

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



ĐỒ ÁN KẾT THÚC MÔN
PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ
HỌC KỲ I, NĂM HỌC 2024-2025

XÂY DỰNG WEBSITE QUẢN LÝ CÂU LẠC BỘ HÀNH TRÌNH SINH VIÊN

Giảng viên hướng dẫn:
ThS. Ngô Thanh Huy

Sinh viên thực hiện:
Họ tên: Nguyễn Đại Hoàng Phúc
MSSV: 110121087
Họ tên: Phạm Thúy Hằng
MSSV: 110121182
Họ tên: Huỳnh Nguyễn Bích Trâm
MSSV: 110121118

Trà Vinh, tháng 12 năm 2024

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



ĐỒ ÁN KẾT THÚC MÔN
PHÁT TRIỂN PHẦN MỀM HƯỚNG DỊCH VỤ
HỌC KỲ I, NĂM HỌC 2024-2025

XÂY DỰNG WEBSITE QUẢN LÝ CÂU LẠC BỘ HÀNH TRÌNH SINH VIÊN

Giảng viên hướng dẫn:
ThS. Ngô Thanh Huy

Sinh viên thực hiện:
Họ tên: Nguyễn Đại Hoàng Phúc
MSSV: 110121087
Họ tên: Phạm Thúy Hằng
MSSV: 110121182
Họ tên: Huỳnh Nguyễn Bích Trâm
MSSV: 110121118

Trà Vinh, tháng 12 năm 2024

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày tháng năm 2024

Giáo viên hướng dẫn

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày tháng năm 2024

Thành viên hội đồng

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Trà Vinh và Khoa Kỹ thuật và Công nghệ đã tạo điều kiện thuận lợi để chúng em có cơ hội thực hiện đề tài "Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên".

Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến Thầy Ngô Thanh Huy, người đã tận tình hướng dẫn, góp ý và hỗ trợ chúng em trong suốt quá trình nghiên cứu và triển khai đồ án. Những kiến thức và kinh nghiệm quý báu mà thầy truyền đạt đã giúp chúng em hoàn thành đề tài một cách tốt nhất.

Chúng em cũng xin gửi lời tri ân đến Ban chủ nhiệm câu lạc bộ Hành Trình Sinh Viên cùng các anh chị, bạn bè đã hỗ trợ, đóng góp ý kiến và tạo cơ hội để chúng em áp dụng thực tiễn các giải pháp công nghệ vào công tác quản lý câu lạc bộ.

Cuối cùng, chúng em xin cảm ơn gia đình, bạn bè đã luôn động viên, hỗ trợ về mặt tinh thần trong suốt quá trình thực hiện đề tài.

Mặc dù đã cố gắng hoàn thiện, nhưng không thể tránh khỏi những thiếu sót. Chúng em rất mong nhận được những góp ý quý báu để có thể cải thiện và phát triển đề tài tốt hơn trong tương lai.

MỤC LỤC

PHẦN I: XÂY DỰNG API WEBSITE QUẢN LÝ CÂU LẠC BỘ HÀNH TRÌNH SINH VIÊN.....	1
CHƯƠNG 1: TỔNG QUAN	1
1.1 Giới thiệu tổng quan vấn đề :	1
1.2 Mục tiêu nghiên cứu:	1
1.3 Phương pháp nghiên cứu:	2
1.4 Đối tượng và phạm vi nghiên cứu:	2
1.4.1 Đối tượng nghiên cứu:	2
1.4.2 Phạm vi nghiên cứu: Câu lạc bộ Hành trình sinh viên thuộc Khoa Kỹ thuật và Công nghệ, trường Đại học Trà Vinh.....	3
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	4
2.1 Cơ sở lý thuyết.....	4
2.1.1 Giới thiệu về công nghệ ASP.NET API:	4
2.1.2 Giới thiệu về công nghệ ASP.NET MVC 6:	5
2.1.3 Entity Framework Core	6
2.1.4 Bootstrap.....	7
2.1.5 Giới thiệu cơ sở dữ liệu:	7
2.1.6 Giới thiệu ngôn ngữ SQL:	9
2.1.7 Giới thiệu Hệ quản trị cơ sở dữ liệu SQL Server:	11
2.2 Cơ sở lý luận và giả thiết khoa học	13
2.2.1 Cơ sở lý luận:.....	13
2.2.2 Giả thiết khoa học:.....	13
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	15
3.1 Mô tả bài toán:	15

3.2 Giải pháp đề xuất:.....	15
3.3 Yêu cầu chức năng của hệ thống:.....	16
3.3.1 Chức năng của quản trị viên.....	16
3.3.2 Chức năng của thành viên.....	17
3.4 Yêu cầu phi chức năng:	17
3.5 Thiết kế sơ đồ Usecase:.....	18
3.5.1 Sơ đồ Usecase:.....	18
3.5.2 Mô tả sơ đồ Usecase:.....	18
3.6 Thiết kế cơ sở dữ liệu:	21
3.6.1 Lược đồ CSDL:	21
3.7 Kết quả đạt được.....	22
3.7.1 Các API đã xây dựng.....	22
3.7.2 Giao diện	26
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	31
4.1 Kết quả đạt được:.....	31
4.2 Hạn chế:.....	31
4.3 Hướng phát triển:.....	31
PHẦN II: MINI CHAT SOCKET	32
CHƯƠNG 1: TỔNG QUAN	32
1.1 Giới thiệu tổng quan vấn đề :	32
1.2 Mục tiêu nghiên cứu:.....	32
1.3 Phương pháp nghiên cứu:	33
1.4 Đối tượng và phạm vi nghiên cứu:	33
1.4.1 Đối tượng nghiên cứu:.....	33
1.4.2 Phạm vi nghiên cứu:	33

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT	35
2.1 Cơ sở lý thuyết.....	35
2.1.1 Giới thiệu về công nghệ:.....	35
2.1.2 Nguyên lý hoạt động:	35
2.2 Cơ sở lý luận và giả thiết khoa học	35
2.2.1 Cơ sở lý luận:.....	35
2.2.2 Giả thiết khoa học:.....	36
2.3 Kiến trúc hệ thống	36
2.3.1 Kiến trúc Server:.....	36
2.3.2 Kiến trúc Client:	36
2.4 Các Chức Năng Chính.....	36
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU	38
3.1 Giao diện mini chat	38
CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	40
4.1 Kết quả đạt được:.....	40
4.2 Hạn chế:.....	40
4.3 Hướng phát triển:.....	40

DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ Usecase.....	18
Hình 2: Lược đồ CSDL	21
Hình 3: API quản lý chức vụ	22
Hình 4: API quản lý hoạt động.....	22
Hình 5: API quản lý nhiệm vụ.....	23
Hình 6: API quản lý tài chính.....	23
Hình 7: API quản lý phân công	23
Hình 8: API quản lý tham gia.....	23
Hình 9: API quản lý quyền.....	24
Hình 10: API quản lý tin tức	24
Hình 11: API quản lý lớp học.....	24
Hình 12: API quản lý khoa trường	25
Hình 13: API quản lý đăng ký.....	25
Hình 14: API quản lý thông kê.....	25
Hình 15: API quản lý người dùng	25
Hình 16: Trang chủ.....	26
Hình 17: Đăng nhập.....	26
Hình 18: Section tin tức.....	27
Hình 19: Section các hoạt động và giới thiệu Ban chủ nhiệm.....	27
Hình 20: Trang chi tiết tin tức	27
Hình 21: Trang đăng ký hoạt động.....	28
Hình 22: Trang Tổng quan Admin	28
Hình 23: Trang quản lý người dùng	29
Hình 24: Trang quản lý hoạt động.....	29

Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên

Hình 25: Trang xem danh sách đăng ký hoạt động	29
Hình 26: Trang xem danh sách tham gia hoạt động	30
Hình 27: Trang quản lý tin tức	30
Hình 28: Trang danh sách đăng ký	30
Hình 29: Đăng nhập	38
Hình 30: Giao diện người dùng chung	38
Hình 31: Giao diện người dùng cá nhân	39

TÓM TẮT ĐỒ ÁN

Đồ án “Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên” nhằm phát triển một hệ thống quản lý thông tin linh hoạt và hiệu quả, hỗ trợ các hoạt động của câu lạc bộ tại Trường Đại học Trà Vinh.

Hệ thống tập trung vào việc đơn giản hóa quản lý thành viên, tổ chức hoạt động, và cập nhật tin tức, góp phần nâng cao tinh thần đoàn kết và kết nối giữa các sinh viên. Được xây dựng trên nền tảng .NET API ở Backend và .NET MVC ở Frontend, hệ thống đảm bảo tính ổn định, hiệu quả, và dễ dàng bảo trì.

Mục tiêu chính của đồ án là tạo ra một ứng dụng web tối ưu hóa các chức năng quản lý cơ bản, đáp ứng nhanh chóng nhu cầu của người dùng, đồng thời đảm bảo thông tin quản lý đầy đủ và chính xác.

Kết quả đạt được sẽ là một hệ thống quản lý trực tuyến hoàn chỉnh, góp phần tăng cường sự tham gia và gắn kết của sinh viên trong các hoạt động câu lạc bộ.

MỞ ĐẦU

Câu lạc bộ Hành trình Sinh viên là một trong những câu lạc bộ thuộc Trường Đại học Trà Vinh, góp phần tạo ra môi trường học thuật, giao lưu và hỗ trợ sinh viên phát triển toàn diện. Tuy nhiên, công tác quản lý hiện tại, bao gồm quản lý thành viên, tổ chức hoạt động và thông tin liên lạc, vẫn còn nhiều hạn chế do chủ yếu dựa trên các phương pháp thủ công. Điều này dễ dẫn đến sai sót, thiếu hiệu quả và khó khăn trong việc đáp ứng nhu cầu ngày càng cao của câu lạc bộ.

Việc ứng dụng công nghệ thông tin vào công tác quản lý không chỉ giúp tối ưu hóa quy trình hoạt động mà còn nâng cao tính chuyên nghiệp, hiện đại hóa hệ thống quản lý, đồng thời tăng cường hiệu quả trong việc kết nối và tổ chức.

Trên cơ sở đó, đề tài “Xây dựng website quản lý câu lạc bộ Hành trình Sinh viên” được lựa chọn nghiên cứu và triển khai, nhằm khắc phục những hạn chế hiện có, đáp ứng tốt hơn nhu cầu thực tiễn và đóng góp vào sự phát triển của câu lạc bộ.

DANH MỤC TỪ VIẾT TẮT

TỪ VIẾT TẮT	DIỄN GIẢI
CLB	Câu lạc bộ
CSDL	Cơ sở dữ liệu
MVC	Model, View, Controller
CI/CD	Continuous Integration, Continuous Delivery/Continuous Deployment
SQL	Structured Query Language
API	Application Programming Interface

PHẦN I: XÂY DỰNG API WEBSITE QUẢN LÝ CÂU LẠC BỘ HÀNH TRÌNH SINH VIÊN

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu tổng quan vấn đề :

Trong bối cảnh các hoạt động của các câu lạc bộ dành cho sinh viên ngày càng phát triển và mở rộng, việc quản lý truyền thống bằng giấy tờ hoặc bảng tính Excel đang bộc lộ nhiều hạn chế. Các phương pháp này không chỉ gây khó khăn trong việc tổ chức dữ liệu mà còn tiêu tốn thời gian và dễ xảy ra sai sót. Điều này ảnh hưởng đến hiệu quả hoạt động và sự phát triển của các câu lạc bộ.

Với sự phát triển của công nghệ, việc ứng dụng một hệ thống quản lý hiện đại là cần thiết để tối ưu hóa quy trình, tiết kiệm nguồn lực và nâng cao trải nghiệm người dùng. Dự án "Xây dựng website quản lý câu lạc bộ Hành trình Sinh Viên" ra đời nhằm giải quyết các hạn chế như công tác quản lý còn chưa được hiện đại, đồng thời cung cấp một công cụ hữu ích giúp câu lạc bộ quản lý thành viên và các hoạt động hiệu quả hơn.

