

**TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO KẾT THÚC MÔN HỌC
CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:
XÂY DỰNG WEB QUẢN LÝ CỬA HÀNG
BÁN ĐỒ THỂ THAO**

Giáo viên hướng dẫn:

Ts. Nguyễn Bảo Ân

Nhóm sinh viên thực hiện:

Đào Công Duy (110122058-DA22TTD)

Nguyễn Phi Hùng (110122079 – DA22TTD)

Trần Tấn Khá (110122087 – DA22TTD)

Trà Vinh, tháng 7 năm 2025

LỜI CẢM ƠN

Trước hết, nhóm xin bày tỏ lòng biết ơn chân thành đến **thầy Nguyễn Bảo Ân**, người đã trực tiếp hướng dẫn, luôn sẵn sàng giải đáp thắc mắc và đưa ra những nhận xét xác đáng, giúp nhóm em hoàn thiện sản phẩm một cách hiệu quả.

Bên cạnh đó, chúng em cũng xin trân trọng cảm ơn **toàn thể giảng viên trường Kỹ thuật và Công nghệ**, đặc biệt là các thầy cô trong **khoa Công nghệ Thông tin**, những người đã truyền đạt kiến thức nền tảng, tạo điều kiện học tập tốt nhất và không ngừng truyền cảm hứng học hỏi cho sinh viên.

Thông qua quá trình thực hiện đồ án, nhóm không chỉ củng cố kiến thức chuyên môn mà còn rèn luyện được kỹ năng làm việc nhóm, giải quyết vấn đề và tổ chức thực hiện dự án. Đây là những kinh nghiệm quý giá đối với mỗi thành viên trong nhóm.

Dù đã nỗ lực hoàn thiện trong khả năng của mình, nhưng chắc chắn bài làm không tránh khỏi những thiếu sót. Nhóm em rất mong nhận được các ý kiến đóng góp từ quý thầy cô để cải thiện và phát triển hơn trong tương lai.

Một lần nữa, nhóm xin gửi lời cảm ơn chân thành đến tất cả quý thầy cô!

Trà Vinh, ngày 23 tháng 7 năm 2025

Sinh viên ký và ghi rõ họ và tên

Sinh viên 1

Sinh viên 2

Sinh viên 3

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU	11
1.1. Lý do chọn đề tài	9
1.2. Mục Tiêu.....	9
CHƯƠNG 2: PHÂN TÍCH YÊU CẦU HỆ THỐNG.....	11
2.1 Các chức năng chính của hệ thống	11
2.1.1. Chức năng dành cho khách hàng.....	11
2.1.2. Chức năng dành cho quản trị viên.....	12
2.2. Các yêu cầu phi chức năng.....	12
2.2.1. Hiệu năng.....	12
2.2.2. Khả năng mở rộng	13
2.2.3. Tính dễ sử dụng	13
2.2.4. Tính duy trì và phát triển.....	13
2.2.5. Tính bảo mật và an toàn.....	13
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG.....	14
3.1. Kiến trúc hệ thống.....	14
3.1.1 Client (Giao diện người dùng Frontend – ReactJS).....	14
3.1.2 Server (Node.js + ExpressJS).....	14
3.1.3 Cơ sở dữ liệu (MySQL).....	14
3.1.4 Docker.....	14
3.1.5 CI/CD.....	14
3.2. Thiết kế cơ sở dữ liệu.....	15
3.2.1. Mô hình thực thể - quan hệ (ERD).....	15
3.3. Thiết kế API.....	16
3.1.1 Tổng quan.....	16
3.1.2 Nhóm endpoint chính.....	17
3.3.3 Cấu trúc parameters/response.....	18
3.4. Thiết kế giao diện (UI/UX).....	26
3.4.1. Giao diện trang chủ người dùng	27
3.4.2. Giao diện đăng nhập và đăng ký người dùng.....	28
3.4.3. Giao diện sản phẩm.....	29
3.4.4. Giao diện giỏ hàng và thanh toán.....	32

3.4.5. Giao diện đơn hàng người dùng.....	34
3.4.6. Giao diện quản trị hệ thống của Admin.....	35
CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG	43
4.1. Ngôn ngữ lập trình và framework.....	43
4.2. Thiết lập CI/CD với GitHub Actions.....	43
4.3. Docker hoá hệ thống.....	43
4.4. Công cụ và dịch vụ hỗ trợ.....	43
4.5. Môi trường triển khai.....	44
4.6. Đánh giá quy trình triển khai.....	44
CHƯƠNG 5: QUẢN LÝ DỰ ÁN	45
5.1. Phương pháp luận phát triển phần mềm.....	45
5.2. Sử dụng Jira để lập kế hoạch và theo dõi tiến độ.....	45
5.2.1.Sprint 1.....	45
5.2.2.Sprint 2.....	45
5.2.3. Sprint 3.....	46
5.2.4. Sprint 4.....	46
5.3. Phân công nhiệm vụ của từng thành viên trong nhóm.....	46
CHƯƠNG 6: KIỂM THỬ	49
6.1. Chiến lược kiểm thử	49
CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN.....	50
7.1. Kết luận.....	50
7.2. Hạn chế.....	50
7.3. Hướng phát triển trong tương lai.....	51
7.3.1. Mở rộng chức năng.....	51
7.3.2. Cải tiến kiến trúc kỹ thuật.....	51
7.3.3. Phát triển ứng dụng di động (Mobile App).....	51
7.3.4. Áp dụng trí tuệ nhân tạo (AI).....	51
TÀI LIỆU THAM KHẢO	52

DANH MỤC HÌNH ẢNH

Hình 2.1. Sơ đồ use-case chức năng.....	11
Hình 3.1. Mô hình ERD.....	15
Hình 3.2. Parameters và response khi đăng nhập thành công của admin.....	18
Hình 3.3. Parameters và response sau khi đăng nhập thành công phía Khách hàng.....	19
Hình 3.4. Parameters và response khi đặt hàng và quản lý đơn hàng.....	20
Hình 3.5. Parameters và response khi hủy đơn hàng.....	21
Hình 3.6. Parameters và response khi xem chi tiết đơn hàng.....	21
Hình 3.7. Parameters và response khi xem đơn hàng.....	22
Hình 3.8. Parameter và response khi xem thông tin profile.....	23
Hình 3.9. Parameters và response khi cập nhật trạng thái đơn hàng.....	24
Hình 3.10. Parameter và response khi thêm sản phẩm mới.....	25
Hình 3.11. Parameter và response khi xem sản phẩm.....	26
Hình 3.12. Parameter và response khi xem chi tiết sản phẩm.....	26
Hình 3.13 Giao diện trang chủ của người dùng	27
Hình 3.14 Giao diện Đăng Nhập Admin và khách hàng.....	28
Hình 3.15. Giao diện đăng ký tài khoản người dùng.....	28
Hình 3.16 Giao diện sau khi đăng nhập thành công.....	29
Hình 3.17 Giao diện phía trên Sản Phẩm.....	30
Hình 3.18 Giao diện phía dưới Sản Phẩm.....	30
Hình 3.19. Giao diện tìm kiếm và lọc sản phẩm.....	31
Hình 3.20. Giao diện chi tiết sản phẩm khi nhấp vào sản phẩm.....	31
Hình 3.21. Giao diện khi người dùng nhấp vào thêm giỏ hàng(nút giỏ hàng góc trên bên phải sẽ hiển thị số sản phẩm sau khi nhấp vào thêm giỏ hàng).....	32
Hình 3.22. Giao diện giỏ hàng khi người dùng nhấp vào giỏ hàng góc phải màn hình. .	33
Hình 3.23. Khi người dùng nhấn vào nút đặt hàng ngay (Khi chưa nhập thông tin cá nhân).....	33
Hình 3.24. Giao diện khi người dùng đặt hàng thành công.....	34
Hình 3.25. Giao diện Đơn Hàng người dùng.....	34
Hình 3.26. Giao diện chi tiết đơn hàng người dùng.....	35
Hình 3.27. Giao diện lọc Đơn Hàng.....	35
Hình 3.28. Giao diện các thành phần trong Quản lý của Admin.....	36

Hình 3.29. Giao diện Quản lý sản phẩm của Admin.....	37
Hình 3.30. Giao diện thêm sản phẩm mới vào cửa hàng của Admin.	38
Hình 3.31. Giao diện Quản lý Đơn Hàng khi khách hàng đặt hàng thành công của Admin	39
Hình 3.32. Giao diện cập nhật trạng thái đơn hàng của Admin.....	39
Hình 3.33. Giao diện sau khi cập nhập lại trạng thái đơn hàng của Admin.....	40
Hình 3.34. Giao diện Quản lý Users của Admin.....	40
Hình 3.35. Giao diện tìm kiếm và bộ lọc Users của Admin.....	41
Hình 3.36. Giao diện báo cáo về số đơn hàng đã đặt và danh thu.....	42
Hình 3.37. Giao diện thống kê tăng trưởng, người dùng và thời gian hoạt động.....	42
Hình 5.1. Sprint Backlog của Sprint 1	45
Hình 5.2. Sprint Backlog của Sprint 2	45
Hình 5.3. Sprint Backlog của Sprint 3	46
Hình 5.4. Sprint Backlog của Sprint 4.....	46

DANH MỤC BẢNG BIỂU

Bảng 1. Mô tả các quan hệ giữa các thực thể trong ERD.....	16
Bảng 2. Cấu hình truy cập API.....	17
Bảng 3. Bảng phân công nhiệm vụ các Sprint 1, 2, 3 và 4 trong Jira.....	46

CHƯƠNG 1: GIỚI THIỆU

1.1. Lý do chọn đề tài

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc ứng dụng các hệ thống quản lý vào hoạt động kinh doanh ngày càng trở nên phổ biến và cần thiết. Đặc biệt, trong lĩnh vực bán lẻ, các cửa hàng truyền thống đang dần chuyển mình sang mô hình quản lý hiện đại hơn nhằm nâng cao hiệu quả hoạt động, tối ưu hóa quy trình làm việc và phục vụ khách hàng tốt hơn.

Ngành hàng thể thao là một lĩnh vực đang phát triển mạnh, nhu cầu mua sắm các sản phẩm như quần áo thể thao, giày dép, dụng cụ thể thao ngày càng gia tăng. Tuy nhiên, nhiều cửa hàng thể thao vẫn quản lý thủ công bằng giấy tờ hoặc file Excel, dẫn đến việc quản lý hàng hóa, đơn hàng, khách hàng và doanh thu gặp nhiều khó khăn, dễ xảy ra sai sót.

Xuất phát từ thực tế đó, em chọn đề tài **"Xây dựng website quản lý cửa hàng bán đồ thể thao"** nhằm mục tiêu xây dựng một hệ thống phần mềm giúp cửa hàng:

- Quản lý sản phẩm, tồn kho, nhập hàng và bán hàng hiệu quả hơn.
- Quản lý thông tin khách hàng, hóa đơn và doanh thu một cách rõ ràng, chính xác.
- Hỗ trợ người quản lý dễ dàng theo dõi hoạt động kinh doanh mọi lúc, mọi nơi thông qua nền tảng web.
- Tạo bước đệm cho việc chuyển đổi số trong kinh doanh bán lẻ.

Bên cạnh đó, việc thực hiện đề tài này cũng giúp em vận dụng kiến thức đã học về lập trình web, cơ sở dữ liệu, phân tích và thiết kế hệ thống vào một ứng dụng thực tế, từ đó nâng cao kỹ năng và kinh nghiệm phát triển phần mềm.

1.2. Mục Tiêu

Đề tài **"Xây dựng website quản lý cửa hàng bán đồ thể thao"** hướng đến việc phát triển một nền tảng quản lý bán hàng mạnh mẽ, bảo mật và dễ triển khai, nhằm phục vụ cho các cửa hàng vừa và nhỏ trong việc số hóa quy trình kinh doanh. Các mục tiêu cụ thể bao gồm:

1. **Xây dựng hệ thống quản lý bán hàng hoàn chỉnh** với các chức năng cốt lõi như quản lý sản phẩm, đơn hàng, người dùng và thống kê doanh thu.
2. **Phát triển API RESTful có cấu trúc rõ ràng, dễ tích hợp với frontend** thông qua tài liệu Swagger UI, hỗ trợ phát triển frontend nhanh chóng và hiệu quả.

3. Áp dụng mô hình xác thực và phân quyền bằng JWT và Role-based Access

Control, giúp đảm bảo an toàn truy cập và phân biệt quyền hạn giữa người dùng thường và quản trị viên.

4. Tích hợp quản lý người dùng với khả năng xác thực, đăng ký và cập nhật hồ sơ cá nhân, bao gồm cả kiểm tra tính đầy đủ và hợp lệ của thông tin.

5. Xây dựng hệ thống quản lý đơn hàng đa tầng, cho phép người dùng tạo, theo dõi và hủy đơn hàng, trong khi admin có thể quản lý toàn bộ đơn hàng và cập nhật trạng thái giao hàng.

6. Triển khai cơ sở dữ liệu MySQL với ORM JPA/Hibernate, đảm bảo hiệu năng truy xuất dữ liệu và dễ bảo trì.

7. Tự động hóa quá trình triển khai bằng Docker và Docker Compose, cho phép khởi động toàn bộ hệ thống chỉ với một lệnh, giảm thiểu lỗi và tiết kiệm thời gian cấu hình.

8. Tích hợp Spring Actuator để theo dõi sức khỏe hệ thống, phục vụ công tác vận hành và giám sát.

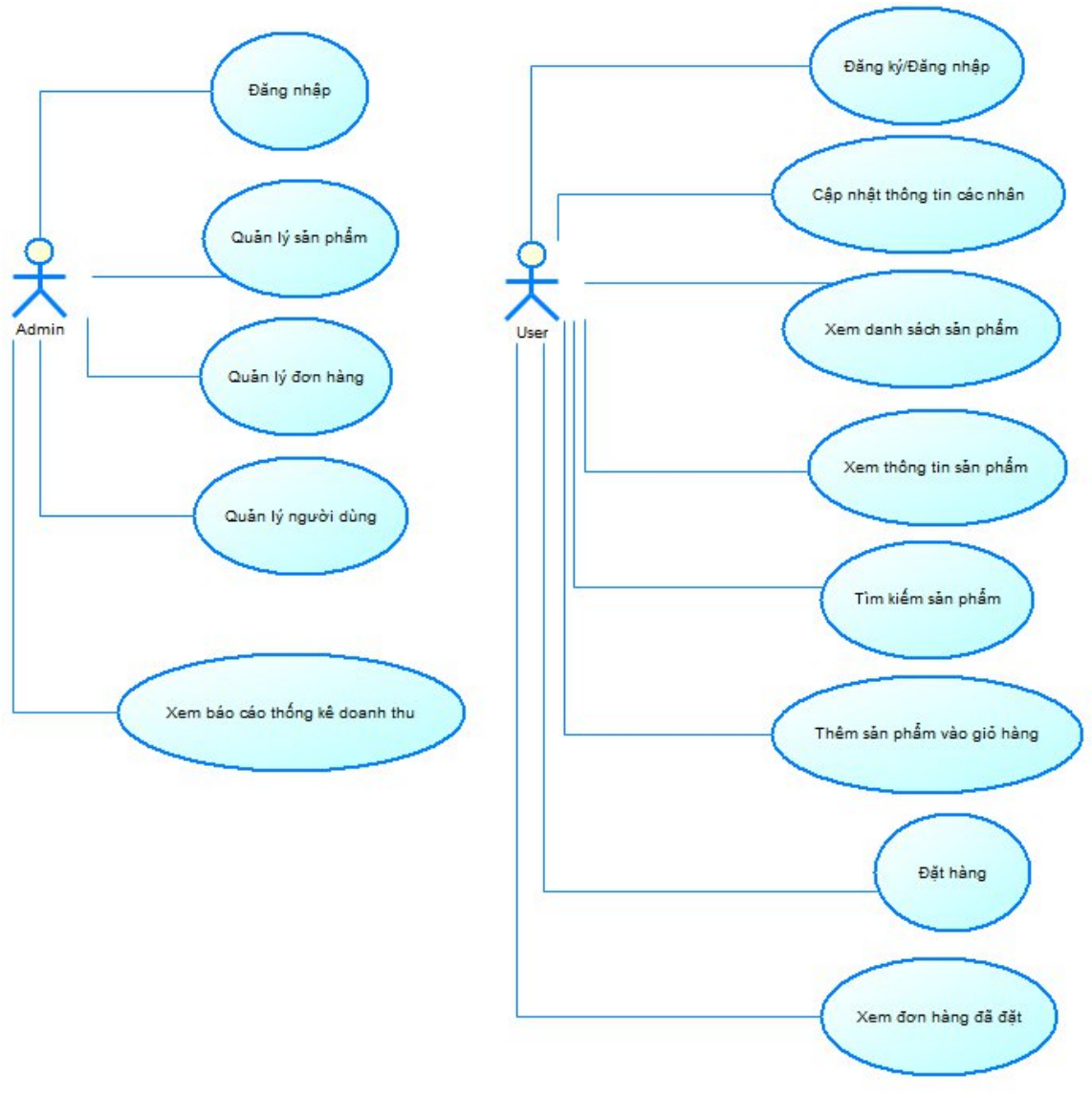
9. Thiết kế hệ thống mở rộng dễ dàng, có thể tích hợp thêm các tính năng nâng cao như: thanh toán trực tuyến, gửi email, theo dõi đơn hàng theo thời gian thực, và frontend UI sau này.

