TRƯỜNG ĐẠI HỌC YERSIN ĐÀ LẠT

**KHOA CÔNG NGHỆ THÔNG TIN**

****



**BÁO CÁO ĐỒ ÁN**

**MÔN HỌC: PHÁT TRIỂN DỰ ÁN ANGULARJS**

**TÊN ĐỀ TÀI: SHOP FASHION WEBSITE**

|  |
| --- |
| **GVHD:** Th.S Thái Thuận Thương |
| **SVTH :** Võ Ngọc Khánh |
| **MSSV :** 12001048 |

Đà Lạt, ngày 23 tháng 6 năm 2023

**LỜI MỞ ĐẦU**

Trong thời đại công nghệ 4.0 hiện nay, việc mua sắm online đang trở thành một xu hướng phổ biến và không thể phủ nhận sức lan tỏa của nó trong đời sống hàng ngày. Với sự phát triển của công nghệ thông tin và internet, mua sắm online đã trở nên tiện lợi và đem lại nhiều lợi ích cho người tiêu dùng.

Bên cạnh đó, nhiều framework đã ra đời nhằm mục đích hỗ trợ việc xây dựng website hiệu quả, trong đó có AngularJS framework. Mặc dù Angular đã ra mắt và trở thành phiên bản tiếp theo của AngularJS, tuy nhiên, việc nghiên cứu và học tập AngularJS vẫn mang lại nhiều giá trị quan trọng.

AngularJS, dù đã lỗi thời nhưng vẫn là một công cụ phát triển web mạnh mẽ và được sử dụng rộng rãi trong quá khứ. Việc tìm hiểu về AngularJS giúp chúng ta có được cái nhìn tổng quan về quá trình phát triển ứng dụng web và hệ thống front-end. Nó cung cấp một cách tiếp cận cấu trúc ứng dụng dựa trên các thành phần có thể tái sử dụng, tăng tính khả maintainability và khả năng sử dụng lại mã nguồn. Việc nắm vững những khái niệm và kỹ thuật này sẽ giúp chúng ta dễ dàng tiếp cận và hiểu rõ hơn về Angular và các framework phát triển web khác.

Với đề tài môn học này, em quyết định áp dụng kiến thức đã học về AngularJS để xây dựng một trang web bán hàng online thực tế. Mục tiêu của em là tận dụng các công nghệ hiện đại và kỹ thuật phát triển phần mềm để tạo ra một trải nghiệm mua sắm trực tuyến tốt nhất cho người dùng và đóng góp vào việc phát triển ứng dụng CNTT trong đời sống hàng ngày.

**NHẬN XÉT CỦA GIẢNG VIÊN**

|  |
| --- |
| **Chữ ký giảng viên nhận xét** |
| (Ký và ghi rõ họ và tên ) |

**Mục lục**

[Danh mục hình 6](#_Toc138142619)

[Chương 1: Giới thiệu 9](#_Toc138142620)

[1.1. Đặt vấn đề 9](#_Toc138142621)

[1.2. Tính cấp thiết của đề tài 9](#_Toc138142622)

[1.3. Mục tiêu nghiên cứu 9](#_Toc138142623)

[Chương 2: Cơ sở lý thuyết 10](#_Toc138142624)

[2.1. Angular Js framework 10](#_Toc138142625)

[2.1.1. Giới thiệu tổng quan về AngularJs 10](#_Toc138142626)

[2.1.2. Các tính năng chung của AngularJs 10](#_Toc138142627)

[2.1.3. Các tính năng cốt lõi của AngularJs 11](#_Toc138142628)

[2.1.4. Một số ưu điểm và nhược điểm của AngularJS: 12](#_Toc138142629)

[2.1.5. Những Components trong AngularJS 12](#_Toc138142630)

[2.2. Spring boot framework 13](#_Toc138142631)

[2.3. Spring security 14](#_Toc138142632)

[2.4. Restful API 16](#_Toc138142633)

[2.5. Mô hình MVC 16](#_Toc138142634)

[Chương 3: Xây dựng đồ án 17](#_Toc138142635)

[3.1. Phân tích yêu cầu chức năng 17](#_Toc138142636)

[3.2. Phân tích nghiệp vụ 18](#_Toc138142637)

[3.3. Xây dựng DB 23](#_Toc138142638)

[3.4. Kết quả thực nghiệm 24](#_Toc138142639)

[Chương 4: Kết luận và hướng phát triển 31](#_Toc138142640)

[4.1. Kết luận 31](#_Toc138142641)

[4.2. Hướng phát triển 31](#_Toc138142642)

[Tài liệu tham khảo 32](#_Toc138142643)

# **Danh mục hình**

[Hình 1: Các thành phần chính của AngularJs 11](#_Toc138142572)

[Hình 2: Cấu trúc và work flow của Spring boot 14](#_Toc138142573)

[Hình 3:Luồng hoạt động của spring security 14](#_Toc138142574)

[Hình 4: Luồng hoạt động của Spring security + JWT 15](#_Toc138142575)

[Hình 5: Sơ đồ mô tả qui trình đặt hàng 21](#_Toc138142576)

[Hình 6: Sơ đồ mô tả qui trình quản lý đơn hàng 22](#_Toc138142577)

[Hình 7: Sơ đồ Usecase tổng quan yêu cầu chức năng 22](#_Toc138142578)

[Hình 8: Giao diện trang chủ 24](#_Toc138142579)

[Hình 9: Tìm kiếm sản phẩm 24](#_Toc138142580)

[Hình 10: Xem chi tiết sản phẩm 25](#_Toc138142581)

[Hình 11: Đăng nhập 25](#_Toc138142582)

[Hình 12: Đăng ký tài khoản 26](#_Toc138142583)

[Hình 13: Xem giỏ hàng 26](#_Toc138142584)

[Hình 14: Trang giỏ hàng 27](#_Toc138142585)

[Hình 15: Trang thanh toán 27](#_Toc138142586)

[Hình 16: Quản lý đơn hàng đã đặt 28](#_Toc138142587)

[Hình 17: Quản lý thể loại sản phẩm 28](#_Toc138142588)

[Hình 18: Quản lý sản phẩm 29](#_Toc138142589)

[Hình 19: Quản lý đơn hàng 29](#_Toc138142590)

[Hình 20: Xem chi tiết đơn hàng 30](#_Toc138142591)

[Hình 21: Cập nhật trạng thái đang giao hàng 30](#_Toc138142592)

[Hình 22: Hủy đơn hàng và nêu lý do 30](#_Toc138142593)

**Danh mục các bảng**

[Bảng 1: Danh mục từ viết tắt 8](#_Toc138099678)

[Bảng 2: Bảng các yêu cầu 19](#_Toc138099679)

[Bảng 3: Bảng các qui định 20](#_Toc138099680)

[Bảng 4: Bảng các yêu cầu về chất lượng 21](#_Toc138099681)

**Danh mục từ viết tắt**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Từ viết tắt** | **Tên đầy đủ** | **Mô tả** |
| 1 | HTML | HyperText Markup Language | Ngôn ngữ đánh dấu siêu văn bản |
| 2 | CSS | Cascading Style Sheets | Ngôn ngữ định dạng kiểu trang web |
| 3 | DOM | Document Object Model | Mô hình đối tượng tài liệu |
| 4 | DB | Database | Cơ sở dữ liệu |
| 5 | MVC | Model-View-Controller | Mô hình MVC giúp phân chia rõ ràng và tách biệt giữa dữ liệu, giao diện và logic xử lý trong ứng dụng, giúp dễ dàng bảo trì, mở rộng và thay đổi các thành phần một cách độc lập. |
| 6 | MVW | Model-View-Whatever | Tương tự như trong MVC, nhưng không bị giới hạn bởi điều khiển |
| 7 | MVVM | Model-View-ViewModel | Một biến thể của mô hình MVC, giúp phân tách logic ứng dụng và giao diện người dùng. |
| 8 | URL | Uniform Resource Locator | Địa chỉ tài nguyên đồng nhất |
| 9 | JPA | Java Persistence API | Giao diện lập trình ứng dụng Java dùng để lưu trữ dữ liệu |
| 10 | XML | eXtensible Markup Language | Ngôn ngữ đánh dấu mở rộng, sử dụng để lưu trữ và truyền thông tin có cấu trúc dưới dạng văn bản. |
| 11 | API | Application Programming Interface | Giao diện lập trình ứng dụng, định nghĩa các quy tắc và phương thức để tương tác và trao đổi dữ liệu giữa các phần mềm khác nhau. |

Bảng 1: Danh mục từ viết tắt

# **Chương 1: Giới thiệu**

## **1.1. Đặt vấn đề**

* Với sự phát triển mạnh mẽ của công nghệ thông tin và Internet, ngành thương mại điện tử đang trở thành một phần không thể thiếu trong cuộc sống hiện đại. Đặc biệt trong lĩnh vực thời trang, việc mua sắm trực tuyến đã trở thành xu hướng phổ biến. Người tiêu dùng có thể dễ dàng duyệt qua các sản phẩm và mua hàng chỉ bằng vài cú nhấp chuột.
* Tuy nhiên, việc xây dựng và phát triển một trang web trức tuyến không phải là điều dễ dàng. Đòi hỏi kiến thức về thiết kế giao diện, quản lý cơ sở dữ liệu, tích hợp thanh toán, và nhiều yếu tố khác. Do đó, vận dụng kiến thức của mình em sẽ nghiên cứu và triển khai một dự án Shop Fashion Website sử dụng AngularJs framework.

## **1.2. Tính cấp thiết của đề tài**

* Sự phổ biến của mua sắm trực tuyến và thị trường thời trang ngày càng tăng đã tạo nên một môi trường cạnh tranh gay gắt. Các doanh nghiệp thời trang cần phải có một nền tảng trực tuyến hiệu quả để thu hút khách hàng và tăng cường doanh số bán hàng.
* Việc xây dựng một trang web chuyên nghiệp và hấp dẫn có thể giúp doanh nghiệp thời trang tiếp cận khách hàng tiềm năng toàn cầu. Ngoài ra, trang web cũng cung cấp các tính năng như giỏ hàng, thanh toán trực tuyến và hệ thống quản lý sản phẩm, giúp quản lý kinh doanh dễ dàng và tiết kiệm thời gian.

## **1.3. Mục tiêu nghiên cứu**

* Thiết kế giao diện hợp lý, dễ sử dụng: Tạo ra một giao diện trực quan, thân thiện với người dùng và thể hiện đúng phong cách của thương hiệu.
* Xây dựng trang hệ thống quản lý linh hoạt, cho phép quản lý thông tin, thêm, xóa, sửa sản phẩm nhanh chóng, quản lý đơn hàng hiệu quả,…
* Xây dựng chức năng giỏ hàng linh hoạt và tiện lợi cho phép khách hàng lựa chọn và quản lý các sản phẩm trước khi tiến hành thanh toán.
* Đảm bảo bảo mật thông tin khách hàng sử dụng dịch vụ của trang web.

# **Chương 2: Cơ sở lý thuyết**

## **2.1. Angular Js framework**

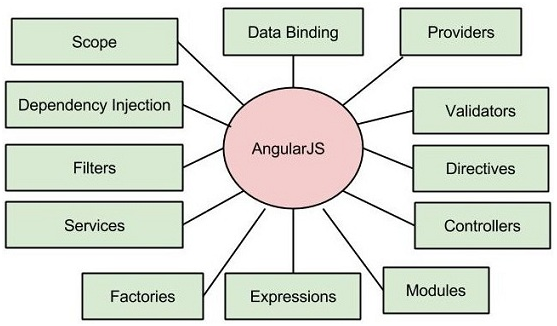
### **2.1.1. Giới thiệu tổng quan về AngularJs**

* AngularJS là một framework mã nguồn mở cho ứng dụng web. Được phát triển đầu tiên vào năm 2009 bởi Misko Hevery và Adam Abrons và hiện được Google duy trì. Phiên bản mới nhất của AngularJS trước khi ngừng cập nhật là 1.7.9.
* Theo tài liệu chính thức từ Google, AngularJS là một framework cấu trúc cho các ứng dụng web động. Nó cho phép sử dụng HTML như ngôn ngữ mẫu và mở rộng cú pháp HTML để diễn tả rõ ràng và ngắn gọn các thành phần ứng dụng. Data binding và dependency injection của nó giúp loại bỏ một phần mã mà hiện tại phải viết, giúp giảm lặp code. Và tất cả điều này xảy ra trong trình duyệt, làm cho nó trở thành đối tác lý tưởng với bất kỳ công nghệ máy chủ nào.

### **2.1.2. Các tính năng chung của AngularJs**

* AngularJS là một Framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).
* AngularJS cung cấp cho nhà phát triển một lựa chọn để viết ứng dụng phía máy khách (font-end) bằng JavaScript theo mô hình MVC rất mạnh mẽ.
* Các ứng dụng viết bằng AngularJS tuân thủ đa trình duyệt, tự động xử lý mã JavaScript phù hợp với mỗi trình duyệt.
* AngularJS là mã nguồn mở, hoàn toàn miễn phí và được sử dụng bởi hàng ngàn nhà phát triển trên toàn thế giới. Nó được cấp phép theo phiên bản 2.0 của giấy phép Apache.
* Tổng thể, AngularJS là một framework để xây dựng các ứng dụng web quy mô lớn, hiệu suất cao và dễ bảo trì.

