

BỘ KHOA HỌC VÀ CÔNG NGHỆ
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO BÀI TẬP
PHÁT TRIỂN HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ

Giảng viên: Kim Ngọc Bách

Đề tài : Xây dựng Website bán hàng thời trang
Nhóm bài tập lớn : Nhóm 16
Thành viên nhóm : Nguyễn Thu Huyền – B21DCCN445
Bùi Thị Thu – B21DCCN697
Chu Minh Tiến - B18DCCN529

Hà Nội – 2025

MỤC LỤC

LỜI CẢM ƠN	1
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	2
I. Giới thiệu đề tài	2
II. Mục tiêu đề tài	2
III. Mô hình kinh doanh	2
IV. Mô hình doanh thu	3
V. Kế hoạch thực thi, triển khai dự án	3
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG	5
1. HTML, CSS, JavaScript	5
2. Tailwind CSS	6
6. Database MongoDB	9
7. Nodemailer	10
8. Cloudinary	10
9. FPGrowth	10
CHƯƠNG 3: PHÂN TÍCH YÊU CẦU HỆ THỐNG	11
I. Yêu cầu chức năng và phi chức năng	11
1. Yêu cầu chức năng	11
1.1. Với khách hàng	11
1.2. Với người quản trị	11
2. Yêu cầu phi chức năng	11
II. Danh sách tác nhân và vai trò	12
III. Biểu đồ và đặc tả Use Case	12
1. Biểu đồ Use case tổng quan	12
2. Đặc tả chi tiết Use case	13
2.1. Đặt hàng online	13
2.2. Xem danh sách đơn hàng	14
2.3. Đổi/ hoàn trả hàng	15
2.4. Đánh giá sản phẩm	16
2.5. Quản lý sản phẩm	17
2.6. Quản lý đơn hàng	18

2.7. Quản lý yêu cầu đổi trả hàng	19
3. Thiết kế hệ thống	20
3.1. Biểu đồ lớp	20
3.2. Thiết kế cơ sở dữ liệu	20
CHƯƠNG 4: DEMO HỆ THỐNG	21
1. Giao diện người dùng	21
2. Giao diện quản trị	26
CHƯƠNG 5: TỔNG KẾT	28
1. Kết luận	28
2. Hướng phát triển trong tương lai	28

LỜI CẢM ƠN

Lời đầu tiên, nhóm 16 chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn - thầy Kim Ngọc Bách đã tận tâm trong việc dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia bộ môn Phát triển hệ thống thương mại điện tử, bọn em đã được tiếp thu những kiến thức giúp chúng em hiểu rõ hơn về quy trình phát triển hệ thống thương mại điện tử trong thực tế. Cảm ơn thầy đã giúp chúng em tiếp cận một cách dễ hơn với những kiến thức khó nhọc này. Mặc dù kiến thức bộ môn rất trừu tượng nhưng với cách giảng dạy đầy tâm huyết, cùng với những ví dụ minh họa dễ hiểu, cách dẫn dắt logic của thầy đã giúp chúng em có thể hiểu bài và tích lũy cho bản thân kiến thức cần có về phát triển hệ thống thương mại điện tử. Một lần nữa chúng em xin cảm ơn về những bài giảng vô cùng hay và tâm huyết của thầy.

Nhóm chúng em đã cùng nhau cố gắng từ bước phân tích chủ đề đến tìm hiểu, rồi cùng nhau xây dựng lên bài tập lớn về chủ đề này. Tuy nhiên thì chắc chắn chúng em vẫn sẽ còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều sai sót. Mặc dù nhóm đã cố gắng hết sức nhưng chắc chắn báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, mong thầy xem xét và góp ý để bài tập lớn của nhóm được hoàn thiện hơn.

Chúng em xin chân thành cảm ơn!

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI.

I. Giới thiệu đề tài

- Tên dự án: Xây dựng website bán hàng thời trang.
- Dự án xây dựng một website thương mại điện tử nhằm tạo ra kênh mua sắm trực tuyến các mặt hàng thời trang. Mục tiêu chính là giúp cung cấp trải nghiệm mua sắm thuận tiện, an toàn và nhanh chóng cho người tiêu dùng đồng thời cũng giúp quản lý hiệu quả danh mục sản phẩm, đơn hàng và khách hàng. Các mặt hàng chủ yếu là sản phẩm thời trang nam/ nữ, phụ kiện với tệp đối tượng khách hàng là người tiêu dùng cá nhân.
- Các tính năng chính của dự án:
 - + Tìm kiếm sản phẩm và khuyến nghị: Hỗ trợ người dùng tìm kiếm mặt hàng theo tên mặt hàng và sắp xếp các mặt hàng theo thứ tự giá bán tăng dần hoặc giảm dần. Hệ thống hiển thị cho người dùng các mặt hàng phù hợp nhất với người dùng dựa trên lịch sử mua hàng.
 - + Thanh toán trực tuyến: Hỗ trợ các hình thức thanh toán trực tuyến và trực tiếp (chuyển khoản ngân hàng, thanh toán khi nhận hàng).
 - + Quản lý giỏ hàng: Hệ thống cho phép người dùng thêm hoặc xóa các mặt hàng đã có trong giỏ hàng, xem thông tin các mặt hàng có trong giỏ bao gồm: tên, số lượng, giá tiền của từng mặt hàng và tổng giá trị của tất cả các mặt hàng trong giỏ.
 - + Đặt hàng và theo dõi đơn hàng: Hỗ trợ người dùng đặt hàng online và theo dõi được trạng thái đơn hàng.

II. Mục tiêu đề tài

- Xây dựng thành công website mua sắm thời trang với giao diện trực quan, dễ sử dụng trên cả máy tính và thiết bị truy cập giúp người dùng dễ dàng truy cập và mua hàng.
- Website cung cấp thông tin các sản phẩm chi tiết, rõ ràng giúp người dùng dễ dàng chọn được sản phẩm mong muốn tốt nhất.
- Hệ thống thanh toán trực tuyến nhanh chóng, an toàn đảm bảo thông tin khách hàng.
- Tích hợp hệ thống khuyến nghị sản phẩm giúp khách hàng tìm kiếm sản phẩm phù hợp, tăng cường trải nghiệm khách hàng.

III. Mô hình kinh doanh

- Dự án áp dụng mô hình kinh doanh B2C (Business to Consumer).
- Mô hình kinh doanh này giúp doanh nghiệp bán sản phẩm trực tiếp đến khách hàng là người tiêu dùng cá nhân qua việc đặt hàng trực tuyến trên website hoặc ứng dụng di động.

- Đây là mô hình với quy trình mua bán ngắn gọn, thường diễn ra nhanh chóng, dễ mở rộng online và tiếp cận được thị trường rộng lớn (nhiều khách hàng cá nhân).

IV. Mô hình doanh thu

- Nguồn doanh thu chính sẽ đến từ bán hàng trực tiếp: Lợi nhuận từ việc chênh lệch giữa giá nhập và giá bán của sản phẩm.
- Hoặc để tăng doanh số và giá trị trung bình đơn hàng, có thể triển khai các chương trình combo ưu đãi như: mua 1 tặng 1, mua theo combo sản phẩm,...

V. Kế hoạch thực thi, triển khai dự án

- Phạm vi dự án: Xây dựng hệ thống quản lý cửa hàng thời trang với 2 role:
 - + Khách hàng: Xem sản phẩm, đặt hàng, thanh toán, đánh giá.
 - + Admin: Quản lý sản phẩm, đơn hàng, yêu cầu đổi trả hàng.
- Kế hoạch thực hiện, triển khai

a, Tuần 1: Phân tích & Thiết kế hệ thống

- Mục tiêu: Hoàn thiện tài liệu thiết kế, wireframe, và cấu trúc cơ bản.

Thành viên	Công việc chi tiết
Cả nhóm	<ul style="list-style-type: none"> - Brainstorming tính năng chính - Phân tích yêu cầu người dùng - Thiết kế Use Case Diagram và User Flow
Front-end	<ul style="list-style-type: none"> - Cài đặt React + Tailwind CSS + Router • Trang đăng nhập/đăng ký • Trang quản trị (CRUD sản phẩm)
Back-end	<ul style="list-style-type: none"> - Thiết kế ERD MongoDB (: User, Product, Order, Review) - Cấu hình dự án Node.js + Express

b, Tuần 2: Phát triển hệ thống back end và front-end

- Mục tiêu: Hoàn thành 80% API backend và giao diện cốt lõi.

Thành viên	Công việc chi tiết
Cả nhóm	Test luồng đặt hàng/thanh toán
Back-end	<ul style="list-style-type: none"> - API Product: CRUD, upload ảnh (Cloudinary), lọc theo danh mục - API Cart: Thêm/xóa sản phẩm, tính tổng tiền
Front-end	<ul style="list-style-type: none"> - Xây dựng: <ul style="list-style-type: none"> • Trang chủ (hiển thị sản phẩm từ API) • Trang chi tiết sản phẩm

c, Tuần 3: Kết nối front-end và backend

- Mục tiêu: Kết nối frontend-backend, triển khai chức năng thanh toán

Thành viên	Công việc chi tiết
Cả nhóm	- Fix bug hiển thị thông báo lỗi
Front-end	- Trang giỏ hàng - Trang thanh toán - Trang quản trị (Dashboard đơn hàng)
Back-end	- API Order: Tạo đơn hàng, lịch sử đơn hàng - API Review: Thêm/xem đánh giá

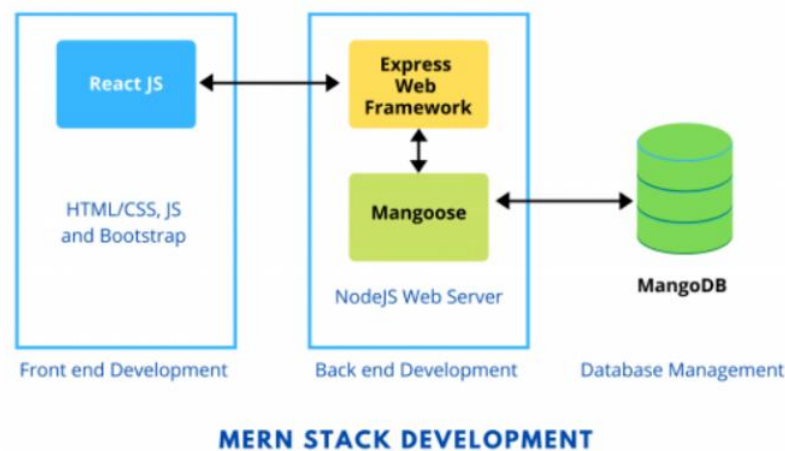
d, Tuần 4: Hoàn thiện và viết báo cáo

- Mục tiêu: Tối ưu UX/UI, deploy và chuẩn bị báo cáo.