1.2 Mục tiêu nghiên cứu:

Mục tiêu của dự án là xây dựng một website quản lý câu lạc bộ Hành trình sinh viên với các tính năng sau:

- Quản lý thông tin thành viên: Cho phép quản trị viên và các thành viên trong câu lạc bộ cập nhật và quản lý thông tin cá nhân, trạng thái hoạt động, và các thông tin liên quan.
- Quản lý hoạt động: Cho phép các thành viên đăng ký để tham gia các hoạt động, cũng như theo dõi và tra cứu danh sách tham gia các hoạt động do câu lạc bộ tổ chức.
- Quản lý tài chính: Cho phép quản trị viên có thể quản lý các khoản thu chi trong câu lạc bộ và có thể thống kê lại theo một khoản thời gian nào đó.

- Phân quyền người dùng: Hệ thống phân quyền cho phép phân biệt giữa các vai trò khác nhau như quản trị viên, thành viên và khách mời, từ đó cung cấp các quyền truy cập và chức năng tương ứng.
- Giao diện người dùng thân thiện và hiện đại: Xây dựng giao diện web sử dụng các công nghệ web hiện đại như HTML, CSS, và JavaScript kết hợp với Bootstrap để tạo ra trải nghiệm người dùng dễ dàng và tối ưu.

1.3 Phương pháp nghiên cứu:

- Khảo sát thực tế công tác quản lý của câu lạc bộ Hành trình sinh viên từ ban chủ nhiệm câu lạc bộ và các thành viên của câu lạc bộ.
- Nghiên cứu tài liệu: Tìm hiểu các hệ thống quản lý câu lạc bộ hiện có, các công nghệ phù hợp như .NET MVC 6, Bootstrap, và các mô hình phân quyền.
- Phân tích yêu cầu: Tiến hành khảo sát người dùng mục tiêu, bao gồm ban chủ nhiệm và các thành viên của câu lạc bộ, để xác định rõ các yêu cầu và mong muốn của họ đối với hệ thống.

1.4 Đối tượng và phạm vi nghiên cứu:

1.4.1 Đối tượng nghiên cứu:

- Các câu lạc bộ dành cho sinh viên, đặc biệt là ban chủ nhiệm và các thành viên của câu lạc bộ Hành trình Sinh Viên – Trường Đại học Trà Vinh.
- Các quy trình quản lý thông tin và tổ chức hoạt động trong câu lạc bộ.
- Công nghệ ASP.NET API.
- Công nghệ ASP.NET MVC 6.
- Java
- Kiến trúc Client-Server
- Entity Framework Core.
- Cơ sở dữ liệu, ngôn ngữ SQL và hệ quản trị cơ sở dữ liệu SQL Server.

1.4.2 Phạm vi nghiên cứu: Câu lạc bộ Hành trình sinh viên thuộc Khoa Kỹ thuật và Công nghệ, trường Đại học Trà Vinh.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Cơ sở lý thuyết

2.1.1 Giới thiệu về công nghệ ASP.NET API:

.NET API là một phần của nền tảng .NET, được sử dụng để xây dựng các dịch vụ web RESTful (Representational State Transfer) hoặc gRPC (gRPC Remote Procedure Call) nhằm cung cấp dữ liệu hoặc chức năng cho các ứng dụng khách (client), chẳng hạn như ứng dụng web, di động, hoặc desktop. Đây là giải pháp mạnh mẽ, linh hoạt và hiệu quả để phát triển các hệ thống phân tán và giao tiếp qua mạng.

- Các điểm nổi bật của .NET API:

+ Cross-Platform: .NET API được xây dựng trên .NET Core (nay là .NET), cho phép chạy trên nhiều nền tảng như Windows, Linux, và macOS.

+ Hiệu năng cao: .NET API được tối ưu hóa để đạt hiệu suất cao, phù hợp cho các ứng dụng yêu cầu xử lý nhanh và đáp ứng tức thời.

+ RESTful API: Với ASP.NET Core, bạn có thể dễ dàng xây dựng các API RESTful để xử lý dữ liệu thông qua các phương thức HTTP (GET, POST, PUT, DELETE).

+ gRPC Support: Ngoài RESTful, .NET API hỗ trợ gRPC, một giao thức hiện đại và hiệu suất cao dựa trên HTTP/2.

+ Dễ dàng tích hợp: .NET API hỗ trợ tích hợp với nhiều cơ sở dữ liệu (SQL Server, MySQL, PostgreSQL, MongoDB), các hệ thống lưu trữ, và dịch vụ bên thứ ba như thanh toán, email, hoặc dịch vụ đám mây.

+ Bảo mật mạnh mẽ: .NET API cung cấp các cơ chế bảo mật mạnh mẽ như xác thực (Authentication), phân quyền (Authorization), và mã hóa dữ liệu (Data Encryption).

+ Middleware và Dependency Injection (DI) .NET API sử dụng mô hình pipeline với middleware giúp xử lý các yêu cầu HTTP hiệu quả. Dependency Injection được tích hợp sẵn, giúp code dễ bảo trì và mở rộng.

1. Ứng dụng của .NET API

- Xây dựng backend cho các ứng dụng web và di động.
- Cung cấp dữ liệu cho các ứng dụng IoT hoặc hệ thống lớn.
- Tích hợp với các dịch vụ bên thứ ba.
- Tạo dịch vụ microservices trong các hệ thống phân tán.
- Làm API để kết nối giữa các hệ thống nội bộ trong doanh nghiệp.

2.1.2 Giới thiệu về công nghệ ASP.NET MVC 6:

ASP.NET MVC (Model-View-Controller) là một framework phát triển ứng dụng web mạnh mẽ được Microsoft phát triển, cho phép tách biệt ba thành phần chính trong ứng dụng:

- Model: Đại diện cho dữ liệu và logic nghiệp vụ. Model chịu trách nhiệm quản lý trạng thái dữ liệu và giao tiếp với cơ sở dữ liệu.
- View: Xử lý giao diện người dùng. View nhận dữ liệu từ Model và hiển thị thông tin theo định dạng thân thiện với người dùng.
- Controller: Là trung gian giữa Model và View, chịu trách nhiệm xử lý yêu cầu từ người dùng, gọi dữ liệu từ Model, và trả về View phù hợp.

ASP.NET MVC 6 là phiên bản nâng cấp trong ASP.NET Core, cung cấp hiệu suất tốt hơn, hỗ trợ đa nền tảng (Windows, Linux, macOS) và cải thiện trải nghiệm lập trình thông qua các tính năng:

- Dependency Injection (DI): Tích hợp DI sẵn có, giúp quản lý sự phụ thuộc giữa các thành phần dễ dàng hơn.

- Entity Framework Core (EF Core): Giải pháp ORM (Object-Relational Mapping) mạnh mẽ để giao tiếp với cơ sở dữ liệu.
- Razor Pages: Một cách tiếp cận đơn giản hơn cho việc xây dựng giao diện tương tác.
- Middleware: Kiến trúc pipeline linh hoạt, cho phép kiểm soát quy trình xử lý yêu cầu HTTP.
- Cross-Platform Support: Hỗ trợ xây dựng ứng dụng chạy được trên nhiều hệ điều hành.

2.1.3 Entity Framework Core

Entity Framework Core là một thư viện ORM (Object-Relational Mapping) mạnh mẽ, được phát triển bởi Microsoft, giúp các nhà phát triển .NET làm việc với cơ sở dữ liệu theo cách hướng đối tượng mà không cần viết nhiều câu lệnh SQL phức tạp. EF Core là phiên bản cải tiến và nhẹ hơn của Entity Framework, được thiết kế để hoạt động hiệu quả trên cả .NET Framework và .NET Core, hỗ trợ đa nền tảng (Windows, Linux, macOS).

EF Core cho phép phát triển ứng dụng một cách nhanh chóng bằng cách ánh xạ dữ liệu từ cơ sở dữ liệu vào các đối tượng trong ứng dụng và ngược lại. Với EF Core, các thao tác như tạo, đọc, cập nhật, và xóa trở nên dễ dàng và trực quan hơn. Các khái niệm chính của EF Core bao gồm:

- DbContext
 - + DbContext là lớp trung tâm trong Entity Framework Core , đóng vai trò như một cầu nối giữa ứng dụng và cơ sở dữ liệu.
 - + Nó quản lý các phiên làm việc với cơ sở dữ liệu, bao gồm việc truy vấn, lưu trữ và thay đổi dữ liệu.
 - + DbContext cung cấp các phương thức và thuộc tính giúp thực hiện các thao tác CRUD và truy vấn dữ liệu bằng LINQ.
- DbSet

- + DbSet đại diện cho các bảng trong cơ sở dữ liệu.
- + Mỗi DbSet ánh xạ một lớp trong ứng dụng với một bảng cụ thể trong cơ sở dữ liệu.
- + Nó cho phép thực hiện các thao tác như thêm, sửa, xóa, và truy vấn dữ liệu.
- Migration
 - + Migration là một tính năng quan trọng của EF Core, cho phép quản lý và theo dõi các thay đổi trong cấu trúc cơ sở dữ liệu theo thời gian.
 - + Khi bạn thêm hoặc chỉnh sửa các lớp mô hình (Model), Migration sẽ tạo ra các tệp mã lệnh để cập nhật cơ sở dữ liệu phù hợp với các thay đổi này.
 - + Điều này giúp đồng bộ hóa cấu trúc dữ liệu giữa ứng dụng và cơ sở dữ liệu mà không cần can thiệp trực tiếp vào cơ sở dữ liệu.

2.1.4 Bootstrap

Bootstrap là một framework HTML, CSS, và JavaScript phổ biến để xây dựng giao diện web thân thiện và đáp ứng tốt trên các thiết bị di động. Bootstrap cung cấp các thành phần giao diện sẵn có như bảng, form, button, và grid system, giúp tăng tốc quá trình phát triển giao diện.

2.1.5 Giới thiệu cơ sở dữ liệu:

Cơ sở dữ liệu (CSDL) là hệ thống lưu trữ và quản lý dữ liệu một cách có tổ chức. CSDL đóng vai trò quan trọng trong các ứng dụng hiện đại, cho phép lưu trữ, truy xuất và xử lý thông tin một cách hiệu quả. Dữ liệu trong CSDL được tổ chức dưới dạng bảng, mỗi bảng chứa các bản ghi (record) đại diện cho thông tin cụ thể và các cột (field) biểu thị thuộc tính của dữ liệu.

- Ưu điểm của cơ sở dữ liệu:

- + Quản lý dữ liệu hiệu quả: CSDL cho phép lưu trữ khối lượng lớn thông tin một cách có tổ chức và dễ dàng truy xuất.
- + Đảm bảo tính toàn vẹn và nhất quán dữ liệu: Các quy tắc như khóa chính (Primary Key) và khóa ngoại (Foreign Key) giúp đảm bảo dữ liệu không bị trùng lặp hoặc không đầy đủ.
- + Truy vấn và thao tác linh hoạt: Ngôn ngữ SQL cho phép thực hiện các thao tác như thêm mới, cập nhật, xóa hoặc truy vấn dữ liệu với hiệu quả cao.
- + Bảo mật và phân quyền: Các hệ quản trị CSDL hiện đại hỗ trợ cơ chế phân quyền và mã hóa dữ liệu, giúp bảo vệ thông tin khỏi truy cập trái phép.
- + Hỗ trợ mở rộng: CSDL có khả năng mở rộng để xử lý dữ liệu lớn và đáp ứng nhu cầu của các ứng dụng phát triển.
- Nhược điểm của cơ sở dữ liệu:
 - + Chi phí triển khai và bảo trì cao: Việc thiết kế, triển khai và duy trì một hệ thống cơ sở dữ liệu phức tạp thường yêu cầu nguồn lực và chuyên môn cao.
 - + Hiệu suất giảm khi dữ liệu lớn: Khi kích thước dữ liệu tăng lên đáng kể, việc truy vấn hoặc xử lý dữ liệu có thể trở nên chậm nếu không tối ưu hóa đúng cách.
 - + Phụ thuộc vào hệ thống quản trị: Các ứng dụng thường phải tương thích với hệ quản trị CSDL cụ thể, gây khó khăn nếu cần chuyển đổi giữa các nền tảng.
 - + Nguy cơ mất dữ liệu: Nếu không có các biện pháp sao lưu và phục hồi dữ liệu, hệ thống có thể bị mất mát thông tin trong trường hợp xảy ra sự cố.

Cơ sở dữ liệu là thành phần không thể thiếu trong việc quản lý thông tin một cách hiệu quả, nhưng cần được thiết kế và vận hành đúng cách để tận dụng tối đa các ưu điểm và hạn chế các nhược điểm.

