

TRƯỜNG ĐẠI HỌC YERSIN ĐÀ LẠT
KHOA CÔNG NGHỆ THÔNG TIN



YERSIN UNIVERSITY

BÁO CÁO MÔN HỌC
LẬP TRÌNH WEB

XÂY DỰNG HỆ THỐNG
API QUẢN LÝ ĐỒ UỐNG
BẰNG ASP.NET CORE

GVHD : ThS. Nguyễn Đức Tấn

SVTH : Nguyễn Anh Bảo-230100071

Nguyễn Thành Nhân-12201100

Nguyễn Xuân Sang -2301010051

Khóa học : 2023 - 2026

Đà Lạt, tháng 10 – 2025.

PHẦN NHẬN XÉT CỦA GIẢNG VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Điểm:

Ngày ... tháng ... năm ...

Ký và ghi rõ họ tên

MỤC LỤC

LỜI NÓI ĐẦU	1
CHƯƠNG 1: TÌM HIỂU VỀ LẬP TRÌNH WEB MVC	2
1.1 .Khái niệm về API:	2
1.1.1. Các thành phần chính của API:	2
1.2. Đặc điểm của API:	3
1.3. Nguyên lý hoạt động của API:	4
1.3.1. Chi tiết các bước trong luồng hoạt động API:	4
1.4. Công nghệ triển khai ASP.NET Core API:	5
1.4.1. Tổng quan về ASP.NET Core API:	5
1.4.2. Khái niệm Framework:	5
1.4.3. EF Core Framework (Entity Framework Core):	6
1.4.4. Authentication (Xác thực) & Authorization (Phân quyền):	6
1.4.5. Triển khai ASP.NET Core MVC để khai thác API (Consuming API):	7
CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG WEB API	9
2.1. Phát biểu bài toán ứng dụng:	9
2.2. Phân tích yêu cầu của hệ thống API:	9
2.2.1. Yêu cầu phi chức năng:	9
2.2.2. Yêu cầu chức năng:	10
2.2.3. Biểu đồ Use case:	12
2.3. Thiết kế cơ sở dữ liệu:	21
2.3.1. Sơ đồ Database:	21
2.3.2. Sơ đồ ERD:	22
2.3.4. Bảng mô tả dữ liệu:	24
2.4. Thiết kế giao diện Wireframe:	31
2.4.1. Wireframe Trang chủ:	31
2.4.2. Wireframe DashBoard Title:	32
2.4.3. Wireframe Đăng nhập:	33
2.4.4. Wireframe Đăng ký:	34
2.5. Thiết kế các thành phần API và Consumin API:	35
2.5.1. API:	35
2.5.2. Consuming API (MVC):	38
2.6. Triển khai và cài đặt:	42
CHƯƠNG 3: KẾT QUẢ CHƯƠNG TRÌNH	44

3.1.Giao diện API:	44
3.1.1.Admin:	44
3.1.2 Auth và Beverages	44
3.1.3 Categories , Order và Reviews	45
3.14 Upload à Voucher	45
3.1.5 Wishlist	46
3.2 Giao diện Consuming API (MVC)	47
3.2.1 Trang Chủ	47
3.2.2 Trang cửa hàng đồ uống	48
3.2.3 Trang giỏ hàng	49
3.2.4 Thanh toán Đơn hàng	49
3.2.5. Đăng ký:	50
3.2.6 Đăng nhập:	50
3.2.7.Tài khoản:	51
3.2.8.Đặt hàng thành công:	52
3.3.Giao diện Admin (Quản trị viên):	53
3.3.1Trang chủ Quản trị:	53
3.3.2.Quản lý Sản phẩm:	54
3.3.3 Thêm Sản phẩm:	55
3.3.4.Chỉnh sửa Sản phẩm:	56
3.3.5. Quản lý Đơn hàng:	57
3.3.6. Chi tiết Đơn hàng:	57
3.3.7. Quản lý Người dùng:	58
3.3.8.Thêm người dùng	59
3.3.9.Chỉnh sửa Người dùng:	59
3.3.10.Quản lý Brands:	60
3.3.11.Thêm Brands	60
KẾT LUẬN	61
TÀI LIỆU THAM KHẢO	63

DANH MỤC HÌNH ẢNH

Hình 1 Khái niệm về API	2
Hình 2: Biểu đồ Use case tổng quát	12
Hình 3: Biểu đồ Use case người dùng	13
Hình 4: Biểu đồ Use case Admin	18
Hình 5: Sơ đồ Database	21
Hình 6: Sơ đồ ERD	22
Hình 7: Wireframe Trang chủ	31
Hình 8: Wireframe DashBoard Title	32
Hình 9: Wireframe Đăng nhập	33
Hình 10: Wireframe Đăng ký	34
Hình 11: Admin	44
Hình 12: Auth và Beverages	44
Hình 13: Categories , Order và Reviews	45
Hình 14: Upload và Voucher	45
Hình 15 : Wishlist	46
Hình 16: Trang chủ	47
Hình 17: Trang cửa hàng đồ uống	48
Hình 18: Trang giỏ hàng	49
Hình 19: Trang thanh toán đơn hàng	49
Hình 20: Đăng ký	50
Hình 21: Đăng nhập	50
Hình 22: Tài khoản	51
Hình 23: Đặt hàng thành công	52
Hình 24: Trang chủ Quản trị	53
Hình 25: Quản lý Sản phẩm	54
Hình 26: Thêm Sản phẩm	55
Hình 27: Chỉnh sửa Sản phẩm	56
Hình 28: Quản lý Đơn hàng	57
Hình 29: Chi tiết Đơn hàng	57
Hình 30: Quản lý Người dùng	58
Hình 31: Thêm Người dùng	59
Hình 32: Chỉnh sửa Người dùng	59
Hình 33: Quản lý Brands.	60
Hình 34: Thêm Brands	60

DANH MỤC BẢNG BIỂU

Bảng 1: Bảng yêu cầu chức năng cho Admin (Quản trị viên)	10
Bảng 2: Bảng yêu cầu chức năng cho Khách hàng	11
Bảng 3: Bảng đặc tả Use case Đăng ký tài khoản	14
Bảng 4: Bảng đặc tả Use case Đăng nhập	14
Bảng 5: Bảng đặc tả Use case Quản lý giỏ hàng	15
Bảng 6: Bảng đặc tả Use case Quản lý tài khoản cá nhân	15
Bảng 7: Bảng đặc tả Use case Quản lý đơn hàng	15
Bảng 8: Bảng đặc tả Use case Đánh giá	16
Bảng 9: Bảng đặc tả Use case Tìm kiếm	16
Bảng 10: Bảng đặc tả Use case Đăng xuất	16
Bảng 11: Bảng đặc tả Use case Sản phẩm	17
Bảng 12: Bảng đặc tả Use case Chi tiết sản phẩm	17
Bảng 13: Bảng đặc tả Use case Liên hệ	17
Bảng 14: Bảng đặc tả Use case Đăng nhập (Admin)	19
Bảng 15: Bảng đặc tả Use case Đăng xuất (Admin)	19
Bảng 16: Bảng đặc tả Use case Quản lý mã giảm giá	19
Bảng 17: Bảng đặc tả Use case Quản lý danh mục	20
Bảng 18: Bảng đặc tả Use case Quản lý sản phẩm	20
Bảng 19: Bảng đặc tả Use case Quản lý đơn hàng	20
Bảng 20: Bảng đặc tả Use case Quản lý tài khoản	21
Bảng 23: Bảng mô tả mối quan hệ giữa các thực thể	23
Bảng 24: Bảng Vouchers	24
Bảng 26: Bảng Wishlists	25
Bảng 27: Bảng OrderItems	25
Bảng 28: Bảng Orders	26
Bảng 29: Bảng Beverages	27
Bảng 30: Bảng Reviews	28
Bảng 31: Bảng Users	29
Bảng 31: Bảng UserPoints	30
Bảng 32: Bảng Model User	35
Bảng 33: Bảng Model Wishlist	35
Bảng 34: Bảng Model Review	36

Bảng 36: Bảng Controller UploadController	36
Bảng 37: Bảng Controller WishlistController	37
Bảng 38: Bảng Controller AuthController	37
Bảng 39: Bảng Model Beverage	38
Bảng 40: Bảng Model Category	39
Bảng 41: Bảng Model Review	39
Bảng 42: Bảng Controller ShopController	40
Bảng 43: Bảng Controller AccountController	41

DANH MỤC TỪ NGỮ VIẾT TẮT

STT	Tên viết tắt	Tên đầy đủ	Dịch ra tiếng Việt
1	MVC	Model – View - Controller	
2	SEO	Search Engine Optimization	Tối ưu hóa công cụ tìm kiếm
3	CSDL	Cơ sở dữ liệu	
4	ERD	Entity-Relationship Diagram	Sơ đồ mối quan hệ thực thể
5	JSON	JavaScript Object Notation.	Ký hiệu đối tượng JavaScript
6	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
7	ASP.NET	Active Server Pages .NET	Trang chủ động .NET
8	AJAX	Asynchronous JavaScript and XML	JavaScript và XML không đồng bộ
9	CSS	Cascading Style Sheets	Tập tin định kiểu theo tầng
10	HTTP	Hypertext Transfer Protocol	Giao thức Truyền tải Siêu Văn Bản
11	URL	Uniform Resource Locator	Hệ thống định vị tài nguyên thống nhất
12	API	Application Programming Interface	Giao diện lập trình ứng dụng

LỜI NÓI ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0, các ứng dụng hiện đại đòi hỏi sự kết nối và trao đổi dữ liệu một cách linh hoạt, hiệu quả, nhanh chóng. Giao diện lập trình ứng dụng (API) đã trở thành một thành phần không thể thiếu, đóng vai trò là cầu nối vững chắc cho các dịch vụ và nền tảng khác nhau có thể giao tiếp với nhau.

Báo cáo này trình bày quá trình thiết kế và xây dựng một hệ thống API quản lý đồ uống. Hệ thống được phát triển nhằm cung cấp các phương thức để quản lý sản phẩm, xử lý đơn hàng và phân quyền người dùng một cách bảo mật và hiệu quả, nhanh chóng. Các chức năng chính bao gồm: quản lý danh mục đồ uống, xử lý đơn hàng, mua đồ uống một cách dễ dàng.

Trong suốt quá trình thực hiện, em đã vận dụng các kiến thức về thiết kế RESTful API, kiến trúc phần mềm cùng với nền tảng ASP.NET Core để xây dựng cả frontend lẫn backend. Đồng thời, việc thiết kế cơ sở dữ liệu một cách hợp lý là yếu tố quan trọng để đảm bảo hiệu năng và tính ổn định cho hệ thống.

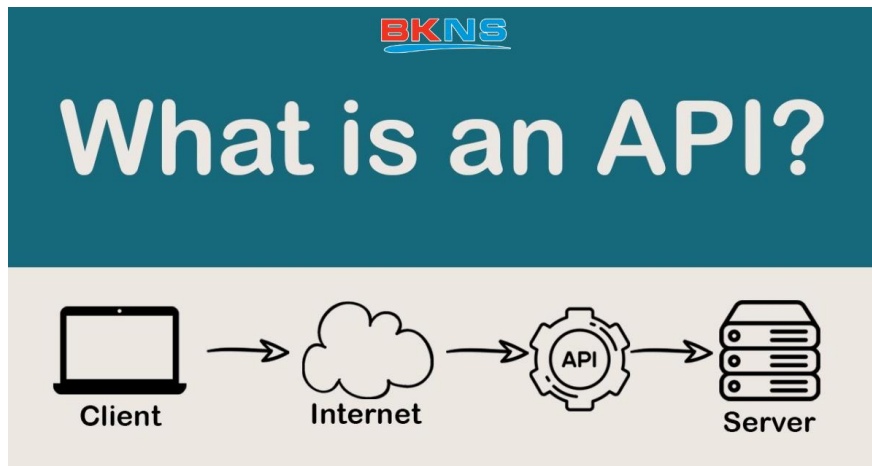
Tuy đã cố gắng hoàn thành tốt đề tài, nhưng do thời gian có hạn và kinh nghiệm còn hạn chế, bài báo cáo khó tránh khỏi những thiếu sót. Em rất mong nhận được những góp ý quý báu để có thể hoàn thiện hơn trong các lần thực hiện sau.

Em xin chân thành cảm ơn thầy Nguyễn Đức Tấn đã tận tình hướng dẫn và hỗ trợ em trong suốt quá trình thực hiện đề tài này.

CHƯƠNG 1: TÌM HIỂU VỀ LẬP TRÌNH WEB MVC

1.1 .Khái niệm về API:

Giao diện lập trình ứng dụng (Application Programming Interface) là một tập hợp các quy tắc, giao thức và công cụ cho phép các thành phần phần mềm khác nhau giao tiếp với nhau. Nó định nghĩa các phương thức (methods) mà một chương trình có thể yêu cầu từ chương trình khác, loại dữ liệu nào cần cung cấp, và định dạng dữ liệu trả về sẽ như thế nào



Hình 1. Khái niệm về API

1.1.1. Các thành phần chính của API:

❖ Cốt lõi của một API bao gồm hai thành phần chính: Request (Yêu cầu) và Response (Phản hồi).

❖ **Đối với Request:** 1 request đúng chuẩn gồm 4 thứ:

- URL (Uniform Resource Locator): Đây là đường dẫn định danh duy nhất cho mỗi yêu cầu, trỏ đến một tài nguyên hoặc một chức năng xử lý logic cụ thể trên máy chủ.
- Method (Phương thức): Đây là "hành động" mà client muốn server thực hiện trên tài nguyên được chỉ định bởi URL. Các phương thức HTTP thông dụng bao gồm:
 - GET: Truy xuất hoặc lấy dữ liệu từ một tài nguyên.
 - HEAD: Tương tự GET, nhưng phản hồi chỉ chứa header, không có body. Dùng để kiểm tra siêu dữ liệu (metadata) của tài nguyên.
 - POST: Gửi dữ liệu đến server để tạo mới một tài nguyên (ví dụ: tạo người dùng mới).
 - PUT: Cập nhật một tài nguyên đã tồn tại bằng cách thay thế toàn bộ dữ liệu của nó.

- PATCH: Cập nhật một phần của tài nguyên đã tồn tại.
- DELETE: Xóa một tài nguyên cụ thể.
- CONNECT: Thiết lập một kết nối mạng hai chiều tới một tài nguyên.
- OPTIONS: Yêu cầu server mô tả các tùy chọn giao tiếp (ví dụ: các method được phép) cho một tài nguyên.

- **Headers:**

Chứa các thông tin bổ sung (metadata) về yêu cầu. Những thông tin này thường không hiển thị cho người dùng cuối nhưng lại rất quan trọng cho việc xử lý. Ví dụ: thông tin xác thực (token), định dạng dữ liệu mong muốn nhận về (Accept), loại nội dung đang gửi đi (Content-Type),...

- **Body** : Là phần chứa dữ liệu chính mà client muốn gửi đến server, chẳng hạn như thông tin trong một biểu mẫu đăng ký hay nội dung một bài viết.

- **Đối với response:**

Sau khi tiếp nhận và xử lý yêu cầu, server sẽ gửi trả lại một Response. Cấu trúc của Response cũng bao gồm 3 phần chính, với Status Code thay thế cho URL và Method:

- Status Code (Mã trạng thái): Con số thông báo kết quả xử lý yêu cầu (ví dụ: 200 OK, 404 Not Found, 500 Internal Server Error).
- Headers: Chứa thông tin bổ sung về chính phản hồi đó.
- Body: Chứa dữ liệu mà client đã yêu cầu.

1.2. Đặc điểm của API:

- **Hỗ trợ đầy đủ kiến trúc RESTful và các thành phần HTTP:** Web API tuân thủ đầy đủ theo tiêu chuẩn REST, hỗ trợ các phương thức HTTP cốt lõi như GET, POST, PUT, DELETE để thực hiện các thao tác dữ liệu (CRUD). Điều này giúp việc xây dựng các dịch vụ HTTP trở nên đơn giản và nhanh chóng. API cũng tương thích hoàn toàn với các khía cạnh của giao thức HTTP, bao gồm cấu trúc URI, header của request/response, cơ chế caching, versioning (quản lý phiên bản), và nhiều định dạng nội dung (content format).

- **Tự động hóa quy trình và nâng cao hiệu suất:** Bằng cách sử dụng Web API, các doanh nghiệp có thể tự động hóa nhiều quy trình công việc, chẳng hạn như tự động cập nhật dữ liệu giữa các hệ thống. Điều này giúp loại bỏ các thao tác thủ công, giảm thiểu sai sót, từ đó tăng năng suất và hiệu quả hoạt động chung.

