**TRƯỜNG ĐẠI HỌC KIẾN TRÚC HÀ NỘI**

  
**KHOA CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**MÔN HỌC:** PHÁT TRIỂN HỆ THỐNG THƯƠNG MẠI ĐIỆN TỬ

**ĐỀ TÀI:** PHÁT TRIỂN WEBSITE THƯƠNG MẠI ĐIỆN TỬ VỀ MỸ PHẨM

Giảng viên hướng dẫn: Phạm Thị Thanh Mai

Sinh viên thực hiện: Nhóm 8

Tống Thị Vân – 2055010287

Nguyễn Thị Ngọc  – 2055010191

Nguyễn Song Toàn – 2055010235

Lớp: 20CN1

*Hà Nội, Ngày    tháng   năm 2024.*

**MỤC LỤC**

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI** 5](#_Toc168297397)

[**1.1. Lý do chọn đề tài** 5](#_Toc168297398)

[**1.2. Thực trạng yêu cầu** 5](#_Toc168297399)

[**1.3. Phạm vi đề tài** 5](#_Toc168297400)

[**1.4. Yêu cầu** 6](#_Toc168297401)

[**1.5. Các khái niệm cơ bản** 6](#_Toc168297402)

[**1.6. Ngôn ngữ và công cụ lập trình** 7](#_Toc168297403)

[1.6.1. HTML 7](#_Toc168297404)

[1.6.2. CSS 8](#_Toc168297405)

[1.6.3. JavaScript 9](#_Toc168297406)

[1.6.4. Node.js 11](#_Toc168297407)

[1.6.5. Visual Studio Code 15](#_Toc168297408)

[1.6.6. Cơ sở dữ liệu MongoDB 18](#_Toc168297409)

[**CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ** 21](#_Toc168297410)

[**2.1. Xác định hệ thống** 21](#_Toc168297411)

[2.1.1. Biểu đồ phân rã chức năng toàn hệ thống 21](#_Toc168297412)

[2.1.2. Danh sách các tác nhân 21](#_Toc168297413)

[2.1.3. Danh sách use case tổng quát 22](#_Toc168297414)

[**2.2. Biểu đồ thiết kế chức năng** 23](#_Toc168297415)

[2.2.1. Biểu đồ usecase tổng quan hệ thống 23](#_Toc168297416)

[2.2.2. Biểu đồ usecase phân rã chức năng của khách hàng 24](#_Toc168297417)

[**2.3. Kịch bản(Scenario)** 25](#_Toc168297418)

[2.3.1. Đăng ký tài khoản 25](#_Toc168297419)

[2.3.2. Đăng nhập tài khoản 27](#_Toc168297420)

[2.3.3. Thêm sản phẩm vào giỏ hàng 29](#_Toc168297421)

[2.3.4. Thanh toán 30](#_Toc168297422)

[**CHƯƠNG 3: PHÁT TRIỂN HỆ THỐNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ VỀ MỸ PHẨM** 33](#_Toc168297423)

[**3.1. Giao diện chung** 33](#_Toc168297424)

[3.1.1. Trang chủ 33](#_Toc168297425)

[3.1.2. Đăng nhập 33](#_Toc168297426)

[**3.2. Giao diện của người quản trị hệ thống** 34](#_Toc168297427)

[3.2.1. Quản lý danh mục 35](#_Toc168297428)

[3.2.2. Tạo mới sản phẩm 35](#_Toc168297429)

[3.2.3. Sửa, cập nhật và xóa sản phẩm 36](#_Toc168297430)

[**3.3. Giao diện sử dụng của người dùng là khách hàng** 37](#_Toc168297431)

[3.3.1. Đăng ký 37](#_Toc168297432)

[3.3.2. Tìm kiếm sản phẩm 38](#_Toc168297433)

[3.3.3. Xem chi tiết sản phẩm 39](#_Toc168297434)

[3.3.4. Thêm vào giỏ hàng 39](#_Toc168297435)

[3.3.3. Thanh toán 40](#_Toc168297436)

[**CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ** 41](#_Toc168297437)

[**4.1. Cài đặt** 41](#_Toc168297438)

[**4.2. Kiểm thử** 42](#_Toc168297439)

[**TỔNG KẾT** 44](#_Toc168297440)

[**TÀI LIỆU THAM KHẢO** 44](#_Toc168297441)

**MỤC LỤC HÌNH ẢNH**

[Hình 1. Biểu tưởng HTML 8](#_Toc168297442)

[Hình 2. Biểu tượng CSS 9](#_Toc168297443)

[Hình 3. JavaScript 11](#_Toc168297444)

[Hình 4. Nodejs 13](#_Toc168297445)

[Hình 5. RESTful API 15](#_Toc168297446)

[Hình 6. Visual Studio Code 18](#_Toc168297447)

[Hình 7. Cơ sở dữ liệu NoSQL MongoDB 20](#_Toc168297448)

[Hình 8. Biểu đồ phân rã chức năng 22](#_Toc168297449)

[Hình 9.Biểu đồ usecase tổng quan hệ thống 25](#_Toc168297450)

[Hình 10. Biểu đồ usecase người dùng là khách hàng 26](#_Toc168297451)

[Hình 11. Giao diện trang chủ 34](#_Toc168297452)

[Hình 12.Giao diện đăng nhập 35](#_Toc168297453)

[Hình 13. Giao diện quản trị 35](#_Toc168297454)

[Hình 14. Giao diện quản lý danh mục 36](#_Toc168297455)

[Hình 15. Giao diện tạo sản phẩm 37](#_Toc168297456)

[Hình 16. Giao diện sửa, cập nhập và xóa sản phẩm 38](#_Toc168297457)

[Hình 17. Giao diện đăng ký 39](#_Toc168297458)

[Hình 18. Giao diện tìm kiếm 39](#_Toc168297459)

[Hình 19. Giao diện xem sản phẩm 40](#_Toc168297460)

[Hình 20. GIao diện thêm vào giỏ hàng 41](#_Toc168297461)

[Hình 21. Giao diện thanh toán 41](#_Toc168297462)

[Hình 22. Kết nối 42](#_Toc168297463)

[Hình 23. CSDL 42](#_Toc168297464)

[Hình 24. Kết nối 43](#_Toc168297465)

[Hình 25. Kiểm thử giao diện 44](#_Toc168297466)

[Hình 26. Kiểm thử bảo mật 44](#_Toc168297467)

MỤC LỤC BẢNG BIỂU

[Bảng 1. Người dùng trong hệ thống 23](#_Toc168297472)

[Bảng 2. Các usecase tổng quan trong hệ thống 24](#_Toc168297473)

[Bảng 3. Kịch bản chức năng "Đăng ký" 28](#_Toc168297474)

[Bảng 4. Kịch bản chức năng "Đăng ký" 30](#_Toc168297475)

[Bảng 5. Kịch bản chức năng "Thêm vào giỏ hàng" 31](#_Toc168297476)

[Bảng 6. Kịch bản chức năng "Thanh toán" 33](#_Toc168297477)

**LỜI MỞ ĐẦU**

Thời đại công nghệ số 4.0, cùng với sự phát triển như vũ bão của Internet, xu hướng kinh doanh trực tuyến hay bán hàng online đã đem lại hiệu quả kinh tế cho rất nhiều ngành nghề kinh doanh tại Việt Nam. Kỷ nguyên số đang bùng nổ với tốc độ chóng mặt, mở ra những cơ hội to lớn cho thương mại điện tử (TMĐT). Những năm gần đây, “thương mại điện tử” (TMĐT) đã không còn là khái niệm xa lạ trong xã hội hay một lĩnh vực mới mẻ tại nước ta. Có thể coi năm 2020, đại dịch COVID-19 đã mang đến nhiều biến động đối với nền kinh tế và sự tăng trưởng bứt phá

Nắm bắt xu hướng này, nhiều doanh nghiệp đã và đang hướng đến việc phát triển website TMĐT riêng để tiếp cận thị trường rộng lớn, mở rộng tệp khách hàng tiềm năng và gia tăng doanh thu. Thương mại điện tử đã không chỉ là một xu hướng mà còn trở thành một phần không thể tách rời trong cuộc sống hàng ngày của chúng ta. Nó đòi hỏi sự đầu tư bài bản về nguồn lực, sự am hiểu thị trường và kỹ năng quản lý chuyên nghiệp. Do đó, việc nghiên cứu và xây dựng chiến lược phát triển website TMĐT một cách khoa học là vô cùng quan trọng. Một website thương mại điện tử nên đáp ứng tính dễ sử dụng, tính an toàn, lại mang tính thẩm mỹ cao.

Tuy nhiên, phát triển website TMĐT hiệu quả không phải là một nhiệm vụ đơn giản. Đề tài "Phát triển website thương mại điện tử về mỹ phẩm" được lựa chọn vì nó phản ánh một nhu cầu mua sắm hiện đại của con người.

Đề tài này không chỉ giải quyết vấn đề kỹ thuật mà còn đòi hỏi sự am hiểu sâu sắc về thị trường, về cách thức mà người tiêu dùng mong muốn điều gì từ một trang web mua sắm mỹ phẩm trực tuyến. Đề tài này có thể thông tin nhóm sưu tập thiếu sót**,** chúng em rất mong nhận được những ý kiến đóng góp của giảng viên hướng dẫn Phạm Thị Thanh Mai để bài tập của chúng em được hoàn thiện. Nhóm em xin cảm ơn ạ.

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

## **1.1. Lý do chọn đề tài**

Trong những năm gần đây, nhu cầu làm đẹp qua việc sử dụng mỹ phẩm đã trở nên phổ biến và không ngừng tăng trưởng. Với sự bùng nổ của công nghệ thông tin, trang web thương mại điện tử chuyên về mỹ phẩm cung cấp một thị trường trực cho cả người bán và người mua.

Việc chọn đề tài này nhóm mong muốn tạo ra một kênh mua sắm mỹ phẩm thuận tiện, nhanh chóng và đáng tin cậy cho người tiêu dùng, đồng thời mở ra cơ hội kinh doanh lớn cho các nhà cung cấp.

## **1.2. Thực trạng yêu cầu**

Thị trường mỹ phẩm trực tuyến hiện nay đang phải đối mặt với nhiều thách thức từ việc đảm bảo chất lượng sản phẩm đến việc cung cấp trải nghiệm mua hàng trọn vẹn cho khách hàng.

Người tiêu dùng ngày càng đòi hỏi một trang web thân thiện với người dùng, bảo mật thông tin cá nhân, và thông tin chi tiết về sản phẩm, bao gồm cả nguồn gốc và thành phần.

Ngoài ra, việc tích hợp các công cụ tư vấn, đánh giá sản phẩm từ cộng đồng, và các chương trình khuyến mãi linh hoạt cũng là các yêu cầu không thể bỏ qua.

## **1.3. Phạm vi đề tài**

Đề tài này chủ yếu tập trung vào thiết kế và triển khai một website thương mại điện tử chuyên sâu về mỹ phẩm, từ giai đoạn nghiên cứu thị trường, lựa chọn sản phẩm, xây dựng chức năng trang web, đến chiến lược marketing và chăm sóc khách hàng.

Trong hệ thống này, khách hàng là người dùng chủ yếu: người có nhu cầu mua sắm

Người truy cập wesite gồm các chức năng:

* Đăng ký tài khoản mua sắm và đăng nhập vào hệ thống của webisite.
* Tìm kiếm thông tin sản phẩm.
* Xem thông tin sản phẩm.
* Thêm sản phẩm vào giỏ hàng.
* Mua và thanh toán an toàn.

## **1.4. Yêu cầu**

* Hệ thống cần đáp ứng các tiêu chí cơ bản sau:
* Danh mục sản phẩm: Sắp xếp sản phẩm thành các danh mục để dễ dàng điều hướng. Phát triển các trang sản phẩm rõ ràng và đầy đủ thông tin cơ bản
* Tìm kiếm và lọc: Triển khai các chức năng tìm kiếm và lọc để giúp người dùng tìm kiếm dễ dàng
* Giỏ hàng và thanh toán: Cho phép khách hàng thêm, xóa và cập nhật các mặt hàng một cách dễ dàng. Cung cấp tính năng thanh toán cho khách và khuyến khích tạo tài khoản để nâng cao sự thuận tiện cho khách hàng.
* Hỗ trợ khách hàng: Cung cấp các kênh hỗ trợ khách hàng đáp ứng nhanh, bao gồm email, trò chuyện trực tiếp hoặc điện thoại, để giải quyết kịp thời các thắc mắc, mối quan tâm và phản hồi. Phát triển các chính sách và hướng dẫn hoàn trả rõ ràng để tương tác với khách hàng được suôn sẻ.

