

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN



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

# XÂY DỰNG HỆ THỐNG QUẢN LÝ DỰ ÁN VÀ CỘNG TÁC NHÓM

*Giảng viên hướng dẫn:*

ThS.GVC. Phan Thị Phương Nam

*Sinh viên thực hiện:*

Họ tên: Kim Thạch Minh Trí

MSSV: 110122191

*Vĩnh Long, tháng 12 năm 2025*

**TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ  
KHOA CÔNG NGHỆ THÔNG TIN**



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

# **XÂY DỰNG HỆ THỐNG QUẢN LÝ DỰ ÁN VÀ CỘNG TÁC NHÓM**

*Giảng viên hướng dẫn:*

ThS.GVC. Phan Thị Phương Nam

*Sinh viên thực hiện:*

Họ tên: Kim Thạch Minh Trí

MSSV: 110122191

*Vĩnh Long, tháng 12 năm 2025*

[illegible]

Sinh viên thực hiện: Kim Thạch Minh Trí

## NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

[illegible]

Vĩnh Long, ngày ..... tháng ..... năm .....

**Thành viên hội đồng**  
(Ký tên và ghi rõ họ tên)

## LỜI CẢM ƠN

Lời đầu tiên, em xin bày tỏ lòng biết ơn sâu sắc đến Cô Phan Thị Phương Nam, Giảng viên Khoa Công nghệ Thông tin, Trường Kỹ thuật và Công nghệ, Trường Đại học Trà Vinh. Cô đã tận tình hướng dẫn, chỉ bảo và truyền đạt những kiến thức quý báu, cũng như đã dành thời gian nhận xét, góp ý và định hướng cho em trong suốt quá trình thực hiện đồ án chuyên ngành này.

Em xin chân thành cảm ơn Ban Giám hiệu Trường Đại học Trà Vinh, quý thầy cô trong Khoa Công nghệ Thông tin, Trường Kỹ thuật và Công nghệ, cùng các phòng ban nhà trường đã tạo điều kiện thuận lợi cho em trong suốt quá trình học tập và nghiên cứu. Nhờ có sự hỗ trợ từ phía nhà trường, em đã có cơ hội tiếp cận với môi trường học tập hiện đại, chuyên nghiệp, từ đó tích lũy được nhiều kiến thức và kỹ năng cần thiết.

Em cũng xin gửi lời cảm ơn chân thành đến gia đình và bạn bè đã luôn động viên, khích lệ và hỗ trợ em cả về vật chất lẫn tinh thần. Sự ủng hộ từ gia đình và bạn bè là nguồn động lực to lớn giúp em vượt qua những khó khăn, thử thách trong quá trình thực hiện đồ án.

Trong quá trình nghiên cứu đề tài, do kiến thức còn hạn chế nên đồ án của em chắc chắn không tránh khỏi những thiếu sót. Em rất mong nhận được những ý kiến đóng góp quý báu từ quý thầy cô và các bạn để em có thể hoàn thiện đồ án của mình hơn nữa, cũng như rút ra những bài học kinh nghiệm cho các nghiên cứu trong tương lai.

Em xin trân trọng 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 .....	viii
DANH MỤC BẢNG BIỂU .....	x
DANH MỤC TỪ VIẾT TẮT.....	xi
TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH .....	xii
MỞ ĐẦU.....	xiii
CHƯƠNG 1 TỔNG QUAN.....	1
1.1. Mở đầu.....	1
1.2. Tổng quan đề tài.....	1
1.3. Mục tiêu và nhiệm vụ .....	2
1.3.1 Mục tiêu.....	2
1.3.2 Nhiệm vụ .....	2
1.3.3 Phạm vi nghiên cứu .....	3
1.4. Phương pháp nghiên cứu .....	3
CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT .....	4
2.1. Quy trình nghiệp vụ về hệ thống quản lý dự án và cộng tác nhóm .....	4
2.1.1 Tổng quan quy trình nghiệp vụ .....	4
2.1.2 Các vai trò trong hệ thống.....	4
2.1.3 Quy trình khởi tạo và quản lý dự án .....	4
2.1.4 Quy trình phân công và thực hiện công việc.....	5
2.1.5 Quy trình theo dõi tiến độ và cộng tác nhóm .....	5
2.1.6 Quy trình kết thúc dự án.....	6
2.1.7 Tổng kết quy trình nghiệp vụ .....	6
2.2. Giới thiệu về PHP.....	6
2.2.1 Khái niệm .....	6
2.2.2 Ứng dụng của ngôn ngữ PHP trong lập trình.....	7
2.2.3 Ưu điểm.....	8
2.2.4 Nhược điểm .....	8

---

2.2.5 Ứng dụng PHP trong dự án .....	8
2.3. Giới thiệu về MySQL .....	9
2.3.1 Khái niệm .....	9
2.3.2 Lịch sử hình thành và phát triển của MySQL .....	10
2.3.3 Đặc điểm của MySQL .....	10
2.3.4 Cách hoạt động .....	11
2.3.5 Ứng dụng MySQL trong dự án .....	12
2.4. Giới thiệu về XAMPP .....	12
2.4.1 XAMPP là gì .....	12
2.4.2 Công dụng chính của XAMPP .....	14
2.4.3 Ứng dụng XAMPP trong dự án .....	14
2.5. Giới thiệu về Tailwind CSS .....	15
2.5.1 Tổng quan về Tailwind CSS .....	15
2.5.2 Mô hình Utility-First trong Tailwind CSS .....	15
2.5.3 Kiến trúc và cơ chế hoạt động của Tailwind CSS .....	16
2.5.4 Hệ thống Responsive và Dark Mode .....	16
2.5.5 Ưu điểm của Tailwind CSS trong thực tế phát triển dự án .....	17
2.5.6 Ứng dụng Tailwind CSS trong dự án .....	17
2.6. Giới thiệu về JavaScript framework Alpine.js .....	17
2.6.1 Tổng quan về Alpine.js .....	17
2.6.2 Triết lý thiết kế và mô hình hoạt động của Alpine.js .....	18
2.6.3 Các thành phần cốt lõi trong Alpine.js .....	18
2.6.4 Cơ chế quản lý trạng thái và phản ứng (Reactivity) .....	18
2.6.5 Ưu điểm và hạn chế của Alpine.js .....	19
2.6.6 Ứng dụng Alpine.js trong dự án .....	19
2.7. Giới thiệu về mô hình MVC .....	20
2.7.1 Tổng quan về mô hình MVC .....	20
2.7.2 Các thành phần trong mô hình MVC .....	20
2.7.3 Ưu điểm và hạn chế của mô hình MVC .....	22
2.7.4 Ứng dụng mô hình MVC trong phát triển web .....	22
2.7.5 Ứng dụng mô hình MVC trong dự án .....	22
CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU .....	23

---

---

3.1. Đặc tả các yêu cầu.....	23
3.1.1 Yêu cầu chức năng.....	23
3.1.2 Yêu cầu phi chức năng.....	24
3.2. Thiết kế cơ sở dữ liệu quan hệ.....	24
3.2.1 Mô hình thực thể kết hợp (ERD - Entity Relationship Diagram) .....	24
3.2.2 Mô hình dữ liệu quan hệ .....	25
3.2.3 Mô hình dữ liệu vật lý (Physical Data Model - PDM) .....	26
3.2.4 Mô tả các bảng.....	26
3.3. Kiến trúc hệ thống .....	30
3.3.1 Sơ đồ Use case.....	30
3.3.2 Đặc tả Use case đăng nhập.....	31
3.3.3 Đặc tả Use case Tạo dự án .....	32
3.3.4 Đặc tả Use case Tạo công việc .....	33
3.3.5 Luồng nghiệp vụ hệ thống.....	34
CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU.....	38
4.1. Kết quả thực hiện và đánh giá hệ thống .....	38
4.1.1 Kết quả thực hiện hệ thống .....	38
4.1.2 Đánh giá mức độ đáp ứng yêu cầu .....	38
4.2. Giao diện trang Đăng ký.....	40
4.3. Giao diện trang Đăng nhập .....	41
4.4. Giao diện các trang của Admin.....	42
4.4.1 Giao diện trang Admin Dashboard.....	42
4.4.2 Giao diện trang Admin Quản lý người dùng.....	43
4.4.3 Giao diện chức năng Thêm người dùng.....	44
4.4.4 Giao diện trang Admin Quản lý dự án.....	45
4.4.5 Giao diện trang Admin Quản lý Công việc.....	46
4.4.6 Giao diện trang Admin Quản lý tài liệu.....	47
4.4.7 Giao diện trang Admin Báo cáo và Thống kê.....	48
4.5. Giao diện các trang của Manager.....	49
4.5.1 Giao diện trang Manager Dashboard .....	49
4.5.2 Giao diện trang Manager Quản lý dự án.....	50
4.5.3 Giao diện chức năng Tạo dự án.....	51

---



---

4.5.4 Giao diện chức năng Tạo công việc .....	52
4.5.5 Giao diện chức năng Thêm thành viên vào dự án .....	53
4.5.6 Giao diện trang Manager Báo cáo .....	54
4.6. Giao diện trang Chi tiết dự án.....	55
4.7. Giao diện trang Chi tiết Công việc.....	58
4.8. Giao diện chức năng Bình luận.....	59
CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	60
5.1. Kết luận.....	60
5.2. Hạn chế của đề tài .....	60
5.3. Hướng phát triển .....	60
DANH MỤC TÀI LIỆU THAM KHẢO.....	62

---

## DANH MỤC HÌNH ẢNH

Hình 2.1 PHP là gì.....	6
Hình 2.2 MySQL là gì .....	9
Hình 2.3 Cách hoạt động của MySQL .....	12
Hình 2.4 XAMPP là gì.....	13
Hình 2.5 Tổng quan về Tailwind CSS.....	15
Hình 2.6 Tổng quan về Alpine.js .....	18
Hình 2.7 Tổng quan về mô hình MVC .....	20
Hình 3.1. Mô hình thực thể kết hợp (ERD - Entity Relationship Diagram).....	25
Hình 3.2 Mô hình dữ liệu vật lý (Physical Data Model - PDM).....	26
Hình 3.3. Sơ đồ Use case .....	31
Hình 3.4 Luồng nghiệp vụ tổng quát.....	34
Hình 3.5 Luồng nghiệp vụ quản lý dự án .....	35
Hình 3.6 Luồng nghiệp vụ quản lý công việc .....	36
Hình 4.1 Giao diện trang Đăng ký .....	40
Hình 4.2 Giao diện trang Đăng nhập.....	41
Hình 4.3 Giao diện trang Admin Dashboard .....	42
Hình 4.4 Giao diện trang Admin Quản lý người dùng.....	43
Hình 4.5 Giao diện chức năng Thêm người dùng .....	44
Hình 4.6 Giao diện trang Admin Quản lý dự án .....	45
Hình 4.7 Giao diện trang Admin Quản lý dự án .....	46
Hình 4.8 Giao diện trang Admin Quản lý tài liệu .....	47
Hình 4.9 Giao diện trang Admin Báo cáo và Thống kê .....	48
Hình 4.10 Giao diện trang Manager Dashboard .....	49
Hình 4.11 Giao diện trang Manager Quản lý dự án .....	50
Hình 4.12 Giao diện chức năng Tạo dự án .....	51
Hình 4.13 Giao diện chức năng Tạo công việc .....	52
Hình 4.14 Giao diện chức năng Thêm thành viên vào dự án .....	53
Hình 4.15 Giao diện trang Manager Báo cáo .....	54
Hình 4.16 Giao diện tổng quan Dự án.....	55
Hình 4.17 Giao diện bảng Công việc của Dự án.....	56
Hình 4.18 Giao diện Tài liệu của Dự án.....	56

---

Hình 4.19 Giao diện Thành viên của Dự án .....	57
Hình 4.20 Giao diện chi tiết Công việc .....	58
Hình 4.21 Giao diện Bình luận của Công việc .....	59

---

## DANH MỤC BẢNG BIỂU

Bảng 3.1 Mô tả bảng Người dùng .....	26
Bảng 3.2. Mô tả bảng Dự án .....	27
Bảng 3.3. Mô tả bảng Công việc .....	27
Bảng 3.4. Mô tả bảng Bình luận.....	28
Bảng 3.5. Mô tả bảng Tập đính kèm .....	28
Bảng 3.6. Mô tả bảng Thông báo .....	29
Bảng 3.7. Mô tả bảng Lịch sử hoạt động.....	29
Bảng 3.8 Mô tả bảng Nhóm .....	30
Bảng 3.9 Mô tả bảng Vai trò.....	30
Bảng 3.10 Đặc tả Use case đăng nhập.....	31
Bảng 3.11 Đặc tả Use case Quản lý dự án.....	32
Bảng 3.12 Đặc tả Use case Quản lý công việc.....	33

---

## DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Diễn giải đầy đủ
CNTT	Công nghệ thông tin
CSDL	Cơ sở dữ liệu
DB	Database
DBMS	Database Management System
MVC	Model – View – Controller
UML	Unified Modeling Language
ERD	Entity Relationship Diagram
UI /UX	User Interface /User Experience
HTTP	HyperText Transfer Protocol
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
PHP	Hypertext Preprocessor
SQL	Structured Query Language
CRUD	Create – Read – Update – Delete
IDE	Integrated Development Environment
XAMPP	Apache – MySQL – PHP – Perl
BPMN	Business Process Model and Notation

## **TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH**

Trong bối cảnh chuyển đổi số ngày càng phát triển, nhu cầu ứng dụng công nghệ thông tin vào quản lý dự án và cộng tác nhóm trở nên cấp thiết đối với các tổ chức, doanh nghiệp cũng như các nhóm làm việc học tập và nghiên cứu. Việc quản lý công việc bằng các công cụ rời rạc như bảng tính, email hoặc ứng dụng nhắn tin thường dẫn đến tình trạng thông tin phân tán, khó theo dõi tiến độ và thiếu tính minh bạch trong quá trình phối hợp.

Xuất phát từ thực tiễn đó, đồ án chuyên ngành này tập trung nghiên cứu và xây dựng một hệ thống quản lý dự án và cộng tác nhóm trên nền tảng web nhằm hỗ trợ người dùng quản lý dự án, phân công công việc, theo dõi tiến độ và trao đổi thông tin một cách tập trung và hiệu quả. Hệ thống hướng đến đối tượng là các nhóm làm việc nhỏ và vừa, đặc biệt là nhóm sinh viên và doanh nghiệp SME, với các chức năng cốt lõi, dễ sử dụng và phù hợp với điều kiện triển khai thực tế.

Về mặt kỹ thuật, hệ thống được thiết kế theo mô hình kiến trúc MVC nhằm tách biệt rõ ràng giữa giao diện, xử lý nghiệp vụ và dữ liệu, qua đó nâng cao khả năng bảo trì và mở rộng. Cơ sở dữ liệu được xây dựng theo mô hình quan hệ, đảm bảo tính toàn vẹn và nhất quán của dữ liệu. Backend của hệ thống được phát triển bằng ngôn ngữ PHP kết hợp với hệ quản trị cơ sở dữ liệu MySQL, trong khi giao diện người dùng được thiết kế theo hướng trực quan, thân thiện, đáp ứng nhu cầu sử dụng thực tế.

Kết quả của đồ án là một hệ thống quản lý dự án và cộng tác nhóm hoạt động ổn định, đáp ứng được các yêu cầu chức năng đề ra như quản lý người dùng, dự án, công việc và trao đổi thông tin trong nhóm. Mặc dù còn tồn tại một số hạn chế về tính năng nâng cao và khả năng mở rộng quy mô lớn, hệ thống đã chứng minh được tính khả thi và là nền tảng để tiếp tục phát triển trong tương lai.

## MỞ ĐẦU

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của công nghệ thông tin và xu hướng chuyển đổi số, nhu cầu ứng dụng các hệ thống phần mềm vào quản lý và điều hành công việc ngày càng trở nên phổ biến. Đặc biệt, trong lĩnh vực quản lý dự án và cộng tác nhóm, việc tổ chức công việc khoa học, theo dõi tiến độ và đảm bảo sự phối hợp hiệu quả giữa các thành viên đóng vai trò quan trọng trong việc nâng cao năng suất và chất lượng công việc.

Trên thực tế, nhiều tổ chức và nhóm làm việc hiện nay vẫn quản lý dự án bằng các phương pháp thủ công hoặc sử dụng các công cụ rời rạc như bảng tính, email và các ứng dụng nhắn tin. Cách tiếp cận này thường dẫn đến tình trạng thông tin bị phân tán, khó kiểm soát tiến độ tổng thể, thiếu tính minh bạch và dễ phát sinh sai sót trong quá trình phối hợp. Mặc dù đã có nhiều hệ thống quản lý dự án chuyên nghiệp được phát triển, song không phải giải pháp nào cũng phù hợp với các nhóm nhỏ, doanh nghiệp vừa và nhỏ hoặc môi trường học tập do hạn chế về chi phí, tính phức tạp hoặc khả năng tùy biến.

Xuất phát từ những vấn đề nêu trên, việc nghiên cứu và xây dựng một hệ thống quản lý dự án và cộng tác nhóm trên nền tảng web, đáp ứng các chức năng cốt lõi, dễ sử dụng và phù hợp với điều kiện triển khai thực tế là một yêu cầu mang tính cấp thiết. Hệ thống không chỉ hỗ trợ quản lý công việc một cách tập trung và hiệu quả mà còn góp phần nâng cao khả năng phối hợp, chia sẻ thông tin và theo dõi tiến độ trong nhóm làm việc.

Thông qua việc thực hiện đồ án chuyên ngành này, người thực hiện có cơ hội vận dụng các kiến thức đã học về phân tích và thiết kế hệ thống thông tin, lập trình web và quản lý cơ sở dữ liệu để xây dựng một sản phẩm phần mềm hoàn chỉnh. Đồng thời, kết quả của đồ án cũng là cơ sở để đánh giá khả năng áp dụng kiến thức lý thuyết vào thực tiễn và định hướng phát triển các hệ thống quản lý dự án trong tương lai..

## CHƯƠNG 1 TỔNG QUAN

### 1.1. Mở đầu

Trong bối cảnh chuyển đổi số đang diễn ra mạnh mẽ, việc ứng dụng công nghệ thông tin vào quản lý và điều hành công việc đã trở thành nhu cầu thiết yếu đối với các tổ chức, doanh nghiệp cũng như các nhóm làm việc học thuật. Đặc biệt, trong quản lý dự án và cộng tác nhóm, các vấn đề như phân công nhiệm vụ, theo dõi tiến độ, quản lý tài liệu và trao đổi thông tin thường gặp nhiều khó khăn nếu không có một hệ thống hỗ trợ tập trung và hiệu quả.

Trên thực tế, nhiều nhóm làm việc hiện nay vẫn sử dụng các công cụ rời rạc như bảng tính Excel, email hoặc các ứng dụng nhắn tin để quản lý dự án. Cách tiếp cận này dẫn đến tình trạng thông tin phân tán, khó kiểm soát tiến độ tổng thể, thiếu tính minh bạch và dễ phát sinh sai sót trong quá trình phối hợp. Mặc dù một số nền tảng quản lý dự án hiện đại như Trello, Jira hay Asana đã phần nào giải quyết được các vấn đề trên, nhưng các hệ thống này thường có chi phí sử dụng cao hoặc chưa phù hợp với nhu cầu tùy biến của các nhóm nhỏ, doanh nghiệp vừa và nhỏ (SME) hoặc môi trường học tập.

Xuất phát từ thực tiễn đó, việc nghiên cứu và xây dựng một hệ thống quản lý dự án và cộng tác nhóm dựa trên nền tảng web, có khả năng đáp ứng các chức năng cốt lõi, dễ sử dụng và phù hợp với điều kiện triển khai trong nước là yêu cầu mang tính cấp thiết. Đồng thời, quá trình thực hiện đề tài cũng giúp người thực hiện củng cố và vận dụng kiến thức về phân tích – thiết kế hệ thống, lập trình web và quản lý cơ sở dữ liệu vào một sản phẩm phần mềm thực tế.

### 1.2. Tổng quan đề tài

Đề tài “Xây dựng hệ thống quản lý dự án và cộng tác nhóm” tập trung nghiên cứu và phát triển một ứng dụng web hỗ trợ quản lý dự án theo hướng tập trung, trong đó cho phép người dùng tạo và quản lý dự án, phân công công việc, theo dõi tiến độ, trao đổi thông tin và chia sẻ tài liệu trong cùng một hệ thống thống nhất.

So với các công cụ quản lý dự án phổ biến hiện nay, hệ thống được xây dựng trong đề tài hướng đến các đối tượng người dùng là nhóm sinh viên, nhóm phát triển phần mềm nhỏ và các doanh nghiệp SME. Các chức năng được thiết kế ở mức độ vừa



phải, tập trung vào các nghiệp vụ cốt lõi thay vì tích hợp quá nhiều tính năng nâng cao, nhằm đảm bảo hệ thống dễ triển khai, dễ sử dụng và dễ mở rộng trong tương lai.

Về mặt kỹ thuật, hệ thống được xây dựng theo kiến trúc web, áp dụng mô hình MVC nhằm tách biệt rõ ràng giữa xử lý dữ liệu, logic nghiệp vụ và giao diện người dùng. Cơ sở dữ liệu được thiết kế theo mô hình quan hệ, đảm bảo tính toàn vẹn và nhất quán dữ liệu. Giao diện người dùng hướng đến tính trực quan, thân thiện và phù hợp với thói quen sử dụng của người dùng Việt Nam.

### **1.3. Mục tiêu và nhiệm vụ**

#### **1.3.1 Mục tiêu**

Mục tiêu chính của đề tài là xây dựng một hệ thống quản lý dự án và cộng tác nhóm trên nền tảng web nhằm hỗ trợ các nhóm làm việc quản lý công việc một cách hiệu quả, minh bạch và tập trung. Cụ thể:

- Xây dựng hệ thống cho phép tạo và quản lý nhiều dự án khác nhau;
- Hỗ trợ phân công công việc, theo dõi trạng thái và tiến độ thực hiện;
- Tạo môi trường trao đổi thông tin, bình luận và chia sẻ tài liệu trong nhóm;
- Thiết kế giao diện thân thiện, dễ dàng sử dụng và phù hợp với nhiều đối tượng người dùng khác nhau.

#### **1.3.2 Nhiệm vụ**

Để đạt được các mục tiêu trên, đề tài cần thực hiện các nhiệm vụ sau:

- Nghiên cứu các mô hình quản lý dự án và cộng tác nhóm phổ biến;
- Phân tích yêu cầu nghiệp vụ và xác định các chức năng chính của hệ thống;
- Thiết kế kiến trúc hệ thống và cơ sở dữ liệu quan hệ;
- Xây dựng các chức năng quản lý người dùng, dự án, công việc và tài liệu;
- Triển khai các chức năng cộng tác nhóm như bình luận, thông báo và lịch sử hoạt động;
- Kiểm thử hệ thống và đánh giá kết quả đạt được.

### **1.3.3 Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài tập trung vào việc xây dựng một hệ thống quản lý dự án và cộng tác nhóm dành cho các nhóm nhỏ đến vừa. Hệ thống tập trung vào các chức năng cốt lõi như quản lý dự án, quản lý công việc, phân quyền người dùng, chia sẻ tài liệu và trao đổi thông tin. Đề tài không đi sâu vào các tính năng nâng cao như phân tích dữ liệu chuyên sâu, quản lý tài chính dự án hoặc tích hợp với các hệ thống bên thứ ba.