2.1.6 Giới thiệu ngôn ngữ SQL:

SQL (Structured Query Language) là ngôn ngữ tiêu chuẩn được sử dụng để giao tiếp và quản lý dữ liệu trong các hệ quản trị cơ sở dữ liệu quan hệ như Microsoft SQL Server, MySQL, PostgreSQL, và Oracle Database. SQL cho phép thực hiện các thao tác trên dữ liệu, từ truy vấn, cập nhật, đến quản lý cấu trúc cơ sở dữ liệu.

- Data Query Language

- + DQL sử dụng lệnh SELECT để truy vấn và lấy dữ liệu từ cơ sở dữ liệu.

- + Cho phép thực hiện các phép lọc, sắp xếp, và nhóm dữ liệu nhằm trả về kết quả theo yêu cầu.

- Data Definition Language

- + DDL được sử dụng để định nghĩa và quản lý cấu trúc cơ sở dữ liệu như tạo bảng, sửa đổi cấu trúc bảng, hoặc xóa bảng.

- + Các lệnh chính:

- CREATE: Tạo mới các đối tượng như bảng, schema.
 - ALTER: Sửa đổi cấu trúc của đối tượng.
 - DROP: Xóa các đối tượng không còn cần thiết.

- Data Manipulation Language

- + DML chịu trách nhiệm thao tác trực tiếp với dữ liệu trong bảng.

- + Các lệnh chính:

- INSERT: Thêm mới dữ liệu.

- UPDATE: Cập nhật dữ liệu hiện có.
 - DELETE: Xóa dữ liệu không còn cần thiết.
- Data Control Language
 - + DCL quản lý quyền truy cập dữ liệu trong cơ sở dữ liệu, bảo đảm an toàn và bảo mật.
 - + Các lệnh chính:
 - GRANT: Cấp quyền cho người dùng.
 - REVOKE: Thu hồi quyền đã cấp.
- Transaction Control Language
 - + TCL hỗ trợ quản lý các giao dịch (transactions) trong cơ sở dữ liệu, đảm bảo tính toàn vẹn của dữ liệu khi thực hiện nhiều thao tác.
 - + Các lệnh chính:
 - BEGIN TRANSACTION: Bắt đầu giao dịch.
 - COMMIT: Xác nhận thay đổi.
 - ROLLBACK: Hủy bỏ các thay đổi trong giao dịch.
- Ưu điểm của SQL
 - + Hỗ trợ thao tác dữ liệu mạnh mẽ và linh hoạt.
 - + Có khả năng xử lý truy vấn dữ liệu phức tạp với hiệu suất cao.
 - + Dễ dàng tích hợp với các ngôn ngữ lập trình và ứng dụng.
 - + Chuẩn hóa, đảm bảo tính nhất quán giữa các hệ quản trị cơ sở dữ liệu.
- Nhược điểm của SQL
 - + Không thích hợp cho dữ liệu phi quan hệ hoặc cấu trúc phức tạp.
 - + Yêu cầu tối ưu hóa truy vấn và cấu trúc dữ liệu để đảm bảo hiệu suất với dữ liệu lớn.

- + Sự khác biệt trong việc triển khai các lệnh SQL giữa các hệ quản trị cơ sở dữ liệu có thể gây khó khăn khi chuyển đổi.

2.1.7 Giới thiệu Hệ quản trị cơ sở dữ liệu SQL Server:

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) được phát triển bởi Microsoft, được thiết kế để lưu trữ, truy vấn và quản lý dữ liệu một cách hiệu quả. SQL Server được sử dụng phổ biến trong các ứng dụng doanh nghiệp nhờ khả năng xử lý dữ liệu mạnh mẽ, tích hợp tốt với hệ sinh thái Microsoft và các công cụ phân tích dữ liệu.

- Các tính năng chính của SQL Server:

- + Khả năng quản lý dữ liệu lớn: SQL Server hỗ trợ xử lý dữ liệu ở quy mô lớn với hiệu suất cao và tính nhất quán cao.
- + Hỗ trợ đa nền tảng: Ngoài Windows, SQL Server còn hoạt động trên các hệ điều hành khác như Linux và Docker.
- + Tích hợp với các công cụ phân tích: Hỗ trợ các công cụ như SQL Server Reporting Services, SQL Server Analysis Services, giúp phân tích và báo cáo dữ liệu hiệu quả.
- + Bảo mật mạnh mẽ: Cung cấp các tính năng bảo mật tiên tiến như mã hóa dữ liệu, quản lý quyền truy cập, và bảo mật cấp hàng.
- + Hỗ trợ lập trình: Tích hợp tốt với các ngôn ngữ lập trình như .NET, Java, và Python, giúp các nhà phát triển dễ dàng xây dựng ứng dụng.
- + High Availability: Cung cấp các giải pháp đảm bảo tính sẵn sàng cao như Always On Availability Groups, Database Mirroring, và Clustering.

- Các thành phần cơ bản của SQL Server:

- + Database Engine: Thành phần cốt lõi chịu trách nhiệm lưu trữ, truy vấn và xử lý dữ liệu; Hỗ trợ các tính năng như giao dịch, stored procedures, và triggers.

- + SQL Server Management Studio (SSMS): Công cụ giao diện đồ họa để quản trị cơ sở dữ liệu, chạy truy vấn, và giám sát hiệu suất.
- + SQL Server Agent: Dịch vụ giúp tự động hóa các tác vụ như sao lưu dữ liệu, bảo trì hệ thống, và lập lịch công việc (job scheduling).
- + Integration Services: Cung cấp khả năng tích hợp dữ liệu từ nhiều nguồn khác nhau, hỗ trợ ETL (Extract, Transform, Load).
- + Analysis Services: Thành phần hỗ trợ phân tích dữ liệu trực tuyến và xây dựng mô hình dữ liệu.
- + Reporting Services: Hỗ trợ tạo và quản lý các báo cáo trực quan, đáp ứng nhu cầu phân tích dữ liệu.
- Ưu điểm của SQL Server:
 - + Hiệu suất cao: Khả năng xử lý nhanh chóng các truy vấn phức tạp.
 - + Tính bảo mật mạnh mẽ: Bảo vệ dữ liệu ở nhiều cấp độ, từ cơ sở dữ liệu đến ứng dụng.
 - + Tích hợp tốt với hệ sinh thái Microsoft: Tương thích với các công cụ như Azure, Power BI, và Microsoft Office.
 - + Hỗ trợ mạnh mẽ cho các giao dịch: Đảm bảo tính toàn vẹn và nhất quán của dữ liệu.
- Nhược điểm của SQL Server:
 - + Chi phí cao: Phiên bản thương mại có chi phí lớn, đặc biệt với các tổ chức quy mô nhỏ.
 - + Yêu cầu tài nguyên: SQL Server cần cấu hình phần cứng tốt để hoạt động hiệu quả với khối lượng dữ liệu lớn.
 - + Hạn chế trong việc tích hợp: So với một số hệ quản trị khác, việc tích hợp SQL Server với các nền tảng phi Microsoft có thể phức tạp hơn.
- Các phiên bản của SQL Server

- + SQL Server Express: Miễn phí, phù hợp cho các ứng dụng nhỏ.
- + SQL Server Standard: Dành cho doanh nghiệp vừa và nhỏ, cung cấp các tính năng cơ bản.
- + SQL Server Enterprise: Phiên bản cao cấp với đầy đủ tính năng, phù hợp cho các tổ chức lớn.
- + SQL Server Developer: Phiên bản miễn phí dành cho phát triển và thử nghiệm.

SQL Server là một hệ quản trị cơ sở dữ liệu mạnh mẽ, đáng tin cậy, được ứng dụng rộng rãi trong các hệ thống thông tin hiện đại, từ ứng dụng web đến các hệ thống doanh nghiệp phức tạp.

2.2 Cơ sở lý luận và giả thiết khoa học

2.2.1 Cơ sở lý luận:

Quản lý thông tin và hoạt động trong các câu lạc bộ dành cho sinh viên là một nhu cầu thực tế, đòi hỏi một hệ thống hiệu quả để:

- Hỗ trợ quản trị viên trong việc quản lý thành viên và tổ chức hoạt động.
- Đáp ứng yêu cầu hiện đại hóa, giúp tiết kiệm thời gian và tài nguyên so với các phương pháp quản lý truyền thống.

Hệ thống quản lý câu lạc bộ được xây dựng trên nền tảng ASP.NET API và MVC 6, kết hợp với Bootstrap và SQL Server, nhằm đảm bảo các yêu cầu về bảo mật và trải nghiệm người dùng.

2.2.2 Giả thiết khoa học:

- Việc áp dụng ASP.NET API và MVC 6 kết hợp với EF Core sẽ giúp tối ưu hóa quy trình phát triển ứng dụng quản lý câu lạc bộ, đảm bảo khả năng mở rộng và bảo trì.
- Sử dụng Bootstrap giúp xây dựng giao diện thân thiện, tăng cường trải nghiệm người dùng trên nhiều loại thiết bị.

- Hệ thống quản lý thành viên và hoạt động được triển khai có thể đáp ứng đầy đủ yêu cầu của các câu lạc bộ sinh viên, giúp cải thiện hiệu quả quản lý và tương tác.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả bài toán:

Câu lạc bộ Hành trình sinh viên là một trong những câu lạc bộ đóng vai trò quan trọng trong việc phát triển kỹ năng mềm, mở rộng mối quan hệ và tạo môi trường học tập, hoạt động ngoại khóa cho sinh viên thuộc Trường Đại học Trà Vinh. Tuy nhiên, việc quản lý các thông tin liên quan đến thành viên, hoạt động và các danh sách... câu lạc bộ hiện nay thường dựa trên phương pháp truyền thống như giấy tờ hoặc bảng tính Excel. Các phương pháp này dẫn đến một số vấn đề:

- Khó khăn trong quản lý thông tin:
 - + Số lượng thành viên ngày càng tăng khiến việc lưu trữ và cập nhật thông tin trở nên phức tạp.
 - + Dữ liệu dễ bị mất mát, sai sót hoặc khó truy xuất.
- Hạn chế trong tổ chức các hoạt động:
 - + Quy trình đăng ký và theo dõi hoạt động thủ công gây mất thời gian, thiếu hiệu quả.
 - + Khó thống kê thông tin tham gia và đánh giá hiệu quả của các hoạt động.
- Phân quyền chưa rõ ràng: Các vai trò như quản trị viên, thành viên và khách mời không được quản lý chặt chẽ, dẫn đến các vấn đề về quyền truy cập dữ liệu.
- Trải nghiệm người dùng hạn chế: Giao diện quản lý thủ công không thân thiện và thiếu tính trực quan, khiến việc sử dụng trở nên khó khăn.

3.2 Giải pháp đề xuất:

Để giải quyết những vấn đề này, hệ thống "Website quản lý câu lạc bộ Hành trình Sinh Viên" được phát triển nhằm:

- Tích hợp các chức năng quản lý thông tin thành viên, tin tức và hoạt động,...của câu lạc bộ vào một nền tảng trực tuyến.
- Cung cấp giao diện thân thiện, hiện đại, hỗ trợ người dùng dễ dàng thao tác.
- Hỗ trợ phân quyền chi tiết cho các vai trò khác nhau để đảm bảo bảo mật và tính chính xác trong truy cập dữ liệu.
- Nâng cao hiệu quả tổ chức và quản lý, đồng thời tạo điều kiện phát triển bền vững cho các câu lạc bộ sinh viên.

Hệ thống không chỉ giúp tối ưu hóa quy trình quản lý mà còn mang lại sự tiện lợi và trải nghiệm tốt hơn cho cả ban chủ nhiệm và thành viên trong câu lạc bộ.

3.3 Yêu cầu chức năng của hệ thống:

3.3.1 Chức năng của quản trị viên

- Quản lý thành viên:
 - + Thêm, sửa, xóa thông tin thành viên.
 - + Xác minh thông tin thành viên và phân quyền.
- Quản lý hoạt động:
 - + Tạo mới, chỉnh sửa, xóa các hoạt động của câu lạc bộ.
 - + Gửi thông báo và cập nhật thông tin hoạt động đến các thành viên.
 - + Theo dõi và quản lý quá trình tham gia hoạt động của thành viên.
- Quản lý tài chính của câu lạc bộ:
 - + Thêm, sửa, xóa các giao dịch tài chính trong câu lạc bộ.
 - + Thống kê các khoản thu chi trong một khoảng thời gian nhất định.
- Quản lý tin tức:
 - + Đăng tải và lưu trữ các tin tức liên quan đến các hoạt động của câu lạc bộ.
 - + Cho phép thành viên tra cứu các thông tin và tin tức cần thiết.