### **2.1.3. Các tính năng cốt lõi của AngularJs**



Hình 1: Các thành phần chính của AngularJs

* Data-binding: Tự động đồng bộ dữ liệu giữa model và view.
* Scope: Là cầu nối giữa controller và view. Scope được sắp xếp theo cấu trúc DOM của ứng dụng. Nó có thể xem biểu thức và truyền các sự kiện.
* Controller: Đây là các hàm JavaScript được liên kết với một scope cụ thể.
* Services: AngularJS đi kèm với một số dịch vụ tích hợp sẵn như $http để thực hiện các XMLHttpRequests. Đây là các đối tượng singleton chỉ được khởi tạo một lần trong ứng dụng.
* Filters: Là cách định dạng hiển thị dữ liệu.
* Directives: Directives là các đánh dấu trên các phần tử DOM như các phần tử, thuộc tính, CSS và nhiều hơn nữa. Chúng có thể được sử dụng để tạo các thẻ HTML tùy chỉnh mới, là các widget tùy chỉnh. AngularJS có các directive tích hợp sẵn như ngBind, ngModel, vv.
* Templates: Đây là các view được tạo ra với thông tin từ controller và model. Chúng có thể là một tệp duy nhất (như index.html) hoặc nhiều view trong một trang sử dụng partials.
* Routing: Đây là khái niệm chuyển đổi giữa các view.
* Model View Whatever (MVW): MVW là một mô hình thiết kế để chia ứng dụng thành các phần khác nhau gọi là Model, View và Controller, mỗi phần có trách nhiệm riêng. AngularJS không triển khai MVC theo cách truyền thống, mà gần hơn là MVVM. Đội ngũ AngularJS gọi nó một cách hài hước là MVW.
* Deep Linking: Deep linking cho phép mã hóa trạng thái của ứng dụng trong URL để có thể đánh dấu trang. Ứng dụng sau đó có thể khôi phục từ URL với cùng một trạng thái.
* Dependency Injection: AngularJS có một hệ thống dependency injection tích hợp sẵn giúp nhà phát triển dễ dàng tạo, hiểu và kiểm thử các ứng dụng.

### **2.1.4. Một số ưu điểm và nhược điểm của AngularJS:**

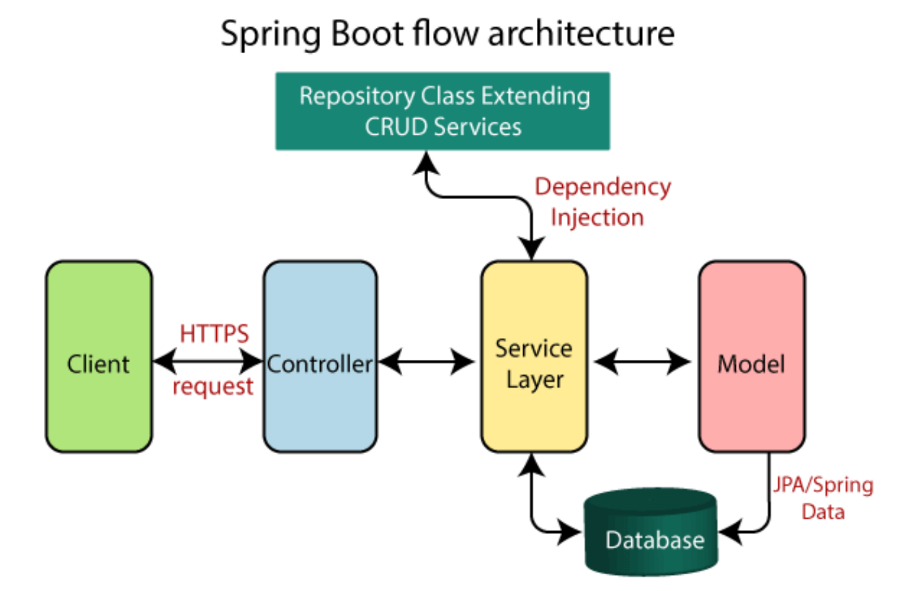
* Về ưu điểm:
* Nó cung cấp khả năng tạo ứng dụng Single Page Application giúp cung cấp các thành phần có thể dễ dàng bảo trì và tái sử dụng.
* Là một framework mã nguồn mở, có cộng đồng hỗ trợ lớn.
* Nó cung cấp khả năng liên kết dữ liệu với HTML, mang lại trải nghiệm phong phú và phản hồi cho người dùng.
* Mã của AngularJS có thể kiểm thử theo đơn vị.
* AngularJS sử dụng dependency injection và tận dụng việc phân chia các quyền trách nhiệm.
* Trong AngularJS, các view là trang HTML thuần túy, và các controller viết bằng JavaScript thực hiện xử lý logic business.
* Trên tất cả, các ứng dụng AngularJS có thể chạy trên tất cả các trình duyệt chính và điện thoại thông minh, bao gồm điện thoại / máy tính bảng Android và iOS.
* Một số điểm đáng lưu ý về nhược điểm của AngularJS bao gồm:
* Không an toàn: Vì AngularJS chỉ là một framework JavaScript, nên ứng dụng viết bằng AngularJS không an toàn. Xác thực và xác nhận quyền phía máy chủ là cần thiết để duy trì tính an toàn của ứng dụng.
* Không thể giảm cấp: Nếu người dùng tắt JavaScript trong ứng dụng của bạn, thì không có gì sẽ hiển thị, ngoại trừ trang cơ bản.

### **2.1.5. Những Components trong AngularJS**

* ng-app: Chỉ định và liên kết một ứng dụng AngularJS với HTML.
* ng-model: Liên kết các giá trị của dữ liệu ứng dụng AngularJS với các điều khiển đầu vào HTML.
* ng-bind: Liên kết dữ liệu ứng dụng AngularJS với các thẻ HTML.

## **2.2. Spring boot framework**

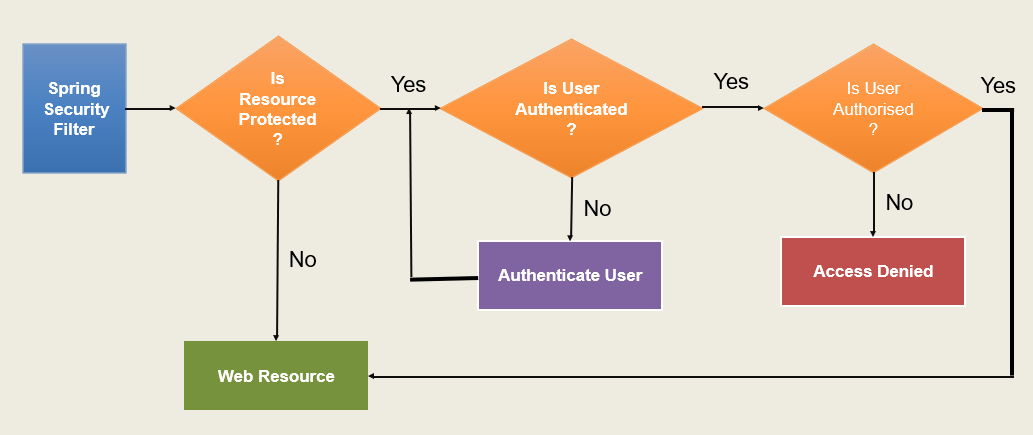
* Spring Boot là một framework phát triển ứng dụng Java dựa trên Spring Framework. Nó cung cấp một cách nhanh chóng và dễ dàng để xây dựng các ứng dụng web Java enterprise-ready. Spring Boot giúp giảm đáng kể việc cấu hình và cung cấp một số tính năng như dependency management tự động, cấu hình tự động, và tích hợp dễ dàng với các công nghệ khác. Trong đề tài này, Spring Boot được sử dụng để xây dựng phần back-end của trang web bán hàng thời trang, xử lý các yêu cầu từ phía người dùng và tương tác với cơ sở dữ liệu.
* Spring Boot tích hợp dễ dàng với các cơ sở dữ liệu quan hệ như MySQL, PostgreSQL và Oracle. Bằng cách sử dụng Spring Data JPA. Spring Data JPA là một thư viện của Spring Framework nó giúp đơn giản hóa việc truy xuất cơ sở dữ liệu trong ứng dụng Java sử dụng JPA .
* Ưu điểm của Spring Boot framework:
* Tự động cấu hình.
* Xây dựng các bean dựa trên annotation thay vì XML.
* Server Tomcat được nhúng trong file JAR, không cần cấu hình server riêng.
* Cấu trúc source code của Spring Boot gồm các thành phần sau:
* Controller: trả về View ( có chứa data sẵn, dạng trang HTML), hoặc Model thể hiện dưới dạng API cho View ( View viết riêng bằng React, Vue hoặc Angular).
* Service: chứa các code tính toán, xử lý. Khi Controller yêu cầu, thì Service tương ứng sẽ tiếp nhận và cho ra dữ liệu trả cho Controller ( trả về Model). Controller sẽ gửi về View như trên, ngoài ra Service còn có thể tương tác với Service khác.
* Repository: là thành phần trực tiếp tương tác, đọc ghi dữ liệu trong DB và trả cho service.



Hình 2: Cấu trúc và work flow của Spring boot

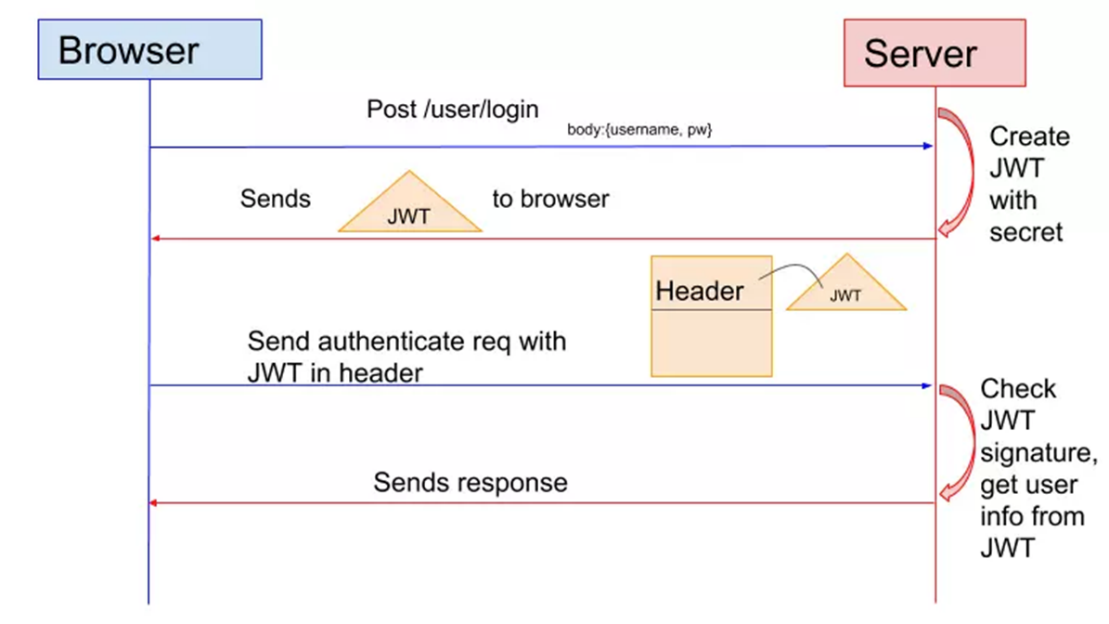
## **2.3. Spring security**

* Spring Security là một framework mạnh mẽ cho việc xác thực và phân quyền trong ứng dụng Java. Nó cung cấp một cơ chế linh hoạt để bảo vệ các tài nguyên của ứng dụng và kiểm soát quyền truy cập của người dùng.



Hình 3:Luồng hoạt động của spring security

* JSON Web Token (JWT) là một phương thức thông dụng để truyền tải thông tin xác thực an toàn giữa các bên trong một ứng dụng. JWT bao gồm ba phần chính: Header, Payload và Signature. Header chứa thông tin về thuật toán mã hóa được sử dụng, Payload chứa các thông tin xác thực và các dữ liệu tùy ý, và Signature được sử dụng để xác thực tính toàn vẹn của token.
* Spring Security cung cấp tích hợp sẵn với JWT thông qua việc sử dụng JWT Authentication trong quá trình xác thực người dùng. Cơ chế này cho phép người dùng đăng nhập và nhận JWT sau khi xác thực thành công. JWT sau đó được sử dụng để xác thực các yêu cầu tiếp theo của người dùng.



Hình 4: Luồng hoạt động của Spring security + JWT

* Mô tả luồng hoạt động:
* Người dùng gửi yêu cầu đăng nhập (ví dụ: gửi tên người dùng và mật khẩu) từ giao diện người dùng của ứng dụng.
* Ứng dụng Java sử dụng Spring Security để xác thực thông tin đăng nhập của người dùng.
* Nếu thông tin đăng nhập hợp lệ, ứng dụng tạo một JWT chứa thông tin người dùng xác thực và các thông tin khác cần thiết.
* JWT được gửi trả lại cho người dùng.
* Người dùng sau đó gắn JWT vào mỗi yêu cầu tiếp theo trong tiêu đề (header) của HTTP.
* Ứng dụng sử dụng Spring Security để kiểm tra tính hợp lệ của JWT và xác thực người dùng.
* Nếu JWT hợp lệ và xác thực thành công, yêu cầu của người dùng được thực hiện.
* Việc sử dụng JWT trong Spring Security giúp giảm sự phụ thuộc vào việc lưu trữ phiên đăng nhập trên máy chủ và cho phép ứng dụng mở rộng dễ dàng trong môi trường phân tán.

