

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO ĐỒ ÁN 2**

**Đề tài: Xây dựng hệ thống quản lý đề tài Đồ án chuyên ngành**  
**Khoa CNPM**

Giảng viên hướng dẫn:

**ThS. Trần Anh Dũng**

**ThS. Đinh Nguyễn Anh Dũng**

Sinh viên thực hiện:

**Nguyễn Huỳnh Gia Huy**

**20520544**

**Nguyễn Gia Bảo**

**20520406**

**Thành phố Hồ Chí Minh**

**ngày 30 tháng 12 năm 2023**

(Ký tên)

## LỜI CẢM ƠN

Để hoàn thành đề tài này và có kiến thức như ngày hôm nay, nhóm chúng em xin gửi lời cảm ơn đến thầy Dũng đã tận tình hỗ trợ, truyền đạt kiến thức cũng như kinh nghiệm quý báu cho chúng em trong suốt quá trình học tập và nghiên cứu.

Trong thời gian thực hiện đề tài, nhóm đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và tiếp thu những kiến thức mới. Chúng em đã cố gắng vận dụng những gì đã thu thập được để hoàn thành đồ án tốt nhất có thể. Tuy nhiên, trong quá trình thực hiện, vẫn không tránh khỏi những thiếu sót, chúng em mong nhận được sự thông cảm và góp ý chân thành từ các thầy.

Cuối cùng, chúng em xin gửi lời chúc thân ái nhất đến thầy

## Mục lục

LỜI CẢM ƠN .....	2
MỞ ĐẦU.....	7
I. GIỚI THIỆU TỔNG QUAN .....	8
1. Thông tin nhóm .....	8
2. Tổng quan đề tài .....	8
2.1. Lý do chọn đề tài .....	8
2.2. Phạm vi nghiên cứu .....	9
2.3. Mục tiêu đề tài .....	9
2.4. Nội dung nghiên cứu .....	10
2.5. Đối tượng sử dụng .....	10
2.6. Phương pháp thực hiện.....	10
2.7. Kết quả hướng tới.....	10
3. Công cụ sử dụng.....	11
4. Kế hoạch thực hiện .....	12
4.1. Công nghệ sử dụng .....	12
4.2. Phân chia công việc .....	12
II. CÔNG NGHỆ .....	15
1. ReactJs .....	15
1.1. Giới thiệu chung.....	15
1.2. Các từ khoá chính (key concepts) .....	15
1.2.1. Component-based approach .....	15
1.2.2. Virtual DOM .....	16
1.2.3. Props - State.....	16
1.2.4. Component Lifecycle .....	17
1.2.5. React Hook.....	18
1.3. Cách thức hoạt động của ReactJS .....	19
1.4. Một số thư viện - package phổ biến được sử dụng trong ReactJS .....	19
1.4.1. React Hook.....	19
1.4.2. React Router .....	21
2. ASP.NET .....	22
2.1. Giới thiệu chung.....	22
2.2. ASP.Net Core .....	22

2.3.	ASP.Net MVC .....	23
2.4.	Cấu trúc và thành phần .....	23
2.5.	Phương thức hoạt động.....	24
3.	Microsoft SQL Server .....	26
3.1.	Định nghĩa .....	26
3.2.	Thành phần .....	26
3.3.	Chức năng .....	27
III.	Website quản lý đề tài Đồ án chuyên ngành Khoa CNPM .....	28
1.	Tổng quan .....	28
2.	Yêu cầu .....	28
3.	Mã nguồn: .....	<b>Error! Bookmark not defined.</b>
4.	Phân tích kiến trúc hệ thống .....	29
4.1.	Khái niệm .....	29
4.2.	Nguyên tắc hoạt động.....	30
4.3.	Ưu điểm .....	31
4.4.	Kiến trúc hệ thống .....	31
5.	Đặc tả use case .....	32
5.1.	Use case toàn hệ thống .....	32
5.2.	Danh mục usecase .....	33
5.3.	Đặc tả .....	34
5.3.1.	Thêm giảng viên.....	34
5.3.2.	Sửa giảng viên .....	35
5.3.3.	Thêm sinh viên.....	35
5.3.4.	Sửa sinh viên .....	36
5.3.5.	Thêm đề tài.....	37
5.3.6.	Sửa đề tài.....	37
5.3.7.	Xóa đề tài .....	38
5.3.8.	Đăng nhập .....	39
5.3.9.	Tìm kiếm đề tài .....	39
5.3.10.	Tìm kiếm đồ án khóa trước .....	40
5.3.11.	Xem thông tin đồ án.....	40
5.3.12.	Thêm tiến độ dự án.....	41
5.3.13.	Cập nhật tiến độ dự án.....	42

5.3.14.	Thêm tài nguyên.....	42
5.3.15.	Xóa tài nguyên .....	43
5.3.16.	Xem điểm.....	44
5.3.17.	Đăng kí đồ án.....	44
5.3.18.	Xem thông tin sinh viên .....	45
5.3.19.	Chấm điểm đồ án.....	46
5.3.20.	Đăng xuất.....	46
6.	Thiết kế cơ sở dữ liệu.....	48
6.1.	Sơ đồ cơ sở dữ liệu .....	48
6.2.	Mô tả chi tiết.....	49
6.3.	Danh mục bảng .....	50
6.3.1.	Bảng Account .....	50
6.3.2.	Bảng AccountType.....	50
6.3.3.	Bảng Student .....	51
6.3.4.	Bảng Term.....	52
6.3.5.	Bảng Instructor.....	52
6.3.6.	Bảng Topic .....	53
6.3.7.	Bảng Tag.....	53
6.3.8.	Bảng TopicDetail .....	54
6.3.9.	Bảng TopicRegister .....	54
6.3.10.	Bảng Project .....	55
6.3.11.	Bảng ProjectDetail.....	55
6.3.12.	Bảng ProjectResources.....	56
6.3.13.	Bảng Subject.....	56
6.3.14.	Bảng ProjectProgress .....	56
6.3.15.	Bảng RegisterCalendar .....	57
6.3.16.	Bảng CurrentSubject .....	57
7.	Thiết kế giao diện .....	59
7.1.	Sơ đồ liên kết giao diện .....	59
7.2.	Danh sách giao diện.....	60
7.3.	Mô tả chi tiết.....	62
7.3.1.	Màn hình đăng nhập .....	62
7.3.2.	Màn hình danh sách đề tài .....	63

7.3.3.	Màn hình thêm đề tài từ excel .....	64
7.3.4.	Màn hình danh sách giảng viên .....	65
7.3.5.	Màn hình thông tin giảng viên.....	66
7.3.6.	Màn hình danh sách sinh viên .....	67
7.3.7.	Màn hình thông tin sinh viên.....	68
7.3.8.	Màn hình danh sách đăng kí đồ án 1 và 2.....	69
7.3.9.	Màn hình cài đặt.....	70
7.3.10.	Màn hình chi tiết đồ án.....	71
7.3.11.	Màn hình chấm điểm .....	71
7.3.12.	Màn hình danh sách keyword.....	73
7.3.13.	Màn hình chỉnh sửa thông tin cá nhân sinh viên.....	74
7.3.14.	Màn hình chỉnh sửa đồ án.....	75
7.3.15.	Màn hình thêm tiến độ .....	77
7.3.16.	Màn hình tìm kiếm đề tài và đăng kí .....	77
7.3.17.	Màn hình tìm kiếm đồ án các khóa trước .....	79
7.3.18.	Màn hình chi tiết đồ án (người dùng) .....	79
7.3.19.	Màn hình chỉnh sửa thông tin giảng viên .....	81
7.3.20.	Màn hình danh sách đề tài bản thân quản lý .....	81
IV.	Cài đặt và thử nghiệm.....	83
V.	Nhận xét và kết luận .....	84
1.	Nhận xét.....	84
2.	Kết luận.....	84
2.1.	Ưu nhược điểm .....	84
2.2.	Hướng phát triển .....	84
VI.	Bảng phân công công việc:.....	85
VII.	Phụ lục .....	86

## MỞ ĐẦU

Hiện nay, nền công nghiệp 4.0 đang ngày càng phát triển, thúc đẩy các hoạt động phát triển theo, đặc biệt là các thông tin, tin tức về các ngành nghề, lĩnh vực.

Việc quản lý đồ án tốt nghiệp sinh viên là một yêu cầu thiết yếu của mỗi trường đại học. Để quản lý đồ án và lưu trữ có hiệu quả thì không đơn giản bởi đòi hỏi kỹ năng của người quản lý. Làm sao để vừa có thể kiểm soát được số lượng đề tài của các khóa, các hệ đào tạo khác nhau vừa có thể lưu trữ nó làm tài liệu cho các bạn sinh viên khóa sau đồng thời phân công giảng viên hướng dẫn đồ án một cách hợp lý. Việc quản lý và lưu trữ đồ án trước đây chủ yếu sử dụng phương pháp thủ công, bản mềm, sourcecode đồ án thường lưu trên đĩa ... chiếm diện tích lưu trữ rất lớn. Do đó, khi quản lý hay tìm kiếm mất rất nhiều thời gian và công sức mà hiệu quả công việc đem lại không cao đôi khi còn xảy ra sai sót mất mát dữ liệu không đáng có.

Hiện nay công tác quản lý đồ án tốt nghiệp sinh viên của một số trường đại học nói chung và khoa CNPM Trường đại học Công Nghệ Thông Tin nói riêng còn chưa đạt hiệu quả cao. Do đó việc đòi hỏi có một phần mềm chuyên dụng trợ giúp cho công việc quản lý và lưu trữ đồ án tốt nghiệp sinh viên là một nhu cầu tất yếu để đảm bảo hiệu quả và tiết kiệm thời gian trong công việc.



# I. GIỚI THIỆU TỔNG QUAN

## 1. Thông tin nhóm

MSSV	Họ tên	Email
20520544	Nguyễn Huỳnh Gia Huy	20520544@gm.uit.edu.vn
20520406	Nguyễn Gia Bảo	20520406@gm.uit.edu.vn

## 2. Tổng quan đề tài

### 2.1. Lý do chọn đề tài

Trong những năm gần đây, nền công nghệ thông tin của nước ta cũng đã có phát triển trên mọi lĩnh vực trong cuộc sống cũng như trong lĩnh vực quản lý xã hội khác. Một trong những lĩnh vực mà máy tính được sử dụng nhiều nhất là các hệ thống thông tin quản lý nói chung. Tuy nhiên, hiện nay do quy mô, tính phức tạp của công việc ngày càng cao nên việc xây dựng hệ thống thông tin quản lý không chỉ là việc lập trình đơn giản mà phải xây dựng một cách có hệ thống

Việc quản lý đồ án sinh viên là một yêu cầu thiết yếu đối với khoa công nghệ phần mềm. Để quản lý đồ án và lưu trữ có hiệu quả thì không đơn giản bởi đòi hỏi kỹ năng của người quản lý. Làm sao để vừa có thể kiểm soát được số lượng đề tài của các khóa vừa có thể lưu trữ nó làm tài liệu cho các bạn sinh viên khóa sau đồng thời phân công giảng viên hướng dẫn đồ án một cách hợp lý.

Do đó việc đòi hỏi có một phần mềm chuyên dụng trợ giúp cho công việc quản lý và lưu trữ đồ án sinh viên là một nhu cầu tất yếu để đảm bảo hiệu quả và tiết kiệm thời gian trong công việc. nên chúng em đã thực hiện đề tài: ” Xây dựng hệ thống quản lý đề tài Đồ án chuyên ngành Khoa CNPM” với

mong muốn đưa lĩnh vực công nghệ thông tin trở nên thiết thực với cuộc sống và hỗ trợ công tác quản lý và lưu trữ đồ án tốt nghiệp sinh viên khoa CNPM- Trường đại học Công nghệ thông tin được dễ dàng và chính xác hơn, đồng thời sẽ là tài liệu tham khảo cho các bạn sinh viên khóa sau.

## 2.2. Phạm vi nghiên cứu

Trong đồ án lần này, nhóm chủ yếu tập trung vào nghiên cứu tổng quan yêu cầu về chức năng và giao diện của một trang web đăng kí đồ án . Đồng thời, nhóm áp dụng các kiến thức về nền tảng react và asp.Net Core vào ứng dụng để tăng thêm khả năng bảo trì và tái sử dụng ứng dụng

**Phạm vi môi trường:** Triển khai sản phẩm đề tài trên môi trường web

**Phạm vi chức năng:**

- Giao diện thân thiện, trực quan, tiện dụng và dễ dàng thao tác cho người dùng, có thể thay đổi giữa
- Chức năng quản lý đề tài
- Chức năng đăng kí đề tài
- Thông báo thời gian đăng kí đề tài
- Quản lý quá trình thực hiện đồ án, upload tài nguyên dự án
- Thống kê điểm đồ án

## 2.3. Mục tiêu đề tài

- ❖ Xây dựng hệ thống quản lý đăng ký đề tài
  - Giảng viên đăng đề tài
  - Sinh viên tiến hành đăng kí
  - Khoa tiến hành xét duyệt đăng kí cho sinh viên
- ❖ Xây dựng hệ thống quản lý trong quá trình thực hiện đồ án
  - Cập nhật tiến độ thực hiện
  - Upload tài nguyên dự án
- ❖ Tra cứu thông tin về các đồ án do các khóa trước thực hiện

- ❖ Cung cấp giao diện biểu đồ thống kê điểm đồ án theo từng khóa
- ❖ Giao diện trực quan, thân thiện người dùng

## 2.4. Nội dung nghiên cứu

Nhóm sẽ tiến hành khảo sát nghiên cứu chi tiết về các yêu cầu chức năng, thiết kế giao diện cho website tìm kiếm thành viên đội nhóm dự án, cải tiến hơn so với ứng dụng hiện tại trên nền tảng công nghệ react và asp.Net Core

## 2.5. Đối tượng sử dụng

Tất cả sinh viên, giảng viên của khoa công nghệ phần mềm Đại học Công Nghệ Thông Tin – ĐHQG TPHCM

## 2.6. Phương pháp thực hiện

- **Phương pháp làm việc:**

Làm việc nhóm 2 thành viên thông qua cả 2 phương thức online và offline dưới sự hướng dẫn của giảng viên hướng dẫn.

- **Phương pháp nghiên cứu:**

Phân tích nhu cầu sử dụng của đối tượng người dùng.

Nghiên cứu tài liệu các công nghệ liên quan.

- **Phương pháp công nghệ:**

Tìm hiểu về công nghệ Reactjs, Asp.Net Core, WebApi

Xây dựng giao diện website bằng figma để người dùng có thể tương tác với ứng dụng

Tìm hiểu về hệ thống khuyến nghị

Quản lý source code thông qua Github

Sử dụng Google doc, Office 365 Word... để quản lý tiến trình và tài liệu

## 2.7. Kết quả hướng tới

Với đề tài này, nhóm đề ra hai mục tiêu chính:

- **Đối với cá nhân các thành viên trong nhóm:** mở rộng kiến thức của thành viên trong nhóm về lĩnh vực tìm kiếm nhân sự thông qua quá trình tìm hiểu, khảo sát và xây dựng ứng dụng cả về mặt giao diện và chức năng. Các kiến thức tìm hiểu được thông qua đồ án thúc đẩy nhóm phát triển cải tiến thêm các chức năng mới cũng như nâng cao khả năng tương tác với người dùng. Đồng thời củng cố thêm kiến thức về công nghệ thư viện reactJS, asp.Net Core. Từ đó nhóm học được các quy trình xây dựng một website hoàn chỉnh cũng như cách thức sử dụng một framework mới cần trải qua quá trình gì nhằm giúp nhóm dễ dàng tiếp xúc với các công nghệ mới hơn trong tương lai.
- **Đối với các lập trình viên dùng đồ án nghiên cứu này làm tài liệu tham khảo:** thông qua tài liệu nghiên cứu và ứng dụng nhóm đã xây dựng, các lập trình viên khác có thể dễ dàng định hướng cần phải xây dựng những chức năng gì cho một ứng dụng trên lĩnh vực đăng kí đồ án. Đồng thời nhóm có ghi một số khái niệm, kiến thức cơ bản và thư viện phổ biến được đề xuất bởi cộng đồng khi sử dụng reactJS và asp.Net Core, các lập trình viên khác có thể tham khảo và tìm hiểu sâu hơn về loại công nghệ để từ đó rút ngắn thời gian xây dựng ứng dụng

### 3. Công cụ sử dụng

Trong quá trình xây dựng phần mềm, nhóm đã sử dụng phần mềm sau:

- **Eclipse IDE:** IDE hỗ trợ xây dựng ứng dụng phía server
- **Visual Studio Code:** hỗ trợ xây dựng giao diện người dùng
- **Postman:** hỗ trợ nhóm test các api để lấy dữ liệu từ phía server.
- **Microsoft SQL Server:** hỗ trợ nhóm lưu trữ cơ sở dữ liệu.
- **Figma:** hỗ trợ nhóm thiết kế giao diện.

