BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**CHUYÊN ĐỀ TỐT NGHIỆP**

**XÂY DỰNG WEBSITE ĐẶT ĐỒ ĂN NHANH**

**SỬ DỤNG WEBFORMS ASP.NET**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn:** | **ThS. Đoàn Vũ Thịnh** |
| **Sinh viên thực hiện:** | **Huỳnh Gia Bảo** |
| **Mã số sinh viên:** | **63130087** |

Khánh Hoà – 2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**CHUYÊN ĐỀ TỐT NGHIỆP**

**XÂY DỰNG WEBSITE ĐẶT ĐỒ ĂN NHANH**

**SỬ DỤNG WEBFORMS ASP.NET**

|  |  |
| --- | --- |
| GVHD: | ThS. Đoàn Vũ Thịnh |
| SVTH: | Huỳnh Gia Bảo |
| MSSV: | 63130087 |

Khánh Hoà – Tháng 6/2025

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa/Viện: Công Nghệ Thông Tin**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ CHUYÊN ĐỀ TỐT NGHIỆP  
*(Dùng cho CBHD và nộp cùng báo cáo CĐTN của sinh viên)***

Tên đề tài: Xây dựng website đặt đồ ăn nhanh sử dụng WebForms ASP.NET

Chuyên ngành: Công nghệ phần mềm

Họ và tên sinh viên: Huỳnh Gia Bảo Mã sinh viên: 63130087

Người hướng dẫn (học hàm, học vị, họ và tên) :

Cơ quan công tác:

**Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí**  **đánh giá** | **Trọng số**  **(%)** | **Mô tả mức chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| Xây dựng đề cương nghiên cứu | 10 |  |  |  |  |  |
| Tinh thần và thái độ làm việc | 10 |  |  |  |  |  |
| Kiến thức và kỹ năng làm việc | 10 |  |  |  |  |  |
| Nội dung và kết quả đạt được | 40 |  |  |  |  |  |
| Kỹ năng viết và trình bày báo cáo | 30 |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*\*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung** (sau khi sinh viên hoàn thành CĐTN):

…………………………………………………………………………………………….…

……………………………………………………………………………………….………

……………………………………………………………………………………………….

…………………………………………………………………………………………….…

……………………………………………………………………………………….………

……………………………………………………………………………………………….

Đồng ý cho sinh viên: Được chấm phản biện:  Không được chấm phản biện:

*Khánh Hòa, ngày……. tháng……. năm………*

**Cán bộ hướng dẫn**

*(Ký và ghi rõ họ tên)*

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa/Viện: Công Nghệ Thông Tin**

**PHIẾU CHẤM ĐIỂM CHUYÊN ĐỀ/ KHÓA LUẬN TỐT NGHIỆP**

***(Dành cho cán bộ chấm điểm)***

Tên đề tài: Xây dựng Website đặt đồ ăn nhanh sử dụng WebForms ASP.NET

Chuyên ngành: Công nghệ phần mềm

Họ và tên sinh viên: Huỳnh Gia Bảo

Mã sinh viên: 63130087

Người phản biện (học hàm, học vị, họ và tên):

Cơ quan công tác: Khoa CNTT

***Phần đánh giá và cho điểm của người phản biện (tính theo thang điểm 10)***

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí**  **đánh giá** | **Trọng số (%)** | **Mô tả mức chất lượng** | | | | **Điểm** | |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |  |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| Hình thức bản  thuyết minh | 30 |  |  |  |  |  |
| Nội dung bản  thuyết minh | 30 |  |  |  |  |  |
| Kết quả  nguyên cứu | 20 |  |  |  |  |  |
| Mức độ trích dẫn và  sao chép | 20 |  |  |  |  |  |
| **ĐIỂM TỔNG:** | | | | | |  | |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

*Khánh Hòa, ngày…….tháng………năm………..*

**Cán bộ chấm điểm CĐ**

(Ký và ghi rõ họ tên)

# LỜI CAM ĐOAN

Em xin cam đoan mọi kết quả của đề tài “Xây dựng Website đặt đồ ăn nhanh FastBite sử dụng WebForms ASP.NET” là công trình nghiên cứu của cá nhân em và được sự hướng dẫn khoa học của ThS. Đoàn Vũ Thịnh. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kì hình thức nào.

Mọi kết quả trong đề tài đều dựa trên những kết quả và sự cố gắng của em từ những kiến thức mà em đã học trong thời gian học tập và thực hành tại trường.

Khánh Hòa, ngày 15 tháng 05 năm 2025

Tác giả chuyên đề

*(ký và ghi rõ họ tên)*

# LỜI CẢM ƠN

Trong suốt thời gian thực hiện đề tài, em xin cảm ơn đến thầy Đoàn Vũ Thịnh đã nhiệt tình hướng dẫn và giúp đỡ em hoàn thành đề tài này.

Em xin chân thành cảm ơn ban giám hiệu, văn phòng khoa Công nghệ thông tin trường Đại Học Nha Trang và các thầy cô bộ môn đã tận tình giảng dạy và trang bị những kiến thức cần thiết và đã tạo điều kiện tốt nhất cho chúng em trong suốt quá trình học tập tại trường.

Mặc dù đã rất cố gắng, nhưng trong một khoảng thời gian cho phép, cũng như hạn chế về mặt kiến thức của em, chuyên đề này không thể tránh khỏi nhiều thiếu sót. Chính vì vậy, em rất mong nhận được sự góp ý của các thầy cô để giúp em cải thiện kỹ năng tốt hơn sau này.

Khánh Hòa, ngày 15 tháng 05 năm 2025

Sinh viên thực hiện

# MỤC LỤC

[LỜI CAM ĐOAN 5](#_Toc199089913)

[LỜI CẢM ƠN 6](#_Toc199089914)

[MỤC LỤC 7](#_Toc199089915)

[DANH MỤC HÌNH 10](#_Toc199089916)

[DANH MỤC BẢNG 12](#_Toc199089917)

[ĐỊNH NGHĨA TỪ VIẾT TẮT 13](#_Toc199089918)

[LỜI MỞ ĐẦU 14](#_Toc199089919)

[CHƯƠNG 1: TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT 15](#_Toc199089920)

[1.1 Xác định vấn đề trong việc đặt đồ ăn 15](#_Toc199089921)

[1.2 ASP.NET WebForms là gì 15](#_Toc199089922)

[1.2.1 So sánh WebForms với các famework khác 17](#_Toc199089923)

[1.3 Lý do chọn WebForms thay vì MVC 19](#_Toc199089924)

[1.4 Các công cụ liên quan 19](#_Toc199089925)

[CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 21](#_Toc199089926)

[2.1 Phân tích đặc tả và thiết kế hệ thống 21](#_Toc199089927)

[2.2 Xác định yêu cầu và lập kế hoạch 22](#_Toc199089928)

[2.2.1 Yêu cầu chức năng 22](#_Toc199089929)

[2.2.2 Yêu cầu phi chức năng 23](#_Toc199089930)

[2.2.3 Lập kế hoạch triển khai 23](#_Toc199089931)

[2.3 Các sơ đồ mô tả hệ thống 24](#_Toc199089932)

[2.3.1 Sơ đồ phân cấp chức năng (BFD) 24](#_Toc199089933)

[2.3.2 Sơ đồ luồng dữ liệu mức bối cảnh 25](#_Toc199089934)

[2.3.3 Sơ đồ luồng dữ liệu mức đỉnh (DFD) 26](#_Toc199089935)

[2.3.4 Sơ đồ luồng dữ liệu – Quản lý đơn hàng 27](#_Toc199089936)

[2.3.5 Sơ đồ luồng dữ liệu – Quản lý tài khoản 28](#_Toc199089937)

[2.3.6 Sơ đồ luồng dữ liệu – Quản lý người dùng 29](#_Toc199089938)

[CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG 30](#_Toc199089939)

[3.1 Triển khai cơ sở dữ liệu 30](#_Toc199089940)

[3.1.1 Tạo cơ sở dữ liệu và các bảng 30](#_Toc199089941)

[3.1.2 Cơ sở dữ liệu của các bảng 30](#_Toc199089942)

[3.1.3 Mối quan hệ giữa các bảng 34](#_Toc199089943)

[3.2 Tạo Stored Procedure 36](#_Toc199089944)

[CHƯƠNG 4: CÀI ĐẶT GIAO DIỆN HỆ THỐNG 39](#_Toc199089945)

[4.1 Tổng quan về giao diện hệ thống FastBite 39](#_Toc199089946)

[4.2 Giao diện quản lý (Admin) 39](#_Toc199089947)

[4.2.1 Giao diện Dashboard 39](#_Toc199089948)

[4.2.2 Giao diện quản lý Danh Mục Sản Phẩm 40](#_Toc199089949)

[4.2.3 Giao diện quản lý Sản Phẩm 41](#_Toc199089950)

[4.2.4 Giao diện quản lý trạng thái đơn hàng 41](#_Toc199089951)

[4.2.5 Giao diện quản lý Tài khoản 42](#_Toc199089952)

[4.2.6 Giao diện quản lý Phản hồi 43](#_Toc199089953)

[4.2.7 Giao diện Gửi email 43](#_Toc199089954)

[4.2.8 Giao diện Bài viết 44](#_Toc199089955)

[4.2.9 Giao diện Báo cáo Doanh thu 45](#_Toc199089956)

[4.3 Giao diện Khách hàng 45](#_Toc199089957)

[4.3.1 Giao diện Đăng nhập và Đăng ký 46](#_Toc199089958)

[4.3.2 Giao diện tài khoản 49](#_Toc199089959)

[4.3.3 Giao diện Menu sản phẩm 51](#_Toc199089960)

[4.3.4 Giao diện Liên hệ 52](#_Toc199089961)

[4.3.5 Giao diện Tin Tức 53](#_Toc199089962)

[4.3.6 Giao diện Giỏ hàng 54](#_Toc199089963)

[4.3.7 Giao diện thanh toán 56](#_Toc199089964)

[CHƯƠNG 5 ĐÁNH GIÁ VÀ HƯỚNG GIẢI QUYẾT 60](#_Toc199089965)

[5.1 Đánh giá kết quả đạt được 60](#_Toc199089966)

[5.2 Hạn chế và vấn đề cần khắc phục 60](#_Toc199089967)

[5.3 Định hướng phát triển 61](#_Toc199089968)

[KẾT LUẬN 62](#_Toc199089969)

[TÀI LIỆU THAM KHẢO 63](#_Toc199089970)

# DANH MỤC HÌNH

[Hình 1.1 Logo ASP.NET WebForms 16](#_Toc198805075)

[Hình 2.1 Logo Website FastBite 21](#_Toc198805079)

[Hình 2.2 Sơ đồ phân cấp chức năng 25](#_Toc198805080)

[Hình 2.3 Sơ đồ luồng dữ liệu (mức bối cảnh) 26](#_Toc198805081)

[Hình 2.4 Sơ đồ luồng dữ liệu mức đỉnh (DFD) 27](#_Toc198805082)

[Hình 2.5 Sơ đồ luồng dữ liệu – Quản lý đơn hàng 28](#_Toc198805083)

[Hình 2.6 Sơ đồ luồng dữ liệu – Quản lý tài khoản 28](#_Toc198805084)

[Hình 2.7 Sơ đồ luồng dữ liệu – Quản lý người dùng 29](#_Toc198805085)

[Hình 3.1 Sơ đồ quan hệ của cơ sở dữ liệu 30](#_Toc198805086)

[Hình 3.2 Mối quan hệ bảng Products và bảng Categories 34](#_Toc198805087)

[Hình 3.3 Mối quan hệ bảng Carts với bảng Products và bảng Users 35](#_Toc198805088)

[Hình 3.4 Mối quan hệ giữa bảng Orders, Products, Users và Payment 35](#_Toc198805089)

[Hình 4.1 Giao diện trang chủ Admin 40](#_Toc198805090)

[Hình 4.2 Giao diện quản lý Danh Mục Sản Phẩm 41](#_Toc198805091)

[Hình 4.3 Giao diện quản lý Sản Phẩm 42](#_Toc198805092)

[Hình 4.4 Giao diện quản lý trạng thái đơn hàng 42](#_Toc198805093)

[Hình 4.5 Giao diện quản lý Tài khoản 43](#_Toc198805094)

[Hình 4.6 Giao diện quản lý Phản hồi 44](#_Toc198805095)

[Hình 4.7 Giao diện Gửi email 44](#_Toc198805096)

[Hình 4.8 Giao diện Bài Viết 45](#_Toc198805097)

[Hình 4.9 Giao diện Báo cáo Doanh thu 45](#_Toc198805098)

[Hình 4.10 Biểu đồ đường biểu diễn tổng doanh thu 46](#_Toc198805099)

[Hình 4.11 Giao diện trang chủ FastBite 47](#_Toc198805100)

[Hình 4.12 Giao diện đăng nhập 47](#_Toc198805101)

[Hình 4.13 Giao diện đăng ký 48](#_Toc198805102)

[Hình 4.14 Giao diện Quên mật khẩu 48](#_Toc198805103)

[Hình 4.15 Giao diện nhập mã OTP 49](#_Toc198805104)

[Hình 4.16 Email gửi mã OTP 49](#_Toc198805105)

[Hình 4.17 Giao diện đặt mật khẩu mới 50](#_Toc198805106)

[Hình 4.18 Giao diện thông tin tài khoản 50](#_Toc198805107)

[Hình 4.19 Giao diện lịch sử mua hàng 51](#_Toc198805108)

[Hình 4.20 Giao diện Menu sản phẩm 52](#_Toc198805109)

[Hình 4.21 Giao diện Liên hệ 53](#_Toc198805110)

[Hình 4.22 Giao diện Tin Tức 54](#_Toc198805111)

[Hình 4.23 Giao diện giỏ hàng 55](#_Toc198805112)

[Hình 4.24 Giao diện thanh toán 57](#_Toc198805113)

[Hình 4.25 Giao diện thanh toán mã QR 57](#_Toc198805114)

[Hình 4.26 Giao diện Hóa đơn 58](#_Toc198805115)

[Hình 4.27 File Hóa đơn mua hàng 58](#_Toc198805116)

# DANH MỤC BẢNG

[Bảng 3.1.1 Bảng User (Người dùng) 30](#_Toc198805117)

[Bảng 3.1.2 Bảng Categories (Danh mục) 31](#_Toc198805118)

[Bảng 3.1.3 Bảng Contact (Liên hệ) 31](#_Toc198805119)

[Bảng 3.1.4 Bảng Product (Sản Phẩm) 32](#_Toc198805120)

[Bảng 3.1.5 Bảng Carts (Giỏ hàng) 32](#_Toc198805121)

[Bảng 3.1.6 Bảng Orders (Đơn hàng) 33](#_Toc198805122)

[Bảng 3.1.7 Bảng Payment (Thanh toán) 33](#_Toc198805123)

[Bảng 3.1.8 Bảng News (Tin tức) 34](#_Toc198805124)

# ĐỊNH NGHĨA TỪ VIẾT TẮT

|  |  |
| --- | --- |
| HTML | : Hypertext Markup Language (Ngôn ngữ đánh dấu siêu văn bản) |
| CSS | : Cascading Style Sheets (Tập tin định kiểu theo tầng) |
| MVC | : Model - View - Controler |
| API | : Application Programming Interface (Giao diện lập trình ứng dụng) |
| CSDL | : Cơ sở dữ liệu |
| COD | : Cash On Delivery |
| VD | : Ví dụ |
| BFD | : Business Function Diagram |
| DFD | : Data Flow Diagram |
| CRUD | : Create, Read, Update, Delete |

# LỜI MỞ ĐẦU

Trong thời đại công nghệ số hiện nay, Internet và các ứng dụng công nghệ đã và đang phát triển vô cùng mạnh mẽ, tạo nên những thay đổi trong cách thức con người tiếp cận thông tin và thực hiện các giao dịch. Đặc biệt, xu hướng mua sắm trực tuyến ngày càng phổ biến và trở thành một phần không thể thiếu trong đời sống hằng ngày, trong đó có cả lĩnh vực ẩm thực và dịch vụ.

Thực tế cho thấy, việc đặt đồ ăn theo cách truyền thống bằng cách gọi điện thoại trực tiếp hay đến tận nơi thường gặp nhiều điểm bất cập như: mất thời gian chờ đợi, khó khăn trong việc tìm hiểu thông tin món ăn, thanh toán không tiện lợi và đặc biệt là khó khăn trong việc so sánh giá cả giữa các cửa hàng. Để giải quyết những vấn đề này, việc xây dựng một hệ thống website đặt đồ ăn nhanh trực tuyến là một giải pháp thiết thực và cần thiết. Mục tiêu xây dựng hệ thống quản lý nhằm tạo ra một môi trường trực tuyến hiện đại, thuận tiện cho người dùng trong việc đặt đồ ăn. Giúp các nhà hàng, quán ăn dễ dàng quản lý và tối ưu hóa quy trình phục vụ khách hàng.

Mục tiêu của chuyên đề này tập trung vào việc phân tích, thiết kế và xây dựng một website đặt đồ ăn nhanh trực tuyến hoàn chỉnh và được thực hiện nhằm tổng kết lại những kiến thức, kỹ năng và kinh nghiệm trong suốt thời gian học tập. Đồng thời lắng nghe những góp ý và đánh giá về những điểm còn thiếu sót cần cải thiện để rút kinh nghiệm cho những dự án sắp tới.

CHƯƠNG 1: TỔNG QUAN VỀ CƠ SỞ LÝ THUYẾT

* 1. Xác định vấn đề trong việc đặt đồ ăn

Việc kinh doanh – mua bán là nhu cầu không thể thiếu trong cuộc sống hằng ngày, nên việc áp dụng công nghệ vào các hoạt động kinh doanh đã trở thành một điều tất yếu, đặc biệt trong lĩnh vực ẩm thực và dịch vụ đặt đồ ăn. Thực tế cho thấy, các phương thức đặt đồ ăn truyền thống như gọi điện thoại trực tiếp hoặc đến tận nơi thường gặp nhiều hạn chế: mất thời gian chờ đợi, khó khăn trong việc tìm hiểu thông tin món ăn, thanh toán không tiện lợi, và đặc biệt là việc so sánh giá cả giữa các nhà hàng không dễ dàng.

Để giải quyết những vấn đề này, việc xây dựng một website đặt đồ ăn trực tuyến là một giải pháp hiệu quả và cần thiết. Website không chỉ hỗ trợ khách hàng đặt món nhanh chóng, tiện lợi mà còn cung cấp thông tin chi tiết về món ăn, giá cả và các chương trình ưu đãi, tại chất lượng và hợp túi tiền của người tiêu dùng. Đồng thời, cũng là một cách quảng bá thương hiệu cửa hàng tiếp cận đến được nhiều đối tượng khách hàng hơn.

Việc xây dựng website, cung cấp nguồn hàng ổn định và trong quy trình quản lý online, đưa sản phẩm đến tay khách hàng một cách hiệu quả nhất. Nắm bắt được tình hình trên. Mong muốn xây dựng website đặt đồ ăn online nhằm giúp người dùng có thể đặt đồ ăn một cách tiện lợi hơn, đồng thời hỗ trợ các nhà hàng, quán ăn quản lý đơn hàng hiệu quả. Do hạn chế về thời gian và lịch làm việc, việc lựa chọn một địa chỉ đặt đồ ăn nhanh chóng, tiện lợi là điều rất cần thiết.

* 1. ASP.NET WebForms là gì

ASP.NET WebForms là một framework phát triển web được Microsoft giới thiệu vào năm 2002, thuộc nền tảng ASP.NET. WebForms được thiết kế để giúp các lập trình viên xây dựng ứng dụng web một cách nhanh chóng và dễ dàng thông qua mô hình lập trình dựa trên sự kiện (event-driven programming), tương tự như cách phát triển ứng dụng (1).



**Hình 1.1 Logo ASP.NET WebForms**

Trong WebForms, các thành phần giao diện được gọi là server controls (các điều khiển phía máy chủ), cho phép lập trình viên kéo-thả các thành phần như nút bấm, ô nhập dữ liệu, bảng biểu,..., trực tiếp trên giao diện thiết kế. Các điều khiển này tự động sinh ra mã HTML, CSS và JavaScript cần thiết, giúp giảm bớt gánh nặng về mặt front-end cho lập trình viên. WebForms sử dụng mô hình code-behind, tách biệt logic xử lý (dùng C#), giúp mã nguồn dễ quản lý hơn. Sau đây ta đi vào chi tiết về các thành phần và đặc điểm có trong ASP.NET WebForms:

* Page (.aspx): Là đơn vị cơ bản của một trang web. Mỗi trang có thể chứa các đoạn mã HTML, CSS để thể hiện nội dung và tính logic của trang.
* Controls: ASP.NET **cũng cung cấp các điều khiển web như textbox, button, label,.. Các điều khiển này giúp xây dựng giao diện người dùng và thể hiện dữ liệu thông qua các điều khiển đấy.**
* State Management: giúp quản lý trạng thái như Viewstate, Session và Cookie để lưu trữ thông tin tạm thời hoặc duy trì trạng thái giữa các yêu cầu.
* Authentication và Authorization: cung cấp các thông tin về cơ chế xác thực và phân quyền người dùng, cho phép kiểm tra danh tính người dùng và kiểm soát quyền truy cập tài nguyên của người dùng.

Một số đặc điểm chính của ASP.NET WebForms:

* **Hỗ trợ phát triển nhanh (RAD - Rapid Application Development):** Với các control có sẵn và tính năng kéo-thả, WebForms giúp phát triển ứng dụng web nhanh chóng, đặc biệt phù hợp với các dự án cần triển khai trong thời gian ngắn.
* **Tích hợp với các công nghệ Microsoft:** WebForms dễ dàng tích hợp với các công nghệ khác của Microsoft như ADO.NET (truy cập cơ sở dữ liệu), IIS (máy chủ web), và các thư viện .NET khác.

Ưu điểm của ASP.NET WebForms:

* **Dễ học và sử dụng:** WebForms cung cấp mô hình lập trình quen thuộc với các lập trình viên đã có kinh nghiệm với lập trình Windows Forms, nhờ cơ chế sự kiện và các điều khiển kéo-thả.
* **Tích hợp mạnh mẽ:** Hỗ trợ tốt các công cụ và thư viện của Microsoft, như Visual Studio, giúp tăng năng suất phát triển.
* **Quản lý trạng thái tự động:** WebForms có cơ chế quản lý trạng thái (ViewState) tích hợp sẵn, giúp lập trình viên dễ dàng duy trì trạng thái của trang web mà không cần xử lý thủ công.
* **Bảo mật cao:** Được tích hợp các tính năng bảo mật như mã hóa dữ liệu, xác thực người dùng, và chống tấn công giả mạo yêu cầu (CSRF), giúp đảm bảo an toàn cho ứng dụng web.

**Nhược điểm của ASP.NET WebForms:**

* **Hạn chế về hiệu suất:** Cơ chế ViewState có thể làm tăng kích thước dữ liệu truyền tải giữa client và server, dẫn đến thời gian tải trang chậm hơn, đặc biệt với các ứng dụng có giao diện phức tạp hoặc lượng người dùng lớn.
* **Khó kiểm soát mã giao diện:** Do các server controls tự động sinh mã HTML, lập trình viên có ít quyền kiểm soát mã giao diện, gây khó khăn khi cần tùy chỉnh giao diện phức tạp hoặc tối ưu hóa cho SEO.
* **Phụ thuộc nhiều vào server:** WebForms dựa nhiều vào xử lý phía server, làm giảm khả năng tương tác động (dynamic interaction) trên phía client so với các framework hiện đại hơn như ASP.NET MVC.
* **Không phù hợp với ứng dụng phức tạp:** WebForms thường không được khuyến nghị cho các ứng dụng web lớn, yêu cầu hiệu suất cao và khả năng mở rộng, do mô hình phát triển của nó có thể làm mã nguồn trở nên khó bảo trì.

1.2.1 So sánh WebForms với các famework khác

Ngoài ASP.NET WebForms ra, WebForms có thể được so sánh với các framework cùng hệ sinh thái .NET như MVC với khả năng tùy chỉnh cao và hiệu suất tốt, độ phức tạp cao nên đòi hỏi lập trình viên phải có kiến thức sâu về HTML, CSS, và JavaScript để xây dựng giao diện. Trong khi đó, WebForms tự động hóa việc tạo mã giao diện thông qua server controls, giúp giảm khối lượng công việc front-end. Tuy nhiên, để đánh giá đầy đủ tính phù hợp của WebForms đối với dự án FastBite, cần so sánh nó với các framework phát triển web khác, để biết được các khía cạnh như hiệu suất, khả năng tùy chỉnh giao diện, hiệu suất và độ phức tạp trong phát triển với quy mô dự án.

