

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



BÀI TẬP LỚN

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

ĐỀ TÀI: PHÁT TRIỂN WEBSITE GIỚI THIỆU NHÀ HÀNG

Giáo viên hướng dẫn: 1. Phạm Văn Tiệp

2. Trần Vũ Đại

Sinh viên thực hiện: Đinh Mai Phương

Hà Nội, 2025

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



BÀI TẬP LỚN

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

ĐỀ TÀI: PHÁT TRIỂN WEBSITE GIỚI THIỆU NHÀ HÀNG

STT	Mã Sinh Viên	Họ và Tên	Ngày Sinh	Điểm	
				Bằng Số	Bằng Chữ
1	1771020552	Đinh Mai Phương	17/10/2025		

CÁN BỘ CHẤM THI

Hà Nội, 2025

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ thông tin ngày càng phát triển mạnh mẽ, việc áp dụng các giải pháp số hóa vào hoạt động kinh doanh đã trở thành xu hướng tất yếu. Đặc biệt, trong ngành dịch vụ ăn uống, việc sở hữu một website chuyên nghiệp không chỉ giúp nhà hàng nâng cao hình ảnh thương hiệu mà còn tạo điều kiện thuận lợi cho khách hàng tiếp cận thông tin và đặt bàn một cách nhanh chóng, tiện lợi.

Nhận thức được tầm quan trọng đó, em được giao đề tài “**Phát triển website giới thiệu nhà hàng**” để thực hiện kết thúc học phần **Thiết kế và lập trình Back-End**. Đề tài nhằm xây dựng một website có chức năng cơ bản như: giới thiệu nhà hàng, hiển thị thực đơn món ăn, tin tức, đặt bàn trực tuyến và liên hệ, đồng thời bảo đảm xử lý dữ liệu hiệu quả từ phía máy chủ (server side).

Thông qua quá trình thực hiện, em không chỉ được áp dụng kiến thức lý thuyết đã học về mô hình MVC, cơ sở dữ liệu, xử lý form và tương tác với người dùng, mà còn rèn luyện kỹ năng lập trình ASP.NET Core, phân tích yêu cầu hệ thống và tổ chức cấu trúc mã nguồn một cách khoa học.

Em xin chân thành cảm ơn các giảng viên đã tận tình hướng dẫn và tạo điều kiện để em hoàn thành đề tài này. Dù đã cố gắng hoàn thiện tốt nhất trong khả năng, nhưng chắc chắn vẫn còn những thiếu sót. Em rất mong nhận được sự góp ý quý báu từ thầy/cô để hoàn thiện hơn trong các dự án sau.

MỤC LỤC

LỜI NÓI ĐẦU.....	1
MỤC LỤC.....	2
MỤC LỤC HÌNH ẢNH.....	4
CHƯƠNG I. TỔNG QUAN GIỚI THIỆU ĐỀ TÀI.....	4
1.1. Giới thiệu đề tài.....	4
1.2. Phân tích đề tài.....	5
1.2.1. Mục tiêu chính:.....	5
1.2.2. Đối tượng sử dụng.....	5
1.2.3. Phạm vi thực hiện.....	5
1.3. Kiến thức tổng quát của hệ thống.....	5
1.3.1. Mô hình kiến trúc (MVC).....	5
1.3.2. Tổng quan Client.....	7
1.3.3. Database.....	7
Vai trò của Database trong website giới thiệu nhà hàng.....	7
1.3.4. Luồng xử lý yêu cầu.....	8
1.4. Phân tích chương trình.....	8
1.4.1. Kiến trúc chương trình.....	8
1.4.2. Các chức năng chính.....	8
1.4.3. Giao tiếp người dùng.....	8
CHƯƠNG II. PHÂN TÍCH THIẾT KẾ, HỆ THỐNG THÔNG TIN WEBSITE GIỚI THIỆU NHÀ HÀNG.....	9
2.1. Mục tiêu của website, các chức năng cần có.....	9
2.1.1. Đối tượng sử dụng:.....	9
2.1.2. Yêu cầu chức năng.....	9
2.1.3. Yêu cầu phi chức năng.....	9
2.1.4. Phát triển một website có chức năng cơ bản về giới thiệu nhà hàng....	10
2.2. Thiết kế CSDL và sơ đồ quan hệ ERD.....	11
2.2.1. Quy trình nghiệp vụ.....	11
2.2.2. Dữ liệu và trạng thái và ràng buộc.....	12
2.2.3. Các tác nhân (Actors).....	12
2.2.4. Danh sách Use Case - Tác nhân và các luồng sự kiện.....	12
2.2.5. Cấu trúc cơ sở dữ liệu.....	13
2.2.6. Các khóa.....	16
2.2.7. Mối quan hệ.....	16
2.2.8. Biểu đồ Use Case tổng thể.....	19
2.2.9. Sơ đồ quan hệ (ERD).....	20
2.2.10. Một số sơ đồ hoạt động.....	22
2.3. Tiến hành tạo bảng trong SQL Server.....	24

CHƯƠNG 3: THIẾT KẾ VÀ PHÂN TÍCH GIAO DIỆN.....	25
3.2. Thiết kế giao diện.....	25
3.2.1. Sơ đồ luồng giao diện.....	26
3.2.2. Trang chủ.....	26
3.2.3. Trang thực đơn.....	27
3.2.4. Trang đặt bàn.....	27
3.2.5. Trang tin tức.....	28
3.2.6. Trang liên hệ.....	29
3.2.7. Trang quản trị.....	30
3.2.8. Trang quản trị Quản lý đặt bàn.....	31
3.2.9. Trang quản trị Quản lý món ăn.....	32
3.2.10. Trang đăng ký.....	33
3.2.11. Trang đăng nhập.....	34
3.3. Một số giao diện khác.....	36
CHƯƠNG 4: TRIỂN KHAI BACK - END.....	37
4.1. Kiến trúc tổng quan.....	37
4.2. Model.....	38
4.2.1. Model LoaiMon.....	38
4.2.2. Model MonAn.....	39
4.2.3. Model DatBan.....	39
4.2.4. Model TinTuc.....	40
4.2.5. LienHe.....	40
4.2.6. KhachHang.....	41
4.2.7. AdminUser.....	41
4.2.8. AdminLogs.....	42
4.2.9. AdminRoles.....	42
4.2.10. Roles.....	42
4.2.11. AdminLogin.....	43
4.2.12. KhachHangLoginModel.....	43
4.2.13. KhachHangRegisterModel.....	43
4.3. Controller.....	43
4.3.1. AdminController.....	43
4.3.2. DatBanController.....	50
4.3.3. HomeController.....	51
4.3.4. KhachHangController.....	52
4.3.5. LienHeController.....	54
4.3.6. MenuController.....	56
4.3.7. TinTucController.....	57
4.4. View.....	57
4.4.1. Admin.....	57

4.4.2. DatBan.....	58
4.4.3. KhachHang.....	58
4.4.4. LienHe.....	59
4.4.5. Home.....	59
4.4.6. Menu.....	59
4.4.7. TinTuc.....	60
4.4.8. Share/_Layout.cshtml.....	61
4.5. Các tệp cấu hình khác.....	61
4.5.1. appsetting.json.....	61
4.5.2. Program.cs.....	62
4.5.3. Phân tích mã nguồn AppDbContext.....	64
4.5. Vị trí của các thư mục và link Github.....	66
4.5.1. Link GitHub.....	66
4.5.2. Vị trí của các thư mục.....	67
4.7. Đề xuất hướng phát triển.....	67
KẾT LUẬN.....	68
TÀI LIỆU THAM KHẢO.....	69

MỤC LỤC HÌNH ẢNH

Hình 1. Khái quát cách thức hoạt động của MVC.....	10
Hình 2. Biểu đồ use case tổng thể.....	23
Hình 3. Sơ đồ ERD.....	24
Hình 4. Sơ đồ hoạt động đặt bàn.....	25
Hình 4. Sơ đồ hoạt động quản lý món ăn.....	26
Hình 5. Tạo các cơ sở dữ liệu trên SQL Server.....	28
Hình 6. Thêm các dữ liệu vào các bảng.....	28
Hình 7. Thanh điều hướng (Header của website).....	29
Hình 8. Sơ đồ luồng giao diện.....	29
Hình 9. Trang chủ và trang Thực đơn.....	30
Hình 10. Trang Đặt bàn.....	31
Hình 11. trang Tin tức.....	32
Hình 12. Trang Liên hệ.....	33
Hình 13. trang Quản trị trước khi đăng nhập.....	34
Hình 14 trang Quản trị sau khi đăng nhập.....	34
Hình 15. trang Quản lý phần quản lý đặt bàn.....	35
Hình 16. trang Quản trị phần quản lý món ăn.....	36
Hình 17. trang Đăng ký tài khoản.....	37
Hình 18. trang Đăng nhập.....	37
Hình 19. sau khi đăng nhập hoặc đăng ký.....	38
Hình 20. Hiển thị danh sách “đơn đặt bàn của tôi” sau khi khách hàng đăng nhập và đặt bàn.....	39
Hình 21. Hiển thị thông báo khi khách hàng đăng xuất thành công.....	40
Hình 22. Vị trí các thành phần back-end.....	70

CHƯƠNG I. TỔNG QUAN GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu đề tài

Trong thời đại phát triển như hiện nay, việc xây dựng một website chuyên nghiệp là một yếu tố quan trọng giúp các doanh nghiệp trong lĩnh vực kinh doanh, đặc biệt là nhà hàng, để tiếp cận khách hàng một cách nhanh chóng và hiệu quả. Website không chỉ là nơi giới thiệu, quảng bá thông tin hình ảnh nhà mà còn hỗ trợ các chức năng tương tác như đặt bàn trực tuyến, tiếp nhận phản hồi ý kiến của khách hàng, cập nhật các khuyến mãi và sự kiện, ...

Xuất phát từ các nhu cầu thực tế đó, nhận được đề tài được giao, em thực hiện đề tài “phát triển website giới thiệu nhà hàng” nhằm xây dựng một hệ thống web động có thể giới thiệu tổng quan về nhà hàng, hiển thị thực đơn có các món ăn, cập nhật tin tức, khuyến mãi, khách hàng được chủ động đặt bàn trực tuyến và gửi liên hệ đến nhà hàng. Đây là cơ hội để em vận dụng những kiến thức đã học trong môn thiết kế, lập trình Back-End, cụ thể là mô hình MVC (Model - View - Controller), tương tác với cơ sở dữ liệu, xử lý form và phát triển website bằng ASP.NET core MVC.

1.2. Phân tích đề tài

1.2.1. Mục tiêu chính:

Xây dựng website giới thiệu nhà hàng có giao diện thân thiện, đơn giản, dễ sử dụng. Cung cấp các chức năng: xem giới thiệu nhà hàng, thực đơn món ăn, tin tức khuyến mãi, đặt bàn trực tuyến, gửi phản hồi.

Thiết kế cơ sở dữ liệu để lưu trữ thông tin khách hàng, món ăn, đơn đặt bàn và tin tức. Phát triển hệ thống back-end để xử lý dữ liệu, xác nhận đơn đặt bàn và phản hồi người dùng.

Đây là một website động, tích hợp giao diện người dùng (khách hàng) và giao diện quản trị (admin).

Hiển thị thực đơn món ăn, phân loại theo danh mục: món chính, món tráng miệng, đồ uống.

Đặt bàn trực tuyến: khách hàng truy cập website, nhập thông tin và thời gian muốn đặt bàn.

1.2.2. Đối tượng sử dụng

- Khách hàng: truy cập website để xem thông tin nhà hàng, thực đơn, đặt bàn được gửi liên hệ.
- Quản trị viên: quản lý món ăn, tin tức và đơn đặt bàn.

1.2.3. Phạm vi thực hiện

- Xây dựng website với các chức năng cơ bản phục vụ nhu cầu giới thiệu và đặt bàn.

- Giao diện đơn giản, dễ nhìn, chỉ có thể mở rộng thêm tính năng đăng nhập quản trị và quản lý nội dung.

1.3. Kiến trúc tổng quát của hệ thống

1.3.1. Mô hình kiến trúc (MVC)

MVC là viết tắt của Model-View-Controller. Cấu trúc Model-View-Controller (MVC) là một mẫu kiến trúc/mẫu thiết kế (design pattern) tách ứng dụng thành ba thành phần logic chính: Model, View và Controller. Mỗi thành phần kiến trúc được xây dựng để xử lý các khía cạnh phát triển cụ thể của một ứng dụng.

Đặc điểm của MVC

- Cung cấp sự phân tách rõ ràng giữa logic nghiệp vụ, logic UI và logic đầu vào.
- Cung cấp toàn quyền kiểm soát HTML và UML, giúp dễ dàng thiết kế kiến trúc ứng dụng web.
- Có thể sử dụng để xây dựng các ứng dụng có URL dễ hiểu và có thể tìm kiếm được
- Hỗ trợ lập trình dựa trên thử nghiệm (Test-driven Development).

MVC là một mẫu thiết kế tiêu chuẩn được nhiều lập trình viên quen thuộc nhờ vào khả năng mở rộng và có thể mở rộng. MVC thường được sử dụng để làm framework phát triển web tiêu chuẩn cũng như các ứng dụng di động.

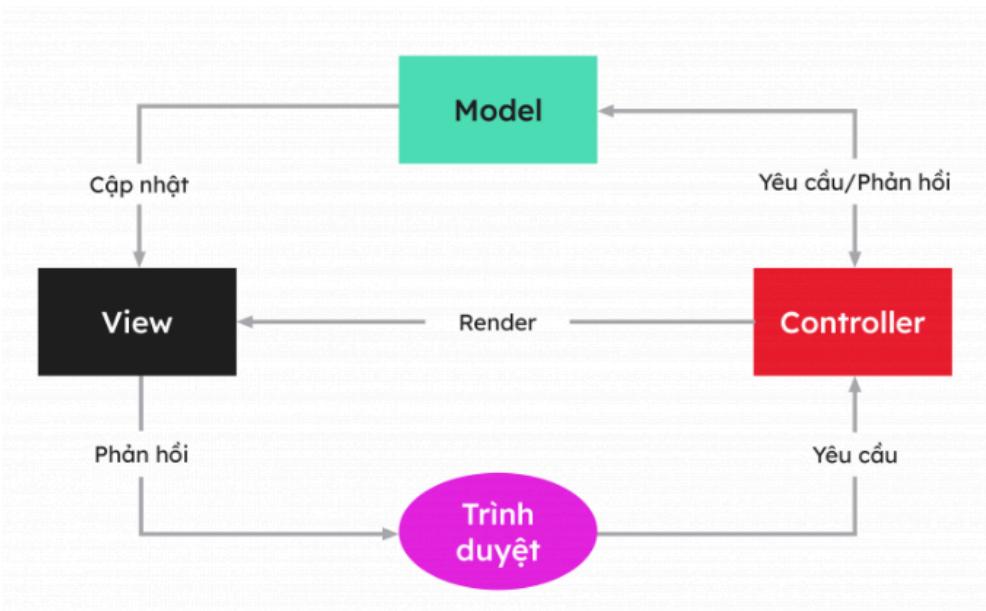
- Các thành phần trong mô hình MVC

- Model: Model là các thành phần của ứng dụng tương ứng với tất cả logic liên quan đến miền dữ liệu (data domain), hoặc nói ngắn gọn đây là phần back-end chứa tất cả logic dữ liệu của ứng dụng. Dữ liệu ở đây có thể là dữ liệu đang được truyền giữa các thành phần View và Controller hoặc bất kỳ dữ liệu nào khác liên quan đến logic của doanh nghiệp. Nếu trạng thái của dữ liệu này thay đổi thì Model thường sẽ thông báo cho View (để màn hình có thể thay đổi khi cần) và đôi khi là Controller (nếu cần logic khác để cập nhật View). Các đối tượng Model có thể xuất hiện từ cơ sở dữ liệu, thao tác và lưu trữ trạng thái Model trong cơ sở dữ liệu.

- Controller: Controller là các thành phần xử lý tương tác của người dùng để làm việc với Model (cập nhật logic dữ liệu) hoặc/ và với View (cập nhật hiển thị giao diện người dùng). Trong ứng dụng MVC, Controller xử lý các giá trị chuỗi truy vấn và chuyển các giá trị này cho Model, từ đó Model sẽ truy vấn cơ sở dữ liệu bằng cách sử dụng các giá trị đó. View hiển thị thông tin do Controller xử lý và phản hồi đầu vào từ tương tác của người dùng.

- View: View là các thành phần hiển thị giao diện người dùng (UI) của ứng dụng. Thông thường, giao diện người dùng này được tạo từ dữ liệu Model.

- Cách thức hoạt động



Hình 1. Khái quát cách thức hoạt động của MVC

Trình duyệt gửi yêu cầu (Request) đến Controller, người dùng tương tác với giao diện website, trình duyệt gửi yêu cầu (HTTP Request) tới Controller tương ứng. Controller xử lý yêu cầu, Controller nhận request, kiểm tra yêu cầu và gọi đến Model nếu cần thao tác dữ liệu. Controller tương tác với Model, Model thực hiện nghiệp vụ như truy vấn, thêm, sửa, xóa dữ liệu trong cơ sở dữ liệu, sau khi xử lý xong, Model gửi phản hồi lại cho Controller. Controller chọn View phù hợp, dựa vào kết quả xử lý, Controller gọi View tương ứng để tạo giao diện phản hồi. View được render và gửi về trình duyệt, View hiển thị dữ liệu do Model cung cấp dưới dạng HTML. Kết quả cuối cùng được render và trả về trình duyệt để người dùng xem.

