

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO GIỮA KÌ
HỌC PHẦN CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

XÂY DỰNG WEBSITE BÁN GIÀY SỬ DỤNG
FRAMEWORK LARAVEL

Người hướng dẫn : PHẠM THỊ DUNG
Sinh viên thực hiện: LƯƠNG THẾ DŨNG
Mã số sinh viên : 2351220152
Lớp : 23CT4

Đà Nẵng, tháng 11 năm 2025

LỜI NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN

[illegible]

[illegible]

TÓM TẮT

Tên đề tài: Xây dựng website bán giày sử dụng framework Laravel

Sinh viên thực hiện: Lương Thế Dũng

Mã số sinh viên: 2351220152

Lớp : 23CT4

Bài báo cáo trình bày thiết kế và triển khai một website bán giày sử dụng framework Laravel nhằm hỗ trợ quản lý danh mục sản phẩm, giỏ hàng, đặt hàng và thanh toán trực tuyến. Hệ thống được xây dựng theo kiến trúc MVC, tận dụng cơ chế routing, migration, Eloquent ORM và hệ thống Blade Template của Laravel để đảm bảo tính mở rộng, bảo mật và dễ bảo trì.

Thành phần chính bao gồm: mô-đun quản trị cho phép thêm, sửa, xóa sản phẩm và quản lý đơn hàng; mô-đun người dùng cho phép duyệt sản phẩm, lọc theo thuộc tính, thêm sản phẩm vào giỏ và thực hiện đặt hàng; tích hợp phương thức thanh toán giả lập và gửi email xác nhận đơn hàng.

Trong quá trình phát triển, tác giả đã áp dụng các biện pháp bảo mật cơ bản như xác thực người dùng, phân quyền, bảo vệ CSRF và xác thực dữ liệu đầu vào. Kết quả thử nghiệm chức năng cho thấy hệ thống hoạt động ổn định, đáp ứng được các nghiệp vụ thương mại điện tử cơ bản, đồng thời dễ dàng mở rộng để tích hợp các cổng thanh toán thực tế và thêm các tính năng như khuyến mãi, đánh giá sản phẩm.

Website này có thể được sử dụng làm nền tảng mẫu cho các cửa hàng bán lẻ trực tuyến quy mô nhỏ và là tài liệu tham khảo cho người muốn học triển khai ứng dụng web bằng Laravel.

LỜI MỞ ĐẦU

1. Lý do chọn đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, thương mại điện tử đã và đang trở thành xu hướng tất yếu trong hoạt động kinh doanh. Theo báo cáo của Hiệp hội Thương mại điện tử Việt Nam, thị trường thương mại điện tử Việt Nam đạt giá trị khoảng 20,5 tỷ USD năm 2024, tăng trưởng 25% so với năm trước. Đặc biệt, ngành hàng thời trang và giày dép chiếm tỷ trọng lớn trong tổng doanh thu thương mại điện tử.

Nhận thấy tiềm năng to lớn của thị trường giày dép trực tuyến, cùng với việc người tiêu dùng ngày càng ưa chuộng mua sắm online nhờ tính tiện lợi, đa dạng và có thể so sánh giá dễ dàng, em đã quyết định chọn đề tài "Xây dựng Website bán giày sử dụng Framework Laravel".

Việc lựa chọn Laravel làm framework phát triển xuất phát từ những ưu điểm vượt trội của nó: cú pháp rõ ràng, dễ học, hệ sinh thái phong phú, bảo mật cao và được cộng đồng developer toàn cầu ủng hộ mạnh mẽ. Laravel giúp quá trình phát triển web trở nên nhanh chóng và hiệu quả hơn, đồng thời đảm bảo code được tổ chức khoa học theo mô hình MVC.

Đề tài này không chỉ giúp em củng cố và vận dụng kiến thức đã học về lập trình web, cơ sở dữ liệu, phân tích thiết kế hệ thống mà còn tạo cơ hội để em tiếp cận với công nghệ thực tế đang được sử dụng rộng rãi trong các doanh nghiệp công nghệ hiện nay.

2. Mục tiêu nghiên cứu

2.1. Mục tiêu chung

Xây dựng một hệ thống website thương mại điện tử chuyên bán giày hoàn chỉnh, đáp ứng nhu cầu mua sắm trực tuyến của khách hàng và quản lý vận hành của người bán.

2.2. Mục tiêu cụ thể

Về chức năng: Xây dựng đầy đủ các chức năng cần thiết cho một website bán hàng bao gồm: quản lý sản phẩm, giỏ hàng, thanh toán, đơn hàng, người dùng, và hệ thống quản trị admin.

Về công nghệ: Thành thạo trong việc sử dụng Framework Laravel, áp dụng mô hình MVC, tích hợp các công nghệ web hiện đại như Bootstrap, jQuery, Ajax.

Về thiết kế: Tạo giao diện thân thiện, responsive trên mọi thiết bị, đảm bảo trải nghiệm người dùng tốt nhất.

Về bảo mật: Áp dụng các biện pháp bảo mật như mã hóa mật khẩu, xác thực người dùng, ngăn chặn SQL Injection, XSS attacks.

Về hiệu năng: Tối ưu hóa tốc độ tải trang, xử lý dữ liệu hiệu quả, sử dụng cache và lazy loading.

3. Đối tượng và phạm vi nghiên cứu

3.1. Đối tượng nghiên cứu

Framework Laravel phiên bản 10.x

Hệ quản trị cơ sở dữ liệu MySQL

Các công nghệ web: HTML5, CSS3, JavaScript, Bootstrap 5

Kiến trúc MVC và các design patterns

Hệ thống thanh toán trực tuyến (COD, chuyển khoản ngân hàng)

3.2. Phạm vi nghiên cứu

Website được phát triển với các phạm vi sau:

Phạm vi chức năng:

Chức năng dành cho khách hàng: đăng ký/đăng nhập, xem sản phẩm, tìm kiếm/lọc sản phẩm, thêm vào giỏ hàng, đặt hàng, theo dõi đơn hàng, đánh giá sản phẩm

Chức năng dành cho admin: quản lý danh mục, sản phẩm, đơn hàng, khách hàng, thống kê doanh thu, quản lý mã giảm giá

Phạm vi kỹ thuật:

Phát triển trên nền tảng web, không bao gồm ứng dụng mobile native

Hỗ trợ thanh toán COD và chuyển khoản, chưa tích hợp ví điện tử

Chỉ áp dụng cho thị trường Việt Nam

Phạm vi thời gian:

Thời gian thực hiện: 3 tuần/tháng

Giai đoạn 1: Phân tích và thiết kế (2 tuần)

Giai đoạn 2: Phát triển chức năng (6 tuần)

Giai đoạn 3: Kiểm thử và hoàn thiện (2 tuần)

4. Phương pháp thực hiện

Nghiên cứu tài liệu về Laravel Framework, PHP và các công nghệ web

Tham khảo các website thương mại điện tử hiện có để học hỏi về UX/UI và chức năng

Nghiên cứu các mô hình, kiến trúc hệ thống thương mại điện tử

Tìm hiểu về bảo mật web và các best practices trong phát triển web

CAM ĐOAN

Tôi xin cam đoan rằng đề tài “Website bán giày sử dụng Framework Laravel” là kết quả nghiên cứu và thực hiện của riêng tôi, dưới sự hướng dẫn của giảng viên hướng dẫn. Các số liệu, kết quả và nội dung trong báo cáo được thu thập, phân tích trung thực, không sao chép hay vi phạm bản quyền của bất kỳ cá nhân hoặc tổ chức nào khác.

Những phần nội dung, tài liệu tham khảo được sử dụng trong quá trình thực hiện đề tài đều được trích dẫn rõ ràng nguồn gốc. Tôi hoàn toàn chịu trách nhiệm về tính trung thực và chính xác của nội dung trong báo cáo này.

Sinh viên thực hiện

Lương Thế Dũng

MỤC LỤC

LỜI NHẬN XÉT CỦA NGƯỜI HƯỚNG DẪN.....	i
NHẬN XÉT CỦA NGƯỜI PHẢN BIỆN.....	ii
TÓM TẮT	iii
LỜI MỞ ĐẦU	iv
CHƯƠNG I: NỀN TẢNG CÔNG NGHỆ VÀ THIẾT KẾ HỆ THỐNG	1
1.1. Nền tảng công nghệ.....	1
1.2. Mô hình hoạt động và thuật toán của hệ thống	7
1.3. Thiết kế cơ sở dữ liệu của hệ thống.....	11
1.4. Tiểu kết chương I	14
CHƯƠNG II: THỰC NGHIỆM.....	17
2.1. Chức năng Đăng ký	17
2.2. Chức năng Đăng nhập	19
2.3. Chức năng Giỏ hàng.....	20
2.4. Chức năng Thanh toán.....	22
2.5. Chức năng Liên hệ.....	25
2.6. Chức năng Tải xuống thông tin cá nhân.....	26
2.7. Chức năng Quản lý người dùng của Admin.....	28
2.8. Chức năng Quản lý Sản phẩm của Admin	31
2.9. Chức năng Quản lý Tin nhắn của Admin.....	33
2.10. Chức năng Quản lý Đơn đặt hàng của Admin	34
2.11. Chức năng Dashboard (Trang tổng quan Admin)	36
2.12. Tiểu kết chương II	39
TÀI LIỆU THAM KHẢO.....	40

DANH MỤC HÌNH ẢNH

Hình 1.1. Sơ đồ tuần tự chức năng “Đăng ký”	8
Hình 1.2. Sơ đồ tuần tự chức năng “Đăng nhập”	8
Hình 1.3. Sơ đồ tuần tự chức năng “Đăng ký”	9
Hình 1.4. Sơ đồ tuần tự chức năng “Thanh toán”	9
Hình 1.5. Mô hình thuật toán “Đăng ký – Đăng nhập”	10
Hình 1.6. Dữ liệu trong bảng User	13
Hình 1.7. Dữ liệu trong bảng Products	14
Hình 1.8. Dữ liệu trong bảng Upload_files	14
Hình 1.9. Dữ liệu trong bảng Contacts	15
Hình 1.10. Dữ liệu trong bảng Orders.....	15
Hình 1.11. Dữ liệu trong bảng OrderItems	16
Hình 2.1. Giao diện “Đăng ký tài khoản”	17
Hình 2.2. Giao diện “Đăng nhập tài khoản”	19
Hình 2.3. Giao diện “Giỏ hàng”	21
Hình 2.4. Giao diện “Thanh toán”	23
Hình 2.5. Giao diện “Liên hệ”	25
Hình 2.6. Giao diện “Thông tin cá nhân”	27
Hình 2.7. Giao diện Quản lý người dùng của Admin	28
Hình 2.8. Giao diện Thêm người dùng của Admin.....	29
Hình 2.9. Giao diện Chỉnh sửa người dùng của Admin.....	29
Hình 2.10. Giao diện Xóa người dùng của Admin	30
Hình 2.11. Giao diện Quản lý sản phẩm của Admin	31
Hình 2.12. Giao diện chức năng Quản lý tin nhắn.....	33
Hình 2.13. Giao diện chức năng Quản lý đơn đặt hàng	35
Hình 2.14. Giao diện Dashboard của Admin	37

DANH MỤC BẢNG

Bảng 1.1: Database bảng User	11
Bảng 1.2: Database bảng Upload_files	11
Bảng 1.3: Database bảng Products	12
Bảng 1.4: Database bảng Contacts	12
Bảng 1.5: Database bảng Orders	12
Bảng 1.6: Database bảng OrderItems	13

CHƯƠNG I: NỀN TẢNG CÔNG NGHỆ VÀ THIẾT KẾ HỆ THỐNG

1.1. Nền tảng công nghệ

1.1.1. Framework Laravel

Laravel là một trong những framework PHP mã nguồn mở hiện đại và được sử dụng rộng rãi nhất hiện nay. Framework này được Taylor Otwell phát triển và chính thức giới thiệu vào năm 2011. Với định hướng xây dựng một nền tảng phát triển web “thanh lịch và diễn đạt rõ ràng”, Laravel cung cấp cho lập trình viên một môi trường làm việc linh hoạt, mạnh mẽ, giàu tính tổ chức và thân thiện với người sử dụng. Nhờ triết lý “developer-friendly”, Laravel tạo điều kiện cho lập trình viên triển khai ứng dụng web một cách nhanh chóng mà vẫn đảm bảo các yếu tố cốt lõi của một sản phẩm chuyên nghiệp như tốc độ, bảo mật, khả năng mở rộng và tính ổn định.

Trong hệ sinh thái PHP, Laravel nổi bật bởi khả năng tổ chức mã nguồn khoa học theo mô hình MVC (Model – View – Controller). Mô hình này giúp phân chia cụ thể logic xử lý, giao diện và dữ liệu, từ đó giảm thiểu sự phụ thuộc và tăng khả năng mở rộng trong tương lai. Laravel cũng xây dựng một hệ thống routing rõ ràng, cho phép lập trình viên định nghĩa các tuyến đường và xử lý request một cách trực quan, dễ hiểu.

Một điểm đặc biệt quan trọng của Laravel là nó không chỉ đóng vai trò là framework chính mà còn là trung tâm của một hệ sinh thái phong phú gồm nhiều công cụ hỗ trợ đắc lực cho quá trình phát triển và vận hành ứng dụng web. Các thành phần đáng chú ý trong hệ sinh thái Laravel bao gồm:

- Laravel Homestead: môi trường phát triển ảo hóa sử dụng Vagrant, giúp đồng bộ môi trường làm việc giữa các lập trình viên, hạn chế lỗi phát sinh do khác biệt hệ điều hành hoặc cấu hình máy.
- Laravel Forge: nền tảng hỗ trợ triển khai và quản lý server cho các ứng dụng web dựa trên PHP, giúp tự động hóa quá trình cài đặt máy chủ.
- Laravel Vapor: nền tảng triển khai serverless trên AWS, cho phép ứng dụng Laravel chạy trên kiến trúc không máy chủ, tối ưu hóa hiệu năng và chi phí vận hành.
- Laravel Nova: hệ thống admin panel cao cấp giúp phát triển bảng điều khiển quản trị một cách nhanh chóng, tùy biến theo nhu cầu.
- Laravel Sanctum và Laravel Passport: cung cấp giải pháp xác thực API mạnh mẽ, phù hợp cho các ứng dụng RESTful hoặc SPA.

- Laravel Horizon: công cụ trực quan hóa và quản lý các hàng đợi xử lý trong ứng dụng, hỗ trợ theo dõi job real-time.

Trong số các tính năng của Laravel, Eloquent ORM được xem là điểm nhấn mạnh mẽ nhất. Đây là một công cụ giúp trừu tượng hóa cơ sở dữ liệu dưới dạng đối tượng, cho phép lập trình viên thao tác dữ liệu giống như làm việc với các đối tượng trong lập trình hướng đối tượng thay vì phải viết các câu lệnh SQL dài dòng và dễ gây lỗi. Eloquent ORM hỗ trợ đầy đủ các thao tác CRUD (Create – Read – Update – Delete), truy vấn nâng cao, quan hệ giữa các bảng như one-to-one, one-to-many, many-to-many,...

Nhờ có Eloquent, việc truy vấn dữ liệu trở nên dễ dàng và tối ưu hơn. Ví dụ:

- Truy vấn sản phẩm có giá > 1 triệu:
`$products = Product::where('don_gia', '>', 1000000)->get();`
- Thêm mới sản phẩm:
`$product = new Product;`
`$product->ten_san_pham = 'Máy hút bụi';`
`$product->don_gia = 1500000;`
`$product->save();`
- Cập nhật dữ liệu sản phẩm:
`$product = Product::find(1);`
`$product->don_gia = 1600000;`
`$product->save();`
- Xóa sản phẩm:
`Product::find(1)->delete();`

Những đoạn mã này không chỉ đơn giản mà còn giúp hạn chế lỗi SQL injection, nâng cao khả năng an toàn của quá trình truy vấn cơ sở dữ liệu. Chính vì vậy, đối với hệ thống thương mại điện tử nói chung và website bán giày trong đề tài này nói riêng, Eloquent ORM đóng vai trò vô cùng quan trọng, giúp quá trình phát triển trở nên dễ dàng, khoa học và nhanh chóng hơn.

