

**TRƯỜNG ĐẠI HỌC TRÀ VINH  
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ**



**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG DIỄN ĐÀN TRỰC TUYẾN  
CHO SINH VIÊN TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giảng viên hướng dẫn : **ThS LÊ MINH TỰ**  
Sinh viên thực hiện: **PHẠM ĐĂNG KHOA**  
Mã số sinh viên: **110121158**  
Lớp : **DA21TTC**  
Khoá : **2021**

Trà Vinh, tháng ... năm 2025

**TRƯỜNG ĐẠI HỌC TRÀ VINH  
TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ**



**KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG DIỄN ĐÀN TRỰC TUYẾN  
CHO SINH VIÊN TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giảng viên hướng dẫn : **ThS LÊ MINH TỰ**  
Sinh viên thực hiện: **PHẠM ĐĂNG KHOA**  
Mã số sinh viên: **110121158**  
Lớp : **DA21TTC**  
Khoá : **2021**

Trà Vinh, tháng ... năm 2025

## LỜI MỞ ĐẦU

### 1. Lý do chọn đề tài

Trong bối cảnh chuyển đổi số đang lan rộng trong lĩnh vực giáo dục, việc ứng dụng công nghệ thông tin vào hoạt động học tập và giao tiếp trong môi trường đại học ngày càng trở nên thiết yếu. Các nền tảng trực tuyến không chỉ hỗ trợ trao đổi kiến thức mà còn thúc đẩy sự gắn kết giữa sinh viên, giảng viên và nhà trường, góp phần nâng cao hiệu quả học tập và phát triển kỹ năng mềm.

Tại các trường đại học, đặc biệt là ở những địa phương đang từng bước hiện đại hóa như Trà Vinh, sinh viên vẫn chủ yếu trao đổi thông tin qua các nhóm mạng xã hội thiếu tính hệ thống và khó kiểm soát nội dung. Điều này gây ra nhiều bất cập trong việc tìm kiếm tài liệu học tập, chia sẻ kinh nghiệm, cũng như kết nối cộng đồng sinh viên theo từng chuyên ngành.

Việc xây dựng một diễn đàn trực tuyến dành riêng cho sinh viên Trường Đại học không chỉ giúp tổ chức lại hoạt động trao đổi học thuật và đời sống sinh viên một cách bài bản, an toàn mà còn góp phần thúc đẩy phong trào học tập, hỗ trợ học tập cộng đồng và phát triển môi trường học đường hiện đại, thân thiện.

### 2. Mục đích nghiên cứu

Đề tài nhằm xây dựng một nền tảng diễn đàn trực tuyến chuyên biệt dành cho sinh viên Trường Đại học, giúp tăng cường khả năng trao đổi học tập, chia sẻ tài liệu, thảo luận chuyên môn và kết nối cộng đồng sinh viên theo từng chuyên ngành.

Qua diễn đàn, sinh viên có thể chủ động tiếp cận nguồn tri thức mở, phát triển kỹ năng giao tiếp học thuật và xây dựng môi trường học tập tích cực, hỗ trợ lẫn nhau. Đồng thời, hệ thống cũng góp phần hiện đại hóa phương thức giao tiếp trong nhà trường, thúc đẩy quá trình chuyển đổi số trong giáo dục đại học.

### 3. Đối tượng và phạm vi nghiên cứu

**Sinh viên:** Là người dùng chính của diễn đàn, có thể đăng ký tài khoản, đăng bài viết, tham gia thảo luận, chia sẻ tài liệu, đặt câu hỏi và trả lời trong các chuyên mục liên quan đến học tập và đời sống sinh viên.

**Quản trị viên:** Là người quản lý toàn bộ hệ thống, có quyền tạo và phân loại chủ đề, kiểm duyệt nội dung bài viết, quản lý người dùng, xử lý vi phạm và đảm bảo hoạt động ổn định, an toàn cho diễn đàn.

Hệ thống được xây dựng và thử nghiệm trong phạm vi Trường Đại học, tập trung vào nhu cầu trao đổi học thuật, kinh nghiệm học tập và kết nối sinh viên trong trường.

#### **4. Phương pháp nghiên cứu**

##### **Phương pháp lý thuyết:**

Nghiên cứu, thu thập và phân tích các tài liệu liên quan đến hệ thống diễn đàn trực tuyến, mô hình hoạt động của các nền tảng trao đổi học tập, giao tiếp trong môi trường giáo dục đại học.

Tìm hiểu các công nghệ, công cụ và nền tảng hỗ trợ xây dựng hệ thống web như React.js, Express.js, cơ sở dữ liệu MongoDB và các mô hình bảo mật người dùng.

Khảo sát nhu cầu và hành vi sử dụng diễn đàn của sinh viên thông qua bảng hỏi, thống kê và phân tích để định hướng thiết kế phù hợp với thực tế.

##### **Phương pháp thực nghiệm:**

- Thiết kế và xây dựng giao diện người dùng dựa trên các yêu cầu đã phân tích.
- Xây dựng hệ thống backend và cơ sở dữ liệu để xử lý các chức năng như đăng ký, đăng nhập, đăng bài viết, bình luận, phân loại chuyên mục và quản lý người dùng.
- Tiến hành kiểm thử hệ thống trên môi trường giả lập và thực tế để đánh giá hiệu năng, tính ổn định và mức độ thân thiện với người dùng.
- Ghi nhận ý kiến phản hồi từ người dùng thử nghiệm (sinh viên và quản trị viên) để cải tiến hệ thống.

## LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn đến quý thầy, cô Trường Đại học Trà Vinh, đặc biệt là các thầy, cô thuộc Khoa Công nghệ Thông tin trong Trường Kỹ thuật và Công nghệ, đã tạo điều kiện thuận lợi để tôi có cơ hội thực hành và tiếp cận với thực tế, giúp tôi có thêm kinh nghiệm trong quá trình học tập và làm việc, giảm bớt những khó khăn và bỡ ngỡ trong quá trình chuẩn bị cho môi trường làm việc sau này.

Tôi cũng trân trọng cảm ơn thầy Lê Minh Tự, người đã tận tình hướng dẫn và hỗ trợ tôi trong suốt quá trình thực hiện khóa luận. Sự đồng hành và chỉ dẫn của Thầy đã giúp tôi hoàn thành công việc đúng thời hạn, tích lũy thêm nhiều kiến thức cần thiết cho bản thân.

Do hạn chế về thời gian cũng như kiến thức chuyên môn, nội dung đề tài của tôi vẫn còn những điểm cần cải thiện. Tôi rất mong nhận được sự đóng góp ý kiến từ các thầy, cô để hoàn thiện và tốt hơn trong tương lai.

Tôi xin chân thành cảm ơn.

## NHẬN XÉT

## **NHẬN XÉT**

## **Giảng viên hướng dẫn** (ký và ghi rõ họ tên)

# BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của giảng viên hướng dẫn)

Họ và tên sinh viên: ..... MSSV: .....

Ngành: ..... Khóa: .....

Tên đê tài: .....

Ho và tên Giáo viên hướng dẫn: .....

Chức danh: ..... Học vi: .....

## NHÂN XÉT

- ## 1. Nội dung đề tài:

- ## 2. Ưu điểm:

.....  
.....  
.....

- ### 3. Khuyết điểm:

.....  
.....  
.....  
.....

- #### 4 Điểm mới đề tài:

.....  
.....  
.....  
.....  
.....

5. Giá trị thực trên đè tài:

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

7. Đề nghị sửa chữa bổ sung:

.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....

8. Đánh giá:

.....  
.....  
.....  
.....

Trà Vinh, ngày tháng năm 20...  
Giảng viên hướng dẫn  
(Ký & ghi rõ họ tên)

## **NHẬN XÉT**

**(Của giảng viên chấm trong đồ án, khoá luận của sinh viên)**

## **Giảng viên chấm**

# BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP

(Của cán bộ chấm đồ án, khóa luận)

Họ và tên người nhận xét: .....

Chức danh: ..... Học vị: .....

Chuyên ngành: .....

Cơ quan công tác: .....

Tên sinh viên: .....

Tên đề tài đồ án, khóa luận tốt nghiệp: .....

## I. Ý KIẾN NHÂN XÉT

## 1. Nội dung:

.....

.....

.....

.....

.....

.....

.....

.....

## 2. Điểm mới các kết quả của đồ án, khóa luận:

.....

### 3. Ứng dụng thực tế:

---

---

---

---

---

## **II. CÁC VẤN ĐỀ CẦN LÀM RÕ**

(Các câu hỏi của giáo viên phản biện)

### III. KẾT LUẬN

(Ghi rõ đồng ý hay không đồng ý cho bảo vệ đồ án khóa luận tốt nghiệp)

.....  
.....  
.....  
.....

....., ngày ..... tháng ..... năm 20..

## Người nhận xét

(Ký & ghi rõ họ tên)

## MỤC LỤC

Trang

LỜI MỞ ĐẦU .....	ii
1. Lý do chọn đề tài.....	ii
2. Mục đích nghiên cứu .....	ii
3. Đối tượng và phạm vi nghiên cứu .....	ii
4. Phương pháp nghiên cứu .....	iii
CHƯƠNG 1 : TỔNG QUAN ĐỀ TÀI .....	1
CHƯƠNG 2 : NGHIÊN CỨU LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG .....	3
2.1 Javascript.....	3
2.1.1 Sơ lược về ngôn ngữ Javascript.....	3
2.1.2 Đặc điểm .....	3
2.1.3 Cách hoạt động .....	3
2.2 React framework.....	4
2.2.1 Giới thiệu .....	4
2.2.2 Đặc điểm nổi bật .....	4
2.2.3 Các thành phần quan trọng .....	4
2.3 NodeJs.....	6
2.3.1 Giới thiệu .....	6
2.3.2 Đặc điểm vượt trội .....	6
2.3.3 Cơ chế hoạt động .....	6
2.4 Express.js framework.....	8
2.4.1 Giới thiệu .....	8
2.4.2 Các tính năng chính .....	8
2.5 Mô hình MVC.....	9
2.5.1 Tổng quan về mô hình MVC .....	9
2.5.2 Các thành phần.....	11
2.5.3 Ưu điểm và Nhược điểm.....	11
2.6 Material UI (MUI) .....	12
2.6.1 Khái quát về Material UI (MUI).....	12
2.6.2 Các ưu điểm .....	12
2.7 Hệ quản trị cơ sở dữ liệu MongoDB.....	13
2.7.1 Giới thiệu .....	13
2.7.2 Cách hoạt động .....	13
2.7.3 Ưu điểm và Nhược điểm.....	14
2.8 Chatbot AI.....	15

2.8.1 Tổng quan .....	15
2.8.2 Lợi ích .....	15
2.8.3 Các chatbot AI hiện nay.....	15
2.9 WebSocket và công nghệ Realtime .....	15
<b>CHƯƠNG 3 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG .....</b>	<b>16</b>
3.1 Mô tả bài toán .....	16
3.2 Xây dựng websites .....	18
3.2.1 Cài đặt môi trường lập trình.....	18
3.2.2 Cơ sở dữ liệu .....	20
3.3 Thiết kế xử lý .....	38
3.3.1 Sơ đồ chức năng.....	38
3.3.2 Biểu đồ Use Case tổng quát.....	39
3.3.3 Biểu đồ Use Case tác nhân người dùng chưa đăng nhập (khách) .....	40
3.3.4 Biểu đồ Use Case tác nhân Sinh viên (thành viên, người dùng) .....	41
3.3.5 Biểu đồ Use case tác nhân Quản trị .....	42
3.3.6 Sơ đồ tuần tự .....	42
3.3.7 Sơ đồ hoạt động .....	50
<b>CHƯƠNG 4 : KẾT QUẢ NGHIÊN CỨU.....</b>	<b>56</b>
4.1 Giao diện người dùng .....	56
4.1.1 Giao diện trang chủ .....	56
4.1.2 Giao diện trang chủ với phần các Chủ đề .....	58
4.1.3 Giao diện trang quy định diễn đàn.....	58
4.1.4 Giao diện trang chi tiết các Chủ đề.....	59
4.1.5 Giao diện trang chi tiết bài viết.....	59
4.1.6 Giao diện trang cá nhân .....	60
4.1.7 Giao diện trang các thành viên diễn đàn.....	61
4.1.8 Giao diện trang trò chuyện.....	62
4.1.9 Giao diện trang đăng bài, câu hỏi thảo luận .....	63
4.1.10 Giao diện trang bình luận.....	63
4.1.11 Giao diện trang đánh giá.....	64
4.1.12 Giao diện trang tương tác lượt thích .....	64
4.1.13 Giao diện trang Chat với ChatBot .....	65
4.1.14 Giao diện hộp thoại thông báo của người dùng:.....	65
4.1.15 Giao diện trang Đăng nhập .....	66
4.1.16 Giao diện trang Đăng ký .....	66
4.1.17 Giao diện trang quên mật khẩu .....	67

4.2 Giao diện quản trị.....	68
4.2.1 Giao diện trang chủ quản trị .....	68
4.2.2 Giao diện trang quản lý bài viết.....	69
4.2.3 Giao diện trang quản lý chủ đề .....	69
4.2.4 Giao diện trang quản lý, huấn luyện ChatBot.....	70
4.2.5 Giao diện trang Thông kê, phân tích.....	70
4.2.6 Giao diện trang quản lý bình luận.....	71
4.2.7 Giao diện trang cá nhân của quản trị viên .....	71
4.2.8 Giao diện trang quản lý các chủ đề thịnh hành và các bài viết nổi bật.....	73
4.2.9 Giao diện trang quản lý dữ liệu .....	73
4.2.10 Giao diện trang quản lý thông báo.....	74
4.2.11 Giao diện hộp thoại (nhận thông báo) của quản trị viên.....	74
4.2.12 Giao diện trang quản lý quy định diễn đàn.....	75
CHƯƠNG 5 : HƯỚNG PHÁT TRIỂN .....	76
5.1 Kết luận .....	76
5.2 Hướng phát triển .....	76
PHỤ LỤC.....	78
DANH MỤC TÀI LIỆU THAM KHẢO.....	79