1.3.2. Tổng quan Client

Trong kiến trúc ứng dụng web, Client (máy khách) là thành phần đầu cuối mà người dùng trực tiếp tương tác thông qua trình duyệt web (như Chrome, Firefox, Edge...). Client gửi các yêu cầu (request) đến máy chủ (server) và nhận lại phản hồi (response) để hiển thị nội dung tương ứng. Client không thực hiện xử lý nghiệp vụ chính hay lưu trữ dữ liệu lâu dài mà chủ yếu đảm nhiệm vai trò hiển thị giao diện và thu thập dữ liệu từ người dùng.

Trong đề tài Phát triển website giới thiệu nhà hàng, phần client đảm nhận các vai trò chính như sau:

- Giao diện người dùng (UI): Hiển thị các trang như Trang chủ, Giới thiệu, Thực đơn món ăn, Tin tức, Đặt bàn, Liên hệ,... với thiết kế thân thiện và dễ sử dụng.
- Tương tác với người dùng: Cho phép người dùng thực hiện các hành động như điền form đặt bàn, gửi liên hệ, xem thực đơn, đọc tin tức,...

- Gửi yêu cầu đến server: Mỗi hành động từ người dùng (ví dụ nhấn nút "Gửi", "Đặt bàn") sẽ gửi một request đến server để xử lý.
- Nhận phản hồi từ server: Sau khi server xử lý xong, client sẽ hiển thị kết quả cho người dùng như: "Đặt bàn thành công", "Thông tin không hợp lệ",...

1.3.3. Database

Cơ sở dữ liệu (Database) là tập hợp có tổ chức các dữ liệu có liên quan, được lưu trữ và quản lý để phục vụ cho việc truy xuất, cập nhật, tìm kiếm và bảo trì một cách hiệu quả. Trong ứng dụng web, cơ sở dữ liệu giữ vai trò trung tâm trong việc lưu trữ thông tin động như thông tin người dùng, sản phẩm, đơn đặt hàng, phản hồi, v.v.

Một hệ quản trị cơ sở dữ liệu (Database Management System – DBMS) như SQL Server sẽ giúp tạo, truy xuất và quản lý dữ liệu một cách an toàn và hiệu quả.

Vai trò của Database trong website giới thiệu nhà hàng

Trong đề tài “Phát triển website giới thiệu nhà hàng”, database đóng vai trò lưu trữ toàn bộ dữ liệu mà website cần sử dụng, bao gồm:

- Danh sách món ăn và thông tin chi tiết của từng món.
- Thông tin đơn đặt bàn từ khách hàng.
- Các tin tức, bài viết khuyến mãi của nhà hàng.
- Thông tin phản hồi, liên hệ từ người dùng gửi đến nhà hàng.

Nhờ vào hệ thống cơ sở dữ liệu, website có thể truy xuất nhanh chóng và hiển thị nội dung phù hợp với nhu cầu người dùng, đồng thời giúp quản trị viên theo dõi hoạt động của hệ thống.

Hệ quản trị cơ sở dữ liệu sử dụng Đề tài sử dụng Microsoft SQL Server làm hệ quản trị cơ sở dữ liệu, kết hợp với Entity Framework Core trong ASP.NET Core để thao tác dữ liệu theo mô hình ORM (Object-Relational Mapping). Điều này giúp việc truy xuất dữ liệu trở nên dễ dàng, nhanh chóng và an toàn hơn.

1.3.4. Luồng xử lý yêu cầu

Luồng xử lý yêu cầu (Request Processing Flow) mô tả cách một yêu cầu từ phía người dùng (Client) được tiếp nhận và xử lý bởi hệ thống web. Quá trình này diễn ra qua nhiều lớp trong mô hình **MVC (Model - View - Controller)** và bao gồm các bước: nhận yêu cầu, xử lý logic, truy xuất dữ liệu, tạo giao diện phản hồi và gửi kết quả về cho trình duyệt.

Luồng xử lý yêu cầu trong mô hình MVC giúp tách biệt rõ ràng giữa giao diện, xử lý logic và quản lý dữ liệu. Điều này giúp hệ thống website nhà hàng dễ bảo trì, mở rộng và nâng cao hiệu suất hoạt động. Việc hiểu rõ quy trình này là rất quan trọng trong thiết kế và lập trình back-end chuyên nghiệp.

1.4. Phân tích chương trình

1.4.1. Kiến trúc chương trình

Website được xây dựng theo mô hình MVC (Model - View - Controller) của ASP.NET Core:

- Model: định nghĩa các lớp dữ liệu như: Món Ăn, Đặt Bàn, Liên Hệ, Tin Tức.
- View: các trang hiển thị thông tin đến người dùng như trang chủ, Menu, Đặt bàn, Giới thiệu, Tin tức, Liên hệ.
- Controller: Xử lý logic và điều hướng dữ liệu giữa View và Model. Ví dụ: HomeController, DatBanController, MonAnController, ...

1.4.2. Các chức năng chính

- Trang chủ (Home): hiển thị tổng quan nhà hàng và các món nổi bật.
- Giới thiệu: cung cấp thông tin về nhà hàng (lịch sử, không gian, đội ngũ, phong cách phục vụ).
- Thực đơn món ăn (Menu): liệt kê các món ăn chia theo danh mục.
- Tin tức (News): hiển thị các sự kiện, khuyến mãi, thông báo mới nhất.
- Đặt bàn (DatBan): cho phép khách điền form đặt bàn, lưu dữ liệu và cơ sở dữ liệu và hiển thị thông báo xác nhận.
- Liên hệ (Contact): người dùng gửi phản hồi hoặc thắc mắc cho nhà hàng.

1.4.3. Giao tiếp người dùng

Các form như đặt bàn và liên hệ được xác thực dữ liệu phía máy chủ (server-side validation). Sau khi đặt bàn thành công, hiển thị thông báo và danh sách đặt bàn ở dưới form. Giao diện sử dụng Razor View, kết hợp với Bootstrap để dễ sử dụng và hiển thị đẹp trên các thiết bị.

CHƯƠNG II. PHÂN TÍCH THIẾT KẾ, HỆ THỐNG THÔNG TIN WEBSITE GIỚI THIỆU NHÀ HÀNG.

2.1. Mục tiêu của website, các chức năng cần có

Website giới thiệu nhà hàng, giới thiệu không gian nhà hàng, thực đơn, website nhận đặt bàn, cập nhật các tin tức, chương trình khuyến mãi, có thông tin liên hệ với nhà hàng và đường chỉ dẫn đến nhà hàng. Ngoài ra trang web còn phải có giao diện đẹp, hình ảnh rõ nét khi hiển thị trên điện thoại, ipad, máy tính.

2.1.1. Đối tượng sử dụng:

Vai trò	Mô tả
Khách hàng	Người truy cập website để xem thông tin, đặt bàn, gửi liên hệ.
Quản trị viên (Admin)	Người quản lý hệ thống: thực đơn, tin tức, đặt bàn, tài khoản, vai trò.
Super Admin	Người có toàn quyền: tạo admin mới, gán vai trò, quản lý nhật ký.

2.1.2. Yêu cầu chức năng

- Chức năng dành cho khách hàng
 - Xem thông tin về món ăn, phân loại theo món.
 - Đặt bàn trực tuyến, cung cấp thông tin cá nhân và thời gian đặt bàn.
 - Gửi liên hệ hoặc thắc mắc thông qua form liên hệ.
- Chức năng dành cho quản trị viên
 - Quản lý loại ăn: thêm, sửa, xóa loại món.
 - Quản lý món ăn: thêm, sửa, xóa món ăn, gán loại món.
 - Quản lý tin tức: thêm, sửa, xóa tin tức.
 - Quản lý đặt bàn: xác nhận, hủy, cập nhật trạng thái đặt bàn.
 - Quản lý khách hàng: xem danh sách khách hàng, xóa, sửa thông tin.
 - Quản lý tài khoản admin: tạo, sửa, xóa, admin.
 - Quản lý vai trò admin: tạo, sửa, xóa các vai trò.
 - Gán vai trò cho admin.
 - Xem nhật ký thao tác của admin.

2.1.3. Yêu cầu phi chức năng

- Website hoạt động 24/7, giao diện thân thiện trên desktop và mobile.
- Dữ liệu khách hàng và admin được bảo mật.
- Dữ liệu được kiểm tra hợp lệ trước khi lưu vào cơ sở dữ liệu.
- Hiệu năng: phản hồi nhanh, hỗ trợ nhiều người dùng đồng thời.

2.1.4. Phát triển một website có chức năng cơ bản về giới thiệu nhà hàng

- Trang chủ: giới thiệu tổng quan, hiển thị thông tin tổng quan về nhà hàng, bao gồm giới thiệu, hình ảnh, và các liên kết chính. Ngoài thanh điều hướng trên header trang chủ còn cáo các nút hành động dẫn sang các trang khác của nhà hàng.
- Thực đơn: Hiển thị danh sách các món ăn với tên, giá, mô tả, và hình ảnh (ví dụ: Phở bò, Bánh flan, Trà đào, v.v.).
- Tin tức (khuyến mãi): cập nhật, hiển thị sự kiện, khuyến mãi mà nhà hàng đang khởi chạy. Cung cấp các thông tin cập nhật hoặc bài viết liên quan đến nhà hàng.
- Liên hệ: Hiển thị thông tin địa chỉ (bao gồm bản đồ), cho phép người dùng gửi thông nhán (họ tên, email, tin nhắn) để liên hệ, gửi đến hệ thống nhà hàng.
- Đặt bàn: đặt bàn trực tuyến. Cho phép người dùng nhập thông tin (họ tên, số điện thoại, email, ngày, giờ, số người, ghi chú) để đặt bàn, sau khi điền các thông tin và tiến hành nhấn nút đặt bàn trình duyệt hiển thị thông báo “Đặt bàn thành công!”. Nếu khách hàng đặt bàn chưa đăng nhập hoặc đăng ký, trình duyệt hiện thông báo “Bạn chưa đăng nhập” và điều hướng sang trang đăng nhập
- Trang quản trị: quản lý món ăn, đơn đặt bàn, quản lý tin tức.
- Trang đăng nhập: dành cho khách hàng đã có tài khoản, giúp khách hàng có thể đặt bàn.
- Trang đăng ký: nếu chưa có tài khoản, lần đầu tiên truy cập nhà hàng, cần đăng ký tài khoản, có tài khoản để đặt bàn.
- Chức năng cho quản trị viên
 - Quản lý món ăn: Thêm món ăn mới (tên, giá, mô tả, ảnh). Sửa hoặc xóa món ăn hiện có. Hiển thị danh sách món ăn với trạng thái (Sửa/Xóa).
 - Quản lý đặt bàn: Xem danh sách đặt bàn (họ tên, số điện thoại, email, ngày, giờ, số người, ghi chú). Xác nhận hoặc xóa các đặt bàn. Hiển thị trạng thái đặt bàn (Xác nhận/Xóa).
 - Quản lý tài khoản (tài khoản của Admin): Đăng nhập với quyền quản trị (ví dụ: tài khoản "admin"). Quản lý thông tin người dùng (có thể bao gồm danh sách tài khoản nếu cần).

- Chức năng bổ sung
 - Hiển thị hình ảnh: Hỗ trợ tải lên và hiển thị ảnh cho món ăn, địa chỉ, hoặc nội dung khác.
 - Phản hồi người dùng: Xử lý và lưu trữ thông tin liên hệ từ khách hàng.
 - Tùy chỉnh giao diện: Sử dụng màu sắc nhất quán (cam, xanh, đỏ) và bố cục đơn giản, thân thiện.
 - Thông tin bản quyền: Hiển thị thông tin bản quyền và chịu trách nhiệm xuất bản (ví dụ: © 2025 - Nhà hàng Hương Việt).

2.2. Thiết kế CSDL và sơ đồ quan hệ ERD

Đối với website giới thiệu nhà hàng ta cần phân tích các đối tượng quan trọng, luôn sẵn có trong nhà hàng, ví dụ như: món ăn, tin tức, liên hệ, đặt bàn, ...

Trong bài báo cáo về website giới thiệu nhà hàng các thành phần kiến trúc cơ sở dữ liệu được thiết kế trên SQL Server gồm có các đối tượng sau:

Các bảng và mục đích

- Loại món: LoaiMon: Danh sách loại món ăn.
- Món ăn: MonAn: Danh sách món ăn, thuộc các loại.
- Đặt bàn: DatBan: Đặt bàn của khách hàng.
- Tin tức: TinTuc: Tin tức được đăng trên website.
- Liên hệ: LienHe: Liên hệ của khách hàng.
- AdminUsers: Tài khoản quản trị.
- AdminLogs: Nhật ký thao tác của admin.
- Roles: Bảng phụ gán vai trò cho admin.
- Khách hàng: KhachHang: Tài khoản của khách hàng.

2.2.1. Quy trình nghiệp vụ

Tác nhân	Chức năng	Kết quả
Khách hàng	Xem món	Danh sách món ăn hiển thị
Khách hàng	Đặt bàn	Gửi thông tin đặt bàn, chờ xác nhận
Khách hàng	Gửi liên hệ	Tin nhắn lưu lại, chờ xử lý
Admin	Quản lý món	Món ăn được cập nhật trong hệ thống
Admin	quản lý tin tức	Tin tức hiển thị hoặc ẩn theo yêu cầu
Admin	Quản lý đặt bàn	Cập nhật trạng thái đặt bàn
Admin	Gán vai trò	Admin có thể thêm hoặc bớt quyền

Admin	Xem nhật ký	Xem được thao tác của các admin
-------	-------------	---------------------------------

2.2.2. *Dữ liệu và trạng thái và ràng buộc*

Bảng	trạng thái
DatBan	chờ xác nhận/ Đã xác nhận/ Đã hủy
MonAn	Hiển thị/ Ẩn (tùy logic sau này)
TinTuc	Hiển thị/ Ẩn
AdminUsers	Active/ Inactive (nếu bổ sung)

Các ràng buộc

- Email của khách hàng và admin phải duy nhất.
- Admin không thể tự gán cho mình quyền cao hơn super admin.
- Nhật ký không được sửa hoặc xóa.
- Một món ăn chỉ thuộc một loại món.
- Một khách hàng có thể đặt nhiều bàn.

2.2.3. *Các tác nhân (Actors)*

Tác nhân (Actors)	Mô tả
Khách hàng	Người sử dụng website, xem thông tin, đặt bàn, gửi liên hệ.
Quản trị viên (Admin)	Người quản lý nội dung website, dữ liệu khách hàng, đặt bàn, món ăn, tin tức.
Super Admin	Người quản lý toàn bộ hệ thống, có toàn quyền quản lý admin, phân quyền, theo dõi nhật ký.

2.2.4. *Danh sách Use Case - Tác nhân và các luồng sự kiện*

Use Case	Tác nhân	Luồng chính	Luồng phụ
Xem thực đơn	Khách hàng		
Đặt bàn	Khách hàng	Nhập đúng dữ liệu, lưu thành công	Nhập sai dữ liệu, báo lỗi
Gửi liên hệ	Khách hàng	nhiệp và gửi dữ liệu	Không nhập đủ dữ liệu, báo lỗi
Xem tin tức	Khách hàng		

Đăng ký tài khoản khách hàng	Khách hàng		
Đăng nhập admin	Admin, Super Admin	Đăng nhập đúng, vào hệ thống	Đăng nhập sai, sai mật khẩu, báo lỗi.
Quản lý loại món ăn	Admin		
Quản lý món ăn	Admin	Thêm, sửa, xóa	Không đủ quyền hoặc dữ liệu sai.
Quản lý tin tức	Admin	Thêm, sửa, xóa	Không đủ quyền hoặc dữ liệu sai.
Quản lý đặt bàn	Admin	Xác nhận, xóa	Không đủ quyền hoặc dữ liệu sai.
Quản lý khách hàng	Admin		
Quản lý tài khoản admin	Super Admin		
Quản lý vai trò cho admin	Super Admin		
Quản lý vai trò	Super Admin		
Xem nhật ký thao tác	Super Admin		

2.2.5. Cấu trúc cơ sở dữ liệu

- **LoaiMon (Loại món ăn):**

- Khóa chính: IdLoai (INT, IDENTITY), mã định danh duy nhất cho mỗi loại món ăn.
- Thuộc tính: Name (NVARCHAR(100), NOT NULL): Tên loại món ăn, ví dụ “Món chính”, “Tráng miệng”, yêu cầu không được để trống.
- Mô tả: Lưu các loại món ăn để phân loại thực đơn, hỗ trợ tổ chức dữ (Món chính, Tráng miệng, Đồ uống).
- Vai trò: Bảng này đóng vai trò là thực thể cha, định nghĩa, phân loại các danh mục lớn của thực đơn.

- **MonAn (Món ăn):**

- Thuộc tính:
 - + Khóa chính: IdMonAn (INT, IDENTITY), mã định danh duy nhất cho từng món ăn.
 - + TenMon (NVARCHAR(150), NOT NULL): Tên của món ăn.

