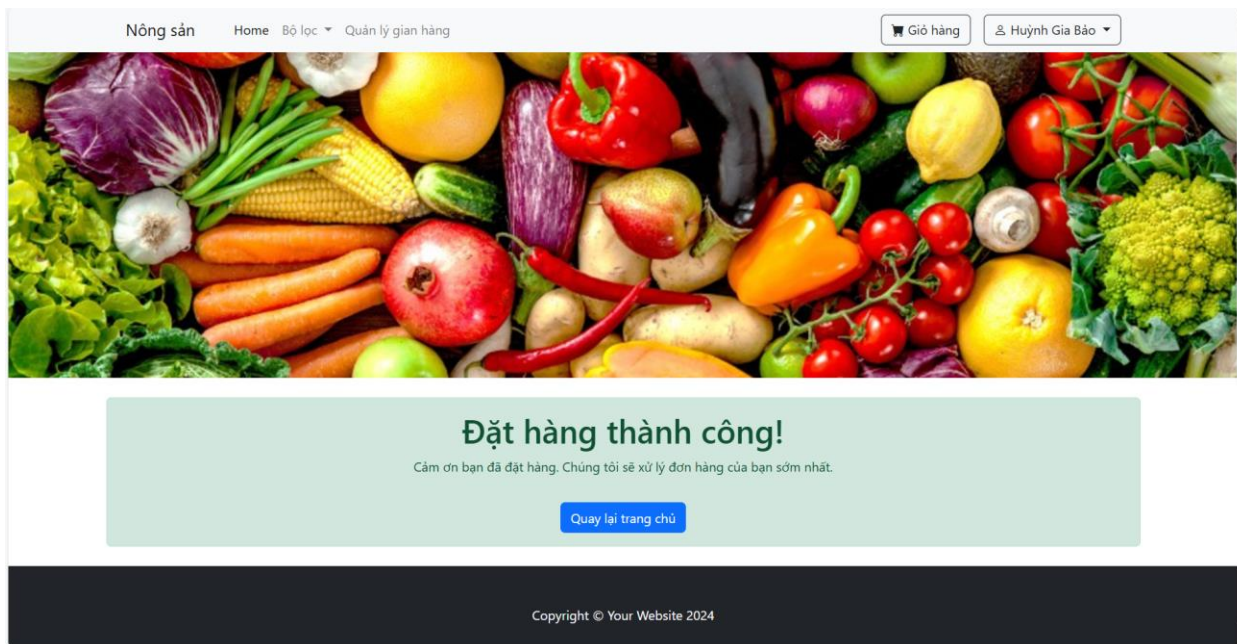
Đây là giao diện trang thanh toán ở đây người dùng có thể nhập thông tin thanh toán, chọn lựa phương thức thanh toán



Hình 4.10 Giao diện trang thanh toán

#### 4.8 Giao diện trang đặt hàng thành công


Đây là giao diện trang đặt hàng thành công



Hình 4.11 Giao diện trang Đặt hàng thành công

#### 4.9 Giao diện trang đăng nhập

Đây là giao diện trang đăng nhập ở đây người dùng có thể nhập thông tin tài khoản để đăng nhập