- **Khả năng tích hợp linh hoạt và an toàn:** API đóng vai trò như một cổng giao tiếp có kiểm soát, cho phép các ứng dụng hoặc website khác truy xuất nội dung và chức năng một cách dễ dàng khi được cấp phép. Nhờ vậy, nó không chỉ làm phong phú thêm trải nghiệm người dùng mà còn cho phép các công ty chia sẻ dữ liệu một cách chọn lọc, bảo vệ hệ thống khỏi những truy cập không mong muốn.

- **Đồng bộ hóa dữ liệu theo thời gian thực:** API cho phép các hệ thống cập nhật và trao đổi thông tin gần như ngay lập tức. Với khả năng này, dữ liệu luôn được đảm bảo tính nhất quán và chính xác, giúp các dịch vụ trở nên linh hoạt và đáp ứng tốt hơn

với những thay đổi trong môi trường kinh doanh.

- **Có tiêu chuẩn chung để sử dụng:** Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ cần sử dụng. Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

1.3. Nguyên lý hoạt động của API:

Hãy tưởng tượng:

- Bạn (Người dùng ứng dụng): Là thực khách.
- Người phục vụ: Là API Client.
- Nhà bếp: Là API Server.
- Thực đơn (Menu): Là tài liệu API (API Documentation), cho bạn biết có thể gọi những món gì và gọi như thế nào.

1.3.1. Chi tiết các bước trong luồng hoạt động API:

- API Client (Người phục vụ) nhận và gửi yêu cầu:
 - Bạn ngồi vào bàn, xem thực đơn và quyết định gọi món "Bò bít tết". Bạn gọi người phục vụ lại và nói: "Cho tôi một phần bò bít tết, chín vừa nhé".
 - Hành động bạn gọi người phục vụ và đưa ra yêu cầu chính là việc "call API". Người phục vụ (API Client) đã tiếp nhận yêu cầu từ bạn và chuẩn bị mang nó đến nhà bếp (API Server).
 - API Request (Phiếu gọi món):
 - Người phục vụ không chỉ chạy vào bếp và hét lên "Bò bít tết!". Anh ta sẽ ghi yêu cầu của bạn vào một phiếu gọi món một cách có cấu trúc để nhà bếp có thể hiểu chính xác. Phiếu gọi món này chính là API Request
 - API Server (Nhà bếp) xử lý yêu cầu:
 - Xác thực và phân quyền: Bếp trưởng nhìn vào phiếu, thấy ghi "Bàn 12" và "Khách VIP". Họ biết yêu cầu này hợp lệ và cần được ưu tiên. (Giống như API Server kiểm tra API key hoặc token xem client có quyền truy cập không).
 - Xử lý dữ liệu:
 - Bếp trưởng kiểm tra trong kho xem còn thịt bò không. (Truy xuất dữ liệu từ database).
 - Nếu còn, đầu bếp sẽ lấy thịt bò và chế biến theo đúng yêu cầu "chín vừa" và "không hành tây". (Xử lý logic nghiệp vụ).
 - API Response (Món ăn được mang ra) gửi kết quả:
 - Người phục vụ mang đĩa thức ăn từ nhà bếp ra bàn cho bạn. Đây chính là API Response.
 - Trường hợp thành công: Người phục vụ đặt đĩa bò bít tết thơm phức xuống trước mặt bạn và nói "Món của quý khách đây ạ!".

- Status Code: 200 OK (Yêu cầu đã được xử lý thành công).
- Response Headers: "Món này nóng, cẩn thận nhé!" (Thông tin thêm về phản hồi).
- Response Body: Đĩa bò bít tết ngon tuyệt. (Dữ liệu mà client yêu cầu).
- Trường hợp thất bại:Nếu nhà bếp hết thịt bò, người phục vụ sẽ quay lại bàn của bạn và nói: "Xin lỗi quý khách, hôm nay nhà hàng đã hết món bò bít tết rồi ạ."
- Status Code: 404 Not Found (Không tìm thấy tài nguyên được yêu cầu).
- Response Body: Lời xin lỗi và thông báo "Hết món". (Thông báo lỗi).

1.4. Công nghệ triển khai ASP.NET Core API:

1.4.1. Tổng quan về ASP.NET Core API:

- **Khái niệm:** ASP.NET Core API là một framework đa nền tảng, mã nguồn mở của Microsoft dùng để xây dựng các dịch vụ web (Web API), cho phép các ứng dụng khác nhau giao tiếp với nhau qua giao thức HTTP hoặc HTTPS. Nó cho phép tạo ra các dịch vụ RESTful hiệu quả và có hiệu suất cao, có thể chạy trên Windows, macOS và Linux, vv.
- **Đặc điểm nổi bật:**
 - **Đa nền tảng:** Có khả năng chạy trên nhiều hệ điều hành khác nhau như Windows, Linux và macOS.
 - **Mã nguồn mở:** Được phát triển với sự đóng góp lớn từ cộng đồng, mang lại sự minh bạch trong quá trình phát triển.
 - **Kiến trúc hướng hiệu năng:** Ngay từ giai đoạn thiết kế, hệ thống đã được xây dựng với mục tiêu cốt lõi là đạt được hiệu suất cao và độ trễ thấp nhất.
 - **Modular (Kiến trúc module):** Cho phép nhà phát triển chỉ sử dụng các thành phần cần thiết, giúp giảm kích thước ứng dụng và tăng tính linh hoạt.
 - **Dễ dàng tích hợp:** Tích hợp tốt với các công nghệ frontend hiện đại như Angular, React, Vue.js.

1.4.2. Khái niệm Framework:

- **Khái niệm:** Hãy hình dung Framework như một bộ khung sườn của một ngôi nhà được xây dựng sẵn. Nó đã có nền móng, cột, và các cấu trúc chính. Nhiệm vụ của bạn là "lấp đầy" các bức tường, trang trí nội thất và hoàn thiện nó theo ý mình. Tương tự, Framework là một bộ khung phần mềm cung cấp cấu trúc nền tảng và các chức năng chung, được đóng gói từ các thư viện, công cụ và quy tắc có sẵn.
- **Mục đích:** Mục tiêu chính là giúp lập trình viên không phải "phát minh lại bánh xe". Bằng cách cung cấp các giải pháp đã được kiểm chứng cho những vấn đề lặp đi lặp lại (như kết nối cơ sở dữ liệu, bảo mật, quản lý phiên), Framework cho phép lập trình viên tập trung nguồn lực và sự sáng tạo vào việc xây dựng các tính năng nghiệp vụ cốt lõi của ứng dụng.
- **Đặc điểm:**

- Tính mở rộng: Mặc dù cung cấp một cấu trúc có sẵn, nó vẫn đủ linh hoạt để bạn có thể tùy chỉnh và mở rộng, giống như việc bạn có thể sơn màu tường hay chọn nội thất khác nhau cho ngôi nhà.
- Luồng điều khiển đảo ngược (Inversion of Control): Thay vì bạn gọi các thư viện, Framework sẽ gọi đến mã nguồn của bạn tại những điểm thích hợp. Bạn viết code và "cắm" nó vào bộ khung đã được định sẵn.

1.4.3. EF Core Framework (Entity Framework Core):

- **Khái niệm và vai trò:** Entity Framework Core (EF Core) là một framework Object-Relational Mapping (ORM) hiện đại, mã nguồn mở được Microsoft xây dựng và bảo trì. Vai trò chính của nó là tạo ra một "cầu nối" trừu tượng, cho phép các nhà phát triển tương tác với cơ sở dữ liệu bằng cách sử dụng các đối tượng (object) và lớp (class) trong C# thay vì phải soạn thảo các câu lệnh SQL thuần túy. Bằng cách tự động hóa việc ánh xạ giữa mô hình dữ liệu trong code và lược đồ của cơ sở dữ liệu, EF Core góp phần đẩy nhanh tiến độ phát triển, tăng tính linh hoạt và hỗ trợ triển khai trên nhiều nền tảng khác nhau trong hệ sinh thái .NET. Các thành phần chính của EF Core:

- **DbContext:** Đây là thành phần trung tâm trong EF Core, hoạt động như một phiên giao tiếp với cơ sở dữ liệu. Nó chịu trách nhiệm quản lý vòng đời của các đối tượng dữ liệu (entity), theo dõi các thay đổi và điều phối các hoạt động truy vấn hay lưu trữ thông tin..

- **DbSet:** Đại diện cho một tập hợp các đối tượng của một loại (entity) cụ thể trong Context. Mỗi DbSet thường tương ứng với một bảng trong cơ sở dữ liệu, cung cấp các phương thức để thực hiện các thao tác CRUD (Create, Read, Update, Delete) trên bảng đó.

- **Entity Classes (Lớp thực thể):** Là các lớp C# thông thường (POCO - Plain Old CLR Object) được tạo ra để biểu diễn cấu trúc của các bảng trong cơ sở dữ liệu. Mỗi thuộc tính (property) của lớp thường ánh xạ tới một cột (column) và mỗi thực thể (instance) của lớp đại diện cho một hàng (row) dữ liệu.

- **LINQ (Language Integrated Query):** EF Core tích hợp sâu với LINQ, cho phép lập trình viên viết các truy vấn dữ liệu phức tạp trực tiếp trong C# một cách tự nhiên và tường minh. EF Core sẽ đảm nhận việc "dịch" các biểu thức LINQ này thành các câu lệnh SQL tối ưu, phù hợp với hệ quản trị cơ sở dữ liệu đang sử dụng.

- **Migrations:** Là một tính năng mạnh mẽ giúp quản lý và phiên bản hóa các thay đổi của cấu trúc cơ sở dữ liệu theo thời gian. Nó cho phép đồng bộ hóa một cách an toàn và có kiểm soát những cập nhật trong mô hình dữ liệu của ứng dụng với lược đồ thực tế của cơ sở dữ liệu.

1.4.4. Authentication (Xác thực) & Authorization (Phân quyền):

- **Authentication (Xác thực):**

- **Khái niệm:** Là quy trình kiểm chứng và xác nhận danh tính của một người dùng, hệ thống, hoặc ứng dụng khi họ cố gắng truy cập. Về cơ bản, nó trả lời cho câu hỏi: *"Thực sự bạn là ai?"*.

- **Mục đích:** Đảm bảo rằng chủ thể đang tương tác với hệ thống đúng là đối tượng mà họ khai báo, thường thông qua việc đối chiếu các thông tin chứng minh như tên người dùng và mật khẩu, mã OTP, hoặc dấu vân tay

- **Ý nghĩa:** Đây là tuyến phòng thủ đầu tiên giúp bảo vệ tài nguyên hệ thống, ngăn chặn sự xâm nhập trái phép, duy trì tính toàn vẹn và bảo mật cho dữ liệu của người dùng, cũng như tuân thủ các tiêu chuẩn an ninh thông tin.

- **Các phương pháp phổ biến:** Bao gồm các kỹ thuật như xác thực dựa trên Session, xác thực bằng Token (ví dụ: JWT - JSON Web Token), các giao thức ủy quyền như OAuth 2.0, OpenID Connect, và tích hợp đăng nhập thông qua các nhà cung cấp định danh bên thứ ba (Google, Facebook, GitHub).

- **Authorization (Phân quyền):**

- **Khái niệm:** Là cơ chế kiểm soát truy cập, có nhiệm vụ phân định các đặc quyền và phạm vi hoạt động của một chủ thể đã xác thực danh tính thành công. Về bản chất, nó trả lời cho câu hỏi: "Với danh tính đã được xác nhận, bạn có quyền làm gì?".

- **Mục đích:** Nhằm giới hạn và kiểm soát quyền tương tác của người dùng, đảm bảo họ chỉ có thể thao tác trên những tài nguyên dữ liệu hoặc các tính năng được bảo vệ mà họ đã được cấp phép.

- **Ví dụ:** Một tài khoản với vai trò "Quản trị viên" (Admin) sẽ được cấp quyền thao tác trên toàn bộ hệ thống, trong khi một "Thành viên" (Member) thông thường sẽ bị giới hạn quyền hạn trong phạm vi dữ liệu cá nhân của chính họ..

1.4.5. Triển khai ASP.NET Core MVC để khai thác API (Consuming API):

ASP.NET Core được thiết kế với sự tích hợp sâu sắc và tối ưu cho kiến trúc

Model-View-Controller (MVC), cung cấp một phương pháp luận rõ ràng và có tổ chức để xây dựng các ứng dụng web hiện đại.

- Cấu trúc dự án MVC điển hình trong ASP.NET Core:

Khi khởi tạo một dự án web ASP.NET Core theo mẫu MVC, Visual Studio sẽ tự động tạo ra một cấu trúc thư mục chuẩn hóa như sau:

- **Controllers:** Là nơi chứa các lớp Controller. Đây là thành phần chịu trách nhiệm tiếp nhận và xử lý các yêu cầu (request) đến từ người dùng, sau đó điều phối các bước tiếp theo.

- **Views:** Chứa các tệp tin liên quan đến tầng trình bày (giao diện người dùng), thường là các tệp Razor (.cshtml). Các tệp này được sắp xếp trong các thư mục con tương ứng với tên của từng Controller (ví dụ: Views/Home, Views/Product).

- **Models:** Là không gian dành cho các lớp Model, nơi định nghĩa cấu trúc của các đối tượng dữ liệu cũng như cài đặt các quy tắc logic nghiệp vụ của ứng dụng.

- **wwwroot:** Đây là thư mục gốc công khai của web, chứa toàn bộ các tài sản tĩnh (static assets) như tệp CSS, mã JavaScript, hình ảnh, và các thư viện phía client.

- **appsettings.json:** Tập tin cấu hình trung tâm của ứng dụng. Nó lưu trữ các thiết lập quan trọng như chuỗi kết nối đến cơ sở dữ liệu, các khóa API, hoặc các tham số tùy chỉnh khác.

- **Program.cs:** Là điểm khởi chạy của ứng dụng. Tại đây, các dịch vụ cần thiết được đăng ký vào hệ thống (sử dụng Dependency Injection) và pipeline xử lý yêu cầu HTTP (middleware) được định cấu hình.

- **Cách triển khai Controller:**

- Một Controller trong ASP.NET Core về cơ bản là một lớp C# kế thừa từ lớp Controller cơ sở. Theo quy ước, các lớp này được đặt trong thư mục Controllers và có hậu tố là "Controller" trong tên (ví dụ: ProductController). Action Method là các phương thức public trong Controller, dùng để xử lý các yêu cầu HTTP. Chúng thường trả về IActionResult hoặc các kiểu dẫn xuất như ViewResult, JsonResult.

- Các phương thức public bên trong một Controller được gọi là Action Method. Mỗi Action Method được thiết kế để xử lý một loại yêu cầu HTTP cụ thể và thường trả về một đối tượng IActionResult (hoặc các kiểu kế thừa như ViewResult, JsonResult).

- **Nhiệm vụ cốt lõi của Controller:** đóng vai trò trung gian, tiếp nhận yêu cầu, phân tích dữ liệu đầu vào (từ route, query string, hoặc body), tương tác với Model để xử lý logic, và cuối cùng là chọn 1 View thích hợp để trả về kết quả cho người dùng.

- **Cách triển khai View:**

- View trong ASP.NET Core được xây dựng bằng cú pháp Razor (trong các tệp .cshtml), một template engine mạnh mẽ cho phép kết hợp liền mạch giữa mã HTML và C# bằng cách sử dụng ký tự @.
- Các tệp View được tổ chức một cách logic trong thư mục Views, phân cấp theo tên của Controller chịu trách nhiệm gọi chúng (ví dụ: Views/Product/Details.cshtml).
- Một View thường nhận dữ liệu được truyền từ Controller (dưới dạng một ViewModel hoặc một tập hợp đối tượng) để sử dụng thông tin đó và tái hiện thành giao diện HTML hoàn chỉnh gửi về trình duyệt.

- **Cách triển khai Model:**

- Trong kiến trúc MVC của ASP.NET Core, Model là các lớp C# đơn giản (thường được gọi là POCO - Plain Old CLR Object). Chúng được đặt trong thư mục Models và có vai trò đại diện cho các thực thể dữ liệu cũng như logic nghiệp vụ của ứng dụng.

CHƯƠNG 2: XÂY DỰNG ỨNG DỤNG WEB API

2.1. Phát biểu bài toán ứng dụng:

❖ Giới thiệu:

Hệ thống API quản lý đồ uống được xây dựng nhằm cung cấp một giao diện lập trình ứng dụng (API) tập trung, phục vụ cho các ứng dụng client (như website, ứng dụng di động) trong lĩnh vực kinh doanh ẩm thực và đồ uống (F&B). Hệ thống này cung cấp các endpoint để quản lý thông tin sản phẩm (đồ uống), danh mục, xử lý đơn hàng, và phân quyền người dùng, tạo ra một backend vững chắc và linh hoạt.

❖ Ý nghĩa và vai trò:

- **Nền tảng cho ứng dụng client:** Cung cấp các phương thức để các ứng dụng phía người dùng (web, mobile) có thể hiển thị menu, cho phép khách hàng đặt món một cách nhanh chóng và tiện lợi.

