

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 DIỄN ĐÀN TIN HỌC

Giảng viên hướng dẫn:
TS. Đoàn Phước Miền

Sinh viên thực hiện:
Họ tên: Trần Lâm Phú Đức
Mã số sinh viên: 110122054
Lớp: DA22TTD

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 DIỄN ĐÀN TIN HỌC

Giảng viên hướng dẫn:
TS. Đoàn Phước Miền

Sinh viên thực hiện:
Họ tên: Trần Lâm Phú Đức
Mã số sinh viên: 110122054
Lớp: DA22TTD

Vĩnh Long, tháng 12 năm 2025

[illegible]

Giảng viên hướng dẫn
(Ký tên và ghi rõ họ tên)

[illegible]

Giảng viên hướng dẫn
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trước tiên, tôi xin cảm ơn sâu sắc đến trường Đại học Trà Vinh đã tạo điều kiện để tôi thực hiện dự án này, những người đã đưa ra những sự hỗ trợ, góp ý, cũng như là sự giúp đỡ thầy Đoàn Phước Miên – người đóng vai trò quan trọng trong việc phát triển và làm cho dự án thành công tốt đẹp.

Tôi biết ơn tất cả những người đã dành thời công sức, kiến thức và thời gian quý báu của mình để hỗ trợ tôi. Điều đó sẽ là nguồn động lực giúp tôi phát triển bản thân, những ý kiến, lời khuyên đó sẽ góp phần tạo nên một môi trường làm việc tích cực.

Xin chân thành cảm ơn thầy Đoàn Phước Miên và trường Đại học Trà Vinh đã giúp đỡ, do đây là dự án đầu tiên cũng như hạn chế về mặt kiến thức lẫn kinh nghiệm nên chắc chắn sẽ xảy ra những sai sót nhỏ, vậy nên tôi mong rằng sẽ nhận được những góp ý và lời khuyên, tôi sẽ đưa những lời khuyên, góp ý đó vào dự án này để tạo nên một thành công tốt đẹp .

Trân trọng,

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN	1
1.1. Giới thiệu bài toán	1
1.2. Hiện trạng và thách thức	1
1.3. Giải pháp đề xuất	1
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT	3
2.1. Kiến trúc hệ thống	3
2.1.1 Kiến trúc ứng dụng web	3
2.1.2 Áp dụng vào hệ thống	4
2.2. Công nghệ Frontend (Giao diện người dùng)	4
2.2.1 ReactJS	4
2.2.2 Các thư viện hỗ trợ Frontend	4
2.3. Công nghệ Backend (Xử lý nghiệp vụ)	5
2.3.1 NodeJS	5
2.3.2 ExpressJS	5
2.4. Hệ quản trị Cơ sở dữ liệu (MySQL)	5
2.5. Các kỹ thuật và thư viện hỗ trợ khác	6
2.5.1 JSON Web Token (JWT) - Cơ chế xác thực	6
2.5.2 Bcryptjs - Mã hóa mật khẩu	6
2.5.3 Multer - Quản lý tải lên tập tin	6
2.5.4 Nodemailer - Gửi Email	7
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	8
3.1. Mô tả vấn đề	8
3.2. Xác định yêu cầu người dùng	8
3.3. Phân loại người dùng	9
3.4. Phân tích thiết kế hệ thống	10

3.4.1 Đặc tả yêu cầu hệ thống	10
3.4.1.1 Yêu cầu chức năng	10
3.4.1.1.1 Nhóm chức năng Tài khoản & Cá nhân hóa.....	10
3.4.1.1.2 Nhóm chức năng Quản lý Nội dung (Bài viết & Tài liệu).....	11
3.4.1.1.3 Nhóm chức năng Tương tác & Cộng đồng.....	11
3.4.1.1.4 Nhóm chức năng Quản trị (Admin)	11
3.4.1.2 Yêu cầu phi chức năng.....	12
3.4.2 Kiến trúc hệ thống.....	12
3.4.2.1 Lớp giao diện người dùng (Presentation Layer):.....	12
3.4.2.2 Lớp xử lý nghiệp vụ (Business Logic Layer):.....	12
3.4.2.3 Lớp dữ liệu (Data Layer):.....	13
3.4.3 Thiết kế dữ liệu.....	14
3.4.3.1 Mô hình ERD	14
3.4.3.2 Danh sách các thực thể và mối kết hợp	14
3.4.3.3 Chi tiết các thực thể.....	15
3.4.4 Thiết kế xử lý.....	22
3.4.4.1 Sơ đồ phân cấp chức năng.....	22
3.4.4.2 Đặc tả quy trình xử lý các chức năng chính.....	23
3.4.5 Thiết kế giao diện.....	25
3.4.5.1 Sơ đồ Use Case	25
3.4.5.1.1 Quan hệ kế thừa giữa các actor.....	25
3.4.5.1.2 Use Case của người dùng khách (Guest)	26
3.4.5.1.3 Use Case của người dùng đã đăng nhập (User).....	27
3.4.5.1.4 Use Case của quản trị viên (Admin).....	29
3.4.5.2 Sơ đồ website	31
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	33

4.1. Dữ liệu thử nghiệm.....	33
4.1.1 Dữ liệu người dùng (Users)	33
4.1.2 Dữ liệu danh mục (Categories)	33
4.1.3 Dữ liệu bài viết (Posts).....	33
4.1.4 Dữ liệu tài liệu (Documents)	34
4.1.5 Dữ liệu Bình luận (Comments)	34
4.1.6 Dữ liệu Báo cáo vi phạm (Reports)	35
4.1.7 Dữ liệu Thẻ (Tags).....	35
4.1.8 Dữ liệu Cấm người dùng (User Bans).....	36
4.2. Kết quả giao diện thực nghiệm.....	36
4.2.1 Trang chủ và Xác thực	36
4.2.2 Chức năng Quên mật khẩu.....	38
4.2.3 Chức năng Diễn đàn (Forum).....	40
4.2.4 Chức năng Thư viện tài liệu	43
4.2.5 Trang cá nhân và Cài đặt.....	45
4.2.6 Chức năng báo cáo.....	46
4.2.7 Trang Quản trị (Admin Dashboard)	49
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	53
5.1. Kết luận.....	53
5.1.1 Về mặt lý thuyết:.....	53
5.1.2 Về mặt thực tiễn (Sản phẩm):.....	53
5.2. Hạn chế	53
5.3. Hướng phát triển.....	54
5.3.1 Nâng cấp công nghệ:	54
5.3.2 Mở rộng chức năng:.....	54
5.3.3 Đa nền tảng:.....	54

DANH MỤC HÌNH ẢNH

Hình 2.1. Kiến trúc phân lớp (Layered Architecture).....	3
Hình 3.1. Mô hình ERD.....	14
Hình 3.2. Quan hệ kế thừa giữa các tác nhân.....	25
Hình 3.3. Use Case của người dùng khách	26
Hình 3.4. Use Case của người dùng đã đăng nhập.....	28
Hình 3.5. Use Case của quản trị viên.....	29
Hình 3.6. Sơ đồ website.....	31
Hình 4.1. Dữ liệu bảng Users.....	33
Hình 4.2. Dữ liệu bảng Categories	33
Hình 4.3. Dữ liệu bảng Posts	34
Hình 4.4. Dữ liệu bảng Documents.....	34
Hình 4.5. Dữ liệu bảng Comments	35
Hình 4.6. Dữ liệu bảng Reports	35
Hình 4.7. Dữ liệu bảng Tags	36
Hình 4.8. Dữ liệu bảng User Bans.....	36
Hình 4.9. Giao diện Trang chủ (Home).....	37
Hình 4.10. Giao diện Đăng nhập.....	37
Hình 4.11. Giao diện Đăng ký tài khoản.....	38
Hình 4.12. Giao diện nhập Email xác thực	39
Hình 4.13. Giao diện nhận mã OTP qua Email	39
Hình 4.14. Giao diện nhập OTP và thiết lập mật khẩu mới.....	40
Hình 4.15. Giao diện Danh sách bài viết và Bộ lọc.....	41
Hình 4.16. Giao diện Soạn thảo bài viết mới (Markdown Editor):.....	41
Hình 4.17. Giao diện Chi tiết bài viết.....	42
Hình 4.18. Tương tác bài viết	42

Hình 4.19. Giao diện comment bài viết.....	43
Hình 4.20. Giao diện Danh sách tài liệu.....	44
Hình 4.21. Giao diện Upload tài liệu	44
Hình 4.22. Giao diện Hồ sơ cá nhân (Profile)	45
Hình 4.23. Giao diện Hồ sơ cá nhân (Profile)	46
Hình 4.24. Báo cáo người dùng.....	47
Hình 4.25. Báo cáo bài viết.....	47
Hình 4.26. Báo cáo tài liệu.....	48
Hình 4.27. Giao diện xem Báo cáo của tôi.....	48
Hình 4.28. Giao diện Dashboard thống kê.....	49
Hình 4.29. Giao diện Quản lý bài viết.....	50
Hình 4.30. Giao diện Quản lý tài liệu.....	50
Hình 4.31. Giao diện Quản lý danh mục.....	51
Hình 4.32. Giao diện Xử lý báo cáo vi phạm.....	51
Hình 4.33. Giao diện Quản lý bình luận.....	52
Hình 4.34. Giao diện Gửi thông báo hệ thống.....	52

DANH MỤC BẢNG

Bảng 3.1. Bảng thực thể.....	14
Bảng 3.2. Bảng người dùng.....	16
Bảng 3.3. Bảng danh mục.....	16
Bảng 3.4. Bảng bài viết.....	17
Bảng 3.5. Bảng tài liệu.....	18
Bảng 3.6. Bảng bình luận	18
Bảng 3.7. Bảng thông báo	19
Bảng 3.8. Bảng báo cáo vi phạm.....	19
Bảng 3.9. Bảng lệnh cấm.....	20
Bảng 3.10. Bảng lưu bài viết.....	20
Bảng 3.11. Bảng thẻ	20
Bảng 3.12. Bảng liên kết bài viết với thẻ	21
Bảng 3.13. Bảng theo dõi người dùng	21
Bảng 3.14. Bảng Theo dõi danh mục.....	21
Bảng 3.15. Bảng cảnh báo.....	22
Bảng 3.16. Bảng thống kê	22
Bảng 3.17. Mô tả Use Case của người dùng khách.....	27
Bảng 3.18. Mô tả Use Case Use Case của người dùng đã đăng nhập	28
Bảng 3.19. Mô tả Use Case Use Case của quản trị viên.....	30

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

1. Vấn đề cần nghiên cứu

Trong đồ án này, tôi tập trung nghiên cứu và xây dựng một hệ thống diễn đàn tin học dành cho sinh viên, nhằm hỗ trợ việc chia sẻ tài liệu học tập và trao đổi hỏi đáp. Vấn đề tôi hướng đến là tạo ra một nền tảng trực quan, dễ sử dụng và phù hợp với nhu cầu trao đổi kiến thức trong môi trường học thuật.

2. Các hướng tiếp cận

Để tiếp cận vấn đề, tiến hành tìm hiểu và phân tích một số trang web, ứng dụng web có chức năng tương tự nhằm nắm bắt cách họ xây dựng hệ thống, bố cục giao diện, các thành phần chức năng cần thiết và công nghệ mà họ sử dụng. Từ đó, xác định mô hình phù hợp để triển khai trong đồ án.

3. Cách giải quyết vấn đề

Giải pháp được lựa chọn là xây dựng ứng dụng web dựa trên NodeJS cho backend, ReactJS cho frontend và MySQL cho cơ sở dữ liệu. Tiến hành thiết kế cấu trúc hệ thống, lập trình các chức năng chính, kiểm thử trong quá trình phát triển và sửa lỗi nhằm đảm bảo hệ thống hoạt động ổn định và đáp ứng yêu cầu đề ra.

4. Kết quả đạt được

Kết quả của quá trình nghiên cứu và phát triển là tôi đã xây dựng được một hệ thống diễn đàn tin học cho phép người dùng tạo và chia sẻ nội dung một cách thuận tiện. Hệ thống hỗ trợ soạn thảo và đăng bài viết bằng Markdown, giúp người dùng trình bày nội dung rõ ràng, trực quan và dễ định dạng. Bên cạnh đó, hệ thống cũng tích hợp chức năng đăng tải tài liệu học tập, cho phép người dùng upload, lưu trữ và chia sẻ tài liệu cho cộng đồng sinh viên. Hệ thống này không chỉ hỗ trợ người dùng trao đổi kiến thức thuận tiện hơn mà còn là nền tảng để tôi có thể tiếp tục mở rộng và phát triển các tính năng nâng cao trong tương lai.

MỞ ĐẦU

1. Lý do chọn đề tài

Trong kỷ nguyên số, Công nghệ thông tin (CNTT) đang phát triển với tốc độ vũ bão, kéo theo nhu cầu cập nhật kiến thức liên tục của sinh viên và lập trình viên. Việc tự học và trao đổi kiến thức qua các cộng đồng trực tuyến đóng vai trò then chốt trong việc nâng cao kỹ năng chuyên môn.

Tuy nhiên, các nền tảng mạng xã hội hiện nay thường chứa quá nhiều thông tin tạp nham, thiếu các công cụ chuyên biệt để chia sẻ mã nguồn (source code) hay tài liệu học tập chuyên ngành một cách hệ thống. Xuất phát từ thực tế đó, đề tài "Xây dựng Diễn đàn Tin học" được lựa chọn thực hiện nhằm tạo ra một môi trường học thuật chuyên biệt. Tại đây, sinh viên có thể thảo luận, chia sẻ các bài viết kỹ thuật (hỗ trợ định dạng code) và trao đổi tài liệu học tập chất lượng cao.

2. Mục đích nghiên cứu

2.1. Mục đích chung: Xây dựng một hệ thống diễn đàn trực tuyến hoàn chỉnh, cho phép người dùng chia sẻ kiến thức, đặt câu hỏi, thảo luận và quản lý tài liệu cá nhân. Hệ thống hướng đến sự ổn định, bảo mật và trải nghiệm người dùng tốt trên đa nền tảng.

2.2. Mục đích cụ thể

Về chức năng: Xây dựng các chức năng cốt lõi như đăng ký/dăng nhập, soạn thảo bài viết hỗ trợ Markdown (Syntax Highlighting), kho tài liệu số (PDF, DOCX) và công cụ quản trị nội dung.

Về công nghệ: Làm chủ quy trình phát triển ứng dụng web Full-stack với ReactJS (Frontend), NodeJS (Backend) và MySQL (Database); áp dụng chuẩn RESTful API và bảo mật với JWT.

3. Đối tượng nghiên cứu

3.1 Đối tượng sử dụng:

Khách (Guest): Người dùng vắng lai muốn tìm kiếm thông tin và tài liệu.

Thành viên (User): Sinh viên, giảng viên có nhu cầu chia sẻ kiến thức, viết bài, thảo luận và tải tài liệu.

Quản trị viên (Admin): Người chịu trách nhiệm vận hành, kiểm duyệt nội dung và quản lý thành viên.

3.2 Đối tượng kỹ thuật:

Các công nghệ lập trình web hiện đại: ReactJS, NodeJS, ExpressJS.

Hệ quản trị cơ sở dữ liệu MySQL.

Các kỹ thuật bảo mật và xử lý dữ liệu: JWT, Bcrypt, Multer .

4. Phạm vi nghiên cứu

Phạm vi chức năng: Tập trung vào các module: Quản lý tài khoản, Quản lý bài viết (Markdown), Hệ thống bình luận đa cấp, Thư viện tài liệu, Thông báo và Báo cáo vi phạm .

Phạm vi công nghệ: Sử dụng React.js 18, Bootstrap 5 cho giao diện; Node.js, Express.js cho backend; MySQL cho cơ sở dữ liệu.

Phạm vi triển khai: Phát triển và kiểm thử trên môi trường local, đảm bảo tương thích trên trình duyệt máy tính và thiết bị di động .

5. Phương pháp nghiên cứu

Đồ án áp dụng các phương pháp sau:

Phương pháp phân tích và tổng hợp: Khảo sát các diễn đàn hiện có để xác định yêu cầu chức năng và trải nghiệm người dùng .

Phương pháp thiết kế hệ thống: Xây dựng sơ đồ Use Case, sơ đồ thực thể kết hợp (ERD) và kiến trúc 3 tầng .

Phương pháp lập trình và kiểm thử: Phát triển từng module chức năng và thực hiện kiểm thử để đảm bảo tính ổn định.

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu bài toán

Trong môi trường học thuật, đặc biệt là lĩnh vực Công nghệ thông tin, nhu cầu trao đổi kiến thức, chia sẻ mã nguồn (source code) và tài liệu tham khảo là vô cùng lớn. Sinh viên cần một không gian không chỉ để hỏi đáp mà còn để lưu trữ và chia sẻ các tài nguyên học tập quý giá.

Vấn đề đặt ra là làm thế nào để xây dựng một nền tảng tập trung, nơi kiến thức được tổ chức khoa học, dễ dàng tìm kiếm và hỗ trợ tốt các định dạng đặc thù của ngành CNTT như đoạn mã lập trình hay các tệp tài liệu chuyên ngành.

1.2. Hiện trạng và thách thức

Hiện nay, sinh viên thường sử dụng các mạng xã hội phổ biến (Facebook, Zalo) để trao đổi học tập. Tuy nhiên, các nền tảng này tồn tại nhiều hạn chế:

- Thông tin rời rạc, khó tìm kiếm: Các bài viết hỏi đáp hay tài liệu chia sẻ thường bị trôi nhanh trên bảng tin (Newsfeed), gây khó khăn cho việc tra cứu lại sau này.
- Hạn chế về định dạng: Các nền tảng chat hoặc mạng xã hội thông thường không hỗ trợ hiển thị mã nguồn (Syntax Highlighting), khiến việc đọc, chia sẻ và gỡ lỗi (debug) code trở nên khó khăn và thiếu trực quan.
- Thiếu tính chuyên biệt: Thông tin thường bị trộn lẫn với các nội dung giải trí, thiếu sự phân loại rõ ràng giữa các chủ đề chuyên môn (Web, Mobile, Thuật toán,...).

1.3. Giải pháp đề xuất

Để giải quyết các vấn đề nêu trên, đồ án tập trung nghiên cứu và xây dựng hệ thống "Diễn đàn Tin học" với các đặc điểm sau:

- Nền tảng tập trung: Tạo ra một kho tri thức có tổ chức, nơi bài viết và tài liệu được phân loại theo danh mục và thẻ (tags) rõ ràng.
- Hỗ trợ chuyên sâu cho CNTT: Tích hợp trình soạn thảo Markdown mạnh mẽ, cho phép hiển thị code với màu sắc cú pháp chuẩn, giúp việc chia sẻ kiến thức lập trình trở nên dễ dàng và chuyên nghiệp.
- Quản lý tài liệu hiệu quả: Xây dựng module thư viện tài liệu riêng biệt, cho phép người dùng đóng góp và tải xuống các tài liệu học tập an toàn.