Thành viên	Công việc chi tiết
Cả nhóm	- Viết báo cáo - Tổng hợp báo cáo (Google Docs) - Chuẩn bị slide
Front-end	- Responsive mobile
Back-end	Deploy backend (Render/Vercel)

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

Trong dự án này, đã nghiên cứu và sử dụng công nghệ MERN Stack để xây dựng trang web. MERN là một thuật ngữ rút gọn của MongoDB, Express.js, React.js và Node.js. Stack MERN là một stack Javascript được thiết kế để giúp phát triển ứng dụng web toàn ngăn xếp dễ dàng hơn và nhanh hơn. Tất cả bốn công nghệ này cung cấp một khuôn khổ hoàn chỉnh cho các nhà phát triển để tạo ra bất kỳ ứng dụng web nào. MERN đang tuân theo kiến trúc 3 tầng truyền thống, bao gồm tầng hiển thị front-end (React.js), tầng ứng dụng (Express.js và Node.js) và tầng cơ sở dữ liệu (MongoDB).



1. HTML, CSS, JavaScript

- HTML (HyperText Markup Language) là ngôn ngữ đánh dấu siêu văn bản được sử dụng để xây dựng và hiển thị các trang web trên Internet. Nó định nghĩa cấu trúc và ý nghĩa của nội dung trên trang web bằng cách sử dụng các thẻ và các thuộc tính.
- CSS (Cascading Style Sheets) là ngôn ngữ được sử dụng để mô tả cách mà các phần tử HTML được hiển thị trên trình duyệt. Nó cho phép lập trình viên điều chỉnh màu sắc, font chữ, khoảng cách, kích thước và nhiều thuộc tính khác của các phần tử trong trang web để tạo ra giao diện khách hàng hấp dẫn và thú vị. CSS giúp tách biệt phần nội dung của trang web và cách mà nó được hiển thị, giúp cho việc thiết kế và bảo trì trang web trở nên dễ dàng hơn.
- JavaScript là một ngôn ngữ lập trình phổ biến được sử dụng chủ yếu trong việc phát triển ứng dụng web. Được tích hợp trực tiếp vào trình duyệt web, JavaScript cho phép tương tác động và thay đổi nội dung của trang web một cách linh hoạt và đáp ứng. Nó được sử dụng để thêm các tính năng động như xử lý sự kiện, thao tác với DOM (Document Object Model), tạo và điều khiển hiệu ứng, gửi và nhận dữ liệu từ máy chủ,

và nhiều nhiệm vụ khác. JavaScript là một phần không thể thiếu trong phát triển web hiện đại, giúp tạo ra các trang web có trải nghiệm khách hàng tốt hơn và tương tác hơn.

2. Tailwind CSS

Tailwind CSS giúp xây dựng website một cách nhanh chóng nhất với các thuộc tính CSS đã được gán thành những class riêng, khi dùng chúng ta chỉ có việc gọi ra để dùng. Đây là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in mà chúng ta có thể dùng. Tailwind CSS có nhiều các class bao gồm các thuộc tính CSS khác nhau và quan trọng, chúng ta có thể dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

3. React.js

- React.js là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, được sử dụng để xây dựng giao diện khách hàng cho các ứng dụng web động. React.js tập trung vào việc tạo ra các thành phần giao diện tái sử dụng (reusable UI components), giúp tăng tính tái sử dụng, hiệu quả và dễ bảo trì của mã nguồn. React.js hoạt động bằng cách sử dụng Virtual DOM và mô hình one-way data binding. Nó xây dựng giao diện khách hàng thành các thành phần, sử dụng JSX để mô tả cấu trúc. Khi dữ liệu hoặc trạng thái thay đổi, React kích thích Reconciliation để cập nhật chỉ những phần thay đổi trong DOM thực sự, tối ưu hóa hiệu suất ứng dụng web.

- Một số điểm nổi bật về React.js:

- + Thành phần (Component): React.js xây dựng trên một kiến trúc dựa trên thành phần, trong đó mỗi phần của giao diện khách hàng được xem xét là một thành phần độc lập. Điều này giúp tạo ra mã nguồn gọn gàng, dễ hiểu và dễ bảo trì.

- + Virtual DOM (DOM Ảo): React.js sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM thực sự, được React.js sử dụng để thực hiện các thay đổi trên giao diện khách hàng một cách hiệu quả mà không cần phải cập nhật DOM thực sự.

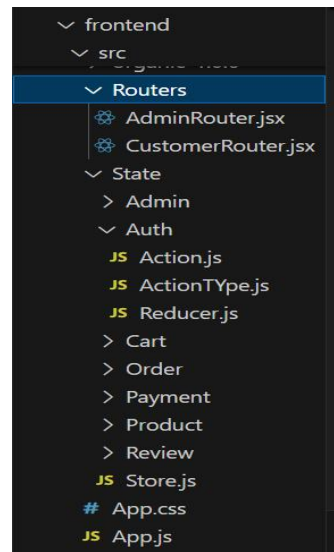
- + Tính năng JSX: JSX là một phần mở rộng của JavaScript cho phép viết HTML trong JavaScript, tạo ra mã nguồn dễ đọc và dễ bảo trì. JSX kết hợp HTML với JavaScript, giúp tạo ra các thành phần giao diện một cách dễ dàng.

- + Cộng đồng lớn và hỗ trợ mạnh mẽ: React.js có một cộng đồng lớn và mạnh mẽ, điều này có nghĩa là có nhiều tài liệu, hướng dẫn và thư viện bổ sung sẵn có để hỗ trợ việc phát triển ứng dụng React.

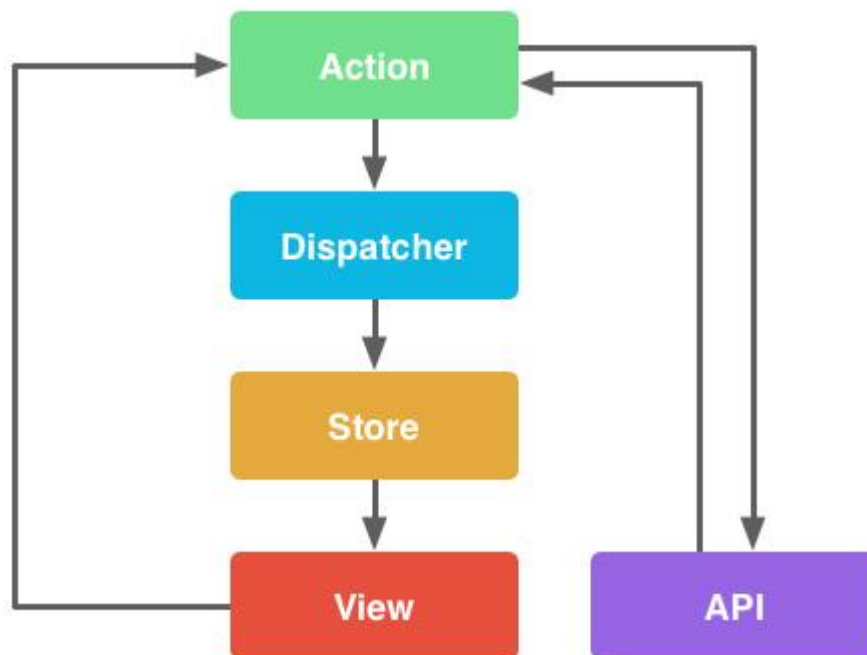
- React.js chạy độc lập, song song với phía Server, để xây dựng tầng Giao diện người dùng. Như đã nói, đây là lý do chúng tôi cần CORS và cấu hình CORS một cách cụ thể, thuận tiện việc gọi API, cũng như đảm bảo tính bảo mật.

- Dưới đây là kiến trúc phía Client:

+ Cấu trúc file, folder:



+ Kiến trúc Front-end sẽ là một tập các Route, được định tuyến đến các Page. Ngoài ra, nhóm sử dụng thêm thư viện Redux cho việc quản lý các State. Nhìn chung, kiến trúc phía Front-end của đề tài có thể mô tả đơn giản như mô hình dưới đây.



+ Routes sẽ có tới 3 loại, phụ thuộc vào Authentication, Authorization của người dùng

- ✓ Route: dành cho Khách, không cần đăng nhập
- ✓ PrivateRoute: dành cho Người mua, Người bán, cần đăng nhập
- ✓ AdminRoute: chỉ cho phép tài khoản Quản trị viên

4. Node.js

- Node.js là một nền tảng runtime mã nguồn mở xây dựng trên JavaScript Engine (V8 Engine) của Google Chrome. Nó cho phép lập trình viên viết mã JavaScript để chạy ở phía máy chủ mà không chỉ ở môi trường trình duyệt web. Node.js cung cấp một môi trường thực thi cho việc phát triển các ứng dụng mạng hiệu quả và mạnh mẽ.

- Một số điểm nổi bật về Node.js:

+ Non-blocking và Asynchronous I/O: Node.js sử dụng mô hình không chặn (nonblocking) và I/O bất đồng bộ (asynchronous I/O), cho phép xử lý nhiều yêu cầu một cách hiệu quả và đồng thời mà không phải chờ đợi kết quả của mỗi yêu cầu trước đó hoàn thành.

+ Sự mở rộng dễ dàng: Node.js cho phép xây dựng các ứng dụng có khả năng mở rộng tốt nhờ vào mô hình sự kiện và luồng không đồng bộ. Điều này làm cho việc xử lý hàng trăm hoặc thậm chí hàng nghìn kết nối cùng một lúc trở nên khả thi và hiệu quả.

+ Phát triển nhanh chóng: Với cộng đồng lớn và sự phát triển liên tục, Node.js cung cấp nhiều công cụ và thư viện để giúp phát triển ứng dụng nhanh chóng và dễ dàng. Các framework như Express.js cũng giúp giảm bớt công việc lặp đi lặp lại và tăng hiệu suất phát triển.

+ Phù hợp cho ứng dụng thời gian thực: Node.js thích hợp cho việc xây dựng các ứng dụng thời gian thực như ứng dụng chat, trò chơi trực tuyến, các ứng dụng phát trực tuyến và nhiều ứng dụng khác cần phản hồi nhanh và xử lý đa khách hàng.

