

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**ĐẠI HỌC PHENIKAA**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN PHENIKAA**

--o0o--



**THIẾT KẾ WEB NÂNG CAO**  
**BÀI TẬP LỚN**

**ĐỀ TÀI**  
**THIẾT KẾ HỆ THỐNG ĐẶT SÂN BÓNG ĐÁ MINI**  
**SỬ DỤNG LARAVEL**

**Giảng viên:**

**Nguyễn Thị Thùy Liên**

**Sinh viên thực hiện:**

**Nguyễn Văn Lượng**

**Lý Hoài Nam**

**Lê Hồng Phúc**

*Hà Nội, 20/06/2025*

## LỜI NÓI ĐẦU

Trong thời đại công nghệ 4.0, sự phát triển mạnh mẽ của công nghệ thông tin đã và đang làm thay đổi toàn diện mọi mặt của đời sống xã hội. Các hoạt động truyền thống dần được số hóa, tự động hóa nhằm nâng cao hiệu quả vận hành, tiết kiệm thời gian và tối ưu chi phí. Đặc biệt, lĩnh vực quản lý dịch vụ đang có sự chuyển dịch mạnh mẽ từ mô hình thủ công sang các mô hình quản lý trực tuyến, thông minh và tiện lợi hơn.

Trong bối cảnh đó, nhu cầu ứng dụng công nghệ vào lĩnh vực thể dục thể thao, đặc biệt là quản lý sân bóng đá mini, ngày càng trở nên cần thiết. Hiện nay, phần lớn các sân bóng vẫn đang áp dụng hình thức quản lý truyền thống như ghi chép bằng giấy, gọi điện thoại để đặt sân hoặc lưu trữ thủ công bằng bảng tính Excel. Phương pháp này bộc lộ nhiều hạn chế như dễ xảy ra sai sót, trùng lịch, mất dữ liệu, khó theo dõi doanh thu, đồng thời gây mất thời gian cho cả người quản lý lẫn khách hàng.

Xuất phát từ thực tế đó, nhóm chúng em đã lựa chọn đề tài “Thiết kế và triển khai hệ thống quản lý đặt sân bóng đá mini sử dụng Laravel” với mong muốn xây dựng một hệ thống quản lý đặt sân trực tuyến hiện đại, tối ưu hóa quy trình vận hành, hỗ trợ chủ sân quản lý lịch đặt, khách hàng, thông tin sân một cách hiệu quả, đồng thời mang lại trải nghiệm thuận tiện cho người dùng khi tra cứu và đặt sân.

Đề tài không chỉ giúp chúng em củng cố và vận dụng những kiến thức đã học trong môn Thiết kế Web nâng cao, mà còn rèn luyện khả năng phân tích bài toán thực tế, thiết kế cơ sở dữ liệu, lập trình backend, frontend, quản lý dự án và triển khai ứng dụng hoàn chỉnh. Trong quá trình thực hiện, nhóm đã nghiên cứu và áp dụng thành thạo các công nghệ như framework Laravel, Blade Template, cơ sở dữ liệu MySQL, thư viện Bootstrap và quy trình thiết kế phần mềm theo mô hình MVC.

Quá trình thực hiện đề tài cũng là một quá trình học hỏi liên tục. Bên cạnh những thuận lợi khi được tiếp cận với các tài liệu hướng dẫn chính thống và sự hỗ trợ từ cộng đồng lập trình, nhóm cũng gặp không ít khó khăn liên quan đến lỗi hệ thống, xử lý logic phức tạp, tối ưu giao diện người dùng và triển khai cơ sở dữ liệu. Tuy nhiên, nhờ sự hỗ trợ nhiệt tình, tận tâm của thầy [Tên Giảng Viên], cùng với sự nỗ lực, tinh thần học hỏi và tinh thần làm việc nhóm nghiêm túc, chúng em đã hoàn thành đề tài đúng kế hoạch đề ra.

Báo cáo này trình bày toàn bộ quá trình nghiên cứu và triển khai đề tài, bao gồm các nội dung: giới thiệu bài toán thực tế, phân tích yêu cầu, thiết kế hệ thống, xây dựng phần mềm, đánh giá kết quả và đề xuất hướng phát triển trong tương lai. Chúng em hi vọng báo cáo sẽ phần nào phản ánh đầy đủ quá trình học tập, nghiên cứu và cống hiến của nhóm trong suốt học kỳ.

Mặc dù đã cố gắng hoàn thiện đề tài và báo cáo với tinh thần trách nhiệm cao nhất, nhưng do hạn chế về thời gian, kiến thức và kinh nghiệm thực tế, nhóm không thể tránh khỏi những thiếu sót nhất định. Chúng em rất mong nhận được những ý kiến đóng góp quý báu từ thầy cô và các bạn để cải thiện và hoàn thiện đề tài tốt hơn trong tương lai.

Cuối cùng, nhóm xin trân trọng gửi lời cảm ơn sâu sắc tới cô Nguyễn Thị Thùy Liên đã hỗ trợ, giúp đỡ nhóm trong quá trình thực hiện đề tài này.

Trân trọng!

NHÓM TÁC GIẢ

## MỤC LỤC

1. GIỚI THIỆU .....	6
1.1. Lý do chọn đề tài.....	6
1.2. Mục tiêu của đề tài.....	6
1.3. Phạm vi và đối tượng áp dụng.....	7
1.4. Phương pháp thực hiện .....	7
1.5. Cấu trúc báo cáo .....	7
2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG.....	8
2.1. Giới thiệu về phát triển web theo mô hình MVC .....	8
2.2. Laravel Framework.....	8
2.3. Blade Template Engine.....	9
2.4. MySQL Database .....	9
2.5. Bootstrap Framework .....	10
2.6. Các công cụ và công nghệ hỗ trợ.....	13
3. PHÂN TÍCH HỆ THỐNG .....	15
3.1. Phân tích yêu cầu hệ thống .....	15
3.2. Sơ đồ Use-Case .....	16
3.3. Thiết kế cơ sở dữ liệu (ERD) .....	17
3.4. Phân tích luồng hoạt động chính.....	18
4. Thiết kế và triển khai hệ thống.....	19
4.1. Thiết kế kiến trúc hệ thống.....	19
4.2. Thiết kế cơ sở dữ liệu chi tiết.....	19
4.3. Thiết kế giao diện người dùng .....	20
4.4. Triển khai Backend.....	21
4.5. Triển khai Frontend.....	24
5. Kết quả thực nghiệm và đánh giá.....	25
5.1. Kết quả triển khai.....	25

5.2.	<b>Kiểm thử chức năng .....</b>	26
5.3.	<b>Đánh giá hiệu suất .....</b>	26
5.4.	<b>Ưu điểm và hạn chế .....</b>	26
5.5.	<b>Bài học và kinh nghiệm .....</b>	26
6.	<b>Kết luận và hướng phát triển.....</b>	27
6.1.	<b>Kết luận.....</b>	27
6.2.	<b>Hạn chế của hệ thống .....</b>	27
6.3.	<b>Hướng phát triển trong tương lai.....</b>	27
7.	<b>Phân công vai trò trong nhóm .....</b>	28
8.	<b>Tài liệu tham khảo.....</b>	28

# 1. GIỚI THIỆU

## 1.1. Lý do chọn đề tài.

Bóng đá là môn thể thao vua, không chỉ phổ biến trên thế giới mà còn rất được yêu thích tại Việt Nam. Cùng với sự phát triển của đời sống xã hội, nhu cầu tham gia thể thao, đặc biệt là bóng đá phong trào, ngày càng tăng cao. Các sân bóng đá mini nhân tạo mọc lên ngày càng nhiều để đáp ứng nhu cầu này.

