

TRƯỜNG ĐẠI HỌC TRÀ VINH
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO MÔN HỌC
CÔNG NGHỆ PHẦN MỀM
(MSHP: 220055)

TÊN ĐỀ TÀI
XÂY DỰNG HỆ THỐNG ĐẶT LỊCH SÂN

Giáo viên hướng dẫn:

TS. Nguyễn Bảo Ân

Sinh viên thực hiện:

Lê Khánh Đăng - 110122047

Châu Gia Bảo - 110122034

Nguyễn Tấn Lợi - 110122014

Lớp: DA22TTA

Trà Vinh, tháng 7 năm 2025

[illegible]

Trà Vinh, ngày tháng năm
Giảng viên hướng dẫn
(Ký tên và ghi rõ họ tên)

[illegible]

Giảng viên phản biện
(Ký tên và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1 GIỚI THIỆU	1
1.1. Tên đề tài	1
1.2. Mục tiêu của ứng dụng	1
1.2.1. Cung cấp nền tảng đặt sân trực tuyến dễ sử dụng	2
1.2.2. Quản lý hoạt động sân bóng một cách hiện đại và trực quan.....	2
1.2.3. Giảm thời gian và công sức trong quản lý thủ công.....	2
1.2.4. Sẵn sàng tích hợp các dịch vụ mở rộng.....	2
1.3. Lý do chọn đề tài	2
1.4. Tính cần thiết của việc áp dụng phần mềm phân tán, kiến trúc Microservices và Cloud Computing	3
CHƯƠNG 2 PHÂN TÍCH YÊU CẦU	5
2.1. Các chức năng chính của hệ thống	5
2.1.1. Quản lý tài khoản người dùng	5
2.1.2. Đặt sân trực tuyến.....	5
2.1.3. Quản lý sân bóng	5
2.1.4. Quản lý lịch đặt sân	5
2.1.5. Quản lý thanh toán.....	5
2.1.6. Tìm kiếm & lọc dữ liệu	5
2.2. Các yêu cầu phi chức năng	5
2.2.1. Hiệu suất (Performance).....	5
2.2.2. Khả năng mở rộng (Scalability)	6
2.2.3. Tính bảo mật (Security).....	6
2.2.4. Khả năng duy trì và bảo trì (Maintainability).....	6
2.2.5. Tính khả dụng (Availability)	6
2.2.6. Tính di động (Portability).....	6
2.2.7. Giao diện người dùng (Usability).....	6
CHƯƠNG 3 THIẾT KẾ HỆ THỐNG	7
3.1. Kiến trúc tổng thể	7
3.2. Thiết kế cơ sở dữ liệu	8
3.2.1. Mô hình cơ sở dữ liệu.....	8
3.2.2. Các bảng (collections) chính	9
3.3. Thiết kế API.....	11

3.3.1. Xác thực người dùng (auth).....	11
3.3.1. Quản lý người dùng (Yêu cầu xác thực)	11
3.3.2. Quản lý sân (Yêu cầu xác thực).....	12
3.3.3. Đặt lịch sân (Yêu cầu xác thực).....	12
3.3.4. Cấu trúc Request/Reponse.....	13
3.4. Thiết kế giao diện (UI/UX)	16
3.4.1. Giao diện Figma	16
3.4.2. Giao diện trang web.....	22
CHƯƠNG 4 TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG.....	26
4.1. Công nghệ và thư viện đã sử dụng	26
4.2. Cấu hình Docker và quy trình triển khai	27
4.2.1. Cấu trúc dockerfile backend.....	27
4.2.2. Cấu trúc dockerfile client	27
4.2.3. Docker Compose	28
4.3. Triển khai trên Vercel và Render	29
CHƯƠNG 5 QUẢN LÝ DỰ ÁN	30
5.1. Sử dụng Jira để lập kế hoạch và theo dõi tiến độ	30
5.2. Phân công nhiệm vụ của từng thành viên trong nhóm.	30
CHƯƠNG 6 KIỂM THỬ	32
6.1. Chiến lược kiểm thử	32
6.2. Công cụ sử dụng	32
6.3. Kết quả kiểm thử API với Postman	32
CHƯƠNG 7 ĐÁNH GIÁ VÀ KẾT LUẬN.....	38
7.1. Những khó khăn gặp phải trong quá trình thực hiện.....	38
7.2. Bài học rút ra	38
7.3. Đề xuất cải thiện trong tương lai	38
TÀI LIỆU THAM KHẢO	39

DANH MỤC HÌNH

Hình 3.1.Sơ đồ kiến trúc client - server.....	7
Hình 3.2.Mô hình cơ sở dữ liệu.....	8
Hình 3.3.Giao diện đăng nhập/ đăng ký	17
Hình 3.4.Giao diện trang chủ	17
Hình 3.5.Giao diện danh sách sân	18
Hình 3.6.Giao diện đặt sân	19
Hình 3.7.Giao diện danh sách sân đã đặt.....	20
Hình 3.8.Giao diện cập nhật thông tin người dùng	20
Hình 3.9.Giao diện quản lý khách hàng	20
Hình 3.10.Giao diện quản lý sân	21
Hình 3.11.Giao diện trang chủ.....	22
Hình 3.12.Giao diện đặt sân	23
Hình 3.13.Giao diện xem danh sách sân	23
Hình 3.14.Giao diện xem danh sách sân đã đặt.....	24
Hình 3.15.Giao diện cập nhật thông tin.....	24
Hình 3.16.Giao diện quản lý khách hàng	24
Hình 3.17.Giao diện quản lý sân	25
Hình 3.18.Giao diện quản lý đặt sân	25
Hình 6.1.Kiểm thử chức năng đăng ký.....	32
Hình 6.2.Kiểm thử chức năng đăng nhập.....	33
Hình 6.3.Chức năng xem thông tin người dùng.....	33
Hình 6.4.Kiểm thử chức năng lấy danh sách sân	34
Hình 6.5.Kiểm thử chức năng đặt sân	34
Hình 6.6.Kiểm thử chức năng admin xác nhận đặt sân và thanh toán	35
Hình 6.7.Kiểm thử chức năng xem danh sách sân đã đặt.....	35
Hình 6.8.Kiểm thử chức năng thêm sân bóng.....	36
Hình 6.9.Kiểm thử chức năng sửa thông tin sân	36
Hình 6.10.Kiểm thử chức năng cập nhật thông tin người dùng	37

DANH MỤC BẢNG

Bảng 3.1.Bảng người dùng (User).....	9
Bảng 3.2.Bảng sân bóng (SanBong).....	10
Bảng 3.3.Bảng đặt sân (DatSan).....	11
Bảng 3.4.Xác thực người dùng.....	11
Bảng 3.5.Quản lý người dùng	12
Bảng 3.6.Quản lý sân.....	12
Bảng 3.7.Quản lý đặt lịch	12
Bảng 3.8.Cấu trúc Request/Reponse	16
Bảng 4.1.Công cụ và thư viện đã sử dụng.....	27
Bảng 4.2.Triển khai trên các nền tảng	29
Bảng 5.1. Phân công nhiệm vụ.....	31

LỜI CẢM ƠN

Trước hết, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Bảo Ân đã tận tình hướng dẫn và truyền đạt những kiến thức quý báu trong suốt quá trình học tập. Sự nhiệt huyết, tận tâm và những bài giảng đầy cảm hứng của thầy đã giúp chúng em không chỉ hoàn thành tốt bài báo cáo này mà còn mở rộng tầm hiểu biết và niềm đam mê với môn học.

Bên cạnh đó, chúng em xin cảm ơn các bạn trong lớp đã luôn đồng hành, hỗ trợ và chia sẻ kinh nghiệm học tập. Những buổi thảo luận nhóm, những giờ học tập chung và sự giúp đỡ lẫn nhau đã tạo nên một môi trường học tập tích cực và hiệu quả, giúp chúng em hoàn thành tốt bài báo cáo này.

Cuối cùng, chúng em xin gửi lời cảm ơn đến gia đình và bạn bè đã luôn động viên, khích lệ và tạo điều kiện tốt nhất để chúng em có thể tập trung vào việc học tập và hoàn thành bài báo cáo này. Sự ủng hộ và tình yêu thương của mọi người là nguồn động lực to lớn giúp chúng em vượt qua mọi khó khăn và thử thách.

Một lần nữa, chúng em xin chân thành cảm ơn tất cả mọi người đã giúp đỡ và ủng hộ Chúng em trong suốt quá trình học tập và hoàn thành bài báo cáo này.

CHƯƠNG 1 GIỚI THIỆU

1.1. Tên đề tài

Tên dự án: “Xây dựng hệ thống đặt lịch sân”

Chủ đề: Với sự phát triển nhanh chóng của thể thao phong trào, nhu cầu đặt sân bóng đá mini ngày càng tăng. Tuy nhiên, hầu hết các cơ sở cho thuê sân hiện nay vẫn sử dụng hình thức quản lý truyền thống như ghi chép tay, gọi điện hoặc quản lý bằng bảng Excel, điều này dễ gây nhầm lẫn, trùng lịch hoặc khó theo dõi thông tin.

Chính vì vậy, đề tài “Xây dựng hệ thống đặt lịch sân” được chọn để giải quyết những bất cập trong quản lý và vận hành dịch vụ cho thuê sân bóng. Hệ thống cho phép người dùng dễ dàng tìm kiếm và đặt sân trực tuyến thông qua website hoặc thiết bị di động, đồng thời giúp chủ sân quản lý lịch đặt, thông tin khách hàng, doanh thu và tình trạng hoạt động của sân một cách chính xác, tiện lợi.

Hệ thống được thiết kế theo mô hình hiện đại với khả năng mở rộng, bảo mật, có thể áp dụng thực tế cho nhiều khu vực sân khác nhau hoặc triển khai như một nền tảng dịch vụ (SaaS) trong tương lai.

1.2. Mục tiêu của ứng dụng

Trong bối cảnh nhu cầu chơi bóng đá phong trào ngày càng tăng cao tại các khu đô thị, trường học, khu công nghiệp và khu dân cư, mô hình sân bóng mini (5, 7, 11 người) đang phát triển nhanh chóng. Tuy nhiên, đa số các sân bóng hiện nay vẫn quản lý hoạt động đặt sân theo phương thức thủ công như gọi điện, ghi chép tay, tin nhắn Zalo,... Điều này dẫn đến nhiều vấn đề phát sinh như:

Lịch đặt sân dễ bị trùng hoặc nhầm giờ do không có hệ thống kiểm tra tự động.

Việc ghi nhận doanh thu và thông tin khách hàng không chính xác, khó thống kê.

Không có giao diện hiển thị rõ ràng các khung giờ trống.

Chủ sân khó quản lý khi mở rộng thêm nhiều sân ở các địa điểm khác nhau.

Từ thực tế đó, nhóm đề xuất xây dựng một hệ thống website đặt sân và quản lý sân bóng đá, nhằm số hóa quy trình hoạt động và cung cấp nền tảng online hỗ trợ khách hàng lẫn chủ sân. Ứng dụng nhắm đến các mục tiêu chính sau:

1.2.1. Cung cấp nền tảng đặt sân trực tuyến dễ sử dụng

Khách hàng có thể truy cập website để tra cứu khung giờ trống, chọn ngày và đặt sân trực tiếp mà không cần gọi điện.

Giao diện thân thiện, hiển thị lịch sân theo ngày/giờ, kèm thông tin sân (loại sân, vị trí, giá thuê).