## 4. Kế hoạch thực hiện

### 4.1. Công nghệ sử dụng

- Front-end: ReactJs
- Back-end: ASP.Net Core
- Database: Microsoft SQL Server

### 4.2. Phân chia công việc

Sprint 1: 23/9/2023 – 7/10/2023

Sprint 2: 7/10/2023 – 21/10/2023

Sprint 3: 21/10/2023 – 4/11/2023

Sprint 4: 4/11/2023 – 11/11/2023

Sprint 5: 11/11/2023 – 25/11/2023

Sprint 6: 25/11/2023 – 9/12/2023

Sprint 7: 9/12/2023 – 23/12/2023

Sprint 8: 23/12/2023 – 30/12/2023

Tuần	Công việc thành viên 1 <i>Nguyễn Huỳnh Gia Huy</i>	Công việc thành viên 2 <i>Nguyễn Gia Bảo</i>
<b>Sprint 1:</b>	Tìm hiểu và xác định phạm vi, hướng phát triển của đề tài Tìm hiểu và phân tích các yêu cầu nghiệp vụ của đề tài Viết đề cương chi tiết	Tìm hiểu về công nghệ Viết tài liệu liên quan đến công nghệ

<b>Sprint 2:</b>	Xác định phạm vi dự án Khảo sát yêu cầu nghiệp vụ dự án	Khảo sát yêu cầu hệ thống Khảo sát yêu cầu chất lượng
<b>Sprint 3:</b>	Mô hình hóa dữ liệu Xác định sơ đồ luồng dữ liệu Thiết kế use case tương ứng Đặc tả use case	Xác định kiến trúc hệ thống phù hợp Thiết kế biểu mẫu và quy định
	Vẽ sơ đồ ERD Thiết kế cơ sở dữ liệu Kết nối cơ sở dữ liệu	
<b>Sprint 4:</b>	Xác định các thành phần giao diện Sử dụng figma để thiết kế giao diện cho sản phẩm Thực hiện việc đánh giá và cải thiện bản thiết kế để đảm bảo tính thẩm mỹ, trải nghiệm người dùng và đảm bảo tính thực tiễn cho ứng dụng.	
<b>Sprint 5:</b>	Lập trình giao diện các chức năng cơ bản của ứng dụng: <ul style="list-style-type: none"> <li>• Đăng nhập</li> <li>• Đăng kí</li> <li>• Cài đặt</li> <li>• Thêm xóa sửa đề tài</li> <li>• Trang thông tin cá nhân</li> </ul> Kiểm thử đơn vị	Lập trình chức năng cơ bản của ứng dụng: <ul style="list-style-type: none"> <li>• Đăng nhập</li> <li>• Đăng kí</li> <li>• Cài đặt</li> <li>• Thêm xóa sửa đề tài</li> <li>• Trang thông tin cá nhân</li> </ul> Kiểm thử đơn vị
<b>Sprint 6:</b>	Lập trình giao diện các chức năng phức tạp của ứng dụng: <ul style="list-style-type: none"> <li>• Quản lý việc đăng kí đề tài sinh viên</li> <li>• Quản lý quá trình thực hiện đề</li> </ul>	Lập trình giao diện các chức năng phức tạp của ứng dụng: <ul style="list-style-type: none"> <li>• Quản lý việc đăng kí đề tài sinh viên</li> <li>• Quản lý quá trình thực hiện</li> </ul>

	án <ul style="list-style-type: none"> <li>Quản lý lịch sử đồ án</li> </ul> Kiểm thử đơn vị	đồ án <ul style="list-style-type: none"> <li>Quản lý lịch sử đồ án</li> </ul> Kiểm thử đơn vị
<b>Sprint 7:</b>	Hoàn thiện giao diện các chức năng còn lại Kiểm thử đơn vị	Hoàn thiện các chức năng còn lại Kiểm thử đơn vị
<b>Sprint 8:</b>	Triển khai ứng dụng Viết tài liệu hướng dẫn sử dụng Lập kế hoạch bảo trì Viết báo cáo	Thiết kế kế hoạch kiểm thử Viết test cases Thực thi test cases Thực hiện fix các lỗi

## II. CÔNG NGHỆ

### 1. ReactJs



#### 1.1. Giới thiệu chung

React là một thư viện Javascript, được phát triển bởi Facebook, hỗ trợ xây dựng giao diện người dùng.

Một ứng dụng có giao diện xây dựng bằng React JS sẽ được tạo bởi nhiều thành phần nhỏ (component), trong đó, output của mỗi component là một đoạn code HTML nhỏ và có thể tái sử dụng xuyên suốt ứng dụng. Component được xem là từ khoá chính (key concept) khi nhắc đến React. Mỗi component có thể chứa các component nhỏ hơn, do đó, một ứng dụng với cấu trúc phức tạp sẽ được hình thành từ nhiều component con. Đồng thời, chỉ cần định nghĩa component con 1 lần và có thể sử dụng ở nhiều nơi.

Một lợi thế khi tìm hiểu ReactJS chính là sau khi có đầy đủ kiến thức cơ bản về ReactJS, lập trình viên có thể dễ dàng học tiếp React Native - framework hỗ trợ xây dựng ứng dụng di động đa nền tảng.

#### 1.2. Các từ khoá chính (key concepts)

##### 1.2.1. Component-based approach

Đây là concept chính của React. Ứng dụng được chia thành các component nhỏ, điều này giúp lập trình viên dễ dàng tái sử dụng các



component này và có thể quản lý, bảo trì và phát triển ứng dụng một cách dễ dàng. Concept này giúp đỡ rất nhiều trong việc quản lý một ứng dụng lớn và phức tạp, bởi lẽ khi xảy ra lỗi, các lập trình viên chỉ cần thay đổi và chỉnh sửa trong một component nhỏ mà không ảnh hưởng đến code của toàn bộ ứng dụng.

### 1.2.2. Virtual DOM

Virtual DOM chỉ là một đại diện ảo của DOM, Virtual DOM ra đời đã tăng hiệu suất các ứng dụng ReactJS một cách đáng kể.

Cách thức hoạt động của virtual DOM trong ReactJS:

- Ứng dụng React được tạo bởi rất nhiều component con, mỗi component này đều chứa một state và React sẽ theo dõi trạng thái của state. Khi state của một component thay đổi, React sẽ so sánh sự khác nhau giữa phiên bản hiện tại và trước đó thông qua thuật toán Diff, đồng thời thực hiện cập nhật trên virtual DOM.
- Một khi nắm được component nào thay đổi, React sẽ tiến hành cập nhật duy nhất component đó trên DOM. Điều này đã hạn chế được việc cập nhật toàn bộ component trên DOM một cách thường xuyên, tăng hiệu năng của ứng dụng web.

### 1.2.3. Props - State

#### 1.2.3.1. Props:

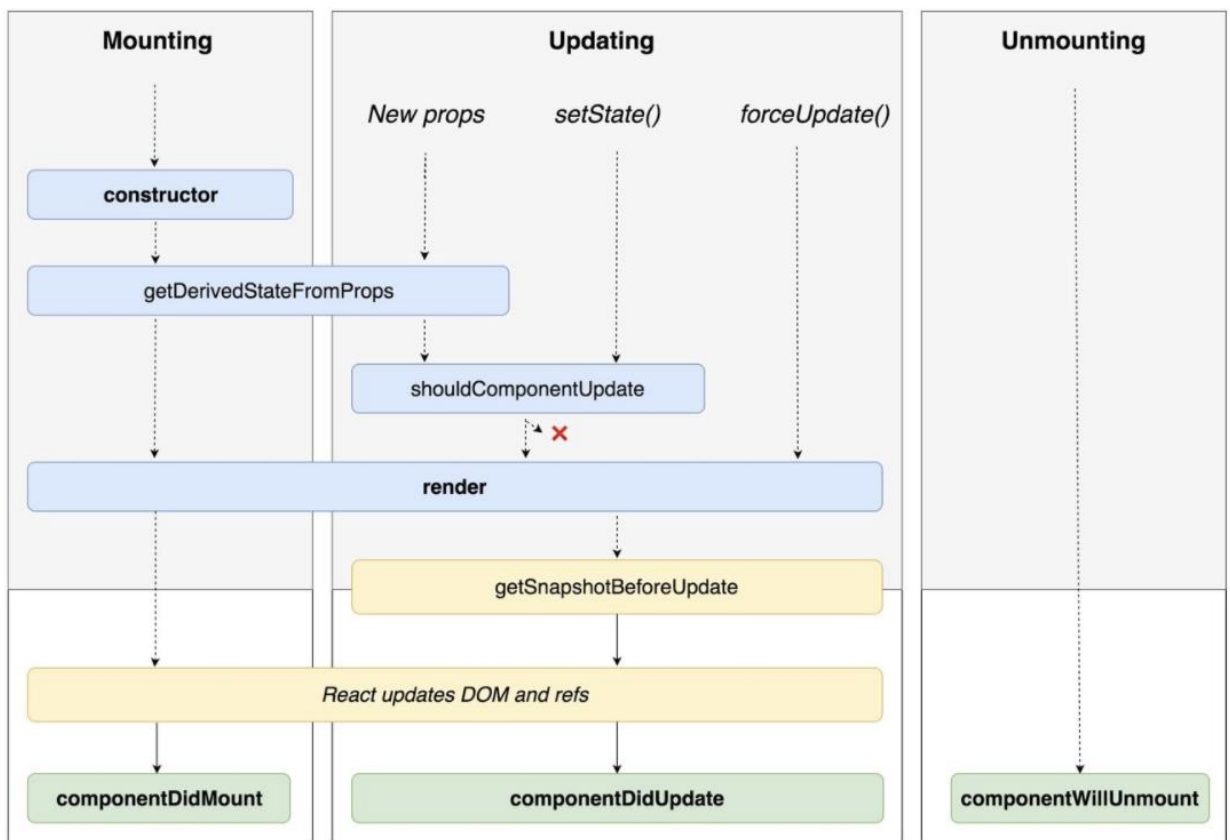
- Để truyền dữ liệu giữa các component, ReactJS cho ra đời khái niệm props (hay còn gọi là properties).
- Đối với ReactJS, component con không được thay đổi giá trị của props, bởi vì, props có thể được truyền cho rất nhiều component con,

nếu các component con thay đổi props sẽ dẫn đến nhập nhằng khi testing, dữ liệu giữa các component không thống nhất.

### 1.2.3.2. *State:*

- Nếu props dùng để hứng dữ liệu từ component cha thì state được dùng để lưu dữ liệu của chính component đó với giá trị có thể thay đổi phụ thuộc vào logic bên trong component đó.
- Tuy nhiên, khi sử dụng thao tác cập nhật giá trị của state, lập trình viên phải chú ý nguyên tắc mỗi khi state thay đổi, component sẽ được render lại. Do đó, cần cân nhắc trong việc thay đổi giá trị state nhằm tránh việc render liên tục, giảm hiệu suất ứng dụng.

### 1.2.4. Component Lifecycle



Để có thể hiểu rõ cách thức hoạt động của một component, lập trình viên cần nắm rõ vòng đời (lifecycle) của React component. Để có thể quản lý dữ liệu từ component cha (props), dữ liệu nội tại (state) và các điều kiện logic khiến component re-render, lập trình viên phải thực sự hiểu rõ định nghĩa và cách sử dụng các phương thức trong lifecycle.

### **Vòng đời của một component được chia làm 3 nhóm chính:**

**Mounting:** Khi một component được khởi tạo, nó sẽ thực thi các câu lệnh dưới đây theo thứ tự nhằm thêm component đó vào DOM để render

- constructor() được sử dụng để khởi tạo các state cho component, do đó, tránh cập nhật giá trị state tại hàm này vì sẽ gây render liên tục.
- render()
- componentDidMount() là nơi thực hiện các side effect như gọi api lấy dữ liệu, thực hiện một số thao tác lên DOM,...

**Updating:** Khi state hoặc props của component thay đổi, các hàm trong nhóm này sẽ được thực thi và quyết định xem có cần phải render lại UI hay không. Trường hợp UI không được re-render khi hàm shouldComponentUpdate() trả về giá trị false.

**Unmounting:** Khi một component bị xóa khỏi DOM, các hàm thuộc nhóm này sẽ được thực thi. Đây nơi là để xử lý các biến thời gian như setInterval, ngắt các network request.

#### **1.2.5. React Hook**

Đối với các lập trình viên vừa tiếp xúc với JS, việc sử dụng class component và các phương thức trong lifecycle rất dễ gây ra lỗi. Trong khi đó, các functional component thì không thể sử dụng state và lifecycle nhưng

lại dễ sử dụng và dễ học hơn class component cho những người mới bắt đầu. Do vậy, React Hook ra đời nhằm giải quyết các vấn đề trên, các hooks cho phép kết nối state và các lifecycle vào functional component.

Ở đồ án này, nhóm đã sử dụng React Hook để các component trở nên gọn nhẹ, dễ đọc, dễ sửa chữa và mở rộng.

### 1.3. Cách thức hoạt động của ReactJS

ReactJS thường được sử dụng để tạo ra giao diện cho các SPA - single page application. Khi sử dụng ReactJS, ứng dụng web được chia thành từng component nhỏ, mỗi component chứa hai thành phần chính là state và props. Dữ liệu được truyền từ component cha sang con gọi là props, trong khi đó state được định nghĩa là dữ liệu của riêng component đó. Component chỉ được re-render khi state của nó thay đổi, chính vì lý do này, ứng dụng không render lại toàn bộ trang web mà chỉ re-render component mà người dùng thao tác lên. Chính vì nguyên tắc hoạt động trên cùng với các component chỉ định nghĩa một lần mà có thể sử dụng ở nhiều nơi, ReactJS đã tăng hiệu suất của ứng dụng web và trở thành thư viện phát triển giao diện được nhiều người sử dụng.

### 1.4. Một số thư viện - package phổ biến được sử dụng trong ReactJS

#### 1.4.1. React Hook

1.4.1.1. **Khái niệm:** Hooks là những hàm cho phép kết nối React state và lifecycle vào các components sử dụng hàm. Với Hooks có thể sử dụng state và lifecycles mà không cần dùng ES6 class. Sự ra đời này đã giúp các lập trình viên tránh sự nhập nhằng về con trỏ this

1.4.1.2. **Các hook phổ biến:** useState, useEffect, useRef.

**a. useState:** cập nhật giá trị của state, trong class component để thay đổi state phải dùng cú pháp this.state. Tuy nhiên với cú pháp này sẽ bị nhập nhằng con trỏ this, do đó, hook useState ra đời để giảm bớt sự nhập nhằng

trên. Để thay đổi state trong functional component chỉ cần dùng hàm `setState()`

**b. *useEffect*:** là nơi thích hợp để xử lý các side effect như gọi api lấy dữ liệu từ server, các thao tác liên quan đến DOM cũng như các vấn đề liên quan đến `setInterval`, network request.

```
useEffect(() => {
  effect
  return () => {
    cleanup
  }
}, [filter])
```

Một `useEffect` sẽ gồm 3 thành phần: side effect, dependencies và cleanup:

- **Side Effect:** những ảnh hưởng từ bên ngoài vd gọi API, tương tác dom hay `setInterval`.
- **Cleanup:** Là nơi xử lý các tác vụ liên quan đến hủy network request, các biến thời gian. Một ví dụ điển hình cho phần cleanup: nếu có `timeInterval` mà ko clear sau khi unmount component thì khi unmount component rồi những `setInterval` vẫn chạy và vẫn update state sẽ dẫn đến lỗi đã unmount rồi mà vẫn cố gắng update state => phải clear `setInterval` trong hàm cleanup của use effect.
- **Có 3 loại dependencies:**
  - Không truyền dependencies vào thì use effect chạy lại sau khi hàm render được gọi

- Nếu truyền vào mảng rỗng [] thì sẽ giống với hàm `componentDidMount`, `use effect` chỉ chạy 1 lần, `side effect` chạy sau render lần đầu và `clean up` chỉ chạy khi `compo unmount`.
- Truyền filter vào mảng `dependencies`: chỉ chạy `use effect` khi filter thay đổi. Ví dụ thanh search, gọi api theo filter của thanh search, filter thay đổi thì sẽ gọi api (chạy lại `use effect`), còn filter ko đổi thì ko gọi api (ko chạy `use effect`).

