

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

....., ngày ..... tháng ..... năm .....

## Giáo viên hướng dẫn

(Ký tên và ghi rõ họ tên)

## NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

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

## LỜI CẢM ƠN

Trong quá trình thực hiện Chuyên đề ASP.NET với Đồ án "*Xây dựng Website đặt phòng khách sạn*", tôi đã nhận được rất nhiều sự giúp đỡ và hỗ trợ từ thầy TS. Đoàn Phước Miền và bạn bè trong lớp VX23TTK13.

Tôi xin gửi lời cảm ơn chân thành đến thầy TS. Đoàn Phước Miền, thầy đã tận tình hướng dẫn, giảng dạy và đồng hành cùng tôi trong quá trình thực hiện chuyên đề này. Những bài chỉ dẫn đơn giản, dễ hiểu và chi tiết của thầy đã tạo nền tảng kiến thức giúp tôi có thể tiếp cận và hiểu rõ hơn về ASP.NET, một framework phát triển web do Microsoft tạo ra, giúp lập trình viên xây dựng các ứng dụng web hiện đại và thực sự là nguồn động lực lớn giúp tôi hoàn thành bài báo cáo.

Cảm ơn các bạn cùng lớp đã chia sẻ tài liệu, trao đổi, thảo luận và hỗ trợ tôi trong suốt quá trình làm chuyên đề.

Mặc dù đã cố gắng để hoàn thiện bài báo cáo tốt nhất có thể, nhưng chắc chắn vẫn còn những thiếu sót. Tôi rất mong nhận được ý kiến đóng góp để làm tốt hơn.

**Một lần nữa, tôi xin chân thành cảm ơn!**

## MỤC LỤC

CHƯƠNG 1: TỔNG QUAN .....	9
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT .....	11
2.1 Tổng quan về ASP.NET framework.....	11
2.1.1 Giới thiệu ASP.NET .....	11
2.1.2 Kiến trúc ASP.NET MVC .....	11
2.1.3 Ưu điểm của ASP.NET MVC.....	11
2.2 Ngôn ngữ lập trình c# .....	12
2.2.1 Tổng quan về C# .....	12
2.2.2 Đặc điểm chính của C# .....	12
2.3 Razor view engine và template.....	13
2.3.1 Razor Syntax .....	13
2.3.2 Layout và Partial Views.....	13
2.3.3 HTML Helpers .....	14
2.4 Javascript và client-side programming .....	14
2.4.1 Vai trò của JavaScript .....	14
2.4.2 Bootstrap Framework.....	14
2.4.3 jQuery và AJAX.....	14
2.5 Cơ sở dữ liệu SQL server .....	14
2.5.1 Giới thiệu SQL Server .....	15
2.5.2 Thiết kế Database.....	15
2.5.3 Chuẩn hóa dữ liệu (Data Normalization).....	15
2.5.4 ADO.NET và Data Access.....	16
2.6 XÁC THỰC VÀ BẢO MẬT HỆ THỐNG .....	16
2.6.1 Quản lý phiên làm việc (Session Management) .....	16
2.6.2 Kiểm tra hợp lệ dữ liệu đầu vào (Input Validation).....	16
2.6.3 Phòng chống tấn công SQL Injection (SQL Injection Prevention) .....	17
2.6.4 Bảo vệ chống giả mạo yêu cầu liên trang (CSRF Protection) .....	17
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU .....	18
3.1 Phân tích và thiết kế hệ thống.....	18
3.1.1 Phân tích yêu cầu .....	18
3.1.2 Use Case Diagram.....	18
3.1.3 Mô hình cơ sở dữ liệu .....	19
3.1.4 Kiến trúc hệ thống.....	21
3.1.5 Công nghệ sử dụng.....	23
3.2 Cài đặt và triển khai .....	25
3.2.1 Thiết lập môi trường phát triển .....	25
3.2.2 Tạo cấu trúc project.....	25
3.2.3 Thiết lập database.....	26
3.2.4 Triển khai tầng truy cập dữ liệu (Data Access Layer) .....	27
3.2.5 Xây dựng Controllers .....	28
3.2.6 Tạo Views .....	29
3.2.7 Triển khai cơ chế kiểm tra dữ liệu (Validation).....	30
3.2.8 Triển khai cơ chế bảo mật (Security) .....	31
3.2.9 Kiểm thử và gỡ lỗi (Testing & Debugging).....	32

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU .....	34
4.1 Chức năng phía người dùng.....	34
4.1.1 Xem danh sách loại phòng .....	34
4.1.2 Xem chi tiết loại phòng .....	35
4.1.3 Thực hiện đặt phòng.....	35
4.1.4 Nhận xác nhận đặt phòng.....	38
4.2 Chức năng phía quản trị viên.....	39
4.2.1 Đăng nhập hệ thống .....	39
4.2.2 Bảng điều khiển quản trị (Admin Dashboard) .....	40
4.2.3 Quản lý loại phòng.....	41
4.2.4 Thêm và sửa loại phòng .....	42
4.2.5 Quản lý đặt phòng .....	44
4.2.6 Xem và cập nhật chi tiết đặt phòng.....	46
4.2.7 Đăng xuất .....	48
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	49
DANH MỤC TÀI LIỆU THAM KHẢO .....	50

## DANH MỤC HÌNH ẢNH

### Chương 3

Hình 3. 1 Sơ đồ Usecase.....	18
Hình 3. 2 Mô hình cơ sở dữ liệu.....	20
Hình 3. 3 Phần mềm Visual Studio 2022 .....	23
Hình 3. 4 SQL Server Management Studio.....	24

### Chương 4

Hình 4. 1 Giao diện trang chủ .....	34
Hình 4. 2 Trang xem chi tiết loại phòng.....	35
Hình 4. 3 Trang đặt phòng.....	36
Hình 4. 4 Kiểm tra dữ liệu trang đặt phòng.....	37
Hình 4. 5 Trang xác nhận đặt phòng .....	38
Hình 4. 6 Trang đăng nhập hệ thống .....	39
Hình 4. 7 Trang Admin Dassboard.....	40
Hình 4. 8 Trang quản lý loại phòng.....	41
Hình 4. 9 Trang thêm loại phòng.....	42
Hình 4. 10 Trang sửa loại phòng .....	43
Hình 4. 11 Trang quản lý đặt phòng.....	44
Hình 4. 12 Trang chi tiết đặt phòng.....	46
Hình 4. 13 Trả lại trang quản trị sau khi đăng xuất.....	48

## TÓM TẮT ĐỒ ÁN

Đồ án tập trung giải quyết bài toán xây dựng hệ thống đặt phòng khách sạn trực tuyến nhằm đáp ứng nhu cầu quản lý và tự động hóa quy trình đặt phòng trong bối cảnh chuyển đổi số của ngành du lịch. Vấn đề nghiên cứu xuất phát từ thực tế là nhiều khách sạn quy mô vừa và nhỏ vẫn đang quản lý thủ công hoặc sử dụng các phương thức đặt phòng truyền thống qua điện thoại, email, dẫn đến nhiều bất cập như mất thời gian, dễ sai sót, khó theo dõi và thiếu tính minh bạch trong việc quản lý thông tin đặt phòng.

Qua quá trình thực hiện đồ án, em đã có cơ hội áp dụng kiến thức lý thuyết vào thực hành, nắm vững quy trình phát triển phần mềm theo mô hình MVC, hiểu sâu về thiết kế cơ sở dữ liệu quan hệ và các nguyên tắc chuẩn hóa, thành thạo trong việc sử dụng ASP.NET MVC framework và SQL Server, cũng như tích lũy kinh nghiệm trong việc xử lý các vấn đề phát sinh trong quá trình phát triển. Đồ án này không chỉ là minh chứng cho năng lực kỹ thuật mà còn là nền tảng vững chắc cho việc tiếp tục nghiên cứu và phát triển các hệ thống phức tạp hơn trong tương lai.

## MỞ ĐẦU

### Lý do chọn đề tài

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ và xu hướng chuyển đổi số đang diễn ra rộng rãi, việc ứng dụng công nghệ thông tin vào quản lý và vận hành khách sạn không chỉ giúp tối ưu hóa quy trình làm việc mà còn nâng cao trải nghiệm của khách hàng, từ đó tăng khả năng cạnh tranh trong thị trường ngày càng khốc liệt.

Xuất phát từ nhu cầu thực tế đó, cùng với mong muốn áp dụng những kiến thức đã được học tập trong chương trình đào tạo về Chuyên đề ASP. NET, em đã lựa chọn đề tài "Xây dựng website đặt phòng khách sạn" cho đồ án môn học của mình. Đề tài này không chỉ có tính ứng dụng cao trong thực tiễn mà còn giúp em củng cố và phát triển kỹ năng lập trình với ASP.NET MVC, thiết kế cơ sở dữ liệu quan hệ, cũng như hiểu sâu hơn về quy trình phát triển một ứng dụng web hoàn chỉnh.

### Mục tiêu nghiên cứu

Mục tiêu chính của đồ án là xây dựng một website quản lý và đặt phòng khách sạn với đầy đủ các chức năng cơ bản, bao gồm:

- Phát triển giao diện thân thiện cho phép khách hàng dễ dàng tìm kiếm, xem thông tin chi tiết và đặt phòng trực tuyến một cách nhanh chóng, tiện lợi.
- Xây dựng hệ thống quản trị cho phép quản trị viên khách sạn quản lý thông tin phòng, theo dõi và xử lý các yêu cầu đặt phòng một cách hiệu quả.
- Áp dụng kiến thức về ASP.NET MVC, ngôn ngữ lập trình C#, và SQL Server để triển khai một ứng dụng web theo mô hình 3 lớp (Three-tier Architecture), đảm bảo tính module hóa và dễ dàng bảo trì, mở rộng.

## CHƯƠNG 1: TỔNG QUAN

Báo cáo tập trung nghiên cứu và giải quyết các vấn đề sau:

**Đối tượng nghiên cứu:** Đối tượng nghiên cứu của đồ án tập trung vào hai nhóm người dùng chính:

- Khách hàng: Là những người có nhu cầu tìm kiếm và đặt phòng khách sạn trực tuyến. Website cung cấp cho họ khả năng xem thông tin chi tiết về các loại phòng (giá cả, tiện nghi, sức chứa), so sánh và lựa chọn phòng phù hợp, sau đó thực hiện đặt phòng với các thông tin cần thiết.

- Quản trị khách sạn: Là những người chịu trách nhiệm quản lý thông tin phòng và xử lý các đơn đặt phòng. Website cung cấp các công cụ để họ thêm, sửa, xóa thông tin loại phòng, xem danh sách các đặt phòng, cập nhật trạng thái đơn hàng (đang chờ, đã xác nhận, đã hủy, hoàn thành).

**Phạm vi nghiên cứu:** Do thời gian và nguồn lực có hạn, đồ án tập trung vào việc xây dựng các chức năng cốt lõi của một hệ thống đặt phòng khách sạn, cụ thể:

- *Về chức năng:* Hệ thống bao gồm các module quản lý loại phòng, quản lý đặt phòng, xác thực và phân quyền người dùng cơ bản (admin). Các tính năng nâng cao như thanh toán trực tuyến, gửi email xác nhận tự động, tích hợp với hệ thống quản lý khách hàng chưa được triển khai trong phạm vi đồ án này.

- *Về công nghệ:* Đồ án sử dụng ASP.NET MVC 5 với .NET Framework 4.7.2, ngôn ngữ lập trình C#, SQL Server cho cơ sở dữ liệu, và Bootstrap 5 cho giao diện người dùng. Ứng dụng được phát triển và chạy trên môi trường localhost với Visual Studio 2022 và IIS Express.

- *Về quy mô:* Hệ thống được thiết kế cho một khách sạn có quy mô vừa và nhỏ, với số lượng loại phòng và đặt phòng ở mức độ vừa phải, phục vụ mục đích học tập và demo.

**Kết quả nghiên cứu:** Đã cho ra một hệ thống hoàn chỉnh với hai module chính phục vụ hai nhóm người dùng. Module dành cho khách hàng bao gồm các chức năng: xem danh sách loại phòng với thông tin đầy đủ về giá cả, sức chứa và tiện nghi; xem

chi tiết từng loại phòng kèm số lượng phòng còn trống được cập nhật real-time; thực hiện đặt phòng với form validation đầy đủ và tính toán tự động tổng chi phí dựa trên số đêm lưu trú; nhận xác nhận đặt phòng ngay lập tức sau khi hoàn tất. Module quản trị viên cung cấp dashboard tổng quan với các chỉ số thống kê; quản lý loại phòng với đầy đủ các thao tác CRUD (Create, Read, Update, Delete); xem danh sách tất cả đặt phòng với khả năng lọc theo trạng thái; xem chi tiết từng đơn đặt phòng bao gồm thông tin khách hàng và yêu cầu đặc biệt; cập nhật trạng thái đặt phòng từ pending sang confirmed, cancelled hoặc completed.

## CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

### 2.1 Tổng quan về ASP.NET framework

#### 2.1.1 Giới thiệu ASP.NET

ASP.NET là một framework phát triển ứng dụng web do Microsoft phát triển, là một phần của nền tảng .NET Framework. ASP.NET cho phép các nhà phát triển xây dựng các ứng dụng web động, trang web và dịch vụ web sử dụng ngôn ngữ lập trình .NET như C#, VB.NET và F#.

ASP.NET cung cấp một mô hình lập trình thống nhất kết hợp các thành phần giao diện người dùng, logic nghiệp vụ và truy cập dữ liệu. Framework này được xây dựng trên Common Language Runtime (CLR), cho phép lập trình viên viết mã ASP.NET bằng bất kỳ ngôn ngữ .NET nào được hỗ trợ.

#### 2.1.2 Kiến trúc ASP.NET MVC

ASP.NET MVC (Model-View-Controller) là một framework để xây dựng ứng dụng web tuân theo mô hình thiết kế MVC. Mô hình này chia ứng dụng thành ba thành phần chính:

**Model (Mô hình):** là các thành phần có nhiệm vụ: Đọc và ghi dữ liệu; Lưu trữ thông tin, trạng thái của các đối tượng; Tất cả các nghiệp vụ logic đều được thực thi ở Model. Dữ liệu được nhập từ người dùng sẽ thông qua View để kiểm tra ở Model trước khi lưu vào CSDL. Việc truy xuất, xác nhận và lưu dữ liệu là phần việc của Model.