## **DANH MỤC HÌNH ẢNH**

Hình 2.1 Thư mục Frontend cơ bản.....	5
Hình 2.2 Các ứng dụng phổ biến của NodeJs [6] .....	7
Hình 2.3 Thư mục backend cơ bản .....	7
Hình 2.4 MVC [9].....	10
Hình 2.5 MongoDB [11].....	13
Hình 3.1 Sơ đồ chức năng.....	38
Hình 3.2 Biểu đồ Use Case tổng quát.....	39
Hình 3.3 Biểu đồ Use Case tác nhân khách (chưa đăng nhập).....	40
Hình 3.4 Biểu đồ Use Case tác nhân Sinh viên .....	41
Hình 3.5 Biểu đồ Use Case tác nhân Quản trị .....	42
Hình 3.6 Sơ đồ tuần tự đăng bài của thành viên .....	44
Hình 3.7 Sơ đồ tuần tự bình luận bài viết .....	46
Hình 3.8 Sơ đồ tuần tự tương tác lượt thích .....	48
Hình 3.9 Sơ đồ tuần tự đánh giá bài viết .....	50
Hình 3.10 Sơ đồ hoạt động đăng bài viết.....	51
Hình 3.11 Sơ đồ hoạt động bình luận bài viết .....	52
Hình 3.12 Sơ đồ hoạt động tương tác bài viết bằng lượt thích.....	53
Hình 3.13 Sơ đồ hoạt động đánh giá bài viết.....	54
Hình 4.1 Giao diện Trang chủ (a) .....	56
Hình 4.2 Giao diện trang chủ (b) .....	56
Hình 4.3 Giao diện trang chủ (c) .....	57
Hình 4.4 Giao diện trang chủ (d) .....	57
Hình 4.5 Giao diện Trang chủ với các Chủ đề .....	58
Hình 4.6 Giao diện trang quy định diễn đàn.....	58
Hình 4.7 Giao diện trang chi tiết Chủ đề .....	59
Hình 4.8 Giao diện trang chi tiết bài viết.....	59
Hình 4.9 Giao diện trang cá nhân người dùng (a) .....	60
Hình 4.10 Giao diện trang cá nhân người dùng (b) .....	60
Hình 4.11 Giao diện trang cá nhân người dùng (c) .....	61
Hình 4.12 Giao diện trang các thành viên diễn đàn.....	61
Hình 4.13 Giao diện trang trò chuyện (a) .....	62

Hình 4.14 Giao diện trang trò chuyện (b) .....	62
Hình 4.15 Giao diện trang đăng bài, câu hỏi thảo luận .....	63
Hình 4.16 Giao diện trang Bình luận .....	63
Hình 4.17 Giao diện trang Đánh giá .....	64
Hình 4.18 Giao diện trang tương tác lượt thích .....	64
Hình 4.19 Giao diện trò chuyện với ChatBot .....	65
Hình 4.20 Giao diện hộp thoại thông báo của người dùng .....	65
Hình 4.21 Giao diện trang Đăng nhập .....	66
Hình 4.22 Giao diện trang Đăng ký .....	66
Hình 4.23 Giao diện trang quên mật khẩu .....	67
Hình 4.24 Nội dung Email nhận được khi đặt lại mật khẩu .....	67
Hình 4.25 Giao diện đặt lại mật khẩu .....	68
Hình 4.26 Giao diện trang admin .....	68
Hình 4.27 Giao diện trang Quản lý bài viết .....	69
Hình 4.28 Giao diện trang Quản lý chủ đề .....	69
Hình 4.29 Giao diện trang Quản lý, huấn luyện ChatBot .....	70
Hình 4.30 Giao diện trang Thống kê, phân tích .....	70
Hình 4.31 Giao diện trang Quản lý Bình luận .....	71
Hình 4.32 Giao diện trang cá nhân của quản trị viên (a) .....	71
Hình 4.33 Giao diện trang cá nhân của quản trị viên (b) .....	72
Hình 4.34 Giao diện trang cá nhân của quản trị viên (c) .....	72
Hình 4.35 Giao diện trang quản lý chủ đề thịnh hành, bài viết nổi bật .....	73
Hình 4.36 Giao diện trang quản lý dữ liệu .....	73
Hình 4.37 Giao diện trang quản lý thông báo .....	74
Hình 4.38 Giao diện hộp thoại thông báo .....	74
Hình 4.39 Giao diện trang quản lý quy định diễn đàn .....	75

## **DANH MỤC BẢNG BIỂU**

Bảng 3.1 Quy trình đăng bài trên hệ thống.....	43
Bảng 3.2 Quy trình bình luận bài viết trên diễn đàn.....	45
Bảng 3.3 Quy trình tương tác bài viết bằng lượt thích .....	47
Bảng 3.4 Quy trình đánh giá bài viết.....	49

## KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT

Từ viết tắt	Điễn giải
Js/js	Javascript
n.d.	no date (không có ngày đăng bài)
MVC	Model-view-controller
Frm	Form
Ctr	Controller
CSDL/csdl	Cơ sở dữ liệu

## CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

---

Trong thời đại công nghệ thông tin phát triển mạnh mẽ, việc ứng dụng công nghệ vào môi trường học đường không chỉ mang lại hiệu quả cao trong việc giảng dạy mà còn tạo điều kiện thuận lợi cho sinh viên trong việc học tập, trao đổi kiến thức và kết nối cộng đồng. Diễn đàn trực tuyến là một trong những công cụ hữu ích giúp sinh viên có thể thảo luận, chia sẻ tài liệu học tập, trao đổi kinh nghiệm cũng như cập nhật thông tin từ nhà trường một cách nhanh chóng và thuận tiện. Tuy nhiên, tại Trường Đại học Trà Vinh hiện nay vẫn chưa có một nền tảng diễn đàn chuyên biệt phục vụ cho sinh viên toàn trường. Nhiều hoạt động thảo luận đang diễn ra rời rạc qua các nền tảng mạng xã hội, thiếu tính hệ thống và khó kiểm soát về mặt nội dung lẫn bảo mật.

Trước thực trạng đó, việc xây dựng một diễn đàn trực tuyến dành riêng cho sinh viên Trường Đại học Trà Vinh là một nhu cầu cấp thiết, giúp hình thành một không gian trao đổi học thuật và tương tác cộng đồng lành mạnh, hiệu quả, đồng thời hỗ trợ nhà trường trong việc nắm bắt ý kiến, nhu cầu và hỗ trợ sinh viên một cách kịp thời.

Là một sinh viên ngành Công nghệ thông tin, tôi nhận thấy rằng việc phát triển một hệ thống diễn đàn trực tuyến mang lại nhiều lợi ích thiết thực không chỉ trong lĩnh vực kỹ thuật mà còn góp phần tạo ra giá trị xã hội, đặc biệt là trong môi trường học đường. Diễn đàn không chỉ đơn thuần là nơi để hỏi và trả lời câu hỏi, mà còn là nơi tạo dựng cộng đồng, phát triển tư duy phản biện, khuyến khích sự chủ động trong học tập và kết nối sinh viên các khóa, các ngành.

Việc chọn đề tài này còn xuất phát từ thực tiễn sử dụng mạng xã hội hiện nay. Nhiều nhóm trao đổi học tập được lập ra trên Facebook, Zalo... nhưng đều mang tính chất tạm thời, thiếu công cụ tổ chức nội dung chuyên sâu, dễ trôi bài, và không có tính năng chuyên biệt dành cho sinh viên. Tôi mong muốn phát triển một nền tảng ổn định, thân thiện với người dùng, phù hợp với nhu cầu trao đổi của sinh viên Trường Đại học Trà Vinh tích hợp với các hệ thống khác của nhà trường trong tương lai.

Phạm vi thực hiện của đề tài bao gồm việc thiết kế và phát triển một nền tảng diễn đàn trực tuyến dưới dạng website, tập trung vào đối tượng sử dụng chính là sinh viên và giảng viên Trường Đại học Trà Vinh. Trong phạm vi đồ án này, hệ thống sẽ được xây dựng với các chức năng cơ bản như: đăng ký/đăng nhập người dùng, tạo và quản lý bài viết, phản hồi bài viết, phân loại chuyên mục, tìm kiếm và quản lý tài khoản.

Hệ thống không tập trung vào các tính năng nâng cao như livestream, học trực tuyến, tích hợp AI,... Các chức năng bảo mật sẽ được triển khai ở mức cơ bản nhằm đảm bảo an toàn cho dữ liệu người dùng trong khuôn khổ dự án. Đề tài sẽ tập trung vào việc xây dựng nền tảng chạy ổn định trên môi trường web, tương thích với các trình duyệt phổ biến và có thể triển khai nội bộ hoặc trên hosting thật để thử nghiệm.

Đề tài sử dụng phương pháp nghiên cứu kết hợp giữa nghiên cứu lý thuyết và ứng dụng thực tiễn nhằm đảm bảo tính khả thi và phù hợp với nhu cầu thực tế của sinh viên. Thực hiện tiến hành phân tích và khảo sát nhu cầu sử dụng thông qua việc phát phiếu khảo sát và tham khảo các nền tảng diễn đàn phổ biến như Reddit, Stack Overflow, cũng như các hệ thống quản lý học tập (LMS) có tích hợp diễn đàn thảo luận. Qua đó, xác định được các tính năng cần thiết và ưu tiên trong quá trình phát triển hệ thống.

## **CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG**

---

### **2.1 Javascript**

#### **2.1.1 Sơ lược về ngôn ngữ Javascript**

JavaScript, còn được viết tắt là JS, là một ngôn ngữ lập trình, chạy dưới dạng ngôn ngữ kịch bản phía máy khách trong môi trường hosting. Môi trường hosting phổ biến nhất là trình duyệt web, dùng ngôn ngữ này để tạo tính tương tác cho trang web.

JavaScript cho phép thực thi mã lệnh ở phía client, tức là ngay trên trình duyệt thay vì ở máy chủ từ xa (client-side).

Client-side JavaScript là mã JavaScript được thực thi trực tiếp trên trình duyệt của người dùng (Google Chrome, Microsoft Edge, Fire Fox), thường được sử dụng để tạo ra các tính năng tương tác và động trên trang web [7].

#### **2.1.2 Đặc điểm**

- Là ngôn ngữ kịch bản (scripting language) hướng đối tượng.
- Là ngôn ngữ nhập động (dynamic typing) dựa trên prototype.
- Là ngôn ngữ lightweight (dễ học và viết code nhanh).
- Là ngôn ngữ đa mô hình (multi-paradigm) và đơn luồng (single threaded) [7].

#### **2.1.3 Cách hoạt động**

##### **Đối với JS phía máy khách (client-side):**

- Tích hợp sẵn bên trong hầu hết các trình duyệt web phổ biến.
- Trình duyệt chuyển đổi trang web đó và tất cả các phần tử của nó thành dạng cấu trúc dữ liệu gọi là DOM
- Công cụ JavaScript của trình duyệt thực hiện chuyển mã JavaScript thành mã byte.
- Khi có sự kiện xảy ra, ví dụ như nhấp vào button, việc thực thi code block JavaScript liên kết sẽ được kích hoạt [7].

### **Đối với JS phía máy chủ (server-side):**

- Truy cập cơ sở dữ liệu
- Thực hiện các tính toán logic khác nhau
- Phản hồi các sự kiện khác nhau do hệ điều hành của máy chủ kích hoạt [7].

## **2.2 React framework**

### **2.2.1 Giới thiệu**

React (ReactJS) là một thư viện JavaScript mã nguồn mở, được dùng để xây dựng giao diện người dùng (frontend) cho web. React chỉ tập trung vào phần hiển thị giao diện (view), chứ không can thiệp vào cách sắp xếp logic nghiệp vụ hoặc cấu trúc ứng dụng.

React tập trung vào việc hiển thị giao diện người dùng và cho phép lập trình viên tự do quyết định cách sắp xếp logic nghiệp vụ.

Khác với các framework có kiến trúc cố định như Angular, React không ép buộc người dùng vào một mô hình cụ thể, khiến nó linh hoạt cho nhiều dự án khác nhau [4].

### **2.2.2 Đặc điểm nổi bật**

- Linh hoạt trong thiết kế kiến trúc

- Kiến trúc Component đơn giản và nhẹ: mỗi thành phần có thể được tái sử dụng, giúp ứng dụng dễ mở rộng và duy trì. Các component trong React rất nhẹ và có thể chỉ là các hàm đơn giản trả về JSX.

- Cộng đồng hỗ trợ lớn: React được duy trì và phát triển bởi Facebook, một trong những công ty công nghệ lớn nhất thế giới. Điều này đảm bảo rằng React sẽ nhận được sự hỗ trợ liên tục và các bản cập nhật mới, giúp lập trình viên yên tâm rằng React sẽ tiếp tục phát triển và phù hợp với các dự án dài hạn [4].

### **2.2.3 Các thành phần quan trọng**

- JSX (JavaScript XML) là một cú pháp mở rộng cho phép lập trình viên viết mã giống như HTML trong JavaScript. Trong các ngôn ngữ khác, lập trình viên thường phải viết code HTML và JavaScript riêng rẽ. Tuy nhiên, với JSX, React cho phép kết

hợp cả hai trong cùng một mã nguồn, giúp quản lý dễ dàng hơn, đặc biệt là trong các ứng dụng phức tạp.

- Virtual DOM (Document Object Model ảo) là một bản sao nhẹ hơn của DOM thật. DOM thật là cấu trúc cây chứa tất cả các thành phần HTML trong trang web. Khi người dùng tương tác với ứng dụng (ví dụ: nhập văn bản, nhấn nút), ứng dụng sẽ thay đổi nội dung và DOM thật phải được cập nhật.

- Liên kết dữ liệu một chiều: trong React, dữ liệu được quản lý và luân chuyển theo hướng từ component cha xuống component con. Các component cha có thể truyền dữ liệu cho component con thông qua props, nhưng các component con không thể trực tiếp thay đổi dữ liệu của component cha [4].

- Các bước cài đặt:

+ Cần phải có môi trường cài đặt, NodeJS: <https://nodejs.org/en>

+ Mở cmd (Command Prompt) và gõ:

**npx create-react-app frontend** (Tạo một thư mục frontend React)

**cd frontend** (Di chuyển vào thư mục frontend)

**npm start** (Chạy ứng dụng React)



**Hình 2.1 Thư mục Frontend cơ bản**

## 2.3 NodeJs

### 2.3.1 Giới thiệu

Nodejs là một môi trường runtime chạy JavaScript đa nền tảng, open-source (mã nguồn mở), được sử dụng rộng rãi bởi lập trình viên cho nhiều dự án. Nodejs cung cấp nhiều thư viện khác nhau, giúp đơn giản hóa việc lập trình.

Nodejs là một môi trường JavaScript runtime, đa nền tảng: Windows, Linux, Unix, Mac OS X,...

Một trong những tính năng chính của Nodejs là, cho phép xử lý đồng thời nhiều yêu cầu, mà không chặn các yêu cầu khác [6].

### 2.3.2 Đặc điểm vượt trội

Lập trình bất đồng bộ: Server không cần chờ dữ liệu mà API trả về, vẫn có thể xử lý các yêu cầu khác.

Tốc độ: Được xây dựng dựa trên engine JavaScript V8, Nodejs thực thi mã nhanh.

NPM: Với hơn 50,000 package khác nhau, lập trình viên dễ dàng tìm kiếm, lựa chọn và tích hợp tính năng cần thiết vào dự án của mình.

Không có buffering: Nodejs xử lý các tệp âm thanh và video hiệu quả bằng cách xuất dữ liệu theo từng phần (chunk) thay vì buffer dữ liệu.

Đơn luồng: Nodejs sử dụng mô hình đơn luồng với vòng lặp sự kiện (Event loop), giúp xử lý số lượng request nhiều hơn so với các server truyền thống như Apache HTTP Server [6].