So sánh với ASP.NET MVC:

ASP.NET MVC, cũng thuộc hệ sinh thái .NET, được giới thiệu vào năm 2009 như một giải pháp thay thế hiện đại hơn cho WebForms. MVC tuân theo mô hình Model-View-Controller, tách biệt rõ ràng giữa dữ liệu (Model), giao diện (View), và logic xử lý (Controller). Điều này mang lại sự linh hoạt trong việc thiết kế giao diện và tối ưu hóa hiệu suất, đặc biệt với các ứng dụng web lớn hoặc yêu cầu khả năng mở rộng cao.

**Ưu điểm của ASP.NET MVC so với WebForms:**

* **Hiệu suất cao hơn**: MVC không sử dụng cơ chế ViewState như WebForms, giúp giảm kích thước dữ liệu truyền tải giữa client và server, từ đó cải thiện tốc độ tải trang.
* **Kiểm soát giao diện tốt hơn**: Lập trình viên có toàn quyền quản lý mã HTML, CSS, và JavaScript, giúp dễ dàng tùy chỉnh giao diện, tối ưu hóa cho SEO, và tích hợp với các thư viện front-end hiện đại như Bootstrap hoặc Tailwind CSS.
* **Hỗ trợ phát triển theo hướng RESTful**: MVC dễ dàng tích hợp với các API REST, phù hợp với các ứng dụng cần tương tác với ứng dụng di động hoặc các hệ thống bên ngoài.
* **Cộng đồng và tài liệu phong phú**: Là một framework hiện đại, MVC có cộng đồng hỗ trợ lớn và nhiều tài liệu cập nhật, giúp lập trình viên dễ dàng giải quyết các vấn đề kỹ thuật.

**Hạn chế của ASP.NET MVC:**

* **Độ phức tạp cao hơn**: MVC yêu cầu lập trình viên có kiến thức sâu về HTML, CSS, JavaScript, và mô hình MVC, điều này có thể gây khó khăn cho những người mới bắt đầu hoặc có hạn chế về kỹ năng front-end.
* **Thời gian phát triển lâu hơn**: Do không có các server controls tự động sinh mã như WebForms, việc xây dựng giao diện trong MVC thường tốn nhiều thời gian hơn, đặc biệt với các dự án nhỏ hoặc có thời gian hạn chế.

1.3 Lý do chọn WebForms thay vì MVC

**Từ ưu và nhược điểm của MVC và WebForms như đã phân tích trên, em đã cân nhắc giữa hai framework phổ biến của ASP.NET là WebForms và MVC. Sau khi phân tích yêu cầu và năng lực bản thân, em quyết định chọn WebForms vì các lý do sau:**

* **Hạn chế về kỹ năng:** Do bản thân còn hạn chế về kỹ năng và thiếu kinh nghiệm trong việc thiết kế giao diện người dùng, đặc biệt là việc hình dung các chức năng, giao diện web để tự viết và quản lý HTML, CSS, và JavaScript theo mô hình phân chia rõ ràng như MVC. Trong khi đó, WebForms cung cấp tính năng kéo-thả, giúp giảm thiểu công việc liên quan đến front-end và cho phép em tập trung vào phát triển các chức năng chính thay vì dành nhiều thời gian thiết kế giao diện cho hệ thống.
* **Tiết kiệm thời gian thiết kế:** Với các control có sẵn (như GridView, FormView, Button,...), Điều này đặc biệt hữu ích trong bối cảnh thời gian thực hiện chuyên đề, giúp em tập trung vào các chức năng chính của hệ thống đặt đồ ăn như quản lý menu, xử lý đơn hàng, và tích hợp thanh toán trực tuyến.
* **Phù hợp với quy mô dự án:** Website đặt đồ ăn nhanh FastBite tập trung vào các chức năng cơ bản như đặt món, quản lý đơn hàng, và thanh toán, không yêu cầu giao diện quá phức tạp hay khả năng tương tác cao. WebForms, với mô hình phát triển đơn giản và nhanh chóng, đáp ứng đủ các yêu cầu, trong khi MVC thường phù hợp hơn với các ứng dụng web phức tạp, cần tối ưu hóa hiệu suất và khả năng tùy chỉnh giao diện.
* Tận dụng tích hợp với công cụ Microsoft: WebForms tích hợp tốt với Visual Studio và SQL Server – các công cụ chính được sử dụng trong dự án này, giúp đơn giản hóa quá trình phát triển và triển khai hệ thống.

****1.4 Các công cụ liên quan****

Để phát triển website đặt đồ ăn nhanh FastBite một cách hiệu quả, ngoài việc sử dụng ASP.NET WebForms, hệ thống còn kết hợp của một số công cụ hỗ trợ khác. Các công nghệ này được lựa chọn dựa trên tính tương thích, hiệu suất và khả năng đáp ứng yêu cầu của đề tài. Cụ thể như sau:

* Ngôn ngữ lập trình C#: Đây là ngôn ngữ lập trình chính được sử dụng để xây dựng hệ thống. C# hỗ trợ lập trình phía server để xử lý các chức năng quan trọng như quản lý giỏ hàng, tính toán đơn hàng và cập nhật trạng thái giao dịch.
* SQL Server: SQL Server được sử dụng để lưu trữ thông tin dữ liệu của hệ thống. Ví dụ như các bảng dữ liệu như: bảng "Product" (lưu trữ thông tin món ăn) và bảng "Users" (lưu trữ tài khoản khách hàng). SQL Server đảm bảo hiệu suất truy xuất dữ liệu nhanh chóng và tính bảo mật cao, phù hợp với yêu cầu lưu trữ thông tin của website.
* HTML và CSS: Mặc dù WebForms đã cung cấp các server controls để tự động sinh mã HTML, việc sử dụng thêm HTML và CSS cơ bản giúp tùy chỉnh giao diện người dùng theo hướng trực quan và thân thiện hơn.
* API: Tích hợp API giúp tăng trải nghiệm người dùng khi tham gia sử dụng website. Như trong FastBite, chức năng thanh toán mã QR đã được triển khai bằng cách sử dụng API VietQR để tạo mã thanh toán động, nhưng trong tương lai, việc tích hợp thêm các API thanh toán khác có thể nâng cao tính linh hoạt.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1 Phân tích đặc tả và thiết kế hệ thống

FastBite là website đặt đồ ăn nhanh được xây dựng nhằm đáp ứng nhu cầu đặt món tiện lợi của khách hàng và hỗ trợ quản lý hiệu quả cho các nhà hàng, quán ăn. Để minh họa rõ hơn về thương hiệu và bản sắc của hệ thống, logo của FastBite được thiết kế đơn giản nhưng dễ nhận diện, với hình ảnh biểu tượng hamburger và khoai tây chiên, thể hiện đúng tinh thần của một nền tảng đặt đồ ăn nhanh.



**Hình 2.1 Logo Website FastBite**

Hệ thống đặt đồ ăn có các chức năng được sử dụng cho những đối tượng người dùng chính là: Người quản lý(Admin), khách hàng (người dùng). Mỗi nhóm được cung cấp các chức năng riêng biệt, được phân tích và thiết kế cẩn thận để đảm bảo đáp ứng đầy đủ nhu cầu và mang lại trải nghiệm tối ưu.

Hệ thống quản trị trang web:

**Admin**: Người thực hiện vận hành và giám sát toàn bộ hệ thống, với các chức năng quản lý được thiết kế các chức năng để hỗ trợ vận hành hiệu quả:

* Quản lý danh mục sản phẩm: Tạo, chỉnh sửa và xóa các danh mục món ăn (Pizza, Hamburger, Pizza).
* Quản lý sản phẩm: Thêm, xóa, cập nhật thông tin về sản phẩm bao gồm tên, giá, mô tả, hình ảnh và số lượng tồn kho, đảm bảo thực đơn luôn cập nhật và chính xác.
* Quản lý trạng thái đơn hàng: Theo dõi và cập nhật trạng thái đơn hàng (Đang xử lý, Đã gửi đi, Đã giao, Đã hủy) để đảm bảo quy trình giao hàng minh bạch và hiệu quả.
* Quản lý thông tin tài khoản người dùng: Hỗ trợ kiểm soát và bảo mật dữ liệu khách hàng.
* Xem đánh giá của khách hàng: Để thu thập và quản lý phản hồi cải thiện chất lượng dịch vụ.
* Đăng tải bài viết, chương trình khuyến mãi: Cập nhật tin tức, thông báo về món ăn mới hoặc ưu đãi để thu hút khách hàng.
* Thống kê doanh thu bán hàng: Cung cấp báo cáo doanh thu chi tiết theo khoảng thời gian, hỗ trợ Admin phân tích hiệu quả kinh doanh và đưa ra quyết định chiến lược.

**Khách Hàng**: Là đối tượng sử dụng chính trong hệ thống, từ các chức năng được thiết kế để mang lại trải nghiệm đặt món ăn trực tuyến tiện lợi, nhanh chóng và thú vị có trong hệ thống:

* Cho phép đăng ký, đăng nhập tài khoản: Cho phép khách hàng tạo tài khoản mới hoặc đăng nhập để truy cập hệ thống một cách an toàn.
* Cung cấp giao diện thông tin sản phẩm (mã sản phẩm, tên sản phẩm, hình ảnh, giá bán, mô tả,…)
* Hỗ trợ sửa đổi thông tin tài khoản người dùng: Cập nhật thông tin tài khoản như địa chỉ, số điện thoại hoặc email để đảm bảo giao hàng chính xác.
* Hỗ trợ đặt món và quản lý giỏ hàng: Chọn món, thêm vào giỏ hàng, cập nhật số lượng và đặt hàng.
* Thanh toán linh hoạt: Hỗ trợ đa dạng phương thức thanh toán như thanh toán thẻ, trả tiền khi nhận hàng(COD) hoặc quét mã QR và hóa đơn thanh toán,…
* Cung cấp lịch sử giao dịch và theo dõi tình trạng đơn hàng (Đang xử lý, đã gửi đi, đã giao, đã hủy đơn)
* Gửi phản hồi và đánh giá: Cho phép khách hàng gửi câu hỏi, ý kiến hoặc đánh giá để cải thiện dịch vụ.
* Cập nhật tin tức và ưu đãi: Xem thông tin về món ăn mới, chương trình khuyến mãi hoặc sự kiện từ nhà hàng để tận hưởng các lợi ích tốt nhất.
  1. Xác định yêu cầu và lập kế hoạch

2.2.1 Yêu cầu chức năng

* Hỗ trợ người dùng**:** Người dùng cần một nền tảng trực tuyến cho phép tìm kiếm, thêm giỏ hàng, tìm kiếm thông tin món ăn, thực hiện đặt hàng và theo dõi trạng thái đơn hàng và gửi đánh giá trải nghiệm sử dụng hệ thống.
* Hỗ trợ quản lý: Cung cấp cho nhà hàng, quán ăn một hệ thống quản lý đơn hàng hiệu quả, bao gồm quản lý thông tin tài khoản khách hàng, dễ dàng cập nhật, thêm sản phẩm, thường xuyên thay đổi hình ảnh, giá tiền các mặt hàng và cập nhật trạng thái đơn hàng(đã giao chưa, đã gửi đi,…), báo cáo tổng doanh thu theo mốc thời gian.

2.2.2 Yêu cầu phi chức năng

* Website dễ dùng, giao diện thân thiện: Website cần có giao diện trực quan, dễ sử dụng, đảm bảo người dùng ở mọi độ tuổi đều có thể dễ dàng sử dụng mà không gặp khó khăn.
* Giao diện đẹp, trực quan, dễ tiếp cận với người sử dụng: Thiết kế giao diện cần đảm bảo tính thẩm mỹ, bố cục rõ ràng, màu sắc hài hòa, và hỗ trợ tốt trên nhiều thiết bị (máy tính, điện thoại, máy tính bảng).

2.2.3 Lập kế hoạch triển khai

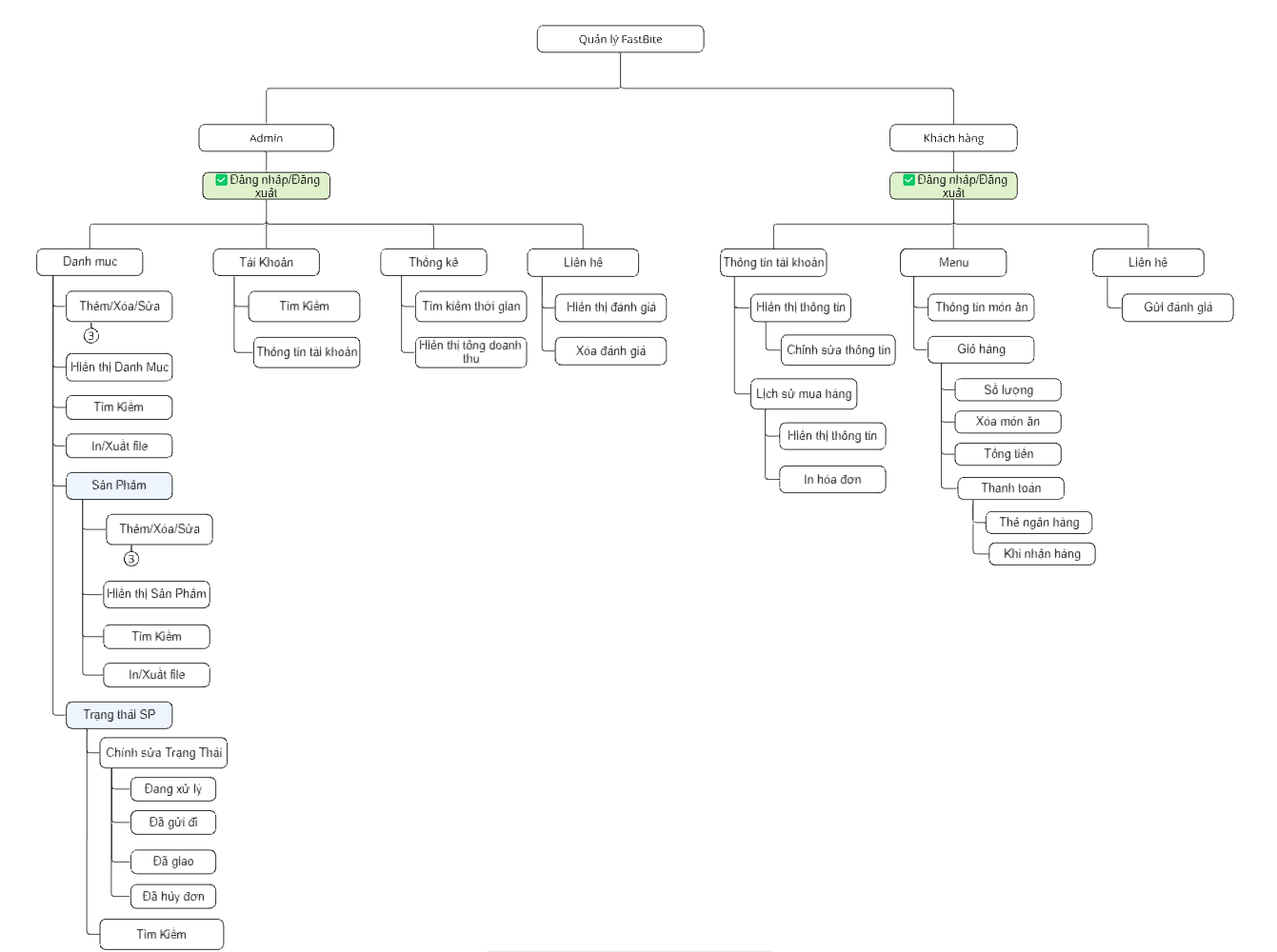
Việc triển khai hệ thống website đặt đồ ăn nhanh FastBite đòi hỏi cần một kế hoạch chi tiết và rõ ràng, để đảm bảo trong quá trình thực hiện đáp ứng đầy đủ các yêu cầu chức năng và phi chức năng đã xác định, đồng thời hoàn thành dự án trong thời gian sớm nhất và nguồn lực cho phép. Kế hoạch triển khai được xây dựng dựa trên các giai đoạn chính, từ phân tích yêu cầu đến bảo trì hệ thống, với sự chú trọng vào quản lý thời gian, tài nguyên, và các rủi ro tiềm ẩn. Dưới đây là các bước cụ thể trong kế hoạch xây dựng chương trình:

* **Phân tích yêu cầu:** Thu thập và phân tích các yêu cầu từ hai nhóm đối tượng người dùng (khách hàng) và nhà hàng (Admin). Các yêu cầu được xác định từ những nhu cầu thực tế, từ đó tạo tiền đề phát triển các tính năng cần thiết.
* **Thiết kế hệ thống:** Thiết kế giao diện và kiến trúc hệ thống dựa trên các yêu cầu về hiệu suất, bảo mật, bao gồm cả thiết kế cơ sở dữ liệu, giao diện người dùng. Từ đó xác định được công nghệ sẽ sử dụng trong website.
* **Xây dựng hệ thống:** Triển khai website bằng WebForms ASP.NET, bao gồm các phần mềm, phần cứng, server, và các tài nguyên cần thiết để đảm bảo hệ thống hoạt động ổn định.
* **Kiểm tra và triển khai:** Kiểm tra toàn diện hệ thống để đảm bảo hệ thống hoạt động ổn định, không có lỗi nghiêm trọng và đáp ứng các yêu cầu đề ra. Sau khi kiểm thử, hệ thống sẽ được triển khai và đưa vào sử dụng thực tế.
* **Bảo trì và nâng cấp:** Theo dõi hoạt động của website, đảm bảo hệ thống được hoạt động lâu dài và thu thập phản hồi từ người dùng để cải tiến, nâng cấp và bảo trì hệ thống trong dài hạn.
* **Quản lý rủi ro:** Để đảm bảo dự án được triển khai thành công, cần lên kế hoạch dự đoán các rủi ro tiềm ẩn liên quan tới kỹ thuật, thời gian hoặc tài nguyên,… Nhằm giảm thiểu các tác động của rủi ro, để đảm bảo dự án hoàn thành đúng tiến độ và đạt chất lượng tốt nhất
  1. Các sơ đồ mô tả hệ thống

Để thiết kế và triển khai hệ thống website đặt đồ ăn nhanh một cách hiệu quả, các sơ đồ mô tả hệ thống đóng vai trò quan trọng trong việc minh họa cấu trúc, luồng dữ liệu và các tương tác giữa các thành phần có trong hệ thống. Các sơ đồ được sử dụng bao gồm Sơ đồ phân cấp chức năng (BFD), Sơ đồ luồng dữ liệu mức bối cảnh (Context Diagram), Sơ đồ luồng dữ liệu mức đỉnh (DFD Level 1. Các sơ đồ này không chỉ giúp làm rõ yêu cầu mà còn hỗ trợ nhóm phát triển, kiểm thử, và bảo trì hệ thống trong suốt vòng đời dự án.

2.3.1 Sơ đồ phân cấp chức năng (BFD)

Sơ đồ phân cấp chức năng (Business Function Diagram - BFD) được xây dựng để minh họa một cách trực quan các chức năng chính của hệ thống và cách chúng được tổ chức thành các chức năng con. Sơ đồ này cung cấp cái nhìn trực quan về cấu trúc chức năng của hệ thống, giúp làm rõ cấu trúc hệ thống. Biểu đồ (BFD) đóng vai trò như một bản đồ dẫn đường, đảm bảo rằng tất cả các chức năng cần thiết đều được xác định và triển khai đầy đủ.

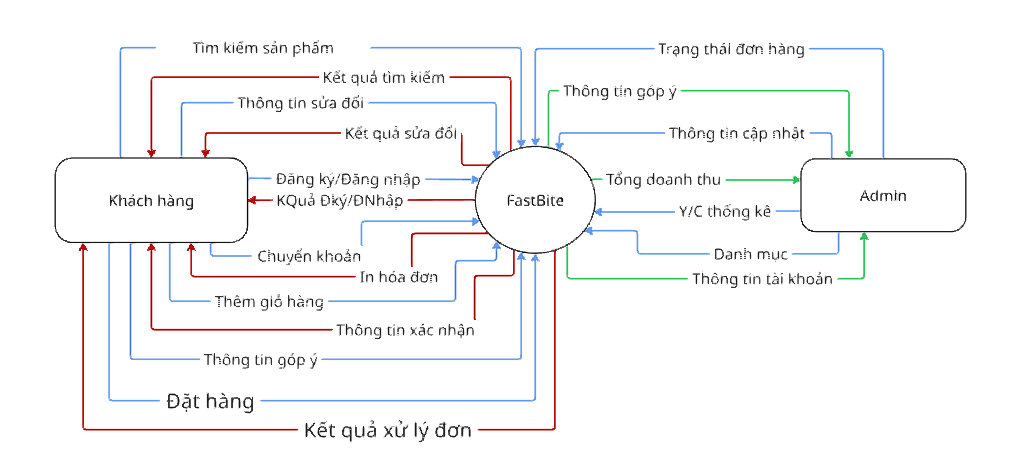


**Hình 2.2 Sơ đồ phân cấp chức năng**

Sơ đồ BFD được sử dụng trong giai đoạn phân tích yêu cầu để xác định các chức năng cần phát triển, làm cơ sở cho việc lập kế hoạch và phân bổ tài nguyên. Ví dụ: Khi thiết kế giao diện, nhóm phát triển sẽ sử dụng BFD để đảm bảo rằng mỗi chức năng (như "Thêm sản phẩm vào giỏ hàng") sẽ có giao diện và logic xử lý tương ứng.

2.3.2 Sơ đồ luồng dữ liệu mức bối cảnh

Sơ đồ luồng dữ liệu mức bối cảnh (Context Diagram) được xây dựng để mô tả tổng quan hệ thống, thể hiện các tương tác giữa hệ thống với các thực thể bên ngoài (External Entities) như Khách hàng và Admin. Sơ đồ dưới đây làm rõ các luồng dữ liệu chính như thông tin đặt hàng, đăng ký tài khoản, quản lý đơn hàng, và báo cáo doanh thu, giúp xác định ranh giới và phạm vi của hệ thống trước khi đi vào thiết kế chi tiết hơn.

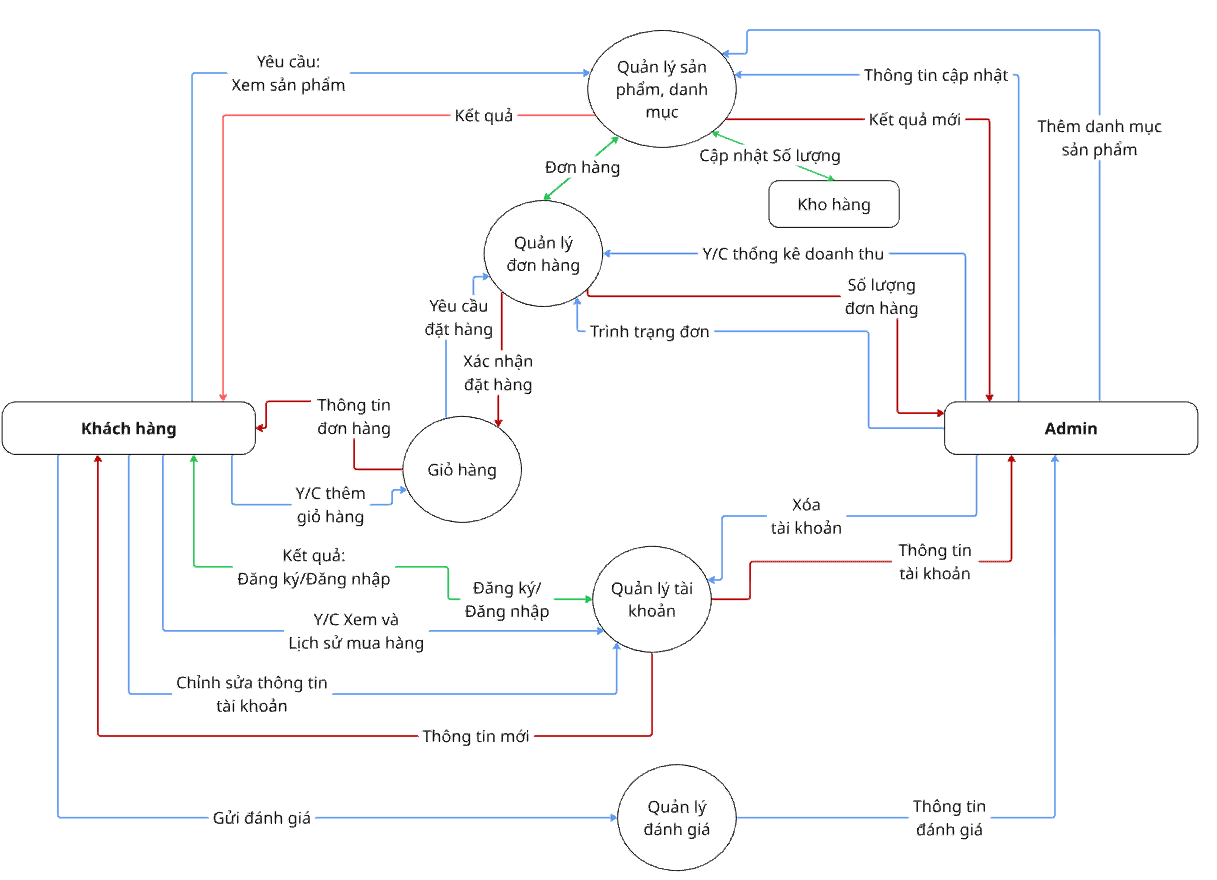


**Hình 2.3 Sơ đồ luồng dữ liệu (mức bối cảnh)**

Sơ đồ này được sử dụng để xác định các điểm tương tác chính giữa hệ thống và người dùng, giúp nhóm phát triển hiểu rõ dữ liệu đầu vào và đầu ra của hệ thống. Ví dụ: Khi thiết kế chức năng thanh toán, nhóm phát triển sử dụng Context Diagram để đảm bảo rằng thông tin thanh toán từ Khách hàng được xử lý đúng và trả về hóa đơn một cách chính xác.