**View (Giao diện):** Là các thành phần chịu trách nhiệm hiển thị thông tin cho người dùng thông qua giao diện. Các thông tin cần hiển thị trên View được tạo ra từ các thành phần Models trong mô hình dữ liệu.

**Controller (Bộ điều khiển):** Là các thành phần xử lý tương tác với người dùng, làm việc với Model, chọn một View thích hợp để hiển thị thông tin giao diện người dùng. Trong một ứng dụng MVC, View hiển thị thông tin, còn điều khiển dòng nhập xuất của người dùng vẫn do Controller đảm trách [3].

#### 2.1.3 Ưu điểm của ASP.NET MVC

### **Ưu điểm:**

- Có tính mở rộng do thay thế từng thành phần một cách dễ dàng.
- Không sử dụng ViewState để điều khiển ứng dụng.
- Bổ sung mới hệ thống định tuyến (Routers).
- Hỗ trợ tối đa cho việc kiểm thử bằng việc tạo và cài đặt các unitests tự động.
- Hỗ trợ kết hợp rất tốt giữa người lập trình và thiết kế giao diện.

## **2.2 Ngôn ngữ lập trình c#**

### **2.2.1 Tổng quan về C#**

C# (C Sharp) là ngôn ngữ lập trình hướng đối tượng hiện đại do Microsoft phát triển như một phần của nền tảng .NET. C# kết hợp sức mạnh và tính linh hoạt của C++ với sự đơn giản của Visual Basic. Ngôn ngữ này được thiết kế để phát triển các ứng dụng chạy trên .NET Framework.

### **2.2.2 Đặc điểm chính của C#**

C# theo một hướng nào đó, là ngôn ngữ lập trình phản ánh trực tiếp nhất đến .NET Framework mà tất cả các chương trình.NET chạy, và nó phụ thuộc mạnh mẽ vào framework này. Mọi dữ liệu cơ sở đều là đối tượng, được cấp phát và hủy bỏ bởi trình dọn rác Garbage-Collector (GC), và nhiều kiểu trùu tượng khác chẳng hạn như class, delegate, interface, exception... phản ánh rõ ràng những đặc trưng của.NET runtime.

#### **Ngoài ra C# còn có những ưu điểm:**

- C# là ngôn ngữ lập trình mã nguồn mở, vì vậy C# là miễn phí với tất cả mọi người, đồng thời mọi người đều có thể cùng tham gia phát triển, để xuất thiết kế ngôn ngữ C#.

- C# là ngôn ngữ đa nền tảng vì vậy có thể biên dịch trên nhiều nền tảng máy tính khác nhau (Windows, Linux, MacOS)

- C# có hiệu suất cao và tốc độ thực thi nhanh do sử dụng trình biên dịch trung gian (CLR), điểm cộng nữa là tốc độ phát triển phần mềm nhanh chóng so với đa số các ngôn ngữ hiện tại.

- C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ. ngoài ra có thể viết C# bằng bất kỳ text editor nào khác như Visual Studio Code, Vim, Netbeam...

- C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, song cũng được bổ sung các yếu tố mang tính hiện đại nên dễ dàng tiếp cận cho người mới học và học nhanh với C#.

- C# có cộng đồng nhà phát triển vô cùng lớn mạnh.

- C# được phát triển và cải tiến không ngừng với tần suất 1 phiên bản/ 1 năm, đáp ứng các mong muốn cải thiện, cải tiến cho phù hợp với nhu cầu công nghệ của các nhà phát triển.

- C# có tài liệu tham khảo và hướng dẫn vô cùng phong phú và chất lượng, đồng thời có các buổi hội thảo giới thiệu tính năng mới và định hướng phát triển ngôn ngữ trong tương lai.

- C# và .NET được đánh giá là có design tốt, vì vậy cú pháp và logic rất nhất quán, mã nguồn C# dễ đọc và mở rộng.

- C# được thiết kế và phát triển bởi Microsoft nên rất được Microsoft quan tâm và hỗ trợ [4].

### 2.3 Razor view engine và template

#### 2.3.1 Razor Syntax

Razor là view engine mặc định trong ASP.NET MVC, sử dụng cú pháp @-symbol để nhúng code C# vào HTML. Razor cho phép viết code server-side một cách tự nhiên và dễ đọc hơn so với các view engine truyền thống. Cú pháp Razor bao gồm: @ để bắt đầu code block, @{} cho multi-line code blocks, @model để declare model type, @Html helpers cho generating HTML, và @Url helpers cho URL generation [3].

#### 2.3.2 Layout và Partial Views

**Layout pages** cho phép tạo một mẫu giao diện chung cho toàn bộ website, giúp các trang có cùng bố cục và kiểu hiển thị. Trong Layout, hàm @RenderBody() được dùng để chèn nội dung của từng trang con vào đúng vị trí. **Partial views** là những view nhỏ, có thể tái sử dụng trong nhiều trang khác nhau. Việc dùng partial giúp tránh lặp lại mã và làm cho việc quản lý, bảo trì dễ dàng hơn. [3].

### 2.3.3 HTML Helpers

ASP.NET MVC cung cấp tập hợp các HTML Helper như Html.BeginForm(), Html.TextBoxFor(), Html.LabelFor() nhằm hỗ trợ sinh mã HTML một cách tự động dựa trên mô hình dữ liệu (strongly-typed binding). Các helper này đảm nhiệm việc liên kết dữ liệu giữa giao diện và mô hình (model binding), xử lý kiểm tra hợp lệ dữ liệu (validation), đồng thời tự động tạo các thuộc tính HTML phù hợp, giúp giảm thiểu mã nguồn thủ công và nâng cao tính nhất quán trong quá trình phát triển giao diện [3].

## 2.4 Javascript và client-side programming

### 2.4.1 Vai trò của JavaScript

JavaScript là ngôn ngữ lập trình phía client được thực thi trực tiếp trên trình duyệt, cho phép xây dựng các chức năng tương tác động với người dùng mà không cần tải lại trang web. Thông qua JavaScript, lập trình viên có thể thao tác trên DOM (Document Object Model), xử lý các sự kiện người dùng, kiểm tra tính hợp lệ của dữ liệu nhập vào biểu mẫu và thực hiện các yêu cầu bắt đồng bộ (AJAX) để trao đổi dữ liệu với máy chủ một cách hiệu quả.

### 2.4.2 Bootstrap Framework

Bootstrap là một trong những framework CSS phổ biến nhất hiện nay, cung cấp sẵn hệ thống các thành phần giao diện người dùng (UI) như thanh điều hướng, thẻ hiển thị (card), nút bấm, biểu mẫu, cửa sổ (modal) và hệ thống lưới (grid system) hỗ trợ thiết kế đáp ứng (responsive). Việc sử dụng Bootstrap giúp quá trình xây dựng giao diện được thực hiện nhanh chóng, đảm bảo tính đồng nhất về thiết kế và khả năng tương thích trên nhiều loại thiết bị khác nhau. Trong phạm vi đồ án, framework Bootstrap 5 được ứng dụng để triển khai toàn bộ giao diện theo hướng responsive.

### 2.4.3 jQuery và AJAX

jQuery là thư viện JavaScript giúp đơn giản hóa DOM manipulation, event handling và AJAX calls. AJAX (Asynchronous JavaScript and XML) cho phép gửi và nhận dữ liệu từ server không đồng bộ, cập nhật parts của page mà không cần reload toàn bộ page [5].

## 2.5 Cơ sở dữ liệu SQL server

### 2.5.1 Giới thiệu SQL Server

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. Là một máy chủ cơ sở dữ liệu, nó là một 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. Có thể chạy trên cùng một máy tính hoặc trên một máy tính khác trên mạng (bao gồm cả Internet).

### 2.5.2 Thiết kế Database

Thiết kế cơ sở dữ liệu quan hệ bao gồm việc xây dựng cấu trúc các bảng dữ liệu, xác định các cột, kiểu dữ liệu, các ràng buộc và mối quan hệ giữa các bảng. Khóa chính (Primary Key) được sử dụng để định danh duy nhất mỗi bản ghi trong một bảng, trong khi khóa ngoại (Foreign Key) được thiết lập nhằm liên kết dữ liệu giữa các bảng khác nhau. Ngoài ra, các ràng buộc như NOT NULL, UNIQUE và CHECK được áp dụng để đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong toàn hệ thống.

### 2.5.3 Chuẩn hóa dữ liệu (Data Normalization)

Chuẩn hóa dữ liệu (Data Normalization) là quá trình tổ chức và cấu trúc dữ liệu theo cách loại bỏ sự trùng lặp, đảm bảo tính nhất quán và giảm thiểu các biến đổi dữ liệu. Nó liên quan đến việc chia dữ liệu thành các bảng nhỏ hơn, có quan hệ logic với nhau, giảm thiểu sự lặp lại dữ liệu và thiết lập mối quan hệ giữa các bảng này thông qua các khóa.

Các dạng chuẩn hóa phổ biến trong chuẩn hóa dữ liệu bao gồm:

- Dạng chuẩn hóa đầu tiên (First normal form – 1NF), loại bỏ các cột trùng lặp trong một bảng và đảm bảo rằng mỗi cột chứa các giá trị nguyên tử (không thể chia được).

- Dạng chuẩn hóa thứ hai (Second normal form – 2NF), xây dựng trên cơ sở của 1NF bằng cách loại bỏ các phụ thuộc một phần. Phụ thuộc một phần xảy ra khi một thuộc tính phụ thuộc vào chỉ một phần của khóa chính.

- Dạng chuẩn hóa thứ ba (Third normal form – 3NF), đáp ứng tất cả các yêu cầu của 2NF và loại bỏ các phụ thuộc chuyên tiếp. Phụ thuộc chuyên tiếp xảy ra khi một thuộc tính phụ thuộc vào một thuộc tính khác không phải là khóa chính [7].

#### 2.5.4 ADO.NET và Data Access

ADO.NET là công nghệ truy cập dữ liệu trong nền tảng .NET Framework, cung cấp hệ thống các lớp (class) hỗ trợ kết nối và làm việc với cơ sở dữ liệu một cách hiệu quả. Trong đó, SqlConnection chịu trách nhiệm quản lý kết nối đến cơ sở dữ liệu, SqlCommand thực thi các câu lệnh SQL, SqlDataAdapter dùng để ánh xạ dữ liệu vào các đối tượng DataTable, và SqlParameter hỗ trợ truyền tham số an toàn nhằm phòng tránh nguy cơ tấn công SQL Injection. Bên cạnh đó, tầng truy cập dữ liệu (Data Access Layer – DAL) được thiết kế để đóng gói toàn bộ logic làm việc với cơ sở dữ liệu, đồng thời cung cấp giao diện lập trình rõ ràng cho tầng xử lý nghiệp vụ.

### 2.6 XÁC THỰC VÀ BẢO MẬT HỆ THỐNG

#### 2.6.1 Quản lý phiên làm việc (Session Management)

Session cho phép lưu trữ các thông tin gắn liền với từng người dùng trên máy chủ trong suốt quá trình tương tác với hệ thống thông qua nhiều yêu cầu khác nhau. Trong ASP.NET, cơ chế Session State được sử dụng để lưu trữ dữ liệu dưới dạng cặp khóa – giá trị (key-value pairs) và được truy cập thông qua đối tượng Session.

Cơ chế Session được áp dụng nhằm:

- Duy trì trạng thái đăng nhập của người dùng,
- Lưu trữ thông tin cấu hình cá nhân,
- Quản lý quyền truy cập trong hệ thống.

Việc quản lý Session giúp hệ thống nhận biết người dùng trong suốt phiên làm việc, góp phần đảm bảo tính liên tục và bảo mật của quá trình sử dụng.

#### 2.6.2 Kiểm tra hợp lệ dữ liệu đầu vào (Input Validation)

Kiểm tra hợp lệ dữ liệu đầu vào là một biện pháp bảo mật quan trọng nhằm ngăn chặn các dữ liệu độc hại được gửi đến hệ thống. ASP.NET MVC hỗ trợ cơ chế kiểm tra tính hợp lệ thông qua Data Annotations trong mô hình dữ liệu như: [Required], [StringLength], [EmailAddress], [Range],...

Bên cạnh đó:

- Kiểm tra phía client được thực hiện bằng thư viện jQuery Validation để cung cấp phản hồi tức thời cho người dùng.

- Kiểm tra phía server trong Controller đảm bảo dữ liệu hợp lệ ngay cả trong trường hợp kiểm tra phía client bị vô hiệu hóa.

Cách tiếp cận kết hợp cả hai phía giúp nâng cao độ an toàn và tính toàn vẹn của dữ liệu.

### **2.6.3 Phòng chống tấn công SQL Injection (SQL Injection Prevention)**

SQL Injection là hình thức tấn công lợi dụng lỗ hổng trong việc xây dựng câu lệnh truy vấn cơ sở dữ liệu để chèn mã độc. Để phòng chống kiểu tấn công này, hệ thống áp dụng phương pháp truy vấn tham số hóa (Parameterized Queries) với đối tượng SqlParameter.

Phương pháp này giúp:

- Tách dữ liệu người dùng khỏi cấu trúc câu lệnh SQL
- Ngăn chặn việc thay đổi logic truy vấn
- Đảm bảo các tham số đầu vào được xử lý an toàn bởi hệ quản trị cơ sở dữ liệu.

Đây là một trong những kỹ thuật chuẩn được khuyến nghị trong phát triển phần mềm an toàn.

### **2.6.4 Bảo vệ chống giả mạo yêu cầu liên trang (CSRF Protection)**

Tấn công Cross-Site Request Forgery (CSRF) lợi dụng phiên đăng nhập của người dùng để thực hiện các hành động trái phép mà người dùng không hay biết. ASP.NET MVC cung cấp cơ chế Anti-Forgery Token nhằm ngăn chặn loại hình tấn công này.

Cơ chế hoạt động như sau:

- Máy chủ tạo một mã Token ngẫu nhiên và gắn vào biểu mẫu,
- Token được kiểm tra khi biểu mẫu được gửi lên máy chủ,
- Nếu Token không hợp lệ, yêu cầu sẽ bị từ chối.

Việc sử dụng thuộc tính [ValidateAntiForgeryToken] trên các phương thức Controller giúp đảm bảo rằng chỉ những yêu cầu được sinh ra từ hệ thống hợp lệ mới được xử lý [8].

## CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1 Phân tích và thiết kế hệ thống

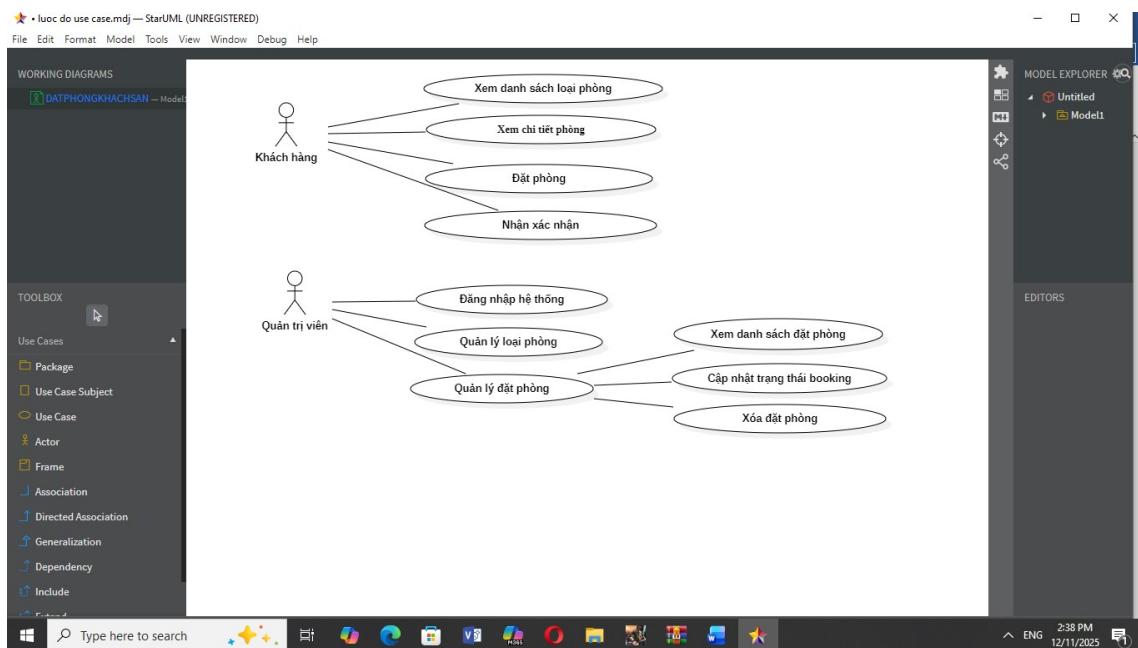
#### 3.1.1 Phân tích yêu cầu

Hệ thống đặt phòng khách sạn trực tuyến được phát triển nhằm đáp ứng nhu cầu quản lý và đặt phòng hiệu quả. Hệ thống phục vụ hai nhóm người dùng chính: khách hàng và quản trị viên.

Khách hàng cần có khả năng xem thông tin chi tiết về các loại phòng bao gồm giá cả, sức chứa, tiện nghi và số lượng phòng còn trống. Họ cần thực hiện đặt phòng trực tuyến một cách dễ dàng với form nhập thông tin đầy đủ validation. Hệ thống cần cung cấp xác nhận ngay lập tức sau khi đặt phòng thành công và giao diện responsive hoạt động tốt trên mọi thiết bị.

Quản trị viên cần đăng nhập vào hệ thống an toàn để quản lý toàn bộ loại phòng với các thao tác thêm, sửa, xóa. Họ cần xem danh sách tất cả đặt phòng với khả năng filter và search, xem chi tiết từng booking với đầy đủ thông tin khách hàng, cập nhật trạng thái đặt phòng từ pending sang confirmed hoặc cancelled, và xem dashboard với các thống kê tổng quan.

#### 3.1.2 Use Case Diagram



Hình 3. 1 Sơ đồ Usecase

---

Hệ thống bao gồm hai actors chính với các use cases tương ứng:

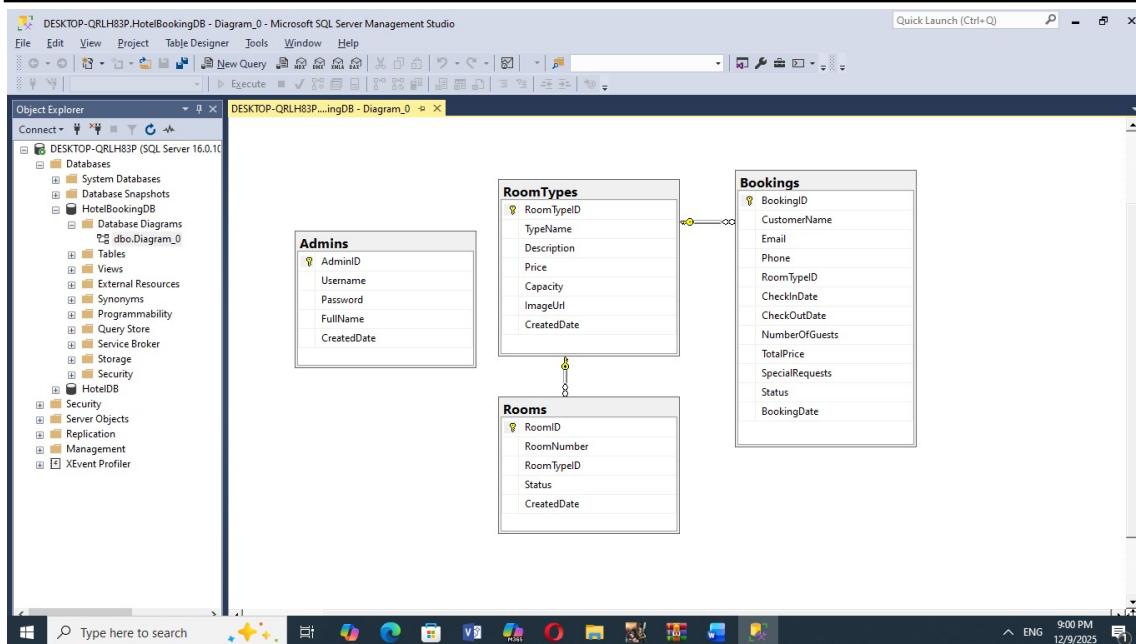
**Actor: Khách hàng**

- Use Case 1: Xem danh sách loại phòng - Khách hàng truy cập trang chủ để browse các loại phòng available.
- Use Case 2: Xem chi tiết phòng - Khách hàng click vào từng loại phòng để xem thông tin chi tiết.
- Use Case 3: Đặt phòng - Khách hàng điền form với thông tin cá nhân và chi tiết booking.
- Use Case 4: Nhận xác nhận - Hệ thống hiển thị trang confirmation sau khi đặt phòng thành công.

**Actor: Quản trị viên**

- Use Case 5: Đăng nhập hệ thống - Admin nhập username và password để authenticate.
- Use Case 6: Quản lý loại phòng - Admin có thể thêm, sửa, xóa room types.
- Use Case 7: Xem danh sách đặt phòng - Admin xem tất cả bookings trong hệ thống.
- Use Case 8: Xem chi tiết đặt phòng - Admin xem thông tin đầy đủ của một booking cụ thể.
- Use Case 9: Cập nhật trạng thái booking - Admin thay đổi status từ pending, confirmed, cancelled, completed.
- Use Case 10: Xóa đặt phòng - Admin có thể delete bookings không cần thiết

### 3.1.3 Mô hình cơ sở dữ liệu



Hình 3. 2 Mô hình cơ sở dữ liệu

Database được thiết kế theo mô hình quan hệ với bốn bảng chính:

**Bảng RoomTypes:** Lưu trữ thông tin các loại phòng với các trường

- RoomTypeID (Primary Key, Identity)
- TypeName (nvarchar(100), NOT NULL) chứa tên loại phòng
- Description (nvarchar(500)) mô tả chi tiết
- Price (decimal(18,2), NOT NULL) giá phòng mỗi đêm
- Capacity (int, NOT NULL) số người tối đa
- ImageUrl (nvarchar(255)) đường dẫn hình ảnh
- CreatedDate (datetime, DEFAULT GETDATE()) ngày tạo record.

**Bảng Rooms:** Quản lý từng phòng cụ thể với

- RoomID (Primary Key, Identity)
- RoomNumber (nvarchar(50), NOT NULL, UNIQUE) số phòng duy nhất
- RoomTypeID (Foreign Key → RoomTypes.RoomTypeID) liên kết đến loại phòng.
  - Status (nvarchar(20), DEFAULT 'Available') trạng thái Available/ Occupied/ Maintenance.

- CreatedDate (datetime, DEFAULT GETDATE()).

**Bảng Bookings:** Lưu trữ thông tin đặt phòng với

- BookingID (Primary Key, Identity)
- CustomerName (nvarchar(100), NOT NULL) tên khách hàng
- Email (nvarchar(100), NOT NULL) email liên hệ
- Phone (nvarchar(20), NOT NULL) số điện thoại
- RoomTypeID (Foreign Key → RoomTypes.RoomTypeID) loại phòng được đặt
- CheckInDate (date, NOT NULL) ngày nhận phòng
- CheckOutDate (date, NOT NULL) ngày trả phòng
- NumberOfGuests (int, NOT NULL) số khách
- TotalPrice (decimal(18,2), NOT NULL) tổng tiền thanh toán
- SpecialRequests (nvarchar(500)) yêu cầu đặc biệt
- Status (nvarchar(20), DEFAULT 'Pending') trạng thái Pending/ Confirmed/ Cancelled/ Completed.
- BookingDate (datetime, DEFAULT GETDATE()) thời điểm đặt phòng.

**Bảng Admins:** Quản lý tài khoản quản trị với

- AdminID (Primary Key, Identity)
- Username (nvarchar(50), NOT NULL, UNIQUE) tên đăng nhập
- Password (nvarchar(255), NOT NULL) mật khẩu
- FullName (nvarchar(100)) họ tên đầy đủ
- CreatedDate (datetime, DEFAULT GETDATE()).

### **Relationships:**

- RoomTypes (1) - (N) Rooms: Một loại phòng có nhiều phòng cụ thể
- RoomTypes (1) - (N) Bookings: Một loại phòng có thể có nhiều bookings.

#### **3.1.4 Kiến trúc hệ thống**

Hệ thống được thiết kế và xây dựng dựa trên mô hình kiến trúc 3-tier (Three-tier)

Architecture) nhằm đảm bảo tính phân tách rõ ràng giữa các thành phần, tăng khả năng bảo trì, mở rộng và nâng cao hiệu suất hoạt động. Ba tầng bao gồm:

- Tầng giao diện (Presentation Layer) chịu trách nhiệm hiển thị thông tin và tiếp nhận dữ liệu từ người dùng. Thành phần này bao gồm:

+ Các tệp giao diện View (Razor .cshtml) được sử dụng để render nội dung HTML ra trình duyệt.

+ Các trang bố cục chung (\_Layout.cshtml, \_AdminLayout.cshtml) nhằm định nghĩa cấu trúc giao diện thống nhất cho toàn bộ hệ thống.

+ Các thư viện phía client như Bootstrap và jQuery nhằm tăng cường khả năng tương tác, kiểm tra dữ liệu đầu vào (validation) và cải thiện trải nghiệm người dùng.

+ Thiết kế giao diện tuân theo chuẩn Responsive Design, đảm bảo hệ thống có thể hiển thị tốt trên nhiều loại thiết bị khác nhau như máy tính, máy tính bảng và điện thoại di động.

- Tầng xử lý nghiệp vụ (Business Logic Layer) đảm nhiệm việc xử lý logic chính của hệ thống và kiểm soát luồng dữ liệu. Tầng này bao gồm:

+ Các Controller (HomeController, BookingController, AdminController) chịu trách nhiệm xử lý yêu cầu từ người dùng, điều phối dữ liệu và thực thi các quy tắc nghiệp vụ.

+ Các Model (RoomType, Room, Booking, Admin) được dùng để định nghĩa cấu trúc dữ liệu và các quy tắc kiểm tra tính hợp lệ (validation rules).

+ Cơ chế quản lý phiên làm việc (Session Management) giúp duy trì trạng thái người dùng trong suốt quá trình sử dụng hệ thống.

+ Chức năng Authentication và Authorization đảm bảo kiểm soát truy cập, chỉ cho phép quản trị viên thực hiện các thao tác quản lý hệ thống.

- Tầng truy cập dữ liệu (Data Access Layer ) chịu trách nhiệm giao tiếp trực tiếp với cơ sở dữ liệu. Thành phần này bao gồm:

+ Lớp DatabaseHelper cung cấp các phương thức dùng chung cho các thao tác truy vấn cơ sở dữ liệu.

+ Các lớp DAL chuyên biệt như RoomTypeDAL, BookingDAL đóng vai trò

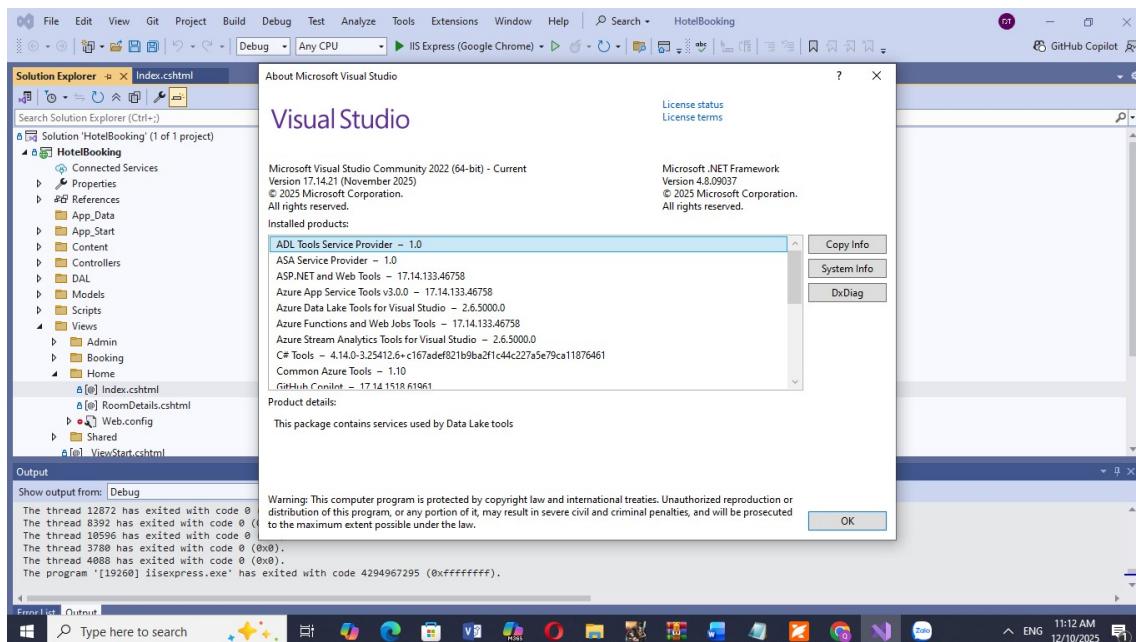
đóng gói logic xử lý dữ liệu.