10. Tạo môi trường thực hành áp dụng kiến thức lập trình Spring Boot, bảo mật web, Docker hóa ứng dụng, và thiết kế hệ thống thực tế, phục vụ học tập và phát triển kỹ năng chuyên môn.

CHƯƠNG 2: PHÂN TÍCH YÊU CẦU HỆ THỐNG

2.1 Các chức năng chính của hệ thống

Hệ thống đặt vé xem phim trực tuyến bao gồm hai nhóm chức năng chính: dành cho người dùng (khách hàng) và dành cho quản trị viên (admin).



Hình 2.1. Sơ đồ use-case chức năng

2.1.1. Chức năng dành cho khách hàng

- Đăng ký tài khoản và đăng nhập: Người dùng có thể tạo tài bằng tên đăng nhập và khẩu.
- Cập nhật thông tin cá nhân.
- Xem danh sách sản phẩm: Hiển thị toàn bộ sản phẩm đang đăng bán (tên sản phẩm, hình ảnh, mô tả, số lượng trong kho, giá sản phẩm).

- Tìm kiếm sản phẩm: Cho phép người dùng tìm kiếm sản phẩm theo mã hoặc tên sản phẩm.
- Thêm sản phẩm vào giỏ hàng: Người dùng khi muốn chọn 1 món đồ muốn mua thì sẽ thêm sản phẩm vào giỏ hàng
- Đặt hàng: Người dùng sau khi thêm sản phẩm vào giỏ hàng, khi quyết định muốn mua thì người dùng sẽ đặt hàng sản phẩm và thanh toán
- Xem đơn hàng đã đặt: Người dùng có thể xem lại đơn hàng đã đặt và theo dõi trạng thái đơn hàng.

2.1.2. Chức năng dành cho quản trị viên

- Đăng nhập quản trị: Chỉ admin mới được phép truy cập vào giao diện quản trị.
- Quản lý sản phẩm:
 - + Xem danh sách sản phẩm đang có.
 - + Thêm, xóa sửa thông tin sản phẩm, tìm kiếm sản phẩm
- Quản lý người dùng:
 - + Xem danh sách người dùng đăng ký.
 - + Sửa, xóa người dùng, phân quyền người dùng.
- Quản lý đơn hàng:
 - + Xem danh sách đơn hàng khách hàng đã đặt.
 - + Xem chi tiết đơn hàng
 - + Cập nhật trạng thái đơn hàng.
 - + Xóa đơn hàng.
- Xem báo cáo thống kê:
 - + Xem báo cáo về số đơn hàng đã đặt và doanh thu
 - + Xem thống kê tăng trưởng và người dùng

2.2. Các yêu cầu phi chức năng

2.2.1. Hiệu năng

- Hệ thống phải có thời gian phản hồi dưới 2 giây cho các thao tác thông thường như đăng nhập, xem sản phẩm, tạo đơn hàng.
- Hệ thống phải có khả năng xử lý đồng thời ít nhất 20 người dùng cùng lúc mà không bị treo hay gián đoạn.

2.2.2. Khả năng mở rộng

Hệ thống được thiết kế linh hoạt, dễ dàng mở rộng thêm chức năng mới như: báo

2.2.3. Tính dễ sử dụng

- Giao diện web thân thiện, dễ hiểu, dễ thao tác cho người dùng phổ thông.
- Sử dụng ngôn ngữ tiếng Việt, thiết kế bố cục rõ ràng.
- Hỗ trợ các thao tác cơ bản như: tìm kiếm, lọc sản phẩm theo loại.

2.2.4. Tính duy trì và phát triển

- Hệ thống được xây dựng với kiến trúc rõ ràng (tách biệt giữa frontend, backend và database), giúp dễ dàng phát hiện và sửa lỗi.
- Mã nguồn có cấu trúc hợp lý, được chú thích đầy đủ để sinh viên khác hoặc người bảo trì sau có thể dễ dàng đọc hiểu và chỉnh sửa.
- Tài liệu kỹ thuật và hướng dẫn sử dụng hệ thống được cung cấp nhằm hỗ trợ việc bảo trì về sau.
- Hệ thống cho phép dễ dàng mở rộng thêm tính năng trong tương lai như: thống kê doanh thu, quản lý khuyến mãi, tích hợp thanh toán trực tuyến,...
- Thiết kế cơ sở dữ liệu linh hoạt, có khả năng mở rộng bảng và mối quan hệ mà không làm ảnh hưởng lớn đến hệ thống hiện tại.
- Có thể tích hợp thêm với các hệ thống khác như API thanh toán, hệ thống email, hoặc ứng dụng di động nếu phát triển thêm.

2.2.5. Tính bảo mật và an toàn

- Người dùng phải đăng nhập để truy cập các chức năng quản trị (thêm, sửa, xóa sản phẩm, đơn hàng).
- Chỉ người dùng có vai trò admin mới được phép truy cập giao diện quản trị.
- Dữ liệu nhạy cảm (mật khẩu người dùng, thông tin khách hàng) phải được mã hóa trong cơ sở dữ liệu.

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1. Kiến trúc hệ thống.

Hệ thống được phát triển theo mô hình Client–Server, chia thành 3 thành phần chính:

3.1.1 Client (Giao diện người dùng Frontend – ReactJS)

Sử dụng NextJS (đã có sẵn) giúp tối ưu hiệu năng và SEO. Frontend triển khai UI/UX từ thiết kế Figma và giao tiếp backend qua RESTful API.

3.1.2 Server (Node.js + ExpressJS)

Được xây dựng với NodeJS (Express). Backend chịu trách nhiệm xử lý nghiệp vụ, truy vấn dữ liệu, bảo mật API, sinh Swagger docs, thực hiện CI/CD.

3.1.3 Cơ sở dữ liệu (MySQL)

Lưu trữ dữ liệu về sản phẩm, đơn hàng, người dùng, loại sản phẩm, phân quyền,...

3.1.4 Docker

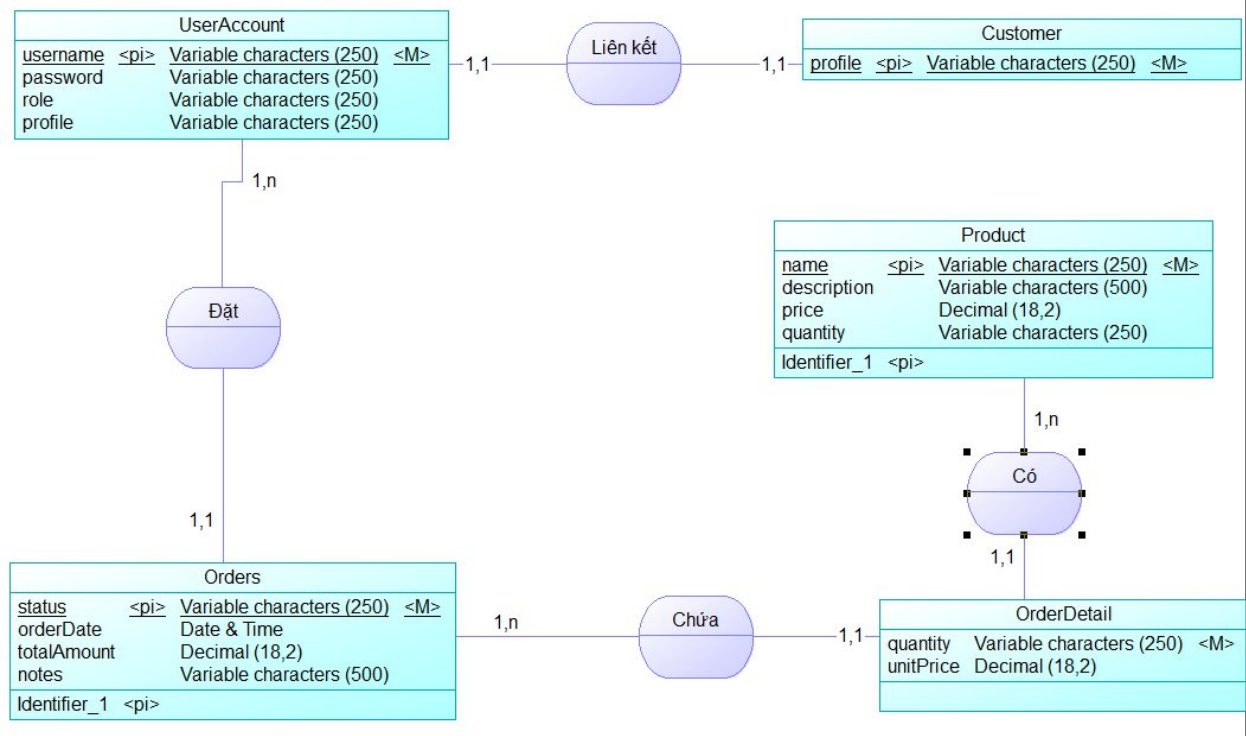
Đóng gói toàn bộ hệ thống frontend, backend, database thành container, đồng nhất môi trường triển khai.

3.1.5 CI/CD

Sử dụng GitHub Actions tự động test backend/API và deploy container lên Docker.

3.2. Thiết kế cơ sở dữ liệu

3.2.1. Mô hình thực thể - quan hệ (ERD)



Hình 3.1. Mô hình ERD

Thực thể	Thuộc tính	Khóa chính	Khóa ngoại	Mối quan hệ
1. UserAccount	username, password, role, profile	username		1:N Orders
2. Product	name, description, price, quantity	name		Product 1:N OrderDetails

3. Orders	status, orderDate, totalAmount, notes	status		Order 1:N OrderDetails
4. OrderDetail	quantity, unitPrice	quantity		OrderDetails 1:1 Order OrderDetails 1:1
5. Customer	profile	profile		User 1:1 Customer

Bảng 1. Mô tả các quan hệ giữa các thực thể trong ERD

- Mỗi quan hệ:
 - + 1-n: Quan hệ một-nhiều (ví dụ: một Cinema Cluster có nhiều Cinema).
 - + n-1: Quan hệ nhiều-một (ví dụ: nhiều Ticket thuộc về 1 Order).

3.3. Thiết kế API

3.1.1 Tổng quan

Tất cả API backend tuân theo kiểu RESTful, chia theo resource:

Mục	Giá trị
API Base URL:	http://localhost:8082/api

Swagger UI:	http://localhost:8082/api/swagger-ui/index.html
Health Check:	http://localhost:8082/api/actuator/health

Bảng 2. Cấu hình truy cập API

3.1.2 Nhóm endpoint chính

a) PUBLIC ENDPOINTS (Không cần đăng nhập)

POST	/api/v1/auth/register	- Đăng ký tài khoản (cần username, password, fullName)
POST	/api/v1/auth/login	- Đăng nhập
GET	/api/v1/products/view	- Xem danh sách sản phẩm
GET	/api/v1/products/view/{id}	- Xem chi tiết sản phẩm
GET	/actuator/health	- Health check

b) USER/ADMIN ENDPOINTS (Cần đăng nhập)

GET	/api/v1/user/dashboard	- Dashboard user
GET	/api/v1/user/profile	- Xem profile cá nhân
PUT	/api/v1/user/profile	- Cập nhật profile
GET	/api/v1/user/profile/status	- Kiểm tra profile đầy đủ
GET	/api/v1/orders/my	- Xem đơn hàng của tôi
GET	/api/v1/orders/{id}	- Xem chi tiết đơn hàng
POST	/api/v1/orders	- Tạo đơn hàng mới
PATCH	/api/v1/orders/{id}/cancel	- Hủy đơn hàng (chỉ PENDING)

c) ADMIN-ONLY ENDPOINTS

Product Management

POST	/api/v1/products	- Tạo sản phẩm mới
PUT	/api/v1/products/{id}	- Cập nhật sản phẩm
DELETE	/api/v1/products/{id}	- Xóa sản phẩm

User Management

GET	/api/v1/admin/users	- Xem tất cả user
-----	---------------------	-------------------

PUT /api/v1/admin/users/{id}/role - Thay đổi role user

DELETE /api/v1/admin/users/{id} - Xóa user

GET /api/v1/admin/dashboard - Dashboard admin

Order Management (Admin)

GET /api/v1/admin/orders - Xem tất cả đơn hàng

GET /api/v1/admin/orders/{id} - Xem chi tiết đơn hàng

PATCH /api/v1/admin/orders/{id}/status - Cập nhật trạng thái đơn hàng

DELETE /api/v1/admin/orders/{id} - Xóa đơn hàng

3.3.3 Cấu trúc parameters/response

Dưới đây là một số ví dụ minh họa cách thức giao tiếp giữa client và server, bao gồm parameters từ phía người dùng và phản hồi trả về từ hệ thống.

a) Đăng nhập người dùng

POST /v1/auth/login Đăng nhập

Đăng nhập và nhận JWT token

Parameters

Name	Description
username * required string (query)	Tên đăng nhập
password * required string (query)	Mật khẩu

Responses

Code	Description	Links
200	Đăng nhập thành công	No links
400	Sai username hoặc password	No links

Example Value

```
{
  "status": "success",
  "message": "Login successful",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2Vybm91dCI6ImFkbWUiLCJyb2kiOiJhZG9iIn0",
  "username": "admin",
  "role": "admin"
}
```

Hình 3.2. Parameters và response khi đăng nhập thành công của admin

Authentication

API xác thực người dùng

POST

/v1/auth/register

Đăng ký tài khoản mới

Tạo tài khoản mới với role mặc định là USER

Parameters

Try it out

Name	Description
username * required string (query)	Tên đăng nhập Example : newuser <input type="text" value="newuser"/>
password * required string (query)	Mật khẩu Example : password123 <input type="text" value="password123"/>
fullName * required string (query)	Họ và tên đầy đủ Example : Nguyễn Văn A <input type="text" value="Nguyễn Văn A"/>

Responses

Code	Description	Links
200	Đăng ký thành công <div>Media type <input type="text" value="*/"/></div> <div>Controls Accept header:</div> <div>Example Value Schema</div> <div><pre>{ "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string"}</pre></div>	No links
400	Username đã tồn tại hoặc lỗi khác <div>Media type <input type="text" value="*/"/></div> <div>Example Value Schema</div> <div><pre>{ "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string"}</pre></div>	No links

Hình 3.3. Paramesters và response sau khi đăng nhập thành công phía Khách hàng

b)Đặt hàng và quản lý đơn hàng

Xây dựng Web quản lý cửa hàng bán đồ thể thao

Order Management

API đặt hàng và quản lý đơn hàng cho User

POST

/v1/orders

Tạo đơn hàng mới

User tạo đơn hàng đơn giản - chỉ cần productId và quantity. Thông tin địa chỉ tự động lấy từ profile

Parameters

Try it out

No parameters

Request body

required

application/json

Example Value

Schema

```
{
  "notes": "string",
  "items": [
    {
      "productId": 0,
      "quantity": 0
    }
  ]
}
```

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value

Schema

```
{}
```

Hình 3.4. Parameters và response khi đặt hàng và quản lý đơn hàng

c)Hủy đơn hàng

PATCH

/v1/orders/{id}/cancel

Hủy đơn hàng của tôi

User có thể hủy đơn hàng của mình (chỉ trạng thái PENDING)

Parameters

Try it out

Name	Description
id * required	
integer(\$int64)	ID của đơn hàng cần hủy
(path)	Example : 1

1

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value

Schema

```
{}
```

d) Xem chi tiết đơn hàng

GET

/v1/orders/{id} Xem chi tiết đơn hàng

⌵

User có thể xem chi tiết đơn hàng của mình

Parameters

Try it out

Name	Description
id * required	
integer(\$int64)	ID của đơn hàng
(path)	Example: 1
<input type="text" value="1"/>	

Responses

Code

Description

Links

200

OK

No links

Media type

/

Controls Accept header.

Example Value | Schema

```
{
  "id": 0,
  "orderDate": "2025-07-23T18:04:48.172Z",
  "status": "string",
  "notes": "string",
  "customer": {
    "id": 0,
    "name": "string",
    "phone": "string",
    "email": "string",
    "address": "string"
  },
  "userAccount": {
    "id": 0,
    "username": "string",
    "role": "string",
    "fullName": "string",
    "email": "string",
    "phone": "string",
    "address": "string"
  },
  "orderDetails": [
    {
      "id": 0,
      "productId": 0,
      "productName": "string",
      "productImageUrl": "string",
      "quantity": 0,

```

Hình 3.6. Parameters và response khi xem chi tiết đơn hàng

e)Xem đơn hàng

The screenshot displays a REST client interface with the following components:

- Code/Description/Links:** A table showing a 200 OK response with no links.
- Media type:** A dropdown menu set to `*/*`.
- Controls:** A button labeled "Accept header".
- Example Value | Schema:** A text area containing a JSON response structure:

```
{
  "id": 0,
  "orderDate": "2025-07-23T18:13:14.375Z",
  "status": "string",
  "notes": "string",
  "customer": {
    "id": 0,
    "name": "string",
    "phone": "string",
    "email": "string",
    "address": "string"
  },
  "userAccount": {
    "id": 0,
    "username": "string",
    "role": "string",
    "fullName": "string",
    "email": "string",
    "phone": "string",
    "address": "string"
  },
  "orderDetails": [
    {
      "id": 0,
      "productId": 0,
      "productName": "string",
      "productImageUrl": "string",

```

Request Details:

- Method:** GET
- URL:** `/v1/orders/my`
- Description:** Xem đơn hàng của tôi

User Action: User xem danh sách đơn hàng của chính mình

Parameters:

Name	Description
status	Lọc theo trạng thái (tùy chọn)
string (query)	Example : PENDING

Input: A text box containing the value "PENDING".