- Xem thống kê các hoạt động.

3.3.2 Chức năng của thành viên

- Tra cứu thông tin:
 - + Đăng nhập để tra cứu thông tin thành viên.
 - + Xem lịch trình hoạt động.
 - + Tra cứu danh sách minh chứng sau khi tham gia hoạt động.
 - + Điểm danh khi tham gia hoạt động (thông qua GPS tại địa điểm diễn ra hoạt động hoặc điểm danh bằng cách nhận diện khuôn mặt).
- Tham gia hoạt động:
 - + Đăng ký tham gia các hoạt động trực tuyến.
 - + Nhận thông báo về các hoạt động và những thay đổi liên quan.

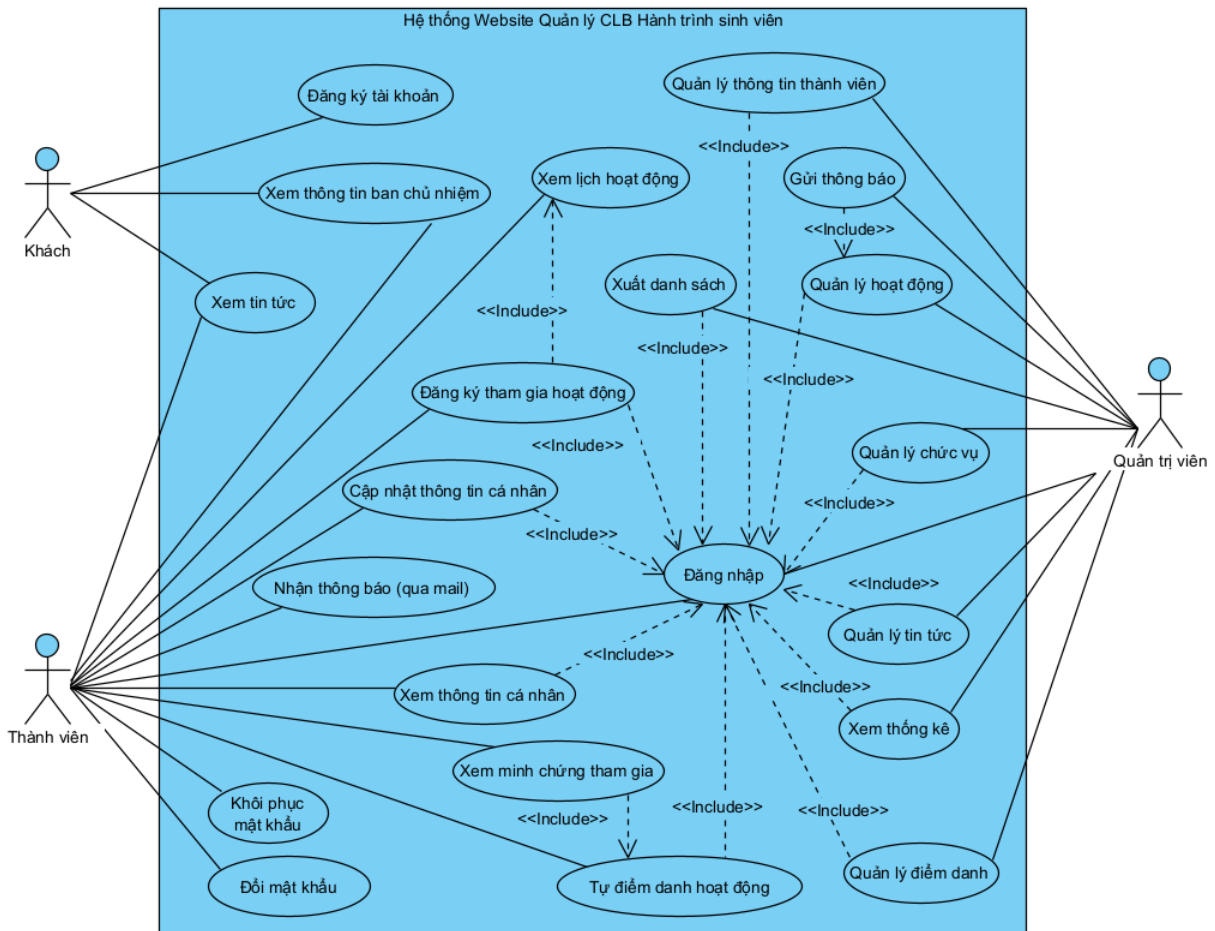
3.4 Yêu cầu phi chức năng:

- Khả năng mở rộng: Hệ thống được thiết kế để dễ dàng bổ sung các chức năng mới trong tương lai mà không làm gián đoạn các chức năng hiện tại.
- Tính bảo mật
 - + Sử dụng HTTPS để mã hóa dữ liệu trao đổi giữa client và server.
 - + Hệ thống quản lý quyền truy cập chặt chẽ dựa trên vai trò
 - + Mật khẩu của người dùng được lưu trữ dưới dạng băm (hash) và mã hóa.
- Khả năng tương thích
 - + Giao diện hoạt động tốt trên các trình duyệt phổ biến như Google Chrome, Firefox, Edge và Safari.
 - + Hỗ trợ trên cả máy tính và thiết bị di động.
- Khả năng bảo trì: Code được viết theo kiến trúc MVC để dễ dàng bảo trì và nâng cấp.

- Tính thẩm mỹ: Giao diện được thiết kế đơn giản, thân thiện, sử dụng màu sắc và bố cục hài hòa để tăng trải nghiệm người dùng.

3.5 Thiết kế sơ đồ Usecase:

3.5.1 Sơ đồ Usecase:



Hình 1: Sơ đồ Usecase

3.5.2 Mô tả sơ đồ Usecase:

- Các Actor chính:

+ Khách: Là những người chưa có tài khoản hoặc chưa đăng nhập vào hệ thống. Có thể thực hiện một số thao tác như:

- Xem tin tức: Truy cập và xem các bài viết, thông tin liên quan đến câu lạc bộ.

- Đăng ký tài khoản: Đăng ký một tài khoản mới để trở thành thành viên.
- Xem thông tin ban chủ nhiệm: Truy cập thông tin về ban chủ nhiệm của CLB.

+ Thành viên: Là người dùng đã đăng nhập vào hệ thống với tư cách thành viên. Có thể thực hiện các chức năng như:

- Cập nhật thông tin cá nhân: Chỉnh sửa các thông tin của bản thân trên hệ thống.
- Đăng ký tham gia hoạt động: Gửi yêu cầu tham gia các hoạt động được tổ chức bởi CLB.
- Xem thông tin cá nhân: Kiểm tra thông tin đã lưu trữ trong tài khoản.
- Xem minh chứng tham gia: Xem và tải xuống các chứng chỉ hoặc minh chứng khi tham gia các hoạt động.
- Điểm danh hoạt động: Tham gia điểm danh khi tham gia các hoạt động của CLB.
- Khôi phục mật khẩu: Lấy lại mật khẩu khi quên.
- Đổi mật khẩu: Thay đổi mật khẩu hiện tại.
- Nhận thông báo (qua mail): Nhận các thông báo về hoạt động, tin tức từ CLB qua email.

+ Quản trị viên: Là những người quản lý hệ thống và có quyền hạn cao nhất. Có thể thực hiện các chức năng như:

- Quản lý thông tin thành viên: Xem, sửa, hoặc xóa thông tin của các thành viên.
- Gửi thông báo: Gửi các thông báo quan trọng đến tất cả thành viên.
- Quản lý hoạt động: Tạo, chỉnh sửa hoặc xóa các hoạt động của CLB.

- Quản lý chức vụ: Phân quyền, thay đổi hoặc hủy bỏ chức vụ của thành viên.
- Quản lý tin tức: Tạo, chỉnh sửa hoặc xóa bài viết.
- Xem thống kê: Xem báo cáo, thống kê về hoạt động, thành viên và các chỉ số khác.
- Quản lý điểm danh: Theo dõi và quản lý điểm danh của thành viên trong các hoạt động.

- Mô tả các mối quan hệ

+ Quan hệ "Include": Các use case "Đăng nhập" liên kết với nhiều chức năng khác như:

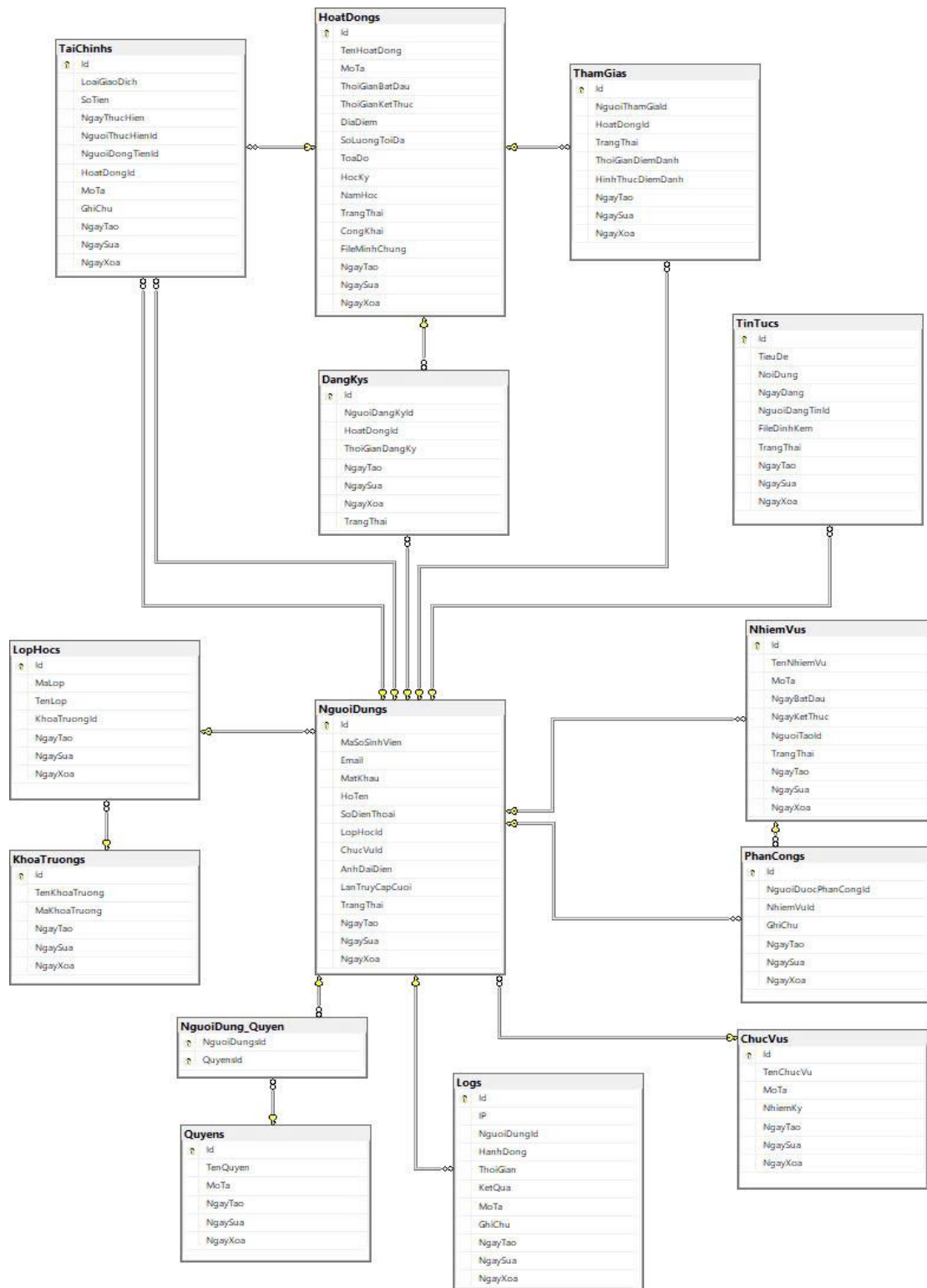
- Cập nhật thông tin cá nhân
- Đăng ký tham gia hoạt động
- Xem thông tin cá nhân
- Xem minh chứng tham gia

+ Các chức năng đặc thù:

- Chỉ quản trị viên mới có thể sử dụng các chức năng quản trị (Quản lý tin tức, thông tin thành viên, hoạt động,...).
- Thành viên có quyền riêng biệt, như xem minh chứng tham gia hoặc tự điểm điểm danh.