### 2.3.3 Cơ chế hoạt động

- Nhiều client sẽ gửi request đến server, sẽ được đưa vào Event Queue.
- Event Loop sẽ lấy lần lượt các request từ hàng đợi và xử lý chúng.
- Nếu request cần thực hiện các tác vụ non-blocking I/O, Event Loop sẽ tự động xử lý rồi trả về response.
- Đối với những tác vụ blocking I/O, Event Loop sẽ đưa tasks qua Thread Pool (qua thư viện Libuv).

- Khi thread đã thực hiện xong, kết quả sẽ được Nodejs đẩy callback của tasks đó về Event Queue để xử lý.

- Cuối cùng, response sẽ được phản hồi lại cho các client [6].



**Hình 2.2 Các ứng dụng phổ biến của NodeJs [6]**

- Các bước cài đặt

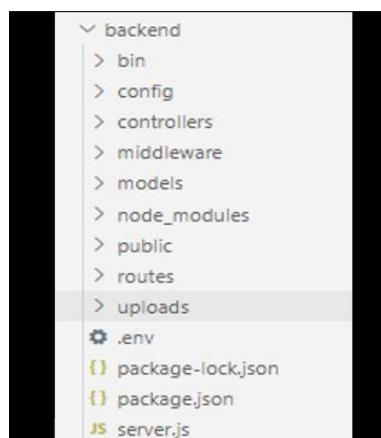
+ Cài đặt Node.js tại đường link: <https://nodejs.org/en>

+ Mở cmd (Command Prompt) và gõ:

**mkdir backend** (Tạo thư mục cho backend)

**cd backend** (Di chuyển vào thư mục:)

**npm init -y** (Khởi tạo một dự án Node.js)



**Hình 2.3 Thư mục backend cơ bản**

## 2.4 Express.js framework

### 2.4.1 Giới thiệu

Express.js là một framework đơn giản được xây dựng trên nền tảng Node.js, ra đời với mục đích làm cho việc phát triển các ứng dụng web và API trở nên đơn giản, hiệu quả và dễ bảo trì hơn. Bằng cách cung cấp các công cụ tiện lợi để xử lý yêu cầu HTTP, quản lý định tuyến (routing), và thêm các chức năng khác vào ứng dụng thông qua middleware.

Với API RESTful, lập trình viên có thể dễ dàng hơn và quản lý dữ liệu hiệu quả mà không cần viết quá nhiều code phức tạp. Express.js giúp lập tập trung vào logic ứng dụng thay vì xử lý chi tiết các yêu cầu HTTP [5].

### 2.4.2 Các tính năng chính

- Templating engines cho phép lập trình viên tạo các mẫu (templates) chứa các placeholder (chỗ trống) để sau đó có thể thay thế bằng dữ liệu thực tế khi trang được hiển thị, giúp tạo ra các trang HTML linh hoạt và dễ dàng hơn, đặc biệt là khi dữ liệu thay đổi thường xuyên.
  - Middleware `express.static()` giúp việc phục vụ các tệp tĩnh trong dự án trở nên đơn giản. Lập trình viên chỉ cần chỉ định thư mục chứa các tệp tĩnh (như hình ảnh, CSS, JavaScript), và Express.js sẽ tự động xử lý việc cung cấp chúng cho người dùng khi họ yêu cầu.
  - Tệp tĩnh (static files) là những tệp không cần xử lý phức tạp từ máy chủ, mà chỉ được gửi trực tiếp đến người dùng, như hình ảnh, tệp CSS để tạo kiểu, hoặc tệp JavaScript để chạy trên trình duyệt.
  - Routing (Định tuyến) trong Express.js là cách lập trình viên thiết lập các quy tắc để xử lý các yêu cầu HTTP từ người dùng đến đúng nơi trong ứng dụng của lập trình viên. Mỗi URL sẽ được gán với một route (định tuyến), và mỗi route có thể xử lý các loại yêu cầu khác nhau như GET, POST, PUT, DELETE,...
  - Middleware là những hàm đặc biệt trong Express.js, được thực thi khi ứng dụng của lập trình viên nhận yêu cầu từ người dùng và trước khi gửi phản hồi về cho họ. Middleware có thể thực hiện nhiều nhiệm vụ như:

- + Ghi lại thông tin về yêu cầu.
- + Xác thực người dùng.
- + Xử lý dữ liệu.
- + Trả về lỗi khi có vấn đề xảy ra [5].

Các bước cài đặt:

- + Sau khi khởi tạo xong thư mục backend với NodeJS ➔ tải Express (framework): **npm install express cors**

Kết nối frontend với backend:

- + Mở tệp frontend/package.json và thêm mục proxy như sau:

```
{
  "name": "frontend",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    // Các phụ thuộc khác
  },
  "proxy": "http://localhost:5000" //Dòng thêm vào
}
```

➔ Thực hiện các yêu cầu từ React (frontend) đến một URL tương đối (ví dụ: /api/), React sẽ tự động chuyển hướng yêu cầu đến server Node.js đang chạy trên <http://localhost:5000>

## 2.5 Mô hình MVC

### 2.5.1 Tổng quan về mô hình MVC

Mô hình MVC, viết tắt của Model-View-Controller, là một kiến trúc phần mềm phổ biến được sử dụng rộng rãi trong phát triển ứng dụng. Được thiết kế để tách biệt logic xử lý dữ liệu (Model), hiển thị giao diện người dùng (View), và điều phối cả hai (Controller), mô hình này mang lại nhiều lợi ích trong việc quản lý mã nguồn và cải thiện khả năng bảo trì của hệ thống.

**Model (M - Model):** Đây là thành phần chịu trách nhiệm cho xử lý dữ liệu và logic kinh doanh của ứng dụng. Model là nơi lưu trữ thông tin, thực hiện các thao tác cập nhật

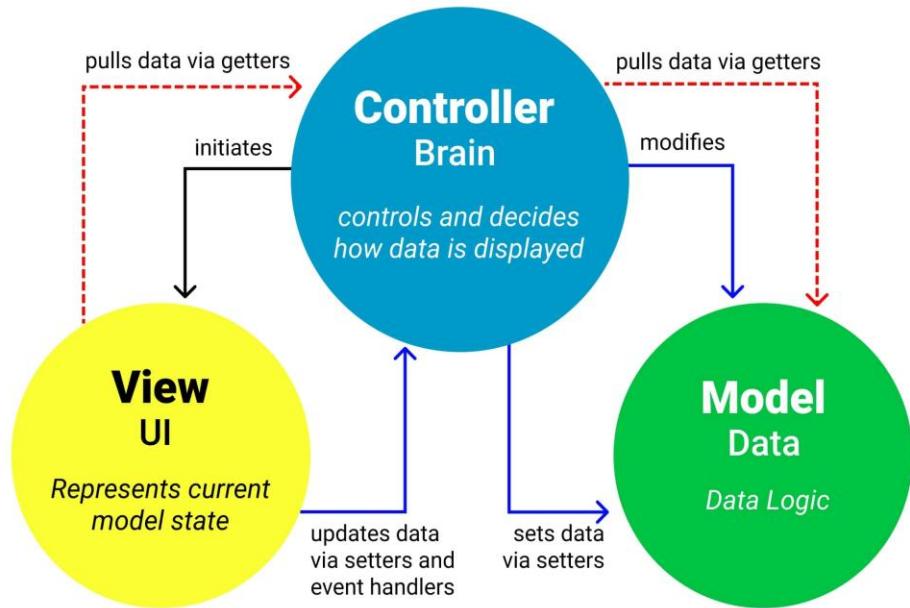
và truy vấn dữ liệu, mà không quan tâm đến cách dữ liệu được hiển thị hoặc tương tác với người dùng.

**View (V - View):** View là thành phần hiển thị giao diện người dùng và đảm nhận trách nhiệm hiển thị thông tin từ Model. View không có logic kinh doanh và chỉ chịu trách nhiệm về việc hiển thị dữ liệu một cách đẹp mắt và dễ hiểu cho người dùng.

**Controller (C - Controller):** Controller là thành phần điều phối và xử lý sự kiện từ người dùng. Nó nhận lệnh từ người dùng thông qua View, sau đó cập nhật Model dựa trên những thay đổi này và điều hướng hiển thị trở lại View. Controller giữ vai trò quan trọng trong việc duy trì sự đồng bộ giữa Model và View.

Mô hình MVC giúp tách biệt logic ứng dụng thành các phần nhỏ, dễ quản lý và bảo trì. Nó cũng thúc đẩy tái sử dụng mã nguồn, khiến cho quá trình phát triển và bảo trì ứng dụng trở nên hiệu quả hơn [9].

## MVC Architecture Pattern



Hình 2.4 MVC [9]

### **2.5.2 Các thành phần**

**Model:** Đại diện cho dữ liệu và logic kinh doanh của ứng dụng. Nó là nơi lưu trữ thông tin, thực hiện các thao tác cập nhật và truy vấn dữ liệu. Model không quan tâm đến cách dữ liệu được hiển thị hoặc tương tác với người dùng.

#### **Trách nhiệm:**

- Lưu trữ và quản lý dữ liệu.
- Thực hiện logic kinh doanh và xử lý sự kiện.
- Cung cấp API cho các thành phần khác để truy xuất và cập nhật dữ liệu.

**View:** View là thành phần chịu trách nhiệm hiển thị thông tin từ Model cho người dùng. Nó không có logic kinh doanh và chỉ tập trung vào việc hiển thị dữ liệu một cách đẹp mắt và dễ hiểu.

#### **Trách nhiệm:**

- Hiển thị dữ liệu cho người dùng.
- Tương tác với người dùng thông qua giao diện người dùng.
- Thông báo về các sự kiện tương tác đến Controller.

**Controller:** Controller là thành phần điều phối và xử lý sự kiện từ người dùng. Nó nhận lệnh từ người dùng thông qua View, sau đó cập nhật Model dựa trên những thay đổi này và điều hướng hiển thị trở lại View.

#### **Trách nhiệm:**

- Nhận các sự kiện từ View.
- Gửi các yêu cầu cập nhật hoặc truy vấn dữ liệu đến Model.
- Cập nhật View dựa trên sự thay đổi trong Model.
- Giữ sự đồng bộ giữa Model và View [9].

### **2.5.3 Ưu điểm và Nhược điểm**

#### **Ưu điểm:**

- + Tách biệt logic: MVC giúp tách biệt logic ứng dụng thành các phần riêng biệt (Model, View, và Controller), làm cho mã nguồn dễ hiểu và dễ bảo trì.
- + Tái sử dụng mã nguồn: Các thành phần trong MVC có thể được tái sử dụng một cách linh hoạt. Model và View có thể được sử dụng lại trong nhiều phần của ứng dụng hoặc trong các ứng dụng khác nhau.
- + Phân chia công việc: Mô hình này giúp phân chia công việc giữa các thành viên trong nhóm phát triển, với người làm Model tập trung vào logic xử lý dữ liệu, người làm View chịu trách nhiệm về giao diện người dùng, và người làm Controller quản lý sự tương tác.
- + Dễ mở rộng: Do sự tách biệt giữa các thành phần, việc mở rộng và thay đổi ứng dụng trở nên dễ dàng hơn. Một thành phần có thể được thay thế hoặc mở rộng mà không làm ảnh hưởng đến các thành phần khác [9].

### **Nhược điểm:**

- + Phức tạp cho ứng dụng nhỏ: Đối với các ứng dụng nhỏ hoặc đơn giản, việc triển khai mô hình MVC có thể quá phức tạp và làm tăng độ phức tạp của mã nguồn.
- + Tăng độ phức tạp của mã nguồn: Trong một số trường hợp, việc chia nhỏ ứng dụng thành nhiều thành phần có thể làm tăng độ phức tạp của mã nguồn và làm cho mã trở nên khó hiểu [9].

## **2.6 Material UI (MUI)**

### **2.6.1 Khái quát về Material UI (MUI)**

Material UI là một thư viện UI mã nguồn mở bao gồm các React component, được tích hợp thêm cả Google's Material Design [8].

### **2.6.2 Các ưu điểm**

Tiết kiệm thời gian: lập trình viên không cần phải tự xây dựng những component giao diện cơ bản từ đầu. Thư viện MUI đã làm điều đó cho người dùng, giúp lập trình viên chỉ cần tập trung vào nghiệp vụ, logic ứng dụng.

Giao diện được thống nhất: sử dụng MUI giúp giao diện của lập trình viên tuân thủ theo phong cách đồng nhất, không bị lệch design giữa các trang trong ứng dụng.

Dễ tùy biến: chỉ cần chỉnh lại màu sắc, kích thước chữ, hay muốn giao diện tối (dark mode). MUI cũng sẽ hỗ trợ toàn diện theme.

Cộng đồng lớn, tài liệu chi tiết: có rất nhiều ví dụ, hướng dẫn, và câu trả lời sẵn có, giúp lập trình viên nhanh chóng khắc phục những khó khăn trong quá trình phát triển [8].

## 2.7 Hệ quản trị cơ sở dữ liệu MongoDB

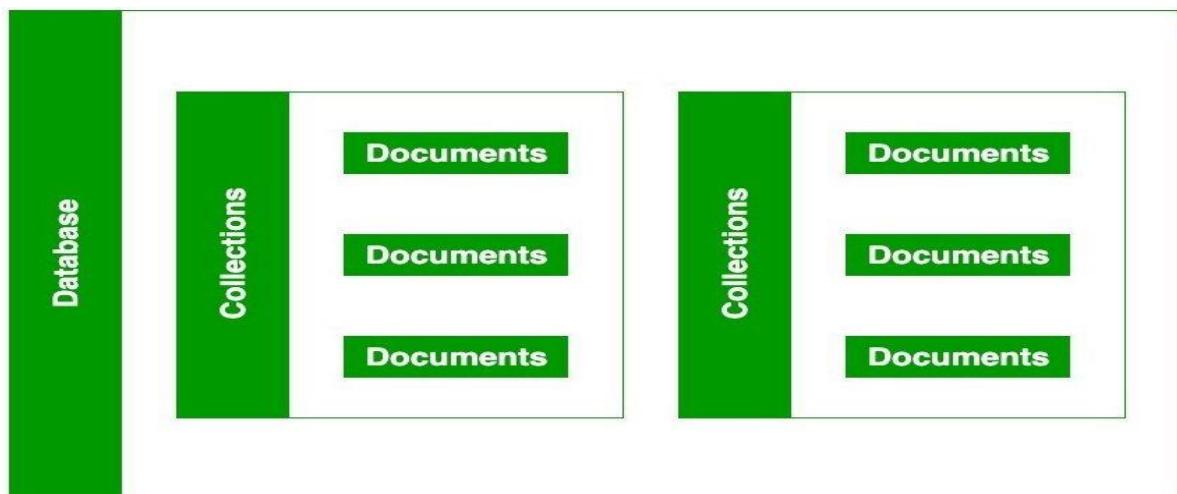
### 2.7.1 Giới thiệu

MongoDB là một trong những cơ sở dữ liệu phi quan hệ (NoSQL) phổ biến nhất hiện nay, được sử dụng rộng rãi trong nhiều lĩnh vực. MongoDB cung cấp các tính năng hữu ích giúp cho việc phân tích, truy vấn và cập nhật dữ liệu trở nên dễ dàng và nhanh chóng.