Responses: A section for displaying response details.

Hình 3.7. Parameters và response khi xem đơn hàng

f) Thông tin profile

GET

/v1/user/profile

Thông tin profile

PUT

/v1/user/profile

Cập nhật profile

Cập nhật thông tin cá nhân (họ tên, email, SĐT, địa chỉ)

Parameters

Try it out

Name	Description
fullName string (query)	Họ và tên
	<input type="text" value="fullName"/>
email string (query)	Email
	<input type="text" value="email"/>
phone string (query)	Số điện thoại
	<input type="text" value="phone"/>
address string (query)	Địa chỉ
	<input type="text" value="address"/>

Responses

Responses

Code	Description	Links
200	OK	No links

Media type

/

Controls Accept header.

Example Value | Schema

```
{
  "additionalProp1": {},
  "additionalProp2": {},
  "additionalProp3": {}
}
```

Hình 3.8. Paramester và response khi xem thông tin profile

g) Cập nhật trạng thái đơn hàng(chỉ admin)

PATCH /v1/admin/orders/{id}/status Cập nhật trạng thái đơn hàng

Chỉ ADMIN mới có thể cập nhật trạng thái đơn hàng. Trạng thái: PENDING, CONFIRMED, SHIPPING, DELIVERED, CANCELLED

Parameters Try it out

Name	Description
id * required integer(int64) (path)	ID của đơn hàng cần cập nhật trạng thái Example: 1

Request body required application/json

Example Value | Schema

```
{  
  "status": "string"  
}
```

Responses

Code	Description	Links
200	OK	No links

Media type: */*

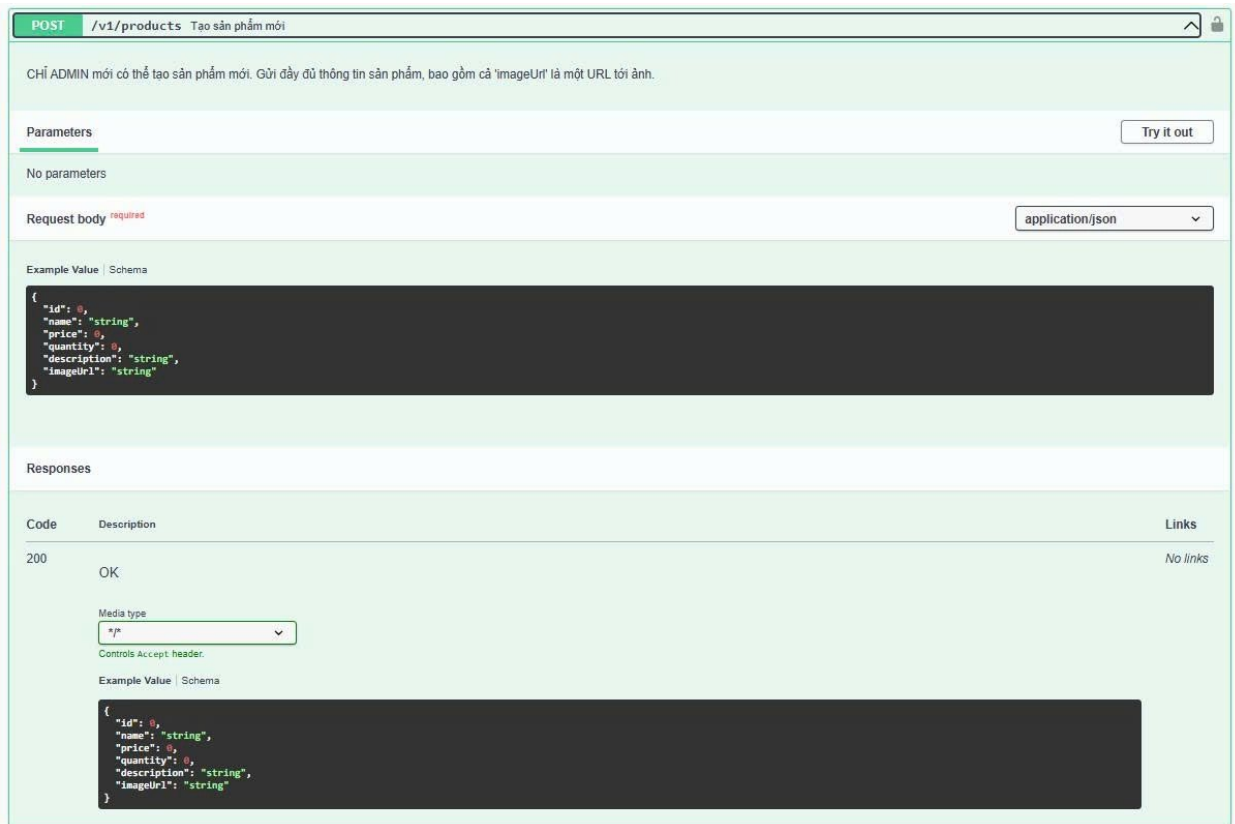
Controls Accept header.

Example Value | Schema

```
{  
  "status": "string"  
}
```

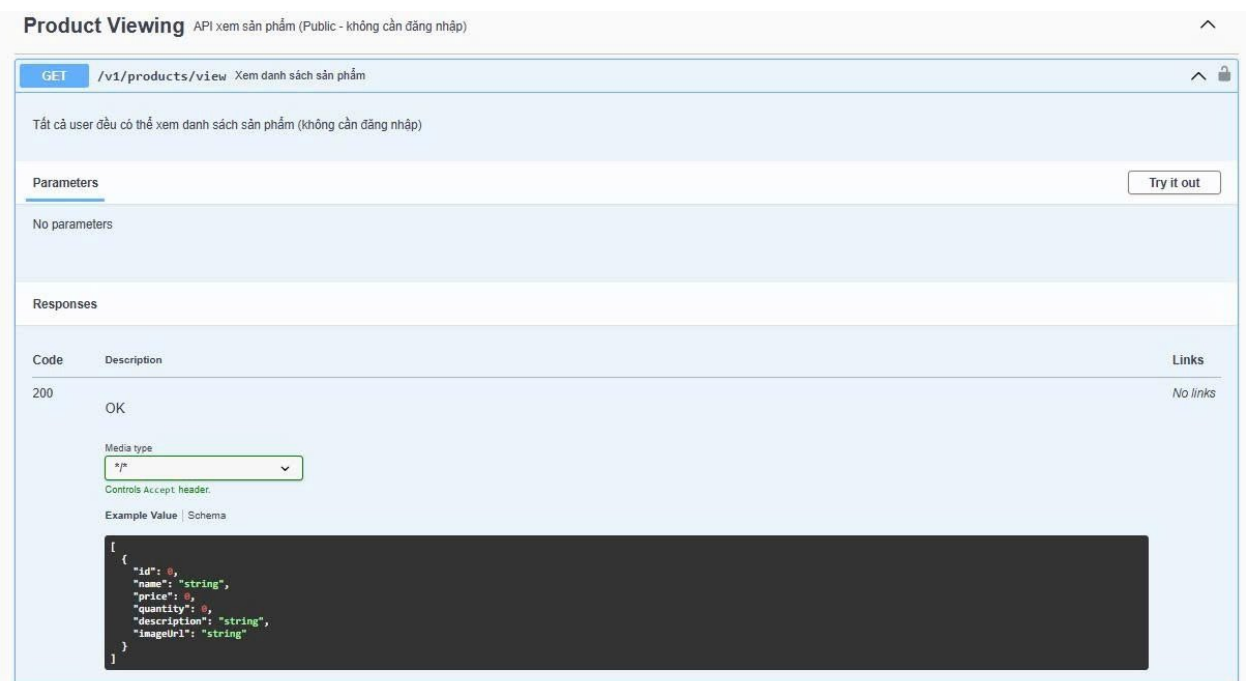
Hình 3.9. Parameters và response khi cập nhật trạng thái đơn hàng

h)Thêm sản phẩm mới(chỉ admin)



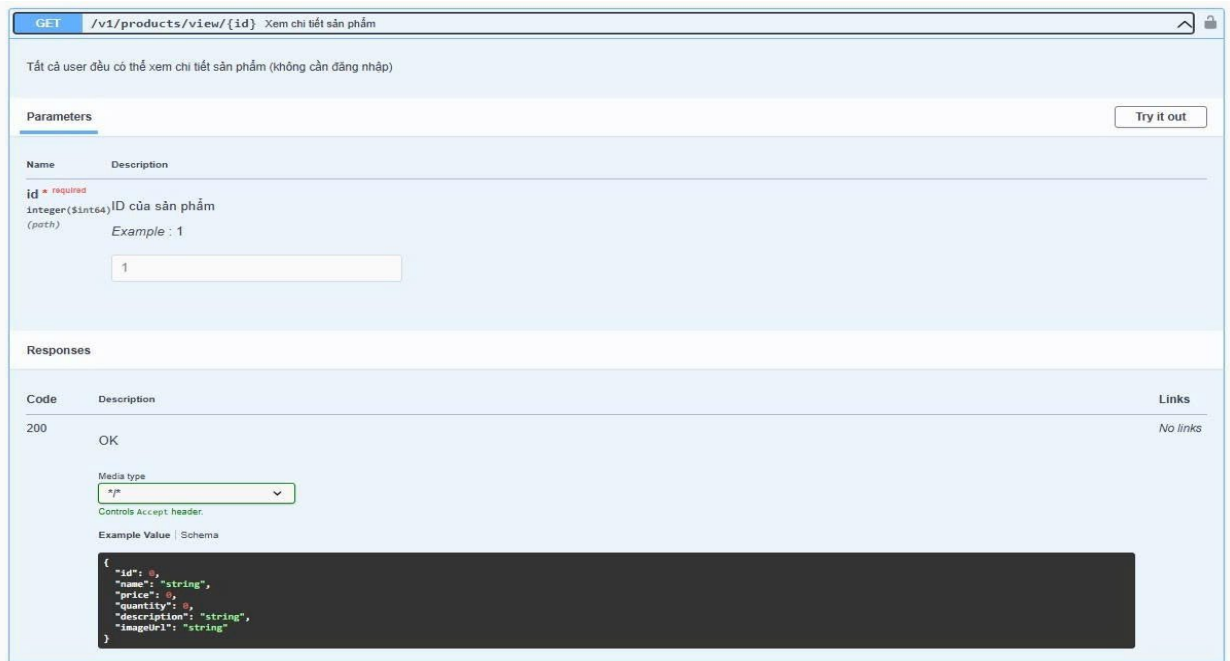
Hình 3.10. Paramester và response khi thêm sản phẩm mới

i)API xem sản phẩm(Không cần đăng nhập)



Hình 3.11. Paramester và response khi xem sản phẩm

j)Xem chi tiết sản phẩm (Không cần đăng nhập)



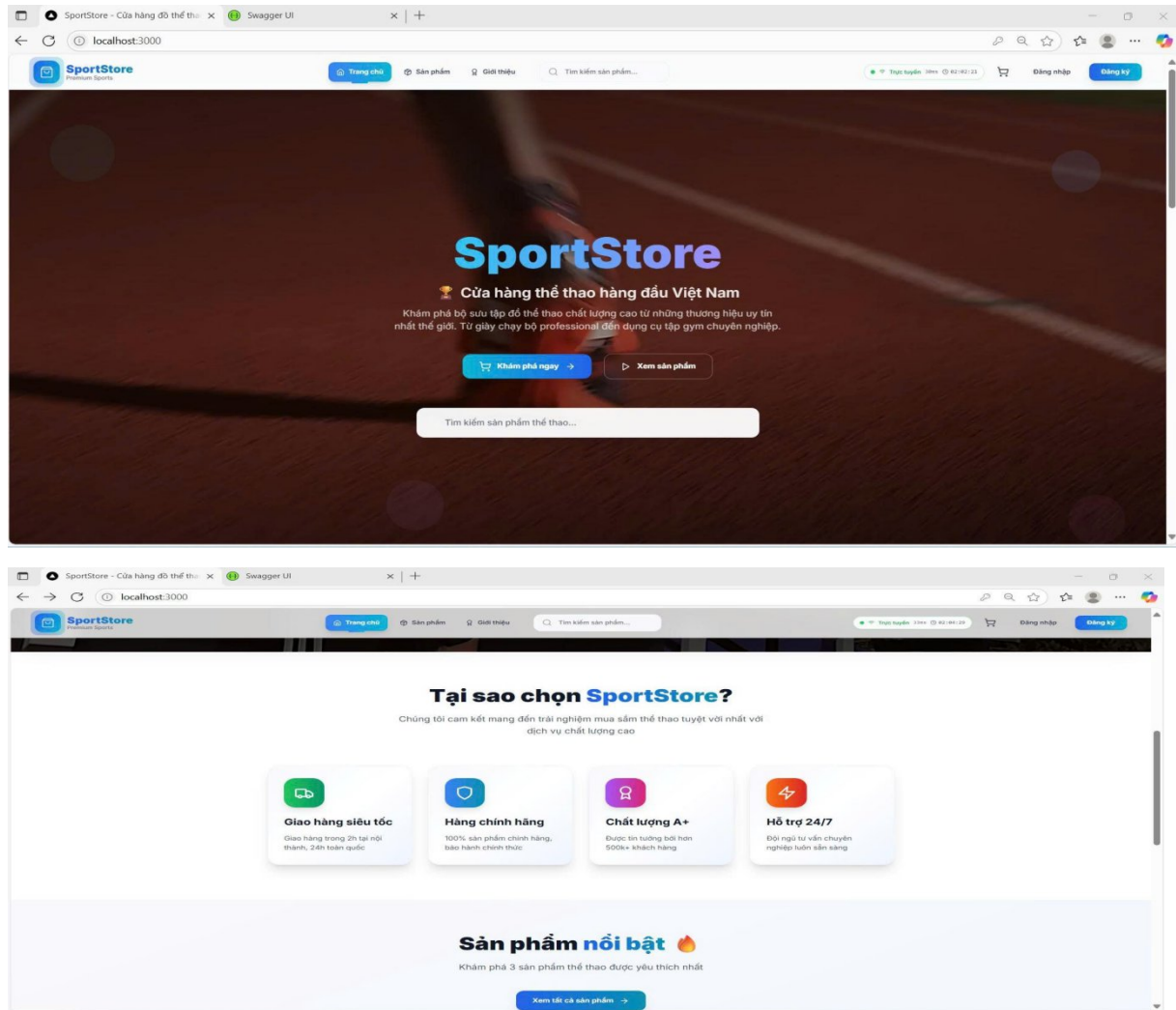
Hình 3.12. Paramester và response khi xem chi tiết sản phẩm

3.4. Thiết kế giao diện (UI/UX)

Giao diện người dùng được lên ý tưởng và thiết kế bằng Figma trước khi dùng ngôn ngữ lập trình để xây dựng, đảm bảo trực quan, dễ sử dụng và nhất quán. Các yếu tố như màu sắc, bố cục và tương tác được tối ưu cho trải nghiệm người dùng. Giao diện của trang web được thiết kế dựa trên các tiêu chí như:

- Thiết kế hiện đại, chuyên nghiệp
- Sử dụng font to – rõ ràng – dễ đọc
- Màu sắc nổi bật, dễ nhận diện thương hiệu
- Giao diện phù hợp với lĩnh vực mua bán đồ thể thao.