## **1.5. Các khái niệm cơ bản**

* Thương mại điện tử (E-commerce): là việc mua bán hàng hóa hoặc dịch vụ thông qua môi trường Internet. Đối với mỹ phẩm, thương mại điện tử giúp người mua có thể trải nghiệm mua sắm không giới hạn về thời gian và địa điểm.
* Cổng thanh toán điện tử (Payment Gateway): là dịch vụ truyền dữ liệu giữa website thương mại điện tử, ngân hàng và thẻ tín dụng để thực hiện thanh toán trực tuyến.
* SEO (Search Engine Optimization): quá trình tối ưu hoá website để tăng thứ hạng trên các công cụ tìm kiếm, giúp website thu hút nhiều lượt truy cập tự nhiên hơn từ người dùng.
* UI/UX (User Interface/User Experience): liên quan đến việc thiết kế giao diện và trải nghiệm người dùng sao cho dễ dàng, tiện lợi và thân thiện với người dùng.
* Responsive Design: thiết kế web linh hoạt theo kích thước của các thiết bị khác nhau như điện thoại di động, tablet và máy tính để bàn.

## **1.6. Ngôn ngữ và công cụ lập trình**

### 1.6.1. HTML

[HTML](https://topdev.vn/viec-lam-it/html-kt75) (Hypertext Markup Language): Ngôn ngữ đánh dấu [siêu văn bản](https://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n). HTML được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes…



Hình 1. Biểu tưởng HTML

* **Ưu điểm của HTML**

**HTML**được sử dụng để tạo bố cục, cấu trúc trang web. Nó có một số ưu điểm sau:

* Có nhiều tài nguyên hỗ trợ với cộng đồng người dùng vô cùng lớn
* Có thể hoạt động mượt mà trên hầu hết mọi trình duyệt hiện nay
* Học HTML khá đơn giản
* Các markup sử dụng trong HTML thường ngắn gọn, có độ đồng nhất cao
* Sử dụng mã nguồn mở, hoàn toàn miễn phí
* HTML là chuẩn web được vận hành bởi W3C
* Dễ dàng để tích hợp với các loại ngôn ngữ backend (ví dụ như: PHP, Node.js,…)

### 1.6.2. CSS

CSS (Cascading Style Sheets), là một [ngôn ngữ](https://glints.com/vn/blog/ban-can-hoc-bao-nhieu-ngon-ngu-lap-trinh-de-tro-thanh-developer/)thiết kế được sử dụng nhằm mục đích đơn giản hóa quá trình tạo nên một website. CSS được ra mắt vào năm 1996 bởi World Wide Web Consortium (W3C).



Hình 2. Biểu tượng CSS

* Cách CSS hoạt động đó chính là tìm kiếm dựa trên vùng chọn chẳng hạn như thẻ HTML, ID, class, v.v. Sau đó, nó sẽ áp dụng những thuộc tính buộc phải thay đổi lên các vùng đã chọn.
* CSS rất dễ học và dễ hiểu nhưng nó cung cấp khả năng kiểm soát mạnh mẽ việc trình bày tài liệu HTML. Thông thường nhất, CSS được kết hợp với các ngôn ngữ đánh dấu HTML hoặc XHTML.
* HTML và CSS có mối quan hệ gắn bó mật thiết với nhau. Nếu HTML là nền tảng của một trang web thì CSS là tất cả tính thẩm mỹ của toàn bộ trang web đó.
* Bố cục CSS thường chủ yếu dựa vào hình hộp và mỗi hộp đều chiếm những khoảng trống trên trang của bạn với các thuộc tính như:
* Padding: Gồm không gian xung quanh nội dung (ví dụ: xung quanh đoạn văn bản).
* Border: Là đường liền nằm ngay bên ngoài phần đệm.
* Margin: Là khoảng cách xung quanh bên ngoài của phần tử.
* Đây là ba lợi ích chính của CSS:
* Giải quyết một vấn đề lớn

Trước khi có CSS, các thẻ như phông chữ, màu sắc, kiểu nền, các sắp xếp phần tử, đường viền và kích thước phải được lặp lại trên mọi trang web. Đây là một quá trình rất dài tốn thời gian và công sức. CSS đã được tạo ra để giải quyết vấn đề này. Đó là một khuyến cáo của W3C.

* Tiết kiệm rất nhiều thời gian

Định nghĩa kiểu CSS được lưu trong các tệp CSS bên ngoài vì vậy có thể thay đổi toàn bộ trang web bằng cách thay đổi chỉ một tệp.

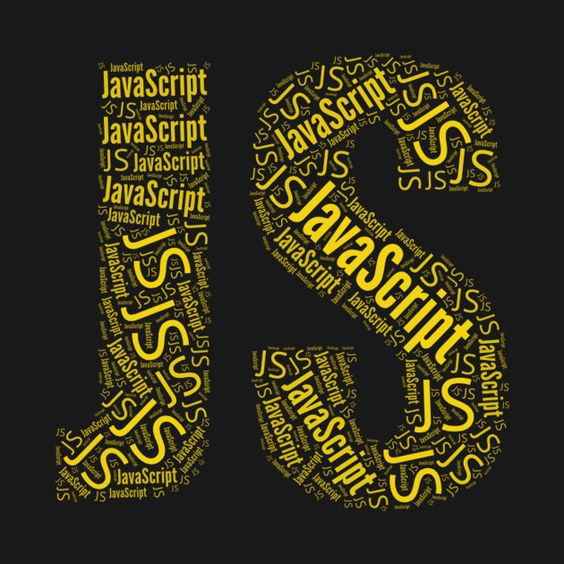
* Cung cấp thêm các thuộc tính

CSS cung cấp các thuộc tính chi tiết hơn HTML để định nghĩa giao diện của trang web.

### 1.6.3. JavaScript

JavaScript là ngôn ngữ lập trình được nhà phát triển sử dụng để tạo trang web tương tác. Là ngôn ngữ kịch bản phía máy khách, JavaScript là một trong những công nghệ cốt lõi của World Wide Web.

Thông thường, các trang web sẽ được nhúng trực tiếp JavaScript vào, hoặc sẽ sử dụng file .js để tham chiếu qua. Đây là ngôn ngữ phía máy khách, nghĩa là thay vì xử lý tập lệnh trên server của trang web, nó sẽ được tải về máy của khách truy cập và xử lý trên chính chiếc máy đó.



Hình 3. JavaScript

* Đối với lập trình viên:
* Đây là ngôn ngữ dễ học, dễ để phát hiện và sửa lỗi hơn.
* JavaScript cũng khá linh hoạt, và nó có thể được sử dụng ở nhiều nền tảng, trình duyệt, và không cần những công cụ quá phức tạp bởi chúng có thể được biên dịch bởi [HTML](https://www.thegioididong.com/hoi-dap/html-html5-la-gi-phan-biet-html-voi-html5-nen-su-dung-dang-1318215#hmenuid1) từ trình duyệt web.
* Đối với khách truy cập:
* Truy cập và tương tác với website hiệu quả hơn.
* Cho phép thực hiện các tác vụ trên trang web nhanh hơn.
* Tuy nhiên, công cụ nào cũng sẽ điểm mạnh và điểm yếu. Dưới đây là một số điểm yếu mà bạn nên cân nhắc qua.
* Dễ bị khai thác, hacker thực hiện tìm kiếm lỗi bảo mật để lợi dụng, từ đó sẽ chèn cắm các mã độc vào [máy tính](https://www.thegioididong.com/may-tinh-de-ban) của người sử dụng.
* Việc linh hoạt hỗ trợ cho các thiết bị cũng có thể tạo ra trải nghiệm không đồng nhất trên các thiết bị này, và đôi khi một số trình duyệt sẽ không hỗ trợ sử dụng JavaScript.

### 1.6.4. Node.js

Được phát hành vào năm 2009, NodeJS, hay còn được biết với tên gọi chính thức là Node.js, là môi trường thời gian chạy (runtime environment) JavaScript đa nền tảng và mã nguồn mở. NodeJS cho phép các lập trình viên tạo cả ứng dụng front-end và back-end bằng JavaScript.

Các đặc trưng nổi bật của Node.js

* NodeJS là mã nguồn mở

Mã nguồn của NodeJS được cung cấp công khai. Và được duy trì bởi những người đóng góp từ khắp nơi trên thế giới.

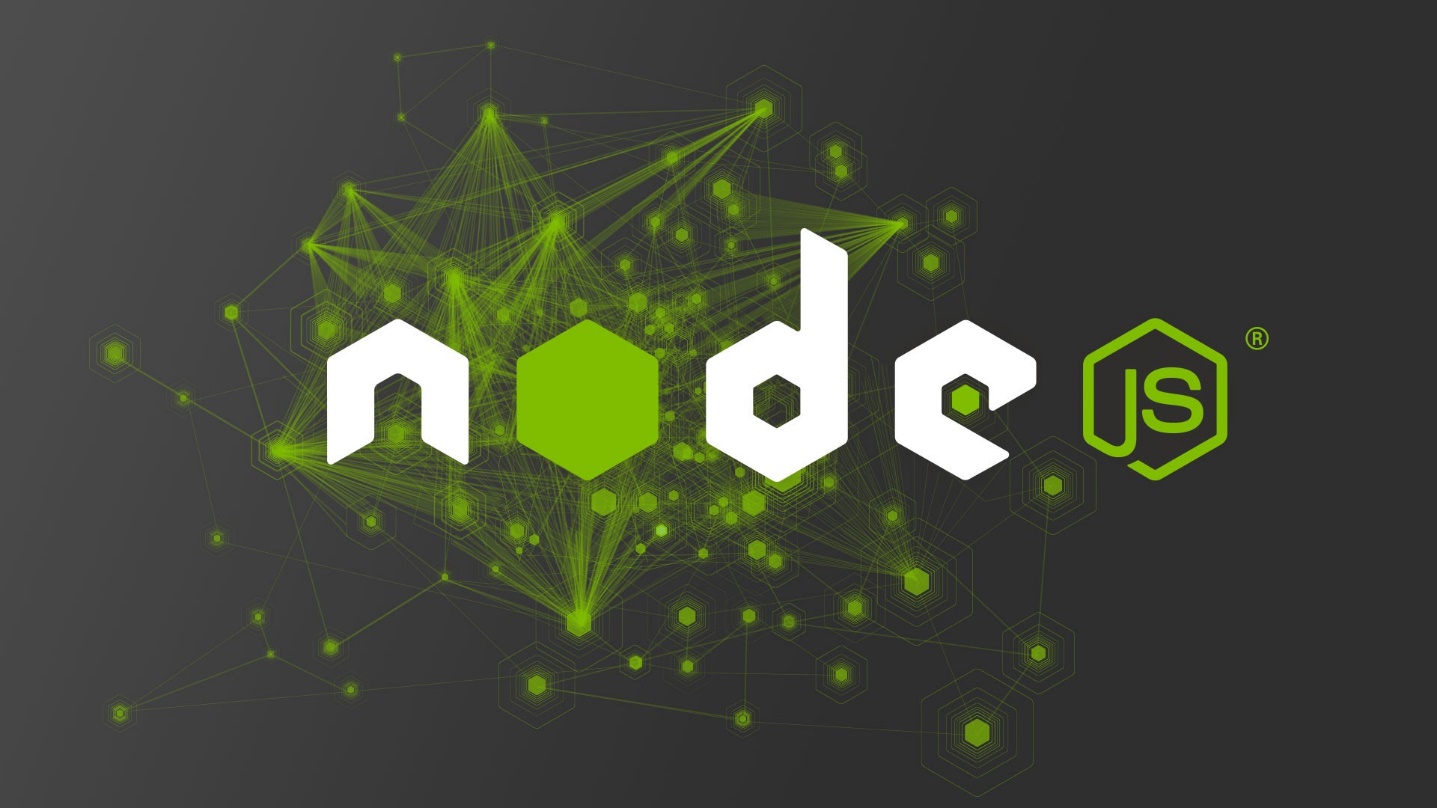
* NodeJS hỗ trợ đa nền tảng

NodeJS không phụ thuộc vào bất kỳ phần mềm hệ điều hành nào mà đều có thể hoạt động trên Linux, macOS hoặc Windows.

* NodeJS là môi trường thời gian chạy mã JavaScript

Các trình duyệt như Chrome và Firefox có môi trường thời gian chạy. Đó là lý do tại sao họ có thể chạy code JavaScript. Trước khi NodeJS được tạo, JavaScript chỉ có thể chạy trên trình duyệt và chỉ được sử dụng để xây dựng các ứng dụng front-end.

NodeJS cung cấp môi trường thời gian chạy bên ngoài trình duyệt. Nó cũng được xây dựng trên công cụ JavaScript của Chrome (V8 Engine). Điều này giúp bạn có thể xây dựng các ứng dụng back-end bằng cách sử dụng cùng ngôn ngữ lập trình JavaScript mà bạn quen thuộc.



Hình 4. Nodejs

**Ưu điểm của NodeJS**

* NodeJS có thể mở rộng

Các ứng dụng NodeJS có khả năng mở rộng cao vì chúng hoạt động không đồng bộ vì các yêu cầu đồng thời có thể được xử lý rất hiệu quả bằng NodeJS.

NodeJS hoạt động trên một luồng đơn nên khi có một yêu cầu đến, NodeJS sẽ bắt đầu xử lý yêu cầu đó và sẵn sàng xử lý yêu cầu tiếp theo.