- + Sử dụng Parameterized Query nhằm hạn chế nguy cơ tấn công SQL Injection.
- + Chuỗi kết nối (Connection String) được quản lý tập trung trong file Web.config nhằm đảm bảo tính linh hoạt và bảo mật cho hệ thống.

### 3.1.5 Công nghệ sử dụng

Hệ thống được xây dựng dựa trên nhiều công nghệ phổ biến và hiện đại nhằm đảm bảo hiệu năng, tính ổn định và khả năng mở rộng. Các công nghệ chính được sử dụng bao gồm:

#### Công nghệ phía Backend



Hình 3. 3 Phần mềm Visual Studio 2022

- Hệ thống sử dụng framework ASP.NET MVC 5 làm nền tảng xây dựng cấu trúc ứng dụng web theo mô hình MVC, giúp tách biệt rõ ràng giữa giao diện, xử lý nghiệp vụ và dữ liệu.
- Ngôn ngữ lập trình C# được sử dụng để xây dựng các chức năng xử lý phía server.
- Môi trường thực thi .NET Framework 4.7.2 đảm bảo tính tương thích và ổn định trong quá trình vận hành hệ thống.

- Bên cạnh đó, công nghệ ADO.NET được sử dụng để kết nối và thao tác với cơ sở dữ liệu một cách hiệu quả.

### Công nghệ phía Frontend

- Giao diện hệ thống được xây dựng dựa trên HTML5 nhằm đảm bảo cấu trúc nội dung chuẩn và dễ mở rộng.

- Ngôn ngữ định dạng CSS3 được sử dụng để thiết kế giao diện, áp dụng hiệu ứng và tạo hoạt ảnh (animations).

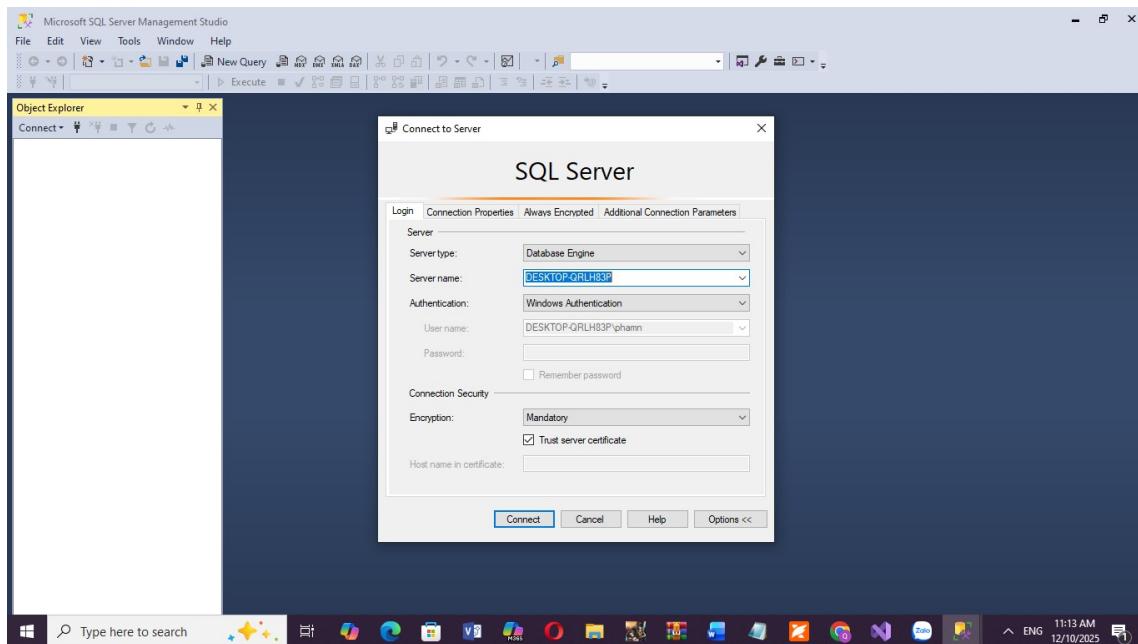
- Framework Bootstrap 5 giúp xây dựng giao diện theo hướng responsive, tương thích trên nhiều thiết bị khác nhau.

- Ngôn ngữ lập trình JavaScript được sử dụng để tăng cường tính tương tác phía người dùng.

- Thư viện jQuery hỗ trợ thao tác với DOM, xử lý sự kiện và thực hiện AJAX.

- Ngoài ra, hệ thống sử dụng Font Awesome để cung cấp các biểu tượng (icons), giúp tăng tính trực quan cho giao diện.

### Công nghệ cơ sở dữ liệu



Hình 3. 4 SQL Server Management Studio

- Hệ quản trị cơ sở dữ liệu Microsoft SQL Server 2022 trở lên được sử dụng để lưu trữ và quản lý dữ liệu hệ thống.

- Ngôn ngữ T-SQL được áp dụng để xây dựng các câu truy vấn và thủ tục lưu trữ (Stored Procedures).

- Công cụ SQL Server Management Studio (SSMS) hỗ trợ quản trị, sao lưu và tối ưu hóa cơ sở dữ liệu.

### Công cụ phát triển

- Môi trường phát triển Visual Studio 2022 được sử dụng làm công cụ chính phục vụ lập trình và quản lý mã nguồn.

- IIS Express hỗ trợ chạy thử và triển khai hệ thống trên môi trường local trong quá trình phát triển.

- Trình quản lý gói NuGet Package Manager được sử dụng để cài đặt, cập nhật và quản lý các thư viện phụ thuộc của dự án.

## 3.2 Cài đặt và triển khai

### 3.2.1 Thiết lập môi trường phát triển

Quá trình setup bắt đầu với việc cài đặt Visual Studio 2022 với workload ASP.NET and web development. Tiếp theo là cài đặt SQL Server 2022 Express hoặc phiên bản cao hơn cùng với SQL Server Management Studio để quản lý database. Sau khi môi trường cơ bản đã sẵn sàng, developer tạo new ASP.NET Web Application project với template MVC và target framework là .NET Framework 4.7.2.

### 3.2.2 Tạo cấu trúc project

Dự án được tổ chức theo đúng quy ước (convention) của mô hình ASP.NET MVC, nhằm đảm bảo tính rõ ràng, dễ quản lý và thuận tiện trong quá trình phát triển cũng như bảo trì hệ thống. Cấu trúc thư mục chính của hệ thống được trình bày như sau:

**Thư mục Models:** Thư mục Models chứa các lớp (class) định nghĩa các thực thể (entity) trong hệ thống, bao gồm: RoomType.cs, Room.cs, Booking.cs, Admin.cs. Các lớp này mô tả cấu trúc dữ liệu thông qua các thuộc tính (properties) và sử dụng Data Annotations để ràng buộc dữ liệu, kiểm tra tính hợp lệ (validation) và hỗ trợ ánh xạ dữ liệu với cơ sở dữ liệu.

**Thư mục DAL (Data Access Layer):** Thư mục DAL chịu trách nhiệm xử lý các thao tác truy cập dữ liệu. Trong đó: Lớp DatabaseHelper.cs cung cấp các phương thức dùng chung cho việc kết nối và làm việc với cơ sở dữ liệu. Các lớp chuyên biệt như RoomTypeDAL.cs, BookingDAL.cs thực hiện các thao tác CRUD (Create, Read, Update, Delete) đối với từng đối tượng dữ liệu cụ thể. Cách tổ chức này giúp phân tách rõ ràng giữa tầng xử lý nghiệp vụ và tầng truy cập dữ liệu, tăng khả năng tái sử dụng mã nguồn.

**Thư mục Controllers:** Thư mục Controllers chứa các bộ điều khiển xử lý yêu cầu từ phía người dùng, bao gồm: HomeController.cs: Quản lý các trang giao diện công khai (public pages). BookingController.cs: Xử lý quy trình đặt phòng của khách hàng. AdminController.cs: Cung cấp các chức năng quản trị hệ thống. Các controller đóng vai trò trung gian, tiếp nhận yêu cầu, xử lý nghiệp vụ và trả kết quả về cho View.

**Thư mục Views:** Thư mục Views được tổ chức theo tên của các controller tương ứng, mỗi thư mục con chứa các tệp Razor View (.cshtml) dùng để hiển thị giao diện người dùng. Việc tổ chức này giúp liên kết rõ ràng giữa Controller và View, tăng tính dễ đọc và dễ quản lý trong quá trình phát triển.

### 3.2.3 Thiết lập database

Quá trình xây dựng cơ sở dữ liệu được thực hiện theo các bước sau:

- Trước tiên, sử dụng công cụ SQL Server Management Studio (SSMS) để kết nối đến máy chủ SQL Server cục bộ (local instance). Sau khi kết nối thành công, nhà phát triển thực thi các tập lệnh SQL (SQL Script) để tạo cơ sở dữ liệu với tên HotelBookingDB.

- Tiếp theo, trong cơ sở dữ liệu này tiến hành khởi tạo các bảng dữ liệu chính bao gồm:

**RoomTypes, Rooms, Bookings, Admins.** Mỗi bảng được thiết kế đầy đủ các trường dữ liệu (columns), kiểu dữ liệu (data types), các ràng buộc (constraints) như khóa chính, khóa ngoại và mối quan hệ (relationships) nhằm đảm bảo tính toàn vẹn dữ liệu.

- Sau khi tạo cấu trúc bảng, dữ liệu mẫu (sample data) được thêm vào hệ thống để phục vụ cho việc thử nghiệm và kiểm tra chức năng (testing).

- Cuối cùng, chuỗi kết nối (Connection String) được cấu hình trong tệp Web.config theo đúng định dạng tương ứng với chế độ xác thực, bao gồm:

- + Windows Authentication (xác thực qua tài khoản hệ thống), hoặc
- + SQL Server Authentication (xác thực bằng tài khoản SQL Server).

Việc cấu hình chính xác chuỗi kết nối giúp hệ thống có thể truy cập cơ sở dữ liệu một cách ổn định và an toàn.

### 3.2.4 Triển khai tầng truy cập dữ liệu (Data Access Layer)

Lớp DatabaseHelper được xây dựng nhằm chuẩn hóa và tái sử dụng các thao tác làm việc với cơ sở dữ liệu. Lớp này cung cấp các phương thức tĩnh (static methods) phục vụ cho quá trình truy vấn và xử lý dữ liệu, bao gồm:

+ ExecuteQuery(): Thực thi các câu lệnh SELECT và trả về kết quả dưới dạng DataTable.

+ ExecuteNonQuery(): Thực hiện các thao tác ghi dữ liệu như INSERT, UPDATE, DELETE.

+ ExecuteScalar(): Thực thi các câu truy vấn chỉ trả về một giá trị duy nhất (single value), thường dùng cho các phép đếm hoặc kiểm tra tồn tại.

+ GetConnection(): Khởi tạo đối tượng SqlConnection dựa trên chuỗi kết nối đã cấu hình trong file Web.config.

Bên cạnh đó, mỗi lớp DAL đảm nhận chức năng xử lý dữ liệu cho một thực thể cụ thể trong hệ thống. Ví dụ lớp RoomTypeDAL cung cấp các phương thức:

+ GetAllRoomTypes(): Lấy danh sách toàn bộ loại phòng, trả về dưới dạng List<RoomType>.

+ GetRoomTypeByID(int id): Truy vấn thông tin chi tiết của một loại phòng theo mã định danh.

+ Insert(RoomType): Thêm mới một loại phòng vào cơ sở dữ liệu.

+ Update(RoomType): Cập nhật thông tin loại phòng.

+ Delete(int id): Xóa bản ghi loại phòng theo mã số.

Việc tổ chức các lớp theo hướng phân tách rõ ràng như trên giúp hệ thống đảm bảo tính dễ mở rộng, dễ bảo trì và tăng khả năng tái sử dụng mã nguồn.

### 3.2.5 Xây dựng Controllers

Các Controller trong hệ thống được xây dựng nhằm xử lý yêu cầu từ phía người dùng, điều phối luồng dữ liệu và liên kết giữa Model và View theo mô hình MVC.

**HomeController:** Lớp HomeController đảm nhiệm việc hiển thị các trang công khai của hệ thống. Các chức năng chính bao gồm:

- Index(): Hiển thị danh sách các loại phòng hiện có trong hệ thống.
- RoomDetails(int id): Hiển thị thông tin chi tiết của một loại phòng cụ thể dựa trên mã định danh.

**BookingController:** Lớp BookingController chịu trách nhiệm xử lý quy trình đặt phòng của khách hàng, bao gồm:

- Create() (GET): Hiển thị biểu mẫu đặt phòng với danh sách lựa chọn loại phòng thông qua danh sách thả xuống (dropdown list).
- Create(Booking) (POST): Tiến hành kiểm tra dữ liệu đầu vào (validation), tính toán tổng chi phí đặt phòng, lưu thông tin đặt phòng vào hệ thống và chuyển hướng người dùng đến trang xác nhận (Confirmation Page) sau khi hoàn tất.

**AdminController:** Lớp AdminController đảm nhận các chức năng quản trị hệ thống, bao gồm:

- Login() (GET/POST): Thực hiện xác thực đăng nhập cho quản trị viên.
- Dashboard(): Hiển thị giao diện tổng quan với các số liệu thống kê (statistics).
- RoomTypes(): Hiển thị danh sách tất cả các loại phòng.
- CreateRoomType(), EditRoomType(), DeleteRoomType(): Thực hiện các thao tác quản lý loại phòng.
- Bookings(): Hiển thị danh sách toàn bộ các lượt đặt phòng.
- BookingDetails(int id): Hiển thị thông tin chi tiết của một lượt đặt phòng.
- UpdateBookingStatus(): Cập nhật trạng thái đặt phòng.

- 
- Logout(): Thực hiện đăng xuất và xóa thông tin phiên làm việc (session).

Việc phân chia chức năng rõ ràng giữa các controller giúp đảm bảo hệ thống vận hành logic, dễ quản lý và thuận tiện trong quá trình mở rộng chức năng.