- + MoTa (TEXT): Mô tả chi tiết về món ăn.
 - + Gia (DECIMAL(10,2), NOT NULL): giá tiền của món ăn.
 - + HìnhAnh (NVARCHAR(255)): Đường dẫn đến hình ảnh của món ăn.
 - + Khóa phụ: IdLoai tham chiếu đến LoaiMon(IdLoai).
 - Mô tả: Lưu thông tin món ăn, liên kết với loại món ăn.
 - Vai trò: Hiển thị chi tiết thực đơn trên website, hỗ trợ khách hàng chọn món.
- **DatBan (Đặt bàn):**
- Khóa chính: IdDatBan (INT, IDENTITY): mã định danh duy nhất cho mỗi lần đặt bàn.
 - Thuộc tính khác:
 - + HoTen (NVARCHAR(100), NOT NULL), tên của khách hàng đặt bàn.
 - + SDT (VARCHAR(20), NOT NULL), số điện thoại của khách hàng đặt bàn.
 - + Email (VARCHAR(100)), địa chỉ email của người đặt.
 - + ThoiGian (DATETIME, NOT NULL), thời gian đặt bàn.
 - + SoNguoi (INT, NOT NULL), số người tham gia.
 - + TrangThai (NVARCHAR(50), DEFAULT 'Chờ xác nhận'), DaXacNhan (BIT, DEFAULT 0), trạng thái đặt bàn.
 - Khóa phụ: IdKH (INT) tham chiếu đến KhachHang(IdKH).
 - Mô tả: Quản lý thông tin đặt bàn từ khách hàng, hỗ trợ xác nhận hoặc hủy bởi admin.
 - Vai trò: Giúp nhà hàng sắp xếp lịch đặt và phục vụ khách hàng.
- **TinTuc (Tin tức):**
- Khóa chính: IdTinTuc (INT, IDENTITY): Mã định danh duy nhất cho mỗi tin tức.
 - Thuộc tính khác:
 - + TieuDe (NVARCHAR(200), NOT NULL), Tiêu đề của tin tức.
 - + NoiDung (TEXT), Nội dung chi tiết của tin tức.
 - + HìnhAnh (NVARCHAR(255)), Đường dẫn đến hình ảnh của tin tức.
 - + NgayTao (DATETIME, DEFAULT GETDATE()), Ngày tạo tin tức.
 - Mô tả: Lưu thông tin tin tức và khuyến mãi, cập nhật trên website.
 - Vai trò: Cung cấp thông tin mới nhất cho khách hàng, tăng tương tác.
- **LienHe (Liên hệ):**
- Khóa chính: IdLienHe (INT, IDENTITY): Mã định danh duy nhất cho mỗi phản hồi.
 - Thuộc tính khác:
 - + HoTen (VARCHAR(100), NOT NULL), tên của người gửi phản hồi.
 - + Email (VARCHAR(100)), địa chỉ email của người gửi.
 - + TinNhan (TEXT), nội dung tin nhắn hoặc phản hồi của khách hàng gửi đến nhà hàng.
 - + NgayTao (DATETIME, DEFAULT GETDATE()), ngày gửi phản hồi.
 - Mô tả: Lưu thông tin liên hệ phản hồi từ khách hàng để xử lý yêu cầu.

- Vai trò: Hỗ trợ giao tiếp giữa khách hàng và nhà hàng.
- **AdminUsers (Tài khoản quản trị):**
 - Khóa chính: IdAD (INT, IDENTITY): Mã định danh duy nhất cho mỗi tài khoản admin.
 - Thuộc tính khác:
 - + Username (VARCHAR(50), NOT NULL, UNIQUE), tên đăng nhập của admin.
 - + Password (VARCHAR(255), NOT NULL), mật khẩu của admin.
 - + Role (VARCHAR(50), DEFAULT 'staff'), vai trò của admin.
 - Mô tả: Lưu thông tin đăng nhập và vai trò của tài khoản quản trị (quản trị viên).
 - Vai trò: Quản lý hệ thống, thực hiện các tác vụ admin trên website.
- **AdminLogs (Nhật ký thao tác admin):**
 - Khóa chính: LogID (INT, IDENTITY): Mã định danh duy nhất cho mỗi nhật ký.
 - Các thuộc tính khác:
 - + HanhDong (TEXT), mô tả hành động của admin.
 - + ThoiGian (DATETIME, DEFAULT GETDATE()), thời gian thực hiện hành động.
 - Khóa phụ: IdAD tham chiếu đến AdminUsers(IdAD).
 - Mô tả: Ghi lại các thao tác của admin để theo dõi và kiểm tra.
 - Vai trò: Đảm bảo tính minh bạch và giám sát hoạt động quản trị.
- **Roles (Vai trò):**
 - Thuộc tính:
 - + Khóa chính: RoleID (INT, IDENTITY): Mã định danh duy nhất cho mỗi vai trò.
 - + RoleName (VARCHAR(50), NOT NULL): Tên vai trò, ví dụ "Quản lý món ăn".
 - Mô tả: Định nghĩa các vai trò quản trị để phân quyền.
 - Vai trò: Hỗ trợ phân bổ quyền hạn cho các admin.
- **adminRoles (Gán vai trò cho admin):**
 - Khóa chính: IdAD (INT) và RoleID (INT) (kết hợp).
 - Khóa phụ: IdAD tham chiếu đến AdminUsers(IdAD), RoleID tham chiếu đến Roles(RoleID).
 - Mô tả: Liên kết nhiều-nhiều giữa tài khoản admin và vai trò, cho phép một admin có nhiều vai trò.
 - Vai trò: Quản lý phân quyền linh hoạt cho các admin.
- **KhachHang:**
 - Tên bảng: KhachHang
 - Chức năng: lưu trữ thông tin tài khoản khách hàng của hệ thống
 - Thuộc tính:

- + IdKH (int, PK): là khóa chính, tự tăng (IDENTITY(1,1), định danh duy nhất cho mỗi khách hàng. Tự động tăng, bắt đầu từ 1. Giúp truy, liên kết với các bảng khác dễ dàng.
- + Hoten (NVARCHAR(100)): họ và tên của khách hàng, cho phép nhập tối đa 100 ký tự Unicode.
- + Email ((NVARCHAR(100)): Địa chỉ email, dùng làm tài khoản đăng nhập, có ràng buộc UNIQUE để không trùng lặp giữa các khách hàng. Email duy nhất, đảm bảo mỗi khách hàng chỉ có một email.
- + MatKhau (NVARCHAR(100)): mật khẩu của khách hàng dạng plain-text.
- Là bảng gốc để quản lý tài khoản khách hàng, lưu thông tin cơ bản để đăng nhập, đặt bàn, liên hệ, ... Kết nối được với đặt bàn bằng IdKH để ghi nhận đơn đặt bàn của khách hàng. Có thể kết nối với bảng LienHe hoặc các bảng giao dịch khác nếu mở rộng.

2.2.6. Các khóa

Bảng	Mô tả	Khóa chính	Khóa ngoại
LoaiMon	Loại món ăn	IdLoai	
MonAn	Thông tin món ăn	IdMonAn	IdLoai -> LoaiMon(IdLoai)
DatBan	Đặt bàn khách hàng	IdDatBan	IdKH -> KhachHang(IdKH)
TinTuc	Tin tức nhà hàng	IdTinTuc	
LienHe	Khách hàng liên hệ	IdLienHe	
AdminUsers	Tài khoản quản trị	IdAD	
AdminLogs	Nhật ký thao tác admin	LogID	IdAD -> AdminUsers(IdAD)
Roles	Vai trò quản trị	RoleID	
AdminRoles	Gán vai trò cho admin	IdAD, RoleID	IdAD ->AdminUsers(Id AD), RoleID -> Roles(RoleID)

KhachHang	Tài khoản khách hàng	IdKH	
-----------	----------------------	------	--

2.2.7. Mối quan hệ

- **LoaiMon - MonAn:**

- Mối quan hệ 1 -n: Một bản ghi trong LoaiMon (loại món ăn) có thể liên kết với nhiều bản ghi trong MonAn (món ăn), nhưng mỗi bản ghi trong MonAn chỉ thuộc về một bản ghi duy nhất trong LoaiMon.
- Cơ chế thực hiện:
 - + Khóa chính: IdLoai trong LoaiMon.
 - + Khóa phụ: IdLoai trong MonAn, được khai báo với câu lệnh FOREIGN KEY (IdLoai) REFERENCES LoaiMon(IdLoai).
- Điều này đảm bảo rằng giá trị IdLoai trong MonAn phải khớp với một giá trị IdLoai tồn tại trong LoaiMon, duy trì tính toàn vẹn tham chiếu.
- Ý nghĩa thực tiễn:
 - + Mối quan hệ này cho phép tổ chức thực đơn theo danh mục. Ví dụ, loại "Món chính" (IdLoai = 1) có thể bao gồm nhiều món như "Phở bò", "Bún chả", trong khi mỗi món chỉ thuộc về một loại duy nhất.
 - + Hỗ trợ truy vấn như lấy tất cả món ăn thuộc loại "Đồ uống" bằng câu lệnh: SELECT * FROM MonAn WHERE IdLoai = 3.
- Biểu diễn trong ERD:
 - + Dòng nối từ LoaiMon đến MonAn.
 - + Ký hiệu: "1" ở đầu LoaiMon (một loại) và "N" ở đầu MonAn (nhiều món).

- **Mối quan hệ DatBan - KhachHang:**

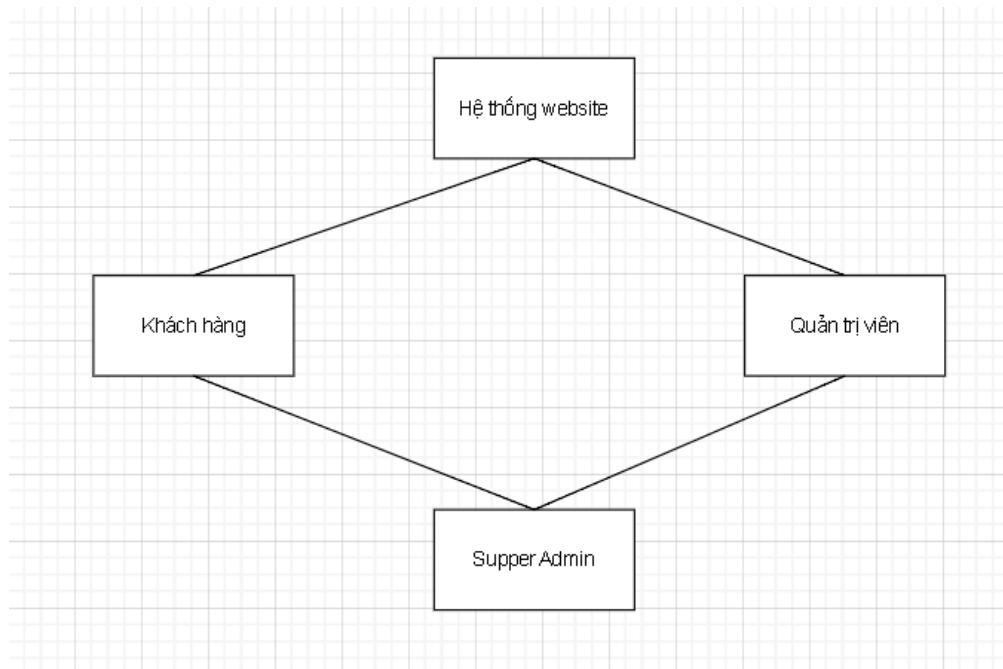
- Loại quan hệ 1-n: Một bản ghi trong KhachHang (khách hàng) có thể liên kết với nhiều bản ghi trong DatBan (đặt bàn), nhưng mỗi bản ghi trong DatBan chỉ thuộc về một khách hàng (nếu khôi phục KhachHang).
- Cơ chế thực hiện:
 - + Khóa chính: IdKH trong KhachHang (nếu khôi phục).
 - + Khóa ngoại: IdKH trong DatBan, được khai báo với ALTER TABLE DatBan ADD CONSTRAINT FK_DatBan_KhachHang FOREIGN KEY (IdKH) REFERENCES KhachHang(IdKH).
 - + Hiện tại, vì KhachHang bị xóa bằng DROP TABLE KhachHang, mối quan hệ này không còn hiệu lực, và cột IdKH trong DatBan trở thành cột không sử dụng.
- Ý nghĩa thực tiễn:
 - + Nếu khôi phục, mối quan hệ này cho phép quản lý lịch sử đặt bàn của từng khách hàng, ví dụ một khách hàng "Nguyễn Văn A" có thể đặt nhiều lần với các IdDatBan khác nhau.

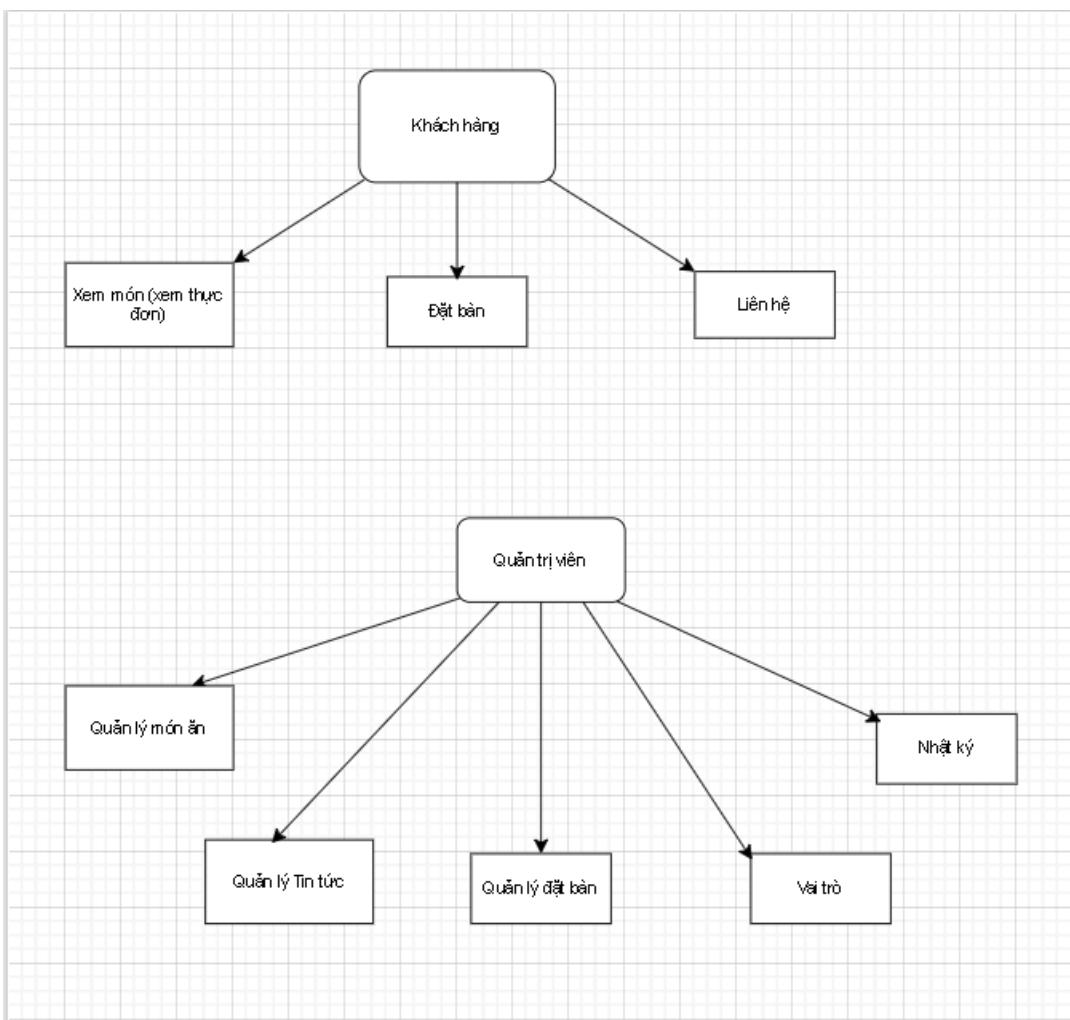
- + Hỗ trợ tính năng như xem lịch sử đặt bàn của khách hàng: SELECT * FROM DatBan WHERE IdKH = 1.
- Biểu diễn trong ERD:
 - + Dòng nối từ KhachHang đến DatBan (nếu khôi phục).
 - + Ký hiệu: "1" ở đầu KhachHang và "N" ở đầu DatBan.
 - + Lưu ý: Hiện tại, DatBan không có mối quan hệ trực tiếp nào do thiếu KhachHang.
- **Mối quan hệ AdminUsers - AdminLogs:**
- Loại quan hệ 1-n: Một bản ghi trong AdminUsers (tài khoản quản trị) có thể liên kết với nhiều bản ghi trong AdminLogs (nhật ký thao tác), nhưng mỗi bản ghi trong AdminLogs chỉ thuộc về một admin.
- Cơ chế thực hiện:
 - + Khóa chính: IdAD trong AdminUsers.
 - + Khóa ngoại: IdAD trong AdminLogs, được khai báo với FOREIGN KEY (IdAD) REFERENCES AdminUsers(IdAD).
 - + Điều này đảm bảo rằng mỗi hành động trong AdminLogs được gắn với một admin cụ thể, và giá trị IdAD phải tồn tại trong AdminUsers.
- Ý nghĩa thực tiễn:
 - + Mối quan hệ này cho phép theo dõi các hoạt động của từng admin, ví dụ admin với IdAD = 1 (username "admin") có thể có nhiều hành động như "Đã thêm món Phở bò" trong AdminLogs.
 - + Hỗ trợ truy vấn: SELECT a.Username, l.HanhDong FROM AdminUsers a JOIN AdminLogs l ON a.IdAD = l.IdAD.
- Biểu diễn trong ERD:
 - + Dòng nối từ AdminUsers đến AdminLogs.
 - + Ký hiệu: "1" ở đầu AdminUsers và "N" ở đầu AdminLogs.
- **Mối quan hệ AdminUsers - AdminRoles và Roles - AdminRoles**
- Loại quan hệ: n-m (Nhiều-nhiều): Một bản ghi trong AdminUsers có thể liên kết với nhiều bản ghi trong Roles (vai trò), và một bản ghi trong Roles có thể thuộc về nhiều bản ghi trong AdminUsers.
- Cơ chế thực hiện:
 - + Khóa chính: (IdAD, RoleID) trong AdminRoles (kết hợp làm khóa chính).
 - + Khóa ngoại: IdAD trong AdminRoles tham chiếu đến AdminUsers(IdAD), RoleID trong AdminRoles tham chiếu đến Roles(RoleID).
 - + Bảng trung gian AdminRoles được sử dụng để giải quyết mối quan hệ n-m, với các câu lệnh FOREIGN KEY (IdAD) REFERENCES AdminUsers(IdAD) và FOREIGN KEY (RoleID) REFERENCES Roles(RoleID).
- Ý nghĩa thực tiễn:

- + Mỗi quan hệ này cho phép phân quyền linh hoạt. Ví dụ, admin với IdAD = 1 có thể được gán vai trò "Quản lý món ăn" (RoleID = 1) và "Quản lý đặt bàn" (RoleID = 2).
- + Hỗ trợ truy vấn: SELECT au.Username, r.RoleName FROM AdminUsers au JOIN AdminRoles ar ON au.IdAD = ar.IdAD JOIN Roles r ON ar.RoleID = r.RoleID WHERE au.IdAD = 1.
- Biểu diễn trong ERD:
 - + Dòng nối từ AdminUsers đến AdminRoles và từ Roles đến AdminRoles.
 - + Ký hiệu: "N" ở cả hai đầu (AdminUsers và Roles), với AdminRoles là bảng trung gian.