5. Express.js

- Express.js là một framework phổ biến được sử dụng để xây dựng ứng dụng web và API thông qua Node.js. Nền tảng được xem là một phương thức xử lý các yêu cầu HTTP, quản lý các tuyến đường, xử lý phần mềm trung gian và nhiều tính năng khác để phát triển hiệu quả ứng dụng web. [7]

- Express.js là một mã nguồn mở hoàn toàn miễn phí. Điều này cho phép các nhà phát triển xây dựng web và API ứng dụng mà không phải trả bất kỳ khoản phí nào khi sử dụng framework này. Công nghệ mang đến điều kiện thuận lợi cho cả những dự án lớn và nhỏ có nhu cầu sử dụng Express.js mà không cần phải lo lắng về các chi phí phát triển.

- Một số điểm nổi bật về Express.js:

+ Tốc độ phát triển: Express.js sở hữu cấu trúc linh hoạt và đơn giản, giúp phát triển trang web trở nên nhanh chóng hơn. Việc có sẵn các tính năng xử lý các yêu cầu HTTP, quản lý các tuyến đường và phần mềm trung gian giúp nhà phát triển tập trung vào việc phát triển chức năng chính của trang web mà không cần phải lo lắng về cấu trúc cơ bản.

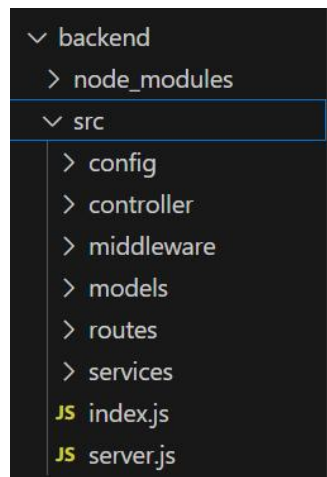
+ Hỗ trợ phần mềm trung gian: Express.js cho phép tích hợp nhiều phần mềm trung gian từ bên ngoài, mở rộng khả năng của trang web thông qua các tính năng bổ sung như khách hàng xác thực, xử lý lỗi, quản lý phiên bản và nhiều tính năng khác mà không cần thiết viết lại từ đầu.

+ Hỗ trợ API và định tuyến: Express.js được sử dụng rộng rãi để xây dựng RESTful API, giúp tạo các API điểm cuối một cách nhanh chóng và dễ dàng. Ngoài ra, công việc quản lý định tuyến trong Express cũng rất linh hoạt, giúp phân chia trang web thành các module và quản lý URL một cách dễ dàng.

+ Cộng đồng mạnh mẽ: Express.js có cộng đồng lớn với rất nhiều tài liệu, khả năng mở rộng gói đa dạng và cơ hội hỗ trợ từ cộng đồng. Những điều này đều có tác dụng xây dựng trang web nhanh chóng và hiệu quả.

- Dưới đây là kiến trúc phía Server:

+ Khởi tạo dự án, Cấu trúc file, folder



+ Khởi tạo ứng dụng, cấu hình Port và kết nối với Cơ sở dữ liệu

```
const express = require('express');

//app
const app = express();

//port
const port = process.env.PORT || 8000;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});

//db
mongoose.connect(process.env.DATABASE, (error) => {
  if (error) throw error;
  console.log('DB connected!');
});
```

6. Database MongoDB

- MongoDB là một hệ thống quản lý cơ sở dữ liệu NoSQL mã nguồn mở phổ biến, được thiết kế để xử lý, lưu trữ và truy xuất dữ liệu một cách linh hoạt và hiệu quả.

- MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.
- MongoDB có nhiều ưu điểm như khả năng lưu trữ dữ liệu phân tán, linh hoạt trong cấu trúc dữ liệu, có thể mở rộng, tốc độ truy vấn nhanh và hỗ trợ các tính năng như indexing, replication, sharding và map-reduce.

7. Nodemailer

- Nodemailer là một thư viện Node.js giúp gửi email dễ dàng bằng cách tạo một “transporter” (bộ vận chuyển) và gửi email với các thông số như người gửi, người nhận, chủ đề và nội dung
- Cấu trúc chính gồm 3 thành phần:
 - + Transporter: Là đối tượng kết nối tới dịch vụ email (SMTP server, Gmail, Mailgun,...).
 - + Mail Options: là đối tượng chứa thông tin email cần gửi (to, from, subject, content,...).
 - + SendMail: hàm gửi email với transporter và mailOptions.
- Trong hệ thống này sử dụng Nodemailer để gửi email tới khách hàng khi đặt hàng thành công.

8. Cloudinary

- Cloudinary là một dịch vụ đám mây phổ biến, cung cấp các giải pháp để quản lý, lưu trữ và tối ưu hóa, phân phát hình ảnh và video trên web và ứng dụng di động.
- Các tính năng nổi bật:
 - + Upload: tải lên file ảnh từ trình duyệt, server hoặc qua API. Hỗ trợ upload ảnh trực tiếp từ URL hoặc từ máy người dùng.
 - + Transformations: Thay đổi kích thước, cắt, xoay, nén. Thêm bộ lọc, watermark, tự động phát hiện khuôn mặt để crop, gắn thẻ AI.
 - + Video Processing: Chuyển đổi định dạng, cắt video, chỉnh bitrate. Tạo thumbnail video.
 - + Optimization: Tự động nén, chọn định dạng tối ưu (WebP, AVIF). Lazy loading cho trang web.

9. FPGrowth

Sử dụng FPGrowth để đưa ra các sản phẩm thường được khách hàng mua cùng với sản phẩm hiện tại với độ tin cậy là 40% ($\text{min_sup} = 0.4$). So với thuật toán Priori truyền thống thì thuật toán này không tạo các itemset trung gian nên giảm thời gian xử lý và sử dụng ít bộ nhớ hơn.

CHƯƠNG 3: PHÂN TÍCH YÊU CẦU HỆ THỐNG

I. Yêu cầu chức năng và phi chức năng

1. Yêu cầu chức năng

1.1. Với khách hàng

- Đăng ký tài khoản.
- Đăng nhập vào hệ thống.
- Xem danh sách sản phẩm.
- Xem được chi tiết thông tin sản phẩm.
- Tìm kiếm sản phẩm theo tên, size, giá.
- Thêm, sửa, xóa sản phẩm trong giỏ hàng.
- Đặt hàng và theo dõi trạng thái đơn hàng.
- Hỗ trợ hoàn trả hàng: Khách hàng được đổi/ trả hàng khi sản phẩm lỗi, bẩn, giao sai hoặc sản phẩm không giống mô tả.
- Đánh giá và bình luận sản phẩm.
- Chức năng gợi ý sản phẩm.

1.2. Với người quản trị

- Quản lý sản phẩm: thêm, sửa, xóa sản phẩm.
- Quản lý đơn hàng: xác nhận/ hủy, cập nhật trạng thái đơn hàng.
- Quản lý yêu cầu đổi/ hoàn trả hàng.

2. Yêu cầu phi chức năng

- Yêu cầu hiệu quả
 - + Hệ thống phải có hiệu suất cao, xử lý đơn hàng nhanh chóng.
 - + Thời gian phản hồi dưới 3 giây cho các thao tác chính
- Yêu cầu tương thích: hệ thống phải hoạt động tốt trên cả máy tính và thiết bị di động.
- Yêu cầu hệ thống:
 - + Bảo mật: phân quyền hệ thống cho từng đối tượng sử dụng tránh tình trạng thông tin, dữ liệu bị mất hoặc bị sửa đổi
- Yêu cầu công nghệ:
 - + Hệ thống phải dễ sử dụng, dễ sửa lỗi và có khả năng tái sử dụng và kế thừa
 - + Hệ thống đảm bảo phục vụ truy cập online trên 50 người cùng lúc
 - + Giao diện phần cứng quản lý: Window 7 trở lên, CPU lớn
 - + Sử dụng cơ sở dữ liệu quan hệ (SQL hoặc NoSQL tùy theo yêu cầu cụ thể).
- Yêu cầu an toàn
 - + Phải có giải pháp bảo đảm an toàn dữ liệu
 - + Có khả năng backup và phục hồi dữ liệu khi có sự cố
- Yêu cầu về pháp lý:

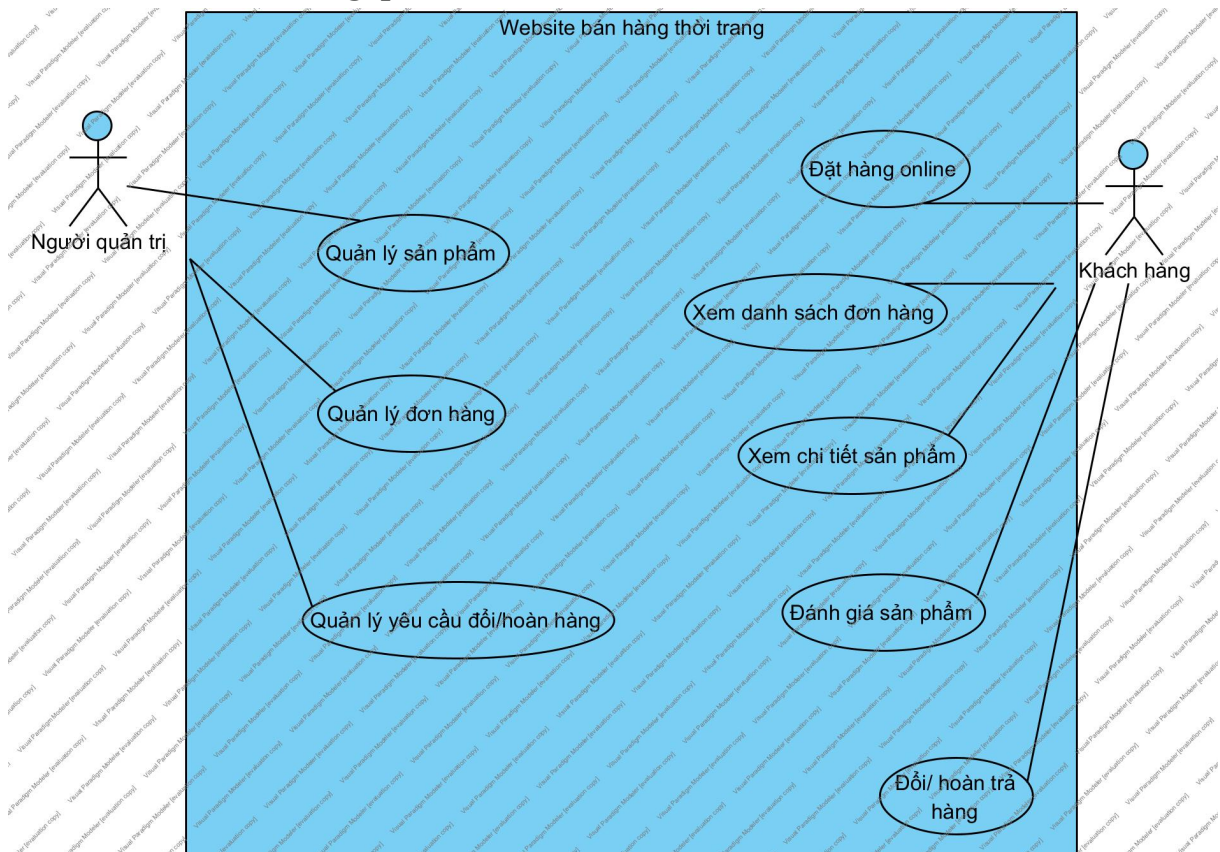
- + Thông tin người dùng được bảo vệ
- + Hệ thống phải tuân thủ các quy định về bảo mật dữ liệu.
- + Đảm bảo quyền riêng tư của khách hàng theo chính sách GDPR hoặc các quy định địa phương khác.