### 3.2.6 Tạo Views

Hệ thống giao diện người dùng được xây dựng dựa trên cơ chế Layout và View của ASP.NET MVC nhằm đảm bảo tính nhất quán và dễ bảo trì trong thiết kế.

**Trang Layout:** Hai trang bố cục chính được sử dụng trong hệ thống bao gồm:

- \_Layout.cshtml: Áp dụng cho các trang giao diện công khai, bao gồm thanh điều hướng (navigation bar), phần chân trang (footer) và khu vực hiển thị nội dung chính (RenderBody).

- \_AdminLayout.cshtml: Dành riêng cho giao diện quản trị, bao gồm menu bên (sidebar), bố cục đặc thù cho quản trị viên và các thành phần giao diện được thiết kế phù hợp với chức năng quản lý.

#### Giao diện phía người dùng (Home và Booking Views):

Nhóm giao diện Home:

- Index.cshtml: Hiển thị danh sách loại phòng dưới dạng thẻ (card layout).
- RoomDetails.cshtml: Trình bày thông tin chi tiết của từng loại phòng.

Nhóm giao diện Booking:

- Create.cshtml: Giao diện biểu mẫu đặt phòng với thiết kế hai cột (two-column layout), hỗ trợ kiểm tra dữ liệu phía client (client-side validation).
- Confirmation.cshtml: Hiển thị thông báo xác nhận đặt phòng thành công.

**Giao diện quản trị (Admin Views):** Các trang giao diện cho quản trị viên bao gồm:

- Login.cshtml: Giao diện đăng nhập đơn giản.
- Dashboard.cshtml: Hiển thị các thẻ thống kê (statistic cards) nhằm cung cấp cái nhìn tổng quan về hoạt động hệ thống.

- RoomTypes.cshtml: Hiển thị danh sách loại phòng dưới dạng bảng, kèm theo các nút thao tác (action buttons).

- CreateRoomType.cshtml và EditRoomType.cshtml: Cung cấp biểu mẫu nhập và chỉnh sửa thông tin loại phòng.

- Bookings.cshtml: Hiển thị danh sách tất cả các lượt đặt phòng với trạng thái được thể hiện bằng nhãn (status badges).

- BookingDetails.cshtml: Hiển thị thông tin chi tiết của từng lượt đặt phòng và các nút cập nhật trạng thái.

Việc tổ chức giao diện rõ ràng theo từng nhóm chức năng giúp tăng tính thân thiện với người dùng, đồng thời hỗ trợ quản trị viên dễ dàng thao tác và theo dõi hoạt động của hệ thống.

### **3.2.7 Triển khai cơ chế kiểm tra dữ liệu (Validation)**

Cơ chế kiểm tra dữ liệu được áp dụng nhằm đảm bảo tính chính xác, đầy đủ và hợp lệ của thông tin đầu vào trong hệ thống.

**Kiểm tra dữ liệu tại Model (Model Validation):** Hệ thống sử dụng các Data Annotations để ràng buộc dữ liệu ở tầng Model, bao gồm các thuộc tính như:

[Required]: Bắt buộc nhập dữ liệu.

[EmailAddress]: Kiểm tra định dạng địa chỉ email.

[Phone]: Kiểm tra định dạng số điện thoại.

[Display(Name = "...")]: Thiết lập nhãn hiển thị cho các trường dữ liệu.

[StringLength]: Giới hạn độ dài chuỗi ký tự.

[Range]: Xác định phạm vi hợp lệ của giá trị.

Các thuộc tính này giúp chuẩn hóa dữ liệu ngay từ giai đoạn nhập liệu.

**Kiểm tra dữ liệu phía client (Client-side Validation):** Hệ thống kích hoạt cơ chế kiểm tra dữ liệu phía trình duyệt thông qua thư viện jQuery Validation. Thư viện này thực hiện kiểm tra dữ liệu ngay khi người dùng nhập, giúp phát hiện lỗi kịp thời và nâng cao trải nghiệm người dùng, đồng thời giảm tải cho máy chủ.

**Kiểm tra dữ liệu phía server (Server-side Validation):** Tại các phương thức trong Controller, dữ liệu được kiểm tra lại bằng cách sử dụng thuộc tính ModelState.IsValid nhằm xác định tính hợp lệ của dữ liệu trước khi xử lý tiếp. Ngoài ra, hệ thống còn áp dụng các quy tắc kiểm tra nghiệp vụ riêng (custom validation) như:

- Ngày nhận phòng (CheckInDate) phải lớn hơn ngày hiện tại.
- Ngày trả phòng (CheckOutDate) phải lớn hơn ngày nhận phòng.
- Tổng tiền (TotalPrice) được tính dựa trên số đêm lưu trú và đơn giá phòng.

Việc kết hợp kiểm tra dữ liệu ở cả phía client và server giúp tăng cường độ tin cậy, đảm bảo an toàn và hạn chế lỗi dữ liệu trong hệ thống.

### 3.2.8 Triển khai cơ chế bảo mật (Security)

Hệ thống áp dụng nhiều cơ chế bảo mật nhằm đảm bảo an toàn dữ liệu và hạn chế các nguy cơ tấn công từ bên ngoài.

**Xác thực người dùng (Authentication):** Cơ chế xác thực dựa trên phiên làm việc (Session-based Authentication) được triển khai trong AdminController.

Tại phương thức Login, hệ thống kiểm tra tên đăng nhập và mật khẩu của quản trị viên. Nếu thông tin hợp lệ, hệ thống sẽ lưu thông tin người dùng vào biến phiên làm việc dưới dạng Session["AdminUsername"].

Ngoài ra, một phương thức hỗ trợ IsAuthenticated() được xây dựng để kiểm tra trạng thái đăng nhập của người dùng thông qua session.

Tất cả các chức năng quản trị đều kiểm tra quyền truy cập, đồng thời tự động chuyển hướng về trang đăng nhập nếu người dùng chưa được xác thực.

**Bảo vệ chống tấn công CSRF:** Hệ thống kích hoạt cơ chế chống tấn công Cross-Site Request Forgery (CSRF) bằng cách:

- Gắn thuộc tính [ValidateAntiForgeryToken] vào các phương thức xử lý yêu cầu POST.
- Chèn token bảo mật thông qua hàm @Html.AntiForgeryToken() trong các biểu mẫu (form).

- Cơ chế này giúp đảm bảo rằng các yêu cầu gửi tới máy chủ đều hợp lệ và được sinh ra từ hệ thống.

**Phòng chống tấn công SQL Injection:** Nhằm ngăn chặn lỗ hổng SQL Injection, hệ thống sử dụng các câu truy vấn có tham số (Parameterized Queries) với đối tượng SqlParameter cho tất cả các thao tác truy xuất dữ liệu. Việc không nối trực tiếp dữ liệu nhập từ người dùng vào chuỗi SQL góp phần giảm thiểu nguy cơ tấn công và đảm bảo an toàn cho cơ sở dữ liệu.

### 3.2.9 Kiểm thử và gỡ lỗi (Testing & Debugging)

Công tác kiểm thử và gỡ lỗi được thực hiện xuyên suốt quá trình phát triển nhằm đảm bảo hệ thống hoạt động ổn định và đúng yêu cầu đề ra.

**Kiểm thử đơn vị (Unit Testing):** Kiểm thử đơn vị được tiến hành đối với từng thành phần riêng lẻ của hệ thống, bao gồm:

- Kiểm tra các phương thức trong tầng DAL bằng dữ liệu mẫu.
- Xác minh các Controller trả về đúng View và dữ liệu tương ứng.
- Kiểm tra tính hợp lệ của các Model với các tập dữ liệu hợp lệ và không hợp lệ.

Mục tiêu của kiểm thử đơn vị là phát hiện sớm lỗi logic trong quá trình phát triển.

**Kiểm thử tích hợp (Integration Testing):** Kiểm thử tích hợp được thực hiện để đánh giá hoạt động của toàn bộ hệ thống khi kết hợp các thành phần với nhau, bao gồm:

- Kiểm tra luồng đặt phòng hoàn chỉnh từ bước chọn phòng đến khi xác nhận đặt phòng.
- Kiểm tra quy trình quản trị từ đăng nhập đến quản lý thông tin đặt phòng.

Loại kiểm thử này giúp đảm bảo các chức năng phối hợp đúng theo luồng nghiệp vụ.

**Kiểm thử trình duyệt (Browser Testing):** Hệ thống được kiểm tra trên nhiều trình duyệt khác nhau như **Google Chrome**, **Mozilla Firefox**, **Microsoft Edge**, đồng thời đánh giá khả năng hiển thị và tương thích trên thiết bị di động (responsive behavior).

**Gỡ lỗi (Debugging):** Quá trình gỡ lỗi được thực hiện bằng công cụ **Visual Studio Debugger** thông qua việc:

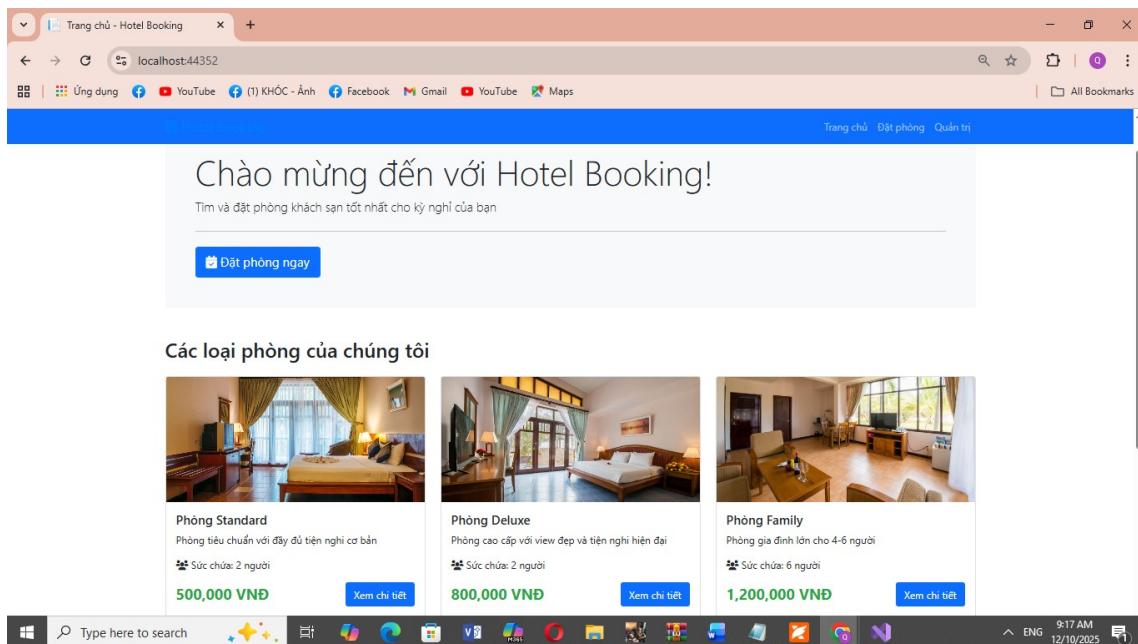
- Đặt điểm dừng (breakpoints),
- Theo dõi giá trị biến bằng Watch Window,
- Thực hiện chạy chương trình từng bước (step-through execution).

Việc áp dụng các kỹ thuật trên giúp phát hiện và khắc phục lỗi kịp thời trong quá trình phát triển.

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

### 4.1 Chức năng phía người dùng

#### 4.1.1 Xem danh sách loại phòng



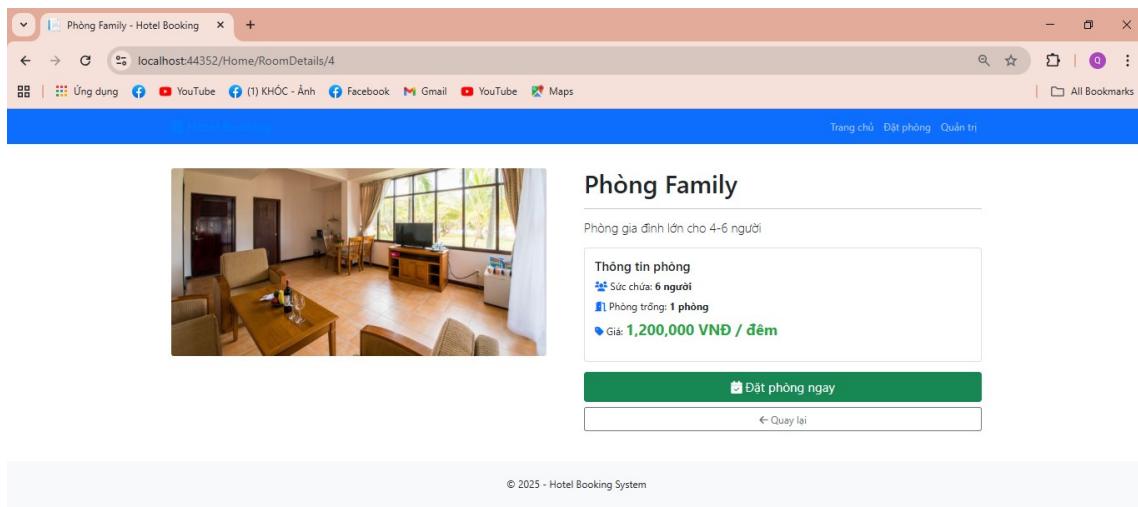
Hình 4. 1 Giao diện trang chủ

Trang chủ của hệ thống có chức năng hiển thị đầy đủ danh sách các loại phòng hiện có trong khách sạn dưới dạng giao diện thẻ (card layout) theo tiêu chuẩn Responsive Design.

Mỗi thẻ phòng bao gồm các thông tin chính như: Hình ảnh đại diện cho loại phòng; Tên loại phòng hiển thị rõ ràng; Mô tả ngắn gọn về tiện nghi và đặc điểm nổi bật; Thông tin sức chứa tối đa; Giá phòng theo đêm, được định dạng có phân cách hàng nghìn và đơn vị tiền tệ VNĐ; Nút chức năng “Xem chi tiết” cho phép người dùng chuyển đến trang thông tin chi tiết của từng loại phòng.

Hệ thống sử dụng Bootstrap Grid System để tổ chức bố cục hiển thị: Ba cột trên màn hình máy tính để bàn (desktop); Hai cột trên máy tính bảng (tablet); Một cột trên thiết bị di động (mobile). Trong trường hợp đường dẫn hình ảnh (**ImageUrl**) rỗng hoặc không tồn tại, hệ thống tự động hiển thị hình ảnh mặc định (placeholder) nhằm đảm bảo giao diện luôn đồng nhất.