Tất cả các mối quan hệ đều sử dụng khóa ngoại để đảm bảo dữ liệu nhất quán (referential integrity). Ví dụ, nếu xóa một LoaiMon, các MonAn liên quan sẽ bị ảnh hưởng (tùy thuộc vào hành động ON DELETE trong SQL). Mọi quan hệ n-m giữa AdminUsers và Roles cho phép mở rộng vai trò mà không cần thay đổi cấu trúc cơ bản. Thiếu mối quan hệ với KhachHang làm giảm khả năng quản lý khách hàng lâu dài. Khuyến nghị khôi phục KhachHang và liên kết với DatBan để hoàn thiện.

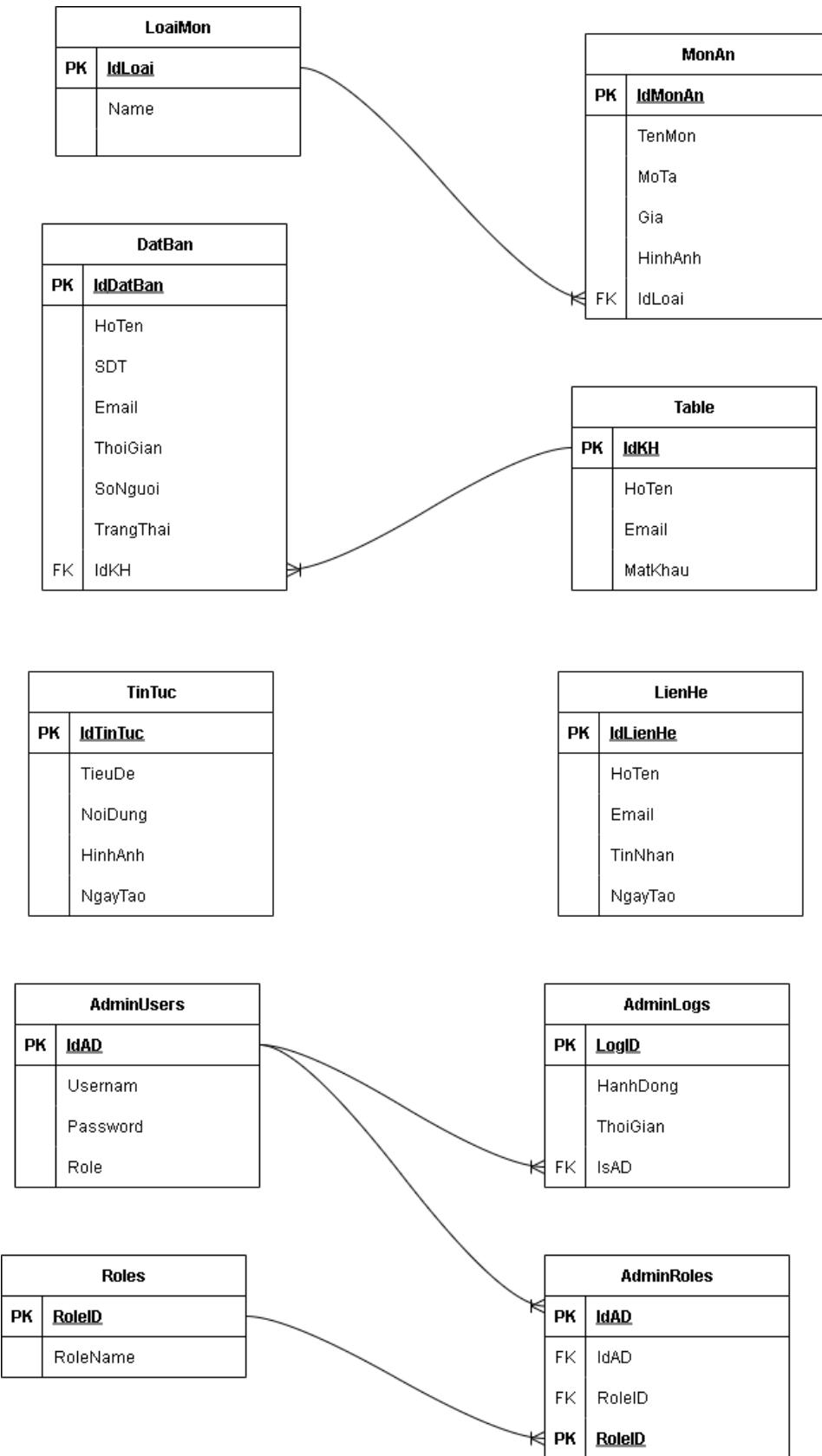
2.2.8. Biểu đồ Use Case tổng thể





Hình 2. Biểu đồ use case tổng thể

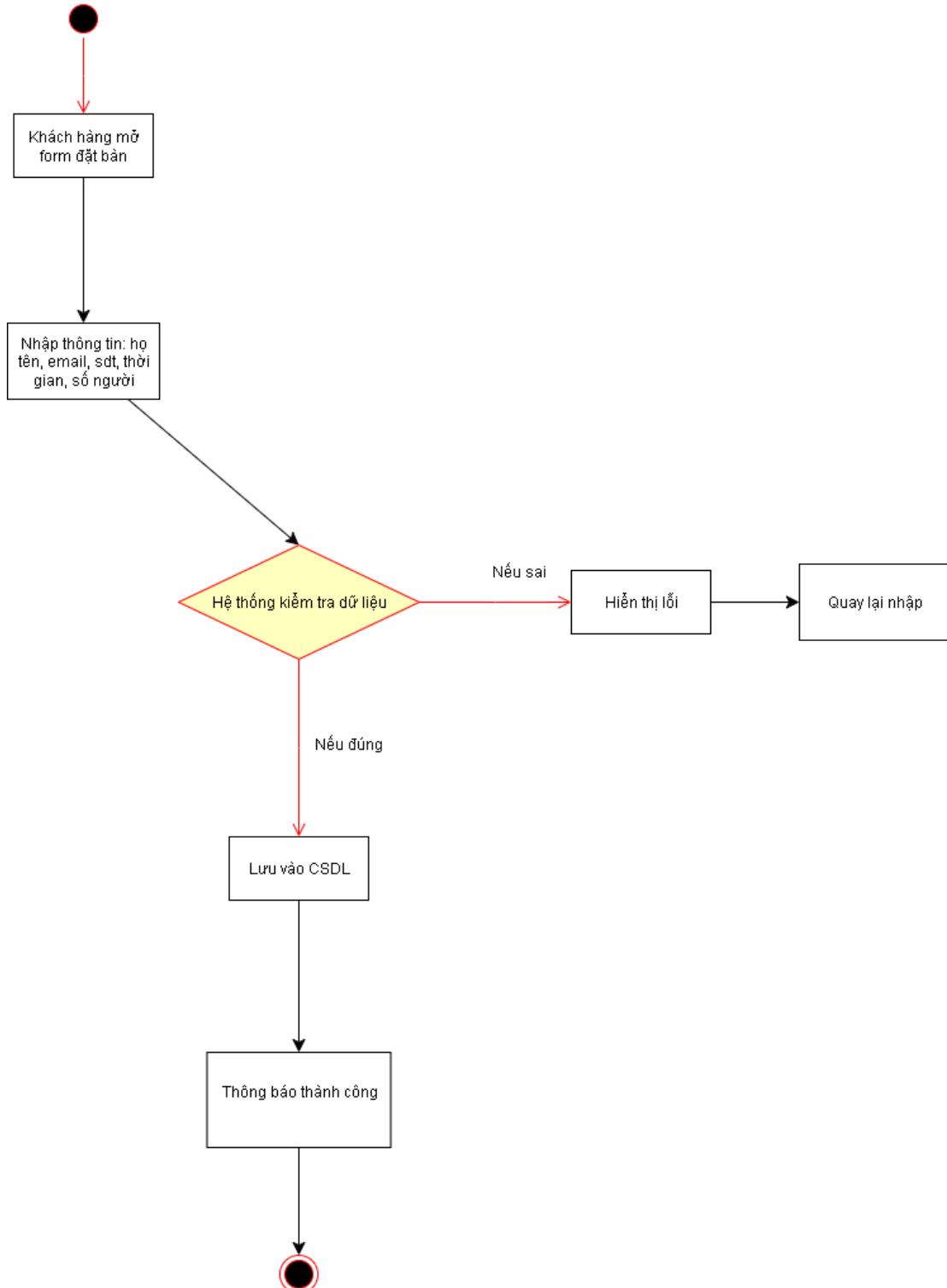
2.2.9. Sơ đồ quan hệ (ERD)



Hình 3. Sơ đồ ERD

2.2.10. Một số sơ đồ hoạt động

- Sơ đồ hoạt động chức năng đặt bàn



Hình 4. Sơ đồ hoạt động đặt bàn

- Sơ đồ hoạt động chức năng quản lý món ăn



Admin đăng nhập

Chọn mục quản lý
món ăn

Thêm/Sửa/Xóa món
ăn

Hệ thống cập nhật
CSDL

Hiển thị danh sách
mới



Hình 4. Sơ đồ hoạt động quản lý món ăn

2.3. Tiến hành tạo bảng trong SQL Server

```
-- Sử dụng master để tạo CSDL
USE master;
GO

-- Tạo cơ sở dữ liệu
CREATE DATABASE WebsiteGioiThieuNhaHang;
GO

-- Dùng CSDL vừa tạo
USE WebsiteGioiThieuNhaHang;
GO

-- 1. Loại món ăn
CREATE TABLE LoaiMon (
    IdLoai INT PRIMARY KEY IDENTITY(1,1),
    Name NVARCHAR(100) NOT NULL
);

-- 2. Món ăn
CREATE TABLE MonAn (
    IdMonAn INT PRIMARY KEY IDENTITY(1,1),
    TenMon NVARCHAR(150) NOT NULL,
    MoTa TEXT,
    Gia DECIMAL(10,2) NOT NULL,
    HinhAnh NVARCHAR(255),
    IdLoai INT,
    FOREIGN KEY (IdLoai) REFERENCES LoaiMon(IdLoai)
);

-- 3. Đặt bàn
CREATE TABLE DatBan (
    IdDatBan INT PRIMARY KEY IDENTITY(1,1),
    HoTen NVARCHAR(100) NOT NULL,
    SDT VARCHAR(20) NOT NULL,
    Email VARCHAR(100),
    ThoiGian DATETIME NOT NULL,
    SoNguoi INT NOT NULL,
    TrangThai NVARCHAR(50) DEFAULT 'Chờ xác nhận'
);

-- 4. Tin tức
CREATE TABLE TinTuc (
    IdTinTuc INT PRIMARY KEY IDENTITY(1,1),
    TieuDe NVARCHAR(200) NOT NULL,
    Noidung TEXT,
    HinhAnh NVARCHAR(255),
    NgayTao DATETIME DEFAULT GETDATE()
);

-- 5. Liên hệ
CREATE TABLE LienHe (
    IdLienHe INT PRIMARY KEY IDENTITY(1,1),
    HoTen VARCHAR(100) NOT NULL,
    Email VARCHAR(100),
    TinNhan TEXT,
    NgayTao DATETIME DEFAULT GETDATE()
);

-- 6. Tài khoản quản trị
CREATE TABLE AdminUsers (
    IdAD INT PRIMARY KEY IDENTITY(1,1),
    Username VARCHAR(50) NOT NULL UNIQUE,
    Password VARCHAR(255) NOT NULL,
    --PasswordHash VARCHAR(255) NOT NULL,
    Role VARCHAR(50) DEFAULT 'staff'
);

-- 7. Nhật ký thao tác của admin
CREATE TABLE AdminLogs (
    LogID INT PRIMARY KEY IDENTITY(1,1),
    IdAD INT,
    HanhDong TEXT,
    ThoiGian DATETIME DEFAULT GETDATE(),
    FOREIGN KEY (IdAD) REFERENCES AdminUsers(IdAD)
);

-- 8. Vai trò
CREATE TABLE Roles (
    RoleID INT PRIMARY KEY IDENTITY(1,1),
    RoleName VARCHAR(50) NOT NULL
);

-- 9. Gán vai trò cho admin
CREATE TABLE AdminRoles (
    IdAD INT,
    RoleID INT,
    PRIMARY KEY (IdAD, RoleID),
    FOREIGN KEY (IdAD) REFERENCES AdminUsers(IdAD),
    FOREIGN KEY (RoleID) REFERENCES Roles(RoleID)
);
```

```

-- 10. Khách hàng
CREATE TABLE KhachHang (
    IdKH INT PRIMARY KEY IDENTITY(1,1),
    HoTen NVARCHAR(100) NOT NULL,
    Email VARCHAR(100) NOT NULL UNIQUE,
    PasswordHash VARCHAR(255) NOT NULL,
    NgayDangKy DATETIME DEFAULT GETDATE()
);

```

Hình 5. Tạo các cơ sở dữ liệu trên SQL Server

```

-- Thêm đặt bàn
INSERT INTO DatBan (HoTen, SDT, Email, ThoiGian, SoNguoi, DaXacNhan, TrangThai) VALUES
('Nguyễn Văn A', '0909123456', 'a@gmail.com', '2025-07-02 18:00:00', 4, 0, N'Chờ xác nhận');

SET IDENTITY_INSERT LoaiMon ON;
INSERT INTO LoaiMon (IdLoai, Name) VALUES
(1, N'Món chính'),
(2, N'Món tráng miệng'),
(3, N'Dessert');
SET IDENTITY_INSERT LoaiMon OFF;
-- Thêm món ăn
INSERT INTO MonAn (TenMon, MoTa, Gia, HìnhAnh, IdLoai) VALUES
(N'Phở bò', N'Phở bò với nước dùng đậm đà nấu trong 12 giờ, thịt bò tươi ngon cùng các loại gia vị đặc trưng.', 120.000, N'/Anhnhahang/nhaHangMenuPhoBo.jpeg', 1),
(N'Bánh flan', N'Bánh flan ngọt', 20000, N'/Anhnhahang/nhaHangMenuBanhFlan.png', 2),
(N'Trà đào', N'Trà đào với hương vị ngọt', 30000, N'/Anhnhahang/nhaHangMenuTraDau.png', 3),
(N'Bún chả', N'Bún chả với thịt nướng thơm lừng, nước mắm chua ngọt đặc trưng và rau sống tươi ngon.', 95000, N'/Anhnhahang/nhaHangMenuBunCha.jpeg', 1),
(N'Gỏi cuốn tôm thịt', N'Gỏi cuốn tươi mát với tôm, thịt heo, bún, rau thơm cuốn trong bánh tráng, kèm nước chấm.', 85000, N'/Anhnhahang/nhaHangMenuGoiCuon.png', 1),
(N'Bánh Xèo', N'Thường thức hương vị đậm chất miền Trung với bánh xèo nhân tôm thịt đặc biệt.', 100000, N'/Anhnhahang/nhaHangMenuBanhXeo.jpeg', 1);
DELETE FROM MonAn;

```

Hình 6. Thêm các dữ liệu vào các bảng

CHƯƠNG 3: THIẾT KẾ VÀ PHÂN TÍCH GIAO DIỆN

3.2. Thiết kế giao diện

Tiêu chí giao diện dễ dùng, dễ nhìn, nổi bật món ăn. Tối ưu được giao diện có thể hiển thị tốt trên điện thoại. Sử dụng màu chủ đạo hài hòa, tông trầm ấm, phù hợp với tên của nhà hàng.