## Đăng nhập

Email

Mật khẩu

☐ Ghi nhớ tài khoản

**Đăng nhập**

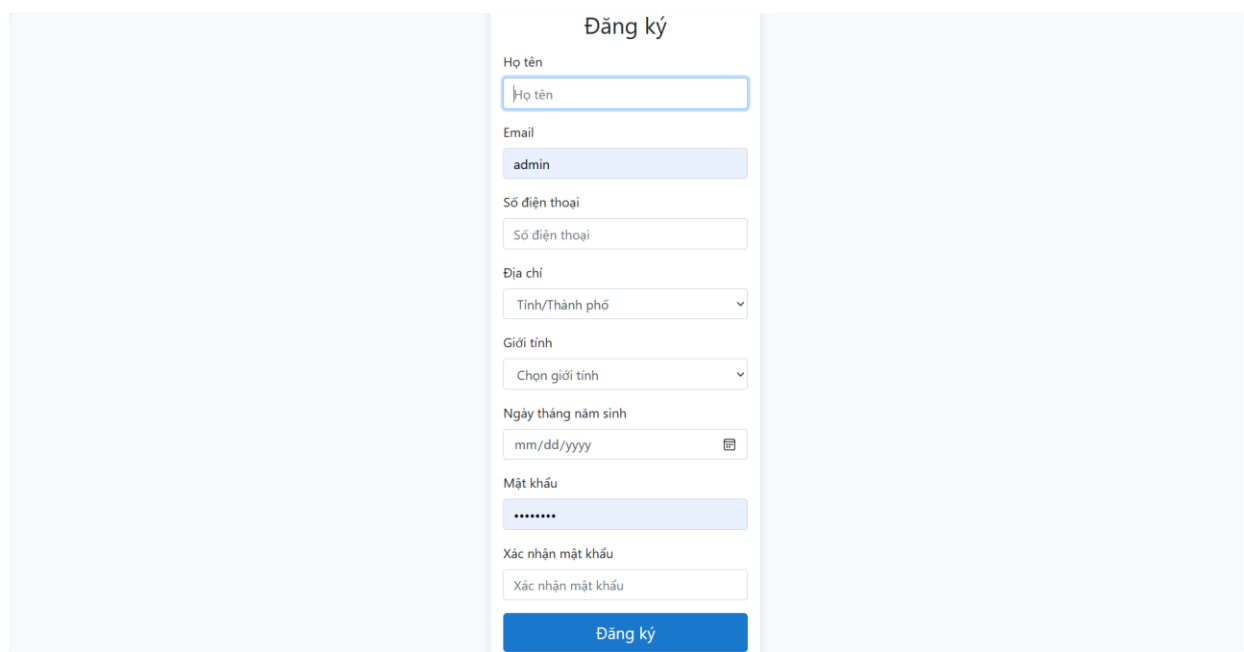
Chưa có tài khoản? [Đăng ký ngay](#)

© 2024

Hình 4.12 Giao diện trang Đăng nhập

#### 4.10 Giao diện trang đăng ký

Đây là giao diện trang đăng ký ở đây người dùng có thể nhập thông tin để đăng ký tài khoản



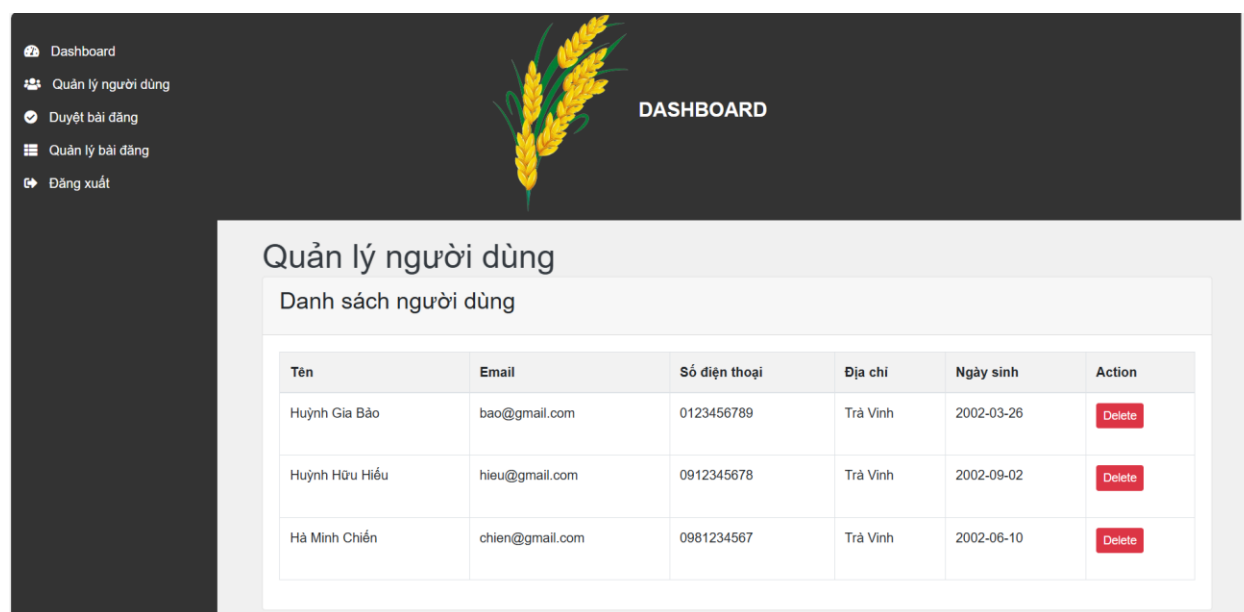
The registration form (Đăng ký) contains the following fields:

- Họ tên:
- Email:
- Số điện thoại:
- Địa chỉ:
- Giới tính:
- Ngày tháng năm sinh:
- Mật khẩu:
- Xác nhận mật khẩu:
- Đăng ký:

Hình 4.13 Giao diện trang Đăng ký

#### 4.11 Giao diện trang quản lý người dùng

Đây là giao diện trang quản lý người dùng ở đây quản trị viên có thể xem và xóa tài khoản người dùng



The dashboard (DASHBOARD) features a sidebar with navigation links: Dashboard, Quản lý người dùng, Duyệt bài đăng, Quản lý bài đăng, and Đăng xuất. The main content area displays the 'Quản lý người dùng' section with a table of users.

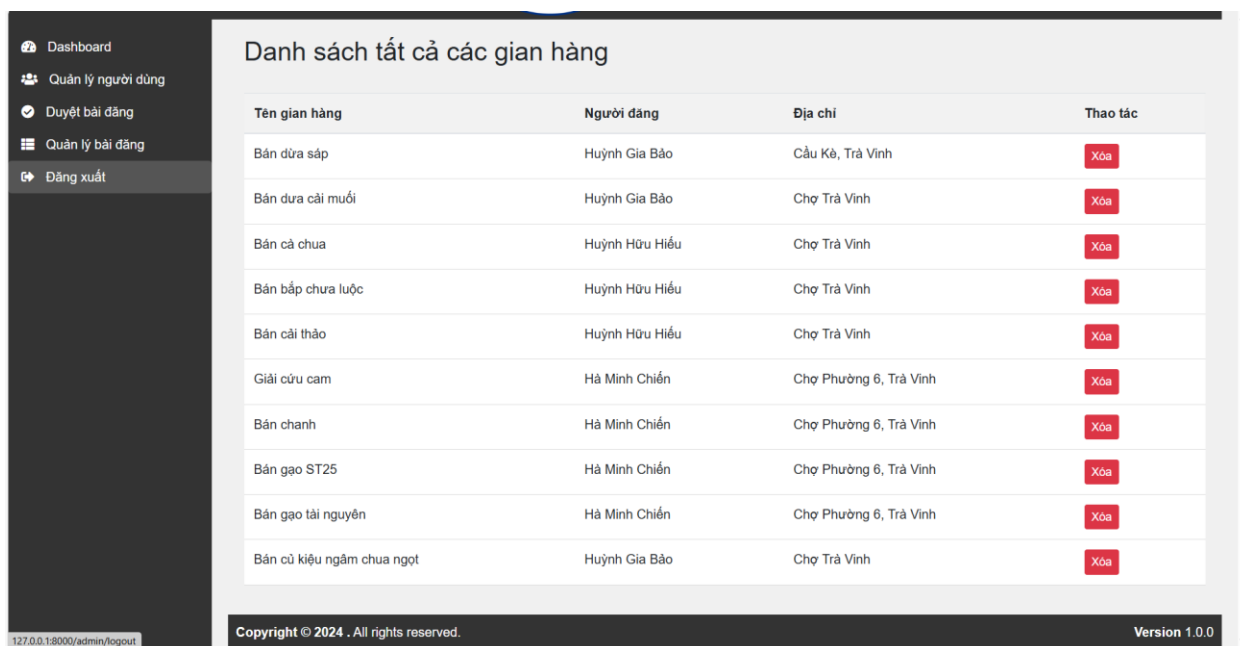
Tên	Email	Số điện thoại	Địa chỉ	Ngày sinh	Action
Huỳnh Gia Bảo	bao@gmail.com	0123456789	Trà Vinh	2002-03-26	<a href="#">Delete</a>
Huỳnh Hữu Hiếu	hiuu@gmail.com	0912345678	Trà Vinh	2002-09-02	<a href="#">Delete</a>
Hà Minh Chiến	chien@gmail.com	0981234567	Trà Vinh	2002-06-10	<a href="#">Delete</a>

Hình 4.14 Giao diện trang Quản lý người dùng



## 4.12 Giao diện trang quản lý tất cả gian hàng

Đây là giao diện trang quản lý sản phẩm ở đây quản trị viên có thể xem, xóa gian hàng của tất cả người dùng

