

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC ĐẠI NAM**



# **BÀI TẬP LỚN**

**TÊN HỌC PHẦN: THIẾT KẾ, LẬP TRÌNH BACK-END**

**ĐỀ TÀI: PHÁT TRIỂN WEBSITE GIỚI THIỆU  
VÀ BÁN CÁC MẶT HÀNG SỮA**

**Giáo viên hướng dẫn: ThS. Phạm Văn Tiệp**

**Sinh viên thực hiện:**

<b>STT</b>	<b>Mã sinh viên</b>	<b>Họ và tên</b>	<b>Lớp</b>
1	1771020729	Nguyễn Thanh Tùng	CNTT 17-15

**Hà Nội, năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: THIẾT KẾ, LẬP TRÌNH BACK-END**

**ĐỀ TÀI: PHÁT TRIỂN WEBSITE GIỚI THIỆU  
VÀ BÁN CÁC MẶT HÀNG SỬA**

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bảng Số	Bảng Chữ
1	1771020729	Nguyễn Thanh Tùng	01/09/2004		

**CÁN BỘ CHẤM THI**

**Hà Nội, năm 2025**

## LỜI NÓI ĐẦU

Cùng với sự phát triển nhanh chóng của công nghệ số, việc ứng dụng các nền tảng trực tuyến trong kinh doanh ngày càng trở nên phổ biến và đóng vai trò quan trọng trong hoạt động thương mại hiện đại. Các hình thức mua sắm truyền thống dần được thay thế bằng các phương tiện mua sắm tiện lợi qua Internet, giúp tiết kiệm thời gian, chi phí và nâng cao trải nghiệm người dùng.

Trong bối cảnh đó, việc xây dựng một website giới thiệu và kinh doanh các sản phẩm thiết yếu như sữa là một hướng đi phù hợp với nhu cầu thực tế và xu hướng phát triển chung. Sữa – với nhiều dạng sản phẩm đa dạng – luôn giữ vị trí quan trọng trong đời sống dinh dưỡng hằng ngày và nhận được sự quan tâm của nhiều đối tượng người tiêu dùng. Việc đưa các sản phẩm này lên môi trường số không chỉ góp phần cải thiện khả năng tiếp cận khách hàng, mà còn giúp nâng cao hiệu quả quản lý và quảng bá thương hiệu.

Đề tài “Phát triển website giới thiệu và bán các mặt hàng sữa” hướng đến việc thiết kế và xây dựng một nền tảng trực tuyến đơn giản, dễ sử dụng và phù hợp với thị hiếu người dùng hiện nay. Đây cũng là cơ hội để người thực hiện áp dụng kiến thức đã học vào thực tế, nâng cao kỹ năng chuyên môn về lập trình web, thiết kế giao diện, xử lý thông tin và triển khai hệ thống ứng dụng thực tiễn.

## MỤC LỤC

LỜI NÓI ĐẦU .....	3
MỤC LỤC .....	4
DANH MỤC HÌNH ẢNH .....	6
DANH MỤC BẢNG BIỂU .....	7
DANH MỤC CÁC TỪ VIẾT TẮT .....	8
CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI .....	9
1.1. Giới thiệu về đề tài .....	9
1.2. Mục tiêu của đề tài .....	9
1.3. Phạm vi của đề tài .....	10
1.4. Phương pháp nghiên cứu .....	12
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT .....	14
2.1. Tổng quan về C# .....	14
2.2. ASP.NET Core MVC .....	15
2.3. SQL Server .....	16
2.4. Entity Framework Core .....	17
2.5. LINQ .....	19
CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG .....	22
3.1. Phân tích đặc tả yêu cầu .....	22
3.1.1. Yêu cầu chức năng .....	22
3.1.2. Yêu cầu phi chức năng .....	23
3.2. Thiết kế hệ thống .....	23
3.2.1. Usecase Diagram .....	23
3.2.2. Class Diagram .....	25
3.2.3. Activity Diagram .....	27

3.2.4. Sequence Diagram .....	31
3.3. Thiết kế cơ sở dữ liệu .....	32
3.4. Thiết kế giao diện .....	35
3.4.1. Giao diện Đăng nhập/ Đăng Ký .....	35
3.4.2. Giao diện khách hàng .....	37
3.4.3. Giao diện Admin .....	40
3.5. Kiểm thử .....	42
3.5.1. Kiểm thử chức năng .....	42
3.5.2. Kiểm thử giao diện .....	43
3.5.3. Kiểm thử hiệu năng .....	43
3.5.4. Kiểm thử bảo mật .....	44
KẾT LUẬN .....	46
TÀI LIỆU THAM KHẢO .....	47

## DANH MỤC HÌNH ẢNH

Hình ảnh 1.0 :Ngôn ngữ C# .....	14
Hình ảnh 2.0: SQL Server .....	16
Hình ảnh 3.1: UserCase Diagram .....	25
Hình ảnh 3.2: Class Diagram .....	27
Hình ảnh 3.4: Sequence Diagram .....	32
Hình ảnh 4.0: Đăng nhập .....	36
Hình ảnh 4.1: Đăng nhập .....	36
Hình ảnh 4.2: Trang chủ .....	37
Hình ảnh 4.3: Xem tất cả sản phẩm .....	37
Hình ảnh 4.4: Chi tiết sản phẩm .....	38
Hình ảnh 4.5: Giỏ Hàng .....	38
Hình ảnh 4.6: Hồ sơ khách hàng .....	38
Hình ảnh 4.7: Giới thiệu Trang web .....	39
Hình ảnh 4.8: Đánh giá nổi bật khách hàng .....	39
Hình ảnh 4.9: Đăng ký tư vấn .....	39
Hình ảnh 5.0: Quản lý tài khoản .....	40
Hình ảnh 5.1: Quản lý khách hàng .....	40
Hình ảnh 5.2: Quản lý sản phẩm .....	40
Hình ảnh 5.3: Chi tiết sản phẩm .....	41
Hình ảnh 5.4: Thêm sản phẩm mới .....	41
Hình ảnh 5.5: Cập nhật sản phẩm .....	41

## DANH MỤC BẢNG BIỂU

Bảng 2.0:Kiểm thử chức năng .....	42
-----------------------------------	----

## DANH MỤC CÁC TỪ VIẾT TẮT

STT	TỪ VIẾT TẮT	VIẾT ĐẦY ĐỦ
1	DBMS	Database Management System
2	LIHQ	Language Integrated Query
3		
4		



# CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

## 1.1. Giới thiệu về đề tài

Trong kỷ nguyên chuyển đổi số, việc ứng dụng công nghệ thông tin vào các hoạt động kinh doanh không còn là xu hướng mới mà đã trở thành một nhu cầu tất yếu để gia tăng hiệu quả vận hành và khả năng cạnh tranh. Thương mại điện tử – đặc biệt là thông qua các nền tảng website – đang dần thay thế phương thức mua sắm truyền thống và khẳng định vị trí quan trọng trong nền kinh tế hiện đại.

Sữa là một trong những loại thực phẩm thiết yếu, chứa nhiều giá trị dinh dưỡng và thường xuyên xuất hiện trong khẩu phần ăn hàng ngày của người dân. Các sản phẩm như sữa tươi, sữa bột và sữa chua phục vụ cho nhiều nhóm đối tượng khác nhau, từ trẻ nhỏ, người cao tuổi, đến phụ nữ mang thai hay người tập thể thao. Tuy vậy, việc phân phối và tiêu thụ các mặt hàng sữa hiện nay tại Việt Nam vẫn còn chịu ảnh hưởng lớn từ các phương thức kinh doanh truyền thống, như bán hàng trực tiếp tại cửa hàng hoặc thông qua các kênh mạng xã hội nhỏ lẻ, thiếu tính chuyên nghiệp và hệ thống.

Từ thực trạng đó, việc xây dựng một website chuyên biệt để giới thiệu và bán các mặt hàng sữa là hoàn toàn phù hợp với nhu cầu thực tiễn và xu hướng phát triển chung của thị trường. Một website có giao diện thân thiện, dễ sử dụng, thông tin rõ ràng và tích hợp đầy đủ chức năng mua sắm sẽ giúp người tiêu dùng tiết kiệm thời gian, tăng tính tiện lợi và đảm bảo nguồn gốc sản phẩm. Đồng thời, đây cũng là giải pháp để đơn vị kinh doanh nâng cao khả năng tiếp cận khách hàng, quản lý hiệu quả sản phẩm – đơn hàng, và phát triển thương hiệu trên nền tảng số.

Đề tài “Phát triển website giới thiệu và bán các mặt hàng sữa tươi, sữa bột và sữa chua” được xây dựng nhằm giải quyết những vấn đề đó, đồng thời tạo điều kiện cho người thực hiện vận dụng kiến thức lý thuyết vào thực hành thông qua một sản phẩm ứng dụng thực tế. Đây không chỉ là một dự án học thuật mà còn mang ý nghĩa thiết thực đối với cộng đồng và phù hợp với xu thế phát triển của nền thương mại điện tử tại Việt Nam hiện nay.

## 1.2. Mục tiêu của đề tài

Trong bối cảnh thương mại điện tử phát triển mạnh mẽ, việc xây dựng một nền tảng bán hàng trực tuyến không chỉ là xu thế tất yếu mà còn là yêu cầu cấp thiết đối với bất kỳ ngành hàng nào, đặc biệt là các sản phẩm tiêu dùng thiết yếu như sữa. Nhận thức được nhu cầu mua sắm tiện lợi, an toàn và nhanh chóng của người tiêu dùng hiện nay, đề tài “Phát triển website giới thiệu và bán các mặt hàng sữa tươi, sữa bột và sữa chua” được triển khai nhằm hướng đến việc xây dựng

một hệ thống website thương mại điện tử hoàn chỉnh, thân thiện với người dùng, vận hành ổn định và có khả năng mở rộng trong tương lai.

Cụ thể, đề tài đặt ra các mục tiêu chính như sau:

- Thiết kế kiến trúc hệ thống theo mô hình MVC (Model - View - Controller): Đây là kiến trúc đa tầng được sử dụng rộng rãi trong lập trình hiện đại. Mục tiêu là phân chia rõ ràng vai trò của từng lớp: Model xử lý dữ liệu và tương tác với cơ sở dữ liệu; View phụ trách hiển thị giao diện người dùng; Controller điều phối luồng dữ liệu giữa hai lớp còn lại. Việc sử dụng mô hình này giúp tăng tính linh hoạt, dễ bảo trì, đồng thời chuẩn hóa quy trình phát triển website.

- Phát triển phần backend bằng ngôn ngữ lập trình C#: Backend đóng vai trò là “bộ não” của hệ thống. Trong đề tài này, các chức năng cốt lõi như quản lý tài khoản người dùng, xử lý đơn hàng, truy xuất cơ sở dữ liệu, xác thực đăng nhập, xử lý giỏ hàng và hệ thống quản trị (admin panel) sẽ được lập trình bằng C# để đảm bảo hiệu suất xử lý cao và bảo mật tốt.

- Thiết kế giao diện hiện đại, dễ dùng bằng HTML, CSS, JavaScript và Tailwind CSS: Giao diện website là yếu tố tác động trực tiếp đến trải nghiệm người dùng. Mục tiêu của đề tài là tạo ra một giao diện đơn giản, rõ ràng, nhưng vẫn đảm bảo tính thẩm mỹ. Tailwind CSS sẽ giúp tăng tốc độ xây dựng layout, tạo phong cách thiết kế đồng bộ, phản hồi tốt trên nhiều thiết bị (responsive design) và dễ bảo trì về lâu dài.

- Xây dựng hệ thống quản lý sản phẩm và đơn hàng: Website sẽ tích hợp đầy đủ chức năng của một cửa hàng trực tuyến như: hiển thị sản phẩm chi tiết (ảnh, mô tả, giá, thành phần), tìm kiếm sản phẩm, thêm vào giỏ hàng, đặt hàng, theo dõi tình trạng đơn hàng. Đồng thời, hệ thống cũng cho phép người quản trị dễ dàng thêm – sửa – xóa sản phẩm, quản lý danh mục và cập nhật thông tin khách hàng.

### 1.3. Phạm vi của đề tài

Đề tài “Phát triển website giới thiệu và bán các mặt hàng sữa tươi, sữa bột và sữa chua” được thực hiện trong phạm vi thiết kế và triển khai một hệ thống website bán hàng trực tuyến cỡ nhỏ, áp dụng kiến thức công nghệ thông tin vào thực tiễn. Hệ thống được xây dựng với định hướng học thuật, tập trung vào nền tảng kỹ thuật và trải nghiệm người dùng, đồng thời phản ánh mô hình hoạt động của một trang thương mại điện tử chuyên nghiệp. Cụ thể, phạm vi của đề tài bao gồm:

- Phạm vi sản phẩm: Website tập trung giới thiệu và bán ba nhóm sản phẩm sữa chính: sữa tươi, sữa bột và sữa chua. Các nhóm sản phẩm khác như phô mai, bơ, sữa đặc không nằm trong nội dung đề tài.

