TRƯỜNG ĐẠI HỌC NAM CẦN THƠ

**KHOA CÔNG NGHỆ THÔNG TIN**

🙠🕮🙢



**BÁO CÁO ĐỒ ÁN CƠ SỞ 1**

**TÊN ĐỀ TÀI**

**Quản lý và kinh doanh**

**quán ăn**

**Ngành Công Nghệ Thông Tin**

**Mã số ngành 748201**

**GVHD: Võ Văn Phúc Sinh viên thực hiện:**

**GVPB:Ngô Hồ Anh Khôi Lê Trần Hoài Bảo - 1910088**

**Văn Hữu Trọng - 192032**

**Cần Thơ, Ngày 03 Tháng 05 Năm 2022**

**CHẤP THUẬN CỦA HỘI ĐỒNG**

Đề tài “**Web quản lý ký túc xá sinh viên đại học”**, do sinh viên Nguyễn Trí Nguyên, Nguyễn Ngọc Sang thực hiện dưới sự hướng dẫn của **GV:Võ Văn Phúc**. Đề tài đã báo cáo và được Hội đồng chấm đồ án thông qua ngày .....................................

|  |  |
| --- | --- |
| **Ủy viên** | **Thư ký** |
|  |  |
| --------------------------------------- | --------------------------------------- |
| **GHI CHỨC DANH, HỌ, TÊN** | **GHI CHỨC DANH, HỌ, TÊN** |

|  |  |
| --- | --- |
| **Phản biện 1** | **Phản biện 2** |
|  |  |
| --------------------------------------- | --------------------------------------- |
| **GHI CHỨC DANH, HỌ, TÊN** | **GHI CHỨC DANH, HỌ, TÊN** |

|  |  |
| --- | --- |
| **Cán bộ hướng dẫn** | **Chủ tịch Hội đồng** |
|  |  |
| --------------------------------------- | --------------------------------------- |
| **GHI CHỨC DANH, HỌ, TÊN** | **GHI CHỨC DANH, HỌ, TÊN** |

# **LỜI CẢM ƠN**

Với lòng biết ơn sâu sắc và tình cảm chân thành cho phép chúng em gửi lời cảm ơn chân thành nhất tới:

Trường Đại học Nam Cần Thơ, khoa kỹ thuật- công nghệ, cùng các giảng viên đã tận tình chỉ dạy và tạo điều kiện giúp đỡ em trong quá trình học tập, nghiên cứu và hoàn thành đề tài.

Đặc biệt chúng em xin bày tỏ lòng biết ơn sâu sắc đến Giảng viên Võ Văn Phúc người hướng dẫn và cũng là người đã luôn tận tình hướng dẫn, chỉ bảo, giúp đỡ và động viên chúng em trong suốt quá trình nghiên cứu và hoàn thành đề tài nghiên cứu này.

Mặc dù, đã cố gắng rất nhiều nhưng đây là đồ án đầu tiên nên không tránh khỏi những thiếu sót; chúng em rất mong nhận được sự thông cảm, chỉ dẫn, giúp đỡ và đóng góp ý kiến của quý thầy cô, các cán bộ quản lí và các bạn đang học cùng chúng em tại Trường Đại học Nam Cần Thơ .

**Em xin chân thành cảm ơn!**

**Cần Thơ, ngày… tháng … năm 202…**

**Người thực hiện**

**Sinh viên**

**-------------------------------**

**Nguyễn Trí Nguyên**

# **LỜI CAM KẾT**

Tôi xin cam kết nội dung Đồ án này được hoàn thành dựa trên kết quả nghiên cứu của tôi và kết quả nghiên cứu này chưa từng được dùng cho bất cứ đồ án cùng cấp nào khác.

Cần Thơ, ngày … tháng … năm ….

**Sinh viện thực hiện**

**Nguyễn Trí Nguyên**

# **NHÂN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

*Cần Thơ, ngày …. tháng …..năm 202…*

**Giảng viên hướng dẫn**

**Võ Văn Phúc**

# **NHÂN XÉT CỦA GIẢNG VIÊN PHẢN BIỆN**

*Cần Thơ, ngày …. tháng …..năm 202…*

**Giảng viên phản biện**

**Ngô Hồ Anh Khôi**

# **MỤC LỤC**

# **CHƯƠNG I: GIỚI THIỆU**

## 1.1: LÝ DO CHỌN ĐỀ TÀI:

Internet hiện đang là công cụ liên kết xã hội lớn nhất thế giới và đang ngày càng trở nên phổ biến và đóng một vai trò quan trọng trong đời sống con người. Internet đem lại lợi ích vô cùng to lớn cho con người hơn bất kì nền công nghiệp truyền thống nào khác cũng như nguồn lợi khổng lồ mà có thể mang lại cho các nhà đầu từ.

Ngày nay, với sự bùng nổ của ngành công nghệ thông tin, Internet lại càng có ý nghĩa quan trọng hơn đối với doanh nghiệp sản xuất kinh doanh, dịch vụ. Với công cụ Internet, các doanh nghiệp có thể quảng cáo, giới thiệu sản phẩm, dịch vụ của mình phổ biến hơn với chi phí thấp hơn và phổ biến hơn công tác tiếp thị, quảng cáo, giới thiệu sản phẩm, dịch vụ của mình đến với người tiêu dùng để đạt được lợi thế cạnh tranh cao trên thị trường.

## Qua đó, chúng em đã nghiên cứu và phát triển đề tài “Website quản lý và kinh doanh quán ăn” với mục đích làm hoàn thiện hơn các công tác quảng cáo, tiếp thị sản phẩm, dịch vụ của cửa hàng tới người dùng. Sau thời gian tìm hiểu chúng em cảm thấy đồ án cơ sở 1 của chúng em còn nhiều thiếu sót và hạn chế. Em rất mong nhận được sự thông cảm và góp ý từ quý Thầy Cô để đồ án ngày càng hoàn thiện hơn. Em xin chân thành cảm ơn.

Sau thời gian tìm hiểu chúng em cảm thấy đồ án cơ sở 1 của chúng em còn nhiều thiếu sót và hạn chế. Em rất mong nhận được sự thông cảm và góp ý từ quý Thầy Cô để đồ án ngày càng hoàn thiện hơn. Em xin chân thành cảm ơn.

## 1.2: SƠ LƯỢC VỀ CÔNG CỤ HỖ TRỢ PHÁT TRIỂN PHẦM MỀM:

**Visual Studio Code** (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

# **CHƯƠNG II: CƠ SỞ LÝ THUYẾT**

## 2.1: NỘI DUNG NGHIÊN CỨU:

Nội dung chính là nghiên cứu về phát triển một website Quản lý ký túc xá đại học… trên nền tảng Visual Studio Code. Sử dụng **Laravel là một PHP framework, có mã nguồn mở và miễn phí, được xây dựng nhằm hỗ trợ phát triển các phần mềm, ứng dụng, theo kiến trúc MVC**.

