

**ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN**

**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**  
**CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM**

**ĐỀ TÀI:**

**HỆ THỐNG DINH DƯỠNG THÔNG MINH HỖ  
TRỢ TẠO THỰC ĐƠN CHO CHÓ & MÈO**

Người hướng dẫn: **TS. LÊ MINH TRÍ, TS. NINH KHÁNH DUY**

Sinh viên thực hiện: **NGUYỄN PHAN BẢO LỘC**

Số thẻ sinh viên: **102210115**

Lớp: **21T-DT2**

**Đà Nẵng, 01/2026**

ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA CÔNG NGHỆ THÔNG TIN

**ĐỒ ÁN TỐT NGHIỆP**  
NGÀNH: CÔNG NGHỆ THÔNG TIN  
CHUYÊN NGÀNH: CÔNG NGHỆ PHẦN MỀM

ĐỀ TÀI:

**HỆ THỐNG DINH DƯỠNG THÔNG MINH HỖ  
TRỢ TẠO THỰC ĐƠN CHO CHÓ & MÈO**

Người hướng dẫn: TS. LÊ MINH TRÍ, TS. NINH KHÁNH DUY  
Sinh viên thực hiện: NGUYỄN PHAN BẢO LỘC  
Số thẻ sinh viên: 102210115  
Lớp: 21T\_DT2

Đà Nẵng, 01/2026

# TÓM TẮT

Tên đề tài: Hệ thống dinh dưỡng thông minh hỗ trợ tạo thực đơn cho chó & mèo.

Sinh viên thực hiện: Nguyễn Phan Bảo Lộc

Số thẻ SV: 102210115

Lớp: 21T\_DT2

Tóm tắt nội dung đồ án

Đồ án tập trung nghiên cứu và xây dựng một hệ thống hỗ trợ chăm sóc dinh dưỡng thú cưng thông minh cho chó và mèo, trong đó ứng dụng trí tuệ nhân tạo nhằm nâng cao khả năng tư vấn và cá nhân hóa trải nghiệm người dùng. Hệ thống cho phép người dùng tương tác thông qua giao diện web để nhận tư vấn, phân tích và lập thực đơn dinh dưỡng phù hợp với đặc điểm và mục tiêu chăm sóc của từng thú cưng.

## NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Họ tên sinh viên: Nguyễn Phan Bảo Lộc Số thẻ sinh viên: 102210115  
Lớp: 21T\_DT2 Khoa: Công nghệ thông tin Ngành: Công nghệ phần mềm

1. Tên đề tài đồ án: *Hệ thống dinh dưỡng thông minh hỗ trợ tạo thực đơn cho chó & mèo.*

2. Đề tài thuộc diện:  Có ký kết thỏa thuận sở hữu trí tuệ đối với kết quả thực hiện

3. Các số liệu và dữ liệu ban đầu:

.....  
.....  
.....

4. Nội dung các phần thuyết minh và tính toán:

Nội dung của phần thuyết minh bao gồm các phần chính sau:

**Mở đầu:** Trình bày nhu cầu thực tế và lý do lựa chọn đề tài, đồng thời giới thiệu về nội dung nghiên cứu và các mục tiêu cần đạt được.

**Chương 1:** Cơ sở lý thuyết: trình bày các khái niệm, kiến thức nền tảng và các tài liệu liên quan làm cơ sở cho việc xây dựng và phát triển đề tài.

**Chương 2:** Phân tích và thiết kế: thực hiện phân tích yêu cầu hệ thống, xác định các chức năng chính, thiết kế mô hình hệ thống, cơ sở dữ liệu và các thành phần liên quan.

**Chương 3:** Triển khai và đánh giá kết quả: trình bày quá trình triển khai hệ thống, mô tả các chức năng đã xây dựng, đồng thời thực hiện kiểm thử và đánh giá kết quả đạt được so với mục tiêu đề ra.

5. Các bản vẽ, đồ thị ( ghi rõ các loại và kích thước bản vẽ ):

.....  
.....  
.....  
.....

6. Họ tên người hướng dẫn: TS. Lê Minh Trí, TS. Ninh Khánh Duy

7. Ngày giao nhiệm vụ đồ án: ...../...../202.....

8. Ngày hoàn thành đồ án: ...../...../202.....

Đà Nẵng, ngày tháng năm 202

Trưởng Bộ môn .....

Người hướng dẫn

## LỜI NÓI ĐẦU

Trong thời gian thực hiện đồ án, với sự giúp đỡ và tạo điều kiện của trường Đại học Bách Khoa – Đại học Đà Nẵng, em đã hoàn thành đồ án và bản báo cáo đúng thời hạn quy định. Tuy nhiên, với khả năng và thời gian có hạn, không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự quan tâm, giúp đỡ, và góp ý từ quý Thầy Cô để em có thể hoàn thiện đề tài nghiên cứu trong thời gian tới.

Em xin chân thành cảm ơn thầy TS. Lê Minh Trí và thầy TS. Ninh Khánh Duy – giảng viên khoa Công nghệ thông tin – Trường Đại học Bách Khoa Đà Nẵng đã tận tình hướng dẫn và giúp đỡ em trong suốt quá trình triển khai đồ án tốt nghiệp, từ hướng dẫn chọn đề tài đến lời khuyên để triển khai, giải quyết vấn đề.

Mặc dù đã cố gắng hoàn thiện đồ án trong phạm vi và khả năng có thể, nhưng khó tránh khỏi những thiếu sót. Em rất mong nhận được sự đóng góp, nhận xét và đánh giá của quý Thầy Cô.

## **CAM ĐOAN**

Em – Nguyễn Phan Bảo Lộc, sinh viên năm 4 Trường Đại học Bách khoa – Đại học Đà Nẵng, xin cam đoan đồ án tốt nghiệp này được thực hiện trên cơ sở nghiên cứu độc lập và nghiêm túc. Toàn bộ nội dung trình bày trong đồ án là kết quả của quá trình học tập, tìm hiểu và thực hiện của bản thân, không sao chép từ bất kỳ nguồn tài liệu nào khác dưới mọi hình thức.

Các dữ liệu, kết quả nghiên cứu và nội dung tham khảo trong đồ án đều được sử dụng có chọn lọc, trích dẫn đầy đủ và tuân thủ các quy định về liêm chính học thuật của trường Đại học Bách khoa Đà Nẵng.

Em hoàn toàn chịu trách nhiệm về tính chính xác, trung thực của các thông tin được trình bày trong đồ án tốt nghiệp này.

Sinh viên thực hiện

# MỤC LỤC

Tóm tắt	
Nhiệm vụ đồ án	
Lời nói đầu và cảm ơn	i
Lời cam đoan liên chính học thuật	ii
Mục lục	iii
Danh sách các hình vẽ, bảng	v
Danh sách các cụm từ viết tắt	viii
<b>MỞ ĐẦU</b> .....	<b>1</b>
<b>Chương 1: CƠ SỞ LÝ THUYẾT</b> .....	<b>3</b>
1.1. Mô hình Client – Server .....	3
1.2. Công nghệ sử dụng .....	4
1.2.1. Tổng quan về framework Nestjs.....	4
1.2.2. Tổng quan về framework Nextjs .....	4
1.2.3. Fastapi.....	5
1.2.4. Websocket.....	5
1.3. Lưu trữ & quản lý dữ liệu.....	5
1.3.1. Tổng quan về Postgresql .....	5
1.3.2. Tổng quan về S3 Storage.....	6
1.4. AI Agent.....	6
1.4.1. Tổng quan .....	6
1.4.2. Mô hình ngôn ngữ lớn (Large Language Model – LLM).....	6
1.4.3. Kỹ thuật Retrieval – Augmented Generation (RAG).....	7
1.4.4. Text Embeddings .....	8
1.4.5. Prompt Engineering.....	8
1.5. Triển khai & vận hành hệ thống .....	8
1.5.1. Docker .....	8
1.5.2. CICD .....	9
1.6. Dữ liệu nghiên cứu và tham khảo .....	9
<b>Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG</b> .....	<b>11</b>
2.1. Mô tả bài toán .....	11
2.2. Phân tích yêu cầu hệ thống .....	11

2.2.1.	Yêu cầu chức năng .....	11
2.2.2.	Yêu cầu phi chức năng.....	12
2.3.	Phân tích thiết kế hệ thống .....	13
2.3.1.	Biểu đồ Use case tổng quát.....	13
2.3.2.	Biểu đồ hoạt động.....	22
2.3.3.	Biểu đồ tuần tự.....	27
2.3.4.	Phân tích thiết kế cơ sở dữ liệu .....	30
2.4.	Ứng dụng LLM triển khai AI Agent.....	36
2.4.1.	Cơ sở nghiên cứu và tham khảo .....	36
2.4.2.	Mô hình ngôn ngữ lớn (LLM) sử dụng trong hệ thống.....	37
2.4.3.	Tiền xử lý dữ liệu bằng Chunking.....	38
2.4.4.	Vector hóa nội dung bằng Embedding.....	38
2.4.5.	Truy xuất và sinh phản hồi với mô hình RAG.....	38
<b>Chương 3: TRIỂN KHAI THỰC TẾ VÀ ĐÁNH GIÁ KẾT QUẢ .....</b>		<b>40</b>
3.1.	Tổ chức triển khai hệ thống.....	40
3.1.1.	Công cụ triển khai.....	40
3.1.2.	Triển khai hệ thống.....	40
3.1.3.	Quản lí mã nguồn (github).....	41
3.2.	Kết quả triển khai chức năng AI.....	42
3.2.1.	Chức năng chatbot.....	42
3.2.2.	Chức năng Tạo thực đơn dinh dưỡng.....	44
3.3.	Đặc tả API .....	46
3.4.	Giao diện website và chức năng .....	48
3.5.	Kiểm thử phần mềm.....	60
<b>Kết luận.....</b>		<b>64</b>
<b>TÀI LIỆU THAM KHẢO.....</b>		<b>66</b>

## DANH SÁCH CÁC BẢNG

Bảng 2.1 Các tác nhân của hệ thống.....	13
Bảng 2.2 Mô tả use case Thêm mới, cập nhật thông tin thú cưng.....	15
Bảng 2.3 Mô tả use case Mua sắm sản phẩm .....	16
Bảng 2.4 Mô tả use case Lập thực đơn dinh dưỡng .....	17
Bảng 2.5 Mô tả use case Chatbot tư vấn dinh dưỡng.....	18
Bảng 2.6 Mô tả use case Quản lý người dùng .....	19
Bảng 2.7 Mô tả use case Quản lý đơn hàng.....	20
Bảng 2.8 Mô tả use case Quản lý wiki dinh dưỡng.....	21
Bảng 2.9 Danh sách các bảng dữ liệu.....	31
Bảng 2.10 Bảng users .....	31
Bảng 2.11 Bảng pets .....	32
Bảng 2.12 Bảng products.....	33
Bảng 2.13 Bảng plans .....	34
Bảng 2.14 Bảng wiki_pages .....	35
Bảng 2.15 Bảng wiki_page_versions .....	35
Bảng 2.16 Bảng wiki_releases.....	35
Bảng 2.17 Bảng wiki_page_releases .....	36
Bảng 2.18 Bảng chat_sessions.....	36
Bảng 2.19 Bảng chat_messages.....	36

## DANH SÁCH CÁC HÌNH VẼ

Hình 1.1 Mô hình Client - Server .....	3
Hình 1.2 Mô hình hoạt động của Retrieval-Augmented Generation (RAG).....	7
Hình 2.1 Use case tổng quát người dùng.....	14
Hình 2.2 Use case tổng quát admin .....	14
Hình 2.3 Use case Thêm mới, cập nhật thông tin thú cưng.....	15
Hình 2.4 Use case Mua sắm sản phẩm .....	16
Hình 2.5 Use case Lập thực đơn dinh dưỡng thông minh .....	17
Hình 2.6 Use case Chatbot tư vấn dinh dưỡng .....	18
Hình 2.7 Use case quản lý người dùng.....	19
Hình 2.8 Use case quản lý đơn hàng .....	20
Hình 2.9 Use case quản lý wiki dinh dưỡng.....	21
Hình 2.10 Biểu đồ hoạt động chức năng "Đăng kí" .....	22
Hình 2.11 Biểu đồ hoạt động chức năng "Tạo mới hồ sơ cho thú cưng" .....	23
Hình 2.12 Biểu đồ hoạt động chức năng "Quản lý người dùng" .....	24
Hình 2.13 Biểu đồ hoạt động chức năng "Tạo mới thực đơn" .....	25
Hình 2.14 Biểu đồ hoạt động chức năng "Chatbot" .....	26
Hình 2.15 Biểu đồ tuần tự chức năng "Đăng nhập" .....	27
Hình 2.16 Biểu đồ tuần tự chức năng "Tạo thực đơn" .....	28
Hình 2.17 Biểu đồ tuần tự chức năng "Đánh giá và phản hồi thực đơn" .....	29
Hình 2.18 Biểu đồ tuần tự chức năng "Chatbot" .....	30
Hình 2.19 Cơ sở dữ liệu.....	30
Hình 3.1 Mô hình triển khai hệ thống.....	40
Hình 3.2 Quản lý mã nguồn bằng Github.....	41
Hình 3.3 Sơ đồ triển khai chức năng Chatbot.....	42
Hình 3.4 Sơ đồ triển khai chức năng Tạo thực đơn dinh dưỡng.....	44
Hình 3.5 Danh sách api module auth, user và admin .....	46
Hình 3.6 Danh sách api module orders và carts .....	46
Hình 3.7 Danh sách api module products, payment .....	47
Hình 3.8 Danh sách api module pets và wiki .....	47
Hình 3.9 Màn hình login.....	48
Hình 3.10 Màn hình đăng ký .....	48
Hình 3.11 Màn hình trang chủ .....	49
Hình 3.12 Màn hình cửa hàng thực phẩm .....	50
Hình 3.13 Màn hình wiki dinh dưỡng .....	51

Hình 3.14 Màn hình chatbot .....	51
Hình 3.15 Màn hình profile người dùng.....	52
Hình 3.16 Màn hình quản lý thú cưng của người dùng .....	52
Hình 3.17 Màn hình tạo mới thực đơn.....	53
Hình 3.18 Màn hình chi tiết thực đơn và đánh giá .....	54
Hình 3.19 Màn hình chi tiết thực đơn và đánh giá .....	55
Hình 3.20 Màn hình mua hàng và thanh toán.....	56
Hình 3.21 Màn hình dashboard của admin .....	57
Hình 3.22 Màn hình quản lý người dùng.....	57
Hình 3.23 Màn hình quản lý wiki dinh dưỡng.....	58
Hình 3.24 Màn hình thêm mới wiki .....	58
Hình 3.25 Màn hình xem chi tiết và cập nhật wiki.....	59
Hình 3.26 Màn hình quản lý sản phẩm.....	59
Hình 3.27 Màn hình quản lý đơn hàng .....	60

## DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

KÝ HIỆU:

STT	Kí hiệu	Ý Nghĩa
1	API	Application Programming Interface
2	RAG	Retrieval-Augmented Generation
3	LLM	Large Language Model
4	JSON	Java Object Notation

## MỞ ĐẦU

### 1. Mục đích thực hiện đề tài

Xu hướng nuôi thú cưng tăng nhanh dẫn đến nhu cầu chăm sóc dinh dưỡng cho chó và mèo ngày càng lớn. Người nuôi không chỉ quan tâm đến sức khỏe thú cưng mà còn xem chúng như một thành viên trong gia đình, từ đó đòi hỏi một chế độ ăn khoa học, an toàn và phù hợp. Tuy nhiên, phần lớn người nuôi vẫn lựa chọn thức ăn dựa trên kinh nghiệm cá nhân, dễ dẫn đến tình trạng thiếu chất, thừa chất hoặc mất cân đối dinh dưỡng.

Trong khi đó, thị trường hiện nay chưa đáp ứng được nhu cầu tính toán năng lượng, cá nhân hóa khẩu phần và xây dựng thực đơn phù hợp cho từng thú cưng. Với sự phát triển mạnh mẽ của Trí tuệ nhân tạo (AI), đặc biệt là các mô hình ngôn ngữ lớn (LLM), việc ứng dụng AI để phân tích đặc điểm cá thể, tự động sinh khẩu phần ăn và tư vấn dinh dưỡng đã trở nên khả thi và hiệu quả.

Từ những lý do trên, em quyết định chọn đề tài “Hệ thống dinh dưỡng thông minh hỗ trợ tạo thực đơn cho chó & mèo” nhằm đáp ứng nhu cầu chăm sóc sức khỏe thú cưng một cách chính xác, tiện lợi và hiệu quả.

### 2. Mục tiêu đề tài

Xây dựng một hệ thống dinh dưỡng thông minh hỗ trợ người nuôi chó và mèo lựa chọn khẩu phần ăn hợp lý, khoa học và phù hợp với từng cá thể thú cưng.

Tự động tạo thực đơn hằng ngày dựa trên các thông tin đặc điểm như giống loài, cân nặng, độ tuổi, mức độ hoạt động và tình trạng sức khỏe.

Tích hợp chức năng gợi ý và bán sản phẩm thực phẩm dành cho thú cưng theo đúng khẩu phần và nhu cầu đã được tính toán.

Mang lại giải pháp hỗ trợ chăm sóc thú cưng tiện lợi, chính xác và có tính ứng dụng cao trong thực tế.

### 3. Phạm vi và đối tượng nghiên cứu

#### 3.1. Phạm vi đề tài

Đề tài tập trung nghiên cứu và tham khảo các tiêu chuẩn dinh dưỡng cơ bản dành cho chó và mèo, dựa trên các khuyến nghị và hướng dẫn của FEDIAF (nguồn cơ sở chính), ngoài ra có tham khảo bổ sung từ NRC, và một số tài liệu tham khảo về cách tính cân nặng, dinh dưỡng khác nhằm xây dựng cơ sở dữ liệu phục vụ cho việc phân tích và đề xuất khẩu phần ăn phù hợp trong hệ thống. Phạm vi nghiên cứu của đề tài chủ yếu giới hạn ở hai đối tượng chính là chó và mèo.

### 3.2. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là người dùng có nhu cầu xây dựng khẩu phần ăn khoa học cho chó hoặc mèo, đồng thời có nhu cầu lựa chọn và mua các sản phẩm phù hợp thông qua hệ thống.

## 4. Phương pháp nghiên cứu

Trong giai đoạn đầu, tiến hành khảo sát và phân tích nhu cầu thực tế của người nuôi thú cưng, qua đó nhận diện các khó khăn trong việc xây dựng khẩu phần ăn khoa học cho chó và mèo do hạn chế về kiến thức chuyên môn và thiếu nguồn thông tin đáng tin cậy. Từ kết quả phân tích, các yêu cầu chức năng và phi chức năng của hệ thống được xác định nhằm giải quyết bài toán đặt ra.

Tiếp theo, thực hiện nghiên cứu và tham khảo các tài liệu, tiêu chuẩn và hướng dẫn về dinh dưỡng cho thú cưng, tiêu biểu là các khuyến nghị của FEDIAF và các nguồn tài liệu chuyên môn uy tín khác, làm cơ sở xây dựng mô hình dinh dưỡng và các nguyên tắc tính toán khẩu phần ăn.

Trên nền tảng các yêu cầu và cơ sở lý thuyết đã xác định, dữ liệu dinh dưỡng được chuẩn hóa, cơ sở dữ liệu thực phẩm được xây dựng và các mô-đun tính toán, đề xuất khẩu phần ăn phù hợp với từng cá thể thú cưng dựa trên thông tin đầu vào được phát triển.

Cuối cùng, hệ thống được triển khai ở mức độ mô phỏng và thử nghiệm, kết hợp kiểm thử và đánh giá nhằm xác định mức độ chính xác, tính hiệu quả và khả năng đáp ứng các yêu cầu đã đề ra.

## 5. Cấu trúc của đề án tốt nghiệp

Đề án bao gồm các nội dung:

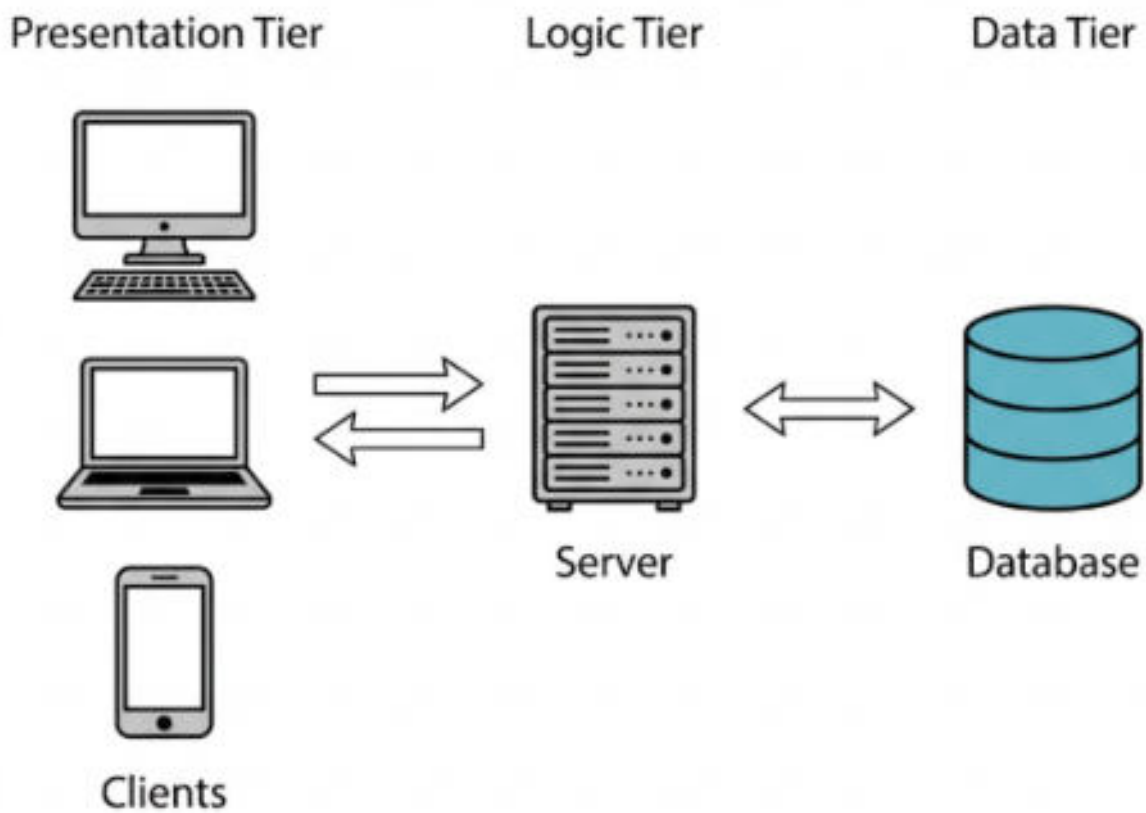
- Mở đầu
- Chương 1: Cơ sở lý thuyết
- Chương 2: Phân tích và thiết kế hệ thống
- Chương 3: Triển khai thực tế và đánh giá kết quả
- Kết luận và hướng phát triển

## Chương 1: CƠ SỞ LÝ THUYẾT

### 1.1. Mô hình Client – Server

Mô hình Client – Server là một kiến trúc hệ thống phổ biến trong các ứng dụng web, trong đó hệ thống được chia thành hai thành phần chính là Client và Server. Client là phía giao diện người dùng, có nhiệm vụ gửi yêu cầu và hiển thị kết quả, trong khi Server chịu trách nhiệm xử lý nghiệp vụ, quản lý dữ liệu và phản hồi lại các yêu cầu từ Client.

Trong mô hình này, Client là phía giao diện, chịu trách nhiệm hiển thị nội dung và tiếp nhận các thao tác từ người dùng, sau đó gửi các yêu cầu đến Server thông qua các giao thức mạng như HTTP/HTTPS hoặc WebSocket. Server có nhiệm vụ tiếp nhận và xử lý các yêu cầu từ Client, thực hiện logic nghiệp vụ, truy xuất và quản lý dữ liệu từ cơ sở dữ liệu, đồng thời phản hồi kết quả xử lý về cho Client. Quá trình trao đổi dữ liệu này giúp đảm bảo hệ thống hoạt động ổn định và nhất quán.



Hình 1.1 Mô hình Client - Server

Việc phân tách rõ ràng giữa Client và Server giúp hệ thống dễ dàng mở rộng, bảo trì và nâng cấp. Client có thể được thay đổi hoặc phát triển độc lập mà không ảnh hưởng trực tiếp đến Server, trong khi Server tập trung xử lý dữ liệu và nghiệp vụ, giúp tăng cường khả năng kiểm soát, bảo mật và tính nhất quán của hệ thống.

Trong phạm vi đề tài, mô hình Client – Server được áp dụng làm nền tảng cho kiến trúc hệ thống, trong đó Client đảm nhiệm việc tương tác với người dùng thông qua giao diện web, còn Server thực hiện xử lý nghiệp vụ, tính toán, quản lý dữ liệu và tích hợp các thành phần công nghệ khác của hệ thống.

## **1.2. Công nghệ sử dụng**

### **1.2.1. Tổng quan về framework Nestjs**

NestJS là một framework backend phát triển trên nền tảng Node.js, được xây dựng dựa trên ngôn ngữ TypeScript. NestJS kết hợp nhiều mô hình lập trình như lập trình hướng đối tượng (Object-Oriented Programming – OOP), lập trình hàm (Functional Programming – FP) và lập trình phản ứng hàm (Functional Reactive Programming – FRP). Framework này cung cấp một cấu trúc ứng dụng rõ ràng theo hướng module hóa, đồng thời hỗ trợ mạnh mẽ cho lập trình hướng đối tượng và cơ chế Dependency Injection. Nhờ đó, NestJS giúp tổ chức mã nguồn một cách khoa học, nâng cao khả năng mở rộng và thuận tiện trong quá trình bảo trì hệ thống.

Bên cạnh đó, NestJS cho phép xây dựng các API theo mô hình RESTful và hỗ trợ tích hợp với nhiều thư viện, công nghệ phổ biến như ORM, middleware, xác thực và phân quyền người dùng. Với kiến trúc linh hoạt và khả năng mở rộng cao, NestJS phù hợp cho việc phát triển các hệ thống backend có quy mô vừa và lớn, yêu cầu xử lý nghiệp vụ phức tạp. Trong phạm vi đề tài, NestJS được sử dụng làm nền tảng chính để triển khai backend, đảm nhiệm vai trò xử lý nghiệp vụ, quản lý dữ liệu và cung cấp các API phục vụ cho phía Client.