II. Danh sách tác nhân và vai trò

Tác nhân	Vai trò
Khách hàng	Người đăng nhập vào website để mua hàng, được hỗ trợ: <ul style="list-style-type: none"> + Tìm kiếm, xem sản phẩm + Thêm sản phẩm vào giỏ hàng + Đặt hàng và thanh toán trực tuyến (có hỗ trợ thanh toán khi nhận hàng) + Đánh giá/ bình luận sản phẩm + Xem lịch sử đặt hàng + Đổi/ hoàn trả hàng
Người quản trị	Người có quyền kiểm soát hệ thống <ul style="list-style-type: none"> + Quản lý sản phẩm + Quản lý đơn hàng + Quản lý hoàn trả hàng

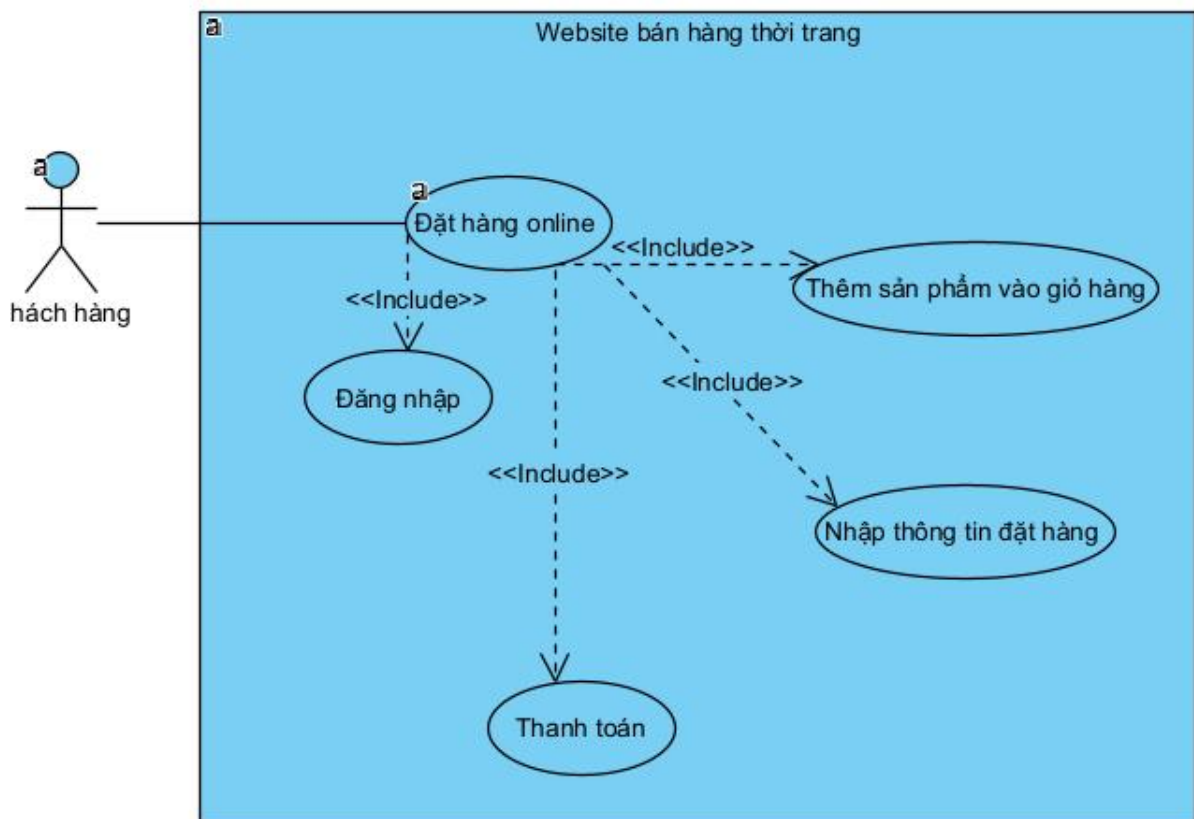
III. Biểu đồ và đặc tả Use Case

1. Biểu đồ Use case tổng quan



2. Đặc tả chi tiết Use case

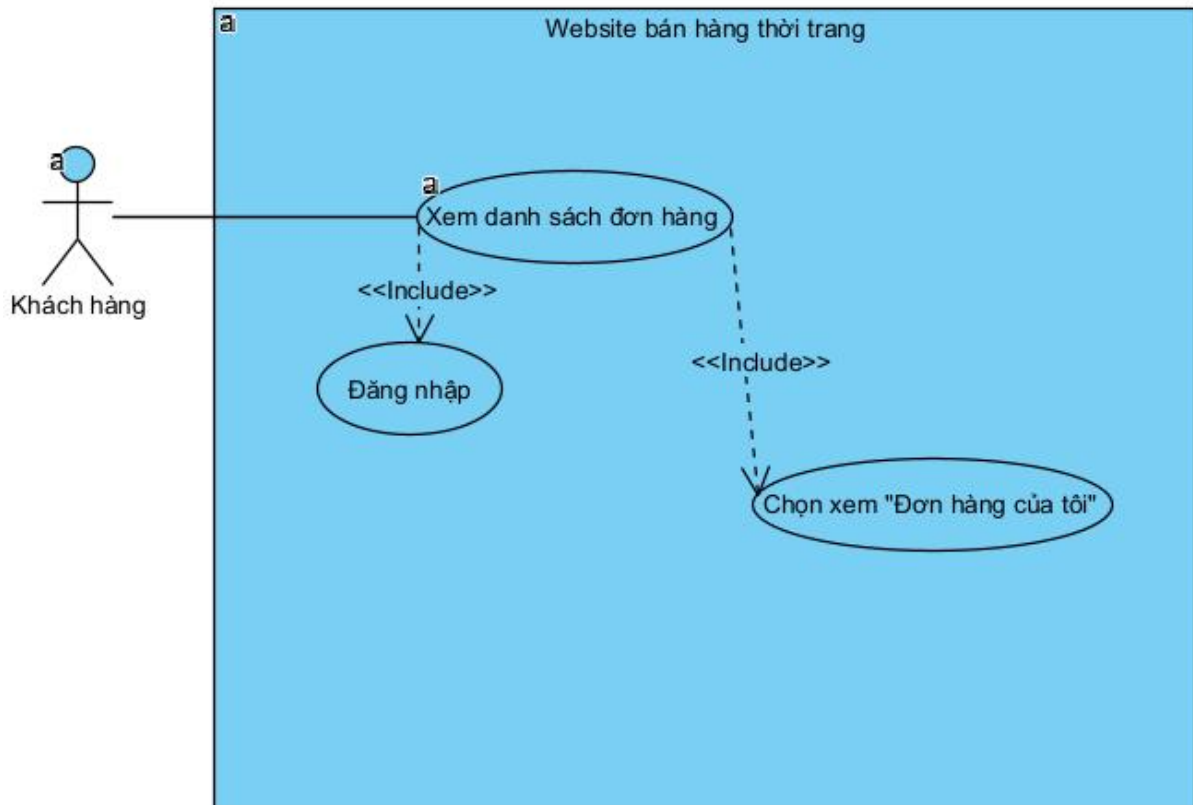
2.1. Đặt hàng online



Use case	Đặt hàng online
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng đăng nhập thành công vào hệ thống
Hậu điều kiện	Khách hàng đặt hàng online thành công
Kịch bản chính	<ol style="list-style-type: none"> 1. Khách hàng chọn sản phẩm muốn mua trong danh sách sản phẩm. 2. Hệ thống hiển thị giao diện chi tiết sản phẩm. 3. Khách hàng chọn số lượng và size sau đó chọn “Thêm vào giỏ hàng” 4. Hệ thống hiển thị thông báo thêm vào giỏ hàng thành công 5. Khách hàng vào chọn vào giỏ hàng để xem chi tiết giỏ hàng 6. Hệ thống hiển thị giao diện giỏ hàng 7. Khách hàng chọn “Thanh toán” 8. Hệ thống hiển thị giao diện đặt hàng để khách hàng nhập thông tin địa chỉ của mình. 9. Khách hàng nhập thông tin địa chỉ của mình và click “Xác nhận địa chỉ” 10. Sau khi xác nhận địa chỉ, hệ thống chuyển qua giao diện để khách hàng chọn hình thức thanh toán. 11. Khách hàng điền thông tin về tài khoản thanh toán

