

TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG  
KHOA CÔNG NGHỆ THÔNG TIN



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI:**  
**XÂY DỰNG WEBSITE QUẢN LÝ PHÒNG GYM**  
**THÔNG MINH**

Người hướng dẫn: **TS. LÊ VĂN MINH**  
Sinh viên thực hiện: **NGÔ TẤN HIẾU**  
Mã số sinh viên: **2151220082**  
Lớp: **21CT2**

**Đà Nẵng, 1/2025**

**TRƯỜNG ĐẠI HỌC KIẾN TRÚC ĐÀ NẴNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**  
**NGÀNH: CÔNG NGHỆ THÔNG TIN**

**ĐỀ TÀI:**  
**XÂY DỰNG WEBSITE QUẢN LÝ PHÒNG GYM**  
**THÔNG MINH**

Người hướng dẫn: **TS. LÊ VĂN MINH**  
Sinh viên thực hiện: **NGÔ TẤN HIẾU**  
Mã số sinh viên: **2151220082**  
Lớp: **21CT2**

**Đà Nẵng, 1/2025**

## TÓM TẮT

Tên đề tài: Xây dựng Website quản lý phòng Gym thông minh

Sinh viên thực hiện: Ngô Tấn Hiếu

Mã số sinh viên: 2151220082

Lớp: 21CT2

Đề tài “**Xây dựng Website quản lý phòng Gym thông minh**” được triển khai với mục tiêu hỗ trợ quá trình vận hành và quản lý hoạt động của phòng gym theo hướng hiện đại hóa, số hóa và tự động hóa. Hệ thống cho phép hội viên dễ dàng đăng ký tài khoản, đăng nhập, quản lý thông tin cá nhân, lựa chọn khóa tập phù hợp, theo dõi lịch sử giao dịch – thanh toán và tương tác thông qua chatbot hỗ trợ.

Bên cạnh đó, hệ thống cũng cung cấp các công cụ cần thiết cho quản trị viên và huấn luyện viên trong việc quản lý hội viên, khóa tập, dịch vụ, lịch hẹn cùng các nghiệp vụ liên quan. Tất cả đều được thực hiện một cách thuận tiện, thống nhất và hiệu quả trên nền tảng web.

Trong suốt quá trình thực hiện, em đã áp dụng các kiến thức về phân tích – thiết kế hệ thống, sử dụng các mô hình như Entity-Relationship Diagram, Use Case, Class Diagram, Sequence Diagram, Activity Diagram để mô tả cấu trúc và luồng xử lý dữ liệu. Hệ thống được xây dựng với ASP.NET Core 8.0 cho phần backend, đảm nhiệm xử lý nghiệp vụ và kết nối dữ liệu, phần giao diện người dùng được thiết kế bằng ReactJS/Next.js nhằm mang lại trải nghiệm trực quan, dễ sử dụng; cơ sở dữ liệu được quản lý bằng PostgreSQL thông qua Entity Framework Core, đảm bảo tính toàn vẹn và hiệu suất truy vấn.

## LỜI NÓI ĐẦU

Để hoàn thành được đề tài “**Xây dựng Website quản lý phòng Gym thông minh**”, em xin gửi lời tri ân sâu sắc đến quý thầy cô Khoa Công Nghệ Thông Tin – Trường Đại Học Kiến Trúc Đà Nẵng. Trong suốt những năm học vừa qua, quý thầy cô không chỉ truyền đạt cho em nền tảng kiến thức chuyên môn vững chắc mà còn định hướng tư duy, khơi dậy tinh thần học hỏi và tạo động lực để em không ngừng nỗ lực hoàn thiện bản thân. Những kiến thức và kinh nghiệm quý báu ấy chính là hành trang quan trọng giúp em tự tin hơn khi tham gia vào các dự án thực tế sau này.

Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến thầy **TS. Lê Văn Minh**, người trực tiếp hướng dẫn và đồng hành cùng em trong suốt quá trình thực hiện đề tài. Thầy đã tận tình giúp em định hướng nội dung nghiên cứu, giải đáp những khó khăn về kỹ thuật, đồng thời luôn động viên, khích lệ em hoàn thành tốt nhiệm vụ được giao. Sự tâm huyết, trách nhiệm và sự hỗ trợ kịp thời của thầy đã giúp em phát triển hệ thống một cách bài bản, khoa học và phù hợp với nhu cầu quản lý thực tế của phòng Gym.

Em cũng xin gửi lời cảm ơn đến các anh chị và bạn bè đã luôn sẵn sàng trao đổi, góp ý, hỗ trợ kiểm thử và chia sẻ kinh nghiệm trong quá trình phát triển hệ thống. Những ý kiến và sự giúp đỡ chân thành ấy đã góp phần quan trọng giúp em hoàn thiện hơn sản phẩm của mình.

Mặc dù đã cố gắng hết sức, nhưng do hạn chế về kinh nghiệm và thời gian thực hiện, đề tài không tránh khỏi những thiếu sót. Em rất mong nhận được các ý kiến đóng góp quý báu từ quý thầy cô để có thể tiếp tục nâng cao kiến thức, kỹ năng và hoàn thiện hơn trong các dự án tương lai.

Một lần nữa, em xin chân thành cảm ơn quý thầy cô đã tận tâm giảng dạy, hướng dẫn và tạo điều kiện thuận lợi cho em trong suốt quá trình học tập và thực hiện đề tài tốt nghiệp.

**Xin chân thành cảm ơn!**

## CAM ĐOAN

Em xin cam đoan rằng đề tài **“Xây dựng Website quản lý phòng Gym thông minh”** là kết quả nghiên cứu và thực hiện của chính bản thân em dưới sự hướng dẫn của thầy **TS. Lê Văn Minh**. Các nội dung, số liệu, hình ảnh và kết quả được trình bày trong báo cáo đều được em thu thập, phân tích và xây dựng một cách trung thực.

Những tài liệu tham khảo được sử dụng trong báo cáo đều được trích dẫn rõ ràng theo đúng quy định. Em xin chịu hoàn toàn trách nhiệm trước nhà trường về tính chính xác và trung thực của toàn bộ nội dung trong đề tài.

Sinh viên thực hiện

Ngô Tấn Hiếu

# MỤC LỤC

|                                                       |             |
|-------------------------------------------------------|-------------|
| <b>Tóm Tắt</b>                                        |             |
| <b>Lời Nói Đầu .....</b>                              | <b>i</b>    |
| <b>Cam Đoan .....</b>                                 | <b>iii</b>  |
| <b>Mục Lục .....</b>                                  | <b>iii</b>  |
| <b>Danh Sách Hình Ảnh .....</b>                       | <b>vi</b>   |
| <b>Danh Sách Bảng .....</b>                           | <b>viii</b> |
| <b>Danh Sách Các Chữ Viết Tắt .....</b>               | <b>ix</b>   |
| <b>Chương 1: GIỚI THIỆU ĐỀ TÀI .....</b>              | <b>2</b>    |
| 1.1. Lý do chọn đề tài .....                          | 2           |
| 1.2. Mục tiêu đề tài .....                            | 3           |
| 1.3. Ý nghĩa khoa học và thực tiễn .....              | 4           |
| 1.3.1 Ý nghĩa khoa học .....                          | 4           |
| 1.3.2 Ý nghĩa thực tiễn .....                         | 4           |
| 1.4. Đối tượng và phạm vi nghiên cứu .....            | 5           |
| 1.4.1 Đối tượng nghiên cứu .....                      | 5           |
| 1.4.2 Phạm vi nghiên cứu .....                        | 5           |
| <b>Chương 2: CƠ SỞ LÝ THUYẾT .....</b>                | <b>7</b>    |
| 2.1. Giới thiệu về cơ sở dữ liệu Postgresql .....     | 7           |
| 2.1.1. Đặc điểm nổi bật của Postgresql .....          | 7           |
| 2.1.2 Đặc điểm của Sql trong Postgresql .....         | 9           |
| 2.1.3 Các lợi ích khi sử dụng Postgresql .....        | 11          |
| 2.2. Giới thiệu về Next.js .....                      | 11          |
| 2.2.1. Các điểm nổi bật của Next.js .....             | 12          |
| 2.2.2. Các ưu điểm khi sử dụng Next.js .....          | 14          |
| 2.3. Giới thiệu Asp.net Core 8.0 .....                | 15          |
| 2.3.1. Đặc điểm nổi bật của Asp.net Core 8.0 .....    | 15          |
| 2.3.2. Ưu điểm và lợi ích của Asp.net Core 8.0 .....  | 18          |
| <b>Chương 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG .....</b> | <b>19</b>   |
| 3.1. Khảo sát yêu cầu hệ thống .....                  | 19          |
| 3.2. Phân tích yêu cầu hệ thống .....                 | 20          |
| 3.2.1. Yêu cầu chức năng .....                        | 20          |

|                                                                         |    |
|-------------------------------------------------------------------------|----|
| 3.2.2. Yêu cầu phi chức năng .....                                      | 22 |
| 3.2.3. Khảo sát, lấy yêu cầu .....                                      | 22 |
| 3.3. Phân tích, đặc tả yêu cầu.....                                     | 22 |
| 3.3.1. Người dùng (hội viên).....                                       | 23 |
| 3.3.2. Quản trị viên(admin) .....                                       | 24 |
| 3.4. Biểu đồ use case.....                                              | 24 |
| 3.4.1. Biểu đồ Use Case tổng quát. ....                                 | 25 |
| 3.4.2. Biểu đồ Use Case đăng kí. ....                                   | 27 |
| 3.4.3. Biểu đồ Use Case đăng nhập.....                                  | 28 |
| 3.4.4. Biểu đồ Use Case cập nhật thông tin hội viên. ....               | 29 |
| 3.4.5. Biểu đồ Use Case thanh toán. ....                                | 30 |
| 3.4.6. Biểu đồ Use Case quản lý người dùng.....                         | 31 |
| 3.4.7. Biểu đồ Use Case quản lý dịch vụ. ....                           | 33 |
| 3.4.8. Biểu đồ Use Case quản lý khoá tập.....                           | 34 |
| 3.4.9. Biểu đồ Use Case xem báo cáo và thống kê. ....                   | 35 |
| 3.4.10. Biểu đồ Use Case xem lịch hẹn. ....                             | 36 |
| 3.4.11. Biểu đồ Use Case chat bot. ....                                 | 37 |
| 3.5. Biểu đồ tuần tự.....                                               | 38 |
| 3.5.1. Biểu đồ tuần tự đăng ký. ....                                    | 38 |
| 3.5.2. Biểu đồ tuần tự đăng nhập. ....                                  | 39 |
| 3.5.3. Biểu đồ tuần tự thanh toán. ....                                 | 39 |
| 3.5.4. Biểu đồ tuần tự cập nhật thông tin. ....                         | 40 |
| 3.5.5. Biểu đồ tuần tự đăng nhập và sử dụng các chức năng quản trị..... | 41 |
| 3.6 Biểu đồ hoạt động. ....                                             | 41 |
| 3.6.1. Biểu đồ hoạt động đăng ký. ....                                  | 42 |
| 3.6.2. Biểu đồ hoạt động đăng nhập.....                                 | 43 |
| 3.6.3. Biểu đồ hoạt động chỉnh sửa thông tin cá nhân. ....              | 44 |
| 3.6.4. Biểu đồ hoạt động thanh toán. ....                               | 45 |
| 3.6.5. Biểu đồ hoạt động quản lý người dùng.....                        | 46 |
| 3.6.6. Biểu đồ hoạt động quản lý khoá tập.....                          | 46 |
| 3.7. Biểu đồ lớp.....                                                   | 47 |
| 3.8. Biểu đồ erd (entity relationship diagram). ....                    | 48 |
| 3.9. Thiết kế cơ sở dữ liệu .....                                       | 52 |

|                                                             |           |
|-------------------------------------------------------------|-----------|
| <b>Chương 4: TRIỂN KHAI HỆ THỐNG.....</b>                   | <b>57</b> |
| 4.1. Kiến trúc hệ thống.....                                | 57        |
| 4.2. Giao diện người dùng(hội viên).....                    | 59        |
| 4.2.1. Giao diện trang chủ. ....                            | 59        |
| 4.2.2. Giao diện đăng ký. ....                              | 61        |
| 4.2.3. Giao diện đăng nhập. ....                            | 61        |
| 4.2.4. Giao diện mua khoá tập. ....                         | 62        |
| 4.2.5. Giao diện khoá tập đã mua.....                       | 63        |
| 4.2.6. Giao diện lịch hẹn. ....                             | 63        |
| 4.2.7. Giao diện lịch sử thanh toán.....                    | 64        |
| 4.2.8. Giao diện hồ sơ. ....                                | 65        |
| 4.2.9. Giao diện giỏ hàng. ....                             | 65        |
| 4.2.10. Giao diện thanh toán. ....                          | 66        |
| 4.2.11. Giao diện chat bot. ....                            | 67        |
| 4.3. Giao diện admin.....                                   | 67        |
| 4.3.1. Giao diện báo cáo và thống kê.....                   | 67        |
| 4.3.2. Giao diện quản lý người dùng.....                    | 68        |
| 4.3.3. Giao diện thêm người dùng.....                       | 69        |
| 4.3.4. Giao diện sửa người dùng. ....                       | 69        |
| 4.3.5. Giao diện quản lý dịch vụ. ....                      | 70        |
| 4.3.6. Giao diện thêm dịch vụ. ....                         | 71        |
| 4.3.7. Giao diện sửa dịch vụ.....                           | 71        |
| 4.3.8. Giao diện quản lý khoá tập. ....                     | 72        |
| 4.3.9. Giao diện thêm khoá tập. ....                        | 73        |
| 4.3.10. Giao diện sửa khoá tập.....                         | 73        |
| 4.3.11. Giao diện quản lý lịch hẹn. ....                    | 74        |
| 4.3.12. Giao diện thêm lịch hẹn. ....                       | 75        |
| 4.3.13. Giao diện sửa lịch hẹn.....                         | 75        |
| 4.4. Giao diện huấn luyện viên. ....                        | 76        |
| 4.4.1. Giao diện khoá học phụ trách (huấn luyện viên). .... | 76        |
| 4.4.2. Giao diện danh sách học viên (huấn luyện viên). .... | 77        |
| <b>KẾT LUẬN .....</b>                                       | <b>78</b> |
| <b>TÀI LIỆU THAM KHẢO.....</b>                              | <b>80</b> |

## DANH SÁCH HÌNH ẢNH

|                                                                             |    |
|-----------------------------------------------------------------------------|----|
| Hình 2.1 Logo PostgreSQL .....                                              | 7  |
| Hình 2.2 Logo Nest.js .....                                                 | 12 |
| Hình 2.3 Logo ASP.NET Core 8.0 .....                                        | 15 |
| Hình 3.1 Biểu đồ Use Case tổng quát.....                                    | 25 |
| Hình 3.2 Biểu đồ Use Case đăng kí.....                                      | 27 |
| Hình 3.3 Biểu đồ Use Case đăng nhập .....                                   | 28 |
| Hình 3.4 Biểu đồ Use Case cập nhật thông tin cá nhân.....                   | 29 |
| Hình 3.5 Biểu đồ Use Case thanh toán.....                                   | 30 |
| Hình 3.6 Biểu đồ Use Case quản lý người dùng. ....                          | 31 |
| Hình 3.7 Biểu đồ Use Case quản lý dịch vụ .....                             | 33 |
| Hình 3.8 Biểu đồ Use Case quản lý khoá tập .....                            | 34 |
| Hình 3.9 Biểu đồ Use Case xem báo cáo và thống kê.....                      | 35 |
| Hình 3.10 Biểu đồ Use Case quản lý lịch hẹn. ....                           | 36 |
| Hình 3.11 Biểu đồ Use Case chatbot. ....                                    | 37 |
| Hình 3.12 Biểu đồ tuần tự đăng ký.....                                      | 38 |
| Hình 3.13 Biểu đồ tuần tự đăng nhập. ....                                   | 39 |
| Hình 3.14 Biểu đồ tuần tự thanh toán. ....                                  | 39 |
| Hình 3.15 Biểu đồ tuần tự cập nhật thông tin.....                           | 40 |
| Hình 3.16 Biểu đồ tuần tự đăng nhập và sử dụng các chức năng quản trị. .... | 41 |
| Hình 3.17 Biểu đồ hoạt động đăng ký. ....                                   | 42 |
| Hình 3.18 Biểu đồ hoạt động đăng nhập. ....                                 | 43 |
| Hình 3.19 Biểu đồ hoạt động chỉnh sửa thông tin cá nhân.....                | 44 |
| Hình 3.20 Biểu đồ hoạt động thanh toán. ....                                | 45 |
| Hình 3.21 Biểu đồ hoạt động quản lý người dùng.....                         | 46 |
| Hình 3.22 Biểu đồ hoạt động quản lý khoá tập. ....                          | 46 |
| Hình 3.23 Biểu đồ lớp.....                                                  | 47 |
| Hình 3.24 Biểu đồ ERD.....                                                  | 48 |
| Hình 4.1 Giao diện trang chủ.....                                           | 60 |
| Hình 4.2 Giao diện đăng ký.....                                             | 61 |
| Hình 4.3 Giao diện đăng nhập. ....                                          | 62 |

|                                               |    |
|-----------------------------------------------|----|
| Hình 4.4 Giao diện mua khoá tập. ....         | 62 |
| Hình 4.5 Giao diện khoá tập đã mua. ....      | 63 |
| Hình 4.6 Giao diện lịch hẹn của tôi. ....     | 64 |
| Hình 4.7 Giao diện lịch sử thanh toán. ....   | 64 |
| Hình 4.8 Giao diện hồ sơ. ....                | 65 |
| Hình 4.9 Giao diện giỏ hàng. ....             | 66 |
| Hình 4.10 Giao diện thanh toán. ....          | 66 |
| Hình 4.11 Giao diện chat bot. ....            | 67 |
| Hình 4.12 Giao diện báo cáo và thống kê. .... | 68 |
| Hình 4.13 Giao diện quản lý người dùng. ....  | 68 |
| Hình 4.14 Giao diện thêm người dùng. ....     | 69 |
| Hình 4.15 Giao diện sửa người dùng. ....      | 70 |
| Hình 4.16 Giao diện quản lý dịch vụ. ....     | 70 |
| Hình 4.17 Giao diện thêm dịch vụ. ....        | 71 |
| Hình 4.18 Giao diện sửa dịch vụ. ....         | 72 |
| Hình 4.19 Giao diện quản lý khoá tập. ....    | 72 |
| Hình 4.21 Giao diện sửa khoá tập. ....        | 74 |
| Hình 4.22 Giao diện quản lý lịch hẹn. ....    | 74 |
| Hình 4.23 Giao diện thêm lịch hẹn. ....       | 75 |
| Hình 4.24 Giao diện sửa lịch hẹn. ....        | 76 |
| Hình 4.25 Giao diện khoá học phụ trách. ....  | 76 |
| Hình 4.26 Giao diện danh sách học viên. ....  | 77 |

## DANH SÁCH BẢNG

|                                                                    |    |
|--------------------------------------------------------------------|----|
| Bảng 3.1 Đặc tả chi tiết Use Case đăng kí.....                     | 27 |
| Bảng 3.2 Đặc tả chi tiết Use Case đăng nhập.....                   | 28 |
| Bảng 3.3 Đặc tả chi tiết Use Case cập nhật thông tin cá nhân. .... | 29 |
| Bảng 3.4 Đặt tả chi tiết Use Case thanh toán. ....                 | 30 |
| Bảng 3.5 Đặt tả chi tiết Use Case quản lý người dùng.....          | 32 |
| Bảng 3.6 Đặt tả chi tiết Use Case quản lý dịch vụ. ....            | 33 |
| Bảng 3.7 Đặt tả chi tiết Use Case quản lý khoá tập.....            | 34 |
| Bảng 3.8 Đặt tả chi tiết Use Case xem báo cáo và thống kê. ....    | 35 |
| Bảng 3.9 Đặt tả chi tiết Use Case quản lý lịch hẹn. ....           | 36 |
| Bảng 3.10 Đặt tả chi tiết Use Case chatbot. ....                   | 37 |

## DANH SÁCH CÁC CHỮ VIẾT TẮT

| Viết tắt | Nghĩa đầy đủ                       | Giải thích                                                                        |
|----------|------------------------------------|-----------------------------------------------------------------------------------|
| UI       | User Interface                     | Giao diện trực quan giúp người dùng tương tác với hệ thống.                       |
| API      | Application Programming Interface  | Giao tiếp trung gian giữa các hệ thống hoặc các module phần mềm.                  |
| ORM      | Object-Relational Mapping          | Kỹ thuật ánh xạ dữ liệu giữa mô hình hướng đối tượng và cơ sở dữ liệu quan hệ.    |
| MVP      | Minimum Viable Product             | Phiên bản sản phẩm tối thiểu có đủ tính năng để vận hành và kiểm thử.             |
| SSR      | Server-Side Rendering              | Render trang web tại server trước khi gửi về trình duyệt để tối ưu tốc độ và SEO. |
| SSG      | Static Site Generation             | Sinh các trang HTML tĩnh ngay trong quá trình build giúp tăng tốc độ tải.         |
| ISR      | Incremental Static Regeneration    | Tái tạo từng phần nội dung tĩnh theo chu kỳ để đảm bảo dữ liệu luôn mới.          |
| HTTP     | HyperText Transfer Protocol        | Giao thức truyền tải dữ liệu trong môi trường web.                                |
| HTTPS    | HyperText Transfer Protocol Secure | Phiên bản mã hóa và an toàn của giao thức HTTP.                                   |
| SQL      | Structured Query Language          | Ngôn ngữ truy vấn dùng để thao tác dữ liệu trên cơ sở dữ liệu quan hệ.            |
| DB       | Database                           | Kho lưu trữ chứa toàn bộ dữ liệu của hệ thống.                                    |
| EF Core  | Entity Framework Core              | ORM của .NET hỗ trợ truy cập và thao tác dữ liệu hiệu quả.                        |
| LINQ     | Language Integrated Query          | Cú pháp truy vấn mạnh mẽ tích hợp sẵn trong .NET dùng để xử lý dữ liệu.           |

|      |                                 |                                                                            |
|------|---------------------------------|----------------------------------------------------------------------------|
| JWT  | JSON Web Token                  | Chuẩn token dùng để xác thực và phân quyền người dùng trong hệ thống.      |
| PT   | Personal Trainer                | Huấn luyện viên cá nhân trong hệ thống quản lý phòng gym.                  |
| SPA  | Single Page Application         | Mô hình ứng dụng web chỉ tải một trang duy nhất và cập nhật nội dung động. |
| CORS | Cross-Origin Resource Sharing   | Cơ chế cho phép chia sẻ tài nguyên giữa các miền khác nhau.                |
| XSS  | Cross-Site Scripting            | Lỗ hổng bảo mật liên quan đến việc chèn mã độc vào giao diện web.          |
| CSRF | Cross-Site Request Forgery      | Tấn công giả mạo yêu cầu từ người dùng hợp lệ để thao tác trái phép.       |
| DI   | Dependency Injection            | Kỹ thuật tiêm phụ thuộc nhằm giảm liên kết giữa các thành phần hệ thống.   |
| REST | Representational State Transfer | Kiến trúc thiết kế API đơn giản và phổ biến hiện nay.                      |
| gRPC | gRPC Remote Procedure Call      | Giao thức gọi thủ tục từ xa hiệu năng cao.                                 |

## CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

### 1.1. Lý do chọn đề tài

Trong bối cảnh đời sống xã hội ngày càng phát triển, nhu cầu rèn luyện sức khỏe, cải thiện thể chất và nâng cao chất lượng cuộc sống của con người cũng ngày một gia tăng. Các phòng Gym, trung tâm thể hình mọc lên ngày càng nhiều với quy mô đa dạng, từ nhỏ lẻ cho đến những chuỗi trung tâm lớn, nhằm đáp ứng nhu cầu tập luyện của đông đảo hội viên. Sự cạnh tranh ngày càng gay gắt giữa các phòng tập cũng đặt ra yêu cầu cấp thiết về việc quản lý hiệu quả, chuyên nghiệp và hiện đại hơn.

Tuy nhiên, trên thực tế, phần lớn các phòng Gym vẫn còn quản lý thủ công bằng giấy tờ hoặc các công cụ đơn giản như Excel. Cách làm này tuy dễ triển khai nhưng lại bộc lộ nhiều hạn chế:

- Dữ liệu phân tán, khó đồng bộ, gây khó khăn trong việc lưu trữ và truy xuất thông tin hội viên, huấn luyện viên, gói tập, thời hạn sử dụng và dịch vụ đi kèm.
- Khó khăn trong việc theo dõi tiến độ và hiệu quả tập luyện của từng hội viên, dẫn đến trải nghiệm khách hàng chưa được tối ưu.
- Tốn nhiều thời gian và nhân lực trong các khâu đăng ký, gia hạn gói tập, quản lý tài chính hay sắp xếp lịch tập.
- Dễ xảy ra sai sót trong nhập liệu, tính toán, thống kê hoặc báo cáo.

Trong khi đó, xu hướng ứng dụng công nghệ thông tin vào quản lý dịch vụ đang trở nên phổ biến ở nhiều lĩnh vực như thương mại, y tế, giáo dục và giải trí. Việc xây dựng một Hệ thống quản lý phòng Gym thông minh không chỉ giúp tự động hóa các quy trình quản lý, mà còn mang lại nhiều lợi ích thiết thực:

- Tăng cường tính chính xác và minh bạch trong quản lý dữ liệu.
- Tiết kiệm thời gian, giảm thiểu chi phí nhân sự, hạn chế sai sót trong công việc.
- Nâng cao trải nghiệm khách hàng thông qua việc quản lý gói tập, thống kê tiến độ tập luyện và hỗ trợ trực tuyến.

- Tạo nền tảng cho việc mở rộng quy mô hoạt động, phát triển dịch vụ kèm theo, đồng thời nâng cao lợi thế cạnh tranh cho các phòng tập trong thời đại công nghệ số.

Với những lý do trên, em quyết định chọn đề tài “**Xây dựng Website quản lý phòng Gym thông minh**”. Đề tài không chỉ mang ý nghĩa khoa học trong việc áp dụng công nghệ để giải quyết một bài toán thực tế, mà còn có ý nghĩa thực tiễn cao, có thể triển khai và ứng dụng rộng rãi tại các phòng Gym hiện nay.

## 1.2. Mục tiêu đề tài

Đề tài “**Xây dựng Website quản lý phòng Gym thông minh**” hướng tới mục tiêu xây dựng một nền tảng hiện đại, hỗ trợ số hóa toàn bộ quy trình quản lý và chăm sóc khách hàng tại phòng Gym. Hệ thống không chỉ phục vụ cho công tác quản lý của Admin mà còn mang lại trải nghiệm tiện lợi, cá nhân hóa cho hội viên, từ đó nâng cao chất lượng dịch vụ và tăng tính cạnh tranh cho phòng Gym.

Cụ thể, đề tài đặt ra các mục tiêu sau:

- **Xây dựng hệ thống đăng nhập và phân quyền:** cho phép Admin, huấn luyện viên (PT) và hội viên đăng nhập, đăng ký, cập nhật thông tin cá nhân một cách an toàn.
- **Quản lý khóa tập, người dùng, lịch hẹn và dịch vụ:** Admin có thể tạo – chỉnh sửa – xóa các khóa tập, dịch vụ của phòng gym; đồng thời quản lý thông tin hội viên, huấn luyện viên, lịch hẹn và toàn bộ hoạt động của hệ thống.
- **Hỗ trợ đăng ký và tham gia khóa tập:** Hội viên có thể dễ dàng tìm kiếm khóa tập phù hợp, đăng ký tham gia khóa tập của mình.
- **Thanh toán trực tuyến:** Tích hợp phương thức thanh toán nhanh, tiện lợi và bảo mật nhằm giúp hội viên thanh toán gói tập hoặc dịch vụ một cách dễ dàng.
- **Chatbot:** Cho phép khách hàng tương tác trực tiếp với hệ thống để hỏi thông tin về giá, khóa tập, dịch vụ, email, thời gian mở cửa và các câu hỏi phổ biến khác.
- **Tìm kiếm và tra cứu thông minh:** Cung cấp công cụ tìm kiếm nhanh các khóa tập.

Với các chức năng trên, đề tài kỳ vọng sẽ giải quyết được các vấn đề thực tế như:

- Giảm tải khối lượng công việc thủ công trong quản lý phòng tập.
- Tạo ra môi trường tập luyện thông minh, thân thiện và cá nhân hóa cho hội viên.
- Nâng cao hiệu quả quản lý, tăng sự hài lòng của khách hàng, từ đó cải thiện doanh thu và uy tín của phòng Gym.
- Hệ thống có thể được áp dụng cho nhiều mô hình khác nhau, từ phòng tập cá nhân, trung tâm thể hình cỡ vừa, cho đến chuỗi Gym quy mô lớn, đáp ứng nhu cầu quản lý đa dạng trong thời đại chuyển đổi số 4.0.

### **1.3. Ý nghĩa khoa học và thực tiễn**

#### **1.3.1 Ý nghĩa khoa học**

Đề tài góp phần minh chứng cho việc ứng dụng công nghệ thông tin trong quản lý dịch vụ, đặc biệt là lĩnh vực thể dục – thể hình, vốn đang phát triển mạnh trong xã hội hiện đại.

Thông qua việc xây dựng hệ thống, em nghiên cứu đã vận dụng và kết hợp nhiều kiến thức khoa học máy tính:

- Phân tích và thiết kế hệ thống thông tin (xác định yêu cầu, mô hình hóa quy trình quản lý phòng tập).
- Lập trình ứng dụng web theo kiến trúc Client – Server, sử dụng Frontend viết bằng TypeScript (ReactJS, Next.js) và Backend phát triển bằng ASP.NET Core 8.0.
- Quản lý cơ sở dữ liệu quan hệ (PostgreSQL) để xử lý và lưu trữ thông tin hội viên, khoá tập, thanh toán và lịch hẹn.

