TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ I, NĂM HỌC 2025-2026**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ HOẠT ĐỘNG CỦA CỐ VẤN HỌC TẬP TẠI TRƯỜNG ĐẠI HỌC TRÀ VINH**

|  |  |
| --- | --- |
| *Giảng viên hướng dẫn:*  ThS.Nguyễn Ngọc Đan Thanh | *Sinh viên thực hiện:*  Họ tên: Phan Nguyễn Hoàng Hân  MSSV:110122003  Lớp: DA22TTA |

**Vĩnh Long, tháng 12 năm 2025**

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH**

**HỌC KỲ I, NĂM HỌC 2025-2026**

**XÂY DỰNG HỆ THỐNG QUẢN LÝ HOẠT ĐỘNG CỦA CỐ VẤN HỌC TẬP TẠI TRƯỜNG ĐẠI HỌC TRÀ VINH**

|  |  |
| --- | --- |
| *Giảng viên hướng dẫn:*  ThS.Nguyễn Ngọc Đan Thanh | *Sinh viên thực hiện:*  Họ tên: Phan Nguyễn Hoàng Hân  MSSV:110122003  Lớp: DA22TTA |

**Vĩnh Long, tháng 12 năm 2025**

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

*Vĩnh Long, ngày … tháng … năm 2025*

**Giảng viên hướng dẫn**

*(Ký tên và ghi rõ họ tên)*

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

*Vĩnh Long, ngày … tháng … năm 2025*

**Thành viên hội đồng**

*(Ký tên và ghi rõ họ tên)*

LỜI CẢM ƠN

Em xin chân thành gửi lời cảm ơn đến quý thầy cô bộ môn trong khoa Kỹ Thuật và Công Nghệ đã trang bị những kiến thức trong suốt quá trình học tập, tạo điều kiện cho chúng em cơ hội thực hành, tiếp xúc để có đủ khả năng thực hiện đồ án cơ sở ngành của bản thân.

Trong quá trình thực hiện đồ án, nhờ sự giúp đỡ tận tình và chỉ bảo của các thầy cô trong Khoa Kỹ thuật và Công nghệ, đặc biệt là cô Nguyễn Ngọc Đan Thanh đã luôn lắng nghe và giành thời gian quý báu chỉ bảo từ lúc bắt đầu cho tới lúc kết thúc đồ án mà em đã hoàn thành đúng thời hạn quy định và tích lũy được cho mình một lượng nền tảng kiến thức cũng như kinh nghiệm vô cùng quý giá.

Với những kiến thức nền tảng hiện có, việc hoàn thành đồ án một cách chỉn chu nhất là một điều khó khăn, em kính mong nhận được sự góp ý, chỉ bảo thêm của các thầy cô để đồ án ngày càng hoàn thiện hơn.

Cuối lời, em xin kính chúc quý Thầy Cô lời chúc sức khỏe và thành công! Em xin chân thành cảm ơn!

Vĩnh Long, ngày … tháng ... năm 2025

SINH VIÊN

Phan Nguyễn Hoàng Hân

MỤC LỤC

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN i](#_Toc6187)

[NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG ii](#_Toc8964)

[LỜI CẢM ƠN iii](#_Toc11501)

[MỤC LỤC iv](#_Toc5126)

[DANH MỤC HÌNH ẢNH viii](#_Toc937)

[DANH MỤC TỪ VIẾT TẮT x](#_Toc27561)

[TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH xi](#_Toc24035)

[MỞ ĐẦU 1](#_Toc23476)

[CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU 2](#_Toc11061)

[1.1 Hệ thống Flightpath – Academic Advising System 2](#_Toc20104)

[1.1.1 Tổng quan hiện trạng các hệ thống cố vấn học tập dựa trên mã nguồn mở 2](#_Toc10492)

[1.1.2 Phân tích về nền tảng công nghệ và kỹ thuật xử lý của FlightPath 2](#_Toc7813)

[1.1.3 Đánh giá về mô hình nghiệp vụ và dịch vụ 2](#_Toc30893)

[1.1.4 Đánh giá về giao diện và trải nghiệm người dùng 3](#_Toc7574)

[1.1.5 Nhận xét chung 3](#_Toc10667)

[1.1.6 Nhận xét riêng 3](#_Toc5680)

[1.1.7 Kết luận và hướng tiếp cận của đề tài 4](#_Toc23237)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 5](#_Toc31052)

[2.1 Python framework: Django 5](#_Toc30930)

[2.1.1 Kiến trúc MVT 6](#_Toc18719)

[2.1.2 Cấu trúc thư mục của Django 7](#_Toc2821)

*[2.1.2.1](#_Toc3743)* [Thư mục gốc dự án 7](#_Toc3743)

*[2.1.2.2](#_Toc17124)* [Thư mục project (trùng tên với dự án) 7](#_Toc17124)

*[2.1.2.3](#_Toc22542)* [Thư mục ứng dụng 8](#_Toc22542)

*[2.1.2.4](#_Toc29969)* [Các thư mục mở rộng thường dùng 8](#_Toc29969)

[2.1.3 Những thành phần chính có trong Django 8](#_Toc28977)

*[2.1.3.1](#_Toc2375)* [Thành phần Views 8](#_Toc2375)

*[2.1.3.2](#_Toc18901)* [URLS 8](#_Toc18901)

*[2.1.3.3](#_Toc9005)* [Models 9](#_Toc9005)

*[2.1.3.4](#_Toc26057)* [Hệ thống quản trị (Admin) 10](#_Toc26057)

*[2.1.3.5](#_Toc4860)* [Những bảng có sẵn trong Django 11](#_Toc4860)

*[2.1.3.6](#_Toc18818)* [Tạo dự án 13](#_Toc18818)

*[2.1.3.7](#_Toc21504)* [Chạy server dự án Django 15](#_Toc21504)

[2.2 Hệ quản trị cơ sở dữ liệu MySQL 15](#_Toc12545)

[2.2.1 Khái quát về MySQL 15](#_Toc20609)

[2.2.2 Ưu và nhược điểm của MySQL 16](#_Toc2731)

[2.2.3 Kết nối MySQL vào Django 17](#_Toc7576)

[2.3 Cài đặt và hiện thực hóa đề tài 19](#_Toc15934)

[2.3.1 Khảo sát bài toán và xác định yêu cầu hệ thống 19](#_Toc14070)

[2.3.2 Lựa chọn công nghệ và mô hình triển khai 19](#_Toc18435)

[2.3.3 Chuẩn bị môi trường và tạo dự án Django 19](#_Toc16815)

[2.3.4 Tạo các ứng dụng chức năng 20](#_Toc13029)

[2.3.5 Xây dựng mô hình dữ liệu (Model) 20](#_Toc22144)

[2.3.6 Hiện thực hóa chức năng nghiệp vụ 21](#_Toc4138)

[2.3.7 Xây dựng giao diện người dùng 21](#_Toc5933)

[2.3.8 Chuẩn bị các file để triển khai Docker 22](#_Toc16844)

[2.3.9 Build và chạy thử hệ thống bằng Docker 23](#_Toc12267)

[2.3.10 Đưa mã nguồn lên GitHub 23](#_Toc21706)

[2.4 Nền tảng Docker 23](#_Toc11502)

[2.4.1 Docker là gì? 23](#_Toc21231)

[2.4.2 Nguyên lý hoạt động của Docker 23](#_Toc3475)

[2.4.3 Lợi ích của Docker 24](#_Toc28639)

[CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU 25](#_Toc2500)

[3.1 Mô tả hệ thống 25](#_Toc21349)

[3.2 Đặc tả yêu cầu 25](#_Toc25476)

[3.2.1 Các chức năng của hệ thống 25](#_Toc4950)

[3.2.2 Yêu cầu phi chức năng 27](#_Toc11712)

[3.3 Thiết kế dữ liệu 28](#_Toc16786)

[3.3.1 Mô hình dữ liệu mức quan niệm 28](#_Toc31456)

[3.3.2 Mô hình vật lý 28](#_Toc22234)

[3.3.3 Mô tả thực thể 28](#_Toc14509)

[3.4 Phân tích và thiết kế hệ thống 32](#_Toc18845)

[3.4.1 Sơ đồ kiến trúc tổng thể 32](#_Toc27777)

[3.4.2 Usecase tổng quát 33](#_Toc22507)

[3.4.3 Biểu đồ Use Case tác nhân Sinh viên 34](#_Toc23006)

[3.4.4 Biểu đồ Use Case tác nhân Cố vấn học tập 35](#_Toc1572)

[3.4.5 Biểu đồ Use Case tác nhân quản trị 36](#_Toc12000)

[3.5 Thiết kế giao diện 37](#_Toc8240)

[3.5.1 Sơ đồ tổ chức thông tin 37](#_Toc27946)

[3.5.2 Giao diện đăng nhập 38](#_Toc14639)

[3.5.3 Giao diện trang chủ 38](#_Toc15157)

[3.5.4 Giao diện trang sinh viên 39](#_Toc17934)

[3.5.5 Giao diện trang lịch tư vấn 40](#_Toc26932)

[3.5.6 Giao diện trang thông báo 40](#_Toc4451)

[3.5.7 Giao diện trang thống kê 41](#_Toc3552)

[3.5.8 Giao diện tranh xuất báo cáo 42](#_Toc31609)

[CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM 43](#_Toc752)

[4.1 Giao diện trang đăng nhập 43](#_Toc20345)

[4.2 Giao diện cố vấn học tập đăng nhập 44](#_Toc1544)

[4.2.1 Giao diện trang chủ 44](#_Toc32522)

[4.2.2 Giao diện trang quản lý sinh viên 44](#_Toc10850)

[4.2.3 Giao diện trang lịch tư vấn 46](#_Toc24558)

[4.2.4 Giao diện thông báo 47](#_Toc26754)

[4.2.5 Trang thống kê kết quả học tập 47](#_Toc9042)

[4.2.6 Giao diện trang thay đổi mật khẩu 48](#_Toc17577)

[4.3 Giao diện sinh viên đăng nhập 49](#_Toc15611)

[4.3.1 Giao diện trang chủ 49](#_Toc2548)

[4.3.2 Giao diện trang thông tin cá nhân 49](#_Toc24687)

[4.3.3 Giao diện trang lịch tư vấn 50](#_Toc18237)

[4.3.4 Giao diện trang thông báo 51](#_Toc11463)

[4.3.5 Trang thống kê kết quả học tập 52](#_Toc25658)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 53](#_Toc9180)

[5.1 Kết quả đạt được 53](#_Toc6261)

[5.2 Hạn chế 53](#_Toc12252)

[5.3 Hướng phát triển 54](#_Toc6920)

[TÀI LIỆU THAM KHẢO 55](#_Toc21134)

DANH MỤC HÌNH ẢNH

[Hình 2.1 Giao diện trang admin của Django 11](#_Toc26032)

[Hình 2.2 Giao diện trang django ban đầu 15](#_Toc13753)

[Hình 2.3 Giao diện kết nối cơ sở dữ lệu của MySQL Workbench 17](#_Toc6207)

[Hình 3.1 Mô hình dữ liệu mức quan niệm 28](#_Toc10231)

[Hình 3.2 Mô hình vật lý 28](#_Toc1379)

[Hình 3.3 Sơ đồ kiến trúc tổng thể hệ thống 33](#_Toc28694)

[Hình 3.3 Sơ đồ tổng quát hệ thống 34](#_Toc27907)

[Hình 3.3 Biểu đồ usecase tác nhân sinh viên 35](#_Toc11612)

[Hình 3.4 Biểu đồ usecase tác nhân Cố vấn học tập 36](#_Toc15462)

[Hình 3.5 Biểu đồ usecase tác nhân Quản trị 37](#_Toc32443)

[Hình 3.6 Sơ đồ tổ chức thông tin 37](#_Toc2640)

[Hình 3.7 Thiết kế giao diện trang đăng nhập hệ thống 38](#_Toc19714)

[Hình 3.8 Thiết kế giao diện trang chủ hệ thống 39](#_Toc32014)

[Hình 3.9 Thiết kế giao diện trang sinh viên 39](#_Toc19252)

[Hình 3.10 Thiết kế giao diện trang lịch tư vấn 40](#_Toc12756)

[Hình 3.11 Thiết kế giao diện trang thông báo 41](#_Toc1775)

[Hình 3.12 Thiết kế giao diện trang thống kê 42](#_Toc14362)

[Hình 3.13 Thiết kế giao diện trang xuất báo cáo 42](#_Toc28885)

[Hình 4.1 Giao diện trang đăng nhập hệ thống 43](#_Toc2271)

[Hình 4.2 Giao diện xác thực tài khoản để lấy lại mật khẩu 43](#_Toc17733)

[Hình 4.3 Giao diện trang chủ của cố vấn học tập 44](#_Toc24820)

[Hình 4.4 Giao diện trang sinh viên 44](#_Toc16370)

[Hình 4.5 Trang thôn tin cá nhân chi tiết của sinh viên 45](#_Toc24261)

[Hình 4.6 Trang hiển thị kết quả học tập chi tiết của sinh viên 45](#_Toc6423)

[Hình 4.7 Trang thống kê học tập của sinh viên 46](#_Toc1844)

[Hình 4.8 Giao diện trang quản lý lịch tư vấn 46](#_Toc339)

[Hình 4.9 Giao diện tạo lịch tư vấn mới 47](#_Toc12932)

[Hình 4.10 Giao diện trang thông báo của cố vấn 47](#_Toc14715)

[Hình 4.11 Giao diện thống kê kết quả học tập của lớp học 48](#_Toc10598)

[Hình 4.12 Giao diện xuất báo cáo chi tiết 48](#_Toc3554)

[Hình 4.13 Giao diện trang thay đổi mật khẩu 49](#_Toc15270)

[Hình 4.14 Giao diện trang chủ sinh viên đăng nhập 49](#_Toc30944)

[Hình 4.15 Giao diện trang thông tin cá nhân của sinh viên 50](#_Toc9485)

[Hình 4.16 Giao diện trang lịch tư vấn của sinh viên 50](#_Toc3751)

[Hình 4.17 Giao diện đăng ký lịch tư vấn mới 51](#_Toc31337)

[Hình 4.18 Giao diện thông báo của sinh viên 51](#_Toc9378)

[Hình 4.19 Giao diện đánh giá phản hồi của cố vấn 52](#_Toc26600)

[Hình 4.20 Giao diện trang thống kê kết quả học tập của sinh viên 52](#_Toc30966)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1 Ưu điểm của Django framework 6](#_Toc1224)

[Bảng 2.2 Nhược điểm của Django framework 7](#_Toc20569)

[Bảng 2.3 Ưu điểm của MySQL 16](#_Toc26935)

[Bảng 2.4 Nhược điểm của MySQL 17](#_Toc30946)

[Bảng 3.1 Bảng Khoa 28](#_Toc5187)

[Bảng 3.2 Bảng lop 29](#_Toc5993)

[Bảng 3.3 Bảng co\_van 29](#_Toc12492)

[Bảng 3.4 Bảng sinh\_vien 29](#_Toc21522)

[Bảng 3.5 Bảng hoc\_ky 30](#_Toc21537)

[Bảng 3.6 Bảng diem\_hoc\_tap 30](#_Toc11718)

[Bảng 3.7 Bảng diem\_ren\_luyen 30](#_Toc3342)

[Bảng 3.8 Bảng mon\_hoc 31](#_Toc32126)

[Bảng 3.9 Bảng ket\_qua\_mon\_hoc 31](#_Toc18566)

[Bảng 3.10 Bảng lich\_tu\_van 31](#_Toc31959)

[Bảng 3.11 Bảng phieu\_tu\_van 32](#_Toc8281)

[Bảng 3.12 Bảng danh\_gia 32](#_Toc12897)

DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Ý nghĩa** |
| CRM | Quản trị quan hệ khách hàng (Customer Relationship Management) |

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Vấn đề nghiên cứu

Đồ án "Xây dựng hệ thống quản lý hoạt động của Cố vấn học tập tại Trường Đại học Trà Vinh" tập trung nghiên cứu và ứng dụng ngôn ngữ lập trình Python cùng framework Django để phát triển một hệ thống hỗ trợ công tác quản lý, theo dõi và đánh giá hoạt động của Cố vấn học tập trong nhà trường.

Mục tiêu của đề tài là xây dựng một hệ thống giúp tự động hóa quy trình quản lý, lưu trữ thông tin sinh viên, ghi nhận hoạt động tư vấn học tập, thống kê báo cáo hiệu quả công tác, góp phần hỗ trợ cho giảng viên cố vấn và Ban chủ nhiệm Khoa trong việc nâng cao chất lượng quản lý sinh viên.

Cách hướng tiếp cận

Để triển khai đề tài, em tiến hành tìm hiểu mô hình hoạt động và quy trình làm việc của Cố vấn học tập tại Trường Đại học Trà Vinh, đồng thời tham khảo các hệ thống quản lý sinh viên và các website nội bộ của Trường để nắm bắt cấu trúc dữ liệu và yêu cầu thực tế.

Song song đó, em nghiên cứu về ngôn ngữ Python, Django framework, cùng MySQL để thiết kế cơ sở dữ liệu và phát triển hệ thống web động. Django được lựa chọn vì có bảo mật cao, quản lý dữ liệu hiệu quả và tích hợp công cụ quản trị mạnh mẽ, rất phù hợp cho hệ thống quản lý nội bộ của trường học.

Cách giải quyết vấn đề

Để giải quyết vấn đề, em sử dụng Django framework để phát triển Backend, nơi quản lý toàn bộ dữ liệu. Django giúp quản lý cơ sở dữ liệu trực tiếp thông qua giao diện người dùng, cho phép thao tác dễ dàng mà không cần truy cập vào cơ sở dữ liệu riêng biệt. Đồng thời, Frontend được xây dựng với HTML, CSSBoostrap và Django Templates để người dùng có thể thao tác trực quan.

Kết quả đạt được

Kết quả từ đồ án "Xây dựng hệ thống quản lý hoạt động của Cố vấn học tập tại Trường Đại học Trà Vinh" là một hệ thống web hoàn chỉnh, có thể: Quản lý thông tin giảng viên cố vấn, lớp sinh viên phụ trách và các buổi tư vấn học tập; Cho phép thêm, sửa, xóa thông tin, ghi nhận kết quả tư vấn, và tổng hợp báo cáo hoạt động; Giao diện thân thiện, dễ sử dụng, phù hợp cho cả người quản trị và giảng viên; Hệ thống phân quyền rõ ràng, đảm bảo tính bảo mật và hiệu quả trong quản lý.

Mô hình MTV trong Django giúp hệ thống được phát triển một cách có tổ chức, dễ dàng bảo trì, mở rộng và nâng cấp các chức năng trong tương lai, như: tích hợp điểm rèn luyện, cảnh báo học vụ, hoặc báo cáo tự động gửi về khoa.

MỞ ĐẦU

**LÝ DO CHỌN ĐỀ TÀI**

Em chọn đề tài “Xây dựng hệ thống quản lý hoạt động của Cố vấn học tập tại Trường Đại học Trà Vinh” vì nhận thấy công tác cố vấn học tập hiện nay vẫn còn nhiều thao tác thủ công, việc tổng hợp báo cáo và thống kê chưa được số hóa hoàn toàn, gây tốn thời gian và dễ sai sót.

Việc áp dụng Python Django – một công nghệ hiện đại, dễ phát triển và mở rộng sẽ giúp tự động hóa và tối ưu hóa quy trình quản lý, đồng thời giúp em rèn luyện kỹ năng lập trình web, hiểu sâu về quy trình phát triển phần mềm và chuẩn bị hành trang cho công việc thực tế sau này.

**Mục đích nghiên cứu**

Mục đích của đề tài là xây dựng hệ thống web quản lý hoạt động cố vấn học tập giúp:

- Quản lý và lưu trữ thông tin sinh viên, lớp học, giảng viên cố vấn.

- Theo dõi, ghi nhận các buổi gặp gỡ tư vấn học tập, đánh giá kết quả, và tự động thống kê báo cáo.

- Hỗ trợ ban chủ nhiệm khoa dễ dàng giám sát, đánh giá hiệu quả hoạt động của giảng viên cố vấn.

- Hệ thống được thiết kế đơn giản, thân thiện và dễ mở rộng, giúp người dùng có thể thao tác dễ dàng mà không cần kiến thức chuyên sâu về công nghệ.

**Đối tượng nghiên cứu**

Đối tượng nghiên cứu của đề tài là hoạt động quản lý công tác Cố vấn học tập tại Trường Đại học Trà Vinh, bao gồm quy trình làm việc của giảng viên cố vấn, dữ liệu về sinh viên, lớp học, buổi tư vấn học tập, cùng quy trình báo cáo và thống kê kết quả tư vấn phục vụ công tác quản lý của Khoa và Nhà trường.

# TỔNG QUAN NGHIÊN CỨU

## Hệ thống Flightpath – Academic Advising System

### Tổng quan hiện trạng các hệ thống cố vấn học tập dựa trên mã nguồn mở

Hiện nay, trong lĩnh vực quản lý cố vấn học tập tại các trường đại học, các hệ thống mã nguồn mở như FlightPath được xem là giải pháp phổ biến nhờ khả năng triển khai nhanh và chi phí thấp. FlightPath được nhiều trường đại học nước ngoài áp dụng nhằm số hóa quy trình tư vấn học tập, quản lý chương trình đào tạo và hỗ trợ sinh viên lập kế hoạch học tập cá nhân. Sự tồn tại của những hệ thống này phản ánh nhu cầu tự động hóa quy trình cố vấn ngày càng tăng cao, đặc biệt trong bối cảnh số lượng sinh viên lớn khiến việc quản lý thủ công trở nên kém hiệu quả. Tuy nhiên, kèm với ưu điểm mở là những thách thức trong việc tối ưu hóa và tùy biến theo yêu cầu đặc thù của từng trường. [7]

### Phân tích về nền tảng công nghệ và kỹ thuật xử lý của FlightPath

FlightPath là hệ thống phát triển bằng PHP, MySQL và chạy trên nền web server truyền thống như Apache.

Ưu điểm: Là mã nguồn mở nên cho phép tùy chỉnh sâu giao diện, chức năng và quy trình nghiệp vụ. Cộng đồng phát triển hỗ trợ mạnh. Hệ thống tích hợp sẵn các module lập kế hoạch học tập (Degree Audit), gợi ý môn học, quản lý hồ sơ cố vấn và lịch sử trao đổi.

Hạn chế kỹ thuật: Do được xây dựng khá lâu, FlightPath mang đặc trưng của các hệ thống PHP thuần, cấu trúc mã nguồn không thực sự tối ưu và thiếu định hướng module hiện đại. Cơ sở dữ liệu có nhiều bảng liên quan đến chương trình đào tạo của các trường Mỹ, đôi khi không phù hợp trực tiếp với mô hình đào tạo tín chỉ của Việt Nam. Ngoài ra, hệ thống thiếu API hiện đại, gây khó khăn khi tích hợp vào cổng thông tin sinh viên hoặc ứng dụng di động.

### Đánh giá về mô hình nghiệp vụ và dịch vụ

FlightPath được thiết kế xoay quanh mô hình tư vấn học tập cá nhân hóa. Hệ thống cho phép:

- Kiểm tra tiến độ học tập theo từng chuẩn đầu ra hoặc chương trình học.

- Gợi ý môn học dựa trên lịch sử học và môn tiên quyết.

- Lưu vết trao đổi giữa cố vấn và sinh viên.

- Quản lý lịch hẹn tư vấn và thông báo tự động.

Mặc dù mô hình nghiệp vụ tương đối đầy đủ, FlightPath lại thiếu nhiều tính năng hiện đại như: dự báo nguy cơ học vụ (Early Warning System), dashboard phân tích dữ liệu lớn, hoặc tích hợp lịch học, điểm, môn tiên quyết theo yêu cầu từng trường. Những thiếu sót này khiến hệ thống khó đáp ứng các nhu cầu nâng cao trong quản lý cố vấn học tập.