* Thời gian thực thi code nhanh

Công cụ thời gian chạy (runtime motor) JavaScript V8 được NodeJS sử dụng và cũng được Google Chrome sử dụng. Một trình bao bọc được trung tâm cung cấp cho JavaScript và vì lý do đó, công cụ thời gian chạy trở nên nhanh hơn.

Đồng thời, việc sử dụng các khái niệm như lập trình không đồng bộ và cách vận hành non-blocking trên các hoạt động I/O cũng giúp nâng cao hiệu suất của NodeJS.

* Khả năng tương thích trên nhiều nền tảng

Các loại hệ điều hành khác nhau như Windows, UNIX, [LINUX](https://itviec.com/blog/linux/), MacOS và các thiết bị di động khác đều có thể sử dụng NodeJS.

* Sử dụng JavaScript

NodeJS sử dụng JavaScript. Hầu hết các lập trình viên đều quen thuộc với JavaScript, vì vậy đối với họ, việc hiểu NodeJS trở nên rất dễ dàng hơn.

* Truyền dữ liệu nhanh

Thời gian xử lý những dữ liệu đã được truyền đến các luồng khác nhau thường sẽ mất nhiều thời gian. Trong khi đó, để xử lý dữ liệu, NodeJS chỉ mất một khoảng thời gian rất ngắn và thực hiện với tốc độ nhanh.

NodeJS tiết kiệm rất nhiều thời gian vì các tệp được NodeJS xử lý và tải lên đồng thời. Do đó, tốc độ tổng thể của truyền dữ liệu và video được cải thiện nhờ NodeJS.

* Không có bộ đệm

Dữ liệu không bao giờ được lưu vào bộ đệm trong ứng dụng NodeJS.

* Tiết kiệm thời gian, công sức và chi phí

NodeJS được trang bị một kho thư viện khổng lồ – NPM (Node Package Manager). Các developer có thể sử dụng lại các module trong code và đưa các chức năng đa dạng vào bất kỳ ứng dụng nào.

* Một ngôn ngữ

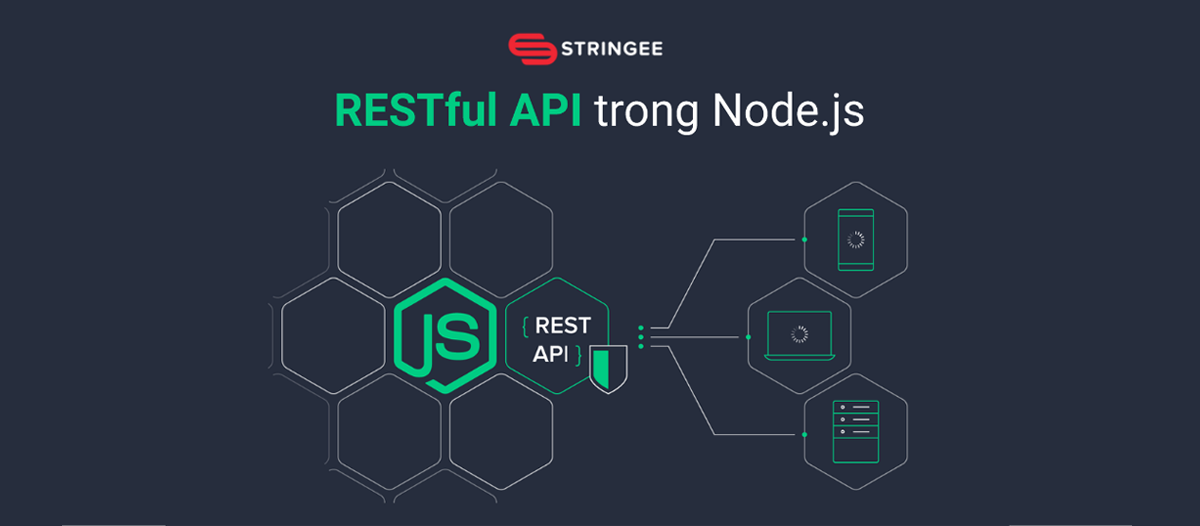
Một trong những lợi ích chính của NodeJS là khả năng viết toàn bộ cơ sở hạ tầng của bất kỳ ứng dụng web nào chỉ bằng một ngôn ngữ – JavaScript. Do đó, lập trình viên không cần tốn nhiều thời gian học các ngôn ngữ khác nhau để đáp ứng.

* Hỗ trợ cộng đồng mạnh mẽ

NodeJS có sự hỗ trợ cộng đồng và hỗ trợ mạnh mẽ vì là mã nguồn mở. Vì vậy, các nhà phát triển có thể tìm kiếm sự trợ giúp từ các chuyên gia từ khắp nơi trên thế giới. Điều này thúc đẩy các dự án phát triển.

**RESTful API trong NodeJS**

REST (Representational State Transfer). Về cơ bản REST là một kiến trúc dựa trên giao thức HTTP để có thể giúp thiết kế các ứng dụng web một cách tối ưu và hiệu quả. Mỗi thành phần của REST được coi là một tài nguyên và chúng sử dụng các phương thức HTTP tiêu chuẩn để có thể truy cập vào các tài nguyên trên. Kiến trúc này lần đầu được giới thiệu bởi Roy Fielding vào năm 2000.



Hình 5. RESTful API

Dựa trên nguyên tắc REST nói trên chúng ta có sự ra đời của RESTful API. Với bản chất là một API (Application Programming Interface), nó cung cấp các cách để truy cập và quản lý tài nguyên thông qua các phương thức HTTP. Mỗi URI xác định một tài nguyên và nó được biểu diễn bởi nhiều dạng như văn bản, JSON, XML,... hiện nay dữ liệu JSON vẫn đã và đang là kiểu dữ liệu được ưu thích và sử dụng phổ biến nhất.

Các phương thức HTTP được sử dụng trong REST:

* GET: Sử dụng để đọc dữ liệu từ tài nguyên. Ví dụ: Lấy thông tin người dùng.
* PUT: Sử dụng để tạo mới tài nguyên. Ví dụ: Tạo một bản ghi người dùng mới.
* DELETE: Sử dụng để xóa tài nguyên. Ví dụ: Xóa một người dùng.
* POST: Sử dụng để cập nhật tài nguyên hiện có và tạo mới tài nguyên. Ví dụ: Cập nhật thông tin người dùng hoặc tạo một người dùng mới.

**RESTful Web Service trong NodeJS**

Một web service là một tập hợp các giao thức và chuẩn được sử dụng để trao đổi dữ liệu giữa các ứng dụng và hệ thống khác nhau. Với web service chúng ta có thể giúp các ứng dụng chạy trên nhiều ngôn ngữ hay nền tảng khác nhau có thể giao tiếp với nhau thông qua trao đổi dữ liệu qua mạng máy tính.

RESTful Web Service cũng như vậy, nó là một loại web service và được xây dựng trên kiến trúc REST. Thông qua phương thức HTTP các web service này triển khai kiến trúc REST. Với, một RESTful web service thường được định nghĩa bằng một URI (Uniform Resource Identifier), giống như một đường dẫn, để đại diện cho một dịch vụ.

RESTful API và NodeJS đã trở thành một công cụ mạnh mẽ cho việc phát triển các ứng dụng web hiệu quả và tiết kiệm thời gian.

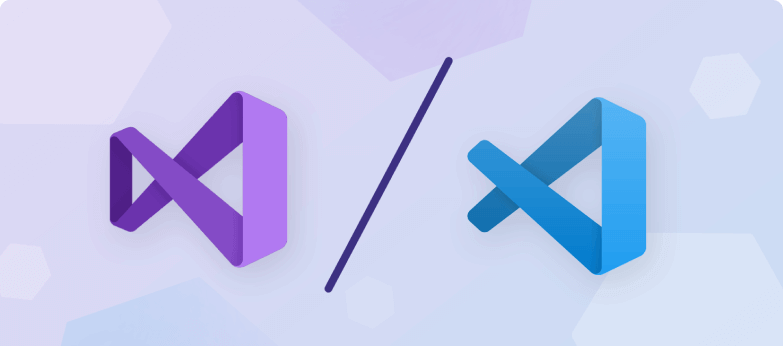
### 1.6.5. Visual Studio Code

**Visual Studio Code (VS Code) là một trình soạn thảo code miễn phí cho phép lập trình viên viết Code, gỡ lỗi và sửa Code bằng phương pháp Intellisense.** Thông thường, trình soạn thảo này tạo điều kiện cho người dùng có thể dễ dàng viết các loại Code một cách dễ dàng. Bất kỳ chương trình hay phần mềm nào được sử dụng đều hoạt động dựa trên các Code chạy ở chế độ nền.

* **Công dụng của Visual Studio Code**

Trình soạn thảo code Visual Studio Code có những tính năng ưu việt, bao gồm:

* Hỗ trợ hầu hết các ngôn ngữ lập trình hỗ trợ phát hiện các lỗi và tham chiếu đa ngôn ngữ một cách dễ dàng.
* Intelli-Sense có thể phát hiện các đoạn code bị bỏ sót hay tự động thực hiện các cú pháp biến phổ thông và khai báo biến. Chẳng hạn như Intelli-Sense sẽ tự động khai báo biến của người dùng được sử dụng trong chương trình.
* Hỗ trợ đa nền tảng có thể hoạt động đồng thời trên cả ba hệ thống đó là Windows, [Linux](https://bkhost.vn/blog/he-dieu-hanh-linux/) và Mac. Tuy nhiên, Code phần mềm mã nguồn mở và phần mềm độc quyền là hai loại khác nhau.
* Tiện ích mở rộng và hỗ trợ dành cho lập trình viên muốn sử dụng ngôn ngữ lập trình không được hỗ trợ. Tuy nhiên, mặc dù tắt đi tiện ích này thì hiệu suất hoạt động của trình chỉnh sửa cũng không bị ảnh hưởng.
* Kho lưu trữ an toàn được kết nối với [Git](https://bkhost.vn/blog/git/) hoặc với kho lưu trữ khác có thể chứa nhiều Code khác nhau. Điều này đồng thời còn giúp kéo và lưu trữ nhiều phiên bản Code khác nhau.
* Hỗ trợ các ứng dụng web có thể xây dựng trong VSC dễ dàng.
* Cấu trúc phân cấp bao gồm các tệp Code được đặt trong tệp và thư mục. Loại tệp Code này được sử dụng cho một số dự án phức tạp hoặc có thể tùy chọn xóa bỏ.
* Cải thiện các đoạn Code được khai báo khác với cấu trúc mặc định.
* Hỗ trợ thiết bị đầu cuối hỗ trợ không chuyển đổi giữa hai màn hình khi người dùng bắt đầu thực thi từ thư mục gốc.
* Hỗ trợ đa dự án chứa nhiều tệp và thư mục được mở cùng một lúc hoặc không liên quan đến nhau.
* Hỗ trợ tiết kiệm các tài nguyên được lấy trực tiếp từ Github Repo trực tuyến và ngược lại. Ngoài ra, nó còn hỗ trợ sao chép Code có sẵn trên Internet để thay đổi và lưu trữ.
* Hỗ trợ nhận xét các Code giúp người dùng có thể nhớ và theo dõi trình tự mong muốn.



Hình 6. Visual Studio Code

* **Ưu điểm của Visual Studio Code**

Ngoài những tính năng ưu việt đã được giới thiệu ở trên, Visual Studio Code có một số ưu điểm nổi bật khác. Cụ thể như sau:

* Hỗ trợ đa nền tảng bao gồm Windows, Linux và Mac.
* Khối lượng nhẹ.
* Cấu trúc mạnh mẽ.
* Intelli-Sense.
* Phần mềm miễn phí.
* Dành cho các ứng dụng máy tính.
* Hỗ trợ các công cụ công nghệ web như [HTML](https://bkhost.vn/blog/html/), [CSS](https://bkhost.vn/blog/css/), [JSON](https://bkhost.vn/blog/json/).

Các ưu điểm này khác với những tính năng của nó đó là Visual Studio Code chủ yếu hỗ trợ các nhà phát triển Front-End so với các nhà phát triển Back-End. Tuy nhiên, đối với một số người dùng thì họ cho rằng chúng đều hữu ích như nhau. Visual Studio Code hỗ trợ hầu hết các ngôn ngữ mà lập trình viên sử dụng. Tuy nhiên, đối với các ngôn ngữ khác yêu cầu tải xuống hay các phần mở rộng cần thiết.

* **Phạm vi Visual Studio**

Các ngôn ngữ phổ biến nhất sử dụng Visual Studio Code như [C#](https://bkhost.vn/blog/c-sharp/), [Visual Basic](https://bkhost.vn/blog/vb-visual-basic/), [Javascript](https://bkhost.vn/blog/javascript-js/), R, [XML](https://bkhost.vn/blog/xml-la-gi/), [Python](https://bkhost.vn/blog/python/), CSS, GO, PERL. Một tính năng được xem là thân thiện với người dùng khi nó có thể dễ dàng được nhận thấy trên các trình chỉnh sửa của Visual Studio Code.