## **2.4. Restful API**

Restful API (Representational State Transfer API) là một kiểu kiến trúc dựa trên giao thức HTTP được sử dụng để truyền tải và trao đổi dữ liệu giữa các hệ thống và ứng dụng. Nó dựa trên các phương thức HTTP như GET, POST, PUT và DELETE để thực hiện các thao tác trên các tài nguyên. Trong đề tài này, Restful API được sử dụng để tạo các endpoint để truy cập và thao tác dữ liệu của trang web bán hàng thời trang. Các yêu cầu từ phía người dùng sẽ được gửi qua các API này để tương tác với dữ liệu từ cơ sở dữ liệu.

## **2.5. Mô hình MVC**

* MVC (Model-View-Controller) là một mô hình phát triển phần mềm phổ biến được sử dụng trong việc phân chia và quản lý logic ứng dụng. Mô hình MVC chia ứng dụng thành ba phần chính:
* Model: Đại diện cho dữ liệu và logic xử lý dữ liệu. Nó có trách nhiệm lưu trữ và cung cấp dữ liệu cho Controller và View.
* View: Đại diện cho giao diện người dùng. Nó hiển thị dữ liệu từ Model và cung cấp giao diện tương tác với người dùng.
* Controller: Điều phối các hoạt động và tương tác giữa Model và View. Nó nhận lệnh từ View, truy xuất và cập nhật dữ liệu từ Model, sau đó cập nhật lại giao diện của View.
* Mô hình MVC giúp tách biệt logic ứng dụng và giao diện người dùng, làm cho ứng dụng dễ bảo trì, dễ kiểm thử và mở rộng. Nó cũng đảm bảo tính nhất quán và khả năng tái sử dụng của code.

# **Chương 3: Xây dựng đồ án**

## **3.1. Phân tích yêu cầu chức năng**

**3.1.1. Yêu cầu chức năng**

* **Đăng ký, đăng nhập**
* Người dùng có thể đăng ký tài khoản mới bằng cách cung cấp thông tin cá nhân như họ và tên, địa chỉ email, password. Ngoài ra người dùng có thể đăng ký, đăng nhập bằng tài khoản mạng xã hội như Facebook, Google .
* Người dùng có thể thay đổi mật khẩu của mình sau khi đăng ký tài khoản.
* Người dùng có thể khôi phục, đổi mật khẩu của mình xác minh qua email, thông qua chức năng quên mật khẩu.
* **Xem sản phẩm và danh mục**
* Người dùng có thể xem danh sách các sản phẩm được bày bán trên trang web.
* Tìm kiếm sản phẩm theo tên sản phẩm, loại, mức giá, v.v.
* Xem chi tiết sản phẩm bao gồm hình ảnh, mô tả, giá, đánh giá, v.v.
* Người dùng có thể xem danh mục các loại sản phẩm để dễ dàng tìm kiếm.
* **Giỏ hàng và thanh toán**
* Người dùng có thể thêm sản phẩm vào giỏ hàng và xem thông tin chi tiết về giỏ hàng.
* Người dùng có thể điều chỉnh số lượng sản phẩm trong giỏ hàng hoặc xóa sản phẩm ra khỏi giỏ hàng.
* Người dùng có thể thực hiện thanh toán cho các mặt hàng trong giỏ hàng sau khi đã đăng nhập vào hệ thống, và có thể thanh toán thông qua các phương thức thanh toán điện tử hoặc thanh toán khi nhận hàng.
* **Quản lý đơn hàng**
* Xem danh sách các đơn hàng đã đặt.
* Xem chi tiết về từng đơn hàng bao gồm sản phẩm, tổng số tiền, địa chỉ giao hàng, v.v.
* Có thể hủy đơn hàng trước khi được giao.
* **Quản lý tài khoản**
* Cập nhật thông tin cá nhân như địa chỉ, số điện thoại, v.v.
* Xem lịch sử mua hàng và đánh giá sản phẩm.
* Gửi phản hồi hoặc yêu cầu hỗ trợ từ nhà cung cấp.

**3.1.2. Yêu cầu phi chức năng**

* Giao diện trang web phải thân thiện, dễ sử dụng và thích hợp với các thiết bị di động.
* Trang web phải đảm bảo an toàn thông tin của người dùng bằng cách sử dụng các biện pháp bảo mật như mã hóa dữ liệu, xác thực người dùng, v.v.
* Người dùng phải được cung cấp quyền kiểm soát riêng tư thông tin cá nhân của mình.
* Trang web phải có khả năng chia sẻ sản phẩm qua các mạng xã hội và gửi thông tin đến email của người dùng.
* Trang web phải có khả năng sao lưu và khôi phục dữ liệu khi cần thiết.

## **3.2. Phân tích nghiệp vụ**

**3.2.1. Bảng xác định các yêu cầu nghiệp vụ**

* **Bảng các yêu cầu**

|  |  |  |  |
| --- | --- | --- | --- |
| Mã số YC | Tên yêu cầu | Loại yêu cầu | Mã số qui định |
| YC001 | Hỗ trợ bán hàng trực tuyến | Nghiệp vụ | QĐ001 |
| YC002 | Tìm kiếm sản phẩm theo danh mục | Chức năng | QĐ002 |
| YC003 | Tìm kiếm sản phẩm theo giá cả | Chức năng | QĐ002 |
| YC004 | Quản lý đơn hàng | Nghiệp vụ | QĐ003 |
| YC005 | Thanh toán đơn hàng | Nghiệp vụ | QĐ004 |
| YC006 | Đăng ký tài khoản | Chức năng | QĐ005 |
| YC007 | Đăng nhập vào hệ thống | Chức năng | QĐ006 |
| YC008 | Quản lý thông tin cá nhân của người dùng | Chức năng | QĐ007 |
| YC009 | Thêm sản phẩm vào giỏ hàng | Chức năng | QĐ008 |
| YC010 | Cập nhật giỏ hàng | Chức năng | QĐ008 |
| YC011 | Xóa sản phẩm khỏi giỏ hàng | Chức năng | QĐ008 |
| YC012 | Tạo và quản lý danh sách yêu thích sản phẩm | Chức năng | QĐ009 |
| YC013 | Thiết kế giao diện thân thiện | Phi chức năng | QĐ010 |
| YC014 | Xử lý đồng thời nhiều yêu cầu | Phi chức năng | QĐ011 |
| YC015 | Bảo mật thông tin cá nhân của người dùng | Phi chức năng | QĐ012 |
| YC016 | Lưu trữ thông tin sản phẩm | Dữ liệu | QĐ013 |
| YC017 | Lưu trữ thông tin người dùng | Dữ liệu | QĐ014 |

Bảng 2: Bảng các yêu cầu

* **Bảng các qui định**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mã số qui định** | **Tên qui định** | **Mô tả chi tiết** |
| 1 | QĐ001 | Hỗ trợ bán hàng trực tuyến | Yêu cầu hệ thống phải hỗ trợ việc bán hàng trực tuyến. |
| 2 | QĐ002 | Tìm kiếm sản phẩm | Yêu cầu hệ thống phải hỗ trợ tìm kiếm sản phẩm theo các tiêu chí nhất định. |
| 3 | QĐ003 | Quản lý đơn hàng | Yêu cầu hệ thống phải cung cấp chức năng quản lý đơn hàng. |
| 4 | QĐ004 | Thanh toán đơn hàng | Yêu cầu hệ thống phải hỗ trợ quá trình thanh toán đơn hàng. |
| 5 | QĐ005 | Đăng ký tài khoản | Yêu cầu hệ thống phải có chức năng đăng ký tài khoản |
| 6 | QĐ006 | Đăng nhập vào hệ thống | Yêu cầu hệ thống phải có chức năng đăng nhập và quản lý tài khoản. |
| 7 | QĐ007 | Quản lý thông tin cá nhân | Yêu cầu hệ thống phải cho phép người dùng quản lý thông tin cá nhân. |
| 8 | QĐ008 | Quản lý giỏ hàng | Yêu cầu hệ thống phải hỗ trợ thao tác thêm, cập nhật và xóa sản phẩm khỏi giỏ hàng. |
| 9 | QĐ009 | Quản lý danh sách yêu thích | Yêu cầu hệ thống phải cho phép người dùng tạo và quản lý danh sách yêu thích sản phẩm. |
| 10 | QĐ010 | Thiết kế giao diện thân thiện | Yêu cầu hệ thống phải có giao diện thân thiện, dễ sử dụng. |
| 11 | QĐ011 | Xử lý đồng thời nhiều yêu cầu | Yêu cầu hệ thống phải có khả năng xử lý đồng thời nhiều yêu cầu từ người dùng. |
| 12 | QĐ012 | Bảo mật thông tin cá nhân | Yêu cầu hệ thống phải đảm bảo bảo mật thông tin cá nhân của người dùng. |
| 13 | QĐ013 | Lưu trữ thông tin sản phẩm | Yêu cầu hệ thống phải lưu trữ thông tin về sản phẩm. |
| 14 | QĐ014 | Lưu trữ thông tin người dùng | Yêu cầu hệ thống phải lưu trữ thông tin về người dùng. |

Bảng 3: Bảng các qui định

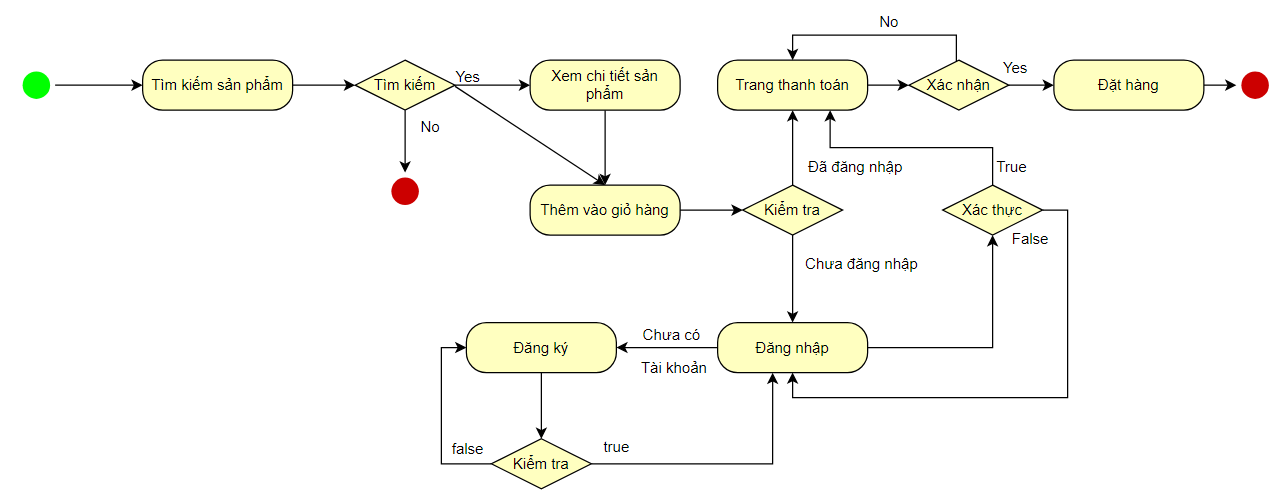
**3.2.2. Bảng các yêu cầu về chất lượng**

|  |  |  |
| --- | --- | --- |
| **STT** | **Tiêu chuẩn** | **Nội dung** |
| 1 | Website phải vận hành thường xuyên | Website đó phải bảo đảm thời gian vận hành thường xuyên, hạn chế tối đa thời gian ngắt quãng. |
| 2 | Website có tốc độ tải trang nhanh. | Theo thống kê chung, người dùng không đủ kiên nhẫn quá 10 giây khi vào một trang web mà vẫn chưa thấy có thành phần nào trong trang web xuất hiện. |
| 3 | Giao diện website đẹp, thiết kế hợp lý | Website phải có giao diện, hình ảnh, màu sắc bắt mắt, thu hút được người dùng truy cập. |
| 4 | Dễ sử dụng | Các chức năng của website phải dễ dàng sử dụng, tập trung vào những nhu cầu cần thiết và tránh các thủ tục rườm rà, không phù hợp với người dùng khó tính và thiếu kiên nhẫn. |
| 5 | Mang lại lợi ích kinh doanh của doanh nghiệp | Website đó phải mang lại lợi ích kinh doanh của doanh nghiệp, là một công cụ hỗ trợ đắc lực giúp doanh nghiệp tăng doanh số và tiết kiệm thời gian, chi phí. |

Bảng 4: Bảng các yêu cầu về chất lượng

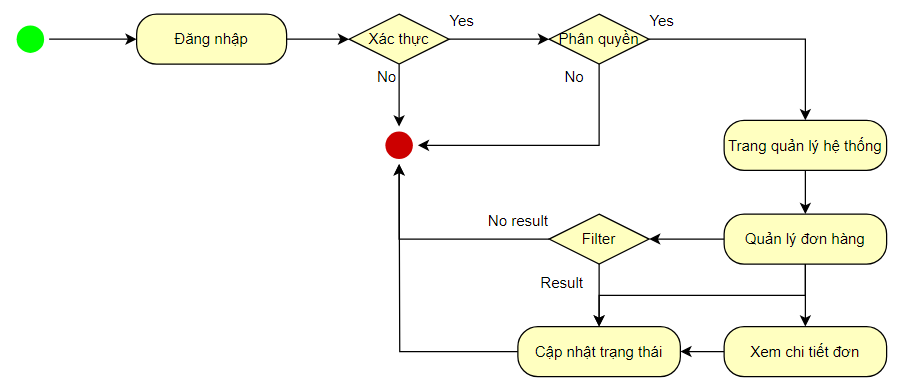
**3.2.3. Các sơ đồ phân tích nghiệp vụ**

