

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

[illegible]

Vĩnh Long, ngày ..... tháng ..... năm .....  
**Giảng viên hướng dẫn**  
 (Ký tên và ghi rõ họ tên)

## NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

[illegible]

Vĩnh Long, ngày ..... tháng ..... năm .....  
**Thành viên hội đồng**  
 (Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Để hoàn thành tốt bài đề tài này, trước hết em xin cảm ơn đến quý thầy cô giáo trong Trường Kỹ Thuật và Công Nghệ của Đại Học Trà Vinh. Em xin cảm ơn đến ban lãnh đạo nhà trường đã tạo mọi điều kiện tốt nhất để em có cơ hội được trình bày và thể hiện ý tưởng của mình, từ đó áp dụng kiến thức, kỹ năng của mình để đưa ý tưởng ấy thành sản phẩm thực tế có thể áp dụng trong đời sống. Đặc biệt, em xin gửi lời cảm ơn chân thành đến thầy Khấu Văn Nhựt - Giảng viên hướng dẫn, thầy đã tận tình giúp đỡ em trong quá trình học tập cũng như trong việc nâng cao kiến thức chuyên môn, phát huy tính tự học, tự tìm hiểu, từ đó hình thành một hướng đi rõ ràng giúp em dễ dàng tiếp cận và ngày càng tiến sâu với chuyên ngành Công Nghệ Thông Tin một cách nhanh chóng hơn. Đồng thời trau dồi cho em kỹ năng học tập năng động và sáng tạo, rèn luyện tư duy cũng như khả năng lập trình. Giúp cho em trau dồi được nhiều bài học mới, được tìm hiểu nhiều ngôn ngữ lập trình.

Do khả năng cũng như kiến thức chuyên môn của bản thân em còn hạn chế, nên trong quá trình làm bài báo cáo không tránh khỏi những sai sót, em rất mong nhận được những ý kiến đóng góp của quý thầy, cô để cho em có thể khắc phục những thiếu sót và có thể hoàn thiện bài báo cáo hơn trong tương lai.

Em xin chân thành cảm ơn!

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN .....	3
1.1 Đặt vấn đề .....	3
1.2 Mục tiêu nghiên cứu và xây dựng hệ thống .....	3
1.3 Phương pháp tiếp cận và công nghệ sử dụng .....	3
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT .....	5
2.1 Tổng quan về công nghệ sử dụng .....	5
2.1.1 Giới thiệu về ReactJS .....	5
2.1.2 Giới thiệu về Tailwind CSS .....	8
2.1.3 Giới thiệu về NodeJS .....	11
2.1.4 Giới thiệu về MongoDB .....	14
2.2 Giới thiệu nghiệp vụ của đề tài .....	16
2.2.1 Tổng quan nghiệp vụ .....	16
2.2.2 Nghiệp vụ dành cho khách hàng .....	16
2.2.3 Nghiệp vụ dành cho quản trị viên .....	17
2.2.4 Nghiệp vụ xử lý đặt hàng và tiếp nhận đơn hàng .....	17
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....	18
3.1. Mô tả bài toán .....	18
3.2 Yêu cầu hệ thống .....	18
3.2.1 Yêu cầu về chức năng .....	18
3.2.2 Yêu cầu phi chức năng .....	19
3.3 Thiết kế cơ sở dữ liệu .....	20
3.3.1 Mô hình dữ liệu của hệ thống .....	20
3.3.2 Danh sách các bảng dữ liệu .....	21
3.4 Mô hình xử lý .....	23
3.4.1 Sơ đồ Use case tổng quát .....	23
3.4.2 Sơ đồ Use case đặt hàng .....	25
3.4.3 Sơ đồ Use case quản lý đơn hàng .....	25
3.4.4 Sơ đồ Use case quản lý sản phẩm .....	26
3.5 Thiết kế API .....	26
3.5.1 API quản lý người dùng .....	26
3.5.2 API danh mục sản phẩm .....	27

3.5.3 API sản phẩm .....	27
3.5.4 API giỏ hàng .....	27
3.5.5 API đặt hàng.....	28
3.5.6 API đánh giá sản phẩm .....	28
3.5.7 API hiển thị nội dung .....	29
<b>CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU .....</b>	<b>30</b>
4.1 Giao diện người dùng.....	30
4.1.1 Giao diện Trang chủ.....	30
4.1.2 Giao diện trang Sản phẩm .....	31
4.1.3 Giao diện trang Chi tiết sản phẩm.....	32
4.1.4 Giao diện trang Giới thiệu.....	33
4.1.5 Giao diện trang Liên hệ.....	33
4.1.6 Giao diện trang Giỏ hàng .....	34
4.1.7 Giao diện trang Thanh toán.....	34
4.1.8 Giao diện trang Chi tiết đơn hàng .....	35
4.1.9 Giao diện trang Quản lý hồ sơ cá nhân .....	35
4.1.10 Giao diện trang Đăng ký .....	36
4.1.11 Giao diện trang Đăng nhập .....	36
4.2 Giao diện Quản trị viên.....	37
4.2.1 Giao diện trang Thống kê.....	37
4.2.2 Giao diện trang Quản lý đơn hàng .....	37
4.2.3 Giao diện trang Quản lý sản phẩm.....	38
4.2.4 Giao diện trang Quản lý Flash Sale .....	38
4.2.5 Giao diện trang Quản lý danh mục .....	39
4.2.6 Giao diện trang Quản lý khách hàng.....	39
4.2.7 Giao diện trang Quản lý đánh giá, bình luận .....	40
4.2.8 Giao diện trang Quản lý Banner .....	40
<b>CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>42</b>
5.1 Kết luận.....	42
5.1.1 Kết quả đạt được .....	42
5.1.2 Hạn chế.....	42
5.2 Hướng phát triển .....	42
<b>DANH MỤC TÀI LIỆU THAM KHẢO .....</b>	<b>43</b>

## DANH MỤC HÌNH ẢNH

Hình 2.1 Khái niệm ReactJS [1].....	5
Hình 2.2 Khái niệm Tailwind CSS [3] .....	8
Hình 2.3 Khái niệm NodeJS [4] .....	11
Hình 2.4 Luồng hoạt động của NodeJS [2] .....	12
Hình 2.5 Khái niệm MongoDB [5].....	14
Hình 3.1 Mô hình dữ liệu của hệ thống.....	20
Hình 3.2 Use case tổng quát .....	24
Hình 3.3 Use case đặt hàng .....	25
Hình 3.4 Use case quản lý đơn hàng .....	25
Hình 3.5 Use case quản lý sản phẩm.....	26
Hình 4.1 Phần trên của giao diện trang chủ.....	30
Hình 4.2 Phần giữa của giao diện trang chủ.....	30
Hình 4.3 Phần cuối của giao diện trang chủ.....	31
Hình 4.4 Giao diện trang sản phẩm .....	32
Hình 4.5 Giao diện trang chi tiết sản phẩm .....	32
Hình 4.6 Giao diện đánh giá, bình luận.....	33
Hình 4.7 Giao diện trang giới thiệu .....	33
Hình 4.8 Giao diện trang liên hệ .....	34
Hình 4.9 Giao diện giỏ hàng.....	34
Hình 4.10 Giao diện trang thanh toán .....	35
Hình 4.11 Giao diện trang chi tiết đơn hàng .....	35
Hình 4.12 Giao diện trang quản lý hồ sơ cá nhân .....	36
Hình 4.13 Giao diện trang đăng ký .....	36
Hình 4.14 Giao diện trang đăng nhập.....	37
Hình 4.15 Giao diện trang thống kê .....	37
Hình 4.16 Giao diện trang quản lý đơn hàng .....	38
Hình 4.17 Giao diện trang quản lý sản phẩm .....	38
Hình 4.18 Giao diện trang quản lý Flash Sale.....	39
Hình 4.19 Giao diện trang quản lý danh mục.....	39
Hình 4.20 Giao diện trang quản lý khách hàng .....	40
Hình 4.21 Giao diện trang Quản lý đánh giá, bình luận.....	40
Hình 4.22 Giao diện trang quản lý banner .....	41

## **DANH MỤC BẢNG BIỂU**

Bảng 3.1 Bảng User.....	21
Bảng 3.2 Bảng Category.....	21
Bảng 3.3 Bảng Product.....	21
Bảng 3.4 Bảng Cart .....	22
Bảng 3.5 Bảng CartItem.....	22
Bảng 3.6 Bảng Order.....	22
Bảng 3.7 Bảng OrderItem.....	22
Bảng 3.8 Bảng Review .....	23
Bảng 3.9 Bảng Banner.....	23
Bảng 3.10 Bảng Contact.....	23
Bảng 3.11 API quản lý người dùng.....	26
Bảng 3.12 API danh mục sản phẩm .....	27
Bảng 3.13 API sản phẩm .....	27
Bảng 3.14 API giỏ hàng .....	28
Bảng 3.15 API đặt hàng.....	28
Bảng 3.16 API đánh giá sản phẩm .....	28
Bảng 3.17 API hiển thị nội dung .....	29

## TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

### Vấn đề nghiên cứu

Đối với đề tài này, em tập trung vào việc xây dựng một website bán bánh kem trực tuyến, giúp cửa hàng có thể giới thiệu sản phẩm, quản lý đơn hàng và hỗ trợ khách hàng đặt bánh một cách nhanh chóng, thuận tiện. Trong thời đại thương mại điện tử phát triển mạnh, việc số hóa hoạt động bán hàng là rất cần thiết, vì vậy em chọn đề tài này để vừa rèn luyện kỹ năng lập trình web, vừa tạo ra một sản phẩm có khả năng ứng dụng thực tế.

### Hướng tiếp cận

Ở phần phát triển giao diện, em sử dụng ReactJS kết hợp với Tailwind CSS để xây dựng các trang web thân thiện, hiện đại. ReactJS giúp em chia giao diện thành các component rõ ràng, dễ tái sử dụng, còn Tailwind CSS hỗ trợ thiết kế giao diện hiện đại, bố cục rõ ràng. Các API được kiểm thử bằng Postman để đảm bảo dữ liệu phản hồi chính xác và hoạt động ổn định trước khi kết nối với giao diện.

Trong quá trình thực hiện, em tiếp cận theo hướng làm từ những chức năng cốt lõi trước, sau đó mở rộng dần. Cụ thể là em sẽ thiết kế giao diện trang chủ, trang sản phẩm, chi tiết sản phẩm, giỏ hàng và trang quản trị. Song song với đó, em tìm hiểu thêm về cách tổ chức dự án React theo cấu trúc hợp lý.

### Cách giải quyết vấn đề

- Phân tích yêu cầu dựa trên hoạt động thực tế của một cửa hàng bánh kem.
- Thiết kế giao diện theo hướng đơn giản, trực quan để người dùng có thể xem hình ảnh, giá và mô tả sản phẩm một cách thuận tiện.
- Xây dựng backend xử lý dữ liệu sản phẩm từ máy chủ. Sau khi hoàn thiện, hệ thống được kiểm tra và điều chỉnh nhằm đảm bảo website hiển thị đúng.
- Các API hoạt động mượt mà và giao diện tương thích trên nhiều kích thước màn hình.

### Kết quả đạt được

Kết quả đạt được là hệ thống giao diện website bán bánh kem có giao diện trực quan, hiển thị danh sách sản phẩm, xem chi tiết sản phẩm, thêm sản phẩm vào giỏ hàng và mô phỏng quy trình đặt hàng cơ bản. Thông qua quá trình thực hiện đồ án, em đã hiểu rõ hơn về quy trình xây dựng một website thực tế, từ khâu phân tích yêu cầu, thiết kế giao diện đến triển khai chức năng. Việc sử dụng ReactJS kết hợp với



Tailwind CSS giúp em nắm vững hơn cách xây dựng giao diện theo hướng component, tối ưu hóa việc tái sử dụng mã nguồn và tạo giao diện responsive.

Bên cạnh đó, đồ án còn giúp em rèn luyện kỹ năng lập trình frontend, kỹ năng phân tích và giải quyết vấn đề phát sinh trong quá trình phát triển hệ thống. Mặc dù hệ thống vẫn còn một số hạn chế và có thể tiếp tục cải tiến thêm trong tương lai, nhưng kết quả đạt được cho thấy đồ án đã hoàn thành đúng mục tiêu đề ra.

## MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong thời đại công nghệ số phát triển mạnh mẽ như hiện nay, thói quen mua sắm của người tiêu dùng đã thay đổi đáng kể. Việc đặt mua các sản phẩm trực tuyến trở nên phổ biến nhờ sự tiện lợi, tiết kiệm thời gian và khả năng so sánh giá cả, lựa chọn sản phẩm dễ dàng. Xuất phát từ thực tế này, em quyết định chọn đề tài “Xây dựng hệ thống website bán bánh kem trực tuyến sử dụng ReactJS và Tailwind CSS” làm đồ án chuyên ngành. Em muốn xây dựng một trang web có giao diện đẹp, dễ sử dụng, cung cấp các chức năng cơ bản và có tính ứng dụng cao. Bên cạnh đó, ReactJS và Tailwind CSS là hai công nghệ hiện đại, được sử dụng rộng rãi trong phát triển giao diện web. Cùng với đó, Node.js – Express là bộ công cụ mạnh mẽ và linh hoạt để xây dựng backend API. Việc kết hợp cả frontend và backend giúp em hiểu rõ quy trình xây dựng một website đầy đủ.

### 2. Mục đích nghiên cứu

- Xây dựng một hệ thống website có khả năng giới thiệu sản phẩm bánh kem một cách trực quan và sinh động, giúp người dùng dễ dàng xem thông tin, lựa chọn mẫu bánh và tìm hiểu chi tiết sản phẩm.
- Rèn luyện kỹ năng thiết kế giao diện bằng ReactJS và Tailwind CSS, cũng như làm chủ quy trình kết nối frontend với backend thông qua các API được xây dựng bằng Node.js và Express.
- Giúp củng cố kỹ năng phân tích yêu cầu và triển khai một hệ thống web hoàn chỉnh ở mức cơ bản.

### 3. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm quy trình xây dựng giao diện website thương mại điện tử, cách thiết kế và quản lý dữ liệu sản phẩm, cũng như phương thức giao tiếp giữa client và server. Đề tài tập trung vào những trải nghiệm cơ bản của người dùng như xem danh sách sản phẩm, xem chi tiết sản phẩm và cách dữ liệu được xử lý phía backend.

### 4. Phạm vi nghiên cứu

Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng một hệ thống website bán bánh kem trực tuyến với các chức năng cơ bản mà người dùng thường cần khi mua sắm online. Cụ thể, website cho phép người dùng xem danh sách sản phẩm,

xem thông tin chi tiết của từng loại sản phẩm, thêm sản phẩm vào giỏ hàng và theo dõi các mặt hàng đã chọn. Đây là những chức năng trọng tâm giúp mô phỏng trải nghiệm mua hàng trực tuyến một cách rõ ràng và gần gũi. Đề tài không mở rộng sang tính năng phức tạp hơn như là quản lý giao hàng. Việc giới hạn phạm vi như vậy giúp tập trung vào việc hoàn thiện các chức năng chính là xem sản phẩm và đặt hàng, đảm bảo website hoạt động ổn định, dễ sử dụng.

## CHƯƠNG 1: TỔNG QUAN

### 1.1 Đặt vấn đề

Trong thời đại công nghệ phát triển mạnh mẽ, người dùng ngày càng quen thuộc với việc mua sắm trực tuyến vì sự tiện lợi và nhanh chóng mà nó mang lại. Đối với các cửa hàng bánh kem, nhu cầu giới thiệu sản phẩm một cách sinh động, rõ ràng và dễ tiếp cận là rất quan trọng. Tuy nhiên, nhiều tiệm bánh nhỏ hiện nay vẫn chưa có website riêng hoặc chỉ đăng hình ảnh trên mạng xã hội. Việc này khiến thông tin bị rời rạc, người mua khó xem giá, mô tả chi tiết hay so sánh các mẫu bánh với nhau. Điều đó tạo ra sự hạn chế không nhỏ trong việc tiếp cận khách hàng và xây dựng hình ảnh chuyên nghiệp cho cửa hàng.

Xuất phát từ thực tiễn đó, việc nghiên cứu và xây dựng một website bán bánh kem trực tuyến là cần thiết, nhằm tạo ra một nền tảng tập trung giúp cửa hàng giới thiệu sản phẩm rõ ràng, hỗ trợ khách hàng đặt mua thuận tiện và góp phần nâng cao hiệu quả hoạt động kinh doanh trong môi trường số.

### 1.2 Mục tiêu nghiên cứu và xây dựng hệ thống

Mục tiêu chính của đề tài là xây dựng một website bán bánh kem trực tuyến ở mức cơ bản nhưng đầy đủ chức năng cần thiết, đóng vai trò như một cửa hàng trực tuyến thu nhỏ. Hệ thống được thiết kế nhằm hỗ trợ khách hàng dễ dàng tìm kiếm, xem thông tin chi tiết và đặt mua các loại bánh kem theo nhu cầu mà không cần đến trực tiếp cửa hàng.