Đề tài cũng là minh chứng cụ thể cho xu hướng chuyển đổi số trong quản lý dịch vụ, góp phần làm rõ vai trò của công nghệ trong việc tối ưu hóa hoạt động vận hành và nâng cao trải nghiệm khách hàng.

#### **1.3.2 Ý nghĩa thực tiễn**

Hệ thống cung cấp giải pháp quản lý tập trung và tự động hóa cho các phòng Gym, giúp chủ phòng tập giảm thiểu công việc thủ công như ghi chép, theo dõi thủ công hay quản lý thông tin bằng Excel.

Đối với hội viên, hệ thống mang lại trải nghiệm hiện đại:

- Đăng ký, đăng nhập và quản lý khoá tập trực tuyến.

- Thanh toán và chatbot hỗ trợ.

Đối với chủ phòng Gym và quản lý, hệ thống hỗ trợ:

- Quản lý hiệu quả nhân sự, khoá tập, dịch vụ và tài chính.
- Đưa ra quyết định chiến lược nhờ báo cáo trực quan, dữ liệu phân tích xu hướng.

Trong bối cảnh nhu cầu chăm sóc sức khỏe và thể hình ngày càng tăng, hệ thống này hoàn toàn có thể được áp dụng thực tế, từ các phòng tập nhỏ đến các chuỗi trung tâm Gym quy mô lớn. Điều này không chỉ nâng cao năng lực cạnh tranh cho các phòng tập mà còn góp phần thúc đẩy thói quen sống lành mạnh trong cộng đồng.

## 1.4. Đối tượng và phạm vi nghiên cứu

### 1.4.1 Đối tượng nghiên cứu

Đề tài tập trung nghiên cứu và giải quyết các vấn đề liên quan đến công tác quản lý trong phòng Gym, bao gồm:

- Người dùng (hội viên):
  - Người đăng ký tài khoản, mua khoá tập, sử dụng các dịch vụ của phòng gym.
  - Có nhu cầu xem thông tin cá nhân, lịch hẹn và lịch sử thanh toán.
- Nhân viên (PT):
  - Quản lý danh sách hội viên được phụ trách, theo dõi lịch dạy, hỗ trợ hội viên trong quá trình tập luyện.
- Quản trị viên (Admin):
  - Quản lý hệ thống tổng thể: tài khoản người dùng, khóa tập, dịch vụ, lịch hẹn, báo cáo doanh thu,...

### 1.4.2 Phạm vi nghiên cứu

Trong phạm vi thực hiện của đề tài, em tập trung “**Xây dựng Website quản lý phòng Gym thông minh**” chạy trên nền tảng web, với các chức năng cơ bản nhưng đáp ứng được nhu cầu quản lý thực tế:

- Hệ thống đăng ký, đăng nhập và quản lý tài khoản người dùng.
- Chức năng quản lý hội viên: sửa xóa, cập nhật thông tin.

- Chức năng quản lý khoá tập và dịch vụ: tạo mới, chỉnh sửa, cập nhật thông tin các khoá tập, dịch vụ.
- Chức năng quản lý nhân viên: lưu trữ thông tin huấn luyện viên và phân công.
- Chức năng thống kê và báo cáo doanh thu theo thời gian, hỗ trợ công tác quản lý tài chính.

Đối tượng người dùng chính mà hệ thống hướng đến gồm có:

- **Quản trị viên:** chịu trách nhiệm quản lý toàn bộ hệ thống, bao gồm hội viên, nhân viên, khoá tập, lịch hẹn và báo cáo tài chính.
- **Người dùng:** có thể đăng ký tài khoản, đăng nhập, tìm kiếm, lựa chọn và đăng ký khoá tập, thanh toán, chat bot.
- **Nhân viên (PT):** trực tiếp hướng dẫn hội viên, quản lý lịch tập, hỗ trợ đặt lịch hẹn của hội viên.

Trong giai đoạn này, hệ thống chủ yếu tập trung phát triển trên nền tảng web, chưa mở rộng sang ứng dụng di động. Tuy nhiên, kiến trúc được xây dựng theo hướng mở để có thể phát triển và tích hợp thêm trong tương lai.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

### 2.1. Giới thiệu về cơ sở dữ liệu PostgreSQL

PostgreSQL, thường được gọi tắt là Postgres, là một trong những hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở nổi tiếng và được sử dụng rộng rãi trên toàn thế giới [4]. PostgreSQL được khởi nguồn từ dự án nghiên cứu POSTGRES tại Đại học California, Berkeley vào giữa những năm 1980 dưới sự dẫn dắt của giáo sư Michael Stonebraker. Trải qua hơn ba thập kỷ phát triển liên tục, PostgreSQL đã trở thành một hệ quản trị cơ sở dữ liệu không chỉ mạnh mẽ về tính năng mà còn nổi bật bởi sự ổn định, tính bảo mật cao và cộng đồng người dùng, nhà phát triển đông đảo, hoạt động tích cực.



Hình 2.1 Logo PostgreSQL

#### 2.1.1. Đặc điểm nổi bật của PostgreSQL

- Mã nguồn mở và miễn phí: PostgreSQL được phân phối dưới giấy phép PostgreSQL License – một dạng giấy phép mã nguồn mở tương thích với GNU General Public License (GPL). Điều này cho phép người dùng tự do sử dụng, chỉnh sửa, phân phối mà không bị giới hạn bởi chi phí bản quyền. Nhờ đó, PostgreSQL được các cá nhân, doanh nghiệp nhỏ đến các tập đoàn lớn tin tưởng triển khai trong nhiều hệ thống quan trọng.
- Hỗ trợ mô hình dữ liệu quan hệ truyền thống: Là một RDBMS, PostgreSQL cho phép quản lý dữ liệu thông qua các bảng (tables) có mối quan hệ chặt chẽ với nhau thông qua khóa chính (Primary Key) và khóa ngoại (Foreign Key). Hệ thống hỗ trợ

- đầy đủ các ràng buộc toàn vẹn dữ liệu như unique, not null, check constraint, foreign key constraint, đảm bảo dữ liệu được lưu trữ một cách nhất quán và chính xác.
- Mở rộng sang dữ liệu phi quan hệ (NoSQL): Một trong những điểm khác biệt lớn của PostgreSQL so với nhiều hệ quản trị cơ sở dữ liệu quan hệ truyền thống khác là khả năng kết hợp dữ liệu quan hệ và phi quan hệ. PostgreSQL hỗ trợ lưu trữ và xử lý dữ liệu JSON, JSONB, XML, HSTORE (dữ liệu dạng key-value), cho phép người dùng vừa tận dụng sức mạnh của cơ sở dữ liệu quan hệ, vừa có tính linh hoạt tương tự như các hệ thống NoSQL (MongoDB, Cassandra). Điều này giúp PostgreSQL thích ứng tốt với các ứng dụng web và dịch vụ hiện đại, nơi dữ liệu thường có cấu trúc phức tạp và thay đổi linh hoạt.
  - Khả năng mở rộng mạnh mẽ: PostgreSQL được thiết kế theo hướng extensible (có thể mở rộng). Người dùng có thể định nghĩa kiểu dữ liệu mới (custom data types), hàm mới (user-defined functions), thậm chí cả toán tử mới phù hợp với yêu cầu của hệ thống. Ngoài ra, PostgreSQL còn hỗ trợ nhiều cơ chế lập chỉ mục (indexing) mạnh mẽ như B-Tree, Hash, GiST, GIN, BRIN, giúp tối ưu hiệu suất truy vấn đối với dữ liệu lớn.
  - Tương thích tiêu chuẩn SQL và khả năng lập trình: PostgreSQL tuân thủ chặt chẽ tiêu chuẩn SQL, đồng thời mở rộng với nhiều tính năng nâng cao như CTE (Common Table Expressions), Window Functions, Recursive Queries, Full-Text Search. Hệ thống còn hỗ trợ nhiều ngôn ngữ lập trình để viết hàm hoặc thủ tục lưu trữ (stored procedures) như PL/pgSQL, PL/Python, PL/Perl, PL/Java, giúp lập trình viên dễ dàng tích hợp và triển khai các nghiệp vụ phức tạp ngay trong cơ sở dữ liệu.
  - Tính bảo mật và an toàn dữ liệu cao: PostgreSQL cung cấp nhiều cơ chế bảo mật như xác thực người dùng (authentication), phân quyền chi tiết (role-based access control), kết nối an toàn thông qua SSL/TLS. Ngoài ra, cơ chế Write-Ahead Logging (WAL) giúp đảm bảo tính bền vững của dữ liệu, hỗ trợ khôi phục sau sự cố (crash recovery) và sao lưu (backup) hiệu quả.
  - Khả năng mở rộng quy mô và hiệu suất cao: PostgreSQL hỗ trợ xử lý song song (parallel query execution), phân mảnh bảng (table partitioning), và tích hợp tốt với các giải pháp replication (sao chép dữ liệu) như Streaming Replication, Logical Replication. Nhờ đó, hệ thống có thể mở rộng để phục vụ hàng triệu người dùng đồng thời và xử lý lượng dữ liệu khổng lồ mà vẫn đảm bảo hiệu năng.

- Ứng dụng rộng rãi trong thực tế: Với sự đa dạng về tính năng, PostgreSQL được sử dụng trong nhiều lĩnh vực khác nhau:
  - Ứng dụng web và thương mại điện tử: Lưu trữ dữ liệu người dùng, đơn hàng, sản phẩm.
  - Tài chính – ngân hàng: Quản lý giao dịch, phân tích dữ liệu với độ chính xác cao.
  - Viễn thông và IoT: Lưu trữ dữ liệu cảm biến, nhật ký thiết bị.
  - Trí tuệ nhân tạo và dữ liệu lớn: Kết hợp với các công cụ phân tích để xử lý dữ liệu phức tạp.

Có thể thấy, PostgreSQL là một hệ quản trị cơ sở dữ liệu mạnh mẽ, linh hoạt và toàn diện, vừa đáp ứng tốt nhu cầu của các ứng dụng truyền thống, vừa thích nghi với xu hướng công nghệ hiện đại. Nhờ sự ổn định, bảo mật, hiệu năng cao và cộng đồng phát triển rộng lớn, PostgreSQL đang ngày càng trở thành lựa chọn ưu tiên của các lập trình viên, nhà quản trị hệ thống và doanh nghiệp trên toàn cầu.

### ***2.1.2 Đặc điểm của SQL trong PostgreSQL***

Ngôn ngữ SQL (Structured Query Language) là thành phần cốt lõi trong hệ quản trị cơ sở dữ liệu PostgreSQL. PostgreSQL không chỉ tuân thủ nghiêm ngặt tiêu chuẩn SQL, mà còn mở rộng thêm nhiều tính năng nâng cao, cho phép người dùng thao tác dữ liệu một cách linh hoạt và hiệu quả. Các đặc điểm chính của SQL trong PostgreSQL có thể phân tích như sau:

#### **Ngôn ngữ chuẩn cho cơ sở dữ liệu quan hệ**

- SQL trong PostgreSQL tuân theo tiêu chuẩn quốc tế của SQL, đảm bảo khả năng tương thích với các hệ quản trị cơ sở dữ liệu quan hệ khác.
- Người dùng có thể dễ dàng định nghĩa, thao tác và quản lý dữ liệu thông qua cú pháp SQL quen thuộc, giảm bớt chi phí học tập và triển khai.
- Các lệnh cơ bản như SELECT, INSERT, UPDATE, DELETE, CREATE, DROP được hỗ trợ đầy đủ, giúp đáp ứng toàn diện nhu cầu quản lý dữ liệu.

#### **Hỗ trợ đầy đủ các thao tác dữ liệu và cấu trúc**

- PostgreSQL cung cấp đầy đủ các lệnh DDL (Data Definition Language) để định nghĩa cấu trúc cơ sở dữ liệu, ví dụ:

- CREATE TABLE để tạo bảng,
  - ALTER TABLE để chỉnh sửa cấu trúc,
  - DROP TABLE để xóa bảng.
- Các lệnh DML (Data Manipulation Language) cho phép thêm, sửa, xóa dữ liệu một cách thuận tiện và an toàn.
  - Người dùng cũng có thể sử dụng DCL (Data Control Language) và TCL (Transaction Control Language) để kiểm soát quyền truy cập và quản lý giao dịch (transaction).

### **Xây dựng các đối tượng quản lý dữ liệu nâng cao**

- View (khung nhìn): cho phép tạo ra các bảng ảo từ kết quả truy vấn, giúp tái sử dụng câu lệnh SQL, tăng cường bảo mật khi chia sẻ dữ liệu cho người dùng.
- Trigger (kích hoạt): hỗ trợ tự động hóa thao tác trong cơ sở dữ liệu, ví dụ: ghi log khi dữ liệu bị thay đổi, hoặc tự động tính toán giá trị khi có bản ghi mới được thêm vào.
- Function/Procedure (hàm, thủ tục): PostgreSQL hỗ trợ viết hàm và thủ tục lưu trữ bằng nhiều ngôn ngữ (PL/pgSQL, Python, Perl, Java...), cho phép lập trình nghiệp vụ ngay trong cơ sở dữ liệu, giúp tối ưu hiệu năng và giảm tải cho tầng ứng dụng.

### **Khả năng mở rộng mạnh mẽ**

- PostgreSQL cho phép định nghĩa kiểu dữ liệu mới (user-defined data types), phù hợp cho các ứng dụng đặc thù như dữ liệu hình học, không gian (GIS), đa phương tiện.
- Hệ thống hỗ trợ nhiều dạng chỉ mục (index) khác nhau như B-Tree, Hash, GiST, GIN, BRIN... và cho phép người dùng tự định nghĩa cơ chế lập chỉ mục riêng, giúp tối ưu hiệu suất truy vấn trên tập dữ liệu lớn.
- PostgreSQL còn hỗ trợ toán tử tùy chỉnh và hàm mở rộng, tạo sự linh hoạt cao khi xây dựng ứng dụng phức tạp.

### **Đảm bảo ràng buộc và toàn vẹn dữ liệu nghiêm ngặt**

- SQL trong PostgreSQL cung cấp nhiều cơ chế ràng buộc (constraint) để duy trì tính chính xác của dữ liệu, bao gồm:
  - PRIMARY KEY, FOREIGN KEY để đảm bảo quan hệ giữa các bảng.
  - UNIQUE, NOT NULL để kiểm soát tính hợp lệ của dữ liệu.

- CHECK để định nghĩa điều kiện logic cho dữ liệu nhập vào.
- Cơ chế transaction tuân theo nguyên tắc ACID (Atomicity, Consistency, Isolation, Durability), đảm bảo mọi thay đổi dữ liệu được thực hiện một cách an toàn và không gây sai lệch trong hệ thống, ngay cả khi có sự cố bất ngờ.

### **2.1.3 Các lợi ích khi sử dụng PostgreSQL**

- Mã nguồn mở, miễn phí: Hoàn toàn miễn phí với giấy phép mở, có thể tùy chỉnh theo nhu cầu.
- Hiệu năng cao: Xử lý khối lượng dữ liệu lớn và nhiều truy vấn đồng thời.
- Hỗ trợ đa nền tảng: Cài đặt và chạy tốt trên nhiều hệ điều hành như Windows, Linux, macOS.
- Khả năng mở rộng: Hỗ trợ dữ liệu quan hệ và phi quan hệ, thích hợp cho cả ứng dụng nhỏ và hệ thống quy mô lớn.
- Bảo mật tốt: Có nhiều cơ chế xác thực, phân quyền chi tiết theo người dùng và vai trò.
- Cộng đồng rộng lớn: Được nhiều tổ chức, doanh nghiệp sử dụng, có cộng đồng phát triển tích cực và nhiều tài liệu hỗ trợ.

### **2.2. Giới thiệu về Next.js**

Next.js là một framework phát triển ứng dụng web hiện đại được xây dựng trên nền tảng React.js, do công ty Vercel phát triển và phát hành [1]. Với sự kết hợp giữa những ưu điểm vốn có của React và các tính năng nâng cao do Next.js bổ sung, framework này đã nhanh chóng trở thành một trong những lựa chọn phổ biến nhất trong cộng đồng lập trình viên front-end và full-stack.

Trong khi React cung cấp nền tảng mạnh mẽ để xây dựng giao diện người dùng (UI) theo dạng component-based (dựa trên thành phần), thì Next.js mở rộng khả năng của React bằng cách bổ sung các tính năng quan trọng như Server-Side Rendering (SSR), Static Site Generation (SSG), API Routes, tối ưu hóa hiệu năng, và hỗ trợ tích hợp linh hoạt với nhiều hệ thống khác nhau. Nhờ vậy, Next.js giúp phát triển ứng dụng web nhanh hơn, tối ưu hơn và thân thiện hơn với công cụ tìm kiếm (SEO).



Hình 2.2 Logo Next.js

### ***2.2.1. Các điểm nổi bật của Next.js***

#### **Hỗ trợ nhiều phương pháp render: SSR và SSG**

- SSR (Server-Side Rendering): Nội dung trang web được render sẵn từ phía server trước khi gửi đến trình duyệt.
- SSG (Static Site Generation): Các trang web được biên dịch thành tệp HTML tĩnh ngay trong quá trình build, giúp tăng tốc độ tải trang, giảm tải cho máy chủ và tiết kiệm tài nguyên.
- ISR (Incremental Static Regeneration): Một tính năng mở rộng cho phép Next.js tái tạo nội dung tĩnh theo từng khoảng thời gian nhất định, đảm bảo dữ liệu luôn cập nhật mà vẫn giữ tốc độ cao của SSG.

#### **Routing tự động, dễ sử dụng**

- Không giống như React thuần túy (cần thư viện bên ngoài như React Router), Next.js cung cấp cơ chế routing tích hợp sẵn dựa trên cấu trúc thư mục.
- Mỗi file trong thư mục /pages sẽ tương ứng với một route trong ứng dụng. Ví dụ:
  - /pages/index.js → /
  - /pages/about.js → /about
- Cách tiếp cận này đơn giản, rõ ràng, dễ duy trì, đồng thời hỗ trợ dynamic routing (định tuyến động) khi cần hiển thị nội dung dựa trên tham số (ví dụ: /products/[id]).

## API Routes – Xây dựng backend đơn giản ngay trong Next.js

- Một điểm mạnh nổi bật của Next.js là hỗ trợ API Routes, cho phép lập trình viên xây dựng các endpoint backend ngay trong cùng dự án front-end.
- Thay vì phải tạo một server riêng bằng Express hoặc NestJS, ta có thể viết trực tiếp các file trong thư mục /pages/api. Mỗi file này sẽ tự động trở thành một API endpoint.
- Ví dụ: file /pages/api/users.js có thể xử lý các yêu cầu HTTP như GET, POST, PUT, DELETE để quản lý dữ liệu người dùng.
- Điều này giúp tiết kiệm thời gian phát triển, đặc biệt đối với các ứng dụng vừa và nhỏ hoặc MVP (Minimum Viable Product).

## Khả năng tích hợp cơ sở dữ liệu linh hoạt

- Next.js không bị ràng buộc vào một hệ quản trị cơ sở dữ liệu cụ thể. Thay vào đó, nó dễ dàng tích hợp với nhiều hệ thống như MySQL, PostgreSQL, MongoDB, SQLite... thông qua ORM (Object-Relational Mapping).
- Hai ORM phổ biến thường được sử dụng với Next.js là:
  - Prisma: Cung cấp API hiện đại, mạnh mẽ, dễ sử dụng, hỗ trợ TypeScript rất tốt.
  - Sequelize: Một ORM lâu đời, hỗ trợ nhiều loại cơ sở dữ liệu quan hệ, cú pháp quen thuộc với lập trình viên Node.js.
- Nhờ tính linh hoạt này, Next.js phù hợp cho nhiều loại ứng dụng, từ hệ thống quản lý dữ liệu nhỏ (như quản lý hội viên phòng Gym) cho đến các nền tảng thương mại điện tử quy mô lớn.

## Hỗ trợ TypeScript – An toàn và dễ bảo trì

- Next.js hỗ trợ TypeScript một cách nguyên bản (built-in). Khi khởi tạo dự án, chỉ cần thêm vài cấu hình nhỏ, toàn bộ project có thể được viết bằng TypeScript.
- Lợi ích của TypeScript trong Next.js:
  - Giúp phát hiện lỗi sớm ngay từ quá trình viết code nhờ cơ chế kiểm tra kiểu dữ liệu (type checking).
  - Dễ dàng mở rộng và bảo trì ứng dụng quy mô lớn.

- Cải thiện trải nghiệm lập trình nhờ gợi ý thông minh từ IDE (IntelliSense).

### **Tối ưu hiệu năng và trải nghiệm người dùng**

Next.js được thiết kế để mang lại trải nghiệm nhanh chóng và mượt mà nhất:

- Code Splitting và Lazy Loading: Chia nhỏ mã nguồn thành các gói nhỏ, chỉ tải những phần cần thiết để giảm thời gian tải ban đầu.
- Image Optimization: Tích hợp sẵn tính năng tối ưu hóa hình ảnh ([next/image](#)), tự động nén ảnh, hỗ trợ lazy loading và nhiều định dạng ảnh hiện đại như WebP.
- Built-in CSS và hỗ trợ Tailwind CSS: Cho phép sử dụng CSS Modules, Sass, hoặc tích hợp các thư viện CSS phổ biến để xây dựng giao diện đẹp và gọn gàng.
- Prefetching routes: Các trang liên kết (links) được tải trước ngầm (preload) để giảm thời gian chờ khi người dùng điều hướng.

### **Cộng đồng lớn và hệ sinh thái mạnh mẽ**

- Là một sản phẩm của Vercel, Next.js nhận được sự đầu tư, phát triển và quảng bá mạnh mẽ.
- Cộng đồng Next.js phát triển nhanh chóng, với hàng nghìn package, plugin, template mã nguồn mở, cũng như nhiều hướng dẫn chi tiết, khóa học trực tuyến.
- Nhiều công ty công nghệ lớn đã chọn Next.js làm nền tảng, chẳng hạn như Netflix, TikTok, Twitch, GitHub, Notion, Uber. Điều này khẳng định tính ổn định và khả năng đáp ứng của framework này trong thực tế.

Nhờ các tính năng trên, Next.js trở thành lựa chọn hàng đầu trong việc xây dựng ứng dụng web hiện đại, trong đó có hệ thống quản lý phòng Gym thông minh.

#### **2.2.2. Các ưu điểm khi sử dụng Next.**

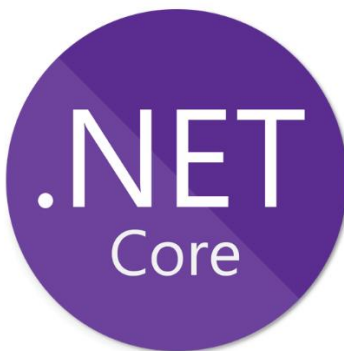
Việc sử dụng Next.js đem lại nhiều lợi ích trong phát triển phần mềm quản lý phòng Gym:

- Hiệu năng cao: Với SSR và SSG, người dùng có thể truy cập nhanh, giảm thời gian tải trang.

- SEO (Search Engine Optimization): Các trang được render sẵn HTML, giúp công cụ tìm kiếm dễ dàng index.
- Dễ phát triển và bảo trì: Cấu trúc dự án rõ ràng, dễ dàng mở rộng thêm tính năng.
- Tích hợp linh hoạt: Có thể kết nối với nhiều dịch vụ khác như API, MySQL, hệ thống thanh toán (Stripe).
- Trải nghiệm người dùng mượt mà: Nhờ cơ chế prefetching, điều hướng giữa các trang nhanh gần như ứng dụng mobile.

## 2.3. Giới thiệu ASP.Net core 8.0

ASP.NET Core 8.0 là phiên bản mới nhất của nền tảng phát triển ứng dụng web do Microsoft phát triển, mang lại hiệu năng vượt trội, tính linh hoạt cao và khả năng xây dựng các ứng dụng hiện đại từ Web, API đến dịch vụ đám mây [3].



Hình 2.3 Logo ASP.NET Core 8.0

### 2.3.1. Đặc điểm nổi bật của ASP.NET Core 8.0

#### Hiệu năng cực cao và tối ưu bộ nhớ

ASP.NET Core 8.0 được đánh giá là một trong những framework web nhanh nhất hiện nay.

Các cải tiến bao gồm:

- Tối ưu hóa xử lý HTTP và routing.
- Giảm phân bổ bộ nhớ (memory allocation).

- Hỗ trợ Ahead-Of-Time compilation (AOT) giúp rút ngắn thời gian khởi chạy và giảm dung lượng ứng dụng.
- Hiệu suất vượt trội, phù hợp cho các hệ thống lớn và ứng dụng nhiều người dùng.

Nhờ đó, ASP.NET Core 8.0 được các doanh nghiệp lớn lựa chọn cho hệ thống có yêu cầu tốc độ cao và độ ổn định lớn.

### **Mã nguồn mở, đa nền tảng**

ASP.NET Core là mã nguồn mở (open-source) và hoạt động trên nhiều hệ điều hành:

- Windows
- Linux
- MacOS

Việc sử dụng hoàn toàn miễn phí, có thể được triển khai trên các server rẻ hơn như Linux để giảm chi phí vận hành.

### **Hỗ trợ kiến trúc hiện đại: API, Microservices, Cloud**

ASP.NET Core 8.0 hỗ trợ xây dựng:

- RESTful API
- Web Application
- Microservices
- Ứng dụng cloud-native
- Ứng dụng chạy trên Docker và Kubernetes

Framework cũng tích hợp mạnh mẽ với Microsoft Azure, AWS, Google Cloud.

### **Bảo mật toàn diện**

ASP.NET Core 8.0 tích hợp nhiều cơ chế bảo mật:

- Authentication & Authorization
- Role-based và Policy-based authorization
- Bảo mật JWT cho RESTful API

- Bảo vệ chống XSS, CSRF, CORS
- Hỗ trợ HTTPS mặc định

Hệ thống bảo mật mạnh mẽ và dễ mở rộng.

### **Hệ thống Dependency Injection tích hợp sẵn**

ASP.NET Core cung cấp DI built-in, không cần thư viện ngoài:

- Giúp quản lý vòng đời đối tượng (Scoped, Transient, Singleton)
- Giảm phụ thuộc, dễ test, dễ mở rộng

Đây là nền tảng quan trọng để xây dựng ứng dụng theo chuẩn hiện đại.

### **Khả năng mở rộng mạnh mẽ và linh hoạt**

ASP.NET Core 8.0 cho phép:

- Tùy chỉnh middleware
- Viết custom logging, custom services
- Tích hợp dễ dàng với thư viện ngoài như:
  - Entity Framework Core
  - SignalR
  - gRPC
  - Hangfire
  - Serilog

Framework linh hoạt, phù hợp với dự án từ nhỏ đến lớn.

### **Tích hợp tốt với Entity Framework Core 8**

EF Core 8 hỗ trợ:

- LINQ mạnh mẽ
- Migrations
- Tracking dữ liệu tối ưu
- Hỗ trợ nhiều loại cơ sở dữ liệu: PostgreSQL, SQL Server, MySQL, SQLite...

Giúp thao tác dữ liệu nhanh và tiện.

### **Cải tiến Blazor (full-stack .NET)**

.NET 8 giới thiệu Blazor mới:

- Chạy client + server bằng C#
- Hỗ trợ SSR (Server-side rendering) nhanh
- Hỗ trợ SPA (Single Page Application)

Blazor trở thành lựa chọn thay thế React/Angular trong một số dự án.

ASP.NET Core có cộng đồng lập trình viên mạnh trên toàn cầu, tài liệu chi tiết từ Microsoft, code mẫu phong phú, hỗ trợ lâu dài.

#### **2.3.2. Ưu điểm và lợi ích của ASP.NET Core 8.0**

##### **Ưu điểm**

ASP.NET Core 8.0 mang nhiều ưu điểm nổi bật như hiệu năng xử lý cao, khả năng chạy đa nền tảng và kiến trúc hiện đại phù hợp với xu hướng phát triển ứng dụng web. Framework được tối ưu hóa tốc độ, sử dụng bộ nhớ hiệu quả và hỗ trợ Minimal API giúp rút ngắn thời gian phát triển. Bên cạnh đó, ASP.NET Core còn tích hợp sẵn Dependency Injection, hệ thống Middleware linh hoạt và bảo mật mạnh mẽ với các cơ chế như JWT, HTTPS, chống XSS và CSRF. Những ưu điểm này giúp framework trở thành lựa chọn mạnh mẽ, ổn định và dễ mở rộng cho nhiều loại dự án.

##### **Lợi ích**

Việc sử dụng ASP.NET Core 8.0 mang lại nhiều lợi ích thiết thực như giảm chi phí triển khai nhờ khả năng hoạt động trên cả Windows, Linux và macOS. Framework đảm bảo hiệu năng ổn định, hỗ trợ nhiều người dùng truy cập đồng thời, thích hợp cho các ứng dụng doanh nghiệp. Ngoài ra, nhờ cấu trúc rõ ràng và hỗ trợ Minimal API, việc phát triển và bảo trì dự án trở nên nhanh chóng và hiệu quả hơn. Hệ thống bảo mật tích hợp cũng giúp ứng dụng an toàn hơn trong môi trường Internet. Với cộng đồng hỗ trợ lớn và tài liệu phong phú, ASP.NET Core 8.0 giúp lập trình viên dễ dàng tiếp cận, nâng cao năng suất và xây dựng các hệ thống linh hoạt, bền vững.

## CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### 3.1. Khảo sát yêu cầu hệ thống

Trong bối cảnh nhu cầu tập luyện thể thao và chăm sóc sức khỏe ngày càng tăng cao, các phòng Gym không chỉ đơn thuần là nơi rèn luyện thể chất mà còn là môi trường dịch vụ hiện đại, đòi hỏi sự quản lý chuyên nghiệp. Tuy nhiên, thực tế cho thấy nhiều phòng Gym hiện nay vẫn sử dụng phương pháp quản lý thủ công bằng giấy tờ hoặc Excel, dẫn đến tình trạng dữ liệu phân tán, khó kiểm soát thông tin hội viên, khoá tập, thời gian hết hạn, hay theo dõi doanh thu. Điều này không chỉ gây mất thời gian, dễ xảy ra sai sót mà còn ảnh hưởng đến trải nghiệm dịch vụ của khách hàng.

Từ thực tiễn trên, nhu cầu đặt ra là “**Xây dựng website quản lý phòng Gym thông minh**”, cho phép tự động hóa quy trình quản lý, tối ưu hiệu quả vận hành và mang lại trải nghiệm tốt hơn cho hội viên.

Qua khảo sát thực tế từ các chủ phòng Gym cũng như hội viên đang tham gia tập luyện, có thể nhận thấy rằng một hệ thống quản lý thông minh cần đáp ứng đồng thời yêu cầu từ phía quản trị và nhu cầu trải nghiệm của khách hàng. Các yêu cầu chính được tổng hợp bao gồm:

- **Đăng ký/đăng nhập và quản lý tài khoản hội viên:** Hệ thống cần cho phép người dùng dễ dàng tạo tài khoản, đăng nhập và cập nhật thông tin cá nhân. Đồng thời, đảm bảo tính bảo mật dữ liệu để tạo sự tin tưởng cho người dùng.
- **Quản lý gói tập:** Cung cấp chức năng cho phép hội viên xem danh sách các khoá tập và đăng ký các gói tập phù hợp với nhu cầu của mình. Thông tin về tên khoá tập, thời lượng, giá cả và huấn luyện viên cần được hiển thị đầy đủ và chính xác.
- **Quản lý hội viên:** Lưu trữ thông tin chi tiết của hội viên bao gồm hồ sơ cá nhân, thông tin liên lạc.
- **Quản lý nhân viên(PT):** Cho phép phân công lịch làm việc, quản lý thông tin cá nhân và hỗ trợ đặt lịch hẹn giữa PT với hội viên.
- **Thanh toán và thống kê doanh thu:** Hệ thống cần tích hợp nhiều hình thức thanh toán online/offline. Đồng thời, cung cấp báo cáo thống kê chi tiết về doanh thu, số lượng hội viên đăng ký, tỷ lệ sử dụng dịch vụ... để hỗ trợ việc ra quyết định quản lý.

- **Chatbot:** Hỗ trợ khách hàng trò chuyện trực tiếp với hệ thống để được gợi ý giá và các tên khoá tập phù hợp hoặc các thông tin cần thiết khác.
- **Giao diện thân thiện, đa nền tảng:** Thiết kế giao diện trực quan, dễ sử dụng, đồng thời tương thích với nhiều thiết bị như máy tính, máy tính bảng, điện thoại thông minh để mang lại sự tiện lợi tối đa.

Việc khảo sát và tổng hợp các yêu cầu trên giúp định hình phạm vi phát triển hệ thống, đảm bảo sản phẩm không chỉ đáp ứng đúng nhu cầu quản lý mà còn nâng cao trải nghiệm cho cả người quản lý và hội viên.

### 3.2. Phân tích yêu cầu hệ thống

Dựa trên kết quả khảo sát nhu cầu thực tế từ các chủ phòng Gym, huấn luyện viên và hội viên, hệ thống được phân tích và chia thành hai nhóm yêu cầu chính: **yêu cầu chức năng** và **yêu cầu phi chức năng**. Việc phân loại này giúp quá trình thiết kế, phát triển và triển khai hệ thống diễn ra hiệu quả, đồng thời đáp ứng được đúng mục tiêu đặt ra ban đầu.

#### 3.2.1. Yêu cầu chức năng

- Chức năng đăng nhập
  - Hội viên: Khi hội viên đăng nhập vào website, hệ thống hiển thị thông tin cá nhân và các chức năng dành riêng cho hội viên như: xem khóa tập, lịch hẹn, giỏ hàng, lịch sử thanh toán,...
  - Nhân viên (PT): Khi PT đăng nhập, hệ thống hiển thị danh sách khoá học do PT phụ trách
  - Admin: Khi Admin đăng nhập, hệ thống hiển thị đầy đủ chức năng quản trị như: quản lý khoản người dùng, quản lý khóa tập và dịch vụ, , quản lý lịch hẹn và xem báo cáo thống kê.
- Chức năng đặt lịch hẹn
  - Hội viên yêu cầu đặt lịch trực tiếp tại quầy lễ tân/phòng gym.
  - Nhân viên quản trị (Admin) là người thao tác tạo lịch hẹn trên hệ thống.
  - Admin chọn hội viên, PT, thời gian, dịch vụ/khoá tập.
  - Hệ thống kiểm tra trùng lịch của PT trước khi xác nhận.
  - Hội viên có thể đăng nhập để xem lịch đã được sắp xếp
- Chức năng quản lý nhân viên (PT)

- Admin có thể thêm mới PT, chỉnh sửa hồ sơ, cập nhật thông tin hoặc xóa PT khỏi hệ thống.
- Thông tin PT được hiển thị khi hội viên chọn khóa tập hoặc đặt lịch hẹn.
- Chức năng quản lý khóa tập & dịch vụ
  - Admin có thể thêm, sửa, xóa các khóa tập, gói dịch vụ như: giảm cân, tăng cơ, yoga, boxing,...
  - Mỗi khóa tập có thông tin chi tiết: mô tả, thời lượng, giá, hình ảnh, PT phụ trách.
  - Hệ thống cho phép hội viên dễ dàng tìm và lựa chọn khóa tập phù hợp.
- Chức năng quản lý hội viên
  - Admin có thể xem thông tin hội viên, thêm, sửa, xóa cập nhật hồ sơ.
- Chức năng Chatbot hỗ trợ
  - Khi hội viên có thắc mắc, chatbot sẽ hỗ trợ trả lời các câu hỏi như: giờ mở cửa, giá khóa tập, tên khóa tập, số điện thoại, địa chỉ...
  - Chatbot giúp giảm tải cho Admin và cung cấp thông tin nhanh cho người dùng.
- Chức năng xem lịch hẹn
  - Hội viên có thể xem các lịch hẹn đã đặt với PT, bao gồm thời gian, PT phụ trách.
- Chức năng thanh toán
  - Hội viên có thể thực hiện thanh toán cho các khóa tập đã đăng ký.
  - Hệ thống hỗ trợ hai hình thức thanh toán: trực tuyến và trực tiếp tại quầy.
  - Sau khi thanh toán thành công, hệ thống cập nhật trạng thái hoá đơn.
- Chức năng xem lịch sử các khóa tập đã mua
  - Hội viên có thể xem danh sách các khóa tập mà mình đã đăng ký/mua trước đó.
  - Thông tin hiển thị bao gồm: tên khóa tập, thời lượng, huấn luyện viên...
  - Hệ thống giúp hội viên theo dõi lịch sử tham gia các khóa tập một cách tiện lợi.
- Chức năng giỏ hàng:
  - Hội viên có thể thêm các khóa tập vào giỏ hàng trước khi thanh toán.
  - Hệ thống hiển thị chi tiết các mục trong giỏ hàng, tổng giá, và cho phép chỉnh sửa số lượng hoặc xóa mục.
  - Sau khi xác nhận giỏ hàng, hội viên có thể tiến hành thanh toán trực tuyến hoặc tại quầy.
- Chức năng tìm kiếm:
  - Hội viên có thể tìm kiếm nhanh các khóa tập.

- Admin có thể tìm kiếm các khóa tập, dịch vụ, huấn luyện viên hoặc các từ khóa liên quan.

### **3.2.2. Yêu cầu phi chức năng**

- Hiệu năng: Hệ thống cần xử lý nhanh, phản hồi tức thì với các thao tác phổ biến.
- Bảo mật: Áp dụng mã hóa mật khẩu, xác thực an toàn và phân quyền chặt chẽ để tránh truy cập trái phép.
- Khả năng mở rộng: Đảm bảo hệ thống có thể hoạt động ổn định ngay cả khi số lượng hội viên tăng cao.
- Tính thân thiện với người dùng: Giao diện phải đơn giản, dễ sử dụng, trực quan và tương thích trên nhiều thiết bị.

Quá trình phân tích yêu cầu giúp làm rõ kiến trúc tổng thể của hệ thống quản lý phòng Gym, đồng thời định hướng cho các bước thiết kế và lập trình tiếp theo. Việc phân tách giữa yêu cầu chức năng và phi chức năng cho phép nhóm phát triển xác định đúng các ưu tiên cần thực hiện, từ đó lựa chọn giải pháp phù hợp, tiết kiệm chi phí và thời gian.

Nhờ đó, hệ thống không chỉ đáp ứng nhu cầu thực tế của các phòng Gym trong việc quản lý hội viên, nhân viên, khóa tập và lịch trình, mà còn đảm bảo các yếu tố quan trọng như hiệu năng, bảo mật, khả năng mở rộng và trải nghiệm thân thiện cho người dùng. Đây chính là nền tảng để hệ thống vận hành ổn định, bền vững và mang lại hiệu quả lâu dài cho các trung tâm thể hình.

### **3.2.3. Khảo sát, lấy yêu cầu**

Hệ thống quản lý phòng Gym cần đảm bảo các yêu cầu chính:

- Quản lý người dùng: Hệ thống quản lý thông tin của các hội viên và nhân viên.
- Quản lý khóa tập, dịch vụ: Hệ thống quản lý thông tin khóa tập và dịch vụ, cho phép admin thêm, sửa, xóa các khóa tập, gói dịch vụ và lịch học.
- Quản lý lịch hẹn: Cho phép admin tạo lịch hẹn giữa hội viên và nhân viên.
- Thống kê, báo cáo doanh thu: Hệ thống cung cấp các báo cáo về doanh thu, số lượng hội viên, số lượng khóa tập.
- Tương tác hội viên: Hội viên có thể đăng ký, đăng nhập, xem khóa tập, thanh toán, và sử dụng chatbot hỗ trợ...

## **3.3. Phân tích, đặc tả yêu cầu**

Hệ thống quản lý phòng Gym được xây dựng nhằm phục vụ hai nhóm đối tượng chính: người dùng (hội viên) và quản trị viên. Mỗi nhóm có các yêu cầu và chức năng riêng biệt, được phân tích cụ thể như sau:

### **3.3.1. Người dùng (Hội viên)**

Người dùng là những khách hàng đăng ký dịch vụ tại phòng Gym. Các yêu cầu chính đối với nhóm này bao gồm:

- **Đăng ký tài khoản:**
  - Người dùng có thể tạo tài khoản mới bằng cách cung cấp các thông tin cơ bản như email, mật khẩu.
  - Sau khi đăng ký thành công, hệ thống sẽ lưu trữ thông tin hội viên để phục vụ việc quản lý lâu dài.
- **Đăng nhập hệ thống:**
  - Người dùng sử dụng email và mật khẩu để đăng nhập.
  - Hệ thống đảm bảo tính bảo mật, chỉ cho phép người dùng hợp lệ truy cập.
- **Xem và cập nhật thông tin cá nhân:**
  - Người dùng có thể truy cập trang hồ sơ cá nhân để xem các thông tin của mình (họ tên, số điện thoại, email, địa chỉ...).
  - Hệ thống cho phép chỉnh sửa một số thông tin cơ bản như số điện thoại, email, địa chỉ, cân nặng, chiều cao.
- **Đăng ký khoá tập:**
  - Người dùng có thể xem danh sách các khoá tập và đăng ký, bao gồm tên khoá tập, thời lượng, giá, nhân viên.
- **Thanh toán:**
  - Người dùng có thể thực hiện thanh toán trực tuyến hoặc trực tiếp cho các khoá tập đã đăng ký thông qua các phương thức được hệ thống hỗ trợ.
- **Chat bot:**
  - Hệ thống cung cấp chat bot để hỗ trợ tư vấn trực tuyến, trả lời các thắc mắc và hướng dẫn người dùng.
- **Đặt lịch hẹn trực tiếp:**
  - Hội viên yêu cầu đặt lịch hẹn tại quầy lễ tân hoặc phòng Gym.

- Nhân viên quản trị (Admin) thực hiện tạo lịch hẹn trên hệ thống và lưu trữ thông tin lịch hẹn.

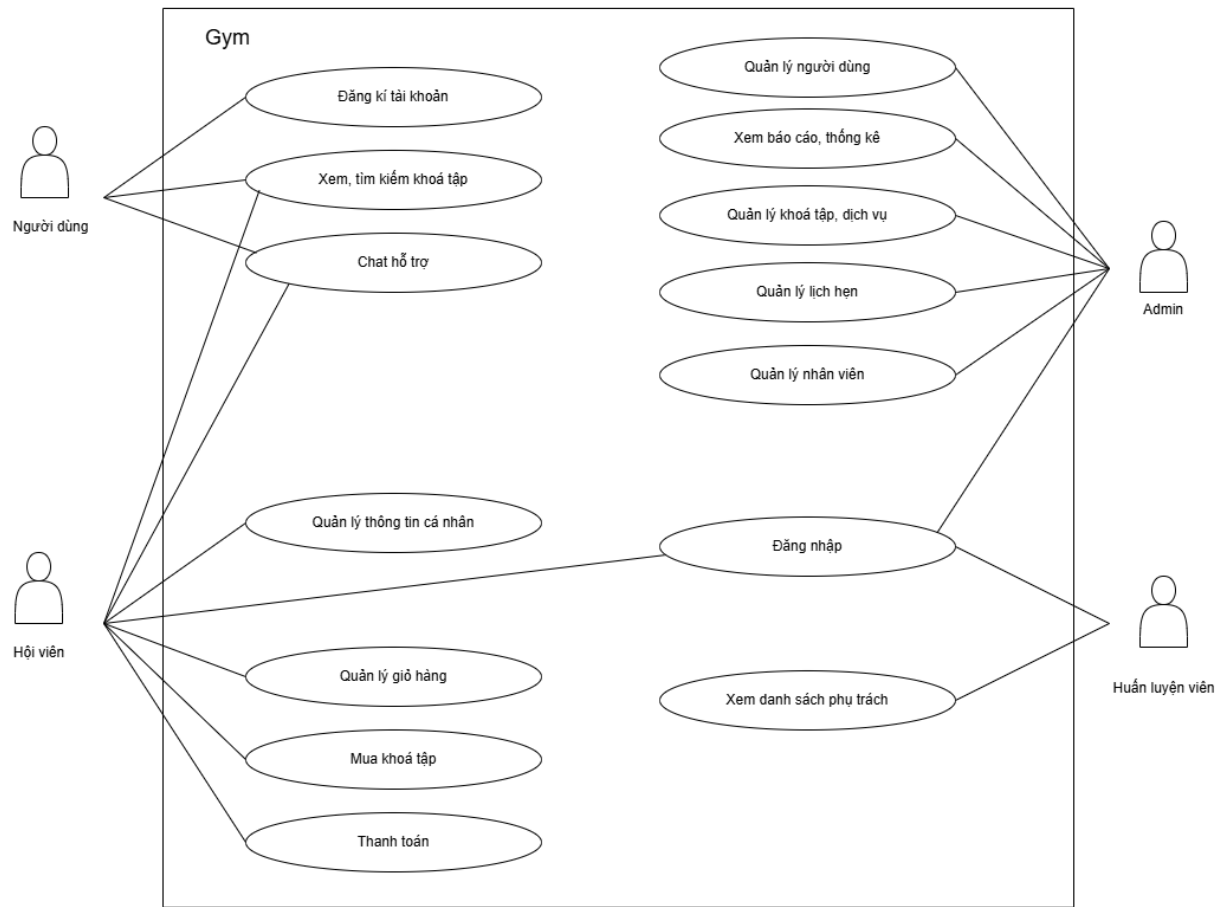
### 3.3.2. Quản trị viên(Admin)

Quản trị viên là những người vận hành và quản lý toàn bộ hệ thống. Các chức năng chính của quản trị viên bao gồm:

- **Quản lý hội viên:**
  - Xem danh sách hội viên kèm theo thông tin cá nhân.
  - Thêm mới, chỉnh sửa hoặc xóa tài khoản hội viên (trong trường hợp vi phạm nội quy hoặc yêu cầu từ hội viên).
- **Quản lý nhân viên:**
  - Lưu trữ thông tin của nhân viên, bao gồm họ tên, địa chỉ email, phân công cho các khoá tập.
- **Quản lý khoá tập, dịch vụ:**
  - Tạo mới, chỉnh sửa, cập nhật hoặc xóa các khoá tập, dịch vụ.
  - Định nghĩa thông tin chi tiết cho từng khoá tập: tên khoá tập, thời lượng, giá, các dịch vụ đi kèm.
- **Thống kê và báo cáo:**
  - Hệ thống cung cấp báo cáo tổng quan theo từng thời kỳ (tháng, tuần), bao gồm tổng số lượng hội viên, số khoá tập, doanh thu.
  - Cho phép quản trị viên đánh giá tình hình hoạt động, từ đó đưa ra các quyết định quản lý và phát triển phù hợp.

### 3.4. Biểu đồ use case.

### 3.4.1. Biểu đồ use-case tổng quát.



Hình 3.1 Biểu đồ Use Case tổng quát  
Danh sách Use Case theo từng tác nhân

#### Người dùng

- Đăng kí tài khoản: Cho phép người dùng mới tạo một tài khoản để trở thành Hội viên.
- Xem, tìm kiếm khoá tập: Cho phép người dùng tra cứu thông tin chi tiết về các khoá tập và dịch vụ mà Gym cung cấp.
- Chat hỗ trợ: Cho phép người dùng tương tác với hệ thống hỗ trợ tự động để nhận câu trả lời cho các thắc mắc chung về dịch vụ.

#### Hội viên

- Đăng nhập: Chức năng cơ sở cho phép Hội viên (cũng như Admin và Huấn luyện viên) truy cập vào hệ thống bằng tài khoản của mình.

- Quản lý thông tin cá nhân: Cho phép hội viên xem và cập nhật các thông tin cá nhân (họ tên, địa chỉ, mật khẩu,...) của mình.
- Mua khoá tập: Cho phép hội viên lựa chọn các khoá tập và thêm vào giỏ hàng để tiến hành mua.
- Quản lý giỏ hàng: Cho phép hội viên thêm, xóa, hoặc điều chỉnh số lượng các mục đã chọn trong giỏ hàng.
- Thanh toán: Cho phép hội viên thực hiện các thao tác thanh toán cho các giao dịch mua khóa tập.

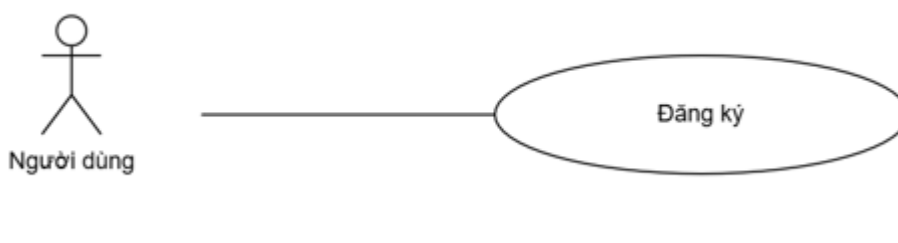
### **Huấn luyện viên**

- Đăng nhập: Chức năng cơ sở cho phép huấn luyện viên truy cập vào hệ thống bằng tài khoản của mình.
- Xem danh sách phụ trách: Cho phép huấn luyện viên kiểm tra lịch làm việc, danh sách học viên.

### **Admin**

- Đăng nhập: Chức năng cơ sở cho Admin truy cập vào hệ thống bằng tài khoản của mình.
- Quản lý người dùng: Cho phép Admin thêm, sửa, xóa, hoặc vô hiệu hóa tài khoản của tất cả người dùng trong hệ thống.
- Quản lý nhân viên: Cho phép Admin thêm, sửa, xóa thông tin cá nhân.
- Quản lý khoá tập, dịch vụ: Cho phép Admin thêm mới, cập nhật, hoặc ngừng cung cấp thông tin về các khoá tập, dịch vụ và giá cả.
- Quản lý lịch hẹn: Cho phép Admin thêm, sửa, xóa, hủy lịch hẹn của hội viên.
- Xem báo cáo, thống kê: Cho phép Admin theo dõi, kiểm soát và ghi nhận các giao dịch tài chính, thu chi của phòng Gym, và tình hình chung của hệ thống.

### 3.4.2. Biểu đồ use case đăng kí.

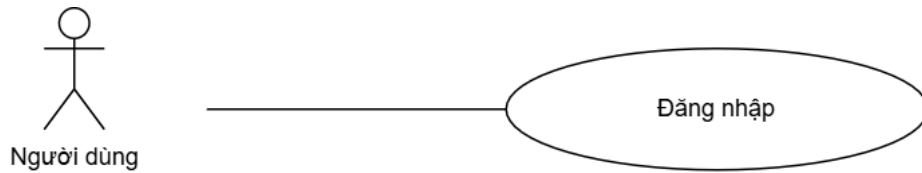


Hình 3.2 Biểu đồ Use Case đăng kí

Bảng 3.1 Đặc tả chi tiết Use Case đăng kí.

| Tên Use Case          | Đăng ký tài khoản                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Người dùng (Khách hàng tiềm năng)                                                                                                                                                                                                                                                                                                                                                                                                    |
| <b>Tiền điều kiện</b> | Người dùng truy cập vào trang web chính thức của GymFitness.                                                                                                                                                                                                                                                                                                                                                                         |
| <b>Hậu điều kiện</b>  | Người dùng đăng ký thành công tài khoản thành viên.                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Kích hoạt</b>      | Người dùng chọn "Đăng ký".                                                                                                                                                                                                                                                                                                                                                                                                           |
| <b>Kịch bản chính</b> | <ol style="list-style-type: none"> <li>1. Người dùng truy cập trang chủ của web GymFitness.</li> <li>2. Người dùng click vào nút/liên kết 'Đăng ký'.</li> <li>3. Hệ thống hiển thị giao diện đăng ký.</li> <li>4. Người dùng điền thông tin cá nhân (Tên, Email, Mật khẩu).</li> <li>5. Người dùng click nút 'Đăng ký'.</li> <li>6. Hệ thống xử lý thông tin.</li> <li>7. Hệ thống hiển thị thông báo đăng ký thành công.</li> </ol> |
| <b>Ngoại lệ</b>       | <ol style="list-style-type: none"> <li>4.1 Không nhập đủ thông tin bắt buộc → Hiển thị lỗi.</li> <li>5.1 Email đã tồn tại → Yêu cầu dùng Email khác.</li> <li>5.2 Mật khẩu yếu → Hiển thị yêu cầu sửa mật khẩu.</li> </ol>                                                                                                                                                                                                           |

### 3.4.3. Biểu đồ Use Case đăng nhập.

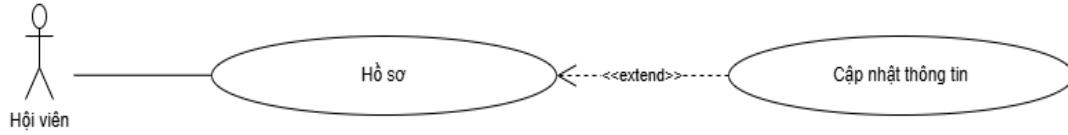


Hình 3.3 Biểu đồ Use Case đăng nhập

Bảng 3.2 Đặc tả chi tiết Use Case đăng nhập.

| Tên Use Case   | Đăng nhập                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Tác nhân chính | Người dùng (Hội viên/Khách hàng)                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Tiền điều kiện | Người dùng đã có tài khoản hợp lệ trên hệ thống GymFitness                                                                                                                                                                                                                                                                                                                                                                                                  |
| Hậu điều kiện  | Người dùng đăng nhập thành công và truy cập vào trang chủ/tài khoản cá nhân                                                                                                                                                                                                                                                                                                                                                                                 |
| Kích hoạt      | Người dùng chọn nút “Đăng nhập”                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Kịch bản chính | <ol style="list-style-type: none"> <li>1. Người dùng truy cập trang web GymFitness.</li> <li>2. Click vào nút “Đăng nhập”.</li> <li>3. Hệ thống hiển thị giao diện đăng nhập.</li> <li>4. Người dùng nhập Email và Mật khẩu.</li> <li>5. Nhấn nút “Đăng nhập”.</li> <li>6. Hệ thống kiểm tra thông tin.</li> <li>7. Hệ thống xác thực đúng → Đăng nhập thành công.</li> <li>8. Hệ thống chuyển hướng người dùng đến trang chủ/tài khoản cá nhân.</li> </ol> |
| Ngoại lệ       | <ol style="list-style-type: none"> <li>6.1. Không nhập đủ Email hoặc Mật khẩu → Hiển thị thông báo yêu cầu nhập đầy đủ thông tin.</li> <li>6.2. Email không tồn tại → Hiển thị "Email không tồn tại".</li> <li>6.3. Mật khẩu sai → Hiển thị "Sai mật khẩu".</li> </ol>                                                                                                                                                                                      |

#### 3.4.4. Biểu đồ Use Case cập nhật thông tin hội viên.

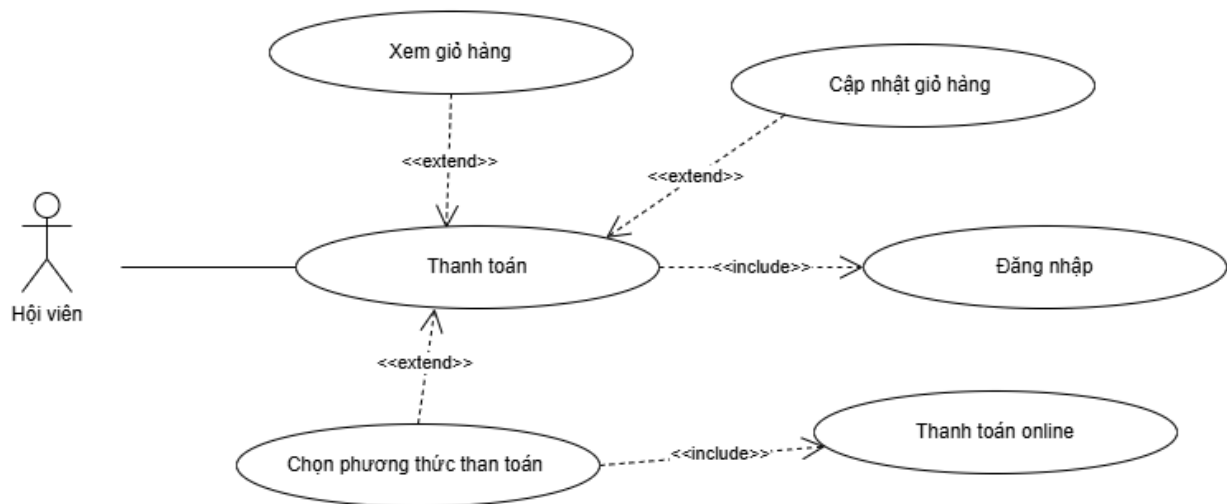


Hình 3.4 Biểu đồ Use Case cập nhật thông tin cá nhân.