Tuy nhiên, vấn đề lớn hiện nay mà nhiều sân bóng gặp phải là việc quản lý lịch đặt sân còn mang tính thủ công, thiếu hệ thống. Chủ sân thường sử dụng các phương pháp truyền thống như ghi chép giấy, file Excel hoặc liên lạc qua điện thoại để quản lý việc đặt sân. Điều này dẫn đến nhiều hạn chế:

- Dễ xảy ra trùng lịch, sai sót hoặc nhầm lẫn.
- Khó theo dõi thông tin khách hàng và lịch sử đặt sân.
- Không kiểm soát hiệu quả doanh thu và công suất sử dụng sân.
- Thiếu sự chuyên nghiệp, gây bất tiện cho cả người quản lý lẫn khách hàng.

Trong bối cảnh công nghệ thông tin phát triển mạnh mẽ, việc xây dựng một hệ thống quản lý đặt sân trực tuyến là vô cùng cần thiết. Hệ thống này sẽ giúp tự động hóa quy trình đặt sân, tiết kiệm thời gian, giảm thiểu rủi ro và nâng cao trải nghiệm người dùng.

Nhận thấy nhu cầu thực tế cũng như mong muốn áp dụng kiến thức lập trình web vào giải quyết bài toán đời sống, nhóm chúng em quyết định thực hiện đề tài “Thiết kế và triển khai hệ thống quản lý đặt sân bóng đá mini sử dụng Laravel”.

Laravel – một framework PHP mạnh mẽ, bảo mật cao và dễ mở rộng – là nền tảng phù hợp để xây dựng hệ thống này. Cùng với sự hỗ trợ của giao diện Blade Template và cơ sở dữ liệu MySQL, hệ thống hứa hẹn mang lại hiệu quả cao trong việc quản lý và vận hành.

## 1.2. Mục tiêu của đề tài.

- Xây dựng một hệ thống website quản lý đặt sân bóng mini trực tuyến với đầy đủ các chức năng cơ bản và cần thiết.
- Hỗ trợ người dùng:
  - Đăng ký, Đăng nhập.
  - Xem lịch sân trống.
  - Đặt sân trực tuyến một cách dễ dàng, nhanh chóng.
- Hỗ trợ chủ sân:
  - Quản lý thông tin các sân bóng.
  - Quản lý lịch đặt, khách hàng và doanh thu.
  - Dễ dàng thêm, sửa, xóa sân, kiểm soát lịch đặt sân.

- Giao diện trực quan, thân thiện, dễ sử dụng trên cả máy tính và thiết bị di động.
- Đảm bảo hệ thống ổn định, bảo mật dữ liệu người dùng và dễ dàng mở rộng trong tương lai.

### 1.3. Phạm vi và đối tượng áp dụng

Phạm vi:

- Xây dựng hệ thống quản lý đặt sân bóng đá mini gồm các chức năng cơ bản: đăng nhập, đăng ký, đặt sân, quản lý sân, quản lý lịch đặt và quản lý người dùng.
- Hệ thống vận hành trong môi trường nội bộ hoặc online với máy chủ web và cơ sở dữ liệu.

Đối tượng sử dụng:

- Người dùng: Các khách hàng có nhu cầu tìm kiếm, tra cứu và đặt sân bóng.
- Người quản lý: Chủ sân hoặc nhân viên quản lý chịu trách nhiệm theo dõi lịch đặt, khách hàng, cập nhật thông tin sân và kiểm soát doanh thu.

### 1.4. Phương pháp thực hiện

- Thu thập và phân tích yêu cầu thực tế từ các mô hình quản lý sân bóng hiện hành.
- Thiết kế hệ thống:
  - Xây dựng sơ đồ Use-case, sơ đồ dữ liệu (ERD), thiết kế giao diện.
- Lập trình và triển khai:
  - Chạy thử nghiệm các chức năng trên môi trường máy chủ local.
- Đánh giá:
  - Kiểm tra hiệu suất, tính năng, tính ổn định và mức độ đáp ứng yêu cầu thực tế.

### 1.5. Cấu trúc báo cáo

Báo cáo được chia thành các chương chính như sau:

- Chương 1: Giới thiệu đề tài.
- Chương 2: Cơ sở lý thuyết và công nghệ sử dụng.
- Chương 3: Phân tích hệ thống.
- Chương 4: Thiết kế và triển khai hệ thống.
- Chương 5: Kết quả thực nghiệm và đánh giá.
- Chương 6: Kết luận và hướng phát triển.
- Chương 7: Phân công và vai trò nhóm.
- Chương 8: Tài liệu tham khảo.

## 2. CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG

### 2.1. Giới thiệu về phát triển web theo mô hình MVC

#### a) Khái niệm MVC

MVC (Model – View – Controller) là mô hình kiến trúc phần mềm được sử dụng phổ biến trong phát triển ứng dụng web hiện đại.

Mục tiêu: Phân tách rõ ràng thành ba thành phần chính:

- Model: Quản lý dữ liệu và logic nghiệp vụ.
- View: Giao diện người dùng, hiển thị dữ liệu.
- Controller: Trung gian xử lý các yêu cầu từ người dùng, tương tác giữa các Model và View.

#### b) Lợi ích của MVC

- Dễ bảo trì và mở rộng.
- Tăng khả năng sử dụng mã nguồn.
- Phân tách rõ ràng giữa giao diện và xử lý logic.
- Hỗ trợ làm việc nhóm hiệu quả (frontend/backend tách biệt).

### 2.2. Laravel Framework

#### a) Giới thiệu về Laravel

- Là một framework PHP mạnh mẽ, miễn phí và mã nguồn mở.
- Ra đời vào năm 2011, Laravel nhanh chóng trở thành framework phổ biến nhất nhờ sự đơn giản, linh hoạt và đầy đủ tính năng.

#### b) Kiến trúc Laravel

Laravel tuân theo mô hình MVC, giúp phân tách rõ ràng giữa dữ liệu, giao diện và xử lý logic.

Cấu trúc thư mục chính:

- /app: Chứa các thành phần của ứng dụng (Models, Controllers).
- /route: Định nghĩa các tuyến đường (routes) của ứng dụng.
- /resources/views: Giao diện (View) sử dụng Blade Templates.
- /database: Cấu hình và migration cơ sở dữ liệu.
- /public: Thư mục chứa các tài nguyên public (CSS, JS, ảnh).

#### c) Các tính năng nổi bật

- Hệ thống Routing đơn giản và mạnh mẽ.
- Blade Template: Template engine mạnh giúp tách biệt giao diện và logic.
- Hệ thống Migration: Quản lý và đồng bộ cơ sở dữ liệu.
- Eloquent ORM: Tương tác cơ sở dữ liệu dưới dạng đối tượng.
- Authentication & Authorization: Tích hợp sẵn chức năng đăng nhập, phân quyền.



- Hỗ trợ Queue, Jobs, Events: Phục vụ các tác vụ phức tạp.
- Bảo mật: CSRF, XSS, SQL Injection đều được xử lý mặc định.

### 2.3. Blade Template Engine

- Giới thiệu
  - Blade là hệ thống template engine mặc định của Laravel.
  - Cho phép sử dụng cú pháp đơn giản để nhúng logic PHP và HTML.
- Ưu điểm của Blade
  - Nhẹ, nhanh, dễ học.
  - Hỗ trợ kế thừa giao diện ( @extends, @section, @yield).

### 2.4. MySQL Database

- Giới thiệu
  - MySQL là hệ quản trị cơ sở dữ liệu quan hệ phổ biến nhất thế giới.
  - Dễ học, miễn phí, hiệu suất cao
- Lý do lựa chọn
  - Tương thích tốt với Laravel thông qua Eloquent ORM.
  - Dễ dàng cài đặt và triển khai trên các máy chủ hosting hoặc local.
- Mô hình dữ liệu

Cơ sở dữ liệu của hệ thống gồm các bảng chính như sau:

Bảng	Mô tả
users	Lưu thông tin người dùng (admin, khách hàng)
fields	Thông tin các sân bóng (tên sân, vị trí, giá,..)
bookings	Quản lý thông tin các lần đặt sân, trạng thái, thời gian.
articles	Quản lý bài viết, tin tức liên quan đến sân bóng hoặc hệ thống.
videos	Video giới thiệu sân hoặc video hướng dẫn.
password_resets	Đặt lại mật khẩu người dùng (Laravel mặc định).
personal_access_tokens	Token API phục vụ cho chức năng đăng nhập qua API (Laravel Sanctum).
failed_jobs	Quản lý các job bị lỗi trong queue của Laravel.
migrations	Theo dõi lịch sử migration để đồng bộ dữ liệu.