### **1.2.2. Tổng quan về framework Nextjs**

NextJS là một framework frontend được xây dựng trên nền tảng React, sử dụng JavaScript và TypeScript, hỗ trợ phát triển các ứng dụng web hiện đại. NextJS cung cấp nhiều cơ chế render như Server-Side Rendering (SSR), Static Site Generation (SSG) và Client-Side Rendering (CSR), giúp tối ưu hiệu năng hiển thị và cải thiện trải nghiệm người dùng.

Framework này cung cấp hệ thống routing dựa trên cấu trúc thư mục, giúp việc tổ chức giao diện và điều hướng trong ứng dụng trở nên rõ ràng và dễ quản lý. Bên cạnh đó, NextJS hỗ trợ tốt cho việc tối ưu hóa công cụ tìm kiếm (SEO) và khả năng tích hợp

linh hoạt với các API backend. Trong phạm vi đề tài, NextJS được sử dụng để xây dựng giao diện người dùng cho hệ thống, đảm nhiệm vai trò hiển thị dữ liệu, tiếp nhận thao tác từ người dùng và giao tiếp với backend thông qua các API đã được triển khai.

### 1.2.3. Fastapi

FastAPI là một framework backend sử dụng ngôn ngữ Python, được thiết kế để xây dựng các dịch vụ API với hiệu năng cao và cú pháp rõ ràng. FastAPI dựa trên các chuẩn hiện đại như ASGI và OpenAPI, cho phép tự động sinh tài liệu API và hỗ trợ kiểm tra, xác thực dữ liệu đầu vào một cách hiệu quả.

Framework này hỗ trợ lập trình bất đồng bộ, giúp nâng cao khả năng xử lý đồng thời và cải thiện hiệu suất của hệ thống. Nhờ khả năng tích hợp tốt với các thư viện khoa học dữ liệu và trí tuệ nhân tạo trong hệ sinh thái Python, FastAPI đặc biệt phù hợp cho các tác vụ xử lý chuyên biệt và tính toán nâng cao. Trong phạm vi đề tài, FastAPI được sử dụng để triển khai các dịch vụ xử lý độc lập, hỗ trợ các chức năng liên quan đến xử lý dữ liệu và trí tuệ nhân tạo, đồng thời giao tiếp với backend chính thông qua các API.

### 1.2.4. Websocket

WebSocket là một giao thức truyền thông cho phép thiết lập kết nối hai chiều liên tục giữa Client và Server thông qua một kết nối duy nhất. Khác với mô hình giao tiếp HTTP truyền thống theo cơ chế request–response, WebSocket duy trì kết nối trong suốt phiên làm việc, giúp giảm độ trễ và nâng cao hiệu quả trao đổi dữ liệu.

WebSocket hỗ trợ truyền dữ liệu theo thời gian thực, cho phép Server chủ động gửi thông tin đến Client mà không cần chờ yêu cầu mới. Nhờ đặc điểm này, WebSocket phù hợp với các hệ thống yêu cầu cập nhật dữ liệu nhanh và liên tục. Trong phạm vi đề tài, WebSocket được sử dụng để hỗ trợ cơ chế giao tiếp thời gian thực giữa Client và Server, góp phần nâng cao khả năng phản hồi và trải nghiệm người dùng của hệ thống.

## 1.3. Lưu trữ & quản lý dữ liệu

### 1.3.1. Tổng quan về Postgresql

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được thiết kế nhằm đảm bảo tính ổn định, độ tin cậy và khả năng mở rộng cao. PostgreSQL hỗ trợ đầy đủ các chuẩn SQL, đồng thời cung cấp nhiều tính năng nâng cao như giao dịch (transaction), ràng buộc toàn vẹn dữ liệu và cơ chế kiểm soát đồng thời, giúp đảm bảo tính nhất quán và an toàn của dữ liệu.

Hệ quản trị cơ sở dữ liệu này cho phép lưu trữ và xử lý hiệu quả các dữ liệu có cấu trúc, đồng thời hỗ trợ mở rộng thông qua các cơ chế chỉ mục, view và stored procedure.

Nhờ khả năng hoạt động ổn định và dễ tích hợp với nhiều framework backend, PostgreSQL phù hợp cho các hệ thống yêu cầu quản lý dữ liệu phức tạp và có quy mô vừa đến lớn. Trong phạm vi đề tài, PostgreSQL được sử dụng để lưu trữ và quản lý dữ liệu chính của hệ thống, bao gồm thông tin người dùng, dữ liệu nghiệp vụ và các dữ liệu có cấu trúc phục vụ cho quá trình xử lý và tính toán.

### 1.3.2. Tổng quan về S3 Storage

S3 Storage là một mô hình lưu trữ đối tượng (Object Storage) được thiết kế để lưu trữ và quản lý các dữ liệu phi cấu trúc như hình ảnh, tài liệu, tệp đa phương tiện và các tệp dữ liệu có dung lượng lớn. Mô hình này cho phép truy cập dữ liệu thông qua các giao thức mạng, đồng thời hỗ trợ khả năng mở rộng linh hoạt và độ bền dữ liệu cao.

S3 Storage tổ chức dữ liệu dưới dạng các đối tượng được lưu trữ trong các bucket, mỗi đối tượng bao gồm dữ liệu và thông tin mô tả (metadata). Cơ chế lưu trữ này giúp việc quản lý, truy xuất và mở rộng dữ liệu trở nên hiệu quả, đặc biệt phù hợp với các hệ thống cần lưu trữ số lượng lớn tệp và dữ liệu không có cấu trúc. Trong phạm vi đề tài, S3 Storage được sử dụng để lưu trữ các tệp dữ liệu và tài nguyên liên quan của hệ thống, góp phần giảm tải cho cơ sở dữ liệu quan hệ và nâng cao khả năng mở rộng cũng như hiệu quả quản lý dữ liệu.

## 1.4. AI Agent

### 1.4.1. Tổng quan

AI Agent là một hệ thống thông minh có khả năng tiếp nhận thông tin đầu vào, phân tích ngữ cảnh, ra quyết định và thực hiện hành động nhằm đạt được mục tiêu cụ thể. Khác với chatbot truyền thống chỉ phản hồi theo kịch bản cố định, AI Agent có khả năng xử lý linh hoạt các tình huống khác nhau nhờ vào việc kết hợp mô hình ngôn ngữ, tri thức và cơ chế ra quyết định. Trong các hệ thống hiện đại, AI Agent thường đóng vai trò trung tâm trong việc tương tác với người dùng, hỗ trợ tự động hóa nghiệp vụ và nâng cao trải nghiệm sử dụng hệ thống.

Trong hệ thống, AI Agent đóng vai trò trung tâm xử lý thông tin thông minh, đặc biệt phù hợp với bài toán đề tài đặt ra:

- Tư vấn dinh dưỡng cho thú cưng
- Phân tích dữ liệu và ra khuyến nghị
- Tự động hóa nghiệp vụ phức tạp

### 1.4.2. Mô hình ngôn ngữ lớn (Large Language Model – LLM)

Large Language Model (LLM) - Mô hình ngôn ngữ lớn là một loại mô hình trí tuệ nhân tạo (AI) được phát triển để xử lý và tạo ngôn ngữ tự nhiên giống con người. Các LLM hoạt động dựa trên nền tảng học sâu (deep learning) và đặc biệt sử dụng kiến trúc

mạng nơ-ron, thường là mô hình Transformer, để nắm bắt cấu trúc và ý nghĩa ngôn ngữ. Được huấn luyện trên lượng dữ liệu khổng lồ, bao gồm sách, bài báo, trang web, và nhiều nguồn tài nguyên ngôn ngữ khác, LLM không chỉ học cách xử lý các mẫu ngôn ngữ phức tạp mà còn nắm bắt được ngữ cảnh, ngữ nghĩa, và cả bối cảnh văn hóa trong nhiều ngôn ngữ khác nhau.

Ưu điểm:

- Khả năng hiểu và tạo phản hồi gần giống con người.
- Có thể phân tích và xử lý thông tin tốt giúp thực hiện các tác vụ đa dạng.
- Có thể tạo phản hồi mới, sáng tạo và phù hợp với ngữ cảnh yêu cầu.

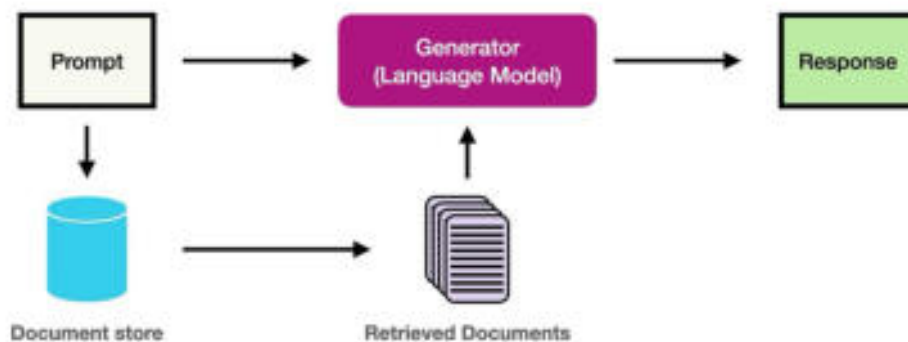
Nhược điểm:

- Để có thể khai thác được các điểm mạnh của LLM thì đòi hỏi phần cứng mạnh mẽ và chi phí tốn kém, đặc biệt là trong môi trường sản xuất.
- Mặc dù mô hình có thể tạo ra các văn bản tự nhiên, tuy nhiên đôi khi vẫn mắc lỗi, và tạo ra các thông tin sai lệch.

### 1.4.3. Kỹ thuật Retrieval – Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) là một phương pháp kết hợp giữa mô hình sinh ngôn ngữ và cơ chế truy xuất thông tin nhằm nâng cao độ chính xác và tính tin cậy của kết quả sinh nội dung. Thay vì chỉ dựa vào kiến thức đã được huấn luyện sẵn trong mô hình ngôn ngữ, RAG cho phép hệ thống khai thác thêm các dữ liệu liên quan từ nguồn tri thức bên ngoài để làm ngữ cảnh cho quá trình sinh câu trả lời.

Trong mô hình RAG (Hình 1.2), dữ liệu văn bản ban đầu được tiền xử lý bằng cách chia nhỏ thành các đoạn (chunk) nhằm đảm bảo tính phù hợp và hiệu quả trong quá trình tìm kiếm. Mỗi đoạn dữ liệu sau đó được chuyển đổi thành vector biểu diễn ngữ nghĩa thông qua kỹ thuật embedding. Các vector này được lưu trữ trong cơ sở dữ liệu vector và được sử dụng để truy xuất các thông tin liên quan dựa trên mức độ tương đồng ngữ nghĩa với truy vấn của người dùng.



Hình 1.2 Mô hình hoạt động của Retrieval-Augmented Generation (RAG)

Quá trình hoạt động của RAG bao gồm hai bước chính là truy xuất và sinh nội dung. Ở bước truy xuất, hệ thống tìm kiếm các đoạn dữ liệu có mức độ liên quan cao nhất so với truy vấn đầu vào. Các thông tin được truy xuất sẽ được kết hợp và cung cấp làm ngữ cảnh cho mô hình sinh ngôn ngữ ở bước tiếp theo, từ đó tạo ra câu trả lời phù hợp và có căn cứ. Cách tiếp cận này giúp giảm thiểu việc sinh ra thông tin không chính xác và tăng khả năng cập nhật kiến thức của hệ thống.

Trong phạm vi đề tài, RAG được áp dụng nhằm hỗ trợ hệ thống khai thác hiệu quả các nguồn dữ liệu sẵn có, nâng cao chất lượng kết quả trả lời và khả năng hỗ trợ người dùng. Việc sử dụng RAG góp phần cải thiện độ chính xác, tính nhất quán và giá trị thực tiễn của các chức năng trí tuệ nhân tạo trong hệ thống.

#### **1.4.4. Text Embeddings**

Text Embeddings là phương pháp biểu diễn văn bản dưới dạng vector số trong không gian nhiều chiều, cho phép đo lường mức độ tương đồng ngữ nghĩa giữa các đoạn văn bản. Các đoạn văn có nội dung tương tự sẽ có vector gần nhau, trong khi các đoạn khác nghĩa sẽ có khoảng cách lớn hơn. Trong hệ thống AI Agent, embeddings được sử dụng chủ yếu cho các bài toán tìm kiếm ngữ nghĩa và truy xuất thông tin, đóng vai trò nền tảng cho kỹ thuật RAG. Nhờ embeddings, hệ thống có thể tìm được các tài liệu liên quan về mặt ý nghĩa thay vì chỉ dựa trên so khớp từ khóa đơn thuần.

#### **1.4.5. Prompt Engineering**

Prompt Engineering là kỹ thuật thiết kế và tối ưu câu lệnh đầu vào nhằm điều hướng hành vi và chất lượng phản hồi của mô hình ngôn ngữ lớn. Một prompt được xây dựng tốt sẽ mô tả rõ vai trò của mô hình, nhiệm vụ cần thực hiện, ngữ cảnh liên quan và yêu cầu đầu ra cụ thể. Trong AI Agent, prompt engineering giúp đảm bảo mô hình trả lời đúng mục tiêu, nhất quán với dữ liệu được cung cấp và hạn chế việc sinh thông tin không chính xác. Kỹ thuật này đóng vai trò quan trọng trong việc kết nối dữ liệu truy xuất từ RAG với khả năng sinh ngôn ngữ của LLM.

### **1.5. Triển khai & vận hành hệ thống**

#### **1.5.1. Docker**

Docker là một nền tảng hỗ trợ đóng gói và phát triển ứng dụng theo mô hình container hóa (containerization). Thay vì cài đặt và cấu hình ứng dụng trực tiếp trên hệ điều hành, Docker cho phép đóng gói toàn bộ ứng dụng cùng các thư viện phụ thuộc, cấu hình và môi trường chạy vào một Docker Image. Từ image này, hệ thống có thể tạo ra các container để chạy ứng dụng một cách nhất quán nhiều môi trường khác nhau.

Docker hoạt động dựa trên cơ chế chia sẻ nhân hệ điều hành (kernel) của máy chủ, giúp container nhẹ hơn và khởi chạy nhanh hơn so với mô hình máy ảo truyền thống.

Mỗi container được cô lập ở mức cần thiết, đảm bảo ứng dụng có thể chạy ổn định mà không bị ảnh hưởng bởi sự khác biệt về môi trường, phiên bản thư viện hoặc cấu hình hệ thống. Nhờ vậy, Docker giúp giảm lỗi phát sinh khi chuyển đổi giữa các môi trường phát triển, kiểm thử và triển khai thực tế.

Trong phạm vi đề tài, Docker được sử dụng để chuẩn hóa môi trường chạy và triển khai các thành phần của hệ thống, giúp quá trình cài đặt, vận hành và mở rộng trở nên thuận tiện. Việc container hóa cũng hỗ trợ tốt cho tích hợp với các quy trình triển khai tự động (CI/CD), từ đó nâng cao tính ổn định và hiệu quả trong quá trình phát triển và vận hành hệ thống.

### 1.5.2. CI/CD

CI/CD (Continuous Integration / Continuous Deployment) là một quy trình tự động hóa trong phát triển phần mềm, nhằm đảm bảo việc tích hợp, kiểm thử và triển khai ứng dụng được thực hiện một cách liên tục và nhất quán. CI/CD giúp rút ngắn chu kỳ phát triển, giảm thiểu lỗi phát sinh và nâng cao chất lượng phần mềm trong quá trình vận hành.

Trong giai đoạn Continuous Integration, mã nguồn được thường xuyên tích hợp vào hệ thống quản lý phiên bản. Mỗi lần cập nhật mã nguồn đều kích hoạt các bước kiểm tra tự động như build và kiểm thử, nhằm phát hiện sớm lỗi và đảm bảo tính ổn định của hệ thống. Giai đoạn Continuous Deployment cho phép tự động triển khai phiên bản đã được kiểm tra lên môi trường chạy thực tế, giúp quá trình cập nhật hệ thống diễn ra nhanh chóng và hạn chế thao tác thủ công.

Trong phạm vi đề tài, CI/CD được áp dụng nhằm tự động hóa quy trình xây dựng và triển khai hệ thống, kết hợp với Docker để đảm bảo môi trường triển khai nhất quán. Việc sử dụng CI/CD giúp nâng cao hiệu quả phát triển, giảm thiểu rủi ro khi cập nhật hệ thống và hỗ trợ tốt cho quá trình vận hành, bảo trì trong thực tế.

## 1.6. Dữ liệu nghiên cứu và tham khảo

FEDIAF (European Pet Food Industry Federation) – Liên đoàn Công nghiệp Thức ăn Thú Cưng Châu Âu [7], đây là tổ chức đại diện cho ngành công nghiệp thức ăn cho chó và mèo tại châu Âu. Được thành lập với mục tiêu đảm bảo an toàn, chất lượng và giá trị dinh dưỡng của thức ăn thú cưng, FEDIAF đã trở thành một trong những cơ quan hàng đầu về nghiên cứu và ban hành tiêu chuẩn dinh dưỡng cho động vật nuôi.

FEDIAF định kỳ xuất bản tài liệu "Nutritional Guidelines for Complete and Complementary Pet Food for Cats and Dogs" (Hướng dẫn dinh dưỡng cho thức ăn hoàn chỉnh và bổ sung dành cho chó và mèo), được cập nhật thường xuyên dựa trên các nghiên cứu khoa học mới nhất. Tài liệu này được các chuyên gia thú y, nhà dinh

dưỡng động vật và các nhà sản xuất thức ăn thú cưng trên toàn thế giới công nhận và sử dụng làm tài liệu tham khảo chuẩn.

## Chương 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 2.1. Mô tả bài toán

Hiện nay, nhu cầu chăm sóc sức khỏe và dinh dưỡng cho thú cưng, đặc biệt là chó và mèo, ngày càng được quan tâm. Tuy nhiên, phần lớn người nuôi thú cưng gặp khó khăn trong việc xây dựng khẩu phần ăn phù hợp do thiếu kiến thức chuyên môn về dinh dưỡng, cũng như khó tiếp cận các nguồn thông tin chính xác và đáng tin cậy. Việc lựa chọn và kết hợp thực phẩm chủ yếu dựa trên kinh nghiệm cá nhân hoặc khuyến nghị không chính thống, dẫn đến nguy cơ mất cân đối dinh dưỡng, ảnh hưởng đến sức khỏe lâu dài của thú cưng.

Bên cạnh đó, các tiêu chuẩn và khuyến nghị dinh dưỡng hiện nay (như FEDIAF) thường mang tính học thuật và khó áp dụng trực tiếp đối với người nuôi thú cưng phổ thông. Người dùng cần một công cụ hỗ trợ có khả năng phân tích các thông tin cơ bản của thú cưng (như loài, độ tuổi, cân nặng, mức độ hoạt động) để từ đó đề xuất khẩu phần ăn phù hợp, dễ hiểu và có tính thực tiễn.

Từ thực tế trên, bài toán đặt ra là xây dựng một hệ thống hỗ trợ dinh dưỡng thú cưng thông minh cho chó và mèo, có khả năng tự động đề xuất thực đơn dựa trên các thông tin đầu vào của người dùng, đồng thời kết hợp với dữ liệu thực phẩm để gợi ý các sản phẩm phù hợp. Hệ thống cần đảm bảo tính chính xác ở mức khuyến nghị chung, dễ sử dụng đối với người dùng phổ thông và có khả năng mở rộng trong tương lai.

Hệ thống không hướng tới việc chẩn đoán hay điều trị bệnh cho thú cưng, mà tập trung vào vai trò hỗ trợ tham khảo về dinh dưỡng và định hướng người dùng tiếp cận các giải pháp chăm sóc phù hợp. Bài toán được giải quyết thông qua việc ứng dụng công nghệ web kết hợp với các kỹ thuật xử lý dữ liệu và trí tuệ nhân tạo nhằm nâng cao hiệu quả và giá trị sử dụng của hệ thống.

### 2.2. Phân tích yêu cầu hệ thống

#### 2.2.1. Yêu cầu chức năng

Hệ thống cần đáp ứng các yêu cầu chức năng cốt lõi nhằm hỗ trợ nuôi thú cưng xây dựng khẩu phần ăn và thực đơn phù hợp cho chó và mèo một cách khoa học, trực quan và thuận tiện. Các chức năng chính của hệ thống bao gồm:

Xử lý thông tin đầu vào từ người dùng

- Cho phép người dùng nhập thông tin cơ bản của thú cưng như loài (chó hoặc mèo), độ tuổi, cân nặng, mức độ hoạt động và các đặc điểm liên quan.

- Kiểm tra và xác thực dữ liệu đầu vào nhằm đảm bảo tính hợp lệ trước khi thực hiện xử lý.
- Lưu trữ thông tin thú cưng để phục vụ cho việc phân tích và đề xuất trong các lần sử dụng tiếp theo.

#### Phân tích nhu cầu dinh dưỡng

- Phân tích các thông tin đầu vào để xác định nhu cầu dinh dưỡng cơ bản của thú cưng.
- Áp dụng các nguyên tắc và khuyến nghị dinh dưỡng để làm cơ sở tính toán khẩu phần ăn.
- Hỗ trợ điều chỉnh kết quả phân tích khi người dùng thay đổi thông tin đầu vào.

#### Đề xuất khẩu phần ăn và thực đơn

- Tự động đề xuất khẩu phần ăn phù hợp với từng cá thể thú cưng dựa trên kết quả phân tích.
- Cung cấp thông tin chi tiết về thành phần và giá trị dinh dưỡng của khẩu phần ăn được đề xuất.
- Cho phép người dùng tham khảo và lựa chọn thực đơn phù hợp với nhu cầu sử dụng.

#### Gợi ý thực phẩm và sản phẩm liên quan

- Gợi ý các loại thực phẩm hoặc sản phẩm phù hợp với khẩu phần ăn đã được đề xuất.
- Hiện thị thông tin cơ bản của sản phẩm như thành phần, giá trị dinh dưỡng và giá bán (nếu có).
- Hỗ trợ người dùng so sánh và lựa chọn sản phẩm phù hợp.

### 2.2.2. Yêu cầu phi chức năng

Bên cạnh các yêu cầu chức năng, hệ thống cần đáp ứng các yêu cầu phi chức năng nhằm đảm bảo tính ổn định, hiệu quả và khả năng sử dụng trong quá trình vận hành. Các yêu cầu phi chức năng của hệ thống bao gồm:

- Yêu cầu về hiệu năng: Hệ thống cần phản hồi nhanh và đảm bảo xử lý ổn định khi có nhiều người dùng truy cập đồng thời.
- Yêu cầu về tính mở rộng: Hệ thống cần có khả năng mở rộng chức năng và quy mô trong tương lai mà không ảnh hưởng đến hoạt động hiện tại.
- Yêu cầu về độ tin cậy: Hệ thống cần hoạt động ổn định, đảm bảo tính nhất quán và an toàn của dữ liệu.
- Yêu cầu về bảo mật: Thông tin người dùng và dữ liệu hệ thống cần được bảo vệ khỏi truy cập trái phép.

- Yêu cầu về khả năng sử dụng: Giao diện cần thân thiện, dễ sử dụng và phù hợp với người dùng phổ thông.
- Yêu cầu về khả năng bảo trì: Hệ thống cần được thiết kế rõ ràng, thuận tiện cho việc bảo trì và nâng cấp.