* **Sơ đồ mô tả qui trình đặt hàng**



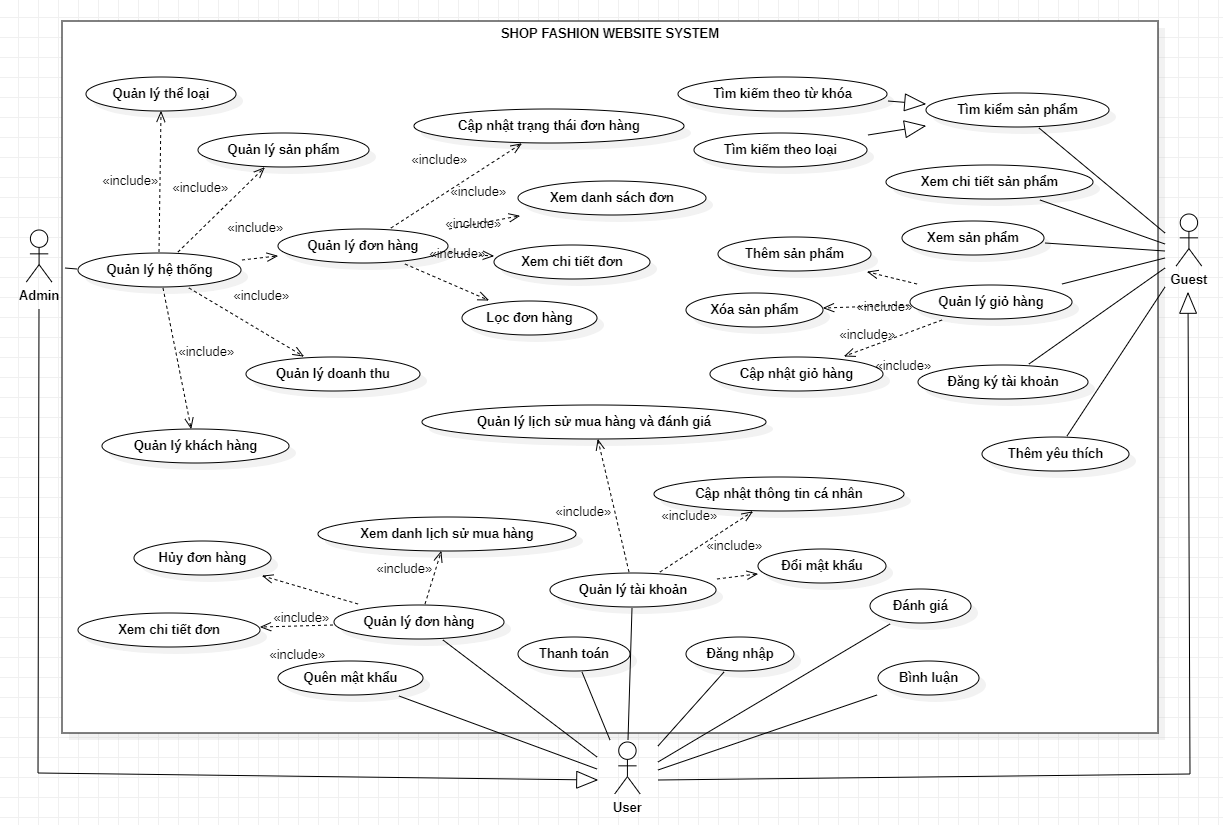
Hình 5: Sơ đồ mô tả qui trình đặt hàng

* **Sơ đồ mô tả qui trình quản lý đơn hàng**



Hình 6: Sơ đồ mô tả qui trình quản lý đơn hàng

* **Sơ đồ UC cho yêu cầu**

****

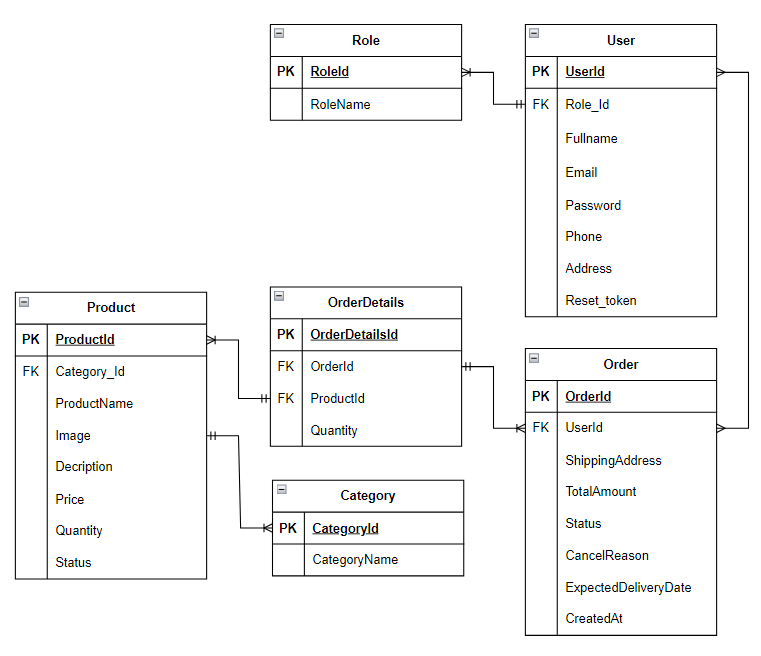
Hình 7: Sơ đồ Usecase tổng quan yêu cầu chức năng

## **3.3. Xây dựng DB**

**3.3.1. Xác định các thực thể**

* Bảng category: categoryId, categoryName
* Bảng product: productId, categoryId, productName, image, description, price, quantity, status
* Bảng role: roleId, roleName
* Bảng user: userId, roleId, fullname, email, password, phone, address, resetToken, status
* Bảng order: orderId, userId, shippingAddress, totalAmount, status, cancelReason, expectedDeliveryDate, createdAt
* Bảng orderDetail: orderDetailId, orderId, productId, quantity

**3.3.2. Sơ đồ ERD mối quan hệ giữa các thực thể**



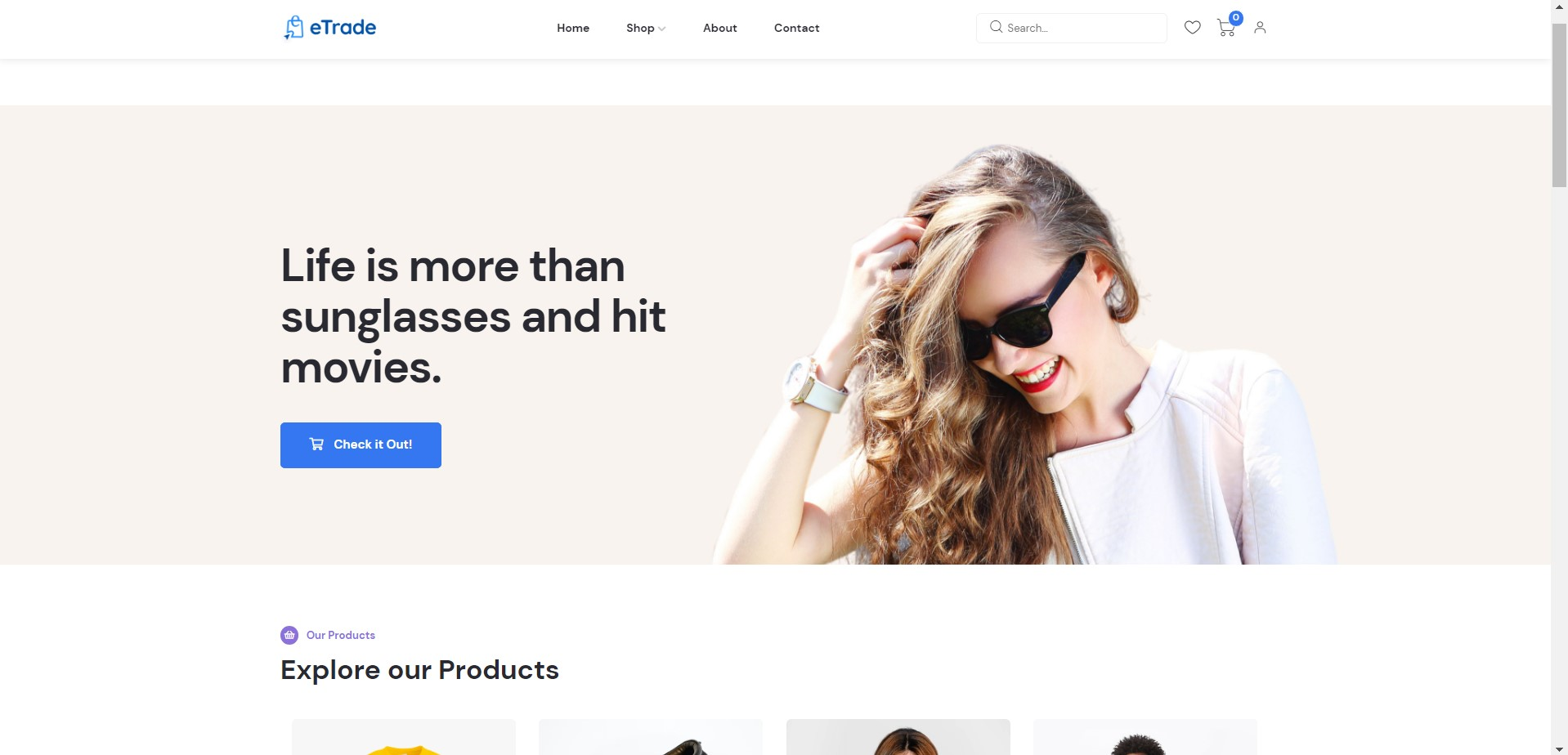
Hình 3.3.2. Sơ đồ ERD mối quan hệ giữa các thực thể

## **3.4. Kết quả thực nghiệm**

**3.4.1. Giao diện người dùng**

* **Trang chủ**

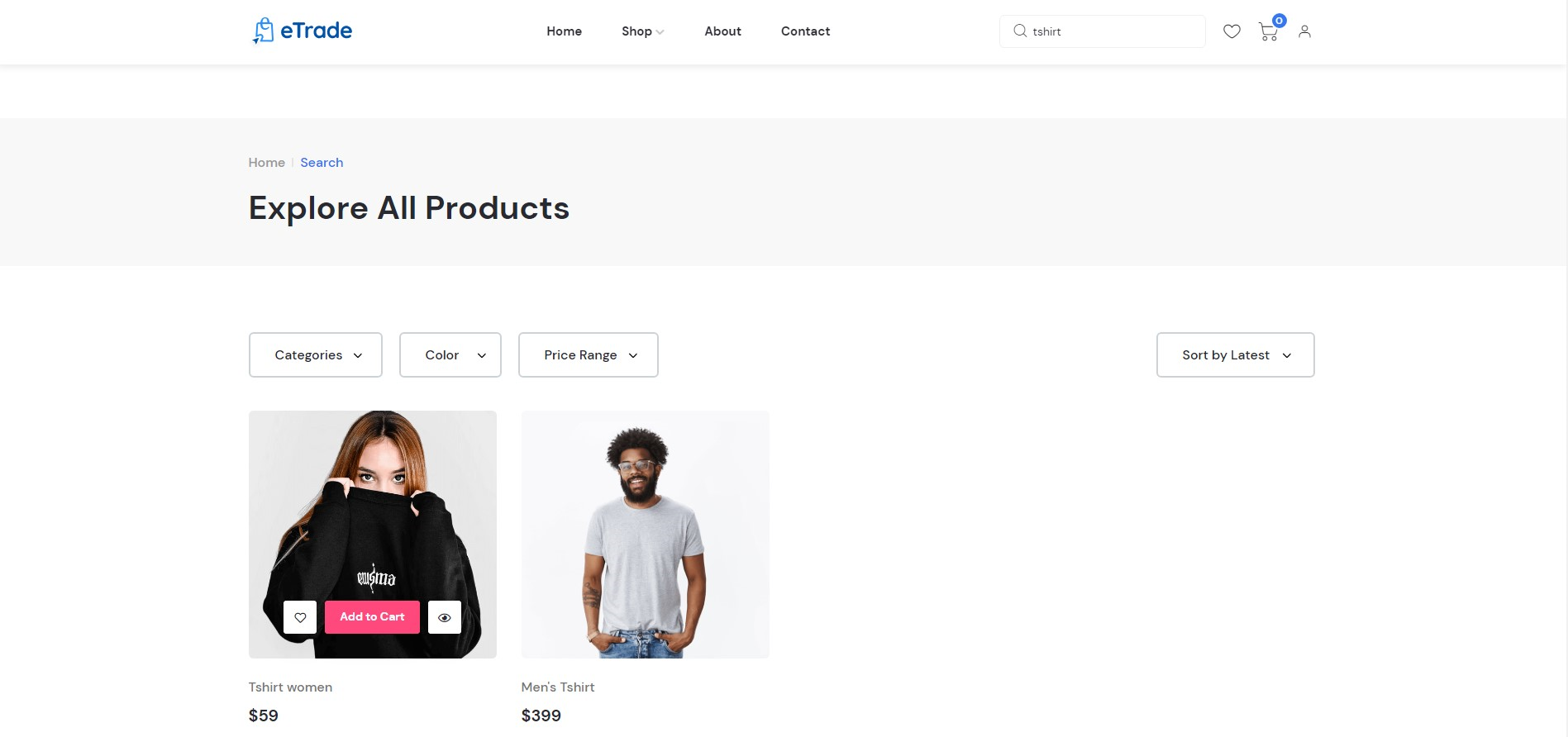
Tại trang chủ người dùng có thể xem những sản phẩm mới nhất của trang web, đồng thời có thể thực hiện tìm kiếm và thêm sản phẩm vào giỏ hàng của mình.



