

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2024-2025

**TÌM HIỂU NGÔN NGỮ LẬP TRÌNH PYTHON
XÂY DỰNG HỆ THỐNG QUẢN LÝ
GIỜ NGHIÊN CỨU KHOA HỌC CỦA
KHOA KT&CN - TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giảng viên hướng dẫn:
Họ tên: **Trần Văn Nam**

Sinh viên thực hiện:
Họ tên: **Phan Nguyễn Hoàng Hân**
MSSV: **110122003**
Lớp: **DA22TTA**

Trà Vinh, 09 tháng 01 năm 2025

KHOA KỸ THUẬT VÀ CÔNG NGHỆ
BỘ MÔN CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ I, NĂM HỌC 2024-2025

**TÌM HIỂU NGÔN NGỮ LẬP TRÌNH PYTHON
XÂY DỰNG HỆ THỐNG QUẢN LÝ
GIỜ NGHIÊN CỨU KHOA HỌC CỦA
KHOA KT&CN - TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giảng viên hướng dẫn:
Họ tên: **Trần Văn Nam**

Sinh viên thực hiện:
Họ tên: **Phan Nguyễn Hoàng Hân**
MSSV: **110122003**
Lớp: **DA22TTA**

Trà Vinh, 09 tháng 01 năm 2025

Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

This image shows a full page of a document template designed for handwriting practice or general note-taking. It consists of approximately 30 evenly spaced horizontal dotted lines across the entire width of the page. The background is plain white, and there are no margins, headers, or footers present.

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 thầy Trần Văn Nam 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 chín 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!

Trà Vinh, ngày ... tháng năm

SINH VIÊN

Phan Nguyễn Hoàng Hân

MỤC LỤC

TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH.....	9
MỞ ĐẦU.....	10
CHƯƠNG 1: TỔNG QUAN.....	11
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT.....	12
2.1 Ngôn ngữ Python.....	12
2.1.1 Giới thiệu về Python.....	12
2.1.2 Ứng dụng.....	12
2.1.3 Tải và cài đặt Python.....	13
2.2 Framework Django.....	14
2.2.1 Khái quát về Django.....	14
2.2.2 Ưu và nhược điểm của Django.....	14
2.2.3 Các tính năng cơ bản.....	15
2.2.4 Cài đặt, tạo và chạy một dự án Django.....	16
2.2.4.1 Cài đặt.....	16
2.2.4.2 Tạo dự án.....	16
2.2.4.3 Chạy server dự án Django.....	16
2.2.5 Các thành phần chính có trong Django.....	17
2.2.5.1 Views.....	17
2.2.5.2 URLS.....	17
2.2.5.3 Models.....	19
2.2.5.4 Hệ thống Admin.....	19
2.3 MySQL.....	21
2.3.1 Khái quát về MySQL.....	21
2.3.2 Ưu và nhược điểm của MySQL.....	21
2.3.3 Kết nối MySQL với Django.....	23
2.4 Mô hình Model-View-Template.....	24
2.4.1 Sơ lược về mô hình.....	24
2.4.2 Ưu và nhược điểm của mô hình MVT.....	26
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU.....	28
3.1 Mô tả bài toán.....	28
3.2 Đặc tả yêu cầu hệ thống.....	28
3.2.1 Yêu cầu chức năng.....	28

3.2.2	Yêu cầu phi chức năng.....	28
3.3	Thiết kế cơ sở dữ liệu trong MySQL Workbench.....	29
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU.....		31
4.1	Giao diện dành cho người dùng.....	31
4.1.1	Giao diện trang chủ.....	31
4.1.2	Giao diện trang đăng nhập.....	32
4.1.3	Giao diện trang đăng ký.....	33
4.1.4	Giao diện trang Sinh viên.....	33
4.1.5	Giao diện trang Giảng viên.....	34
4.1.6	Giao diện trang tìm kiếm.....	35
4.2	Giao diện dành cho người quản trị.....	36
4.2.1	Trang chủ quản trị.....	36
4.2.2	Danh sách danh mục.....	37
4.2.3	Thêm, sửa, xóa sản phẩm.....	37
4.2.4	Quản lý người dùng.....	37
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....		39
5.1	Kết quả đạt được.....	39
5.2	Hạn chế và hướng phát triển.....	39
DANH MỤC TÀI LIỆU THAM KHẢO.....		40

DANH MỤC HÌNH ẢNH

Hình 1. Trang web download python	13
Hình 2. Giao diện trang django ban đầu.....	17
Hình 3. Mã nguồn tạo giao diện trang base.....	18
Hình 4. Mã nguồn giao diện trang home	19
Hình 5. Tạo supperuser	20
Hình 6. Giao diện đăng nhập trang admin	20
Hình 7. Giao diện chính trang admin django	21
Hình 8. Giao diện kết nối cơ sở dữ liệu của MySQL Workbench	23
Hình 9. Kết nối cơ sở dữ liệu trong settings.py.....	24
Hình 10. Mô hình MTV	25
Hình 11. Giao diện đầu trang chủ	31
Hình 12. Giao diện giữa trang chủ	32
Hình 13. Footer.....	32
Hình 14. Giao diện trang đăng nhập	33
Hình 15. Giao diện trang đăng ký	33
Hình 16. Giao diện trang Sinh viên chưa đăng nhập	34
Hình 17. Giao diện trang Sinh viên đã đăng nhập	34
Hình 18. Giao diện trang Giảng viên chưa đăng nhập	35
Hình 19. Giao diện trang Giảng viên đã đăng nhập	35
Hình 20. Giao diện trang tìm kiếm	36
Hình 21. Giao diện trang admin	36
Hình 22. Giao diện hiển thị danh sách các danh mục.....	37
Hình 23. Giao diện trang chỉnh sửa thông tin sinh viên hoặc giảng viên.....	37
Hình 24. Giao diện trang quản lý người dùng.....	38

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
KT&CN	Kỹ thuật và Công nghệ
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
URL	Uniform Resource Locator
HTTP	HyperText Transfer Protocol

TÓM TẮT ĐỒ ÁN CƠ SỞ NGÀNH

Vấn đề nghiên cứu

Đồ án "Tìm hiểu ngôn ngữ lập trình Python. Xây dựng hệ thống quản lý giờ Nghiên cứu Khoa học của Khoa KT&CN - Trường Đại học Trà Vinh" nghiên cứu và ứng dụng Python để phát triển hệ thống quản lý giờ Nghiên cứu Khoa học của Khoa KT&CN - Trường Đại học Trà Vinh đáp ứng các nhu cầu cơ bản của người dùng cũng như của người quản trị hệ thống.

Các hướng tiếp cận

Tiến hành, tham khảo tìm hiểu một số website của Trường Đại học Trà Vinh xem cấu trúc và các nội dung có trong các website của Trường. Ngoài ra em cũng tìm hiểu kiến thức về ngôn ngữ lập trình, công nghệ hiện đại như Python, framework Django, MySQL để áp dụng kết hợp chúng vào việc xây dựng một hệ thống với các chứng năng cơ bản, dễ sử dụng.

Các 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, CSS/Bootstrap và Django Templates để người dùng có thể thao tác trực quan.

Kết quả đạt được

Kết quả đạt được từ đồ án "Tìm hiểu ngôn ngữ lập trình Python. Xây dựng hệ thống quản lý giờ Nghiên cứu Khoa học của Khoa KT&CN - Trường Đại học Trà Vinh" xây dựng được một hệ thống tương đối hoàn chỉnh với các chức năng cơ bản như thêm sửa xóa thông tin người dùng. Giao diện thân thiện với người dùng, người quản trị cũng dễ dàng thêm, sửa, xóa các tài khoản và thông tin người dùng. Mô hình MTV trong Django giúp em phát triển hệ thống quản lý giờ Nghiên cứu Khoa học Khoa KT&CN - Trường Đại học Trà Vinh hiệu quả bằng cách tách biệt rõ ràng các thành phần: Model quản lý dữ liệu, Template tạo giao diện người dùng dễ dàng tùy chỉnh và View xử lý logic và tương tác với người dùng. Điều này giúp hệ thống dễ bảo trì và phát triển thêm các tính năng khác.

MỞ ĐẦU

Lý do chọn đề tài

Em lựa chọn đề tài xây dựng hệ thống bán sách và sử dụng ngôn ngữ lập trình Python vì nó là một ngôn ngữ phổ biến, dễ học, và có ứng dụng rộng rãi trong phát triển web, phần mềm. Django, một framework mạnh mẽ của Python, hỗ trợ xây dựng backend hiệu quả với các tính năng bảo mật, xử lý dữ liệu và logic phức tạp. Về mặt thực tiễn, Xây dựng hệ thống quản lý giờ Nghiên cứu Khoa học của Khoa KT&CN - Trường Đại học Trà Vinh đáp ứng nhu cầu quản lý tài khoản, các thông tin của người dùng. Ngoài ra, đề tài này còn giúp em nâng cao kỹ năng lập trình, áp dụng lý thuyết vào thực tế, làm quen với quy trình phát triển phần mềm hiện đại và là bước đệm, sự chuẩn bị cho môi trường làm việc khi ra trường.