**c. *useRef*:** dùng để lưu trữ giá trị của một biến qua các lần render. Sau mỗi lần render, giá trị của một số biến trong component sẽ quay lại giá trị ban đầu, sử dụng `useRef` để lưu trữ các giá trị này

#### 1.4.2. React Router

1.4.2.1. ***Khái niệm:*** là một thư viện định tuyến, với Url này sẽ tương đương với Route này và render ra giao diện tương ứng.

1.4.2.2. ***Các tính năng - tiện ích:*** `<Link>`, `<NavLink>`, `<Redirect>`, `useLocation`, `useParams`

**a. *<Link> <NavLink>*:** đối với HTML, cặp thẻ để chuyển hướng là `<a></a>`, còn ở React thì dùng `<Link>`, `<NavLink>`

**b. *<Redirect>*:** dùng để chuyển trang, có thể lấy thông tin các trang trước để sử dụng thông qua `useLocation`

**c. *useParams*** (chỉ có ở React router v5): dùng để lấy thông tin các params trên url

**d. *useLocation*** (chỉ có ở React router v5): với react router v4, để lấy được thông tin từ url (params, pathname, ...) thì phải dùng đối tượng `location` được truyền như props vào component. Tuy nhiên, ở v5, chỉ cần

dùng hook useLocation thì vấn đề này có thể giải quyết mà không cần truyền đối tượng location

## 2. ASP.NET



### 2.1. Giới thiệu chung

ASP.NET là một mã nguồn mở dành cho web được tạo bởi Microsoft. Hiện mã nguồn này chạy trên nền tảng Windows và được bắt đầu vào đầu những năm 2000. ASP.NET cho phép các nhà phát triển tạo các ứng dụng web, dịch vụ web và các trang web động.

### 2.2. ASP.Net Core

ASP.NET Core là một phiên bản mới của ASP.NET chạy trên mọi nền tảng mọi máy tính, bao gồm Windows, MacOS và Linux. Giống như ASP.NET, nó là mã nguồn mở, được tạo bởi Microsoft. Mã nguồn cho phép các nhà phát triển tạo app, dịch vụ web và các trang web động.

Được phát hành lần đầu tiên vào năm 2016, ASP.NET Core tương đối mới, nhưng đã nhận được cập nhật bản 2.0. Hiện là một giải pháp thay thế ổn định cho các ứng dụng web ASP.NET được lưu trữ trên Windows.

### 2.3. ASP.Net MVC

ASP.NET MVC là một framework được phát triển bởi Microsoft để xây dựng ứng dụng web dựa trên mô hình MVC (Model-View-Controller). Đây là một phần của nền tảng ASP.NET, được thiết kế để tạo ra các ứng dụng web linh hoạt, dễ bảo trì và kiểm thử.

MVC là viết tắt của “Model-View-Controller,” một mô hình thiết kế phần mềm được sử dụng để phân chia logic và trách nhiệm trong ứng dụng. Mô hình này giúp tổ chức mã nguồn một cách có tổ chức và dễ quản lý.

Các thành phần chính trong ASP.NET MVC bao gồm:

**Model:** Đại diện cho dữ liệu và logic xử lý dữ liệu của ứng dụng. Model đóng vai trò như là lớp chứa dữ liệu và các phương thức để truy cập hoặc xử lý dữ liệu.

**View:** Là thành phần hiển thị giao diện người dùng. Nó đảm nhiệm nhiệm vụ hiển thị dữ liệu từ Model và nhận đầu vào từ người dùng. View không thực hiện logic xử lý, mà chỉ làm nhiệm vụ hiển thị dữ liệu theo cách được định nghĩa.

**Controller:** Là thành phần chứa logic điều khiển. Nó xử lý các sự kiện và tương tác của người dùng, sau đó gọi các phương thức tương ứng trong Model để thay đổi dữ liệu và cập nhật View. Controller làm nhiệm vụ điều phối giữa Model và View

### 2.4. Cấu trúc và thành phần

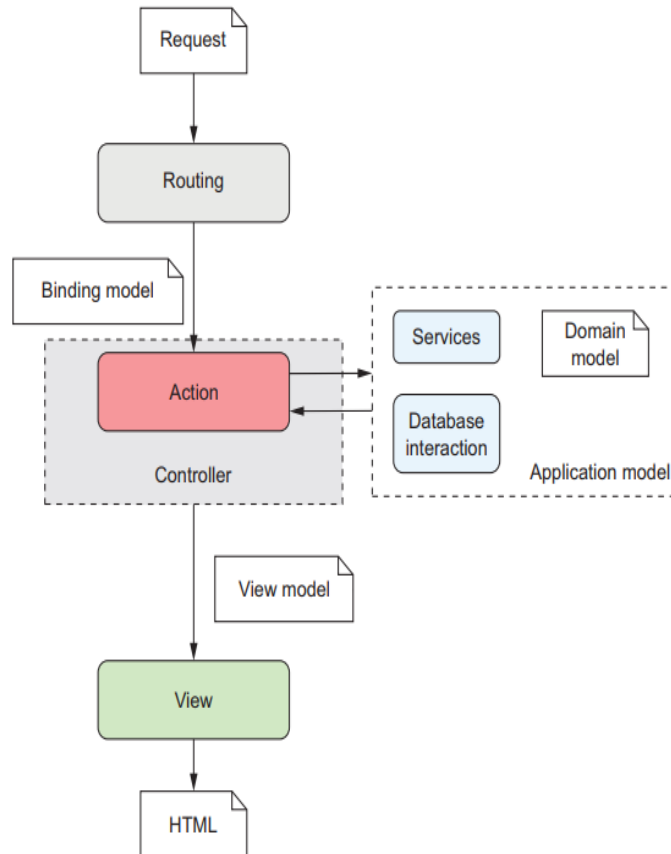
**Language/Ngôn ngữ:** Là tập con của .NET Framework, tại đây có rất nhiều ngôn ngữ lập trình khác nhau phải kể đến như C#, VB.net, PHP, JavaScript,... Trong đó C# và VB.net được sử dụng phổ biến nhất trong ứng dụng phát triển ứng dụng web.

**Library/Thư viện:** ASP.NET Framework gồm có một bộ các lớp library chuẩn. Web library là thư viện được dùng phổ biến nhất cho các ứng dụng web. Web library bao gồm tất cả các thành phần cần thiết sử dụng trong phát triển các ứng dụng web-based.

**Common Language Runtime/Thời gian chạy CLR:** CLR là một trong các cơ sở hạ tầng của phần đông các kiểu ngôn ngữ lập trình phổ thông. Trong đó, CLR thực hiện các tác vụ chính để xử lý các trường hợp đặc biệt và thu gom rác.

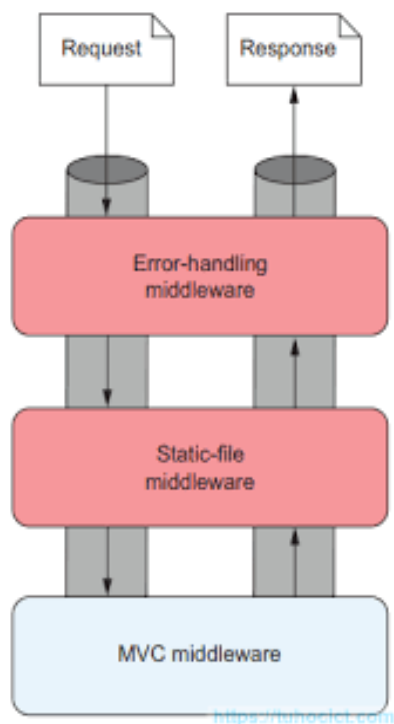


## 2.5. Phương thức hoạt động



- Truy vấn HTTP sẽ được cơ chế routing ánh xạ sang một phương thức xác định gọi là action.
- Action là phương thức được thực thi để đáp ứng lại một truy vấn.
- Binding model là một object đóng vai trò “thùng chứa” dữ liệu trích xuất ra từ truy vấn để cung cấp cho action. Binding model là kết quả hoạt động của cơ chế model binding và là tham số đầu vào cho action.
- Controller trong ASP.NET Core là class chứa các action có quan hệ nhất định.
- Action khi thực thi sẽ tương tác với các thành phần còn lại của ứng dụng như các dịch vụ, cơ sở dữ liệu.
- Với cách tiếp cận DDD (Domain-driven Design), phần dữ liệu nghiệp vụ được thể hiện qua các domain model.

- Tất cả các thành phần dịch vụ, domain model, v.v., được gọi chung là application model.
- Quá trình tương tác này sẽ sinh ra dữ liệu phục vụ cho hiển thị, gọi là view model.
- View model là object đơn giản chứa dữ liệu cần thiết để sinh ra giao diện. Thông thường view model là một biến thể của dữ liệu lấy được từ application model cùng với dữ liệu phụ trợ cho hiển thị (như tiêu đề, phân trang, v.v.).
- View trong ASP.NET Core MVC là các trang Razor chứa loại mã hỗn hợp C# + HTML theo cú pháp Razor. Kết quả xử lý của Razor view là HTML.



MVC MIDDLEWARE đặt ở cuối chuỗi xử lý của ASP.NET Core.

Tất cả các thành phần cần thiết cho một ứng dụng MVC như routing, base controller class, model binding, Razor view engine, v.v., đều được thực thi sẵn.

### 3. Microsoft SQL Server



#### 3.1. Định nghĩa

SQL là viết tắt của từ Structured Query Language, đây là một dạng ngôn ngữ truy vấn có cấu trúc. SQL là tập hợp các câu lệnh để tương tác với cơ sở dữ liệu. Trong thực tế, SQL là một ngôn ngữ truy vấn tiêu chuẩn, được dùng trong hầu hết các hệ quản trị cơ sở dữ liệu như: SQL Server, MySQL, Oracle, MS Access,...

SQL Server hay Microsoft SQL Server, là viết tắt của MS SQL Server. Đây là một phần mềm được phát triển bởi Microsoft dùng để lưu trữ, thao tác với dữ liệu theo tiêu chuẩn RDBMS (Relational Database Management System).

#### 3.2. Thành phần

SQL Server được cấu thành từ 7 thành phần khác nhau, mỗi thành phần mang một chức năng riêng. Tất cả tạo nên một hệ thống SQL Server hoàn chỉnh, đáp ứng chức năng phân tích và lưu trữ dữ liệu.

- Database Engine: Có chức năng lưu trữ dữ liệu dưới dạng table và support, có thể tự điều chỉnh được.
- Integration Services: Là tập hợp các đối tượng lập trình, các công cụ đồ họa, nhằm thực hiện các thao tác sao chép, di chuyển, chuyển đổi dữ liệu.
- Analysis Services: Là dịch vụ phân tích dữ liệu bằng kỹ thuật khai thác hình khối đa chiều và dữ liệu có sẵn.

- **Notification Services:** Là nền tảng phát triển và triển khai các ứng dụng soạn và gửi thông báo. Ngoài ra, dịch vụ này cũng cung cấp tính năng gửi thông báo đến hàng ngàn người đăng ký sử dụng trên các thiết bị khác nhau.
- **Reporting Services:** Là một công cụ có chức năng tạo, quản lý và triển khai các báo cáo cho Server và Client. Đây cũng là nền tảng đầy xây dựng ứng dụng báo cáo.
- **Full Text Search Services:** Đây là thành phần đặc biệt trong việc truy vấn và đánh giá các chỉ mục dữ liệu văn bản được lưu trữ trong database.
- **Service Broker:** Là môi trường lập trình tạo ra các ứng dụng.

### 3.3. Chức năng

Microsoft đã giới thiệu về SQL Server là sản phẩm được phát triển với mục đích chính nhằm lưu trữ dữ liệu.

Bên cạnh đó, nó còn cung cấp những tính năng giúp người dùng làm việc hiệu quả hơn, cụ thể là:

- Giúp lưu trữ dữ liệu bền vững, không bị mất dữ liệu.
- Cho phép tạo và lưu trữ dữ liệu với dung lượng lớn.
- Cung cấp khả năng sử dụng SSAS để phân tích dữ liệu.
- Khả năng bảo mật dữ liệu cao.
- Hỗ trợ tạo báo báo bằng SSRS-SQL dễ dàng.
- Thực hiện các quá trình bằng SSIS-SQL.

### III. Website quản lý đề tài Đồ án chuyên ngành Khoa CNPM

#### 1. Tổng quan

Hệ thống là trang web cung cấp các chức năng như đăng kí đồ án, tìm kiếm đề tài đăng kí và tham khảo các đồ án của các khóa trước

Ngoài ra trang web còn có thêm chức năng upload tài nguyên, cập nhật tiến độ dự án cho phép giảng viên dễ dàng theo dõi quá trình làm việc của sinh viên

Có hệ thống xét duyệt đăng kí tự động giúp rút ngắn thời gian chờ đợi cho sinh viên

Hệ thống tìm kiếm dựa vào tên đồ án, giảng viên cũng như có các thẻ đề thao tác dễ dàng và đa dạng

Trang web lưu trữ đầy đủ thông tin các đồ án qua các năm là nguồn thư viện cho sinh viên các khóa tham khảo

#### 2. Yêu cầu

Về mặt chức năng:

- ❖ Đăng nhập, quản lý thông tin cá nhân
- ❖ Quản trị viên tạo và cung cấp tài khoản
- ❖ Quản lý danh sách sinh viên và giảng viên
- ❖ Thay đổi quy định liên quan
- ❖ Xây dựng hệ thống quản lý đăng ký đề tài
  - Giảng viên đăng đề tài
  - Sinh viên tra cứu đề tài
  - Sinh viên tiến hành đăng kí
  - Khoa tiến hành xét duyệt đăng kí cho sinh viên
- ❖ Xây dựng hệ thống quản lý trong quá trình thực hiện đồ án
  - Cập nhật tiến độ thực hiện
  - Upload tài nguyên dự án

- Chấm điểm
  - ❖ Tra cứu thông tin về các đồ án do các khóa trước thực hiện
  - ❖ Cung cấp giao diện biểu đồ thống kê điểm đồ án theo từng khóa

#### **Về mặt giao diện:**

- ❖ Giao diện trực quan, thân thiện người dùng
- ❖ Cho phép chuyển đổi giao diện

#### **Hiệu suất:**

- Hệ thống hoạt động liên tục 24/7
- Mọi tương tác giữa người dùng và hệ thống không vượt quá 1 giây
- Có khả năng chịu tải 30 TPS
- Có phân quyền người dùng

#### **Pháp luật:**

- Hệ thống phải tuân thủ luật về quyền riêng tư của người dùng
- Hệ thống đảm bảo tuân thủ luật liên quan đến giáo dục theo luật pháp hiện hành ở Việt Nam

### **4. Phân tích kiến trúc hệ thống**

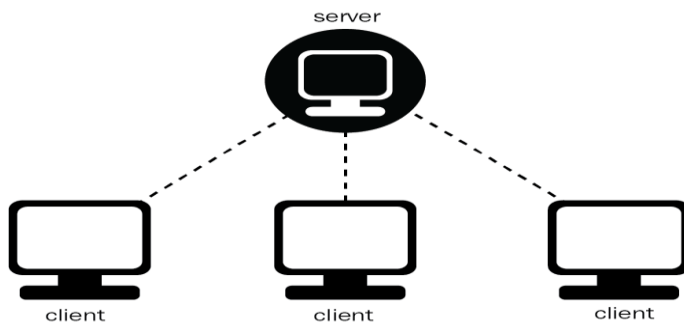
#### **4.1. Khái niệm**

**Mô hình Client Server** là mô hình mạng máy tính trong đó các máy tính con được đóng vai trò như một máy khách, chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ. Để máy chủ xử lý yêu cầu và trả kết quả về cho máy khách đó.