Bảng 3.3 Đặc tả chi tiết Use Case cập nhật thông tin cá nhân.

| Tên Use Case          | Cập nhật thông tin hội viên                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Hội viên (Người dùng đã đăng nhập)                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <b>Tiền điều kiện</b> | Hội viên đã đăng nhập vào hệ thống GymFitness                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Hậu điều kiện</b>  | Thông tin cá nhân của hội viên được cập nhật thành công trong hệ thống                                                                                                                                                                                                                                                                                                                                                                                                            |
| <b>Kích hoạt</b>      | Hội viên vào trang hồ sơ cá nhân                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| <b>Kịch bản chính</b> | <ol style="list-style-type: none"> <li>1. Hội viên truy cập trang hồ sơ cá nhân.</li> <li>2. Chọn nút “Chỉnh sửa thông tin”.</li> <li>3. Hệ thống hiển thị form thông tin cá nhân hiện tại.</li> <li>4. Hội viên chỉnh sửa các thông tin cần cập nhật (Tên, Email, Số điện thoại, Địa chỉ, ...).</li> <li>5. Nhấn nút “Lưu” .</li> <li>6. Hệ thống kiểm tra tính hợp lệ của dữ liệu.</li> <li>7. Hệ thống lưu thông tin mới và hiển thị thông báo cập nhật thành công.</li> </ol> |
| <b>Ngoại lệ</b>       | <ul style="list-style-type: none"> <li>- Hội viên để trống trường bắt buộc → Hệ thống hiển thị “Vui lòng điền đầy đủ thông tin”.</li> <li>- Hệ thống lỗi khi lưu dữ liệu → Hiển thị thông báo “Cập nhật thất bại, thử lại sau”.</li> </ul>                                                                                                                                                                                                                                        |

### 3.4.5. Biểu đồ Use Case thanh toán.



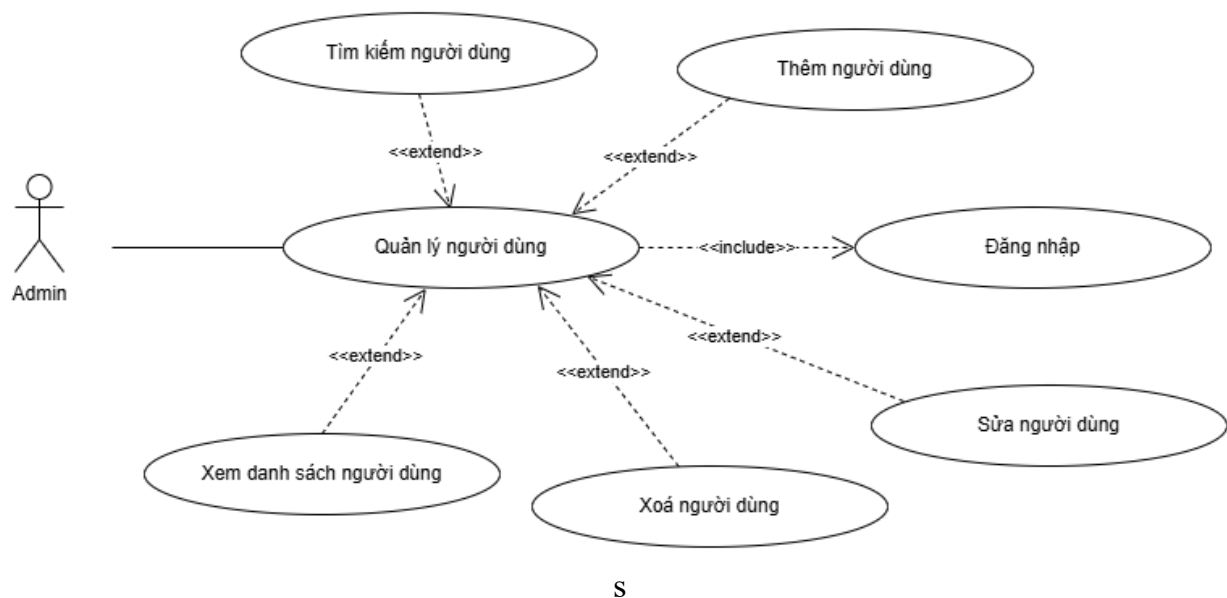
Hình 3.5 Biểu đồ Use Case thanh toán

Bảng 3.4 Đặt tả chi tiết Use Case thanh toán.

| Tên Use Case          | Thanh toán                                                                                                                                                                                                                                                                                 |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Hội viên                                                                                                                                                                                                                                                                                   |
| <b>Tiền điều kiện</b> | <ul style="list-style-type: none"> <li>- Hội viên đã có sản phẩm/gói tập trong giỏ hàng.</li> <li>- Hội viên đã đăng nhập (nếu chưa → hệ thống chuyển sang Use Case Đăng nhập).</li> </ul>                                                                                                 |
| <b>Hậu điều kiện</b>  | <ul style="list-style-type: none"> <li>- Đơn hàng được tạo thành công và lưu vào hệ thống.</li> <li>- Hội viên nhận thông báo thanh toán thành công.</li> </ul>                                                                                                                            |
| <b>Kích hoạt</b>      | Hội viên chọn chức năng “ <b>Thanh toán</b> ” từ giỏ hàng.                                                                                                                                                                                                                                 |
| <b>Kịch bản chính</b> | <ol style="list-style-type: none"> <li>1. Hội viên chọn “Thanh toán”.</li> <li>2. Hệ thống kiểm tra trạng thái đăng nhập.</li> <li>3. Hệ thống hiển thị giỏ hàng để hội viên kiểm tra.</li> <li>4. Hội viên có thể xem giỏ hàng.</li> <li>5. Hội viên có thể cập nhật giỏ hàng.</li> </ol> |

|                 |                                                                                                                                                                                                                                                                                                                        |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | 6. Hội viên chọn phương thức thanh toán.<br>7. Nếu chọn thanh toán online → hệ thống thanh toán online.<br>8. Hệ thống xử lý thanh toán.<br>9. Hệ thống hiển thị “Thanh toán thành công”.                                                                                                                              |
| <b>Ngoại lệ</b> | 4.1. Giỏ hàng trống → Hiển thị: “Giỏ hàng đang trống, không thể thanh toán”.<br>2.1. Chưa đăng nhập → Hệ thống yêu cầu đăng nhập.<br>7.1. Không chọn phương thức thanh toán → Hiển thị: “Vui lòng chọn phương thức thanh toán”.<br>7.2. Thanh toán online thất bại → Hiển thị: “Giao dịch thất bại, vui lòng thử lại”. |

### 3.4.6. Biểu đồ Use Case quản lý người dùng.

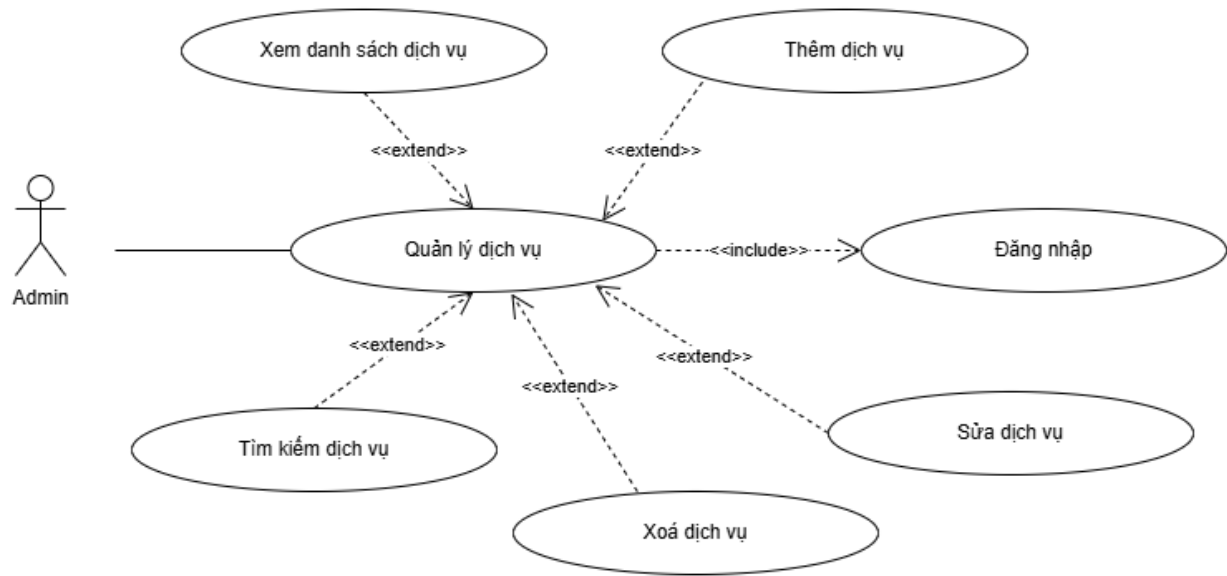


Hình 3.6 Biểu đồ Use Case quản lý người dùng.

Bảng 3.5 Đặt tả chi tiết Use Case quản lý người dùng.

| Tên Use Case          | Quản lý người dùng                                                                                                                                                                                                                                                                                                                                                                                 |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Admin                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Tiền điều kiện</b> | Admin đã đăng nhập và có quyền quản trị.                                                                                                                                                                                                                                                                                                                                                           |
| <b>Hậu điều kiện</b>  | Thông tin người dùng được thêm, cập nhật hoặc xóa thành công khỏi hệ thống.                                                                                                                                                                                                                                                                                                                        |
| <b>Kích hoạt</b>      | Admin chọn chức năng " <b>Quản lý người dùng</b> " trên giao diện quản trị.                                                                                                                                                                                                                                                                                                                        |
| <b>Kịch bản chính</b> | <ol style="list-style-type: none"> <li>Admin mở mục "Quản lý người dùng".</li> <li>Hệ thống hiển thị danh sách người dùng.</li> <li>Admin chọn một thao tác: Thêm mới / Sửa / Xóa / Tìm kiếm.</li> <li>Admin nhập hoặc chỉnh sửa thông tin.</li> <li>Admin nhấn Lưu hoặc Cập nhật.</li> <li>Hệ thống kiểm tra dữ liệu.</li> <li>Hệ thống lưu thay đổi và hiển thị thông báo thành công.</li> </ol> |
| <b>Ngoại lệ</b>       | <ul style="list-style-type: none"> <li>– Thiếu thông tin bắt buộc → Hệ thống báo lỗi.</li> <li>– Email hoặc số điện thoại trùng → Báo "Người dùng đã tồn tại".</li> <li>– Không thể xóa người dùng đang có lịch tập hoặc giao dịch → Hiển thị thông báo lỗi.</li> </ul>                                                                                                                            |

### 3.4.7. Biểu đồ Use Case quản lý dịch vụ.



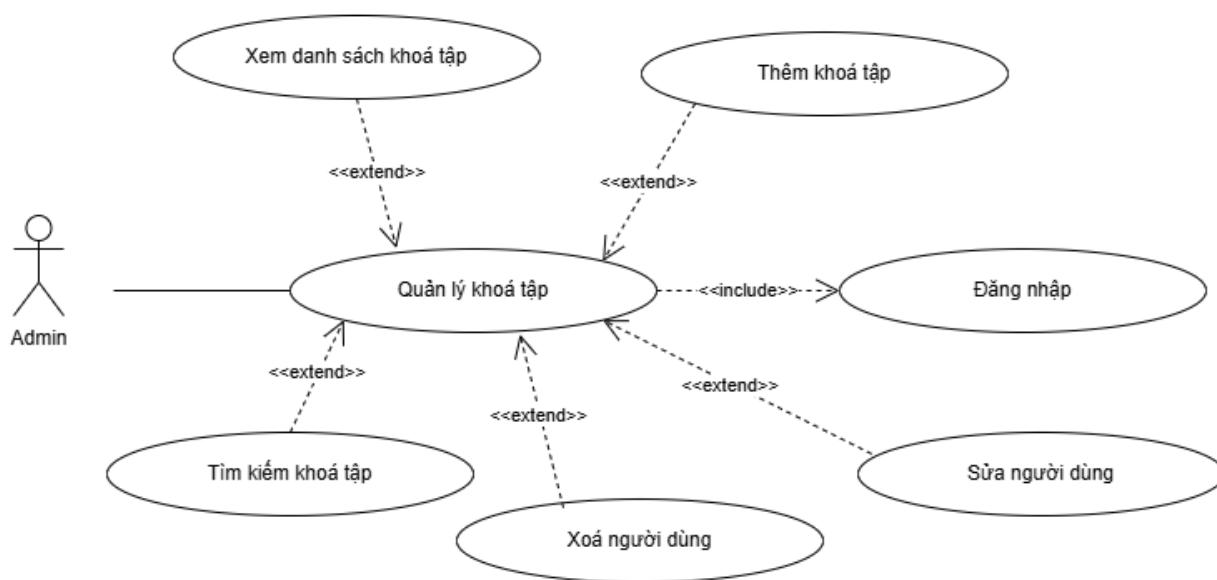
Hình 3.7 Biểu đồ Use Case quản lý dịch vụ

Bảng 3.6 Đặt tả chi tiết Use Case quản lý dịch vụ.

| Tên Use Case          | Quản lý dịch vụ                                                                                                                                                                                                                                                                                                                                                                                   |
|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Admin                                                                                                                                                                                                                                                                                                                                                                                             |
| <b>Tiền điều kiện</b> | Admin đã đăng nhập hệ thống và có quyền quản trị.                                                                                                                                                                                                                                                                                                                                                 |
| <b>Hậu điều kiện</b>  | Dịch vụ được thêm mới, cập nhật hoặc xóa thành công trong hệ thống.                                                                                                                                                                                                                                                                                                                               |
| <b>Kích hoạt</b>      | Admin chọn mục “ <b>Quản lý dịch vụ</b> ”.                                                                                                                                                                                                                                                                                                                                                        |
| <b>Kịch bản chính</b> | <ol style="list-style-type: none"> <li>Admin mở mục “Quản lý dịch vụ”.</li> <li>Hệ thống hiển thị danh sách các dịch vụ.</li> <li>Admin chọn hành động: Thêm mới / Sửa / Xóa.</li> <li>Admin nhập hoặc chỉnh sửa thông tin dịch vụ (Tên, giá, mô tả).</li> <li>Admin nhấn Lưu / Sửa / Xóa.</li> <li>Hệ thống kiểm tra dữ liệu.</li> <li>Hệ thống lưu và hiển thị thông báo thành công.</li> </ol> |
| <b>Ngoại lệ</b>       | <ul style="list-style-type: none"> <li>Thiếu thông tin dịch vụ → Báo lỗi.</li> <li>Tên dịch vụ trùng → Báo “Dịch vụ đã tồn tại”.</li> </ul>                                                                                                                                                                                                                                                       |

– Không thể xóa dịch vụ đang được khách hàng đăng ký → Hiện thị cảnh báo.

### 3.4.8. Biểu đồ Use Case quản lý khoá tập.



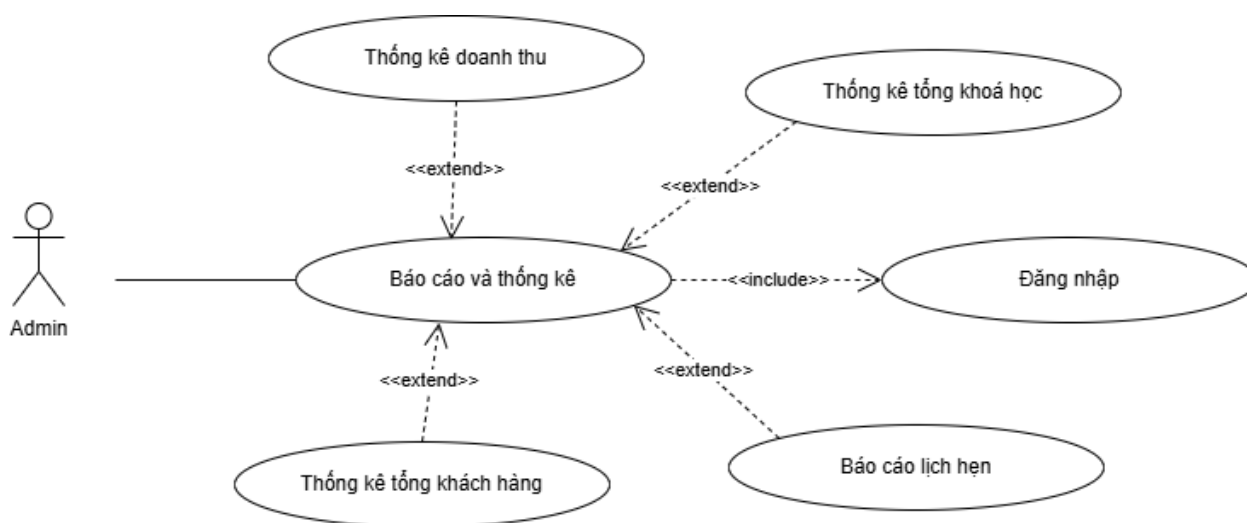
Hình 3.8 Biểu đồ Use Case quản lý khoá tập

Bảng 3.7 Đặt tả chi tiết Use Case quản lý khoá tập.

| Tên Use Case          | Quản lý khóa tập                                                                                                                                                                                                                                                                                               |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Admin                                                                                                                                                                                                                                                                                                          |
| <b>Tiền điều kiện</b> | Admin đã đăng nhập và có quyền quản lý khóa học.                                                                                                                                                                                                                                                               |
| <b>Hậu điều kiện</b>  | Khóa học được thêm mới, sửa hoặc xóa khỏi hệ thống.                                                                                                                                                                                                                                                            |
| <b>Kích hoạt</b>      | Admin chọn mục “ <b>Quản lý khóa tập</b> ”.                                                                                                                                                                                                                                                                    |
| <b>Kịch bản chính</b> | <ol style="list-style-type: none"> <li>1. Admin truy cập “Quản lý khóa tập”.</li> <li>2. Hệ thống hiển thị danh sách khóa tập.</li> <li>3. Admin chọn Thêm / Sửa / Xóa.</li> <li>4. Admin nhập thông tin khóa học.</li> <li>5. Admin nhấn Lưu hoặc Cập nhật.</li> <li>6. Hệ thống kiểm tra dữ liệu.</li> </ol> |

|                 |                                                                                                                                                        |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | 7. Hệ thống lưu thay đổi và hiển thị thông báo thành công.                                                                                             |
| <b>Ngoại lệ</b> | – Thiếu thông tin khóa tập → Hiển thị lỗi.<br>– Trùng tên khóa tập → Báo “Khóa tập đã tồn tại”.<br>– Không thể xóa khóa tập có học viên đang tham gia. |

### 3.4.9. Biểu đồ Use Case xem báo cáo và thống kê.



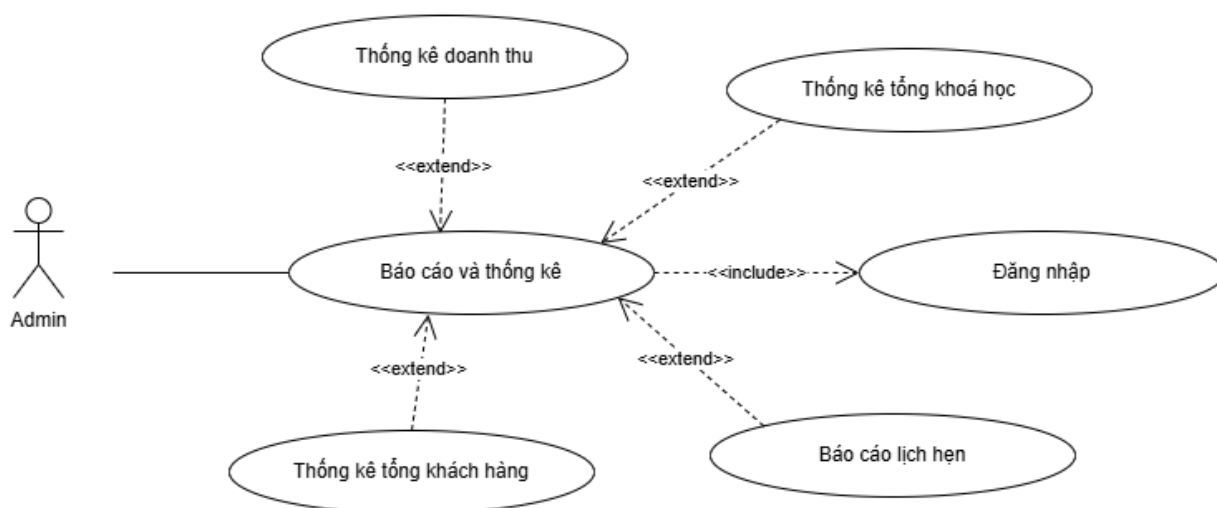
Hình 3.9 Biểu đồ Use Case xem báo cáo và thống kê

Bảng 3.8 Đặt tả chi tiết Use Case xem báo cáo và thống kê.

| Tên Use Case          | Xem báo cáo & thống kê                                                                                                                                       |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Admin                                                                                                                                                        |
| <b>Tiền điều kiện</b> | Admin đã đăng nhập và có quyền xem báo cáo.                                                                                                                  |
| <b>Hậu điều kiện</b>  | Báo cáo hiển thị đầy đủ, giúp admin nắm tình hình hoạt động.                                                                                                 |
| <b>Kích hoạt</b>      | Admin chọn mục <b>“Báo cáo – Thống kê”</b> .                                                                                                                 |
| <b>Kịch bản chính</b> | 1. Admin truy cập mục báo cáo.<br>2. Hệ thống hiển thị danh sách loại báo cáo: Doanh thu, Số lượng hội viên, Dịch vụ.<br>3. Admin chọn loại báo cáo cần xem. |

|                 |                                                                                                                                             |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------|
|                 | 4. Hệ thống truy vấn dữ liệu.<br>5. Hệ thống hiển thị bảng thống kê.<br>6. Admin có thể lọc theo thời gian (, tháng, tuần).                 |
| <b>Ngoại lệ</b> | – Không có dữ liệu trong khoảng thời gian chọn → Hiển thị “Không có dữ liệu”.<br>– Lỗi truy vấn dữ liệu → Hiển thị “Không thể tải báo cáo”. |

### 3.4.10. Biểu đồ Use Case xem lịch hẹn.



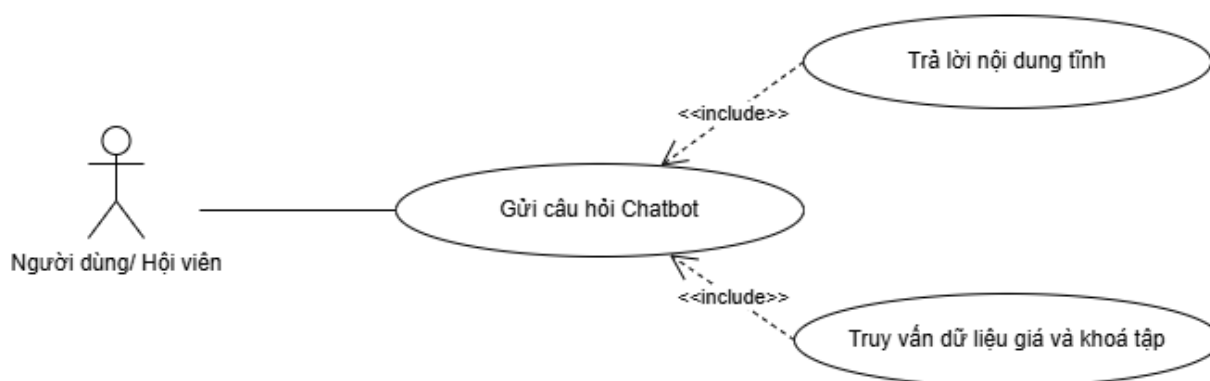
Hình 3.10 Biểu đồ Use Case quản lý lịch hẹn.

Bảng 3.9 Đặt tả chi tiết Use Case quản lý lịch hẹn.

| Tên Use Case          | Quản lý lịch hẹn                                                                                                     |
|-----------------------|----------------------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Admin                                                                                                                |
| <b>Tiền điều kiện</b> | Admin đã đăng nhập và có quyền quản lý lịch hẹn.                                                                     |
| <b>Hậu điều kiện</b>  | Lịch hẹn được thêm mới, sửa hoặc hủy khỏi hệ thống.                                                                  |
| <b>Kích hoạt</b>      | Admin chọn mục “Quản lý lịch hẹn”.                                                                                   |
| <b>Kịch bản chính</b> | 1. Admin truy cập “Quản lý lịch hẹn”.<br>2. Hệ thống hiển thị danh sách lịch hẹn.<br>3. Admin chọn Thêm / Sửa / Hủy. |

|                 |                                                                                                                                                                                                                             |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | <p>4. Admin nhập thông tin lịch hẹn (Học viên, PT, ngày, giờ, loại lớp).</p> <p>5. Admin nhấn Lưu hoặc Cập nhật.</p> <p>6. Hệ thống kiểm tra dữ liệu.</p> <p>7. Hệ thống lưu thay đổi và hiển thị thông báo thành công.</p> |
| <b>Ngoại lệ</b> | <p>– Thiếu thông tin lịch hẹn → Hiển thị lỗi.</p> <p>– Lịch hẹn trùng với thời gian PT hoặc học viên khác → Báo lỗi.</p> <p>– Không thể hủy lịch đã kết thúc hoặc học viên đã điểm danh.</p>                                |

### 3.4.11. Biểu đồ Use Case chat bot.



Hình 3.11 Biểu đồ Use Case chatbot.

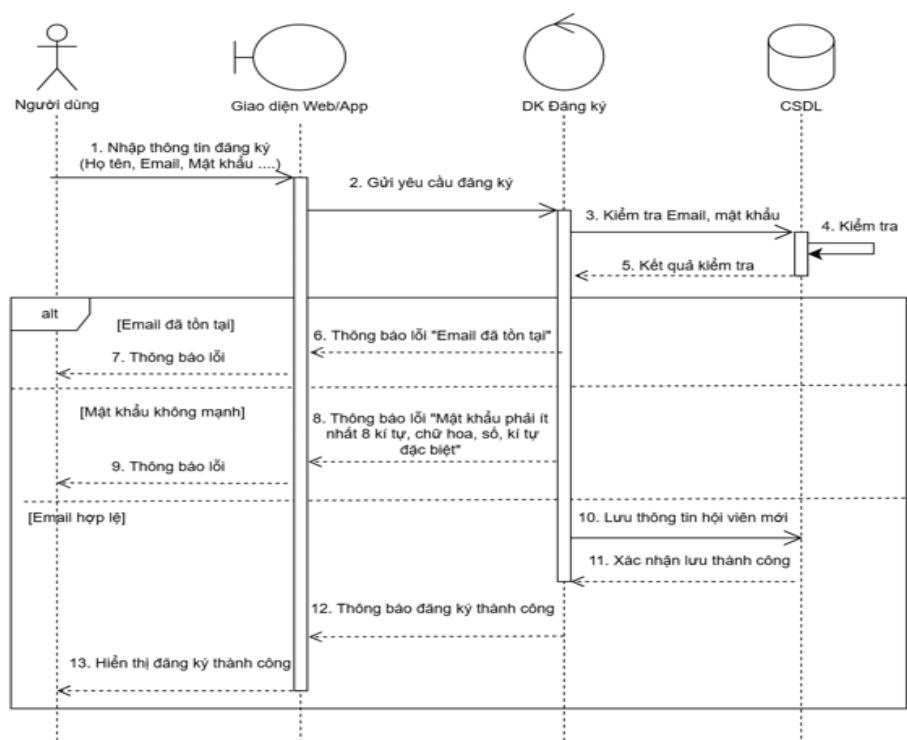
Bảng 3.10 Đặt tả chi tiết Use Case chatbot.

| Tên Use Case          | Chatbot                                                                                              |
|-----------------------|------------------------------------------------------------------------------------------------------|
| <b>Tác nhân chính</b> | Người dùng                                                                                           |
| <b>Tiền điều kiện</b> | Người dùng truy cập website/app và chatbot sẵn sàng hoạt động.                                       |
| <b>Hậu điều kiện</b>  | Người dùng nhận được thông tin tĩnh hoặc giá đúng theo cơ sở dữ liệu.                                |
| <b>Kích hoạt</b>      | Người dùng nhập câu hỏi về tỉnh hoặc giá vào chatbot.                                                |
| <b>Kịch bản chính</b> | <p>1. Người dùng gửi câu hỏi đến chatbot.</p> <p>2. Hệ thống nhận câu hỏi và phân tích nội dung.</p> |

|                 |                                                                                                                                                                                                                                  |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                 | <p>3. Hệ thống truy vấn cơ sở dữ liệu để lấy thông tin phù hợp.</p> <p>4. Hệ thống gửi câu trả lời lại cho người dùng.</p> <p>5. Người dùng nhận thông tin chính xác về câu hỏi tình hoặc giá.</p>                               |
| <b>Ngoại lệ</b> | <p>– Câu hỏi không rõ → Chatbot yêu cầu người dùng nhập lại.</p> <p>– Thông tin không có trong cơ sở dữ liệu → Chatbot thông báo “Không tìm thấy thông tin”.</p> <p>– Lỗi truy cập cơ sở dữ liệu → Chatbot báo lỗi hệ thống.</p> |

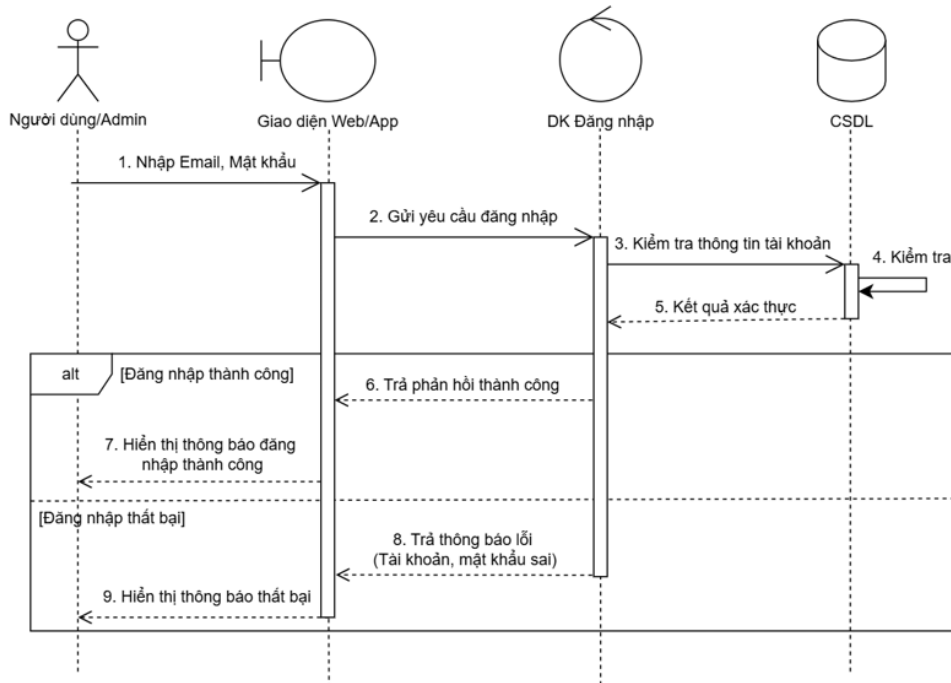
### 3.5. Biểu đồ tuần tự

#### 3.5.1. Biểu đồ tuần tự đăng ký.



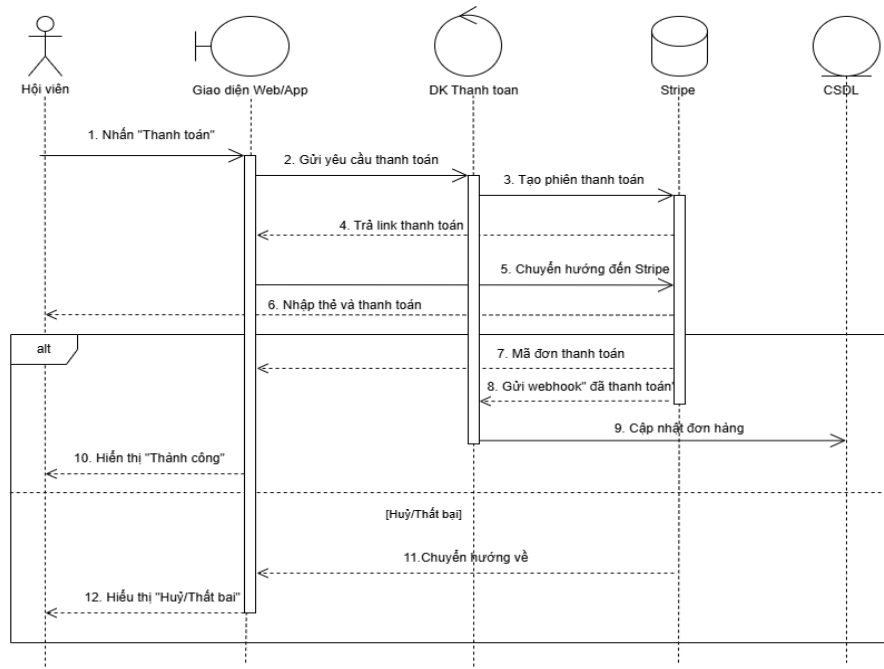
Hình 3.12 Biểu đồ tuần tự đăng ký.