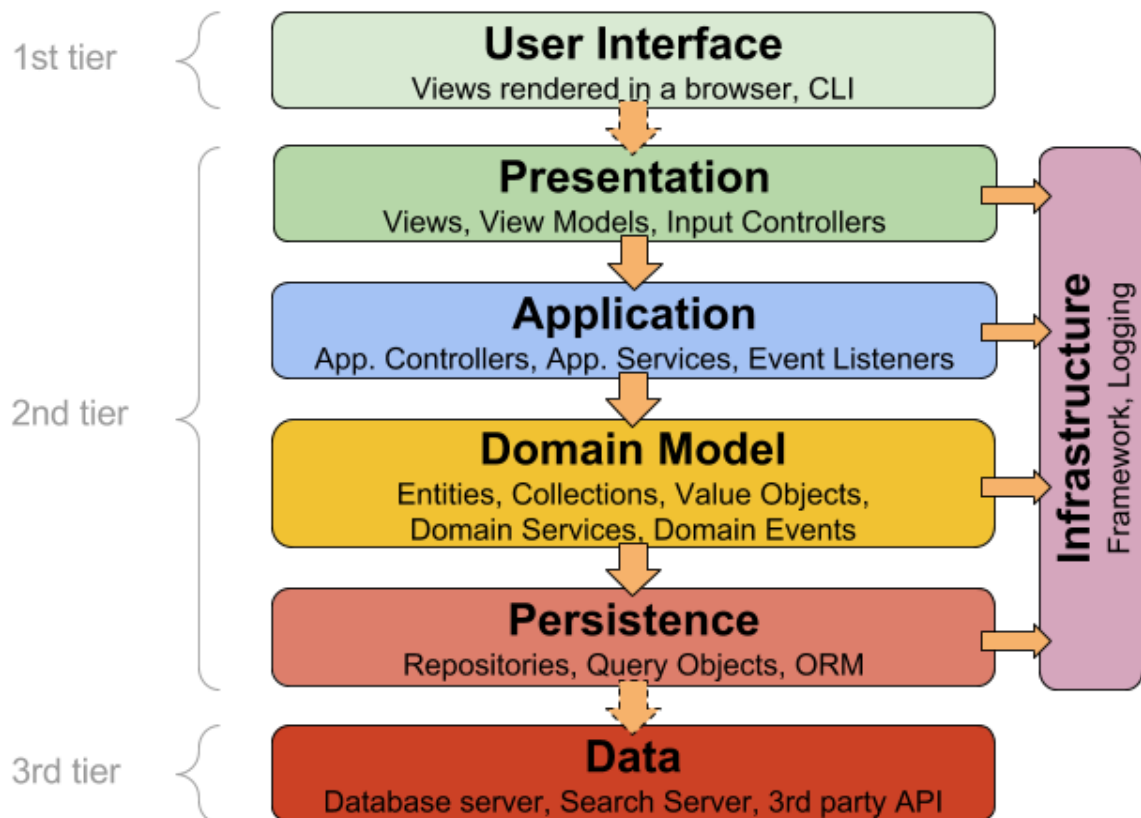
- Hệ thống được xây dựng dựa trên kiến trúc Full-stack Javascript (ReactJS và NodeJS) kết hợp với MySQL, hướng tới mục tiêu tạo ra một diễn đàn hiện đại, tốc độ cao và dễ dàng mở rộng trong tương lai.

CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

2.1. Kiến trúc hệ thống

2.1.1 Kiến trúc ứng dụng web

Để đảm bảo tính mở rộng, bảo trì và bảo mật, hệ thống "Diễn đàn Tin học" được thiết kế dựa trên kiến trúc 3 tầng (Three-Tier Architecture). Đây là mô hình kiến trúc phổ biến trong phát triển phần mềm hiện đại, giúp tách biệt các thành phần của ứng dụng thành các lớp logic độc lập [1].



Hình 2.1. Kiến trúc phân lớp (Layered Architecture)

Kiến trúc phân lớp chia hệ thống thành ba lớp chính:

Lớp giao diện người dùng (Presentation Layer): Là tầng tương tác trực tiếp với người dùng (Client-side). Nhiệm vụ của tầng này là hiển thị thông tin và thu thập dữ liệu đầu vào từ người dùng.

Lớp xử lý nghiệp vụ (Business Logic Layer): Đóng vai trò trung gian (Server-side), xử lý các yêu cầu từ tầng giao diện, thực thi các quy tắc nghiệp vụ và tính toán trước khi truy xuất hoặc lưu trữ dữ liệu.

Lớp dữ liệu (Data Layer): Là nơi lưu trữ và quản lý dữ liệu bền vững của hệ thống (Database). Tầng này đảm bảo tính toàn vẹn và an toàn của dữ liệu.

2.1.2 Áp dụng vào hệ thống

Trong đề án này, kiến trúc 3 tầng được hiện thực hóa thông qua bộ công nghệ (Tech stack) như sau:

- **Frontend (Presentation Tier):** Sử dụng thư viện **ReactJS** để xây dựng giao diện người dùng đơn trang (SPA), giao tiếp với Server qua chuẩn RESTful API.
- **Backend (Logic Tier):** Sử dụng **NodeJS** và **ExpressJS** để xây dựng API, xử lý xác thực người dùng và các logic nghiệp vụ của diễn đàn.
- **Database (Data Tier):** Sử dụng hệ quản trị cơ sở dữ liệu **MySQL** để lưu trữ thông tin người dùng, bài viết và tài liệu.

2.2. Công nghệ Frontend (Giao diện người dùng)

2.2.1 ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook, chuyên dùng để xây dựng giao diện người dùng (UI), đặc biệt là cho các ứng dụng đơn trang (Single Page Applications) [2].

Những đặc điểm nổi bật của ReactJS được áp dụng trong đề án:

- **Component-based (Hướng thành phần):** React cho phép chia nhỏ giao diện phức tạp thành các thành phần (Components) độc lập, có thể tái sử dụng. Ví dụ: *Header*, *Footer*, *PostItem*, *CommentSection*. Điều này giúp mã nguồn dễ quản lý và bảo trì.
- **Virtual DOM (DOM ảo):** React sử dụng một bản sao ảo của DOM thực. Khi trạng thái (State) của ứng dụng thay đổi, React sẽ so sánh DOM ảo mới với DOM ảo cũ và chỉ cập nhật những phần thay đổi lên DOM thực. Cơ chế này giúp tối ưu hóa hiệu suất hiển thị [2].
- **JSX (JavaScript XML):** Cho phép viết mã HTML ngay trong JavaScript, giúp việc mô tả giao diện trở nên trực quan hơn.

2.2.2 Các thư viện hỗ trợ Frontend

Để hoàn thiện các chức năng của diễn đàn, đề án sử dụng thêm các thư viện hỗ trợ sau:

- **React Router DOM:** Quản lý định tuyến (Routing) trong ứng dụng SPA, cho phép chuyển đổi giữa các trang (Trang chủ, Đăng nhập, Chi tiết bài viết) mà không cần tải lại toàn bộ trang web [3].
- **Axios:** Thư viện HTTP Client dựa trên Promise, dùng để gửi các yêu cầu (Request) từ React đến Backend API một cách dễ dàng và hiệu quả hơn so với `fetch` mặc định.
- **React Markdown & Syntax Highlighter:** Do đặc thù là diễn đàn tin học, việc hiển thị mã nguồn (source code) là rất quan trọng. Thư viện này giúp chuyển đổi văn bản Markdown thành HTML và tô màu cú pháp (highlight) cho các đoạn code, giúp người đọc dễ dàng theo dõi.

2.3. Công nghệ Backend (Xử lý nghiệp vụ)

2.3.1 NodeJS

NodeJS là một môi trường chạy mã JavaScript (Runtime Environment) phía máy chủ, được xây dựng trên công cụ V8 JavaScript của Google Chrome [4]. NodeJS sử dụng mô hình I/O không chặn (Non-blocking I/O) và hướng sự kiện (Event-driven), cho phép xử lý hàng ngàn kết nối đồng thời với hiệu suất cao, rất phù hợp cho các ứng dụng mạng xã hội hoặc diễn đàn cần tốc độ phản hồi nhanh.

2.3.2 ExpressJS

ExpressJS là một khung ứng dụng web (Framework) tối giản và linh hoạt dành cho NodeJS. Nó cung cấp một bộ công cụ mạnh mẽ để xây dựng các máy chủ web và API [5]. Trong đồ án, ExpressJS đóng vai trò:

- **Định tuyến (Routing):** Xử lý các yêu cầu HTTP (GET, POST, PUT, DELETE) đến các đường dẫn cụ thể (ví dụ: `/api/posts`, `/api/users`).
- **Middleware:** Xử lý các tác vụ trung gian như xác thực người dùng, kiểm tra dữ liệu đầu vào trước khi đến bước xử lý chính.

2.4. Hệ quản trị Cơ sở dữ liệu (MySQL)

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) mã nguồn mở phổ biến nhất thế giới [6]. MySQL lưu trữ dữ liệu dưới dạng các bảng (Tables) có quan hệ với nhau, đảm bảo tính chặt chẽ và toàn vẹn của dữ liệu.

Lý do chọn MySQL cho đề tài:

- **Cấu trúc rõ ràng:** Phù hợp để quản lý các dữ liệu có cấu trúc phức tạp như thông tin người dùng, bài viết, bình luận, và các mối quan hệ (ví dụ: *Một người dùng có nhiều bài viết*).
- **Hỗ trợ giao dịch (Transaction):** Đảm bảo an toàn dữ liệu trong các thao tác quan trọng.
- **Cộng đồng lớn:** Dễ dàng tìm kiếm tài liệu hỗ trợ và khắc phục lỗi.

Trong ứng dụng NodeJS, thư viện mysql2 được sử dụng làm driver kết nối, hỗ trợ Promise giúp việc viết mã truy vấn cơ sở dữ liệu trở nên gọn gàng và dễ xử lý lỗi hơn.

2.5. Các kỹ thuật và thư viện hỗ trợ khác

2.5.1 JSON Web Token (JWT) - Cơ chế xác thực

JWT là một tiêu chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền tải thông tin an toàn giữa các bên dưới dạng đối tượng JSON [7]. Trong hệ thống, JWT được sử dụng để xác thực người dùng (Authentication):

- Khi người dùng đăng nhập thành công, Server tạo ra một chuỗi Token (chứa thông tin người dùng) và gửi về Client.
- Client lưu trữ Token này (thường trong LocalStorage) và gửi kèm trong Header của các yêu cầu tiếp theo.
- Server kiểm tra Token để xác định danh tính người dùng mà không cần truy vấn lại mật khẩu trong cơ sở dữ liệu.

2.5.2 Bcryptjs - Mã hóa mật khẩu

Để bảo mật thông tin người dùng, mật khẩu không bao giờ được lưu dưới dạng văn bản thuần (plain text). Thư viện **Bcryptjs** được sử dụng để băm (hash) mật khẩu trước khi lưu vào cơ sở dữ liệu. Bcrypt sử dụng thuật toán băm chậm và thêm một chuỗi ngẫu nhiên (salt) để chống lại các cuộc tấn công dò mật khẩu (Brute-force attack).

2.5.3 Multer - Quản lý tải lên tập tin

Diễn đàn cho phép người dùng chia sẻ tài liệu và thay đổi ảnh đại diện. **Multer** là một middleware của NodeJS chuyên dùng để xử lý dữ liệu multipart/form-data,

giúp hệ thống tiếp nhận, đổi tên và lưu trữ các file tải lên từ người dùng vào thư mục trên máy chủ một cách an toàn [8].

2.5.4 Nodemailer - Gửi Email

Thư viện **Nodemailer** được tích hợp để thực hiện các chức năng liên quan đến email, cụ thể là gửi mã OTP (One Time Password) xác thực khi người dùng thực hiện chức năng "Quên mật khẩu", giúp tăng tính bảo mật và tin cậy cho hệ thống.

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả vấn đề

Trong môi trường học thuật, đặc biệt là lĩnh vực Công nghệ thông tin, nhu cầu trao đổi kiến thức, chia sẻ mã nguồn (source code) và tài liệu tham khảo là vô cùng lớn. Tuy nhiên, hiện nay sinh viên thường gặp một số khó khăn khi sử dụng các nền tảng mạng xã hội phổ thông để trao đổi học tập:

- **Thông tin rời rạc:** Các bài viết hỏi đáp, chia sẻ tài liệu thường bị trôi nhanh trên Newsfeed, khó tìm kiếm lại khi cần thiết.
- **Hạn chế về định dạng:** Các nền tảng chat hoặc mạng xã hội thông thường không hỗ trợ tốt việc hiển thị mã nguồn (Syntax Highlighting), khiến việc đọc và debug code trở nên khó khăn.
- **Thiếu tính chuyên biệt:** Không có sự phân loại rõ ràng giữa các chủ đề (ví dụ: Lập trình Web, Mobile, Thuật toán...), dẫn đến việc tiếp cận thông tin không đúng trọng tâm.

Từ những vấn đề trên, việc xây dựng một hệ thống **Diễn đàn Tin học** chuyên biệt là giải pháp cần thiết. Hệ thống này sẽ đóng vai trò là kho tri thức tập trung, nơi sinh viên có thể đăng tải câu hỏi với định dạng code chuẩn, chia sẻ tài liệu học tập có tổ chức và thảo luận chuyên sâu về các chủ đề công nghệ.

3.2. Xác định yêu cầu người dùng

Để đảm bảo hệ thống giải quyết được các vấn đề nêu trên và mang lại trải nghiệm tốt nhất, các yêu cầu người dùng được xác định như sau :

Yêu cầu về giao diện và trải nghiệm:

- Giao diện phải trực quan, hiện đại và thân thiện với người dùng, tối ưu hóa cho việc đọc tài liệu và mã nguồn.
- Hỗ trợ tương thích trên nhiều thiết bị (máy tính, điện thoại) để sinh viên có thể truy cập mọi lúc mọi nơi.

Yêu cầu về chức năng cốt lõi:

- **Soạn thảo nội dung:** Người dùng cần một công cụ soạn thảo mạnh mẽ, hỗ trợ định dạng **Markdown** để viết bài hướng dẫn kỹ thuật và chèn các đoạn code dễ đọc.

- **Chia sẻ tài liệu:** Hệ thống phải cho phép tải lên (upload) và tải xuống (download) các định dạng tài liệu phổ biến như PDF, DOCX, hình ảnh để phục vụ việc chia sẻ giáo trình, slide bài giảng.
- **Tìm kiếm và Tra cứu:** Cung cấp công cụ tìm kiếm hiệu quả theo từ khóa, thẻ (tags) hoặc danh mục để người dùng nhanh chóng tìm thấy nội dung mình cần.
- **Tương tác cộng đồng:** Người dùng có nhu cầu bình luận, trả lời câu hỏi, đánh giá bài viết và theo dõi các thành viên uy tín để cập nhật kiến thức mới.

3.3. Phân loại người dùng

Người dùng trong Diễn đàn Tin học được phân thành 3 nhóm đối tượng chính với các quyền hạn cụ thể như sau:

3.3.1. Khách (Guest)

Đây là nhóm người dùng chưa đăng nhập hoặc chưa có tài khoản. Họ có quyền truy cập hạn chế vào hệ thống để tham khảo thông tin.

Quyền hạn:

- Xem danh sách bài viết, xem chi tiết bài viết và các bình luận.
- Xem danh sách và chi tiết các tài liệu được chia sẻ.
- Tìm kiếm bài viết và tài liệu theo từ khóa hoặc bộ lọc.
- Xem hồ sơ công khai của các thành viên khác.
- Thực hiện Đăng ký, Đăng nhập hoặc khôi phục mật khẩu (Quên mật khẩu).

3.3.2. Thành viên (User)

Đây là nhóm người dùng đã đăng ký tài khoản và đăng nhập thành công. Nhóm này kế thừa toàn bộ quyền hạn của Khách và có thêm các quyền tương tác sâu với hệ thống.

Quyền hạn:

- Quản lý bài viết: Tạo bài viết mới (có hỗ trợ gắn thẻ tags, chọn danh mục), chỉnh sửa hoặc xóa bài viết của chính mình.
- Quản lý tài liệu: Tải lên tài liệu học tập, quản lý và xóa tài liệu do mình đóng góp.

- Tương tác: Viết bình luận, trả lời bình luận, theo dõi (Follow) hoặc bỏ theo dõi người dùng khác.
- Cá nhân: Lưu bài viết (Bookmark) để xem sau, cập nhật thông tin cá nhân, thay đổi ảnh đại diện và đổi mật khẩu.
- Báo cáo: Gửi báo cáo vi phạm đối với các bài viết, tài liệu hoặc người dùng xấu đến Ban quản trị.

3.3.3. Quản trị viên (Admin)

Đây là nhóm người dùng có quyền hạn cao nhất, chịu trách nhiệm vận hành và kiểm soát nội dung của diễn đàn. Nhóm này kế thừa các quyền của Thành viên.

Quyền hạn:

- Thống kê: Xem Dashboard tổng quan về số lượng người dùng, bài viết, tài liệu và lượt truy cập.
- Quản lý người dùng: Xem danh sách thành viên, thực hiện khóa (block) hoặc mở khóa tài khoản vi phạm.
- Quản lý nội dung: Ghim các bài viết quan trọng, xóa bài viết hoặc tài liệu vi phạm quy định của diễn đàn.
- Quản lý danh mục: Tạo mới, chỉnh sửa hoặc xóa các danh mục bài viết/tài liệu.
- Xử lý báo cáo: Xem xét và xử lý các báo cáo vi phạm từ thành viên (gỡ bỏ nội dung hoặc đánh dấu báo cáo sai).

3.4. Phân tích thiết kế hệ thống

3.4.1 Đặc tả yêu cầu hệ thống

3.4.1.1 Yêu cầu chức năng

Dựa trên nhu cầu thực tế và cấu trúc dữ liệu đã thiết kế, hệ thống cần đáp ứng các nhóm chức năng sau:

3.4.1.1.1 Nhóm chức năng Tài khoản & Cá nhân hóa

- **Đăng ký/Đăng nhập:** Người dùng có thể tạo tài khoản mới và đăng nhập vào hệ thống. Mật khẩu phải được mã hóa để đảm bảo an toàn.
- **Quản lý hồ sơ:** Cập nhật thông tin cá nhân, thay đổi ảnh đại diện (Avatar).

- **Cài đặt giao diện (Theme Settings):** Người dùng có thể tùy chỉnh giao diện (màu sắc, font chữ, hình nền) và hệ thống sẽ lưu lại cấu hình này.
- **Theo dõi (Follow):** Cho phép theo dõi người dùng khác hoặc theo dõi các danh mục bài viết quan tâm để nhận thông báo mới nhất.

3.4.1.1.2 Nhóm chức năng Quản lý Nội dung (Bài viết & Tài liệu)

Bài viết (Posts):

- Soạn thảo và đăng bài viết mới với nội dung hỗ trợ định dạng phong phú (Markdown).
- Gắn thẻ (Tags) và chọn danh mục (Categories) để phân loại bài viết.
- Ghim bài viết (Pin) lên đầu trang (dành cho Admin).

Tài liệu (Documents):

- Tải lên các tệp tài liệu (PDF, DOCX, PPTX...).
- Hệ thống ghi nhận số lượt tải xuống (Downloads) để thống kê mức độ phổ biến.

Lưu trữ (Bookmarks): Người dùng có thể lưu lại các bài viết hay để xem sau.

3.4.1.1.3 Nhóm chức năng Tương tác & Cộng đồng

- **Bình luận (Comments):** Hỗ trợ bình luận đa cấp (trả lời một bình luận cụ thể) thông qua cơ chế `parent_id`.
- **Thông báo (Notifications):** Hệ thống tự động gửi thông báo khi có người bình luận, theo dõi, hoặc khi bị cảnh báo vi phạm.
- **Báo cáo vi phạm (Reports):** Người dùng có thể báo cáo bài viết, tài liệu hoặc người dùng khác nếu thấy nội dung xấu (Spam, quấy rối, sai sự thật...).

3.4.1.1.4 Nhóm chức năng Quản trị (Admin)

- **Thống kê (Statistics):** Xem tổng quan số lượng người dùng, bài viết, bình luận và tài liệu theo thời gian.
- **Xử lý báo cáo:** Xem xét các báo cáo từ người dùng và đưa ra quyết định (Xóa nội dung, bỏ qua).
- **Quản lý người dùng & Cấm (Bans):** Admin có quyền cấm người dùng thực hiện các hành động cụ thể (Cấm đăng bài, cấm bình luận, cấm tải tài liệu) hoặc khóa toàn bộ tài khoản.

- **Hệ thống cảnh báo (Warnings):** Theo dõi số lần vi phạm của người dùng để áp dụng hình phạt tự động.

3.4.1.2 Yêu cầu phi chức năng

Hiệu năng: Hệ thống phải xử lý nhanh các truy vấn tìm kiếm và tải trang, đảm bảo trải nghiệm mượt mà ngay cả khi cơ sở dữ liệu lớn.

Tính toàn vẹn dữ liệu: Dữ liệu phải đảm bảo tính nhất quán thông qua các ràng buộc khóa ngoại (Foreign Keys) chặt chẽ (ví dụ: khi xóa User, các bài viết liên quan cũng được xử lý tương ứng).

Bảo mật:

- Mật khẩu phải được băm bằng bcrypt trước khi lưu.
- Cấu hình thông tin nhạy cảm của người dùng (như theme_settings) được lưu trữ an toàn dưới dạng JSON.

Khả năng mở rộng: Cấu trúc CSDL được thiết kế để dễ dàng thêm các tính năng mới (như thêm loại thông báo mới, thêm quyền hạn mới) mà không phá vỡ hệ thống hiện tại.

3.4.2 Kiến trúc hệ thống

Hệ thống được xây dựng dựa trên **Kiến trúc Ba Tầng (Three-Tier Architecture)**, đảm bảo sự tách biệt rõ ràng giữa giao diện, xử lý nghiệp vụ và dữ liệu.

3.4.2.1 Lớp giao diện người dùng (Presentation Layer):

Công nghệ: ReactJS.