- Mô hình dữ liệu và mối quan hệ
  - users ↔ bookings: Một người dùng có thể thực hiện nhiều lần đặt sân.
  - users ↔ booking: Một người dùng có thể có nhiều lịch sử đặt sân.
  - fields ↔ bookings: Một sân có thể được đặt nhiều lần bởi nhiều người.
  - fields ↔ booking: Ghi nhận lịch sử đặt của từng sân.

- Mô hình dữ liệu tuân theo chuẩn quan hệ 1-N giữa các bảng như users – bookings, fields – bookings.
- e) Ưu điểm khi sử dụng MySQL trong hệ thống.
- Quản lý dữ liệu khoa học, chặt chẽ thông qua mô hình quan hệ.
  - Hỗ trợ tối ưu các truy vấn phức tạp, cần thiết cho việc kiểm tra lịch trống, thống kê doanh thu.
  - Kết hợp hiệu quả với Laravel thông qua Eloquent ORM, giúp việc xây dựng CRUD trở nên đơn giản.
  - Tăng khả năng mở rộng hệ thống: dễ dàng bổ sung thêm chức năng như thanh toán online, thông báo tự động... trong tương lai.
- f) Tổng kết
- Việc sử dụng MySQL cho hệ thống quản lý đặt sân bóng mini giúp đảm bảo dữ liệu được lưu trữ an toàn, dễ dàng truy xuất và mở rộng. Kết hợp cùng với Laravel, hệ thống vận hành ổn định, hiệu quả và dễ dàng bảo trì trong tương lai.

## 2.5. Bootstrap Framework

### a) Giới thiệu về Bootstrap

Bootstrap là một framework front-end mã nguồn mở, được phát triển bởi nhóm kỹ sư của Twitter. Đây là bộ công cụ mạnh mẽ giúp xây dựng giao diện web hiện đại, trực quan, tương thích tốt với nhiều thiết bị (desktop, tablet, mobile) theo chuẩn responsive design.

Bootstrap cung cấp sẵn các thành phần giao diện như: navbar, buttons, forms, tables, carousel, modal, alert... cùng với hệ thống lưới (grid system) linh hoạt giúp lập trình viên dễ dàng xây dựng bố cục trang web nhanh chóng và nhất quán.

### b) Lý do lựa chọn Bootstrap

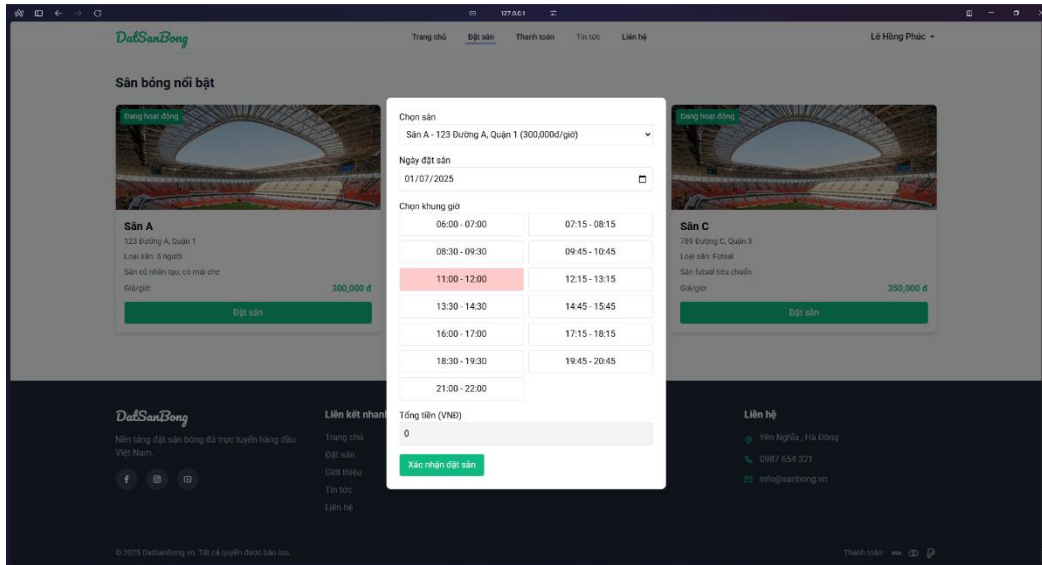
- Thiết kế responsive: Giao diện tự động thích ứng với mọi kích thước màn hình (máy tính, điện thoại, máy tính bảng...).
- Dễ sử dụng: Cú pháp đơn giản, dễ tích hợp với Blade Template của Laravel.
- Tiết kiệm thời gian: Cung cấp sẵn nhiều thành phần UI giúp giảm thời gian thiết kế giao diện.
- Tùy chỉnh linh hoạt: Có thể mở rộng hoặc tùy biến theo nhu cầu dự án.
- Cộng đồng lớn: Tài liệu đầy đủ, nhiều ví dụ thực tế, dễ tra cứu khi gặp vấn đề.

### c) Ứng dụng của Bootstrap trong hệ thống

Bootstrap được áp dụng xuyên suốt trong toàn bộ hệ thống để xây dựng giao diện người dùng, cụ thể:

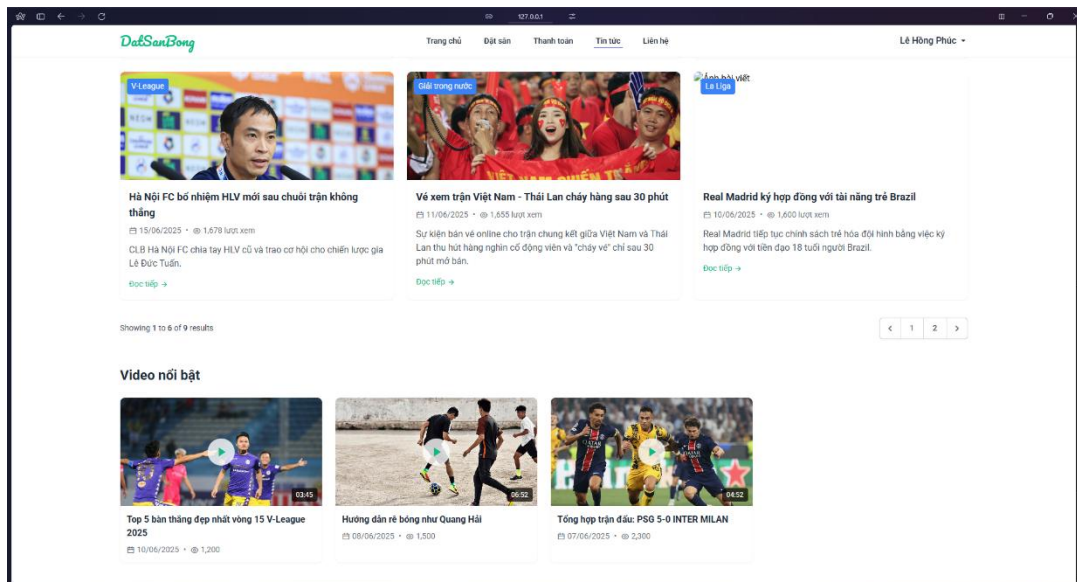
- Trang chủ: Thiết kế bố cục hiển thị thông tin sân bóng, tin tức, video giới thiệu.

- Trang đặt sân:
  - Form đặt sân sử dụng layout form của Bootstrap giúp hiển thị gọn gàng, đẹp mắt.
  - Kiểm tra lịch trống bằng các component như table, calendar.