Hình 8: Giao diện trang chủ

* **Tìm kiếm sản phẩm**

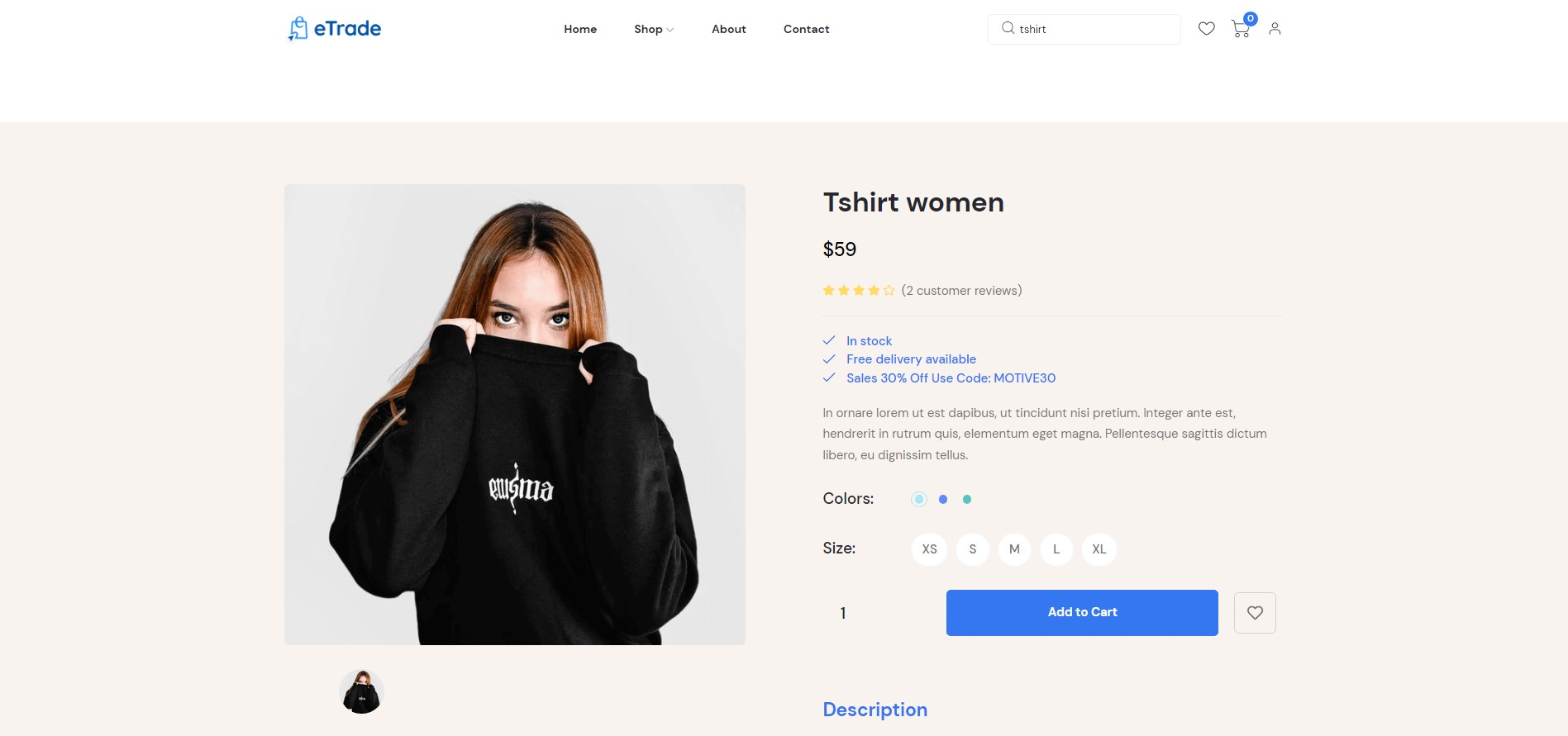
Người dùng có thể tìm kiếm sản phẩm theo từ khóa hoặc theo loại sản phẩm.Ví dụ, hình sau đây tìm kiếm sản phẩm theo từ khóa ‘Tshirt’ trên thanh menu tìm kiếm.



Hình 9: Tìm kiếm sản phẩm

* **Xem chi tiết sản phẩm**

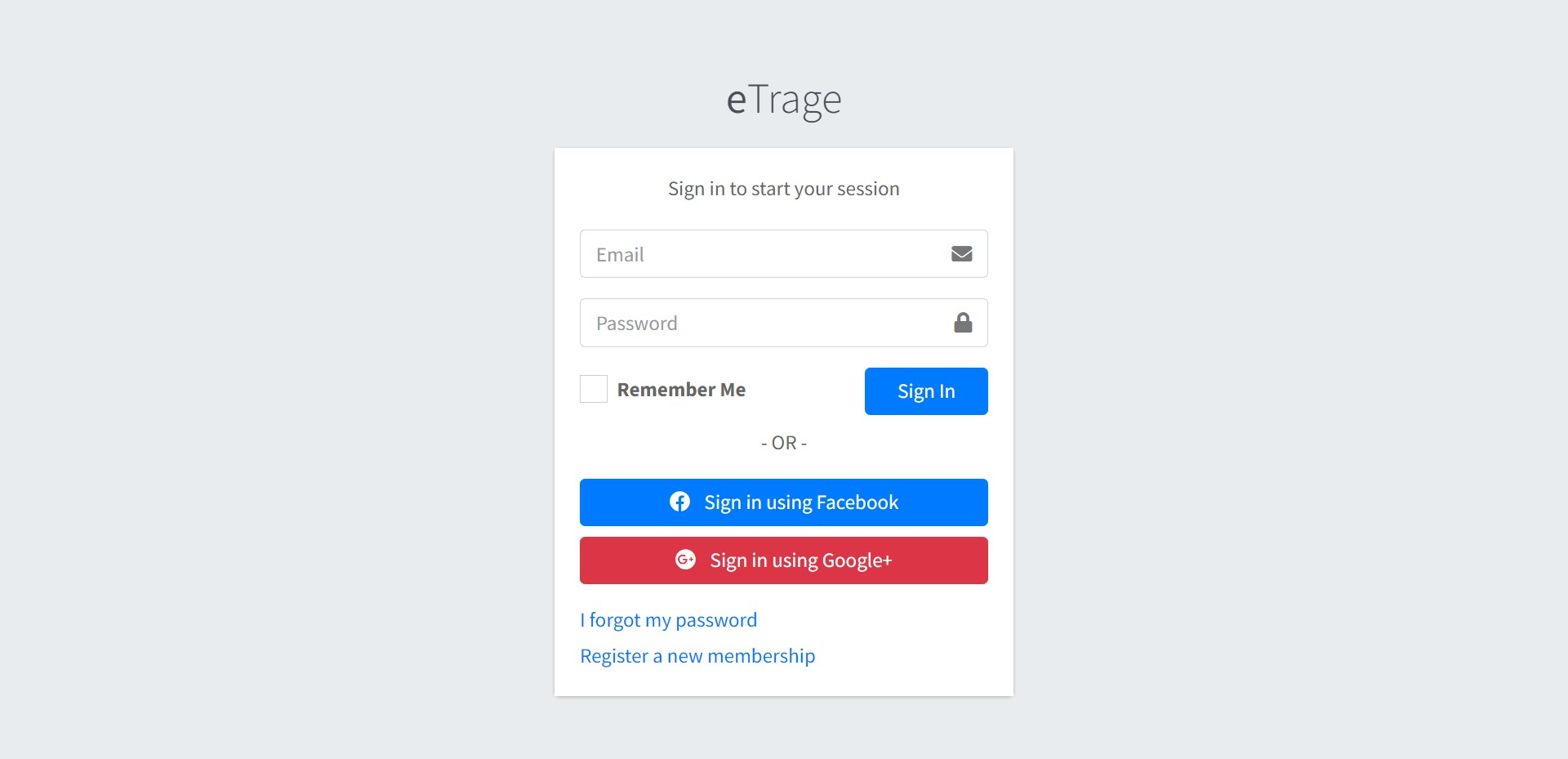
Người dùng có thể xem chi tiết sản phẩm khi click vào mỗi sản phẩm, tại trang này người dùng có thể xem chi tiết về giá, mô tả, (chọn màu sắc, kích cỡ) số lượng đặt hàng, sau đó tiến hành thêm vào giỏ hàng



Hình 10: Xem chi tiết sản phẩm

* **Đăng nhập**

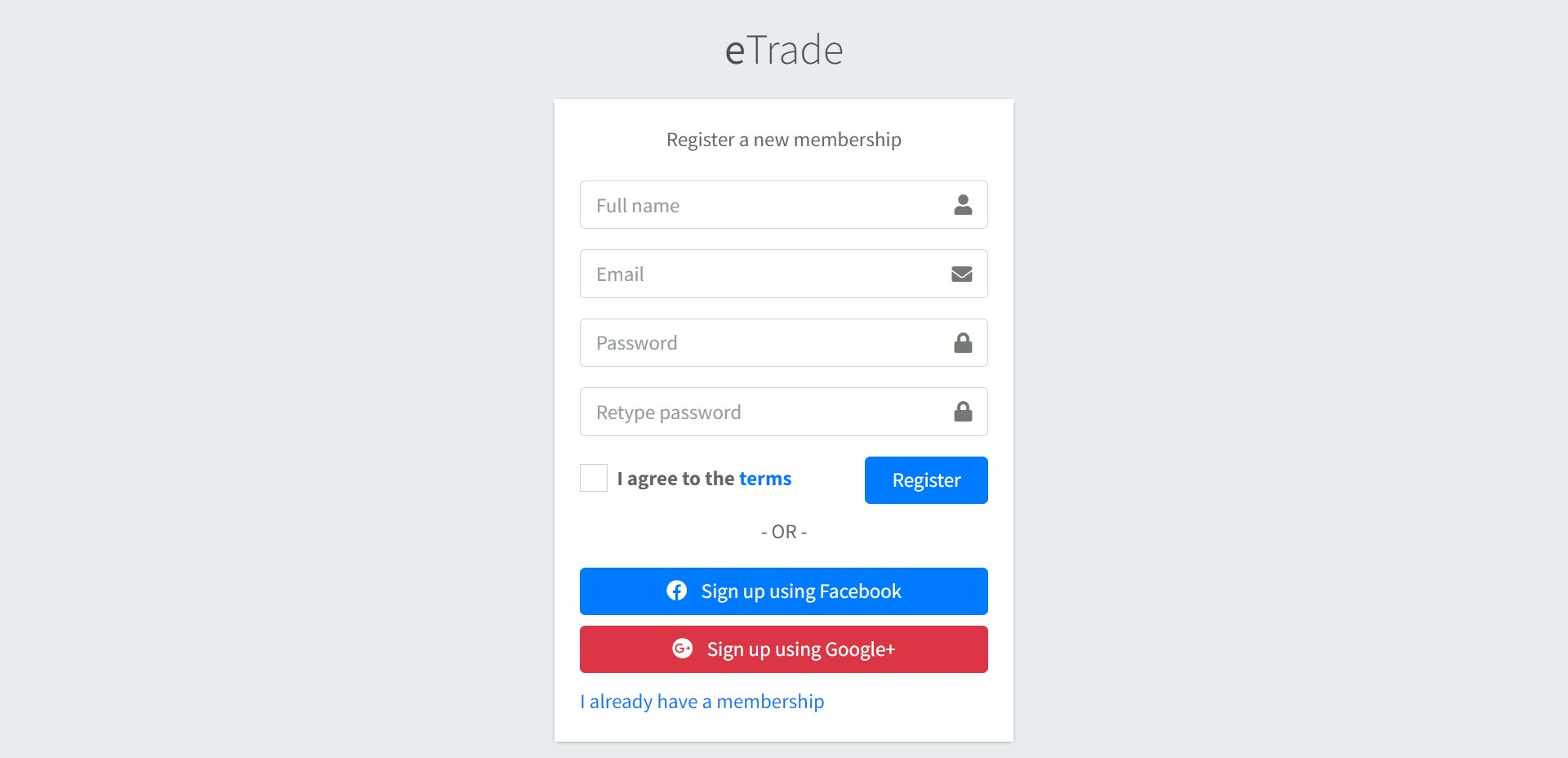
Để tiến hành quản lý tài khoản hoặc quản lý hệ thống, thanh toán người dùng cần có tài khoản để đăng nhập xác thực và phân quyền người dùng, người dùng có thể đăng nhập bằng email, password.