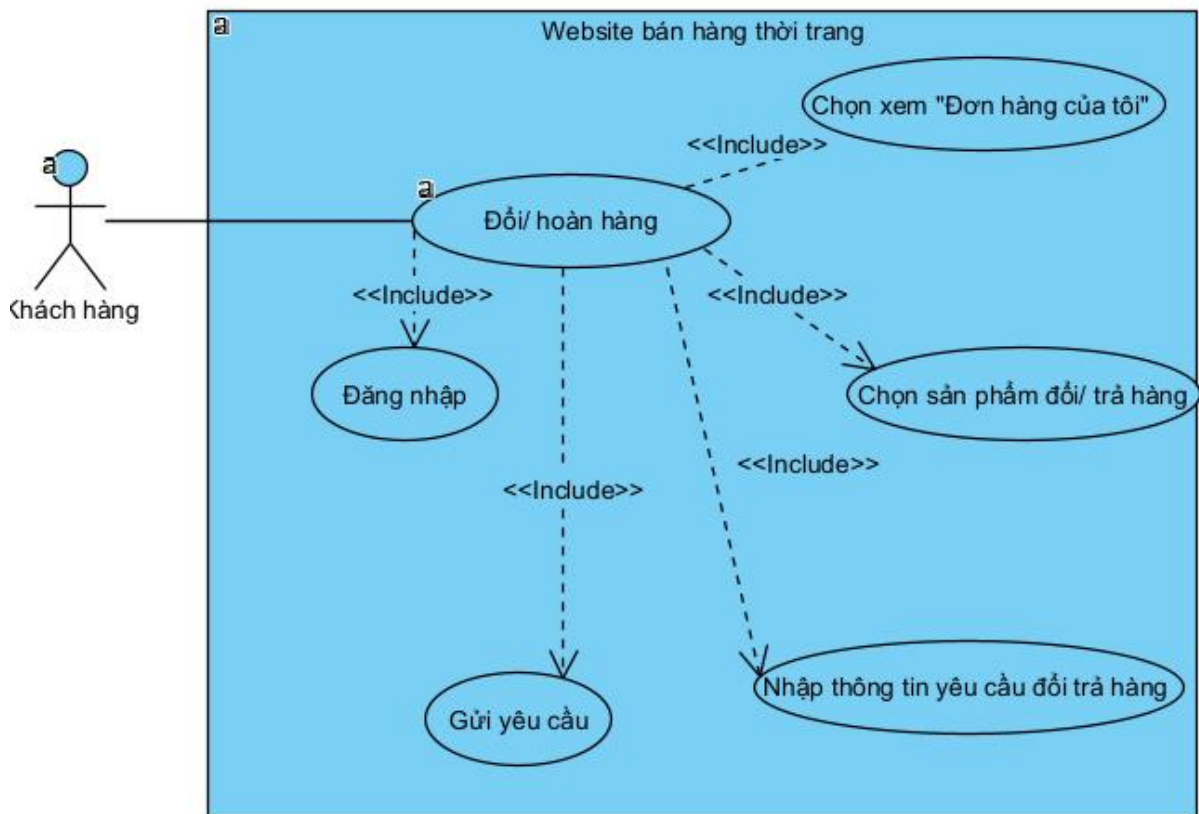
	12. Hệ thống hiển thị thông báo đặt hàng thành công
Ngoại lệ	Không có sản phẩm.

2.2. Xem danh sách đơn hàng



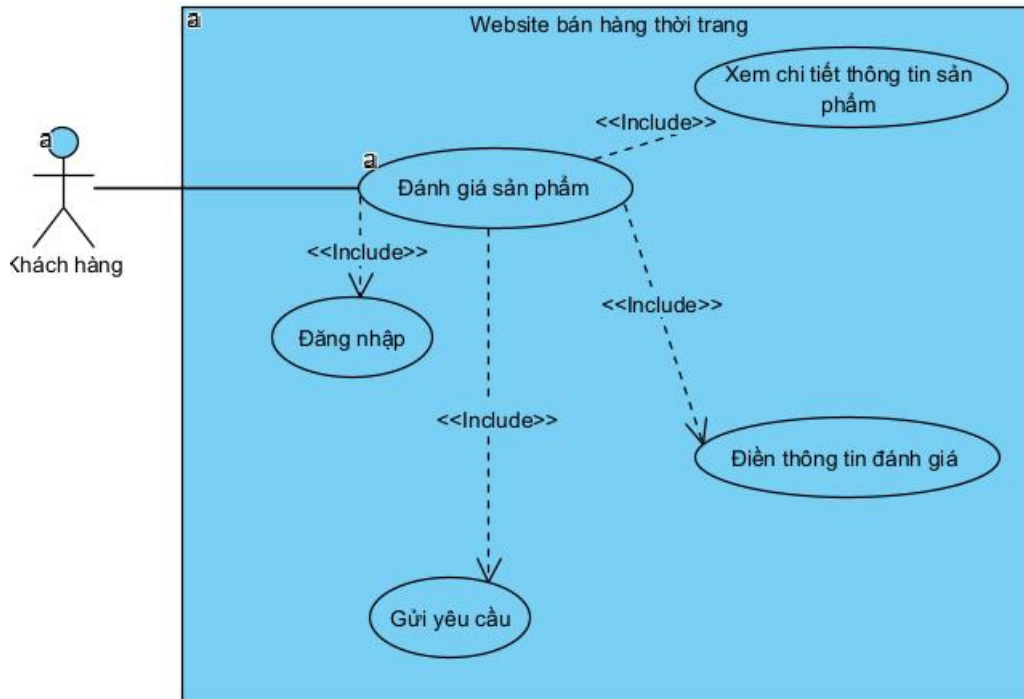
Use case	Xem danh sách đơn hàng
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng đăng nhập thành công vào hệ thống
Hậu điều kiện	Khách hàng xem được danh sách đơn hàng đã đặt
Kịch bản chính	<ol style="list-style-type: none"> Khách hàng chọn “Đơn hàng của tôi” Hệ thống hiển thị danh sách đơn hàng đã đặt với các thông tin: sản phẩm, số lượng, giá tiền, trạng thái đơn, thanh toán, yêu cầu đổi trả.
Ngoại lệ	2.1. Không có đơn hàng nào

2.3. Đổi/ hoàn trả hàng



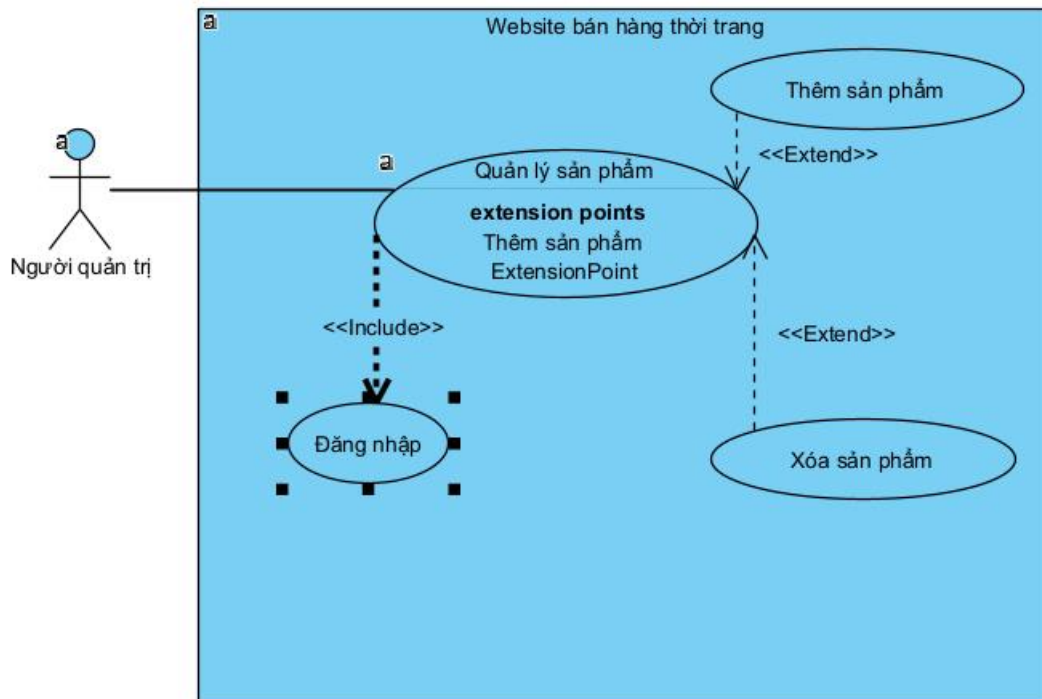
Use case	Hoàn trả hàng
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng đăng nhập thành công vào hệ thống
Hậu điều kiện	Khách hàng gửi thành công yêu cầu đổi/ hoàn hàng
Kịch bản chính	<ol style="list-style-type: none"> 1. Khách hàng chọn “Đơn hàng của tôi” 2. Hệ thống hiển thị danh sách đơn hàng đã đặt với các thông tin: sản phẩm, số lượng, giá tiền, trạng thái đơn, thanh toán, yêu cầu đổi trả. 3. Khách hàng click “Yêu cầu đổi trả” 4. Hệ thống hiển thị form nhập thông tin về yêu cầu đổi trả gồm thông tin sản phẩm đổi trả, loại yêu cầu, lý do, mô tả. 5. Khách hàng nhập loại yêu cầu, lý do đổi trả và mô tả chi tiết, sau đó click “Gửi yêu cầu” 6. Hệ thống hiển thị thông báo thành công
Ngoại lệ	2.1. Không có đơn hàng nào

2.4. Đánh giá sản phẩm



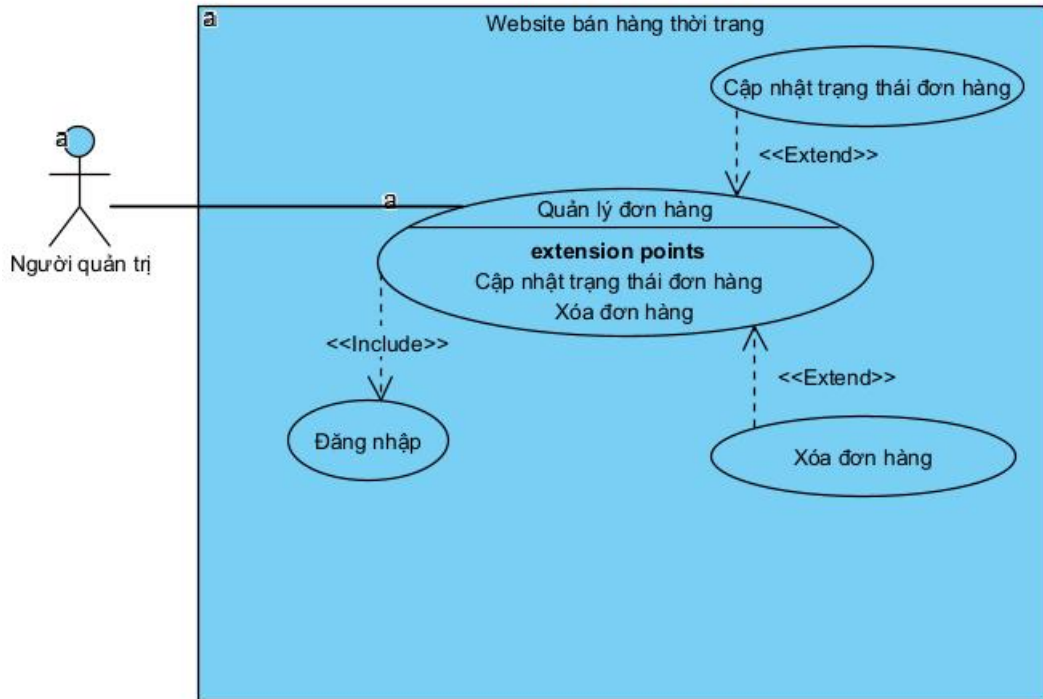
Use case	Đánh giá sản phẩm
Tác nhân chính	Khách hàng
Tiền điều kiện	Khách hàng đăng nhập thành công vào hệ thống
Hậu điều kiện	Đánh giá của khách hàng hiển thị lên hệ thống
Kịch bản chính	1. Khách hàng chọn sản phẩm muốn đánh giá trong danh sách sản phẩm 2. Hệ thống hiển thị giao diện trang thông tin chi tiết về sản phẩm đó 3. Khách hàng nhập thông tin đánh giá vào phần đánh giá sản phẩm và click “Đánh giá” 4. Hệ thống hiển thị đánh giá của khách hàng trên hệ thống
Ngoại lệ	