### **1.4. Phương pháp nghiên cứu**

Trong quá trình thực hiện đề tài, các phương pháp nghiên cứu sau được áp dụng vào đề tài:

- Nghiên cứu tài liệu liên quan đến quản lý dự án, cộng tác nhóm và thiết kế hệ thống thông tin;
- Khảo sát thực tế nhu cầu sử dụng hệ thống quản lý dự án trong các nhóm học tập và làm việc;
- Phân tích yêu cầu và mô hình hóa hệ thống bằng các công cụ UML như Use Case Diagram và ERD;
- Thiết kế cơ sở dữ liệu theo mô hình quan hệ, áp dụng các nguyên tắc chuẩn hóa;
- Phát triển hệ thống theo mô hình MVC với backend sử dụng PHP và MySQL;
- Thiết kế giao diện người dùng với HTML, CSS và các framework hỗ trợ;
- Thực hiện kiểm thử chức năng và đánh giá kết quả triển khai hệ thống.

## CHƯƠNG 2 NGHIÊN CỨU LÝ THUYẾT

### 2.1. Quy trình nghiệp vụ về hệ thống quản lý dự án và cộng tác nhóm

#### 2.1.1 Tổng quan quy trình nghiệp vụ

Quy trình nghiệp vụ của hệ thống quản lý dự án và cộng tác nhóm được xây dựng nhằm hỗ trợ các nhóm làm việc tổ chức, theo dõi và kiểm soát quá trình thực hiện dự án một cách tập trung và hiệu quả. Hệ thống cho phép người dùng quản lý toàn bộ vòng đời của dự án, từ khởi tạo, phân công công việc, theo dõi tiến độ cho đến khi hoàn thành dự án.

Quy trình nghiệp vụ trong hệ thống được thiết kế dựa trên mô hình làm việc cộng tác, trong đó các thành viên trong nhóm có thể cùng tham gia vào một hoặc nhiều dự án với các vai trò khác nhau. Mỗi vai trò được phân quyền cụ thể nhằm đảm bảo tính kiểm soát và minh bạch trong quá trình quản lý và thực hiện công việc [1].

#### 2.1.2 Các vai trò trong hệ thống

Hệ thống quản lý dự án và cộng tác nhóm hỗ trợ các vai trò người dùng chính như sau:

- **Quản trị viên (Admin):** Có quyền quản lý toàn bộ hệ thống, bao gồm quản lý người dùng, phân quyền, tạo và chỉnh sửa dự án, theo dõi tình trạng hoạt động của hệ thống.
- **Quản lý dự án (Manager):** Chịu trách nhiệm tạo và quản lý các dự án, phân công công việc cho các thành viên, theo dõi tiến độ thực hiện và cập nhật trạng thái dự án.
- **Thành viên (Member):** Tham gia thực hiện các công việc được phân công, cập nhật trạng thái công việc, trao đổi thông tin và phối hợp với các thành viên khác trong nhóm.

Việc phân chia vai trò giúp hệ thống vận hành rõ ràng, hạn chế xung đột quyền hạn và nâng cao hiệu quả quản lý dự án.

#### 2.1.3 Quy trình khởi tạo và quản lý dự án

Quy trình khởi tạo dự án trong hệ thống được thực hiện theo các bước sau:

1. Quản lý dự án đăng nhập vào hệ thống;
2. Tạo mới dự án với các thông tin cơ bản như tên dự án, mô tả, thời gian thực hiện;
3. Thiết lập danh sách thành viên tham gia dự án và phân quyền tương ứng;
4. Khởi tạo các nhóm công việc hoặc giai đoạn (nếu có) để phục vụ việc tổ chức dự án.

Sau khi dự án được tạo, hệ thống cho phép người quản lý cập nhật thông tin, theo dõi tiến độ tổng thể và điều chỉnh kế hoạch khi cần thiết.

#### **2.1.4 Quy trình phân công và thực hiện công việc**

Mỗi dự án được chia thành nhiều công việc cụ thể nhằm dễ dàng quản lý và theo dõi tiến độ. Quy trình phân công và thực hiện công việc bao gồm:

1. Quản lý dự án tạo công việc mới và gán cho một hoặc nhiều thành viên;
2. Thiết lập các thông tin cho công việc như mô tả, thời hạn, mức độ ưu tiên;
3. Thành viên nhận công việc và cập nhật trạng thái thực hiện (chưa thực hiện, đang thực hiện, hoàn thành);
4. Các thành viên có thể trao đổi, bình luận và cập nhật tiến độ trong quá trình thực hiện công việc.

Quy trình này giúp đảm bảo mỗi công việc đều có người chịu trách nhiệm rõ ràng, đồng thời hỗ trợ theo dõi tiến độ một cách minh bạch.

#### **2.1.5 Quy trình theo dõi tiến độ và cộng tác nhóm**

Hệ thống cung cấp các công cụ hỗ trợ theo dõi tiến độ dự án theo thời gian thực. Người quản lý có thể theo dõi trạng thái công việc của từng thành viên, từ đó đánh giá mức độ hoàn thành của dự án.

Bên cạnh đó, hệ thống hỗ trợ các chức năng cộng tác như bình luận, thông báo và lịch sử hoạt động, giúp các thành viên trao đổi thông tin và phối hợp hiệu quả trong quá trình thực hiện dự án. Các thay đổi quan trọng trong dự án đều được ghi nhận và hiển thị, đảm bảo tính minh bạch và khả năng truy vết khi cần thiết.

### **2.1.6 Quy trình kết thúc dự án**

Khi các công việc trong dự án đã hoàn thành, quản lý dự án thực hiện kết thúc dự án trên hệ thống. Dữ liệu dự án được lưu trữ để phục vụ việc tra cứu và đánh giá sau này. Việc kết thúc dự án giúp tổng hợp kết quả thực hiện, rút ra kinh nghiệm và làm cơ sở cho các dự án tiếp theo.

### **2.1.7 Tổng kết quy trình nghiệp vụ**

Quy trình nghiệp vụ của hệ thống quản lý dự án và cộng tác nhóm được thiết kế theo hướng đơn giản, rõ ràng và phù hợp với các nhóm làm việc nhỏ và vừa. Việc phân chia vai trò, quy trình khởi tạo, phân công, theo dõi và kết thúc dự án giúp hệ thống đáp ứng tốt nhu cầu quản lý và cộng tác trong thực tế, đồng thời tạo nền tảng cho việc mở rộng và phát triển hệ thống trong tương lai.

## **2.2. Giới thiệu về PHP**

### **2.2.1 Khái niệm**

Ngôn ngữ PHP là từ viết tắt của Personal Home Page nay đã chuyển thành Hypertext Preprocessor. Thuật ngữ này là một dạng mã lệnh hoặc một chuỗi ngôn ngữ kịch bản được dùng để phát triển các ứng dụng web chạy trên máy chủ .



*Hình 2.1 PHP là gì*

(Nguồn: [https://vi.wikipedia.org/wiki/T%E1%BA%ADp\\_tin:PHP-logo.svg](https://vi.wikipedia.org/wiki/T%E1%BA%ADp_tin:PHP-logo.svg))

Khi các lập trình viên PHP viết chương trình, chuỗi lệnh sẽ được xử lý trên server sau đó sinh ra mã HTML trên client. Dựa vào đó, các ứng dụng trên website sẽ hoạt động một cách dễ dàng.

Ngôn ngữ PHP thường được dùng trong việc xây dựng và phát triển website bởi nó có thể kết nối dễ dàng với các website khác có sử dụng HTML. PHP cũng là ngôn ngữ lập trình có mã nguồn mở, tương thích với nhiều nền tảng khác nhau như MacOS, Linux, Windows,... PHP được nhiều người dùng đánh giá là dễ đọc nên đa số các lập trình viên sẽ lựa chọn học PHP trước khi bắt đầu vào nghề [3] [4] [4].

### 2.2.2 Ứng dụng của ngôn ngữ PHP trong lập trình

Ngôn ngữ lập trình PHP thường tập trung vào việc thiết lập chương trình cho máy chủ, tạo các cơ sở dữ liệu, xây dựng nội dung website, nhận dữ liệu cookie. Chưa hết, bạn còn có thể thực hiện được nhiều thao tác, công năng khác khi sử dụng ngôn ngữ này.

Một số ứng dụng phổ biến của PHP trong ngành IT:

**Thiết lập chương trình cho hệ thống máy chủ:** Đây là một ứng dụng chủ yếu nhất của PHP. Các PHP Developer sẽ phải thực hiện các thao tác như phân tích ngôn ngữ lập trình PHP, xây dựng máy chủ web và trình duyệt web.

**Tạo các dòng tập lệnh:** Các lập trình viên sẽ tạo ra một dòng tập lệnh để vận hành chương trình PHP mà không cần đến máy chủ. Kiểu lập trình này được sử dụng trên các hệ điều hành phổ biến như Linux hay Windows.

**Xây dựng các ứng dụng làm việc:** Bạn có thể ứng dụng những điểm mạnh vốn có của PHP để xây dựng ứng dụng phần mềm. Các lập trình viên thường dùng PHP – GTK làm nền tảng xây dựng phần mềm vì đây là nhánh mở rộng của ngôn ngữ lập trình này và không có sẵn trong các bản phân phối chính thức hiện nay.

**Hỗ trợ cho mọi loại cơ sở dữ liệu khác nhau:** Khi một website có hỗ trợ cơ sở dữ liệu tốt sẽ giúp ích cho việc vận hành, sao lưu và đặc biệt là backup dữ liệu đề phòng trường hợp xảy ra an ninh mạng [2] [3] [5].

### 2.2.3 Ưu điểm

**Mã nguồn mở và miễn phí:** PHP sử dụng miễn phí nên giúp tiết kiệm đáng kể ngân sách dự án. Việc cài đặt và sử dụng ngôn ngữ này cũng rất dễ dàng, bạn chỉ cần học chăm chỉ trong 3 – 6 tháng là đã có thể sử dụng thuần thục.

**Tính linh hoạt:** PHP là một ngôn ngữ đa nền tảng, có thể hoạt động trên bất kỳ hệ điều hành nào (Windows, Linux, macOS,...). Hơn nữa, PHP còn có thể kết hợp với nhiều ngôn ngữ lập trình khác để xây dựng các tính năng công nghệ một cách hiệu quả nhất.

**Hệ thống thư viện phong phú, tính cộng đồng cao:** Do sự phổ biến của ngôn ngữ PHP nên việc tìm các thư viện code hay hàm liên quan đến PHP sẽ cực kỳ đơn giản. Chưa kể, bạn sẽ nhận được sự trợ giúp từ các diễn đàn, đội nhóm chuyên sâu của PHP giúp việc học tập hay làm việc trở nên dễ dàng.

**Cơ sở dữ liệu đa dạng:** PHP cho phép kết nối với hầu hết các cơ sở dữ liệu khác như MySQL, SQLite, PostgreSQL, MS-SQL,... [2] [3].

### 2.2.4 Nhược điểm

Mặc dù sở hữu nhiều lợi ích nhưng ngôn ngữ PHP vẫn có một số hạn chế nhất định, trong đó vấn đề bảo mật được nhiều người quan tâm nhất. Bởi bản chất của PHP có mã nguồn mở nên các lỗ hổng của mã nguồn sẽ bị công khai ngay sau khi chúng được tìm thấy. Và các lỗ hổng này có thể bị khai thác cho các mục đích xấu trước khi chúng ta kịp sửa chữa.

Bên cạnh đó, ngôn ngữ lập trình PHP chỉ hoạt động được trên các website và giao diện không được gọn gàng, đẹp mắt. Độ bảo mật và hiệu suất của ngôn ngữ này cũng chưa tốt [2] [3].

### 2.2.5 Ứng dụng PHP trong dự án

Trong phạm vi đồ án, PHP được lựa chọn làm ngôn ngữ phát triển backend do có cú pháp tương đối đơn giản, dễ tiếp cận và phù hợp với các hệ thống web vừa và nhỏ. Bên cạnh đó, PHP hỗ trợ tốt mô hình MVC, giúp tách biệt rõ ràng giữa xử lý nghiệp vụ, dữ liệu và giao diện, từ đó nâng cao khả năng bảo trì và mở rộng hệ thống. Việc sử dụng PHP cũng giúp hệ thống dễ dàng triển khai trên các môi trường máy chủ phổ biến hiện nay.

Trong dự án PHP được sử dụng làm ngôn ngữ lập trình phía server để xử lý các chức năng cốt lõi như xác thực người dùng, quản lý dự án, quản lý công việc, phân quyền truy cập và xử lý dữ liệu từ cơ sở dữ liệu. Việc lựa chọn PHP giúp hệ thống đạt được sự ổn định, dễ triển khai trên môi trường thực tế và phù hợp với quy mô của đề án chuyên ngành

## 2.3. Giới thiệu về MySQL

### 2.3.1 Khái niệm

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được sử dụng rộng rãi để lưu trữ, quản lý và truy xuất dữ liệu. MySQL hoàn toàn miễn phí nằm trong nhóm LAMP (Linux – Apache – MySQL – PHP) [6].

Vì hệ quản trị cơ sở dữ liệu MySQL ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh và Mysql cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL chính vì thế nên MySQL được sử dụng và hỗ trợ của những lập trình viên yêu thích mã nguồn mở.



Hình 2.2 MySQL là gì

(Nguồn: <https://bizflycloud.vn/tin-tuc/mysql-la-gi-tai-sao-nen-su-dung-mysql-20200917180705499.htm>)



Nhưng MySQL không bao quát toàn bộ những câu truy vấn cao cấp như SQL Server. Vì vậy MySQL chỉ đáp ứng việc truy xuất đơn giản trong quá trình vận hành của website, thích hợp cho các ứng dụng có truy cập CSDL trên internet và có thể giải quyết hầu hết các bài toán trong PHP, Perl.

MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OSX, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...[7] [8].

### 2.3.2 Lịch sử hình thành và phát triển của MySQL

MySQL được phát triển vào năm 1994 bởi công ty Thụy Điển MySQL AB. Sau khi trải qua nhiều giai đoạn phát triển và chuyển giao, với những cột mốc quan trọng như việc trở thành mã nguồn mở vào năm 2000, sau đó Sun Microsystems đã mua lại MySQL vào năm 2008. Nhưng sau đó chỉ hai năm gần khổng lồ Oracle đã giữ quyền sở hữu MySQL vào năm 2010.

Sự ra đời của MariaDB như một “fork” của MySQL đã tạo ra một cuộc cạnh tranh lành mạnh, thúc đẩy cả hai hệ thống không ngừng cải tiến. Nhờ tính năng mã nguồn mở, hiệu suất cao, dễ sử dụng và cộng đồng lớn, MySQL đã trở thành một trong những hệ quản trị cơ sở dữ liệu phổ biến nhất thế giới, được sử dụng rộng rãi trong các ứng dụng web và các hệ thống lưu trữ dữ liệu [7] [8].

### 2.3.3 Đặc điểm của MySQL

**Hiệu suất cao và khả năng mở rộng:** MySQL được tối ưu hóa để xử lý một lượng lớn dữ liệu một cách nhanh chóng và hiệu quả. Hệ thống có thể dễ dàng mở rộng để đáp ứng nhu cầu lưu trữ và xử lý dữ liệu ngày càng tăng.

**Bảo mật và tính toàn vẹn dữ liệu:** MySQL cung cấp các tính năng bảo mật mạnh mẽ như xác thực người dùng, phân quyền, mã hóa dữ liệu và phát hiện xâm nhập, giúp bảo vệ dữ liệu khỏi các mối đe dọa an ninh.

**Tính linh hoạt và tùy biến cao:** MySQL hỗ trợ nhiều nền tảng, nhiều loại engine lưu trữ và cho phép người dùng tùy chỉnh cấu hình để phù hợp với nhu cầu cụ thể của ứng dụng.

**Cộng đồng lớn và hỗ trợ mạnh mẽ:** MySQL có một cộng đồng người dùng rất lớn, sẵn sàng chia sẻ kinh nghiệm và giải đáp thắc mắc.

**Mã nguồn mở:** MySQL là một phần mềm mã nguồn mở, người dùng có thể tự do sử dụng, sửa đổi và phân phối.

**Cơ chế quản lý dữ liệu:** MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu (database). Mỗi cơ sở dữ liệu có thể chứa nhiều bảng (table), mỗi bảng lưu trữ một loại dữ liệu cụ thể. Các bảng được liên kết với nhau thông qua các khóa (key) để tạo thành các mối quan hệ.

**Cơ chế phân quyền:** MySQL có cơ chế phân quyền người dùng rất chi tiết. Mỗi người dùng có thể được cấp quyền truy cập đến một hoặc nhiều cơ sở dữ liệu, với các quyền hạn khác nhau như đọc, ghi, sửa đổi cấu trúc bảng, v.v.

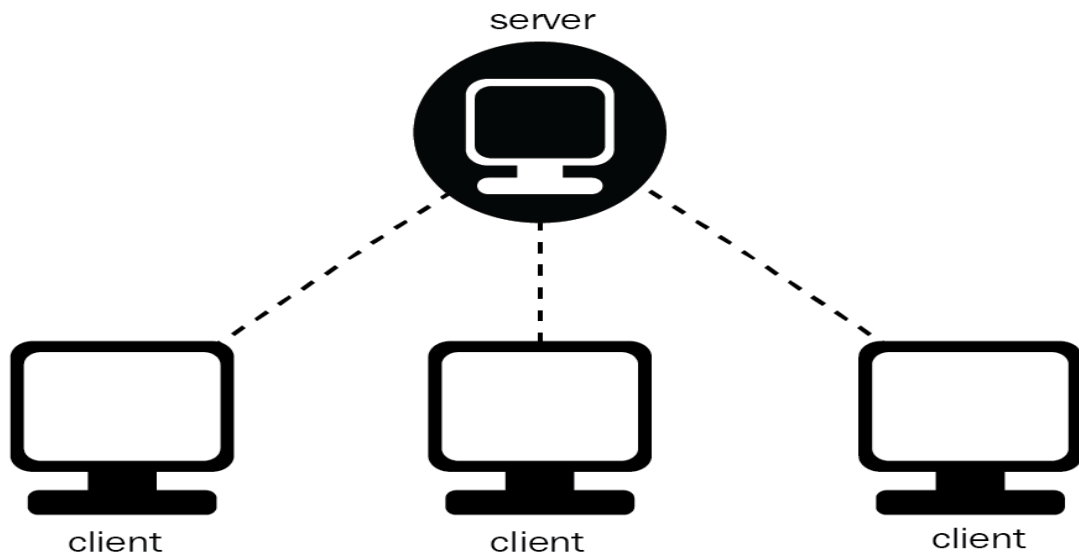
**Ngôn ngữ SQL:** MySQL sử dụng ngôn ngữ SQL (Structured Query Language) để tương tác với dữ liệu. SQL là ngôn ngữ chuẩn để truy vấn, cập nhật, xóa và quản lý dữ liệu trong các hệ quản trị cơ sở dữ liệu quan hệ [7] [8].

#### **2.3.4 Cách hoạt động**

MySQL hoạt động theo mô hình client-server, nghĩa là có một máy chủ (server) lưu trữ phần mềm MySQL và các cơ sở dữ liệu và có các máy khách (client) gửi yêu cầu đến máy chủ để thực hiện các thao tác với cơ sở dữ liệu.

MySQL sử dụng ngôn ngữ SQL để thực hiện các tác vụ. Các câu lệnh SQL là các lệnh được gửi đến máy chủ MySQL để đọc, thao tác hoặc kiểm soát dữ liệu được lưu trữ trong các cơ sở dữ liệu của nó. Ví dụ, lệnh SELECT được sử dụng để lấy dữ liệu từ cơ sở dữ liệu. Để thêm dữ liệu mới, lệnh INSERT được sử dụng. Tương tự, các lệnh DELETE và UPDATE được sử dụng để xóa hoặc cập nhật dữ liệu hiện có.

Trong mô hình client-server, máy khách thiết lập kết nối với máy chủ thông qua mạng. Kết nối này cho phép người dùng truy cập vào các cơ sở dữ liệu trên server và thực hiện các thao tác trên chúng bằng cách sử dụng các câu lệnh SQL. Khi một request từ giao diện người dùng (Graphical user interface – GUI), server sẽ phản hồi thông tin và trả về kết quả trên máy client. Quá trình này được minh họa qua sơ đồ sau:



*Hình 2.3 Cách hoạt động của MySQL*

(Nguồn: [https://codelearn.io/sharing/tim-hieu-ve-mo-hinh-client-server?srsId=AfmBOoqNYHroYKK2rgVZ25w22QO0rRKZs282xWx6IP09GXPsN4zXM-M\\_](https://codelearn.io/sharing/tim-hieu-ve-mo-hinh-client-server?srsId=AfmBOoqNYHroYKK2rgVZ25w22QO0rRKZs282xWx6IP09GXPsN4zXM-M_))

### **2.3.5 Ứng dụng MySQL trong dự án**

Trong dự án MySQL được sử dụng làm hệ quản trị cơ sở dữ liệu chính để lưu trữ thông tin người dùng, dự án, công việc, bình luận và lịch sử thao tác. Việc áp dụng MySQL giúp hệ thống đảm bảo hiệu năng ổn định, dữ liệu có cấu trúc rõ ràng và dễ dàng mở rộng khi số lượng người dùng tăng lên. Sự kết hợp giữa MySQL và PHP thông qua PDO giúp tăng tính bảo mật và hiệu quả trong quá trình xử lý dữ liệu

## **2.4. Giới thiệu về XAMPP**

### **2.4.1 XAMPP là gì**

XAMPP là một phần mềm mã nguồn mở đa nền tảng, được thiết kế để giả lập máy chủ web cục bộ (localhost), phù hợp cho môi trường phát triển website trên máy tính cá nhân. Phần mềm này được phát triển bởi Apache Friends, tích hợp sẵn các công nghệ lõi như Apache HTTP Server, MariaDB, ngôn ngữ lập trình PHP và Perl, cùng với các công cụ hỗ trợ như phpMyAdmin, FTP Server và Mail Server [9].



*Hình 2.4 XAMPP là gì*

(Nguồn: <https://bizflycloud.vn/tin-tuc/xampp-la-gi-huong-dan-su-dung-xampp-20181219170456717.htm>)

XAMPP là từ viết tắt của 5 chữ cái:

X - Cross-Platform (Ứng dụng đa nền tảng): Là viết tắt của các hệ điều hành mà nó hoạt động, đó là Windows, Linux, Mac OS X và Solaris.

A - Apache Server: Là web server được biết đến là máy chủ được sử dụng rộng rãi trên khắp thế giới với mục đích phân phối nội dung web. Đây là ứng dụng được cung cấp miễn phí dưới dạng phần mềm miễn phí được phát hành bởi Apache Software Foundation.

M - Hệ quản trị cơ sở dữ liệu MariaDB: Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở, được phát triển từ MySQL và hiện là một trong những hệ quản trị phổ biến trên thế giới. MariaDB được tích hợp trong XAMPP để cung cấp khả năng lưu trữ, truy xuất và quản lý dữ liệu cho các ứng dụng web, hoạt động hiệu quả khi kết hợp với PHP và các thành phần khác trong phần mềm XAMPP.

P - Ngôn ngữ lập trình PHP: Là ngôn ngữ lập trình phía máy chủ (server-side) phổ biến, được sử dụng để phát triển các trang web. PHP có thể chạy trên hầu hết các hệ điều hành như Windows, Linux, macOS và hỗ trợ nhiều hệ quản trị cơ sở dữ liệu như MySQL, MariaDB, PostgreSQL, Oracle,... Trong XAMPP, PHP được tích hợp sẵn, giúp người dùng dễ dàng thiết lập môi trường phát triển web nhanh chóng.