### Đánh giá về giao diện và trải nghiệm người dùng

Giao diện của FlightPath tuy đơn giản và nhẹ, nhưng mang phong cách cũ (legacy UI), ít yếu tố trực quan. Các thao tác tra cứu chương trình học và đăng ký tư vấn vẫn còn rườm rà do bố cục thông tin chưa hợp lý. Hệ thống cũng chưa hỗ trợ tối ưu mobile, khiến trải nghiệm của sinh viên khi truy cập qua điện thoại chưa tốt. Tình trạng tải lại trang liên tục (full page reload) khiến tương tác bị gián đoạn, không phù hợp với tiêu chuẩn giao diện hiện đại vốn đòi hỏi mượt mà và thân thiện.

### Nhận xét chung

FlightPath mang lại nền tảng tốt cho các trường muốn triển khai một hệ thống cố vấn học tập giá rẻ và có tính tùy biến cao. Tuy nhiên hệ thống đang dừng lại ở mức chức năng cơ bản; chưa đáp ứng được kỳ vọng về tối ưu giao diện, tốc độ, khả năng phân tích dữ liệu và tích hợp sâu với các hệ thống khác. Điều này tạo ra khoảng trống lớn cho các hệ thống mới phát triển theo hướng hiện đại, giàu tính năng hơn.

### Nhận xét riêng

Về nền tảng công nghệ: Mã nguồn PHP cũ, cấu trúc còn nặng, khó mở rộng cho các tính năng phức tạp như dashboard phân tích hoặc API mobile.

Về UX: Quy trình tư vấn bị phân tán giữa nhiều màn hình, thiết kế chưa hiện đại, thông tin hiển thị dày đặc gây khó theo dõi tiến độ học tập.

Về bảo mật và khả năng mở rộng: Chưa áp dụng các chuẩn bảo mật mới như OAuth2, JWT. Khả năng mở rộng theo dạng microservices gần như không khả thi nếu không tái cấu trúc.

### Kết luận và hướng tiếp cận của đề tài

Khi tham khảo hệ thống FlightPath – một giải pháp mã nguồn mở được sử dụng để hỗ trợ công tác cố vấn học tập – có thể thấy rằng hệ thống này sở hữu nhiều chức năng cốt lõi hữu ích như đánh giá lộ trình học tập, ghi chú tư vấn, theo dõi tiến độ và quản lý lịch hẹn. Tuy nhiên, FlightPath được xây dựng từ khá lâu với kiến trúc cũ, giao diện thiếu tính hiện đại và chưa phù hợp hoàn toàn với mô hình đào tạo tín chỉ tại Việt Nam. Do đó, hướng tiếp cận hợp lý cho đề tài là kế thừa các ý tưởng chức năng và quy trình nghiệp vụ của FlightPath nhưng tiến hành xây dựng lại hệ thống trên nền tảng Django và MySQL nhằm tối ưu bảo mật, khả năng mở rộng và tiện ích trong phát triển lâu dài. Việc thiết kế lại giao diện bằng HTML, CSS, JavaScript và Bootstrap giúp hệ thống thân thiện và dễ sử dụng hơn, trong khi quá trình triển khai được hỗ trợ bởi Figma để dựng UI/UX, GitHub để quản lý mã nguồn và MySQL Workbench để mô hình hóa cơ sở dữ liệu. Bằng cách tái hiện các chức năng cần thiết của FlightPath trong một kiến trúc hiện đại, hệ thống mới đảm bảo vừa kế thừa được các ưu điểm của giải pháp gốc, vừa phù hợp với thực tế hoạt động cố vấn học tập tại trường đại học trong nước.

# CƠ SỞ LÝ THUYẾT

## Python framework: Django

Python là một ngôn ngữ lập trình bậc cao, thông dịch và đa năng, được phát triển bởi Guido van Rossum vào cuối những năm 1980 và ra mắt lần đầu tiên vào năm 1991. Python nổi bật nhờ cú pháp đơn giản, dễ hiểu và dễ đọc, giúp lập trình viên nhanh chóng nắm bắt và phát triển ứng dụng mà không phải lo lắng nhiều về các chi tiết phức tạp.

Python hỗ trợ nhiều phương pháp lập trình, bao gồm lập trình hướng đối tượng, lập trình hàm và lập trình thủ tục, giúp người lập trình linh hoạt trong việc xây dựng các ứng dụng. Nó được sử dụng rộng rãi trong nhiều lĩnh vực như phát triển web, khoa học dữ liệu, trí tuệ nhân tạo, tự động hóa, phân tích dữ liệu, và nhiều ứng dụng khác.

Một trong những điểm mạnh của Python là thư viện phong phú, với hàng nghìn thư viện và framework có sẵn, giúp tiết kiệm thời gian phát triển và làm cho việc lập trình trở nên dễ dàng hơn. Các thư viện như Django, Flask (cho phát triển web), Pandas, NumPy (cho khoa học dữ liệu), TensorFlow, PyTorch (cho trí tuệ nhân tạo) là những công cụ phổ biến hỗ trợ các nhà phát triển trong nhiều lĩnh vực khác nhau.

Với tính năng dễ học, dễ sử dụng và sự hỗ trợ mạnh mẽ từ cộng đồng, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến và được ưa chuộng nhất trên thế giới.[1]

Python là ngôn ngữ được ứng dụng đa dạng trong các lĩnh vực.

Phát triển web phía máy chủ: Bao gồm những hàm backend phức tạp mà các trang web thực hiện để hiển thị thông tin cho người dùng. Ví dụ: các trang web phải tương tác với cơ sở dữ liệu, giao tiếp với các trang web khác và bảo vệ dữ liệu khi truyền qua mạng. Django và Flask là 2 framework phổ biến hiện nay dành cho các lập trình viên Python để tạo ra các website.

Tự động hóa bằng các tập lệnh Python: Dùng Python để tự động hóa các tác vụ lặp đi lặp lại, như quét dữ liệu với BeautifulSoup, Selenium; tự động hóa hệ thống với os hoặc shutil.

Khoa học máy tính: Trong Python có rất nhiều thư viện quan trọng phục vụ cho ngành khoa học máy tính như: OpenCV cho xử lý ảnh và Machine Learning; Scipy và Numpys cho lĩnh vực toán học, đại số tuyến tính; Pandas cho việc phân tích dữ liệu, …

Lĩnh vực IoT: Python có thể viết được các ứng dụng cho nền tảng nhúng, đồg thời cũng được lựa chọn cho việc xử lý dữ liệu lớn. Vì thế Python là một ngôn ngữ quen thuộc trong lĩnh vực Internet kết nối vạn vật.

Lập trình game: Pygame là một bộ module Python cross-platform được thiết kế để viết game cho cả máy tính và các thiết bị di động

Django là một web framework cấp cao được xây dựng trên ngôn ngữ Python, ra đời nhằm giải quyết bài toán cốt lõi trong nghiên cứu và thực tiễn phát triển phần mềm:

“Làm thế nào để xây dựng các hệ thống web có khả năng mở rộng, bảo trì tốt, đảm bảo tính nhất quán kiến trúc và giảm thiểu lỗi do con người”.

Trong nghiên cứu khoa học máy tính, Django thường được xem là một case study tiêu biểu cho:

- Áp dụng lập trình hướng đối tượng trong hệ thống web.

- Hiện thực hóa các design patterns kinh điển.

Tối ưu hóa vòng đời phát triển phần mềm.

### Kiến trúc MVT

Khác với mô hình MVC (Model-View-Controller) truyền thống, Django sử dụng biến thể MVT:

- Model: Quản lý dữ liệu và cấu trúc cơ sở dữ liệu (tương tác qua ORM).

- Template: Phần giao diện (HTML) hiển thị cho người dùng.

- View: trung tâm xử lý logic, lấy dữ liệu từ Model và đưa vào Template.

Bảng 2.1 Ưu điểm của Django framework

|  |  |
| --- | --- |
| **Ưu điểm** | **Diễn giải** |
| **Hiệu suất cao** | Django được thiết kế để xử lý lượng truy cập lớn một cách hiệu quả, nhờ vào việc tối ưu hóa các quy trình bên trong và sử dụng các cơ sở dữ liệu mạnh mẽ. |
| Bảo mật cao | Django có hệ thống bảo mật mạnh mẽ, bao gồm mã hóa mật khẩu, phòng tránh lỗi cơ bản và xử lý lỗ hổng bảo mật hiệu quả. |
| **Dễ học và sử dụng** | Cú pháp Python rõ ràng, cộng với tài liệu phong phú và cộng đồng lớn, giúp việc học và sử dụng Django trở nên dễ dàng hơn. |
| **Linh hoạt** | Django có thể sử dụng để xây dựng nhiều loại ứng dụng web khác nhau, từ các trang web đơn giản đến các ứng dụng phức tạp. |
| Cộng đồng người dùng | Django có cộng đồng người dùng lớn và tài liệu phong phú, hỗ trợ qua các diễn đàn và website. |

Bảng 2.2 Nhược điểm của Django framework

|  |  |
| --- | --- |
| **Nhược điểm** | **Diễn giải** |
| "Batteries included" quá tải | Việc tích hợp sẵn quá nhiều tính năng khiến hệ thống trở nên nặng nề, dư thừa nếu không sử dụng hết. |
| Không hợp dự án nhỏ | Gây lãng phí tài nguyên và tốn thời gian thiết lập không cần thiết cho những ứng dụng web đơn giản. |
| Quy ước gò bó | Áp đặt các quy tắc thiết kế (Convention over Configuration) khiến nhà phát triển khó linh hoạt theo phong cách riêng. |

### Cấu trúc thư mục của Django

#### Thư mục gốc dự án

**manage.py**

- Công cụ dòng lệnh dùng để quản lý dự án Django.

- Thực hiện các lệnh như chạy server, tạo app, migrate cơ sở dữ liệu, tạo superuser,…

#### Thư mục project (trùng tên với dự án)

**\_\_init\_\_.py**: Giúp Python nhận diện thư mục như một module.

**settings.py**: Chứa toàn bộ cấu hình của hệ thống như cơ sở dữ liệu, middleware, app, bảo mật, ngôn ngữ, timezone,…

**urls.py**: Quản lý định tuyến URL, ánh xạ các đường dẫn tới các view hoặc app tương ứng.

**wsgi.py**: Cấu hình triển khai ứng dụng trên máy chủ theo chuẩn WSGI.

**asgi.py**: Cấu hình triển khai ứng dụng theo chuẩn ASGI, hỗ trợ các tác vụ bất đồng bộ.

#### Thư mục ứng dụng

**models.py**: Định nghĩa các bảng và quan hệ trong cơ sở dữ liệu.

**views.py**: Xử lý logic nghiệp vụ và trả về dữ liệu hoặc giao diện cho người dùng.

**admin.py**: Đăng ký và quản lý các model trong trang quản trị Django Admin.

**apps.py**: Cấu hình thông tin của app.

**tests.py**: Chứa các đoạn mã kiểm thử hệ thống.

**migrations/**: Lưu trữ lịch sử thay đổi cấu trúc cơ sở dữ liệu.

#### Các thư mục mở rộng thường dùng

**templates/**: Chứa các tệp HTML để hiển thị giao diện người dùng.

**static/**: Lưu trữ các tệp tĩnh như CSS, JavaScript, hình ảnh.

**media/**: Lưu trữ các tệp do người dùng tải lên (ảnh, tài liệu,…).

### Những thành phần chính có trong Django

#### Thành phần Views

Trong Django, thuật ngữ "**Views**" (lượt xem) đề cập đến một phần quan trọng trong kiến trúc của framework, cụ thể là trong mẫu MVT (Model-View-Template) mà Django sử dụng (một biến thể của MVC - Model-View-Controller). Mặc dù tên gọi có vẻ đơn giản, "**Views**" đóng vai trò trung tâm trong xử lý logic ứng dụng và tương tác giữa dữ liệu **models** và giao diện người dùng **templates**.

#### URLS

Tệp urls.py thường sẽ nằm trong thư mục dự án và cả ứng dụng, đóng vai trò quan trọng trong việc định tuyến URL của một dự án Django. Nó như một "bản đồ" giúp Django hiểu được khi người dùng truy cập một địa chỉ (URL) cụ thể, thì cần phải thực hiện hành động gì, cụ thể là gọi view (hàm xử lý) nào để trả về nội dung tương ứng. Và phần urlpatterns sẽ là nơi để chứa liên kết chỉ dẫn đến các trang trong website.

Cấu trúc khai báo cơ bản: path(gồm có route, views, name)

Route: Chuỗi ký tự, là đường dẫn (URL path) mà Django sử dụng để khớp yêu cầu từ người dùng. Đây là phần xác định URL mà bạn muốn ánh xạ đến một hàm xử lý (view).(Ví dụ ' ' hoặc ‘home/’ là trang đầu tiên khi truy cập vào web).

Views: Hàm hoặc lớp view, chịu trách nhiệm xử lý logic khi một URL được khớp. Đây là nơi bạn viết mã xử lý dữ liệu, truy vấn cơ sở dữ liệu, và trả về phản hồi HTTP.(Ví dụ views.home)

Name: Chuỗi ký tự, là một tên duy nhất được đặt cho URL pattern. Đây là cách bạn tham chiếu đến URL trong các template, view hoặc các nơi khác trong ứng dụng Django.

Phần include(): Giúp chia nhỏ urlpatterns theo từng ứng dụng hoặc chức năng, giữ file urls.py chính ngắn gọn, dễ quản lý. Mỗi ứng dụng tự quản lý các URL của mình qua urls.py, dễ mở rộng và chỉnh sửa mà không ảnh hưởng đến ứng dụng khác. Cho phép các ứng dụng có thể được sử dụng lại trong các dự án khác chỉ bằng cách include.

#### Models

Trong framework Django, Model giữ một vị trí then chốt trong kiến trúc MVT (Model-View-Template), đóng vai trò như một lớp trừu tượng mạnh mẽ để quản lý và tương tác với dữ liệu được lưu trữ trong cơ sở dữ liệu. Nó không chỉ đơn thuần là một bản sao của cấu trúc bảng biểu mà còn cung cấp một loạt các tính năng giúp đơn giản hóa quá trình phát triển ứng dụng web.

Trong thiết kế cơ sở dữ liệu, mỗi thực thể được biểu diễn bằng một bảng.

Các thuộc tính của thực thể là các cột của bảng, và mỗi cá thể của thực thể là một hàng trong bảng. Hệ thống quản lý cơ sở dữ liệu cung cấp các công cụ để tạo bảng, định nghĩa quan hệ giữa các bảng, và thực hiện các thao tác thêm, sửa, xóa và truy vấn dữ liệu một cách trực quan, giúp đơn giản hóa việc quản lý dữ liệu mà không cần kiến thức chuyên sâu về SQL

Models sẽ giúp bảo trì dữ liệu, tự động hóa các thay đổi, đảm bảo an toàn khi thao tác với cơ sở dữ liệu

#### Hệ thống quản trị (Admin)

Hệ thống Admin của Django là một công cụ quản trị cực kỳ hữu ích và mạnh mẽ, được tích hợp sẵn trong Django framework. Nó cho phép người quản trị (thường là quản trị viên hệ thống hoặc các nhà phát triển) dễ dàng quản lý và thao tác với dữ liệu trong cơ sở dữ liệu thông qua giao diện web trực quan, mà không cần phải sử dụng câu lệnh SQL phức tạp hay công cụ dòng lệnh. Django Admin không chỉ giúp tiết kiệm thời gian mà còn cung cấp một phương pháp trực quan để theo dõi và quản lý các mô hình dữ liệu trong ứng dụng.

Một trong những điểm mạnh của Django Admin là khả năng tùy chỉnh linh hoạt. Người dùng có thể thay đổi cách thức hiển thị dữ liệu, thêm các tính năng như tìm kiếm, lọc, phân trang, và nhiều tính năng khác để phù hợp với các nhu cầu cụ thể của từng dự án. Điều này giúp các quản trị viên có thể dễ dàng kiểm soát và thao tác với các bản ghi trong cơ sở dữ liệu mà không cần phải viết mã phức tạp.

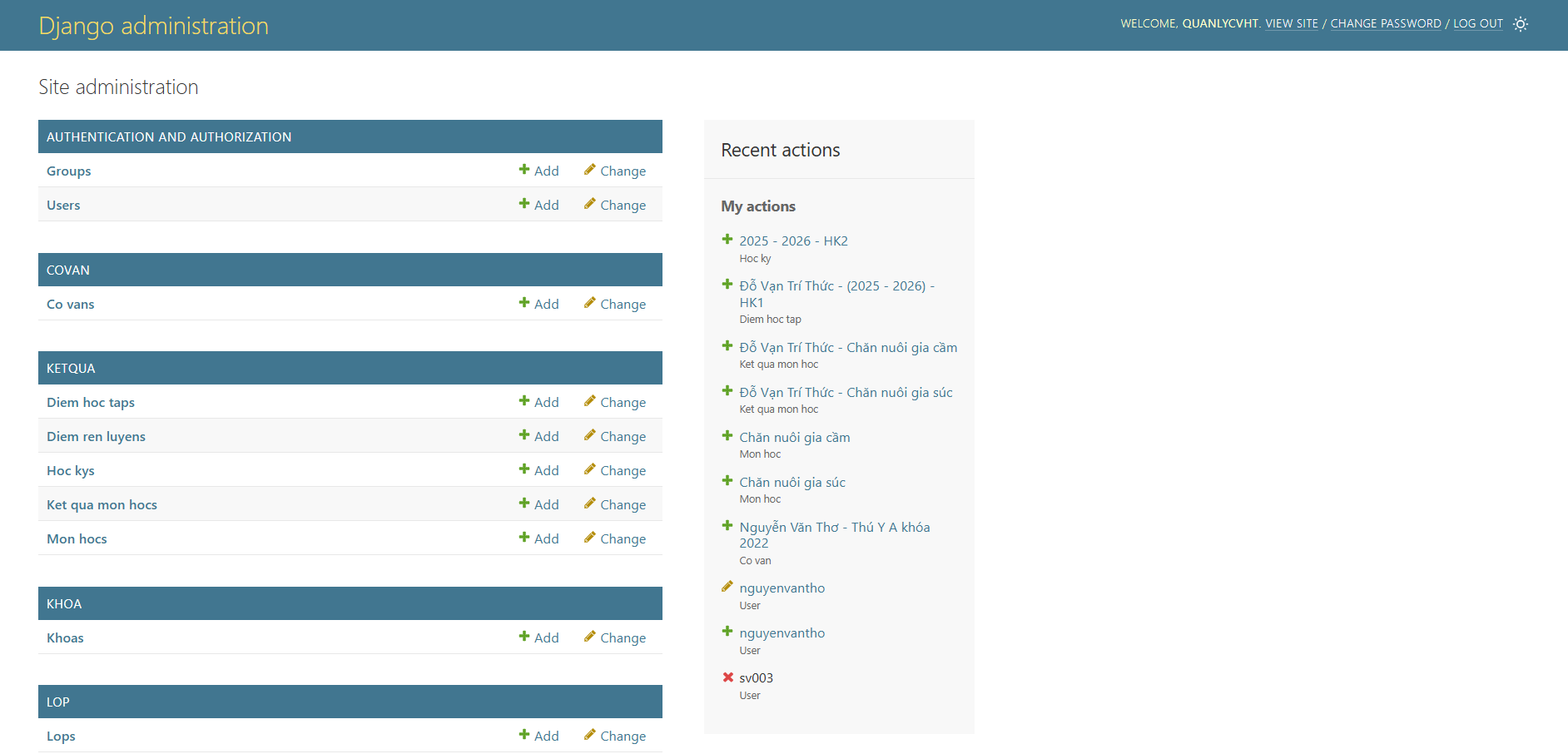
Django Admin là một biểu hiện rõ ràng của mô hình phát triển hướng mô hình (Model-Driven Development). Trong đó, mô hình dữ liệu (Model) không chỉ đóng vai trò lưu trữ thông tin mà còn trở thành trung tâm chi phối việc sinh tự động giao diện quản trị, các thao tác CRUD và các cơ chế kiểm soát dữ liệu. Từ góc nhìn nghiên cứu, đây là một ví dụ tiêu biểu cho việc sử dụng siêu dữ liệu (metadata) nhằm rút ngắn khoảng cách giữa thiết kế khái niệm và hiện thực hệ thống.

Bên cạnh đó, Django Admin hiện thực hóa triệt để nguyên lý DRY (Don’t Repeat Yourself). Thay vì xây dựng thủ công các chức năng quản trị cho từng thực thể, hệ thống cho phép tái sử dụng trực tiếp các định nghĩa trong Model, Form và Validation. Điều này góp phần làm giảm độ phức tạp của mã nguồn, nâng cao khả năng bảo trì và tăng độ tin cậy của hệ thống – những tiêu chí quan trọng trong đánh giá học thuật về chất lượng phần mềm.

Một đóng góp đáng chú ý khác của Django Admin là việc tích hợp sẵn hệ thống xác thực và phân quyền. Cơ chế quản lý người dùng, nhóm và quyền truy cập của Django có thể được xem như một hiện thực chuẩn của mô hình Role-Based Access Control (RBAC). Dưới góc nhìn nghiên cứu hệ thống thông tin, đây là nền tảng quan trọng để đảm bảo tính an toàn, kiểm soát truy cập và phân quyền trách nhiệm trong các hệ thống quản lý quy mô tổ chức, đặc biệt là trong lĩnh vực giáo dục và hành chính.

Django Admin kế thừa trực tiếp các cơ chế kiểm tra dữ liệu (validation), xử lý biểu mẫu và quản lý vòng đời dữ liệu của Django. Việc kiểm soát dữ liệu đầu vào một cách chặt chẽ góp phần đảm bảo tính toàn vẹn và chất lượng dữ liệu, đồng thời giảm thiểu sai sót trong quá trình nhập liệu. Dưới góc độ học thuật, đây là biểu hiện của lập trình phòng thủ (defensive programming) và quản trị dữ liệu có kiểm soát.

Các công cụ hỗ trợ sẵn như migration, logging và ghi nhận lịch sử thao tác trong Django Admin cho phép theo dõi sự tiến hóa của lược đồ cơ sở dữ liệu và truy vết các thay đổi dữ liệu theo thời gian. Điều này mang ý nghĩa quan trọng trong nghiên cứu các hệ thống phát triển dài hạn, nơi tính minh bạch, khả năng kiểm chứng và tái lập kết quả là những yêu cầu thiết yếu.[2] [4]



Hình 2.1 Giao diện trang admin của Django

#### Những bảng có sẵn trong Django

Trong Django, bên cạnh các bảng do người phát triển tự định nghĩa thông qua Model, framework còn cung cấp một tập các bảng hệ thống nhằm hỗ trợ các chức năng nền tảng như xác thực, phân quyền, quản lý phiên và nội dung. Dưới góc độ học thuật, các bảng này có thể được xem là hạ tầng dữ liệu mặc định phục vụ cho vận hành và bảo mật của hệ thống web.

**Nhóm bảng xác thực và phân quyền (*django.contrib.auth*)**

Đây là nhóm bảng quan trọng nhất, phục vụ cho quản lý người dùng và kiểm soát truy cập:

*auth\_user*

Lưu trữ thông tin người dùng: tên đăng nhập, mật khẩu (đã băm), email, trạng thái hoạt động, quyền quản trị.

Trong nghiên cứu hệ thống thông tin, bảng này đại diện cho thực thể người dùng.

*auth\_group*

Lưu trữ các nhóm người dùng, phục vụ cho phân quyền theo vai trò.

*auth\_permission*

Lưu trữ các quyền truy cập (add, change, delete, view) gắn với từng mô hình dữ liệu.

*auth\_user\_groups, auth\_user\_user\_permissions, auth\_group\_permissions*

Các bảng trung gian thể hiện mối quan hệ nhiều–nhiều giữa người dùng, nhóm và quyền.

**Nhóm bảng quản lý nội dung (*django.contrib.contenttypes*)**

*django\_content\_type*

Lưu trữ thông tin về các model trong hệ thống (app label, model name). Đây là nền tảng cho generic relationships cho phép Django xử lý các đối tượng một cách động, thể hiện tư duy siêu mô hình trong thiết kế phần mềm.

