**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**

--------🙠🕮🙢--------



**BÁO CÁO ĐỒ ÁN CHUYÊN NGÀNH**

**ĐỀ TÀI: XÂY DỰNG ỨNG DỤNG QUẢN LÝ**

**NGHIÊN CỨU KHOA HỌC SINH VIÊN**

Giảng viên hướng dẫn: ThS. Trần Thị Phương Linh

Sinh viên thực hiện: 2011382 – Trần Trung Hiếu

2015749 – Nguyễn Hoàng Nhật Tiến

2015840 – Nguyễn Ngọc Minh Tiến

**Đà Lạt, tháng 06 năm 2023**

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Đà Lạt, ngày ….. tháng …… năm ……

Giảng viên hướng dẫn

[Ký tên và ghi rõ họ tên]

LỜI CẢM ƠN

Đầu tiên, nhóm chúng em xin được gửi lời cám ơn đến Ban chủ nhiệm Khoa Công nghệ thông tin trường Đại học Đà Lạt cùng toàn thể quý thầy cô đã tạo điều kiện, giúp đỡ chúng em thực hiện và hoàn thành chương trình Đồ án chuyên ngành.

Khoảng thời gian thực hiện Đề tài học phần Đồ án chuyên ngành là khoảng thời gian quý giá đối với mỗi cá nhân chúng em nói riêng và các bạn sinh viên lớp CTK44B nói chung. Nhóm chúng em chân thành gửi lời cảm ơn đến cô hướng dẫn **Cô Trần Thị Phương Linh** – Giảng viên khoa Công nghệ Thông tin trường Đại học Đà Lạt, người đã đồng hành cùng chúng em trong suốt quá trình thực hiện Đồ án. Một lần nữa, nhóm em xin chân thành cảm ơn và chúc cô thật nhiều sức khoẻ.

Trong suốt quá trình tham gia và phát triển đề tài, nhóm em không tránh khỏi mắc phải những sai sót dẫn đến giảm hiệu quả công việc và chậm tiến độ mục tiêu đề ra. Tuy nhiên nhờ sự giúp đỡ của cô hướng dẫn và phía bạn bè trong lớp đã tiếp thêm động lực cho chúng em hoàn thành tốt các công việc được giao.

Vì kiến thức chuyên môn còn hạn chế và bản thân còn nhiều thiếu sót kinh nghiệm nên nội dung báo cáo không tránh khỏi những thiếu sót, chúng em rất mong nhận sự góp ý, chỉ bảo thêm của quý thầy cô.

Một lần nữa, xin chân thành gửi lời cảm ơn đến tất cả thầy cô và tất cả bạn bè đã tận tình giúp đỡ, góp ý cho chúng em hoàn thành học phần Đồ án chuyên ngành này.

Nhóm chúng em xin chân thành cảm ơn!

Đà Lạt, Tháng 6 năm 2023

**Trường Đại học Đà Lạt**

**Khoa Công nghệ Thông Tin**

---🙜🙞**---**

# **ĐỀ CƯƠNG THỰC HIỆN ĐỒ ÁN**

**Tên đề tài: Xây dựng ứng dụng quản lý nghiên cứu khoa học Sinh viên**

**Sinh viên thực hiện:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Lớp** | **Email liên hệ** |
| 1 | Nguyễn Hoàng Nhật Tiến | 2015749 | CTK44B | 2015749@dlu.edu.vn |
| 2 | Trần Trung Hiếu | 2011382 | CTK44B | 2011382@dlu.edu.vn |
| 3 | Nguyễn Ngọc Minh Tiến | 2015840 | CTK44B | 2015840@dlu.edu.vn |

**Giáo viên hướng dẫn:** Cô Trần Thị Phương Linh

1. **Mục tiêu đề tài**

* Đề tài giúp sinh viên vận dụng các kiến thức về ReactJs và ASP.NET để xây dựng ứng dụng web quản lý NCKH sinh viên.
* Nâng cao kĩ năng cá nhân: Kĩ năng đọc tài liệu tiếng Anh, ebook, học tiếng Anh và phát triễn kĩ năng làm việc nhóm.

1. **Nội dung đề tài**

* Tổng quan các kiến thức cơ bản về ReactJs và ASP.NET Core.
* Nghiên cứu, tham khảo về các ứng dụng quản lý NCKH sinh viên.
* Thiết kế giao diện và cơ sở dữ liệu.
* Xây dựng các chức năng của ứng dụng (đăng ký, đăng nhập, xem các bài NCKH của sinh viên, …).
* Phát triển và triển khai ứng dụng.

1. **Phần mềm và công cụ sử dụng**

* Công nghệ sử dụng: ReactJs và ASP.NET Core
* Công cụ sử dụng:
* Visual Studio 2022 / Visual Studio Code
* Git/Github
* SQL Server

1. **Dự kiến kết quả đạt được­**

* Có được kĩ năng làm việc nhóm.
* Có thêm vốn từ tiếng Anh.
* Nắm rõ và sử dụng được ReactJs và ASP.NET Core để xây dựng ứng dụng.
* Xây dựng được ứng dụng web hoàn chỉnh.

1. **Kế hoạch thực hiện**

**Giai đoạn 1:** 03/03/2023 – 21/03/2023

- Viết đề cương đồ án

- Ôn tập các kiến thức về HTML, CSS, JavaScript

- Tiến hành tìm hiểu về ReactJs và ASP.NET

- Thực hiện viết báo cáo đồ án

**Giai đoạn 2:** 21/03/2023 – 27/05/2023

- Xây dựng ứng dụng web quản lý NCKH sinh viên

**Giai đoạn 3:** 28/05/2023 – 22/06/2023

- Hoàn thiện ứng dụng

- Hoàn thiện viết báo cáo đồ án

1. **Tài liệu tham khảo**

**• ReactJs:**

- <https://reactjs.org/docs/getting-started.html>

- [Ebook] React JS – Pro React (Cássio de Sousa Antonio)

- [Ebook] React JS – ReactJS by Example – Building Modern Web Applications with React (Packt)

- [Ebook] React JS – Fullstack React (Anthony Accomazzo, Ari Lerner, Clay Allsopp, David Guttman, Tyler Mcginnis, Nate Murray: 2020)

- [Ebook] React JS – Lập trình React thật đơn giản (VNTALKING)

- [Ebook] <https://drive.google.com/drive/folders/1AkQOVpxtRyPcppivQNqJnHxCfgEudCYV>

**• ASP.NET:**

- <https://dotnet.microsoft.com/en-us/learn/aspnet>

- <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-7.0&tabs=visual-studio>

- <https://tedu.com.vn/series/hoc-aspnet-core-can-ban.html>

Đà Lạt, ngày 13 tháng 03 năm 2023­

**Giáo viên hướng dẫn Sinh viên thực hiện**

Nguyễn Hoàng Nhật Tiến

Trần Trung Hiếu

Trần Thị Phương Linh Nguyễn Ngọc Minh Tiến

**BCN Khoa Tổ trưởng Bộ môn**

# 

Mục lục

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN 2](#_Toc138451405)

[LỜI CẢM ƠN 3](#_Toc138451406)

[ĐỀ CƯƠNG THỰC HIỆN ĐỒ ÁN 4](#_Toc138451407)

[CHƯƠNG 1 – TỔNG QUAN ĐỀ TÀI 9](#_Toc138451408)

[1.1. Giới thiệu đề tài 9](#_Toc138451409)

[1.2. Các tính năng của ứng dụng 9](#_Toc138451410)

[1.3. Lí do chọn đề tài 10](#_Toc138451411)

[1.4. Mục đích của đề tài 10](#_Toc138451412)

[1.5. Phương pháp và phạm vi nghiên cứu 11](#_Toc138451413)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 12](#_Toc138451414)

[2.1. ReactJs và các ngôn ngữ liên quan 12](#_Toc138451415)

[2.1.1. HTML 12](#_Toc138451416)

[2.1.2. CSS 13](#_Toc138451417)

[2.1.3. JavaScript 14](#_Toc138451418)

[2.1.4. NodeJs 14](#_Toc138451419)

[2.1.5. ReactJs 15](#_Toc138451420)

[2.2. API và ASP.NET Core 17](#_Toc138451421)

[2.2.1. API 17](#_Toc138451422)

[2.2.2. ASP.NET Core 18](#_Toc138451423)

[2.3. SQL Server 20](#_Toc138451424)

[CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH 23](#_Toc138451425)

[3.1. Hướng dẫn cài đặt 23](#_Toc138451426)

[3.2. Xây dựng Cơ sở dữ liệu 27](#_Toc138451427)

[3.3. Xây dựng API 30](#_Toc138451428)

[3.4. Xây dựng giao diện 35](#_Toc138451429)

[3.4.1. Trang chủ 35](#_Toc138451430)

[3.4.2. Giao diện danh sách các khoa 36](#_Toc138451431)

[3.4.3. Giao diện sinh viên nghiên cứu 39](#_Toc138451432)

[3.4.4. Giao diện danh sách giảng viên 40](#_Toc138451433)

[3.4.5. Giao diện liên hệ admin 41](#_Toc138451434)

[3.4.6. Giao diện đăng nhập, đăng ký 42](#_Toc138451435)

[3.4.7. Giao diện sinh viên 44](#_Toc138451436)

[3.4.8. Giao diện Admin 48](#_Toc138451437)

[Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 56](#_Toc138451438)

[4.1. Kết luận 56](#_Toc138451439)

[4.2. Những mặt hạn chế khi xây dựng ứng dụng 56](#_Toc138451440)

[4.3. Hướng phát triển 56](#_Toc138451441)

[TÀI LIỆU THAM KHẢO 57](#_Toc138451442)

CHƯƠNG 1 – TỔNG QUAN ĐỀ TÀI

1. Giới thiệu đề tài

* Ngày nay, tin học ngày càng phát triển với quy mô rộng lớn và chất lượng cao hơn, tối ưu hơn, hiện đại hơn. Công nghệ thông tin được sử dụng rất nhiều vào các ngành khoa học kỹ thuật. Một trong những ứng dụng của nó là việc áp dụng vào các công tác quản lý các hoạt động đời sống kinh tế xã hội.
* Với những lợi ích do công nghệ thông tin mang lại, các nhà quản lý đã kịp thời đưa ra những ứng dụng tin học vào công tác quản lý nghiên cứu khoa học sinh viên.
* Bài toán “Quản lý nghiên cứu khoa học sinh viên” nhằm giải quyết và đáp ứng các nhu cầu về mặt quản lý nghiên cứu khoa học trong trường đại học. Tin học hóa trong công tác quản lý nhằm giảm bớt sức lao động của con người, tiết kiệm thời gian, gọn nhẹ và tiện hơn rất nhiều so với làm thủ tục.

1. Các tính năng của ứng dụng

* Phía người dùng
* Hiển thị các đề tài mà sinh viên đã thực hiện và đã được nghiệm thu.
* Tìm kiếm đề tài theo tên
* Hiển thị các khoa kèm theo sinh viên, giảng viên, đề tài của khoa đó
* Xem chi tiết của khoa, sinh viên, giảng viên, đề tài
* Xem danh sách các sinh viên thực hiện và giảng viên hướng dẫn
* Sinh viên đăng ký, đăng nhập để có thể đăng ký được đề tài.
* Cho phép sinh viên đăng ký đề tài và tải file (pdf, word) lên khi thực hiện đề tài xong.
* Sinh viên, giảng viên có thể chỉnh sửa thông tin cá nhân.
* Phía Admin
* Hiển thị thông tin dashboard hệ thống
* Phân quyền đăng nhập
* Quản lý đề tài: Thêm mới đề tài, Sửa thông tin đề tài, Xóa đề tài, Phân công giảng viên hướng dẫn
* Quản lý tài khoản sinh viên, giảng viên
* Quản lý các Khoa: Thêm khoa, Sửa thông tin khoa, và xóa khoa

1. Lí do chọn đề tài

* Hiện nay, đào tạo hệ đại học đang được Đảng và Nhà nước quan tâm chú ý đến nhằm nâng cao chất lượng cử nhân, kỹ sư… ngang tầm khu vực.Việc đào tạo cũng như học tập và nghiên cứu khoa học của sinh viên được nhà trường quan tâm khuyến khích và tạo điều kiện thuận lợi. Mục tiêu là hoạt động nghiên cứu khoa học trở thành một hoạt động thường xuyên và hiệu quả nhằm nâng cao chất lượng đào tạo. Để thực hiện tốt điều này, công tác quản lý và hỗ trợ giáo viên và sinh viên đối với việc nghiên cứu khoa học cần phải được hoàn thiện.
* Công tác quản lý là công việc phức tạp và tốn nhiều công sức, việc quản lý trong hệ thống trường học còn gặp không ít khó khăn. Bên cạnh đó, việc tra cứu thông tin về các đề tài khoa học cũng như các tài liệu báo cáo khoa học đã có rất mất nhiều thời gian và công sức.Những hạn chế này đã không khuyến khích và gây nhiều khó khăn cho sinh viên tham gia nghiên cứu khoa học. Đặc biệt trong thời đại 4.0, các áp dụng khoa học kỹ thuật và công nghệ thông tin trở nên đặc biệt hữu ích đối với mọi lĩnh vực trong cuộc sống thì việc xây dựng một hệ thống nhằm quản lý và hỗ trợ công tác nghiên cứu khoa học cho sinh viên là hết sức cần thiết.

1. Mục đích của đề tài

* Đề tài giúp chúng em ôn lại các kiến thức về HTML, CSS, JavaScript, C#
* Đề tài giúp sinh viên vận dụng các kiến thức về ReactJs và ASP.NET để xây dựng ứng dụng web quản lý NCKH sinh viên.
* Nâng cao kĩ năng cá nhân: Kĩ năng đọc tài liệu tiếng Anh, ebook, học tiếng Anh và phát triễn kĩ năng làm việc nhóm.
* Cũng như củng cố lại các kiến thức đã học từ môn Phát triển ứng dụng Web nâng cao, chúng em sử dụng các công nghệ đã học để xây dựng ứng dụng quản lý nghiên cứu khoa học sinh viên
* Xây dựng ứng dụng có thể quản lý các đề tài nghiên cứu khoa học của sinh viên bao gồm các chức năng như: tìm kiếm đề tài, đăng ký đề tài, …

1. Phương pháp và phạm vi nghiên cứu

* Về phương pháp tiếp cận đề tài, Cô đã yêu cầu gặp mặt, trao đổi và phân tích giúp nhóm em hiểu rõ hơn về vấn đề. Cô đã hỗ trợ cho chúng em các phương pháp và kỹ thuật phù hợp với dự án đang tìm hiểu để có thể đi đúng hướng hơn. Ngoài ra Cô cũng gửi cho chúng em các tài liệu liên quan đến đề tài để cho nhóm chúng em dễ dàng nghiên cứu, xây dựng ứng dụng.
* Đối với phạm vi nghiên cứu, nhóm nghiên cứu xây dựng về nghiên cứu dựa trên những kiến thức đã học và tham khảo thêm từ nhiều nguồn tài liệu khác nhau. Sử dụng các kiến thức và công nghệ hiện đại hiện nay là ReactJs và ASP.NET Core. Để quản lý mã nguồn dễ dàng, nhóm sử dụng GitHub.

**Từ khóa:** ASP.NET Core, ReactJs, cách xây dựng website, API

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1. ReactJs và các ngôn ngữ liên quan**

* + 1. **HTML**

1. **Giới thiệu**

* HTML hay HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản, là ngôn ngữ được sử dụng cho các tài liệu web.
* Nhưng HTML không phải là một ngôn ngữ lập trình, HTML giống như một ngôn ngữ xác định đâu là ý nghĩa, mục đích và cấu trúc của một tài liệu. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các website.

1. **Vai trò của HTML**

* Một website thường chứa nhiều trang con. Mỗi trang con sẽ có một tập tin HTML riêng. Dù bạn lập trình bằng bất cứ ngôn ngữ nào, trên bất cứ Framework nào, khi chạy trên nền website, chúng đều được biên dịch ra ngôn ngữ HTML. Đa số các trình soạn thảo văn bản trên website đều có 2 chế độ xem: HTML và văn bản thường.
* Chức năng chính của HTML là xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc. HTML thường được dùng để phân chia các đoạn văn, heading, links, blockquotes,…
* Với HTML, bạn sẽ có thể:
* Thêm tiêu đề, định dạng đoạn văn, ngắt dòng điều khiển.
* Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh, tạo liên kết.
* Xây dựng bảng, điều khiển một số kiểu mẫu
* Như đã đề cập ở trên, HTML không phải là ngôn ngữ lập trình. Do đó, HTML cũng không thể tạo ra các chức năng “động” được. Hiểu đơn giản, HTML giống như Microsoft Word, dùng để bố cục và định dạng trang web.
* Tóm lại, HTML là ngôn ngữ markup, dễ học, dễ hiểu, dễ áp dụng. Tuy nhiên, một website được viết bằng HTML rất đơn giản, nếu không muốn nói là nhàm chán. Để gây hứng thú với người truy cập, website cần có sự hỗ trợ của CSS và JavaScript.
* Nếu HTML cung cấp cấu trúc cơ bản của các trang web thì CSS và JavaScript sẽ cải tiến và sửa đổi website sinh động hơn.
  + 1. **CSS**

1. **Giới thiệu**

* CSS là viết tắt của cụm từ “Cascading Style Sheets”, tạm dịch: ngôn ngữ tạo phong cách cho trang web. Ngôn ngữ lập trình này quy định cách các thành phần HTML của trang web thực sự sẽ xuất hiện trên frontend như thế nào.
* Hiểu đơn giản, CSS sẽ giúp webmaster xác định styles và định nghĩa nhiều loại nội dung của website. CSS được tạo ra để kết hợp với ngôn ngữ markup HTML để tạo phong cách cho trang web.
* Có 3 loại style CSS chính:
* Style CSS Internal: là style được tải lên mỗi khi trang web được refresh.
* Style CSS Inline: bạn có thể chỉnh sửa một yếu tố nào đó mà không cần truy cập trực tiếp vào file CSS.
* External style: bạn có thể tạo phong cách ở file khác áp dụng CSS vào trang bạn muốn. External style sẽ cải thiện thời gian tải trang rất nhiều.
* Đặc biệt, bạn có thể sử dụng External Style CSS để tạo phong cách cho nhiều trang cùng lúc.

1. **Vai trò của CSS**

* Do đó, sử dụng CSS, bạn có thể:
* Tạo phong cách và định kiểu cho những yếu tố được viết dưới dạng ngôn ngữ đánh dấu, như HTML.
* Tiết kiệm công sức của lập trình viên nhờ điều khiển định dạng của nhiều trang web
* Phân biệt cách hiển thị của trang web với nội dung chính của trang bằng cách điều khiển bố cục, màu sắc và font chữ.
* Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một trang web thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn.
* Ví dụ: HTML sẽ đánh dấu từng phần văn bản để biết được đó là yếu tố gì. CSS sẽ sử dụng cấu trúc tiếng Anh đơn giản để tạo ra một bộ các quy tắc tạo phong cách cho các yếu tố.
* Có thể nói, CSS gần như tạo nên bộ mặt của một website. Và CSS cũng không phải là tất cả. Để có một trang web đẹp bạn không chỉ dựa vào CSS mà phải kết hợp với nhiều ngôn ngữ khác. Các ngôn ngữ phải được thực hiện dựa trên các bản thiết kế đã thống nhất.
  + 1. **JavaScript**

1. **Giới thiệu**

* Thường được viết tắt là “JS”, JavaScript là một ngôn ngữ lập trình được Brendan Eich (đồng sáng lập dự án Mozilla, quỹ Mozilla và tập đoàn Mozilla) cho ra mắt vào năm 1995 với tên LiveScript.
* JavaScript được biết đến đầu tiên với tên Mocha, và sau đó là LiveScript, nhưng công ty Netscape đã đổi tên của nó thành JavaScript, bởi vì sự phổ biến như là một hiện tượng của Java lúc bấy giờ. Các slideshow, pop-up quảng cáo và tính năng autocomplete của Google đều được viết bằng JavaScript.
* JS có tác dụng giúp chuyển website từ trạng thái tĩnh sang động, tạo tương tác để cải thiện hiệu suất máy chủ và nâng cao trải nghiệm người dùng. Hiểu đơn giản, JavaScript là ngôn ngữ được sử dụng rộng rãi khi kết hợp với HTML/CSS để thiết kế web động.