### 2.3. Phân tích thiết kế hệ thống

#### 2.3.1. Biểu đồ Use case tổng quát

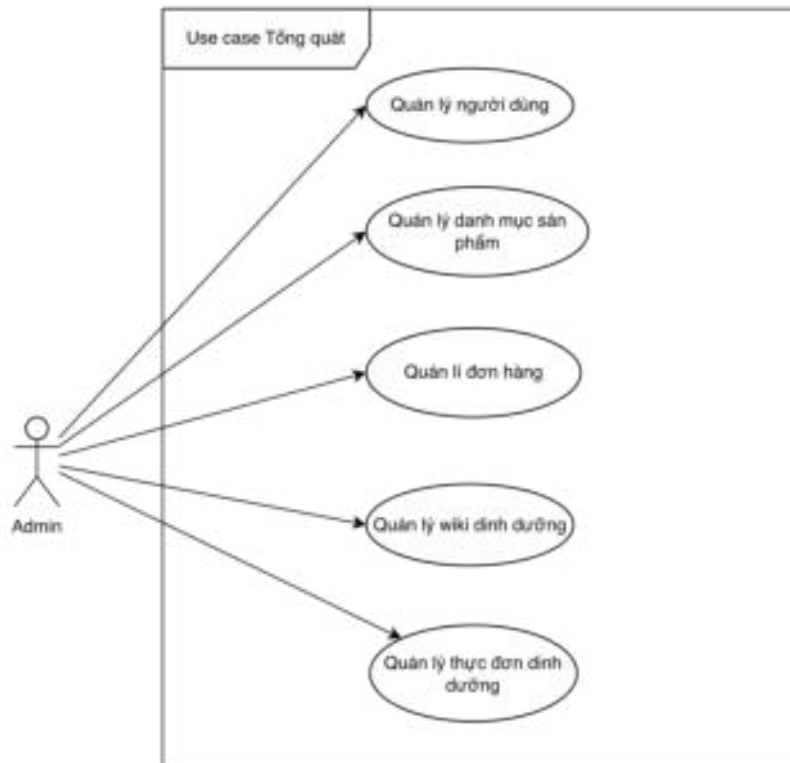
Tác nhân của hệ thống: Người dùng, quản trị viên

Bảng 2.1 Các tác nhân của hệ thống

Tên tác nhân	Mô tả
Người dùng	<ul style="list-style-type: none"> <li>• Đăng ký/ Đăng nhập</li> <li>• Quản lý thông tin cá nhân</li> <li>• Thêm mới, cập nhật thông tin thú cưng</li> <li>• Quản lý thông báo</li> <li>• Lập thực đơn dinh dưỡng thông minh (tạo, xem)</li> <li>• Mua sắm sản phẩm</li> <li>• Quản lý giỏ hàng</li> <li>• Quản lý đơn hàng</li> <li>• Chatbot tư vấn dinh dưỡng</li> </ul>
Quản trị viên	<ul style="list-style-type: none"> <li>• Dashboard thống kê</li> <li>• Quản lý người dùng</li> <li>• Quản lý danh mục sản phẩm</li> <li>• Quản lý đơn hàng</li> <li>• Quản lý wiki dinh dưỡng</li> <li>• Quản lý thực đơn dinh dưỡng</li> <li>• Quản lý thông báo</li> </ul>



Hình 2.1 Use case tổng quát người dùng

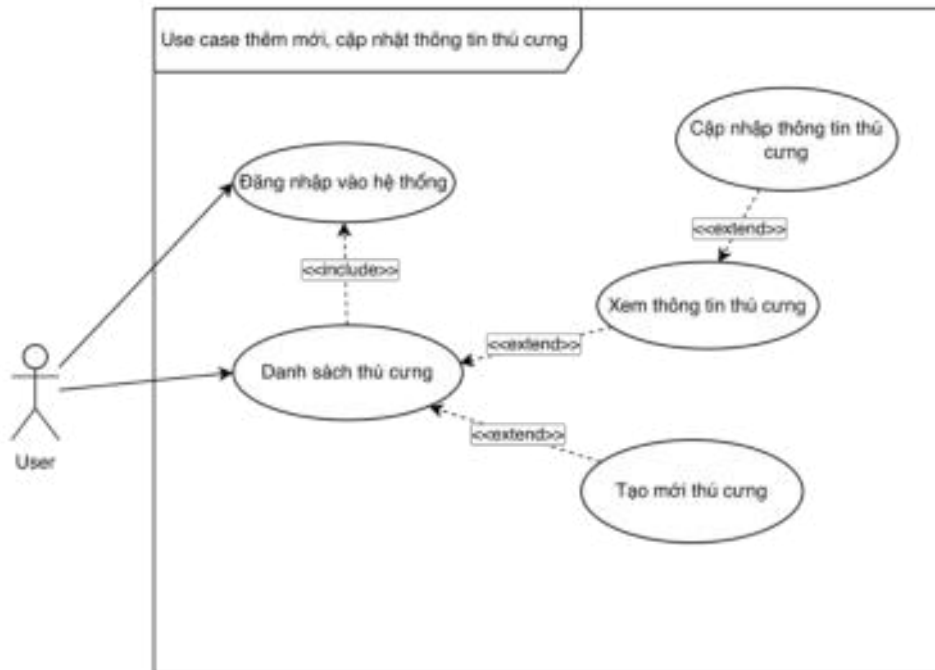


Hình 2.2 Use case tổng quát admin

a. Use case Thêm mới, cập nhật thông tin thú cưng

Bảng 2.2 Mô tả use case Thêm mới, cập nhật thông tin thú cưng

Mã Use Case	UC01
Tên Use Case	Thêm mới, cập nhật thông tin thú cưng
Tác nhân	Người dùng
Mô tả	Người dùng thêm mới hoặc cập nhật thông tin thú cưng (đã thêm mới thú cưng trước đó) nhằm phục vụ cho việc phân tích nhu cầu dinh dưỡng và đề xuất thực đơn phù hợp trong hệ thống.
Điều kiện	Yêu cầu người dùng phải đăng nhập hệ thống

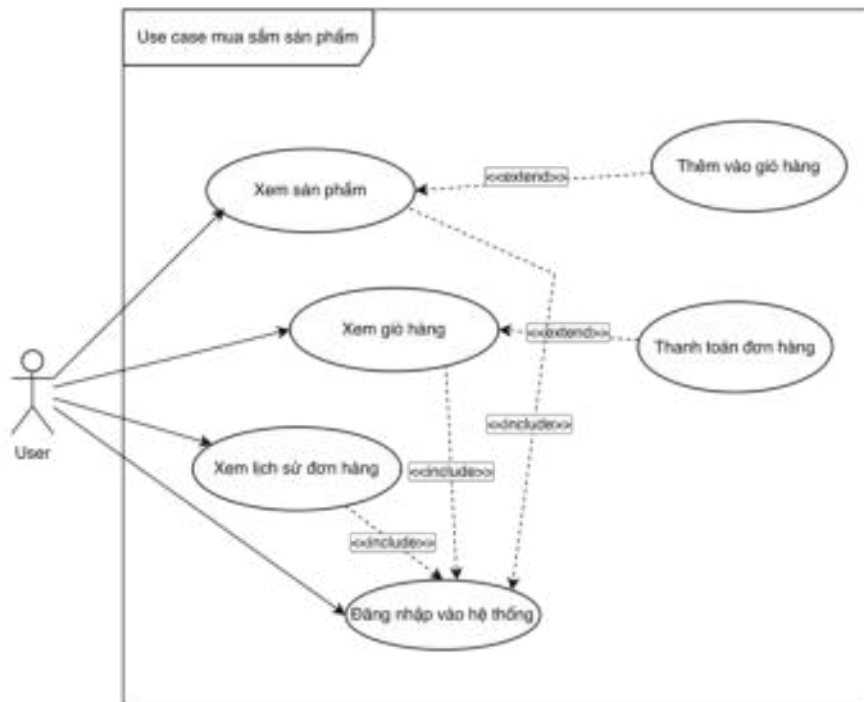


Hình 2.3 Use case Thêm mới, cập nhật thông tin thú cưng

b. Use case Mua sắm sản phẩm

Bảng 2.3 Mô tả use case Mua sắm sản phẩm

Mã Use Case	UC02
Tên Use Case	Mua sắm sản phẩm
Tác nhân	Người dùng
Mô tả	Người dùng tham khảo và lựa chọn sản phẩm phù hợp. Có thể xem thông tin chi tiết sản phẩm, lựa chọn số lượng và thực hiện thao tác mua sắm theo nhu cầu.
Điều kiện	Yêu cầu người dùng phải đăng nhập hệ thống và hệ thống tồn tại sản phẩm.

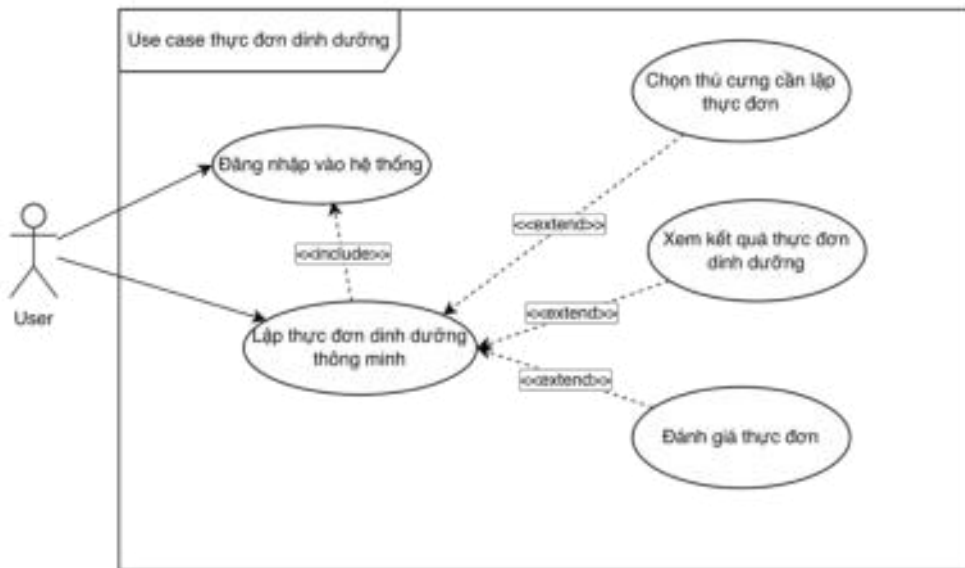


Hình 2.4 Use case Mua sắm sản phẩm

c. Use case Lập thực đơn dinh dưỡng thông minh

Bảng 2.4 Mô tả use case Lập thực đơn dinh dưỡng

Mã Use Case	UC03
Tên Use Case	Lập thực đơn dinh dưỡng thông minh
Tác nhân	Người dùng
Mô tả	Người dùng lập thực đơn dinh dưỡng cho thú cưng dựa trên các thông tin đã cung cấp và kết quả phân tích nhu cầu dinh dưỡng. Hệ thống tự động đề xuất thực đơn phù hợp nhằm hỗ trợ người dùng xây dựng khẩu phần ăn khoa học cho chó hoặc mèo.
Điều kiện	Yêu cầu người dùng đã đăng nhập và đã tạo hồ sơ thú cưng.

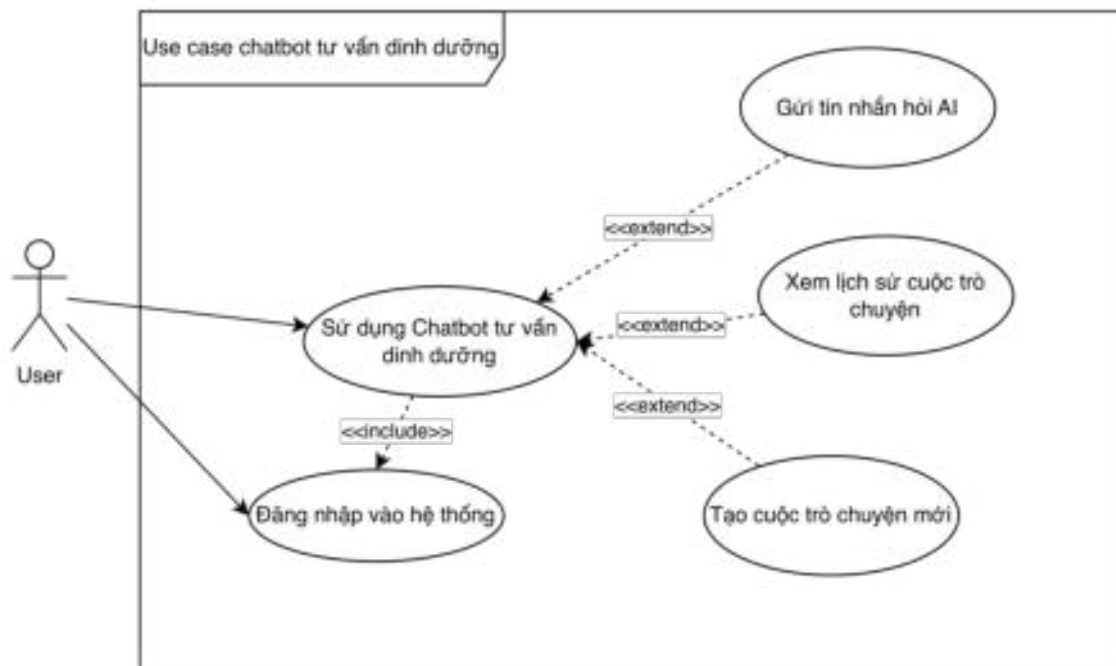


Hình 2.5 Use case Lập thực đơn dinh dưỡng thông minh

d. Use case Chatbot tư vấn dinh dưỡng

Bảng 2.5 Mô tả use case Chatbot tư vấn dinh dưỡng

Mã Use Case	UC04
Tên Use Case	Chatbot tư vấn dinh dưỡng
Tác nhân	Người dùng
Mô tả	Người dùng tương tác với chatbot để nhận tư vấn và giải đáp các câu hỏi liên quan đến dinh dưỡng cho thú cưng. Hệ thống tiếp nhận nội dung câu hỏi, thực hiện phân tích và phản hồi thông tin tư vấn dựa trên dữ liệu và kiến thức được tích hợp trong hệ thống.
Điều kiện	Yêu cầu người dùng đã đăng nhập.

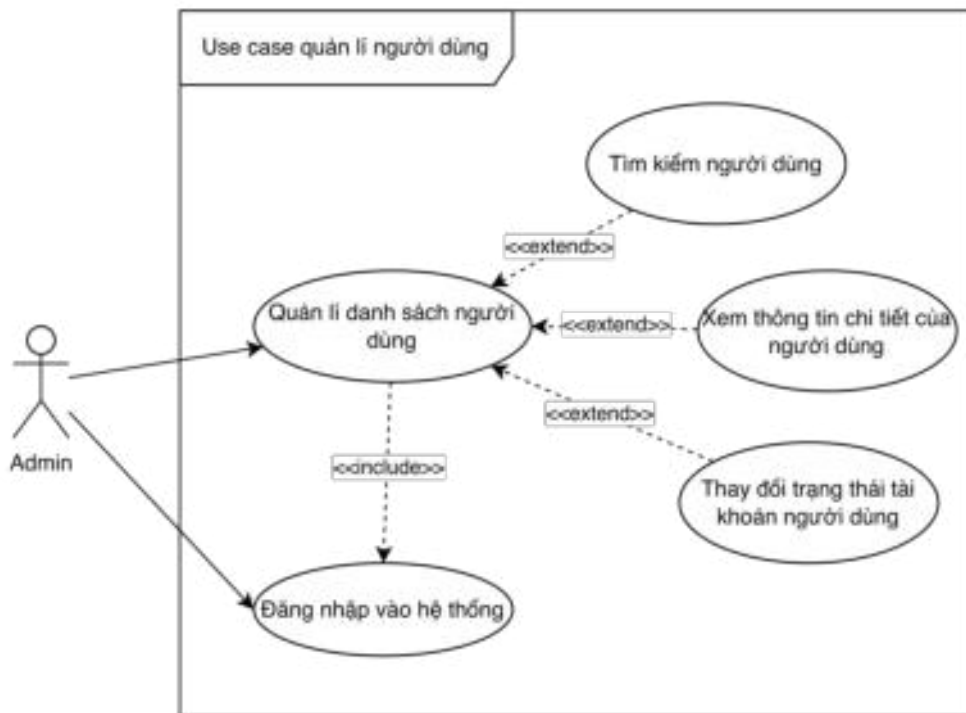


Hình 2.6 Use case Chatbot tư vấn dinh dưỡng

e. Use case Quản lý người dùng

Bảng 2.6 Mô tả use case Quản lý người dùng

Mã Use Case	UC05
Tên Use Case	Quản lý người dùng
Tác nhân	Quản trị viên
Mô tả	Quản trị viên quản lý danh sách người dùng, xem thông tin chi tiết và kiểm soát trạng thái hoạt động của các tài khoản trong hệ thống.
Điều kiện	Yêu cầu người dùng đăng nhập với vai trò là Quản trị viên

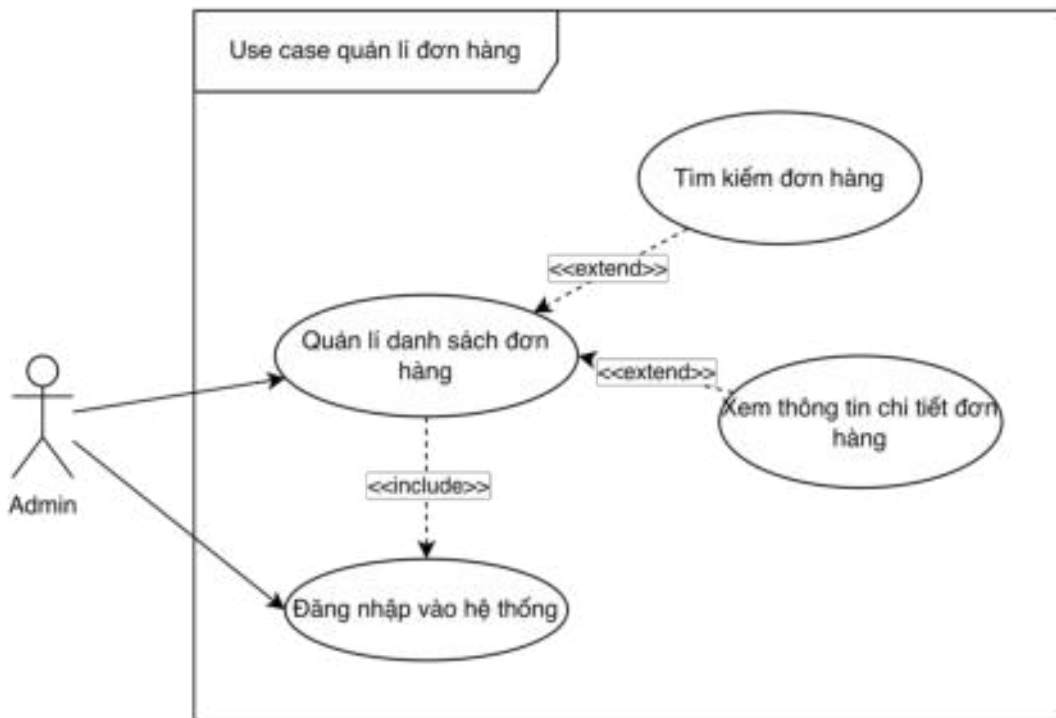


Hình 2.7 Use case quản lý người dùng

f. Use case Quản lý đơn hàng

Bảng 2.7 Mô tả use case Quản lý đơn hàng

Mã Use Case	UC06
Tên Use Case	Quản lý đơn hàng
Tác nhân	Quản trị viên
Mô tả	Quản trị viên quản lý các đơn hàng phát sinh trong hệ thống, bao gồm xem danh sách đơn hàng, xem thông tin chi tiết từng đơn.
Điều kiện	Yêu cầu người dùng đăng nhập với vai trò là Quản trị viên

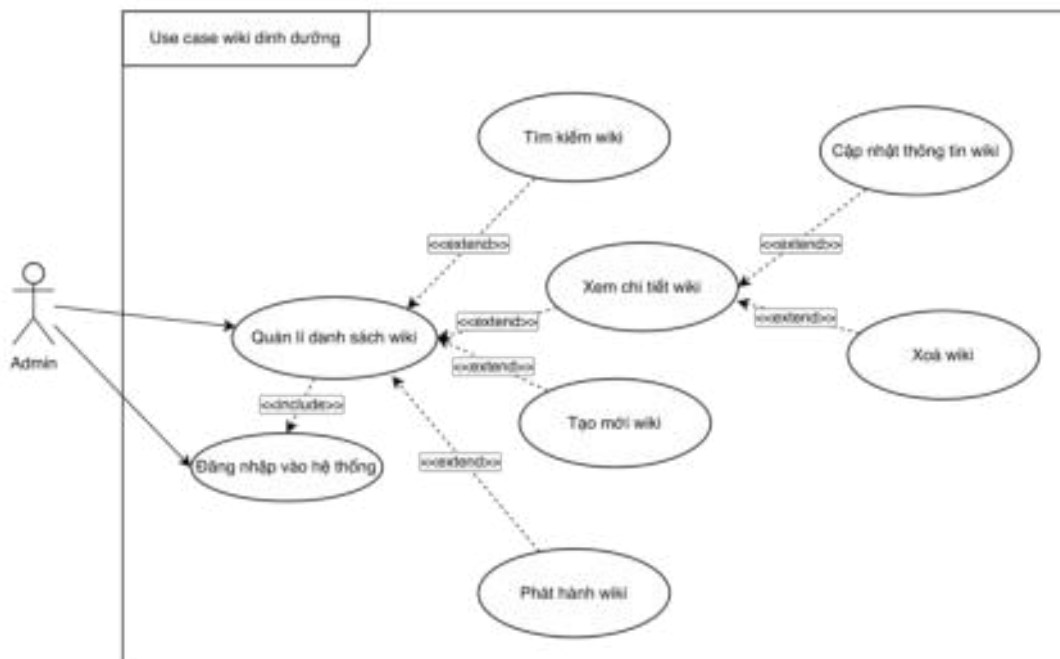


Hình 2.8 Use case quản lý đơn hàng

g. Use case Quản lý wiki dinh dưỡng

Bảng 2.8 Mô tả use case Quản lý wiki dinh dưỡng

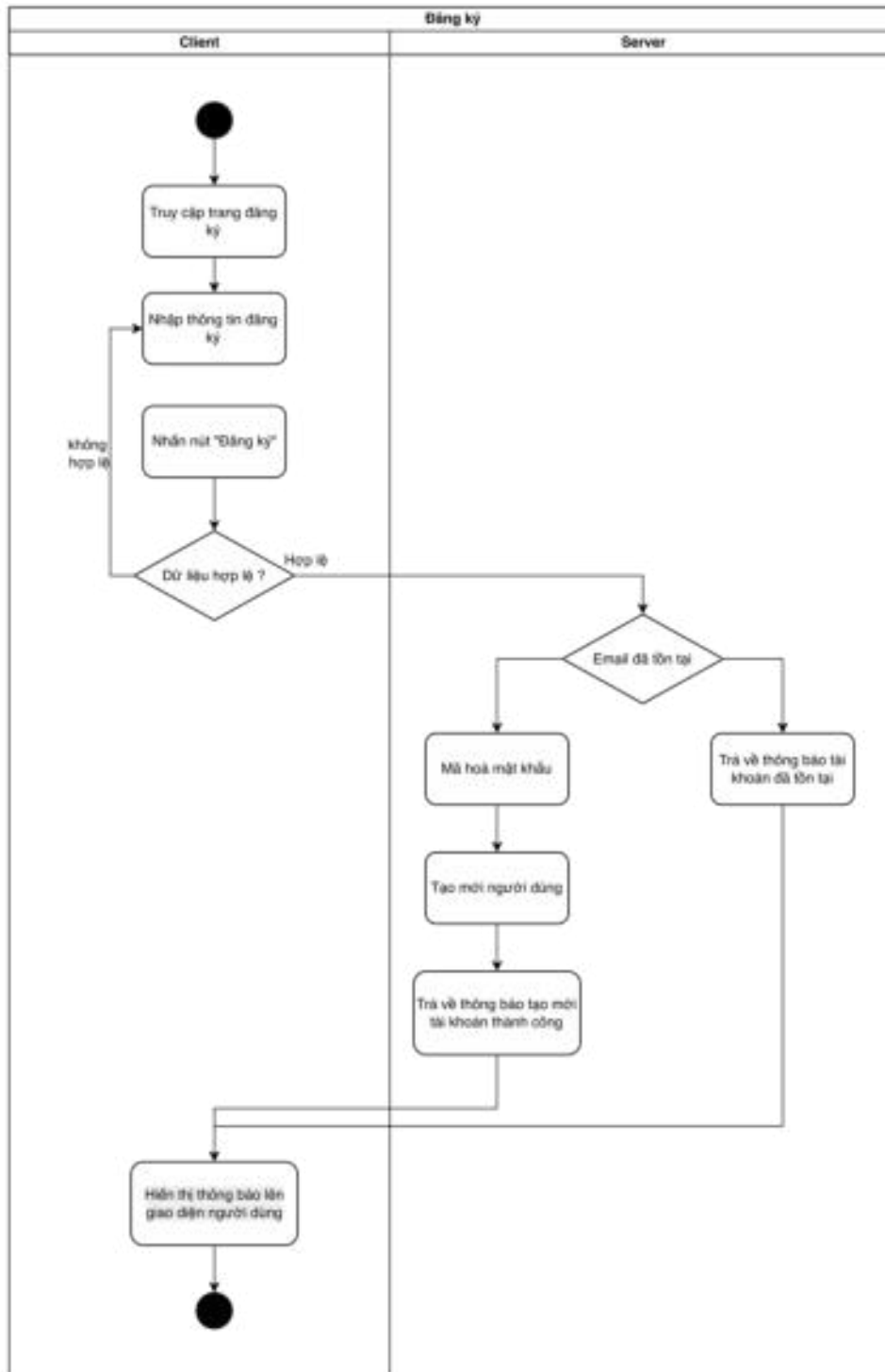
Mã Use Case	UC07
Tên Use Case	Quản lý wiki dinh dưỡng
Tác nhân	Quản trị viên
Mô tả	Quản trị viên quản lý nội dung wiki dinh dưỡng trong hệ thống, bao gồm tạo mới, cập nhật, chỉnh sửa, xoá cũng như publish các bài viết. Nội dung wiki được sử dụng để cung cấp thông tin tham khảo và hỗ trợ tư vấn dinh dưỡng cho thú cưng.
Điều kiện	Yêu cầu người dùng đăng nhập với vai trò là Quản trị viên



Hình 2.9 Use case quản lý wiki dinh dưỡng

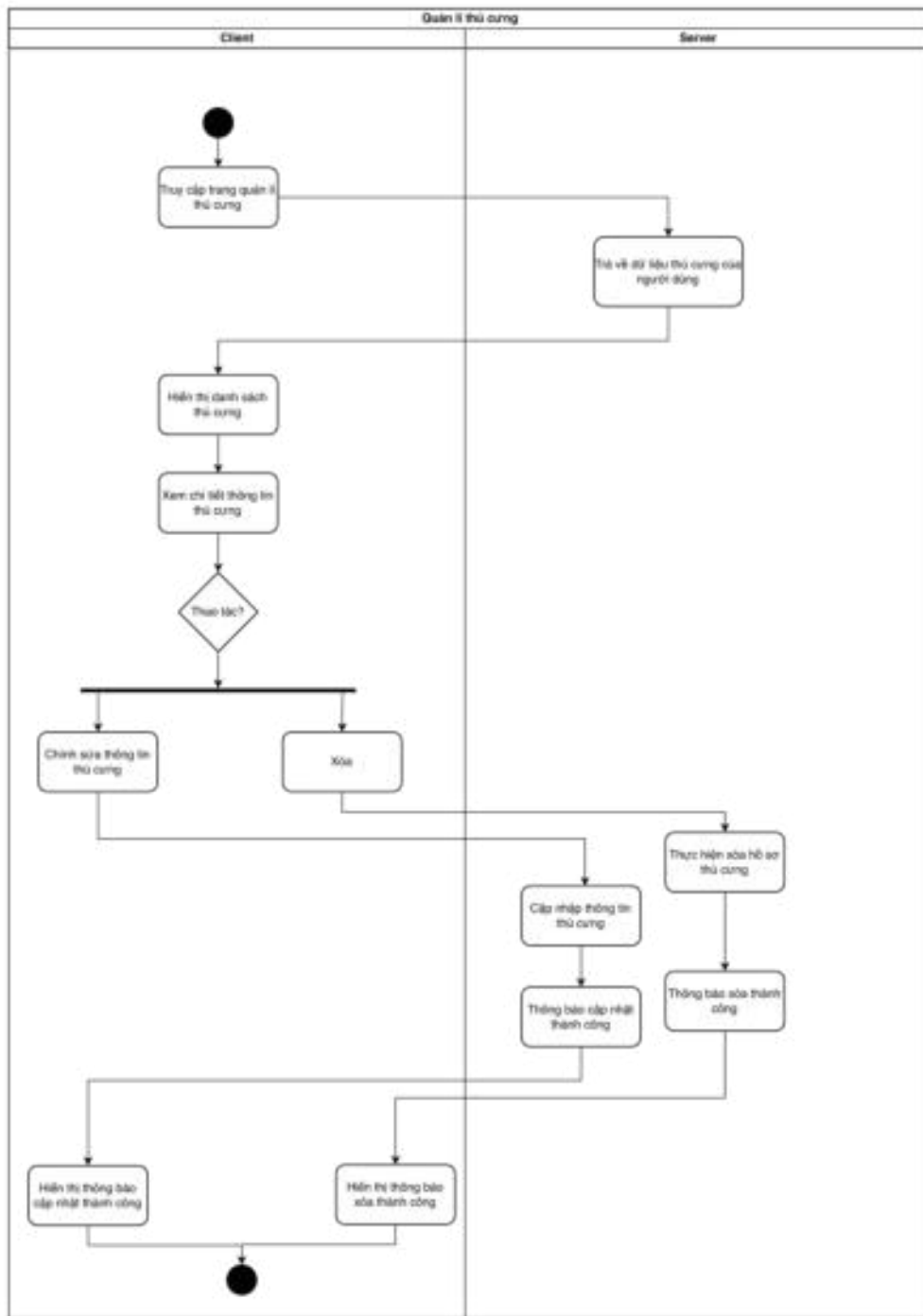
### 2.3.2. Biểu đồ hoạt động

Biểu đồ hoạt động chức năng “Đăng ký”



