TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

Khoa Công Nghệ Thông Tin

🙠🙟🕮🙝🙢



**ĐỒ ÁN TỐT NGHIỆP**

**ĐẠI HỌC NGÀNH CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI: XÂY DỰNG WEBSITE BÁN GIÀY NỮ SỬ DỤNG REACTJS , MONGODB ,NODEJS**

**GVHD : TS.Nguyễn Bá Nghiễn**

**Sinh viên : Chu Thế Mạnh**

**Mã sinh viên : 2017601707**

**Lớp : CNTT2 - K12**

***Hà Nội 05/2021***

# LỜI CẢM ƠN

Em xin chân thành cảm ơn các thầy các cô trong khoa Công Nghệ Thông Tin đã trang bị những kiến thức quý báu cho em trong quá trình học tập và làm đề tài tại trường Đại Học Công Nghiệp Hà Nội.

Nhờ công lao giảng dạy, và chỉ bảo tận tình của các thầy các cô mà em đã có được những kiến thức chuyên ngành về công nghệ thông tin để thực hiện đề tài này.

Đặc biêt, em cảm ơn thầy Nguyễn Bá Nghiễn đã tận tình hướng dẫn, chỉ bảo, đưa ra những đóng góp ý kiến rất hữu ích và quý báo cho em trong suốt quá trình thực hiện đề tài thực tập cho đến đồ án tốt nghiệp này.

Em gửi lời cảm ơn đến những người bạn, người thầy đã chia sé và cung cấp các tài liệu hữu ích cho em.

Mặc dù rất nỗ lực thực hiện đề tài nhưng chắc hẳn vẫn không thể tránh khỏi những sai sót.

Em rất mong nhận được những ý kiến đóng góp của thầy cô và các bạn để đề tài của em ngày càng hoàn thiện và hướng tới áp dụng thực tế hơn.

Em xin chân thành cảm ơn!

**MỤC LỤC**

[LỜI CẢM ƠN 1](#_Toc71717538)

[DANH MỤC HÌNH ẢNH 5](#_Toc71717539)

[CHƯƠNG 1. TỔNG QUAN 7](#_Toc71717540)

[1.1. Giới thiệu 7](#_Toc71717541)

[1.2. Lý do chọn đề tài 7](#_Toc71717542)

[1.3. Đánh giá hiện trạng 8](#_Toc71717543)

[1.4. Yêu cầu đối với hệ thống 8](#_Toc71717544)

[1.5. Mục tiêu đề tài 8](#_Toc71717545)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 10](#_Toc71717546)

[2.1. Xác định các tác nhân 10](#_Toc71717547)

[**2.2** **. Mô tả chi tiết** 12](#_Toc71717548)

[**2.2.1.** **Mô tả Use case Xem sản phẩm** 12](#_Toc71717549)

[**2.2.2.** **Mô tả Use Case Tìm sản phẩm** 12](#_Toc71717550)

[**2.2.3.** **Mô tả Use Case Mua sản phẩm** 13](#_Toc71717551)

[**2.2.4.** **Mô tả Use Case Xem lịch sử mua** 15](#_Toc71717552)

[**2.2.6.** **Mô tả Use Case Thêm sản phẩm** 15](#_Toc71717553)

[**2.2.7.** **Mô tả Use Case Sửa sản phẩm** 16](#_Toc71717554)

[**2.2.8.** **Mô tả Use Case Theo dõi đơn hàng** 17](#_Toc71717555)

[**2.2.9.** **Mô tả Use Case Xem sản phẩm(Admin)** 18](#_Toc71717556)

[**2.2.10.** **Mô tả Use Case Thêm bình luận đánh giá** 19](#_Toc71717557)

[**2.2.11.** **Mô tả Use Case Quản lý đơn hàng** 20](#_Toc71717558)

[**2.2.12.** **Mô tả Use Case Xem báo cáo** 21](#_Toc71717559)

[**2.2.13.** **Mô tả Use Case Tạo tài khoản** 22](#_Toc71717560)

[**2.2.14.** **Mô tả Use Case Đăng nhập** 23](#_Toc71717561)

[**2.2.15.** **Mô tả Use Case Quản lý User** 24](#_Toc71717562)

[**2.2.16.** **Mô tả Use Case Đăng Xuất** 25](#_Toc71717563)

[**2.3.** **Biểu đồ trình tự** 26](#_Toc71717564)

[**2.3.1.** **Use case Đăng nhập** 26](#_Toc71717565)

[**2.3.2.** **Use Case Đăng ký** 27](#_Toc71717566)

[**2.3.3.** **Use Case Xem sản phẩm** 28](#_Toc71717567)

[**2.3.4.** **Use Case Tìm kiếm sản phẩm** 28](#_Toc71717568)

[**2.3.5.** **Use Case Mua sản phẩm vào giỏ hàng** 29](#_Toc71717569)

[**2.3.6.** **Use Case Thêm bình luận** 30](#_Toc71717570)

[**2.3.7.** **Use Case Quản lý sản phẩm** 31](#_Toc71717571)

[**2.3.8.** **Use Case Thêm yêu thích và hiển thị** 32](#_Toc71717572)

[**2.3.9.** **Use Case Cập nhật đơn hàng** 33](#_Toc71717573)

[**2.3.10.** **Use case Thêm đơn hàng** 34](#_Toc71717574)

[**2.4.** **Biểu đồ lớp** 35](#_Toc71717575)

[**2.5.** **Sơ đồ hoạt động** 36](#_Toc71717576)

[**2.6.** **Cơ sở dữ liệu** 41](#_Toc71717577)

[**2.7.** **Thiết kế một số giao diện chính** 46](#_Toc71717578)

[CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM 55](#_Toc71717579)

[3.1. Giới thiệu công nghệ lựa chọn để cài đặt 55](#_Toc71717580)

[3.1.1. Công nghệ Frontend 55](#_Toc71717581)

[3.1.2. Công nghệ Backend 56](#_Toc71717582)

[3.2. Cài đặt các chức năng chính của hệ thống 60](#_Toc71717583)

[3.3. Một số kịch bản thử nghiệm 61](#_Toc71717584)

[3.3.1. Kịch bản mua hàng 61](#_Toc71717585)

[**3.3.2.** **Kịch bản gửi đánh giá** 64](#_Toc71717586)

[**3.3.3.** **Kịch bản đăng ký tài khoản** 65](#_Toc71717587)

[KẾT LUẬN 67](#_Toc71717588)

[TÀI LIỆU THAM KHẢO 68](#_Toc71717589)

**DANH MỤC TỪ VIẾT TẮT**

|  |  |
| --- | --- |
| **BE** | Backend |
| **FE** | Frontend |
| **UI** | Giao diện |
| **NSX** | Nhà sản xuất |
| **CSDL** | Cơ sở dữ liệu |
| **DB** | Database |
|  |  |

# DANH MỤC HÌNH ẢNH

[Hình 1. Sơ đồ chi tiết use case 11](#_Toc71717590)

[Hình 2 . Biểu đồ trình tự use case Đăng nhập. 26](#_Toc71717591)

[Hình 3 . Biểu đồ trình tự use case Đăng nhập. 27](#_Toc71717592)

[Hình 4. Biểu đồ trình tự usecase xem sản phẩm và chi tiết sản phẩm. 28](#_Toc71717593)

[Hình 5. Biểu đồ trình tự usecase tìm kiếm sản phẩm. 28](#_Toc71717594)

[Hình 6. Biểu đồ trình tự usecase mua sản phẩm vào giỏ hàng. 29](#_Toc71717595)

[Hình 7. Biểu đồ trình tự usecase thêm bình luận. 30](#_Toc71717596)

[Hình 8. Biểu đồ trình tự usecase quản lý sản phẩm. 31](#_Toc71717597)

[Hình 9. Biểu đồ trình tự usecase thêm yêu thích. 32](#_Toc71717598)

[Hình 10. Biểu đồ trình tự usecase quản lý đơn hàng và tình trạng đơn hàng 33](#_Toc71717599)

[Hình 11. Biểu đồ trình tự usecase thêm đơn hàng. 34](#_Toc71717600)

[Hình 12. Biểu đồ lớp 35](#_Toc71717601)

[Hình 13. Sơ đồ hoạt động đăng ký tài khoản. 36](#_Toc71717602)

[Hình 14. Sơ đồ hoạt động đăng nhập. 36](#_Toc71717603)

[Hình 15. Sơ đồ hoạt động cập nhật thông tin cá nhân. 37](#_Toc71717604)

[Hình 16. Sơ đồ hoạt động xem sản phẩm và thêm giỏ hàng. 37](#_Toc71717605)

[Hình 17. Sơ đồ hoạt động cập nhật thông tin trong giỏ hàng. 38](#_Toc71717606)

[Hình 18. Sơ đồ hoạt động xem lịch sử đặt hàng. 38](#_Toc71717607)

[Hình 19. Sơ đồ hoạt động quy trình mua hàng trên website. 39](#_Toc71717608)

[Hình 20. Sơ đồ hoạt động quản lý thêm sản phẩm trên website. 40](#_Toc71717609)

[Hình 21. Sơ đồ hoạt động quản lý sửa và xóa sản phẩm . 40](#_Toc71717610)

[Hình 22. Sơ đồ hoạt động quản lý tài khoản. 41](#_Toc71717611)

[Hình 23. Biểu đồ các collection 41](#_Toc71717612)

[Hình 24. Biểu đồ order model 42](#_Toc71717613)

[Hình 25. Biểu đồ user model 43](#_Toc71717614)

[Hình 26. Biểu đồ product model 43](#_Toc71717615)

[Hình 27. Biểu đồ yeuthich model 44](#_Toc71717616)

[Hình 28. Hình ảnh trang chủ của website 46](#_Toc71717617)

[Hình 29. Hình ảnh trang sản phẩm 46](#_Toc71717618)

[Hình 30. Hình ảnh trang chi tiết sản phẩm 47](#_Toc71717619)

[Hình 31. Hình ảnh trang giỏ hàng 47](#_Toc71717620)

[Hình 32. Hình ảnh trang đăng nhập 48](#_Toc71717621)

[Hình 33. Hình ảnh trang đăng ký tài khoản 48](#_Toc71717622)

[Hình 34. Hình ảnh trang nhập địa chỉ giao hàng 49](#_Toc71717623)

[Hình 35. Hình ảnh trang xác nhận mua 49](#_Toc71717624)

[Hình 37. Hình ảnh trang update thông tin tài khoản 50](#_Toc71717625)

[Hình 38. Hình ảnh trang xem lịch sử mua hàng 51](#_Toc71717626)

[Hình 39. Hình ảnh trang thống kê. 51](#_Toc71717627)

[Hình 40. Hình ảnh trang quản lý sản phẩm 52](#_Toc71717628)

[Hình 41. Hình ảnh trang quản lý order 52](#_Toc71717629)

[Hình 42. Hình ảnh trang quản lý users 53](#_Toc71717630)

[Hình 43. Hình ảnh trang update thông tin user và phân quyền tài khoản 53](#_Toc71717631)

[Hình 44. Hình ảnh trang update trạng thái đơn hàng. 54](#_Toc71717632)

[Hình 45. Hình ảnh trang tìm kiếm sản phẩm. 54](#_Toc71717633)

[Hình 47. Hình ảnh header của user. 61](#_Toc71717634)

[Hình 48. Hình ảnh trang sản phẩm. 61](#_Toc71717635)

[Hình 49. Hình ảnh trang giỏ hàng. 62](#_Toc71717636)

[Hình 50. Hình ảnh trang điền thông tin địa chỉ nhận. 62](#_Toc71717637)

[Hình 51. Hình ảnh trang đăng nhập. 63](#_Toc71717638)

[Hình 52. Hình ảnh trang order và method thanh toán. 63](#_Toc71717639)

[Hình 53. Hình ảnh trang đánh giá. 64](#_Toc71717640)

[Hình 54. Hình ảnh trang đăng ký tài khoản. 66](#_Toc71717641)

# CHƯƠNG 1. TỔNG QUAN

## Giới thiệu

Trong xu thế phát triển mạnh mẽ của thương mại điện tử trên thế giới, các doanh nghiệp Việt Nam , đặc biệt là các doanh nghiệp vừa và nhỏ cũng đã bắt đầu nhận thức được sự quan trọng của việc ứng dụng thương mại điện tử vào kinh doanh và buôn bán. Tuy nhiên, do còn nhiều hạn chế trong nhận thức và cơ sở hạ tầng nên việc ứng dụng thương mại điện tử vẫn còn chậm và khó khăn.

Tuy nhiên, để lập một website thương mại điện tử cho công ty mình là không hề khó khăn. Bạn có thể thuê các công ty uy tín hoặc cũng có thể mua lại trang. Các bạn trẻ hoặc người mua hàng thời đại này rất thích việc mua sắm online .Do vậy,em quyết định thực hiện đề tài này với mong muốn đưa đến nhiều sự lựa chọn hơn cho khách hàng, và tăng số lượng đơn hàng cho gia đình.Tên đề tài của em là : Xây dựng website bán giày nữ sử dụng ReactJs , MongoDB , NodeJs.

## 1.2. Lý do chọn đề tài

- Do cửa hàng gia đình chưa có trang web riêng để tiếp cận được với nhiều khách hàng hơn

- Công nghệ phát triển rất nhanh nên em muốn tìm hiểu thêm và xây dựng website này bằng ReactJs , MongoDB , NodeJs (những công nghệ nổi tiếng hiện nay ).

- Đam mê xây dựng một website cho bản thân để hoàn thiện hơn về kiến thức thực tế .

## 1.3. Đánh giá hiện trạng

- Những điểm cần khắc phục:

+ Năng suất hiệu quả của công việc chưa cao.

+ Chưa có hình thức quản lý hiệu quả.

+ Chưa có trang web chính thức để quản lý.

## 1.4. Yêu cầu đối với hệ thống

- Yêu cầu về chức năng

+ Quản trị: Quản lý được danh sách tất cả mọi người dùng trên hệ thống, quản lý hàng , quản lý đơn hàng , thống kế ,…

+ Khách hàng: Xem được danh mục sản phẩm , lựa chọn mua sản phẩm, xem được giỏ hàng của mình, xem lịch sử mua hàng, cập nhật thông tin tài khoản, ( phát triển thêm theo dõi đơn hàng đã đặt, chat với admin, thanh toán bằng thẻ, …)

+ Sản phẩm : Hiển thị danh sách những mặt hàng đang được bán nhiều nhất, và theo giá thấp nhất đến cao.

- Yêu cầu phi chức năng

+ Hệ thống phải có giao diện dễ nhìn và dễ sử dụng đối với người dùng.

+ Hê thống phải có khả năng bảo trì được.

+ Hệ thống phải có tính bảo mật cao.

## 1.5. Mục tiêu đề tài

- Hoàn thành được các yêu cầu cơ bản của một website E-commerce (đã nêu ở mục 1.4 bên trên).

- Có thể làm thêm một số chức năng nâng cao , bảo mật tốt hơn để có thể ứng dụng vào thực tế.

- Biết cách phân tích thiết kế một hệ thống E-commerce.

- Học được cách sử dụng công nghệ mới hiện nay (ReactJs, NodeJs, …).

- Nắm vững hơn các kiến thức căn bản về Html, Css, Jss…

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## 2.1. Xác định các tác nhân

- Quản trị hệ thống : Admin là người quản lý website chung

+ Quản lý sản phẩm

+ Quản lý hóa đơn

+ Thêm tài khoản khách hàng hoặc nhân viên

+ Quản lý báo cáo thống kê

+ Quản lý bán hàng

+ Đăng nhập, đăng xuất, kích hoạt tài khoản người dùng

+ Chat với khách hàng

- Khách hàng: là người dùng truy cập vào website

+ Có thể xem thông tin sản phẩm

+ Mua sản phẩm

+ Xem lịch sử mua hàng

+ Theo dõi đơn hàng

+ Đăng nhập, đăng xuất, tạo tài khoản

+ Thêm sửa bình luận

+ Tìm kiếm sản phẩm

+ Liên hệ với người bán



Hình 1. Sơ đồ chi tiết use case

* 1. **. Mô tả chi tiết**
     1. **Mô tả Use case Xem sản phẩm**

Use case này cho phép người dùng có thể truy cập và xem thông tin sơ bộ của tất cả sản phẩm hoặc chi tiết từng sản phẩm như : tên sản phẩm , ảnh sản phẩm , mô tả chi tiết , số sao , số người đánh giá . Hệ thống sẽ lấy dữ liệu từ API về màn hình . Ngoài ra người dùng có thể ấn xem chi tiết để hiện thị ra thêm những thông tin khác của sản phẩm.