#### 4.1.2 Xem chi tiết loại phòng



Hình 4. 2 Trang xem chi tiết loại phòng

Khi người dùng nhấp vào nút “Xem chi tiết” hoặc chọn trực tiếp vào thẻ (card) của một loại phòng, hệ thống sẽ tự động chuyển hướng đến trang RoomDetails với tham số roomTypeId trên thanh địa chỉ URL nhằm xác định loại phòng tương ứng.

- Giao diện trang được thiết kế theo bố cục hai cột (two-column layout): Cột bên trái hiển thị hình ảnh lớn đại diện cho loại phòng; Cột bên phải hiển thị các thông tin chi tiết liên quan.

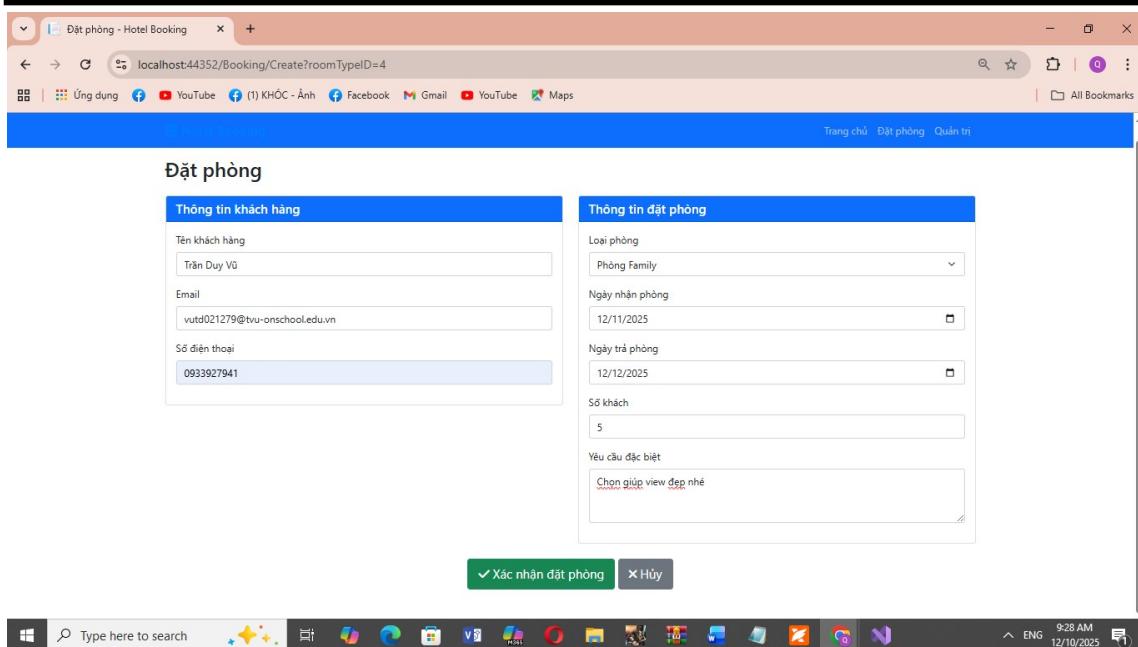
- Các thông tin được hiển thị bao gồm:

+ Tên loại phòng được đặt trong tiêu đề lớn (heading)

+ Mô tả chi tiết về đặc điểm và tiện nghi của phòng

+ Thẻ thông tin (information card) hiển thị các thông số: Sức chứa tối đa (kèm biểu tượng người dùng); Số lượng phòng trống hiện tại (kèm biểu tượng cửa) – dữ liệu được truy vấn trực tiếp từ cơ sở dữ liệu theo thời gian thực; Giá phòng mỗi đêm được làm nổi bật bằng màu sắc và kích thước chữ lớn để dễ quan sát.

#### 4.1.3 Thực hiện đặt phòng



Hình 4. 3 Trang đặt phòng

Trang đặt phòng có thể được truy cập thông qua thanh điều hướng hoặc trực tiếp từ trang chi tiết phòng. Biểu mẫu đặt phòng được thiết kế theo bố cục hai cột (two-column grid) với các khung thẻ (card containers) nhằm phân tách nội dung một cách rõ ràng.

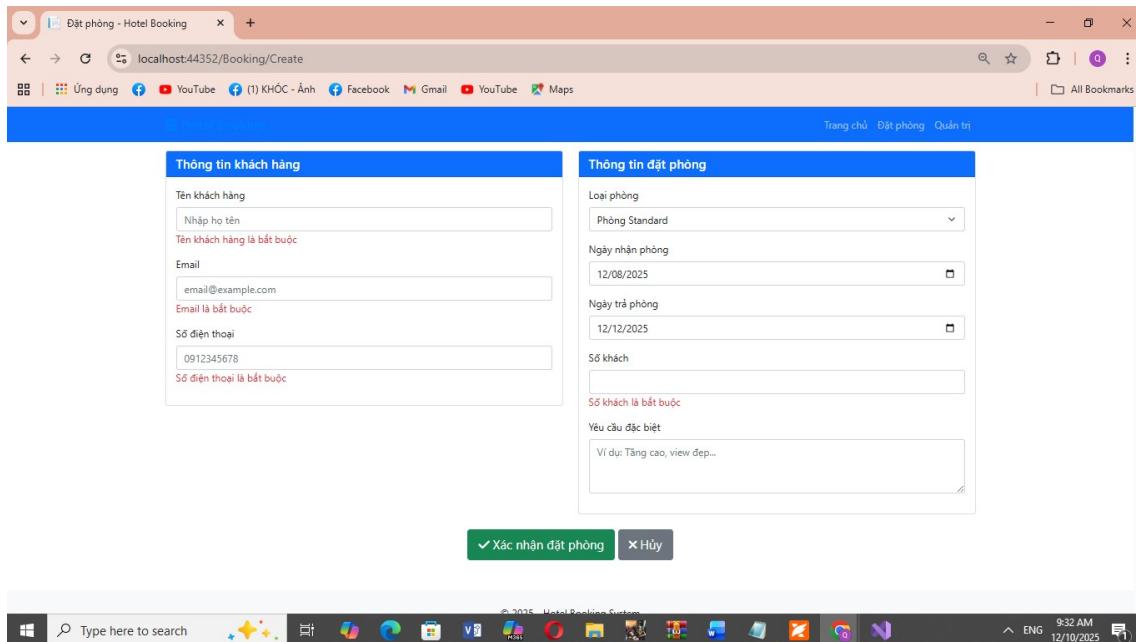
**Khu vực Thông tin khách hàng:** Cột bên trái của giao diện hiển thị phần “Thông tin khách hàng”, bao gồm các trường dữ liệu sau:

- Tên khách hàng: Sử dụng TextBoxFor kết hợp với cơ chế kiểm tra dữ liệu (validation)
- Email: Kiểu dữ liệu email, áp dụng ràng buộc [EmailAddress].
- Số điện thoại: Áp dụng ràng buộc [Phone] trong Model.

**Khu vực Thông tin đặt phòng:** Cột bên phải hiển thị phần “Thông tin đặt phòng”, bao gồm:

- Loại phòng: Danh sách thả xuống (dropdown list) được nạp động qua ViewBag.RoomTypes. Trong trường hợp người dùng truy cập từ trang chi tiết phòng, loại phòng sẽ được tự động chọn sẵn.
- Ngày nhận phòng (CheckInDate): Kiểu date, giá trị mặc định là ngày kế tiếp.

- Ngày trả phòng (CheckOutDate): Kiểu date, được kiểm tra phải sau ngày nhận phòng.
- Số lượng khách (NumberOfGuests): Kiểu number, giá trị tối thiểu là 1.
- Yêu cầu đặc biệt: Trường nhập liệu dạng textarea, cho phép nhập các yêu cầu thêm như tầng cao, hướng nhìn đẹp hoặc loại giường.



Hình 4. 4 Kiểm tra dữ liệu trang đặt phòng

**Kiểm tra dữ liệu và xử lý biểu mẫu:** Cơ chế kiểm tra dữ liệu phía client sử dụng thư viện jQuery Validation, giúp hiển thị thông báo lỗi ngay khi người dùng bỏ focus (blur) khỏi trường dữ liệu. Các trường bắt buộc được đánh dấu bằng dấu sao đỏ (\*), giúp người dùng dễ nhận biết.

Biểu mẫu bao gồm:

- Nút “Xác nhận đặt phòng” (màu xanh lá, kèm biểu tượng xác nhận),
- Nút “Hủy” (màu xám) chuyển hướng về trang chủ.

Token chống giả mạo yêu cầu (Anti-forgery token) được tích hợp để phòng chống tấn công CSRF.

Ở phía máy chủ, phương thức POST trong Controller thực hiện:

- Kiểm tra ModelState.IsValid

- Áp dụng các quy tắc nghiệp vụ: Ngày nhận phòng không được nằm trong quá khứ, Ngày trả phòng phải sau ngày nhận phòng ít nhất một ngày.

- Tự động tính tổng tiền (TotalPrice) dựa trên giá phòng và số đêm lưu trú.

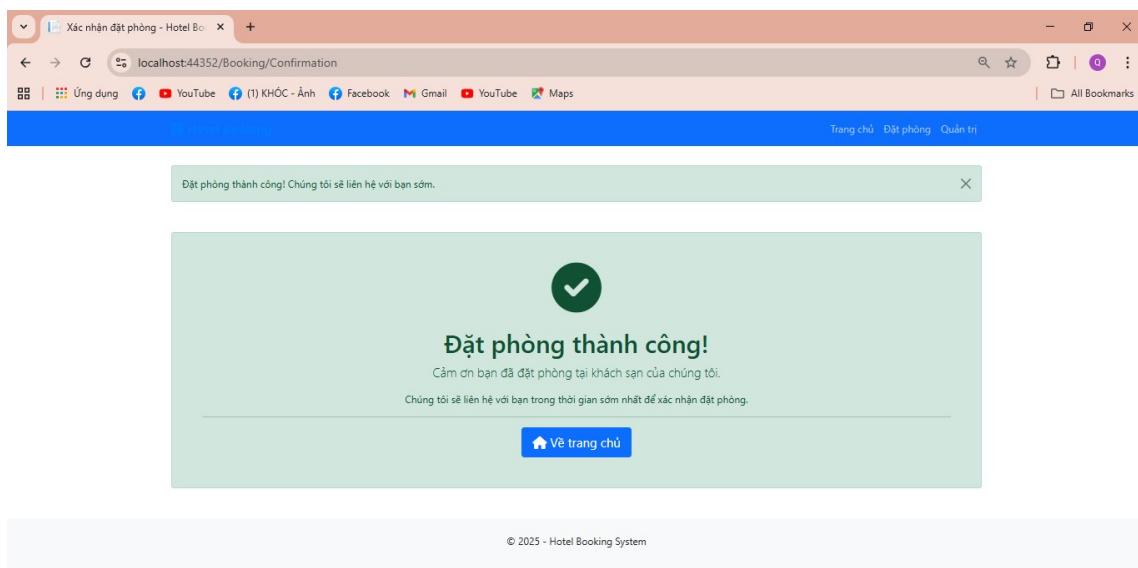
- Thiết lập trạng thái đặt phòng mặc định là Pending.

Sau khi dữ liệu hợp lệ, phương thức BookingDAL.Insert() được gọi để lưu thông tin vào cơ sở dữ liệu.

Nếu thao tác thành công, hệ thống thiết lập thông báo thông qua TempData và chuyển người dùng đến trang xác nhận (Confirmation Page).

Trong trường hợp xảy ra lỗi, người dùng sẽ được giữ lại biểu mẫu kèm thông báo chi tiết.

#### 4.1.4 Nhận xác nhận đặt phòng



Hình 4.5 Trang xác nhận đặt phòng

Trang Confirmation được hiển thị sau khi người dùng hoàn tất quá trình đặt phòng thành công. Giao diện được thiết kế theo bố cục căn giữa (centered layout) nhằm thu hút sự chú ý và làm nổi bật trạng thái hoàn tất giao dịch.

Tiêu đề “Đặt phòng thành công!” có vai trò xác nhận rõ ràng rằng thao tác của người dùng đã được thực hiện thành công.

Nội dung thông báo cho biết khách sạn sẽ liên hệ với khách hàng trong thời gian

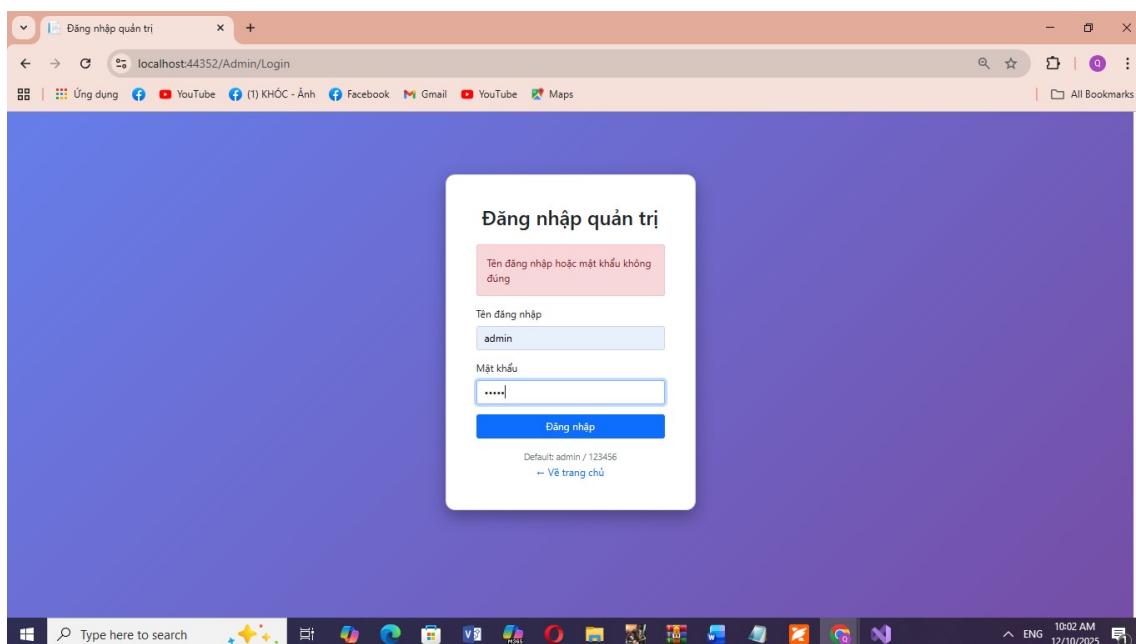
sớm nhất để xác nhận thông tin đặt phòng.