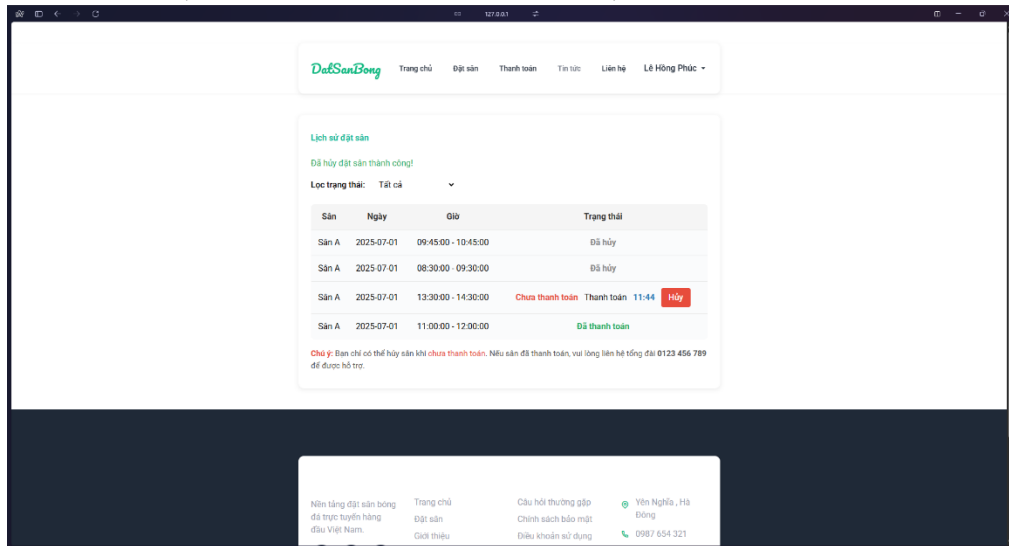
Hình 2.1: Giao diện chọn lịch đặt sân (lịch đã đặt được bôi đỏ như hình)

- Tiến hành thanh toán sân đã đặt, nếu không thanh toán, sân sẽ tự động hủy sau 15 phút.
- Trang tin tức:
  - Cho phép người dùng tìm kiếm, truy cập và đọc tin tức, xem video thể thao. Các video có lượt xem cao sẽ được hiển thị ở mục đề xuất.



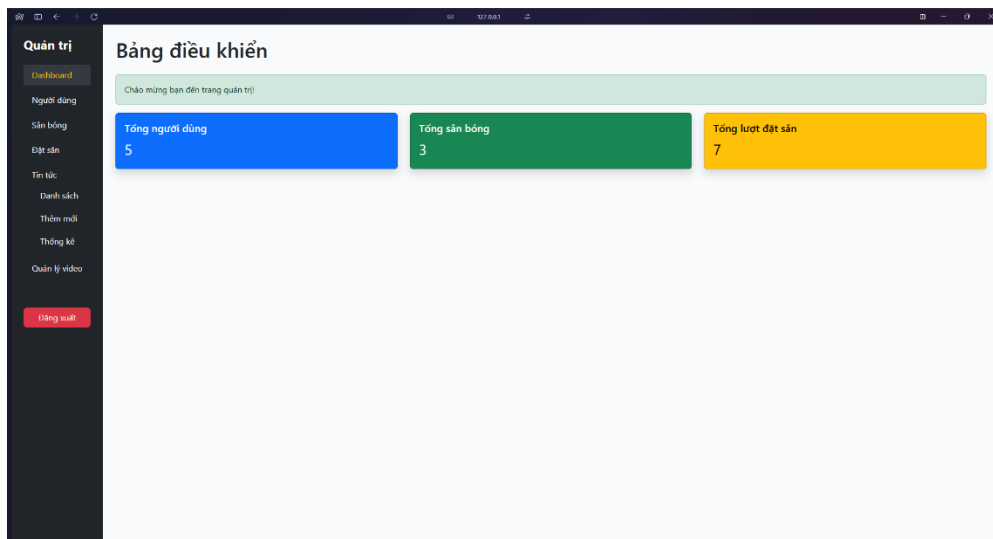
Hình 2.2: Giao diện trang tin tức

- Trang sân đã đặt:
  - Hiện thị lịch sử đặt sân của người dùng, người dùng có thể lọc theo trạng thái (Chưa thanh toán/đã thanh toán).



Hình 2.3: Giao diện trang lịch sử đặt sân

- Trang quản lý (Admin):
  - Giao diện dashboard quản trị.
  - Quản lý user, sân bóng, lịch đặt, bài viết, video.



Hình 2.4: Giao diện trang Admin

- Trang đăng nhập/đăng ký:
  - Form xác thực sử dụng component form group, input, button của Bootstrap.
- Bố cục chung:
  - Sử dụng hệ thống grid (container, row, col) giúp nội dung hiển thị chuẩn hóa, cân đối trên mọi thiết bị.

- Navbar cố định trên cùng giúp người dùng dễ dàng điều hướng giữa các trang.
- Footer đồng nhất ở tất cả các trang.

d) Thành phần chính của Bootstrap được sử dụng

Thành phần	Mục đích sử dụng
Grid System	Thiết kế bố cục theo lưới linh hoạt.
Navbar	Tạo thanh điều hướng chính.
Buttons	Tạo các nút chức năng đăng nhập, đăng ký, đặt sân...
Forms	Thiết kế các form đăng nhập, đăng ký, đặt sân...
Cards	Hiển thị thông tin sân bóng, bài viết, video...
Table	Quản lý dữ liệu như danh sách đặt sân, người dùng.
Alerts & Modals	Thông báo, popup xác nhận hành động.

e) Ưu điểm của Bootstrap trong hệ thống

- Giao diện đẹp, chuyên nghiệp, nhất quán.
- Đáp ứng tiêu chuẩn web hiện đại với khả năng tương thích đa thiết bị.
- Tiết kiệm thời gian phát triển frontend, giúp tập trung hơn vào logic backend.
- Dễ dàng mở rộng, bổ sung thêm tính năng hoặc giao diện mới.
- Kết hợp hoàn hảo với Blade Template của Laravel, giúp quản lý layout và component frontend hiệu quả.

f) Tổng kết

Bootstrap đóng vai trò quan trọng trong việc thiết kế giao diện cho hệ thống quản lý đặt sân bóng mini. Nhờ vào các component sẵn có, hệ thống đảm bảo tính trực quan, hiện đại và dễ sử dụng. Sự kết hợp giữa Bootstrap và Laravel Blade Template giúp xây dựng giao diện đồng bộ, thân thiện với người dùng, đồng thời tiết kiệm đáng kể thời gian và công sức trong quá trình phát triển.

## 2.6. Các công cụ và công nghệ hỗ trợ

a) Công cụ phát triển phần mềm

Công cụ	Mục đích sử dụng
Vissual Studio Code	Trình soạn thảo mã nguồn, hỗ trợ nhiều extension mạnh mẽ.

XAMPP	Tạo môi trường máy chủ web cục bộ với Apache, PHP, MySQL.
MySQL / phpMyAdmin	Quản lý cơ sở dữ liệu, thực thi truy vấn SQL.
Git	Quản lý phiên bản mã nguồn, theo dõi thay đổi.
Github	Lưu trữ mã nguồn online, làm việc nhóm, backup dự án

b) Công nghệ sử dụng trong dự án

Công nghệ	Mô tả
PHP 8.x	Ngôn ngữ lập trình backend chính, nền tảng cho Laravel
Laravel Framework 10.x	Framework PHP theo mô hình MVC, mạnh mẽ, bảo mật và dễ mở rộng.
Blade Template	Template engine của Laravel, kết hợp frontend và backend.
Bootstrap 5.x	Framework CSS giúp xây dựng giao diện responsive, đẹp mắt.
Eloquent ORM	Hệ thống ORM giúp thao tác database bằng mô hình hướng đối tượng
HTML5, CSS3, JavaScript	Công nghệ nền tảng cho frontend web.

c) Quy trình phát triển phần mềm

Dự án được triển khai theo quy trình phát triển phần mềm truyền thống (Waterfall) với các bước chính:

1. Khảo sát và phân tích yêu cầu.
2. Thiết kế hệ thống:
  - Thiết kế cơ sở dữ liệu (ERD).
  - Thiết kế giao diện người dùng (UI/UX).
3. Lập trình:
  - Phát triển backend với Laravel.
  - Phát triển frontend với Blade Template + Bootstrap.
4. Kiểm thử
  - Kiểm thử chức năng (Functional Testing).
  - Kiểm thử giao diện (UI Testing).
5. Triển khai
  - Đưa hệ thống chạy trên môi trường Local.