- Phạm vi người dùng: Hệ thống bao gồm hai nhóm người dùng:

+ Người dùng thông thường: Có thể xem sản phẩm, tìm kiếm, thêm vào giỏ hàng và gửi yêu cầu mua hàng.

+ Quản trị viên (Admin): Có thể đăng nhập vào hệ thống quản trị để thêm – sửa – xóa sản phẩm, xem và xử lý các đơn đặt hàng, quản lý danh mục và thông tin khách hàng.

- Phạm vi kỹ thuật và công nghệ: Website được xây dựng dựa trên kiến trúc đa tầng (Multi-tier Architecture), trong đó áp dụng mô hình MVC để phân tách rõ ràng:

#### 1. Web Layer (Tầng trình bày & tiếp nhận yêu cầu):

Là nơi tiếp nhận các yêu cầu HTTP từ người dùng (client-side). Sử dụng HTML, CSS, JavaScript và thư viện Tailwind CSS để hiển thị giao diện người dùng.

Chịu trách nhiệm gửi dữ liệu đến tầng ứng dụng và hiển thị kết quả phản hồi lại người dùng. Cũng bao gồm các View và Controller trong mô hình MVC.

#### 2. Application Layer (Tầng ứng dụng):

Là cầu nối giữa tầng Web và tầng nghiệp vụ. Điều phối các hành vi của hệ thống như xử lý logic giỏ hàng, đặt hàng, đăng nhập, đăng ký... Các controller trong MVC sẽ gọi đến các service trong tầng này. Viết bằng C#, tổ chức theo nguyên tắc SOLID để dễ mở rộng.

#### 3. Domain Layer (Tầng nghiệp vụ – cốt lõi miền):

Chứa các lớp đối tượng nghiệp vụ (entity, value object) và logic xử lý chính, đại diện cho các khái niệm trong thực tế (ví dụ: Sản phẩm, Người dùng, Đơn hàng). Xử lý các quy tắc nghiệp vụ, đảm bảo hệ thống hoạt động đúng theo yêu cầu thực tiễn. Là tầng có mức độ trừu tượng cao nhất và độc lập với kỹ thuật triển khai.

#### 4. Infrastructure Layer (Tầng hạ tầng):

Phụ trách kết nối và tương tác với cơ sở dữ liệu, file, API hoặc các nguồn lưu trữ khác. Viết các lớp thực thi giúp domain layer và application layer dễ dàng thao tác với dữ liệu thật.

Trong đề tài này, dữ liệu người dùng, sản phẩm, đơn hàng được lưu vào hệ quản trị CSDL thông qua các repository được triển khai tại tầng này.

- Phạm vi triển khai: Hệ thống được triển khai và kiểm thử trên máy tính cá nhân (localhost). Chưa tích hợp thanh toán trực tuyến, gửi email, hoặc API vận chuyển.

## 1.4. Phương pháp nghiên cứu

Trong quá trình thực hiện đề tài chúng tôi đã thực hiện đã áp dụng tổng hợp nhiều phương pháp nghiên cứu, từ lý thuyết đến thực nghiệm, nhằm đảm bảo hệ thống được thiết kế chặt chẽ và có tính ứng dụng thực tiễn. Cụ thể, các phương pháp nghiên cứu được triển khai bao gồm:

### 1. Nghiên cứu tài liệu (Lý thuyết)

Tìm hiểu kiến thức nền tảng về thương mại điện tử, giao diện người dùng, kinh doanh trực tuyến các mặt hàng tiêu dùng như sữa. Nghiên cứu mô hình kiến trúc đa tầng trong lập trình ứng dụng web: Web Layer, Application Layer, Domain Layer, và Infrastructure Layer, đồng thời vận dụng mô hình MVC vào tổ chức hệ thống theo từng tầng rõ ràng.

Tham khảo tài liệu về công nghệ sử dụng: C#, HTML/CSS/JavaScript, Tailwind CSS, các quy tắc thiết kế UX/UI, và cách tổ chức cơ sở dữ liệu quan hệ.

### 2. Phân tích hệ thống

Khảo sát các website bán sữa hiện có để đánh giá ưu/nhược điểm và xác định nhu cầu cần giải quyết. Xác định đối tượng người dùng, hành vi tương tác cơ bản (xem sản phẩm, đặt hàng, quản trị nội dung...), từ đó phân tích chức năng và luồng hoạt động tổng thể.

Phân tích chức năng thành từng module cụ thể cho từng tầng của hệ thống: tầng giao diện, tầng xử lý nghiệp vụ và tầng dữ liệu.

### 3. Thiết kế hệ thống

Thiết kế sơ đồ kiến trúc hệ thống theo hướng đa tầng: phân tách rõ giao diện, xử lý nghiệp vụ và truy cập dữ liệu. Thiết kế giao diện người dùng (UI) sử dụng HTML + Tailwind CSS theo nguyên tắc hiện đại, tối ưu thao tác và trải nghiệm khách hàng.

Thiết kế cơ sở dữ liệu bằng mô hình thực thể – liên kết (ERD), chuẩn hóa dữ liệu để đảm bảo tính nhất quán và khả năng mở rộng.

### 4. Cài đặt và lập trình

Lập trình tầng Web với các công nghệ: HTML, CSS, JavaScript, Tailwind để xây dựng giao diện tương tác người dùng.

Lập trình tầng ứng dụng (Application) và tầng nghiệp vụ (Domain) bằng C#, áp dụng nguyên tắc SOLID và tổ chức theo repository pattern/service layer nhằm tách biệt rõ các chức năng nghiệp vụ và truy xuất dữ liệu. Cài đặt tầng hạ tầng (Infrastructure) để xử lý tương tác với cơ sở dữ liệu (chèn, sửa, xóa, truy vấn).

## 5. Kiểm thử và đánh giá

Tiến hành kiểm thử chức năng (Function Testing): kiểm tra đăng nhập, hiển thị sản phẩm, đặt hàng, xử lý đơn hàng... Kiểm thử giao diện (UI Testing) để đảm bảo bố cục đúng, tương thích đa thiết bị, hoạt động ổn định trên các trình duyệt phổ biến.

Đánh giá kết quả theo các tiêu chí: độ ổn định, mức độ hoàn thiện chức năng, hiệu quả giao diện và khả năng mở rộng hệ thống.

## CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

### 2.1. Tổng quan về C#

C# là một ngôn ngữ lập trình hiện đại, thuộc nền tảng .NET của Microsoft, được thiết kế nhằm phục vụ đa dạng nhu cầu phát triển phần mềm, từ ứng dụng web, desktop, đến dịch vụ nền và ứng dụng di động. Điểm mạnh của C# là tính hướng đối tượng rõ ràng, cú pháp dễ tiếp cận, khả năng tích hợp mạnh mẽ và cộng đồng hỗ trợ lớn.



*Hình ảnh 1.0 :Ngôn ngữ C#*

Ngoài ra nó còn là một trong những ngôn ngữ có cú pháp thân thiện với lập trình viên chịu ảnh hưởng bởi C / C+ và Java. Ngôn ngữ C# cũng hỗ trợ lập trình với hướng đối tượng từ đó giúp cho ta dễ dàng tổ chức và tái sử dụng lại code. Về tính năng an toàn kiểm tra kiểu dữ liệu chặt chẽ và cơ chế quản lý bộ nhớ tự động (garbage collector). Thư viện của nó rất phong phú: cung cấp hàng ngàn lớp và rất nhiều hàm đã có sẵn phục vụ xử lý chuỗi, tệp mạng, cơ sở .... Không những thế nó còn có khả năng tích hợp với các công nghệ hiện đại như: ASP.NET Core, Entity Framework, và các API web.

Vai trò của C# trong đề tài này là vô cùng quan trọng nó đóng vai trò cho sự sống còn của phía BackEnd cụ thể gồm:

- Xử lý logic nghiệp vụ như thêm sản phẩm vào giỏ hàng, xử lý đơn đặt hàng, xác thực người dùng...
- Tương tác với cơ sở dữ liệu thông qua Entity Framework Core và LINQ.
- Tổ chức mã nguồn theo mô hình MVC và kiến trúc đa tầng, đảm bảo hệ thống rõ ràng, dễ mở rộng và dễ bảo trì.

Nhờ vào sự ổn định và khả năng mở rộng cao, C# là lựa chọn phù hợp giúp đảm bảo tính an toàn, hiệu năng và sự chuyên nghiệp cho hệ thống web được phát triển trong khuôn khổ đề tài này.

## 2.2. ASP.NET Core MVC

ASP.NET Core MVC là một framework thuộc nền tảng ASP.NET Core, dùng để xây dựng các ứng dụng web động theo mô hình MVC (Model – View – Controller). Đây là một kiến trúc giúp phân tách rõ ràng phần giao diện (View), xử lý yêu cầu (Controller), và dữ liệu – nghiệp vụ (Model), từ đó giúp quá trình phát triển phần mềm được rõ ràng, dễ kiểm soát và dễ mở rộng.

- Ưu điểm của việc kết hợp ASP.NET Core MVC trong kiến trúc đa tầng

- Tách biệt rõ trách nhiệm (Separation of Concerns): View chỉ xử lý giao diện, Controller điều hướng, trong khi Domain xử lý logic nghiệp vụ, giúp hệ thống dễ hiểu và dễ bảo trì.
- Mã nguồn dễ kiểm thử và bảo trì: Các tầng riêng biệt giúp dễ dàng viết unit test hoặc thay thế một thành phần mà không ảnh hưởng đến toàn bộ hệ thống.
- Khả năng tái sử dụng cao: Các logic nghiệp vụ và logic dữ liệu có thể tái sử dụng giữa các ứng dụng hoặc module khác nhau mà không phụ thuộc vào tầng giao diện.
- Hỗ trợ quy mô lớn: Mô hình này phù hợp cho cả dự án nhỏ và mở rộng được với những hệ thống có lượng người dùng lớn, tích hợp nhiều chức năng phức tạp sau này.
- Cộng đồng hỗ trợ mạnh mẽ: ASP.NET Core MVC có tài liệu đầy đủ, cộng đồng đông đảo, dễ tra cứu và học hỏi.

- Nhược điểm và thách thức khi áp dụng

- Yêu cầu người phát triển nắm chắc mô hình kiến trúc: Cần hiểu rõ vai trò của từng tầng, nếu không sẽ dễ dẫn đến chồng chéo hoặc viết code sai vị trí.
- Phức tạp hơn ban đầu: Với dự án nhỏ, việc chia tầng nhiều có thể tạo cảm giác "quá tải", mất thêm thời gian cấu hình ban đầu.
- Tăng số lượng lớp và file mã nguồn: Dễ gây rối nếu không quản lý mã nguồn tốt, nhất là khi làm việc nhóm hoặc khi cập nhật luồng xử lý.

### 2.3. SQL Server

Microsoft SQL Server là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển, sử dụng ngôn ngữ T-SQL để quản lý, lưu trữ và truy vấn dữ liệu. Đây là giải pháp phổ biến cho các ứng dụng doanh nghiệp yêu cầu độ ổn định, bảo mật cao và khả năng xử lý giao dịch đồng thời (concurrency) lớn.

Mục đích và công dụng chính của SQL Server trong hệ thống:

- Lưu trữ toàn bộ dữ liệu quan hệ như thông tin người dùng, danh mục sản phẩm (sữa tươi, sữa bột, sữa chua), giỏ hàng, đơn hàng và nhật ký hệ thống.
- Đảm bảo tính toàn vẹn dữ liệu (ACID) khi thực hiện các giao dịch thêm, sửa, xóa đa bước.
- Cung cấp công cụ truy vấn mạnh mẽ để lọc, tính toán và tổng hợp dữ liệu, phục vụ cho báo cáo hoặc thống kê bán hàng.
- Cho phép sao lưu (backup) và phục hồi (restore) dữ liệu dễ dàng, duy trì tính sẵn sàng và bảo vệ dữ liệu trước sự cố.



*Hình ảnh 2.0: SQL Server*

Trong đề tài “Phát triển website giới thiệu và bán các mặt hàng sữa...”, SQL Server được đặt tại Infrastructure Layer của kiến trúc đa tầng. Cụ thể:

- Tầng Domain định nghĩa các lớp đối tượng (entity) tương ứng với các bảng trong cơ sở dữ liệu.
- Tầng Application sử dụng Entity Framework Core để gửi yêu cầu truy vấn hoặc cập nhật dữ liệu thông qua các repository, service.
- Tầng Web (MVC) điều phối yêu cầu người dùng đến tầng Application, tầng Application chuyển tiếp đến SQL Server qua Entity Framework.



=> Nhờ vậy, toàn bộ hoạt động như đăng ký/đăng nhập, thêm sản phẩm vào giỏ, đặt hàng, xem lịch sử đơn... đều được lưu và xử lý ổn định qua SQL Server.

✧ Ưu điểm nổi bật:

- Khả năng xử lý giao dịch song song tốt, đảm bảo tính toàn vẹn và nhất quán dữ liệu.
- Tính năng bảo mật cao (authentication, authorization, encryption) giúp bảo vệ thông tin khách hàng và đơn hàng.
- Công cụ quản trị (SQL Server Management Studio) và tích hợp chặt chẽ với .NET/C# giúp phát triển, gỡ lỗi và tối ưu dễ dàng.
- Hỗ trợ dữ liệu lớn, tối ưu chỉ mục, cateches, phân vùng để tăng hiệu năng.