Ngoài ra, thông báo thành công được lưu trong TempData.Success và hiển thị tại phía trên trang dưới dạng cảnh báo có thể đóng (dismissible alert).

Mặc dù được thiết kế đơn giản, trang xác nhận có vai trò quan trọng trong việc khép lại quy trình đặt phòng, giúp người dùng nhận biết trạng thái xử lý và các bước tiếp theo.

## 4.2 Chức năng phía quản trị viên

### 4.2.1 Đăng nhập hệ thống



Hình 4. 6 Trang đăng nhập hệ thống

Trang đăng nhập dành cho quản trị viên được thiết kế với phong cách đơn giản nhưng chuyên nghiệp, sử dụng bố cục căn giữa (centered card layout). Khối đăng nhập (card) được thiết kế có bo góc (border-radius) và hiệu ứng đổ bóng (box-shadow) nhằm tạo chiều sâu cho giao diện. Tiêu đề “Đăng nhập quản trị” được hiển thị nổi bật nhằm tăng khả năng nhận diện.

Biểu mẫu đăng nhập bao gồm:

- Trường tên đăng nhập (Username) với ràng buộc bắt buộc nhập dữ liệu,
- Trường mật khẩu (Password) với kiểu password để che nội dung nhập,

- Nút “Đăng nhập” hiển thị toàn chiều ngang, sử dụng màu sắc chủ đạo của hệ thống.

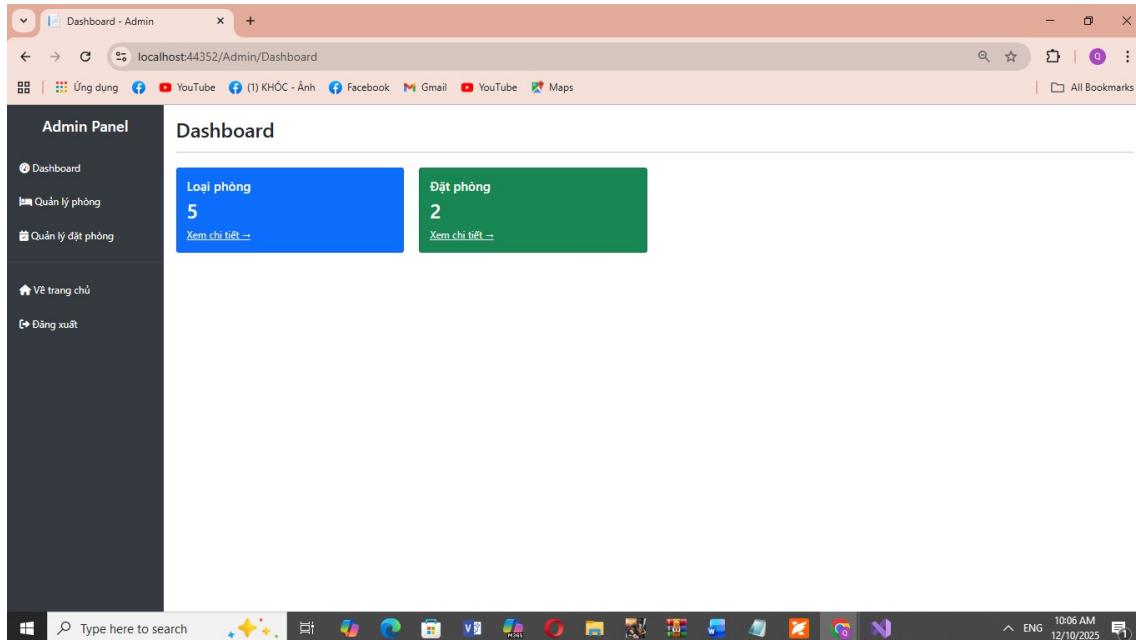
Trong trường hợp thông tin đăng nhập không hợp lệ, hệ thống hiển thị thông báo lỗi dưới dạng cảnh báo (alert danger). Ngoài ra, liên kết “Về trang chủ” cho phép người dùng quay lại giao diện công khai của hệ thống. Một ghi chú nhỏ cung cấp tài khoản mặc định nhằm phục vụ mục đích kiểm thử.

Ở phía xử lý nghiệp vụ, phương thức POST của Controller kiểm tra thông tin đăng nhập với dữ liệu trong cơ sở dữ liệu. Trong phiên bản thử nghiệm, hệ thống sử dụng thông tin định sẵn (**ví dụ: admin/123456**). Đối với môi trường triển khai thực tế, mật khẩu cần được mã hóa bằng các thuật toán băm như BCrypt hoặc tương đương để đảm bảo an toàn.

Nếu thông tin hợp lệ, hệ thống thiết lập Session["AdminUsername"] và chuyển hướng đến trang Dashboard.

Trường hợp không hợp lệ, thông báo lỗi được gán vào ViewBag.Error và hiển thị lại biểu mẫu đăng nhập.

#### 4.2.2 Bảng điều khiển quản trị (Admin Dashboard)



Hình 4. 7 Trang Admin Dassboard

Admin Dashboard là trang đầu tiên được hiển thị sau khi quản trị viên đăng nhập thành công. Phần tiêu đề của trang hiển thị thông điệp chào mừng kèm theo tên tài khoản quản trị viên được lấy từ biến phiên làm việc (session).

Khu vực thống kê được trình bày dưới dạng các thẻ thông tin (statistics cards) sắp xếp theo hàng (row) thông qua hệ thống lưới của Bootstrap. Mỗi thẻ có màu nền và biểu tượng minh họa khác nhau nhằm giúp phân biệt các loại thông tin:

- Thẻ “Loại phòng” (màu xanh dương): Hiển thị tổng số loại phòng hiện có, kèm biểu tượng giường và liên kết “Xem chi tiết” dẫn đến trang quản lý loại phòng.
- Thẻ “Đặt phòng” (màu xanh lá): Hiển thị tổng số lượt đặt phòng trong hệ thống, kèm biểu tượng lịch và liên kết chuyển đến trang quản lý đặt phòng.

#### 4.2.3 Quản lý loại phòng

ID	Tên loại phòng	Giá (VNĐ)	Sức chứa	Thao tác
1	Phòng Standard	500.000	2 người	<input checked="" type="button"/> Sửa <input type="button"/> Xóa
2	Phòng Deluxe	800.000	2 người	<input checked="" type="button"/> Sửa <input type="button"/> Xóa
4	Phòng Family	1.200.000	6 người	<input checked="" type="button"/> Sửa <input type="button"/> Xóa
3	Phòng Suite	1.500.000	4 người	<input checked="" type="button"/> Sửa <input type="button"/> Xóa
5	Phòng VIP	3.000.000	2 người	<input checked="" type="button"/> Sửa <input type="button"/> Xóa

Hình 4. 8 Trang quản lý loại phòng

Trang quản lý loại phòng hiển thị danh sách toàn bộ các loại phòng trong hệ thống dưới dạng bảng (table). Bảng dữ liệu bao gồm các cột sau: Mã loại phòng (RoomTypeID); Tên loại phòng; Giá phòng (được định dạng có phân cách hàng nghìn); Sức chứa (hiển thị kèm đơn vị “người”); Cột thao tác.

Trong cột thao tác, hệ thống cung cấp các nút chức năng: Sửa (Edit): Nút màu vàng kèm biểu tượng chỉnh sửa; Xóa (Delete): Nút màu đỏ kèm biểu tượng thùng rác;

Nút “Thêm loại phòng” được bố trí ở góc trên bên phải của tiêu đề trang, cho phép quản trị viên chuyển đến trang tạo mới loại phòng.

Danh sách loại phòng được tải từ cơ sở dữ liệu thông qua phương thức RoomTypeDAL.GetAllRoomTypes() và hiển thị bằng vòng lặp foreach trong View.

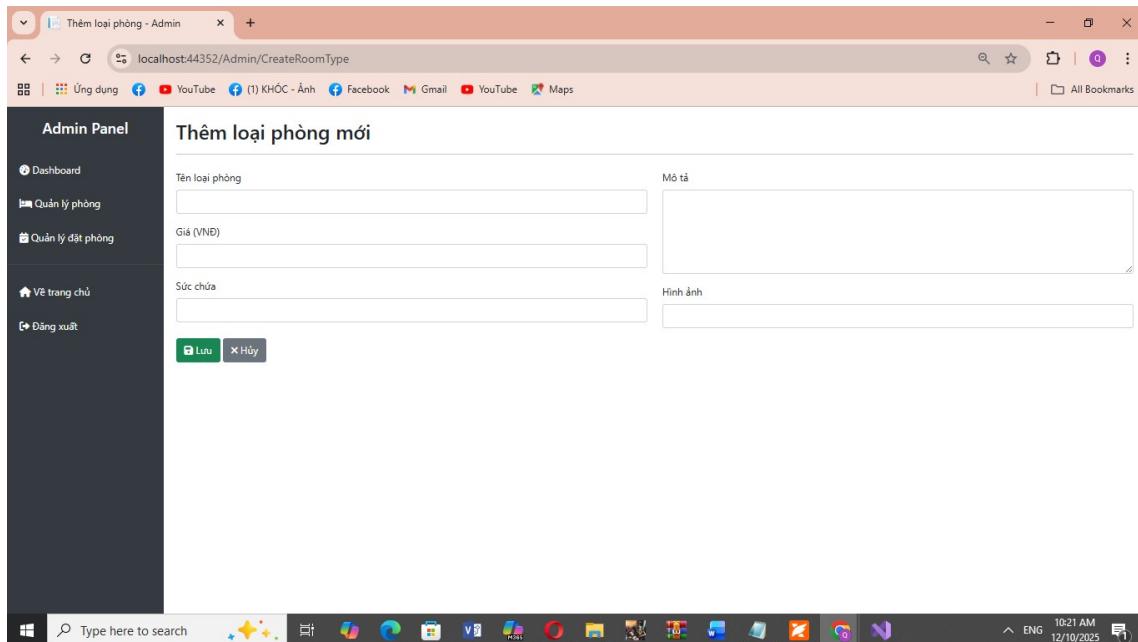
Nút Sửa chuyển đến phương thức EditRoomType với tham số roomTypeId.

Nút Xóa gọi đến phương thức DeleteRoomType và hiển thị hộp thoại xác nhận với thông báo: “Bạn có chắc muốn xóa?”.

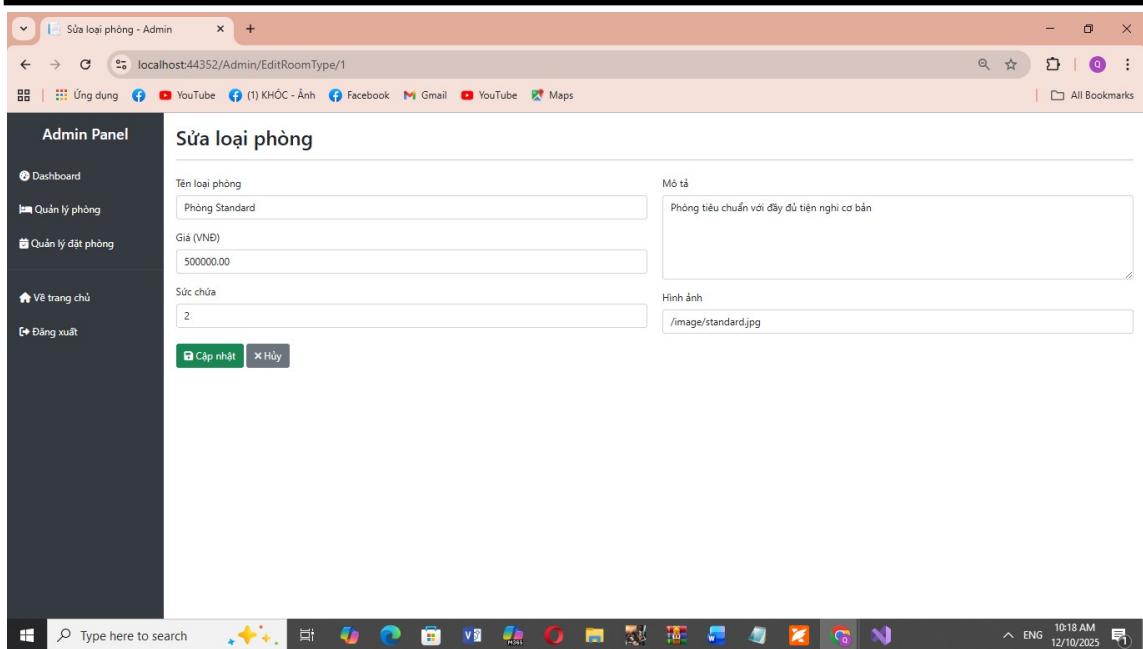
Khi thao tác thành công, hệ thống hiển thị thông báo thông qua TempData ở đầu trang dưới dạng cảnh báo màu xanh lá.

Trong trường hợp xảy ra lỗi, thông báo hiển thị dưới dạng cảnh báo màu đỏ.

#### 4.2.4 Thêm và sửa loại phòng (CreateRoomType và EditRoomType)



Hình 4. 9 Trang thêm loại phòng



Hình 4. 10 Trang sửa loại phòng

Hai trang CreateRoomType và EditRoomType được thiết kế với bố cục biểu mẫu tương đồng nhằm đảm bảo tính nhất quán trong giao diện người dùng. Giao diện biểu mẫu được chia thành hai cột:

### Nội dung biểu mẫu

Cột bên trái bao gồm các trường dữ liệu chính:

- Tên loại phòng: Trường nhập văn bản, có ràng buộc bắt buộc.
- Giá phòng: Trường nhập số thập phân, có ràng buộc bắt buộc.
- Sức chứa: Trường nhập số nguyên, có ràng buộc bắt buộc.

Cột bên phải bao gồm:

- Mô tả: Trường văn bản nhiều dòng (textarea) dùng để nhập thông tin chi tiết.
- Đường dẫn hình ảnh: Trường nhập văn bản cho đường dẫn đến hình ảnh đại diện.

### Chức năng thêm và cập nhật

Trang CreateRoomType cung cấp:

- Nút “Lưu” (màu xanh lá, có biểu tượng lưu),
- Nút “Hủy” (màu xám, quay về danh sách loại phòng).