2.5. Quản lý sản phẩm



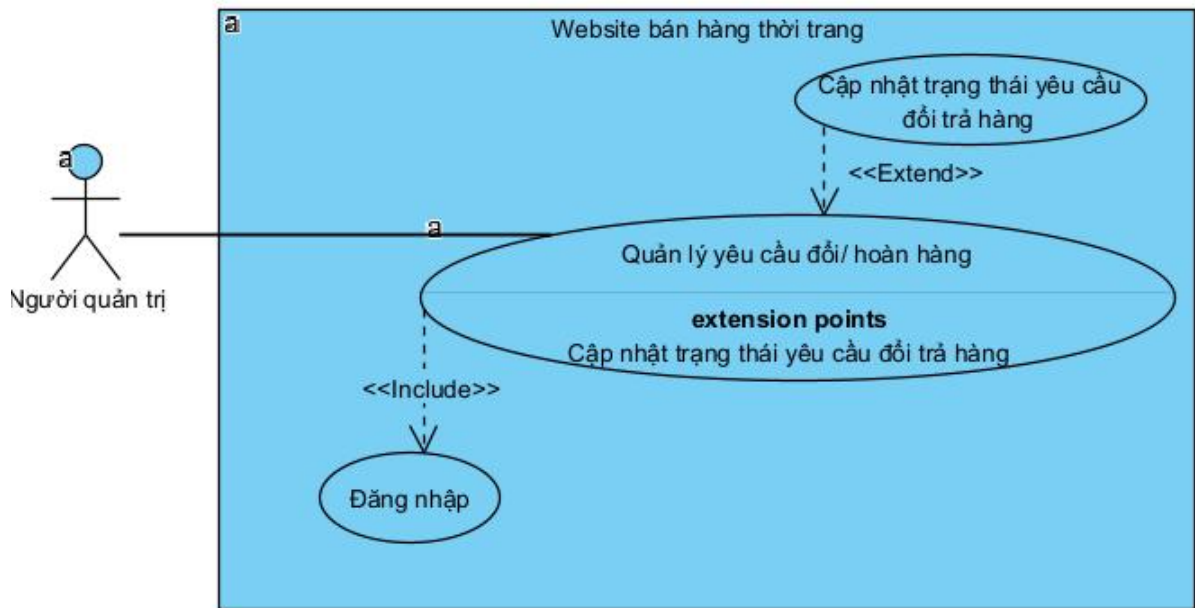
Use case	Quản lý sản phẩm
Tác nhân chính	Người quản trị
Tiền điều kiện	Người quản trị đăng nhập thành công vào hệ thống
Hậu điều kiện	Người quản trị thêm/ xóa sản phẩm thành công
Kịch bản chính	1. Người quản trị chọn “Quản lý sản phẩm” 2. Hệ thống hiển thị ra danh sách sản phẩm 3. Người quản trị nhập thông tin để thêm mới sản phẩm hoặc xóa sản phẩm trong danh sách 4. Hệ thống hiển thị thông báo thành công
Ngoại lệ	2.1. Không có sản phẩm

2.6. Quản lý đơn hàng



Use case	Quản lý đơn hàng
Tác nhân chính	Người quản trị
Tiền điều kiện	Người quản trị đăng nhập thành công vào hệ thống
Hậu điều kiện	Người quản trị cập nhật/ xóa thành công đơn hàng
Kịch bản chính	1. Người quản trị chọn “Quản lý đơn hàng” 2. Hệ thống hiển thị danh sách đơn hàng 3. Người quản trị thực hiện cập nhật hoặc xóa đơn hàng 4. Hệ thống hiển thị thông báo thành công
Ngoại lệ	4.1. Không có đơn hàng nào

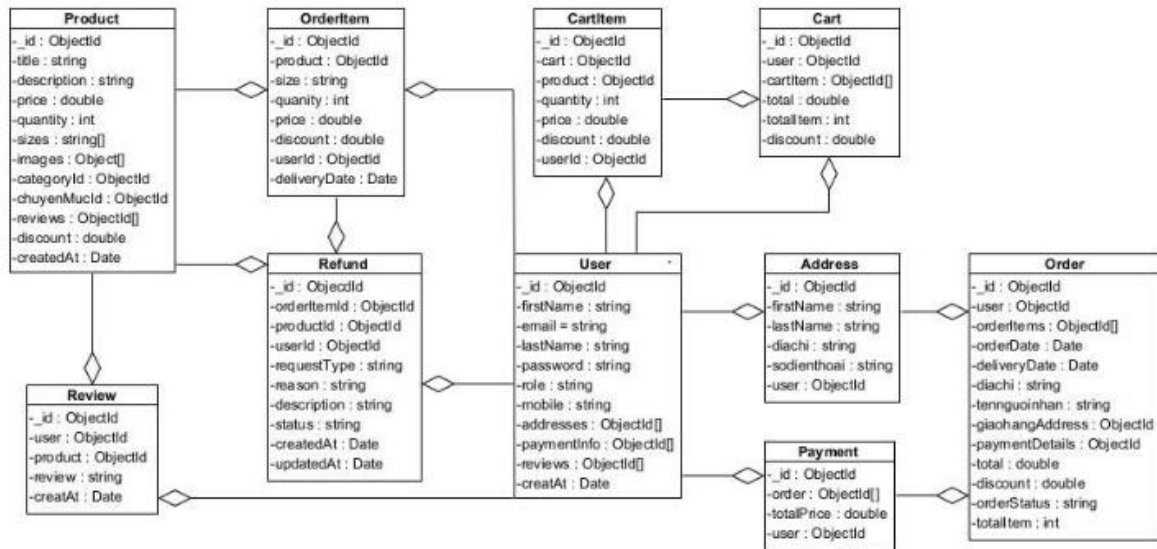
2.7. Quản lý yêu cầu đổi trả hàng



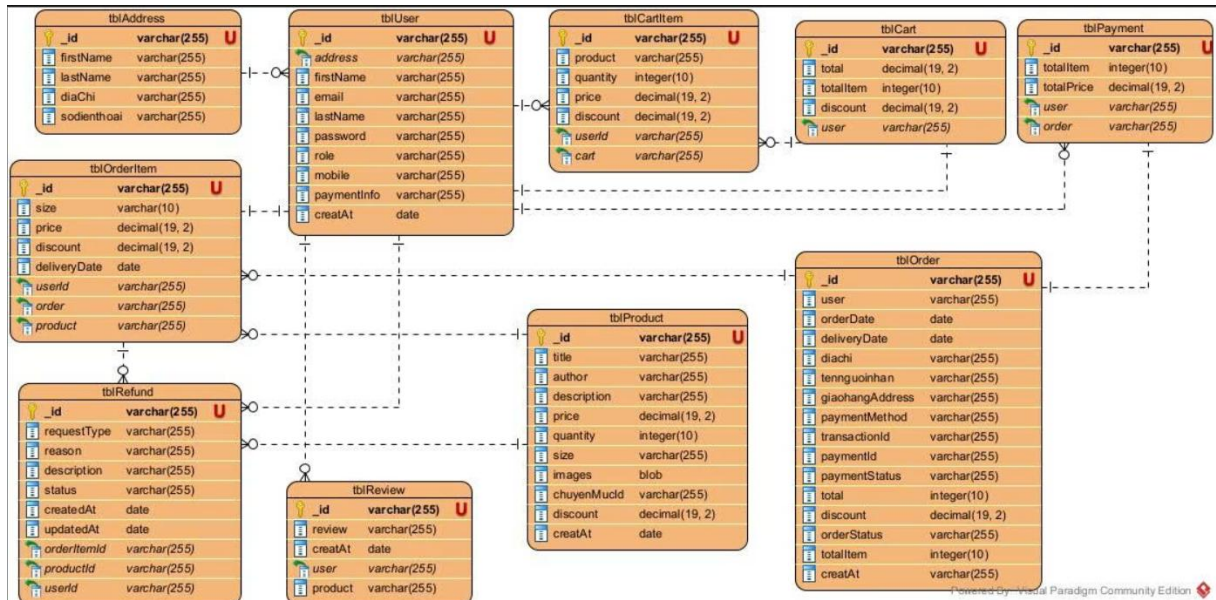
Use case	Quản lý đổi trả hàng
Tác nhân chính	Người quản trị
Tiền điều kiện	Người quản trị đăng nhập thành công vào hệ thống
Hậu điều kiện	Người quản trị cập nhật trạng thái yêu cầu đổi/ trả hàng thành công
Kịch bản chính	<ol style="list-style-type: none"> 1. Người quản trị chọn “Đổi/ trả hàng” 2. Hệ thống hiển thị danh sách các yêu cầu đổi trả hàng. 3. Người quản trị click “Cập nhật” để cập nhật trạng thái các yêu cầu đổi/ trả hàng 4. Hệ thống hiển thị thông báo thành công
Ngoại lệ	2.1. Không có yêu cầu đổi/ trả nào