Ưu điểm của Laravel:

- Cộng đồng đông đảo, tài liệu phong phú giúp người học dễ dàng tiếp cận và giải quyết vấn đề.
- Cấu trúc framework tách bạch rõ ràng, dễ mở rộng và duy trì.
- Hỗ trợ bảo mật mạnh mẽ, chống lại SQL injection, XSS, CSRF.
- Blade template giúp xây dựng giao diện linh hoạt, dễ đọc.
- Artisan CLI giúp tự động hóa các tác vụ nhanh chóng.
- Dễ dàng phát triển API cũng như tích hợp với các dịch vụ ngoài.

- Hệ sinh thái đầy đủ giúp triển khai, quản lý ứng dụng hiệu quả.
Nhược điểm của Laravel
- Khá nặng và chiếm nhiều bộ nhớ hơn các framework PHP đơn giản khác.
- Tốc độ thực thi không phải là cao nhất khi so sánh với các ngôn ngữ hoặc framework khác.
- Yêu cầu máy chủ có cấu hình tốt khi ứng dụng mở rộng.
- Cần thời gian làm quen với nhiều khái niệm như ORM, middleware, service container.
- Việc cập nhật phiên bản major có thể gây gián đoạn nếu dự án lớn.

1.1.2. Ngôn ngữ PHP

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình kịch bản phía máy chủ (server-side scripting language) được sử dụng phổ biến trong phát triển web. Đây là một ngôn ngữ mã nguồn mở, miễn phí, linh hoạt và dễ học, phù hợp cho cả người mới bắt đầu lẫn các lập trình viên chuyên nghiệp. PHP được thiết kế đặc biệt để nhúng trực tiếp vào mã HTML, giúp tạo ra các trang web động (dynamic website), nơi nội dung có thể thay đổi tùy theo dữ liệu và tương tác của người dùng.

Trải qua hơn ba thập kỷ phát triển, PHP vẫn giữ vai trò quan trọng trong công nghệ web, là nền tảng vận hành của hàng triệu website, hệ thống quản trị nội dung như WordPress, Joomla, Drupal, và nhiều nền tảng thương mại điện tử lớn. Sự phổ biến của PHP không chỉ nằm ở tính đơn giản mà còn ở khả năng mở rộng, tốc độ xử lý tốt, cộng đồng rộng lớn và kho thư viện phong phú.

Sức mạnh của PHP thể hiện rõ nhất khi kết hợp cùng các framework hiện đại, trong đó Laravel là framework được ưa chuộng nhất hiện nay nhờ vào khả năng hỗ trợ tối ưu hóa, bảo mật và tổ chức mã nguồn khoa học. Điều này giúp PHP tiếp tục giữ vai trò quan trọng trong lập trình web dù nhiều ngôn ngữ mới xuất hiện.

1.1.2.1. Lịch sử và sự phát triển của PHP

PHP được ra đời vào năm 1993 khi Rasmus Lerdorf viết một tập hợp các script mang tên Personal Home Page Tools để theo dõi lượng truy cập vào trang cá nhân của ông. Mặc dù mục đích ban đầu chỉ là một tiện ích nhỏ, nhưng sức ảnh hưởng và nhu cầu phát triển web động vào thập niên 90 đã nhanh chóng biến PHP trở thành một ngôn ngữ độc lập.

Những cột mốc quan trọng của PHP:

- PHP 2.x (1995): chính thức trở thành một công cụ lập trình web đơn giản, hỗ trợ form và database cơ bản.

- PHP 3.x (1997): đánh dấu bước đột phá lớn, được tái thiết kế hoàn chỉnh, hỗ trợ nhiều module mở rộng, trở thành phiên bản được cộng đồng đón nhận rộng rãi.

- PHP 4.x (2000): sử dụng Zend Engine 1.0, cải thiện tốc độ xử lý và tính ổn định.

- PHP 5.x (2004): hỗ trợ lập trình hướng đối tượng OOP đầy đủ hơn, mở đường cho các framework hiện đại.

- PHP 7.x (2015): cải tiến mạnh về hiệu năng (nhanh gấp 2–3 lần PHP 5), giảm tải bộ nhớ, hỗ trợ kiểu dữ liệu tốt hơn.

- PHP 8.x (2020): bổ sung JIT compiler, tăng tốc độ thực thi vượt trội và hỗ trợ cú pháp hiện đại, giúp PHP cạnh tranh với các ngôn ngữ mới.

Nhờ quá trình phát triển liên tục, PHP đã từ một tập hợp script đơn giản trở thành một trong những ngôn ngữ lập trình chủ lực của web hiện đại.

1.1.2.2. Các đặc điểm nổi bật của PHP

PHP mang nhiều đặc điểm nổi bật khiến ngôn ngữ này trở thành lựa chọn hàng đầu trong phát triển web:

- Mã nguồn mở, miễn phí và dễ tiếp cận: PHP hoàn toàn miễn phí và có mã nguồn mở, giúp lập trình viên dễ dàng cài đặt, sử dụng và tùy chỉnh. Cộng đồng PHP đông đảo đóng góp hàng nghìn thư viện, package và framework, giúp mở rộng tính năng mà không cần lập trình lại từ đầu.

- Cú pháp đơn giản, linh hoạt: PHP có cú pháp dễ học, gần gũi với người mới bắt đầu. Người học có thể viết mã PHP ngay trong HTML bằng cách sử dụng thẻ `<?php ... ?>`, điều này giúp phát triển web nhanh và tự nhiên.

- Tích hợp mạnh mẽ với HTML, CSS, JavaScript: Không giống như nhiều ngôn ngữ khác yêu cầu biên dịch hoặc cấu hình phức tạp, PHP có thể được chèn trực tiếp vào file HTML, tạo ra trang web động một cách mượt mà.

- Hỗ trợ nhiều cơ sở dữ liệu: PHP có thể kết nối với hầu hết các hệ quản trị cơ sở dữ liệu phổ biến như: MySQL, PostgreSQL, SQLite, MariaDB, Oracle, SQL Server. MySQL là lựa chọn phổ biến nhất nhờ độ ổn định và dễ tích hợp, đặc biệt trong các hệ thống thương mại điện tử sử dụng Laravel.

- Hỗ trợ lập trình hướng đối tượng (OOP): Từ PHP 5 trở đi, PHP hỗ trợ OOP mạnh mẽ với các đặc tính như: Class, object, Kế thừa (inheritance), Đa hình (polymorphism), Tính bao đóng (encapsulation), Interface, abstract class. Điều này cho phép sử dụng các mô hình kiến trúc hiện đại như MVC, giúp mã nguồn sạch và dễ bảo trì.

- Khả năng mở rộng mạnh với hệ sinh thái phong phú, Các framework mạnh mẽ như: Laravel, Symfony, CodeIgniter, Yii... được xây dựng trên PHP, cho thấy sự linh hoạt và khả năng mở rộng lớn của ngôn ngữ này.

- Cộng đồng lớn và tài liệu phong phú: PHP có một cộng đồng đông đảo gồm hàng triệu lập trình viên trên toàn thế giới, hỗ trợ qua: StackOverflow, Forum PHP quốc tế

- Tài liệu chính thức PHP Manual: Hàng nghìn khóa học miễn phí và trả phí. Điều này giúp việc học và phát triển với PHP trở nên dễ dàng hơn rất nhiều.

1.1.2.3. Ưu điểm của PHP

PHP mang đến nhiều lợi thế thực tiễn, đặc biệt trong môi trường phát triển web:

- Chi phí thấp: Vì PHP là mã nguồn mở, không cần trả phí bản quyền nên hầu hết doanh nghiệp và cá nhân đều có thể sử dụng. Máy chủ PHP cũng rẻ hơn so với các nền tảng khác.

- Tương thích với nhiều hệ điều hành: PHP có thể chạy trên: Windows, Linux, macOS, Unix. Điều này giúp dễ triển khai trên nhiều loại server khác nhau.

- Tốc độ xử lý tốt, PHP được tối ưu cho xử lý HTTP và truy vấn cơ sở dữ liệu, đặc biệt khi sử dụng PHP 7 trở lên, tốc độ cải thiện đáng kể.

- Phù hợp cho cả website nhỏ và hệ thống lớn, những website nổi tiếng như: Facebook (giờ dùng HHVM nhưng vẫn dựa trên PHP), WordPress (CMS lớn nhất thế giới), Wikipedia... đều sử dụng PHP – chứng minh khả năng mở rộng của ngôn ngữ này.

1.1.2.4. Nhược điểm của PHP

- Cấu trúc và thư viện không hoàn toàn thống nhất: Do lịch sử phát triển lâu dài, một số hàm trong PHP có tên gọi không thống nhất, khiến người học khó nhớ.

- Hỗ trợ OOP chưa mạnh như Java hoặc C#: PHP có OOP nhưng không hoàn toàn thuần như các ngôn ngữ hướng đối tượng khác.

- Có nguy cơ bảo mật nếu lập trình không đúng cách, các lỗi bảo mật phổ biến: SQL Injection, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery). Tuy nhiên, nếu kết hợp PHP với Laravel, các vấn đề này gần như được giải quyết thông qua: CSRF token, Query Builder và Eloquent ORM, Input validation, Xử lý dữ liệu đầu vào tự động

1.1.3. Công cụ cài đặt và môi trường phát triển Laravel

Để xây dựng và phát triển ứng dụng web bằng Laravel một cách hiệu quả, việc chuẩn bị đúng môi trường phát triển là điều vô cùng quan trọng. Laravel hoạt động dựa trên ngôn ngữ PHP, sử dụng hệ quản trị cơ sở dữ liệu như MySQL và

quản lý các thư viện thông qua Composer. Do đó, lập trình viên cần thiết lập đầy đủ các công cụ này trước khi bắt đầu lập trình.

Quá trình cài đặt Laravel tuy không phức tạp nhưng yêu cầu thực hiện theo đúng trình tự nhằm đảm bảo dự án vận hành ổn định, tránh lỗi trong quá trình phát triển:

- Cài đặt Composer – Trình quản lý thư viện PHP: Composer là công cụ không thể thiếu trong mọi dự án Laravel. Nó giúp tự động tải về các thư viện liên quan, đồng thời quản lý phiên bản và phụ thuộc của từng package trong dự án.

Trang tải Composer: <https://getcomposer.org/>

- Kiểm tra sau khi cài đặt: `composer -v`. Composer đóng vai trò tương tự như npm trong Node.js hoặc pip trong Python, giúp phát triển dự án PHP nhanh gọn và chuẩn hóa hơn.

- Tạo mới dự án Laravel qua Composer: Sau khi cài đặt Composer thành công, lập trình viên có thể khởi tạo một dự án Laravel bằng lệnh: `composer create-project laravel/laravel ten-du-an`. Ví dụ với đề tài này: `composer create-project laravel/laravel THEDUNGSHOE`. Lệnh này giúp tự động tải phiên bản Laravel mới nhất và thiết lập đầy đủ cấu trúc thư mục của framework.

- Cấu hình tệp môi trường (.env): Laravel sử dụng file .env để quản lý các thông số quan trọng của hệ thống như: thông tin cơ sở dữ liệu, chế độ chạy ứng dụng, khóa bảo mật, các cấu hình mail, queue, broadcasting...

- Ngay sau khi cài đặt project, thư mục sẽ chứa file mẫu .env.example. Lập trình viên cần sao chép và đổi tên thành .env, sau đó cấu hình:

```
DB_CONNECTION=mysql
```

```
DB_HOST=127.0.0.1
```

```
DB_PORT=3306
```

```
DB_DATABASE=thedungshoe
```

```
DB_USERNAME=root
```

```
DB_PASSWORD=
```

Sau đó tạo khóa ứng dụng:

```
php artisan key:generate
```

- Khởi động server phát triển: Laravel tích hợp sẵn web server phục vụ cho việc lập trình. Để chạy ứng dụng: `php artisan serve`. Trang web sẽ chạy tại địa chỉ: `http://127.0.0.1:8000`

Điều này giúp lập trình viên kiểm thử chức năng mà không cần cấu hình server phức tạp.

Mô tả cấu trúc thư mục của dự án Laravel: Laravel được thiết kế theo mô hình MVC với hệ thống thư mục rõ ràng, phân tách chức năng cụ thể:

- app/ – chứa logic ứng dụng
- Http/ (controller, middleware)
- Models/ (mô hình dữ liệu)
- bootstrap/ – khởi tạo framework
- config/ – chứa các tệp cấu hình
- database/ – migration, seeder
- public/ – tài nguyên người dùng truy cập (ảnh, CSS, JS)
- resources/ – views, template Blade
- routes/ – file định tuyến web.php và api.php
- storage/ – cache, file tạm, log
- vendor/ – thư viện bên thứ ba do Composer cài đặt

Nhờ cấu trúc thư mục chặt chẽ này, Laravel giúp việc quản lý mã nguồn trở nên thống nhất, dễ bảo trì và hỗ trợ nhiều lập trình viên cùng làm việc trên một dự án.

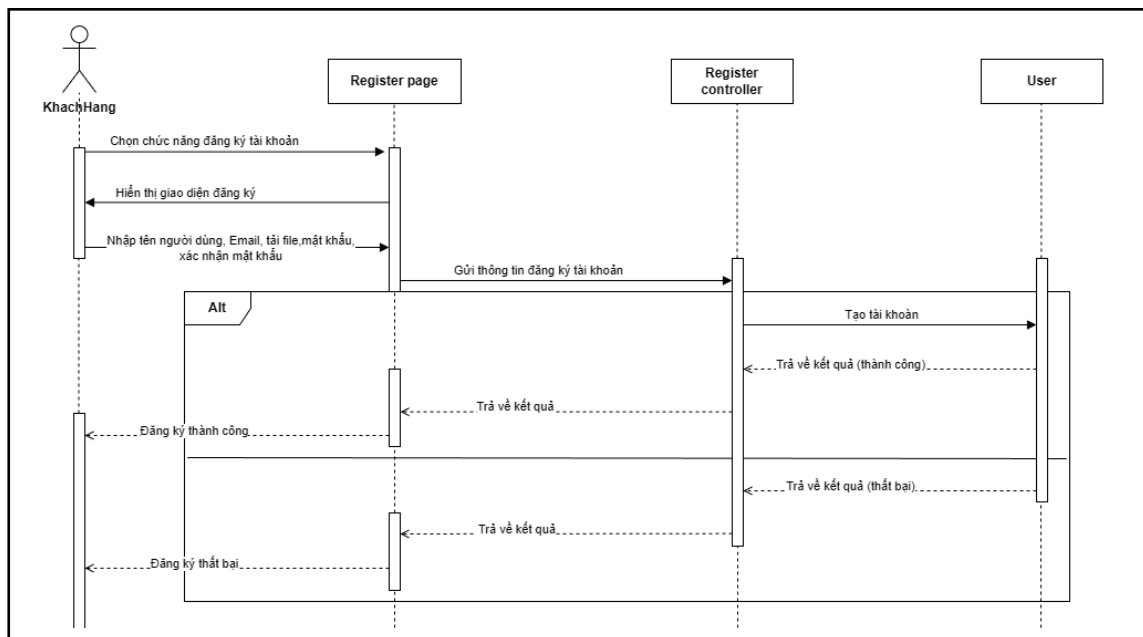
1.2. Mô hình hoạt động và thuật toán của hệ thống

1.2.1. Sơ đồ tuần tự các chức năng

Trong một hệ thống thương mại điện tử, việc mô tả sự tương tác giữa người dùng và hệ thống thông qua các sơ đồ tuần tự (Sequence Diagram) giúp làm rõ cách dữ liệu di chuyển và cách các thành phần phản hồi lại yêu cầu của người dùng. Các sơ đồ tuần tự đóng vai trò quan trọng trong giai đoạn phân tích và thiết kế hệ thống vì chúng mô tả trực quan toàn bộ luồng nghiệp vụ.