### 3.5.2. Biểu đồ tuần tự đăng nhập.



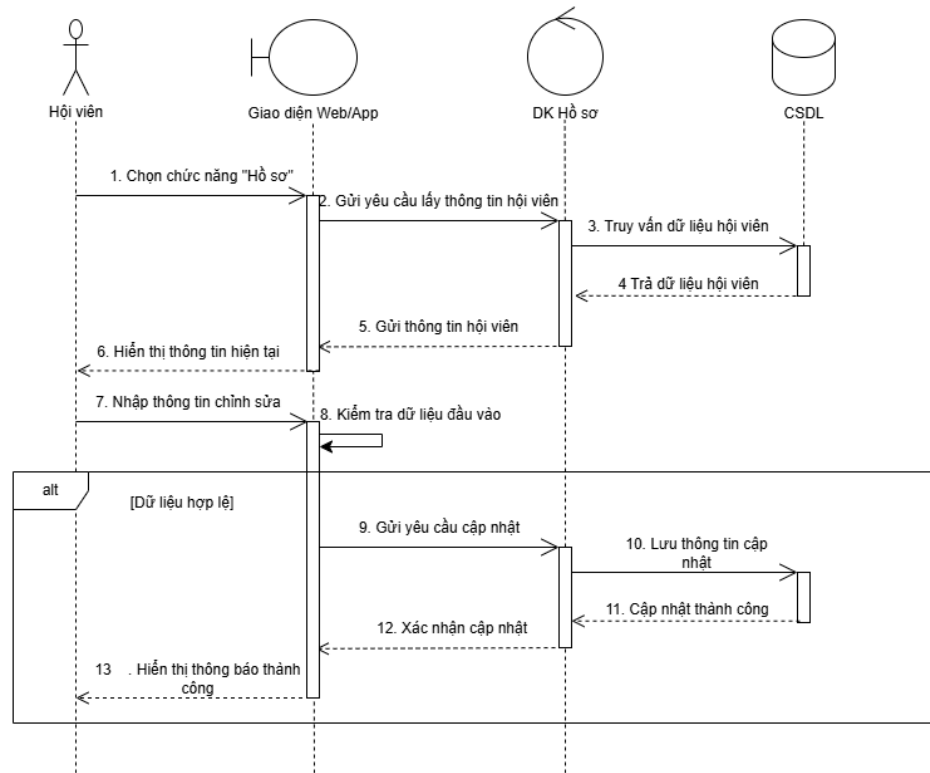
Hình 3.13 Biểu đồ tuần tự đăng nhập.

### 3.5.3. Biểu đồ tuần tự thanh toán.



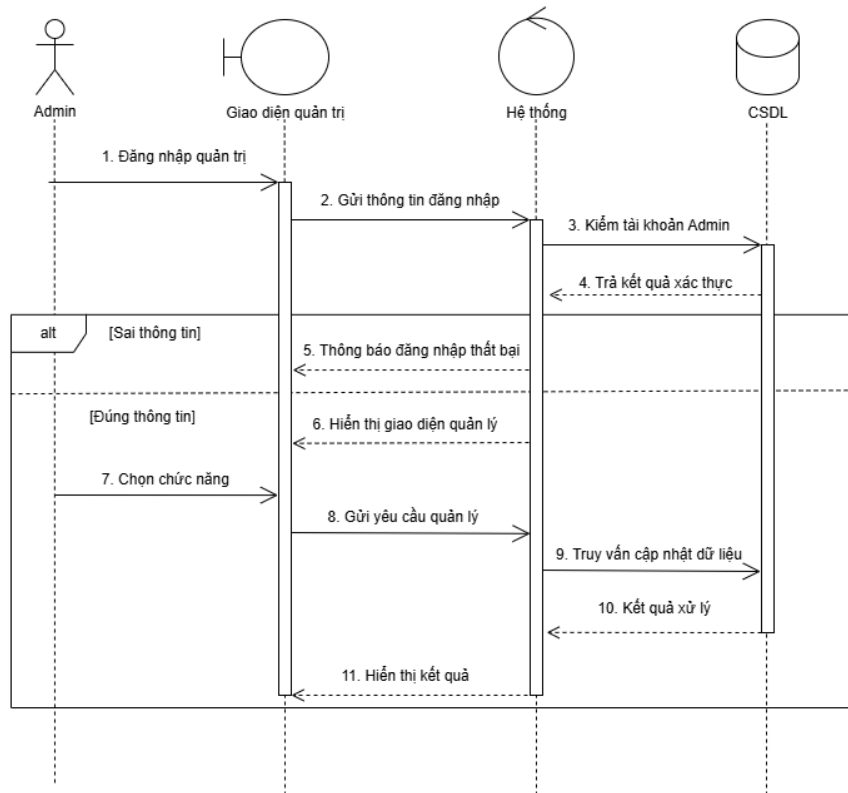
Hình 3.14 Biểu đồ tuần tự thanh toán.

### 3.5.4. Biểu đồ tuần tự cập nhật thông tin.



Hình 3.15 Biểu đồ tuần tự cập nhật thông tin.

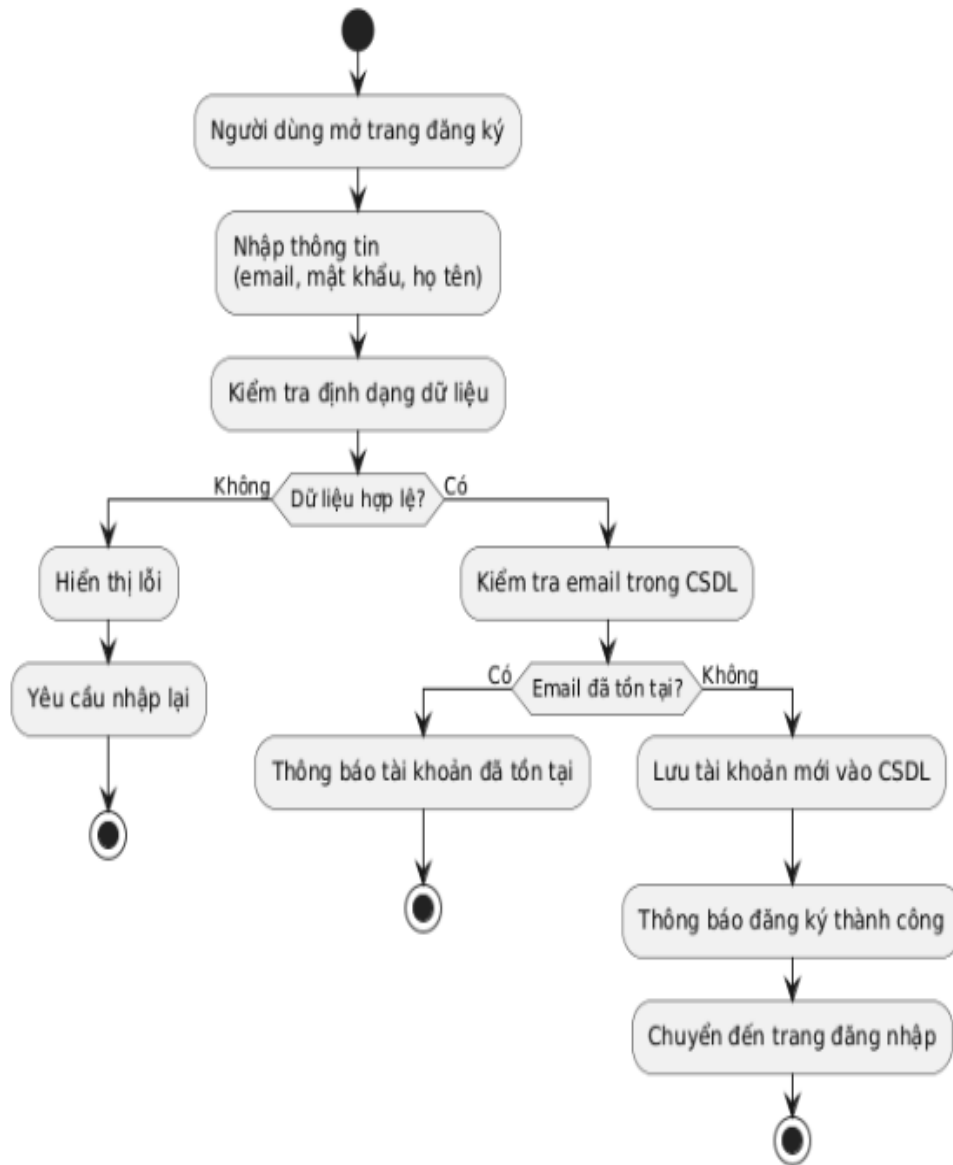
### 3.5.5. Biểu đồ tuần tự đăng nhập và sử dụng các chức năng quản trị.



Hình 3.16 Biểu đồ tuần tự đăng nhập và sử dụng các chức năng quản trị.

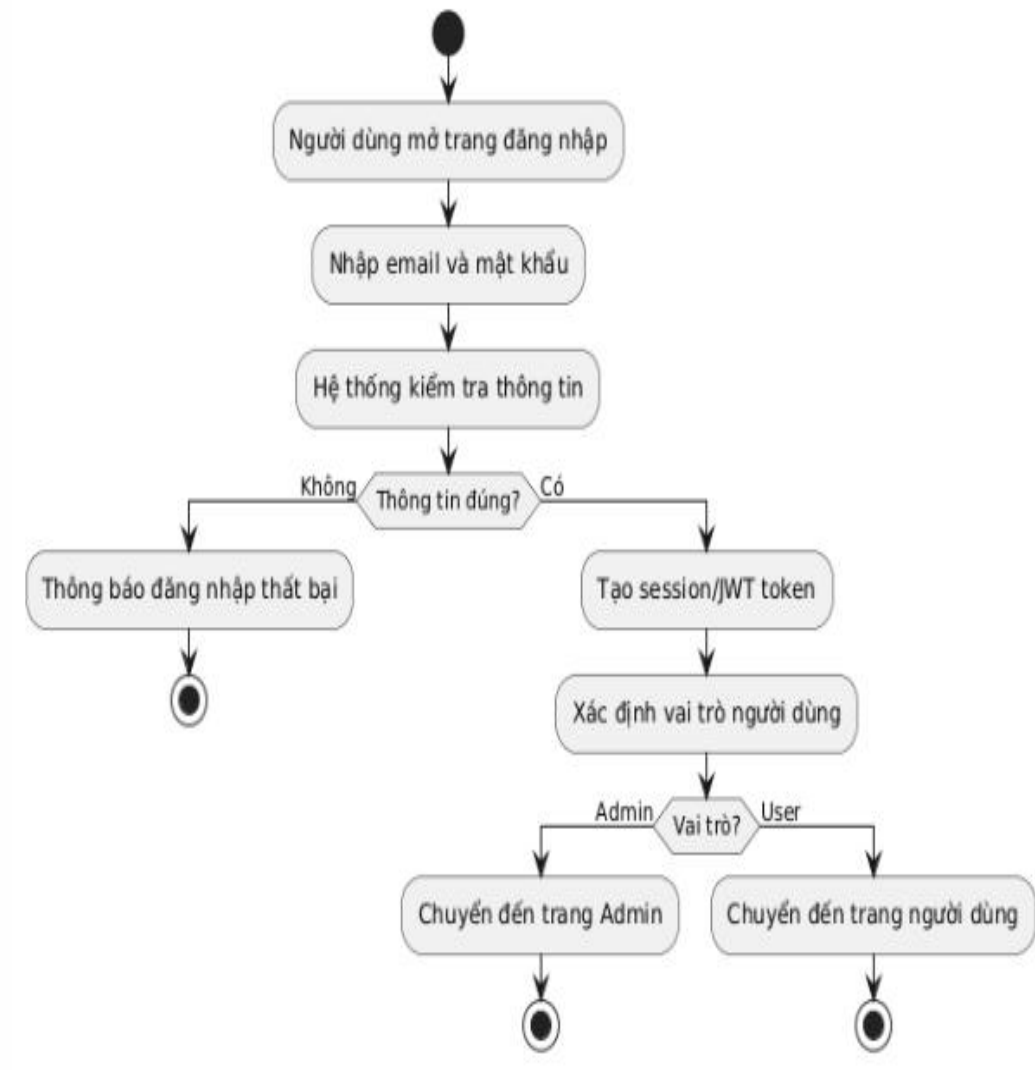
### 3.6 Biểu đồ hoạt động.

### 3.6.1. Biểu đồ hoạt động đăng ký.

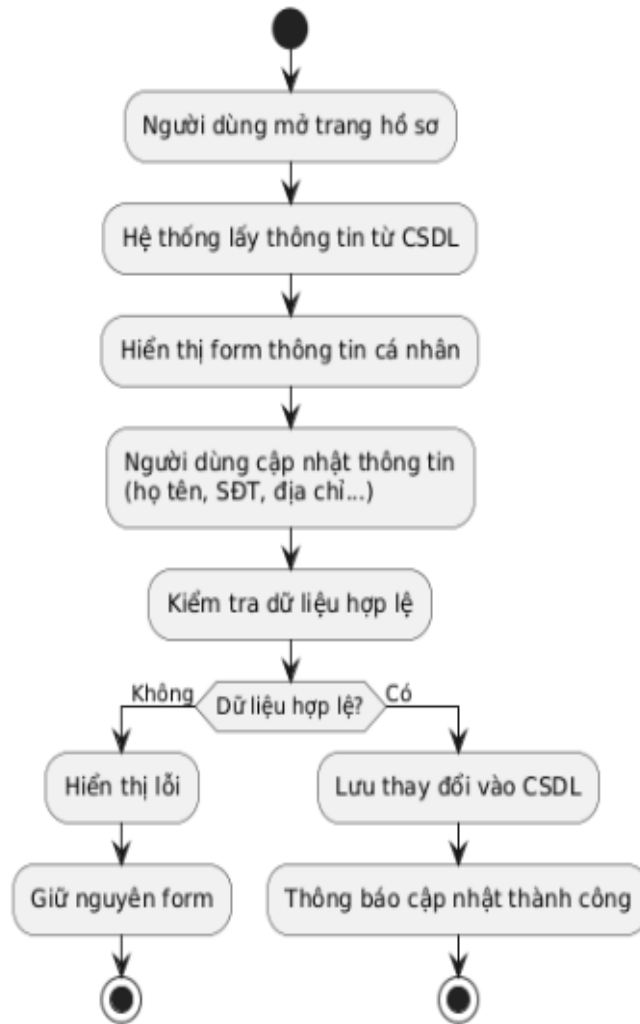


Hình 3.17 Biểu đồ hoạt động đăng ký.

### 3.6.2. Biểu đồ hoạt động đăng nhập

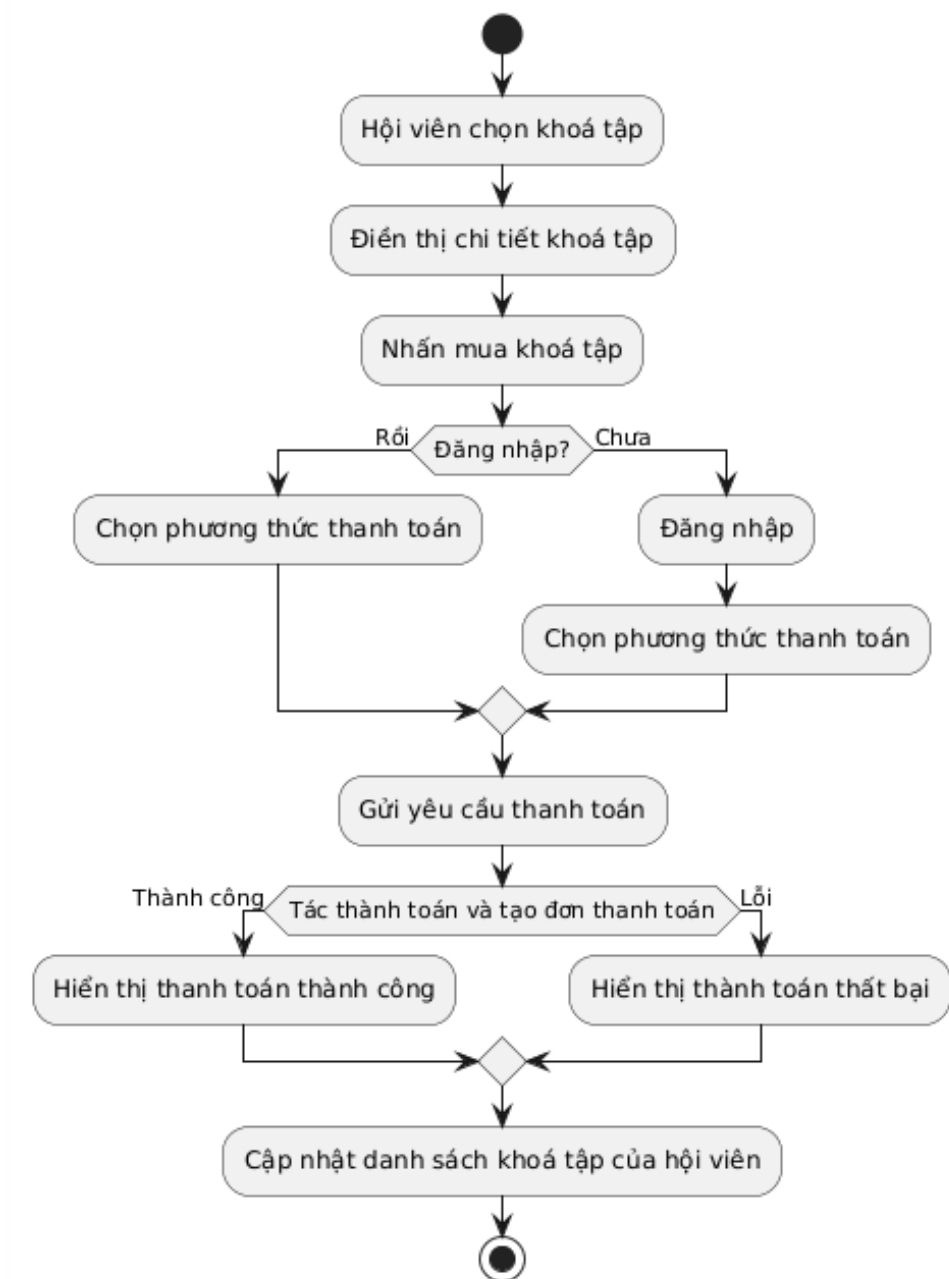


Hình 3.18 Biểu đồ hoạt động đăng nhập.

**3.6.3. Biểu đồ hoạt động chỉnh sửa thông tin cá nhân.**

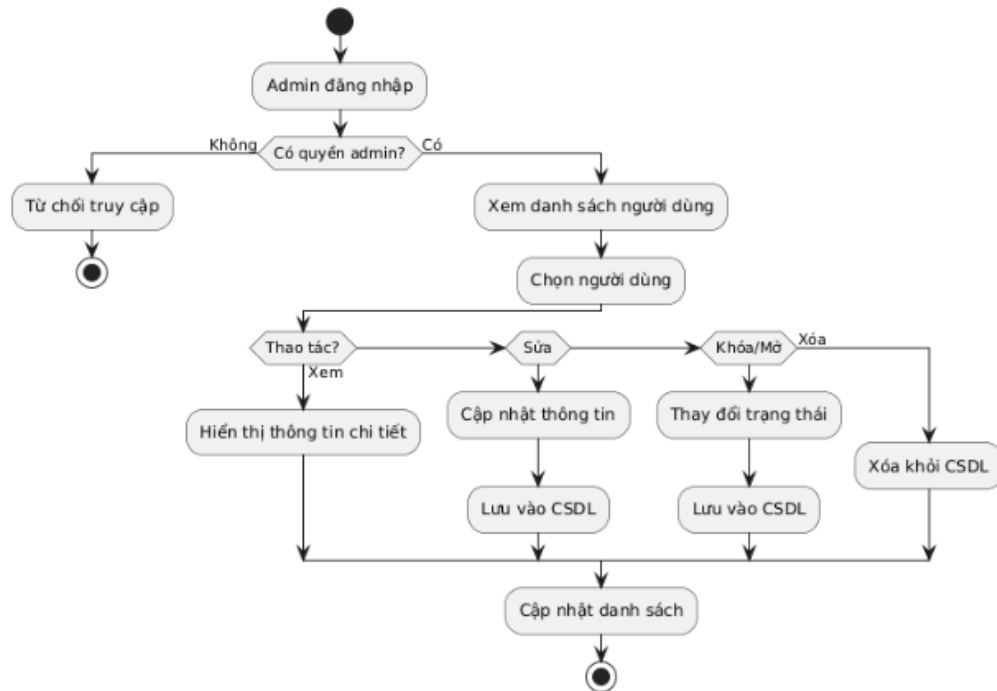
Hình 3.19 Biểu đồ hoạt động chỉnh sửa thông tin cá nhân.

#### 3.6.4. Biểu đồ hoạt động thanh toán.



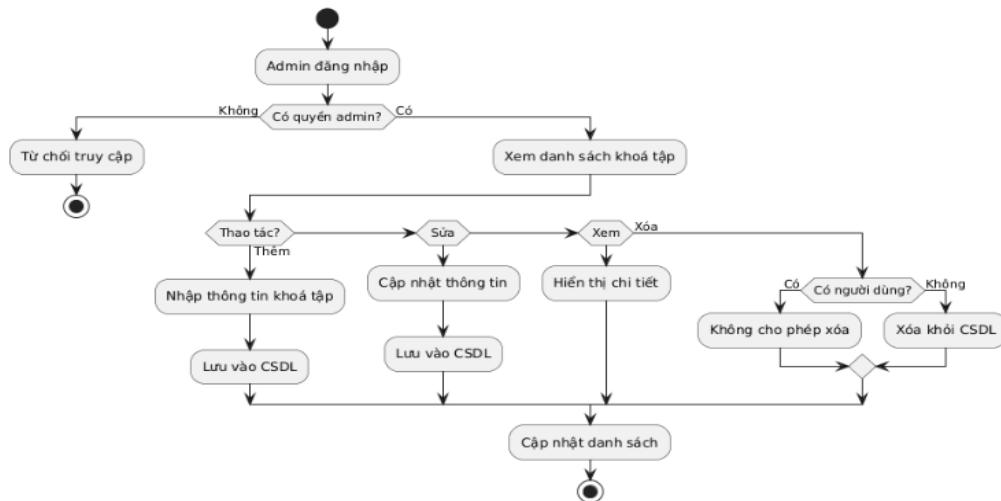
Hình 3.20 Biểu đồ hoạt động thanh toán.

### 3.6.5. Biểu đồ hoạt động quản lý người dùng.



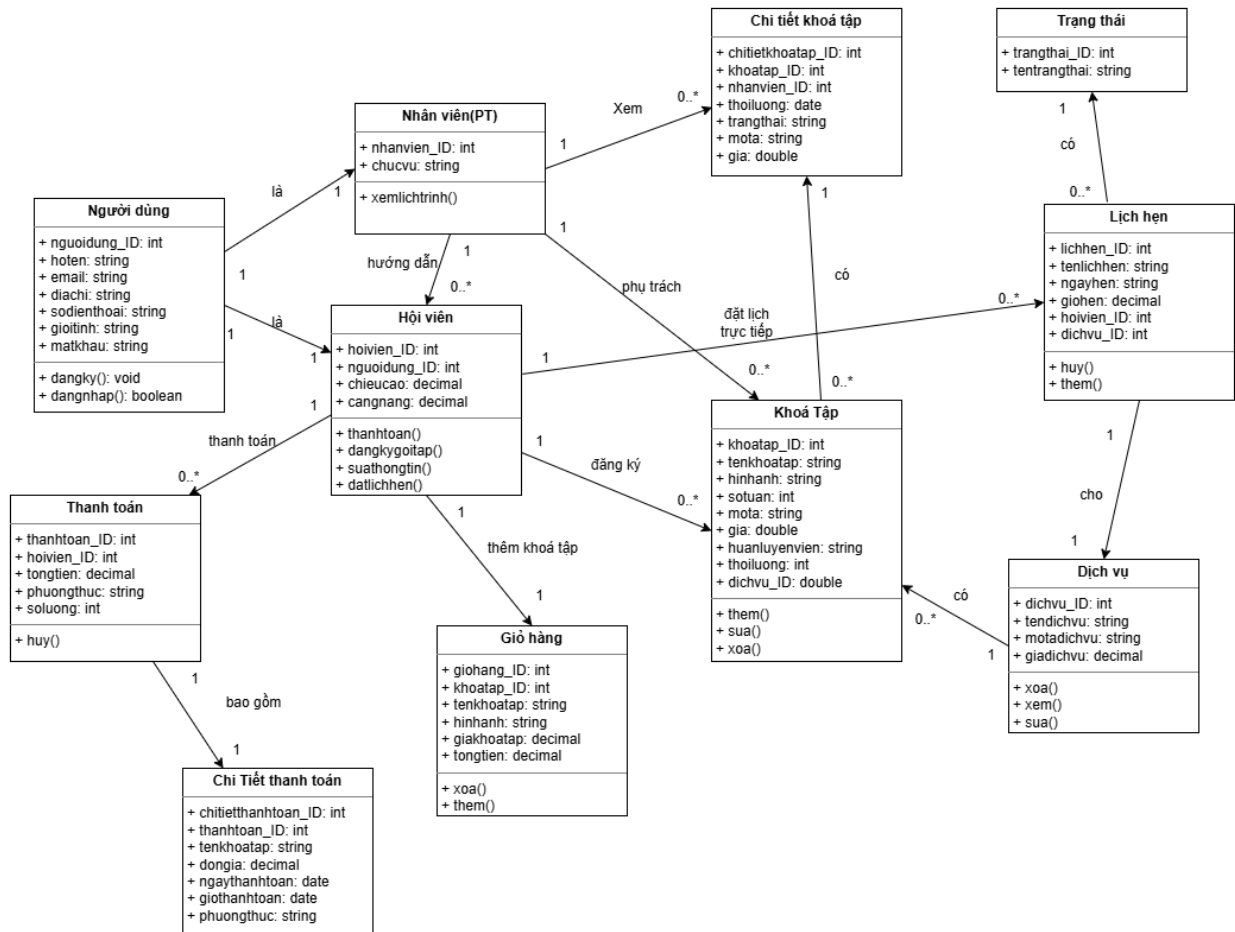
Hình 3.21 Biểu đồ hoạt động quản lý người dùng.

### 3.6.6. Biểu đồ hoạt động quản lý khoá tập.



Hình 3.22 Biểu đồ hoạt động quản lý khoá tập.

### 3.7. Biểu đồ lớp.



Hình 3.23 Biểu đồ lớp.



- HọTên
- Email
- SDT
- ĐịaChỉ
- GiớiTính
- MậtKhẩu
- VaiTrò (0: Hội viên, 1: Nhân viên, 2: Admin, ...)

#### Hội viên

- MãHộiViên (khóa chính)
- MãNgườiDùng (khóa ngoại sang Người dùng)
- CânNặng
- ChiềuCao

#### Nhân viên (PT)

- MãNhânViên (khóa chính)
- MãNgườiDùng (khóa ngoại sang Người dùng – nếu dùng chung tài khoản)

#### Khóa tập

- MãKhóaTập (khóa chính)
- TênKhóaTập
- ThờiLượng
- Giá (giá niêm yết hiện tại)
- HìnhẢnh
- MôTả

#### Chi tiết khóa tập

- MãChiTietKhoaTap (khóa chính)
- MãKhóaTập (khóa ngoại)
- ThờiLượng
- MôTả
- Giá (nếu buổi đó có phụ phí riêng)

#### Dịch vụ

- MãDịchVụ (khóa chính)
- TênDịchVụ
- GiáDịchVụ

- MôTảDịchVụ

Lịch hẹn

- MãLịchHẹn (khóa chính)
- NgàyHẹn
- GiờHẹn
- MãHộiViên (khóa ngoại)
- MãNhânViên (khóa ngoại – PT phục vụ)
- MãDịchVụ (khóa ngoại)
- MãTrạngThái (khóa ngoại)

Trạng thái

- MãTrạngThái (khóa chính)
- TênTrạngThái (Chờ xác nhận, Đã xác nhận, Đã hủy,...)

Giỏ hàng

- MãGiỏHàng (khóa chính)
- MãHộiViên (khóa ngoại, có thể NULL nếu khách vắng lai)
- TổngTiền

Thanh toán

- MãThanhToán (khóa chính)
- MãHộiViên (khóa ngoại, NULL nếu khách vắng lai)
- PhươngThức (Stripe)
- TổngTiền

Chi tiết thanh toán

- MãChiTietThanhToán (khóa chính)
- MãThanhToán (khóa ngoại)
- MãKhóaTập hoặc MãDịchVụ (khóa ngoại tới loại mục được mua)
- SốLượng
- Giá

– Các mối quan hệ:

Người dùng – Hội viên:

- Quan hệ: là / sở hữu tài khoản hội viên
- Bội số: 1 NgườiDùng – 1 HộiViên (hoặc  $1 \leftrightarrow 0..1$  tùy thiết kế).