✧ Nhược điểm và lưu ý:

- Là phần mềm bản quyền, chi phí cấp phép cho môi trường sản xuất có thể cao.
- Cấu hình và tối ưu hóa ban đầu đòi hỏi kiến thức chuyên sâu về quản trị cơ sở dữ liệu.
- Triển khai đa nền tảng còn hạn chế hơn so với các giải pháp mã nguồn mở như MySQL, PostgreSQL.

Tóm lại SQL Server đóng vai trò đảm bảo lưu trữ và truy xuất mọi thông tin về người dùng, sản phẩm, giỏ hàng .... Nhờ vào khả năng xử lý giao dịch mạnh mẽ, bảo mật cao an toàn. Cùng tính năng backup/restore linh hoạt, hệ thống thương mại điện tử bán sữa của đề tài hoạt động ổn định, hiệu quả và dễ dàng mở rộng khi cần thiết.

## 2.4. Entity Framework Core

Entity Framework Core (EF Core) là thư viện Object-Relational Mapper (ORM) đa nền tảng do Microsoft phát triển, cho phép lập trình viên .NET tương tác với cơ sở dữ liệu quan hệ thông qua các đối tượng C#. Thay vì viết thủ công các câu lệnh SQL, bạn chỉ cần làm việc với lớp, thuộc tính và phương thức trong C#, EF Core sẽ tự động sinh và thực thi SQL phù hợp.

1. Nguyên lý hoạt động:

- EF Core vận hành dựa trên hai thành phần chính:

+ DbContext: Là cầu nối giữa ứng dụng và cơ sở dữ liệu. Mỗi DbContext chứa một tập hợp các DbSet đại diện cho bảng trong database. Khi khởi tạo, DbContext quản lý kết nối, theo dõi sự thay đổi của các thực thể và điều phối giao dịch.

+ DbSet: Tượng trưng cho một tập thực thể (entity) trong database. Bạn thao tác thêm, sửa, xóa hoặc truy vấn qua DbSet, và khi gọi phương thức SaveChanges(), EF Core sẽ chuyển đổi những thay đổi đó thành các câu lệnh SQL tương ứng.

- Quá trình làm việc điển hình gồm:

+ Lấy dữ liệu: Viết biểu thức LINQ trên DbSet để EF Core sinh SQL truy vấn.

+ Thay đổi dữ liệu: Thêm, sửa hoặc xóa thực thể trong DbSet.

+ Lưu thay đổi: Gọi SaveChanges() để EF Core thực thi giao dịch (transaction), đảm bảo tính toàn vẹn ACID.

## 2. Hai chiến lược chính

- Code-First

+ Bạn định nghĩa lớp thực thể và mối quan hệ qua C# (dùng data annotations hoặc Fluent API). Sau đó, EF Core tạo hoặc cập nhật cấu trúc database thông qua cơ chế migrations. Phù hợp khi dự án khởi tạo từ đầu và chuẩn hóa mô hình nghiệp vụ trong code.

- Database-First

+ EF Core đọc cấu trúc database đã tồn tại và tự động sinh các lớp thực thể tương ứng. Tính năng scaffolding giúp bạn nhanh chóng xây dựng model dựa trên các bảng, view hay stored procedure có sẵn.

## 3. Cấu hình và Migrations

Cấu hình kết nối: Trong lớp DbContext, bạn ghi connection string trong phương thức OnConfiguring() hoặc cấu hình qua service container trong ASP.NET Core.

Tạo migration: Sử dụng lệnh dotnet ef migrations add để EF Core so sánh model hiện tại với database, sinh file migration.

Áp dụng migration: Chạy dotnet ef database update để cập nhật cấu trúc database theo file migration.

## 4. Ứng dụng trong đề tài

Trong đề tài “Phát triển website giới thiệu và bán các mặt hàng sữa...”, EF Core nằm ở Infrastructure Layer, chịu trách nhiệm kết nối và thao tác dữ liệu trên SQL Server:

+ Khai báo entity: Các lớp User, Product, Order... định nghĩa thuộc tính, quan hệ và ràng buộc cơ bản (ví dụ: khóa chính, khóa ngoại, độ dài chuỗi).

+ Khởi tạo ApplicationDbContext: Kế thừa DbContext, khai báo DbSet cho từng entity và cấu hình mapping nếu cần.

+ Giao tiếp qua Application Layer: Repository hoặc service trong Application Layer inject AppDbContext, gọi phương thức LINQ để truy vấn hoặc cập nhật.

+ Quản lý migration: Mỗi khi thay đổi cấu trúc entity, bạn tạo migration mới để tự động đồng bộ database, giúp việc phát triển và nâng cấp hệ thống diễn ra mạch lạc.

## 5. Ưu – nhược điểm

### - Ưu điểm:

- + Giảm thiểu viết SQL thủ công, đẩy nhanh tiến độ phát triển.
- + Tự động quản lý transaction và change tracking, đảm bảo tính nhất quán dữ liệu.
- + Hỗ trợ mạnh mẽ cho Unit Testing khi sử dụng DbContext in-memory.
- + Tích hợp sẵn với ASP.NET Core và Dependency Injection.

### - Nhược điểm:

+ Vướng phải overhead của ORM, đôi khi gây chậm với truy vấn phức tạp hoặc khối lượng lớn.

- + Cần hiểu rõ cách EF Core sinh SQL để tránh truy vấn không tối ưu.
- + Khi cần tối ưu sâu, vẫn phải viết thủ công SQL hoặc stored procedure.

Entity Framework Core (EF Core) là thư viện ORM của .NET, giúp kết nối các lớp C# với cơ sở dữ liệu quan hệ mà không phải viết SQL thủ công. Nhờ nó mà khiến cho chúng ta đẩy nhanh tiến độ phát triển và giữ cho tầng Infrastructure luôn đồng bộ nhất quán.

Nhưng cũng cần lưu ý với những truy vấn cực nặng hoặc tối ưu chuyên sâu, bạn vẫn nên can thiệp trực tiếp bằng SQL hoặc stored procedure để đạt được hiệu suất cao nhất.

## 2.5. LINQ

LINQ là tập hợp các công nghệ do Microsoft phát triển, cho phép tích hợp khả năng truy vấn dữ liệu ngay trong ngôn ngữ C# (và các ngôn ngữ .NET khác) một cách thống nhất, an toàn và trực quan.

1. Mục đích: LINQ (Language Integrated Query) mang đến cú pháp truy vấn gần giống SQL nhưng được viết ngay trong C#. Mục tiêu chính của LINQ là:

- Cung cấp một giao diện truy vấn chung cho nhiều nguồn dữ liệu khác nhau (mảng, danh sách, XML, cơ sở dữ liệu quan hệ...).
- Giảm thiểu việc phải viết tay chuỗi SQL hay thao tác DOM XML một cách thủ công.

- Đảm bảo an toàn kiểu (type safety) và tận dụng khả năng kiểm tra lỗi ngay thời điểm biên dịch (compile-time).

## 2. Thành phần cốt lõi

### 1. *Standard Query Operators*

Là tập hợp các phương thức mở rộng (extension methods) như *Where*, *Select*, *OrderBy*, *GroupBy*, *Join*... được định nghĩa trong *namespace System.Linq*. Đây chính là “các phép toán” mà LINQ cho phép bạn sử dụng để lọc, chiếu, sắp xếp, nhóm hoặc ghép nối dữ liệu.

### 2. *Expression Trees*

Khi bạn viết truy vấn dưới dạng IQueryable, LINQ xây dựng thành cây biểu thức (expression tree) thể hiện cú pháp truy vấn. Mỗi nút trong cây biểu thức mô tả một phép toán, và provider (ví dụ EF Core) sẽ dịch cây này thành SQL hoặc ngôn ngữ truy vấn tương ứng.

### 3. *Providers*

Là các thành phần chịu trách nhiệm “dịch” biểu thức LINQ thành các lệnh thao tác nguồn dữ liệu cụ thể. Ví dụ:

- LINQ to Objects: thực thi truy vấn trên các đối tượng trong bộ nhớ (IEnumerable).
- LINQ to Entities: chuyển thành SQL tương tác với cơ sở dữ liệu qua Entity Framework.
- LINQ to XML, LINQ to DataSet...: tương tác với XML, DataTable, v.v.

## 3. Cơ chế hoạt động

- *Deferred Execution (Thực thi trì hoãn):*

Truy vấn LINQ với IQueryable chỉ xây dựng biểu thức và chưa thực thi ngay SQL; chỉ khi bạn “tiêu thụ” kết quả (ví dụ gọi ToList(), FirstOrDefault()) thì truy vấn mới được gửi xuống datasource.

- *Immediate Execution (Thực thi ngay):*

Một số phương thức như Count(), Sum(), Max() sẽ sinh ra truy vấn và thực thi ngay lập tức để trả về kết quả đơn lẻ.

- *Change Tracking (với EF Core):*

Khi dùng LINQ to Entities, các đối tượng được trả về nằm trong DbContext, EF Core tự động theo dõi sự thay đổi và gom các thao tác cập nhật vào trong giao dịch.

## 4. Các dạng LINQ phổ biến

- LINQ to Objects: Truy vấn trên tập hợp trong bộ nhớ như mảng, danh sách, dictionary.
- LINQ to Entities: Truy vấn cơ sở dữ liệu quan hệ qua Entity Framework Core.
- LINQ to XML: Xử lý và truy vấn tài liệu XML dưới dạng đối tượng.
- LINQ to DataSet/DataTable: Làm việc với ADO.NET DataSet, DataTable.

## 5. Ưu điểm

- Thống nhất cú pháp truy vấn cho nhiều nguồn dữ liệu, giúp lập trình viên không phải học nhiều API khác nhau.
- An toàn kiểu – lỗi được phát hiện ngay lúc biên dịch, tránh runtime error.
- Tính đọc được cao – mã truy vấn gọn gàng, dễ hiểu và dễ bảo trì.
- Tích hợp sâu với C# – hỗ trợ IntelliSense, refactoring và debugging trực tiếp trong IDE.

## 6. Hạn chế

- Hiệu năng đôi lúc không tối ưu với các truy vấn phức tạp hoặc khi provider sinh SQL không hiệu quả.
- Giới hạn khả năng biểu đạt so với SQL thuần túy (ví dụ pivot, window function), đôi khi phải kết hợp với stored procedure.
- Độ phức tạp của Expression Tree có thể gây khó khăn khi debug hoặc xử lý lỗi do provider sinh ra.

=> LINQ đóng vai trò quan trọng trong đề tài khi giúp Application Layer và Infrastructure Layer của bạn có một cơ chế truy vấn dữ liệu chung, dễ đọc, dễ mở rộng và an toàn từ miền nghiệp vụ (Domain) đến cơ sở dữ liệu (SQL Server).

## CHƯƠNG 3. XÂY DỰNG ỨNG DỤNG

### 3.1. Phân tích đặc tả yêu cầu

Phân tích đặc tả yêu cầu là bước quan trọng để xác định các tính năng và đặc điểm cần thiết cho website giới thiệu và bán các mặt hàng sữa. Phần này bao gồm các yêu cầu chức năng, phi chức năng, ràng buộc, giả định, và rủi ro để đảm bảo hệ thống đáp ứng nhu cầu của khách hàng và quản trị viên.

#### 3.1.1. Yêu cầu chức năng

Yêu cầu chức năng xác định các tính năng chính của website, phục vụ khách hàng và quản trị viên.

##### *a. Chức năng dành cho khách hàng*

- Duyệt và tìm kiếm sản phẩm:
  - + Xem danh sách sản phẩm sữa (sữa bột, sữa tươi, sữa chua, v.v.) với thông tin chi tiết: tên, giá, Thành phần và khuyến mãi.
  - + Tìm kiếm sản phẩm theo tên hoặc danh mục.
  - + Lọc sản phẩm theo loại sữa, hoặc thương hiệu.
- Giỏ hàng và thanh toán:
  - + Thêm, chỉnh sửa, hoặc xóa sản phẩm trong giỏ hàng.
  - + Tính tổng giá trị đơn hàng (bao gồm thuế và phí vận chuyển).
- Quản lý tài khoản:
  - + Đăng ký/đăng nhập bằng email hoặc số điện thoại.
  - + Cập nhật thông tin cá nhân (tên, địa chỉ, số điện thoại).
  - + Xem lịch sử đơn hàng và lưu sản phẩm yêu thích.
- Tương tác với sản phẩm:
  - + Đánh giá (rating) và bình luận sản phẩm sau khi mua.
  - + Xem đánh giá từ khách hàng khác.

- Hỗ trợ khách hàng:
  - + Liên hệ qua biểu mẫu
  - + Hiện thị thông tin liên hệ (email, số điện thoại)

#### ***b. Chức năng dành cho quản trị viên***

- Quản lý sản phẩm:
  - + Thêm, sửa, xóa sản phẩm (hình ảnh, mô tả, giá, tồn kho).
  - + Quản lý danh mục sản phẩm và chương trình khuyến mãi.
- Quản lý đơn hàng:
  - + Xem và cập nhật trạng thái đơn hàng (chờ xử lý, đang giao, hoàn thành).
  - + Thông báo trạng thái đơn hàng cho khách hàng.
- Quản lý người dùng:
  - + Xem danh sách khách hàng và khóa/mở tài khoản nếu cần.
- Báo cáo: Xem báo cáo doanh thu, sản phẩm bán chạy, và tồn kho.