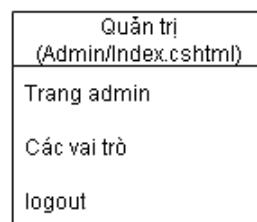
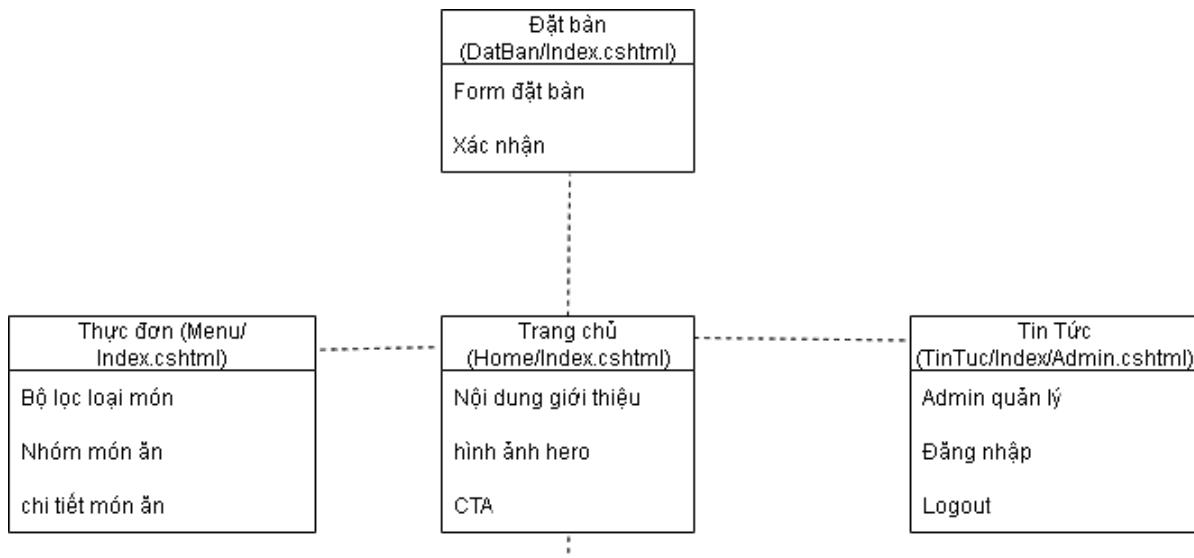
Giao diện website giới thiệu nhà hàng gồm các thành phần chính sau:

- Phần đầu trang <Header>: cũng là phần giao diện chung, xuất hiện ở tất cả các trang, bao gồm logo của nhà hàng, tên của nhà hàng “Hương Vị Việt”. Thanh điều hướng (Trang chủ, Thực , Tin tức, Liên hệ, Đặt bàn, Trang quản trị). Đối với giao diện website trên điện thoại, thanh điều hướng nằm dọc bên trái giúp người dùng dễ dàng di chuyển giữa các trang.
- Banner/ hero section: gồm hình ảnh nổi bật, có kích thước lớn ở đầu trang, cạnh đó bên dưới hình ảnh là một mô tả ngắn.

Chào mừng đến với nhà hàng HƯƠNG VỊ VIỆT

Hình 7. Thanh điều hướng (Header của website)

3.2.1. Sơ đồ luồng giao diện



Hình 8. Sơ đồ luồng giao diện

3.2.2. Trang chủ

- là trang giới thiệu nhà hàng. Trang chủ thể hiện các mục: giới thiệu sơ lược về nhà hàng, thời gian hoạt động của nhà hàng, kèm slogan của nhà hàng kèm ngay dưới có các nút điều hướng ẩn cam nổi bật, kêu gọi người dùng đặt bàn.

sang trang Thực đơn và trang Đặt bàn, bên dưới có phần giới thiệu về nhà hàng kèm lời bạt và nút điều hướng đến trang tin tức, trang chủ còn hiển thị 3 món nổi bật: Phở bò, Bún chả Hà Nội, Gỏi cuốn tôm thịt, hiển thị dạng thẻ với hình ảnh, mô tả ngắn gọn, giá tiền và nút điều hướng sang trang Thực đơn. Kết thúc cho Trang chủ là một lời mời “Đặt bàn ngay hôm nay” cùng nút điều hướng sang trang đặt bàn.

- Nút “Quản trị” được tô màu, dễ nhận biết.
- Màu sắc chủ đạo: trắng, vàng cam, đen, tạo cảm giác sạch sẽ và trang trọng.
- Dễ dàng truy cập các mục khác từ thanh điều hướng trên cùng.

The screenshot displays two pages of the website 'Hương Vị Việt'.

Main Page:

- Header:** Includes the logo 'Hương Vị Việt', navigation links (Trang chủ, Thực đơn, Đặt bàn, Tin tức, Liên hệ, Quản trị, Đăng nhập, Đăng ký), and a search bar.
- Welcome Message:** 'Chào mừng đến với nhà hàng HƯƠNG VỊ VIỆT'
- Section: Hương Vị Việt Nam Truyền Thống**
 - Image: A night view of the restaurant's exterior.
 - Description: Khám phá tinh hoa ẩm thực Việt Nam với những món ăn đặc sắc được chế biến từ nguyên liệu tươi ngon nhất.
 - Buttons: Xem Thực Đơn and Đặt Bàn Ngay.
- Section: Về Nhà Hàng Chúng Tôi**
 - Image: A night view of the restaurant's interior.
 - Description: Nhà hàng Hương Vị Việt tự hào mang đến cho quý khách những trải nghiệm ẩm thực Việt Nam đích thực.
 - Description: Với hơn 15 năm kinh nghiệm, chúng tôi đã phục vụ hàng ngàn thực khách với những món ăn truyền thống được chế biến từ nguyên liệu tươi ngon nhất.
 - Button: Tìm Hiểu Thêm.
- Section: Món Ăn Đặc Sắc**
 - Image: Two dishes: Phở Bò Truyền Thống and Bún Chả Hà Nội.
 - Description: Phở bò với nước dùng đậm đà nấu trong 12 giờ, thịt bò tươi ngon cùng các loại gia vị đặc trưng.
 - Price: 120.000đ
 - Button: Đặt Bàn Ngay.
 - Image: Gỏi Cuốn Tôm Thịt.
 - Description: Gỏi cuốn tươi mới với tôm, thịt heo, bún, rau thơm cuốn trong bánh tráng, kèm nước chấm.
 - Price: 85.000đ
 - Button: Đặt Bàn Ngay.
- Footer:** 'Đặt Bàn Ngay Hôm Nay' button.

Hình 9. Trang chủ và trang Thực đơn

3.2.3. Trang thực đơn

- Bố cục: tiêu đề “Thực đơn” nổi bật, bộ lọc danh mục 3 nút tương ứng với 3

loại món: Món chính, Tráng miệng, Đồ uống.

- Danh sách món ăn: mỗi món ăn hiển thị dạng thẻ: hình ảnh, tên món, mô tả, giá tiền. Các món có sẵn theo danh sách: Món chính: Phở bò, Bún chả, Gỏi cuốn tôm thịt, Bánh xèo. Tráng miệng: Bánh flan. Đồ uống: Trà đào.

3.2.4. Trang đặt bàn

- Tiêu đề: “Đặt bàn trực tuyến”.
- Biểu mẫu đặt bàn: Họ tên, Số điện thoại, Email, Ngày đặt bàn (có chọn ngày và giờ), số người, nút “Đặt Bàn”.
- Danh sách đặt bàn (dành cho admin): Bảng liệt kê các đặt bàn với các cột: Họ tên, SĐT, Email, Thời gian, Số người, Trạng thái (hiển thị trạng thái “Chờ xác nhận”).

Hình 10. Trang Đặt bàn

3.2.5. Trang tin tức

Hiện các thông báo tin tức, khuyến mãi hiện có mà nhà hàng đang khởi chạy. Chia thành hai phần tin tức và khuyến mãi.

- Phần tin tức: thông báo về các sự kiện sắp tới của nhà hàng: ra mắt món mới “Bánh xèo miền trung”.
- Phần khuyến mãi: các ưu đãi khi khách hàng đến sử dụng dịch vụ tại nhà hàng: Ưu đãi tháng 6 giảm 20% cho hóa đơn trên 500k, giảm 10% khi thanh toán bằng các thẻ ngân hàng VIB, TECHCOMBANK, TPBANK.

Tin Tức



Ra mắt món mới: Bánh xèo miền Trung

Thưởng thức hương vị đậm chất miền Trung với bánh xèo nhân tôm thịt đặc biệt.

Khuyến Mãi

Ưu đãi tháng 6: Giảm 20% cho đơn trên 500k

Chương trình khuyến mãi hấp dẫn đang chờ đón bạn từ ngày 1 đến 30 tháng 6!

Ưu đãi Siêu Hời: Giảm 10% khi thanh toán bằng các Thẻ Ngân Hàng VIB - TECHCOMBANK - VPBANK - TPBANK

Khách hàng được giảm giá khi dùng thanh toán bằng thẻ ngân hàng liên kết với nhà hàng.



© 2025 - Nhà hàng Hương Vị Việt | [Chính sách](#)

Hình 11. trang Tin tức

3.2.6. Trang liên hệ

Khách hàng truy cập trang Liên hệ, gửi tin nhắn phản ánh, bình luận về nhà hàng và gửi đến nhà hàng, kín đáo không công khai.

- Tiêu đề: “Liên hệ” được hiển thị bằng màu cam, nổi bật ở đầu trang. Giao diện bao gồm các trường nhập liệu và thông tin liên hệ, với thiết kế đơn giản và rõ ràng.
- Biểu mẫu liên hệ: Họ tên, Email, nội dung tin nhắn, nút “Gửi liên hệ” màu xám, nằm dưới các trường nhập liệu, cho phép người dùng gửi thông tin.

- Phần "Địa chỉ" hiển thị một bản đồ nhỏ (có thể là từ Google Maps) với chú thích "Kênh đào kênh" và tọa độ. Bản đồ cung cấp vị trí địa lý, giúp người dùng xác định địa điểm liên hệ.

Liên hệ

Họ tên

Email

Tin nhắn

Gửi liên hệ

Địa chỉ



© 2025 - Nhà hàng Hương Vị Việt | [Chính sách](#)

Hình 12. Trang Liên hệ

3.2.7. Trang quản trị

- Giao diện trong ảnh là một trang đăng nhập (login page) bằng tiếng Việt, có thể thuộc về một hệ thống quản trị hoặc trang web chính thức.
- Tiêu đề “Đăng nhập” hiển thị rõ ràng.
- Phần đăng nhập: hai thanh nhập liệu “Email” và “Mật khẩu”, Nút “Đăng nhập” có màu xanh lá, nổi bật để người dùng nhấn thực hiện hành động.
- Liên kết bổ sung: sau khi đăng nhập thành công, nhận thông báo “Chào mừng đến trang quản trị” với hai tùy chọn: “Quản lý món ăn” (liên kết màu xanh), “Quản lý đặt bàn” (liên kết màu xanh). Các liên kết này gợi ý rằng trang dành cho quản trị viên với các chức năng quản lý tài chính và đơn hàng.

Đăng nhập

Email

admin

Mật khẩu

.....

Đăng nhập

Hình 13. trang Quản trị trước khi đăng nhập

Chào mừng đến trang quản trị

Chọn chức năng quản lý bên dưới:

- Quản lý món ăn
- Quản lý đặt bàn

Hình 14 trang Quản trị sau khi đăng nhập

3.2.8. Trang quản trị Quản lý đặt bàn

- Tiêu đề "Quản lý đặt bàn" được hiển thị rõ ràng, tập trung quản lý đặt bàn.
- Bảng danh sách đặt bàn bao gồm các cột: Họ tên, Số điện thoại, Email, Ngày, Giờ, Số người, Ghi chú, Trạng thái, và Hành động.
- Các đặt bàn được liệt kê:
 - + Đăng Thị Hoài Trần: 0357566231, Nhan123@gmail.com, 7/4/2025, 3 giờ, Chỗ đã nh�n, "Xác nhận" và "Xóa".
 - + Đăng Thị Hoài Trần: 0357566231, Nhan123@gmail.com, 7/4/2025, 3 giờ, Chỗ đã nh�n, "Xác nhận" và "Xóa".
 - + Nguyễn Văn A: 0909123456, a@gmail.com, 7/2/2025, 4 giờ, Chỗ đã nh�n, "Xác nhận" và "Xóa".
- Tùy chọn "Xác nhận" (xanh) và "Xóa" (đỏ) cho mỗi đặt bàn.
- Thiết kế tổng thể: Giao diện sử dụng bảng để hiển thị thông tin đặt bàn, với màu sắc rõ ràng (xanh cho "Xác nhận", đỏ cho "Xóa").
- Bố cục gọn gàng, phù hợp cho quản trị viên quản lý lịch đặt bàn.



Quản lý đặt bàn

Họ tên	Số điện thoại	Email	Ngày	Giờ	Số người	Ghi chú	Trạng thái	Hành động
Đặng Thị Hoài Trần	0357566231	Nhan123@gmail.com	7/4/2025	3	Chờ xác nhận	Chưa	Xác nhận Xóa	
Đặng Thị Hoài Trần	0357566231	Nhan123@gmail.com	7/4/2025	3	Chờ xác nhận	Chưa	Xác nhận Xóa	
Nguyễn Văn A	0909123456	a@gmail.com	7/2/2025	4	Chờ xác nhận	Chưa	Xác nhận Xóa	

© 2025 - Nhà hàng Hương Vị Việt | [Chính sách](#)

Hình 15. trang Quản lý phần quản lý đặt bàn

3.2.9. Trang quản trị Quản lý món ăn

- Tiêu đề "Quản lý món ăn" được hiển thị rõ ràng, tập trung quản lý món ăn trong thực.
- Chức năng chính: Nút "Thêm món ăn" màu xanh, cho phép người dùng thêm món mới vào danh sách. Bảng danh sách món ăn bao gồm các cột: Tên món, Giá, Mô tả, Ảnh, và Hành động.
- Các món được liệt kê:
 - + Phở bò: 120.00 VNĐ, mô tả về phở bò, có ảnh, trạng thái "Sửa" và "Xóa".
 - + Bánh flan: 20000.00 VNĐ, mô tả về bánh flan, có ảnh, trạng thái "Sửa" và "Xóa".
 - + Trà đào: 30000.00 VNĐ, mô tả về trà đào, có ảnh, trạng thái "Sửa" và "Xóa".
 - + Bún chả: 95000.00 VNĐ, mô tả về bún chả, có ảnh, trạng thái "Sửa" và "Xóa".
 - + Gỏi cuốn tôm thịt: 85000.00 VNĐ, mô tả về gỏi cuốn, có ảnh, trạng thái "Sửa" và "Xóa".
 - + Bánh xèo: 100000.00 VNĐ, mô tả về bánh xèo, có ảnh, trạng thái "Sửa" và "Xóa".
- Thiết kế tổng thể: Giao diện sử dụng bảng để hiển thị dữ liệu, với màu sắc nổi bật (xanh cho "Sửa", đỏ cho "Xóa"). Ảnh minh họa cho mỗi món ăn tăng tính trực quan. Phù hợp cho quản trị viên quản lý thực đơn.



Quản lý món ăn

[Thêm món ăn](#)

Tên món	Giá	Mô tả	Ảnh	Hành động
Phở bò	120.00	Phở bò với nước dùng đậm đà nấu trong 12 giờ, thịt bò tươi ngon cùng các loại gia vị đặc trưng.		Sửa Xóa
Bánh flan	20000.00	Món tráng miệng ngọt		Sửa Xóa
Trà đào	30000.00	Đồ uống giải khát		Sửa Xóa
Bún chả	95000.00	Bún chả với thịt nướng thơm lừng, nước mắm chua ngọt đặc trưng và rau sống tươi ngon.		Sửa Xóa
Gỏi cuốn tôm thịt	85000.00	Gỏi cuốn tươi mát với tôm, thịt heo, bún, rau thơm cuộn trong bánh tráng, kèm nước chấm.		Sửa Xóa
Bánh Xèo	100000.00	Thường thức hương vị đậm chất miền Trung với bánh xèo nhân tôm thịt đặc biệt.		Sửa Xóa

© 2025 - Nhà hàng Hương Vị Việt | [Chính sách](#)

Hình 16. trang Quản trị phần quản lý món ăn

3.2.10. Trang đăng ký

- Tiêu đề "Đăng ký tài khoản" được hiển thị rõ ràng, cho biết mục đích của trang là tạo tài khoản mới.
- Phần nhập thông tin: Có bốn trường nhập liệu: Họ tên, Email, Mật khẩu, Nhập lại mật khẩu. Các trường này yêu cầu thông tin cơ bản để đăng ký, bao gồm tên, email, và xác nhận mật khẩu.
- Nút "Đăng ký" màu xanh dương, nổi bật, cho phép người dùng hoàn tất quá trình đăng ký sau khi nhập thông tin.

Đăng ký tài khoản

Họ tên

Email

Mật khẩu

Nhập lại mật khẩu

Đăng ký

Hình 17. trang Đăng ký tài khoản

3.2.11. Trang đăng nhập

- Tiêu đề "Đăng nhập" được hiển thị rõ ràng.
- Phần nhập thông tin đăng nhập: tên tài khoản (địa chỉ email của khách hàng), mật khẩu, nút "Đăng nhập" màu cam, nổi bật, cho phép người dùng thực hiện hành động đăng nhập.
-

The screenshot shows the login page of a website. At the top, there is a navigation bar with the logo 'Hương Vị Việt' and links for Trang chủ, Thực đơn, Đặt bàn, Tin tức, Liên hệ, Quản trị (highlighted in red), Đăng nhập, and Đăng ký. The main content area has a light orange background. It features a heading 'Đăng nhập' and a note '< a>tài khoản email của bạn'. Below this are two input fields: one containing 'admin' and another containing '.....'. At the bottom is a large orange button labeled 'Đăng nhập'.

Hình 18. trang Đăng nhập



Trang chủ

Thực đơn

Đặt bàn

Tin tức

Liên hệ

Quản trị



Đăng xuất

Đặt bàn trực tuyến

Họ Tên

Số điện thoại

Email

Hình 19. sau khi đăng nhập hoặc đăng ký.