Mô hình MVC (MVC Pattern): Mô hình kiến trúc Model-View-Controller (MVC) chia ứng dụng ra làm ba nhóm thành phần chính: Models, Views và Controllers. Mô hình này giúp đạt được sự tách biệt của các mốt quan tâm. Sử dụng mô hình này, bạn có thể hình dung, khi  một yêu cầu từ máy client gửi tới server, Controller sẽ thực hiện nhiệm vụ của mình đó là tiếp nhận và xử lý yêu cầu. Trong trường hợp cần thiết, nó có thể liên hệ Model – bộ phận làm việc với database để hỗ trợ.

* Model (M): Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là một cầu nối giữa 2 thành phần bên dưới là View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một [file XML](https://monamedia.co/file-xml-la-gi/) bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu,…
* View (V): Đây là phần giao diện (theme) dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. Thông thường, các ứng dụng web sử dụng MVC View như một phần của  hệ thống, nơi các thành phần [HTML](https://monamedia.co/html-va-css-la-gi/) được tạo ra. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của  người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực tiếp với Controller, cũng không được lấy dữ liệu từ Controller mà chỉ hiển thị yêu cầu chuyển cho Controller mà thôi.
* Controller (C):Bộ phận có nhiệm vụ xử lý các yêu cầu người dùng đưa đến thông qua view. Từ đó, C đưa ra dữ liệu phù hợp với người dùng. Bên cạnh đó, Controller còn có chức năng kết nối với model.

## 2.2: CÁC CÔNG NGHỆ ÁP DỤNG VÀO ĐỀ TÀI:

Để phát triển và hoàn thiện đề tài, bằng cách tiếp cận công nghệ mới hiện nay. Nhóm em đã áp dụng các công nghệ như sau vào đề tài:**Net.core**, thư viện Jquery để viết code Javascript nhanh hơn.

**ASP.NET CORE:** Là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux. Bạn có thể tạo ra các ứng dụng web có thể testing theo mô hình MVC(Model-View-Controller).Bạn có thể xây dựng HTTP services cái mà hỗ trợ nhiều định dạng và có đầy đủ hỗ trợ cho nội dung của dữ liệu trả về. Razor cung cấp một ngôn ngữ hiệu quả để tạo Views.Tag Helper cho phép code server side tham gia vào việc tạo và render phần tử HTML.Model Binding tự động ánh xạ dữ liệu từ HTTP request tới tham số của method action.Model Validation tự động thực hiện validate client và server. ASP.NET Core được thiết kế để tích hợp một cách liên tục với nhiều client-side frameworks, bao gồm AngularJS, KnockoutJS và Bootstrap. Trong đồ án cơ sở 1 này chúng em sử dụng ASP.NET CORE xuyên suốt từ đầu đến cuối của chương trình.

**AJAX:** AJAX là chữ viết tắt của cụm từ Asynchronous Javascript and XML. AJAX là phương thức trao đổi dữ liệu với máy chủ và cập nhật một hay nhiều phần của trang web, hoàn toàn không reload lại toàn bộ trang. Ajax được viết bằng Javascript chạy trên client, tức là mỗi browser sẽ chạy độc lập hoàn toàn không ảnh hưởng lẫn nhau. Về mặt kỹ thuật, nó đề cập đến việc sử dụng các đối tượng XmlHttpRequest để tương tác với một máy chủ web thông qua Javascript. AJAX là một trong những công cụ giúp chúng ta đem lại cho người dùng trải nghiệm tốt hơn. Khi cần một thay đổi nhỏ thì sẽ không cần load lại cả trang web, làm trang web phải tải lại nhiều thứ không cần thiết. AJAX được sử dụng để thực hiện một callback. Được dùng để thực hiện việc truy xuất dữ liệu hoặc lưu trữ dữ liệu mà không cần phải reload lại toàn bộ trang web. Với những server nhỏ thì việc này cũng tiết kiệm được băng thông cho chúng ta hơn. Cần gì thì chỉ gửi dữ liệu phần đó, load lại 1 phần nhỏ để cập nhật thông tin chứ không load cả trang. Bằng cách này thì có thể giảm thiểu được tốc độ tải trang giúp người dùng có trải nghiệm tốt hơn.Trang web bạn tạo ra cũng sẽ đa dạng và động hơn.

**JQUERY:** jQuery là một thư viện JavaScript đa tính năng, nhỏ gọn, nhanh, được tạo bởi John Resig vào năm 2006 với một phương châm hết sức ý nghĩa: **Write less, do more - Viết ít hơn, làm nhiều hơn**. Nó đã giúp lập trình viên tiết kiệm được rất nhiều thời gian và công sức trong việc thiết kế website. jQuery cung cấp các phương thức xử lý sự kiện, hiệu ứng, tương tác chỉ với những dòng lệnh đơn giản. Các module chính mà của jQuery sử bao gồm: Ajax – xử lý Ajax, Atributes – xử lý thuộc tính của đối tượng HTML, Effect – xử lý hiệu ứng, Event – xử lý sự kiện, Form – xử lý form, DOM – xử lý Data Object Model, Selector – xử lý luồng lách các đối tượng HTML, Hiệu ứng và hoạt hình. Jquery rất nhanh và có thể mở rộng, Jquery cải thiện được hiệu suất của một chương trình ứng dụng. Giúp các ứng dụng web tương thích với các trình duyệt khác nhau. Mỗi trình duyệt khác nhau có thể quy định cách viết mã lệnh Javascript để thực hiện cùng một công việc theo các cách khác nhau. Ví dụ một số trình duyệt không hỗ trợ sự kiện DOMContentLoaded trong phương thức addEventListener() mà lại hỗ trợ sự kiện load hoặc onreadystatechange hoặc onload. Do đó dẫn tới việc lập trình viên Javascript thường phải rất vất vả khi xử lý sự khác biệt này. Sử dụng thư viện jQuery cho phép bạn có thể viết mã lệnh Javascript chạy được trên hầu hết các loại trình duyệt khác nhau với cùng một code.

**CSS:** **CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu ([HTML](https://topdev.vn/blog/html-la-gi/)). Nói ngắn gọn hơn là ngôn ngữ tạo phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,…thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

CSS được phát triển bởi **W3C** ([World Wide Web Consortium](https://www.w3.org/)) vào năm 1996, vì HTML không được thiết kế để gắn tag để giúp định dạng trang web.

Phương thức hoạt động của CSS là nó sẽ tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ markup (nền tảng của site) và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời.

