

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐO ÁN CHUYÊN NGÀNH
HỌC KỲ I, NĂM HỌC 2025-2026

**XÂY DỰNG HỆ THỐNG QUẢN LÝ
VÀ HỖ TRỢ TÌNH NGUYỆN VIÊN
CHO TỔ CHỨC PHI LỢI NHUẬN**

Giảng viên hướng dẫn:
ThS. GVC. Phan Thị Phương Nam

Sinh viên thực hiện:
Họ tên: Nguyễn Phúc Vinh
MSSV: 110122205
Lớp: DA22TTC

Vĩnh Long, tháng 12 năm 2025

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐO ÁN CHUYÊN NGÀNH
HỌC KỲ I, NĂM HỌC 2025-2026

XÂY DỰNG HỆ THỐNG QUẢN LÝ
VÀ HỖ TRỢ TÌNH NGUYỆN VIÊN
CHO TỔ CHỨC PHI LỢI NHUẬN

Giảng viên hướng dẫn:
ThS. GVC. Phan Thị Phương Nam

Sinh viên thực hiện:
Họ tên: Nguyễn Phúc Vinh
MSSV: 110122205
Lớp: DA22TTC

Vĩnh Long, tháng 12 năm 2025

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

Vĩnh Long, ngày tháng năm
Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

Vĩnh Long, ngày tháng năm

Thành viên hội đồng

LỜI CẢM ƠN

Em muốn cảm ơn sâu sắc cho sự giúp đỡ và hướng dẫn, mà em đã nhận được từ giảng viên hướng dẫn trong quá trình hoàn thành đồ án chuyên ngành trong lĩnh vực Công nghệ Thông tin.

Thời gian qua, em đã học được nhiều kinh nghiệm quý giá và kiến thức từ giảng viên hướng dẫn và em rất biết ơn sự hỗ trợ và động viên mà giảng viên hướng dẫn đã cung cấp cho em trong suốt quá trình nghiên cứu.

Thông qua báo cáo này, em xin gửi lời cảm ơn đến ThS. Phan Thị Phương Nam - giảng viên Khoa Công nghệ Thông tin đã trang bị cho em những kiến thức quý báu thông qua đồ án chuyên ngành với đề tài “Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận” tạo cơ sở để em thực hiện hoàn thành đồ án.

Trong quá trình nghiên cứu đề tài, do kiến thức còn hạn chế nên em vẫn còn nhiều thiếu sót trong quá trình tìm hiểu, đánh giá và trình bày đề tài. Em rất mong nhận được những đóng góp ý kiến của cô để em khắc phục và phát triển để có thể từng bước hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN	i
NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG	ii
LỜI CẢM ƠN.....	iii
MỤC LỤC	iv
DANH MỤC HÌNH ẢNH.....	vi
DANH MỤC BẢNG BIỂU.....	vii
DANH MỤC TỪ VIẾT TẮT	viii
TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH.....	ix
MỞ ĐẦU	xii
CHƯƠNG 1. TỔNG QUAN	1
1.1. Lý do chọn đề tài	1
1.2. Những vấn đề cần giải quyết	1
1.3. Các nội dung nghiên cứu chính	2
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT	4
2.1. Quy trình nghiệp vụ về hệ thống quản lý và hỗ trợ tình nguyện viên	4
2.1.1. Nghiệp vụ của hệ thống quản lý và hỗ trợ tình nguyện viên.....	4
2.1.2. Quy trình nghiệp vụ quản lý tình nguyện viên	5
2.2. Giới thiệu về các công nghệ và công cụ sử dụng trong đề tài	6
2.2.1. React	6
2.2.2. Vite	6
2.2.3. Tailwind CSS.....	7
2.2.4. Node.js	8
2.2.5. Express	10
2.2.6. PostgreSQL.....	11
2.2.7. Prisma ORM	12
2.2.8. JWT và bcrypt	12
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	14
3.1. Đặc tả các yêu cầu	14
3.1.1. Yêu cầu chức năng	14
3.1.2. Yêu cầu phi chức năng	15
3.2. Thiết kế cơ sở dữ liệu quan hệ.....	16
3.2.1. Mô hình thực thể kết hợp (ERD - Entity Relationship Diagram)	16
3.2.2. Mô hình dữ liệu quan hệ.....	18
3.2.3. Mô hình dữ liệu vật lý (Physical Data Model - PDM)	19

3.3. Mô tả các bảng.....	21
3.4. Các khóa ràng buộc	27
3.5. Kiến trúc hệ thống	30
3.5.1. Sơ đồ Use case.....	30
3.5.2. Sơ đồ DFD.....	31
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	33
4.1. Giao diện Đăng ký	33
4.2. Giao diện Đăng nhập	34
4.3. Quản trị viên	35
4.3.1. Bảng điều khiển trung tâm	35
4.3.2. Quản lý Tổ chức	36
4.3.3. Quản lý Người dùng	36
4.3.4. Kiểm duyệt Hoạt động.....	37
4.3.5. Thống kê hệ thống.....	38
4.4. Phân hệ Tổ chức	38
4.4.1. Dashboard của Tổ chức	38
4.4.2. Quản lý Hoạt động của Tổ chức.....	39
4.4.3. Danh sách Tình nguyện viên tham gia	40
4.4.4. Báo cáo thống kê Tổ chức	40
4.5. Phân hệ Tình nguyện viên	41
4.5.1. Trang chủ cá nhân	41
4.5.2. Danh sách hoạt động khả dụng.....	42
4.5.3. Quản lý hoạt động đã đăng ký	43
4.5.4. Cập nhật hồ sơ cá nhân.....	43
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	45
5.1. Kết luận.....	45
5.2. Hạn chế	46
5.3. Hướng phát triển.....	46
DANH MỤC TÀI LIỆU THAM KHẢO	47
PHỤ LỤC	48

DANH MỤC HÌNH ẢNH

Hình 2.1 Giới thiệu về React	6
Hình 2.2 Vite là gì?	7
Hình 2.3 Tailwind CSS có gì hot?	8
Hình 2.4 Node.js là gì?	8
Hình 2.5 Nodejs hoạt động ra sao?	9
Hình 2.6 PostgreSQL mạnh mẽ như thế nào?	11
Hình 2.7 Prisma là gì?	12
Hình 3.1 Mô hình thực thể kết hợp	17
Hình 3.2 Mô hình dữ liệu vật lý	20
Hình 3.3 Sơ đồ Use case.....	31
Hình 3.4 Sơ đồ DFD mức 0.....	31
Hình 3.5 Sơ đồ DFD mức 1	32
Hình 3.6 Sơ đồ DFD mức 2.....	32
Hình 4.1 Giao diện trang đăng ký	33
Hình 4.2 Giao diện trang đăng nhập.....	34
Hình 4.3 Giao diện Dashboard quản trị hệ thống.....	35
Hình 4.6 Giao diện Quản lý hoạt động của Admin	37
Hình 4.7 Giao diện Thống kê của Admin.....	38
Hình 4.8 Giao diện Dashboard Tổ chức	38
Hình 4.9 Giao diện Quản lý hoạt động của Tổ chức	39
Hình 4.10 Giao diện Danh sách tình nguyện viên của tổ chức	40
Hình 4.11 Giao diện Thống kê của Tổ chức	40
Hình 4.12 Giao diện Dashboard của Tình nguyện viên	41
Hình 4.13 Giao diện Danh sách hoạt động khả dụng	42
Hình 4.14 Giao diện Danh sách hoạt động đã đăng ký	43
Hình 4.15 Giao diện tùy chỉnh thông tin cá nhân.....	43

DANH MỤC BẢNG BIỂU

Bảng 3.1 Danh sách và mô tả thuộc tính cho bảng COUNTRY	21
Bảng 3.2 Danh sách và mô tả thuộc tính cho bảng VOLUNTEER	21
Bảng 3.3 Danh sách và mô tả thuộc tính cho bảng ORGANIZATION.....	22
Bảng 3.4 Danh sách và mô tả thuộc tính cho bảng EVENT	22
Bảng 3.5 Danh sách và mô tả thuộc tính cho bảng SKILL	23
Bảng 3.6 Danh sách và mô tả thuộc tính cho bảng USERSKILL.....	24
Bảng 3.7 Danh sách và mô tả thuộc tính cho bảng EVENTSkill	24
Bảng 3.8 Danh sách và mô tả thuộc tính cho bảng JOIN.....	24
Bảng 3.9 Danh sách và mô tả thuộc tính cho bảng PARTICIPATION	25
Bảng 3.10 Danh sách và mô tả thuộc tính cho bảng LIST_OF_SUPPORT_TASKS ..	25
Bảng 3.11 Danh sách và mô tả thuộc tính cho bảng SUPPORTFORVOLUNTEERS.	25
Bảng 3.12 Danh sách và mô tả thuộc tính cho bảng NOTIFICATION	26
Bảng 3.13 Danh sách và mô tả thuộc tính cho bảng NOTIFICATION_USER.....	26
Bảng 3.14 Danh sách và mô tả thuộc tính cho bảng PROVINCE	26
Bảng 3.15 Danh sách và mô tả thuộc tính cho bảng COMMUNE	27
Bảng 3.16 Danh sách và mô tả thuộc tính cho bảng VOLUNTEERLOCATION.....	27
Bảng 3.17 Bảng các khóa chính	27
Bảng 3.18 Bảng các khóa ngoại	28

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Giải thích
1	API	Application Programming Interface
2	CSDL	Cơ sở dữ liệu
3	CSS	Cascading Style Sheets
4	DFD	Data Flow Diagram
5	ERD	Entity Relationship Diagram
6	FK	Foreign Key
7	JWT	JSON Web Token
8	ORM	Object-Relational Mapping
9	PDM	Physical Data Model
10	PK	Primary Key

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Mục tiêu của đề tài

Đề tài “Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận” nhằm thiết kế và xây dựng một hệ thống web hỗ trợ các tổ chức phi lợi nhuận trong việc quản lý thông tin tình nguyện viên, hoạt động tình nguyện một cách hiệu quả, đồng bộ và thuận tiện. Hệ thống hướng đến việc tin học hóa quy trình quản lý, hỗ trợ điều phối công việc, theo dõi tiến độ hoạt động và nâng cao hiệu quả phối hợp giữa tổ chức và tình nguyện viên, đặc biệt là tình nguyện viên quốc tế tham gia hoạt động tại Việt Nam.

Phạm vi nghiên cứu

Đề tài tập trung nghiên cứu, thiết kế và xây dựng hệ thống quản lý nội bộ cho một tổ chức phi lợi nhuận giả định. Phạm vi chức năng bao gồm: quản lý thông tin cá nhân tình nguyện viên, quản lý hoạt động tình nguyện, đăng ký tham gia hoạt động, phân công nhiệm vụ, theo dõi tiến độ và thống kê, báo cáo kết quả hoạt động. Đề tài không đề cập đến các chức năng liên quan đến tài chính, quyên góp hoặc thanh toán trực tuyến.

Thiết kế cơ sở dữ liệu

Hệ thống được thiết kế dựa trên việc xác định các thực thể chính như tình nguyện viên, hoạt động, phân công nhiệm vụ và người dùng hệ thống, cùng các mối quan hệ giữa chúng. Cơ sở dữ liệu được xây dựng trên PostgreSQL với các ràng buộc khóa chính, khóa ngoại nhằm đảm bảo tính toàn vẹn và hiệu năng truy xuất dữ liệu. Prisma ORM được sử dụng để quản lý mô hình dữ liệu, migration và khởi tạo dữ liệu mẫu phục vụ quá trình phát triển và kiểm thử hệ thống.

Kiểm thử và triển khai

Hệ thống được triển khai theo mô hình web với Backend sử dụng Node.js và Express, Frontend sử dụng React kết hợp Vite và TailwindCSS. Quá trình kiểm thử được thực hiện trên môi trường phát triển thông qua việc kiểm thử các chức năng chính của hệ thống và kiểm thử API bằng các công cụ như Postman. Kết quả kiểm thử cho thấy hệ thống đáp ứng tốt các yêu cầu đã đề ra, hoạt động ổn định với dữ liệu mẫu.

Ý nghĩa của đề tài

Đề tài góp phần hỗ trợ các tổ chức phi lợi nhuận trong việc quản lý và điều phối tình nguyện viên một cách khoa học và hiệu quả hơn. Hệ thống giúp giảm thiểu công việc thủ công, hạn chế sai sót trong quản lý thông tin, nâng cao khả năng theo dõi và báo cáo hoạt động tình nguyện. Đồng thời, đề tài có thể làm cơ sở để tiếp tục phát triển, mở rộng thêm các chức năng hỗ trợ khác trong tương lai, phù hợp với nhu cầu thực tế của các tổ chức phi lợi nhuận.

MỞ ĐẦU

1) Mô tả

Trong những năm gần đây, các hoạt động tình nguyện và thiện nguyện ngày càng phát triển mạnh mẽ, thu hút sự tham gia của đông đảo tình nguyện viên trong nước và quốc tế. Các tổ chức phi lợi nhuận thường xuyên triển khai nhiều hoạt động với quy mô, địa điểm và thời gian khác nhau, đòi hỏi công tác quản lý tình nguyện viên phải chính xác, kịp thời và hiệu quả.

Tuy nhiên, trên thực tế, nhiều tổ chức vẫn đang quản lý tình nguyện viên bằng các phương pháp thủ công như bảng tính, nhóm trò chuyện hoặc hồ sơ giấy. Điều này gây ra nhiều khó khăn trong việc lưu trữ thông tin, phân công công việc, theo dõi tiến độ hoạt động và tổng hợp báo cáo. Đặc biệt, khi có sự tham gia của tình nguyện viên nước ngoài, việc quản lý và hỗ trợ càng trở nên phức tạp hơn.

Xuất phát từ thực tế đó, đề tài tập trung vào việc xây dựng một hệ thống phần mềm quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận. Hệ thống cho phép quản lý thông tin tình nguyện viên, quản lý hoạt động tình nguyện, hỗ trợ đăng ký tham gia, phân công nhiệm vụ và theo dõi tiến độ thực hiện. Ngoài ra, hệ thống còn cung cấp các chức năng thống kê và báo cáo nhằm hỗ trợ tổ chức đánh giá hiệu quả hoạt động và đưa ra các quyết định điều phối phù hợp. Việc áp dụng xác thực người dùng bằng JWT giúp đảm bảo an toàn và kiểm soát truy cập hệ thống.

2) Phương pháp thực hiện

2.1) Mục tiêu nghiên cứu

- Xây dựng hệ thống web quản lý và hỗ trợ tình nguyện viên có thể áp dụng cho một tổ chức phi lợi nhuận;
- Hỗ trợ tổ chức trong việc quản lý thông tin tình nguyện viên, hoạt động tình nguyện một cách hiệu quả;
- Giảm thiểu các hạn chế của phương pháp quản lý thủ công, nâng cao độ chính xác và khả năng theo dõi, thống kê hoạt động tình nguyện.

2.2) Nội dung nghiên cứu

- Tìm hiểu mô hình tổ chức và quy trình quản lý tình nguyện viên tại các tổ chức phi lợi nhuận;

Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận

- Xác định các nhóm người dùng tham gia hệ thống (quản trị viên, điều phối viên, tình nguyện viên);
- Phân tích yêu cầu chức năng và phi chức năng của hệ thống;
- Thiết kế mô hình cơ sở dữ liệu quan hệ phục vụ cho việc quản lý tình nguyện viên và hoạt động;
- Xây dựng giao diện web trực quan, dễ sử dụng;
- Xây dựng các chức năng cốt lõi của hệ thống, bao gồm:
 - + Quản lý thông tin tình nguyện viên;
 - + Quản lý hoạt động tình nguyện;
 - + Đăng ký tham gia và phân công nhiệm vụ;
 - + Theo dõi tiến độ hoạt động;
 - + Thống kê và báo cáo kết quả hoạt động.