#### 6. Bảo trì

- Sửa lỗi, tối ưu hiệu suất.
- Phát triển thêm các chức năng mới.

#### d) Môi trường chạy ứng dụng

Thành phần	Thông tin
Hệ điều hành	Windows, MacOS hoặc Linux
Web Server	Apache (XAMPP, Laragon) hoặc Nginx
PHP Version	PHP 8.x
Database	MySQL
Trình duyệt	Chrome, Arc, Edge, Safari...
IDE	Visual Studio Code với các extention Laravel, PHP...

### 3. PHÂN TÍCH HỆ THỐNG

#### 3.1. Phân tích yêu cầu hệ thống

##### a) Yêu cầu chức năng

Hệ thống được thiết kế với hai nhóm người dùng chính: Người dùng (Khách hàng) và Người quản trị (Admin). Các chức năng chính bao gồm:

- Đối với khách hàng:
  - Đăng ký tài khoản, đăng nhập, quên mật khẩu.
  - Xem danh sách các sân bóng có sẵn.
  - Xem lịch trống của từng sân.
  - Đặt sân online theo ngày và khung giờ mong muốn.
  - Quản lý thông tin cá nhân.
  - Theo dõi lịch sử đặt sân.
  - Thanh toán/hủy sân đã đặt (hệ thống chỉ cho phép hủy với các sân chưa tiến hành thanh toán).
  - Xem tin tức, bài viết hoặc video giới thiệu sân bóng.
  - Tìm kiếm tin tức
- Đối với người quản trị (Admin):
  - Đăng nhập vào trang quản trị.
  - Quản lý thông tin người dùng (thêm, sửa, xóa).

- Quản lý sân bóng (fields): thêm, sửa, xóa, cập nhật giá, trạng thái sân.
- Quản lý lịch đặt sân (bookings): xác nhận, hủy, kiểm tra tình trạng.
- Quản lý tin tức (news): Thêm, sửa, xóa các tin tức hiển thị.
- Quản lý video (videos): Thêm, sửa, xóa video.

b) Yêu cầu phi chức năng.

- Khả năng mở rộng: Dễ dàng thêm các chức năng như thanh toán online, thông báo email/SMS.
- Hiệu suất: Đảm bảo phản hồi nhanh dưới 3 giây với các thao tác thông thường.
- Tính ổn định: Hệ thống hoạt động liên tục, hạn chế downtime.
- Tính bảo mật:
  - Xác thực và phân quyền người dùng.
  - Bảo mật thông tin cá nhân và lịch sử đặt sân.
  - Phòng chống các lỗi phổ biến: SQL Injection, XSS, CSRF.
- Tính dễ sử dụng:
  - Giao diện trực quan, dễ hiểu cho cả người dùng và admin.
  - Responsive – tương thích với mọi thiết bị (PC, tablet, mobile).

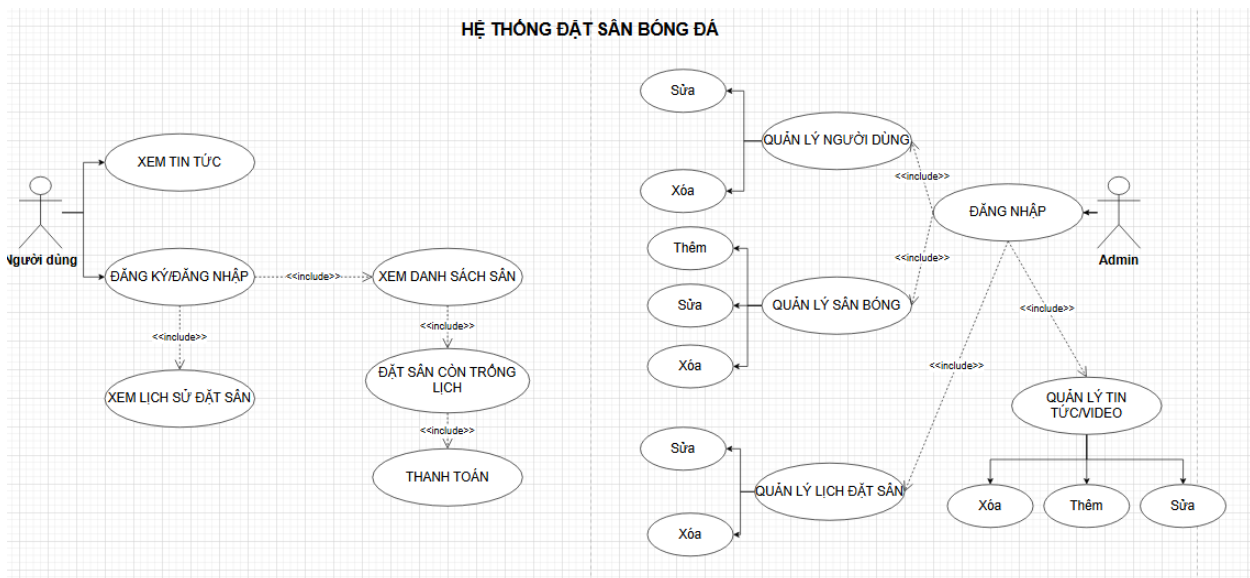
### 3.2. Sơ đồ Use-Case

Mô tả Use-case chính:

- Đối với người dùng (Khách hàng):
  - Đăng ký, đăng nhập, cập nhật thông tin.
  - Xem danh sách sân, xem chi tiết sân.
  - Kiểm tra lịch trống.
  - Đặt sân.
  - Xem lịch sử đặt sân.
  - Đọc bài viết, xem video.
- Đối với quản trị viên (Admin):
  - Đăng nhập hệ thống quản trị.
  - Quản lý user.
  - Quản lý sân bóng.
  - Quản lý đặt sân.
  - Quản lý bài viết, video.
  - Xem báo cáo, thống kê.

Sơ đồ use-case tổng quát:





*Hình 3.1: Sơ đồ Use-case hệ thống đặt sân*

### 3.3. Thiết kế cơ sở dữ liệu (ERD)

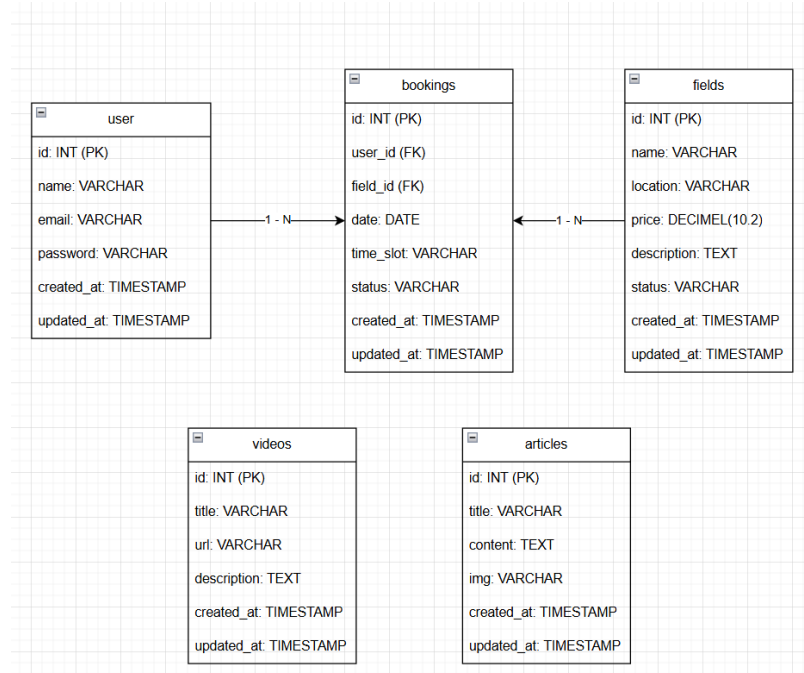
Các bảng chính trong cơ sở dữ liệu:

Bảng	Mô tả
users	Lưu thông tin người dùng (admin, khách hàng)
fields	Thông tin các sân bóng (tên sân, vị trí, giá,..)
bookings	Quản lý thông tin các lần đặt sân, trạng thái, thời gian.
articles	Quản lý bài viết, tin tức liên quan đến sân bóng hoặc hệ thống.
videos	Video giới thiệu sân hoặc video hướng dẫn.
password_resets	Đặt lại mật khẩu người dùng (Laravel mặc định).
personal_access_tokens	Token API phục vụ cho chức năng đăng nhập qua API (Laravel Sanctum).
failed_jobs	Quản lý các job bị lỗi trong queue của Laravel.
migrations	Theo dõi lịch sử migration để đồng bộ dữ liệu.

Mối quan hệ giữa các bảng:

- Một user có thể có nhiều bookings.
- Một field có thể được đặt nhiều lần (quan hệ 1-N với bookings).
- bookings liên kết với users và fields.

## Sơ đồ ERD (Entity Relationship Diagram)



Hình 3.2: Mô hình ERD hệ thống

### 3.4. Phân tích luồng hoạt động chính

Luồng hoạt động đặt sân:

1. Người dùng đăng nhập vào hệ thống.
2. Truy cập danh sách sân → Xem thông tin sân → Kiểm tra lịch trống.
3. Chọn ngày, giờ → Đặt sân → Hệ thống kiểm tra xem sân trống hay không.
4. Nếu trống → Xác nhận đặt sân → Lưu vào bảng bookings.
5. Người dùng tiến hành thanh toán sân đã đặt. Nếu không tiến hành thanh toán, hệ thống sẽ tự động hủy lịch đặt sau 15 phút.

Luồng hoạt động quản lý sân:

1. Admin đăng nhập vào dashboard.
2. Truy cập mục quản lý sân → Thêm mới, chỉnh sửa hoặc xóa sân.
3. Cập nhật giá thuê, tình trạng hoạt động của sân.

Luồng quản lý lịch đặt (Admin):

1. Admin xem danh sách booking → Kiểm tra trạng thái.
2. Xác nhận booking hoặc hủy nếu cần thiết.
3. Ghi nhận thay đổi vào booking.

## 4. Thiết kế và triển khai hệ thống

### 4.1. Thiết kế kiến trúc hệ thống.

#### a) Kiến trúc tổng thể

Hệ thống được xây dựng theo mô hình Client – Server, trong đó:

- Client:
  - Trình duyệt web trên máy tính, điện thoại di động, máy tính bảng.
  - Giao diện xây dựng bằng HTML, CSS, JavaScript, Blade Template và Bootstrap.
- Server:
  - Backend sử dụng Laravel Framework, xử lý các yêu cầu từ client.
  - Giao tiếp với cơ sở dữ liệu MySQL để lấy và ghi dữ liệu.
- Database:
  - Lưu trữ dữ liệu người dùng, sân bóng, lịch đặt sân, bài viết, video, lịch sử đặt,...

#### b) Mô hình kiến trúc MVC

Thành phần	Mô tả
Model	Quản lý dữ liệu và logic nghiệp vụ (User, Booking, Fields,..)
View	Giao diện hiển thị (Blade Template)
Controller	Xử lý yêu cầu người dùng, điều phối dữ liệu giữa Model và View

### 4.2. Thiết kế cơ sở dữ liệu chi tiết

#### a) Bảng user

Trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
name	VARCHAR	Tên người dùng
email	VARCHAR	Địa chỉ email
role	ENUM	Quyền (admin, user)
create_at	TIMESTAMP	Thời gian tạo
update_at	TIMESTAMP	Thời gian cập nhật

#### b) Bảng fields

Trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
name	VARCHAR	Tên sân bóng
location	VARCHAR	Địa chỉ

price	DECIMAL	Giá thuê theo giờ
status	ENUM	Trạng thái (available, busy,...)
create_at	TIMESTAMP	Thời gian tạo
update_at	TIMESTAMP	Thời gian cập nhật

c) Bảng bookings

Trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
user_id	INT	Liên kết bảng User
field_id	INT	Liên kết bảng Field
date	DATE	Ngày đặt sân
time_slot	VARCHAR	Khung giờ (VD: 17h-18h)
status	ENUM	Trạng thái (pending, confirmed, cancelled)
create_at	TIMESTAMP	Thời gian booking
update_at	TIMESTAMP	Thời gian cập nhật

d) Bảng articles

Trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
title	VARCHAR	Tiêu đề bài viết
content	VARCHAR	Nội dung bài viết
image	VARCHAR	Đường dẫn ảnh
create_at	TIMESTAMP	Thời gian tạo
update_at	TIMESTAMP	Thời gian cập nhật

e) Bảng video

Trường	Kiểu dữ liệu	Mô tả
id	INT	Khóa chính
title	VARCHAR	Tiêu đề bài viết
url	VARCHAR	Đường dẫn video (Youtube...)
description	TEXT	Mô tả
create_at	TIMESTAMP	Thời gian tạo
update_at	TIMESTAMP	Thời gian cập nhật

### 4.3. Thiết kế giao diện người dùng

c) Giao diện người dùng (User)

- Trang chủ hiển thị danh sách sân, bài viết, video.

- Trang đặt sân với form chọn ngày, giờ và sân.
  - Trang lịch sử đặt sân.
  - Trang chi tiết sân bóng.
  - Trang đăng nhập, đăng ký, quản lý tài khoản cá nhân.
- d) Giao diện quản trị (Admin)
- Dashboard tổng quan: Thống kê lượt đặt, doanh thu.
  - Quản lý user: Thêm, sửa, xóa tài khoản.
  - Quản lý sân bóng: Thêm mới, cập nhật trạng thái sân.
  - Quản lý tin tức: Thêm, sửa, xóa bài viết, video.

#### 4.4. Triển khai Backend

Backend là phần lõi xử lý toàn bộ logic nghiệp vụ, tương tác với cơ sở dữ liệu và xử lý các yêu cầu từ phía người dùng. Hệ thống backend được xây dựng trên nền tảng Laravel Framework phiên bản 10, với mô hình kiến trúc MVC (Model – View – Controller) rõ ràng và khoa học.

##### a) Cấu trúc thư mục trong Laravel

Laravel cung cấp cấu trúc thư mục rõ ràng và tổ chức tốt, bao gồm:

Thư mục	Mô tả
app/	Chứa code logic chính, Models, Controllers, Services.
app/Models/	Định nghĩa các Models tương ứng với bảng trong CSDL.
app/Http/Controller	Xử lý Request từ client, điều hướng dữ liệu.
resources/views/	Giao diện frontend sử dụng Blade Template.
routes/	Định nghĩa các đường truyền (route) cho hệ thống.
database/migrations/	File định nghĩa cấu trúc bảng trong CSDL.
public/	Chứa thư mục public, chứa ảnh, CSS, JS.

##### b) Quản lý Routing

Laravel quản lý các đường dẫn (route) thông qua file routes/web.php (dành cho web) và routes/api.php (dành cho API nếu cần).

Cấu trúc routing chính:

- Dành cho người dùng:
  - / → Trang chủ.
  - /fields → Danh sách sân bóng.