**HTML:** **HTML** là chữ viết tắt của **Hypertext Markup Language**. Nó giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes, vâng vâng. HTML không phải là ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng “động” được. Nó chỉ giống như Microsoft Word, dùng để bố cục và định dạng trang web. Khi làm việc với HTML, chúng ta sẽ sử dụng cấu trúc code đơn giản (tags và attributes) để đánh dấu lên trang web. Ví dụ, chúng ta có thể tạo một đoạn văn bằng cách đặt văn bản vào trong cặp tag mở và đóng văn bản ***<p>*** và ***</p>*.**

**JAVASCRIPT:** JavaScript là ngôn ngữ phức tạp hơn HTML hay CSS. Và nó không được phát hành ở dạng beta cho đến năm 1995. Ngày nay, JavaScript được hỗ trợ bởi tất cả các trình duyệt web hiện đại. Và được sử dụng trên hầu hết mọi trang web trên web để có chức năng mạnh mẽ và phức tạp hơn.

JavaScript là ngôn ngữ lập trình dựa trên logic. Nó có thể được sử dụng để sửa đổi nội dung trang web. Và khiến nó hoạt động theo nhiều cách khác nhau để đáp ứng với hành động của người dùng. Các cách sử dụng phổ biến cho JavaScript bao gồm các hộp xác nhận, kêu gọi hành động và thêm các danh tính mới vào thông tin hiện có.

JavaScript là ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế các trang web tương tác. Hầu hết các hành vi động mà bạn sẽ thấy trên một trang web là nhờ JavaScript. Nó giúp tăng cường các hành vi và kiểm soát mặc định của trình duyệt. Ngôn ngữ lập trình này có thể làm rất nhiều thứ trên trang web của bạn. Nó làm cho mọi thứ trở nên tiện nghi hơn bao giờ hết. Tuy nhiên để học được ngôn ngữ này bạn cần phải rất kiên trì. Bởi đây là ngôn ngữ khá phức tạp và khó để học.

Với những chia sẻ về vai trò của lập trình HTML, CSS và JavaScript trong việc xây dựng website. Mong rằng các bạn sẽ có những cái nhìn tổng quan hơn về ngành lập trình này. Chúc bạn thành công. Nếu yêu thích, bạn hãy bắt tay ngay vào tìm hiểu sâu hơn về HTML, CSS và JavaScript.

**CÁC GIAO THỨC API:** API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng. Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức nằng login thông Google, Facebook, Twitter, Github… Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API. API trên hệ điều hành: Windows hay Linux có rất nhiều API, họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành. API của thư viện phần mềm hay framework: API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau và nó cũng giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng thư viện được viết bằng ngôn ngữ khác. Ví dụ bạn có thể dùng Php để yêu cầu một thư viện tạo file PDF được viết bằng C++. API hiện nay đều tuân thủ theo tiêu chuẩn REST và HTTP, tạo sự thân thiện dễ sử dụng với nhà phát triển. Giúp người dùng dễ dàng truy cập, dễ hiểu hơn. Web API hiện đại dùng cho các đối tượng cụ thể, chẳng hạn như mobile developer với document, version khác nhau. API key: Đây là loại code (string) được truyền tải bởi các chương trình máy tính gọi là API để xác định chương trình, nhà phát triển hoặc người dùng nó tới trang web. Các API key được sử dụng với mục đích nhằm giới hạn, kiểm soát sử dụng API. Chẳng hạn như ngăn chặn sự việc lạm

# **CHƯƠNG III: NỘI DUNG CHÍNH ĐỀ TÀI ĐỒ ÁN CƠ SỞ 1**

## 3.1: PHẦN GIỚI THIỆU:

### **3.1.1: MÔ TẢ BÀI TOÁN CỦA ĐỀ TÀI:**

Hiện nay việc phát triển của internet giúp con người thực hiện mọi công việc 1 cách hiệu quả hơn, nhờ đó hình thức đặt đồ ăn online đã ra đời. Do thời buổi dịch bệnh, nhiều người ngại ra đường nên quán sẽ có thêm hình thức đặt đồ ăn online và khách hàng cũng có thể đặt bàn trước cho mình bằng hình thức đặt online. Quán sẽ quản lý các đơn hàng, các đơn đặt hàng, quản lý luôn cả các nhân viên của quán

**3.2: PHẦN PHÂN TÍCH CÀI ĐẶT:**

### **3.2.1: QUAN NIỆM DỮ LIỆU:**

1. **Các loại thực thể:**
2. **Thực thể Tài Khoản**

Mô tả thông tin Tài khoản. Mỗi Tài khoản sẽ có một id tài khoản(id), tên đăng nhập (user), mật khẩu (password),sait(passwordsail) , tên (FullName), tài khoản là admin(IsAdmin),

1. **Thực thể Nhân Viên**

Mô tả thông tin nhân viên. Mỗi nhân viên sẽ có id sinh viên (id), Họ và tên(HoTen),Địa chỉ(Diachi),Số điện thoại(SDT),Lương(Luong),Ngày vào làm(NgayVaoLam),Làm ở khu vực nào(IdKVLamViec),Chức vụ là gì(IdChucVu),thứ tự hiển thị(DisplayOrder)

1. **Thực thể KVLamViec**

Mô tả thông tin Khu vực làm việc. Mỗi khu vực là việc sẽ có mã riêng(id),

Tên Khu vực làm việc(KVucLamViec),thứ tự hiển thị(DisplayOrder)

1. **Thực thể ChucVu**

Mô tả thông tin Chức vụ. Mỗi chức vụ là việc sẽ có mã riêng(id), Tên chức vụ (ChucVuNV),thứ tự hiển thị(DisplayOrder)

1. **Thực thể KhoQuan**

Mô tả thông tin Kho quán. Mỗi Món hàng trong kho sẽ có mã riêng(id), Tên món hàng(Thucpham),Số lượng(Soluong),Ngày nhập(ngaynhap), thứ tự hiển thị(DisplayOrder),Mã hàng (IdMaHang),tài khoản thêm món hàng(IdTaiKhoan)

1. **Thực thể MaHang**

Mô tả Mã Hàng . Mỗi mã hàng có một id,Tên mã hàng(LoaiHang), thứ tự hiển thị(DisplayOrder)

1. **Thực thể TrangThaiKho**

Mô tả thực thể Trạng thái kho. Mỗi trạng thái có ID duy nhất (id) , tên trạng thái(TrangThai), thứ tự hiển thị(DisplayOrder),

1. **Thực thể Menu**

Mô tả thực thể Menu. Mỗi món có ID duy nhất (id), tên món ăn(ThucDon),mô tả món(Mota),Giá(GiaTien),đường dẫn lưu tên ảnh của món ăn(duongdan),Số lượng món đã bán(SLDaBan),Thuộc danh mục nào(IdDanhMuc),Thuộc thể loại nào(IdTheLoai), thứ tự hiển thị(DisplayOrder).

1. **Thực thể TheLoaiMenu**

Mô tả thực thể Thể loại meniu. Mỗi thể loại có ID duy nhất (id), tên thể loại(TheLoai), thứ tự hiển thị(DisplayOrder).

1. **Thực thể DanhMucMenu**

Mô tả thực thể Danh mục Menu. Mỗi trạng thái có ID duy nhất (id), tên danh mục (DanhMuc), thứ tự hiển thị(DisplayOrder).