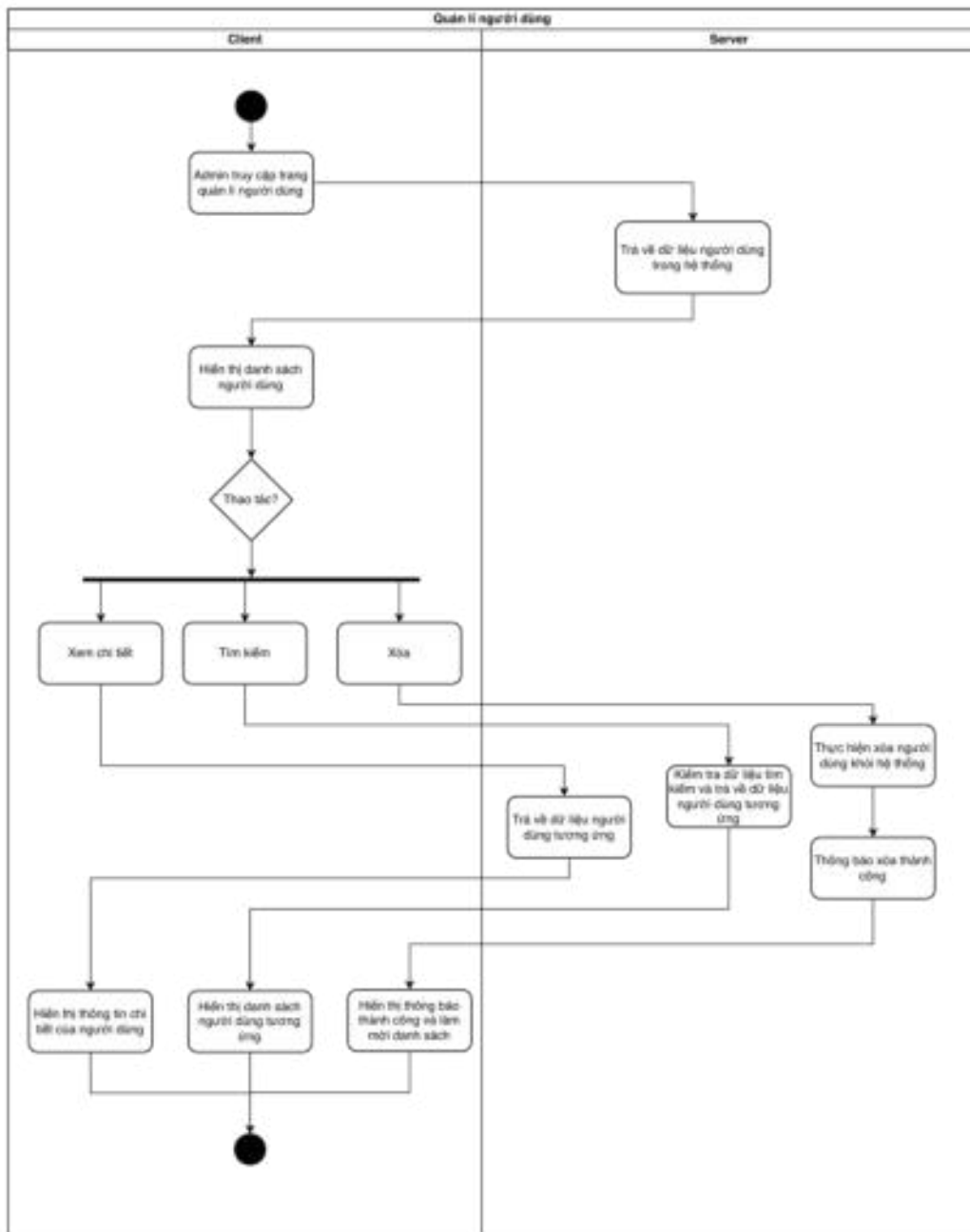
Hình 2.10 Biểu đồ hoạt động chức năng "Đăng kí"

Biểu đồ hoạt động chức năng “Quản lý thú cưng”



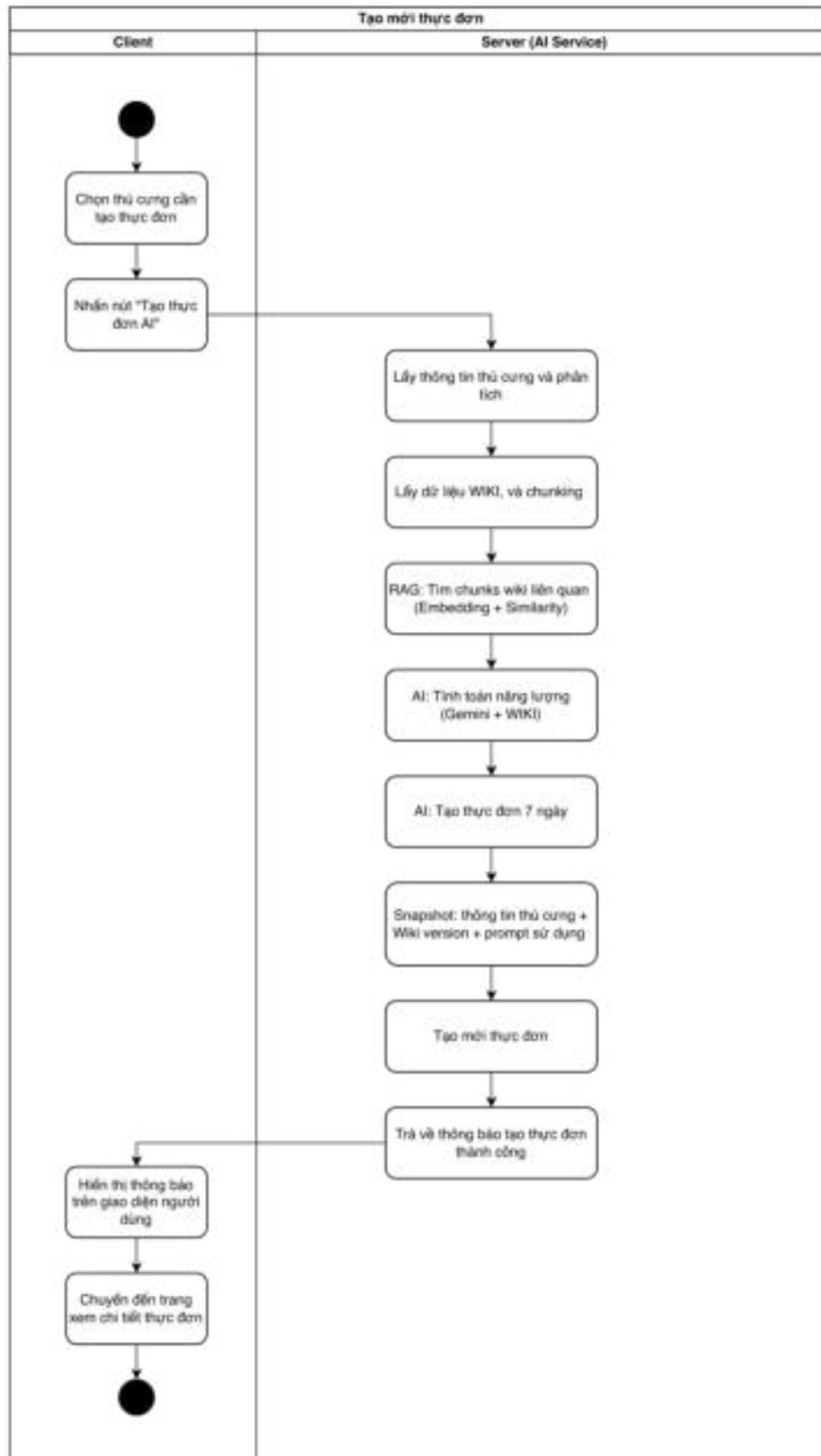
Hình 2.11 Biểu đồ hoạt động chức năng "Tạo mới hồ sơ cho thú cưng"

Biểu đồ hoạt động chức năng "Quản lí người dùng"



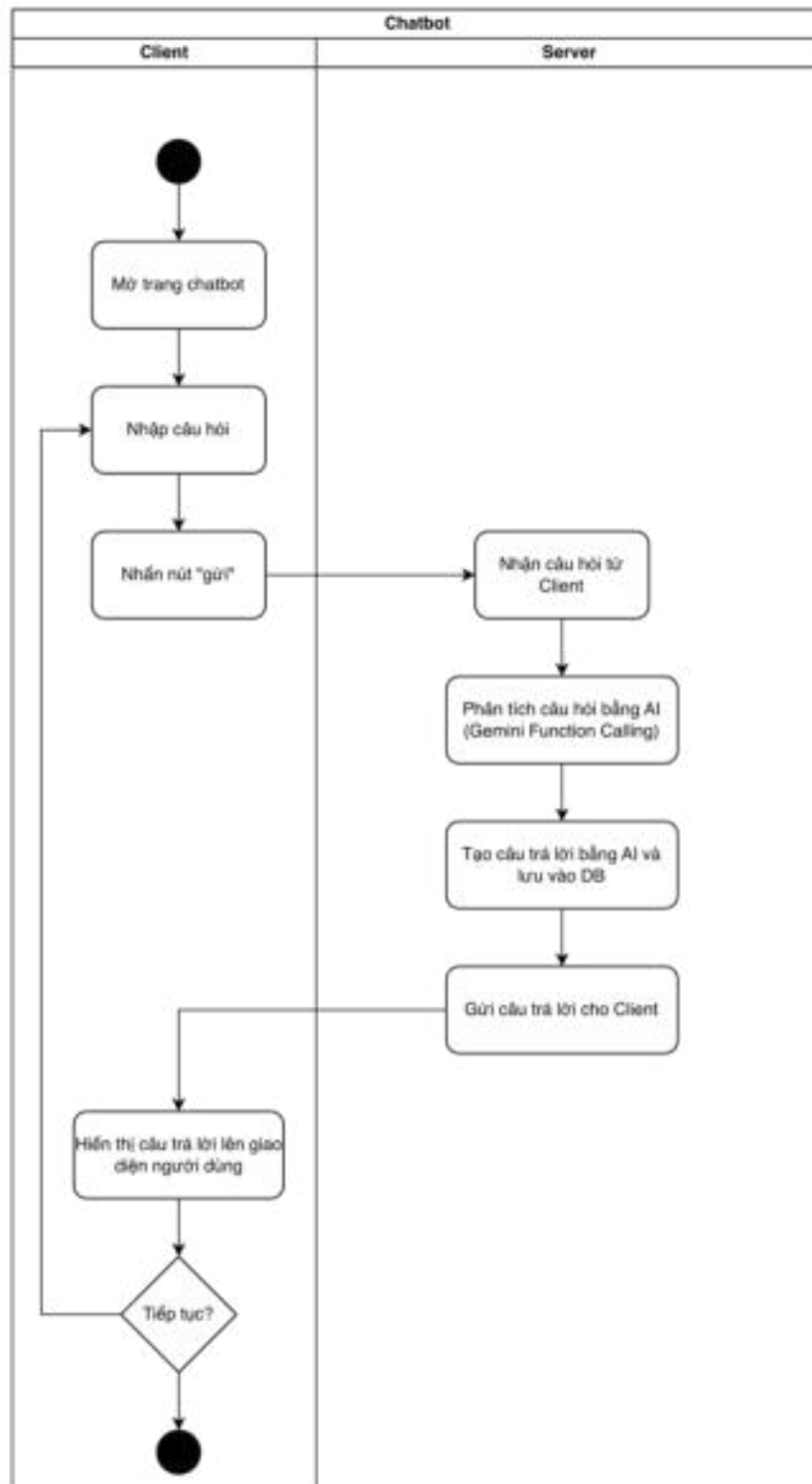
Hình 2.12 Biểu đồ hoạt động chức năng "Quản lí người dùng"

Biểu đồ hoạt động chức năng "Tạo mới thực đơn"



Hình 2.13 Biểu đồ hoạt động chức năng "Tạo mới thực đơn"

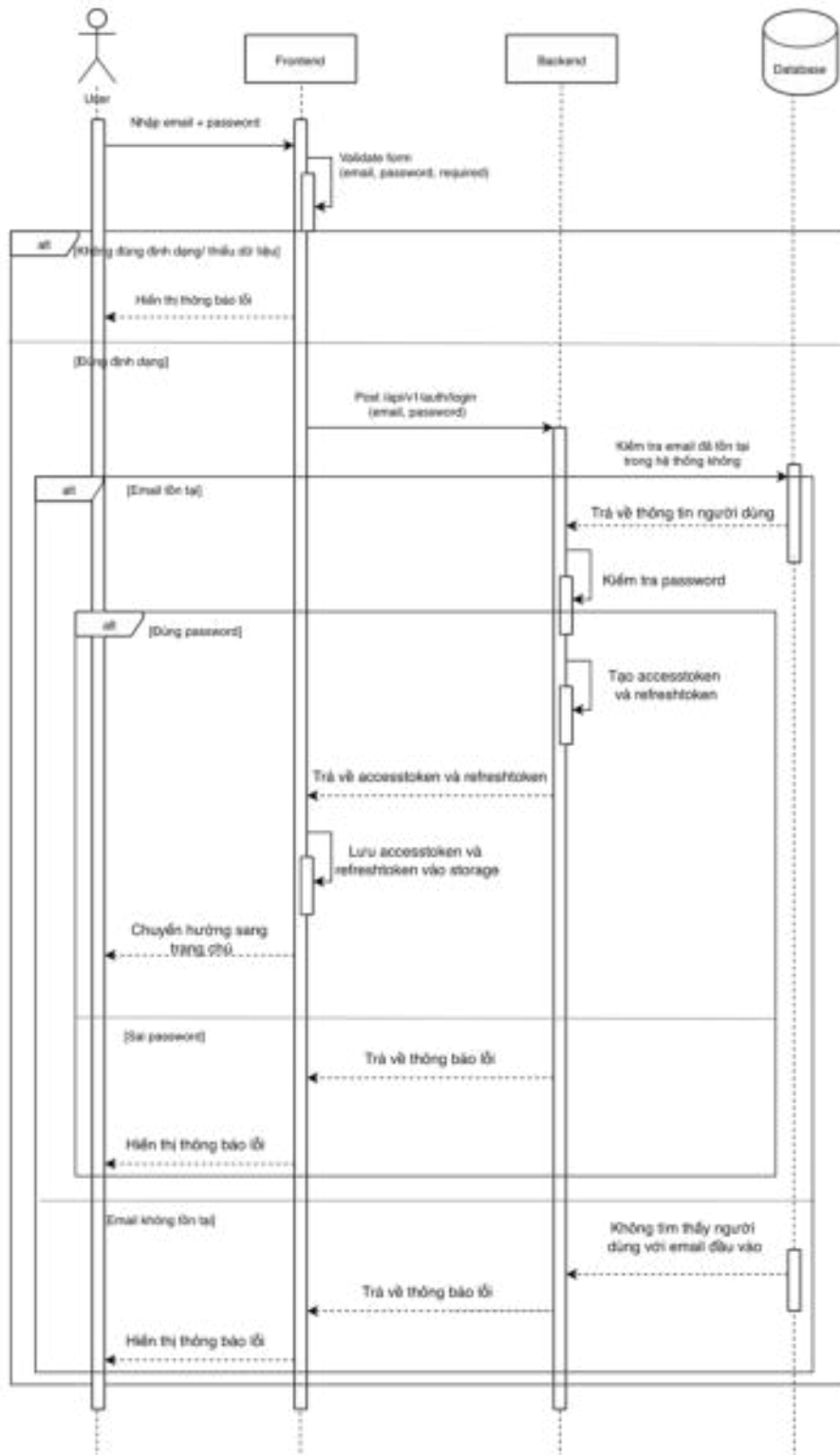
Biểu đồ hoạt động chức năng "Chatbot"



Hình 2.14 Biểu đồ hoạt động chức năng "Chatbot"

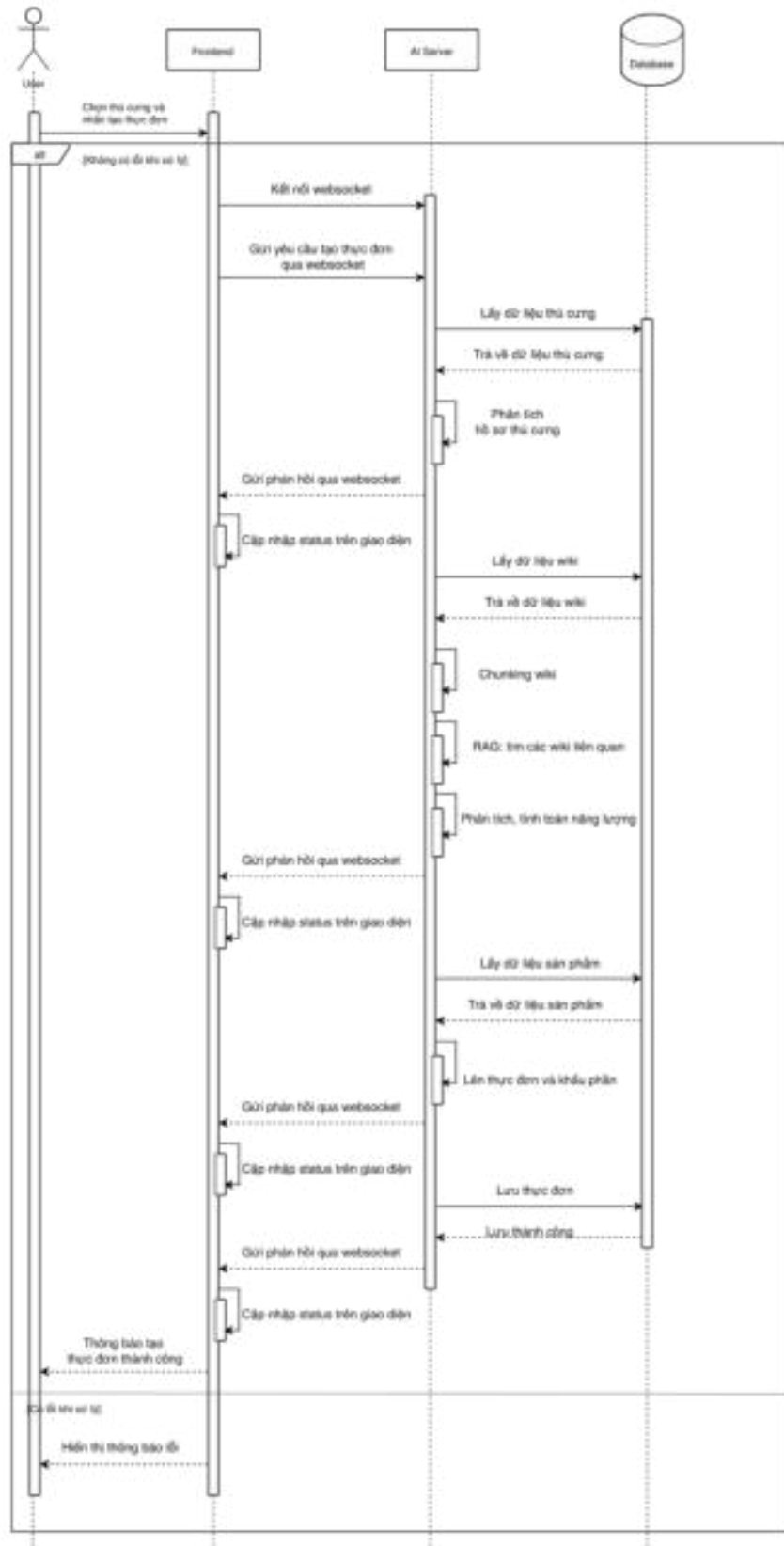
### 2.3.3. Biểu đồ tuần tự

Biểu đồ tuần tự chức năng “Đăng nhập”



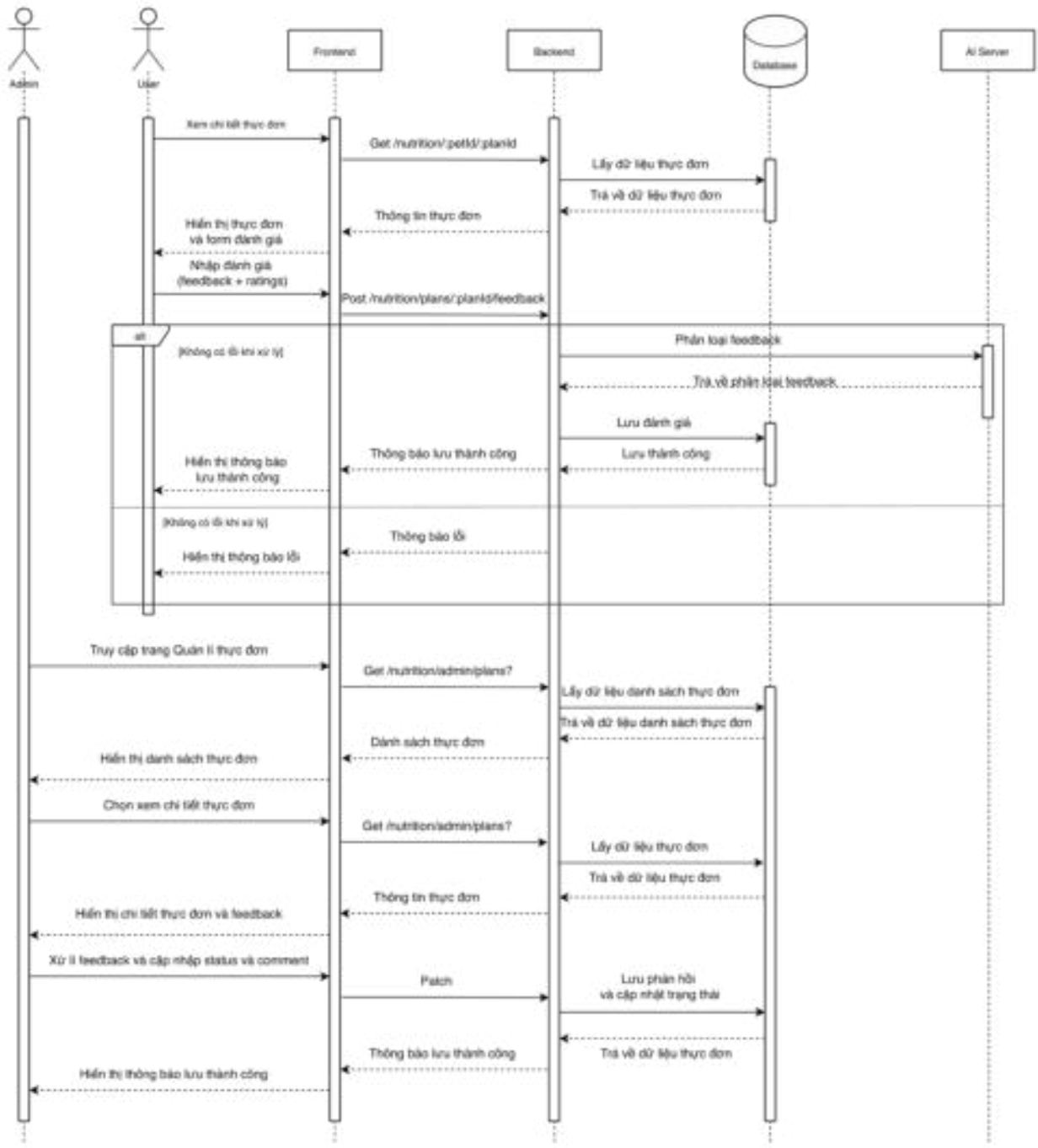
Hình 2.15 Biểu đồ tuần tự chức năng "Đăng nhập"

Biểu đồ tuần tự chức năng “Tạo thực đơn”



Hình 2.16 Biểu đồ tuần tự chức năng "Tạo thực đơn"

Biểu đồ tuần tự chức năng “Đánh giá và phản hồi thực đơn”



Hình 2.17 Biểu đồ tuần tự chức năng "Đánh giá và phản hồi thực đơn"



## Danh sách các bảng dữ liệu

Bảng 2.9 Danh sách các bảng dữ liệu

Tên bảng	Mô tả
users	Lưu thông tin về người dùng hệ thống
pets	Lưu thông tin thú cưng
carts	Lưu thông tin giỏ hàng
cart_items	Lưu chi tiết sản phẩm và số lượng của giỏ hàng tương ứng
orders	Lưu thông tin đơn hàng
order_items	Lưu chi tiết sản phẩm và số lượng của đơn hàng tương ứng
plans	Lưu thực đơn dinh dưỡng
products	Lưu thông tin sản phẩm
product_packs	Lưu thông tin gói của sản phẩm tương ứng
wiki_pages	Lưu thông tin cơ bản của các trang wiki dinh dưỡng
wiki_page_versions	Lưu các phiên bản chỉnh sửa của các trang wiki
wiki_releases	Lưu phiên bản publish
wiki_page_release	Lưu các trang wiki được publish theo từng phiên bản publish
chat_sessions	Lưu thông tin phiên trò chuyện chatbot
chat_messages	Lưu nội dung chi tiết các tin nhắn trong từng phiên trò chuyện
notifications	Lưu các thông báo trong hệ thống