* Luồng sự kiện
  + Luồng cơ bản

1. Sau khi người dùng click vào “Product” sẽ hiện lên danh sách các sản phẩm của cửa hàng.
   1. Thông tin sản phẩm gồm tên sản phẩm, ảnh , số sao và số đánh giá , giá của sản phẩm.
   2. Người dùng có thể kích vào tên - ảnh trang sẽ đưa đến thông tin chi tiết của từng sản phẩm

Use case kết thúc

* Luồng rẽ nhánh
* Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
* Không có.
* Tiền điều kiện
* Không có.
* Hậu điều kiện
* Thông tin các sản phẩm phải được hiển thị.
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Tìm sản phẩm**

Use case này cho phép người dùng có thể truy cập và tìm kiếm các sản phẩm theo nhiều tiêu chí khác nhau

* Luồng sự kiện
  + Luồng cơ bản
  1. Sau khi người dùng click vào “Product” sẽ hiện lên danh sách tìm kiếm các sản phẩm theo nhiều tiêu chí : “Theo tên” , “Theo giá” , “Theo loại” , …
     1. Khi người dùng chọn “Theo giá” hệ thống sẽ trả về danh sách các sản phẩm mà người dùng đã thực hiện. Người dùng chọn một trong các tìm kiếm, hoặc kết hợp các tìm kiếm và click vào “Tìm” để tìm sản phẩm theo ý mình, hệ thống sẽ trả về danh sách tìm kiếm.
     2. Khi người dùng chọn “Theo loại” hệ thống sẽ trả về danh sách các sản phẩm mà người dùng đã thực hiện. Người dùng chọn một trong các tìm kiếm, hoặc kết hợp các tìm kiếm và click vào “Tìm” để tìm sản phẩm theo ý mình, hệ thống sẽ trả về danh sách tìm kiếm.

Use case kết thúc

* Luồng rẽ nhánh

1. Tại bước 1.1.1 và 1.1.2, nếu người dùng nhập thông tin, Thì sẽ trả về kết quả không tìm thấy sản phẩm tương ứng.
2. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.

* Các yêu cầu đặc biệt
* Không có.
* Tiền điều kiện
* Không có
* Hậu điều kiện
* Thông tin các sản phẩm phải được hiển thị.
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Mua sản phẩm**

Use case này cho phép người dùng có thể mua đăng ký mua sản phẩm qua website .

* Luồng sự kiện
  + Luồng cơ bản

1. Sau khi người dùng click vào “Product” từ trang sản phẩm trong tại Use Case Xem sản phẩm, trang web sẽ được đưa đến đường dẫn hiển thị một form gồm các trường thông tin của sản phẩm. Người dùng nhập thông tin thêm vào form là số lượng và cỡ và click “Add to cart” hệ thống chuyển đến trang Cart . Người dùng xác nhận mua thì ấn “Mua” hoặc không muốn mua thì ấn vào nút “Xóa” để remove sản phẩm ra khỏi giỏ
2. Người dùng có thể gửi sao và đánh giá khi mua hàng .

Use case kết thúc

* Luồng rẽ nhánh

1. Tại bước 2, nếu sản phẩm hết hàng sẽ hiện ra thông tin sản phẩm và không có button Add to cart . Người dùng sẽ ấn trờ về “Back to result” để về trang Product
2. Nếu đã đăng nhập thì màn hình sẽ chuyển đến trang thanh toán và ngược lại sẽ chuyển đến trang đăng nhập
3. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.

* Các yêu cầu đặc biệt
* Không có.
* Tiền điều kiện
* Tài khoản phải được đăng nhập
* Hậu điều kiện
* Không có.
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Xem lịch sử mua**

Use case này cho phép người dùng có thể xem thông tin các lần mua hàng trước đó của bản thân.

* Luồng sự kiện
  + Luồng cơ bản
    1. Sau khi người dùng click vào “Cart” thì sẽ hiện ra thông tin giỏ hàng hiện tại . Người dùng click vào “Xem lịch sử mua hàng” màn hình hiển thị ra những lần order trước đó của tài khoản khách hàng .
    2. Người dùng có thể tìm kiếm theo năm để rút gọn khoảng tìm kiếm .

Use case kết thúc

* Luồng rẽ nhánh
  + 1. Nếu khách hàng chưa có lịch sử mua hàng thì hiển thị ra “Bạn chưa đặt hàng bao giờ”.
    2. Tại bất kì bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
* Không có.
* Tiền điều kiện
* Tài khoản phải được đăng nhập
* Hậu điều kiện
* Không có.
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Thêm sản phẩm**

Use case này cho phép Admin có thể thêm mới sản phẩm

* Luồng sự kiện
  + Luồng cơ bản
    1. Sau khi Admin click vào “Thêm sản phẩm” từ trang Hiển thị sản phẩm, trang web sẽ được đưa đến đường dẫn hiển thị một form gồm các trường của sản phẩm. Admin nhập thông tin thêm vào form và click “Thêm sản phẩm” hệ thống sẽ thêm sản phẩm vào data “Products” hoặc không muốn thêm thì ấn vào nút “Hủy” để về lại trang trước

Use case kết thúc

* Luồng rẽ nhánh
  1. Tại bước 1, nếu người dùng nhập sai mã sản phẩm hoặc sai bất cứ định dạng dữ liệu nào từ bảng sản phẩm “Nhập sai định dạng dữ liệu, xin nhập lại”, click OK thì trang sẽ load lại và tiếp tục nhập các tìm kiếm
  2. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
* Không có.
* Tiền điều kiện
* Tài khoản phải được đăng nhập
* Hậu điều kiện
* Sản phẩm phải được thêm vào db, nếu không thành công phải có thông báo
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Sửa sản phẩm**

Use case này cho phép Admin có thể sửa thông tin sản phẩm

* Luồng sự kiện
  + Luồng cơ bản
    1. Sau khi người dùng click vào biểu tượng “Sửa” từ bất kì dòng nào của dữ liệu các dòng trong trang Hiển thị sản phẩm trong tại Use Case Quản lý sản phẩm, trang web sẽ được đưa đến đường dẫn hiển thị một form gồm các trường của sản phẩm. Người dùng nhập thông tin sửa vào form và click “Sửa sản phẩm” hệ thống sẽ sửa update sản phẩm vào data “Products” hoặc không muốn thêm thì ấn vào nút “Hủy” để về lại trang trước

Use case kết thúc

* Luồng rẽ nhánh
  + 1. Tại bước 1, nếu người dùng nhập sai mã sản phẩm hoặc sai bất cứ định dạng dữ liệu nào từ bảng sản phẩm “Nhập sai định dạng dữ liệu, xin nhập lại”, click OK thì trang sẽ load lại và tiếp tục nhập các tìm kiếm
    2. Đồng thời tại bước 1 nếu người dùng không được phân quyền trong hoạt động nào thì trang web sẽ trả về thông báo “Bạn không có quyền thực hiện hành động này”.
    3. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
* Không có.
* Tiền điều kiện
* Tài khoản phải được đăng nhập
* Hậu điều kiện
* Sản phẩm phải được sửa vào db, nếu không thành công phải có thông báo
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Theo dõi đơn hàng**

Use case này cho phép người dùng có thể truy cập và tìm kiếm các thông tin về đơn hàng mình đang đặt.

* Luồng sự kiện
  + Luồng cơ bản

Sau khi người dùng click vào “Profile” sẽ hiện lên danh sách các thông tin cá nhân và đơn hàng. Người dùng click vào đơn hàng để hiển thị ra trạng thái chính xác của đơn hàng .

Khi đơn hàng ở trạng thái “Chờ xác nhận” thì khách hàng có thể gửi trang thái hủy hàng.

Use case kết thúc

* Luồng rẽ nhánh

1. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.

* Các yêu cầu đặc biệt
* Không có.
* Tiền điều kiện
* Tài khoản phải được đăng nhập
* Hậu điều kiện
* Thông tin các đơn hàng phải được hiển thị.
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Xem sản phẩm(Admin)**

Use case này cho phép Admin có thể truy cập và xem các sản phẩm theo nhiều trang khác nhau

* Luồng sự kiện
  + Luồng cơ bản
    1. Sau khi Admin click vào “Quản trị product” sẽ hiện lên danh sách các sản phẩm theo nhiều trang , nhiều tiêu chí.
    2. Khi Admin chọn “Theo giá” hệ thống sẽ trả về danh sách các sản phẩm mà admin đã thực hiện. Admin chọn một trong các tìm kiếm, hoặc kết hợp các tìm kiếm và click vào “Tìm” để tìm sản phẩm theo ý mình, hệ thống sẽ trả về danh sách tìm kiếm.
    3. Khi Admin chọn “Theo loại” hệ thống sẽ trả về danh sách các sản phẩm mà admin đã thực hiện. Admin chọn một trong các tìm kiếm, hoặc kết hợp các tìm kiếm và click vào “Tìm” để tìm sản phẩm theo ý mình, hệ thống sẽ trả về danh sách tìm kiếm.

Use case kết thúc

* Luồng rẽ nhánh
  + 1. Tại bước 1 và 2, nếu Admin nhập thông tin, Thì sẽ trả về kết quả không tìm thấy sản phẩm tương ứng.
    2. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
* Tài khoản là Admin.
* Tiền điều kiện
* Phải đăng nhập mới có thể truy cập.
* Hậu điều kiện
* Thông tin các sản phẩm phải được hiển thị.
* Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Thêm bình luận đánh giá**

Use case này cho phép Khách Hàng có tài khoản có thể gửi đánh giá của mình với mỗi sản phẩm.

* Luồng sự kiện
  + Luồng cơ bản
    1. Sau khi Người click vào Sản phẩm sẽ hiện lên trang detailsProduct.
  1. Khi Người dùng chọn “Đánh giá” hệ thống sẽ trả về danh sách các đánh giá đã có trước đó của sản phẩm. Người dùng có thể thêm đánh giá ngay bên dưới.
  2. Khi bạn không thể gửi 2 đánh giá , hệ thống sẽ ghi đè lại đánh giá trước đó của bạn coi như là lần mới nhất.

Use case kết thúc

* Luồng rẽ nhánh
  + Tại bước 1.1 và 1.2, nếu người dùng không nhập thông tin comment thì sẽ cảnh báo.
  + Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
  + Tài khoản là người dùng.
* Tiền điều kiện
  + Phải đăng nhập trước đó.
* Hậu điều kiện
  + Thông tin các đánh giá phải được hiển thị.
* Điểm mở rộng
  + Không có
    1. **Mô tả Use Case Quản lý đơn hàng**

Use case này cho phép Admin có thể truy cập và xác nhận đặt hàng với các đơn hàng và cập nhật trạng thái của đơn hàng cho người dùng.

* + - * Luồng sự kiện
* Luồng cơ bản

1. Sau khi người dùng click vào “Đơn hàng” sẽ hiển thị ra bảng các đơn hàng đang được đặt .

1.1. Nếu admin chọn “Chấp nhận”, thì đơn hàng đã được xác nhận đặt và người dùng có thể thấy trang thái đơn hàng chuyển sang “Đang đóng gói”.

1.2. Nếu người dùng chọn “Cập nhật”, thì màn hình chuyển đến form có thông tin của sản phẩm vừa click . Admin cập nhật lại thông tin trang thái cho đơn hàng.

Use case kết thúc

* Luồng rẽ nhánh
  + 1. Tại bước 2 nếu khách hàng nhập sai định dạng một trong các thông tin của form nhập thì hệ thống sẽ đưa ra thông báo “Nhập sai dữ liệu, xin sửa lại thông tin”
    2. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
       - Các yêu cầu đặc biệt
* Không có
  + - * Tiền điều kiện
* Tài khoản phải được đăng nhập
  + - * Hậu điều kiện
* Không có.
  + - * Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Xem báo cáo**

Use case này cho phép người dùng có thể truy cập xem doanh số , số đơn hàng , số tiền trong tháng.

* Luồng sự kiện
  + Luồng cơ bản
    1. Sau khi người dùng click vào “Thống kê” từ trang Admin, trang web sẽ được đưa đến đường dẫn hiển thị một form gồm các thông tin doanh số , đơn hàng , số đơn hủy , …

Use case kết thúc

* Luồng rẽ nhánh
  1. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
  + - Không có.
* Tiền điều kiện
  + - Tài khoản phải được đăng nhập
* Hậu điều kiện
  + - Không có
* Điểm mở rộng
  + - Không có.
    1. **Mô tả Use Case Tạo tài khoản**

Use case này cho phép người dùng tạo tài khoản người dùng nếu chưa có tài khoản

* Luồng sự kiện
  + Luồng cơ bản
    1. Sau khi người dùng click vào biểu tượng “Tạo tài khoản” từ trang Login, trang web sẽ được đưa đến đường dẫn hiển thị một form gồm các trường của thông tin người dùng. Người dùng nhập thông tin cần có vào form và click “Tạo tài khoản” hệ thống gửi sẽ trả thông báo tới user về việc lập thành công và đợi kiểm duyệt từ admin hoặc không muốn tạo thì ấn vào nút “Quay về” để về lại trang trước

Use case kết thúc

* Luồng rẽ nhánh
  + 1. Tại bước 2 nếu khách hàng nhập sai định dạng một trong các thông tin của form nhập thì hệ thống sẽ đưa ra thông báo “Không đúng kiểu dữ liệu”
* Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
* Các yêu cầu đặc biệt
  + - Không có.
* Tiền điều kiện
  + - Tài khoản phải được đăng nhập
* Hậu điều kiện
  + - Không có.
* Điểm mở rộng
  + - Không có.
    1. **Mô tả Use Case Đăng nhập**
* Luồng sự kiện:
  + Luồng cơ bản:

1. UC bắt đầu bằng form thông tin cần đăng nhập gồm tài khoản và mật khẩu yêu cầu người dùng nhập để đăng nhập và sử dụng các chức năng với quyền từ tài khoản

2. Người dùng nhập tên tài khoản và mật khẩu sau đó click nút đăng nhập.

3. Hệ thống kiểm tra thông tin đăng nhập trên bảng “User” trong API.Sau đó hiển thị thông báo thành công màn hình hiển thị thông tin người dung. UC kết thúc.

* + Luồng rẽ nhánh:

1. Tại bất kì thời điểm nào trong quá trình thực hiện UC nếu hệ thống không thể kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi vào use case kết thúc.

2. Nếu hệ thống không tìm thấy thông tin được nhập giống thông tin trong CSDL thì hệ thống hiển thị thông báo nhập sai tên tài khoản hoặc mật khẩu và yêu cầu nhập lại.

* Các yêu cầu đặc biệt:
* Không có.
* Tiền điều kiện:
* Không có.
* Hậu điều kiện:
* Màn hình hiển thị thông báo đăng nhập thành công và truy cập được vào trang đăng nhập
* Điểm mở rộng:

-Không có

* + 1. **Mô tả Use Case Quản lý User**

Use case này cho phép Admin có thể truy cập và quản lý user và cập nhật thông tin, active các loại tài khoản của cho người dùng.

* + - * Luồng sự kiện
* Luồng cơ bản

1. Sau khi Admin click vào “Quản lý User” sẽ hiển thị ra bảng các tài khoản có trong hệ thống .

1.1. Nếu admin chọn “Cập Nhật”, thì trang sẽ chuyển đến trang chỉnh sửa thông tin user.Admin có thể cấp quyền cho tài khoản tại đây.

1.2. Nếu Admin chọn vào “Xóa”, thì trang sẽ hiện ra thông báo có chắc muôn xóa hay không? Có thì thông báo xóa thành công.