### 1.6.6. Cơ sở dữ liệu MongoDB

MongoDB là một phần mềm mã nguồn mở dùng để quản trị cơ sở dữ liệu NoSQL. NoSQL (Not only SQL) được sử dụng thay thế cho cơ sở dữ liệu quan hệ (Relational Database – RDB) truyền thống. Cơ sở dữ liệu NoSQL khá hữu ích trong khi làm việc với các tập dữ liệu phân tán lớn. MongoDB là một công cụ có thể quản lý thông tin hướng document cũng như lưu trữ hoặc truy xuất thông tin.

Hiện nay, có nhiều công ty toàn cầu sử dụng MongoDB để lưu trữ lượng dữ liệu “khổng lồ” của họ như Facebook, Nokia, eBay, Adobe, Google,…

**Công dụng của MongoDB**

MongoDB giúp các tổ chức lưu trữ lượng lớn dữ liệu trong khi vẫn hoạt động nhanh chóng. Ngoài lưu trữ dữ liệu, MongoDB còn được sử dụng trong các trường hợp sau:

* Tích hợp một lượng lớn dữ liệu đa dạng
* Mô tả các cấu trúc dữ liệu phức tạp, biến hoá
* Cung cấp dữ liệu cho các ứng dụng hiệu suất cao
* Hỗ trợ các ứng dụng [đám mây lai](https://itviec.com/blog/dien-toan-dam-may/) và đa đám mây
* Hỗ trợ phương pháp phát triển [Agile](https://itviec.com/blog/agile-la-gi-scrum-la-gi/)

Thay vì sử dụng các table và row như trong cơ sở dữ liệu quan hệ, vì là cơ sở dữ liệu NoSQL, MongoDB được tạo thành từ collection và document. Document được tạo thành từ các cặp khóa-giá trị (là đơn vị dữ liệu cơ bản của MongoDB). Còn collection, tương đương với table trong SQL, là nơi chứa các bộ document.



Hình 7. Cơ sở dữ liệu NoSQL MongoDB

**Ưu điểm MongoDB**

MongoDB mang đến cho người sử dụng một số ưu điểm:

* Không schema: Giống như các cơ sở dữ liệu NoSQL khác, MongoDB không yêu cầu các schema được xác định trước.
* MongoDB lưu trữ bất kỳ loại dữ liệu nào: Điều này cho phép người dùng linh hoạt tạo số lượng trường trong document theo nhu cầu, và giúp việc mở rộng cơ sở dữ liệu MongoDB trở nên dễ dàng hơn so với cơ sở dữ liệu quan hệ truyền thống.
* Hướng document: Một trong những ưu điểm của việc sử dụng document là các đối tượng này ánh xạ tới các kiểu dữ liệu gốc trong một số ngôn ngữ lập trình. Việc có các document được nhúng cũng làm giảm nhu cầu kết nối cơ sở dữ liệu, điều này có thể làm giảm chi phí.
* Khả năng mở rộng: Kiến trúc mở rộng theo chiều ngang của MongoDB giúp bạn tạo ra một ứng dụng có thể xử lý được lưu lượng truy cập tăng đột biến khi doanh nghiệp của bạn phát triển. Ngoài ra, việc phân chia dữ liệu (sharding) cho phép cơ sở dữ liệu phân phối dữ liệu trên một cụm máy. MongoDB cũng hỗ trợ tạo vùng dữ liệu dựa trên shard key.
* Hỗ trợ bên thứ ba: MongoDB hỗ trợ một số công cụ lưu trữ và cung cấp API công cụ lưu trữ có thể cắm được (pluggable storage engine API) cho phép các bên thứ ba phát triển công cụ lưu trữ dữ liệu riêng.
* Linh hoạt lưu trữ tệp dung lượng lớn: MongoDB phát triển hệ thống tệp riêng GridFS, gần giống với hệ thống tệp phân tán Hadoop. Việc sử dụng hệ thống tệp nhằm để lưu trữ các tệp vượt qua kích thước giới hạn của BSON (16 MB cho mỗi document).

**Khuyết điểm MongoDB**

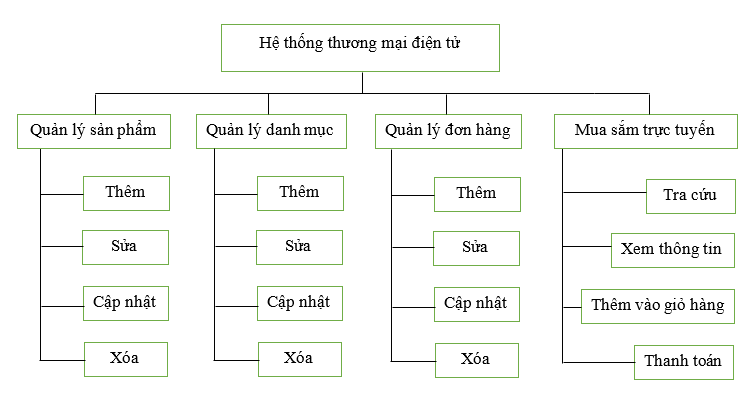
Mặc dù MongoDB mang lại nhiều giá trị lớn, công cụ này vẫn có một số nhược điểm:

* Tính liên tục: Với chiến lược chuyển đổi dự phòng tự động, người dùng chỉ có thể thiết lập một node master trong cụm MongoDB. Nếu node master bị lỗi, một node khác sẽ tự động chuyển đổi thành master mới. Quá trình chuyển đổi này đảm bảo tính liên tục, nhưng không diễn ra tức thời mà có thể mất tới một phút.
* Giới hạn ghi: Node master duy nhất của MongoDB cũng làm giới hạn lại tốc độ ghi dữ liệu vào cơ sở dữ liệu. Việc ghi dữ liệu phải được ghi trên node master và việc ghi thông tin mới vào cơ sở dữ liệu bị giới hạn bởi khả năng của node master đó.
* Tính nhất quán của dữ liệu: MongoDB không cung cấp tính toàn vẹn tham chiếu đầy đủ thông qua việc sử dụng các ràng buộc khóa ngoại (foreign-key), điều này có thể ảnh hưởng đến tính nhất quán của dữ liệu.
* Bảo mật: Tính năng xác thực người dùng không được mặc định bật trong cơ sở dữ liệu MongoDB. Để bảo mật hệ thống trước các cuộc tấn công của tin tặc, bạn có thể thủ công thiết lập các cài đặt chặn những kết nối lạ và không an toàn.

# **CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ**

## **2.1. Xác định hệ thống**

### 2.1.1. Biểu đồ phân rã chức năng toàn hệ thống



Hình 8. Biểu đồ phân rã chức năng

Hệ thống gồm 3 chức năng cho người quản lý:

* Quản lý sản phẩm
* Quản lý danh mục
* Quản lý đơn hàng

Người dùng là khách hàng vào hệ thống để tham khảo và mua sắm trực tuyến

### 2.1.2. Danh sách các tác nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Tác nhân** | **Ý nghĩa** |
| 1 | Quản lý hệ thống (Admin) | Người quản lý chính trong hệ thống, có quyền cao nhất |
| 2 | Người dùng (Khách hàng) | Người sử dụng chính hệ thống website có nhu cầu tham khảo thông tin và mua sắm. |

Bảng 1. Người dùng trong hệ thống

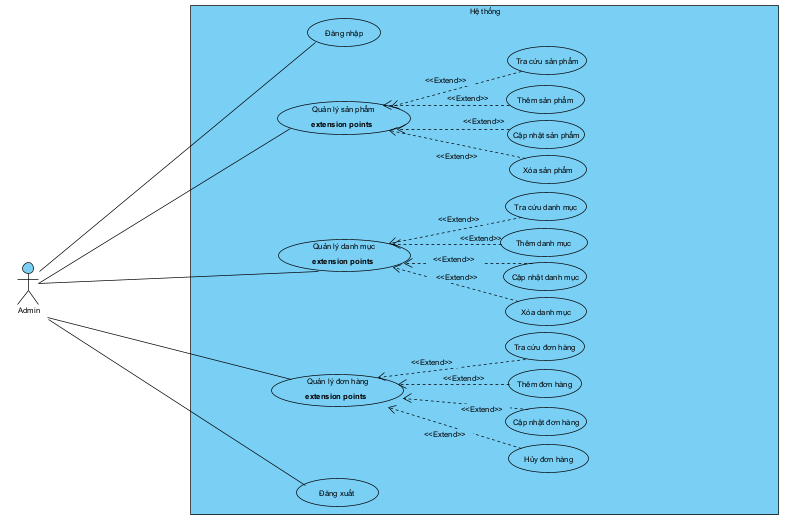
### 2.1.3. Danh sách use case tổng quát

|  |  |  |
| --- | --- | --- |
| **Actor** | **Use Case** | **Mô tả** |
| Người dùng (Khách hàng) | Đăng ký | Người dùng đăng ký tài khoản và mật khẩu để đăng nhập hệ thống |
|  | Đăng nhập | Sau khi đăng ký thành công, người dùng đăng nhập thành công vào hệ thống để thực hiện các hành động khác |
|  | Tra cứu nội dung | Người dùng được cấp quyền tra cứu nội dung thông qua từ khóa |
|  | Xem nội dung | Người dùng xem được nội dung kết quả sau chức năng tìm kiếm |
|  | Thêm sản phẩm vào giỏ hàng | Người dùng chọn sản phẩm thêm vào giỏ hàng |
|  | Xóa sản phẩm trong giỏ hàng | Người dùng chọn sản phẩm xóa khỏi giỏ hàng |
|  | Thanh toán | Người dùng lựa chọn phương thức thanh toán phù hợp |
|  | Đăng xuất | Người dùng kết thúc các thao tác với hệ thống |
| Quản lý hệ thống (Admin) | Đăng nhập | Người dùng đăng nhập vào hệ thống với tài khoản được cấp để thực hiện các hành động khác |
|  | Đăng xuất | Người dùng kết thúc các thao tác với hệ thống |
|  | Quản lý sản phẩm | Người dùng thực hiện thao tác tra cứu, thêm, cập nhật xóa sản phẩm trong hệ thống |
|  | Quản lý danh mục | Người dùng thực hiện thao tác tra cứu, thêm, cập nhật xóa danh mục trong hệ thống |
|  | Quản lý đơn hàng | Người dùng thực hiện thao tác tra cứu, thêm, cập nhật xóa đơn hàng của khách hàng |

Bảng 2. Các usecase tổng quan trong hệ thống

## **2.2. Biểu đồ thiết kế chức năng**

### 2.2.1. Biểu đồ usecase tổng quan hệ thống

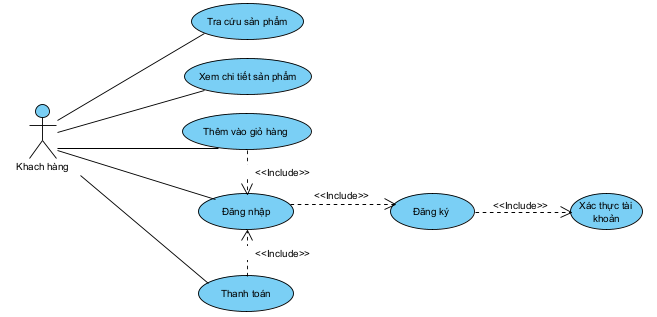


Hình 9.Biểu đồ usecase tổng quan hệ thống

Usecase này mô tả người quản lý toàn bộ hệ thống. Người này đăng nhập vào hệ thống thực hiện các quyền hạn sau:

* Đăng nhập/đăng xuất
* Quản lý sản phẩm
* Quản lý danh mục
* Quản lý đơn hàng

### 2.2.2. Biểu đồ usecase phân rã chức năng của khách hàng



Hình 10. Biểu đồ usecase người dùng là khách hàng

Usecase này cho phép sau khi đăng ký thành công tài khoản, người dùng có thể đăng nhập vào hệ thống thực hiện các chức năng sau:

* Tra cứu sản phẩm: tra cứu bằng thông tin tên sản phẩm hoặc nhãn hàng
* Xem chi tiết thông tin sản phẩm: xem các thông tin về sản phẩm, công dụng, bên cung cấp,...
* Thêm sản phẩm vào giỏ hàng: thêm số lượng và mặt hàng mong muốn
* Thanh toán: sau khi tham khảo thông tin, khách hàng có thể thanh toán đơn hàng lựa chọn phương thức thanh toán