1. **Vai trò của JavaScript**

* Sử dụng JavaScript, bạn sẽ:
* Dễ dàng bắt đầu với các bước nhỏ, với thư viện ảnh, bố cục có tính thay đổi … nhờ sự linh hoạt của JavaScript.
* Có thể tạo ra các trò chơi, hoạt họa 2D hoặc 3D, ứng dụng cơ sở dữ liệu toàn diện, …
* Tăng cường các hành vi và kiểm soát mặc định của trình duyệt.
* JavaScript là ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế các trang web tương tác. Hầu hết các hành vi động mà bạn sẽ thấy trên một trang web đều có được là nhờ JavaScript. Vì thế, đây là một ngôn ngữ phức tạp và khó học.
  + 1. **NodeJs**

1. **Giới thiệu**

* NodeJS là một môi trường runtime chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

1. **Ưu và nhược điểm**

* **Ưu điểm**
* IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
* Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
* Chia sẻ cùng code ở cả phía client và server.
* NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
* Cộng đồng hỗ trợ tích cực.
* Cho phép stream các file có kích thước lớn.
* **Nhược điểm**
* Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
* Khó thao tác với cơ sử dữ liệu quan hệ.
* Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
* Cần có kiến thức tốt về JavaScript.
* Không phù hợp với các tác vụ đòi hỏi nhiều CPU.
  + 1. **ReactJs**

1. **Giới thiệu**

* ReactJS là một thư viện trong đó có chứa nhiều JavaScript mã nguồn mở và cha đẻ của ReactJS đó chính là Facebook. Mục đích của việc tạo ra ReactJS là để tạo ra những ứng dụng website hấp dẫn với tốc độ nhanh và hiệu quả cao với những mã hóa tối thiểu. Và mục đích chủ chốt của ReactJS đó chính là mỗi website khi đã sử dụng ReactJS thì phải chạy thật mượt thật nhanh và có khả năng mở rộng cao và đơn giản thực hiện.
* React (còn được gọi là Reactjs hay React.js) là một Thư viện javascript được tạo ra bởi sự cộng tác giữa Facebook và Instagram. Nó cho phép những nhà phát triển web tạo ra giao diện người dung nhanh chóng. Phần Views của Reactjs thường được hiển thị bằng việc chủ yếu dung các component mà chứa các component cụ thể hoặc các thẻ HTML. Một trong những đặc trưng duy nhất của Reactjs là việc render dữ liệu không những có thể thực hiện ở tầng server mà còn ở tầng client.
* Nó cũng sử dụng khái niệm là Virtual DOM (DOM ảo). Virtual DOM tạo ra bản cache cấu trúc dữ liệu của ứng dụng trên bộ nhớ. Sau đó, ở mỗi vòng lặp, nó liệt kê những thay đổi và sau đó là cập nhật lại sự thay đổi trên DOM của trình duyệt một cách hiệu quả. Điều này cho phép ta viết các đoạn code như thể toàn bộ trang được render lại dù thực tế là Reactjs chỉ render những component hay subcomponent nào thực sự thay đổi.

1. **Đặc tính của ReactJs**

* Tư tưởng ReactJS là xây dựng lên các components có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp chúng ta dễ dàng quản lý, mở rộng hệ thống.
* Luôn giữ các components ở trạng thái stateless (nhiều nhất có thể) khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang static HTML. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiện thị ra dựa vào các đầu vào đó, điều này lý giải tại sao nó lại mang tính tái sử dụng (reuse) và dễ dàng cho việc test.

1. **Thế mạnh của ReactJs**

* Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể them vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
* Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.

1. **Cài đặt ReactJs**

* Đầu tiên các bạn cần cài đặt Node.js để tạo môi trường chạy React, việc cài đặt rất đơn giản và nhanh chóng.
* Sử dụng Create React App: Create React App là một môi trường tiện lợi để học React, và đó là cách tốt nhất để xây dựng một single-page app với React.
* Nó thiết lập môi trường phát triển để bạn có thể sử dụng các tính năng Javascript mới nhất, cung cấp những trải nghiệm tốt cho nhà phát triển và tối ưu hoá app của bạn để deploy lên product. Máy bạn cần phải cài đặt Nodejs, npm và yêu cầu phiên bản NodeJS >= 8.10 và npm >= 5.6.
* Để tạo project, chạy lệnh sau: npx create-react-app my-app; cd my-app; npm start
* Bây giờ bạn có thể xem website trên trình duyệt tại đường dẫn <http://localhost:3000>
  1. **API và ASP.NET Core**
     1. **API**

1. **Giới thiệu**

* API là viết tắt của Application Programming Interface – phương thức trung gian kết nối các ứng dụng và thư viện khác nhau.
* Nó cung cấp khả năng truy xuất đến một tập các hàm hay dùng, từ đó có thể trao đổi dữ liệu giữa các ứng dụng.
* Thi thoảng vẫn có người lầm tưởng API là một ngôn ngữ lập trình nhưng thực ra, API chỉ là các hàm hay thủ tục thông thường. Các hàm này được viết trên nhiều ngôn ngữ lập trình khác nhau.

1. **4 đặc điểm nổi bật của API**

* API sử dụng mã nguồn mở, dùng được với mọi client hỗ trợ XML, JSON.
* API có khả năng đáp ứng đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content forma…. Bạn có thể sử dụng các host nằm trong phần ứng dụng hoặc trên IIS.
* Mô hình web API dùng để hỗ trợ MVC như: unit test, injection, ioc container, model binder, action result, filter, routing, controller. Ngoài ra, nó cũng hỗ trợ RESTful đầy đủ các phương thức như: GET, POST, PUT, DELETE các dữ liệu.
* Được đánh giá là một trong những kiểu kiến trúc hỗ trợ tốt nhất với các thiết bị có lượng băng thông bị giới hạn như smartphone, tablet…

1. **Ưu và nhược điểm**

* **Ưu điểm**
* Giao tiếp hai chiều phải được xác nhận trong các giao dịch sử dụng API. Cũng chính vì vậy mà các thông tin rất đáng tin cậy.
* API là công cụ mã nguồn mở, có thể kết nối mọi lúc nhờ vào Internet.
* Hỗ trợ chức năng RESTful một cách đầy đủ.
* Cấu hình đơn giản khi được so sánh với WCF (Window Communication Foundation). Cung cấp cấp trải nghiệm thân thiện với người dùng.
* **Nhược điểm**
* Tốn nhiều chi phí phát triển, vận hành, chỉnh sửa.
* Đòi hỏi kiến thức chuyên sâu.
* Có thể gặp vấn đề bảo mật khi bị tấn công hệ thống.

1. **Ứng dụng của API**

* Web API: Là hệ thống API được sử dụng trong các hệ thống website, chẳng hạn: Google, Facebook… Hầu hết các website đều cung cấp hệ thống API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Đa số Web API được thiết kế theo tiêu chuẩn RESTful.
* API trên hệ điều hành: Windows hay Linux có rất nhiều API. Họ cung cấp các tài liệu API là đặc tả các hàm, phương thức cũng như các giao thức kết nối. Nó giúp lập trình viên có thể tạo ra các phần mềm ứng dụng có thể tương tác trực tiếp với hệ điều hành.
* API của thư viện phần mềm (framework): API mô tả và quy định các hành động mong muốn mà các thư viện cung cấp. Một API có thể có nhiều cách triển khai khác nhau, giúp cho một chương trình viết bằng ngôn ngữ này có thể sử dụng được thư viện viết bằng ngôn ngữ khác.
  + 1. **ASP.NET Core**

1. **Giới thiệu**

* ASP.NET Core là một web framework mã nguồn và được tối ưu hóa cho cloud để phát triển các ứng dụng web chạy trên nhiều nền tảng như Windows, Linux và Mac. Hiện tại, nó bao gồm MVC framework được kết hợp các tính năng của MVC và Web API thành một web framework duy nhất.
* Các ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên .NET Framework hoàn chỉnh.
* Nó đã được thiết kế để cung cấp một framework tối ưu cho các ứng dụng để triển khai tới cloud hoặc chạy on-premises.
* Nó bao gồm những modular với các thành phần tối thiểu, do đó bạn giữ được tính linh hoạt trong quá trình xây dựng các giải pháp của mình.
* Bạn có thể phát triển và chạy các ứng dụng đa nền tảng từ ASP.NET Core trên Windows, Mac và Linux.

1. **Lịch sử**

* ASP.NET đã được sử dụng từ nhiều năm để phát triển các ứng dụng web. Kể từ đó, framework này đã trải qua một sự thay đổi tiến hóa đều đặn và cuối cùng đã đưa chúng ta đến với ASP.NET Core 1.0 hậu duệ mới gần đây nhất của nó.
* ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET 4.6
* Đó là một framework hoàn toàn mới, may mắn rằng nó một dự án side-by-side tương tự với mọi thứ mà chúng ta biết.
* Nó thực ra được viết lại trên framework ASP.NET 4.6 hiện tại những kích thước nhỏ hơn và nhiều modular hơn.

1. **Ưu điểm**

* ASP.NET Core đi kèm với những ưu điểm sau:
* ASP.NET Core có một số thay đổi kiến trúc dẫn đến modular framework nhỏ hơn.
* ASP.NET Core không còn dựa trên System.Web.dll. Nó dựa trên một tập hợp nhiều yếu tố của Nuget packages.
* Điều này cho phép bạn tối ưu ứng dụng của mình chỉ cần những NuGet packages cần thiết.
* Lợi ích của diện tích bề mặt ứng dụng nhỏ hơn thì bảo mật chặt chẽ hơn, giảm dịch vụ, cải thiện hiệu suất và giảm chi phí.
* Với ASP.NET Core, bạn có thể nhận được các cải tiến sau:
* Xây dựng và chạy các ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.
* Được xây dựng trên .NET Core, hỗ trợ side-by-side app versioning.
* Công cụ mới giúp đơn giản hóa việc phát triển web hiện đại.
* Liên kết đơn các web stack như Web UI và API Web.
* Cấu hình dựa trên môi trường đám mây sẵn có.
* Được xây dựng dựa trên cho DI (Dependency Injection).
* Tag Helpers làm cho các Razor makup trở nên tự nhiên hơn với HTML.
* Có khả năng host trên IIS hoặc self-host.
  1. **SQL Server**

1. **Giới thiệu**

* SQL server hay còn được gọi là Microsoft SQL Server, nó từ viết tắt của MS SQL Server. Đây chính là một loại phần mềm đã được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ cho những dữ liệu dựa theo tiêu chuẩn RDBMS.
* Từ đó, người ta sẽ lưu trữ dữ liệu dựa vào tiêu chuẩn RDBMS và nó cũng là một trong những hệ quản trị cơ sở dữ liệu dạng quan hệ đối tượng.
* SQL Server có khả năng cung cấp đầy đủ các công cụ cho việc quản lý từ giao diện GUI đến sử dụng ngôn ngữ cho việc truy vấn SQL. Điểm mạnh của SQL điểm mạnh của nó là có nhiều nền tảng được kết hợp cùng như: ASP.NET, C# để xây dựng Winform cũng chính nó có khả năng hoạt động độc lập. Tuy nhiên, SQL Server thường đi kèm với việc thực hiện riêng các ngôn ngữ SQL, T-SQL,...
* Cụ thể như sau:
* T-SQL là một trong những loại ngôn ngữ thuộc quyền sở hữu của Microsoft và được gọi với cái tên Transact-SQL. Nó thường cung cấp thêm rất nhiều cho các khả năng khai báo biến, thủ tục lưu trữ và xử lý ngoại lệ,...
* SQL Server Management Studio là một loại công cụ giao diện chính cho máy chủ cơ sở của chính dữ liệu SQL, thông thường thì nó hỗ trợ cho cả môi trường 64 bit và 32 bit.

1. **Những thành phần của SQL Server**

* SQL Server đã trải qua hơn 20 năm phát triển và đã đề ra những version cụ thể khác nhau. Các mô hình Client - Server được chia làm 2 thành phần chính bao gồm:
* Workstation: Nó được cài đặt trên các thiết bị vận hành để trở thành phần mềm tương tác với hệ thống máy chủ Server.
* Server: Được cài đặt ở trên máy chủ chính, nó có thể là: SQL server, SQL Server Agent, SSIS, SSAS,...
* Ngoài ra, bạn hoàn toàn có thể cài đặt nhiều phiên bản của SQL Server trên cùng một máy chủ và điều này sẽ giúp tiết kiệm được các chi phí mua Server để hoạt động và cần nhiều phiên bản khác nhau. Nó bảo mật và cũng được tách biệt hoàn toàn giúp cho hệ thống được an toàn hơn.

1. **Mục đích khi sử dụng SQL Server**

* SQL Server thông thường được sử dụng cho mục đích lưu trữ dữ liệu. Ngoài ra, nó còn mang lại những tính năng làm việc giúp người dùng làm việc hiệu quả hơn như sau:
* Giúp người sử dụng có thể duy trì việc lưu trữ bền vững.
* Cho phép bạn tạo ra nhiều cơ sở dữ liệu hơn.
* Có khả năng phân tích dữ liệu bằng SSAS
* Nó có khả năng bảo mật cao
* Việc tạo ra được các báo cáo bằng SSRS — SQL Server Reporting Services sẽ được dễ dàng hơn.
* Các quá trình sẽ được thực hiện bằng SSIS — SQL Server Integration Services.

1. **Một số khái niệm cơ bản về các xử lý**

* Thông thường, các xử lý bên trong một ứng dụng thường có thể sẽ được chia thành 2 loại xử lý, bao gồm: xử lý trên máy chủ và xử lý trên trạm.
* Nó có khả năng đọc và cập nhật được các dữ liệu.
* Quá trình tính toán, hiển thị dữ liệu ở trên màn hình giao diện.
* Bạn có thể sử dụng được nhiều loại ngôn ngữ lập trình khác nhau như là: C#, Java,...
* Xử lý các tiêu chuẩn trên máy chủ Database Server
* Xử lý những yêu cầu liên quan đến ghi/đọc dữ liệu.
* Thực hiện quản lý đồng bộ mọi dữ liệu giữa những yêu cầu đọc ghi từ các máy trạm gửi tới.
* Thực hiện các dịch vụ quản trị dữ liệu tự động dựa theo định kỳ như backup/restore dữ liệu.

**CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH**

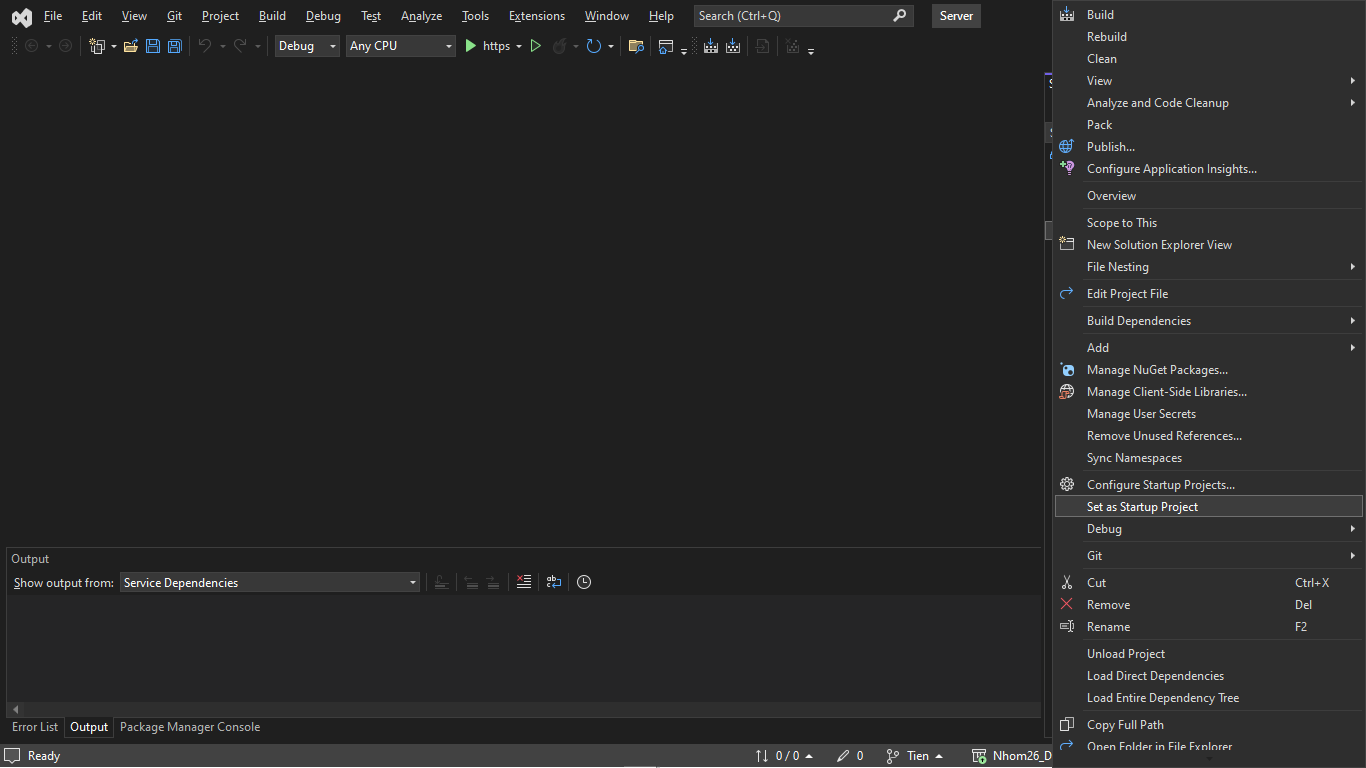
## **3.1. Hướng dẫn cài đặt**

Yêu cầu: Đã cài đặt Visual Studio 2022, Visual Studio Code, SQL Server và NodeJs

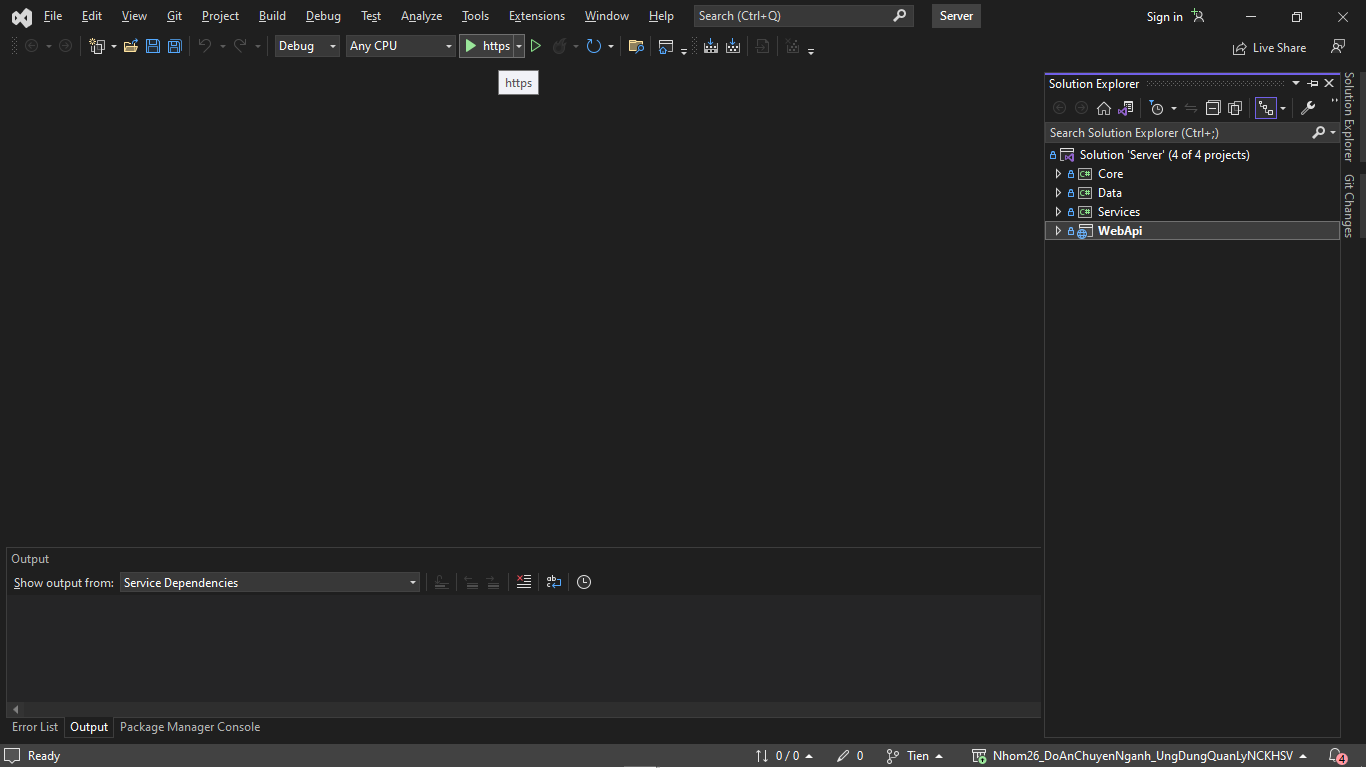
Bắt đầu vào src -> back-end -> Server, khởi chạy Server.sln