3.3. Một số giao diện khác

The screenshot shows a mobile application interface for booking a table. At the top, there is a header with the logo 'Hương Vị Việt' and a menu icon (three horizontal lines). Below the header, the title 'Đặt bàn trực tuyến' (Online Table Booking) is displayed. The form consists of several input fields:

- Họ Tên (Name): An empty text input field.
- Số điện thoại (Phone number): An empty text input field.
- Email: An empty text input field.
- Thời gian (Time): A date/time input field showing 'mm/dd/yyyy --:-- --' with a calendar icon to its right.
- Số Người (Number of people): An empty text input field.

At the bottom left of the form area is a blue button labeled 'Đặt bàn' (Book Table).

Đơn của tôi

Ngày	Giờ	Số người	Trạng thái
7/9/2025	1:00 PM	43	Chờ xác nhận

Hình 20. Hiển thị danh sách “đơn đặt bàn của tôi” sau khi khách hàng đăng nhập và đặt bàn

Đặt bàn trực tuyến

Bạn đã đăng xuất.

Họ Tên

Số điện thoại

Email

Thời gian

mm/dd/yyyy --:-- --



Số Người

Đặt bàn

Bạn đã đăng xuất.

Bạn đã đăng xuất.

Đơn của tôi

Bạn chưa có đơn nào.

Hình 21. Hiển thị thông báo khi khách hàng đăng xuất thành công

CHƯƠNG 4: TRIỂN KHAI BACK - END

4.1. Kiến trúc tổng quan

Hệ thống được xây dựng theo mô hình MVC (Model - View - Controller) nhằm tách biệt dữ liệu, xử lý nghiệp vụ và giao diện. Back-end được lập trình trong template ASP.NET Core Web App (Model-View-Controller), framework ASP.NET Core MVC, dùng ngôn ngữ C# (.NET). Giúp website chạy trên đa nền tảng

Các thành phần back - end:

- Model: xử lý dữ liệu và tương tác cơ sở dữ liệu.
- Controller: xử lý yêu cầu từ client, điều phối luồng dữ liệu.
- View: trả về giao diện hoặc JSON cho client.

Mô tả quy trình **Client → Server → Database → Client**.

Người dùng gửi yêu cầu đến Controller, Controller xử lý & gọi Model. Model thao tác dữ liệu trong Database và trả kết quả về Controller, rồi trả về Client.

4.2. Model

Trong dự án này, Model là các lớp C# đại diện cho bảng dữ liệu trong cơ sở dữ liệu.

Các Model chịu trách nhiệm:

- Định nghĩa cấu trúc dữ liệu.
- Gán ràng buộc (validation) cho các trường dữ liệu.
- Tạo mối quan hệ giữa các bảng.
- Được Entity Framework Core dùng để phản xạ với cơ sở dữ liệu.

Danh sách các Model và vai trò:

Model	Vai trò
LoaiMon	Quản lý loại món ăn
MonAn	Quản lý chi tiết món ăn
DatBan	Quản lý đặt bàn của khách
TinTuc	Quản lý tin tức
LienHe	Quản lý phản hồi liên hệ
KhachHang	Quản lý thông tin khách hàng
AdminUser	Quản lý tài khoản admin
AdminLogs	Nhật ký thao tác admin
AdminRoles	Gán vai trò admin
Roles	Danh sách vai trò
AdminLogin, KhachHangLoginModel, KhachHangRegisterModel	Model phục vụ form đăng nhập, form đăng ký

4.2.1. Model LoaiMon

- Thuộc tính:
 - IdLoai (int, PK): khóa chính.
 - Name (string): Tên loại món ăn.
- Vai trò: Phân loại món ăn.
- Quan hệ: 1 - nhiều với MonAn

```
public class LoaiMon
```

```
{
```

```
[Key]
```

```
public int IdLoai { get; set; }  
public string Name { get; set; }  
public ICollection<MonAn> MonAns { get; set; }
```

```
}
```

4.2.2. Model MonAn

- Thuộc bảng dữ liệu: Món ăn
- Thuộc tính:
 - IdMonAn (int, PK): mã món ăn.
 - TenMon (string): tên món ăn.
 - MoTa(string): mô tả chi tiết món ăn.
 - Gia (decimal): giá bán của món.
 - Hinhanh (string): Đường dẫn ảnh minh họa.
 - IdLoai (FK): Khóa ngoại liên kết LoaiMon.
- Quan hệ: Thuộc về 1 LoaiMon.
- Vai trò: Quản lý các món ăn cụ thể trong thực đơn.

```
public class MonAn
```

```
{
```

```
[Key]
```

```
public int IdMonAn { get; set; }  
public string TenMon { get; set; }  
public string MoTa { get; set; }  
[Column(TypeName = "decimal(10,2)")]  
public decimal Gia { get; set; }  
public string Hinhanh { get; set; }  
// FK  
public int IdLoai { get; set; }  
public LoaiMon? LoaiMon { get; set; }
```

```
}
```

4.2.3. Model DatBan

- Bảng dữ liệu: Đặt bàn
- Thuộc tính:

- IdDatBan (int, PK): Mã đặt bàn.
- HoTen, SDT, Email: Thông tin khách đặt.
- ThoiGian: Thời điểm đặt bàn.
- SoNguoi: Số lượng khách.
- TrangThai: Trạng thái đặt bàn (Chờ xác nhận, Đã xác nhận...).
- DaXacNhan: Xác nhận (true/false).
- IdKH (FK): Khóa ngoại đến KhachHang.
- Quan hệ: Nhiều đặt bàn có thể thuộc về 1 khách hàng.
- Vai trò: Ghi nhận thông tin đặt chỗ của khách.

public class DatBan

{

[Key]

public int IdDatBan { get; set; }

[Required(ErrorMessage = "Vui lòng nhập họ tên")]

public string HoTen { get; set; }

[Required(ErrorMessage = "Vui lòng nhập số điện thoại")]

public string SDT { get; set; }

[Required(ErrorMessage = "Vui lòng nhập email")]

[EmailAddress(ErrorMessage = "Email không hợp lệ")]

public string Email { get; set; }

[Required]

// public DateTime? ThoiGian { get; set; }

[Display(Name = "Ngày đặt bàn")]

[DataType(DataType.Date)] // Gợi ý Razor dùng input date

//[Required(ErrorMessage = "Vui lòng chọn ngày đặt bàn")]

public DateTime ThoiGian { get; set; }

public int SoNguoi { get; set; }

public string TrangThai { get; set; } = "Chờ xác nhận";

public bool DaXacNhan { get; set; } // nullable bool?

public int IdKH { get; set; }

}

4.2.4. Model TinTuc

- Thuộc tính:
- IdTinTuc (int, PK): mã tin tức.
- TieuDe, NoiDung, HinhAnh, NgayTao
- Vai trò: Lưu trữ thông tin bài viết, tin tức, sự kiện của nhà hàng

public class TinTuc

{

[Key]

public int IdTinTuc { get; set; }

public string TieuDe { get; set; }

```

public string NoiDung { get; set; }
public string Hinhanh { get; set; }
public DateTime NgayTao { get; set; } = DateTime.Now;
}

```

4.2.5. LienHe

- Thuộc tính:
 - IdLienHe (int, PK): mã liên hệ
 - HoTen, Email, TinNhan, NgayTao.
- Vai trò: Lưu ý kiến phản hồi, góp ý của khách hàng.

```
public class LienHe
```

```

{
  [Key]
  public int IdLienHe { get; set; }
  public string HoTen { get; set; }
  public string Email { get; set; }
  public string TinNhan { get; set; }
  public DateTime NgayTao { get; set; } = DateTime.Now;
}
```

4.2.6. KhachHang

- Thuộc tính:
 - IdKH (int, PK): mã khách hàng.
 - HoTen, Email (duy nhất), MatKhau.
- Quan hệ: Có nhiều DatBan
- Vai trò: Quản lý thông tin khách hàng đã đăng ký trên hệ thống.

```
public class KhachHang
```

```

{
  [Key]
  public int IdKH { get; set; }
  public string HoTen { get; set; }
  public string Email { get; set; }
  public string MatKhau { get; set; }
}
```

4.2.7. AdminUser

- Thuộc tính:
 - IdAD (int, PK): mã admin
 - Username, Password, Role.
- Quan hệ:
 - 1 admin có thể có nhiều bản ghi AdminLogs.
 - 1 admin có thể có nhiều vai trò (Roles) thông qua AdminRoles.
- Vai trò: Quản lý, vận hành hệ thống, xử lý các nghiệp vụ back-end.

```
public class AdminUser
```

```

{
    [Key] // Bắt buộc phải có để EF nhận diện
    public int IdAD { get; set; }
    [Required]
    public string Username { get; set; }
    [Required]
    public string Password { get; set; } // dùng Password thay PasswordHash
    public string Role { get; set; }
    // Điều hướng đến bảng log
    public ICollection<AdminLogs> Logs { get; set; }

    // Điều hướng đến bảng vai trò
    public ICollection<AdminRoles> AdminRoles { get; set; }
}

```

4.2.8. AdminLogs

- Thuộc tính:
 - LogID (int, PK): mã log.
 - IdAD (FK): khóa ngoại tới admin.
 - HanhDong, ThoiGian.
- Quan hệ: Ghi nhận hoạt động của một admin
- Vai trò: Theo dõi, audit các hoạt động quản trị.

```

public class AdminLogs
{
    [Key]
    public int LogID { get; set; }
    public int IdAD { get; set; }
    public string HanhDong { get; set; }
    public DateTime ThoiGian { get; set; } = DateTime.Now;
    // Navigation
    public AdminUser AdminUser { get; set; }
}

```

4.2.9. AdminRoles

- Thuộc tính:
 - IdAD, RoleID (cặp PK).
- Quan hệ: Liên kết N - N giữa AdminUser và Roles
- Vai trò: Xác định vai trò cụ thể của mỗi admin.

```

public class AdminRoles
{
    [Key]
    public int IdAD { get; set; }
    public int RoleID { get; set; }
    // Navigation

```

```

public AdminUser AdminUser { get; set; }
public Roles Role { get; set; }
}

```

4.2.10. Roles

- Thuộc tính: RoleID (int, PK), RoleName.
- Vai trò: Danh sách quyền hạn quản lý: admin, editor, staff, ...

4.2.11. AdminLogin

- Thuộc tính: Username, Password.
- Không lưu trong cơ sở dữ liệu, dùng cho giao diện đăng nhập admin.

4.2.12. KhachHangLoginModel

- Thuộc tính: Email, Matkhau.
- Không lưu trong cơ sở dữ liệu, dùng cho giao diện đăng nhập khách hàng.

4.2.13. KhachHangRegisterModel

- Thuộc tính: HoTen, Email, MatKhau, ConfirmPassword.
- Không lưu trong cơ sở dữ liệu, dùng cho giao diện đăng ký của khách hàng.

Các mối quan hệ trong Model:

- LoaiMon: quan hệ một nhiều với MonAn.
- MonAn: N - 1 LoaiMon.
- KhachHang: quan hệ một nhiều với DatBan.
- AdminUser: 1-N AdminLogs, N-N Roles.
- AdminRoles: Trung gian giữa AdminUser và Roles.

4.3. Controller

Controller đóng vai trò xử lý các yêu cầu HTTP từ client.

- Nhận request từ giao diện (View).
- Gọi các Model để tương tác với dữ liệu.
- Trả lại response dưới dạng View hoặc Redirect.
- Sử dụng mô hình MVC của ASP.NET Core.

4.3.1. AdminController

- Chức năng: Quản lý toàn bộ nghiệp vụ của admin: đăng nhập, dashboard, quản lý món ăn, quản lý đặt bàn. Các Controller của admin đều kiểm tra seAdminId trước khi xử lý, nếu chưa login sẽ redirect về login.

Method/Action	URL	Metho d	Chức năng
Index()	/Admin	GET	Kiểm tra session và điều hướng dashboard
Login()	/Admin/Login	GET	Hiển thị form login
Login(model)	/Admin/Login	POST	Xác thực admin, lưu session

Logout()	/Admin/Logout	GET	Xóa session, đăng xuất
Dashboard()	/Admin/Dashboard	GET	Hiển thị trang dashboard
ManageMonAn()	/Admin/ManageMonAn	GET	Danh sách món ăn
CreateMonAn()	/Admin/CreateMonAn	GET	form thêm món
CreateMonAn(model)	/Admin/CrateMonAn	POST	Lưu món mới
EditMOnAn(id)	/Admin/EditMonAn	GET	Form sửa món
EditMOnAn(model)	/Admin/EditMOnAn	POST	cập nhật món
DeleteMonAn(id)	/Admin/DeleteMonAn	GET	Xóa món
ManageDatBan()	/Admin/MângéDatBan	GET	Danh sách đặt bàn
ConfirmDatBan(id)	/Admin/ConfirmDatBan	POST	Xác nhận đặt bàn
DeleteDatBan(id)	/Admin/DeleteDatBan	POST	Xóa đặt bàn

- Luồng xử lý chính:
 - Người dùng gửi request GET /Admin/Login, Controller trả về View trong form login.
 - Người dùng nhập thông tin, gửi POST /Admin/Login, Controller kiểm tra tài khoản và mật khẩu. Nếu đúng, lưu session và chuyển Dashboard, nếu sai, hiển thị lỗi và quay lại form.
 - Các action khác đều kiểm tra session trước khi xử lý.

```

public class AdminController : Controller
{
    private readonly AppDbContext _context;
    private readonly ILogger<AdminController> _logger;

    public AdminController(AppDbContext context, ILogger<AdminController>
        logger)
    {
        _context = context;
        _logger = logger;
    }

    // === Mặc định chuyển vào Dashboard nếu đã đăng nhập ===
    public IActionResult Index()
  
```

```

{
    if (!HttpContext.Session.Keys.Contains("AdminId"))
        return RedirectToAction("Login");

    return RedirectToAction("Dashboard");
}

// === Đăng nhập ===
//
// Hiển thị form đăng nhập (GET)
[HttpGet]
public IActionResult Login()
{
    return View();
}

[HttpPost]
public IActionResult Login(AdminLogin model)
{
    if (ModelState.IsValid)
    {
        Console.WriteLine("Nhập: " + model.Username + " / " + model.Password);

        //var admin = _context.AdminUsers.FirstOrDefault(x => x.Username ==
model.Username);
        var admin = _context.AdminUsers
.FirstOrDefault(x => x.Username.ToLower() == model.Username.ToLower());

        if (admin == null)
        {
            ModelState.AddModelError("", "X Không tìm thấy tên đăng nhập.");
        }
        //else if (admin.PasswordHash != model.Password)
        else if (admin.Password != model.Password)
        {
            ModelState.AddModelError("", "X Mật khẩu không đúng.");
        }
        else
        {
            // Đăng nhập thành công
            HttpContext.Session.SetInt32("AdminId", admin.IdAD);
            HttpContext.Session.SetString("AdminUsername", admin.Username);
        }
    }
}

```

```

        HttpContext.Session.SetString("AdminRole", admin.Role);
        return RedirectToAction("Dashboard");
    }

}

return View(model);
}

// === Đăng xuất ===
public IActionResult Logout()
{
    HttpContext.Session.Clear();
    return RedirectToAction("Login");
}

// === Trang chính sau khi đăng nhập ===
public IActionResult Dashboard()
{
    var adminId = HttpContext.Session.GetInt32("AdminId");
    if (adminId == null)
    {
        return RedirectToAction("Login");
    }

    return View(); // Có View tên Dashboard.cshtml?
}

=====

// === QUẢN LÝ MÓN ĂN ===

=====

public IActionResult ManageMonAn()
{
    if (!HttpContext.Session.Keys.Contains("AdminId"))
        return RedirectToAction("Login");

    var monanList = _context.MonAn.ToList();
    return View(monanList);
}

public IActionResult CreateMonAn()

```

```

    {
        if (!HttpContext.Session.Keys.Contains("AdminId"))
            return RedirectToAction("Login");
        ViewBag.LoaiMon = new SelectList(_context.LoaiMon, "IdLoai", "Name");
        return View();
    }

    [HttpPost]
    public IActionResult CreateMonAn(MonAn monAn)
    {
        if (!HttpContext.Session.Keys.Contains("AdminId"))
            return RedirectToAction("Login");

        if (ModelState.IsValid)
        {
            _context.MonAn.Add(monAn);
            _context.SaveChanges();
            return RedirectToAction("ManageMonAn");
        }
        return View(monAn);
    }

    public IActionResult EditMonAn(int id)
    {
        if (!HttpContext.Session.Keys.Contains("AdminId"))
            return RedirectToAction("Login");
        ViewBag.LoaiMon = new SelectList(_context.LoaiMon, "IdLoai", "Name");

        var item = _context.MonAn.Find(id);
        if (item == null)// return NotFound();
        {
            _logger.LogWarning("❌ Không tìm thấy món ăn với Id = {id}", id);
            return NotFound();
        }

        _logger.LogInformation("📝 Đang sửa món ăn Id = {id}", id);
        return View(item);
    }

    [HttpPost]

```

```

public IActionResult EditMonAn(MonAn monAn)
{
    if (!HttpContext.Session.Keys.Contains("AdminId"))
        return RedirectToAction("Login");

    if (ModelState.IsValid)
    {
        _context.MonAn.Update(monAn);
        var affected = _context.SaveChanges();
        _logger.LogInformation("✅ Đã cập nhật món ăn Id = {id}, Số dòng ảnh
hướng = {affected}", monAn.IdMonAn, affected);
        return RedirectToAction("ManageMonAn");
    }

    _logger.LogWarning("⚠️ ModelState không hợp lệ khi sửa món ăn Id = {id}",
monAn.IdMonAn);
    return View(monAn);
}

// [HttpPost]
public IActionResult DeleteMonAn(int id)
{
    Console.WriteLine($"🚀 Gọi DeleteMonAn với id = {id}");
    if (!HttpContext.Session.Keys.Contains("AdminId"))
        return RedirectToAction("Login");

    var item = _context.MonAn.Find(id);
    if (item != null)
    {
        _context.MonAn.Remove(item);
        _context.SaveChanges();
    }
    return RedirectToAction("ManageMonAn");
}
=====

// === QUẢN LÝ ĐẶT BÀN ===
// =====

```

public IActionResult ManageDatBan()

```

if (!HttpContext.Session.Keys.Contains("AdminId"))
    return RedirectToAction("Login");

var datbanList = _context.DatBan.ToList();
return View(datbanList);
}

[HttpPost]
public IActionResult ConfirmDatBan(int id)
{
    if (!HttpContext.Session.Keys.Contains("AdminId"))
        return RedirectToAction("Login");

    var item = _context.DatBan.Find(id);
    if (item != null)
    {
        item.DaXacNhan = true;
        item.TrangThai = "Đã xác nhận";
        var affected = _context.SaveChanges();
        _logger.LogInformation("✅ Đã xác nhận đặt bàn Id = {id}, Số dòng ảnh
hưởng = {affected}", id, affected);
    }
    else
    {
        _logger.LogWarning("🔴 Không tìm thấy đặt bàn Id = {id} để xác nhận", id);
    }

    return RedirectToAction("ManageDatBan");
}
[HttpPost]
public IActionResult DeleteDatBan(int id)
{
    if (!HttpContext.Session.Keys.Contains("AdminId"))
        return RedirectToAction("Login");

    var item = _context.DatBan.Find(id);
    if (item != null)
    {
        _context.DatBan.Remove(item);
        var affected = _context.SaveChanges();
        _logger.LogInformation("🗑️ Đã xóa đặt bàn Id = {id}, Số dòng ảnh hưởng =
{affected}", id, affected);
    }
}

```

```

    else
    {
        _logger.LogWarning("X Không tìm thấy đặt bàn Id = {id} để xóa", id);
    }
    return RedirectToAction("ManageDatBan");
}
public IActionResult Test()
{
    return Content("Trang quản trị hoạt động!");
}

```

4.3.2. DatBanController

- Chức năng: Xử lý đặt bàn của khách hàng.