## **2.3. Kịch bản(Scenario)**

### 2.3.1. Đăng ký tài khoản

|  |  |  |
| --- | --- | --- |
| 1 | Use Case ID | UC-1.1 |
| 2 | Use Case Name | Đăng ký tài khoản |
| 3 | Description | Quá trình đăng ký tài khoản của khách hàng A |
| 4 | Actor(s) | Khách hàng |
| 5 | Priority | Must Have |
| 6 | Trigger | Khách hàng đăng ký tài khoản |
| 7 | Pre-Condition(s) | + Khách hàng là người có nhu cầu mua hàng  + Được phép truy cập vào hệ thống nhưng chỉ có chức năng xem sản phẩm |
| 8 | Post-Condition(s) | Màn hình hiển thị giao diện đăng nhập/đăng ký |
| 9 | Basic Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Hệ thống hiển thị giao diện trang chủ hiển thị bộ lọc, tất cả danh mục và sản phẩm.  3. Khách hàng A chọn biểu tượng actor. Giao diện chuyển sang trang đăng ký/đăng nhập  4. Khách hàng chọn đăng ký và nhập thông tin cá nhân: tên tài khoản, mật khẩu,…nhấn chọn đăng ký.  5. Hệ thống thông báo đăng ký tài khoản mới thành công |
| 10 | Alternative Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Hệ thống hiển thị giao diện trang chủ hiển thị bộ lọc, tất cả danh mục và sản phẩm.  3. Khách hàng A xem thông tin sản phẩm và thêm giỏ hàng. Giao diện chuyển sang trang đăng ký/đăng nhập  4. Khách hàng chọn đăng ký và nhập thông tin cá nhân: tên tài khoản, mật khẩu,…nhấn chọn đăng ký.  5. Hệ thống thông báo đăng ký tài khoản mới thành công  6. Khách hàng đăng nhập vào hệ thống |
| 11 | Exception Flow | 5. Hệ thống thông báo tài khoản đã đăng ký hoặc chứa ký tự không hợp lệ. Yêu cầu khách hàng kiểm tra lại.  =>Lặp lại bước 4, tài khoản khách hàng vẫn không đăng ký thành công. Kết thúc. |

Bảng 3. Kịch bản chức năng "Đăng ký"

### 2.3.2. Đăng nhập tài khoản

|  |  |  |
| --- | --- | --- |
| 1 | Use Case ID | UC-1.3 |
| 2 | Use Case Name | Đăng nhập tài khoản |
| 3 | Description | Quá trình đăng nhập tài khoản của khách hàng A |
| 4 | Actor(s) | Khách hàng |
| 5 | Priority | Must Have |
| 6 | Trigger | Khách hàng đăng ký tài khoản |
| 7 | Pre-Condition(s) | + Khách hàng là người có nhu cầu mua hàng  + Được phép truy cập vào hệ thống |
| 8 | Post-Condition(s) | Màn hình hiển thị giao diện đăng nhập |
| 9 | Basic Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Hệ thống hiển thị giao diện trang chủ hiển thị bộ lọc, tất cả danh mục và sản phẩm.  3.Khách hàng A chọn biểu tượng actor  4. Khách hàng đăng nhập vào hệ thống với tên và tài khoản đã được đăng ký.  5. Hệ thống thông báo đăng nhập tài khoản thành công |
| 10 | Alternative Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Hệ thống hiển thị giao diện trang chủ gồm thông tin sản phẩm được phân loại: mặt hàng mới, thương hiệu, chăm sóc da, đồ trang điểm và nước hoa.  3. Khách hàng A xem thông tin sản phẩm và thêm giỏ hàng. Giao diện chuyển sang trang đăng ký/đăng nhập  4. Khách hàng chọn đăng nhập và nhập thông tin cá nhân: tên tài khoản, mật khẩu,…nhấn chọn đăng nhập.  5. Hệ thống thông báo đăng nhập tài khoản thành công |
| 11 | Exception Flow | 5. Hệ thống thông báo tài khoản đăng nhập không thành công do sai thông tin đăng nhập. Yêu cầu khách hàng kiểm tra lại.  =>Lặp lại bước 4, tài khoản khách hàng vẫn không đăng nhập thành công, hệ thống tạm khóa tài khoản trong vòng 1 ngày. Kết thúc. |

Bảng 4. Kịch bản chức năng "Đăng ký"

### 2.3.3. Thêm sản phẩm vào giỏ hàng

|  |  |  |
| --- | --- | --- |
| 1 | Use Case ID | UC-1.2 |
| 2 | Use Case Name | Thêm sản phẩm vào giỏ hàng |
| 3 | Description | Quá trình thêm sản phẩm vào giỏ hàng của khách hàng A |
| 4 | Actor(s) | Khách hàng |
| 5 | Priority | Must Have |
| 6 | Trigger | Khách hàng đăng ký tài khoản |
| 7 | Pre-Condition(s) | + Khách hàng là người có nhu cầu mua hàng  + Được phép truy cập vào hệ thống |
| 8 | Post-Condition(s) | Màn hình hiển thị giao diện đăng ký/đăng nhập  Màn hình hiển thị giao diện trang chủ  Màn hình hiển thị giao diện chi tiết sản phẩm |
| 9 | Basic Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Khách hàng A đăng nhập thành công vào hệ thống  3. Khách hàng A chọn sản phẩm, xem chi tiết sản phẩm, chọn số lượng và chọn thêm vào giỏ hàng  4. Hệ thống thông báo thêm thành công |
| 10 | Alternative Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Khách hàng A đăng nhập thành công vào hệ thống  3. Khách hàng A tìm kiếm sản phẩm, xem chi tiết sản phẩm, chọn số lượng và thêm vào giỏ hàng  4. Hệ thống thông báo thêm thành công |
| 11 | Exception Flow | 4. Hệ thống thông báo thêm không thành công vì đã hết hàng.  => Kết thúc |

Bảng 5. Kịch bản chức năng "Thêm vào giỏ hàng"

### 2.3.4. Thanh toán

|  |  |  |
| --- | --- | --- |
| 1 | Use Case ID | UC-1.2 |
| 2 | Use Case Name | Thanh toán đơn hàng |
| 3 | Description | Quá trình thanh toán đơn hàng của khách hàng A |
| 4 | Actor(s) | Khách hàng |
| 5 | Priority | Must Have |
| 6 | Trigger | Khách hàng đăng ký tài khoản |
| 7 | Pre-Condition(s) | + Khách hàng là người có nhu cầu mua hàng  + Được phép truy cập vào hệ thống |
| 8 | Post-Condition(s) | Màn hình hiển thị giao diện đăng ký/đăng nhập  Màn hình hiển thị giao diện trang chủ  Màn hình hiển thị giao diện chi tiết sản phẩm  Màn hình hiển thị giỏ hàng của khách hàng |
| 9 | Basic Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Khách hàng A đăng nhập thành công vào hệ thống  3. Khách hàng A chọn sản phẩm, xem chi tiết sản phẩm, chọn số lượng và chọn thêm vào giỏ hàng  4. Hệ thống thông báo thêm thành công  5. Khách hàng A vào giỏ hàng, chọn các sản phẩm cần mua. Màn hình hiện ra phương thức thanh toán. Khách hàng chọn thanh toán trực tuyến. Hệ thống gửi mã OTP xác nhận.  6. Khách hàng A nhập mã OTP để xác nhận. Hệ thống thông báo trừ tiền trong tài khoản và đặt đơn thành công. |
| 10 | Alternative Flow | 1. Khách hàng A truy cập vào website TMĐT Myskin.  2. Khách hàng A đăng nhập thành công vào hệ thống  3. Khách hàng A chọn sản phẩm, xem chi tiết sản phẩm, chọn số lượng và chọn thêm vào giỏ hàng  4. Hệ thống thông báo thêm thành công  5. Khách hàng A vào giỏ hàng, chọn các sản phẩm cần mua. Màn hình hiện ra phương thức thanh toán. Khách hàng chọn thanh toán khi nhận hàng. Hệ thống gửi mã OTP xác nhận.  6. Khách hàng A nhập mã OTP để xác nhận. Hệ thống thông báo đặt đơn thành công. |

Bảng 6. Kịch bản chức năng "Thanh toán"

# **CHƯƠNG 3: PHÁT TRIỂN HỆ THỐNG WEBSITE THƯƠNG MẠI ĐIỆN TỬ VỀ MỸ PHẨM**

## **3.1. Giao diện chung**

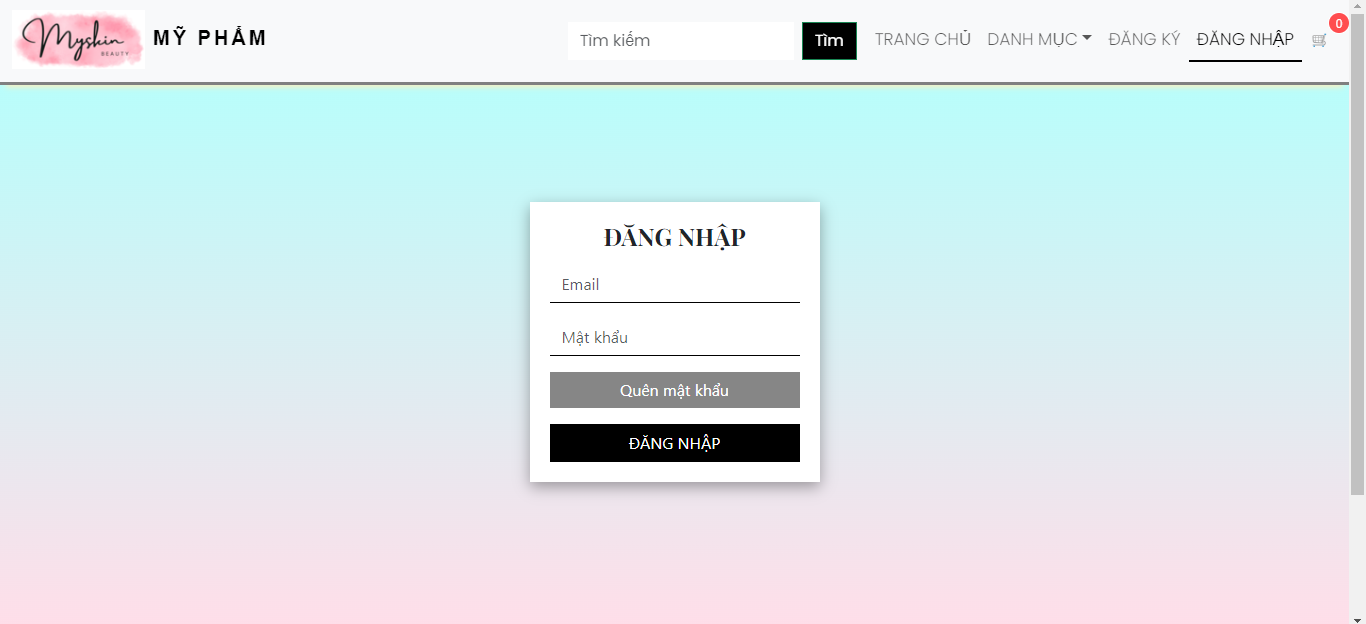
### 3.1.1. Trang chủ



Hình 11. Giao diện trang chủ

Khi người dùng vào hệ thống, trang web sẽ hiện ra giao diện trang chủ chỉ xem thông tin. Để thao tác các chức năng của trang web, người dùng cần đăng nhập vào hệ thống

### 3.1.2. Đăng nhập



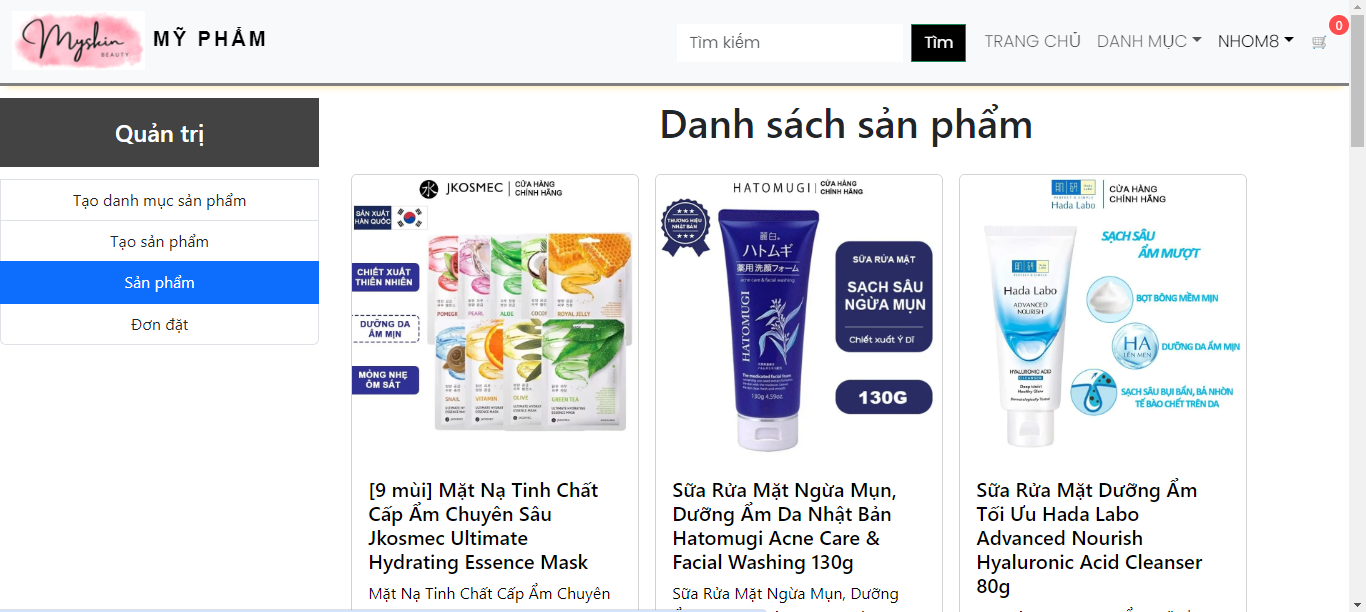
Hình 12.Giao diện đăng nhập

Giao diện này xuất hiện yêu cầu người dùng đăng nhập vào hệ thống để các chức năng với quyền hạn tương ứng.