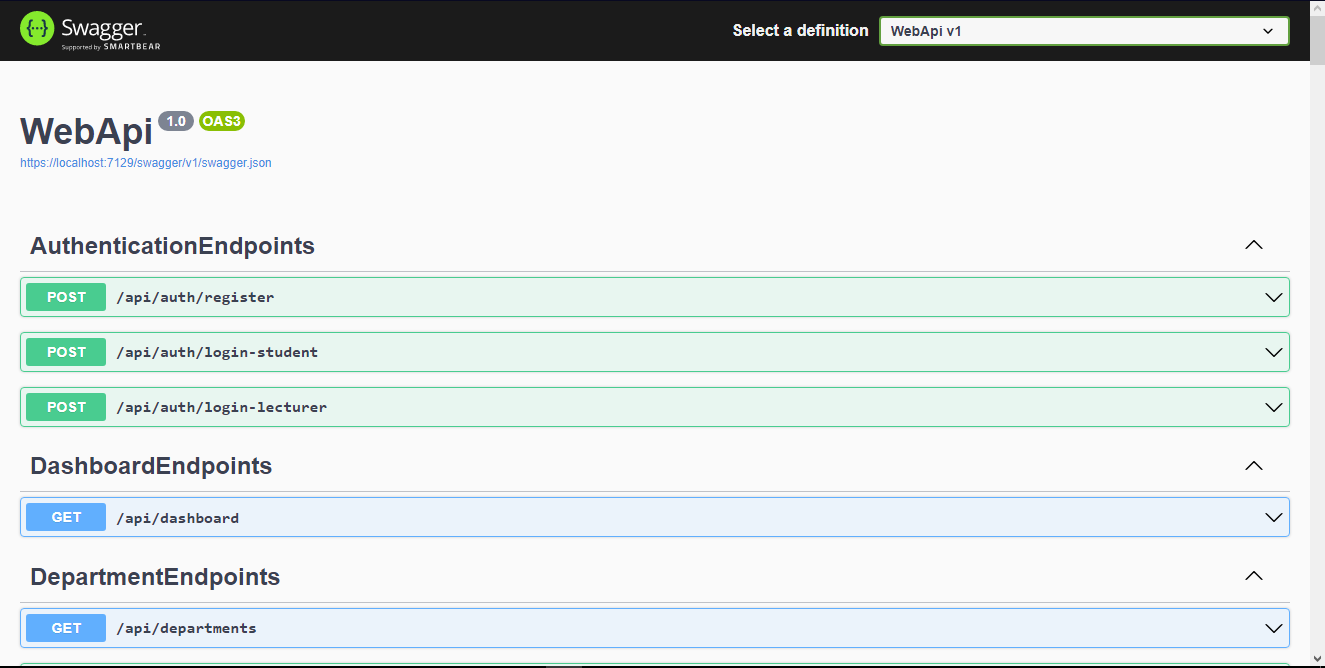
Chuột phải vào WebApi -> chọn Set as Startup Projects …



Sau đó chạy chương trình bằng cách ấn vào mũi tên có chữ https hoặc là F5



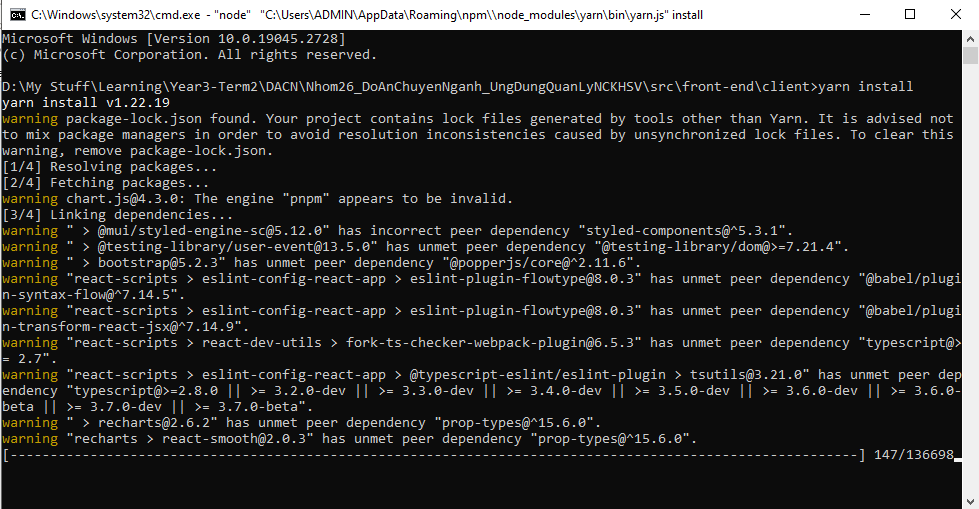
Giao diện API sẽ hiển thị



Sau đó vào thư mục src-> front-end -> client

Tại thanh đường dẫn nhập vào cmd 

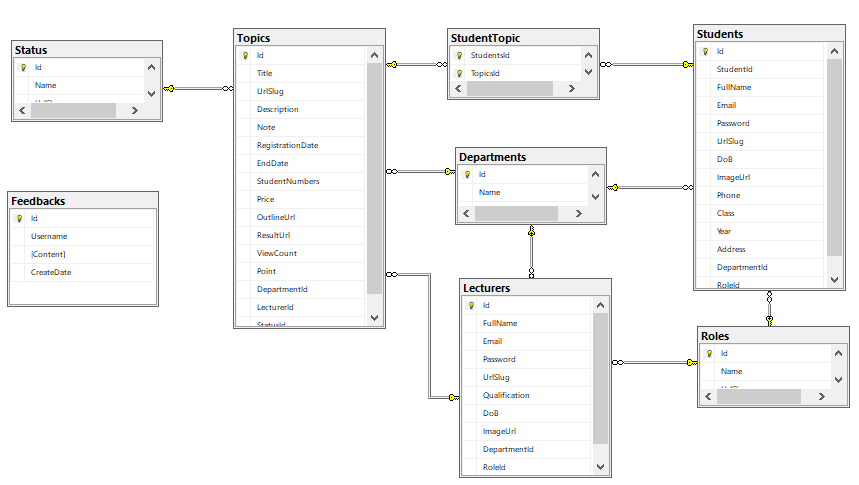
Sau đó thực hiện lệnh yarn install

 Sau khi cài xong sẽ có thư mục node-modules 

Tiếp tục tại cmd gõ lệnh yarn start để khởi động client

****

## **3.2. Xây dựng Cơ sở dữ liệu**

****

Các bảng có trong cơ sở dữ liệu:

**Topics:** Chứa các dữ liệu về các đề tài

|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| ID | Mã đề tài |
| Title | Tên đề tài |
| UrlSlug | Mã định danh của đề tài |
| Description | Mô tả về đề tài |
| Note | Ghi chú về đề tài |
| RegistrationDate | Ngày bắt đầu để đăng ký đề tài |
| EndDate | Ngày nghiệm thu đề tài |
| StudentNumbers | Số lượng thành viên thực hiện đề tài |
| Price | Chi phí của đề tài |
| OutlineUrl |  |
| ResultUrl | Chứa file PDF kết quả của đề tài |
| ViewCount | Lượt xem của đề tài |
| Point | Điểm của đề tài |
| DepartmentId | (Khóa ngoại) Đề tài thuộc khoa |
| LecturerId | (Khóa ngoại) Giảng viên hướng dẫn đề tài |
| StatusId | (Khóa ngoại) Trạng thái đề tài |

**Students:** Chứa dữ liệu sinh viên thực hiện

|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| ID | Id của sinh viên |
| StudentId | Mã số sinh viên |
| FullName | Họ và tên sinh viên |
| Email | Email sinh viên |
| Password | Mật khẩu đăng nhập |
| UrlSlug | Mã định danh của sinh viên |
| DoB | Ngày tháng năm sinh của sinh viên |
| ImageUrl | Hình của sinh viên |
| Phone | Số điện thoại của sinh viên |
| Class | Lớp |
| Year | Niên khóa |
| Address | Địa chỉ của sinh viên |
| DepartmentId | (Khóa ngoại) Sinh viên thuộc khoa |
| RoleId | (Khóa ngoại) Phân quyền |

**StudentTopic:** Chứa Id của Sinh viên và đề tài (mối quan hệ n-n)

|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| StudentsId | (Khóa ngoại) Sinh viên thực hiện đề tài |
| TopicsId | (Khóa ngoại) Đề tài do sinh viên thực hiện |

**Lecturers:** Chứa dữ liệu giảng viên hướng dẫn

|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| ID | Id của giảng viên |
| FullName | Họ và tên giảng viên |
| Email | Email giảng viên |
| Password | Mật khẩu đăng nhập |
| UrlSlug | Mã định danh của giảng viên |
| Qualification |  |
| DoB | Ngày tháng năm sinh của giảng viên |
| ImageUrl | Hình của giảng viên |
| DepartmentId | (Khóa ngoại) Sinh viên thuộc khoa |
| RoleId | (Khóa ngoại) Phân quyền |

**Departments:** Chứa dữ liệu về khoa

|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| **ID** | Id của khoa |
| **Name** | Tên khoa |
| **UrlSlug** | Mã định danh của khoa |

**Status:** Chứa dữ liệu về các trạng thái của đề tài

|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| **ID** | Id của trạng thái |
| **Name** | Tên trạng thái |
| **UrlSlug** | Mã định danh của trạng thái |

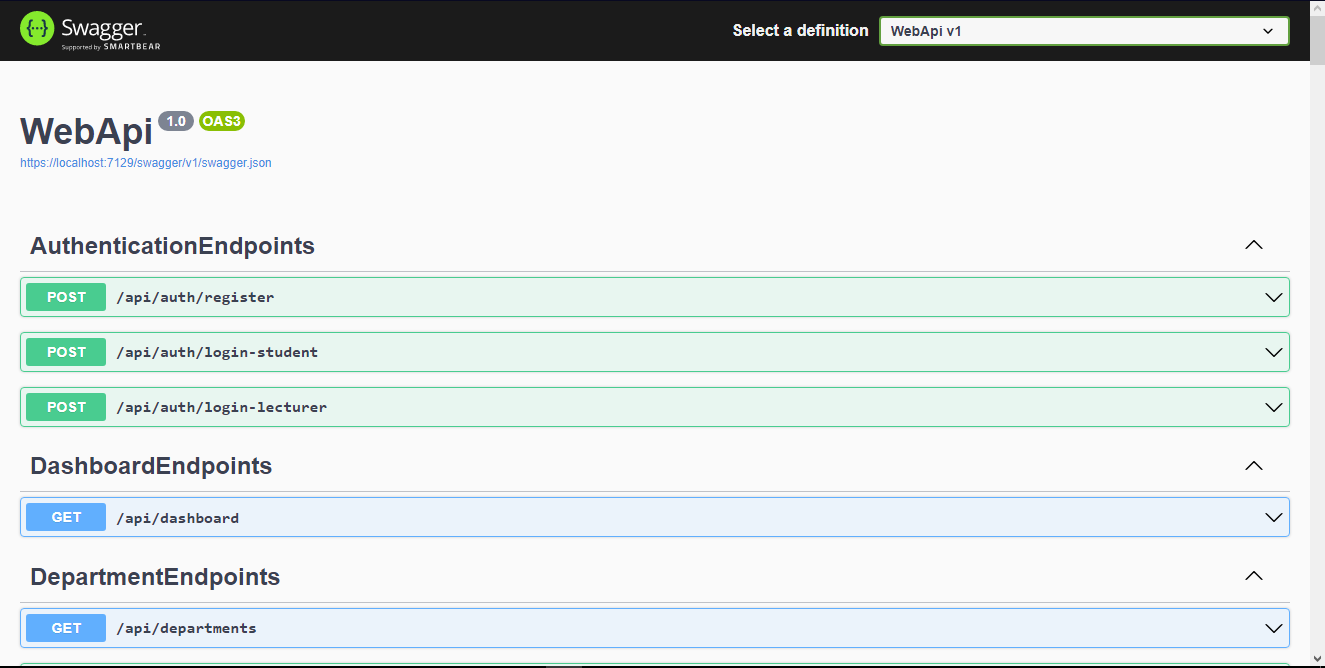
**Roles:** Chứa dữ liệu về các phân quyền người dùng

|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| **ID** | Id của trạng thái |
| **Name** | Tên trạng thái |
| **UrlSlug** | Mã định danh của trạng thái |

**Feedback:** Chứa dữ liệu về các liên hệ của người dùng

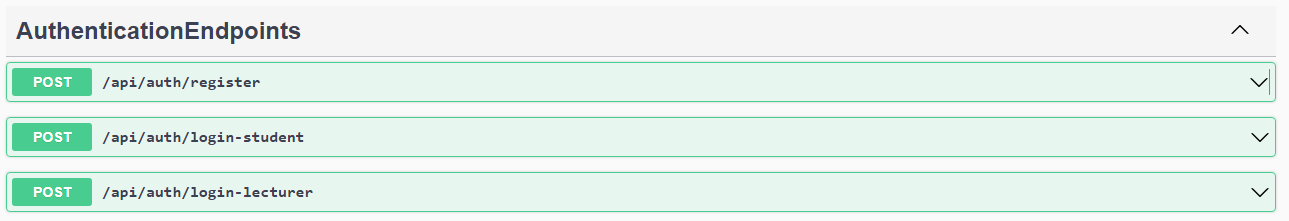
|  |  |
| --- | --- |
| **Tên cột** | **Nội dung** |
| **ID** | Id của phản hồi |
| **Username** | Tên người dùng |
| **Content** | Nội dung |
| **CreateDate** | Ngày gửi |

## **3.3. Xây dựng API**

****

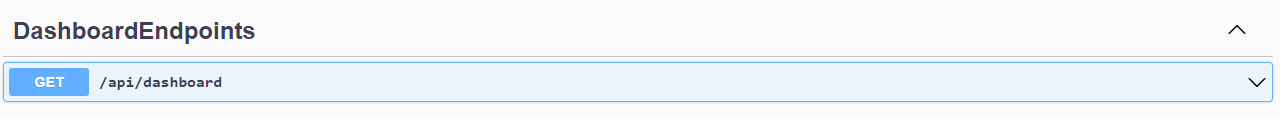
Các Endpoint

**AuthenticationEndpoints:** API quản lý đăng ký và đăng nhập



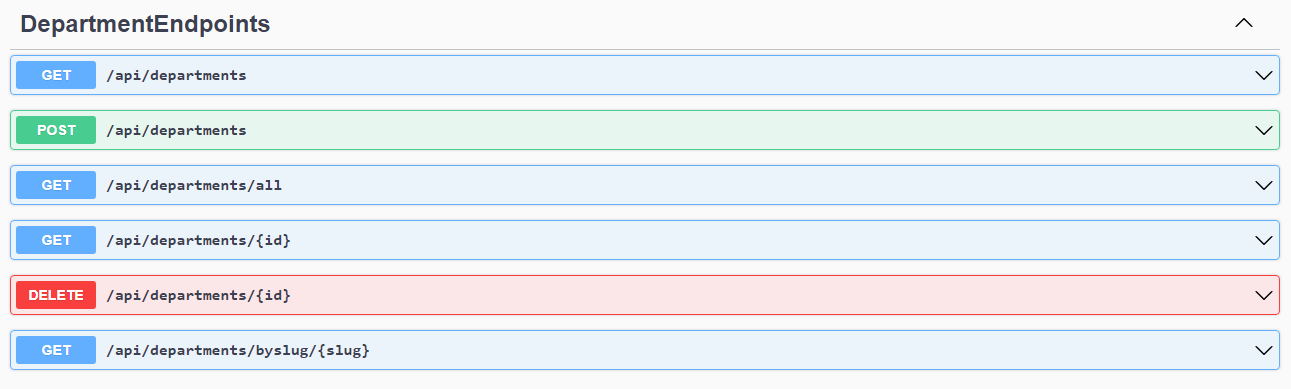
* Đăng ký tài khoản sinh viên
* Đăng nhập bằng tài khoản sinh viên
* Đăng nhập bằng tài khoản giảng viên