## Mô tả chi tiết giữa các bảng

Bảng 2.10 Bảng users

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID người dùng

email	Varchar		Email người dùng
password	Varchar		Mật khẩu được mã hoá
address	Varchar		Địa chỉ người dùng
role	Enum		Vai trò người dùng
phone	Varchar		SDT người dùng
status	Enum		Trạng thái tài khoản
first_name	Varchar		Tên người dùng
last_name	Varchar		Họ và tên đệm người dùng
created_at	Date		Thời điểm tạo tài khoản
updated_at	Date		Thời điểm cập nhật gần nhất

Bảng 2.11 Bảng pets

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID thú cưng
user_id	Varchar		ID người dùng
name	Varchar		Tên thú cưng
species	Enum		Loài thú cưng
age_months	Integer		Tuổi thú cưng
weight_kg	Float		Cân nặng thú cưng
bcs	Integer		Điểm tình trạng cơ thể
neutered	Boolean		Trạng thái triệt sản
allergies	Varchar		Thông tin dị ứng
conditions	Varchar		Tình trạng sức khỏe
created_at	Date		Thời điểm tạo hồ sơ thú cưng
updated_at	Date		Thời điểm cập nhật hồ sơ thú cưng
expected_weight_kg	Float		Cân nặng mục tiêu
gender	Enum		Giới tính của thú cưng
note	Varchar		Ghi chú

breed	Enum		Giống thú cưng (nhỏ, lớn, ...)
activity_level	Enum		Mức độ hoạt động
goal	Enum		Mục tiêu
life_stage	Enum		Giai đoạn sống
gestation_week	Integer		Tuần mang thai
lactation_week	Integer		Tuần cho con bú
num_puppies	Integer		Số con

Bảng 2.12 Bảng products

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID sản phẩm
name	Varchar		Tên sản phẩm
description	Text		Mô tả sản phẩm
life_stage	Enum		Giai đoạn phát triển phù hợp
species	Enum		Loài phù hợp
food_type	Enum		Loại sản phẩm
moisture_pct	Float		Mức độ ẩm
allergens	Varchar		Thông tin về thành phần dị ứng
created_at	Date		Thời điểm tạo sản phẩm
updated_at	Date		Thời điểm cập nhật sản phẩm
nutritional_value_analysis	Text		Thông tin phân tích giá trị dinh dưỡng
dosage_recommendation	Text		Khuyến nghị liều lượng sử dụng
storage_instruction	Text		Hướng dẫn bảo quản
image	Text		Ảnh sản phẩm
usage	Text		Hướng dẫn sử dụng
ingredients	Text		Danh sách thành phần nguyên liệu

Bảng 2.13 Bảng plans

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID thực đơn
pet_id	Varchar		ID thú cưng
created_at	Date		Thời điểm tạo thực đơn
updated_at	Date		Thời điểm cập nhật thực đơn
daily_calories	Float		Tổng năng lượng cần thiết hằng ngày
hydration_target	Float		Mục tiêu nước
wiki_version	Varchar		Phiên bản wiki làm cơ sở
plan_nutrient_targets	Json		Mục tiêu dinh dưỡng của thực đơn
plan_cooking_guides	Json		Hướng dẫn chế biến
plan_meal_entries	Json		Khẩu phần ăn chi tiết
plan_items	Json		Danh sách sản phẩm đề xuất
goal	Enum		Mục tiêu thực đơn
note	Text		Ghi chú tổng quan
feedback_type	Enum		Loại đánh giá
feedback_payload	Text		Đánh giá thực đơn
feedback_status	Enum		Trạng thái đánh giá
ingredients_snapshot	Json		Bản ghi lại nguyên liệu được sử dụng tạo thực đơn
pet_profile_snapshot	Json		Bản ghi lại thông tin thú cưng tạo thực đơn
feedback_created_at	Date		Thời điểm gửi phản hồi
feedback_reviewed_at	Date		Thời điểm phản hồi được xử lý
admin_comment	Text		Phản hồi của admin
feedback_text	Text		Nội dung đánh giá của người dùng

Bảng 2.14 Bảng wiki\_pages

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID wiki
parent_id	Varchar		ID wiki cha
display_order	Integer		Thứ tự hiển thị
slug	Varchar		Định danh url
deleted_at	Date		Thời điểm xóa wiki

Bảng 2.15 Bảng wiki\_page\_versions

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID phiên bản của wiki
wiki_page_id	Varchar		ID wiki tương ứng
nav_label	Varchar		Nội dung hiển thị trên menu
title	Varchar		Tiêu đề nội dung
content_md	Text		Nội dung chi tiết
version	Integer		Phiên bản của wiki
created_at	Date		Thời điểm tạo phiên bản nội dung wiki
deleted_at	Date		Thời điểm xóa phiên bản nội dung wiki

Bảng 2.16 Bảng wiki\_releases

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID đợt phát hành wiki
version	Integer		Phiên bản của bản phát hành wiki
title	Varchar		Tiêu đề phát hành
created_at	Date		Thời điểm tạo đợt phát hành wiki
deleted_at	Date		Thời điểm đợt phát hành bị xóa

Bảng 2.17 Bảng wiki\_page\_releases

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID liên kết đợt phát hành và phiên bản nội dung
wiki_release_id	Varchar		ID đợt phát hành wiki
wiki_page_version_id	Varchar		ID phiên bản nội dung wiki

Bảng 2.18 Bảng chat\_sessions

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID phiên trò chuyện
user_id	Varchar		ID người dùng
pet_id	Varchar		ID thú cưng
created_at	Date		Thời điểm bắt đầu phiên trò chuyện
updated_at	Date		Thời điểm cập nhật phiên trò chuyện

Bảng 2.19 Bảng chat\_messages

Tên trường	Kiểu dữ liệu	Ràng buộc	Mô tả
id	Varchar	Khoá chính	ID tin nhắn
session_id	Varchar		ID phiên trò chuyện
created_at	Date		Thời điểm tạo tin nhắn
updated_at	Date		Thời điểm cập nhật tin nhắn
role	Enum		Vai trò người gửi tin nhắn
content	Text		Nội dung tin nhắn

## 2.4. Ứng dụng LLM triển khai AI Agent

Hệ thống Dinh dưỡng Thú cưng áp dụng mô hình AI Agent với 3 lớp chính: LLM (bộ não ngôn ngữ), Memory (dữ liệu/ bộ nhớ) và Tool/ Action. Trong phạm vi đề tài, sử dụng Google Gemini API làm nền tảng LLM chính cho toàn bộ hệ thống.

### 2.4.1. Cơ sở nghiên cứu và tham khảo

Để xây dựng hệ thống tư vấn dinh dưỡng thú cưng, nghiên cứu được thực hiện dựa trên việc tổng hợp và phân tích các nguồn dữ liệu khoa học uy tín trong lĩnh vực dinh dưỡng thú y. Trong đó, nguồn tài liệu chính thức được sử dụng là bộ hướng dẫn dinh dưỡng do FEDIAF ban hành.

FEDIAF là Liên đoàn Công nghiệp Thức ăn Thú cưng Châu Âu, được công nhận rộng rãi như là tài liệu tham khảo chính về dinh dưỡng thú cưng tại châu Âu. Nghiên cứu sử dụng tài liệu "FEDIAF Nutritional Guidelines" phiên bản 2024 làm nguồn tham khảo chính với các nội dung:

- Nghiên cứu công thức tính toán năng lượng duy trì (MER): FEDIAF cung cấp công thức tính  $MER = k \times BW^{0.75}$ , trong đó  $k$  là hệ số năng lượng phụ thuộc vào loài, giai đoạn sống và mức độ hoạt động. Ví dụ, với chó hoạt động bình thường  $k = 110 \text{ kcal/kg}^{0.75}$ ; với mèo trong nhà bình thường:  $k = 85 \text{ kcal/kg}^{0.75}$ .
- Phương trình dự đoán cân nặng trưởng thành: Các phương trình logarithm tự nhiên để dự đoán cân nặng trưởng thành cho chó con dựa trên tuổi và cân nặng hiện tại.
- Hệ thống đánh giá tình trạng cơ thể (BCS - Body Condition Score): Thang điểm từ 1-9 để đánh giá tình trạng dinh dưỡng của thú cưng.
- Danh sách nguyên liệu an toàn và phù hợp: Các khuyến nghị về nguyên liệu dinh dưỡng có thể sử dụng trong khẩu phần ăn, cũng như các thành phần cần lưu ý đối với chó và mèo

Các tài liệu và kiến thức dinh dưỡng được tổng hợp và tổ chức thành một hệ thống Wiki nội bộ, lưu trữ trong cơ sở dữ liệu của hệ thống nhằm phục vụ cho quá trình truy xuất thông tin và cung cấp các thông tin dinh dưỡng cho người dùng. Dữ liệu được xây dựng với các đặc điểm:

- Phân loại theo chủ đề: nhu cầu năng lượng, mục tiêu dinh dưỡng, nguyên liệu, sản phẩm bổ sung.
- Cấu trúc Markdown: hỗ trợ chia nhỏ nội dung thành các đoạn văn bản (chunks).
- Quản lý phiên bản (versioning): theo dõi lịch sử thay đổi, đảm bảo tính nhất quán và cập nhật của dữ liệu theo thời gian.

#### 2.4.2. Mô hình ngôn ngữ lớn (LLM) sử dụng trong hệ thống

Hệ thống sử dụng Gemini 2.5 Flash, đây là mô hình LLM (Large Language Model) do Google phát triển được thiết kế nhằm hỗ trợ các tác vụ xử lý ngôn ngữ tự nhiên và hội thoại thông minh trong thời gian thực.

Hệ thống chọn Gemini 2.5 Flash làm mô hình trung tâm cho chức năng chatbot nhờ sở hữu nhiều ưu điểm phù hợp với yêu cầu triển khai thực tế. Mô hình có khả năng phản hồi nhanh, giúp cải thiện trải nghiệm tương tác của người dùng, đồng thời tối ưu chi phí vận hành so với các mô hình có kích thước lớn hơn.

Ngoài khả năng sinh ngôn ngữ, hệ thống còn tận dụng tính năng Gemini Function Calling, cho phép mô hình xác định khi nào cần gọi các hàm nghiệp vụ như tính toán nhu cầu năng lượng, đề xuất sản phẩm hoặc tra cứu dữ liệu dinh dưỡng. Nhờ đó chatbot không chỉ trả lời dựa trên văn bản mà còn có thể thực hiện được các tác vụ chuyên môn một cách chính xác và hiệu quả.

#### **2.4.3. Tiền xử lý dữ liệu bằng Chunking**

Sau khi tổng hợp dữ liệu và xây dựng wiki dinh dưỡng (Wiki FEDIAF), hệ thống tiến hành thực hiện chunking trên toàn bộ tài liệu. Đây là bước tiền xử lý quan trọng nhằm chuẩn bị dữ liệu cho quá trình truy xuất thông tin và sinh câu trả lời trong mô hình RAG.

Trong hệ thống, chunking được thực hiện theo từng page/version của wiki nhằm đảm bảo dữ liệu được tổ chức rõ ràng, dễ quản lý và thuận tiện cho việc truy xuất theo từng chủ đề cụ thể. Bên cạnh đó, để hạn chế việc mất ngữ cảnh giữa các đoạn văn bản liền kề, hệ thống áp dụng cơ chế overlap, trong đó các chunk được tạo ra sẽ có một phần nội dung giao nhau nhất định. Cách tiếp cận này giúp duy trì tính liên kết thông tin và cải thiện hiệu quả truy xuất trong các trường hợp câu hỏi liên quan đến nhiều đoạn nội dung khác nhau.

Sau khi chia nhỏ, mỗi chunk được lưu trữ dưới dạng JSON, bao gồm nội dung văn bản và các metadata đi kèm như page, version, hoặc nguồn tài liệu giúp hệ thống dễ dàng quản lý dữ liệu và phục vụ cho các bước embedding và semantic search ở giai đoạn tiếp theo.

#### **2.4.4. Vector hóa nội dung bằng Embedding**

Sau bước chunking, tiến hành chuyển đổi các đoạn văn bản đã chia nhỏ thành dạng vector embedding để phục vụ cho quá trình truy xuất ngữ nghĩa trong mô hình RAG.

Hệ thống sử dụng model AITeamVN/Vietnamese Embedding để thực hiện embedding cho các chunk văn bản. Đây là mô hình được tối ưu hóa cho tiếng Việt, phù hợp với nội dung wiki dinh dưỡng và hỗ trợ hiệu quả cho việc xử lý các truy vấn từ người dùng bằng tiếng Việt.

#### **2.4.5. Truy xuất và sinh phản hồi với mô hình RAG**

Hệ thống triển khai mô hình RAG (Retrieval-Augmented Generation) nhằm nâng cao độ chính xác khi chatbot trả lời người dùng. Thay vì chuyển trực tiếp câu hỏi vào LLM để sinh phản hồi ngay lập tức, hệ thống thực hiện thêm bước truy xuất dữ liệu để

đảm bảo câu trả lời dựa trên dữ liệu tham khảo trong cơ sở dữ liệu. Nhờ vậy, kết quả được trả lời bằng chatbot sẽ dựa trên thông tin đã truy xuất từ cơ sở dữ liệu, giúp giảm sai lệch và tăng độ tin cậy trong quá trình tư vấn dinh dưỡng.

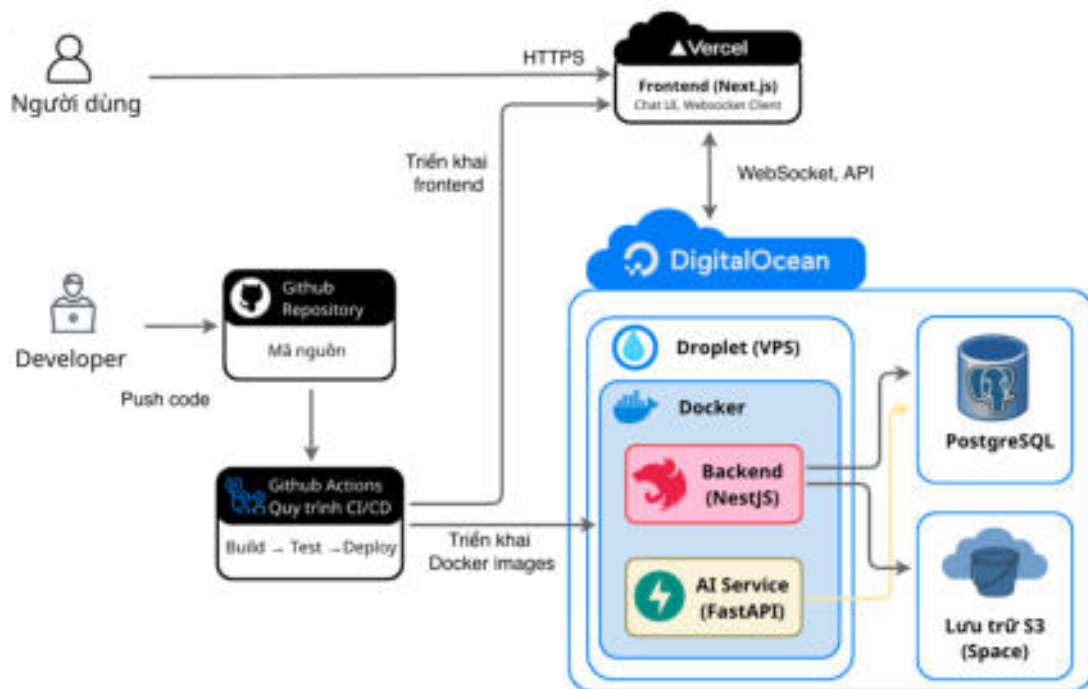
## Chương 3: TRIỂN KHAI THỰC TẾ VÀ ĐÁNH GIÁ KẾT QUẢ

### 3.1. Tổ chức triển khai hệ thống

#### 3.1.1. Công cụ triển khai

- Frontend: Nextjs
- Backend: Nestjs, FastAPI
- Giao tiếp realtime: Websocket
- Database: Postgresql
- CI/CD: Sử dụng Github Actions
- Lưu trữ hình ảnh: S3 Storage của DigitalOcean
- Đóng gói và triển khai ứng dụng: Docker

#### 3.1.2. Triển khai hệ thống



Hình 3.1 Mô hình triển khai hệ thống

Trong giai đoạn phát triển, mã nguồn của hệ thống được quản lý tập trung trên Github. Khi có thay đổi, cập nhật về mã nguồn và được đẩy lên Github, Github Actions sẽ tự động kích hoạt quy trình CI/CD, bao gồm các bước xây dựng (build), kiểm thử (test) và triển khai (deploy), giúp giảm thiểu thao tác thủ công và hạn chế lỗi trong quá trình triển khai.

Phần giao diện người dùng (Frontend) được xây dựng bằng Next.js và triển khai trên nền tảng Vercel. Nhờ đó ứng dụng web có khả năng phân phối nội dung nhanh, hỗ

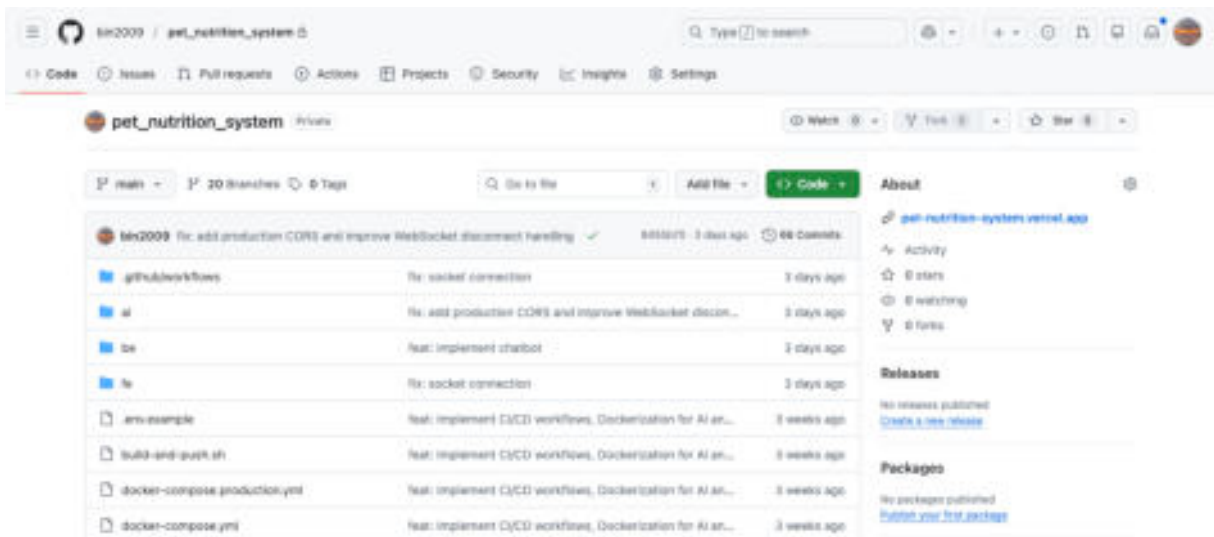
trợ giao tiếp thời gian thực thông qua WebSocket và tương tác với hệ thống Backend và AI thông qua các API bảo mật bằng giao thức HTTPS.

Phần Backend và AI được triển khai trên hạ tầng DigitalOcean. Cụ thể, các thành phần Backend (NestJS) và AI Service (FastAPI) được đóng gói dưới dạng Docker containers và chạy trên một Droplet (VPS). Cách triển khai này giúp đảm bảo tính nhất quán của môi trường chạy, đồng thời dễ dàng mở rộng hoặc nâng cấp hệ thống khi cần thiết.

Hệ thống sử dụng cơ sở dữ liệu PostgreSQL để lưu trữ dữ liệu nghiệp vụ, đảm bảo tính toàn vẹn và khả năng truy vấn hiệu quả. Ngoài ra, các dữ liệu hình ảnh được lưu trữ trên dịch vụ Object Storage S3 (DigitalOcean Spaces), giúp tối ưu hóa dung lượng lưu trữ và hiệu năng truy xuất.

### 3.1.3. Quản lý mã nguồn (github)

Hệ thống sử dụng Github để quản lý mã nguồn và kiểm soát phiên bản trong suốt quá trình phát triển.

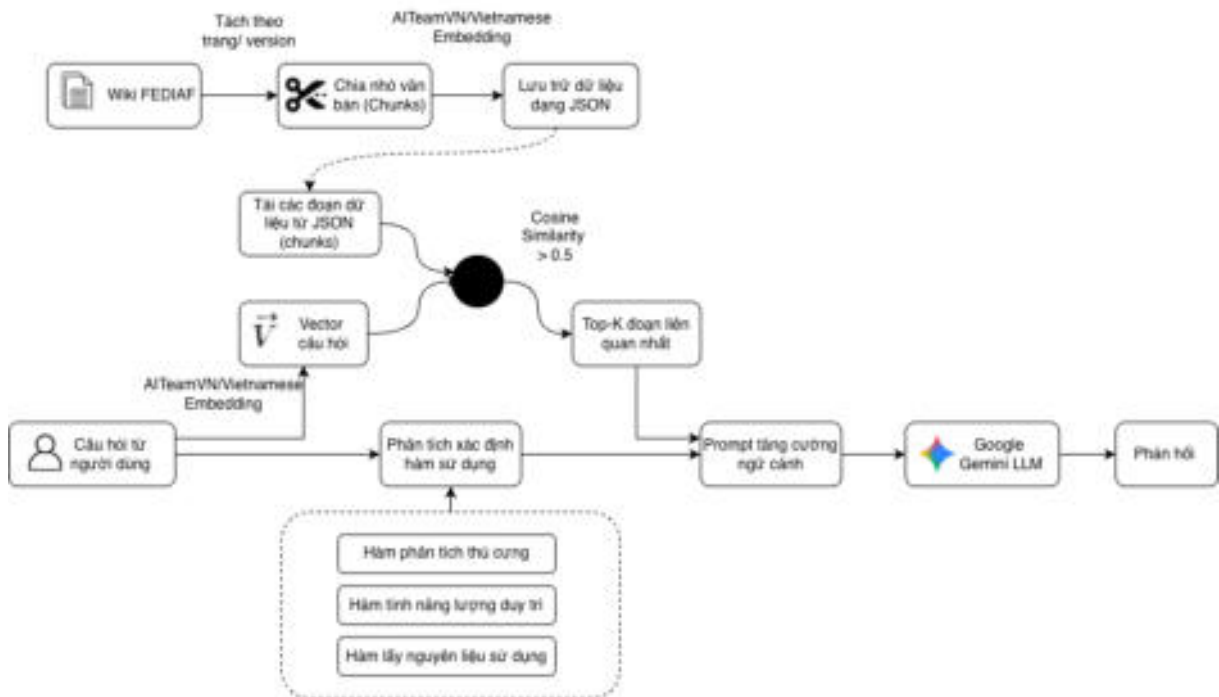


Hình 3.2 Quản lý mã nguồn bằng Github

## 3.2. Kết quả triển khai chức năng AI

### 3.2.1. Chức năng chatbot

#### a. Sơ đồ triển khai chức năng



Hình 3.3 Sơ đồ triển khai chức năng Chatbot

#### Quy trình hoạt động của chức năng được thực hiện như sau:

Người dùng nhập câu hỏi hoặc yêu cầu tư vấn thông qua giao diện chatbot trên hệ thống.

- Hệ thống tiếp nhận truy vấn và thực hiện phân tích nội dung câu hỏi để xác định mục đích sử dụng (ví dụ: phân tích thú cưng, hỏi về dinh dưỡng, tra cứu thông tin,...).
- Gemini Function Calling được áp dụng để lựa chọn hàm nghiệp vụ phù hợp trong trường hợp truy vấn yêu cầu xử lý chuyên môn (tính toán năng lượng, phân tích khẩu phần,...).
- Nếu truy vấn cần tham chiếu kiến thức dinh dưỡng, hệ thống thực hiện truy xuất dữ liệu từ Wiki FEDIAF bằng cơ chế RAG:
  - Chuyển câu hỏi thành vector embedding
  - So sánh cosine similarity với các chunk đã lưu trữ
  - Lựa chọn Top-K đoạn liên quan nhất
  - Các đoạn thông tin truy xuất được sẽ được đưa vào prompt tăng cường ngữ cảnh trước khi gửi đến mô hình Gemini 2.5 Flash.

- Mô hình Gemini sinh phản hồi cuối cùng dựa trên câu hỏi của người dùng kết hợp với ngữ cảnh từ wiki dinh dưỡng.
- Kết quả trả lời được hiển thị lại cho người dùng, đảm bảo tính chính xác và nhất quán trong quá trình tư vấn.