Vai trò: Đây là tầng tương tác trực tiếp với người dùng. Nó chịu trách nhiệm hiển thị dữ liệu (Bài viết, Tài liệu, Dashboard Admin) và thu thập hành động của người dùng (Click, Nhập liệu).

Đặc điểm: Sử dụng các thành phần (Components) tái sử dụng, quản lý trạng thái giao diện và gửi các yêu cầu HTTP (thông qua Axios) đến lớp xử lý nghiệp vụ.

3.4.2.2 Lớp xử lý nghiệp vụ (Business Logic Layer):

Công nghệ: NodeJS và ExpressJS.

Vai trò: Đóng vai trò trung gian, tiếp nhận yêu cầu từ Frontend, xử lý logic và trả về kết quả.

Các tác vụ chính:

- Xác thực & Phân quyền: Kiểm tra xem người dùng có quyền đăng bài, xóa bài hay truy cập trang Admin không.
- Xử lý dữ liệu: Validate dữ liệu đầu vào (ví dụ: kiểm tra định dạng email, độ dài mật khẩu).
- Quản lý tệp tin: Xử lý việc upload ảnh avatar và tài liệu vào thư mục uploads trên server.
- Logic nghiệp vụ: Xử lý các logic phức tạp như đếm số lượt xem, gửi thông báo tự động, xử lý logic cấm người dùng.

3.4.2.3 Lớp dữ liệu (Data Layer):

Công nghệ: MySQL.

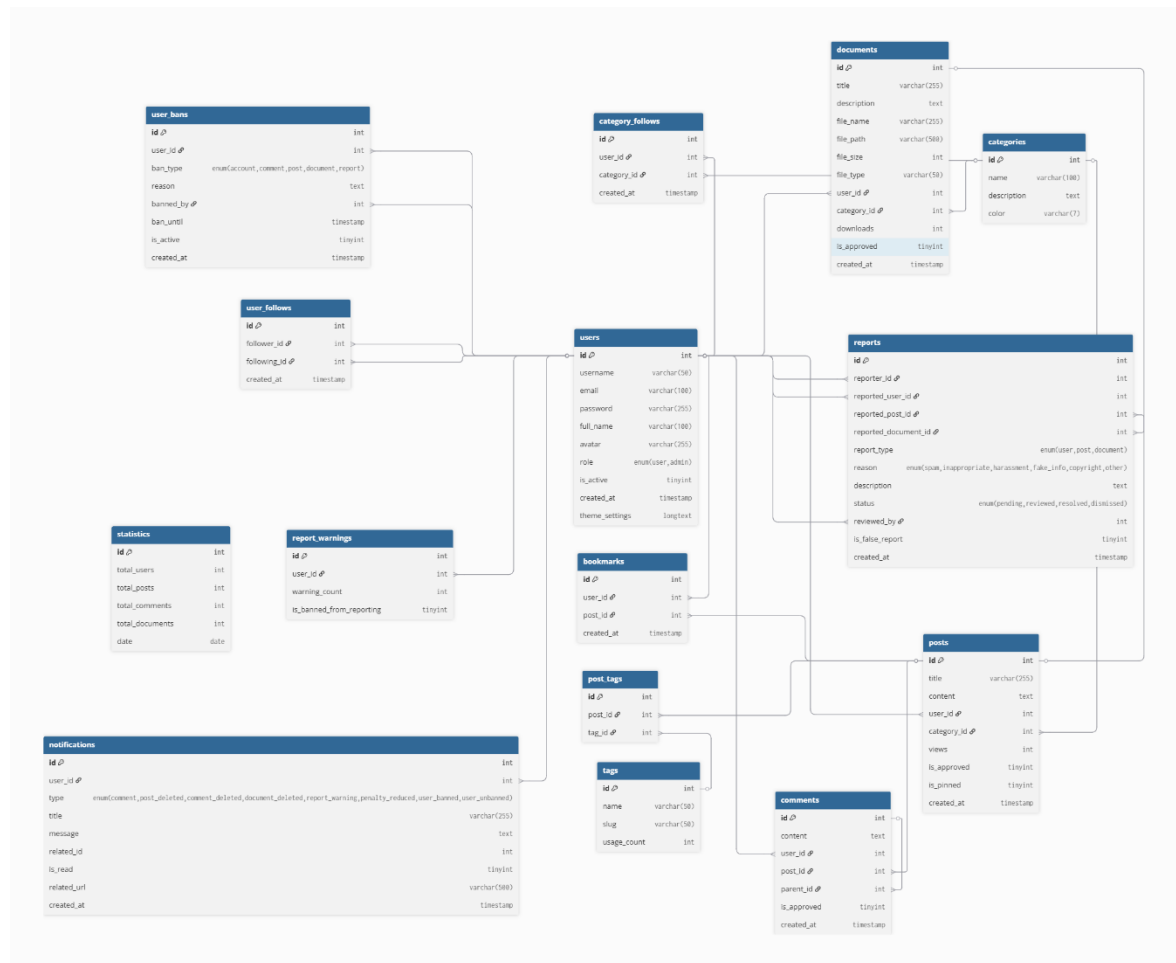
Vai trò: Lưu trữ bền vững toàn bộ dữ liệu của hệ thống.

Cấu trúc thực tế (Dựa trên file SQL):

- Hệ thống sử dụng các bảng quan hệ chặt chẽ như users, posts, categories.
- Sử dụng các kiểu dữ liệu hiện đại như JSON cho bảng users (cột theme_settings) để lưu cấu hình linh hoạt.
- Sử dụng ENUM cho các trạng thái (ví dụ: role trong bảng users, type trong bảng notifications) để đảm bảo dữ liệu chuẩn xác.
- Sử dụng TIMESTAMP để ghi nhận chính xác thời gian tạo và cập nhật dữ liệu.

3.4.3 Thiết kế dữ liệu

3.4.3.1 Mô hình ERD



Hình 3.1. Mô hình ERD

3.4.3.2 Danh sách các thực thể và mối kết hợp

Hệ thống bao gồm 15 thực thể chính được mô tả trong bảng dưới đây:

Bảng 3.1. Bảng thực thể

STT	Tên thực thể	Diễn giải
1	users	Lưu trữ thông tin tài khoản, thông tin cá nhân và cấu hình của người dùng.
2	categories	Quản lý các chuyên mục (chủ đề) của bài viết và tài liệu.
3	posts	Lưu trữ nội dung các bài viết, thảo luận do người dùng đăng tải.
4	documents	Lưu trữ thông tin metadata của các tệp tài liệu học tập được chia sẻ.

STT	Tên thực thể	Diễn giải
5	comments	Lưu trữ các bình luận, phản hồi dưới bài viết.
6	notifications	Hệ thống thông báo gửi đến người dùng về các tương tác hoặc cảnh báo.
7	reports	Quản lý các báo cáo vi phạm nội dung hoặc người dùng xấu.
8	user_bans	Lưu trữ lịch sử và trạng thái cấm (khóa) tài khoản người dùng.
9	bookmarks	Lưu trữ danh sách bài viết mà người dùng đã đánh dấu (lưu lại).
10	tags	Quản lý các thẻ (từ khóa) để phân loại bài viết chi tiết hơn.
11	post_tags	Bảng trung gian liên kết giữa Bài viết và Thẻ (quan hệ nhiều-nhiều).
12	user_follows	Lưu trữ thông tin theo dõi giữa các người dùng với nhau.
13	category_follows	Lưu trữ thông tin người dùng theo dõi các chuyên mục.
14	report_warnings	Quản lý số lần cảnh báo vi phạm của từng người dùng.
15	statistics	Lưu trữ số liệu thống kê hoạt động của hệ thống theo ngày.

3.4.3.3 Chi tiết các thực thể

Dưới đây là mô tả chi tiết các thuộc tính và mối quan hệ của 15 bảng trong cơ sở dữ liệu:

Bảng users (Người dùng):

Lưu trữ thông tin tài khoản và thiết lập cá nhân. Đây là bảng trung tâm kết nối với hầu hết các bảng khác.

Bảng 3.2. Bảng người dùng

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã định danh duy nhất của người dùng.
2	username	VARCHAR(50)	Tên đăng nhập (duy nhất).
3	email	VARCHAR(100)	Email đăng nhập (duy nhất).
4	password	VARCHAR(255)	Mật khẩu đã mã hóa.
5	full_name	VARCHAR(100)	Họ và tên hiển thị.
6	avatar	VARCHAR(255)	Đường dẫn ảnh đại diện.
7	role	ENUM	Vai trò: 'user' hoặc 'admin'.
8	theme_settings	LONGTEXT	Cấu hình giao diện (JSON).

Mối kết hợp:

- Quan hệ 1-N với posts, documents, comments, notifications: Một người dùng có thể tạo nhiều bài viết, tài liệu, bình luận và nhận nhiều thông báo.
- Quan hệ 1-N với reports: Một người dùng có thể gửi nhiều báo cáo hoặc bị báo cáo nhiều lần.

Bảng categories (Danh mục):

Quản lý các chủ đề của diễn đàn.

Bảng 3.3. Bảng danh mục

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã danh mục.
2	name	VARCHAR(100)	Tên danh mục (Ví dụ: Lập trình Web).
3	description	TEXT	Mô tả chi tiết danh mục.
4	color	VARCHAR(7)	Mã màu hiển thị (HEX).

Mối kết hợp:

- Quan hệ 1-N với posts và documents: Một danh mục chứa nhiều bài viết và tài liệu.

- Quan hệ 1-N với category_follows: Một danh mục có thể được nhiều người dùng theo dõi.

Bảng posts (Bài viết):

Lưu trữ nội dung thảo luận chính.

Bảng 3.4. Bảng bài viết

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã bài viết.
2	title	VARCHAR(255)	Tiêu đề bài viết.
3	content	TEXT	Nội dung bài viết (Markdown).
4	user_id	INT (FK)	Liên kết với bảng users.
5	category_id	INT (FK)	Liên kết với bảng categories.
6	is_pinned	TINYINT	Trạng thái ghim (1: Có, 0: Không).

Mối kết hợp:

- Quan hệ N-1 với users và categories.
- Quan hệ 1-N với comments: Một bài viết có nhiều bình luận.
- Quan hệ 1-N với bookmarks: Một bài viết được nhiều người lưu.
- Quan hệ N-N với tags (thông qua bảng trung gian post_tags).

Bảng documents (Tài liệu):

Lưu trữ thông tin tệp tin chia sẻ.

Bảng 3.5. Bảng tài liệu

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã tài liệu.
2	title	VARCHAR(255)	Tên hiển thị.
3	file_path	VARCHAR(500)	Đường dẫn file trên server.
4	file_type	VARCHAR(50)	Loại file (PDF, DOCX...).
5	user_id	INT (FK)	Người tải lên (Link users).
6	category_id	INT (FK)	Thuộc danh mục (Link categories).

Mối kết hợp:

- Quan hệ N-1 với users và categories.
- Quan hệ 1-N với reports: Tài liệu có thể bị báo cáo vi phạm.

Bảng comments (Bình luận):

Lưu trữ thảo luận dưới bài viết.

Bảng 3.6. Bảng bình luận

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã bình luận.
2	content	TEXT	Nội dung bình luận.
3	user_id	INT (FK)	Người bình luận (Link users).
4	post_id	INT (FK)	Bình luận tại bài viết nào (Link posts).
5	parent_id	INT (FK)	Mã bình luận cha (nếu là trả lời).

Mối kết hợp:

- Quan hệ N-1 với users và posts.
- Quan hệ 1-N với chính nó (comments): Để thực hiện chức năng trả lời bình luận (Reply/Nested comments).

Bảng notifications (Thông báo):

Hệ thống thông báo cho người dùng.

Bảng 3.7. Bảng thông báo

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã thông báo.
2	user_id	INT (FK)	Người nhận (Link users).
3	type	ENUM	Loại: comment, warning, ban...
4	message	TEXT	Nội dung thông báo.
5	is_read	TINYINT	Trạng thái đã xem.

Mỗi kết hợp: Quan hệ N-1 với users.

Bảng reports (Báo cáo vi phạm):

Quản lý các khiếu nại về nội dung xấu.

Bảng 3.8. Bảng báo cáo vi phạm

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã báo cáo.
2	reporter_id	INT (FK)	Người gửi báo cáo (Link users).
3	report_type	ENUM	Đối tượng: 'user', 'post', 'document'.
4	reported_post_id	INT (FK)	ID bài viết bị báo cáo (nếu có).
5	reported_doc_id	INT (FK)	ID tài liệu bị báo cáo (nếu có).
6	status	ENUM	Trạng thái xử lý.

Mỗi kết hợp: Liên kết đa chiều đến users (người báo cáo/bị báo cáo), posts, documents.

Bảng user_bans (Lệnh cấm):

Quản lý lịch sử và trạng thái cấm tài khoản.

Bảng 3.9. Bảng lệnh cấm

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã lệnh cấm.
2	user_id	INT (FK)	Người bị cấm (Link users).
3	ban_type	ENUM	Loại cấm (account, comment...).
4	ban_until	TIMESTAMP	Thời hạn cấm.
5	banned_by	INT (FK)	Admin thực hiện (Link users).

Mối kết hợp: Quan hệ N-1 với users (đối tượng bị cấm và người thực hiện).

Bảng bookmarks (Lưu bài viết):

Lưu trữ bài viết yêu thích của người dùng.

Bảng 3.10. Bảng lưu bài viết

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã bookmark.
2	user_id	INT (FK)	Người lưu (Link users).
3	post_id	INT (FK)	Bài viết được lưu (Link posts).

Mối kết hợp: Bảng trung gian thể hiện quan hệ N-N giữa users và posts.

Bảng tags (Thẻ):

Từ khóa phân loại bài viết.

Bảng 3.11. Bảng thẻ

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã thẻ.
2	name	VARCHAR(50)	Tên thẻ (Ví dụ: Java, React).
3	slug	VARCHAR(50)	Đường dẫn tĩnh.

Mối kết hợp: Quan hệ N-N với posts thông qua bảng post_tags.

Bảng post_tags (Liên kết Bài viết - Thẻ):

Bảng trung gian để gắn thẻ.

Bảng 3.12. Bảng liên kết bài viết với thẻ

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã liên kết.
2	post_id	INT (FK)	Mã bài viết.
3	tag_id	INT (FK)	Mã thẻ.

Mối kết hợp: Kết nối bảng posts và tags.

Bảng user_follows (Theo dõi người dùng)

Mạng lưới xã hội giữa các thành viên.

Bảng 3.13. Bảng theo dõi người dùng

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã theo dõi.
2	follower_id	INT (FK)	Người theo dõi (Link users).
3	following_id	INT (FK)	Người được theo dõi (Link users).

Mối kết hợp: Quan hệ N-N tự tham chiếu trên bảng users.

Bảng category_follows (Theo dõi danh mục)

Cho phép người dùng nhận thông báo từ chuyên mục yêu thích.

Bảng 3.14. Bảng Theo dõi danh mục

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã theo dõi.
2	user_id	INT (FK)	Người theo dõi (Link users).
3	category_id	INT (FK)	Danh mục (Link categories).

Mối kết hợp: Bảng trung gian giữa users và categories.

Bảng report_warnings (Cảnh báo)

Theo dõi số lần vi phạm của người dùng.

Bảng 3.15. Bảng cảnh báo

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã cảnh báo.
2	user_id	INT (FK)	Người bị cảnh báo (Link users).
3	warning_count	INT	Số lần vi phạm tích lũy.

Mỗi kết hợp: Quan hệ 1-1 hoặc N-1 với users.

Bảng statistics (Thống kê)

Lưu trữ dữ liệu tổng hợp hệ thống.

Bảng 3.16. Bảng thống kê

STT	Thuộc tính	Kiểu dữ liệu	Diễn giải
1	id	INT (PK)	Mã thống kê.
2	total_users	INT	Tổng số thành viên.
3	total_posts	INT	Tổng số bài viết.
4	date	DATE	Ngày thống kê.

Mỗi kết hợp: Bảng độc lập, dữ liệu được tổng hợp từ các bảng khác để phục vụ Dashboard Admin.

3.4.4 Thiết kế xử lý

3.4.4.1 Sơ đồ phân cấp chức năng

Dựa trên yêu cầu và phân tích người dùng, các chức năng của hệ thống "Diễn đàn Tin học" được phân cấp thành các nhóm quản lý chính như sau:

Nhóm chức năng hệ thống & Tài khoản:

- Đăng ký, Đăng nhập, Đăng xuất.
- Quên mật khẩu (Xác thực OTP qua Email).
- Cập nhật thông tin cá nhân, Đổi mật khẩu, Thay đổi Avatar.
- Xem thông báo hệ thống.

Nhóm chức năng Quản lý Bài viết (Diễn đàn):

- Xem danh sách, Tìm kiếm, Lọc bài viết.
- Tạo bài viết mới (Soạn thảo Markdown, Thêm thẻ Tags).
- Chỉnh sửa, Xóa bài viết (của chính mình).

- Lưu bài viết (Bookmark).
- Bình luận, Trả lời bình luận (Reply).

Nhóm chức năng Quản lý Tài liệu:

- Xem danh sách, Tìm kiếm tài liệu.
- Tải lên (Upload) tài liệu mới.
- Tải xuống (Download) tài liệu.
- Xóa tài liệu đã đăng.

Nhóm chức năng Quản trị (Dành cho Admin):

- Xem Dashboard thống kê (Người dùng, Bài viết, Tài liệu).
- Quản lý người dùng (Khóa, Mở khóa, Cấm).
- Quản lý nội dung (Ghim bài, Xóa bài/Tài liệu vi phạm).
- Quản lý danh mục (Thêm, Sửa, Xóa).
- Xử lý báo cáo vi phạm.

3.4.4.2 Đặc tả quy trình xử lý các chức năng chính

Dưới đây là mô tả chi tiết luồng xử lý dữ liệu (Data Flow) cho các chức năng cốt lõi của hệ thống, áp dụng kiến trúc 3 tầng đã thiết kế.

Quy trình Đăng bài viết mới (Create Post): Chức năng cho phép thành viên chia sẻ kiến thức lên diễn đàn.

- Bước 1 (Giao diện): Người dùng nhập tiêu đề, nội dung (Markdown), chọn danh mục và gắn thẻ (Tags) tại form đăng bài.
- Bước 2 (Xử lý - Frontend): ReactJS kiểm tra tính hợp lệ (tiêu đề không rỗng, nội dung đủ độ dài). Gửi yêu cầu POST kèm Token xác thực đến API.
- Bước 3 (Xử lý - Backend):
 - + Middleware xác thực JWT kiểm tra người dùng đã đăng nhập và không bị cấm đăng bài (user_bans).
 - + Controller nhận dữ liệu, làm sạch nội dung (Sanitize) để tránh lỗi XSS.
- Bước 4 (Dữ liệu):
 - + Lưu bài viết vào bảng posts.
 - + Lưu các thẻ vào bảng tags (nếu thẻ mới) và tạo liên kết trong bảng post_tags.

- Bước 5 (Kết quả): Trả về thông báo thành công và chuyển hướng người dùng về trang chi tiết bài viết vừa đăng.

Quy trình Tải lên tài liệu (Upload Document): Chức năng cho phép chia sẻ tệp tin học tập.

- Bước 1 (Giao diện): Người dùng chọn tệp tin từ máy tính, nhập tiêu đề mô tả.
- Bước 2 (Xử lý - Backend):
 - + Middleware Multer tiếp nhận file, kiểm tra định dạng (chỉ cho phép PDF, DOCX, PPTX) và kích thước (giới hạn 50MB).
 - + Đổi tên file (để tránh trùng lặp) và lưu vào thư mục uploads/documents trên Server.
- Bước 3 (Dữ liệu): Lưu đường dẫn file (file_path), tên gốc, kích thước và người đăng vào bảng documents trong CSDL MySQL.
- Bước 4 (Kết quả): Hiển thị thông báo thành công và cập nhật danh sách tài liệu.

Quy trình Bình luận và Trả lời (Comment & Reply): Chức năng tương tác thảo luận dưới bài viết.

- Bước 1 (Giao diện): Người dùng nhập nội dung bình luận vào ô nhập liệu dưới bài viết hoặc nhấn nút "Trả lời" tại một bình luận có sẵn.
- Bước 2 (Xử lý - Backend):
 - + Kiểm tra quyền bình luận của người dùng (check bảng user_bans).
 - + Nếu là trả lời bình luận, hệ thống ghi nhận parent_id là ID của bình luận cha.
- Bước 3 (Dữ liệu): Thêm dòng mới vào bảng comments.
- Bước 4 (Thông báo):
 - + Hệ thống tự động tạo bản ghi trong bảng notifications để thông báo cho tác giả bài viết hoặc người được trả lời.
 - + Cập nhật giao diện danh sách bình luận ngay lập tức.

