TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ**

**CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

**Website Quản Lý Quán Karaoke Bằng C#**

*Người hướng dẫn*: **VŨ ĐÌNH HỒNG**

*Người thực hiện*: **PHẠM VŨ QUỐC CƯỜNG – 517H0040**

**TRẦN PHẠM ANH TUẤN–517H0093**

**PHẠM CÔNG–517H0039**

**TÔ QUỐC THẮNG–517H0082**

Lớp : 17050210-17050211

Khoá  **: 21**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ**

**CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

**Website Quản Lý Quán Karaoke Bằng C#**

*Người hướng dẫn*: **VŨ ĐÌNH HỒNG**

*Người thực hiện*: **PHẠM VŨ QUỐC CƯỜNG – 517H0040**

**TRẦN PHẠM ANH TUẤN–517H0093**

**PHẠM CÔNG–517H0039**

**TÔ QUỐC THẮNG–517H0082**

Lớp : 17050210-17050211

Khoá  **: 21**

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu trong suốt thời gian em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công Nghệ Thông Tin, đặc biệt là thầy Vũ Đình Hồng đã tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình hoàn thành đề tài này. Với sự chỉ bảo tận tình của thầy, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu của đề tài. Tuy nhiên, với vốn kiến thức còn hạn chế nên sai sót là điều không thể tránh khỏi trong suốt quá trình làm bài. Chúng em rất mong nhận được những ý kiến đóng góp, sửa chữa, phê bình từ phía thầy cô. Để có thể cải thiện thêm kiến thức của mình và biến chúng thành hành trang vô cùng đáng quý của chúng em trong tương lai sau này.

Ngoài ra, em cũng xin gửi lời cảm ơn tới tất cả bạn bè, đặc biệt các bạn trong lớp luôn gắn bỏ, cùng học tập và đưa ra quan điểm góp phần xây dựng đề tài này.

Em xin chân thành cảm ơn!

# ĐỒ ÁN ĐƯỢC HOÀN THÀNH

# TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Vũ Đình Hồng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 30 tháng 11 năm 2020*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Phạm Vũ Quốc Cường*

*Phạm Công*

*Tô Quốc Thắng*

*Trần Phạm Anh Tuấn*

# PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

Tp. Hồ Chí Minh, ngày tháng năm 2020

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

Tp. Hồ Chí Minh, ngày tháng năm 2020

(kí và ghi họ tên)

MỤC LỤC

[LỜI CẢM ƠN ii](#_Toc51884751)

[ĐỒ ÁN ĐƯỢC HOÀN THÀNH iii](#_Toc51884752)

[TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG iii](#_Toc51884753)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN iv](#_Toc51884754)

[MỤC LỤC 1](#_Toc51884755)

[DANH MỤC CÁC HÌNH ẢNH VÀ BẢNG BIỂU 3](#_Toc51884756)

[I. Phương thức truy cập CSDL trong Java bằng JDBC 5](#_Toc51884757)

[1. JDBC là gì? 5](#_Toc51884758)

[2. Các thành phần hoặc tính năng của JDBC. 5](#_Toc51884759)

[2.1 DriveManager. 5](#_Toc51884760)

[2.2 Driver. 5](#_Toc51884761)

[2.3 Connection. 8](#_Toc51884762)

[2.4 Statement, Prepared Statement, Callable Statement Interface. 8](#_Toc51884763)

[2.5 ResultSet. 9](#_Toc51884764)

[2.6 SQLException. 10](#_Toc51884765)

[3. Code demo 10](#_Toc51884766)

[3.1 Chuẩn bị. 10](#_Toc51884767)

[3.2 Thực hiện. 10](#_Toc51884768)

[II. Phương thức truy cập CSDL trong java bằng Active JDBC 20](#_Toc51884769)

[1. ActiveJDBC là gì? 20](#_Toc51884770)

[2. Các thành phần của ActiveJDBC. 20](#_Toc51884771)

[2.1 Tạo bảng 20](#_Toc51884772)

[2.2 Tạo model 20](#_Toc51884773)

[2.3 Kết nối với Database 21](#_Toc51884774)

[2.4 Thêm dữ liệu 21](#_Toc51884775)

[2.5 Các lệnh truy vấn dữ liệu 22](#_Toc51884776)

[2.6 Cập nhật/ thay đổi dữ liệu 22](#_Toc51884777)

[2.7 Xóa dữ liệu 22](#_Toc51884778)

[2.8 Phân trang 23](#_Toc51884779)

[III. Phương thức truy cập CSLD trong Java bằng JPA 24](#_Toc51884780)

[1. JPA là gì ? 24](#_Toc51884781)

[2. Kiến trúc JPA 24](#_Toc51884782)

[2.1 Entity 25](#_Toc51884783)

[2.2 Entity Manager 25](#_Toc51884784)

[2.3 Entity Manager Factory 26](#_Toc51884785)

[2.4 Persistence 26](#_Toc51884786)

[2.5 Entity Transaction 26](#_Toc51884787)

[2.6 Query 26](#_Toc51884788)

[3. Các tính năng của JPA 27](#_Toc51884789)

[4. JPQL là gì? 28](#_Toc51884790)

[4.1. Các thành phần hoặc tính năng của JPQL. 28](#_Toc51884791)

[5. Criteria API là gì? 29](#_Toc51884792)

[5.1. Các thành phần hoặc tính năng của Criteria API. 30](#_Toc51884793)