Gửi thông báo xác nhận đặt sân tự động qua email hoặc hệ thống.

1.2.2. Quản lý hoạt động sân bóng một cách hiện đại và trực quan

Chủ sân có thể thêm/sửa/xóa thông tin sân bóng, cập nhật tình trạng từng sân (còn trống, đã đặt, bảo trì).

Tự động kiểm tra trùng lịch và báo lỗi nếu có xung đột giờ.

Giao diện quản lý lịch đặt sân dưới dạng bảng hoặc lịch kéo-thả (calendar view).

1.2.3. Giảm thời gian và công sức trong quản lý thủ công

Thay thế hoàn toàn các phương pháp truyền thống như sổ tay, tin nhắn Zalo.

Hỗ trợ tìm kiếm, lọc lịch sử đặt sân theo ngày, tên khách hàng, số điện thoại.

Giảm thiểu sai sót trong quá trình thu tiền, xác nhận đặt chỗ.

1.2.4. Sẵn sàng tích hợp các dịch vụ mở rộng

Thiết kế hệ thống theo mô hình API để có thể tích hợp thêm:

Cổng thanh toán online (Momo, VNPay, ZaloPay, v.v.).

Đánh giá – xếp hạng sau mỗi lần đặt sân.

Lịch sử đặt sân cá nhân cho khách hàng.

Triển khai hệ thống trên nền tảng cloud để đảm bảo khả năng truy cập 24/7, tính ổn định và mở rộng.

1.3. Lý do chọn đề tài

Hiện nay, nhu cầu chơi thể thao, đặc biệt là bóng đá mini (5 người, 7 người) đang ngày càng tăng cao, đặc biệt ở các khu đô thị, trường học, công ty hay khu dân cư đông đúc. Sự phát triển mạnh mẽ của bộ môn thể thao vua kéo theo sự mọc lên nhanh chóng của các cụm sân bóng trên khắp cả nước. Tuy nhiên, việc vận hành và quản lý sân bóng vẫn còn mang tính thủ công, thiếu hiệu quả và dễ gây nhầm lẫn.

Cụ thể, một số vấn đề nổi bật trong việc quản lý sân bóng hiện nay bao gồm:

Quản lý lịch đặt sân rời rạc, dễ dẫn đến trùng giờ hoặc bỏ sót lịch đặt.

Ghi nhận và thu tiền thủ công, không minh bạch, gây thất thoát doanh thu.

Thiếu hệ thống theo dõi khung giờ trống, dẫn đến khó tư vấn lịch phù hợp cho khách hàng.

Không có thống kê báo cáo tài chính, gây khó khăn trong việc đánh giá hiệu quả kinh doanh.

Khách hàng không thể chủ động đặt sân online, chủ yếu phải gọi điện, nhắn tin, gây tốn thời gian.

Trong thời đại công nghệ số và xu hướng chuyển đổi số hiện nay, các hệ thống phần mềm đóng vai trò vô cùng quan trọng trong việc tối ưu hóa quy trình quản lý. Vì vậy, nhóm thực hiện đề tài nhằm giải quyết các vấn đề nêu trên bằng một giải pháp số hóa, hiện đại, thuận tiện cho cả chủ sân và khách hàng.

Ngoài ra, đề tài này cũng là cơ hội để nhóm:

Vận dụng kiến thức đã học về công nghệ phần mềm, thiết kế hệ thống, lập trình web/mobile,...

Trải nghiệm thực tế việc phát triển một hệ thống phần mềm phục vụ quản lý, từ phân tích – thiết kế – lập trình – kiểm thử – triển khai.

Tìm hiểu và áp dụng mô hình phần mềm phân tán / Microservices / Cloud, một xu hướng tất yếu trong phát triển hệ thống hiện đại, giúp tăng tính mở rộng, dễ bảo trì và triển khai linh hoạt.

1.4. Tính cần thiết của việc áp dụng phần mềm phân tán, kiến trúc Microservices và Cloud Computing

Trong bối cảnh chuyển đổi số và xu hướng kinh doanh hiện đại, việc xây dựng một hệ thống phần mềm quản lý sân bóng không chỉ cần đáp ứng các chức năng cơ bản, mà còn phải đảm bảo khả năng mở rộng, tính linh hoạt, khả năng chịu tải và khả năng duy trì lâu dài. Việc ứng dụng phần mềm phân tán, kiến trúc microservices, và điện toán đám mây (Cloud Computing) là cần thiết để đảm bảo những yêu cầu đó.

- Phần mềm phân tán (Distributed System)

Trong trường hợp có nhiều sân bóng được đặt ở các khu vực địa lý khác nhau (quận, tỉnh khác nhau), hệ thống phân tán giúp đảm bảo các cụm máy chủ xử lý độc lập nhưng vẫn đồng bộ hóa dữ liệu nhanh chóng.

Người quản lý có thể truy cập và điều khiển hệ thống tại từng sân một cách dễ dàng mà không cần tập trung toàn bộ dữ liệu và xử lý tại một điểm duy nhất.

Đảm bảo tính sẵn sàng cao, giảm thiểu rủi ro mất mát dữ liệu khi có sự cố cục bộ.

- Kiến trúc Microservices

Hệ thống được chia nhỏ thành các dịch vụ riêng biệt như:

Dịch vụ đặt sân

Dịch vụ quản lý người dùng

Dịch vụ quản lý sân

Mỗi dịch vụ có thể được phát triển, triển khai và nâng cấp độc lập, giúp giảm thiểu rủi ro ảnh hưởng toàn hệ thống khi bảo trì hoặc nâng cấp một phần.

Microservices giúp tăng tốc độ phát triển và triển khai, cũng như dễ dàng tích hợp các công nghệ mới vào từng phần của hệ thống.

- Điện toán đám mây (Cloud Computing)

Hệ thống được triển khai trên các nền tảng cloud (như AWS, Azure, Google Cloud, hoặc các cloud nội địa như VNPT Cloud, FPT Cloud...), giúp:

Tăng khả năng mở rộng: Dễ dàng đáp ứng lưu lượng lớn khi nhu cầu tăng đột biến

Tiết kiệm chi phí: Không cần đầu tư vào hạ tầng vật lý ban đầu. Chi trả tiền theo mức sử dụng.

Dữ liệu được sao lưu và đồng bộ tự động, tránh mất mát và gián đoạn dịch vụ.

Hệ thống có thể tích hợp các dịch vụ sẵn có như: gửi email, SMS, thanh toán trực tuyến.

CHƯƠNG 2 PHÂN TÍCH YÊU CẦU

2.1. Các chức năng chính của hệ thống

Hệ thống Quản lý sân bóng đá cần đáp ứng các chức năng chính sau:

2.1.1. Quản lý tài khoản người dùng

Đăng ký, đăng nhập tài khoản cho khách hàng và quản trị viên.

Phân quyền người dùng: khách hàng, chủ sân.

Quản lý thông tin cá nhân, đổi mật khẩu, đăng xuất.

2.1.2. Đặt sân trực tuyến

Cho phép khách hàng tra cứu và đặt sân theo khung giờ, ngày, loại sân (5 người, 7 người, 11 người).

Xem lịch trống của từng sân.

Đặt sân theo giờ cố định hoặc đặt định kỳ.

2.1.3. Quản lý sân bóng

Thêm/xóa/cập nhật thông tin các sân bóng: tên sân, loại sân, vị trí, giá thuê theo khung giờ.

Quản lý trạng thái sân: đang hoạt động, bảo trì, tạm ngưng.

2.1.4. Quản lý lịch đặt sân

Hiển thị lịch đặt sân.

Ghi nhận và cập nhật lịch đặt sân từ khách hàng hoặc nhân viên nhập thủ công.

2.1.5. Quản lý thanh toán

Xác nhận thanh toán khi đặt sân: chuyển khoản, ví điện tử, tiền mặt.

2.1.6. Tìm kiếm & lọc dữ liệu

Tìm kiếm sân theo tên sân, doanh mục sân.

2.2. Các yêu cầu phi chức năng

2.2.1. Hiệu suất (Performance)

Hệ thống phải xử lý nhanh và phản hồi yêu cầu người dùng dưới 3 giây.

Hệ thống chịu được tối thiểu 500 người dùng đồng thời.

2.2.2. Khả năng mở rộng (Scalability)

Thiết kế theo hướng microservices và cloud để dễ dàng thêm chức năng mới, mở rộng vùng quản lý.

Hỗ trợ đa chi nhánh, đa sân, và nhiều khu vực địa lý.

2.2.3. Tính bảo mật (Security)

Mã hóa mật khẩu người dùng (bằng bcrypt hoặc tương đương).

Xác thực người dùng bằng token (JWT).

Phân quyền rõ ràng giữa khách hàng và quản trị viên.

Bảo vệ API khỏi các cuộc tấn công như SQL Injection, XSS.

2.2.4. Khả năng duy trì và bảo trì (Maintainability)

Mã nguồn tách biệt theo từng module (user, sân, lịch, báo cáo...).

Sử dụng cấu trúc RESTful API giúp dễ bảo trì và tích hợp hệ thống bên ngoài.

2.2.5. Tính khả dụng (Availability)

Hệ thống hoạt động liên tục 24/7.

Hỗ trợ chạy đa máy chủ, có cơ chế failover (dự phòng khi lỗi).

2.2.6. Tính di động (Portability)

Có thể triển khai trên nhiều nền tảng: Windows, Linux, cloud providers như AWS, Azure, GCP.

Hỗ trợ chạy trên web, và có thể phát triển ứng dụng mobile trong tương lai.

2.2.7. Giao diện người dùng (Usability)

Giao diện thân thiện, trực quan, dễ sử dụng với cả người không rành công nghệ.

Hỗ trợ đa ngôn ngữ (nếu mở rộng quốc tế).

CHƯƠNG 3 THIẾT KẾ HỆ THỐNG

3.1. Kiến trúc tổng thể

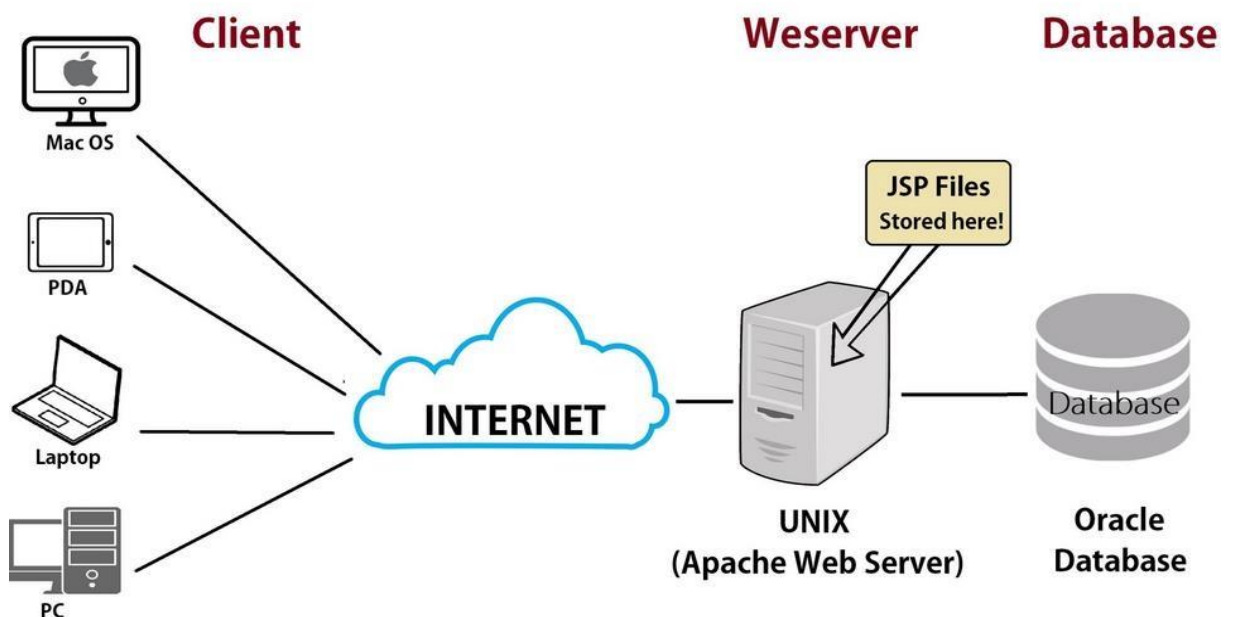
Hệ thống đặt lịch sân bóng được xây dựng theo mô hình Client – Server, bao gồm ba thành phần chính:

Client (Frontend): Ứng dụng React.js giao tiếp với người dùng, hiển thị giao diện, thực hiện các thao tác như xem sân, đặt lịch, đăng ký/đăng nhập, tra cứu lịch sử...