- **Tối ưu hóa quản lý:** Giúp chủ cửa hàng hoặc người quản trị có thể quản lý sản phẩm, theo dõi tồn kho, và xử lý đơn hàng một cách hiệu quả thông qua các endpoint được bảo mật.

- **Thúc đẩy chuyển đổi số:** Góp phần vào quá trình hiện đại hóa và tự động hóa trong lĩnh vực kinh doanh ẩm thực, đồ uống.

2.2. Phân tích yêu cầu của hệ thống API:

- Hệ thống cần cung cấp các endpoint và cơ chế bảo mật để phục vụ cho hai nhóm đối tượng chính: Quản trị viên (Admin) và Người dùng (tương tác thông qua một ứng dụng client).

2.2.1. Yêu cầu phi chức năng:

- Bên cạnh các chức năng nghiệp vụ, hệ thống API cần đáp ứng các yêu cầu phi chức năng sau để đảm bảo hiệu suất, bảo mật và tính ổn định trong quá trình vận hành:

- **API được thiết kế rõ ràng, nhất quán:** Các endpoint cần được thiết kế theo chuẩn RESTful, có cấu trúc logic và tài liệu đi kèm rõ ràng, giúp các lập trình viên phía client dễ dàng hiểu và tích hợp.

- **Bảo mật và phân quyền chặt chẽ:** Hệ thống API phải yêu cầu xác thực (Authentication) cho mỗi yêu cầu truy cập đến các tài nguyên được bảo vệ. Mỗi người dùng có một vai trò cụ thể (ví dụ: Admin, User) với các quyền hạn (Authorization) truy cập và thao tác khác nhau. Người dùng chỉ có thể thực hiện các hành động phù hợp với quyền được cấp.

- **Đảm bảo tính toàn vẹn của dữ liệu:** Toàn bộ dữ liệu trong hệ thống phải được kiểm soát chặt chẽ để duy trì tính chính xác và nhất quán. Các endpoint cho phép thay đổi hoặc xóa dữ liệu quan trọng.

2.2.2. Yêu cầu chức năng:

a. Chức năng cho Admin (Quản trị viên):

STT	Chức năng	Mô tả
1	Quản lý sản phẩm	Thêm, xóa sửa sản phẩm đồ uống
2	Quản lý mã giảm giá	Thêm, sửa xóa các mã giảm giá
3	Quản lý đơn hàng	Xem danh sách đơn hàng đã đặt
4	Quản lý tài khoản	Xem, xóa, thay đổi thông tin và thay đổi quyền người dùng (Admin, khách hàng)
5	Quản lý liên hệ	Xem và phản hồi câu hỏi hoặc phản hồi của khách hàng
6	Quản lý đánh giá	Xem, xóa các đánh giá không phù hợp từ khách hàng

Bảng 1: Bảng yêu cầu chức năng cho Admin (Quản trị viên)

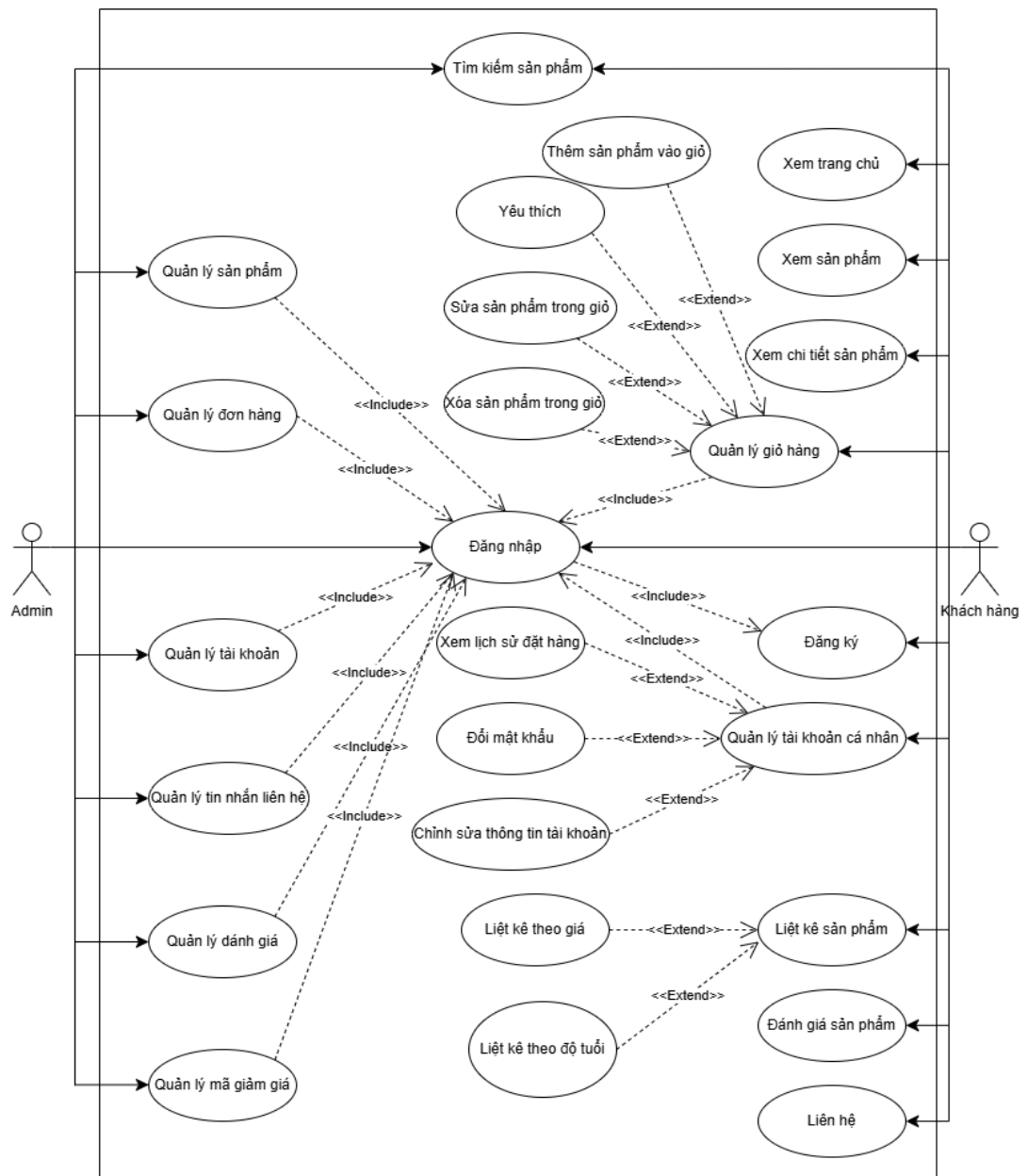
b. Chức năng cho Khách hàng:

STT	Chức năng	Mô tả
1	Đăng ký	Tạo tài khoản mới để sử dụng hệ thống
2	Đăng nhập	Truy cập hệ thống bằng tài khoản đã đăng ký
3	Xem sản phẩm	Xem đồ uống theo danh mục hoặc theo tên
4	Xem chi tiết sản phẩm	Xem chi tiết đồ uống và xem đánh giá về đồ uống đó
5	Tìm kiếm sản phẩm	Tìm kiếm theo tên , giá, độ tuổi có thể sắp xếp theo thứ tự
6	Quản lý giỏ hàng	Thêm, sửa, xóa đồ uống có trong giỏ hàng, chỉ có những người đã đăng nhập mới có thể mua hàng
7	Đặt hàng	Đặt hàng từ giỏ hàng có thể áp mã giảm giá nếu có
8	Đánh giá sản phẩm	Viết nhận xét, đánh giá trong chi tiết sản phẩm
9	Liên hệ	Gửi phản hồi, yêu cầu hỗ trợ
10	Chỉnh sửa thông tin tài khoản	Cập nhật username, email, mật khẩu

Bảng 2: Bảng yêu cầu chức năng cho Khách hàng

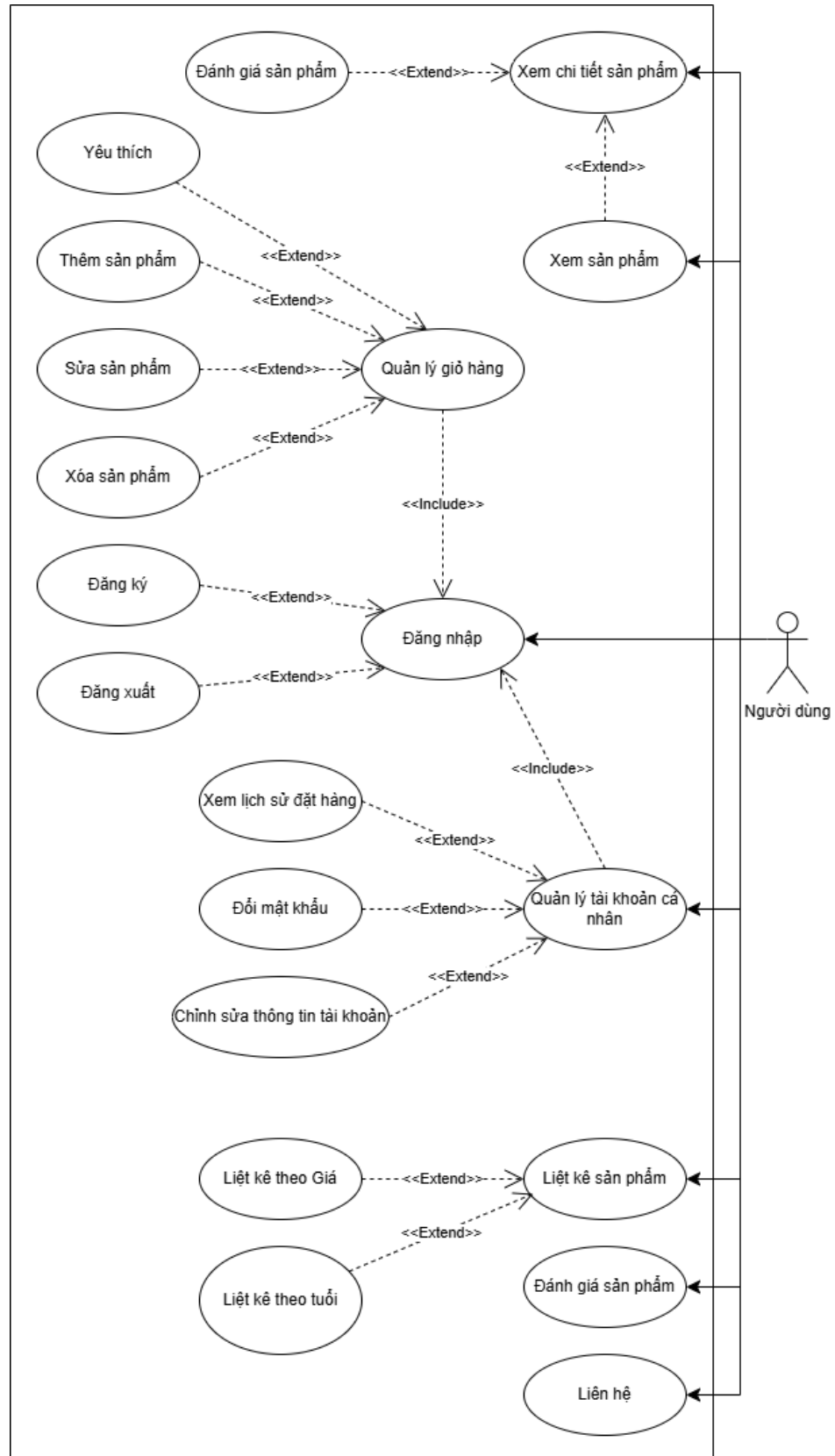
2.2.3. Biểu đồ Use case:

a. Biểu đồ Use case tổng quát của hệ thống:



Hình 2: Biểu đồ Use case tổng quát

b. Biểu đồ Use case cho người dùng:



Hình 3: Biểu đồ Use case người dùng

Đặc tả Use case Người dùng:

1)Đăng ký tài khoản:

Mô tả	Người dùng mới tạo tài khoản bằng cách cung cấp thông tin cá nhân (username, email, mật khẩu,xác nhận mật khẩu, họ tên, sdt, địa chỉ)
Tác nhân chính	Người dùng
Luồng chính	Người dùng chọn “Đăng ký” Nhập Username, Email, mật khẩu,xác nhận mật khẩu,họ tên, sdt, địa chỉ Hệ thống kiểm tra thông tin và tạo tài khoản Người dùng nhận thông báo đăng ký thành công
Ngoại lệ	Username/Email đã tồn tại hoặc nhập sai thông tin định dạng Mật khẩu và xác nhận mật khẩu không giống nhau

Bảng 3: Bảng đặc tả Use case Đăng ký tài khoản

2)Đăng nhập:

Mô tả	Người dùng nhập tên đăng nhập và mật khẩu để truy cập vào hệ thống
Tác nhân chính	Người dùng
Luồng chính	Người dùng chọn “Đăng nhập” Nhập Tên người dùngvà mật khẩu Hệ thống xác thực thông tin, xuất thông báo thành công và chuyển người dùng đến Trang chủ
Ngoại lệ	Tên người dùng hoặc mật khẩu không chính xác

Bảng 4: Bảng đặc tả Use case Đăng nhập

3)Quản lý giỏ hàng:

Mô tả	Cho phép người thêm, cập nhật số lượng hoặc xoá sản phẩm đã thêm khỏi giỏ hàng
Tác nhân chính	Người dùng
Luồng chính	Người dùng chọn đồ uống muốn mua Thêm vào giỏ hàng, có thể cập nhật số lượng hoặc xoá sản phẩm trong giỏ
Ngoại lệ	Sản phẩm hết hàng hoặc lỗi khi thêm vào giỏ

Bảng 5: Bảng đặc tả Use case Quản lý giỏ hàng

4)Quản lý tài khoản cá nhân:

Mô tả	Người dùng xem thông tin tài khoản
Tác nhân chính	Người dùng
Luồng chính	Người dùng vào mục “Tài khoản ” Xem thông tin tài khoản,
Ngoại lệ	Thông tin không khớp

Bảng 6: Bảng đặc tả Use case Quản lý tài khoản cá nhân

5)Quản lý đơn hàng:

Mô tả	Người dùng xem lịch sử đơn hàng đã đặt và chi tiết đơn hàng
Tác nhân chính	Người dùng
Luồng chính	Người dùng vào mục “Đơn hàng” Người dùng nhấn vào “Xem” Xem thông tin chi tiết của đơn hàng đó
Ngoại lệ	Không có đơn hàng hoặc lỗi truy vấn dữ liệu

Bảng 7: Bảng đặc tả Use case Quản lý đơn hàng

6)Đánh giá:

Mô tả	Người dùng đánh giá sản phẩm
Tác nhân chính	Người dùng
Luồng chính	Người dùng vào chi tiết sản phẩm Người dùng chuyển qua tab Đánh giá trong chi tiết sản phẩm Nhập nội dung đánh giá và xếp hạng sao (từ 1 đến 5 sao) Gửi đánh giá
Ngoại lệ	Không nhấn đánh giá

Bảng 8: Bảng đặc tả Use case Đánh giá

7)Tìm kiếm:

Mô tả	Người dùng tìm kiếm sản phẩm theo tên sản phẩm
Tác nhân chính	Người dùng
Luồng chính	Người dùng nhập từ khoá vào thanh tìm kiếm Hệ thống trả về danh sách sản phẩm phù hợp
Ngoại lệ	Không có kết quả phù hợp với từ khoá

Bảng 9: Bảng đặc tả Use case Tìm kiếm

8)Đăng xuất:

Mô tả	Người dùng đăng xuất khỏi tài khoản
Tác nhân chính	Người dùng
Luồng chính	Người dùng chọn chức năng “Đăng xuất” Hệ thống đăng xuất người dùng và trở về trang chủ
Ngoại lệ	Không có

Bảng 10: Bảng đặc tả Use case Đăng xuất

9) Sản phẩm:

Mô tả	Người dùng xem danh sách sản phẩm theo danh mục hoặc theo giá hoặc độ tuổi có sắp xếp theo thứ tự và xem chi tiết sản phẩm
Tác nhân chính	Người dùng
Luồng chính	Người dùng chọn “loại sản phẩm” Hệ thống hiển thị danh sách sản phẩm Người dùng có thể chọn theo giá hoặc độ tuổi để hiển thị danh sách sản phẩm phù hợp hơn với nhu cầu Người dùng chọn sản phẩm để xem chi tiết
Ngoại lệ	Không có sản phẩm trong danh mục hoặc giá hoặc độ tuổi sai khi tải dữ liệu

Bảng 11: Bảng đặc tả Use case Sản phẩm

10) Chi tiết sản phẩm:

Mô tả	Người dùng xem chi tiết sản phẩm
Tác nhân chính	Người dùng
Luồng chính	Người dùng chọn sản phẩm để xem chi tiết sản phẩm Hệ thống hiển thị chi tiết sản phẩm gồm tên, danh mục, size, Còn hay hết hàng, giá, mô tả , số lượng
Ngoại lệ	Lỗi khi tải dữ liệu

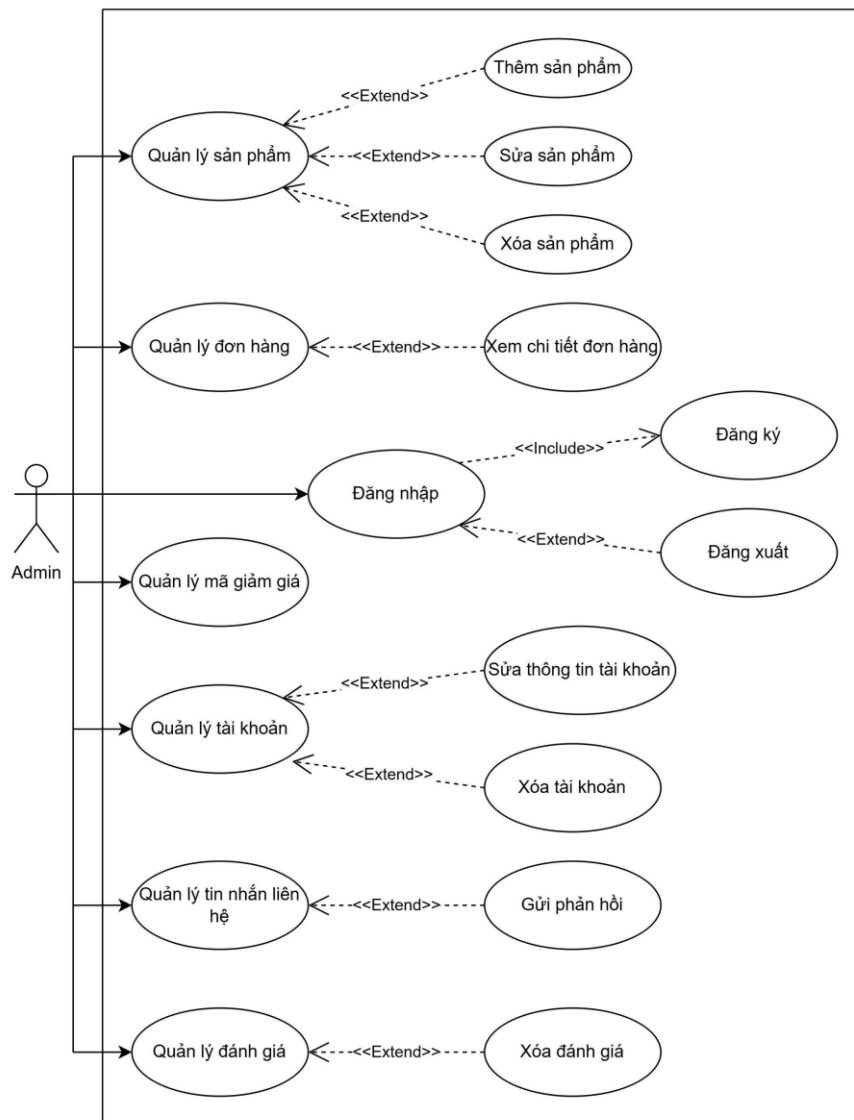
Bảng 12: Bảng đặc tả Use case Chi tiết sản phẩm

11) Liên hệ:

Mô tả	Người dùng gửi phản hồi hoặc yêu cầu hỗ trợ
Tác nhân chính	Người dùng
Luồng chính	Người dùng chọn Liên hệ với gmail hoặc số hotline trên màn hình
Ngoại lệ	Không

Bảng 13: Bảng đặc tả Use case Liên hệ

c. Biểu đồ Use case cho Admin (Quản trị viên):



Hình 4: Biểu đồ Use case Admin

Đặc tả Use case Admin (Quản trị viên):

1)Đăng nhập:

Mô tả	Admin đăng nhập vào hệ thống bằng tài khoản được cấp
Tác nhân chính	Admin
Luồng chính	Admin chọn “Đăng nhập” Nhập Tên đăng nhập và mật khẩu Hệ thống xác thực thông tin, xuất thông báo đăng nhập thành công và chuyển admin đến trang Quản trị
Ngoại lệ	Sai thông tin đăng nhập

Bảng 14: Bảng đặc tả Use case Đăng nhập (Admin)

2)Đăng xuất:

Mô tả	Admin kết thúc làm việc
Tác nhân chính	Admin
Luồng chính	Admin chọn “Đăng xuất” Hệ thống quay lại trang chủ
Ngoại lệ	Không có

Bảng 15: Bảng đặc tả Use case Đăng xuất (Admin)

3)Quản lý mã giảm giá:

Mô tả	Admin tạo mã giảm giá
Tác nhân chính	Admin
Luồng chính	Admin chọn “Tạo mã giảm giá” Nhập Mã voucher, tên voucher, mô tả, loại giảm giá, giá trị giảm,giảm tối đa, đơn hàng tối thiểu,số lượng,ngày bắt đầu, ngày kết thúc, áp dụng cho mã sản phẩm Admin nhấn “Tạo voucher” Hệ thống cập nhật thông tin vừa lưu
Ngoại lệ	Nhập các giá trị sai không đúng logic

Bảng 16: Bảng đặc tả Use case Quản lý mã giảm giá

4)Quản lý doanh thu:

Mô tả	Admin xem tổng doanh thu
Tác nhân chính	Admin
Luồng chính	Hệ thống hiển thị tổng doanh thu ở phần “Dashboard quản trị”
Ngoại lệ	Tổng doanh thu sai

Bảng 17: Bảng đặc tả Use case Quản lý danh mục

5)Quản lý sản phẩm:

Mô tả	Admin thêm, xoá hoặc sửa sản phẩm có trong hệ thống
Tác nhân chính	Admin
Luồng chính	Admin chọn “Quản lý Sản phẩm” Chọn sản phẩm Thêm mới, xoá hoặc chỉnh sửa thông tin sản phẩm
Ngoại lệ	Trùng mã sản phẩm hoặc thiếu thông tin

Bảng 18: Bảng đặc tả Use case Quản lý sản phẩm

6)Quản lý đơn hàng:

Mô tả	Admin xem danh sách và thông tin đơn hàng
Tác nhân chính	Admin
Luồng chính	Admin chọn “Quản lý Đơn hàng” Xem danh sách đơn hàng
Ngoại lệ	Không có đơn hàng hoặc lỗi truy vấn dữ liệu

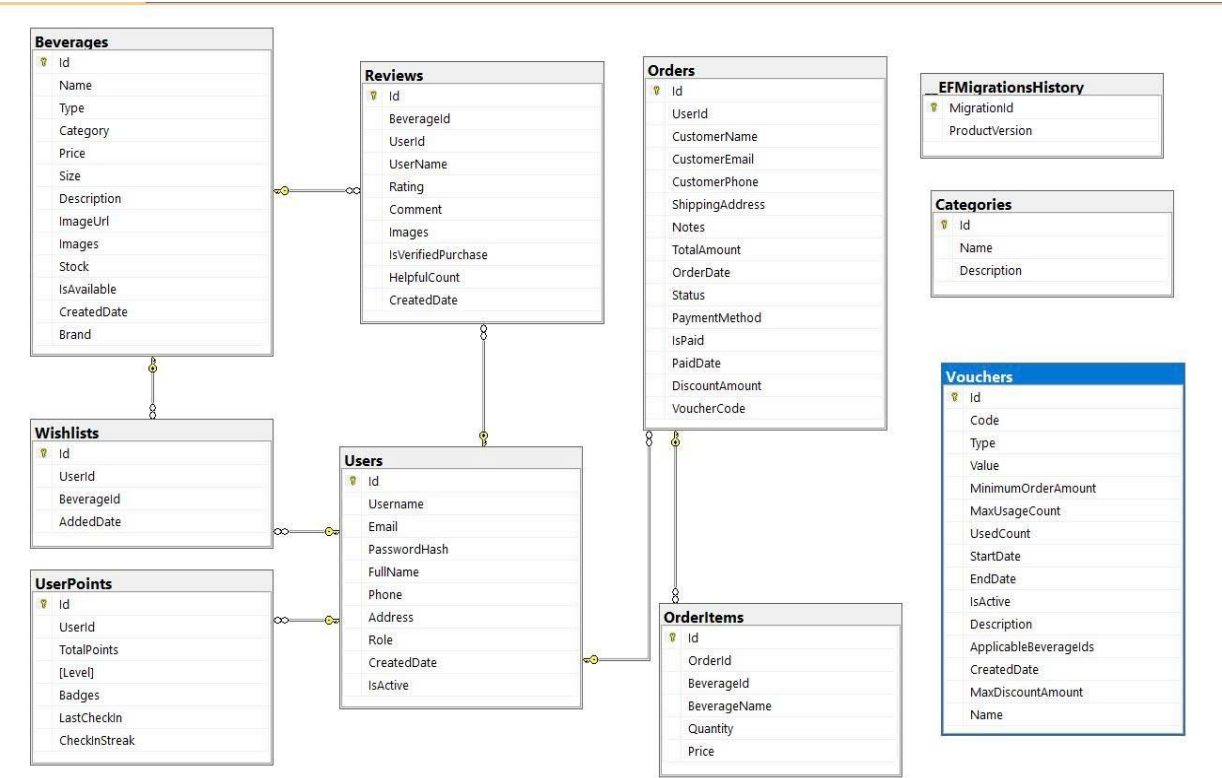
Bảng 19: Bảng đặc tả Use case Quản lý đơn hàng

7)Quản lý tài khoản:

Mô tả	Admin theo dõi, sửa đổi thông tin người dùng (nếu cần thiết)
Tác nhân chính	Admin
Luồng chính	Admin chọn “Quản lý Tài khoản” Xem danh sách tài khoản Chọn tài khoản cụ thể để chỉnh sửa/xoá
Ngoại lệ	Không tìm thấy người dùng

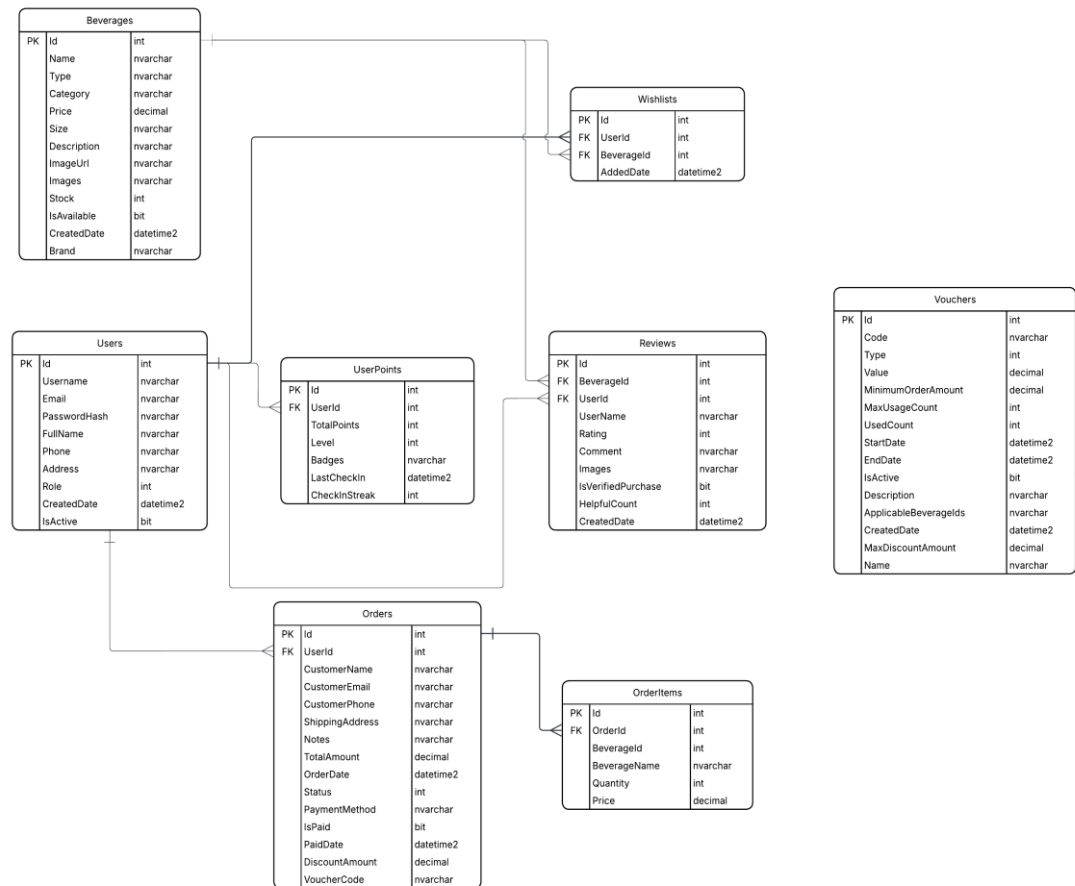
Bảng 20: Bảng đặc tả Use case Quản lý tài khoản

2.3. Thiết kế cơ sở dữ liệu:
2.3.1. Sơ đồ Database:



Hình 5: Sơ đồ Database

2.3.2.Sơ đồ ERD:



Hình 6: Sơ đồ ERD

Mối quan hệ giữa các thực thể:

Thực thể	Mối quan hệ
Users	Một User có thể tạo nhiều Order (1:N) Một User có thể viết nhiều Review (1:N) Một User có thể có nhiều Wishlist (1:N) Một User có một hoặc nhiều UserPoint (1:N)
Orders	Một Order được tạo bởi một User (N:1) Một Order có một OrderItem (N:1)
Beverage	Một Beverage có thể có nhiều Review (1:N) Một Beverage có thể nằm trong nhiều OrderItem (1:N) Một Beverage có thể nằm trong nhiều Wishlist (1:N)
Reviews	Một Review được viết bởi một User Một Review dành cho một Beverage (N:1)
Wishlists	Một Wishlist thuộc về một User(N:1) Một Wishlist dành cho một Beverage(N:1)
UserPoints	Một UserPoint thuộc về User(1:1)

Bảng 23: Bảng mô tả mối quan hệ giữa các thực thể

2.3.4. Bảng mô tả dữ liệu:

1) Bảng Vouchers:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính, tự động tăng.
2	Name	nvarchar(MAX)	Not null	Tên của voucher để dễ quản lý (ví dụ: "Giảm giá 20% mừng khai trương").
3	Code	nvarchar(MAX)	Not null	Mã voucher duy nhất để người dùng nhập khi thanh toán (ví dụ: "SALE20").
4	Type	int	Not null	Loại voucher (ví dụ: 1 cho giảm theo %, 2 cho giảm số tiền cố định).
5	Value	decimal(18, 2)	Not null	Giá trị của voucher (là % hoặc số tiền tùy theo Type).
6	MinimumOrderAmount	decimal(18, 2)	Not null	Giá trị đơn hàng tối thiểu để có thể áp dụng voucher.
7	MaxUsageCount	int	Not null	Tổng số lần voucher có thể được sử dụng.
8	UsedCount	int	Not null	Số lần voucher đã được sử dụng.
9	StartDate	datetime2(7)	Not null	Ngày và giờ voucher bắt đầu có hiệu lực.
10	EndDate	datetime2(7)	Not null	Ngày và giờ voucher hết hiệu lực.
11	IsActive	bit	Not null	Trạng thái của voucher (1: đang hoạt động, 0: đã tắt).
12	Description	nvarchar(MAX)	Not null	Mô tả chi tiết về các điều kiện và quyền lợi của voucher.
13	ApplicableBeverageIds	nvarchar(MAX)	Null	Danh sách ID của các sản phẩm có thể áp dụng voucher. Nếu là NULL thì áp dụng cho tất cả sản phẩm.
14	MaxDiscountAmount	decimal(18, 2)	Not null	Số tiền giảm giá tối đa có thể nhận được (áp dụng cho voucher loại %).
15	CreatedDate	datetime2(7)	Not null	Ngày tạo voucher trong hệ thống.

Bảng 24: Bảng Vouchers

2)Bảng Wishlist:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính của bảng.
2	UserId	int	FK	Khóa ngoại, liên kết đến người dùng (Users).
3	BeverageId	int	FK	Khóa ngoại, liên kết đến sản phẩm (Beverages) được yêu thích.
4	AddedDate	datetime2	Not null	Ngày thêm sản phẩm vào danh sách yêu thích.