**DashboardEnpoints:** API quản lý



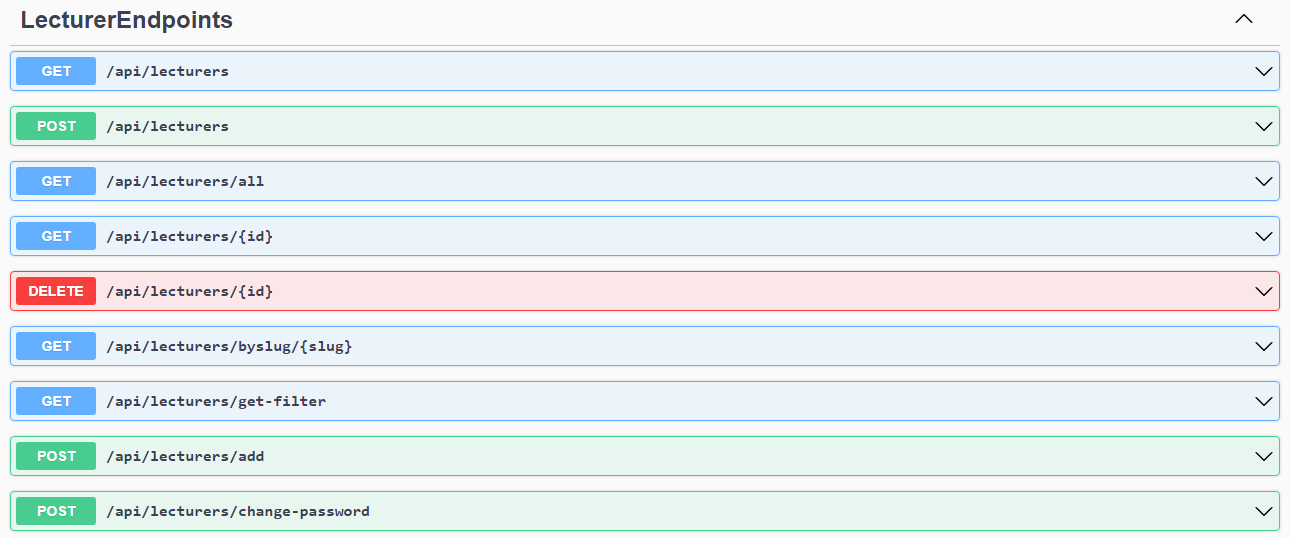
* Lấy thông tin quản lý (số lượng đề tài, khoa, sinh viên, giảng viên, tổng đề tài đã nghiệm thu)

**DepartmentEndpoint:** API của khoa

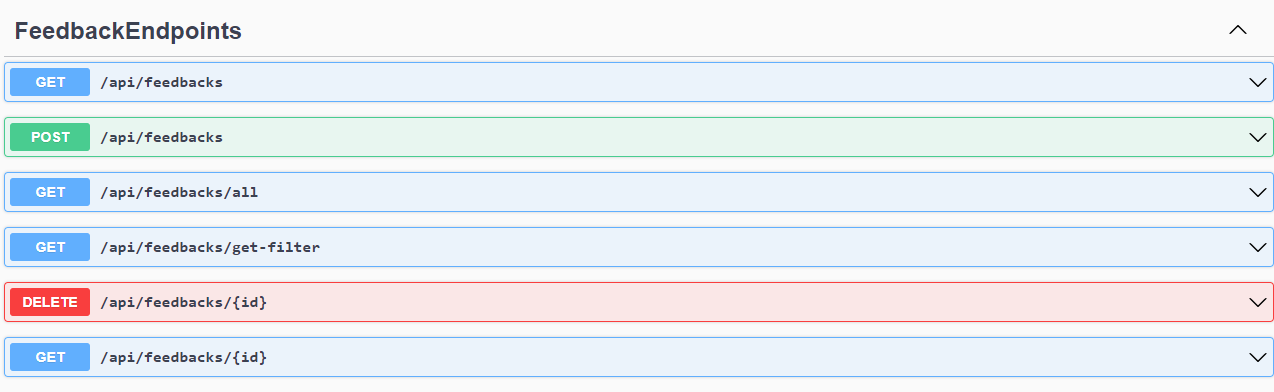
****

* Lấy danh sách khoa (có phân trang và lọc theo tên)
* Thêm/Cập nhật khoa (Theo Id)
* Lấy danh sách tất cả các khoa
* Lấy thông tin khoa theo Id khoa
* Xóa khoa đã có sẵn (Theo Id)
* Lấy thông tin khóa theo UrlSlug( Mã định danh) của khoa

**LecturerEndpoints:** API giảng viên

****

* Lấy danh sách giảng viên ( có phân trang và lọc theo tên, khoa)
* Cập nhật thông tin giảng viên
* Lấy danh sách tất cả giảng viên
* Lấy thông tin giảng viên theo Id giảng viên
* Xóa giảng viên
* Lấy thông tin khóa theo UrlSlug( Mã định danh) của giảng viên
* Lấy các thuộc tính lọc của giảng viên (Khoa)
* Thêm tài khoản giảng viên
* Đổi mật khẩu giảng viên

**FeedbackEnpoints**: API quản lý các liện hệ ****

- Lấy danh sách các feedback (có phân trang)

- Thêm feedback

- Lấy danh sách tất cả các feedback

- Lấy bộ lọc cho feedback

- Xoá feedback

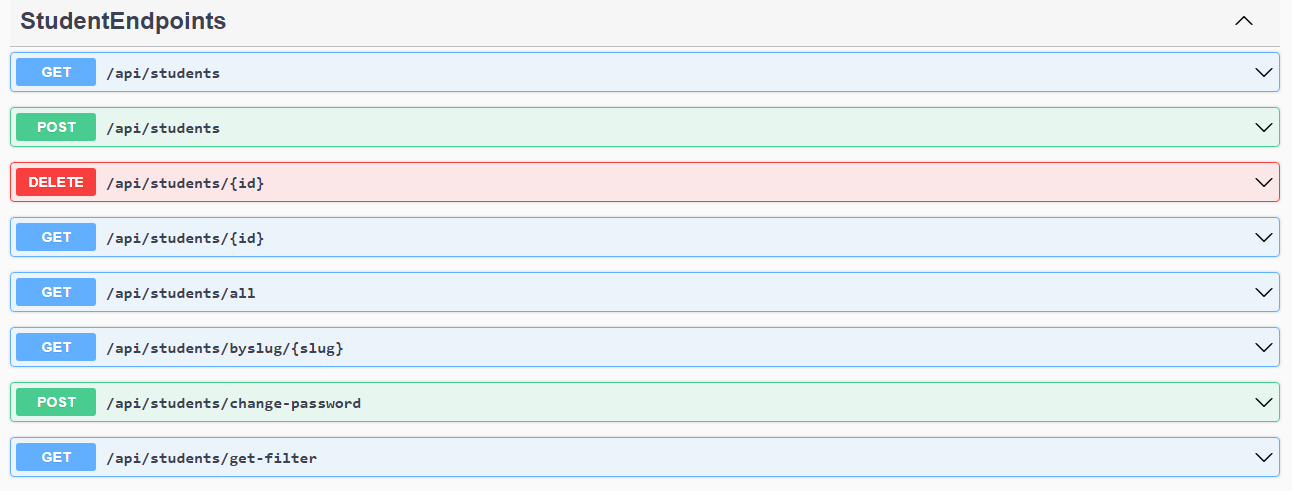
- Lấy thông tin feedback theo id

**OtherEndpoint:** API lấy trạng thái và phân quyền

****

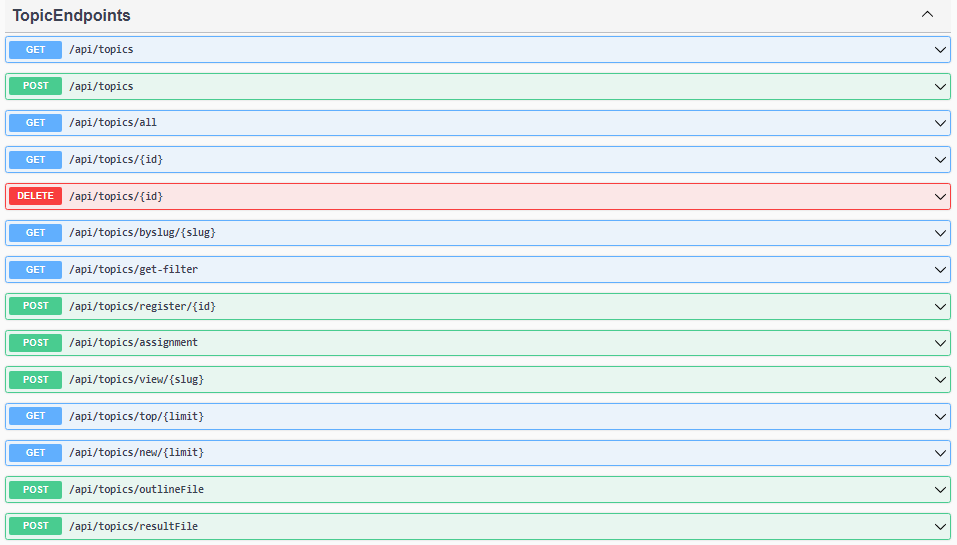
* Lấy danh sách các trạng thái của đề tài
* Lấy danh sách các phân quyền

**StudentEndpoints:** API sinh viên

****

* Lấy danh sách sinh viên ( có phân trang và lọc theo tên, khoa)
* Thêm/Cập nhật sinh viên
* Xóa sinh viên
* Lấy thông tin giảng viên theo Id sinh viên
* Lấy danh sách tất cả sinh viên
* Lấy thông tin khóa theo UrlSlug( Mã định danh) của sinh viên
* Đổi mật khẩu sinh viên
* Lấy các thuộc tính lọc của sinh viên (Khoa)

**TopicEndpoints**: API đề tài

****

* Lấy danh sách đề tài (có phân trang và lọc)
* Thêm/Cập nhật đề tài
* Lấy danh sách tất cả đề tài
* Lấy chi tiết đề tài theo Id đề tài
* Xóa đề tài
* Lấy chi tiết đề tài theo UrlSlug( Mã định danh) của đề tài
* Lấy danh sách các mục lọc của đề tài (Khoa, Giảng viên)
* Đăng ký đề tài ( dành cho sinh viên)
* Phân công giảng viên cho 1 đề tài
* Tăng lượt xem đề tài (Theo Slug)
* Lấy n danh sách đề tài có nhiều lượt xem
* Lấy n danh sách đề tài mới nhất
* Thêm file thuyết minh cho đề tài
* Thêm file kết quả cho đề tài

## **3.4. Xây dựng giao diện**

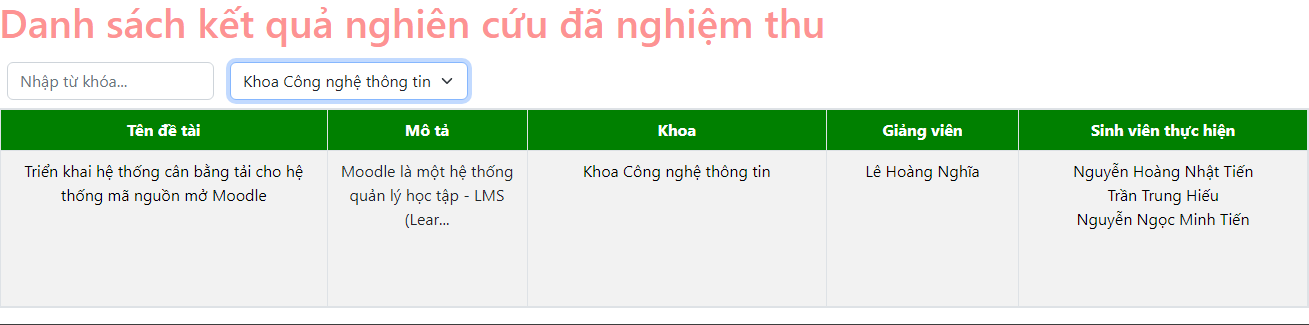
### **3.4.1. Trang chủ**

****

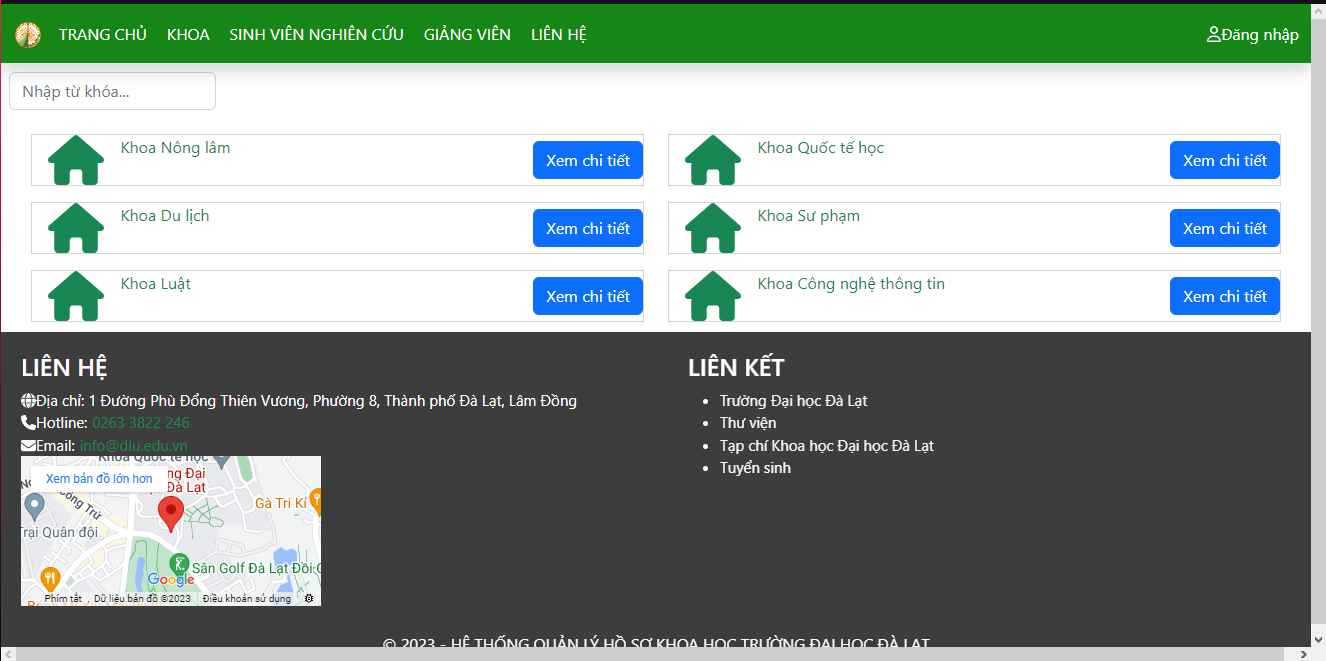
Đây là giao diện trang chủ. Tại đây sẽ có thể xem danh sách các kết quả nghiên cứu đã được nghiệm thu của sinh viên và các đề tài được xem nhiều nhất cũng như mới nhất 

Danh sách các kết quả nghiên cứu của sinh viên, bao gồm tên đề tài, mô tả, ngày thực hiện, ngày nghiệm thu, số người thực hiện, khoa, giảng viên hướng dẫn, sinh viên thực hiện và trạng thái của đề tài

Có thể lọc theo từ khóa hoặc khoa

****

### **3.4.2. Giao diện danh sách các khoa**

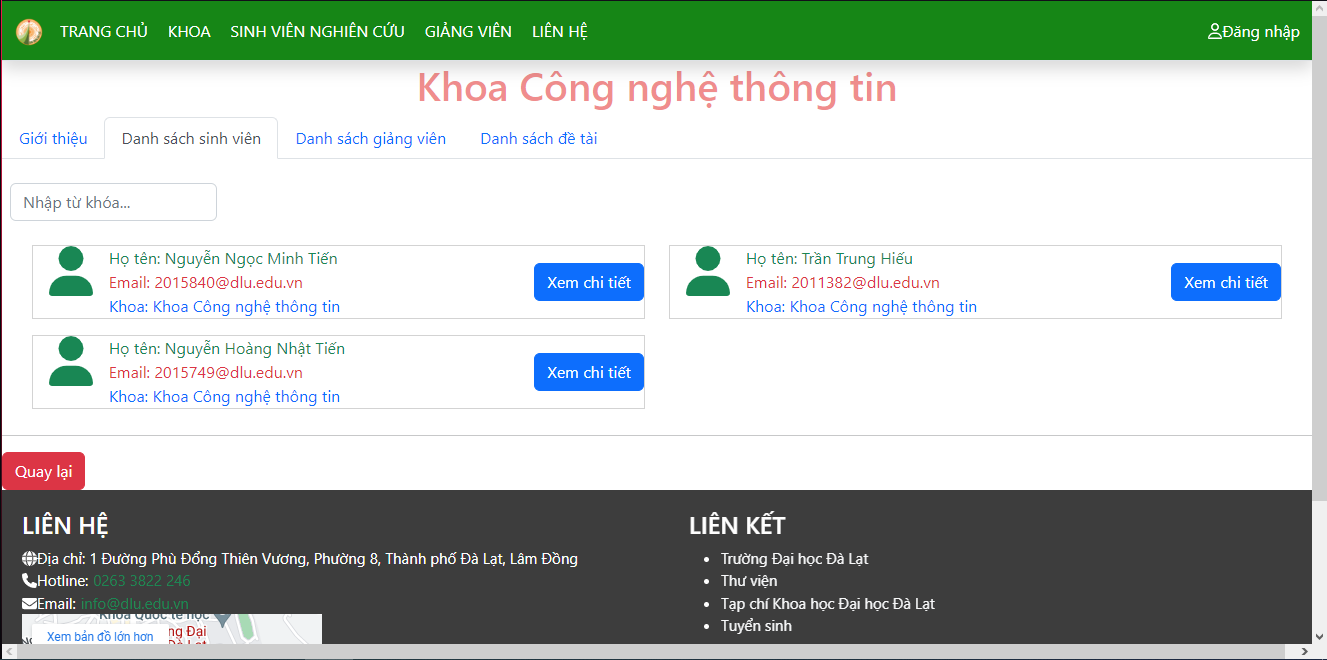
****

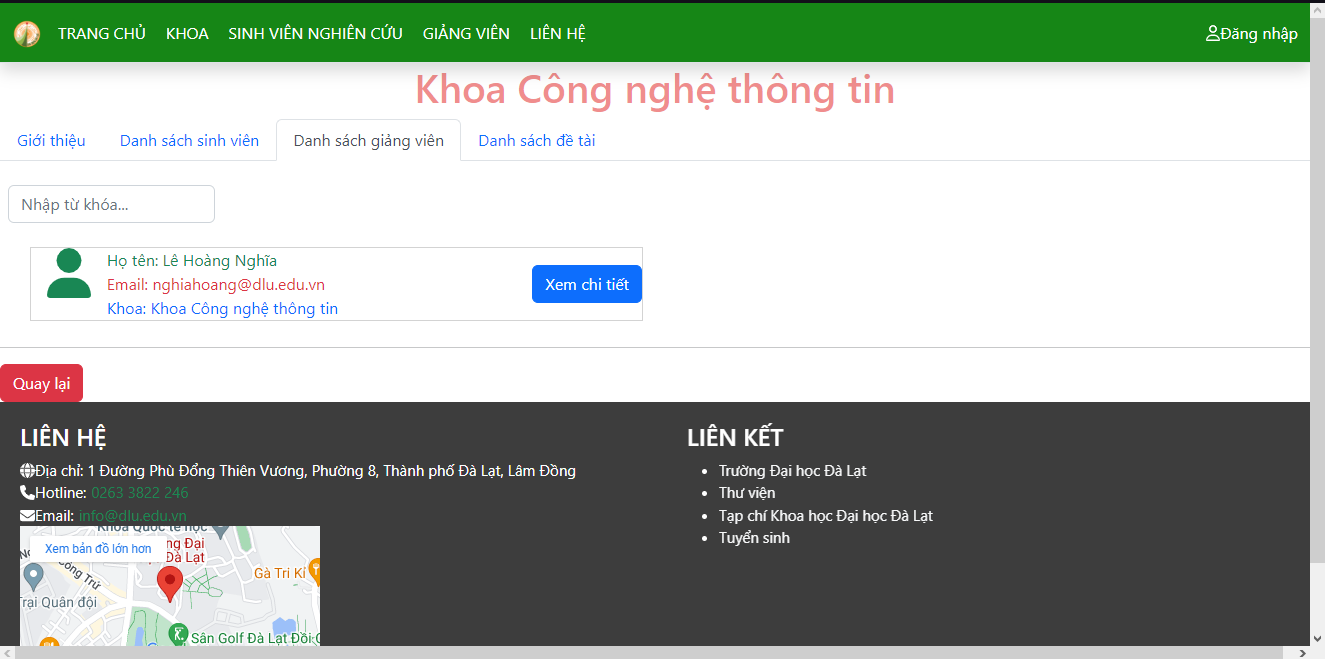
Hiển thị tất cả các khoa của đề tài có thể xem được