2.3.3 Sơ đồ luồng dữ liệu mức đỉnh (DFD)

Sơ đồ luồng dữ liệu mức đỉnh (DFD Mức 1) được xây dựng để mô tả chi tiết các quy trình chính trong hệ thống, tập trung vào luồng dữ liệu giữa các chức năng như quản lý danh sách sản phẩm, xử lý đơn hàng, và quản lý tài khoản... Sơ đồ này thể hiện cách quy trình đưa dữ liệu di chuyển từ Khách hàng đến Admin và ngược lại. Cũng như các tương tác với các kho dữ liệu như thông tin món ăn, đơn hàng, và tài khoản.

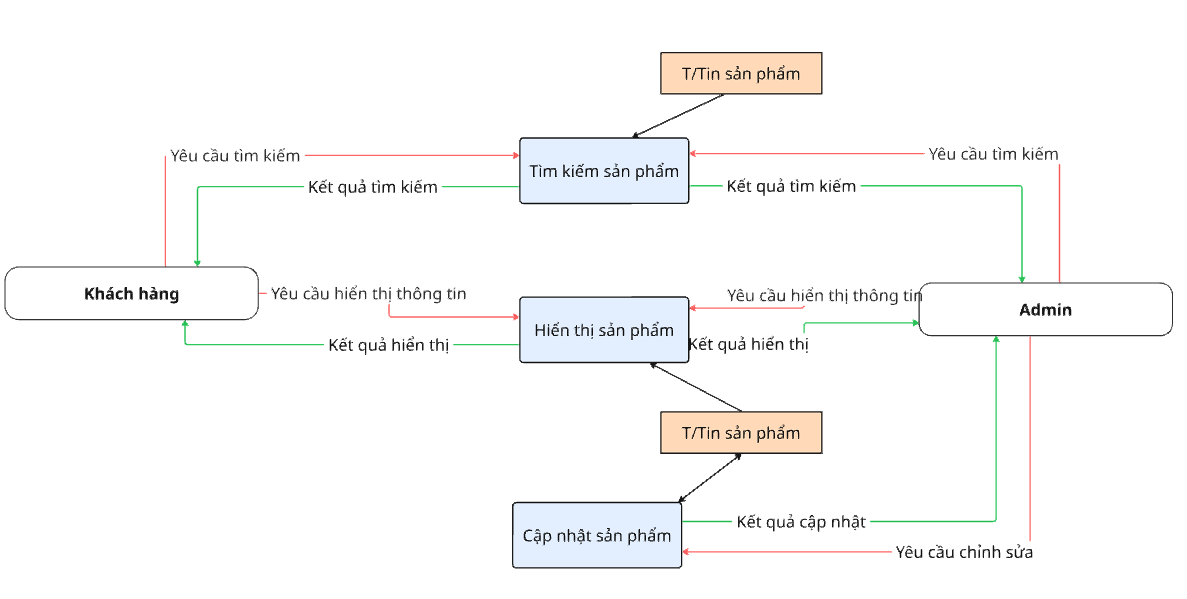


**Hình 2.4 Sơ đồ luồng dữ liệu mức đỉnh (DFD)**

Sơ đồ luồng mức đỉnh DFD Level 1 được sử dụng trong giai đoạn thiết kế hệ thống để xác định các quy trình và các kho dữ liệu cần tạo trong cơ sở dữ liệu.

2.3.4 Sơ đồ luồng dữ liệu – Quản lý đơn hàng

Sơ đồ luồng dữ liệu – quản lý đơn hàng dùng, cho phép người dùng và người quản lý thực hiện các truy cập có trong hệ thống: tìm kiếm thông tin sản phẩm, người quản lý được yêu cầu xử lý, thay đổi thông tin của sản phẩm, từ đó hệ thống sẽ xử lý và trả về kết quả cho người dùng trong hệ thống.

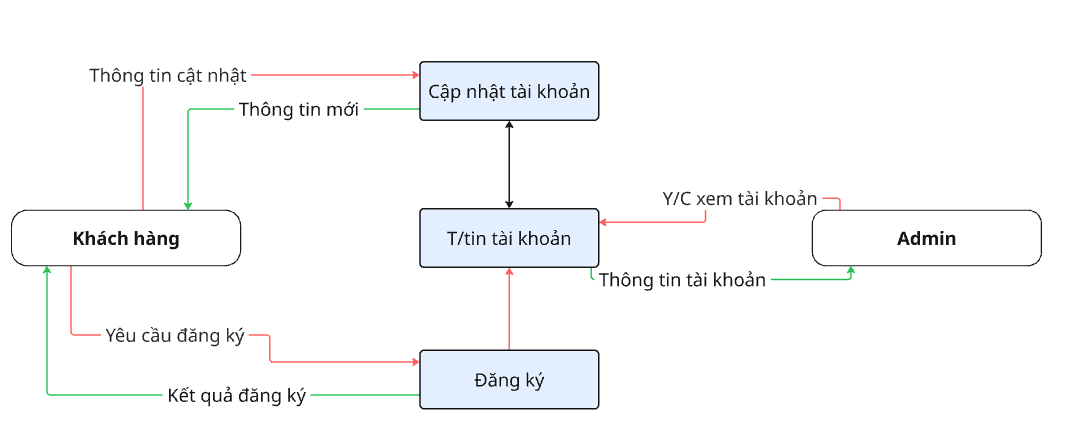


**Hình 2.5 Sơ đồ luồng dữ liệu – Quản lý đơn hàng**

Sơ đồ đảm bảo các thao tác liên quan tới đơn hàng sẽ được xử lý chính xác và đồng bộ khi thanh toán hệ thống sẽ kiểm tra lại số lượng sản phẩm trước khi cập nhật lại trạng thái đơn hàng.

2.3.5 Sơ đồ luồng dữ liệu – Quản lý tài khoản

Sơ đồ luồng dữ liệu – Quản lý tài khoản mô tả chi tiết quy trình xử lý thông tin tài khoản trong hệ thống, bao gồm các tương tác giữa khách hàng, người quản lý và CSDL. Sơ đồ này tập trung vào các hoạt động liên quan đến việc quản lý thông tin tài khoản người dùng, như đăng ký, đăng nhập, cập nhật thông tin, và xóa tài khoản.

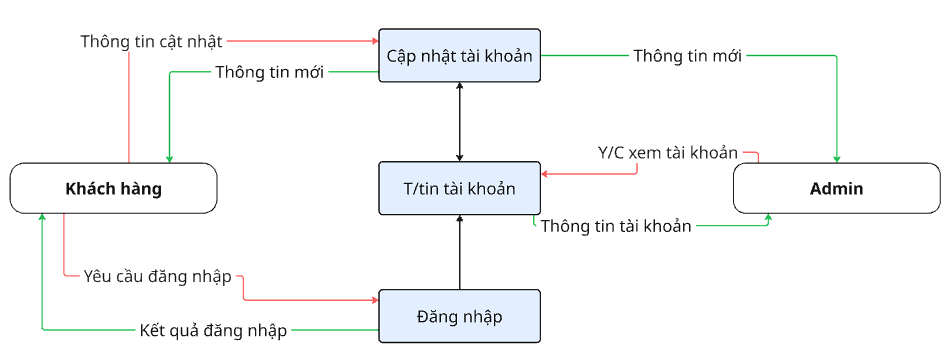


**Hình 2.6 Sơ đồ luồng dữ liệu – Quản lý tài khoản**

Sơ đồ hỗ trợ cho đăng ký tài khoản người dùng, sơ đồ đảm bảo dữ liệu sẽ được xử lý an toàn và an toàn.

2.3.6 Sơ đồ luồng dữ liệu – Quản lý người dùng

Sơ đồ luồng dữ liệu – Quản lý người dùng tương tự với quản lý thông tin tài khoản, tập trung vào vai trò của Admin trong việc theo dõi và xử lý dữ liệu liên quan đến khách hàng. Sơ đồ này hỗ trợ cho người quản lý trong việc quản lý thông tin người dùng một cách tập trung và hiệu quả, đảm bảo dữ liệu được duy trì chính xác liên quan đến các hành động quản lý tài khoản.



**Hình 2.7 Sơ đồ luồng dữ liệu – Quản lý người dùng**

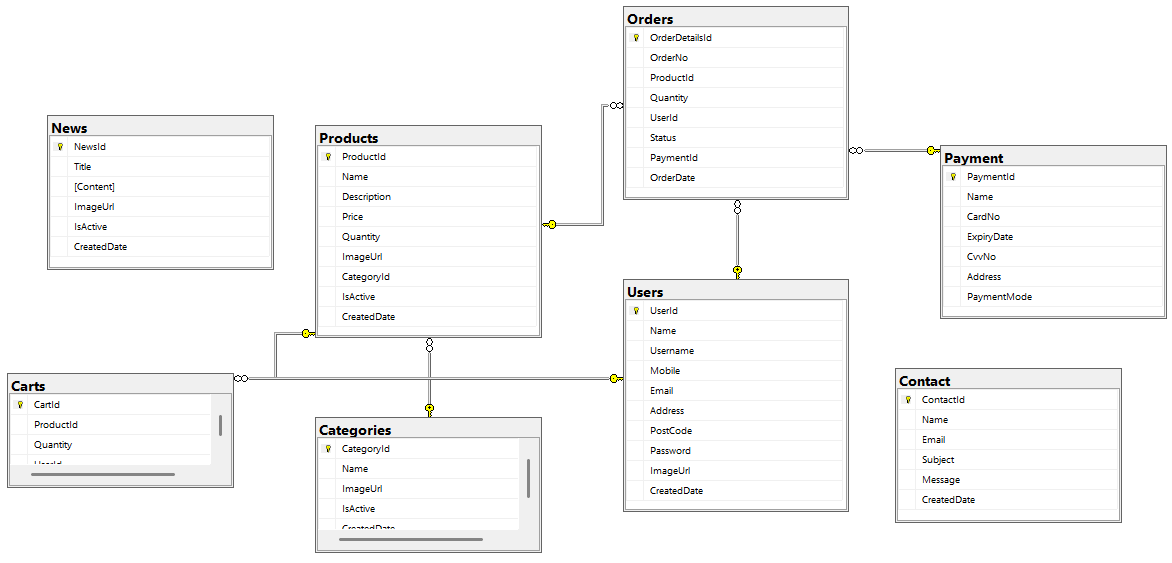
Sơ đồ hỗ trợ đăng nhập tài khoản người dùng, sơ đồ đảm bảo dữ liệu sẽ được xử lý an toàn và bảo mật khi khách hàng quên mật khẩu, hệ thống sử dụng quy trình khôi phục mật khẩu để gửi mã xác thực và cập nhật mật khẩu mới cho hệ thống,

CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG

3.1 Triển khai cơ sở dữ liệu

3.1.1 Tạo cơ sở dữ liệu và các bảng

Trong quá trình triển khai hệ thống đặt đồ ăn nhanh FastBite, cơ sở dữ liệu đóng vai trò lưu trữ và quản lý thông tin trong hệ thống. Sử dụng công cụ SQL Server Management Studio (SSMS) để thiết kế và quản lý. Cơ sở dữ liệu được đặt tên là FoodieDB, bao gồm các bảng chính như: Users (Người dùng), Categories (Danh mục), Products (Sản phẩm), Carts (Giỏ hàng), Orders (Đặt hàng), Payment (Thanh toán), Contact (Liên hệ) và News (Tin tức) nhằm đáp ứng yêu cầu lưu trữ thông tin về người dùng, sản phẩm, đơn hàng, và các giao dịch liên quan.



**Hình 3.1 Sơ đồ quan hệ của cơ sở dữ liệu**

3.1.2 Cơ sở dữ liệu của các bảng

* **Bảng User (Người dùng):** dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến thông tin cá nhân của người dùng từ các trường tương ứng:

**Bảng 3.1.1 Bảng User (Người dùng)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| UserId | INT | Lưu Id người dùng | Khóa chính |
| Name | NVARCHAR(150) | lưu tên người dùng |  |
| Username | NVARCHAR(150) | Lưu tên đăng nhập |  |
| Mobile | NVARCHAR(12) | Lưu số điện thoại |  |
| Email | NVARCHAR(150) | Lưu địa chỉ email |  |
| Address | NVARCHAR(MAX) | Lưu địa chỉ nhận đơn |  |
| PostCode | NVARCHAR(50) | Lưu mã Bưu Chính |  |
| Password | NVARCHAR(50) | Lưu mật khẩu người dùng |  |
| ImageUrl | NVARCHAR(MAX) | Lưu đường dẫn ảnh đại diện |  |
| CreatedDate | DATETIME | Lưu thời gian tạo tài khoản, |  |

Các mối quan hệ khóa ngoại

* **Bảng Categories (Danh mục):** Dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến danh mục của từng loại sản phẩm VD như: Pizza, Hamburger, Gà,…

**Bảng 3.1.2 Bảng Categories (Danh mục)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| CategoryId | INT | Lưu Id danh mục | Khóa chính |
| Name | NVARCHAR(150) | Lưu tên danh mục |  |
| ImageUrl | NVARCHAR(MAX) | Lưu ảnh sản phẩm |  |
| IsActive | BIT | Cho phép hiển thị danh mục |  |
| CreatedDate | DATETIME | Lưu thời gian tạo danh mục |  |

* **Bảng Contact (Liện hệ):** Dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến các câu hỏi, đánh giá của người dùng.

**Bảng 3.1.3 Bảng Contact (Liên hệ)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| ContactId | INT | Lưu Id liên hệ | Khóa chính |
| Name | NVARCHAR(150) | Lưu tên người dùng |  |
| Email | NVARCHAR(150) | Lưu địa chỉ email |  |
| Subject | NVARCHAR(200) | Lưu tên tiêu đề |  |
| Message | NVARCHAR(MAX) | Lưu nội dung liên hệ |  |
| CreatedDate | DATETIME | Lưu thời gian tạo liên hệ |  |

* **Bảng Product (Sản phẩm):** Dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến thông tin sản phẩm.

**Bảng 3.1.4 Bảng Product (Sản Phẩm)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| ProductId | INT | Lưu Id sản phẩm | Khóa chính |
| Name | NVARCHAR(150) | Lưu tên sản phẩm |  |
| Description | NVARCHAR(MAX) | Lưu mô tả sản phẩm |  |
| Price | DECIMAL(18,2) (12) | Lưu giá tiền của sản phẩm |  |
| Quantity | INT | Lưu số lượng sản phẩm |  |
| ImageUrl | NVARCHAR(MAX) | Lưu đường dẫn ảnh đại diện |  |
| CategoryId | INT | Liên kết đến bảng Categories | Khóa phụ |
| IsActive | BIT | Cho phép hiển thị sản phẩm |  |
| CreatedDate | DATETIME | Lưu thời gian tạo sản phẩm |  |

* **Bảng Carts (Giỏ hàng):** Dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến thông tin các sản phẩm trong tài khoản người dùng.

**Bảng 3.1.5 Bảng Carts (Giỏ hàng)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| CartId | INT | Lưu Id giỏ hàng | Khóa chính |
| ProductId | INT | Liên kết đến bảng Products | Khóa phụ |
| Quantity | INT | Lưu số lượng sản phẩm |  |
| UserId | INT | Liên kết đến bảng Users | Khóa phụ |

* **Bảng Orders (Đơn hàng):** Dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến thông tin các đơn đặt hàng của người dùng.

**Bảng 3.1.6 Bảng Orders (Đơn hàng)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| OrderDetailsId | INT | Lưu Id sản phẩm | Khóa chính |
| OrderNo | NVARCHAR(MAX) | Lưu mã đơn hàng |  |
| ProductId | INT | Liên kết đến bảng Products | Khóa phụ |
| Quantity | INT | Lưu số lượng sản phẩm của đơn hàng |  |
| UserId | INT | Liên kết đến bảng Users | Khóa phụ |
| Status | NVARCHAR(50) | Lưu trạng thái đơn hàng |  |
| PaymentId | INT | Liên kết đến bảng Payment | Khóa phụ |
| OrderDate | DATETIME | Lưu thời gian tạo đơn hàng |  |

* **Bảng Payment (Thanh toán):** Dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến thông tin giao dịch của người dùng.

**Bảng 3.1.7 Bảng Payment (Thanh toán)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| PaymentId | INT | Lưu Id thanh toán | Khóa chính |
| Name | NVARCHAR(150) | Lưu tên chủ thẻ tín dụng |  |
| CardNo | NVARCHAR(150) | Lưu số thẻ tín dụng |  |
| ExpiryDate | NVARCHAR(150) | Lưu ngày hết hạn của thẻ |  |
| CvvNo | INT | Mã bảo mật của thẻ |  |
| Address | NVARCHAR(MAX) | Lưu địa chỉ nhận hàng |  |
| PaymentMode | NVARCHAR(50) | Phương thức thanh toán (thẻ, trả khi nhận hàng) |  |

* **Bảng News (Tin tức):** Dùng để lưu trữ hoặc sao lưu các dữ liệu liên quan đến bài viết, tin tức của hệ thống.

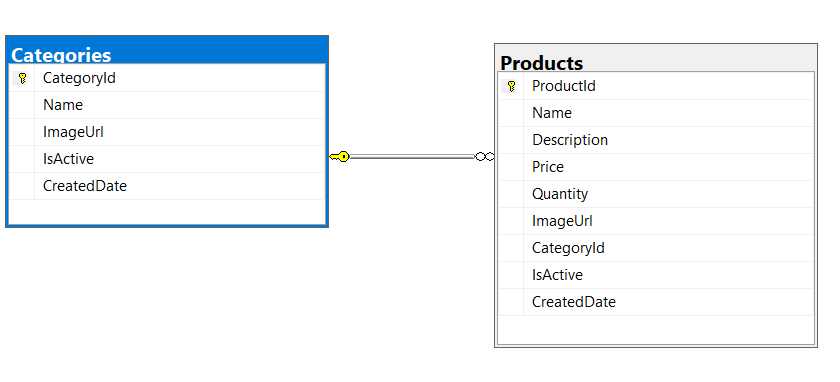
**Bảng 3.1.8 Bảng News (Tin tức)**

|  |  |  |  |
| --- | --- | --- | --- |
| Tên Trường dữ liệu | Kiểu dữ liệu | Mô tả | Ràng buộc dữ liệu |
| NewsId | INT | Lưu Id tin tức | Khóa chính |
| Title | NVARCHAR(200) | Lưu tên tiêu đề |  |
| Content | NVARCHAR(MAX) | Lưu nội dung bài viết |  |
| ImageUrl | NVARCHAR(MAX) | Lưu đường dẫn ảnh đại diện |  |
| IsActive | BIT | Cho phép hiển thị danh mục |  |
| CreatedDate | DATETIME | Phương thức thanh toán (thẻ, trả khi nhận hàng) |  |

3.1.3 Mối quan hệ giữa các bảng

Cơ sở dữ liệu **FoodieDB** được liên kết với nhau thông qua các mối quan hệ giữa các bảng bằng các khóa ngoại, đảm bảo tính toàn vẹn dữ liệu và hỗ trợ việc truy vấn thông tin một cách hiệu quả. Sơ đồ quan hệ của cơ sở dữ liệu đã được trình bày trong hình bên dưới, minh họa rõ ràng các mối quan hệ giữa các bảng. Dưới đây là mô tả chi tiết về các mối quan hệ khóa ngoại:

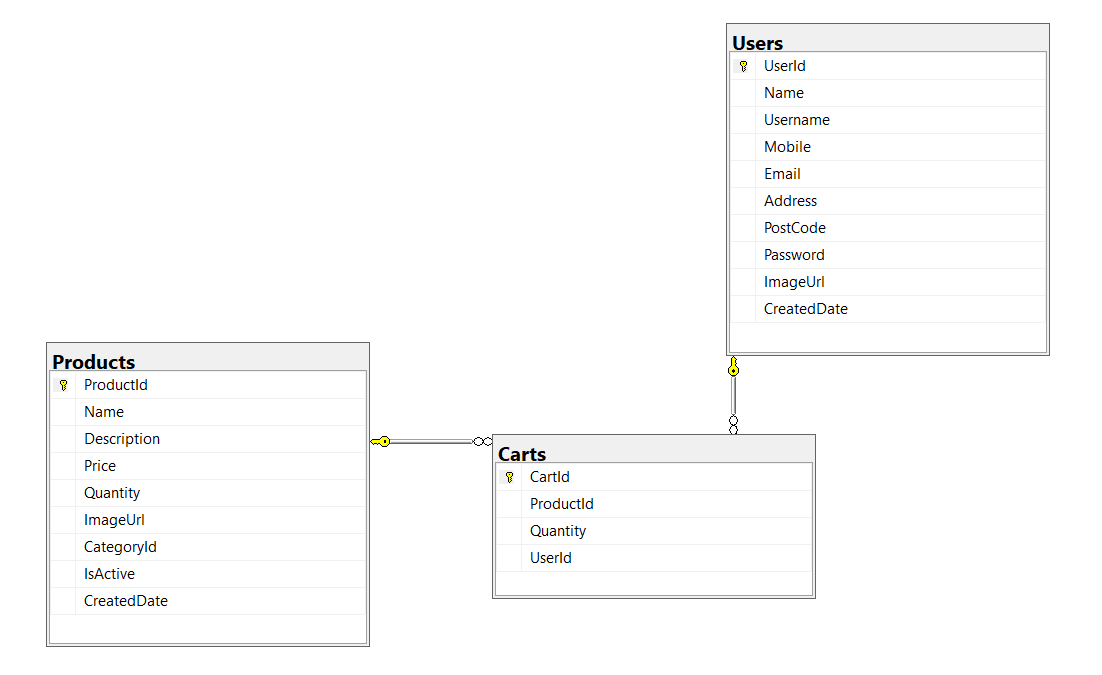
* Mối quan hệ giữa bảng Products và bảng Categories:



**Hình 3.2 Mối quan hệ bảng Products và bảng Categories**

Trường CategoryId trong bảng Products được thiết lập là khóa ngoại, tham chiếu đến khóa chính CategoryId của bảng Categories. Mối quan hệ này thể hiện rằng mỗi sản phẩm sẽ thuộc về một danh mục tương ứng, cho phép hệ thống phân loại và quản lý sản phẩm theo các nhóm (món ăn, đồ uống,…). Điều này hỗ trợ việc hiển thị sản phẩm theo danh mục trên giao diện người dùng và tối ưu hóa truy vấn dữ liệu.