## **3.2. Giao diện của người quản trị hệ thống**

Người dùng nhập tên tài khoản và mật khẩu sau để vào hệ thống với quyền quản trị:

* Tên tài khoản: [nhom8@gmail.com](mailto:nhom8@gmail.com)
* Mật khẩu: abcdef

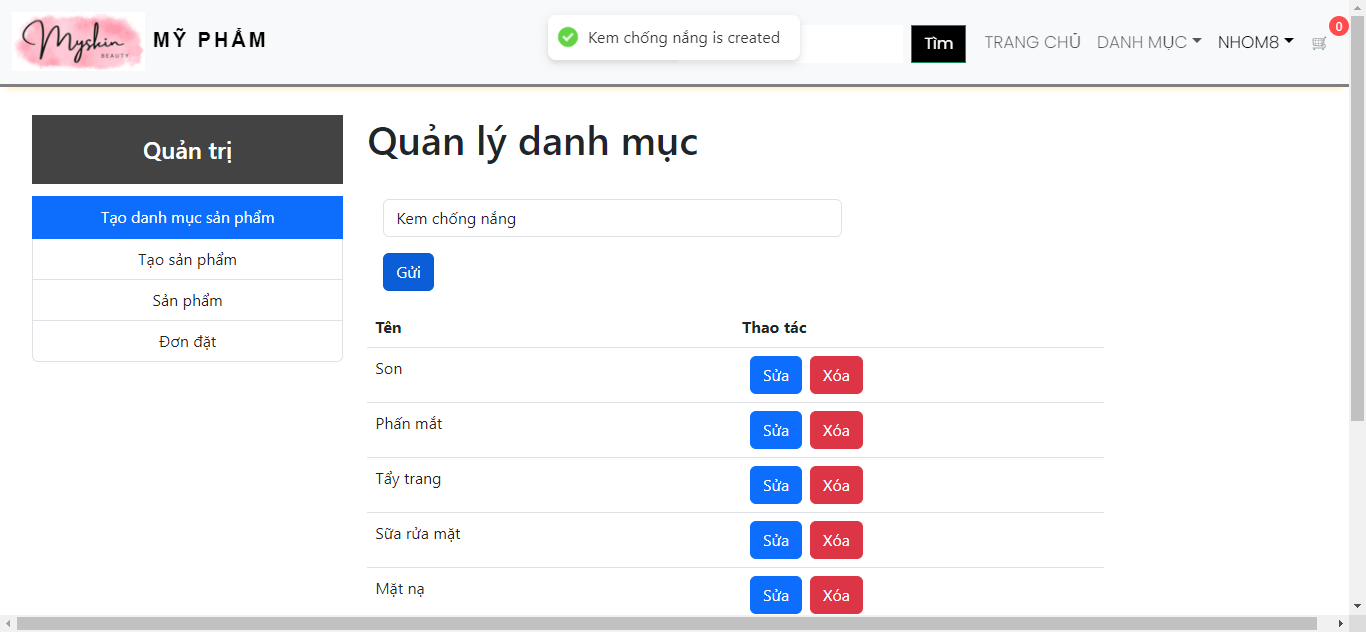


Hình 13. Giao diện quản trị

Đây là tài khoản người dùng được cài mặc định trong hệ thống thao tác thêm, sửa, cập nhật, xóa với danh mục, sản phẩm và đơn hàng của hệ thống thương mại điện tử.

### 3.2.1. Quản lý danh mục

Quản trị thêm mới danh mục: Kem chống nắng. Hệ thống thông báo thêm danh mục này thành công



Hình 14. Giao diện quản lý danh mục

Nếu người quản trị muốn sửa tên hoặc xóa danh mục có thể chọn nút sửa hoặc xóa. Hệ thống thông báo sửa/xóa thành công

### 3.2.2. Tạo mới sản phẩm

Người quản trị thêm mới sản phẩm, nhập lần lượt các thông tin yêu cầu:

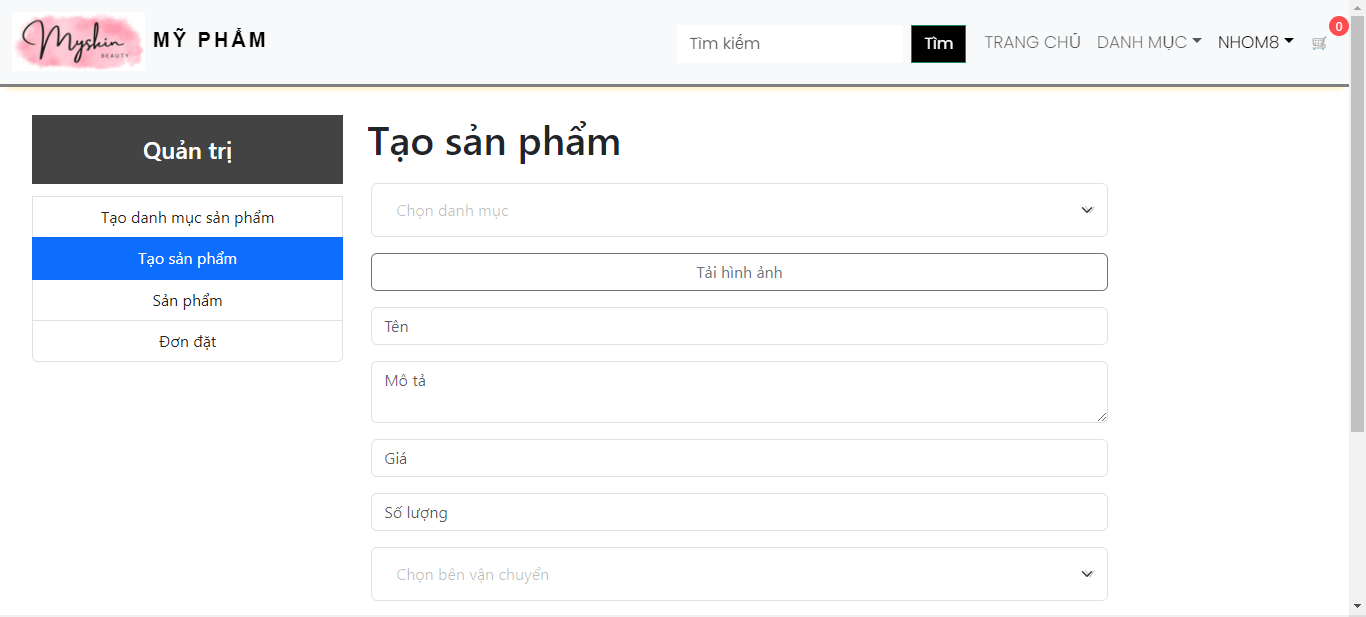
Chọn danh mục: phân loại sản phẩm dễ dàng tìm kiếm

Tải hình ảnh: minh họa hình ảnh thực tế của sản phẩm

Mô tả: các thông tin giới thiệu/ ưu điểm sản phẩm

Giá: số tiền, giá trị sản phẩm

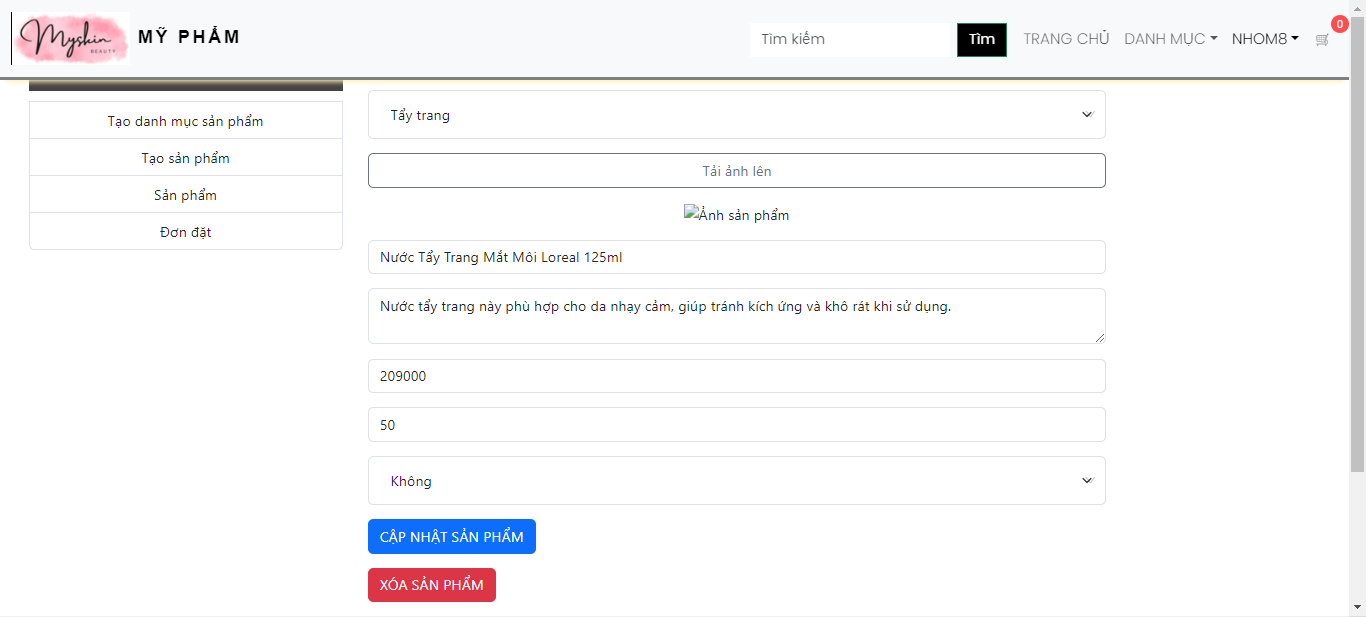
Chọn bên vận chuyển: có/không, hệ thống mặc định bên giao hàng hoặc người dùng lựa chọn.



Hình 15. Giao diện tạo sản phẩm

### 3.2.3. Sửa, cập nhật và xóa sản phẩm

Người quản trị nhấp chuột vào sản phẩm giao diện hiện ra thông tin tương ứng:

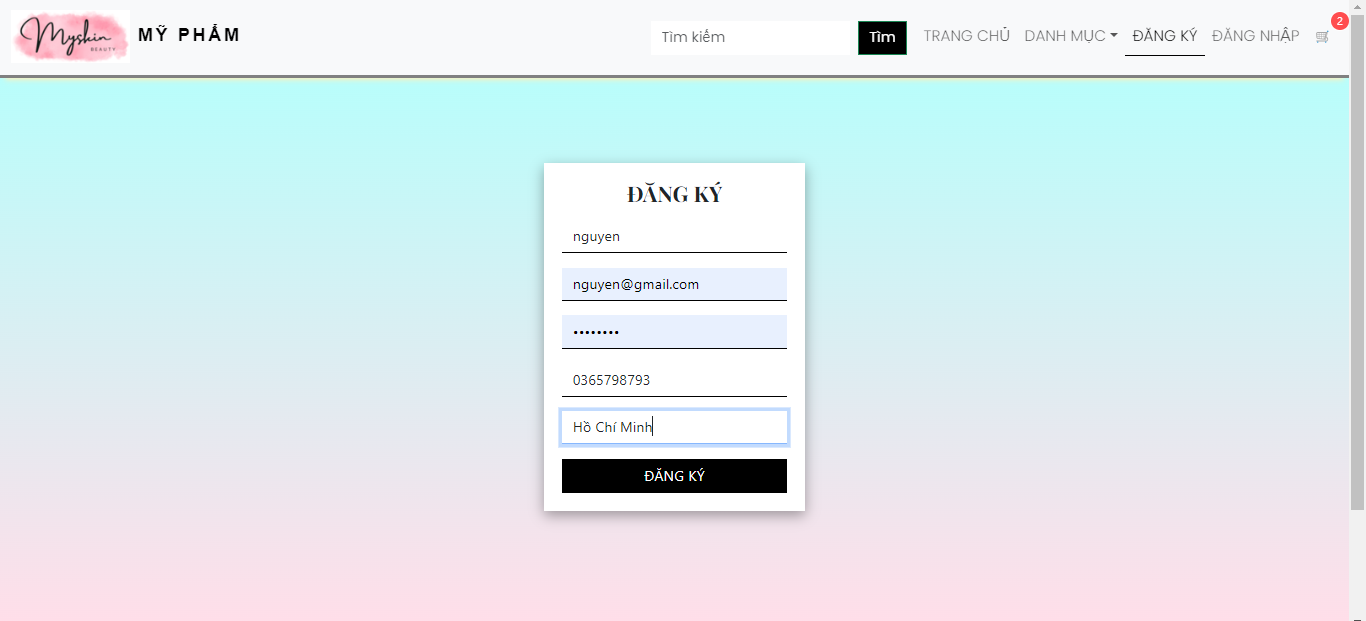


Hình 16. Giao diện sửa, cập nhập và xóa sản phẩm

Ở đây, người dùng có thể sửa và chọn vào nút cập nhật, hoặc chọn xóa sản phẩm khỏi hệ thống.