*b. Kiểm thử và nhận xét*

Nội dung thử nghiệm	Điều kiện thử nghiệm	Kết quả thu được	Đánh giá
Kiểm thử chatbot trả lời câu hỏi dinh dưỡng cơ bản	User 1 đặt câu hỏi về nhu cầu năng lượng thú cưng	Chatbot phản hồi đúng nội dung tư vấn và thông tin liên quan	Chức năng hoạt động ổn định
Kiểm thử tính nhất quán khi hỏi lặp lại nhiều lần	User 1 sử dụng cùng hồ sơ thú cưng và đặt lại câu hỏi ở nhiều thời điểm	Kết quả trả lời đồng bộ	Đảm bảo tính nhất quán của kết quả
Kiểm thử tính nhất quán giữa nhiều tài khoản người	User 1 và User 2 có cùng thông tin thú cưng và đặt cùng nội dung câu hỏi	Chatbot đưa ra kết quả tương tự và nhất quán	Dữ liệu chính xác, không phụ thuộc tài khoản
Kiểm thử khả năng cập nhật tri thức theo wiki (Wiki FEDIAF)	User hỏi lần 1 cho kết quả A. Sau khi Admin cập nhật nội dung wiki, User hỏi lại cùng câu hỏi	Chatbot phản hồi theo dữ liệu phiên bản mới, và cho ra kết quả B đã được điều chỉnh	Hệ thống truy xuất tri thức đúng đắn

**Kết quả thu được:**

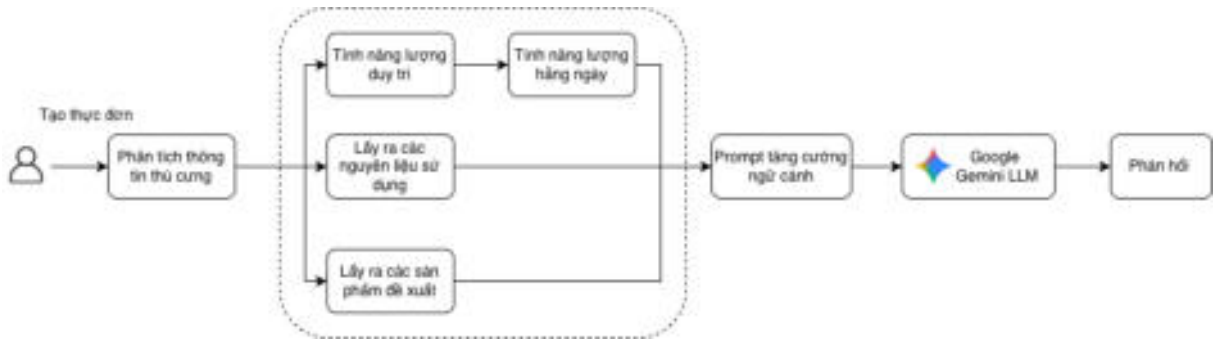
- Chatbot AI hoạt động ổn định và đáp ứng đúng các yêu cầu đề ra.
- Người dùng có thể đặt câu hỏi và nhận tư vấn trực tiếp thông qua giao diện web.
- Hệ thống sử dụng thông tin thú cưng đầu vào (loài, cân nặng, độ tuổi, mục tiêu chăm sóc) để phân tích và phản hồi phù hợp.
- Khi người dùng hỏi lặp lại nhiều lần với cùng hồ sơ thú cưng, chatbot vẫn đảm bảo kết quả nhất quán, không xảy ra sai lệch dữ liệu.
- Cơ chế RAG hỗ trợ truy xuất thông tin từ Wiki FEDIAF trước khi sinh phản hồi, giúp tăng độ chính xác và hạn chế trả lời thiếu căn cứ.
- Chatbot phản hồi nhanh và duy trì kết nối ổn định thông qua WebSocket trong quá trình hội thoại liên tục.

**Nhận xét:**

Kết quả thử nghiệm cho thấy chatbot AI đáp ứng tốt vai trò tư vấn và phân tích dinh dưỡng thú cưng. Việc kết hợp RAG và Function Calling giúp nâng cao độ tin cậy của phản hồi, đồng thời tăng giá trị ứng dụng thực tiễn của hệ thống.

**3.2.2. Chức năng Tạo thực đơn dinh dưỡng**

*a. Sơ đồ triển khai chức năng*



Hình 3.4 Sơ đồ triển khai chức năng Tạo thực đơn dinh dưỡng

Quy trình hoạt động của chức năng được thực hiện như sau:

- Người dùng nhập thông tin thú cưng và lựa chọn mục tiêu dinh dưỡng (tăng cân, duy trì, giảm cân,...).
- Hệ thống tính toán nhu cầu năng lượng (MER) và các chỉ số dinh dưỡng cần thiết dựa trên tiêu chuẩn tham khảo từ Wiki FEDIAF.
- Gemini Function Calling được sử dụng để gọi các hàm nghiệp vụ liên quan như tính khẩu phần, phân tích nguyên liệu và đề xuất thực đơn.
- Hệ thống kết hợp cơ chế RAG để truy xuất thông tin dinh dưỡng từ kho tri thức nhằm đảm bảo thực đơn có căn cứ khoa học.
- Kết quả thực đơn được tổng hợp và hiển thị cho người dùng dưới dạng gợi ý khẩu phần ăn theo ngày hoặc theo bữa.

*b. Kiểm thử và nhận xét*

Nội dung thử nghiệm	Điều kiện thử nghiệm	Kết quả thu được	Đánh giá
Kiểm thử tạo thực đơn cơ bản	User nhập chó 5kg, mục tiêu duy trì	Hệ thống sinh thực đơn phù hợp theo MER	Hoạt động ổn định
Kiểm thử thay đổi mục tiêu dinh dưỡng	User đổi mục tiêu từ duy trì sang giảm cân	Khẩu phần và năng lượng được điều chỉnh tương ứng	Đúng theo yêu cầu
Kiểm thử với nhiều tài khoản	User 1 và User 2 tạo thực đơn với cùng dữ liệu đầu vào	Kết quả sinh ra nhất quán giữa các tài khoản	Đảm bảo đồng bộ

Kiểm thử cập nhật wiki nguyên liệu	Admin cập nhật thông tin nguyên liệu trong Wiki FEDIAF	Thực đơn mới phản ánh đúng dữ liệu cập nhật	Dữ liệu được cập nhật hiệu quả
------------------------------------	--	---	--------------------------------

Kết quả thu được:

- Chức năng lập thực đơn dinh dưỡng được triển khai ổn định và đáp ứng đúng yêu cầu đề ra.
- Người dùng có thể nhập thông tin thú cưng (loài, cân nặng, độ tuổi, mục tiêu chăm sóc) để hệ thống tạo khẩu phần ăn phù hợp.
- Hệ thống thực hiện tính toán nhu cầu năng lượng (MER) và các chỉ số dinh dưỡng dựa trên tiêu chuẩn tham khảo từ Wiki FEDIAF.
- Khi người dùng thay đổi mục tiêu dinh dưỡng (duy trì, tăng cân, giảm cân), thực đơn được điều chỉnh tương ứng theo dữ liệu đầu vào.
- Trong quá trình thử nghiệm, Function Calling hỗ trợ gọi các hàm nghiệp vụ như tính khẩu phần và gợi ý thực đơn một cách chính xác thay vì chỉ trả lời mô tả chung.
- Khi dữ liệu wiki được cập nhật hoặc bổ sung nguyên liệu, kết quả thực đơn mới phản ánh đúng nội dung đã thay đổi trong kho tri thức.
- Hệ thống phản hồi nhanh và hiển thị thực đơn rõ ràng, đảm bảo trải nghiệm sử dụng liên tục cho người dùng.

Nhận xét:

Kết quả đánh giá cho thấy chức năng lập thực đơn dinh dưỡng đáp ứng tốt yêu cầu cá nhân hóa theo từng thú cưng. Việc kết hợp dữ liệu Wiki FEDIAF, cơ chế RAG và Function Calling giúp thực đơn được tạo ra có căn cứ khoa học, nâng cao độ chính xác và tính ứng dụng thực tiễn của hệ thống.

### 3.3. Đặc tả API

API được tổ chức theo module chức năng, ví dụ /user, /admin, /orders,...

The screenshot displays a list of API endpoints categorized into three modules: 'auth', 'users', and 'admin/users'. Each endpoint is represented by a colored bar indicating its HTTP method (POST, GET, PUT, DELETE, PATCH) and a brief description of its function. For example, under 'auth', there are endpoints for verifying, registering, and logging in. Under 'users', there are endpoints for getting and updating the current user profile. The 'admin/users' section includes endpoints for statistics, listing all users, getting a user by ID, updating a user, deleting a user, and activating/deactivating users.

Module	Method	Endpoint	Description
auth	POST	/v1/auth/verify	
	POST	/v1/auth/register	
	POST	/v1/auth/login	
users	GET	/v1/users/me	Get current user profile
	PUT	/v1/users/me	Update current user profile
admin/users	GET	/v1/admin/users/statistics	Get user statistics for dashboard (Admin-only)
	GET	/v1/admin/users	Get all users (Admin-only)
	GET	/v1/admin/users/{userId}	Get user by ID (Admin-only)
	PUT	/v1/admin/users/{userId}	Update user (Admin-only)
	DELETE	/v1/admin/users/{userId}	Delete user (Admin-only)
	PATCH	/v1/admin/users/{userId}/activate	Activate user (Admin-only)
	PATCH	/v1/admin/users/{userId}/deactivate	Deactivate user (Admin-only)

Hình 3.5 Danh sách api module auth, user và admin

The screenshot displays a list of API endpoints for two modules: 'orders' and 'carts'. The 'orders' module includes endpoints for creating a new order, getting orders by user, updating order status, getting orders by admin, getting an order by ID, deleting an order, and getting an order by ID from an admin. The 'carts' module includes endpoints for getting a cart by user, creating a cart, deleting an item from a cart, and adding an item to a cart.

Module	Method	Endpoint	Description
orders	POST	/v1/orders	Create a new order
	GET	/v1/orders	Get all orders by user
	PATCH	/v1/orders/{orderId}/status	Update order status
	GET	/v1/orders/admin	Get all orders by admin
	GET	/v1/orders/{orderId}	Get order by ID
	DELETE	/v1/orders/{orderId}	Delete order
	GET	/v1/orders/admin/{orderId}	Get order by ID
	carts	GET	/v1/carts
POST		/v1/carts	Create cart
PUT		/v1/carts/items/{itemId}	Delete cart item quantity
DELETE		/v1/carts/items/{itemId}	Delete item from cart
POST		/v1/carts/items	Add item to cart

Hình 3.6 Danh sách api module orders và carts

The screenshot displays two sections of API documentation. The first section, titled 'products', lists 11 endpoints with their respective HTTP methods and descriptions: GET /v1/products/single (Retrieve a product of each name single), GET /v1/products (Retrieve all products), POST /v1/products (Create a new product), GET /v1/products/{productId} (Retrieve a product by ID), PUT /v1/products/{productId} (Update a product by ID), DELETE /v1/products/{productId} (Delete a product by ID), GET /v1/products/{productId}/packs (Retrieve all product packs by product ID), POST /v1/products/{productId}/packs (Create a new product pack by product ID), GET /v1/products/{productId}/packs/{packId} (Retrieve a product pack by product ID and pack ID), PUT /v1/products/{productId}/packs/{packId} (Update a product pack by product ID and pack ID), and DELETE /v1/products/{productId}/packs/{packId} (Delete a product pack by product ID and pack ID). The second section, titled 'payment', lists 5 endpoints: GET /v1/payment/{orderCode}, POST /v1/payment/{orderCode}/cancel, POST /v1/payment/{orderCode}/expired, GET /v1/payment/{orderCode}/resume, and POST /v1/payment/webhook (Notify PayPal webhook).

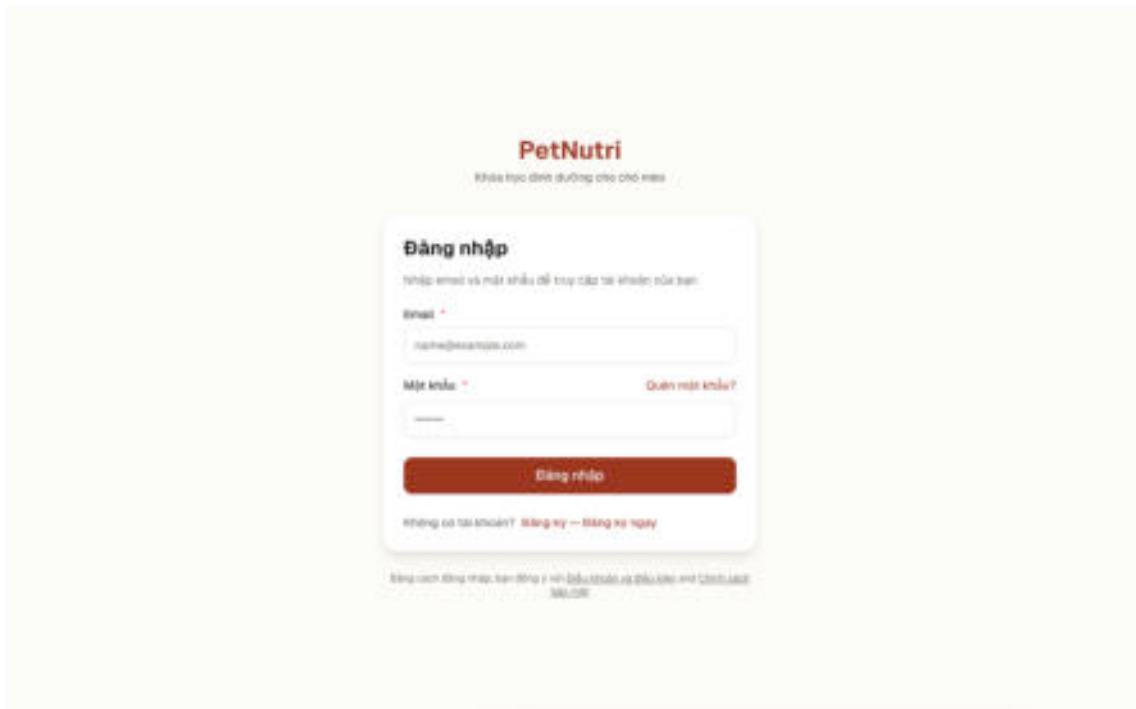
Hình 3.7 Danh sách api module products, payment

The screenshot displays two sections of API documentation. The first section, titled 'pets', lists 5 endpoints: GET /v1/pets (Get all pets by current user), POST /v1/pets (Create a new pet), GET /v1/pets/{petId} (Get pet by ID), PUT /v1/pets/{petId} (Update pet information), and DELETE /v1/pets/{petId} (Delete pet). The second section, titled 'wiki-modules', lists 15 endpoints: POST /v1/wiki-pages (Create a new wiki page), GET /v1/wiki-pages (Get all wiki pages), GET /v1/wiki-pages/public (Get all wiki pages), GET /v1/wiki-pages/releases (Get all release versions), POST /v1/wiki-pages/releases (Publish a new release), GET /v1/wiki-pages/affected (Get affected items that need reindexing), POST /v1/wiki-pages/releases/force-publish (Auto-publish all changes (includes Drafts) with force content), GET /v1/wiki-pages/pending (Get all pending wiki pages), GET /v1/wiki-pages/all-versions (Get all wiki pages with all versions), GET /v1/wiki-pages/tree (Get wiki pages as tree), GET /v1/wiki-pages/tree-public (Get wiki pages as tree), GET /v1/wiki-pages/public/{slug} (Get a wiki page by slug), GET /v1/wiki-pages/{slug} (Get a wiki page by slug), and PUT /v1/wiki-pages/{slug} (Update an existing wiki page).

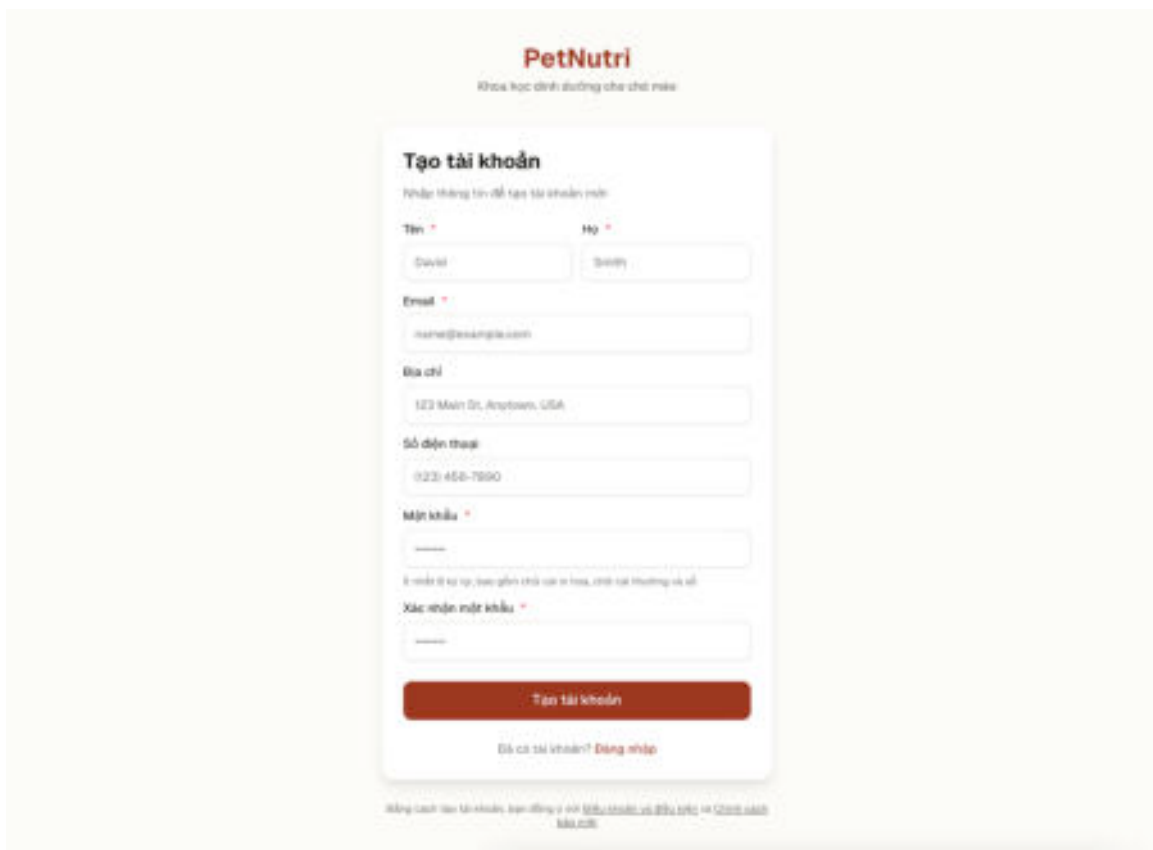
Hình 3.8 Danh sách api module pets và wiki