Server (Backend API): Node.js kết hợp Express.js xử lý logic nghiệp vụ, xác thực người dùng, quản lý dữ liệu, trả kết quả dưới dạng JSON.

Cơ sở dữ liệu (Database): MongoDB lưu trữ toàn bộ thông tin sân bóng, người dùng, lịch đặt, hóa đơn, v.v.

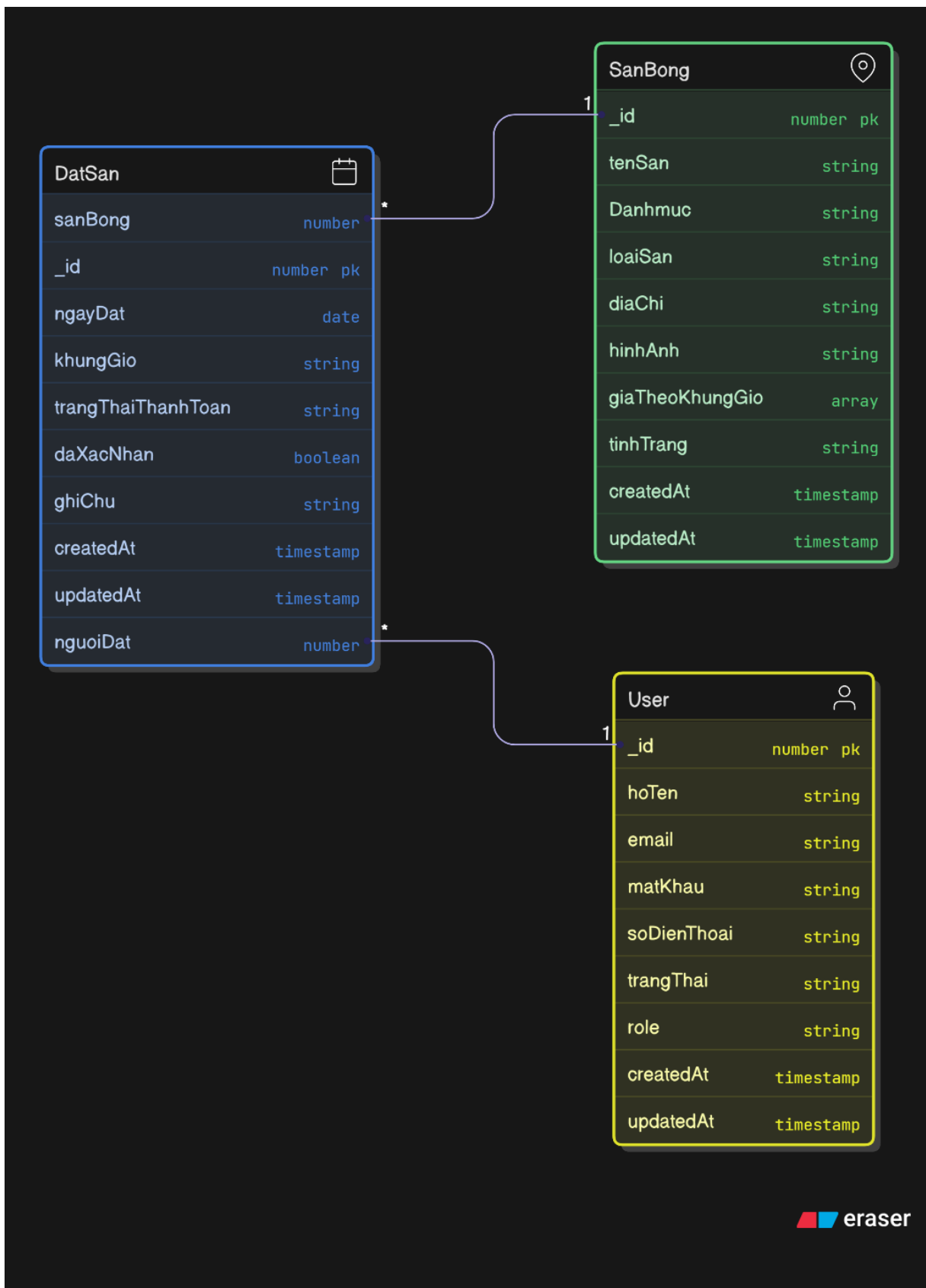
Sơ đồ kiến trúc hệ thống :



Hình 3.1.Sơ đồ kiến trúc client - server

3.2. Thiết kế cơ sở dữ liệu

3.2.1. Mô hình cơ sở dữ liệu



Hình 3.2.Mô hình cơ sở dữ liệu

3.2.2. Các bảng (collections) chính

Hệ thống đặt lịch sân bóng sử dụng cơ sở dữ liệu MongoDB với cấu trúc dữ liệu dạng tài liệu (document-based). Dưới đây là mô tả chi tiết về các mô hình dữ liệu (Schema) được sử dụng trong hệ thống:

1. Bảng người dùng (User)

Tên trường	Kiểu dữ liệu	Mô tả
_id	Number	Khóa chính, tự động tăng
hoTen	String	Họ tên người dùng
email	String	Email duy nhất, định dạng hợp lệ
matKhau	String	Mật khẩu đã mã hóa
soDienThoai	String	Số điện thoại người dùng
trangThai	String	Trạng thái tài khoản ('Hoạt động', 'Đã khóa')
role	String	Vai trò ('user', 'admin')
timestamps	Date	Thời gian tạo/cập nhật bản ghi

Bảng 3.1. Bảng người dùng (User)

2. Bảng sân bóng (SanBong)

Tên trường	Kiểu dữ liệu	Mô tả
_id	Number	Khóa chính, tự động tăng
tenSan	String	Tên sân bóng

Danhmuc	String	Tên danh mục
loaiSan	String	Loại sân ('2 người', '5 người', '7 người', '11 người')
diaChi	String	Địa chỉ sân
hinhAnh	String	Đường dẫn ảnh sân
giaTheoKhungGio	Array	Danh sách khung giờ và giá
└ khungGio	String	Khung giờ (ví dụ: 06:00-07:30)
└ gia	Number	Giá thuê
└ Trạngthai	String	Tình trạng sân trong khung giờ
trinhTrang	String	Trạng thái chung ('Đang hoạt động', 'Bảo trì')
timestamps	Date	Thời gian tạo/cập nhật bản ghi

Bảng 3.2. Bảng sân bóng (SanBong)

3. Bảng đặt sân (DatSan)

Tên trường	Kiểu dữ liệu	Mô tả
_id	Number	Khóa chính, tự động tăng
sanBong	Number	Tham chiếu tới bảng SanBong
nguoiDat	Number	Tham chiếu tới bảng User

ngayDat	Date	Ngày đặt sân
khungGio	String	Khung giờ đặt sân
trangThaiThanhToan	String	Trạng thái thanh toán ('Đã thanh toán', 'Chưa thanh toán', 'Đã hủy')
daXacNhan	Boolean	Xác nhận đặt sân
ghiChu	String	Ghi chú thêm từ người đặt
timestamps	Date	Thời gian tạo/cập nhật bản ghi

Bảng 3.3. Bảng đặt sân (DatSan)

3.3. Thiết kế API

Hệ thống backend cung cấp RESTful API, các endpoint chia theo tài nguyên:

3.3.1. Xác thực người dùng (auth)

Phương thức	Endpoint	Mô tả
POST	/auth/register	Đăng ký tài khoản
POST	/auth/login	Đăng nhập tài khoản

Bảng 3.4. Xác thực người dùng

3.3.1. Quản lý người dùng (Yêu cầu xác thực)

Phương thức	Endpoint	Mô tả
GET	/user	Lấy danh sách người dùng
PUT	/updateuser/{id}	Cập nhật người dùng (admin)
DELETE	/xoouser/{id}	Xóa người dùng (admin)

PUT	/cap-nhat-thong-tin	Cập nhật thông tin cá nhân
PUT	/doi-mat-khau	Đổi mật khẩu người dùng

Bảng 3.5.Quản lý người dùng

3.3.2. Quản lý sân (Yêu cầu xác thực)

Phương thức	Endpoint	Mô tả
GET	/danhsachsân	Lấy danh sách sân bóng
GET	/sanbong/{id}	Lấy thông tin chi tiết sân bóng
POST	/addsan	Thêm sân bóng (admin)
PUT	/suasan/{id}	Cập nhật sân bóng (admin)
DELETE	/xoasan/{id}	Xóa sân bóng (admin)

Bảng 3.6.Quản lý sân

3.3.3. Đặt lịch sân (Yêu cầu xác thực)

Phương thức	Endpoint	Mô tả
POST	/datsân	Đặt sân bóng
GET	/danhsachdadat	Lấy danh sách đặt sân theo người dùng
PUT	/ {id} /xacnhandatsân	Admin xác nhận đặt sân

Bảng 3.7.Quản lý đặt lịch

3.3.4. Cấu trúc Request/Response

Endpoint	Request Body	Response Thành Công	Response Thất Bại
POST /auth/register	json { "hoTen": "A", "email": "a@gmail.com", "matKhau": "123456", "soDienThoai": "0123456789" }	201 Created: { "message": "Đăng ký thành công" }	400: { "message": "Email đã tồn tại" }
POST /auth/login	json { "email": "a@gmail.com", "matKhau": "123456" }	200 OK: { "token": "JWT...", "user": {...} }	401: { "message": "Sai thông tin" }
GET /danhachsachsan		200 OK: { "message": "Danh sách sân bóng" }	
GET /sanbong/{id}		200 OK: { "message": "Thông tin sân bóng" }	404: { "message": "không tìm thấy sân bóng" }
POST /datsan	json { "sanBong": "123", "khungGio": "18:00 - 19:00", "ghiChu": "..." }	201: { "message": "Đặt sân thành công" }	400: { "message": "Khung giờ đã được đặt" }
GET /danhachsachdadat		200: [{ "id": "...",	