Khi người dùng bấm vào xem chi tiết

****

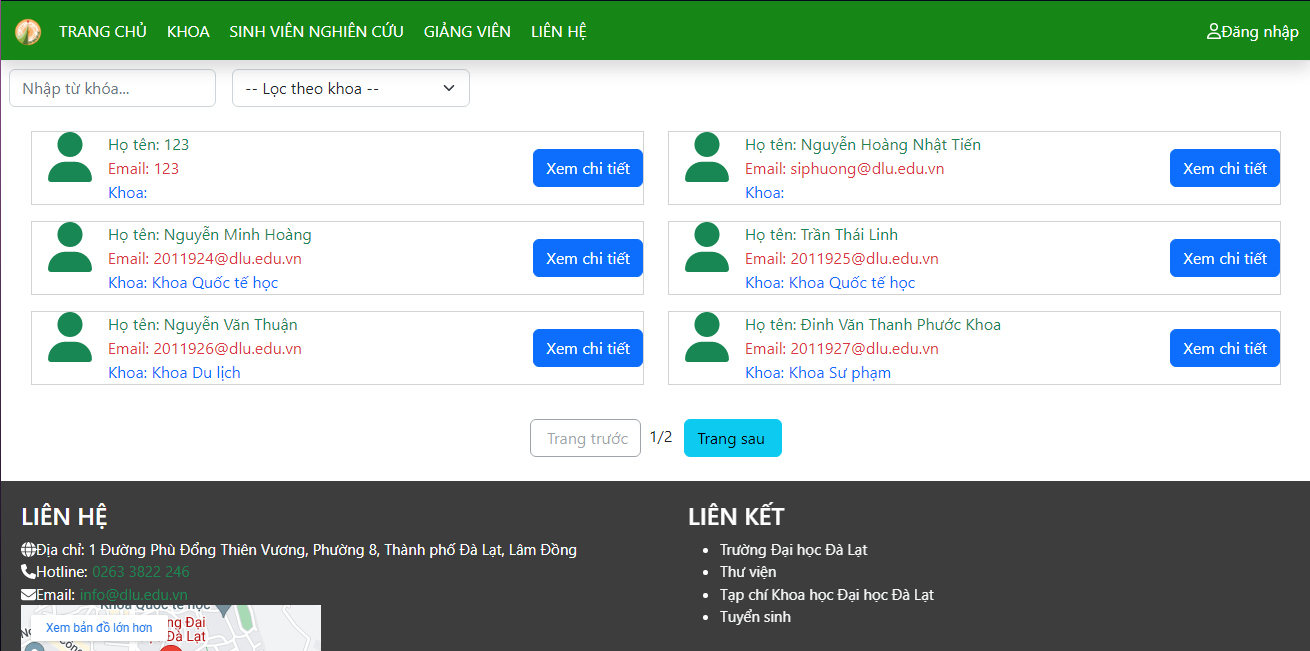
Tại đây sẽ hiển thị chi tiết về khoa bao gồm Giới thiệu, Danh sách sinh viên của khoa, Danh sách giảng viên hướng dẫn và danh sách đề tài

****

****

****

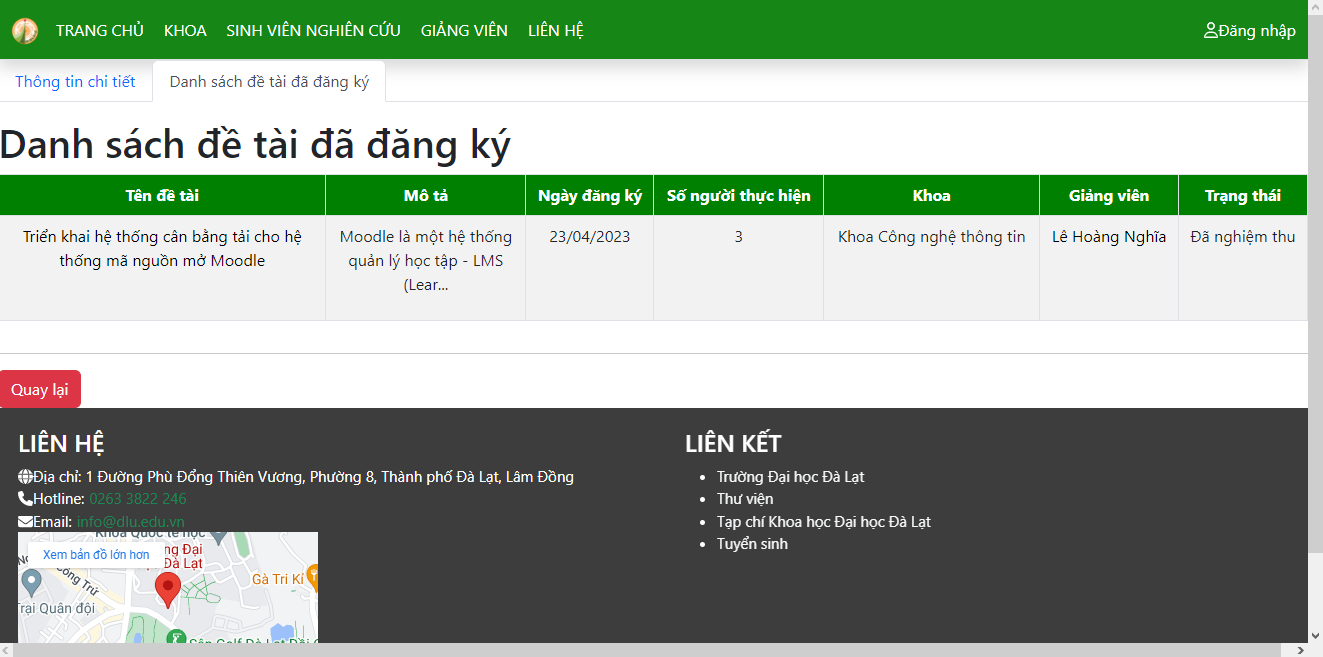
### **3.4.3. Giao diện sinh viên nghiên cứu**

****

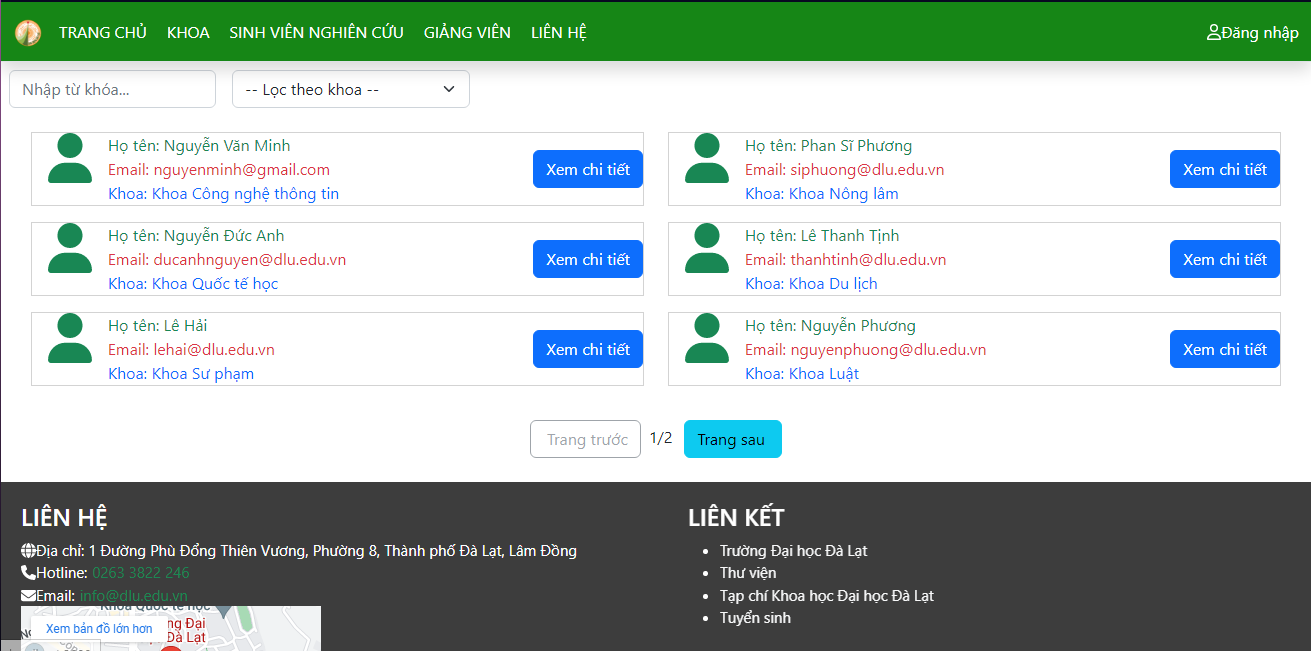
Hiển thị tất cả sinh viên thực hiện nghiên cứu khoa học

Khi người dùng bấm vào xem chi tiết sẽ hiển thị chi tiết sinh viên bao gồm thông tin chi tiết của sinh viên và các đề tài đã đăng ký

****

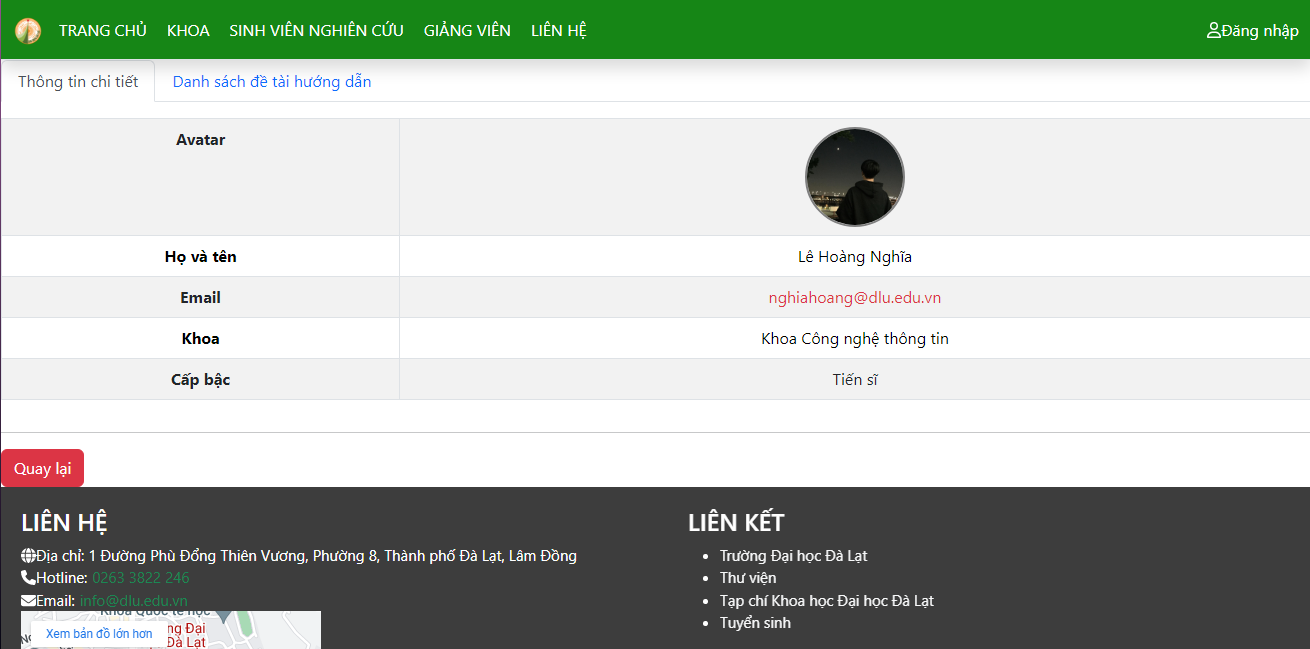
****

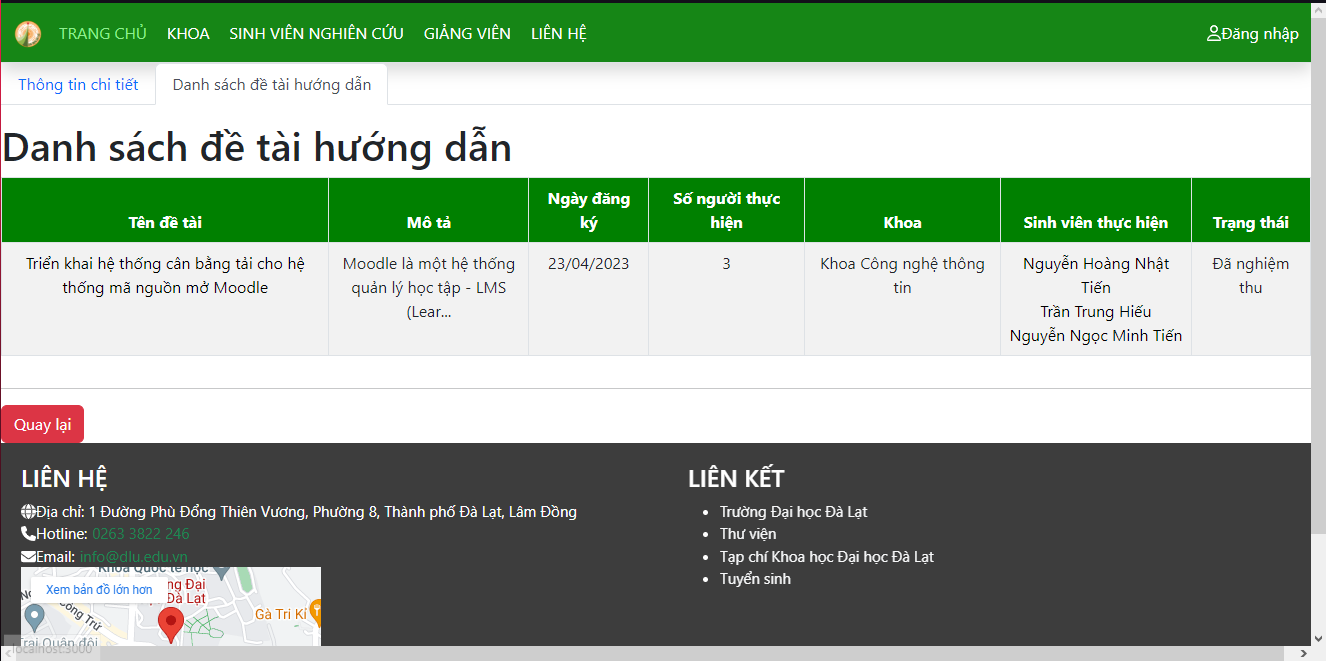
### **3.4.4. Giao diện danh sách giảng viên**

****

Hiển thị tất cả giảng viên hướng dẫn

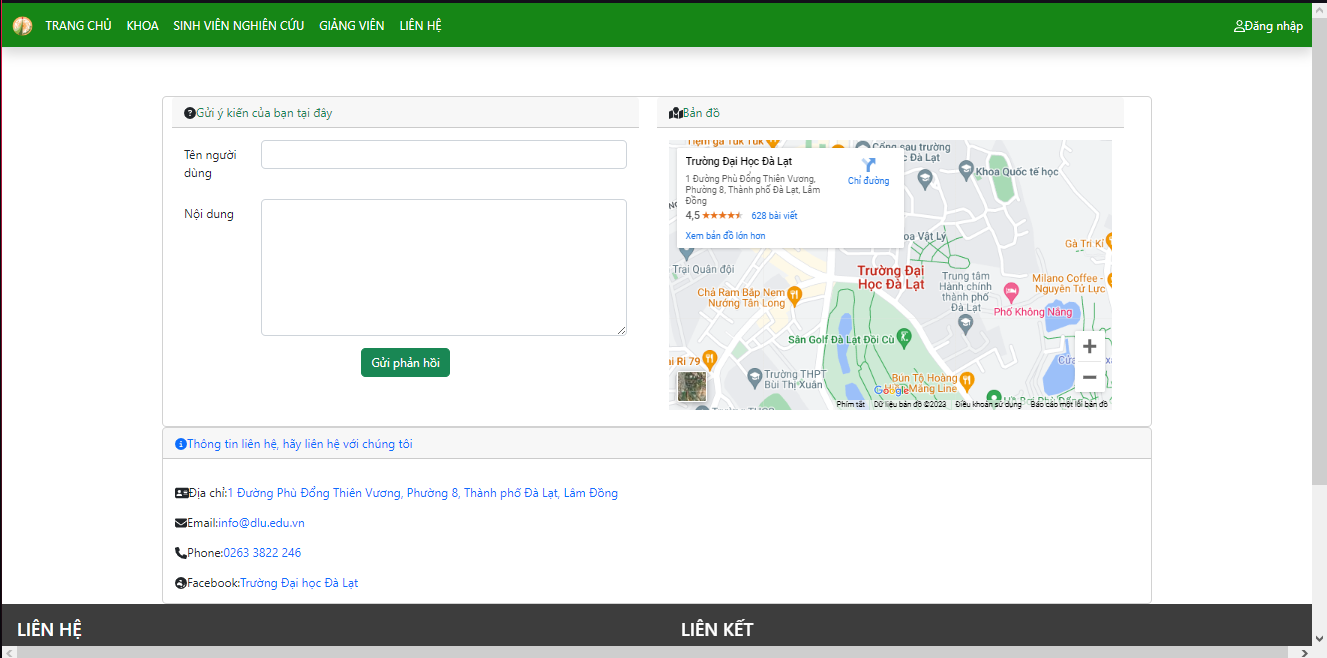
Khi người dùng bấm vào xem chi tiết sẽ hiển thị chi tiết giảng viên bao gồm thông tin chi tiết của giảng viên và các đề tài đang hướng dẫn