1. **Thực thể DatBan**

Mô tả thực thể Đặt bàn. Mỗi đơn đặt bàn có ID duy nhất (id),Tên Khách Hàng(TenKH), Số điện thoại(SDT),Khu Vực(KhuVuc), Ngày Đặt (Ngay),Giờ Đặt(Gio),Đã được xác nhận hay chưa (Duyet), thứ tự hiển thị(DisplayOrder).

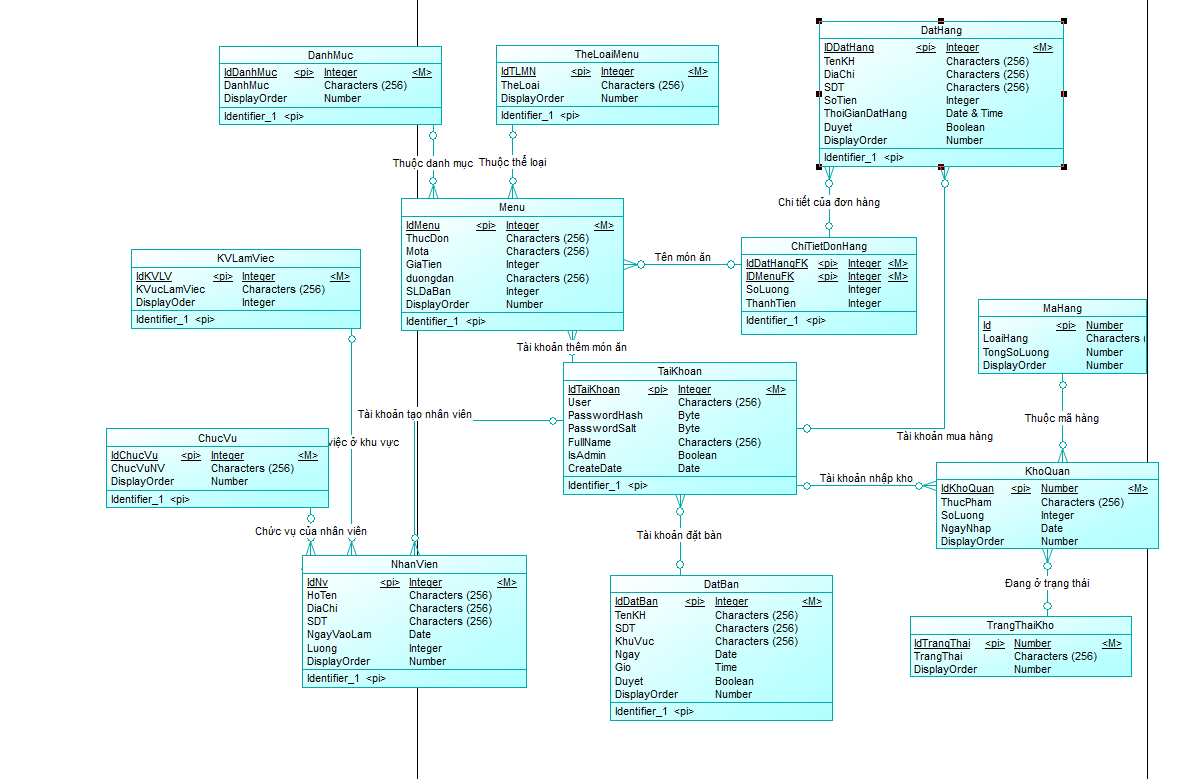
1. **Thực thể DatHang**

Mô tả thực thể Đặt hàng. Mỗi trạng thái có ID duy nhất (id), ),Tên Khách Hàng(TenKH), Số điện thoại(SDT),Số tiền(SoTien),Thời gian đặt hàng (ThoiGianDatHang),Tài khoản đặt hàng( IdTaiKhoan) ,Đã được xác nhận hay chưa (Duyet), thứ tự hiển thị(DisplayOrder).

1. **Thực thể ChiTietDonHang**

Mô tả thực thể Chi tiết đơn hàng. Chi tiết của đơn đặt hàng nào (IdDatHang), Tên món ăn(IdMenu),Số lượng(SoLuong), Thành tiền (ThanhTien).

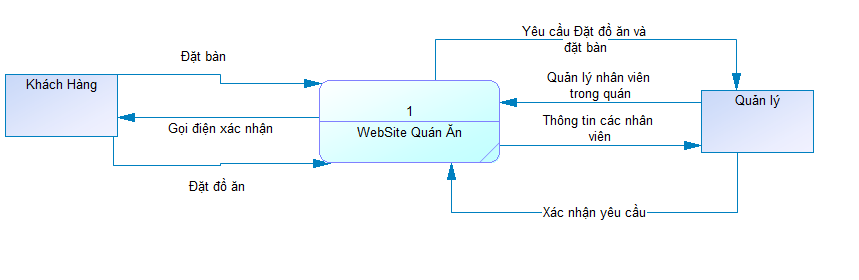
1. Mô hình dữ liệu quan hệ (ERD):



Hình 1: Mô hình dữ liệu quan hệ (ERD)

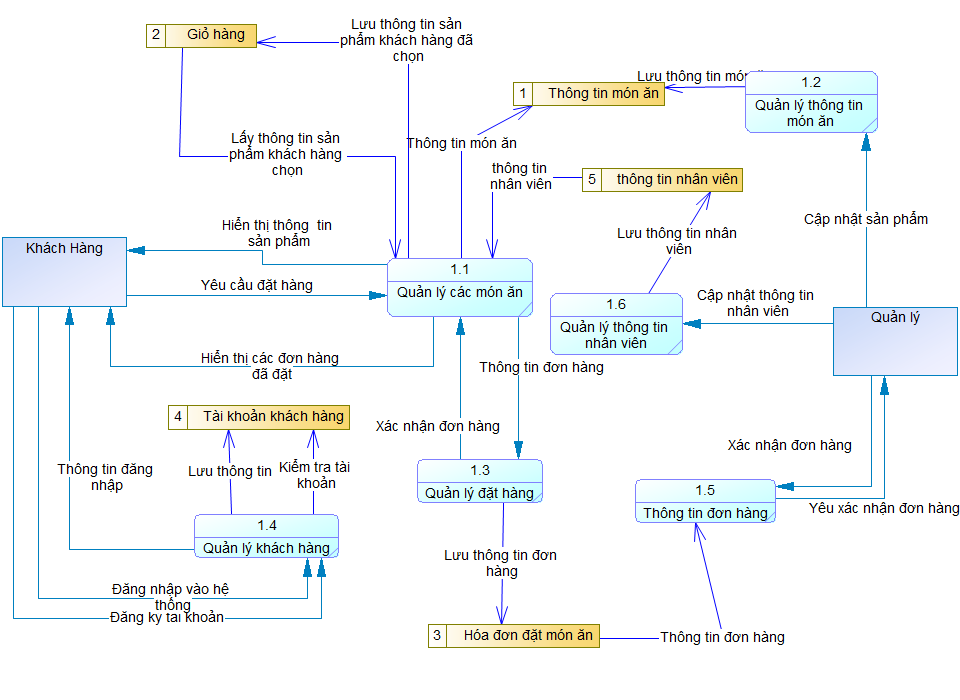
1. Mô hình luồng dữ liệu phân cấp (DFD):