**Nhóm bảng quản lý phiên làm việc (*django.contrib.sessions*)**

*django\_session*

Lưu trữ dữ liệu phiên đăng nhập của người dùng, bảng này đóng vai trò quản lý trạng thái, hỗ trợ duy trì phiên làm việc trong kiến trúc HTTP không trạng thái.

**Nhóm bảng hỗ trợ giao diện quản trị (*django.contrib.admin*)**

*django\_admin\_log*

Ghi nhận lịch sử thao tác trong Django Admin: thêm, sửa, xóa dữ liệu. Đây là nền tảng cho audit trail phục vụ nghiên cứu về tính minh bạch và truy vết dữ liệu trong hệ thống quản lý.

**Nhóm bảng migration (*django.contrib.migrations*)**

*django\_migrations*

Lưu trữ lịch sử các migration đã được áp dụng lên cơ sở dữ liệu, bảng này thể hiện khái niệm schema evolution cho phép kiểm soát và tái lập trạng thái CSDL theo thời gian.

#### Tạo dự án

Để bắt đầu dự án với Django python, trước tiên cần cài đặt python và các thư viện cần thiết. [3]

Tải python và chọn phiên bản 3.10 trở lên qua đường dẫn:

https://www.python.org/downloads/

Trong quá trình cài đặt phải tick chọn **“Add Python to PATH”.**

Sau khi cài đặt thành công, kiểm tra lại bằng cách gõ lệnh:

python --version

pip --version

Nếu màn hình hiển thị ra phiên bản đúng với phiên bản download thì python đã sẵn sàng sử dụng.

Tạo môi trường ảo để chạy dự án:

python -m venv venv

Kích hoạt môi trường ảo đối với window:

venv\Scripts\activate

Kích hoạt môi trường ảo đối với Linux / macOS:

source venv/bin/activate

Khi kích hoạt thành công, terminal sẽ có ***(venv)*** phía trước.

Cài đặt Django:

pip install django

Kiểm tra phiên bản Django bằng cách gõ lệnh:

django-admin --version

Cài thư viện kết nối MySQL:

pip install mysqlclient

Cài các thư viện hỗ trợ cơ bản:

pip install pillow python-dotenv

Tạo tài khoản quản trị (Admin):

python manage.py createsuperuser

Sau khi chạy lệnh, nhập username, mail, password để hoàn tất tạo admin.

Để tạo 1 dự án trong Django, ta phải gõ câu lệnh vào terminal như sau: django-admin startproject (tên dự án cần tạo). Ví dụ django-admin startproject my\_project

django-admin startproject my\_project

Tiếp theo, mở thư mục dự án vừa tạo để xem cấu trúc và các thành phần chính.

Trong Django, kiến trúc được khuyến nghị là chia dự án thành các ứng dụng nhỏ hơn, mỗi ứng dụng chịu trách nhiệm cho một phần chức năng cụ thể. Để tạo một ứng dụng mới có tên là myapp thì sử dụng câu lệnh:

python manage.py startapp myapp

Sau khi tạo ứng dụng, bạn cần thêm nó vào **INSTALLED\_APPS** trong tệp settings.py của dự án để Django nhận diện ứng dụng.

INSTALLED\_APPS = [

**'myapp',**

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

]

#### Chạy server dự án Django

Sau khi đã thêm ứng dụng vào trong tệp settings.py, cần phải gõ lệnh:

python manage.py makemigrations

python manage.py migrate

Để cập nhật các thay đổi và cấu hình lại cho website. Tiếp đó, bạn phải gõ lệnh:

python manage.py runserver

Để chạy server, khi đó server sẽ để địa chỉ trong terminal, Ctrl + click vào:

http://127.0.0.1:8000/

Khi đã truy cập vào địa chỉ http://127.0.0.1:8000/, nếu thấy thông báo **The install worked successfully! Congratulations!**, thì đã thành công. Đến đây dự án đang sẵn sàng để phát triển.

**A screenshot of a computer

Description automatically generated**

Hình 2.2 Giao diện trang django ban đầu

Cuối cùng là cần triển khai định nghĩa của các models trong *models.py*, tạo các views để xử lý logic, kết nối giữa views và urls trong urls.py và thiết kế giao diện bằng cách tạo file templates.

## Hệ quản trị cơ sở dữ liệu MySQL

### Khái quát về MySQL

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (Relational Database Management System - RDBMS) dựa trên ngôn ngữ truy vấn có cấu trúc (SQL) được phát triển, phân phối và hỗ trợ bởi tập đoàn Oracle. MySQL chạy trên hầu hết tất cả các nền tảng, bao gồm cả Linux , UNIX và Windows. MySQL thường được kết hợp với các ứng dụng web.

Để quản lý cơ sở dữ liệu của MySQL trong đề tài này, em sử dụng công cụ MySQL Workbench. Công cụ này sẽ cung cấp giao diện người dùng trực quan, hỗ trợ người dùng tương tác với hệ quản trị cơ sở dữ liệu MySQL một cách mạnh mẽ. Nó cung cấp các công cụ để thiết kế cấu trúc dữ liệu, quản lý các đối tượng cơ sở dữ liệu, tối ưu hiệu suất truy vấn và thực thi các tác vụ quản trị một cách thuận tiện và hiệu quả, thay vì phải sử dụng dòng lệnh phức tạp.

### Ưu và nhược điểm của MySQL

**Ưu điểm:**

Bảng 2.3 Ưu điểm của MySQL

|  |  |
| --- | --- |
| **Ưu điểm** | **Diễn giải** |
| **Dễ dàng sử dụng** | Là hệ thống khá dễ sử dụng, khiến nó trở thành sự chọn lựa khá phổ biến dành cho các nhà phát triển ở tất cả cấp độ kỹ năng. Sử dụng ngôn ngữ truy vấn đơn giản sẽ giúp dễ dàng truy xuất và thao tác dữ liệu |
| **Khả năng tương thích rộng** | MySQL tương thích với nhiều nền tảng, bao gồm Windows, Linux và macOS. Điều này có nghĩa là nó có thể được sử dụng trên nhiều loại thiết bị, từ máy tính để bàn đến thiết bị di động |
| **Hiệu suất cao** | Được thiết kế cho hiệu suất cao, có nghĩa là nó có thể xử lý lượng lớn dữ liệu và lưu lượng truy cập mà không bị chậm. Nó sử dụng các kỹ thuật lập chỉ mục nâng cao và cơ chế lưu vào bộ nhớ đệm để tối ưu hóa hiệu suất |
| **Các tính năng bảo mật mạnh mẽ** | Cung cấp các tính năng bảo mật mạnh mẽ, bao gồm xác thực người dùng, mã hóa và kiểm soát truy cập. Điều này có nghĩa là dữ liệu nhạy cảm có thể được bảo vệ khỏi truy cập trái phép, giúp ngăn chặn vi phạm dữ liệu và các sự cố bảo mật khác |
| **Các tính năng bảo mật mạnh mẽ** | Có một cộng đồng lớn và tích cực bao gồm các nhà phát triển và người dùng, những người đóng góp cho sự phát triển của nó và cung cấp hỗ trợ cho những người khác sử dụng nó |

**Nhược điểm:**

Bảng 2.4 Nhược điểm của MySQL

|  |  |
| --- | --- |
| **Nhược điểm** | **Diễn giải** |
| **Hạn chế dung lượng** | Trong trường hợp số lượng bản ghi ngày càng gia tăng, quá trình truy xuất dữ liệu sẽ trở nên khó khăn. Để giải quyết vấn đề này, bạn có thể áp dụng nhiều biện pháp như tạo cache MySQL hoặc phân tải cơ sở dữ liệu ra nhiều máy chủ |
| **Độ tin cậy thấp** | Một nhược điểm khác của MySQL là trong quá trình thực hiện các chức năng cụ thể như kiểm toán, giao dịch và quản lý tài liệu tham khảo, MySQL có thể trở nên kém tin cậy hơn một số hệ quản trị cơ sở dữ liệu quan hệ khác |
| **Giới hạn chức năng** | MySQL không thiết kế để thực hiện toàn bộ các chức năng và nó đi kèm với những hạn chế về chức năng mà một số ứng dụng có thể cần |

### Kết nối MySQL vào Django

Để kết nối MySQL với Django, trước tiên ta cần phải cài đặt MySQL client và MySQL server nếu chưa có, để Django có thể giao tiếp với MySQL.

Để cài đặt ta gõ lệnh như sau: **pip install mysqlclient** và **pip install PyMySQL**.

Sau đó, tải MySQL Workbench và MySQL Installer Web Community. Cả 2 phải cùng một phiên bản ( ví dụ 8.0.40 ) hoặc chênh lệch nhau 1 đến 2 phiên bản. Khi đã cài thành công, tạo một kết nối tới MySQL. [5] [6]



Hình 2.3 Giao diện kết nối cơ sở dữ lệu của MySQL Workbench

Để thiết lập kết nối giữa ứng dụng Django và cơ sở dữ liệu MySQL, bước đầu tiên là tạo một cơ sở dữ liệu trên máy chủ MySQL. Bạn có thể thực hiện việc này bằng lệnh **CREATE DATABASE (tên database);**. Ví dụ, để tạo một cơ sở dữ liệu có tên mydatabase, bạn sẽ sử dụng lệnh:

CREATE DATABASE ql\_cvht;

Sau khi cơ sở dữ liệu được tạo, bạn cần cấu hình Django để kết nối với nó. Điều này được thực hiện bằng cách chỉnh sửa tệp **settings.py** trong thư mục dự án Django của bạn. Trong tệp này, bạn sẽ cần cung cấp thông tin kết nối đến cơ sở dữ liệu MySQL vừa tạo.

DATABASES = {

'default': {

'ENGINE': 'django.db.backends.mysql',

'NAME': 'cvht\_db',

'USER': 'root',

'PASSWORD': 'Nhanh1234@',

'HOST': '127.0.0.1',

'PORT': '3307',

'OPTIONS': {

'init\_command': "SET sql\_mode='STRICT\_TRANS\_TABLES'",

},

}

}

Để hoàn tất, hãy chạy lệnh migrate để áp dụng các thay đổi vào cơ sở dữ liệu và sau đó chạy runserver. Nếu quá trình runserver không gặp lỗi nào trong terminal, điều đó xác nhận kết nối đến cơ sở dữ liệu đã thành công.

## Cài đặt và hiện thực hóa đề tài

### Khảo sát bài toán và xác định yêu cầu hệ thống

Trong quá trình thực hiện đề tài cá nhân “Xây dựng Hệ thống Quản lý Hoạt động Cố vấn Học tập”, tiến hành tìm hiểu thực tế công tác cố vấn học tập tại trường đại học. Qua đó nhận thấy việc quản lý sinh viên, theo dõi kết quả học tập và lịch tư vấn chủ yếu thực hiện thủ công, gây khó khăn trong việc tổng hợp và theo dõi thông tin.

Từ thực tế trên, xác định các yêu cầu chính của hệ thống gồm: quản lý thông tin sinh viên, quản lý cố vấn học tập, phân công cố vấn, theo dõi lịch và nội dung tư vấn, phân quyền người dùng. Đây là cơ sở để tiến hành xây dựng hệ thống phù hợp với bài toán thực tế.

### Lựa chọn công nghệ và mô hình triển khai

Dựa trên yêu cầu đề tài và kiến thức đã học, tôi lựa chọn Django (Python) làm framework backend do hỗ trợ tốt cho việc xây dựng ứng dụng web và quản lý cơ sở dữ liệu. Phần giao diện được xây dựng bằng HTML, CSS và Bootstrap. Docker được sử dụng để đóng gói ứng dụng nhằm đảm bảo tính nhất quán môi trường. Cuối cùng, Render được chọn làm nền tảng triển khai hệ thống lên môi trường Internet.

### Chuẩn bị môi trường và tạo dự án Django

Cài đặt Python và tạo môi trường ảo để quản lý các thư viện của dự án:

python -m venv venv

source venv/bin/activate # Linux/Mac

venv\Scripts\activate # Windows

Cài đặt Django:

pip install django

Tạo dự án Django cho hệ thống:

django-admin startproject ql\_cvht

cd ql\_cvht

### Tạo các ứng dụng chức năng

Dựa trên nghiệp vụ của hệ thống, tiến hành tạo các app tương ứng với từng chức năng:

python manage.py startapp sinhvien

python manage.py startapp covan

python manage.py startapp lop

Sau khi tạo app, tiến hành khai báo các app này trong file **settings.py**:

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

'sinhvien',

'covanhoctap',

'lop',

]

### Xây dựng mô hình dữ liệu (Model)

Ở bước này, sử dụng Django ORM để xây dựng các model đại diện cho dữ liệu trong hệ thống. Ví dụ, model Sinh viên được xây dựng như sau:

from django.db import models

class SinhVien(models.Model):

ma\_sv = models.CharField(max\_length=10, primary\_key=True)

ho\_ten = models.CharField(max\_length=100)

email = models.EmailField()

lop = models.CharField(max\_length=50)

def \_\_str\_\_(self):

return self.ho\_ten

Sau khi định nghĩa model, tiến hành migrate để tạo bảng trong cơ sở dữ liệu:

python manage.py makemigrations

python manage.py migrate

### Hiện thực hóa chức năng nghiệp vụ

Tiến hành xây dựng các view xử lý nghiệp vụ như hiển thị danh sách sinh viên, thêm mới sinh viên và phân công cố vấn học tập. Ví dụ một view hiển thị danh sách sinh viên:

from django.shortcuts import render

from .models import SinhVien

def danh\_sach\_sinh\_vien(request):

ds\_sinh\_vien = SinhVien.objects.all()

return render(request, 'sinhvien/danh\_sach.html', {

'ds\_sinh\_vien': ds\_sinh\_vien

})

### Xây dựng giao diện người dùng

Phần giao diện được xây dựng bằng HTML kết hợp Django Template Language. Ví dụ template hiển thị danh sách sinh viên:

<table class="table">

<tr>

<th>Mã SV</th>

<th>Họ tên</th>

<th>Email</th>

</tr>

{% for sv in ds\_sinh\_vien %}

<tr>

<td>{{ sv.ma\_sv }}</td>

<td>{{ sv.ho\_ten }}</td>

<td>{{ sv.email }}</td>

</tr>

{% endfor %}

</table>

### Chuẩn bị các file để triển khai Docker

Để đóng gói ứng dụng, tạo file requirements.txt:

Django>=4.2

gunicorn

Tiếp theo, tạo file Dockerfile:

FROM python:3.11-slim

WORKDIR /app

COPY requirements.txt .

RUN pip install --no-cache-dir -r requirements.txt

COPY . .

CMD ["gunicorn", "cvht\_project.wsgi:application", "--bind", "0.0.0.0:8000"]

### Build và chạy thử hệ thống bằng Docker

Sau khi hoàn tất cấu hình, tiến hành build image và chạy container để kiểm thử:

docker build -t ql\_cvht.

docker run -p 8000:8000 ql\_cvht

### Đưa mã nguồn lên GitHub

Sau khi hệ thống hoạt động ổn định,khởi tạo repository Git và đẩy mã nguồn lên GitHub:

git init

git add .

git commit -m "Initial commit CVHT system"

git branch -M main

git remote add origin https://github.com/username/cvht\_project.git

git push -u origin main

## Nền tảng Docker

### Docker là gì?

Docker là một nền tảng mã nguồn mở hỗ trợ đóng gói, triển khai và vận hành ứng dụng trong các môi trường cô lập gọi là container. Mỗi container chứa đầy đủ các thành phần cần thiết để ứng dụng hoạt động như mã nguồn, thư viện, biến môi trường và cấu hình hệ thống. Nhờ đó, ứng dụng có thể chạy ổn định và đồng nhất trên nhiều môi trường khác nhau như máy cá nhân, máy chủ hoặc nền tảng đám mây.

### Nguyên lý hoạt động của Docker

Docker hoạt động dựa trên cơ chế ảo hóa ở mức hệ điều hành, sử dụng chung nhân (kernel) của hệ điều hành chủ nhưng vẫn đảm bảo sự cách ly giữa các container. So với máy ảo truyền thống, container có kích thước nhỏ gọn hơn, khởi động nhanh hơn và tiêu tốn ít tài nguyên hệ thống hơn.

### Lợi ích của Docker

Docker mang lại nhiều lợi ích quan trọng trong phát triển phần mềm, bao gồm:

- Tính nhất quán môi trường: Giảm lỗi “chạy được trên máy tôi nhưng không chạy trên máy khác”.

- Triển khai nhanh chóng: Ứng dụng có thể được khởi chạy hoặc dừng chỉ trong vài giây.

- Dễ dàng mở rộng và bảo trì: Thuận lợi cho việc nâng cấp, mở rộng hệ thống.

- Hỗ trợ làm việc nhóm: Giúp các thành viên sử dụng chung một môi trường phát triển thống nhất.

# HIỆN THỰC HÓA NGHIÊN CỨU

## Mô tả hệ thống

Cơ sở giáo dục hiện nay đang thực hiện công tác quản lý sinh viên chủ yếu theo phương thức thủ công và trực tiếp tại văn phòng khoa, phòng cố vấn học tập. Tuy nhiên, trước sự phát triển mạnh mẽ của công nghệ thông tin và nhu cầu quản lý, theo dõi sinh viên ngày càng tăng, trường quyết định phát triển một hệ thống quản lý hoạt động cố vấn học tập trực tuyến, giúp số hóa các hoạt động quản lý, theo dõi và tư vấn học tập.

Mục tiêu chính của hệ thống là xây dựng một nền tảng trực tuyến chuyên nghiệp, giúp các cố vấn học tập, giảng viên và quản lý khoa có thể dễ dàng tiếp cận thông tin, theo dõi tiến độ học tập của sinh viên, quản lý lớp học, đồng thời thực hiện các hoạt động tư vấn một cách thuận tiện và hiệu quả.

Một trong những điểm khác biệt và cũng là trọng tâm công nghệ của hệ thống là khả năng liên kết trực tiếp giữa lớp học và cố vấn học tập, đồng thời cung cấp các báo cáo thống kê chi tiết về tình hình học tập, tư vấn và hoạt động của sinh viên. Giải pháp này nhằm giải quyết một trong những hạn chế lớn nhất của phương thức quản lý truyền thống, đó là thiếu tính minh bạch, khó theo dõi tiến độ của sinh viên và tốn nhiều thời gian cho việc tổng hợp dữ liệu thủ công.

Ngoài ra, hệ thống còn hỗ trợ phân quyền người dùng theo vai trò, đảm bảo bảo mật thông tin và cho phép bộ phận quản lý tiếp nhận, xử lý các yêu cầu từ sinh viên và cố vấn học tập một cách nhanh chóng, góp phần nâng cao hiệu quả quản lý, đồng thời tạo dựng môi trường học tập chuyên nghiệp, minh bạch và thân thiện với người dùng.

## Đặc tả yêu cầu

### Các chức năng của hệ thống

|  |  |  |
| --- | --- | --- |
| **Nhóm người dùng** | **Mô tả vai trò** | **Chức năng chính** |
| Cố vấn học tập | Giảng viên được phân công phụ trách, hỗ trợ và theo dõi sinh viên trong quá trình học tập. | - Đăng nhập hệ thống.  - Xem danh sách sinh viên được phân công phụ trách. - Theo dõi tiến độ học tập và tư vấn của sinh viên.  - Xem chi tiết thông tin sinh viên (kết quả học tập,lịch sử tư vấn, các vấn đề liên quan).  - Xuất báo cáo, thống kê. |
| Sinh viên | Xem kết quả học tập và rèn luyện, đăng ký hỗ trợ. | - Đăng nhập hệ thống.  - Xem kết quả học tập và lịch sử tư vấn của bản thân. - Liên hệ với cố vấn học tập để nhận hỗ trợ. - Đặt câu hỏi, gửi yêu cầu hoặc yêu cầu tư vấn trực tuyến thông qua hệ thống. |
| Quản trị | Bộ phận quản lý có quyền cao nhất, chịu trách nhiệm vận hành và duy trì hệ thống. | - Quản lý toàn bộ dữ liệu và hoạt động hệ thống. - Thêm, cập nhật, xóa thông tin sinh viên, lớp học, giảng viên và cố vấn học tập. - Phân công và điều chỉnh cố vấn cho các lớp hoặc sinh viên. - Cập nhật dữ liệu sinh viên khi có thay đổi thực tế. - Quản lý các yêu cầu tư vấn. - Theo dõi lịch sử tư vấn của toàn hệ thống. - Xuất báo cáo, thống kê về hoạt động cố vấn và tiến độ học tập sinh viên. - Tiếp nhận và xử lý các yêu cầu hỗ trợ từ người dùng. |

### Yêu cầu phi chức năng

Sinh viên hoặc cố vấn truy cập vào hệ thống thông qua trình duyệt web bằng cách nhập trực tiếp URL hoặc thông qua liên kết từ trang nội bộ của trường. Người dùng cần đăng nhập bằng tài khoản đã được cấp để truy cập các chức năng tương ứng theo vai trò.

Sau khi đăng nhập, cố vấn học tập có thể truy cập danh sách lớp học mà mình phụ trách. Tại đây, cố vấn có thể xem chi tiết danh sách sinh viên, tiến độ học tập, kết quả các môn học, và các vấn đề liên quan đến sinh viên mà cố vấn đó phụ trách. Cố vấn có thể thêm thông tin tư vấn cho từng sinh viên.

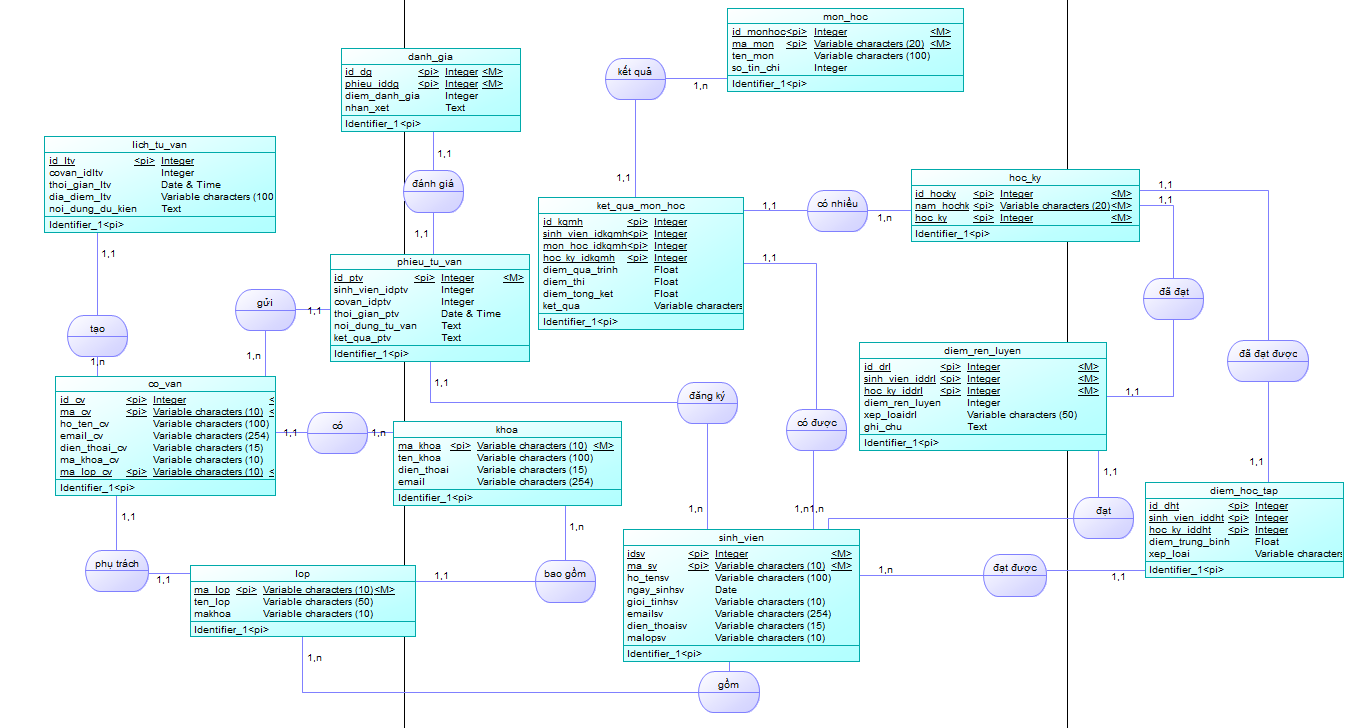
Sinh viên sau khi đăng nhập có thể truy cập thông tin cá nhân, kết quả học tập và lịch sử tư vấn của mình. Sinh viên có thể gửi yêu cầu tư vấn, đặt lịch hẹn gặp cố vấn qua hệ thống để nhận hỗ trợ. Khi có phản hồi từ cố vấn, sinh viên sẽ nhận thông báo tự động và có thể xem lại nội dung tư vấn bất cứ lúc nào.