Mục đích nghiên cứu

Mục đích chính của đề tài là Xây dựng hệ thống quản lý giờ Nghiên cứu Khoa học của Khoa KT&CN - Trường Đại học Trà Vinh dựa trên ngôn ngữ lập trình Python và framework Django giúp quản lý thông tin cũng như thời gian nghiên cứu khoa học của sinh viên và giảng viên Khoa KT&CN. Với giao diện thân thiện, đơn giản, người dùng có thể tiếp cận và dễ dàng sử dụng một cách hiệu quả.

Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài này bao gồm hệ thống phần mềm quản lý và quy trình nghiên cứu khoa học tại Khoa KT&CN - Trường Đại học Trà Vinh, đặc biệt là việc quản lý và tối ưu hóa thời gian nghiên cứu khoa học.

CHƯƠNG 1: TỔNG QUAN

Hiện nay, sự phát triển của công nghệ tiên tiến ngày càng mạnh mẽ và ứng dụng của nó ngày càng rộng rãi hơn.

Trước đây, việc quản lý giờ nghiên cứu khoa học thường quản lý một cách trực tiếp thông qua các hồ sơ, bảng tính nên còn gặp nhiều khó khăn trong việc quản lý do các hồ sơ có thể làm mất, bị lạc, hoặc bị côn trùng phá hoại dẫn đến nhiều vấn đề khác sẽ phát sinh trong việc quản lý.

Xuất phát từ thực tế trên, đề tài "đề tài Xây dựng hệ thống quản lý giờ Nghiên cứu Khoa học của Khoa KT&CN - Trường Đại học Trà Vinh" được triển khai nhằm mang đến một giải pháp công nghệ toàn diện. Sử dụng các công nghệ như Python, Django Framework, và MySQL, hệ thống sẽ xây dựng một nền tảng quản lý chuyên nghiệp, đáp ứng đầy đủ các yêu cầu về thêm, sửa, xóa thông tin, quản lý người dùng một cách hiệu quả và dễ dàng sử dụng.

Python là ngôn ngữ lập trình linh hoạt, dễ học và có cộng đồng phát triển mạnh mẽ. Đặc biệt, Python hỗ trợ nhiều thư viện và công cụ mạnh mẽ cho phát triển web, như xử lý dữ liệu, tạo giao diện người dùng, và quản lý cơ sở dữ liệu.

Tuy nhiên, việc phát triển ứng dụng web chỉ với Python thuần có thể mất nhiều thời gian và dễ mắc lỗi, vì lập trình viên phải tự xử lý mọi khía cạnh, từ routing, bảo mật đến quản lý session. Sử dụng một framework như Django giúp tối ưu hóa quy trình phát triển web. Framework cung cấp các công cụ tích hợp sẵn để quản lý cơ sở dữ liệu, bảo mật, xử lý yêu cầu HTTP và tạo giao diện. Hơn nữa, Django còn tuân theo các nguyên tắc tốt nhất như DRY (Don't Repeat Yourself) và mô hình MVT (Model-View- Template), giúp tổ chức mã nguồn gọn gàng và dễ bảo trì. Vì vậy, chọn một framework tốt như Django không chỉ tiết kiệm thời gian mà còn đảm bảo tính ổn định và mở rộng của ứng dụng web.

MySQL được chọn để lưu trữ dữ liệu sản phẩm, khách hàng, và báo cáo, với công cụ MySQL Workbench hỗ trợ trực quan hóa và quản lý cơ sở dữ liệu.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Ngôn ngữ python

2.1.1 Giới thiệu về Python

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.

2.1.2 Ứng dụng

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, đồng 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

2.1.3 Tải và cài đặt Python

Truy cập trang web chính thức của Python tại python.org, vào mục Downloads, nơi website sẽ tự động đề xuất phiên bản phù hợp với hệ điều hành của bạn (Windows hoặc Mac). Tại đây, bạn có thể chọn tải phiên bản Python mong muốn hoặc phiên bản mới nhất được đề xuất, sau đó nhấp vào nút Download để tải về máy.



Hình 1. Trang web download python

Sau khi tải xong, mở tập tin đã tải và làm theo các hướng dẫn trên màn hình để cài đặt. Đảm bảo rằng bạn đã chọn tùy chọn Add Python to PATH

trong quá trình cài đặt, điều này sẽ giúp bạn có thể chạy Python từ dòng lệnh của máy tính.

Để kiểm tra xem Python đã được cài đặt thành công hay chưa, mở Terminal (Linux/macOS) hoặc Command Prompt (Windows), sau đó chạy lệnh: **python -- version** hoặc **python3 --version**

2.2 Framework Django

2.2.1 Khái quát về Django

Django là một framework (khung làm việc) phát triển web mã nguồn mở được viết bằng ngôn ngữ Python. Nó được thiết kế để giúp các nhà phát triển xây dựng các ứng dụng web phức tạp một cách nhanh chóng và bảo mật. Django cung cấp một bộ công cụ và quy tắc sẵn có, giúp bạn tập trung vào việc xây dựng các tính năng độc đáo cho ứng dụng của mình thay vì phải lo lắng về các vấn đề cơ bản của phát triển web.

Django còn nổi bật với triết lý "batteries-included", tích hợp sẵn mọi công cụ cần thiết để xây dựng từ ứng dụng đơn giản đến các hệ thống phức tạp. Đặc biệt, nó đảm bảo bảo mật cao, giảm thiểu các lỗ hổng như SQL Injection hay Cross-Site Scripting (XSS). Với Django, lập trình viên có thể tập trung vào logic ứng dụng mà không cần lo ngại về các chi tiết kỹ thuật phức tạp.

2.2.2 Ưu và nhược điểm của Django

- Ưu điểm:
 - + 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 lớn mạnh: Django có cộng đồng người dùng lớn và tài liệu phong phú, hỗ trợ qua các diễn đàn và website.
- Nhược điểm:
 - + "Batteries included" có thể quá nhiều: Đôi khi, việc có quá nhiều tính năng tích hợp sẵn có thể khiến Django trở nên nặng nề và phức tạp hơn so với các framework nhỏ gọn hơn.
 - + Không phù hợp với các dự án nhỏ: Đối với các dự án nhỏ và đơn giản, việc sử dụng Django có thể hơi quá sức.
 - + Quy ước lớn hơn cấu hình: Django đặt ra nhiều quy ước mà bạn cần tuân theo, điều này có thể gây hạn chế đối với một số nhà phát triển.

2.2.3 Các tính năng cơ bản

Dưới đây là một số tính năng cơ bản và thường sử dụng:

- Admin Interface tự động: Django tự động tạo bảng điều khiển quản trị dựa trên các models trong ứng dụng. Nó cung cấp giao diện thân thiện cho quản trị viên để quản lý dữ liệu mà không cần phải lập trình thủ công.
- Hệ thống quản lý cơ sở dữ liệu với ORM: Django cung cấp một ORM mạnh mẽ giúp chuyển đổi dữ liệu giữa mô hình đối tượng Python và cơ sở dữ liệu. Các lập trình viên có thể tương tác với cơ sở dữ liệu mà không cần viết SQL thủ công, giúp tăng hiệu quả và giảm lỗi.
- Templates: Django có một hệ thống template đơn giản nhưng linh hoạt, cho phép kết hợp HTML với các biến và logic hiển thị một cách dễ dàng.
- URL Routing: Hệ thống định tuyến URL của Django cho phép ánh xạ các URL thân thiện với người dùng đến các chức năng trong ứng dụng mà không cần phụ thuộc vào cấu trúc tệp thực tế.
- Kiến trúc MVC (Model-View-Controller): Django thực chất áp dụng mô hình tương tự, được gọi là MVT (Model-View-Template). Cách tổ chức này giúp mã nguồn dễ bảo trì, tách biệt logic xử lý, dữ liệu và giao diện.
- Authentication và Authorization: Django cung cấp các công cụ mạnh mẽ để xác thực người dùng và kiểm soát quyền truy cập thông qua hệ thống user và groups.

2.2.4 Cài đặt, tạo và chạy dự án

2.2.4.1 Cài đặt Django

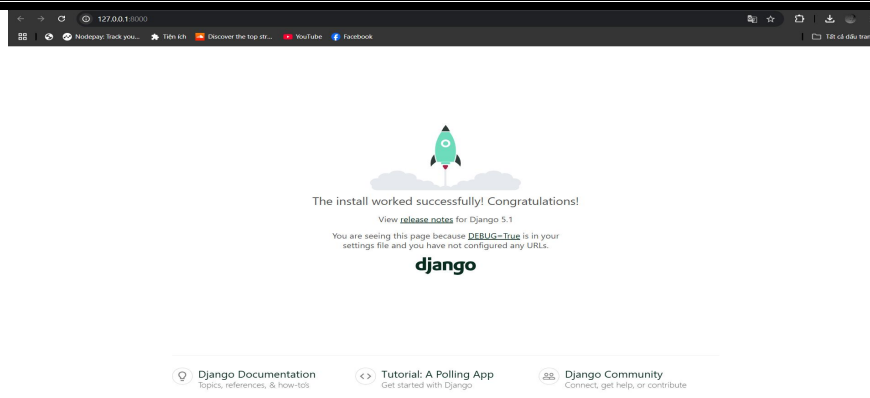
- Để cài đặt Django, ta tiến hành gõ lệnh “pip install django” trong terminal ngay sau khi đã cài đặt xong Python
- Để kiểm tra phiên bản Django đã cài đặt, ta gõ lệnh “python -m django – version”.