1.2.1.1. Sơ đồ tuần tự chức năng “Đăng ký tài khoản”

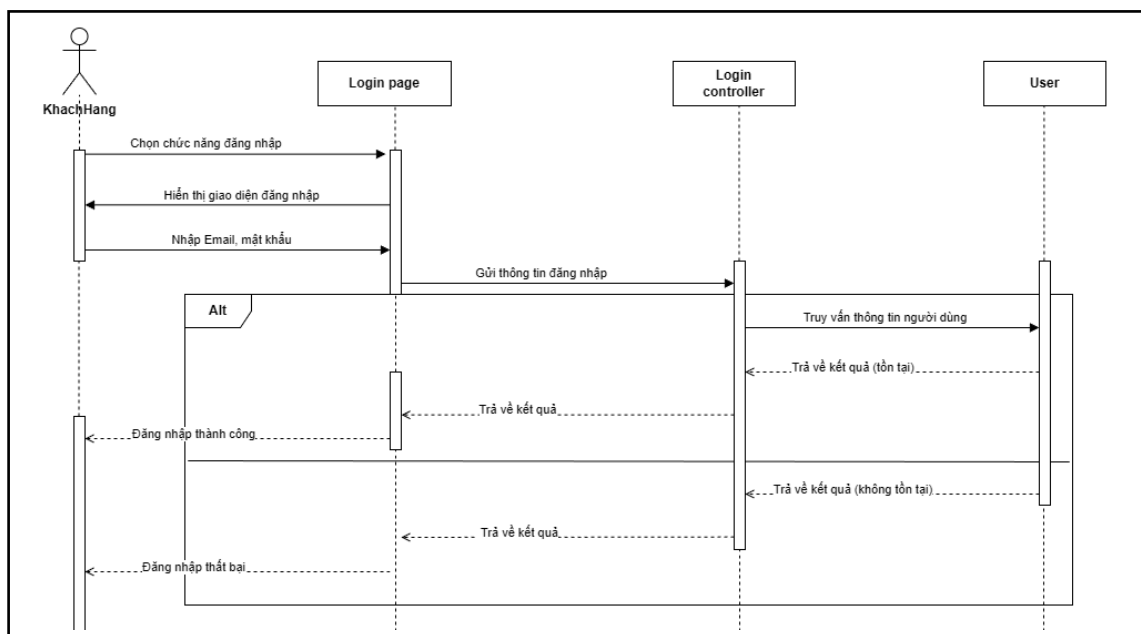
Mình họa quá trình tạo tài khoản mới, bao gồm: người dùng nhập thông tin → hệ thống kiểm tra dữ liệu hợp lệ → mã hóa mật khẩu → lưu vào cơ sở dữ liệu → phản hồi thành công.



Hình 1.1. Sơ đồ tuần tự chức năng “Đăng ký”

1.2.1.2. Sơ đồ tuần tự chức năng “Đăng nhập tài khoản”

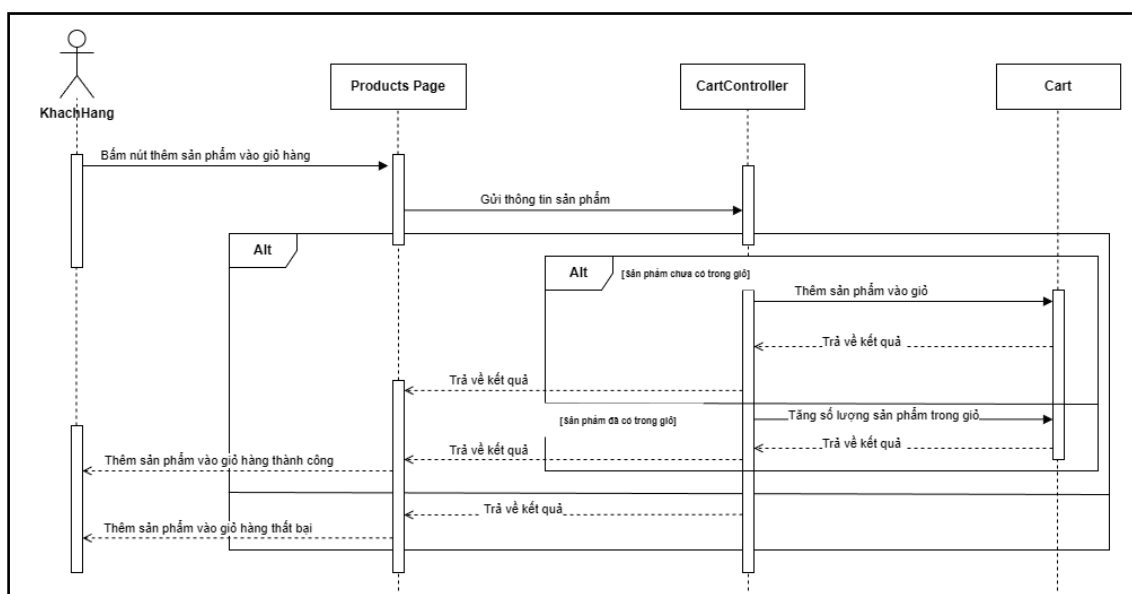
Mô tả quá trình người dùng nhập thông tin tài khoản → gửi yêu cầu đăng nhập → hệ thống kiểm tra thông tin → trả kết quả thành công hoặc thất bại.



Hình 1.2. Sơ đồ tuần tự chức năng “Đăng nhập”

1.2.1.3. Sơ đồ tuần tự chức năng “Thêm sản phẩm vào giỏ hàng”

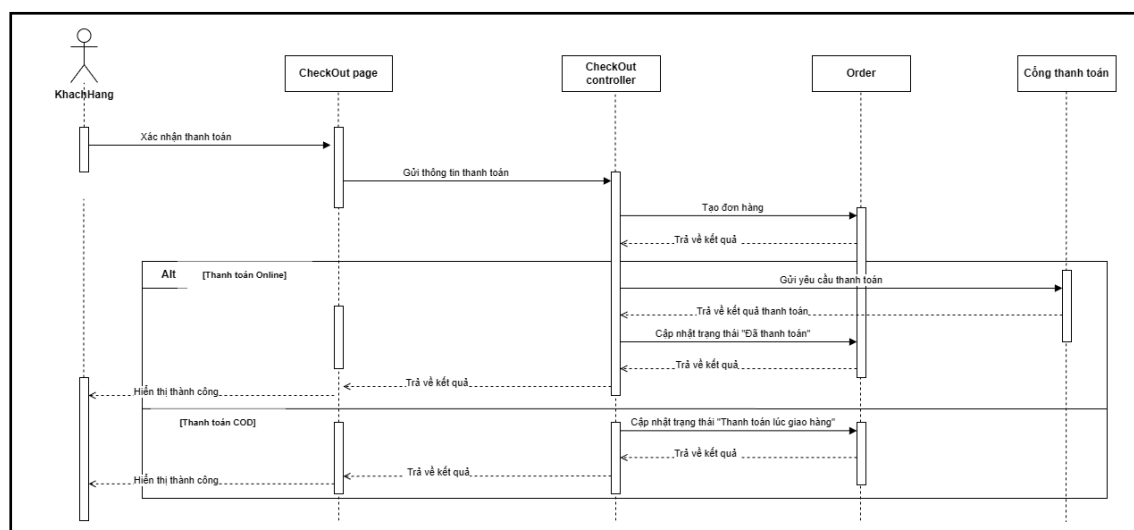
Thể hiện sự tương tác giữa người dùng, trang sản phẩm và cơ chế lưu trữ giỏ hàng (session hoặc database).



Hình 1.3. Sơ đồ tuần tự chức năng “Đăng ký”

1.2.1.3. Sơ đồ tuần tự chức năng “Thanh toán”

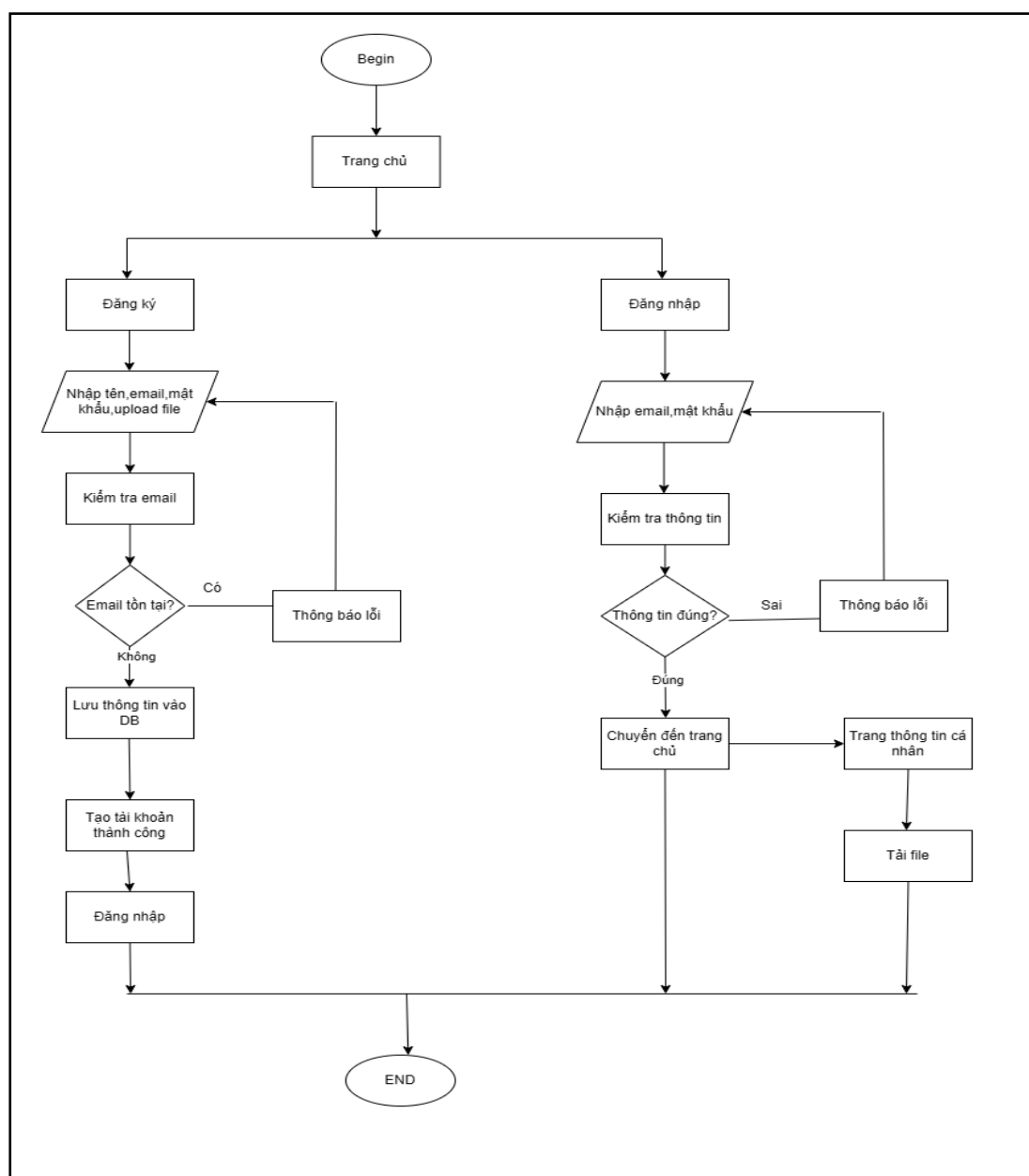
Diễn tả các bước từ khi người dùng xác nhận sản phẩm → nhập thông tin thanh toán → hệ thống xử lý đơn hàng → lưu thông tin → thông báo hoàn tất giao dịch.



Hình 1.4. Sơ đồ tuần tự chức năng “Thanh toán”

1.2.2. Thuật toán xử lý nghiệp vụ

Bên cạnh sơ đồ tuần tự, hệ thống còn được mô tả bằng các mô hình thuật toán nhằm làm rõ logic xử lý trong từng chức năng. Một trong những thuật toán quan trọng nhất là thuật toán “Đăng ký – Đăng nhập”, bởi nó đảm bảo vấn đề an toàn và xác thực người dùng.



Hình 1.5. Mô hình thuật toán “Đăng ký – Đăng nhập”

1.3. Thiết kế cơ sở dữ liệu của hệ thống

1.3.1. Giới thiệu

Cơ sở dữ liệu là một phần quan trọng trong hệ thống, giúp lưu trữ và quản lý thông tin hiệu quả. Trong hệ thống này, em sử dụng MySQL, một hệ quản trị cơ sở dữ liệu quan hệ phổ biến. MySQL hỗ trợ các truy vấn SQL mạnh mẽ, giúp việc quản lý dữ liệu trở nên dễ dàng và hiệu quả. Cơ sở dữ liệu sẽ lưu trữ thông tin người dùng, tài khoản, mật khẩu và các thông tin liên quan khác để phục vụ cho chức năng đăng nhập và quản lý hệ thống.

1.3.2. Thiết kế cơ sở dữ liệu

Trong đề tài "Website bán giày" được xây dựng trên nền tảng framework Laravel, cơ sở dữ liệu của hệ thống bao gồm các bảng chính sau đây, mỗi bảng đảm nhận vai trò quan trọng trong việc lưu trữ và quản lý thông tin liên quan đến các hoạt động kinh doanh trên website:

Cột	Kiểu	Null	Mặc định	Liên kết tới
id	bigint(20)	Không		
name	varchar(255)	Không		
email	varchar(255)	Không		
phone	varchar(200)	Có	NULL	
email_verified_at	timestamp	Có	NULL	
password	varchar(255)	Không		
role	enum('user', 'admin')	Không	user	
remember_token	varchar(100)	Có	NULL	
created_at	timestamp	Có	NULL	
updated_at	timestamp	Có	NULL	

Bảng 1.1: Database bảng User

Cột	Kiểu	Null	Mặc định	Liên kết tới
id	bigint(20)	Không		
user_id	bigint(20)	Không		users (id)
file_name	varchar(255)	Không		
path	varchar(255)	Không		
created_at	timestamp	Có	NULL	
updated_at	timestamp	Có	NULL	

Bảng 1.2: Database bảng Upload_files

Cột	Kiểu	Null	Mặc định	Liên kết tới
id	bigint(20)	Không		
name	varchar(255)	Không		
description	text	Có	NULL	
price	decimal(10,2)	Không		
image	varchar(255)	Có	NULL	
brand	varchar(200)	Không		
created_at	timestamp	Có	NULL	
updated_at	timestamp	Có	NULL	

Bảng 1.3: Database bảng Products

Cột	Kiểu	Null	Mặc định	Liên kết tới
id	bigint(20)	Không		
name	varchar(255)	Không		
email	varchar(255)	Không		
message	text	Không		
is_read	tinyint(1)	Không	0	
created_at	timestamp	Có	NULL	
updated_at	timestamp	Có	NULL	

Bảng 1.4: Database bảng Contacts

Cột	Kiểu	Null	Mặc định	Liên kết tới
id	bigint(20)	Không		
user_id	bigint(20)	Có	NULL	
name	varchar(255)	Không		
email	varchar(255)	Có	NULL	
phone	varchar(255)	Không		
city	varchar(255)	Có	NULL	
address	text	Không		
delivery method	varchar(255)	Không		
payment method	varchar(255)	Không		
total price	int(11)	Không		
status	varchar(255)	Không	pending	
created_at	timestamp	Có	NULL	
updated_at	timestamp	Có	NULL	
is_read	tinyint(1)	Không	0	

Bảng 1.5: Database bảng Orders

Cột	Kiểu	Null	Mặc định	Liên kết tới
id	bigint(20)	Không		
order_id	bigint(20)	Không		orders (id)
product_id	bigint(20)	Không		
name	varchar(255)	Không		
qty	int(11)	Không		
price	int(11)	Không		
image	varchar(255)	Có	NULL	
created_at	timestamp	Có	NULL	
updated_at	timestamp	Có	NULL	

Bảng 1.6: Database bảng OrderItems

1.3.3. Dữ liệu trong các bảng

- Dữ liệu bảng User

✓ Hiện thị các hàng 0 - 6 (tổng 7, Truy vấn cần 0,0023 giây.)