* Mối quan hệ giữa bảng Carts với bảng Products và bảng Users:



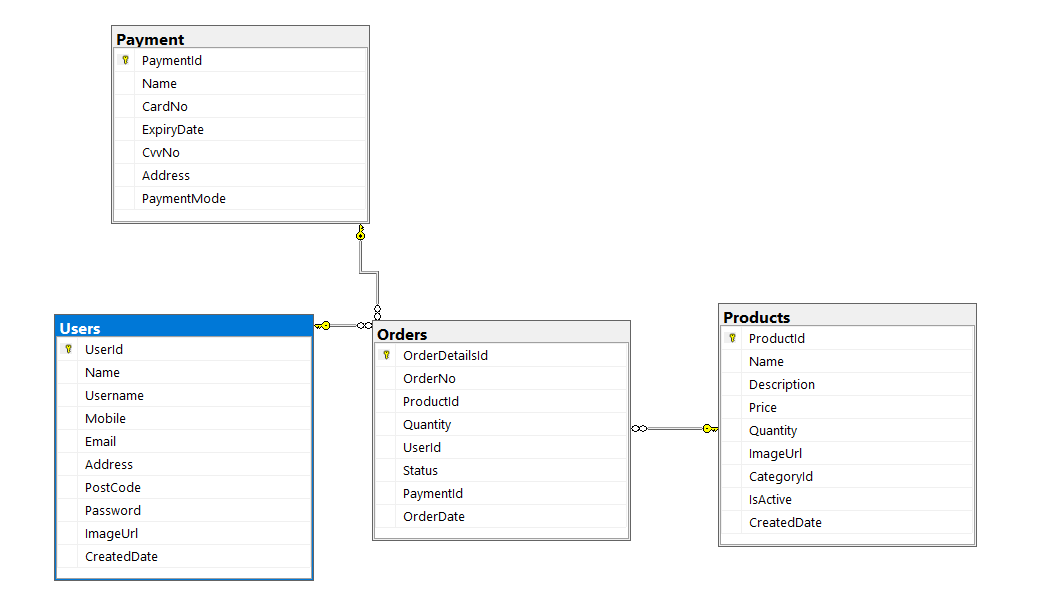
**Hình 3.3 Mối quan hệ bảng Carts với bảng Products và bảng Users**

Bảng **Carts** có hai trường khóa ngoại:

* Trường *ProductId* tham chiếu đến khóa chính *ProductId* của bảng **Products**, thể hiện sản phẩm được thêm vào giỏ hàng.
* Trường *UserId* tham chiếu đến khóa chính *UserId* của bảng **Users**, xác định người dùng sở hữu giỏ hàng.

Mối quan hệ này cho phép hệ thống quản lý giỏ hàng của từng người dùng, lưu trữ thông tin về các sản phẩm được chọn và số lượng tương ứng, từ đó hỗ trợ cho quy trình đặt hàng.

* Mối quan hệ giữa bảng Orders với bảng Products, bảng Users và bảng Payment:



**Hình 3.4 Mối quan hệ giữa bảng Orders, Products, Users và Payment**

Bảng Orders được liên kết với ba bảng khác thông qua các khóa ngoại:

* Trường ProductId tham chiếu đến khóa chính ProductId của bảng Products, xác định sản phẩm trong đơn hàng.
* Trường UserId tham chiếu đến khóa chính UserId của bảng Users, xác định người dùng đặt đơn hàng.
* Trường PaymentId tham chiếu đến khóa chính PaymentId của bảng Payment, liên kết đơn hàng với thông tin thanh toán.

Mối quan hệ này thể hiện rằng mỗi đơn hàng bao gồm các sản phẩm được mua bởi một người dùng cụ thể và có thông tin thanh toán liên quan, đảm bảo tính đồng bộ trong quy trình xử lý đơn hàng.

Đó là tất cả các mối quan hệ khóa ngoại giữa các bảng trong cơ sở dữ liệu, nó không chỉ hỗ trợ việc quản lý thông tin về sản phẩm, giỏ hàng, và đơn hàng một cách hiệu quả, mà còn tạo nền tảng vững chắc cho các chức năng cốt lõi của hệ thống FastBite hoạt động mượt mà và đáp ứng nhu cầu đặt đồ ăn của người dùng.

3.2 Tạo Stored Procedure

Sau khi thiết lập cơ sở dữ liệu và các mối quan hệ giữa các bảng, việc triển khai các Stored Procedure để hỗ trợ quản lý và tối ưu hóa quy trình xử lý dữ liệu trong hệ thống FastBite. Các Stored Procedure này được thiết kế để thực hiện các hành động như thêm, sửa, xóa dữ liệu, cũng như hỗ trợ tạo báo cáo, thống kê, giúp cải thiện hiệu suất, tăng tính bảo mật. Trong hệ thống bao gồm 11 Stored Procedure, phục vụ các chức năng chính của website.

Dưới đây là một số Stored Procedure tiêu biểu và được dùng nhiều nhất:

* **Category\_Crud**: Được dùng để quản lý danh mục sản phẩm. Stored Procedure này hỗ trợ các hành động như: thêm, cập nhật, xóa, truy vấn tất cả danh mục, truy vấn danh mục theo ID, và truy vấn danh mục đang hoạt động. Một điểm đáng chú ý là trong hành động cật nhật, Stored Procedure sẽ kiểm tra xem có cập nhật hình ảnh không? để cập nhật dữ liệu phù hợp, đảm bảo tính linh hoạt.
* **Product\_Crud**: Quản lý thông tin sản phẩm với các hành động tương tự như Category\_Crud bao gồm: thêm, sửa, xóa, truy vấn sản phẩm và cập nhật số lượng hàng. Stored Procedure này cũng cho phép truy vấn sản phẩm đang hoạt động và kết hợp với bảng Categories để lấy thông tin danh mục liên quan.
* **Invoice**: Hỗ trợ tạo hóa đơn và quản lý trạng thái đơn hàng. Stored Procedure này cung cấp các chức năng như lấy chi tiết hóa đơn theo ID của PaymentId và UserId, truy xuất lịch sử đơn hàng, lấy trạng thái đơn hàng, và cập nhật trạng thái đơn hàng. Đây là một Stored Procedure quan trọng trong việc quản lý giao dịch và theo dõi đơn hàng.
* **Dashboard**: Cung cấp các số liệu thống kê cho giao diện quản trị, bao gồm: số lượng danh mục, sản phẩm, đơn hàng, đơn hàng đã giao, đơn hàng đang xử lý, đơn hàng bị hủy, số lượng tài khoản người dùng, tổng doanh thu, và số lượng liên hệ từ khách hàng. Stored Procedure này giúp người quản lý có cái nhìn tổng quan hơn về hoạt động của hệ thống.

Các Stored Procedure còn lại được dùng để:

* **ContactSp**: Quản lý thông tin liên hệ từ khách hàng, hỗ trợ cho các hành động thêm mới, lấy danh sách liên hệ giúp hệ thống thu thập ý kiến người dùng để cải thiện dịch vụ.
* **Cart\_Crud**: Quản lý các sản phẩm có trong giỏ hàng của người dùng, đảm bảo kiểm tra và quản lý chính xác, đồng bộ dựa trên Id sản phẩm và Id người dùng tương ứng.
* **Save\_Payment**: Lưu thông tin thanh toán của người dùng sau khi hoàn tất đơn hàng, hỗ trợ lưu thông tin của các phương thức thanh toán
* **SellingReport**: Tạo báo cáo doanh thu theo khoảng thời gian, cung cấp thông tin về đơn hàng, khách hành và phục vụ nhu cầu thống kê cho người quản lý.
* **User\_Crud**: Quản lý thông tin tài khoản người dùng, bao gồm liên kết với Id người dùng với id sản phẩm, phục vụ có việc dễ dàng quản lý giỏ hàng của người dùng một cách hiệu quả hoặc chọn id người dùng thuận tiện cho việc quản lý liên hệ đến với khách hàng một cách chính xác.
* **Save\_Orders**: Lưu thông tin các đơn hàng từ một bảng dữ liệu tạm thời và đảm bảo dữ liệu đơn hàng được lưu trữ chính xác và đồng bộ với dữ liệu đơn hàng.
* **News\_Crud**: Quản lý thông tin bài viết hoặc tin tức trên hệ thống, bao gồm thêm, sửa, xóa, truy vấn bài viết, và hiển thị bài viết đang hoạt động, đảm bảo cập nhật nhanh chóng và hiển thị đúng đối tượng

Tất cả các Stored Procedure đều được thiết kế với cấu trúc rõ ràng, chi tiết mã nguồn của từng Stored Procedure được đính kèm đầy đủ ở file thư mục của báo cáo để tiện cho việc đánh giá. Các Stored Procedure dùng để xác định hành động cần thực hiện, và để giảm thiểu thông báo trả về, từ đó tối ưu hiệu suất truy vấn.

CHƯƠNG 4: CÀI ĐẶT GIAO DIỆN HỆ THỐNG

4.1 Tổng quan về giao diện hệ thống FastBite

Hệ thống giao diện của website đặt đồ ăn nhanh FastBite được thiết kế dành cho hai nhóm đối tượng chính: Admin (người quản lý) và Khách hàng (người dùng). Mỗi nhóm người dùng được cung cấp hai giao diện riêng biệt, đáp ứng nhu cầu và vai trò cụ thể trong hệ thống.

Đối với Admin, giao diện được xây dựng để hỗ trợ quản lý hiệu quả các hoạt động của hệ thống, bao gồm việc theo dõi thông tin sản phẩm, đơn hàng, tài khoản khách hàng và các báo cáo liên quan. Giao diện Admin được thiết kế tối giản, chỉ tập trung vào tính năng quản lý và thống kê, giúp người quản lý dễ dàng quản lý và đưa ra quyết định mà không đặt nặng về tính thẩm mĩ.

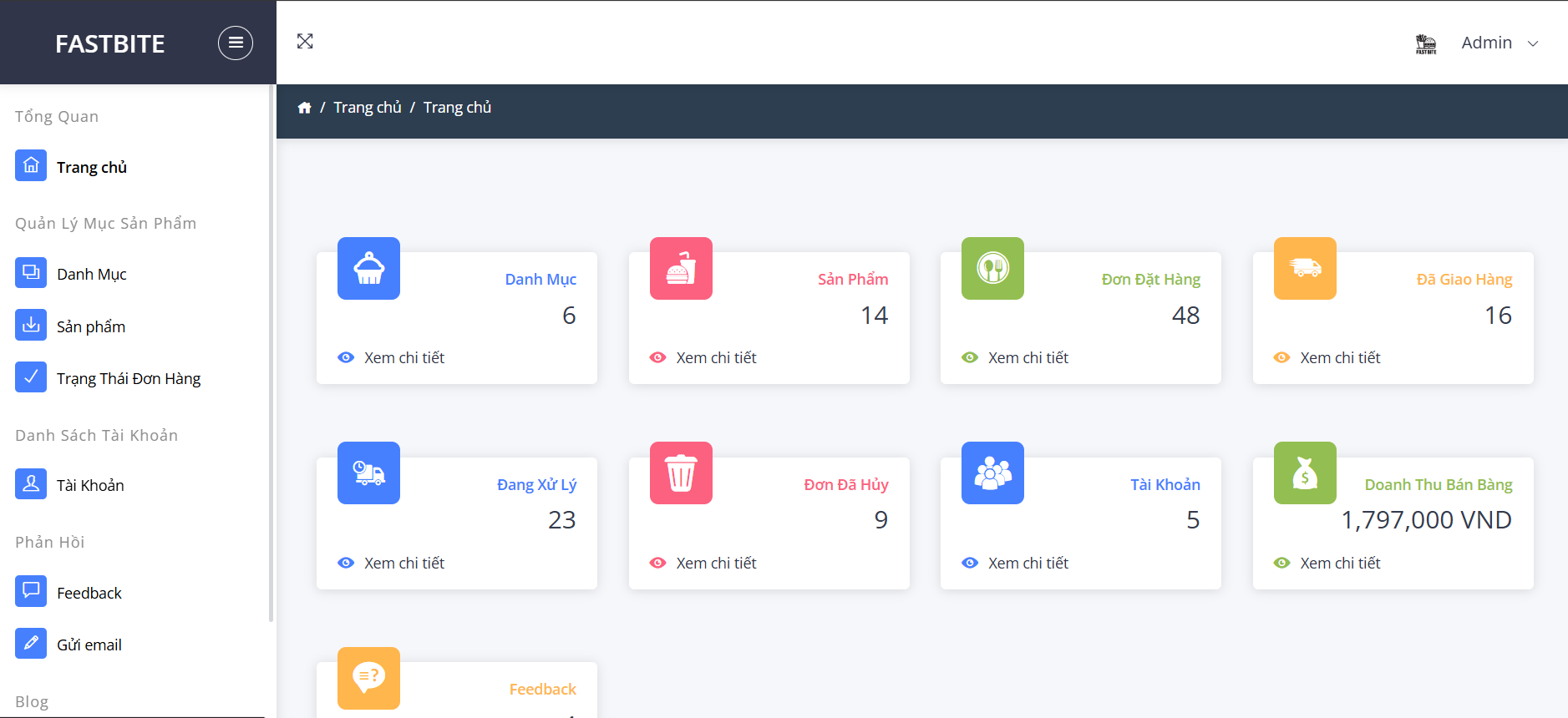
Đối với Khách hàng, phải mang đến một giao diện thân thiện, bắt mắt, trực quan, hỗ trợ các chức năng như tìm kiếm món ăn, thêm vào giỏ hàng, đặt hàng, theo dõi trạng thái đơn hàng và gửi đánh giá. Giao diện này được tối ưu hóa để đảm bảo trải nghiệm người dùng mượt mà, dễ sử dụng trên nhiều thiết bị khác nhau.

4.2 Giao diện quản lý (Admin)

Giao diện quản lý của hệ thống FastBite được thiết kế để hỗ trợ Admin thực hiện các tác vụ quản lý các hoạt động của hệ thống một cách hiệu quả. Giao diện bao gồm nhiều chức năng chính như quản lý danh mục sản phẩm, sản phẩm, trạng thái đơn hàng, tài khoản người dùng, phản hồi từ khách hàng và báo cáo doanh thu… Dưới đây là phần mô tả chi tiết các thành phần giao diện quản lý.

4.2.1 Giao diện Dashboard

Giao diện Dashboard của hệ thống FastBite được thiết kế cho Admin, giúp cung cấp cái nhìn tổng quan về các số liệu quan trọng của hệ thống thông qua các thẻ thông tin trực quan. Các số liệu bao gồm: số lượng danh mục, sản phẩm, đơn hàng, đơn hàng đã giao, đang xử lý, bị hủy, tài khoản người dùng, tổng doanh thu, và phản hồi từ khách hàng. Giao diện sử dụng bố cục lưới (grid layout) với các biểu tượng và màu sắc nổi bật, giúp Admin dễ dàng nhận biết và truy cập chi tiết thông qua các liên kết "Xem chi tiết".

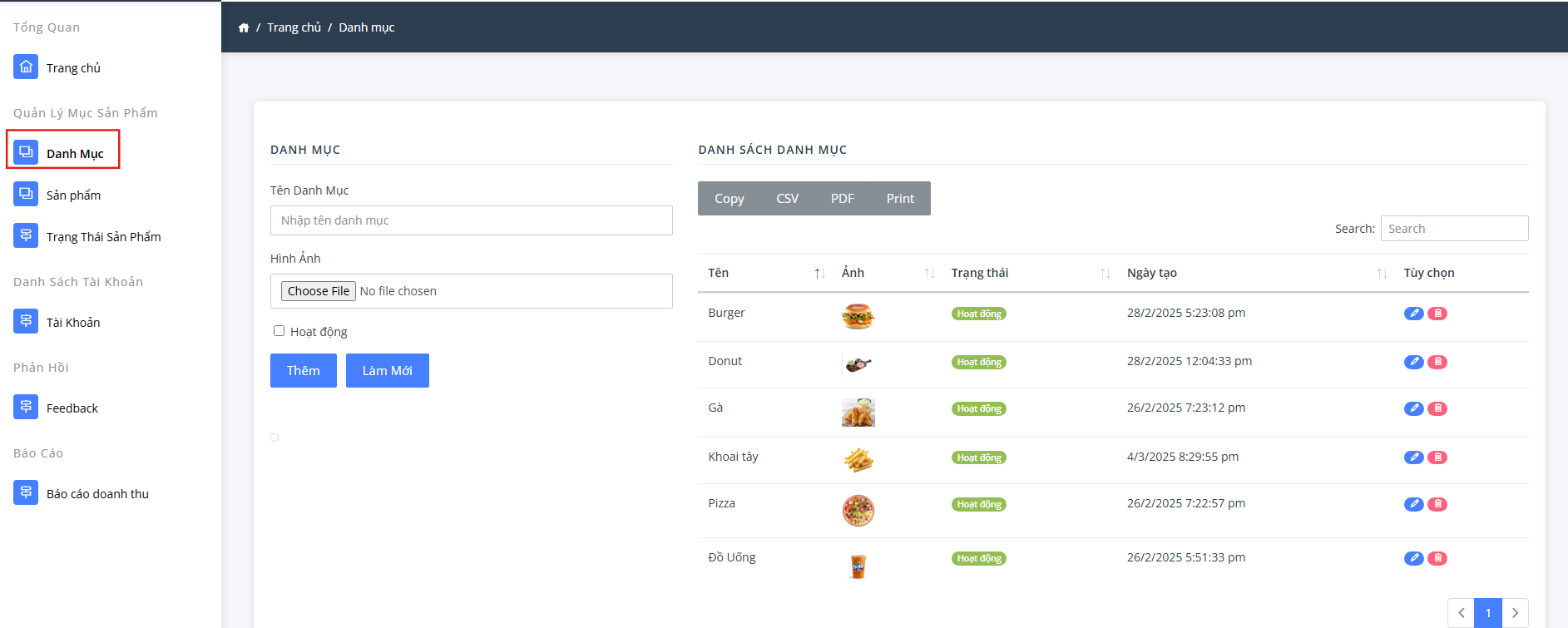


**Hình 4.1 Giao diện trang chủ Admin**

Giao diện tập trung vào hiển thị dữ liệu thống kê của Dashboard. Giao diện sử dụng các thẻ để tích hợp với Master Page để đảm bảo tính đồng nhất về giao diện quản lý. Mỗi thẻ thông tin hiển thị một số liệu cụ thể, ví dụ: số lượng danh mục, số lượng sản phẩm, hay doanh thu,... Mỗi một thẻ đều liên kết đến trang chi tiết (VD: Category.aspx, OrderStatus.aspx), cho phép Admin xem và quản lý dữ liệu sâu hơn.

4.2.2 Giao diện quản lý Danh Mục Sản Phẩm

Giao diện quản lý Danh Mục Sản Phẩm được thiết kế để hỗ trợ Admin thực hiện các chức năng cơ bản như: tìm kiếm thêm mới, chỉnh sửa, xóa và tìm kiếm danh sách danh mục sản phẩm. Giao diện được chia thành hai phần chính: Form nhập dữ liệu cho phép Admin nhập hoặc chỉnh sửa thông tin danh mục, và Danh sách danh mục hiển thị thông tin chi tiết các danh mục hiện có dưới dạng bảng với các tùy chọn chỉnh sửa hoặc xóa.

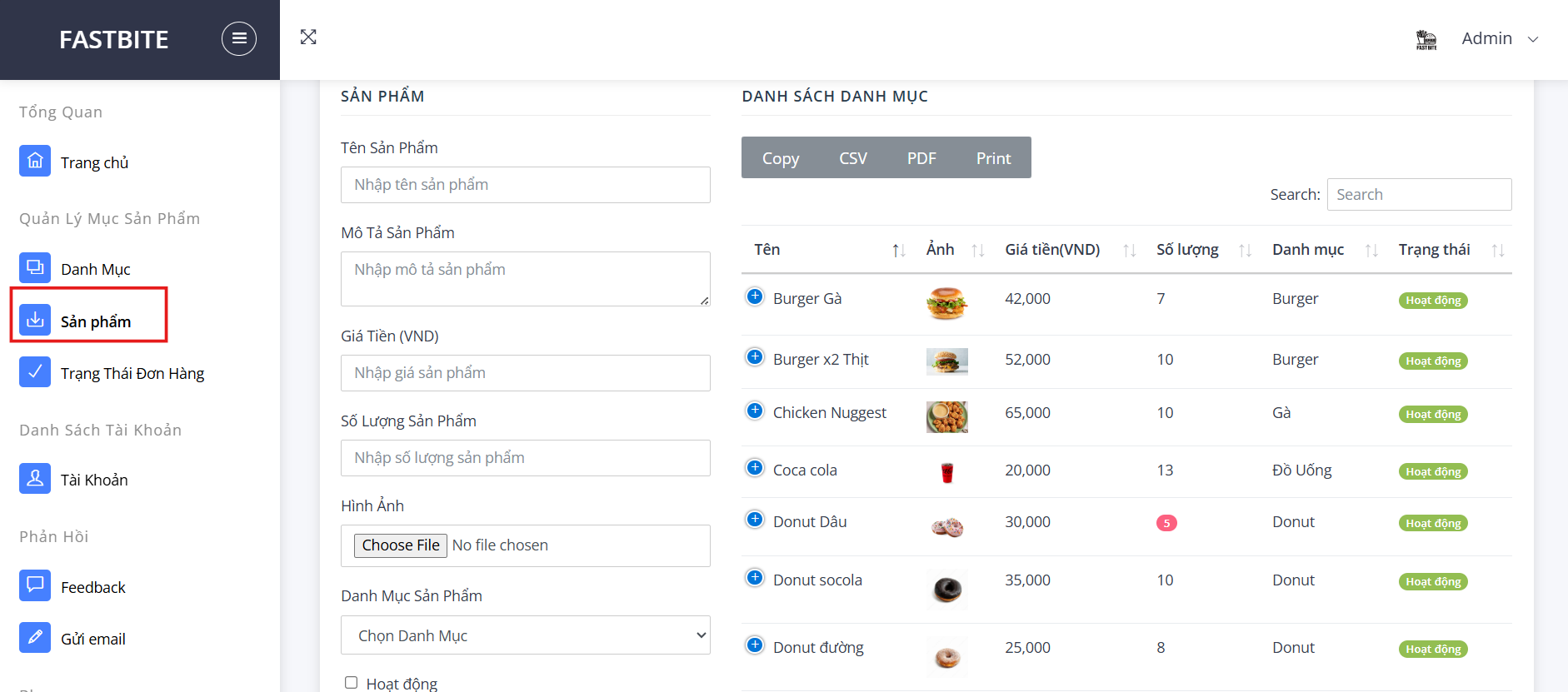


**Hình 4.2 Giao diện quản lý Danh Mục Sản Phẩm**

Giao diện quản lý Danh Mục được tích hợp với Master Page để đảm bảo tính đồng nhất với các trang giao diện quản lý khác trong hệ thống. Forms nhập liệu bao gồm các trường như tên danh mục, hình ảnh đại diện, và trạng thái hoạt động, giúp Admin nhanh chóng cập nhật thông tin mà không cần thao tác phức tạp. Danh sách danh mục hiển thị các thông tin chính như tên, hình ảnh, trạng thái, và ngày tạo, kèm theo các nút chỉnh sửa hoặc xóa vô cùng tiện lợi. Giao diện này đảm bảo Admin có thể quản lý danh mục một cách hiệu quả, giữ thực đơn luôn được cập nhật sớm nhất và đưa tới khách hàng một cách nhanh chóng.

4.2.3 Giao diện quản lý Sản Phẩm

Giao diện quản lý Sản Phẩm được thiết kế để hỗ trợ Admin thực hiện các chức năng cơ bản như thêm mới, chỉnh sửa, xóa và tìm kiếm từ danh sách sản phẩm. Giao diện này được chia thành hai phần chính: Form nhập dữ liệu cho phép Admin nhập hoặc chỉnh sửa thông tin sản phẩm, và Danh sách sản phẩm hiển thị thông tin chi tiết các sản phẩm hiện có dưới dạng bảng với các tùy chọn chỉnh sửa hoặc xóa.

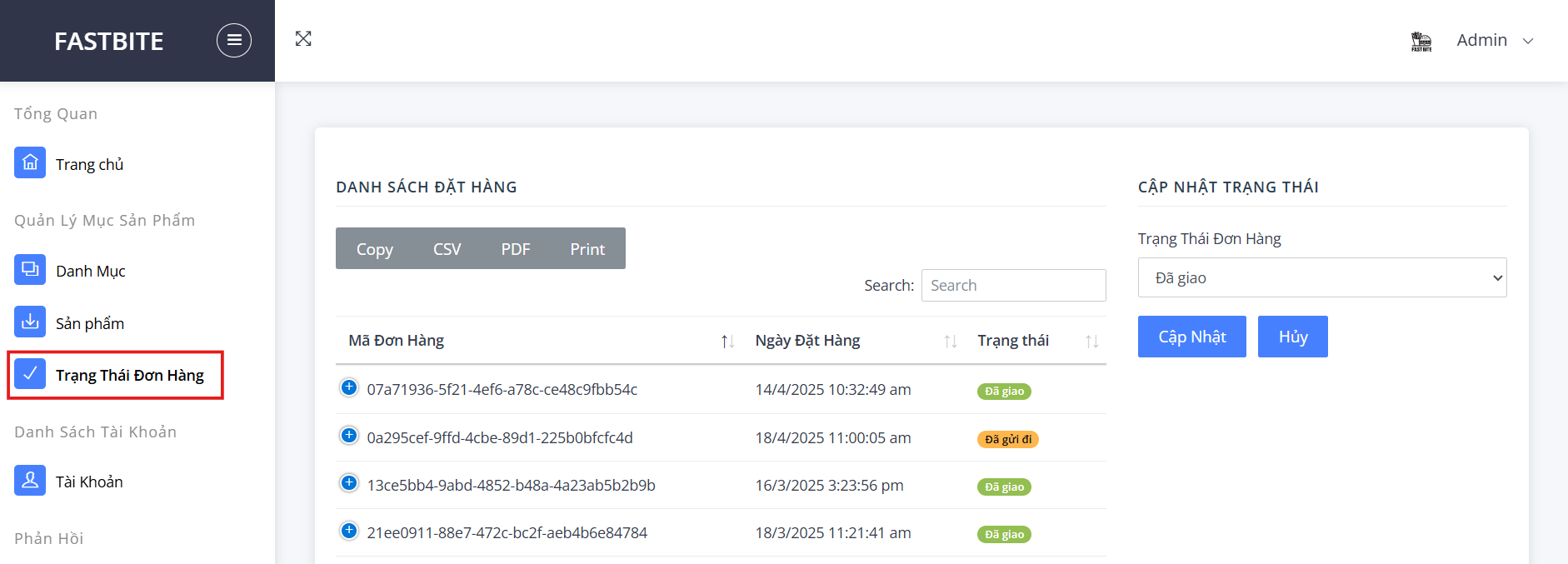


**Hình 4.3 Giao diện quản lý Sản Phẩm**

Form nhập liệu cho phép Admin nhập thông tin như tên sản phẩm, mô tả, giá tiền, số lượng, danh mục, và hình ảnh đại diện. Giao diện sẽ hiển thị cảnh báo khi số lượng sản phẩm thấp (≤ 5), giúp Admin kịp thời bổ sung hàng tồn kho. Danh sách sản phẩm trình bày thông tin rõ ràng với các cột như tên, giá, số lượng, trạng thái, và tùy chọn chỉnh sửa/xóa. Giao diện này giúp Admin duy trì thực đơn chính xác, đảm bảo khách hàng luôn có thông tin món ăn mới nhất.