Bên cạnh mục tiêu tạo ra một sản phẩm ứng dụng có khả năng sử dụng thực tế, đề tài còn hướng đến mục đích học thuật, giúp em vận dụng các kiến thức đã học về lập trình web mã nguồn mở vào giải quyết một bài toán cụ thể. Thông qua quá trình phân tích yêu cầu, thiết kế giao diện, xây dựng chức năng và xử lý dữ liệu, đề tài góp phần rèn luyện tư duy hệ thống, kỹ năng lập trình và khả năng tổ chức một website hoàn chỉnh.

### 1.3 Phương pháp tiếp cận và công nghệ sử dụng

Về phương pháp tiếp cận, tiến hành phân tích nhu cầu thực tế của người dùng khi mua bánh kem trực tuyến, từ đó xác định các chức năng cốt lõi cần có của hệ thống như xem sản phẩm, tìm kiếm, quản lý giỏ hàng và đặt hàng. Dựa trên các yêu cầu đó, xây dựng hệ thống theo mô hình ứng dụng web hiện đại, tách biệt giữa phía giao diện

người dùng (frontend) và phía xử lý dữ liệu (backend), giúp hệ thống dễ bảo trì và mở rộng trong tương lai.

Về công nghệ sử dụng, frontend của website được phát triển bằng ReactJS, cho phép xây dựng giao diện theo mô hình component, tăng khả năng tái sử dụng và mang lại trải nghiệm người dùng mượt mà. Kết hợp với đó, Tailwind CSS được sử dụng để thiết kế giao diện, giúp xây dựng bố cục trực quan, thống nhất và dễ dàng tùy chỉnh giao diện phù hợp với lĩnh vực kinh doanh bánh kem.

Phần backend của hệ thống được xây dựng bằng NodeJS, đóng vai trò xử lý các nghiệp vụ chính như quản lý người dùng, sản phẩm, giỏ hàng và đơn hàng. Dữ liệu của hệ thống được lưu trữ và quản lý bằng MongoDB, một hệ quản trị cơ sở dữ liệu NoSQL, phù hợp với mô hình dữ liệu linh hoạt và hỗ trợ tốt cho các ứng dụng web hiện đại.

Tổng quan lại, đề tài tập trung nghiên cứu cách xây dựng một website bán bánh kem đơn giản nhưng đáp ứng đầy đủ các chức năng cơ bản của một trang bán hàng trực tuyến. Mục tiêu của nghiên cứu không chỉ là tạo ra một sản phẩm có thể sử dụng được, mà còn giúp người thực hiện hiểu rõ hơn về quy trình phát triển web, từ khâu thiết kế, tổ chức nội dung đến cách mô phỏng các hành vi người dùng.

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

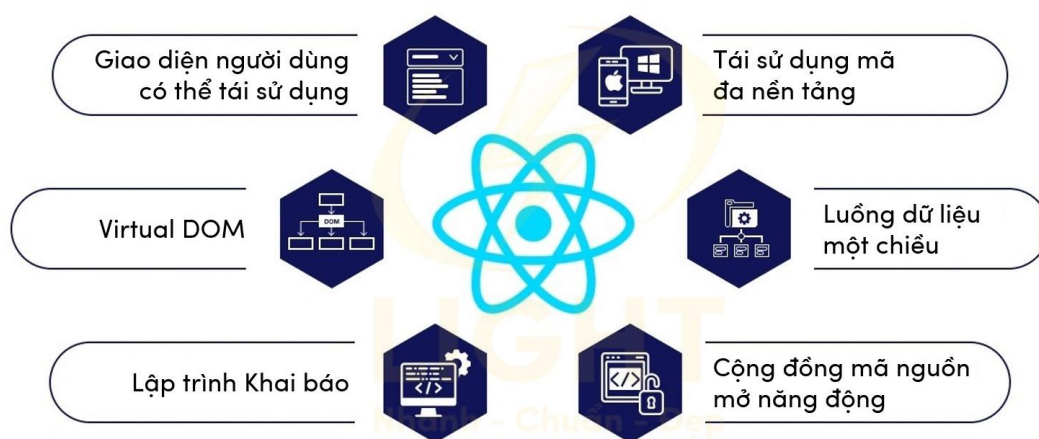
### 2.1 Tổng quan về công nghệ sử dụng

#### 2.1.1 Giới thiệu về ReactJS

##### 2.1.1.1 Khái niệm

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, chuyên dùng để xây dựng giao diện người dùng (UI) cho các ứng dụng web. Thay vì thao tác trực tiếp trên toàn bộ trang web, ReactJS cho phép chia nhỏ giao diện thành các component độc lập, có thể tái sử dụng và dễ dàng quản lý.

Một điểm nổi bật của ReactJS là Virtual DOM, giúp tăng hiệu suất hiển thị. Khi dữ liệu thay đổi, React chỉ cập nhật những phần của giao diện cần thiết thay vì tải lại toàn bộ trang, từ đó cải thiện tốc độ phản hồi và trải nghiệm người dùng [1].



Hình 2.1 Khái niệm ReactJS [1]

##### 2.1.1.2 Các tính năng nổi bật của ReactJS

ReactJS có nhiều tính năng hữu ích cho việc phát triển ứng dụng web, bao gồm:

- **Components:** ReactJS cho phép phát triển ứng dụng web theo mô hình component. Các component là các phần tử UI độc lập có thể được tái sử dụng trong nhiều phần khác nhau của ứng dụng.
- **Virtual DOM:** ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM là một bản sao của DOM được lưu trữ trong bộ nhớ và được cập nhật một cách nhanh chóng khi có thay đổi, giúp tăng tốc độ và hiệu suất của ứng dụng.
- **JSX:** JSX là một ngôn ngữ lập trình phân biệt được sử dụng trong ReactJS để mô tả các thành phần UI. JSX kết hợp HTML và JavaScript, giúp cho việc viết mã dễ hiểu và dễ bảo trì hơn.

- **State và Props:** ReactJS cho phép quản lý trạng thái của các thành phần UI thông qua State và Props. State là trạng thái của một thành phần được quản lý bởi nó chính, trong khi Props là các giá trị được truyền vào từ bên ngoài để tùy chỉnh hoặc điều khiển hành vi của một thành phần.
- **Hỗ trợ đa nền tảng:** ReactJS không chỉ được sử dụng để phát triển ứng dụng web, mà còn được sử dụng để phát triển ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

### 2.1.1.3 Cách thức hoạt động của ReactJS

ReactJS có thể cho phép truyền mã HTML với JavaScript với những lợi ích:

- DOM là một cấu trúc cây biểu diễn tài liệu HTML và JavaScript có thể tương tác với DOM để thay đổi nội dung và cấu trúc của trang web.
- ReactJS sử dụng một thuật toán diff hiệu quả để tìm ra những phần tử cần thay đổi và chỉ cập nhật những phần đó trên DOM thực tế, tránh việc cập nhật lại toàn bộ DOM.
- Khi có nhu cầu đọc và ghi vào DOM, JSX sẽ sử dụng DOM ảo của nó. Sau đó DOM ảo sẽ cố gắng tìm cách hiệu quả để cập nhật DOM của trình duyệt [1].

Trong ReactJS, bạn tạo ra các phần tử React bằng cách sử dụng các hàm JSX như `<div>`, `<button>`,... Các phần tử này không phải là DOM thực, mà là các đối tượng đơn giản được tạo ra dễ dàng.

ReactJS sử dụng một DOM ảo (Virtual DOM) để tối ưu quá trình cập nhật DOM thực. Khi bạn cập nhật một phần tử React, ReactJS sẽ so sánh phần tử mới với phần tử cũ trong DOM ảo, sau đó chỉ cập nhật những phần cần thiết trong DOM thực, nhờ tốc độ xử lý nhanh của JavaScript [1].

Ngoài ra, mặc dù được tạo ra để sử dụng trong trình duyệt nhưng thiết kế của ReactJS sẽ khiến chúng có lợi khi sử dụng trên máy chủ Node.JS.

### 2.1.1.4 So sánh ReactJS và các thư viện Javascript khác

Khi so sánh ReactJS với các framework hoặc thư viện JavaScript phổ biến khác như Angular hoặc Vue.js thì tính linh hoạt sẽ là điểm nổi bật của framework ReactJS. Không giống như Angular hay Vue.js, chúng chỉ ra những cách cụ thể để giải quyết vấn đề thì ReactJS không có quan điểm nhất định. Điều này cho phép các lập

trình viên có nhiều quyền tự do hơn trong việc lựa chọn cách xây dựng cấu trúc ứng dụng của họ.

Khi nói về giao tiếp giữa máy khách (client-side) và máy chủ (server-side), cả Vue.js và ReactJS thường sử dụng Axios trong khi Angular cung cấp module client-side HTTP. Một điểm khác biệt khác về kích thước khi xem xét các đối tác nhẹ như Preact, một giải pháp thay thế nhỏ hơn cung cấp chức năng tương tự như ReactJS.

Tóm lại, ReactJS nổi bật hơn so với các framework và thư viện JavaScript khác nhờ vào tính linh hoạt nó mang lại. Giúp các lập trình viên xây dựng ứng dụng một trang (SPA) một cách dễ dàng hơn [1].

### **2.1.1.5 Ưu điểm và nhược điểm của ReactJS**

#### **• Ưu điểm**

- Hiệu suất cao nhờ Virtual DOM: ReactJS sử dụng Virtual DOM để theo dõi mọi thay đổi trong giao diện. Khi state hoặc props thay đổi, React so sánh Virtual DOM và DOM thực và chỉ cập nhật những phần khác biệt, giúp tăng hiệu suất và giảm thời gian render.

- Quản lý trạng thái hiệu quả: ReactJS có state và props để quản lý dữ liệu nội bộ và dữ liệu truyền giữa các component. Luồng dữ liệu một chiều giúp kiểm soát trạng thái dễ dàng và giảm xung đột dữ liệu trong ứng dụng.

- Dễ dàng tích hợp với các thư viện và công nghệ khác: ReactJS có thể kết hợp với các thư viện như Tailwind CSS, Redux, React Router hoặc các thư viện API khác, giúp xây dựng các ứng dụng web phức tạp một cách linh hoạt.

- Hỗ trợ phát triển đa nền tảng: React Native kế thừa kiến thức từ ReactJS, cho phép phát triển ứng dụng di động cho cả iOS và Android bằng cùng một mã nguồn.

- Cộng đồng lớn và tài nguyên phong phú: ReactJS có cộng đồng lớn, nhiều thư viện, plugin và tài liệu hướng dẫn chi tiết, thuận tiện cho việc học tập và phát triển ứng dụng.

#### **• Nhược điểm**

- Chỉ là thư viện, không phải framework toàn diện: ReactJS chỉ tập trung vào UI, không cung cấp sẵn giải pháp cho routing, quản lý state toàn cục hay tương tác với backend, vì vậy cần kết hợp với các thư viện bổ sung.

- Đường cong học tập cao: ReactJS sử dụng JSX, Virtual DOM, component-based architecture và Hooks, có thể khó tiếp cận với người mới học.



- Cập nhật liên tục: Các phiên bản mới và thay đổi trong ReactJS và các thư viện liên quan có thể gây khó khăn khi bảo trì ứng dụng.

- Phụ thuộc vào JavaScript: Ứng dụng ReactJS yêu cầu trình duyệt hỗ trợ JavaScript. Nếu JavaScript bị tắt hoặc trình duyệt quá cũ, ứng dụng có thể không hiển thị đúng.

## 2.1.2 Giới thiệu về Tailwind CSS

### 2.1.2.1 Khái niệm

Tailwind CSS là một framework CSS utility-first, tương tự như Bootstrap, cung cấp một tập hợp các lớp CSS được tích hợp sẵn mà chúng ta có thể sử dụng trong ứng dụng của mình. Tuy nhiên, điểm đặc biệt của Tailwind CSS là nó cung cấp một số lượng lớn các lớp CSS với các thuộc tính, quy tắc CSS khác nhau.[3] Thay vì sử dụng các lớp CSS được định nghĩa sẵn cho các thành phần cụ thể, Tailwind CSS tập trung vào việc cung cấp các lớp CSS cơ bản mà chúng ta có thể kết hợp để xây dựng giao diện.



*Hình 2.2 Khái niệm Tailwind CSS [3]*

Ví dụ, giả sử bạn muốn tạo một nút có chiều cao cố định, đệm ngang, màu nền đen, các cạnh bo tròn và phông chữ màu trắng, in đậm. Sau đây là HTML bạn sẽ sử dụng:

```
<button class="h-10 px-6 font-semibold rounded-md bg-black text-white" type="submit">Buy now</button>
```

HTML chứa 6 lớp tiện ích. Hãy phân tích từng lớp bên dưới:

- **h-10**: Lớp này đặt nút ở chiều cao cố định là 10 đơn vị.
- **px-6**: Lớp này đặt đệm ngang của nút ở 6 đơn vị.

- **font-semibold:** Lớp này đặt độ đậm của phông chữ của nút thành semibold.
- **rounded-md:** Lớp này đặt bán kính đường viền của nút sao cho các góc được bo tròn.
- **bg-black:** Lớp này đặt màu nền của nút thành màu đen.
- **text-white:** Lớp này đặt màu văn bản của nút thành màu trắng [3].

#### 2.1.2.2 Các đặc điểm nổi bật của Tailwind CSS

- **Khả năng Tùy Biến Cao (Highly Customizable)**

- Tailwind được thiết kế để dễ dàng tùy chỉnh thông qua một tệp cấu hình JavaScript (tailwind.config.js).

- Thiết kế hệ thống (Design System): Bạn có thể định nghĩa toàn bộ hệ thống thiết kế của mình, bao gồm bảng màu, phông chữ, khoảng cách, breakpoints,...

- Mở rộng: Dễ dàng thêm các lớp tiện ích mới hoặc mở rộng các lớp hiện có.

- Ví dụ cấu hình: Thay đổi thang màu mặc định hoặc thêm một font chữ mới.

- **Thiết Kế Đáp Ứng (Responsive Design) Dễ Dàng**

- Tailwind sử dụng các tiền tố để áp dụng các lớp tiện ích cho các kích thước màn hình cụ thể (Breakpoints).

- Các tiền tố mặc định:

- sm: màn hình nhỏ (Small)
- md: màn hình trung bình (Medium)
- lg: màn hình lớn (Large)
- xl: màn hình cực lớn (Extra-Large)
- 2xl: màn hình siêu lớn

- Cách sử dụng:

**<div class="w-full md:w-1/2 lg:w-1/3">...</div>**

- Mặc định (mọi màn hình): chiếm 100% chiều rộng (w-full).
- Từ màn hình trung bình trở lên (md:): chiếm 50% chiều rộng (w-1/2).
- Từ màn hình lớn trở lên (lg:): chiếm 33.3% chiều rộng (w-1/3) [3].

#### 2.1.2.3 Cấu trúc hệ thống và cách Tailwind CSS tổ chức giao diện

- **Hệ thống phân chia theo thang đo (Design Scale)**

- Tailwind thiết kế giao diện dựa trên các thang đo được chuẩn hóa. Ví dụ:

- khoảng cách (spacing)
- cỡ chữ (typography scale)
- độ bo góc
- độ đậm nhạt màu sắc
- độ rộng – độ cao theo tỷ lệ

- Nhờ việc mọi giá trị đều nằm trong quy chuẩn chung, giao diện của sản phẩm trở nên đồng bộ, tránh tình trạng mỗi thành phần sử dụng một kích thước khác nhau.

- **Hệ thống giao diện responsive**

Tailwind áp dụng phương pháp thiết kế tăng tiến (progressive enhancement), hay còn gọi là *mobile-first*. Giao diện được thiết kế phù hợp cho thiết bị nhỏ trước, sau đó mở rộng dần cho thiết bị lớn. Hệ thống breakpoint của Tailwind rõ ràng, dễ nhớ và có thể tùy chỉnh theo nhu cầu dự án. Điều này giúp quá trình xây dựng giao diện responsive trở nên dễ dàng hơn nhiều.

- **Hỗ trợ các trạng thái và chế độ hiển thị phức tạp**

- Không chỉ hỗ trợ các trạng thái như rê chuột, focus hay active, Tailwind còn hỗ trợ các trạng thái nâng cao như:

- chế độ nền tối (Dark Mode)
- trạng thái nhóm (group modifier)
- trạng thái phụ thuộc vào phần tử cha
- các hiệu ứng chuyển động cơ bản

#### **2.1.2.4 Ưu nhược điểm của Tailwind CSS**

- **Ưu điểm**

- Tốc độ phát triển nhanh: Do có thể viết style trực tiếp trong HTML, Tailwind giúp rút ngắn đáng kể thời gian thiết kế giao diện. Bên cạnh đó, việc sử dụng hệ thống lớp tiện ích chuẩn hoá giúp hạn chế tình trạng phải điều chỉnh nhiều lần.

- Không áp đặt thiết kế: Không giống các framework như Bootstrap vốn có phong cách mặc định tương đối rõ rệt, Tailwind hoàn toàn không áp đặt bất kỳ giao diện nào. Điều này giúp lập trình viên tự do thiết kế giao diện phù hợp với cá tính thương hiệu hoặc yêu cầu của sản phẩm mà không cần tốn công ghi đè style có sẵn.