Bảng 26: Bảng Wishlists

3) Bảng OrderItems:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính của bảng.
2	OrderId	int	FK	Khóa ngoại, liên kết đến bảng Orders để biết chi tiết này thuộc đơn hàng nào.
3	BeverageId	int	FK	Khóa ngoại, liên kết đến bảng Beverages để biết sản phẩm được mua là gì.
4	BeverageName	nvarchar	Not null	Tên của sản phẩm tại thời điểm mua hàng.
5	Quantity	int	Not null	Số lượng sản phẩm được mua.
6	Price	decimal	Not null	Đơn giá của sản phẩm tại thời điểm mua hàng.

Bảng 27: Bảng OrderItems

4) Bảng Orders:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính, mã định danh duy nhất cho mỗi đơn hàng.
2	UserId	int	FK	Khóa ngoại, liên kết đến bảng Users để biết đơn hàng này của ai.
3	CustomerName	nvarchar	Not null	Tên của khách hàng đặt hàng.
4	CustomerEmail	nvarchar	Not null	Email của khách hàng.
5	CustomerPhone	nvarchar	Not null	Số điện thoại của khách hàng.
6	ShippingAddress	nvarchar	Not null	Địa chỉ giao hàng.
7	Notes	nvarchar	Null	Ghi chú của khách hàng cho đơn hàng.
8	TotalAmount	decimal	Not null	Tổng số tiền của đơn hàng.
9	OrderDate	datetime2	Not null	Ngày và giờ đặt hàng.
10	Status	int	Not null	Trạng thái của đơn hàng (ví dụ: Đang xử lý, Đã giao).
11	PaymentMethod	nvarchar	Not null	Phương thức thanh toán (ví dụ: COD, Chuyển khoản).
12	IsPaid	bit	Not null	Trạng thái thanh toán (Đã thanh toán hay chưa).
13	PaidDate	datetime2	Null	Ngày và giờ thanh toán.
14	DiscountAmount	decimal	Null	Số tiền được giảm giá.
15	VoucherCode	nvarchar	Null	Mã voucher đã sử dụng cho đơn hàng.

Bảng 28: Bảng Orders

5) Bảng Beverages:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính, mã định danh duy nhất cho mỗi đồ uống.
2	Name	nvarchar	Not null	Tên của đồ uống.
3	Type	nvarchar	Not null	Loại đồ uống (ví dụ: Cà phê, Trà sữa, Nước ép).
4	Category	nvarchar	Null	Danh mục cụ thể của đồ uống.
5	Price	decimal	Not null	Giá bán của đồ uống.
6	Size	nvarchar	Null	Kích thước của đồ uống (ví dụ: S, M, L).
7	Description	nvarchar	Null	Mô tả chi tiết về đồ uống.
8	ImageUrl	nvarchar	Null	Đường dẫn đến hình ảnh chính của sản phẩm.
9	Images	nvarchar	Null	Danh sách các đường dẫn đến hình ảnh khác của sản phẩm.
10	Stock	int	Not null	Số lượng sản phẩm còn lại trong kho.
11	IsAvailable	bit	Not null	Trạng thái có sẵn để bán của sản phẩm (True/False).
12	CreatedDate	datetime2	Not null	Ngày sản phẩm được thêm vào hệ thống.
13	Brand	nvarchar	Null	Thương hiệu của sản phẩm.

Bảng 29: Bảng Beverages

6) Bảng Reviews:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính, mã định danh duy nhất cho mỗi đánh giá.
2	BeverageId	int	FK	Khóa ngoại, liên kết đến sản phẩm (Beverages) được đánh giá.
3	UserId	int	FK	Khóa ngoại, liên kết đến người dùng (Users) đã viết đánh giá.
4	UserName	nvarchar	Not null	Tên của người dùng viết đánh giá.
5	Rating	int	Not null	Điểm đánh giá (ví dụ: từ 1 đến 5 sao).
6	Comment	nvarchar	Null	Nội dung bình luận, nhận xét chi tiết.
7	Images	nvarchar	Null	Hình ảnh đính kèm trong đánh giá.
8	IsVerifiedPurchase	bit	Not null	Xác nhận người dùng đã thực sự mua sản phẩm này.
9	HelpfulCount	int	Not null	Số lượt người khác thấy đánh giá này là hữu ích.
10	CreatedDate	datetime2	Not null	Ngày và giờ đánh giá được tạo.

Bảng 30: Bảng Reviews

7) Bảng Users:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính, mã định danh duy nhất cho mỗi người dùng.
2	Username	nvarchar	Not null, Unique	Tên đăng nhập của người dùng.
3	Email	nvarchar	Not null, Unique	Địa chỉ email của người dùng.
4	PasswordHash	nvarchar	Not null	Mật khẩu của người dùng đã được mã hóa.
5	FullName	nvarchar	Not null	Họ và tên đầy đủ của người dùng.
6	Phone	nvarchar	Null	Số điện thoại của người dùng.
7	Address	nvarchar	Null	Địa chỉ của người dùng.
8	Role	int	Not null	Vai trò của người dùng trong hệ thống (ví dụ: Admin, User).
9	CreatedDate	datetime2	Not null	Ngày và giờ tài khoản được tạo.
10	IsActive	bit	Not null	Trạng thái hoạt động của tài khoản (True/False).

Bảng 31: Bảng Users

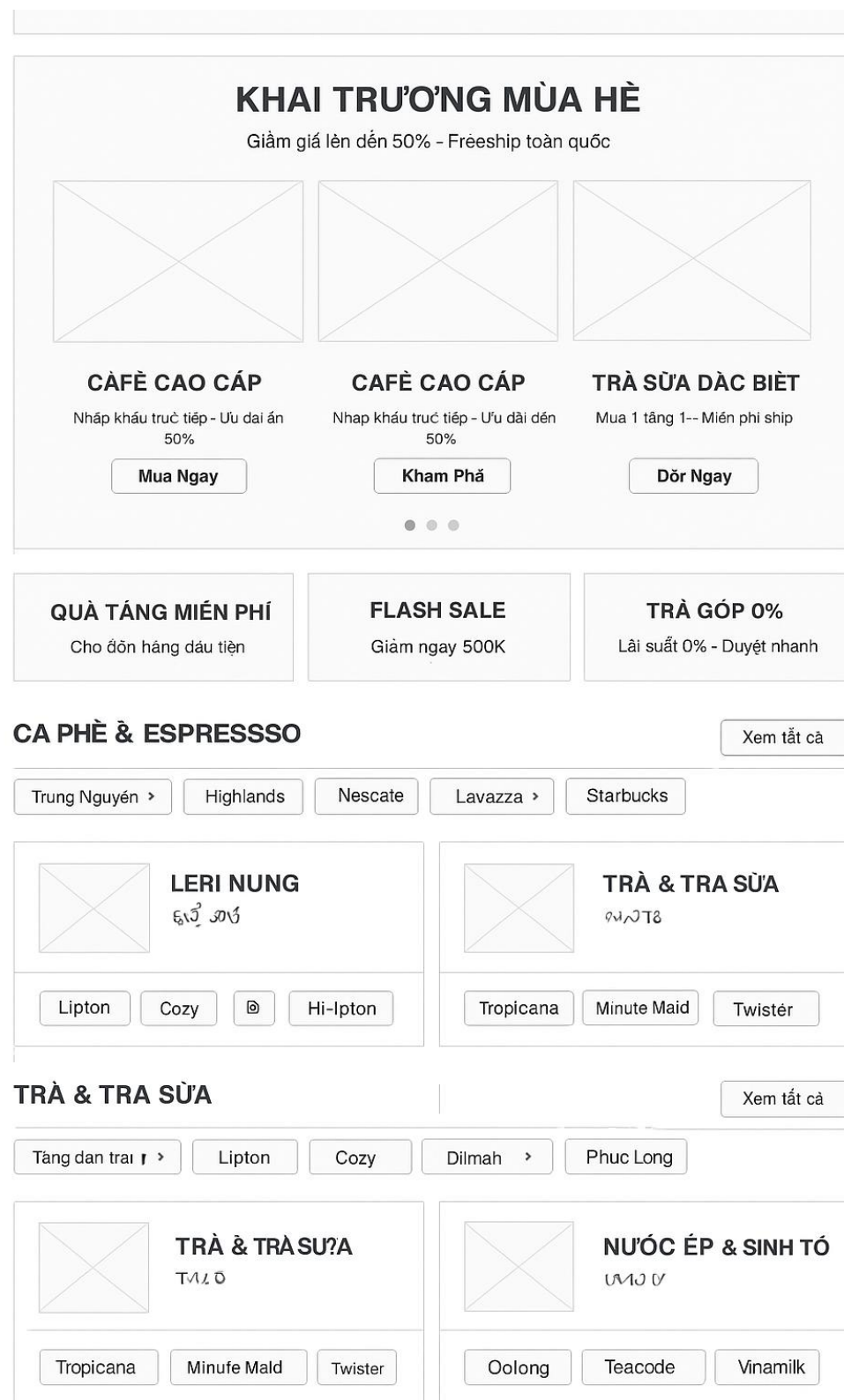
8) Bảng UserPoints:

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	Id	int	PK	Khóa chính của bảng.
2	UserId	int	FK	Khóa ngoại, liên kết đến người dùng (Users).
3	TotalPoints	int	Not null	Tổng số điểm người dùng đang có.
4	Level	nvarchar	Not null	Cấp độ thành viên của người dùng (ví dụ: Đồng, Bạc, Vàng).
5	Badges	nvarchar	Null	Các huy hiệu mà người dùng đạt được.
6	LastCheckIn	datetime2	Null	Thời gian lần cuối cùng người dùng điểm danh.
7	CheckInStreak	int	Null	Số ngày điểm danh liên tiếp.

Bảng 31: Bảng UserPoints

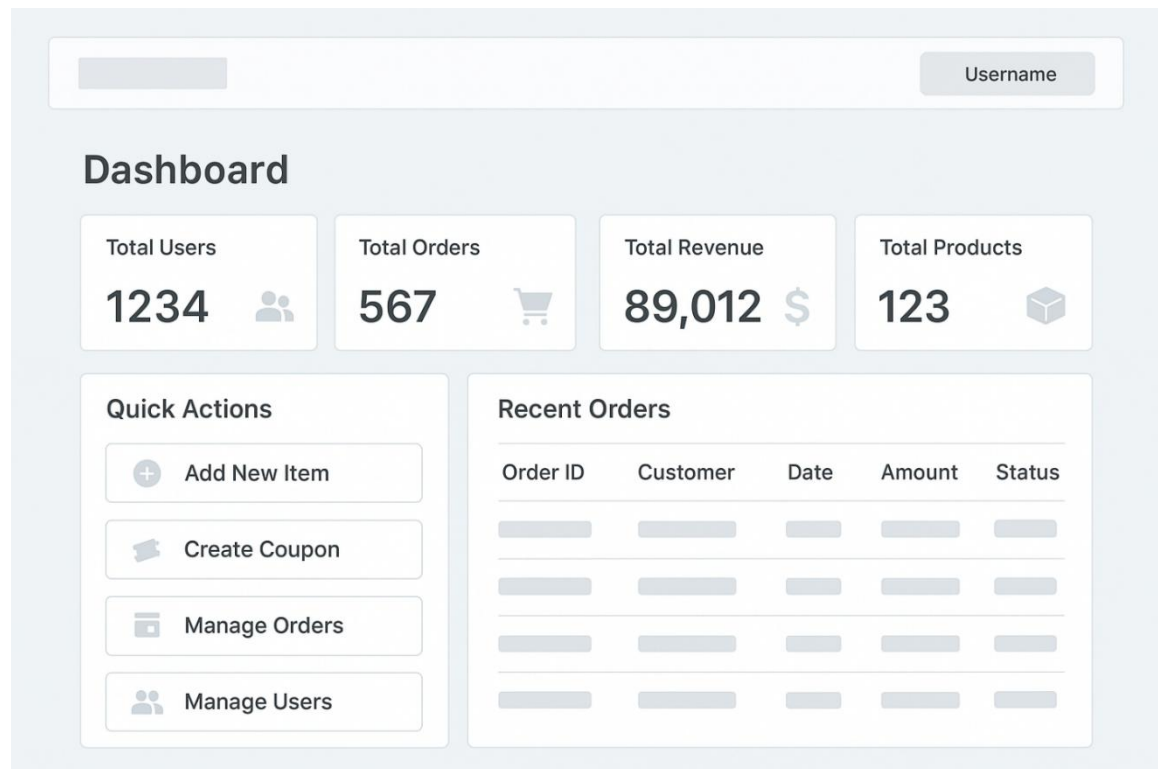
2.4. Thiết kế giao diện Wireframe:

2.4.1.Wireframe Trang chủ:



Hình 7: Wireframe Trang chủ

2.4.2.Wireframe DashBoard Title:



Hình 8: Wireframe DashBoard Title

2.4.3. Wireframe Đăng nhập:

The wireframe shows a login interface. At the top, there is a header with a house icon and the text 'Đăng nhập'. Below this, there are two input fields: 'Tên đăng nhập' (Username) and 'Mật khẩu' (Password). A dark grey button labeled 'Đăng nhập' is positioned below the password field. A horizontal line separates the login section from the registration section, which contains the text 'Chưa có tài khoản? Đăng ký ngay'. At the bottom, a light grey box contains the text 'Tài khoản demo:' followed by 'Username: admin | Password: admin123'.

Đăng nhập

Tên đăng nhập

Nhập tên đăng nhập

Mật khẩu

Nhập mật khẩu

Đăng nhập

Chưa có tài khoản? Đăng ký ngay

Tài khoản demo:

Username: admin | Password: admin123

Hình 9: Wireframe Đăng nhập

2.4.4.Wireframe Đăng ký:

 **Đăng ký tài khoản**

Tên đăng nhập *

Email *

Xác nhận mật khẩu

Mật khẩu

Họ và tên *

Số điện thoại *

Địa chỉ

Đăng ký

Đã có tài khoản? Đăng nhập ngay

Hình 10: Wireframe Đăng ký

2.5. Thiết kế các thành phần API và Consumin API:

2.5.1. API:

❖ Models:

- User

Tên thuộc tính	Mô tả
Id	Mã định danh duy nhất cho mỗi người dùng.
Username	Tên dùng để đăng nhập vào hệ thống.
Email	Địa chỉ email của người dùng.
PasswordHash	Mật khẩu đã được mã hóa để bảo mật.
FullName	Họ và tên đầy đủ của người dùng.
Phone	Số điện thoại liên lạc của người dùng.
Address	Địa chỉ của người dùng.
Role	Vai trò của người dùng (ví dụ: Khách hàng, Quản trị viên).
CreatedDate	Ngày tài khoản được tạo.
IsActive	Trạng thái hoạt động của tài khoản (đang hoạt động hoặc bị khóa).

Bảng 32: Bảng Model User

- WishList:

Tên thuộc tính	Mô tả
Id	Mã định danh duy nhất cho mỗi mục trong danh sách yêu thích.
UserId	Mã của người dùng đã thêm sản phẩm này vào danh sách.
BeverageId	Mã của sản phẩm (đồ uống) được yêu thích.
AddedDate	Ngày sản phẩm được thêm vào danh sách yêu thích.

Bảng 33: Bảng Model Wishlist

- Review:

Tên thuộc tính	Mô tả
Id	Mã định danh duy nhất cho mỗi bài đánh giá.
BeverageId	Mã của sản phẩm (đồ uống) được đánh giá.
UserId	Mã của người dùng đã viết bài đánh giá.
UserName	Tên hiển thị của người dùng viết đánh giá.
Rating	Điểm xếp hạng (từ 1 đến 5 sao) mà người dùng đưa ra.
Comment	Nội dung bình luận, nhận xét chi tiết về sản phẩm.
Images	Các đường dẫn đến hình ảnh đính kèm trong bài đánh giá.
IsVerifiedPurchase	Dấu hiệu xác nhận người dùng đã thực sự mua sản phẩm này.
HelpfulCount	Số lượt người khác thấy bài đánh giá này hữu ích.
CreatedDate	Ngày bài đánh giá được tạo.

Bảng 34: Bảng Model Review

❖ **Controller (một vài ví dụ cơ bản):**

- UploadController:

Hành động	Chức năng chính
UploadBeverageImage(file)	Tải lên một ảnh duy nhất cho sản phẩm đồ uống, đồng thời tạo một ảnh thu nhỏ (thumbnail).
UploadMultipleImages(files)	Tải lên nhiều ảnh (tối đa 10 ảnh) cho sản phẩm đồ uống trong một lần yêu cầu.
DeleteImage(fileName)	Xóa một ảnh sản phẩm và ảnh thu nhỏ tương ứng khỏi máy chủ dựa vào tên tệp.