3.4.1. Giao diện trang chủ người dùng

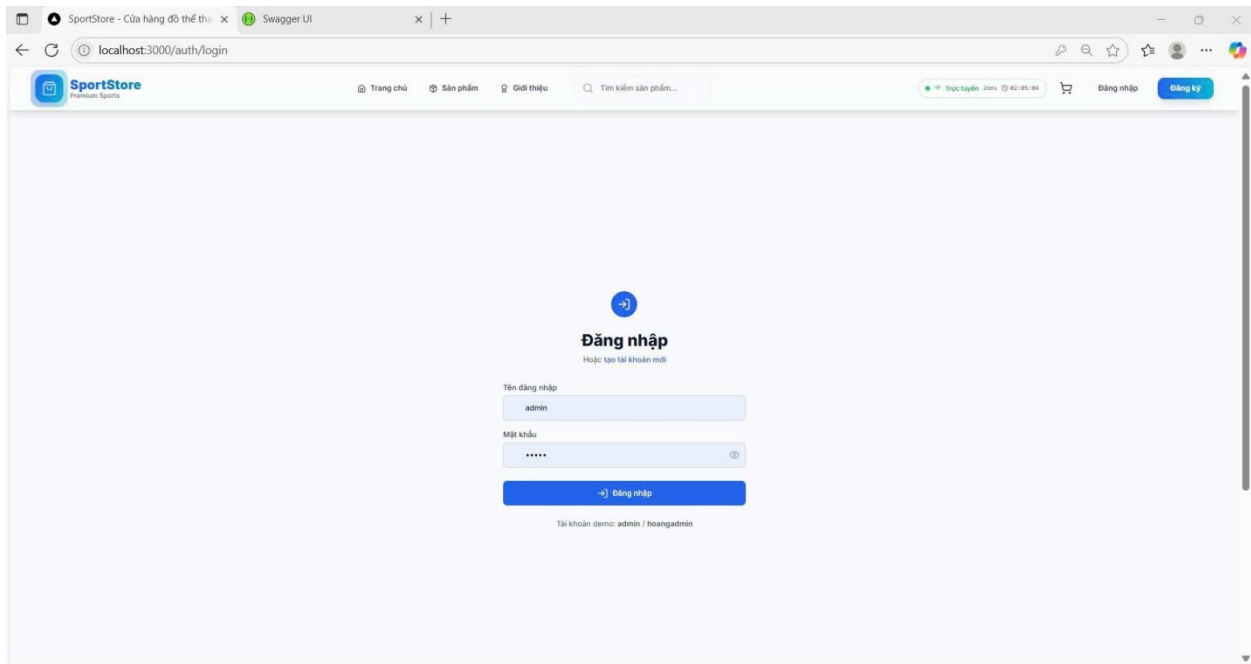


Hình 3.13 Giao diện trang chủ của người dùng

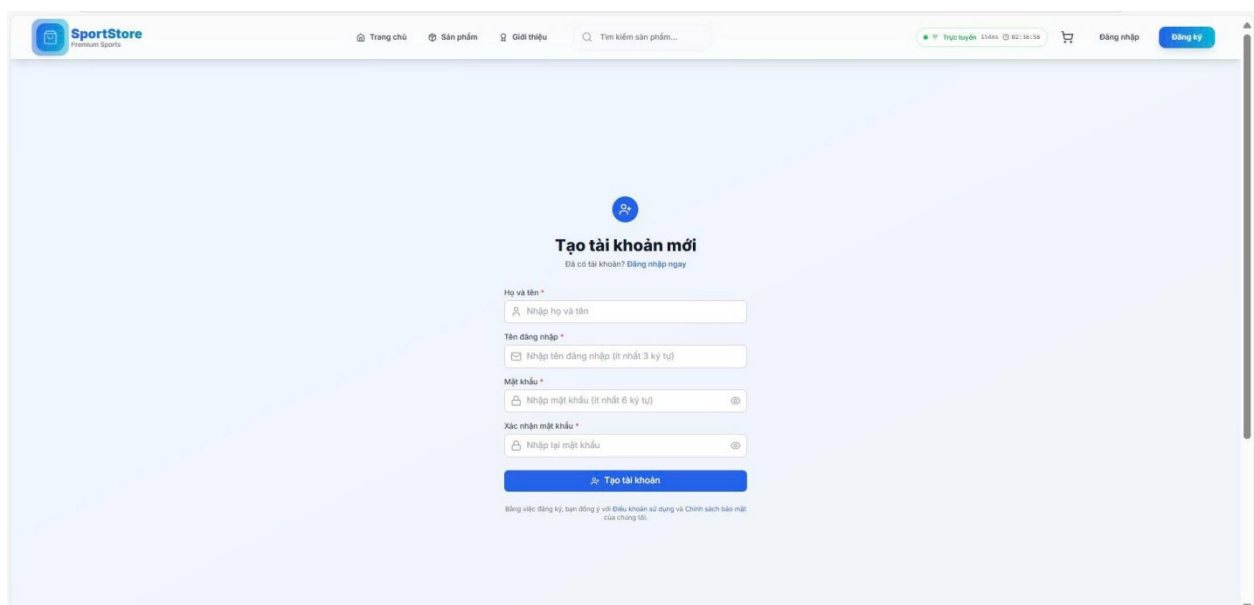
3.4.2. Giao diện đăng nhập và đăng ký người dùng

Trang đăng nhập và đăng ký được thiết kế tối giản, hiện đại, tập trung vào trải nghiệm người dùng.

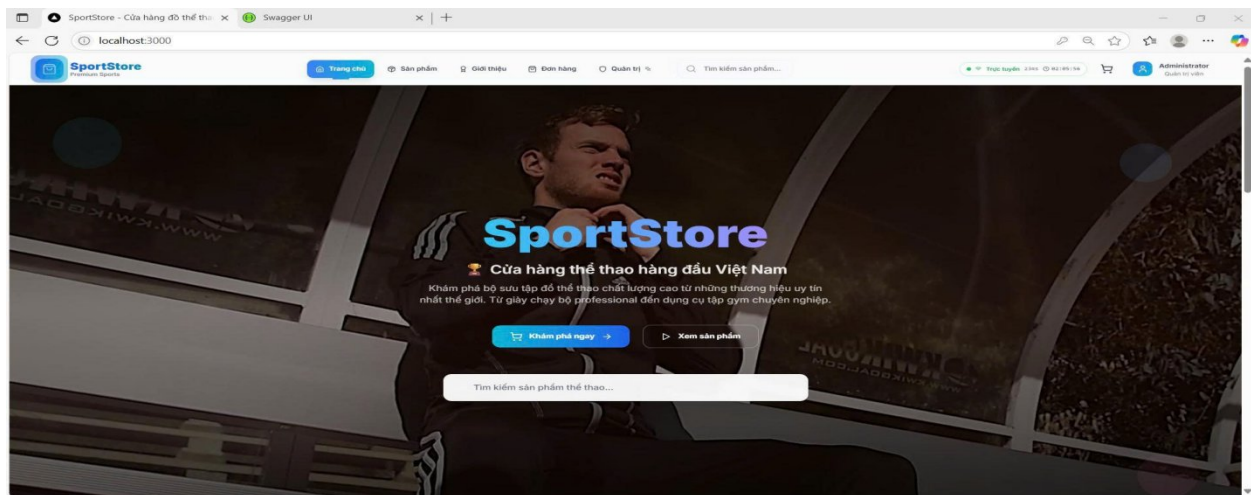
- Giao diện thống nhất cho cả **quản lý** và **Khách Hàng** → phân quyền có thể xử lý **sau khi đăng nhập**.
- Hỗ trợ người dùng mới dễ hiểu, thao tác nhanh
- Giao diện responsive – gọn gàng trên cả desktop và tablet.



Hình 3.14 Giao diện Đăng Nhập Admin và khách hàng



Hình 3.15. Giao diện đăng ký tài khoản người dùng



Hình 3.16 Giao diện sau khi đăng nhập thành công

◆ Khách hàng

- **Giao diện thân thiện:** Dễ sử dụng với bố cục rõ ràng, các danh mục sản phẩm được phân loại.
- **Tìm kiếm nhanh chóng:** Có thanh tìm kiếm và bộ lọc theo giá, loại sản phẩm, thương hiệu.
- **Chi tiết sản phẩm rõ ràng:** Mỗi sản phẩm hiển thị giá, mô tả ngắn, tình trạng còn hàng và nút “Thêm vào giỏ”.
- **Thanh toán đơn giản:** Mua hàng và thanh toán trực tuyến thuận tiện
- **Đăng nhập/Đăng ký dễ dàng:** Chỉ cần vài bước để tạo tài khoản và đăng nhập.

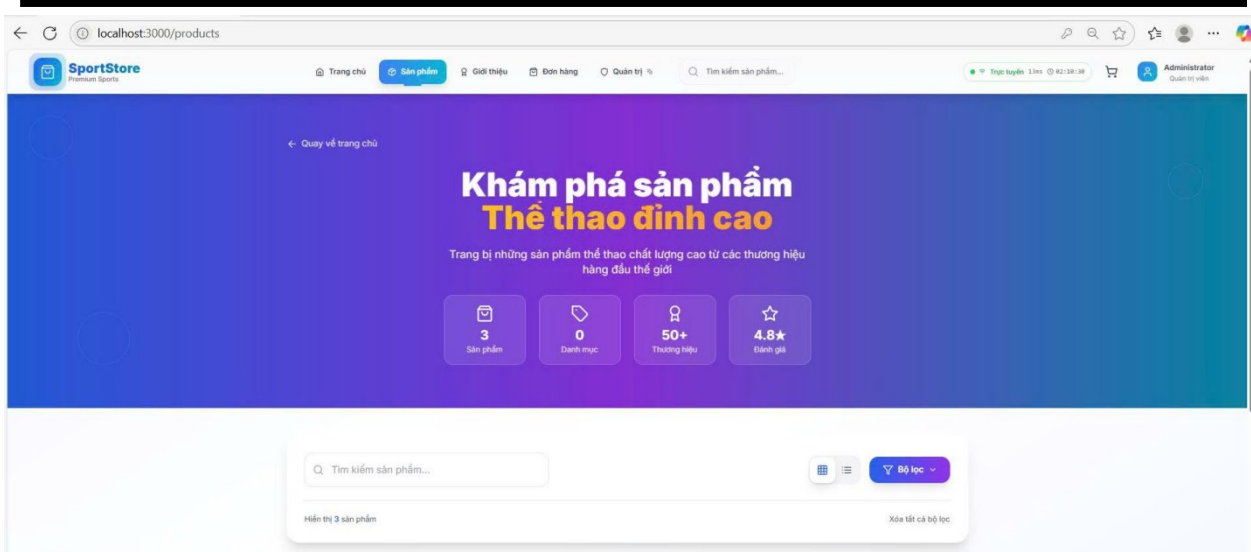
◆ Quản trị viên

- **Đăng nhập bằng tài khoản nội bộ.**
- **Quản lý sản phẩm:** Thêm, sửa, xóa, kiểm tra tồn kho.
- **Hỗ trợ khách hàng:** Xử lý đơn hàng, phản hồi thông tin.
- **Quản lý toàn bộ hệ thống:** Sản phẩm, danh mục, người dùng, đơn hàng.
- **Xem thống kê:** Số lượng sản phẩm bán ra, lượng hàng tồn kho, hiệu suất bán hàng.

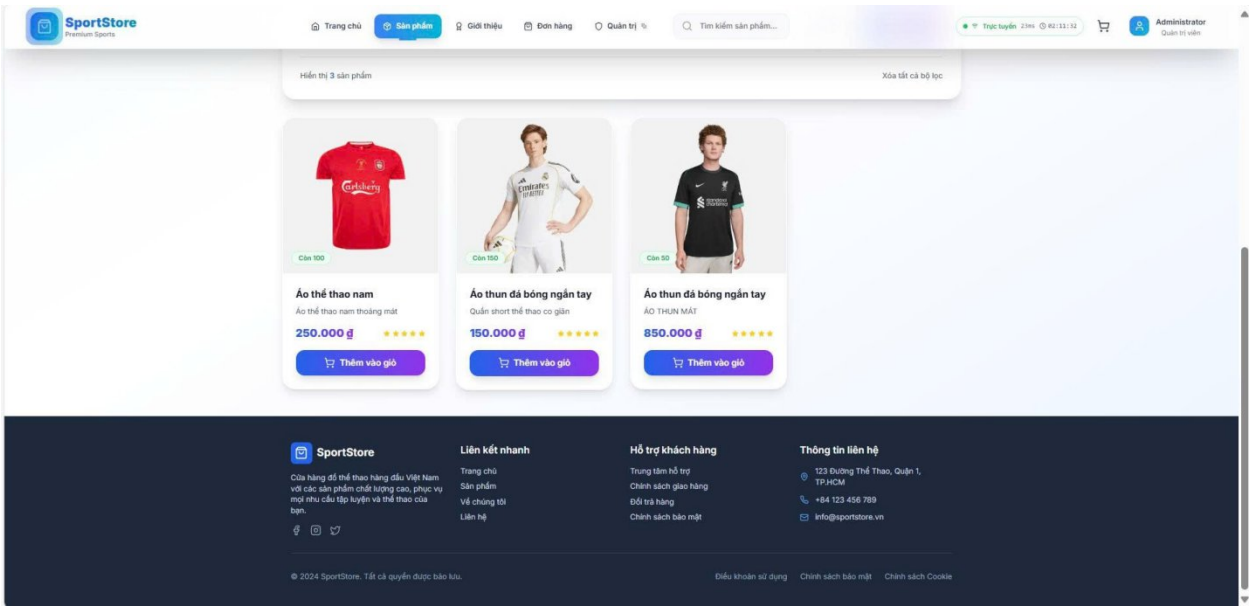
3.4.3. Giao diện sản phẩm

Các sản phẩm sẽ được hiển thị đầy đủ các thành phần thông tin như: Tên sản phẩm, hình ảnh sản phẩm, mô tả, số lượng, đánh giá, giá sản phẩm, và giỏ hàng.

Xây dựng Web quản lý cửa hàng bán đồ thể thao



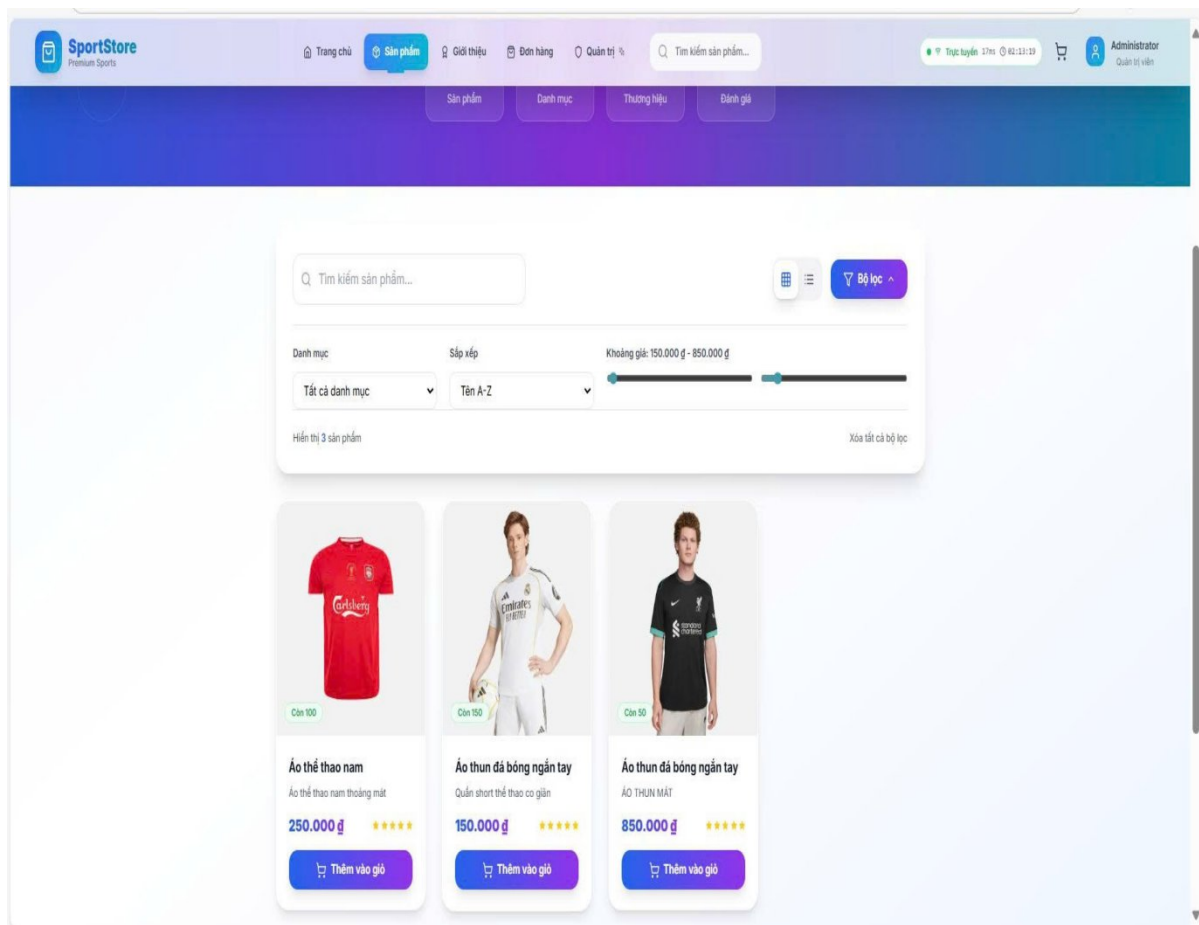
Hình 3.17 Giao diện phía trên Sản Phẩm



Hình 3.18 Giao diện phía dưới Sản Phẩm

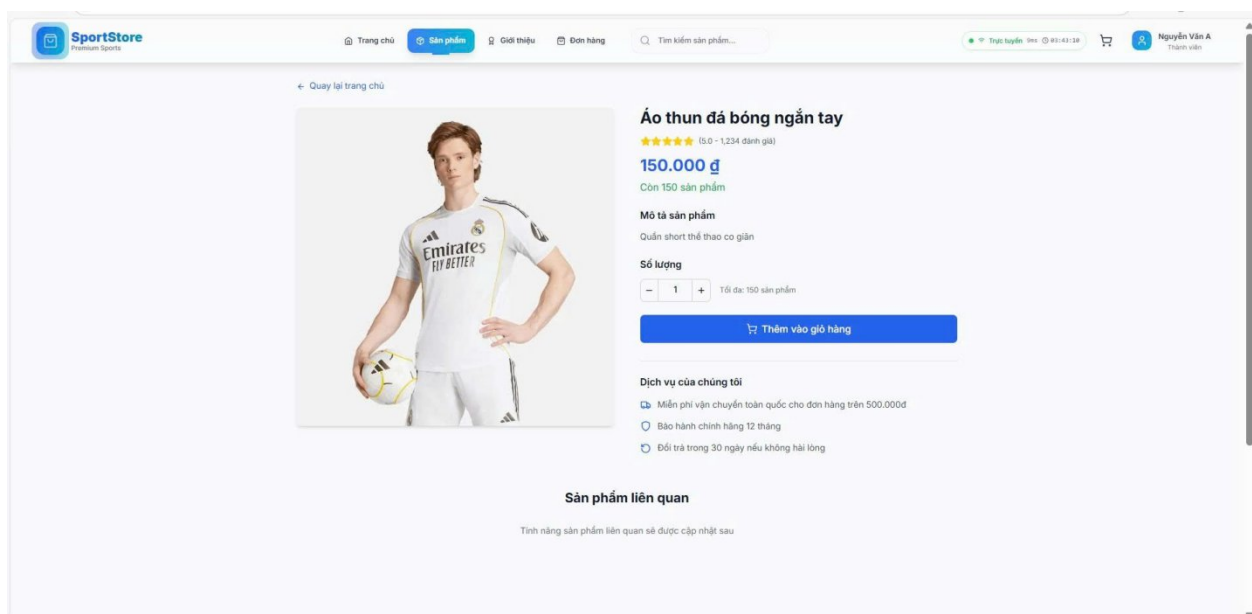
Giao diện sản phẩm có chức năng tìm kiếm và lọc sản phẩm, giúp người dùng tìm kiếm sản phẩm một cách nhanh chóng.