Quy trình Xử lý Báo cáo vi phạm (Handle Report - Admin): Chức năng giúp Admin duy trì môi trường diễn đàn sạch.

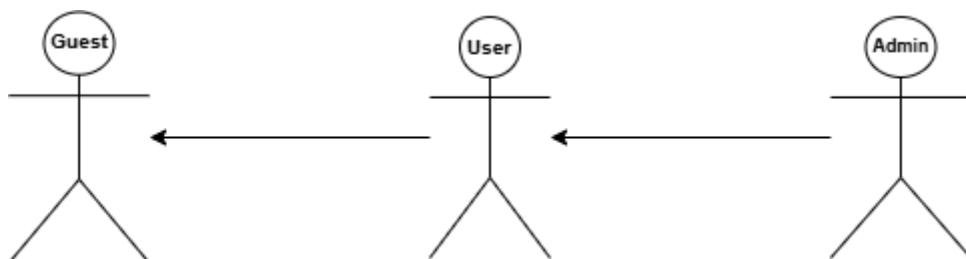
- Bước 1 (Giao diện Admin): Admin xem danh sách các báo cáo trong bảng reports có trạng thái 'pending'.

- Bước 2 (Xử lý): Admin xem chi tiết nội dung bị báo cáo (Bài viết spam, Tài liệu sai, v.v.).
- Bước 3 (Quyết định):
 - + *Nếu vi phạm*: Admin chọn "Xóa nội dung" -> Hệ thống xóa bài viết/tài liệu trong CSDL (posts/documents) -> Cập nhật trạng thái báo cáo thành 'resolved'.
 - + *Nếu không vi phạm*: Admin chọn "Bỏ qua" -> Cập nhật trạng thái báo cáo thành 'dismissed'.
- Bước 4 (Cảnh báo): Nếu người dùng vi phạm nghiêm trọng, Admin có thể tạo lệnh cấm trong bảng user_bans, hệ thống sẽ chặn quyền truy cập của người dùng đó.

3.4.5 Thiết kế giao diện

3.4.5.1 Sơ đồ Use Case

3.4.5.1.1 Quan hệ kế thừa giữa các actor



Hình 3.2. Quan hệ kế thừa giữa các tác nhân

Sơ đồ mô tả mối quan hệ kế thừa giữa ba tác nhân trong hệ thống: Guest, User và Admin.

Trong mô hình này:

- User kế thừa Guest: nghĩa là một người dùng đã đăng nhập (User) có đầy đủ các quyền của khách (Guest), đồng thời được mở rộng thêm nhiều chức năng khác.
- Admin kế thừa User: người quản trị viên (Admin) sở hữu toàn bộ quyền của User và thêm các chức năng quản trị nâng cao.

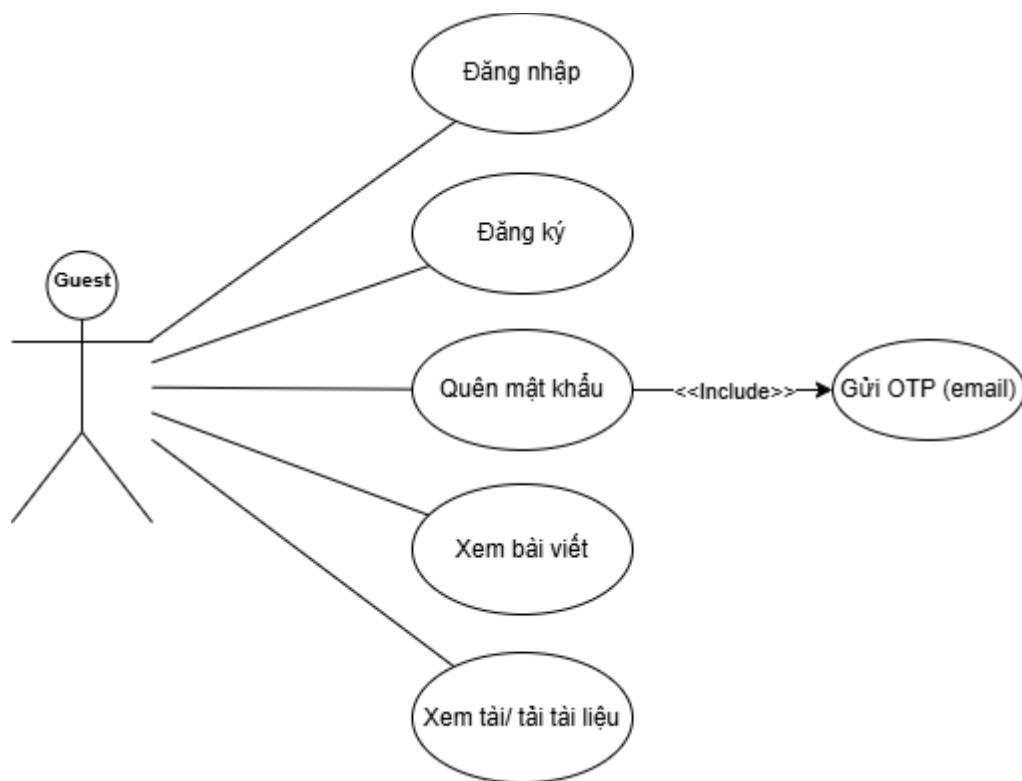
Sử dụng mối quan hệ kế thừa giúp giảm trùng lặp và mô tả chính xác cấp độ quyền hạn của từng vai trò.

Các mối quan hệ khác được sử dụng:

- Association (Kết hợp): Là mối quan hệ cơ bản nhất thể hiện sự tương tác trực tiếp giữa tác nhân và chức năng (VD: Tác nhân Guest tương tác với chức năng Đăng nhập).
- Include (Bao gồm): Thể hiện chức năng phụ bắt buộc phải thực hiện để hoàn thành chức năng chính (VD: Chức năng Quên mật khẩu luôn phải gọi chức năng Gửi OTP).
- Extend (Mở rộng): Thể hiện chức năng phụ là tùy chọn, chỉ được thực hiện trong điều kiện nhất định (VD: Chức năng Thêm tags là một mở rộng không bắt buộc khi Tạo bài viết).

3.4.5.1.2 Use Case của người dùng khách (Guest)

Người dùng khách là nhóm người dùng chưa có tài khoản hoặc chưa đăng nhập trong quá trình sử dụng diễn đàn.



Hình 3.3. Use Case của người dùng khách

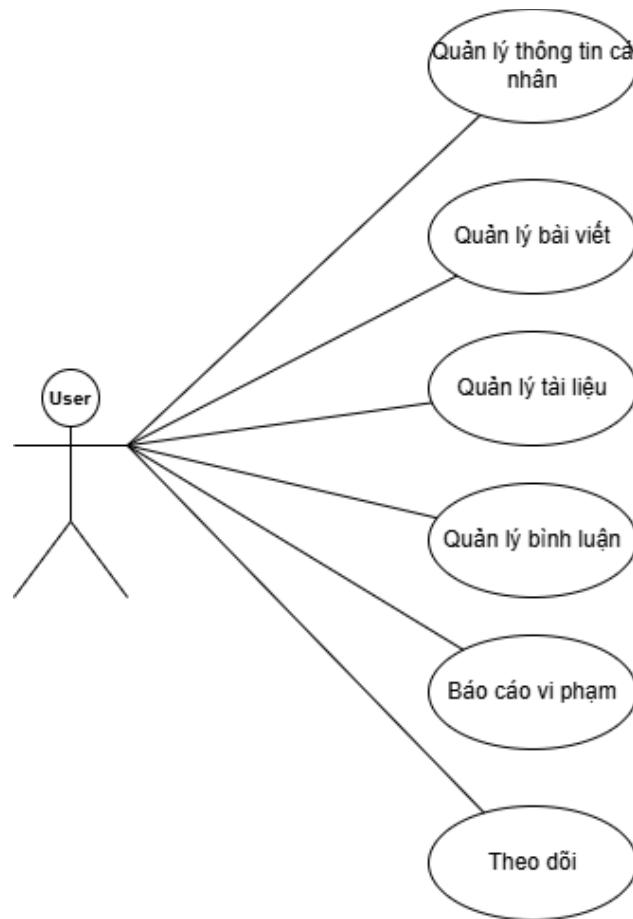
Mô tả chi tiết:

Bảng 3.17. Mô tả Use Case của người dùng khách

Tên Use Case	Mối quan hệ	Mô tả chi tiết
Đăng ký	Association	Người dùng nhập các thông tin cá nhân cần thiết (Tên đăng nhập, Email, Mật khẩu) để tạo tài khoản mới trên hệ thống.
Đăng nhập	Association	Người dùng nhập Tên đăng nhập và Mật khẩu để xác thực danh tính và truy cập vào các chức năng dành cho thành viên.
Quên mật khẩu	Association Include << Gửi OTP >>	Người dùng yêu cầu cấp lại mật khẩu mới thông qua email khi không nhớ mật khẩu đăng nhập.
Gửi OTP (Email)	Include <<Quên mật khẩu>>	Hệ thống thực hiện gửi mã xác thực (OTP) về email của người dùng để xác minh trước khi cho phép đặt lại mật khẩu.
Tra cứu bài viết	Association	Người dùng xem danh sách bài viết, xem chi tiết nội dung, tìm kiếm bài viết theo từ khóa hoặc lọc theo Chuyên mục/Thẻ.
Tra cứu tài liệu	Association	Người dùng xem danh sách các tài liệu trong thư viện, xem thông tin chi tiết và tải xuống (Download) tài liệu về máy.

3.4.5.1.3 Use Case của người dùng đã đăng nhập (User)

Là nhóm người dùng đã đăng nhập vào diễn đàn, nhóm này có tất cả chức năng của người dùng khách và một số chức năng khác.



Hình 3.4. Use Case của người dùng đã đăng nhập

Mô tả chi tiết:

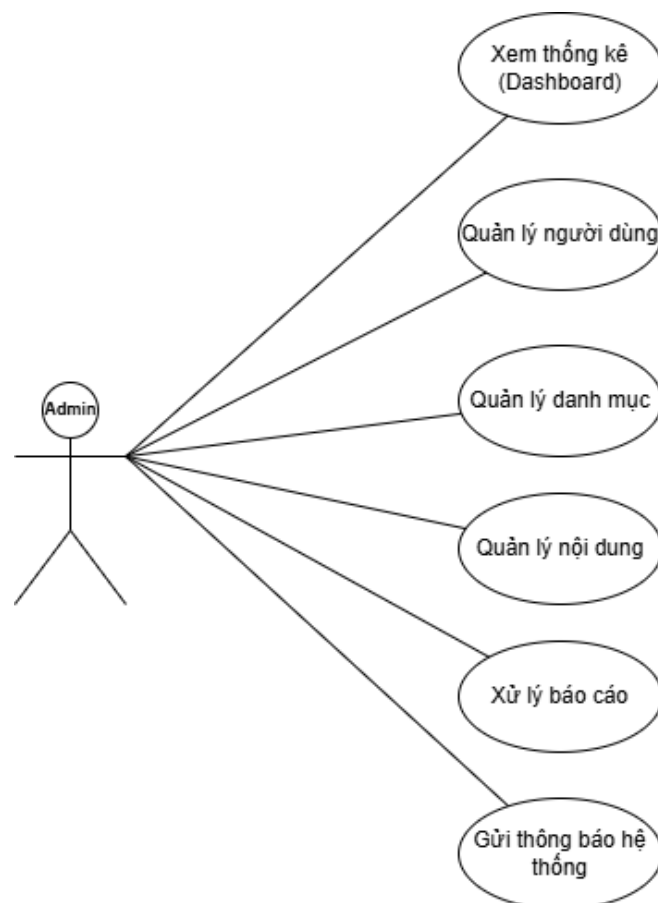
Bảng 3.18. Mô tả Use Case Use Case của người dùng đã đăng nhập

Tên Use Case	Mối quan hệ	Mô tả chi tiết
Quản lý Thông tin cá nhân	Association	Người dùng thực hiện cập nhật hồ sơ cá nhân (Họ tên, Giới thiệu), thay đổi ảnh đại diện (Avatar) và đổi mật khẩu đăng nhập.
Quản lý Bài viết	Association	Người dùng thực hiện các quyền đối với bài viết cá nhân: Soạn thảo và Đăng bài viết mới ; Chỉnh sửa nội dung bài viết đã đăng; Xóa bài viết của chính mình.
Quản lý Tài liệu	Association	Người dùng thực hiện Tải lên (Upload) các tệp tin tài liệu (PDF, DOCX...) để chia sẻ và Xóa các tài liệu mà mình đã tải lên.

Tên Use Case	Mối quan hệ	Mô tả chi tiết
Quản lý Bình luận	Association	Người dùng thực hiện tương tác thảo luận: Gửi bình luận mới vào bài viết; Trả lời (Reply) bình luận của người khác; Sửa hoặc Xóa bình luận của chính mình.
Báo cáo vi phạm	Association	Người dùng gửi phiếu báo cáo đến Ban quản trị khi phát hiện Bài viết, Tài liệu hoặc Thành viên khác vi phạm quy định.
Theo dõi (Follow)	Association	Người dùng nhấn theo dõi các thành viên khác hoặc theo dõi các bài viết để nhận thông báo khi có cập nhật mới.

3.4.5.1.4 Use Case của quản trị viên (Admin)

Đây là những người điều hành quản lý diễn đàn, có tất cả chức năng của người dùng cùng các chức năng quản lý.



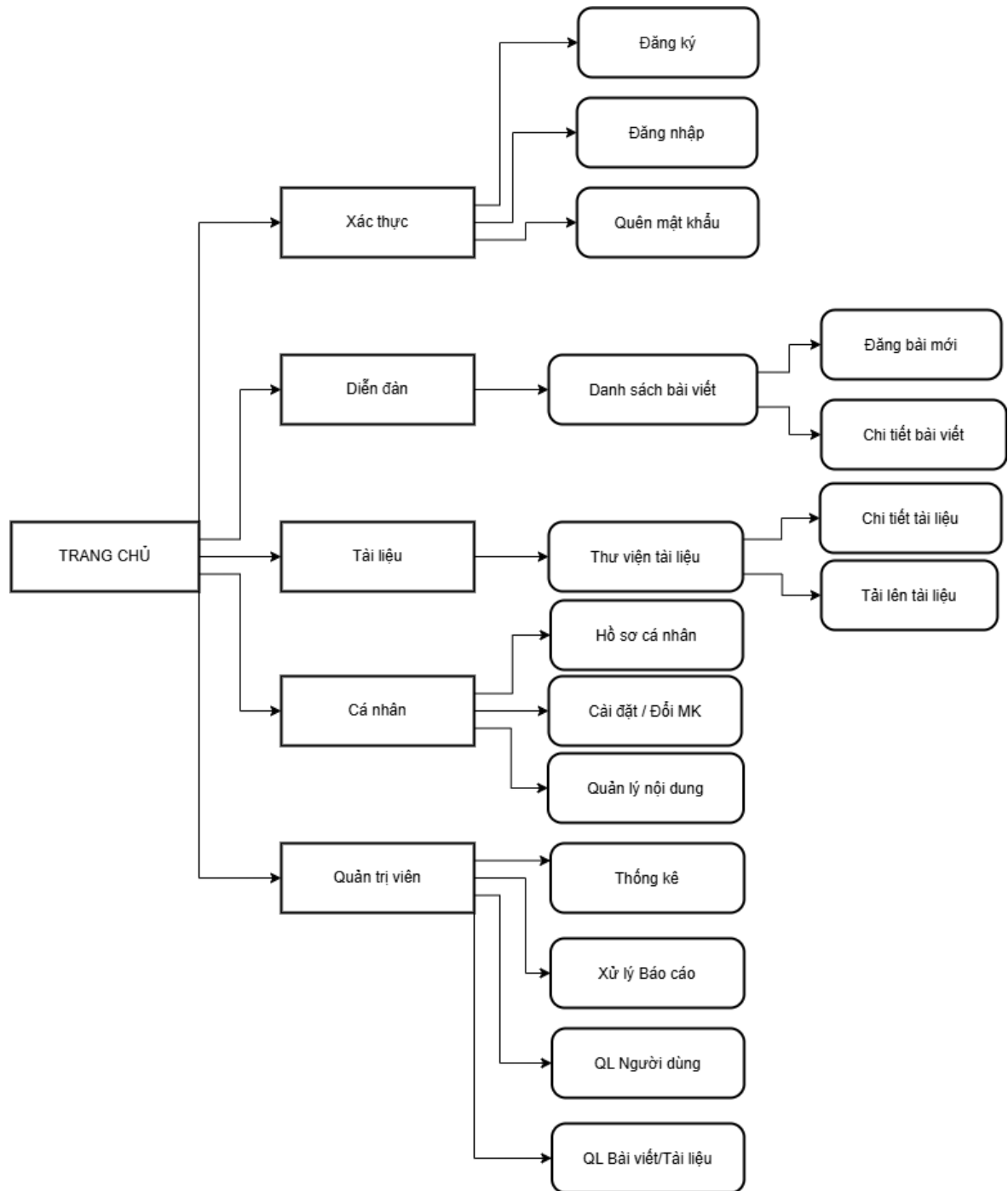
Hình 3.5. Use Case của quản trị viên

Mô tả chi tiết:

Bảng 3.19. Mô tả Use Case Use Case của quản trị viên

Tên Use Case	Mối quan hệ	Mô tả chi tiết
Xem Thống kê	Association	Xem bảng điều khiển tổng quan (Dashboard) hiển thị số liệu về: Số lượng người dùng, Bài viết, Tài liệu và Báo cáo mới.
Quản lý Người dùng	Association	Admin xem danh sách thành viên, tìm kiếm và thực hiện hành động Khóa tài khoản (Ban) đối với user vi phạm hoặc Mở khóa (Unban) .
Quản lý Danh mục	Association	Admin thực hiện Thêm mới , Chỉnh sửa tên hoặc Xóa các Chuyên mục (Categories) bài viết và tài liệu trên hệ thống.
Quản lý Nội dung	Association	Admin kiểm duyệt nội dung trên diễn đàn, bao gồm: Ghim bài viết quan trọng, Xóa bài viết hoặc Xóa tài liệu vi phạm quy định.
Xử lý Báo cáo	Association	Admin xem danh sách các báo cáo từ người dùng gửi lên và đưa ra quyết định xử lý (Xóa nội dung bị báo cáo hoặc Bác bỏ báo cáo).

3.4.5.2 Sơ đồ website



Hình 3.6. Sơ đồ website

Mô tả sơ đồ tổ chức Website

Hệ thống được tổ chức theo cấu trúc phân cấp (Hierarchical), lấy Trang chủ (Homepage) làm trung tâm điều hướng chính. Từ trang chủ, người dùng có thể truy cập đến các nhóm chức năng thông qua thanh thực đơn (Navigation Bar) như sau:

- Nhóm Xác thực:** Bao gồm các trang Đăng ký, Đăng nhập và Quên mật khẩu, là cổng vào để người dùng truy cập các quyền hạn nâng cao.

2. **Nhóm Diễn đàn (Forum):** Dẫn đến trang Danh sách bài viết. Từ đây, người dùng có thể điều hướng tiếp đến trang Chi tiết bài viết hoặc trang Soạn thảo bài viết mới.
3. **Nhóm Tài liệu (Documents):** Dẫn đến Thư viện tài liệu chung, nơi hiển thị danh sách và trang chi tiết để tải xuống tài liệu.
4. **Nhóm Cá nhân:** Dành cho thành viên đã đăng nhập, bao gồm Trang hồ sơ cá nhân (Profile) và Trang cài đặt tài khoản.
5. **Khu vực Quản trị (Admin Panel):** Là một nhánh riêng biệt được bảo mật, chỉ dành cho Quản trị viên truy cập để vào Dashboard và các trang quản lý (Người dùng, Nội dung, Báo cáo).