****

****

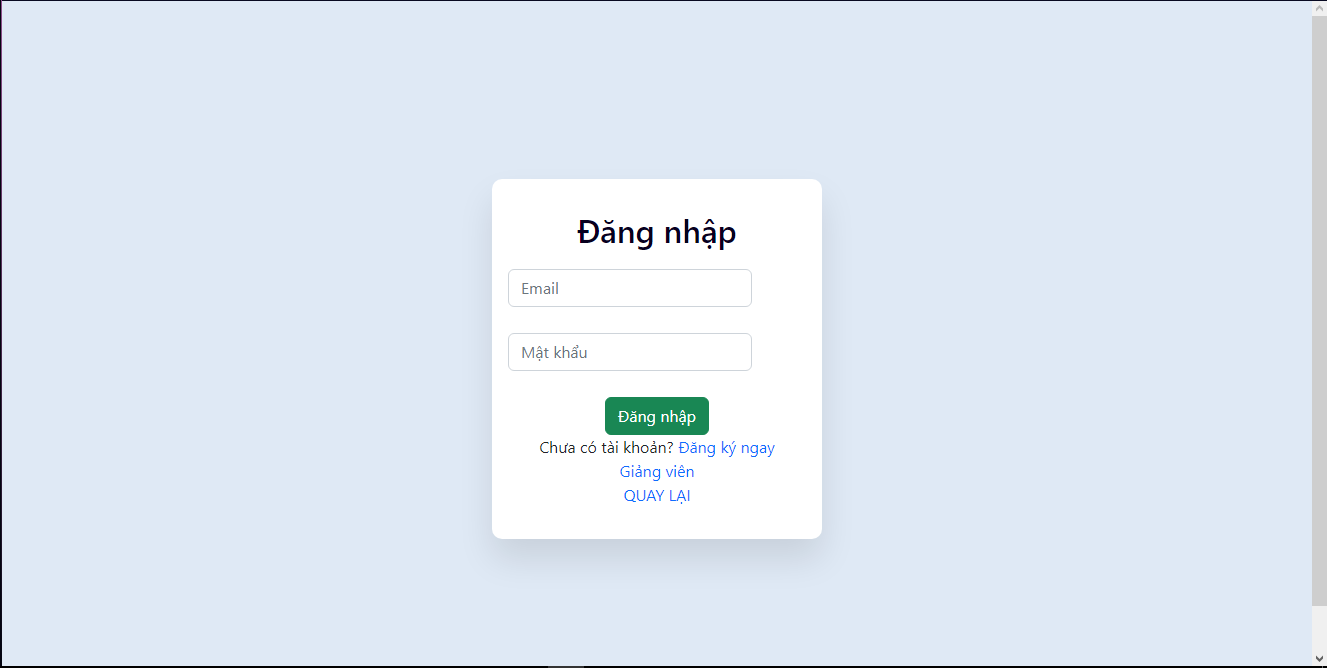
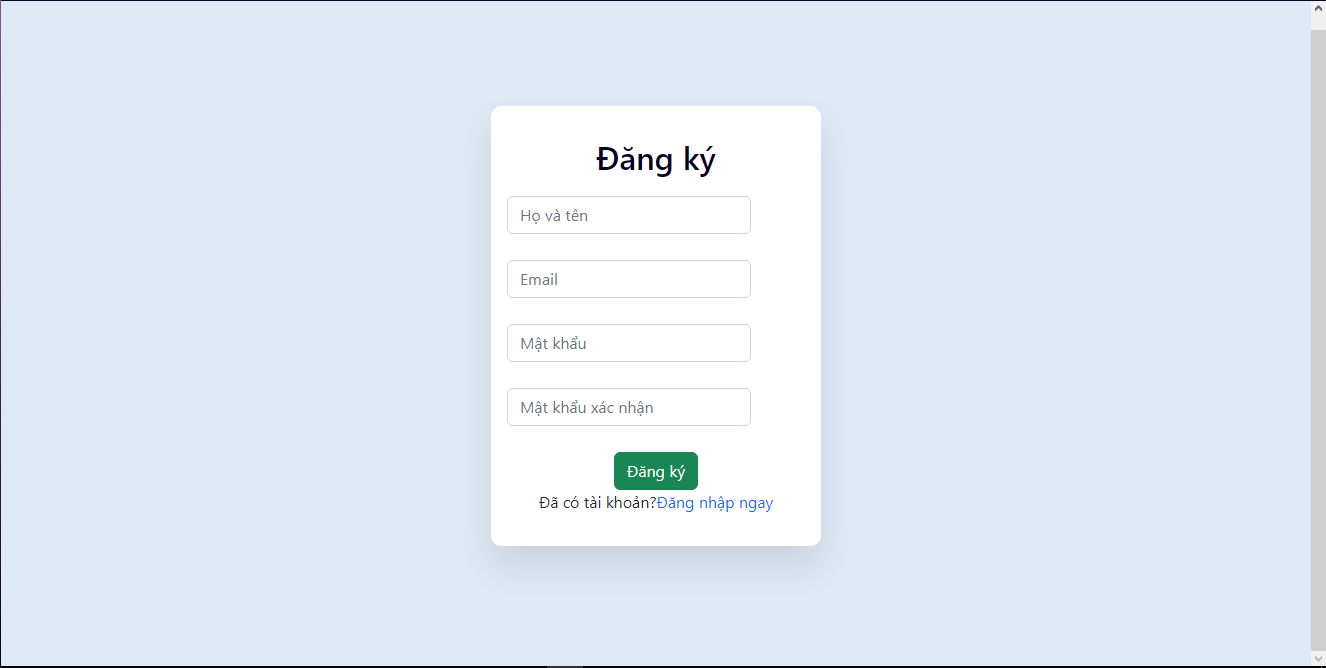
### **3.4.5. Giao diện liên hệ admin**

Tại đây sẽ hiển thị form để liên hệ và các thông tin liên hệ

****

### **3.4.6. Giao diện đăng nhập, đăng ký**

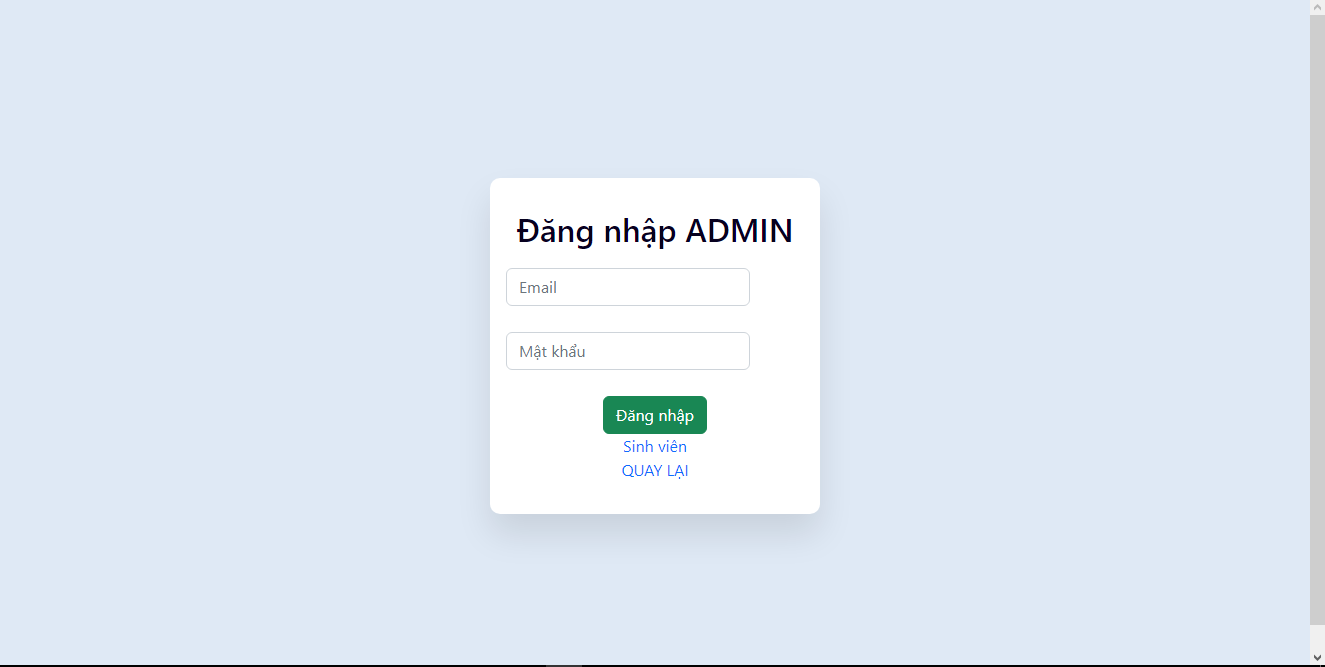
Giao diện đăng nhập, đăng ký sinh viên

**** ****

Sau khi sinh viên đăng ký, sẽ có thể quản lý các đề tài bằng cách ấn vào avatar

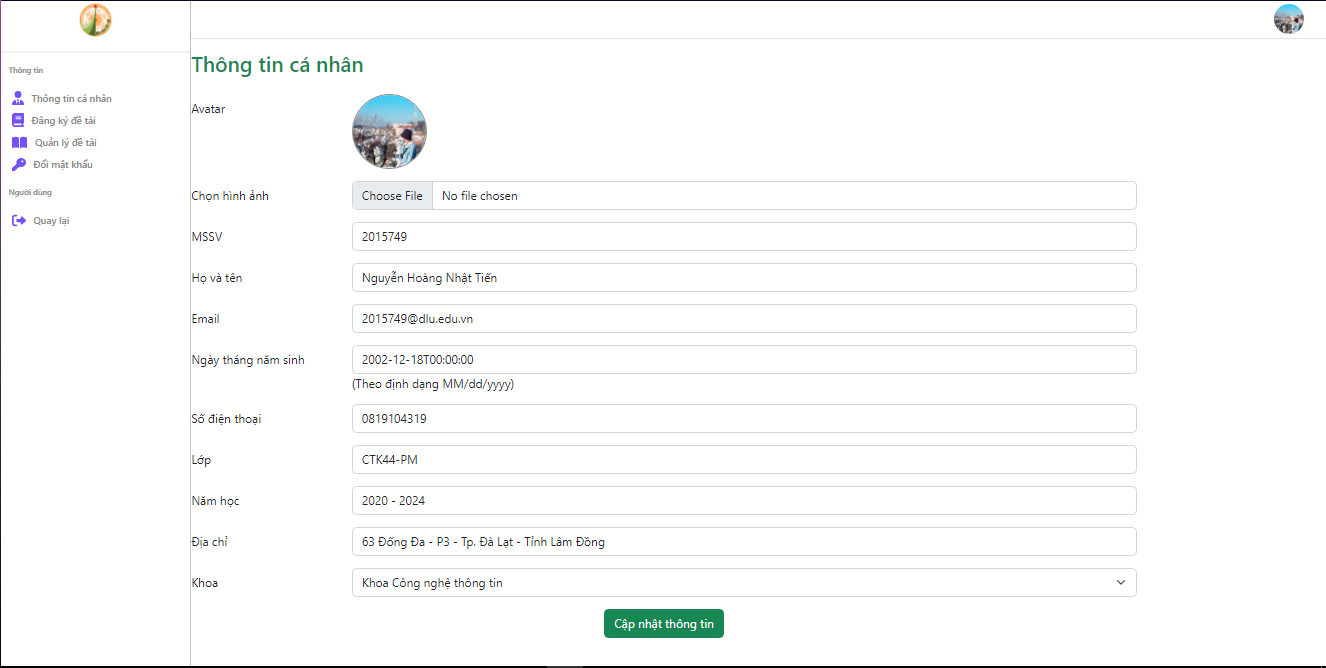
****

Giao diện đăng nhập giảng viên

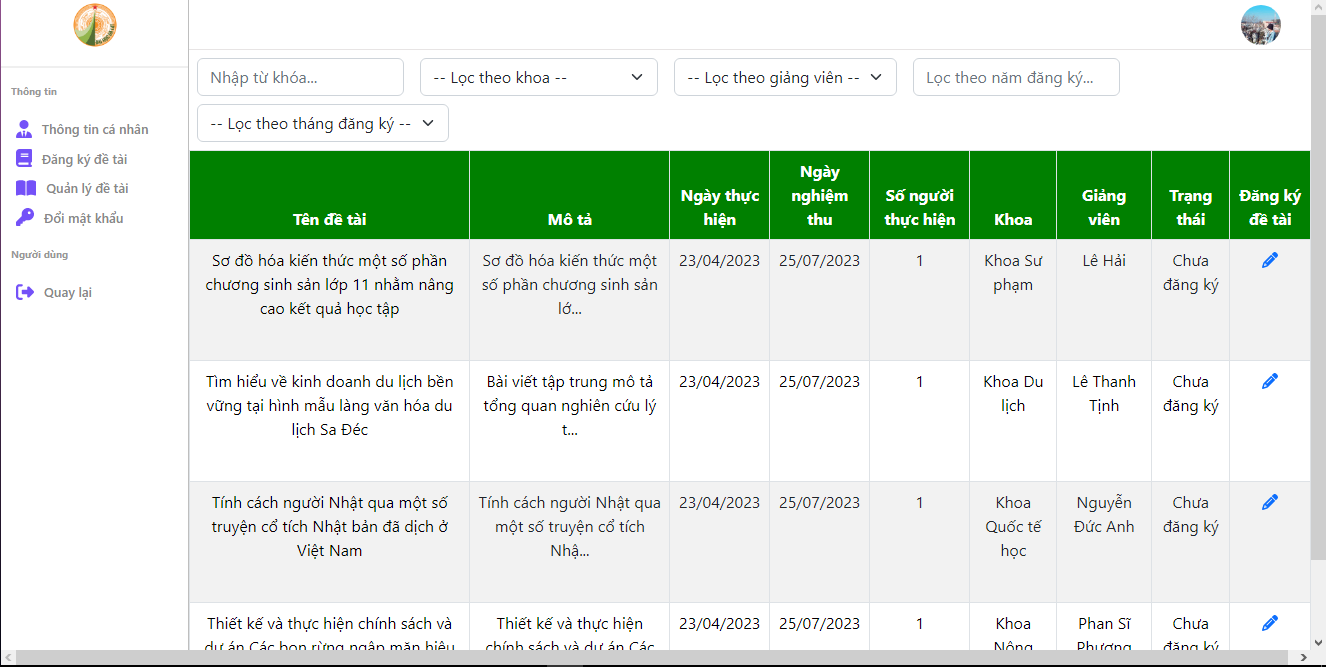


### **3.4.7.** **Giao diện sinh viên**

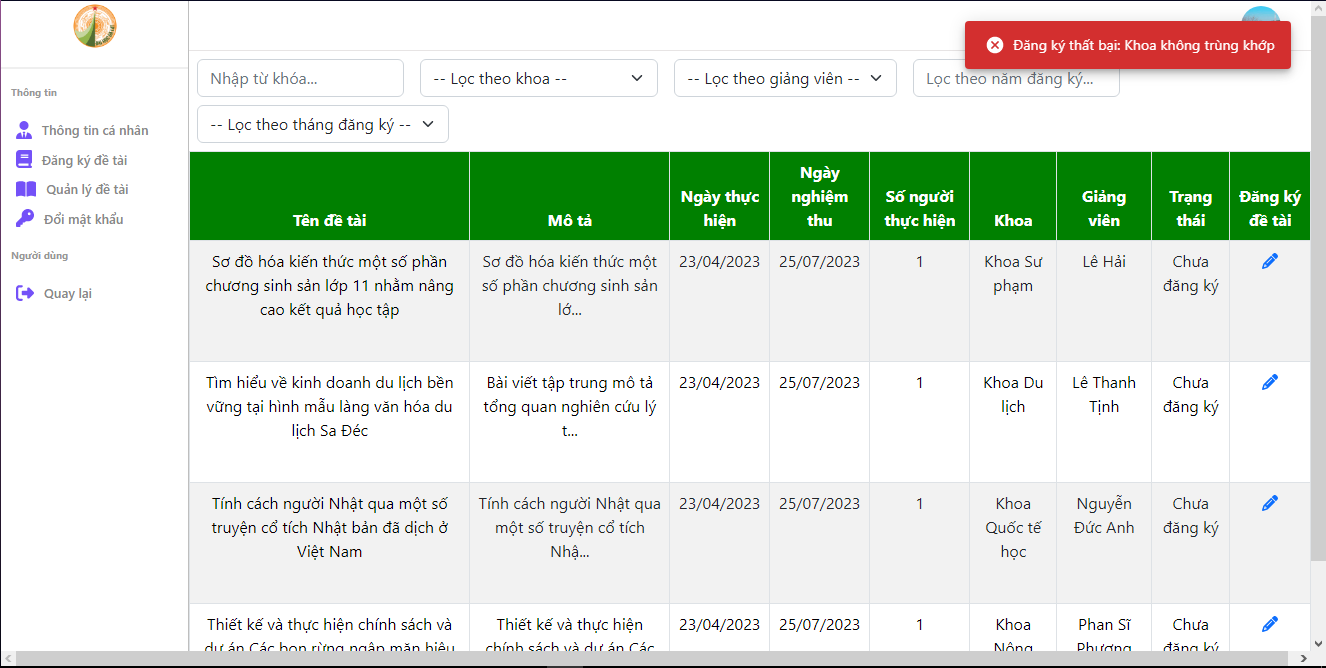
Chỉnh sửa thông tin sinh viên



Đăng ký đề tài



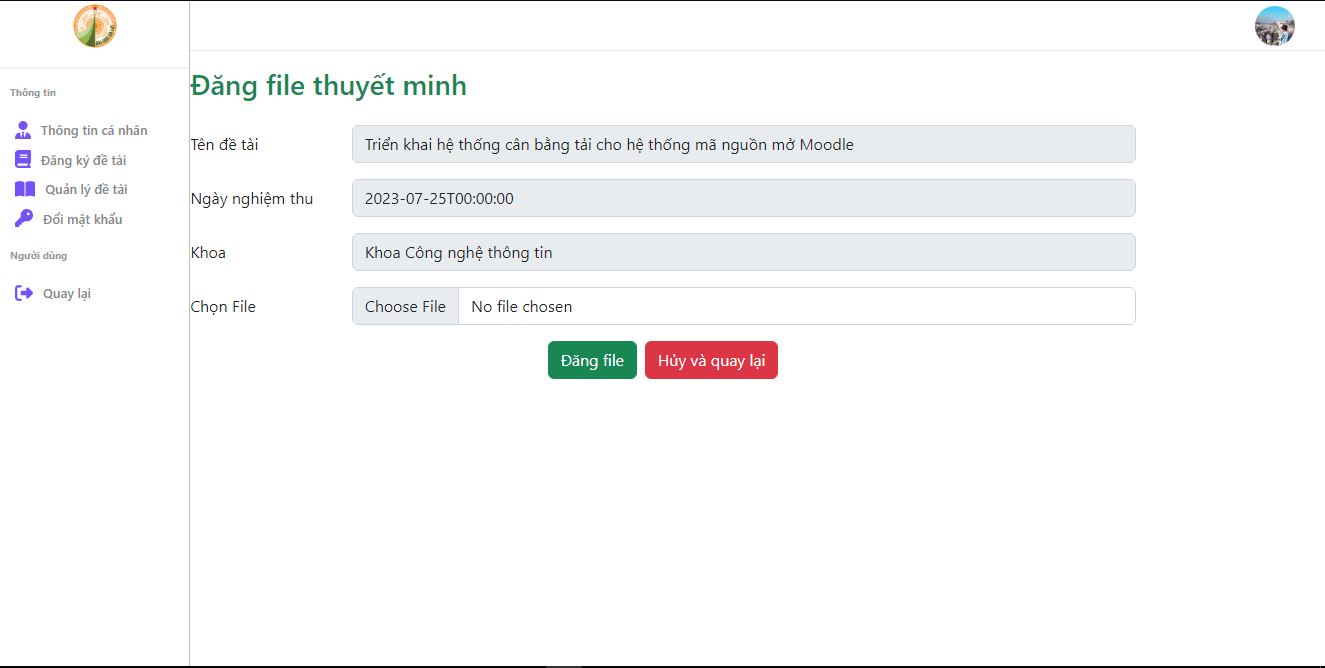
Nếu như sinh viên không cùng khoa với đề tài sẽ báo lỗi