3.6 Thiết kế cơ sở dữ liệu:

3.6.1 Lược đồ CSDL:



Hình 2: Lược đồ CSDL

CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

3.7 Kết quả đạt được

3.7.1 Các API đã xây dựng

- API quản lý chức vụ

ChucVu			^
GET	/api/ChucVu/{id}	Lấy thông tin chi tiết của một chức vụ	✓ 🔒
PUT	/api/ChucVu/{id}	Cập nhật thông tin của một chức vụ	✓ 🔒
DELETE	/api/ChucVu/{id}	Xóa một chức vụ	✓ 🔒
GET	/api/ChucVu	Lấy danh sách chức vụ có phân trang	✓ 🔒
POST	/api/ChucVu	Tạo mới một chức vụ	✓ 🔒
GET	/api/ChucVu/nhiem-ky/{nhiemKy}	Lấy danh sách chức vụ theo nhiệm kỳ	✓ 🔒

Hình 3: API quản lý chức vụ

- API quản lý hoạt động

HoatDong			^
POST	/api/HoatDong/update-status	Cập nhật trạng thái hoạt động theo thời gian thực	✓ 🔒
GET	/api/HoatDong	Lấy danh sách hoạt động có phân trang và tìm kiếm	✓ 🔒
POST	/api/HoatDong	Tạo mới hoạt động	✓ 🔒
GET	/api/HoatDong/{id}	Lấy thông tin chi tiết của một hoạt động	✓ 🔒
PUT	/api/HoatDong/{id}	Cập nhật thông tin hoạt động	✓ 🔒
DELETE	/api/HoatDong/{id}	Xóa hoạt động	✓ 🔒
GET	/api/HoatDong/hoc-ky-nam-hoc	Lấy danh sách hoạt động theo học kỳ và năm học	✓ 🔒
GET	/api/HoatDong/sap-dien-ra	Lấy danh sách hoạt động sắp diễn ra	✓ 🔒
GET	/api/HoatDong/trang-thai/{trangThai}	Lấy danh sách hoạt động theo trạng thái	✓ 🔒
PUT	/api/HoatDong/{id}/trang-thai	Cập nhật trạng thái hoạt động	✓ 🔒
PUT	/api/HoatDong/{id}/minh-chung	Cập nhật file minh chứng cho hoạt động	✓ 🔒
GET	/api/HoatDong/export	Xuất danh sách hoạt động ra file Excel	✓ 🔒

Hình 4: API quản lý hoạt động

- API quản lý nhiệm vụ

NhiemVu			^
GET	/api/NhiemVu/{id}	Lấy thông tin chi tiết của một nhiệm vụ	✓ 🔒
PUT	/api/NhiemVu/{id}	Cập nhật thông tin của một nhiệm vụ	✓ 🔒
DELETE	/api/NhiemVu/{id}	Xóa một nhiệm vụ	✓ 🔒
GET	/api/NhiemVu	Lấy danh sách nhiệm vụ có phân trang	✓ 🔒
POST	/api/NhiemVu	Tạo mới một nhiệm vụ	✓ 🔒
GET	/api/NhiemVu/nguoi-tao/{nguoiTaoId}	Lấy danh sách nhiệm vụ theo người tạo	✓ 🔒
GET	/api/NhiemVu/trang-thai/{trangThai}	Lấy danh sách nhiệm vụ theo trạng thái	✓ 🔒
PUT	/api/NhiemVu/{id}/trang-thai	Cập nhật trạng thái của nhiệm vụ	✓ 🔒

Hình 5: API quản lý nhiệm vụ

- API quản lý tài chính

TaiChinh			^
GET	/api/TaiChinh/{id}	Lấy thông tin chi tiết của một giao dịch tài chính	✓ 🔒
PUT	/api/TaiChinh/{id}	Cập nhật thông tin giao dịch tài chính	✓ 🔒
DELETE	/api/TaiChinh/{id}	Xóa giao dịch tài chính	✓ 🔒
GET	/api/TaiChinh	Lấy danh sách giao dịch tài chính có phân trang	✓ 🔒
POST	/api/TaiChinh	Tạo mới giao dịch tài chính	✓ 🔒
GET	/api/TaiChinh/nguoi-thuc-hien	Lấy danh sách giao dịch của người thực hiện	✓ 🔒
GET	/api/TaiChinh/nguoi-dong-tien	Lấy danh sách giao dịch của người đóng tiền	✓ 🔒
GET	/api/TaiChinh/hoat-dong/{hoatDongId}	Lấy danh sách giao dịch của một hoạt động	✓ 🔒
GET	/api/TaiChinh/loai-giao-dich/{loaiGiaoDich}	Lấy danh sách giao dịch theo loại	✓ 🔒
GET	/api/TaiChinh/thong-ke	Lấy thống kê tài chính trong khoảng thời gian	✓ 🔒

Hình 6: API quản lý tài chính

- API quản lý phân công

PhanCong			^
GET	/api/PhanCong/{id}	Lấy thông tin chi tiết của một phân công	✓ 🔒
PUT	/api/PhanCong/{id}	Cập nhật thông tin phân công	✓ 🔒
DELETE	/api/PhanCong/{id}	Xóa một phân công	✓ 🔒
GET	/api/PhanCong	Lấy danh sách phân công có phân trang	✓ 🔒
POST	/api/PhanCong	Tạo mới một phân công	✓ 🔒
GET	/api/PhanCong/nguoi-duoc-phan-cong	Lấy danh sách phân công của một người dùng	✓ 🔒
GET	/api/PhanCong/nhiem-vu/{nhiemVuId}	Lấy danh sách phân công của một nhiệm vụ	✓ 🔒
POST	/api/PhanCong/phan-cong-nhom	Phân công nhiệm vụ cho một nhóm người dùng	✓ 🔒

Hình 7: API quản lý phân công

- API quản lý tham gia

ThamGia			^
GET	/api/ThamGia/{id}	Lấy thông tin chi tiết của một lượt tham gia	✓ 🔒
DELETE	/api/ThamGia/{id}		✓ 🔒
GET	/api/ThamGia	Lấy danh sách tham gia có phân trang	✓ 🔒
GET	/api/ThamGia/nguoi-tham-gia	Lấy danh sách tham gia của người dùng hiện tại	✓ 🔒
GET	/api/ThamGia/hoat-dong/{hoatDongId}	Lấy danh sách tham gia của một hoạt động	✓ 🔒
GET	/api/ThamGia/hoat-dong/{hoatDongId}/trang-thai/{trangThai}		✓ 🔒
POST	/api/ThamGia/diem-danh	Điểm danh tham gia hoạt động	✓ 🔒
POST	/api/ThamGia/diem-danh-nhom	Điểm danh nhóm tham gia hoạt động	✓ 🔒
POST	/api/ThamGia/diem-danh-gps	Điểm danh tham gia hoạt động bằng GPS	✓ 🔒
POST	/api/ThamGia/kiem-tra-tham-gia	Kiểm tra trạng thái tham gia hoạt động	✓ 🔒
PUT	/api/ThamGia/{id}/trang-thai	Cập nhật trạng thái tham gia	✓ 🔒

Hình 8: API quản lý tham gia

Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên

- API quản lý quyền

Quyền			^
GET	/api/Quyền/{id}	Lấy thông tin chi tiết của một quyền	✓ 🔒
PUT	/api/Quyền/{id}	Cập nhật thông tin của một quyền	✓ 🔒
DELETE	/api/Quyền/{id}	Xóa một quyền	✓ 🔒
GET	/api/Quyền	Lấy danh sách quyền có phân trang	✓ 🔒
POST	/api/Quyền	Tạo mới một quyền	✓ 🔒
GET	/api/Quyền/người-dùng	Lấy danh sách quyền của một người dùng	✓ 🔒
POST	/api/Quyền/phan-quyền	Phân quyền cho người dùng	✓ 🔒
DELETE	/api/Quyền/người-dùng/quyền	Xóa quyền của người dùng	✓ 🔒

Hình 9: API quản lý quyền

- API quản lý tin tức

TinTuc			^
GET	/api/TinTuc/{id}	Lấy thông tin chi tiết của một tin tức	✓ 🔒
PUT	/api/TinTuc/{id}	Cập nhật thông tin tin tức	✓ 🔒
DELETE	/api/TinTuc/{id}	Xóa tin tức	✓ 🔒
GET	/api/TinTuc	Lấy danh sách tin tức có phân trang	✓ 🔒
POST	/api/TinTuc	Tạo mới tin tức	✓ 🔒
GET	/api/TinTuc/người-dang-tin	Lấy danh sách tin tức của người đăng tin	✓ 🔒
GET	/api/TinTuc/da-xuat-ban	Lấy danh sách tin tức đã xuất bản	✓ 🔒
PUT	/api/TinTuc/{id}/trang-thai	Cập nhật trạng thái xuất bản của tin tức	✓ 🔒

Hình 10: API quản lý tin tức

- API quản lý lớp học

LopHoc			^
GET	/api/LopHoc/{id}	Lấy thông tin chi tiết của một lớp học	✓ 🔒
PUT	/api/LopHoc/{id}	Cập nhật thông tin của một lớp học	✓ 🔒
DELETE	/api/LopHoc/{id}	Xóa một lớp học	✓ 🔒
GET	/api/LopHoc	Lấy danh sách lớp học có phân trang	✓ 🔒
POST	/api/LopHoc	Tạo mới một lớp học	✓ 🔒
GET	/api/LopHoc/khoa-truong/{khoaTruongId}	Lấy danh sách lớp học theo khoa/trường	✓ 🔒

Hình 11: API quản lý lớp học

- API quản lý khoa trường

Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên

KhoaTruong			^
GET	/api/KhoaTruong/{id}	Lấy thông tin chi tiết của một khoa/trường	✓ 🔒
PUT	/api/KhoaTruong/{id}	Cập nhật thông tin khoa/trường	✓ 🔒
DELETE	/api/KhoaTruong/{id}	Xóa khoa/trường	✓ 🔒
GET	/api/KhoaTruong	Lấy danh sách khoa/trường có phân trang	✓ 🔒
POST	/api/KhoaTruong	Tạo mới khoa/trường	✓ 🔒
POST	/api/KhoaTruong/seed	Khởi tạo dữ liệu mẫu cho khoa/trường	✓ 🔒

Hình 12: API quản lý khoa trường

- API quản lý đăng ký

DangKy			^
GET	/api/DangKy/{id}	Lấy thông tin chi tiết của một đăng ký	✓ 🔒
DELETE	/api/DangKy/{id}	Hủy một đăng ký	✓ 🔒
GET	/api/DangKy	Lấy danh sách đăng ký có phân trang	✓ 🔒
POST	/api/DangKy	Tạo mới một đăng ký	✓ 🔒
GET	/api/DangKy/nguoi-dang-ky	Lấy danh sách đăng ký của người dùng hiện tại	✓ 🔒
GET	/api/DangKy/hoat-dong/{hoatDongId}	Lấy danh sách đăng ký của một hoạt động	✓ 🔒