P - Ngôn ngữ lập trình Perl: Là ngôn ngữ lập trình mạnh mẽ, thường được sử dụng trong quản trị hệ thống, quản trị mạng, xử lý văn bản và phát triển web. Perl hỗ trợ lập trình phía máy chủ và có thể được dùng để xây dựng các ứng dụng web động. Trong XAMPP, Perl được tích hợp sẵn nhằm hỗ trợ người phát triển thử nghiệm và triển khai các ứng dụng viết bằng Perl một cách thuận tiện.

XAMPP cung cấp Control Panel trực quan, cho phép người dùng dễ dàng bật/tắt hoặc khởi động lại từng dịch vụ máy chủ một cách linh hoạt. Nhờ khả năng cài đặt nhanh, vận hành đơn giản và hỗ trợ đa nền tảng (Windows, Linux, Mac OS X và Solaris), XAMPP trở thành lựa chọn phổ biến cho lập trình viên web khi cần xây dựng và kiểm thử dự án trong môi trường nội bộ.

Hiện nay, XAMPP có 6 thông tin quan trọng được nhiều người dùng quan tâm là: Nhà phát triển, Phiên bản ổn định, Phiên bản xem thử, Ngôn ngữ lập trình, Hệ điều hành và Thể loại [9] [10].

#### **2.4.2 Công dụng chính của XAMPP**

XAMPP là một phần mềm giả lập máy chủ web cục bộ (local web server), giúp lập trình viên dễ dàng xây dựng, phát triển và kiểm thử website hoặc ứng dụng web ngay trên máy tính cá nhân của mình mà không cần kết nối internet hay thuê server riêng. Với khả năng tương thích cao, hoạt động ổn định trên nhiều hệ điều hành (Windows, Linux, Mac OS X và Solaris) và tích hợp sẵn các công cụ như Apache, MariaDB, PHP và Perl, XAMPP đặc biệt phù hợp cho việc học tập, nghiên cứu và thử nghiệm mã nguồn PHP (PHP open source). Nhờ sử dụng máy chủ chạy trên máy tính cá nhân, máy tính có thể đóng vai trò như một server, sử dụng ổ cứng làm nơi lưu trữ dữ liệu cho website.

#### **2.4.3 Ứng dụng XAMPP trong dự án**

Trong dự án XAMPP được sử dụng làm môi trường phát triển và kiểm thử cục bộ. Apache trong XAMPP đảm nhiệm vai trò máy chủ web, PHP xử lý logic nghiệp vụ theo mô hình MVC và MySQL lưu trữ dữ liệu của hệ thống. Việc sử dụng XAMPP giúp nhóm phát triển triển khai nhanh môi trường làm việc, dễ dàng kiểm tra các chức năng quản lý dự án, quản lý công việc và cộng tác nhóm trước khi hoàn thiện sản phẩm. Điều này giúp phát hiện và khắc phục lỗi trong quá trình phát triển, đồng thời đảm bảo hệ thống hoạt động ổn định trước khi đưa vào sử dụng.

## 2.5. Giới thiệu về Tailwind CSS

### 2.5.1 Tổng quan về Tailwind CSS

Tailwind CSS là một framework CSS theo hướng utility-first, được phát triển bởi Adam Wathan và cộng đồng, nhằm hỗ trợ xây dựng giao diện web một cách nhanh chóng, linh hoạt và dễ bảo trì. Khác với các framework CSS truyền thống như Bootstrap hay Material UI vốn cung cấp sẵn các thành phần giao diện hoàn chỉnh, Tailwind CSS tập trung cung cấp các lớp tiện ích (utility classes) nhỏ gọn, mỗi lớp đảm nhiệm một chức năng CSS cụ thể như màu sắc, khoảng cách, kích thước, căn chỉnh hay hiển thị.

Cách tiếp cận này cho phép lập trình viên xây dựng giao diện trực tiếp trong file HTML mà không cần viết nhiều CSS tùy chỉnh, từ đó giảm thiểu sự phụ thuộc vào các file CSS riêng lẻ và tăng tính nhất quán trong toàn bộ hệ thống giao diện [11] [12].



*Hình 2.5 Tổng quan về Tailwind CSS*

(Nguồn: <https://viblo.asia/p/tim-hieu-ve-tailwind-css-924Ijp6WKPM>)

### 2.5.2 Mô hình Utility-First trong Tailwind CSS

Mô hình utility-first là đặc trưng cốt lõi của Tailwind CSS. Thay vì định nghĩa các class mang tính ngữ nghĩa như .btn, .card, .header, Tailwind sử dụng các lớp tiện

ích như p-4, text-center, bg-blue-500, rounded-lg để mô tả trực tiếp cách hiển thị của phần tử.

Ví dụ, một nút bấm trong Tailwind CSS có thể được xây dựng bằng cách kết hợp nhiều utility class khác nhau, mỗi class đại diện cho một thuộc tính CSS cụ thể. Cách tiếp cận này giúp:

- Giảm việc viết CSS tùy chỉnh;
- Tránh trùng lặp kiểu dáng;
- Dễ dàng tái sử dụng và mở rộng giao diện;

### 2.5.3 Kiến trúc và cơ chế hoạt động của Tailwind CSS

Tailwind CSS hoạt động dựa trên quá trình build-time, trong đó framework sẽ quét toàn bộ mã nguồn (HTML, PHP, JavaScript, template engine) để xác định những utility class được sử dụng. Sau đó, chỉ các class cần thiết mới được đưa vào file CSS cuối cùng thông qua cơ chế Purge (Content Scanning).

Cơ chế này mang lại các lợi ích:

- Giảm đáng kể dung lượng file CSS;
- Tăng hiệu năng tải trang;
- Phù hợp cho các ứng dụng web quy mô lớn;

Ngoài ra, Tailwind CSS cho phép cấu hình sâu thông qua file `tailwind.config.js`, giúp tùy chỉnh màu sắc, font chữ, breakpoints, spacing và các thành phần thiết kế khác theo yêu cầu dự án [11] [12].

### 2.5.4 Hệ thống Responsive và Dark Mode

Tailwind CSS hỗ trợ responsive design một cách tự nhiên thông qua các breakpoint được định nghĩa sẵn như sm, md, lg, xl. Các utility class có thể được áp dụng linh hoạt theo từng kích thước màn hình, giúp giao diện hiển thị tốt trên nhiều thiết bị khác nhau.

Bên cạnh đó, Tailwind CSS hỗ trợ Dark Mode thông qua cơ chế class hoặc media query, cho phép xây dựng giao diện sáng – tối một cách hiệu quả, đáp ứng xu hướng trải nghiệm người dùng hiện đại.

---

### 2.5.5 Ưu điểm của Tailwind CSS trong thực tế phát triển dự án

Áp dụng Tailwind CSS trong các dự án web mang lại nhiều lợi ích thực tiễn:

- Tốc độ phát triển nhanh: Giảm thời gian viết CSS và thiết kế giao diện;
- Dễ bảo trì: Giao diện gắn liền với cấu trúc HTML, hạn chế xung đột CSS;
- Tính nhất quán cao: Thiết kế đồng bộ nhờ hệ thống utility chuẩn hóa;
- Khả năng mở rộng tốt: Dễ dàng tích hợp với các framework frontend và backend trong dự án;

Nhờ những ưu điểm này, Tailwind CSS được sử dụng rộng rãi trong các hệ thống quản lý, dashboard, ứng dụng web nội bộ và các dự án thương mại.

### 2.5.6 Ứng dụng Tailwind CSS trong dự án

Trong phạm vi đề án, Tailwind CSS được lựa chọn thay cho Bootstrap do khả năng tùy biến cao, giảm phụ thuộc vào component dựng sẵn, đồng thời phù hợp với các hệ thống dashboard và quản lý nội bộ có yêu cầu giao diện linh hoạt.

Trong dự án Tailwind CSS được sử dụng để xây dựng giao diện người dùng cho các chức năng như quản lý dự án, công việc và trang cá nhân người dùng. Việc sử dụng Tailwind CSS giúp hệ thống đạt được giao diện trực quan, hiện đại, responsive và dễ mở rộng, đồng thời giảm đáng kể khối lượng CSS tùy chỉnh cần quản lý.

## 2.6. Giới thiệu về JavaScript framework Alpine.js

### 2.6.1 Tổng quan về Alpine.js

Alpine.js là một framework JavaScript nhẹ (lightweight JavaScript framework) được phát triển bởi Caleb Porzio, nhằm hỗ trợ xây dựng các tương tác phía client một cách đơn giản và hiệu quả. Alpine.js được thiết kế để bổ sung khả năng xử lý động cho HTML mà không cần đến các framework JavaScript phức tạp như React, Vue hay Angular. Với triết lý “JavaScript behavior directly in HTML”, Alpine.js cho phép lập trình viên khai báo logic xử lý ngay trong các thuộc tính HTML thông qua các directive đặc trưng.

Nhờ dung lượng nhỏ, dễ học và dễ tích hợp, Alpine.js đặc biệt phù hợp với các ứng dụng web vừa và nhỏ, các hệ thống quản lý nội bộ hoặc các dự án sử dụng server-side rendering [13] [14].





Hình 2.6 Tổng quan về Alpine.js

(Nguồn: <https://alpinejs.dev/>)

### 2.6.2 Triết lý thiết kế và mô hình hoạt động của Alpine.js

Alpine.js được xây dựng dựa trên triết lý declarative programming, trong đó hành vi của giao diện được mô tả trực tiếp thông qua các thuộc tính HTML. Các directive như x-data, x-bind, x-on, x-show, x-if cho phép lập trình viên khai báo trạng thái và hành vi của thành phần giao diện một cách trực quan.

Mô hình hoạt động của Alpine.js có thể xem như một phiên bản tối giản của Vue.js, tập trung vào việc xử lý trạng thái cục bộ và các tương tác đơn giản. Điều này giúp giảm độ phức tạp trong mã nguồn và tăng khả năng bảo trì hệ thống.

### 2.6.3 Các thành phần cốt lõi trong Alpine.js

Alpine.js cung cấp một tập hợp các directive cơ bản nhưng đủ mạnh để xây dựng các tương tác phổ biến trong giao diện web:

- x-data: Khai báo trạng thái dữ liệu cục bộ cho một thành phần;
- x-bind: Ràng buộc thuộc tính HTML với dữ liệu;
- x-on: Lắng nghe và xử lý sự kiện người dùng;
- x-show / x-if: Điều khiển hiển thị phần tử;
- x-model: Liên kết hai chiều giữa dữ liệu và form input;

Các directive này cho phép xây dựng thành phần giao diện động như dropdown, modal, form validation, tab navigation mà không cần viết JavaScript phức tạp.

### 2.6.4 Cơ chế quản lý trạng thái và phản ứng (Reactivity)

Alpine.js sử dụng cơ chế quản lý trạng thái và phản ứng để tự động cập nhật giao diện khi trạng thái dữ liệu thay đổi. Khi một thuộc tính trong x-data được cập

nhật, Alpine.js sẽ tự động phản ánh sự thay đổi đó lên DOM mà không cần thao tác trực tiếp với JavaScript thuần.

Cơ chế này giúp giảm thiểu mã lệnh xử lý DOM thủ công, đồng thời nâng cao tính rõ ràng và dễ hiểu của mã nguồn. Mặc dù không hỗ trợ quản lý trạng thái toàn cục phức tạp như các framework lớn, Alpine.js vẫn đáp ứng tốt các nhu cầu tương tác giao diện phổ biến trong thực tế [13] [14].

### **2.6.5 Ưu điểm và hạn chế của Alpine.js**

#### **2.6.5.1 Ưu điểm**

- Nhẹ và nhanh: Dung lượng nhỏ, tải nhanh, không ảnh hưởng đến hiệu năng;
- Dễ học, dễ sử dụng: Cú pháp đơn giản, gần gũi với HTML;
- Dễ dàng tích hợp: Phù hợp với các dự án sử dụng PHP, Laravel, Blade hoặc HTML thuần;
- Giảm độ phức tạp cho frontend: Không cần build tool phức tạp cho các tương tác nhỏ;

#### **2.6.5.2 Hạn chế**

- Không phù hợp cho các ứng dụng SPA lớn, phức tạp;
- Khả năng quản lý state toàn cục còn hạn chế;
- Phụ thuộc nhiều vào cấu trúc HTML;

### **2.6.6 Ứng dụng Alpine.js trong dự án**

Trong dự án Alpine.js được sử dụng để xử lý các tương tác phía người dùng như hiển thị/ẩn modal, dropdown menu, xác nhận hành động, xử lý form và cập nhật trạng thái giao diện theo thời gian thực. Việc áp dụng Alpine.js giúp giảm khối lượng JavaScript thuần, tăng tính trực quan của mã nguồn và đảm bảo hiệu năng hoạt động ổn định cho hệ thống.

Sự kết hợp giữa Alpine.js và Tailwind CSS giúp giao diện hệ thống vừa linh hoạt về thiết kế, vừa hiệu quả trong xử lý tương tác, phù hợp với định hướng phát triển hệ thống quản lý dự án.

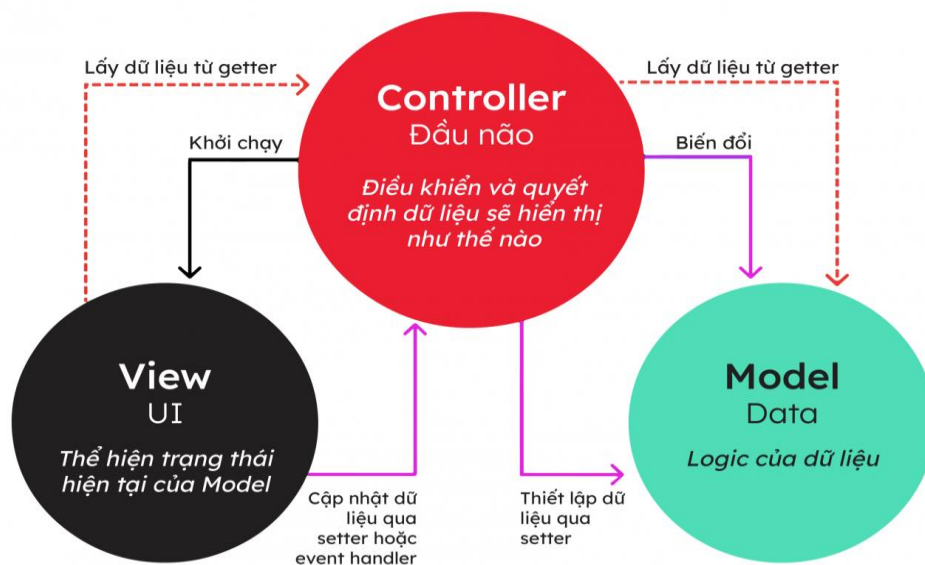
## 2.7. Giới thiệu về mô hình MVC

### 2.7.1 Tổng quan về mô hình MVC

Mô hình MVC (Model – View – Controller) là một mẫu kiến trúc phần mềm (architectural pattern) được sử dụng rộng rãi trong phát triển ứng dụng web nhằm tách biệt rõ ràng giữa dữ liệu, giao diện người dùng và logic xử lý nghiệp vụ. MVC giúp hệ thống trở nên dễ bảo trì, dễ mở rộng và thuận lợi cho làm việc nhóm trong các dự án phần mềm quy mô vừa và lớn .

Trong mô hình MVC, mỗi thành phần đảm nhiệm một vai trò riêng biệt nhưng phối hợp chặt chẽ với nhau để xử lý yêu cầu của người dùng, từ đó nâng cao tính tổ chức và chất lượng mã nguồn [15] [16].

### Mẫu Kiến trúc MVC



Hình 2.7 Tổng quan về mô hình MVC

(Nguồn: <https://itviec.com/blog/mvc-la-gi/>)

### 2.7.2 Các thành phần trong mô hình MVC

#### 2.7.2.1 Model

Model đại diện cho dữ liệu và logic nghiệp vụ của hệ thống. Thành phần này chịu trách nhiệm tương tác với cơ sở dữ liệu, thực hiện các thao tác truy vấn, cập nhật,

kiểm tra ràng buộc và xử lý quy tắc nghiệp vụ. Model không phụ thuộc vào giao diện hiển thị, giúp dữ liệu được quản lý độc lập và nhất quán [15] [16].

Trong các hệ thống quản lý, Model thường được ánh xạ trực tiếp với các bảng trong cơ sở dữ liệu và đóng vai trò trung tâm trong việc đảm bảo tính toàn vẹn dữ liệu.

#### **2.7.2.2 View**

View là thành phần chịu trách nhiệm hiển thị dữ liệu cho người dùng dưới dạng giao diện đồ họa hoặc nội dung web. View chỉ nhận dữ liệu từ Controller và không trực tiếp xử lý logic nghiệp vụ hay truy cập cơ sở dữ liệu. Việc tách View ra khỏi Model giúp giao diện dễ thay đổi mà không ảnh hưởng đến logic xử lý phía sau.

Trong thực tế, View thường được xây dựng bằng HTML kết hợp với CSS và JavaScript, đồng thời có thể tích hợp các framework frontend như Tailwind CSS hoặc Alpine.js để tăng tính tương tác [15] [16].

#### **2.7.2.3 Controller**

Controller đóng vai trò trung gian điều phối, tiếp nhận yêu cầu từ người dùng, xử lý logic điều hướng và tương tác với Model để lấy hoặc cập nhật dữ liệu. Sau khi xử lý xong, Controller sẽ truyền dữ liệu cần thiết cho View để hiển thị.

Controller giúp cô lập logic điều khiển luồng xử lý, từ đó giảm sự phụ thuộc giữa Model và View, nâng cao khả năng bảo trì và mở rộng hệ thống [15] [16].

#### **2.7.2.4 Cơ chế hoạt động của mô hình MVC**

Quy trình xử lý trong mô hình MVC thường diễn ra theo các bước sau:

- Người dùng gửi yêu cầu (request) thông qua giao diện View;
- Controller tiếp nhận và phân tích yêu cầu;
- Controller gọi Model để xử lý dữ liệu hoặc nghiệp vụ;
- Model trả kết quả về cho Controller;
- Controller chuyển dữ liệu sang View để hiển thị cho người dùng;

Quy trình này giúp luồng xử lý rõ ràng, dễ kiểm soát và hạn chế sự phụ thuộc chéo giữa các thành phần [2].

### **2.7.3 Ưu điểm và hạn chế của mô hình MVC**

#### **2.7.3.1 Ưu điểm**

- Tách biệt rõ ràng trách nhiệm giữa các thành phần;
- Dễ bảo trì và mở rộng hệ thống;
- Hỗ trợ làm việc nhóm hiệu quả, phân chia công việc rõ ràng;
- Tái sử dụng mã nguồn tốt;

#### **2.7.3.2 Hạn chế**

- Độ phức tạp ban đầu cao hơn so với kiến trúc đơn giản;
- Yêu cầu lập trình viên nắm vững cấu trúc và nguyên tắc thiết kế;
- Có thể dư thừa lớp đối với các dự án nhỏ;

### **2.7.4 Ứng dụng mô hình MVC trong phát triển web**

Mô hình MVC được áp dụng rộng rãi trong nhiều framework và nền tảng phát triển web như Laravel, ASP.NET MVC, Spring MVC và Ruby on Rails. Các framework này cung cấp sẵn cấu trúc MVC, giúp lập trình viên tập trung vào logic nghiệp vụ thay vì xử lý kiến trúc từ đầu.

Trong các hệ thống quản lý dự án, việc áp dụng MVC giúp tách biệt rõ ràng giữa xử lý nghiệp vụ, quản lý dữ liệu và hiển thị giao diện, từ đó nâng cao chất lượng và độ ổn định của hệ thống.

### **2.7.5 Ứng dụng mô hình MVC trong dự án**

Trong dự án mô hình MVC được sử dụng làm kiến trúc chính cho hệ thống. Model đảm nhiệm việc quản lý dữ liệu người dùng, dự án và công việc; Controller xử lý các yêu cầu nghiệp vụ như phân công công việc, cập nhật trạng thái và xác thực người dùng; View chịu trách nhiệm hiển thị giao diện và tương tác với người dùng. Việc áp dụng MVC giúp hệ thống có cấu trúc rõ ràng, dễ bảo trì và thuận lợi cho việc mở rộng chức năng trong tương lai.