## **3.3. Giao diện sử dụng của người dùng là khách hàng**

### 3.3.1. Đăng ký



Hình 17. Giao diện đăng ký

Khi người dùng vào hệ thống, trang web sẽ hiện ra giao diện trang chủ chỉ xem thông tin. Để thao tác các chức năng của trang web, người dùng cần đăng ký hệ thống:

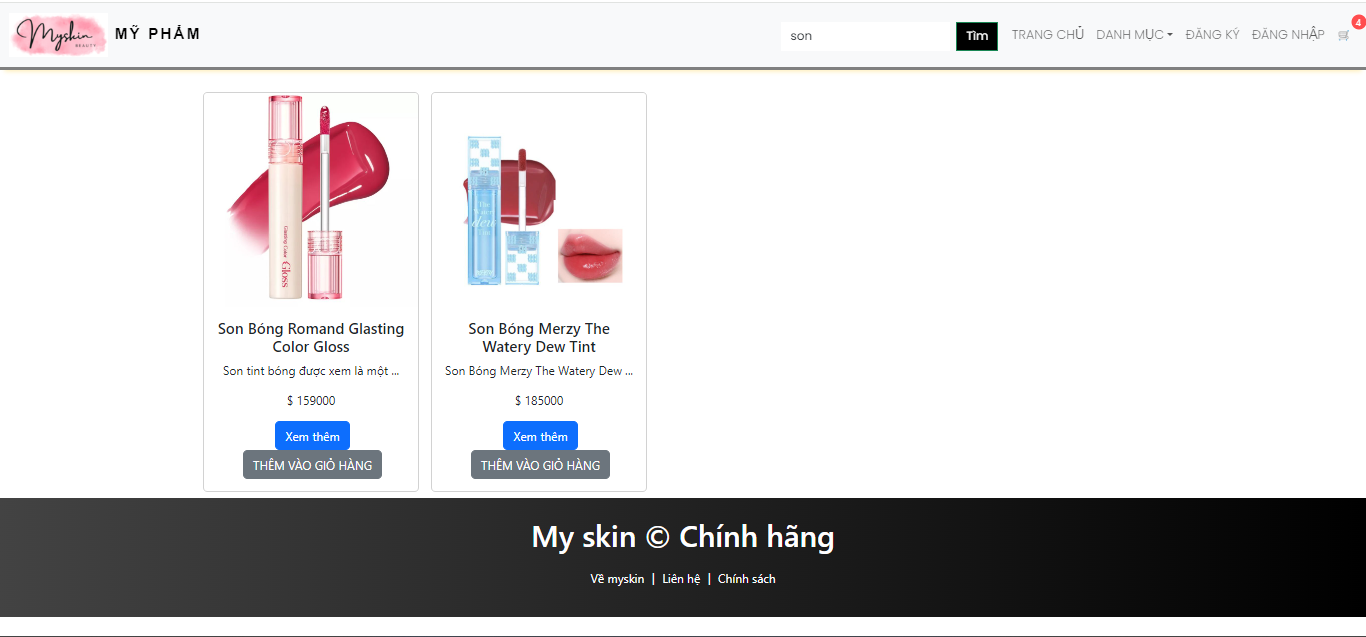
Tài khoản: nguyen@gmail.com

Mật khẩu: 12345678

Số điện thoại: 0365798793

Địa chỉ: Hồ Chí Minh

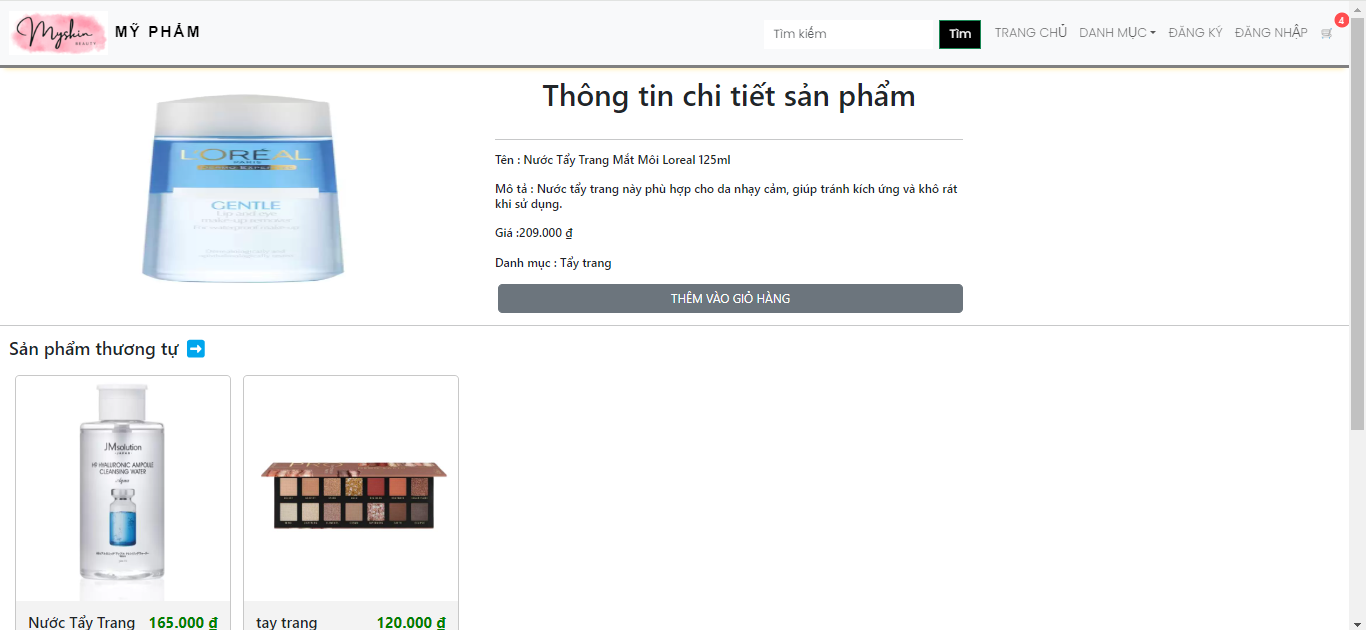
### 3.3.2. Tìm kiếm sản phẩm



Hình 18. Giao diện tìm kiếm

Sau khi đăng ký thành công hệ thống, giao diện sẽ xuất hiện giao diện trang chủ. Ở giao diện này, người dùng có thể tìm kiếm sản phẩm với từ khóa hoặc bộ lọc danh mục và giá sản phẩm.

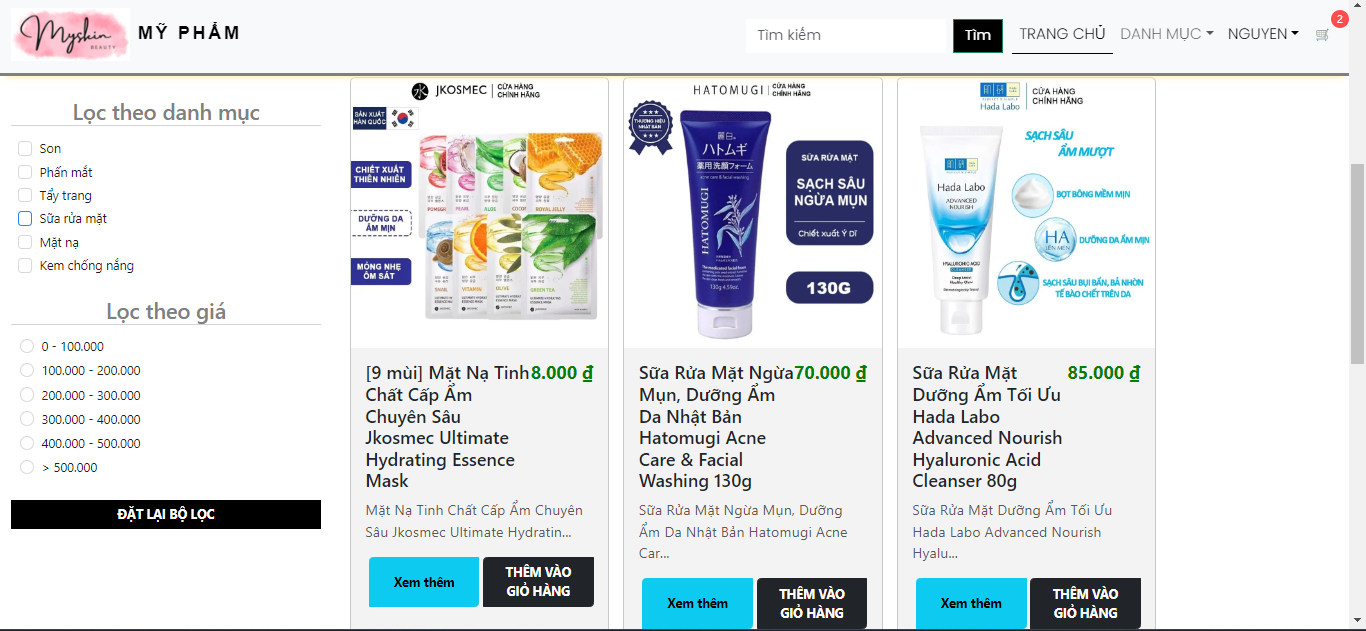
### 3.3.3. Xem chi tiết sản phẩm



Hình 19. Giao diện xem sản phẩm

Người dùng chọn “Xem thêm”, giao diện chi tiết về thông tin sản phẩm sẽ hiển thị. Bên dưới, hệ thống sẽ đề xuất các sản phẩm tương tự.

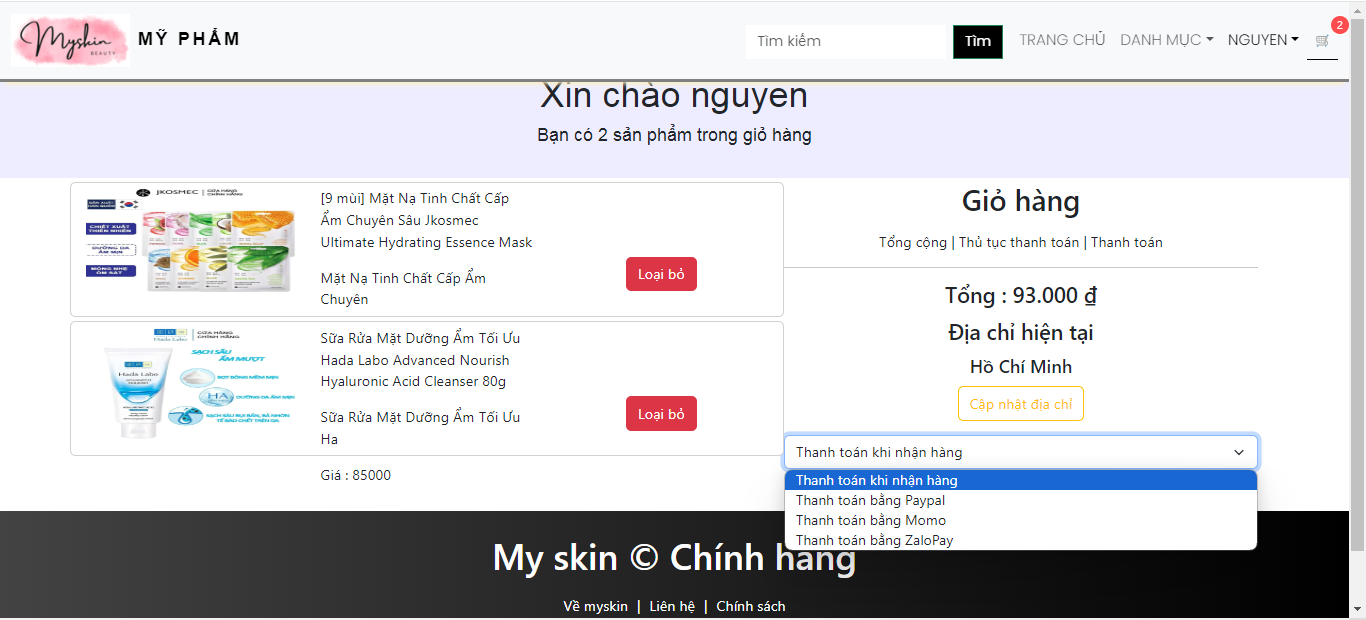
### 3.3.4. Thêm vào giỏ hàng



Hình 20. GIao diện thêm vào giỏ hàng

Người dùng sau khi tìm thấy sản phẩm có thể thêm sản phẩm trực tiếp vào giỏ hàng hoặc xem chi tiết sản phẩm và thêm vào giỏ hàng.