**Mức 0:**

****

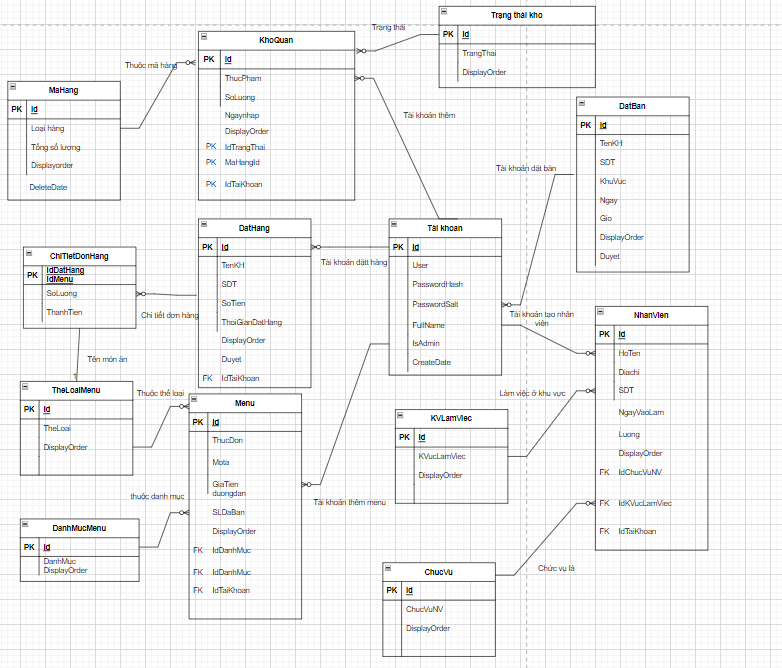
Hình 2: DFD mức 0

**Mức 1:**

****

Hình 3: DFD mức 1

1. Mô hình mức luận lý:



Hình 4: Mô hình luận lý

### **3.3.2: CHUẨN HÓA DỮ LIỆU QUAN HỆ:**

**User** (**Id int**, Name nvarchar(100), Username nvarchar(100), Password nvarchar(200), Email nvarchar(100), PhoneNumber nvarchar(20), Address nvarchar(200), Role\_id int).

**Student** (**Id int**, Mssv int, Fullname nvarchar(200), PhoneNumber nvarchar(20), Address nvarchar(200), Birthday date, Cmnd nvarchar(20), Avatar nvarchar(200), id\_phong int).

**Room** (**Id int**, Room name nvarchar(100), id\_tang int).

**Floor** (**Id int**, number\_floors nvarchar(200)).

**Registration slip** (**Id int**, id\_sinhvien int, condition nvarchar(100)).

**Announcement** (**Id int**, content nvarchar(100)).

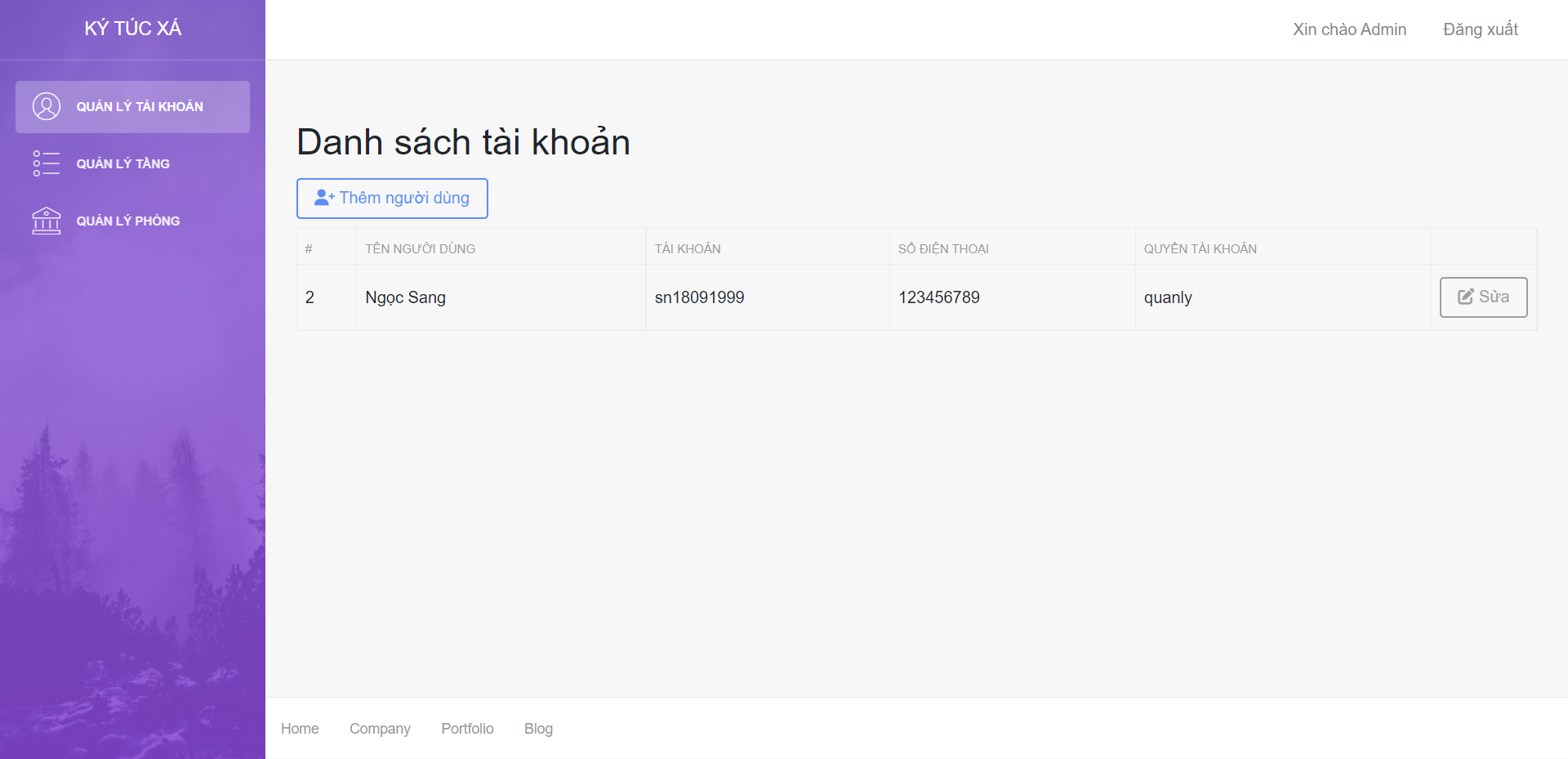
**Role (Id int,** Name nvarchar(100)).