- /bookings → Trang đặt sân.
- /history → Lịch sử đặt sân
- Dành cho quản trị viên (Admin):
  - /admin/dashboard → Trang thống kê
  - /admin/user → Quản lý người dùng
  - /admin/fields → Quản lý sân bóng
  - /admin/article → Quản lý bài viết
  - /admin/videos → Quản lý video

Các route này được nhóm theo middleware auth và admin để kiểm soát quyền truy cập.

#### c) Controller – Xử lý logic nghiệp vụ

Mỗi Controller đảm nhận một vai trò riêng biệt tương ứng với từng nghiệp vụ.

Controller	Chức năng chính
AuthController	Đăng ký, Đăng nhập, Đăng xuất, Đặt lại mật khẩu.
UserController	Quản lý thông tin người dùng
BookingController	Đặt sân, kiểm tra lịch trống, hủy Booking
PaymentController	Thanh toán sân đã đặt
AdminController	Dashboard quản trị, báo cáo thống kê
ArticleController	Đăng bài viết tin tức
VideoController	Quản lý video giới thiệu

Ví dụ luồng xử lý đặt sân trong BookingController:

- Người dùng gửi request đặt sân → Controller nhận request.
- Kiểm tra xem sân trống trong ngày và khung giờ đó hay không.
- Nếu trống:
  - Tạo bản ghi trong bảng Bookings
- Trả về thông báo xác nhận đặt sân thành công hoặc thất bại

#### d) Model – Tương tác cơ sở dữ liệu

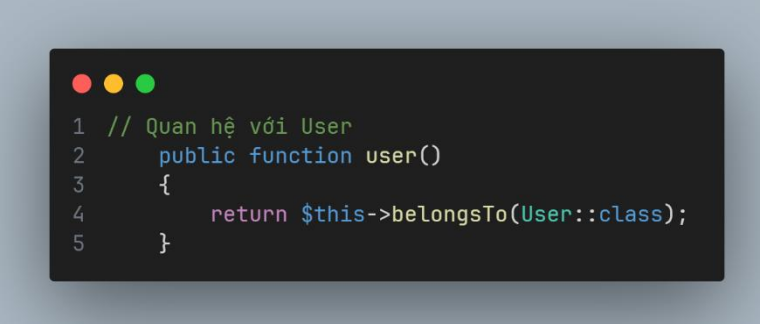
Laravel sử dụng Eloquent ORM, giúp mỗi bảng trong cơ sở dữ liệu tương ứng với một model.

Model	Bảng tương ứng
User	users
Field	fields
Booking	bookings

Model	Bảng tương ứng
Article	articles
Video	videos

Đặc điểm:

- Mỗi model định nghĩa mối quan hệ với các model khác (One to Many, Many to Many,...)
- Ví dụ:



```

1 // Quan hệ với User
2 public function user()
3 {
4     return $this->belongsTo(User::class);
5 }

```

Hình 4.1: Quan hệ belongsTo với User – Laravel Eloquent

#### e) Authentication & Authorization (Xác thực và Phân quyền)

Laravel hỗ trợ hệ thống xác thực mạnh mẽ:

- Authentication:
  - Laravel Breeze hoặc Laravel Jetstream hỗ trợ nhanh chóng đăng ký, đăng nhập.
  - Hash mật khẩu bằng thuật toán bcrypt để đảm bảo an toàn.
- Authorization (Phân quyền):
  - Middleware auth đảm bảo người dùng đã đăng nhập mới truy cập được chức năng.
  - Middleware admin kiểm tra quyền hạn của tài khoản admin.

#### f) Quản lý lỗi và xử lý ngoại lệ

Laravel tự động quản lý lỗi 404, 500, lỗi truy cập sai route.

Ngoài ra, có thể custom lỗi trong `app/Exceptions/Handler.php` để hiển thị thông báo thân thiện hơn với người dùng.

#### g) Gửi thông báo

Gửi thông báo khi:

- Đặt sân thành công.
- Thay đổi trạng thái booking.

#### h) Bảo mật hệ thống Backend

- Hash mật khẩu bằng bcrypt.
- Bảo vệ khỏi SQL Injection nhờ Eloquent ORM.

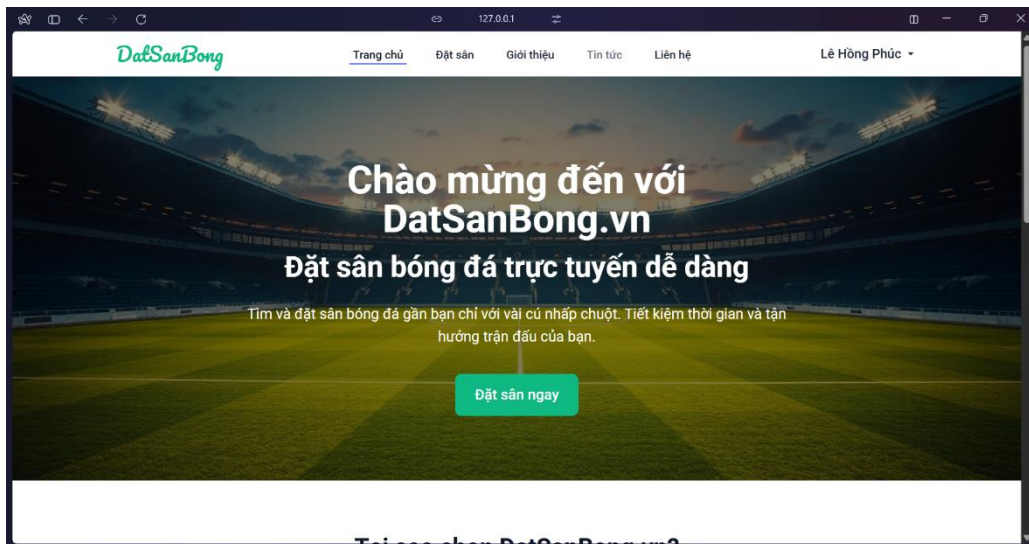
- Chống CSRF với token tự động trong form.
- Chống XSS thông qua việc escape dữ liệu trong Blade Template.
- Phân quyền chặt chẽ giữa người dùng và admin.

#### 4.5. Triển khai Frontend.

Frontend được xây dựng bằng Blade Template kết hợp với Bootstrap 5, đảm bảo giao diện trực quan, thân thiện và tương thích trên nhiều thiết bị.

a) Cấu trúc giao diện chính:

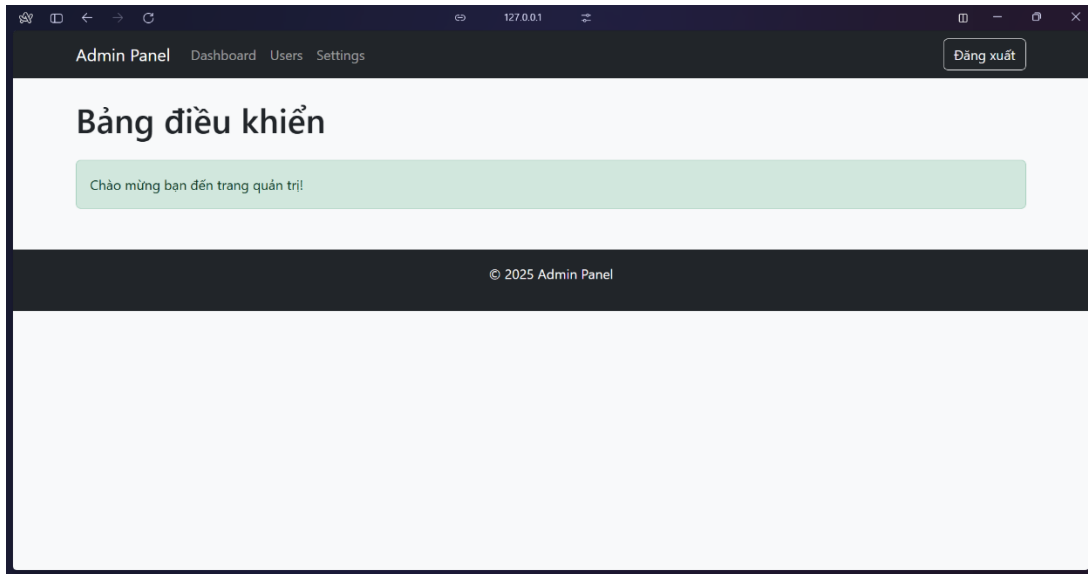
- Trang người dùng:
  - Trang chủ: Hiển thị danh sách sân, bài viết, video.
  - Trang đặt sân: Form chọn sân, ngày, giờ.
  - Trang chi tiết sân: Mô tả, hình ảnh, giá.
  - Lịch sử đặt sân: Xem lại các booking đã thực hiện.
  - Đăng nhập, đăng ký, quản lý tài khoản.