Xây dựng Web quản lý cửa hàng bán đồ thể thao



Hình 3.19. Giao diện tìm kiếm và lọc sản phẩm

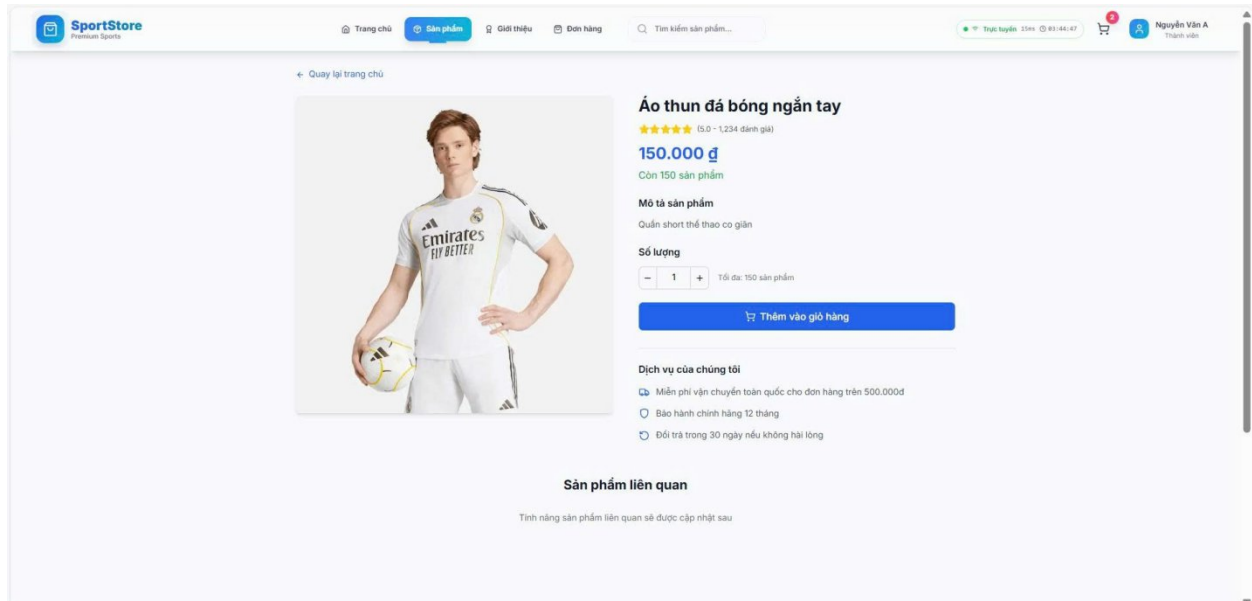
Khi người dùng muốn xem chi tiết một sản phẩm, thì người dùng sẽ nhấn vào một sản phẩm để xem, thông tin chi tiết được thêm vào như: tùy chọn số lượng đặt hàng, số lượt đánh giá, số sản phẩm còn lại trong kho, và các dịch vụ của hàng



Hình 3.20. Giao diện chi tiết sản phẩm khi nhấp vào sản phẩm

3.4.4. Giao diện giỏ hàng và thanh toán

Khi người dùng nhấn vào nút thêm giỏ hàng ở giao diện sản phẩm, giỏ hàng ở góc phải sẽ hiện thị số sản phẩm sau khi thêm.

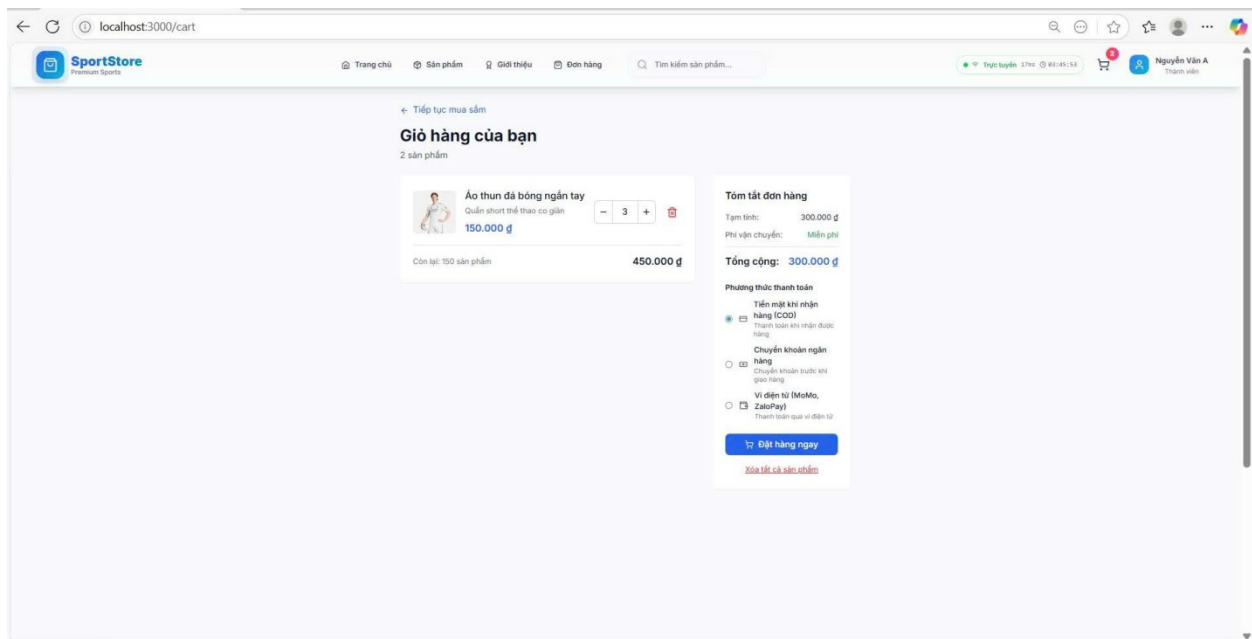


Hình 3.21. Giao diện khi người dùng nhấp vào thêm giỏ hàng(nút giỏ hàng góc trên bên phải sẽ hiển thị số sản phẩm sau khi nhấp vào thêm giỏ hàng)

Khi người dùng nhấp vào giỏ hàng góc phải và sẽ được đưa tới trang giỏ hàng, trang giỏ hàng sẽ hiển thị thông tin đặt hàng như:

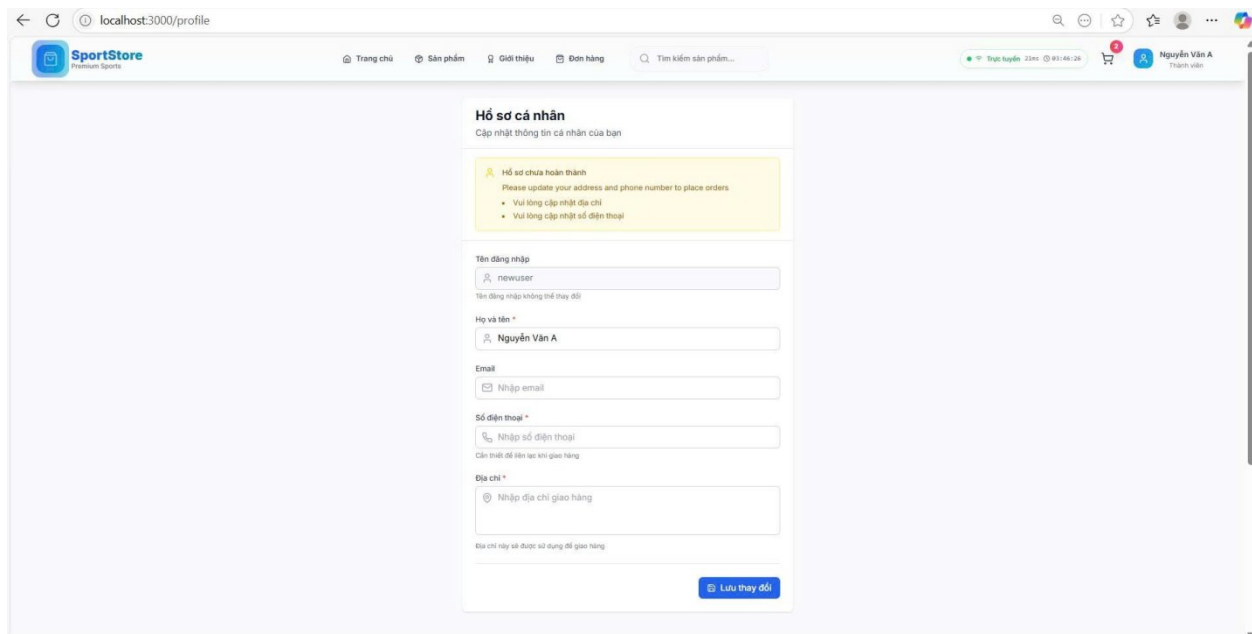
- Thông tin sản phẩm
- Số lượng mua
- Số lượng sản phẩm còn lại trong kho
- Giá sản phẩm tương ứng với số lượng mua
- Phí vận chuyển
- Tổng đơn hàng
- Các Phương thức thanh toán để người dùng có thể chọn như:
 - + Thanh toán tiền mặt khi nhận hàng
 - + Chuyển khoản bằng ngân hàng trước khi giao hàng
 - + Thanh toán qua ví điện tử(MoMo, ZaloPay)

Xây dựng Web quản lý cửa hàng bán đồ thể thao

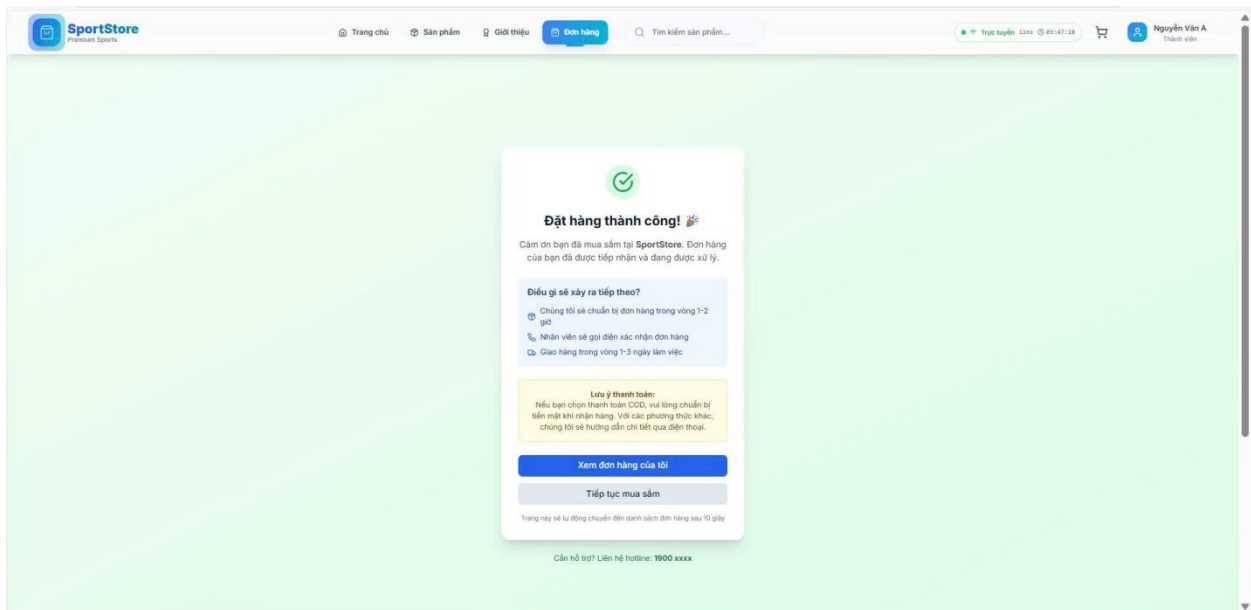


Hình 3.22. Giao diện giỏ hàng khi người dùng nhấp vào giỏ hàng góc phải màn hình.

Khi người dùng nhấn vào nút đặt hàng ngay, hệ thống sẽ thông báo đặt hàng thành công. Khi người dùng chưa nhập thông tin cá nhân ở phần User, thì hệ thống sẽ yêu cầu nhập thông tin cá nhân, sau đó mới cho phép đặt hàng.



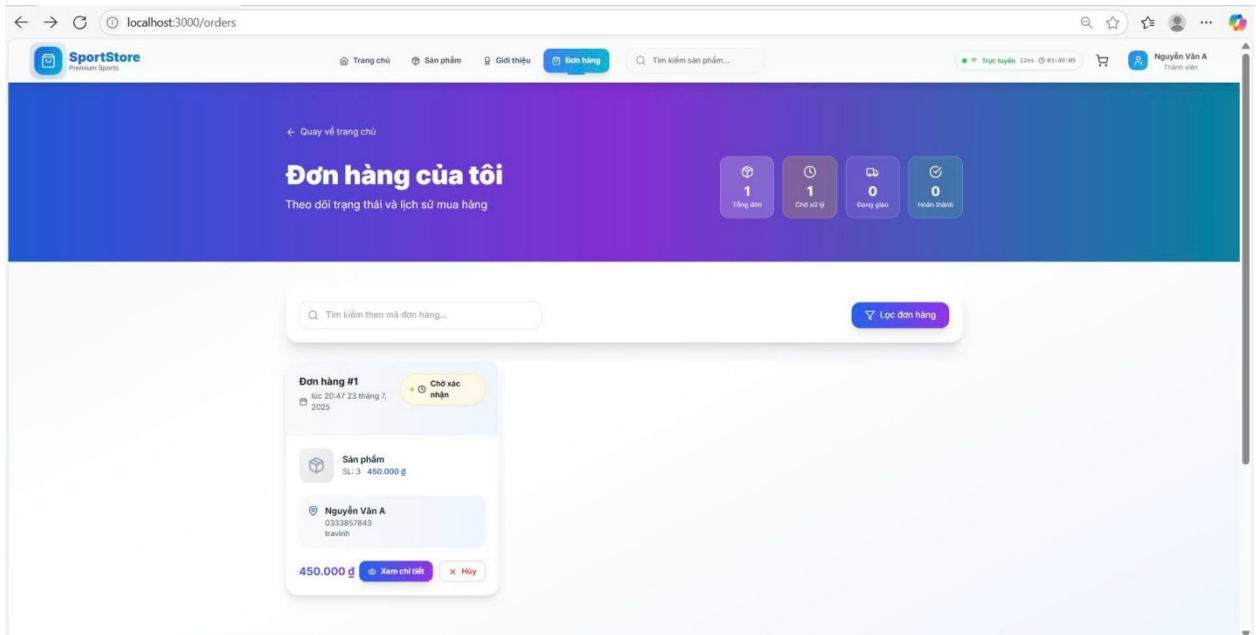
Hình 3.23. Khi người dùng nhấn vào nút đặt hàng ngay (Khi chưa nhập thông tin cá nhân)



Hình 3.24. Giao diện khi người dùng đặt hàng thành công

3.4.5. Giao diện đơn hàng người dùng

Giao diện đơn hàng của người dùng sẽ hiển thị đơn đặt hàng của khách hàng sau khi mua hàng thành công, đơn hàng sẽ hiển thị đầy đủ các thông tin như: mã đơn hàng, số lượng, trạng thái đơn hàng, sản phẩm mua, giá tiền, thông tin khách hàng, xem chi tiết đơn hàng, và hủy đơn hàng.