2.2.4.2 Tạo dự án

- 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 myproject**
- 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, sử dụng lệnh **python manage.py startapp (tên_ứng_dụng)**. Ví dụ **python manage.py startapp myapp**. Lệnh này sẽ tạo một thư mục mới với cấu trúc thư mục và các tệp cơ bản cần thiết cho ứng dụng.
- 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

2.2.4.3 Chạy server dự án Django

- Sau khi đã thêm ứng dụng vào trong tệp settings.py, bạn cần phải gõ lệnh **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 đề địa chỉ trong terminal để ta click vào (Ctrl + click)
- 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.



Hình 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.

2.2.5 Những thành phần chính có trong Django

2.2.5.1 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**.

2.2.5.2 URLS

- Tập 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).

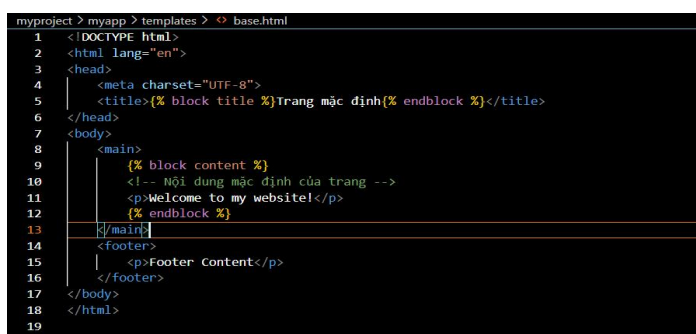
- + Views: Hàm hoặc lớp view, chịu trách nhiệm xử lý logic khi một URL được khớp. Đây là nơi bạn viết mã xử lý dữ liệu, truy vấn cơ sở dữ liệu, và trả về phản hồi HTTP.(Ví dụ views.home)

- + 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, để 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.

Để có thể hiển thị nội dung trang home.html lên trang base.html ta sẽ sử dụng cặp dấu ngoặc {% %} như sau:

- Tạo giao diện trang base và tạo block để truyền templates của giao diện trang home vào.



Hình 3. mã nguồn tạo giao diện trang base

- Sau khi đã tạo giao diện trang base, đặt nội dung cần truyền vào block đã được đặt tên giống với tên của block trong base (block content), lưu ý thêm dòng lệnh {% extends "base.html" %} vào đầu trang.

```
myproject > myapp > templates > < home.html
1 {% extends "base.html" %}
2 {% block title %}
3 Trang chủ
4 {% endblock %}
5 {% block content %}
6 <body>
7 | <h1>Chào mừng đến với trang chủ</h1>
8 </body>
9 </html>
10 {% endblock %}
11
```

Hình 4. Mã nguồn giao diện trang home

2.2.5.3 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.

2.2.5.4 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.

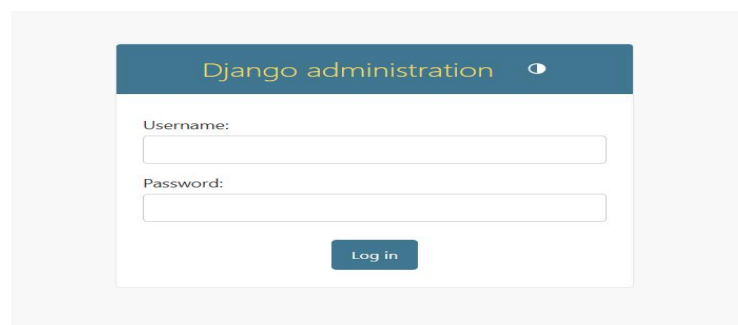
- Một trong những điểm mạnh của Django Admin là khả năng tùy chỉnh linh hoạt. Người dùng có thể thay đổi cách thức hiển thị dữ liệu, thêm các tính năng như tìm kiếm, lọc, phân trang, và nhiều tính năng khác để phù hợp với các nhu cầu cụ thể của từng dự án. Điều này giúp các quản trị viên có thể dễ dàng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

- Để truy cập trang admin của Django, bạn chỉ cần thêm **/admin** vào địa chỉ URL, thường là **http://127.0.0.1:8000/admin** khi chạy trên localhost. Tuy nhiên, trước tiên bạn cần tạo tài khoản quản trị viên bằng lệnh **python manage.py createsuperuser**. Sau khi cung cấp tên người dùng, email và mật khẩu, bạn có thể đăng nhập vào giao diện quản trị và sử dụng Django Admin để quản lý ứng dụng của mình.

```
PS C:\Users\PC MSI\NCKH> python manage.py createsuperuser
Username (leave blank to use 'pcmsi'): hoanghan
Email address: han@gmail.com
Password:
Password (again):
```

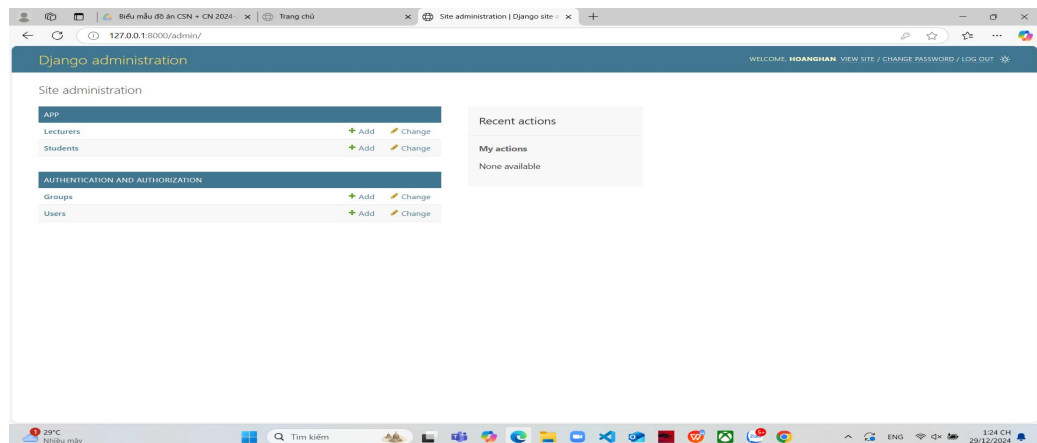
Hình 5. Tạo supperuser

- Sau khi đã tạo thành công tài khoản, truy cập vào ngay trang quản trị admin, đăng nhập tài khoản vừa tạo.



Hình 6. Giao diện đăng nhập trang admin

- Đăng nhập với tài khoản đã tạo, sẽ hiện ra giao diện của trang admin như sau:



Hình 7. Giao diện chính trang admin django

2.3 MySQL

2.3.1 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.

2.3.2 Ưu và nhược điểm của MySQL

- **Ưu điểm:**

- + 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.
- + Hỗ trợ từ cộng đồng: 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:**

- + 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.

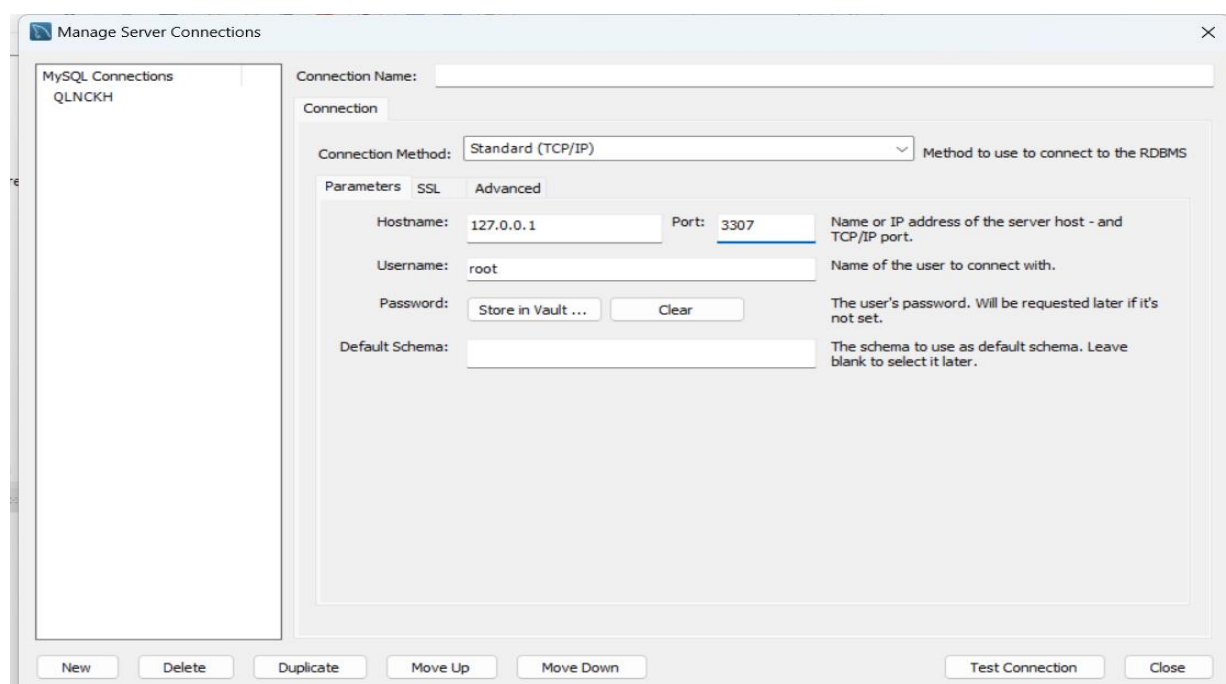
- + Giới hạn chức năng: MySQL không thiết kế để thực hiện toàn bộ các chức năng và nó đi kèm với những hạn chế về chức năng mà một số ứng dụng có thể cần.