MongoDB hỗ trợ tất cả các ngôn ngữ phổ biến như C, C++, C# và .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid [11].

### 2.7.2 Cách hoạt động

MongoDB hoạt động dưới dạng một hệ thống cơ sở dữ liệu phi quan hệ, lưu trữ dữ liệu dưới dạng tài liệu (document) JSON. Dữ liệu được lưu trữ trong collections và documents.



Hình 2.5 MongoDB [11]

Cơ sở dữ liệu MongoDB lưu trữ tài liệu trong các collections, tương tự như bảng trong cơ sở dữ liệu quan hệ. Mỗi collection có thể chứa nhiều tài liệu (documents) có cấu trúc dữ liệu tùy ý, có thể chứa nhiều fields dữ liệu, mỗi field được định danh bằng tên và có giá trị tương ứng.

Các tài liệu (documents) được tạo bằng cách sử dụng các field. Các field là các key-value pair trong tài liệu, nó giống như các cột trong cơ sở dữ liệu quan hệ.

Giá trị của fields có thể thuộc bất kỳ loại dữ liệu BSON nào như double, string, boolean,...

MongoDB hỗ trợ việc tạo index cho các field dữ liệu trong collection, giúp tăng tốc độ truy vấn. Chúng còn hỗ trợ sao chép dữ liệu giữa các node trong một cluster giúp đảm bảo tính khả dụng và độ tin cậy của hệ thống.

MongoDB phân tán dữ liệu trên nhiều node, giúp tăng khả năng mở rộng của hệ thống, đồng thời chúng còn hỗ trợ tính toán phân tán bằng cách sử dụng MapReduce giúp xử lý dữ liệu lớn một cách hiệu quả [11].

### 2.7.3 Ưu điểm và Nhược điểm

#### Ưu điểm:

- **Tính linh hoạt:** MongoDB là một hệ thống cơ sở dữ liệu phi quan hệ, nó cung cấp khả năng lưu trữ dữ liệu bất cứ khi nào, bất cứ nơi đâu, không cần phải tuân thủ một mô hình quan hệ cụ thể.

- **Khả năng mở rộng, truy xuất nhanh:** MongoDB có khả năng mở rộng dễ dàng, nhờ tính năng sharding cho phép phân chia dữ liệu thành nhiều phần và lưu trữ trên nhiều máy chủ, truy vấn dữ liệu trong thời gian ngắn.

- **Tính khả dụng cao:** MongoDB cung cấp tính năng sao lưu và phục hồi dữ liệu, giúp người dùng bảo vệ dữ liệu của mình khỏi những rủi ro.

- **Dễ sử dụng:** MongoDB cung cấp các công cụ quản lý dữ liệu trực quan và dễ sử dụng, giúp người dùng tối ưu hóa hiệu suất và quản lý cơ sở dữ liệu một cách dễ dàng.

#### Nhược điểm:

- Cần sử dụng bộ nhớ cao để lưu trữ dữ liệu (data storage).
- Không được phép lưu trữ hơn 16MB data trong tài liệu.
- Data nesting trong BSON cũng bị hạn chế, lập trình viên không được phép nest data quá 100 cấp độ [11].

➔ Cài đặt tại đường link: <https://www.mongodb.com/try/download/community>

## 2.8 Chatbot AI

### 2.8.1 Tổng quan

Chatbot AI là một chương trình kết hợp với trí tuệ nhân tạo (AI) và ngôn ngữ lập trình NLP để tương tác với con người. Công cụ này thay thế cho nhân viên để trả lời những gì khách hàng thắc mắc [12].

### 2.8.2 Lợi ích

- Giảm chi phí.
- Trả lời khách hàng nhanh chóng.
- Tăng tính chuyên nghiệp [12].

### 2.8.3 Các chatbot AI hiện nay

- Chatbot AI bán hàng
- Chatbot AI chăm sóc khách hàng
- Chatbot trò chuyện [12].

## 2.9 WebSocket và công nghệ Realtime

WebSocket - một giao thức mạng giúp tạo kết nối hai chiều, liên tục, full-duplex giữa client và server, khác với mô hình HTTP truyền thống chỉ theo kiểu request-response. WebSocket giúp giảm độ trễ và tăng hiệu quả giao tiếp trong các ứng dụng yêu cầu cập nhật dữ liệu thời gian thực như chat, game, trading hay thông báo trực tiếp [13].

Frontend sử dụng đối tượng WebSocket để kết nối và gửi/nhận tin nhắn từ server, xây dựng giao diện chat cơ bản. Cách làm này giúp người dùng hiểu cơ chế hoạt động của WebSocket và cách triển khai app chat realtime nhanh, hiệu quả [13].

## CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

---

### 3.1 Mô tả bài toán

Hiện nay, sinh viên tại nhiều trường đại học, đặc biệt là ở các khu vực đang từng bước hiện đại hóa như Trà Vinh, vẫn chủ yếu trao đổi thông tin qua các nền tảng mạng xã hội như Facebook, Zalo,... Những công cụ này tuy phổ biến nhưng thiếu tính hệ thống, khó kiểm soát nội dung và không được thiết kế chuyên biệt cho môi trường học tập.

Vấn đề đặt ra là cần xây dựng một hệ thống diễn đàn trực tuyến có cấu trúc rõ ràng, nội dung được phân loại theo từng chuyên ngành, cho phép sinh viên đăng ký tài khoản, đăng bài, bình luận, tìm kiếm tài liệu và tương tác học thuật một cách thuận tiện, an toàn và hiệu quả. Hệ thống cũng cần có phân quyền quản trị để đảm bảo kiểm soát nội dung, xử lý vi phạm và duy trì môi trường học tập lành mạnh.

Giải quyết bài toán này sẽ góp phần tạo ra một không gian học tập số chuyên nghiệp cho sinh viên, hỗ trợ giao tiếp học thuật, chia sẻ kinh nghiệm học tập và thúc đẩy quá trình chuyển đổi số trong giáo dục đại học.

#### Yêu cầu về chức năng:

Đối với sinh viên (người dùng):

- Đăng ký và đăng nhập:

- Cho phép sinh viên tạo tài khoản cá nhân và đăng nhập vào hệ thống.
- Hỗ trợ chức năng khôi phục mật khẩu trong trường hợp quên.

- Tạo bài viết chia sẻ:

- Cho phép người dùng đăng bài viết liên quan đến học tập, kinh nghiệm, đời sống sinh viên.
- Bài viết có thể đính kèm hình ảnh hoặc video.

- Bình luận và phản hồi bài viết:

- Người dùng có thể bình luận bên dưới các bài viết, tương tác với tác giả và người khác.

- Đánh giá bài viết:

- Cung cấp tính năng thích (like), chia sẻ, hoặc đánh giá bài viết để tăng tính tương tác.

- Phân loại bài viết:

- Khi đăng bài, người dùng có thể chọn chủ đề (ví dụ: Học tập, Chia sẻ kinh nghiệm, Học bổng...).

- Trò chuyện với chatbot hỗ trợ:

- Hệ thống tích hợp chatbot để hướng dẫn người dùng sử dụng các chức năng cơ bản.

Đối với quản trị viên:

- Kiểm duyệt bài viết và bình luận:

- Xem, phê duyệt, chỉnh sửa hoặc xóa bài viết, bình luận vi phạm.

- Quản lý người dùng:

- Xem thông tin người dùng, cảnh báo hoặc khóa tài khoản vi phạm.

- Quản lý chuyên mục:

- Thêm, sửa, xóa chủ đề phù hợp với nhu cầu học tập và xu hướng sinh viên.

- Thống kê và phân tích hoạt động:

- Thống kê số lượng người dùng, bài viết, chủ đề phổ biến.

- Phân tích hành vi người dùng như thời gian truy cập, từ khóa tìm kiếm.

- Quản lý chatbot:

- Cập nhật, huấn luyện chatbot để cải thiện chất lượng phản hồi.

- Bảo trì và cập nhật hệ thống:

- Theo dõi lỗi, cập nhật tính năng mới, và lỗi bảo mật khi cần thiết.

## **Yêu cầu phi chức năng:**

Tính dễ sử dụng:

- Giao diện thân thiện, dễ sử dụng cho cả sinh viên và quản trị viên.

Tính bảo mật:

- Bảo vệ thông tin tài khoản người dùng, kiểm soát truy cập và ngăn chặn hành vi xâm nhập.

Hiệu năng hệ thống:

- Hệ thống hoạt động ổn định với số lượng lớn người dùng truy cập cùng lúc.

Khả năng mở rộng:

- Hệ thống dễ nâng cấp thêm các chủ đề mới hoặc tích hợp tính năng mở rộng như chat trực tiếp, thông báo đẩy.

Khả năng tương thích:

- Hệ thống hoạt động tốt trên nhiều trình duyệt và thiết bị (máy tính, điện thoại, máy tính bảng).

Khả năng bảo trì:

- Mã nguồn được tổ chức rõ ràng, dễ sửa lỗi và cập nhật về sau.

## 3.2 Xây dựng websites

### 3.2.1 Cài đặt môi trường lập trình

- Frontend: ReactJs + Vite:
  - + Bước 1: cài đặt Node.js từ <https://nodejs.org> (nên chọn bản LTS).
  - + Bước 2: Mở terminal và chạy lệnh sau để tạo dự án React với Vite:  
**npm create vite@latest dien\_dan\_TVU --template react**
  - + Bước 3: cd frontend
  - + Bước 4: npm install
  - + Bước 5: npm run dev
- Backend: ExpressJs + Nodemon

+ Bước 1: Tạo thư mục backend và khởi tạo dự án Node.js:

```
mkdir backend  
cd backend  
npm init -y
```

+ Bước 2:

```
npm install express
```

+ Bước 3: Cài đặt nodemon để hỗ trợ tự động khởi động lại server khi có thay đổi mã:

```
npm install --save-dev nodemon
```

+ Bước 4: Chỉnh sửa file package.json, thêm dòng "start" và "dev" vào mục scripts:

```

"scripts": {
    "start": "node index.js",
    "dev": "nodemon index.js"
}

```

+ Bước 5: Tạo file index.js với nội dung server cơ bản:

```

const express = require('express');
const app = express();
const PORT = 3000;

app.get('/', (req, res) => {
    res.send('Hello from Express');
});

app.listen(PORT, () => {
    console.log(`Server is running on port ${PORT}`);
});

```

+ Bước 6: Khởi chạy server bằng nodemon:

**npm run dev**

➔ Khi dùng nodemon không cần restart server mỗi lần sửa file, nodemon sẽ tự làm việc đó.

- Hệ quản trị cơ sở dữ liệu MongoDB:

+ **Tùy chọn 1:** Dùng MongoDB Atlas (cloud, miễn phí):

Truy cập: <https://www.mongodb.com/cloud/atlas>

Tạo tài khoản, tạo Cluster, tạo Database và lấy connection string (URI).

+ **Tùy chọn 2:** Cài MongoDB Local:

Tải về từ <https://www.mongodb.com/try/download/community>

Cài đặt theo hướng dẫn, sau đó chạy MongoDB service trên máy.

+ Cài thư viện mongoose để kết nối MongoDB trong Node.js:

**npm install mongoose**

+ Kết nối cơ bản trong backend index.js:

```

const mongoose = require('mongoose');

mongoose.connect('mongodb://localhost:27017/ten_database')
    .then(() => console.log('Connected to MongoDB'))

```

```
.catch(err => console.error('MongoDB connection error:', err));
```

### 3.2.2 Cơ sở dữ liệu

Gồm 17 Collection:

#### User Collection

```
{  
    _id: ObjectId,  
    fullName: String,  
    email: String (unique),  
    password: String,  
    username: String (unique, sparse),  
    phone: String,  
    address: String,  
    avatarUrl: String (default: ""),  
    role: String (enum: ['user', 'admin'], default: 'user'),  
  
    // Google OAuth fields  
    googleId: String (unique, sparse),  
    authProvider: String (enum: ['local', 'google'], default: 'local'),  
    isEmailVerified: Boolean (default: false),  
  
    // Account management  
    status: String (enum: ['active', 'suspended', 'banned'], default: 'active'),  
    warnings: [{  
        message: String,  
        reason: String,  
        createdBy: ObjectId (ref: 'User'),  
        createdAt: Date (default: Date.now)  
    }],  
    suspensionInfo: {  
        reason: String,  
        suspendedBy: ObjectId (ref: 'User'),  
        suspendedAt: Date,  
        suspendedUntil: Date  
    },  
    banInfo: {  
        reason: String,  
        bannedBy: ObjectId (ref: 'User'),  
        bannedAt: Date,  
        isPermanent: Boolean (default: false)  
    },  
    lastLogin: Date,  
    loginAttempts: Number (default: 0),
```

```

    createdAt: Date (default: Date.now),
    updatedAt: Date (default: Date.now)
}

```

➔ Chức năng: Quản lý thông tin người dùng, authentication, phân quyền

Tính năng chính:

- Thông tin cá nhân (fullName, email, phone, address)
- Authentication (password, Google OAuth)
- Phân quyền (user, admin)
- Quản lý trạng thái (active, suspended, banned)
- Lịch sử cảnh báo và khóa tài khoản

## Posts Collection

```

{
  _id: ObjectId,
  authorId: ObjectId (ref: 'User', required),
  title: String (required, trim),
  content: String (required),
  topicId: ObjectId (ref: 'Topics', required),
  tags: [String],
  views: Number (default: 0),
  commentCount: Number (default: 0),
  likeCount: Number (default: 0),
  ratingCount: Number (default: 0),
  type: String (enum: ['discussion', 'question', 'event', 'job', 'news', 'other'], default: 'discussion'),
  status: String (enum: ['draft', 'pending', 'published', 'archived', 'flagged', 'deleted'], default: 'pending'),
}

createdAt: Date (auto),
updatedAt: Date (auto)
}

```

➔ Chức năng: Quản lý bài viết trong diễn đàn

Tính năng chính:

- Nội dung bài viết (title, content, tags)
- Phân loại (discussion, question, event, job, news)
- Thống kê (views, commentCount, likeCount, ratingCount)
- Trạng thái (draft, pending, published, archived, flagged, deleted)

## Images Collection

```
{  
  _id: ObjectId,  
  refType: String (enum: ['post', 'topic'], required),  
  refId: ObjectId (required, refPath: 'refType'),  
  url: String (required),  
  caption: String (default: ""),  
  isMain: Boolean (default: false),  
  likeCount: Number (default: 0),  
  commentCount: Number (default: 0),  
  createdAt: Date (default: Date.now)  
}
```