Người dùng – Nhân viên:

- Quan hệ: là / có tài khoản nhân viên
- Bội số: 1 NgườiDùng – 1 NhânViên.

Nhân viên – Chi tiết khóa tập:

- Quan hệ: Hướng dẫn / Xem / Phụ trách
- Bội số: 1 NhânViên – n ChiTietKhoaTap.

Nhân viên – Khóa tập:

- Quan hệ: Quản lý / Giám giá... (trong hình có “Giảm giá”, “Quản lý”)
- Bội số: 1 NhânViên – n KhoaTap.

Hội viên – Khóa tập:

- Quan hệ: Mua / Đăng ký
- Bội số: 1 HộiViên – n KhoaTap (thông qua giỏ hàng hay thanh toán).

Khóa tập – Chi tiết khóa tập:

- Quan hệ: Có (một khóa tập có nhiều chi tiết)
- Bội số: 1 KhoaTap – n ChiTietKhoaTap.

Hội viên – Giỏ hàng:

- Quan hệ: Có / Sở hữu
- Bội số: 1 HộiViên – n GioHang.

Giỏ hàng – Khóa tập:

- Quan hệ: Thêm khóa tập vào giỏ
- Bội số: 1 GioHang – n KhoaTap (thường là bảng trung gian).

Hội viên – Thanh toán:

- Quan hệ: Thực hiện / Thanh toán
- Bội số: 1 HộiViên – n ThanhToan.

Thanh toán – Chi tiết thanh toán:

- Quan hệ: Có
- Bội số: 1 ThanhToan – n ChiTietThanhToan.

Chi tiết thanh toán – Khóa tập:

- Quan hệ: Thanh toán cho khóa tập
- Bội số: n ChiTietThanhToan – 1 KhoaTap.

Hội viên – Lịch hẹn:

- Quan hệ: Đặt

- Bội số: 1 HộiViên – n LichHen.

Nhân viên – Lịch hẹn:

- Quan hệ: Được đặt / Hướng dẫn trong lịch hẹn
- Bội số: 1 NhânVien – n LichHen.

Dịch vụ – Lịch hẹn:

- Quan hệ: Có (lịch hẹn cho một dịch vụ cụ thể)
- Bội số: 1 DichVu – n LichHen.

Trạng thái – Lịch hẹn:

- Quan hệ: Có / mang trạng thái
- Bội số: 1 TrangThai – n LichHen.

### 3.9. Thiết kế cơ sở dữ liệu

Thiết kế gồm các bảng quan hệ, khóa chính (Primary Key), khóa ngoại (Foreign Key), ràng buộc toàn vẹn dữ liệu.

#### Bảng người dùng:

Bảng 3.11 Mô tả bảng người dùng.

| Tên cột      | Kiểu dữ liệu | Ràng buộc / Mô tả                |
|--------------|--------------|----------------------------------|
| nguoidung_id | INT          | PRIMARY KEY, AUTO INCREMENT      |
| hoten        | VARCHAR(100) | NOT NULL, họ tên người dùng      |
| email        | VARCHAR(100) | NOT NULL, UNIQUE, địa chỉ email  |
| sodienthoai  | VARCHAR(20)  | Số điện thoại                    |
| gioitinh     | VARCHAR(10)  | Giới tính người dùng             |
| matkhau      | VARCHAR(255) | NOT NULL, lưu mật khẩu đã mã hóa |

**Bảng nhân viên:**

Bảng 3.12 Mô tả bảng nhân viên.

| Tên cột      | Kiểu dữ liệu | Ràng buộc / Mô tả                                 |
|--------------|--------------|---------------------------------------------------|
| nhanvien_id  | INT          | PRIMARY KEY, AUTO INCREMENT                       |
| chucvu       | VARCHAR(100) | NOT NULL, chức vụ của nhân viên                   |
| nguoidung_id | INT          | Có thể NULL, liên kết với NguoiDung(nguoidung_id) |

**Bảng hội viên:**

Bảng 3.13 Mô tả bảng hội viên.

| Tên cột      | Kiểu dữ liệu | Ràng buộc / Mô tả                              |
|--------------|--------------|------------------------------------------------|
| hoivien_id   | INT          | PRIMARY KEY, AUTO INCREMENT                    |
| nguoidung_id | INT          | NOT NULL, liên kết với NguoiDung(nguoidung_id) |
| chieucaoc    | DECIMAL(5,2) | Chiều cao của hội viên (m, cm)                 |
| cannang      | DECIMAL(5,2) | Cân nặng của hội viên (kg)                     |

**Bảng khoá tập:**

Bảng 3.14 Mô tả bảng khoá tập.

| Tên cột       | Kiểu dữ liệu  | Ràng buộc / Mô tả                 |
|---------------|---------------|-----------------------------------|
| khoatap_id    | INT           | PRIMARY KEY, AUTO INCREMENT       |
| tenkhoatap    | VARCHAR(100)  | NOT NULL, tên khóa tập            |
| hinhanh       | VARCHAR(255)  | Hình ảnh minh họa khóa tập        |
| mota          | TEXT          | Mô tả chi tiết về khóa tập        |
| gia           | DECIMAL(12,2) | Giá khóa tập                      |
| huanluyenvien | VARCHAR(100)  | Tên huấn luyện viên               |
| thoiluong     | INT           | Thời lượng (số buổi hoặc số ngày) |

|           |     |                                   |
|-----------|-----|-----------------------------------|
| dichvu_id | INT | Liên kết với<br>DichVu(dichvu id) |
|-----------|-----|-----------------------------------|

**Bảng chi tiết khoá tập:**

Bảng 3.14 Mô tả bảng chi tiết khoá tập.

| Tên cột           | Kiểu dữ liệu  | Ràng buộc / Mô tả                              |
|-------------------|---------------|------------------------------------------------|
| chitietkhoatap_id | INT           | PRIMARY KEY,<br>AUTO INCREMENT                 |
| khoatap_id        | INT           | NOT NULL, liên kết với<br>KhoaTap(khoatap id)  |
| thoiluong         | DATE          | Ngày tập (hoặc thời gian<br>dự kiến)           |
| mota              | TEXT          | Mô tả chi tiết cho buổi tập                    |
| gia               | DECIMAL(12,2) | Giá cho buổi tập (có thể<br>khác với giá khóa) |

**Bảng giỏ hàng:**

Bảng 3.15 Mô tả bảng giỏ hàng.

| Tên cột    | Kiểu dữ liệu  | Ràng buộc / Mô tả                              |
|------------|---------------|------------------------------------------------|
| giohang_id | INT           | PRIMARY KEY,<br>AUTO INCREMENT                 |
| hoivien_id | INT           | NOT NULL, liên kết với<br>HoiVien(hoivien id)  |
| tenkhoatap | VARCHAR(100)  | NOT NULL, tên khóa tập                         |
| hinhanh    | VARCHAR(255)  | Hình ảnh minh họa khóa<br>tập                  |
| gia        | DECIMAL(12,2) | Giá cho buổi tập (có thể<br>khác với giá khóa) |
| tongtien   | DECIMAL(12,2) | NOT NULL, tổng số tiền<br>thanh toán           |

**Bảng thanh toán:**

Bảng 3.16 Mô tả bảng thanh toán.

| Tên cột | Kiểu dữ liệu | Ràng buộc / Mô tả |
|---------|--------------|-------------------|
|---------|--------------|-------------------|

|              |               |                                                            |
|--------------|---------------|------------------------------------------------------------|
| thanhtoan_id | INT           | PRIMARY KEY,<br>AUTO INCREMENT                             |
| hoivien_id   | INT           | NOT NULL, liên kết với<br>HoiVien(hoivien_id)              |
| tongtien     | DECIMAL(12,2) | NOT NULL, tổng số tiền<br>thanh toán                       |
| phuongthuc   | VARCHAR(50)   | NOT NULL, phương thức<br>thanh toán (Tiền mặt,<br>Thẻ,...) |
| soluong      | INT           | Số lượng sản phẩm được<br>thanh toán                       |

**Bảng chi tiết thanh toán:**

Bảng 3.17 Mô tả bảng chi tiết thanh toán.

| Tên cột             | Kiểu dữ liệu  | Ràng buộc / Mô tả                                 |
|---------------------|---------------|---------------------------------------------------|
| chitietthanhtoan_id | INT           | PRIMARY KEY,<br>AUTO INCREMENT                    |
| thanhtoan_id        | INT           | NOT NULL, liên kết với<br>ThanhToan(thanhtoan_id) |
| khoatap_id          | INT           | Liên kết với<br>KhoaTap(khoatap_id)               |
| soluong             | INT           | Số lượng khóa tập/dịch vụ<br>thanh toán           |
| dongia              | DECIMAL(12,2) | Giá từng đơn vị                                   |
| ngaythanhtoan       | DATETIME      | Ngày giờ thanh toán chi<br>tiết                   |
| phuongthuc          | VARCHAR(50)   | Phương thức thanh toán chi<br>tiết                |

**Bảng trạng thái:**

Bảng 3.18 Mô tả bảng trạng thái.

| Tên cột      | Kiểu dữ liệu | Ràng buộc / Mô tả                                                                      |
|--------------|--------------|----------------------------------------------------------------------------------------|
| trangthai_id | INT          | PRIMARY KEY,<br>AUTO INCREMENT                                                         |
| tentrangthai | VARCHAR(100) | NOT NULL, tên trạng thái<br>(VD: Chưa xác nhận, Đã<br>xác nhận, Hoàn thành,<br>Hủy...) |

**Bảng dịch vụ:**

Bảng 3.19 Mô tả bảng dịch vụ.

| Tên cột    | Kiểu dữ liệu  | Ràng buộc / Mô tả           |
|------------|---------------|-----------------------------|
| dichvu_id  | INT           | PRIMARY KEY, AUTO INCREMENT |
| tendichvu  | VARCHAR(100)  | NOT NULL, tên dịch vụ       |
| motadichvu | TEXT          | Mô tả chi tiết về dịch vụ   |
| giadichvu  | DECIMAL(12,2) | Giá dịch vụ                 |

**Bảng lịch hẹn:**

Bảng 3.20 Mô tả bảng lịch hẹn.

| Tên cột      | Kiểu dữ liệu | Ràng buộc / Mô tả                          |
|--------------|--------------|--------------------------------------------|
| lichhen_id   | INT          | PRIMARY KEY, AUTO INCREMENT                |
| tendichvu    | VARCHAR(100) | Tên dịch vụ (có thể lấy từ DichVu)         |
| ngayhen      | DATE         | Ngày hẹn                                   |
| giohen       | TIME         | Giờ hẹn                                    |
| hoivien_id   | INT          | NOT NULL, liên kết với HoiVien(hoivien_id) |
| dichvu_id    | INT          | NOT NULL, liên kết với DichVu(dichvu_id)   |
| trangthai_id | INT          | Liên kết với TrangThai(trangthai_id)       |

## CHƯƠNG 4: TRIỂN KHAI HỆ THỐNG

### 4.1. Kiến trúc hệ thống

#### 1. Phần Frontend (ReactJS / Next.js)

– Công nghệ

- Sử dụng ReactJS (framework Next.js – thể hiện qua cấu trúc src/app/...) để xây dựng giao diện người dùng.
- Dùng TypeScript để tăng tính an toàn kiểu dữ liệu.
- Dùng Tailwind CSS (và có thể Vite hoặc công cụ build của Next.js) để tạo giao diện hiện đại, responsive.

– Vai trò

Hiển thị các màn hình cho:

- Đăng ký, đăng nhập người dùng.
- Trang hội viên: xem/thanh toán khoá tập, lịch hẹn PT.
- Trang nhân viên/PT: xem danh sách dạy các khoá tập.
- Trang Admin: quản lý khoá tập, dịch vụ, người dùng, lịch hẹn, báo cáo doanh thu.
- Quản lý state trên trình duyệt (giỏ hàng, khoá tập, thông tin người dùng, token đăng nhập, ...).
- Gửi/nhận dữ liệu với Backend thông qua RESTful API (hoặc GraphQL nếu mở rộng).

#### 2. Backend (ASP.NET Core 8.0)

– Công nghệ:

- ASP.NET Core 8.0
- Entity Framework Core để làm ORM truy cập DB

– Vai trò:

- Cài đặt nghiệp vụ:
  - Quản lý người dùng, phân quyền (Người dùng, PT, Admin)
  - Quản lý khoá tập, chi tiết khoá tập, dịch vụ
  - Quản lý lịch hẹn

- Quản lý giỏ hàng, đơn thanh toán, chi tiết thanh toán
- Thống kê / báo cáo cho admin
- Cung cấp các RESTful API cho frontend:
  - /api/auth/login, /api/auth/register
  - /api/members, /api/trainers, /api/classes, /api/services
  - /api/bookings, /api/payments, /api/admin/reports, ...
- Xác thực, phân quyền bằng JWT Bearer hoặc cookie auth.

### 3. Cơ sở dữ liệu (Database)

- Hệ QTCSDL: PostgreSQL.
- Lưu trữ:
  - Bảng: NguoiDung, HoiVien, NhanVien, KhoaTap, ChiTietKhoaTap, DichVu, LichHen, TrangThai, GioHang, ThanhToan, ChiTietThanhToan, ...
- Truy cập:
  - Backend dùng Entity Framework Core (DbContext, DbSet<>) để truy vấn và cập nhật.

### 4. Tương tác giữa các phần

- Frontend → Backend
  - Khi người dùng thao tác (ví dụ: đăng ký khoá tập, cập nhật thông tin cá nhân), Frontend gửi HTTP Request (thường là JSON) đến các API của ASP.NET Core.
- Backend → Database
  - Backend kiểm tra quyền truy cập, validate dữ liệu, sau đó truy vấn hoặc cập nhật CSDL (tạo đơn hàng, tạo lịch hẹn, cập nhật trạng thái thanh toán, ...) thông qua Entity Framework Core.
- Backend → Frontend
  - Sau khi xử lý, Backend trả về HTTP Response (JSON: kết quả / lỗi / dữ liệu mới).
  - Frontend nhận dữ liệu, cập nhật UI (hiển thị danh sách khoá tập đã mua, lịch hẹn của hội viên, v.v.).

### 5. Ưu điểm của kiến trúc

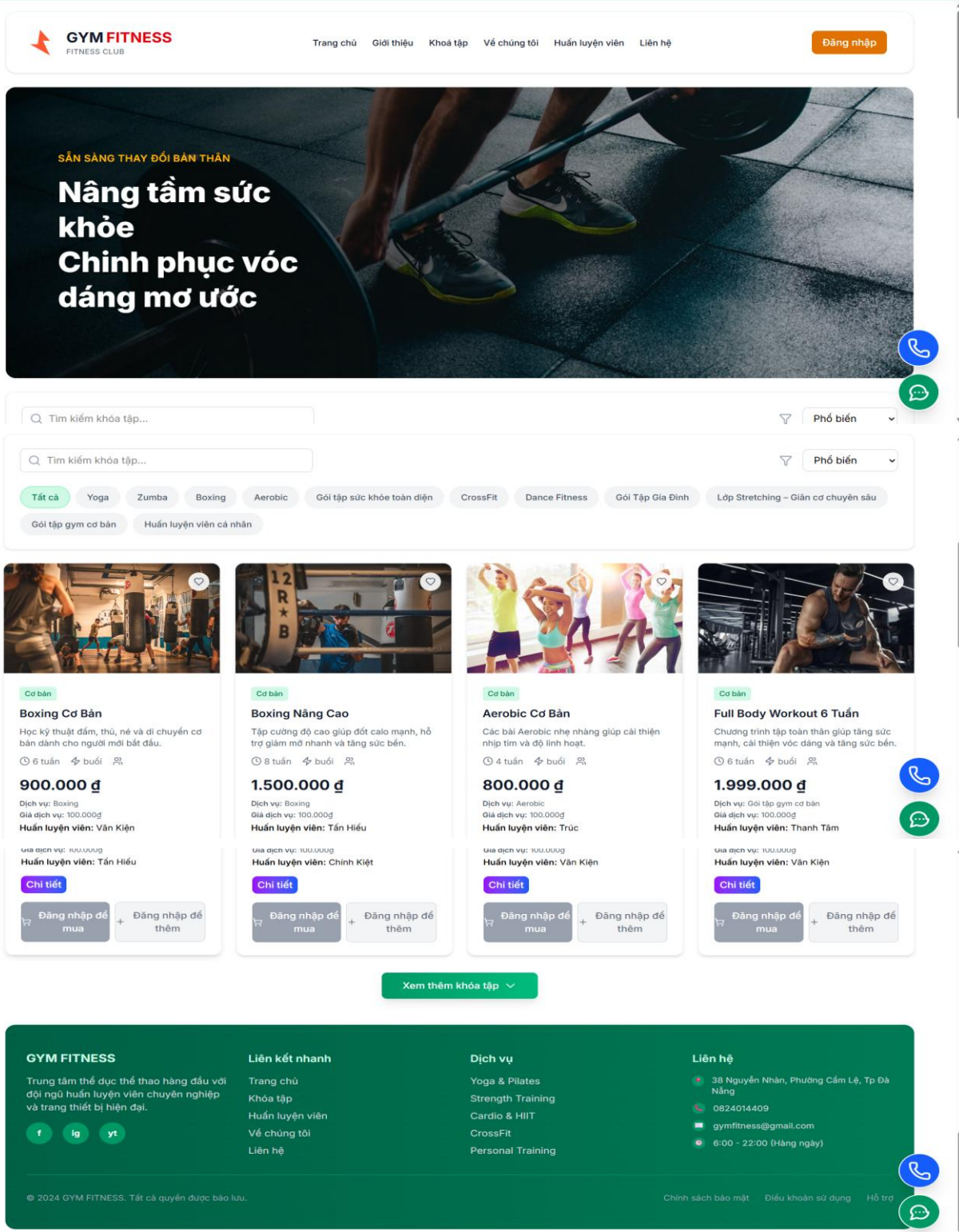
- Tách biệt rõ ràng giữa UI (Frontend) và nghiệp vụ + dữ liệu (Backend + DB).

- Dễ mở rộng:
  - Có thể thay đổi giao diện (React/Next) mà không đụng đến Asp.net Core.
  - Có thể thêm mobile app dùng chung API của Backend.
- Dễ bảo trì & phân quyền phát triển: team frontend, backend, DBA làm việc song song.

## **4.2. Giao diện người dùng(hội viên).**

### **4.2.1. Giao diện trang chủ.**

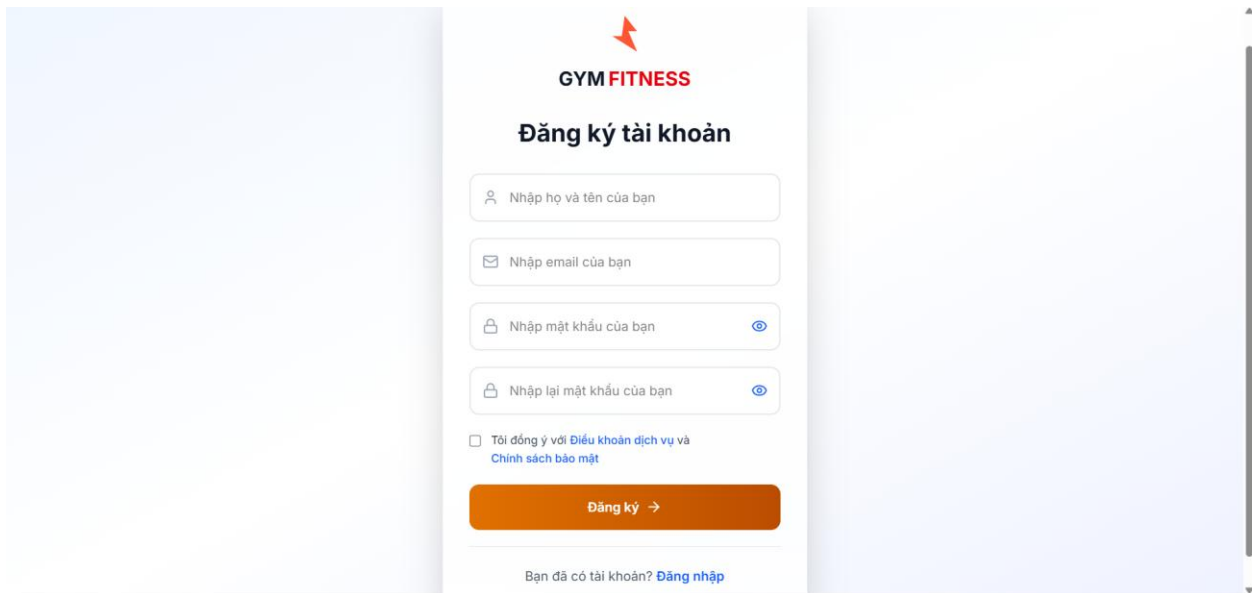
Trang chủ cung cấp góc nhìn tổng quan về hệ thống và các dịch vụ của phòng Gym. Các mục quan trọng được trình bày nổi bật như khóa tập, dịch vụ và các nút dẫn đến những chức năng chính. Giao diện được bố trí hài hòa, tạo cảm giác chuyên nghiệp và thân thiện ngay từ lần truy cập đầu tiên.



Hình 4.1 Giao diện trang chủ.

#### 4.2.2. Giao diện đăng ký.

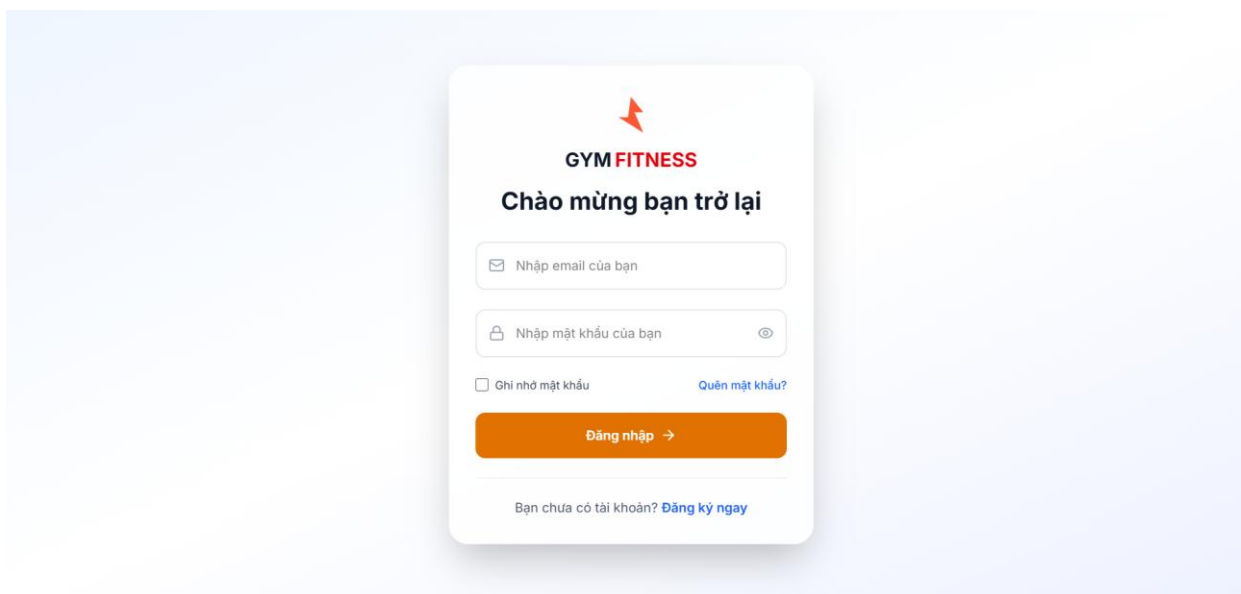
Giao diện đăng ký hỗ trợ người dùng tạo tài khoản mới một cách nhanh chóng. Các trường thông tin được bố trí khoa học, có kiểm tra dữ liệu nhập vào để tránh sai sót. Sau khi hoàn tất đăng ký, người dùng được chuyển tới trang đăng nhập để tiếp tục sử dụng hệ thống.



Hình 4.2 Giao diện đăng ký.

#### 4.2.3. Giao diện đăng nhập.

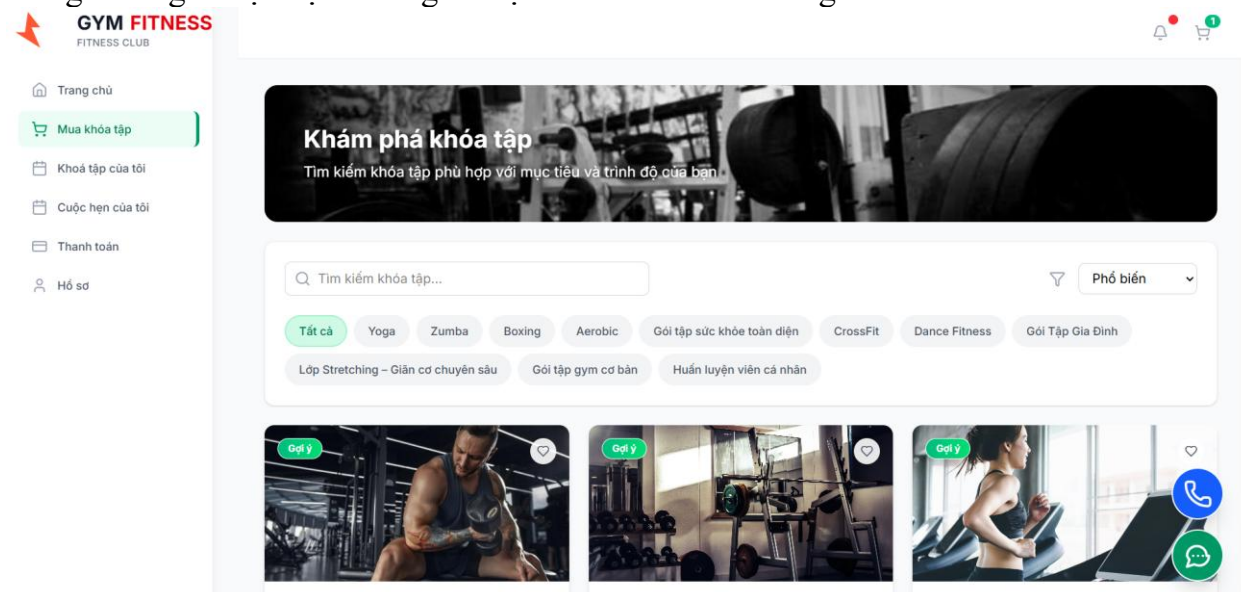
Tại giao diện đăng nhập, người dùng chỉ cần cung cấp email và mật khẩu để vào hệ thống. Trang có thông báo lỗi rõ ràng nếu nhập sai thông tin, giúp người dùng dễ dàng nhận biết và điều chỉnh. Sau khi xác thực, hệ thống tự động chuyển hướng theo đúng vai trò của người dùng.



Hình 4.3 Giao diện đăng nhập.

#### 4.2.4. Giao diện mua khoá tập.

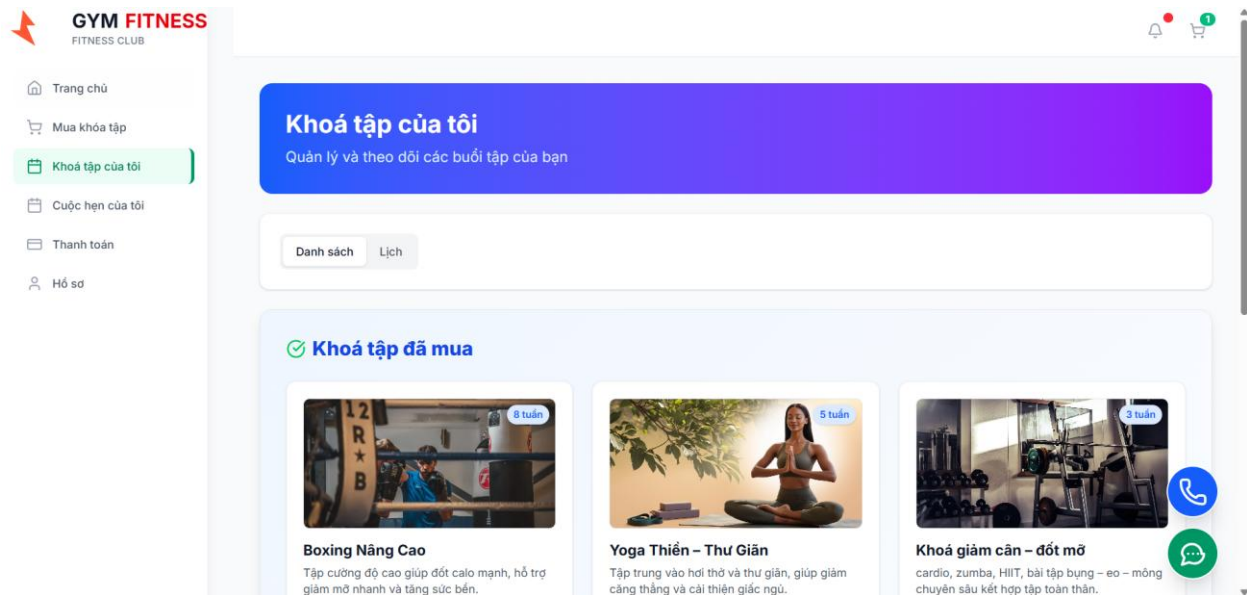
Tại giao diện này, người dùng có thể xem thông tin đầy đủ của một khóa tập như mô tả chi tiết, huấn luyện viên phụ trách, thời lượng và giá. Nút “Mua ngay” hoặc “Thêm vào giỏ hàng” được đặt rõ ràng để hội viên thao tác dễ dàng.