[6. Ưu điểm của JPA 30](#_Toc51884794)

[7. Một số ORM Framework hỗ trọ JPA 31](#_Toc51884795)

[8. Code demo 31](#_Toc51884796)

[8.1 Sử dụng annotation để mapping với Database 31](#_Toc51884797)

[8.2 Các cách tạo class truy suất dữ liệu 33](#_Toc51884798)

[IV. Phương thức truy cập CSDL trong java bằng 37](#_Toc51884799)

[1. Hibernate Framework là gì? 37](#_Toc51884800)

[2. Kiến trúc của Hibernate 37](#_Toc51884801)

[3. Hibernate caching 38](#_Toc51884802)

[4. HQL (Hibernate Query Language) 39](#_Toc51884803)

[4.1 Ưu điểm của HQL 39](#_Toc51884804)

[4.2 Các phương thức thông dụng của HQL 39](#_Toc51884805)

[5. Hibernate annotation 39](#_Toc51884806)

[6. Hibernate transaction management 40](#_Toc51884807)

[7. Các ưu điểm của Hibernate. 41](#_Toc51884808)

[7.1 Mã nguồn mở và nhẹ. 41](#_Toc51884809)

[7.2 Hiệu suất nhanh. 41](#_Toc51884810)

[7.3 Truy vấn độc lập với cơ sở dữ liệu. 41](#_Toc51884811)

[7.4 Tự động tạo ra các bảng của cơ sở dữ liệu. 41](#_Toc51884812)

[7.5 Đơn giản hóa các lệnh hợp bảng (Join, InnerJoin). 41](#_Toc51884813)

[7.6 Cung cấp các thống kê truy vấn và trạng thái cơ sở dữ liệu. 41](#_Toc51884814)

[7.7 Hỗ trợ nhiều Database 41](#_Toc51884815)

[7.8 Hỗ trợ nhiều công nghệ 42](#_Toc51884816)

[8. Code demo 42](#_Toc51884817)

[8.1 Mapping Hibernate sử dụng XML 42](#_Toc51884818)

[8.2 Mapping Hibernate sử dụng annotation 45](#_Toc51884819)

[V. SO SÁNH GIỮA CÁC CÁCH TRUY CẬP CƠ SỞ DỮ LIỆU 54](#_Toc51884820)

[VI. REFERENCE 56](#_Toc51884821)

# DANH MỤC CÁC HÌNH ẢNH VÀ BẢNG BIỂU

**DANH MỤC HÌNH**

[Hình 1 1: Java JDBC Logo 6](#_Toc51884915)

[Hình 1 2: JDBC-ODBC birdge driver 7](#_Toc51884916)

[Hình 1 3: Native-API Driver 7](#_Toc51884917)

[Hình 1 4: Network Protocol driver 8](#_Toc51884918)

[Hình 1 5: Thin driver 8](#_Toc51884919)

[Hình 2 1: Tạo Table 21](#_Toc51884926)

[Hình 2 2: Tạo Model 22](#_Toc51884927)

[Hình 2 3: Kết nối Database (DB) 22](#_Toc51884928)

[Hình 2 4: Cập nhật/ thay đổi dữ liệu 23](#_Toc51884929)

[Hình 2 5: Xóa 1 dữ liệu 23](#_Toc51884930)

[Hình 2 6: Xóa tất cả dữ liệu 24](#_Toc51884931)

[Hình 3 1: Kiến trúc JPA 26](#_Toc51884937)

[Hình 3 2: Mối quan hệ giữa các thành phần 28](#_Toc51884938)

[Hình 3 3: Criteria Queries 30](#_Toc51884939)

[Hình 4 1: Hibernate Logo 38](#_Toc51884944)

[Hình 4 2: Kiến trúc Hibernate 39](#_Toc51884945)

[Hình 4 3: Hibernate Caching 40](#_Toc51884946)

[Hình 4 4: Hibernate transaction management 41](#_Toc51884947)

**DANH MỤC BẢNG BIỂU**

[Bảng 1: So sáng giữa JDBC và JPA 1 55](#_Toc51884857)

[Bảng 2: So sánh giữa JPA và Hibernate 2 56](#_Toc51884870)

# I. GIỚI THIỆU ĐỀ TÀI

## I.1. Lời nói đầu về đề tài:

Ngày nay công nghệ thông tin vô cùng phát triển thì mọi người đều sử dụng máy vi tính để làm việc. Công nghệ thông tin cũng được áp dụng rất nhiều vào các lĩnh vực mà điển hình là lĩnh vực quản lý kinh doanh.

Trong một xã hội hiện đại, năng động ngày nay, giải trí đóng vai trò rất quan trọng. Nó giúp con người giải tỏa những căn thẳng trong công việc, học tập… Karaoke là một trong những loại hình giải trí được đông đảo người dân lựa chọn. Như chúng ta đã biết tại các quán karaoke, việc quản lý kinh doanh thì rất quan trọng nhưng lại còn thủ công, chỉ xây dựng và lưu lại trên sổ sách gây mất thời gian, khó khăn trong việc thống nhất, theo dõi và quản lý cho người dùng. Từ thực tế đó, việc xây dựng được phần mềm quản lý kinh doanh cho quán karaoke là rất cần thiết.