➔ Chức năng: Quản lý hình ảnh trong hệ thống

Tính năng chính:

- Lưu trữ đa đối tượng (post, topic)
- Thông tin ảnh (url, caption, isMain)
- Thống kê (likeCount, commentCount)

## Topics Collection

```
{  
  id: ObjectId,  
  name: String (required, unique, trim),  
  description: String (required, trim),  
  category: String (enum: ['Học tập', 'Nghiên cứu', 'Thực tập', 'Việc làm', 'Hoạt động sinh viên', 'Công nghệ', 'Kỹ năng mềm', 'Trao đổi học thuật', 'Thông báo', 'Khác'], default: 'Học tập'),  
  priority: Number (default: 0, min: 0, max: 10),  
  status: String (enum: ['active', 'inactive', 'archived'], default: 'active'),  
  color: String (default: '#1976d2', match: /^#[A-Fa-f0-9]{6}#[A-Fa-f0-9]{3}\$/),  
  icon: String (default: 'topic'),  
  imageUrl: String (default: ""),  
  tags: [String (trim)],  
  
  // Statistics  
  postCount: Number (default: 0),  
  viewCount: Number (default: 0),  
  
  // Management  
  createdBy: ObjectId (ref: 'User', required),  
  updatedBy: ObjectId (ref: 'User'),  
  
  // Display settings
```

```

    isVisible: Boolean (default: true),
    allowPosts: Boolean (default: true),
    requireApproval: Boolean (default: false),

    createdAt: Date (auto),
    updatedAt: Date (auto)
}

```

➔ Chức năng: Quản lý chủ đề bài viết

Tính năng chính:

- Thông tin chủ đề (name, description, category)
- Hiển thị (color, icon, imageUrl, priority)
- Thống kê (postCount, viewCount)
- Cài đặt (isVisible, allowPosts, requireApproval)

### Likes Collection

```

{
  _id: ObjectId,
  userId: ObjectId (ref: 'User', required),
  postId: ObjectId (ref: 'Post'),
  commentId: ObjectId (ref: 'Comment'),
  targetType: String (enum: ['post', 'comment', 'image'], required),
  createdAt: Date (default: Date.now)
}

```

➔ Chức năng: Quản lý lượt thích cho bài viết và hình ảnh

Tính năng chính:

- Thích đa đối tượng (post, comment, image)
- Liên kết với user và target object

### Comments Collection

```

{
  _id: ObjectId,
  postId: ObjectId (ref: 'Post', required),
  authorId: ObjectId (ref: 'User', required),
  content: String (required),
  parentCommentId: ObjectId (ref: 'Comment', default: null),
  imageId: ObjectId (ref: 'Image', default: null),
  level: Number (default: 0),
  rootCommentId: ObjectId (ref: 'Comment', default: null),
}

```

```
    replyCount: Number (default: 0),  
    likeCount: Number (default: 0),  
  
    createdAt: Date (auto),  
    updatedAt: Date (auto)  
}
```

➔ Chức năng: Quản lý bình luận và phản hồi

Tính năng chính:

- Bình luận đa cấp (parentCommentId, level, rootCommentId)
- Thống kê (replyCount, likeCount)
- Hỗ trợ hình ảnh (imageId)

### **CommentLikes Collection**

```
{  
  _id: ObjectId,  
  commentId: ObjectId (ref: 'Comments', required),  
  userId: ObjectId (ref: 'User', required),  
  createdAt: Date (default: Date.now)  
}
```

➔ Chức năng: Quản lý lượt thích riêng cho bình luận

Tính năng chính:

- Thích bình luận cụ thể
- Đảm bảo mỗi user chỉ thích 1 lần/comment

### **Ratings Collection**

```
{  
  _id: ObjectId,  
  postId: ObjectId (ref: 'Posts', required),  
  userId: ObjectId (ref: 'User', required),  
  rating: Number (min: 1, max: 5, required),  
  createdAt: Date (default: Date.now)  
}
```

➔ Chức năng: Quản lý đánh giá bài viết

Tính năng chính:

- Đánh giá 1-5 sao
- Liên kết với user và post

## **Notifications Collection**

```
{  
  _id: ObjectId,  
  userId: ObjectId (ref: 'User', required),  
  message: String (required),  
  isRead: Boolean (default: false),  
  createdAt: Date (default: Date.now)  
}
```

➔ Chức năng: Quản lý thông báo cho người dùng

Tính năng chính:

- Thông báo cá nhân
- Trạng thái đọc/chưa đọc

## **ChatbotIntent Collection**

```
{  
  _id: ObjectId,  
  name: String (required, unique),  
  displayName: String (required),  
  description: String,  
  category: String (enum: ['greeting', 'information', 'support', 'navigation',  
  'academic', 'general'], default: 'general'),  
  
  trainingPhrases: [  
    text: String (required),  
    entities: [  
      entity: String,  
      value: String,  
      start: Number,  
      end: Number  
    ]  
  ],  
  
  responses: [  
    type: String (enum: ['text', 'card', 'quick_replies', 'custom'], default: 'text'),  
    content: String,  
    richContent: Mixed,  
    conditions: Mixed  
  ],  
  
  parameters: [  
    name: String (required),
```

```
entityType: String (required),
required: Boolean (default: false),
prompts: [String],
defaultValue: String
}],
inputContexts: [String],
outputContexts: [{
  name: String,
  lifespanCount: Number (default: 5),
  parameters: Mixed
}],
events: [String],
webhook: {
  enabled: Boolean (default: false),
  url: String,
  headers: Mixed
},
status: String (enum: ['active', 'inactive', 'draft'], default: 'active'),
stats: {
  triggerCount: Number (default: 0),
  successCount: Number (default: 0),
  lastTriggered: Date,
  avgConfidence: Number (default: 0)
},
dialogflow: {
  intentId: String,
  projectId: String,
  lastSynced: Date,
  syncStatus: String (enum: ['synced', 'pending', 'error'], default: 'pending'),
  syncError: String
},
createdBy: ObjectId (ref: 'User', required),
updatedBy: ObjectId (ref: 'User'),
tags: [String],
createdAt: Date (auto),
updatedAt: Date (auto)
}
```

➔ Chức năng: Quản lý intent cho chatbot AI

Tính năng chính:

- Training phrases và responses
- Tích hợp Dialogflow
- Thống kê trigger và success rate
- Parameters và contexts

### **ChatbotConversation Collection**

```
{  
    _id: ObjectId,  
    sessionId: String (required, index),  
    userId: ObjectId (ref: 'User', default: null),  
  
    sessionInfo: {  
        ipAddress: String,  
        userAgent: String,  
        deviceType: String (enum: ['desktop', 'mobile', 'tablet', 'unknown'], default: 'unknown'),  
        platform: String (default: 'web'),  
        language: String (default: 'vi')  
    },  
  
    messages: [{  
        type: String (enum: ['user', 'bot'], required),  
        text: String (required),  
        detectedIntent: {  
            name: String,  
            displayName: String,  
            confidence: Number,  
            parameters: Mixed  
        },  
        botResponse: {  
            responseType: String (enum: ['text', 'card', 'quick_replies', 'custom']),  
            fulfillmentText: String,  
            richResponses: Mixed,  
            webhookResponse: Mixed  
        },  
        contexts: [Mixed],  
        timestamp: Date (default: Date.now),  
        responseTime: Number,  
        hasError: Boolean (default: false),  
    }],  
}
```

```

error: {
  code: String,
  message: String,
  details: Mixed
},
],
status: String (enum: ['active', 'ended', 'abandoned'], default: 'active'),
startedAt: Date (default: Date.now),
endedAt: Date,
duration: Number,
feedback: {
  rating: Number (min: 1, max: 5),
  comment: String,
  feedbackAt: Date,
  issues: [String (enum: ['wrong_intent', 'poor_response', 'missing_info', 'technical_error', 'other'])]
},
stats: {
  totalMessages: Number (default: 0),
  userMessages: Number (default: 0),
  botMessages: Number (default: 0),
  successfulIntents: Number (default: 0),
  failedIntents: Number (default: 0),
  avgConfidence: Number (default: 0),
  avgResponseTime: Number (default: 0)
},
tags: [String],
adminNotes: [
  note: String,
  addedBy: ObjectId (ref: 'User'),
  addedAt: Date (default: Date.now)
],
needsReview: Boolean (default: false),
reviewedBy: ObjectId (ref: 'User'),
reviewedAt: Date,
createdAt: Date (auto),
updatedAt: Date (auto)
}

```

➔ Chức năng: Lưu trữ lịch sử hội thoại với chatbot

Tính năng chính:

- Session management
- Message history (user + bot)
- Intent detection results
- Feedback và rating
- Thống kê conversation

### **ChatbotFAQ Collection**

```
{  
  _id: ObjectId,  
  question: String (required),  
  answer: String (required),  
  tags: [String]  
}
```

➔ Chức năng: Quản lý câu hỏi thường gặp cho chatbot

Tính năng chính:

- Cặp question-answer
- Tags để phân loại

### **SearchLog Collection**

```
{  
  _id: ObjectId,  
  userId: ObjectId (ref: 'User', default: null),  
  
  query: String (required, trim),  
  normalizedQuery: String (required),  
  searchType: String (enum: ['posts', 'topics', 'users', 'global']), default: 'global'),  
  
  results: {  
    count: Number (default: 0),  
    hasResults: Boolean (default: false),  
    processingTime: Number (default: 0)  
  },  
  
  filters: {  
    topicId: ObjectId,  
    dateRange: {  
      start: Date,  
      end: Date  
    }  
  }  
}
```

```

        from: Date,
        to: Date
    },
    sortBy: String (enum: ['relevance', 'date', 'popularity'], default: 'relevance'),
    contentType: String (enum: ['all', 'posts', 'topics'], default: 'all')
},

sessionInfo: {
    ipAddress: String,
    userAgent: String,
    deviceType: String (enum: ['desktop', 'mobile', 'tablet', 'unknown'], default: 'unknown'),
    referrer: String
},
timestamp: Date (default: Date.now),

createdAt: Date (auto),
updatedAt: Date (auto)
}

```

➔ Chức năng: Ghi lại lịch sử tìm kiếm

Tính năng chính:

- Từ khóa tìm kiếm và kết quả
- Phân loại tìm kiếm (posts, topics, users, global)
- Thống kê hiệu suất tìm kiếm

### UserActivity Collection

```

{
    _id: ObjectId,
    userId: ObjectId (ref: 'User', required),

    activityType: String (enum: ['login', 'logout', 'view_post', 'create_post',
'edit_post', 'delete_post', 'comment', 'like', 'search', 'view_topic', 'register',
'profile_update', 'page_view'], required),

    details: {
        targetId: ObjectId,
        targetType: String (enum: ['post', 'topic', 'user', 'comment']),
        metadata: Mixed (default: {})
    },
}

```

```

    sessionInfo: {
      ipAddress: String,
      userAgent: String,
      deviceType: String (enum: ['desktop', 'mobile', 'tablet', 'unknown'], default: 'unknown'),
      browser: String,
      os: String,
      referrer: String,
      currentUrl: String
    },
    timestamp: Date (default: Date.now),
    sessionDuration: Number (default: 0),
    createdAt: Date (auto),
    updatedAt: Date (auto)
  }

```

➔ Chức năng: Theo dõi hoạt động người dùng cho analytics

Tính năng chính:

- Ghi lại các hoạt động (login, view\_post, create\_post, comment, like, search)
- Thông tin session (IP, device, browser)
- Metadata chi tiết cho từng hoạt động

### **ForumRules Collection**

```

{
  title: {
    type: String,
    required: true,
    default: 'Quy định diễn đàn'
  },
  content: {
    type: String,
    required: true,
    default: `

# Quy định diễn đàn sinh viên TVU

```

#### ## 1. Quy định chung

- Tôn trọng các thành viên khác trong diễn đàn
- Không sử dụng ngôn từ thô tục, xúc phạm
- Không spam hoặc đăng nội dung không liên quan

## ## 2. Quy định về nội dung

- Nội dung phải phù hợp với chủ đề diễn đàn
- Không đăng nội dung vi phạm pháp luật
- Không chia sẻ thông tin cá nhân của người khác

## ## 3. Quy định về hình ảnh

- Không đăng hình ảnh không phù hợp
- Tôn trọng bản quyền hình ảnh
- Hình ảnh phải liên quan đến nội dung bài viết

## ## 4. Hình phạt

- Cảnh báo lần đầu
- Khóa tài khoản tạm thời
- Khóa tài khoản vĩnh viễn đối với vi phạm nghiêm trọng

## ## 5. Liên hệ

Mọi thắc mắc xin liên hệ admin qua hệ thống chat.

```
`.trim()  
},  
isActive: {  
    type: Boolean,  
    default: true  
},  
version: {  
    type: Number,  
    default: 1  
},  
lastUpdatedBy: {  
    type: mongoose.Schema.Types.ObjectId,  
    ref: 'User'  
},  
createdAt: {  
    type: Date,  
    default: Date.now  
},  
updatedAt: {  
    type: Date,  
    default: Date.now  
}  
}
```

➔ **Chức năng:** Dùng để lưu trữ các quy định, nội quy của diễn đàn.

Tính năng chính:

- o Các nội dung chi tiết: tiêu đề, nội dung quy định, trạng thái hiệu lực (title, continent, isActive)
- o Thông tin phiên bản, người dùng cuối cùng cập nhật quy định (version, lastUpdatedBy)
- o Thông tin ngày tạo (createdAt, updatedAt)

## **Conversation Collection**

```
{
  // Danh sách người tham gia (2 người cho chat 1-1)
  participants: [
    {
      type: mongoose.Schema.Types.ObjectId,
      ref: 'User',
      required: true
    },
    // Loại cuộc trò chuyện
    type: {
      type: String,
      enum: ['direct', 'group'], // Hiện tại chỉ support direct (1-1)
      default: 'direct'
    },
    // Tin nhắn cuối cùng
    lastMessage: {
      type: mongoose.Schema.Types.ObjectId,
      ref: 'Message',
      default: null
    },
    // Thời gian tin nhắn cuối
    lastMessageAt: {
      type: Date,
      default: Date.now
    },
    // Trạng thái cuộc trò chuyện
    status: {
      type: String,
      enum: ['active', 'archived', 'blocked'],
      default: 'active'
    },
    // Thông tin về việc đã đọc tin nhắn cuối của từng người
  ]
}
```

```

readStatus: [
    userId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'User'
    },
    lastReadMessageId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'Message'
    },
    lastReadAt: {
        type: Date,
        default: Date.now
    }
],
// Cài đặt thông báo cho từng người
notificationSettings: [
    userId: {
        type: mongoose.Schema.Types.ObjectId,
        ref: 'User'
    },
    muted: {
        type: Boolean,
        default: false
    },
    mutedUntil: {
        type: Date,
        default: null
    }
],
// Thông kê
messageCount: {
    type: Number,
    default: 0
},
// Metadata bổ sung
metadata: {
    type: mongoose.Schema.Types.Mixed,
    default: {}
},
{
    timestamps: true
}

```

➔ **Chức năng:** Dùng để lưu trữ thông tin về các cuộc trò chuyện (hội thoại) giữa các user. Mỗi document thường đại diện cho một cuộc chat (giữa 2 người hoặc nhóm), chứa thông tin như: danh sách thành viên, thời gian tạo, trạng thái,...