Bộ phận quản trị (admin) sẽ kiểm tra và duyệt các thông tin cập nhật từ cố vấn, bao gồm đánh giá và kế hoạch tư vấn. Quản trị viên cũng quản lý toàn bộ dữ liệu sinh viên, lớp học và cố vấn để đảm bảo tính chính xác và đồng bộ trong hệ thống. Khi có lớp học mới hoặc thay đổi phân công cố vấn, quản trị sẽ cập nhật kịp thời để hệ thống phản ánh đúng thực tế.

Ngoài ra, hệ thống tự động tổng hợp và hiển thị báo cáo thống kê về tình hình học tập, tiến độ tư vấn và các hoạt động của sinh viên, giúp quản trị viên và khoa nắm bắt kịp thời thông tin để ra quyết định quản lý. Khi có vấn đề hoặc yêu cầu hỗ trợ từ sinh viên hoặc cố vấn, quản trị viên tiếp nhận và xử lý để đảm bảo hoạt động của hệ thống diễn ra liên tục, minh bạch và hiệu quả.

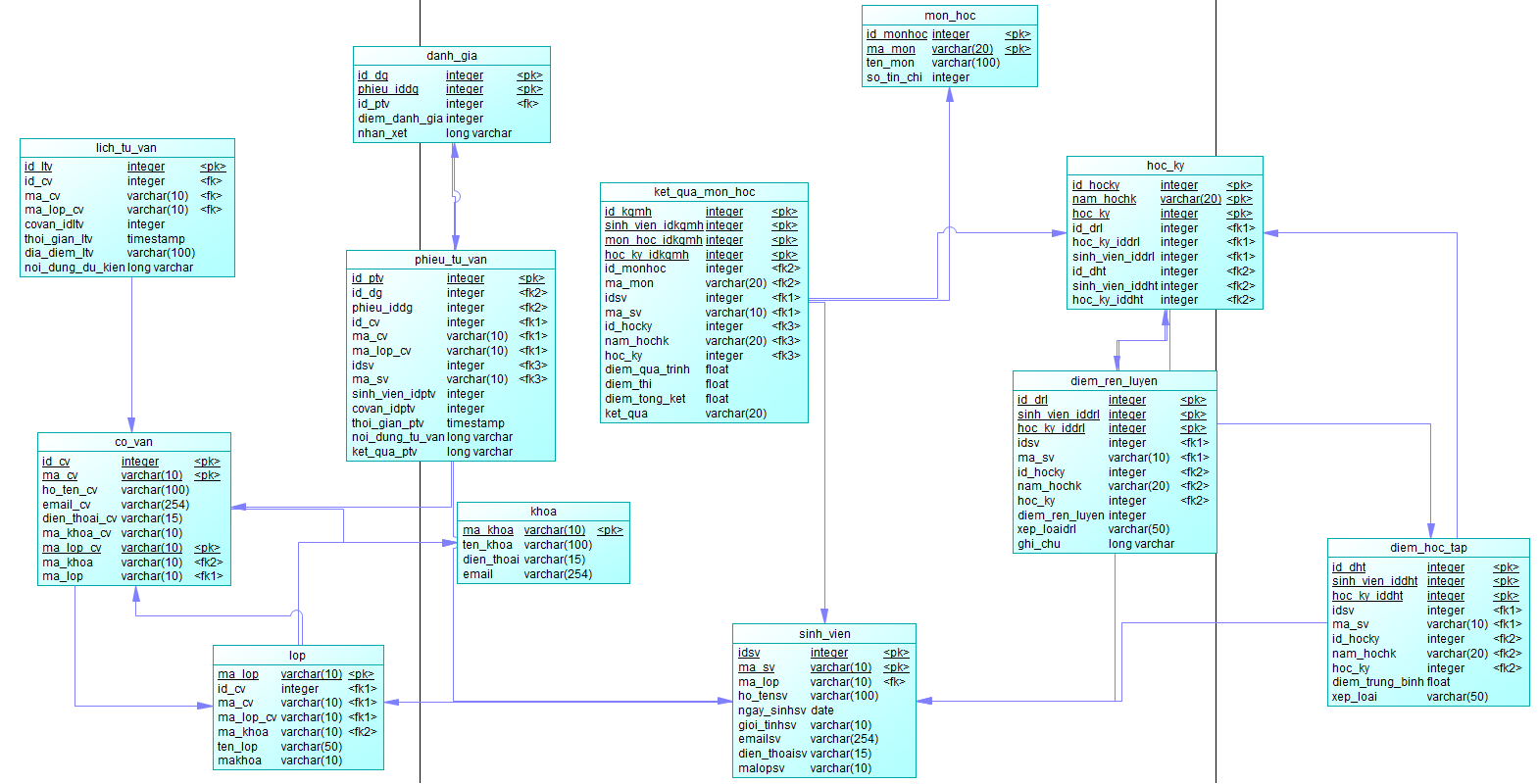
## Thiết kế dữ liệu

### Mô hình dữ liệu mức quan niệm



Hình 3.1 Mô hình dữ liệu mức quan niệm

### Mô hình vật lý



Hình 3.2 Mô hình vật lý

### Mô tả thực thể

Cơ sở dữ liệu được thiết kế như sau:

Bảng Khoa:

Bảng 3.1 Bảng Khoa

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| ma\_khoa | varchar | Mã khoa (khóa chính) |
| ten\_khoa | varchar | Tên khoa |
| dien\_thoai | varchar | Số điện thoại khoa |
| email | varchar | Email khoa |

Bảng lop:

Bảng 3.2 Bảng lop

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| ma\_lop | varchar | Mã lớp (khóa chính) |
| ten\_lop | varchar | Tên lớp |
| ma\_khoa | varchar | Mã khoa (khóa ngoại) |

Bảng co\_van:

Bảng 3.3 Bảng co\_van

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| ma\_cv | varchar | Mã cố vấn (duy nhất) |
| ho\_ten | varchar | Họ tên cố vấn |
| email | varchar | Email cố vấn |
| dien\_thoai | varchar | Số điện thoại |
| ma\_khoa | varchar | Mã khoa (khóa ngoại) |
| ma\_lop | varchar | Mã lớp phụ trách (khóa ngoại) |

Bảng sinh\_vien:

Bảng 3.4 Bảng sinh\_vien

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| ma\_sv | varchar | Mã sinh viên (duy nhất) |
| ho\_ten | varchar | Họ tên sinh viên |
| ngay\_sinh | date | Ngày sinh |
| gioi\_tinh | varchar | Giới tính |
| email | varchar | Email sinh viên |
| dien\_thoai | varchar | Số điện thoại |
| ma\_lop | varchar | Mã lớp (khóa ngoại) |

Bảng hoc\_ky:

Bảng 3.5 Bảng hoc\_ky

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| nam\_hoc | varchar | Năm học |
| hoc\_ky | Int | Học kỳ |

Bảng diem\_hoc\_tap:

Bảng 3.6 Bảng diem\_hoc\_tap

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| sinh\_vien\_id | int | Sinh viên (khóa ngoại) |
| hoc\_ky\_id | int | Học kỳ (khóa ngoại) |
| diem\_trung\_binh | iloat | Điểm trung bình học tập |
| xep\_loai | varchar | Xếp loại học tập |

Bảng diem\_ren\_luyen:

Bảng 3.7 Bảng diem\_ren\_luyen

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| sinh\_vien\_id | int | Sinh viên (khóa ngoại) |
| hoc\_ky\_id | int | Học kỳ (khóa ngoại) |
| diem\_ren\_luyen | int | Điểm rèn luyện |
| xep\_loai | varchar | Xếp loại rèn luyện |
| ghi\_chu | Text | Ghi chú |

Bảng mon\_hoc:

Bảng 3.8 Bảng mon\_hoc

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| ma\_mon | varchar | Mã môn học |
| ten\_mon | Int | Tên môn học |
| so\_tin\_chi | Int | Số tín chỉ |

Bảng ket\_qua\_mon\_hoc:

Bảng 3.9 Bảng ket\_qua\_mon\_hoc

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| sinh\_vien\_id | int | Sinh viên (khóa ngoại) |
| mon\_hoc\_id | int | Môn học (khóa ngoại) |
| hoc\_ky\_id | int | Học kỳ (khóa ngoại) |
| diem\_qua\_trinh | float | Điểm quá trình |
| diem\_thi | float | Điểm thi |
| diem\_tong\_ket | float | Điểm tổng kết |
| ket\_qua | varchar | Kết quả môn học |

Bảng lich\_tu\_van:

Bảng 3.10 Bảng lich\_tu\_van

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| covan\_id | int | Mã Cố vấn (khóa ngoại) |
| thoi\_gian | datetime | Thời gian tư vấn |
| noi\_dung\_tu\_van | text | Nội dung tư vấn |
| ket\_qua | text | Kết quả |

Bảng phieu\_tu\_van:

Bảng 3.11 Bảng phieu\_tu\_van

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| sinhvien\_id | int | Mã Sinh viên (khóa ngoại) |
| covan\_id | int | Mã Cố vấn (khóa ngoại) |
| thoi\_gian | datetime | Thời gian tư vấn |
| noi\_dung\_tu\_van | text | Nội dung tư vấn |
| ket\_qua | text | Kết quả |

Bảng danh\_gia:

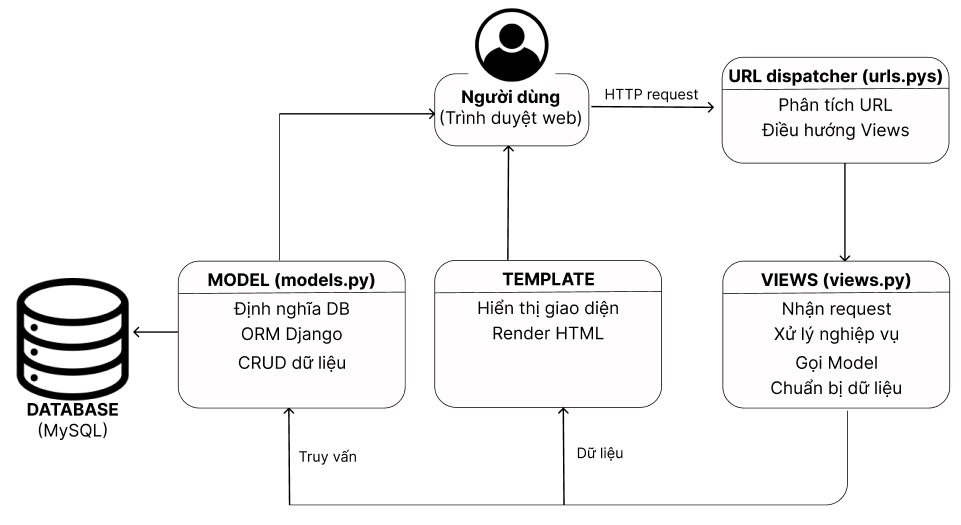
Bảng 3.12 Bảng danh\_gia

|  |  |  |
| --- | --- | --- |
| **Cột** | **Kiểu Dữ Liệu** | **Mô Tả** |
| id | int | Khóa chính tự tăng |
| phieu\_id | int | Phiếu tư vấn (khóa ngoại) |
| diem\_danh\_gia | int | Điểm đánh giá |
| nhan\_xet | text | Nhận xét |

## Phân tích và thiết kế hệ thống

### Sơ đồ kiến trúc tổng thể

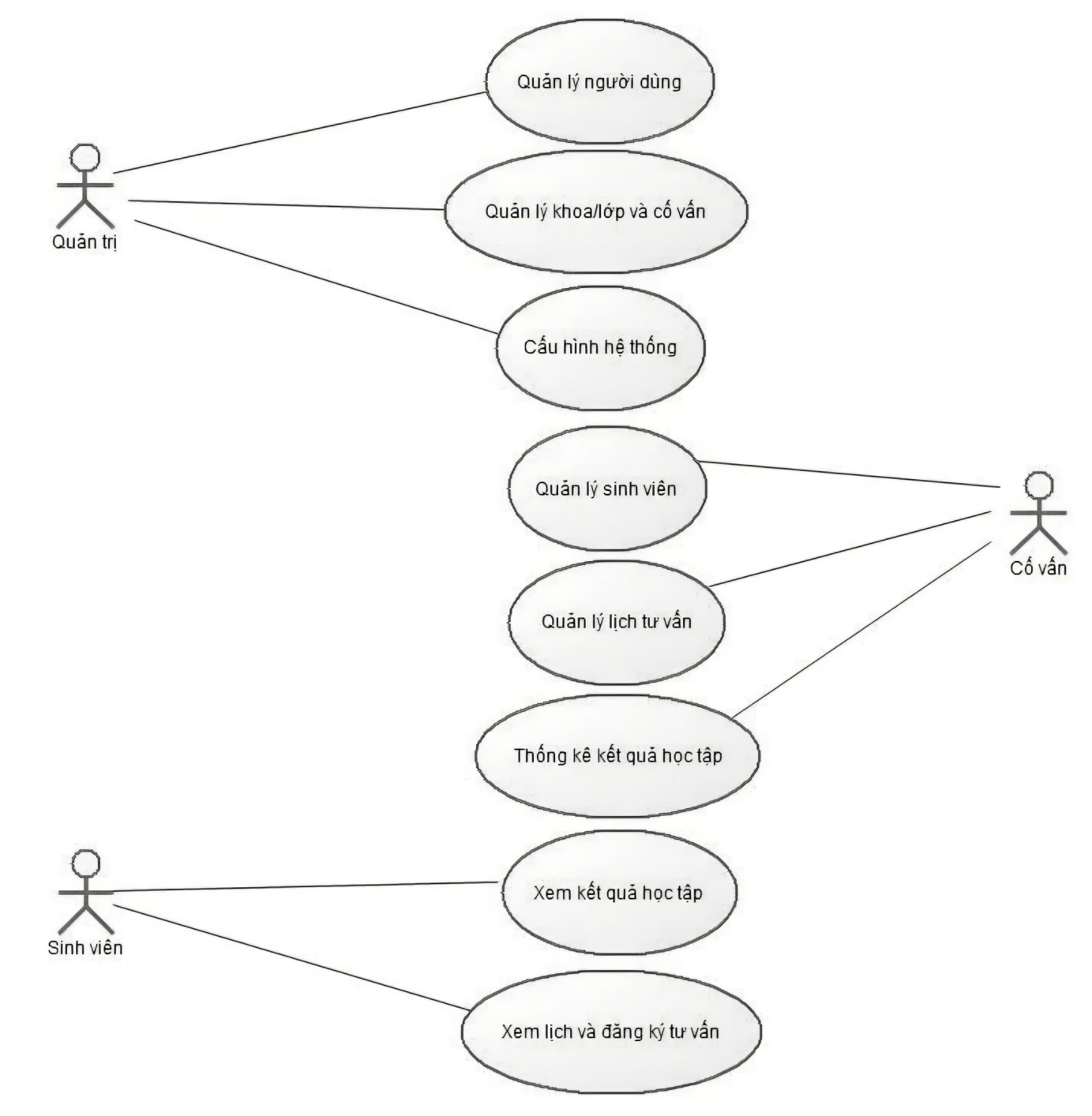
Hệ thống quản lý hoạt động cố vấn học tập được thiết kế theo kiến trúc web nhiều lớp, dựa trên mô hình MVT (Model – View – Template) của Django, nhằm đảm bảo tính phân tách rõ ràng giữa dữ liệu, xử lý nghiệp vụ và trình bày giao diện, đồng thời nâng cao khả năng mở rộng, bảo trì và vận hành lâu dài.



Hình 3.3 Sơ đồ kiến trúc tổng thể hệ thống

### Usecase tổng quát

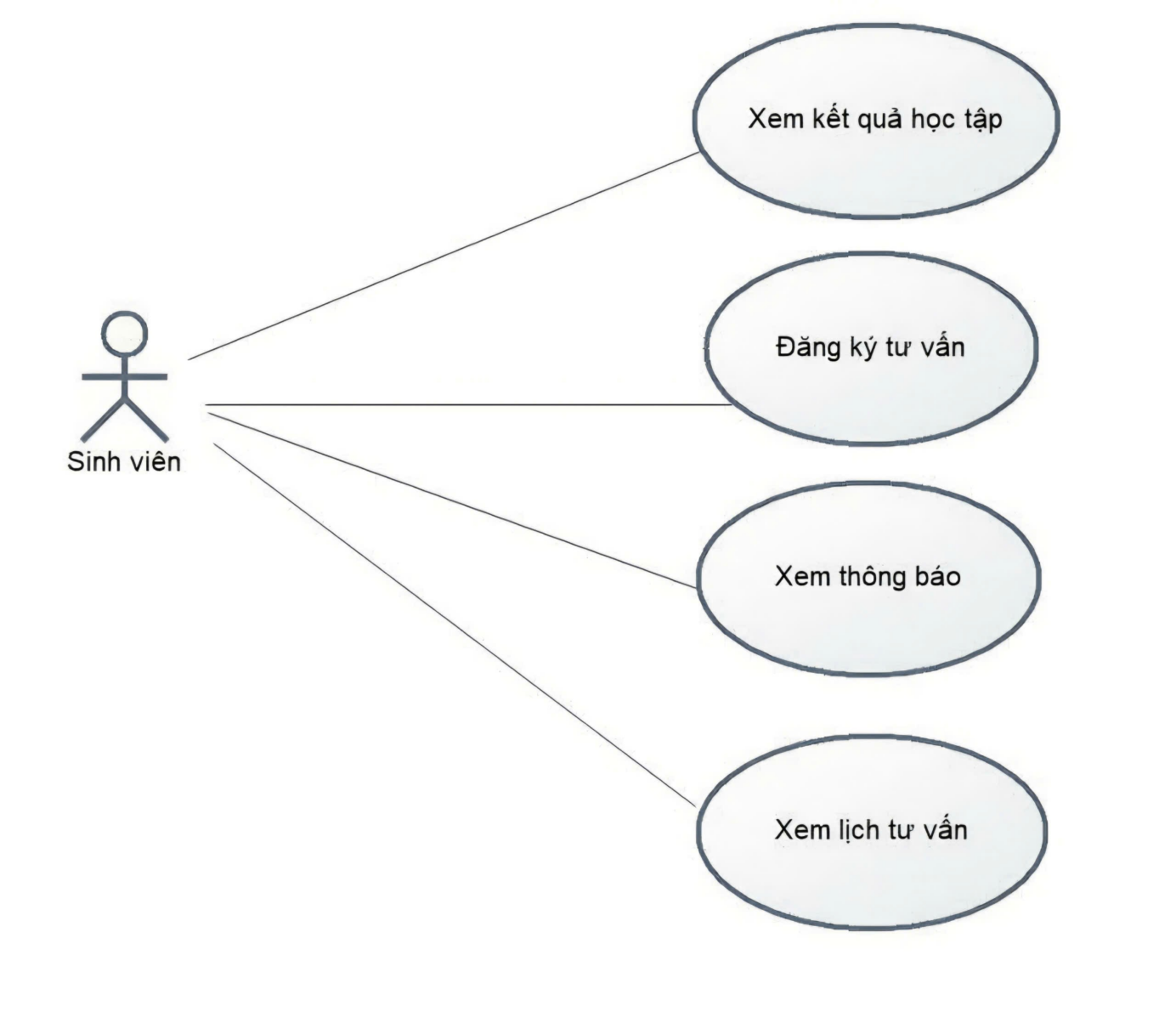
Sơ đồ use case tổng quát mô tả toàn bộ các tương tác chính giữa người dùng và hệ thống quản lý cố vấn học tập. Hệ thống bao gồm ba tác nhân chính là quản trị viên, cố vấn học tập và sinh viên. Quản trị viên chịu trách nhiệm quản lý dữ liệu và cấu hình hệ thống. Cố vấn học tập thực hiện các chức năng liên quan đến theo dõi sinh viên, quản lý lịch tư vấn và gửi thông báo. Sinh viên tương tác với hệ thống để xem kết quả học tập, đăng ký tư vấn và theo dõi thông tin liên quan. Sơ đồ thể hiện rõ phạm vi chức năng của từng nhóm người dùng và mối quan hệ giữa các tác nhân với hệ thống.



Hình 3.3 Sơ đồ tổng quát hệ thống

### Biểu đồ Use Case tác nhân Sinh viên

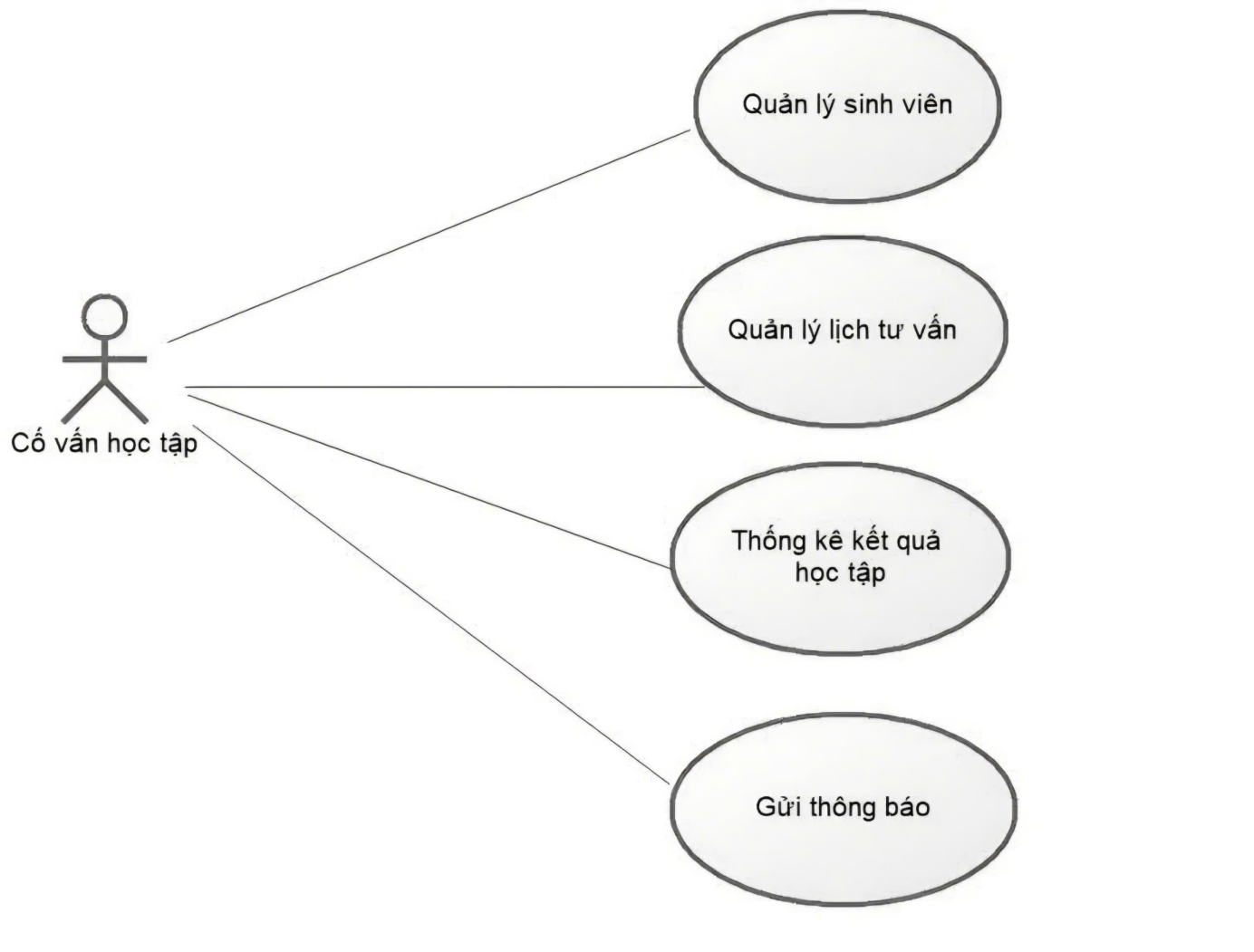
Sinh viên có thể đăng nhập vào hệ thống để xem thông tin cá nhân, theo dõi kết quả học tập và rèn luyện theo từng học kỳ. Ngoài ra, sinh viên còn có thể đăng ký hoặc tham gia các buổi tư vấn, cũng như gửi yêu cầu hỗ trợ học tập khi cần thiết. Hệ thống cho phép sinh viên theo dõi các thông báo và các nội dung liên quan đến quá trình học tập mà không cần thực hiện các thao tác phức tạp. Những chức năng này giúp sinh viên chủ động trong việc quản lý quá trình học tập, tăng cường sự kết nối với cố vấn học tập và nâng cao hiệu quả hỗ trợ học tập trong suốt thời gian học tại trường.