## CHƯƠNG 3 HIỆN THỰC HÓA NGHIÊN CỨU

### 3.1. Đặc tả các yêu cầu

#### 3.1.1 Yêu cầu chức năng

##### *1. Quản lý người dùng*

- Cho phép người dùng đăng ký, đăng nhập và đăng xuất hệ thống;
- Cho phép quản trị viên tạo, chỉnh sửa, khóa hoặc xóa tài khoản người dùng;
- Cho phép cập nhật thông tin cá nhân và đổi mật khẩu;
- Hỗ trợ phân quyền người dùng theo vai trò (Admin, Manager, Member);

##### *2. Quản lý dự án*

- Cho phép tạo mới, chỉnh sửa, xóa dự án;
- Cho phép thêm, xóa thành viên khỏi dự án;
- Phân công vai trò của từng thành viên trong dự án;
- Hiện thị danh sách dự án mà người dùng tham gia;
- Lưu trữ thông tin mô tả, thời gian và trạng thái dự án;

##### *3. Quản lý công việc*

- Cho phép tạo, chỉnh sửa, xóa công việc trong dự án;
- Phân công công việc cho người dùng cụ thể;
- Thiết lập mức độ ưu tiên và thời hạn hoàn thành;
- Thay đổi trạng thái công việc dựa trên mô hình Kanban (To Do – In Progress – Done);
- Theo dõi tiến độ thực hiện công việc;

##### *4. Quản lý tài liệu*

- Cho phép tải lên, xem và tải xuống tài liệu liên quan đến dự án;
- Kiểm soát quyền truy cập tài liệu theo vai trò người dùng;
- Lưu trữ tài liệu theo từng dự án;

### **5. Theo dõi và báo cáo**

- Ghi nhận lịch sử thao tác của người dùng;
- Hiện thị thống kê số lượng công việc theo trạng thái;
- Hỗ trợ báo cáo tiến độ dự án ở mức cơ bản;

### **3.1.2 Yêu cầu phi chức năng**

#### **1. Hiệu năng**

- Thời gian phản hồi cho các thao tác chính nhanh chóng;
- Hệ thống hoạt động ổn định với 5-20 người dùng đồng thời;
- Tối ưu truy vấn cơ sở dữ liệu nhằm giảm độ trễ;

#### **2. Bảo mật**

- Mật khẩu người dùng phải được mã hóa;
- Chống các tấn công phổ biến: SQL Injection;;
- Quản lý session an toàn, tự động hết hạn khi không hoạt động;
- Phân quyền truy cập rõ ràng và chặt chẽ;

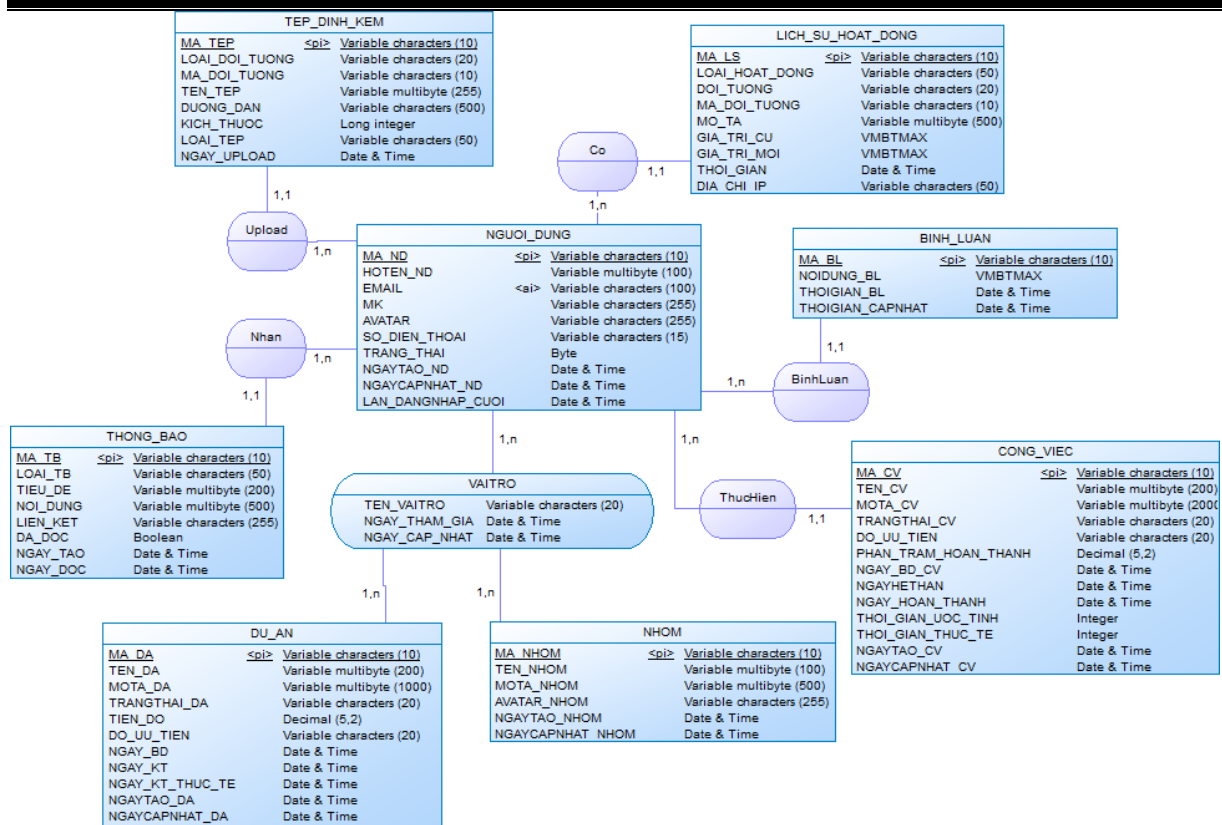
#### **3. Khả năng sử dụng**

- Giao diện trực quan, dễ sử dụng, hỗ trợ tiếng Việt;
- Người dùng mới có thể làm quen trong thời gian ngắn;
- Hỗ trợ nhiều kích thước màn hình (responsive);

### **3.2. Thiết kế cơ sở dữ liệu quan hệ**

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

Mô hình ERD được thiết kế nhằm đảm bảo tính toàn vẹn dữ liệu và giảm thiểu dư thừa thông tin. Các thực thể chính bao gồm Người dùng, Dự án, Công việc và các bảng liên kết nhằm biểu diễn mối quan hệ giữa các thực thể. Dưới đây là mô hình thực thể kết hợp (ERD):



Hình 3.1. Mô hình thực thể kết hợp (ERD - Entity Relationship Diagram)

### 3.2.2 Mô hình dữ liệu quan hệ

Cấu trúc các bảng:

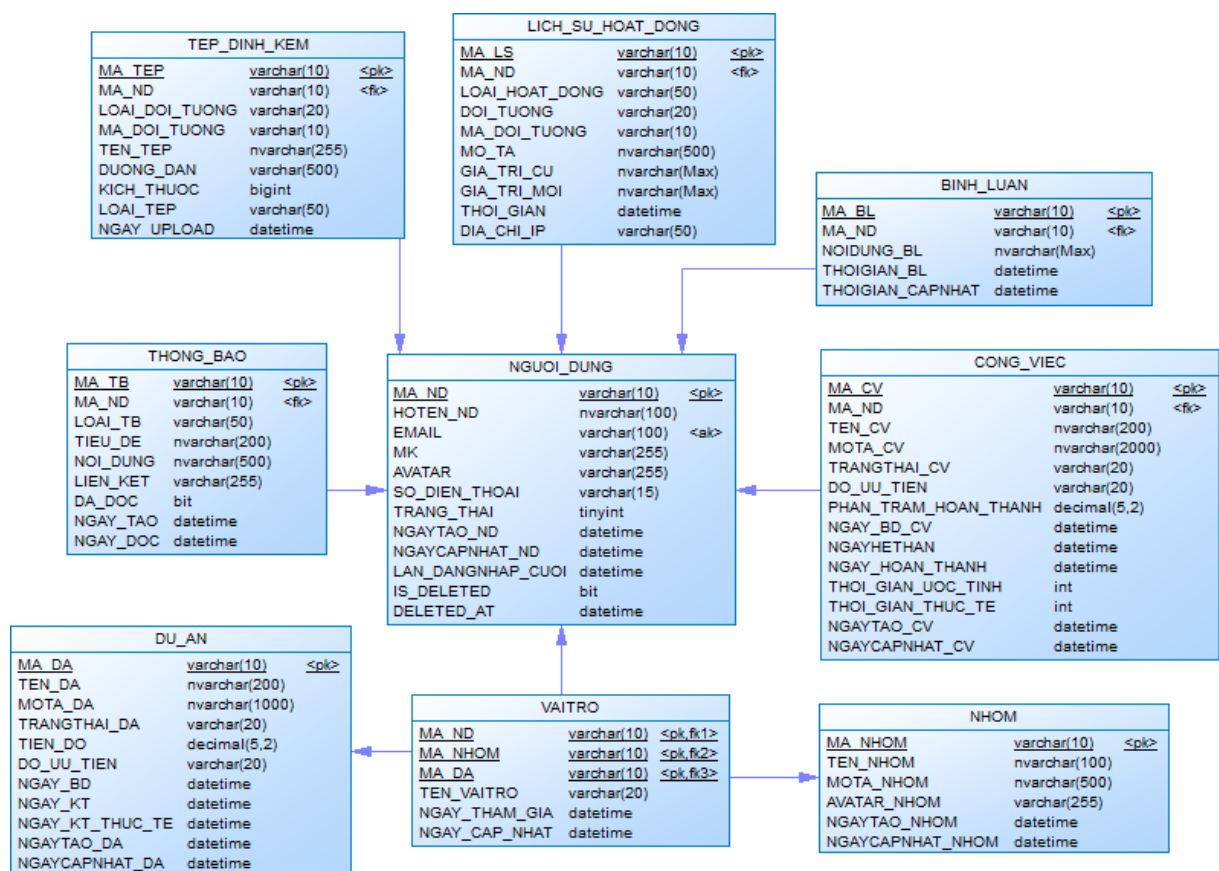
- 1) NGUOIDUNG( **Ma\_Nd**, HoTen\_Nd, Email, Mk, Avatar, So\_Dien\_Thoai, Trang\_Thai, NgayTao\_Nd, NgayCapNhat\_Nd, Lan\_DangNhap\_Cuoi)
- 2) DUAN( **Ma\_Da**, Ten\_Da, MoTa\_Da, TrangThai\_Da, SoDuTien, NgayBd, NgayKt, NgayTao\_Da, NgayCapNhat\_Da)
- 3) VAITRO( **Mand, MaDa**, TenVaiTro, NgayThamGia, NgayCapNhat)
- 4) NHOM( **MaNhom**, TenNhom, MoTa\_Nhom, Avatar\_Nhom, NgayTao, NgayCapNhat)
- 5) CONGVIEC( **MaCv**, TenCv, MoTa\_Cv, TrangThai\_Cv, Do\_UuTien, PhanTramHoanThanh, NgayBd\_Cv, NgayHetHan, NgayHoanThanh, ThoiGianUocTinh, NgayTao, NgayCapNhat)
- 6) BINHLUAN( **MaBl**, NoiDung\_Bl, ThoiGian\_Bl, ThoiGian\_CapNhat, Mand, MaCv)



- 7) TEPDINHKEM( **MaTep**, Loai\_DoiTuong, Ma\_DoiTuong, TenTep, DuongDan, KichThuoc, LoaiTep, NgayUpload)
- 8) THONGBAO( **MaTb**, Loai\_Tb, TieuDe, NoiDung, LienKet, DaDoc, NgayTao, NgayDoc, Mand)
- 9) LICHSUHOATDONG( **MaLs**, LoaiHoatDong, DoiTuong, MaDoiTuong, MoTa, GiaTri\_Cu, GiaTri\_Moi, ThoiGian, DiaChi\_Ip)

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

Dưới đây là mô hình dữ liệu vật lý:



Hình 3.2 Mô hình dữ liệu vật lý (Physical Data Model - PDM)

### 3.2.4 Mô tả các bảng

1. Bảng Người dùng:

Bảng 3.1 Mô tả bảng Người dùng

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>Ma_Nd</u></b>	Varchar(10)	Mã người dùng
2	HoTen_Nd	Nvarchar(100)	Họ và tên người dùng

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
3	Email	Nvarchar(100)	Email đăng nhập
4	Mk	Nvarchar(255)	Mật khẩu người dùng
5	Avatar	Nvarchar(255)	Ảnh đại diện
6	SoDienThoai	Varchar(15)	Số điện thoại
7	TrangThai	Tinyint	Trạng thái hoạt động
8	NgayTaoNd	Datetime	Ngày tạo tài khoản
9	NgayCapNhatNd	Datetime	Ngày cập nhật thông tin
10	LanDangNhapCuoi	Datetime	Lần đăng nhập cuối

## 2. Bảng Dự án:

*Bảng 3.2. Mô tả bảng Dự án*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaDa</u></b>	Varchar(10)	Mã dự án
2	TenDa	Nvarchar(200)	Tên dự án
3	MoTaDa	Nvarchar(1000)	Mô tả dự án
4	TrangThaiDa	Varchar(20)	Trạng thái dự án
5	TienDo	Decimal(5,2)	Tiến độ dự án
6	SoDuTien	Varchar(20)	Số dư/ngân sách dự án
7	NgayBd	Datetime	Ngày bắt đầu
8	NgayKt	Datetime	Ngày kết thúc
9	NgayTaoDa	Datetime	Ngày tạo dự án
10	NgayCapNhatDa	Datetime	Ngày cập nhật dự án

## 3. Bảng Công việc:

*Bảng 3.3. Mô tả bảng Công việc*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaCv</u></b>	Varchar(10)	Mã công việc
2	MaNd	Varchar(10)	Mã người thực hiện
3	TenCv	Nvarchar(200)	Tên công việc

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
4	MoTaCv	Nvarchar(1000)	Mô tả công việc
5	TrangThaiCv	Varchar(20)	Trạng thái công việc
6	DoUuTien	Int	Độ ưu tiên
7	PhanTramHoanThanh	Decimal(5,2)	Phần trăm hoàn thành
8	NgayBd	Datetime	Ngày bắt đầu
9	NgayHetHan	Datetime	Hạn hoàn thành
10	NgayHoanThanh	Datetime	Ngày hoàn thành
11	ThoiGianUocTinh	Int	Thời gian ước tính
12	NgayTaoCv	Datetime	Ngày tạo công việc
13	NgayCapNhatCv	Datetime	Ngày cập nhật

#### 4. Bảng Bình luận:

*Bảng 3.4. Mô tả bảng Bình luận*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaBl</u></b>	Varchar(10)	Mã bình luận
2	MaNd	Varchar(10)	Người bình luận
3	NoiDungBl	Nvarchar(Max)	Nội dung bình luận
4	ThoiGianBl	Datetime	Thời gian bình luận
5	ThoiGianCapNhat	Datetime	Thời gian cập nhật

#### 5. Bảng Tập đính kèm

*Bảng 3.5. Mô tả bảng Tập đính kèm*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaTep</u></b>	Varchar(10)	Mã tập đính kèm
2	MaNd	Varchar(10)	Người tải lên
3	LoaiDoiTuong	Varchar(20)	Loại đối tượng
4	MaDoiTuong	Varchar(10)	Mã đối tượng
5	TenTep	Nvarchar(255)	Tên tập
6	DuongDan	Nvarchar(500)	Đường dẫn lưu trữ

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
7	KichThuoc	Bigint	Kích thước tệp
8	LoaiTep	Varchar(50)	Loại tệp
9	NgayUpload	Datetime	Ngày tải lên

6. Bảng Thông báo:

*Bảng 3.6. Mô tả bảng Thông báo*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaTb</u></b>	Varchar(10)	Mã thông báo
2	MaNd	Varchar(10)	Người nhận
3	LoaiTb	Varchar(50)	Loại thông báo
4	TieuDe	Nvarchar(200)	Tiêu đề
5	NoiDung	Nvarchar(500)	Nội dung
6	LienKet	Nvarchar(255)	Liên kết
7	DaDoc	Bit	Đã đọc hay chưa
8	NgayTao	Datetime	Ngày tạo
9	NgayDoc	Datetime	Ngày đọc

7. Bảng Lịch sử hoạt động:

*Bảng 3.7. Mô tả bảng Lịch sử hoạt động*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaLs</u></b>	Varchar(10)	Mã lịch sử
2	MaNd	Varchar(10)	Người thực hiện
3	LoaiHoatDong	Varchar(50)	Loại hoạt động
4	DoiTuong	Varchar(50)	Đối tượng tác động
5	MaDoiTuong	Varchar(10)	Mã đối tượng
6	MoTa	Nvarchar(Max)	Mô tả hoạt động
7	GiaTriCu	Nvarchar(Max)	Giá trị cũ
8	GiaTriMoi	Nvarchar(Max)	Giá trị mới
9	ThoiGian	Datetime	Thời gian

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
10	DiaChiIp	Varchar(50)	Địa chỉ IP

8. Bảng Nhóm:

*Bảng 3.8 Mô tả bảng Nhóm*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaNhom</u></b>	Varchar(10)	Mã nhóm
2	TenNhom	Nvarchar(200)	Tên nhóm
3	MoTaNhom	Nvarchar(500)	Mô tả nhóm
4	AvatarNhom	Nvarchar(255)	Ảnh đại diện nhóm
5	NgayTaoNhom	Datetime	Ngày tạo nhóm
6	NgayCapNhatNhom	Datetime	Ngày cập nhật nhóm

9. Bảng Vai trò:

*Bảng 3.9 Mô tả bảng Vai trò*

STT	Tên thuộc tính	Kiểu dữ liệu	Mô tả
1	<b><u>MaNd</u></b>	Varchar(10)	Mã người dùng
2	MaNhom	Varchar(10)	Mã nhóm
3	MaDa	Varchar(10)	Mã dự án
4	TenVaiTro	Varchar(20)	Tên vai trò
5	NgayThamGia	Datetime	Ngày tham gia
6	NgayCapNhat	Datetime	Ngày cập nhật

### 3.3. Kiến trúc hệ thống

#### 3.3.1 Sơ đồ Use case

Sơ đồ Use Case của hệ thống mô tả các chức năng chính và mối quan hệ giữa người dùng với hệ thống. Các tác nhân được phân quyền rõ ràng theo vai trò quản trị, quản lý dự án và thành viên, đảm bảo hệ thống hỗ trợ đầy đủ các nghiệp vụ quản lý dự án, công việc và tương tác nhóm:

- Member: Xem chi tiết dự án, xem chi tiết công việc và bình luận, cập nhật công việc, upload, download tài liệu.
- Manager: Tạo, sửa, xóa dự án, công việc, xem danh sách dự án, công việc, xem chi tiết dự án, chi tiết công việc, bình luận, cập nhật công việc, upload, download tài liệu.
- Admin: Quản lý dự án, công việc, người dùng, tài liệu, báo cáo thống kê.



Hình 3.3. Sơ đồ Use case

### 3.3.2 Đặc tả Use case đăng nhập

Bảng 3.10 Đặc tả Use case đăng nhập

Thuộc tính	Mô tả
Tên	Đăng nhập
Actor	Member, Manager, Admin

Thuộc tính	Mô tả
Mô tả	Người dùng đăng nhập vào hệ thống
Tiền điều kiện	Người dùng đã có tài khoản
Hậu điều kiện	Người dùng được xác thực và chuyển đến Dashboard

Luồng chính:

1. Người dùng truy cập trang đăng nhập;
2. Hệ thống hiển thị form đăng nhập;
3. Người dùng nhập email và mật khẩu;
4. Người dùng nhấn nút “Đăng nhập”;
5. Hệ thống xác thực thông tin;
6. Hệ thống tạo session và chuyển đến Dashboard;

Luồng thay thế:

- 5.a. Email không tồn tại: Hiển thị thông báo lỗi;
- 5.b. Mật khẩu sai: Hiển thị thông báo lỗi;
- 5.c. Tài khoản bị khóa: Hiển thị thông báo tài khoản bị khóa;

### 3.3.3 Đặc tả Use case Tạo dự án

*Bảng 3.11 Đặc tả Use case Quản lý dự án*

Thuộc tính	Mô tả
Tên	Tạo dự án
Actor	Manager
Mô tả	Người dùng tạo, xem, sửa, xóa dự án
Tiền điều kiện	Người dùng đã đăng nhập
Hậu điều kiện	Dự án được tạo/cập nhật/xóa thành công

Luồng chính (Tạo dự án):

1. Người dùng nhấn nút “Tạo dự án mới”;

2. Hệ thống hiển thị form tạo dự án;
3. Người dùng nhập thông tin: tên, mô tả, màu sắc, ngày bắt đầu/kết thúc;
4. Người dùng nhấn “Luu”;
5. Hệ thống validate dữ liệu;
6. Hệ thống tạo dự án và thêm người tạo làm Owner;
7. Hệ thống chuyển đến trang chi tiết dự án;

### 3.3.4 Đặc tả Use case Tạo công việc

*Bảng 3.12 Đặc tả Use case Quản lý công việc*

Thuộc tính	Mô tả
Tên	Tạo công việc mới
Actor	Manager
Mô tả	Người dùng tạo công việc mới trong dự án
Tiền điều kiện	Người dùng đã đăng nhập và có quyền tasks.create
Hậu điều kiện	Công việc được tạo và hiển thị trong Kanban Board

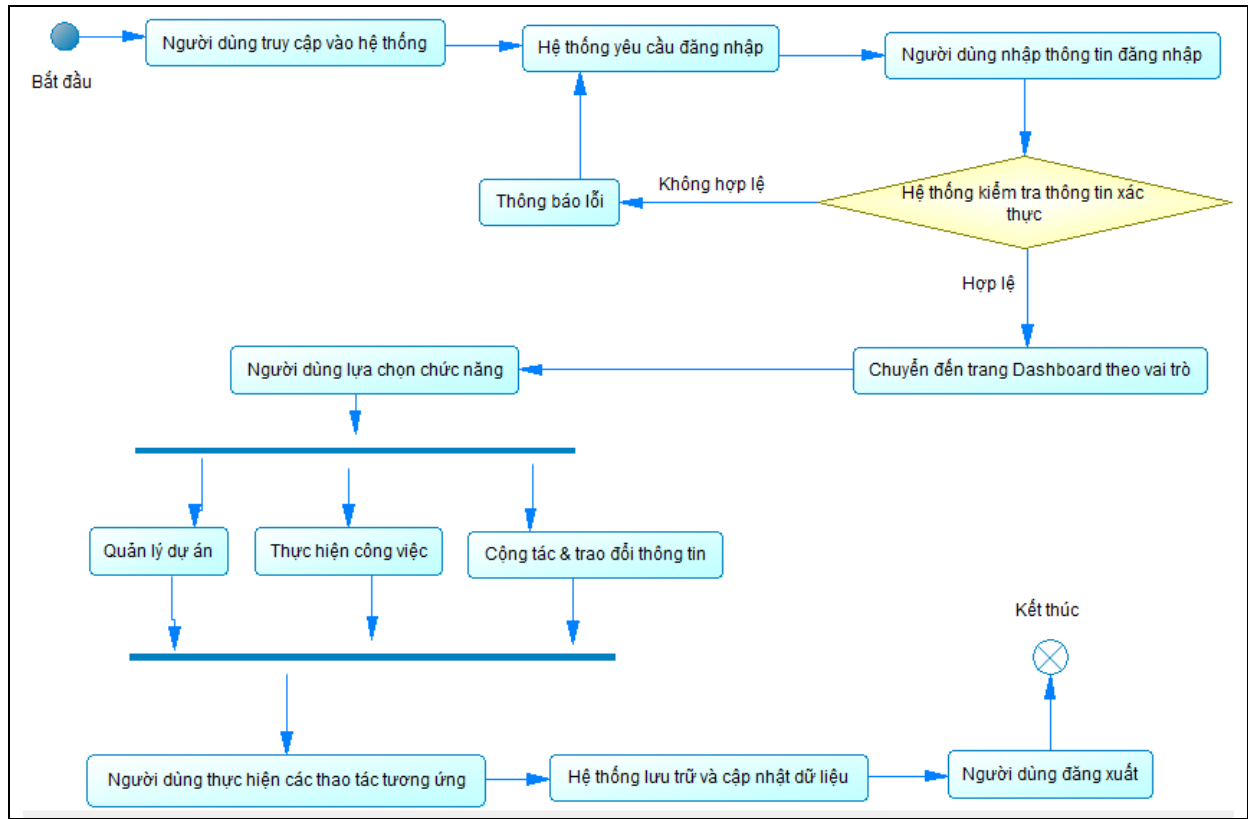
Luồng chính:

1. User nhấn nút “Tạo công việc mới”;
2. Hệ thống hiển thị form tạo task;
3. User nhập thông tin: tiêu đề, mô tả, dự án, độ ưu tiên, ngày hết hạn;
4. User nhấn “Luu”;
5. Hệ thống validate dữ liệu;
6. Hệ thống tạo UUID cho task;
7. Hệ thống lưu task vào database;
8. Hệ thống ghi activity log;
9. Hệ thống hiển thị thông báo thành công;
10. Hệ thống chuyển đến trang chi tiết task;



### 3.3.5 Luồng nghiệp vụ hệ thống

#### 3.3.5.1 Luồng nghiệp vụ tổng quát



Hình 3.4 Luồng nghiệp vụ tổng quát

Mô tả:

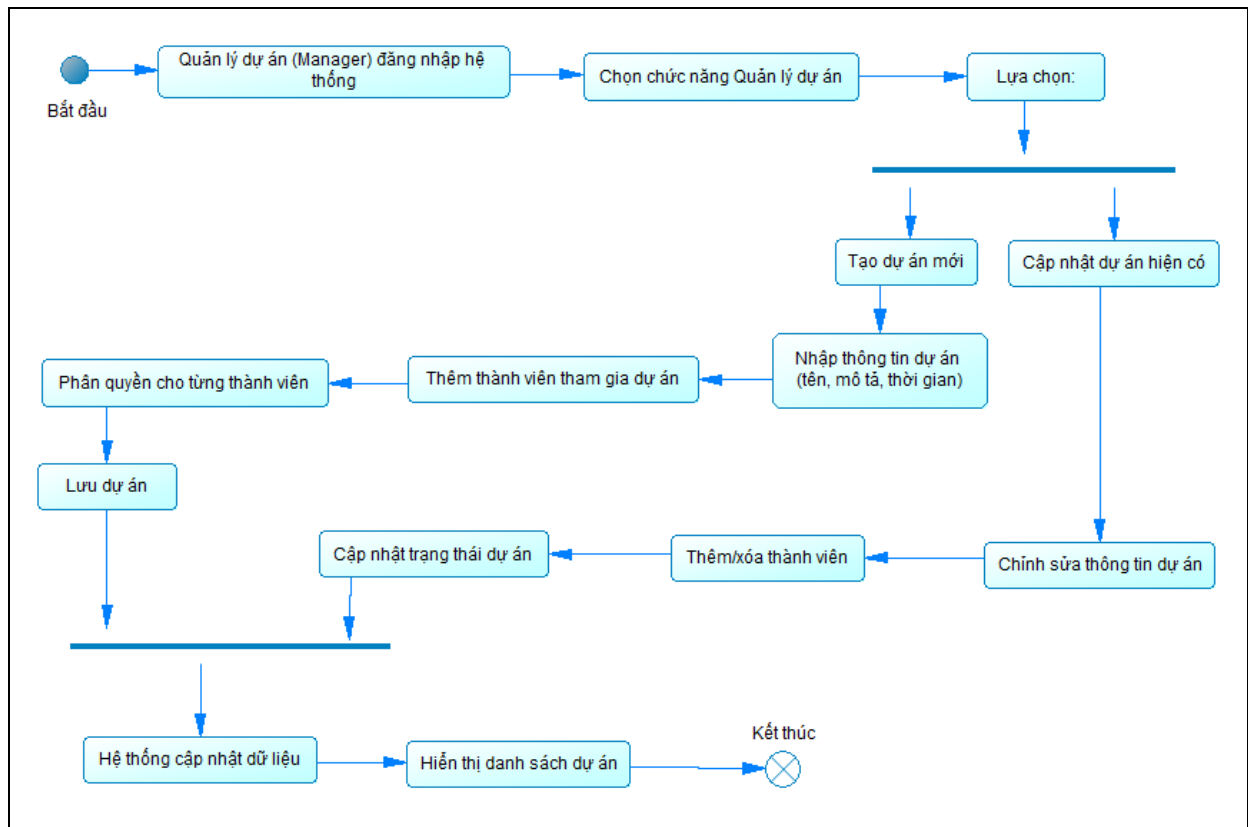
Bắt đầu

1. Người dùng truy cập hệ thống;
2. Hệ thống yêu cầu đăng nhập;
3. Người dùng nhập thông tin đăng nhập;
4. Hệ thống kiểm tra thông tin xác thực;
  - Nếu không hợp lệ → Thông báo lỗi → Quay lại bước đăng nhập;
  - Nếu hợp lệ → Chuyển đến trang chính theo vai trò;
5. Người dùng lựa chọn chức năng:
  - Quản lý dự án;
  - Thực hiện công việc;

- Cộng tác & trao đổi thông tin;
6. Người dùng thực hiện các thao tác tương ứng;
  7. Hệ thống lưu trữ và cập nhật dữ liệu;
  8. Người dùng đăng xuất;

Kết thúc.