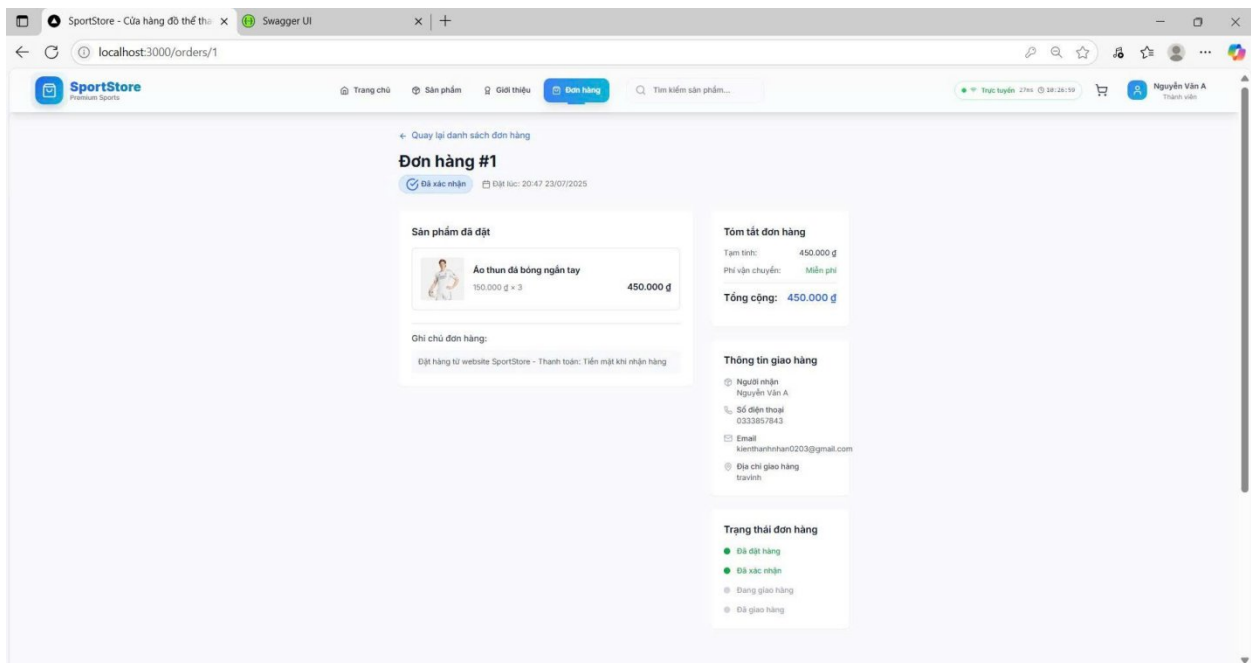


Hình 3.25. Giao diện Đơn Hàng người dùng

Khi người dùng muốn xem chi tiết một đơn hàng, người dùng có thể nhấp vào xem chi tiết

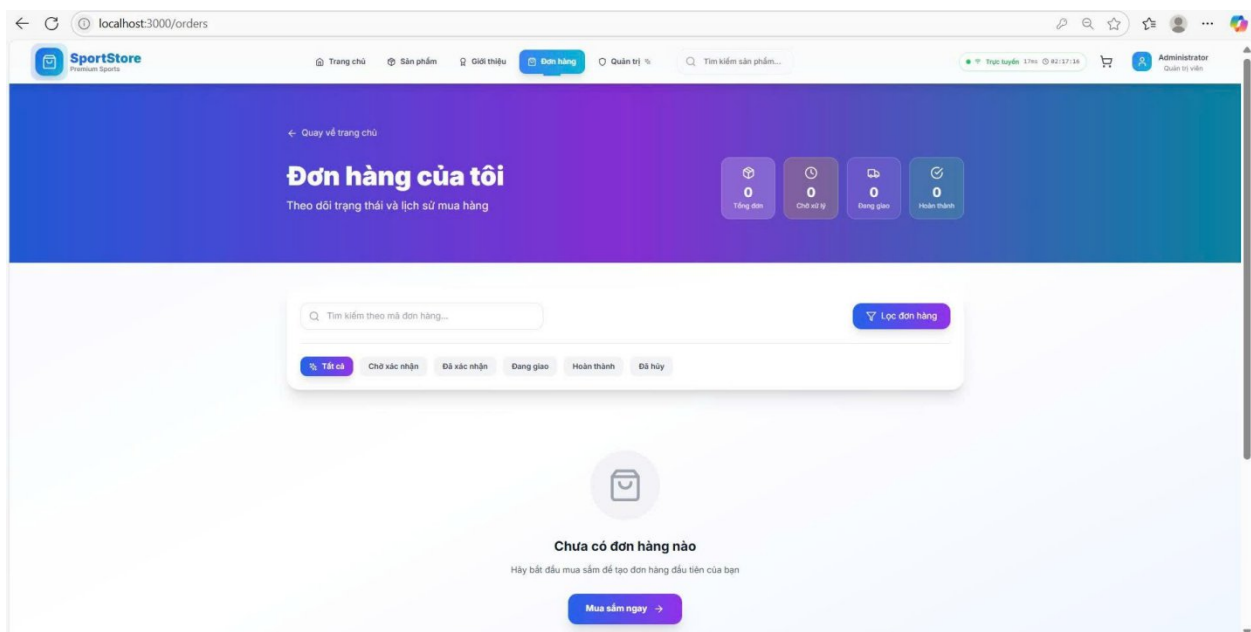
Xây dựng Web quản lý cửa hàng bán đồ thể thao

để xem, thông tin chi tiết được thêm vào như: thông tin chi tiết của khách hàng, ghi chú đơn hàng, trạng thái của đơn hàng, phí vận chuyển.



Hình 3.26. Giao diện chi tiết đơn hàng người dùng

Giao diện đơn hàng còn có chức năng tìm kiếm và lọc đơn hàng, cho phép người dùng tìm kiếm đơn hàng nhanh nhất.



Hình 3.27. Giao diện lọc Đơn Hàng

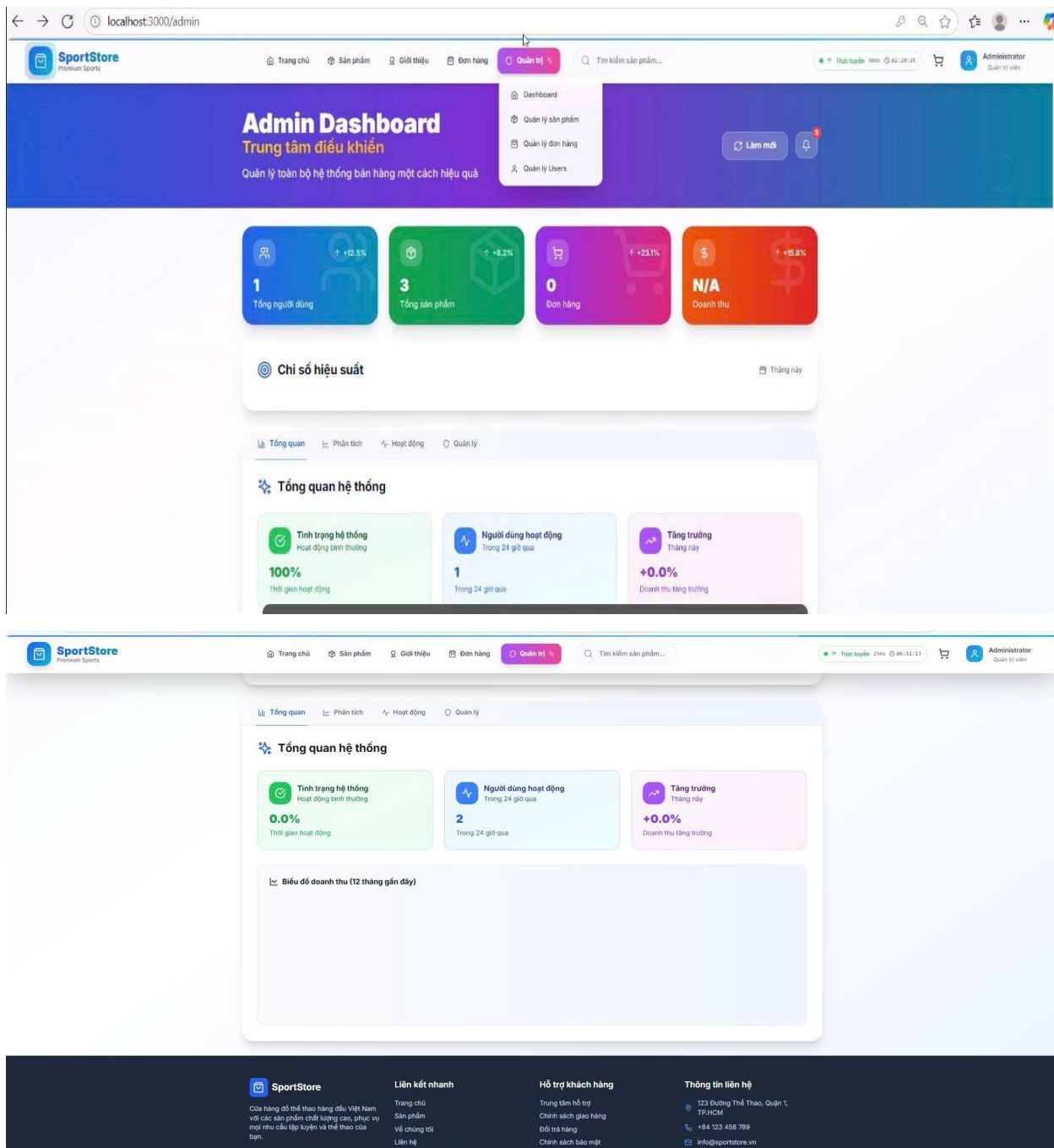
3.4.6. Giao diện quản trị hệ thống của Admin

Giao diện quản trị hệ thống của admin sẽ là trang trung tâm điều khiển của hệ thống,

Xây dựng Web quản lý cửa hàng bán đồ thể thao

admin sẽ quản lý toàn bộ như: User, sản phẩm, đơn hàng.

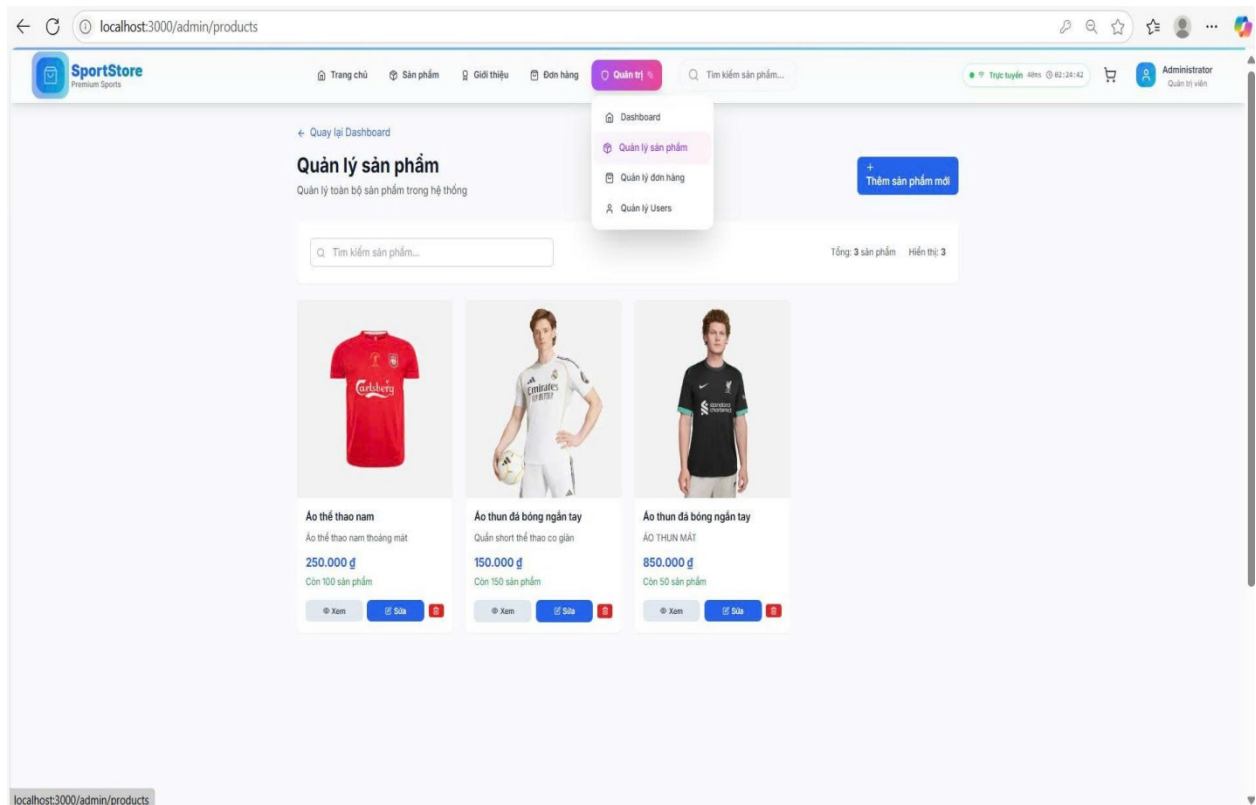
Giao diện cho phép admin xem hiệu suất, tình trạng và doanh thu sản phẩm



Hình 3.28. Giao diện các thành phần trong Quản lý của Admin

Chức năng quản lý sản phẩm của Admin trong phần quản trị sẽ gồm các chức năng thêm, sửa, xóa, tìm kiếm sản phẩm, xem chi tiết sản phẩm, số lượng sản phẩm hiển thị.

Xây dựng Web quản lý cửa hàng bán đồ thể thao



Hình 3.29. Giao diện Quản lý sản phẩm của Admin

Giao diện chức năng thêm sản phẩm mới vào cửa hàng của admin sẽ gồm các thông tin sản phẩm cần nhập như:

- Tên sản phẩm
- Giá sản phẩm
- Số lượng sản phẩm
- URL hình ảnh sản phẩm
- Mô tả sản phẩm

Ngoài ra còn cho phép admin xem trước sản phẩm để kiểm tra lại khi thêm sản phẩm vào cửa hàng.

Xây dựng Web quản lý cửa hàng bán đồ thể thao

The screenshot shows the 'Thêm sản phẩm mới' (Add new product) form in the SportStore admin interface. The form is divided into two main sections: 'Xem trước sản phẩm' (Product preview) and 'Thông tin sản phẩm' (Product information).

Header: The top navigation bar includes links for 'Trang chủ' (Home), 'Sản phẩm' (Products), 'Giới thiệu' (Introduction), 'Đơn hàng' (Orders), and 'Quản trị' (Admin). A search bar is also present. The user is logged in as 'Administrator'.

Form Fields:

- Tên sản phẩm *** (Product name): A text input field with the example value 'VD: Giày chạy bộ Nike Air Max'.
- Giá (VNĐ) *** (Price): A text input field with the value '0'.
- Số lượng *** (Quantity): A text input field with the value '0'.
- URL ảnh sản phẩm** (Product image URL): A text input field with the value 'https://example.com/product-image.jpg'. A note below states: 'Để trống nếu không có ảnh (sẽ sử dụng ảnh mặc định)' (Leave empty if no image (will use default image)).
- Mô tả sản phẩm *** (Product description): A text area with the example text: 'Giày chạy bộ cao cấp với công nghệ đệm Air Max, phù hợp cho việc tập luyện thể thao và chạy bộ hàng ngày...' (High-end running shoes with Air Max cushioning technology, suitable for daily sports and running practice...).

Buttons: At the bottom of the form are two buttons: 'Hủy bỏ' (Cancel) and '+ Tạo sản phẩm' (Create product).

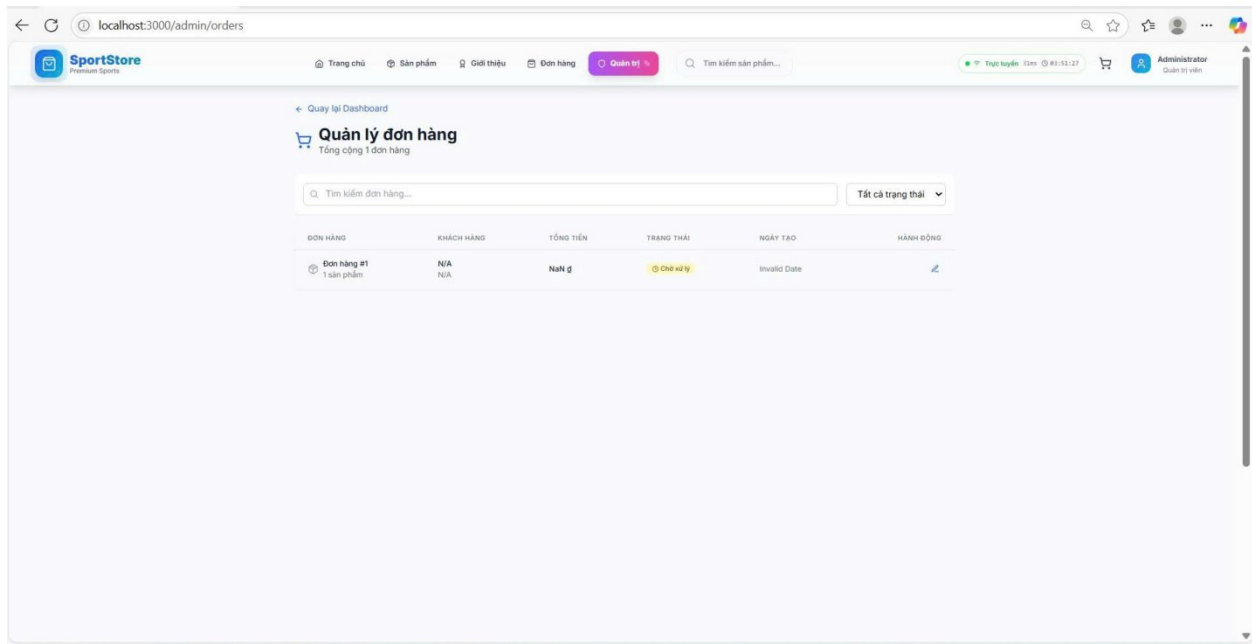
Preview Section: The 'Xem trước sản phẩm' section shows a placeholder for the product image and the product name 'Tên sản phẩm'. Below it, a note says 'Mô tả sản phẩm sẽ hiển thị ở đây...' (Product description will be displayed here...). There are also buttons for '0 đ' and 'Hết hàng' (Out of stock).

Hình 3.30. Giao diện thêm sản phẩm mới vào cửa hàng của Admin.