3. Thiết kế hệ thống

3.1. Biểu đồ lớp



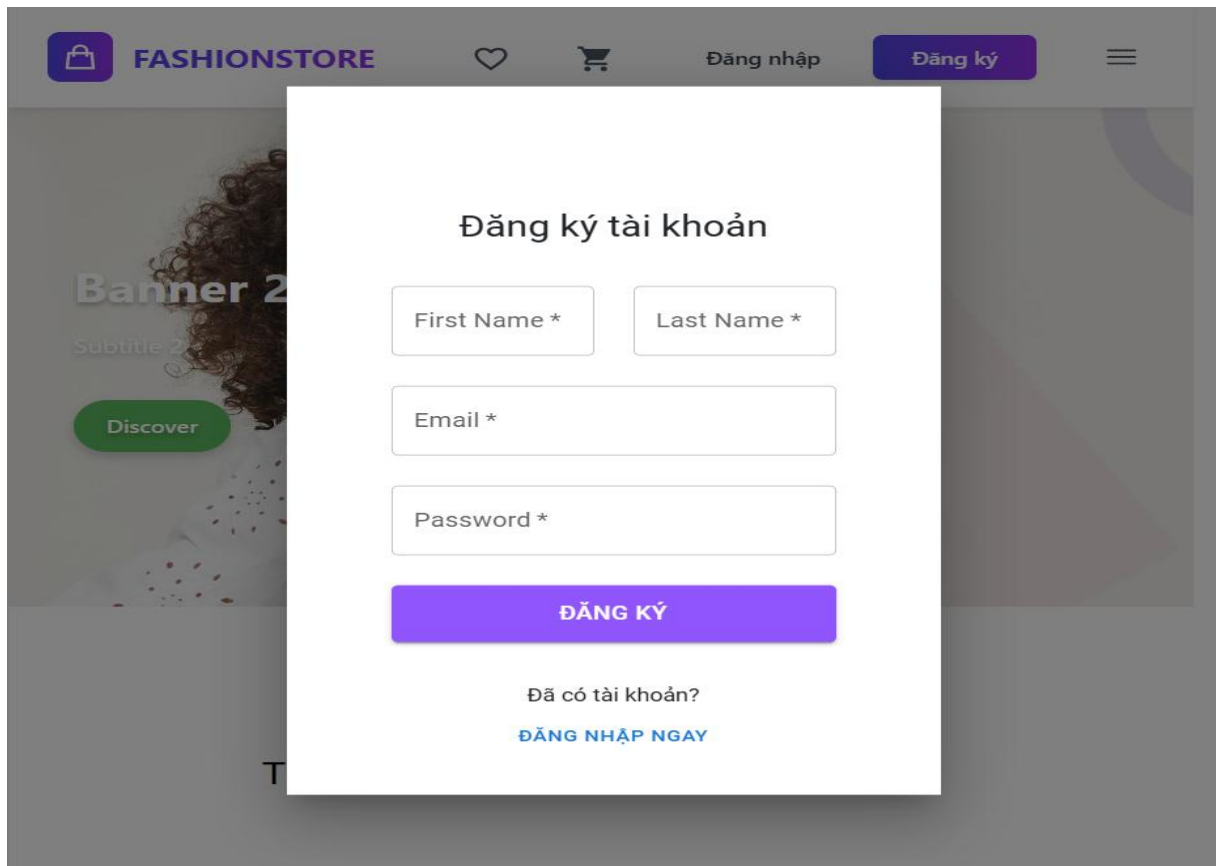
3.2. Thiết kế cơ sở dữ liệu



CHƯƠNG 4: DEMO HỆ THỐNG

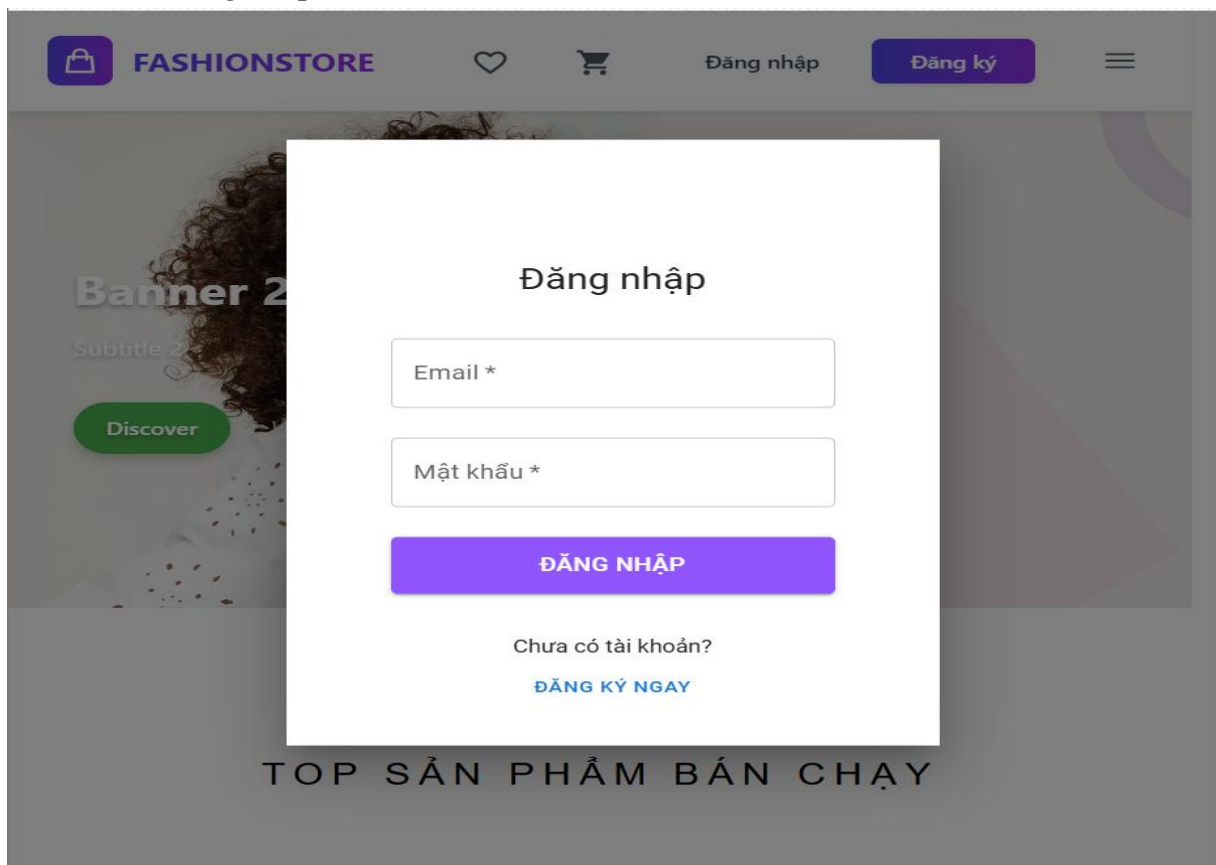
1. Giao diện người dùng

- Giao diện đăng ký tài khoản



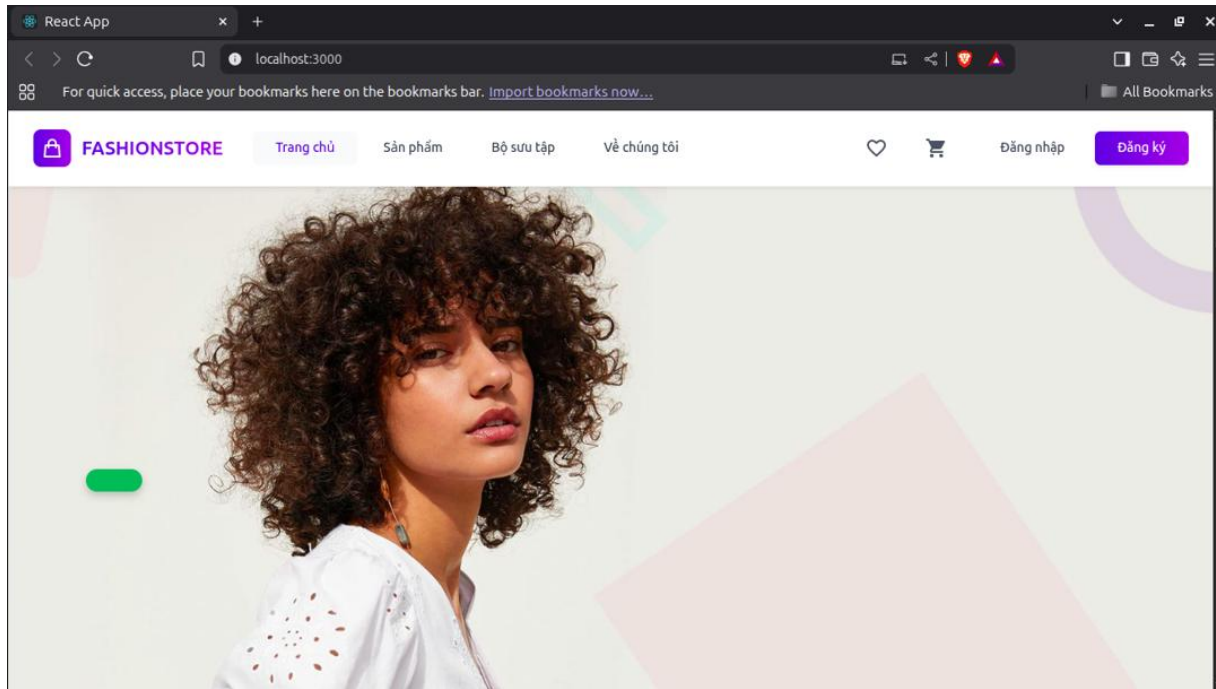
The screenshot shows the registration form for FashionStore. The form is titled "Đăng ký tài khoản" and is centered on the page. It includes input fields for "First Name *", "Last Name *", "Email *", and "Password *". Below these fields is a purple button labeled "ĐĂNG KÝ". At the bottom of the form, there is a link that says "Đã có tài khoản? ĐĂNG NHẬP NGAY". The background of the page shows a banner with a woman's face and the text "Banner 2" and "Discover".