Method/Action	URL	Method	Chức năng
Index()	/DatBan	GET	Hiển thị form và đơn của tôi
DatBan(model)	/DatBan/DatBan	POST	Đặt bàn mới

- Luồng xử lý:
 - GET /DatBan:
 - + Nếu đã đăng nhập -> hiển thị form và danh sách đơn của tôi.
 - + nếu chưa -> hiển thị form.
 - POST /DatBan/DatBan:
 - + Nếu chưa login -> yêu cầu login -> redirect về login.
 - + Nếu đã login -> nhập thông tin hợp lệ -> lưu đơn đặt bàn với trạng thái “Chờ xác nhận”, hiển thị thông báo thành công.

```

public class DatBanController : Controller
{
    private readonly ApplicationDbContext _context;

    public DatBanController(ApplicationDbContext context)
    {
        _context = context;
    }

    // Hiển thị form đặt bàn + đơn của tôi
    public IActionResult Index()
    {
        int? IdKH = HttpContext.Session.GetInt32("KhachHangId");
    }
}

```

```

if (IdKH != null)
{
    ViewBag.DonCuaToi = _context.DatBan
        .Where(d => d.IdKH == IdKH)
        .OrderByDescending(d => d.ThoiGian)
        .ToList();
}
else
{
    ViewBag.DonCuaToi = new List<DatBan>();
}
return View();
}

[HttpPost]
[ValidateAntiForgeryToken]
public IActionResult DatBan(DatBan datBan)
{
    int? IdKH = HttpContext.Session.GetInt32("KhachHangId");
    if (IdKH == null)
    {
        TempData["ReturnUrl"] = Url.Action("Index", "DatBan");
        TempData["ThongBao"] = "Bạn phải đăng nhập hoặc đăng ký trước khi đặt
bàn!";
        return RedirectToAction("Login", "KhachHang");
    }
    if (ModelState.IsValid)
    {
        datBan.TrangThai = "Chờ xác nhận";
        datBan.IdKH = IdKH.Value;
        _context.DatBan.Add(datBan);
        _context.SaveChanges();

        TempData["ThongBao"] = "✓ Đặt bàn thành công!";
        return RedirectToAction("Index");
    }
    // cập nhật lại danh sách đơn của tôi
    ViewBag.DonCuaToi = _context.DatBan
        .Where(d => d.IdKH == IdKH)
        .OrderByDescending(d => d.ThoiGian)
        .ToList();
}

```

```

        return View("Index");
    }

```

4.3.3. HomeController

- Chức năng: Trang chủ.

Method/Action	URL	Method	Chức năng
Index()	/	GET	Trang chủ

- Luồng xử lý:
 - GET / -> trả về View trang chủ.

```

public class HomeController : Controller
{
    public IActionResult Index() => View();
}

```

4.3.4. KhachHangController

- Chức năng: đăng ký, đăng nhập, đăng xuất khách hàng.

Method/Action	URL	Method	Chức năng
Register()	/KhachHang/Register	GET	Form đăng ký
Register(model)	/KhachHang/Register	POST	Lưu khách hàng mới
Login()	/KhachHang/Login	GET	Form đăng nhập
Login(model)	/KhachHang/Login	POST	Xác thực khách
Logout()	/KhachHang/Logout	GET	Đăng xuất

- Luồng xử lý:
 - GET / KhachHang/ Register -> hiển thị form đăng ký.
 - POST / KhachHang/ Register -> nếu hợp lệ, lưu thông tin khách hàng -> lưu session -> redirect về trang đặt bàn.
 - GET/ KhachHang/ Login -> hiển thị form đăng nhập (login).
 - POST / KhachHang/ Login:
 - + Nếu hợp lệ và đúng mật khẩu -> lưu session -> redirect về trang đặt bàn.
 - + Nếu sai -> hiển thị lỗi.
 - + GET/ KhachHang/ Logout -> xóa session -> quay lại trang đặt bàn.

```
public KhachHangController(AppDbContext context)
```

```

{
    _context = context;
}

[HttpPost]
public IActionResult Register(KhachHangRegisterModel model)
{
    if (ModelState.IsValid)
    {
        var kh = new KhachHang
        {
            HoTen = model.HoTen,
            Email = model.Email,
            MatKhau = model.MatKhau
        };

        _context.KhachHang.Add(kh);
        _context.SaveChanges();

        HttpContext.Session.SetInt32("KhachHangId", kh.IdKH);
        HttpContext.Session.SetString("KhachHangEmail", kh.Email);

        if (TempData["ReturnUrl"] != null)
        {
            string returnUrl = TempData["ReturnUrl"].ToString();
            return Redirect(returnUrl);
        }

        return RedirectToAction("Index", "DatBan");
    }

    return View(model);
}

[HttpPost]
public IActionResult Login(KhachHangLoginModel model)
{
    if (!ModelState.IsValid)
        return View(model);

    var kh = _context.KhachHang.FirstOrDefault(k => k.Email == model.Email);
    if (kh == null)

```

```

    {
        TempData["ThongBao"] = "Bạn chưa có tài khoản, hãy đăng ký!";
        return RedirectToAction("Register");
    }

    if (kh.MatKhau != model.MatKhau)
    {
        ModelState.AddModelError("", "Mật khẩu không đúng.");
        return View(model);
    }
    // lưu session
    HttpContext.Session.SetInt32("KhachHangId", kh.IdKH);
    HttpContext.Session.SetString("KhachHangEmail", kh.Email);

    // về trang ban đầu nếu có
    if (TempData["ReturnUrl"] != null)
    {
        string returnUrl = TempData["ReturnUrl"].ToString();
        return Redirect(returnUrl);
    }
    return RedirectToAction("Index", "DatBan");
}

public IActionResult Logout()
{
    HttpContext.Session.Clear(); // Xóa toàn bộ session của khách hàng
    TempData["ThongBao"] = "Bạn đã đăng xuất.";
    return RedirectToAction("Index", "DatBan"); // Chuyển về trang đặt bàn (hoặc
    Trang chủ nếu muốn)
}

[HttpGet]
public IActionResult Register()
{
    return View();
}

[HttpGet]
public IActionResult Login()
{
    return View();
}

```

}

4.3.5. LienHeController

- Chức năng: gửi liên hệ và email cảm ơn.

Method/Action	URL	Method	Chức năng
Index()	/LienHe	GET	Form liên hệ
Gui(model)	/LienHe/Gui	POST	Lưu liên hệ và gửi email
CamOn()	/LienHe/CamOn	GET	Trang cảm ơn

- Luồng xử lý:
 - GET/ LienHe -> hiển thị form liên hệ.
 - POST/ LienHe/ Gui -> nếu hợp lệ -> lưu tin nhắn -> gửi email cảm ơn - redirect sang trang cảm ơn.
 - GET/ LienHe/ CamOn -> hiển thị trang cảm ơn.

```
public LienHeController(AppDbContext context, EmailService emailService)
{
    _context = context;
    _emailService = emailService;
}

// Hiển thị form liên hệ
public IActionResult Index()
{
    return View();
}

// Xử lý gửi form và gửi email
[HttpPost]
public IActionResult Gui(LienHe model)
{
    if (ModelState.IsValid)
    {
        model.NgayTao = DateTime.Now;
        _context.LienHe.Add(model);
        _context.SaveChanges();

        if (!string.IsNullOrWhiteSpace(model.Email))
        {
            string subject = "Nhà hàng ABC - Đã nhận liên hệ";
            string body = $"@"
        }
    }
}
```

```

<p>Chào {model.HoTen},</p>
<p>Chúng tôi đã nhận được tin nhắn của bạn:</p>
<blockquote>{model.TinNhan}</blockquote>
<p>Chúng tôi sẽ phản hồi bạn trong thời gian sớm nhất.</p>
<p>Trân trọng,<br/>Nhà hàng ABC</p>";
}

// Gửi email
_emailService.SendEmailAsync(model.Email, subject, body);
}

return RedirectToAction("CamOn");
}

return View("Index", model);
}

// Trang cảm ơn
public IActionResult CamOn()
{
    return View();
}

```

4.3.6. MenuController

- Chức năng: hiển thị thực đơn.

Method/Action	URL	Method	Chức năng
Index(loaiId)	/Menu	GET	Hiển thị menu (lọc theo loại món nếu có)
DanhSachMonAn()	/Menu/DanhSachMonAn	GET	Hiển thị tất cả món ăn

- Luồng xử lý:
 - GET/ Menu -> lấy danh sách các món ăn (có thể lọc theo loại món) -> trả về View.
 - GET/ Menu/ DanhSachMonAn -> lấy toàn bộ danh sách món ăn -> trả về View.

```

public IActionResult Index(int? loaiId)
{
    var loaiMons = _context.LoaiMon.ToList();
    var MonAn = _context.MonAn
        .Where(m => loaiId == null || m.IdLoai == loaiId)
        .Include(m => m.LoaiMon)

```

```

    .ToList();

ViewBag.LoaiMons = loaiMons;
return View(MonAn);
}
public IActionResult DanhSachMonAn()
{
    var monAnList = _context.MonAn
        .Include(m => m.LoaiMon) // nếu bạn muốn lấy tên loại
        .OrderByDescending(m => m.IdMonAn)
        .ToList();

    return View(monAnList);
}

```

4.3.7. TinTucController

- Chức năng: hiển thị tin tức

Method/Action	URL	Method	Chức năng
Index()	/TinTuc	GET	Danh sách tin

- Luồng xử lý: GET/ TinTuc -> lấy danh sách tin tức (mới nhất lên trước) -> trả về View.

```

public TinTucController(AppDbContext context)
{
    _context = context;
}

// Hiển thị danh sách tin tức (mới nhất lên đầu)
public IActionResult Index()
{
    var tinTucs = _context.TinTuc
        .OrderByDescending(t => t.NgayTao)
        .ToList();

    return View(tinTucs);
}

```

4.4. View

4.4.1. Admin

Giao diện quản trị (các View của AdminController)

Sơ đồ luồng giao diện

Các View:

- Dashboard (bảng điều khiển): hiển thị các menu quản lý
 - Quản lý món ăn
 - Quản lý đặt bàn
- Thêm món ăn (CreateMonAn): form nhập
 - Tên món
 - Giá
 - Mô tả
 - Link ảnh (ở trong thư mục wwwroot/AnhnhaHnag/TênAnh.png/jpeg)
 - Loại món (select list từ View.bag.LoaiMon)
 - Submit form POST về /Admin/CreateMonAn
- Sửa món ăn (EditMonAn)
 - Tương tự form thêm, nhưng load sẵn dữ liệu món ăn và có IdMonAn (hidden).
 - Submit form POST về /Admin/EditMonAn.
- Đăng nhập quản trị (Login)
 - Nhập:
 - + Username
 - + Password
 - Validation lỗi trực tiếp trên form
 - Submit POST về /Admin/Login
- Quản lý món ăn (ManageMonAn): hiển thị bảng danh sách món ăn (Tên, giá, mô tả, ảnh)
 - Hành động:
 - + Sửa (link /Admin/EditMonAn)
 - + Xóa (form POST /Admin/DeleteMonAn)
- Quản lý đặt bàn (ManageDatBan): hiển thị bảng danh sách các đơn đặt bàn
 - Các cột: Họ tên, SDT, Email, Ngày/Giờ, Số người, Trạng thái, Đã xác nhận
 - Hành động:
 - + Xác nhận (form POST về /Admin/ConfirmDatBan)
 - + Xóa (form POST /Admin/DeleteMonAn)

4.4.2. DatBan

Trang đặt bàn và quản lý đặt bàn dành cho khách hàng

Các view:

- Đặt bàn:
 - Form nhập:
 - + Họ tên, SDT, Email

- + Thời gian (datetime-local)
- + Số người
- POST về /DatBan/ Datban
- Hiển thị danh sách đơn hàng của khách hiện tại (View.bag.Doncuatoi)
- Quản lý đặt bàn
- Có thể lặp: Bảng danh sách đặt bàn, giống trang quản trị nhưng hành động đơn giản:
 - + Xác nhận
 - + Xóa

4.4.3. KhachHang

Trang giao diện khách hàng đăng nhập hoặc đăng ký

Các view:

- Đăng nhập
- Form:
 - + Email
 - + Mật khẩu
- Submit POST /KhachHang/Login
- Hiển thị thông báo lỗi hoặc tạm thời từ ViewBag hoặc TempData
- Đăng ký
- Form:
 - + Họ tên
 - + Email
 - + Mật khẩu
 - + Xác nhận mật khẩu
- Validation lỗi từng trường
- Submit POST về /KhachHang/ Register

4.4.4. LienHe

Trang gửi liên hệ và trang cảm ơn

Các view:

- Liên hệ:
- Form:
 - + Họ tên
 - + Email
 - + Tin nhắn
- Submit POST về /LienHe/ Gui
- Cảm ơn
- Sau khi gửi liên hệ thành công.

4.4.5. Home

Trang chủ và các trang tĩnh

Các View:

- Trang chủ
- Giới thiệu nhà hàng
- Menu đặc sắc
- Các nút dẫn tới:
 - + Thực đơn (/Menu/ Index)
 - + Đặt bàn (/ DatBan/ Index)
 - + Tin tức (/ TinTuc/ Index)
- Privacy: trang chính sách bảo mật (mẫu)

4.4.6. Menu

Hiển thị danh sách món ăn trong nhà hàng. Cho phép lọc món ăn theo loại, hiển thị giá, hình ảnh, mô tả của từng món. Có liên kết để đặt bàn ngay

- Nội dung
- @model IEnumerable<GioiThieuNhaHang.Models.MonAn>; View nhận danh sách các món (MonAn) từ Controller
- ViewBag.LoaiMons : Danh sách các loại món ăn, dùng để lọc hoặc phân nhóm.
- Bộ lọc các loại món: Hiển thị các loại món dưới dạng link lọc

```
<ul>
  @foreach (var loai in ViewBag.LoaiMons)
  {
    <li><a href="/Menu?loaiId=@loai.IdLoai">@loai.Name</a></li>
  }
</ul>
```

- Danh sách món ăn: Nhóm món theo từng loại

```
@foreach (var loai in ViewBag.LoaiMons)
{
  var monTheoLoai = Model.Where(m => m.IdLoai == loai.IdLoai).ToList();
  if (monTheoLoai.Any())
  {
    <h3>@loai.Name</h3>
    <div class="menu-items">
      @foreach (var item in monTheoLoai)
      {
        <div class="menu-card">
          
          <h3>@Html.Raw(item.TenMon)</h3>
          <p>@item.MoTa</p>
        </div>
      }
    </div>
  }
}
```

```

        <div class="price">@item.Gia VNĐ</div>
    </div>
}
</div>
}
}

```

4.4.7. TinTuc

View quản trị nội dung quản lý tin tức, bài viết. Đăng nhập admin, hiển thị thông tin admin sau khi đăng nhập.

- Nội dung

- Session lưu tên và vai trò admin:

```

<h2>Xin chào @Context.Session.GetString("AdminUsername")</h2>
<p>Vai trò: @Context.Session.GetString("AdminRole")</p>
    - Nút đăng xuất:
<a href="@Url.Action("Logout", "Admin")" class="btn btn-danger">Đăng xuất</a>
    - Form đăng nhập:
<form asp-action="Login" method="post">
    <input asp-for="Username" />
    <input asp-for="Password" />
</form>
    • Sử dụng Model là AdminLogin chứa thông tin đăng nhập
    • Có xác thực ModelState và ValidationScripts.