4.2.4 Giao diện quản lý trạng thái đơn hàng

Giao diện quản lý trạng thái đơn hàng được thiết kế để hỗ trợ theo dõi và cập nhật trạng thái các đơn hàng trong hệ thống. Giao diện được chia thành hai phần chính: Danh sách đơn hàng để hiển thị thông tin chi tiết các đơn hàng dưới dạng bảng với tùy chọn chỉnh sửa, và Form cập nhật trạng thái cho phép Admin thay đổi trạng thái của đơn hàng.

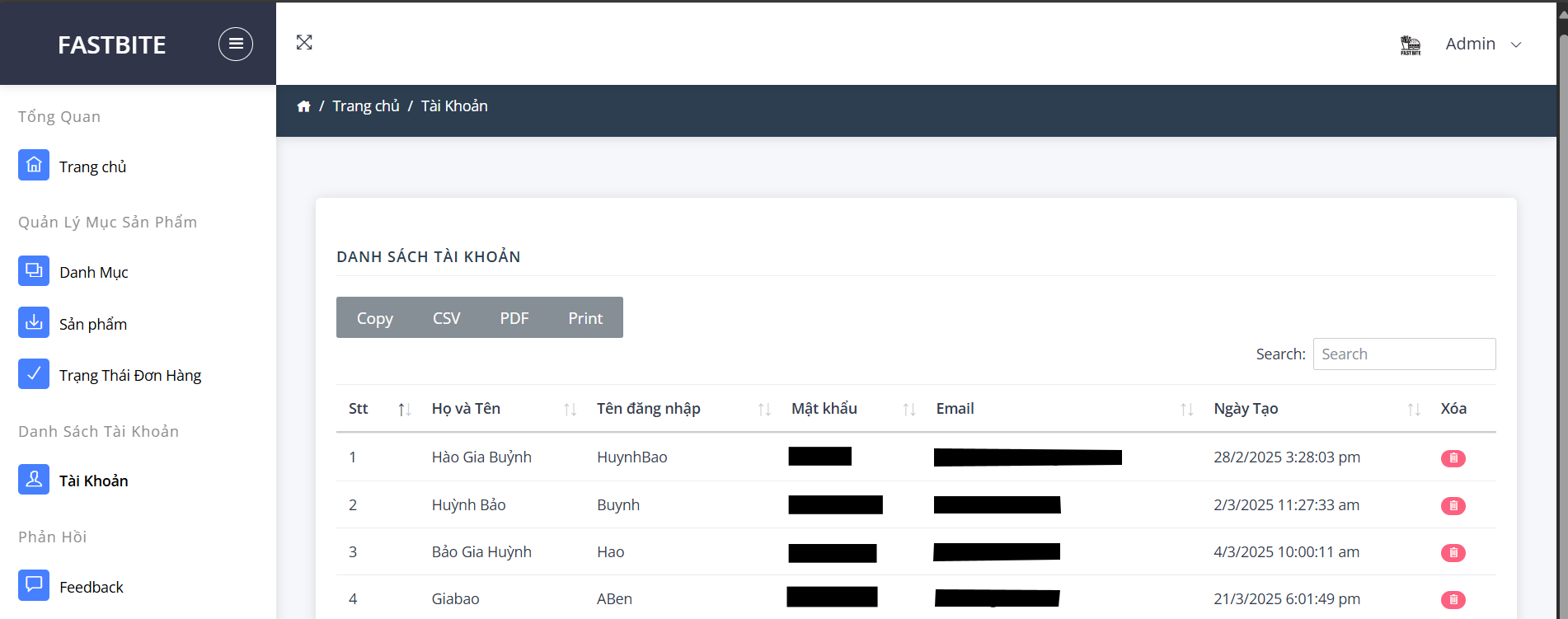


**Hình 4.4 Giao diện quản lý trạng thái đơn hàng**

Danh sách đơn hàng hiển thị thông tin như mã đơn hàng, ngày đặt, tên sản phẩm, tổng tiền, phương thức thanh toán, và trạng thái (Đang xử lý, Đã gửi đi, Đã giao, Đã hủy). Admin có thể chọn một đơn hàng để cập nhật trạng thái với các tùy chọn rõ ràng và nút xác nhận/hủy thao tác. Giao diện này đảm bảo quy trình giao hàng được quản lý chặt chẽ, mang lại sự tin cậy cho cả Admin và khách hàng.

4.2.5 Giao diện quản lý Tài khoản

Giao diện quản lý Tài khoản được thiết kế để hỗ trợ thực hiện các hành động quản lý thông tin tài khoản người dùng trong hệ thống. Giao diện tập trung vào việc hiển thị danh sách tài khoản dưới dạng bảng với tùy chọn xóa tài khoản.



**Hình 4.5 Giao diện quản lý Tài khoản**

Danh sách tài khoản hiển thị các thông tin như họ tên, tên đăng nhập, email, và ngày tạo, với nút xóa kèm thông báo xác nhận để đảm bảo thao tác an toàn. Giao diện này giúp Admin dễ dàng kiểm soát dữ liệu khách hàng, đảm bảo tính bảo mật và hỗ trợ quản lý hệ thống hiệu quả.

4.2.6 Giao diện quản lý Phản hồi

Giao diện quản lý Phản hồi được thiết kế để hỗ trợ Admin theo dõi và quản lý các phản hồi từ khách hàng trong hệ thống FastBite. Giao diện tập trung vào việc hiển thị danh sách phản hồi dưới dạng bảng dữ liệu.

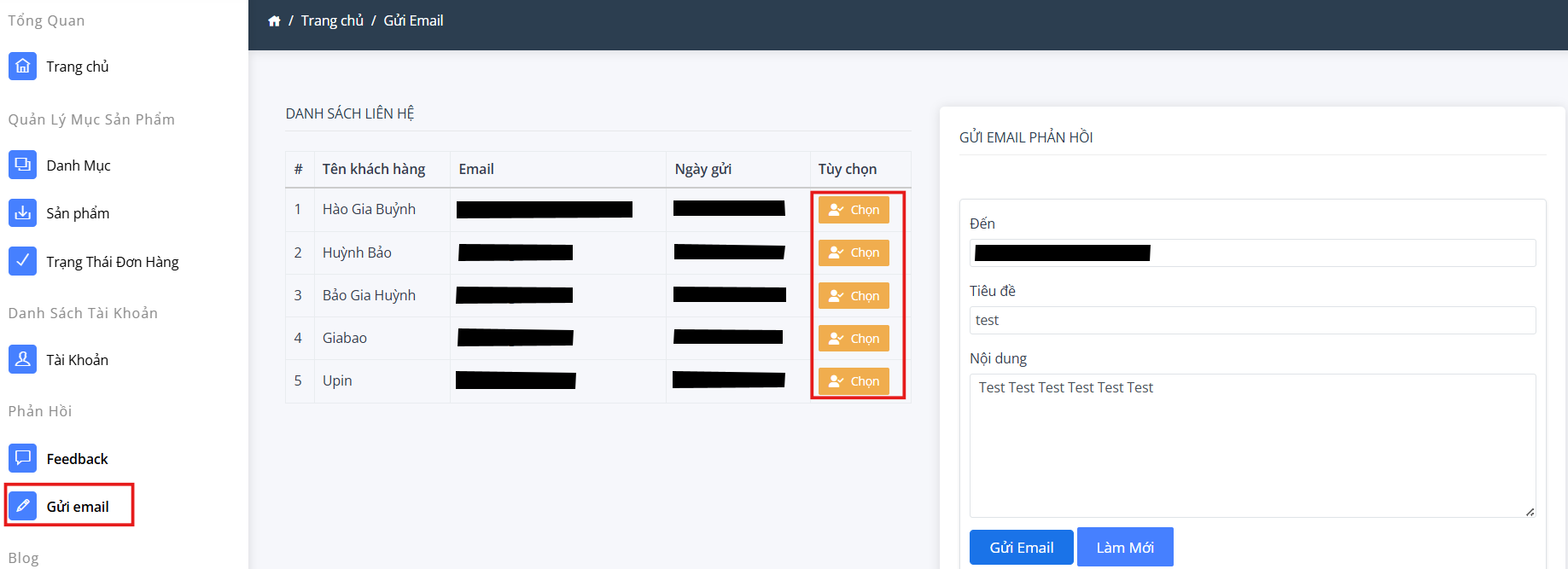


**Hình 4.6 Giao diện quản lý Phản hồi**

Danh sách phản hồi bao gồm các thông tin như tên người gửi, email, tiêu đề, nội dung, và ngày gửi, kèm nút xóa với thông báo xác nhận. Giao diện này giúp Admin dễ dàng nắm bắt nhanh chóng các phản hồi, ý kiến từ khách hàng, từ đó cải thiện chất lượng dịch vụ và nâng cao trải nghiệm người dùng.

4.2.7 Giao diện Gửi email

Giao diện gửi mail trong hệ thống FastBite được thiết kế để hỗ trợ Admin gửi email phản hồi đến khách hàng, đặc biệt là để trả lời các câu hỏi, phản hồi các góp ý của khách hàng đã gửi thông qua chức năng "Liên hệ" trên website.

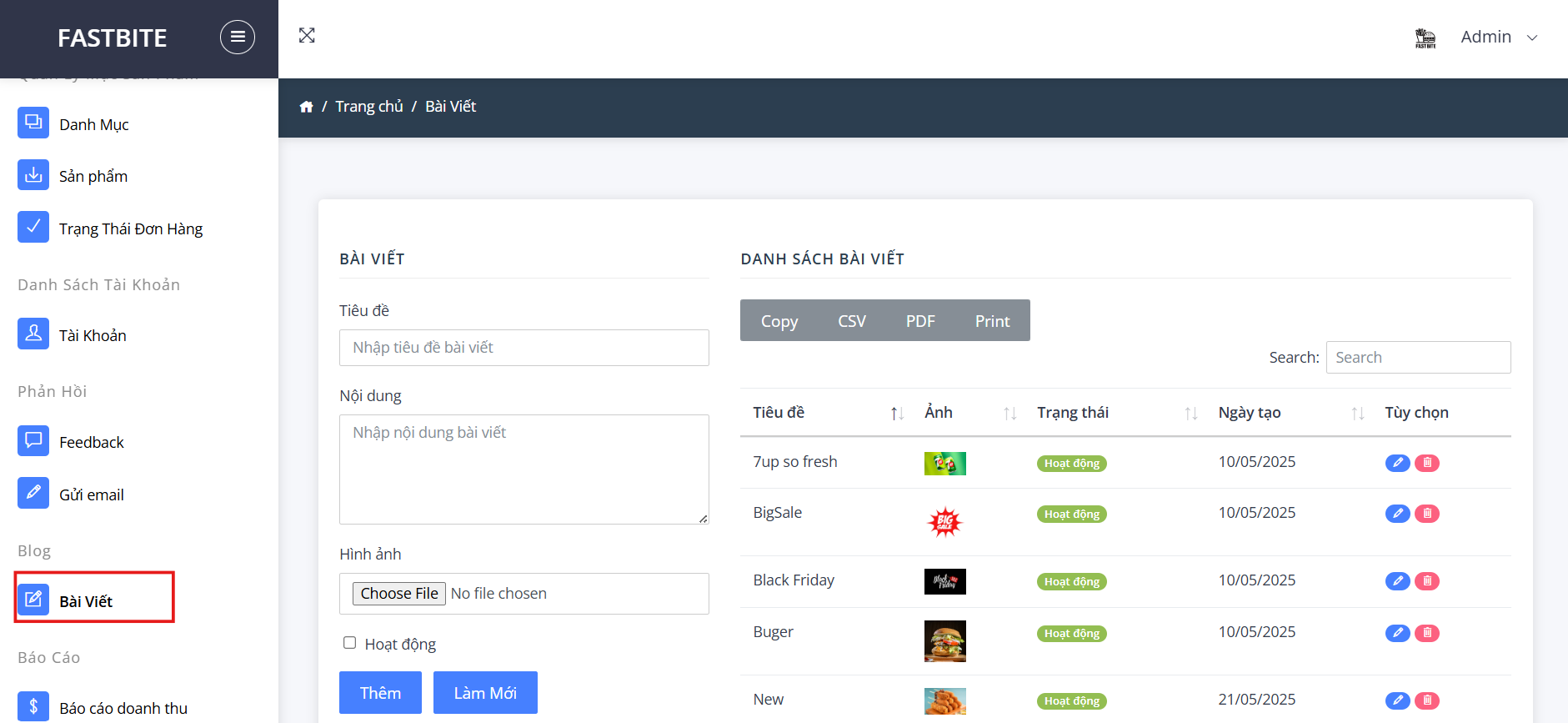


**Hình 4.7 Giao diện Gửi email**

Giao diện này giúp Admin xử lý các yêu cầu từ khách hàng một cách chuyên nghiệp, nhanh chóng, nhằm tăng trải nghiệm và giúp có thiện cảm đối với khách hàng, đồng thời cung cấp khả năng quản lý danh sách liên hệ và soạn email trực tiếp, nâng cao hiệu quả giao tiếp và chất lượng dịch vụ.

4.2.8 Giao diện Bài viết

Giao diện quản lý Bài Viết trong hệ thống FastBite, được thiết kế dành riêng cho Admin để quản lý các bài viết hoặc tin tức liên quan đến hệ thống, chẳng hạn như thông báo khuyến mãi, cập nhật thực đơn, hoặc các bài blog ẩm thực.

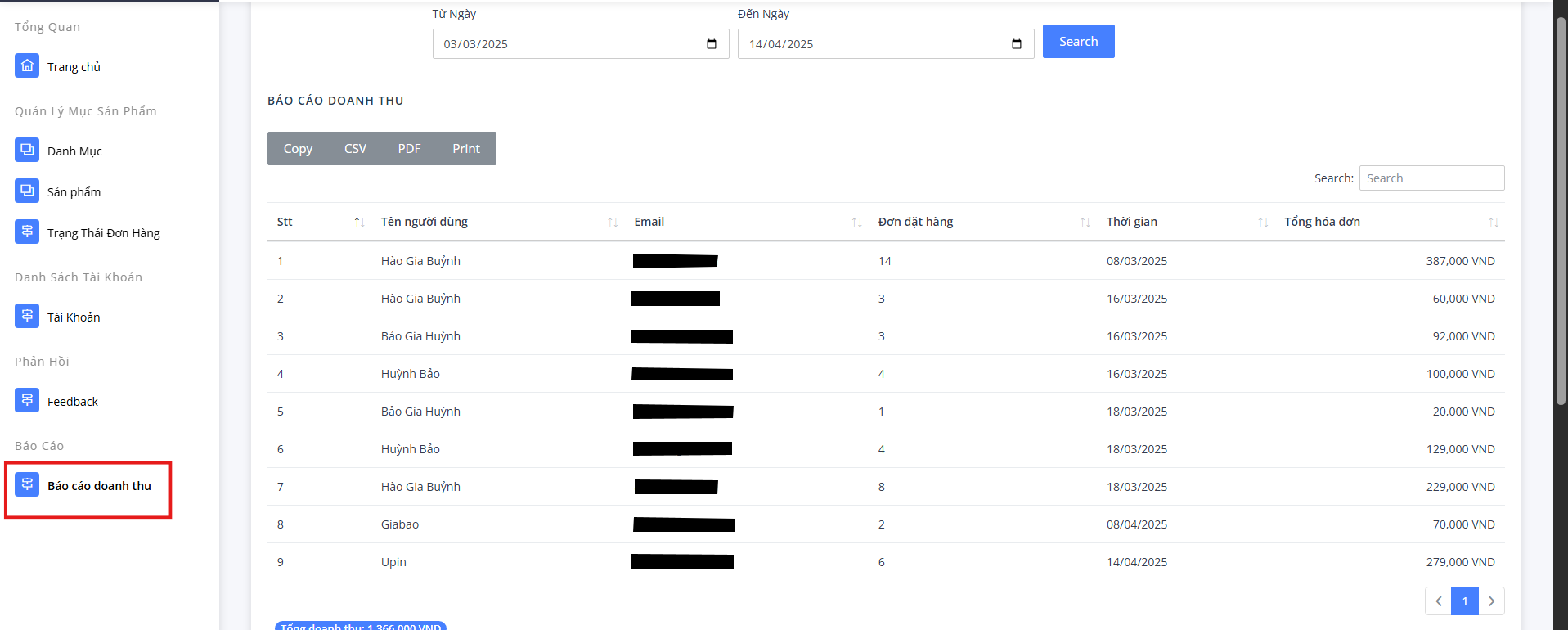


**Hình 4.8 Giao diện Bài Viết**

Giao diện bao gồm hai phần chính: Form đăng bài và danh sách bài viết. Khu vực nhập liệu cho phép Admin thêm, chỉnh sửa hoặc xóa thông tin bài viết. Danh sách bài viết hiển thị các bài hiện có dưới dạng bảng. Giao diện này giúp Admin dễ dàng quản lý nội dung, đảm bảo thông tin được cập nhật nhanh chóng và hiệu quả, góp phần thu hút sự quan tâm của khách hàng.

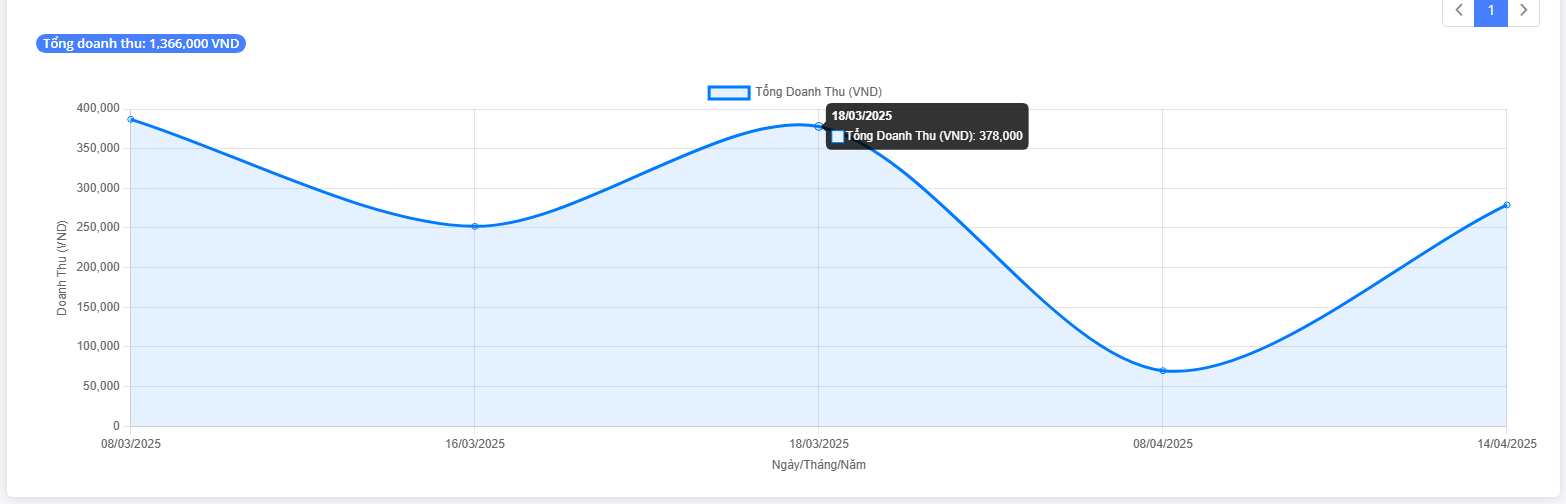
4.2.9 Giao diện Báo cáo Doanh thu

Giao diện Báo cáo Doanh thu được thiết kế để hỗ trợ Admin theo dõi và phân tích doanh thu của hệ thống trong một khoảng thời gian nhất định. Giao diện này cung cấp công cụ tìm kiếm theo ngày và hiển thị thông tin doanh thu chi tiết dưới dạng bảng, cùng với tổng doanh thu được tính toán tự động.



**Hình 4.9 Giao diện Báo cáo Doanh thu**

Cấu trúc Giao diện được chia thành hai phần chính: Bộ lọc thời gian và Danh sách báo cáo doanh thu. Bộ lọc thời gian cho phép Admin chọn các khoảng thời gian để xem doanh thu, trong khi danh sách báo cáo hiển thị thông tin như tên khách hàng, email, số đơn hàng, thời gian, và tổng hóa đơn.

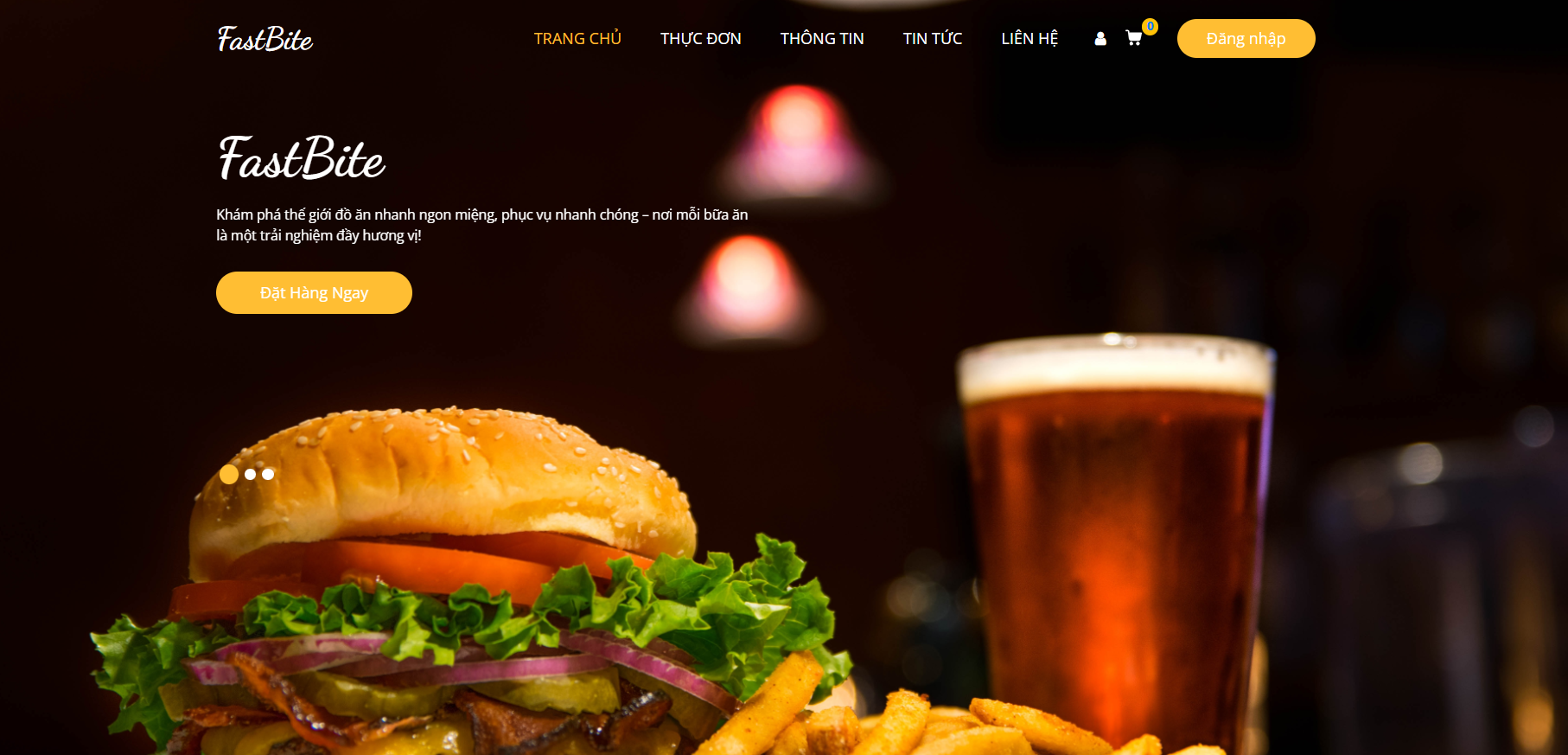


**Hình 4.10 Biểu đồ đường biểu diễn tổng doanh thu**

Biểu đồ doanh thu thể hiện sự biến động của doanh thu theo thời gian, mang đến một góc nhìn trực quan và dễ hiểu giúp cho người quản lý dễ dàng nhận diện các giai đoạn doanh thu nổi bật hoặc biến động, hỗ trợ hiệu quả trong việc phân tích và đưa ra định hướng chiến lược phát triển cho khoảng thời gian sắp tới.