SELECT * FROM `users`

☐ Hồ sơ ☐ Sửa chung dòng ☐ Sửa ☐ Giải thích SQL ☐ Tạo mã PHP ☐ Làm tươi lại

☐ Hiện tất | Số hàng: 25 | Số hàng: Tìm kiếm trong bảng này | Sort by key: Không

Extra options

	id	name	email	adress	phone	role_id	email_verified_at	password	role	remember
<input type="checkbox"/> Sửa <input type="checkbox"/> Chép <input type="checkbox"/> Xóa bỏ	7	Thế Dũng	luongthedung2003@gmail.com	NULL	NULL	0	NULL	\$2y\$10\$jl73X05zjcyh0sICt1O0iczo8xv9zYRb3B.kaG5M...	admin	NULL
<input type="checkbox"/> Sửa <input type="checkbox"/> Chép <input type="checkbox"/> Xóa bỏ	8	Thế Dũng	luongthedungx@gmail.com	NULL	NULL	0	NULL	\$2y\$10\$gIRtgc4unpV3C0u6a983.pNIHwNQwzYjxpDEVU5HVK...	admin	5n30PLFT
<input type="checkbox"/> Sửa <input type="checkbox"/> Chép <input type="checkbox"/> Xóa bỏ	12	Thế Dũng	luongthe@gmail.com	NULL	NULL	0	NULL	\$2y\$10\$iczSO7yIEE5AC9ALeT/Gc./QsAzd5LxFJyfunuw.Shp...	admin	NULL
<input type="checkbox"/> Sửa <input type="checkbox"/> Chép <input type="checkbox"/> Xóa bỏ	16	Lê Chí Duy	lechiduy@gmail.com	NULL	NULL	0	NULL	\$2y\$10\$NPIUduJpcKERT9OtpSCTCeeu/nGAFBrZuZ7QLJgJWYL...	admin	NULL
<input type="checkbox"/> Sửa <input type="checkbox"/> Chép <input type="checkbox"/> Xóa bỏ	17	Thế Dũng	luongthee@gmail.com	NULL	NULL	0	NULL	\$2y\$10\$6MjetlONqcByNJ1Xz7RYRu17IZf8G4INM/i5mfdUj/x...	user	NULL
<input type="checkbox"/> Sửa <input type="checkbox"/> Chép <input type="checkbox"/> Xóa bỏ	18	Xuân Mai	xuanmai@gmail.com	NULL	NULL	0	NULL	\$2y\$10\$fmeGKvhSMT7Qceksa16i4ObB9opZyWy1oIOJzsgV.rN...	user	NULL

Hình 1.6. Dữ liệu trong bảng User

-Dữ liệu trong bảng Products

✓ Hiện thị các hàng 0 - 9 (tổng 10, Truy vấn cần 0,0018 giây.)

SELECT * FROM `products`

☐ Hồ sơ [\[Sửa chung dòng \]](#) [\[Sửa \]](#) [\[Giải thích SQL \]](#) [\[Tạo mã PHP \]](#) [\[Làm tươi lại \]](#)

☐ Hiện tất | Số hàng: 25 | Số hàng: Tìm kiếm trong bảng này | Sort by key: Không

Extra options

				id	name	description	price	image	brand	created_at	updated_at
<input type="checkbox"/>				1	Nike Air Force 1	Dòng giày huyền thoại, thiết kế đơn giản, cổ điển	200000.00	/products/png-1.jpg	NIKE	NULL	NULL
<input type="checkbox"/>				2	Air Jordan 1	Huyền thoại bóng rổ, nhiều phối màu	300000.00	/products/png-2.jpg	NIKE	NULL	NULL
<input type="checkbox"/>				3	Air Max 270	Đệm Air lớn, êm ái	1000000.00	/products/png-3.jpg	NIKE	NULL	NULL
<input type="checkbox"/>				4	Dunk Low	Streetwear được ưa chuộng	2500000.00	/products/png-4.jpg	NIKE	NULL	NULL
<input type="checkbox"/>				5	Blazer Mid 77	Phong cách retro vintage	2000000.00	/products/png-5.jpg	NIKE	NULL	NULL
<input type="checkbox"/>				6	Nike React Infinity Run	Chạy bộ êm, hỗ trợ tốt	500000.00	/products/png-6.jpg	NIKE	NULL	NULL
<input type="checkbox"/>				7	Superstar	Mũi sò đặc trưng, streetwear	300000.00	/products/png-7.jpg	ADIDAS	NULL	NULL
<input type="checkbox"/>				8	Ultraboost 22	Đệm Boost êm ái, chạy bộ	350000.00	/products/png-8.jpg	ADIDAS	NULL	NULL
<input type="checkbox"/>				9	Adidas Samba	Thời trang retro, trending TikTok	1500000.00	/products/png-9.jpg	ADIDAS	NULL	NULL
<input type="checkbox"/>				10	NMD R1	Thiết kế hiện đại, thoải mái	450000.00	/products/png-10.jpg	ADIDAS	NULL	NULL

↑ ☐ Theo dõi bảng | Lưu mục đã chọn | Sửa | Chép | Xóa bỏ | Xuất

Hình 1.7. Dữ liệu trong bảng Products

- Dữ liệu trong bảng Upload_files

✓ Hiện thị các hàng 0 - 2 (tổng 3, Truy vấn cần 0,0014 giây.)

SELECT * FROM `upload_files`

☐ Hồ sơ [\[Sửa chung dòng \]](#) [\[Sửa \]](#) [\[Giải thích SQL \]](#) [\[Tạo mã PHP \]](#) [\[Làm tươi lại \]](#)

☐ Hiện tất | Số hàng: 25 | Số hàng: Tìm kiếm trong bảng này | Sort by key: Không

Extra options

				id	user_id	file_name	path	created_at	updated_at
<input type="checkbox"/>				11	17	Mô tả chức năng hệ thống.docx	profile_files/CAZss52s3fIEBFPeR0b2qaKtLpN0XAmWR4J...	2025-10-06 13:00:09	2025-10-06 13:00:09
<input type="checkbox"/>				12	18	NPM.pdf	profile_files/iZAVP177hTNzr0yzJfR0CCXiLDZ6YAEfE1S6...	2025-10-06 13:59:26	2025-10-06 13:59:26
<input type="checkbox"/>				14	20	01 BIÊN BẢN KIỂM PHIẾU.docx	profile_files/5Yxaa7Vaz9AjseXea3Y8isOvoJbkwd0JsQF...	2025-10-08 16:09:39	2025-10-08 16:09:39

↑ ☐ Theo dõi bảng | Lưu mục đã chọn | Sửa | Chép | Xóa bỏ | Xuất

☐ Hiện tất | Số hàng: 25 | Số hàng: Tìm kiếm trong bảng này | Sort by key: Không

Hình 1.8. Dữ liệu trong bảng Upload_files

-Dữ liệu trong bảng Contacts

✔ Hiển thị các hàng 0 - 8 (tổng 9, Truy vấn cần 0,0012 giây.)

SELECT * FROM `contacts`

☐ Hồ sơ [\[Sửa chung dòng \]](#) [\[Sửa \]](#) [\[Giải thích SQL \]](#) [\[Tạo mã PHP \]](#) [\[Làm tươi lại \]](#)

☐ Hiện tất | Số hàng: 25 | Số hàng: Sort by key:

Extra options

				id	name	email	message	is_read	created_at	updated_at
<input type="checkbox"/>				1	Lương	luongthe@gmail.com	trải nghiệm tệ	0	2025-10-05 15:43:48	2025-10-05 15:43:48
<input type="checkbox"/>				2	Thế Dũng	luongthedung2003@gmail.com	Sản phẩm chất lượng,sẽ mua lần 2	0	2025-10-05 16:12:51	2025-10-05 16:12:51
<input type="checkbox"/>				3	Phạm thị Dung	dung_2351220152@dau.edu.vn	tệ	0	2025-10-05 16:34:33	2025-10-05 16:34:33
<input type="checkbox"/>				4	Lê chỉ duy	chiduy@gmail.com	Lê chỉ duy nè	0	2025-10-06 05:59:07	2025-10-06 05:59:07
<input type="checkbox"/>				5	Lương Thế Dũng	luongthedung@gmail.com	aaaaaaaaaaaaaaaaaaaaa	0	2025-10-18 17:25:30	2025-10-18 17:25:30
<input type="checkbox"/>				6	Mai	x@gmail.com	Tốt nha	0	2025-10-18 17:32:40	2025-10-18 17:32:40
<input type="checkbox"/>				7	Mai	x@gmail.com	zzzzzzzzzzzz	0	2025-10-18 17:34:19	2025-10-18 17:34:19
<input type="checkbox"/>				8	Thế Dũng	e@gmail.com	ngoannnnn	0	2025-10-18 17:37:31	2025-10-18 17:37:31
<input type="checkbox"/>				9	Thế Dũng	e@gmail.com	ssssssss	0	2025-10-25 15:04:06	2025-10-25 15:04:06

☐ Theo dõi bảng [Lưu mục đã chọn](#) Sửa Chép Xóa bỏ Xuất

Hình 1.9. Dữ liệu trong bảng Contacts

-Dữ liệu trong bảng Orders

✔ Hiển thị các hàng 0 - 6 (tổng 7, Truy vấn cần 0,0016 giây.)

SELECT * FROM `orders`

☐ Hồ sơ [\[Sửa chung dòng \]](#) [\[Sửa \]](#) [\[Giải thích SQL \]](#) [\[Tạo mã PHP \]](#) [\[Làm tươi lại \]](#)

☐ Hiện tất | Số hàng: 25 | Số hàng: Sort by key:

Extra options

				id	user_id	name	email	phone	city	address	delivery_method	payment_method	shipping_fee	total_price
<input type="checkbox"/>				1	32	Thế Dũng	luongthedung@gmail.com	0829096815	Đà Nẵng	Đà Nẵng	delivery	COD	30000	53000
<input type="checkbox"/>				2	32	Thế Dũng	luongthedung@gmail.com	0829096815	Đà Nẵng	s	delivery	COD	30000	33000
<input type="checkbox"/>				3	32	Thế Dũng	luongthedung@gmail.com	0829096815	Đà Nẵng	Đà Nẵng	delivery	COD	30000	23000
<input type="checkbox"/>				4	32	Thế Dũng	luongthedung@gmail.com	0829096815	Đà Nẵng	e	delivery	COD	30000	103000
<input type="checkbox"/>				5	32	Thế Dũng	luongthedung@gmail.com	0829096815	Đà Nẵng	j	delivery	COD	30000	23000

☐ Theo dõi bảng [Lưu mục đã chọn](#) Sửa Chép Xóa bỏ Xuất

Hình 1.10. Dữ liệu trong bảng Orders

-Dữ liệu trong bảng OrderItems

✓ Hiển thị các hàng 0 - 5 (tổng 6, Truy vấn cần 0,0018 giây.)

SELECT * FROM `order_items`

☐ Hồ sơ ☐ Sửa chung dòng ☐ Sửa ☐ Giải thích SQL ☐ Tạo mã PHP ☐ Làm tươi lại

☐ Hiện tất | Số hàng: 25 | Số hàng: Sort by key:

Extra options

				id	order_id	product_id	name	qty	price	image	created_at	updated_at
<input type="checkbox"/>				1	1	1	Nike Air Force 1	1	200000	/products/png-1.jpg	2025-11-15 22:50:26	2025-11-15 22:50:26
<input type="checkbox"/>				2	1	2	Air Jordan 1	1	300000	/products/png-2.jpg	2025-11-15 22:50:26	2025-11-15 22:50:26
<input type="checkbox"/>				3	2	2	Air Jordan 1	1	300000	/products/png-2.jpg	2025-11-15 23:10:31	2025-11-15 23:10:31
<input type="checkbox"/>				4	3	1	Nike Air Force 1	1	200000	/products/png-1.jpg	2025-11-15 23:11:56	2025-11-15 23:11:56
<input type="checkbox"/>				5	4	3	Air Max 270	1	1000000	/products/png-3.jpg	2025-11-15 23:15:58	2025-11-15 23:15:58
<input type="checkbox"/>				6	5	1	Nike Air Force 1	1	200000	/products/png-1.jpg	2025-11-15 23:38:55	2025-11-15 23:38:55

☐ Theo dõi bảng ☐ Lưu mục đã chọn

Hình 1.11. Dữ liệu trong bảng OrderItems

1.4. Tiểu kết chương I

Chương I đã trình bày những kiến thức nền tảng và cơ sở kỹ thuật phục vụ cho việc xây dựng website bán giày bằng Framework Laravel. Nội dung chương tập trung vào ba phần chính: nền tảng công nghệ, mô hình hoạt động của hệ thống và thiết kế cơ sở dữ liệu.

Trước hết, chương đã phân tích các công nghệ được sử dụng, đặc biệt là Laravel – framework PHP mạnh mẽ hỗ trợ mô hình MVC, Eloquent ORM, Blade Template và các công cụ tiện ích giúp phát triển ứng dụng web nhanh, bảo mật và dễ mở rộng. Bên cạnh đó, chương cũng đề cập đến PHP, MySQL và các công cụ thiết lập môi trường như Composer, góp phần tạo nên hạ tầng kỹ thuật vững chắc cho hệ thống.

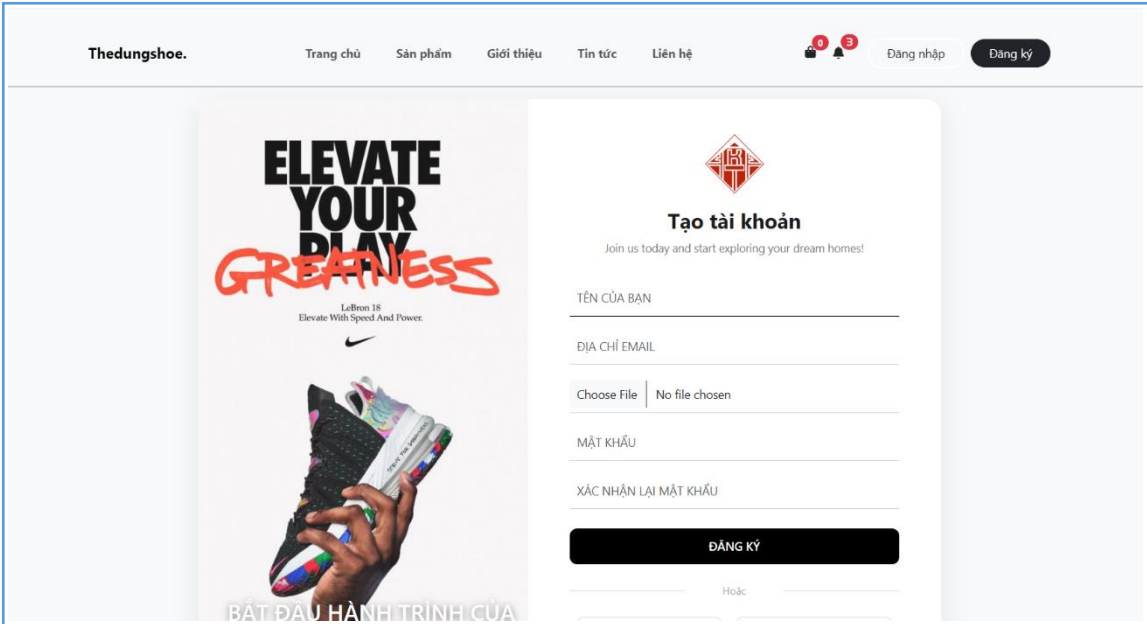
Tiếp theo, chương trình bày các sơ đồ tuần tự và thuật toán mô tả luồng hoạt động của các chức năng cốt lõi. Những mô hình này giúp làm rõ logic xử lý của hệ thống trong quá trình thiết kế và triển khai.

Cuối cùng, chương đã mô tả thiết kế cơ sở dữ liệu với các bảng chính như Users, Products, Upload_files và Contacts. Đây là nền tảng dữ liệu quan trọng để hệ thống hoạt động ổn định và hỗ trợ đầy đủ các chức năng thương mại điện tử trong các chương tiếp theo.

CHƯƠNG II: THỰC NGHIỆM

2.1. Chức năng Đăng ký

2.1.1. Giao diện



The screenshot shows the 'Đăng ký tài khoản' (Create Account) form on the Thedungshoe website. The form is titled 'Tạo tài khoản' (Create Account) with the tagline 'Join us today and start exploring your dream homes!'. It includes the following fields and elements:

- TÊN CỦA BẠN** (Your Name): A text input field.
- ĐỊA CHỈ EMAIL** (Email Address): A text input field.
- Choose File** / **No file chosen**: A file upload section.
- MẬT KHẨU** (Password): A text input field.
- XÁC NHẬN LẠI MẬT KHẨU** (Confirm Password): A text input field.
- ĐĂNG KÝ** (Register): A black button.
- Hoặc** (Or): A link to the login page.