Hình 4.4 Giao diện mua khoá tập.

#### 4.2.5. Giao diện khoá tập đã mua.

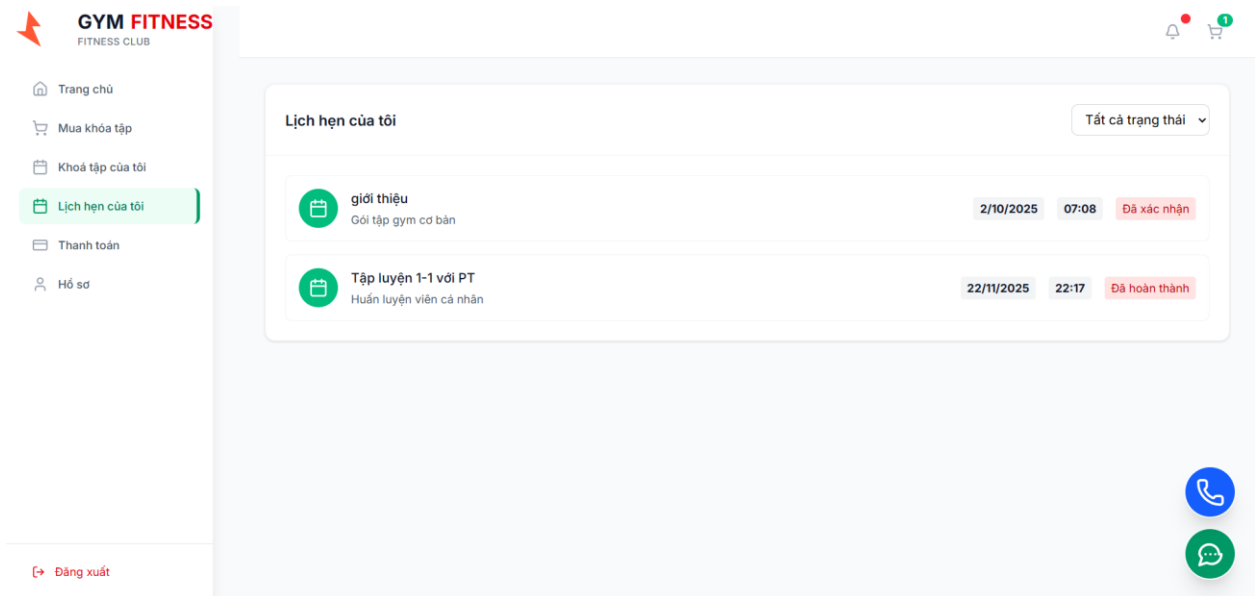
Trang này liệt kê các khóa tập mà hội viên đã đăng ký hoặc mua trước đó. Mỗi mục bao gồm: tên khóa, ngày mua, thời gian hiệu lực, trạng thái khóa tập và nút xem chi tiết. Giao diện giúp hội viên theo dõi tiến trình học tập và chủ động sắp xếp lịch tập.



Hình 4.5 Giao diện khoá tập đã mua.

#### 4.2.6. Giao diện lịch hẹn.

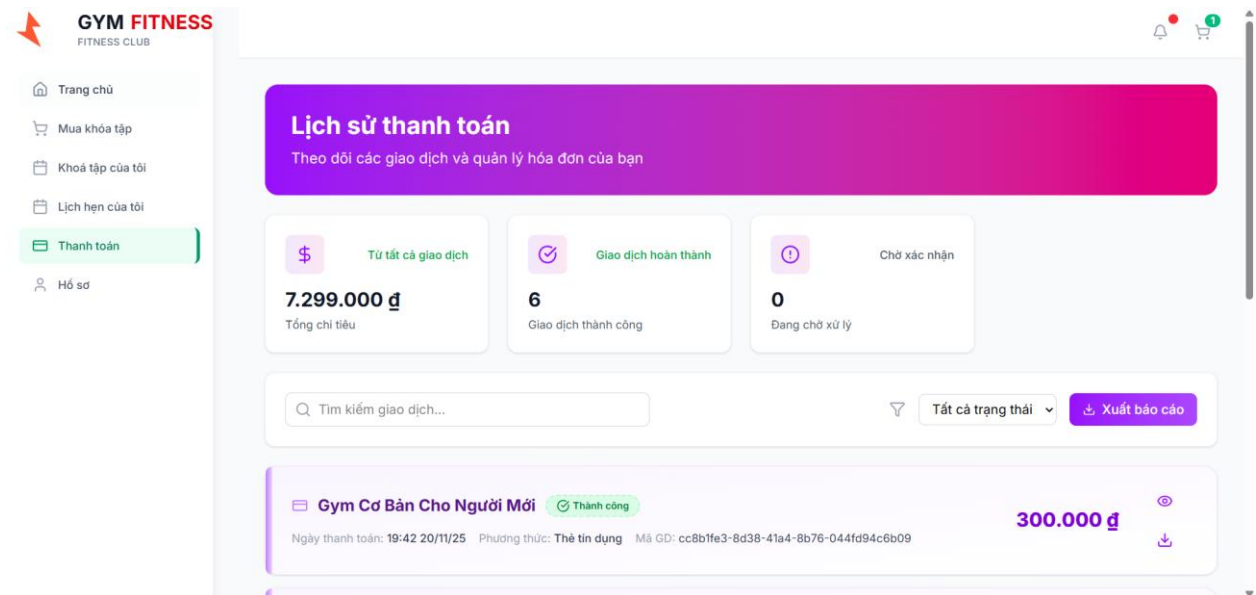
Giao diện lịch hẹn dành cho hội viên hiển thị toàn bộ các buổi tập mà người dùng đã đặt. Thông tin được trình bày theo dạng danh sách bao gồm: ngày – giờ tập, huấn luyện viên phụ trách, loại dịch vụ hoặc khóa tập liên quan, giúp hội viên theo dõi lịch tập một cách thuận tiện và chủ động hơn trong việc sắp xếp thời gian.



Hình 4.6 Giao diện lịch hẹn của tôi.

#### 4.2.7. Giao diện lịch sử thanh toán.

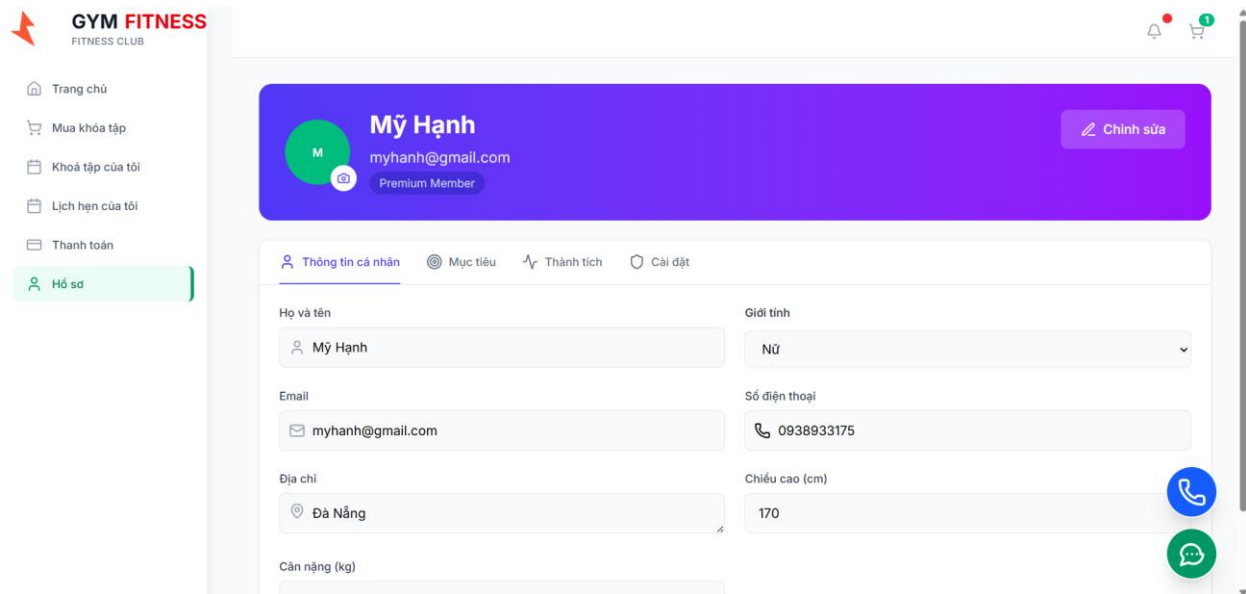
Giao diện trình bày danh sách các giao dịch mà hội viên đã thực hiện. Mỗi giao dịch hiển thị ngày giờ thanh toán, mã đơn hàng, số tiền.



Hình 4.7 Giao diện lịch sử thanh toán.

#### 4.2.8. Giao diện hồ sơ.

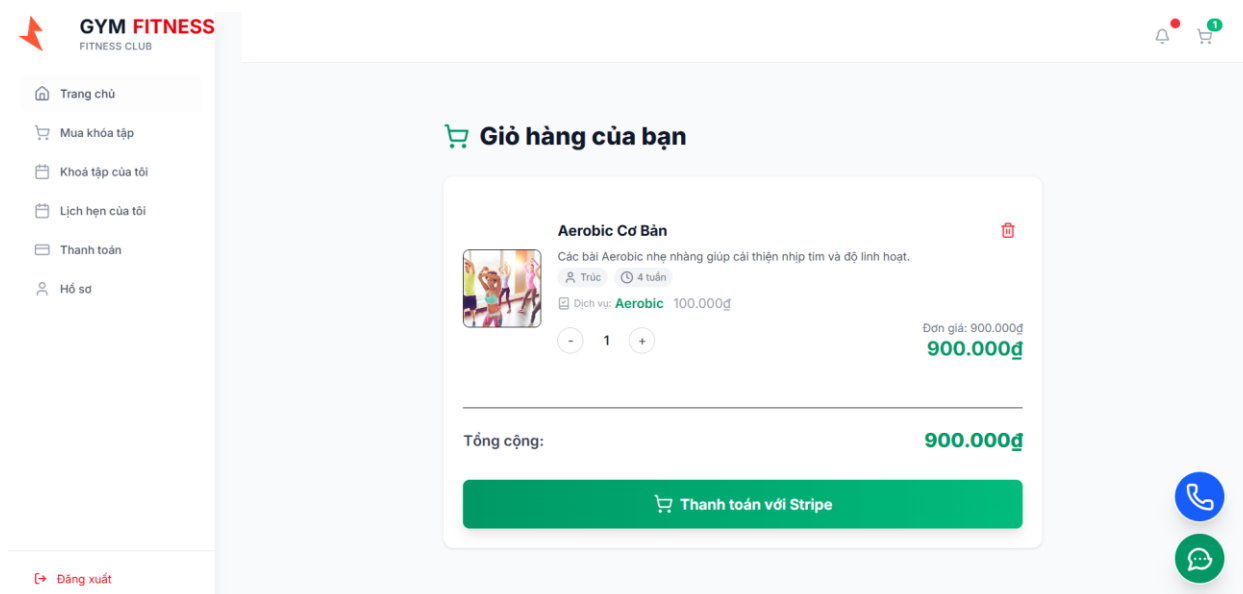
Hội viên có thể xem và chỉnh sửa thông tin cá nhân như họ tên, email, số điện thoại... Giao diện đơn giản, hỗ trợ cập nhật nhanh và đồng bộ dữ liệu với hệ thống.



Hình 4.8 Giao diện hồ sơ.

#### 4.2.9. Giao diện giỏ hàng.

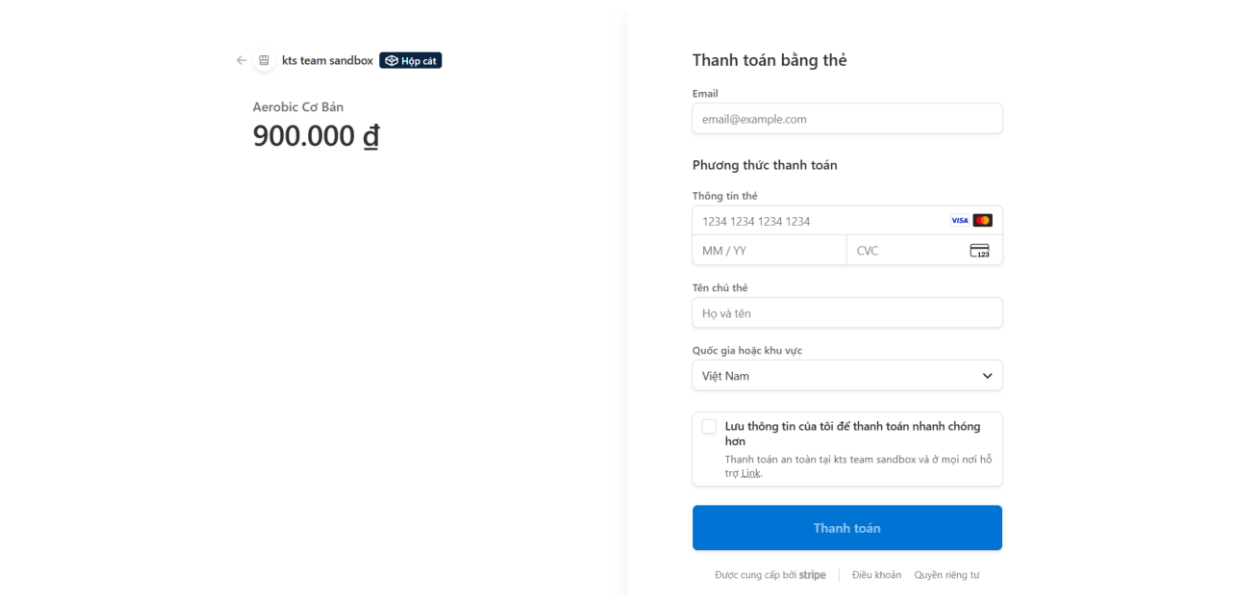
Giao diện giỏ hàng cho phép hội viên kiểm tra khóa tập đã chọn mua. Người dùng có thể thay đổi số lượng, loại bỏ sản phẩm. Tổng chi phí được hiển thị rõ ràng để người dùng nắm được số tiền cần thanh toán.



Hình 4.9 Giao diện giỏ hàng.

#### 4.2.10. Giao diện thanh toán.

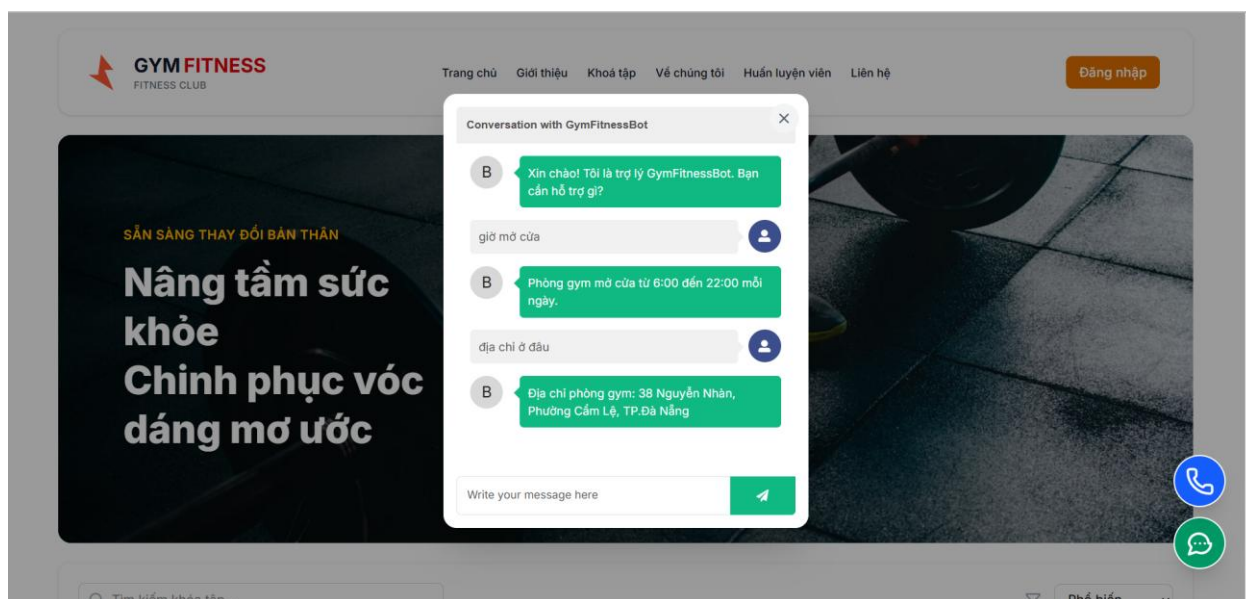
Trang thanh toán trình bày chi tiết hóa các khóa tập đã chọn. Người dùng có thể xác nhận thông tin và thực hiện thanh toán. Sau khi hoàn tất, hệ thống hiển thị thông báo thành công và lưu lại lịch sử giao dịch.



Hình 4.10 Giao diện thanh toán.

#### 4.2.11. Giao diện chat bot.

Chatbot cung cấp kênh hỗ trợ tự động cho người dùng. Người dùng có thể đặt câu hỏi về giá khóa tập, giờ mở cửa,... và chatbot sẽ trả lời ngay lập tức dựa trên dữ liệu hệ thống. Giao diện được thiết kế giống hộp trò chuyện, dễ sử dụng và thân thiện

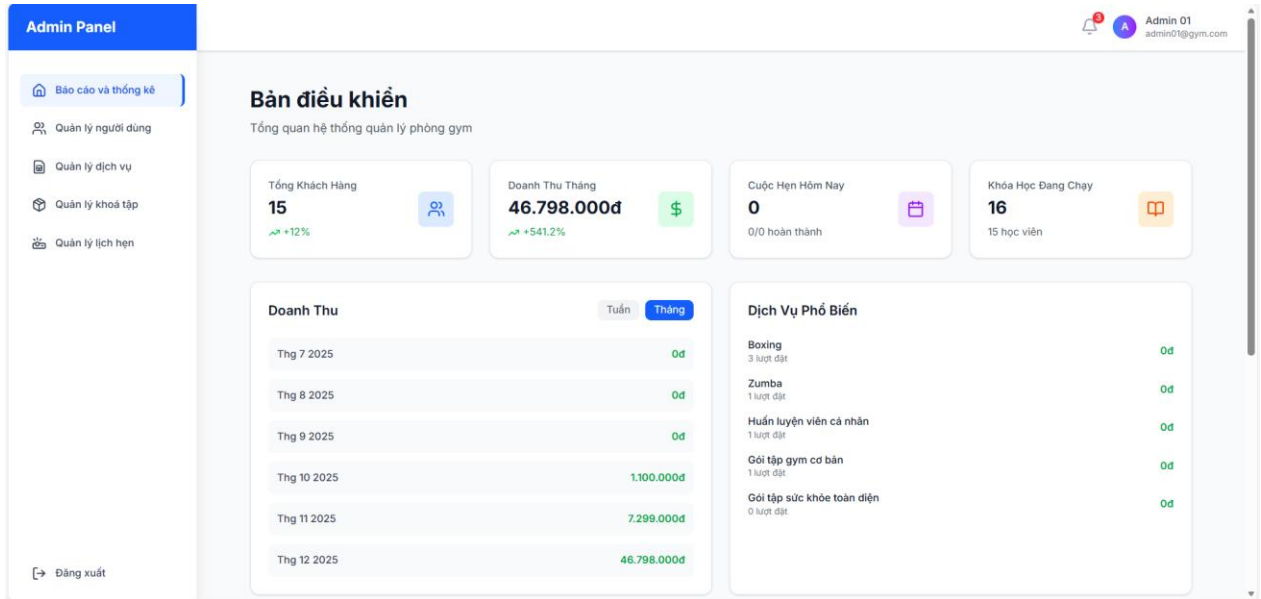


Hình 4.11 Giao diện chat bot.

### 4.3. Giao diện Admin.

#### 4.3.1. Giao diện báo cáo và thống kê.

Giao diện Báo cáo và Thống kê cung cấp cho Admin cái nhìn tổng quan về tình trạng hoạt động của hệ thống. Các chỉ số quan trọng như số lượng hội viên, doanh thu theo tuần/tháng, số lượng khóa tập.



Hình 4.12 Giao diện báo cáo và thống kê.

#### 4.3.2. Giao diện quản lý người dùng.

Đây là nơi Admin theo dõi toàn bộ thông tin của người dùng thuộc hệ thống. Dữ liệu được trình bày dưới dạng bảng danh sách, cho phép Admin thực hiện các thao tác như thêm mới, chỉnh sửa hoặc xóa hội viên. Chức năng tìm kiếm giúp Admin tra cứu nhanh khi cần.

The screenshot displays the 'Quản lý người dùng' (User Management) section of the Admin Panel. It includes a search bar, a dropdown for 'Tất cả trạng thái', and buttons for 'Bộ lọc' and 'Xuất ra'. Below is a table of users:

| TÊN NGƯỜI DÙNG | LOẠI      | SỐ ĐIỆN THOẠI | GIỚI TÍNH | EMAIL             | THAO TÁC                                                         |
|----------------|-----------|---------------|-----------|-------------------|------------------------------------------------------------------|
| Admin 01       | Admin     | 0900000001    |           | admin01@gym.com   | <a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| PT 01          | PT        | 090586984     | Nam       | pt@gym.com        | <a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| Mạnh           | Người Tập | 0983124563    | Nam       | baomanh@gmail.com | <a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| Văn Kiện       | PT        | 082456123     | Nam       | vankien@gmail.com | <a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a> |
| Hoàng Vũ       | PT        | 0905753159    | Nam       | hoangvu@gmail.com | <a href="#">View</a> <a href="#">Edit</a> <a href="#">Delete</a> |

Hình 4.13 Giao diện quản lý người dùng.

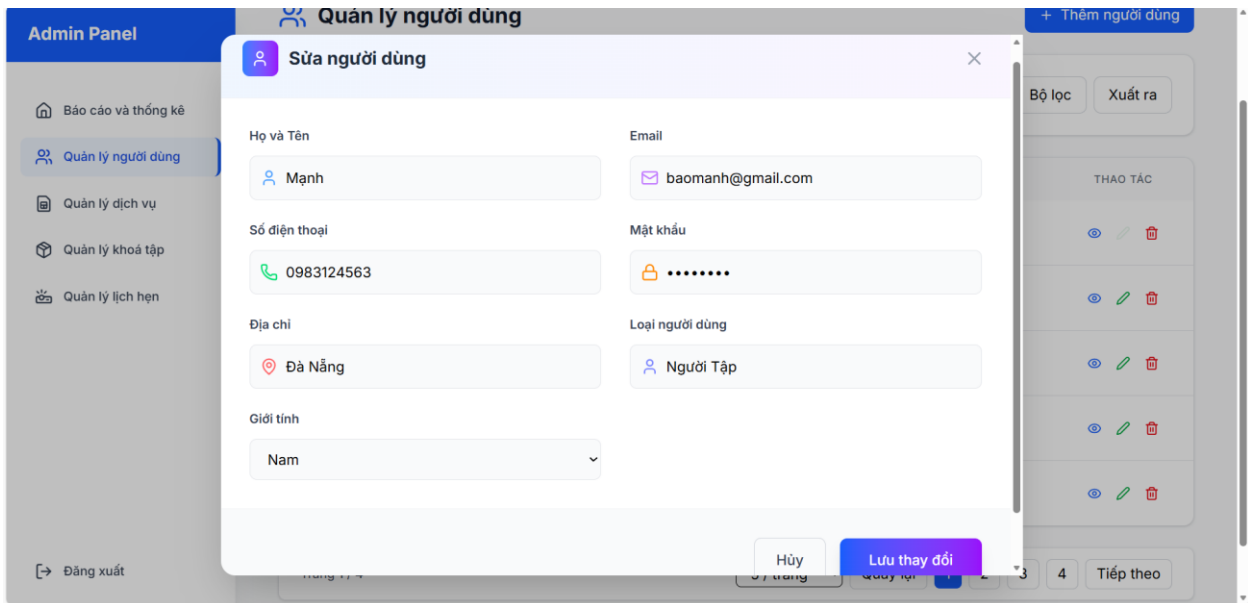
#### 4.3.3. Giao diện thêm người dùng.

Giao diện thêm người dùng giúp Admin tạo tài khoản hội viên mới trực tiếp trên hệ thống. Các trường thông tin quan trọng bao gồm: họ tên, email, số điện thoại, địa chỉ, mật khẩu và giới tính, loại người dùng. Form được thiết kế dễ thao tác, có kiểm tra tính hợp lệ của dữ liệu (như email đúng định dạng, mật khẩu đủ mạnh). Sau khi chọn “Thêm người dùng”, người dùng mới sẽ được thêm ngay vào danh sách hội viên và có thể đăng nhập vào hệ thống.

Hình 4.14 Giao diện thêm người dùng.

#### 4.3.4. Giao diện sửa người dùng.

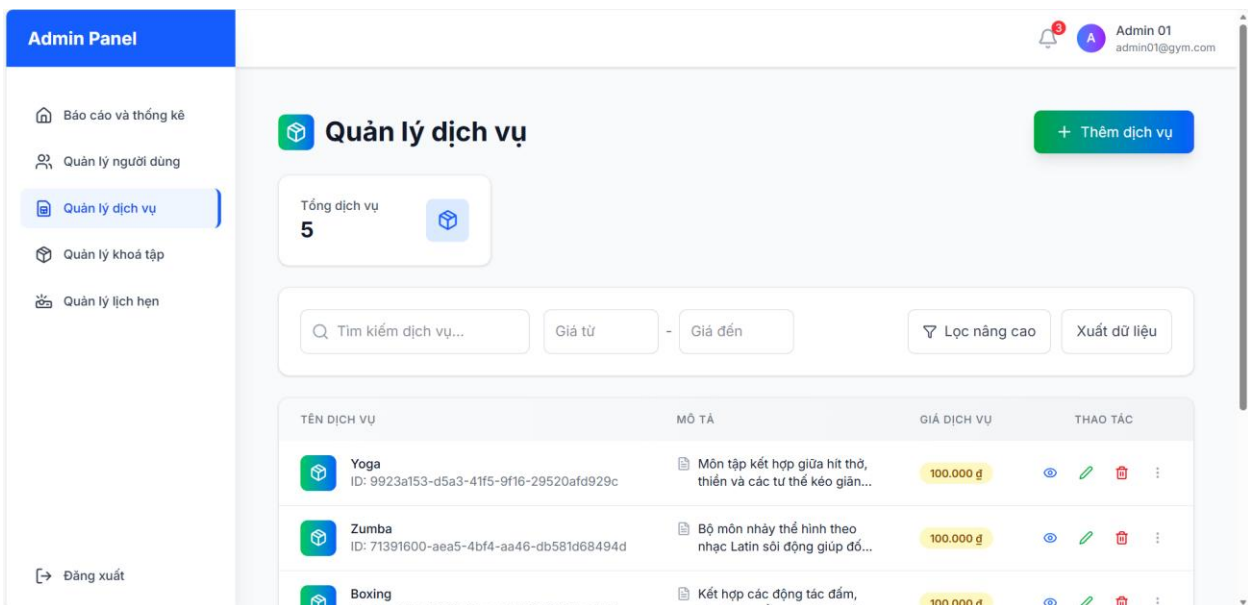
Giao diện chỉnh sửa cho phép Admin cập nhật thông tin hội viên khi có thay đổi hoặc sai sót. Toàn bộ thông tin hiện tại của hội viên được hiển thị rõ ràng, giúp Admin dễ dàng điều chỉnh. Admin có thể chỉnh sửa các thông tin như họ tên, số điện thoại, email... Sau khi lưu, dữ liệu được cập nhật ngay và đồng bộ trong toàn hệ thống.



Hình 4.15 Giao diện sửa người dùng.

#### 4.3.5. Giao diện quản lý dịch vụ.

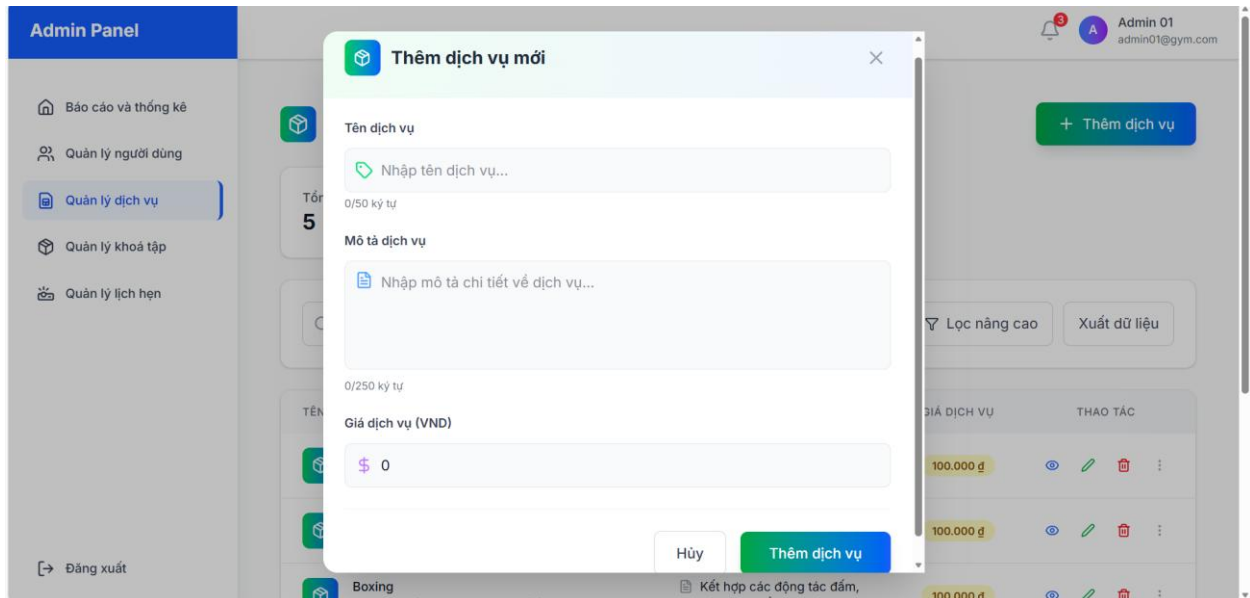
Trang quản lý dịch vụ cho phép Admin giám sát toàn bộ dịch vụ hiện có. Mỗi dịch vụ được trình bày với tên, mô tả và giá. Admin có thể dễ dàng điều chỉnh, thêm hoặc xóa dịch vụ khi cần thay đổi.