- Giảm độ phức tạp của CSS thủ công: Tailwind giúp giảm thiểu sự phụ thuộc vào file CSS riêng. Việc không phải viết CSS mới cho từng thành phần giúp tránh tình

trạng phân mảnh style, đặc biệt trong những dự án lớn với nhiều lập trình viên cùng tham gia.

- Tối ưu hiệu năng: Nhờ cơ chế chỉ giữ lại các lớp được sử dụng thực tế, Tailwind giúp kích thước CSS trong môi trường sản phẩm (production) nhỏ hơn nhiều so với các framework truyền thống. Điều này giúp cải thiện tốc độ tải trang, ảnh hưởng tích cực đến trải nghiệm người dùng và cả điểm chất lượng SEO [3].

- **Nhược điểm**

- Mã HTML có thể trở nên dài và khó đọc: Do mọi kiểu dáng đều được đưa trực tiếp vào HTML, các thẻ có thể chứa nhiều lớp tiện ích, khiến mã nguồn trở nên dài. Tuy nhiên, điều này có thể được khắc phục bằng việc tổ chức code hợp lý hoặc tách các khối giao diện thành component.

- Đòi hỏi thời gian làm quen: Người mới phải nhớ hệ thống class khá nhiều (ví dụ: bg-blue-500, px-4, text-lg). Ban đầu có thể cảm thấy phức tạp so với CSS thuần.

- Khó quản lý khi thiếu quy ước chung: Nếu team không có guideline, code dễ bị trùng lặp class và khó duy trì về lâu dài.

- Không phù hợp cho dự án nhỏ, đơn giản: Với website tĩnh, ít trang, việc tích hợp Tailwind có thể “thừa tính năng” so với CSS thuần [3].

### 2.1.3 Giới thiệu về NodeJS

#### 2.1.3.1 Khái niệm

Node.js là một môi trường chạy JavaScript phía máy chủ (server-side runtime environment), được xây dựng trên JavaScript V8 Engine của Google – đây cũng chính là động cơ thực thi JavaScript của trình duyệt Chrome. Khác với JavaScript truyền thống vốn chỉ chạy trên trình duyệt, Node.js cho phép lập trình viên sử dụng JavaScript để xây dựng các ứng dụng phía máy chủ như API, dịch vụ web, hệ thống real-time và các ứng dụng mạng [4].



Hình 2.3 Khái niệm NodeJS [4]

- **Nguồn mở (Open-source):** Mã nguồn của Node.js được công bố công khai, điều này có nghĩa là bất kỳ ai cũng có thể truy cập, sử dụng và đóng góp vào mã nguồn. Node.js được duy trì bởi cộng đồng lập trình viên trên toàn thế giới và hướng dẫn đóng góp của Node.js hướng dẫn bạn cách để bạn có thể góp phần phát triển nó.

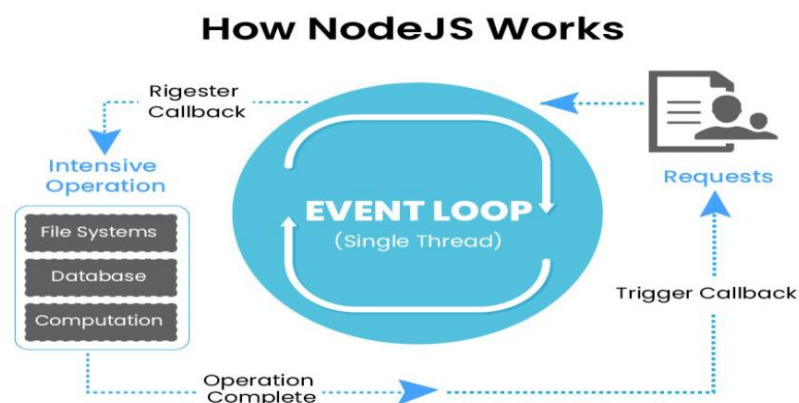
- **Đa nền tảng (Cross-platform):** Node.js không phụ thuộc vào bất kỳ hệ điều hành nào cụ thể nào, nghĩa là nó có thể chạy trên Linux, macOS hoặc Windows. Điều này làm cho Node.js trở thành một lựa chọn linh hoạt cho các nhà phát triển muốn xây dựng các ứng dụng có thể hoạt động trên nhiều nền tảng khác nhau mà không cần thay đổi mã nguồn.

- **Môi trường thực thi JavaScript (JavaScript runtime environment):** Để mã JavaScript có thể được thực thi, nó cần một môi trường chạy phù hợp. Trong khi trình duyệt như Chrome và Firefox cung cấp một môi trường thực thi cho JavaScript, Node.js mở rộng khả năng này ra ngoài trình duyệt. Node.js cho phép chạy JavaScript trên máy chủ hoặc trong bất kỳ môi trường máy tính nào khác, không chỉ trong trình duyệt.

- **Dựa trên V8 JavaScript Engine:** Node.js được xây dựng dựa trên V8, động cơ JavaScript được phát triển bởi Google cho trình duyệt Chrome. Điều này giúp Node.js có khả năng thực thi JavaScript nhanh và hiệu quả, đồng thời hỗ trợ các tính năng mới nhất của ngôn ngữ JavaScript [2].

### 2.1.3.2 Cách hoạt động của NodeJS

Node.js hoạt động dựa trên một số nguyên tắc cơ bản giúp nó hiệu quả trong việc xử lý các ứng dụng có nhiều hoạt động nhập/xuất (I/O) mà không bị chặn, đồng thời giảm đáng kể sự phức tạp trong quản lý các luồng thực thi. Dưới đây là một số thành phần chính giải thích cách thức hoạt động của Node.js [2].



Hình 2.4 Luồng hoạt động của NodeJS [2]

- Kiến trúc Non-blocking I/O và Event-Driven:

Node.js sử dụng một mô hình non-blocking I/O (input/output) và event-driven, nghĩa là các hoạt động như đọc file, truy vấn cơ sở dữ liệu hoặc giao tiếp mạng được thực hiện mà không chặn tiến trình chính.

- V8 JavaScript Engine

Node.js được xây dựng trên động cơ JavaScript V8 của Google Chrome, đây là một động cơ rất nhanh cho phép biên dịch mã JavaScript thành mã máy để thực thi trực tiếp trên phần cứng, làm tăng hiệu suất thực thi.

- Single-Threaded

Mặc dù Node.js hoạt động trên một luồng duy nhất cho logic ứng dụng của người dùng, nó vẫn sử dụng nhiều luồng ở tầng thấp hơn thông qua thư viện libuv để xử lý các hoạt động I/O. Tuy nhiên, những chi tiết này được ẩn giấu khỏi người dùng, giúp việc lập trình đơn giản hơn mà vẫn đảm bảo hiệu suất.

- Event Loop

Trái tim của Node.js là “event loop”. Đây là vòng lặp sự kiện mà ở đó Node.js tiếp tục lắng nghe sự kiện và thực hiện các hàm gọi lại khi một sự kiện được kích hoạt. Vòng lặp sự kiện cho phép Node.js xử lý hàng nghìn kết nối đồng thời mà không cần phải tạo ra chi phí quản lý luồng.

- NPM (Node Package Manager)

NPM là hệ thống quản lý gói cho Node.js, cho phép các nhà phát triển dễ dàng chia sẻ và sử dụng mã nguồn từ nhau. NPM là một trong những kho lưu trữ mã nguồn mở lớn nhất thế giới và chứa hàng ngàn module có thể được tích hợp vào ứng dụng của bạn.

- Require

Require làm 3 thứ:

- Tải module đi kèm với Node.js như hệ thống file và HTTP từ Node.js API.
- Tải thư viện thứ 3 như Express và Mongoose mà bạn cài đặt từ npm.
- Giúp bạn require file của bạn và mô-đun hoá project.

Require là 1 chức năng và nó nhận tham số path tĩnh chính và trả về *module.export* [2].

### 2.1.3.3 Ưu nhược điểm của NodeJS

- **Ưu điểm**

- Có tốc độ xử lý nhanh nhờ cơ chế xử lý bất đồng bộ (non-blocking). Bạn có thể dễ dàng xử lý hàng ngàn kết nối trong khoảng thời gian ngắn nhất.
- Giúp dễ dàng mở rộng khi có nhu cầu phát triển website.
- Nhận và xử lý nhiều kết nối chỉ với một single-thread. Nhờ đó, hệ thống xử lý sẽ sử dụng ít lượng RAM nhất và giúp quá trình xử lý Nodejs nhanh hơn rất nhiều.
- Có khả năng xử lý nhiều Request/s cùng một lúc trong thời gian ngắn nhất.
- Có khả năng xử lý hàng ngàn Process cho hiệu suất đạt mức tối ưu nhất.

- **Nhược điểm**

- Nodejs gây hao tốn tài nguyên và thời gian. Nodejs được viết bằng C++ và JavaScript nên khi xử lý cần phải trải qua một quá trình biên dịch. Nếu bạn cần xử lý những ứng dụng tốn tài nguyên CPU thì không nên sử dụng Nodejs.
- Nodejs so với các ngôn ngữ khác như PHP, Ruby và Python sẽ không có sự chênh lệch quá nhiều. Nodejs có thể sẽ phù hợp với việc phát triển ứng dụng mới. Tuy nhiên khi xây dựng và triển khai dự án quan trọng thì Nodejs không phải là sự lựa chọn hoàn hảo nhất [4].

### 2.1.4 Giới thiệu về MongoDB

#### 2.1.4.1 Khái niệm

MongoDB là một chương trình cơ sở dữ liệu mã nguồn mở được thiết kế theo kiểu hướng đối tượng trong đó các bảng được cấu trúc một cách linh hoạt cho phép các dữ liệu lưu trên bảng không cần phải tuân theo một dạng cấu trúc nhất định nào. Chính do cấu trúc linh hoạt này nên MongoDB có thể được dùng để lưu trữ các dữ liệu có cấu trúc phức tạp và đa dạng và không cố định (hay còn gọi là Big Data).



Hình 2.5 Khái niệm MongoDB [5]

#### 2.1.4.2 Các tính năng nổi bật của MongoDB

Mongodb được sử dụng rất nhiều và được đánh giá vô cùng cao nhờ sở hữu nhiều đặc điểm nổi trội là:

- Mongodb chính là một database hướng tài liệu, nên khi đó mọi dữ liệu sẽ được lưu trữ trong document theo kiểu JSON thay vì lưu theo dạng bảng như CSDL quan hệ nên việc truy cập vẫn sẽ nhanh chóng hơn.
- Với các CSDL quan hệ thì chúng ta sẽ có khái niệm bảng, khi đó các cơ sở dữ liệu quan hệ sẽ sử dụng các bảng để có thể lưu trữ dữ liệu, còn với Mongodb thì bạn cần phải sử dụng khái niệm collection thay cho bảng.
- Mongodb chính là một hệ quản trị cơ sở dữ liệu mà trong đó mã nguồn mở là CSDL thường thuộc NoSql và được hàng triệu người sử dụng.
- So với RDBMS thì trong Mongodb collection thường sẽ ứng với table, còn document sẽ tương ứng với row. Với các collection có trong Mongodb thường sẽ được cấu trúc rất linh hoạt nên nó cho phép các dữ liệu được lưu trữ mà không cần phải tuân theo bất kỳ một cấu trúc nhất định nào [5].

#### 2.1.4.3 Ưu nhược điểm của MongoDB

- **Ưu điểm**

- Bởi vì Mongodb sử dụng các dữ liệu dưới dạng Document JSON nên mỗi một collection đều có kích cỡ và document khác nhau. Nhưng chúng lại rất linh hoạt khi thực hiện lưu trữ bởi vậy nếu bạn muốn thứ gì thì chỉ cần insert thoải mái.

- Các dữ liệu có trong Mongodb thường không ràng buộc lẫn nhau, chúng không có join như trong RDBMS, nên khi bạn insert, xóa hoặc update thì sẽ không phải bỏ ra quá nhiều thời gian để kiểm tra chúng có thỏa mãn các ràng buộc như trong RDBMS hay không.

- Mongodb dễ mở rộng được và trong Mongodb luôn có khái niệm cluster chính là cụm các node sẽ có chứa các dữ liệu giao tiếp với nhau. Nên chỉ cần bạn muốn mở rộng hệ thống thì chỉ việc thêm một node mới vào cluster.

- Các trường hợp dữ liệu “\_id” sẽ luôn được đánh tự động index, nên tốc độ truy vấn thông tin sẽ luôn đạt hiệu suất cao nhất.

- Tốc độ truy vấn của Mongodb luôn nhanh hơn so với các hệ quản trị cơ sở dữ liệu quan hệ.



- Nếu như có một truy vấn dữ liệu, thì bản ghi sẽ được cached lên bộ nhớ Ram. Khi đó, việc phục vụ lượt truy vấn sau sẽ diễn ra nhanh hơn mà bạn không cần phải đọc từ ổ cứng.

- **Nhược điểm**

- Mongoddb không sở hữu các tính chất ràng buộc như trong RDBMS nên khi bạn thao tác với Mongoddb cần phải cẩn thận hết sức.

- Có thể sẽ tốn bộ nhớ do dữ liệu được lưu trữ dưới dạng key-value, nên các collection sẽ chỉ khác về value do vậy mà key có thể sẽ bị lặp lại. Mongoddb còn không hỗ trợ join nên rất dễ bị dữ thừa dữ liệu.

- Khi thực hiện insert/update/remove bản ghi thì MongoDB sẽ chưa thể cập nhật ngay vào ổ cứng. Chỉ sau 60 giây thì Mongoddb mới có thể ghi được toàn bộ dữ liệu được thay đổi từ RAM xuống phần ổ cứng. Điều này chính là nhược điểm bởi nó có thể mang lại nguy cơ mất dữ liệu khi các tình huống xấu như mất điện xảy ra [5].

## **2.2 Giới thiệu nghiệp vụ của đề tài**

### **2.2.1 Tổng quan nghiệp vụ**

Hệ thống website bán bánh kem được xây dựng nhằm đáp ứng nhu cầu đặt mua bánh trực tuyến của khách hàng một cách nhanh chóng, tiện lợi và chính xác. Thay vì phải đến trực tiếp cửa hàng, khách hàng có thể dễ dàng tham khảo các mẫu bánh, lựa chọn kích thước, hương vị, kiểu trang trí và đặt hàng thông qua website. Hệ thống đóng vai trò là kênh bán hàng trực tuyến, hỗ trợ cửa hàng bánh kem mở rộng phạm vi tiếp cận khách hàng.

Website được phát triển bằng ReactJS kết hợp với TailwindCSS, hướng đến giao diện hiện đại, trực quan và khả năng tương tác cao. Về mặt nghiệp vụ, hệ thống đảm bảo sự phối hợp chặt chẽ giữa khách hàng và bộ phận quản lý cửa hàng trong toàn bộ vòng đời đơn hàng, từ khâu lựa chọn sản phẩm, đặt hàng, thanh toán cho đến giao nhận bánh.

### **2.2.2 Nghiệp vụ dành cho khách hàng**

Khi truy cập vào website, khách hàng có thể xem danh sách các loại bánh kem hiện có, được phân loại theo nhiều tiêu chí như bánh sinh nhật, bánh entrement, bánh mousse,.... Mỗi sản phẩm bánh kem được cung cấp đầy đủ thông tin bao gồm hình ảnh minh họa, mô tả ngắn, thành phần cơ bản, giá bán. Việc hiển thị rõ ràng thông tin giúp khách hàng dễ dàng lựa chọn sản phẩm phù hợp với nhu cầu và ngân sách.

Khách hàng có thể thêm bánh vào giỏ hàng để tiếp tục mua sắm hoặc tiến hành đặt hàng ngay. Trong quá trình đặt hàng, khách hàng cần cung cấp thông tin nhận hàng như tên người nhận, số điện thoại, địa chỉ giao hàng và thời gian mong muốn nhận bánh. Đối với các đơn hàng đặc biệt hoặc bánh theo yêu cầu riêng, khách hàng có thể liên hệ trực tiếp để cửa hàng nắm rõ yêu cầu cần thực hiện.

Sau khi xác nhận đơn hàng, khách hàng thực hiện thanh toán theo các phương thức mà hệ thống hỗ trợ. Khi đơn hàng được ghi nhận thành công, hệ thống sẽ gửi thông báo xác nhận cho khách hàng thông qua email hoặc hiển thị trực tiếp trên website. Khách hàng có thể theo dõi trạng thái đơn hàng từ lúc tiếp nhận, đang chuẩn bị bánh cho đến khi xác nhận đơn hàng.

### **2.2.3 Nghiệp vụ dành cho quản trị viên**

Quản trị viên là người chịu trách nhiệm quản lý và vận hành toàn bộ website bán bánh kem. Thông qua trang quản trị, quản trị viên có thể thực hiện nghiệp vụ quản lý sản phẩm. Mỗi sản phẩm được quản lý chi tiết về hình ảnh, giá bán, mô tả và trạng thái kinh doanh nhằm đảm bảo thông tin hiển thị trên website luôn chính xác và đầy đủ.