### 3.4. Giao diện website và chức năng

#### a. Màn hình đăng nhập, và đăng ký cho người dùng

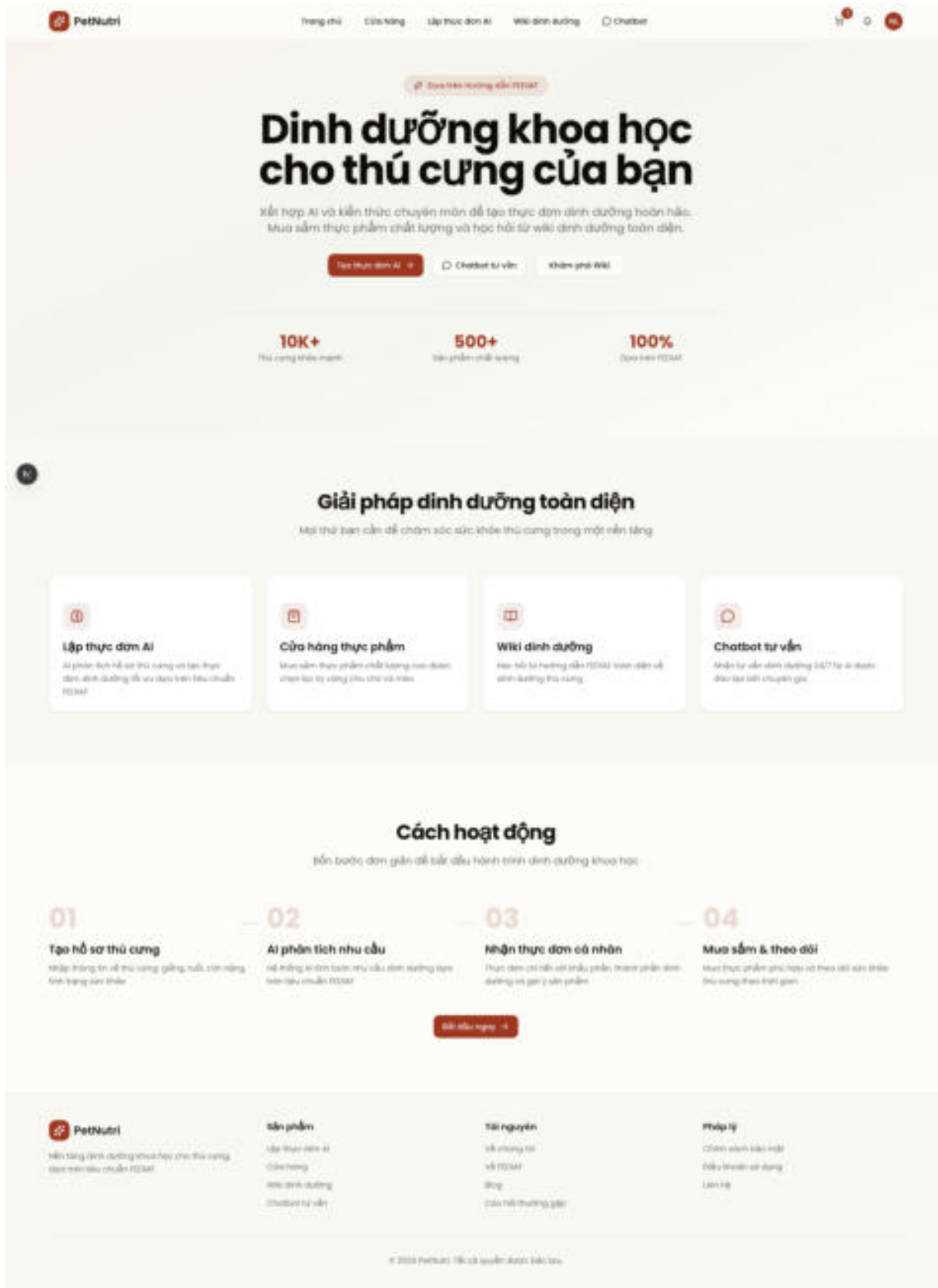


Hình 3.9 Màn hình login



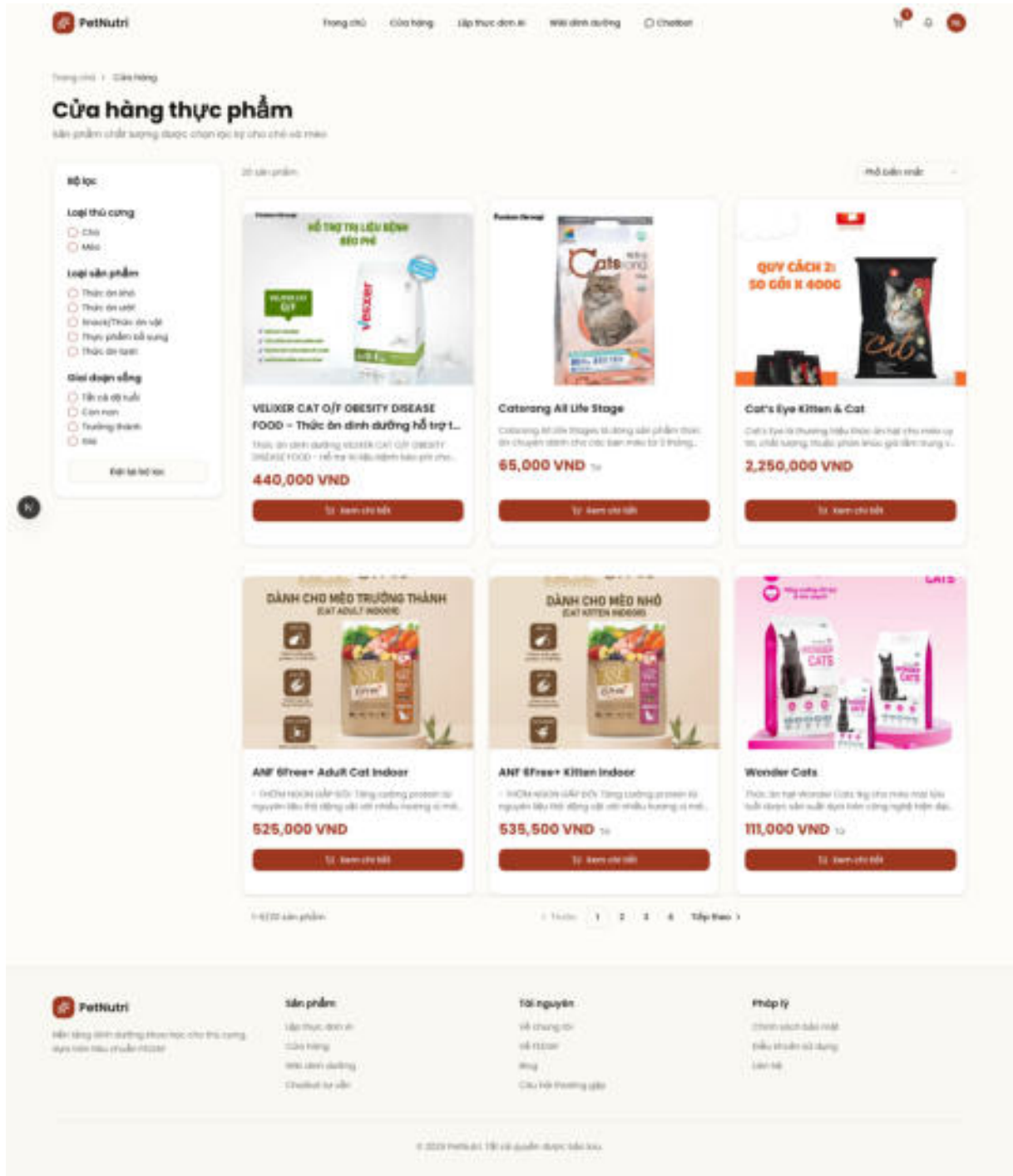
Hình 3.10 Màn hình đăng ký

b. Màn hình chính của hệ thống



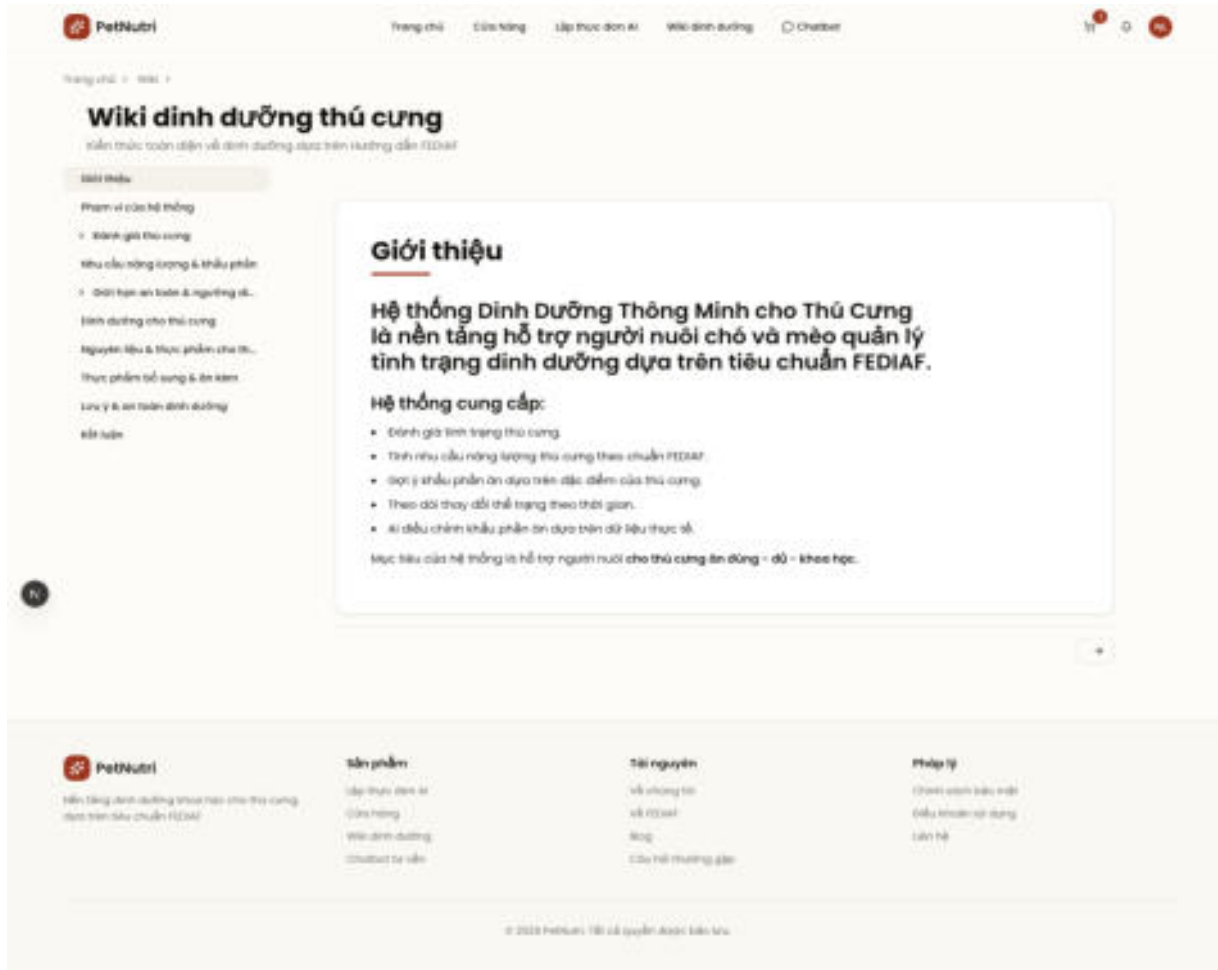
Hình 3.11 Màn hình trang chủ

c. Màn hình trang cửa hàng thực phẩm



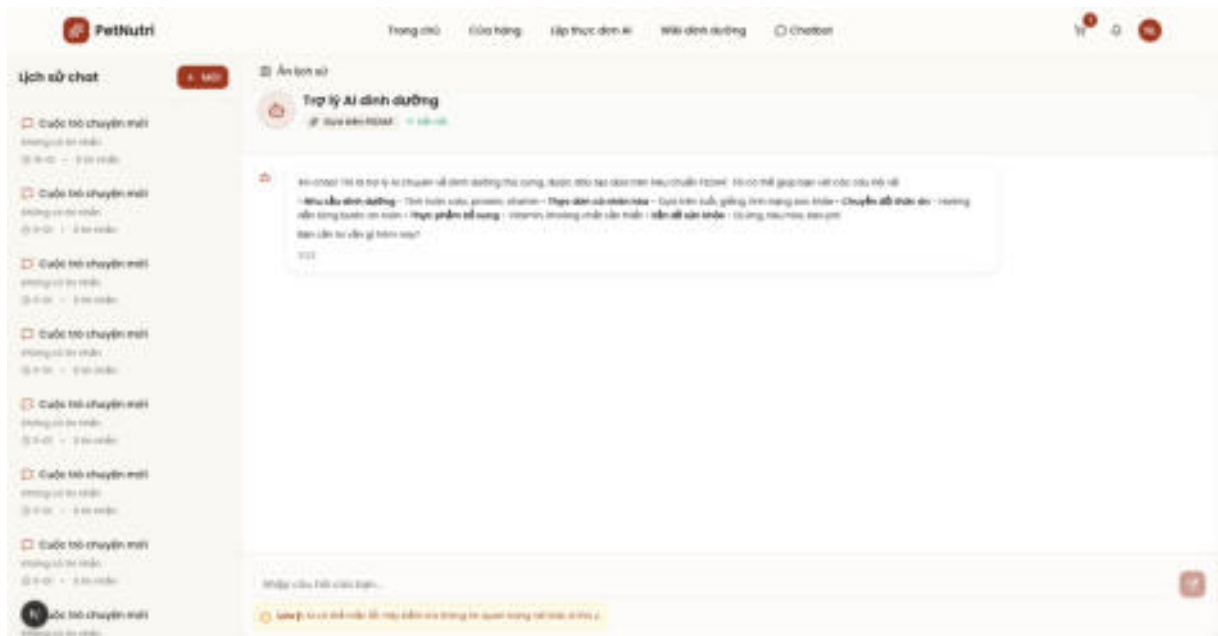
Hình 3.12 Màn hình cửa hàng thực phẩm

d. Màn hình wiki dinh dưỡng



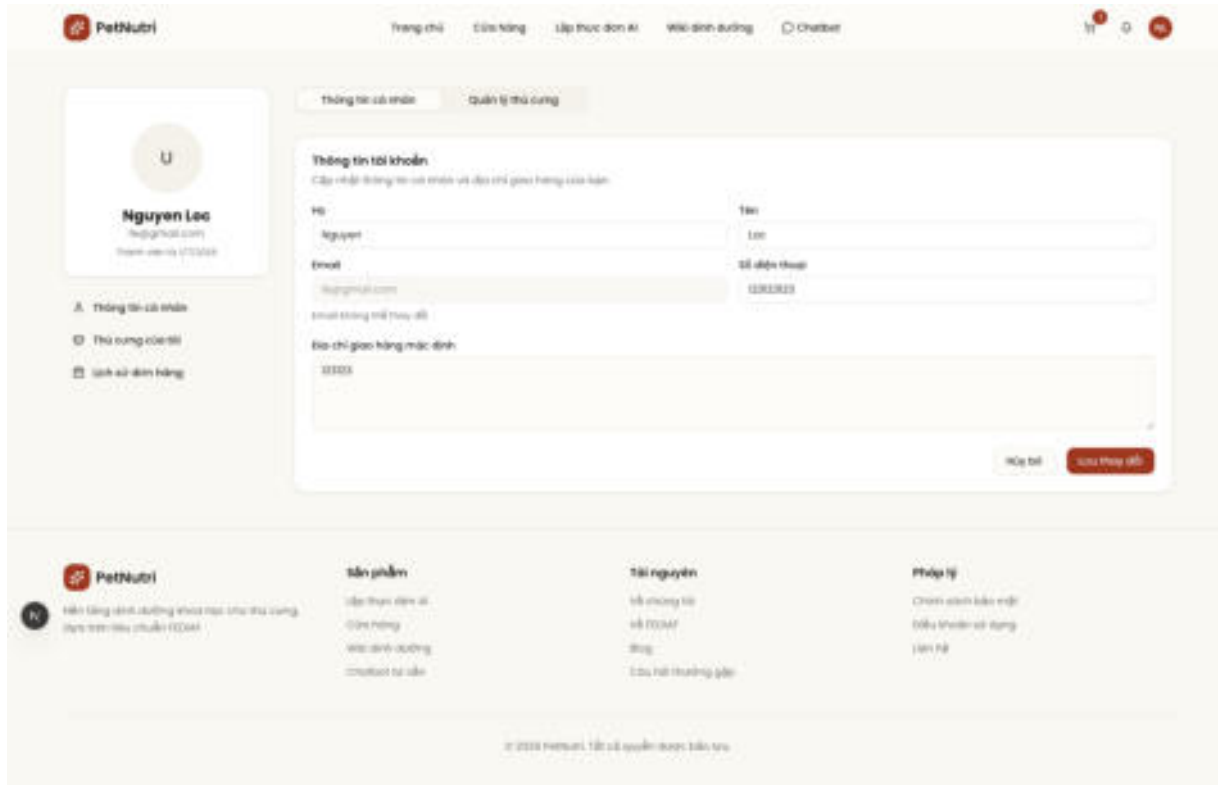
Hình 3.13 Màn hình wiki dinh dưỡng

e. Màn hình chatbot



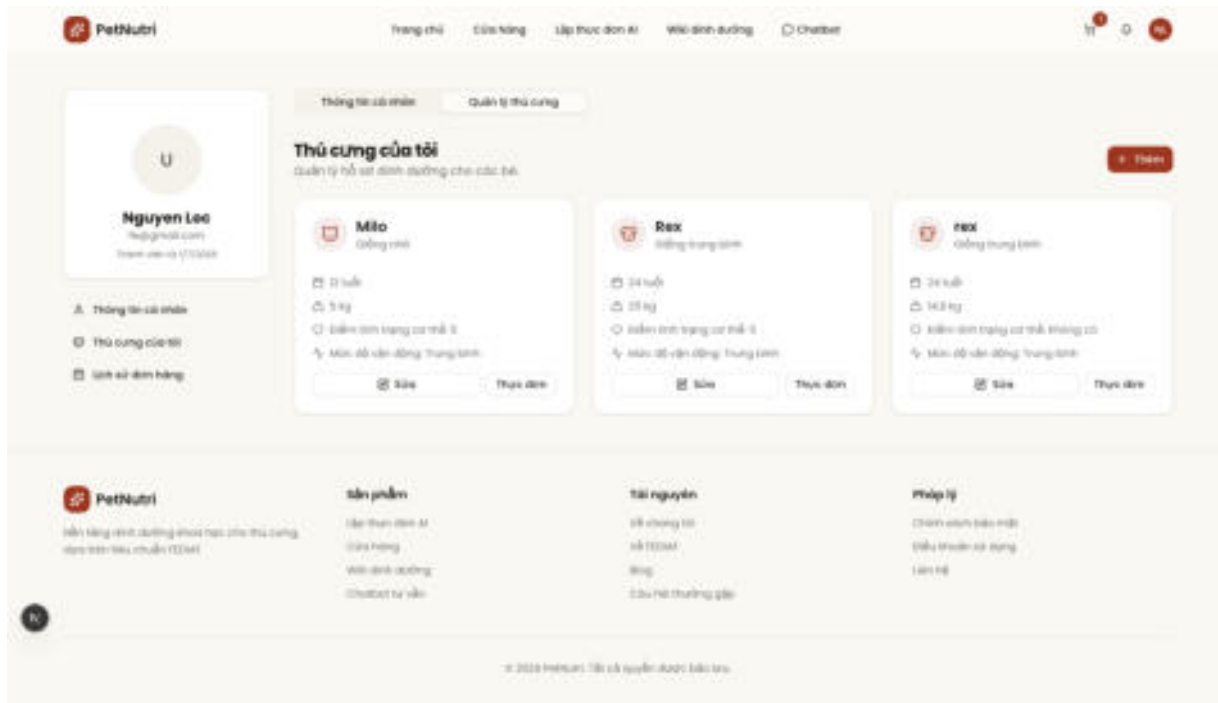
Hình 3.14 Màn hình chatbot

f. Màn hình profile người dùng



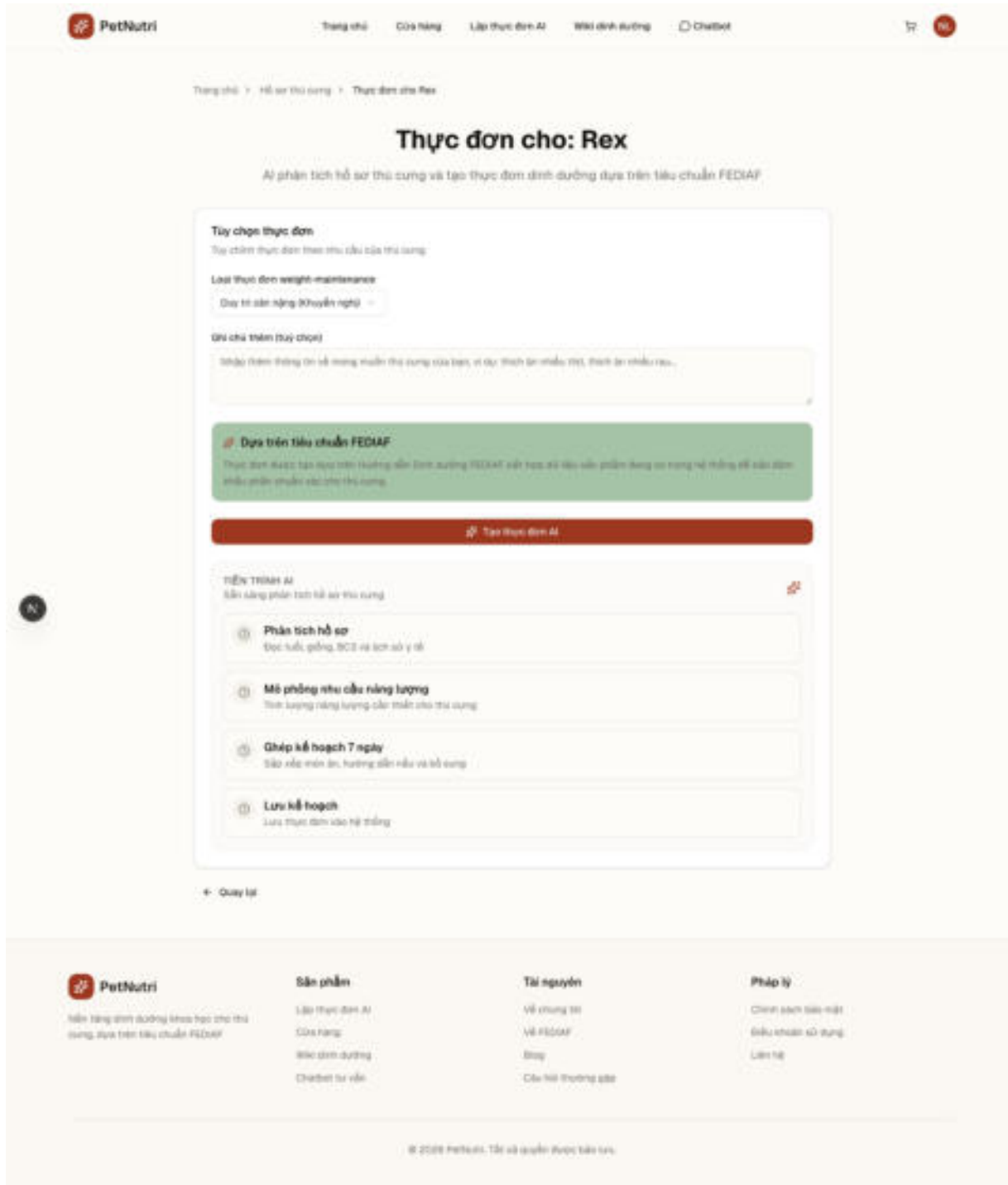
Hình 3.15 Màn hình profile người dùng

g. Màn hình quản lý thú cưng



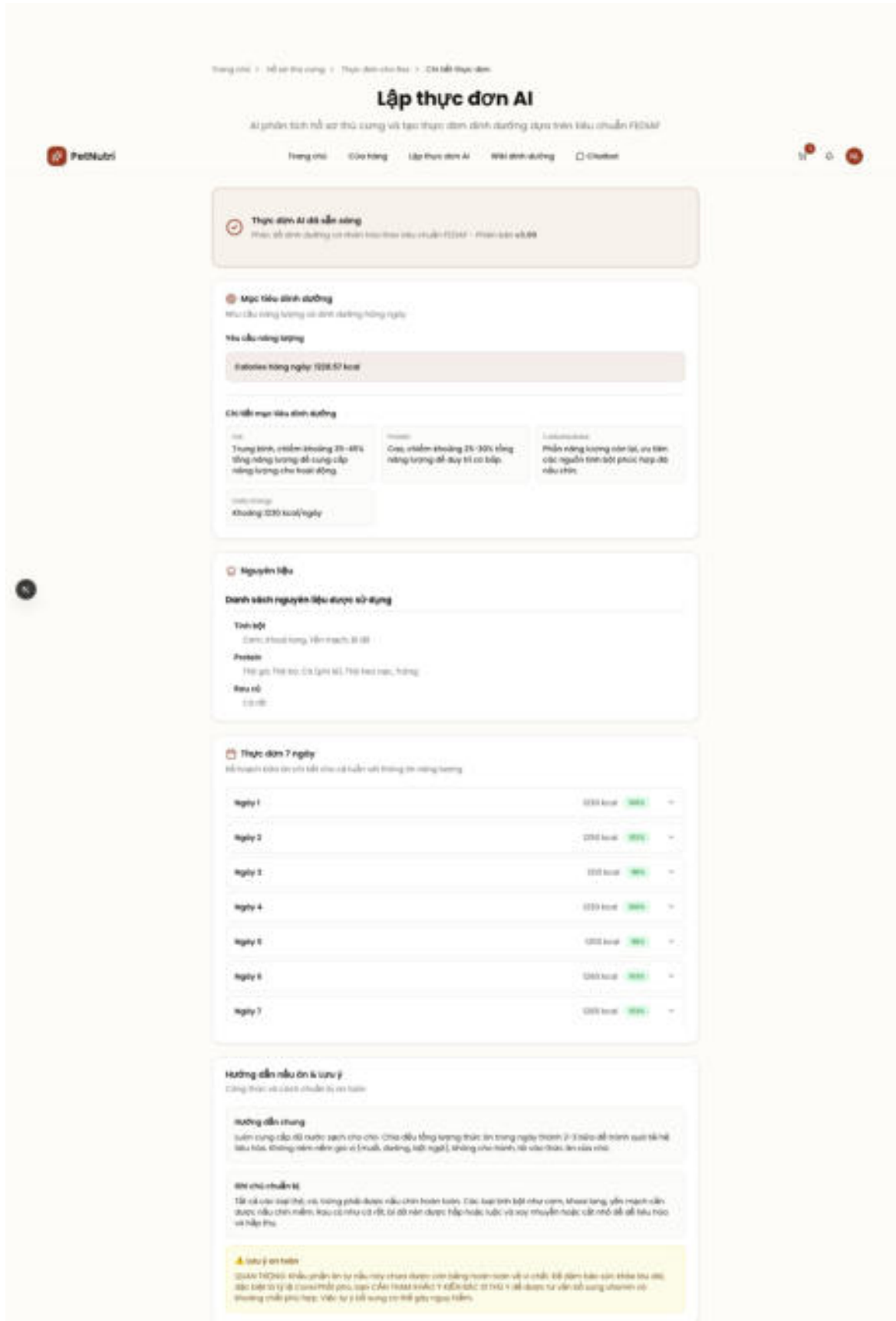
Hình 3.16 Màn hình quản lý thú cưng của người dùng

h. Màn hình tạo mới thực đơn



Hình 3.17 Màn hình tạo mới thực đơn

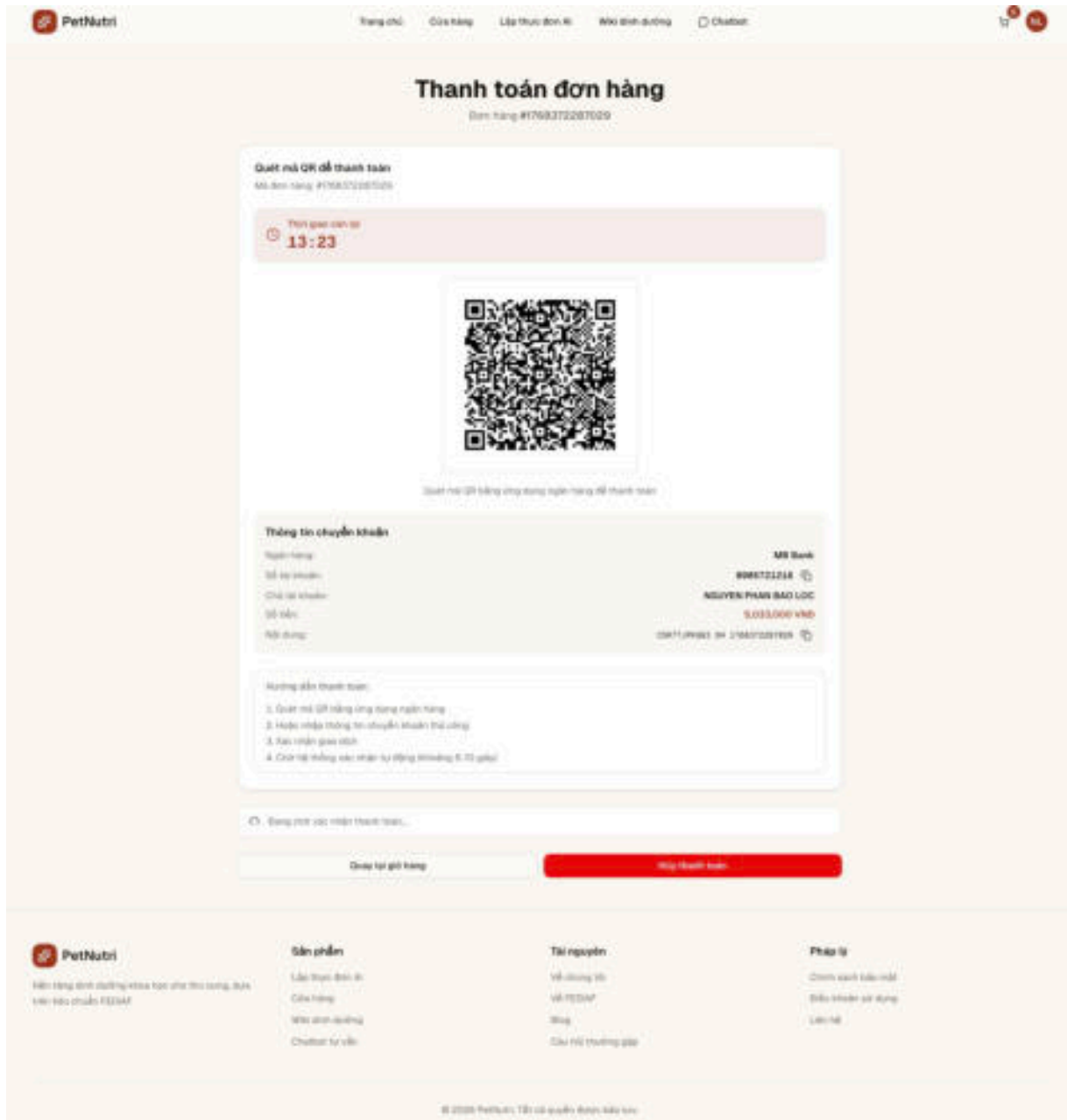
i. Màn hình chi tiết thực đơn và đánh giá