Vì vậy em đã “Xây dựng Website quản lý quán karaoke” với mục đích nghiên cứu, tìm hiểu về và xây dựng website để có thể đáp ứng được nhu cầu quản lý sổ sách cho người dùng, giúp cho mọi người có thể tiết kiệm công sức, thời gian và quản lý dễ dàng, tiện lợi hơn.

## I.2. Phân tích phần mềm:

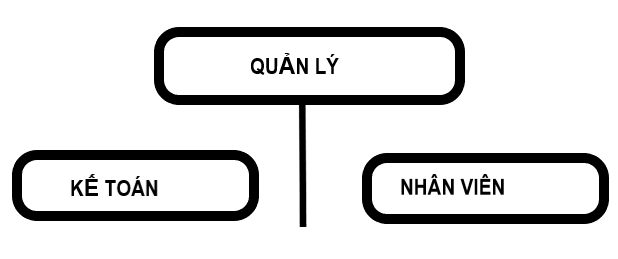
I.2.1 Khảo sát nghiệp vụ

Qua quá trình khảo sát hiện tại đã thu thập được những thông tin nghiệp vụ như sau:

Doanh nghiệp kinh doanh Karaoke có nhiều phòng, mỗi phòng có những thông tin lưu trữ như mã phòng, loại phòng, giá phòng và trạng thái của phòng. Đồng thời sẽ có một trang menu quản lý những thức ăn, nước uống để có thể phục vụ cho khách hàng. Mỗi món sẽ chứa các thông tin như là mã thức ăn, tên thức ăn, giá cả và số lượng hiện đang có trong kho hàng. Mỗi khách hàng sau khi sử dụng xong dịch vụ sẽ được tính tiền và nhận hóa đơn từ nhân viên, thông tin hóa đơn bao gồm: tên nhân viên xử lý hóa đơn, mã hóa đơn, thông tin khách hàng, thời gian sử dụng phòng, thông tin phòng, tiền giờ, tiền dịch vụ, thuế thu nhập VAT.

I.2.2 Mô tả hệ thống:

Doanh nghiệp Karaoke có người quản lý tất cả các hoạt động của doanh nghiệp. Mô hình tổ chức như sau:



Hình 1: Mô hình tổ chức quản lý hoạt động của doanh nghiệp.

* Trong đó:
* Kế toán (Accountant): tham mưu quản lý các hạng mục doanh nghiệp để tra cứu, thanh toán hóa đơn cho khách hàng. Tổng kết doanh thu, đơn hàng và báo cáo tình hình của doanh nghiệp.
* Nhân viên (Staff): Tham mưu quản lý các phòng, dịch vụ của doanh nghiệp. Tiếp nhận thanh toán, xử lý các yêu cầu dịch vụ của khách hàng.
* Quản lý (Admin): Quản lý tất cả các hoạt động của doanh nghiệp, bộ phận thu ngân, nhân viên. Thu nhận tổng kết doanh thu, chi phí của doanh nghiệp. Đề ra kế hoạch kinh doanh, các hoạt động của doanh nghiệp.
* Các hoạt động nghiệp vụ hệ thống:
  + Quản lý thuê phòng:
    - Chi tiết sử dụng dịch vụ.
    - Nhu cầu đổi phòng vì những sự cố bất thường.
    - Thanh toán.
    - In hóa đơn.
  + Quản lý các doanh mục:
    - Doanh mục phòng:
      * Thêm phòng mới.
      * Xóa phòng hiện tại.
    - Doanh mục dịch vụ:
      * Thêm dịch vụ mới.
      * Xóa dịch vụ hiện tại.
      * Sửa thông tin dịch vụ. Các thông tin như giá tiền và số lượng đang có.
    - Doanh mục tài khoản: (Do quản lý thực hiện)
      * Thêm tài khoản mới.
      * Xóa tài khoản hiện tại.
      * Sửa thông tin tài khoản. Các thông tin như tên, số điện thoại, phân quyền tài khoản, mật khẩu tài khoản.
  + Thống kê doanh thu: (Do bộ phận kế toán quản lý)
    - Thống kê theo doanh thu theo tháng.
    - Thống kê số lượng đơn được đặt theo tháng.
    - Thống kê theo các loại phòng đã được đặt.
    - Báo cáo doanh thu theo năm.
  + Tra cứu thông tin, lọc danh sách thuê phòng. (Theo ngày và tên khách hàng).

## I.3. Xác định yêu cầu:

I.3.1 Yêu cầu chức năng nghiệp vụ:

Hỗ trợ các chức năng quản lý tổng quát của phần mềm: quản lý thuê phòng, quản lý các doanh mục, tài khoản, thống kê, tra cứu. Ứng dụng có giao diện thân thiện dễ sử dụng, có khả năng nâng cấp và bổ sung sữa chữa sau này.

I.3.2 Yêu cầu hệ thống:

1. Phân quyền sử dụng:
   * Người quản trị quản lý (admin): được phép sử dụng tất cả các chức năng công cụ của toàn hệ thống.
   * Bộ phận thu ngân và nhân viên: chỉ được phép sử dụng các chức năng công cụ phù thuộc vào chức vụ của mình.
2. Môi trường:
   * Máy cần cài đặt mạng.
   * Máy in trắng đen hoặc cao hơn và khổ giấy cần dùng là A4 theo yêu cầu của file pdf.