Bên cạnh đó, quản trị viên thực hiện nghiệp vụ quản lý đơn hàng. Hệ thống cho phép quản trị viên theo dõi danh sách các đơn hàng theo từng trạng thái như đơn mới, đang xử lý, đã hoàn thành hoặc đã hủy. Đối với mỗi đơn hàng, quản trị viên có thể xem chi tiết thông tin sản phẩm, yêu cầu tùy chỉnh của khách hàng và thời gian giao hàng để chủ động trong việc chuẩn bị bánh.

### **2.2.4 Nghiệp vụ xử lý đặt hàng và tiếp nhận đơn hàng**

Trước tiên, hệ thống tiếp nhận thông tin đặt hàng từ khách hàng, bao gồm sản phẩm đã chọn, các tùy chỉnh liên quan đến bánh và thông tin liên hệ cần thiết. Toàn bộ dữ liệu này được hệ thống kiểm tra nhằm đảm bảo tính hợp lệ trước khi chuyển sang bước xác nhận đơn hàng. Việc kiểm tra giúp hạn chế sai sót trong quá trình xử lý và đảm bảo yêu cầu của khách hàng được ghi nhận đầy đủ.

Trong quá trình xử lý đơn hàng, hệ thống cho phép cửa hàng theo dõi danh sách các đơn đặt bánh để chuẩn bị sản phẩm đúng theo nhu cầu khách hàng. Trường hợp khách hàng hủy đơn trước khi bánh được chuẩn bị, hệ thống cập nhật trạng thái đơn hàng tương ứng nhằm đảm bảo dữ liệu quản lý luôn chính xác. Nghiệp vụ đặt hàng kết thúc khi đơn hàng được hoàn tất và khách hàng thực hiện thanh toán khi nhận bánh.

## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Mô tả bài toán

Hệ thống website bán bánh kem trực tuyến được xây dựng nhằm mục tiêu cung cấp một nền tảng thương mại điện tử giúp khách hàng có thể dễ dàng tìm kiếm, tham khảo thông tin chi tiết và đặt mua các loại bánh yêu thích từ nhiều danh mục khác nhau bánh sinh nhật, bánh ngọt,... Bài toán đặt ra cho hệ thống là thiết kế và hiện thực hóa một quy trình mua sắm, cho phép người dùng tương tác hiệu quả với cơ sở dữ liệu sản phẩm, bao gồm việc tra cứu thông tin bánh, quản lý giỏ hàng cá nhân,... Các yêu cầu cơ bản của bài toán có thể được mô tả như sau:

- **Quản lý người dùng:** Hệ thống cần cho phép người dùng đăng ký tài khoản mới và đăng nhập vào hệ thống để thực hiện các giao dịch mua hàng. Việc xác thực người dùng giúp lưu trữ lịch sử mua hàng và thông tin giao nhận cho các lần sau.
- **Tìm kiếm và Tra cứu sản phẩm:** Một tính năng quan trọng của hệ thống là khả năng tìm kiếm bánh kem theo tên, loại bánh hoặc khoảng giá. Người dùng có thể sử dụng các bộ lọc tìm kiếm danh mục để nhanh chóng tìm ra sản phẩm phù hợp với nhu cầu.
- **Quản lý giỏ hàng và Đặt hàng:** Website phải có khả năng xử lý nghiệp vụ giỏ hàng, cho phép người dùng thêm sản phẩm vào giỏ, cập nhật số lượng hoặc xóa bỏ sản phẩm. Hệ thống cung cấp quy trình thanh toán minh bạch, tính toán chính xác tổng tiền bao gồm phí vận chuyển và xác nhận đơn hàng thành công.
- **Giao diện người dùng thân thiện:** Giao diện người dùng của website cần phải được thiết kế bắt mắt phù hợp với lĩnh vực thực phẩm, trực quan và dễ sử dụng.
- **Quản lý thông tin (Dành cho Admin):** Hệ thống cần cung cấp các công cụ để quản trị viên có thể cập nhật thông tin sản, quản lý danh mục, quản lý tài khoản, quản lý đơn hàng, quản lý banner, quản lý liên hệ, quản lý đánh giá, thống kê doanh thu một cách nhanh chóng vào cơ sở dữ liệu.

### 3.2 Yêu cầu hệ thống

#### 3.2.1 Yêu cầu về chức năng

- **Đối với khách hàng:**
  - Đăng ký và Đăng nhập: Cho phép tạo tài khoản mới và đăng nhập để lưu trữ thông tin cá nhân.

- Tìm kiếm và Lọc sản phẩm: Hỗ trợ tìm bánh theo tên, lọc theo danh mục và mức giá.
- Xem chi tiết sản phẩm: Hiện thị đầy đủ hình ảnh, giá bán, mô tả thành phần và tình trạng còn hàng.
- Bình luận và đánh giá sản phẩm: Sau khi hoàn thành một đơn hàng, khách hàng có thể để lại lời đánh giá của mình.
- Quản lý giỏ hàng: Cho phép thêm sản phẩm, cập nhật số lượng hoặc xóa sản phẩm khỏi giỏ hàng.
- Đặt hàng trực tuyến: Nhập thông tin giao hàng, xác nhận đơn hàng và hiện thị hóa đơn tạm tính.
- Lịch sử đơn hàng: Xem lại danh sách các đơn hàng đã đặt và trạng thái hiện tại.

- **Đối với quản trị viên:**

- Quản lý sản phẩm: Thực hiện các chức năng thêm mới, cập nhật thông tin (giá, ảnh, mô tả) hoặc xóa bánh khỏi danh sách bán.
- Quản lý danh mục: Phân loại sản phẩm để dễ dàng quản lý.
- Quản lý banner: Thực hiện các chức năng thêm mới, cập nhật và xóa banner cho hệ thống.
- Quản lý khách hàng: cho phép quản trị viên theo dõi thông tin khách hàng đã đăng ký hoặc mua hàng có thể xóa các tài khoản vi phạm.
- Quản lý đơn hàng: Xem danh sách đơn hàng cũng như xem thông tin chi tiết của từng đơn hàng, cập nhật trạng thái đơn hàng.
- Quản lý đánh giá: Quản trị viên có thể xem tất cả các bài đánh giá, từ đó có thể chọn ẩn một số bình luận vi phạm và có thể phản hồi lại bình luận cho khách hàng.
- Quản lý liên hệ: Quản trị viên nhận tin nhắn thắc mắc hoặc cần tư vấn của khách hàng và trả lời lại tin nhắn.
- Thống kê báo cáo: Xem tổng quan số lượng đơn hàng và doanh thu theo thời gian thực.

### 3.2.2 Yêu cầu phi chức năng

Các yêu cầu về chất lượng và ràng buộc kỹ thuật của hệ thống bao gồm:

- Hiệu năng: Tốc độ tải trang nhanh, phản hồi thao tác người dùng mượt mà dưới 2 giây thông qua cơ chế SPA của ReactJS.

- Tính khả dụng: Hệ thống hoạt động ổn định 24/7, giao diện trực quan, dễ sử dụng ngay cả với người dùng không am hiểu công nghệ.

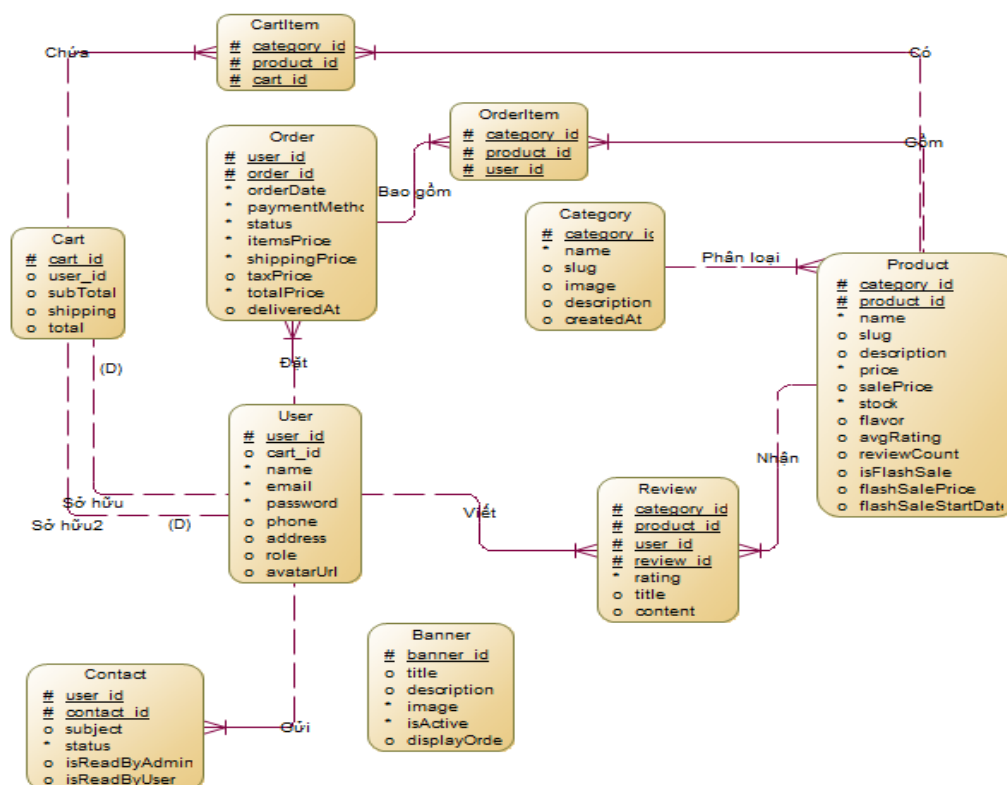
### 3.3 Thiết kế cơ sở dữ liệu

#### 3.3.1 Mô hình dữ liệu của hệ thống

Sơ đồ thể hiện cấu trúc dữ liệu của một hệ thống bán hàng trực tuyến. Người dùng (User) giữ vai trò trung tâm, mỗi người dùng sở hữu một giỏ hàng (Cart) riêng để lưu các sản phẩm muốn mua. Trong giỏ hàng, mỗi sản phẩm được quản lý thông qua CartItem, thể hiện mối quan hệ một giỏ hàng có thể chứa nhiều sản phẩm khác nhau.

Khi người dùng tiến hành đặt hàng, hệ thống tạo ra đơn hàng (Order) và mỗi đơn hàng có thể bao gồm nhiều sản phẩm thông qua OrderItem. Các sản phẩm (Product) được tổ chức theo từng danh mục (Category), giúp việc phân loại và hiển thị sản phẩm trở nên rõ ràng.

Ngoài chức năng mua hàng, người dùng còn có thể đánh giá sản phẩm thông qua Review, thể hiện mối quan hệ nhiều đánh giá gắn với một sản phẩm và một người dùng cụ thể. Bên cạnh đó, người dùng có thể gửi phản hồi hoặc liên hệ với hệ thống qua Contact. Cuối cùng, Banner là thành phần độc lập dùng để quản lý các nội dung quảng bá và hiển thị trên giao diện trang web.



Hình 3.1 Mô hình dữ liệu của hệ thống

### 3.3.2 Danh sách các bảng dữ liệu

*Bảng 3.1 Bảng User*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
user_id	INT	PK	Mã người dùng
name	VARCHAR		Họ tên người dùng
email	VARCHAR		Email đăng nhập
password	VARCHAR		Mật khẩu đã mã hóa
phone	VARCHAR		Số điện thoại
address	VARCHAR		Địa chỉ
role	VARCHAR		Vai trò (user / admin)
avatarUrl	VARCHAR		Ảnh đại diện
createdAt	DATETIME		Ngày tạo tài khoản

*Bảng 3.2 Bảng Category*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
category_id	INT	PK	Mã danh mục
name	VARCHAR		Tên danh mục
slug	VARCHAR		Đường dẫn SEO
image	VARCHAR		Ảnh đại diện danh mục
description	VARCHAR		Mô tả danh mục
createdAt	DATETIME		Ngày tạo

*Bảng 3.3 Bảng Product*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
product_id	INT	PK	Mã sản phẩm
name	VARCHAR		Tên bánh
slug	VARCHAR		Đường dẫn SEO
description	VARCHAR		Mô tả sản phẩm
price	DECIMAL		Giá bán
salePrice	DECIMAL		Giá khuyến mãi
stock	INT		Số lượng tồn kho
flavor	VARCHAR		Hương vị
avgRating	FLOAT		Điểm đánh giá trung bình
reviewCount	INT		Số lượt đánh giá
category_id	INT	FK	Tham chiếu danh mục
createdAt	DATETIME		Ngày tạo

*Bảng 3.4 Bảng Cart*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
cart_id	INT	PK	Mã giỏ hàng
user_id	INT	FK	Người sở hữu giỏ hàng
subTotal	DECIMAL		Tổng tiền sản phẩm
shipping	DECIMAL		Phí vận chuyển
total	DECIMAL		Tổng tiền thanh toán
createdAt	DATETIME		Ngày tạo

*Bảng 3.5 Bảng CartItem*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
cart_item_id	INT	PK	Mã giỏ hàng
product_id	INT	PK	Mã sản phẩm
quantity	INT		Số lượng
price	DECIMAL		Giá tại thời điểm thêm
image	VARCHAR		Hình ảnh sản phẩm

*Bảng 3.6 Bảng Order*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
order_id	INT	PK	Mã đơn hàng
user_id	INT	FK	Người đặt hàng
status	VARCHAR		Trạng thái đơn
itemsPrice	DECIMAL		Tổng tiền sản phẩm
shippingPrice	DECIMAL		Phí vận chuyển
taxPrice	DECIMAL		Thuế
totalPrice	DECIMAL		Tổng thanh toán
createdAt	DATETIME		Ngày đặt hàng

*Bảng 3.7 Bảng OrderItem*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
order_item_id	INT	PK	Mã đơn hàng
product_id	INT	PK	Mã sản phẩm
quantity	INT		Số lượng
price	DECIMAL		Giá tại thời điểm mua
category_id			
user_id			
order_id			

*Bảng 3.8 Bảng Review*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
review_id	INT	PK	Mã đánh giá
user_id	INT	FK	Người đánh giá
product_id	INT	FK	Sản phẩm được đánh giá
rating	INT		Số sao
title	VARCHAR		Tiêu đề đánh giá
content	VARCHAR		Nội dung đánh giá
createdAt	DATETIME		Ngày đánh giá

*Bảng 3.9 Bảng Banner*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
banner_id	INT	PK	Mã banner
title	VARCHAR		Tiêu đề
description	VARCHAR		Mô tả
image	VARCHAR		Hình ảnh
isActive	BOOLEAN		Trạng thái hiển thị
createdAt	DATETIME		Ngày tạo

*Bảng 3.10 Bảng Contact*

Tên thuộc tính	Kiểu dữ liệu	Khóa	Mô tả
banner_id	INT	PK	Mã banner
title	VARCHAR		Tiêu đề
description	VARCHAR		Mô tả
image	VARCHAR		Hình ảnh
isActive	BOOLEAN		Trạng thái hiển thị
createdAt	DATETIME		Ngày tạo

### 3.4 Mô hình xử lý

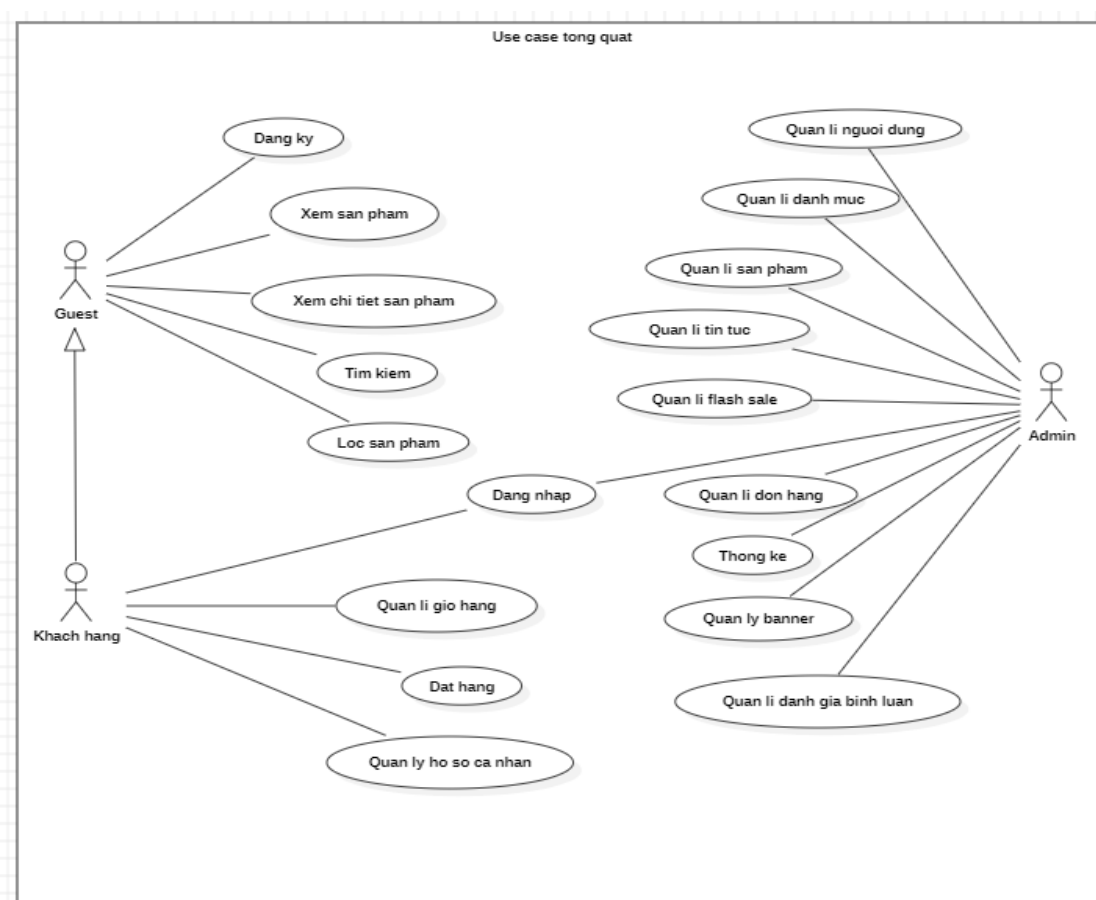
#### 3.4.1 Sơ đồ Use case tổng quát

**Khách** là những người chưa đăng ký hoặc chưa đăng nhập vào hệ thống. Tác nhân này có thể thực hiện một số hành động cơ bản như duyệt qua các sản phẩm, xem chi tiết sản phẩm và tìm kiếm sản phẩm trong hệ thống. Khách cũng có thể lọc các sản phẩm theo các tiêu chí khác nhau, giúp việc tìm kiếm trở nên dễ dàng hơn. Tuy nhiên, họ không thể thực hiện các thao tác mua hàng hoặc quản lý thông tin cá nhân.