		"sanBong": "Sân A", ... }]	
POST /addsan	<pre> Json { "tenSan": "string", "Danhmuc": "string", "loaiSan": "2 người", "diaChi": "string", "hinhAnh": "string", "giaTheoKhungGio": [{ "khungGio": "string", "gia": 0, "Trangthai": "Còn sân" }], "tinhTrang": "Đang hoạt động" } </pre>	201: { "message": "Thêm sân thành công" }	403: { "message": "Không có quyền" }
PUT /cap-nhat-thong-tin	<pre> json { "hoTen": "B", "email": "b@gmail.com", "soDienThoai": "0909..." } </pre>	200: { "message": "Cập nhật thành công" }	—
PUT /suasan/{id}	<pre> Json { "tenSan": "string", "Danhmuc": "string", "loaiSan": "2 người", </pre>	200: { "message": "Cập nhật thành công" }	403: { "message": "không có quyền" }

	<pre> "diaChi": "string", "anh": "string", "giaTheoKhungGio": [{ "khungGio": "string", "gia": 0, "Trangthai": "Còn sân" }], "trinhTrang": "Đang hoạt động" } </pre>		
DELETE /xaosan/{id}		200: { "message": "Xóa sân thành công" }	403: { "message": "không có quyền" }
PUT /{id}/xacnhandatsa n	<pre> Json { "daXacNhan": true, "trangThaiThanhToan": "Đã thanh toán" } </pre>	200: { "message": "Xác nhận thành công" }	403: { "Không có quyền" }
DELETE / xoouser/{id}		200: { "message": "Xóa người dùng thành công" }	403: { "message": "Không có quyền" }

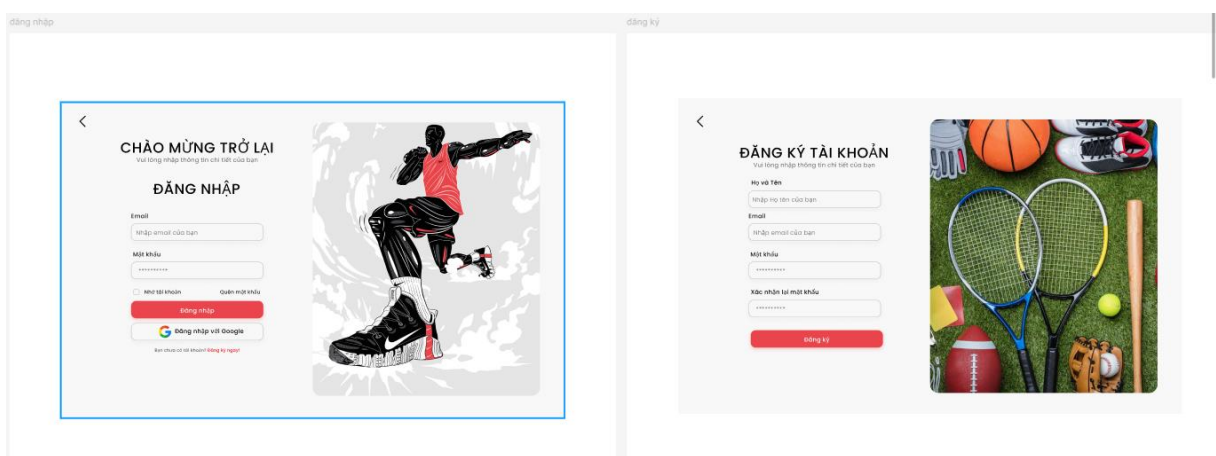
PUT /updateuser/{id}	Json { "hoTen": "string", "soDienThoai": "string", "trangThai": "string", "role": "string" }	200: { "message": "Cập nhật thành công" }	403: { "message": "Không có quyền" }
GET /user		200: { "message": "Lấy danh sách người dùng thành công" }	
PUT /doi-mat-khau	json { "matKhauCu": "123456", "matKhauMoi": "abcxyz" }	200: { "message": "Đổi mật khẩu thành công" }	—

Bảng 3.8.Cấu trúc Request/Reponse

Tài liệu Swagger : <http://localhost:5000/api-docs/>

3.4. Thiết kế giao diện (UI/UX)

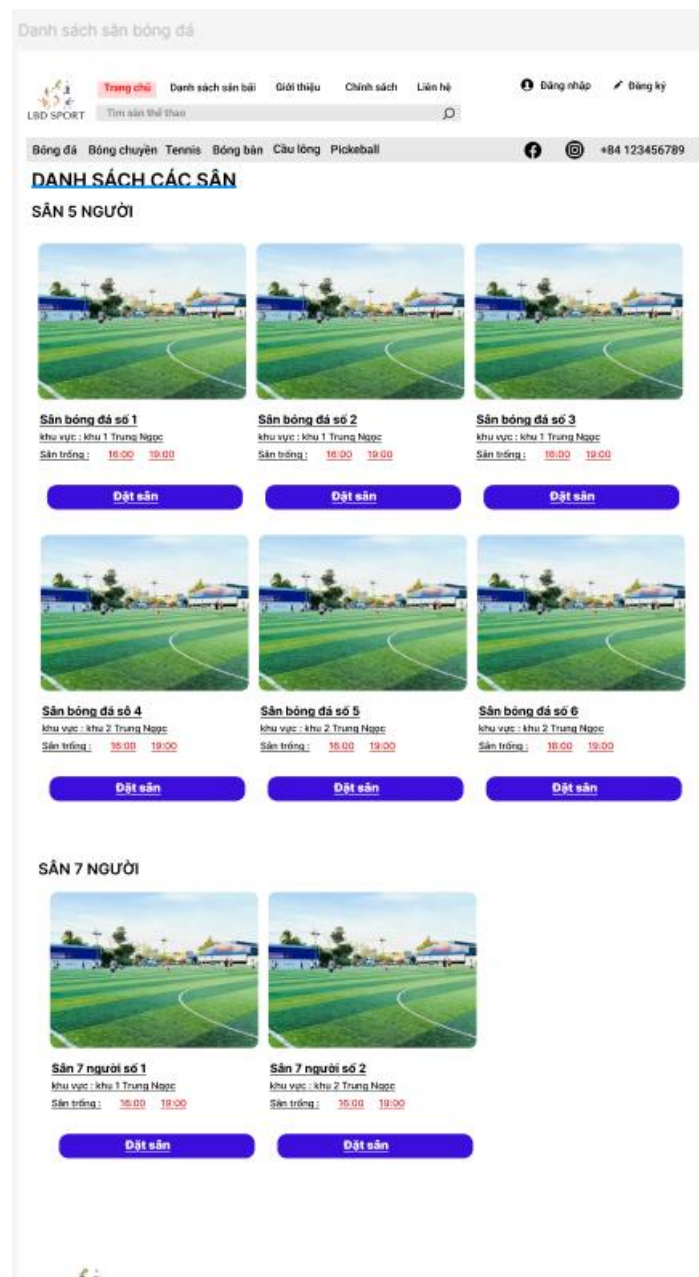
3.4.1. Giao diện Figma



Hình 3.3. Giao diện đăng nhập/ đăng ký



Hình 3.4. Giao diện trang chủ



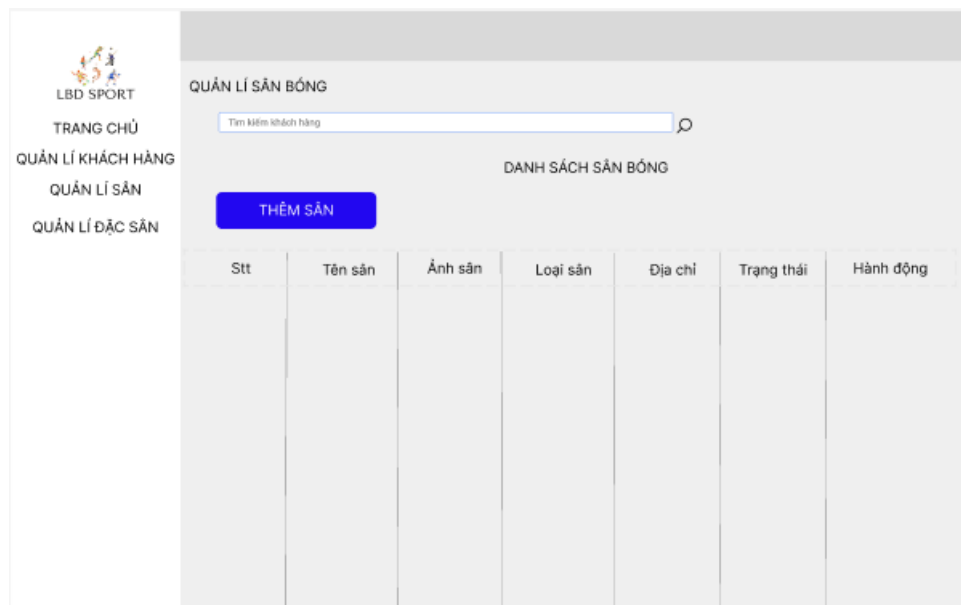
Hình 3.5. Giao diện danh sách sân

Hình 3.7. Giao diện danh sách sân đã đặt

Hình 3.8. Giao diện cập nhật thông tin người dùng

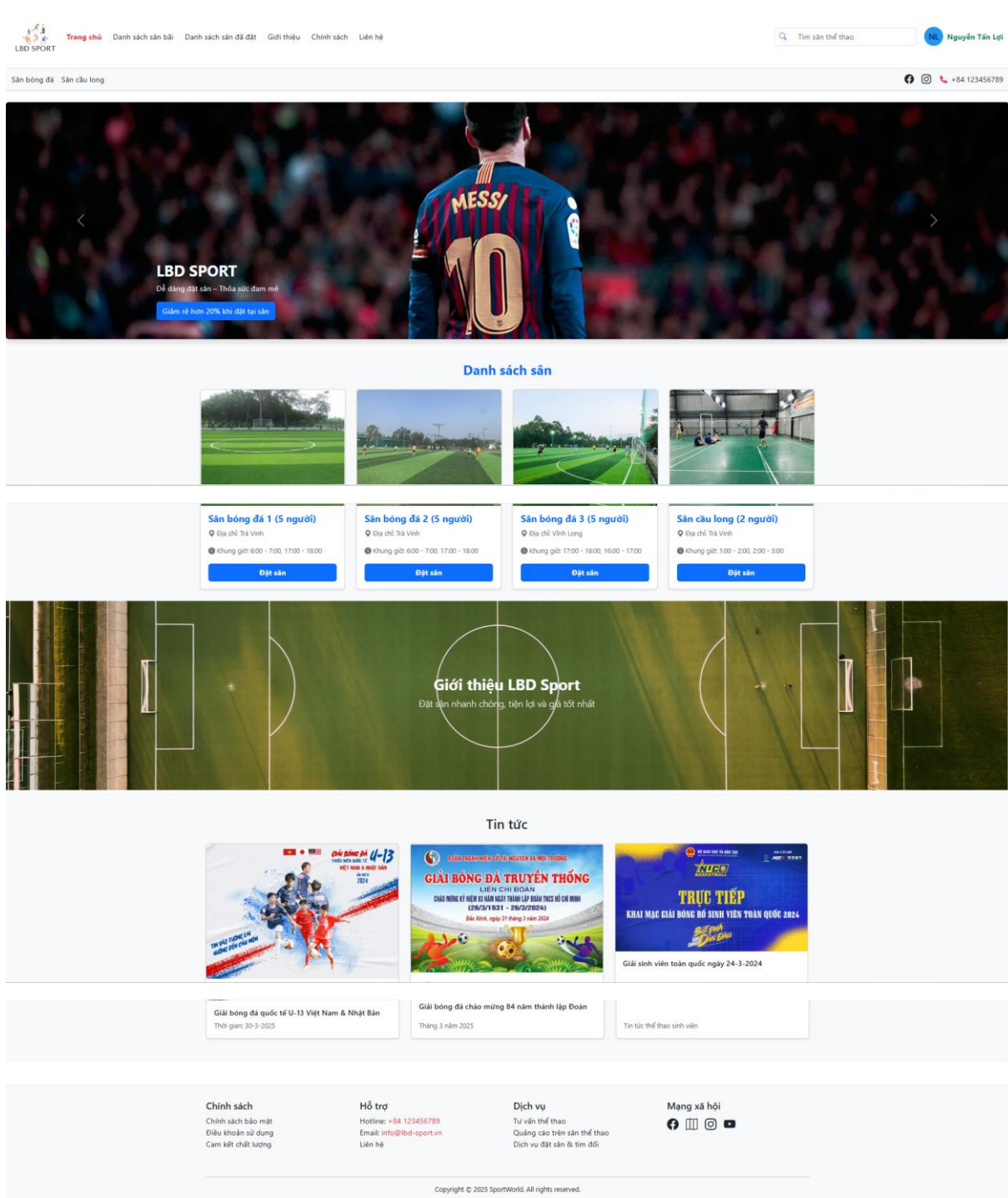
STT	Tài khoản	Số điện thoại	Địa chỉ	Thời gian	Chức năng
1	chaugiabao	0123456789	TP.Trà Vinh	27/4/2024	Xóa Khoá
2	lekhanhdang	0123456788	TP.Trà Vinh	27/4/2024	Xóa Khoá
3	nguyentanloi	0123456787	TP.Trà Vinh	27/4/2024	Xóa Khoá