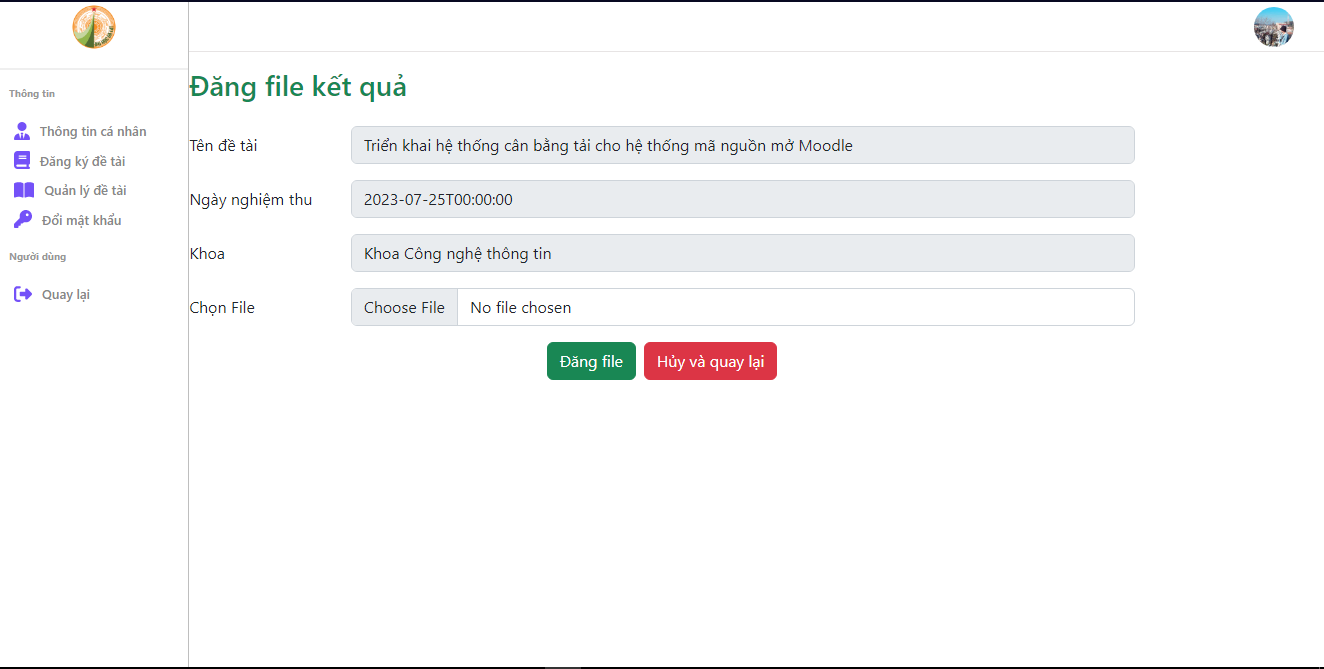
Quản lý đề tài



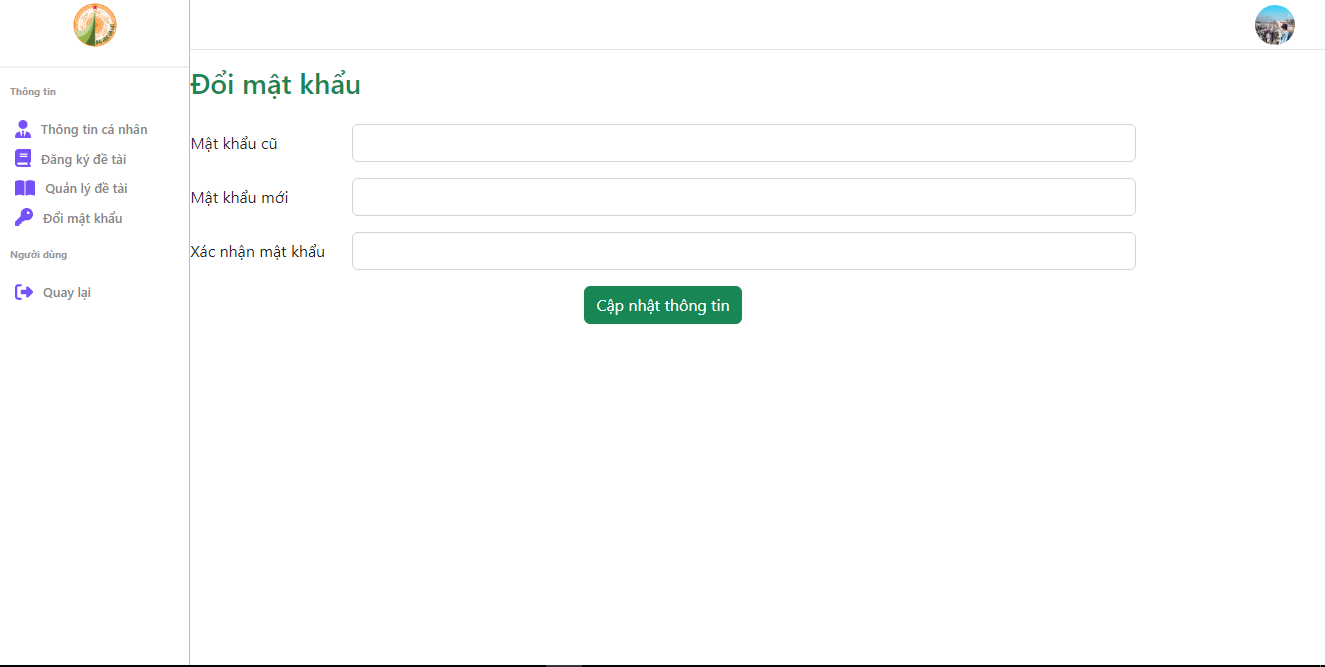
Đăng file thuyết minh cho đề tài



Đăng file kết quả cho đề tài



Đổi mật khẩu



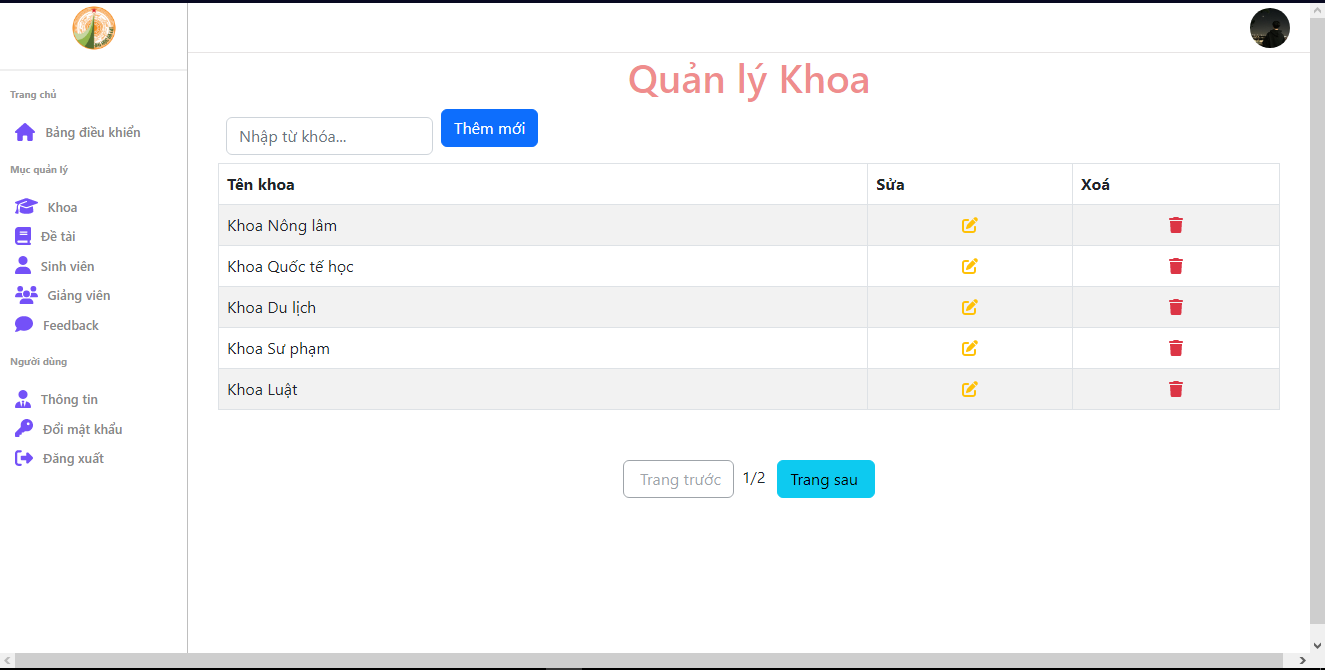
### **3.4.8. Giao diện Admin**

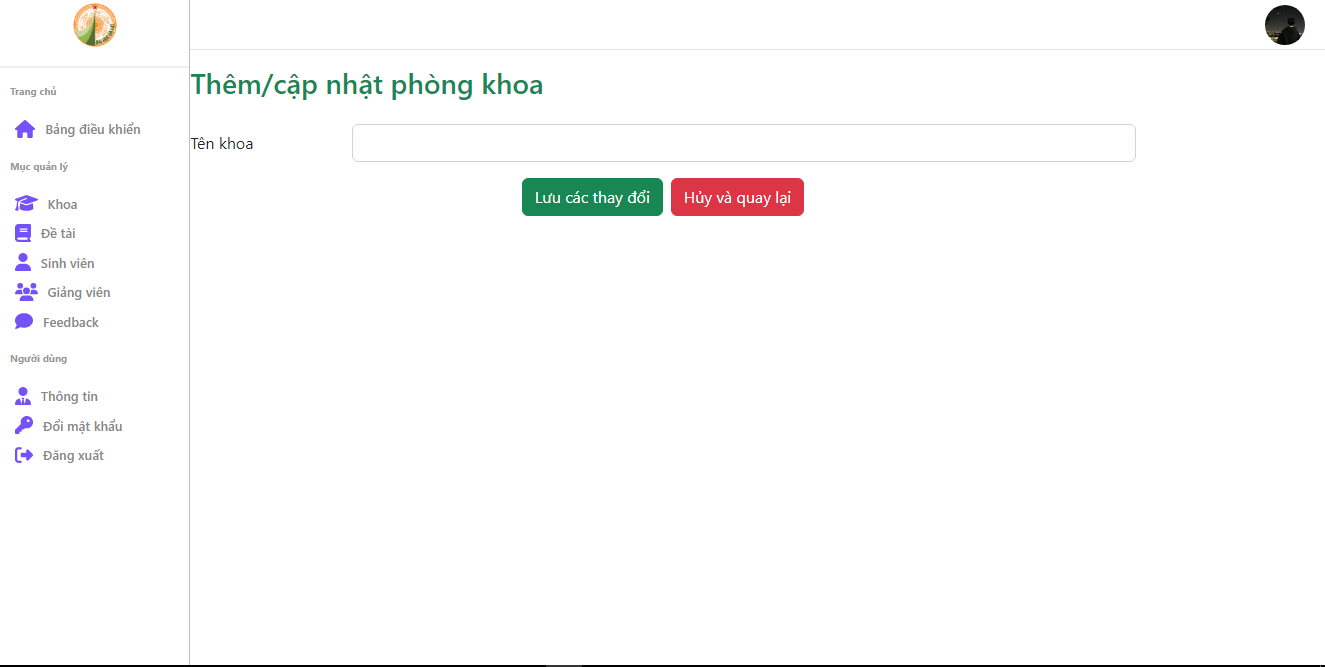
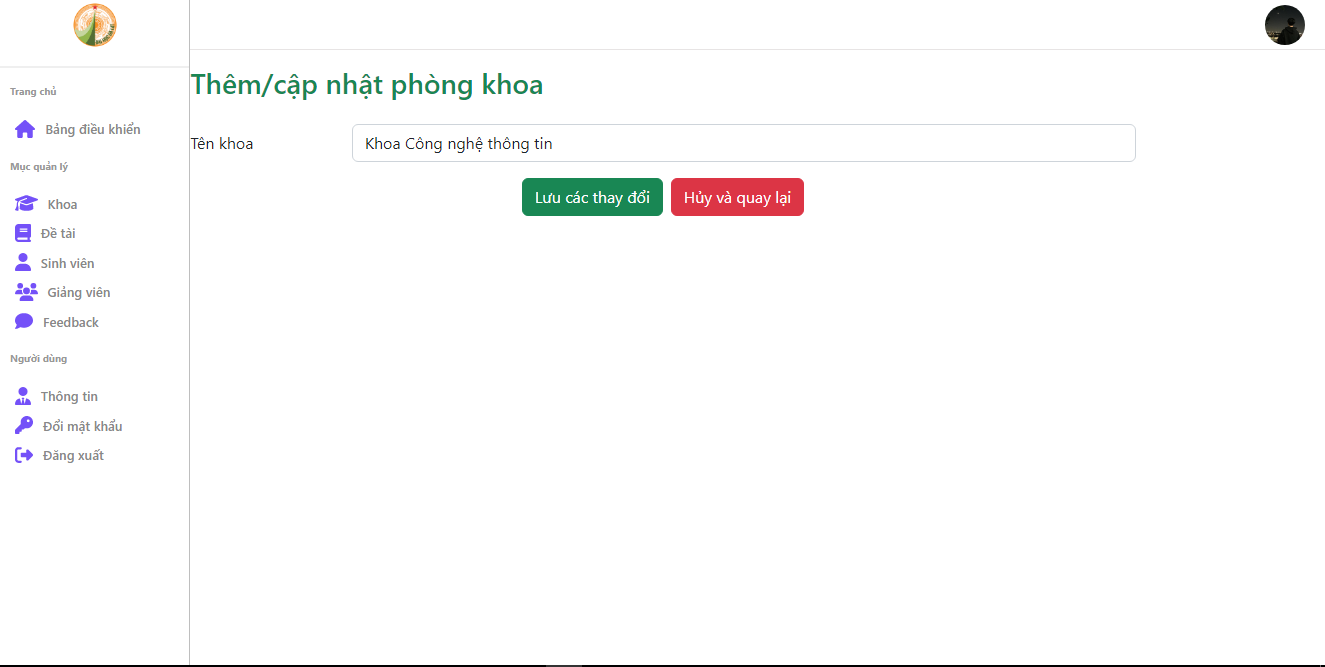
Sau khi giảng viên đăng nhập, giao diện Admin sẽ hiện lên

****

Tại đây gồm các mục Dashboard, khoa, đề tài, sinh viên, giảng viên, feedback của người dùng, thông tin admin và đăng xuất

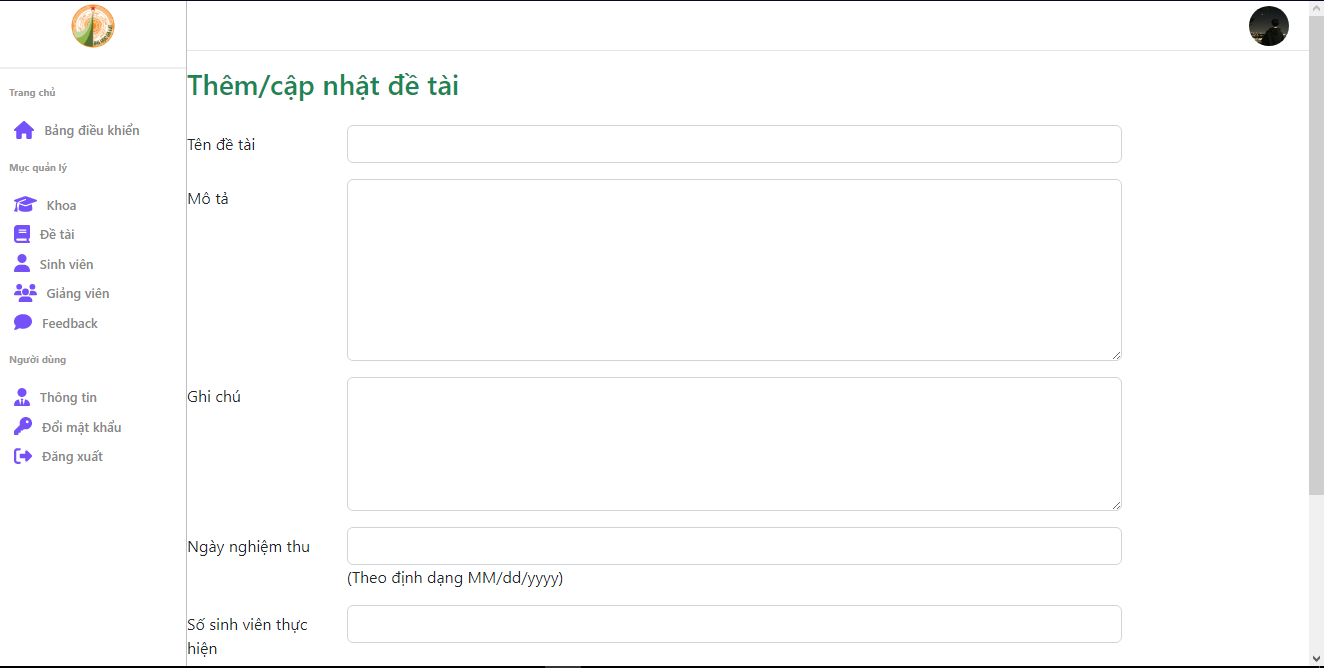
Tại Dashboard sẽ hiển thị số lượng đề tài, khoa, sinh viên, giảng viên và số lượng đề tài đã nghiệm thu

Tại Khoa sẽ có thể lọc khoa, thêm mới, chỉnh sửa và xóa khoa 

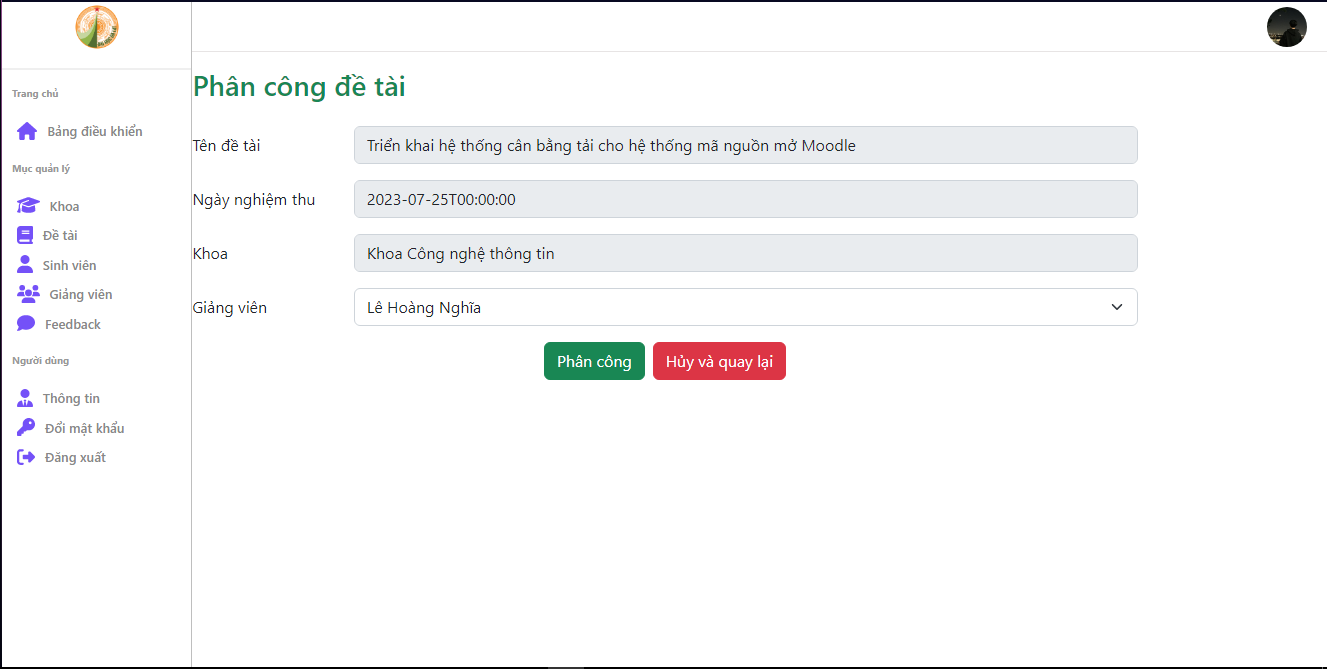
Thêm mới khoa Chỉnh sửa khoa 

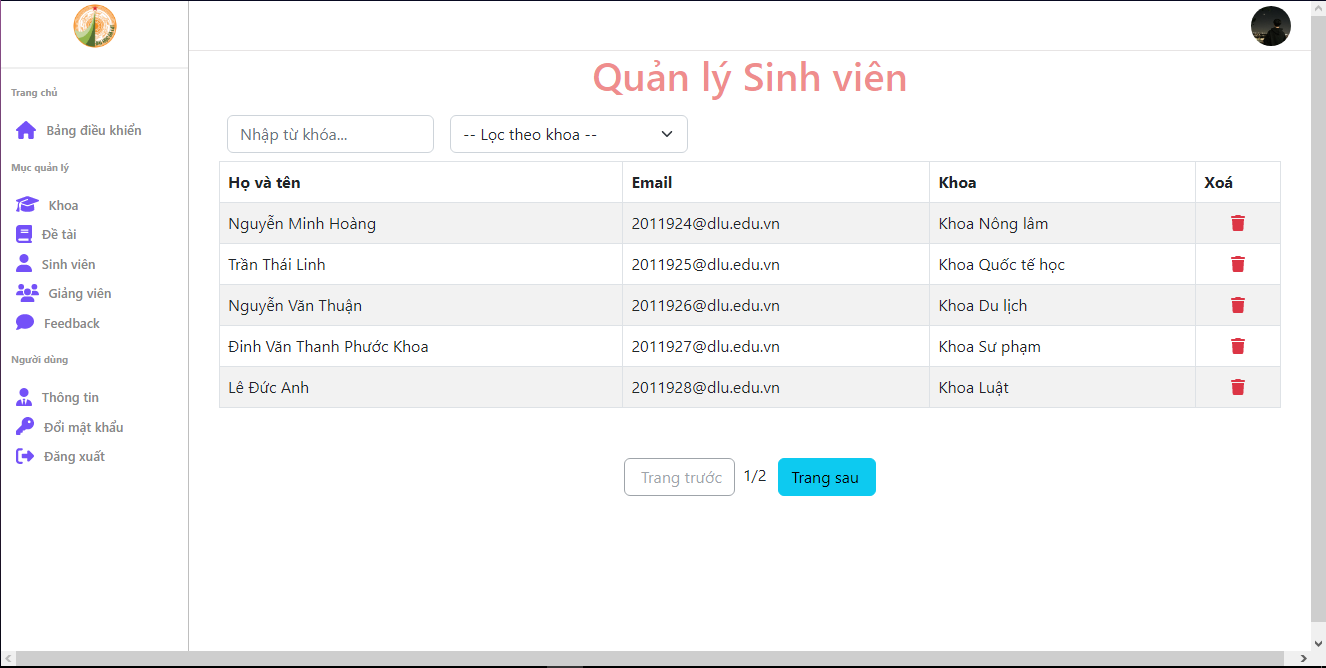
Tương tự như đề tài, sinh viên và giảng viên