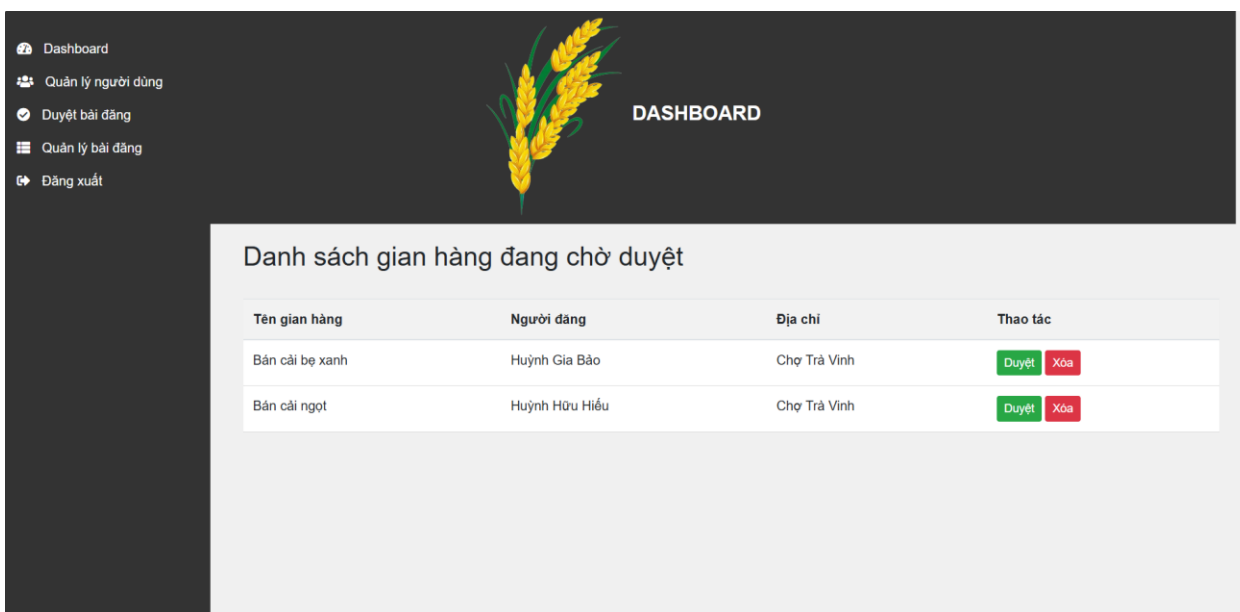


Tên gian hàng	Người đăng	Địa chỉ	Thao tác
Bán dừa sáp	Huỳnh Gia Bảo	Cầu Kè, Trà Vinh	Xóa
Bán dừa cải muối	Huỳnh Gia Bảo	Chợ Trà Vinh	Xóa
Bán cà chua	Huỳnh Hữu Hiếu	Chợ Trà Vinh	Xóa
Bán bắp chưa luộc	Huỳnh Hữu Hiếu	Chợ Trà Vinh	Xóa
Bán cải thảo	Huỳnh Hữu Hiếu	Chợ Trà Vinh	Xóa
Giải cứu cam	Hà Minh Chiến	Chợ Phường 6, Trà Vinh	Xóa
Bán chanh	Hà Minh Chiến	Chợ Phường 6, Trà Vinh	Xóa
Bán gạo ST25	Hà Minh Chiến	Chợ Phường 6, Trà Vinh	Xóa
Bán gạo tẻ nguyên	Hà Minh Chiến	Chợ Phường 6, Trà Vinh	Xóa
Bán củ kiệu ngâm chua ngọt	Huỳnh Gia Bảo	Chợ Trà Vinh	Xóa

Hình 4.15 Giao diện trang Quản lý sản phẩm

## 4.13 Giao diện trang duyệt bài đăng

Đây là giao diện trang duyệt bài đăng ở đây quản trị viên có thể thấy được các bài đăng chưa được duyệt của tất cả người dùng, quản trị viên có quyền duyệt hoặc không.



Tên gian hàng	Người đăng	Địa chỉ	Thao tác
Bán cải bẹ xanh	Huỳnh Gia Bảo	Chợ Trà Vinh	Duyệt Xóa
Bán cải ngọt	Huỳnh Hữu Hiếu	Chợ Trà Vinh	Duyệt Xóa

Hình 4.16 Giao diện trang duyệt bài đăng

#### 4.14 Giao diện trang thanh toán Momo

Đây là giao diện trang quản lý sản phẩm ở đây quản trị viên có thể xem, xóa gian hàng của tất cả người dùng

The screenshot displays the MoMo payment interface. At the top, there is a header with the MoMo logo and the text "Cổng thanh toán MoMo". Below the header, a blue banner contains the text "Diễn thông tin thẻ để thanh toán. Lưu ý: Không tắt trình duyệt này khi chưa hoàn tất.".