Hình 11: Đăng nhập

* **Đăng ký**

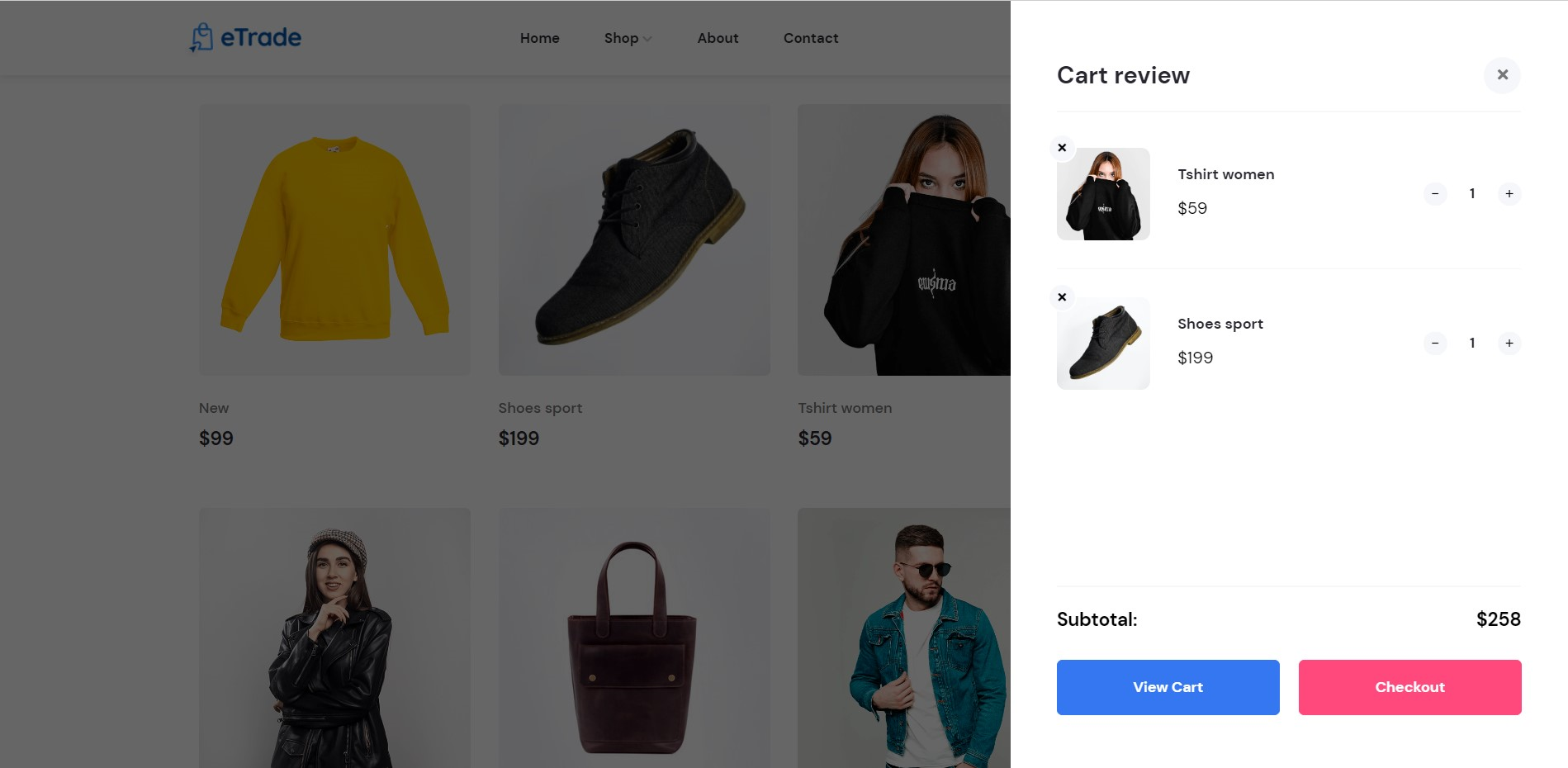
Nếu chưa có tài khoản, người dùng có thể đăng ký tài khoản mới bằng họ tên, email, password, mỗi account chỉ được dùng 1 email duy nhất và mật khẩu đăng ký phải tối đai 8 kí tự, có 1 ký tự viết hoa và 1 ký tự đặc biệt theo qui định an toàn bảo mật của google.



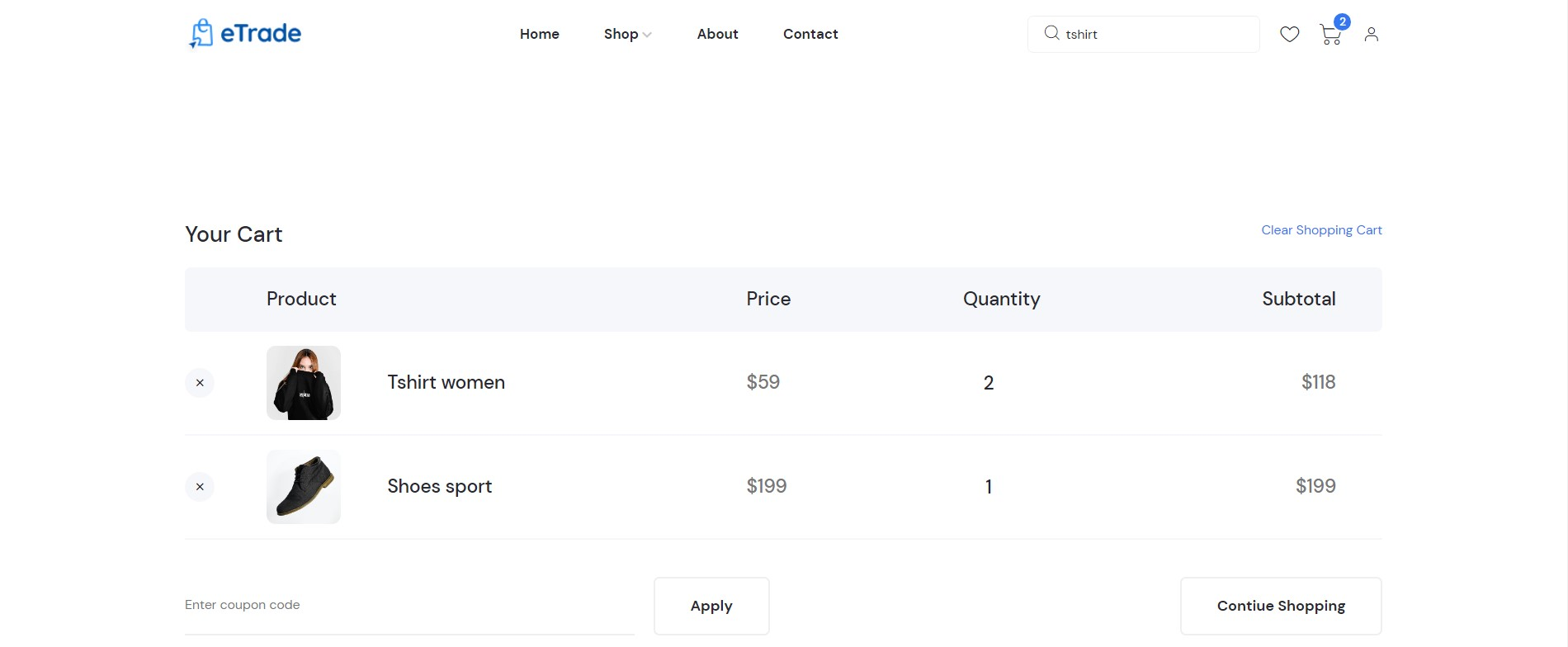
Hình 12: Đăng ký tài khoản

* **Giỏ hàng**

Sau khi thêm giỏ hàng, người dùng có thể xem giỏ hàng của mình, bằng cách click vào button giỏ hàng trên thanh menu trang web. Tại đây người dùng có thể cập nhật giỏ hàng của mình ( cập nhật số lượng, xóa sản phẩm khỏi giỏ hàng) và có thể tiến hành thanh toán để đặt hàng.



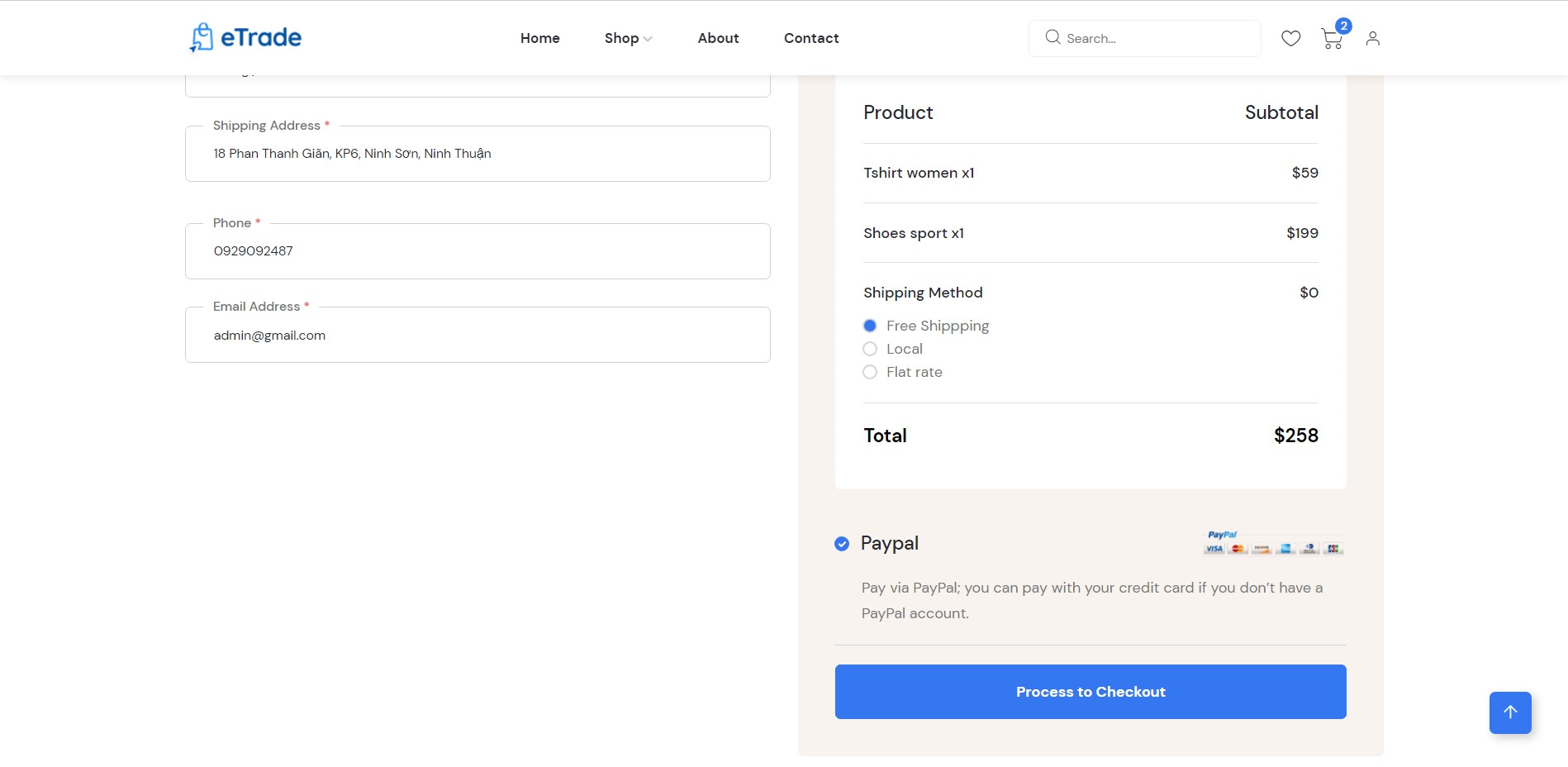
Hình 13: Xem giỏ hàng



Hình 14: Trang giỏ hàng

* **Thanh toán**

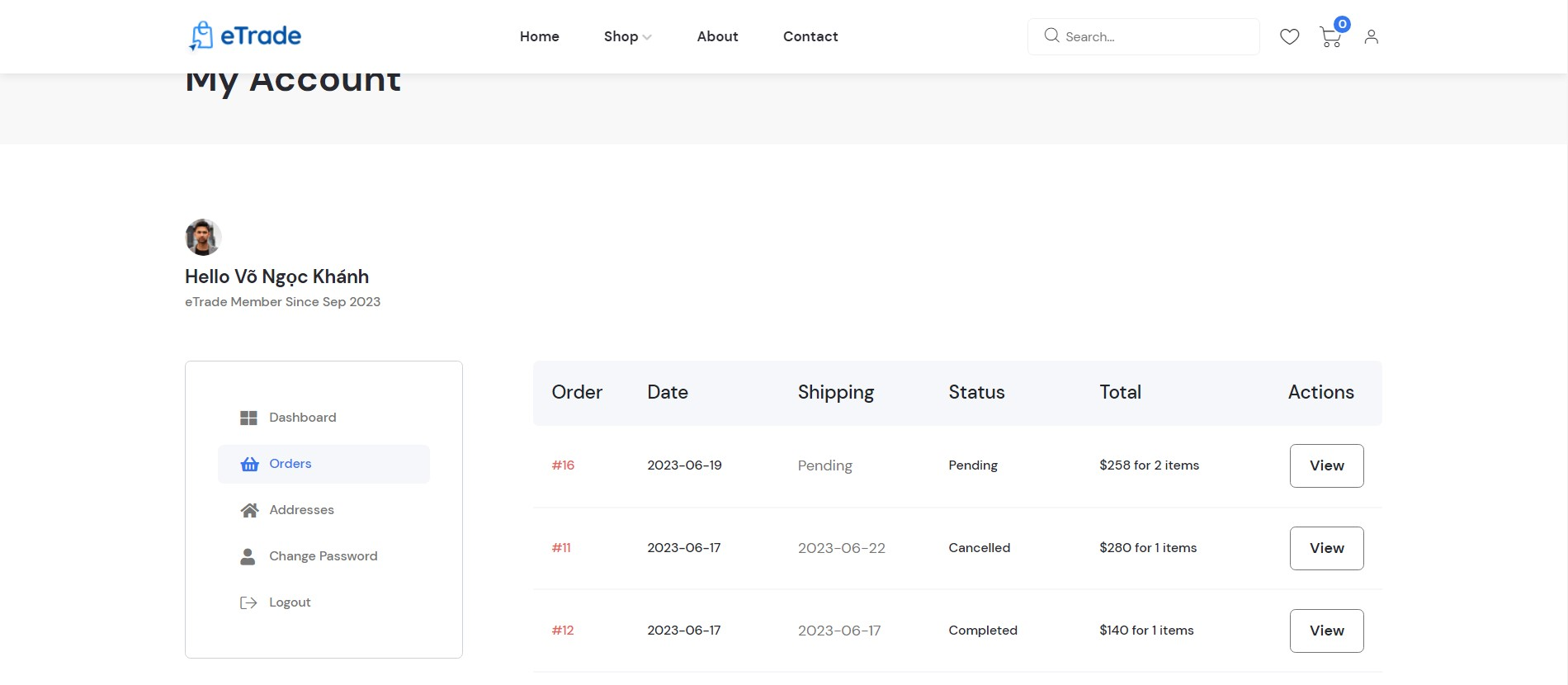
Để thanh toán người dùng cần cung cấp thêm thông tin cá nhân như số điện thoại và địa chỉ nhận hàng để hệ thống xác nhận đơn và ship đến người dùng. ( Ngoài ra, người dùng có thể chọn các phương thức vận chuyển và thanh toán khác nhau tùy nhau cầu của người dùng)



Hình 15: Trang thanh toán

* **Quản lý tài khoản**

Người dùng có thể xem lịch sử đặt hàng của mình tại trang quản lý tài khoản, bằng cách click vào icon người dùng trên thanh menu website, tại đây người dùng có thể xem các đơn mà mình đã đặt, xem thông tin cá nhân, đổi mật khẩu,..