Hình 3.3 Biểu đồ usecase tác nhân sinh viên

### Biểu đồ Use Case tác nhân Cố vấn học tập

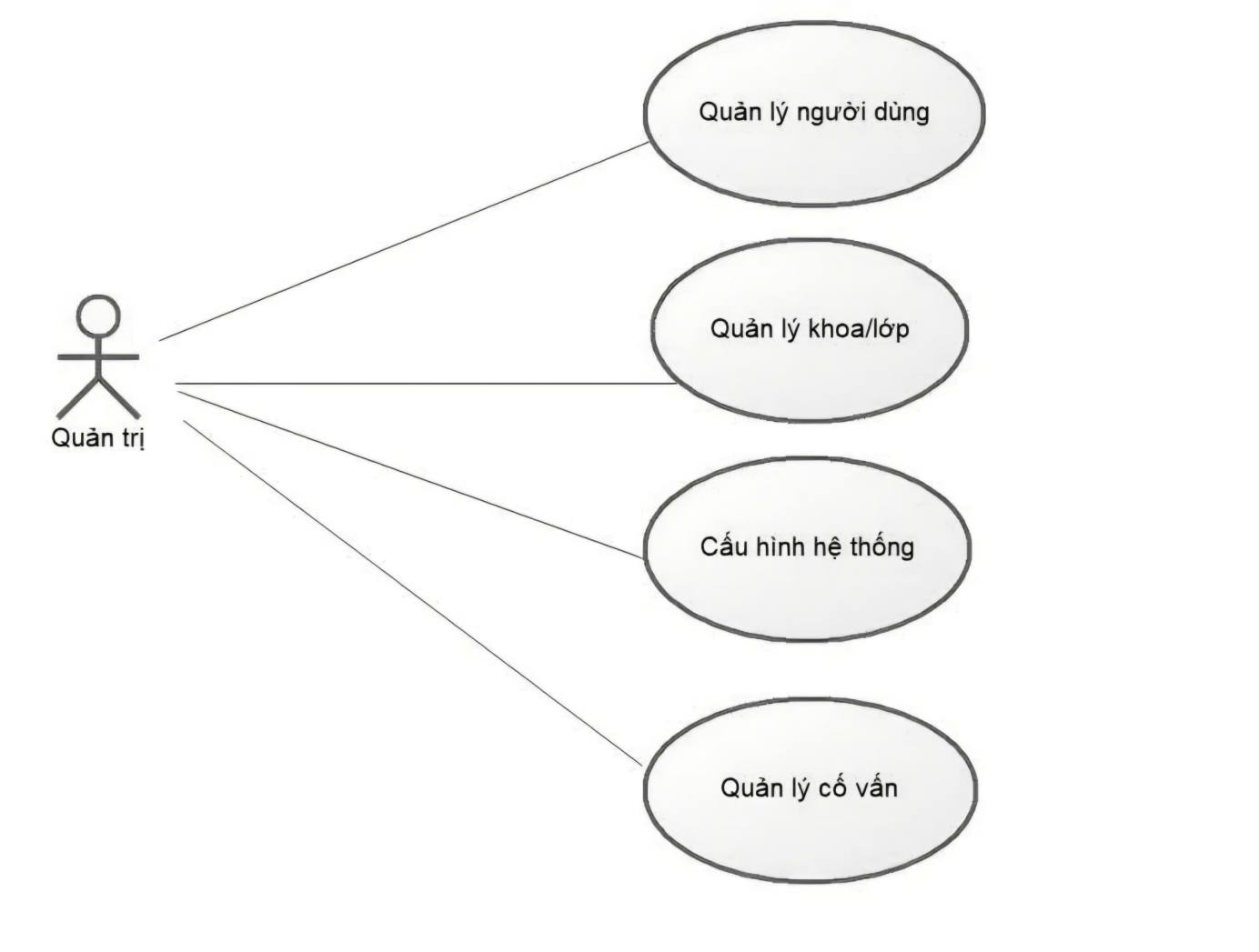
Cố vấn học tập là người dùng đã được hệ thống cấp tài khoản và phân quyền tương ứng. Sau khi đăng nhập, cố vấn có thể xem danh sách sinh viên thuộc lớp mình phụ trách, theo dõi tình hình học tập, tiếp nhận và phản hồi các yêu cầu hỗ trợ từ sinh viên. Ngoài ra, cố vấn còn có thể quản lý lịch tư vấn, cập nhật nội dung tư vấn, gửi thông báo đến sinh viên và xem các thống kê liên quan đến kết quả học tập. Những chức năng này giúp cố vấn theo dõi sát sao quá trình học tập của sinh viên, kịp thời hỗ trợ và nâng cao hiệu quả công tác cố vấn học tập.



Hình 3.4 Biểu đồ usecase tác nhân Cố vấn học tập

### Biểu đồ Use Case tác nhân quản trị

Quản trị viên là người chịu trách nhiệm quản lý và vận hành toàn bộ hệ thống. Admin có thể quản lý thông tin người dùng như sinh viên, cố vấn và phân quyền truy cập cho từng đối tượng. Bên cạnh đó, quản trị viên còn có thể quản lý dữ liệu khoa, khóa, lớp, cấu hình các chức năng hệ thống và theo dõi các thống kê tổng hợp về hoạt động cố vấn học tập. Các chức năng này giúp đảm bảo hệ thống hoạt động ổn định, dữ liệu được quản lý tập trung và hỗ trợ hiệu quả cho công tác quản lý đào tạo của nhà trường.

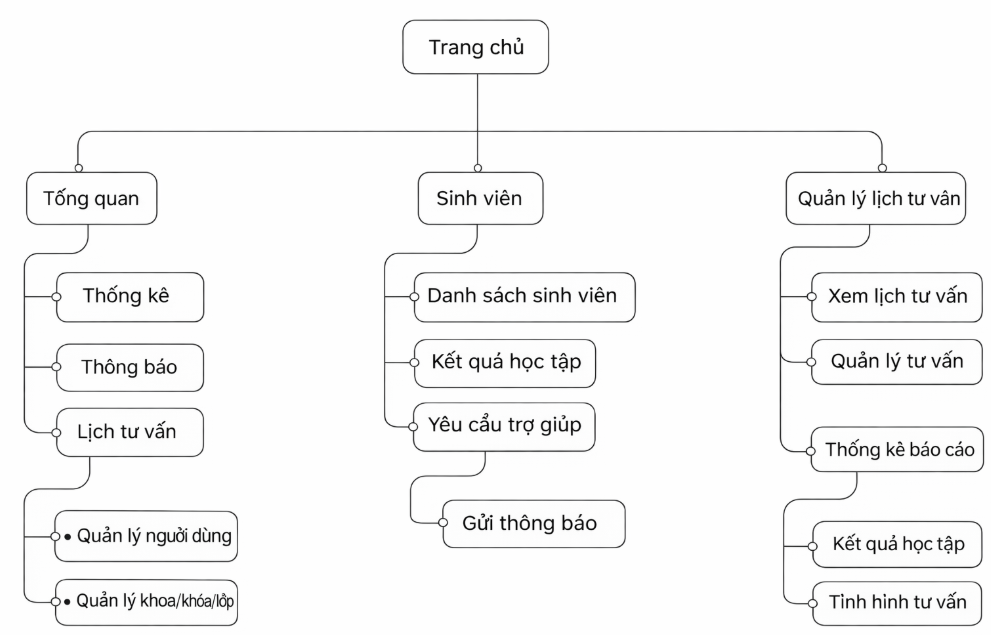


Hình 3.5 Biểu đồ usecase tác nhân Quản trị

## Thiết kế giao diện

### Sơ đồ tổ chức thông tin

Sơ đồ tổ chức thông tin thể hiện các luồng chức năng của hệ thống.



Hình 3.6 Sơ đồ tổ chức thông tin

### Giao diện đăng nhập

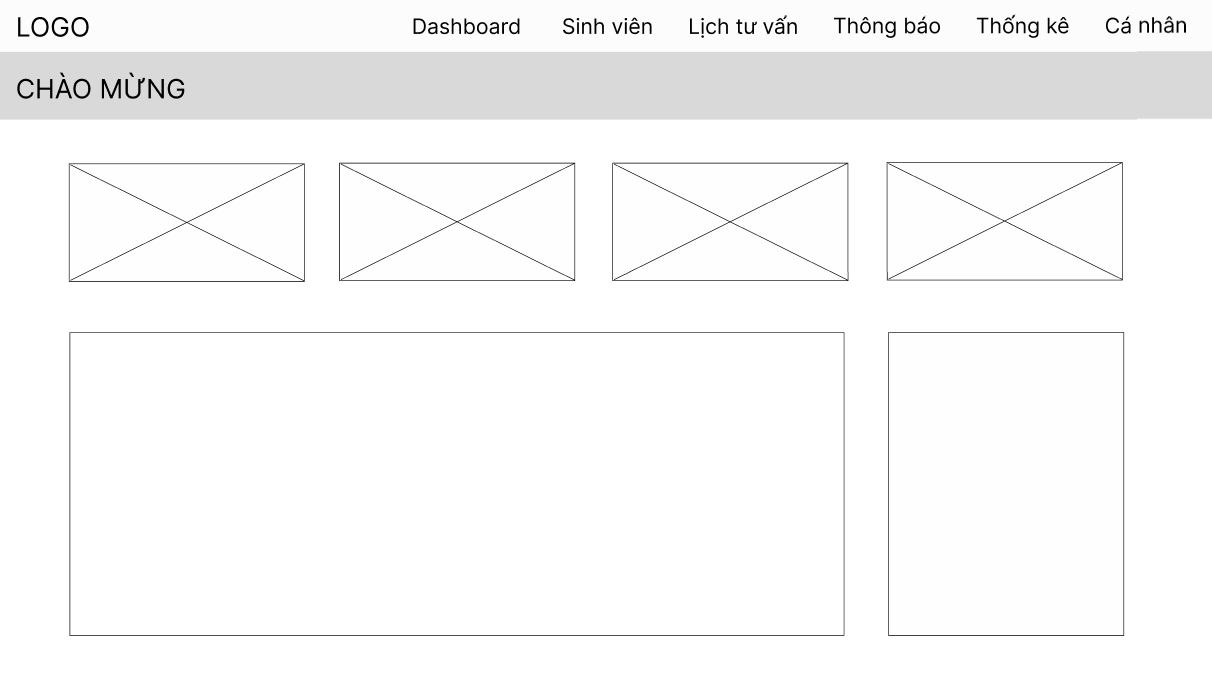
Trang đăng nhập hiển thị logo của Trường Đại học Trà Vinh, tên của hệ thống và phần đăng nhập yêu cầu người dùng điền đúng thông tin tên đăng nhập và mật khẩu. Tài khoản do hệ thống cung cấp, người dùng không thể tự đăng ký.



Hình 3.7 Thiết kế giao diện trang đăng nhập hệ thống

### Giao diện trang chủ

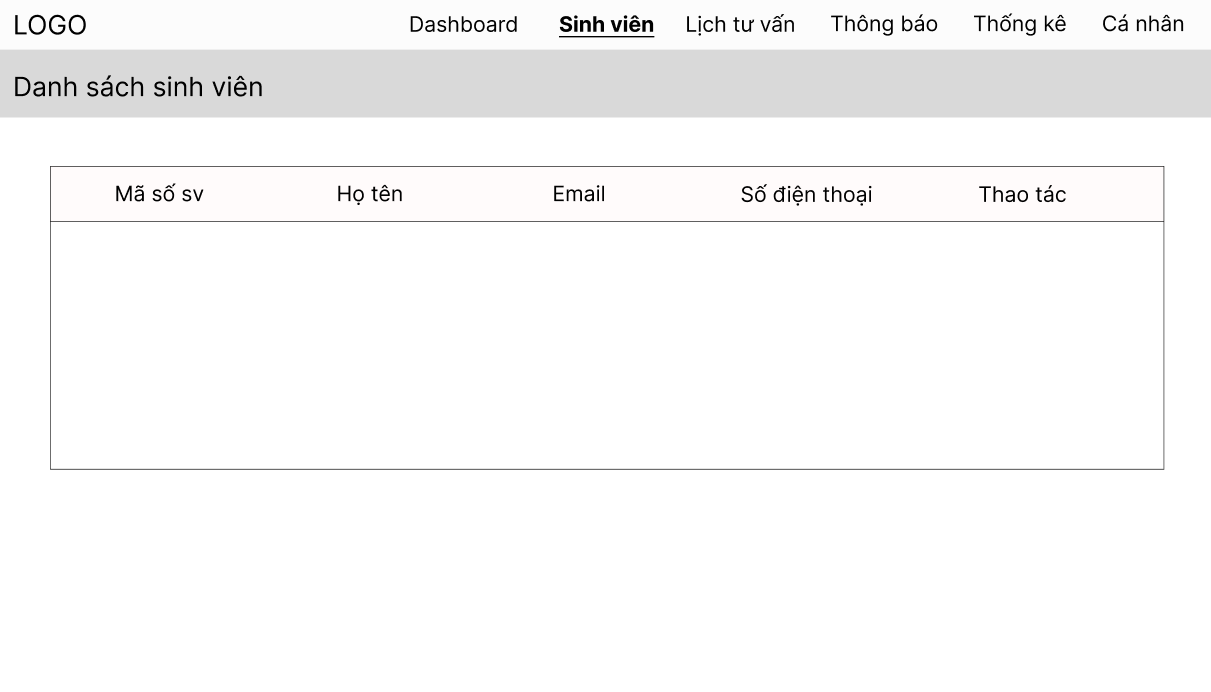
Giao diện trang chủ hiển thị các chức năng cơ bản của cố vấn cũng như sinh viên, các chức năng truy cập nhanh của hệ thống.



Hình 3.8 Thiết kế giao diện trang chủ hệ thống

### Giao diện trang sinh viên

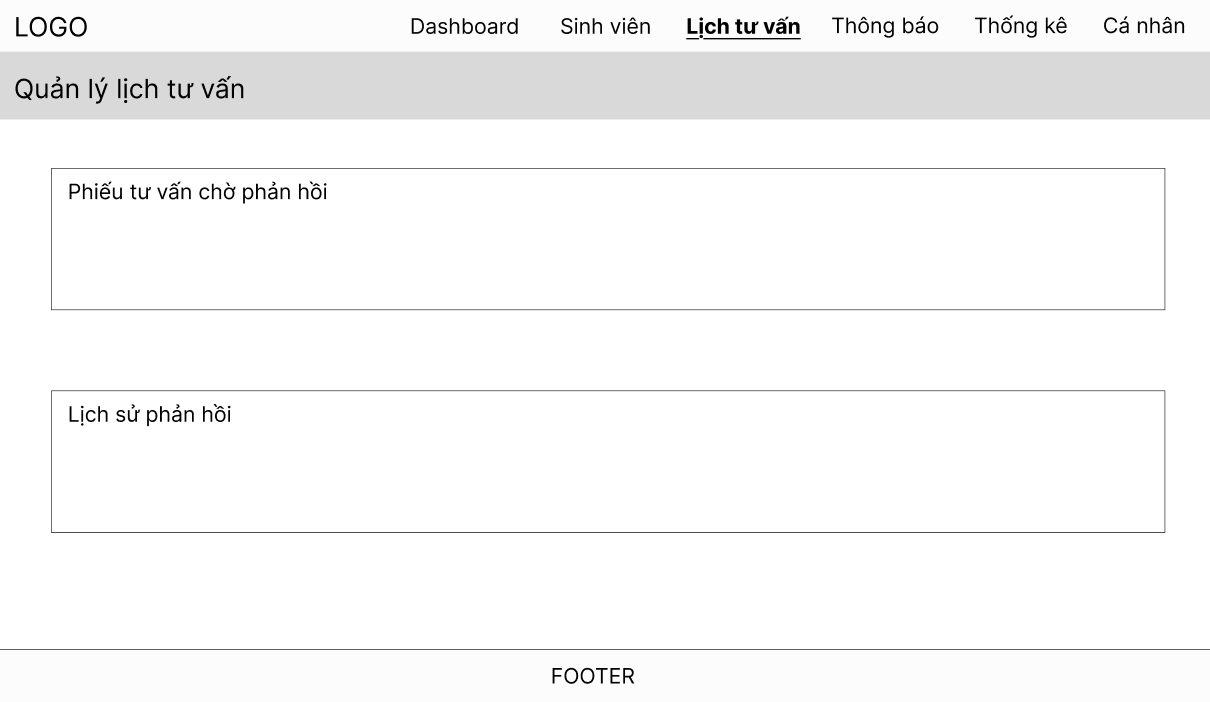
Trang sinh viên sẽ hiển thị toàn bộ sinh viên mà cố vấn đó phụ trách, còn đối với sinh viên chỉ hiển thị thông tin của riêng sinh viên đó.



Hình 3.9 Thiết kế giao diện trang sinh viên

### Giao diện trang lịch tư vấn

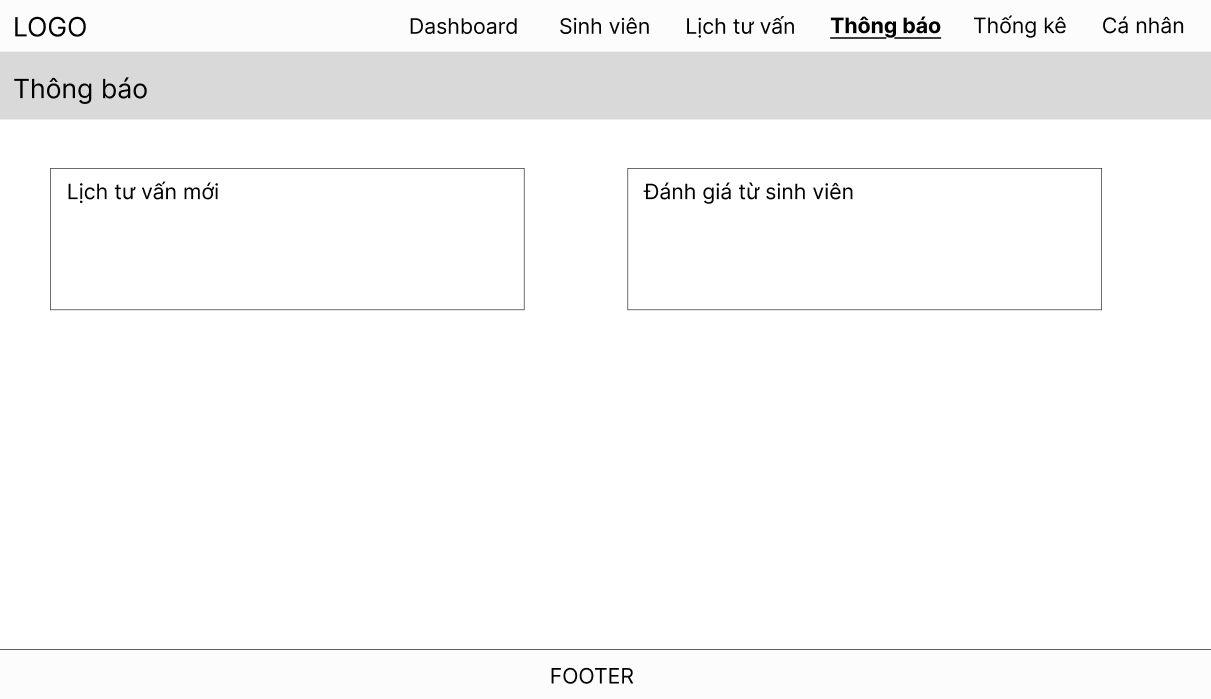
Trang lịch tư vấn sẽ hiển thị những phiếu tư vấn chờ phản hồi và lịch sử phản hồi đối với cố vấn học tập, hiển thị phiếu tư vấn đã đăng ký và lịch sử đánh giá đối với sinh viên.



Hình 3.10 Thiết kế giao diện trang lịch tư vấn

### Giao diện trang thông báo

Trang thông báo sẽ hiển thị lịch tư vấn mới chưa phản hồi và đánh giá của sinh viên đối với cố vấn, hiển thị phản hồi của cố vấn đối với sinh viên đăng nhập.



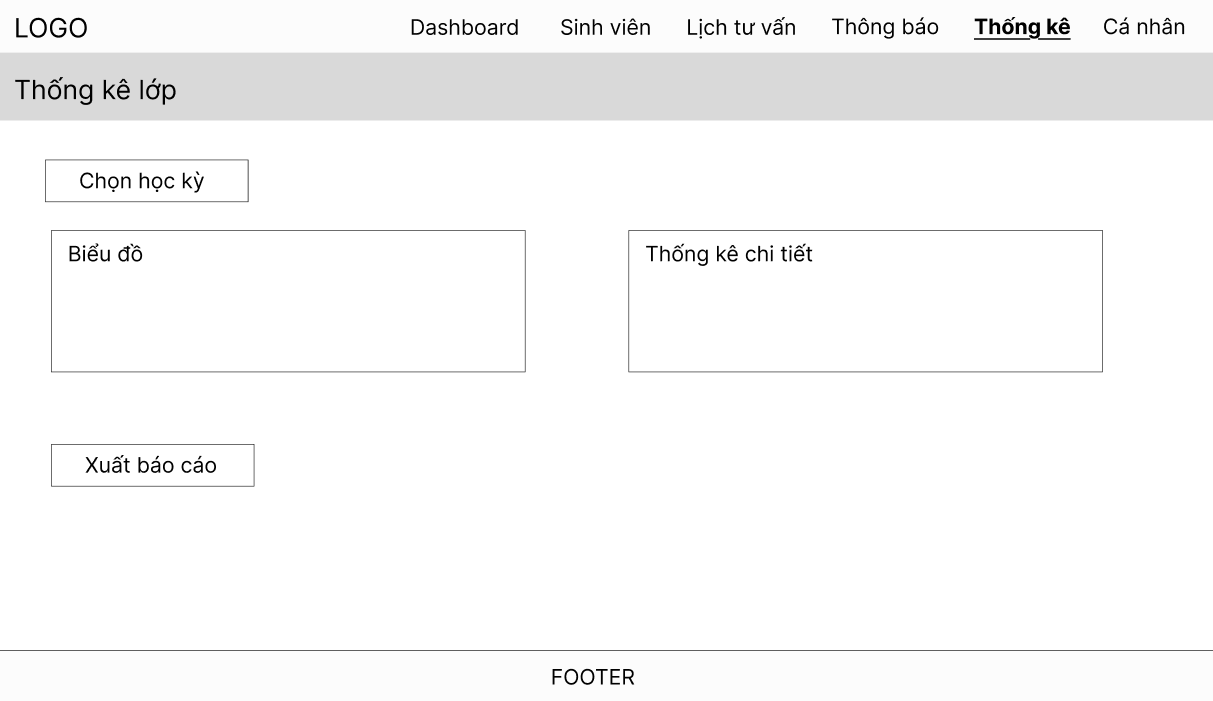
Hình 3.11 Thiết kế giao diện trang thông báo

### Giao diện trang thống kê

Trang thống kê đối với cố vấn: thể hiện biểu đồ và thống kê chi tiết tình hình học tập của tất cả sinh viên mà cố vấn đó phụ trách.