2.3) Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là các quy trình và nghiệp vụ liên quan đến công tác quản lý và hỗ trợ tình nguyện viên trong tổ chức phi lợi nhuận.

Các nhóm người dùng chính bao gồm:

- Tình nguyện viên: đăng ký tài khoản, cập nhật thông tin cá nhân, đăng ký tham gia hoạt động và theo dõi nhiệm vụ được phân công;
- Điều phối viên/nhân sự tổ chức: quản lý danh sách tình nguyện viên, hoạt động, phân công công việc và theo dõi tiến độ;
- Admin: quản trị hệ thống, quản lý người dùng và giám sát dữ liệu tổng thể.

2.4) Phương pháp nghiên cứu

- Nghiên cứu tài liệu liên quan đến phân tích và thiết kế hệ thống thông tin;
- Nghiên cứu thực tế về công tác quản lý tình nguyện viên tại các tổ chức phi lợi nhuận;
- Sử dụng các công nghệ trong đề tài:
 - + Frontend: React + Vite + TailwindCSS;

- + Backend: Node.js + Express;
- + ORM: Prisma ORM;
- + Cơ sở dữ liệu: PostgreSQL;
- Tích hợp JWT và bcrypt để xác thực và bảo mật thông tin người dùng.
- Thực nghiệm, kiểm thử các chức năng của hệ thống với dữ liệu mẫu.

2.5) Phạm vi thực hiện đề tài

Hệ thống được xây dựng cho một tổ chức phi lợi nhuận giả định, tập trung vào công tác quản lý và hỗ trợ tình nguyện viên, hoạt động tình nguyện. Đề tài không bao gồm các chức năng liên quan đến tài chính, quyên góp hoặc thanh toán trực tuyến.

3) Kết quả đạt được

Kết quả của đề tài là hoàn thành một hệ thống web quản lý và hỗ trợ tình nguyện viên với các chức năng cơ bản, bao gồm:

- Giao diện web thân thiện, dễ sử dụng cho các nhóm người dùng;
- Quản lý thông tin tình nguyện viên và hoạt động;
- Hỗ trợ đăng ký tham gia, phân công nhiệm vụ và theo dõi tiến độ hoạt động;
- Cung cấp chức năng thống kê và báo cáo tổng hợp;
- Hệ thống hoạt động ổn định với dữ liệu mẫu, đáp ứng các yêu cầu đã đề ra trong đề cương.

CHƯƠNG 1. TỔNG QUAN

1.1. Lý do chọn đề tài

Trong những năm gần đây, các hoạt động tình nguyện và hoạt động cộng đồng tại Việt Nam ngày càng phát triển và thu hút sự tham gia của đông đảo tình nguyện viên trong nước và quốc tế. Nhiều tổ chức phi lợi nhuận, tổ chức xã hội và hoạt động cộng đồng thường xuyên tiếp nhận tình nguyện viên người nước ngoài đến làm việc và sinh sống tại Việt Nam trong các lĩnh vực như giáo dục, y tế, bảo vệ môi trường, phát triển cộng đồng và hỗ trợ các nhóm yếu thế.

Đối với tình nguyện viên quốc tế, ngoài việc tham gia vào các hoạt động chuyên môn, họ còn cần được hỗ trợ về nhiều mặt như thông tin hoạt động, lịch trình hoạt động, phân công nhiệm vụ, liên lạc với tổ chức tiếp nhận, cũng như các vấn đề liên quan đến sinh hoạt và thích nghi với môi trường, văn hóa địa phương. Điều này đặt ra yêu cầu cao hơn đối với công tác quản lý và hỗ trợ tình nguyện viên so với các mô hình quản lý truyền thống.

Tuy nhiên, trên thực tế, nhiều tổ chức phi lợi nhuận hiện nay vẫn quản lý tình nguyện viên bằng các phương pháp thủ công hoặc sử dụng các công cụ rời rạc như bảng tính, email và các nhóm trò chuyện. Việc quản lý phân tán này gây ra nhiều hạn chế như khó khăn trong việc lưu trữ và tra cứu thông tin, thiếu sự đồng bộ giữa các bộ phận, khó theo dõi quá trình tham gia hoạt động và hỗ trợ kịp thời cho tình nguyện viên, đặc biệt là tình nguyện viên người nước ngoài.

Việc xây dựng một hệ thống quản lý và hỗ trợ tình nguyện viên tập trung là cần thiết nhằm tin học hóa công tác quản lý, nâng cao hiệu quả điều phối nhân sự và cải thiện khả năng hỗ trợ tình nguyện viên. Hệ thống cho phép tổ chức theo dõi thông tin tình nguyện viên, quản lý hoạt động và hoạt động, phân công nhiệm vụ, theo dõi tiến độ thực hiện và tổng hợp báo cáo một cách nhanh chóng và chính xác. Đây chính là cơ sở để đề tài “Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận” được lựa chọn và triển khai.

1.2. Những vấn đề cần giải quyết

Để xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên đáp ứng yêu cầu thực tế của tổ chức phi lợi nhuận, đề tài tập trung giải quyết các vấn đề chính sau:

Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận

- Thiết kế cấu trúc cơ sở dữ liệu: Xác định các thực thể chính như tình nguyện viên, hoạt động, nhiệm vụ và người dùng hệ thống; đồng thời xây dựng mối quan hệ giữa các thực thể để hình thành mô hình dữ liệu phù hợp;
- Lựa chọn hệ quản trị cơ sở dữ liệu: Sử dụng PostgreSQL làm hệ quản trị cơ sở dữ liệu nhờ khả năng xử lý ổn định, hỗ trợ ràng buộc chặt chẽ và truy vấn hiệu quả. Kết hợp Prisma ORM để thao tác dữ liệu an toàn, giảm lỗi và hỗ trợ quản lý migration trong quá trình phát triển;
- Xây dựng và ràng buộc dữ liệu: Thiết kế các bảng dữ liệu cùng với khóa chính, khóa ngoại và các ràng buộc nhằm đảm bảo tính toàn vẹn và nhất quán của dữ liệu trong hệ thống;
- Kết nối cơ sở dữ liệu với ứng dụng: Tích hợp cơ sở dữ liệu PostgreSQL với backend sử dụng Node.js và Express, đồng thời cung cấp các API phục vụ frontend React. Hệ thống cần đảm bảo khả năng xử lý nhiều yêu cầu đồng thời và hoạt động ổn định;
- Đảm bảo an toàn và bảo mật hệ thống: Áp dụng cơ chế xác thực và phân quyền người dùng bằng JWT, kết hợp bcrypt để mã hóa thông tin nhạy cảm, đảm bảo an toàn dữ liệu và kiểm soát truy cập hệ thống;
- Đảm bảo khả năng mở rộng và bảo trì: Hệ thống cần được thiết kế theo hướng dễ mở rộng và bảo trì, đáp ứng nhu cầu phát triển trong tương lai của tổ chức phi lợi nhuận;
- Xây dựng và kiểm thử API: Thiết kế các API theo mô hình RESTful, kiểm thử bằng các công cụ như Postman nhằm đảm bảo API hoạt động đúng chức năng và xử lý lỗi hợp lý.

1.3. Các nội dung nghiên cứu chính

Các nội dung nghiên cứu chính của đề tài bao gồm:

- Phân tích yêu cầu hệ thống: Thu thập và phân tích các yêu cầu chức năng và phi chức năng của hệ thống quản lý và hỗ trợ tình nguyện viên;
- Thiết kế cơ sở dữ liệu: Thiết kế mô hình dữ liệu mức quan niệm (ERD) và mô hình dữ liệu mức logic phù hợp với nghiệp vụ quản lý tình nguyện viên;
- Cài đặt và cấu hình cơ sở dữ liệu: Thiết lập PostgreSQL, xây dựng migration và dữ liệu mẫu phục vụ cho quá trình phát triển và kiểm thử;

Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận

- Thiết kế và triển khai backend: Xây dựng backend với Node.js và Express, sử dụng Prisma ORM làm lớp truy xuất dữ liệu; thiết kế các API phục vụ các chức năng nghiệp vụ;
- Thiết kế và triển khai frontend: Xây dựng giao diện người dùng bằng React kết hợp Vite và TailwindCSS, đảm bảo giao diện trực quan và dễ sử dụng;
- Bảo mật hệ thống: Áp dụng xác thực JWT và mã hóa dữ liệu nhạy cảm nhằm đảm bảo an toàn thông tin;
- Kiểm thử và đánh giá: Thực hiện kiểm thử các chức năng chính của hệ thống với dữ liệu mẫu và đánh giá mức độ đáp ứng yêu cầu đã đề ra.

CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

2.1. Quy trình nghiệp vụ về hệ thống quản lý và hỗ trợ tình nguyện viên

2.1.1. Nghiệp vụ của hệ thống quản lý và hỗ trợ tình nguyện viên

Hệ thống quản lý và hỗ trợ tình nguyện viên đóng vai trò quan trọng trong việc giúp các tổ chức phi lợi nhuận quản lý nhân sự, tổ chức hoạt động và hỗ trợ tình nguyện viên một cách hiệu quả. Hệ thống hướng đến việc quản lý tập trung thông tin, giảm thiểu thao tác thủ công và nâng cao khả năng theo dõi, đánh giá hoạt động tình nguyện. Các nghiệp vụ chính của hệ thống được phân theo từng nhóm người dùng như sau:

Đối với Tình nguyện viên:

- Đăng ký và cập nhật thông tin cá nhân: Cập nhật thông tin liên hệ, kỹ năng và các thông tin cần thiết khác;
- Tra cứu hoạt động: Xem danh sách các hoạt động đang diễn ra hoặc sắp triển khai;
- Đăng ký tham gia hoạt động: Gửi yêu cầu tham gia các hoạt động phù hợp với khả năng và thời gian;
- Theo dõi nhiệm vụ được phân công: Xem thông tin nhiệm vụ, thời gian, địa điểm và trạng thái thực hiện;
- Nhận thông báo: Nhận các thông báo từ tổ chức về lịch trình, thay đổi hoặc yêu cầu hỗ trợ.

Đối với Điều phối viên/Nhân sự tổ chức:

- Quản lý tình nguyện viên: Tạo, cập nhật và quản lý hồ sơ tình nguyện viên;
- Quản lý hoạt động: Tạo và cập nhật thông tin hoạt động;
- Phân công nhiệm vụ: Gán nhiệm vụ với từng hoạt động;
- Theo dõi tiến độ: Theo dõi trạng thái thực hiện nhiệm vụ và mức độ tham gia;
- Thống kê và báo cáo: Tổng hợp dữ liệu về số lượng tình nguyện viên, hoạt động và kết quả thực hiện.

Đối với Quản trị viên:

- Quản lý tài khoản người dùng: Tạo, chỉnh sửa, vô hiệu hóa hoặc xóa tài khoản khi cần thiết;
- Phân quyền truy cập: Thiết lập quyền cho từng nhóm người dùng trong hệ thống;
- Giám sát hệ thống: Theo dõi tình trạng hoạt động và dữ liệu tổng thể của hệ thống;
- Báo cáo tổng quan: Xem các báo cáo thống kê tổng hợp phục vụ công tác quản lý.

2.1.2. Quy trình nghiệp vụ quản lý tình nguyện viên

- Tiếp nhận và quản lý tình nguyện viên
 - Tình nguyện viên đăng ký tài khoản và cung cấp thông tin cá nhân;
 - Cập nhật trạng thái tài khoản của tình nguyện viên;
 - Hệ thống lưu trữ lịch sử tham gia hoạt động của từng tình nguyện viên.
- Quản lý hoạt động tình nguyện
 - Điều phối viên tạo các hoạt động cụ thể;
 - Cập nhật thông tin về mục tiêu, thời gian, địa điểm và số lượng tình nguyện viên cần thiết.
- Đăng ký và phân công nhiệm vụ
 - Tình nguyện viên đăng ký tham gia hoạt động;
 - Điều phối viên xét duyệt và phân công nhiệm vụ cho từng tình nguyện viên;
 - Hệ thống ghi nhận trạng thái phân công và tiến độ thực hiện.
- Theo dõi và đánh giá hoạt động
 - Điều phối viên theo dõi tiến độ thực hiện nhiệm vụ;
 - Hệ thống cập nhật trạng thái hoàn thành và lưu trữ kết quả hoạt động;
 - Thống kê số lượng tình nguyện viên tham gia và hiệu quả hoạt động.

2.2. Giới thiệu về các công nghệ và công cụ sử dụng trong đề tài

2.2.1. React

React còn được gọi là ReactJS hoặc React.js, là một thư viện JavaScript mã nguồn mở được phát triển bởi đội ngũ kỹ sư đến từ Facebook; nó được giới thiệu vào năm 2011, tuy nhiên đến năm 2013 mới được giới thiệu cho cộng đồng lập trình viên [1].

Tác giả của React là Jordan Walke - một kỹ sư phần mềm tại Facebook. Ban đầu, nó được dùng nội bộ tại Facebook và Instagram để giải quyết các vấn đề về hiệu suất khi xây dựng các giao diện người dùng phức tạp. Sau đó, nền tảng này đã mở mã nguồn cho cộng đồng sử dụng và phát triển.



Hình 2.1 Giới thiệu về React

(*Nguồn: <https://github.com/juansedev/react>*)

React ra đời với mục tiêu dùng công nghệ để tạo ra giao diện người dùng dễ dàng hơn. Với nền tảng này, các lập trình viên có thể tạo ra các thành phần mà dùng lại được và tăng khả năng tương tác.

Từ đầu giai đoạn đó, React đã rất nhanh trở thành một trong những thư viện Javascript được dùng phổ biến nhất trong việc thiết kế giao diện người dùng.

Qua mỗi giai đoạn phát triển, thư viện này đã không ngừng cải tiến và mở rộng. Sự phát triển của React cũng được thúc đẩy mạnh mẽ bởi cộng đồng và sự hỗ trợ liên tục từ Facebook [2].

2.2.2. Vite

Vite là một công cụ build mới được tạo bởi Evan You, người sáng lập Vue.js. Vite được thiết kế để cung cấp một trải nghiệm phát triển nhanh chóng và hiệu quả hơn so với các công cụ build truyền thống như Webpack.



Hình 2.2 Vite là gì?

(*Nguồn: <https://viblo.asia/p/vitejs-nguo Choi-he-hackspeed-trong-lang-frontend-build-tools-maGK7A8Llj2>*)

Khác với các công cụ cũ thường phải đóng gói toàn bộ mã nguồn trước khi khởi động, Vite tận dụng khả năng hỗ trợ Native ES Modules trực tiếp của các trình duyệt hiện đại, giúp ứng dụng khởi động gần như tức thì trong môi trường phát triển. Ngoài ra, Vite còn cung cấp cơ chế Hot Module Replacement (HMR) với tốc độ phản hồi cực nhanh, cho phép lập trình viên theo dõi ngay lập tức các thay đổi khi chỉnh sửa mã nguồn mà không cần tải lại toàn bộ trang. Trong phạm vi đồ án, Vite được lựa chọn kết hợp cùng React để tối ưu hóa hiệu suất phát triển, rút ngắn thời gian build và hỗ trợ quá trình kiểm thử giao diện một cách hiệu quả nhất.

2.2.3. Tailwind CSS

Tailwind CSS là một framework CSS hiện đại theo hướng “utility-first”, cho phép lập trình viên xây dựng giao diện web bằng cách sử dụng các lớp tiện ích nhỏ gọn (utility classes) thay vì viết CSS thủ công hoặc dựa vào các thành phần thiết kế sẵn.