**Khách hàng:** là người đã đăng ký và có tài khoản trong hệ thống. Sau khi đăng nhập, khách hàng có thể thực hiện các hành động như thêm sản phẩm vào giỏ hàng, thay đổi số lượng sản phẩm trong giỏ, hoặc tiến hành đặt hàng khi đã hoàn tất việc chọn sản phẩm. Họ cũng có thể quản lý thông tin cá nhân của mình, bao gồm việc cập nhật địa chỉ giao hàng hoặc thay đổi thông tin thanh toán. Những quyền này giúp khách hàng có thể trải nghiệm quá trình mua sắm trực tuyến một cách tiện lợi và nhanh chóng.

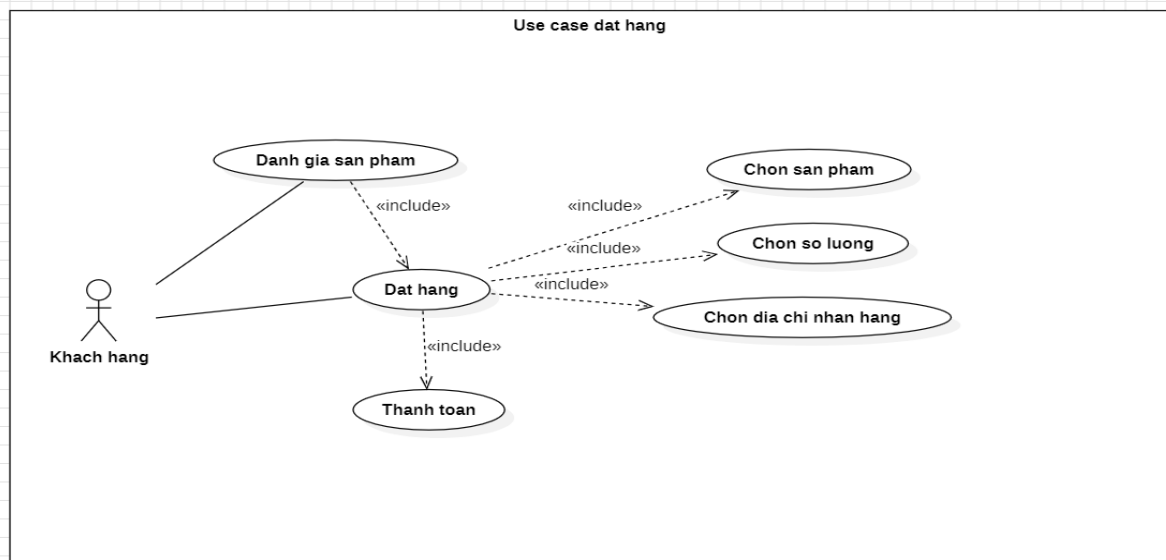
**Quản trị viên:** là người có quyền quản lý toàn bộ hệ thống và các hoạt động của người dùng. Tác nhân này có thể thực hiện nhiều chức năng quan trọng như quản lý người dùng, bao gồm việc cấp quyền hoặc khóa tài khoản. Quản trị viên cũng có quyền quản lý sản phẩm, thêm mới, chỉnh sửa hoặc xóa sản phẩm trong danh mục. Bên cạnh đó, họ còn có thể quản lý các chương trình giảm giá flash sale, tin tức trên website, và thậm chí là các banner quảng cáo. Quản trị viên cũng có thể theo dõi và xử lý các đơn hàng, thực hiện các thống kê báo cáo và kiểm duyệt các bình luận, đánh giá của khách hàng về sản phẩm.



Hình 3.2 Use case tổng quát

### 3.4.2 Sơ đồ Use case đặt hàng

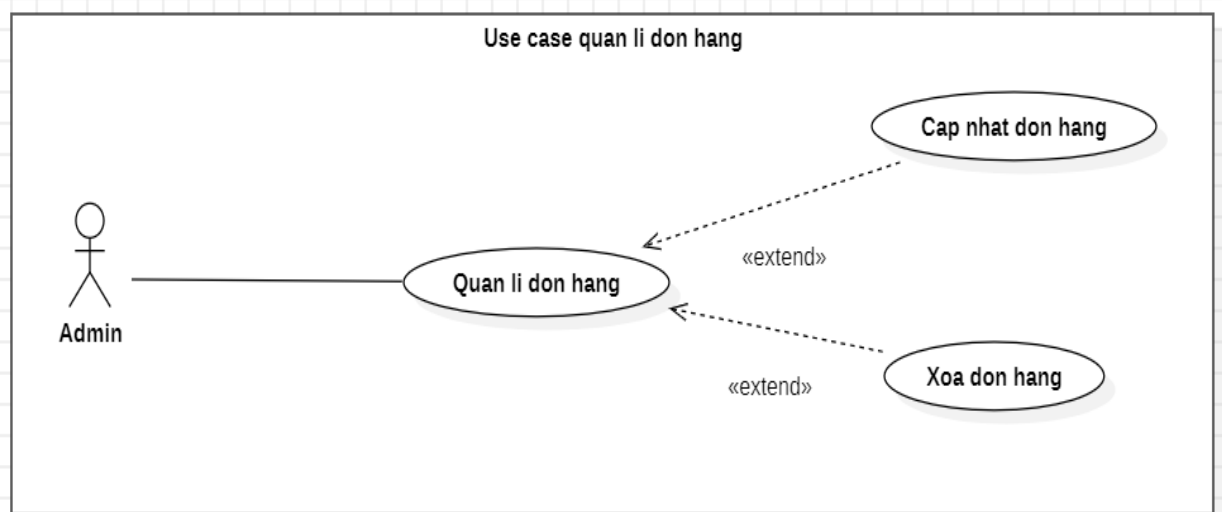
Sơ đồ xác định rõ các bước bắt buộc để thực hiện thành công Use Case 'Đặt hàng' khách hàng phải hoàn tất chuỗi thao tác gồm chọn sản phẩm, xác định số lượng, cung cấp địa chỉ nhận hàng và thực hiện thanh toán.



Hình 3.3 Use case đặt hàng

### 3.4.3 Sơ đồ Use case quản lý đơn hàng

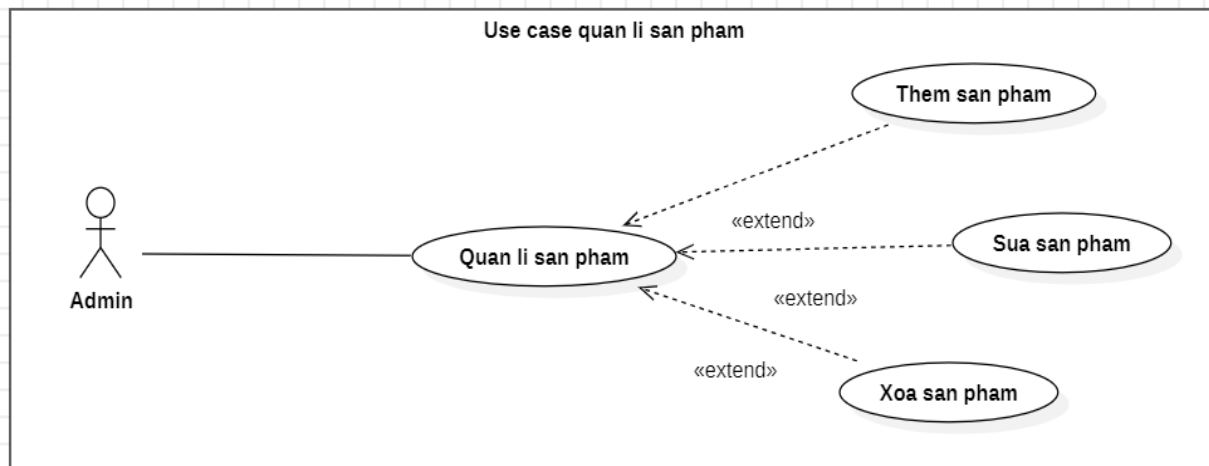
Use Case 'Quản lý sản phẩm' đóng vai trò là chức năng cơ sở, liên kết với các Use Case mở rộng bao gồm: Thêm, Sửa và Xóa sản phẩm. Cơ chế này cho phép Admin linh hoạt thực hiện các tác vụ duy trì và cập nhật danh mục hàng hóa theo nhu cầu thực tế.



Hình 3.4 Use case quản lý đơn hàng

### 3.4.4 Sơ đồ Use case quản lý sản phẩm

Từ Use Case chính là "Quản lý sản phẩm", Admin có thể thực hiện các chức năng mở rộng tùy theo nhu cầu bao gồm: thêm sản phẩm mới vào kho, cập nhật thông tin sản phẩm hoặc xóa sản phẩm khỏi hệ thống.



Hình 3.5 Use case quản lý sản phẩm

## 3.5 Thiết kế API

### 3.5.1 API quản lý người dùng

Nhóm API quản lý người dùng phục vụ cho các chức năng đăng ký, đăng nhập và quản lý thông tin người dùng. Các API này thao tác trực tiếp với bảng User trong cơ sở dữ liệu, giúp hệ thống xác thực người dùng và phân quyền giữa khách hàng và quản trị viên. Các API này đảm bảo rằng người dùng có thể dễ dàng tạo tài khoản, truy cập hệ thống, cũng như cập nhật thông tin cá nhân của mình. Ngoài ra, chúng còn đóng vai trò quan trọng trong việc kiểm soát quyền truy cập và bảo mật thông tin của từng loại người dùng, đảm bảo rằng mỗi đối tượng người dùng chỉ có quyền truy cập vào những chức năng phù hợp với vai trò của mình trong hệ thống.

Bảng 3.11 API quản lý người dùng

Method	Endpoint	Mô tả
POST	/api/users/register	Đăng ký tài khoản người dùng
POST	/api/users/login	Đăng nhập hệ thống
GET	/api/users/profile	Lấy thông tin cá nhân
PUT	/api/users/profile	Cập nhật thông tin người dùng

### 3.5.2 API danh mục sản phẩm

Nhóm API danh mục hỗ trợ việc phân loại bánh kem theo nhóm, giúp khách hàng dễ dàng tìm kiếm sản phẩm và giúp quản trị viên quản lý nội dung. Các API này thao tác với bảng Category trong cơ sở dữ liệu để quản lý các danh mục sản phẩm, đảm bảo sự phân loại rõ ràng và hiệu quả. Chúng cho phép quản trị viên thêm, cập nhật và xóa các danh mục, trong khi khách hàng có thể dễ dàng truy cập và tìm kiếm sản phẩm trong từng nhóm danh mục cụ thể.

*Bảng 3.12 API danh mục sản phẩm*

Method	Endpoint	Mô tả
GET	/api/categories	Lấy danh sách danh mục
POST	/api/categories	Thêm danh mục mới
PUT	/api/categories/{id}	Cập nhật danh mục
DELETE	/api/categories/{id}	Xóa danh mục

### 3.5.3 API sản phẩm

Nhóm API sản phẩm phục vụ cho nghiệp vụ hiển thị, tìm kiếm và quản lý bánh kem trong hệ thống. Các API này thao tác với bảng Product và liên kết với bảng Category để quản lý thông tin các sản phẩm bánh kem, giúp khách hàng tìm kiếm và xem chi tiết sản phẩm, đồng thời giúp quản trị viên quản lý sản phẩm hiệu quả.

*Bảng 3.13 API sản phẩm*

Method	Endpoint	Mô tả
GET	/api/products	Lấy danh sách bánh kem
GET	/api/products/{id}	Lấy chi tiết bánh kem
POST	/api/products	Thêm sản phẩm mới
PUT	/api/products/{id}	Cập nhật thông tin sản phẩm
DELETE	/api/products/{id}	Xóa sản phẩm

### 3.5.4 API giỏ hàng

API giỏ hàng là một tập hợp các dịch vụ mà hệ thống cung cấp để hỗ trợ người dùng trong việc quản lý các sản phẩm mà họ đã chọn trước khi quyết định đặt hàng. Nhóm API này chủ yếu tương tác với hai bảng trong cơ sở dữ liệu: Cart và CartItem, Những API này giúp người dùng dễ dàng thao tác và quản lý giỏ hàng, từ việc thêm

sản phẩm mới đến việc chỉnh sửa hoặc xóa sản phẩm không cần thiết trước khi tiến hành thanh toán.

*Bảng 3.14 API giỏ hàng*

Method	Endpoint	Mô tả
GET	/api/cart	Lấy giỏ hàng của người dùng
POST	/api/cart/add	Thêm sản phẩm vào giỏ
PUT	/api/cart/update	Cập nhật số lượng sản phẩm
DELETE	/api/cart/remove/{id}	Xóa sản phẩm khỏi giỏ

### 3.5.5 API đặt hàng

API đặt hàng phục vụ nghiệp vụ tiếp nhận đơn hàng với hình thức thanh toán khi nhận hàng. Các API này sử dụng dữ liệu từ bảng Order và OrderItem. Các API này cung cấp các chức năng cơ bản để quản lý giỏ hàng trước khi tạo đơn hàng và thanh toán. Sau khi các sản phẩm trong giỏ hàng đã được xác nhận, hệ thống sẽ tiếp tục xử lý đơn hàng và tiến hành giao hàng với hình thức thanh toán khi nhận hàng.

*Bảng 3.15 API đặt hàng*

Method	Endpoint	Mô tả
GET	/api/cart	Lấy giỏ hàng của người dùng
POST	/api/cart/add	Thêm sản phẩm vào giỏ
PUT	/api/cart/update	Cập nhật số lượng sản phẩm
DELETE	/api/cart/remove/{id}	Xóa sản phẩm khỏi giỏ

### 3.5.6 API đánh giá sản phẩm

Nhóm API đánh giá cho phép người dùng gửi nhận xét và đánh giá sản phẩm sau khi mua. Các API này thao tác với bảng Review trong cơ sở dữ liệu, giúp người dùng có thể chia sẻ trải nghiệm về sản phẩm, đồng thời cung cấp thông tin phản hồi cho những người dùng khác.

*Bảng 3.16 API đánh giá sản phẩm*

Method	Endpoint	Mô tả
POST	/api/reviews	Thêm đánh giá sản phẩm
GET	/api/reviews/product/{id}	Lấy đánh giá theo sản phẩm
DELETE	/api/reviews/{id}	Xóa đánh giá

### 3.5.7 API hiển thị nội dung

Các API này phục vụ việc hiển thị banner quảng cáo và thông tin liên hệ trên website. Dữ liệu được lấy từ bảng Banner và Contact.