Client và server về bản chất thì nó là 2 máy tính giao tiếp và truyền tải dữ liệu cho nhau.

Máy tính đóng vai trò là máy khách – **Client**: Với vai trò là máy khách, chúng sẽ không cung cấp tài nguyên đến các máy tính khác mà chỉ sử dụng tài nguyên được cung cấp từ máy chủ. Một client trong mô hình này có thể là một server cho mô hình khác, tùy thuộc vào nhu cầu sử dụng của người dùng.

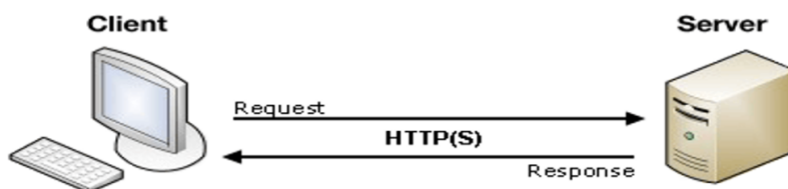
Máy tính đóng vai trò là máy chủ – **Server**: Là máy tính có khả năng cung cấp tài nguyên và các dịch vụ đến các máy khách khác trong hệ thống mạng. Server đóng vai trò hỗ trợ cho các hoạt động trên máy khách client diễn ra hiệu quả hơn.



#### 4.2. Nguyên tắc hoạt động

Trong mô hình Client Server, server chấp nhận tất cả các yêu cầu hợp lệ từ mọi nơi khác nhau trên Internet, sau đó trả kết quả về máy tính đã gửi yêu cầu đó

Máy tính được coi là máy khách khi chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ và đợi câu trả lời được gửi về.



Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định, và chuẩn đó được gọi là giao thức. (Giao thức là gì thì các bạn có thể tham khảo [tại đây](#)) Một số giao thức được sử dụng phổ biến hiện nay như: HTTPS, TCP/IP, FTP,...

Nếu máy khách muốn lấy được thông tin từ máy chủ, chúng phải tuân theo một giao thức mà máy chủ đó đưa ra. Nếu yêu cầu đó được chấp nhận thì máy chủ sẽ thu thập thông tin và trả về kết quả cho máy khách yêu cầu. Bởi vì Server - máy chủ luôn luôn trong trạng thái sẵn sàng để nhận request từ client nên chỉ cần client gửi

yêu cầu tín hiệu và chấp nhận yêu cầu đó thì server sẽ trả kết quả về phía client trong thời gian ngắn nhất.

#### 4.3. Ưu điểm

- Giúp chúng ta có thể làm việc trên bất kì một máy tính nào có hỗ trợ giao thức truyền thông. Giao thức chuẩn này cũng giúp các nhà sản xuất tích hợp lên nhiều sản phẩm khác nhau mà không gặp phải khó khăn gì.
- Có thể có nhiều server cùng làm một dịch vụ, chúng có thể nằm trên nhiều máy tính hoặc một máy tính.
- Chỉ mang đặc điểm của phần mềm mà không hề liên quan đến phần cứng, ngoài yêu cầu duy nhất là server phải có cấu hình cao hơn các client.
- Hỗ trợ người dùng nhiều dịch vụ đa dạng và sự tiện dụng bởi khả năng truy cập từ xa.
- Cung cấp một nền tảng lý tưởng, cho phép cung cấp tích hợp các kỹ thuật hiện đại như mô hình thiết kế hướng đối tượng, hệ chuyên gia, hệ thông tin địa lý (GIS).

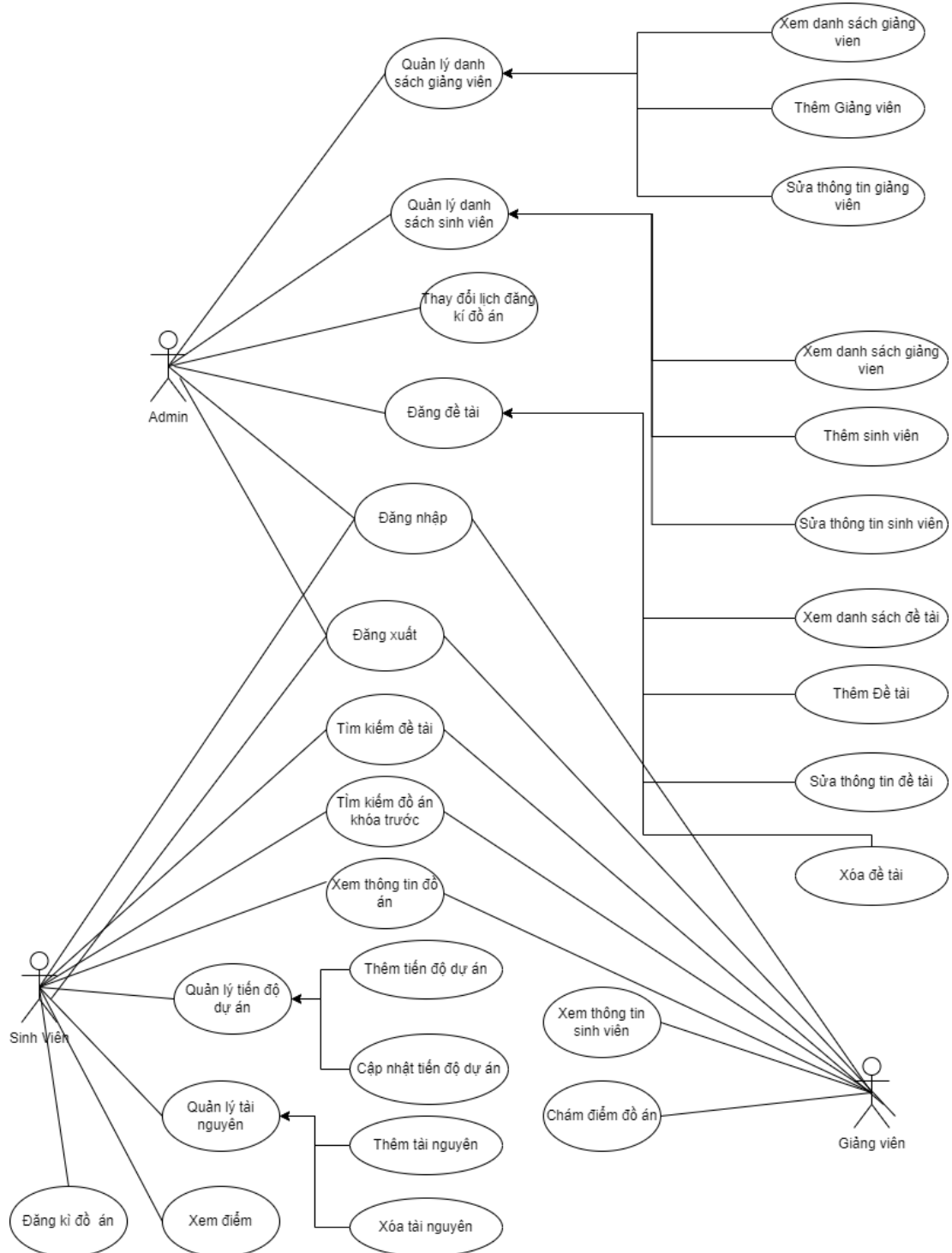
#### 4.4. Kiến trúc hệ thống

STT	Tên	Chức năng
1	Tầng giao diện	Tầng tương tác trực tiếp với người dùng, cho phép người dùng thực hiện các thao tác cũng như hiển thị, thông báo thành công, thất bại của các thao tác
2	Tầng xử lý	Xử lý các yêu cầu của người dùng được gửi từ tầng giao diện và thực hiện thay đổi trong cơ sở dữ liệu
3	Cơ sở dữ liệu	Lưu trữ dữ liệu của website



## 5. Đặc tả use case

### 5.1. Use case toàn hệ thống



## 5.2. Danh mục usecase

STT	Tên usecase	Chức năng
1	Thêm giảng viên	Thêm thông tin giảng viên mới
2	Sửa giảng viên	Sửa thông tin giảng viên
3	Thêm sinh viên	Thêm thông tin sinh viên mới
4	Sửa sinh viên	Sửa thông tin sinh viên
5	Thêm đề tài	Thêm thông tin đề tài mới
6	Sửa đề tài	Sửa thông tin đề tài
7	Xóa đề tài	Xóa thông tin đề tài
8	Đăng nhập	Đăng nhập vào hệ thống
9	Tìm kiếm đề tài	Tìm kiếm các danh sách đề tài
10	Tìm kiếm đồ án khóa trước	Tìm kiếm các danh sách đồ án khóa trước
11	Xem thông tin đồ án	Xem thông tin chi tiết đồ án
12	Thêm tiến độ dự án	Thêm tiến dự án mới
13	Cập nhật tiến độ dự án	Thay đổi thông tin tiến độ
14	Thêm tài nguyên	Thêm tài nguyên mới vào dự án
15	Xóa tài nguyên	Xóa tài nguyên đã có
16	Xem điểm	Xem điểm đồ án củ bản thân

17	Đăng kí đồ án	Đăng kí thực hiện đồ án
18	Xem thông tin sinh viên	Xem chi tiết thông tin sinh viên, đồ án đã thực hiện
19	Chấm điểm đồ án	Chấm điểm cho đồ án sinh viên
20	Đăng xuất	Thoát khỏi hệ thống

### 5.3. Đặc tả

#### 5.3.1. Thêm giảng viên

Tên Use case	Thêm giảng viên
Mô tả	Cho phép actor thay đổi thông tin giảng viên
Actor	Admin
Điều kiện kích hoạt	Nhấn nút thêm
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Thêm thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Actor điền đầy đủ các trường thông tin bắt buộc</li> <li>2. Hệ thống kiểm tra giảng viên đó tồn tại hay chưa và các trường bắt buộc được điền đầy đủ chưa , nếu đủ tiến hành thêm và thông báo thành công</li> <li>3. Kết thúc</li> </ol>
Luồng sự kiện phụ	<p>Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi</p> <p>Nếu giảng viên đó tồn tại thì hệ thống báo giảng đã có</p>

### 5.3.2. Sửa giảng viên

Tên Use case	Chỉnh sửa thông tin giảng viên
Mô tả	Cho phép actor thay đổi thông tin giảng viên
Actor	Admin
Điều kiện kích hoạt	Nhấn nút sửa
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Sửa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị thông tin chi tiết giảng viên</li> <li>2. Actor chỉnh sửa các trường thông tin</li> <li>3. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ hay chưa, nếu đủ tiến hành chỉnh sửa và thông báo thành công</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi

### 5.3.3. Thêm sinh viên

Tên Use case	Thêm sinh viên
Mô tả	Cho phép actor thay đổi thông tin sinh viên
Actor	Admin
Điều kiện kích hoạt	Nhấn nút thêm

Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Thêm thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Actor điền đầy đủ các trường thông tin bắt buộc</li> <li>2. Hệ thống kiểm tra sinh viên đó tồn tại hay chưa và các trường bắt buộc được điền đầy đủ chưa , nếu đủ tiến hành thêm và thông báo thành công</li> <li>3. Kết thúc</li> </ol>
Luồng sự kiện phụ	<p>Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi</p> <p>Nếu sinh viên đó tồn tại thì hệ thống báo giảng đã có</p>

#### 5.3.4. Sửa sinh viên

Tên Use case	Chỉnh sửa thông tin sinh viên
Mô tả	Cho phép actor thay đổi thông tin sinh viên
Actor	Admin
Điều kiện kích hoạt	Nhấn nút sửa
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Sửa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị thông tin chi tiết sinh viên</li> <li>2. Actor chỉnh sửa các trường thông tin</li> <li>3. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ hay chưa, nếu đủ tiến hành chỉnh sửa và thông báo thành công</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi

### 5.3.5. Thêm đề tài

Tên Use case	Thêm đề tài
Mô tả	Cho phép actor thêm đề tài
Actor	Admin
Điều kiện kích hoạt	Nhấn nút thêm
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Thêm thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Actor nhập đầy đủ thông tin cần thiết vào đề tài</li> <li>2. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ không</li> <li>3. Nếu có, hệ thống thông báo thêm thành công</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi

### 5.3.6. Sửa đề tài

Tên Use case	Chỉnh sửa thông tin đề tài
Mô tả	Cho phép actor thay đổi thông tin đề tài
Actor	Admin
Điều kiện kích hoạt	Nhấn nút sửa
Điều kiện trước	Đã đăng nhập vào hệ thống thành công

Điều kiện sau	Sửa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị thông tin chi tiết đề tài</li> <li>2. Actor chỉnh sửa các trường thông tin</li> <li>3. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ hay chưa, nếu đủ tiến hành chỉnh sửa và thông báo thành công</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi

### 5.3.7. Xóa đề tài

Tên Use case	Xóa đề tài
Mô tả	Cho phép actor xóa đề tài
Actor	Admin
Điều kiện kích hoạt	Nhấn nút xóa
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Xóa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị thông tin chi tiết giảng viên</li> <li>2. Actor chọn giảng viên muốn xóa</li> <li>3. Hệ thống hiển thị thông báo kiểm tra chắc chắn xóa, nhấn đồng ý sẽ tiến hành xóa</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Hệ thống hiển thị thông báo kiểm tra chắc chắn xóa, nhấn không sẽ không thực hiện thao tác

### 5.3.8. Đăng nhập

Tên Use case	Đăng nhập
Mô tả	Cho phép actor đăng nhập vào hệ thống
Actor	Admin, sinh viên, giảng viên
Điều kiện kích hoạt	Chọn nút login
Điều kiện trước	Tài khoản tồn tại
Điều kiện sau	Đăng nhập vào hệ thống thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình đăng nhập</li> <li>2. Actor đăng nhập qua gmail</li> <li>3. Hệ thống kiểm tra thông tin</li> <li>4. Nếu hợp lệ, hệ thống sẽ đưa actor đến trang Admin ( tài khoản admin) hoặc Home (tài khoản sinh viên/ giảng viên)</li> <li>5. Kết thúc</li> </ol>
Luồng sự kiện phụ	<ol style="list-style-type: none"> <li>1. Nếu tài khoản không tồn tại: hệ thống thông báo tài khoản không tồn tại</li> <li>2. Nếu mật khẩu sai: hệ thống thông báo mật khẩu sai</li> </ol>

### 5.3.9. Tìm kiếm đề tài

Tên Use case	Tìm kiếm đề tài
Mô tả	Cho phép actor tìm kiếm đề tài
Actor	Sinh viên



Điều kiện kích hoạt	Nhấn nút tìm
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Actor điền vào các trường của bộ công cụ search</li> <li>2. Hệ thống hiển thị danh sách đề tài phù hợp yêu cầu</li> <li>3. Kết thúc</li> </ol>
Luồng sự kiện phụ	

#### 5.3.10. Tìm kiếm đề án khóa trước

Tên Use case	Tìm kiếm đề án
Mô tả	Cho phép actor tìm kiếm đề án
Actor	Giảng viên, Sinh viên
Điều kiện kích hoạt	Nhấn nút tìm
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Actor điền vào các trường của bộ công cụ search</li> <li>2. Hệ thống hiển thị danh sách đề án phù hợp yêu cầu</li> <li>3. Kết thúc</li> </ol>
Luồng sự kiện phụ	

#### 5.3.11. Xem thông tin đề án

Tên Use case	Xem thông tin đề án
--------------	---------------------

Mô tả	Cho phép actor xem thông tin chi tiết đồ án
Actor	Tất cả
Điều kiện kích hoạt	Chọn đồ án muốn xem trên danh sách
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	
Luồng sự kiện chính	1. Hệ thống hiển thị màn hình chi tiết đồ án 2. Kết thúc
Luồng sự kiện phụ	

### 5.3.12. Thêm tiến độ dự án

Tên Use case	Thêm tiến độ
Mô tả	Cho phép actor thêm tiến độ đồ án mình quản lý
Actor	Sinh viên
Điều kiện kích hoạt	Chọn nút thêm
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	
Luồng sự kiện chính	1. Actor nhập đầy đủ thông tin cần thiết vào tiến độ 2. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ không 3. Nếu có, hệ thống thông báo thêm thành công 4. Kết thúc

Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi
-------------------	--

### 5.3.13. Cập nhật tiến độ dự án

Tên Use case	Cập nhật tiến độ dự án
Mô tả	Cho phép actor thay đổi thông tin tiến độ
Actor	Sinh viên
Điều kiện kích hoạt	Nhấn nút sửa
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Sửa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị thông tin chi tiết tiến độ</li> <li>2. Actor chỉnh sửa các trường thông tin</li> <li>3. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ hay chưa, nếu đủ tiến hành chỉnh sửa và thông báo thành công</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi

### 5.3.14. Thêm tài nguyên

Tên Use case	Thêm tài nguyên
Mô tả	Cho phép actor thêm tài nguyên vào đồ án mình quản lý
Actor	Sinh viên

Điều kiện kích hoạt	Chọn nút thêm tài nguyên
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Actor nhập đầy đủ thông tin cần thiết vào popup tài nguyên</li> <li>2. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ không</li> <li>3. Nếu có, hệ thống thông báo thêm thành công</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi

#### 5.3.15. Xóa tài nguyên

Tên Use case	Xóa tài nguyên
Mô tả	Cho phép actor xóa tài nguyên
Actor	Sinh viên
Điều kiện kích hoạt	Nhấn nút xóa
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Xóa thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị thông tin chi tiết giảng viên</li> <li>2. Actor chọn giảng viên muốn xóa</li> <li>3. Hệ thống hiển thị thông báo kiểm tra chắc chắn xóa, nhấn đồng ý sẽ tiến hành xóa</li> <li>4. Kết thúc</li> </ol>

Luồng sự kiện phụ	Hệ thống hiển thị thông báo kiểm tra chắc chắn xóa, nhấn không sẽ không thực hiện thao tác
-------------------	--

### 5.3.16. Xem điểm

Tên Use case	Xem điểm
Mô tả	Cho phép actor xem điểm thống kê
Actor	Admin
Điều kiện kích hoạt	Không có
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình xem điểm</li> <li>2. Actor sử dụng filter để lọc kết quả</li> <li>3. Hệ thống hiển thị thống kê theo điều kiện lọc</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	

### 5.3.17. Đăng kí đồ án

Tên Use case	Đăng kí đồ án
Mô tả	Cho phép actor đăng kí đồ án
Actor	Sinh viên
Điều kiện kích hoạt	Nhấn nút đăng kí

Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Đăng kí thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị danh sách đề tài</li> <li>2. Actor chọn mục muốn đăng kí</li> <li>3. Hệ thống kiểm tra actor có đang trong hàng đợi đăng kí hay không</li> <li>4. Nếu không, hệ thống thông báo đăng kí thành công</li> <li>5. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu đang trong hàng đợi sẽ không tiến hành đăng kí

#### 5.3.18. Xem thông tin sinh viên

Tên Use case	Xem thông tin sinh viên
Mô tả	Cho phép actor xem thông tin chi tiết sinh viên
Actor	Admin, Giảng viên
Điều kiện kích hoạt	Chọn sinh viên muốn xem trên danh sách
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị màn hình chi tiết sinh viên</li> <li>2. Kết thúc</li> </ol>
Luồng sự kiện phụ	

### 5.3.19. Chấm điểm đồ án

Tên Use case	Chấm điểm đồ án
Mô tả	Cho phép actor thay đổi thông tin điểm số đồ án
Actor	Giảng viên
Điều kiện kích hoạt	Nhấn nút cho điểm
Điều kiện trước	Đã đăng nhập vào hệ thống thành công
Điều kiện sau	Thêm điểm thành công
Luồng sự kiện chính	<ol style="list-style-type: none"> <li>1. Hệ thống hiển thị bảng cho điểm</li> <li>2. Actor điền vào các trường thông tin</li> <li>3. Hệ thống kiểm tra các trường bắt buộc được điền đầy đủ hay chưa, nếu đủ tiến hành thêm và thông báo thành công</li> <li>4. Kết thúc</li> </ol>
Luồng sự kiện phụ	Nếu điền không đầy đủ các trường bắt buộc chỉnh sửa sẽ báo lỗi

### 5.3.20. Đăng xuất

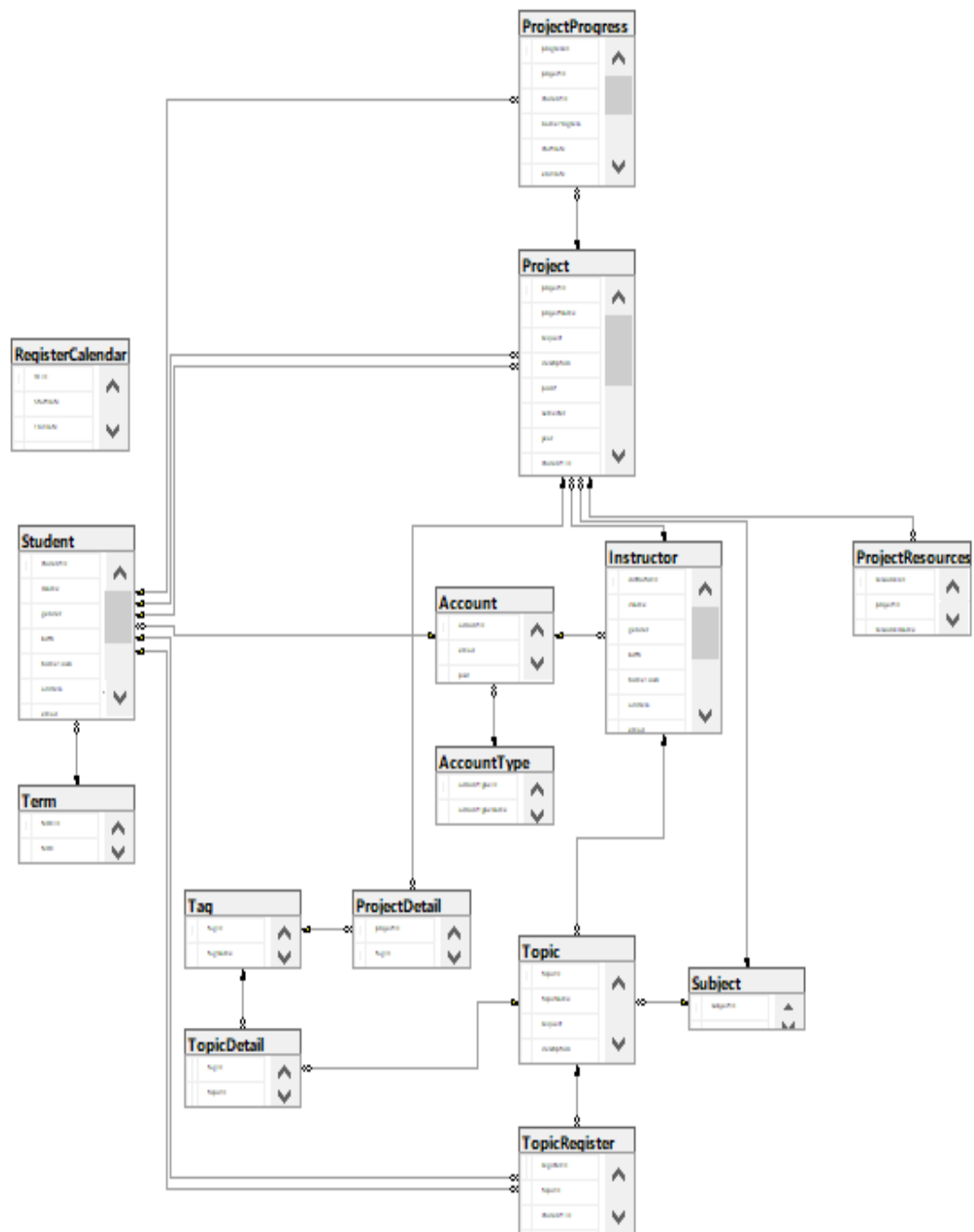
Tên Use case	Đăng xuất
Mô tả	Cho phép actor đăng xuất khỏi hệ thống
Actor	Admin, Sinh viên, giảng viên
Điều kiện kích hoạt	Chọn nút đăng xuất

Điều kiện trước	Đã đăng nhập
Điều kiện sau	Đăng xuất thành công
Luồng sự kiện chính	1. Hệ thống đưa actor về lại màn hình đăng nhập 2. Kết thúc
Luồng sự kiện phụ	



## 6. Thiết kế cơ sở dữ liệu

### 6.1. Sơ đồ cơ sở dữ liệu



## 6.2. Mô tả chi tiết

STT	Tên	Mục đích
1	Account	Lưu trữ thông tin tài khoản để xác thực cho việc đăng nhập
2	AccountType	Lưu trữ thông tin loại tài khoản
3	Student	Lưu trữ thông tin sinh viên
4	Term	Lưu trữ thông tin khóa
5	Instructor	Lưu thông tin giảng viên
6	Topic	Lưu danh sách đề tài
7	Tag	Lưu trữ thẻ cho việc tìm kiếm
8	TopicDetail	Lưu trữ thông tin chi tiết đề tài
9	TopicRegister	Lưu trữ danh sách đăng kí đồ án
10	Project	Lưu trữ thông tin đồ án
11	ProjectDetail	Lưu trữ thông tin chi tiết đồ án
12	ProjectResources	Lưu trữ tài nguyên đồ án

13	Subject	Lưu trữ loại đồ án (1/2)
14	ProjectProgress	Lưu trữ thông tin tiến độ dự án
15	RegisterCalendar	Thời gian mở/ đóng đăng kí đồ án
16	CurrentSubject	Đồ án đang làm hiện tại

### 6.3. Danh mục bảng

#### 6.3.1. Bảng Account

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	accountID	int	khóa chính	mã tài khoản
2	email	nvarchar(150)		email
3	pwd	nvarchar(150)		mật khẩu
4	accountTypeID	int	khóa ngoại	mã loại tài khoản

#### 6.3.2. Bảng AccountType

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	accountTypeID	int	khóa chính	mã loại tài khoản

2	accountTypeNam	nvarchar(150)	tên loại tài khoản
3	permission	nvarchar(150)	quyền hạn

### 6.3.3. Bảng Student

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	studentID	int	khóa chính	Mã số sinh viên
2	sName	nvarchar(150)		tên sinh viên
3	gender	nvarchar(4)		giới tính
4	birth	datetime		
5	homeTown	nvarchar(150)		quê quán
6	address	nvarchar(150)		địa chỉ
7	email	nvarchar(150)		email
8	phoneNumber	int		
9	termID	int		khóa ngoại

<b>10</b>	accountID	int	khóa ngoại
-----------	-----------	-----	------------

#### 6.3.4. Bảng Term

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
<b>1</b>	termID	int	khóa chính	mã khóa
<b>2</b>	term	int		khóa
<b>3</b>	Note	nvarchar(150)		ghi chú

#### 6.3.5. Bảng Instructor

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
<b>1</b>	instructorID	int	khóa chính	mã giảng viên
<b>2</b>	iName	nvarchar(150)		tên giảng viên
<b>3</b>	gender	nvarchar(150)		giới tính
<b>4</b>	birth	datetime		ngày sinh
<b>5</b>	homeTown	nvarchar(150)		quê quán

<b>6</b>	address	nvarchar(150)		địa chỉ
<b>7</b>	email	nvarchar(150)		email
<b>8</b>	phoneNumber	int		sdt
<b>9</b>	degree	nvarchar(150)		học vị
<b>10</b>	accountID	int	khóa ngoại	mã tài khoản

#### 6.3.6. Bảng Topic

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
<b>1</b>	topicID	int	khóa chính	Mã đề tài
<b>2</b>	topicName	nvarchar(150)		tên đề tài
<b>3</b>	request	nvarchar(150)		yêu cầu
<b>4</b>	description	nvarchar(150)		mô tả
<b>5</b>	instructorID	int	khóa ngoại	mã giảng viên
<b>6</b>	subjectID	int	khóa ngoại	loại đồ án

#### 6.3.7. Bảng Tag

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	tagID	int	khóa chính	mã thẻ
2	tagName	nvarchar(150)		tên thẻ
3	description	nvarchar(150)		mô tả

#### 6.3.8. Bảng TopicDetail

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	tagID	int	khóa chính	mã thẻ
2	topicID	int	khóa chính	Mã đề tài
3	Note	nvarchar(150)		ghi chú

#### 6.3.9. Bảng TopicRegister

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	registerID	int	khóa chính	mã đăng kí
2	topicID	int	khóa ngoại	Mã đề tài
3	student1ID	int	khóa ngoại	Mssv 1
4	student2ID	int	khóa ngoại	Mssv 2
5	status	nvarchar(150)		trạng thái

### 6.3.10. Bảng Project

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	projectID	int	khóa chính	mã đồ án
2	projectName	nvarchar(150)		tên đồ án
3	request	nvarchar(150)		yêu cầu
4	description	nvarchar(150)		mô tả
5	point	decimal		điểm
6	semester	int		kì
7	year	int		năm
8	student1ID	int	khóa ngoại	Mssv 1
9	student2ID	int	khóa ngoại	Mssv 2
10	status	nvarchar(150)		trạng thái
11	subjectID	int	khóa ngoại	loại đồ án
12	instructorID	int	khóa ngoại	Mã giảng viên

### 6.3.11. Bảng ProjectDetail

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	projectID	int	khóa chính	mã đồ án



2	tagID	int	khóa chính	mã thẻ
3	Note	nvarchar(150)		ghi chú

#### 6.3.12. Bảng ProjectResources

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	resourcesID	int	khóa chính	mã tài nguyên
2	projectID	int	khóa ngoại	mã đồ án
3	resourcesName	nvarchar(150)		tên tài nguyên
4	filePath	nvarchar(150)		đường dẫn

#### 6.3.13. Bảng Subject

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	subjectID	int	khóa chính	loại đồ án
2	subjectName	nvarchar(150)		tên loại đồ án

#### 6.3.14. Bảng ProjectProgress

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	progressID	int	khóa chính	mã tiến độ
2	projectID	int	khóa ngoại	mã đồ án

3	studentID	int	khóa ngoại	mssv
4	progressName	nvarchar(200)		tên tiến độ
5	startDate	datetime		ngày bắt đầu
6	endDate	datetime		ngày kết thúc
7	Status	nvarchar(150)		trạng thái

#### 6.3.15. Bảng RegisterCalenDar

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	RCID	int	khóa chính	mã thời gian đăng kí
2	StartDate	datetime		ngày bắt đầu
3	EndDate	datetime		ngày kết thúc
4	semester	int		kì
5	Ryear	int		năm