2.3.3 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:



Hình 8. Giao diện kết nối cơ sở dữ liệu 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_diemdoan;**

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', #sử dụng mysql  
        'NAME': 'webNCKH', #tên databasedatabase  
        'USER': 'root', #tên người dùng  
        'PASSWORD': 'Nhanh1234@', #mật khẩu  
        'HOST': 'localhost', #địa chỉ máy chủ  
        'PORT': '3307', #cổng  
    }  
}
```

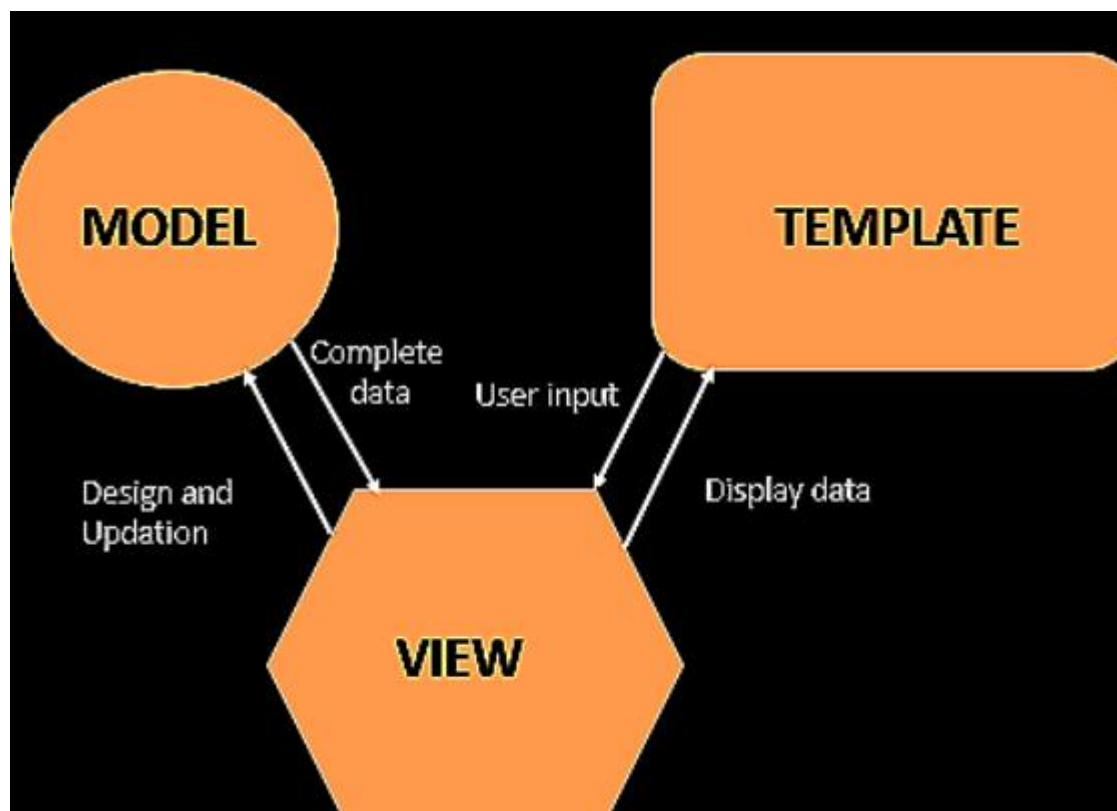
Hình 9. Kết nối cơ sở dữ liệu trong settings.py

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.

2.4 Mô hình Model-View-Template

2.4.1 Sơ lược về mô hình

Mô hình MTV (Model-Template-View): Django sử dụng mô hình này để phân chia logic dữ liệu, giao diện người dùng và logic xử lý trong ứng dụng web. Bộ công cụ tích hợp: Django cung cấp các công cụ tích hợp mạnh mẽ cho việc xử lý đăng nhập người dùng, quản lý admin, tự động tạo giao diện quản trị và nhiều tính năng khác.



Hình 10. Mô hình MTV

Model (M): trong Django được xây dựng trên ORM (Object-Relational Mapping), cho phép lập trình viên tương tác với cơ sở dữ liệu thông qua các đối tượng Python thay vì viết trực tiếp các truy vấn SQL. Điều này giúp cho việc phát triển ứng dụng trở nên dễ dàng và hiệu quả hơn, đồng thời cũng giảm thiểu nguy cơ phát sinh lỗi từ các truy vấn SQL phức tạp.

View (V): trong Django, không giống như controller trong MVC, không chỉ xử lý logic mà còn chịu trách nhiệm phản hồi lại người dùng. Các view có thể là function-based hoặc class-based, tùy thuộc vào yêu cầu và phong cách lập trình của từng nhà phát triển. View lấy dữ liệu từ model, xử lý chúng và gửi chúng đến template để hiển thị.

Template (T): là lớp cuối cùng trong mô hình MVT, nơi định nghĩa cách dữ liệu được trình bày. Templates sử dụng ngôn ngữ đánh dấu (thường là HTML) và có khả năng nhúng mã Python để tạo ra các trang web động. Cơ chế kế thừa template của Django cung cấp khả năng tái sử dụng và quản lý mã hiệu quả, giúp duy trì sự nhất quán trong giao diện người dùng trên toàn bộ ứng dụng.

Bảng 1. Sự khác biệt giữa MVC và MVT

Mô hình MVC	Mô hình MVT	Chức năng
Model	Model	Làm việc với cơ sở dữ liệu
View	Template	Hiển thị giao diện
Controller	View	Xử lý logic

2.4.2 Ưu nhược điểm của mô hình

Ưu điểm:

- **Cấu trúc phân lớp mạch lạc:** Việc phân chia ứng dụng thành ba lớp riêng biệt – Model, View và Template – tạo ra một cấu trúc rõ ràng và dễ quản lý. Mỗi lớp đảm nhận một nhiệm vụ cụ thể, giúp lập trình viên dễ dàng hiểu, sửa đổi và bảo trì từng phần của ứng dụng mà không bị rối bởi sự phức tạp của toàn bộ hệ thống.
- **Quản lý dữ liệu hiệu quả:** Model sẽ tương tác với cơ sở dữ liệu trở nên đơn giản như làm việc với các đối tượng Python. Bạn không cần phải viết SQL, nhờ đó giảm thiểu lỗi và tăng tốc độ phát triển.
- **Khả năng mở rộng và bảo trì linh hoạt:** Khi cần thêm chức năng mới hoặc sửa đổi chức năng hiện có, chỉ cần tập trung vào lớp liên quan mà không cần phải lo lắng về việc ảnh hưởng đến các phần khác.