On the left side of the form, there is a promotional image for the 'ELEVATE YOUR PLAY GREATNESS' campaign featuring a hand holding a colorful sneaker, with the text 'LeBron 18 Elevate With Speed And Power.' and the Nike logo.

Hình 2.1. Giao diện “Đăng ký tài khoản”

2.1.2. Giới thiệu chức năng đăng ký

Chức năng Đăng ký tài khoản (User Registration) trong Laravel đóng vai trò quan trọng trong việc hình thành hệ thống khách hàng cho website bán giày. Đây là bước đầu tiên giúp người dùng tạo lập danh tính số trong hệ thống, từ đó có thể truy cập và sử dụng các tính năng mở rộng như đặt hàng, theo dõi đơn hàng, lưu giỏ hàng, đánh giá sản phẩm hoặc nhận ưu đãi dành riêng cho thành viên.

Laravel cung cấp sẵn nhiều công cụ mạnh mẽ như Laravel Breeze, Laravel Jetstream, hoặc Laravel UI, giúp triển khai luồng đăng ký nhanh chóng, an toàn và có cấu trúc rõ ràng. Các gói này hỗ trợ sẵn xác thực, mã hóa mật khẩu, kiểm tra dữ liệu đầu vào và bảo vệ hệ thống trước những lỗ hổng bảo mật phổ biến. Nhờ vậy, quá trình đăng ký đảm bảo trải nghiệm mượt mà cho người dùng và đồng thời giảm thiểu rủi ro cho nhà phát triển.

2.1.3. Mục đích

Tạo tài khoản người dùng mới: Cho phép khách hàng tạo hồ sơ cá nhân để sử dụng các tính năng yêu cầu xác thực như đặt mua giày, thanh toán trực tuyến, theo dõi trạng thái giao hàng hoặc quản lý đơn mua.

Tăng mức độ tiện lợi và trải nghiệm người dùng: Khách hàng sau khi đăng ký có thể lưu thông tin giao hàng, quản lý giỏ hàng dài hạn, xem lại lịch sử mua sắm và nhận thông báo khuyến mãi nhanh hơn.

Quản lý dữ liệu khách hàng một cách khoa học: Hệ thống lưu trữ thông tin cá nhân, lịch sử đơn hàng, thông tin yêu thích hoặc đánh giá sản phẩm, từ đó hỗ trợ phân tích hành vi mua sắm và cải thiện chiến lược kinh doanh.

Hỗ trợ xây dựng cơ chế phân quyền: Việc phân tách tài khoản khách hàng, quản trị viên và nhân viên có thể dễ dàng thiết lập, đảm bảo vận hành website an toàn, minh bạch và hiệu quả.

Tăng độ tin cậy và bảo mật cho hệ thống: Laravel đảm bảo mật khẩu người dùng được mã hóa, quy trình đăng ký được kiểm tra dữ liệu đầu vào nghiêm ngặt, hạn chế lỗi bảo mật và nâng cao tính an toàn cho website thương mại điện tử.

2.1.3. Quy trình hoạt động:

1. Người dùng truy cập trang đăng ký .
2. Nhập thông tin cá nhân như: Tên, Email, File thông tin, Mật khẩu và xác nhận mật khẩu
3. Hệ thống kiểm tra tính hợp lệ của dữ liệu:
 - Email đúng định dạng, chưa tồn tại
 - Mật khẩu đủ mạnh
 - Xác nhận mật khẩu trùng khớp
4. Mã hóa mật khẩu bằng bcrypt (thông qua Hash::make()).
5. Lưu thông tin vào bảng users trong database.

2.1.4. Các thành phần liên quan

Model – User

- Đại diện cho đối tượng người dùng.
- Chứa thuộc tính được phép fill (fillable).
- Tự động mã hóa mật khẩu qua “Attribute Casting”.

Controller – RegisterController

- Validate request
- Tạo user

Routes

- Route::get('/register', [RegisteredUserController::class, 'create']);
- Route::post('/register', [RegisteredUserController::class, 'store']);

d. Validation: Laravel sử dụng Validator hoặc \$request->validate() để đảm bảo dữ liệu hợp lệ, ví dụ:

```
$request->validate([  
'name' => ['required', 'string', 'max:255'],  
'email' => ['required', 'string', 'email', 'max:255', 'unique:users'],  
'password' => ['required', 'confirmed', Rules\Password::defaults()],
```

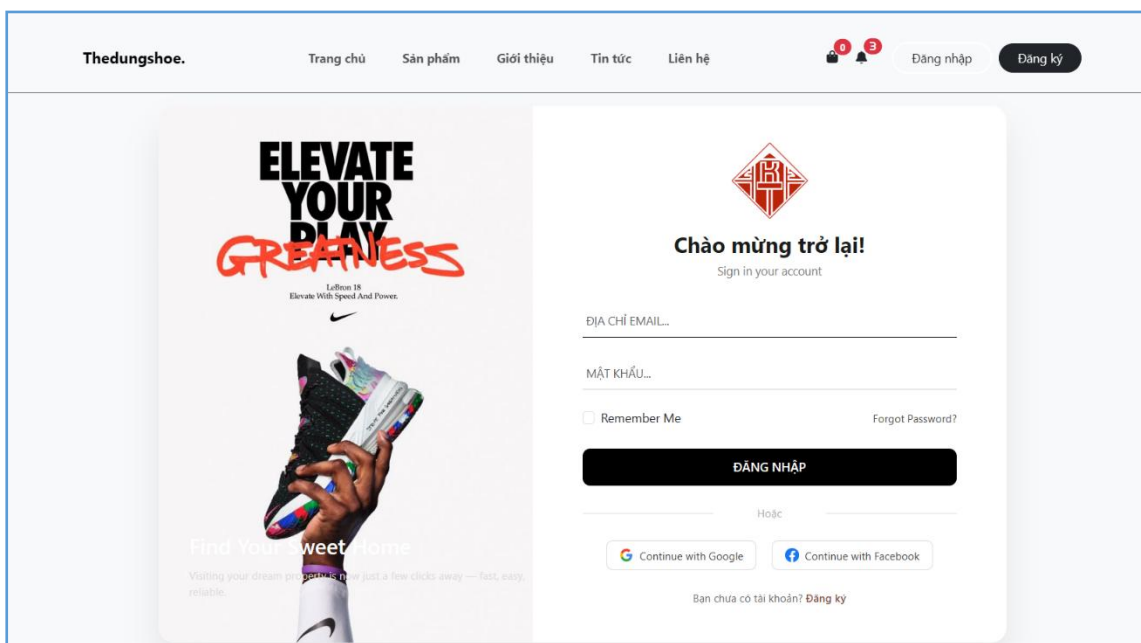
]);

2.1.5. Tính năng bảo mật: Laravel đảm bảo an toàn cho quá trình đăng ký nhờ:

- CSRF Token khi gửi form
- Hash mật khẩu bằng bcrypt
- Validation chặt chẽ
- Middleware bảo vệ route
- Xác thực session bảo mật

2.2. Chức năng Đăng nhập

2.2.1. Giao diện



Hình 2.2. Giao diện “Đăng nhập tài khoản”

2.2.2. Giới thiệu chức năng đăng nhập

Chức năng Đăng nhập tài khoản (User Login) trong Laravel cho phép người dùng đã có tài khoản truy cập vào hệ thống thông qua email và mật khẩu. Laravel cung cấp cơ chế xác thực mạnh mẽ, dễ dùng và đảm bảo tính bảo mật cao thông qua hệ thống Auth tích hợp sẵn hoặc các package như Laravel Breeze và Laravel Jetstream.

2.2.3. Mục đích

- Xác thực người dùng hợp lệ trước khi truy cập các tính năng yêu cầu quyền truy cập.
- Quản lý phiên làm việc (session) của người dùng.
- Tạo nền tảng cho việc phân quyền, cá nhân hoá chức năng trong hệ thống.

2.2.4. Quy trình hoạt động

1. Người dùng truy cập trang đăng nhập.
2. Nhập thông tin email và mật khẩu.
3. Hệ thống kiểm tra tính hợp lệ của dữ liệu nhập vào
4. Laravel sử dụng Auth::attempt() để xác thực thông tin đăng nhập.
5. Nếu thông tin đúng:
 - Khởi tạo session mới cho người dùng.
 - Chuyển hướng đến trang chủ.
6. Nếu sai, hiển thị thông báo lỗi và yêu cầu người dùng nhập lại.

2.2.5. Các thành phần liên quan

Model – User

- Đại diện cho đối tượng người dùng trong hệ thống.
- Lưu trữ thông tin email và mật khẩu đã được mã hóa bcrypt.

Controller – AuthenticatedSessionController

Xử lý logic đăng nhập:

- Validate thông tin đăng nhập.
- Kiểm tra thông tin bằng Auth::attempt().
- Khởi tạo session và chuyển hướng người dùng.

Routes

Laravel cung cấp sẵn routes đăng nhập, ví dụ:

```
Route::get('/login', [AuthenticatedSessionController::class, 'create']);
```

```
Route::post('/login', [AuthenticatedSessionController::class, 'store']);
```

d. Validation

Laravel kiểm tra dữ liệu đầu vào để đảm bảo tính hợp lệ, ví dụ:

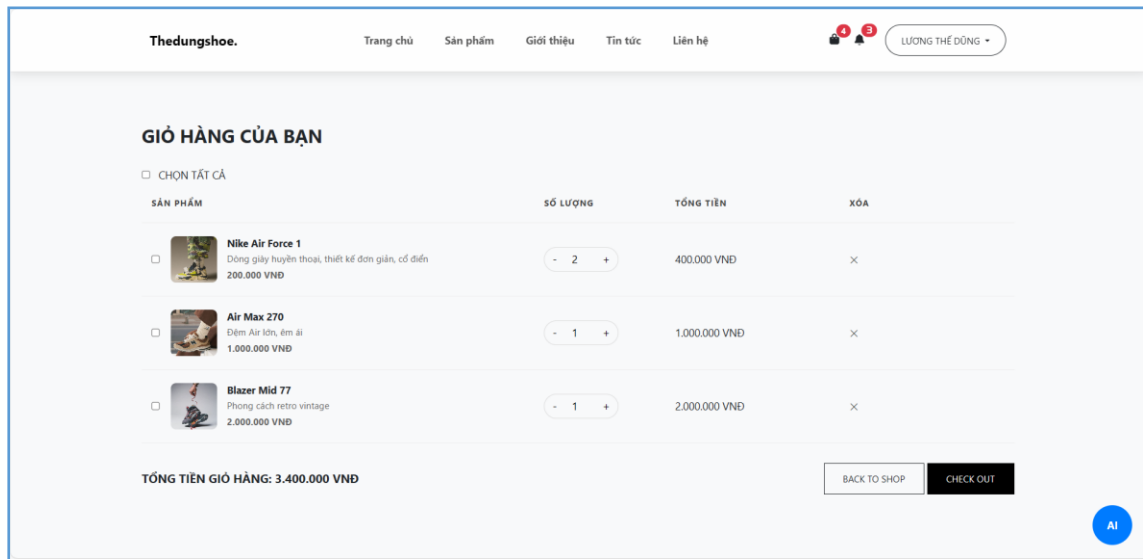
```
$request->validate([  
'email' => ['required', 'email'],  
'password' => ['required'],  
]);
```

2.2.6. Tính năng bảo mật

- CSRF Token khi gửi form.
- Mã hóa mật khẩu bằng bcrypt.
- Regenerate session khi đăng nhập thành công để tránh tấn công đánh cắp session.
- Giới hạn số lần đăng nhập sai (Login Throttling).
- Middleware auth bảo vệ các route yêu cầu đăng nhập.

2.3. Chức năng Giỏ hàng

2.3.1. Giao diện



Hình 2.3. Giao diện “Giỏ hàng”

2.3.2. Giới thiệu chức năng Giỏ hàng

Chức năng Giỏ hàng (Shopping Cart) cho phép người dùng lựa chọn và lưu tạm các sản phẩm muốn mua trước khi tiến hành thanh toán. Đây là một chức năng quan trọng trong các hệ thống thương mại điện tử, giúp nâng cao trải nghiệm mua sắm, đảm bảo người dùng có thể quản lý các mặt hàng đã chọn một cách dễ dàng và trực quan.

2.3.3. Mục đích

- Lưu trữ tạm thời các sản phẩm mà người dùng quan tâm hoặc muốn mua.
- Cho phép người dùng xem lại, tăng/giảm số lượng hoặc xóa sản phẩm khỏi giỏ.
- Hỗ trợ tính tổng giá trị đơn hàng trước khi thanh toán.
- Tạo trải nghiệm mua sắm thuận tiện, hạn chế sai sót trong quá trình đặt hàng.

2.3.4. Quy trình hoạt động

1. Người dùng duyệt sản phẩm và chọn “Thêm vào giỏ hàng”.
2. Sản phẩm được lưu vào session (nếu chưa đăng nhập) hoặc lưu trong database (nếu người dùng đăng nhập).
3. Người dùng truy cập trang giỏ hàng để xem danh sách sản phẩm.
4. Hệ thống cung cấp các thao tác:
 - Tăng hoặc giảm số lượng sản phẩm.
 - Xóa sản phẩm khỏi giỏ.
 - Tính tổng tiền tạm tính và hiển thị phí vận chuyển (nếu có).

5. Khi người dùng chọn “Tiến hành thanh toán”, hệ thống chuyển sang bước thanh toán để hoàn tất đơn hàng.

2.3.5. Các thành phần liên quan

Model – Product / CartItem

- Lưu thông tin sản phẩm: tên, giá, hình ảnh, tồn kho.
- Với hệ thống nâng cao, có thể có model Cart và CartItem để lưu giỏ hàng cho user đăng nhập.

Controller – CartController: Xử lý các chức năng chính của giỏ hàng:

- Thêm sản phẩm vào giỏ (addToCart).
- Hiện thị danh sách sản phẩm trong giỏ (index).
- Cập nhật số lượng (update).
- Xóa sản phẩm (remove).

Routes

- Route::get('/user/shoppingcart',[CartController::class,'show'])
- Route::get('/user/shoppingcart/add/{id}',[CartController::class,'add'])
- Route::get('/user/shoppingcart/remove/{rowId}',[CartController::class,'remove'])

Lưu dữ liệu giỏ hàng: Tùy hệ thống, giỏ hàng có thể được lưu:

- Trong session → phù hợp người dùng không đăng nhập.
- Trong database → phù hợp người dùng đăng nhập, đồng bộ giữa nhiều thiết bị.

2.3.6. Tính năng bảo mật

- Bảo vệ CSRF Token khi thực hiện các thao tác thêm/xóa/cập nhật.
- Kiểm tra hợp lệ số lượng sản phẩm trước khi thêm vào giỏ.
- Xác minh giá sản phẩm từ database, tránh giả mạo dữ liệu từ phía client.
- Kiểm soát sản phẩm hết hàng hoặc bị ẩn.

2.4. Chức năng Thanh toán

2.4.1. Giao diện

Hình 2.4. Giao diện “Thanh toán”

2.4.2. Giới thiệu chức năng Thanh toán

Chức năng Thanh toán (Checkout) là bước cuối cùng trong quá trình mua sắm, cho phép người dùng xác nhận giỏ hàng, cung cấp thông tin giao hàng và lựa chọn phương thức thanh toán để hoàn tất đơn hàng. Đây là chức năng quan trọng quyết định việc tạo ra đơn hàng và chuyển dữ liệu sang hệ thống xử lý đơn của cửa hàng.

2.4.3. Mục đích

- Cho phép người dùng hoàn tất quá trình mua sắm một cách nhanh chóng và chính xác.
- Thu thập thông tin giao hàng, ghi chú, và phương thức thanh toán.
- Kiểm tra và xác nhận số lượng, giá sản phẩm trước khi tạo đơn hàng.
- Tạo đơn hàng (Order) và lưu vào hệ thống để xử lý.