Hình 3.9. Giao diện quản lý khách hàng

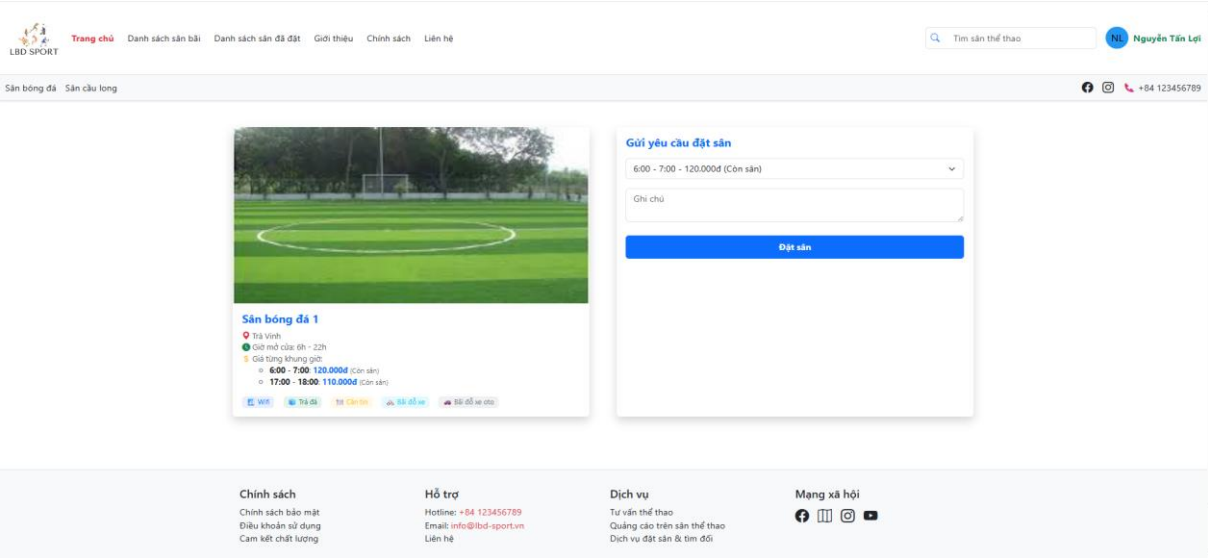


Hình 3.10. Giao diện quản lý sân

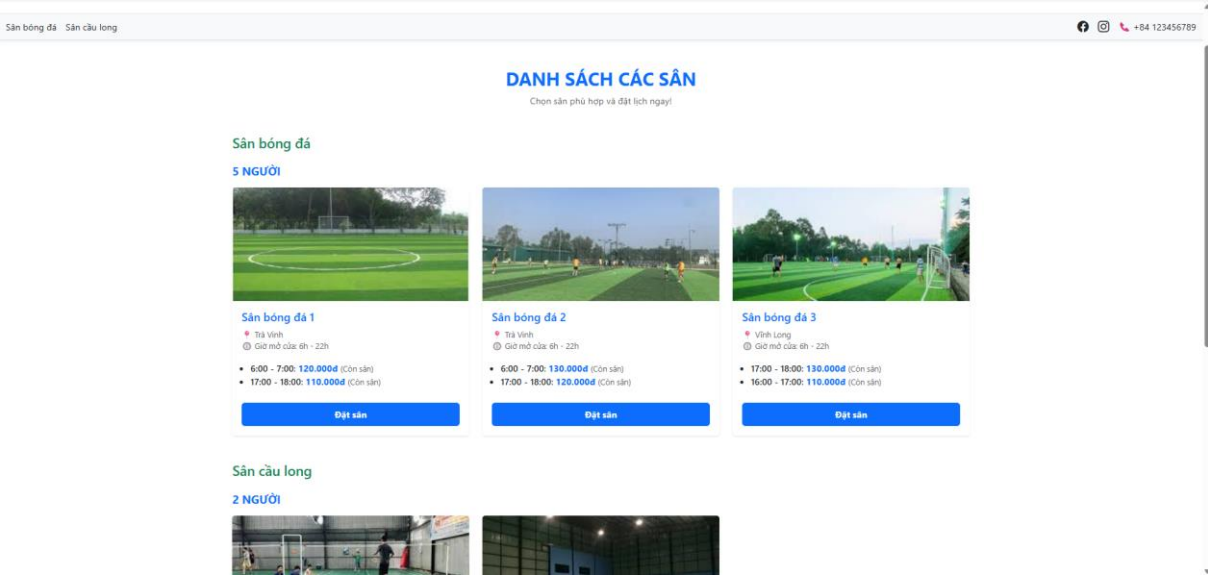
3.4.2. Giao diện trang web



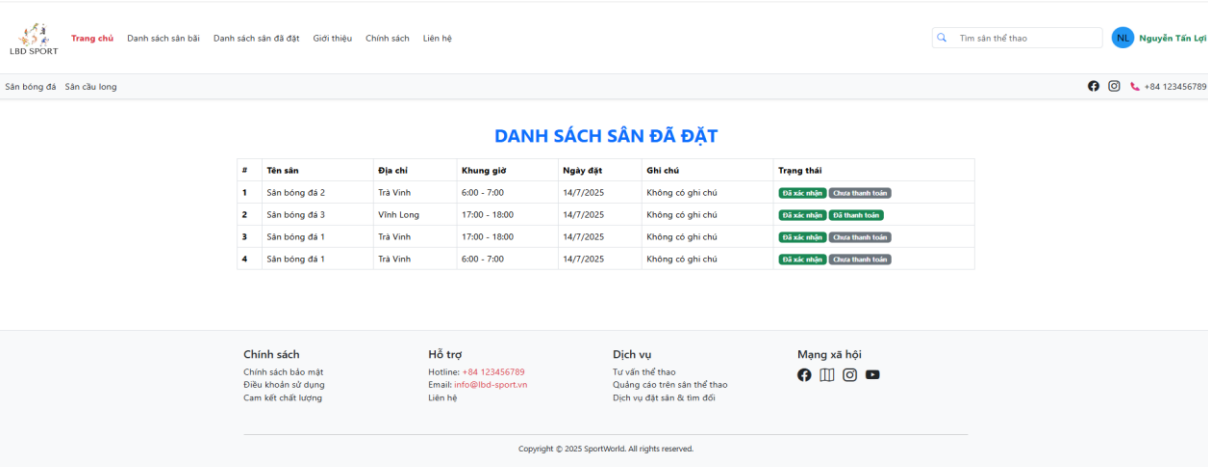
Hình 3.11. Giao diện trang chủ

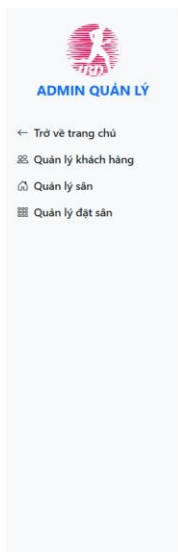


Hình 3.12.Giao diện đặt sân



Hình 3.13.Giao diện xem danh sách sân





Quản lý sân bóng

STT	ID Sân	Tên sân	Loại sân	Danh mục	Địa chỉ	Hình ảnh	Tình trạng	Chức năng
1	1	Sân bóng đá 1	5 người	Sân bóng đá	Trà Vinh		Dang hoạt động	Sửa Xóa
2	2	Sân bóng đá 2	5 người	Sân bóng đá	Trà Vinh		Dang hoạt động	Sửa Xóa
3	3	Sân bóng đá 3	5 người	Sân bóng đá	Vĩnh Long		Dang hoạt động	Sửa Xóa
4	4	Sân cầu long	2 người	Sân cầu long	Trà Vinh		Dang hoạt động	Sửa Xóa
5	5	Sân cầu long	2 người	Sân cầu long	Trà Vinh		Dang hoạt động	Sửa Xóa

Hình 3.17. Giao diện quản lý sân



Quản lý đặt sân

STT	Tên sân	Loại sân	Mã khách	Địa chỉ sân	Khung giờ	Ngày đặt	Ghi chú	Trạng thái	thanh toán	Chức năng
1	Sân bóng đá 2	5 người	Test test@test.com 000000000000	Trà Vinh	17:00 - 18:00	17/7/2025	Không có ghi chú	Chờ xác nhận	Chưa thanh toán	Xác nhận
2	Sân bóng đá 3	5 người	Test test@test.com 000000000000	Vĩnh Long	17:00 - 18:00	17/7/2025	Không có ghi chú	Chờ xác nhận	Chưa thanh toán	Xác nhận
3	Sân bóng đá 2	5 người	Test test@test.com 000000000000	Trà Vinh	6:00 - 7:00	17/7/2025	Không có ghi chú	Chờ xác nhận	Chưa thanh toán	Xác nhận
4	Sân bóng đá 1	5 người	Test test@test.com 000000000000	Trà Vinh	6:00 - 7:00	17/7/2025	Không có ghi chú	Chờ xác nhận	Chưa thanh toán	Xác nhận
5	Sân bóng đá 1	5 người	Test test@test.com 000000000000	Trà Vinh	17:00 - 18:00	17/7/2025	Không có ghi chú	Chờ xác nhận	Chưa thanh toán	Xác nhận
6	Sân cầu long	2 người	Test test@test.com 000000000000	Trà Vinh	1:00 - 2:00	17/7/2025	Không có ghi chú	Chờ xác nhận	Chưa thanh toán	Xác nhận
7	Sân cầu long	2 người	Test test@test.com 000000000000	Trà Vinh	2:00 - 3:00	17/7/2025	Không có ghi chú	Chờ xác nhận	Chưa thanh toán	Xác nhận

Hình 3.18. Giao diện quản lý đặt sân

CHƯƠNG 4 TRIỂN KHAI VÀ CÔNG NGHỆ SỬ DỤNG

4.1. Công nghệ và thư viện đã sử dụng

Dự án được xây dựng trên kiến trúc client-server, với các công nghệ hiện đại nhằm đảm bảo tính mở rộng, hiệu suất và dễ dàng bảo trì:

Thành phần	Công nghệ sử dụng	Mô tả
Frontend	ReactJS	Thư viện JavaScript xây dựng giao diện web linh hoạt và hiện đại.
Backend	Node.js + Express.js	Nền tảng máy chủ cho API RESTful.
Cơ sở dữ liệu	MongoDB	Cơ sở dữ liệu NoSQL, phù hợp với dữ liệu động và không có cấu trúc cố định.
Giao tiếp HTTP	Axios	Gửi và nhận request từ client đến backend.
Quản lý JWT	jsonwebtoken	Thư viện tạo và xác thực token đăng nhập.
Phân quyền	middleware tự viết	Kiểm tra quyền truy cập (admin, user).
Giao diện đẹp	Bootstrap, React Icons , React bootstrap	Tăng tốc thiết kế giao diện.
Triển khai FE	Vercel	Triển khai React frontend miễn phí, dễ cấu hình.
Triển khai BE	Render	Deploy backend NodeJS kèm MongoDB Atlas.