#### **3.1.2. Yêu cầu phi chức năng**

Yêu cầu phi chức năng đảm bảo chất lượng và hiệu suất của website.

- + Hiệu suất: Tải trang trong vòng 2-3 giây trên kết nối internet thông thường.
- + Hỗ trợ tối thiểu 500 người dùng truy cập đồng thời.
- + Khả năng mở rộng: Dễ dàng thêm danh mục sản phẩm hoặc tích hợp tính năng mới.
- + Bảo mật: Sử dụng HTTPS để mã hóa dữ liệu. Bảo vệ thông tin khách hàng theo quy định pháp luật.
- + Xác thực người dùng bằng OTP khi đăng ký/đăng nhập.
- + Khả năng sử dụng: Giao diện responsive, thân thiện trên máy tính và di động. Hỗ trợ người dùng khuyết tật (tương thích trình đọc màn hình).
- + Khả năng bảo trì: Mã nguồn rõ ràng, có tài liệu hướng dẫn. Sao lưu dữ liệu định kỳ.

### **3.2. Thiết kế hệ thống**

#### **3.2.1. Usecase Diagram**

Biểu đồ Usecase là một công cụ trực quan trong phân tích và thiết kế hệ thống, giúp thể hiện rõ ràng mối quan hệ giữa người dùng (tác nhân) và các chức năng chính mà hệ thống cung cấp. Trong trường hợp này, biểu đồ Usecase được xây dựng nhằm mô tả toàn diện các chức năng từ góc nhìn của hai nhóm người dùng chính: *Khách hàng* và *Quản trị viên*.

- Các tác nhân (Actors):

+ Khách hàng: Người dùng cuối thực hiện mua sắm, đăng ký tư vấn, và đánh giá sản phẩm.

+ Quản trị viên: Người quản lý sản phẩm, đơn hàng, và khách hàng.

- Các chức năng :

■ Khách hàng:

Xem sản phẩm

Quản lý giỏ hàng

Đặt đơn hàng

Đăng ký tài khoản

Đăng ký tư vấn

Đánh giá sản phẩm

■ Quản trị viên:

Quản lý sản phẩm

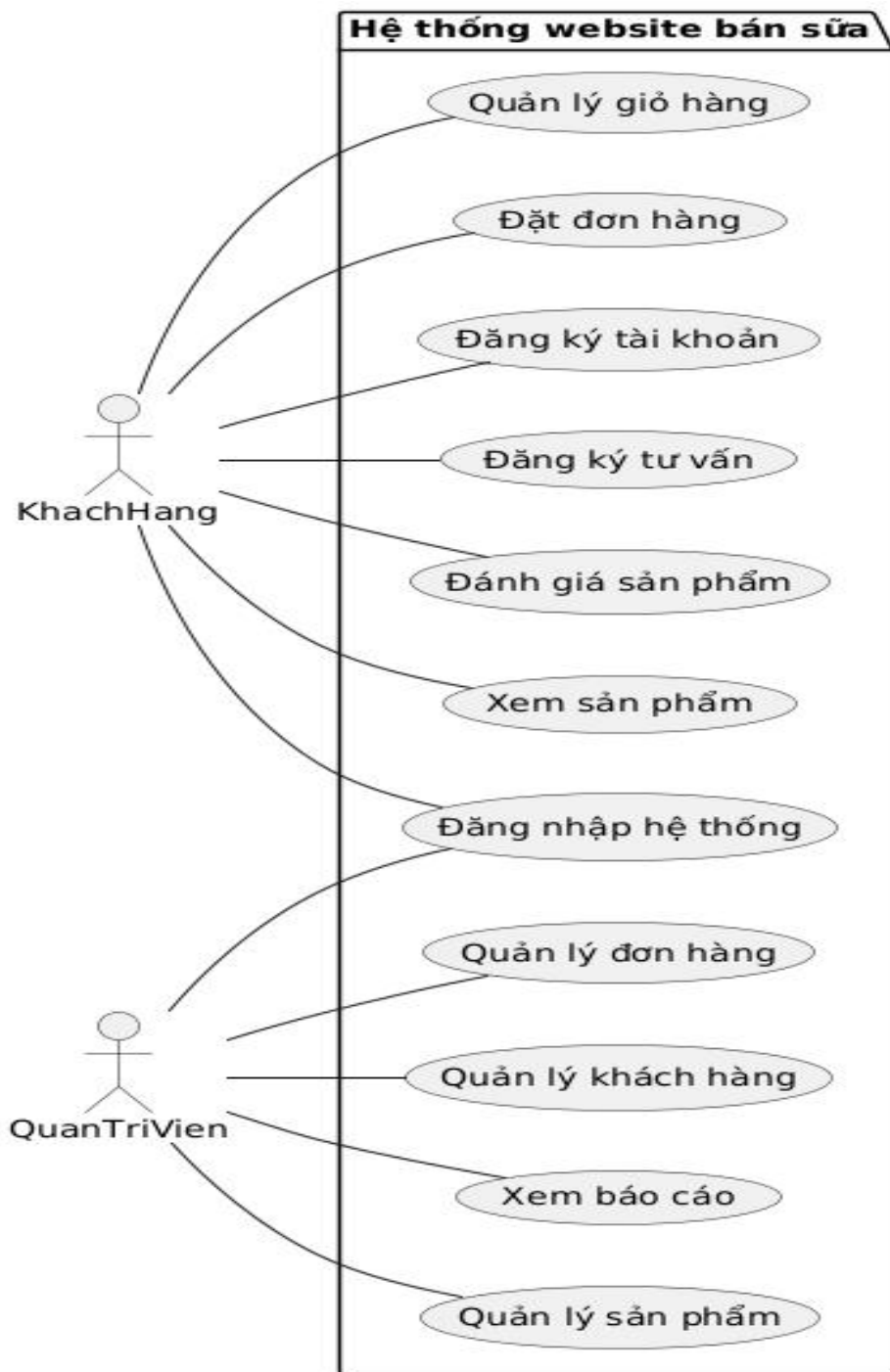
Quản lý đơn hàng

Quản lý khách hàng

Xem báo cáo

Biểu đồ Usecase sử dụng các tác nhân *KhachHang* và *QuanTriVien*, với các usecase được nhóm trong package "Hệ thống website bán sữa". Biểu đồ bên dưới:





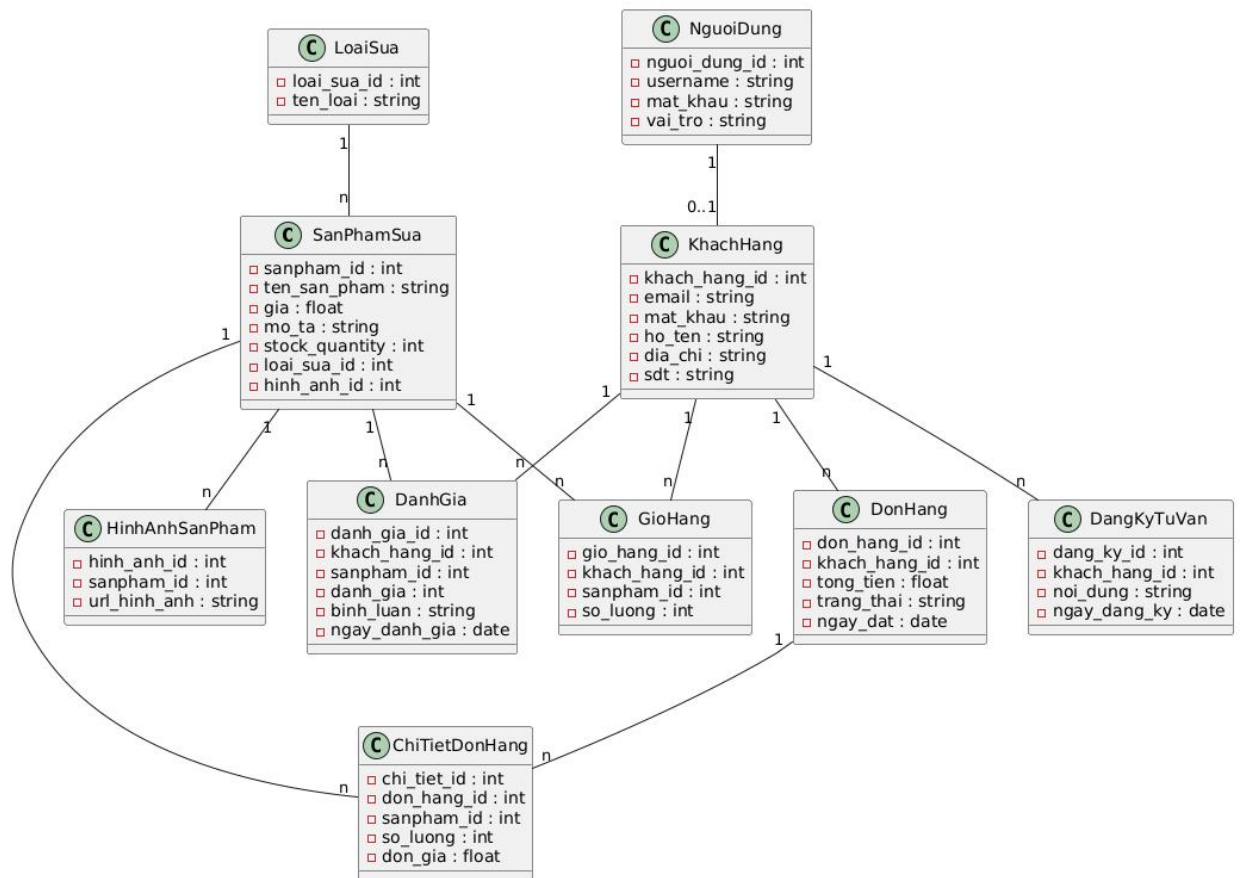
Hình ảnh 3.1: UserCase Diagram

### 3.2.2. Class Diagram

Trong phần này chúng tôi sẽ vẽ biểu đồ thực thể chính trong hệ thống, từ đó cho ta thấy được mối quan hệ giữa chúng. Để hiểu rõ hơn hãy xem bảng phía dưới đây nói về thuộc tính và quan hệ.

<i>Thực thể</i>	<i>Thuộc tính</i>	<i>Quan hệ</i>
SanPhamSua	sanpham_id (khóa chính), ten_san_pham, gia, mo_ta,	Liên kết với LoaiSua (1-n)

	stock_quantity, loai_sua_id, hình_anh_id.	và ChiTietDonHang (1-n).
LoaiSua	loai_sua_id (khóa chính), ten_loai.	SanPhamSua (1-n).
KhachHang	khach_hang_id (khóa chính), email, mat_khau, ho_ten, dia_chi, sdt.	DonHang (1-n) và DanhGia (1-n).
DonHang	don_hang_id (khóa chính), khach_hang_id, tong_tien, trang_thai, ngay_dat.	KhachHang (n-1) và ChiTietDonHang (1-n).
ChiTietDonHang	chi_tiet_id (khóa chính), don_hang_id, sanpham_id, so_luong, don_gia.	DonHang (n-1) và SanPhamSua (n-1).
DanhGia	danh_gia_id (khóa chính), khach_hang_id, sanpham_id, danh_gia, binh_luan, ngay_danh_gia.	KhachHang (n-1) và SanPhamSua (n-1).
GioHang	gio_hang_id (khóa chính), khach_hang_id, sanpham_id, so_luong.	KhachHang (n-1) và SanPhamSua (n-1).
HinhAnhSanPham NguoiDung	hình_anh_id (khóa chính), sanpham_id, url_hình_anh. nguai_dung_id (khóa chính), username, mat_khau, vai_tro.	SanPhamSua (n-1). KhachHang hoặc Quản trị viên.
DangKyTuVan	dang_ky_id (khóa chính), khach_hang_id, noi_dung, ngay_dang_ky.	KhachHang (n-1).



Hình ảnh 3.2: Class Diagram

### 3.2.3. Activity Diagram

Bên trên tôi đã nói về UseCase Diagram và Class Diagram rồi ở phần này tôi sẽ nói đến Biểu đồ hoạt động mô tả luồng xử lý của quy trình đặt hàng, một chức năng cốt lõi của website bán sữa, liên quan đến các thực thể như KhachHang, GioHang, DonHang, và ChiTietDonHang.