2.4.4. Quy trình hoạt động

1. Người dùng nhấn “Thanh toán” từ trang giỏ hàng.
2. Hệ thống hiển thị trang nhập thông tin thanh toán, bao gồm:
 - Tên người nhận
 - Số điện thoại
 - Địa chỉ giao hàng
 - Ghi chú (tùy chọn)
 - Phương thức thanh toán (COD, chuyển khoản, ví điện tử, v.v.)
3. Hệ thống kiểm tra hợp lệ dữ liệu:
 - Địa chỉ không được để trống
 - Số điện thoại đúng định dạng

- Giỏ hàng phải có sản phẩm
- 4. Xác nhận lại sản phẩm trong giỏ (check tồn kho, giá có thay đổi không).
- 5. Tạo đơn hàng (Order) và lưu thông tin xuống database.
- 6. Chuyển người dùng đến trang thông báo “Đặt hàng thành công”.

2.4.5. Các thành phần liên quan

Model – Order / OrderItem

- Order: lưu thông tin tổng quan của đơn hàng như người nhận, địa chỉ, trạng thái, tổng tiền.
- OrderItem: lưu các sản phẩm trong đơn, số lượng và giá tại thời điểm mua.

Controller – CheckoutController

- Hiển thị form thanh toán.
- Xác nhận thông tin người dùng gửi lên.
- Tạo đơn hàng và lưu xuống database.

Routes

- `Route::get('/user/checkout',[CheckOutController::class,'show']);`

Validation

```
$request->validate([
'name' => ['required', 'string'],
'phone' => ['required', 'digits_between:9,11'],
'address' => ['required', 'string'],
]);
```

2.4.6. Tính năng bảo mật

Chức năng thanh toán cần đảm bảo nhiều lớp bảo mật:

- CSRF Token bảo vệ form thanh toán.
- Kiểm tra lại giá sản phẩm và tồn kho để tránh việc chỉnh sửa dữ liệu từ client.
- Bảo vệ thông tin cá nhân người dùng.
- Hạn chế thanh toán nhiều lần (double submit).
- Với thanh toán online: cần xác minh giao dịch từ cổng thanh toán (Momo, VNPAY...).

2.5. Chức năng Liên hệ

2.5.1. Giao diện

The screenshot shows the 'Liên hệ' (Contact) page of the Thedungshoe website. The header includes the site name 'Thedungshoe.' and navigation links: 'Trang chủ', 'Sản phẩm', 'Giới thiệu', 'Tin tức', and 'Liên hệ'. There is also a user profile icon with a notification badge and a 'Thế Dừng' button. The main content area is titled 'Contact Us' and includes three columns of information: 'THỜI GIAN MỞ CỬA' (Opening Hours), 'EXHIBITIONS GALLERY', and 'GALLERY OFFICES'. Below this is a contact form with fields for 'TÊN CỦA BẠN' (Your Name), 'ĐỊA CHỈ EMAIL' (Email Address), and 'NỘI DUNG LIÊN HỆ' (Contact Content), followed by a 'GỬI' (Send) button. To the right of the form is a map showing the location of the gallery in Da Nang. Below the map is a section titled 'TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG' with a placeholder text 'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore.' and a blue 'AI' button.

Hình 2.5. Giao diện “Liên hệ”

2.5.2. Giới thiệu chức năng Thanh toán

Chức năng Liên hệ (Contact Form) cho phép người dùng gửi thông tin, phản hồi hoặc yêu cầu hỗ trợ đến hệ thống. Đây là chức năng quan trọng giúp doanh nghiệp giao tiếp trực tiếp với khách hàng, tiếp nhận câu hỏi và xử lý các vấn đề liên quan đến sản phẩm hoặc dịch vụ.

2.5.3. Mục đích

- Người dùng gửi câu hỏi, góp ý hoặc yêu cầu hỗ trợ đến bộ phận quản trị.
- Hệ thống thu thập thông tin liên hệ để phản hồi khách hàng một cách hiệu quả.
- Tạo kênh giao tiếp chính thức giữa người dùng và quản trị website.
- Hỗ trợ nâng cao chất lượng dịch vụ dựa trên phản hồi người dùng.

2.5.4. Quy trình hoạt động

1. Người dùng truy cập trang Liên hệ.
2. Nhập các thông tin yêu cầu:
 - Họ tên
 - Email
 - Số điện thoại (tùy chọn)
 - Nội dung cần liên hệ
3. Hệ thống kiểm tra tính hợp lệ của dữ liệu:
 - Email đúng định dạng
 - Nội dung không được để trống

- Tên không được rỗng
4. Hệ thống lưu thông tin liên hệ vào database hoặc gửi email đến quản trị viên.
 5. Hiện thị thông báo “Gửi thành công” và phản hồi người dùng nếu cần.

2.5.5. Các thành phần liên quan

Model – Contact: Nếu hệ thống lưu liên hệ vào cơ sở dữ liệu:

- Lưu họ tên, email, số điện thoại và nội dung liên hệ.
- Lưu thời gian gửi để quản trị viên theo dõi.

Controller – ContactController: Xử lý các thao tác liên hệ:

- Hiện thị form liên hệ.
- Validate dữ liệu người dùng gửi lên .
- Lưu vào database hoặc gửi mail cho admin.

Routes

- `Route::get('/user/contact',[ContactController::class,'contact'])->name('user.contact');`
- `Route::get('/user/contact/send',[ContactController::class,'send']);`

Validation: Laravel kiểm tra dữ liệu đầu vào:

```
$request->validate([
    'name' => ['required', 'string'],
    'email' => ['required', 'email'],
    'message' => ['required', 'string'],
]);
```

2.5.6. Tính năng bảo mật:

- CSRF Token bảo vệ form gửi dữ liệu.
- Kiểm tra validation để tránh spam hoặc dữ liệu độc hại.
- Giới hạn tần suất gửi liên hệ (nếu cần) để chống spam.
- Mã hoá dữ liệu nhạy cảm nếu liên hệ chứa thông tin quan trọng.

2.6. Chức năng Tải xuống thông tin cá nhân

2.6.1. Giao diện

Hình 2.6. Giao diện “Thông tin cá nhân”

2.6.2. Giới thiệu chức năng Tải xuống thông tin cá nhân

Chức năng “Tải xuống thông tin cá nhân” trong Laravel cho phép người dùng xuất toàn bộ dữ liệu cá nhân của họ dưới dạng file, giúp đảm bảo tính minh bạch và tuân thủ các quy định về bảo vệ dữ liệu. Cụ thể:

2.6.3. Mục đích của chức năng

- Tên người nhận
- Cung cấp cho người dùng quyền kiểm soát dữ liệu của chính họ.
- Giúp người dùng dễ dàng xem, sao lưu hoặc chuyển dữ liệu sang hệ thống khác.
- Tăng mức độ tin cậy và bảo mật cho ứng dụng.

2.6.3. Nội dung dữ liệu được tải xuống

- Thông tin tài khoản (tên, email, ngày tạo tài khoản...).
- Thông tin hồ sơ cá nhân hoặc các dữ liệu liên quan đến người dùng trong hệ thống.
- Lịch sử hoạt động hoặc các bản ghi liên quan (tùy theo cách lập trình của ứng dụng).

2.6.4. Cách hoạt động trong Laravel:

Laravel sử dụng tính năng Laravel Data Export / Personal Data Export hoặc các gói mở rộng như Laravel-Privacy, Laravel-Datatables Export, hoặc cơ chế tự xây dựng.

Khi người dùng nhấn nút “Tải xuống dữ liệu cá nhân”, hệ thống tạo file chứa dữ liệu người dùng, sau đó gửi về dưới dạng file download.

Thông thường file được xuất ở dạng JSON, CSV, hoặc ZIP tùy mục đích sử dụng.

2.6.5. Quy trình cho người dùng

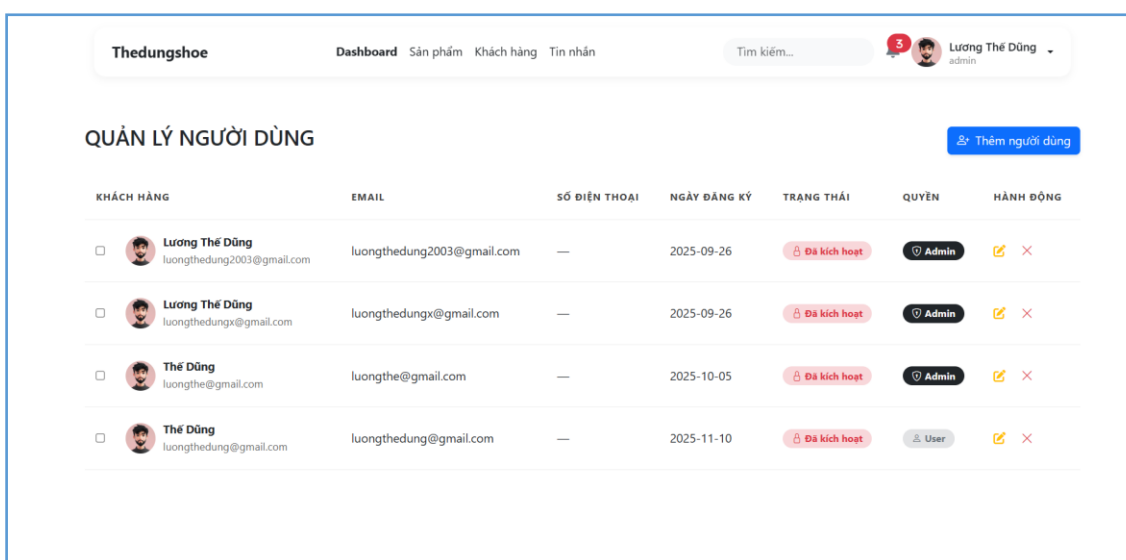
- Người dùng truy cập Trang thông tin cá nhân.
- Chọn mục “Tải xuống dữ liệu cá nhân”.
- Hệ thống xử lý và trả về file chứa đầy đủ dữ liệu cá nhân đã được xuất.













Lợi ích của chức năng:

- Giúp ứng dụng thân thiện hơn và tôn trọng quyền riêng tư người dùng.
- Tuân thủ các tiêu chuẩn quốc tế về bảo vệ dữ liệu cá nhân.
- Người dùng có thể dễ dàng kiểm tra, xem lại hoặc sao lưu thông tin của chính họ.

2.7. Chức năng Quản lý người dùng của Admin

2.7.1. Giao diện



KHÁCH HÀNG	EMAIL	SỐ ĐIỆN THOẠI	NGÀY ĐĂNG KÝ	TRẠNG THÁI	QUYỀN	HÀNH ĐỘNG
<input type="checkbox"/>  Lương Thế Dũng luongthedung2003@gmail.com	luongthedung2003@gmail.com	—	2025-09-26	Đã kích hoạt	Admin	 
<input type="checkbox"/>  Lương Thế Dũng luongthedungx@gmail.com	luongthedungx@gmail.com	—	2025-09-26	Đã kích hoạt	Admin	 
<input type="checkbox"/>  Thế Dũng luongthe@gmail.com	luongthe@gmail.com	—	2025-10-05	Đã kích hoạt	Admin	 
<input type="checkbox"/>  Thế Dũng luongthedung@gmail.com	luongthedung@gmail.com	—	2025-11-10	Đã kích hoạt	User	 

Hình 2.7. Giao diện Quản lý người dùng của Admin

2.7.2. Giới thiệu chức năng Quản lý người dùng

Chức năng “Quản lý người dùng” cho phép quản trị viên (Admin) thực hiện các thao tác quản lý toàn bộ tài khoản trong hệ thống. Đây là một phần quan trọng trong các ứng dụng Laravel có nhiều người dùng, giúp đảm bảo hệ thống vận hành ổn định, bảo mật và đúng quy định. Cụ thể:

2.7.3. Mục đích của chức năng

- Giúp Admin kiểm soát và theo dõi hoạt động của người dùng trong hệ thống.

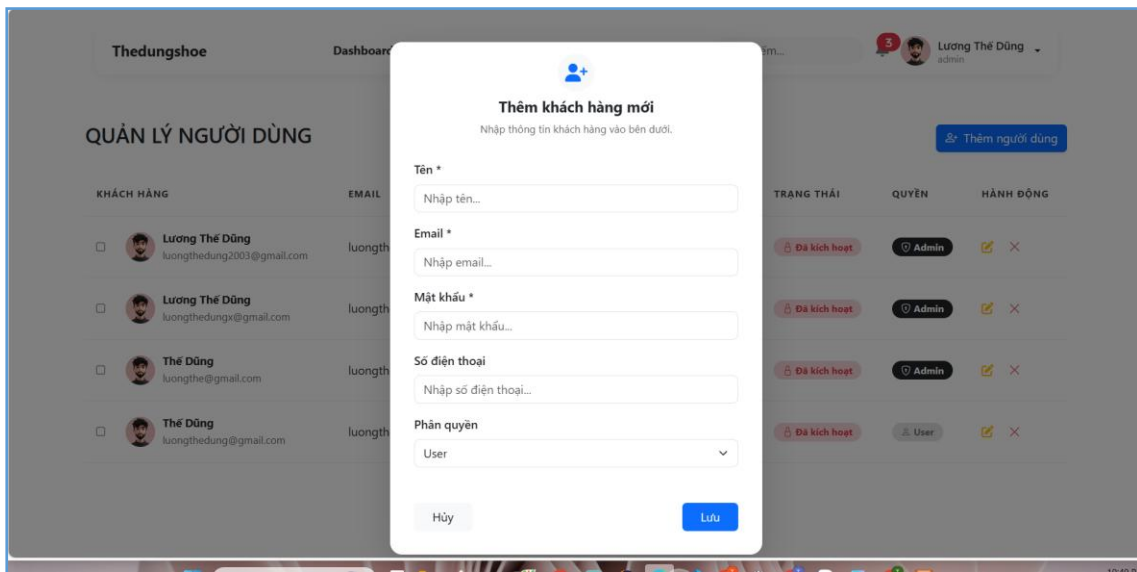
- Hỗ trợ Admin xử lý các vấn đề liên quan đến tài khoản như khóa, mở khóa hoặc cập nhật thông tin.

- Đảm bảo hệ thống luôn an toàn và không bị lạm dụng.

2.7.4. Các thao tác chính mà Admin có thể thực hiện:

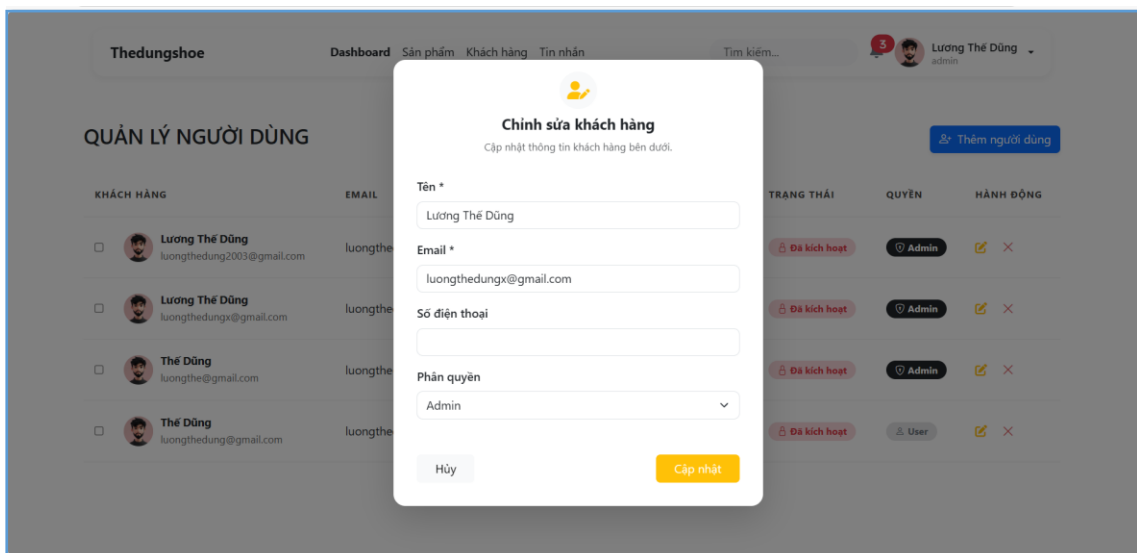
- Xem danh sách người dùng: Hiển thị toàn bộ tài khoản, bao gồm thông tin cơ bản như tên, email, vai trò, trạng thái hoạt động.