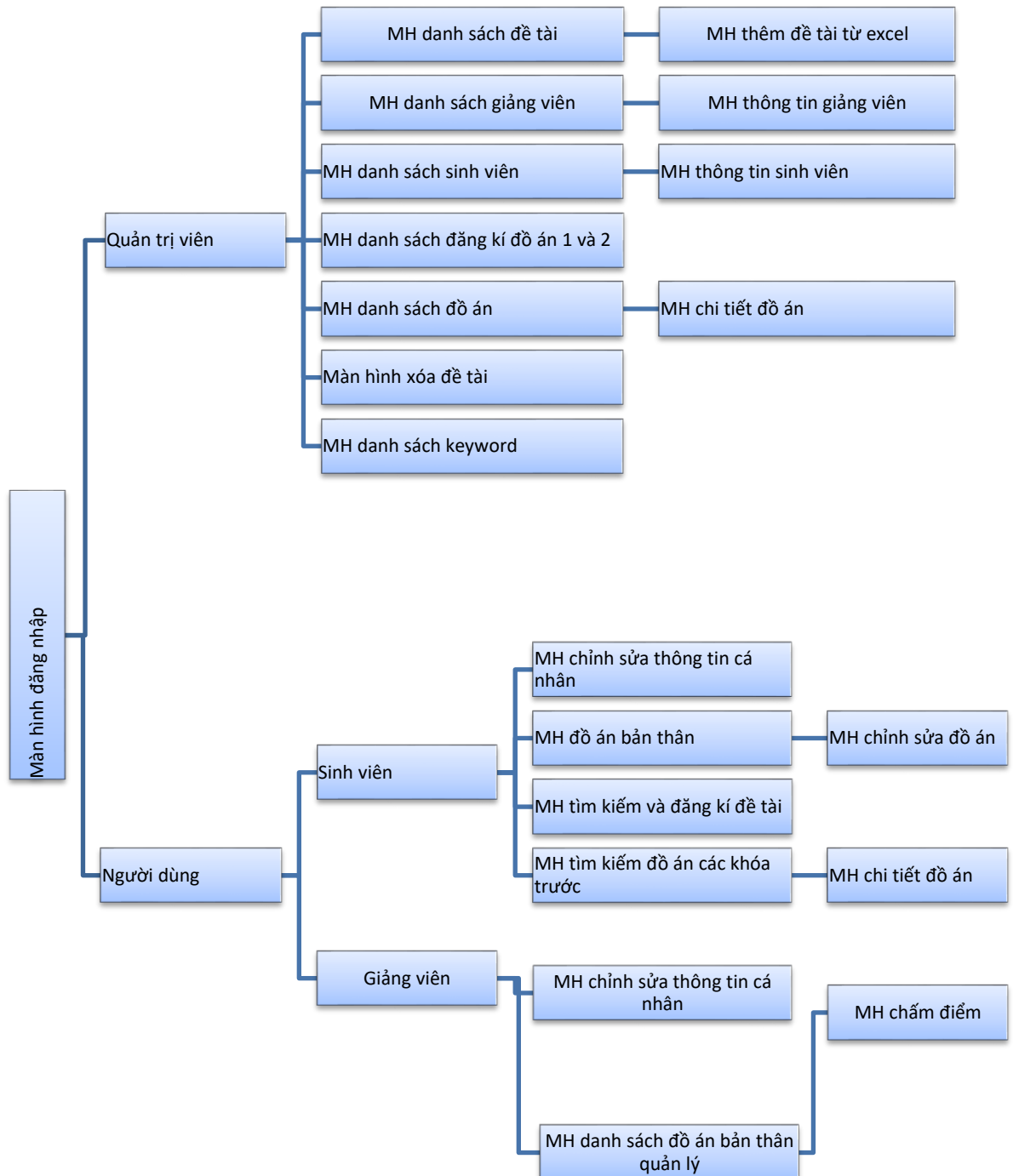
#### 6.3.16. Bảng CurrentSubject

STT	Tên thuộc tính	Kiểu dữ liệu	Ràng buộc	Mô tả
1	cSubjectID	int	khóa chính	mã đồ án hiện tại

<b>2</b>	studentID	int	khóa ngoại	mssv
<b>3</b>	subjectID	int	khóa ngoại	loại đồ án

## 7. Thiết kế giao diện

### 7.1. Sơ đồ liên kết giao diện



## 7.2. Danh sách giao diện

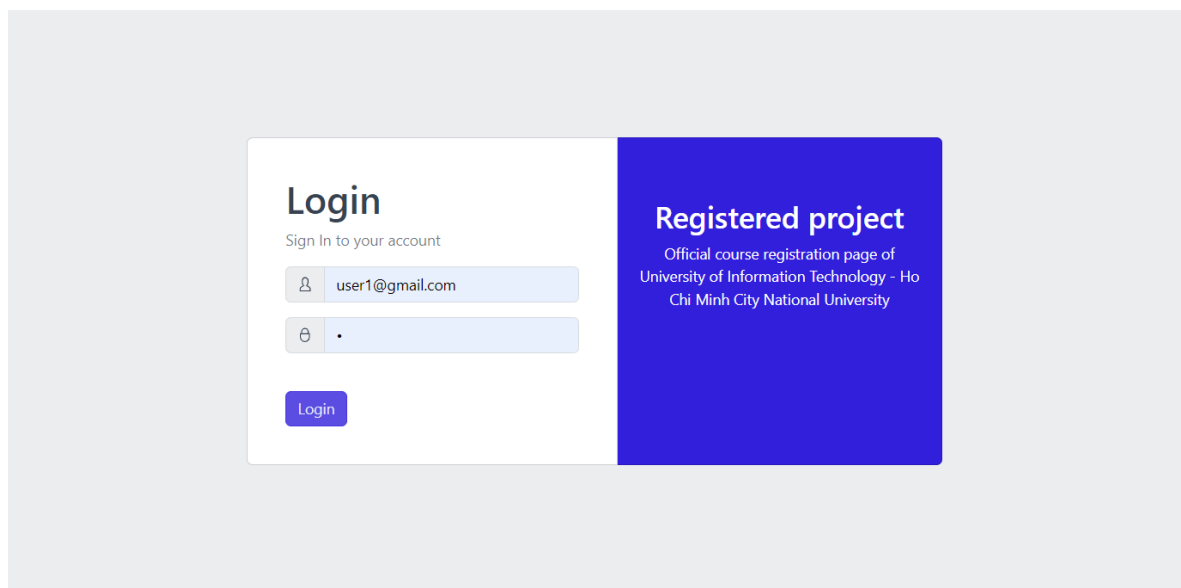
STT	Màn hình	Actor	Chức năng
1	Màn hình đăng nhập	Tất cả	Đăng nhập
2	Màn hình danh sách đề tài	Admin	Hiện thị danh sách các đề tài Tìm kiếm đề tài Upload đề tài Chỉnh sửa đề tài
3	Màn hình thêm đề tài từ excel	Admin	Thêm đề tài mới từ file excel
4	Màn hình danh sách giảng viên	Admin	Hiện thị danh sách các đề tài Tìm kiếm đề tài Thêm mới giảng viên
5	Màn hình thông tin giảng viên	Admin	Hiện thị thông tin chi tiết của giảng viên đó Chỉnh sửa thông tin cơ bản
6	Màn hình danh sách sinh viên	Admin	Hiện thị danh sách các đề tài Tìm kiếm đề tài Thêm mới sinh viên
7	Màn hình thông tin sinh viên	Admin	Hiện thị thông tin chi tiết của sinh viên đó

			Chỉnh sửa thông tin cơ bản
<b>8</b>	Màn hình danh sách đăng kí đồ án 1 và 2	Admin	Hiển thị danh sách đăng kí đồ án 1 và 2 của sinh viên  Duyệt đăng kí
<b>9</b>	Màn hình cài đặt	Admin	Cài đặt chỉ số cho trang web
<b>10</b>	Màn hình chi tiết đồ án	Admin	Hiển thị chi tiết tiến độ quá trình của đồ án
<b>11</b>	Màn hình chấm điểm	Giảng viên	Cho điểm đồ án thuộc quyền quản lý
<b>12</b>	Màn hình danh sách keyword	Admin	Danh sách keyword để gắn vào đề tài/đồ án  Thêm, xóa, sửa keyword
<b>13</b>	Màn hình chỉnh sửa thông tin cá nhân sinh viên	Sinh viên	Xem, thay đổi thông tin được phép
<b>14</b>	Màn hình chỉnh sửa đồ án bản thân	Sinh viên	Cập nhật tiến độ đồ án  Upload file tài nguyên dự án
<b>15</b>	Màn hình thêm tiến độ	Sinh viên	Thêm tiến độ vào đồ án
<b>16</b>	Màn hình tìm kiếm đề tài và đăng kí	Sinh viên	Hiện thị danh sách đề tài hiện mở
<b>17</b>	Màn hình tìm kiếm đồ án các khóa trước	Sinh viên	Hiện thị danh sách đồ án khóa trước

18	Màn hình chi tiết đồ án người dùng	Sinh viên, Giảng viên	Hiện thị thông tin chi tiết đồ án
19	Màn hình chỉnh sửa thông tin cá nhân	Giảng viên	Xem, thay đổi thông tin được phép
20	Màn hình danh sách đồ án bản thân quản lý	Giảng viên	Hiện thị danh sách đồ án quản lý theo kì

### 7.3. Mô tả chi tiết

#### 7.3.1. Màn hình đăng nhập



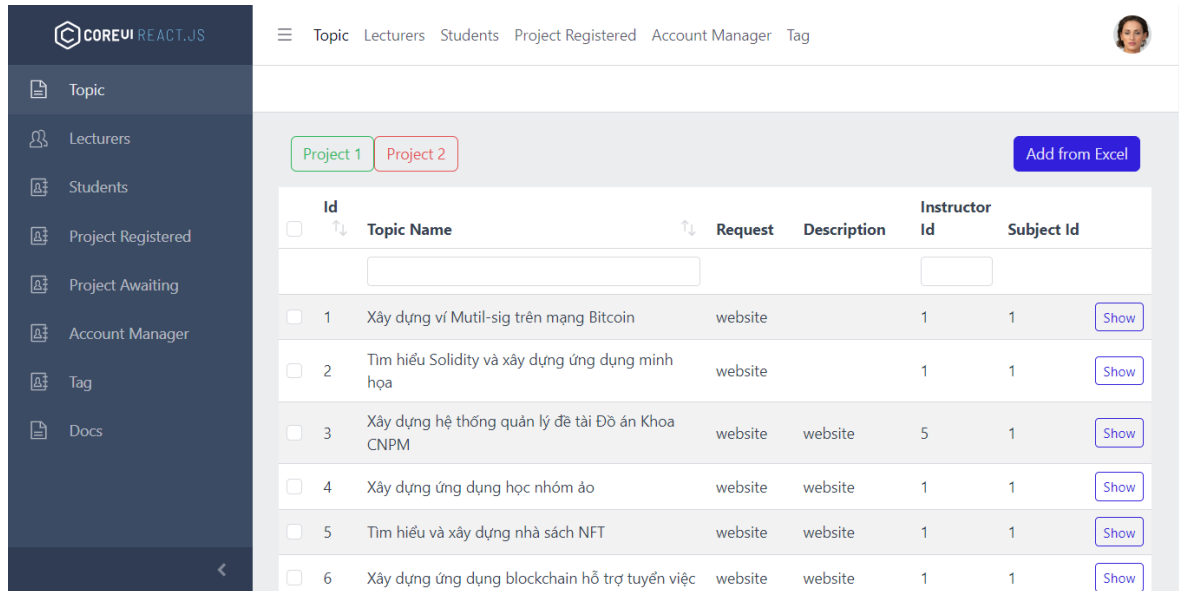
Hình 7.1: Màn hình đăng nhập

STT	Tên phần tử	Chức năng
1	userInput	Nhập tên tài khoản
2	pwInput	Nhập mật khẩu

3 loginBtn

Đăng nhập

### 7.3.2. Màn hình danh sách đề tài



Hình 7.2: Màn hình danh sách quản lý đề tài

STT	Tên phần tử	Chức năng
1	topicNameInput	Tìm kiếm đề tài theo tên
2	instructorInput	Tìm kiếm đề tài theo tên giảng viên
3	project1Btn	Lọc đề án 1
4	project2Btn	Lọc đề án 2
5	addExcelBtn	Thêm đề tài từ file excel
6	topicTbl	Hiển thị danh sách đề tài



7.3.3. Màn hình thêm đề tài từ excel

List of topics

Chọn tệp

Không có tệp nào được chọn

<input checked="" type="checkbox"/>	Id	Topic Name		Request	Description	Instructor Id	Subject Id
		<input type="text"/>				<input type="text"/>	

No items found

Items per page: 10

Save

Hình 7.3: Màn hình thêm đề tài excel

STT	Tên phần tử	Chức năng
1	fileInput	Lựa chọn file
2	topicNameInput	Tìm kiếm đề tài theo tên
3	instructorInput	Tìm kiếm đề tài theo tên giảng viên
4	topicTbl	Hiển thị danh sách đề tài
5	saveBtn	Thêm vào bảng đề tài

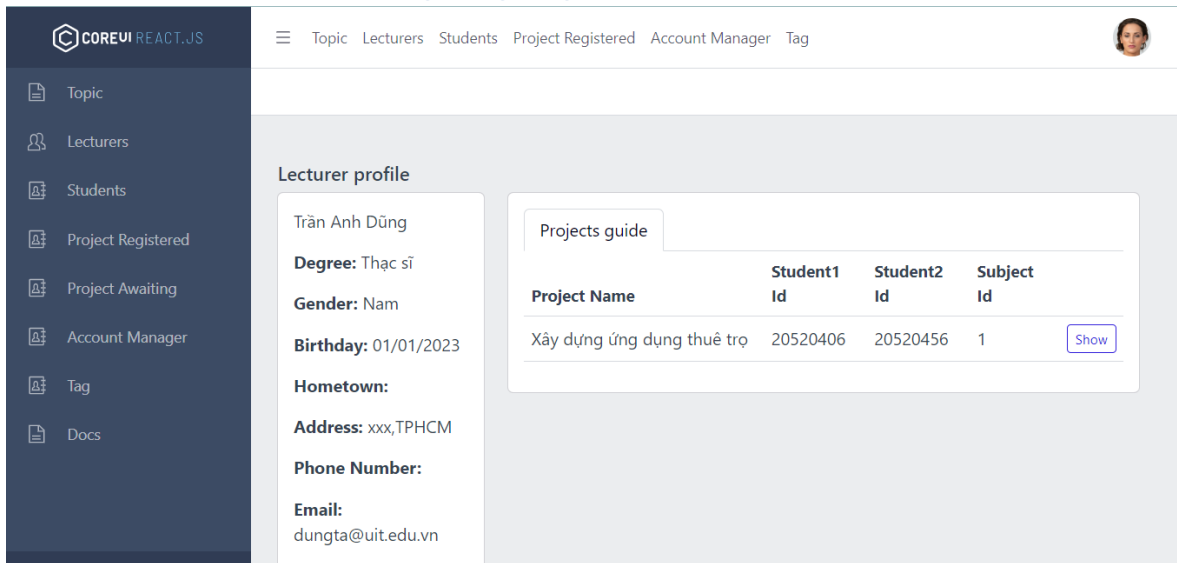
### 7.3.4. Màn hình danh sách giảng viên

	Name	Gender	Email	Phone Number	Degree	Account Id	
1	Trần Anh Dũng	Nam	dungta@uit.edu.vn	0	Thạc sĩ	2	<a href="#">Detail</a>
2	Trần Anh Dũng	Nam	dungta@uit.edu.vn	0	Thạc sĩ	3	<a href="#">Detail</a>
3	Huỳnh Hồ Thị Mộng Trinh	Nữ	trinhhhtm@uit.edu.vn	0	Thạc sĩ	4	<a href="#">Detail</a>
4	Nguyễn Công Hoan	Nam	hoannc@uit.edu.vn	0	Thạc sĩ	5	<a href="#">Detail</a>
5	Nguyễn Thị Thanh Trúc	Nữ	trucntt@uit.edu.vn	0	Thạc sĩ	6	<a href="#">Detail</a>

Hình 7.4: Màn hình danh sách giảng viên

STT	Tên phần tử	Chức năng
1	excelBtn	Thêm mới giảng viên vào từ file excel
2	nameSearchBar	Lọc danh sách theo tên
3	homeTownSearchBar	Lọc danh sách theo địa chỉ
4	degreeSearchBar	Lọc danh sách theo học vị
5	instructorTable	Hiển thị danh sách giảng viên
6	detailBtn	Chuyển đến trang thông tin chi tiết giảng viên

### 7.3.5. Màn hình thông tin giảng viên



Hình 7.5: Màn hình thông tin chi tiết giảng viên

STT	Tên phần tử	Chức năng
1	informationCard	Hiển thị thông tin cơ bản của giảng viên
2	projectCard	Hiển thị thông tin các đề án do giảng viên hướng dẫn
3	showBtn	Chuyển đến trang thông tin chi tiết đề án