4.3 Giao diện Khách hàng

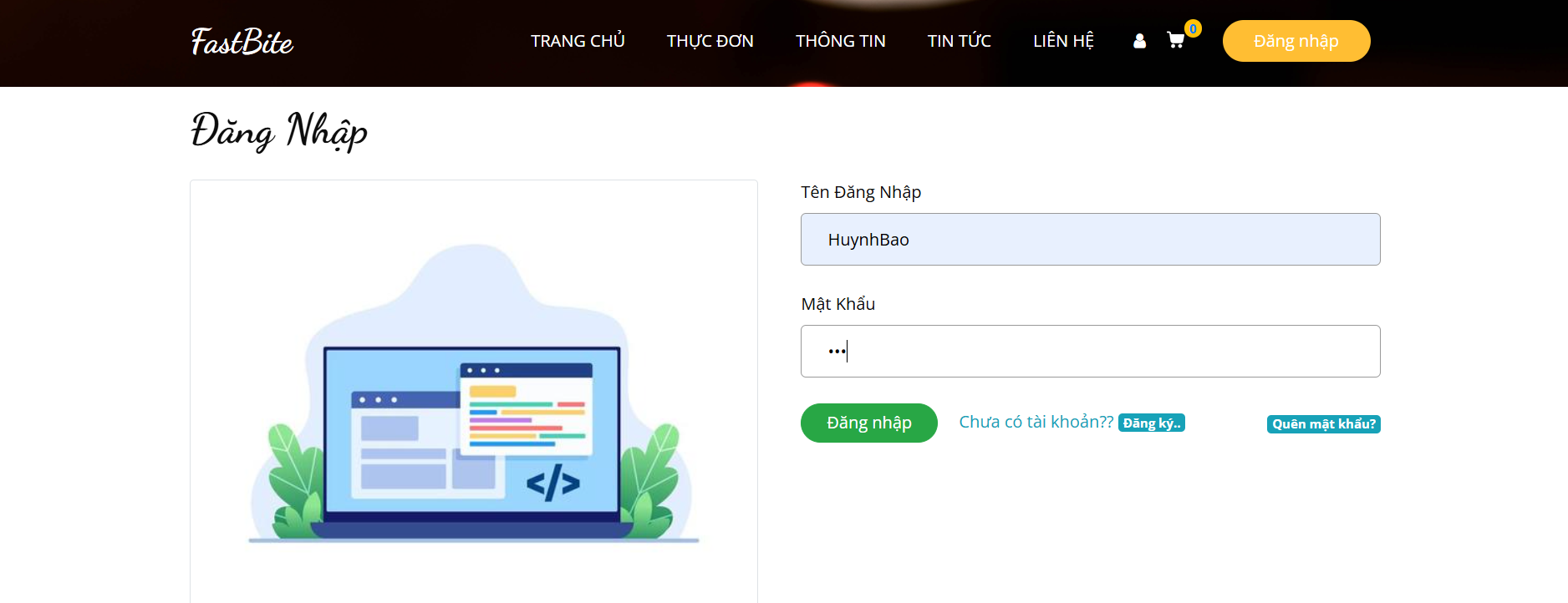
Giao diện dành cho khách hàng của website đặt đồ ăn FastBite được thiết kế để mang lại trải nghiệm thân thiện, trực quan và dễ sử dụng, đáp ứng nhu cầu đặt đồ ăn nhanh chóng và hiệu quả. Giao diện tập trung vào các chức năng chính như: tìm kiếm món ăn, thêm vào giỏ hàng, đặt hàng, theo dõi trạng thái đơn hàng và gửi phản hồi. Giao diện khách hàng không chỉ chú trọng tính thẩm mỹ mà còn đảm bảo hiệu quả sử dụng, mà còn tạo nền tảng cho các chức năng chính của hệ thống được hoạt động mượt mà, sẽ được trình bày chi tiết ở các mục tiếp theo.



**Hình 4.11 Giao diện trang chủ FastBite**

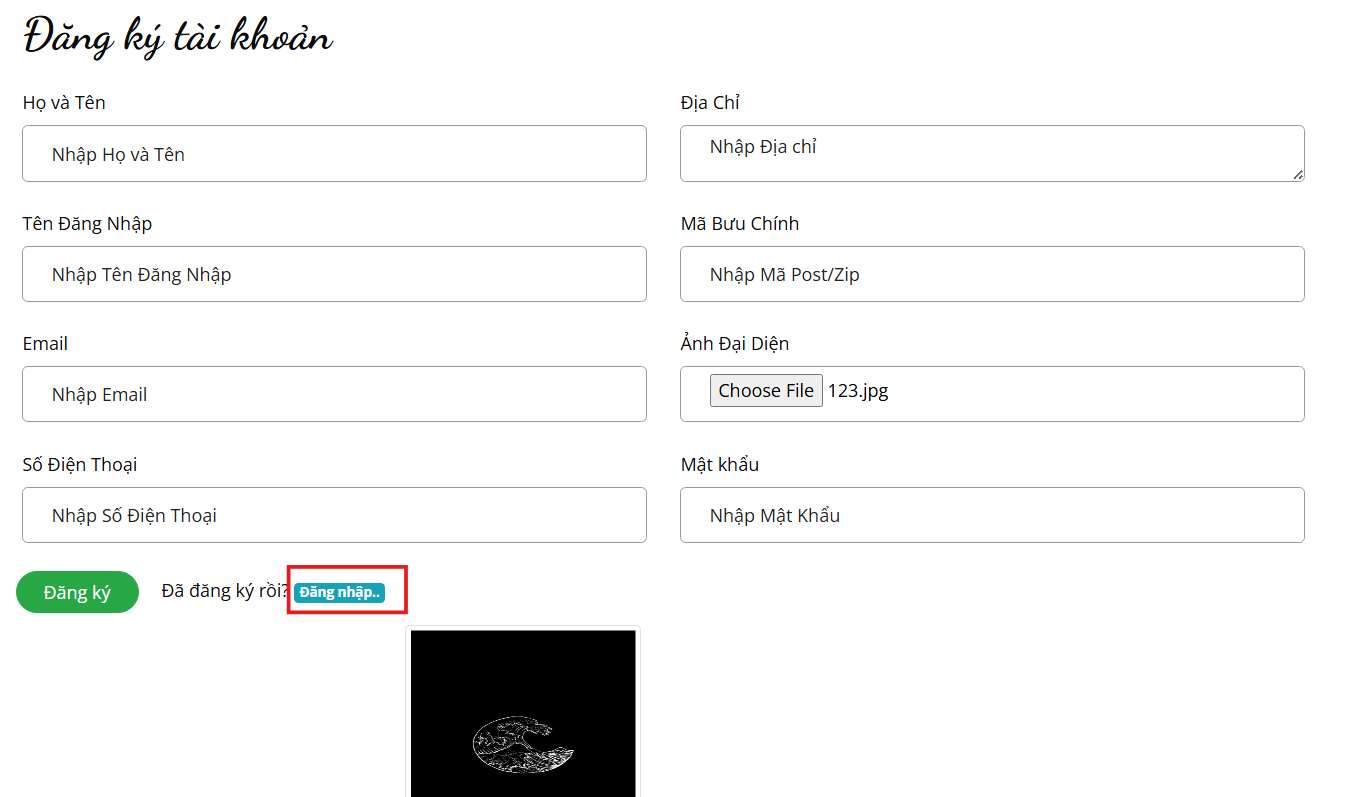
4.3.1 Giao diện Đăng nhập và Đăng ký

Giao diện Đăng nhập được thiết kế để hỗ trợ người dùng truy cập hệ thống FastBite một cách nhanh chóng. Giao diện bao gồm form nhập dữ liệu với các trường tên đăng nhập và mật khẩu, tích hợp kiểm tra dữ liệu để đảm bảo thông tin hợp lệ khi đăng nhập. Nút “Đăng nhập” xác thực thông tin bằng cách so sánh thông tin đăng nhập với dữ liệu trong bảng Users. Nếu xác thực thành công, hệ thống chuyển hướng người dùng đến trang chủ. Ngược lại, thông báo lỗi được hiển thị. Liên kết “Đăng ký” chuyển hướng đến giao diện đăng ký nếu chưa cho tài khoản.



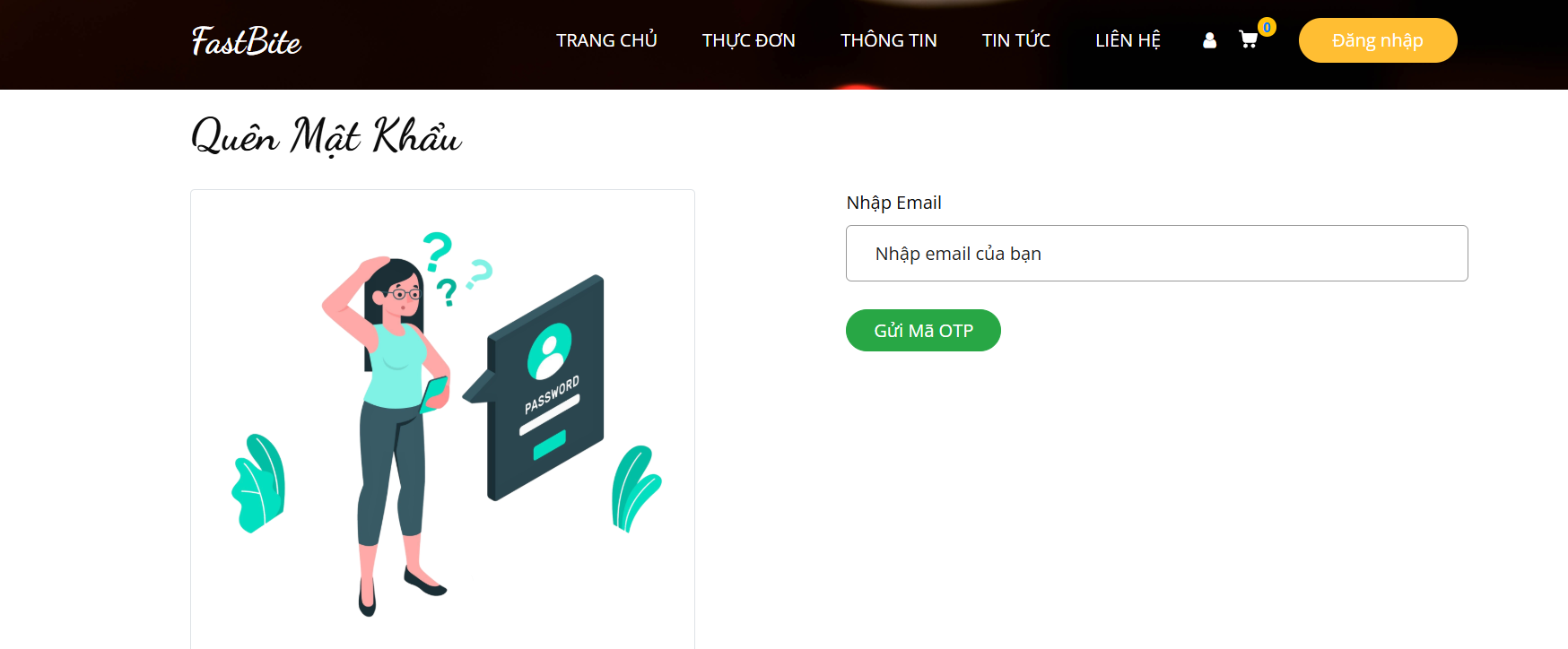
**Hình 4.12 Giao diện đăng nhập**

Giao diện Đăng ký cho phép người dùng tạo tài khoản mới với form gồm các trường: họ tên, tên đăng nhập, email,... Form tích hợp xác thực dữ liệu, kiểm tra định dạng email, độ dài mật khẩu và định dạng ảnh (.jpg, .jpeg, .png). Khi Đăng ký”, hệ thống sẽ lưu thông tin vào bảng Users và tải ảnh đại diện lên thư mục lưu trữ. Phần xem trước ảnh đại diện hiển thị hình ảnh đã chọn trước khi lưu, hỗ trợ người dùng kiểm tra. Liên kết “Đăng nhập” cho phép quay lại giao diện đăng nhập



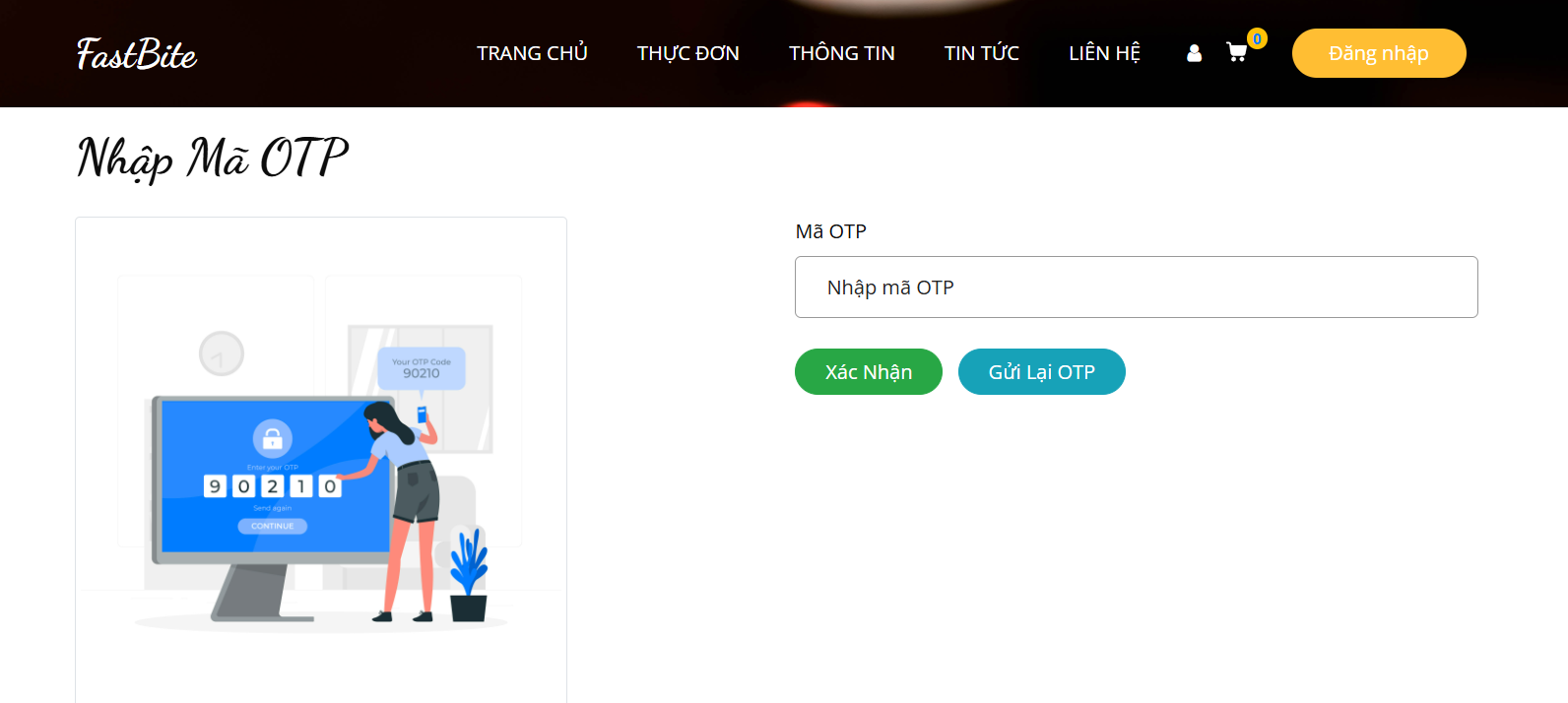
**Hình 4.13 Giao diện đăng ký**

Giao diện Quên Mật Khẩu được thiết kế để hỗ trợ người dùng khôi phục tài khoản khi quên mật khẩu hoặc tên đăng nhập đăng nhập thông qua email đã đăng ký trước đó thông qua gửi mã xác thực OTP.

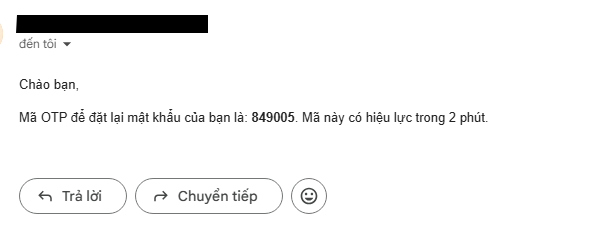


**Hình 4.14 Giao diện Quên mật khẩu**

Giao diện này cung cấp một cách tiếp cận đơn giản, cho phép người dùng dễ dàng bắt đầu quá trình khôi phục thông qua việc cung cấp thông tin liên hệ. Sau khi điền thông tin email hiện thống sẽ tự động kiểm tra tính hợp và gửi mã xác nhận đến email đã điền.

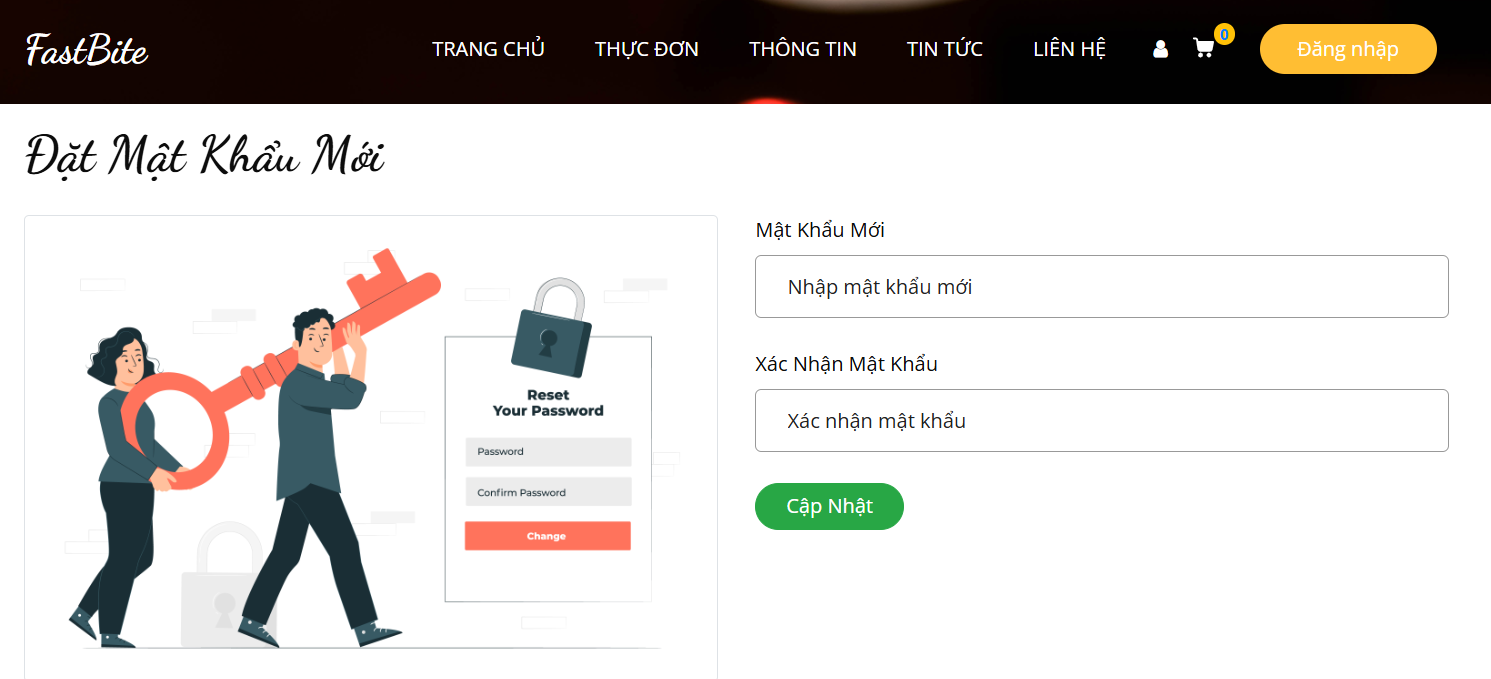


**Hình 4.15 Giao diện nhập mã OTP**



**Hình 4.16 Email gửi mã OTP**

Giao diện Nhập Mã Xác Nhận OTP được sử dụng để người dùng nhập mã OTP nhận được qua email, nhằm xác minh danh tính trước khi đặt lại mật khẩu. Giao diện này tập trung vào việc đảm bảo tính bảo mật và sự thuận tiện cho người dùng. Nếu mã chính xác, người dùng sẽ được chuyển đến bước đặt mật khẩu mới. Trong trường hợp mã không đúng hoặc mã hết hạn sử dụng, hệ thống sẽ hiển thị thông báo và cung cấp tùy chọn yêu cầu gửi lại mã mới.

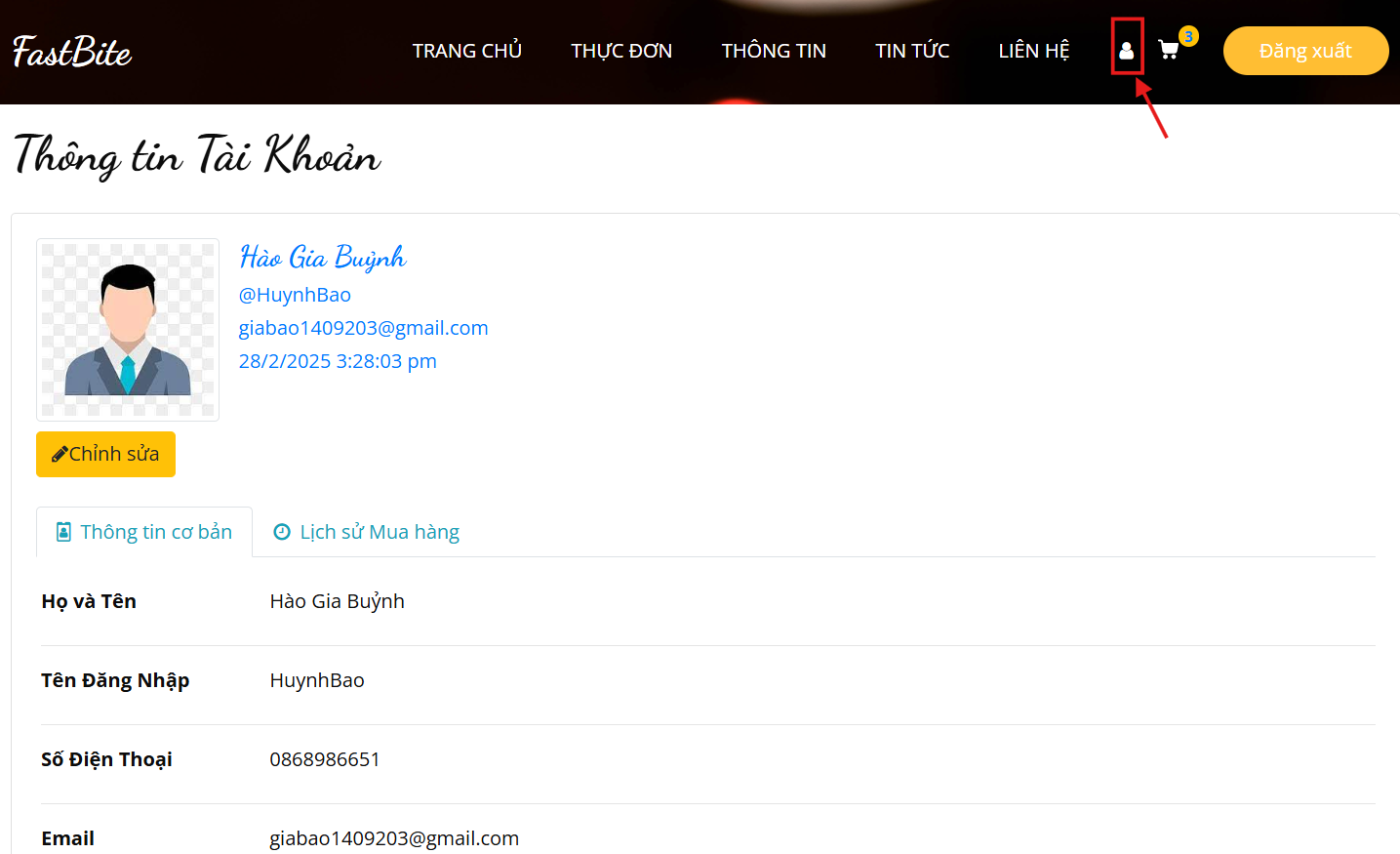


**Hình 4.17 Giao diện đặt mật khẩu mới**

Giao diện Đặt Mật Khẩu Mới cho phép người dùng tạo mật khẩu mới sau khi xác minh danh tính thành công. Đây là bước cuối cùng trong quy trình khôi phục tài khoản, đảm bảo người dùng có thể tiếp tục sử dụng hệ thống với thông tin đăng nhập mới. Sau khi gửi, hệ thống sẽ lưu mật khẩu mới và thông báo thành công, đồng thời chuyển hướng người dùng về trang đăng nhập để sử dụng tài khoản.