Hình 13: API quản lý đăng ký

- API quản lý thống kê

ThongKe			^
GET	/api/ThongKe	Lấy thông tin thống kê tổng quan	✓ 🔒

Hình 14: API quản lý thông kê

- API quản lý người dùng

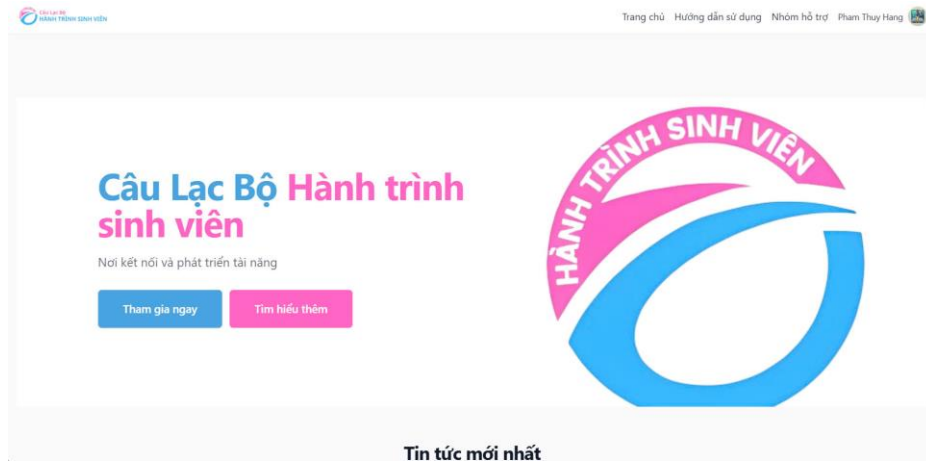
NguoiDung			^
GET	/api/NguoiDung/ban-chu-nhiem	GetBanChuNhiem là những người dùng có mã chức vụ là 4, 5, 6, 7	✓ 🔒
GET	/api/NguoiDung/{id}	Lấy thông tin chi tiết của một người dùng	✓ 🔒
PUT	/api/NguoiDung/{id}	Cập nhật thông tin của một người dùng	✓ 🔒
DELETE	/api/NguoiDung/{id}	Xóa một người dùng	✓ 🔒
GET	/api/NguoiDung	Lấy danh sách người dùng có phân trang	✓ 🔒
POST	/api/NguoiDung	Tạo mới một người dùng	✓ 🔒
GET	/api/NguoiDung/lop-hoc/{lopHocId}	Lấy danh sách người dùng theo lớp học	✓ 🔒
GET	/api/NguoiDung/chuc-vu/{chucVuId}	Lấy danh sách người dùng theo chức vụ	✓ 🔒
PUT	/api/NguoiDung/{id}/mat-khau	Cập nhật mật khẩu của người dùng	✓ 🔒
PUT	/api/NguoiDung/{id}/trang-thai	Cập nhật trạng thái của người dùng	✓ 🔒
POST	/api/NguoiDung/login	Đăng nhập vào hệ thống	✓ 🔒
POST	/api/NguoiDung/register	Đăng ký tài khoản mới	✓ 🔒

Hình 15: API quản lý người dùng

3.7.2 Giao diện

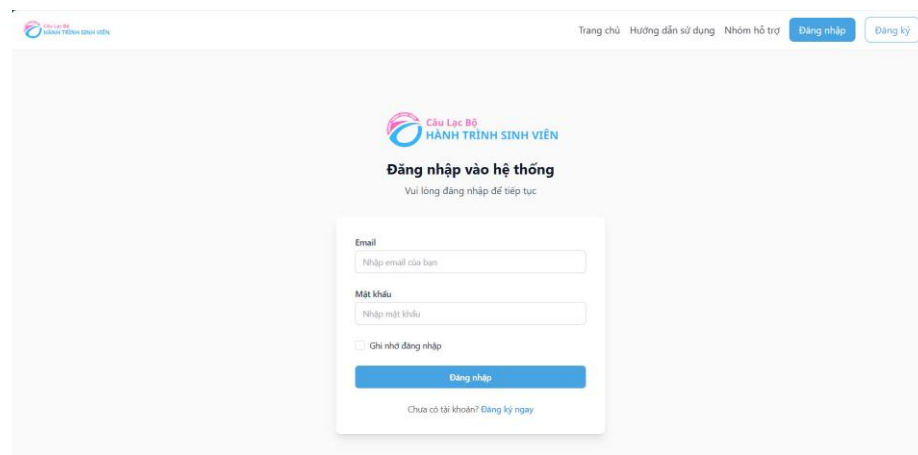
Giao diện người dùng:

- Trang chủ



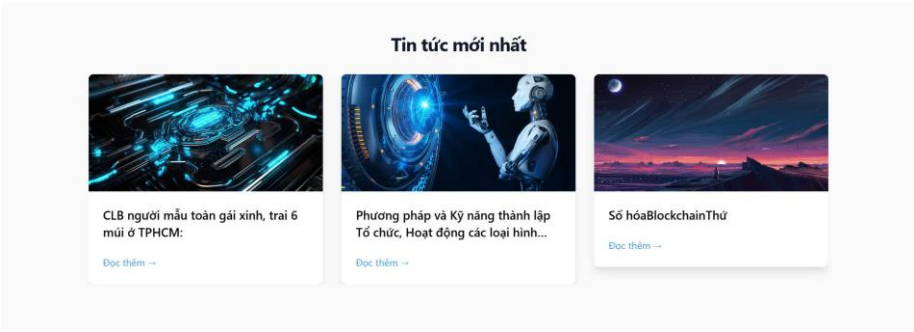
Hình 16: Trang chủ

- Đăng nhập



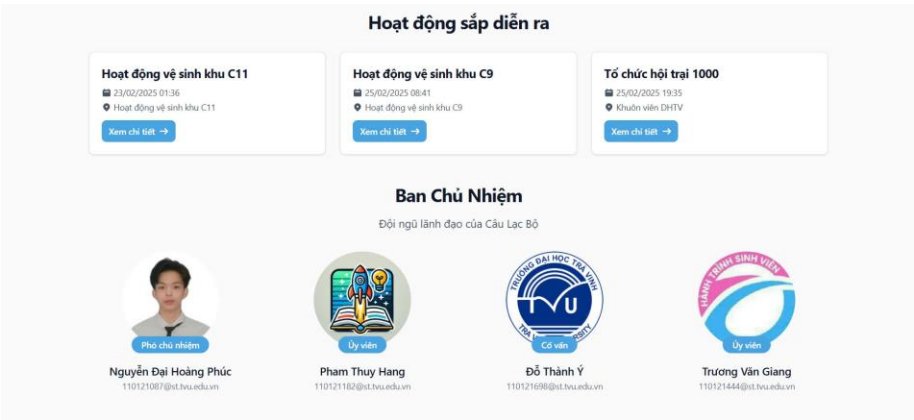
Hình 17: Đăng nhập

+ Section tin tức:



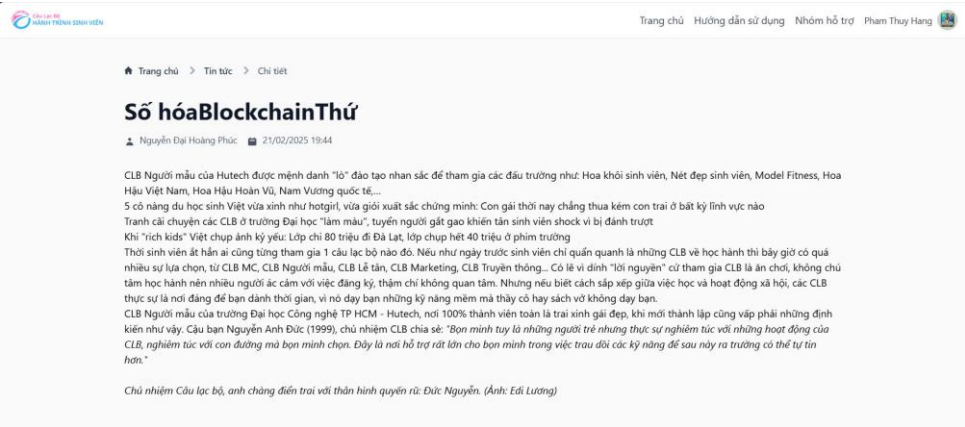
Hình 18: Section tin tức

+ Section các hoạt động sắp diễn ra và giới thiệu Ban chủ nhiệm:



Hình 19: Section các hoạt động và giới thiệu Ban chủ nhiệm

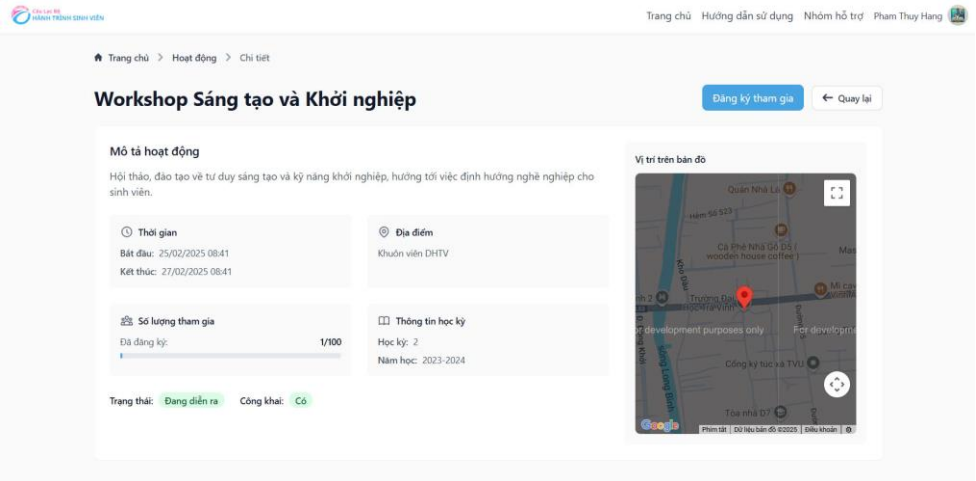
- Trang chi tiết tin tức



Hình 20: Trang chi tiết tin tức

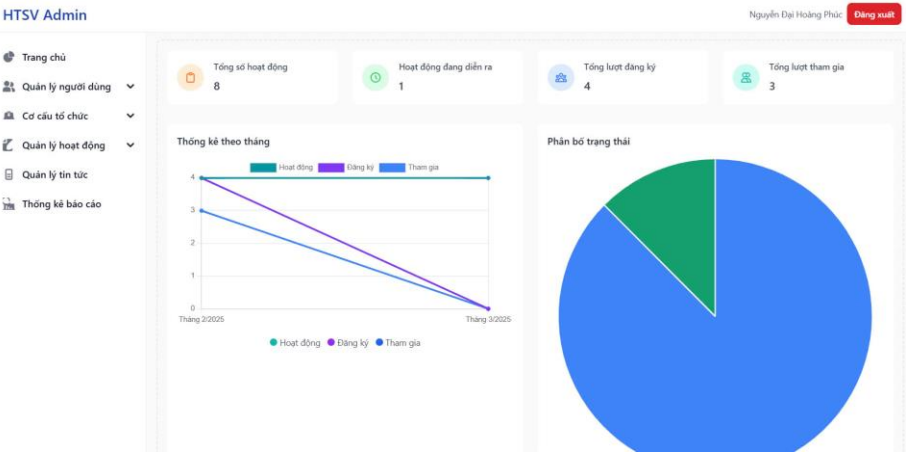
- Trang đăng ký hoạt động

Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên



Hình 21: Trang đăng ký hoạt động

- Trang Tổng quan Admin



Hình 22: Trang Tổng quan Admin

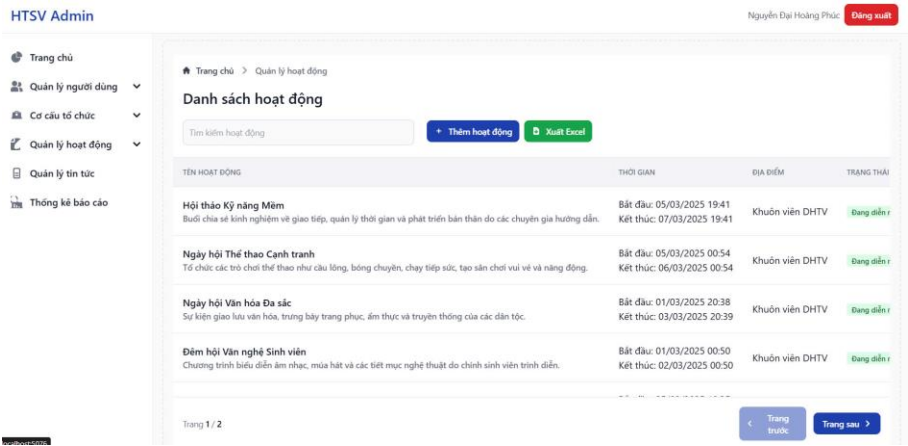
- Trang quản lý người dùng

The screenshot shows the HTSV Admin user management page. It displays a table of users with columns for name, ID, email, role, status, and actions. The table lists four users: Nguyễn Đại Hoàng Phúc, Phạm Thụy Hằng, Đỗ Thành Ý, and Trương Văn Giang.