# **CHƯƠNG IV: GIAO DIỆN VÀ KẾT QUẢ**

### **4.1: PHẦN THIẾT KẾ GIAO DIỆN:**

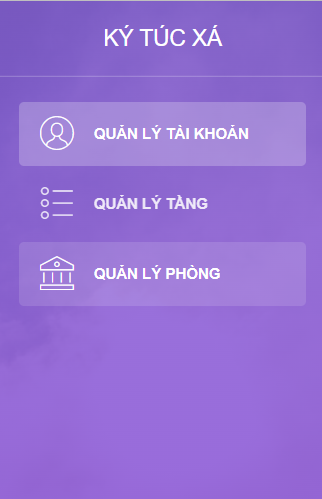
**Phần giao diện của người dùng**

1. **Thiết kế màn hình chính:**

****

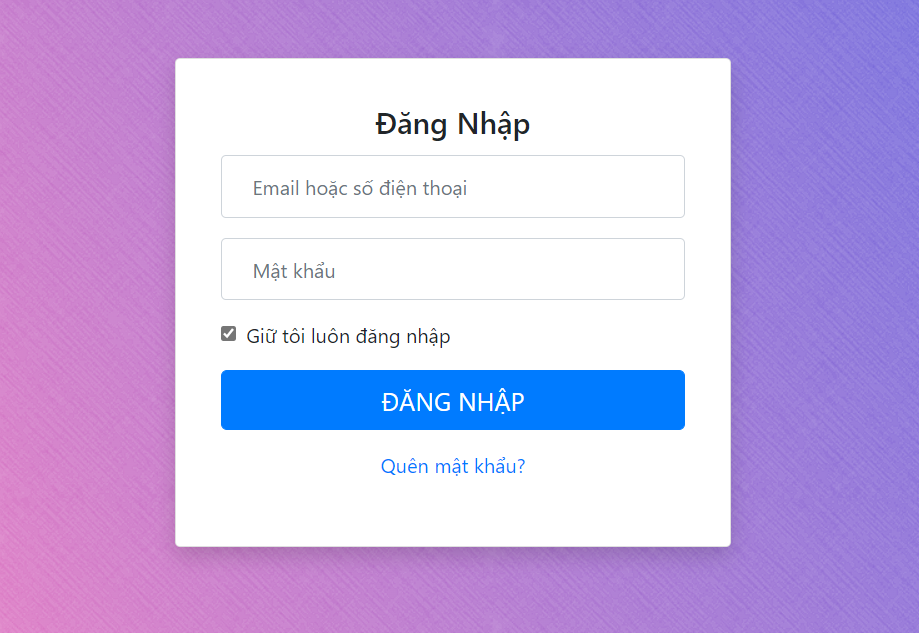
Hình 5: Màn hình chính

1. **Thiết kế menu chính:**

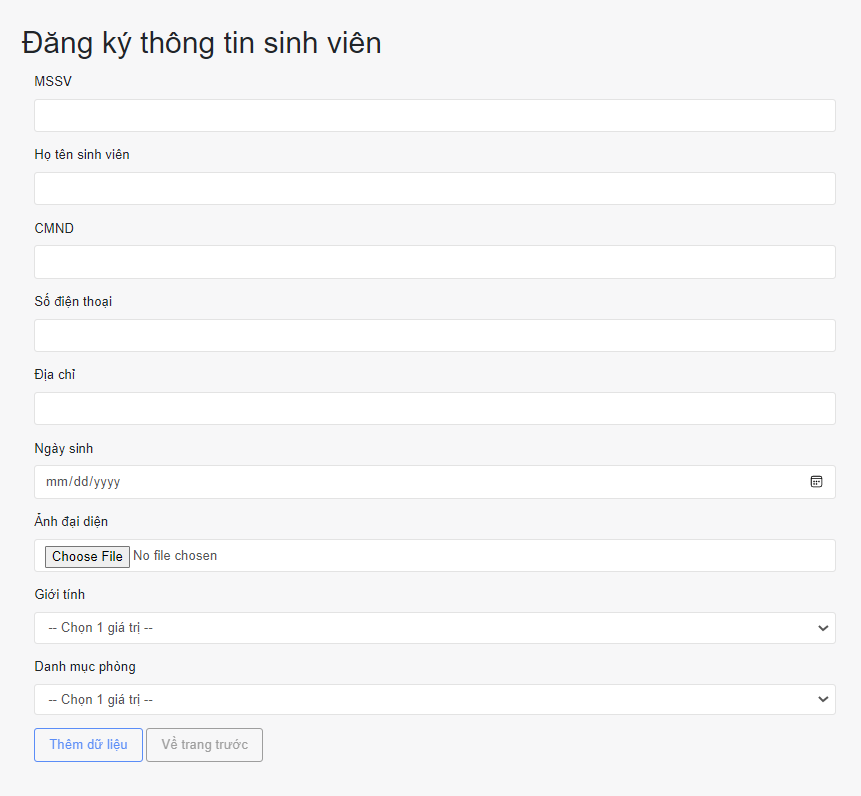


Hình 6: Menu chính

1. **Thiết kế giao diện đăng nhập/ đăng ký (Login/SignIn):**

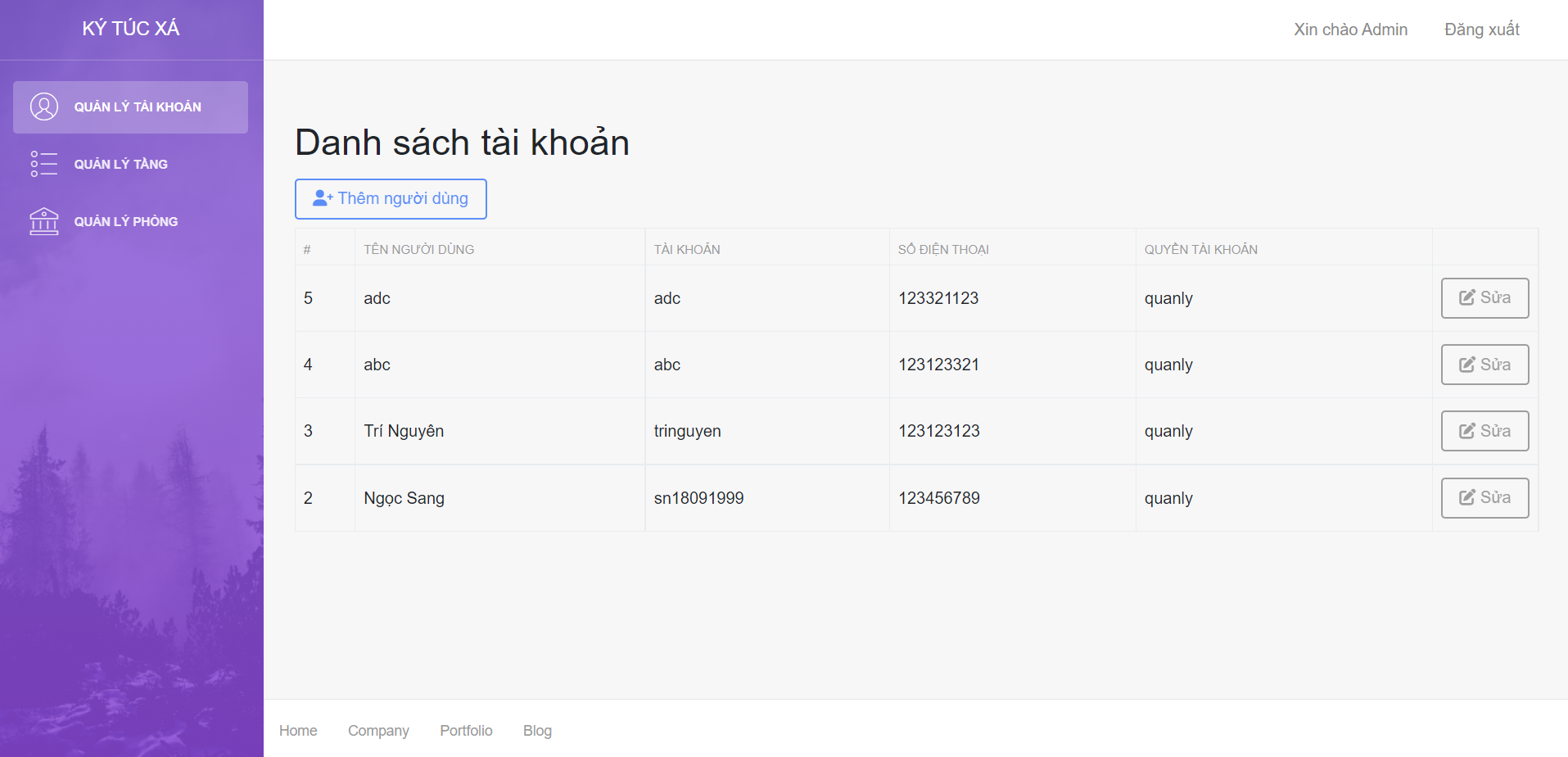


Hình 8: Giao diện đăng nhập (Login) User



Hình 9: Giao diện đăng ký (SignIn) User

1. **Giao diện quản lý tài khoản Admin:**



Hình 13: Giao diện quản lý tài khoản Admin

### **3.4.2: PHẦN THIẾT KẾ CODE:**

#### **1. Code dành cho Area Admin:**

1. **Code AdminControllerBase:**

namespace App\Http\Controllers;

use Illuminate\Foundation\Auth\Access\AuthorizesRequests;

use Illuminate\Foundation\Bus\DispatchesJobs;

use Illuminate\Foundation\Validation\ValidatesRequests;

use Illuminate\Routing\Controller as BaseController;

class Controller extends BaseController

{

use AuthorizesRequests, DispatchesJobs, ValidatesRequests;

}

1. **Code AdminAccountController:**

namespace App\Http\Controllers;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

class AccountController extends Controller

{

public function login() {

return view('admin.account.login');

}

public function userLogin(Request $request)

{

$user = User::where('username', '=', $request->username)->first();

$loginFail = redirect()

->back()

->with("login-err-msg", "Tên đăng nhập hoặc tài khoản không hợp lệ");

if ($user == null) {

return $loginFail;

}

$userData = [

'username' => $request->username,

'password' => $request->password

];

if (Auth::attempt($userData)) {

$request->session()->regenerate();

return redirect('/admin');

} else {

return $loginFail;

}

}

public function logout(Request $request)

{

Auth::logout();

$request->session()->invalidate();

$request->session()->regenerateToken();

return redirect()->route('login');

}

}

1. **Code UserController:**

<?php

<?php

namespace App\Http\Controllers;

use App\Models\User;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Auth;

use Illuminate\Support\Facades\Hash;

use Illuminate\Support\Facades\Validator;

use Illuminate\Validation\Rule;

class UserController extends Controller

{

public function index()

{

$result = User::orderByDesc('id')->where('id', '!=', Auth::id())->paginate(4);

return view('admin.user.index')->with('data', $result);

}