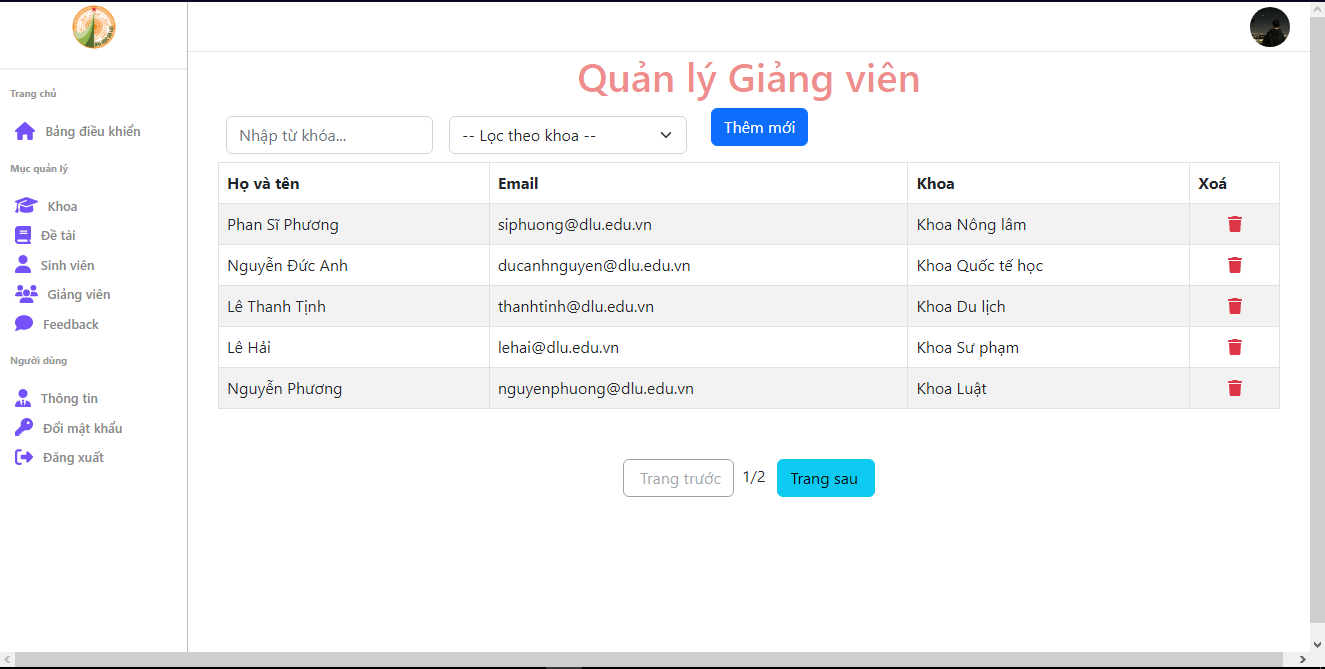
Quản lý đề tài 

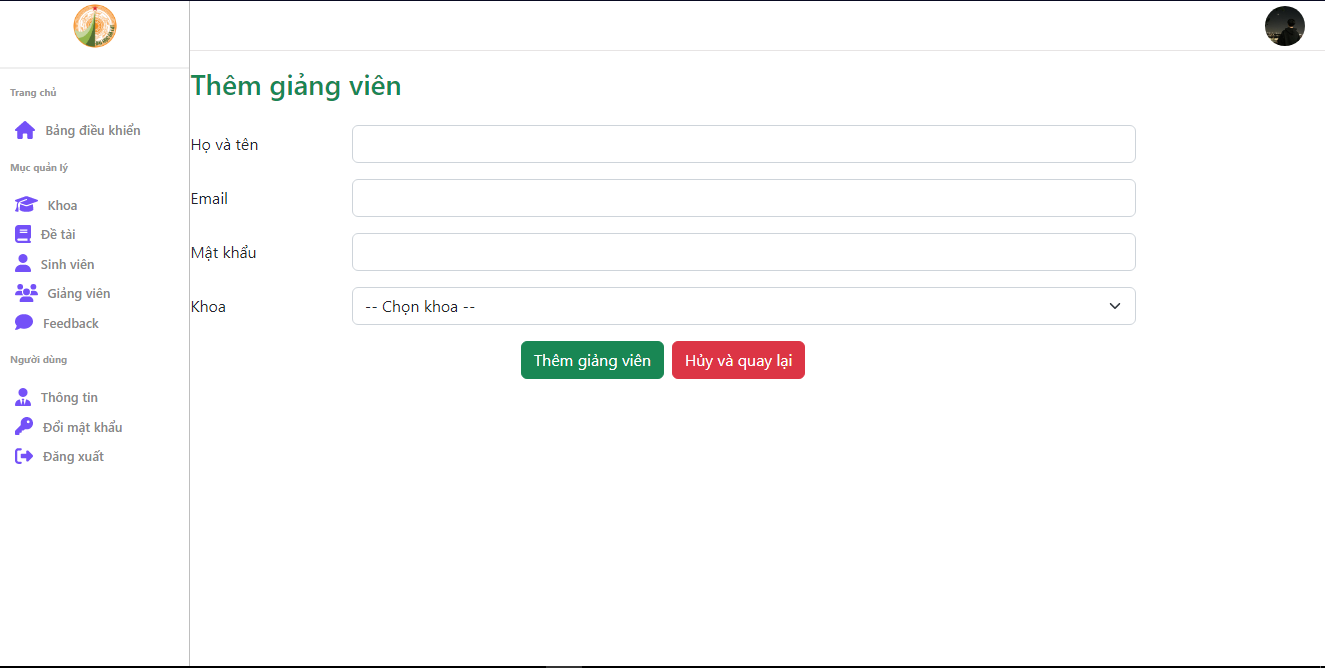
Thêm mới 

Chỉnh sửa 

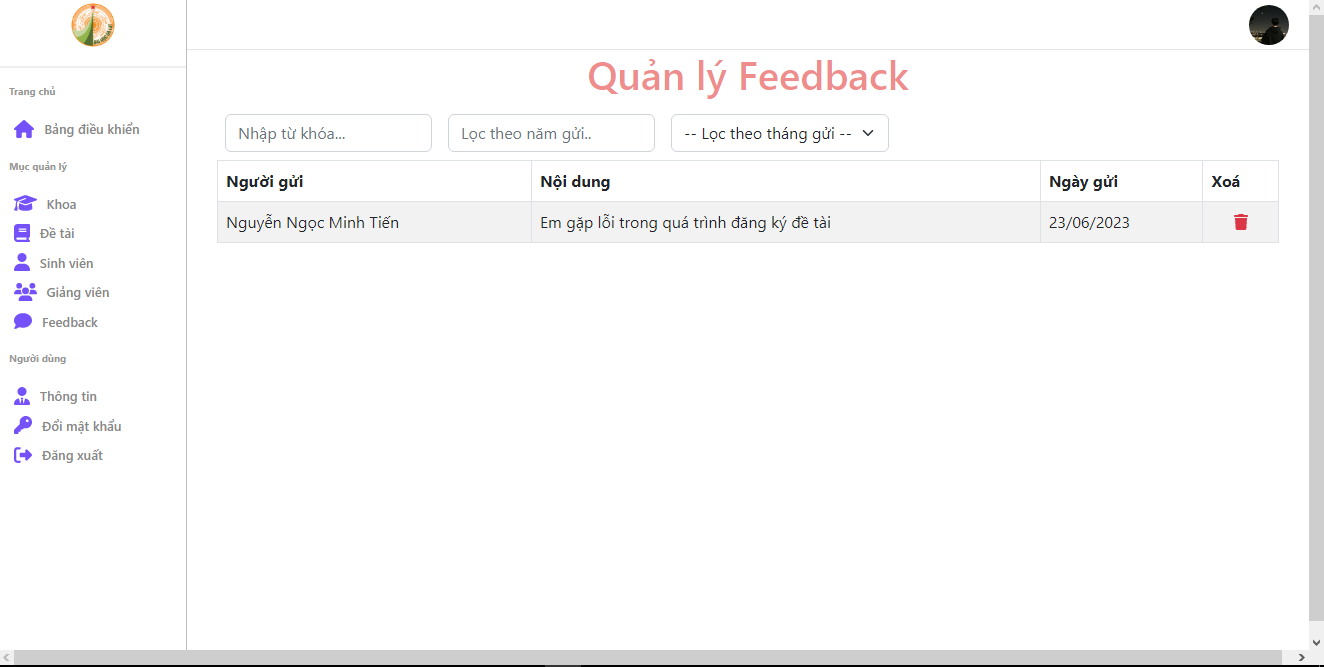
Phân công giảng viên 

Quản lý sinh viên 

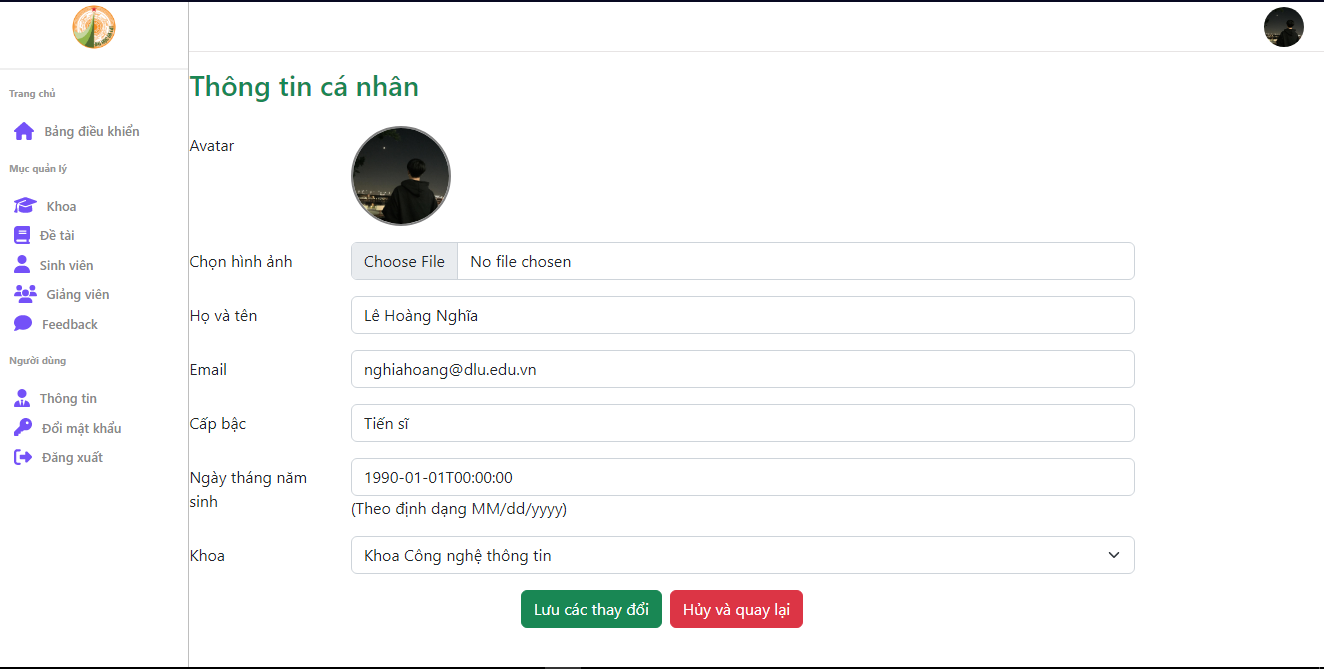
Quản lý giảng viên 

Thêm tài khoản giảng viên 

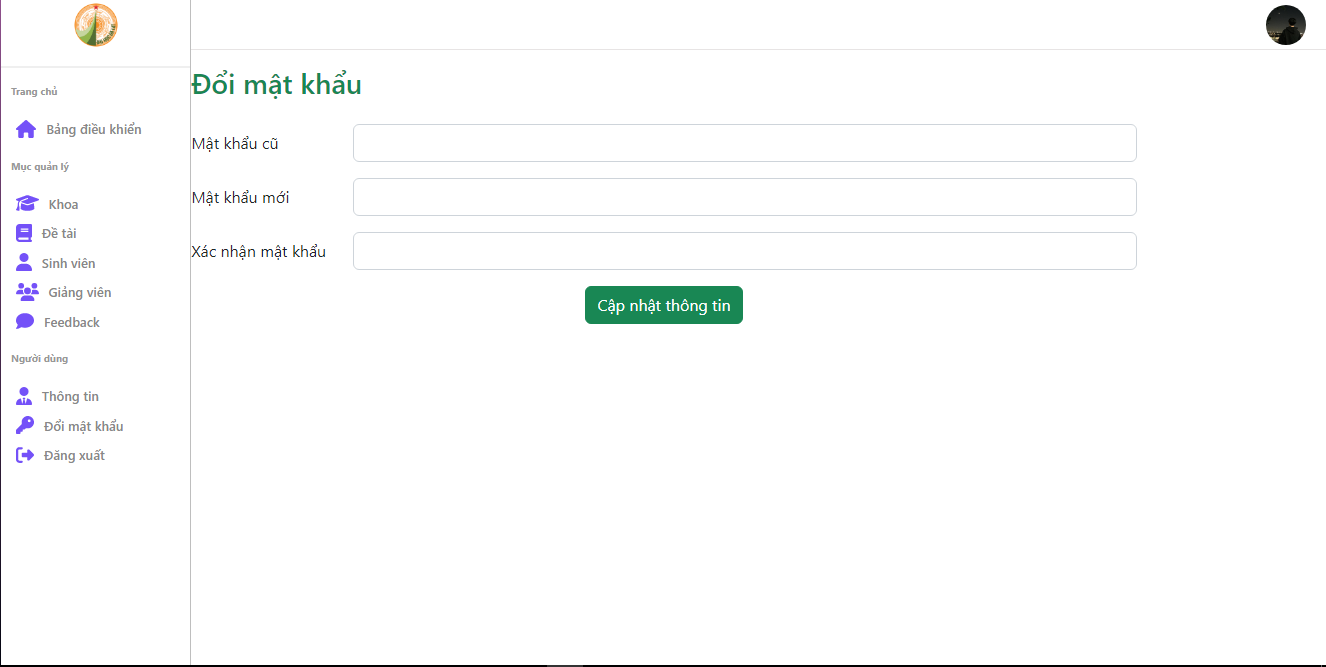
Quản lý feedback



Chỉnh sửa thông tin cá nhân (sau khi giảng viên đăng nhập)



Đổi mật khẩu



Sau khi đăng xuất



# **Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## **4.1. Kết luận**

* Đứng trước xu hướng phát triển nhanh và mạnh mẽ của công nghệ thông tin như hiện nay thì việc tin học hóa trong công tác quản lý là vô cùng quan trọng và cần thiết. Vì vậy, việc xây dựng một phần mềm quản lý các quy trình làm đồ án của sinh viên rất cần thiết. Nó giúp tiết kiệm thời gian, hạn chế những sai sót trong công tác quản lý được thuận tiện và đơn giản hơn.

## **4.2. Những mặt hạn chế khi xây dựng ứng dụng**

* Dù đã hoàn thành được các chức năng đã đề ra của ứng dụng, nhưng nhóm chúng em vẫn chưa tối ưu hóa được chúng, nguy cơ tràn bộ nhớ khá cao
* Kiến thức còn hạn hẹp
* Vốn từ tiếng Anh còn hơi ít

## **4.3. Hướng phát triển**

* Tiếp tục thực hiện và phát triển các chức năng mở rộng còn thiếu.
* Mở rộng quy mô, cơ sở dữ liệu
* Thiết kế giao diện trở nên trực quan, sinh động, dễ dàng sử dụng cho người dùng.
* Học thêm tiếng Anh để có thể tìm hiểu các tài liệu tham khảo giúp tìm kiếm các hướng dẫn để hoàn thành đề tài

**TÀI LIỆU THAM KHẢO**

**ReactJs:**

- <https://reactjs.org/docs/getting-started.html>

- [Ebook] React JS – Pro React (Cássio de Sousa Antonio)

- [Ebook] React JS – ReactJS by Example – Building Modern Web Applications with React (Packt)

- [Ebook] React JS – Fullstack React (Anthony Accomazzo, Ari Lerner, Clay Allsopp, David Guttman, Tyler Mcginnis, Nate Murray: 2020)

- [Ebook] React JS – Lập trình React thật đơn giản (VNTALKING)

- [Ebook] <https://drive.google.com/drive/folders/1AkQOVpxtRyPcppivQNqJnHxCfgEudCYV>

**ASP.NET:**

- <https://dotnet.microsoft.com/en-us/learn/aspnet>

- <https://learn.microsoft.com/vi-vn/aspnet/core/tutorials/first-mvc-app/start-mvc?view=aspnetcore-7.0&tabs=visual-studio>

- <https://tedu.com.vn/series/hoc-aspnet-core-can-ban.html>