### 3.3.5.2 Luồng nghiệp vụ quản lý dự án



Hình 3.5 Luồng nghiệp vụ quản lý dự án

Mô tả chi tiết:

Bắt đầu

1. Quản lý dự án (Manager) đăng nhập hệ thống;
2. Chọn chức năng Quản lý dự án;
3. Lựa chọn:
  - Tạo dự án mới;
  - Cập nhật dự án hiện có;

4. Nhánh quyết định:

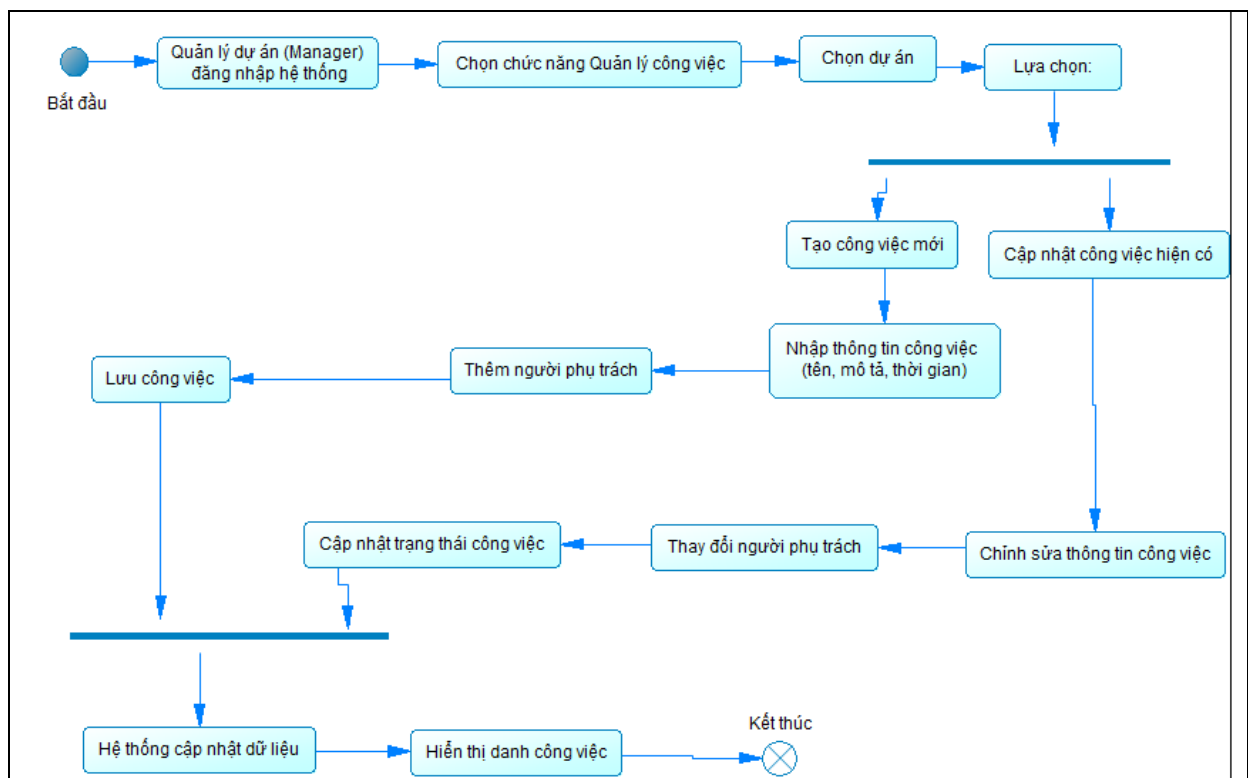
- Tạo dự án mới:
  - Nhập thông tin dự án (tên, mô tả, thời gian);
  - Thêm thành viên tham gia dự án;
  - Phân quyền cho từng thành viên;
  - Lưu dự án;
- Cập nhật dự án:
  - Chỉnh sửa thông tin dự án;
  - Thêm/xóa thành viên;
  - Cập nhật trạng thái dự án;

5. Hệ thống cập nhật dữ liệu;

6. Hiển thị danh sách dự án;

Kết thúc.

**3.3.5.3 Luồng nghiệp vụ quản lý công việc**



Hình 3.6 Luồng nghiệp vụ quản lý công việc

Mô tả chi tiết:

Bắt đầu

1. Quản lý (Manager) đăng nhập hệ thống;
2. Chọn chức năng Quản lý công việc;
3. Chọn dự án;
4. Lựa chọn:
  - Tạo công việc mới;
  - Cập nhật công việc hiện có;
5. Nhánh quyết định:
  - Tạo công việc mới:
    - Nhập thông tin công việc (tên, mô tả, thời gian);
    - Thêm người phụ trách công việc;
    - Lưu công việc;
  - Cập nhật công việc:
    - Chỉnh sửa thông tin công việc;
    - Thay đổi người phụ trách;
    - Cập nhật trạng thái công việc;
6. Hệ thống cập nhật dữ liệu;
7. Hiển thị danh sách công việc;

Kết thúc.

## CHƯƠNG 4 KẾT QUẢ NGHIÊN CỨU

### 4.1. Kết quả thực hiện và đánh giá hệ thống

#### 4.1.1 Kết quả thực hiện hệ thống

Sau quá trình phân tích, thiết kế và hiện thực, hệ thống quản lý dự án và cộng tác nhóm đã được xây dựng và triển khai thành công trên môi trường web. Hệ thống đáp ứng đầy đủ các yêu cầu chức năng đã đề ra, bao gồm quản lý người dùng, quản lý dự án, quản lý công việc và hỗ trợ cộng tác nhóm.

Hệ thống cho phép người dùng đăng nhập và thao tác theo vai trò được phân quyền. Quản trị viên có thể quản lý người dùng và giám sát hệ thống; quản lý dự án có thể tạo dự án, phân công công việc và theo dõi tiến độ; thành viên có thể thực hiện các công việc được giao và trao đổi thông tin trong nhóm. Các chức năng được tích hợp trong một nền tảng thống nhất, giúp giảm thiểu tình trạng phân tán thông tin và nâng cao hiệu quả phối hợp.

#### 4.1.2 Đánh giá mức độ đáp ứng yêu cầu

##### 4.1.2.1 Đánh giá yêu cầu chức năng

Hệ thống đã đáp ứng đầy đủ các yêu cầu chức năng cơ bản của một hệ thống quản lý dự án và cộng tác nhóm, bao gồm:

- Quản lý người dùng và phân quyền theo vai trò;
- Quản lý dự án và danh sách thành viên tham gia;
- Quản lý công việc với các trạng thái thực hiện;
- Hỗ trợ trao đổi thông tin thông qua bình luận;

Các chức năng hoạt động ổn định, dữ liệu được xử lý và lưu trữ nhất quán, đáp ứng được nhu cầu quản lý và cộng tác trong các nhóm làm việc nhỏ và vừa.

##### 4.1.2.2 Đánh giá yêu cầu phi chức năng

- Tính dễ sử dụng: Giao diện hệ thống được thiết kế theo hướng trực quan, dễ tiếp cận với người dùng không chuyên.
- Tính ổn định: Hệ thống hoạt động ổn định trong quá trình thử nghiệm với số lượng người dùng đồng thời ở mức nhỏ.

- Tính bảo mật: Việc phân quyền theo vai trò giúp kiểm soát truy cập và hạn chế thao tác trái phép.
- Khả năng mở rộng: Kiến trúc MVC tạo nền tảng thuận lợi cho việc mở rộng chức năng trong tương lai.

#### **4.1.2.3 Ưu điểm của hệ thống**

Hệ thống quản lý dự án và cộng tác nhóm có một số ưu điểm nổi bật như sau:

- Giao diện đơn giản, trực quan, phù hợp với nhiều đối tượng người dùng;
- Các chức năng tập trung vào nghiệp vụ cốt lõi, dễ sử dụng và dễ triển khai;
- Kiến trúc hệ thống rõ ràng, thuận lợi cho việc bảo trì và phát triển;
- Hỗ trợ tốt việc theo dõi tiến độ và phối hợp giữa các thành viên trong nhóm.

Những ưu điểm này cho thấy hệ thống có tính khả thi và phù hợp với phạm vi của đề tài.

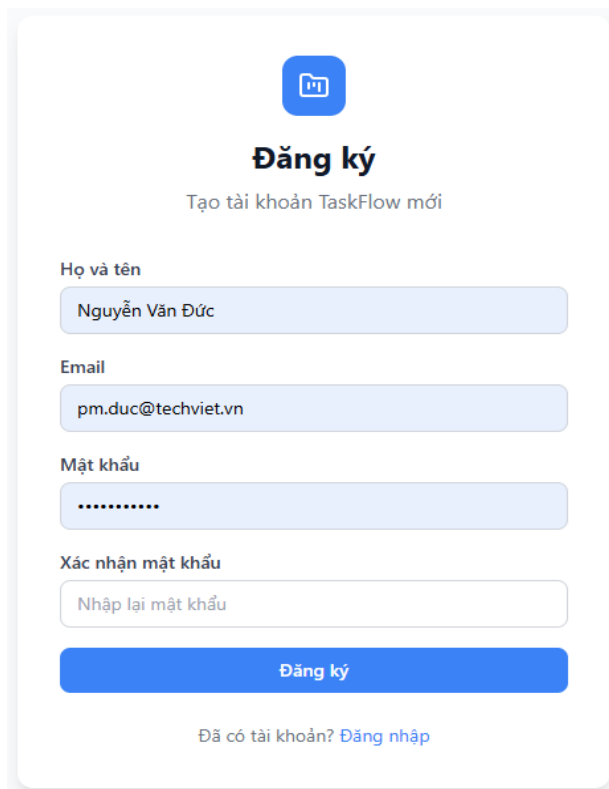
#### **4.1.2.4 Hạn chế của hệ thống**

Bên cạnh các ưu điểm đạt được, hệ thống vẫn còn tồn tại một số hạn chế:

- Chưa hỗ trợ các chức năng nâng cao như báo cáo thống kê chi tiết hoặc phân tích hiệu suất dự án;
- Khả năng mở rộng quy mô lớn chưa được kiểm chứng;
- Chưa tích hợp các cơ chế bảo mật nâng cao như xác thực đa yếu tố hoặc mã hóa dữ liệu chuyên sâu;
- Chưa hỗ trợ tích hợp với các hệ thống bên thứ ba.

Việc nhận diện các hạn chế giúp định hướng cho quá trình cải tiến và phát triển hệ thống trong tương lai.

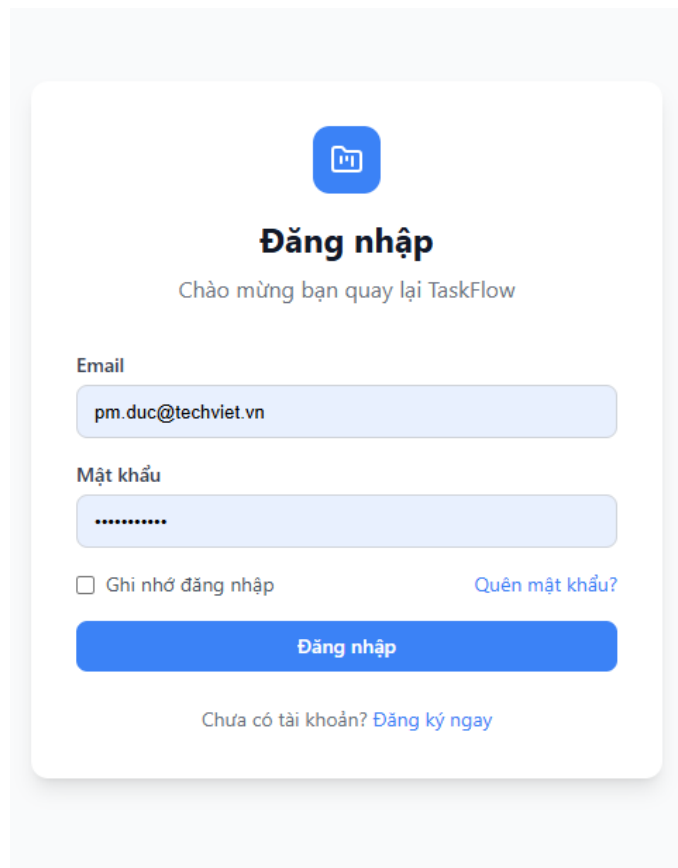
## 4.2. Giao diện trang Đăng ký



Hình 4.1 Giao diện trang Đăng ký

Giao diện đăng ký tài khoản mới của hệ thống TaskFlow được bố trí theo dạng biểu mẫu một cột, đảm bảo sự tập trung và dễ dàng thao tác cho người dùng. Phần tiêu đề “Đăng ký” cùng với dòng mô tả “Tạo tài khoản TaskFlow mới” được đặt ở phía trên, giúp người dùng nhận diện ngay mục đích của giao diện. Bên dưới là các trường nhập liệu bao gồm “Họ và tên”, “Email”, “Mật khẩu” và “Xác nhận mật khẩu”. Trường mật khẩu hiển thị dưới dạng ký tự ẩn nhằm đảm bảo tính bảo mật, trong khi trường xác nhận mật khẩu yêu cầu người dùng nhập lại để kiểm tra tính chính xác. Người dùng tiếp nhận thông tin theo trình tự từ trên xuống dưới, dễ dàng quan sát và điền dữ liệu theo yêu cầu. Nút “Đăng ký” màu xanh dương nổi bật đóng vai trò là hành động chính, khi được nhấn hệ thống sẽ tiến hành kiểm tra dữ liệu đầu vào, phản hồi bằng cách tạo tài khoản mới nếu hợp lệ hoặc hiển thị thông báo lỗi nếu có sai sót. Phía dưới cùng là liên kết “Đã có tài khoản? Đăng nhập”, cho phép người dùng quay lại giao diện đăng nhập trong trường hợp đã có thông tin tài khoản. Giao diện này thể hiện rõ tính trực quan, thao tác đơn giản và phản hồi kịp thời, đáp ứng yêu cầu cơ bản của quá trình khởi tạo tài khoản trong hệ thống quản lý công việc.

### 4.3. Giao diện trang Đăng nhập



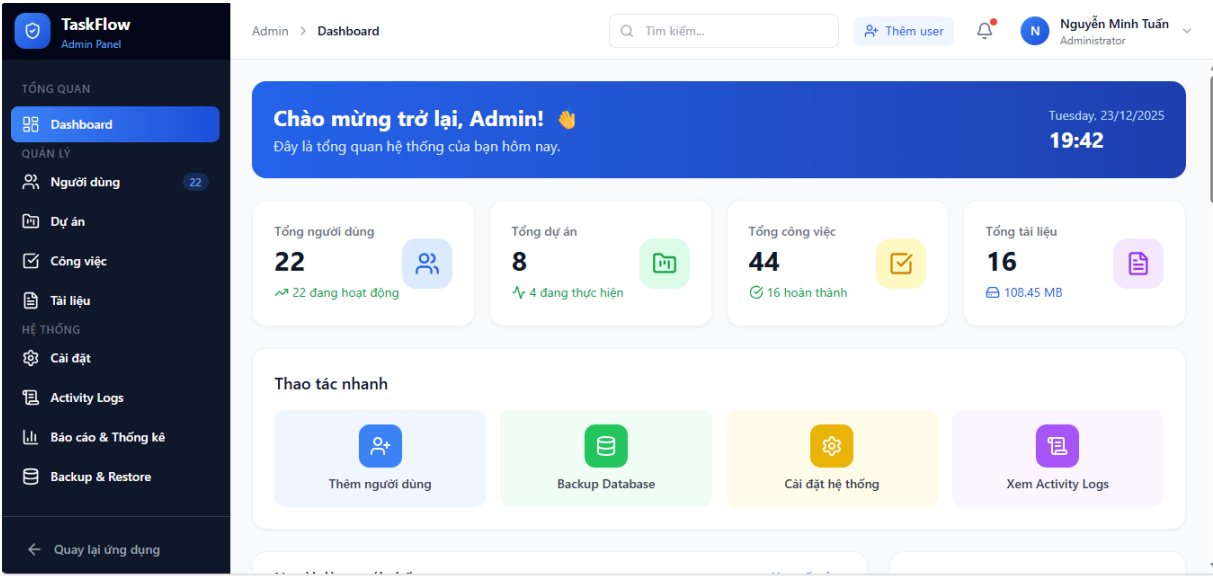
Hình 4.2 Giao diện trang Đăng nhập

Giao diện đăng nhập của hệ thống TaskFlow được bố trí theo cấu trúc một cột dọc, tập trung vào tính đơn giản và hiệu quả trong thao tác người dùng. Phía trên cùng là tiêu đề “Đăng nhập” kèm lời chào “Chào mừng bạn quay lại TaskFlow”, giúp người dùng nhận diện nhanh mục đích của giao diện. Bên dưới là hai trường nhập liệu: trường “Email” và trường “Mật khẩu” hiển thị dưới dạng ký tự ẩn nhằm đảm bảo tính bảo mật. Người dùng có thể đánh dấu vào ô “Ghi nhớ đăng nhập” để lưu thông tin đăng nhập cho các phiên làm việc sau, hoặc chọn liên kết “Quên mật khẩu?” để thực hiện quy trình khôi phục tài khoản. Nút “Đăng nhập” màu xanh dương nổi bật đóng vai trò là điểm nhấn hành động chính, khi người dùng nhấn vào, hệ thống sẽ thực hiện xác thực thông tin và phản hồi bằng cách chuyển hướng đến giao diện chính nếu thành công hoặc hiển thị thông báo lỗi nếu thất bại. Phía dưới cùng là liên kết “Chưa có tài khoản? Đăng ký ngay”, cho phép người dùng mới truy cập vào quy trình tạo tài khoản. Giao diện này đảm bảo tính trực quan, dễ tiếp cận và phản hồi rõ ràng sau mỗi thao tác, phù hợp với yêu cầu xác thực người dùng trong hệ thống quản lý công việc.



#### 4.4. Giao diện các trang của Admin

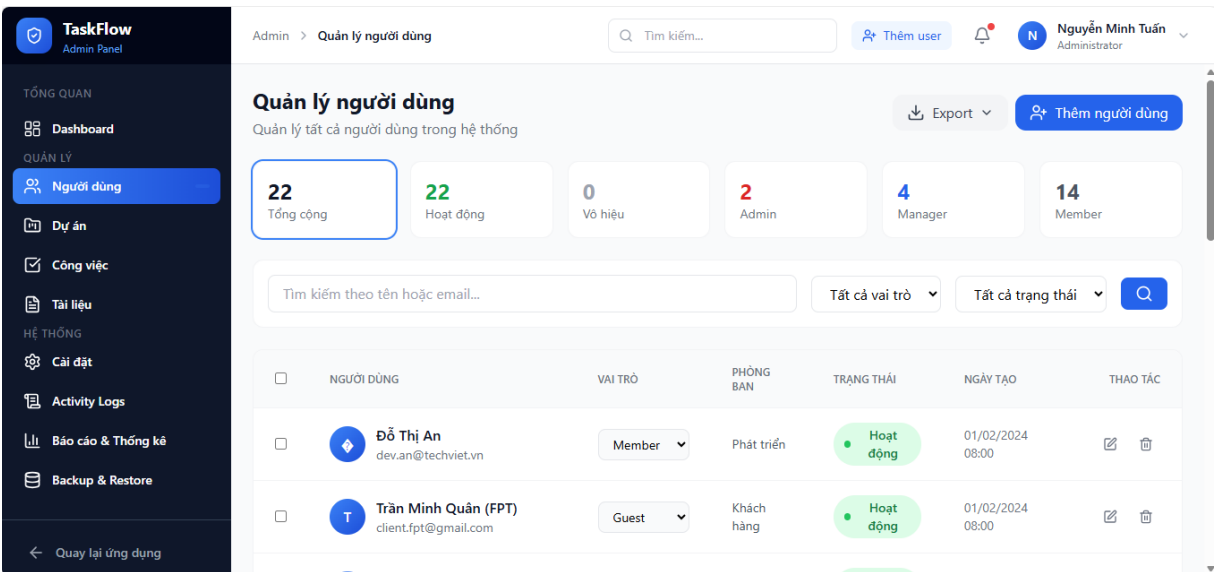
##### 4.4.1 Giao diện trang Admin Dashboard



Hình 4.3 Giao diện trang Admin Dashboard

Giao diện bảng điều khiển quản trị của hệ thống TaskFlow được thiết kế theo bố cục hai phần rõ rệt: thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Thanh điều hướng có nền tối, chứa các mục chức năng như Dashboard, Người dùng, Dự án, Công việc, Tài liệu, Cài đặt, Activity Logs, Báo cáo & Thống kê và Backup & Restore, giúp người quản trị dễ dàng chuyển đổi giữa các phân hệ. Vùng nội dung chính hiển thị lời chào cá nhân hóa dành cho quản trị viên, kèm theo thời gian thực tế. Bên dưới là các thẻ thống kê tổng hợp gồm số lượng người dùng, dự án, công việc và tài liệu, mỗi thẻ đều hiển thị số liệu tổng và trạng thái cụ thể. Người dùng tiếp nhận thông tin trực quan thông qua màu sắc phân biệt và biểu tượng minh họa từng loại dữ liệu. Phía dưới là khu vực “Thao tác nhanh” gồm bốn nút chức năng: Thêm người dùng, Backup Database, Cài đặt hệ thống và Xem Activity Logs. Khi người dùng thao tác vào một nút bất kỳ, hệ thống sẽ điều hướng đến giao diện tương ứng hoặc thực hiện hành động như sao lưu dữ liệu, đồng thời phản hồi bằng cách chuyển trang hoặc hiển thị trạng thái xử lý. Giao diện đảm bảo tính trực quan, dễ thao tác và phản hồi tức thời, phù hợp với vai trò quản trị hệ thống.