*Bảng 3.17 API hiển thị nội dung*

Method	Endpoint	Mô tả
GET	/api/banners	Lấy danh sách banner
POST	/api/banners	Thêm banner
GET	/api/contacts	Lấy thông tin liên hệ

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

### 4.1 Giao diện người dùng

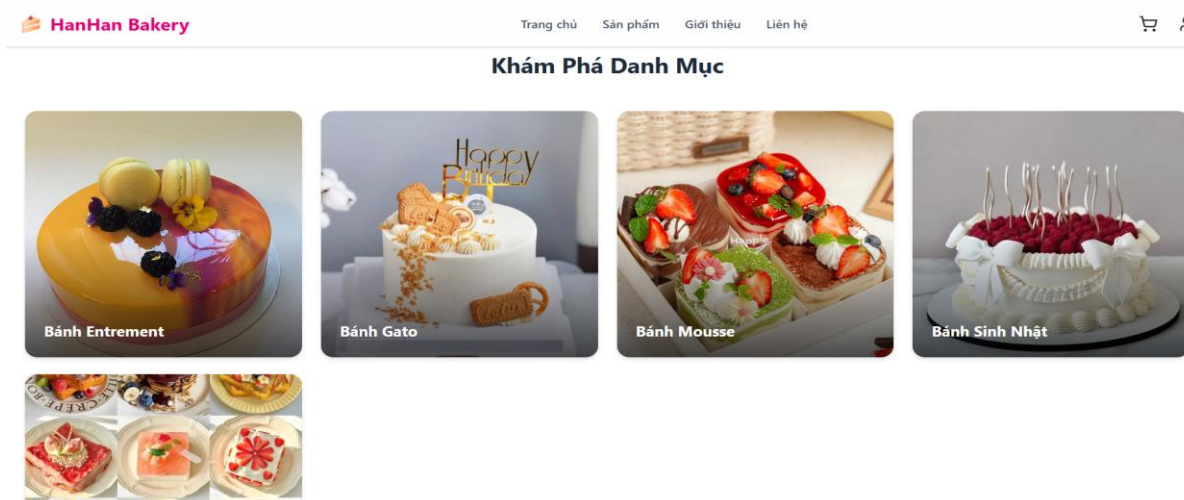
#### 4.1.1 Giao diện Trang chủ

Giao diện được thiết kế theo phong cách hiện đại với tông màu chủ đạo trắng - hồng, tập trung vào trải nghiệm thị giác thông qua banner quảng cáo lớn. Menu điều hướng chính nằm ở trung tâm gồm Trang chủ, Sản phẩm, Giới thiệu, Liên hệ và góc phải tích hợp các biểu tượng tiện ích quan trọng là Giỏ hàng và Tài khoản người dùng.



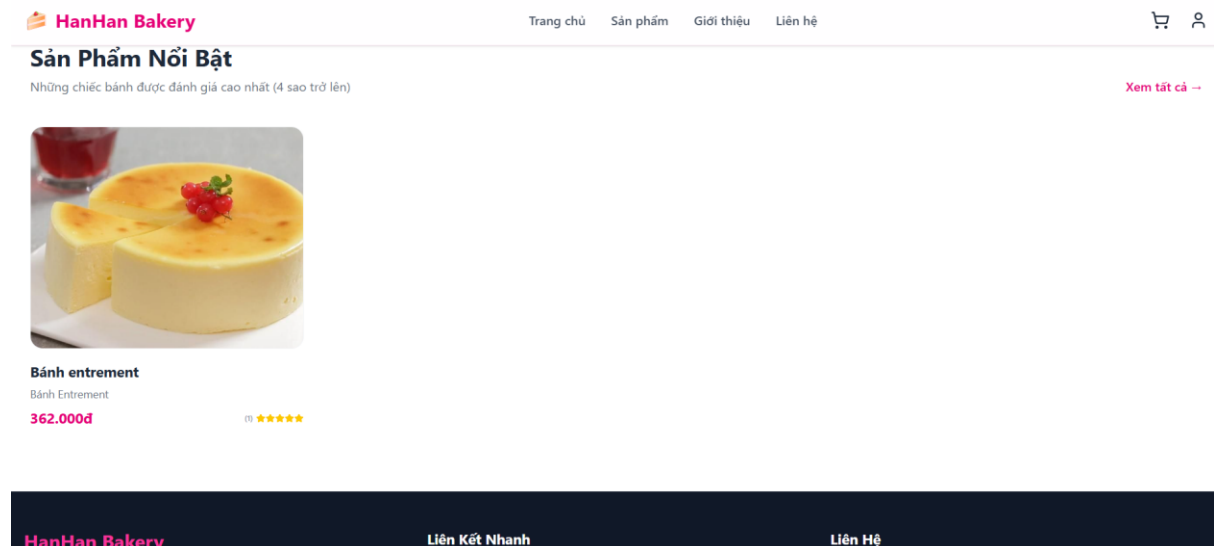
Hình 4.1 Phần trên của giao diện trang chủ

Nội dung chính được phân chia thành hai khu vực chức năng phần khám phá danh mục hiển thị trực quan các nhóm sản phẩm như Bánh Gato, Mousse, Bánh Sinh Nhật, ...giúp khách hàng khoanh vùng nhanh nhu cầu tìm kiếm.



Hình 4.2 Phần giữa của giao diện trang chủ

Mục "Sản Phẩm Nổi Bật" được thiết kế để tự động lọc và hiển thị những chiếc bánh nhận được phản hồi tích cực nhất từ cộng đồng đánh giá từ 4 sao trở lên, nhằm gia tăng độ tin cậy cho thương hiệu. Thông tin trên từng thẻ sản phẩm được trình bày đầy đủ và bao gồm hình ảnh minh họa, tên sản phẩm, giá bán niêm yết và thang điểm đánh giá sao, giúp người mua dễ dàng so sánh và tham khảo.



Hình 4.3 Phần cuối của giao diện trang chủ

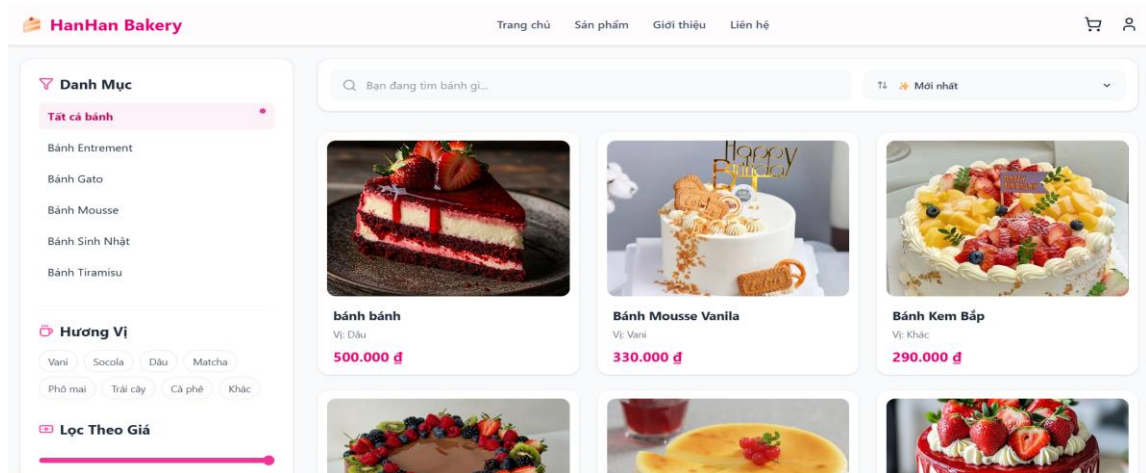
#### 4.1.2 Giao diện trang Sản phẩm

Giao diện trang "Sản Phẩm" được thiết kế với bố cục hai cột. Cột bên trái đóng vai trò là bộ lọc cho phép khách hàng khoanh vùng nhu cầu theo danh mục, lựa chọn theo hương vị yêu thích hoặc sử dụng thanh trượt để lọc giá phù hợp với ngân sách.

Khu vực hiển thị chính chiếm phần lớn không gian bên phải, được trang bị thanh công cụ tiện ích ở phía trên bao gồm ô tìm kiếm và bộ lọc sắp xếp thứ tự hiển thị giúp người dùng chủ động tiếp cận sản phẩm một cách nhanh chóng nhất.

Danh sách sản phẩm bên dưới được trình bày dưới dạng thẻ hiển thị đầy đủ các thông tin như hình ảnh minh họa, tên bánh, ghi chú hương vị và giá bán niêm yết.

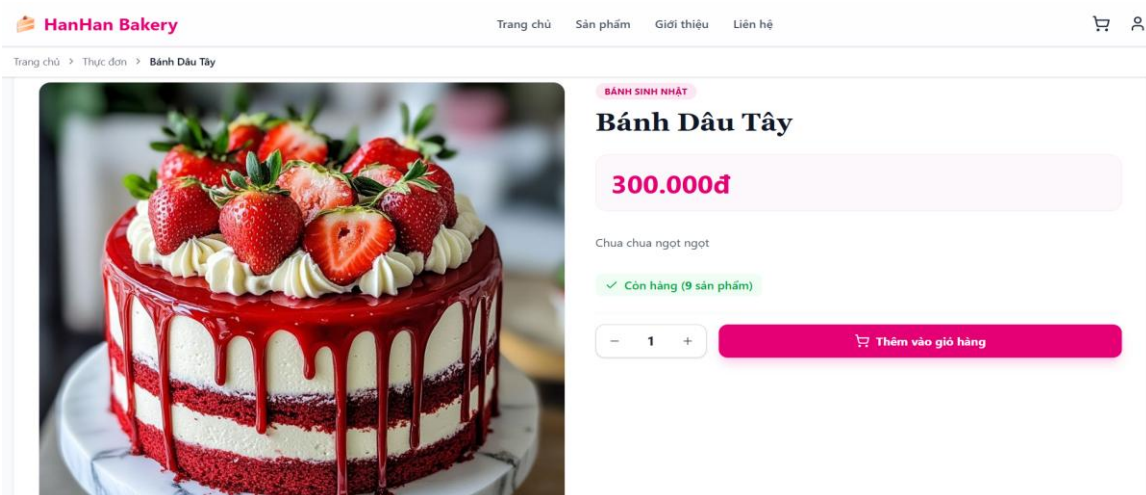




Hình 4.4 Giao diện trang sản phẩm

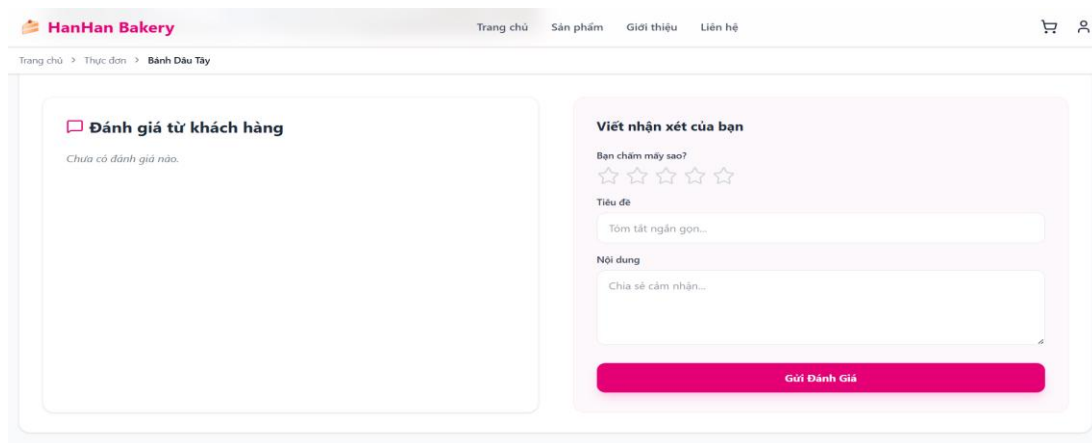
#### 4.1.3 Giao diện trang Chi tiết sản phẩm

Trang chi tiết sản phẩm được thiết kế trực quan với bố cục chia đôi, một bên hiển thị hình ảnh sản phẩm kích thước lớn, bên còn lại cung cấp các thông tin chi tiết như tên, giá bán, mô tả ngắn và tình trạng kho hàng. Các chức năng nghiệp vụ như chọn số lượng và nút "Thêm vào giỏ hàng" được bố trí nổi bật để thúc đẩy hành vi mua sắm. Đồng thời, thanh điều hướng ở đầu trang giúp người dùng dễ dàng định vị vị trí và quay lại các danh mục trước đó khi cần thiết.



Hình 4.5 Giao diện trang chi tiết sản phẩm

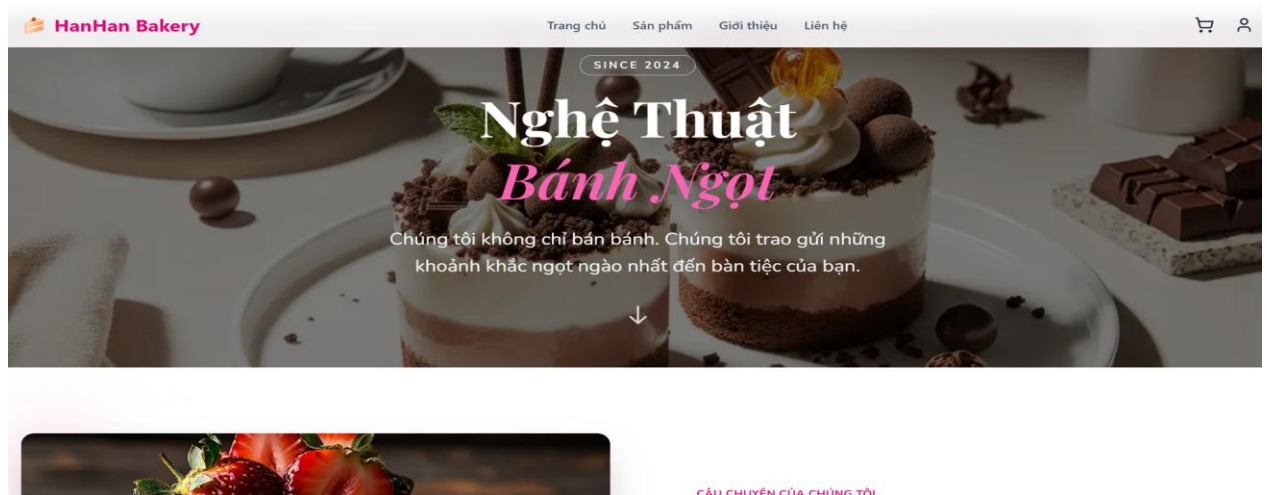
Khu vực tương tác được phân chia thành hai phần rõ rệt: bên trái hiển thị danh sách phản hồi từ khách hàng để người mua tham khảo, bên phải là biểu mẫu nhập liệu cho đánh giá mới. Form chức năng được thiết kế trực quan, cho phép người dùng chấm điểm theo thang sao, điền tiêu đề và nội dung góp ý, sau đó hoàn tất bằng nút "Gửi Đánh Giá" nổi bật.



Hình 4.6 Giao diện đánh giá, bình luận

#### 4.1.4 Giao diện trang Giới thiệu

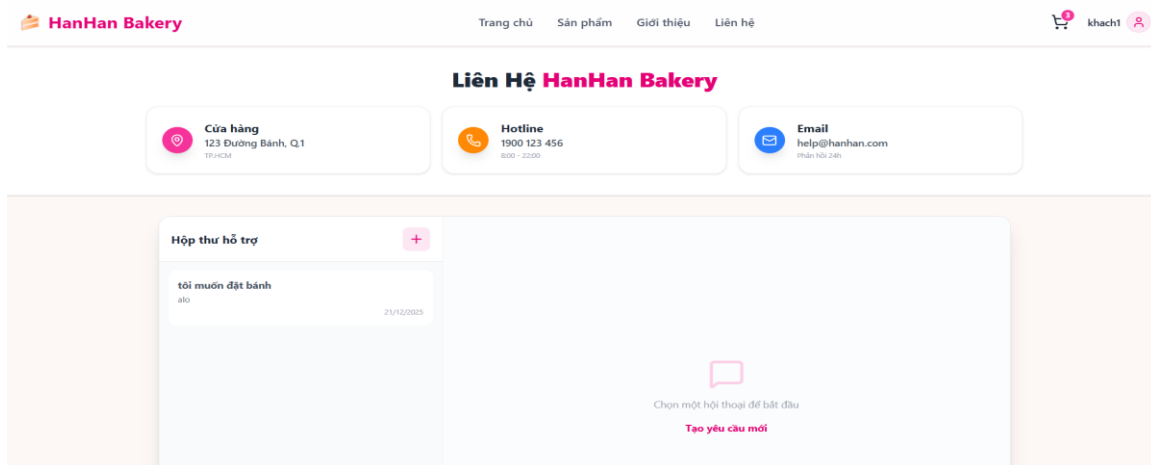
Trang Giới thiệu được thiết kế nhằm xây dựng hình ảnh thương hiệu chuyên nghiệp và gần gũi. Bố cục trang thường được chia thành các khối nội dung rõ ràng, tiếp đến là câu chuyện về lịch sử hình thành và sứ mệnh của cửa hàng. Ngoài ra, giao diện thường hiển thị thêm các phần cam kết về chất lượng nguyên liệu và quy trình.



Hình 4.7 Giao diện trang giới thiệu

#### 4.1.5 Giao diện trang Liên hệ

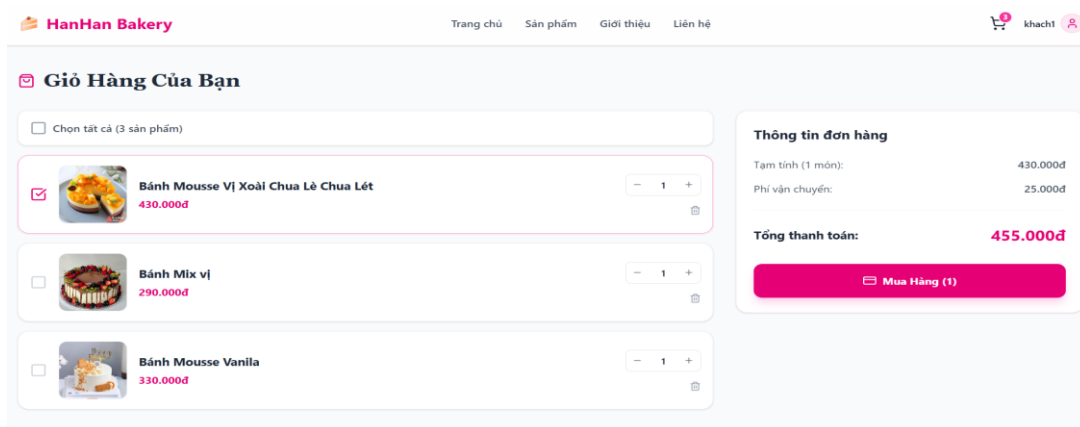
Giao diện trang Liên hệ được thiết kế trực quan với hai khu vực chức năng chính. Phần trên hiển thị thông tin liên lạc cố định bao gồm địa chỉ cửa hàng, hotline và email giúp khách hàng dễ dàng tiếp cận. Khu vực trung tâm là hệ thống "Hộp thư hỗ trợ" tích hợp, cho phép người dùng tạo yêu cầu mới hoặc theo dõi lịch sử trò chuyện trực tiếp với nhân viên tư vấn. Tính năng này giúp tối ưu hóa quy trình chăm sóc khách hàng, đảm bảo mọi thắc mắc được tiếp nhận và phản hồi nhanh chóng.