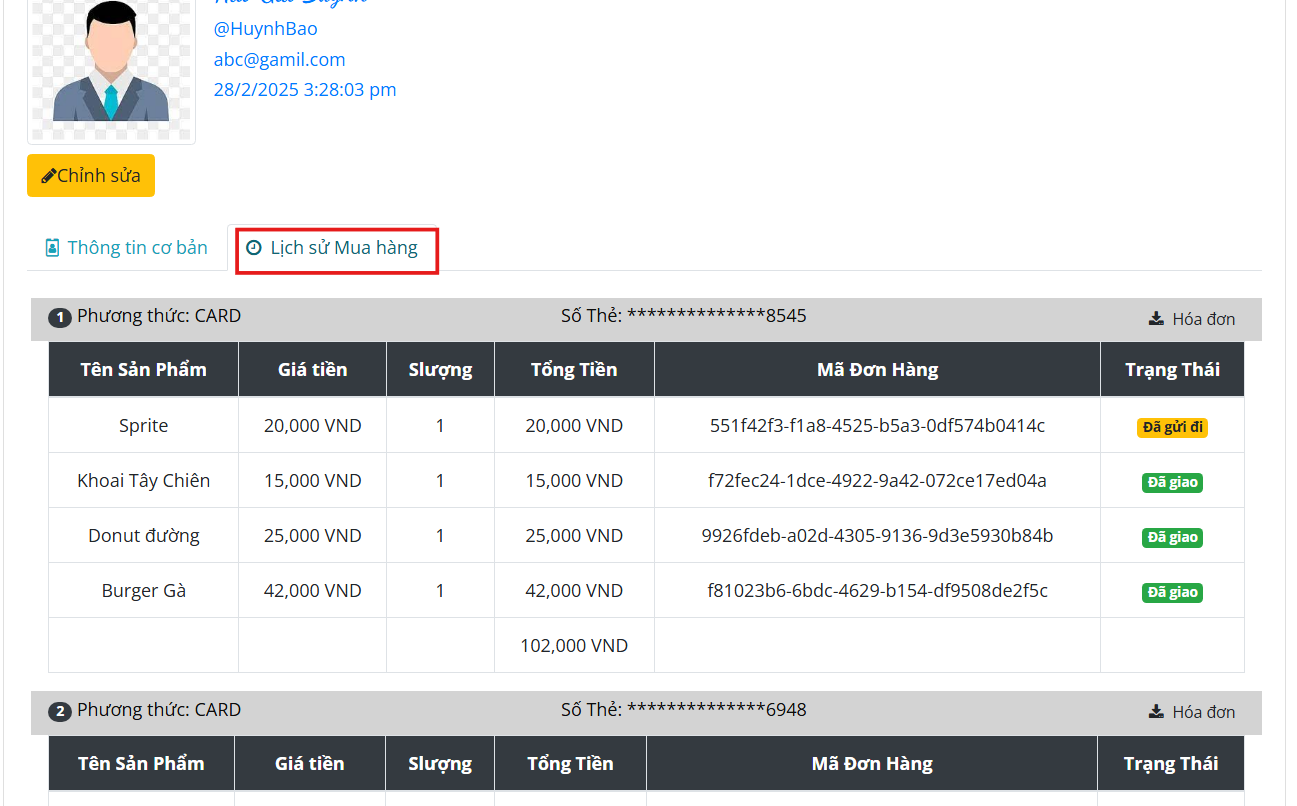
4.3.2 Giao diện tài khoản

Giao diện Tài khoản được thiết kế để hỗ trợ người dùng quản lý thông tin cá nhân và theo dõi chi tiết lịch sử mua hàng của mình trên hệ thống FastBite. Giao diện được chia thành hai tab chính: Thông tin cơ bản và Lịch sử mua hàng. Chức năng hoạt động chính của 2 tab:



**Hình 4.18 Giao diện thông tin tài khoản**

Tab Thông tin tài khoản hiển thị thông tin chi tiết tài khoản người dùng, bao gồm: họ tên, tên đăng nhập, email,... Dữ liệu được truy xuất từ bảng Users thông qua Stored Procedure User\_Crud để hiển thị thông tin tài khoản người dùng tương ứng. Hình ảnh đại diện tài khoản người dùng nếu không có ảnh đại diện, hệ thống sẽ hiển thị ảnh mặc định nếu không tìm thấy file ảnh. Nút “Chỉnh sửa” chuyển hướng đến trang đăng ký với tham số id người dùng tương ứng, cho phép cập nhật thông tin người dùng.

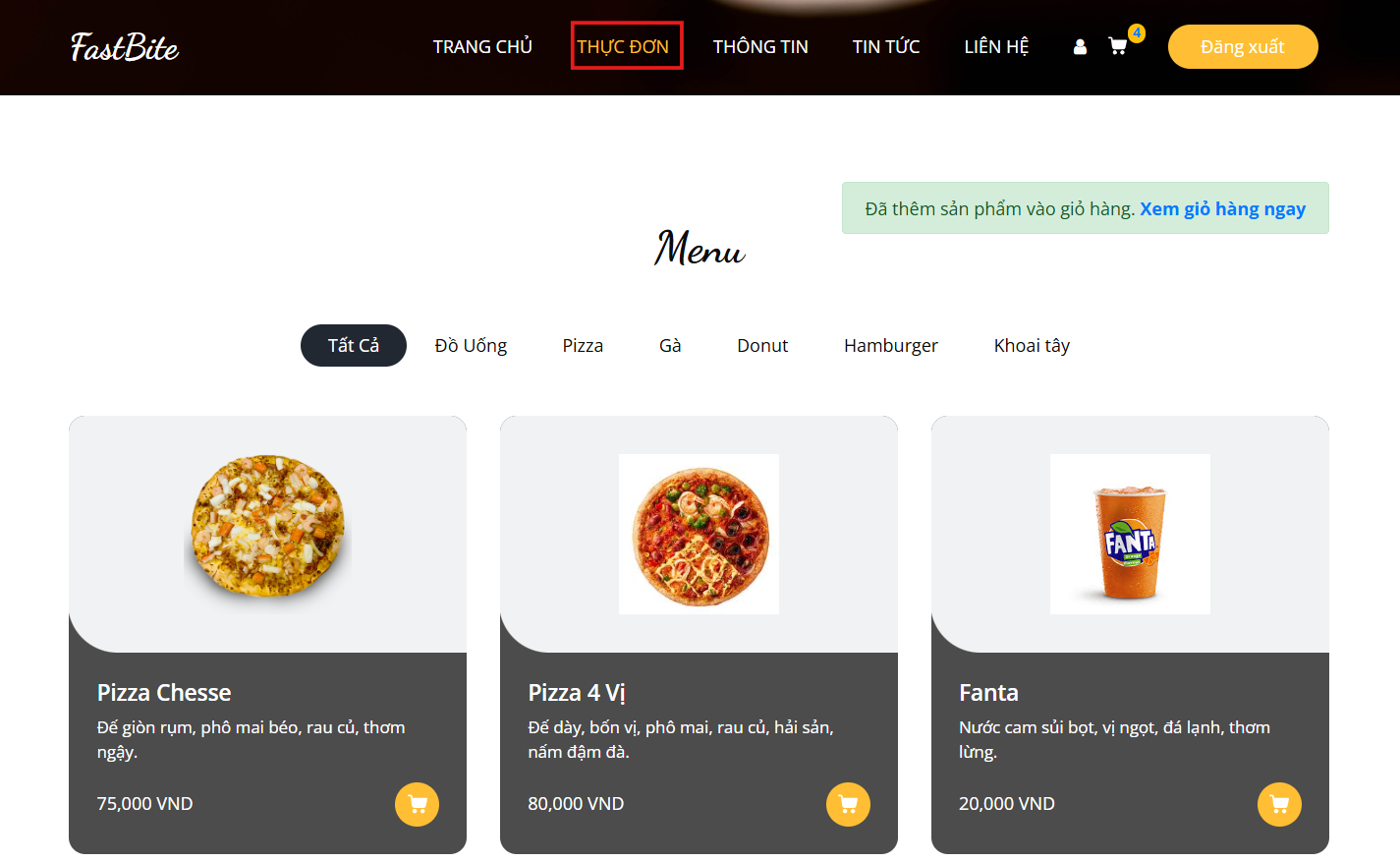


**Hình 4.19 Giao diện lịch sử mua hàng**

Tab Lịch sử mua hàng hiển thị danh sách tất cả giao dịch của người dùng, lấy dữ liệu từ bảng Payment và Orders. Mỗi giao dịch đều hiển thị phương thức thanh toán, số thẻ tín dụng (nếu có), và nút tải hóa đơn ứng với id giao dịch PaymentId của người dùng. Chi tiết đơn hàng được hiển thị chi tiết bên dưới, bao gồm tên sản phẩm, giá, số lượng, tổng tiền, mã đơn hàng và trạng thái đơn hàng. Dữ liệu chi tiết được truy xuất qua Stored Procedure Invoice dựa trên PaymentId.

4.3.3 Giao diện Menu sản phẩm

Giao diện Menu sản phẩm của hệ thống FastBite được thiết kế để hiển thị danh sách các món ăn, cho phép người dùng tìm kiếm các món ăn thông qua bộ lọc theo danh mục sản phẩm và thêm sản phẩm vào giỏ hàng một cách thuận tiện. Dưới đây là các chức năng chính cho các hoạt động:



**Hình 4.20 Giao diện Menu sản phẩm**

Bộ lọc danh mục hiển thị danh sách các danh mục sản phẩm dưới dạng danh sách, cho phép người dùng lọc món ăn theo danh mục (ví dụ: Pizza, Hamburger, Gà). Dữ liệu danh mục được truy xuất từ bảng Categories thông qua Stored Procedure Category\_Crud, chỉ lấy các danh mục đang hoạt động. Khi người dùng muốn tìm kiếm món ăn trên bộ lọc, bộ lọc sẽ ẩn/hiện các sản phẩm tương ứng dựa tên danh mục, đảm bảo trải nghiệm lọc mượt mà mà không cần tải lại trang.

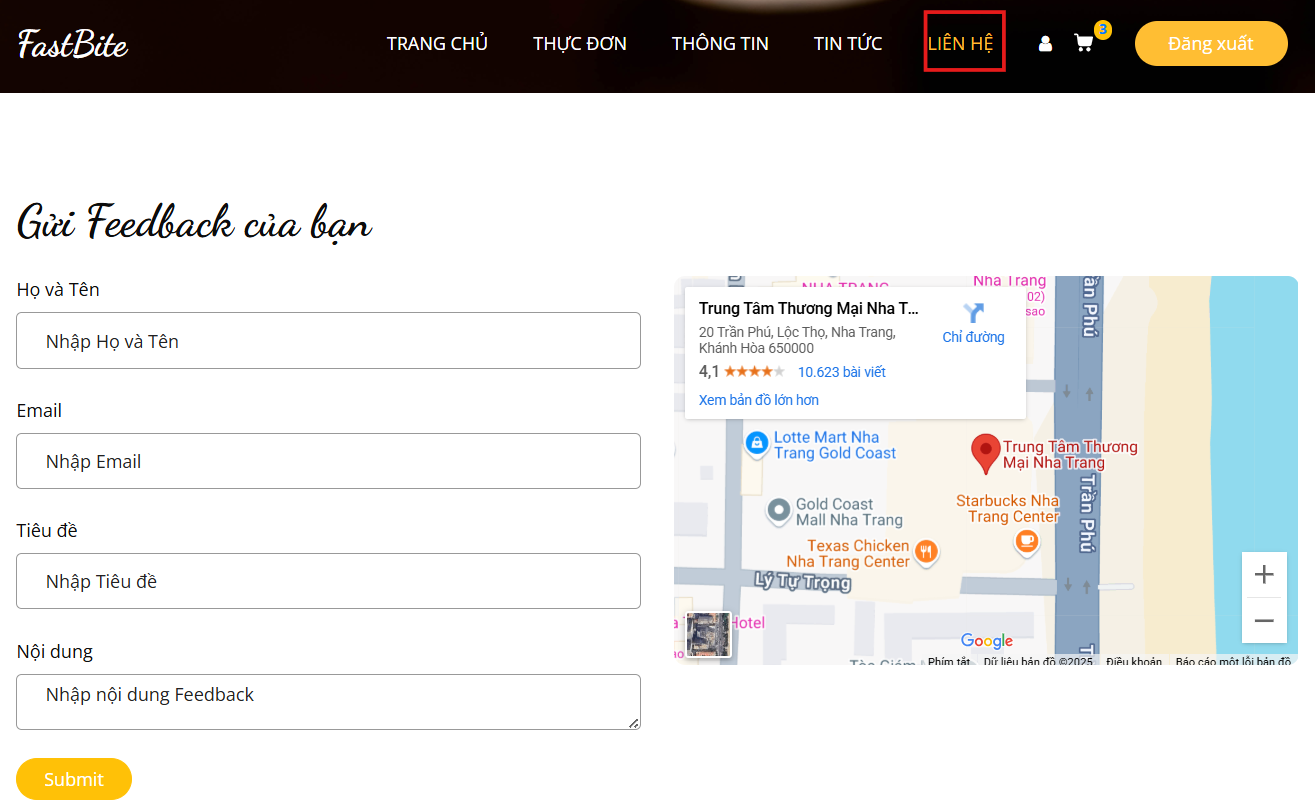
Danh sách sản phẩm hiển thị các món ăn dưới dạng lưới. Dữ liệu sản phẩm được truy xuất từ bảng Products thông qua Stored Procedure Product\_Crud. Mỗi sản phẩm hiển thị các thông tin: hình ảnh (ảnh mặc định nếu không có ảnh), tên sản phẩm, mô tả, giá, và nút “Thêm vào giỏ hàng”.

Chức năng thêm vào giỏ hàng. Hệ thống kiểm tra đã đăng nhập chưa. Nếu chưa đăng nhập, hệ thống sẽ chuyển người dùng đến trang đăng nhập. Nếu đã đăng nhập, hệ thống sẽ kiểm tra sản phẩm đã có trong giỏ hàng chưa. Nếu sản phẩm chưa có thì số lượng mặc định trong giỏ bằng 0, Nếu thêm sản phẩm vào giỏ hàng với Id sản phẩm, số lượng sản phẩm được thêm mặc định tăng lên là 1.

Sau khi thêm thành công, thông báo “Đã thêm sản phẩm vào giỏ hàng”. Thông báo cũng chứa đường dẫn liên kết đến giỏ hàng để thuận tiện cho việc kiểm tra giỏ hàng. Thông báo hiển thị trạng thái của hành động thêm giỏ hàng, đảm bảo người dùng biết được đã thêm sản phẩm thành công.

4.3.4 Giao diện Liên hệ

Giao diện Liên hệ của hệ thống FastBite được thiết kế để hỗ trợ khách hàng gửi phản hồi, câu hỏi hoặc đánh giá về dịch vụ, đảm bảo trải nghiệm người dùng. Giao diện tích hợp với Master Page để duy trì tính đồng nhất về bố cục và phong cách với các trang khác dành cho khách hàng. Giao diện bao gồm một biểu mẫu và một khu vực bản đồ vị trí cửa hàng, mang lại sự trực quan và dễ sử dụng. Chức năng hoạt động chính:

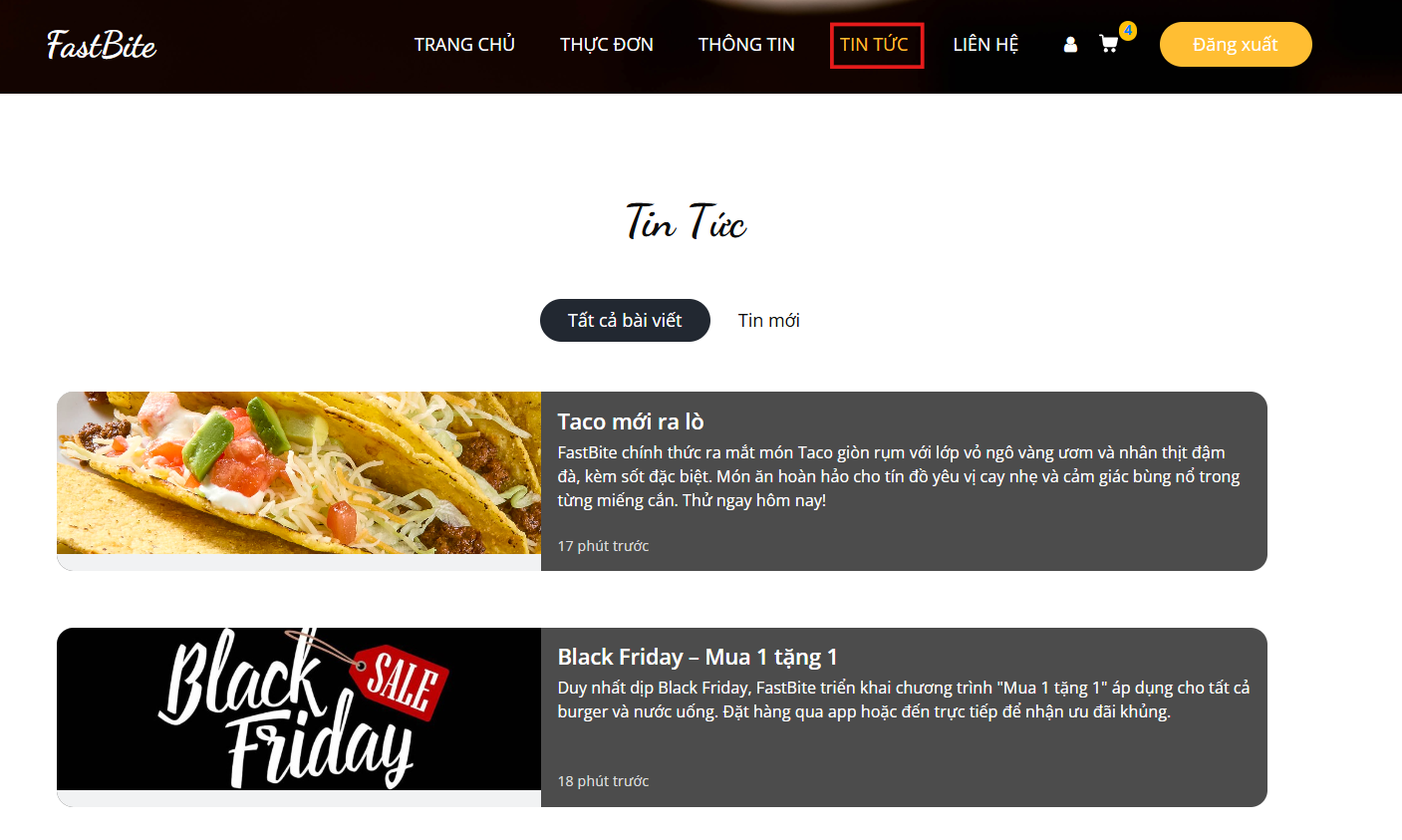


**Hình 4.21 Giao diện Liên hệ**

Biểu mẫu gửi phản hồi được thiết kế với các trường nhập dữ liệu để thu thập thông tin từ người dùng, bao gồm: Họ và Tên, Email, Tiêu đề, Nội dung. Mỗi trường đều được đảm bảo người dùng không để trống dữ liệu. Thông báo lỗi hiển thị bằng màu đỏ với nội dung "Không được để trống" nếu vi phạm, đồng thời tự động đưa con trỏ đến trường bị lỗi, nâng cao trải nghiệm người dùng.

4.3.5 Giao diện Tin Tức

Giao diện Tin Tức dành cho khách hàng trên website FastBite, nơi hiển thị thông tin các bài viết hoặc tin tức về các chương trình khuyến mãi, món ăn mới, hoặc sự kiện liên quan sắp tới.

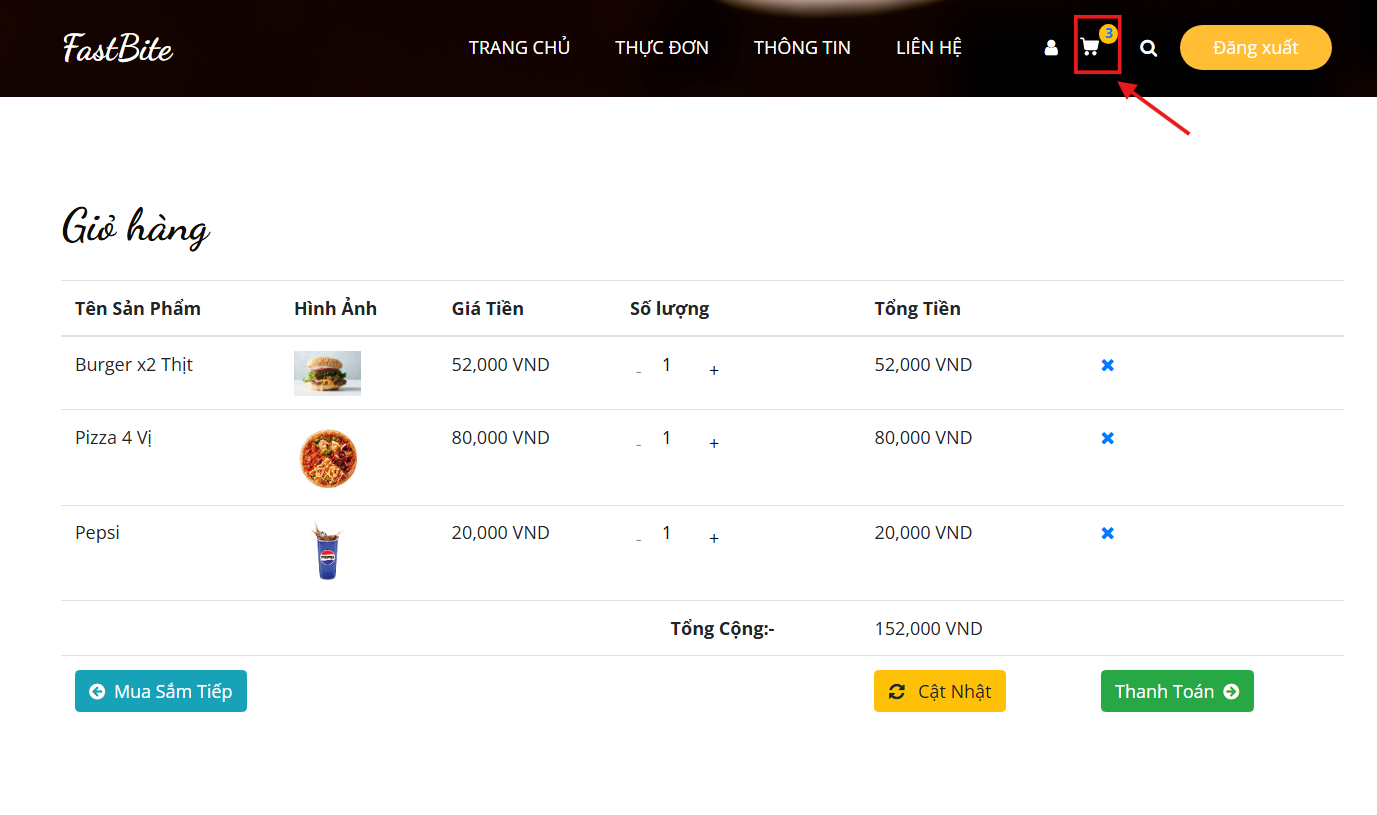


**Hình 4.22 Giao diện Tin Tức**

Giao diện trình bày danh sách bài viết theo dạng lưới hoặc danh sách, mỗi bài gồm tiêu đề, hình ảnh đại diện, tóm tắt nội dung và ngày đăng. Người dùng có thể sử dụng bộ lọc để tìm kiếm tin tức mới nhất hoặc tất cả các tin. Với thiết kế trực quan, hỗ trợ tốt trên nhiều thiết bị, giao diện này mang đến trải nghiệm đọc tin tức hấp dẫn và thuận tiện, giúp khách hàng dễ dàng cập nhật thông tin từ hệ thống.