Đối với sinh viên: hiển thị biểu đồ và chi tiết tình hình học tập cá nhân (số môn đạt và chưa đạt).

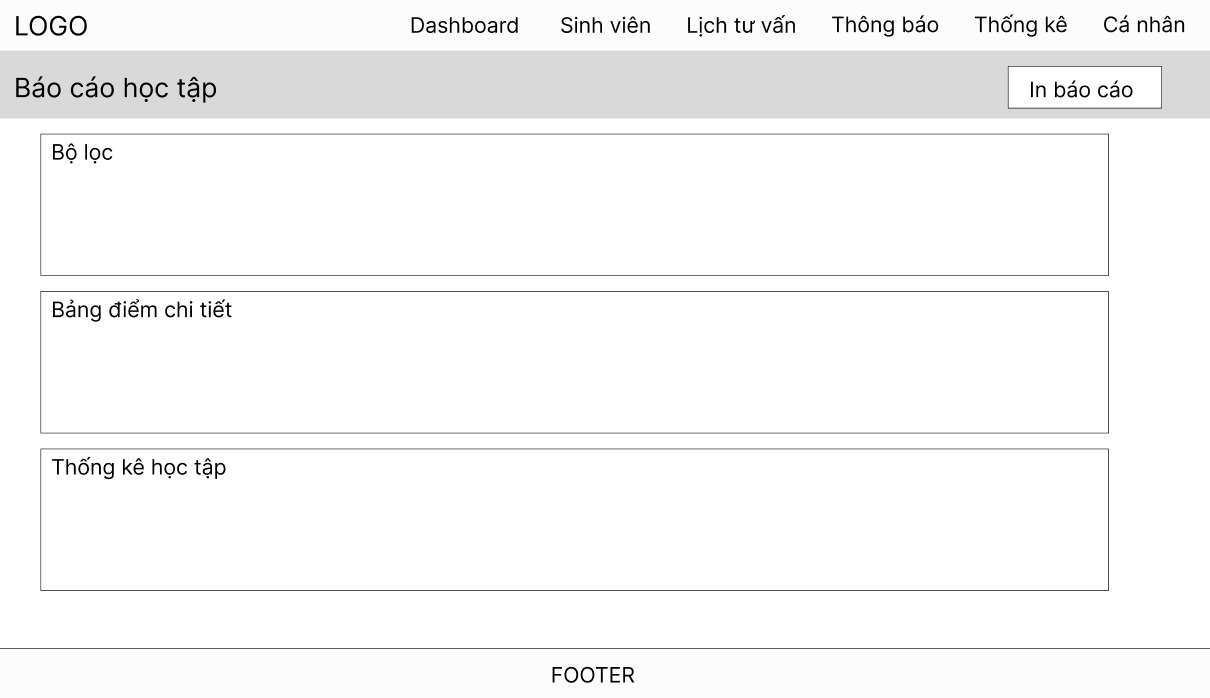
Người dùng có thể lựa chọn học kỳ để hiển thị và có thể xuất báo cáo nếu cần thiết.



Hình 3.12 Thiết kế giao diện trang thống kê

### Giao diện tranh xuất báo cáo

Xuất ra báo cáo tình hình học tập của tất cả sinh viên mà cố vấn đó phụ trách.

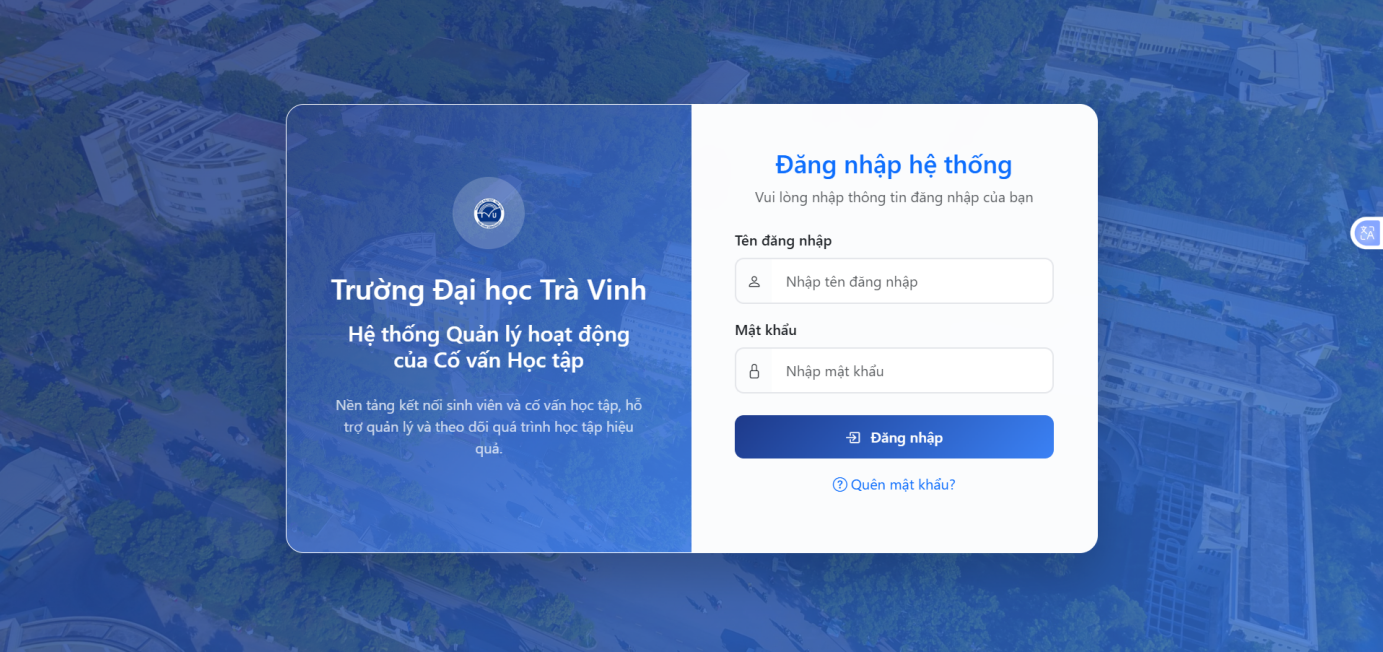


Hình 3.13 Thiết kế giao diện trang xuất báo cáo

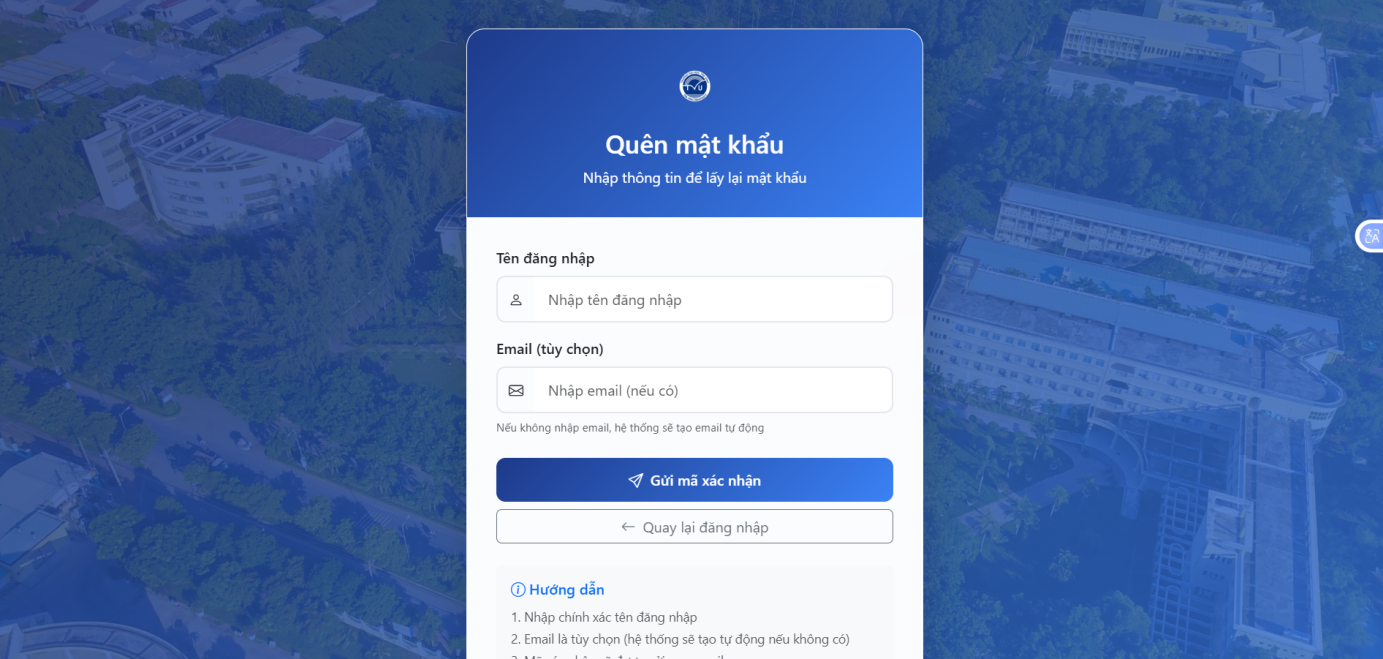
# KẾT QUẢ THỰC NGHIỆM

## Giao diện trang đăng nhập

Giao diện trang đăng nhập được thiết kế đơn giản, dễ sử dụng người dùng cần điền đúng thông tin tên đăng nhập và mật khẩu để đăng nhập, tài khoản do hệ thống cung cấp nên người dùng không thể đăng ký tài khoản. Nếu người dùng quên mật khẩu, chọn quên mật khẩu để tạo lại mật khẩu mới thông qua xác thực email.



Hình 4.1 Giao diện trang đăng nhập hệ thống

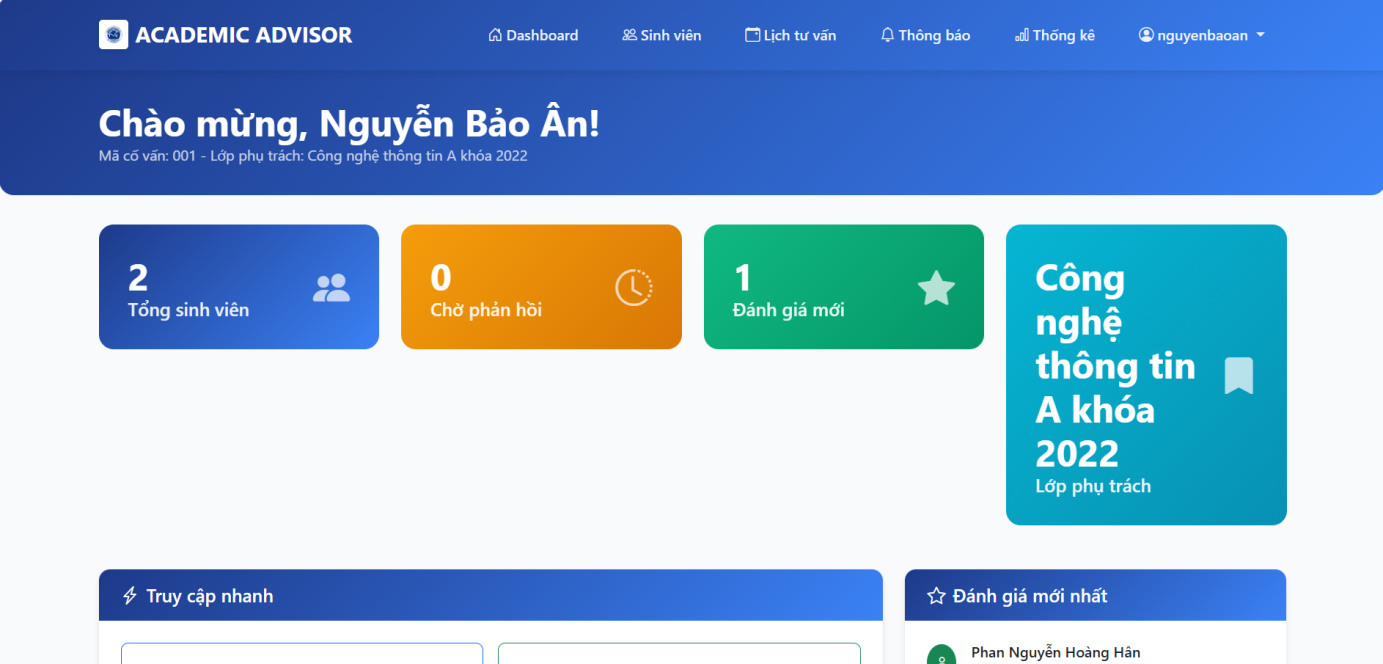


Hình 4.2 Giao diện xác thực tài khoản để lấy lại mật khẩu

## Giao diện cố vấn học tập đăng nhập

### Giao diện trang chủ

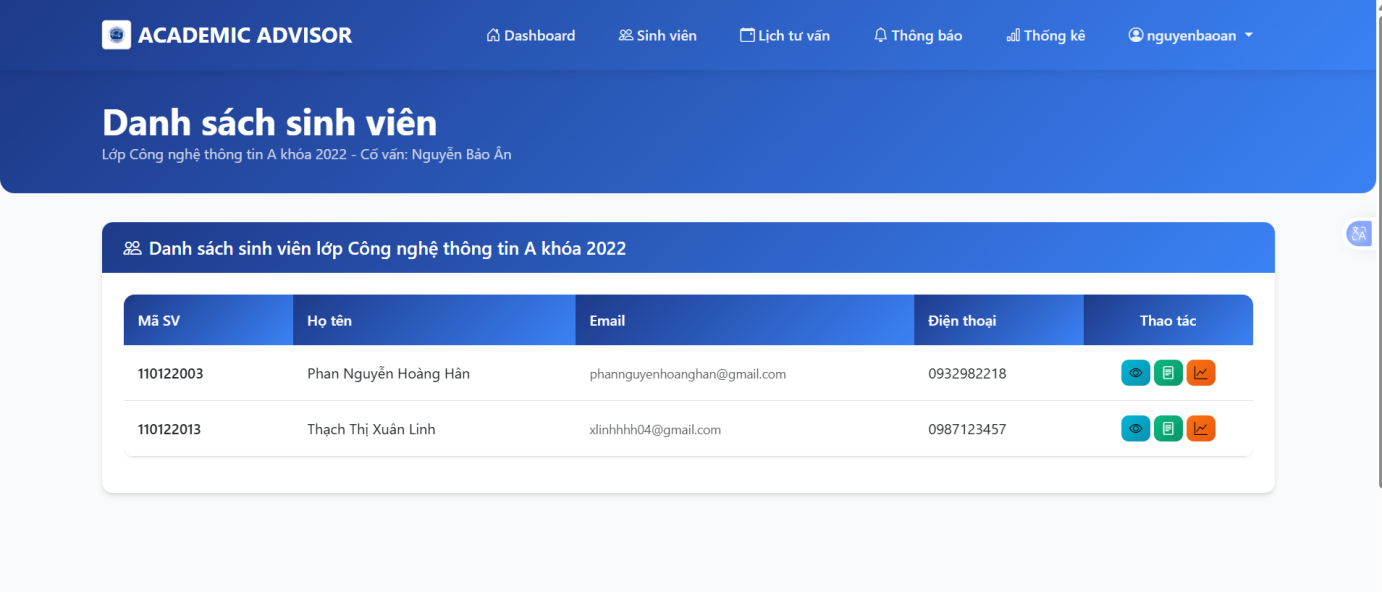
Giao diện trang chủ hiện thị tổng số sinh viên của lớp mình phụ trách, số lượng phiếu tư vấn của sinh viên chờ phản hồi và đánh giá mới từ sinh viên.



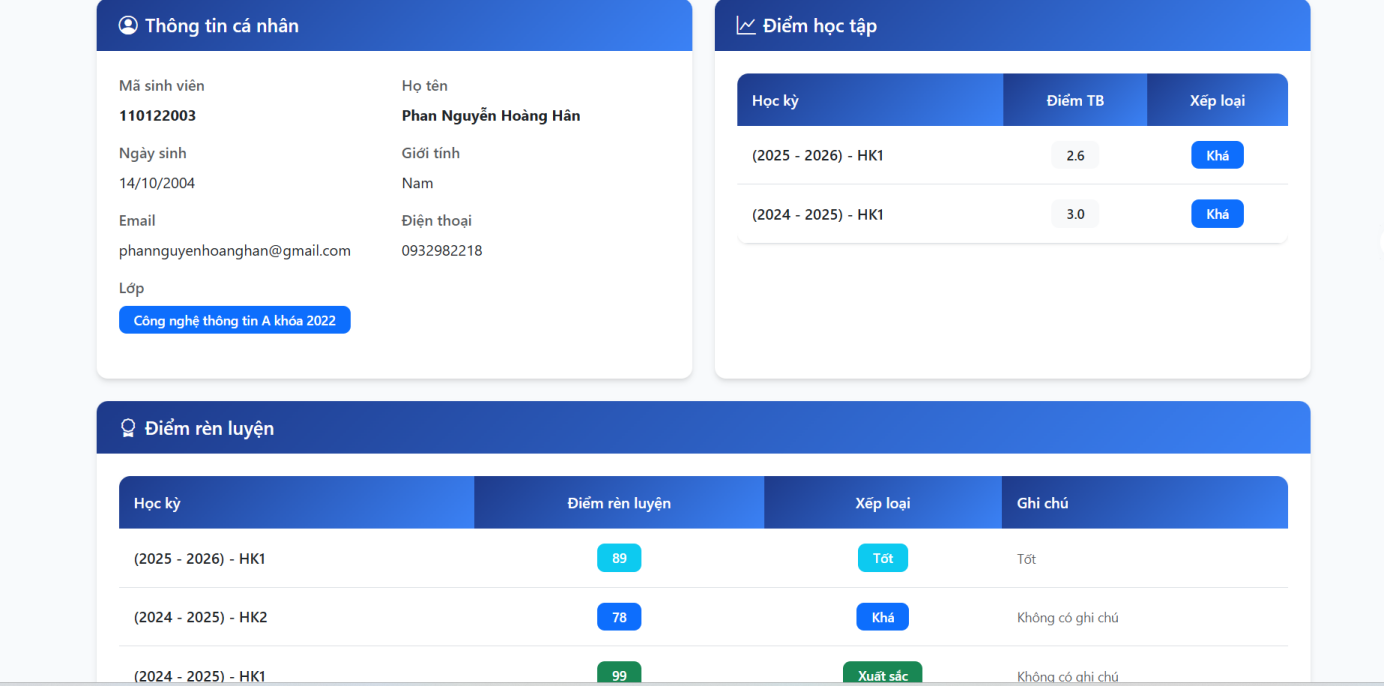
Hình 4.3 Giao diện trang chủ của cố vấn học tập

### Giao diện trang quản lý sinh viên

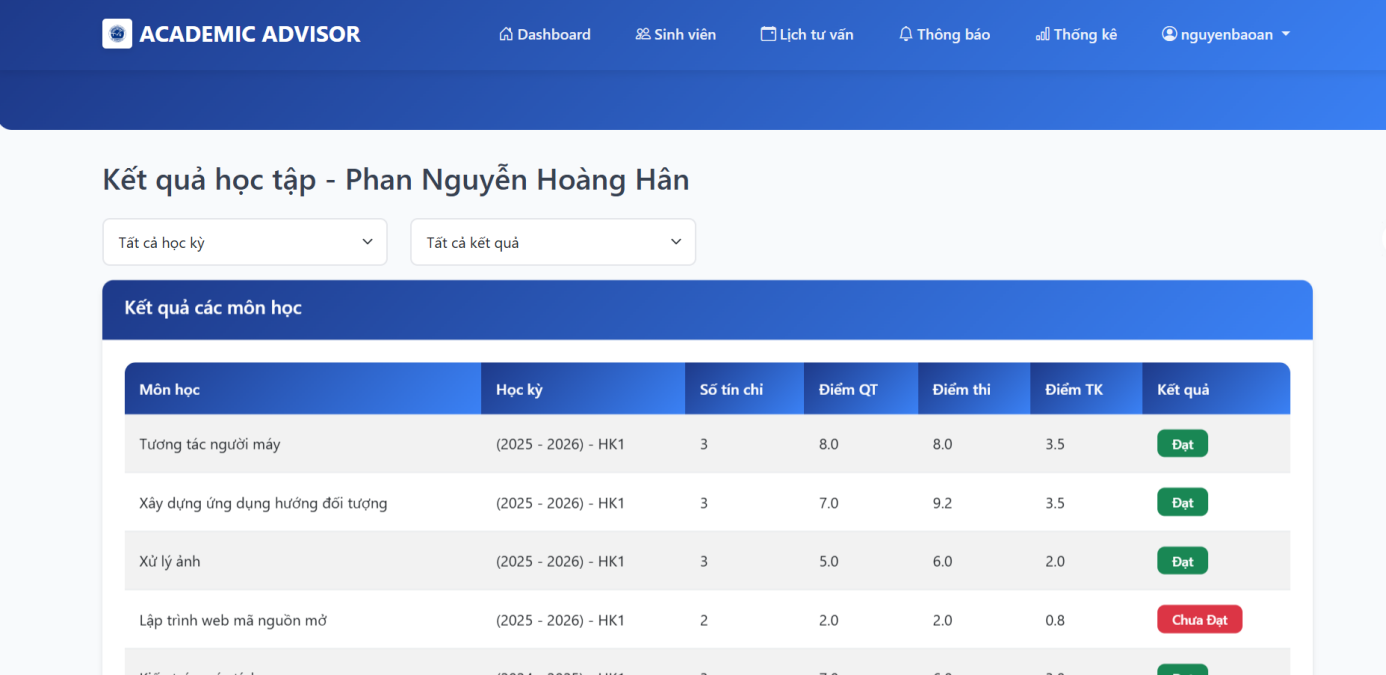
Giao diện hiển thị danh sách sinh viên của lớp mà cố vấn đang đăng nhập phụ trách, cố vấn có thể lựa chọn xem thông tin chi tiết, kết quả học tập, rèn luyện và thống kê kết quả của từng sinh viên.