3.3 Yêu cầu phi chức năng:

* Hình thức tra cứu tiện dụng, dễ sử dụng cho cả những người không chuyên về tin học. Hỗ trợ tra cứu nâng cao, có hình ảnh minh họa dễ hiểu, dễ sử dụng, thiết kế đơn giản, sắp xếp hợp lí, dễ dàng tìm thấy các công cụ chức năng khi cần.
* Các màn hình có sự nhất quán chung: Các trang đều được hiển thị theo một form chính.
* Tốc độ tra cứu và sử dụng các tính năng từ trung bình đến nhanh.

# II. Cơ sở lý thuyết

## 1. Hỗ trợ:

* **MSSQL:**
  + SQL Server được phát triển bởi Microsoft.
  + Là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System hay được viết tắt là RDBMS) sử dụng câu lệnh SQL (**Transact-SQL**) SQL Server là một máy chủ cơ sở dữ liệu, là sản phẩm phần mềm có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác.
  + Ngoài ra SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đển Tera-Byte và có thể phục vụ cùng lúc cho nhiều user. SQL Server cũng có thể kết hợp với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server… hay các nền tảng như ASP.NET, C# xây dựng winform cũng có thể chạy hoàn hảo với SQL Server.
  + Các hàng và các cột của các bảng dữ liệu chính là đối tượng của SQL Server. Hàng là bảng ghi của bảng, cột được gọi là trường dữ liệu. Cột dữ liệu và kiểu dữ liệu xác định tạo nên cấu trúc của bảng. Khi bảng được tổ chức thành một hệ thống cho một mục đích sử dụng cụ thể vào công việc nào đó sẽ trở thành một cơ sở dữ liệu.
* **HTML (Hypertext Makeup Language):** Giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, blockquotes… .
* **JavaScript:** Là ngôn ngữ lập trình phổ biến được sử dụng để tạo ra những trang web động. Nó được tích hợp và nhúng trong HTML. JavaScript cho phép kiếm soát các hành vi của trang web tốt hơn so với khi chỉ có một mình HTML. JavaScript là ngôn ngữ lập trình được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome… thậm chí các trình duyệt trên thiết bị di động.
* **Entity Framework:** là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. Entity Framework cho phép các nhà phát triển web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. Entity Framework là công cụ làm việc với cơ sở dữ liệu được Microsoft khuyến nghị.

## 2. ASP.NET MVC:

* **ASP.NET:** là một nền tảng dành cho phát triển web, được Microsoft phát hànhvà cung cấp đầu tiên vào năm 2002. Nền tảng được sủ dụng và tạo ra các ứng dụng web-based. Thành phần của ASP.NET và đặc điểm chính:
  + - **Ngôn ngữ:** Có rất nhiều ngôn ngữ lập trình khác nhau tồn tại trong .NET framework. Các ngôn ngữ này là VB.net và C#, có thể được sử dụng để phát triển các ứng dụng web.
    - **Thư viện:** .NET Framework gồm có một bộ các lớp library chuẩn. Library phổ biến nhất được sử dụng cho các ứng dụng web trong .NET là Web library. Web library bao gồm tất cả các thành phần cần thiết để sử dụng trong phát triển các ứng dụng web-based.
    - **Thời gian chạy ngôn ngữ chung (CLR):** các chương trình .net đều được chạy trên nền tảng này. CLR được sử dụng để thực hiện các hoạt động chính. Các hoạt động bao gồm xứ lý ngoại lệ (các lỗi xảy ra trong quá trình thực hiện ứng dụng) và thu gom rác (loại bỏ các tài nguyên không mong muốn khi chúng không còn cần thiết).
    - **Code Behind Mode/ trạng thái code rời:** khái niệm về tách rời thiết kế và mã code giúp việc duy trì ứng dụng ASP.NET dễ dàng hơn. Loại file thông dụng ASP.NET là aspx.
    - **State Management / Quản lí trạng thái:** ASP.NET có cơ sở để quản lý trạng thái. HTTP được biết đến như một giao thức phi trạng thái.
    - **Caching / bộ nhớ đệm:** ASP.NET cũng có thể thực hiện chức năng của caching, qua đó cải thiện hiệu suất làm việc cho ứng dụng. Với việc lưu bộ nhớ đệm, các trang thường xuyên được người dùng yêu cầu có thể lưu trữ ở một vị trí tạm thời. Các trang này có thể truy xuất nhanh hơn và người dùng có thể nhận được phản hồi tốt hơn.
* **MVC:**. viết tắt của Model – View – Controller, là mô hình phân bố source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.
  + - **Model:** là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là cầu nối giữa 2 thành phần 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 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:** là phần giao diện 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. Trong MVC View là nơi tạo ra các HTML. 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 nào với Controller, cũng không lấy được dữ liệu từ Controller mà chỉ hiện thị yêu cầu chuyển cho Controller.
    - **Controller:** là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/function xử ;ý nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiện thì dữ liệu đó ra cho người dùng nhớ lớp View.

**2.1 Sử dụng chú thích (Data Annotation)**

Chú thích dữ liệu (Data Annotation) trong [Entity Framework](https://www.dammio.com/glossary/entity-framework) giúp bạn định nghĩa dữ liệu thuộc tính trực tiếp trong mã nguồn được gieo từ các bảng cơ sở dữ liệu và giúp bạn kiểm soát được dữ liệu nhập vào từ người dùng tốt nhất cũng như cách định nghĩa dữ liệu hiển thị trên giao diện.