4.3.6 Giao diện Giỏ hàng

Giao diện giỏ hàng hỗ trợ người dùng quản lý các sản phẩm đã thêm trong quá trình sử dụng dịch vụ đặt món ăn trực tuyến. Đây là một phần quan trọng trong hệ thống người dùng, cung cấp cái nhìn tổng quan và chi tiết về các món hàng mà người dùng đã thêm vào giỏ và hiển thị thông tin sản phẩm. Giao diện không chỉ dừng lại ở việc hiển thị thông tin mà còn cho phép người dùng thực hiện các chức năng chỉnh sửa số lượng, xóa, tiếp tục mua sắm nếu muốn thêm món, hoặc tiến hành thanh toán khi đã sẵn sàng. Với thiết kế trực quan và thân thiện, giao diện này giúp người dùng dễ dàng kiểm soát giỏ hàng, đảm bảo quá trình mua sắm diễn ra thuận lợi và hiệu quả.



**Hình 4.23 Giao diện giỏ hàng**

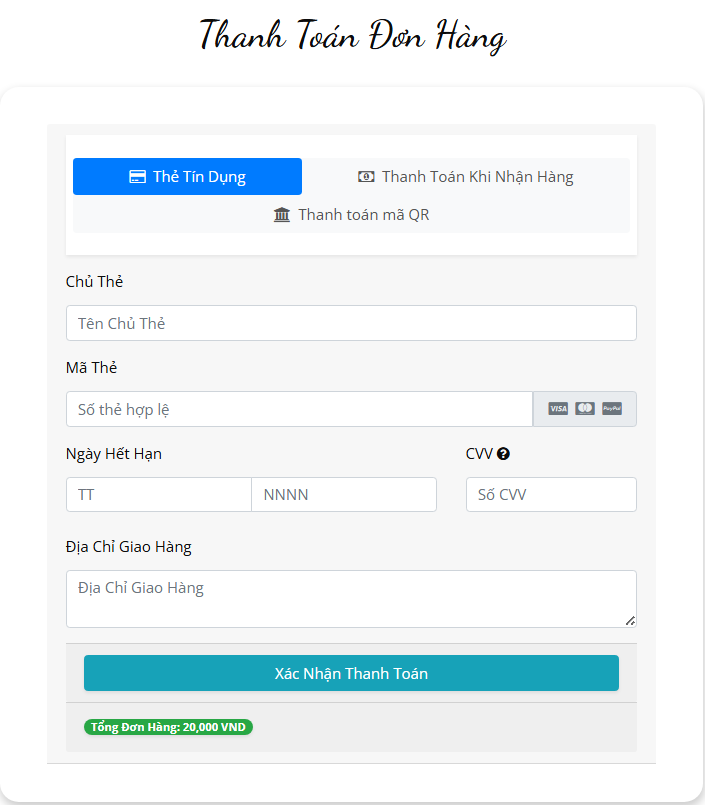
Giao diện giỏ hàng hoạt động thông qua các bước và chức năng được thiết kế để đáp ứng nhu cầu của người dùng, từ việc hiển thị sản phẩm đến xử lý các thao tác quản lý giỏ hàng và thanh toán:

* **Hiển thị danh sách sản phẩm**: Khi người dùng truy cập vào trang giỏ hàng, hệ thống sẽ truy xuất dữ liệu từ CSDL dựa trên thông tin tài khoản của người dùng để lấy danh sách các sản phẩm đã được thêm vào giỏ. Danh sách hiển thị các cột thông tin cụ thể bao gồm: tên sản phẩm, hình ảnh, giá tiền, số lượng, tổng tiền của từng sản phẩm, tùy chọn xóa nếu không muốn mua nữa.
* **Tính tổng giá trị đơn hàng**: Tổng tiền của từng sản phẩm được tính như sau ( Giá tiền *x* Số lượng). Tổng giá trị đơn hàng được tính bằng cách cộng tất cả tổng tiền của các sản phẩm trong giỏ và hiển thị ở cuối bảng.
* **Hỗ trợ chuyển hướng**: Giao diện cung cấp các nút điều hướng hỗ trợ người dùng trong quá trình mua sắm. Nút quay lại trang menu để chọn thêm sản phẩm nếu họ muốn bổ sung món ăn hoặc đồ uống. Nút thanh toán đưa người dùng đến trang thanh toán nếu giỏ hàng hợp lệ.
* **Xử lý trường hợp giỏ hàng trống**: Nếu người dùng chưa thêm sản phẩm nào vào giỏ hoặc đã xóa hết sản phẩm, giao diện sẽ hiển thị thông báo "Giỏ hàng trống" kèm theo một đường dẫn để người dùng có thể quay lại trang menu và tiếp tục mua sắm. Điều này giúp tránh trường hợp người dùng cảm thấy bối rối khi không thấy sản phẩm nào trong giỏ.
* **Xử lý lỗi và thông báo**: Trong trường hợp có sản phẩm trong giỏ đã hết không đủ để đáp ứng số lượng yêu cầu, hệ thống sẽ hiển thị thông báo lỗi ngay trên giao diện, VD: "Sản phẩm 'Pizza 4 Vị' đã hết hàng!". Người dùng sẽ không thể tiếp thanh toán cho đến khi điều chỉnh giỏ hàng.
* **Cập nhật số lượng sản phẩm trong giỏ trên thanh điều hướng**: Bên cạnh giao diện giỏ hàng, hệ thống cũng cập nhật số lượng sản phẩm trong giỏ trên thanh điều hướng (phía trên cùng của trang). Ví dụ, nếu thêm vào giỏ hàng 3 sản phẩm, biểu tượng giỏ hàng trên thanh điều hướng sẽ hiển thị số "3" để người dùng biết tổng số món hiện có trong giỏ hàng.

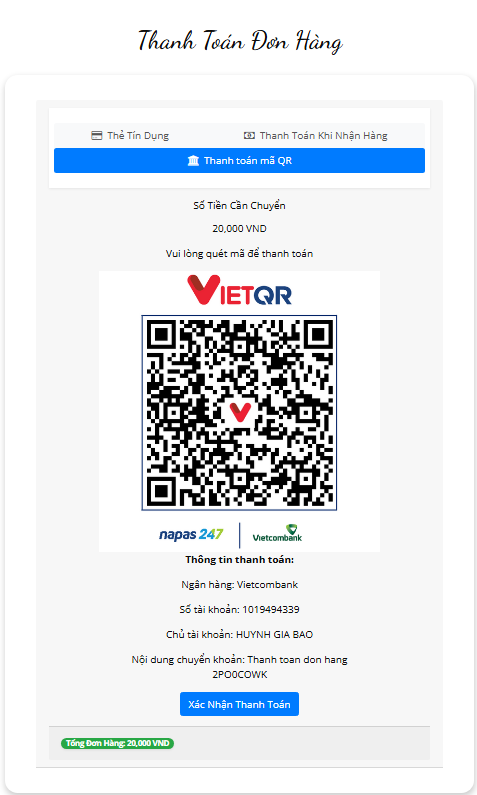
Giao diện giỏ hàng trên hệ thống FastBite không chỉ mang lại sự tiện lợi trong việc quản lý sản phẩm mà còn đảm bảo tính minh bạch và dễ sử dụng, giúp người dùng có trải nghiệm mua sắm trực tuyến mượt mà và hiệu quả.

4.3.7 Giao diện thanh toán

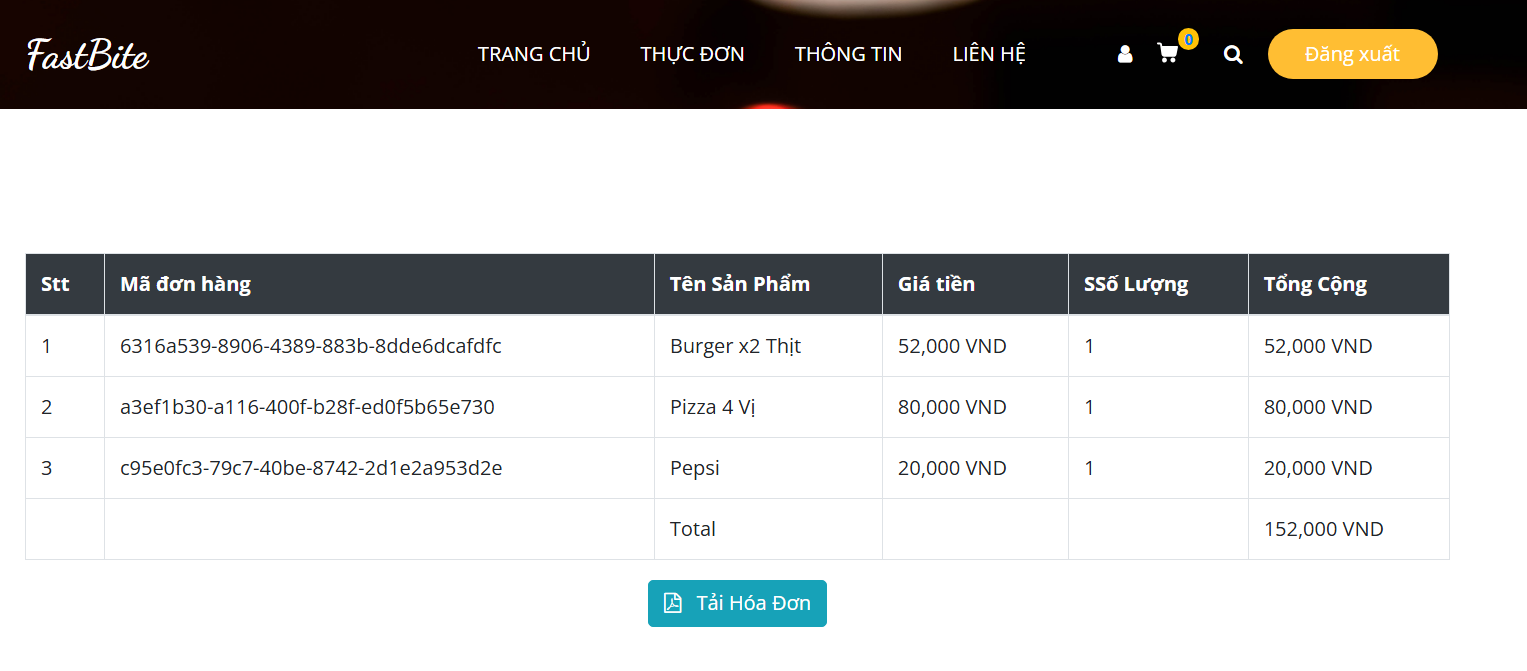
Giao diện thanh toán là bước cuối cùng trong quy trình đặt hàng, cho phép người dùng lựa chọn các phương thức thanh toán và hoàn tất giao dịch. Giao diện tập trung vào ba phương thức thanh toán chính: Thanh toán bằng thẻ tín dụng, Thanh toán khi nhận hàng (COD) và Thanh toán mã QR đáp ứng nhu cầu của người dùng. Đồng thời, giao diện giúp lưu trữ thông tin giao dịch vào cơ sở dữ liệu, hỗ trợ quản lý đơn hàng và tạo hóa đơn, đáp ứng yêu cầu quản lý hệ thống của FastBite.



**Hình 4.24 Giao diện thanh toán**



**Hình 4.25 Giao diện thanh toán mã QR**



**Hình 4.26** **Giao diện Hóa đơn**



**Hình 4.27 File Hóa đơn mua hàng**

Giao diện hoạt động như sau: Người dùng có thể chọn một trong ba phương thức thanh toán bằng: "Thẻ Tín Dụng", "Thanh Toán Khi Nhận Hàng" hoặc "Thanh Toán mã QR"

* Nếu chọn "Thẻ Tín Dụng", người dùng phải nhập thông tin thẻ và địa chỉ giao hàng, ở mỗi trường nhập thông tin để kiểm tra kiểu dữ liệu để đảm bảo đều là thông tin hợp lệ chẳng hạn số thẻ phải đủ 16 chữ số, CVV 3 chữ số,… Sau đó hệ thống sẽ lưu thông tin thanh toán vào bảng Payment, sau đó lấy thông tin từ giỏ hàng để cập nhật lại số lượng sản phẩm và xóa hết sản phẩm trong giỏ hàng và cuối cùng lưu thông tin đơn hàng vào trong bảng Order.
* Nếu chọn “Thanh toán khi nhận hàng”, cũng tương tự thanh toán bằng thẻ mà chỉ khác ở chỗ chỉ cần nhập thông tin địa chỉ nhận hàng.
* Nếu chọn "Thanh Toán mã QR" hệ thống sẽ tạo tự động một mã QR được tạo động thông qua API VietQR, chứa thông tin thanh toán bao gồm số tài khoản ngân hàng, số tiền cần thanh toán, và tạo nội dung chuyển khoản ngẫu nhiên (VD: "Thanh toan don hang ABCXYZ12"). Người dùng có thể quét mã QR bằng ứng dụng ngân hàng hoặc ví điện tử để thực hiện thanh toán. Sau khi thanh toán xong, nhấn nút “Xác Nhận Thanh Toán”, hệ thống sẽ lưu thông tin thanh toán vào bảng Payment.

Hệ thống kiểm tra nếu có lỗi (VD: sản phẩm trong kho không đủ), giao dịch sẽ trả về và hiển thị thông báo lỗi "Sản phẩm…không đủ số lượng trong kho!"). Còn nếu thanh toán thành công, sẽ thông báo "Bạn đã đặt hàng thành công!!" sau đó tự động chuyển hướng đến trang hóa đơn sản phẩm, để xem chi tiết hóa đơn và tổng tiền.

Từ trang hóa đơn người dùng có thể tải file hóa đơn dưới định dạng PDF, có tên file là HoaDon\_ThanhToan.pdf. Nội dung file chứa thông tin chi tiết bao gồm: Tên người đặt, phương thức thanh toán, mã đơn hàng, ngày đặt hàng, danh sách sản phẩm, tổng giá trị đơn hàng.

CHƯƠNG 5 ĐÁNH GIÁ VÀ HƯỚNG GIẢI QUYẾT

5.1 Đánh giá kết quả đạt được

Hệ thống website đặt đồ ăn nhanh FastBite đã được triển khai thành công. Dự án không chỉ giúp tổng kết lại những kiến thức và kỹ năng đã học trong suốt quá trình học tập tại trường mà còn mang lại một sản phẩm thực tế, có khả năng áp dụng vào thực tiễn. Cụ thể, các kết quả đạt được bao gồm việc xây dựng một website đặt đồ ăn trực tuyến với giao diện thân thiện, dễ sử dụng cho cả khách hàng và người quản lý, hỗ trợ đầy đủ các chức năng chính như đăng ký/đăng nhập tài khoản, tìm kiếm món ăn,... Đối với Admin, hệ thống cung cấp các công cụ quản lý hệ thống hiệu quả, giúp tối ưu hóa quy trình vận hành của cửa hàng.

Về mặt kỹ thuật, website được phát triển bằng ASP.NET WebForms, tận dụng các server controls và tính năng kéo-thả để giảm thời gian thiết kế giao diện, đồng thời tích hợp với SQL Server để quản lý cơ sở dữ liệu một cách hiệu quả. Các Stored Procedure được thiết kế tối ưu, đảm bảo hiệu suất truy vấn, tính bảo mật và đảm bảo tính toàn vẹn dữ liệu trong các thao tác phức tạp như xử lý thanh toán và lưu đơn hàng. Giao diện khách hàng được tối ưu hóa để hỗ trợ dùng trên nhiều thiết bị.

5.2 Hạn chế và vấn đề cần khắc phục

* Mặc dù hệ thống FastBite đã hoàn thành các mục tiêu đề ra, nhưng trong quá trình phát triển và triển khai, vẫn còn một số hạn chế cần khắc phục:
* Do sử dụng ASP.NET WebForms, nhược điểm của framework này, đó là cơ chế ViewState làm tăng kích thước dữ liệu truyền tải giữa client và server, dẫn đến thời gian tải trang có thể chậm hơn khi xử lý các giao diện phức tạp hoặc với lượng người dùng lớn đồng thời.
* Hệ thống chưa phân quyền rõ các chức năng cho các chức vụ nhân viên và người quản lý hệ thống.
* Hệ thống chưa có hệ thống kho hàng để quản lý số lượng hàng hóa đã nhập để dễ dàng cho việc quản lý các dòng tiền đầu ra và vào của hệ thống.
* Hệ thống gặp lỗi khi tải ảnh đại diện tài khoản người dùng trong giao diện tài khoản (Profile), cụ thể là khi người dùng tải ảnh lên, ảnh đại diện đôi khi không được lưu đúng định dạng đường dẫn vào bảng Users, dẫn đến ảnh đại diện vẫn hiển thị ảnh mặc định ngay cả khi đã chọn ảnh mới.
* Chức năng thanh toán bằng mã QR, hệ thống chưa phát triển tích hợp API ngân hàng để tự động kiểm tra, do ngân hàng Vietcombank không cung cấp API công khai để kiểm tra trạng thái giao dịch miễn phí mà phải tạo tài khoản doanh nghiệp và đăng ký API riêng. Do đó hệ thống buộc phải xác nhận thủ công, điều này tiềm ẩn nguy cơ gian lận hoặc nhầm lẫn từ phía người dùng.

5.3 Định hướng phát triển

Dựa trên các hạn chế, hệ thống FastBite có định hướng được cải thiện như sau:

* Chuyển sang MVC hoặc Blazor để cải thiện hiệu suất, giảm ViewState, tăng khả năng tùy chỉnh giao diện và hỗ trợ SEO.
* Sửa lỗi tải ảnh đại diện tài khoản, đảm bảo ảnh lưu và hiển thị đúng, thêm tính năng xem trước ảnh.
* Tích hợp API doanh nghiệp (nếu có điều kiện tài chính) hoặc tích hợp webhook từ các dịch vụ thanh toán để nhận thông báo giao dịch theo thời gian thực. Ngoài ra, hệ thống sẽ bổ sung cơ chế lưu thông tin người chuyển khoản để hỗ trợ đối chiếu giao dịch, đảm bảo tính minh bạch và chính xác trong quản lý đơn hàng.
* Nâng cao trải nghiệm người dùng với tính năng thông minh: Thêm các tính năng như gợi ý món ăn dựa trên lịch sử đặt hàng hoặc sở thích cá nhân. Hỗ trợ đánh giá món ăn bằng sao và bình luận, cùng với chức năng tìm kiếm nâng cao để giúp khách hàng dễ dàng chọn món phù hợp.
* Xây dựng hệ thống quản lý kho hàng: Tích hợp quản lý kho để theo dõi số lượng hàng nhập, xuất, và tồn kho theo thời gian thực. Việc này hỗ trợ Admin quản lý dòng tiền đầu vào/đầu ra, dự đoán nhu cầu dựa trên lịch sử bán hàng, và tự động cảnh báo khi hàng tồn kho thấp, từ đó tối ưu hóa quy trình vận hành.
* Hỗ trợ đa ngôn ngữ và mở rộng thị trường: Thêm tính năng đa ngôn ngữ (tiếng Anh, tiếng Việt, và các ngôn ngữ khác) để phục vụ khách hàng quốc tế hoặc khách du lịch tại Việt Nam. Điều này giúp FastBite mở rộng thị trường, đặc biệt tại các khu vực du lịch như Khánh Hòa, và tăng khả năng cạnh tranh với các nền tảng đặt đồ ăn quốc tế.

# KẾT LUẬN

Chuyên đề tốt nghiệp với đề tài “Xây dựng website đặt đồ ăn nhanh sử dụng WebForms ASP.NET” đã hoàn thành các mục tiêu đề ra, mang lại một hệ thống trực tuyến hỗ trợ đặt đồ ăn tiện lợi cho khách hàng và quản lý hiệu quả cho nhà hàng. Hệ thống không chỉ đáp ứng các yêu cầu cơ bản như đăng ký, đăng nhập, tìm kiếm món ăn, quản lý giỏ hàng,…, mà còn cung cấp giao diện quản lý chuyên nghiệp cho Admin với các công cụ quản lý hệ thống đã giúp tối ưu hóa quá trình phát triển quản lý cửa hàng, từ việc thiết kế cơ sở dữ liệu, triển khai Stored Procedure, đến xây dựng giao diện người dùng, đảm bảo hệ thống hoạt động ổn định và đáp ứng đầy đủ nhu cầu.

Dự án FastBite không chỉ dừng lại ở một sản phẩm hoàn chỉnh mà còn mang lại giá trị thực tiễn, có thể áp dụng vào các quán ăn hoặc nhà hàng nhỏ, giúp họ chuyển đổi sang mô hình kinh doanh trực tuyến, tối ưu hóa quy trình phục vụ khách hàng, và tăng khả năng cạnh tranh trong thời đại công nghệ số. Đối với bản thân, chuyên đề này là một cột mốc quan trọng, đánh dấu sự kết thúc của hành trình học tập tại trường và mở ra những cơ hội mới trong lĩnh vực phát triển phần mềm. Dù vẫn còn một số hạn chế như đã nêu ở phần trên, những kết quả đạt được và các hướng phát triển trong tương lai sẽ là nền tảng vững chắc để bản thân tiếp tục cải tiến hệ thống, đáp ứng tốt hơn nhu cầu của người dùng và phù hợp với xu hướng công nghệ hiện đại, chẳng hạn như tích hợp các công nghệ mới, tối ưu hóa hiệu suất, và mở rộng tính năng.

Cuối cùng, quá trình thực hiện chuyên đề này cũng là cơ hội quý báu để bản thân có thể lắng nghe những góp ý, đánh giá từ giảng viên hướng dẫn và các bạn, từ đó nhận ra những điểm còn thiếu sót trong quá trình thực hiện, học hỏi thêm kinh nghiệm và cải thiện kỹ năng. Những bài học rút ra từ dự án này sẽ là hành trang quan trọng, giúp bản thân tự tin hơn khi bước vào môi trường làm việc thực tế và chuẩn bị tốt hơn cho các dự án lớn hơn trong tương lai. Với tất cả những gì đã đạt được, chuyên đề không chỉ là một nhiệm vụ học tập mà còn là một trải nghiệm ý nghĩa, đánh dấu bước phát triển mới trong hành trình trở thành một lập trình viên chuyên nghiệp.

TÀI LIỆU THAM KHẢO

1. ASP.NET, WebForms. *https://en.wikipedia.org/wiki/ASP.NET\_Web\_Forms.*

2. HTML, Form. *https://bbbootstrap.com/snippets/payment-form-three-different-payment-options-13285516.*

3. UI, Web opensource. *https://themewagon.com/themes/free-bootstrap-4-html5-restaurant-website-template-feane/.*

4. Tài Liệu mẫu quản lý. *https://docx.com.vn/tai-lieu/bao-cao-chuyen-de-xay-dung-website-ban-do-an-811.*

5. webforms, Ưu nhược. *https://suns.com.vn/phan-mem-tren-webform-hay-winform-cai-nao-moi-thuc-su-tot/.*

6. *Hà Thị Thanh Ngà.;Bài giảng "Phân tích thiết kế hệ thống thông tin". Nha Trang, Khánh Hòa: Đại Học Nha Trang.*

7. *Nguyễn Thị Thu Thúy. Bài giảng "Hệ quản trị cơ sở dữ liệu". Nha Trang, Khánh Hòa: Đại Học Nha Trang.*

8. *Nguyễn Đức Thuần. Bài giảng "Cơ sở dữ liệu". Nha Trang, Khánh Hòa: Đại Học Nha Trang.*

9. FastBite, Sơ đồ hệ thống. *https://miro.com/app/board/uXjVIEm-hOk=/?share\_link\_id=420912866023.*

10. Lịch sử WebForms, tính ứng dụng. *https://fptshop.com.vn/tin-tuc/danh-gia/asp-net-la-gi-165617.*