Trang EditRoomType có giao diện tương tự, tuy nhiên các trường dữ liệu đã được tự động điền (pre-filled) dựa trên thông tin của đối tượng RoomType hiện tại.

Trường ẩn RoomTypeID được sử dụng để xác định bản ghi cần cập nhật.

**Kiểm tra dữ liệu và xử lý phía server:** Các trường dữ liệu được kiểm tra thông qua ValidationMessageFor và Data Annotations.

Khi người dùng gửi biểu mẫu:

- Controller kiểm tra ModelState.IsValid.

- Nếu hợp lệ, hệ thống gọi:

- + RoomTypeDAL.Insert() đối với chức năng thêm mới,

- + RoomTypeDAL.Update() đối với chức năng cập nhật.

- Sau khi xử lý thành công, hệ thống chuyển hướng về trang danh sách loại phòng kèm thông báo TempData.

- Nếu xảy ra lỗi, biểu mẫu được hiển thị lại kèm thông báo lỗi tương ứng.

Cơ chế ngăn chặn gửi biểu mẫu khi dữ liệu không hợp lệ được hỗ trợ bằng JavaScript. Thủ Anti-forgery token được tích hợp để phòng chống tấn công CSRF.

#### 4.2.5 Quản lý đặt phòng

ID	Khách hàng	Loại phòng	Check-in	Check-out	Tổng tiền	Trạng thái	Thao tác
2	Trần Duy Vũ 0933927941	Phòng Family	11/12/2025	12/12/2025	1,200,000 VNĐ	Chờ xác nhận	
1	Duy vũ 0933927941	Phòng Standard	03/12/2025	04/12/2025	500,000 VNĐ	Đã xác nhận	

Hình 4. 11 Trang quản lý đặt phòng

Trang Quản lý đặt phòng hiển thị bảng danh sách đầy đủ các lượt đặt phòng trong hệ thống, cung cấp cho quản trị viên cái nhìn tổng thể và khả năng thao tác thuận tiện.

### Giao diện bảng danh sách:

Bảng dữ liệu hiển thị các cột thông tin sau:

- BookingID: Mã định danh của lượt đặt phòng.
- Khách hàng: Tên khách hàng hiển thị chính, bên dưới là số điện thoại với kiểu chữ mờ (muted text).
- Loại phòng: Tên loại phòng khách sạn.
- Ngày nhận phòng: Định dạng dd/MM/yyyy.
- Ngày trả phòng: Định dạng dd/MM/yyyy.
- Tổng tiền: Được định dạng có phân tách hàng nghìn và đơn vị VND.

Cột Trạng thái hiển thị nhãn (badge) được mã hóa màu theo từng trạng thái:

- Chờ xác nhận (Pending) — màu vàng,
- Đã xác nhận (Confirmed) — màu xanh lá,
- Đã hủy (Cancelled) — màu đỏ,
- Hoàn tất (Completed) — màu xám.

Cột Thao tác bao gồm một nhóm nút chức năng:

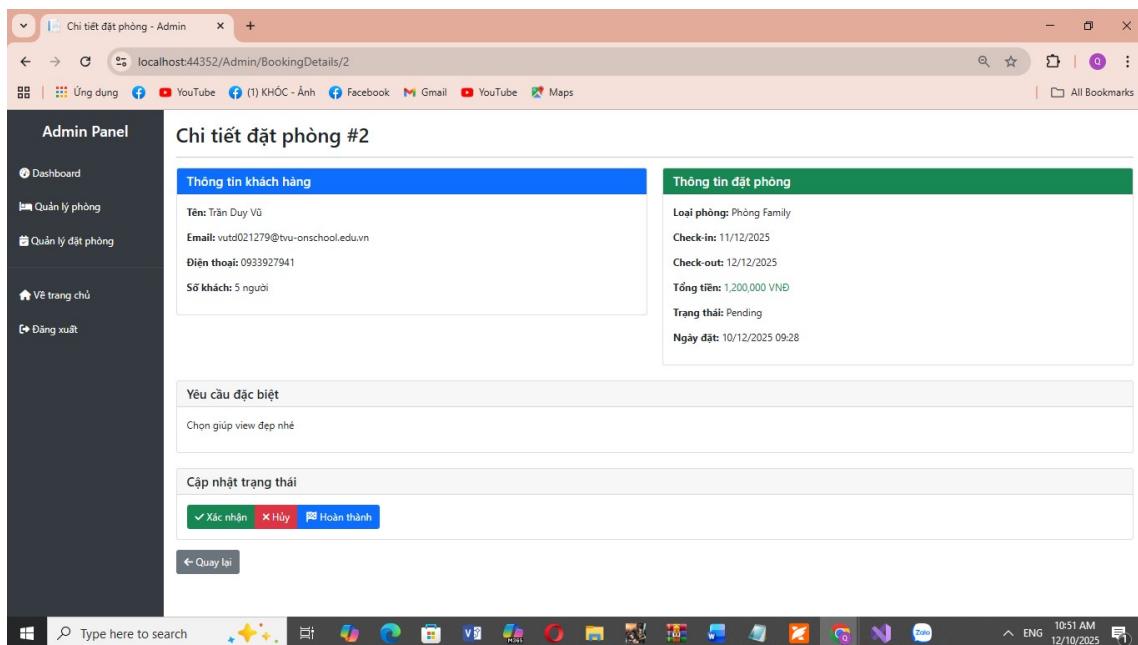
- Nút Xem chi tiết (màu xanh lam nhạt, biểu tượng mắt) liên kết tới trang BookingDetails,
- Nút Xóa (màu đỏ, biểu tượng thùng rác) kèm hộp thoại xác nhận trước khi thực hiện.

### Xử lý dữ liệu:

Dữ liệu được tải thông qua phương thức BookingDAL.GetAllBookings().

Phương thức này thực hiện truy vấn JOIN với bảng RoomTypes để hiển thị tên loại phòng tương ứng.

#### 4.2.6 Xem và cập nhật chi tiết đặt phòng



Hình 4. 12 Trang chi tiết đặt phòng

Trang Chi tiết đặt phòng hiển thị đầy đủ thông tin chi tiết của một lượn đặt phòng dưới dạng bố cục thẻ (card layout) khoa học, rõ ràng và dễ theo dõi.

#### Bố cục giao diện:

Giao diện được thiết kế theo bố cục hai cột:

- Cột bên trái – Thông tin khách hàng: Hiển thị trong một card có nền màu chủ đạo, bao gồm: Họ và tên khách hàng, Địa chỉ email, Số điện thoại liên hệ, Số lượng khách lưu trú.

- Cột bên phải – Thông tin đặt phòng: Hiển thị trong card nền màu thành công (success), bao gồm: Loại phòng đã chọn, Ngày nhận phòng, Ngày trả phòng, Tổng chi phí (được làm nổi bật bằng màu xanh lá), Trạng thái hiện tại của đơn đặt phòng, Ngày đặt phòng kèm dấu thời gian (timestamp).

Nếu trường Yêu cầu đặc biệt (SpecialRequests) có dữ liệu, hệ thống sẽ hiển thị thêm một thẻ thông tin riêng với tiêu đề "Yêu cầu đặc biệt" và nội dung chi tiết yêu cầu của khách hàng.

#### Chức năng cập nhật trạng thái:

Phía cuối trang có thẻ "Cập nhật trạng thái", cho phép quản trị viên thay đổi trạng thái đơn đặt phòng thông qua các nút chức năng:

- Xác nhận (màu xanh lá) → cập nhật trạng thái Confirmed,
- Hủy (màu đỏ) → cập nhật trạng thái Cancelled,
- Hoàn thành (màu xanh dương) → cập nhật trạng thái Completed.

Biểu mẫu này gửi dữ liệu đến action UpdateBookingStatus của Controller, kèm theo:

bookingId: mã định danh đơn đặt phòng, newStatus: trạng thái mới cần cập nhật.

Controller thực hiện: Kiểm tra dữ liệu hợp lệ, Gọi phương thức BookingDAL.UpdateStatus() để cập nhật trong cơ sở dữ liệu, Lưu thông báo kết quả vào TempData, Điều hướng người dùng quay lại trang danh sách đặt phòng (Bookings).

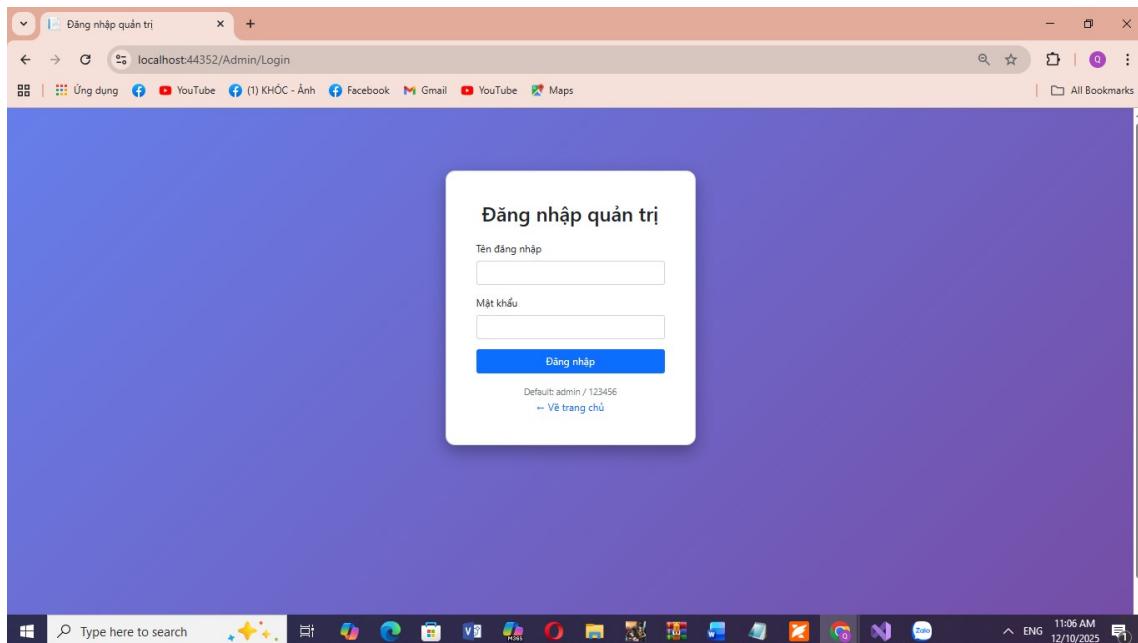
#### Ràng buộc dữ liệu và hiển thị:

Tất cả thông tin trên trang được hiển thị thông qua Strongly-Typed Model Binding, sử dụng các thuộc tính của @Model, đảm bảo:

- Tính chính xác của dữ liệu.
- Tránh lỗi sai khi truyền tham số
- Dễ mở rộng và bảo trì mã nguồn.

Nút "**Quay lại**" được bố trí cuối trang giúp người quản trị dễ dàng quay về danh sách đặt phòng.

#### 4.2.7 Đăng xuất



Hình 4. 13 Trả lại trang quản trị sau khi đăng xuất

Chức năng đăng xuất được triển khai trong AdminController thông qua action Logout(). Khi quản trị viên thực hiện thao tác đăng xuất, hệ thống sẽ gọi phương thức Session.Clear() để xóa toàn bộ dữ liệu phiên làm việc (session), sau đó chuyển hướng người dùng về trang đăng nhập quản trị.

Liên kết đăng xuất được bố trí trong thanh điều hướng bên (sidebar navigation) của giao diện quản trị, cho phép quản trị viên thực hiện thao tác đăng xuất từ bất kỳ trang nào trong khu vực quản trị.

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Website đặt phòng khách sạn đã được phát triển với đầy đủ các chức năng cốt lõi. Ứng dụng thể hiện sự hiểu biết về kiến trúc ASP.NET MVC, các nguyên tắc thiết kế cơ sở dữ liệu và các thực tiễn tốt trong phát triển web. Chất lượng mã nguồn được đảm bảo với việc phân tách rõ ràng các tầng, sử dụng lại các thành phần hợp lý và cấu trúc dễ bảo trì.

Dự án là một nền tảng tốt để tiếp tục mở rộng với các tính năng bổ sung, hướng tới triển khai trong môi trường thực tế. Kết quả học tập đạt được bao gồm kinh nghiệm thực hành trong phát triển web, thiết kế và triển khai cơ sở dữ liệu, các vấn đề về bảo mật, thiết kế giao diện người dùng, cũng như các nguyên tắc trong kỹ thuật phần mềm. Những kỹ năng này có thể áp dụng cho nhiều dự án phát triển ứng dụng web khác trong tương lai.

Hướng phát triển tiếp theo bao gồm hoàn thiện các tính năng nâng cao còn lại, thực hiện kiểm thử toàn diện (bao gồm kiểm thử bảo mật), tối ưu hiệu năng bằng các công cụ phân tích, triển khai hệ thống lên máy chủ thực tế với cấu hình phù hợp, kiểm thử chấp nhận người dùng và cải tiến liên tục dựa trên phản hồi.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] Tài liệu giảng dạy môn Chuyên đề ASP.net. Tài liệu giảng viên cung cấp
- [2] Ths. Nguyễn Minh Đạo, Lập trình Web với ASP.NET, Nhà xuất bản ĐHQG HCM, 2014.
- [3] Hướng dẫn ASP.NET, <https://dotnet.microsoft.com/en-us/>, ngày truy cập: 22/11/2025.
- [4] Ngôn ngữ c#, [https://vi.wikipedia.org/wiki/C\\_Sharp\\_\(ngôn\\_ngữ\\_lập\\_trình\)](https://vi.wikipedia.org/wiki/C_Sharp_(ngôn_ngữ_lập_trình)), ngày truy cập: 22/11/2025.
- [5] Tài liệu ASP.NET, <http://learn.microsoft.com/vi-vn/aspnet/>, Ngày truy cập: 23/11/2025.
- [6] Cơ sở dữ liệu SQL server, <https://aws.amazon.com/vi/what-is/sql-database/>, ngày truy cập: 20/11/2025.
- [7] Chuẩn hóa dữ liệu, <https://datapot.vn/chuan-hoa-du-lieu-va-phi-chuan-hoa-du-lieu/>, ngày truy cập: 20/11/2025.
- [8] CSRF Protection, <https://learn.microsoft.com/en-us/dotnet/api/system.web.mvc.htmlhelper.antiforgerytoken?view=aspnet-mvc-5.2>, ngày truy cập 25/11/2025.