Use case kết thúc

* Luồng rẽ nhánh
  + 1. Tại bước 1.1 nếu Admin nhập sai định dạng một trong các thông tin của form nhập thì hệ thống sẽ đưa ra thông báo “Nhập sai dữ liệu, xin sửa lại thông tin”
    2. Tại bất kỳ bước nào trong quá trình use case thực hiện mà hệ thống không thể kết nối được với cơ sở dữ liệu, hệ thống sẽ thông báo lỗi, use case kết thúc.
       - Các yêu cầu đặc biệt
* Tài khoản là admin
  + - * Tiền điều kiện
* Tài khoản phải được đăng nhập
  + - * Hậu điều kiện
* Thông tin các user phải được hiển thị.
  + - * Điểm mở rộng
* Không có.
  + 1. **Mô tả Use Case Đăng Xuất**
* Luồng sự kiện:
  + Luồng cơ bản:

1. UC bắt đầu bằng việc người dùng click vào nút “Đăng xuất”

2. Hệ thống hiển thị thông báo thành công màn hình hiển thị đăng nhập. UC kết thúc.

* + Luồng rẽ nhánh:

1. Tại bất kì thời điểm nào trong quá trình thực hiện UC nếu hệ thống không thể kết nối được với CSDL thì hệ thống sẽ hiển thị thông báo lỗi vào use case kết thúc.

2. Nếu hệ thống không tìm thấy thông tin được nhập giống thông tin trong CSDL thì hệ thống hiển thị thông báo nhập sai tên tài khoản hoặc mật khẩu và yêu cầu nhập lại.

* Các yêu cầu đặc biệt:
* Không có.
* Tiền điều kiện:
* Không có.
* Hậu điều kiện:
* Không có.
* Điểm mở rộng:
* Không có
  1. **Biểu đồ trình tự**
     1. **Use case Đăng nhập**

****

Hình 2 . Biểu đồ trình tự use case Đăng nhập.

* + 1. **Use Case Đăng ký**

****

Hình 3 . Biểu đồ trình tự use case Đăng nhập.

* + 1. **Use Case Xem sản phẩm**

****

Hình 4. Biểu đồ trình tự usecase xem sản phẩm và chi tiết sản phẩm.

* + 1. **Use Case Tìm kiếm sản phẩm**

****

Hình 5. Biểu đồ trình tự usecase tìm kiếm sản phẩm.

* + 1. **Use Case Mua sản phẩm vào giỏ hàng**



Hình 6. Biểu đồ trình tự usecase mua sản phẩm vào giỏ hàng.

* + 1. **Use Case Thêm bình luận**

****

Hình 7. Biểu đồ trình tự usecase thêm bình luận.

* + 1. **Use Case Quản lý sản phẩm**

****

Hình 8. Biểu đồ trình tự usecase quản lý sản phẩm.

* + 1. **Use Case Thêm yêu thích và hiển thị**

****

Hình 9. Biểu đồ trình tự usecase thêm yêu thích.

* + 1. **Use Case Cập nhật đơn hàng**

****

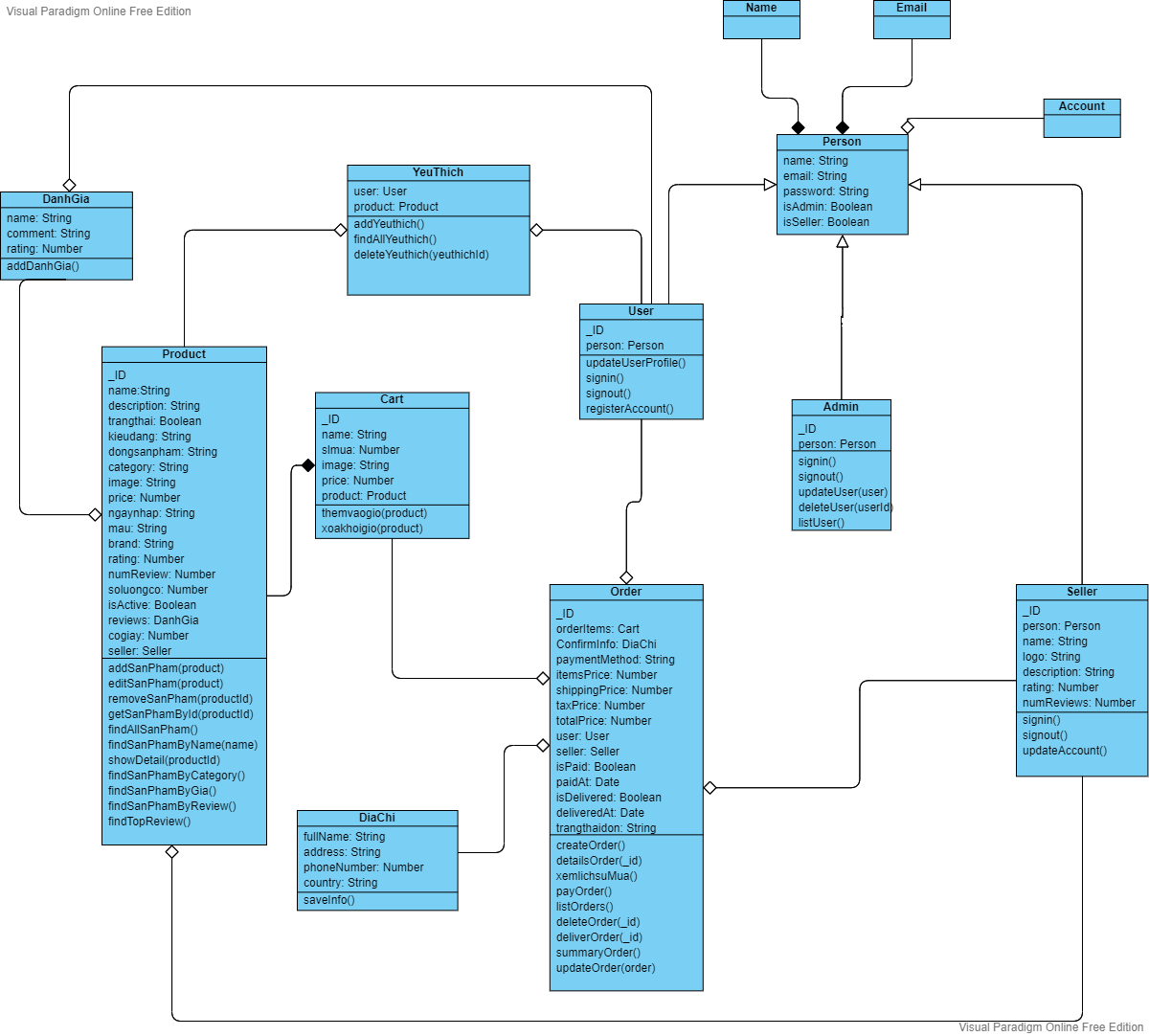
Hình 10. Biểu đồ trình tự usecase quản lý đơn hàng và tình trạng đơn hàng

* + 1. **Use case Thêm đơn hàng**



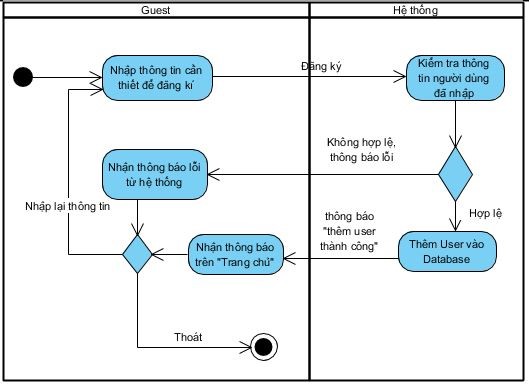
Hình 11. Biểu đồ trình tự usecase thêm đơn hàng.

* 1. **Biểu đồ lớp**

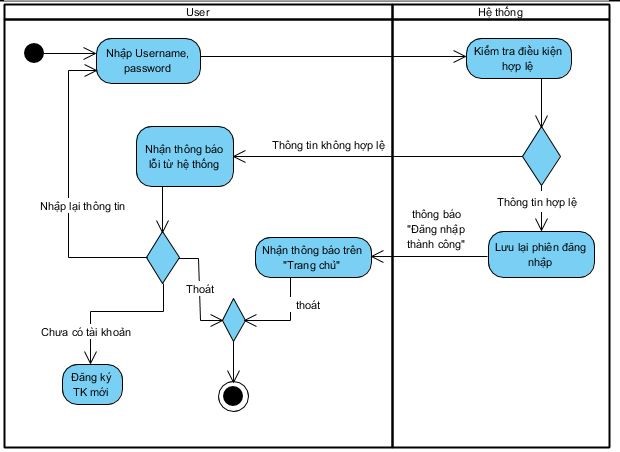
****

Hình 12. Biểu đồ lớp

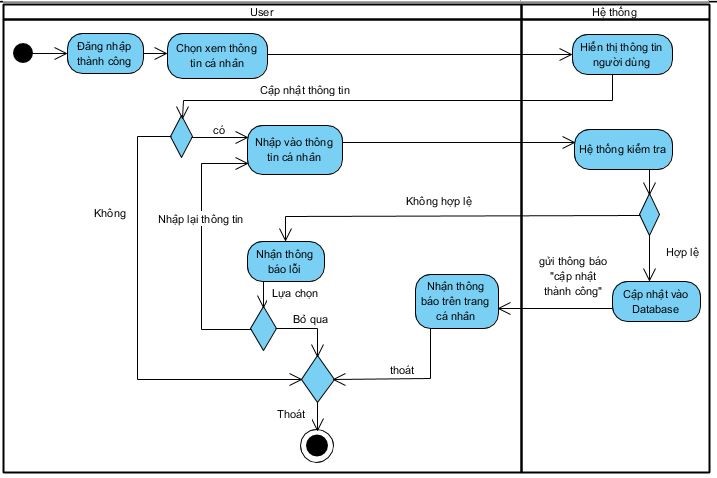
* 1. **Sơ đồ hoạt động**



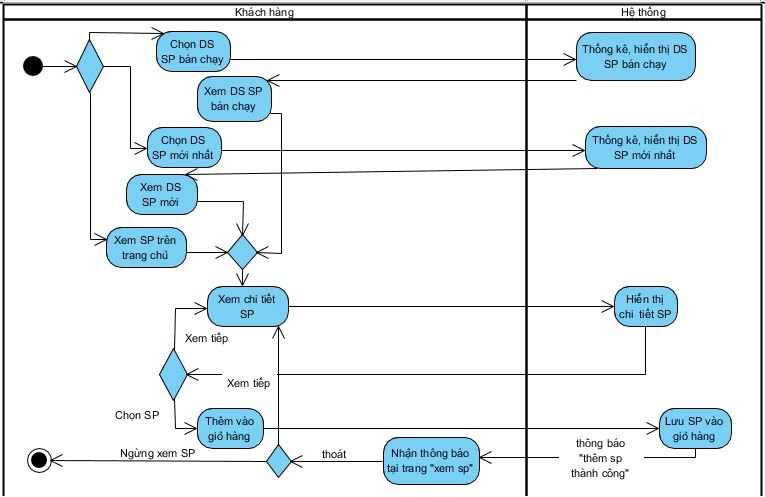
Hình 13. Sơ đồ hoạt động đăng ký tài khoản.



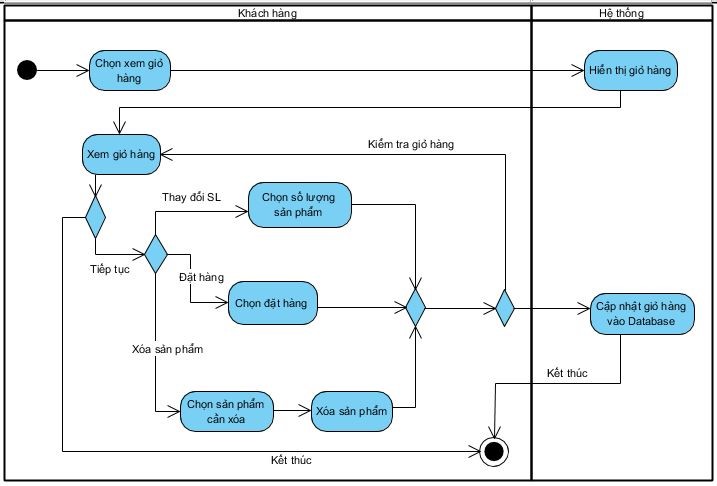
Hình 14. Sơ đồ hoạt động đăng nhập.



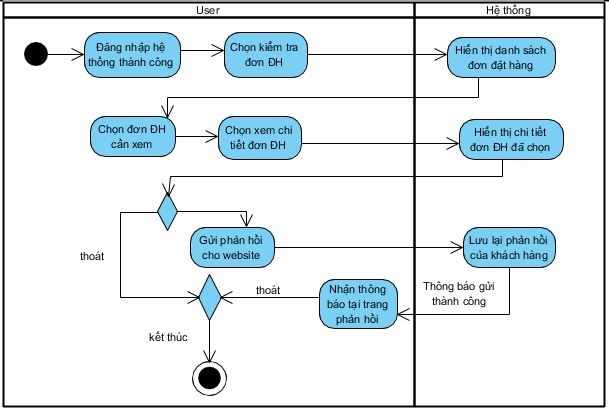
Hình 15. Sơ đồ hoạt động cập nhật thông tin cá nhân.



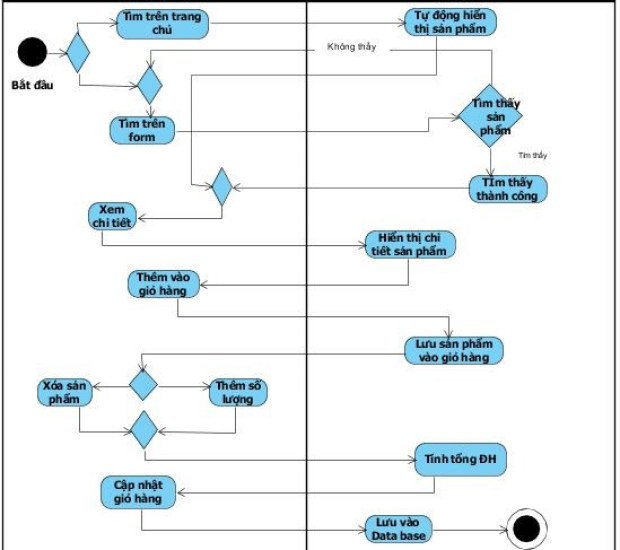
Hình 16. Sơ đồ hoạt động xem sản phẩm và thêm giỏ hàng.



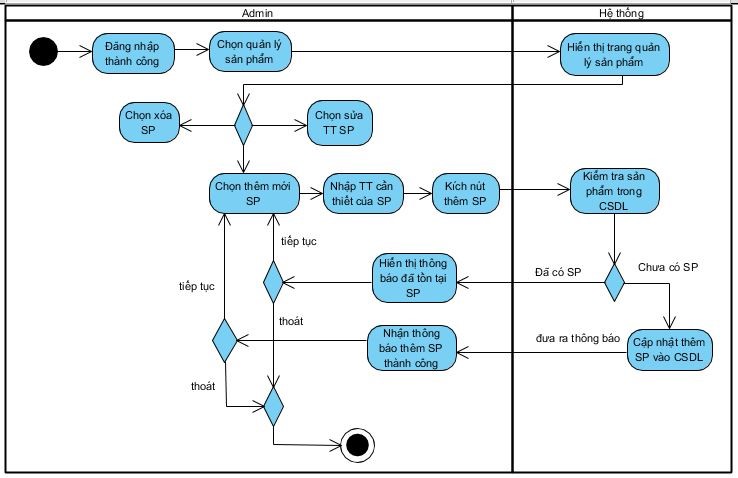
Hình 17. Sơ đồ hoạt động cập nhật thông tin trong giỏ hàng.



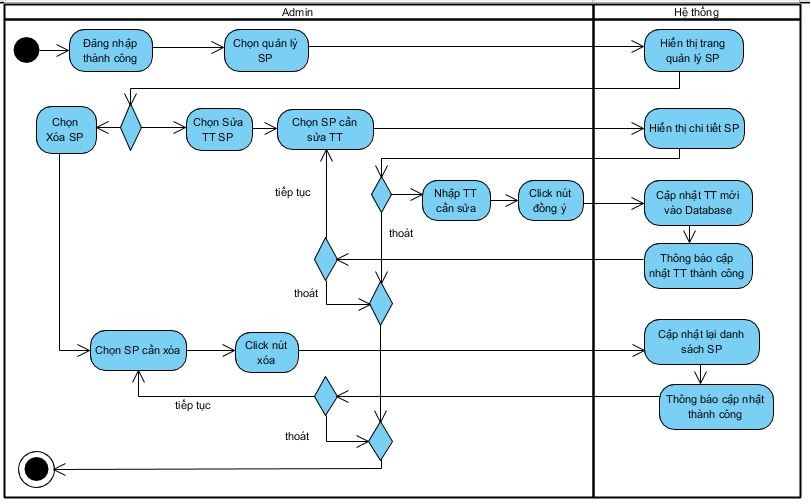
Hình 18. Sơ đồ hoạt động xem lịch sử đặt hàng.