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Dữ liệu thử nghiệm

Để kiểm thử các chức năng của hệ thống, một bộ dữ liệu mẫu đã được nạp vào cơ sở dữ liệu, bao gồm danh mục, bài viết, tài liệu và các tài khoản người dùng với các vai trò khác nhau. Dưới đây là mô tả chi tiết các dữ liệu thử nghiệm:

4.1.1 Dữ liệu người dùng (Users)

Để thực hiện kiểm thử chức năng phân quyền và bảo mật, hệ thống được khởi tạo với danh sách các tài khoản mẫu đại diện cho các vai trò khác nhau. Hình 4.1 dưới đây liệt kê các tài khoản được sử dụng để mô phỏng hành vi của Quản trị viên (Admin) và Thành viên (User) trong quá trình nghiệm thu.

id	username	email	password	full_name	avatar	role	is_active	created_at	updated_at	reset_otp	reset_otp_expiry	theme_settings
1	admin	admin@diendan.com	\$2a\$10\$92lXUNpky0rOQ5yMlYe4oKcEa3Ro6lCjogja2...	Quản trị viên	NULL	admin	1	2025-11-19 18:40:31	2025-12-07 19:38:54	722252	2025-11-19 23:31:37	["backgroundColor":"#f9f9f9","backgroundImage":"","
2	tranlamphuduc	katozamata@gmail.com	\$2a\$10\$92lXUNpky0rOQ5yMlYe4oKcEa3Ro6lCjogja2...	Trần Lâm Phú Đức	uploads/avatars/1785110553785-619534274.png	user	1	2025-11-19 18:40:31	2025-12-07 19:43:53	332305	2025-11-20 00:00:57	["backgroundColor":"#f9f9f9","backgroundImage":"","
3	nguyenthanhduy	tranlamphuduc3teucan2@gmail.com	\$2a\$10\$92lXUNpky0rOQ5yMlYe4oKcEa3Ro6lCjogja2...	Nguyễn Thanh Duy	NULL	user	1	2025-11-19 18:40:31	2025-12-07 19:48:17	498869	2025-11-19 23:50:19	["backgroundColor":"#f9f9f9","backgroundImage":"","
4	nguyendinhankhoa	lequangc@gmail.com	\$2a\$10\$92lXUNpky0rOQ5yMlYe4oKcEa3Ro6lCjogja2...	Nguyễn Đình Tuấn Khoa	NULL	user	1	2025-11-19 18:40:31	2025-11-19 18:40:31	NULL	NULL	NULL

Hình 4.1. Dữ liệu bảng Users

4.1.2 Dữ liệu danh mục (Categories)

Để thực hiện kiểm thử chức năng quản lý bài viết và tổ chức nội dung diễn đàn, hệ thống được khởi tạo với danh sách các danh mục mẫu đại diện cho các lĩnh vực công nghệ khác nhau. Hình 4.2 dưới đây liệt kê các chủ đề được sử dụng để phân loại và gắn nhãn cho các thảo luận trong quá trình nghiệm thu.

id	name	description	color	created_at
1	Lập trình Web	Thảo luận về HTML, CSS, JavaScript, React, Node.js	#e74c3c	2025-11-19 18:40:31
2	Cơ sở dữ liệu	MySQL, MongoDB, PostgreSQL và các hệ quản trị CSDL	#3498db	2025-11-19 18:40:31
3	Thuật toán	Cấu trúc dữ liệu và giải thuật	#9b59b6	2025-11-19 18:40:31
4	Lập trình di động	Android, iOS, React Native, Flutter	#2ecc71	2025-11-19 18:40:31
5	Trí tuệ nhân tạo	Machine Learning, Deep Learning, AI	#f39c12	2025-11-19 18:40:31
6	Bảo mật	An toàn thông tin, mã hóa, penetration testing	#e67e22	2025-11-19 18:40:31
7	DevOps	Docker, Kubernetes, CI/CD, Cloud Computing	#34495e	2025-11-19 18:40:31
8	Game Development	Unity, Unreal Engine, phát triển game	#8e44ad	2025-11-19 18:40:31

Hình 4.2. Dữ liệu bảng Categories

4.1.3 Dữ liệu bài viết (Posts)

Để đánh giá khả năng hiển thị nội dung, đặc biệt là việc hỗ trợ định dạng văn bản Markdown và tô màu mã nguồn (Syntax Highlighting), một số bài viết mẫu đã

được đăng tải với nội dung và tác giả khác nhau. Bảng dưới đây mô tả chi tiết các bài viết này:

id	title	content	user_id	category_id	views	is_approved	created_at	updated_at	is_pinned	pinned_at
1	Hướng dẫn học React.js cho người mới bắt đầu	# Giới thiệu về React.js React.js là một thư viện... # Stack và Queue	2	1	172	1	2025-11-19 18:40:31	2025-12-07 19:41:33	0	NULL
2	Cấu trúc dữ liệu Stack và Queue	## Stack (Ngăn xếp) Stack là cấu... # Tối ưu hóa MySQL	3	3	95	1	2025-11-19 18:40:31	2025-12-07 18:36:33	0	NULL
3	Tối ưu hóa truy vấn MySQL	## 1. Sử dụng Index Index giúp... # Bảo mật Web Application	4	2	235	1	2025-11-19 18:40:31	2025-12-07 18:21:50	0	NULL
4	Bảo mật ứng dụng web cơ bản	## 1. SQL Injection Lu...	2	6	180	1	2025-11-19 18:40:31	2025-12-07 18:33:09	0	NULL
6	asd	**asd** **dasdasd** asd "asd" # asd# > SAD https://www.yo...	1	6	6	1	2025-11-19 22:31:18	2025-12-07 18:38:24	0	NULL
10	asd		2	6	8	1	2025-11-19 22:46:59	2025-12-07 18:36:12	0	NULL

Hình 4.3. Dữ liệu bảng Posts

4.1.4 Dữ liệu tài liệu (Documents)

Chức năng chia sẻ tài liệu được kiểm thử thông qua việc tải lên các tệp tin với nhiều định dạng phổ biến (PDF, DOCX, PPTX) và kích thước khác nhau. Dữ liệu này giúp xác minh tính ổn định của chức năng Upload (Tải lên) và Download (Tải xuống).

id	title	description	file_name	file_path	file_size	file_type	user_id	category_id	downloads	is_approved	created_at
1	Hướng dẫn React.js cơ bản	Tài liệu hướng dẫn React.js từ cơ bản đến nâng cao	react-guide.pdf	documents/react-guide.pdf	2048576	application/pdf	2	1	45	1	2025-11-19 18:40:31
2	Cấu trúc dữ liệu và giải thuật	Bài giảng về cấu trúc dữ liệu và giải thuật	data-structures.pptx	documents/data-structures.pptx	5242880	application/vnd.openxmlformats-officedocument.presentationml.presentation	3	3	78	1	2025-11-19 18:40:31
3	Hướng dẫn MySQL	Tài liệu hướng dẫn sử dụng MySQL	mysql-tutorial.docx	documents/mysql-tutorial.docx	1048576	application/vnd.openxmlformats-officedocument.wordprocessingml.document	4	2	32	1	2025-11-19 18:40:31
4	Bảo mật ứng dụng web	Tài liệu về bảo mật ứng dụng web	web-security.pdf	documents/web-security.pdf	3145728	application/pdf	2	6	67	1	2025-11-19 18:40:31
6	adsad	NULL	SOAP.docx	C:\xampp\htdocs\onibackend\uploads\documents\17642...	81995	application/vnd.openxmlformats-officedocument.wordprocessingml.document	1	1	3	1	2025-11-27 15:00:49
7	Coursera QBS90B2AZYUL	NULL	Coursera QBS90B2AZYUL.pdf	C:\xampp\htdocs\onibackend\uploads\documents\17642...	299421	application/pdf	1	6	3	1	2025-11-27 15:02:05

Hình 4.4. Dữ liệu bảng Documents

4.1.5 Dữ liệu Bình luận (Comments)

Hệ thống cho phép người dùng thảo luận dưới mỗi bài viết. Dữ liệu bình luận được khởi tạo nhằm kiểm tra khả năng hiển thị danh sách bình luận và tính năng trả lời bình luận (Nested Comments - Bình luận đa cấp) thông qua trường parent_id.

id	content	user_id	post_id	parent_id	is_approved	created_at
1	Bài viết rất hữu ích! Cảm ơn bạn đã chia sẻ.	3	1	NULL	1	2025-11-19 18:40:31
2	Mình đã thử làm theo và thành công. Thanks!	4	1	NULL	1	2025-11-19 18:40:31
3	Có thể giải thích thêm về lifecycle methods không?	2	1	NULL	1	2025-11-19 18:40:31
4	Stack và Queue là nền tảng quan trọng của lập trình...	2	2	NULL	1	2025-11-19 18:40:31
5	Code example rất dễ hiểu, upvote!	4	2	NULL	1	2025-11-19 18:40:31
6	Index thực sự giúp tăng performance rất nhiều.	3	3	NULL	1	2025-11-19 18:40:31
7	Bảo mật là vấn đề quan trọng mà nhiều dev bỏ qua.	4	4	NULL	1	2025-11-19 18:40:31

Hình 4.5. Dữ liệu bảng Comments

4.1.6 Dữ liệu Báo cáo vi phạm (Reports)

Để đảm bảo môi trường thảo luận lành mạnh, tính năng báo cáo được kiểm thử với các kịch bản khác nhau (spam, ngôn từ không phù hợp, tài liệu sai lệch). Dữ liệu dưới đây giúp kiểm tra quy trình xử lý của Quản trị viên (Duyệt/Xóa/Bỏ qua).

id	reporter_id	reported_user_id	reported_post_id	reported_document_id	report_type	reason	description	status	admin_note	reviewed_by	reviewed_at	is_false_report	created_at
1	2	3	NULL	NULL	user	spam	Người dùng này liên tục gửi tin nhắn spam	dismissed	NULL	1	2025-12-07 19:21:43	0	2025-11-19 18:40:32
2	3	NULL	1	NULL	post	inappropriate	Nội dung bài viết không phù hợp với diễn đàn	pending	NULL	1	2025-12-07 19:52:17	0	2025-11-19 18:40:32
3	4	2	NULL	NULL	user	harassment	Người dùng này có hành vi quấy rối	resolved	NULL	1	2025-12-07 19:27:17	1	2025-11-19 18:40:32
4	2	NULL	3	NULL	post	fake_info	Bài viết chứa thông tin sai lệch về công nghệ	dismissed	NULL	NULL	NULL	0	2025-11-19 18:40:32
5	3	NULL	NULL	1	document	copyright	Tài liệu này vi phạm bản quyền	pending	NULL	NULL	NULL	0	2025-11-19 18:40:32
6	4	NULL	NULL	2	document	inappropriate	Nội dung tài liệu không phù hợp	reviewed	NULL	NULL	NULL	0	2025-11-19 18:40:32

Hình 4.6. Dữ liệu bảng Reports

4.1.7 Dữ liệu Thẻ (Tags)

Các thẻ (Tags) được định nghĩa trước hoặc do người dùng tạo mới khi đăng bài. Dữ liệu này phục vụ cho việc kiểm thử thuật toán tìm kiếm bài viết theo từ khóa và gom nhóm các nội dung liên quan.

id	name	slug	usage_count	created_at
1	javascript	javascript	0	2025-12-07 16:41:45
2	python	python	0	2025-12-07 16:41:45
3	java	java	0	2025-12-07 16:41:45
4	csharp	csharp	0	2025-12-07 16:41:45
5	php	php	0	2025-12-07 16:41:45
6	nodejs	nodejs	0	2025-12-07 16:41:45
7	reactjs	reactjs	0	2025-12-07 16:41:45
8	vuejs	vuejs	0	2025-12-07 16:41:45
9	angular	angular	0	2025-12-07 16:41:45
10	html	html	0	2025-12-07 16:41:45
11	css	css	0	2025-12-07 16:41:45
12	mysql	mysql	0	2025-12-07 16:41:45
13	mongodb	mongodb	0	2025-12-07 16:41:45
14	postgresql	postgresql	0	2025-12-07 16:41:45
15	docker	docker	0	2025-12-07 16:41:45
16	git	git	0	2025-12-07 16:41:45
17	linux	linux	0	2025-12-07 16:41:45
18	windows	windows	0	2025-12-07 16:41:45
19	android	android	0	2025-12-07 16:41:45
20	ios	ios	0	2025-12-07 16:41:45

Hình 4.7. Dữ liệu bảng Tags

4.1.8 Dữ liệu Cấm người dùng (User Bans)

Tính năng "phạt" người dùng được kiểm thử thông qua bảng user_bans. Dữ liệu này xác nhận hệ thống có ngăn chặn đúng các hành động bị cấm (Ví dụ: Người bị cấm bình luận sẽ không thấy ô nhập liệu, người bị khóa tài khoản sẽ không thể đăng nhập).

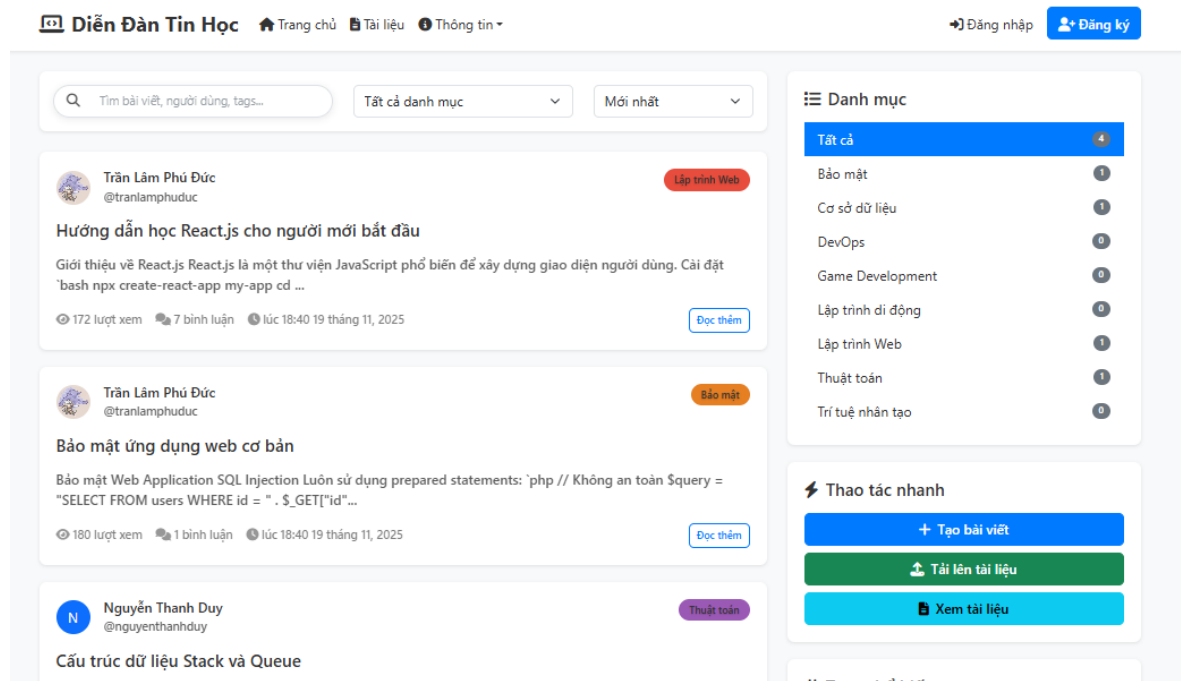
id	user_id	ban_type	reason	banned_by	ban_until	is_active	created_at	updated_at
1	4	comment	NULL	1	2025-12-08 14:17:59	0	2025-12-07 14:17:59	2025-12-07 14:18:04
2	3	post	spam	1	2025-12-08 16:57:01	0	2025-12-07 16:57:01	2025-12-07 16:58:15

Hình 4.8. Dữ liệu bảng User Bans

4.2. Kết quả giao diện thực nghiệm

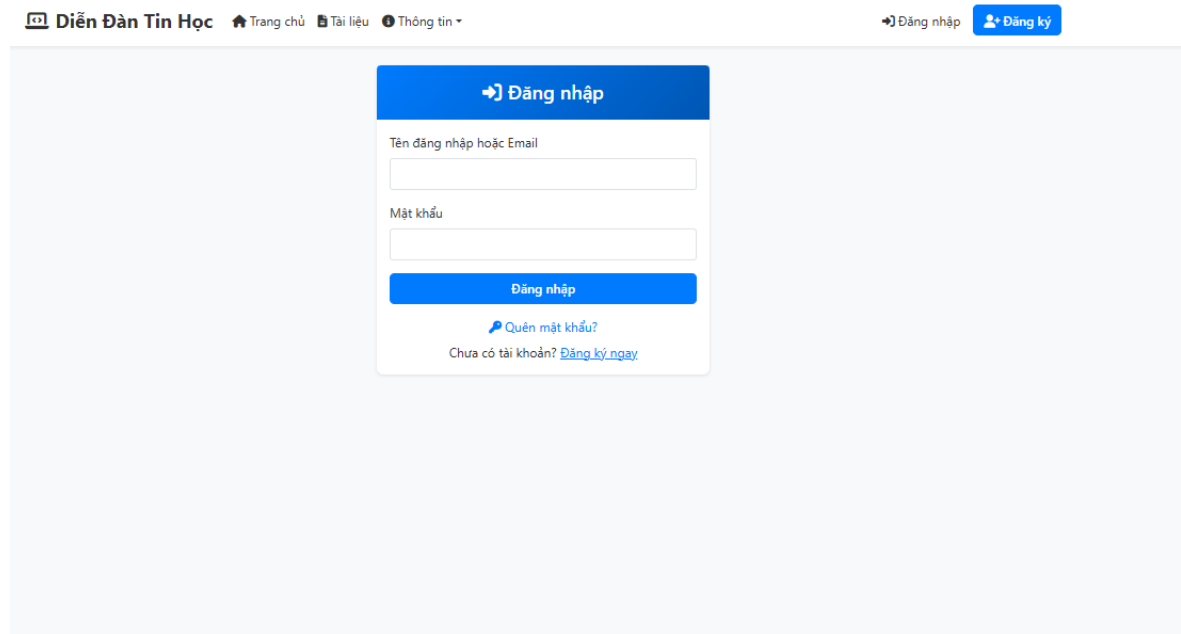
4.2.1 Trang chủ và Xác thực

Giao diện trang chủ hiển thị tổng quan các bài viết mới nhất và nổi bật. Người dùng chưa đăng nhập (Khách) có thể xem nội dung nhưng bị hạn chế các tính năng tương tác.



Hình 4.9. Giao diện Trang chủ (Home)

Chức năng Đăng nhập và Đăng ký cho phép người dùng truy cập vào hệ thống. Giao diện được thiết kế đơn giản, tập trung vào trải nghiệm người dùng.



Hình 4.10. Giao diện Đăng nhập

The screenshot shows the registration form for the 'Diễn Đàn Tin Học' website. The form is titled 'Đăng ký tài khoản' and contains the following fields and instructions:

- Họ và tên ***: Full name field. Instruction: 'Tên hiển thị phải là duy nhất trong hệ thống' (Display name must be unique in the system).
- Tên đăng nhập ***: Username field. Instruction: 'ít nhất 3 ký tự, phải là duy nhất trong hệ thống' (at least 3 characters, must be unique in the system).
- Email ***: Email field.
- Mật khẩu ***: Password field. Instruction: 'ít nhất 6 ký tự' (at least 6 characters).
- Xác nhận mật khẩu ***: Confirm password field.

At the bottom of the form, there is a blue 'Đăng ký' button and a link 'Đã có tài khoản? [Đăng nhập](#)' (Already have an account? [Login](#)).