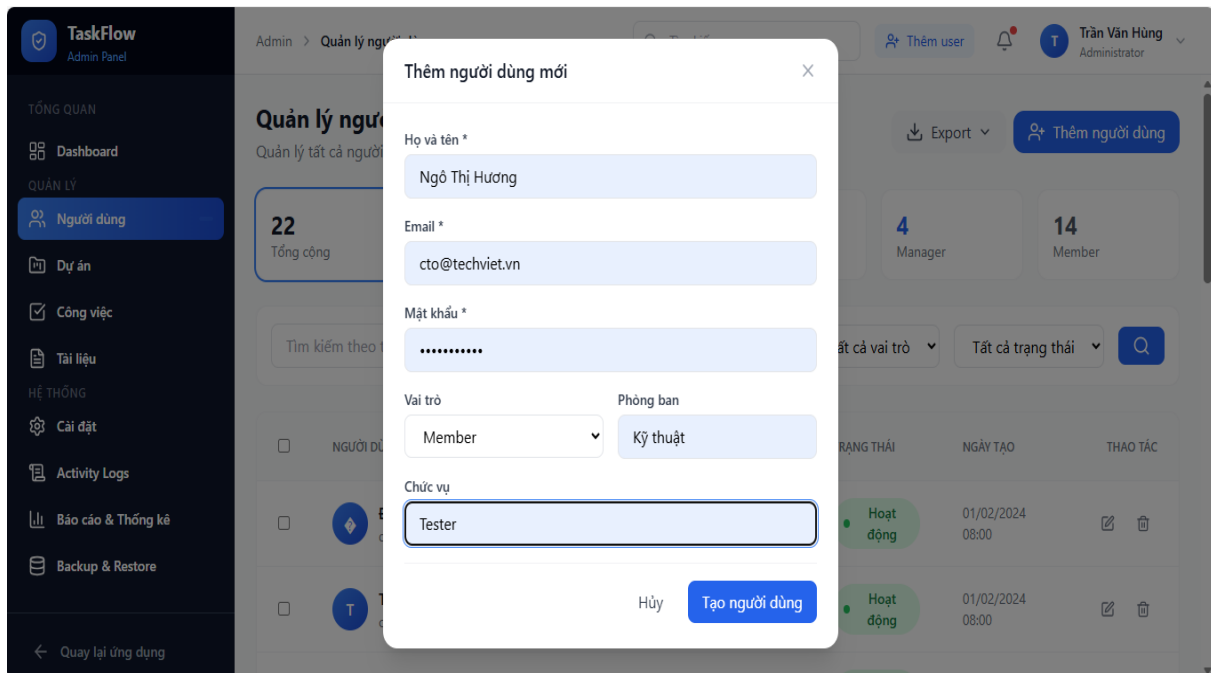
4.4.2 Giao diện trang Admin Quản lý người dùng



Hình 4.4 Giao diện trang Admin Quản lý người dùng

Giao diện quản lý người dùng của hệ thống TaskFlow được thiết kế theo bố cục hai phần: thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu với tiêu đề “Quản lý người dùng” cùng các thẻ thống kê tổng hợp, bao gồm số lượng người dùng tổng cộng, số người dùng đang hoạt động, số người dùng vô hiệu và phân loại theo vai trò như Admin, Manager và Member. Ngay bên dưới là thanh tìm kiếm kết hợp bộ lọc theo vai trò và trạng thái, giúp người quản trị dễ dàng truy xuất thông tin theo tiêu chí cụ thể. Bảng danh sách người dùng được trình bày theo dạng lưới với các cột: Người dùng, Vai trò, Phòng ban, Trạng thái, Thời gian và Thao tác. Trong bảng, mỗi dòng hiển thị chi tiết một người dùng. Người dùng quan sát thông tin trực quan theo từng hàng, dễ dàng nhận diện trạng thái và vai trò qua bố cục rõ ràng. Các thao tác quản trị được thực hiện thông qua nút chức năng ở cột cuối, cho phép chỉnh sửa hoặc xóa thông tin người dùng, đồng thời hệ thống phản hồi bằng cách cập nhật bảng dữ liệu hoặc hiển thị thông báo xác nhận. Ngoài ra, phía trên bảng còn có các nút “Xuất dữ liệu” và “Thêm người dùng”, khi được kích hoạt hệ thống sẽ phản hồi bằng việc tải xuống dữ liệu hoặc mở biểu mẫu thêm mới. Giao diện này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi tức thời, đáp ứng yêu cầu quản lý người dùng trong hệ thống quản trị.

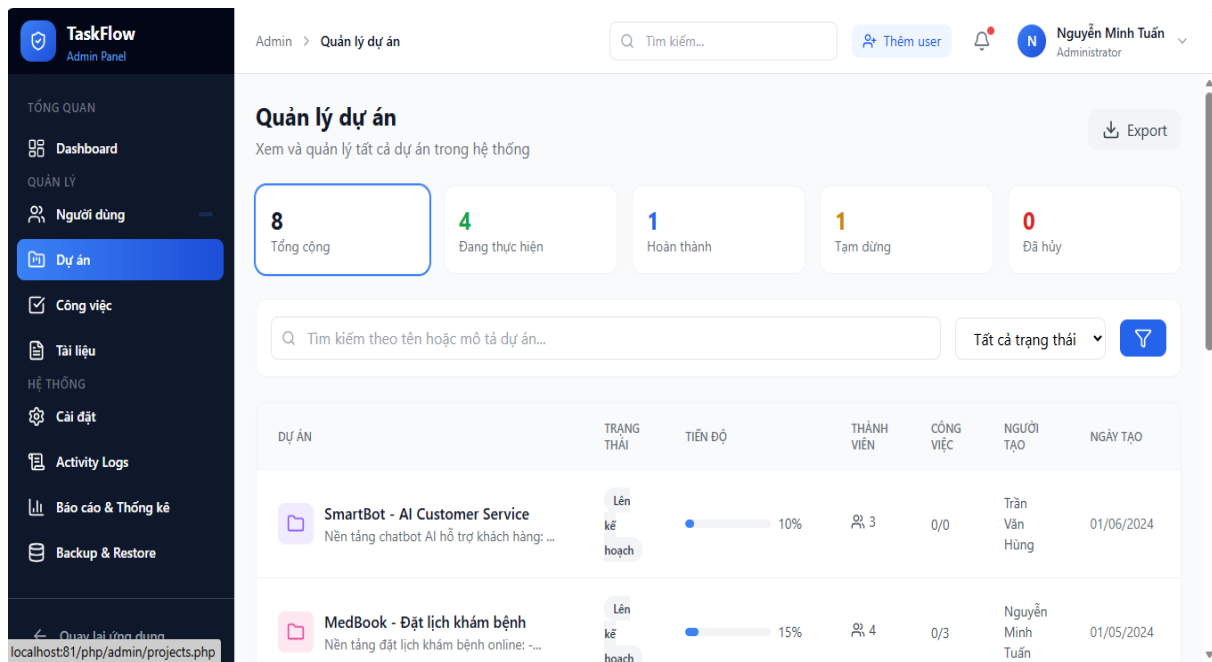
#### 4.4.3 Giao diện chức năng Thêm người dùng



Hình 4.5 Giao diện chức năng Thêm người dùng

Giao diện thêm người dùng mới của hệ thống TaskFlow được hiển thị dưới dạng cửa sổ pop-up nằm ở trung tâm màn hình, đảm bảo sự tập trung của người quản trị vào thao tác hiện tại. Tiêu đề “Thêm người dùng mới” được đặt ở phần đầu biểu mẫu, ngay bên dưới là các trường nhập liệu bắt buộc gồm “Họ và tên”, “Email” và “Mật khẩu”. Các trường bổ sung bao gồm “Vai trò”, “Phòng ban” và “Chức vụ”, giúp định nghĩa rõ ràng vị trí của người dùng trong hệ thống. Người dùng quan sát thông tin theo trình tự từ trên xuống dưới, dễ dàng tiếp nhận và điền dữ liệu theo yêu cầu. Hai nút chức năng “Hủy” và “Tạo người dùng” được bố trí ở cuối biểu mẫu, trong đó nút “Tạo người dùng” đóng vai trò hành động chính. Khi người quản trị nhấn nút này, hệ thống sẽ tiến hành kiểm tra dữ liệu đầu vào, phản hồi bằng cách thêm người dùng mới vào danh sách nếu hợp lệ hoặc hiển thị thông báo lỗi nếu có sai sót. Ngược lại, thao tác “Hủy” sẽ đóng biểu mẫu và trả giao diện về trạng thái trước đó mà không lưu thay đổi. Bộ cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi tức thời, đáp ứng yêu cầu quản lý người dùng trong hệ thống quản trị.

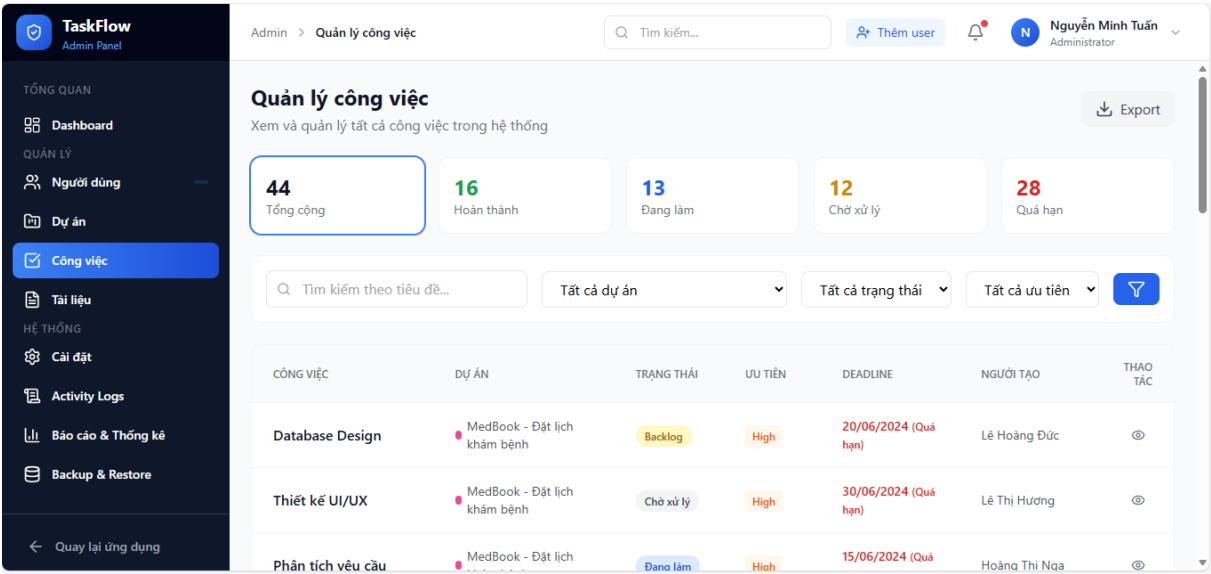
#### 4.4.4 Giao diện trang Admin Quản lý dự án



Hình 4.6 Giao diện trang Admin Quản lý dự án

Giao diện quản lý dự án của hệ thống TaskFlow được bố trí theo dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng tiêu đề “Quản lý dự án” cùng các thẻ thống kê tổng hợp, bao gồm số lượng dự án, số dự án đang thực hiện, hoàn thành, tạm dừng và đã hủy. Ngay bên dưới là thanh tìm kiếm kết hợp bộ lọc trạng thái “Tất cả trạng thái”, cho phép người quản trị dễ dàng lọc và truy xuất thông tin dự án theo tiêu chí cụ thể. Bảng danh sách dự án được trình bày theo dạng lưới, hiển thị chi tiết từng dự án như tên, trạng thái, tiến độ phần trăm, số lượng công việc, số thành viên tham gia và ngày bắt đầu. Người dùng quan sát thông tin trực quan theo từng hàng, dễ dàng nhận diện tiến độ và trạng thái qua bố cục rõ ràng. Khi thao tác tìm kiếm hoặc lọc, hệ thống phản hồi tức thời bằng cách cập nhật danh sách hiển thị theo điều kiện đã chọn. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý dự án trong hệ thống quản trị.

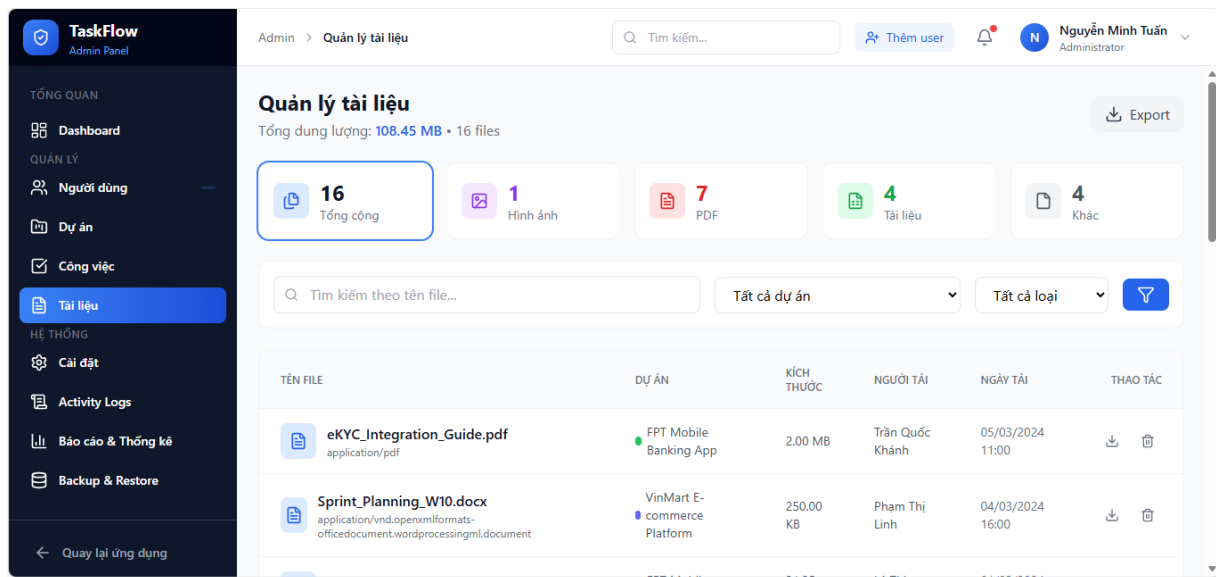
4.4.5 Giao diện trang Admin Quản lý Công việc



Hình 4.7 Giao diện trang Admin Quản lý dự án

Giao diện quản lý công việc của hệ thống TaskFlow được thiết kế theo dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng tiêu đề “Công việc” cùng các thẻ thống kê tổng hợp, bao gồm số lượng công việc, số công việc đã hoàn thành, đang thực hiện, đang chờ và quá hạn. Ngay bên dưới là bảng danh sách công việc được trình bày theo dạng lưới, với các cột thông tin gồm: Công việc, Dự án, Trạng thái, Ưu tiên, Deadline và Người tạo. Người dùng quan sát thông tin trực quan theo từng hàng, dễ dàng nhận diện trạng thái và mức độ ưu tiên qua bố cục rõ ràng. Các thao tác quản trị như lọc theo dự án, trạng thái hoặc ưu tiên, cũng như xuất dữ liệu, đều được thực hiện thông qua các nút chức năng và hệ thống phản hồi tức thời bằng cách cập nhật bảng hiển thị hoặc tải xuống dữ liệu. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý công việc trong hệ thống quản trị.

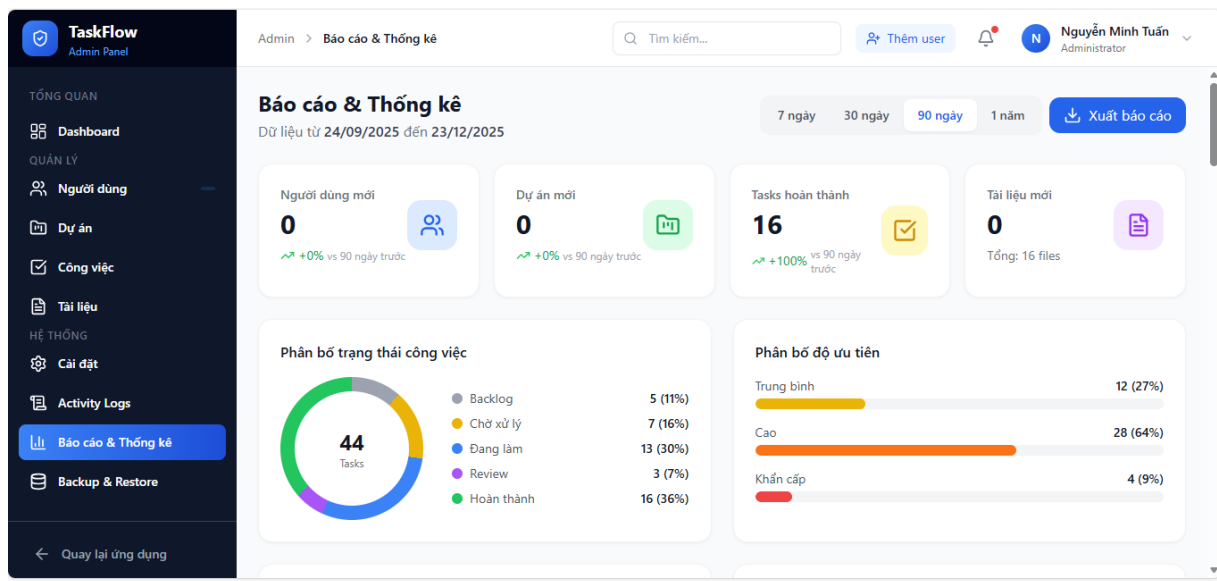
#### 4.4.6 Giao diện trang Admin Quản lý tài liệu



Hình 4.8 Giao diện trang Admin Quản lý tài liệu

Giao diện quản lý tài liệu của hệ thống TaskFlow được bố trí theo dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng tiêu đề “Quản lý tài liệu” cùng các thẻ thống kê tổng hợp, bao gồm số lượng tài liệu, dung lượng lưu trữ và phân loại chi tiết theo định dạng: hình ảnh, tệp PDF, tài liệu văn bản và tệp khác. Ngay bên dưới là thanh tìm kiếm kết hợp bộ lọc theo dự án và loại tệp, cho phép người quản trị dễ dàng truy xuất thông tin theo tiêu chí cụ thể. Bảng danh sách tài liệu được trình bày theo dạng lưới với các cột thông tin gồm: tên tệp, dự án, dung lượng, người tải lên và ngày tải lên. Người dùng quan sát thông tin trực quan theo từng hàng, dễ dàng nhận diện dự án và dung lượng qua bố cục rõ ràng. Các thao tác quản trị như tìm kiếm, lọc hoặc xuất dữ liệu đều được thực hiện thông qua các nút chức năng và hệ thống phản hồi tức thời bằng cách cập nhật bảng hiển thị hoặc tải xuống dữ liệu. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý tài liệu trong hệ thống quản trị.

#### 4.4.7 Giao diện trang Admin Báo cáo và Thống kê

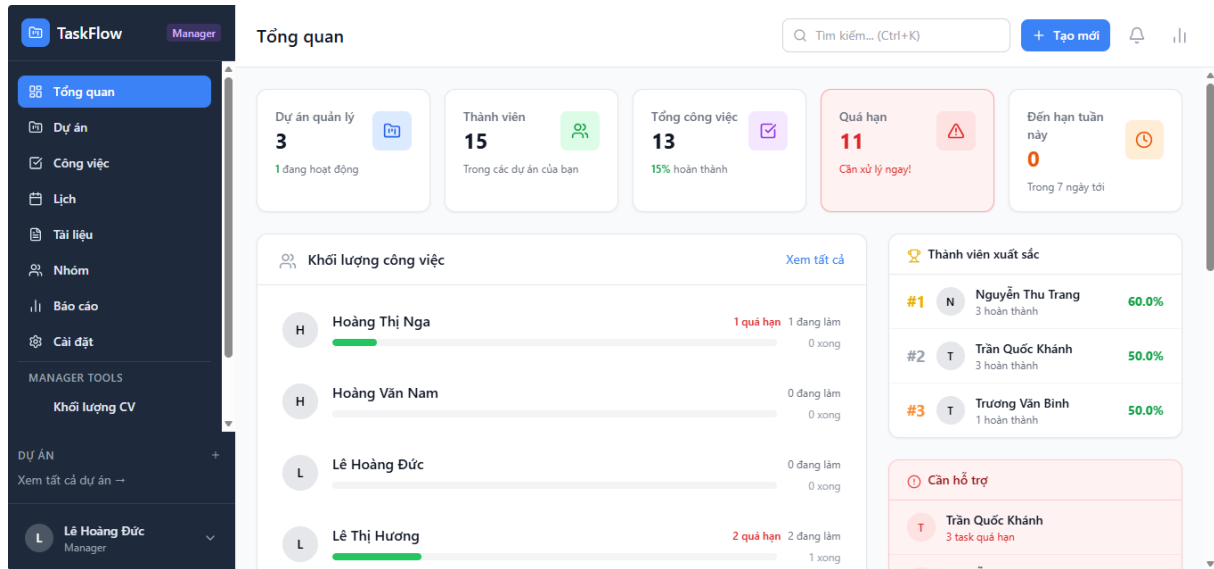


Hình 4.9 Giao diện trang Admin Báo cáo và Thống kê

Giao diện báo cáo và thống kê của hệ thống TaskFlow được bố trí theo dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng tiêu đề “Báo cáo & Thống kê” cùng khoảng thời gian báo cáo. Các thẻ thống kê tổng hợp hiển thị số liệu cụ thể như: người dùng mới, dự án mới, số lượng công việc hoàn thành và tài liệu. Bên dưới là biểu đồ phân bố trạng thái công việc, thể hiện tổng số công việc với các trạng thái: Backlog, Chờ xử lý, Đang làm, Review và Hoàn thành. Tiếp theo là biểu đồ phân bố mức độ ưu tiên, bao gồm Trung bình, Cao và Khẩn cấp. Người dùng quan sát thông tin trực quan thông qua các biểu đồ và số liệu, dễ dàng tiếp nhận tình hình tổng quan của hệ thống. Khi thao tác thay đổi khoảng thời gian hoặc bộ lọc, hệ thống phản hồi tức thời bằng cách cập nhật dữ liệu và biểu đồ hiển thị. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu phân tích và theo dõi hiệu suất trong hệ thống quản trị.