Tính năng chính:

- Danh sách user tham gia cuộc trò chuyện, loại, tham chiếu đến tin nhắn cuối cùng (participants, type, lastMessage).
- Thời gian gửi tin nhắn cuối cùng, trạng thái, theo dõi user đã đọc tin nhắn hay chưa (lastMessageAt, status, readStatus).
- Cài đặt thông báo, tổng số tin nhắn trong cuộc trò chuyện, lưu trữ thông tin bổ sung (notificationSettings, messageCount, metadata).

## Message Collection

```
{  
    // ID cuộc trò chuyện  
    conversationId: {  
        type: mongoose.Schema.Types.ObjectId,  
        ref: 'Conversation',  
        required: true,  
        index: true  
    },  
  
    // Người gửi  
    senderId: {  
        type: mongoose.Schema.Types.ObjectId,  
        ref: 'User',  
        required: true  
    },  
  
    // Người nhận  
    receiverId: {  
        type: mongoose.Schema.Types.ObjectId,  
        ref: 'User',  
        required: true  
    },  
  
    // Nội dung tin nhắn  
    content: {  
        type: String,  
        required: true,  
    },
```

```

        trim: true,
        maxlen: 2000
    },
    // Loại tin nhắn
    messageType: {
        type: String,
        enum: ['text', 'image', 'file', 'system'],
        default: 'text'
    },
    // File đính kèm (nếu có)
    attachments: [
        fileName: String,
        fileUrl: String,
        fileSize: Number,
        mimeType: String
    ],
    // Trạng thái tin nhắn
    status: {
        type: String,
        enum: ['sent', 'delivered', 'read'],
        default: 'sent'
    },
    // Thời gian đọc tin nhắn
    readAt: {
        type: Date,
        default: null
    },
    // Tin nhắn đã bị xóa
    isDeleted: {
        type: Boolean,
        default: false
    },
    // Thời gian xóa
    deletedAt: {
        type: Date,
        default: null
    },
    // Tin nhắn được chỉnh sửa

```

```

isEdited: {
    type: Boolean,
    default: false
},

// Thời gian chỉnh sửa cuối
editedAt: {
    type: Date,
    default: null
},

// Nội dung gốc (trước khi chỉnh sửa)
originalContent: {
    type: String,
    default: null
},

// Metadata bổ sung
metadata: {
    type: mongoose.Schema.Types.Mixed,
    default: {}
},
{
    timestamps: true
}

```

➔ Chức năng: Dùng để lưu trữ các tin nhắn trong từng cuộc trò chuyện.

Mỗi document là một tin nhắn, liên kết với một conversation cụ thể, chứa nội dung, người gửi, thời gian gửi, trạng thái đã đọc,...

Tính năng chính:

- conversationId: Tham chiếu đến cuộc trò chuyện chứa tin nhắn này.
- senderId: User gửi tin nhắn.
- receiverId: User nhận tin nhắn (1-1).
- content: Nội dung tin nhắn (text).
- messageType: Loại tin nhắn (text, image, file, system).
- attachments: Danh sách file đính kèm (nếu có).
- status: Trạng thái tin nhắn (sent, delivered, read).
- readAt: Thời gian tin nhắn được đọc.
- isDeleted: Đánh dấu tin nhắn đã bị xóa hay chưa.

- deletedAt: Thời gian xóa tin nhắn.
- isEdited: Đánh dấu tin nhắn đã được chỉnh sửa hay chưa.

### 3.3 Thiết kế xử lý

#### 3.3.1 Sơ đồ chức năng

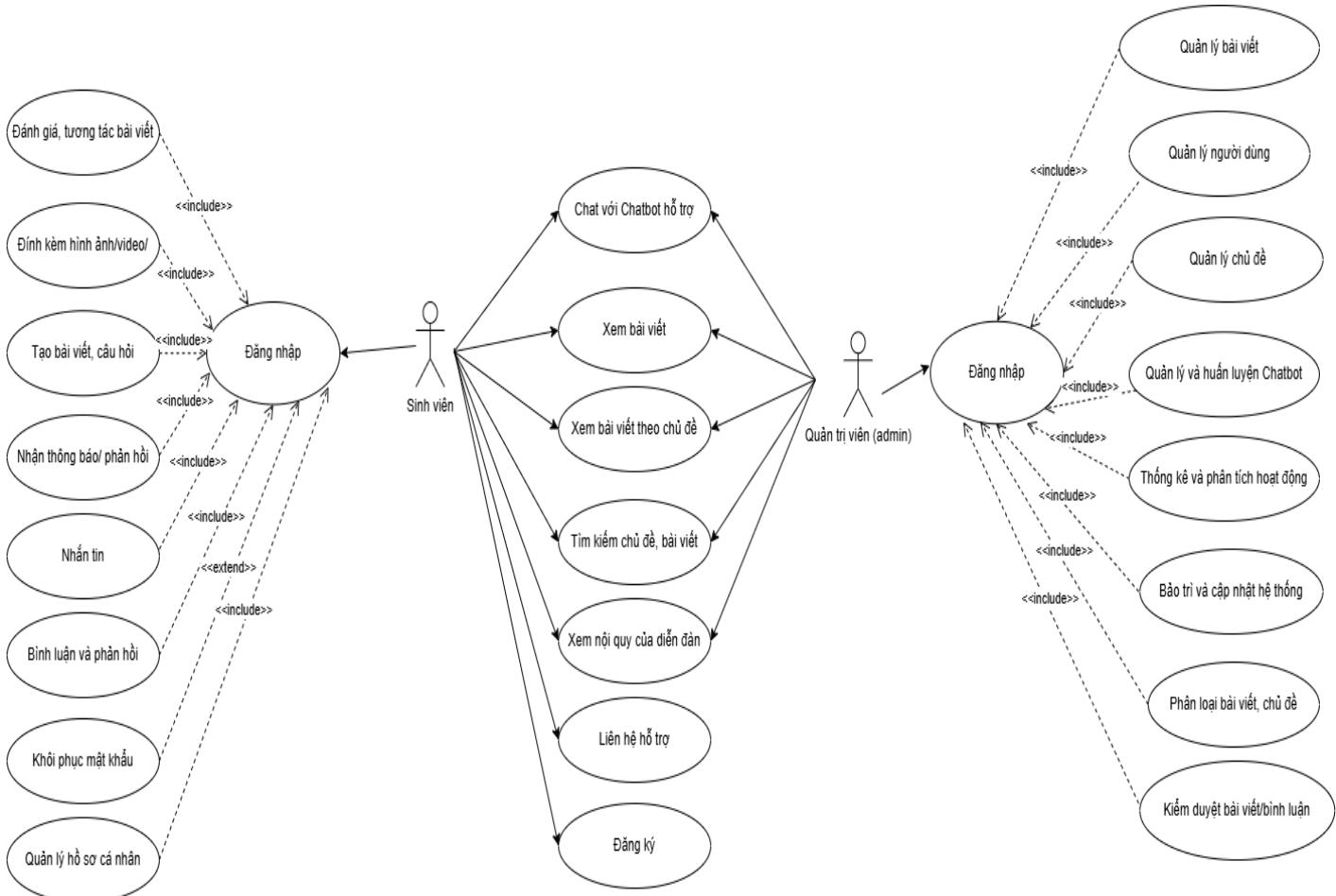


**Hệ thống kết nối của diễn đàn** bao gồm các hoạt động: Đăng ký, Đăng nhập, thông qua xác thực Email. Người dùng có thể chỉnh sửa trang cá nhân của mình nếu họ đã có tài khoản đăng nhập trong diễn đàn. Phân quyền của hệ thống gồm hai quyền: user (thành viên, người dùng) và admin (quản trị viên).

**Hệ thống đăng bài:** thành viên có thể xem các bài viết, chi tiết bài viết. Đăng bài, chỉnh sửa, xóa bài viết của họ. Tương tác với các bài viết khác như: tương tác lượt thích, bình luận, phản hồi bình luận và đánh giá.

**Hệ thống thông báo:** hệ thống sẽ thông báo các hoạt động như: khi người dùng nhận được các tương tác về bài viết của họ: lượt thích, bình luận, đánh giá. Các thông báo quan trọng đến từ quản trị viên. Nhận các thông báo, phản hồi từ ChatBot: các thông tin trả lời câu hỏi người dùng. Các thành viên có thể trò chuyện, nhận tin nhắn giữa các thành viên với nhau thông qua tính năng chat của diễn đàn. Liên hệ với quản trị viên thông qua các thông tin liên hệ như: số điện thoại,...

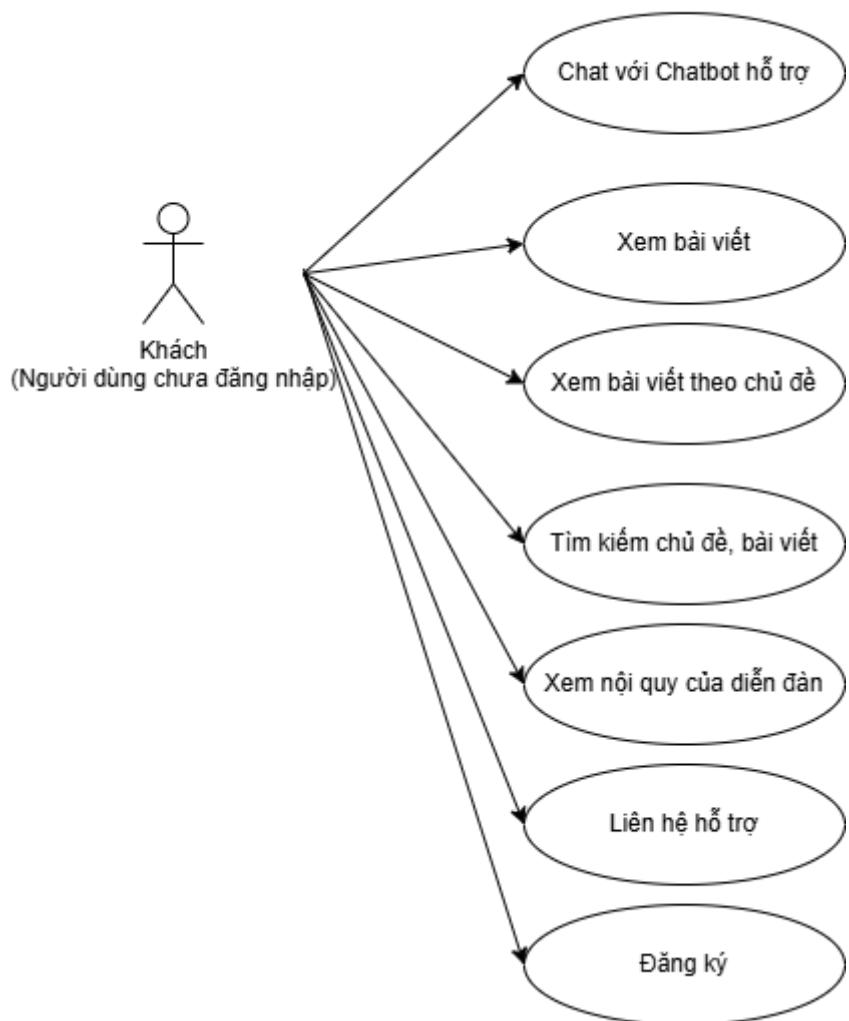
### 3.3.2 Biểu đồ Use Case tổng quát



**Hình 3.2 Biểu đồ Use Case tổng quát**

Biểu đồ usecase mô tả tổng quát các tác nhân trong hệ thống diễn đàn: Người dùng mới (khách) chưa có tài khoản đăng nhập, Sinh viên là thành viên thuộc diễn đàn và Quản trị viên của hệ thống.

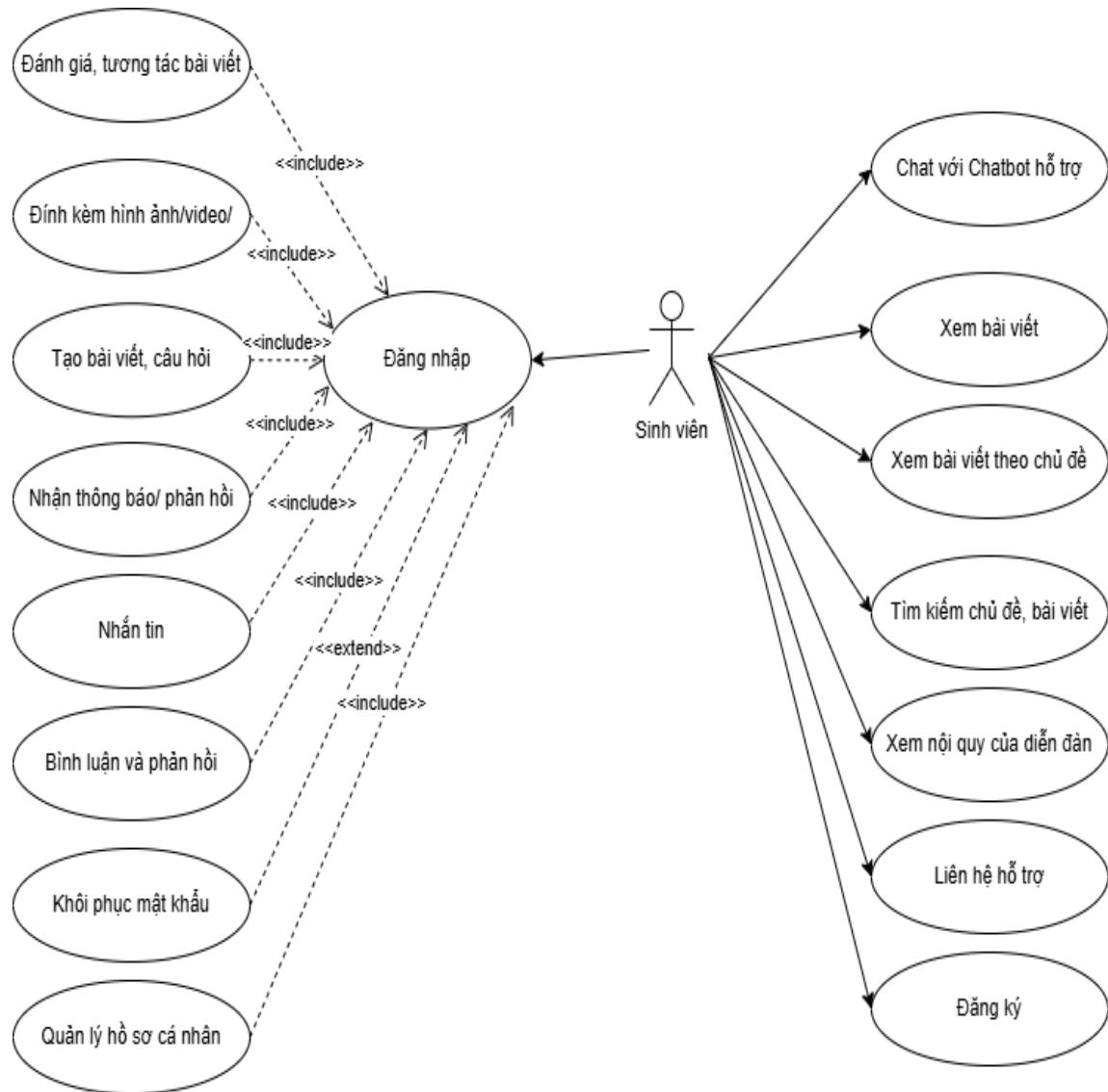
### 3.3.3 Biểu đồ Use Case tác nhân người dùng chưa đăng nhập (khách)



**Hình 3.3 Biểu đồ Use Case tác nhân khách (chưa đăng nhập)**

Người dùng chưa đăng nhập (khách) có thể xem các bài viết trong các chủ đề, tìm kiếm, chat với chatbot để được trả lời các câu hỏi, thắc mắc. Xem nội quy diễn đàn, tiến hành đăng ký để trở thành thành viên diễn đàn và có thể liên hệ trực tiếp đội ngũ hỗ trợ hoặc quản trị viên để được hướng dẫn chi tiết.