Bảng 36: Bảng Controller UploadController

- WishlistController:

Hành động	Chức năng chính
GetUserWishlist(userId)	Lấy danh sách tất cả các sản phẩm yêu thích của một người dùng cụ thể.
AddToWishlist(wishlist)	Thêm một sản phẩm mới vào danh sách yêu thích của người dùng.
RemoveFromWishlist(id)	Xóa một sản phẩm khỏi danh sách yêu thích dựa vào mã (ID) của mục đó.
RemoveByUserAndBeverage(userId, beverageId)	Xóa một sản phẩm khỏi danh sách yêu thích dựa vào mã người dùng và mã sản phẩm.
CheckInWishlist(userId, beverageId)	Kiểm tra xem một sản phẩm cụ thể đã có trong danh sách yêu thích của người dùng hay chưa.

Bảng 37: Bảng Controller WishlistController

- AuthController:

Hành động	Chức năng chính
Register(request)	Tiếp nhận thông tin và tạo một tài khoản người dùng mới trong hệ thống.
Login(request)	Kiểm tra và xác thực thông tin đăng nhập của người dùng để cấp quyền truy cập.
HashPassword(password)	Mã hóa mật khẩu người dùng thành một chuỗi an toàn để lưu trữ vào cơ sở dữ liệu.
VerifyPassword(password, hash)	So sánh mật khẩu người dùng cung cấp với chuỗi đã được mã hóa để xác thực.
GenerateToken(user)	Tạo một chuỗi định danh (token) duy nhất cho người dùng sau khi đăng nhập thành công.

Bảng 38: Bảng Controller AuthController

2.5.2. Consuming API (MVC):

- **2.5.2.1. Models:**
- Beverage

Tên thuộc tính	Mô tả
Id	Mã định danh duy nhất cho mỗi sản phẩm đồ uống.
Name	Tên của đồ uống (ví dụ: "Cà Phê Sữa Đá").
Type	Loại chi tiết của đồ uống (ví dụ: "Arabica", "Trà Oolong").
Category	Danh mục chung chứa đồ uống (ví dụ: "Cà phê", "Trà sữa").
Brand	Thương hiệu của sản phẩm (ví dụ: "Trung Nguyên", "Phúc Long").
Price	Giá bán của đồ uống.
Size	Kích thước của đồ uống (ví dụ: "S", "M", "L").
Description	Mô tả chi tiết về thành phần và hương vị của sản phẩm.
ImageUrl	Đường dẫn đến hình ảnh hiển thị của đồ uống.
Stock	Số lượng sản phẩm còn lại trong kho.
IsAvailable	Trạng thái cho biết sản phẩm có sẵn để bán hay không.
CreatedDate	Ngày sản phẩm được thêm vào hệ thống.

Bảng 39: Bảng Model Beverage

- Category:

Tên thuộc tính	Mô tả
Id	Mã định danh duy nhất cho mỗi danh mục.
Name	Tên của danh mục (ví dụ: "Cà phê", "Trà sữa").
Description	Mô tả chi tiết hơn về danh mục sản phẩm.

Bảng 40: Bảng Model Category

- Review:

Tên thuộc tính	Mô tả
Id	Mã định danh duy nhất cho mỗi bài đánh giá.
BeverageId	Mã của sản phẩm (đồ uống) được đánh giá.
UserId	Mã của người dùng đã viết bài đánh giá.
UserName	Tên hiển thị của người dùng viết đánh giá.
Rating	Điểm xếp hạng (từ 1 đến 5 sao) mà người dùng đưa ra.
Comment	Nội dung bình luận, nhận xét chi tiết về sản phẩm.
Images	Các đường dẫn đến hình ảnh đính kèm trong bài đánh giá.
IsVerifiedPurchase	Dấu hiệu xác nhận người dùng đã thực sự mua sản phẩm này.
HelpfulCount	Số lượt người khác thấy bài đánh giá này hữu ích.
CreatedDate	Ngày bài đánh giá được tạo.

Bảng 41: Bảng Model Review

❖ **Controller :**

- ShopController:

Hành động	Chức năng chính
Index(category)	Hiển thị danh sách các sản phẩm đồ uống, có thể lọc theo danh mục.
Details(id)	Hiển thị thông tin chi tiết của một sản phẩm, bao gồm cả các đánh giá liên quan.
AddReview(...)	Xử lý việc người dùng gửi một đánh giá mới cho sản phẩm.
Search(keyword)	Tìm kiếm và hiển thị các sản phẩm khớp với từ khóa người dùng nhập vào.
Wishlist()	Hiển thị trang danh sách sản phẩm yêu thích của người dùng.
AddToCart(...)	Thêm một sản phẩm vào giỏ hàng trong session.
Cart()	Hiển thị trang giỏ hàng với tất cả sản phẩm đã được thêm.
UpdateCart(...)	Cập nhật số lượng của một sản phẩm có trong giỏ hàng.
RemoveFromCart(...)	Xóa một sản phẩm ra khỏi giỏ hàng.
Checkout()	Hiển thị trang thanh toán với các sản phẩm từ giỏ hàng.
PlaceOrder(...)	Tiếp nhận thông tin khách hàng và tạo đơn hàng mới từ giỏ hàng.
ManageImages()	Hiển thị trang quản lý hình ảnh sản phẩm.

Bảng 42: Bảng Controller ShopController

- AccountController:

Hành động	Chức năng chính
Login(GET)	Hiển thị trang đăng nhập cho người dùng.
Login(POST)	Xử lý thông tin đăng nhập, xác thực người dùng và lưu thông tin vào session.
Register(GET)	Hiển thị trang đăng ký tài khoản mới.
Register(POST)	Xử lý thông tin đăng ký, tạo tài khoản mới và tự động đăng nhập cho người dùng.
Logout()	Xóa thông tin người dùng khỏi session để đăng xuất khỏi hệ thống.
Orders()	Lấy và hiển thị danh sách các đơn hàng của người dùng đang đăng nhập.
OrderDetails(id)	Lấy và hiển thị thông tin chi tiết của một đơn hàng cụ thể.
Profile()	Hiển thị trang thông tin cá nhân của người dùng đang đăng nhập.

Bảng 43: Bảng Controller AccountController

2.6. Triển khai và cài đặt:

❖ Kiến trúc Hệ thống:

- Backend: Xây dựng bằng ASP.NET Core Web API, chịu trách nhiệm xử lý logic nghiệp vụ, giao tiếp với cơ sở dữ liệu và cung cấp các điểm cuối (endpoints) để Frontend truy cập.
- Frontend: Xây dựng bằng ASP.NET Core MVC, chịu trách nhiệm hiển thị giao diện người dùng (UI) và gọi đến các API của Backend để lấy hoặc gửi dữ liệu.

❖ Môi trường triển khai:

- Hệ điều hành: Windows 11
- IDE (Môi trường phát triển): Visual Studio 2022
- Cơ sở dữ liệu: SQL Server Management Studio 20
- Công nghệ Backend (API):
- Ngôn ngữ: C# với ASP.NET Core Web API

❖ Thư viện:

- Entity Framework Core: Truy cập và thao tác cơ sở dữ liệu theo mô hình Code First.
- Microsoft.EntityFrameworkCore.SqlServer: Cung cấp kết nối từ EF Core đến SQL Server.
- Microsoft.EntityFrameworkCore.Tools: Hỗ trợ tạo migration, cập nhật database từ dòng lệnh.

❖ Công nghệ Frontend (Giao diện):

- Ngôn ngữ: C# với ASP.NET Core MVC
- Giao diện: HTML5, CSS3, Bootstrap 5
- Công cụ kiểm thử API: Swagger, Postman

❖ Cách triển khai:

- **Bước 1:**Thiết lập Backend (API):
 - Tạo project ASP.NET Core Web API, chọn phiên bản .NET phù hợp.
 - Kết nối CSDL trong file appsettings.json của API.
 - Cấu hình DbContext trong file Program.cs của API.
 - Thực hiện quy trình Code First (tạo Migration, cập nhật Database) trên project API.
 - Xây dựng các API Controller để xử lý các chức năng CRUD và logic

- **Bước 2:**Thiết lập Frontend (MVC):
 - Tạo một project ASP.NET Core MVC riêng biệt.
 - Trong file appsettings.json của MVC, cấu hình địa chỉ URL của Backend API để có thể gửi yêu cầu.
 - Thiết kế giao diện người dùng (các file View).
 - Xây dựng các MVC Controller để nhận tương tác từ người dùng, sau đó gọi đến các endpoint trên API để xử lý dữ liệu và hiển thị kết quả ra View.
- **Bước 3:**Triển khai (Deploy):
 - Xuất bản (Publish) cả hai project API và MVC ra các thư mục riêng.
 - Triển khai cả hai lên máy chủ (ví dụ: deploy local trên IIS).
 - Cấu hình CORS (Cross-Origin Resource Sharing) trên API để cho phép ứng dụng MVC có thể truy cập tài nguyên.

CHƯƠNG 3: KẾT QUẢ CHƯƠNG TRÌNH

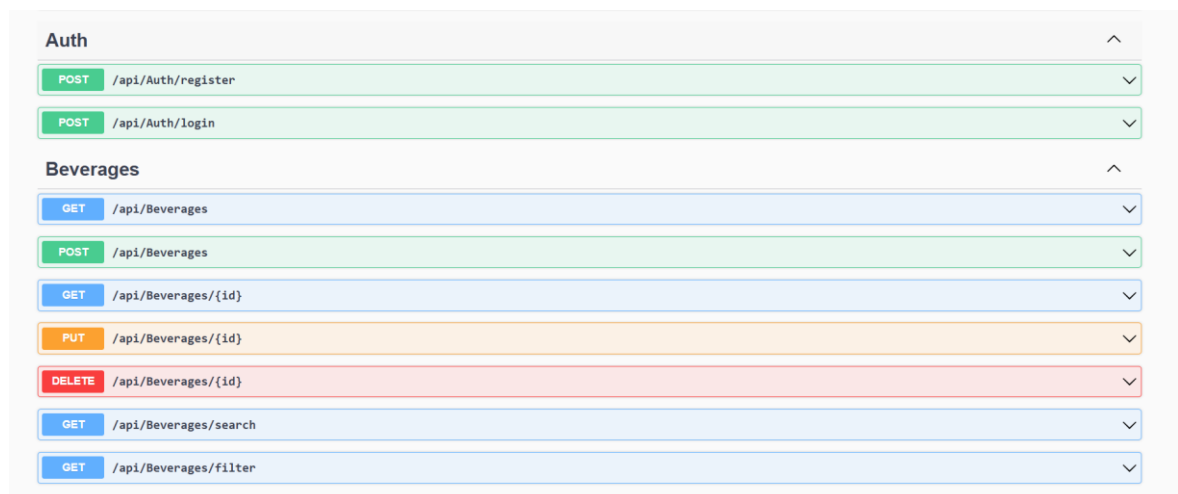
3.1.Giao diện API:

3.1.1.Admin:



Hình 11: Admin

3.1.2 Auth và Beverages



Hình 12: Auth và Beverages

3.1.3 Categories , Order và Reviews

Categories		^
GET	/api/Categories	▼
Orders		^
GET	/api/Orders	▼
POST	/api/Orders	▼
GET	/api/Orders/{id}	▼
GET	/api/Orders/user/{userId}	▼
Reviews		^
GET	/api/Reviews/beverage/{beverageId}	▼
POST	/api/Reviews	▼
DELETE	/api/Reviews/{id}	▼
GET	/api/Reviews/stats/{beverageId}	▼

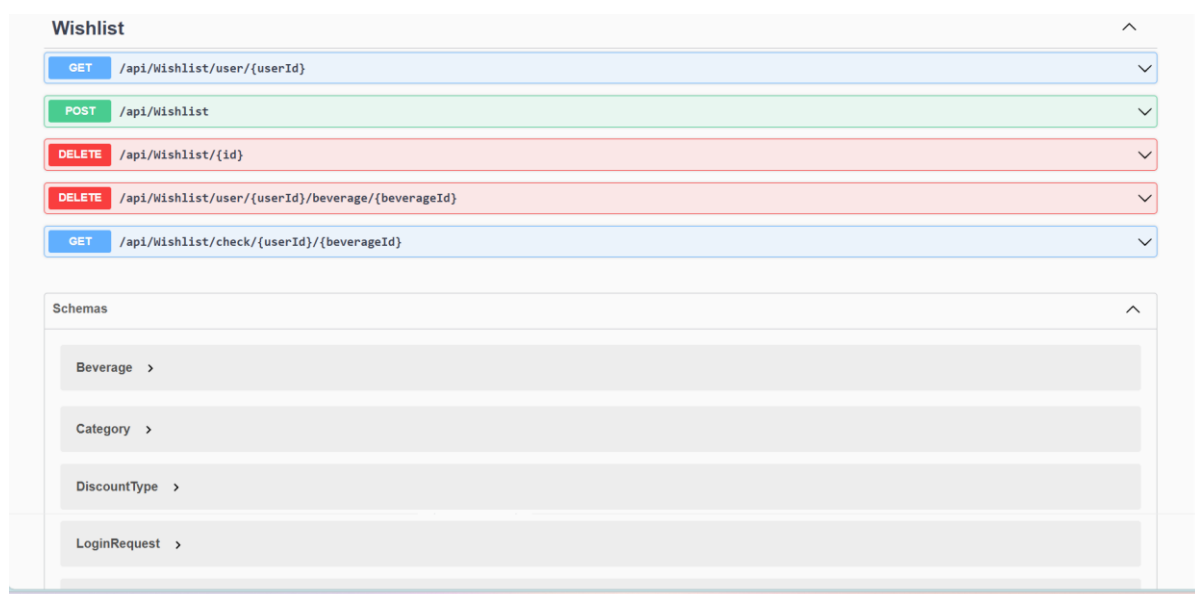
Hình 13: Categories , Order và Reviews

3.14 Upload à Voucher

Upload		^
POST	/api/Upload/beverage-image	▼
DELETE	/api/Upload/beverage-image	▼
POST	/api/Upload/beverage-images	▼
Voucher		^
GET	/api/Voucher	▼
POST	/api/Voucher	▼
GET	/api/Voucher/all	▼
GET	/api/Voucher/{id}	▼
PUT	/api/Voucher/{id}	▼
DELETE	/api/Voucher/{id}	▼
GET	/api/Voucher/validate/{code}	▼
POST	/api/Voucher/apply	▼