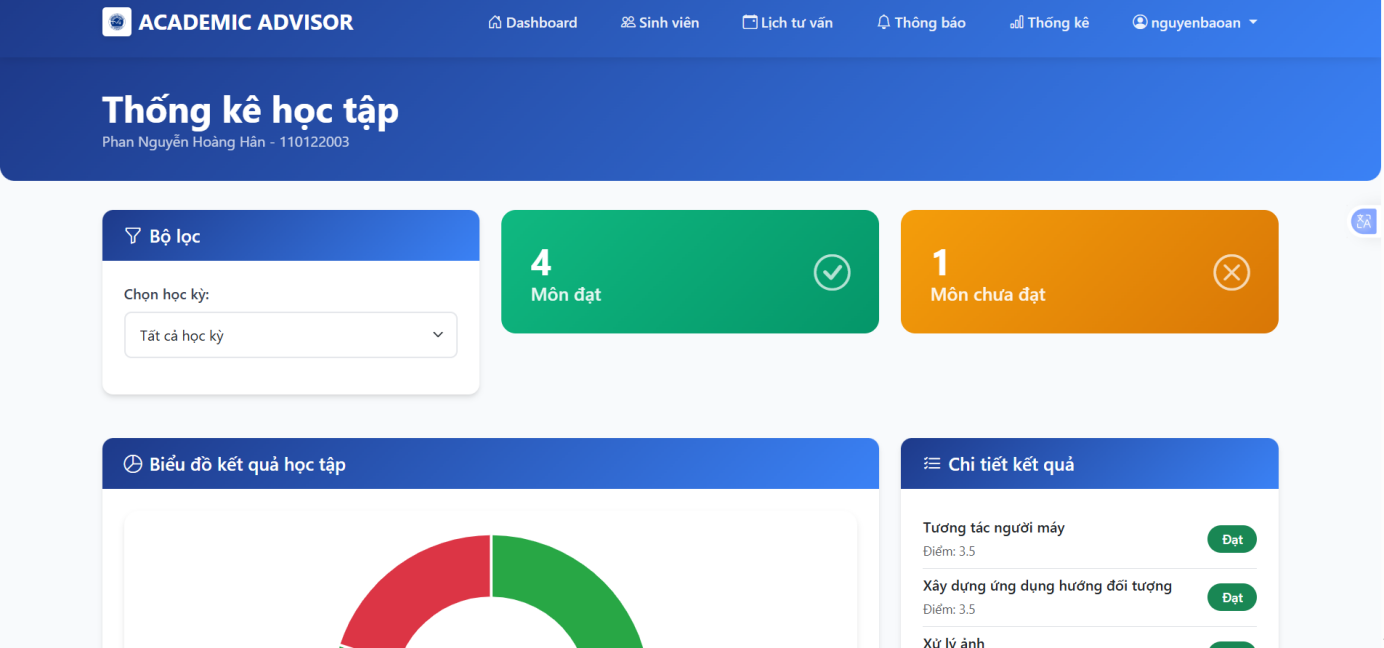
Hình 4.4 Giao diện trang sinh viên



Hình 4.5 Trang thôn tin cá nhân chi tiết của sinh viên



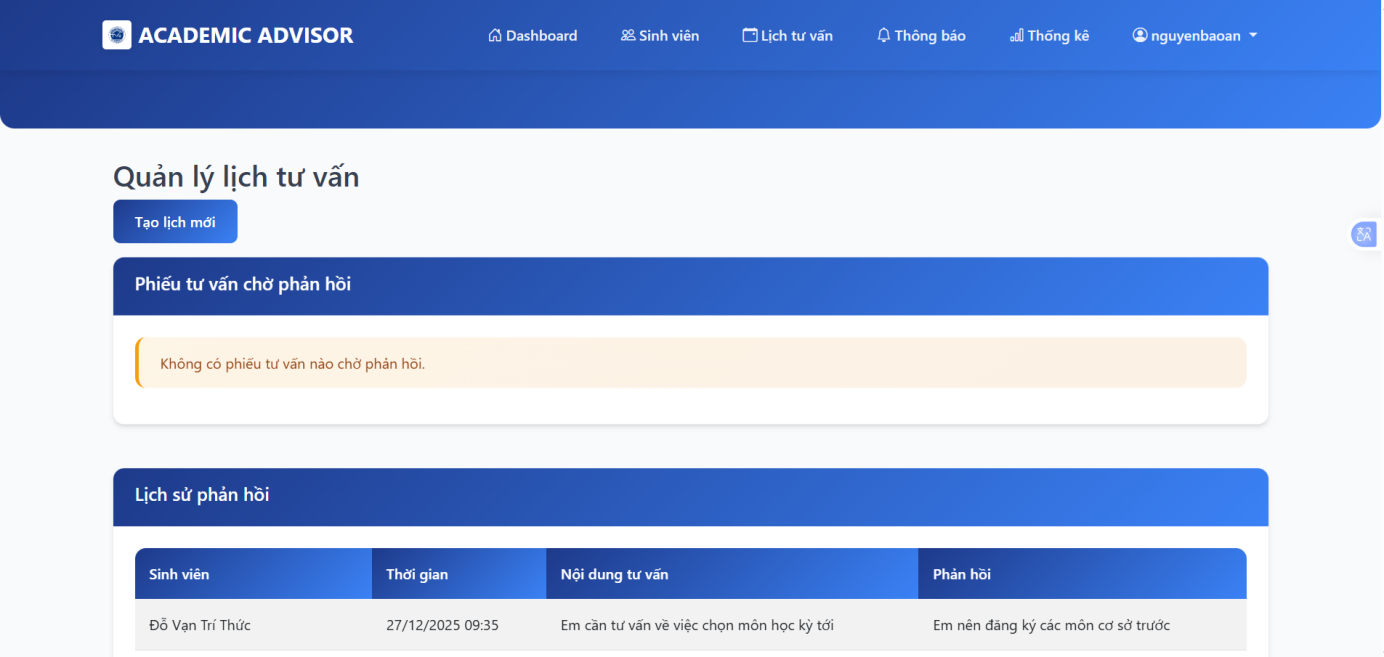
Hình 4.6 Trang hiển thị kết quả học tập chi tiết của sinh viên



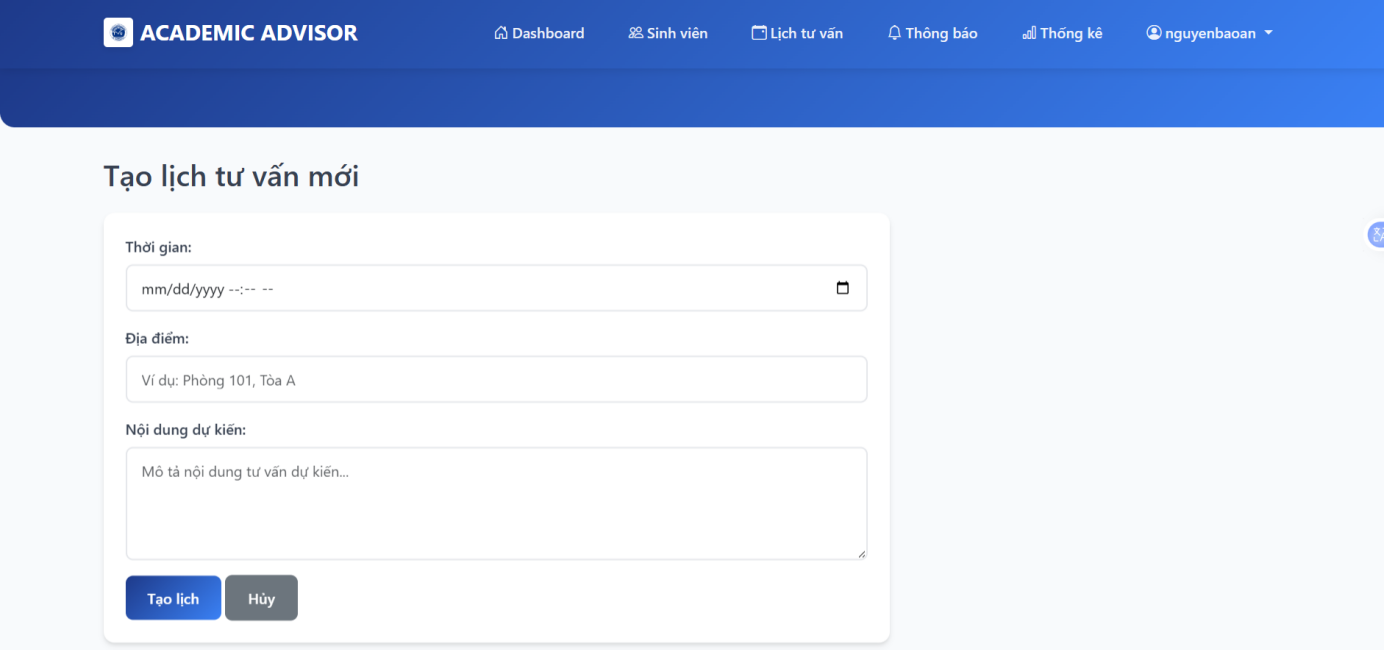
Hình 4.7 Trang thống kê học tập của sinh viên

### Giao diện trang lịch tư vấn

Hiển thị phiếu tư vấn chờ phản hồi của sinh viên và lịch sử phản hôi của cố vấn. Cố vấn có thể tạo lịch tư vấn mới để sinh viên đăng ký.



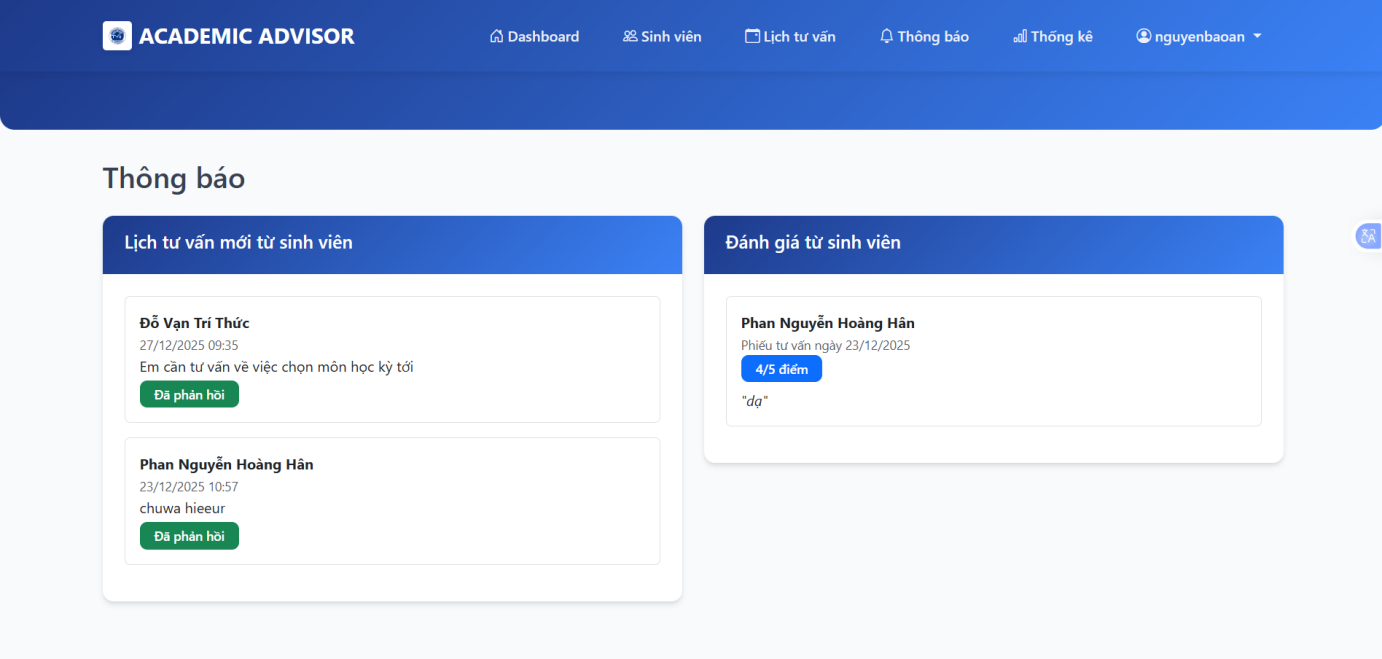
Hình 4.8 Giao diện trang quản lý lịch tư vấn



Hình 4.9 Giao diện tạo lịch tư vấn mới

### Giao diện thông báo

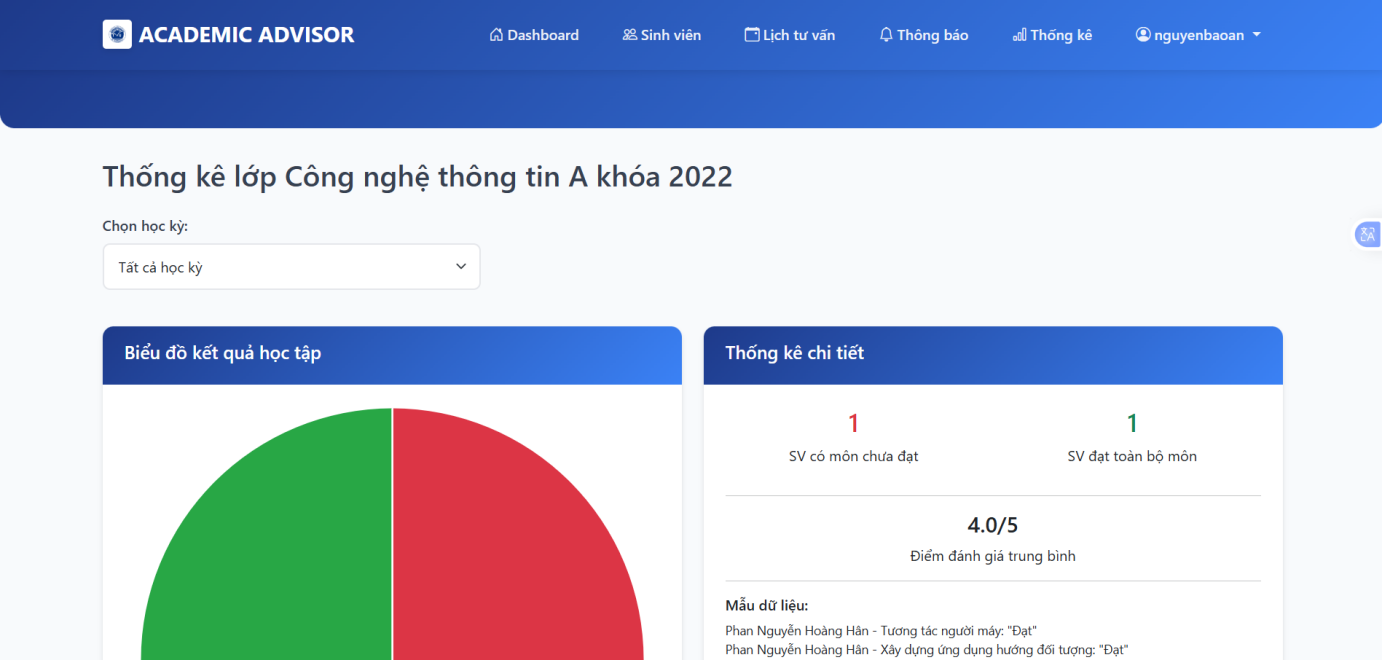
Giao diện hiển thị lịch tư vấn mới của sinh viên đăng ký tư vấn và những đánh giá mới của sinh viên về phản hồi của giảng viên.



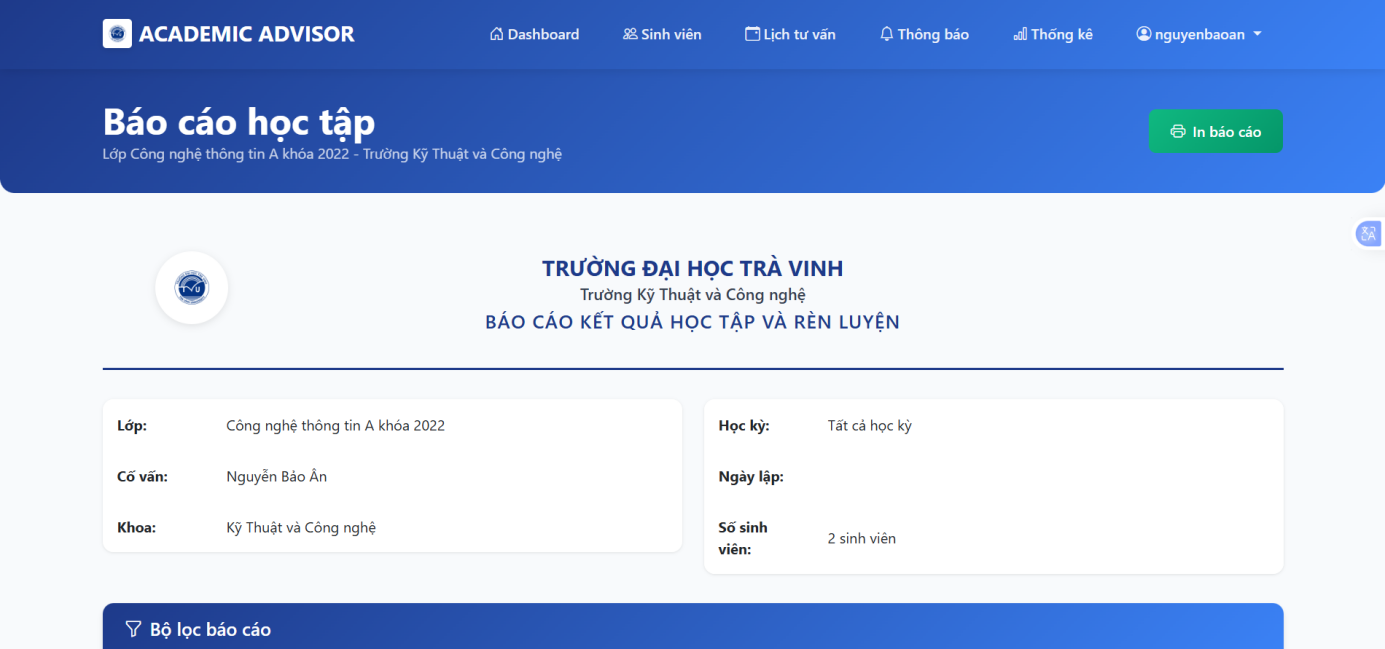
Hình 4.10 Giao diện trang thông báo của cố vấn

### Trang thống kê kết quả học tập

Giao diện hiển thị thống kê về kết quả học tập của tất cả sinh viên trong lớp của cố vấn phụ trách, cố vấn có thể lựa chọn xem theo từng học kỳ và xuất báo cáo nếu cần thiết.



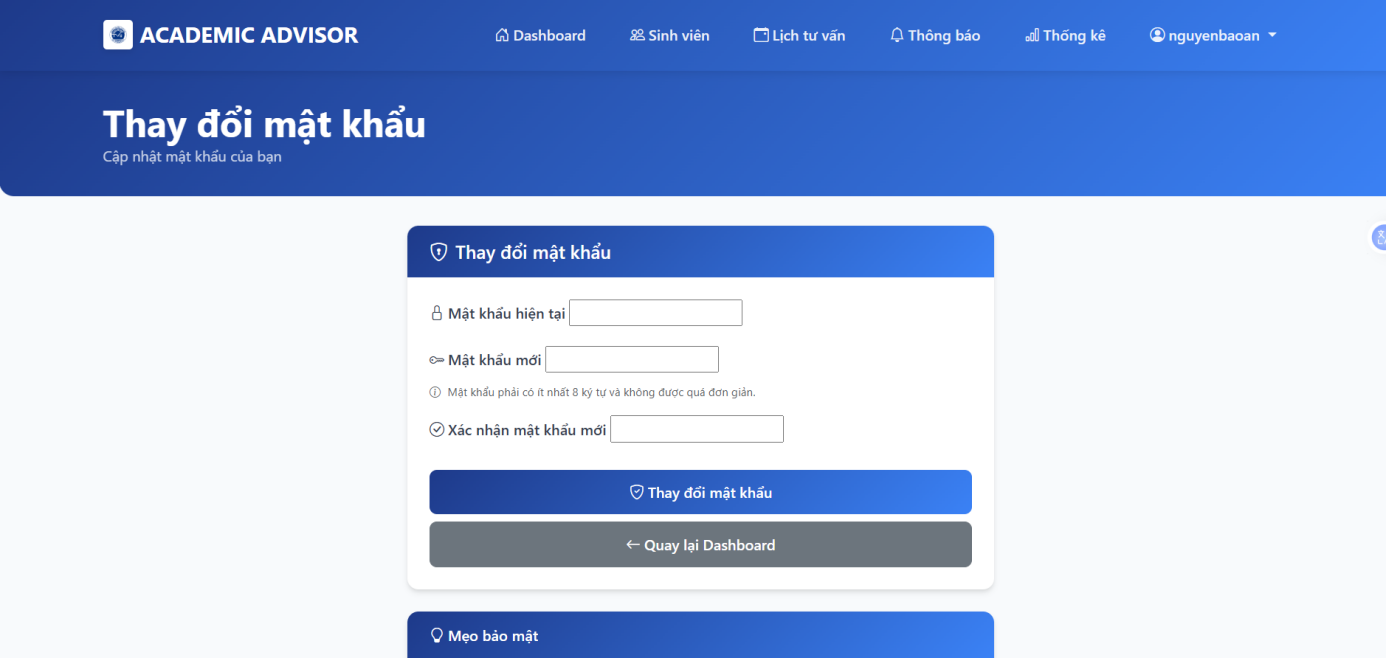
Hình 4.11 Giao diện thống kê kết quả học tập của lớp học



Hình 4.12 Giao diện xuất báo cáo chi tiết

### Giao diện trang thay đổi mật khẩu

Để đảm bảo tính bảo mật, người dùng có thể thay đổi mật khẩu nếu cần thiết.

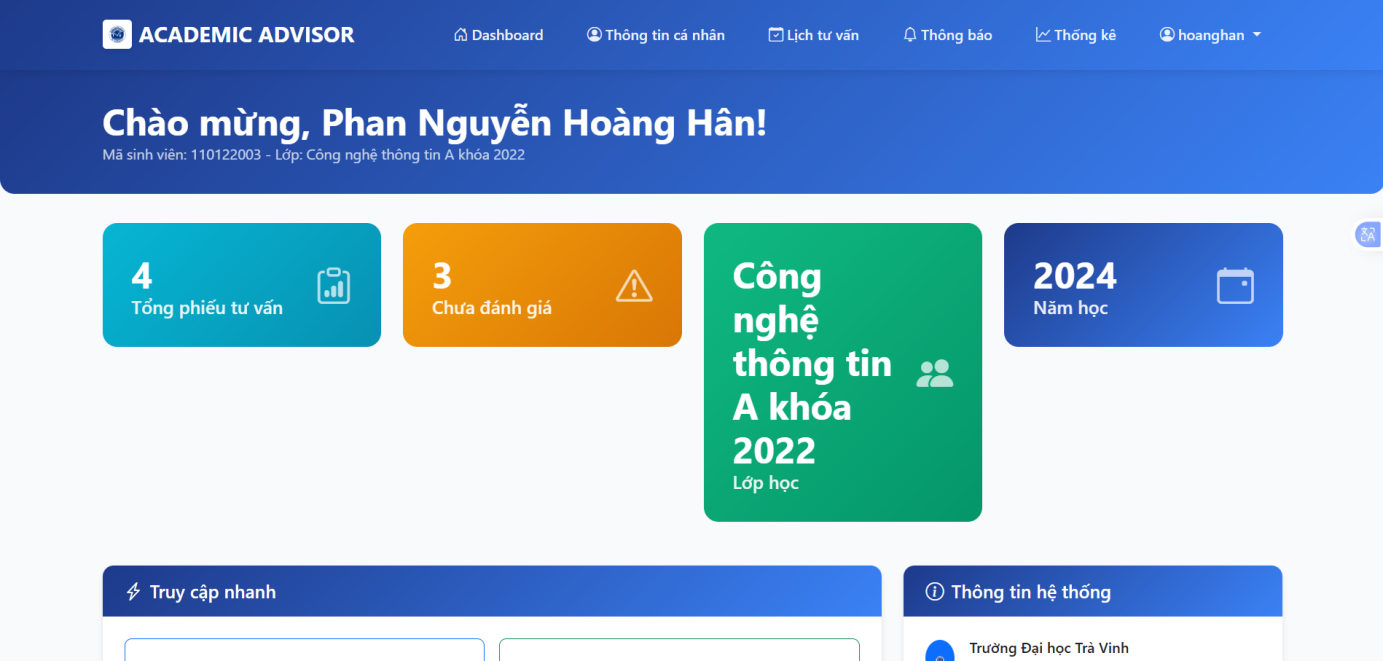


Hình 4.13 Giao diện trang thay đổi mật khẩu

## Giao diện sinh viên đăng nhập

### Giao diện trang chủ

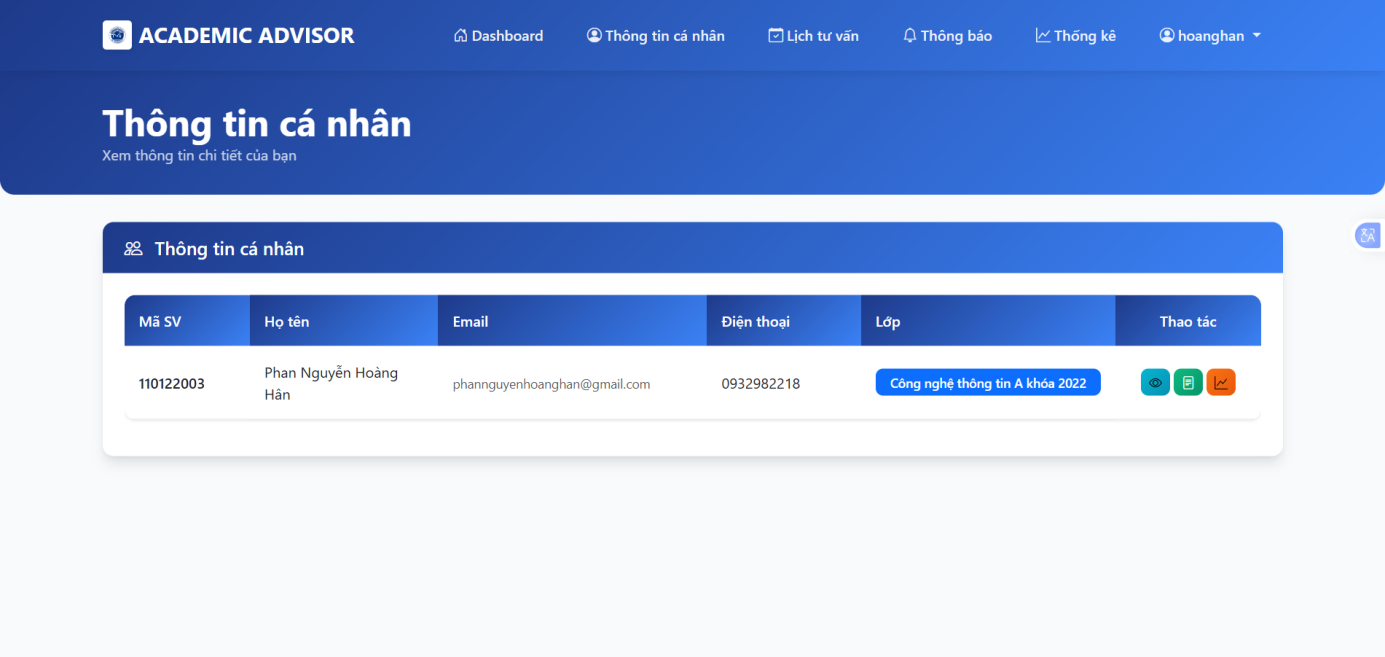
Tương tự như trang chủ đối với cố vấn học tập, trang chủ của sinh viên hiển thị các thông tin cơ bản như lớp, số phiếu tư vấn và số phiếu chưa đánh giá phản hồi của cố vấn học tập.