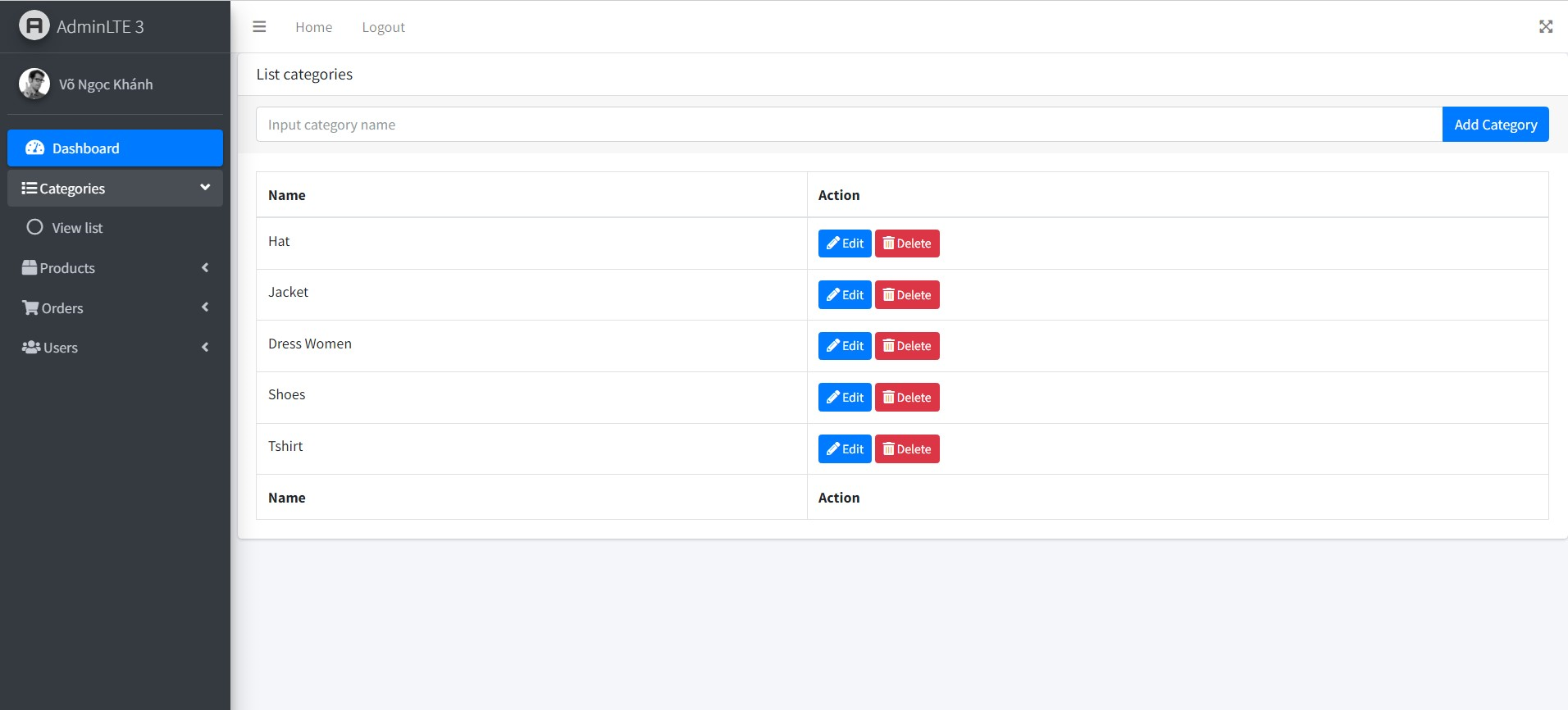
Hình 16: Quản lý đơn hàng đã đặt

**3.4.2. Giao diện quản lý hệ thống**

Để truy cập vào được giao diện quản lý hệ thống người dùng cần có role là admin

* **Quản lý thể loại**

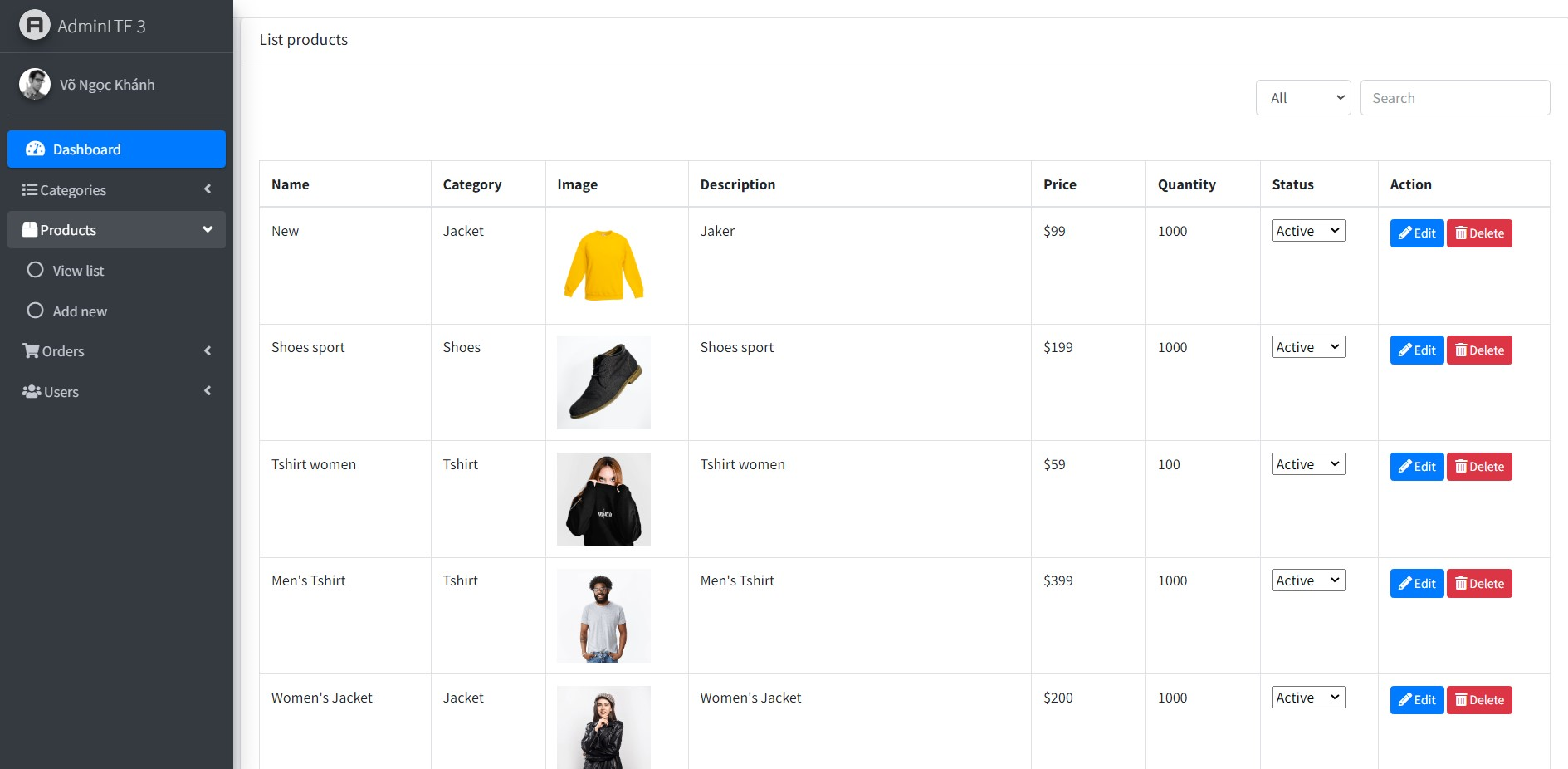
Tại trang này, admin có thể thêm, xóa, cập nhật thể loại sản phẩm, điều kiện là thể loại đó không được trùng lặp.



Hình 17: Quản lý thể loại sản phẩm

* **Quản lý sản phẩm**

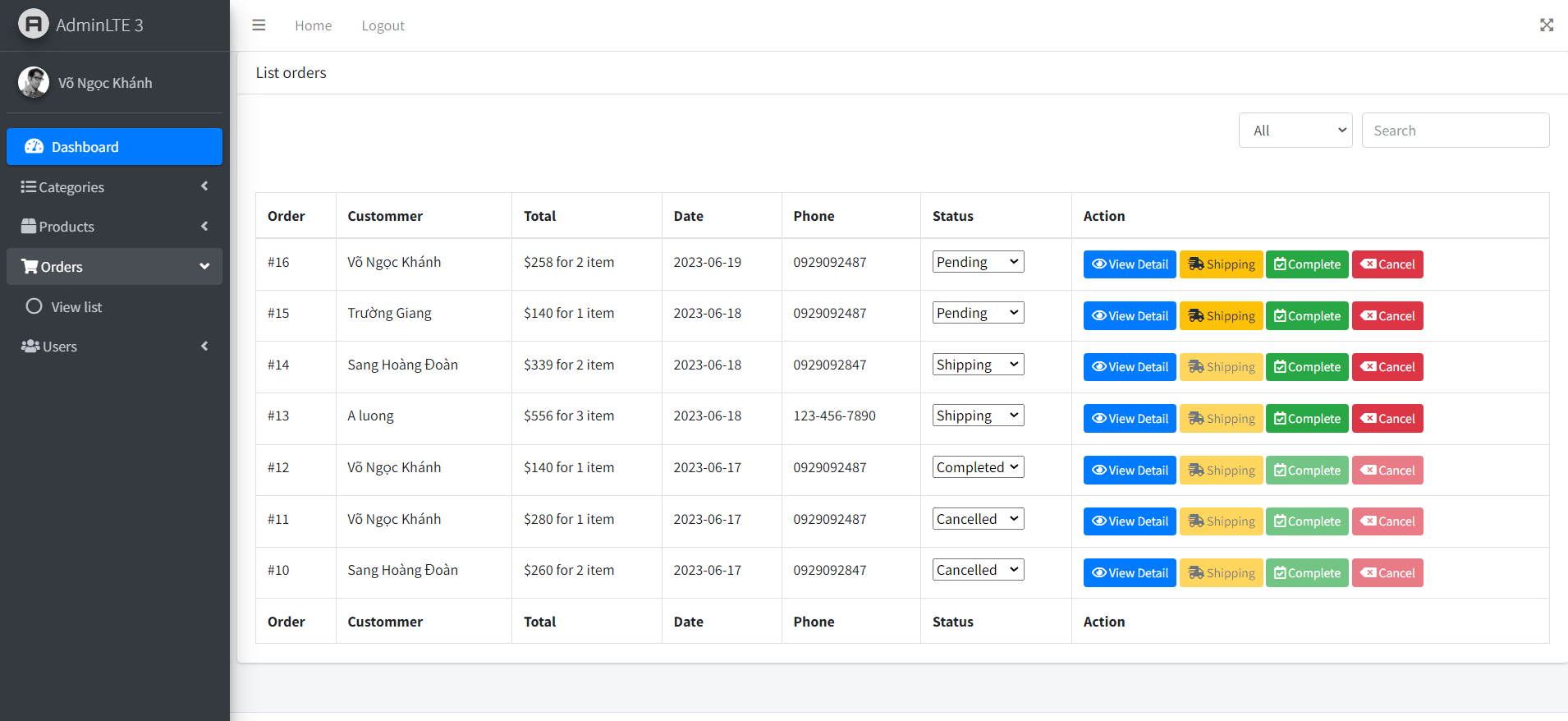
Tại trang này, admin có thể thêm, xóa, sửa sản phẩm. Khi sản phẩm hết hàng, người dùng có thể cập nhật trạng thái sản phẩm là Inactive để vô hiệu hóa sản phẩm. Sau khi Inactive sản phẩm sẽ không được hiển thị trên trang người dùng.