- **Tối ưu hóa khả năng tái sử dụng mã:** View và Template được thiết kế để có thể tái sử dụng trong nhiều phần khác nhau. Điều này giúp giảm thiểu sự trùng lặp code, tiết kiệm thời gian phát triển và đảm bảo tính nhất quán.

- **Bảo mật được ưu tiên hàng đầu:** Django được tích hợp nhiều tính năng bảo mật mạnh mẽ, giúp bảo vệ ứng dụng khỏi các cuộc tấn công phổ biến như CSRF.

Nhược điểm:

- **Độ khó trong việc tùy biến giao diện phức tạp:** Mặc dù Django cung cấp hệ thống template mạnh mẽ, việc tùy chỉnh giao diện người dùng phức tạp, đặc biệt là khi yêu cầu hiệu ứng động và tương tác người dùng cao.
- **Khả năng tách biệt:** Việc tách biệt các thành phần yêu cầu khả năng hiểu biết chuyên sâu về cả 3 lĩnh vực (Model-Views-Template) để có thể hiệu quả.
- **Hạn chế trong việc xây dựng giao diện người dùng động(SPA):** Django chủ yếu tập trung vào việc xử lý phía server và không được thiết kế tối ưu cho việc xây dựng các ứng dụng một trang với giao diện người dùng động và tương tác cao.
- **Xử lý logic:** Việc đồng bộ và xử lý các logic cũng trở nên phức tạp và khó khăn hơn trong quá trình phát triển.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả bài toán

Bài toán xây dựng hệ thống bán sách bao gồm việc phát triển một nền tảng trực tuyến cho phép người dùng xem, thêm, chỉnh sửa, xóa các thông tin về thời gian nghiên cứu khoa học. Admin cần có khả năng quản lý các tài khoản, và quản lý các chức năng khác giống với người dùng. Mục tiêu của hệ thống là giúp quản lý thời gian nghiên cứu khoa học trực tuyến một cách dễ dàng.

3.2 Đặc tả yêu cầu hệ thống

3.2.1 Yêu cầu chức năng

- Người quản trị (admin):
 - + Quản lý sinh viên: có thể thêm, sửa, xóa thông tin của các sinh viên.
 - + Quản lý giảng viên: có thể thêm, sửa, xóa thông tin của các giảng viên.
 - + Quản lý người dùng: Hệ thống hỗ trợ quản lý tài khoản người dùng có thể thêm, xóa các tài khoản người dùng.
- Người dùng (khách hàng):
 - + Đăng ký và đăng nhập: Người dùng có thể tạo tài khoản cá nhân, đăng nhập để truy cập các tính năng thêm sửa xóa thông tin.

3.2.2 Yêu cầu phi chức năng

Để nói về một website quản lý giờ nghiên cứu khoa học, đầu tiên phải nói đến giao diện, giao diện phải ưa nhìn, rõ ràng. Đảm bảo tính thẩm mỹ và gây ấn tượng tốt với người dùng khi họ vừa truy cập vào website.

Các chức năng phải dễ dàng sử dụng, phù hợp với hầu hết độ tuổi của khách hàng, không nên quá phức tạp sẽ gây khó khăn cho người dùng trong quá trình sử dụng.

3.3 Thiết kế cơ sở dữ liệu trong MySQL Workbench

Bảng thực thể

Bảng 1. Bảng Auth_User (Bản user của Django)

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải	Đặc điểm
1	id	int	Mã người dùng	- Primary Key - Not null - auto_increment
2	password	varchar	Mật khẩu của người dùng	Not null
3	last_login	datetime	Thời gian đăng nhập lần cuối của người dùng	Có thể null