## 4.5. Giao diện các trang của Manager

### 4.5.1 Giao diện trang Manager Dashboard

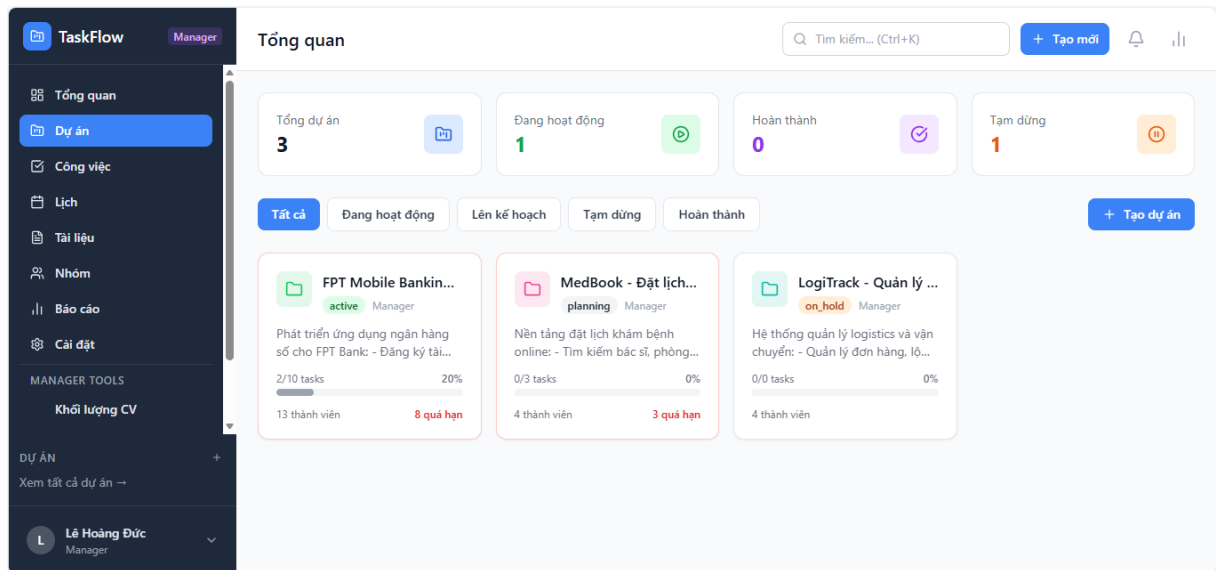


Hình 4.10 Giao diện trang Manager Dashboard

Giao diện tổng quan quản lý nhóm và hiệu suất của hệ thống TaskFlow được thiết kế theo bố cục ba phần: thanh điều hướng bên trái, vùng hiển thị nội dung chính ở giữa và thanh công cụ phía trên. Thanh điều hướng chứa các mục chức năng như Dự án, Công việc, Lịch, Tài liệu, Nhóm, Báo cáo, Cài đặt và các công cụ dành riêng cho quản lý như Khối lượng CV. Vùng nội dung chính hiển thị các chỉ số tổng quan gồm: số lượng dự án quản lý, thành viên nhóm, tổng công việc với tỷ lệ hoàn thành, số công việc quá hạn kèm cảnh báo “Cần xử lý ngay!” và số công việc diễn ra trong tuần. Bên dưới là bảng phân bổ khối lượng công việc theo từng thành viên. Người dùng quan sát thông tin theo từng khối rõ ràng, dễ dàng nhận diện tình trạng nhóm và hiệu suất cá nhân. Các thao tác như tìm kiếm, tạo mới hoặc truy cập hồ sơ người dùng đều được thực hiện qua thanh công cụ phía trên và hệ thống phản hồi tức thời bằng cách cập nhật dữ liệu hoặc chuyển hướng đến giao diện tương ứng. Giao diện này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, hỗ trợ hiệu quả cho vai trò quản lý nhóm trong hệ thống TaskFlow.



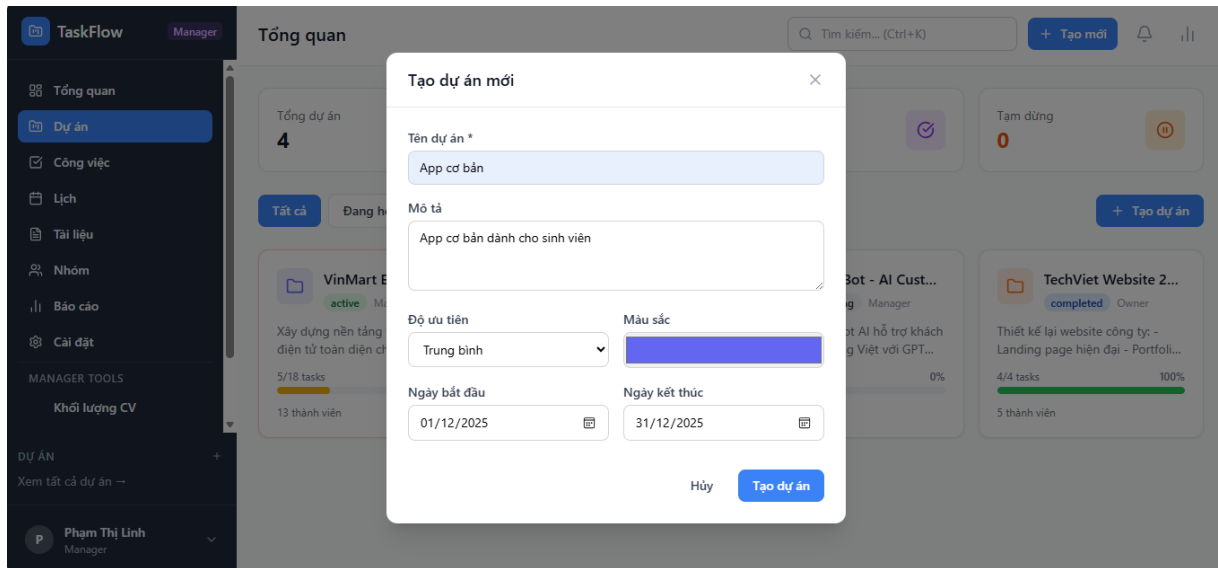
## 4.5.2 Giao diện trang Manager Quản lý dự án



Hình 4.11 Giao diện trang Manager Quản lý dự án

Giao diện tổng quan dự án của hệ thống TaskFlow được bố trí theo dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng tiêu đề “Tổng quan” cùng các thẻ thống kê tổng hợp, bao gồm số lượng dự án, số dự án đang hoạt động, số dự án hoàn thành và số dự án tạm dừng. Ngay bên dưới là các thẻ dự án được trình bày theo dạng thẻ riêng biệt, mỗi thẻ hiển thị chi tiết thông tin dự án như tên, trạng thái, người quản lý, mô tả, số lượng công việc, số thành viên và số công việc quá hạn. Người dùng quan sát thông tin trực quan theo từng thẻ, dễ dàng nhận diện trạng thái và tiến độ của từng dự án. Các thao tác như tìm kiếm hoặc truy cập chi tiết dự án được thực hiện qua các nút chức năng và hệ thống phản hồi tức thời bằng cách hiển thị thông tin chi tiết hoặc cập nhật dữ liệu. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý dự án trong hệ thống quản trị.

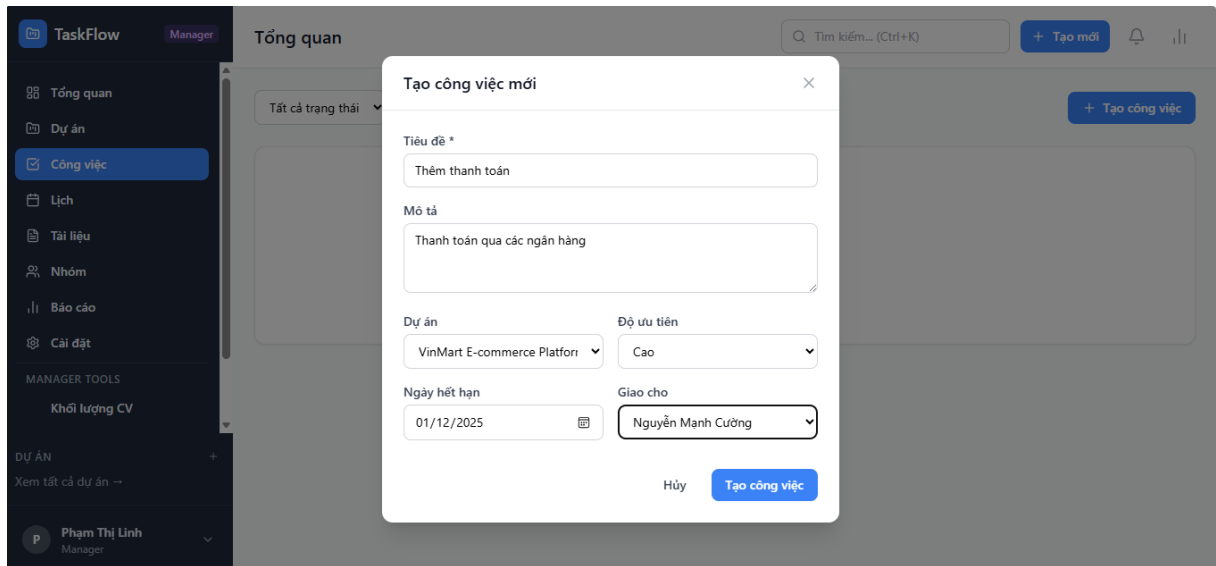
### 4.5.3 Giao diện chức năng Tạo dự án



Hình 4.12 Giao diện chức năng Tạo dự án

Giao diện tạo dự án mới của hệ thống TaskFlow được hiển thị dưới dạng cửa sổ pop-up nằm ở trung tâm màn hình, đảm bảo sự tập trung của người dùng vào thao tác hiện tại. Tiêu đề “Tạo dự án mới” được đặt ở phần đầu biểu mẫu, ngay bên dưới là các trường nhập liệu bao gồm “Tên dự án”, “Mô tả”, “Độ ưu tiên”, “Màu sắc”, “Ngày bắt đầu” và “Ngày kết thúc”. Người dùng tiếp nhận thông tin theo trình tự từ trên xuống dưới, dễ dàng quan sát và điền dữ liệu theo yêu cầu. Hai nút chức năng “Hủy” và “Tạo dự án” được bố trí ở cuối biểu mẫu, trong đó nút “Tạo dự án” đóng vai trò hành động chính. Khi người dùng nhấn nút này, hệ thống sẽ tiến hành kiểm tra dữ liệu đầu vào, phản hồi bằng cách tạo mới dự án và hiển thị trong danh sách nếu hợp lệ, hoặc thông báo lỗi nếu có sai sót. Ngược lại, thao tác “Hủy” sẽ đóng biểu mẫu và trả giao diện về trạng thái trước đó mà không lưu thay đổi. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý dự án trong hệ thống quản trị.

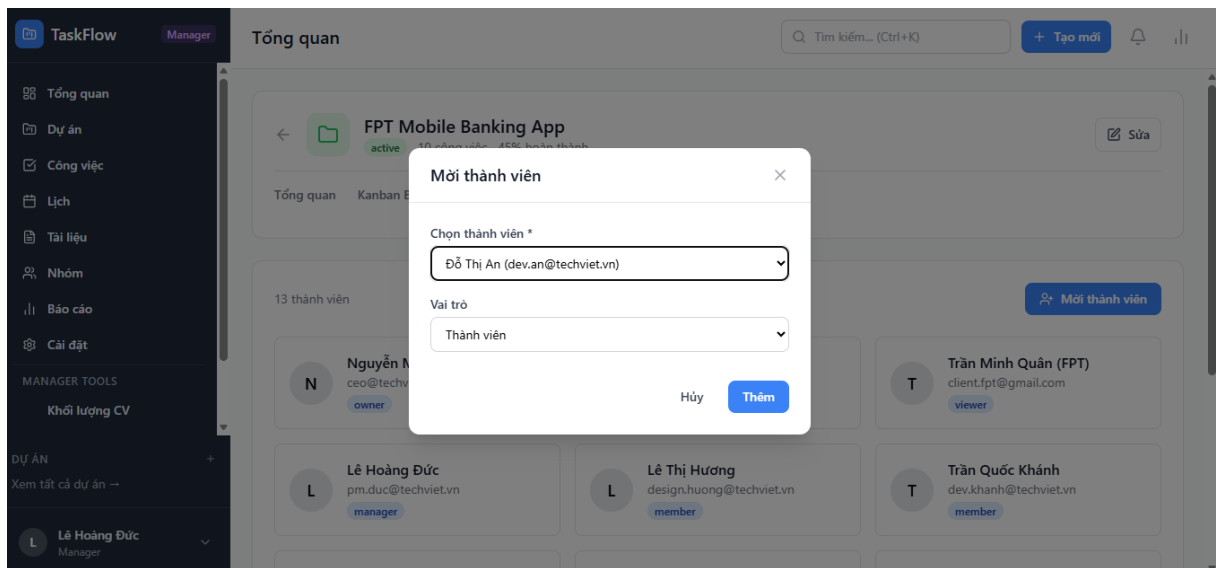
#### 4.5.4 Giao diện chức năng Tạo công việc



Hình 4.13 Giao diện chức năng Tạo công việc

Giao diện tạo công việc mới của hệ thống TaskFlow được hiển thị dưới dạng cửa sổ pop-up, đặt ở trung tâm màn hình nhằm đảm bảo sự tập trung của người dùng vào thao tác hiện tại. Tiêu đề “Tạo công việc mới” được đặt ở phần đầu biểu mẫu, ngay bên dưới là các trường nhập liệu bao gồm “Tiêu đề”, “Mô tả”, “Dự án”, “Độ ưu tiên”, “Ngày hết hạn” và “Giao cho”. Người dùng tiếp nhận thông tin theo trình tự từ trên xuống dưới, dễ dàng quan sát và điền dữ liệu theo yêu cầu. Hai nút chức năng “Hủy” và “Tạo công việc” được bố trí ở cuối biểu mẫu, trong đó nút “Tạo công việc” đóng vai trò hành động chính. Khi người dùng nhấn nút này, hệ thống sẽ tiến hành kiểm tra dữ liệu đầu vào, phản hồi bằng cách thêm công việc mới vào danh sách nếu hợp lệ hoặc hiển thị thông báo lỗi nếu có sai sót. Ngược lại, thao tác “Hủy” sẽ đóng biểu mẫu và trả giao diện về trạng thái trước đó mà không lưu thay đổi. Bộ cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý công việc trong hệ thống quản trị.

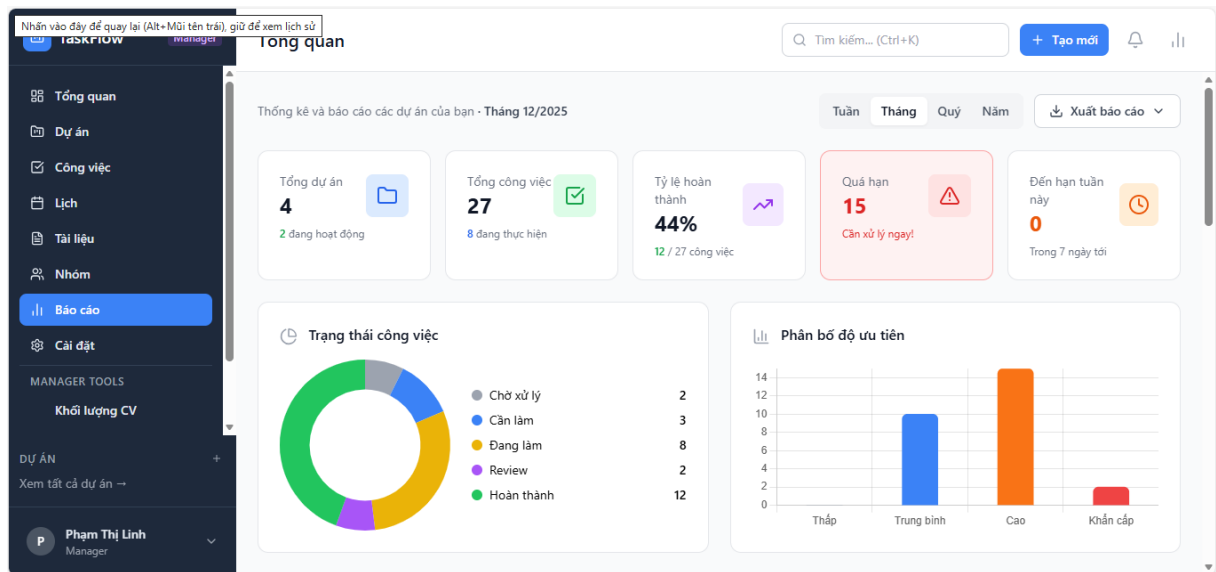
#### 4.5.5 Giao diện chức năng Thêm thành viên vào dự án



Hình 4.14 Giao diện chức năng Thêm thành viên vào dự án

Giao diện mời thành viên trong hệ thống TaskFlow được hiển thị dưới dạng cửa sổ pop-up, đặt ở trung tâm màn hình nhằm đảm bảo sự tập trung của người dùng vào thao tác hiện tại. Tiêu đề “Mời thành viên” được đặt ở phần đầu biểu mẫu, ngay bên dưới là các trường lựa chọn bao gồm “Chọn thành viên” và “Vai trò”. Người dùng tiếp nhận thông tin theo trình tự từ trên xuống dưới, dễ dàng quan sát và thực hiện thao tác lựa chọn. Hai nút chức năng “Hủy” và “Thêm” được bố trí ở cuối biểu mẫu, trong đó nút “Thêm” đóng vai trò hành động chính. Khi người dùng nhấn nút này, hệ thống sẽ phản hồi bằng cách thêm thành viên mới vào danh sách dự án, hiển thị ngay trong bảng quản lý thành viên ở nền giao diện phía sau. Ngược lại, thao tác “Hủy” sẽ đóng biểu mẫu và trả giao diện về trạng thái trước đó mà không lưu thay đổi. Bộ cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý thành viên trong dự án của hệ thống TaskFlow.

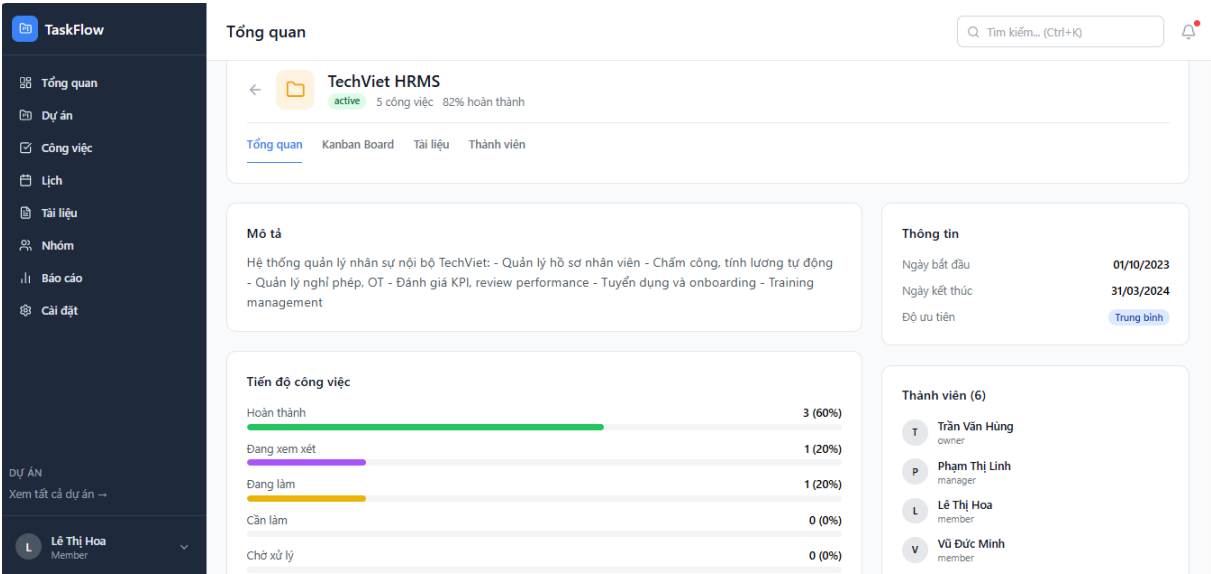
#### 4.5.6 Giao diện trang Manager Báo cáo



Hình 4.15 Giao diện trang Manager Báo cáo

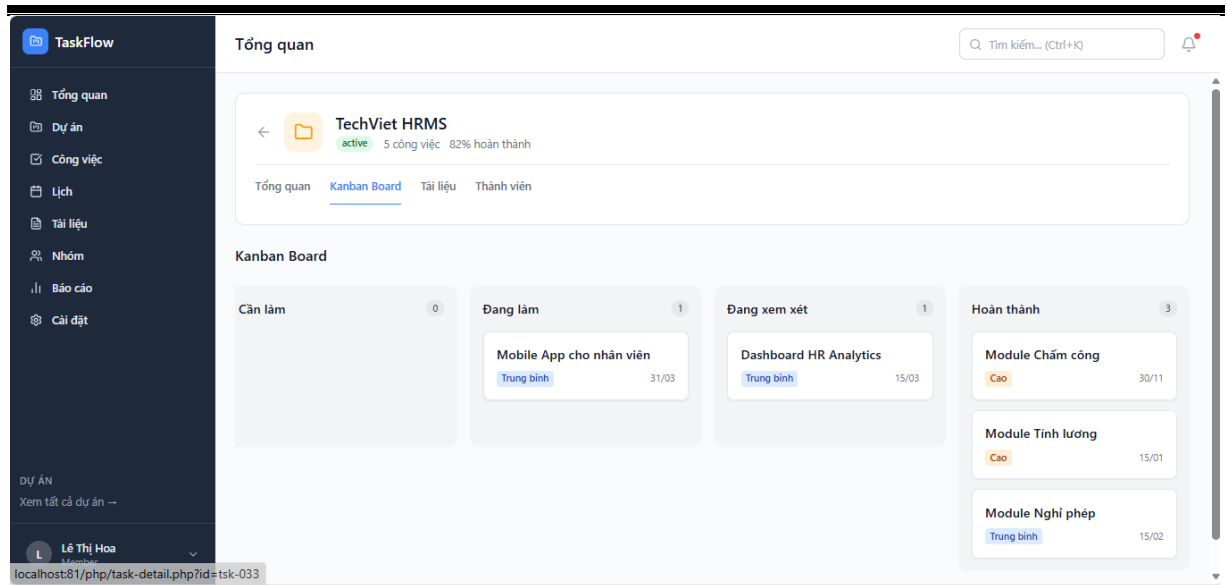
Giao diện tổng quan báo cáo dự án của hệ thống TaskFlow được bố trí theo dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng tiêu đề “Tổng quan” cùng dòng mô tả “Thông tin và báo cáo các dự án của bạn”, giúp người dùng nhận diện rõ phạm vi thời gian của báo cáo. Các thẻ thống kê tổng hợp hiển thị số liệu cụ thể như: tổng dự án, tổng công việc, tỷ lệ hoàn thành, số công việc quá hạn (kèm cảnh báo “Cần xử lý ngay!”) và số công việc đến hạn trong tuần. Bên dưới là các biểu đồ trực quan, bao gồm biểu đồ tròn “Trạng thái công việc” thể hiện phân bố các trạng thái như Chờ xử lý, Cần làm, Đang làm, Review và Hoàn thành; cùng biểu đồ cột “Phân bố độ ưu tiên” hiển thị các mức Thấp, Trung bình, Cao và Khẩn cấp. Người dùng quan sát thông tin theo từng khối rõ ràng, dễ dàng tiếp nhận tình hình tổng quan của dự án và công việc. Các thao tác như chuyển đổi giữa các tab Tuần, Tháng, Quý, Năm hoặc xuất báo cáo được thực hiện qua các nút chức năng và hệ thống phản hồi tức thời bằng cách cập nhật dữ liệu hoặc tải xuống báo cáo. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu phân tích và theo dõi tiến độ dự án trong hệ thống quản trị.