Quản lý mã nguồn	GitHub	Lưu trữ mã nguồn và phối hợp làm việc nhóm.
CI/CD	GitHub Actions	Tự động build & deploy sau mỗi lần đẩy code.
Đóng gói	Docker	Đóng gói toàn bộ hệ thống frontend/backend để triển khai.

Bảng 4.1.Công cụ và thư viện đã sử dụng

4.2. Cấu hình Docker và quy trình triển khai

4.2.1. Cấu trúc dockerfile backend

```
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 5000
CMD ["node", "app.js"]
```

4.2.2. Cấu trúc dockerfile client

```
FROM node:18
WORKDIR /app
COPY package*.json ./
RUN npm install
COPY . .
EXPOSE 3000
CMD ["npm", "start"]
```

4.2.3. Docker Compose

version: '3'

services:

backend:

build: ./backend

ports:

- "5000:5000"

volumes:

- ./backend:/app

- /app/node_modules # ngăn node_modules từ máy host ghi đè

env_file:

- ./backend/.env

restart: always

client:

build: ./client

ports:

- "3000:3000"

volumes:

- ./client:/app

restart: always

Lợi ích của Docker

Triển khai nhanh mà không phụ thuộc môi trường.

Có thể dùng cho local dev, staging hoặc production.

4.3. Triển khai trên Vercel và Render

Thành phần	Nền tảng	Mô tả
Frontend	Vercel	Tự động triển khai từ GitHub, hỗ trợ HTTPS, CDN và rollback.
Backend	Render.com	Dễ dàng deploy NodeJS với MongoDB, tự động restart khi có lỗi.
Database	MongoDB Atlas	Lưu trữ trên cloud, dễ tích hợp với NodeJS.

Bảng 4.2. Triển khai trên các nền tảng

Quy trình triển khai cụ thể

- Frontend:

Push code lên GitHub.

Vercel nhận thay đổi, build và deploy frontend tự động.

- Backend:

Push code backend lên GitHub.

Render theo dõi repo, tự động deploy server Node.js.

- Database:

Cấu hình MONGO_URI trỏ đến MongoDB Atlas trong .env hoặc Render.

CHƯƠNG 5 QUẢN LÝ DỰ ÁN

5.1. Sử dụng Jira để lập kế hoạch và theo dõi tiến độ

Trong quá trình phát triển hệ thống, nhóm sử dụng Jira Software – một công cụ quản lý dự án theo mô hình Agile/Scrum nhằm đảm bảo theo dõi sát tiến độ và phân công công việc rõ ràng. Cụ thể:

Lập kế hoạch Sprint: Dự án được chia thành nhiều Sprint kéo dài 1–2 tuần, mỗi Sprint có mục tiêu rõ ràng.

Tạo và quản lý Issue:

Các công việc được tạo dưới dạng Tasks hoặc Bugs.

Mỗi Task có mô tả chi tiết, thời hạn và người phụ trách.

Theo dõi tiến độ với Kanban board:

Các trạng thái công việc: To Do, In Progress, Code Review, Done.

Giúp giảng viên/nhóm trưởng dễ dàng kiểm tra tiến độ thực tế.

Báo cáo: Jira cung cấp biểu đồ Burndown Chart, Velocity Chart để đánh giá năng suất nhóm.

5.2. Phân công nhiệm vụ của từng thành viên trong nhóm.

Dự án được chia thành 3 vai trò chính tương ứng với 3 thành viên: Frontend, Backend, và Quản lý dự án. Mỗi thành viên đảm nhận các nhiệm vụ cụ thể nhằm đảm bảo hiệu quả phát triển và kiểm thử phần mềm.

Thành viên	Vai trò	Nhiệm vụ đảm nhận
Lê Khánh Đăng	Backend Developer	<ul style="list-style-type: none">- Thiết kế và tối ưu cơ sở dữ liệu (SCRUM-13)- Xử lý đăng ký, đăng nhập (SCRUM-2)- Xử lý trạng thái và logic đặt sân (SCRUM-10, SCRUM-8)

		<ul style="list-style-type: none"> - Tối ưu hóa hiệu suất API (SCRUM-12) - Xử lý đổi mật khẩu (SCRUM-18)
Nguyễn Tấn Lợi	Frontend Developer	<ul style="list-style-type: none"> - Tạo giao diện đăng ký, đăng nhập (SCRUM-1) - Giao diện đặt sân (SCRUM-7) - Logic đặt sân phía client (SCRUM-8) - Hiển thị thông tin tài khoản (SCRUM-16) - Bảo trì và sửa lỗi (SCRUM-15)
Châu Gia Bảo	Scrum Master / Tester / Frontend Support	<ul style="list-style-type: none"> - Quản lý toàn bộ tiến độ dự án trên Jira (SCRUM-21, SCRUM-22) - Thiết kế giao diện tổng quan (SCRUM-17) - Giao diện cập nhật trạng thái sân (SCRUM-9) - Hiển thị lịch sử đặt sân (SCRUM-6) - Tích hợp Docker và Swagger (SCRUM-11) - Kiểm thử tích hợp (SCRUM-14)

Bảng 5.1. Phân công nhiệm vụ

CHƯƠNG 6 KIỂM THỬ

6.1. Chiến lược kiểm thử

Nhằm đảm bảo hệ thống hoạt động đúng chức năng, bảo mật và ổn định, dự án đã thực hiện kiểm thử theo hai cấp độ:

Kiểm thử API (Integration Test) : Đảm bảo các endpoint hoạt động đúng logic và trả về dữ liệu chính xác.

Kiểm thử đơn vị (Unit Test) : Kiểm tra các hàm xử lý riêng lẻ như xác thực dữ liệu, tính toán, token.

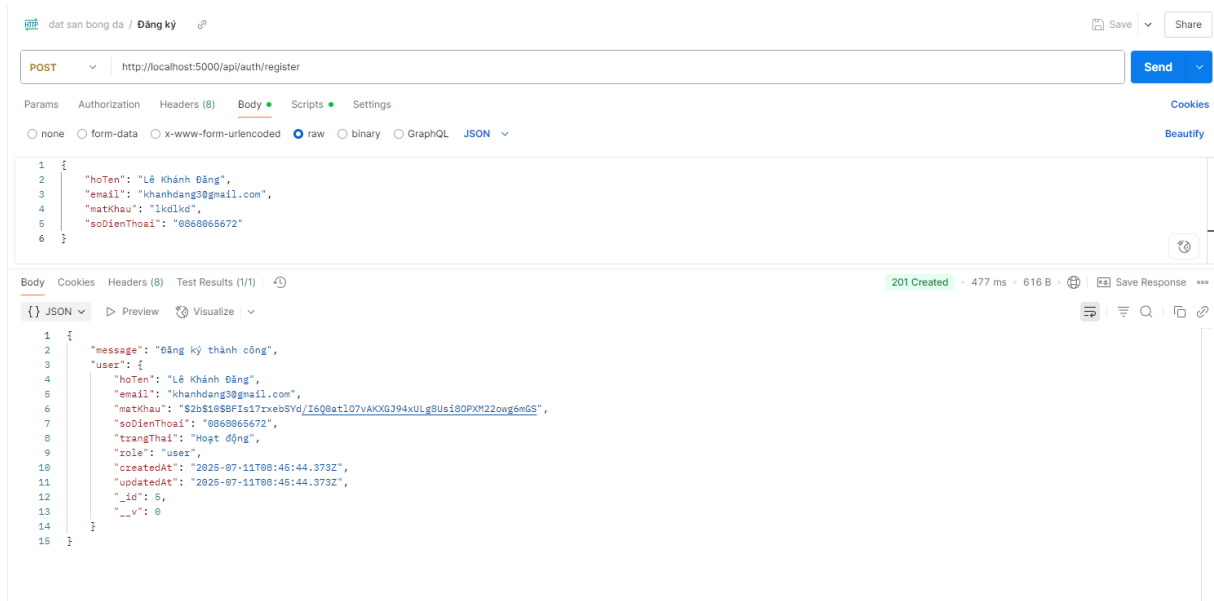
Kiểm thử thủ công giao diện : Trực tiếp thao tác trên giao diện React để phát hiện lỗi UI/UX.

6.2. Công cụ sử dụng

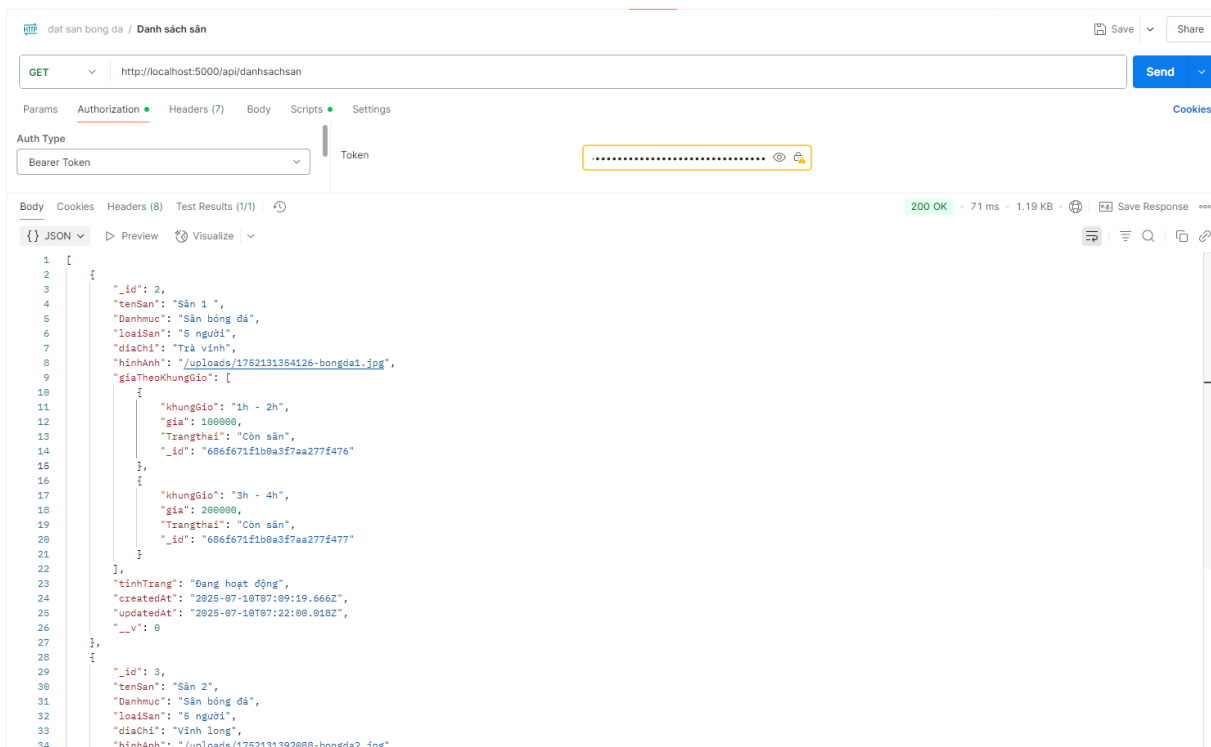
- Postman : Kiểm thử thủ công các endpoint API REST (POST/GET/PUT/DELETE).