Không giống như các framework truyền thống cung cấp sẵn các thành phần như nút, bảng hoặc biểu mẫu với phong cách mặc định, Tailwind cung cấp cho người dùng một bộ công cụ linh hoạt để tự kết hợp và tạo ra bất kỳ kiểu giao diện nào một cách tự do, trực quan và có kiểm soát [3].



Hình 2.3 Tailwind CSS có gì hot?

(Nguồn: <https://viblo.asia/p/tailwind-css-v2-co-gi-moi-6J3ZgNMMKmB>)

Ưu điểm của Tailwind CSS:

- **Tùy chỉnh linh hoạt:** Tailwind cho phép cấu hình chi tiết thông qua file tailwind.config.js, từ màu sắc, khoảng cách cho đến font chữ. Nhờ đó, lập trình viên có thể thiết lập hệ thống thiết kế riêng phù hợp với từng dự án mà không cần ghi đè hoặc viết lại CSS nhiều lần.
- **Hỗ trợ responsive và dark mode dễ dàng:** Các lớp tiện ích của Tailwind đi kèm với các biến thẻ như `md:`, `lg:`, `dark:` giúp tạo giao diện phản hồi và chế độ tối chỉ với vài class đơn giản mà không cần viết media queries thủ công [4].

2.2.4. Node.js

NodeJS là một nền tảng được xây dựng trên “V8 Javascript engine” được viết bằng C++ và JavaScript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009 [5].



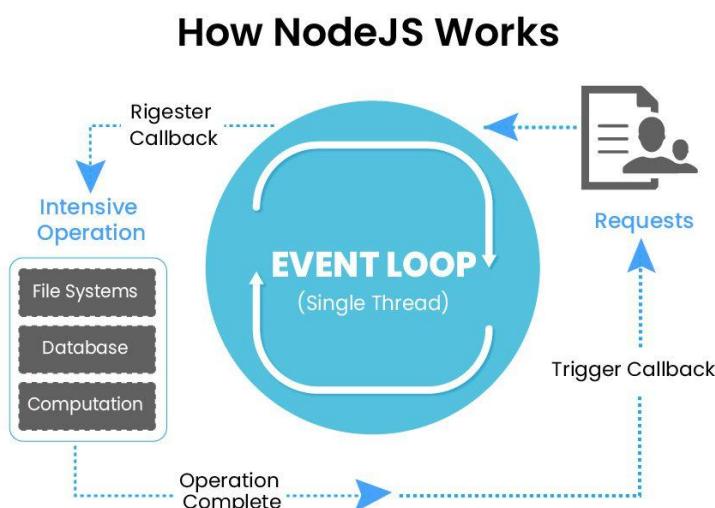
Hình 2.4 Node.js là gì?

(Nguồn: <https://topdev.vn/blog/node-js-la-gi/>)

Node.js ra đời khi các developer đài đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy trên máy của mình dưới dạng ứng dụng độc lập.

- **Nguồn mở (Open-source):** Mã nguồn của Node.js được công bố công khai, được duy trì bởi cộng đồng lập trình viên trên toàn thế giới
- **Đa nền tảng (Cross-platform):** Node.js không phụ thuộc vào bất kỳ hệ điều hành nào cụ thể nào, nghĩa là nó có thể chạy trên Linux, macOS hoặc Windows.
- **Môi trường thực thi JavaScript (JavaScript runtime environment):** Để mã JavaScript có thể được thực thi, nó cần một môi trường chạy phù hợp. Trong khi trình duyệt như Chrome và Firefox cung cấp một môi trường thực thi cho JavaScript, Node.js mở rộng khả năng này ra ngoài trình duyệt. Node.js cho phép chạy JavaScript trên máy chủ, hoặc trong bất kỳ môi trường máy tính nào khác, không chỉ trong trình duyệt.
- **Dựa trên V8 JavaScript Engine:** Node.js được xây dựng dựa trên V8, động cơ JavaScript được phát triển bởi Google cho trình duyệt Chrome. Điều này giúp Node.js có khả năng thực thi JavaScript nhanh và hiệu quả, đồng thời hỗ trợ các tính năng mới nhất của ngôn ngữ JavaScript.

Node.js đã mở rộng khả năng của JavaScript từ việc chỉ phát triển front-end trong trình duyệt để bao gồm cả phát triển back-end.



Hình 2.5 Nodejs hoạt động ra sao?

(Nguồn: <https://topdev.vn/blog/node-js-la-gi/>)

Node.js hoạt động dựa trên một số nguyên tắc cơ bản giúp nó hiệu quả trong việc xử lý các ứng dụng có nhiều hoạt động nhập/xuất (I/O) mà không bị chặn, đồng thời giảm đáng kể sự phức tạp trong quản lý các luồng thực thi. Dưới đây là một số thành phần chính giải thích cách thức hoạt động của Node.js:

- Kiến trúc Non-blocking and event-driven: Node.js xử lý các tác vụ như đọc file, truy vấn cơ sở dữ liệu hoặc gửi yêu cầu mạng mà không chặn tiến trình chính. Thay vì chờ xong một việc mới làm tiếp, Node.js tiếp tục xử lý việc khác và sẽ quay lại khi có kết quả - thông qua hàm callback.
- V8 Engine: Node.js được xây dựng trên động cơ JavaScript V8 của Google Chrome, đây là một động cơ rất nhanh cho phép biên dịch mã JavaScript thành mã máy để thực thi trực tiếp trên phần cứng, làm tăng hiệu suất thực thi.
- Single-thread + Event Loop: Dù chỉ chạy trên một luồng chính, Node.js vẫn xử lý hàng ngàn yêu cầu nhờ cơ chế event loop - một vòng lặp lắng nghe và xử lý sự kiện bất cứ khi nào có kết quả trả về.
- npm - Hệ sinh thái mạnh mẽ: Node.js có trình quản lý gói npm - nơi chứa hàng ngàn thư viện mã nguồn mở giúp phát triển ứng dụng nhanh hơn [5].

2.2.5. Express

Express là một framework web nhẹ được xây dựng trên nền Node.js, cung cấp các cơ chế routing, middleware và hỗ trợ xây dựng API theo kiến trúc RESTful. Việc kết hợp Node.js và Express giúp đơn giản hóa quá trình phát triển backend, tăng tính linh hoạt và khả năng mở rộng của hệ thống [6].

Điểm mạnh của Express nằm ở cơ chế hỗ trợ middleware (phần mềm trung gian), cho phép lập trình viên dễ dàng triển khai các chức năng quan trọng như xác thực người dùng, phân quyền, ghi nhật ký (logging) và xử lý lỗi một cách hệ thống. Nhờ cấu trúc đơn giản, hiệu năng cao và cộng đồng hỗ trợ lớn, Express đã trở thành một trong những công cụ phổ biến nhất trong các ứng dụng web hiện đại. Trong phạm vi đề tài này, Express đóng vai trò là nền tảng cốt lõi để xây dựng hệ thống phía máy chủ (Backend), chịu trách nhiệm xử lý logic nghiệp vụ và cung cấp các API cần thiết để phục vụ cho giao diện React ở phía người dùng.

2.2.6. PostgreSQL

PostgreSQL là một hệ thống cơ sở dữ liệu quan hệ đối tượng mã nguồn mở mạnh mẽ, sử dụng và mở rộng ngôn ngữ SQL kết hợp với nhiều tính năng giúp lưu trữ và mở rộng an toàn các khối lượng công việc dữ liệu phức tạp nhất. Nguồn gốc của PostgreSQL bắt nguồn từ năm 1986, là một phần của dự án POSTGRES tại Đại học California ở Berkeley và đã có gần 40 năm phát triển tích cực trên nền tảng cốt lõi.

PostgreSQL đã tạo dựng được danh tiếng vững chắc nhờ kiến trúc đã được chứng minh, độ tin cậy, tính toàn vẹn dữ liệu, bộ tính năng mạnh mẽ, khả năng mở rộng và sự tận tâm của cộng đồng nguồn mở đãng sau phần mềm để liên tục cung cấp các giải pháp hiệu suất cao và sáng tạo. PostgreSQL chạy trên tất cả các hệ điều hành chính, tuân thủ ACID từ năm 2001 và có các tiện ích bổ sung mạnh mẽ như trình mở rộng cơ sở dữ liệu không gian địa lý PostGIS phổ biến.

Không có gì ngạc nhiên khi PostgreSQL đã trở thành cơ sở dữ liệu quan hệ nguồn mở được nhiều cá nhân và tổ chức lựa chọn. Bắt đầu sử dụng PostgreSQL chưa bao giờ dễ dàng hơn thế - hãy chọn một dự án bạn muốn xây dựng và để PostgreSQL lưu trữ dữ liệu của bạn một cách an toàn và mạnh mẽ [7].



Hình 2.6 PostgreSQL mạnh mẽ như thế nào?

(Nguồn: <https://viblo.asia/p/postgresql-co-tot-hon-mysql-khong-MkNLr5mqJgA>)

PostgreSQL đi kèm với nhiều tính năng nhằm giúp các nhà phát triển xây dựng ứng dụng, quản trị viên bảo vệ tính toàn vẹn dữ liệu và xây dựng môi trường chịu lỗi, đồng thời giúp bạn quản lý dữ liệu bất kể tập dữ liệu lớn hay nhỏ. Ngoài việc miễn phí và mã nguồn mở, PostgreSQL còn có khả năng mở rộng cao.

2.2.7. Prisma ORM

Prisma là công cụ ORM hiện đại hỗ trợ kết nối giữa ứng dụng Node.js và cơ sở dữ liệu quan hệ như PostgreSQL. Prisma cho phép thao tác dữ liệu thông qua các mô hình định nghĩa sẵn, giúp giảm lỗi khi truy vấn và tăng tính nhất quán giữa mã nguồn và cấu trúc cơ sở dữ liệu [8].

Dù bạn đang sử dụng PostgreSQL, MySQL, SQLite, SQL Server hay MongoDB, Prisma sẽ giúp bạn quản lý dữ liệu một cách hiệu quả và an toàn [8].



Hình 2.7 Prisma là gì?

(Nguồn: <https://www.marketenterprise.vn/blog/prisma-ho-tro-phat-trien-phan-1.html>)

Tính năng nổi bật của Prisma

- Phát triển dựa trên schema: Định nghĩa mô hình dữ liệu trong tệp **schema.prisma**, sau đó Prisma tự động tạo các truy vấn và cấu trúc dữ liệu.
- An toàn kiểu dữ liệu: Tích hợp chặt chẽ với TypeScript, Prisma đảm bảo sự an toàn về kiểu dữ liệu trong toàn bộ ứng dụng.
- Migration dữ liệu: Công cụ di chuyển (migrations) giúp quản lý các thay đổi trong cơ sở dữ liệu một cách dễ dàng và có tổ chức.
- Truy vấn tự động: Tự động tạo các truy vấn CRUD (Create, Read, Update, Delete) cho các mô hình dữ liệu.
- Prisma Studio: Giao diện người dùng trực quan giúp quản lý và tương tác với dữ liệu trong cơ sở dữ liệu.

2.2.8. JWT và bcrypt

JWT (JSON Web Token) là một chuẩn mở theo RFC 7519, được sử dụng rộng rãi trong các ứng dụng web nhằm phục vụ cho mục đích xác thực và phân quyền người

dùng. JWT cho phép truyền tải thông tin dưới dạng JSON một cách an toàn và hỗ trợ cơ chế xác thực không trạng thái (stateless), giúp máy chủ không cần lưu trữ thông tin phiên làm việc, từ đó nâng cao hiệu suất và khả năng mở rộng của hệ thống.

bcrypt là một thư viện mã hóa mật khẩu phổ biến, sử dụng thuật toán băm một chiều kết hợp với cơ chế sinh salt tự động và hệ số chi phí (cost factor), giúp bảo vệ mật khẩu người dùng trước các cuộc tấn công brute-force và rainbow table. Việc kết hợp JWT để xác thực, phân quyền và bcrypt để mã hóa mật khẩu giúp đảm bảo tính an toàn, bảo mật và độ tin cậy cho hệ thống quản lý tình nguyện viên, đồng thời đáp ứng các yêu cầu về bảo mật trong các ứng dụng web hiện đại. Các nội dung lý thuyết trên được tham khảo từ tài liệu chính thức của JWT và các thư viện triển khai phổ biến trong môi trường Node.js.

bcrypt sử dụng phổ biến trong các hệ thống web, giúp bảo vệ thông tin người dùng khỏi các hình thức tấn công dò mật khẩu. Việc kết hợp JWT và bcrypt góp phần nâng cao mức độ bảo mật cho hệ thống quản lý người dùng [9].

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Đặc tả các yêu cầu

3.1.1. Yêu cầu chức năng

Hệ thống quản lý tình nguyện viên được xây dựng nhằm hỗ trợ công tác quản lý tình nguyện viên, hoạt động tình nguyện một cách hiệu quả và minh bạch. Các yêu cầu chức năng chính của hệ thống bao gồm:

a) Quản lý tài khoản và phân quyền

- Hệ thống cho phép tạo, đăng nhập và đăng ký tài khoản người dùng;
- Phân quyền người dùng theo vai trò:
 - + Admin: Quản lý toàn bộ hệ thống, phân quyền cho các vai trò khác;
 - + Điều phối viên/Tổ chức: Quản lý và theo dõi hoạt động;
 - + Tình nguyện viên: Đăng ký tham gia hoạt động tình nguyện, cập nhật thông tin cá nhân, xem lịch sử hoạt động.
- Mỗi vai trò được cấp các quyền truy cập và thao tác phù hợp với nghiệp vụ.

b) Quản lý thông tin tình nguyện viên

- Cho phép thêm mới, cập nhật và xem thông tin cá nhân của tình nguyện viên (họ tên, ngày sinh, giới tính, số điện thoại, email, địa chỉ, kỹ năng, tình trạng hoạt động...);
- Hiển thị lịch sử tham gia hoạt động của từng tình nguyện viên;
- Cho phép tình nguyện viên cập nhật thông tin cá nhân trực tiếp qua giao diện.

c) Quản lý hoạt động

- Điều phối viên hoặc Admin có thể tạo, chỉnh sửa và xóa hoạt động;
- Mỗi hoạt động bao gồm thông tin: tên hoạt động, mô tả, địa điểm, thời gian bắt đầu - kết thúc, số lượng tình nguyện viên cần tham gia;
- Hệ thống tự động hiển thị tình trạng hoạt động (Sắp diễn ra, Đang thực hiện, Hoàn thành).

d) Quản lý hoạt động tình nguyện

- Điều phối viên lập hoạt động cụ thể trong hoạt động, gồm: ngày tổ chức, địa điểm, số lượng tình nguyện viên, mô tả công việc;
- Tình nguyện viên có thể đăng ký tham gia hoặc rút đăng ký trước ngày diễn ra hoạt động;
- Hệ thống ghi nhận số giờ tình nguyện và trạng thái tham gia (Đã tham gia, Vắng mặt, Hủy đăng ký).

e) Quản lý phân công tình nguyện viên

- Điều phối viên có thể phân công tình nguyện viên vào các hoạt động cụ thể;
- Cập nhật tình trạng tham gia và thông báo tự động cho tình nguyện viên;
- Cho phép điều phối viên theo dõi danh sách tình nguyện viên theo hoạt động.

f) Tra cứu và thống kê

- Cho phép tìm kiếm và lọc dữ liệu tình nguyện viên, hoạt động theo nhiều tiêu chí (họ tên, số điện thoại, kỹ năng, trạng thái tham gia...);
- Thống kê số lượng tình nguyện viên tham gia, số giờ tình nguyện, hoạt động đang thực hiện và hoàn thành;
- Hiển thị báo cáo tổng quan phục vụ công tác quản lý.