public function create()

{

return view('admin.user.create');

}

public function save(Request $request, $id = null)

{

$data = $request->all();

$rules = [

"name" => ['required'],

"username" => ['required', 'unique:users'],

"email" => ['required', 'email', 'unique:users'],

"dia\_chi" => ["required"],

"role\_id" => ["required"],

"so\_dien\_thoai" => ["required"],

];

$fields = [

'name' => 'Họ và tên',

'username' => "Tên tài khoản",

'email' => "Email",

"dia\_chi" => "required",

"role\_id" => "Danh mục quyền",

"so\_dien\_thoai" => "Số điên thoại",

];

if ($id == null) {

$rules['password'] = ['required', 'min:4'];

$rules['confirmPassword'] = ['same:password'];

$fields['password'] = "Mật khẩu";

$fields["confirmPassword"] = "Xác nhận mật khẩu";

}

if ($id != null) {

// Bỏ qua check unique khi update

$rules["username"] = ['required', Rule::unique('users')->ignore($id)];

$rules["email"] = ['required', 'email', Rule::unique('users')->ignore($id)];

}

unset($data["\_token"]);

$validator = Validator::make($data, $rules, [], $fields);

$validator->validate();

if ($id == null) {

$user = new User();

} else {

$user = User::findOrFail($id);

}

$user->name = $request->name;

$user->username = $request->username;

$user->dia\_chi = $request->dia\_chi;

$user->role\_id = $request->role\_id;

$user->so\_dien\_thoai = $request->so\_dien\_thoai;

$user->email = $request->email;

if ($id == null) {

$user->password = Hash::make($request->password);

}

$user->save();

return redirect()->route('users');

}

public function edit($id)

{

$user = User::findOrFail($id);

return view('admin.user.edit')->with('user', $user);

}

}

1. **Code PhieuDangKyController:**

<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class PhieuDangKyController extends Controller

{

//

}

1. **Code PhongController:**

<?php

namespace App\Http\Controllers;

use App\Models\Phong;

use Illuminate\Http\Request;

class PhongController extends Controller

{

public function index()

{

$result = Phong::withCount(['sinhviens'])->orderBy('id\_tang')->orderByDesc('id')->paginate(5);

return view('admin.phong.index')->with('data', $result);

}

public function save(Request $request, $id = null)

{

if ($id == null) {

$phong = new Phong();

} else {

$phong = Phong::findOrFail($id);

}

$phong->ten\_phong = $request->ten\_phong;

if ($id == null) {

if (isset($request->id\_tang) && $request->id\_tang > 0) {

$phong->id\_tang = $request->id\_tang;

} else {

$phong->id\_tang = 1;