ẢNH ĐẠI DIỆN	HỌ TÊN	MSV	EMAIL	LỚP	CHỨC VỤ	TRẠNG THÁI	THAO TÁC
	Nguyễn Đại Hoàng Phúc	110121087	110121087@st.bvu.edu.vn	Công nghệ thông tin K21B	Phó chủ nhiệm	Hoạt động	Sửa Đổi mật khẩu
	Phạm Thụy Hằng	110121182	110121182@st.bvu.edu.vn	Công nghệ thông tin K21B	Ủy viên	Hoạt động	Sửa Đổi mật khẩu
	Đỗ Thành Ý	110121698	110121698@st.bvu.edu.vn	Công nghệ thông tin K21A	Cố vấn	Hoạt động	Sửa Đổi mật khẩu
	Trương Văn Giang	110121444	110121444@st.bvu.edu.vn	Công nghệ thông tin K21A	Ủy viên	Hoạt động	Sửa Đổi mật khẩu

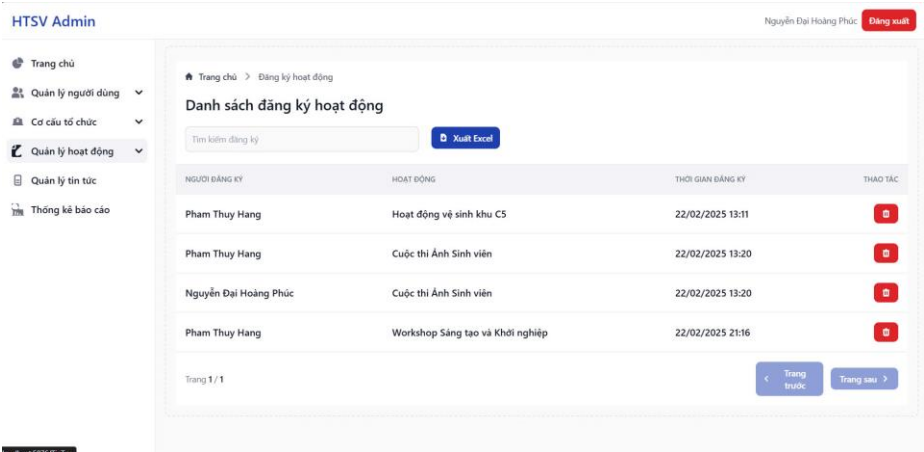
Hình 23: Trang quản lý người dùng

- Trang quản lý hoạt động



Hình 24: Trang quản lý hoạt động

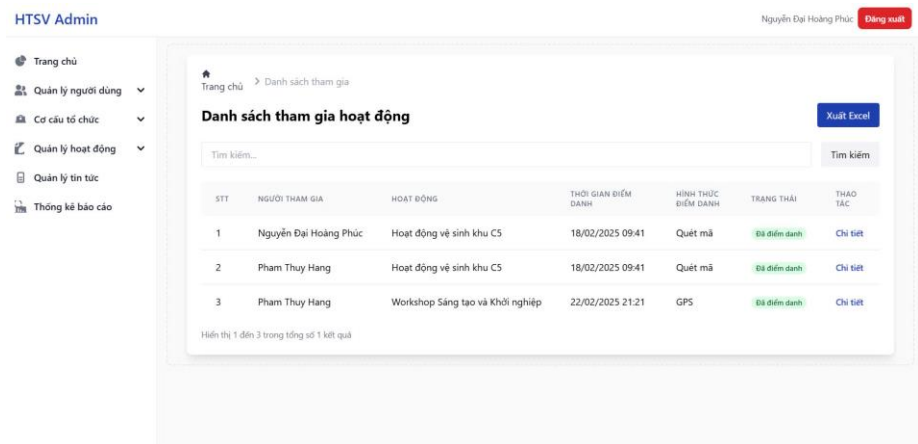
- Trang xem danh sách đăng ký hoạt động



Hình 25: Trang xem danh sách đăng ký hoạt động

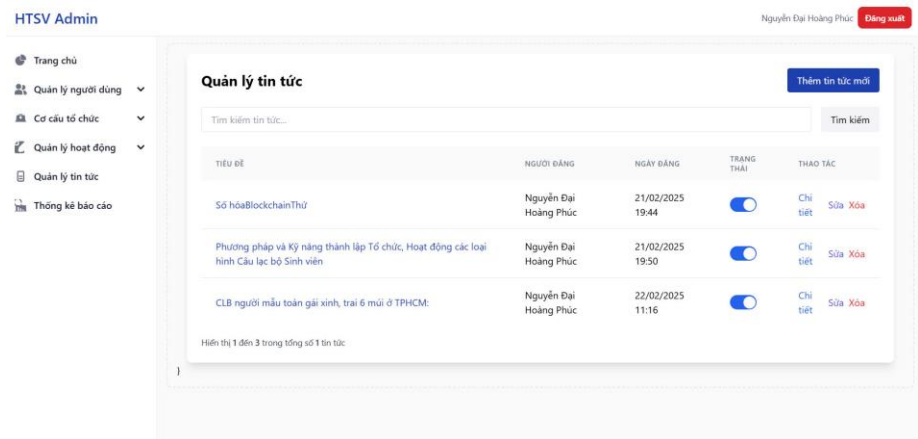
- Trang xem danh sách tham gia hoạt động

Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên



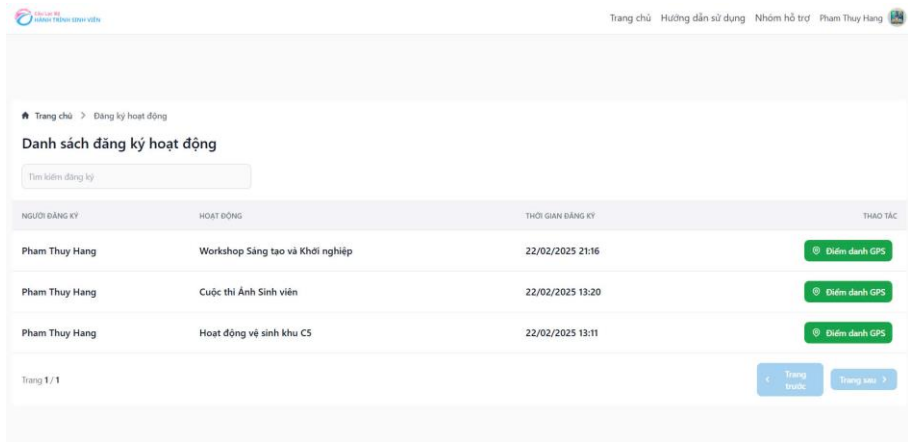
Hình 26: Trang xem danh sách tham gia hoạt động

- Trang quản lý tin tức



Hình 27: Trang quản lý tin tức

- Trang danh sách đăng ký



Hình 28: Trang danh sách đăng ký

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết quả đạt được:

- Phân tích và hiểu được nghiệp vụ về công tác quản lý câu lạc bộ.
- Xây dựng được một cơ sở dữ liệu quản lý câu lạc bộ Hành trình sinh viên.
- Xây dựng được một trang web hỗ trợ công tác quản lý cho câu lạc bộ Hành trình sinh viên.

4.2 Hạn chế:

- Quy mô áp dụng của đề tài còn hạn chế, chỉ áp dụng cho một câu lạc bộ nên dữ liệu còn chưa đầy đủ và hoàn chỉnh.
- Việc xử lý và tối ưu hóa trang web để nó hoạt động với tốc độ phản hồi nhanh hơn còn chưa tốt.

4.3 Hướng phát triển:

- Tối ưu hóa hiệu suất của trang web.
- Phát triển thêm các tính năng cần thiết để mở rộng quy mô của trang web để có thể áp dụng đối với nhiều câu lạc bộ.

PHẦN II: MINI CHAT SOCKET

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu tổng quan vấn đề :

Trong thời đại công nghệ số phát triển mạnh mẽ, việc giao tiếp trực tuyến ngày càng trở thành nhu cầu thiết yếu của xã hội. Các ứng dụng chat trực tuyến đóng vai trò quan trọng trong việc kết nối con người, không chỉ phục vụ nhu cầu cá nhân mà còn hỗ trợ công việc, học tập, và kinh doanh.

Với sự phát triển của công nghệ thời gian thực, việc áp dụng một hệ thống chat hiện đại là cần thiết để đáp ứng nhu cầu giao tiếp linh hoạt, tiết kiệm thời gian và nâng cao trải nghiệm người dùng. Mini Chat Socket ra đời nhằm khắc phục các hạn chế của giao tiếp truyền thống, cung cấp một giải pháp chat trực tuyến thời gian thực, đơn giản nhưng hiệu quả, giúp người dùng trao đổi thông tin nhanh chóng và thuận tiện hơn.

1.2 Mục tiêu nghiên cứu:

- Xây dựng hệ thống chat thời gian thực: Phát triển một ứng dụng chat trực tuyến có khả năng gửi và nhận tin nhắn tức thời giữa các người dùng, đảm bảo tốc độ truyền tải nhanh và ổn định.
- Áp dụng công nghệ WebSocket: Nghiên cứu và triển khai giao thức WebSocket kết hợp với thư viện Socket.IO để tạo ra kênh giao tiếp hai chiều giữa máy khách và máy chủ một cách hiệu quả.
- Thiết kế giao diện thân thiện và dễ sử dụng: Xây dựng một giao diện người dùng đơn giản, trực quan, đáp ứng các thao tác cơ bản như gửi tin nhắn, hiển thị danh sách người dùng đang trực tuyến.
- Khả năng mở rộng và tùy chỉnh: Tạo ra một hệ thống có cấu trúc linh hoạt, dễ dàng mở rộng thêm tính năng hoặc tích hợp vào các hệ thống lớn hơn trong tương lai.

- Giảm tải và tối ưu tài nguyên hệ thống: Tìm hiểu cách tối ưu hiệu năng khi có nhiều người dùng đồng thời, đảm bảo hệ thống vận hành trơn tru ngay cả khi lưu lượng truy cập tăng cao.
- Phát triển kỹ năng lập trình thời gian thực: Cung cấp cơ hội học tập và rèn luyện cho người thực hiện về các công nghệ hiện đại như WebSocket, Node.js, và các nguyên lý lập trình phía máy chủ và máy khách.

1.3 Phương pháp nghiên cứu:

- Khảo sát thực tế: Tìm hiểu nhu cầu giao tiếp trực tuyến thời gian thực thông qua các ứng dụng chat phổ biến.
- Nghiên cứu tài liệu: Tìm hiểu về Socket, Socket.IO và các công nghệ liên quan.
- Phân tích yêu cầu: Xác định các tính năng cần thiết như gửi/nhận tin nhắn, quản lý người dùng trực tuyến và tối ưu hiệu suất.

1.4 Đối tượng và phạm vi nghiên cứu:

1.4.1 Đối tượng nghiên cứu:

- Người dùng cuối: Các cá nhân hoặc nhóm cần một công cụ giao tiếp trực tuyến đơn giản và hiệu quả, chẳng hạn như sinh viên, nhân viên văn phòng, hoặc nhóm làm việc nhỏ.
- Công nghệ sử dụng: Các phương pháp truyền tải dữ liệu thời gian thực như AJAX long-polling, Server-Sent Events (SSE).
- Tương tác hệ thống: Mối quan hệ giữa máy khách và máy chủ trong việc xử lý và truyền tải tin nhắn thời gian thực.
- Tính năng cần thiết: Các chức năng cơ bản như gửi/nhận tin nhắn, hiển thị trạng thái trực tuyến và quản lý người dùng.

1.4.2 Phạm vi nghiên cứu:

- Công nghệ: WebSocket, Socket.IO, Node.js, JWT, MongoDB/SQL Server.

- Đối tượng: Nhóm làm việc nhỏ, sinh viên, nhân viên văn phòng.
- Chức năng: Nhắn tin thời gian thực, quản lý người dùng trực tuyến, bảo mật dữ liệu.
- Giới hạn: Chưa hỗ trợ gửi file, video, chat nhóm, gọi thoại/video.
- Mở rộng: Hỗ trợ gửi đa phương tiện, AI chatbot, tích hợp hệ thống khác.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Cơ sở lý thuyết

2.1.1 Giới thiệu về công nghệ:

Mini Chat Socket là một hệ thống chat thời gian thực sử dụng giao thức WebSocket, giúp người dùng giao tiếp nhanh chóng và linh hoạt. Hệ thống này được phát triển độc lập nhằm hỗ trợ việc trao đổi thông tin một cách hiệu quả.

WebSocket là một giao thức truyền thông hai chiều, giúp duy trì kết nối liên tục giữa máy khách và máy chủ, khác với giao thức HTTP thông thường chỉ hỗ trợ giao tiếp một chiều. WebSocket giảm thiểu độ trễ và giảm tải tài nguyên hệ thống so với kỹ thuật HTTP polling truyền thống.