Các thuộc tính bao gồm:

* **Validation Attributes (thuộc tính xác thực)**: Dùng để thêm các tập luật xác nhận cho dữ liệu**:**
  + **Timestamp:** chỉ áp dụng cho kiểu dữ liệu thuộc mảng byte, dùng để tạo ta cột dữ liệu thời gian trong database SQL.
  + **ConcurrencyCheck:** áp dụng cho một hay nhiều thuộc tính và các cột tương ứng trong database sẽ được kiếm tra đồng bộ qua mệnh để Where.
  + **Require:** Chỉ định thuộc tính phải có dữ liệu nhập vào trước khi summit về server.
  + **MinLength:** Chiều dài tối thiểu của thuộc tính.
  + **MaxLength:** Chiều dài tối đã của thuộc tính.
  + **StringLength:** định nghĩa chiều dài của thuộc tính, cho phép đặt tả cả chiều dài tối đa lẫn tối thiểu.
  + **Range:** định nghĩa giá trị tối thiểu và tối đa của một thuộc tính.
* **Display Attributes**: để định nghĩa thuộc tính hiển thị, đơn giản bạn chỉ cần thêm dòng [DisplayName(“ tên hiển thị ”)] vào trước các thuộc tính cần hiển thị.
* **Modelling Attributes**: Nếu tạo dự án theo kiểu Code First thì bạn bắt buộc phải dùng các chú thích thuộc tính để định nghĩa rõ thuộc tính nào là khóa chính, khóa ngoại… cũng như các mối quan hệ giữa các lớp bảng trong database sẽ được đặt từ Code First.
  + **Key:** định nghĩa thuộc tính nào là khóa chính.
  + **Table:** bảng.
  + **Column:** cột.
  + **ForeignKey:** Khóa ngoại.

**2.2 Xác nhận dữ liệu (Model Validation)**

Kiểm tra input của User có hợp lệ hay không. Giúp kiểm tra và hiển thị lỗi nếu cần. Việc xác nhận dữ liệu có thể diễn ra ở 2 nơi:

* **Client-side validation:** dữ liệu được kiểm tra ngay ở trên trình duyệt mà không cần phải gửi về server giúp tiết kiệm được băng thông và tăng sự trải nghiệm thay vì phải đợi request đến server và trả về.
* **Server-side validation:** dữ liệu được kiểm tra ở servertrong trường hợp các javascript trên trình duyệt bị disable hoặc là không thể chạy. Ngoài ra còn giúp những dữ liệu mà ở client-side không thể xác nhận được thì sẽ được server xác nhận lại.

Ngoài ra ta còn có **ModelState** là nơi lưu chỉ tiết các thuộc tính của các giá trị được cập nhật lên model và các thông tin lỗi trong quá trình validation xuất hiện trong mỗi thuộc tính.

Những validation attributes thường dùng: Required, MinLength, MaxLength, Range, Datatype, RegularExpression, Url, Compare.

## 2.3 Định tuyến URL (Routing)

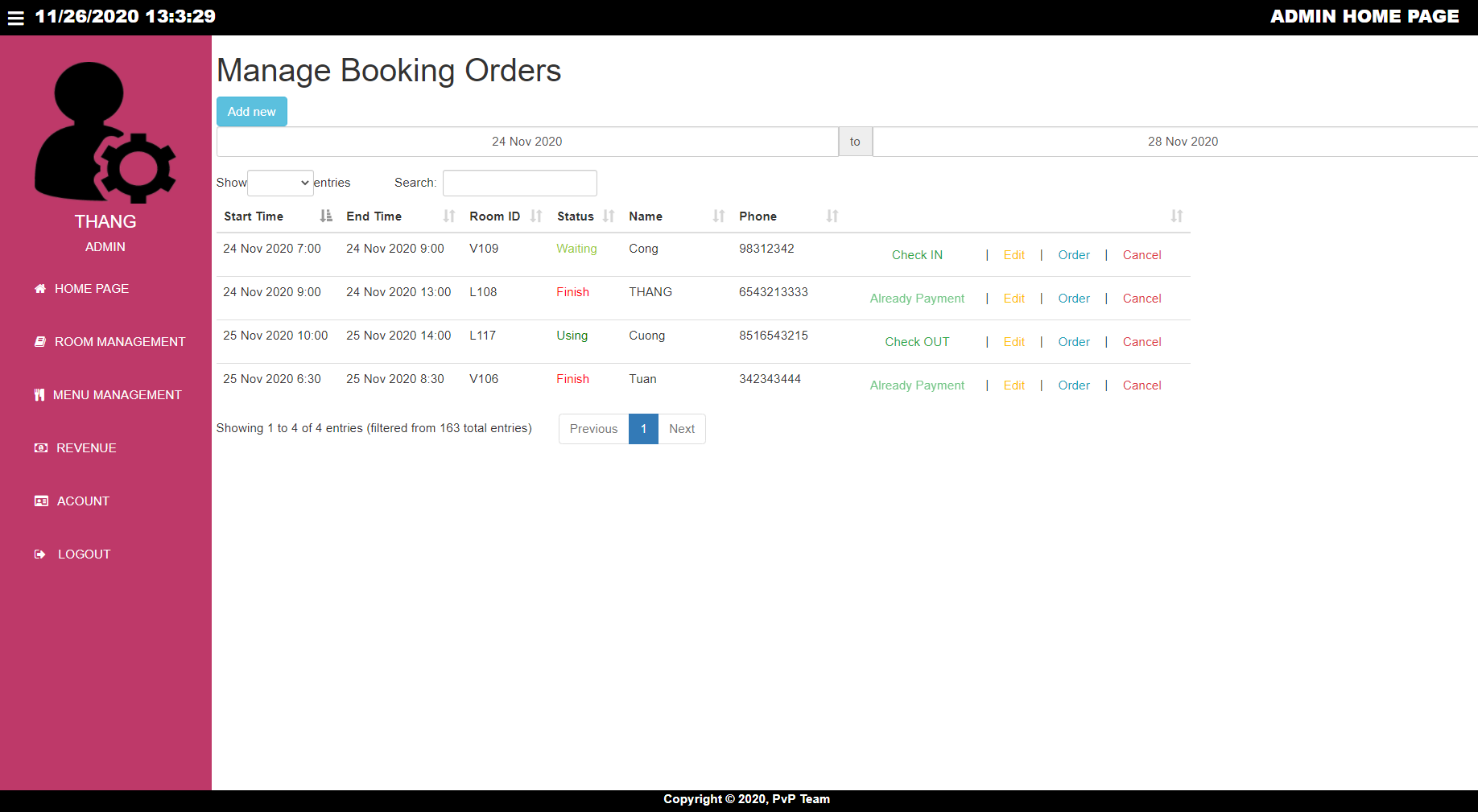
ASP.NET MVC Framework có một hệ thống định tuyến URL (URL Routing System) linh hoạt cho phép xác định các quy tắc ánh xạ địa chỉ URL bên trong ứng dụng. Mỗi hệ thống định tuyến có 2 tác dụng:

* Xây dựng một tập hợp cái URL đi vào ứng dụng và định tuyến chúng tới các Controller và thực thi các phương thức Action để xử lý.
* Xây dựng các URL gửi đi mà có thể gọi ngược trở lại Controllers Action.

Sử dụng các quy tắc ánh xạ URL để điều khiển URL đi vào và đi ra để tăng tính mềm dẻo cho việc lập trình ứng dụng, nghĩa là nếu muốn thay đổi cấu trúc URL (ví dụ Catalog thành Products) có thể thay đổi một tập hợp quy tác ánh xạ mức ứng dụng mà không cần phải viết lại mã lập tình bên trong Controllers và Views.

# VI. KẾT QUẢ ĐẠT ĐƯỢC

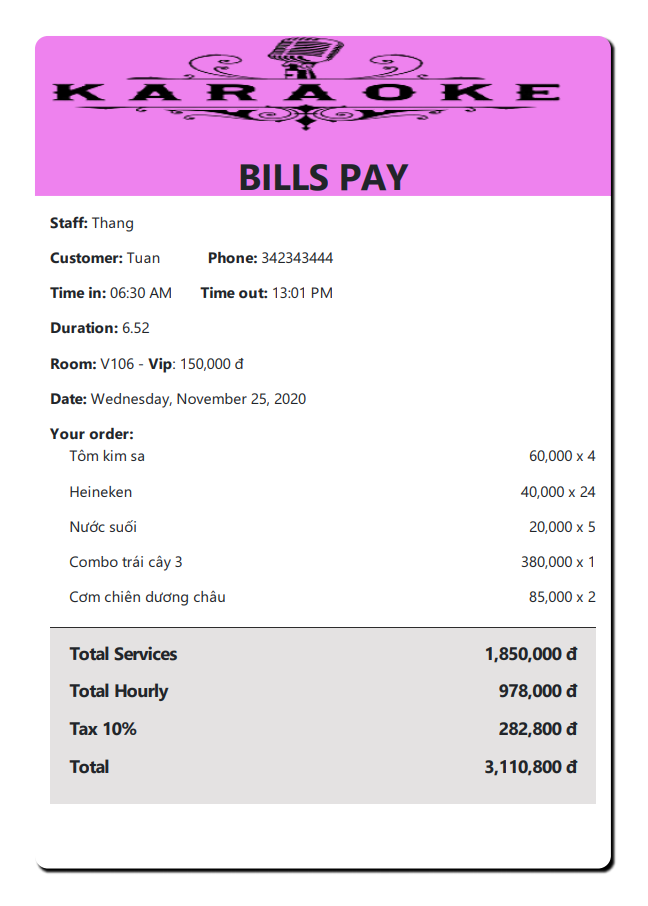
## 1. Home Page:



**Hình 2:** Home Page

* Hiển thị danh sách của các phòng hiện tại đã được đặt bởi khách hàng. Bao gồm:
  + **Start time & End time:** Thời gian mà khách hàng dự muốn bắt đầu và dự kiến kết thúc việc sử dụng phòng.
  + **Room ID:** ID của phòng đang được đặt hoặc đang được sử dụng bới khách hàng.
  + **Status:** tình trạng hiện tại của phòng (có 3 loại trạng thái phòng được cài đặt là: **Waiting –** chưa được khách hàng sử dụng, **Using –** đang được khách hàng sử dụng, **Finish –** đã được khách hàng kết thúc thanh toán).
  + **Name:** tên khách hàng đặt phòng.
  + **Phone:** Số điện thoại của khách hàng đặt phòng.
  + **Nút bấm:**
    - **Đặt trạng thái phòng:** nút có 3 trạng thái tương đương với 3 trạng thái của **Status**. Khi **Status** phòng đang là **Waiting** thì nút sẽ trong trạng thái **Check-in** và khi được bấm vào thì nút sẽ chuyển sang trạng thái **Check-out** và **Status** của phòng cũng sẽ được thay đổi thành **Using**. Cuối cùng khi kết thúc việc sử dụng phòng thì nút **Check-out** sẽ được bấm và sẽ đưa nhân viên đến một page hóa đơn để có thể in ra hóa đơn, đồng thời **Status** của phòng sẽ chuyển sang trạng thái **Finish**.
    - **Edit:** Chỉnh sửa tất cả thông tin của phòng.
    - **Order:** Ghi lại thông tin các món ăn, nước uống mà khách hàng đã đặt trong thời gian sử dụng phòng đồng thời cũng sẽ được thêm vào trong hóa đơn khi thanh toán.
    - **Cancel:** Hủy phòng trong ra khỏi danh sách nếu khách hàng không muốn sử dụng phòng.
* Thanh chọn thời gian để hiển thị danh sách tất cả các phòng trong khoảng thời gian được chọn tránh trường hợp hiển thị những phòng không cần thiết.
* Thanh tìm kiếm để tìm kiếm thông tin cụ thể của 1 phòng bao gồm **Status**, **Room ID**, **Name**, **Phone** trong khoảng thời gian hiển thì của thanh chọn thời gian.
* Drop list để giới hạn số lượng phòng hiện thị trong khoảng thời gian hiển thị của thanh chọn thời gian
* Nút bấm ADD NEW nơi thêm danh sách các phòng được đặt bởi khách hàng.

## 2. Hóa đơn:

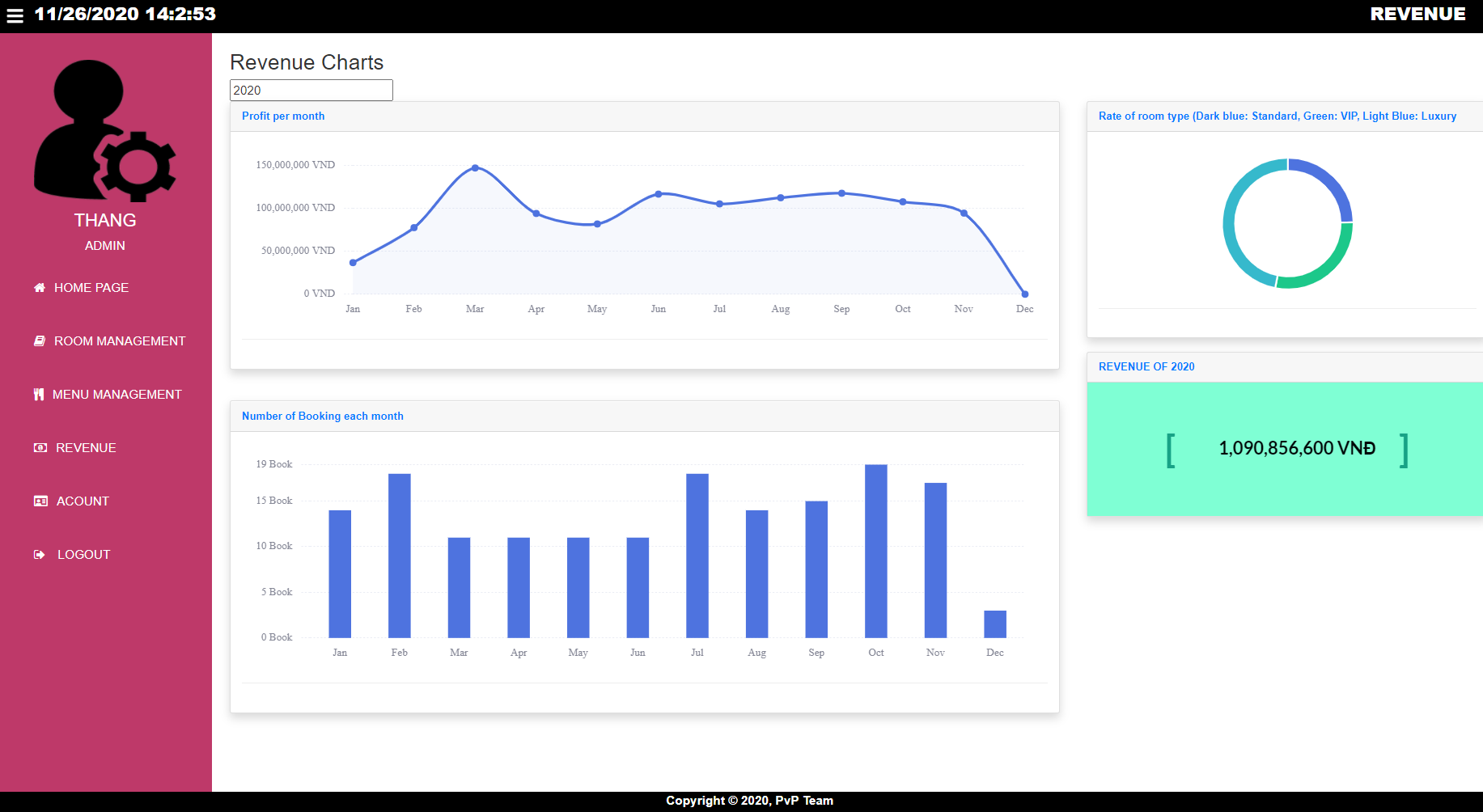


**Hình 3:** Hóa đơn

Hiển thị thông tin phòng, khách hàng và danh sách các món mà khách hàng đã sử dụng trong quá trình sử dụng phòng cùng với tổng giá tiền của từng loại và tổng chi phí khách hàng phải trả.