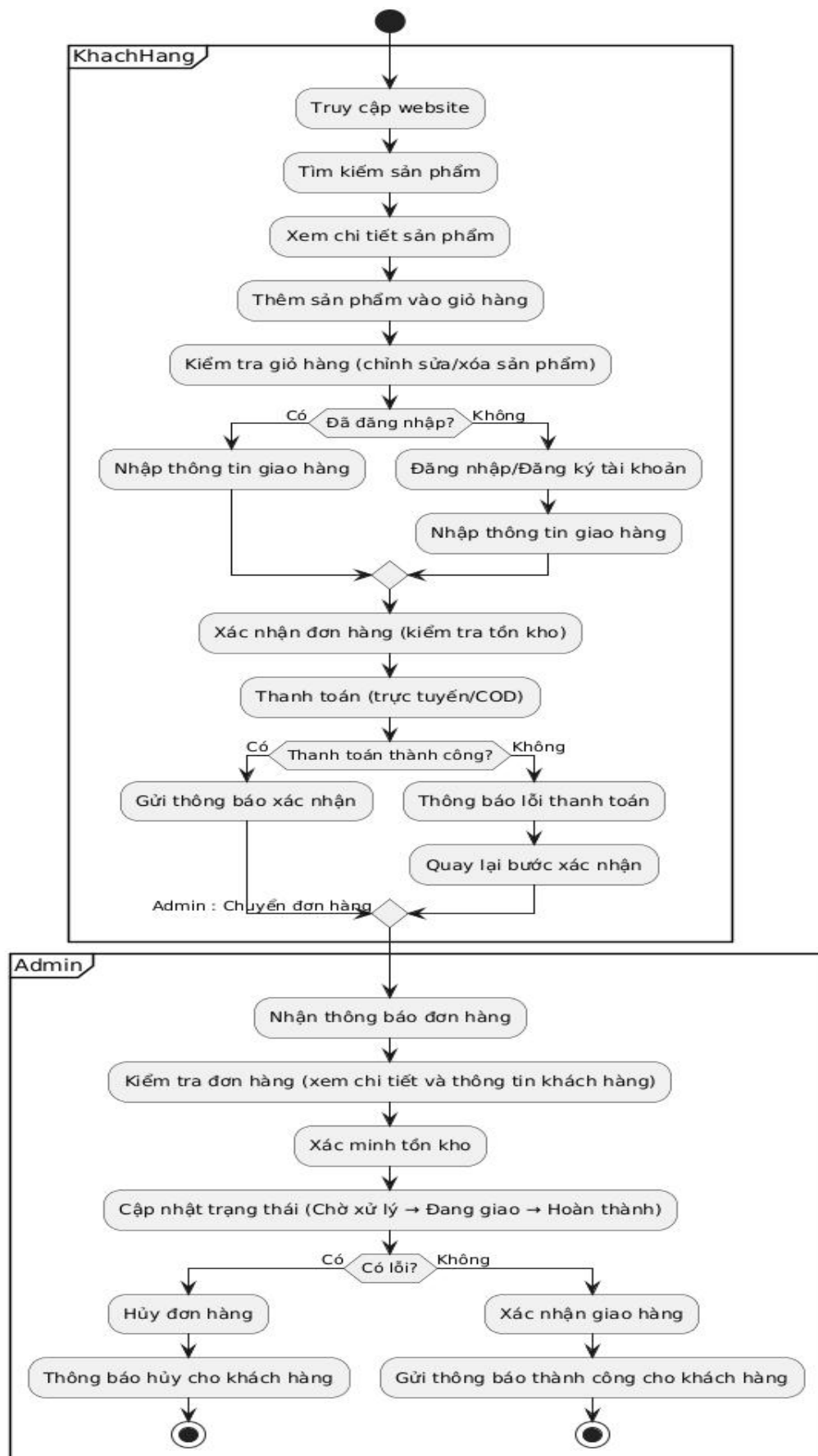
- Quy trình hoạt động của Đặt Hàng.

Luồng	Bước	Quy trình đặt xe
	1	Bắt đầu: Khách hàng truy cập website.
	2	Tìm kiếm sản phẩm: Khách hàng nhập từ khóa hoặc chọn danh mục để tìm sản phẩm
	3	Xem chi tiết sản phẩm: Khách hàng chọn sản phẩm để xem thông tin chi tiết
	4	Thêm vào giỏ hàng: Khách hàng thêm sản phẩm

Khách Hàng		vào giỏ hàng và điều chỉnh số lượng
	5	Kiểm tra giỏ hàng: Hệ thống hiển thị danh sách sản phẩm, tổng giá trị, và cho phép chỉnh sửa hoặc xóa sản phẩm.
	6	Đăng nhập (nếu cần): Nếu chưa đăng nhập, khách hàng thực hiện đăng nhập hoặc đăng ký tài khoản
	7	Nhập thông tin giao hàng: Khách hàng cung cấp địa chỉ giao hàng và chọn phương thức thanh toán
	8	Xác nhận đơn hàng: Hệ thống kiểm tra tồn kho (SanPhamSua) và tạo đơn hàng (DonHang) với chi tiết (ChiTietDonHang).
	9	Thanh toán: Khách hàng hoàn tất thanh toán thông qua cổng thanh toán hoặc chọn thanh toán khi nhận hàng.
	10	Hoàn tất: Hệ thống gửi thông báo xác nhận qua email hoặc giao diện, cập nhật trạng thái đơn hàng thành "Chờ xử lý".
	11	Nhận thông báo đơn hàng: Admin nhận thông báo về đơn hàng mới qua giao diện quản trị
Admin	1	Kiểm tra đơn hàng: Admin xem chi tiết đơn hàng, thông tin khách hàng, và danh sách sản phẩm
	2	Xác minh tồn kho: Admin kiểm tra số lượng tồn kho (SanPhamSua) để đảm bảo đủ hàng giao.  Cập nhật trạng thái: Admin thay đổi trạng thái đơn hàng (từ "Chờ xử lý" sang "Đang giao" hoặc "Hoàn thành" nếu đã giao).
	3	Xử lý vấn đề (nếu có): Nếu có lỗi (hết hàng, địa

		chỉ sai), Admin hủy đơn hàng và thông báo cho khách hàng.
	4	Xác nhận giao hàng: Admin xác nhận giao hàng thành công hoặc xử lý hoàn tiền (nếu hủy).
	5	Gửi thông báo: Admin gửi thông báo kết quả (thành công hoặc hủy) cho khách hàng qua email hoặc hệ thống.
	6	Kết thúc: Quy trình xử lý hoàn tất.

*Bảng 1.0: Activate Diagram*



Hình ảnh 3.2.1. Activate Diagram

### 3.2.4. Sequence Diagram

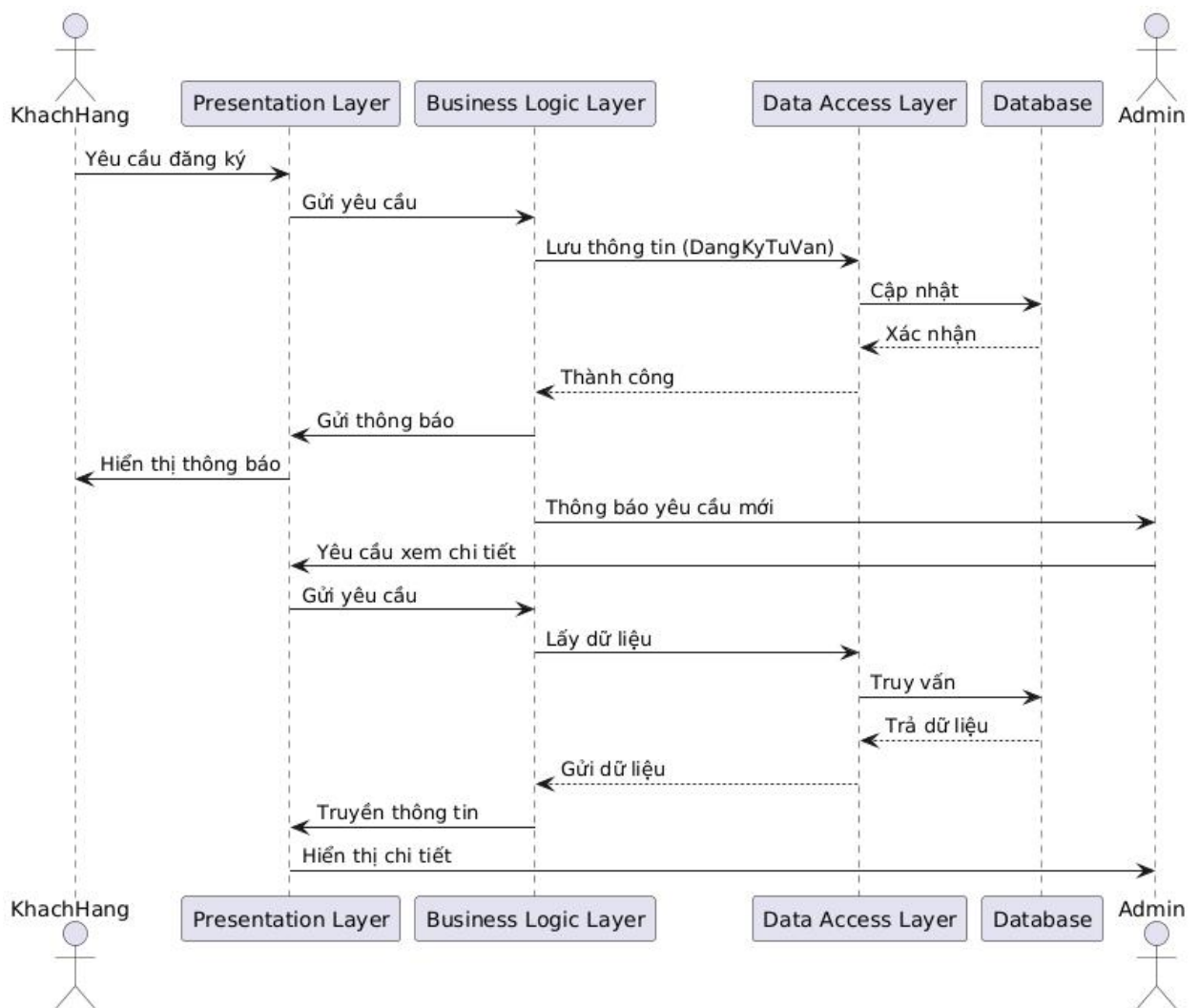
Sơ đồ Sequence mô tả luồng tương tác cơ bản theo kiến trúc nhiều tầng cho quy trình đăng ký tư vấn, từ Khách hàng đến Admin.

#### Các đối tượng:

- **KhachHang:** Khách hàng thực hiện đăng ký tư vấn.
- **Presentation Layer:** Lớp giao diện người dùng hiển thị thông tin.
- **Business Logic Layer:** Lớp xử lý lô-gic nghiệp vụ.
- **Data Access Layer:** Lớp truy cập dữ liệu.
- **Database:** Lưu trữ thông tin.
- **Admin:** Quản trị viên xem chi tiết.

#### Luồng tương tác (Quy trình đăng ký tư vấn):

1. KhachHang gửi yêu cầu đăng ký qua Presentation Layer.
2. Presentation Layer chuyển yêu cầu đến Business Logic Layer.
3. Business Logic Layer yêu cầu Data Access Layer lưu thông tin (DangKyTuVan) vào Database.
4. Data Access Layer lưu và nhận xác nhận từ Database.
5. Business Logic Layer nhận thông báo thành công và gửi đến Presentation Layer.
6. Presentation Layer hiển thị thông báo cho KhachHang.
7. Business Logic Layer thông báo yêu cầu mới cho Admin.
8. Admin gửi yêu cầu xem chi tiết qua Presentation Layer.
9. Presentation Layer chuyển đến Business Logic Layer.
10. Business Logic Layer yêu cầu Data Access Layer lấy dữ liệu.
11. Data Access Layer truy vấn và trả dữ liệu từ Database.
12. Presentation Layer hiển thị chi tiết cho Admin.



Hình ảnh 3.4: Sequence Diagram

### 3.3. Thiết kế cơ sở dữ liệu

Thiết kế cơ sở dữ liệu cung cấp cấu trúc lưu trữ thông tin hiệu quả cho website bán sữa, hỗ trợ các chức năng quản lý sản phẩm, đơn hàng, khách hàng, và đăng ký tư vấn.

#### 1. Bảng KhachHang (Khách hàng)

- **Mô tả:** Lưu thông tin về khách hàng sử dụng website.
- **Thuộc tính:**
  - `khach_hang_id` (Khóa chính, kiểu INT, AUTO\_INCREMENT)
  - `ho_ten` (VARCHAR(100), NOT NULL)
  - `email` (VARCHAR(100), UNIQUE, NOT NULL)
  - `mat_khau` (VARCHAR(255), NOT NULL)



- dia\_chi (TEXT)
- sdt (VARCHAR(15), NOT NULL)
- ngay\_dang\_ky (DATE)
- **Mối quan hệ:**
  - Liên kết 1-n với Bảng DonHang (một khách hàng có nhiều đơn hàng).
  - Liên kết 1-n với Bảng DangKyTuVan (một khách hàng có nhiều yêu cầu tư vấn).