Hình 4.16 Giao diện quản lý dịch vụ

#### 4.3.6. Giao diện thêm dịch vụ.

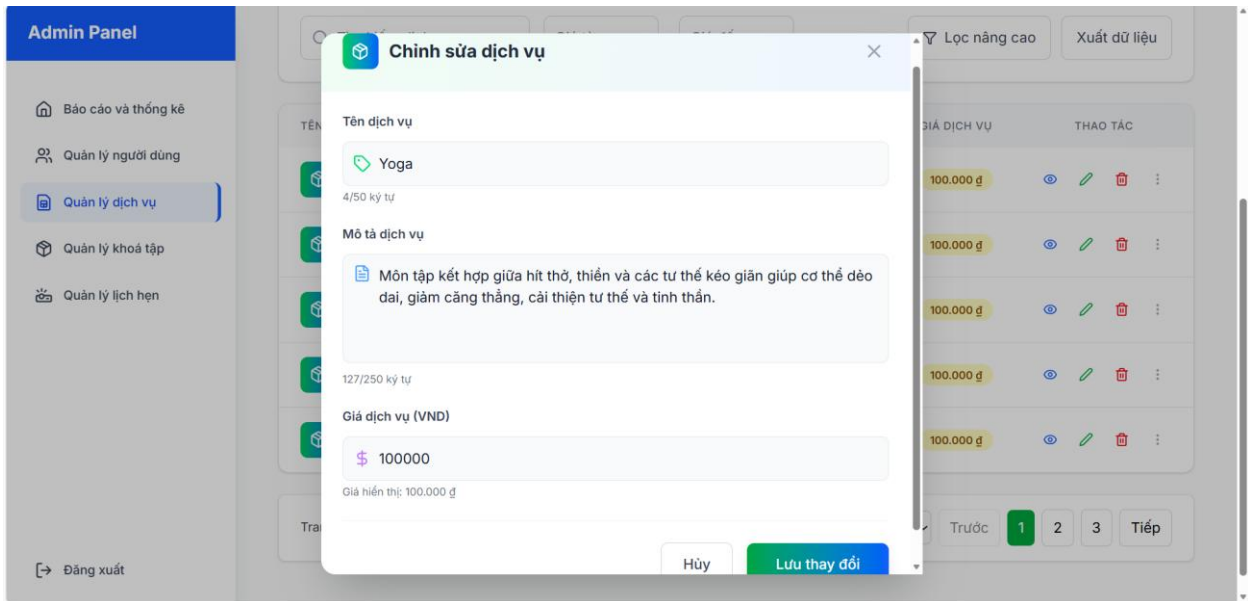
Giao diện này cho phép Admin thêm dịch vụ mới của phòng Gym. Thông tin dịch vụ bao gồm: tên dịch vụ, mô tả, giá. Form thiết kế đơn giản, trực quan, hỗ trợ nhập dữ liệu nhanh và chính xác.



Hình 4.17 Giao diện thêm dịch vụ.

#### 4.3.7. Giao diện sửa dịch vụ.

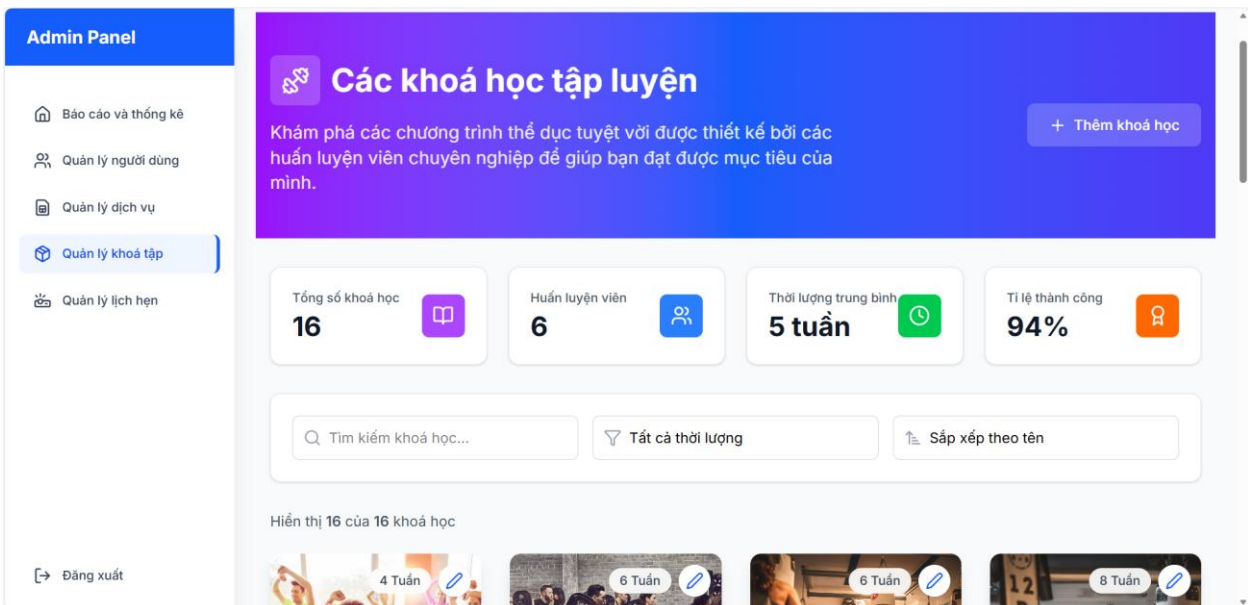
Tang chỉnh sửa dịch vụ cho phép Admin sửa lại giá, mô tả hoặc tên dịch vụ. Giao diện hiển thị toàn bộ dữ liệu hiện có để Admin dễ dàng điều chỉnh. Sau khi lưu, dịch vụ được cập nhật tức thời trên hệ thống.



Hình 4.18 Giao diện sửa dịch vụ.

#### 4.3.8. Giao diện quản lý khoá tập.

Giao diện này cho phép Admin theo dõi chi tiết các khóa tập đang được cung cấp. Thông tin khóa tập như tên khóa, mô tả, thời lượng và giá được trình bày đầy đủ. Admin có thể cập nhật hoặc tạo mới khóa tập để phù hợp với nhu cầu thực tế.



Hình 4.19 Giao diện quản lý khoá tập.

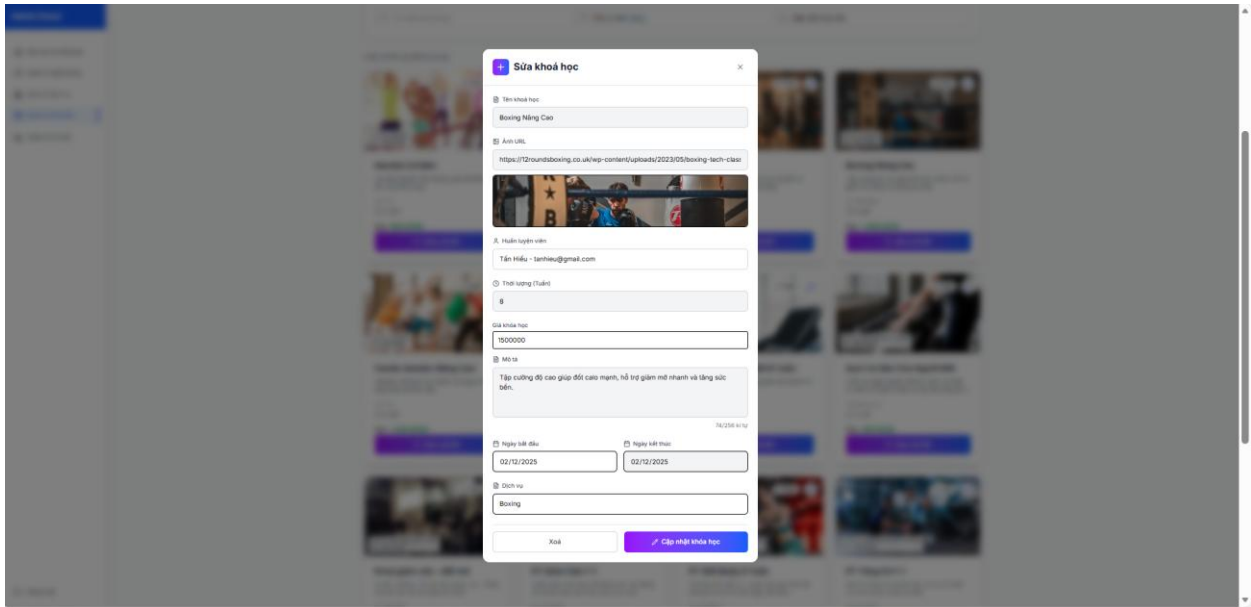
#### 4.3.9. Giao diện thêm khoá tập.

Giao diện này cho phép Admin tạo mới một khóa tập. Các trường thông tin bao gồm: tên khóa tập, mô tả, thời lượng, giá, huấn luyện viên... Form nhập liệu được bố trí khoa học, có kiểm tra dữ liệu (validation) để đảm bảo thông tin hợp lệ trước khi lưu. Sau khi nhấn “Thêm khóa tập”, hệ thống tự động cập nhật vào danh sách khóa tập.

Hình 4.20 Giao diện thêm khoá tập.

#### 4.3.10. Giao diện sửa khóa tập.

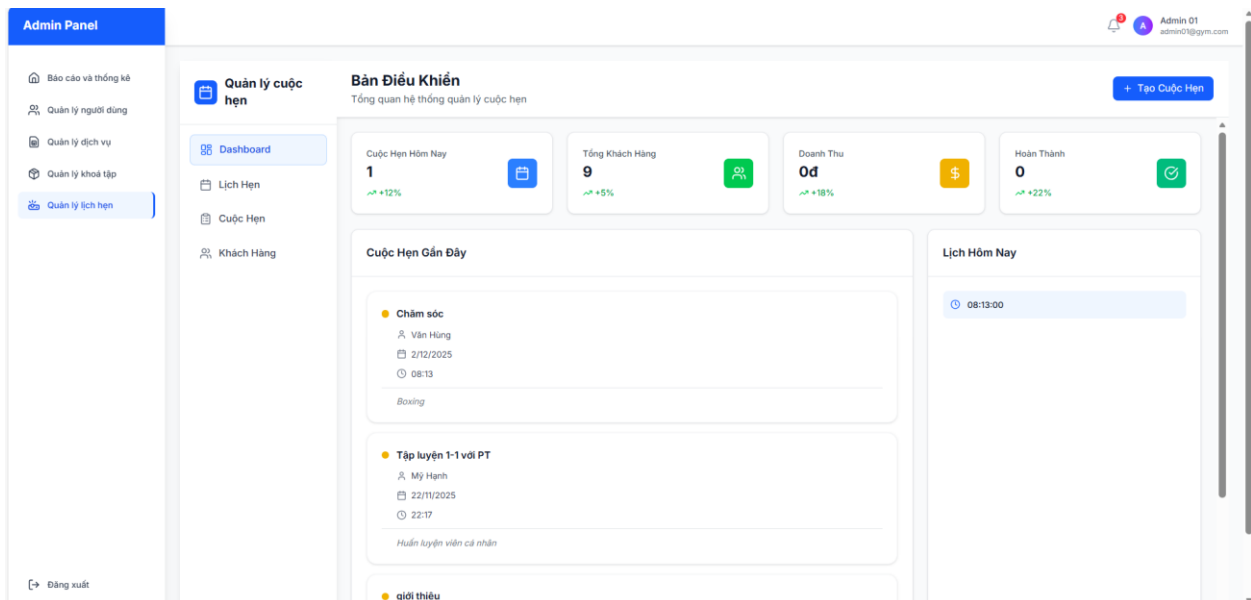
Khi Admin chọn chỉnh sửa một khóa tập, giao diện sẽ hiển thị đầy đủ thông tin hiện tại để dễ dàng thay đổi. Admin có thể cập nhật nội dung như mô tả, giá, hình ảnh hoặc thay đổi PT phụ trách... Sau khi chỉnh sửa và lưu lại, thông tin được cập nhật ngay trên hệ thống.



Hình 4.21 Giao diện sửa khoá tập.

#### 4.3.11. Giao diện quản lý lịch hẹn.

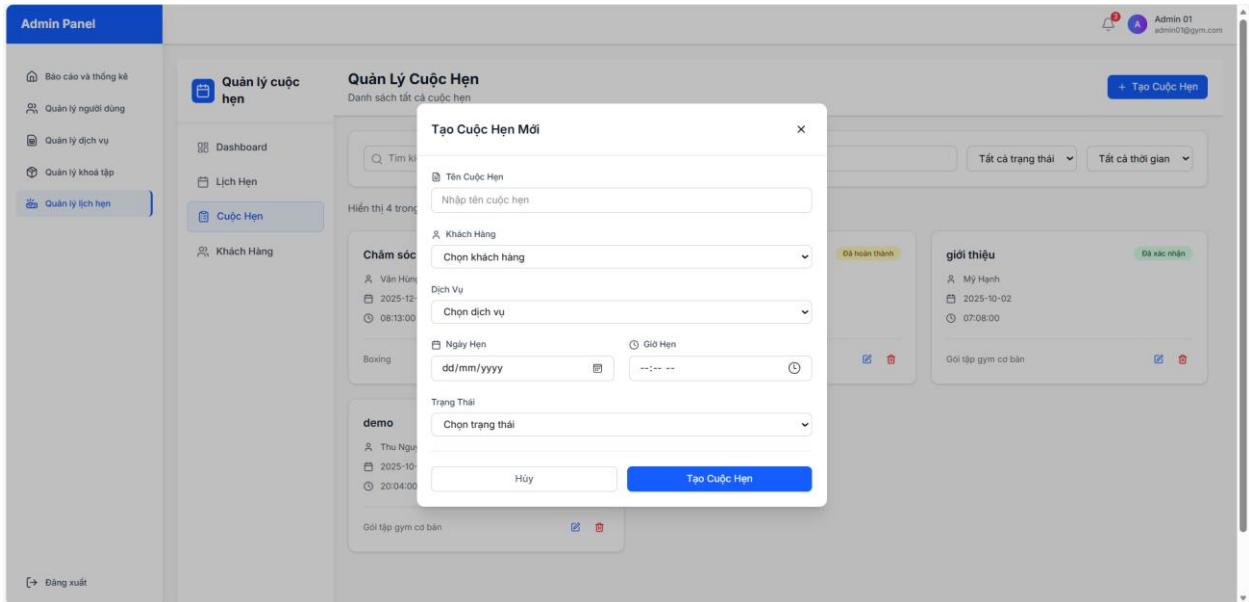
Trong giao diện lịch hẹn, Admin có thể xem danh sách các buổi hẹn giữa hội viên và PT. Giao diện hỗ trợ truy cập nhanh vào chi tiết lịch hẹn, chỉnh sửa thông tin hoặc tạo lịch hẹn mới. Điều này giúp điều phối thời gian tập luyện một cách hợp lý.



Hình 4.22 Giao diện quản lý lịch hẹn.

#### 4.3.12. Giao diện thêm lịch hẹn.

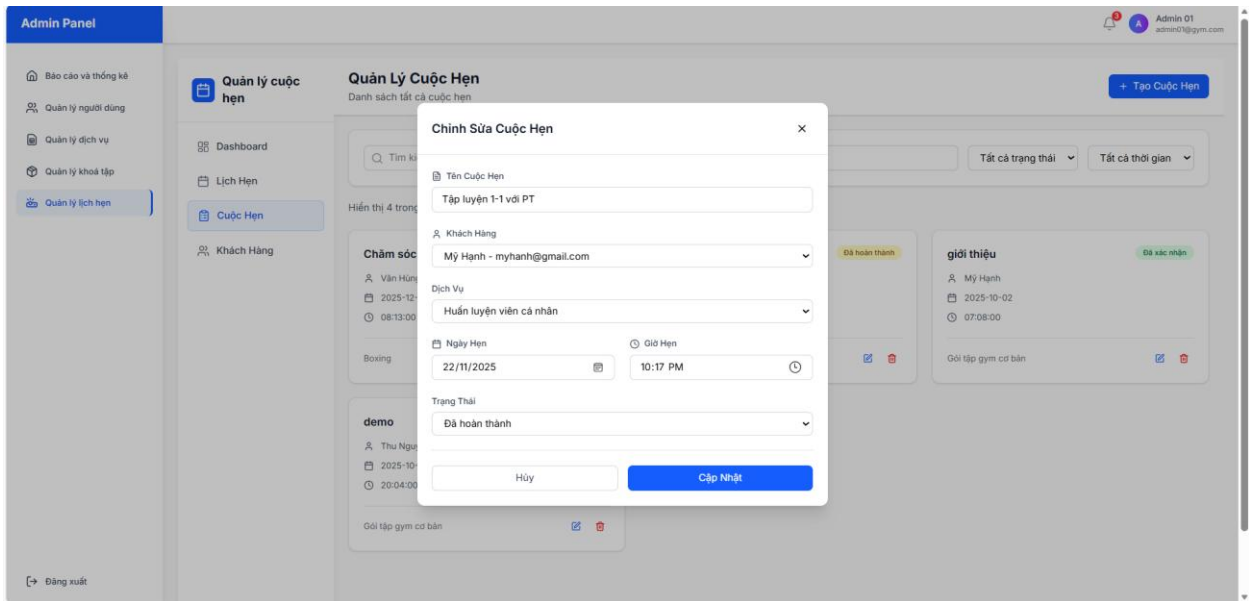
Giao diện này cho Admin tạo lịch hẹn mới giữa hội viên và PT. Các thông tin cần nhập gồm: tên hội viên, huấn luyện viên, ngày – giờ hẹn... Form nhập liệu giúp chọn ngày giờ bằng lịch (datepicker) để đảm bảo tính chính xác.



Hình 4.23 Giao diện thêm lịch hẹn.

#### 4.3.13. Giao diện sửa lịch hẹn.

Khi cần thay đổi thời gian hoặc PT, giao diện chỉnh sửa lịch hẹn cho phép cập nhật các trường cần thiết. Thông tin hiện tại của lịch hẹn được hiển thị sẵn để Admin tiện chỉnh sửa. Sau khi lưu, lịch hẹn được cập nhật ngay lập tức.

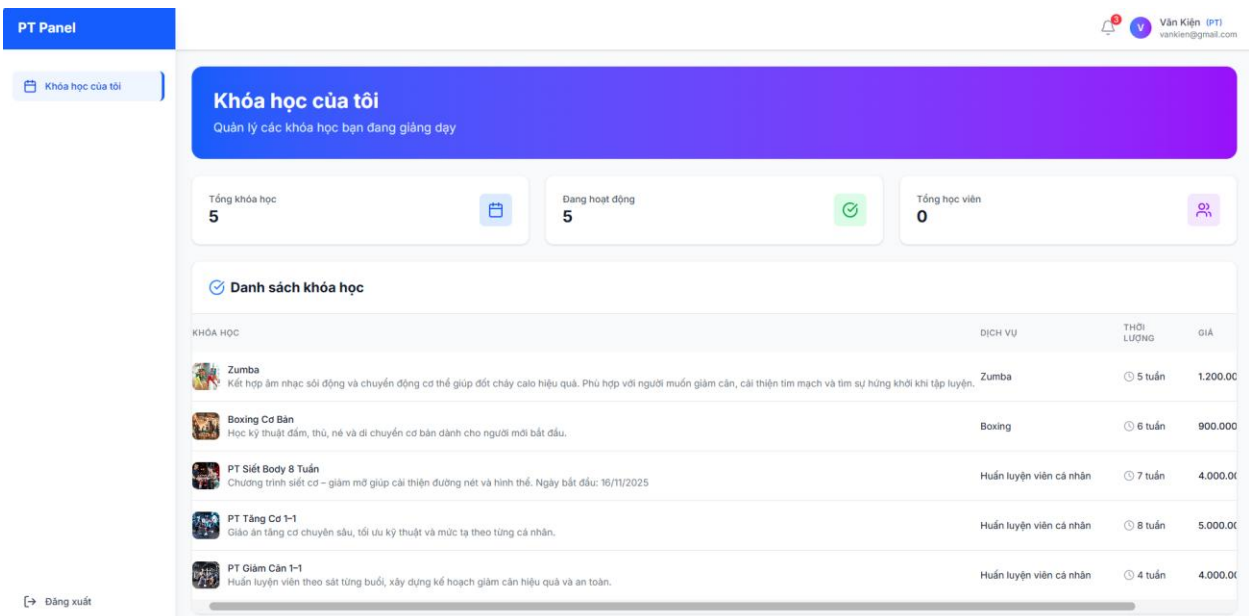


Hình 4.24 Giao diện sửa lịch hẹn.

#### 4.4. Giao diện huấn luyện viên.

##### 4.4.1. Giao diện khoá học phụ trách (Huấn luyện viên).

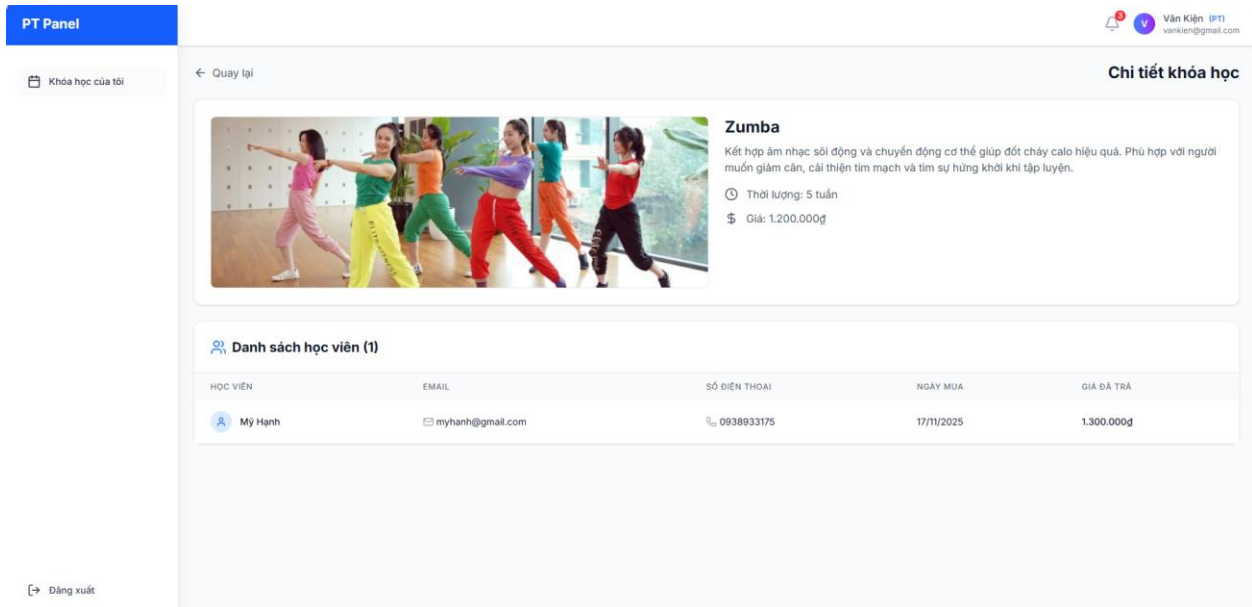
Huấn luyện viên có thể xem danh sách các khóa tập mình đang đảm nhiệm, bao gồm thông tin về thời lượng, tên khóa học. Giao diện này giúp PT chủ động tổ chức lịch giảng dạy và quản lý học viên hiệu quả hơn.



Hình 4.25 Giao diện khoá học phụ trách.

#### 4.4.2. Giao diện danh sách học viên (Huấn luyện viên).

Trang này liệt kê các học viên thuộc các lớp huấn luyện viên phụ trách. Thông tin hiển thị ngắn gọn, dễ quan sát giúp huấn luyện viên có thể theo dõi và liên hệ khi cần thiết.



Hình 4.26 Giao diện danh sách học viên.

## KẾT LUẬN

### 1. Kết quả đạt được

Sau quá trình xây dựng và triển khai, hệ thống website quản lý phòng Gym thông minh đã hoàn thiện với đầy đủ các thành phần từ giao diện người dùng đến xử lý dữ liệu phía server. Ứng dụng được phát triển trên nền React/Next.js ở front-end và ASP.NET Core 8.0 ở back-end, kết nối với cơ sở dữ liệu PostgreSQL, tạo nên một hệ thống hoạt động ổn định, hiện đại và dễ mở rộng.

Các kết quả nổi bật gồm có:

- Xây dựng thành công hệ thống web gym full-stack hoàn chỉnh, đảm bảo hoạt động ổn định.
- Giao diện thân thiện, dễ sử dụng, nhờ áp dụng React và Next.js giúp tốc độ tải nhanh, điều hướng mượt mà, hỗ trợ SEO tốt.
- Back-end rõ ràng, dễ bảo trì, sử dụng ASP.NET Core 8.0 và kiến trúc controller/service/repository, giúp việc mở rộng chức năng sau này thuận lợi.
- Cơ sở dữ liệu PostgreSQL hoạt động ổn định, lưu trữ an toàn dữ liệu hội viên, gói tập, nhân viên, lịch sử thanh toán,...
- Tích hợp được chức năng thanh toán và xem lịch sử giao dịch, cho phép người dùng theo dõi các lần thanh toán trực tiếp từ giao diện Front-end thông qua API.

### 2. Hạn chế

Bên cạnh các kết quả đạt được, hệ thống vẫn còn một số hạn chế và chưa khai thác hết tiềm năng của các công nghệ sử dụng. Một số tính năng nâng cao của Next.js và ASP.NET Core chưa được tận dụng tối đa, quá trình kiểm thử và tối ưu hiệu năng còn hạn chế, đồng thời các yếu tố bảo mật và triển khai thực tế cần được cải thiện thêm.

Một số hạn chế cụ thể:

- Chưa tận dụng đầy đủ các tính năng nâng cao của Next.js như tối ưu hình ảnh, ISG, middleware,...

- Chưa sử dụng hết sức mạnh của ASP.NET Core 8.0, chẳng hạn như minimal APIs, endpoint filters, background services.
- Thiếu bộ kiểm thử đầy đủ (unit test, integration test) cho cả front-end và back-end.
- Hiệu năng chưa được benchmark chi tiết, bao gồm truy vấn PostgreSQL, tối ưu index, connection pool,...
- Bảo mật mới ở mức cơ bản, cơ chế xác thực, phân quyền, logging, monitoring, CI/CD chưa hoàn thiện cho môi trường production.

### 3. Định hướng phát triển trong tương lai

Trong thời gian tới, hệ thống có thể được mở rộng theo hướng hoàn thiện hơn về tính năng người dùng, tăng cường kiến trúc back-end, tối ưu hiệu năng, và nâng cao trải nghiệm người dùng. Bên cạnh đó, việc tích hợp thêm các giải pháp thanh toán thực tế và triển khai hệ thống trên môi trường sản xuất chuyên nghiệp cũng là mục tiêu quan trọng.

Các hướng phát triển chính bao gồm:

- Hoàn thiện tính năng người dùng và quản lý
  - Thêm dashboard cá nhân: theo dõi tiến trình tập luyện, số đo, lịch sử buổi tập.
  - Chức năng đặt lịch PT, đăng ký lớp nhóm, gửi thông báo nhắc lịch.
  - Mở rộng trang thanh toán với VNPay, MoMo, PayOS và xuất hóa đơn.
- Nâng cao kiến trúc hệ thống
  - Áp dụng Clean Architecture hoặc DDD cho ASP.NET Core.
  - Sử dụng Entity Framework Core với migration chặt chẽ, tối ưu truy vấn PostgreSQL.
  - Bổ sung bộ kiểm thử đầy đủ (unit, integration, E2E).
- Tối ưu front-end và trải nghiệm người dùng
  - Sử dụng tốt hơn các tính năng mạnh của Next.js: Server Actions, App Router, caching, image optimization.
  - Tối ưu responsive, lazy load, skeleton loading, code-splitting.
- Triển khai và vận hành thực tế
  - Hoàn thiện cấu hình production: logging, monitoring, backup DB, CI/CD, tối ưu bảo mật hệ thống và quản lý phiên người dùng.

## TÀI LIỆU THAM KHẢO

- [1] Vercel, Next.js Documentation – The React Framework for the Web,  
<https://nextjs.org/docs>
- [2] React Team, React Documentation (React 19.0.0), Meta,  
<https://react.dev>
- [3] Microsoft, ASP.NET Core 8.0 Documentation, Microsoft Learn,  
<https://learn.microsoft.com/aspnet/core>
- [4] PostgreSQL Global Development Group, PostgreSQL 16 Documentation,  
<https://www.postgresql.org/docs/>
- [5] Stripe, Stripe.NET API Reference & Integration Guide,  
<https://docs.stripe.com>
- [6] Visual Studio 2022 Documentation – Microsoft  
<https://learn.microsoft.com/en-us/visualstudio/>
- [7] Fredrik Oseberg, React Chatbot Kit Documentation,  
<https://fredrikoseberg.github.io/react-chatbot-kit-docs>
- [8] Giáo trình phân tích và thiết kế hệ thống thông tin,  
<https://thuvienso.dau.edu.vn:88/handle/DHKTDN/1178?locale=vi>

---Hết---