Giao diện chức năng quản lý đơn hàng cho phép Admin xem đơn hàng của khách hàng sau khi đặt hàng thành công. Admin có thể xem cập nhật và chỉnh sửa các thành phần trong quản lý đơn hàng của Admin như:

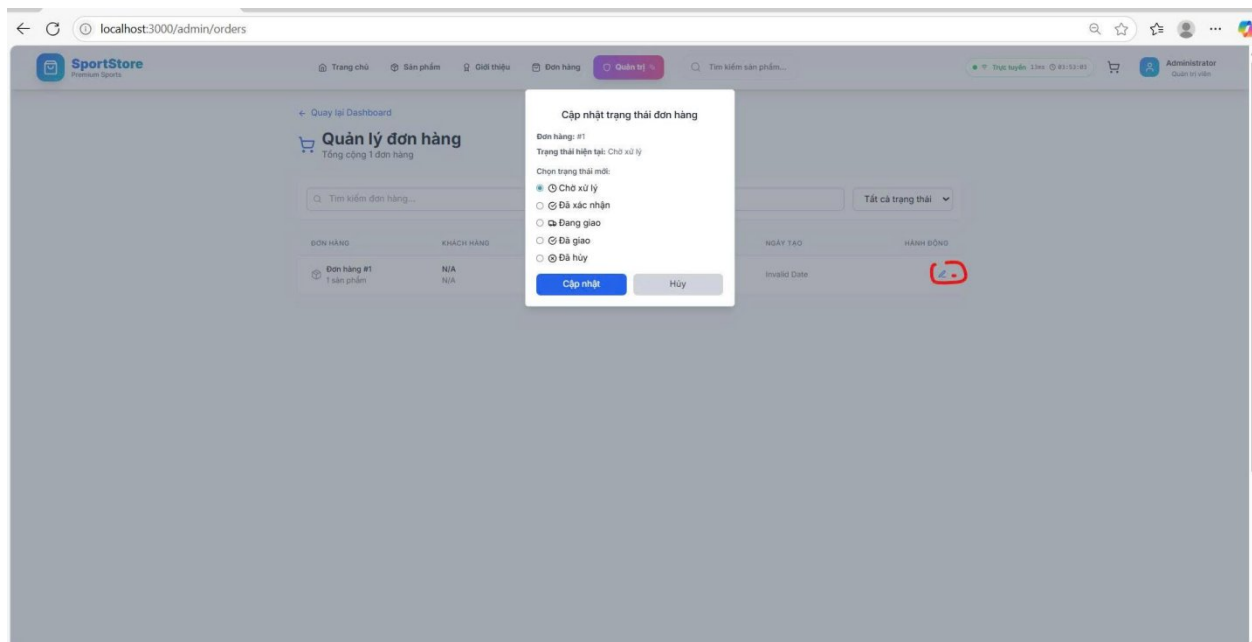
- Tổng Số lượng đơn hàng
- Mã đơn hàng
- Khách hàng
- Trạng thái đơn hàng
- Ngày tạo
- Hành động

Xây dựng Web quản lý cửa hàng bán đồ thể thao



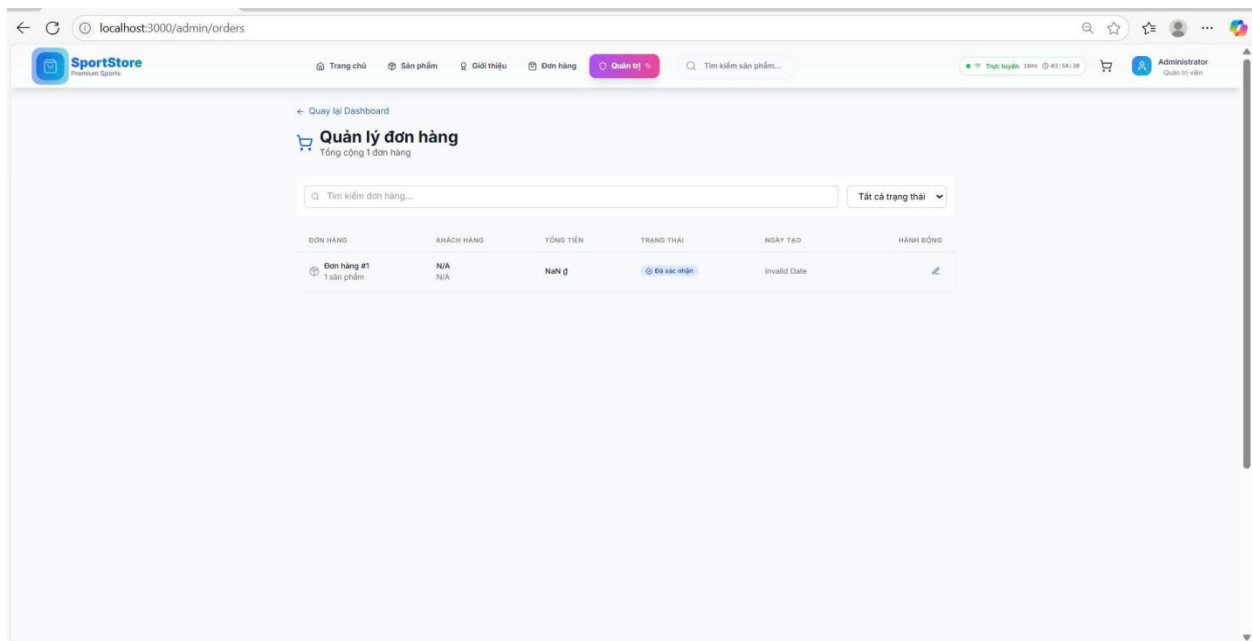
Hình 3.31. Giao diện Quản lý Đơn Hàng khi khách hàng đặt hàng thành công của Admin

Khi người dùng đặt hàng, đơn hàng sẽ được Admin liên tục cập nhật trạng thái đơn hàng để khách hàng tiện theo dõi như: Chờ xử lý, đã xác nhận, đang giao, đã giao, đã hủy



Hình 3.32. Giao diện cập nhật trạng thái đơn hàng của Admin

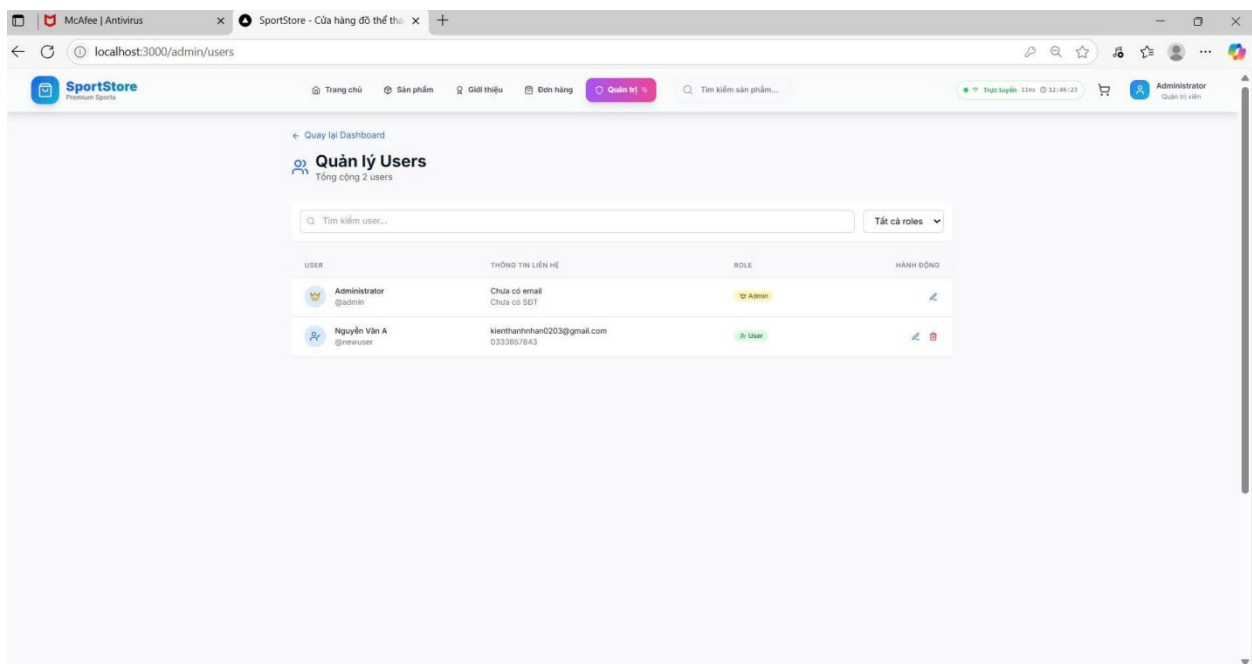
Xây dựng Web quản lý cửa hàng bán đồ thể thao



Hình 3.33. Giao diện sau khi cập nhật lại trạng thái đơn hàng của Admin

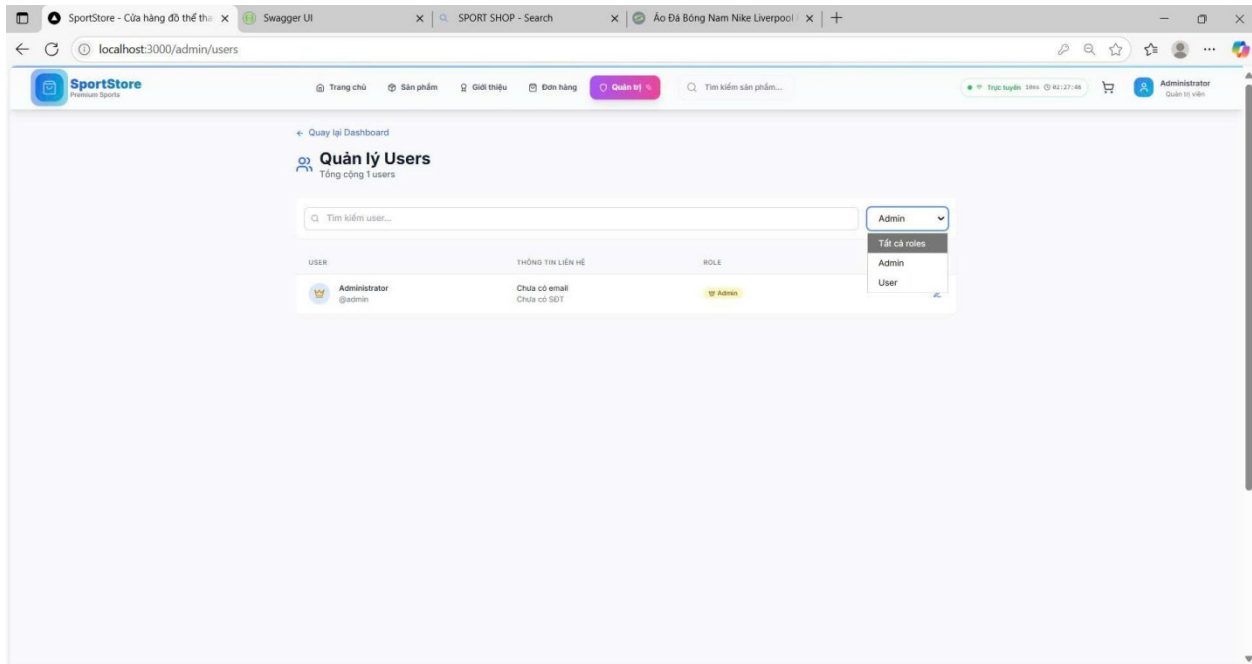
Giao diện chức năng quản lý người dùng của Admin, cho phép Admin xem các tài khoản người dùng, và quyền người dùng, các thành phần chính bao gồm:

- Tên người dùng và tài khoản
- Thông tin liên hệ
- Quyền
- Phân quyền



Hình 3.34. Giao diện Quản lý Users của Admin

Giao diện quản lý User của Admin cho phép tìm kiếm và lọc người dùng, giúp Admin nhanh chóng kiểm soát tài khoản người dùng

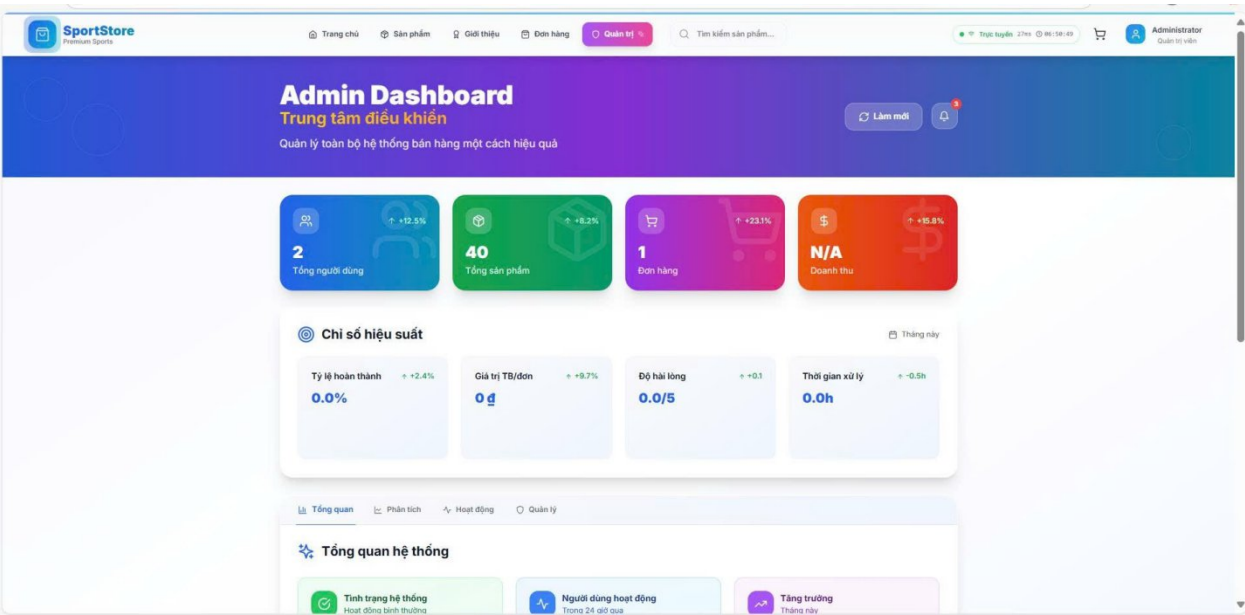


Hình 3.35. Giao diện tìm kiếm và bộ lọc Users của Admin

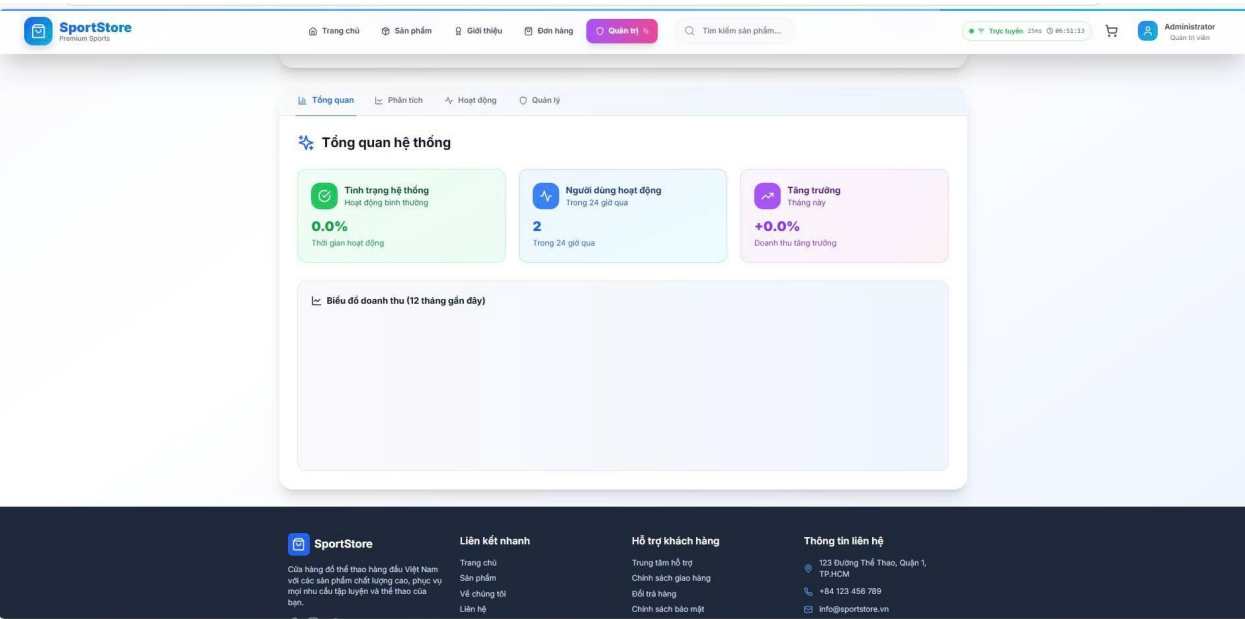
Chức năng báo cáo và thống kê giúp Admin có thể:

- Nắm bắt được số lượng đơn hàng mà khách hàng đã thanh toán thành công và doanh thu tất cả các đơn hàng.
- Xem tất cả các người dùng đã đang nhập vào hệ thống cửa hàng.
- Xem tổng sản phẩm đang bán tại cửa hàng.
- Xem thống kê doanh thu tăng trưởng, thống kê hoạt động người dùng , và thời gian hoạt động.

Xây dựng Web quản lý cửa hàng bán đồ thể thao



Hình 3.36. Giao diện báo cáo về số đơn hàng đã đặt và doanh thu



Hình 3.37. Giao diện thống kê tăng trưởng, người dùng và thời gian hoạt động

CHƯƠNG 4: TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG

4.1. Ngôn ngữ lập trình và framework

- Frontend:

+ **NextJS (React framework)**: hỗ trợ rendering phía server và static generation, giúp tối ưu SEO và tốc độ tải trang.