Hình 14: Upload và Voucher

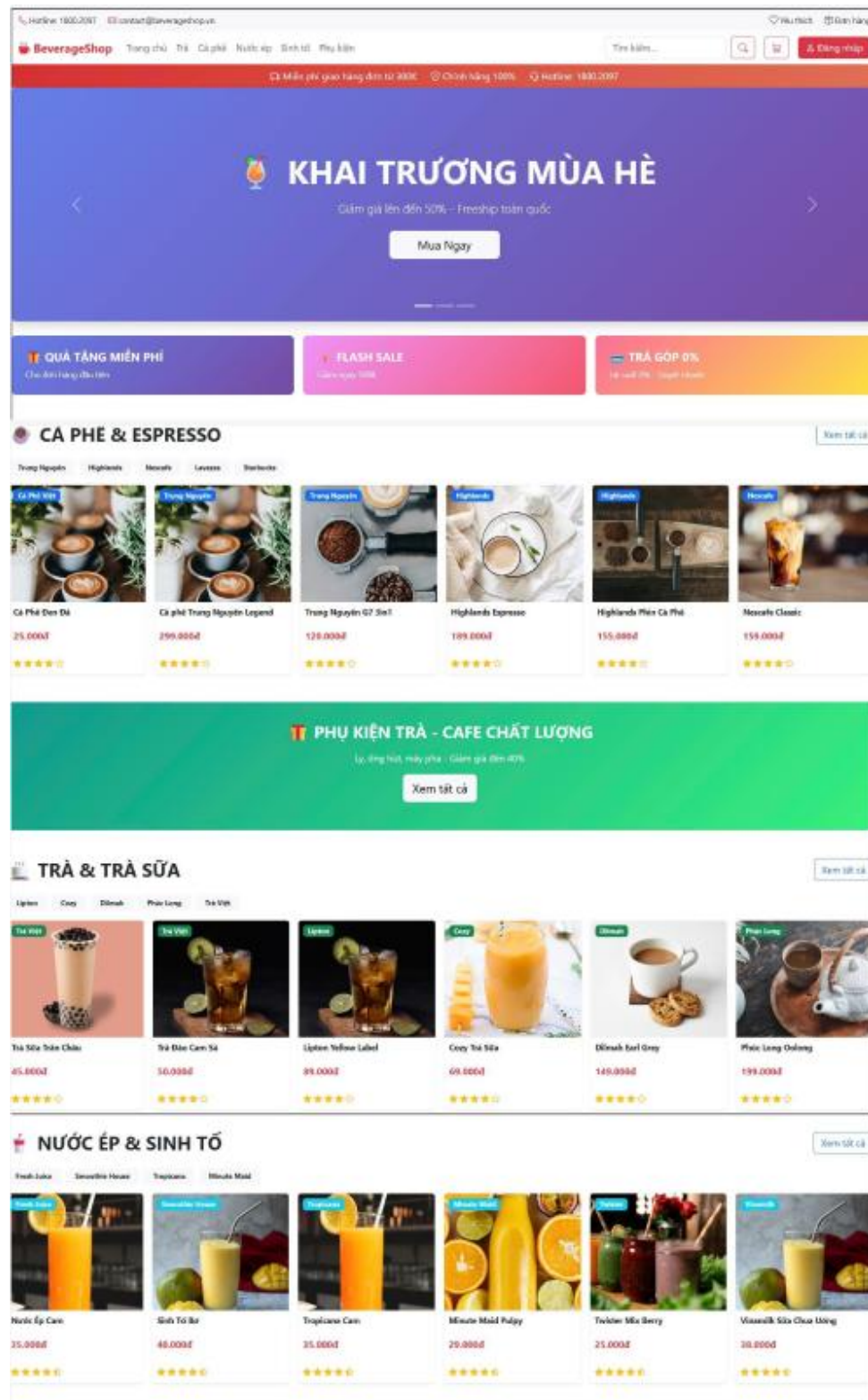
3.1.5 Wishlist



Hình 15 : Wishlist

3.2 Giao diện Consuming API (MVC)

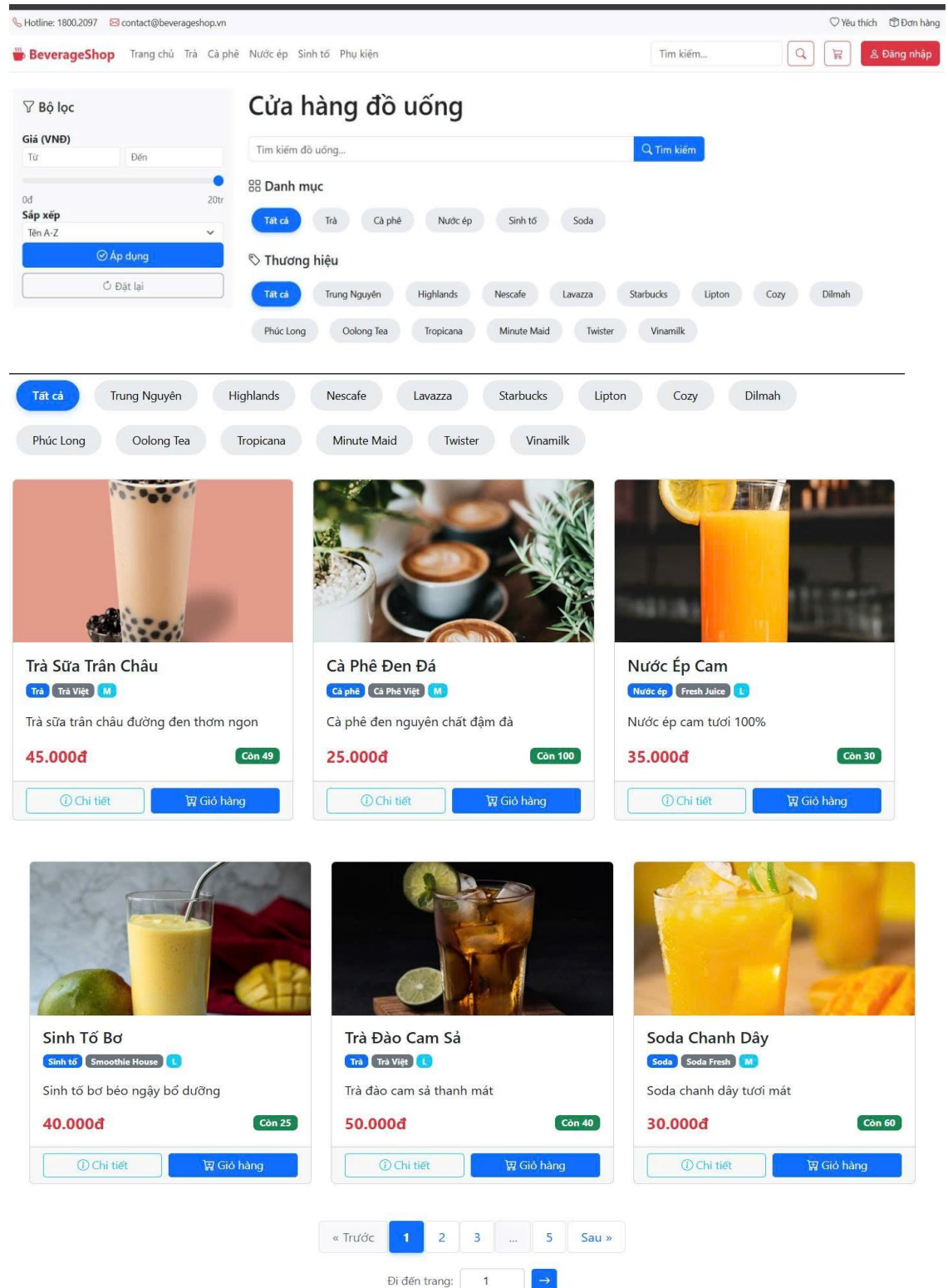
3.2.1 Trang Chủ



Hình 16: Trang chủ

Trang chủ hiện thị giao diện đồ uống , quảng cáo banner,các chương trình khuyến mãi

3.2.2 Trang cửa hàng đồ uống



Hình 17: Trang cửa hàng đồ uống

Người dùng có thể thêm sản phẩm vào giỏ hàng hoặc xem chi tiết sản phẩm. Ngoài ra còn có thể sắp xếp đồ uống theo giá cả hoặc độ tuổi

3.2.3 Trang giỏ hàng

BeverageShop Trang chủ Trà Cà phê Nước ép Sinh tố Phụ kiện

Giỏ hàng của bạn

Tên đồ uống	Loại	Đơn giá	Số lượng	Thành tiền	Thao tác
Trà Sữa Trân Châu	Trà sữa	45,000 đ	2 <input type="button" value="⬅"/>	90,000 đ	<input type="button" value="🗑 Xóa"/>
Cà Phê Đen Đá	Cà phê đen	25,000 đ	1 <input type="button" value="⬅"/>	25,000 đ	<input type="button" value="🗑 Xóa"/>
Tổng cộng:				115,000 đ	

BeverageShop
 Cửa hàng đồ uống chất lượng cao, phục vụ tận tâm.

Liên hệ
 Hotline: 1800.2097
 contact@beverageshop.vn
 Hà Nội, Việt Nam

Hỗ trợ khách hàng
 Chính sách đổi trả
 Chính sách vận chuyển
 Điều khoản sử dụng

Hình 18: Trang giỏ hàng

Khi người dùng đã thêm đồ uống vào giỏ hàng thì có thể vào trang giỏ hàng để xem các sản phẩm mình đã thêm . Có thể kiểm tra lại số lượng , giá tiền , tổng tiền và nếu muốn mua sắm thêm thì có nút “ tiếp tục mua sắm” hoặc “ thanh toán”

3.2.4 Thanh toán Đơn hàng

Thanh toán đơn hàng

Thông tin khách hàng

Họ và tên *

Email *

Số điện thoại *

Địa chỉ giao hàng *

Mã giảm giá

Gợi ý: WELCOME10, FREESHIP, DRINK15

Đơn hàng của bạn

Sản phẩm	SL	Giá
Trà Sữa Trân Châu	2	90,000 đ
Cà Phê Đen Đá	1	25,000 đ
Tạm tính:		115,000 đ
Tổng cộng:		115,000 đ

Hình 19: Trang thanh toán đơn hàng

Khi người dùng nhấn nút thanh toán ở trang giỏ hàng sẽ hiện ra trang thanh toán đơn hàng . người dùng cần nhập đúng thông tin gồm họ tên, email, số điện thoại, địa chỉ để thanh toán.Ngoài ra người dùng có thể nhập mã giảm giá để giảm tiền.

3.2.5. Đăng ký:

Đăng ký tài khoản

Tên đăng nhập *

Email *

Mật khẩu *

Xác nhận mật khẩu *

Họ và tên *

Số điện thoại *

Địa chỉ

Đăng ký

Đã có tài khoản? [Đăng nhập ngay](#)

Hình 20: Đăng ký

Người dùng sẽ đăng ký tài khoản bằng các thông tin cơ bản như tên đăng nhập, email, mật khẩu, xác nhận mật khẩu, họ tên, số điện thoại, địa chỉ. Sau khi đăng ký xong, người dùng sẽ nhận thông báo “Đăng ký thành công” và chuyển hướng về trang Đăng nhập.

3.2.6 Đăng nhập:

Đăng nhập

Tên đăng nhập

Mật khẩu

Đăng nhập


Chưa có tài khoản? [Đăng ký ngay](#)

Tài khoản demo:
Username: **admin** | Password: **admin123**


Hình 21: Đăng nhập


Ở trang này, người dùng có thể Đăng nhập bằng tài khoản đã Đăng ký.

3.2.7.Tài khoản:


Thông tin cá nhân

Tên đăng nhập:	admin
Họ và tên:	Administrator
Email:	admin@beverageshop.com
Số điện thoại:	0123456789
Địa chỉ:	Hà Nội
Vai trò:	Quản trị viên
Ngày tạo:	19/10/2025


ĐƠN HÀNG CỦA TÔI


TIẾP TỤC MUA SẮM

Hình 22: Tài khoản

Sau khi đăng nhập thành công người dùng có thể xem lại thông tin của mình đã tạo ở phần đăng ký, biết được vai trò và ngày tạo của mình.Ngoài ra còn có hiển thị nút “Đơn hàng của tôi “ và “ tiếp tục mua sắm”

3.2.8.Đặt hàng thành công:

Đặt hàng thành công!

Cảm ơn bạn đã đặt hàng!

Đơn hàng của bạn đã được tiếp nhận. Chúng tôi sẽ liên hệ với bạn sớm nhất.

Thông tin đơn hàng

Mã đơn hàng:

#1

Khách hàng:

bao anh

Email:

abao2932@gmail.com

Số điện thoại:

0334867140

Địa chỉ giao hàng:

99999

Ngày đặt:

19/10/2025 14:41

Trạng thái:

Pending

Chi tiết đơn hàng

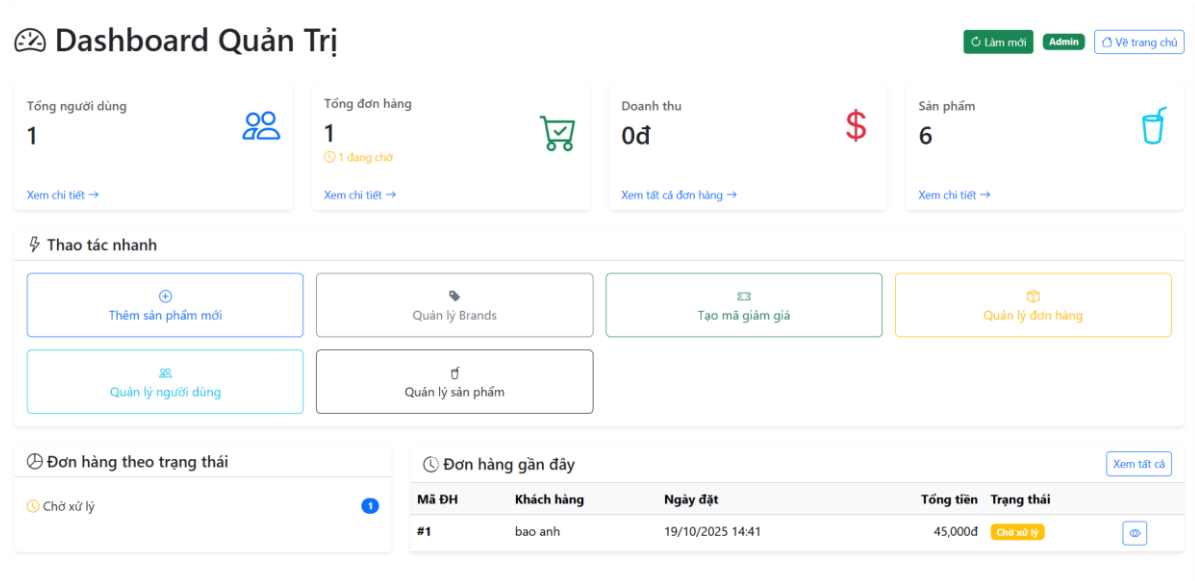
Sản phẩm	Số lượng	Đơn giá	Thành tiền
Trà Sữa Trân Châu	1	45,000 đ	45,000 đ
Tổng cộng:			45,000 đ

Hình 23: Đặt hàng thành công

Sau khi đặt hàng thành công người dùng có thể xem thông tin đơn hàng của mình và biết được trạng thái đơn hàng.Ngoài ra còn có hiển thị nút “ tiếp tục mua sắm”

3.3.Giao diện Admin (Quản trị viên):

3.3.1Trang chủ Quản trị:



Hình 24: Trang chủ Quản trị

Đây là trang Dashboard Quản Trị, nơi hiển thị tổng quan về hoạt động của cửa hàng với các thống kê chính như "Tổng người dùng", "Tổng đơn hàng", "Doanh thu" và "Sản phẩm".

Ở giữa là khu vực "Thao tác nhanh", cho phép quản trị viên truy cập trực tiếp vào các mục quản lý quan trọng như "Quản lý người dùng", "Quản lý sản phẩm" và "Quản lý đơn hàng". Bên dưới là danh sách các đơn hàng gần đây và thống kê đơn hàng theo trạng thái. Admin có thể nhấn nút "Về trang chủ" ở góc trên bên phải để quay lại trang web chính của cửa hàng.

3.3.2.Quản lý Sản phẩm:

Quản Lý Sản Phẩm

[+ Thêm sản phẩm mới](#)
[← Quay lại Dashboard](#)

Ảnh	Tên Sản Phẩm	Danh Mục	Loại	Size	Giá	Tồn Kho	Trạng Thái	Thao Tác
	Trà Sữa Trân Châu Trà sữa trân châu đường đen thơm ngon	Trà	Trà sữa	Size M	45.000 đ	49	Còn hàng	
	Cà Phê Đen Đá Cà phê đen nguyên chất đậm đà	Cà phê	Cà phê đen	Size M	25.000 đ	100	Còn hàng	
	Nước Ép Cam Nước ép cam tươi 100%	Nước ép	Nước ép	Size L	35.000 đ	30	Còn hàng	
	Sinh Tố Bơ Sinh tố bơ béo ngậy bổ dưỡng	Sinh tố	Sinh tố	Size L	40.000 đ	25	Còn hàng	
	Trà Đào Cam Sả Trà đào cam sả thanh mát	Trà	Trà trái cây	Size L	50.000 đ	40	Còn hàng	
	Soda Chanh Dây Soda chanh dây tươi mát	Soda	Soda	Size M	30.000 đ	60	Còn hàng	
	Ly Giữ Nhiệt 500ml Ly giữ nhiệt cao cấp, giữ nóng/lạnh 6-8 giờ	Phụ kiện	Phụ kiện	Size 500ml	150.000 đ	30	Còn hàng	
	Ống Hút Inox (Set 4 cái) Ống hút inox 304 thân thiện môi trường	Phụ kiện	Phụ kiện	Size Set	25.000 đ	100	Còn hàng	

Hình 25: Quản lý Sản phẩm

Trang Quản lý Sản phẩm dùng để quản lý danh sách các sản phẩm, Admin có thể Thêm sản phẩm, sửa sản phẩm có sẵn hoặc xóa sản phẩm đó đi, tiện lợi cho việc theo dõi những mặt hàng hiện đang có trong trang web của mình.

3.3.3 Thêm Sản phẩm:

+

Thêm Đồ Uống Mới

Tên đồ uống *

VD: Trà Đào Cam Sả

Danh mục *

VD: Cà phê, Trà, Nước ép

Loại *

VD: Trà trái cây

Size *

-- Chọn size --

Giá (VNĐ) *

VD: 45000

Số lượng *

VD: 100

Mô tả

Mô tả chi tiết về đồ uống...

Hình ảnh *

☒

Nhập URL ảnh

https://example.com/image.jpg hoặc /images/product.jpg

URL từ internet hoặc đường dẫn local (VD: /images/ca-phe.jpg)

☐

Hoặc upload ảnh từ máy tính

Choose File

No file chosen

Chức năng upload đang phát triển

← Quay lại danh sách

✓

Thêm đồ uống


Hình 26: Thêm Sản phẩm

Quản trị viên thêm Sản phẩm bằng cách nhập các thông tin cần thiết như là tên đồ uống,danh mục,loại, size, giá, số lượng, mô tả, hình ảnh,vvv.