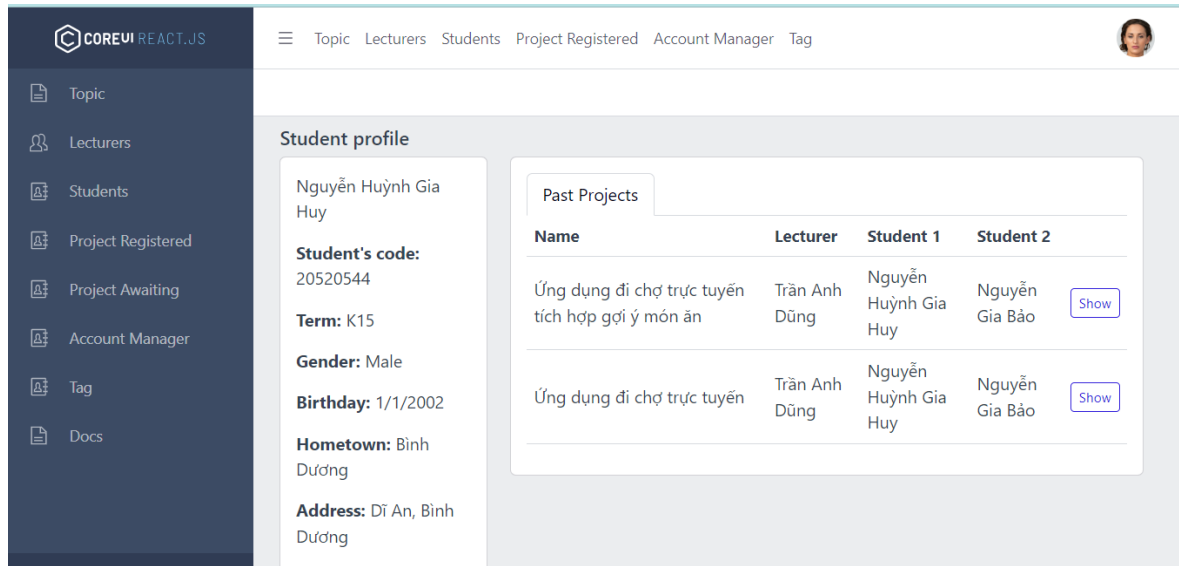
### 7.3.6. Màn hình danh sách sinh viên

	Name	Gender	Home Town	Address	Email	Phone Number	Term Id	Account Id	
20520406	Nguyễn Gia Bảo	Nam	TPHCM	xxx,TPHCM	20520406@gm.uit.edu.vn	0	6	15	Show
20520456	Đặng Đình Dũng	Nam	TPHCM	xxx,TPHCM	20520456@gm.uit.edu.vn	0	6	16	Show
20520544	Nguyễn Huỳnh Gia Huy	Nam	TPHCM	xxx,TPHCM	20520544@gm.uit.edu.vn	0	6	14	Show

Hình 7.6: Màn hình danh sách sinh viên

STT	Tên phần tử	Chức năng
1	excelBtn	Thêm mới sinh viên vào từ file excel
2	nameSearchBar	Lọc danh sách theo tên
3	homeTownSearchBar	Lọc danh sách theo địa chỉ
4	termSearchBar	Lọc danh sách theo khóa
5	studentTable	Hiển thị danh sách sinh viên
6	detailBtn	Chuyển đến trang thông tin chi tiết sinh viên

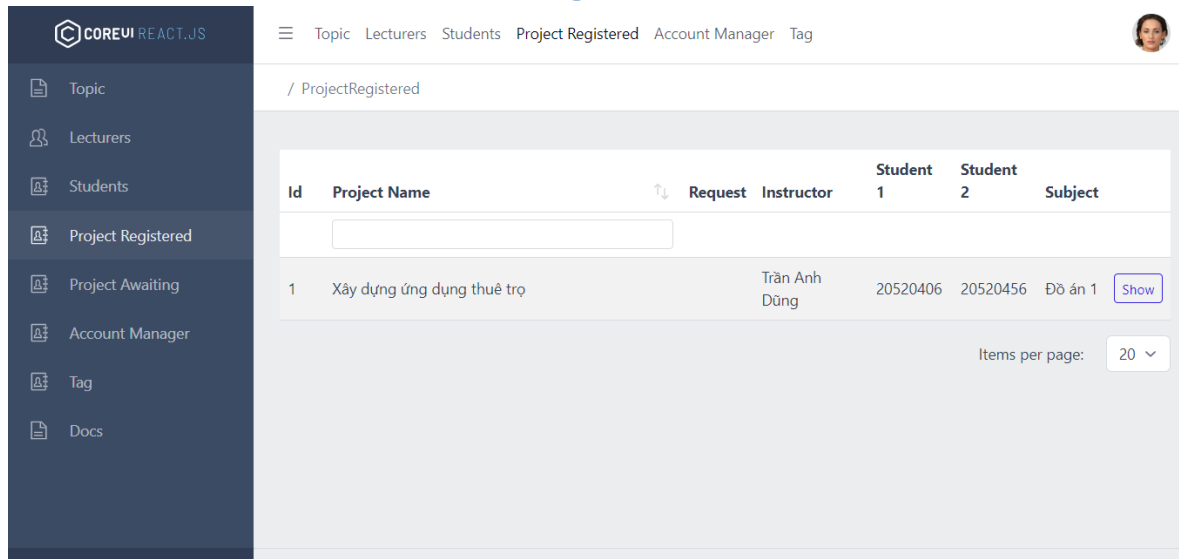
### 7.3.7. Màn hình thông tin sinh viên



Hình 7.8: Màn hình chi tiết thông tin sinh viên

STT	Tên phần tử	Chức năng
1	informationCard	Hiển thị thông tin cơ bản của sinh viên
2	projectCard	Hiển thị thông tin các đề án do sinh viên thực hiện
3	showBtn	Chuyển đến trang thông tin chi tiết đề án

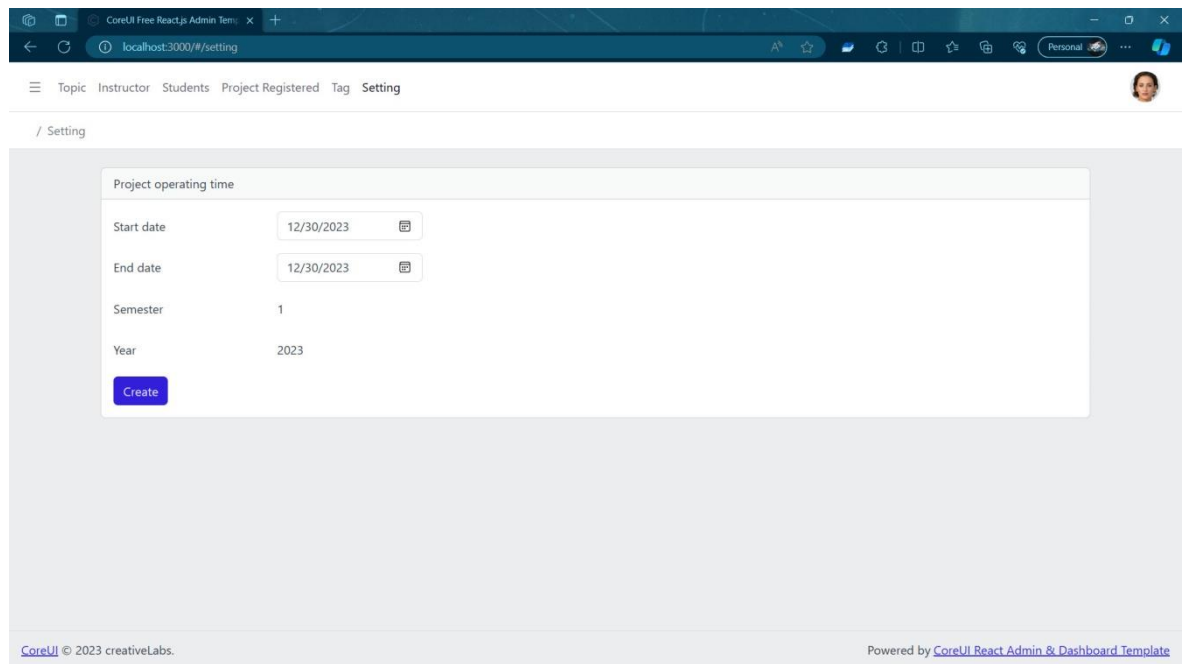
### 7.3.8. Màn hình danh sách đăng kí đồ án 1 và 2



Hình 7.8: Màn hình danh sách đăng kí đồ án

STT	Tên phần tử	Chức năng
1	projectNameInput	Tìm kiếm đồ án theo tên
2	showBtn	Chuyển đến trang thông tin chi tiết đồ án
3	projectTbl	Hiển thị danh sách đề tài

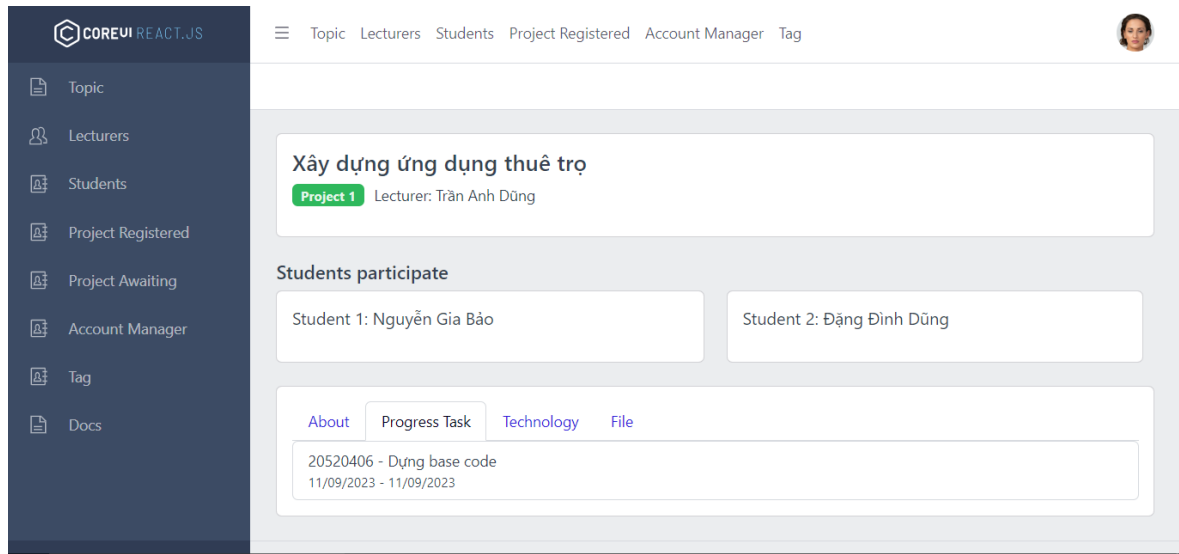
### 7.3.9. Màn hình cài đặt



Hình 7.9: Màn hình cài đặt

STT	Tên phần tử	Chức năng
1	startDatePicker	Chọn ngày bắt đầu
2	endDatePicker	Chọn ngày kết thúc
3	createBtn	Xác nhận tạo ngày đăng kí đồ án

### 7.3.10. Màn hình chi tiết đồ án

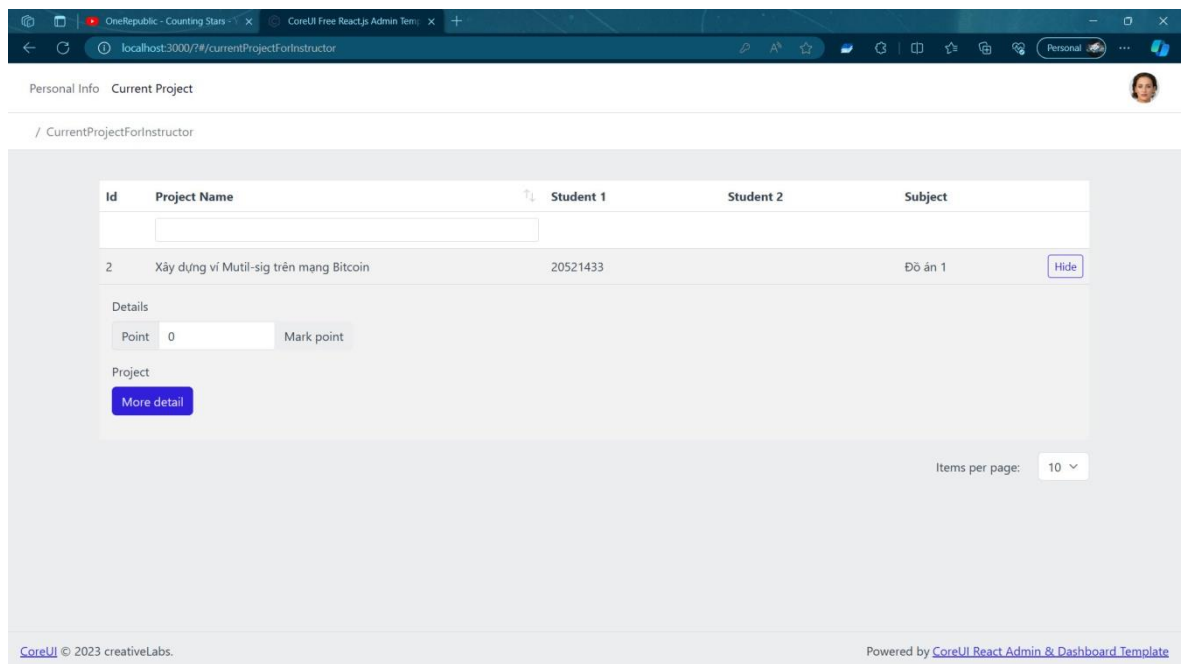


Hình 7.10: Màn hình chi tiết đồ án

STT	Tên phần tử	Chức năng
1	informationCard	Thông tin cơ bản dự án
2	studentCard	Tên sinh viên thực hiện
3	aboutTab	Mô tả đồ án
4	progressTab	Tiến độ dự án
5	techTab	Mã công nghệ đồ án
6	fileTab	Tài nguyên đồ án

### 7.3.11. Màn hình chấm điểm

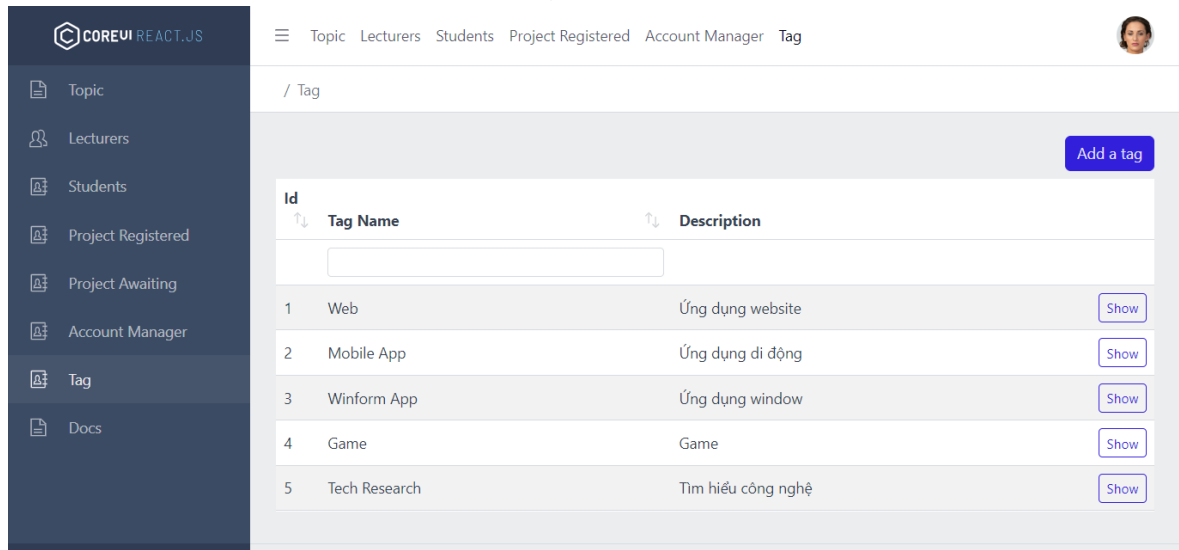




Hình 7.11: Màn hình chấm điểm

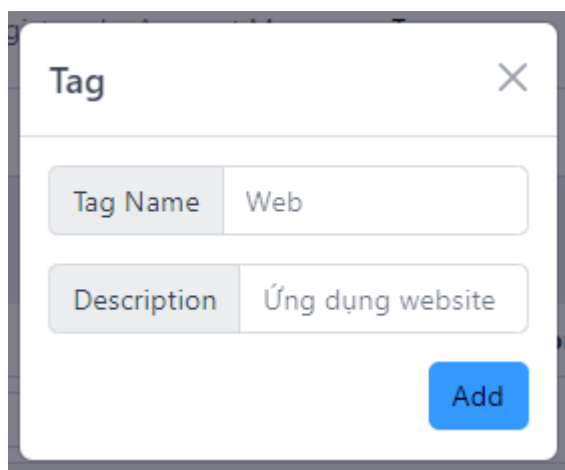
STT	Tên phần tử	Chức năng
1	projectSearchBar	Tìm kiếm theo tên
2	scoreInput	Nhập điểm
3	moreBtn	Chuyển đến trang chi tiết đồ án
4	projectTbl	Hiển thị danh sách đồ án