+ **TailwindCSS**: framework CSS utility-first giúp thiết kế giao diện nhanh chóng, responsive.

- Backend:

+ **NodeJS + ExpressJS**: xây dựng API REST, xử lý nghiệp vụ backend.

+ **Sequelize ORM**: giản lược truy vấn SQL, hỗ trợ mapping giữa object và bảng dữ liệu.

- Database:

+ **MySQL**: hệ quan hệ lựa chọn vì ổn định, hiệu năng cao, dễ deploy trên Docker.

4.2. Thiết lập CI/CD với GitHub Actions

- CI (Continuous Integration):

+ Mỗi khi push code lên GitHub, hệ thống tự động kiểm tra chất lượng code (lint), chạy các unit test, và build frontend/backend.

+ Tránh lỗi build thủ công, tiết kiệm thời gian.

- CD (Continuous Deployment):

+ GitHub Actions tích hợp cùng Docker để build image và deploy tự động.

+ Có thể triển khai lên máy chủ local, server cloud (Heroku, Render, VPS).

+ Quá trình CD bao gồm kiểm thử, đóng gói và xuất bản bản release.

4.3. Docker hoá hệ thống

- Docker-compose.yml định nghĩa toàn bộ hệ thống gồm:

+ frontend - NextJS app.

+ backend - ExpressJS API.

+ mysql - Cơ sở dữ liệu.

- Dễ dàng cài đặt bằng lệnh docker-compose up -d.

- Docker giúp đồng nhất môi trường giữa các thành viên nhóm.

4.4. Công cụ và dịch vụ hỗ trợ

- **Postman**: Kiểm thử API, quản lý token, tự động hoá test.

- **Swagger (OpenAPI):** Tạo tài liệu API trực quan, hỗ trợ team frontend hiểu và dùng endpoint chính xác.
- **Docker:** Đóng gói môi trường làm việc giúp triển khai nhanh.
- **Jira:** Lập kế hoạch sprint, phân chia công việc, tracking bug.
- **Figma:** Thiết kế UI/UX, tạo prototype trước khi code thật.
- **GitHub:** Quản lý version code, pull request, branch cho từng chức năng riêng biệt

4.5. Môi trường triển khai

- **Localhost:**
 - + Sử dụng Visual Studio Code kết hợp Docker Desktop.
 - + Dùng terminal để chạy **cd Backend** (Backend), **docker compose up --build, cd Frontend, npm run dev**(Frontend)
- **Môi trường staging:**
 - + Có thể triển khai thử nghiệm bằng Render hoặc Railway trước khi chính thức đẩy production.
- **Môi trường production:**
 - + Sử dụng VPS Linux với cấu hình tối thiểu 2GB RAM, Docker Engine.
 - + Build frontend thành static file và đặt tại **nginx**, backend chạy trên port riêng thông qua reverse proxy.

4.6. Đánh giá quy trình triển khai

- CI/CD tự động giúp rút ngắn thời gian phát hành.
- Docker loại bỏ lỗi do khác biệt môi trường cài đặt.
- Việc chia nhỏ hệ thống theo hướng microservices giúp dễ bảo trì và nâng cấp.
- Công cụ như Swagger và Postman giúp đảm bảo giao tiếp frontend - backend chính xác và hiệu quả.

CHƯƠNG 5: QUẢN LÝ DỰ ÁN

5.1. Phương pháp luận phát triển phần mềm

Dự án được phát triển theo mô hình Agile Scrum, với các ưu điểm nổi bật:

- Cho phép chia nhỏ dự án thành các giai đoạn (Sprint), dễ theo dõi tiến độ.
- Phù hợp với nhóm sinh viên vì tính linh hoạt, phản hồi nhanh với thay đổi.
- Mỗi sprint kéo dài 1–2 tuần, kết thúc với phiên demo để đánh giá kết quả.

5.2. Sử dụng Jira để lập kế hoạch và theo dõi tiến độ

Nhóm đã sử dụng Jira để quản lý công việc, lập kế hoạch và theo dõi tiến độ. Dự án được chia thành 4 giai đoạn từ 10/6/2025 đến 24/07/2025, với các story lớn và các task đã được định nghĩa. Dưới đây là kế hoạch chi tiết cho từng sprint:

5.2.1.Sprint 1

- Thời gian: 10/06/2025 đến 30/6/2025.
- Kế hoạch:



Task ID	Task Description	Status	Assignee
SCRUM-51	Tạo giao diện đăng nhập	DONE	ĐC
SCRUM-52	Viết code đăng nhập	DONE	NH
SCRUM-53	Phân quyền đăng nhập	DONE	NH
SCRUM-54	Thiết kế giao diện sản phẩm và thêm sản phẩm	QUẢN LÝ SẢN PHẨM	ĐC
SCRUM-56	Viết code thêm, xóa, tìm kiếm sản phẩm	QUẢN LÝ SẢN PHẨM	NH
SCRUM-57	Hiển thị danh sách và thông tin sản phẩm từ database	QUẢN LÝ SẢN PHẨM	ĐC
SCRUM-58	Thiết kế giao diện giỏ hàng và mua sản phẩm	QUẢN LÝ SẢN PHẨM	ĐC
SCRUM-59	Viết code giỏ hàng và mua sản phẩm	DONE	NH

Hình 5.1. Sprint Backlog của Sprint 1

- Kết quả: Hoàn thành Sprint 1 với tổng cộng 8 công việc được thực hiện thành công,

5.2.2.Sprint 2

- Thời gian: 30/06/2025 đến 10/07/2025.
- Kế hoạch:



Task ID	Task Description	Status	Assignee
SCRUM-60	Thiết kế giao diện quản lý người dùng	QUẢN LÝ NGƯỜI DÙNG	ĐC
SCRUM-61	Hiển thị danh sách người dùng từ database	QUẢN LÝ NGƯỜI DÙNG	NH
SCRUM-62	Viết code phân quyền người dùng, xóa, tìm kiếm người dùng	QUẢN LÝ NGƯỜI DÙNG	ĐC
SCRUM-63	Tạo giao diện hồ sơ thông tin và đơn hàng người dùng	QUẢN LÝ NGƯỜI DÙNG	ĐC
SCRUM-64	Viết code cập nhật thông tin người dùng	DONE	ĐC

Hình 5.2. Sprint Backlog của Sprint 2

- Kết quả: Hoàn thành Sprint 2 với tổng cộng 5 công việc được thực hiện thành công

5.2.3. Sprint 3

- Thời gian: 10/07/2025 đến 17/07/2025.
- Kế hoạch:



Hình 5.3. Sprint Backlog của Sprint 3

- Kết quả: Hoàn thành Sprint 3 với tổng cộng 3 công việc được thực hiện thành công.

5.2.4. Sprint 4

- Thời gian: 17/07/2025 đến 24/07/2025.
- Kế hoạch:



Hình 5.4. Sprint Backlog của Sprint 4

- Kết quả: Hoàn thành Sprint 4 với tổng cộng 3 công việc được thực hiện thành công.

5.3. Phân công nhiệm vụ của từng thành viên trong nhóm

Bảng 3. Bảng phân công nhiệm vụ các Sprint 1, 2, 3 và 4 trong Jira

STT	Công việc	Sprint	Phân hệ	Thành viên thực hiện
1	SCRUM 51: Tạo giao diện đăng nhập	Sprint 1	Đăng nhập	Đào Công Duy
2	SCRUM 52: Viết code đăng nhập	Sprint 1	Đăng nhập	Nguyễn Phi Hùng

3	SCRUM 53: Phân quyền đăng nhập	Sprint 1	Đăng nhập	Nguyễn Phi Hùng
4	SCRUM 54: Thiết kế giao diện sản phẩm và thêm sản phẩm	Sprint 1	Quản lý sản phẩm	Đào Công Duy
5	SCRUM 56: Viết code thêm, sửa, xóa, tìm kiếm sản phẩm	Sprint 1	Quản lý sản phẩm	Trần Tấn Khả
6	SCRUM 57: Hiển thị danh sách và thông tin sản phẩm từ database	Sprint 1	Quản lý sản phẩm	Trần Tấn Khả
7	SCRUM 58: Thiết kế giao diện giỏ hàng và mua sản phẩm	Sprint 1	Quản lý sản phẩm	Đào Công Duy
8	SCRUM 59 : Viết code giỏ hàng và mua sản phẩm	Sprint 1	Quản lý sản phẩm	Nguyễn Phi Hùng
9	SCRUM 60: Thiết kế giao diện quản lý người dùng	Sprint 2	Quản lý người dùng	Đào Công Duy
10	SCRUM 61: Hiển thị danh sách người dùng từ database	Sprint 2	Quản lý người dùng	Nguyễn Phi Hùng
11	SCRUM 62: Viết code phân quyền người dùng, xóa, tìm kiếm người dùng	Sprint 2	Quản lý người dùng	Trần Tấn Khả
12	SCRUM 63: Tạo giao diện hồ sơ thông tin và đơn hàng người dùng	Sprint 2	Quản lý người dùng	Đào Công Duy

13	SCRUM 64: Viết code cập nhật thông tin người dùng	Sprint 2	Quản lý người dùng	Trần Tấn Khá
14	SCRUM 65: Thiết kế giao diện quản lý đơn hàng	Sprint 3	Quản lý đơn hàng	Đào Công Duy
15	SCRUM 66: Viết code hiển thị đơn hàng, tìm kiếm đơn hàng	Sprint 3	Quản lý đơn hàng	Nguyễn Phi Hùng
16	SCRUM 67: Viết code hiển thị trạng thái đơn hàng	Sprint 3	Quản lý đơn hàng	Trần Tấn Khá
17	SCRUM 68: Tạo giao diện danh thu, báo cáo	Sprint 4	Báo cáo thống kê	Đào Công Duy
18	SCRUM 69: Viết code hiển thị danh thu, báo cáo	Sprint 4	Báo cáo thống kê	Nguyễn Phi Hùng
19	SCRUM 70: Hiển thị lịch sử hoạt động	Sprint 4	Báo cáo thống kê	Trần Tấn Khá

CHƯƠNG 6: KIỂM THỬ

6.1. Chiến lược kiểm thử

- Phát hiện và loại bỏ lỗi (bug) trước khi đưa phần mềm vào sử dụng.
- Đảm bảo hệ thống hoạt động đúng theo yêu cầu đã đặt ra.
- Đảm bảo các chức năng hoạt động ổn định, không phát sinh lỗi không mong muốn.
- Tối ưu hiệu suất và trải nghiệm người dùng.

Các loại kiểm thử được áp dụng

- Kiểm thử đơn vị (Unit Testing)
 - + Mỗi hàm, module nhỏ trong backend (controller, service) được viết test riêng.
 - + Sử dụng framework Jest (cho NodeJS) để chạy unit test.
 - + Ví dụ: Kiểm tra logic tính tổng tiền đơn hàng, kiểm tra validation form,...
- Kiểm thử tích hợp (Integration Testing)
 - + Kiểm tra các module hoạt động chung với nhau.
 - + Tích hợp giữa API backend và cơ sở dữ liệu.
 - + Kiểm tra API có phản hồi đúng dữ liệu hay không.
- Kiểm thử hệ thống (System Testing)
 - + Kiểm thử toàn bộ hệ thống theo các quy trình nghiệp vụ:
 - + Thêm sản phẩm → tạo đơn hàng → cập nhật tồn kho.
 - + Thêm khách hàng → tạo đơn mua hàng.
- Sử dụng Postman để viết test kịch bản RESTful API.

CHƯƠNG 7: ĐÁNH GIÁ VÀ KẾT LUẬN

7.1. Kết luận

Dự án “**Xây dựng Web Quản lý Cửa hàng Bán Đồ Thể Thao**” đã được triển khai theo đầy đủ quy trình phát triển phần mềm trong phạm vi môn học Công nghệ phần mềm, từ giai đoạn phân tích yêu cầu, thiết kế hệ thống, xây dựng giao diện, lập trình backend, kiểm thử cho đến triển khai thử nghiệm. Trong quá trình thực hiện, nhóm đã áp dụng mô hình phát triển linh hoạt Agile với phương pháp Scrum để chia nhỏ công việc, lập kế hoạch Sprint và đánh giá hiệu quả từng giai đoạn.

Kết quả triển khai cho thấy hệ thống đã đáp ứng đầy đủ các yêu cầu chức năng như: quản lý sản phẩm, kho hàng, khách hàng, đơn hàng, báo cáo và phân quyền nhân viên. Giao diện người dùng thân thiện, phù hợp với trải nghiệm thực tế, hỗ trợ tốt trên cả thiết bị máy tính và di động. Việc ứng dụng các công nghệ hiện đại như **React/NextJS** cho frontend, **NodeJS/Express** cho backend, cùng với **MongoDB** và công cụ kiểm thử tự động đã giúp nâng cao chất lượng phần mềm, giảm thiểu lỗi và cải thiện tốc độ phát triển. Thông qua quá trình kiểm thử toàn diện, hệ thống đạt tỷ lệ test case thành công cao (trên 97%), không còn lỗi nghiêm trọng tồn đọng, đảm bảo khả năng triển khai thực tế. Nhóm cũng đã hoàn thiện các tài liệu kỹ thuật như sơ đồ kiến trúc, tài liệu thiết kế cơ sở dữ liệu, tài liệu kiểm thử, hướng dẫn triển khai và sử dụng.

Ngoài ra, trong quá trình thực hiện, nhóm đã nâng cao được các kỹ năng quan trọng như: kỹ năng làm việc nhóm, kỹ năng quản lý thời gian, kỹ năng giao tiếp, kỹ năng sử dụng công cụ Jira, GitHub, Figma và công cụ kiểm thử phần mềm.

7.2. Hạn chế

Mặc dù hệ thống đạt được phần lớn mục tiêu đề ra, nhưng vẫn tồn tại một số hạn chế nhất định:

- Chưa có hệ thống quản lý hóa đơn chi tiết (**PDF, in ấn**).
- Chưa triển khai hệ thống phân tích nâng cao (**BI – Business Intelligence**) để
- hỗ trợ ra quyết định.
- Chưa tích hợp chức năng thanh toán trực tuyến.
- Một số tính năng nâng cao như tự động gửi email/SMS, nhắc nhở tồn kho thấp... chưa được tích hợp do giới hạn về thời gian và phạm vi đồ án.

7.3. Hướng phát triển trong tương lai

Để nâng cao hiệu quả vận hành và mở rộng ứng dụng trong thực tế, nhóm đề xuất một số hướng phát triển tiếp theo như sau:

7.3.1. Mở rộng chức năng

- **Tạo hóa đơn điện tử:** xuất file PDF hóa đơn, tích hợp in hóa đơn qua máy in nhiệt.
- **Thông báo tự động:** hệ thống gửi thông báo đến người dùng khi đơn hàng được xác nhận, sản phẩm sắp hết hàng, v.v.
- **Quản lý điểm thưởng/thẻ thành viên** cho khách hàng thân thiết.
- Thêm chức năng đánh giá sản phẩm và quản lý phản hồi từ khách hàng.

7.3.2. Cải tiến kiến trúc kỹ thuật

- **Chuyển sang mô hình Microservices** để dễ mở rộng, bảo trì từng phần riêng biệt.
- **Tối ưu hóa cơ sở dữ liệu**, tạo chỉ mục, caching với Redis để tăng hiệu suất.

7.3.3. Phát triển ứng dụng di động (Mobile App)

- Xây dựng phiên bản di động sử dụng React Native hoặc Flutter để hỗ trợ quản lý trên điện thoại, máy tính bảng.
- Hỗ trợ quản lý đơn hàng, thống kê doanh thu từ xa.

7.3.4. Áp dụng trí tuệ nhân tạo (AI)

- Dự đoán sản phẩm bán chạy trong tương lai.
- Gợi ý sản phẩm cho khách hàng dựa vào lịch sử mua hàng.
- Phân tích hành vi người dùng để tối ưu chiến dịch marketing.

TÀI LIỆU THAM KHẢO

1. Ian Sommerville (2015) – Software Engineering, 10th Edition, Pearson Education.

→ Tài liệu nền tảng về quy trình phát triển phần mềm, các mô hình như Waterfall, Agile, nguyên lý thiết kế và kiểm thử.

2. Roger S. Pressman & Bruce R. Maxim (2014) – Software Engineering: A Practitioner's Approach, 8th Edition, McGraw-Hill.

→ Cung cấp kiến thức sâu rộng về kỹ thuật phần mềm, quy trình phân tích yêu cầu, kiểm thử và bảo trì hệ thống.

3. Eric Freeman, Elisabeth Robson (2020) – Head First Design Patterns, O'Reilly Media.

→ Giới thiệu các mẫu thiết kế phần mềm phổ biến áp dụng trong kiến trúc MVC, RESTful API.

4. Documentation - ReactJS: <https://reactjs.org/docs/getting-started.html>

→ Hướng dẫn sử dụng thư viện ReactJS để phát triển frontend theo kiến trúc component.

5. Documentation - Next.js: <https://nextjs.org/docs>

→ Tài liệu chính thức về framework NextJS được sử dụng trong frontend của hệ thống.

6. Documentation - Node.js: <https://nodejs.org/en/docs/>

→ Tài liệu về môi trường chạy backend, xử lý API, middleware, routing.

7. ExpressJS Guide – <https://expressjs.com/en/starter/installing.html>

→ Hướng dẫn chi tiết về xây dựng backend server bằng ExpressJ.