## 2. Bảng SanPhamSua (Sản phẩm sữa)

- **Mô tả:** Lưu thông tin về các sản phẩm sữa được bán.
- **Thuộc tính:**
  - sanpham\_id (Khóa chính, INT, AUTO\_INCREMENT)
  - ten\_san\_pham (VARCHAR(100), NOT NULL)
  - gia (DECIMAL(10,2), NOT NULL)
  - mo\_ta (TEXT)
  - so\_luong\_ton (INT, NOT NULL, DEFAULT 0)
  - loai\_sua\_id (Khóa ngoại, INT, NOT NULL)
- **Mối quan hệ:**
  - Liên kết n-1 với Bảng LoaiSua (một sản phẩm thuộc một loại sữa).
  - Liên kết 1-n với Bảng ChiTietDonHang (một sản phẩm có thể xuất hiện trong nhiều chi tiết đơn hàng).

## 3. Bảng LoaiSua (Loại sữa)

- **Mô tả:** Phân loại các loại sữa (ví dụ: sữa bột, sữa tươi).
- **Thuộc tính:**
  - loai\_sua\_id (Khóa chính, INT, AUTO\_INCREMENT)
  - ten\_loai (VARCHAR(50), NOT NULL, UNIQUE)
- **Mối quan hệ:**

- Liên kết 1-n với Bảng SanPhamSua (một loại sữa có nhiều sản phẩm).

#### 4. Bảng DonHang (Đơn hàng)

- **Mô tả:** Lưu thông tin về các đơn hàng được đặt.
- **Thuộc tính:**
  - don\_hang\_id (Khóa chính, INT, AUTO\_INCREMENT)
  - khach\_hang\_id (Khóa ngoại, INT, NOT NULL)
  - tong\_tien (DECIMAL(10,2), NOT NULL)
  - trang\_thai (VARCHAR(50), NOT NULL, DEFAULT 'Chờ xử lý')
  - ngay\_dat (DATE, NOT NULL)
- **Mối quan hệ:**
  - Liên kết n-1 với Bảng KhachHang (một đơn hàng thuộc một khách hàng).
  - Liên kết 1-n với Bảng ChiTietDonHang (một đơn hàng có nhiều chi tiết).

#### 5. Bảng ChiTietDonHang (Chi tiết đơn hàng)

- **Mô tả:** Lưu thông tin chi tiết của từng sản phẩm trong đơn hàng.
- **Thuộc tính:**
  - chi\_tiet\_id (Khóa chính, INT, AUTO\_INCREMENT)
  - don\_hang\_id (Khóa ngoại, INT, NOT NULL)
  - sanpham\_id (Khóa ngoại, INT, NOT NULL)
  - so\_luong (INT, NOT NULL)
  - don\_gia (DECIMAL(10,2), NOT NULL)
- **Mối quan hệ:**
  - Liên kết n-1 với Bảng DonHang (một chi tiết thuộc một đơn hàng).
  - Liên kết n-1 với Bảng SanPhamSua (một chi tiết liên kết với một sản phẩm).

#### 6. Bảng DangKyTuVan (Đăng ký tư vấn)

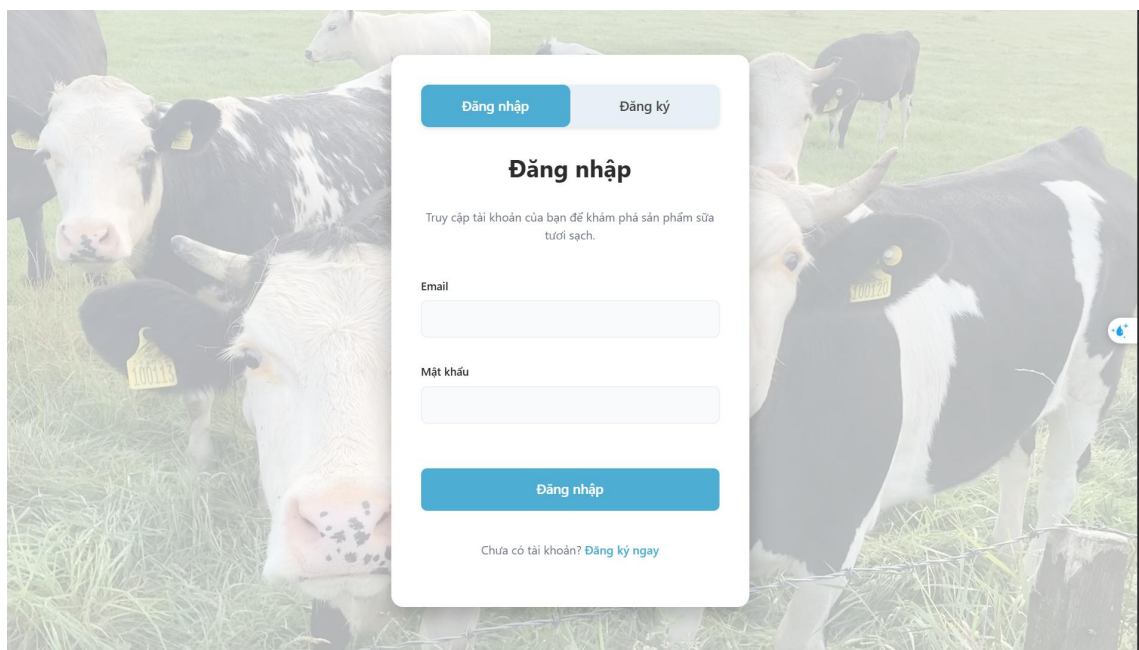
- **Mô tả:** Lưu thông tin yêu cầu tư vấn từ khách hàng.
- **Thuộc tính:**
  - `dang_ky_id` (Khóa chính, INT, AUTO\_INCREMENT)
  - `khach_hang_id` (Khóa ngoại, INT, NOT NULL)
  - `noi_dung` (TEXT, NOT NULL)
  - `ngay_dang_ky` (DATE, NOT NULL)
  - `trang_thai` (VARCHAR(50), NOT NULL, DEFAULT 'Chưa xử lý')
- **Mối quan hệ:**
  - Liên kết n-1 với Bảng KháchHang (một yêu cầu thuộc một khách hàng).

#### 7. Bảng NguoiDung (Người dùng)

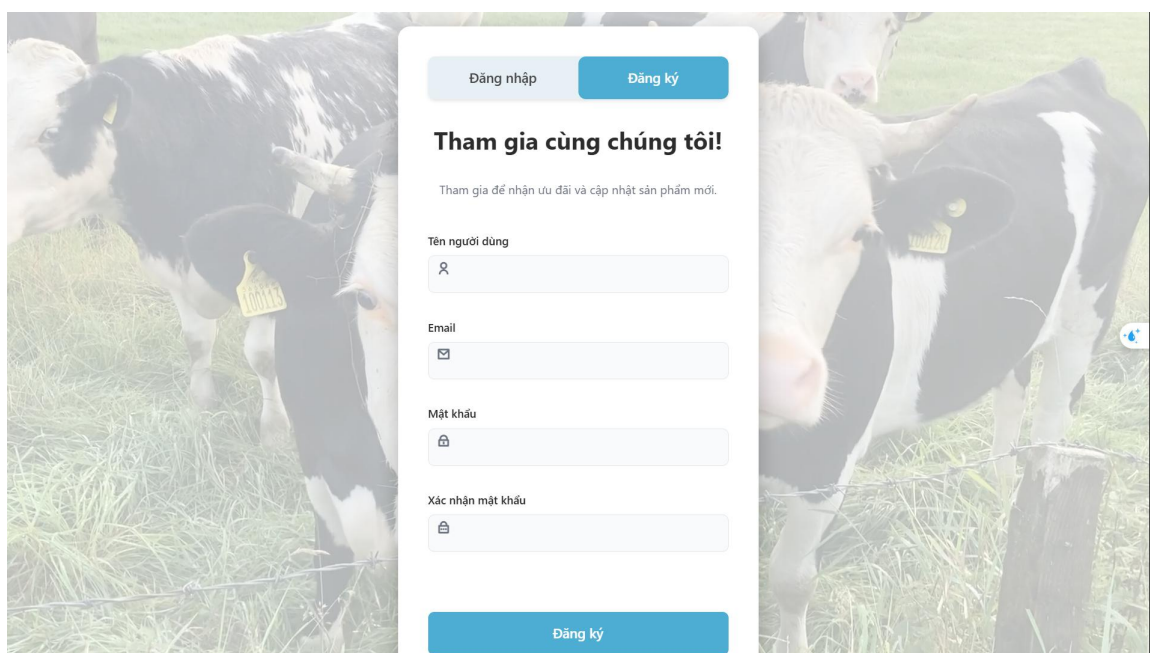
- **Mô tả:** Lưu thông tin tài khoản quản trị viên hoặc khách hàng.
- **Thuộc tính:**
  - `nguai_dung_id` (Khóa chính, INT, AUTO\_INCREMENT)
  - `username` (VARCHAR(50), UNIQUE, NOT NULL)
  - `mat_khau` (VARCHAR(255), NOT NULL)
  - `vai_tro` (VARCHAR(50), NOT NULL, DEFAULT 'KhachHang')
- **Mối quan hệ:**
  - Liên kết 1-0..1 với Bảng KháchHang (một người dùng có thể là khách hàng).