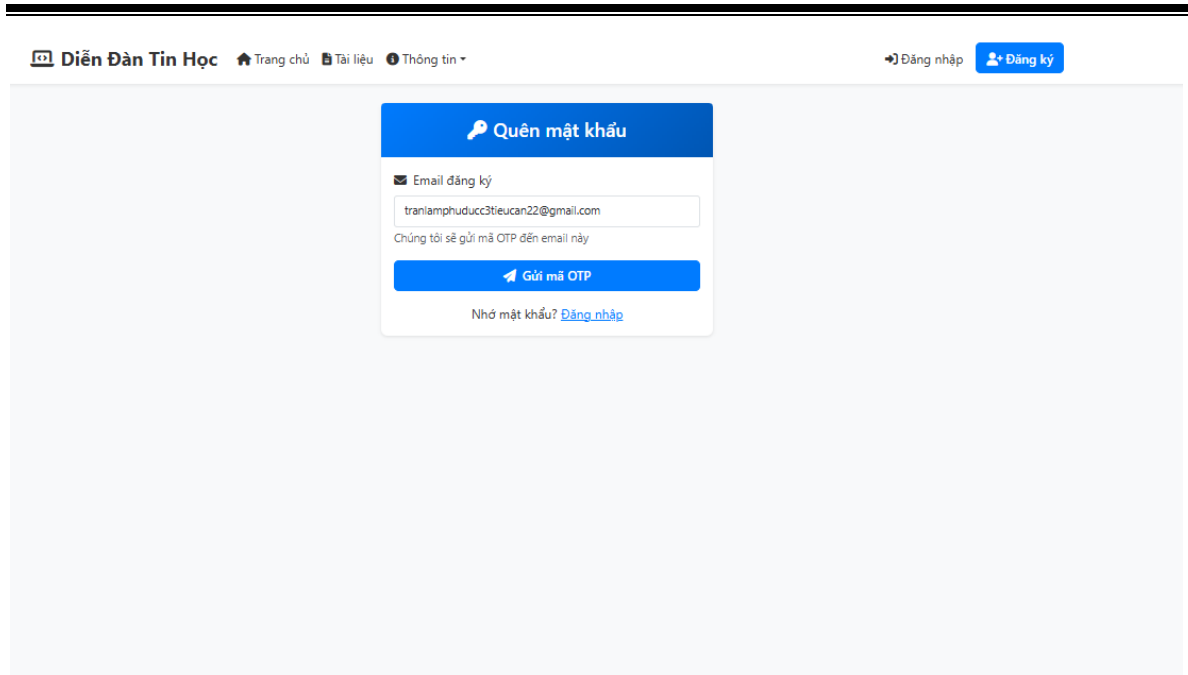
Hình 4.11. Giao diện Đăng ký tài khoản

4.2.2 Chức năng Quên mật khẩu

Chức năng này được xây dựng nhằm hỗ trợ người dùng khôi phục quyền truy cập tài khoản khi không nhớ mật khẩu đăng nhập. Quy trình được thực hiện thông qua cơ chế xác thực 2 bước bằng mã OTP (One-Time Password) gửi qua Email để đảm bảo tính bảo mật.

Giao diện nhập Email xác thực: Tại màn hình đăng nhập, khi người dùng chọn "Quên mật khẩu", hệ thống sẽ chuyển hướng đến giao diện yêu cầu nhập địa chỉ Email.

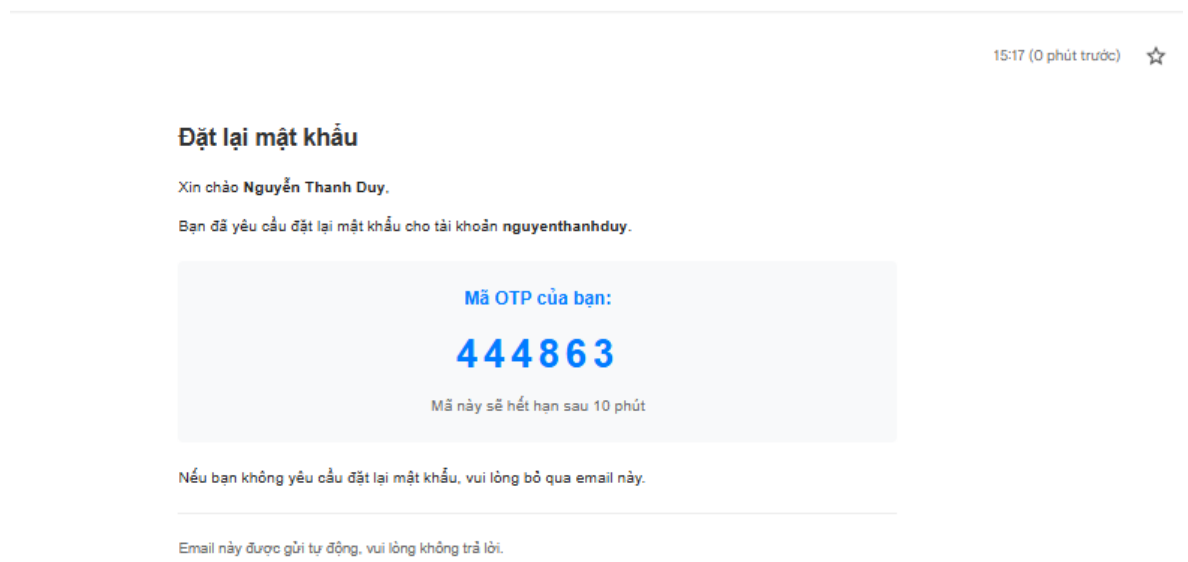
- **Xử lý:** Người dùng nhập email đã đăng ký tài khoản. Hệ thống sẽ kiểm tra định dạng email và xác minh xem email này có tồn tại trong cơ sở dữ liệu hay không.
- **Kết quả:** Nếu email hợp lệ, hệ thống sẽ kích hoạt API gửi mã OTP.



Hình 4.12. Giao diện nhập Email xác thực

Giao diện nhận mã OTP qua Email: Sau khi gửi yêu cầu thành công, hệ thống sử dụng thư viện **Nodemailer** (backend) để gửi một email tự động chứa mã xác thực (OTP) đến hòm thư của người dùng.

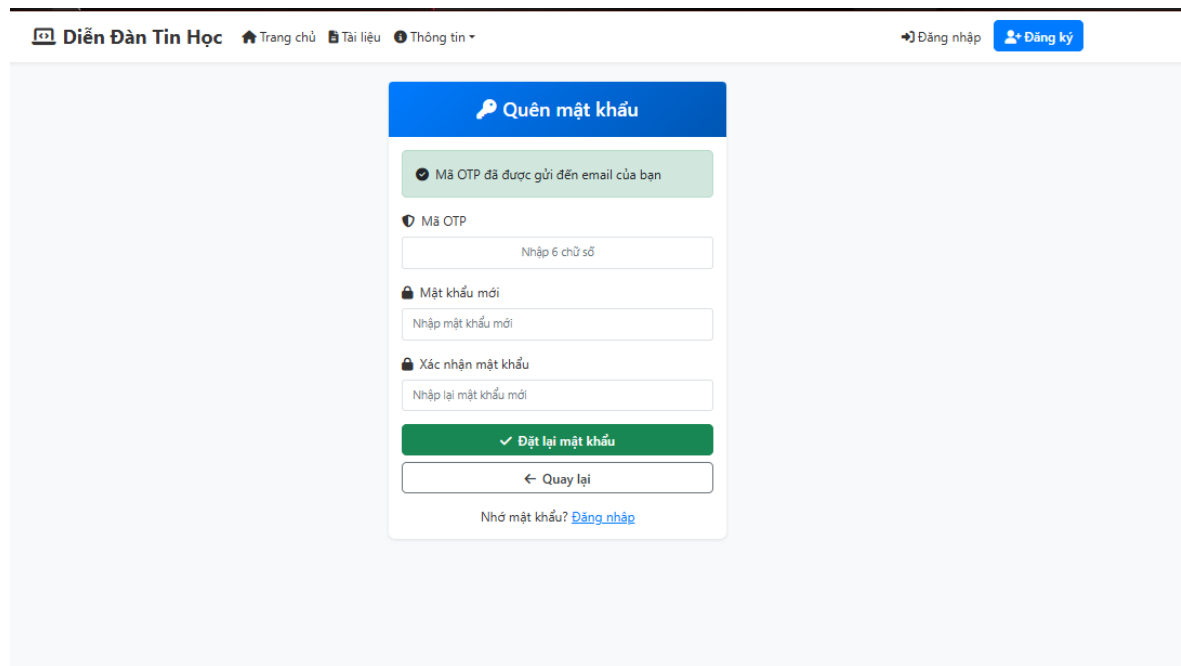
- Nội dung: Email chứa mã số gồm 6 chữ số ngẫu nhiên, có hiệu lực trong thời gian ngắn (ví dụ: 5 phút).
- Mục đích: Xác minh người đang thực hiện thao tác chính là chủ sở hữu của email đó.



Hình 4.13. Giao diện nhận mã OTP qua Email

Giao diện nhập OTP và thiết lập mật khẩu mới: Sau khi nhận được mã, người dùng quay lại website để nhập mã OTP và tiến hành đặt lại mật khẩu mới.

- Xử lý: Người dùng nhập mã OTP, mật khẩu mới và xác nhận mật khẩu mới.
- Kiểm tra: Hệ thống đối chiếu mã OTP người dùng nhập với mã đã lưu trong phiên làm việc (hoặc Database). Nếu khớp, mật khẩu mới sẽ được mã hóa (bằng Bcrypt) và cập nhật vào cơ sở dữ liệu.
- Hoàn tất: Hiển thị thông báo thành công và chuyển hướng người dùng về trang Đăng nhập.



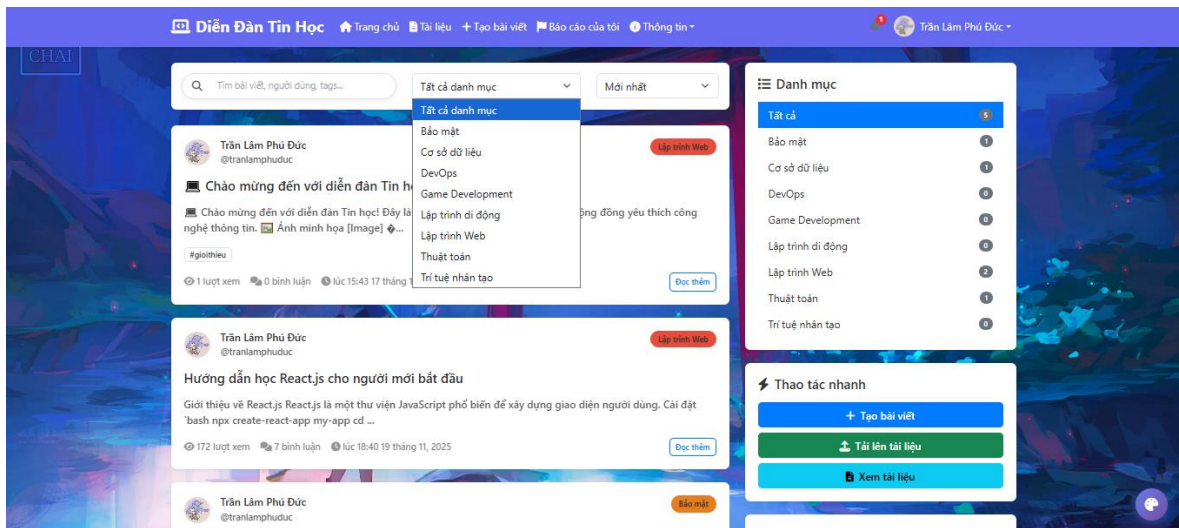
Hình 4.14. Giao diện nhập OTP và thiết lập mật khẩu mới

4.2.3 Chức năng Diễn đàn (Forum)

Đây là phân hệ cốt lõi của hệ thống, nơi diễn ra các hoạt động chia sẻ kiến thức và tương tác giữa các thành viên. Chức năng này bao gồm quy trình từ việc tìm kiếm thông tin, soạn thảo nội dung đến việc thảo luận đa chiều.

Giao diện Danh sách bài viết và Bộ lọc: Khi truy cập vào khu vực diễn đàn, người dùng được cung cấp một cái nhìn tổng quan về các chủ đề đang được thảo luận.

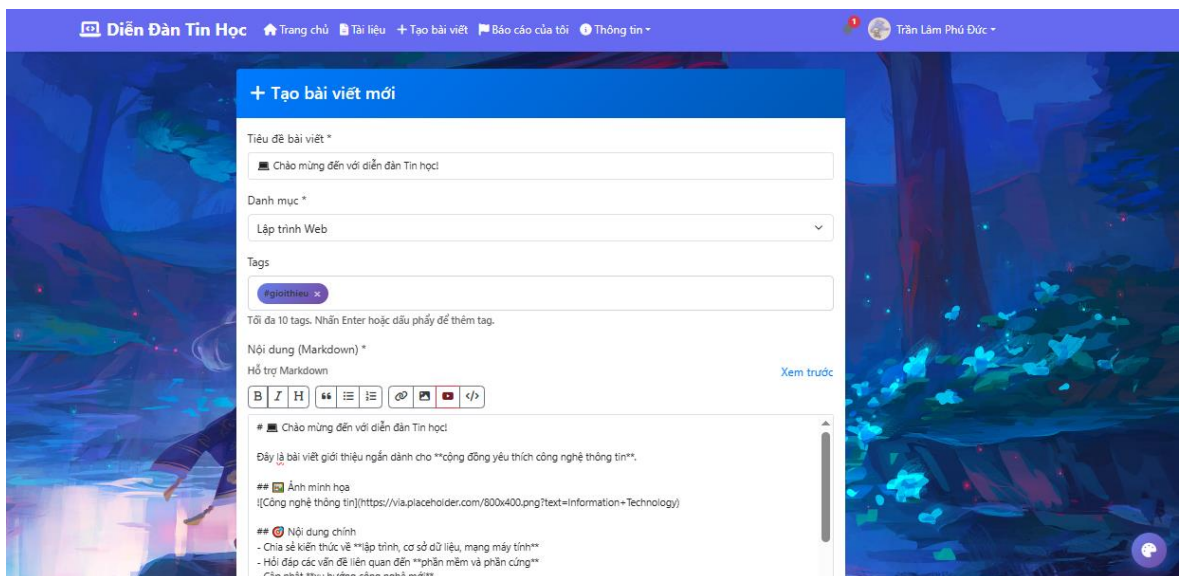
- Hiển thị: Danh sách bài viết được phân trang (Pagination) để tối ưu tốc độ tải trang. Mỗi thẻ bài viết hiển thị tóm tắt thông tin gồm: Tiêu đề, Tác giả, Ngày đăng, Chuyên mục, Tags và số lượt xem.
- Chức năng: Hệ thống hỗ trợ thanh công cụ tìm kiếm theo từ khóa và bộ lọc theo Chuyên mục (Categories) hoặc Thẻ (Tags), giúp người dùng nhanh chóng tiếp cận nội dung quan tâm.



Hình 4.15. Giao diện Danh sách bài viết và Bộ lọc

Giao diện Soạn thảo bài viết mới (Markdown Editor): Để đảm bảo chất lượng nội dung chuyên ngành CNTT, hệ thống tích hợp trình soạn thảo văn bản hỗ trợ định dạng Markdown.

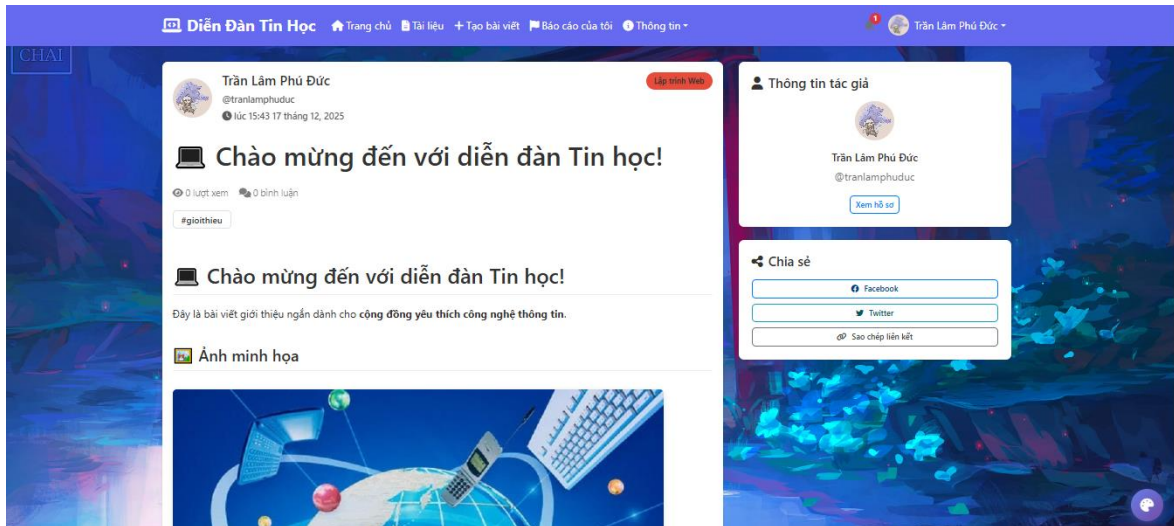
- Xử lý: Người dùng nhập tiêu đề, chọn chuyên mục và soạn thảo nội dung. Trình soạn thảo cho phép chèn các đoạn mã nguồn (Code Block), định dạng in đậm/nghe, tạo danh sách và trích dẫn.
- Tính năng: Chế độ "Xem trước" (Preview) giúp tác giả kiểm tra hiển thị thực tế của bài viết trước khi chính thức đăng tải lên hệ thống. Các thẻ (Tags) mới cũng có thể được tạo trực tiếp tại đây.



Hình 4.16. Giao diện Soạn thảo bài viết mới (Markdown Editor):

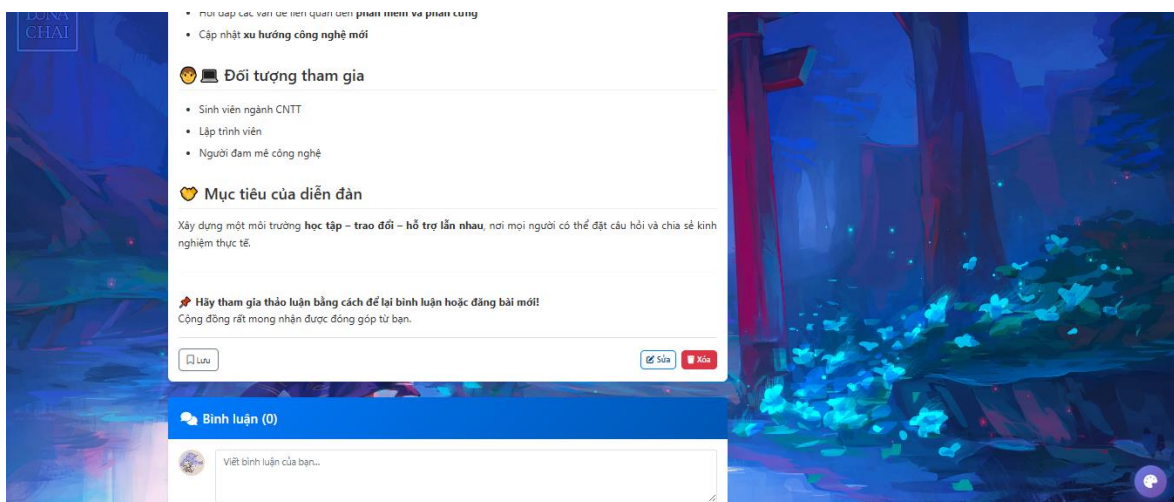
Giao diện Chi tiết bài viết và Bình luận đa cấp: Đây là nơi hiển thị toàn bộ nội dung kiến thức và khu vực tương tác cộng đồng.

- **Hiển thị bài viết:** Nội dung Markdown được chuyển đổi thành HTML an toàn. Các đoạn mã nguồn (Code snippet) được hệ thống tự động phát hiện ngôn ngữ và tô màu cú pháp (Syntax Highlighting) để dễ đọc.



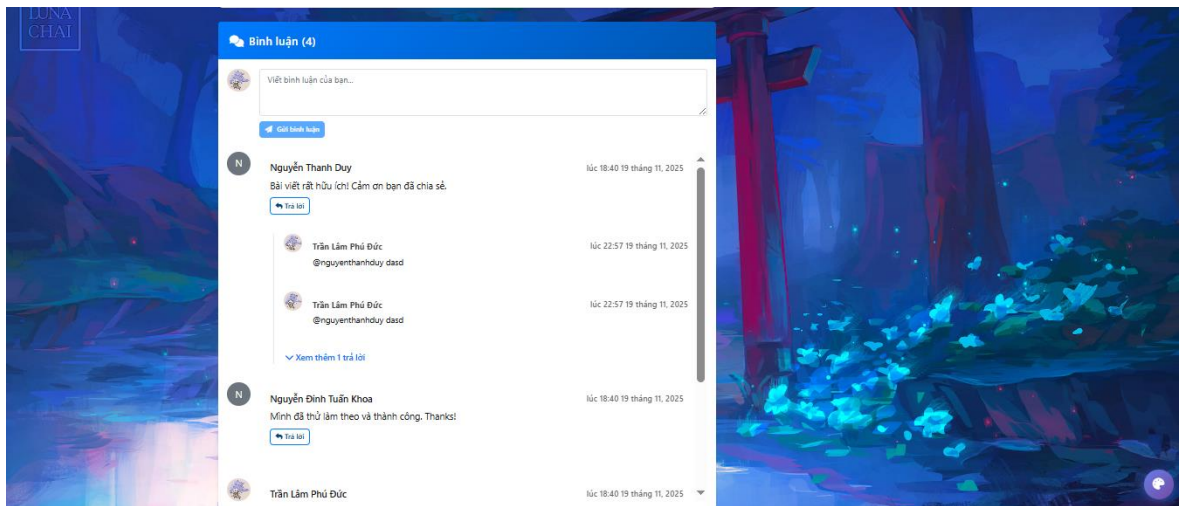
Hình 4.17. Giao diện Chi tiết bài viết

- **Tương tác:** Người dùng có thể thực hiện các hành động như Lưu bài viết (Bookmark) vào kho cá nhân hoặc Báo cáo vi phạm nếu nội dung không phù hợp.