}

if (isset($request->so\_sv\_toida) && $request->so\_sv\_toida > 0) {

$phong->so\_sv\_toida = $request->so\_sv\_toida;

} else {

$phong->so\_sv\_toida = 10;

}

}

$phong->save();

return redirect()->route('phongs');

}

public function getPhong($id)

{

$result = Phong::findOrFail($id);

return response()->json(["data" => $result]);

}

}

1. **Code SinhVienController:**

<?php

namespace App\Http\Controllers;

use App\Models\PhieuDangKy;

use App\Models\SinhVien;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

use Illuminate\Validation\Rule;

class SinhVienController extends Controller

{

public function index()

{

$result = SinhVien::orderByDesc('id')->paginate(5);

return view('admin.sinhvien.index')->with('data', $result);

}

public function create()

{

return view('admin.sinhvien.create');

}

public function save(Request $request, $id = null)

{

$data = $request->all();

$rules = [

"mssv" => ['required', 'unique:sinh\_viens'],

"ho\_ten" => ['required'],

"dia\_chi" => ["required"],

"id\_phong" => ["required"],

"so\_dien\_thoai" => ["required"],

"cmnd" => ["required", 'unique:sinh\_viens'],

"ngay\_sinh" => ["required"],

"gioi\_tinh" => ["required"],

];

$fields = [

'mssv' => "Mã số sinh viên",

'ho\_ten' => 'Họ và tên',

"dia\_chi" => "Địa chỉ",

"id\_phong" => "Danh mục phòng",

"so\_dien\_thoai" => "Số điên thoại",

"cmnd" => "CMND",

"ngay\_sinh" => "Ngày sinh",

"gioi\_tinh" => "Giới tính",

];

unset($data["\_token"]);

if ($id != null) {

// Bỏ qua check unique khi update

$rules["mssv"] = ['required', Rule::unique('sinh\_viens')->ignore($id)];

$rules["cmnd"] = ['required', Rule::unique('sinh\_viens')->ignore($id)];

}

$validator = Validator::make($data, $rules, [], $fields);

$validator->validate();

if ($id == null) {

$sv = new SinhVien();

} else {

$sv = SinhVien::findOrFail($id);

}

$file = $request->file('image');

if ($file) {

$fileName = $file->hashName(); // tạo tên file ngẫu nhiên

$file->storeAs("/public/img/avatar", $fileName);

$sv->anh\_dai\_dien = $fileName;

}

$sv->mssv = $request->mssv;

$sv->ho\_ten = $request->ho\_ten;

$sv->dia\_chi = $request->dia\_chi;

$sv->id\_phong = $request->id\_phong;

$sv->so\_dien\_thoai = $request->so\_dien\_thoai;

$sv->cmnd = $request->cmnd;

$sv->ngay\_sinh = $request->ngay\_sinh;

$sv->gioi\_tinh = $request->gioi\_tinh;

$sv->save();

if ($id == null) {

$sv->phieudangky()->create(['trang\_thai' => false]);

}

return redirect()->route('sinhviens');

}

public function edit($id) {

$sv = SinhVien::findOrFail($id);

return view('admin.sinhvien.edit')->with(['data' => $sv]);

}

public function detail($id) {

$sv = SinhVien::findOrFail($id);

return view('admin.sinhvien.detail')->with(['data' => $sv]);

}

public function delete($id) {

$sv = SinhVien::findOrFail($id);

$sv->delete();

return redirect()->route('sinhviens');

}

public function updateStatus($id) {

$p = PhieuDangKy::where('id\_sinhvien', '=', $id)->first();

$p->trang\_thai = true;

$p->save();

return redirect()->back();

}

public function search(Request $request) {

if(isset($request->search\_sv)) {

$sv = SinhVien::where('mssv', '=', $request->search\_sv)->first();

return view('search')->with(['data' => $sv, 'valInput' => $request->search\_sv]);

} else {

return view('search');

}

}

}

1. **Code TangController:**

<?php

namespace App\Http\Controllers;

use App\Models\Tang;

use Illuminate\Http\Request;

use Illuminate\Support\Facades\Validator;

class TangController extends Controller

{

public function index() {

$result = Tang::withCount(['phongs'])->orderByDesc('id')->paginate(4);

return view('admin.tang.index')->with('data', $result);

}

public function save(Request $request, $id = null) {

if($id == null) {

$tang = new Tang();

} else {

$tang = Tang::findOrFail($id);

}

$tang->so\_tang = $request->so\_tang;

$tang->save();

return redirect()->route('tangs');

}

public function getTang($id) {

$result = Tang::findOrFail($id);

return response()->json(["data" => $result]);

}

}

1. **Code ThongBaoController:**

<?php

namespace App\Http\Controllers;

use App\Models\ThongBao;

use Illuminate\Http\Request;

class ThongBaoController extends Controller

{

public function index()

{

$tb = ThongBao::orderByDesc('id')->paginate(5);

return view('admin.thongbao.index')->with(['data' => $tb]);

}

public function save(Request $request, $id = null)

{

if ($id == null) {

$tang = new ThongBao();

} else {

$tang = ThongBao::findOrFail($id);

}

$tang->noi\_dung = $request->noi\_dung;

$tang->save();

return redirect()->route('thongbaos');

}

public function getThongBao($id)

{

$result = ThongBao::findOrFail($id);

return response()->json(["data" => $result]);

}

public function delete($id) {

$tb = ThongBao::findOrFail($id);

$tb->delete();

return redirect()->route('thongbaos');

}

public function indexClient() {

$tb = ThongBao::orderByDesc('id')->take(5)->get();

return view('welcome')->with(['data' => $tb]);

}

}

# **CHƯƠNG V: KẾT LUẬN VÀ ĐÁNH GIÁ**

**1. Nhận xét:** Sau khi hoàn thiện một số chức năng chính cho website thì đã có thể đáp ứng nhu cầu sử dụng của khách hàng, tuy nhiên đây là lần đầu tiên nhóm tự phát triển và hoàn thiện một website nên còn nhiều sai sót trong quá trình làm cũng như hoàn thiện và chưa tối ưu như những website bán hàng online khác trên thị trường.

**2.** **Ưu điểm:**

* Giao diện thân thiện với người dùng.
* Không có nhiều thao tác phức tạp nên giúp người dùng dễ dàng tiếp cận và sử dụng trang web.
* Giải quyết việc mua điện thoại, laptop,… nhanh chóng thông qua website, tiết kiện thời gian đi lại.

**3. Nhược điểm:** Chức năng, giao diện vẫn chưa chuyên nghiệp so với các website đã có trên Internet hiện nay.

**4. Hướng phát triển:** Từ giao diện chương trình hoạt động trên nền tảng Web có thể phát triển thêm khả năng tương thích về giao diện khi chạy trên nền tảng di động ( Android, Ios,..) để khách hàng đặt mua hàng sẽ nhanh hơn khi không có máy tính từ đó sẽ tiếp cận được với nhiều người dùng hơn.