3.1.2. Yêu cầu phi chức năng

a) Hiệu năng

- Hệ thống phải phản hồi nhanh với các thao tác truy vấn dữ liệu, kể cả khi có nhiều người dùng truy cập đồng thời;
- Tối ưu truy vấn để đảm bảo tốc độ tải trang và xử lý thông tin hoạt động.

b) Bảo mật

- Thông tin cá nhân tình nguyện viên và dữ liệu hoạt động phải được bảo mật tuyệt đối;
- Áp dụng cơ chế xác thực và phân quyền rõ ràng, chỉ người dùng được phép mới truy cập dữ liệu liên quan.

c) Tính chính xác và toàn vẹn dữ liệu

- Dữ liệu được kiểm tra và ràng buộc qua khóa chính, khóa ngoại;
- Hệ thống đảm bảo không xảy ra mâu thuẫn dữ liệu trong quá trình cập nhật thông tin tình nguyện viên hoặc hoạt động.

d) Khả năng mở rộng và bảo trì

- Hệ thống được thiết kế theo mô hình client-server hoặc API-Driven, dễ dàng mở rộng và nâng cấp;
- Cấu trúc mã nguồn rõ ràng, thuận tiện cho việc thêm các module mới (ví dụ: đánh giá tình nguyện viên, phản hồi hoạt động).

e) Tính thân thiện với người dùng

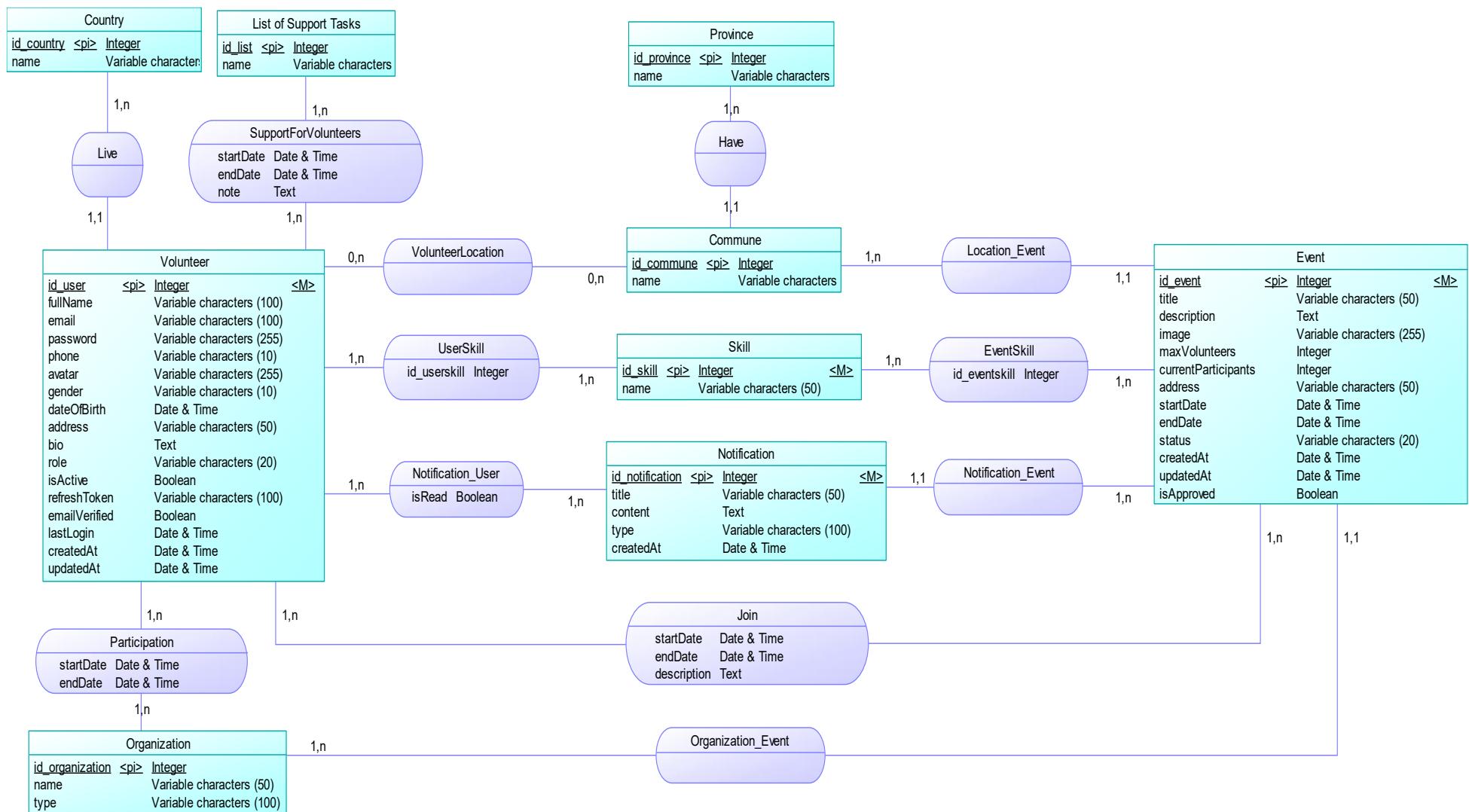
- Giao diện trực quan, dễ sử dụng;
- Thao tác được thiết kế đơn giản, giảm thiểu sai sót khi nhập dữ liệu;
- Hỗ trợ cả desktop và mobile, có thông báo đầy khi có sự kiện liên quan.

3.2. Thiết kế cơ sở dữ liệu quan hệ

3.2.1. Mô hình thực thể kết hợp (ERD - Entity Relationship Diagram)

Trang mô hình thực thể kết hợp ERD – trang sau

Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận



Hình 3.1 Mô hình thực thể kết hợp

3.2.2. Mô hình dữ liệu quan hệ

Cấu trúc các bảng:

1) Bảng COUNTRY (Quốc gia)

COUNTRY (Id_Country, Name)

2) Bảng VOLUNTEER (Tình nguyện viên)

VOLUNTEER (Id_User, Id_Country, Fullname, Email, Password, Phone, Avatar, Gender, Dateofbirth, Address, Bio, Role, Isactive, RefreshToken, Emailverified, Lastlogin, Createdat, Updatedat)

3) Bảng ORGANIZATION (Tổ chức)

ORGANIZATION (Id_Organization, Name, Type)

4) Bảng EVENT (Hoạt động / Sự kiện tình nguyện)

EVENT (Id_Event, Id_Organization, Id_Commune, Title, Description, Image, Maxvolunteers, Currentparticipants, Address, Startdate, Enddate, Status, Isapproved, Createdat, Updatedat)

5) Bảng SKILL (Kỹ năng)

SKILL (Id_Skill, Name)

6) Bảng USERSKILL (Kỹ năng của tình nguyện viên)

USERSKILL (Id_User, Id_Skill)

7) Bảng EVENTSKILL (Kỹ năng yêu cầu cho sự kiện)

EVENTSKILL (Id_Event, Id_Skill)

8) Bảng JOIN (Tham gia sự kiện)

JOIN (Id_User, Id_Event, Startdate, Enddate, Description)

9) Bảng PARTICIPATION (Tham gia tổ chức)

PARTICIPATION (Id_User, Id_Organization, Startdate, Enddate)

10) Bảng LIST_OF_SUPPORT_TASKS (Danh sách hỗ trợ)

LIST_OF_SUPPORT_TASKS (Id_List, Name)

11) Bảng SUPPORTFORVOLUNTEERS (Hỗ trợ tình nguyện viên)

SUPPORTFORVOLUNTEERS (**Id_List**, **Id_User**, Startdate, Enddate, Note)

12) Bảng NOTIFICATION (Thông báo)

NOTIFICATION (**Id_Notification**, **Id_Event**, Title, Content, Type, Createdat)

13) Bảng NOTIFICATION_USER (Thông báo - người dùng)

NOTIFICATION_USER (**Id_User**, **Id_Notification**, Isread)

14) Bảng PROVINCE (Tỉnh/Thành phố)

PROVINCE (**Id_Province**, Name)

15) Bảng COMMUNE (Quận/Huyện/Xã)

COMMUNE (**Id_Commune**, **Id_Province**, Name)

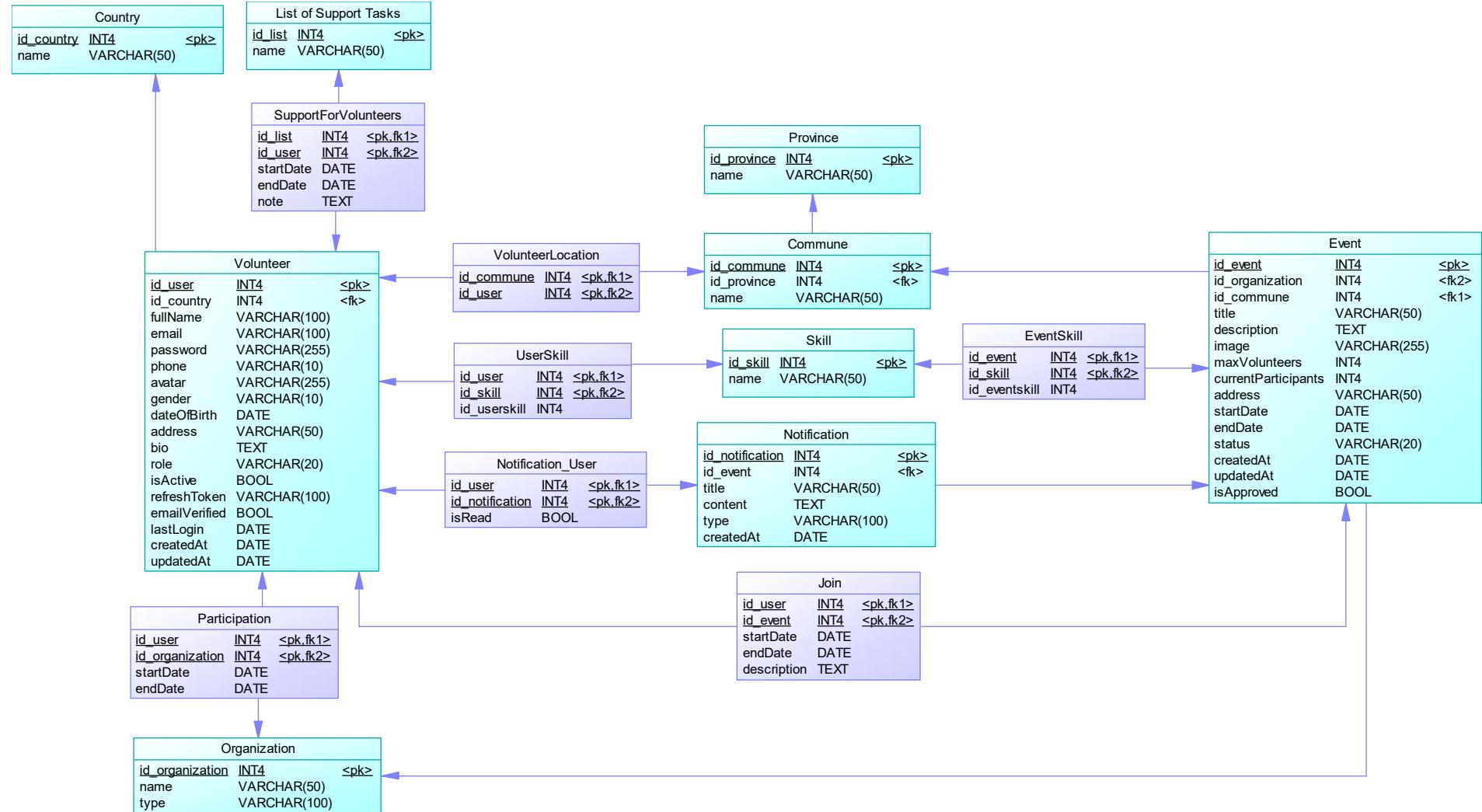
16) Bảng VOLUNTEERLOCATION (Địa bàn hoạt động của tình nguyện viên)

VOLUNTEERLOCATION (**Id_Commune**, **Id_User**)

3.2.3. Mô hình dữ liệu vật lý (Physical Data Model - PDM)

Trang mô hình dữ liệu vật lý – trang sau

Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận



Hình 3.2 Mô hình dữ liệu vật lý

3.3. Mô tả các bảng

1) Bảng COUNTRY

Bảng 3.1 Danh sách và mô tả thuộc tính cho bảng COUNTRY

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_country</u>	Int	Mã quốc gia (PK)
2	Name	Varchar(50)	Tên quốc gia

2) Bảng VOLUNTEER

Bảng 3.2 Danh sách và mô tả thuộc tính cho bảng VOLUNTEER

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_user</u>	Int	Mã tình nguyện viên (PK)
2	<u>Id_country</u>	Int	Mã quốc gia (FK)
3	Fullname	Varchar(100)	Họ tên tình nguyện viên
4	Email	Varchar(100)	Email đăng nhập
5	Password	Varchar(255)	Mật khẩu đã mã hóa
6	Phone	Varchar(10)	Số điện thoại
7	Avatar	Varchar(255)	Ảnh đại diện
8	Gender	Varchar(10)	Giới tính
9	Dateofbirth	Date	Ngày sinh
10	Address	Varchar(50)	Địa chỉ
11	Bio	Text	Giới thiệu bản thân

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
12	Role	Varchar(20)	Vai trò (MEMBER, ORG, ADMIN, SUPER_ADMIN)
13	Isactive	Boolean	Trạng thái hoạt động
14	RefreshToken	Varchar(100)	Token làm mới
15	Emailverified	Boolean	Trạng thái xác thực email
16	Lastlogin	Date	Lần đăng nhập gần nhất
17	Createdat	Date	Ngày tạo tài khoản
18	Updatedat	Date	Ngày cập nhật thông tin

3) Bảng ORGANIZATION

Bảng 3.3 Danh sách và mô tả thuộc tính cho bảng ORGANIZATION

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id organization</u>	Int	Mã tổ chức (PK)
2	Name	Varchar(50)	Tên tổ chức
3	Type	Varchar(100)	Loại hình tổ chức

4) Bảng EVENT

Bảng 3.4 Danh sách và mô tả thuộc tính cho bảng EVENT

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id event</u>	Int	Mã sự kiện (PK)
2	<u>Id organization</u>	Int	Mã tổ chức (FK)

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
3	<u>Id_commune</u>	Int	Mã địa phương (FK)
4	Title	Varchar(50)	Tên hoạt động
5	Description	Text	Mô tả chi tiết
6	Image	Varchar(255)	Hình ảnh minh họa
7	Maxvolunteers	Int	Số TNV tối đa
8	Currentparticipants	Int	Số TNV hiện tại
9	Address	Varchar(50)	Địa điểm
10	Startdate	Date	Ngày bắt đầu
11	Enddate	Date	Ngày kết thúc
12	Status	Varchar(20)	Trạng thái sự kiện
13	Isapproved	Boolean	Đã được duyệt hay chưa
14	Createdat	Date	Ngày tạo
15	Updatedat	Date	Ngày cập nhật

5) Bảng SKILL

Bảng 3.5 Danh sách và mô tả thuộc tính cho bảng SKILL

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_skill</u>	Int	Mã kỹ năng (PK)
2	Name	Varchar(50)	Tên kỹ năng

6) Bảng USERSKILL

Bảng 3.6 Danh sách và mô tả thuộc tính cho bảng USERSKILL

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id user</u>	Int	Mã tình nguyện viên (FK)
2	<u>Id skill</u>	Int	Mã kỹ năng (FK)