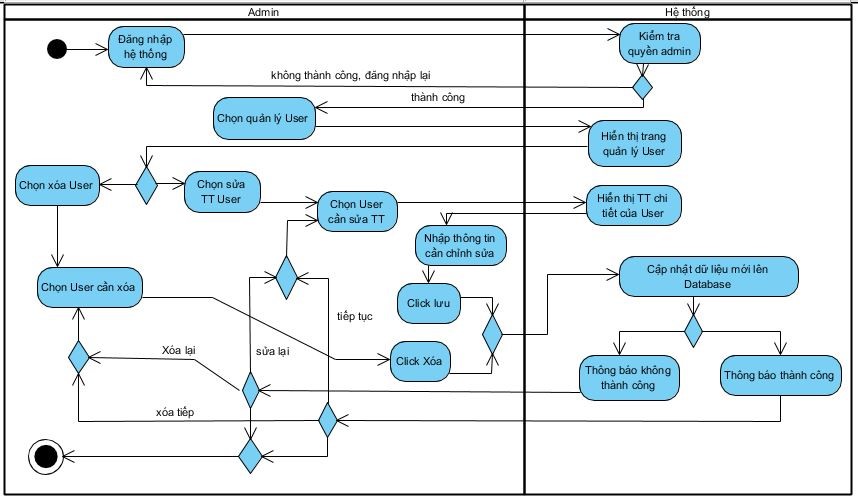
Hình 19. Sơ đồ hoạt động quy trình mua hàng trên website.



Hình 20. Sơ đồ hoạt động quản lý thêm sản phẩm trên website.



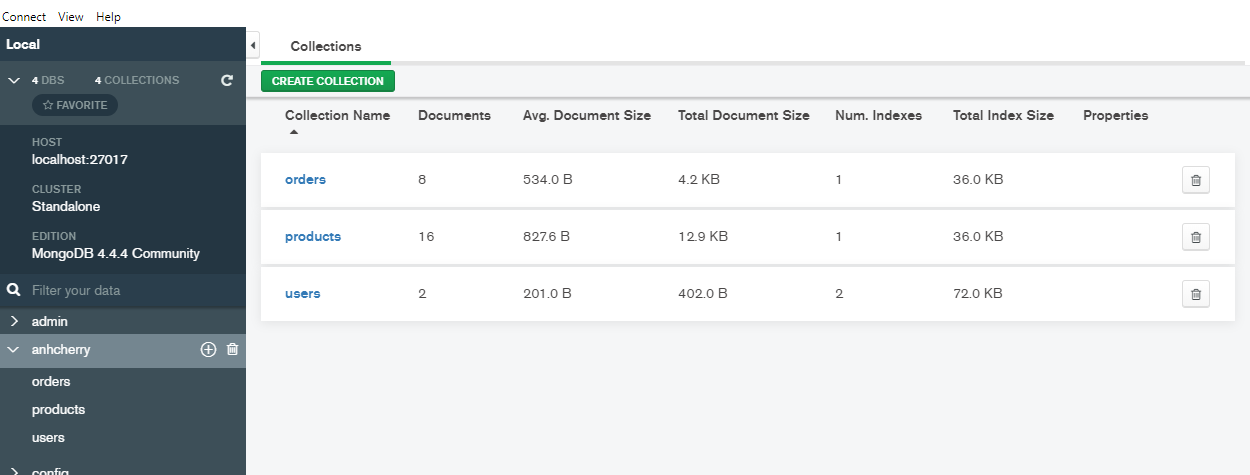
Hình 21. Sơ đồ hoạt động quản lý sửa và xóa sản phẩm .



Hình 22. Sơ đồ hoạt động quản lý tài khoản.

* 1. **Cơ sở dữ liệu**

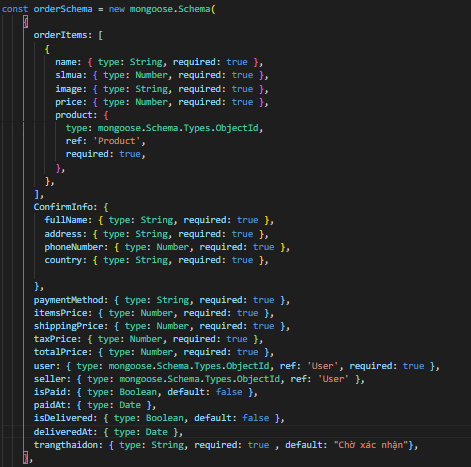
**Các collection:**

****

Hình 23. Biểu đồ các collection

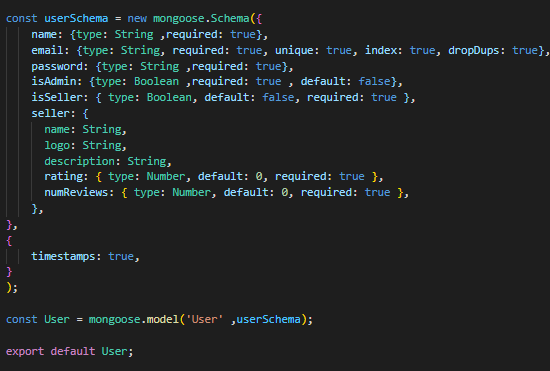
**Các Schema:**

Đây là OrderSchema

****

Hình 24. Biểu đồ order model

Đây là userSchema

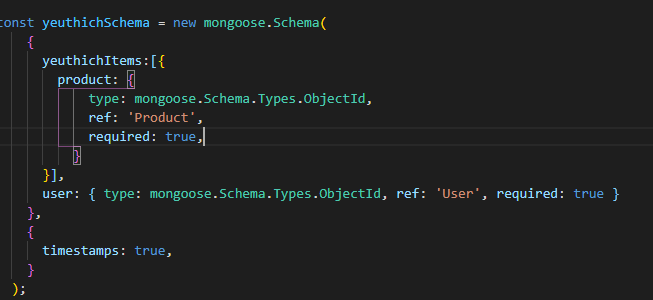


Hình 25. Biểu đồ user model

Đây là productSchema



Hình 26. Biểu đồ product model

****

Hình 27. Biểu đồ yeuthich model

Các chức năng :

User:

















Order:



















Product:













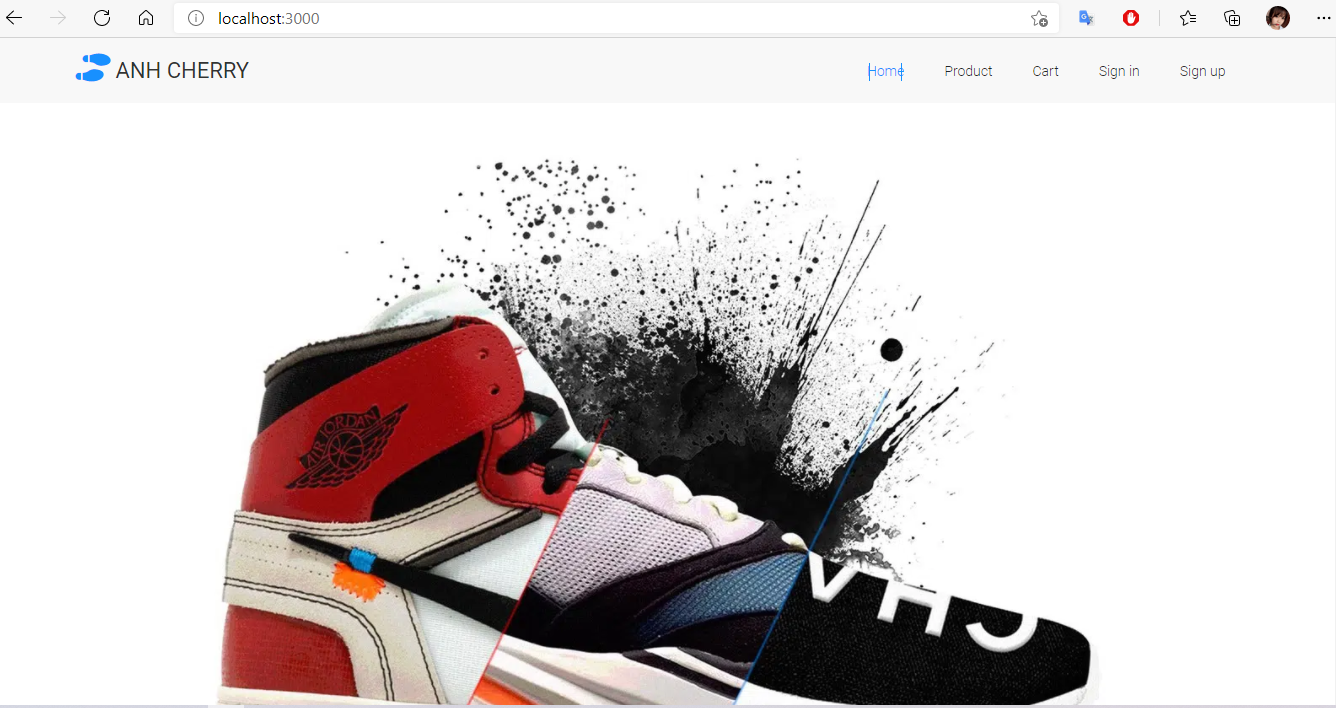






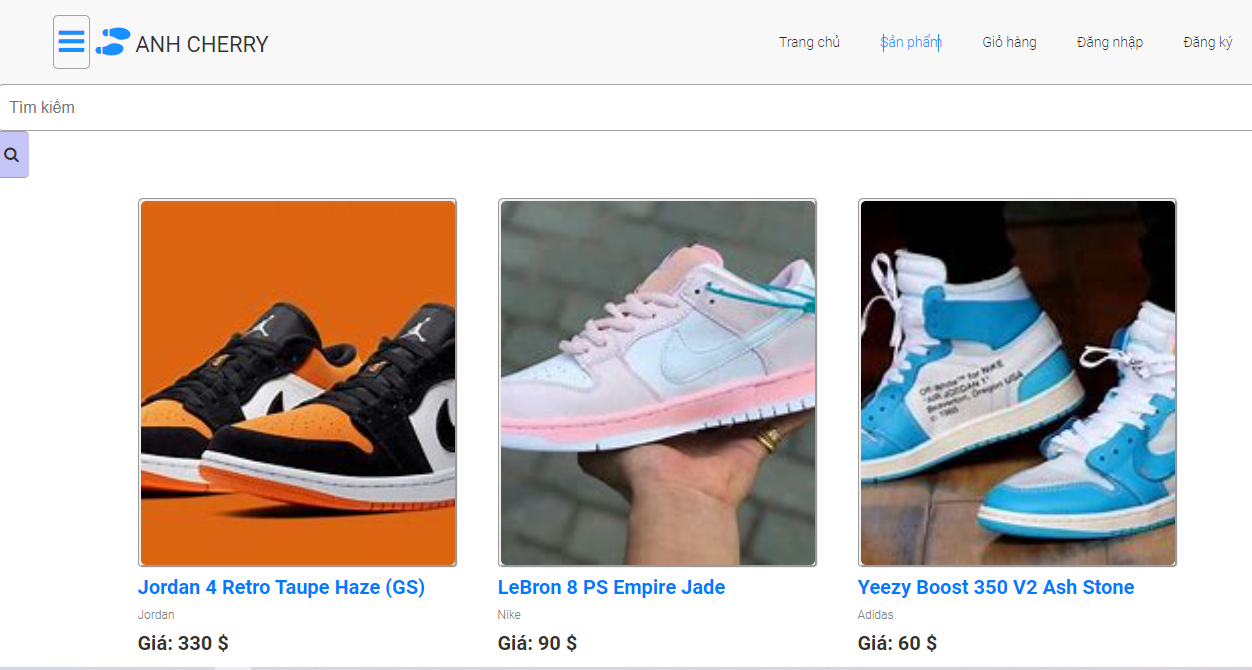
* 1. **Thiết kế một số giao diện chính**

Trang chủ màn hình :



Hình 28. Hình ảnh trang chủ của website

Trang Sản phẩm:



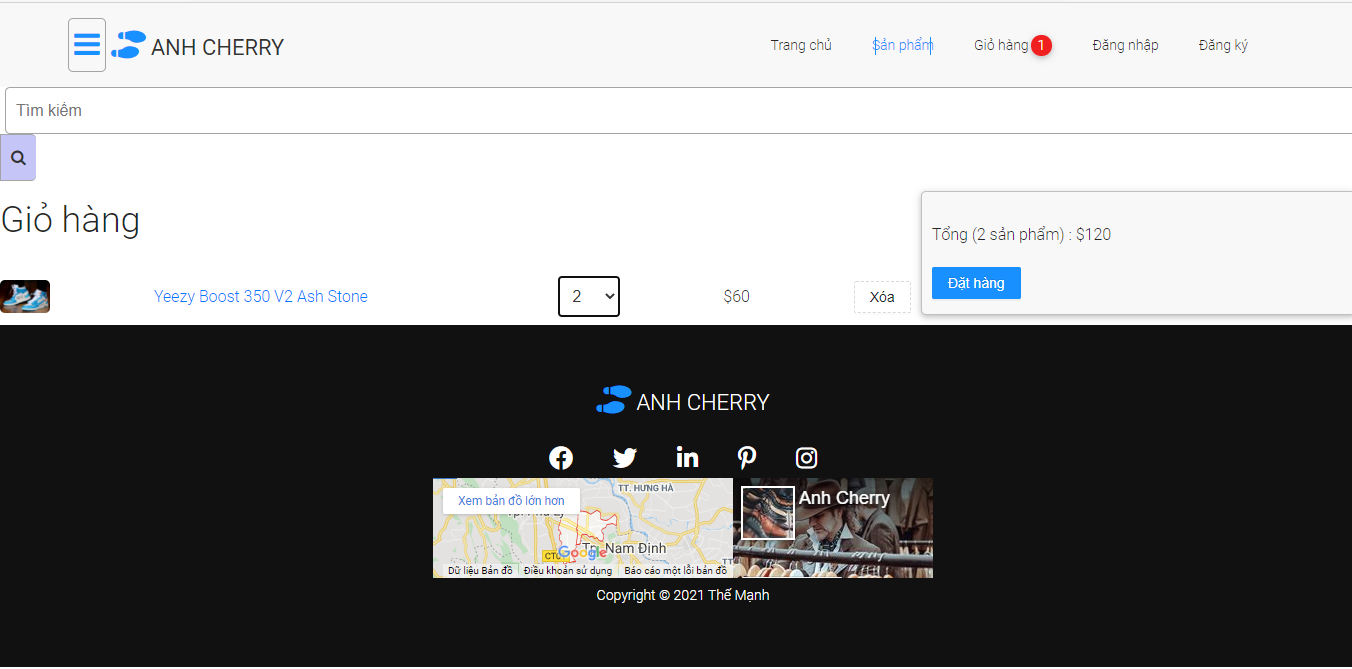
Hình 29. Hình ảnh trang sản phẩm

Trang chi tiết sản phẩm :