Hình 4.1: Giao diện trang chủ của người dùng

- Trang admin:
  - Dashboard tổng quan: Thống kê booking, người d
  - Quản lý người dùng.
  - Quản lý sân.
  - Quản lý lịch đặt sân.
  - Quản lý tin tức.





*Hình 4.2: Giao diện admin hệ thống*

- b) Công nghệ sử dụng:
  - Blade Template: Kết nối dữ liệu từ backend tới giao diện.
  - Bootstrap 5: Đảm bảo responsive, thiết kế hiện đại.
  - HTML, CSS, JavaScript, JQuery: Xây dựng tương tác người dùng.
- c) Tính năng giao diện:
  - Form có validate phía frontend và backend.
  - Thông báo thành công/thất bại bằng Toast hoặc Alert.
  - Bố cục rõ ràng, dễ sử dụng cho cả người dùng và quản trị viên.

## 5. Kết quả thực nghiệm và đánh giá

### 5.1. Kết quả triển khai

Hệ thống đã được triển khai thành công trên môi trường localhost.

- a. Các chức năng chính hoạt động tốt
  - Đăng ký, đăng nhập, phân quyền user/admin.
  - Đặt sân theo ngày và khung giờ, kiểm tra lịch trống.
  - Quản lý sân, bài viết, video.
  - Theo dõi lịch sử đặt sân và thay đổi booking.
  - Dashboard admin hiển thị số lượng booking, doanh thu.
- b. Giao diện thực tế
  - Giao diện người dùng thân thiện, trực quan, responsive.
  - Trang quản trị dễ sử dụng, quản lý dữ liệu hiệu quả.

## 5.2. Kiểm thử chức năng

- Phương pháp:
  - Kiểm thử thủ công trên trình duyệt (Chrome, Firefox).
  - Kiểm thử theo từng chức năng: đăng nhập, đặt sân, huỷ sân, thêm sân, quản lý user...
- Kết quả:

Chức năng	Kết quả
Đăng nhập	Đạt
Đăng ký	Đạt
Đặt sân	Đạt
Kiểm tra lịch trống	Đạt
Quản lý sân	Đạt
Quản lý user	Đạt
Quản lý booking	Đạt
Dashboard thống kê	Đạt

## 5.3. Đánh giá hiệu suất

- Tốc độ phản hồi: Trung bình < 5 giây cho các thao tác thông thường.
- Khả năng mở rộng: Dễ dàng nâng cấp với các chức năng như thanh toán online, gửi email thông báo.

## 5.4. Ưu điểm và hạn chế

- Ưu điểm:
  - Giao diện thân thiện, dễ sử dụng.
  - Phân quyền rõ ràng user và admin.
  - Quản lý lịch đặt sân minh bạch, chính xác.
  - Bảo mật tốt với Laravel: CSRF, XSS, SQL Injection.
  - Dễ dàng mở rộng trong tương lai.
- Hạn chế:
  - Chưa tích hợp thanh toán online.
  - Hệ thống báo lỗi còn đơn giản, cần làm đẹp hơn.
  - Hiệu suất có thể chưa tối ưu khi triển khai quy mô lớn.

## 5.5. Bài học và kinh nghiệm

- Hiểu sâu hơn về mô hình MVC trong Laravel.
- Kỹ năng thiết kế cơ sở dữ liệu tốt hơn.
- Nâng cao khả năng xử lý frontend với Blade và Bootstrap.
- Quan trọng hoá việc kiểm thử phần mềm trước khi triển khai.

## **6. Kết luận và hướng phát triển**

### **6.1. Kết luận**

Trong quá trình thực hiện đề tài “Xây dựng hệ thống quản lý đặt sân bóng mini”, nhóm đã hoàn thành các nội dung chính sau:

- Thiết kế và xây dựng hệ thống quản lý đặt sân trực tuyến đầy đủ chức năng.
- Áp dụng mô hình MVC trong Laravel để tổ chức và quản lý mã nguồn hiệu quả.
- Hệ thống cho phép người dùng thực hiện các thao tác như:
  - Đăng ký, đăng nhập, quản lý tài khoản.
  - Xem danh sách sân, kiểm tra lịch trống và đặt sân online.
  - Quản lý lịch sử booking, bài viết và video.
- Phía admin có thể dễ dàng quản lý toàn bộ dữ liệu: sân bóng, người dùng, lịch đặt, thống kê doanh thu.
- Đảm bảo các yếu tố bảo mật cơ bản, tính ổn định và khả năng mở rộng của hệ thống.

Hệ thống đã chạy ổn định trong môi trường thử nghiệm và sẵn sàng triển khai thực tế.

### **6.2. Hạn chế của hệ thống**

- Chưa tích hợp tính năng thanh toán online (VNPay, Momo...).
- Hệ thống gửi thông báo mới dùng ở mức cơ bản, chưa có SMS/email.
- Báo cáo thống kê còn đơn giản, cần trực quan hơn bằng biểu đồ, dashboard nâng cao.

### **6.3. Hướng phát triển trong tương lai**

- Tích hợp thanh toán trực tuyến: Giúp người dùng thanh toán dễ dàng qua ví điện tử, thẻ ngân hàng.
- Xây dựng ứng dụng mobile: Triển khai app Android/iOS đồng bộ với hệ thống web.
- Nâng cấp báo cáo thống kê: Thêm biểu đồ trực quan, báo cáo doanh thu, số lượt đặt sân theo thời gian.
- Tích hợp thông báo đa kênh: Gửi email, SMS, thông báo đẩy trên mobile khi có thay đổi booking.
- Tối ưu hiệu suất: Áp dụng cache Redis, tối ưu truy vấn database và sử dụng load balancing khi cần.

- Hỗ trợ nhiều ngôn ngữ: Phát triển hệ thống hỗ trợ song ngữ (Việt – Anh) để mở rộng đối tượng khách hàng.

## 7. Phân công vai trò trong nhóm

STT	Thành viên	Công việc phụ trách
1	Nguyễn Văn Lượng	Phát triển Back-end (Laravel, DB)
2	Lý Hoài Nam	Thiết kế giao diện (HTML/CSS, Blade)
3	Lê Hồng Phúc	Triển khai & Kiểm thử (Git, Test, Deploy)

## 8. Tài liệu tham khảo

- Bootstrap Official Documentation – <https://getbootstrap.com>
- MySQL Documentation – <https://dev.mysql.com/doc/>
- GitHub Repository của nhóm – <https://github.com/honpphuc/CSE702051-Project>
- W3Schools – <https://www.w3schools.com/>
- Stack Overflow – <https://stackoverflow.com/>
- Các bài giảng, slide và tài liệu môn học Thiết kế web nâng cao trên nền tảng Coursera và giảng viên cung cấp – Trường Công nghệ thông tin Phenikaa – Đại học Phenikaa
- Tài liệu hướng dẫn UML – <https://www.uml-diagrams.org/>
- Một số nguồn tài liệu tham khảo khác từ Internet phục vụ cho quá trình học tập và phát triển hệ thống.