7) Bảng EVENTSKILL

Bảng 3.7 Danh sách và mô tả thuộc tính cho bảng EVENTSKILL

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id event</u>	Int	Mã sự kiện (FK)
2	<u>Id skill</u>	Int	Mã kỹ năng (FK)

8) Bảng JOIN

Bảng 3.8 Danh sách và mô tả thuộc tính cho bảng JOIN

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id user</u>	Int	Mã tình nguyện viên (FK)
2	<u>Id event</u>	Int	Mã sự kiện (FK)
3	Startdate	Date	Ngày bắt đầu tham gia
4	Enddate	Date	Ngày kết thúc
5	Description	Text	Ghi chú

9) Bảng PARTICIPATION

Bảng 3.9 Danh sách và mô tả thuộc tính cho bảng PARTICIPATION

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_user</u>	Int	Mã tình nguyện viên (FK)
2	<u>Id_organization</u>	Int	Mã tổ chức (FK)
3	Startdate	Date	Ngày bắt đầu
4	Enddate	Date	Ngày kết thúc

10) Bảng LIST_OF_SUPPORT_TASKS

Bảng 3.10 Danh sách và mô tả thuộc tính cho bảng LIST_OF_SUPPORT_TASKS

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_list</u>	Int	Mã hỗ trợ (PK)
2	Name	Varchar(50)	Tên loại hỗ trợ

11) Bảng SUPPORTFORVOLUNTEERS

Bảng 3.11 Danh sách và mô tả thuộc tính cho bảng SUPPORTFORVOLUNTEERS

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_list</u>	Int	Mã hỗ trợ (FK)
2	<u>Id_user</u>	Int	Mã tình nguyện viên (FK)
3	Startdate	Date	Ngày bắt đầu hỗ trợ
4	Enddate	Date	Ngày kết thúc
5	Note	Text	Ghi chú

12) Bảng NOTIFICATION

Bảng 3.12 Danh sách và mô tả thuộc tính cho bảng NOTIFICATION

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id notification</u>	Int	Mã thông báo (PK)
2	<u>Id event</u>	Int	Mã sự kiện (FK)
3	Title	Varchar(50)	Tiêu đề
4	Content	Text	Nội dung
5	Type	Varchar(100)	Loại thông báo
6	Createdat	Date	Ngày tạo

13) Bảng NOTIFICATION_USER

Bảng 3.13 Danh sách và mô tả thuộc tính cho bảng NOTIFICATION_USER

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id user</u>	Int	Mã tình nguyện viên (FK)
2	<u>Id notification</u>	Int	Mã thông báo (FK)
3	Isread	Boolean	Trạng thái đã đọc

14) Bảng PROVINCE

Bảng 3.14 Danh sách và mô tả thuộc tính cho bảng PROVINCE

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id province</u>	Int	Mã tỉnh/thành phố (PK)
2	Name	Varchar(50)	Tên tỉnh/thành phố

15) Bảng COMMUNE

Bảng 3.15 Danh sách và mô tả thuộc tính cho bảng COMMUNE

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_commune</u>	Int	Mã địa phương (PK)
2	<u>Id_province</u>	Int	Mã tỉnh/thành phố (FK)
3	Name	Varchar(50)	Tên địa phương

16) Bảng VOLUNTEERLOCATION

Bảng 3.16 Danh sách và mô tả thuộc tính cho bảng VOLUNTEERLOCATION

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<u>Id_commune</u>	Int	Mã địa phương (FK)
2	<u>Id_user</u>	Int	Mã tình nguyện viên (FK)

3.4. Các khóa ràng buộc

Khóa chính (Primary Key)

Bảng 3.17 Bảng các khóa chính

STT	Tên bảng	Khóa chính
1	Country	Id_country
2	Volunteer	Id_user
3	Organization	Id_organization
4	Event	Id_event
5	Skill	Id_skill
6	Province	Id_province

STT	Tên bảng	Khóa chính
7	Commune	Id_commune
8	List_of_support_tasks	Id_list
9	Notification	Id_notification
10	Userskill	(Id_user, Id_skill)
11	Eventskill	(Id_event, Id_skill)
12	Join	(Id_user, Id_event)
13	Participation	(Id_user, Id_organization)
14	Supportforvolunteers	(Id_list, Id_user)
15	Notification_user	(Id_user, Id_notification)
16	Volunteerlocation	(Id_commune, Id_user)

Khóa ngoại (Foreign Key)

Bảng 3.18 Bảng các khóa ngoại

STT	Bảng	Khóa ngoại	Tham chiếu đến
1	Volunteer	Id_country	Country (Id_country)
2	Event	Id_organization	Organization (Id_organization)
3	Event	Id_commune	Commune (Id_commune)
4	Commune	Id_province	Province (Id_province)

Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận

STT	Bảng	Khóa ngoại	Tham chiếu đến
5	Eventskill	Id_event	Event (Id_event)
6	Eventskill	Id_skill	Skill (Id_skill)
7	Userskill	Id_user	Volunteer (Id_user)
8	Userskill	Id_skill	Skill (Id_skill)
9	Join	Id_user	Volunteer (Id_user)
10	Join	Id_event	Event (Id_event)
11	Participation	Id_user	Volunteer (Id_user)
12	Participation	Id_organization	Organization (Id_organization)
13	Supportfor Volunteers	Id_list	List_of_support_tasks (Id_list)
14	Supportfor Volunteers	Id_user	Volunteer (Id_user)
15	Notification	Id_event	Event (Id_event)
16	Notification_ User	Id_user	Volunteer (Id_user)
17	Notification_ User	Id_notification	Notification (Id_notification)
18	Volunteer Location	Id_commune	Commune (Id_commune)

STT	Bảng	Khóa ngoại	Tham chiếu đến
19	Volunteer Location	Id_user	Volunteer (Id_user)

3.5. Kiến trúc hệ thống

3.5.1. Sơ đồ Use case

Các tác nhân chính:

- **Tình nguyện viên (Volunteer):** người Việt Nam hoặc người nước ngoài tham gia hoạt động tình nguyện.
- **Tổ chức (Organization):** đơn vị phi lợi nhuận.
- **Quản trị viên (Admin):** người quản lý hệ thống, kiểm duyệt và giám sát hoạt động.

Các usecase chính:

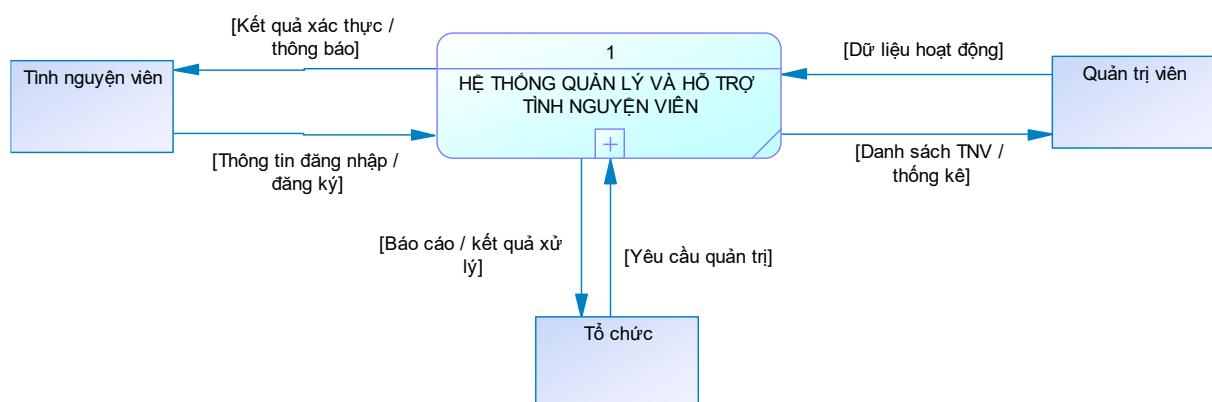
- 1) Đăng ký tài khoản.
- 2) Đăng nhập.
- 3) Quản lý sự kiện tình nguyện (tạo, cập nhật sự kiện).
- 4) Xem danh sách tình nguyện viên tham gia.
- 5) Gửi thông báo.
- 6) Xem thống kê.
- 7) Phân công và theo dõi hỗ trợ cho tình nguyện viên.
- 8) Quản lý thông tin tài khoản cá nhân.
- 9) Xem danh sách hoạt động.
- 10) Xem chi tiết hoạt động.
- 11) Đăng ký tham gia hoạt động.
- 12) Hủy tham gia hoạt động.
- 13) Xem lịch sử tham gia.
- 14) Nhận thông báo
- 15) Quản trị hệ thống và duyệt sự kiện.



Hình 3.3 Sơ đồ Use case

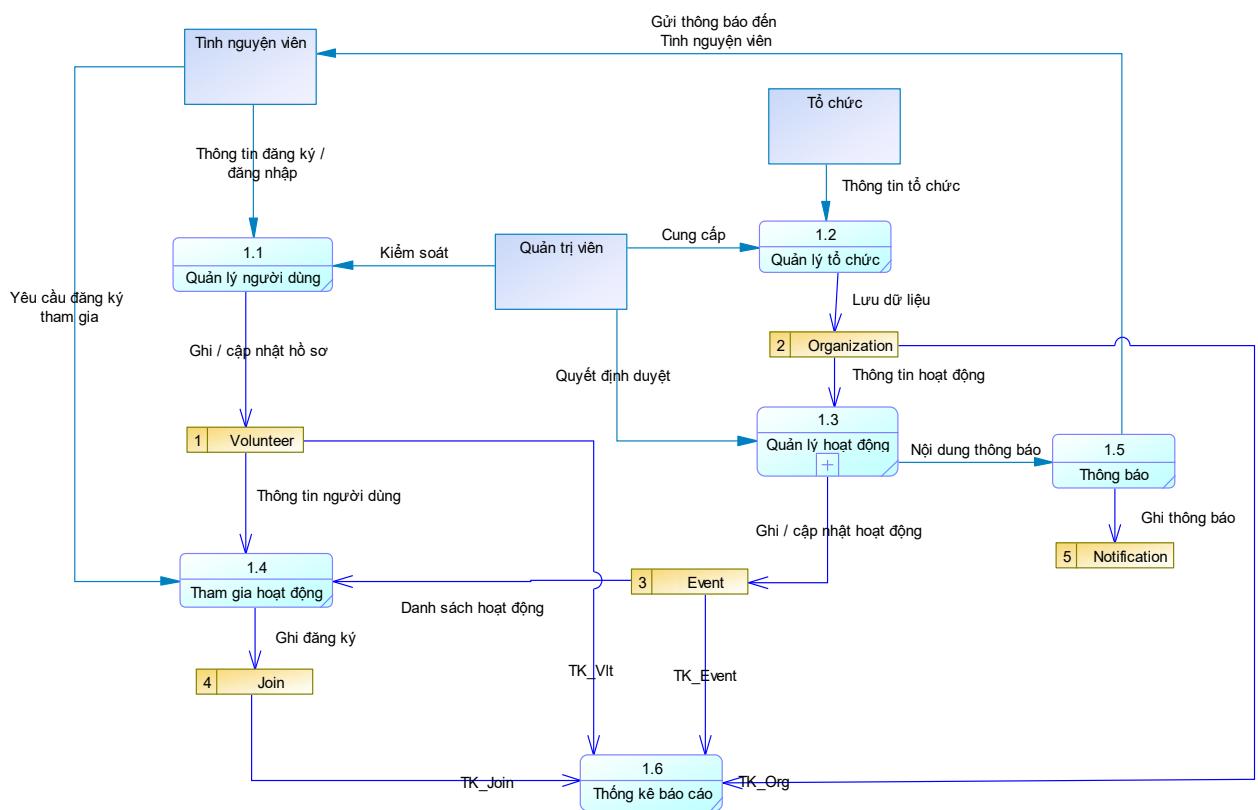
3.5.2. Sơ đồ DFD

DFD mức 0 - Sơ đồ ngũ cảnh



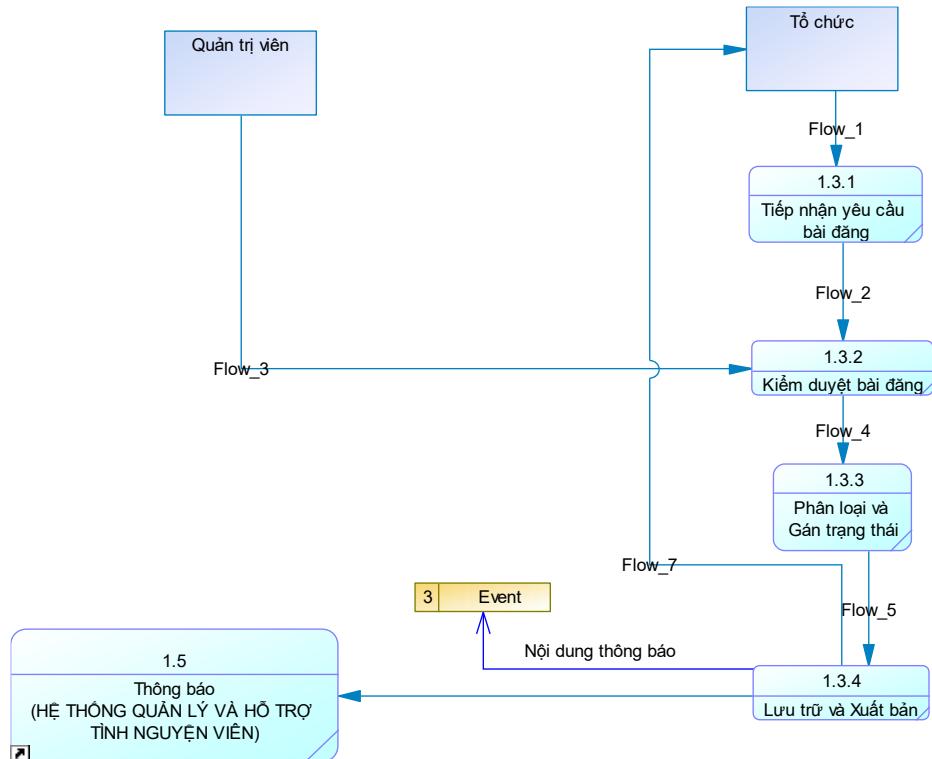
Hình 3.4 Sơ đồ DFD mức 0

DFD mức 1 – Phân rã hệ thống



Hình 3.5 Sơ đồ DFD mức 1

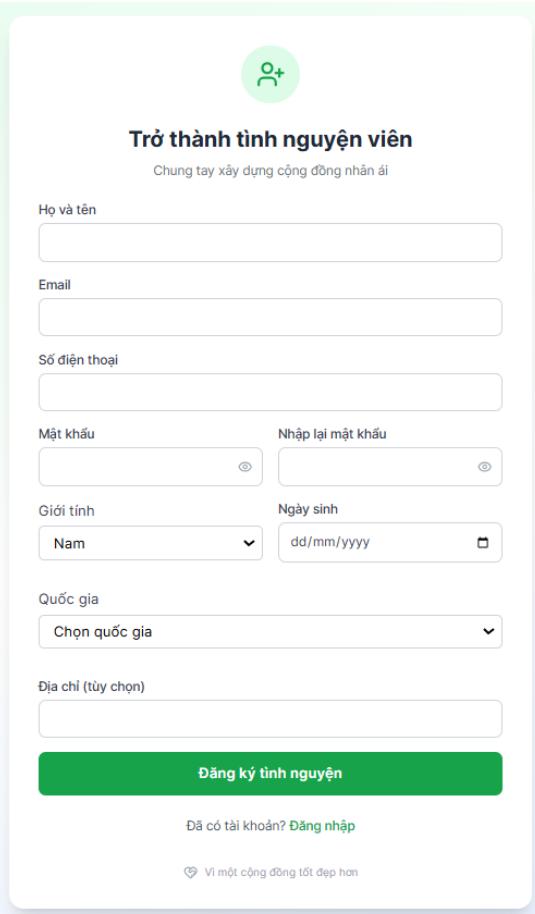
DFD mức 2 – Phân rã quy trình Quản lý hoạt động