### 7.3.12. Màn hình danh sách keyword



Hình 7.12a: Màn hình danh sách keyword

STT	Tên phần tử	Chức năng
1	addBtn	Thêm mới thẻ
2	tagNameSearchBar	Lọc danh sách theo tên
3	tagTbl	Hiển thị danh sách thẻ
4	showBtn	Hiển thị chi tiết thẻ



Hình 7.12b: Màn hình thêm thẻ

STT	Tên phần tử	Chức năng
1	tagNameInput	Nhập tên thẻ
2	descriptionInput	Nhập mô tả
3	addBtn	Thêm mới tài khoản

### 7.3.13. Màn hình chỉnh sửa thông tin cá nhân sinh viên

Hình 7.13: Màn hình chỉnh sửa thông tin sinh viên

STT	Tên phần tử	Chức năng
1	infomationCard	Hiển thị thông tin cơ bản
2	saveBtn	Lưu thông tin chỉnh sửa
3	projectCard	Hiển thị thông tin danh sách đồ án bản thân

### 7.3.14. Màn hình chỉnh sửa đồ án

ProjectPersonal InfoCurrent ProjectProject List

/ CurrentProject

Xây dựng ứng dụng thuê trọ

Project 1

Lecturer: Trần Anh Dũng

Students participate

Student 1: Nguyễn Gia Bảo

Student 2: Đặng Đình Dũng

DescriptionProgress TaskTechnologyFileQuestion

20520406 - Dự án base code  
11/09/2023 - 11/09/2023

Add progress

Hình 7.14a: Màn hình chỉnh sửa đồ án

DescriptionProgress TaskTechnologyFileQuestion

Save

Hình 7.14b: Tab mô tả

DescriptionProgress TaskTechnologyFileQuestion

20520406 - Dự án base code  
11/09/2023 - 11/09/2023

Add progress

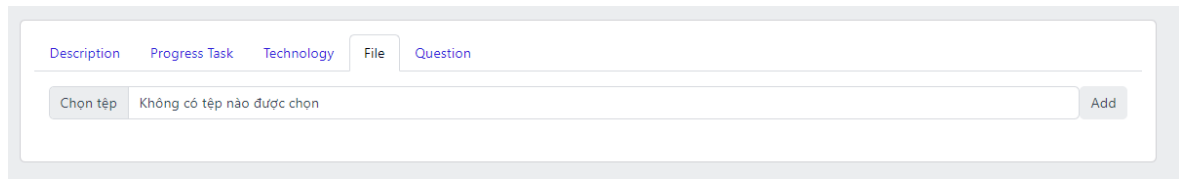
Hình 7.14c: Tab tiến độ

DescriptionProgress TaskTechnologyFileQuestion

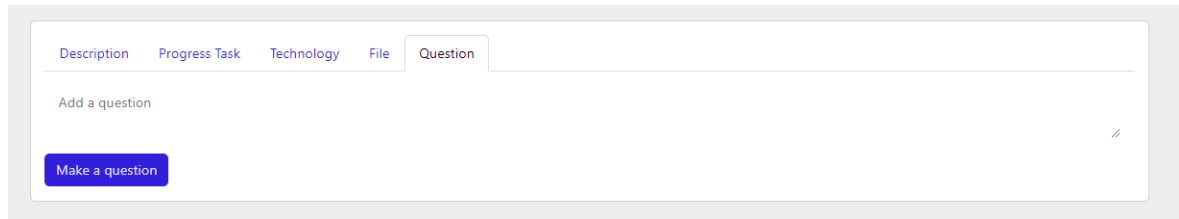
Web

Mobile App

Hình 7.14d: Tab công nghệ



Hình 7.14e: Tab tài nguyên



Hình 7.14f: Tab câu hỏi

STT	Tên phần tử	Chức năng
1	informationCard	Thông tin cơ bản dự án
2	studentCard	Tên sinh viên thực hiện
3	aboutTab	Mô tả đồ án
4	saveBtn	Lưu thay đổi
5	progressTab	Tiến độ dự án
6	addProgressBtn	Thêm tiến độ
7	techTab	Mã công nghệ đồ án
8	addBtn	Thêm thẻ vào đồ án
9	fileTab	Tài nguyên đồ án
10	fileSelect	Lựa chọn file
11	addFileBtn	Thêm tài nguyên


12	questionInput	Nhập câu hỏi
13	addQuestionBtn	Thêm câu hỏi

### 7.3.15. Màn hình thêm tiến độ

Hình 7.15: Màn hình thêm tiến độ

STT	Tên phần tử	Chức năng
1	taskInput	Tên tiến độ
2	participateSelect	Lựa chọn người thực hiện
3	startDatePicker	Ngày bắt đầu
4	endDatePicker	Ngày kết thúc
5	addBtn	Thêm tiến độ

### 7.3.16. Màn hình tìm kiếm đề tài và đăng kí

Project Personal Info Current Project Project List 

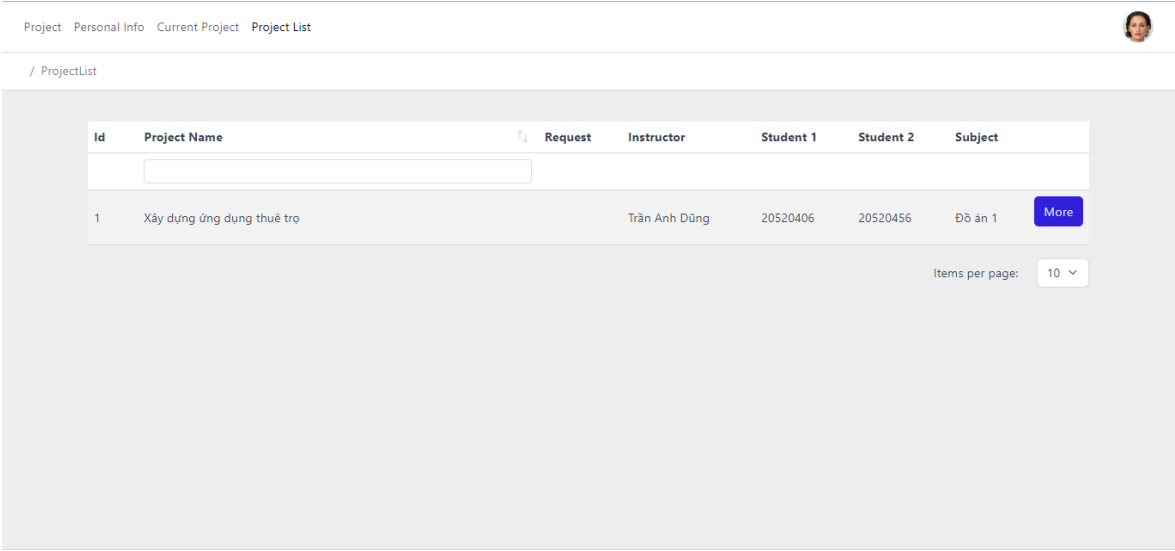
/ Project

<input type="checkbox"/>	Id ↑↓	Topic Name	Request	Instructor	Subject	
<input type="checkbox"/>	20	Xây dựng ứng dụng học nhóm ảo	website	Huỳnh Tuấn Anh	Đồ án 2	<a href="#">Hide</a>
Student's informations						
Student's code 1:		20520544	Student's code 2:	Student 2		
<a href="#">Register</a>						
<input type="checkbox"/>	21	Tìm hiểu và xây dựng nhà sách NFT	website	Huỳnh Tuấn Anh	Đồ án 2	<a href="#">Show</a>
<input type="checkbox"/>	22	Xây dựng ứng dụng blockchain hỗ trợ tuyển việc	website	Huỳnh Tuấn Anh	Đồ án 2	<a href="#">Show</a>
<input type="checkbox"/>	23	Xây dựng ứng dụng thuê trọ	website	Huỳnh Tuấn Anh	Đồ án 2	<a href="#">Show</a>

Hình 7.16: Màn hình tìm kiếm đề tài và đăng kí

STT	Tên phần tử	Chức năng
1	topicNameSearchBox	Tìm kiếm theo tên
2	instructorSearchBox	Tìm kiếm theo giảng viên
3	topicTbl	Danh sách đề tài còn trống
4	showBtn	Hiện thị chi tiết đề tài
5	Student2Input	Nhập sinh viên thứ 2 nếu có
6	registerBtn	Đăng kí đề tài

7.3.17. Màn hình tìm kiếm đề án các khóa trước

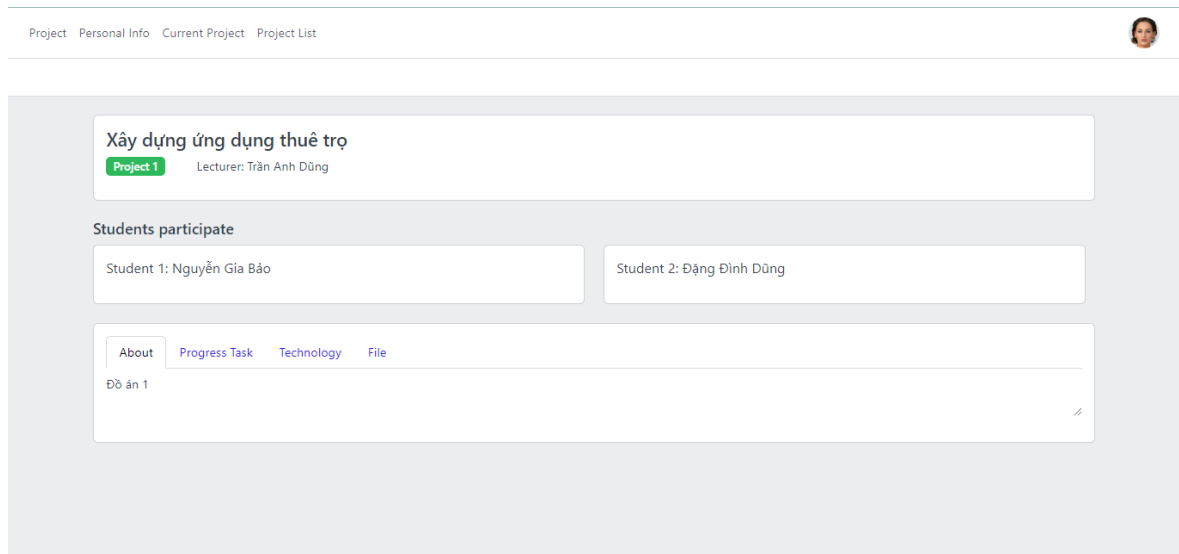


Hình 7.17: Màn hình danh sách đăng kí đề án

STT	Tên phần tử	Chức năng
1	projectNameInput	Tìm kiếm đề án theo tên
2	moreBtn	Chuyển đến trang thông tin chi tiết đề án

7.3.18. Màn hình chi tiết đề án (người dùng)






Hình 7.18: Màn hình chi tiết đồ án (người dùng)

STT	Tên phần tử	Chức năng
1	informationCard	Thông tin cơ bản dự án
2	studentCard	Tên sinh viên thực hiện
3	aboutTab	Mô tả đồ án
4	progressTab	Tiến độ dự án
5	techTab	Mã công nghệ đồ án
6	fileTab	Tài nguyên đồ án

### 7.3.19. Màn hình chỉnh sửa thông tin giảng viên

Personal Info


Thông tin cá nhân

Họ và tênĐỗ Thị Thanh Tuyền

Giới tínhNữ

Ngày sinh01/01/1990 12:00 SA

Địa chỉxxx,TPHCM

Quê quánTPHCM

Emailtuyendtt@uit.edu.vn

Số điện thoại

Học vịThạc sĩ

Lưu thông tin

Hình 7.19: Màn hình chỉnh sửa thông tin giảng viên

STT	Tên phần tử	Chức năng
1	infomationCard	Hiển thị thông tin cơ bản
2	saveBtn	Lưu thông tin chỉnh sửa

### 7.3.20. Màn hình danh sách đề tài bản thân quản lý

Guide project

<input type="checkbox"/> Id	Topic	Request	Description	Instructor	Subject	
<input type="checkbox"/> 1	Ứng dụng đi chợ trực tuyến tích hợp gợi ý món ăn	Yêu thích lập trình Web	Project 1	1	1	More
<input type="checkbox"/> 2	Ứng dụng đi chợ trực tuyến	Yêu thích lập trình Web	Project 1	1	1	More

Items per page: 20

Hình 7.20: Màn hình danh sách đề án bản thân quản lý

STT	Tên phần tử	Chức năng
1	topicNameInput	Tìm kiếm đề án theo tên

---

2	subjectInput	Lọc theo đồ án 1,2
3	projectTbl	Hiển thị danh sách đề tài
4	moreBtn	Chuyển đến trang chi tiết đồ án

---

## IV. Cài đặt và thử nghiệm

STT	Chức năng	Mức độ hoàn thành (%)	Ghi chú
1	Đăng nhập	100	
2	Quản lý giảng viên	100	
3	Quản lý sinh viên	100	
4	Đăng kí đồ án	100	
5	Quản lý tiến độ đồ án	100	
6	Lưu trữ tài nguyên đồ án	100	
7	Quản lý đề tài	100	

## V. Nhận xét và kết luận

### 1. Nhận xét

- Hoàn thiện tất cả nhiệm vụ được giao
- Giao diện được làm chỉnh chu, dễ dàng thao tác cho người dùng.
- Có sự phân quyền kết hợp với việc mã hóa mật khẩu giúp ứng dụng có tính bảo mật cao
- Có thống kê tổng số các dự án dưới dạng đồ thị giúp dễ dàng thống kê hơn
- Hỗ trợ doanh nghiệp quản lý dự án một cách tối ưu thông qua tích hợp việc lưu trữ tài nguyên
- Công việc của nhóm được hoàn thành căn bản..
- Chương trình được đóng gói và cài đặt hoàn hiện.

### 2. Kết luận

#### 2.1. Ưu nhược điểm

##### • Ưu điểm

Hoàn thành đồ án đúng tiến độ

Đáp ứng đầy đủ các yêu cầu của bài toàn

Phát triển thêm một số chức năng

##### • Nhược điểm

Tốc độ xử lý còn hạn chế nên chỉ áp dụng cho quy mô vừa và nhỏ

Chưa có chức năng chat để tương tác giữa thành viên với nhau

#### 2.2. Hướng phát triển

- Cải thiện giao diện trực quan, bắt mắt hơn
- Tối ưu hóa tốc độ xử lý
- Bổ sung thêm các tính năng mới như chat, gọi điện

## VI. Bảng phân công công việc:

Công việc	20520406	20520544
Giới thiệu các bài toán cần giải quyết và mô tả quy trình thực hiện các công việc chính	X	X
Xác định và mô hình hóa yêu cầu phần mềm		X
Thiết kế hệ thống	X	
Thiết kế dữ liệu	X	
Thiết kế giao diện		X
Cài đặt	X	X
Kiểm chứng	X	X
Viết báo cáo		X
Thuyết trình	X	X
Mức độ hoàn thành (%)	100	100

## VII. Phụ lục

*Tài liệu tham khảo:*

React templates installation: <https://coreui.io/react/docs/templates/installation>

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

API là gì? Tại sao API được sử dụng nhiều hiện nay?: <https://topdev.vn/blog/api-la-gi/>

ASP.NET core doc: <https://learn.microsoft.com/en-us/aspnet/core/?view=aspnetcore-8.0>

ASP.NET mvc doc: <https://learn.microsoft.com/en-us/aspnet/mvc/>

SQL server doc: <https://learn.microsoft.com/en-us/sql/?view=sql-server-ver16>