### 3.3.3. Thanh toán



Hình 21. Giao diện thanh toán

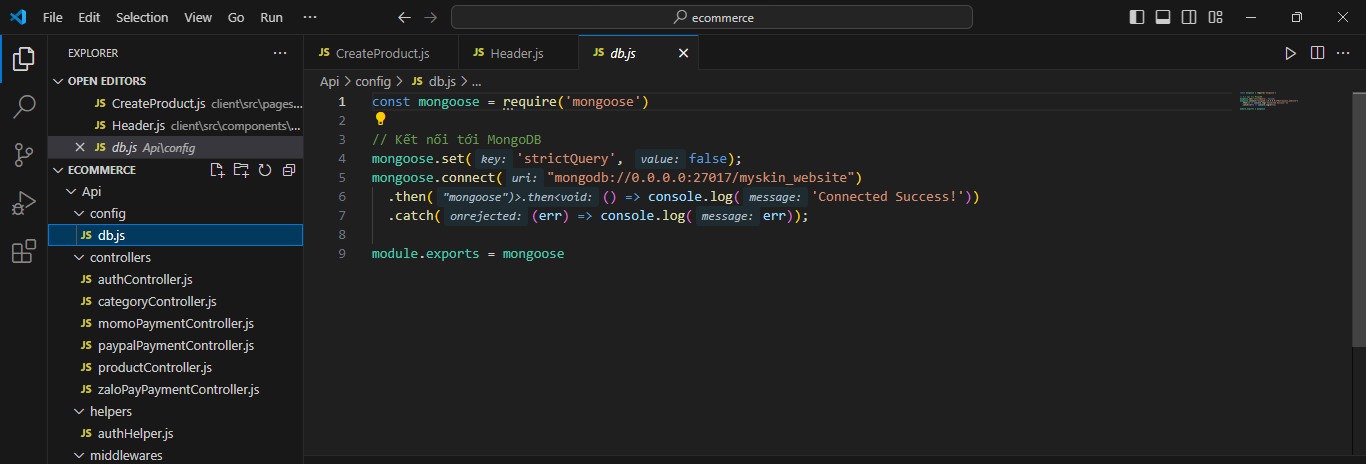
Người dùng lựa chọn các sản phẩm cần mua, hệ thống sẽ tính tổng giá trị tất cả mặt hàng đã được thêm vào giỏ hàng. Người dùng chọn phương thức thanh toán và đặt hàng.

# **CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ**

## **4.1. Cài đặt**

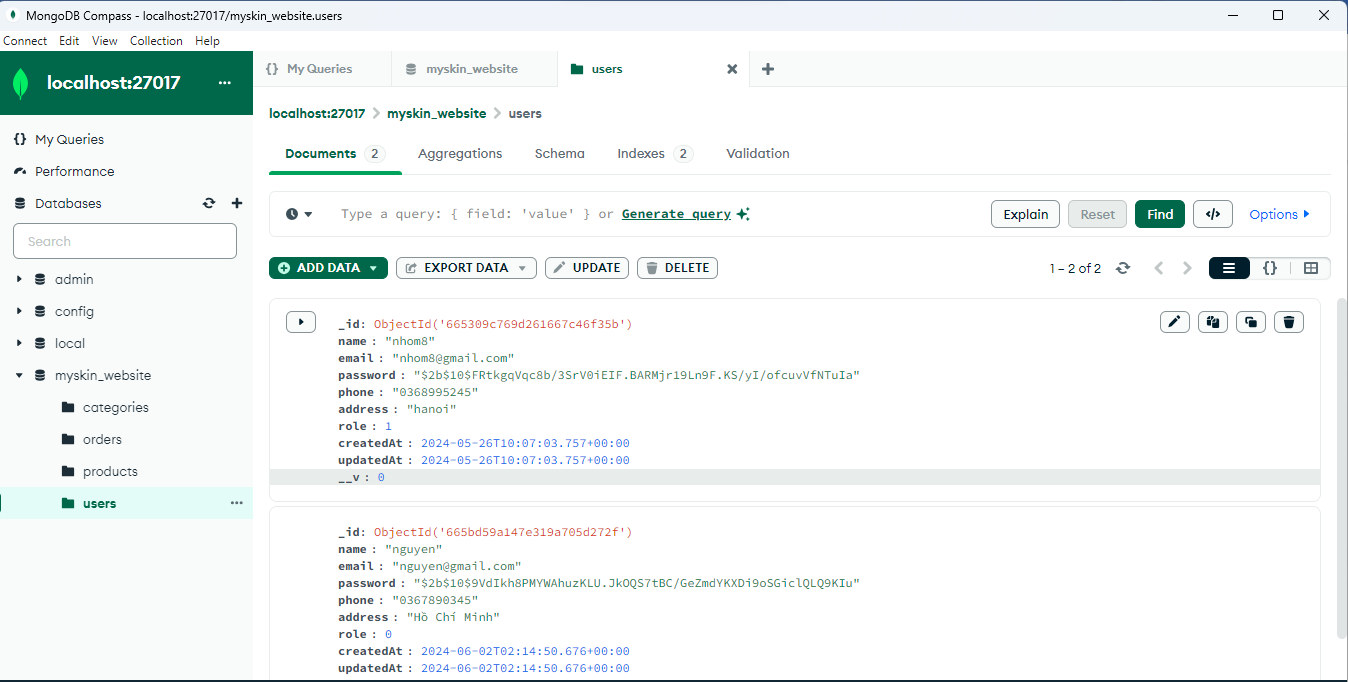
Các bước Visual Studio Code thao tác với dữ liệu với MongoDB

1. Tạo kết nối



Hình 22. Kết nối

Kết nối chương trình với CSDL MongoDB đặt tên là myskin\_website

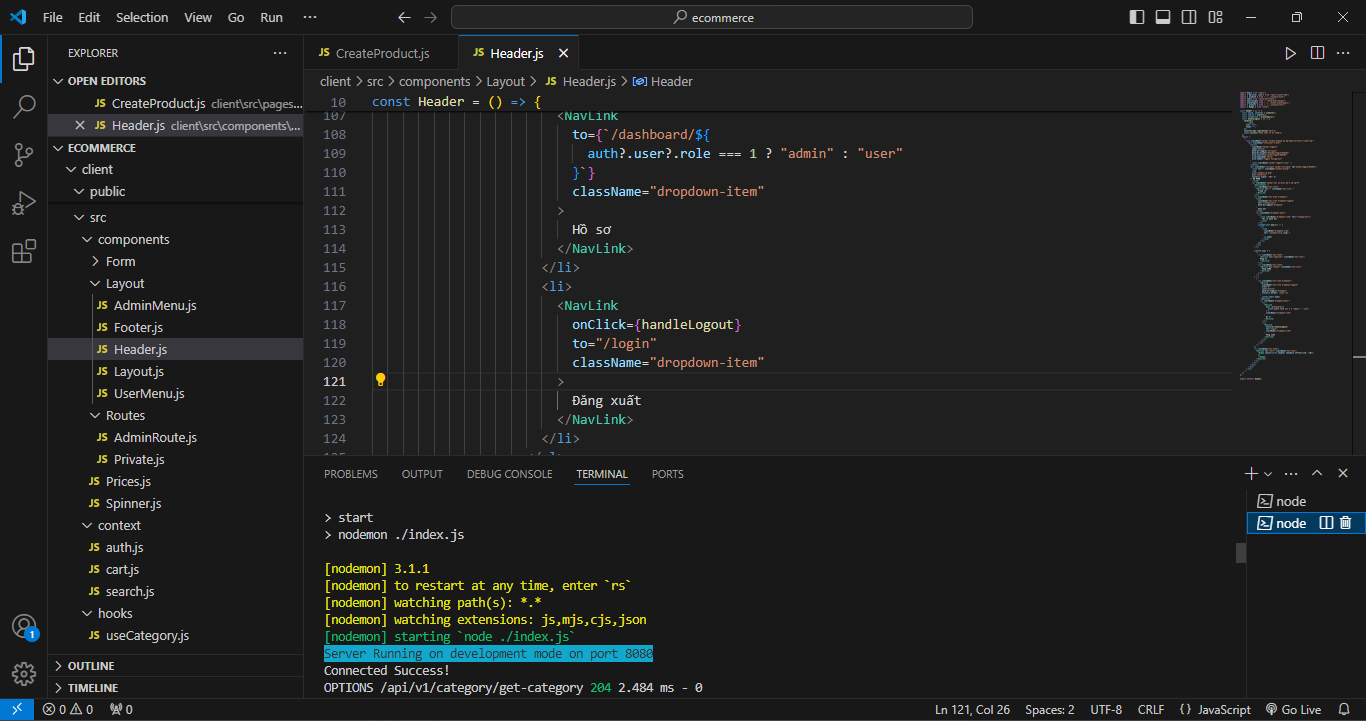


Hình 23. CSDL

1. Mở kết nối

Sử dụng câu lệnh cd . /Api để truy cập vào Server kết nối với CSDL MongoDB

Khỏi động chương trình bằng câu lệnh npm start



Hình 24. Kết nối

Tương tự, ta cũng sử dụng câu lệnh cd . /client và npm start để chạy chương trình

1. Đóng kết nối

Tắt chương trình ứng dụng Visual Studio và trình duyệt hỗ trợ Chrome.

## **4.2. Kiểm thử**

Quá trình kiểm thử cho website thương mại điện tử về mỹ phẩm là một bước cực kỳ quan trọng để đảm bảo rằng website hoạt động chính xác, hiệu quả và đem lại trải nghiệm người dùng tốt nhất.

**Kiểm thử giao diện người dùng (UI Testing):**

- Kiểm tra đảm bảo rằng giao diện người dùng thân thiện, dễ sử dụng và hiển thị đúng trên các thiết bị khác nhau, bao gồm cả điện thoại di động, tablet và máy tính để bàn.

**Kiểm thử chức năng (Functional Testing):**

- Đảm bảo rằng tất cả các chức năng đều hoạt động đúng như mong đợi

Các chức năng: thêm sản phẩm vào giỏ hàng, thực hiện đơn hàng, tìm kiếm sản phẩm, lọc và sắp xếp, tạo tài khoản, đăng nhập/xuất, và phục hồi mật khẩu.

# **TỔNG KẾT**

Sau khi hoàn thành bài tập lớn với đề tài “**Phát triển website thương mại điện tử về mỹ phẩm”** nhóm đã biết cách thức để xây dựng một hệ thống website thương mại điện tử cơ bản. Để có thể hoàn thành đề tài này một cách thành công và đúng thời hạn đều là nhờ sự hướng dẫn giúp đỡ từ giảng viên **Phạm Thị Thanh Mai**.

Khi đánh giá toàn bộ về đề tài, nhóm tổng kết được kết quả đạt được

* **Kết quả đạt được**
* Website đã được thiết kế và triển khai với giao diện người dùng đẹp mắt, thân thiện và dễ dàng sử dụng.
* Các chức năng như tìm kiếm, lọc sản phẩm, thêm vào giỏ hàng, thanh toán và quản lý tài khoản người dùng đã được kiểm thử và hoạt động ổn định.
* Hệ thống bảo mật được trang bị để bảo vệ thông tin khách hàng và dữ liệu thanh toán cơ bản chấp nhận được.
* **Hạn chế**
* Vấn đề bảo mật thông tin khách hàng và dữ liệu giao dịch là rất quan trọng nhưng cũng rất thách thức, đặc biệt khi đối mặt với nguy cơ tấn công mạng.
* Ngành mỹ phẩm là một thị trường với mức độ cạnh tranh cực kỳ cao, việc xây dựng thương hiệu và thu hút khách hàng có thể mất nhiều thời gian và tài nguyên.
* **Hướng phát triển**
* Phát triển ứng dụng di động để gia tăng trải nghiệm người dùng và mở rộng đối tượng khách hàng.
* Tích hợp AI và chatbot để hỗ trợ khách hàng tìm sản phẩm và giải đáp thắc mắc mọi lúc mọi nơi.
* Mở rộng phạm vi kinh doanh qua việc thêm sản phẩm mới và mở rộng thị trường quốc tế.

# **TÀI LIỆU THAM KHẢO**

[1] Khái niệm về hương mại điện tử:

<https://vi.wikipedia.org/wiki/Th%C6%B0%C6%A1ng_m%E1%BA%A1i_%C4%91i%E1%BB%87n_t%E1%BB%AD>

[2] Mối quan hệ RESTful API, RESTful Web Service trong NodeJS

https://stringee.com/vi/blog/post/resrful-api-trong-nodejs-giai-phap-tich-hop-cua-stringee#1-RESTful-API-trong-NodeJS-la-gi

[3] Cơ sở dữ liệu MongoDB

https://itviec.com/blog/mongodb-la-gi/