### 3.4. Thiết kế giao diện

#### 3.4.1. Giao diện Đăng nhập/ Đăng Ký

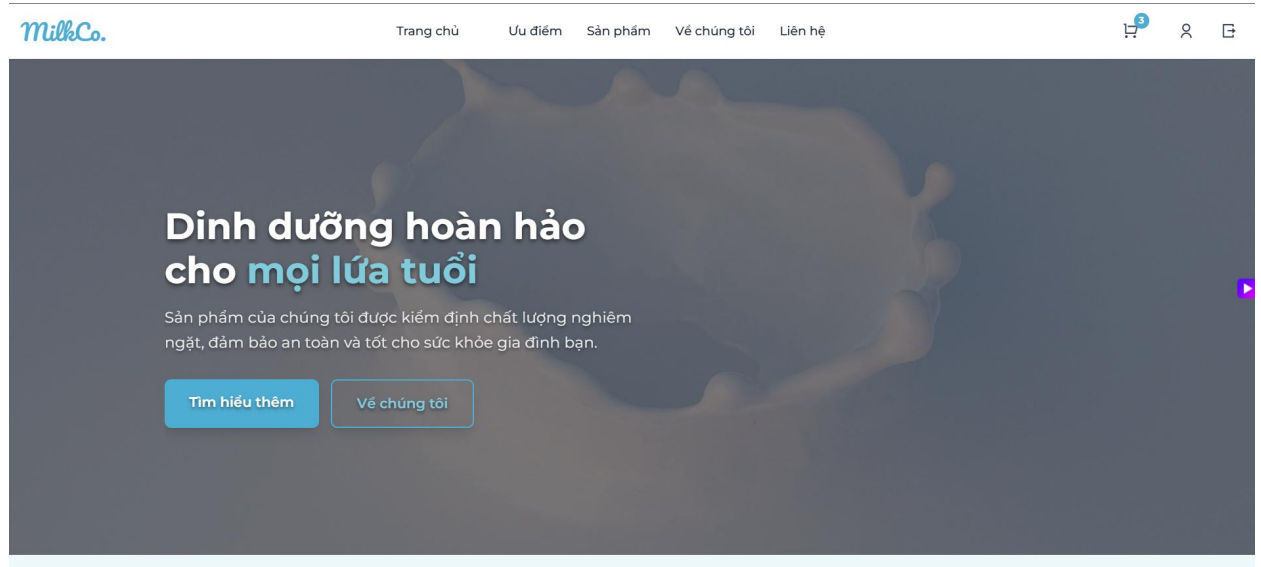


*Hình ảnh 4.0: Đăng nhập*

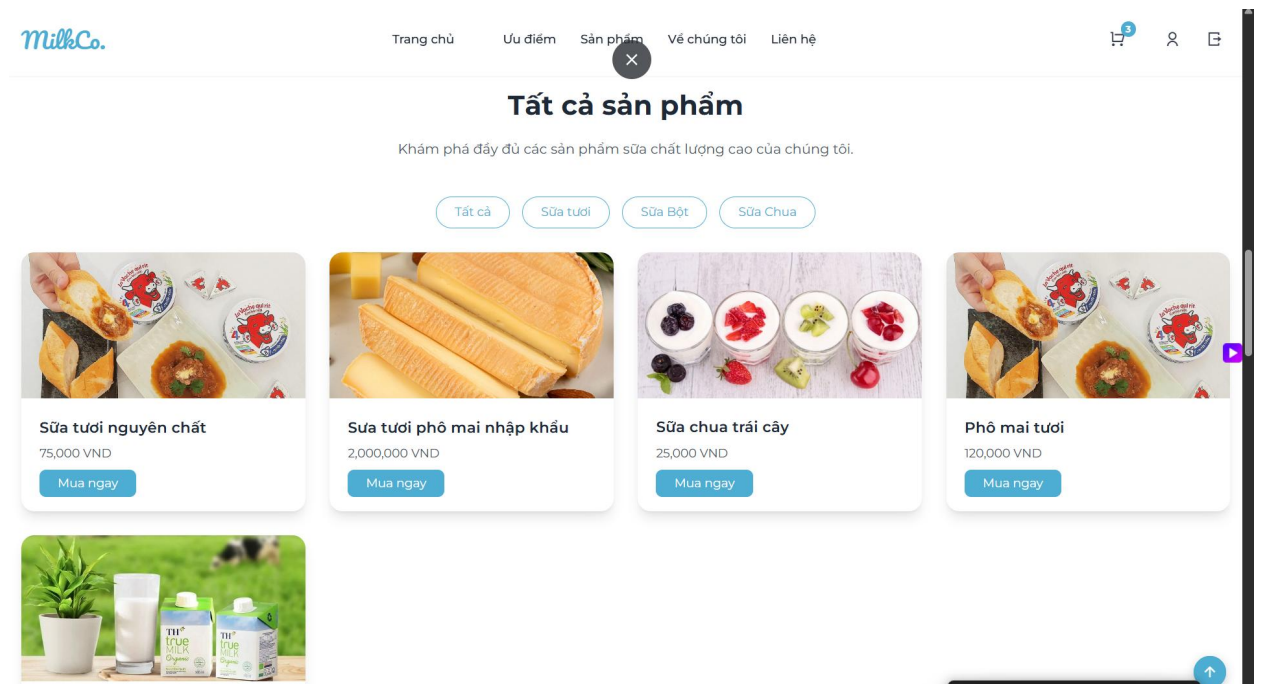


*Hình ảnh 4.1: Đăng nhập*

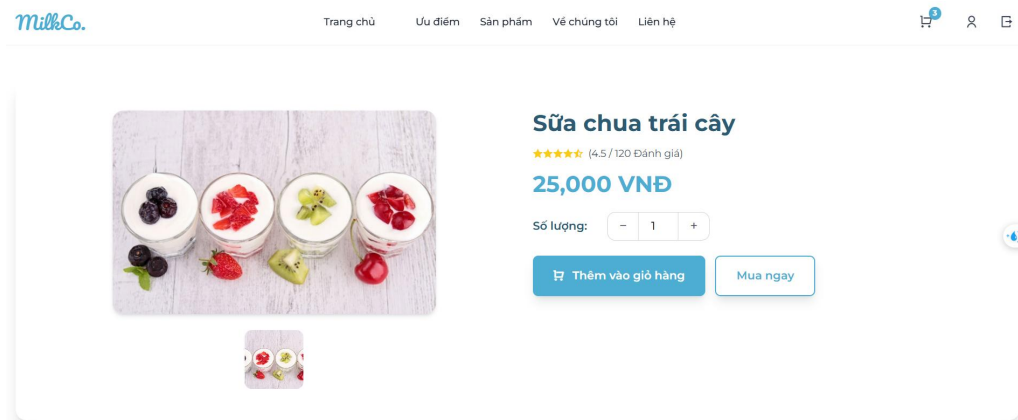
### 3.4.2. Giao diện khách hàng



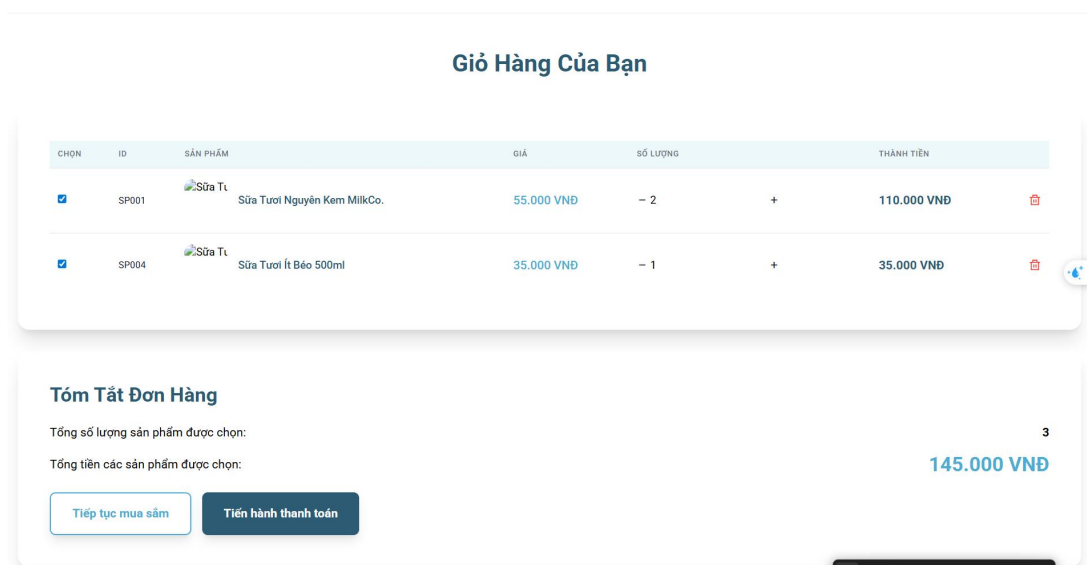
Hình ảnh 4.2: Trang chủ



Hình ảnh 4.3: Xem tất cả sản phẩm



Hình ảnh 4.4: Chi tiết sản phẩm



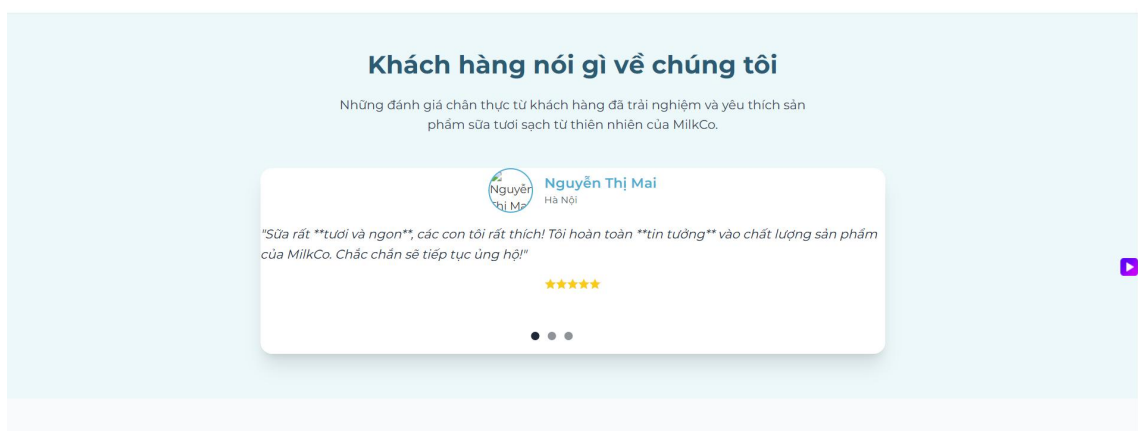
Hình ảnh 4.5: Giỏ Hàng



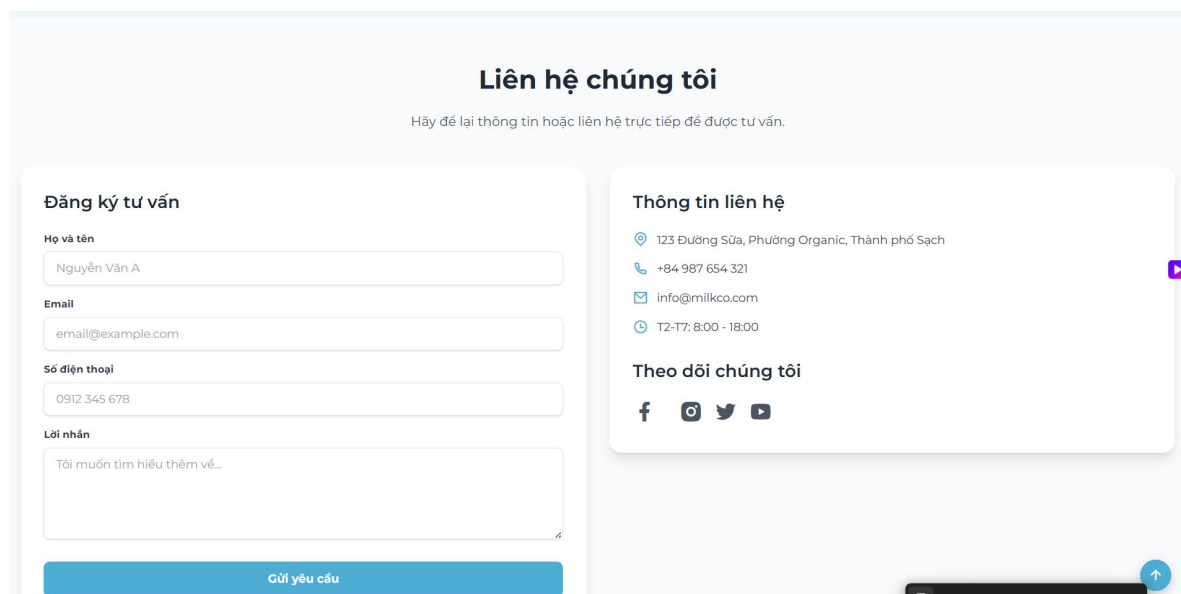
Hình ảnh 4.6: Hồ sơ khách hàng



Hình ảnh 4.7: Giới thiệu Trang web



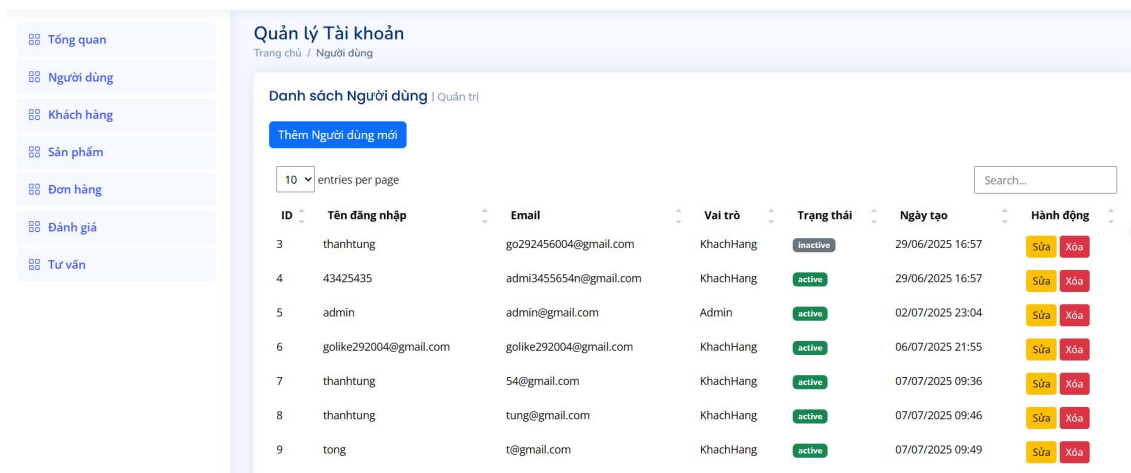
Hình ảnh 4.8: Đánh giá nổi bật khách hàng



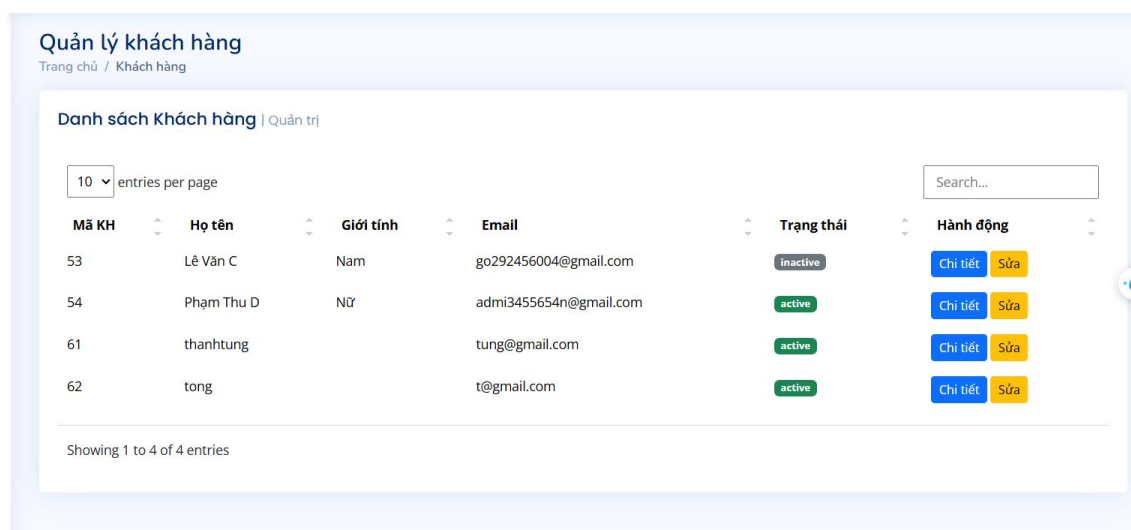
Hình ảnh 4.9: Đăng ký tư vấn



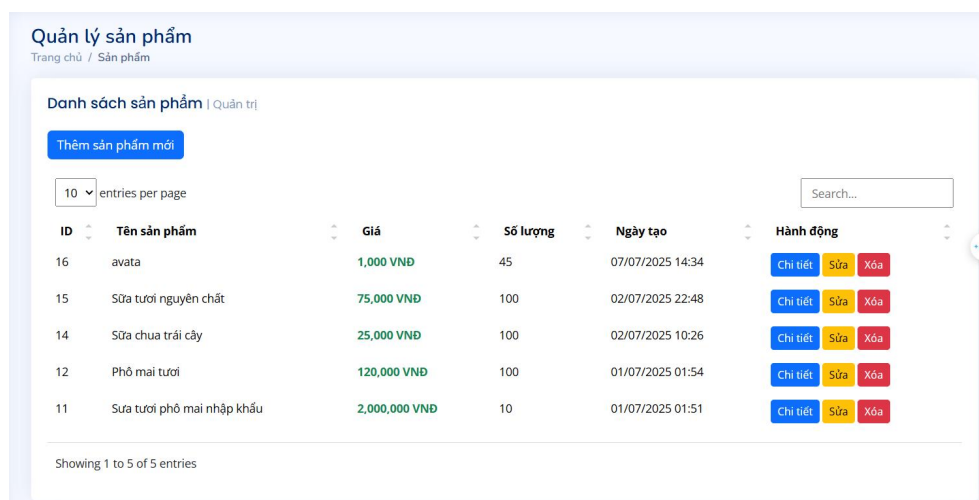
### 3.4.3. Giao diện Admin



Hình ảnh 5.0: Quản lý tài khoản



Hình ảnh 5.1: Quản lý khách hàng



Hình ảnh 5.2: Quản lý sản phẩm



Chi tiết Sản phẩm

[Trang chủ Admin](#) / [Sản phẩm](#) / Chi tiết

Sửa

← Quay lại

avata

Mã Sữa:

16

Mã Loại:

Sữa Tươi

Giá:

1,000 VNĐ

Số lượng:

45

Trạng thái:

active

Ngày tạo:

07/07/2025 14:34

Mô tả:

sfdsgdf hhj

Hình ảnh 5.3: Chi tiết sản phẩm

Thêm sản phẩm mới

Thêm sản phẩm thành công.