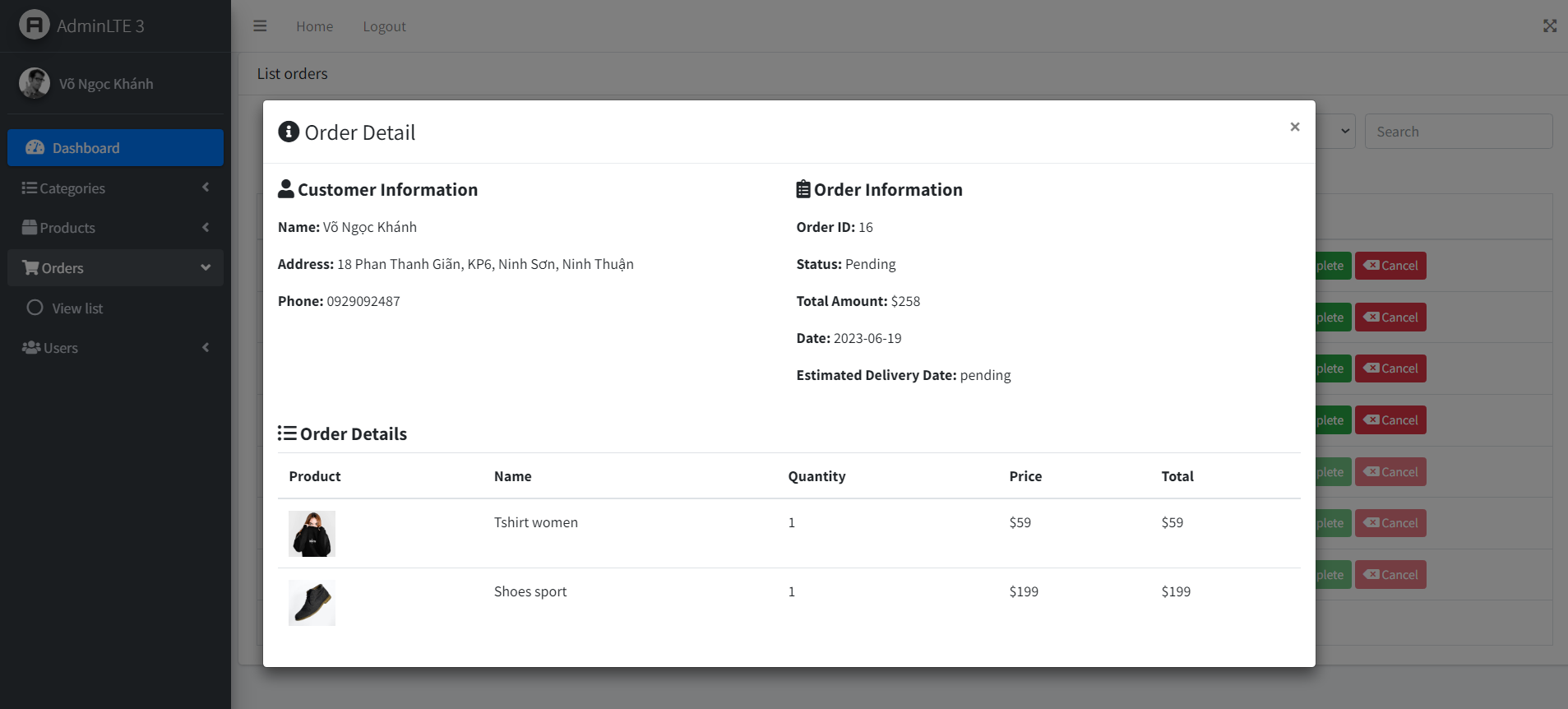
Hình 18: Quản lý sản phẩm

* **Quản lý đơn hàng**

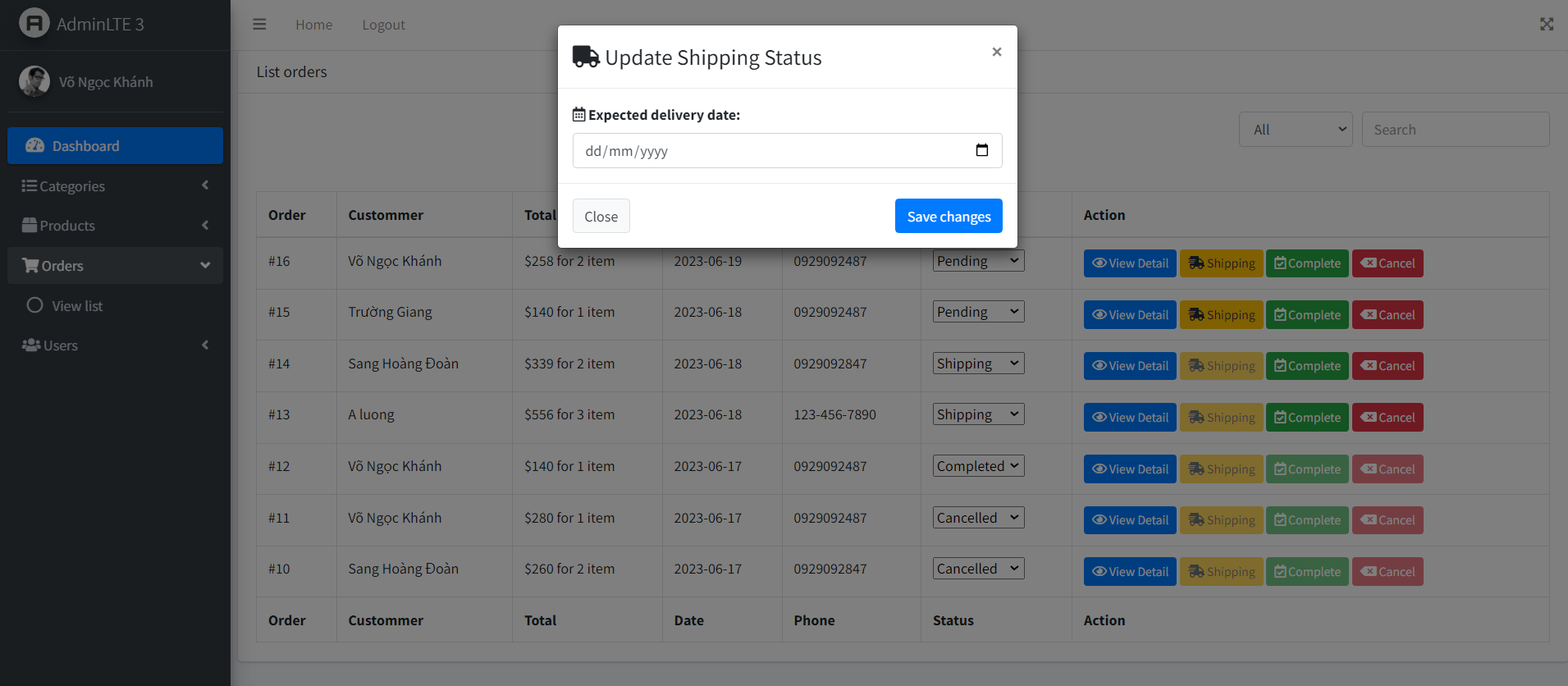
Tại trang này, admin có thể quản lý các đơn hàng từ người dùng và thực hiện có chức năng như filter đơn hàng, xem chi tiết đơn, cập nhật trạng thái đơn (shipping cần nhập ngày giao hàng dự kiến, completed, cancelled cần nhập lý do hủy đơn). Trạng thái cancelled có thể được cập nhật từ người dùng or admin trước khi đơn hàng được giao. Để tránh tình trạng spam cancelled, admin nên gọi điện xác nhận với người dùng trước khi ship.

****

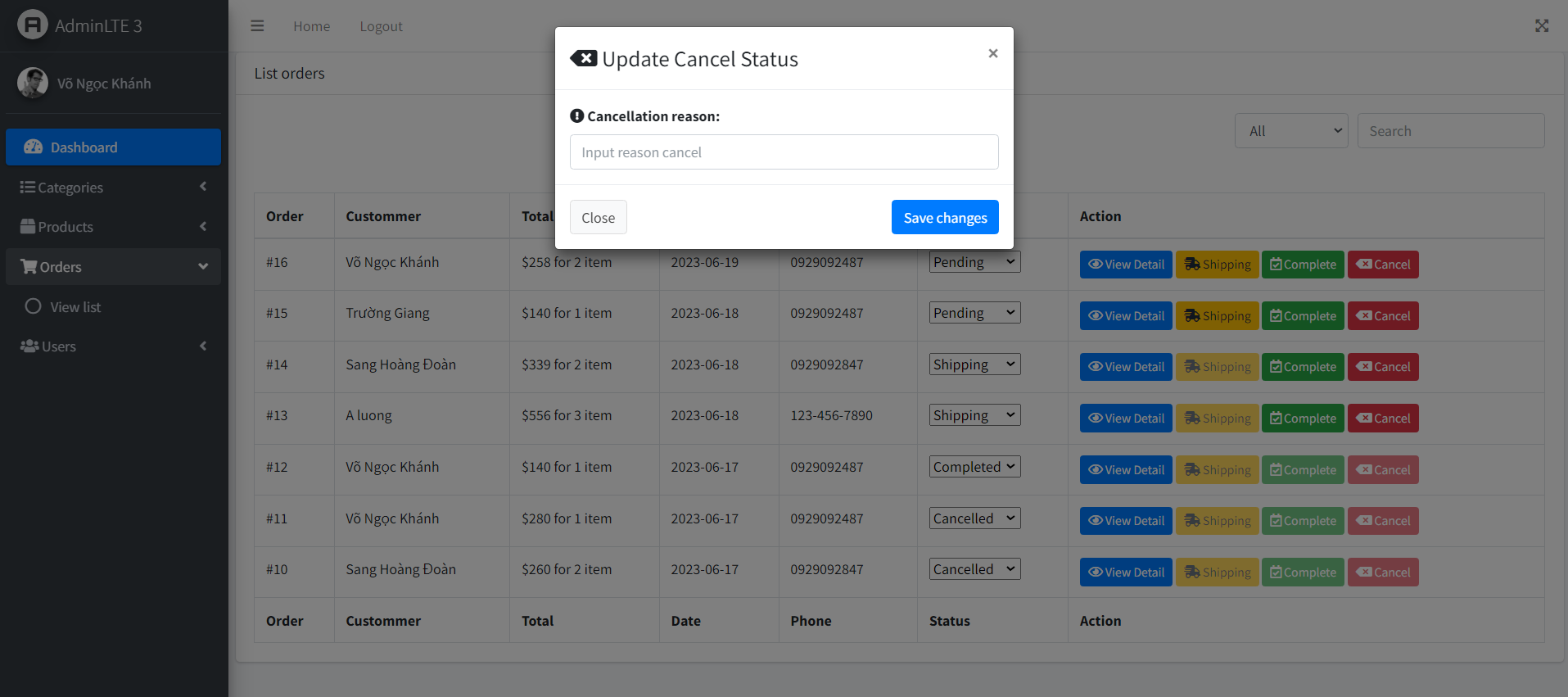
Hình 19: Quản lý đơn hàng

****

Hình 20: Xem chi tiết đơn hàng

****

Hình 21: Cập nhật trạng thái đang giao hàng

****

Hình 22: Hủy đơn hàng và nêu lý do

# **Chương 4: Kết luận và hướng phát triển**

## **4.1. Kết luận**

* Hiện tại, thì các chức năng của trang web đã thực hiện tương đối đầy đủ về qui trình đặt hàng và quản lý đơn hàng, tuy nhiên do thời gian có hạn chế nên em chỉ tập trung vào các chức năng chình, còn một số chức năng chưa được thực hiện.
* Trang web được xây dựng với giao diện thân thiện với người dùng và dễ dàng sử dụng.
* Xây dựng thành công website theo mô hình client-server. Sử dụng AngularJs làm hệ thống font-end và Spring boot framework để làm backend.
* Qua quá trình làm đồ án, em được tìm hiểu và học hỏi thêm nhiều về kinh nghiệm phân tích và thiết kế 1 dự án thực tế, giúp em có cái nhìn tổng quan hơn về quá trình xây dựng ứng dụng web và hệ thống font-end.

## **4.2. Hướng phát triển**

* Tiếp tục nghiên cứu để cải thiện hiệu xuất của trang web, tối ưu hóa trải nghiệm người dùng giúp website mang lại hiệu quả, chất lượng cao hơn.
* Phân tích kỹ hơn về logic nghiệp vụ trong qui trình shopping online
* Tiếp tục hoàn thiện các chức năng :
* Quản lý tài khoản : Quên mật khẩu, đổi mật khẩu xác minh qua email
* Thanh toán: thêm lựa chọn về đơn vị vận chuyển, phương thức thanh toán
* Thêm chức năng thông báo đơn hàng mới cho người dùng qua email, triển khai trải nghiệm người dùng xem thông báo đơn trên trang web.
* Mở rộng đăng ký, đăng nhập bằng tài khoản google, facebook.
* Chức năng đặt hàng: lựa chọn màu sắc, kích cỡ
* Bình luận, đánh giá sản phẩm,…

# **Tài liệu tham khảo**

* Tài liệu hướng dẫn AngularJs framework

[https://docs.angularjs.org/guide]

* Tài liệu hướng dẫn Spring boot framework [https://www.tutorialspoint.com/spring\_boot/index.htm]
* Tài liệu tìm hiểu mô hình MVC

[https://mona.software/mo-hinh-mvc/]

* Tài liệu do giảng viên cung cấp, biên soạn – Ths.Thái Thuận Thương