4	is_superuser	tinyint	Xác định người dùng có phải là quản trị viên siêu quyền (1: có quyền, 0: không)	Not null
5	username	varchar	Tên đăng nhập	Not null
6	first_name	varchar	Tên của người dùng	Có thể null
7	last_name	varchar	Họ của người dùng	Có thể null
8	email	varchar	Địa chỉ email	Có thể null
9	is_staff	tinyint	Xác định người dùng có phải là nhân viên quản lý hay không (1: có quyền, 0: không)	Not null
10	is_active	tinyint	Xác định người dùng có còn hoạt động hay không (1: hoạt động, 0: không)	Not null
11	date_joined	datetime	Thời gian người dùng tham gia hệ thống	auto_now_add=True

Bảng 2. Bảng Student

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải	Đặc điểm
1	student_id	varchar	Mã số sinh viên	- Primary Key - Not null - auto_increment
2	full_name	varchar	Tên sinh viên	Not null
3	class_code	varchar	Mã lớp	Not null
4	time_spent	int	Tổng thời gian nghiên cứu khoa học	Có thể null

Bảng 3. Bảng Lecturer

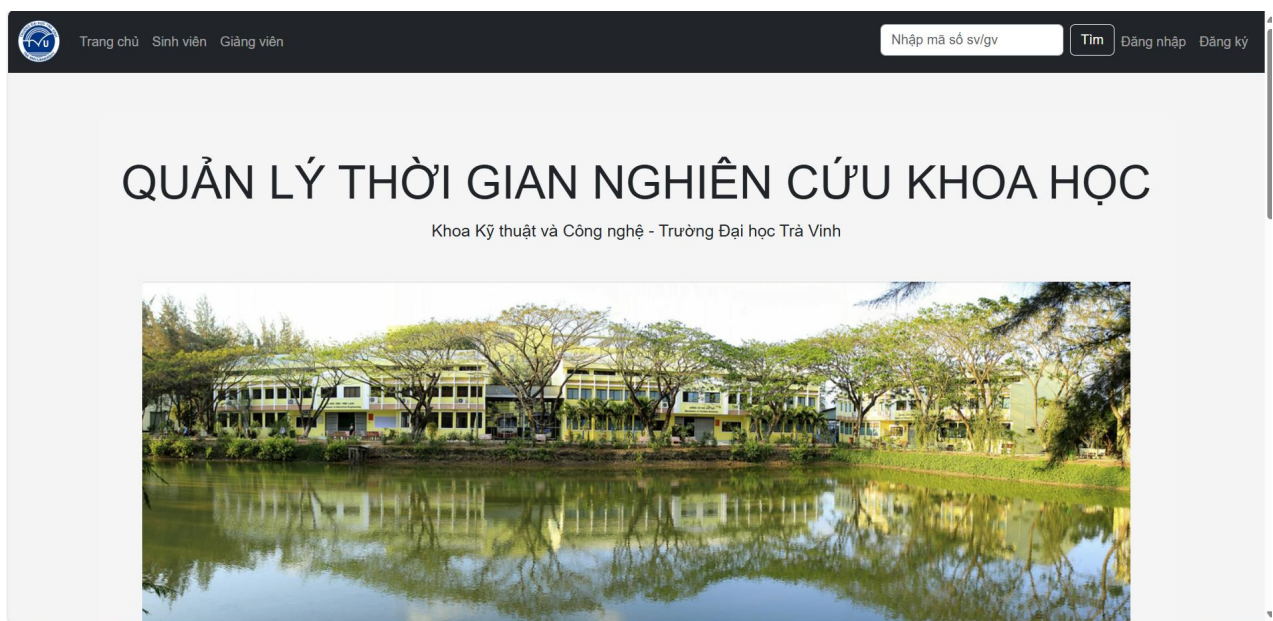
STT	Thuộc tính	Kiểu dữ liệu	Diễn giải	Đặc điểm
1	lecturer_id	varchar	Mã số giảng viên	- Primary Key - Not null - auto_increment
2	full_name	varchar	Tên giảng viên	Not null
4	time_spent	int	Tổng thời gian nghiên cứu khoa học	Có thể null

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

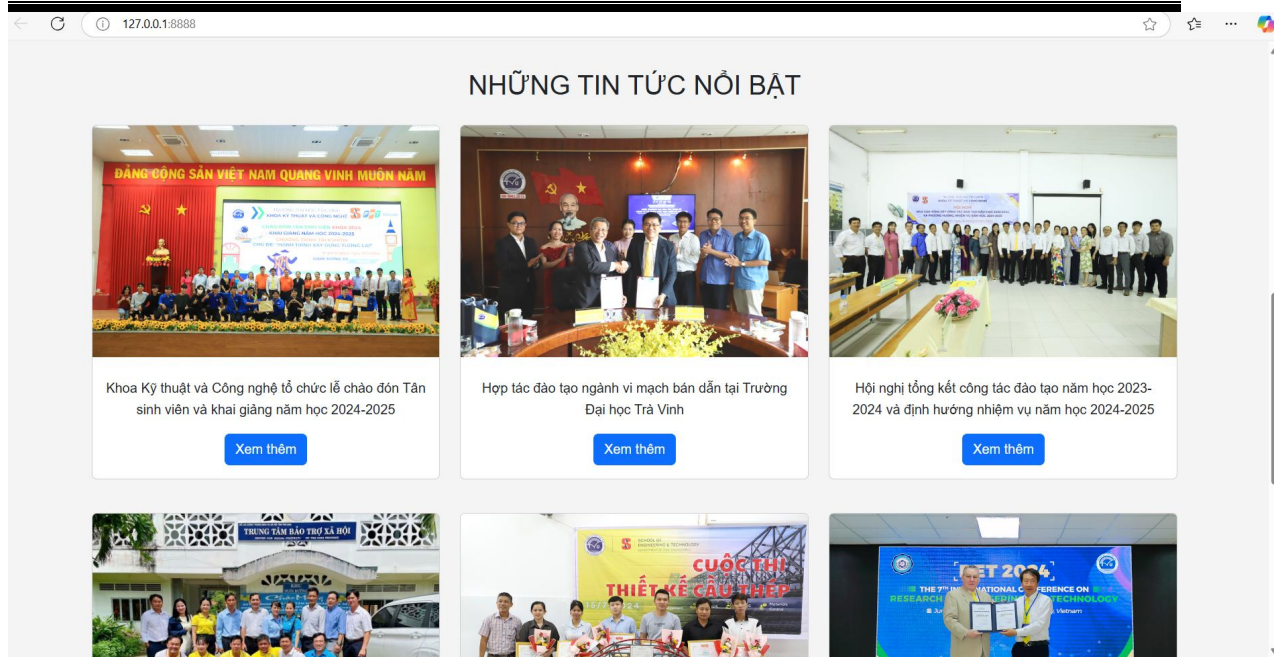
4.1 Giao diện dành cho người dùng

4.1.1 Giao diện trang chủ

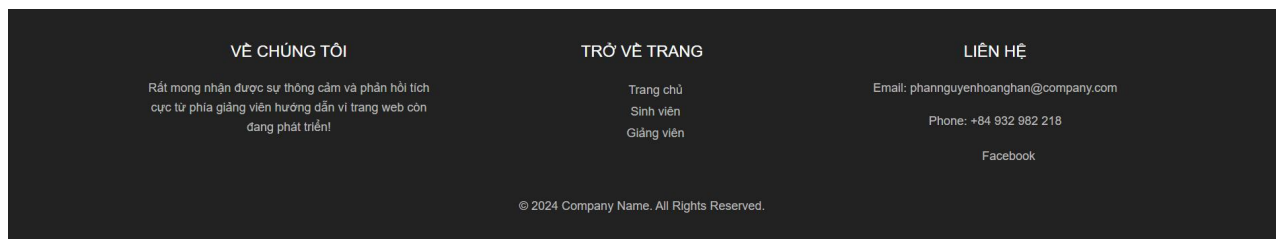
Trang chủ là trang đầu tiên hiển thị khi người dùng truy cập vào website. Giao diện trang chủ bao gồm thanh menu ngang, nội dung trang, footer. Các tin tức nổi bật của TVU - SET Giao diện tương đối dễ sử dụng và gọn gàng.



Hình 11. Giao diện đầu trang chủ



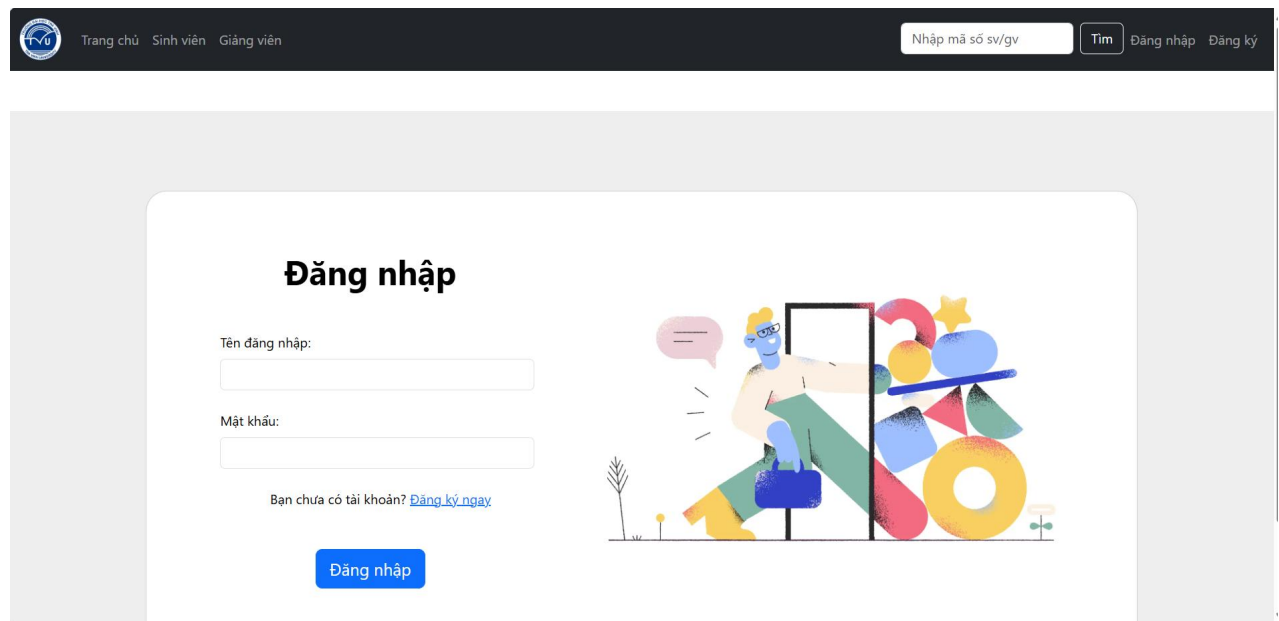
Hình 12. Giao diện giữa trang chủ



Hình 13. Footer

4.1.2 Giao diện trang đăng nhập

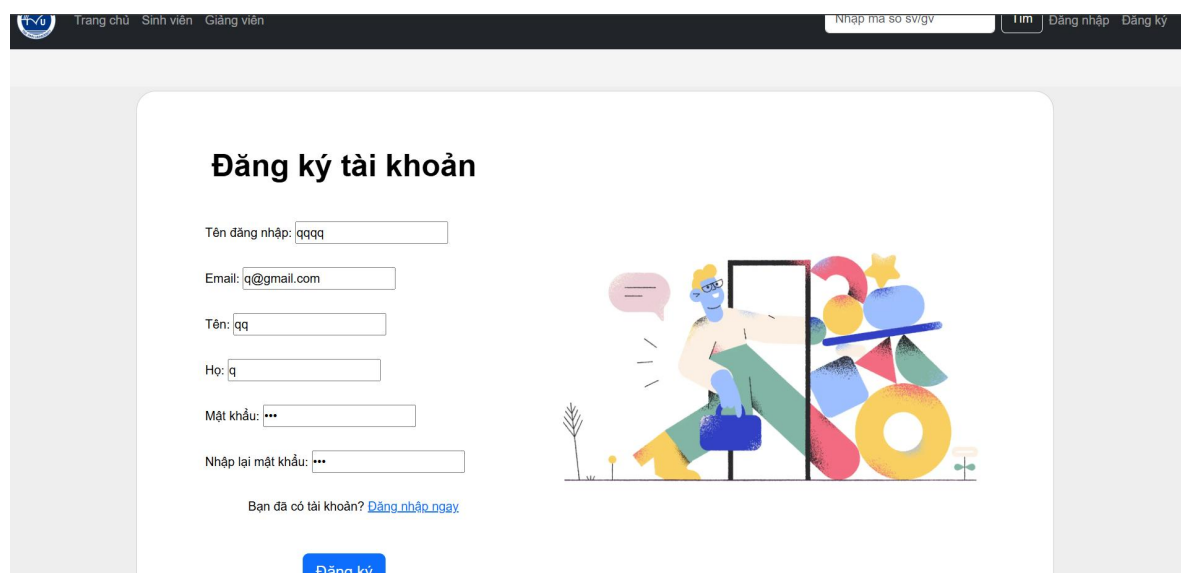
Khi nhấn vào “Đăng nhập” trên thanh menu thì sẽ dẫn tới trang đăng nhập của trang web, tại đây người dùng có thể đăng nhập với tài khoản đã đăng ký mới có thể thực hiện chức năng thêm, sửa, xóa các thông tin. Nếu chưa có tài khoản thì người dùng có thể chọn vào link đăng ký trên form đăng nhập hoặc trên thanh menu để đăng ký tài khoản mới.



Hình 14. Giao diện trang đăng nhập

4.1.3 Giao diện trang đăng ký

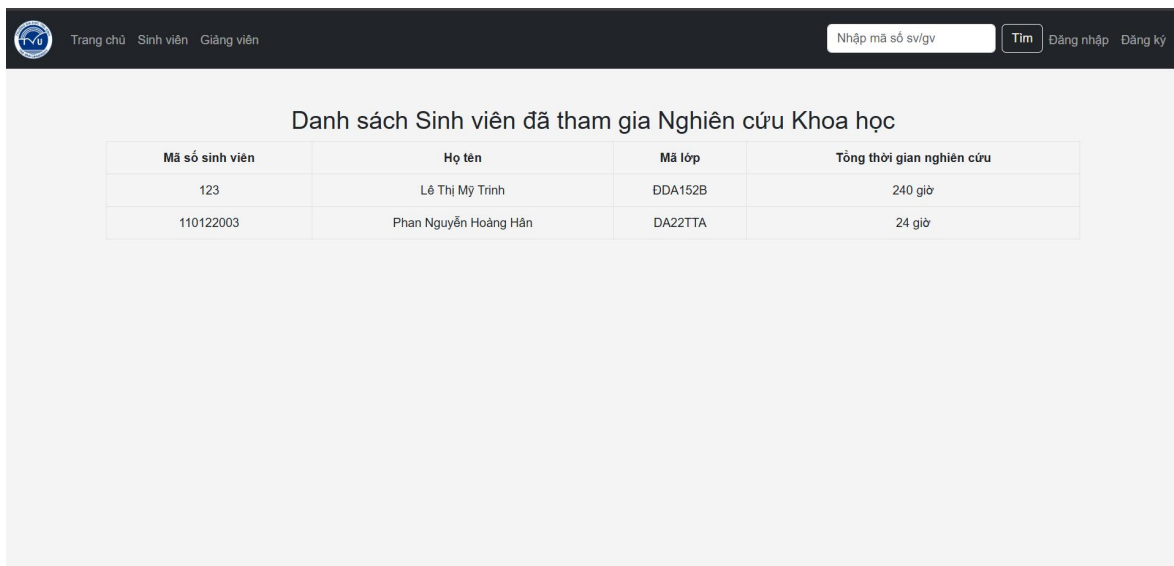
Khi người dùng nhấn chọn vào link đăng ký trên form đăng nhập hoặc trên thanh menu sẽ dẫn đến trang đăng ký. Người dùng có thể nhập các thông tin như tên người dùng, email, tên, họ, mật khẩu và nhấn vào nút đăng kí để tiến hành đăng ký.



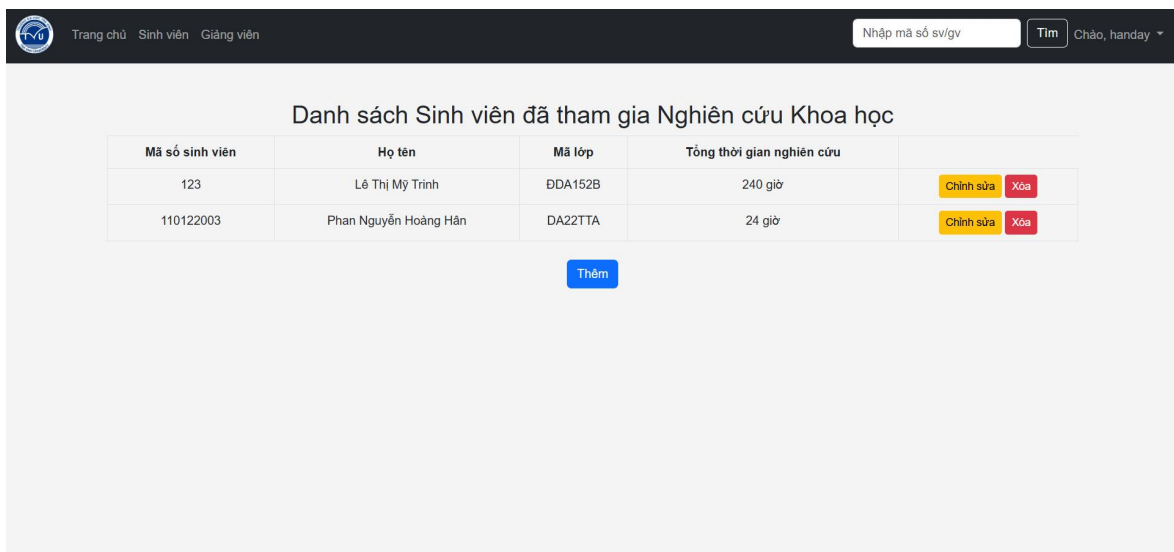
Hình 15. Giao diện trang đăng ký

4.1.4 Giao diện trang Sinh viên

Đây là giao diện hiển thị thông tin của các sinh viên đã tham gia nghiên cứu khoa học. Chỉ khi đã đăng nhập người dùng mới có thể thực hiện các chức năng thêm, sửa, xóa các thông tin.



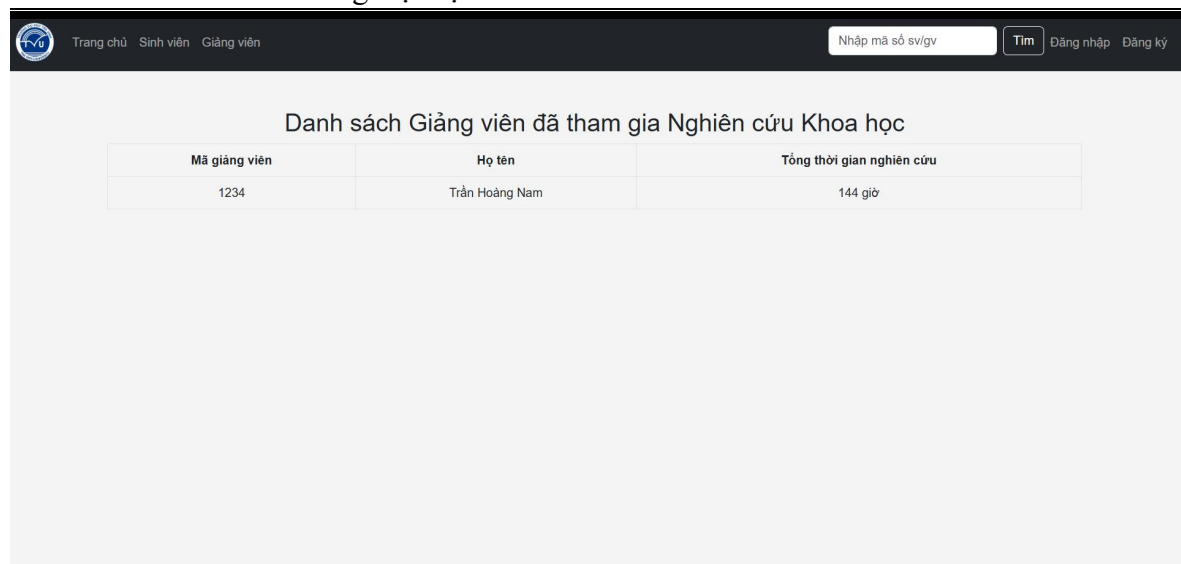
Hình 16. Giao diện trang Sinh viên chưa đăng nhập



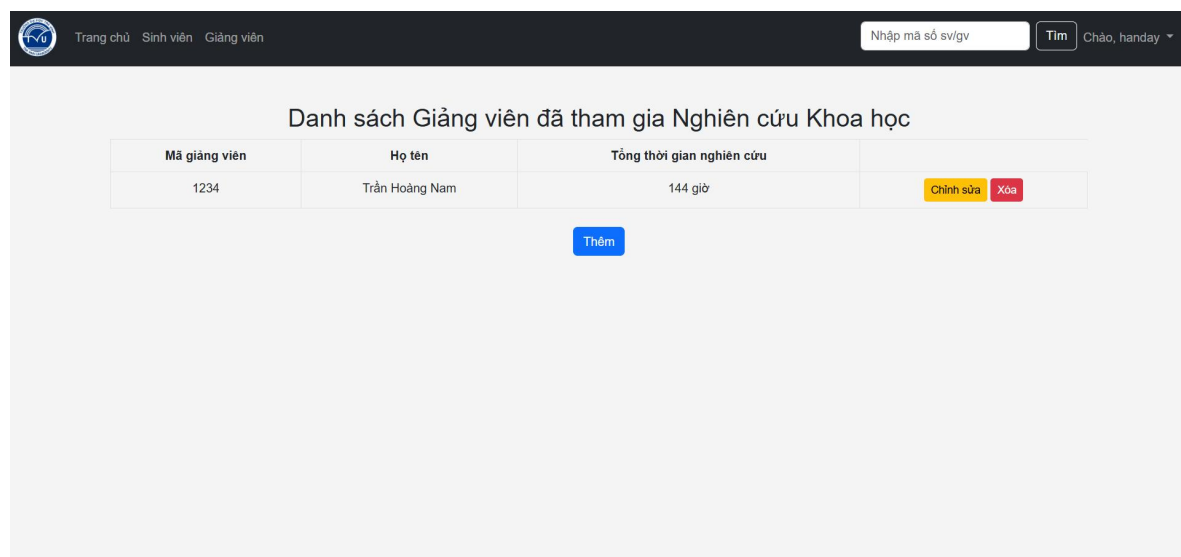