Hình 4.14 Giao diện trang chủ sinh viên đăng nhập

### Giao diện trang thông tin cá nhân

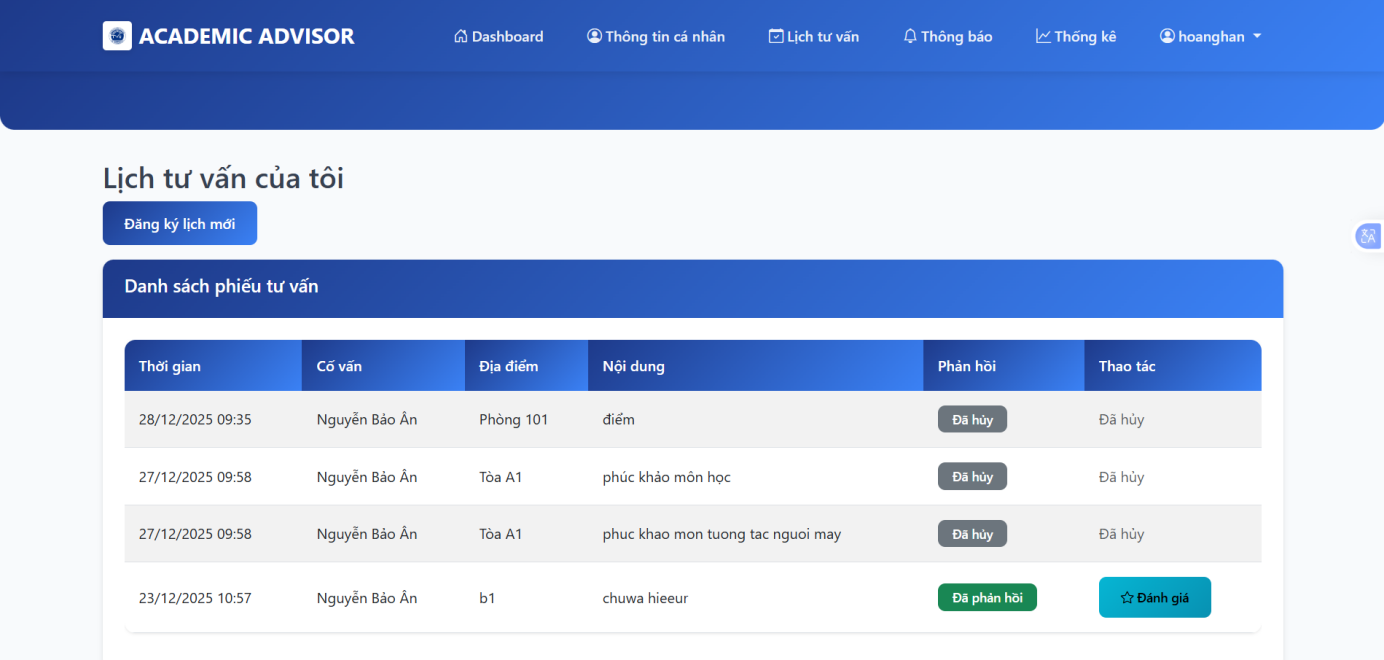
Trang này hiển thị các thông tin cá nhân của sinh viên đăng nhập vào hệ thống, sinh viên có thêm xem chi tiết thông tin, kết quả học tập và rèn luyện, thống kê kết quả học tập của mình.



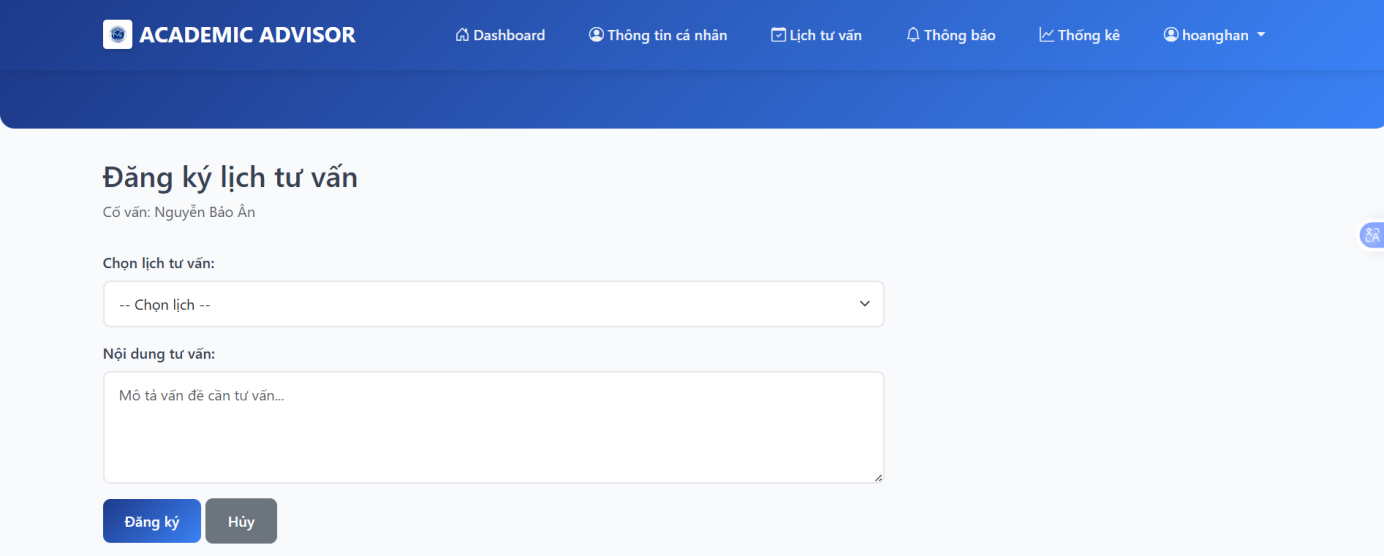
Hình 4.15 Giao diện trang thông tin cá nhân của sinh viên

### Giao diện trang lịch tư vấn

Giao diện hiển thị danh sách tư vấn của bản thân sinh viên và lịch sử đánh giá phản hồi của cố vấn học tập, sinh viên có thể đăng ký lịch tư vấn mới nếu cần thiết.



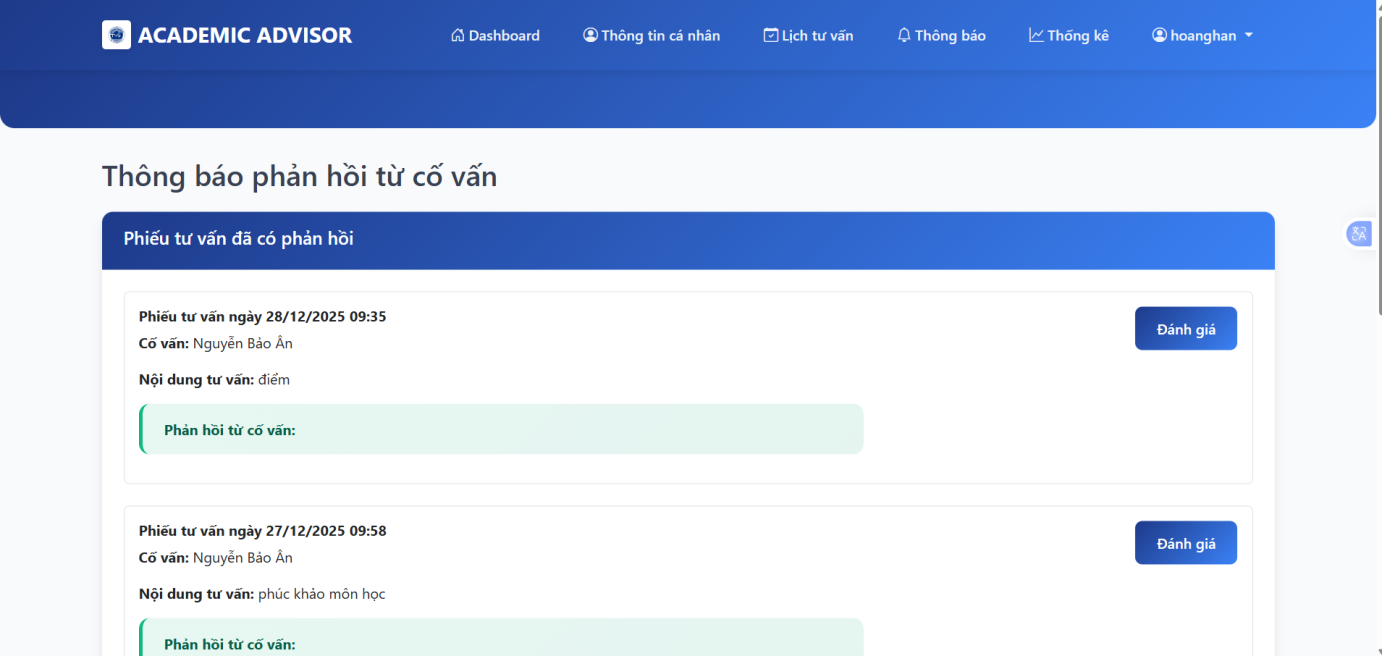
Hình 4.16 Giao diện trang lịch tư vấn của sinh viên



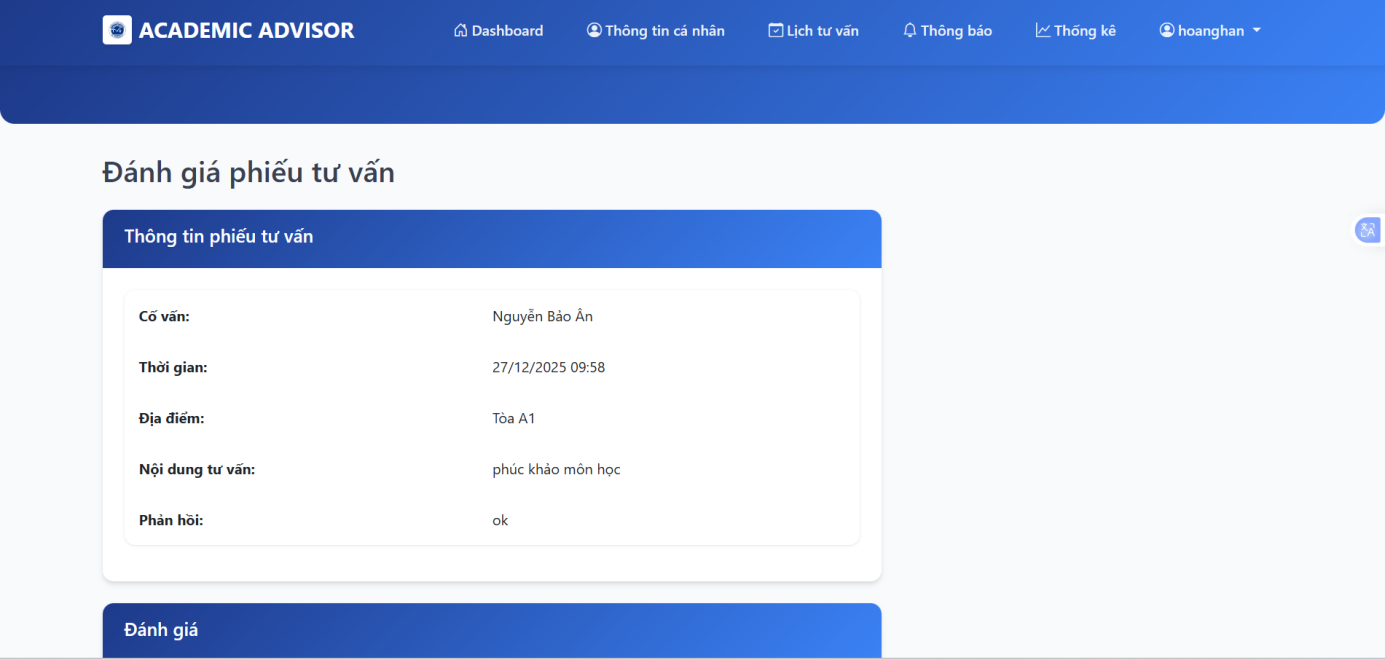
Hình 4.17 Giao diện đăng ký lịch tư vấn mới

### Giao diện trang thông báo

Trang thông báo của sinh viên hiển thị những phản hồi của cố vấn học tập về các phiếu tư vấn đã gửi, sinh viên cũng có thể đánh giá phản hồi của cố vấn tại đây.



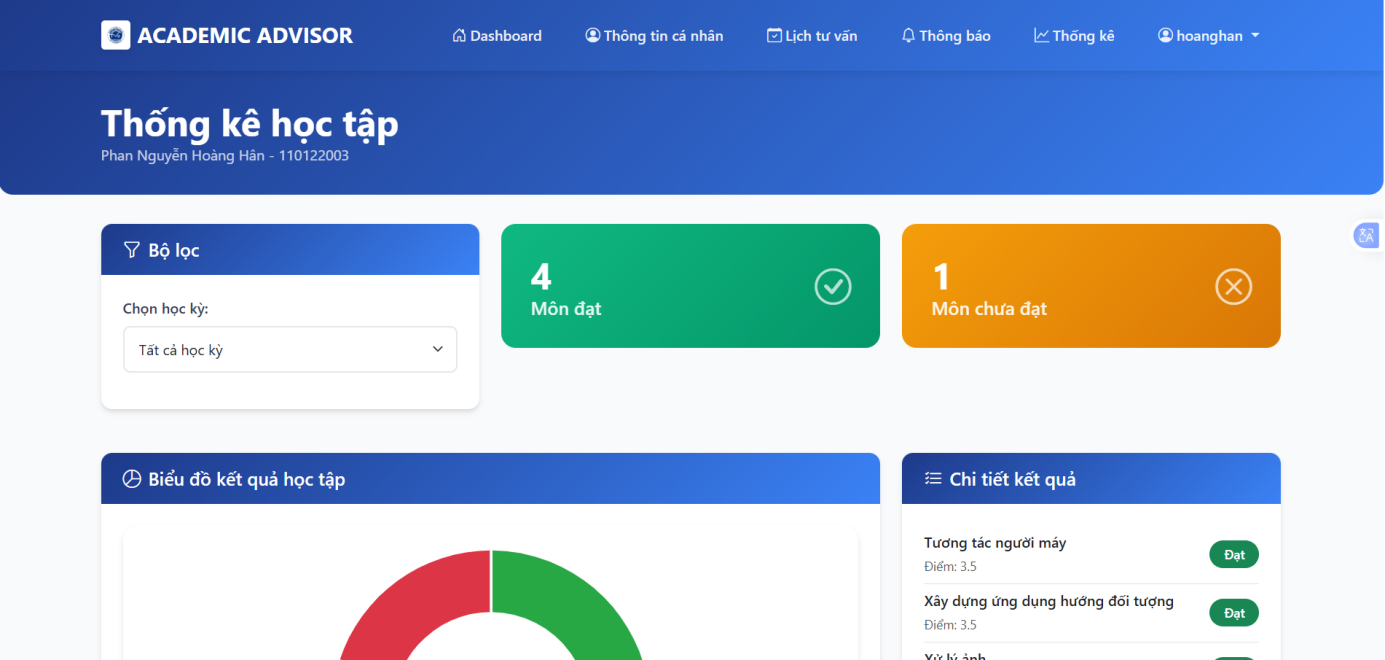
Hình 4.18 Giao diện thông báo của sinh viên



Hình 4.19 Giao diện đánh giá phản hồi của cố vấn

### Trang thống kê kết quả học tập

Giao diện hiển thị biểu đồ kết quả học tập của sinh viên đăng nhập vào hệ thống, sinh viên có thể lựa chọn xem theo từng học kỳ.



Hình 4.20 Giao diện trang thống kê kết quả học tập của sinh viên

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết quả đạt được

Sau quá trình nghiên cứu, thiết kế và hiện thực hóa, đề tài đã đạt được những kết quả chính sau:

Thứ nhất, đề tài đã xây dựng thành công một hệ thống web quản lý hoạt động của Cố vấn học tập dựa trên nền tảng Python Django, đáp ứng đúng định hướng nghiên cứu ban đầu. Hệ thống cho phép quản lý tập trung thông tin sinh viên, giảng viên cố vấn, lớp học, học kỳ và các hoạt động tư vấn học tập, góp phần thay thế các phương thức quản lý thủ công trước đây.

Thứ hai, hệ thống đã hiện thực hóa đầy đủ các chức năng nghiệp vụ cốt lõi, bao gồm: quản lý thông tin sinh viên và cố vấn học tập; ghi nhận lịch tư vấn, nội dung tư vấn và kết quả đánh giá; phân quyền người dùng rõ ràng theo các vai trò như quản trị viên, cố vấn học tập và sinh viên; hỗ trợ thống kê và xuất báo cáo phục vụ công tác quản lý của Khoa và Nhà trường.

Thứ ba, mô hình kiến trúc MVT của Django được áp dụng một cách hợp lý, giúp hệ thống có cấu trúc rõ ràng, dễ bảo trì và mở rộng. Việc sử dụng Django ORM giúp giảm thiểu sự phụ thuộc vào các câu lệnh SQL phức tạp, đồng thời đảm bảo tính toàn vẹn và an toàn dữ liệu.

Thứ tư, hệ thống đã được triển khai thử nghiệm thành công bằng Docker và nền tảng Render, chứng minh tính khả thi của mô hình triển khai trên môi trường thực tế. Điều này cho thấy hệ thống không chỉ dừng lại ở mức nghiên cứu học thuật mà còn có khả năng ứng dụng trong thực tiễn.

Cuối cùng, thông qua đề tài, sinh viên đã nâng cao kiến thức và kỹ năng thực hành về phân tích yêu cầu, thiết kế cơ sở dữ liệu, lập trình web với Django, quản lý mã nguồn bằng GitHub và triển khai hệ thống trên môi trường cloud – đây là những kỹ năng quan trọng đối với công việc sau khi ra trường.

## Hạn chế

Bên cạnh những kết quả đạt được, đề tài vẫn còn tồn tại một số hạn chế nhất định:

Chức năng phân tích dữ liệu và cảnh báo học vụ mới chỉ dừng lại ở mức thống kê cơ bản, chưa tích hợp các mô hình phân tích nâng cao hoặc trí tuệ nhân tạo để dự đoán nguy cơ học tập của sinh viên như một số hệ thống thương mại hiện nay.

Giao diện người dùng tuy đã đảm bảo tính trực quan và dễ sử dụng, nhưng vẫn còn đơn giản, chưa có nhiều yếu tố tương tác nâng cao và chưa được tối ưu hoàn toàn cho các thiết bị di động (mobile-first design).

Hệ thống hiện tại chưa tích hợp sâu với các hệ thống nội bộ khác của nhà trường như hệ thống quản lý đào tạo, hệ thống điểm, hoặc cổng thông tin sinh viên. Do đó, dữ liệu vẫn cần nhập thủ công ở một số khâu.

Do giới hạn về thời gian và phạm vi đồ án cá nhân, đề tài chưa triển khai đầy đủ các cơ chế bảo mật nâng cao như xác thực đa yếu tố (MFA), API token, hay chuẩn OAuth2/JWT cho các dịch vụ mở rộng trong tương lai.

## Hướng phát triển

Nâng cấp chức năng phân tích và dự báo học vụ, bằng cách tích hợp các thuật toán học máy (machine learning) để phát hiện sớm sinh viên có nguy cơ học tập yếu, từ đó hỗ trợ cố vấn học tập đưa ra các biện pháp can thiệp kịp thời.

Mở rộng hệ thống báo cáo và dashboard trực quan, bổ sung các biểu đồ thống kê, KPI và báo cáo tự động theo học kỳ, năm học, giúp Ban chủ nhiệm Khoa và Nhà trường dễ dàng theo dõi hiệu quả công tác cố vấn học tập.

Tối ưu giao diện người dùng và trải nghiệm người dùng (UI/UX) theo hướng hiện đại, thân thiện với thiết bị di động, đồng thời nâng cao khả năng tương tác giữa sinh viên và cố vấn học tập thông qua thông báo thời gian thực hoặc chat nội bộ.

Tích hợp hệ thống với các nền tảng quản lý đào tạo hiện có của nhà trường thông qua API, nhằm đồng bộ dữ liệu điểm số, lịch học và kết quả rèn luyện, giảm thiểu thao tác nhập liệu thủ công.

Hệ thống có thể được phát triển theo mô hình dịch vụ (SaaS) để áp dụng cho nhiều khoa hoặc nhiều trường đại học khác nhau, góp phần nâng cao hiệu quả công tác quản lý cố vấn học tập trong bối cảnh chuyển đổi số giáo dục hiện nay.

# TÀI LIỆU THAM KHẢO

[1] N. Idris, C. F. M. Foozy, “A Generic Review of Web Technology: Django and Flask,” International Journal of Advanced Computing Science and Engineering, vol. 2, no. 1, pp. 34–40, Apr. 2020.

[2] M. Gendreau, “Using Python, Django and MySQL in a Database Course,” MICS Symposium Proceedings, 2011.

[3] Python Web Development With Django, GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/python/python-web-development-django/. [Accessed Dec. 2025].

[4] Django Documentation, “Django Web Framework,” Django Project. [Online]. Available: https://docs.djangoproject.com/en/6.0/. [Accessed Dec. 2025].

[5] MySQL Workbench Reference Manual, MySQL. [Online]. Available: https://dev.mysql.com/doc/workbench/en/. [Accessed Dec. 2025].

[6] MySQL Workbench, Oracle MySQL Products website. [Online]. Available: https://www.mysql.com/products/workbench/. [Accessed Dec. 2025].

[7] FlightPath, “FlightPath,” Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/FlightPath. [Accessed: Dec. 2025].