- Browser DevTools : Kiểm thử phản hồi từ frontend, kiểm tra lỗi JavaScript.

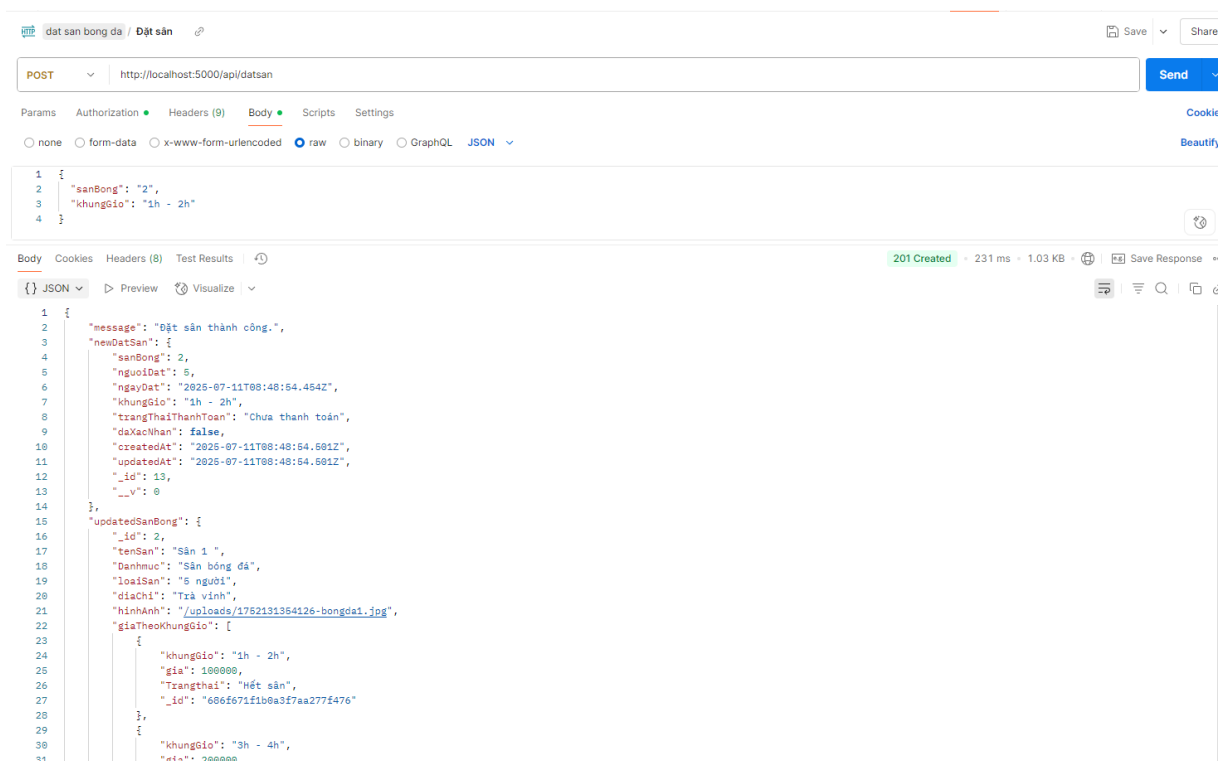
6.3. Kết quả kiểm thử API với Postman



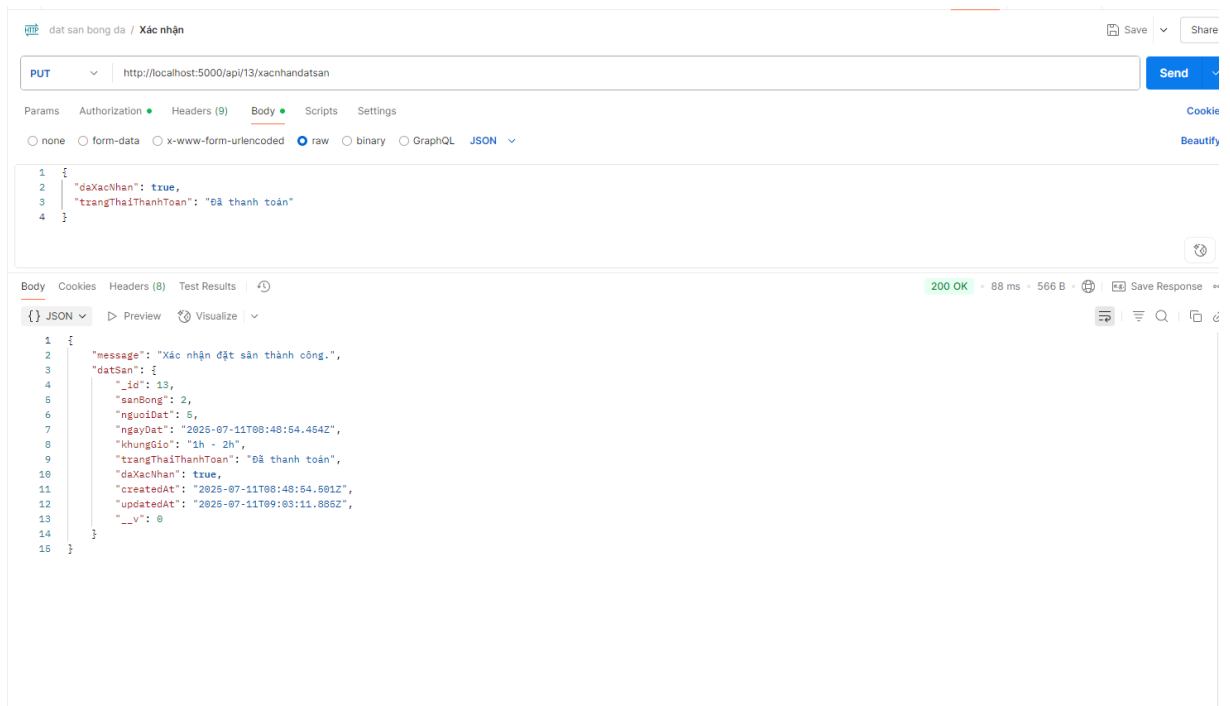
Hình 6.1.Kiểm thử chức năng đăng ký



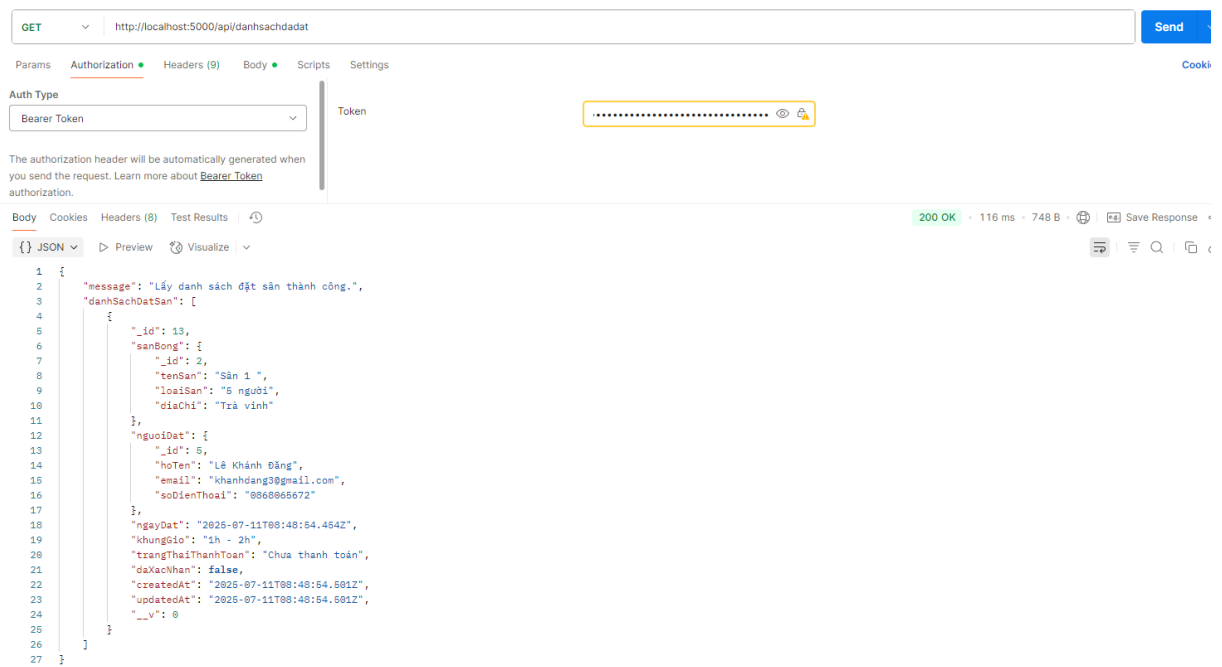
Hình 6.4.Kiểm thử chức năng lấy danh sách sân