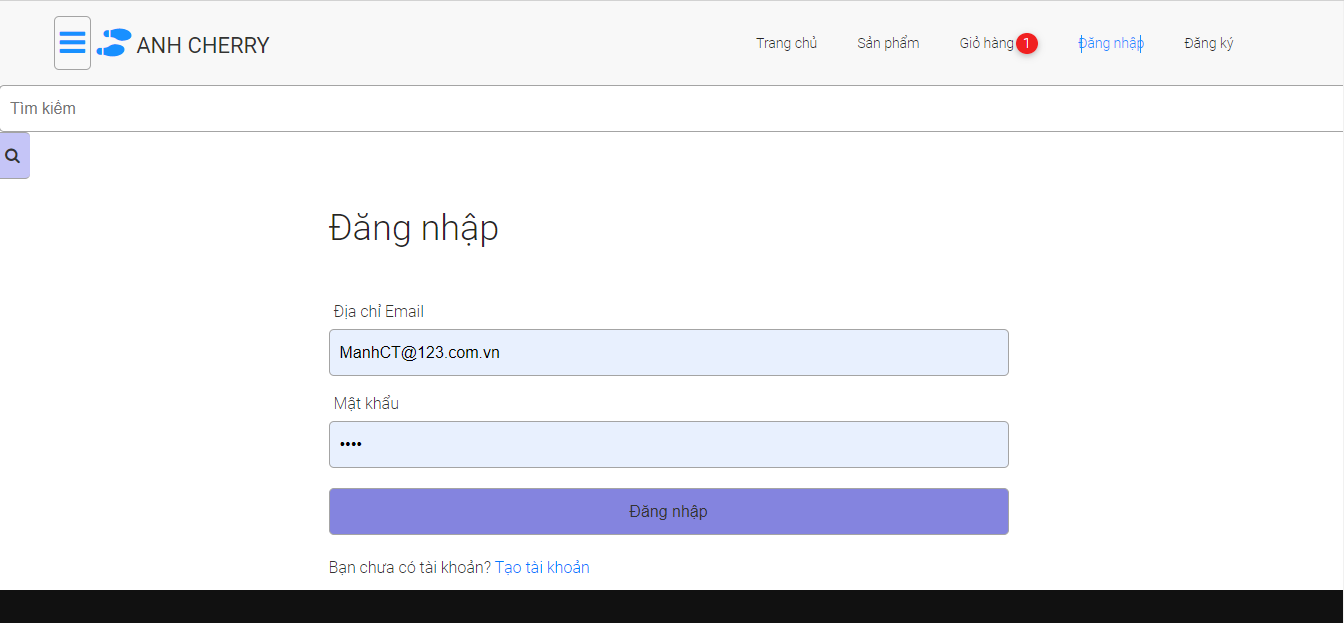
Hình 30. Hình ảnh trang chi tiết sản phẩm

Trang giỏ hàng:



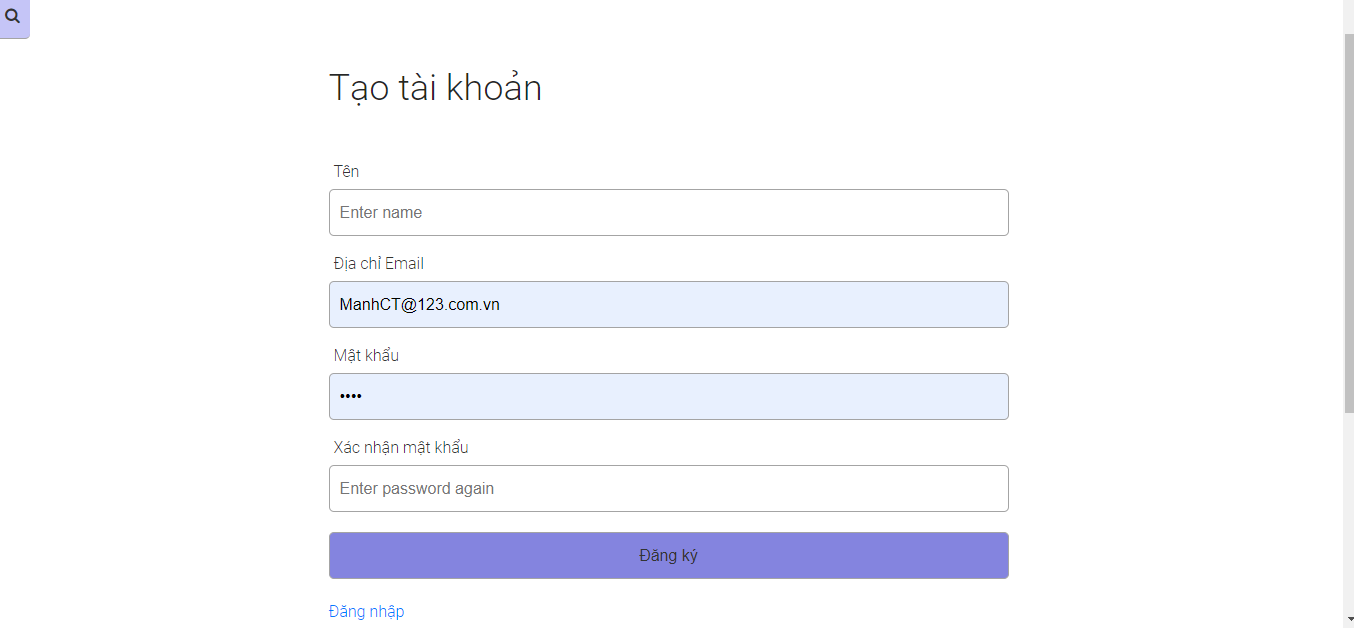
Hình 31. Hình ảnh trang giỏ hàng

Trang đăng nhập:



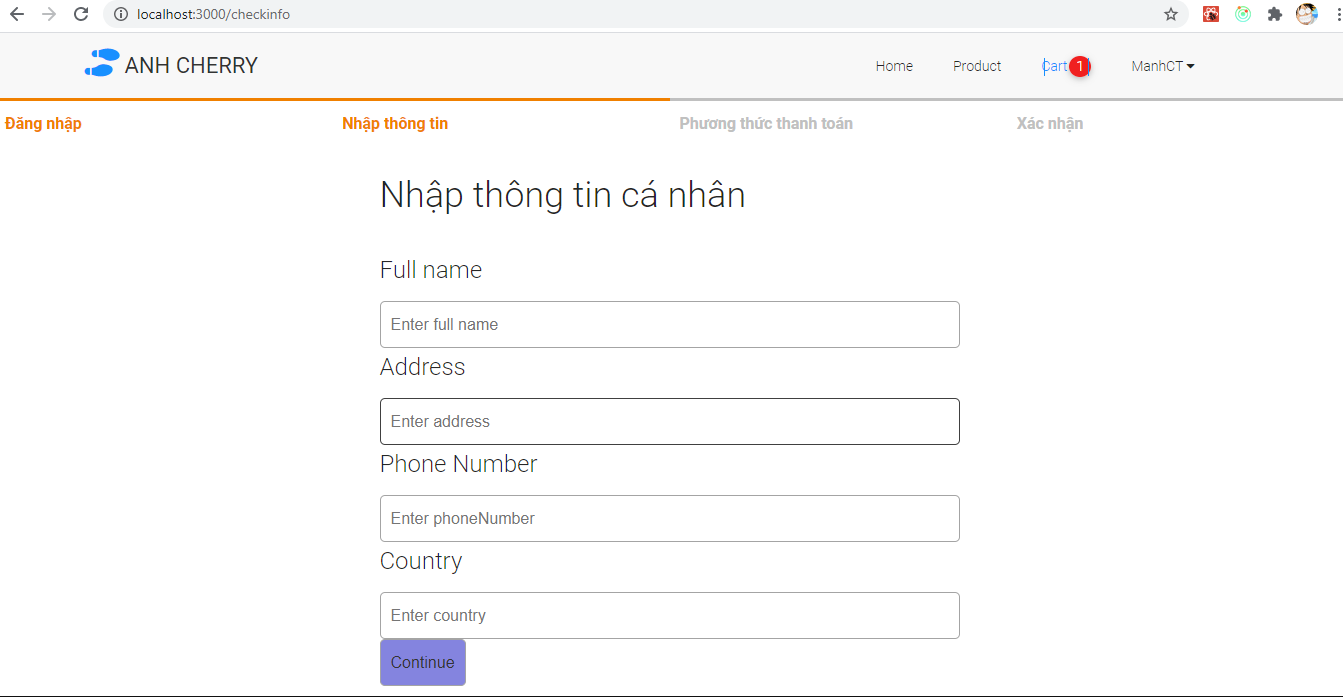
Hình 32. Hình ảnh trang đăng nhập

Trang đăng ký tài khoản:



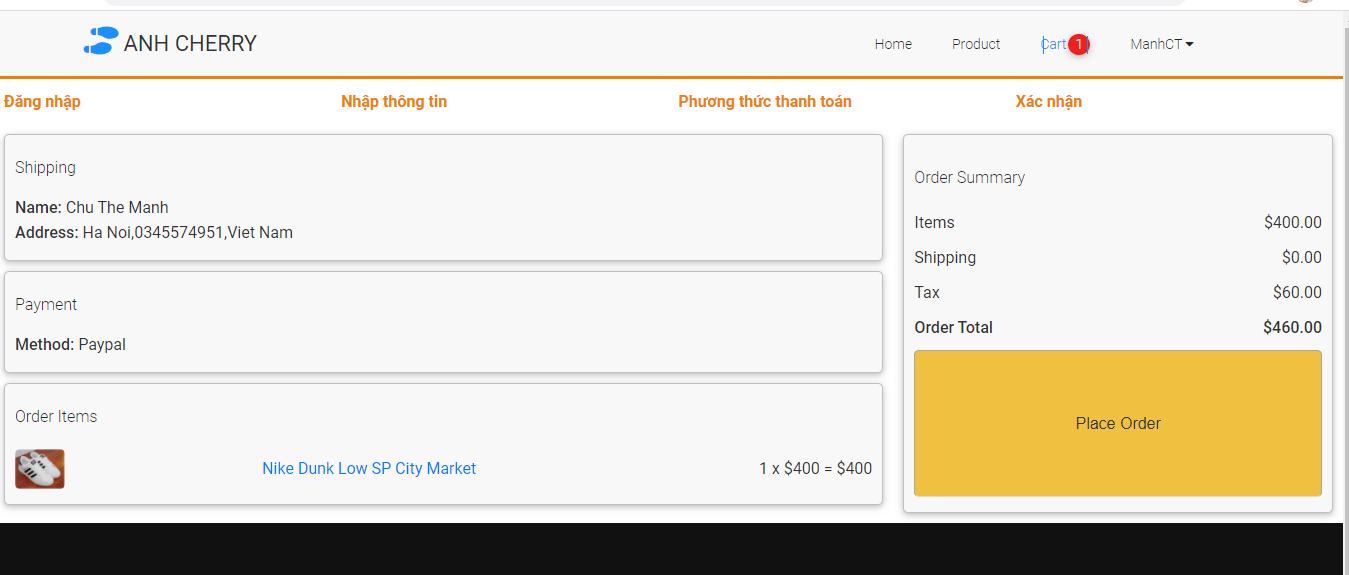
Hình 33. Hình ảnh trang đăng ký tài khoản

Trang nhập địa chỉ shipping:



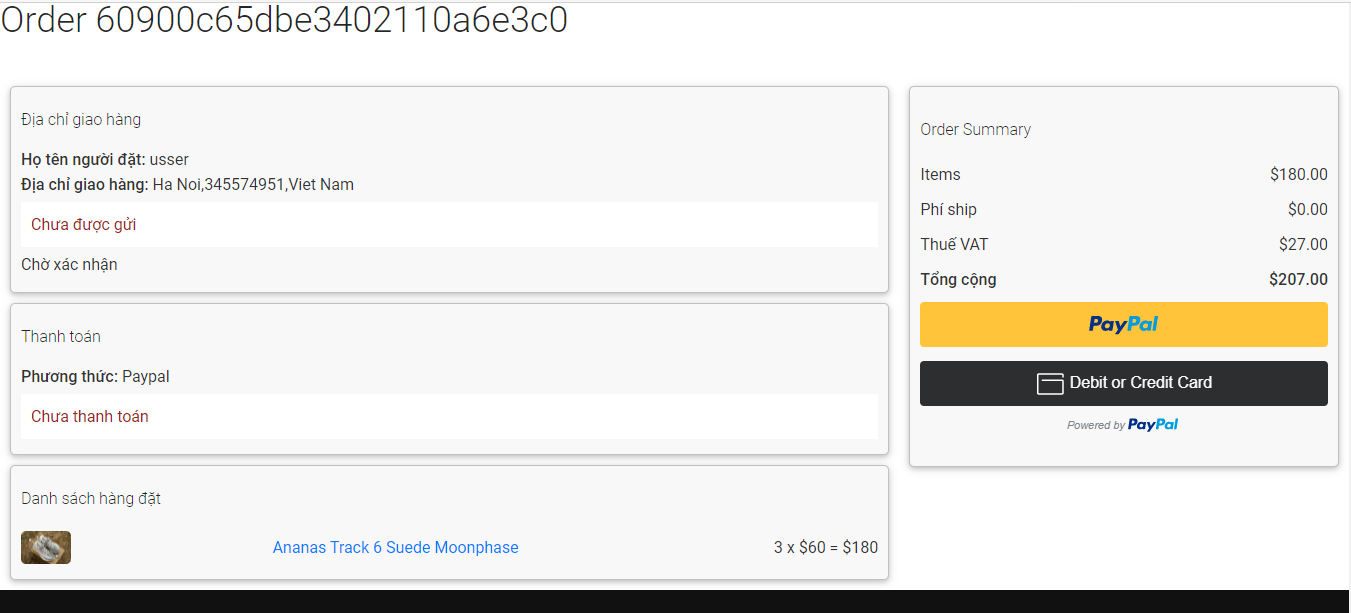
Hình 34. Hình ảnh trang nhập địa chỉ giao hàng

Trang xác nhận mua hàng:

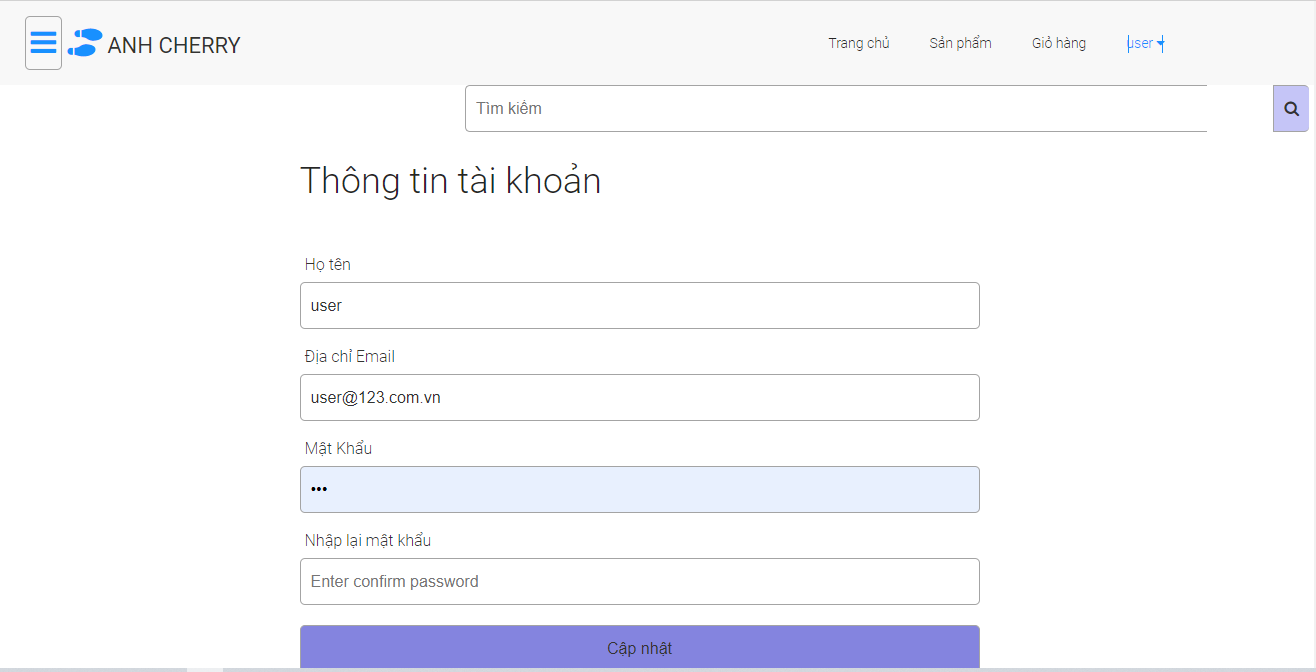


Hình 35. Hình ảnh trang xác nhận mua

Trang xem chi tiết order:



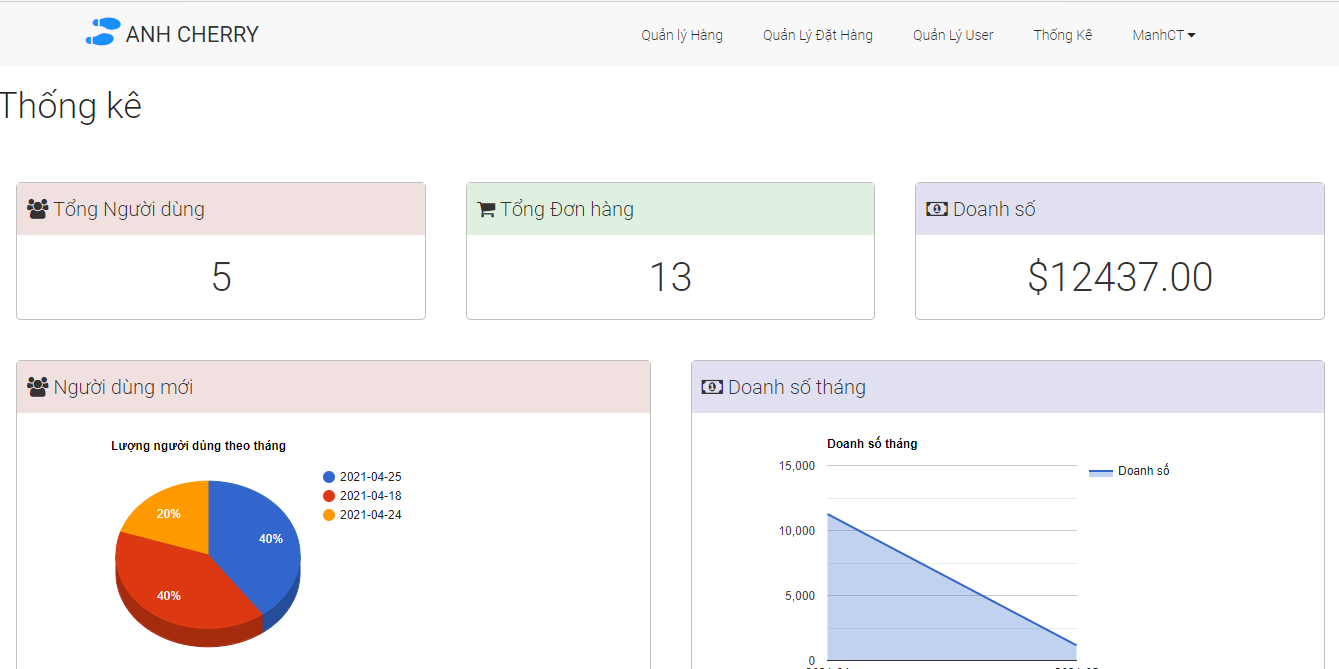
*Hình 36. Hình ảnh trang xem chi tiết order*

**

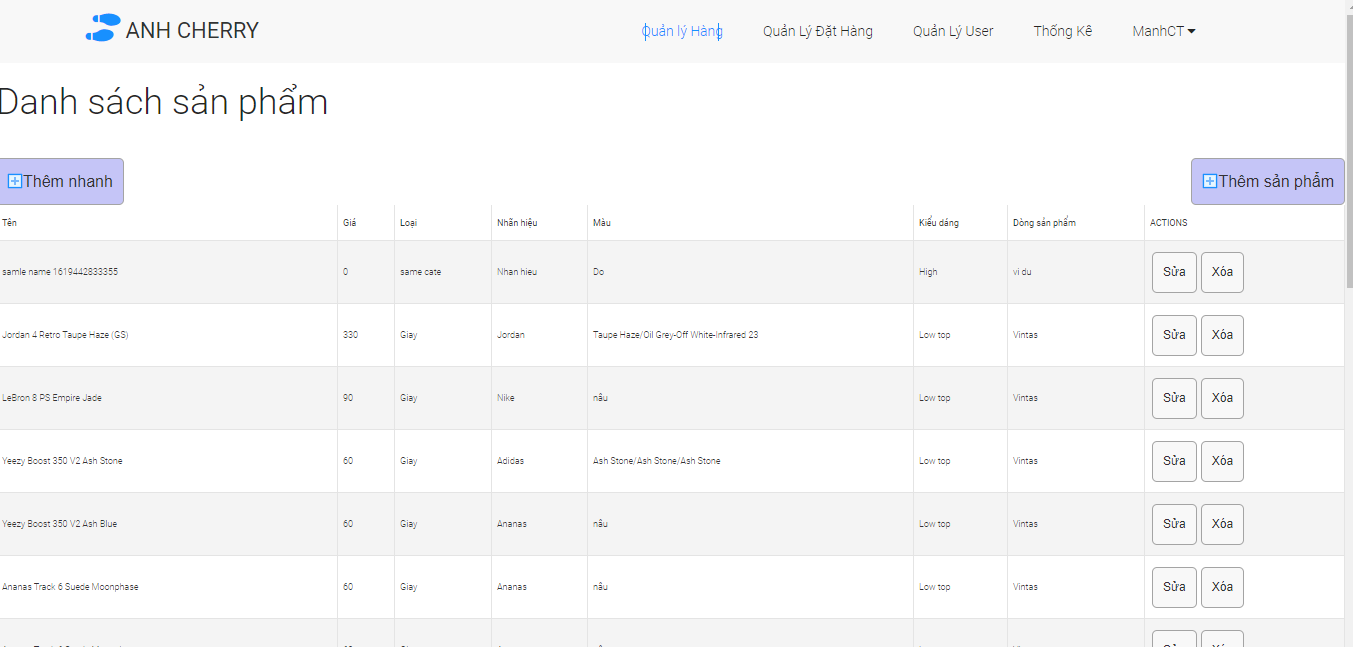
Hình 37. Hình ảnh trang update thông tin tài khoản



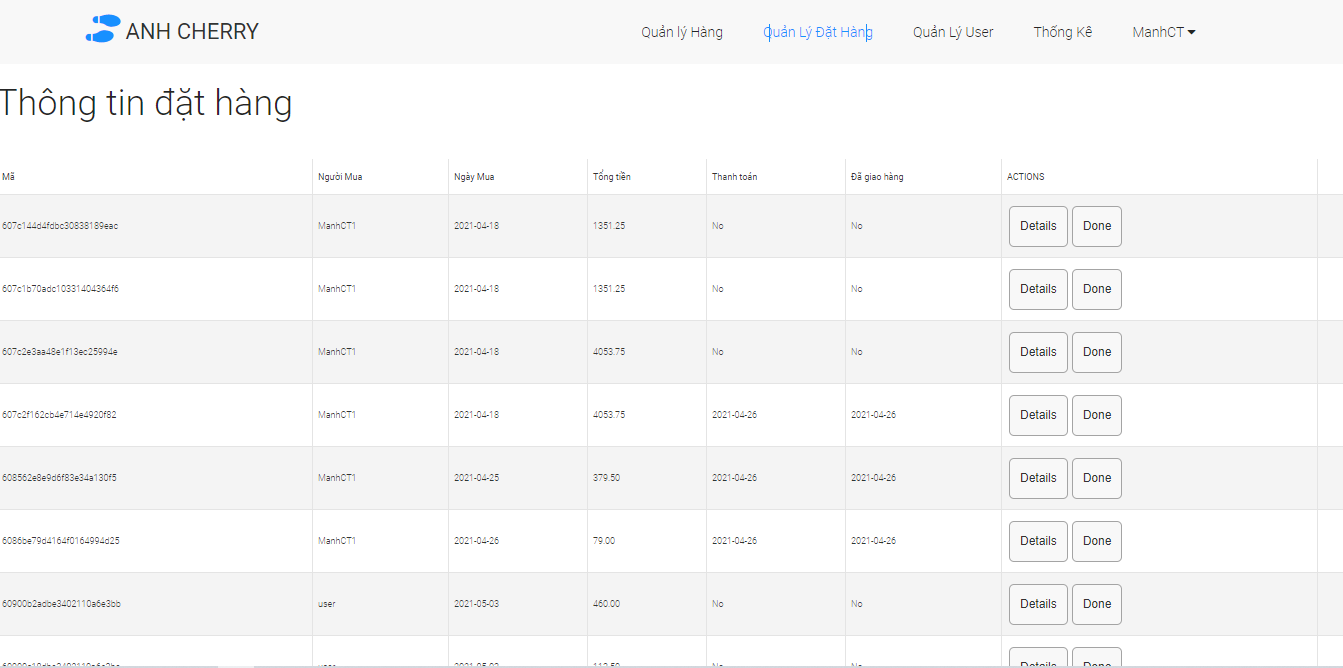
Hình 38. Hình ảnh trang xem lịch sử mua hàng

**

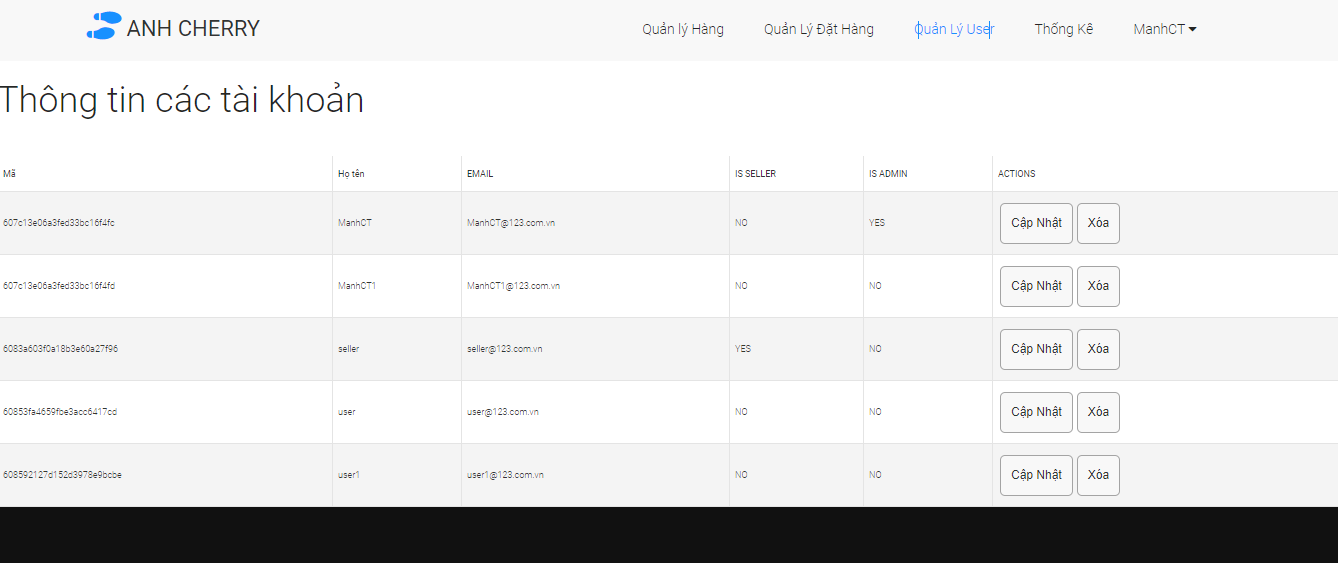
Hình 39. Hình ảnh trang thống kê.

**

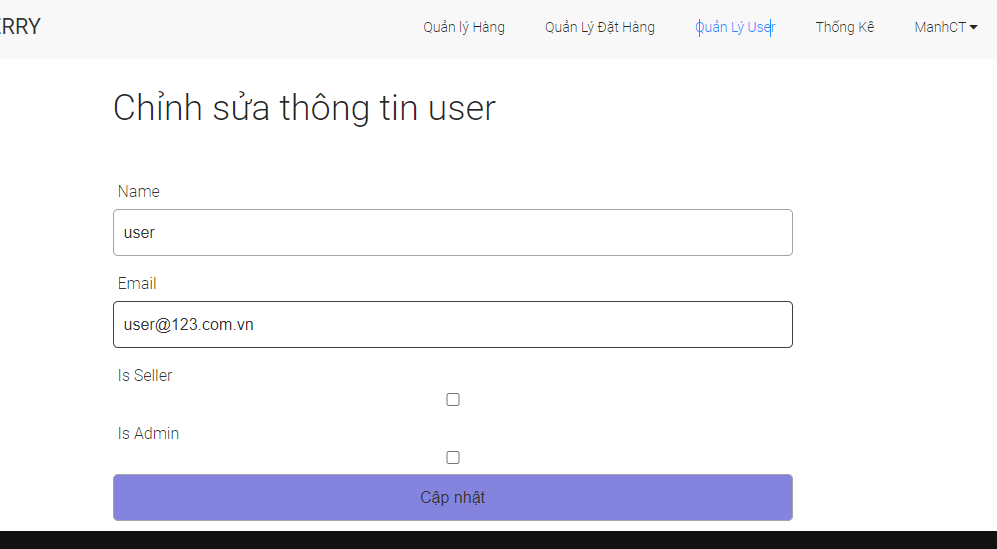
Hình 40. Hình ảnh trang quản lý sản phẩm

**

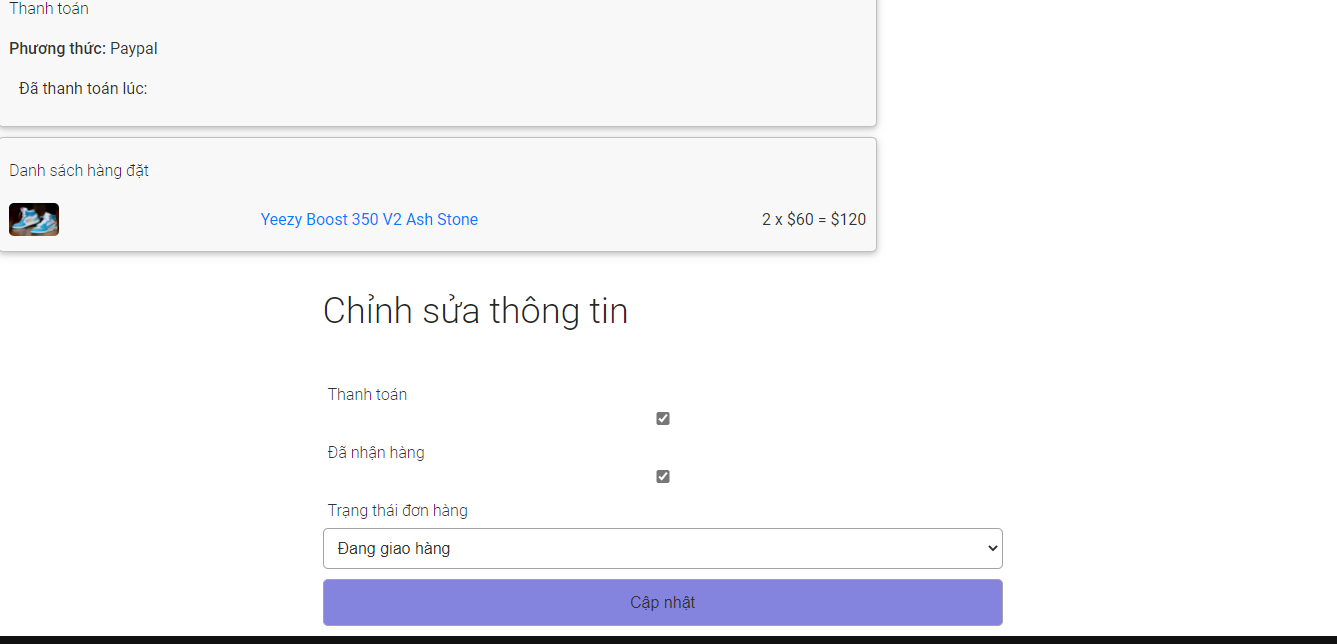
Hình 41. Hình ảnh trang quản lý order

**

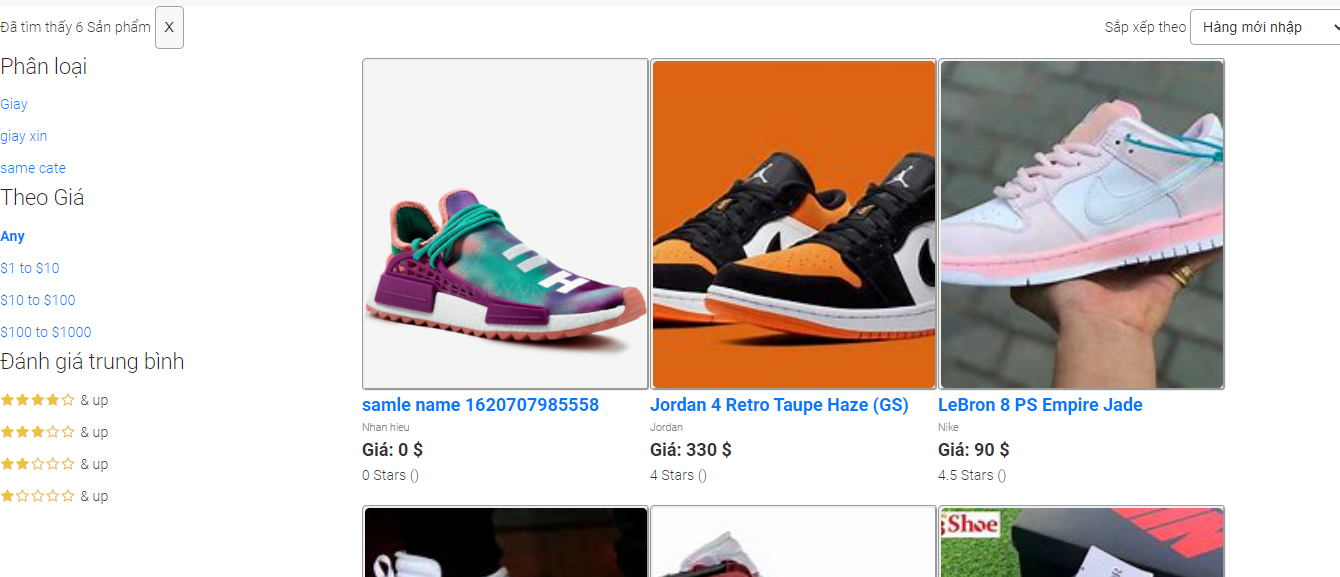
Hình 42. Hình ảnh trang quản lý users

**

Hình 43. Hình ảnh trang update thông tin user và phân quyền tài khoản

**

Hình 44. Hình ảnh trang update trạng thái đơn hàng.