Hình 4.8 Giao diện trang liên hệ

#### 4.1.6 Giao diện trang Giỏ hàng

Giao diện Giỏ hàng được phân chia thành hai khu vực chính. Khu vực bên trái hiển thị danh sách chi tiết các sản phẩm người dùng đã chọn, tích hợp đầy đủ các công cụ quản lý như hộp kiểm checkbox để chọn sản phẩm muốn mua, điều chỉnh tăng/giảm số lượng và nút xóa để loại bỏ mặt hàng không cần tới. Khu vực bên phải là bảng thông tin đơn hàng, hệ thống sẽ tự động tính toán và cập nhật theo thời gian thực tổng số tiền thanh toán bao gồm tạm tính và phí vận chuyển dựa trên các sản phẩm được tích chọn, giúp người dùng nắm rõ chi phí trước khi nhấn nút "Mua Hàng" để tiến hành thanh toán.



Hình 4.9 Giao diện giỏ hàng

#### 4.1.7 Giao diện trang Thanh toán

Giao diện Thanh toán được thiết kế với cột bên trái là khu vực nhập liệu thông tin giao hàng bắt buộc như họ tên, số điện thoại, địa chỉ nhận hàng và mục lựa chọn phương thức thanh toán. Cột bên phải là bảng tóm tắt đơn hàng cuối cùng, hiển thị lại

thông tin sản phẩm, số lượng và tổng chi phí giúp người dùng rà soát lại lần cuối trước khi nhấn nút xác nhận để hoàn tất giao dịch.

Hình 4.10 Giao diện trang thanh toán

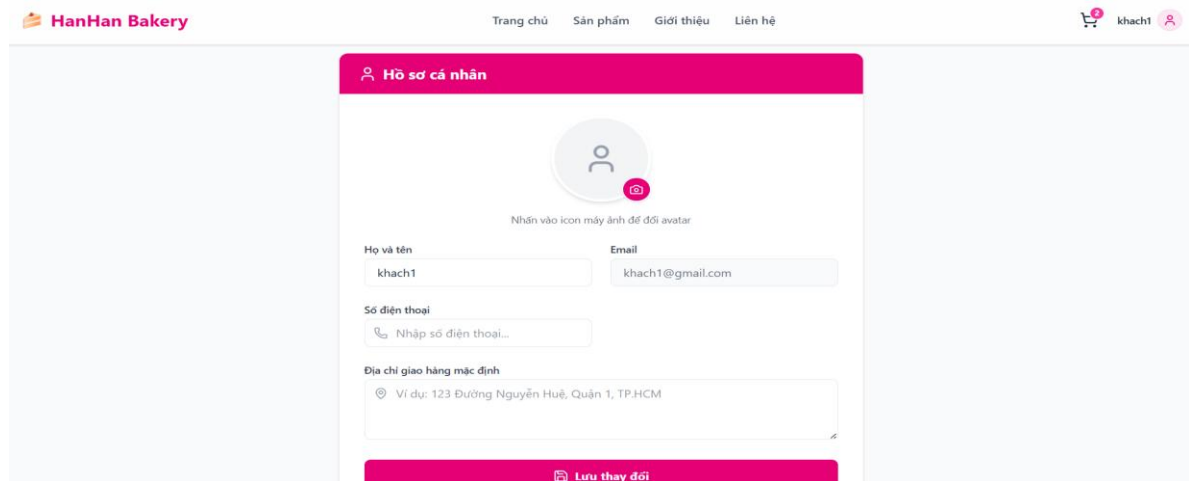
#### 4.1.8 Giao diện trang Chi tiết đơn hàng

Phần tiêu đề hiển thị mã đơn hàng, ngày giờ đặt và trạng thái xử lý hiện tại kèm nút chức năng "Hủy đơn hàng" cho phép người dùng thao tác nếu đơn chưa được xác nhận. Khu vực nội dung bên dưới gồm thông tin giao nhận địa chỉ, người nhận, hình thức thanh toán ở cột trái, trong khi cột phải liệt kê chi tiết sản phẩm đã đặt và bảng tổng kết chi phí cuối cùng, giúp người dùng dễ dàng đối chiếu và theo dõi tiến trình đơn hàng của mình.

Hình 4.11 Giao diện trang chi tiết đơn hàng

#### 4.1.9 Giao diện trang Quản lý hồ sơ cá nhân

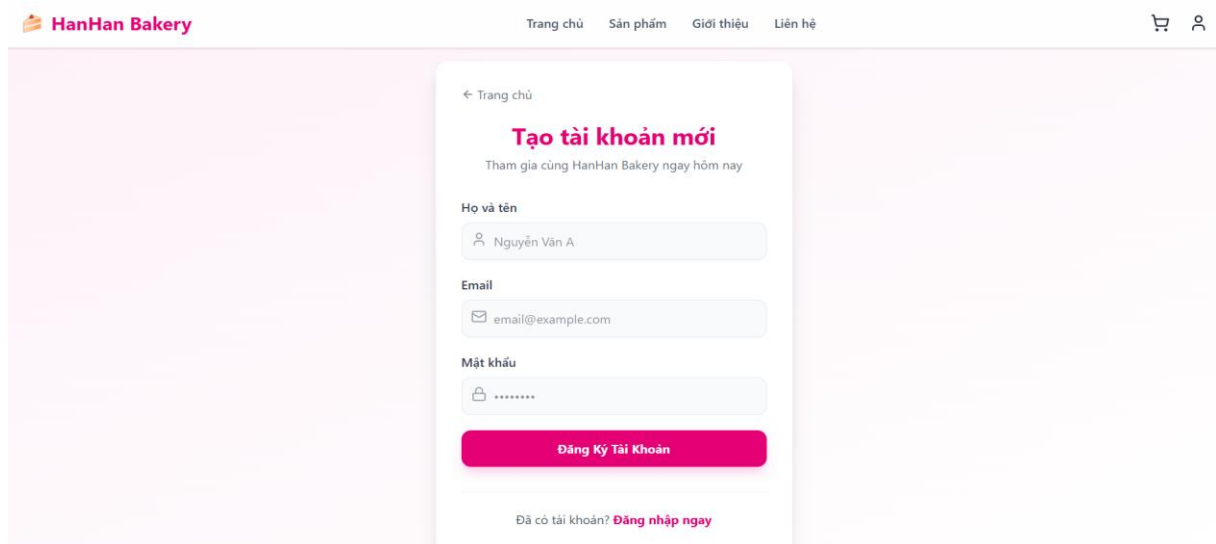
Người dùng có thể dễ dàng chỉnh sửa họ tên, số điện thoại, địa chỉ giao hàng và thay đổi ảnh avatar thông qua trang chỉnh sửa hồ sơ cá nhân.



Hình 4.12 Giao diện trang quản lý hồ sơ cá nhân

#### 4.1.10 Giao diện trang Đăng ký

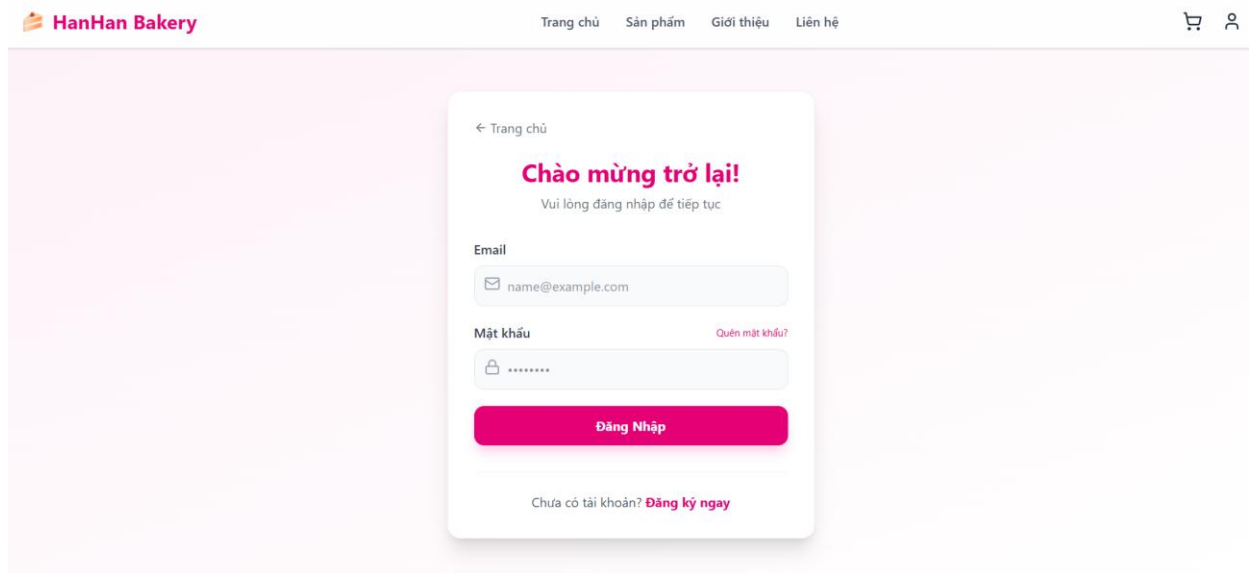
Giao diện trang Đăng ký được thiết kế đơn giản với bố cục rõ ràng, gồm các trường nhập họ tên, email và mật khẩu. cho phép người dùng tạo tài khoản mới.



Hình 4.13 Giao diện trang đăng ký

#### 4.1.11 Giao diện trang Đăng nhập

Giao diện trang Đăng nhập được thiết kế hiện đại, tối giản với bố cục tập trung vào form đăng nhập, sử dụng màu sắc nhẹ nhàng giúp người dùng dễ thao tác. Các trường nhập email và mật khẩu được hiển thị rõ ràng kèm thông báo lỗi trực quan khi đăng nhập không thành công.

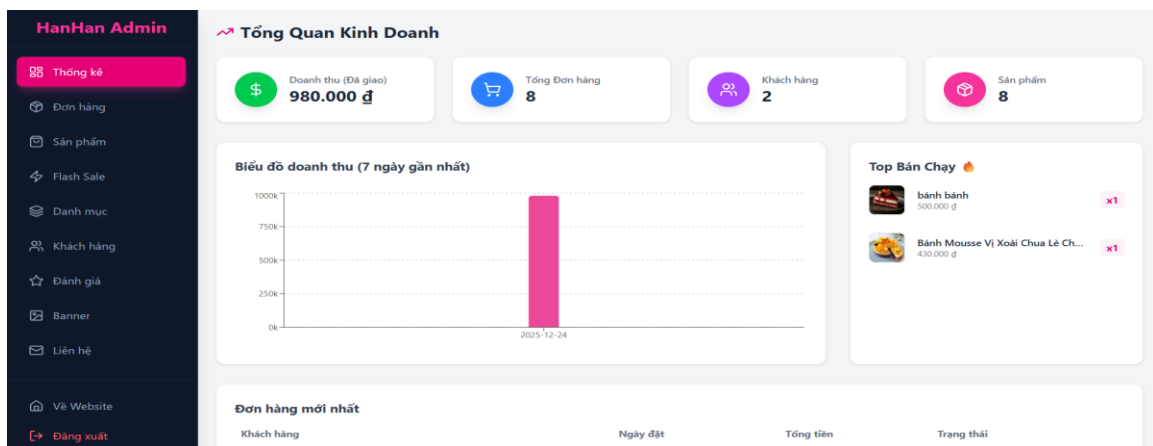


Hình 4.14 Giao diện trang đăng nhập

## 4.2 Giao diện Quản trị viên

### 4.2.1 Giao diện trang Thống kê

Giao diện trang Dashboard được thiết kế trực quan, hiển thị tổng quan tình hình kinh doanh thông qua các thẻ thống kê và biểu đồ. Các thông tin quan trọng như doanh thu, số đơn hàng, số khách hàng và số sản phẩm được trình bày rõ ràng bằng các khối số liệu nổi bật, giúp quản trị viên dễ dàng nắm bắt tình trạng hệ thống. Ngoài ra, giao diện còn hiển thị biểu đồ doanh thu theo thời gian, danh sách sản phẩm bán chạy và các đơn hàng mới nhất.



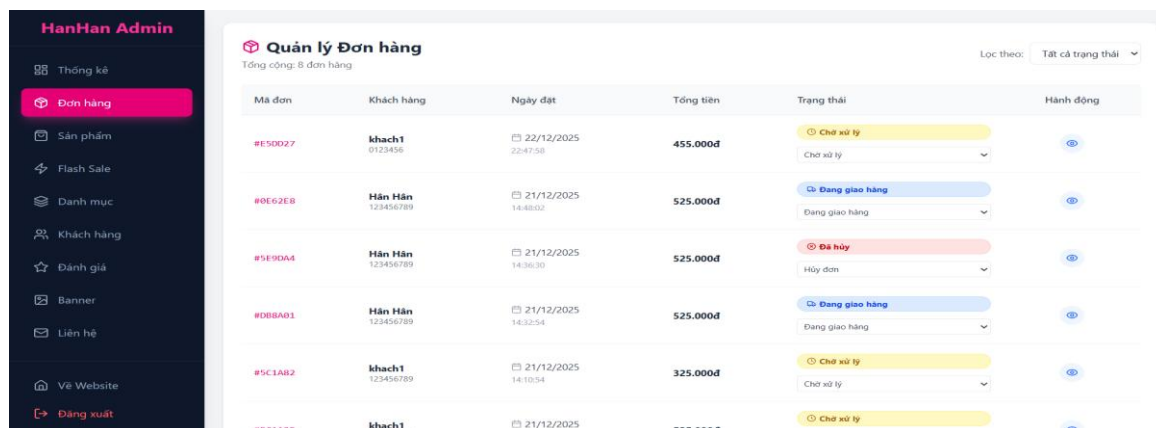
Hình 4.15 Giao diện trang thống kê

### 4.2.2 Giao diện trang Quản lý đơn hàng

Giao diện trang Quản lý đơn hàng được thiết kế rõ ràng, hiển thị danh sách các đơn hàng dưới dạng bảng với đầy đủ thông tin như mã đơn, khách hàng, ngày đặt, tổng



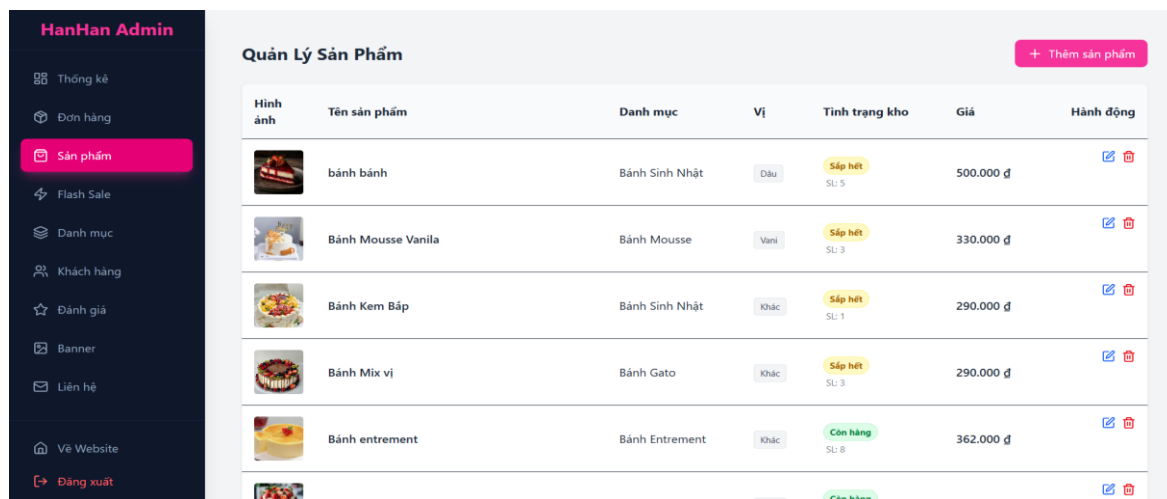
tiền và trạng thái. Hệ thống sử dụng màu sắc và nhãn trạng thái trực quan giúp quản trị viên dễ dàng phân biệt tình trạng xử lý của từng đơn hàng. Ngoài ra, giao diện hỗ trợ lọc đơn hàng theo trạng thái nhằm tăng hiệu quả quản lý.