- Thêm người dùng mới: Admin có thể tự tạo tài khoản cho nhân viên



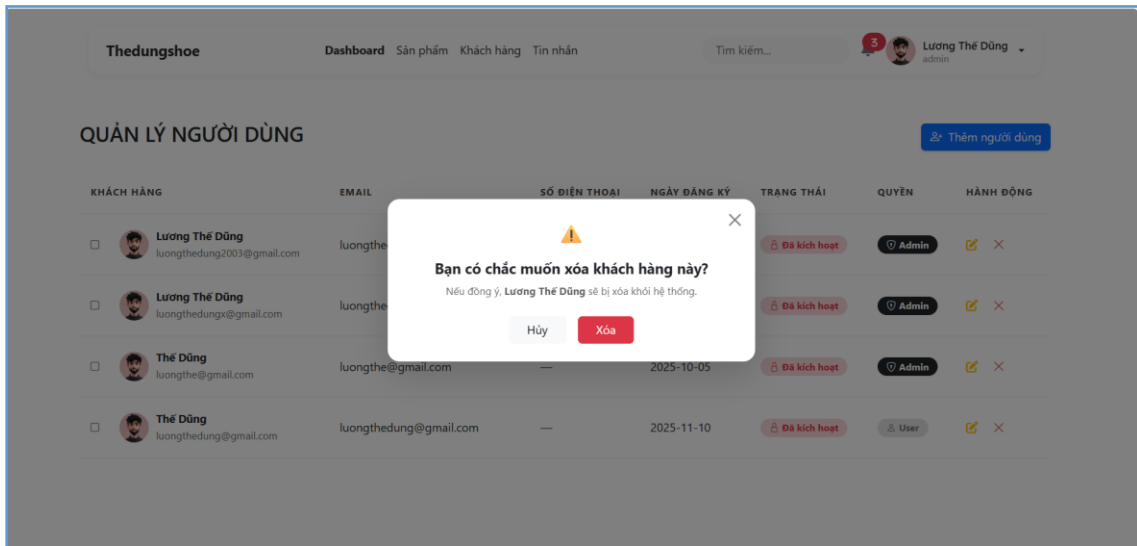
Hình 2.8. Giao diện Thêm người dùng của Admin

- Chỉnh sửa thông tin người dùng: Cho phép sửa đổi dữ liệu như họ tên, email, mật khẩu hoặc phân quyền.



Hình 2.9. Giao diện Chỉnh sửa người dùng của Admin

- Phân quyền / Thay đổi vai trò: Admin có thể đặt vai trò như *User*, *Moderator*, *Admin* để điều chỉnh mức độ truy cập.
- Khóa hoặc mở khóa tài khoản: Dùng để xử lý tài khoản vi phạm hoặc tạm ngưng hoạt động.
- Xóa người dùng: Loại bỏ tài khoản khỏi hệ thống khi không còn sử dụng.



Hình 2.10. Giao diện Xóa người dùng của Admin

- Xem chi tiết người dùng: Bao gồm thông tin cá nhân, lịch sử hoạt động hoặc các dữ liệu liên quan khác (nếu có).

Cách hoạt động trong Laravel:

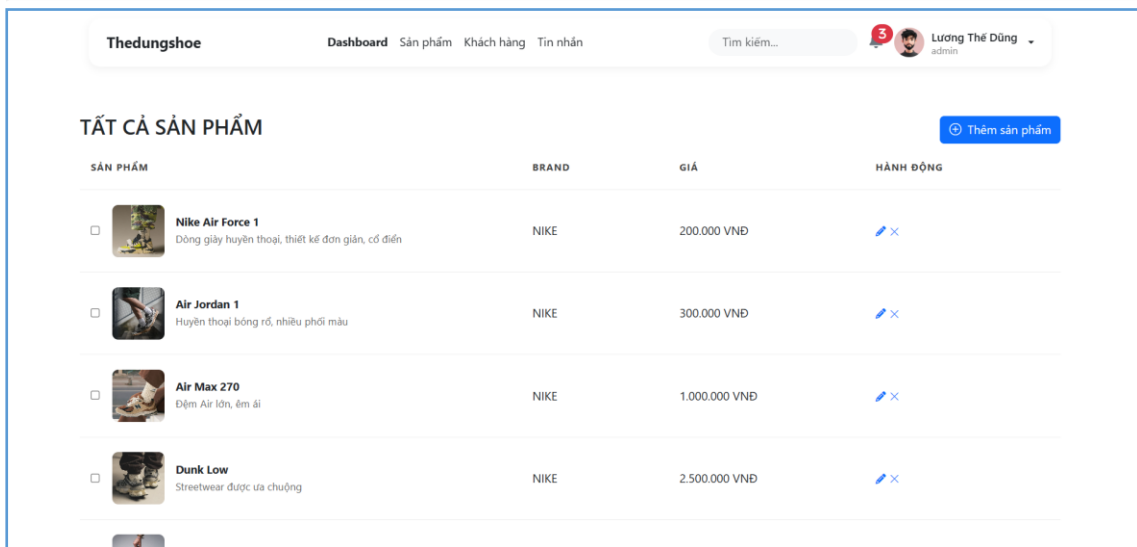
- Admin truy cập trang Dashboard quản trị.
- Hệ thống sử dụng Middleware (ví dụ: `isAdmin`) để chỉ cho phép Admin vào trang quản lý người dùng.
- Laravel sử dụng Eloquent để truy vấn bảng `users` và xử lý các thao tác CRUD.
- Các chức năng phân quyền có thể được triển khai bằng Gate, Policy, hoặc Spatie Laravel Permission.
- Mọi thao tác nhạy cảm (xóa, khóa, đổi quyền) có thể được ghi lại Log để đảm bảo tính bảo mật.

Lợi ích đối với hệ thống:

- Dễ dàng quản lý và kiểm soát toàn bộ tài khoản.
- Giảm rủi ro do tài khoản xấu gây ra.
- Giúp hệ thống vận hành ổn định và an toàn hơn.
- Cải thiện hiệu quả hỗ trợ người dùng khi có vấn đề xảy ra.

2.8. Chức năng Quản lý Sản phẩm của Admin

2.8.1. Giao diện



Hình 2.11. Giao diện Quản lý sản phẩm của Admin

2.8.2. Giới thiệu chức năng Quản lý sản phẩm

Chức năng Quản lý sản phẩm cho phép quản trị viên (Admin) thực hiện các thao tác thêm, sửa, xóa, cập nhật và kiểm soát toàn bộ sản phẩm trong hệ thống. Đây là chức năng quan trọng trong các website thương mại điện tử hoặc ứng dụng quản lý kho, giúp hệ thống luôn duy trì dữ liệu sản phẩm chính xác, đầy đủ và minh bạch.

2.8.3. Mục đích

- Cho phép Admin thêm sản phẩm mới vào hệ thống, đảm bảo thông tin sản phẩm đầy đủ.
- Quản lý số lượng tồn kho, giá bán, trạng thái sản phẩm để cập nhật kịp thời.
- Hỗ trợ Admin chỉnh sửa thông tin, phân loại hoặc xóa sản phẩm không còn kinh doanh.
- Giúp hệ thống luôn đồng bộ và cung cấp dữ liệu chính xác cho người dùng cuối.

2.8.4. Quy trình hoạt động

1. Admin truy cập trang Quản lý sản phẩm trên Dashboard.
2. Hệ thống hiển thị danh sách sản phẩm hiện có, bao gồm: tên, hình ảnh, giá, tồn kho, trạng thái hiển thị.
3. Admin thực hiện các thao tác:

- Thêm sản phẩm mới: Nhập thông tin tên, giá, mô tả, hình ảnh, tồn kho, danh mục.
 - Chỉnh sửa sản phẩm: Cập nhật thông tin hoặc thay đổi trạng thái hiển thị.
 - Xóa sản phẩm: Loại bỏ sản phẩm khỏi danh sách nếu không còn kinh doanh.
 - Cập nhật tồn kho: Điều chỉnh số lượng sản phẩm còn lại.
4. Hệ thống lưu dữ liệu mới vào cơ sở dữ liệu và thông báo kết quả thành công hoặc lỗi.

2.8.5. Các thành phần liên quan

Model – Product

- Lưu trữ thông tin sản phẩm: tên, mô tả, giá, hình ảnh, tồn kho, trạng thái.
- Hỗ trợ truy vấn và thao tác CRUD bằng Eloquent.

Controller – ProductController

Xử lý logic quản lý sản phẩm:

- index() – hiển thị danh sách sản phẩm.
- create() – hiển thị form thêm sản phẩm mới.
- store() – lưu sản phẩm mới vào database.
- edit() / update() – chỉnh sửa sản phẩm.
- destroy() – xóa sản phẩm.

Routes

- Route::get('/admin/products', [ProductController::class, 'show']);
- Route::get('/admin/products/create', [ProductController::class, 'create']);
- Route::post('/admin/products', [ProductController::class, 'add']);
- Route::get('/admin/products/{id}/edit', [ProductController::class, 'edit']);
- Route::put('/admin/products/{id}', [ProductController::class, 'update']);
- Route::delete('/admin/products/{id}', [ProductController::class, 'delete']);

Validation

```
$request->validate([
    'name' => ['required', 'string', 'max:255'],
    'price' => ['required', 'numeric', 'min:0'],
    'image' => ['nullable', 'image', 'mimes:jpg,png,jpeg,gif'],
]);
```

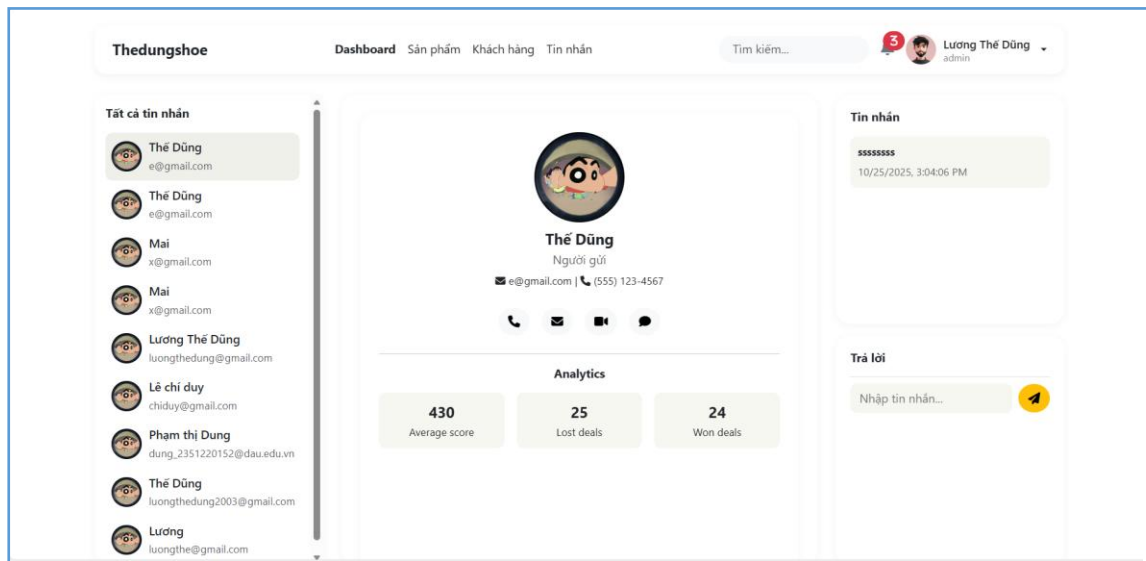
2.8.6. Tính năng bảo mật

Chức năng quản lý sản phẩm được bảo mật nhờ:

- Middleware auth và isAdmin chỉ cho phép Admin truy cập.
- CSRF Token bảo vệ các form thêm/sửa/xóa sản phẩm.
- Validation dữ liệu đầu vào để tránh lỗi hoặc tấn công từ client.
- Kiểm soát quyền thao tác để hạn chế thay đổi dữ liệu quan trọng.

2.9. Chức năng Quản lý Tin nhắn của Admin

2.9.1. Giao diện



Hình 2.12. Giao diện chức năng Quản lý tin nhắn

2.9.2. Giới thiệu chức năng Quản lý tin nhắn

Chức năng Quản lý tin nhắn liên hệ cho phép quản trị viên (Admin) xem, xử lý và quản lý toàn bộ các tin nhắn hoặc phản hồi mà người dùng gửi thông qua form Liên hệ. Đây là chức năng quan trọng giúp Admin duy trì giao tiếp hiệu quả với khách hàng, theo dõi các yêu cầu và xử lý kịp thời các vấn đề phát sinh.

2.9.3. Mục đích

- Cho phép Admin xem danh sách các tin nhắn mà người dùng gửi.
- Xử lý các yêu cầu, góp ý, khiếu nại hoặc thắc mắc của khách hàng.
- Đánh dấu trạng thái tin nhắn (chưa xử lý, đang xử lý, đã phản hồi) để quản lý hiệu quả.

- Lưu trữ dữ liệu liên hệ nhằm phục vụ báo cáo hoặc tra cứu sau này.

2.9.4. Quy trình hoạt động

1. Người dùng gửi tin nhắn thông qua form Liên hệ trên website.
2. Hệ thống lưu tin nhắn vào cơ sở dữ liệu hoặc gửi email đến Admin.
3. Admin đăng nhập vào Dashboard quản trị và truy cập mục Quản lý tin nhắn liên hệ.
4. Hệ thống hiển thị danh sách tin nhắn, gồm thông tin:
 - Tên người gửi
 - Email / số điện thoại
 - Nội dung tin nhắn
 - Ngày gửi
 - Trạng thái xử lý

5. Admin thực hiện các thao tác:

- Xem chi tiết tin nhắn.
- Đánh dấu trạng thái (Chưa xử lý / Đang xử lý / Đã phản hồi).
- Xóa tin nhắn khi cần.
- Gửi phản hồi cho người dùng (nếu cần).

2.9.5. Các thành phần liên quan

Model – ContactMessage: Lưu thông tin người gửi: tên, email, số điện thoại, nội dung tin nhắn, ngày gửi và trạng thái.

Controller – ContactMessageController

- index() – hiển thị danh sách tin nhắn.
- show() – xem chi tiết tin nhắn.
- update() – cập nhật trạng thái xử lý.
- destroy() – xóa tin nhắn.

Routes

- Route::get('/admin/messages', [ContactMessageController::class, 'index']);
- Route::get('/admin/messages/{id}', [ContactMessageController::class, 'show']);
- Route::put('/admin/messages/{id}', [ContactMessageController::class, 'update']);
- Route::delete('/admin/messages/{id}', [ContactMessageController::class, 'destroy']);

Validation

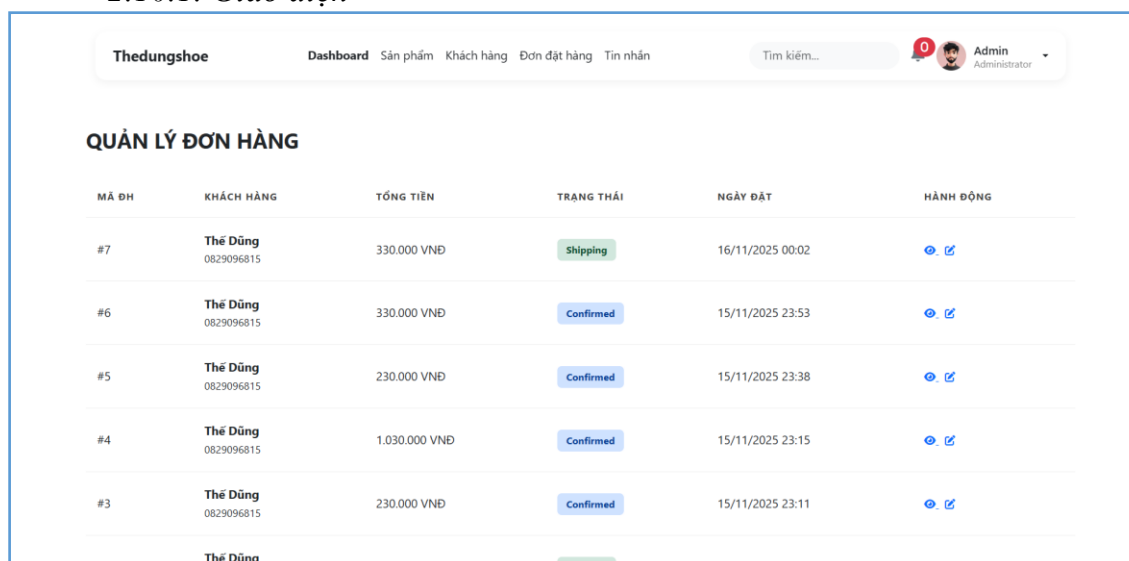
```
$request->validate([  
    'status' => ['required', 'in:unprocessed,processing,processed'],  
]);
```

2.9.6. Tính năng bảo mật

- Middleware auth và isAdmin để chỉ Admin mới truy cập được.
- CSRF Token bảo vệ form cập nhật trạng thái hoặc xóa tin nhắn.
- Validation dữ liệu đầu vào để đảm bảo trạng thái hợp lệ và tránh lỗi.
- Ghi log hành động nếu muốn theo dõi ai đã xử lý tin nhắn.