2.1.2 Nguyên lý hoạt động:

- **WebSocket:** Mini Chat Socket sử dụng giao thức WebSocket để thiết lập kênh kết nối hai chiều giữa máy chủ và máy khách, giúp giảm độ trễ so với các phương pháp truyền thống như AJAX polling.
- **Real-time Communication:** Theo nghiên cứu của Pimentel & Nickerson (2012), hệ thống chat thời gian thực giúp cải thiện tốc độ phản hồi và tăng hiệu quả giao tiếp trong môi trường làm việc nhóm.
- **Event-driven Model:** Hệ thống lắng nghe và phản hồi ngay khi có tin nhắn mới mà không yêu cầu tải lại trang, tối ưu hiệu suất.
- **Node.js & Socket.IO:** Sử dụng Node.js giúp tăng hiệu suất xử lý đồng thời, trong khi thư viện Socket.IO hỗ trợ WebSocket dễ dàng hơn.

2.2 Cơ sở lý luận và giả thiết khoa học

2.2.1 Cơ sở lý luận:

- Tương tác nhanh chóng: hệ thống chat thời gian thực giúp cải thiện sự phối hợp trong nhóm làm việc.

- Tăng cường gắn kết: việc tích hợp hệ thống chat giúp tăng cường khả năng hợp tác giữa các thành viên.
- Bảo mật: Việc mã hóa dữ liệu bằng JWT (JSON Web Token) và SSL (Secure Sockets Layer) giúp bảo vệ thông tin người dùng khỏi các cuộc tấn công trung gian (Man-in-the-Middle Attacks).

2.2.2 Giả thiết khoa học:

- Hiệu suất cao: Giao thức WebSocket giúp truyền dữ liệu nhanh hơn so với HTTP polling.
- Mô hình publish-subscribe: Hỗ trợ việc gửi tin nhắn đối với nhóm người dùng.
- Tính năng quản lý: Hỗ trợ lưu trữ tin nhắn, hiển thị danh sách online.

2.3 Kiến trúc hệ thống

2.3.1 Kiến trúc Server:

- + Node.js & Express.js: Xây dựng server nhẹ, tối ưu hiệu năng.
- + Socket.IO: Xử lý kết nối và truyền tin nhắn theo thời gian thực.
- + JWT Authentication: Xác thực và bảo mật thông tin người dùng.
- + Database (MongoDB/SQL Server): Lưu trữ tin nhắn, danh sách người dùng.

2.3.2 Kiến trúc Client:

- React/Vue.js: Xây dựng giao diện trực quan.
- Socket.IO-client: Kết nối với server qua WebSocket.
- UI/UX Friendly: Thiết kế gọn gàng, hiển thị danh sách tin nhắn, người dùng online.

2.4 Các Chức Năng Chính

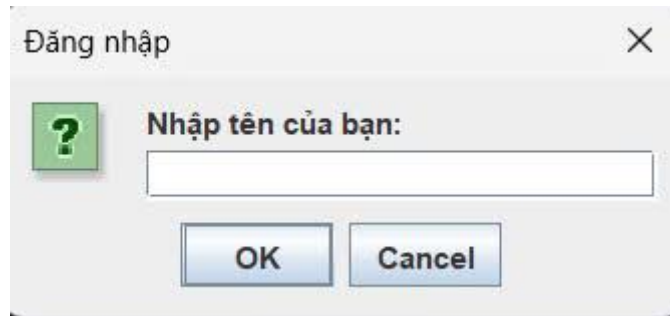
- Gửi và nhận tin nhắn thời gian thực: Hỗ trợ trao đổi tin nhắn nhanh chóng giữa người dùng.

- Quản lý danh sách người dùng online: Hiển thị trạng thái trực tuyến của từng thành viên.
- Hệ thống hoạt động độc lập: Không phụ thuộc vào bất kỳ nền tảng nào khác.
- Bảo mật dữ liệu: Sử dụng mã hóa JWT và SSL để đảm bảo an toàn thông tin.

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Giao diện mini chat

❖ Đăng nhập:



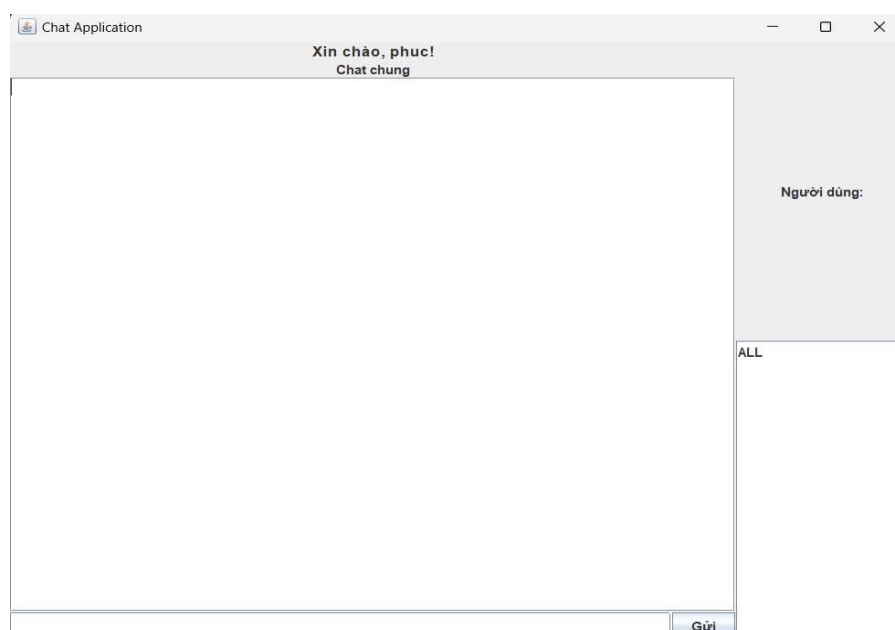
Hình 29: Đăng nhập

- Khi mở ứng dụng, người dùng sẽ được yêu cầu nhập tên đăng nhập thông qua một cửa sổ nhỏ với tiêu đề "Đăng nhập".

Cửa sổ này có một ô nhập liệu với nhãn "Nhập tên của bạn:", nơi người dùng điền tên.

Hai nút "OK" và "Cancel" cho phép xác nhận hoặc hủy đăng nhập.

❖ Giao diện người dùng chung:



Hình 30: Giao diện người dùng chung

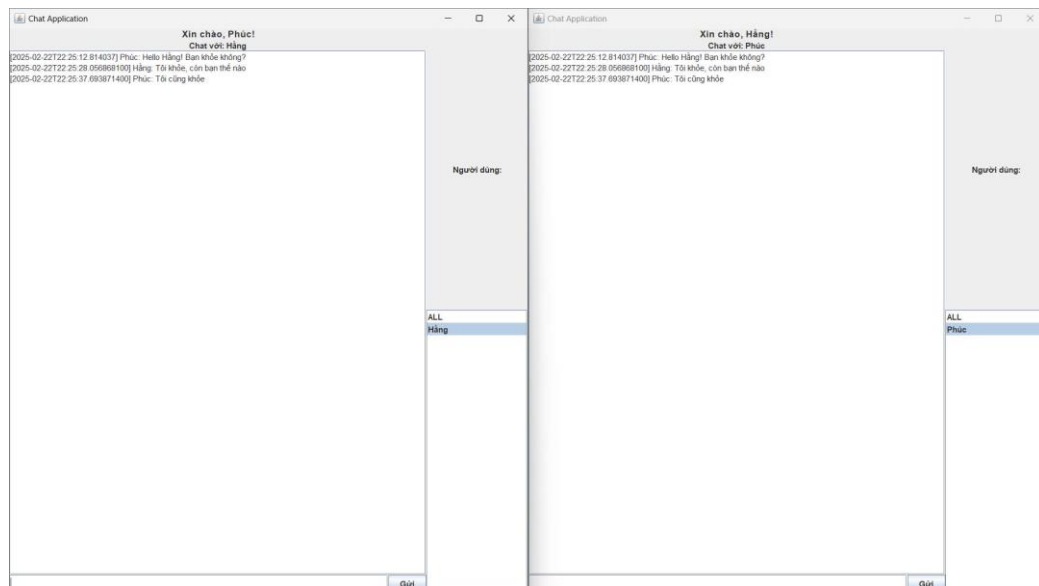
Xây dựng website quản lý câu lạc bộ Hành Trình Sinh Viên

- Sau khi đăng nhập, người dùng được chào đón bằng thông điệp "Xin chào, [Tên người dùng]!" ngay phía trên khung chat.

Giao diện chính gồm các phần:

- Khung chat lớn ở giữa để hiển thị tin nhắn.
- Danh sách người dùng bên phải, hiển thị tất cả những ai đang tham gia chat.
- Ô nhập tin nhắn ở phía dưới cùng, nơi người dùng nhập nội dung chat.
- Nút "Gửi" để gửi tin nhắn

❖ Giao diện người dùng cá nhân:



Hình 31: Giao diện người dùng cá nhân

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 Kết quả đạt được:

- Xây dựng hệ thống chat thời gian thực với khả năng gửi/nhận tin nhắn nhanh chóng.

Sử dụng WebSocket và Socket.IO, đảm bảo tốc độ phản hồi nhanh, kết nối ổn định.

Giao diện đơn giản, dễ sử dụng, hỗ trợ các chức năng cơ bản.

Bảo mật dữ liệu bằng JWT, giúp xác thực và mã hóa thông tin cá nhân.

Tối ưu hiệu suất với Node.js, xử lý đồng thời nhiều kết nối mà không ảnh hưởng hiệu năng

4.2 Hạn chế:

- Giao diện chưa đa dạng: Hiện tại, giao diện còn đơn giản và chưa có nhiều tùy chọn cá nhân hóa.

- Chưa hỗ trợ đa phương tiện: Hệ thống chỉ hỗ trợ gửi tin nhắn văn bản, chưa có chức năng gửi hình ảnh, video hoặc file đính kèm.

- Bảo mật cần cải thiện: Cần nâng cấp hệ thống bảo mật để chống lại các cuộc tấn công như XSS (Cross-Site Scripting) hoặc CSRF (Cross-Site Request Forgery).

- Hiệu suất khi có nhiều người dùng: Khi số lượng người dùng tăng cao, hệ thống có thể gặp tình trạng tải chậm hoặc mất kết nối.

4.3 Hướng phát triển:

- Cải thiện giao diện người dùng: Tích hợp các theme, chế độ tối (dark mode), và khả năng tùy chỉnh giao diện theo sở thích cá nhân.

- Hỗ trợ gửi file và đa phương tiện: Bổ sung tính năng gửi hình ảnh, video, emoji và tài liệu.

- Nâng cao bảo mật: Tích hợp các phương pháp mã hóa nâng cao như AES hoặc TLS để bảo vệ dữ liệu người dùng.
- Mở rộng tính năng nhóm chat: Hỗ trợ tạo nhóm chat riêng tư, quản lý nhóm và gửi thông báo hệ thống.
- Tích hợp AI chatbot: Sử dụng AI để hỗ trợ người dùng trong việc giải đáp câu hỏi hoặc tự động phản hồi tin nhắn.
- Cải thiện hiệu suất: Tối ưu hệ thống xử lý đồng thời, giảm độ trễ khi có nhiều người dùng truy cập.

DANH MỤC TÀI LIỆU THAM KHẢO

1. Esposito, D. Lập trình Microsoft ASP.NET MVC. Redmond, Washington: Microsoft Press, 2011.
 2. Phạm Minh Dương. Phân tích và thiết kế hệ thống thông tin. Tài liệu giảng dạy, Trường Đại học Trà Vinh, 2014.
 3. Phan Thị Phương Nam. Hệ quản trị cơ sở dữ liệu. Tài liệu giảng dạy, Trường Đại học Trà Vinh, 2015.
 4. Freeman, A. (2022). Pro ASP.NET Core 6: Develop Cloud-Ready Web Applications Using MVC, Blazor, and Razor Pages. Apress, 9th edition. DOI: <https://doi.org/10.1007/978-1-4842-7957-1>.
 5. Istifarulah, M. H. R., & Tiaharyadini, R. (2023). Các phương pháp DevOps, Continuous Integration và Continuous Deployment cho tự động hóa triển khai phần mềm. JISA (Jurnal Informatika dan Sains), 6(2), 116. DOI: <https://doi.org/10.31326/jisa.v6i2.1751>.
-