Hình 4.18. Tương tác bài viết

- **Thảo luận:** Hệ thống bình luận được thiết kế theo mô hình phân cấp (Nested Comments). Người dùng có thể trả lời trực tiếp (Reply) một bình luận cụ thể, tạo ra các luồng thảo luận rõ ràng, mạch lạc.



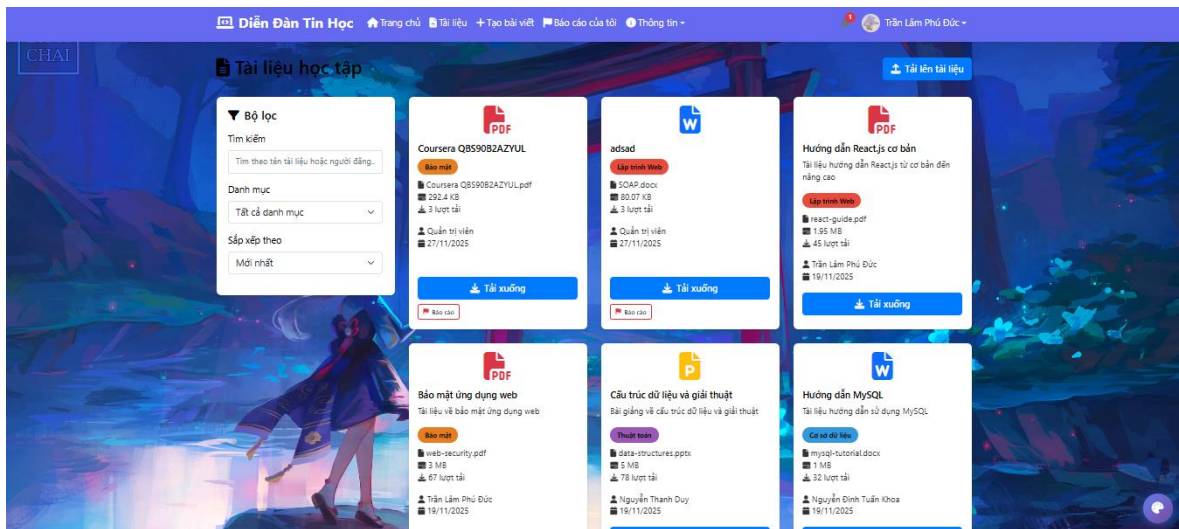
Hình 4.19. Giao diện comment bài viết

4.2.4 Chức năng Thư viện tài liệu

Bên cạnh việc thảo luận, nhu cầu chia sẻ tài liệu học tập (Slide bài giảng, Ebook, Đề thi) là rất lớn. Phân hệ Thư viện tài liệu được xây dựng như một kho lưu trữ tập trung, cho phép người dùng đóng góp và khai thác tài nguyên số một cách dễ dàng.

Giao diện Thư viện và Tìm kiếm tài liệu: Giao diện chính của thư viện hiển thị danh sách các tài liệu hiện có, được trình bày dưới dạng lưới (Grid) hoặc danh sách (List) để tối ưu không gian hiển thị.

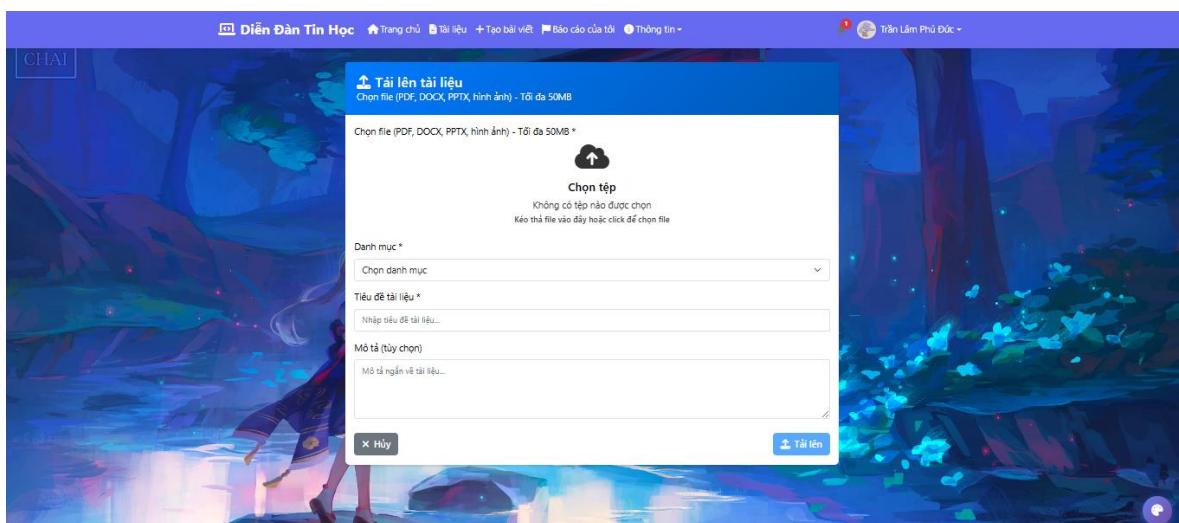
- **Hiển thị:** Mỗi tài liệu đi kèm với các thông tin định danh quan trọng như: Tên tài liệu, Loại tệp (PDF, DOCX, PPTX), Kích thước tệp (File size) và Số lượt tải xuống.
- **Tìm kiếm:** Hệ thống cung cấp bộ lọc nâng cao, cho phép người dùng tìm kiếm tài liệu theo Tên, Chuyên mục hoặc sắp xếp theo mức độ phổ biến (được tải nhiều nhất), giúp sinh viên nhanh chóng tiếp cận nguồn tài liệu cần thiết.



Hình 4.20. Giao diện Danh sách tài liệu

Giao diện Tải lên tài liệu (Document Upload): Chức năng này cho phép các thành viên đóng góp tài nguyên cho cộng đồng.

- Xử lý: Người dùng nhập tiêu đề, mô tả ngắn và chọn tệp tin từ máy tính.
- Kiểm tra (Validation): Để đảm bảo an toàn và tối ưu lưu trữ, hệ thống (Backend) thực hiện kiểm tra nghiêm ngặt về định dạng tệp (chỉ cho phép các định dạng học tập an toàn như .pdf, .docx, .pptx, hình ảnh) và giới hạn kích thước tệp tối đa.
- Lưu trữ: Tệp tin sau khi tải lên thành công sẽ được lưu trữ tại thư mục chỉ định trên Server, đồng thời đường dẫn và metadata được lưu vào cơ sở dữ liệu.



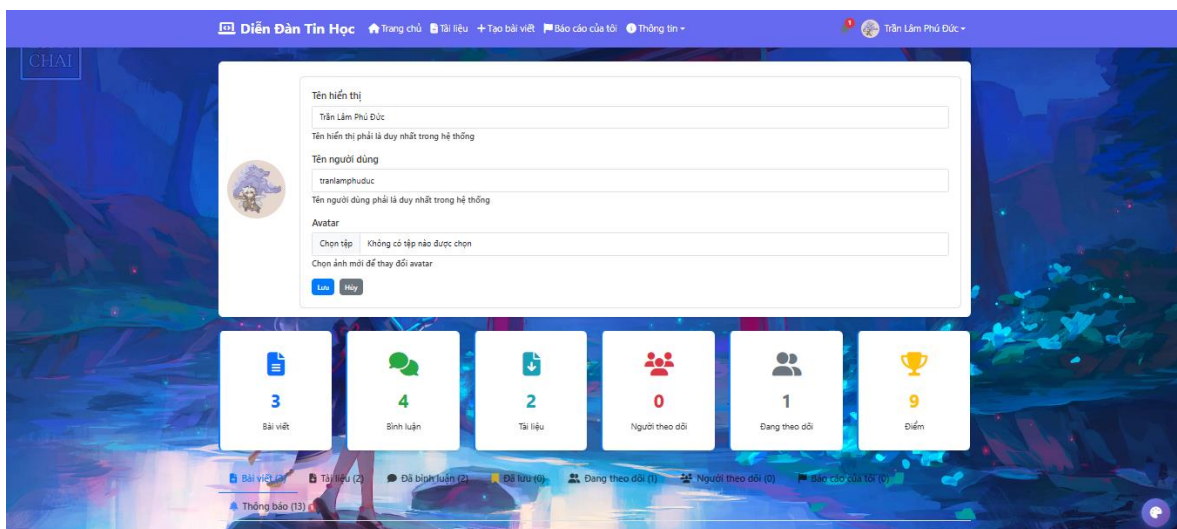
Hình 4.21. Giao diện Upload tài liệu

4.2.5 Trang cá nhân và Cài đặt

Phân hệ quản lý tài khoản cá nhân được xây dựng nhằm giúp người dùng định danh bản thân trên diễn đàn, quản lý các nội dung đã đóng góp và tùy chỉnh trải nghiệm sử dụng hệ thống.

Giao diện Hồ sơ người dùng (Public Profile): Đây là trang hiển thị thông tin công khai của thành viên, giúp tăng tính kết nối trong cộng đồng.

- Thông tin hiển thị: Bao gồm Ảnh đại diện (Avatar), Tên hiển thị, tên người dùng.
- Thống kê: Hiển thị tổng quan số lượng bài viết đã đăng và các tương tác của người dùng với diễn đàn.



Hình 4.22. Giao diện Hồ sơ cá nhân (Profile)

Cấu hình giao diện: Chức năng này cho phép người dùng tự tùy chỉnh giao diện cho cá nhân của mình.

- Cài đặt giao diện (Theme Settings): Dựa trên trường dữ liệu `theme_settings` (JSON) trong cơ sở dữ liệu, hệ thống cho phép người dùng thay đổi linh hoạt hoặc tự tùy chỉnh giao diện cho bản thân, giúp bảo vệ mắt và nâng cao trải nghiệm người dùng trong các điều kiện ánh sáng khác nhau.

Tùy chỉnh giao diện

Nền

Màu sắc

Chữ

Mẫu

Màu nền

#f8f9fa

Ảnh nền

Tải ảnh lên

Ảnh sẽ được lưu trên server và đồng bộ trên mọi thiết bị

Hoặc nhập URL ảnh nền

https://example.com/image.jpg

URL ảnh từ internet (không đồng bộ)

Kích thước ảnh nền

Phủ kín (Cover)

Vị trí ảnh nền

Giữa

Cố định ảnh nền

Cố định khi cuộn

Import

Export

Reset

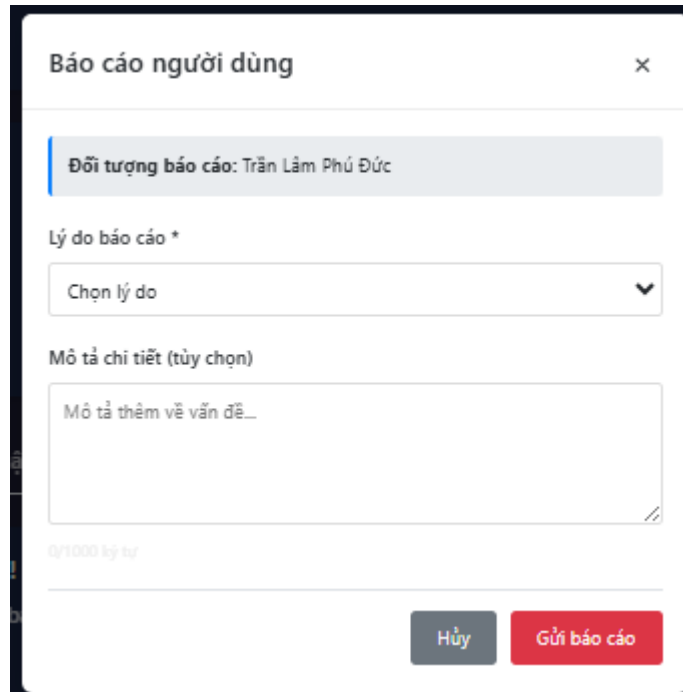
Hình 4.23. Giao diện Hồ sơ cá nhân (Profile)

4.2.6 Chức năng báo cáo

Để duy trì môi trường thảo luận lành mạnh và chất lượng, hệ thống cung cấp công cụ cho phép người dùng báo cáo (Report) các nội dung hoặc hành vi vi phạm tiêu chuẩn cộng đồng. Chức năng này áp dụng cho 3 đối tượng chính: Bài viết, Tài liệu và Người dùng.

Quy trình và Giao diện gửi báo cáo: Khi phát hiện các nội dung vi phạm tiêu chuẩn cộng đồng (như spam, thông tin sai lệch, quấy rối, vi phạm bản quyền...), người dùng có thể gửi phản hồi về cho Ban quản trị thông qua quy trình thống nhất sau:

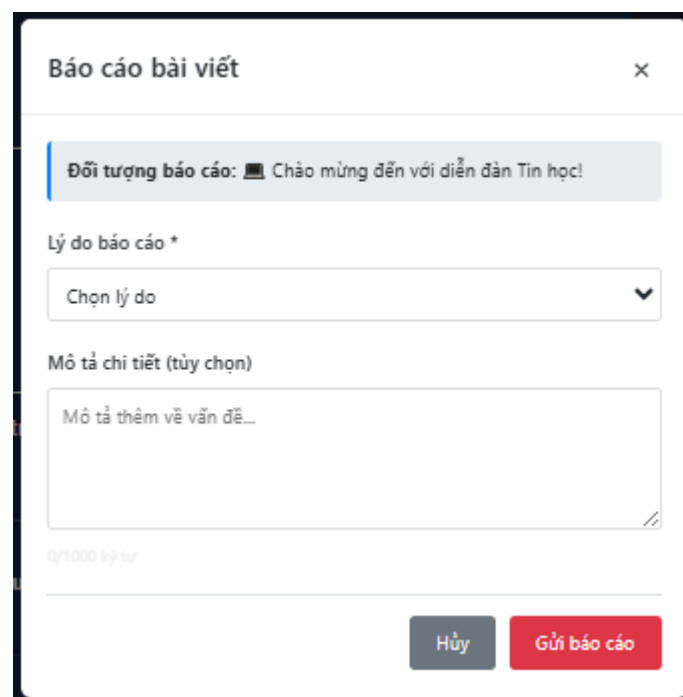
- Kích hoạt: Nhấn vào biểu tượng "Báo cáo" tại bài viết, trang chi tiết tài liệu hoặc hồ sơ người dùng.
- Nhập liệu: Một hộp thoại (Modal) xuất hiện yêu cầu người dùng chọn Lý do báo cáo (tương ứng với từng loại đối tượng) và nhập Mô tả chi tiết.
- Xử lý: Hệ thống ghi nhận thông tin và chuyển trạng thái sang "Chờ duyệt" để Admin xử lý.



The screenshot shows a modal window titled "Báo cáo người dùng" (Report user) with a close button (X) in the top right corner. The form contains the following elements:

- A header bar with the text "Đối tượng báo cáo: Trần Lâm Phú Đức" (Reported object: Trần Lâm Phú Đức).
- A label "Lý do báo cáo *" (Reason for report *) followed by a dropdown menu with the placeholder text "Chọn lý do" (Select reason).
- A label "Mô tả chi tiết (tùy chọn)" (Detailed description (optional)) followed by a text area with the placeholder text "Mô tả thêm về vấn đề..." (Describe the problem further...). Below the text area is a character count "0/1000 ký tự" (0/1000 characters).
- At the bottom right, there are two buttons: "Hủy" (Cancel) and "Gửi báo cáo" (Submit report).

Hình 4.24. Báo cáo người dùng



The screenshot shows a modal window titled "Báo cáo bài viết" (Report article) with a close button (X) in the top right corner. The form contains the following elements:

- A header bar with the text "Đối tượng báo cáo: 📖 Chào mừng đến với diễn đàn Tin học!" (Reported object: 📖 Welcome to the Tin học forum!).
- A label "Lý do báo cáo *" (Reason for report *) followed by a dropdown menu with the placeholder text "Chọn lý do" (Select reason).
- A label "Mô tả chi tiết (tùy chọn)" (Detailed description (optional)) followed by a text area with the placeholder text "Mô tả thêm về vấn đề..." (Describe the problem further...). Below the text area is a character count "0/1000 ký tự" (0/1000 characters).
- At the bottom right, there are two buttons: "Hủy" (Cancel) and "Gửi báo cáo" (Submit report).

Hình 4.25. Báo cáo bài viết

Hình 4.26. Báo cáo tài liệu

Giao diện xem Báo cáo của tôi (My Reports) Hệ thống cung cấp trang "Lịch sử báo cáo" giúp người dùng minh bạch hóa quá trình xử lý phản hồi.

- Chức năng: Người dùng có thể xem lại danh sách tất cả các báo cáo mình đã gửi.
- Trạng thái: Mỗi báo cáo sẽ hiển thị trạng thái xử lý hiện tại: *Pending* (Đang chờ), *Resolved* (Đã xử lý - Nội dung đã bị xóa/xử phạt), hoặc *Dismissed* (Bị từ chối - Nội dung không vi phạm).

ID	Loại	Đối tượng	Lý do	Trạng thái	Ngày gửi	Ghi chú admin
#10	Tài liệu	Hướng dẫn React.js cơ bản	Spam	Chờ xử lý	17:08:14 17/12/2025	-

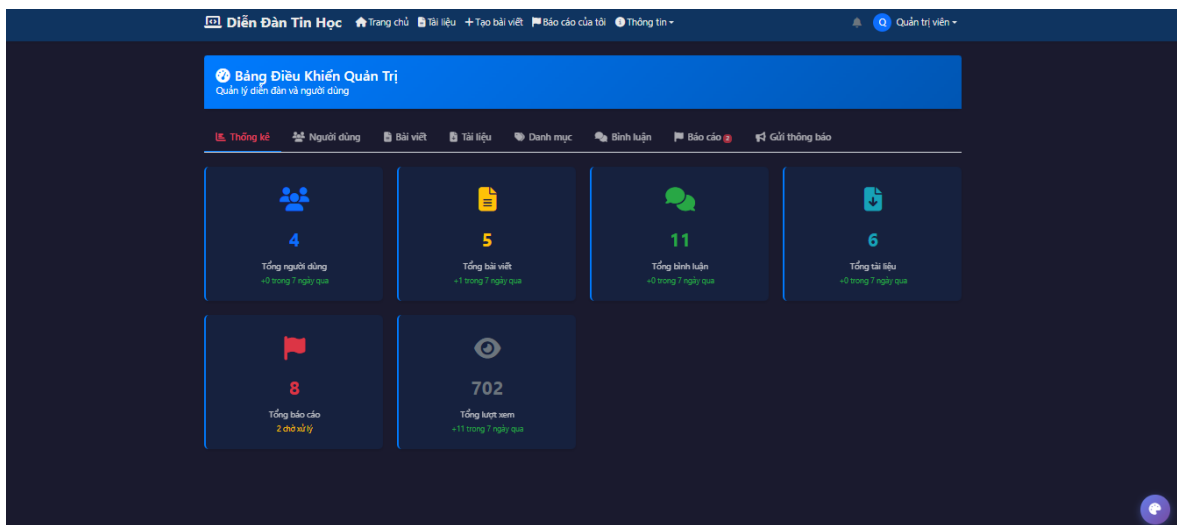
Hình 4.27. Giao diện xem Báo cáo của tôi

4.2.7 Trang Quản trị (Admin Dashboard)

Phân hệ quản trị được xây dựng dành riêng cho Admin, đóng vai trò trung tâm điều phối và kiểm soát mọi hoạt động của diễn đàn. Để truy cập phân hệ này, tài khoản phải có quyền role = 'admin'.

Giao diện Tổng quan Thống kê (Dashboard): Ngay khi vào trang quản trị, hệ thống hiển thị các số liệu quan trọng để Admin nắm bắt tình hình hoạt động.