Hình 3.18 Màn hình chi tiết thực đơn và đánh giá

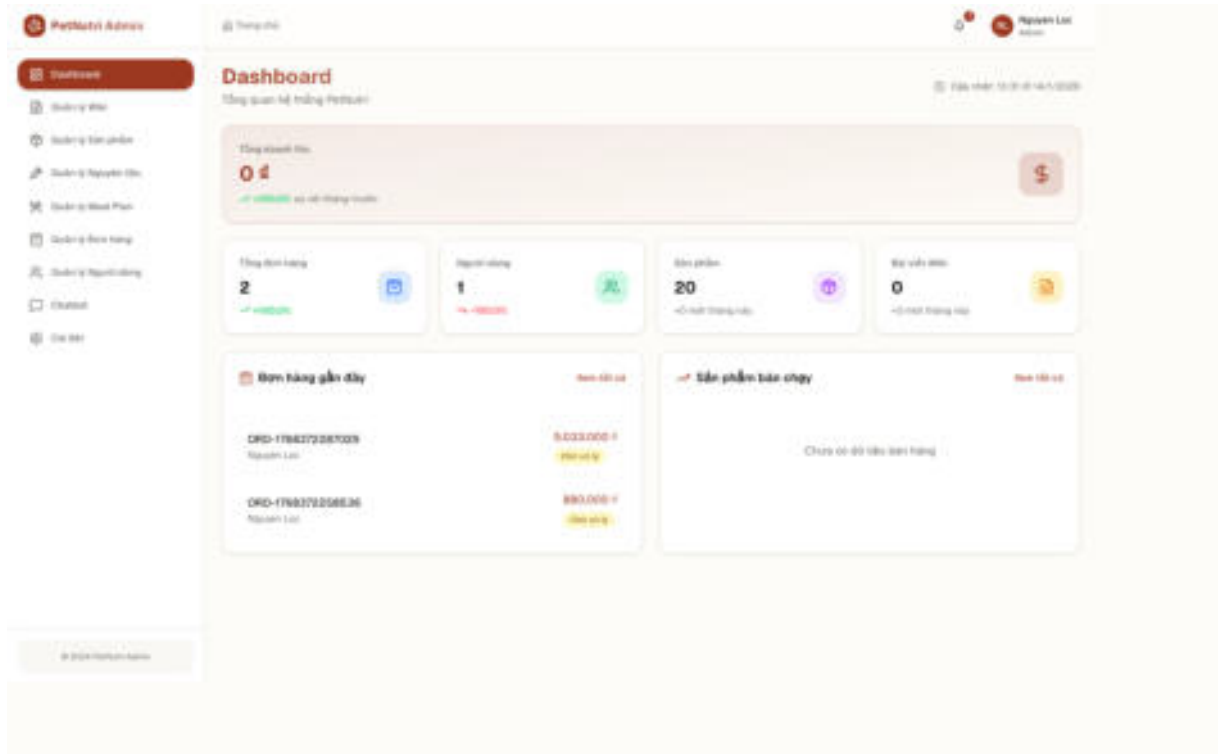


j. Màn hình mua hàng và thanh toán



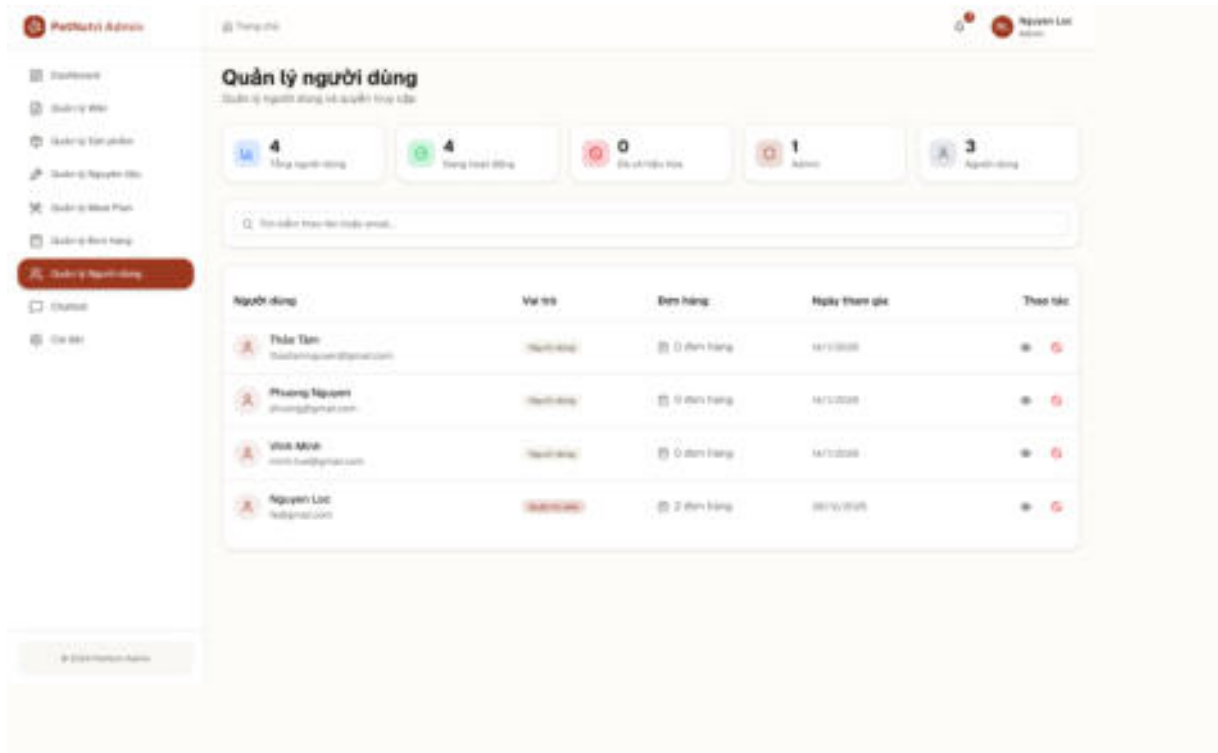
Hình 3.20 Màn hình mua hàng và thanh toán

k. Màn hình dashboard của admin



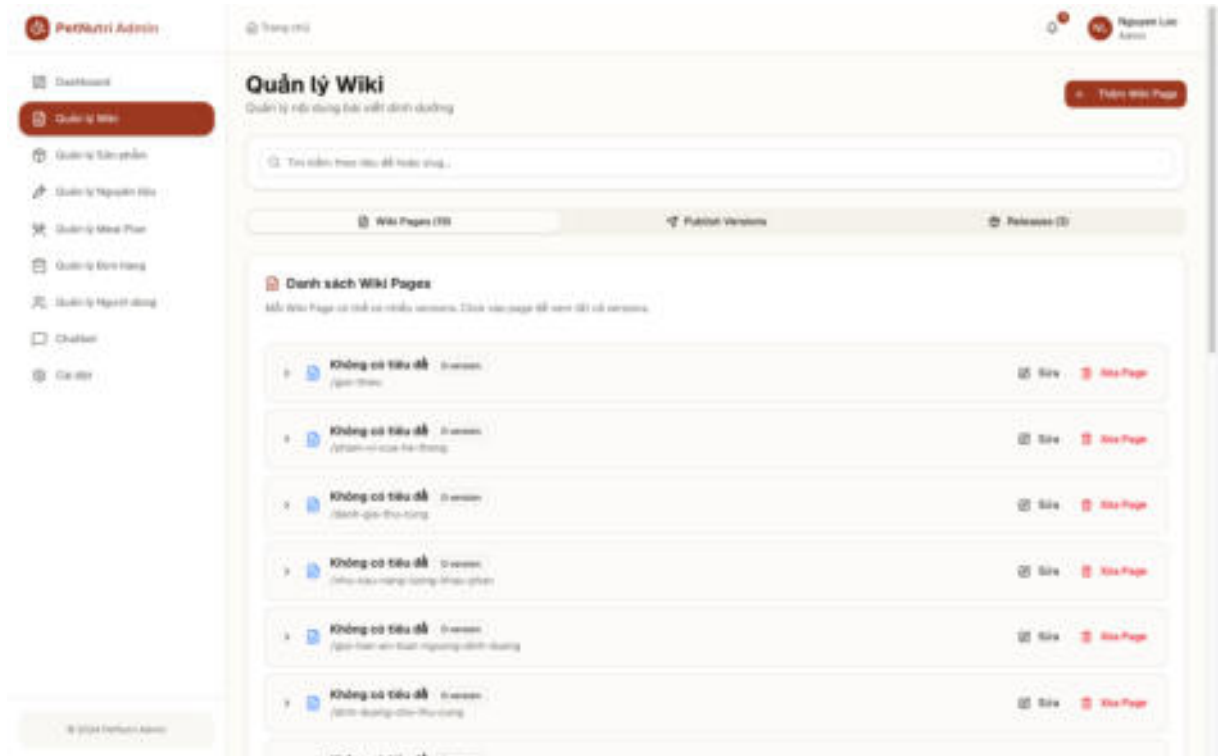
Hình 3.21 Màn hình dashboard của admin

l. Màn hình quản lí người dùng



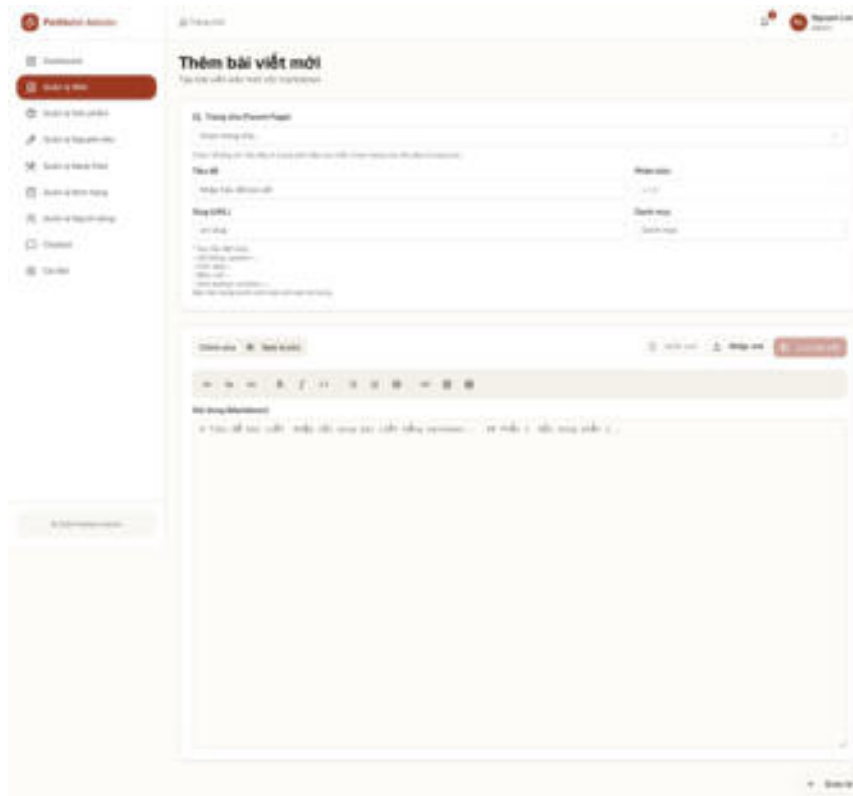
Hình 3.22 Màn hình quản lí người dùng

m. Màn hình quản lý Wiki dinh dưỡng



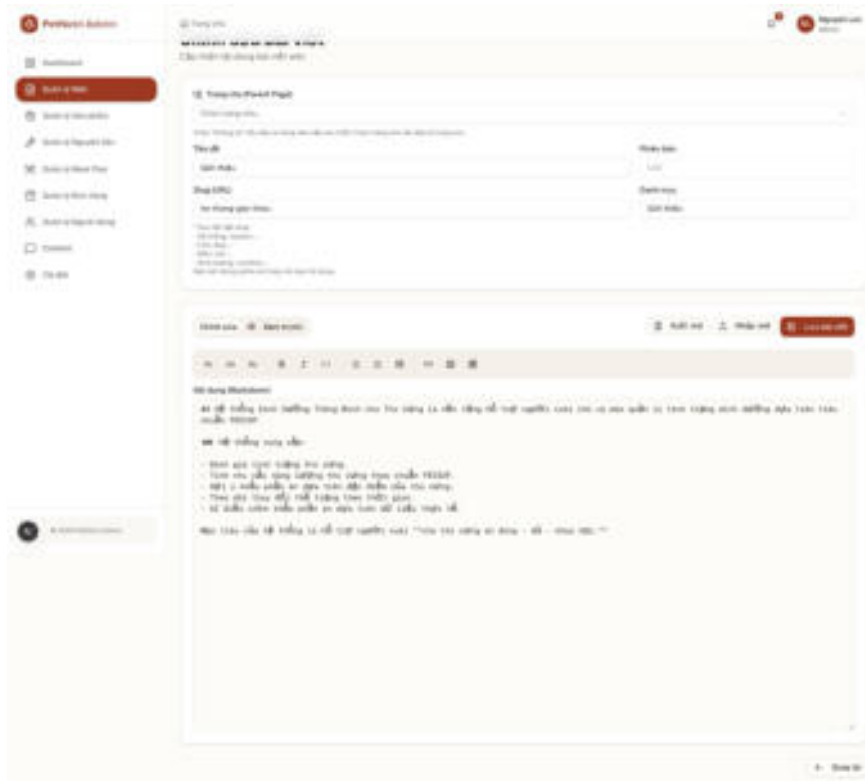
Hình 3.23 Màn hình quản lý wiki dinh dưỡng

n. Màn hình tạo mới Wiki dinh dưỡng



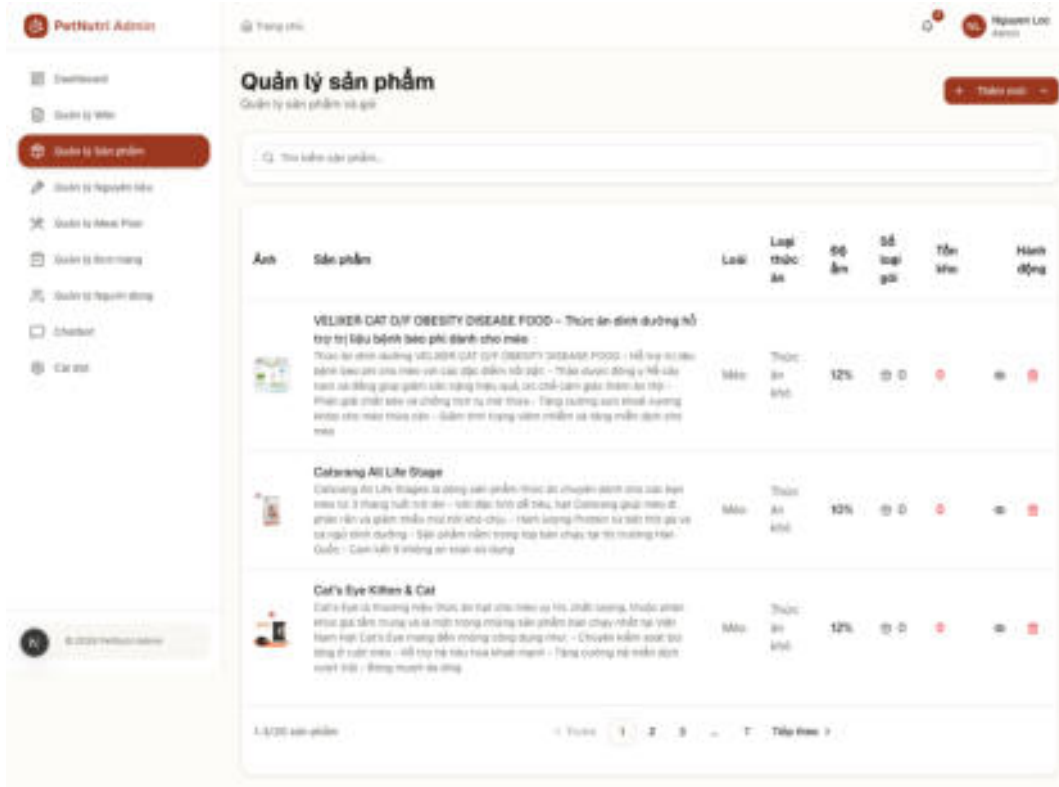
Hình 3.24 Màn hình thêm mới wiki

o. Màn hình xem chi tiết và cập nhật Wiki



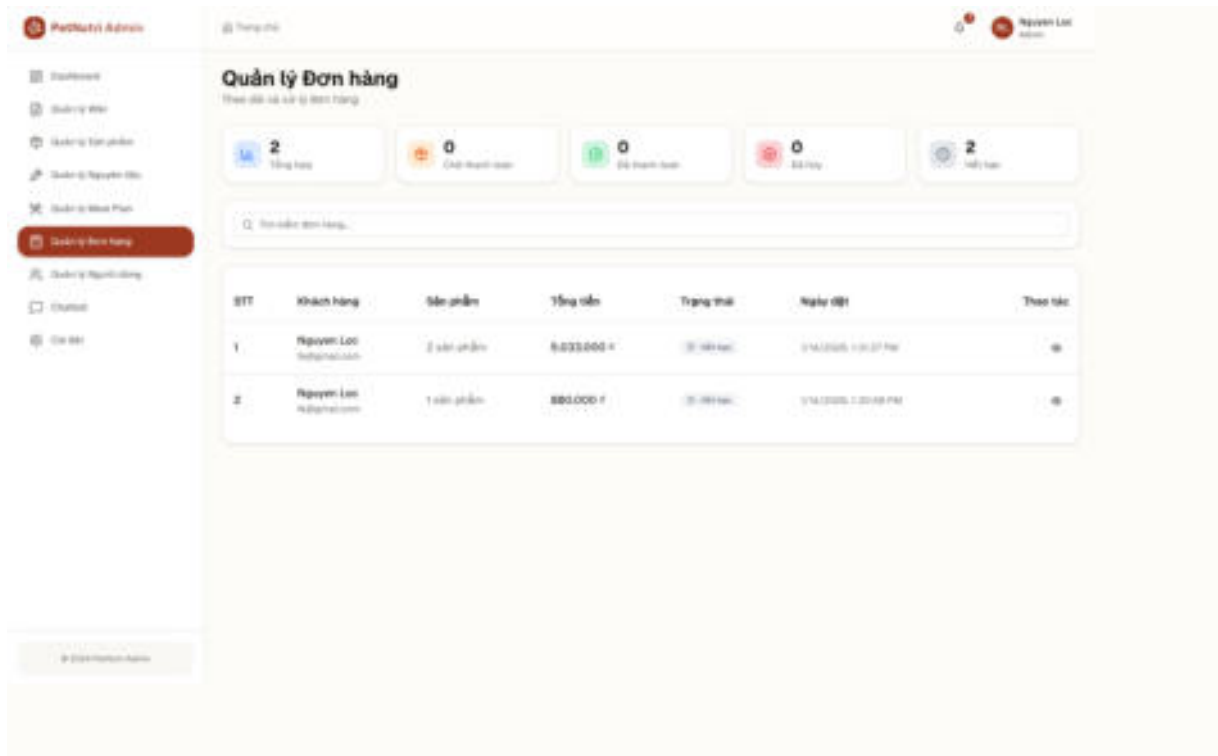
Hình 3.25 Màn hình xem chi tiết và cập nhật wiki

p. Màn hình quản lý sản phẩm



Hình 3.26 Màn hình quản lý sản phẩm

q. Màn hình quản lý đơn hàng



Hình 3.27 Màn hình quản lý đơn hàng

3.5. Kiểm thử phần mềm

Kiểm thử chức năng đăng nhập

Chức năng			Đăng nhập			
Số lượng testcase			6			
Kết quả kiểm thử			Passed	Failed	Pending	
			6	0	0	
ID	Điều kiện	Tiến Trình	Kết quả mong muốn	Kết quả thực tế	Kết quả	Ngày
1	Truy cập vào website	- Không nhập email và password - Nhấn đăng nhập	- Yêu cầu nhập đầy đủ thông tin	- Yêu cầu nhập đầy đủ thông tin	PASSED	15/01/2025
2	Truy cập vào website	- Nhập email - Không nhập password - Nhấn đăng nhập	- Yêu cầu nhập password	- Yêu cầu nhập password	PASSED	15/01/2025

3	Truy cập vào website	- Không nhập email - Nhập password - Nhấn đăng nhập	- Yêu cầu nhập email	- Yêu cầu nhập email	PASSED	15/01/2025
4	Truy cập vào website	- Nhập sai email - Nhập password - Nhấn đăng nhập	- Thông báo “Người dùng không tồn tại”	- Thông báo “Người dùng không tồn tại”	PASSED	15/01/2025
5	Truy cập vào website	- Nhập đúng email - Nhập sai password - Nhấn đăng nhập	- Thông báo “Thông tin đăng nhập không hợp lệ”	- Thông báo “Thông tin đăng nhập không hợp lệ”	PASSED	15/01/2025
6	Truy cập vào website	- Nhập đúng email và password - Nhấn đăng nhập	- Thông báo “Đăng nhập thành công” - Chuyển hướng sang trang chủ	- Thông báo “Đăng nhập thành công” - Chuyển hướng sang trang chủ	PASSED	15/01/2025

Kiểm thử chức năng tạo mới wiki

Chức năng		Tạo mới wiki				
Số lượng testcase		3				
Kết quả kiểm thử		Passed		Failed	Pending	
		3		0	0	
ID	Điều kiện	Tiến Trình	Kết quả mong muốn	Kết quả thực tế	Kết quả	Ngày
1	- Truy cập vào quản lí wiki - Là Admin	- Không nhập tiêu đề hoặc không nhập slug hoặc không nhập danh mục hoặc không nhập nội dung	- Yêu cầu nhập đầy đủ thông tin	- Yêu cầu nhập đầy đủ thông tin	PASSED	15/01/2025

		- Nhấn tạo wiki				
2	- Truy cập vào quản lí wiki - Là Admin	- Nhập slug đã tồn tại - Nhấn tạo wiki	Thông báo “Lỗi khi lưu bài viết”	Thông báo “Lỗi khi lưu bài viết”	PASSED	15/01/2025
3	- Truy cập vào quản lí wiki - Là Admin	- Nhập slug chưa tồn tại - Nhập tiêu đề và nhập danh mục và nhập nội dung	- Thông báo “Lưu bài viết thành công” - Lưu vào database	- Thông báo “Lưu bài viết thành công” - Lưu vào database	PASSED	15/01/2025

Kiểm thử chức năng Tạo hồ sơ thú cưng

Chức năng		Tạo mới hồ sơ thú cưng				
Số lượng testcase		5				
Kết quả kiểm thử		Passed		Failed	Pending	
		5		0	0	
ID	Điều kiện	Tiến Trình	Kết quả mong muốn	Kết quả thực tế	Kết quả	Ngày
1	Truy cập trang tạo hồ sơ thú cưng	- Không nhập tên thú cưng - Không nhập cân nặng - Nhấn “Lưu hồ sơ”	- Thông báo “Vui lòng điền đầy đủ thông tin bắt buộc”	- Thông báo “Vui lòng điền đầy đủ thông tin bắt buộc”	PASSED	15/01/2025
2	Truy cập trang tạo hồ sơ thú cưng	- Nhập tên thú cưng - Không nhập cân nặng - Để trống các trường bắt buộc khác - Nhấn”Lưu hồ sơ”	- Thông báo “Vui lòng điền đầy đủ thông tin bắt buộc”	- Thông báo “Vui lòng điền đầy đủ thông tin bắt buộc”	PASSED	15/01/2025
3	Truy cập trang tạo hồ sơ thú cưng	- Nhập tên thú cưng - Chọn loài - Chọn giống	- Thông báo “Vui lòng điền đầy đủ	- Thông báo “Vui lòng điền đầy đủ	PASSED	15/01/2025

		- Không nhập cân nặng - Nhấn "Lưu hồ sơ"	thông tin bắt buộc"	thông tin bắt buộc"		
4	Truy cập trang tạo hồ sơ thú cưng	- Nhập đầy đủ thông tin nhưng không chọn điểm BCS - Nhấn "Lưu hồ sơ"	- Thông báo "Vui lòng điền đầy đủ thông tin bắt buộc"	- Thông báo "Vui lòng điền đầy đủ thông tin bắt buộc"	PASSED	15/01/2025
5	Truy cập trang tạo hồ sơ thú cưng	- Nhập đầy đủ thông tin hợp lệ (tên, loài, giống, cân nặng, tuổi, bcs, mức hoạt động) - Nhấn "Lưu hồ sơ"	- Thông báo "Tạo hồ sơ thành công" - Chuyển đến trang danh sách hồ sơ	- Thông báo "Tạo hồ sơ thành công" - Chuyển đến trang danh sách hồ sơ	PASSED	15/01/2025

## Kết luận

### Kết quả đạt được

Trong phạm vi đề tài, trong suốt quá trình nghiên cứu lý thuyết và triển khai ứng dụng thực tiễn, đề án đã đạt được một số kết quả quan trọng sau:

Về mặt kỹ thuật

- Vận dụng mô hình client-server để thiết kế và triển khai hệ thống web hoàn chỉnh với 3 thành phần: Frontend (Next.js), Backend (NestJS), và AI Service (FastAPI).
- Xây dựng backend theo kiến trúc RESTful API, đảm bảo khả năng mở rộng và tách biệt giữa các tầng hệ thống.
- Tích hợp cơ sở dữ liệu PostgreSQL với TypeORM, thiết kế schema phù hợp cho việc quản lý thú cưng, thực đơn và nội dung wiki.
- Triển khai giao tiếp thời gian thực (WebSocket) cho chức năng chatbot, cho phép tương tác liền mạch giữa người dùng và AI.
- Ứng dụng kỹ thuật RAG (Retrieval-Augmented Generation) và Function Calling của Google Gemini API để xây dựng AI Agent thông minh, có khả năng truy xuất kiến thức và thực thi hành động.
- Thiết lập quy trình CI/CD tự động với GitHub Actions, triển khai production trên DigitalOcean sử dụng Docker.

Về mặt ứng dụng thực tiễn

- Phát triển thành công hệ thống web cho phép người dùng quản lý hồ sơ thú cưng, tra cứu kiến thức dinh dưỡng và sử dụng chatbot tư vấn.
- Xây dựng chức năng tự động tạo thực đơn dinh dưỡng cá nhân hóa dựa trên đặc điểm của từng thú cưng (tuổi, cân nặng, tình trạng sức khỏe, mức độ hoạt động).
- Tích hợp AI tư vấn dinh dưỡng dựa trên tiêu chuẩn FEDIAF, nâng cao tính chính xác và giá trị tham khảo của nội dung.
- Xây dựng hệ thống quản lý wiki kiến thức dinh dưỡng với tính năng versioning và publishing.
- Phát triển giao diện người dùng trực quan, responsive, hỗ trợ tiếng Việt, phù hợp với người dùng phổ thông.

### Hạn chế

Bên cạnh những kết quả đạt được, hệ thống vẫn còn tồn tại một số hạn chế:

- Chức năng tư vấn và lập thực đơn chưa thể thay thế hoàn toàn vai trò của chuyên gia thú y, các kết quả chỉ mang tính chất tham khảo.
- Dữ liệu dinh dưỡng và kiến thức chuyên sâu còn hạn chế về số lượng và phạm vi.
- Hệ thống chưa được kiểm thử trên quy mô người dùng lớn, hiệu năng và khả năng mở rộng chưa được đánh giá đầy đủ. Chỉ dừng lại ở mức thử nghiệm.

### **Hướng phát triển**

Hệ thống có thể được mở rộng và hoàn thiện theo các hướng sau:

- Bổ sung và cập nhật dữ liệu dinh dưỡng, mở rộng phạm vi cho nhiều giống loài và độ tuổi khác nhau.
- Nâng cao độ chính xác của chatbot AI bằng cách tự huấn luyện, tự cải tiến mô hình.
- Tích hợp thêm các chức năng theo dõi sức khỏe thú cưng, lịch sử ăn uống và gợi ý điều chỉnh chế độ dinh dưỡng.
- Xây dựng có chế kiểm duyệt và xác thực nội dung nhằm nâng cao độ tin cậy của các dữ liệu dinh dưỡng.

## TÀI LIỆU THAM KHẢO

- [1] Cole Stryker, “What are large language models (LLMs)”. Có tại: <https://www.ibm.com/think/topics/large-language-models> [Truy cập 10/11/2025]
- [2] GeeksforGeeks, “What is Retrieval-Augmented Generation (RAG) ?”. Có tại: <https://www.geeksforgeeks.org/nlp/what-is-retrieval-augmented-generation-rag/> [Truy cập 10/11/2025]
- [3] FastAPI Documentation, “WebSockets – FastAPI”. Có tại: <https://fastapi.tiangolo.com/advanced/websockets/> [Truy cập 10/11/2025]
- [4] GeeksforGeeks, “How To Use WebSocket With FastAPI”. Có tại: <https://www.geeksforgeeks.org/python/how-to-use-websocket-with-fastapi/> [Truy cập 10/11/2025]
- [5] DigitalOcean Docs, “Spaces: Object Storage”. Có tại: <https://docs.digitalocean.com/products/spaces/> [Truy cập 10/11/2025]
- [6] Red Hat, “CI/CD Pipeline Explained”. Có tại: <https://www.redhat.com/en/topics/devops/what-is-ci-cd> [Truy cập 10/11/2025]
- [7] FEDIAF, “FEDIAF EuropeanPetFood”. Có tại: <https://europeanpetfood.org/> [Truy cập 10/11/2025]