**

Hình 45. Hình ảnh trang tìm kiếm sản phẩm.

# CHƯƠNG 3: CÀI ĐẶT THỬ NGHIỆM

## 3.1. Giới thiệu công nghệ lựa chọn để cài đặt

### 3.1.1. Công nghệ Frontend

Công nghệ chính em sử dụng trong project để làm UI đó chính là ReactJs - một thư viện rất hot hiện nay của lập trình javascript.

**React** (Hay **ReactJS, React.js**) là một thư viện **Javascript** mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi **Jordan Walke**, một kỹ sư phần mềm tại **Facebook**. Người bị ảnh hưởng bởi **XHP** (Một nền tảng thành phần **HTML** cho **PHP**). **React** lần đầu tiên được triển khai cho ứng dụng **Newsfeed** của **Facebook** năm 2011, sau đó được triển khai cho **Instagram.com** năm 2012. Nó được mở mã nguồn (open-sourced) tại **JSConf US** tháng 5 năm 2013.

Lý do tại sao chọn ReactJs:

* Nó cho phép lập trình viên tạo ra các component tương ứng với các phần của giao diện . Chúng có thể tái sử dụng , kết hợp với các component khác để tạo ra nhiều giao diện khác nhau.
* Có nhiều thư viện hỗ trợ làm UI do đó hiệu quả và chất lượng cũng cao hơn ( sử dụng thư viện Ant Design thiết kết nhanh hơn so với code thuần html ,css).
* Tạo ra ứng dụng web single page application – website chỉ cần 1 trang duy nhất để xử lý tất cả tính năng . Nó có hiệu quả về tốc độ load trang , mượt mà hơn với người dùng , BE và FE ở đây tách riêng ra rõ ràng.

Cách cài đặt một chương trình ReactJs:

Vào trình Visual Studio Code bật terminal và nhập :

* Create-react-app ten\_app
* Cd ten\_app
* Npm start

Như thế là ta đã tạo được một ứng dụng ReactJs và chạy chúng .Tất cả các components nhỏ sẽ được import và load vào trong file App.js

### 3.1.2. Công nghệ Backend

**NodeJs**: NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Qua phần tìm hiểu **NodeJS là gì** mình có giới thiệu một đặc tính rất quan trọng đó là Realtime, tuy nhiên vẫn còn khá nhiều đặc tính mà bạn cần phải biết trước khi học NodeJS.

* **Không đồng bộ**: Tất cả các API của NodeJS đều không đồng bộ (none-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* **Chạy rất nhanh**: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* **Đơn luồng nhưng khả năng mở rộng cao**: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* **Không đệm**: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* **Có giấy phép**: NodeJS đã được cấp giấy phép bởi [MIT License](https://raw.githubusercontent.com/joyent/node/v0.12.0/LICENSE).

Em sử dụng framework của NodeJs đó là Express để xây dựng Api

Bước đầu phải import express vào app backend và setting trong file package.json

"express": "^4.16.3",

Sau đó tạo file server.js và bắt đầu tạo một app express

const app = express();

app.use(express.json());

app.use(express.urlencoded({ extended:true}));

từ đó ta có thể sử dụng app để use or get data từ API.

**JWT:**

**Token-based** **authentication** là phương thức xác thực bằng chuỗi má hóa. Một hệ thống sử dụng **Token-based** authentication cho phép người dùng nhập user/password để nhận về 1 chuỗi token. Chuỗi Token này được sử dụng để “xác minh” quyền truy cập vào tài nguyên mà không cần phải cung cấp lại username/password nữa.

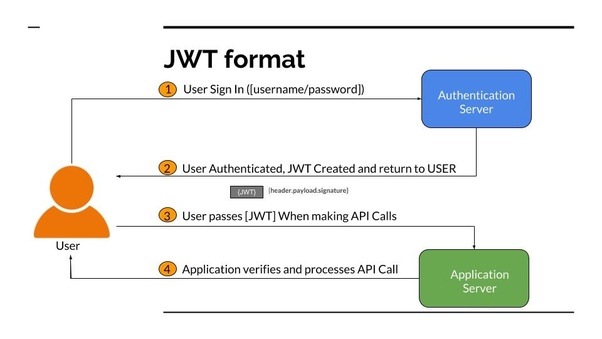
**JSON Web Token** là một chuỗi mã hóa mà nguồn gốc ban đầu là một chuỗi **JSON**. Chuỗi thông tin dạng JSON bằng phương pháp mã hóa nào đó, nó trở thành 1 chuỗi ký tự lộn xộn nhìn vào sẽ rất khó hiểu.

**JWT** bao gồm 3 phần:

* **Header**
* **Payload**
* **Signature**

**Flow hệ thống sử dụng JWT**

Chúng ta sẽ dùng sở đồ sau để hình dung

*Hình 46. Biểu đồ JWT format.*

**Nhìn vào hình ta có thể thấy flow đi như sau:**

1. User thực hiện login bằng cách gửi id/password hay sử dụng các tài khoản mạng xã hội lên phía Authentication Server (Server xác thực)
2. Authentication Server tiếp nhận các dữ liệu mà User gửi lên để phục vụ cho việc xác thực người dùng. Trong trường hợp thành công, Authentication Server sẽ tạo một JWT và trả về cho người dùng thông qua response.
3. Người dùng nhận được JWT do Authentication Server vừa mới trả về làm “chìa khóa” để thực hiện các “lệnh” tiếp theo đối với Application Server.
4. Application Server trước khi thực hiện yêu cầu được gọi từ phía User, sẽ verify JWT gửi lên. Nếu OK, tiếp tục thực hiện yêu cầu được gọi.

**Hệ thống Verify chuỗi JWT thế nào?**

Câu hỏi đặt ra ở đây là hệ thống Verify JWT thế nào:

* Chuỗi JWT có cấu trúc **H.P.S** được **Client** gửi lên. **Server** sẽ làm tương tự như sau
  + Set S1 = S
  + Set S2 = **HMAC**(SHA256(**H.P**) vỡi secret key của hệ thống) (Giả sử hệ thống sử dụng encryption algorithms SHA256)
  + So sánh S1 == S2 ?
* Nếu S1 và S2 khớp nhau, tức là chữ ký hợp lệ, hệ thống mới tiếp decode payload và tục kiểm tra các data trong payload. Ví dụ trường exp (expiration date).

**Khi nào sử dụng JWT?**

Với JWT, bạn không cần phải giữ session data trên server để xác thực user. Luồng đi như sau:

* + Người dùng gọi authentication service để gửi username/password
  + Authentication service phản hồi cho người dùng mã JWT, cái này sẽ định nghĩa xem user là ai
  + Người dùng yêu cầu truy cập một dịch vụ được bảo mật bằng việc gửi token lên
  + Lớp bảo mật sẽ check chữ ký trên token và nếu đó là quyền truy cập hợp lệ thì được tiếp tục truy cập

Các sessions sẽ có thời hạn hết hạn và cần phải được xử lý kiểu xoá đi các session hết hạn này. JWT hoàn toàn có thể sở hữu chính expiry date của chính nó kèm với dữ liệu user. Cho nên khi tầng Security check authen của JWT, nó có thể check expiry time của token và đơn giản là từ chối truy cập.

Nếu không sử dụng session thì bạn mới có thể ứng dụng tạo một service thuần RESTful, bởi vì một service thuần RESTful được định nghĩa là phải [stateless](https://topdev.vn/blog/stateless-la-gi-stateful-la-gi/). Với dung lượng nhỏ, JWT có thể được gửi lên với mọi request cũng giống như session cookie. Nhưng ko giống với session cookie, nó ko cần phải trỏ đến bất kì dữ liệu nào được lưu trữ trên server, bản thân JWT đã có dữ liệu.

## 3.2. Cài đặt các chức năng chính của hệ thống

- Đã cài đặt được trang web hiển thị phía khách hàng , cho phép khách hàng có thể xem sản phẩm chọn và thêm vào giỏ hàng , mua hàng , đăng nhập , đăng xuất tài khoản , đăng ký tài khoản, tìm kiếm theo các tiêu chí, thanh toán bằng Paypal, cập nhật thông tin tài khoản , thêm đánh giá .

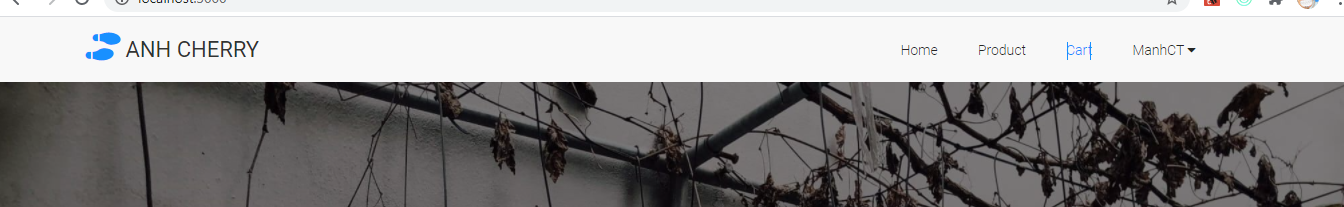
- Phía admin đã có quản lý product, user, order, thống kê .

- Cho phép thanh toán online bằng Paypal.

## 3.3. Một số kịch bản thử nghiệm

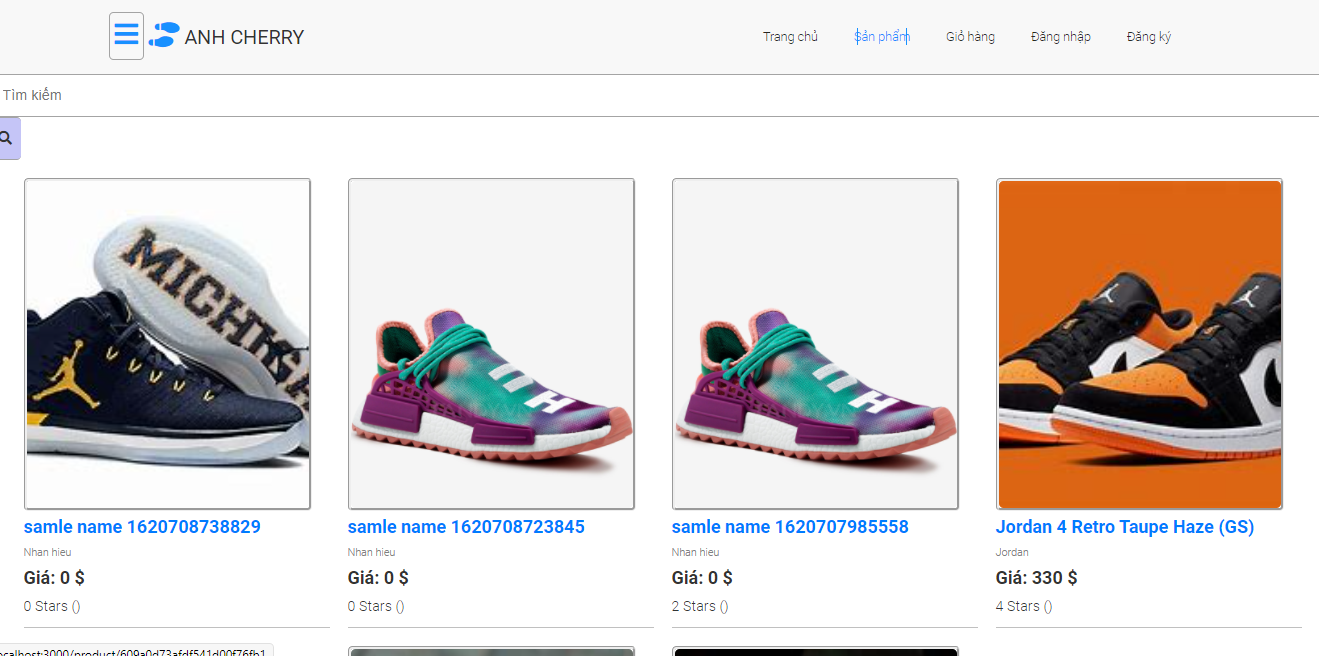
### 3.3.1. Kịch bản mua hàng

Người dùng click vào trang web trên header có các thao tác product sign in , sign up,cart .Có thể đăng nhập trước hoặc xem sản phẩm trước



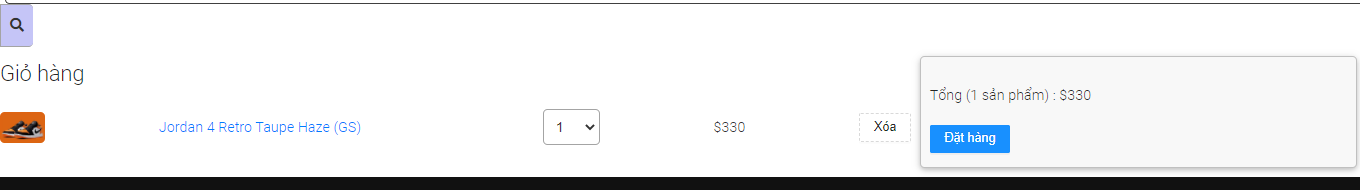
Hình 47. Hình ảnh header của user.

Khi ta click vào product màn hình sẽ hiển thị trang Product show ra các sản phẩm cho ta chọn .