Hình 3.6 Sơ đồ DFD mức 2

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Giao diện Đăng ký

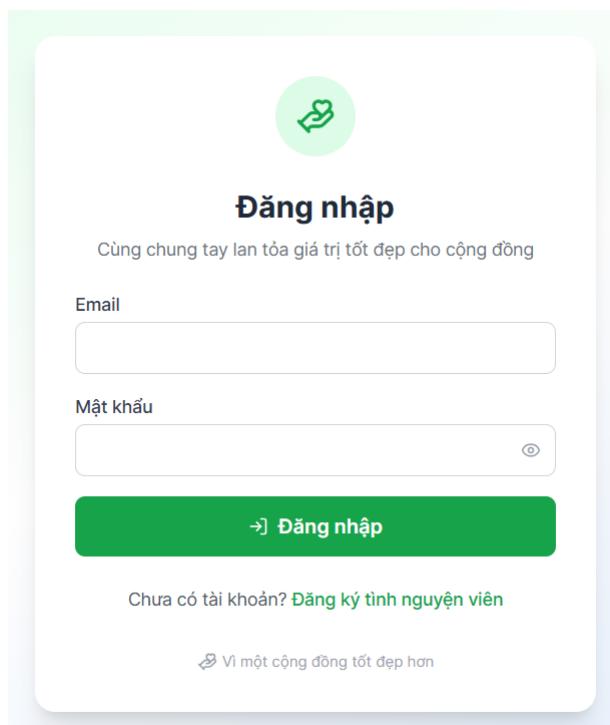


Hình 4.1 Giao diện trang đăng ký

Giao diện đăng ký của hệ thống được thiết kế theo cấu trúc biểu mẫu một cột trung tâm trên nền trắng thanh lịch, giúp tối ưu hóa sự tập trung và giảm thiểu sai sót khi nhập liệu. Phía trên cùng là tiêu đề “Trở thành tình nguyện viên” giúp nâng cao tinh thần cho tình nguyện viên hơn, giúp người dùng xác định rõ mục đích tham gia cộng đồng.

Biểu mẫu bao gồm các trường dữ liệu chi tiết: “Họ và tên”, “Email”, “Số điện thoại”, “Mật khẩu”, “Giới tính”, “Ngày sinh”, “Quốc gia” và “Địa chỉ”. Trường mật khẩu được tích hợp tính năng ẩn/hiện ký tự qua biểu tượng hình mắt, đảm bảo tính riêng tư khi thao tác nơi công cộng. Người dùng điền thông tin theo trình tự logic từ trên xuống dưới. Nút “Đăng ký tình nguyện” màu xanh lá đậm là điểm nhấn hành động chính; khi kích hoạt, hệ thống sẽ kiểm tra tính hợp lệ của dữ liệu, tiến hành mã hóa mật khẩu bằng thư viện bcrypt và phản hồi bằng thông báo thành công hoặc lỗi định dạng cụ thể. Liên kết “Đã có tài khoản? Đăng nhập” ở cuối trang cho phép chuyển đổi nhanh chóng giữa các trạng thái, tạo nên một quy trình khép kín và thân thiện.

4.2. Giao diện Đăng nhập



Hình 4.2 Giao diện trang đăng nhập

Giao diện Đăng nhập: Người dùng truy cập vào hệ thống thông qua Email và Mật khẩu. Sau khi xác thực thành công, hệ thống sẽ cấp một mã JWT (JSON Web Token) để duy trì phiên làm việc không trạng thái.

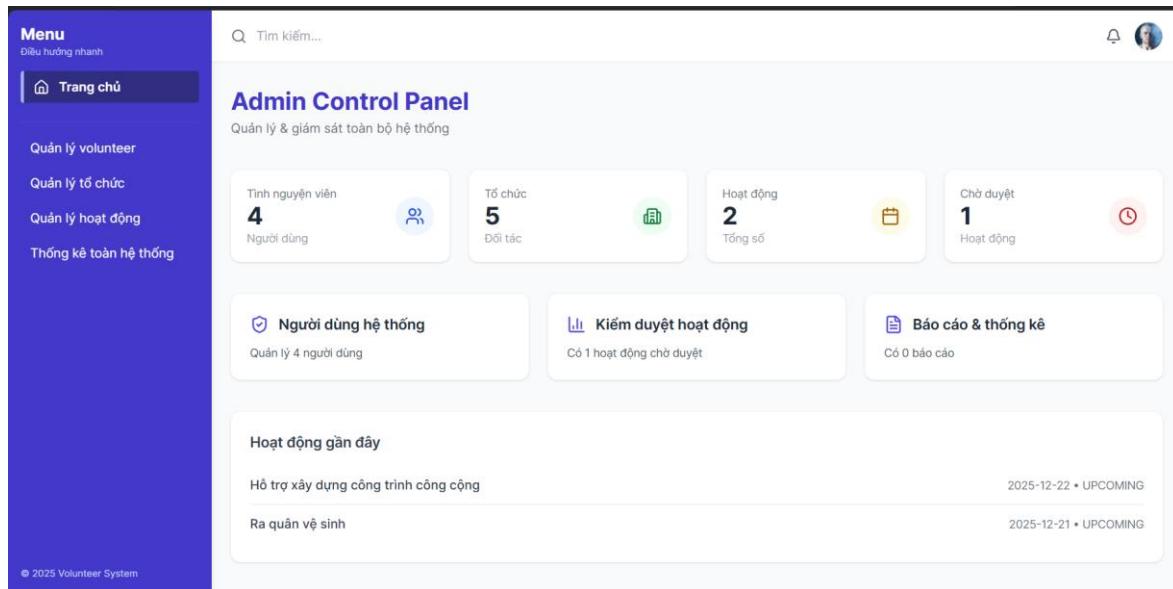
Giao diện Đăng nhập hệ thống: Tương tự như giao diện đăng ký, màn hình đăng nhập duy trì sự đồng bộ về mặt nhận diện thương hiệu với biểu tượng trái tim xanh ở vị trí trung tâm. Giao diện tập trung tối đa vào hai trường dữ liệu cốt lõi là “Email” và “Mật khẩu”. Cấu trúc này giúp tinh giản hóa quy trình truy cập, giảm bớt rào cản về mặt thao tác cho người dùng.

Hệ thống cung cấp phản hồi tức thì thông qua nút “Đăng nhập” màu xanh lá. Tại bước này, hệ thống sẽ thực hiện đối soát dữ liệu với kho lưu trữ; nếu thông tin chính xác, một mã JWT sẽ được khởi tạo để cấp quyền truy cập vào các tính năng bên trong Dashboard. Dòng chữ “Vì một cộng đồng tốt đẹp hơn” ở dưới cùng không chỉ là lời kết mà còn là slogan xuyên suốt, nhắc nhở người dùng về mục đích nhân văn của ứng dụng.

4.3. Quản trị viên

4.3.1. Bảng điều khiển trung tâm

Giao diện Dashboard cung cấp cái nhìn tổng quan về quản lý của toàn hệ thống.



Hình 4.3 Giao diện Dashboard quản trị hệ thống

Giao diện Dashboard dành cho quản trị viên được tổ chức theo bố cục hiện đại: thanh điều hướng (Sidebar) cố định bên trái và vùng hiển thị số liệu động bên phải. Sidebar chứa các danh mục quản lý như: Dashboard, Người dùng, Tổ chức, Hoạt động và Thống kê, cho phép Admin chuyển đổi phân hệ chỉ với một cú nhấp chuột.

Vùng nội dung chính hiển thị các thẻ thống kê tổng quát (Statistics Cards) về tổng số tình nguyện viên, tổ chức đối tác và các chiến dịch đang triển khai. Mỗi thẻ được minh họa bằng biểu tượng và màu sắc riêng biệt, giúp nhận diện nhanh các chỉ số quan trọng.

Hệ thống phản hồi dữ liệu theo thời gian thực, đảm bảo Admin luôn nắm bắt được thực tế của toàn hệ thống để đưa ra các quyết định điều phối kịp thời.

4.3.2. Quản lý Tổ chức

The screenshot shows the 'Quản lý Tổ chức' (Organization Management) section of the Admin dashboard. On the left sidebar, there's a 'Menu' with 'Trang chủ' (Home) and 'Dashboard'. The main area has a search bar and a 'Thêm' (Add) button. A table lists organizations with columns for ID, TỔ CHỨC, THÀNH VIÊN, and LOẠI. The data includes:

ID	TỔ CHỨC	THÀNH VIÊN	LOẠI
#1	Tổ chức của Chính phủ	0	NGO
#2	Tổ chức Cộng đồng	0	COMMUNITY
#4	Tổ chức của Bệnh viện	0	HOSPITAL
#6	Tổ chức của Trường học các cấp	0	SCHOOL
#11	Tổ chức Môi trường	0	ENVIRONMENTAL

Hình 4.4 Giao diện Quản lý tổ chức của Admin

Giao diện này hiển thị danh sách tất cả các tổ chức phi lợi nhuận đã đăng ký tham gia nền tảng. Bảng dữ liệu cung cấp thông tin về ID tổ chức, tên, loại hình (NGO, School, Hospital...) và số lượng thành viên. Các nhãn trạng thái được tô màu sắc trực quan giúp Admin dễ dàng phân loại.

Người quản trị có thể kiểm soát tính chính xác của thông tin tổ chức và thực hiện các điều chỉnh cần thiết. Khi có tổ chức mới đăng ký, hệ thống sẽ tự động cập nhật danh sách này để Admin theo dõi và phê duyệt thông tin định danh.

4.3.3. Quản lý Người dùng

The screenshot shows the 'Quản lý Volunteer' (Volunteer Management) section of the Admin dashboard. The sidebar includes 'Menu', 'Trang chủ', and 'Dashboard'. The main area features a search bar and a 'Thêm' button. A table lists volunteers with columns for ID, Tên, Email, Role, and Active status. The data includes:

ID	Tên	Email	Role	Active
#10	Lê Thanh Huy	abc@gmail.com	MEMBER	✓ Active
#15	Tôi là Admin	admin@vlt.com	ADMIN	✓ Active
#16	Organization	org@vlt.com	ORG	✓ Active
#3	Nguyen Phuc Vinh	vinh24122004@gmail.com	MEMBER	✓ Active

Hình 4.5 Giao diện Quản lý tình nguyện viên của Admin

Giao diện quản lý người dùng liệt kê chi tiết các tài khoản trong hệ thống dưới dạng bảng dữ liệu. Phần nội dung chính bao gồm tiêu đề và các thẻ thông kê về số lượng người dùng đang hoạt động hoặc bị vô hiệu hóa.

Bảng danh sách hiển thị các cột: Tên, Email, Vai trò và Trạng thái. Tại mỗi dòng, Admin có thể thực hiện thao tác khóa hoặc phân quyền lại thông qua các nút chức năng ở cột cuối. Hệ thống phản hồi tức thời bằng cách cập nhật bảng dữ liệu hoặc hiển thị thông báo xác nhận thao tác.

4.3.4. Kiểm duyệt Hoạt động

The screenshot shows the 'Quản lý Hoạt động' (Activity Management) page. On the left is a sidebar with 'Menu' and 'Trang chủ'. The main area has a search bar and a title 'Quản lý Hoạt động'. Below it is a table with two rows of activity data:

ID	Tiêu đề	TỔ CHỨC	ĐỊA CHỈ	TRẠNG THÁI	NGÀY BẮT ĐẦU	HÀNH ĐỘNG
#67	Ra quân vệ sinh	Tổ chức của Chính phủ	Phường Bến Tre, Vĩnh Long	Chưa duyệt	22/12/2025	<button>Duyệt</button> <button>Xem</button>
#68	Hỗ trợ xây dựng công trình công cộng	Tổ chức của Chính phủ	Phường Long Đức, Vĩnh Long	Sắp diễn ra	22/12/2025	<button>Duyệt</button> <button>Xem</button>

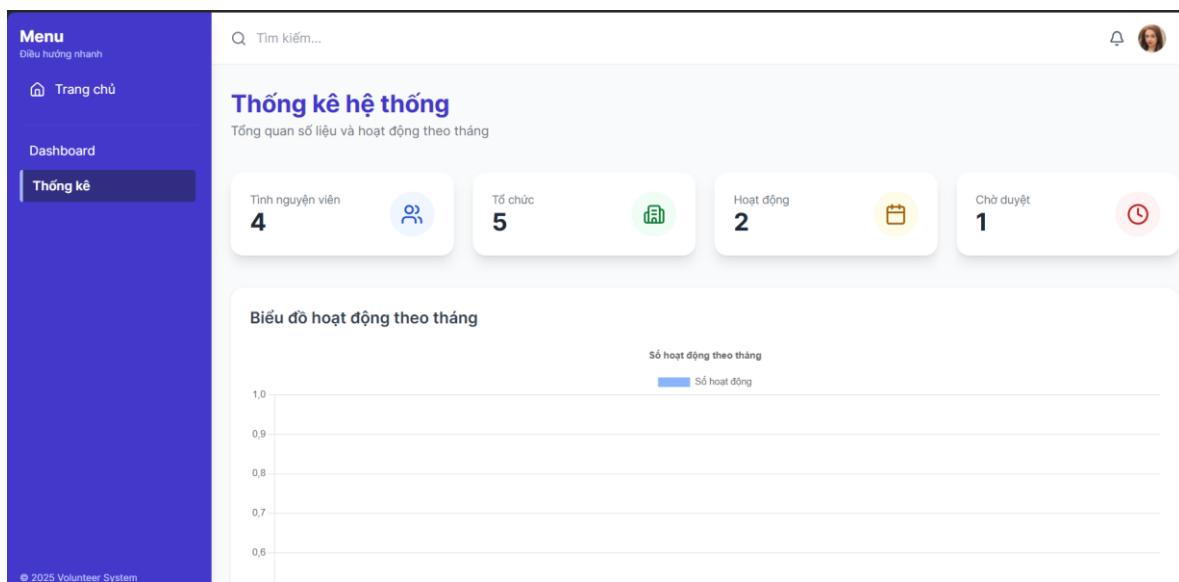
Hình 4.4 Giao diện Quản lý hoạt động của Admin

Đây là giao diện quan trọng để đảm bảo chất lượng nội dung trên nền tảng. Admin xem danh sách các hoạt động do các Tổ chức gửi lên dưới dạng hàng đợi chờ duyệt.

Mỗi dòng thông tin cho biết tổ chức đăng bài, tên hoạt động và trạng thái hiện tại (Chờ duyệt). Admin có nút hành động để xem chi tiết nội dung và quyết định "Phê duyệt" hoặc "Từ chối".

Khi nút Duyệt được nhấn, hệ thống phản hồi bằng cách chuyển trạng thái sang "Published" và cho phép hoạt động hiển thị công khai trên trang của Tình nguyện viên.