Lập trình Web 2

55

3.3.4.Chỉnh sửa Sản phẩm:

 **Sửa Đồ Uống**

Tên đồ uống *

Trà Sữa Trân Châu

Danh mục *

Trà

Loại *

Trà sữa

Size *

Size M

Giá (VNĐ) *

45000.00

Số lượng *

49

Mô tả

Trà sữa trân châu đường đen thơm ngon, ngọt dịu

Hình ảnh

☐

☒ **Nhập URL ảnh mới**

/images/tra-sua-tran-chau.jpg

(i) Để trống nếu giữ nguyên ảnh cũ

☐ **Hoặc upload ảnh mới từ máy tính**

Choose File No file chosen

(i) Chức năng upload đang phát triển

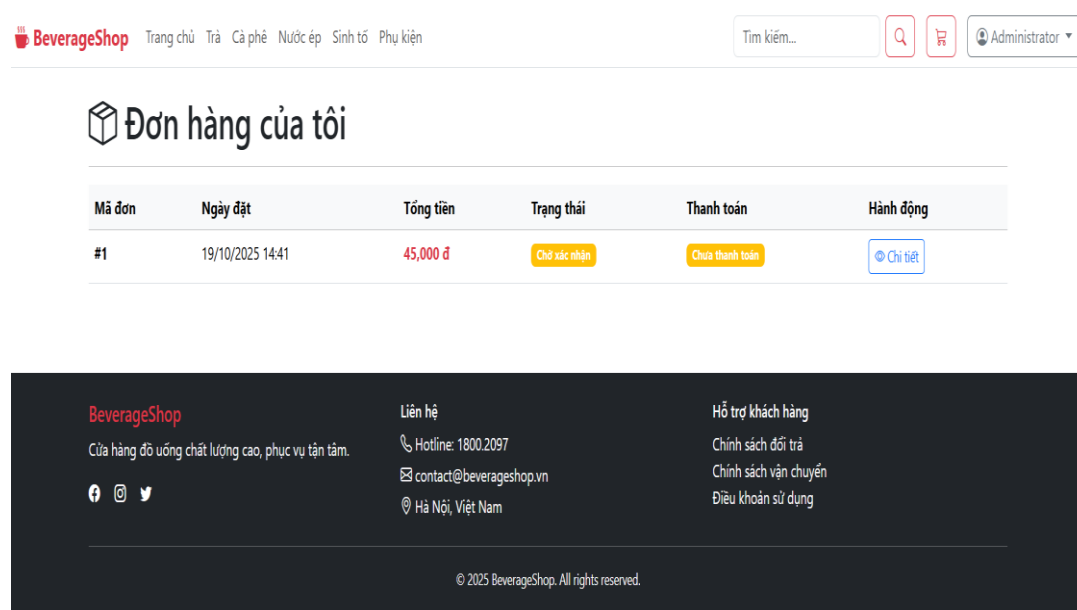
☒ **Còn hàng**

← Quay lại

Cập nhật

Hình 27: Chỉnh sửa Sản phẩm

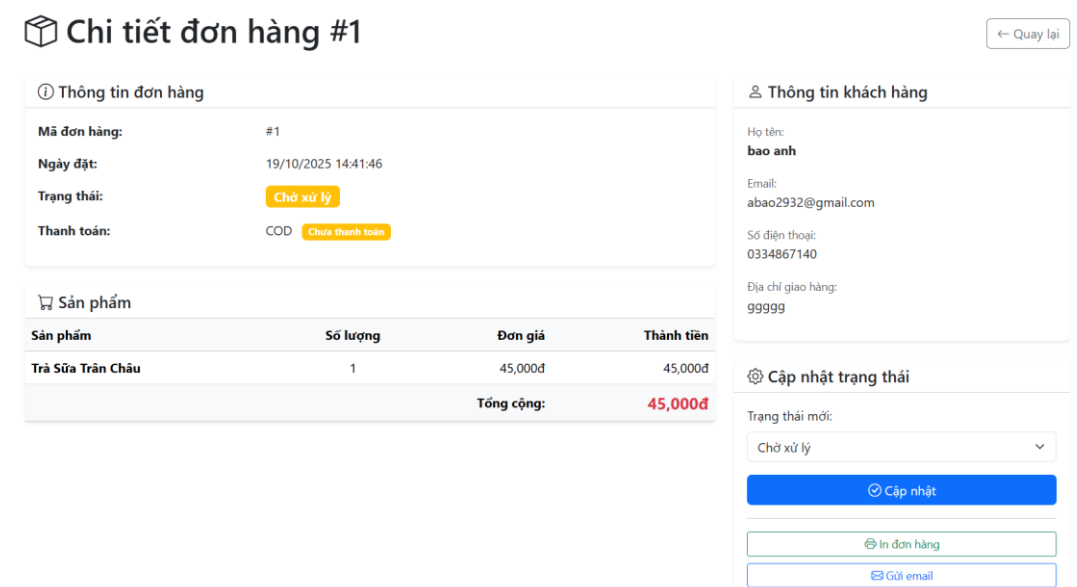
3.3.5. Quản lý Đơn hàng:



Hình 28: Quản lý Đơn hàng

Đây là danh sách đơn hàng đã được đặt. Bảng cũng hiển thị “Ngày đặt”, ”Trạng thái” để Admin dễ dàng theo dõi và quản lý.

3.3.6. Chi tiết Đơn hàng:



Hình 29: Chi tiết Đơn hàng

Chi tiết các sản phẩm mà người dùng đã mua, thông tin khách hàng hàng, admin có thể chỉnh sửa trạng thái và in hóa đơn đơn hàng.

3.3.7. Quản lý Người dùng:

The screenshot displays the 'Quản Lý Người Dùng' (User Management) page of the BeverageShop application. The page features a header with contact information and navigation links. The main content area includes a table listing users and a summary bar.

ID	Họ Tên	Email	Số Điện Thoại	Địa Chỉ	Vai Trò	Trạng Thái	Thao Tác
#1	Administrator ★ Hệ thống	admin@beverageshop.com @admin	0123456789	Hà Nội	Admin	Hoạt động	 Bảo vệ

Tổng cộng: 1 người dùng | Hoạt động: 1 | Admin: 1

The footer section contains contact details for BeverageShop, including a hotline, email, and address, along with social media links and a copyright notice for 2025.

Hình 30: Quản lý Người dùng

Trang Quản lý Người dùng hiển thị tất cả người dùng đã Đăng ký tài khoản trong hệ thống, Admin có thể chỉnh sửa quyền của người dùng và cập nhật vai trò khi cần thiết.

3.3.8.Thêm người dùng

Thêm Người Dùng Mới

Tên đăng nhập *

admin

Email *

VD: user@example.com

Mật khẩu *

.....

Mật khẩu phải có ít nhất 6 ký tự

Họ và tên *

VD: Nguyễn Văn A

Số điện thoại

VD: 0123456789

Vai trò *

Khách hàng

Địa chỉ

VD: 123 Đường ABC, Quận XYZ, Hà Nội

ⓘ Lưu ý:

Tài khoản mới sẽ được tạo với trạng thái "Hoạt động" mặc định.

← Quay lại danh sách

✓ Tạo tài khoản

Hình 31: Thêm Người dùng

3.3.9.Chỉnh sửa Người dùng:

Xem Thông Tin Tài Khoản Hệ Thống

Tài khoản hệ thống: Không thể thay đổi vai trò, trạng thái hoặc xóa tài khoản này. Chỉ có thể cập nhật thông tin cá nhân.

Tên đăng nhập

Email *

admin

admin@beverageshop.com

Không thể thay đổi tên đăng nhập

Họ và tên *

Administrator

Số điện thoại

Vai trò *

0123456789

Admin

Không thể thay đổi vai trò tài khoản hệ thống

Địa chỉ

Hà Nội

☒

Tài khoản hoạt động

Tài khoản hệ thống luôn hoạt động

← Quay lại danh sách

Cập nhật thông tin

Hình 32: Chỉnh sửa Người dùng

3.3.10.Quản lý Brands:

#	Tên Thương hiệu	Số sản phẩm	Tổng tồn kho	Thao tác
1	Cà Phê Việt	1 sản phẩm	100	Chi tiết Thêm Xóa
2	Cozy	1 sản phẩm	120	Chi tiết Thêm Xóa
3	Dilmah	1 sản phẩm	50	Chi tiết Thêm Xóa
4	Fresh Juice	1 sản phẩm	30	Chi tiết Thêm Xóa
5	Highlands	2 sản phẩm	105	Chi tiết Thêm Xóa
6	Lavazza	1 sản phẩm	35	Chi tiết Thêm Xóa
7	Lipton	1 sản phẩm	70	Chi tiết Thêm Xóa
8	Minute Maid	1 sản phẩm	110	Chi tiết Thêm Xóa
9	Nescafe	2 sản phẩm	135	Chi tiết Thêm Xóa
10	Oolong Tea	1 sản phẩm	30	Chi tiết Thêm Xóa
11	Phúc Long	1 sản phẩm	45	Chi tiết Thêm Xóa
12	Smoothie House	1 sản phẩm	25	Chi tiết Thêm Xóa
13	Soda Fresh	1 sản phẩm	60	Chi tiết Thêm Xóa
14	Starbucks	1 sản phẩm	40	Chi tiết Thêm Xóa
15	Trà Việt	2 sản phẩm	89	Chi tiết Thêm Xóa
16	Tropicana	1 sản phẩm	90	Chi tiết Thêm Xóa
17	Trung Nguyên	2 sản phẩm	150	Chi tiết Thêm Xóa

Hình 33: Quản lý Brands.

3.3.11.Thêm Brands

Thông tin Thương hiệu

Tên Thương hiệu *

VD: Starbucks, Nescafe...

Nhập tên thương hiệu duy nhất

Chọn sản phẩm để gắn thương hiệu *

☐

Nescafe Gold

Nescafe

☐

Lavazza Premium

Lavazza

☐

Starbucks Pike Place

Starbucks

☐

Lipton Yellow Label

Lipton

☐

Cozy Trà Sữa

Cozy

☐

Dilmah Earl Grey

Dilmah

☐

Phúc Long Oolong

Phúc Long

☐

Trà Oolong Cao Cấp

Oolong Tea

☐

Tropicana Cam

Tropicana

☐

Minute Maid Pulp

Minute Maid

☐

Twister Mix Berry

Twister

☐

Vinamilk Sữa Chua Uống

Vinamilk

Cà phê

Cà phê

Cà phê

Trà

Trà

Trà

Trà

Nước ép

Nước ép

Nước ép

Sinh tố

Chọn ít nhất 1 sản phẩm. Có thể chọn nhiều sản phẩm cùng lúc.

Hủy

Tạo Thương hiệu

Hình 34: Thêm Brands

KẾT LUẬN

Đề tài "Xây dựng hệ thống quản lý cửa hàng đồ uống với kiến trúc Backend API và Frontend MVC" đã được hoàn thành với những kết quả cụ thể. Quá trình thực hiện đã góp phần vận dụng hiệu quả kiến thức từ lý thuyết vào thực tiễn, bao gồm việc thiết kế API, xây dựng giao diện người dùng, quản lý cơ sở dữ liệu và xử lý các tình huống phát sinh. Qua đó, chúng tôi đã nâng cao kỹ năng lập trình full-stack và hiểu rõ hơn về quy trình phát triển một hệ thống hoàn chỉnh, tách biệt rõ ràng giữa logic và giao diện.

1) Tổng kết kiến thức đạt được:

- Về lý thuyết: Các kiến thức nền tảng về một hệ thống hiện đại đã được củng cố. Kiến trúc tách biệt giữa Backend (ASP.NET Core Web API) và Frontend (ASP.NET Core MVC) được áp dụng triệt để. Theo đó, Backend chịu trách nhiệm xử lý toàn bộ logic nghiệp vụ và cung cấp dữ liệu thông qua các endpoint, trong khi Frontend đóng vai trò là một "người tiêu dùng" (consumer), gọi đến các API này để hiển thị thông tin và tương tác với người dùng. Mô hình này giúp nâng cao tính tổ chức, bảo mật và khả năng mở rộng của hệ thống.
- Về thực tế:
 - Backend API: Đã triển khai đầy đủ các endpoint cho các chức năng chính như xác thực (đăng ký, đăng nhập), quản lý sản phẩm, xử lý đơn hàng, quản lý người dùng, quản lý thương hiệu, tạo mã giảm giá.
 - Frontend MVC: Đã xây dựng hoàn chỉnh giao diện cho cả người dùng và quản trị viên, các chức năng này đều hoạt động dựa trên việc gọi và xử lý dữ liệu từ Backend API.
 - Trang người dùng: Đầy đủ các tính năng như đăng ký, đăng nhập, xem/tìm kiếm, lọc sản phẩm, quản lý giỏ hàng và đặt hàng.
 - Trang quản trị: Cung cấp các công cụ cần thiết để quản lý toàn bộ hệ thống như quản lý sản phẩm, đơn hàng, người dùng và các tác vụ cơ bản.
 - Các chức năng phân quyền và kiểm soát truy cập được đảm bảo, giúp hệ thống hoạt động an toàn và đúng với vai trò của từng đối tượng.

2) Khuyết điểm:

- **Giao diện:** Một vài trang ở cả khu vực người dùng và trang quản trị vẫn chưa được bố trí hợp lý, thiết kế còn đơn giản, chưa thực sự tối ưu về mặt thẩm mỹ và trải nghiệm người dùng.
- **Chức năng còn thiếu:** Phía API chưa có các endpoint chuyên dụng cho việc thống kê, báo cáo dữ liệu phức tạp. Do đó, trang quản trị ở Frontend chưa thể tích hợp chức năng in báo cáo, ảnh hưởng đến khả năng theo dõi tổng quan hệ thống.
- **Hiệu suất và bảo mật:** Hệ thống hiện tại vẫn còn nhiều điểm có thể được tối ưu hơn để tăng tốc độ xử lý của API. Đồng thời, cần bổ sung các biện pháp bảo mật nâng cao (như sử dụng JWT - JSON Web Tokens) để đảm bảo an toàn thông tin khi triển khai thực tế.
- **Kinh nghiệm còn hạn chế:** Vì được xây dựng chủ yếu dựa trên quá trình tự nghiên cứu, nên khả năng phân tích, tối ưu hệ thống và xử lý các tình huống phức tạp trong môi trường thực tế vẫn còn nhiều hạn chế.

3) Hướng phát triển của đề tài:

- **Cải thiện UI/UX:** Nâng cấp toàn bộ giao diện phía Frontend (MVC), chú trọng vào việc sắp xếp bố cục hợp lý và cải thiện trải nghiệm tương tác để tạo cảm giác thân thiện và dễ sử dụng hơn.
- **Bổ sung chức năng:**
- **Backend:** Phát triển thêm các endpoint API cho chức năng báo cáo, thống kê doanh thu theo thời gian, lọc tìm kiếm nâng cao.
- **Frontend:** Xây dựng các giao diện tương ứng trên trang quản trị để hiển thị các báo cáo này một cách trực quan.
- **Tối ưu hóa kỹ thuật:**
- **Backend:** Cải thiện hiệu suất xử lý của API bằng cách tối ưu truy vấn cơ sở dữ liệu, áp dụng caching. Tăng cường bảo mật bằng việc chuyển sang cơ chế xác thực sử dụng JWT.
- **Frontend:** Tối ưu tốc độ tải trang và tương tác của người dùng.
- **Nghiên cứu và ứng dụng công nghệ mới:** Tiếp tục cập nhật các xu hướng mới, ví dụ như xây dựng Frontend bằng một framework JavaScript hiện đại (React, VueJS) để tạo ra trải nghiệm người dùng phong phú hơn (Single Page Application).

TÀI LIỆU THAM KHẢO

- [1] Tài liệu giảng dạy môn Lập trình Web API – Giảng viên Nguyễn Đức Tấn – Trường Đại học YERSIN Đà Lạt – 2025.
- [2] Adam Freeman (2023). *Pro ASP.NET Core 8, Tenth Edition*. Manning Publications.
- [3] Andrew Lock (2023). *ASP.NET Core in Action, Third Edition*. Manning Publications.
- [4] Microsoft Docs. *Tài liệu chính thức về ASP.NET Core*. Lấy từ: <https://docs.microsoft.com/en-us/aspnet/core/>
- [5] Tieg Phan (2018). *Tìm hiểu kiến thức cơ bản và nâng cao về API và ASP.NET Core API*. Viblo. Lấy từ: <https://viblo.asia/p/tim-hieu-kiem-thuc-co-ban-ve-api>
- [6] Source code Hệ thống API quản lý đồ uống bằng ASP.net Core 2025. Lấy từ: [urikasorey/BeverageShop](https://github.com/urikasorey/BeverageShop) và : <https://github.com/nguyenanhbao123/BaocaoLTWEB2.git>