4.6. Giao diện trang Chi tiết dự án



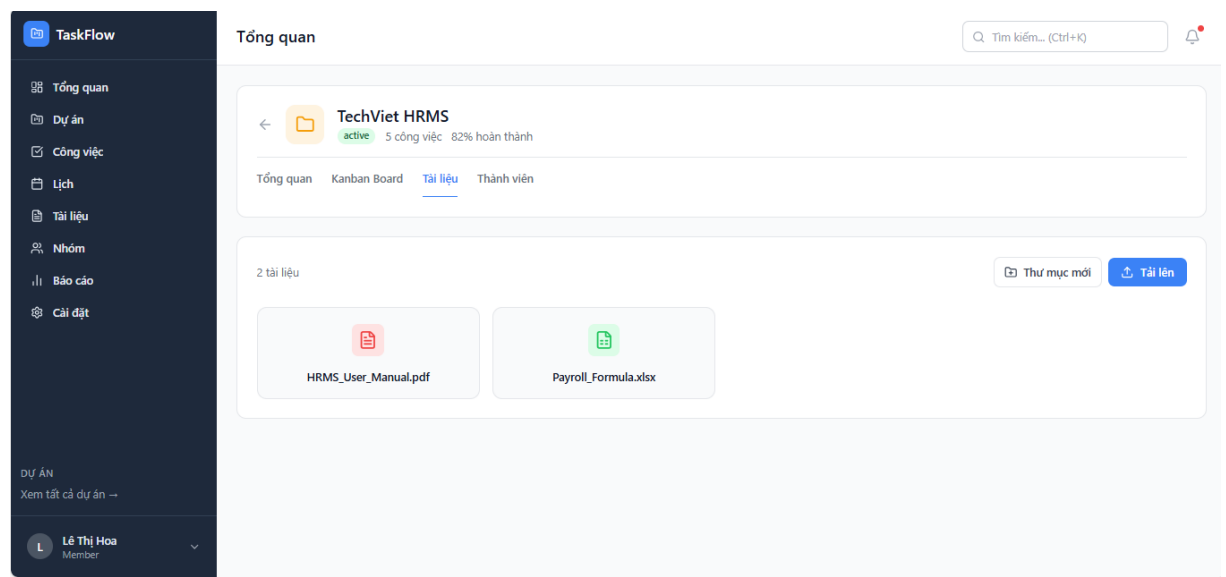
Hình 4.16 Giao diện tổng quan Dự án

Giao diện tổng quan chi tiết dự án trong hệ thống TaskFlow được bố trí theo dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng tiêu đề tên dự án cùng trạng thái, số lượng công việc hiện tại và tỷ lệ hoàn thành. Bên dưới là các tab chức năng gồm “Tổng quan”, “Kanban Board”, “Tài liệu” và “Thành viên”, cho phép người dùng chuyển đổi nhanh giữa các phân hệ quản lý. Khu vực mô tả dự án trình bày chi tiết mục tiêu và phạm vi, bao gồm quản lý hồ sơ nhân viên, đánh giá KPI. Phần thông tin dự án hiển thị ngày bắt đầu, ngày kết thúc và phân loại dự án. Bên dưới là bảng tiến độ công việc, thể hiện phân bố trạng thái: Hoàn thành, Đang xem xét, Đang làm, Chưa làm và Chờ xử lý. Khu vực thành viên hiển thị danh. Người dùng quan sát thông tin theo từng khối rõ ràng, dễ dàng tiếp nhận tình hình tổng quan của dự án. Khi thao tác chuyển đổi tab hoặc cập nhật dữ liệu, hệ thống phản hồi tức thời bằng cách hiển thị nội dung tương ứng. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, đáp ứng yêu cầu quản lý dự án trong hệ thống quản trị.



Hình 4.17 Giao diện bảng Công việc của Dự án

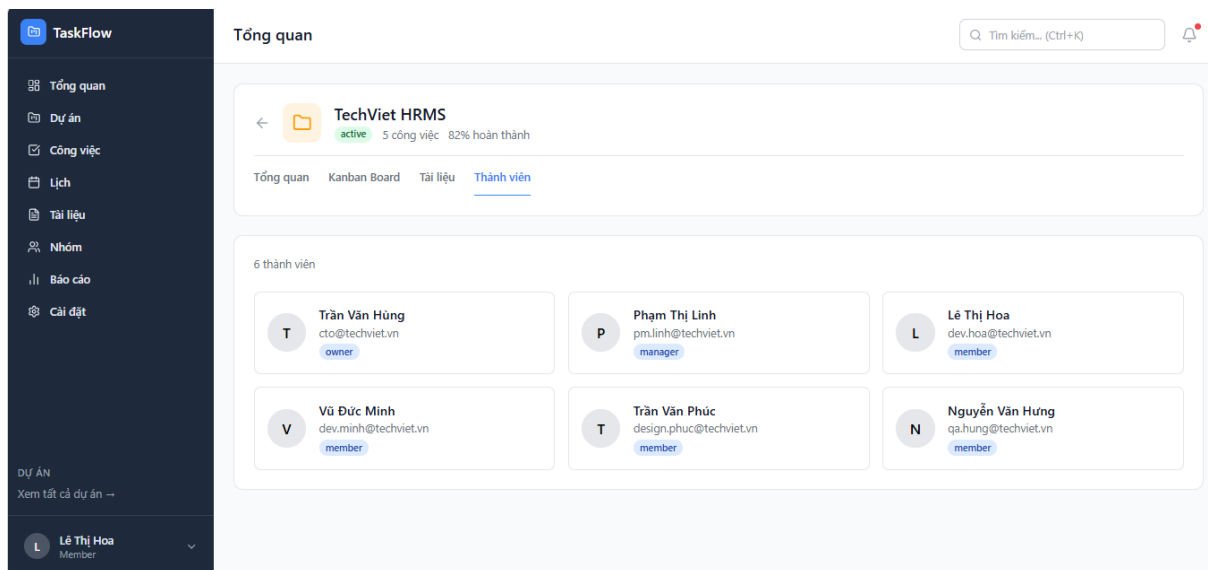
Giao diện Kanban Board của dự án trong hệ thống TaskFlow được thiết kế theo dạng bảng trực quan, chia thành bốn cột chính: “Cần làm”, “Đang làm”, “Đang xem xét” và “Hoàn thành”. Mỗi cột hiển thị các thẻ công việc tương ứng với trạng thái hiện tại, giúp người dùng dễ dàng theo dõi tiến độ và phân bổ công việc. Người dùng quan sát thông tin trực quan qua từng thẻ công việc, dễ dàng nhận diện trạng thái, mức độ ưu tiên và deadline. Khi thao tác kéo thả thẻ giữa các cột hoặc cập nhật thông tin, hệ thống phản hồi tức thời bằng cách thay đổi trạng thái công việc và hiển thị tiến độ mới. Bộ cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, hỗ trợ hiệu quả cho việc quản lý tiến độ dự án theo phương pháp Kanban trong hệ thống TaskFlow.



Hình 4.18 Giao diện Tài liệu của Dự án

Giao diện quản lý tài liệu của dự án trong hệ thống TaskFlow được hiển thị dưới dạng bảng điều khiển, với thanh điều hướng bên trái và vùng hiển thị nội dung chính bên phải. Phần nội dung chính mở đầu bằng các tab chức năng gồm “Tổng quan”, “Kanban Board”, “Tài liệu” và “Thành viên”, trong đó tab “Tài liệu” đang được chọn. Bên trong, hệ thống hiển thị danh sách tài liệu liên quan đến dự án. Ngoài ra, người dùng có thể tạo thư mục mới hoặc tải lên tệp thông qua các nút chức năng được bố trí ở khu vực này.

Người dùng quan sát thông tin trực quan theo từng hàng, dễ dàng nhận diện tên tệp, định dạng và mục đích sử dụng. Khi thao tác tải lên hoặc tạo thư mục, hệ thống phản hồi tức thời bằng cách cập nhật danh sách hiển thị, đảm bảo dữ liệu được quản lý tập trung và đồng bộ. Thanh điều hướng bên trái cung cấp các mục chức năng như Tổng quan, Dự án, Công việc, Lịch, Tài liệu, Nhóm, Báo cáo và Cài đặt, giúp người dùng chuyển đổi nhanh giữa các phân hệ quản lý. Bố cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, hỗ trợ hiệu quả cho việc quản lý tài liệu trong dự án.



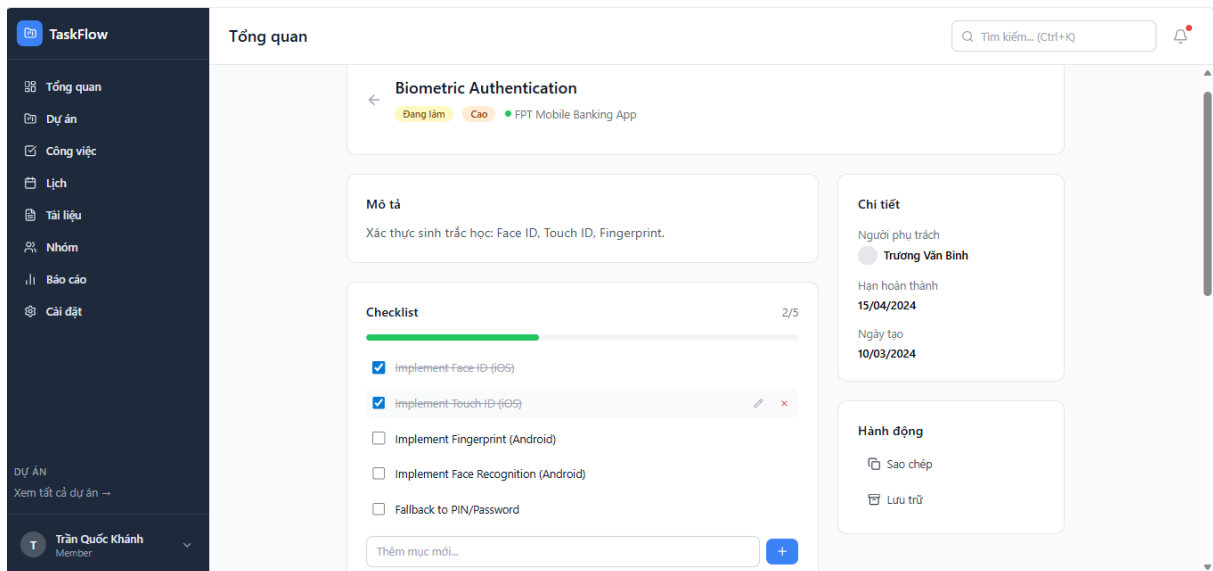
Hình 4.19 Giao diện Thành viên của Dự án

Giao diện quản lý thành viên của dự án trong hệ thống TaskFlow được hiển thị dưới dạng bảng danh sách, nằm trong tab “Thành viên”. Phần nội dung chính mở đầu bằng tiêu đề dự án cùng trạng thái, hiển thị số lượng công việc và tỷ lệ hoàn thành. Bên dưới là danh sách thành viên dự án, mỗi dòng thể hiện thông tin chi tiết gồm họ tên, địa chỉ email và vai trò.



Người dùng quan sát thông tin trực quan theo từng hàng, dễ dàng nhận diện vai trò và phân bổ trách nhiệm trong nhóm. Khi thao tác thêm, chỉnh sửa hoặc xóa thành viên, hệ thống phản hồi tức thời bằng cách cập nhật danh sách hiển thị, đảm bảo dữ liệu luôn đồng bộ. Thanh điều hướng bên trái cung cấp các mục chức năng như Tổng quan, Dự án, Công việc, Lịch, Tài liệu, Nhóm, Báo cáo và Cài đặt, giúp người dùng chuyển đổi nhanh giữa các phân hệ quản lý. Bộ cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, hỗ trợ hiệu quả cho việc quản lý thành viên và phối hợp nhóm trong dự án.

#### 4.7. Giao diện trang Chi tiết Công việc



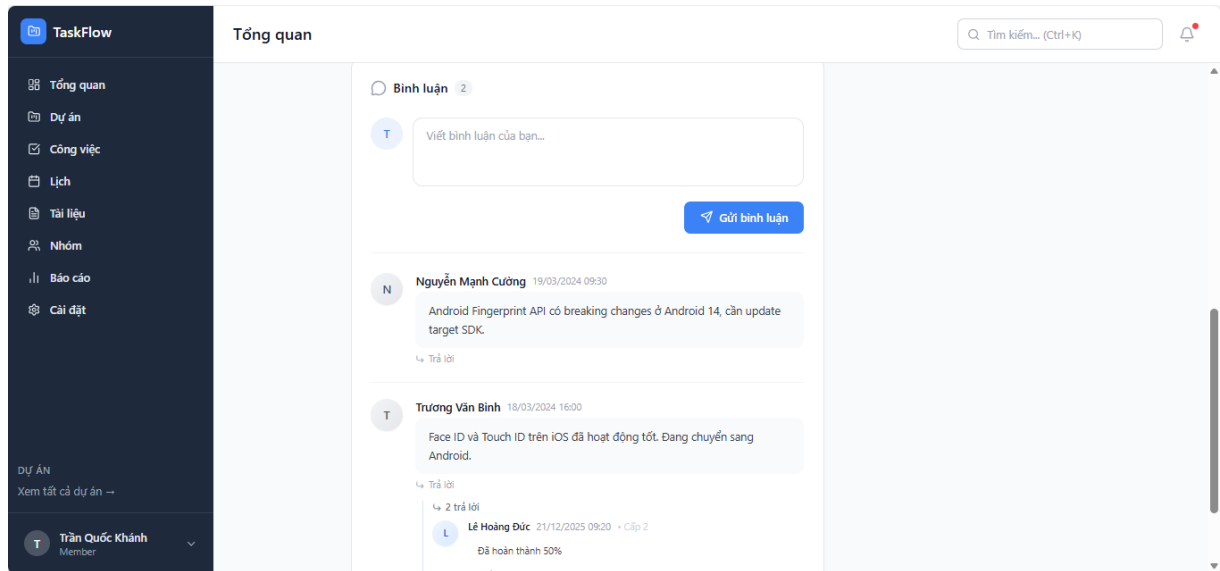
Hình 4.20 Giao diện chi tiết Công việc

Giao diện quản lý dự án trong hệ thống TaskFlow được hiển thị dưới dạng bảng điều khiển chi tiết, tập trung vào tiến độ và các hạng mục công việc. Phần nội dung chính mở đầu bằng tiêu đề dự án cùng trạng thái, giúp người dùng nhận diện rõ mục tiêu phát triển. Bên dưới là danh sách checklist công việc, hiển thị hạng mục công việc nhỏ. Người dùng có thể thêm hạng mục mới thông qua trường nhập liệu “Thêm mục mới...”, đảm bảo khả năng mở rộng và cập nhật liên tục.

Phần chi tiết công việc hiển thị thông tin người phụ trách, ngày bắt đầu và ngày kết thúc. Các nút chức năng “Sao chép” và “Lưu trữ” được bố trí ở cuối, cho phép người quản trị thực hiện thao tác quản lý nhanh chóng. Người dùng quan sát thông tin trực quan theo từng khối, dễ dàng nhận diện tiến độ và trạng thái công việc. Khi thao

tác đánh dấu hoàn thành hoặc thêm mới, hệ thống phản hồi tức thời bằng cách cập nhật checklist và tỷ lệ hoàn thành. Bộ cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, hỗ trợ hiệu quả cho việc quản lý tiến độ phát triển tính năng xác thực sinh trắc học trong dự án.

#### 4.8. Giao diện chức năng Bình luận



Hình 4.21 Giao diện Bình luận của Công việc

Giao diện bình luận và trao đổi nhóm trong hệ thống TaskFlow được hiển thị dưới dạng bảng thảo luận, nằm trong phần “Tổng quan” của dự án. Phần nội dung chính mở đầu bằng ô nhập liệu “Bình luận” cùng nút chức năng “Gửi bình luận”, cho phép người dùng thêm ý kiến hoặc cập nhật tiến độ công việc. Bên dưới là danh sách các bình luận đã được gửi, hiển thị theo thứ tự thời gian, mỗi bình luận bao gồm tên người dùng, thời gian đăng và nội dung.

Người dùng quan sát thông tin trực quan theo từng dòng, dễ dàng nhận diện nội dung trao đổi và tiến độ công việc. Khi thao tác gửi bình luận mới, hệ thống phản hồi tức thời bằng cách hiển thị bình luận trong danh sách, đảm bảo tính liên tục và minh bạch trong quá trình cộng tác. Bộ cục này đảm bảo tính trực quan, thao tác thuận tiện và phản hồi nhanh chóng, hỗ trợ hiệu quả cho việc giao tiếp và phối hợp nhóm trong hệ thống quản trị TaskFlow.

## CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết luận

Trong khuôn khổ đồ án chuyên ngành, đề tài “Xây dựng hệ thống quản lý dự án và cộng tác nhóm” đã được nghiên cứu, phân tích và triển khai dựa trên các kiến thức về phân tích – thiết kế hệ thống thông tin, lập trình web và quản lý cơ sở dữ liệu. Đồ án đã hoàn thành mục tiêu đề ra là xây dựng một hệ thống quản lý dự án hoạt động trên nền tảng web, hỗ trợ các chức năng cốt lõi phục vụ quản lý và cộng tác nhóm.

Hệ thống cho phép quản lý người dùng theo vai trò, quản lý dự án, phân công và theo dõi công việc, đồng thời hỗ trợ trao đổi thông tin giữa các thành viên trong nhóm. Việc áp dụng mô hình MVC giúp hệ thống có cấu trúc rõ ràng, dễ bảo trì và thuận lợi cho việc mở rộng trong tương lai.

Kết quả thực hiện cho thấy hệ thống hoạt động ổn định trong phạm vi thử nghiệm, đáp ứng tốt các yêu cầu chức năng và phi chức năng đã đề ra. Thông qua đồ án, người thực hiện đã có cơ hội vận dụng kiến thức lý thuyết vào thực tiễn, nâng cao kỹ năng phân tích nghiệp vụ, thiết kế hệ thống và phát triển ứng dụng web.

### 5.2. Hạn chế của đề tài

Bên cạnh những kết quả đạt được, đồ án vẫn còn tồn tại một số hạn chế nhất định do giới hạn về thời gian và phạm vi nghiên cứu:

- Hệ thống chưa hỗ trợ các chức năng nâng cao như báo cáo thống kê chi tiết, phân tích hiệu suất dự án;
- Chưa triển khai các cơ chế bảo mật nâng cao như xác thực đa yếu tố hoặc phân quyền chi tiết theo từng hành động;
- Khả năng mở rộng cho các dự án có quy mô lớn chưa được kiểm chứng;
- Chưa tích hợp với các hệ thống hoặc dịch vụ bên thứ ba.

Việc nhận diện rõ các hạn chế là cơ sở quan trọng cho việc cải tiến và phát triển hệ thống trong các giai đoạn tiếp theo.

### 5.3. Hướng phát triển

Trong tương lai, hệ thống quản lý dự án và cộng tác nhóm có thể được tiếp tục phát triển theo các hướng sau:

- Triển khai các cơ chế bảo mật nâng cao như xác thực đa yếu tố, mã hóa dữ liệu và phân quyền chi tiết theo vai trò;
- Tích hợp các công nghệ thời gian thực (Realtime) để hỗ trợ thông báo và cập nhật trạng thái công việc tức thời;
- Tối ưu hiệu năng và khả năng mở rộng để đáp ứng nhu cầu sử dụng ở quy mô lớn hơn.

Những hướng phát triển này sẽ giúp hệ thống hoàn thiện hơn, đáp ứng tốt hơn nhu cầu quản lý dự án và cộng tác nhóm trong thực tế.

## DANH MỤC TÀI LIỆU THAM KHẢO

- [1] H. Kerzner, *Project management: a systems approach to planning, scheduling, and controlling*, 10th ed. Hoboken, N.J: John Wiley & Sons, 2009.
- [2] B. biên tập TopDev, “PHP là gì? Tất tần tật những điều bạn cần biết về PHP”, TopDev. Truy cập: 13 Tháng Mười-Một 2025. [Online]. Available at: <https://topdev.vn/blog/ngon-ngu-lap-trinh-php-la-gi-tat-tan-tat-nhung-dieu-ban-can-biet-ve-php/>
- [3] “PHP: Introduction - Manual”. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://www.php.net/manual/en/introduction.php>
- [4] A. Forbes, *The Joy of PHP: A Beginner's Guide to Programming Interactive Web Applications with PHP and mySQL*. BeakCheck LLC.
- [5] R. Nixon, *Learning PHP, MySQL, JavaScript, CSS & HTML5: A Step-by-Step Guide to Creating Dynamic Websites*. O'Reilly Media, Inc., 2014.
- [6] TopDev, “MySQL là gì? Giới thiệu và hướng dẫn cài đặt MySQL chi tiết”, TopDev. Truy cập: 13 Tháng Mười-Một 2025. [Online]. Available at: <https://topdev.vn/blog/gioi-thieu-ve-mysql/>
- [7] “Học MySQL cơ bản với lộ trình và tài liệu tổng quan – ITviec Blog”. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://itviec.com/blog/lo-trinh-hoc-mysql-co-ban/>
- [8] A. Beaulieu, *Learning SQL*. O'Reilly Media, Inc., 2005.
- [9] “XAMPP là gì? Hướng dẫn tải, cài đặt và sử dụng phần mềm XAMPP trên Windows & Linux”. Truy cập: 3 Tháng Chạp 2025. [Online]. Available at: <https://fptcloud.com/xampp-la-gi/>
- [10] B. biên tập TopDev, “Xampp là gì? Cách sử dụng và cài đặt XAMPP”, TopDev. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://topdev.vn/blog/cai-dat-xampp/>
- [11] Hoà N. Đ., “Tailwind CSS Là Gì? Cách cài đặt và sử dụng Tailwind CSS chi tiết”, LANIT - Nhà cung cấp dịch vụ Máy Chủ, Hosting, Vps giá rẻ - uy tín. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://lanit.com.vn/tailwind-css-la-gi.html>

- [12] “Tailwind CSS: Hướng dẫn 3 cách thiết lập Tailwind chi tiết – ITviec Blog”. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://itviec.com/blog/tailwind-css/>
- [13] “Thư viện Alpine.js - thay thế nhẹ nhàng hơn cho JQuery”, QS. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://quicksite.vn/thu-vien-alpine-js-thay-the-nhe-nhang-hon-cho-jquery/>
- [14] “Alpinejs - Nếu React là ‘quá thừa’”. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://viblo.asia/p/alpinejs-neu-react-la-qua-thua-yMnKMjaQZ7P>
- [15] “Mô Hình MVC | PDF”, Scribd. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://www.scribd.com/document/502792421/Mo-Hinh-MVC>
- [16] monamedia, “Tìm Hiểu Chi Tiết Về Mô Hình MVC Trong Lập Trình Từ A - Z”, Mona Software. Truy cập: 28 Tháng Chạp 2025. [Online]. Available at: <https://mona.software/mo-hinh-mvc/>