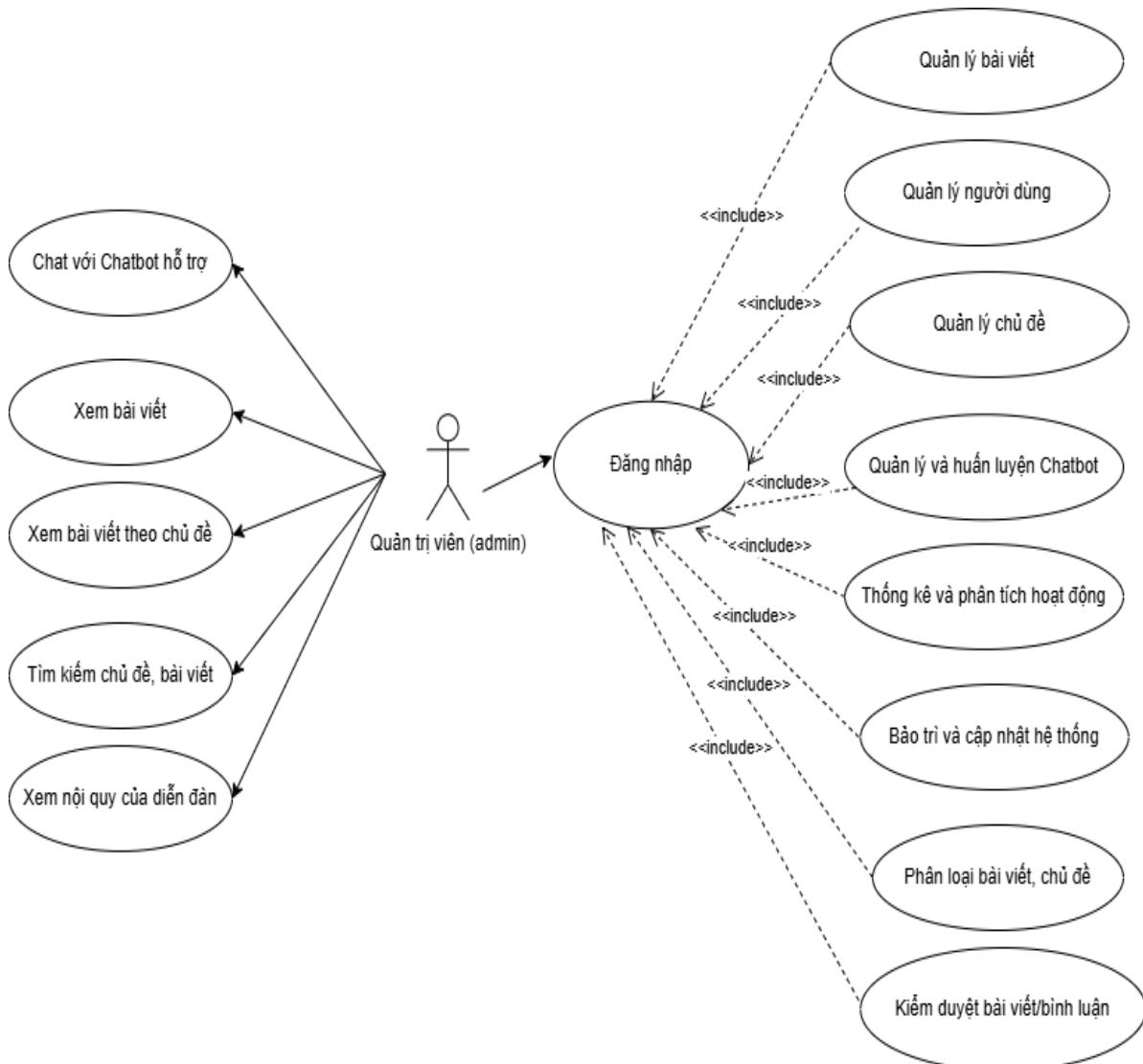
### 3.3.4 Biểu đồ Use Case tác nhân Sinh viên (thành viên, người dùng)



Hình 3.4 Biểu đồ Use Case tác nhân Sinh viên

Người dùng đã có tài khoản đăng nhập: thành viên diễn đàn, sinh viên,... được phép thao tác: đăng bài viết, câu hỏi thảo luận trong chủ đề lựa chọn. Tương tác bằng các lượt thích, tiến hành đánh giá bài viết của các thành viên. Nội dung bài viết được đính kèm hình ảnh, video đảm bảo tuân theo quy định đăng bài có trong hệ thống. Nhận được các thông báo từ hệ thống, quản trị viên về các phản hồi và các tin tức mới. Thành viên trong diễn đàn có thể nhắn tin cho nhau thông qua hệ thống tin nhắn TVU\_message để trao đổi hoặc trò chuyện.

### 3.3.5 Biểu đồ Use case tác nhân Quản trị



**Hình 3.5 Biểu đồ Use Case tác nhân Quản trị**

Quản trị viên có toàn quyền trong hệ thống diễn đàn: Quản lý bài viết, các bài viết được duyệt sẽ xuất hiện trên trang diễn đàn... Các thành viên vi phạm quy chế sẽ bị nhắc nhở, thậm chí là khóa tài khoản. Chủ đề có nhiều bài viết, được phân loại là chủ đề phổ biến, tương tự cho các bài viết có nhiều tương tác có đóng góp tích cực cho diễn đàn sẽ được phân loại tốt. Thông kê các bài viết, chủ đề, các lượt tương tác, thành viên. Ghi nhận các câu hỏi phổ biến của người dùng để nâng cao hiệu quả Chatbot.

### 3.3.6 Sơ đồ tuần tự

#### Quy trình đăng bài trên hệ thống diễn đàn:

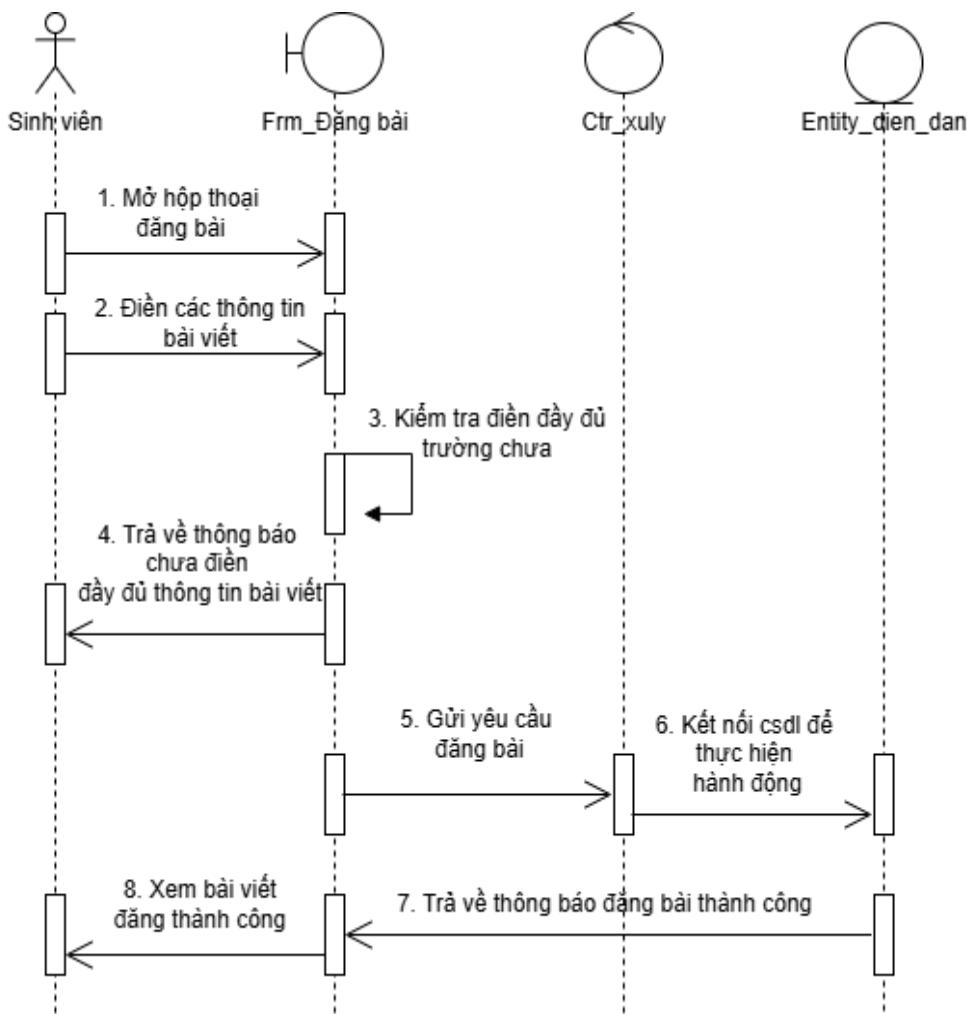
- Tác nhân: Sinh viên (thành viên, người dùng đã đăng nhập).

- Mục đích: Đăng bài lên diễn đàn.
- Điều kiện: Bắt buộc đăng nhập.
- Mô tả: Khi muốn đăng bài, người dùng bắt buộc phải đăng nhập vào hệ thống. Tiến hành nhập đầy đủ các thông tin của bài viết và nhấn nút đăng bài.

**Bảng 3.1 Quy trình đăng bài trên hệ thống**

Hành động của Actor	Phản ứng của hệ thống
1. Người dùng mở hộp thoại đăng bài  3. Nhập các thông tin bài viết cần đăng  6. Xem thông báo và bài đăng thành công  7. Tiếp tục hoặc kết thúc phiên đăng bài	2. Hiện hộp thoại đăng bài  4. Kiểm tra đã nhập đầy đủ thông tin chưa  5. Trả về thông báo đăng bài thành công

- Sự kiện thay thế: Bước 5: Sau khi kiểm tra đã nhập thông tin đầy đủ trong hộp thoại đăng bài chưa, hệ thống phát hiện thiếu thông tin (ví dụ thiếu tiêu đề hoặc nội dung để trống). Hệ thống trả về thông báo “cần nhập đầy đủ thông tin đăng bài”. Bước 6: người đăng xem thông báo để thêm đầy đủ thông tin.



**Hình 3.6 Sơ đồ tuần tự đăng bài của thành viên**

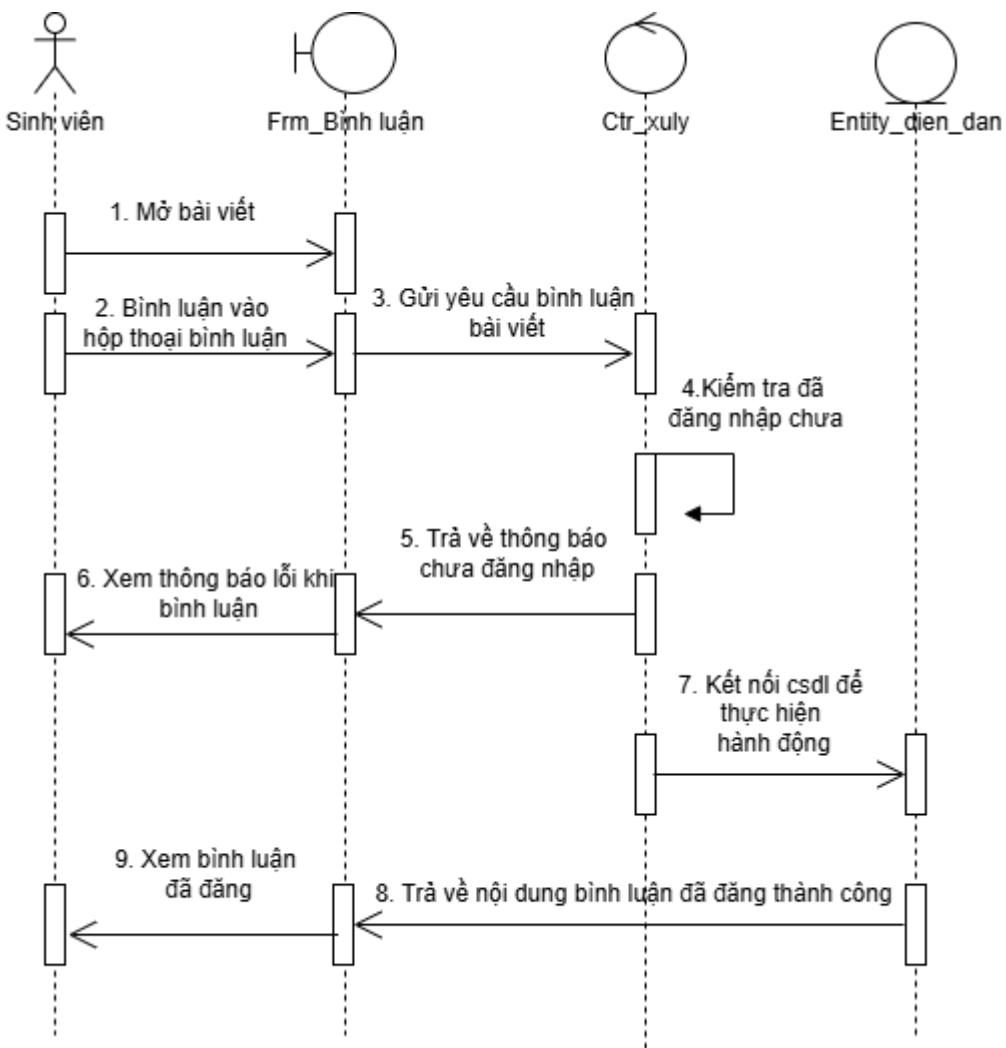
#### Quy trình bình luận bài viết:

- Tác nhân: Sinh viên (thành viên, người dùng đã đăng nhập).
- Mục đích: Bình luận bài viết trên diễn đàn.
- Điều kiện: Bắt buộc đăng nhập.
- Mô tả: Khi muốn bình luận bài viết nào đó trên hệ thống diễn đàn, người dùng bắt buộc phải đăng nhập vào hệ thống. Tiến hành bình luận vào hộp thoại bình luận.