Hình 17. Giao diện trang Sinh viên đã đăng nhập

4.1.5 Giao diện trang Giảng viên

Đây là giao diện hiển thị thông tin của các sinh viên đã tham gia nghiên cứu khoa học. Chỉ khi đã đăng nhập người dùng mới có thể thực hiện các chức năng thêm, sửa, xóa các thông tin.



Hình 18. Giao diện trang Giảng viên chưa đăng nhập



Hình 19. Giao diện trang Giảng viên đã đăng nhập

4.1.6 Giao diện trang tìm kiếm

Người dùng có thể tìm kiếm sinh viên hoặc giảng viên theo mã số của họ, giúp người dùng có thể tìm kiếm thông tin một cách dễ dàng. Khi click vào nút “Tìm” trang web sẽ hiển thị sinh viên và giảng viên có thông tin liên quan đến mã số tìm kiếm.

Mã sinh viên	Họ tên	Mã lớp	Số giờ
110122003	Phan Nguyễn Hoàng Hân	DA22TTA	24

Hình 20. Giao diện trang tìm kiếm

4.2 Giao diện dành cho người quản trị

Để truy cập vào trang quản trị cần thêm /admin vào cuối địa chỉ của trang web, thường là <http://127.0.0.1:8888/admin>.

4.2.1 Trang chủ quản trị

Là trang đầu tiên khi truy cập vào trang quản trị, gồm các models đã tạo như Category, Product, Order, OrderItem, User và Group được django tự động tạo. Phía bên phải là các hành động quản trị gần đây.

Site administration

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#) [Change](#)

Users [+ Add](#) [Change](#)

CORE

Lecturers [+ Add](#) [Change](#)

Students [+ Add](#) [Change](#)

Recent actions

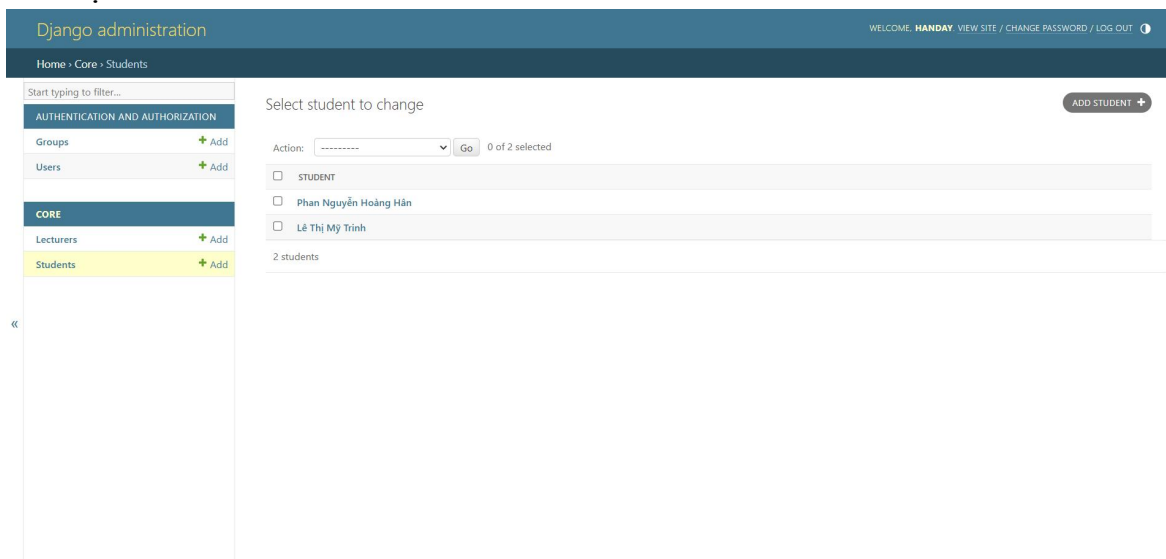
My actions

- ✖ hanne User
- ✖ hann User
- ✖ han12 User
- ✖ han User
- ✖ Phan Nguyễn Hoàng Hân Lecturer
- ✖ han Lecturer
- ✖ hanne User
- ✖ hanhan User
- ✖ han User

Hình 21. Giao diện trang admin

4.2.2 Danh sách danh mục

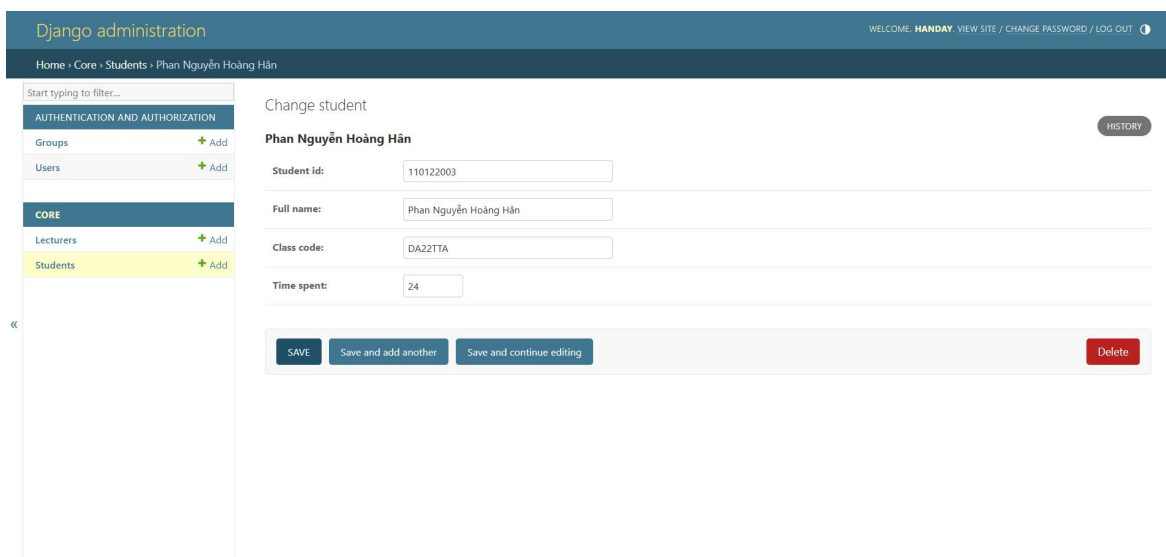
Chứa tất cả danh mục và bấm vào danh mục nào sẽ hiện các thông tin của danh mục đó.



Hình 22. Giao diện hiển thị danh sách các danh mục

4.2.3 Thêm, sửa, xóa thông tin

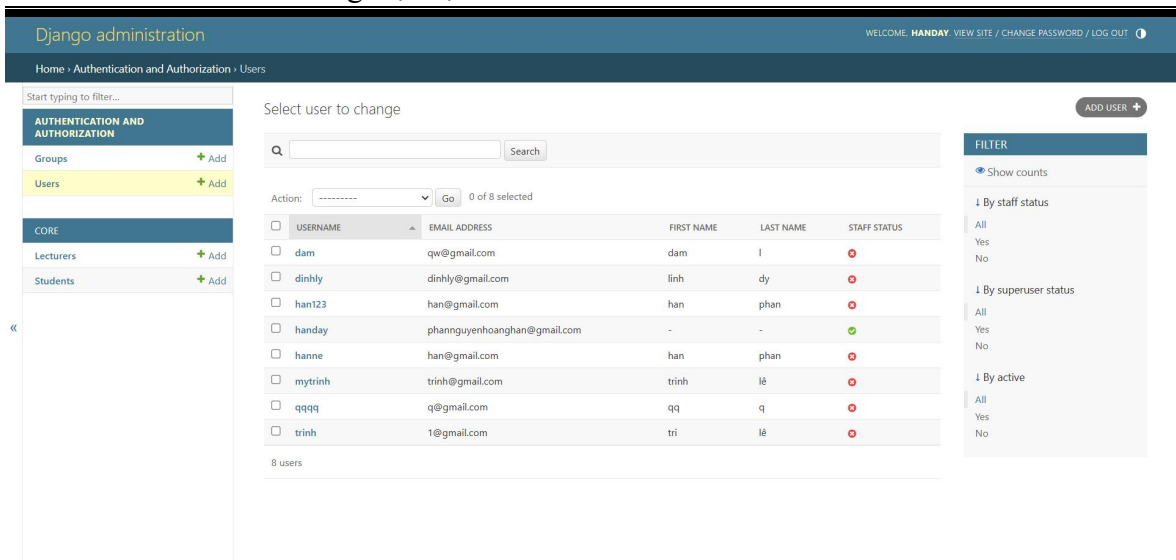
Người quản trị có thể thêm, sửa thông tin của giảng viên hoặc giảng viên nào đó. Nếu muốn xóa có thể click vào Delete để xóa. Khi một giảng viên hoặc sinh viên được xóa thì nó cũng tự được xóa trong phần quản lý danh mục của nó.



Hình 23. Giao diện trang chỉnh sửa thông tin sinh viên hoặc giảng viên

4.2.4 Quản lý người dùng

Người quản trị có thể thêm, sửa, xóa thông tin của người dùng.



The screenshot displays the Django administration interface for user management. The top navigation bar shows the user is logged in as 'HANDAY'. The breadcrumb trail indicates the current location: Home > Authentication and Authorization > Users. The left sidebar contains navigation links for 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users) and 'CORE' (Lecturers, Students). The main content area is titled 'Select user to change' and features a search bar and a table of users. The table has columns for USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. The right sidebar contains filter options for staff status and superuser status.

USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
dam	qw@gmail.com	dam	l	○
dinhly	dinhly@gmail.com	linh	dy	○
han123	han@gmail.com	han	phan	○
handay	phannguyenhoangan@gmail.com	-	-	●
hanne	han@gmail.com	han	phan	○
mytrinh	trinh@gmail.com	trinh	le	○
qqqq	q@gmail.com	qq	q	○
trinh	1@gmail.com	tri	le	○

Hình 24. Giao diện trang quản lý người dùng

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được

“Tìm hiểu ngôn ngữ lập trình python xây dựng hệ thống quản lý giờ nghiên cứu khoa học của khoa KT&CN - Trường Đại học Trà Vinh” là một đề tài rộng mở theo hướng quản lý các hồ sơ của môi trường học tập. Qua quãng thời gian thực hiện đề tài, em đã hoàn thành cơ bản yêu cầu của bài toán với các chức năng thêm, sửa, xóa, tìm hiểu được các mẹo kiến thức thú vị trong lĩnh vực coder. Đồng thời, qua quá trình tìm hiểu em cũng đã tiếp xúc sâu hơn với trang web hiện hành của Trường Đại học Trà Vinh nhất là trang web của khoa Kỹ Thuật và Công nghệ.

Quan trọng nhất, đề tài đã giúp em rèn luyện khả năng lập trình, sử dụng framework Django để thực hiện đề tài, khả năng tìm kiếm dữ liệu, xử lý thông tin và một số kỹ năng khác cũng được cải thiện đáng kể.

5.2 Hạn chế và hướng phát triển

5.2.1 Hạn chế

Qua đề tài, ngoài những ảnh hưởng cũng như lợi ích tích cực đã đạt được, em cũng nhận thấy nhiều hạn chế cần được khắc phục.

Giao diện trang web chỉ ở mức trung bình, không có điểm nhấn, màu sắc mờ nhạt, chưa hấp dẫn người dùng.

Các chức năng còn sơ sài, chỉ có các chức năng cơ bản như đăng nhập, đăng ký, thêm, sửa, xóa các thông tin. Các chức năng chưa có tính logic chặt chẽ, chưa đủ đáp ứng chức năng quản lý giờ nghiên cứu khoa học của đề tài.

Khả năng quản lý, sắp xếp thời gian kém, thụ động dẫn ảnh hưởng đến quá trình thực hiện đề tài.

5.2.2 Hướng phát triển

Xây dựng hoàn thành giao diện đẹp mắt, hấp dẫn hơn theo xu hướng của hiện nay.

Phát triển thêm các chức năng cần có của một trang web quản lý giờ nghiên cứu khoa học thực thụ. Thêm các tính năng quản lý hồ sơ, các sự kiện liên quan đến nghiên cứu khoa học.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] "File title or description," Google Drive, Accessed: Jan. 7, 2025. [Online]. Available: <https://drive.google.com/file/d/1QZyDXqv1mrpK9YjVjhTY7TFyIpVnUS5/view>
- [2] "Django - A Comprehensive Guide," Sach.chat, Accessed: Jan. 7, 2025. [Online]. Available: <https://www.sach.chat/books/django-1688048590>
- [3] "Black Python," Sach.chat, Accessed: Jan. 7, 2025. [Online]. Available: <https://www.sach.chat/books/black-python-1619490217>
- [4] Hưng Phạm, "Báo cáo môn Python Django," Studocu, (2023). [Trực tuyến]. Có tại: <https://www.studocu.vn/vn/document/truong-dai-hoc-cong-nghe-thong-tin-va-truyen-thong/phan-tich-thiet-ke-huong-doi-tuong/bao-cao-mon-python-django/92330996>
- [5] Cao Lê Viết Tiến, "Django là gì? Hướng dẫn cài đặt Django dành cho người mới," Vietnix, (03/07/2024). [Trực tuyến]. Có tại: <https://vietnix.vn/django-la-gi/>.