- Giao diện đăng nhập

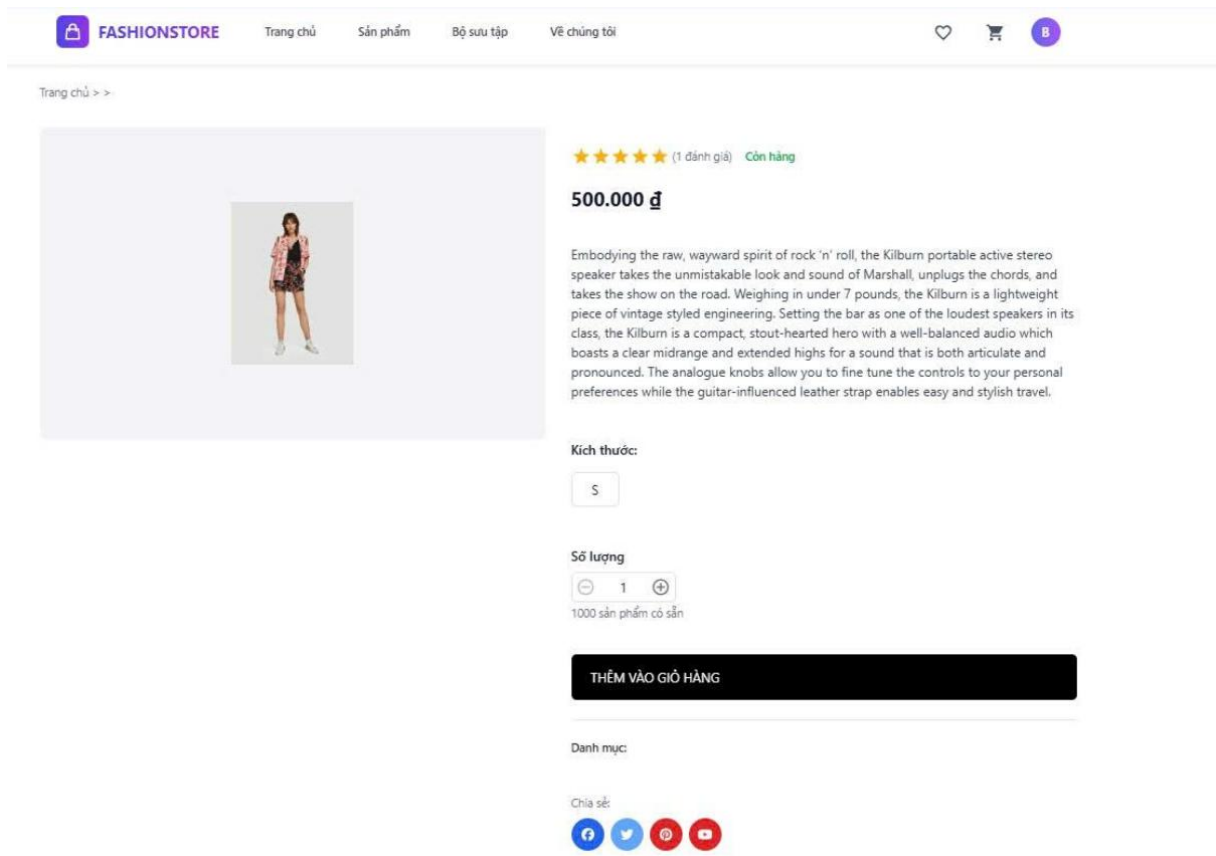


The screenshot shows the login form for FashionStore. The form is titled "Đăng nhập" and is centered on the page. It includes input fields for "Email *" and "Mật khẩu *". Below these fields is a purple button labeled "ĐĂNG NHẬP". At the bottom of the form, there is a link that says "Chưa có tài khoản? ĐĂNG KÝ NGAY". The background of the page shows a banner with a woman's face and the text "Banner 2" and "Discover". At the bottom of the page, there is a section titled "TOP SẢN PHẨM BÁN CHẠY".

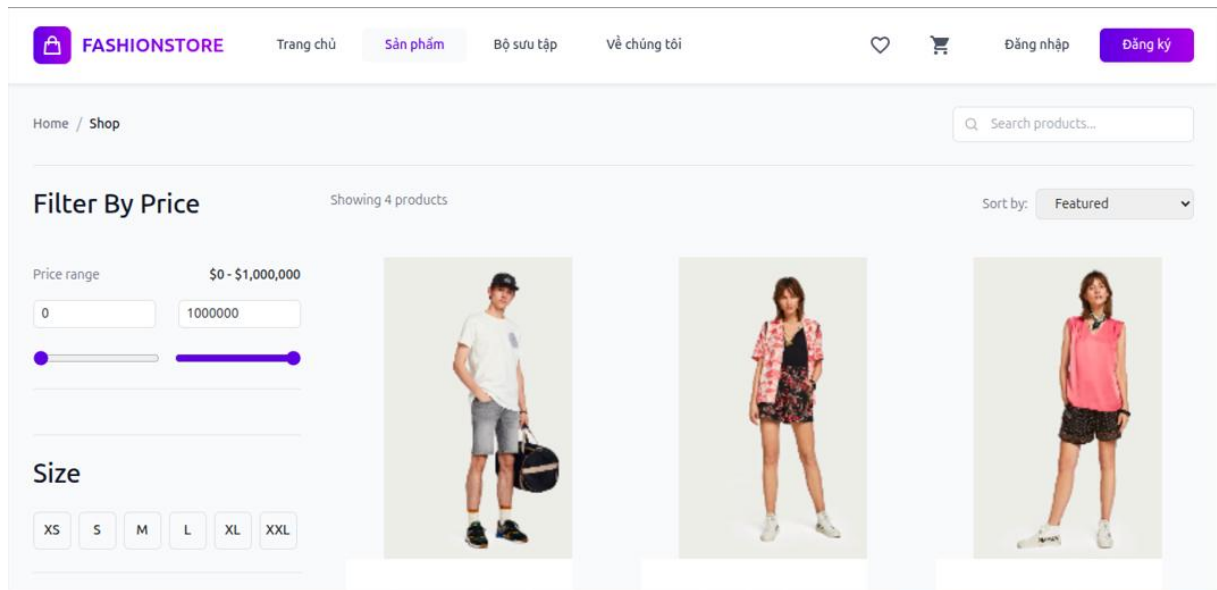
- Giao diện trang chủ



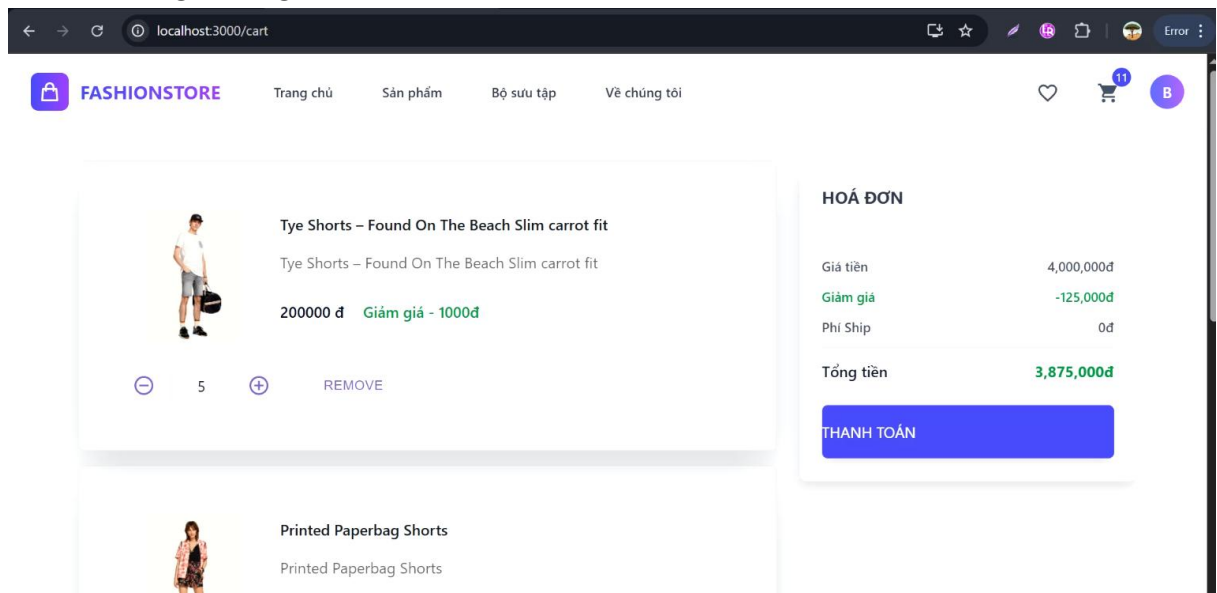
- Giao diện xem chi tiết sản phẩm



- Giao diện danh sách sản phẩm



- Giao diện giỏ hàng



- Giao diện đánh giá sản phẩm

ĐÁNH GIÁ SẢN PHẨM

Đánh giá sản phẩm

Đánh giá *

ĐÁNH GIÁ

Bùi Thị Thu
2025-06-08T11:39:39.514Z
hay

Bùi Thị Thu
2025-06-06T08:52:18.797Z
Sản phẩm tốt

1

2. Giao diện quản trị

- Giao diện quản lý sản phẩm

Hình ảnh	Tên sản phẩm	Số lượng	Danh mục	Giá	Mã sản phẩm	Xóa
	Tye Shorts – Found On The Beach Slim carrot fit			200000	#6842783ee189cc1654c99bf7	XOA
	Printed Paperbag Shorts			500000	#6842a45d797ed2da23215deb	XOA
	Sleeveless Pleated Top			550000	#6842a4d0797ed2da23215df2	XOA
	Pleated Shoulder Top			1000000	#6842a578797ed2da23215df7	XOA

- Giao diện quản lý đơn hàng

Dashboard

Products

Create New

View List

Categories

Create New

View List



Orders

View Orders

Return

Logout

Danh sách đơn hàng

Hình ảnh	Mã đơn hàng	Khách hàng	Sản phẩm	Tổng tiền	Trạng thái	Thanh toán	Thao tác
	68428775...	Bùi Thị Thu Bùi Thị Thu	SL: 1	200,000 VND	Đang xử lý	Đã thanh toán	<div>CẬP NHẬT</div> <div>XÓA</div>
	68428966...	Bùi Thị Thu đ Bùi Thị Thu	SL: 1	200,000 VND	Đang xử lý	Đã thanh toán	<div>CẬP NHẬT</div> <div>XÓA</div>
		Bùi Thị					<div>CẬP</div>

- Giao diện quản lý yêu cầu đổi/ trả hàng

Dashboard

Products

Create New

View List

Categories

Create New

View List

Orders

View Orders

Return

View Returns

Quản lý yêu cầu đổi/trả

Lọc theo trạng thái
Tất cả

Tìm kiếm

ID	Sản phẩm	Người yêu cầu	Loại yêu cầu	Lý do	Ngày tạo	Trạng thái	Thao tác
684565dd...	Số lượng: Giá: đ	Khách hàng	Trả hàng	Sai sản phẩmLý do khác	08/06/2025 17:28	Chờ xử lý	<div>CẬP NHẬT</div>
6843a005...	Đã xóa Số lượng:	Khách hàng	Đổi hàng	Sản phẩm lỗiLý do khác	07/06/2025 09:12	Đã chấp nhận	<div>CẬP NHẬT</div>

CHƯƠNG 5: TỔNG KẾT

1. Kết luận

- Hệ thống website bán hàng thời trang cơ bản nhằm cung cấp, đáp ứng nhu cầu mua sắm trực tuyến của người tiêu dùng. Thông qua việc xác định các yêu cầu và phân tích, hệ thống đã đáp ứng được cơ bản các chức năng cốt lõi đồng thời là một nền tảng cho quá trình phát triển mở rộng trong tương lai.
- Đồng thời hệ thống cũng có giao diện đơn giản, dễ dàng cho người dùng.
- Tuy nhiên hệ thống cũng vẫn còn những hạn chế nhất định: như đầy đủ được các tính năng của một trang thương mại điện tử chuyên nghiệp, các hình thức thanh toán còn hạn chế,...

2. Hướng phát triển trong tương lai

- Hệ thống có thể cải tiến, mở rộng thêm các chức năng:
 - + Tích hợp thêm nhiều hệ thống thanh toán: Momo, ZaloPay, Solana,....
 - + Tích hợp AI: tích hợp hệ thống chatbot hỗ trợ khách hàng
 - + Hỗ trợ thêm việc đưa ra kế hoạch kinh doanh,...