```

4.4.8. Share/_Layout.cshtml

Layout chung cho toàn site, chứa phần Header, Navbar, Footer, @RenderBody(), Scripts.

- Navbar: nếu khách đã đăng nhập thì hiện tên và nút đăng xuất, ngược lại hiện nút đăng nhập hoặc đăng ký

```

<ul>
    <li><a asp-controller="Home">Trang chủ</a></li>
    <li><a asp-controller="Menu">Thực đơn</a></li>
    <li><a asp-controller="DatBan">Đặt bàn</a></li>
    <li><a asp-controller="TinTuc">Tin tức</a></li>
    <li><a asp-controller="LienHe">Liên hệ</a></li>
    <li><a asp-controller="Admin">Quản trị</a></li>
</ul>

```

- Footer: thông tin bản quyền

- Script:

- jQuery
- Bootstrap
- [site.js](#)

- validate

4.5. Các tệp cấu hình khác

4.5.1. *appsetting.json*

Đây là file cấu hình chính của ứng dụng [ASP.NET](#) Core, chứa các tham số quan trọng như:

- Chuỗi kết nối cơ sở dữ liệu
- Thông số môi trường
- Thiết lập gửi email
- Mức log
- Các host được phép

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=DESKTOP-EEPI9VF\SQLEXPRESS;Database=WebsiteGioiThieuNhaHang;Trusted_Connection=True;Encrypt=False"
  },
  "environmentVariables": {
    "ASPNETCORE_ENVIRONMENT": "Development"
  },
  "EmailSettings": {
    "SenderEmail": "your-email@gmail.com",
    "SenderName": "Nhà hàng ABC",
    "SenderPassword": "nhahangabc",
    "SmtpServer": "smtp.gmail.com",
    "Port": 587
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft.AspNetCore": "Warning"
    }
  },
  "AllowedHosts": "*"
}
```

- ConnectionString: Khai báo kết nối SQL đến database WebsiteGioiThieuNhaHang. Sử dụng xác thực Windows(Trusted_Connection=True), không mã hóa.

- environmentVariables: Thiết lập môi trường chạy của ứng dụng là Development để dễ debug.
- EmailSettings: Cấu hình để ứng dụng gửi mail thông qua SMTP Gmail. Bao gồm email người gửi, tên hiển thị, mật khẩu ứng dụng (cần đảm bảo bảo mật), máy chủ SMTP và port (587).
- Logging: Ghi log ở mức Information cho ứng dụng và Warning cho các component của Microsoft.AspNetCore.
- AllowedHosts: Cho phép tất cả các host (*) truy cập.

4.5.2. Program.cs

Đây là entry point của ứng dụng ASP.NET Core (.NET 6+), nơi cấu hình dịch vụ, middleware và pipeline.

Các phần chính:

- Cấu hình dịch vụ (builder.Services)
- Đăng ký DbContext: cho phép EF Core làm việc với SQL Server và kết nối cơ sở dữ liệu theo DefaultConnection.

builder.Services.AddDbContext

<AppDbContext>

(options =>

```
options.UseSqlServer(builder.Configuration.GetConnectionString("DefaultConnection"));
});
```

- Đăng ký EmailService: Cho phép inject EmailService đã cấu hình EmailSetting.

builder.Services.Configure

<EmailSettings>

(builder.Configuration.GetSection("EmailSettings"));

builder.Services.AddScoped<EmailService>();

- Đăng ký MVC và Session: Kích hoạt session, bộ nhớ đệm và hỗ trợ MVC (Controllers + Views).

builder.Services.AddControllersWithViews();

builder.Services.AddDistributedMemoryCache();

builder.Services.AddSession();

- Pipeline(app)

- Xử lý lỗi

if (!app.Environment.IsDevelopment())

{

 app.UseExceptionHandler("/Home/Error");

 app.UseHsts();

}

else

{

```
    app.UseDeveloperExceptionPage();
}
```

- Các middleware:

```
app.UseStaticFiles();      // Phục vụ file tĩnh (ảnh, CSS, JS...)
app.UseRouting();         // Xác định routing
app.UseSession();          // Cho phép lưu thông tin session
app.UseAuthorization();   // Xác thực người dùng (nếu có)
```

- Định tuyến:

```
app.UseEndpoints(endpoints =>
{
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller=Home}/{action=Index}/{id?}");
});

Hoặc

app.MapControllerRoute(
    name: "default",
    pattern: "{controller=Home}/{action=Index}/{id?}");
```

- Chạy ứng dụng:

```
app.Run();
```

Luồng khởi chạy theo trình độ sau:

- Đọc cấu hình từ appsettings.json
- Đăng ký các dịch vụ (DbContext, Email, Session, MVC...).
- Xây dựng và chạy ứng dụng với các middleware:
 - + Xử lý lỗi
 - + Phục vụ file tĩnh
 - + Xử lý session.
 - + Xác thực.
 - + Định tuyến đến các Controller/Action.
- Lắng nghe và phục vụ yêu cầu HTTP.

4.5.3. Phân tích mã nguồn AppDbContext.

Đây là lớp **DbContext**, là cầu nối giữa ứng dụng ASP.NET Core và cơ sở dữ liệu, được viết dựa trên Entity Framework Core (EF Core).

- Cấu trúc cơ bản:

- Kế thừa từ Microsoft.EntityFrameworkCore.DbContext.
- Sử dụng Dependency Injection để lấy DbContextOptions chứa thông tin kết nối & cấu hình.

```
public class AppDbContext : DbContext
{
    public AppDbContext(DbContextOptions<AppDbContext> options)
        : base(options)
    {
    }
}
```

- Các bảng trong database(DbSet): Mỗi database đại diện cho một bảng trong cơ sở dữ liệu.

```
public DbSet<LoaiMon> LoaiMon { get; set; }
public DbSet<MonAn> MonAn { get; set; }
public DbSet<DatBan> DatBan { get; set; }
public DbSet<TinTuc> TinTuc { get; set; }
public DbSet<LienHe> LienHe { get; set; }
public DbSet<AdminUser> AdminUsers { get; set; }
public DbSet<AdminLogs> AdminLog { get; set; }
public DbSet<Roles> Role { get; set; }
public DbSet<AdminRoles> AdminRole { get; set; }
public DbSet<KhachHang> KhachHang { get; set; }
```

- Cấu hình quan hệ và khóa trong OnModelCreating.

Phương thức OnModelCreating dùng để:

- Xác định khóa chính (primary key).
- Cấu hình các quan hệ giữa các bảng (1-n, n-n).
- Quan hệ AdminUser - AdminRoles - Roles: là quan hệ nhiều - nhiều
- Một admin có thể có nhiều role.
- Một role có thể được gán cho nhiều admin.
- Bảng trung gian: AdminUser
- Khóa chính tổng hợp (composite key) là (IdAD, RoleID)

```
modelBuilder.Entity<AdminRoles>()
    .HasKey(ar => new { ar.IdAD, ar.RoleID });
```

- Cấu hình quan hệ 1-n từ AdminUser -> AdminRoles và Roles -> AdminRoles

```
modelBuilder.Entity<AdminRoles>()
    .HasOne(ar => ar.AdminUser)
    .WithMany(u => u.AdminRoles)
    .HasForeignKey(ar => ar.IdAD);
```

```
modelBuilder.Entity<AdminRoles>()
```

```
.HasOne(ar => ar.Role)
.WithMany(r => r.AdminRoles)
.HasForeignKey(ar => ar.RoleID);
```

- Quan hệ AdminUser - AdminLogs: Một admin có thể có nhiều bản ghi log.
Mỗi log gắn với 1 admin AdminUser có tập hợp Logs

```
modelBuilder.Entity<AdminLogs>()
    .HasKey(al => al.LogID);
```

```
modelBuilder.Entity<AdminLogs>()
    .HasOne(log => log.AdminUser)
    .WithMany(user => user.Logs)
    .HasForeignKey(log => log.IdAD);
```

- Quan hệ MonAn - LoaiMon: Một loại món có nhiều món ăn. MonAn gắn với LoaiMon thông qua IdLoai

```
modelBuilder.Entity<MonAn>()
    .HasOne(m => m.LoaiMon)
    .WithMany(l => l.MonAns)
    .HasForeignKey(m => m.IdLoai);
```

Vai trò của DbContext:

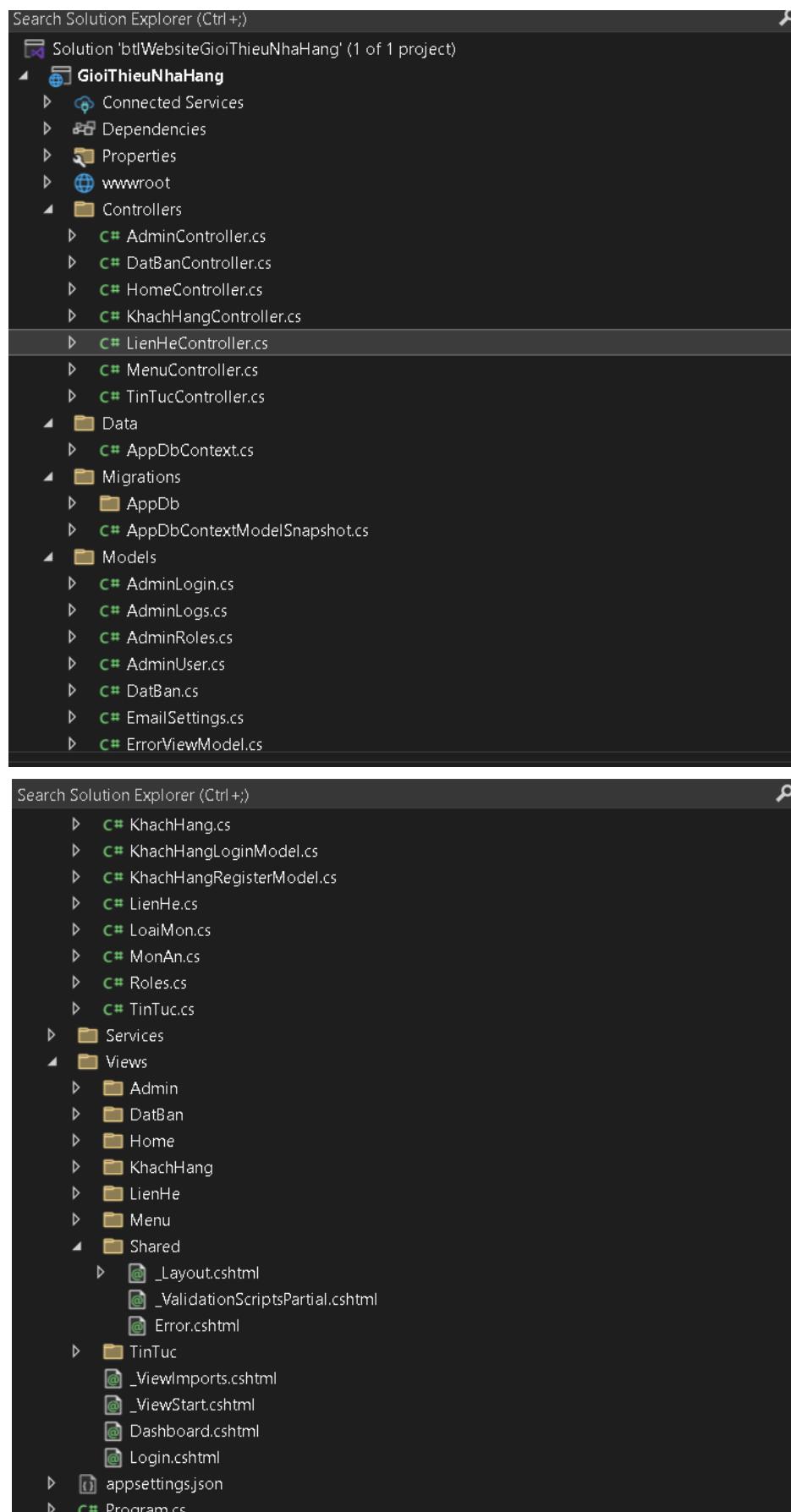
- Khởi tạo DbContext để EF Core biết các bảng và quan hệ trong CSDL.
- Cấu hình chi tiết các quan hệ, khóa chính và khóa ngoại để đảm bảo dữ liệu nhất quán.
- Là điểm giao tiếp duy nhất giữa code và CSDL trong ứng dụng.

4.5. Vị trí của các thư mục và link Github.

4.5.1. Link GitHub

Link GitHub: <https://github.com/my1710/WebsiteGioiThieuNhaHangTest.git>

4.5.2. Vị trí của các thư mục



Hình 22. Vị trí các thành phần back-end

4.7. Đề xuất hướng phát triển

Để nâng cao chất lượng và tính ứng dụng của website trong thực tiễn, nhóm chúng em đề xuất một số hướng phát triển tiếp theo như sau:

- Tích hợp thanh toán trực tuyến: Cho phép khách hàng đặt bàn kèm thanh toán trước qua các cổng thanh toán như Momo, ZaloPay, Visa/MasterCard...
- Hỗ trợ đa ngôn ngữ: Thêm tính năng chuyển đổi ngôn ngữ (ví dụ: Tiếng Việt - Tiếng Anh) để phục vụ khách hàng quốc tế.
- Ứng dụng công nghệ AI: Gợi ý món ăn theo sở thích, lịch sử đặt bàn, hoặc xu hướng của người dùng.
- Tích hợp chatbot: Hỗ trợ khách hàng tự động trả lời các câu hỏi thường gặp, đặt bàn nhanh chóng.
- Tối ưu SEO và quảng bá: Để website dễ dàng xuất hiện trên các công cụ tìm kiếm, thu hút thêm nhiều khách hàng.
- Xây dựng ứng dụng di động (Mobile app): Tăng khả năng tiếp cận và thuận tiện hơn cho người dùng.

KẾT LUẬN

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, việc xây dựng một website giới thiệu nhà hàng không chỉ là một nhu cầu tất yếu để quảng bá thương hiệu, mà còn là một giải pháp hiệu quả nhằm nâng cao trải nghiệm khách hàng, tối ưu hóa quy trình phục vụ và quản lý dữ liệu.

Thông qua quá trình phân tích và hiện thực hóa đề tài, em đã phát triển được một hệ thống website với những đặc điểm nổi bật cả về mặt kỹ thuật lẫn trải nghiệm người dùng, cụ thể:

Về giao diện người dùng: Website được thiết kế với phong cách hiện đại, thân thiện, dễ sử dụng, đồng thời tối ưu cho nhiều thiết bị (responsive). Các thành phần giao diện (thực đơn, đặt bàn, tin tức, liên hệ...) được tổ chức khoa học, rõ ràng và trực quan, giúp khách hàng dễ dàng tìm kiếm thông tin và tương tác với nhà hàng.

Về kiến trúc phần mềm: Ứng dụng được xây dựng theo mô hình **MVC (Model - View - Controller)** của ASP.NET Core, giúp tách bạch rõ ràng giữa xử lý dữ liệu, giao diện và điều phối, dễ bảo trì và mở rộng.

Về cơ sở dữ liệu: Hệ thống sử dụng SQL Server để lưu trữ và quản lý dữ liệu, với các bảng được thiết kế chặt chẽ, phản ánh đúng các thực thể kinh doanh của nhà hàng (món ăn, loại món, khách hàng, đặt bàn, admin, vai trò...). Các quan hệ giữa các bảng được định nghĩa rõ ràng, đảm bảo tính toàn vẹn dữ liệu (qua khóa chính, khóa ngoại, quan hệ 1-nhiều, nhiều-nhiều).

Về bảo mật và quản trị: Hệ thống hỗ trợ quản lý tài khoản admin, phân quyền theo vai trò, lưu lại nhật ký hoạt động của admin, đồng thời đảm bảo an toàn dữ liệu người dùng nhờ cơ chế đăng nhập, logout và session.

Về mở rộng: Website đã chuẩn bị sẵn sàng cho việc tích hợp các dịch vụ bổ sung như email marketing, thông báo tự động qua email (với cấu hình SMTP), và có thể mở rộng thêm các tính năng thương mại điện tử hoặc đặt bàn trực tuyến có thanh toán.

Em xin chân thành cảm ơn quý thầy/cô trong khoa Công nghệ Thông tin đã tận tình giảng dạy, cung cấp những kiến thức nền tảng cũng như những chỉ dẫn quý báu trong suốt quá trình học tập và thực hiện đề tài.

Đặc biệt, em xin gửi lời cảm ơn sâu sắc đến giảng viên hướng dẫn, người đã tận tâm góp ý, định hướng, giúp em hoàn thiện đề tài này một cách tốt nhất.

Em cũng xin cảm ơn các bạn đồng hành và những người đã hỗ trợ, đóng góp ý kiến giúp em hoàn thiện website.

Mặc dù đã nỗ lực hết sức, nhưng chắc chắn không thể tránh khỏi những thiếu sót. Em rất mong nhận được sự góp ý, phê bình từ quý thầy/cô để đề tài được hoàn thiện hơn.

Xin cảm ơn.

TÀI LIỆU THAM KHẢO

[1] Nguyễn Văn Hiếu, *Lập trình Web với ASP.NET MVC 5*, Nhà xuất bản Lao động, 2017.

[2] Trần Đình Chiến, *Cơ sở dữ liệu và ứng dụng trong quản lý*, NXB Bách Khoa Hà Nội, 2019.

[3] Nguyễn Thanh Tùng, *Phân tích & thiết kế hệ thống thông tin*, NXB Giáo Dục, 2020.

[4] GitHub: Tham khảo các dự án open-source về website nhà hàng.