* **Staff:** tên nhân viên thanh toán được lấy từ thông tin tài khoản đã đăng nhập từ lúc nút bấm **Check-out** được bấm.
* **Customer:** tên của khách hàng đã đặt phòng được lấy từ thông tin phòng.
* **Phone:** Số điện thoại của khách hàng đã đặt phòng được lấy từ thông tin phòng.
* **Time in:** Thời gian mà phòng bắt đầu được sử dụng được lấy theo thông tin phòng.
* **Time out:** Thời gian mà phòng được kết thúc sử dụng và được lấy theo thời gian hiển thị của máy tính.
* **Duration:** Thời gian mà phòng được sử dụng trong bao lâu.
* **Room:** Mã của phòng được sử dụng, loại phòng đồng thời là giá tiền của loại phòng đó trong một giờ sử dụng.
* **Date:** Ngày, tháng, năm của hóa đơn được thanh toán.
* **Your order:** danh sách các món ăn, nước uống và số lượng đã được đặt bởi khách hàng trong quá trình sử dụng phòng.
* **Total Services:** Tổng giá tiền của các món ăn dựa trên các món ăn, nước uống đã được đặt ở **Your order**.
* **Total Hourly:** Giá tiền phòng được quy đổi từ **Duration** (giá tiền 1h nhân với thời gian sử dụng).
* **Tax (10%):** khoảng phí phụ thu VAT là giá trị 10% của tổng tiền **Total Services** và **Total Hourly**.
* **Total:** tổng giá tiền của **Total Services**, **Total Hourly** và **Tax**.

## 3. Biểu đồ:



**Hình 4:** Biểu đồ

* **Biểu đồ đường:** Thể hiện doanh thu hàng tháng của quán trải dài từ tháng 1 cho đến tháng 12 trong một năm dựa trên **Total** của hóa đơn mà khi khách hàng đã hoàn thành thanh toán.
* **Biểu đồ cột:** Thể hiện số lượng khách hàng đã đặt phòng bắt đầu từ tháng 1 cho đến tháng 12 trong năm dựa trên số lượng phòng đã được đặt hàng tháng.
* **Biểu đồ tròn:** Thể hiện tỉ lệ các loại phòng được đặt bởi khách hàng dựa trên loại phòng mà khách hàng đã đặt.
* Hiển thị tổng doanh thu của quán trong 1 năm là tổng của doanh thu từng tháng trong năm.

# VII. KẾT LUẬN

## 1. Ưu, khuyết điểm của đề tài

* Ưu điểm:
  + Có cái nhìn rõ hơn về cách vận hành của một trang web quản lí và cụ thể là quản lí karaoke.
  + Biết được bố cục của một trang web như thế nào.
  + Thành thạo hơn trong quản lí và làm việc với cơ sở dữ liệu.
  + Hiểu rõ hơn về cách hoạt động của mô hình MVC.
* Khuyết điểm:
  + Khó khăn trong việc tìm kiếm tài liệu tham khảo, cụ thể là trang quản lí đã được đưa lên sử dụng.
  + Có nhiều kiến thức mới nên sẽ bị mất khoảng thời gian đầu để làm quen.

## 2. Những gì đã làm được:

* Lọc được loại account khi đăng nhập vào trong web.
* Hiển thị được các tính năng liên quan đến thời gian.
* Hiển thị được trạng thái và thông tin của từng phòng.
* Tương tác qua lại giữa các table trong cơ sơ dữ liệu.
* Thống kê doanh thu theo tháng qua biểu đồ.
* Tính toán theo thời gian thực.
* Nhiều lựa chọn trong quá trình đặt phòng.
* Thanh toán và order món ăn.
* In được hóa đơn thành file pdf.
* Có thanh filter thời gian để dễ dàng quản lý.

## 3. Những gì chưa làm được:

* Thống kê doanh thu theo năm qua biểu đồ.
* Giao diện còn đơn giản.
* Chưa có nhiều tính năng ngoài những tính năng tối thiểu.
* Không có nhiều hình ảnh.

## 4. Hướng phát triển cho đề tài:

* Mở rộng hệ thống cho người dùng ngoài những nhân viên của quán.
* Có tính năng đặt online để khách hàng có thể tự chủ động đặt phòng (cần phải đăng nhập).
* Hệ thống order chủ động dành cho khách hàng (chỉ áp dụng cho loại tài khoản là khách hàng)
* Tính năng khuyến mãi.

# VIII. TÀI LIỆU THAM KHẢO

1. <https://www.chartjs.org/docs/latest/>
2. <https://www.howkteam.vn/learn/lap-trinh/lap-trinh-c-net-7-5>
3. <https://www.w3schools.com/>
4. <https://weblog.west-wind.com/posts/2007/Sep/24/ASPNET-and-Styles-CSS-Embedding>
5. <https://stackoverflow.com/>
6. <https://www.codemag.com/>
7. <https://dotnet.microsoft.com/apps/aspnet/mvc>
8. <https://viblo.asia/newest>
9. <https://xuanthulab.net/asp-net-core-mvc-tao-ung-dung-mvc-dau-tien-c-csharp.html>