Hình 4.16 Giao diện trang quản lý đơn hàng

#### 4.2.3 Giao diện trang Quản lý sản phẩm

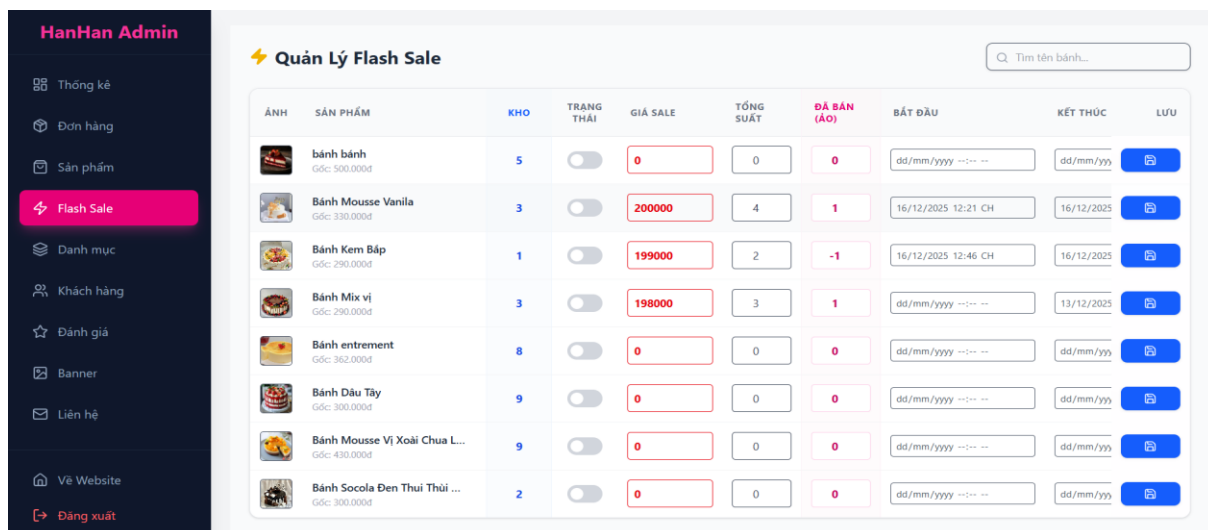
Giao diện trang Quản lý sản phẩm được thiết kế dạng bảng, hiển thị đầy đủ thông tin của từng sản phẩm bao gồm hình ảnh, tên sản phẩm, danh mục, hương vị, tình trạng kho và giá bán. Trạng thái tồn kho được biểu diễn trực quan bằng màu sắc (còn hàng, sắp hết, hết hàng), giúp quản trị viên dễ dàng theo dõi số lượng sản phẩm.



Hình 4.17 Giao diện trang quản lý sản phẩm

#### 4.2.4 Giao diện trang Quản lý Flash Sale

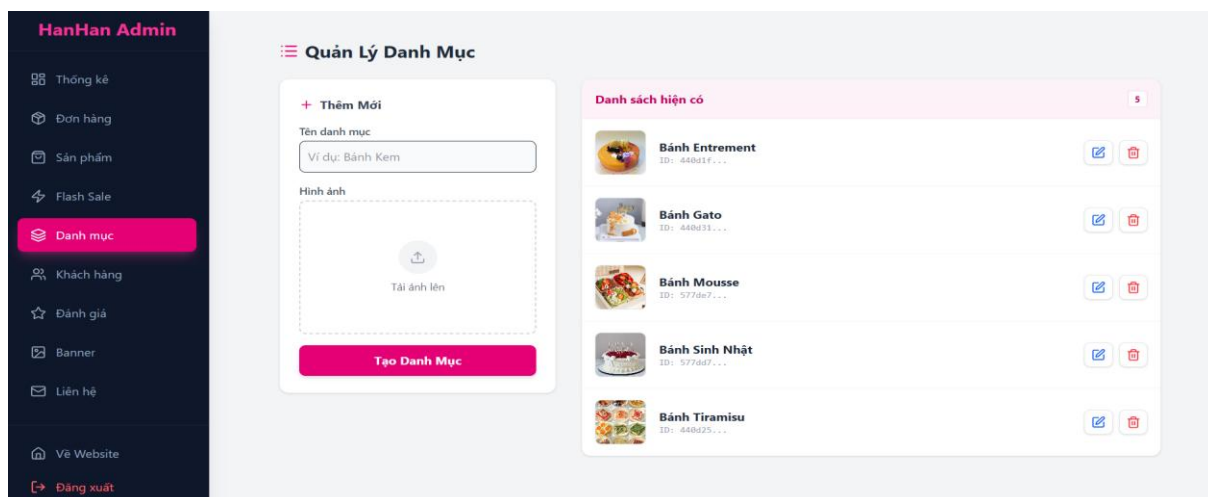
Giao diện trang Quản lý Flash Sale cho phép quản trị viên theo dõi và cấu hình chương trình Flash Sale cho từng sản phẩm thông qua bảng dữ liệu trực quan. Quản trị viên có thể tìm kiếm sản phẩm, bật/tắt Flash Sale, thiết lập giá sale, số lượng áp dụng và thời gian bắt đầu và kết thúc.



Hình 4.18 Giao diện trang quản lý Flash Sale

#### 4.2.5 Giao diện trang Quản lý danh mục

Giao diện trang Quản lý danh mục cho phép quản trị viên thêm mới, chỉnh sửa và xóa các danh mục sản phẩm một cách trực quan. Người dùng có thể nhập tên danh mục, tải lên hình ảnh minh họa và xem trước trước khi lưu. Danh sách danh mục được hiển thị rõ ràng kèm hình ảnh, hỗ trợ thao tác chỉnh sửa và xóa nhanh chóng, giúp việc quản lý dữ liệu hiệu quả hơn.

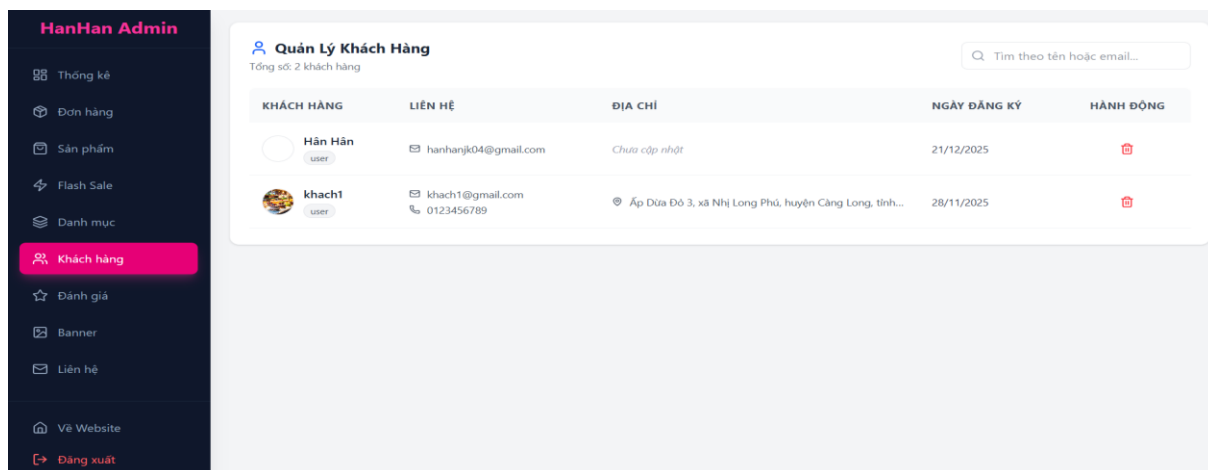


Hình 4.19 Giao diện trang quản lý danh mục

#### 4.2.6 Giao diện trang Quản lý khách hàng

Giao diện trang Quản lý khách hàng cho phép quản trị viên theo dõi danh sách người dùng đã đăng ký trên hệ thống. Thông tin khách hàng được hiển thị đầy đủ gồm họ tên, email, số điện thoại, địa chỉ, vai trò và ngày đăng ký. Hệ thống hỗ trợ tìm kiếm nhanh theo tên hoặc email và cho phép xóa khách hàng khi cần thiết.

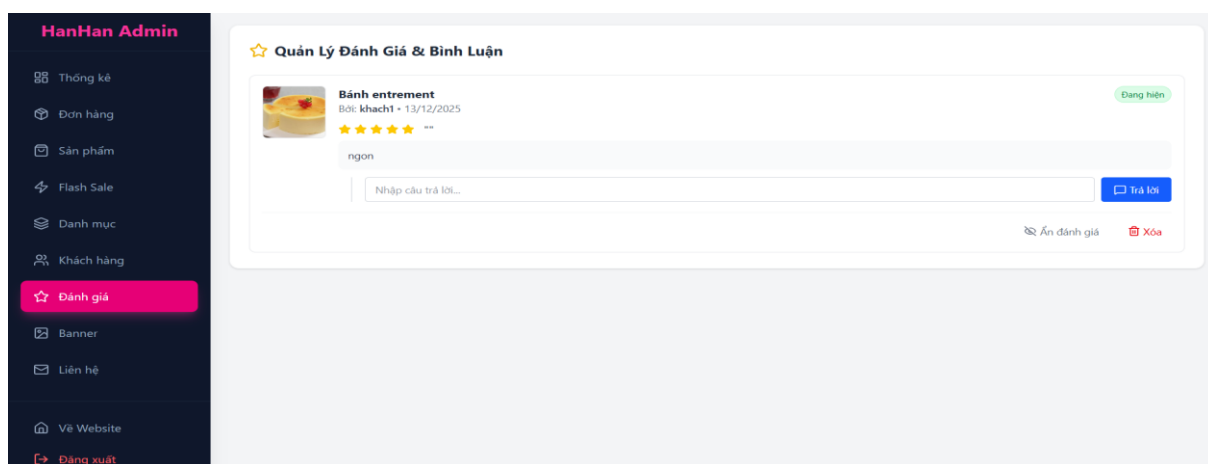




Hình 4.20 Giao diện trang quản lý khách hàng

#### 4.2.7 Giao diện trang Quản lý đánh giá, bình luận

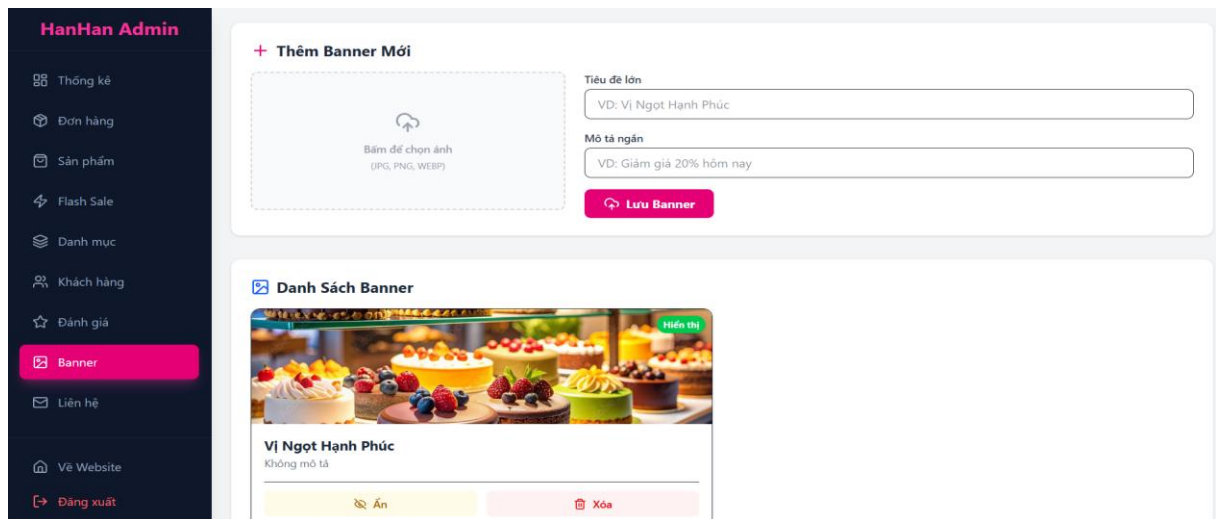
Giao diện trang Quản lý đánh giá cho phép quản trị viên theo dõi và kiểm soát các đánh giá, bình luận của khách hàng đối với sản phẩm. Hệ thống hỗ trợ duyệt hiển thị hoặc ẩn đánh giá, xóa đánh giá không phù hợp và phản hồi trực tiếp cho khách hàng. Thông tin đánh giá được trình bày rõ ràng kèm hình ảnh sản phẩm, số sao và nội dung nhận xét.



Hình 4.21 Giao diện trang Quản lý đánh giá, bình luận

#### 4.2.8 Giao diện trang Quản lý Banner

Giao diện Quản lý Banner cho phép quản trị viên thêm mới, hiển thị và quản lý các banner quảng cáo trên trang chủ. Hệ thống hỗ trợ tải ảnh trực tiếp từ máy tính, xem trước hình ảnh trước khi lưu, bật/tắt trạng thái hiển thị banner và xóa banner không còn sử dụng. Các banner được trình bày trực quan kèm tiêu đề, mô tả và trạng thái



Hình 4.22 Giao diện trang quản lý banner

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận

#### 5.1.1 Kết quả đạt được

Sau quá trình nghiên cứu tìm hiểu và thực hiện đề tài "Xây dựng hệ thống website bán bánh kem trực tuyến sử dụng ReactJS và Tailwind CSS", em đã hoàn thành được mục tiêu ban đầu đề ra. Về mặt công nghệ, đã nắm bắt và ứng dụng được thư viện ReactJS kết hợp với TailwindCSS để tạo ra một trang web có giao diện đẹp, hiện đại và hoạt động mượt mà.

Tích hợp cơ sở dữ liệu, sử dụng MongoDB để lưu trữ thông tin người dùng, đảm bảo dữ liệu được tổ chức một cách khoa học và hiệu quả.

Sản phẩm cuối cùng là một website hoàn chỉnh, giúp giải quyết việc bán hàng thủ công trước đây. Giờ đây, khách hàng có thể dễ dàng vào web xem ảnh bánh, chọn bánh bỏ vào giỏ và đặt hàng ngay mà không cần phải đến trực tiếp cửa hàng. Về phía quản trị viên, hệ thống cũng đã có trang quản lý riêng để có thể quản lý cửa hàng một cách dễ dàng. Nhìn chung, website đã chạy ổn định và đạt các chức năng cơ bản ban đầu đề ra, sẵn sàng để sử dụng.

#### 5.1.2 Hạn chế

Hệ thống chưa có các chức năng nâng cao như gợi ý sản phẩm theo hành vi người dùng hoặc chatbot hỗ trợ tự động.

Giao diện tuy đã đáp ứng được nhu cầu cơ bản nhưng vẫn còn đơn giản, chưa tối ưu hoàn toàn cho mọi kích thước thiết bị.

### 5.2 Hướng phát triển

Mặc dù hệ thống đã đáp ứng tốt các yêu cầu cơ bản, nhưng trong quá trình nghiên cứu nhận thấy vẫn còn vấn đề để nâng cấp và hoàn thiện sản phẩm nhằm bắt kịp xu hướng công nghệ hiện đại.

Ưu tiên trong thời gian tới là tích hợp các cổng thanh toán trực tuyến phổ biến như MoMo, ZaloPay hoặc VNPAY, giúp đa dạng hóa phương thức thanh toán và mang lại sự tiện lợi tối đa cho khách hàng thay vì chỉ phụ thuộc vào hình thức thanh toán khi nhận hàng.

Bên cạnh đó, việc ứng dụng các thuật toán gợi ý sản phẩm dựa trên lịch sử mua sắm hoặc tích hợp Chatbot hỗ trợ tự động cũng là những tính năng cần thiết để cá nhân hóa trải nghiệm người dùng.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Viblo, “ReactJS là gì? Tổng quan về ReactJS,”. Truy cập tại: <https://viblo.asia/p/reactjs-tim-hieu-thong-qua-vi-du-gDVK2Oe2ZLj>. Ngày truy cập: [12/11/2025].
- [2] Viblo, “NodeJS là gì? Tổng quan về NodeJS,”. Truy cập tại: <https://viblo.asia/p/nodejs-la-gi-tong-quan-ve-nodejs-3Q75w4xq5Wb>. Ngày truy cập: [18/11/2025].
- [3] Viblo, “Tailwind CSS là gì? Giới thiệu và cách sử dụng Tailwind CSS,”. Truy cập tại: <https://viblo.asia/p/tailwind-css-la-gi-gioi-thieu-va-cach-su-dung-4dbZN8R4lYM>. Ngày truy cập: [10/12/2025].
- [4] Nguyễn Văn Tuấn, Lập trình Node.js cơ bản và nâng cao, Nhà xuất bản Thanh Niên, Hà Nội, 2021.
- [5] Viblo, “MongoDB là gì? Tổng quan về MongoDB,”. Truy cập tại: <https://viblo.asia/p/mongodb-la-gi-tong-quan-ve-mongodb-gAm5yD8a5db>. Ngày truy cập: [21/12/2025].