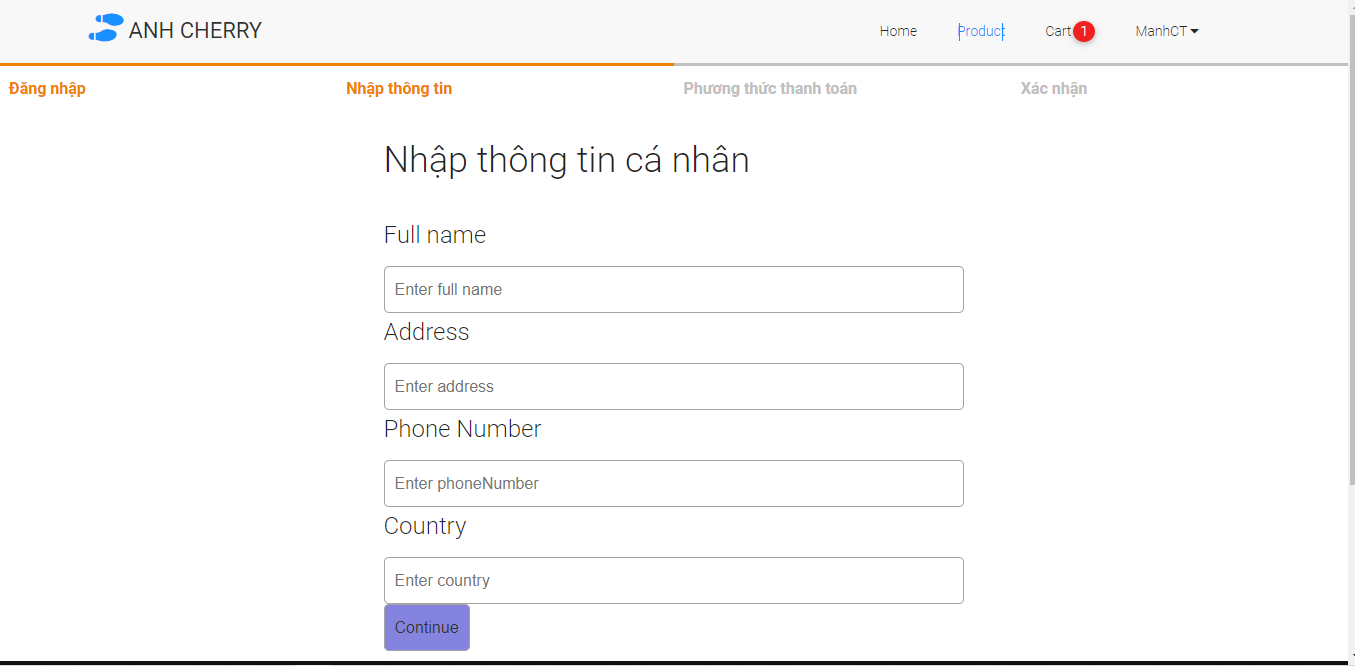
Hình 48. Hình ảnh trang sản phẩm.

Ta sẽ click vào ảnh or tên để xem chi tiết sản phẩm . Nếu người dùng mua thì click vào button “Add to cart” nếu k mua có thể click “Back to result” để trở về trang chủ.



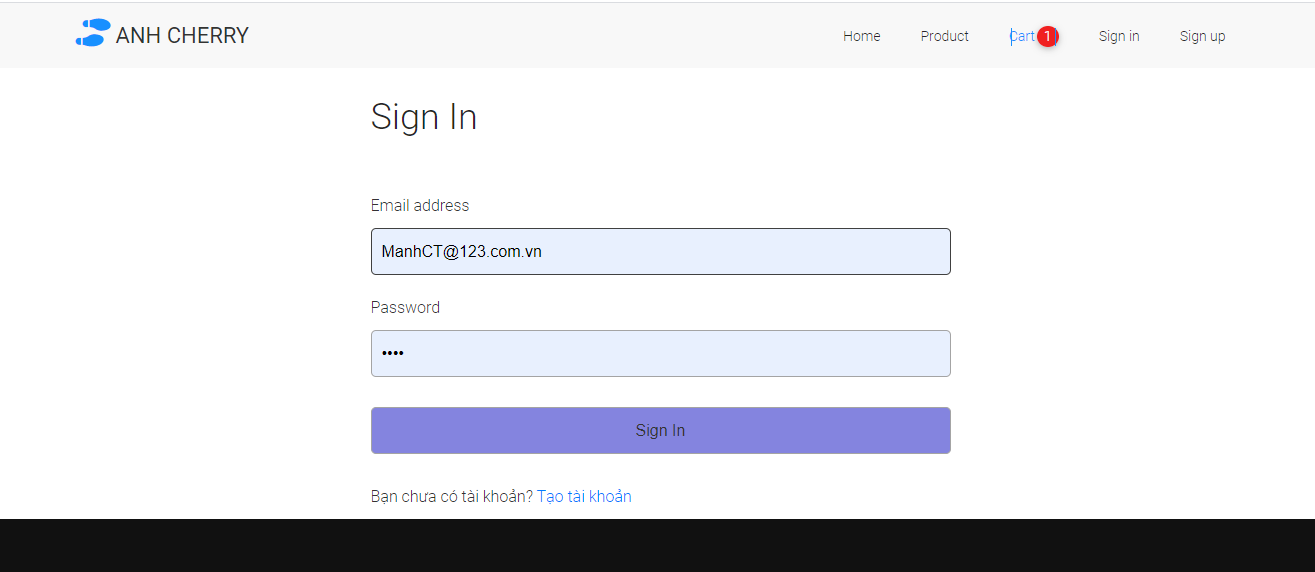
Hình 49. Hình ảnh trang giỏ hàng.

Đây là trang cart . Người dùng có thể chỉnh lại số lượng lần cuối ở đây và ấn đặt hàng.Có thể ấn ra trang product để tiếp tục thêm sản phẩm vào giỏ hàng.Sau khi ấn “Đặt hàng” nếu người dùng đã đăng nhập rồi thì sẽ chuyển đến trang xác nhận thông tin shipping



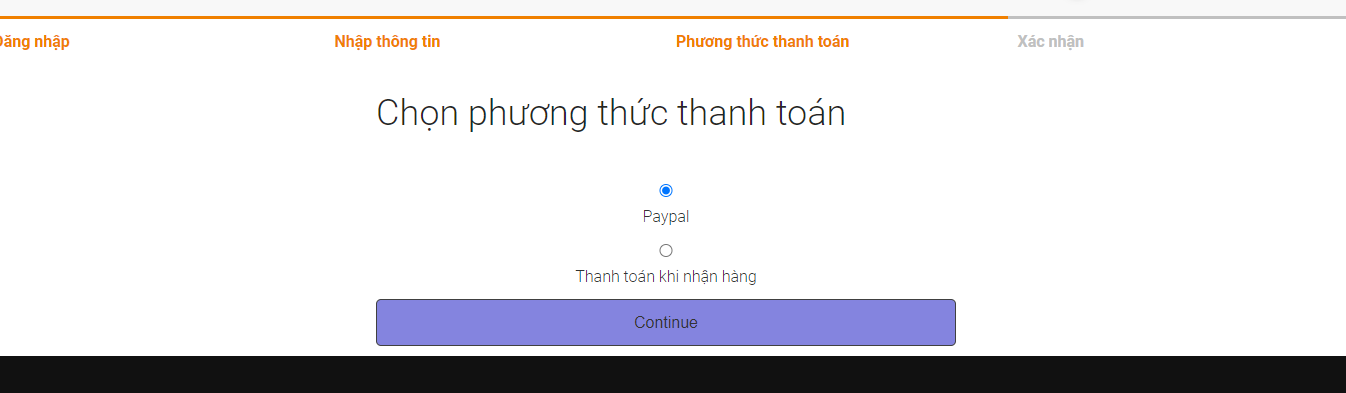
Hình 50. Hình ảnh trang điền thông tin địa chỉ nhận.

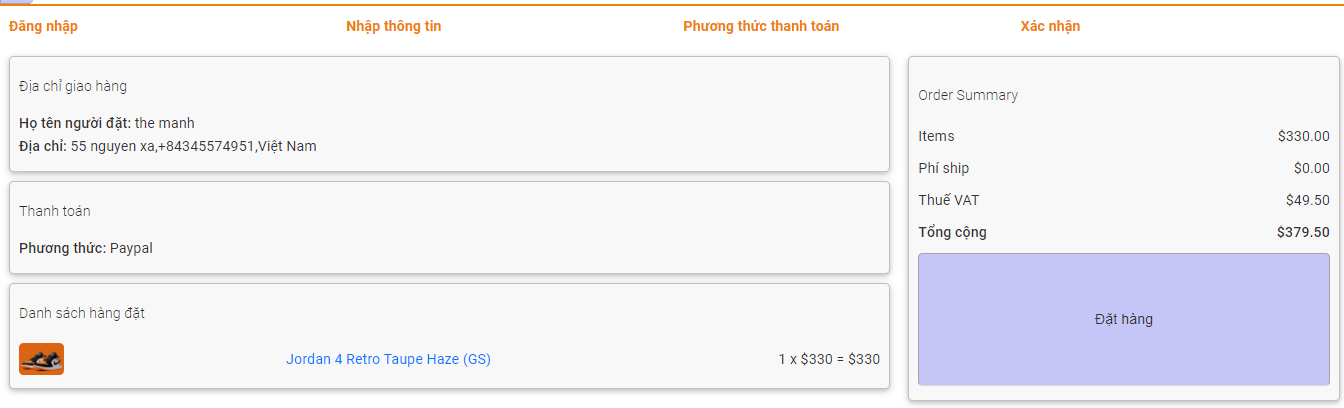
Nếu chưa đăng nhập thì sẽ chuyển đến trang Sign in



Hình 51. Hình ảnh trang đăng nhập.

Nếu chưa có tài khoản có thể click vào “Tạo tài khoản” trang sẽ đưa bạn đến giao diện tạo tài khoản.Khi click button mua sẽ chuyển đến chọn phương thức thanh toán



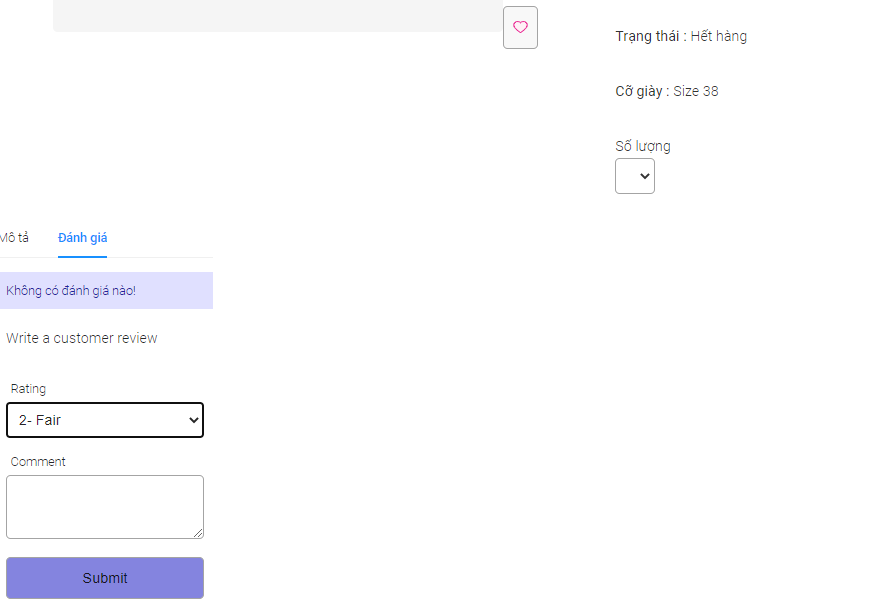


Hình 52. Hình ảnh trang order và method thanh toán.

Sau đó sẽ xác nhận lần cuối các khoản chi phí cuối cùng để thanh toán với khách hàng . Kết thúc quy trình mua hàng

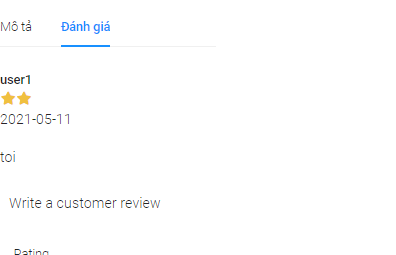
* + 1. **Kịch bản gửi đánh giá**

Muốn gửi đánh giá trang web phải yêu cầu bạn đăng nhập. Trước tiên ta đăng nhập bằng tài khoản của mình vào. Trong trang xem chi tiết sản phẩm ta lăn xuống vùng đánh giá.



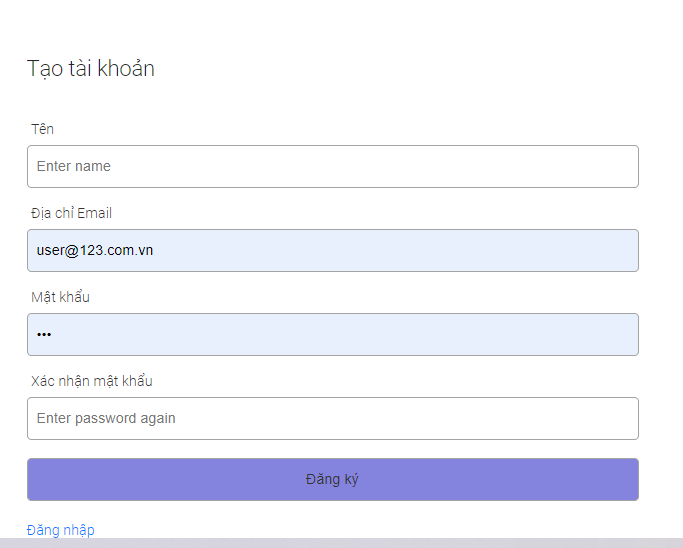
Hình 53. Hình ảnh trang đánh giá.

Trang sẽ hiển thị ra các đánh giá trước đó và ô input để mình nhập đánh giá. Người dùng chọn số sao và nhập comment vào rồi ấn gửi lên server. Như vậy là hoàn thành đánh giá sản phẩm.Trang web sẽ hiển thị ra thông báo thành công và hiển thị đánh giá đó lên.



* + 1. **Kịch bản đăng ký tài khoản**

Khi truy cập vào trang web để đăng ký tài khoản, người dùng sẽ click vào “Đăng ký” để tạo tài khoản. Trang web sẽ đưa đến trang register, người dùng sẽ nhập các thông tin cần có trên màn hình.



Hình 54. Hình ảnh trang đăng ký tài khoản.

Người dùng có thể ấn đăng nhập để trở lại trang đăng nhập.Khi đăng ký thành công trang web sẽ tự động đăng nhập và chuyển về trang home cho người dùng.

# KẾT LUẬN

Trong thời gian thực tập, thực hiện đề tài, em đã nghiên cứu, học được cách sử dụng, vận dụng ReactJS và NodeJS trong công việc việc thiết kế lên một trang web. Tiếp cận được với môi trường doanh nghiệp một cách thực tế, tiếp thu được thêm kinh nghiệm làm việc nhóm và ý thức được độ quan trọng của mỗi công việc mình được giao. Biết được quy trình làm việc tại doanh nghiệp trong quá trình thực tập.

***Kết quả đạt được:***

* Xây dựng được Website bán giày nữ áp dụng công nghệ ReactJS và NodeJS , MongoDB.
* Nắm vững hơn về html , css, js
* Hiểu được cách xây dựng website
* Có kiến thức cơ bản về công nghệ ReactJS và NodeJS
* Phân tích thiết kế hệ thống tương đối đầy đủ.
* Giao diện Website được thiết kế đơn giản, thân thiện và dễ sử dụng

***Hạn chế:***

* Chưa hoàn thành được hết tất cả các chức năng đặt ra
* Tìm hiểu, nghiên cứu còn chậm chạp do làm việc cá nhân.

***Hướng phát triển tương lai:***

* Hoàn thiện các chức năng còn lại của một website e-commerce
* Nâng cấp lên để tránh bị quá tải và mất cắp thông tin
* Có thể ứng dụng thực tế luôn cho gia đình
* Hoàn thiện giao diện Frontend thu hút hơn.
* Responsive trên mọi thiết bị và đưa vào sử dụng được

# TÀI LIỆU THAM KHẢO

1. ReactJs <https://reactjs.org/docs/getting-started.html>
2. Nodejs <https://nodejs.org/en/docs/>
3. Redux <https://redux.js.org/introduction/getting-started>
4. React-router <https://reactrouter.com/web/guides/quick-start>
5. ExpressJS <https://expressjs.com/en/4x/api.html>

Youtube <https://www.youtube.com/channel/UC2xRE4hUCQ3xO3ymEtMh1Hw>

1. **Github – basir :** [basir/amazona: Build Ecommerce Like Amazon By MERN Stack (github.com)](https://github.com/basir/amazona)
2. **Ant Design :** [Ant Design - The world's second most popular React UI framework](https://ant.design/)
3. **Youtube-Nghiepuit:** [1 . Lập Trình ReactJS : Giới Thiệu ( ReactJS Introduction ) - YouTube](https://www.youtube.com/watch?v=gL5HBA_1bDQ&list=PLJ5qtRQovuEOoKffoCBzTfvzMTTORnoyp&ab_channel=nghiepuit)
4. **Stackoverflow:** [Stack Overflow - Where Developers Learn, Share, & Build Careers](https://stackoverflow.com/)
5. **Medium**