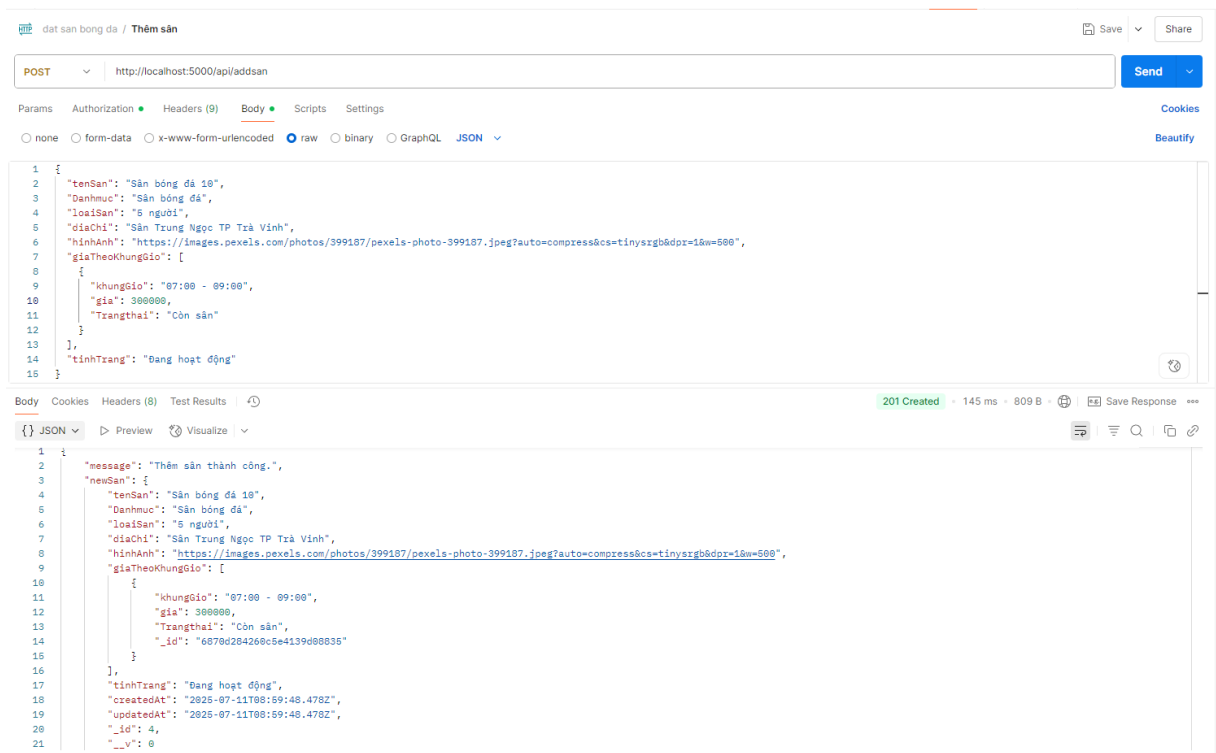
Hình 6.5.Kiểm thử chức năng đặt sân



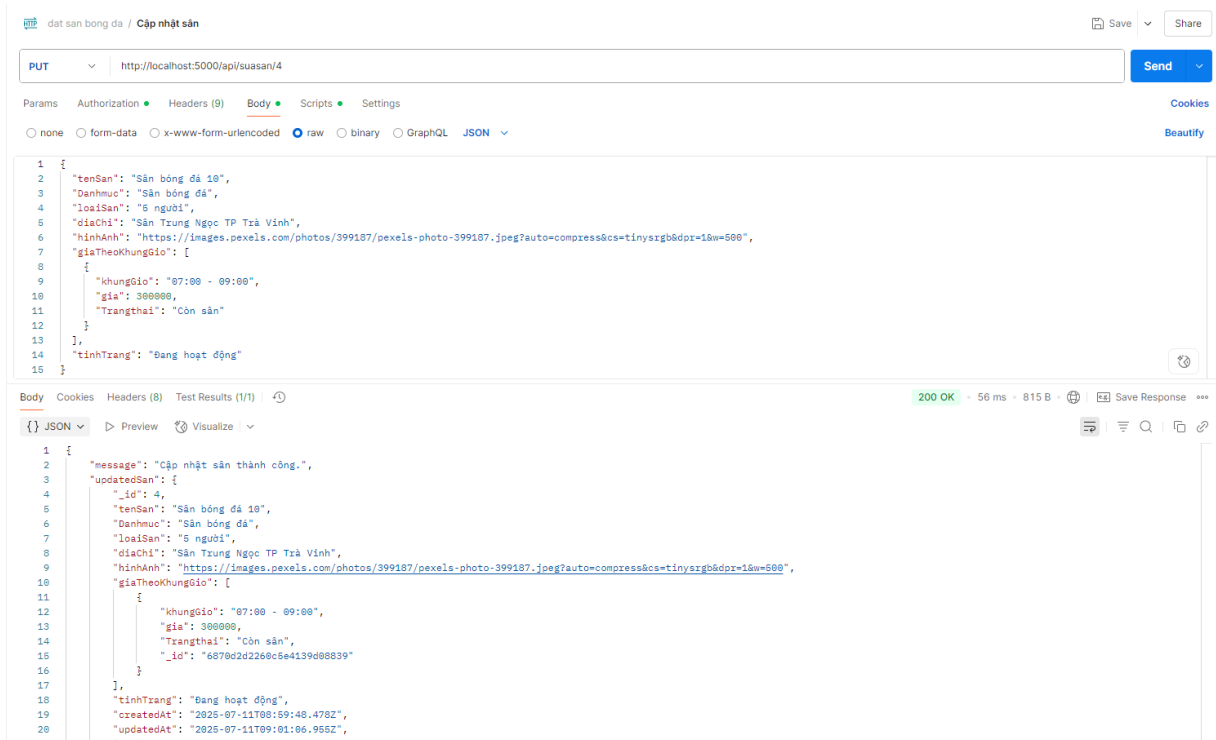
Hình 6.6.Kiểm thử chức năng admin xác nhận đặt sân và thanh toán



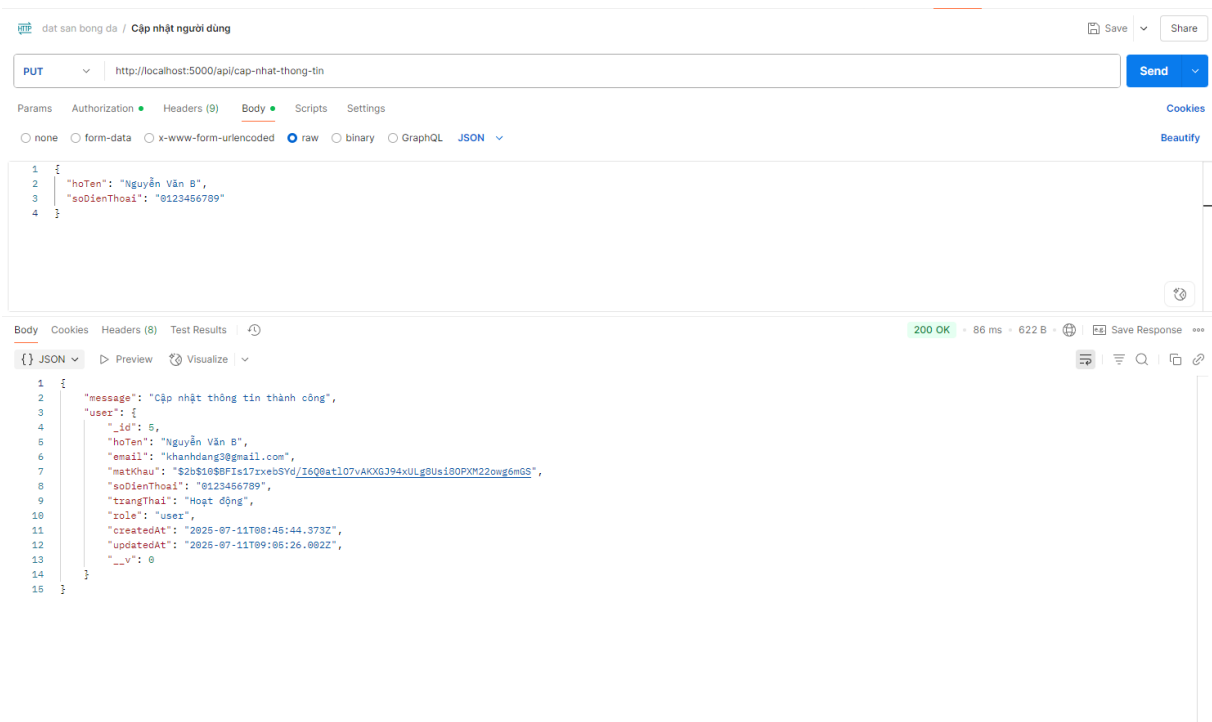
Hình 6.7.Kiểm thử chức năng xem danh sách sân đã đặt



Hình 6.8.Kiểm thử chức năng thêm sân bóng



Hình 6.9.Kiểm thử chức năng sửa thông tin sân



Hình 6.10.Kiểm thử chức năng cập nhật thông tin người dùng

CHƯƠNG 7 ĐÁNH GIÁ VÀ KẾT LUẬN

7.1. Những khó khăn gặp phải trong quá trình thực hiện

Trong quá trình xây dựng hệ thống phần mềm Quản lý sân, nhóm đã gặp phải một số khó khăn như sau:

Phân tích nghiệp vụ thực tế: Việc tìm hiểu quy trình đặt sân, thanh toán, quản lý lịch trống của các sân bóng thực tế đòi hỏi nhiều thời gian khảo sát và tổng hợp. Mỗi sân có quy trình khác nhau nên việc chuẩn hóa là một thách thức.

Thiết kế giao diện người dùng: Do đối tượng người dùng đa phần là chủ sân không chuyên về công nghệ, việc thiết kế giao diện sao cho đơn giản, dễ sử dụng nhưng vẫn đầy đủ chức năng là một yêu cầu phức tạp.

Khả năng mở rộng và tích hợp: Việc lựa chọn kiến trúc phù hợp để dễ dàng mở rộng trong tương lai, chẳng hạn như thêm tính năng thanh toán online, thông báo SMS, kết nối nhiều sân ở nhiều khu vực.

7.2. Bài học rút ra

Tầm quan trọng của phân tích yêu cầu: Việc phân tích kỹ lưỡng nhu cầu của người dùng ngay từ đầu sẽ giúp tiết kiệm thời gian và tránh sửa đổi nhiều lần về sau.

Làm việc nhóm hiệu quả: Phân chia công việc hợp lý, cập nhật tiến độ thường xuyên giúp nhóm làm việc hiệu quả hơn, hạn chế chậm trễ.

Áp dụng công nghệ phù hợp: Việc lựa chọn ReactJS, Node.js và MongoDB đã giúp hệ thống linh hoạt, dễ mở rộng và triển khai nhanh chóng.

Thực hành kỹ năng quản lý dự án: Việc sử dụng các công cụ như Jira giúp theo dõi tiến độ, kiểm soát công việc và cải thiện năng suất làm việc nhóm.

7.3. Đề xuất cải thiện trong tương lai

Phát triển ứng dụng mobile để người dùng có thể đặt sân, theo dõi lịch nhanh chóng qua điện thoại.

Tích hợp thanh toán điện tử (VNPay, Momo, ZaloPay) giúp việc thanh toán tiện lợi hơn.

Hệ thống thông báo (email/SMS) tự động nhắc lịch đặt sân, tránh bỏ lỡ lịch hẹn.

Bổ sung AI phân tích dữ liệu nhằm dự đoán thời gian cao điểm, tối ưu doanh thu và gợi ý gói khuyến mãi.

TÀI LIỆU THAM KHẢO

- [1] Ian Sommerville's , 2005, Software Engineering; 7th Ed.,
- [2] Roger S. Pressman (dịch: Ngô Trung Việt), 1997, Kỹ nghệ phần mềm, Tập I,II,III, NXB Giáo dục.
- [3] Lê Đức Trung, 2001, Công nghệ phần mềm, NXB Khoa học và Kỹ thuật,.
- [4] Ngô Trung Việt, Nguyễn Kim ánh (biên soạn), 2003, Nhập môn Công nghệ phần mềm, NXB Khoa học và kỹ thuật.
- [5] Stephen R. Schach, 1999, Classical and Objecture Oriented Software Engineering with UML and C++, 4th ed., McGraw-Hill.
- [MCG91] McLaughli,R., Một số chú ý về thiết kế chương trình, Software Engineering Notes,vol.16, no.4,oct 1991, pp53-54.
- [SHA95a] Shaw,M and D.Garlan, Formulation and formalisms in software achitecture, volume 100-lecture Notes in computer Science, Springer-verlag,1995.