4.3.5. Thống kê hệ thống



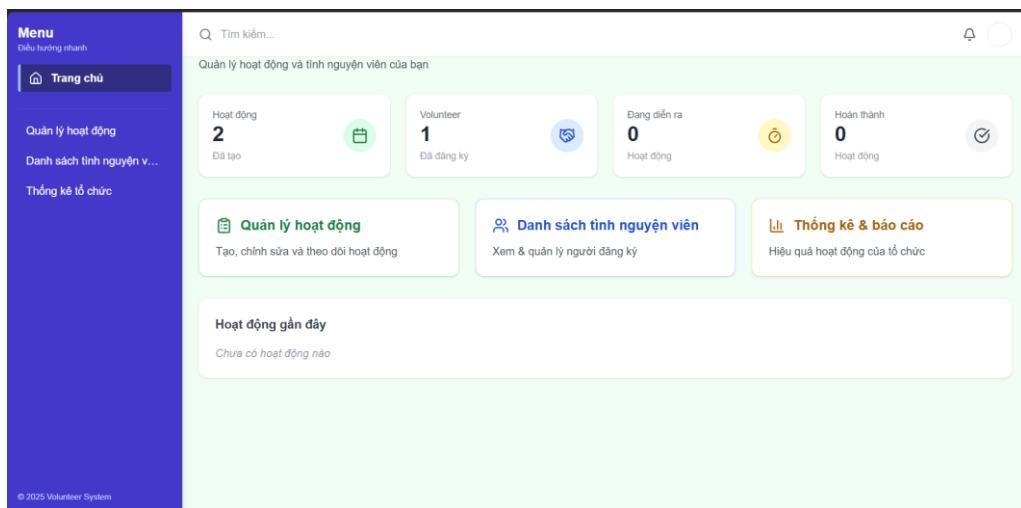
Hình 4.5 Giao diện Thống kê của Admin

Giao diện báo cáo chuyên sâu cung cấp các biểu đồ trực quan hóa dữ liệu của toàn bộ hệ thống. Phần trung tâm hiển thị biểu đồ theo dõi lượng “Hoạt động theo tháng” và mức độ tham gia của cộng đồng.

Người dùng tiếp nhận thông tin trực quan thông qua các hình khối và số liệu, dễ dàng đánh giá xu hướng phát triển của nền tảng. Khi Admin thay đổi bộ lọc thời gian, hệ thống phản hồi tức thời bằng cách cập nhật lại các chỉ số trên biểu đồ, phục vụ cho việc lập báo cáo tổng kết.

4.4. Phân hệ Tổ chức

4.4.1. Dashboard của Tổ chức



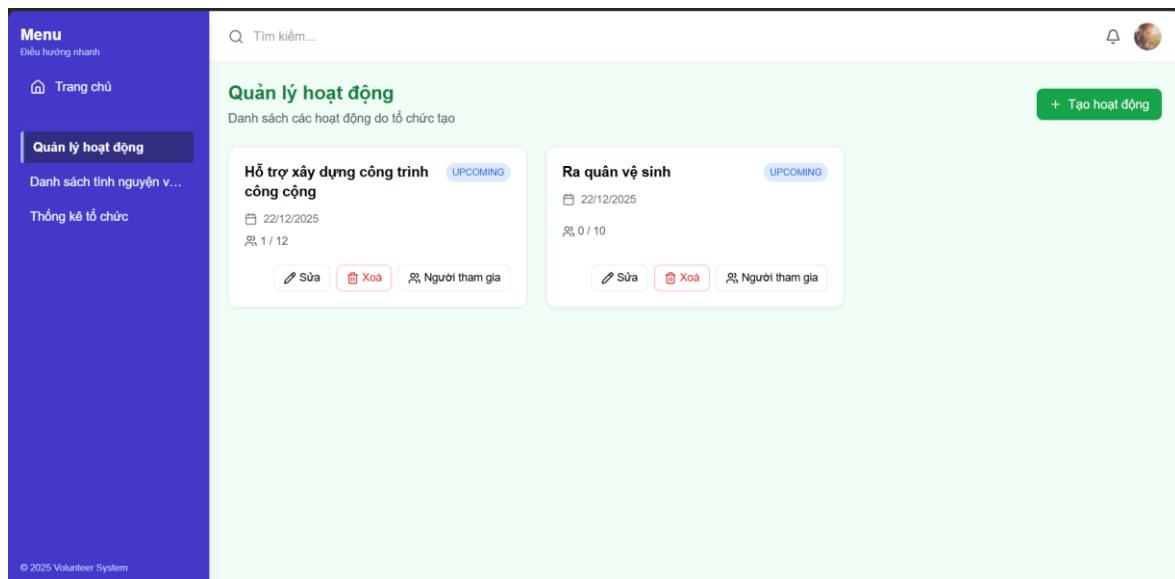
Hình 4.6 Giao diện Dashboard Tổ chức

Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận

Giao diện này cung cấp cái nhìn tổng quan về hiệu quả của các chiến dịch mà tổ chức đang triển khai.

Các thẻ thông kê hiển thị số hoạt động đã tạo và tổng số tình nguyện viên đã đăng ký tham gia. Bố cục được tối ưu để đại diện tổ chức nắm bắt nhanh các con số tăng trưởng riêng biệt của mình.

4.4.2. Quản lý Hoạt động của Tổ chức

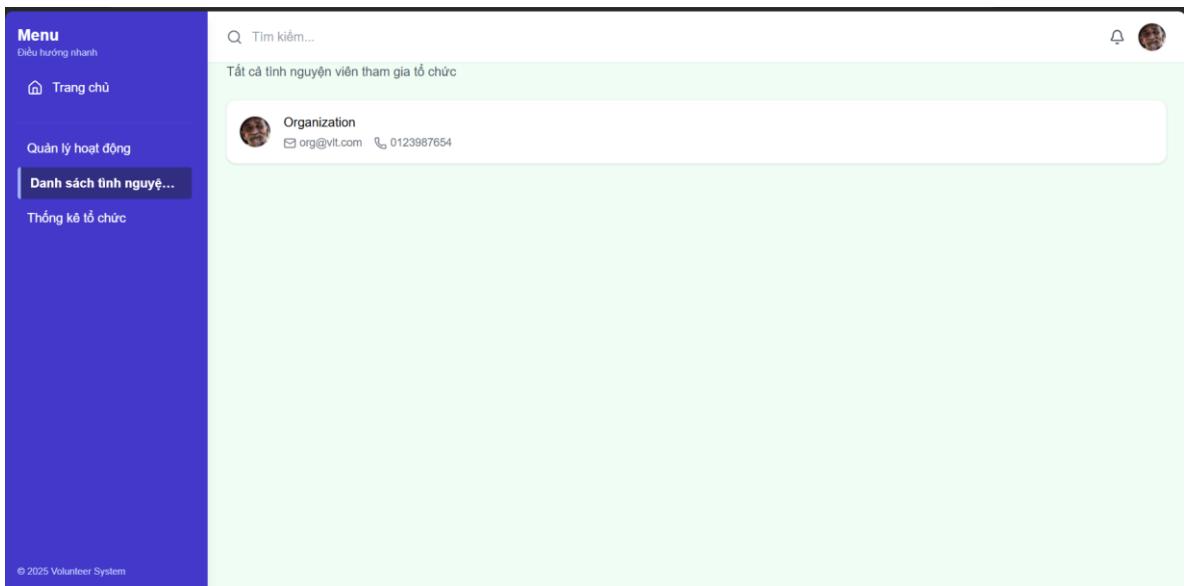


Hình 4.7 Giao diện Quản lý hoạt động của Tổ chức

Giao diện hiển thị các chiến dịch dưới dạng thẻ (Card view) bao gồm: tên hoạt động, ngày diễn ra và trạng thái (Upcoming/Completed). Mỗi thẻ tích hợp thanh tiến độ cho thấy số người đã đăng ký trên tổng chỉ tiêu. Tổ chức có thể nhấn vào thẻ để chỉnh sửa nội dung hoặc xem chi tiết danh sách người tham gia

Mỗi thẻ hiển thị tên hoạt động, ngày diễn ra, trạng thái (Upcoming/Completed) và số lượng người đã đăng ký trên tổng số cần tuyển (ví dụ: 1/12). Tổ chức có thể thực hiện các thao tác: Sửa nội dung, Xóa hoạt động hoặc Xem danh sách người tham gia cụ thể của hoạt động đó.

4.4.3. Danh sách Tình nguyện viên tham gia

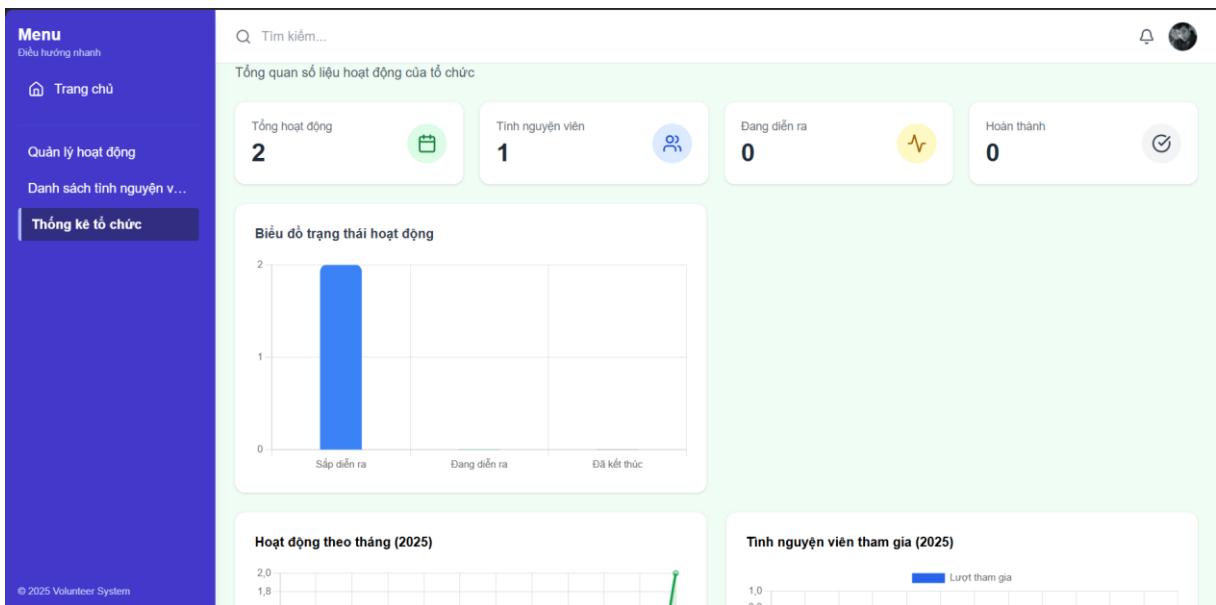


Hình 4.8 Giao diện Danh sách tình nguyện viên của tổ chức

Danh sách tình nguyện viên của tổ chức Giao diện liệt kê chi tiết hồ sơ các cá nhân đã đăng ký tham gia hoạt động cụ thể. Thông tin bao gồm Họ tên, Email và Số điện thoại liên lạc.

Đây là công cụ đắc lực giúp tổ chức thực hiện việc điều phối, liên lạc và quản lý nhân sự cho từng dự án thiện nguyện.

4.4.4. Báo cáo thống kê Tổ chức



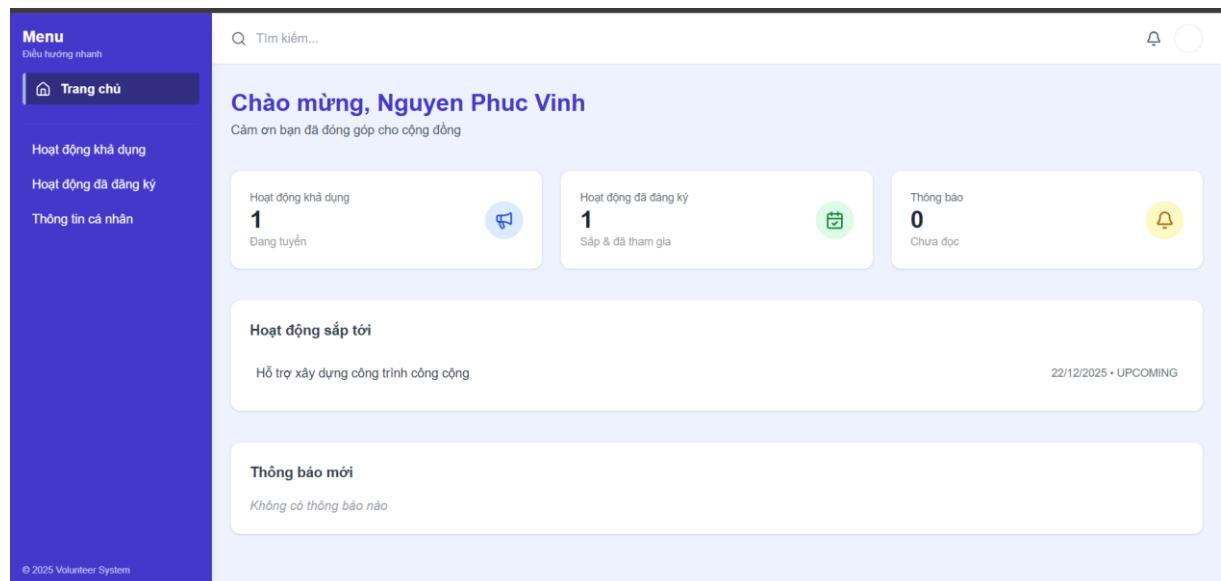
Hình 4.9 Giao diện Thống kê của Tổ chức

Các biểu đồ phân tích hiệu quả hoạt động của riêng tổ chức. Bao gồm biểu đồ cột thể hiện trạng thái hoạt động (Sắp diễn ra, Đang diễn ra) và biểu đồ đường theo dõi lượng tình nguyện viên tham gia theo thời gian, giúp tổ chức đánh giá được sức hút của các chiến dịch.

4.5. Phân hệ Tình nguyện viên

Giao diện phía người dùng cuối, tập trung vào sự đơn giản, dễ sử dụng để khuyến khích đăng ký tham gia.

4.5.1. Trang chủ cá nhân



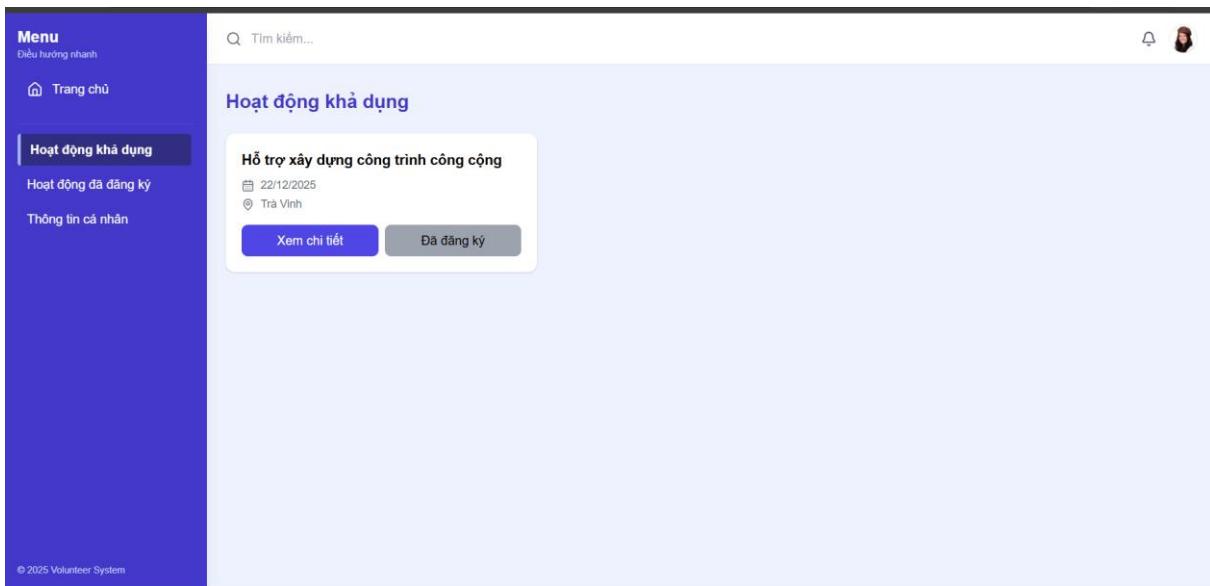
Hình 4.10 Giao diện Dashboard của Tình nguyện viên

Mô tả: Màn hình chào mừng cá nhân hóa với tên người dùng.