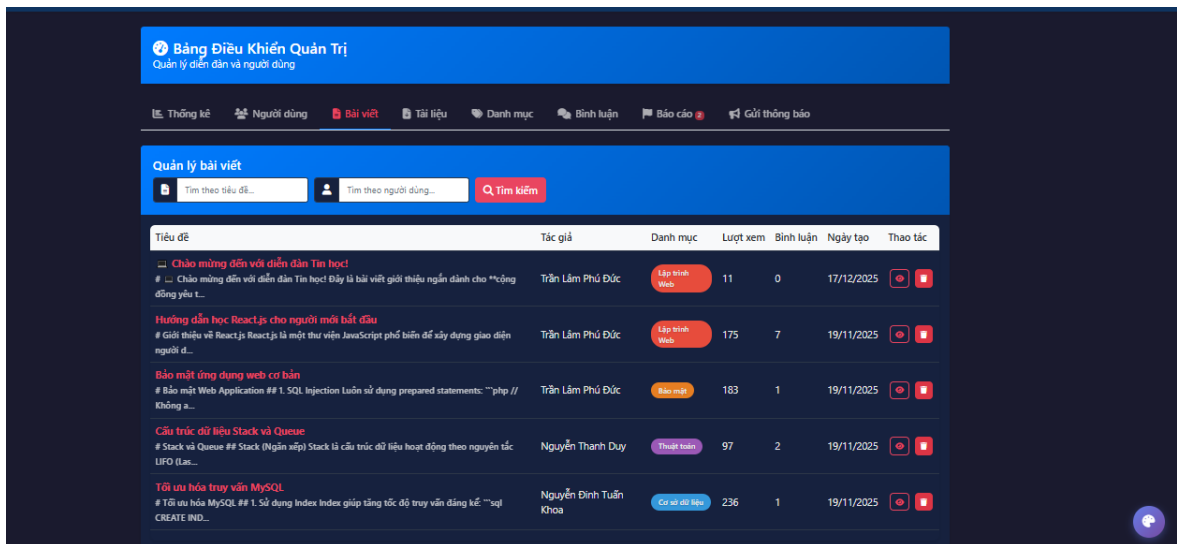
- Thẻ số liệu: Hiển thị tổng số Người dùng, Bài viết, Tài liệu và Số báo cáo chưa xử lý trong ngày.



Hình 4.28. Giao diện Dashboard thống kê

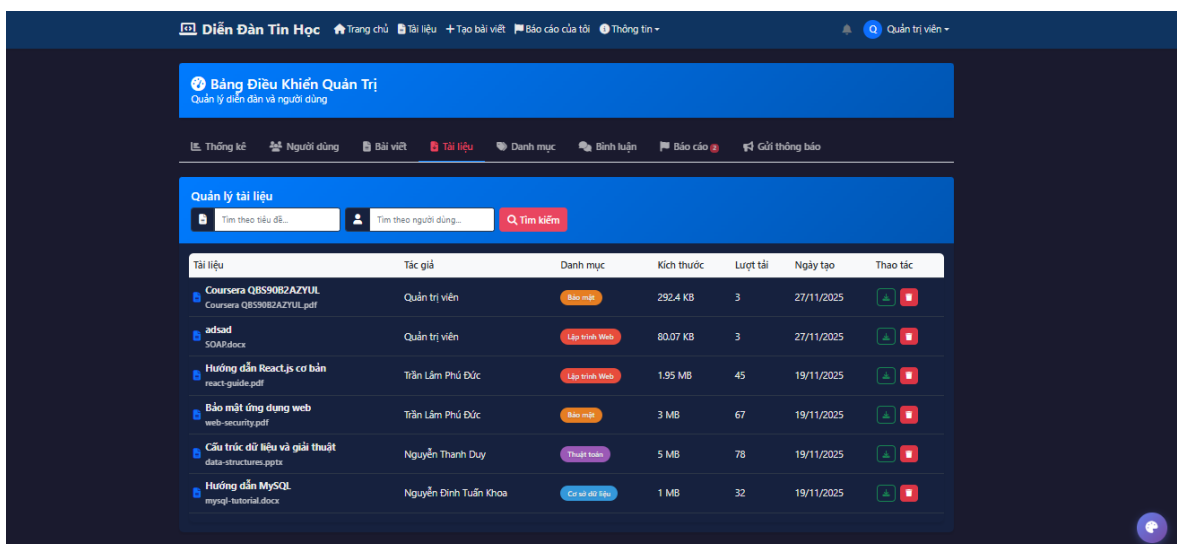
Giao diện Quản lý Bài viết và Tài liệu: Admin có quyền can thiệp vào toàn bộ nội dung được đăng tải để đảm bảo chất lượng.

- Quản lý Bài viết: Cho phép xem danh sách tất cả bài viết trên diễn đàn. Admin có thể thực hiện Ghim bài (Pin) lên đầu trang chủ đối với các bài viết quan trọng hoặc Xóa bài nếu nội dung vi phạm tiêu chuẩn cộng đồng.



Hình 4.29. Giao diện Quản lý bài viết

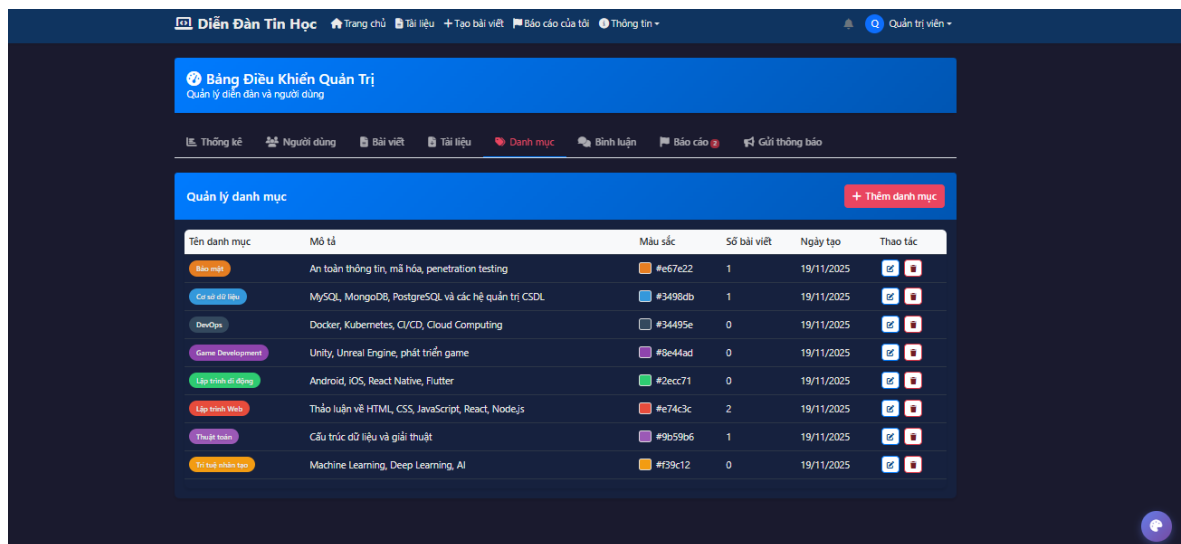
- Quản lý Tài liệu: Kiểm soát các tệp tin được tải lên, cho phép Admin xem trước thông tin và gỡ bỏ các tài liệu không hợp lệ hoặc dính bản quyền.



Hình 4.30. Giao diện Quản lý tài liệu

Giao diện Quản lý Danh mục: Hệ thống cho phép cấu hình động các chuyên mục mà không cần can thiệp vào Code.

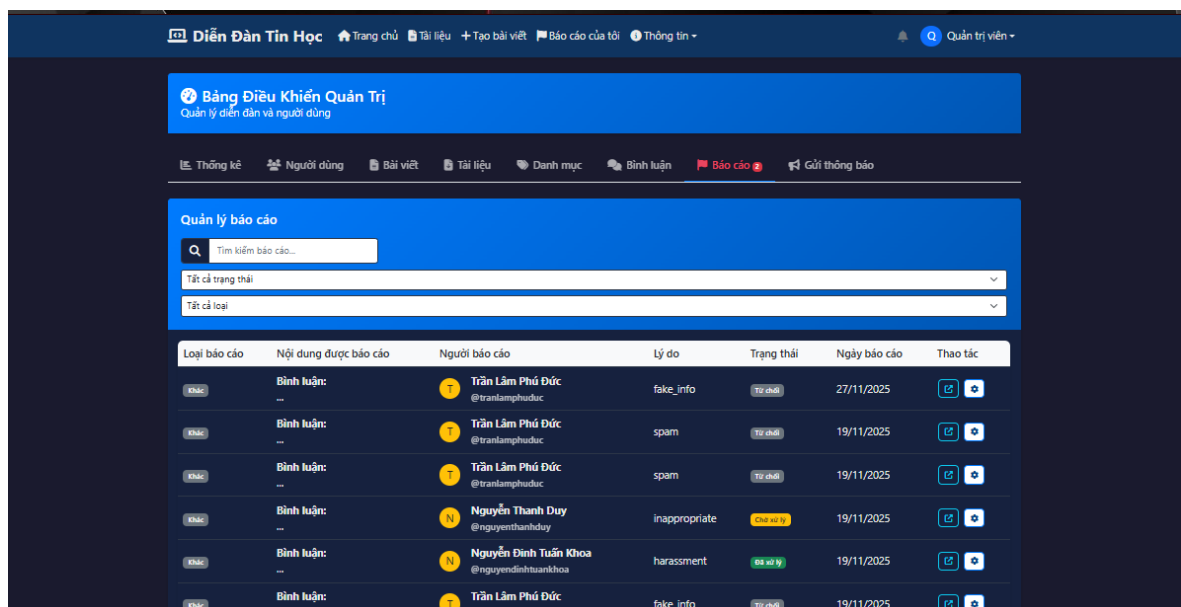
- Chức năng: Admin có thể Thêm mới, Chỉnh sửa tên/mô tả hoặc Xóa các chuyên mục.



Hình 4.31. Giao diện Quản lý danh mục

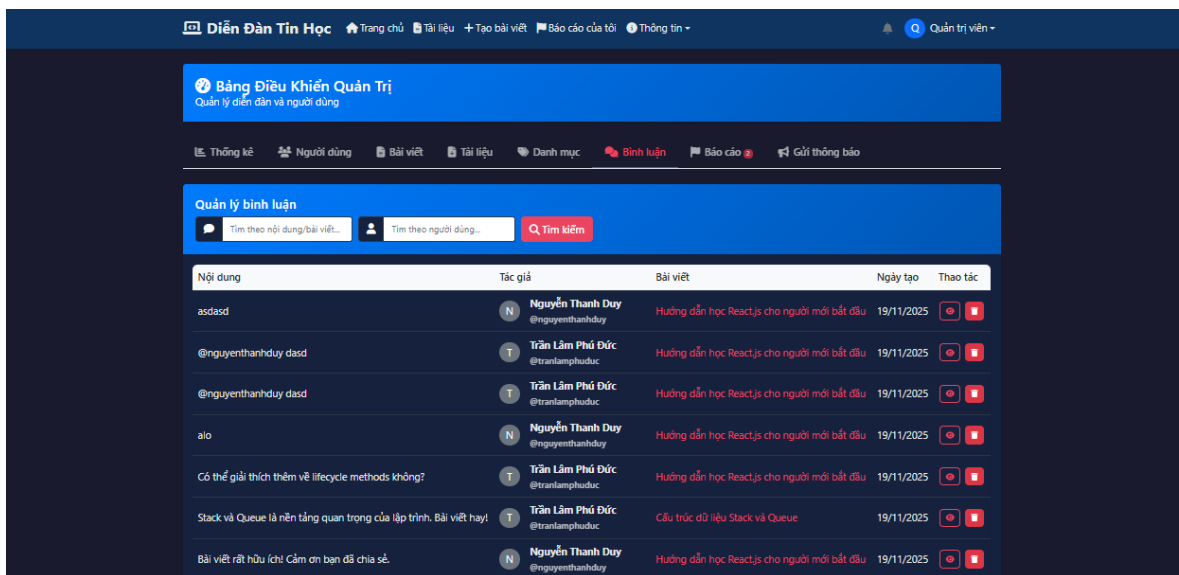
Giao diện Quản lý Bình luận và Báo cáo vi phạm Đây là công cụ chính để duy trì sự trong sạch của diễn đàn (Moderation).

- **Quản lý Báo cáo (Reports):** Danh sách các báo cáo từ người dùng (spam, ngôn từ kích động...) được liệt kê với trạng thái "Chờ xử lý". Admin xem xét lý do báo cáo và đưa ra quyết định: Chấp nhận (đồng nghĩa với việc xóa nội dung/phạt user) hoặc bác bỏ báo cáo đó (đồng nghĩa người báo cáo sai có thể bị phạt nếu báo cáo sai nhiều lần).



Hình 4.32. Giao diện Xử lý báo cáo vi phạm

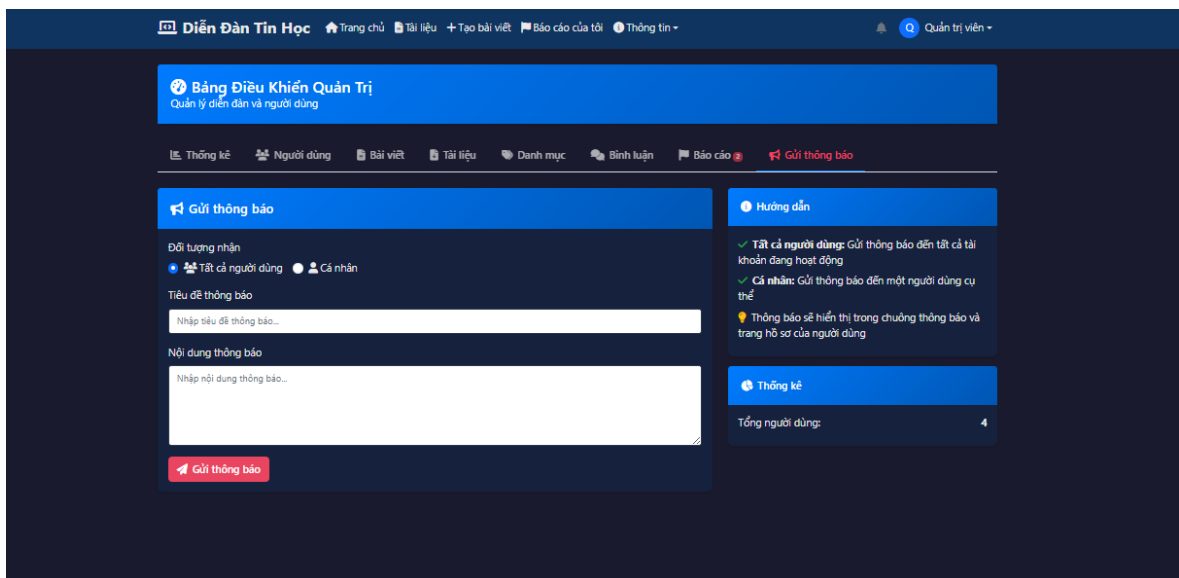
- **Quản lý Bình luận:** Cho phép Admin tìm kiếm và xóa các bình luận spam hoặc thiếu văn hóa ngay cả khi chưa có ai báo cáo.



Hình 4.33. Giao diện Quản lý bình luận

Giao diện Gửi thông báo hệ thống (Notifications) Chức năng giúp Admin gửi thông điệp đến cộng đồng.

- Xử lý: Admin soạn tiêu đề, nội dung thông báo và chọn đối tượng nhận (Gửi cho một người cụ thể hoặc Gửi toàn bộ hệ thống).
- Kết quả: Thông báo sẽ xuất hiện trên thanh menu của người dùng và được lưu vào bảng notifications.



Hình 4.34. Giao diện Gửi thông báo hệ thống

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Sau quá trình nghiên cứu, phân tích và triển khai thực hiện đề tài "**Xây dựng Diễn đàn Tin học**", đồ án đã hoàn thành các mục tiêu đề ra ban đầu và đạt được những kết quả cụ thể sau:

5.1.1 Về mặt lý thuyết:

- Nắm vững quy trình phát triển phần mềm theo mô hình thác nước, từ khâu khảo sát yêu cầu, phân tích thiết kế hệ thống đến lập trình và kiểm thử.
- Hiểu sâu về kiến trúc ứng dụng web 3 tầng (Three-Tier Architecture) và cách áp dụng vào thực tế.
- Làm chủ các công nghệ lập trình web hiện đại: **ReactJS** (Frontend), **NodeJS/ExpressJS** (Backend) và **MySQL** (Database).

5.1.2 Về mặt thực tiễn (Sản phẩm):

- Xây dựng thành công website diễn đàn với đầy đủ các chức năng cốt lõi: Đăng ký/Đăng nhập, Quản lý bài viết, Thảo luận (Bình luận), và Chia sẻ tài liệu.
- Hệ thống hỗ trợ tốt việc soạn thảo văn bản kỹ thuật thông qua **Markdown** và **Syntax Highlighting**, giải quyết đúng nhu cầu của sinh viên ngành CNTT.
- Cơ chế phân quyền chặt chẽ giữa Khách, Thành viên và Quản trị viên, đảm bảo tính bảo mật và an toàn dữ liệu.
- Giao diện người dùng được thiết kế trực quan, hiện đại và thân thiện (User-friendly).

5.2. Hạn chế

Bên cạnh những kết quả đạt được, do giới hạn về thời gian và kinh nghiệm thực tế, đồ án vẫn còn tồn tại một số hạn chế cần khắc phục:

- **Khả năng mở rộng:** Hệ thống chưa được kiểm thử chịu tải (Load Testing) với lượng dữ liệu lớn (Big Data) hoặc lượng người dùng truy cập đồng thời cao.

- **Tìm kiếm:** Chức năng tìm kiếm hiện tại dựa trên truy vấn cơ sở dữ liệu thuần túy (LIKE query), chưa tích hợp các công cụ tìm kiếm toàn văn mạnh mẽ (như Elasticsearch) để tối ưu hóa kết quả và hiệu suất.
- **Nền tảng:** Ứng dụng mới chỉ dừng lại ở nền tảng Web, chưa có phiên bản ứng dụng di động (Mobile App) riêng biệt.

5.3. Hướng phát triển

Để hoàn thiện sản phẩm và nâng cao tính ứng dụng trong thực tế, các hướng phát triển tiếp theo được đề xuất như sau:

5.3.1 Nâng cấp công nghệ:

Sử dụng **Redis** để Cache dữ liệu, giúp tăng tốc độ tải trang và giảm tải cho Database.

5.3.2 Mở rộng chức năng:

- Phát triển hệ thống **Gamification** (Điểm thưởng, Danh hiệu) để khuyến khích thành viên tích cực đóng góp bài viết và tài liệu chất lượng.
- Tích hợp **AI (Trí tuệ nhân tạo)** để tự động kiểm duyệt nội dung xấu (spam, ngôn từ đả kích) và gợi ý bài viết phù hợp với sở thích người dùng.
- Thêm tính năng đăng nhập nhanh qua Google/Facebook (Social Login).

5.3.3 Đa nền tảng:

Phát triển phiên bản ứng dụng di động (sử dụng **React Native**) để sinh viên có thể truy cập diễn đàn thuận tiện hơn trên điện thoại thông minh.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] N. Đ. Thuân and M. X. Hùng, Giáo trình Phát triển ứng dụng Web, TP. Hồ Chí Minh: Giáo trình Phát triển ứng dụng Web, Nhà xuất bản Đại học Quốc gia TP. Hồ Chí Minh, 2015.
- [2] Remix Software Inc., "React Router - Declarative Routing for React.js," [Online]. Available: <https://reactrouter.com/>. [Accessed 20 11 2025].
- [3] Facebook Inc., "React - A JavaScript library for building user interfaces," [Online]. Available: <https://react.dev/>. [Accessed 20 11 2025].
- [4] OpenJS Foundation, "Node.js - JavaScript runtime built on Chrome's V8 JavaScript engine," [Online]. Available: <https://nodejs.org/>. [Accessed 20 11 2025].
- [5] StrongLoop, IBM, "Express - Fast, unopinionated, minimalist web framework for Node.js," [Online]. Available: <https://expressjs.com/>. [Accessed 20 11 2025].
- [6] Oracle Corporation, "MySQL Documentation," [Online]. Available: <https://dev.mysql.com/doc/>. [Accessed 20 11 2025].
- [7] Internet Engineering Task Force, "RFC 7519 - JSON Web Token (JWT)," 5 2015. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc7519>. [Accessed 20 11 2025].
- [8] ExpressJS Ecosystem, "Multer - Node.js middleware for handling multipart/form-data," [Online]. Available: <https://www.npmjs.com/package/multer>. [Accessed 20 11 2025].