2.10. Chức năng Quản lý Đơn đặt hàng của Admin

2.10.1. Giao diện



MÃ ĐH	KHÁCH HÀNG	TỔNG TIỀN	TRẠNG THÁI	NGÀY ĐẶT	HÀNH ĐỘNG
#7	Thế Dũng 0829096815	330.000 VNĐ	Shipping	16/11/2025 00:02	🔍 🗑️
#6	Thế Dũng 0829096815	330.000 VNĐ	Confirmed	15/11/2025 23:53	🔍 🗑️
#5	Thế Dũng 0829096815	230.000 VNĐ	Confirmed	15/11/2025 23:38	🔍 🗑️
#4	Thế Dũng 0829096815	1.030.000 VNĐ	Confirmed	15/11/2025 23:15	🔍 🗑️
#3	Thế Dũng 0829096815	230.000 VNĐ	Confirmed	15/11/2025 23:11	🔍 🗑️
#2	Thế Dũng	330.000 VNĐ	Confirmed	15/11/2025 23:10	🔍 🗑️

Hình 2.13. Giao diện chức năng Quản lý đơn đặt hàng

2.10.2. Giới thiệu chức năng Quản lý đơn đặt hàng

Chức năng Quản lý đơn đặt hàng cho phép quản trị viên (Admin) xem, theo dõi, cập nhật trạng thái và quản lý toàn bộ các đơn hàng mà khách hàng đã đặt trên hệ thống. Đây là chức năng cốt lõi trong việc vận hành cửa hàng trực tuyến, giúp Admin kiểm soát quy trình bán hàng từ khi đơn hàng được tạo đến khi giao hàng thành công.

2.10.3. Mục đích

- Cho phép Admin xem danh sách tất cả các đơn hàng trong hệ thống.
- Theo dõi chi tiết thông tin từng đơn hàng: sản phẩm, số lượng, giá trị, thông tin khách hàng.
- Cập nhật trạng thái đơn hàng theo quy trình xử lý (Chờ xác nhận → Đã xác nhận → Đang giao → Hoàn thành → Đã hủy).
- Quản lý thông tin thanh toán và phương thức giao hàng.
- Thống kê doanh thu và báo cáo theo từng trạng thái đơn hàng.
- In hóa đơn hoặc xuất file đơn hàng để lưu trữ.

2.10.4. Quy trình hoạt động

- Khách hàng đặt hàng thông qua giỏ hàng và hoàn tất thông tin giao hàng.
- Hệ thống tạo đơn hàng với trạng thái mặc định là "Chờ xác nhận" và lưu vào cơ sở dữ liệu.
- Admin đăng nhập vào Dashboard quản trị và truy cập mục Quản lý đơn hàng.

- Hệ thống hiển thị danh sách đơn hàng với các thông tin: Mã đơn hàng, Tên khách hàng, Tổng giá trị đơn hàng, Trạng thái đơn hàng, Phương thức thanh toán, Ngày đặt hàng
- Admin thực hiện các thao tác: Xem chi tiết đơn hàng (sản phẩm, địa chỉ giao hàng, ghi chú), Cập nhật trạng thái đơn hàng, Xác nhận thanh toán (nếu thanh toán COD), Hủy đơn hàng (kèm lý do nếu cần)., In hóa đơn hoặc phiếu giao hàng, Gửi email thông báo cho khách hàng về trạng thái đơn hàng.

2.10.5. Các thành phần liên quan

Model – Order : Model Order lưu trữ thông tin tổng quan của đơn hàng

Model – OrderItem: Model OrderItem lưu thông tin chi tiết sản phẩm trong đơn hàng

Controller – OrderController

- Hiển thị danh sách đơn hàng
- Tìm kiếm theo mã đơn hoặc tên khách hàng
- Xem chi tiết đơn hàng
- Cập nhật trạng thái đơn hàng
- Gửi thông báo cho khách hàng
- Hủy đơn hàng

2.10.6. Tính năng bảo mật

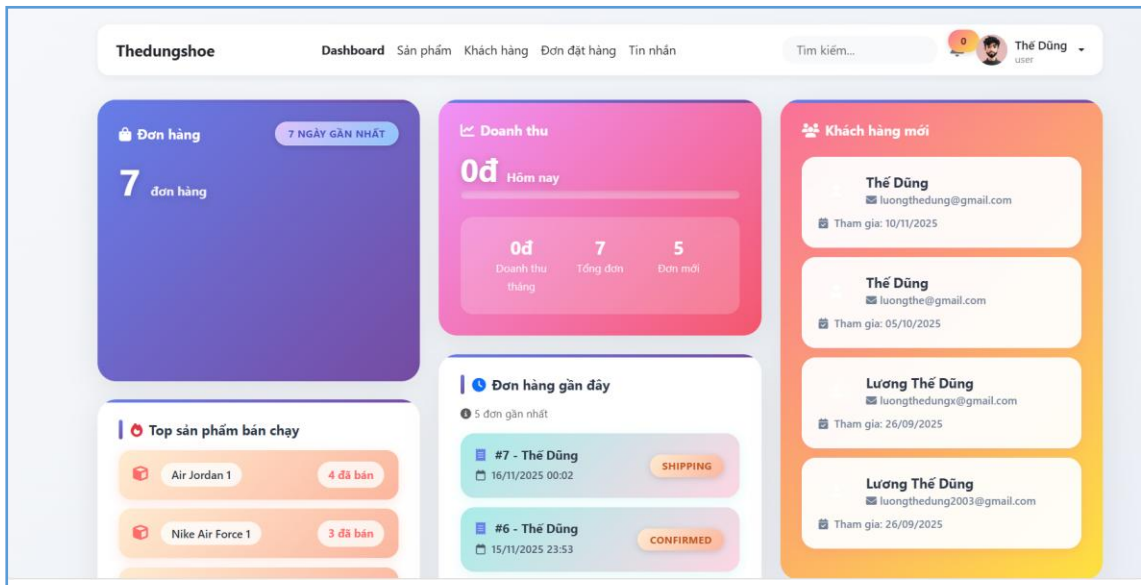
- Middleware auth và admin: Chỉ Admin mới có quyền truy cập chức năng quản lý đơn hàng.
- CSRF Token: Bảo vệ các form cập nhật trạng thái, hủy đơn và xóa đơn hàng.
- Validation dữ liệu: Đảm bảo trạng thái hợp lệ và tránh lỗi logic.
- Authorization Policy: Kiểm tra quyền thao tác trên từng đơn hàng cụ thể.
- Ghi log hành động: Theo dõi ai đã thực hiện thao tác gì trên đơn hàng.
- Kiểm tra điều kiện: Chỉ cho phép hủy/xóa đơn hàng trong trạng thái phù hợp.

2.10.7. Tính năng mở rộng

- Thống kê doanh thu: Báo cáo theo ngày/tháng/năm.
- Xuất Excel: Export danh sách đơn hàng.
- Gửi email tự động: Thông báo khách hàng khi trạng thái thay đổi.
- Tích hợp vận chuyển: Kết nối API Giao Hàng Nhanh, Giao Hàng Tiết Kiệm.
- In mã vận đơn: Tạo mã tracking cho đơn hàng.

2.11. Chức năng Dashboard (Trang tổng quan Admin)

2.11.1. Giao diện



Hình 2.14. Giao diện Dashboard của Admin

2.11.2. Giới thiệu chức năng Dashboard

Chức năng Dashboard (Trang tổng quan Admin) là màn hình đầu tiên khi quản trị viên đăng nhập vào hệ thống. Đây là nơi hiển thị toàn bộ thông tin khái quát về hoạt động của website như số lượng người dùng, sản phẩm, đơn hàng, doanh thu, thông báo mới... Giúp Admin theo dõi tình trạng hệ thống một cách trực quan và nhanh chóng.

Dashboard đóng vai trò trung tâm, kết nối tất cả các module quản trị, tạo điều kiện để Admin điều hướng đến các chức năng như: Quản lý sản phẩm, người dùng, đơn hàng, thống kê, thương hiệu...

2.11.3. Mục đích của chức năng

- Giúp Admin nắm bắt tình hình tổng quan của hệ thống trong thời gian thực.
- Hỗ trợ quản lý các dữ liệu quan trọng thông qua biểu đồ và số liệu thống kê.
- Tạo giao diện điều hướng nhanh đến các module chính trong khu vực quản trị.
- Tăng hiệu quả quản lý nhờ việc trình bày trực quan và rõ ràng.

2.11.4. Quy trình hoạt động

1. Admin đăng nhập vào hệ thống.
2. Hệ thống kiểm tra quyền truy cập thông qua middleware isAdmin.

3. Nếu hợp lệ, hệ thống chuyển hướng đến trang Dashboard.
4. Dashboard tự động lấy dữ liệu từ database như:
 - Tổng số người dùng
 - Tổng số sản phẩm
 - Số lượng đơn hàng
 - Đơn hàng mới, đơn hàng đang xử lý
 - Doanh thu (nếu có)
5. Dữ liệu được truyền sang View và hiển thị dưới dạng:
 - Hộp thống kê (stat box)
 - Bảng danh sách (latest orders)
 - Biểu đồ (chart)
6. Admin có thể bấm vào từng mục để chuyển đến trang quản lý tương ứng.

2.11.5. Các thành phần liên quan

Model – User, Product, Order

- Lấy số lượng người dùng (User::count()).
- Lấy số lượng sản phẩm (Product::count()).
- Lấy số lượng đơn hàng (Order::count()).
- Lấy danh sách đơn hàng gần đây từ bảng Orders.

Controller – AdminStatisticController

- Tên người nhận
- Truy vấn dữ liệu thống kê từ database.
- Tính toán các giá trị như tổng doanh thu, số đơn theo trạng thái.
- Gửi dữ liệu sang View bằng compact hoặc mảng.

Routes: Route::get('/admin/dashboard', [AdminStatisticController::class, 'index'])

View – admin/dashboard.blade.php

- Các thẻ thống kê (cards): tổng đơn hàng, người dùng, sản phẩm.
- Bảng hiển thị đơn hàng mới nhất.
- Biểu đồ thống kê (nếu có, dùng Chart.js).

2.11.6. Tính năng bảo mật

- Middleware isAdmin: chỉ cho phép tài khoản có quyền Admin truy cập.
- Kiểm soát session: bắt buộc Admin phải đăng nhập mới được xem Dashboard.

- CSRF Protection: bảo vệ các biểu mẫu và thao tác trong trang Dashboard.
- Validation dữ liệu đầu vào: áp dụng cho các thao tác liên quan (nếu có).
- Ẩn thông tin nhạy cảm: như doanh thu hoặc thông báo nội bộ chỉ dành cho Admin.
- Bảo vệ route bằng group prefix admin/ để hạn chế truy cập trái phép.

2.11.7. Kết luận chức năng

Chức năng Dashboard cung cấp góc nhìn tổng quan về toàn bộ hệ thống. Nhờ dữ liệu trực quan và khả năng điều hướng nhanh, Admin có thể quản lý và xử lý các vấn đề một cách hiệu quả, giúp hệ thống vận hành ổn định và dễ dàng kiểm soát.

2.12. Tiểu kết chương II

Chương II đã trình bày toàn bộ quá trình xây dựng và triển khai các chức năng chính của hệ thống website bán giày sử dụng Framework Laravel. Dựa trên nền tảng công nghệ đã phân tích ở Chương I, hệ thống được hiện thực hóa thông qua các chức năng cốt lõi như: đăng ký – đăng nhập tài khoản, quản lý giỏ hàng, xử lý thanh toán, gửi liên hệ, tải xuống dữ liệu cá nhân và các chức năng quản trị dành cho Admin. Mỗi chức năng đều được mô tả chi tiết cả về giao diện, quy trình hoạt động, thành phần kỹ thuật và các cơ chế bảo mật cần thiết.

Kết quả thực nghiệm cho thấy hệ thống hoạt động ổn định, đáp ứng đầy đủ những yêu cầu nghiệp vụ của một website thương mại điện tử quy mô nhỏ. Các chức năng tương tác trơn tru, dữ liệu được xử lý chính xác và giao diện thân thiện với người sử dụng. Nhờ tận dụng sức mạnh của Laravel, quá trình phát triển diễn ra thuận lợi, đồng thời đảm bảo tính bảo mật, khả năng mở rộng và dễ dàng bảo trì trong tương lai.

Tuy hệ thống đã hoàn thiện theo mục tiêu đề tài, vẫn còn một số hạn chế nhất định như chưa tích hợp các cổng thanh toán điện tử thực tế (Momo, VNPay), chưa có tính năng gợi ý sản phẩm theo hành vi người dùng, chưa hỗ trợ đánh giá – bình luận nâng cao và chưa xây dựng API dành cho ứng dụng di động. Đây cũng chính là những hướng phát triển khả thi trong tương lai nhằm nâng cao chất lượng hệ thống và mang lại trải nghiệm hoàn thiện hơn cho người dùng.

Tiểu kết lại, chương này đã minh chứng cho tính khả thi của mô hình thiết kế được trình bày ở Chương I, đồng thời hoàn thiện quá trình xây dựng hệ thống bán giày trực tuyến theo các yêu cầu đặt ra ban đầu của đề tài.

TÀI LIỆU THAM KHẢO

- [1] Taylor Otwell. Laravel Documentation. Laravel Framework Official Website.
Truy cập từ: <https://laravel.com/docs>
- [2] PHP Group. PHP Manual – The Official PHP Documentation.
Truy cập từ: <https://www.php.net/manual/en/>
- [3] W3Schools. “PHP Tutorial – Learn PHP Programming.”
Truy cập từ: <https://www.w3schools.com/php/>
- [4] MySQL Documentation Team. MySQL Reference Manual.
Truy cập từ: <https://dev.mysql.com/doc/>
- [5] MDN Web Docs. “Client-Side Web Development: HTML, CSS, JavaScript.”
Truy cập từ: <https://developer.mozilla.org/>
- [6] FreeCodeCamp. “Introduction to Web Development, Laravel & PHP Articles.”
Truy cập từ: <https://www.freecodecamp.org/news/>
- [7] Stack Overflow Community. "Laravel, PHP and Web Development Q&A Discussions."
Truy cập từ: <https://stackoverflow.com/>
- [8] Gloudemans, B. Laravel Shoppingcart Package Documentation.
Truy cập từ: <https://github.com/bumbummen99/LaravelShoppingcart>
- [9] Bootstrap Team. Bootstrap 5 Documentation.
Truy cập từ: <https://getbootstrap.com/docs/5.0/>
- [10] Jeffrey Way. Laracasts – Laravel Tutorials & Training Videos.
Truy cập từ: <https://laracasts.com/>
- [11] VNPAY Payment Gateway. “Quy trình tích hợp thanh toán trực tuyến.”
Truy cập từ: <https://vnpay.vn/>
- [12] Nguyễn Thị Ngọc & Trần Quang Huy. Giáo trình Lập trình Web với PHP và MySQL. NXB Đại học Quốc gia TP.HCM, 2020.
- [13] Nguyễn Văn Hiếu. Laravel Cơ bản đến Nâng cao. Nhà xuất bản Hồng Đức, 2022.