The main content area is divided into two columns. The left column, titled "Thông tin đơn hàng", contains the following information: "Nhà cung cấp" (Supplier) with a MoMo logo and the text "DO NOT USE FOR INTERNAL TEST. Merchant integrate test payment.", "Mã đơn hàng" (Order code) "1720004816", "Mô tả" (Description) "Thanh toán qua MoMo", and "Số tiền" (Amount) "200.000đ". Below this, a yellow box indicates the remaining time for the order: "Đơn hàng sẽ hết hạn sau:" followed by a timer showing "69" days, "10" hours, "39" minutes, and "54" seconds. At the bottom of the left column, there are links for "Hướng dẫn thanh toán an toàn" (Safe payment guide) and "Quay về" (Return).

The right column, titled "Nhập thông tin thẻ để thanh toán" (Enter card information to pay), features a visual representation of a pink Napas card. Below the card, there are input fields for "Số thẻ" (Card number), "Tên chủ thẻ" (Cardholder name), "Ngày phát hành" (Issuance date), and "Số điện thoại" (Phone number). A prominent pink button labeled "Thanh Toán" (Pay) is located at the bottom of this section.

Hình 4.17 Giao diện trang thanh toán bằng ví Momo

## CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết luận

Trong đồ án "Xây dựng sàn giao dịch nông sản theo mô hình C2C", tôi đã hoàn thành các mục tiêu đề ra và đạt được những kết quả quan trọng như sau:

- Tôi đã nghiên cứu và ứng dụng thành công Laravel Framework và MySQL để xây dựng sàn giao dịch nông sản.
- Giao diện người dùng (UI) và trải nghiệm người dùng (UX) được thiết kế thân thiện và dễ sử dụng, giúp người dùng dễ dàng mua sắm sản phẩm.
- Các chức năng cơ bản như đăng nhập/dăng ký, hiển thị danh sách sản phẩm, thêm vào giỏ hàng và thanh toán đã được triển khai hoàn chỉnh.
- Tính năng bộ lọc giúp người dùng tìm kiếm sản phẩm theo loại sản phẩm.
- Hệ thống giỏ hàng và thanh toán đa dạng, hỗ trợ nhiều phương thức thanh toán như thanh toán trực tuyến, thanh toán khi nhận hàng (COD).

### 5.2. Hướng phát triển

Nghiên cứu và phát triển thêm các tính năng nâng cao như hệ thống đánh giá và nhận xét sản phẩm từ người dùng, tích hợp AI để gợi ý sản phẩm, và tối ưu hóa hiệu suất website.

Khám phá và ứng dụng các công nghệ mới như GraphQL, TypeScript để cải thiện hiệu suất và bảo mật cho website.

Tăng cường khả năng tương tác người dùng qua các kênh truyền thông xã hội và ứng dụng di động.

Thêm chức năng xác nhận email của người dùng.

Thêm chức năng quản lý đơn hàng, xác nhận đơn hàng.

Thêm chức năng tổng hợp số giao dịch mà người dùng đã thực hiện.



## DANH MỤC TÀI LIỆU THAM KHẢO

### Sách/Giáo trình:

- [1] John Dean, Web Programming with HTML5, CSS, and JavaScript, Jones & Bartlett Learning, 2018.
- [2] Silvio Moreto, Bootstrap 4 By Example, O'Reilly Media, 2019.
- [3] Luke Welling, PHP and MySQL Web Development (Developer's Library), Addison-Wesley, 2016.
- [4] Jennifer Niederst Robbins, Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics; 5th Edition, O'Reilly Media, 2018.

### Website:

- [1] Topdev, “Ngôn ngữ PHP”, <https://topdev.vn/blog/ngon-ngu-lap-trinh-php-la-gi-tat-tan-tat-nhung-dieu-ban-can-biet-ve-php/>
- [2] Mona.Media, “Laravel framework”, <https://mona.media/laravel-la-gi/>
- [3] Mắt bão, “Laravel framework”, <https://wiki.matbao.net/laravel-la-gi-huong-dan-cai-dat-laravel-nhanh-chong/#framework-la-gi-php-framework-la-gi>
- [4] Hưng Nguyễn, “Mô hình MVC”, <https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/>