Chức năng:

- Hiển thị tóm tắt: Số hoạt động khả dụng đang tuyển, số hoạt động bản thân đã đăng ký.
- Hoạt động sắp tới: Nhắc nhở người dùng về sự kiện gần nhất họ cần tham gia (kèm thời gian và trạng thái UPCOMING).
- Thông báo: Khu vực hiển thị các thông báo mới từ hệ thống hoặc từ các tổ chức mà họ theo dõi.

4.5.2. Danh sách hoạt động khả dụng



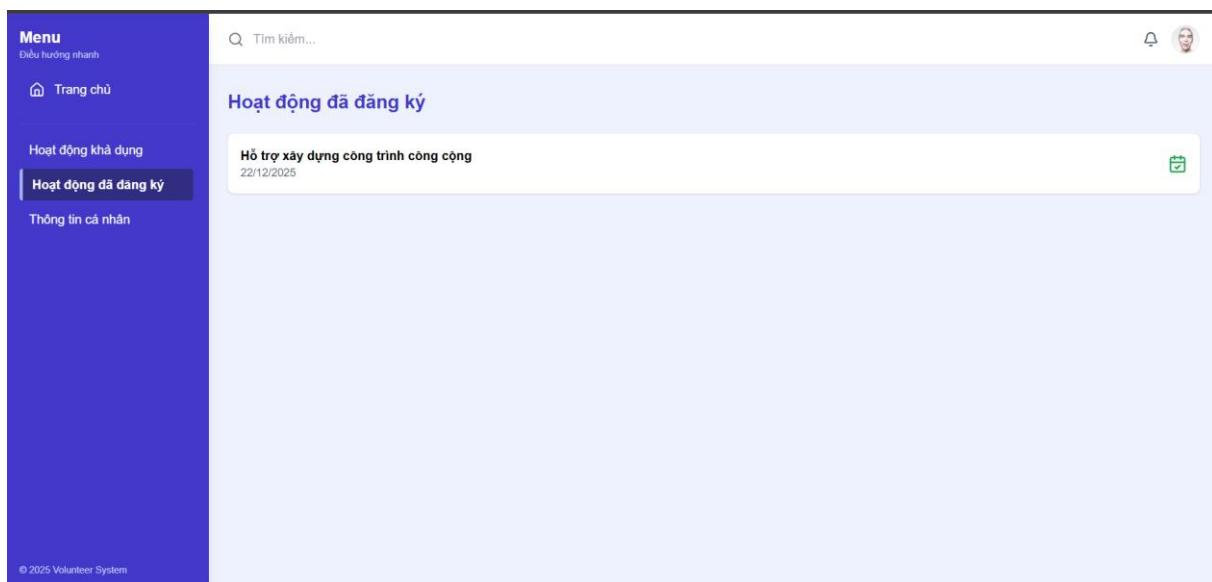
Hình 4.11 Giao diện Danh sách hoạt động khả dụng

Mô tả: Đây là giao diện chính nơi tình nguyện viên tìm kiếm các cơ hội đóng góp. Các hoạt động được trình bày dưới dạng danh sách hoặc thẻ (Card) gọn gàng.

Chức năng:

- Hiển thị thông tin tóm tắt: Tên hoạt động (ví dụ: “Hỗ trợ xây dựng công trình công cộng”), thời gian diễn ra và địa điểm (Trà Vinh).
- Tương tác: Người dùng có nút “Xem chi tiết” để đọc nội dung cụ thể trước khi quyết định. Nút trạng thái (ví dụ: “Đã đăng ký”) giúp người dùng biết được mình đã tham gia hoạt động nào, tránh đăng ký trùng lặp.

4.5.3. Quản lý hoạt động đã đăng ký



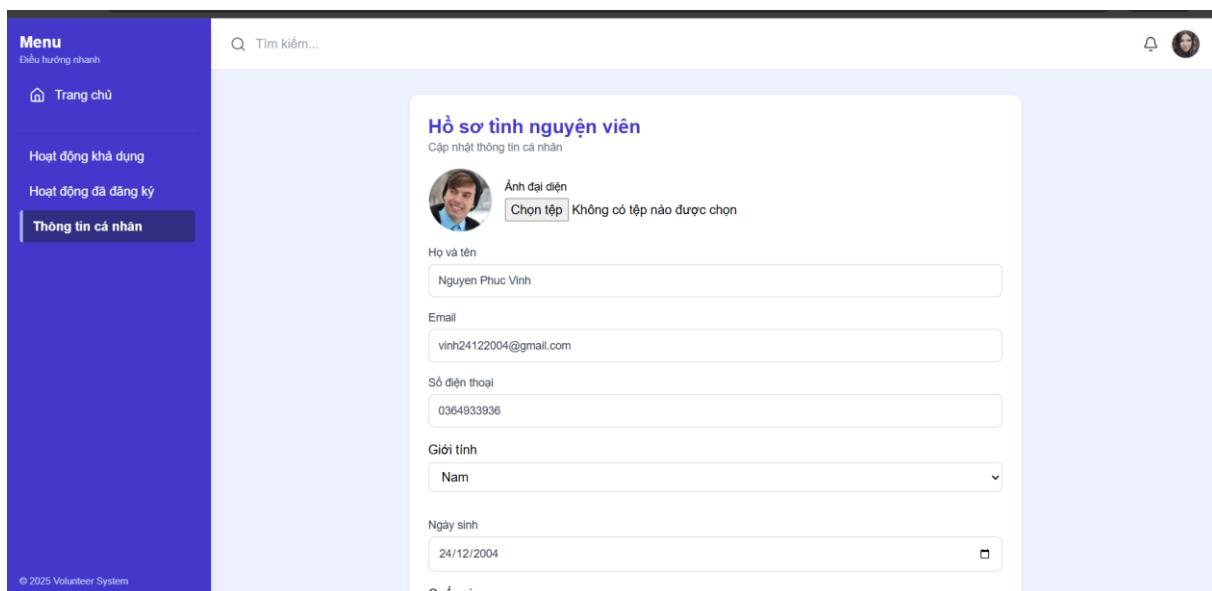
Hình 4.12 Giao diện Danh sách hoạt động đã đăng ký

Mô tả: Trang quản lý lịch trình cá nhân của tình nguyện viên.

Chức năng:

- Lưu trữ danh sách lịch sử các hoạt động mà người dùng đã nhấn nút tham gia.
- Giúp tình nguyện viên dễ dàng theo dõi lịch trình sắp tới (ngày 22/12/2025) để sắp xếp thời gian, đảm bảo cam kết với tổ chức.

4.5.4. Cập nhật hồ sơ cá nhân



Hình 4.13 Giao diện tùy chỉnh thông tin cá nhân

Mô tả: Form nhập liệu cho phép người dùng chỉnh sửa thông tin cá nhân trong hệ thống.

Chức năng:

- Hỗ trợ tải lên ảnh đại diện (Avatar) để tăng tính nhận diện.
- Các trường thông tin quan trọng: Họ và tên, Email (thường bị khóa hoặc chỉ đọc để định danh), Số điện thoại liên lạc, Giới tính và Ngày sinh.
- Đây là dữ liệu quan trọng để các Tổ chức liên hệ với tình nguyện viên khi cần thiết hoặc dùng để in giấy chứng nhận sau này.

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Trong phạm vi đề tài “Xây dựng hệ thống quản lý và hỗ trợ tình nguyện viên cho tổ chức phi lợi nhuận”, em đã tiến hành nghiên cứu, phân tích yêu cầu và xây dựng một hệ thống web hỗ trợ công tác quản lý tình nguyện viên và các hoạt động tình nguyện, đặc biệt hướng đến nhóm tình nguyện viên, trong đó có tình nguyện viên nước ngoài tham gia hoạt động tại Việt Nam.

Hệ thống được thiết kế dựa trên các nghiệp vụ thực tế của tổ chức phi lợi nhuận, bao gồm quản lý thông tin tình nguyện viên, quản lý tổ chức, hoạt động/sự kiện tình nguyện, đăng ký tham gia hoạt động, theo dõi quá trình tham gia, quản lý hỗ trợ dành cho tình nguyện viên và gửi thông báo liên quan. Việc ứng dụng các công nghệ hiện đại như React, Node.js, PostgreSQL và Prisma ORM đã giúp hệ thống đảm bảo tính logic, dễ sử dụng và thuận tiện trong việc mở rộng sau này.

Kết quả đạt được của đề tài gồm:

- Phân tích và xây dựng được mô hình nghiệp vụ của hệ thống quản lý tình nguyện viên cho tổ chức phi lợi nhuận;
- Thiết kế các sơ đồ phân tích hệ thống như Use Case Diagram, DFD các mức nhằm mô tả rõ ràng luồng xử lý và mối quan hệ giữa các thành phần trong hệ thống;
- Xây dựng giao diện web thân thiện, dễ sử dụng cho các đối tượng người dùng như tình nguyện viên, tổ chức và quản trị viên;
- Thiết kế và cài đặt cơ sở dữ liệu PostgreSQL phục vụ việc lưu trữ thông tin tình nguyện viên, tổ chức, sự kiện, kỹ năng, hỗ trợ và thông báo;
- Hệ thống hoạt động ổn định, đáp ứng được các yêu cầu cơ bản đã đề ra trong phạm vi đề án;
- Qua quá trình thực hiện, đề tài đã giúp em củng cố kiến thức về phân tích - thiết kế hệ thống thông tin, lập trình web và làm quen với quy trình xây dựng một ứng dụng quản lý hoàn chỉnh.

5.2. Hạn chế

Mặc dù đã đạt được những kết quả nhất định, hệ thống vẫn còn một số hạn chế do giới hạn về thời gian, kinh nghiệm và phạm vi của đồ án, cụ thể:

- Chưa triển khai các chức năng thống kê các loại, báo cáo nâng cao phục vụ việc phân tích dữ liệu tình nguyện viên và hiệu quả hoạt động;
- Chưa tích hợp các cơ chế bảo mật ở mức cao như xác thực đa yếu tố hoặc phân quyền chi tiết theo từng nghiệp vụ;
- Giao diện và trải nghiệm người dùng chưa được tối ưu cho nhiều thiết bị khác nhau;
- Hệ thống mới chỉ dùng ở mức mô phỏng cho một tổ chức phi lợi nhuận giả định, chưa được triển khai và đánh giá trong môi trường thực tế.

5.3. Hướng phát triển

Trong thời gian tới, hệ thống có thể tiếp tục được mở rộng và hoàn thiện theo các hướng sau:

- Mở rộng các chức năng nghiệp vụ như nhiều loại thống kê, báo cáo tổng hợp theo thời gian, theo sự kiện hoặc theo tình nguyện viên.
- Tích hợp hệ thống thông báo tự động qua email hoặc ứng dụng di động để hỗ trợ tình nguyện viên, đặc biệt là tình nguyện viên nước ngoài.
- Nâng cao mức độ bảo mật và phân quyền người dùng, đảm bảo an toàn dữ liệu và quyền riêng tư.
- Phát triển hệ thống theo hướng đa nền tảng, hỗ trợ tốt trên thiết bị di động.
- Triển khai thử nghiệm hệ thống trong môi trường thực tế tại các tổ chức phi lợi nhuận để đánh giá và cải tiến.
- Việc tiếp tục phát triển theo các hướng trên sẽ giúp hệ thống quản lý và hỗ trợ tình nguyện viên ngày càng hoàn thiện, đáp ứng tốt hơn nhu cầu thực tế của các tổ chức phi lợi nhuận và góp phần nâng cao hiệu quả các hoạt động tình nguyện.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] TopDev, “React là gì? Kiến thức căn bản và lộ trình học ReactJS”, TopDev. Truy cập: 11 Tháng Mười-Một 2025. [Online]. Available at: <https://topdev.vn/blog/react-la-gi-lo-trinh/>
- [2] “React là gì? Hướng dẫn chi tiết cho người mới bắt đầu”. Truy cập: 11 Tháng Mười-Một 2025. [Online]. Available at: <https://cellphones.com.vn/sforum/react-la-gi>
- [3] “Tailwind CSS Là Gì? Ưu Nhược điểm Và Ứng Dụng Thực Tế”. Truy cập: 11 Tháng Mười-Một 2025. [Online]. Available at: <https://www.careerlink.vn/cam-nang-viec-lam/tu-van-nghe-nghiep/tailwind-css-la-gi>
- [4] A. Mendoza, *Learn Tailwind CSS Quickly: The Complete Beginner's Handbook to Responsive Design, Clean Layouts, and Practical Web Projects*. Amazon Digital Services LLC - Kdp, 2025.
- [5] TopDev, “Node.js là gì? Tổng hợp kiến thức NodeJS thật chi tiết”, TopDev. Truy cập: 11 Tháng Mười-Một 2025. [Online]. Available at: <https://topdev.vn/blog/node-js-la-gi/>
- [6] E. Brown, *Web Development with Node and Express*, O'Reilly Media, 2014.
- [7] “PostgreSQL: About”. Truy cập: 11 Tháng Mười-Một 2025. [Online]. Available at: <https://www.postgresql.org/about/>
- [8] Prisma Team, *Prisma ORM Documentation*, 2023.
- [9] N. Provos and D. Mazieres, “A Future-Adaptable Password Scheme” Proc. USENIX Annual Technical Conference, 1999.

PHỤ LỤC

HƯỚNG DẪN CÀI ĐẶT VÀ SỬ DỤNG HỆ THỐNG

A. Yêu cầu môi trường

Trước khi cài đặt, máy tính cần có các phần mềm sau:

- Node.js (phiên bản $\geq 18.x$)
- PostgreSQL
- pgAdmin 4 (để quản lý cơ sở dữ liệu)

B. Clone dự án từ GitHub

Thực hiện clone dự án về máy:

```
git clone <link-repository>
```

Sau khi clone xong, di chuyển vào thư mục dự án:

```
cd <ten-thu-muc-du-an>
```

Cấu trúc tổng quát của dự án gồm:

- frontend: Giao diện người dùng
- backend: Xử lý nghiệp vụ và API
- prisma: Quản lý schema và migration cơ sở dữ liệu

C. Cài đặt Backend

Cài đặt thư viện

Di chuyển vào thư mục backend:

```
cd backend  
npm install
```

Cấu hình biến môi trường

Tạo file .env trong thư mục backend và cấu hình chuỗi kết nối CSDL:

```
DATABASE_URL="postgresql://username:password@localhost:5432/te  
n_database"
```

D. Cài đặt cơ sở dữ liệu

Tạo database

- Mở pgAdmin 4
- Tạo database mới (ví dụ: ten_database)
- Khởi tạo schema bằng Prisma

Tại thư mục backend, chạy lệnh:

```
npx prisma migrate deploy
```

Hoặc nếu phát triển cục bộ:

```
npx prisma migrate dev
```

Kiểm tra dữ liệu bằng Prisma Studio (tùy chọn):

```
npx prisma studio
```

E. Chạy Backend

Khởi động server backend:

```
npm run dev
```

Backend sẽ chạy tại port:

```
http://localhost:5000
```

F. Cài đặt và chạy Frontend

Mở terminal mới, di chuyển vào thư mục frontend:

```
cd frontend  
npm install  
npm run dev
```

Frontend sẽ chạy tại:

```
http://localhost:5173
```