Tên sản phẩm

Mã loại

Sữa bột

Giá

Số lượng

Mô tả

Hình ảnh 5.4: Thêm sản phẩm mới

Cập nhật sản phẩm

[Trang chủ](#) / [Sản phẩm](#) / [Cập nhật](#)

Tên sữa

Sữa tươi nguyên chất

Giá

75000.00

Mô tả

Thông số sản phẩm:

Dung tích: 1 Lít

Thành phần: 100% Sữa bò tươi nguyên chất

Bảo quản: Giữ lạnh ở 2-4°C

Số lượng

100

Ngày tạo

07/02/2025 10:48:20, 730 PM

Trạng thái

Hoạt động

Ảnh hiện tại:

Chọn ảnh mới (nếu muốn thay)

Choose Files | No file chosen

Hình ảnh 5.5: Cập nhật sản phẩm

Chức năng thêm sửa xóa của các chức năng khác của Admin tương tự như chức năng thêm sửa xóa của sản phẩm.

### 3.5. Kiểm thử

#### 3.5.1. Kiểm thử chức năng

Sau khi quá trình làm thử nghiệm và test thử bên dưới là bảng đã được kiểm thử và đưa ra kết luận.

STT	Chức năng kiểm thử	Kết quả đạt được
1	Đăng ký	Cho phép tạo tài khoản mới, lưu thông tin khách hàng vào CSDL (bảng KháchHang).
2	Đăng nhập	Xác thực tài khoản và phân quyền, chuyển hướng đúng giao diện (khách hàng/quản trị).
3	Đăng xuất	Hủy session và quay lại trang đăng nhập.
4	Xem danh sách sản phẩm	Hiển thị tất cả sản phẩm sữa có trạng thái "Còn hàng" để khách hàng lựa chọn.
5	Thêm sản phẩm vào giỏ hàng	Gửi thông tin sản phẩm, cập nhật số lượng, lưu vào bảng GioHang.
6	Xem lại giỏ hàng	Hiển thị danh sách sản phẩm trong giỏ, cho phép chỉnh sửa hoặc xóa.
7	Tạo sản phẩm mới	Thêm mới sản phẩm sữa với thông tin (tên, giá, mô tả) vào bảng SanPhamSua.
8	Sửa sản phẩm	Cập nhật thông tin sản phẩm (giá, mô tả) trong bảng SanPhamSua.

*Bảng 2.0: Kiểm thử chức năng*

### 3.5.2. Kiểm thử giao diện

Mục tiêu của kiểm thử giao diện: Đảm bảo giao diện website bán sữa thân thiện, dễ sử dụng, và đáp ứng tốt trên các thiết bị khác nhau (máy tính, điện thoại, máy tính bảng) theo đặc tả thiết kế.

- Phương pháp:

- Kiểm thử thủ công trên các trình duyệt phổ biến (Chrome, Firefox, Safari) và thiết bị khác nhau.
- Sử dụng công cụ như BrowserStack để kiểm tra tính tương thích và responsive design.

- Trường hợp kiểm thử:

- Giao diện trang chủ hiển thị đúng trên màn hình 1920x1080 (máy tính) và 360x640 (điện thoại).
- Nút "Thêm vào giỏ hàng" và "Đăng ký tư vấn" dễ nhìn, hoạt động chính xác trên mọi thiết bị.
- Trang danh sách sản phẩm không bị lỗi chồng chéo hoặc sai lệch bố cục khi thay đổi kích thước màn hình.
- Form đăng nhập và đăng ký hiển thị đầy đủ trường (tên, email, mật khẩu) và điều hướng đúng khi nhấn "Đăng nhập".
- Giao diện quản trị (dành cho Admin) hiển thị rõ danh sách sản phẩm và đơn hàng trên màn hình 1366x768.

=> Kết quả mong đợi: Giao diện mượt mà, không có lỗi hiển thị, tương thích với mọi thiết bị, và cung cấp trải nghiệm người dùng tốt.

### 3.5.3. Kiểm thử hiệu năng

Giai đoạn kiểm thử hiệu năng tập trung vào việc đánh giá khả năng phản hồi, tốc độ xử lý và hiệu quả tải trang của hệ thống trong điều kiện hoạt động thực tế (Local). Các thử nghiệm được tiến hành với các chức năng quan trọng đặc thù của một website thương mại điện tử trong ngành sữa:

- Thời gian phản hồi: Hệ thống được kiểm tra với các thao tác như đăng nhập người dùng, duyệt danh mục sản phẩm, thêm sữa vào giỏ hàng, và tiến hành thanh toán. Tất cả các thao tác đều phản hồi dưới 2 giây, giúp đảm bảo trải nghiệm mua sắm mượt mà và nhanh chóng.

- Xử lý truy vấn dữ liệu: Các truy vấn LINQ sử dụng Include và ThenInclude để lấy thông tin sản phẩm, chi tiết loại sữa, thương hiệu, đánh giá và trạng thái đơn hàng vận hành ổn định, không phát sinh độ trễ hay xung đột dữ liệu. Điều này cho thấy hệ thống đã được tổ chức dữ liệu hợp lý và tối ưu hóa cơ sở dữ liệu tốt.

- Tải và xử lý file: Việc người quản trị tải lên hình ảnh sản phẩm sữa (dưới 5MB) diễn ra trơn tru, đảm bảo hình ảnh được cập nhật nhanh vào hệ thống và hiển thị tức thời trên giao diện người dùng.

- Hiển thị danh sách sản phẩm lớn: Khi số lượng sản phẩm, đơn hàng và người dùng tăng cao, trang danh mục và quản lý vẫn tải ổn định. Hệ thống chưa cần phân trang tại thời điểm kiểm thử nhưng sẵn sàng mở rộng tính năng này khi quy mô dữ liệu lớn hơn.

=> **Tóm lại** : Qua kết quả kiểm thử, hệ thống đã chứng minh khả năng hoạt động hiệu quả, phản hồi nhanh, và có tiềm năng mở rộng phù hợp với nhu cầu thực tế của một website bán sữa chuyên nghiệp. Kiến trúc hệ thống được thiết kế linh hoạt, hỗ trợ tốt cho cả trải nghiệm người dùng lẫn công tác quản trị sản phẩm và đơn hàng.

#### 3.5.4. Kiểm thử bảo mật

Trong môi trường thương mại điện tử, bảo mật là yếu tố then chốt quyết định đến lòng tin của khách hàng cũng như sự bền vững của hệ thống. Đối với website bán sữa, nhóm phát triển đã tiến hành các kiểm thử bảo mật với các mục tiêu cụ thể nhằm phát hiện và ngăn chặn các lỗ hổng tiềm ẩn:

- + Kiểm soát quyền truy cập: Hệ thống được kiểm thử để đảm bảo các vai trò như Khách hàng và Quản trị viên chỉ truy cập được vào đúng phạm vi chức năng cho phép. Các đường dẫn nhạy cảm được bảo vệ chặt chẽ và không thể truy cập trực tiếp thông qua URL nếu không có quyền.

- + Thử nghiệm đăng nhập sai/đúng: Hệ thống xử lý hiệu quả các tình huống đăng nhập sai thông tin nhiều lần liên tiếp. Cơ chế khóa tài khoản tạm thời và cảnh báo đăng nhập được kích hoạt đúng quy trình, ngăn chặn nguy cơ tấn công brute-force.

- + Kiểm thử lỗ hổng đầu vào (SQL Injection/XSS): Các trường nhập liệu như tìm kiếm, đăng ký, phản hồi sản phẩm được kiểm tra kỹ để ngăn chặn mã độc. Hệ thống thực hiện kiểm tra và lọc dữ liệu đầu vào trước khi xử lý, loại bỏ rủi ro tấn công chèn mã (SQL Injection) hay JavaScript độc hại (XSS).

+ Mã hóa và bảo mật dữ liệu: Dữ liệu nhạy cảm như mật khẩu người dùng được mã hóa bằng thuật toán băm an toàn (ví dụ: SHA-256) trước khi lưu trữ. Các thông tin giao dịch được truyền tải qua kết nối HTTPS để đảm bảo tính bảo mật trong quá trình truyền nhận.

+ Kiểm tra lỗi logic bảo mật: Các chức năng như đặt hàng, quản lý đơn và cập nhật sản phẩm đều được kiểm tra logic để đảm bảo không thể bị lợi dụng hoặc khai thác sai quy trình nhằm gây tổn hại cho hệ thống.

Qua các bài kiểm thử bảo mật, hệ thống trang web bán sữa đã đạt được mức độ an toàn đáng tin cậy, bảo vệ tốt thông tin người dùng và ngăn ngừa các rủi ro thường gặp trong môi trường web. Đây là nền tảng vững chắc giúp hệ thống sẵn sàng triển khai thực tế và mở rộng quy mô kinh doanh lâu dài.

## KẾT LUẬN

Hệ thống website bán sữa đã được xây dựng và triển khai thành công với các chức năng cơ bản như: quản lý sản phẩm, giỏ hàng, đơn hàng và quản lý người dùng. Giao diện thân thiện, tốc độ phản hồi tốt và đảm bảo độ ổn định trong môi trường local.

Hiện hệ thống vẫn đang chạy trong môi trường cục bộ (local), chưa triển khai lên môi trường thực tế hoặc tích hợp các giải pháp mạng. Một số chức năng nâng cao như thanh toán trực tuyến, phân trang, thống kê doanh thu theo thời gian và hỗ trợ thiết bị di động chưa được thực hiện.

Trong giai đoạn tiếp theo, hệ thống sẽ được phát triển thêm các tính năng nâng cao như tích hợp chatbot hỗ trợ khách hàng, triển khai hệ thống trên hosting/Cloud để hoạt động online, mở rộng chức năng thanh toán trực tuyến, phân tích dữ liệu bán hàng, và tối ưu trải nghiệm trên thiết bị di động. Đây sẽ là bước chuyển quan trọng giúp hệ thống đáp ứng tốt hơn nhu cầu thực tế.

## TÀI LIỆU THAM KHẢO

- [1] Alan Beaulieu, “Learning SQL: Generate, Manipulate, and Retrieve Data”, O’Reilly Media, 2020
- [2] Adam Freeman, “Pro ASP.NET Core 7”, Manning, 2023.
- [3] Mark J. Price, “C# 13 and .NET 9 Modern Cross-Platform Development Fundamentals: Start building websites and services with ASP.NET Core 9, Blazor, and EF Core 9”, Packt Publishing, 2024.
- [4] RB Whitaker, “The C# Player’s Guide”, RB Whitaker, 2022.
- [5] Đặng Văn Hiệp, “Đồ án xây dựng website quản lý bán sữa,” Trường Đại học Dân lập Hải Phòng, 2020. [Online]. Available: <https://thuvientailieu.org/do-an-xay-dung-website-quan-ly-ban-sua-11221/>
- [6] Bizfly, “Thiết kế website bán sữa chuyên nghiệp – Các chức năng cần có,” Bizfly.vn, 2023. [Online]. Available: <https://bizfly.vn/techblog/thiet-ke-website-ban-sua.html>
- [7] Webnganh.vn, “Mẫu website bán sữa – Dự án thực tế,” Webnganh.vn. [Online]. Available: <https://webnganh.vn/du-an/mau-website-ban-sua>