**Bảng 3.2 Quy trình bình luận bài viết trên diễn đàn**

Hành động của Actor	Phản ứng của hệ thống
1. Người dùng mở hộp thoại bình luận  3. Nhập nội dung bình luận vào hộp thoại  6. Xem thông báo và bình luận thành công  7. Tiếp tục hoặc kết thúc phiên bình luận	2. Hiện hộp thoại bình luận  4. Kiểm tra đã đăng nhập chưa  5. Trả về thông báo bình luận thành công

- Sự kiện thay thế: Bước 5: Sau khi kiểm tra đã đăng nhập chưa, hệ thống phát hiện chưa đăng nhập. Hệ thống trả về thông báo “bạn cần đăng nhập để bình luận”. Bước 6: người dùng xem thông báo để tiến hành đăng nhập.



**Hình 3.7 Sơ đồ tuần tự bình luận bài viết**

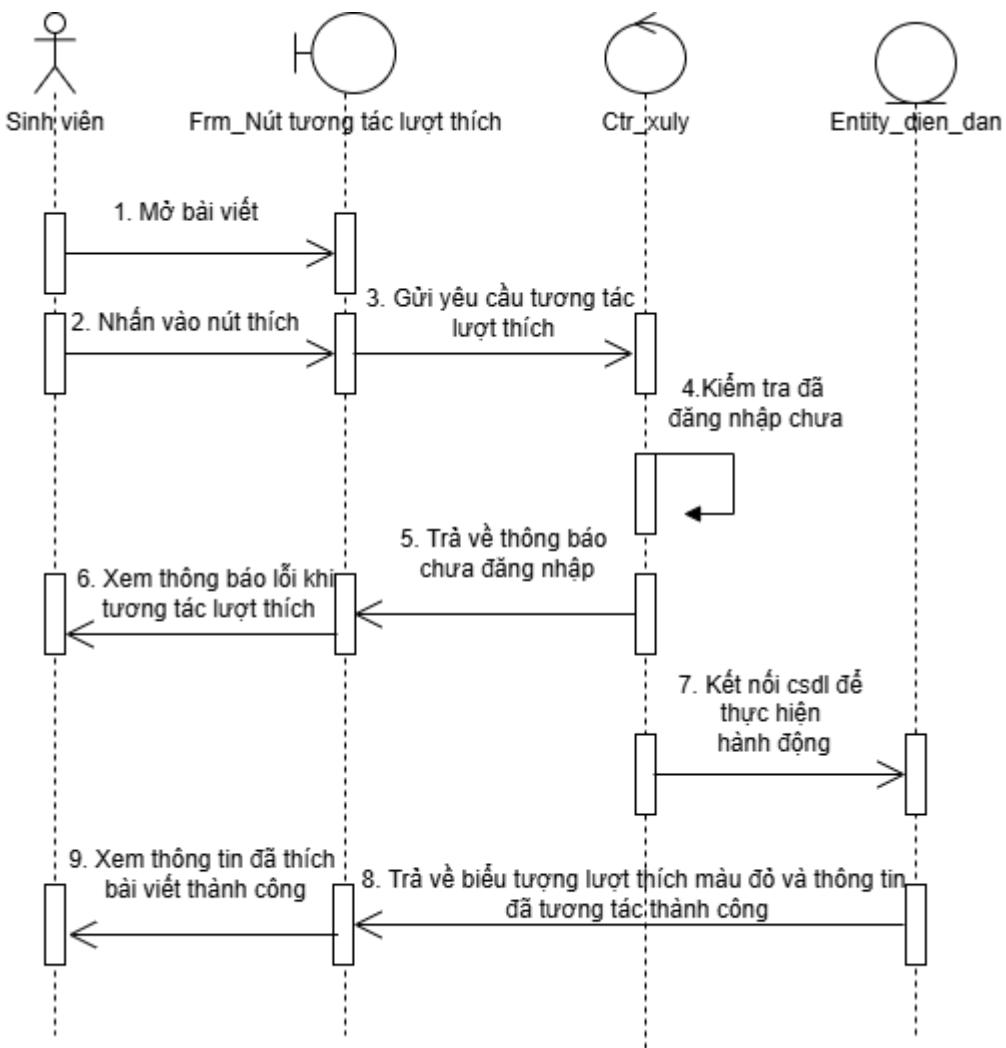
#### Quy trình tương tác lượt thích bài viết:

- Tác nhân: Sinh viên (thành viên, người dùng đã đăng nhập).
- Mục đích: Tương tác bài viết bằng lượt thích trên diễn đàn.
- Điều kiện: Bắt buộc đăng nhập.
- Mô tả: Khi muốn tương tác bài viết bằng lượt thích trên hệ thống diễn đàn, người dùng bắt buộc phải đăng nhập vào hệ thống. Tiến hành nhấp vào nút thích để tương tác.

**Bảng 3.3 Quy trình tương tác bài viết bằng lượt thích**

Hành động của Actor	Phản ứng của hệ thống
1. Người dùng mở bài viết cần tương tác  3. Nhấn vào nút thích để tương tác  6. Xem biểu tượng và thông báo tương tác lượt thích thành công  7. Tiếp tục hoặc kết thúc phiên tương tác lượt thích.	2. Hiện bài viết lên giao diện  4. Kiểm tra đã đăng nhập chưa  5. Trả về biểu tượng “thích” đã đỗ

- Sự kiện thay thế: Bước 5: Sau khi kiểm tra đã đăng nhập chưa, hệ thống phát hiện chưa đăng nhập. Hệ thống trả về thông báo “bạn cần đăng nhập để tương tác lượt thích cho bài viết”. Bước 6: người dùng xem thông báo để tiến hành đăng nhập.



**Hình 3.8 Sơ đồ tuần tự tương tác lượt thích**

#### Quy trình đánh giá bài viết:

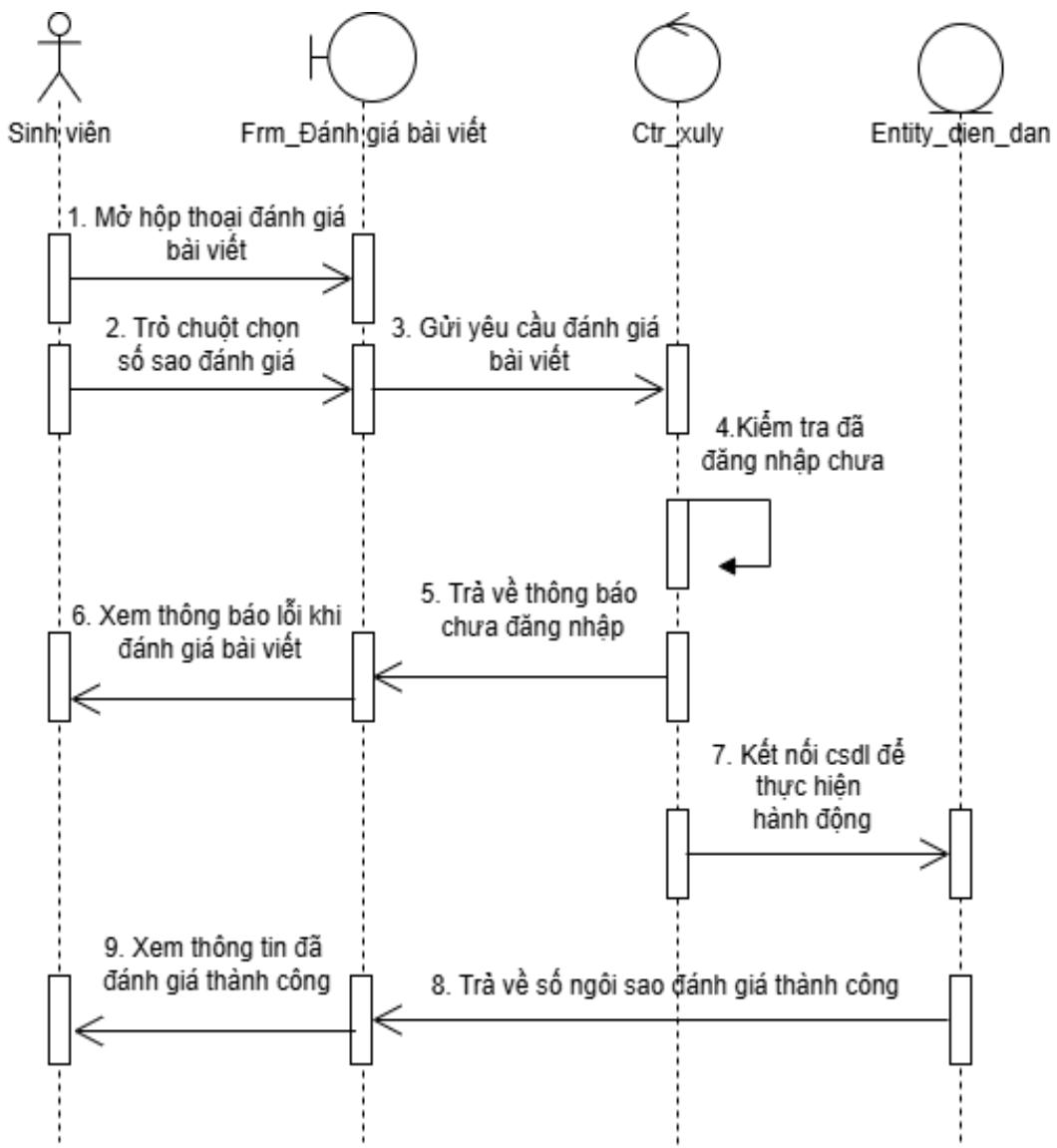
- Tác nhân: Sinh viên (thành viên, người dùng đã đăng nhập).
- Mục đích: Đánh giá bài viết trên diễn đàn.
- Điều kiện: Bắt buộc đăng nhập.

Mô tả: Khi muốn đánh giá bài viết trên hệ thống diễn đàn, người dùng bắt buộc phải đăng nhập vào hệ thống. Tiến hành chọn số sao (từ 1 đến 5) để đánh giá bài viết.

**Bảng 3.4 Quy trình đánh giá bài viết**

Hành động của Actor	Phản ứng của hệ thống
<p>1. Người dùng mở bài viết cần đánh giá</p> <p>3. Chọn số sao (từ 1 đến 5) để đánh giá</p> <p>6. Xem số sao đánh giá thành công và thông tin trong hộp thoại đánh giá</p> <p>7. Tiếp tục hoặc kết thúc phiên đánh giá bài viết.</p>	<p>2. Hiện bài viết lên giao diện</p> <p>4. Kiểm tra đã đăng nhập chưa</p> <p>5. Trả về số sao đã đánh giá lên bài viết</p>

- Sự kiện thay thế: Bước 5: Sau khi kiểm tra đã đăng nhập chưa, hệ thống phát hiện chưa đăng nhập. Hệ thống trả về thông báo “bạn cần đăng nhập để đánh giá bài viết”. Bước 6: người dùng xem thông báo để tiến hành đăng nhập.

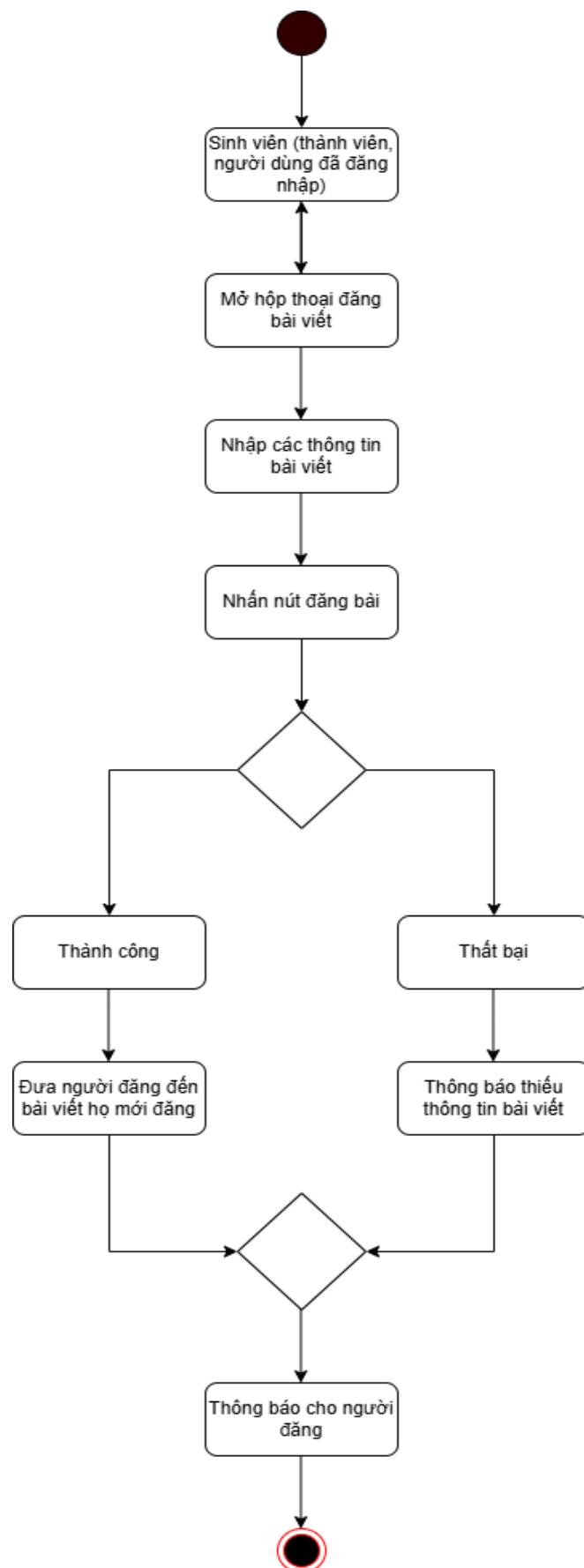


**Hình 3.9 Sơ đồ tuần tự đánh giá bài viết**

Sơ đồ tuần tự giúp khái quát một cách chi tiết, giúp người dùng hiểu rõ cách mà các thành phần trong website giao tiếp với nhau. Từ đó hiểu rõ luồng dữ liệu, tối ưu hóa hiệu suất cho hệ thống.

### 3.3.7 Sơ đồ hoạt động

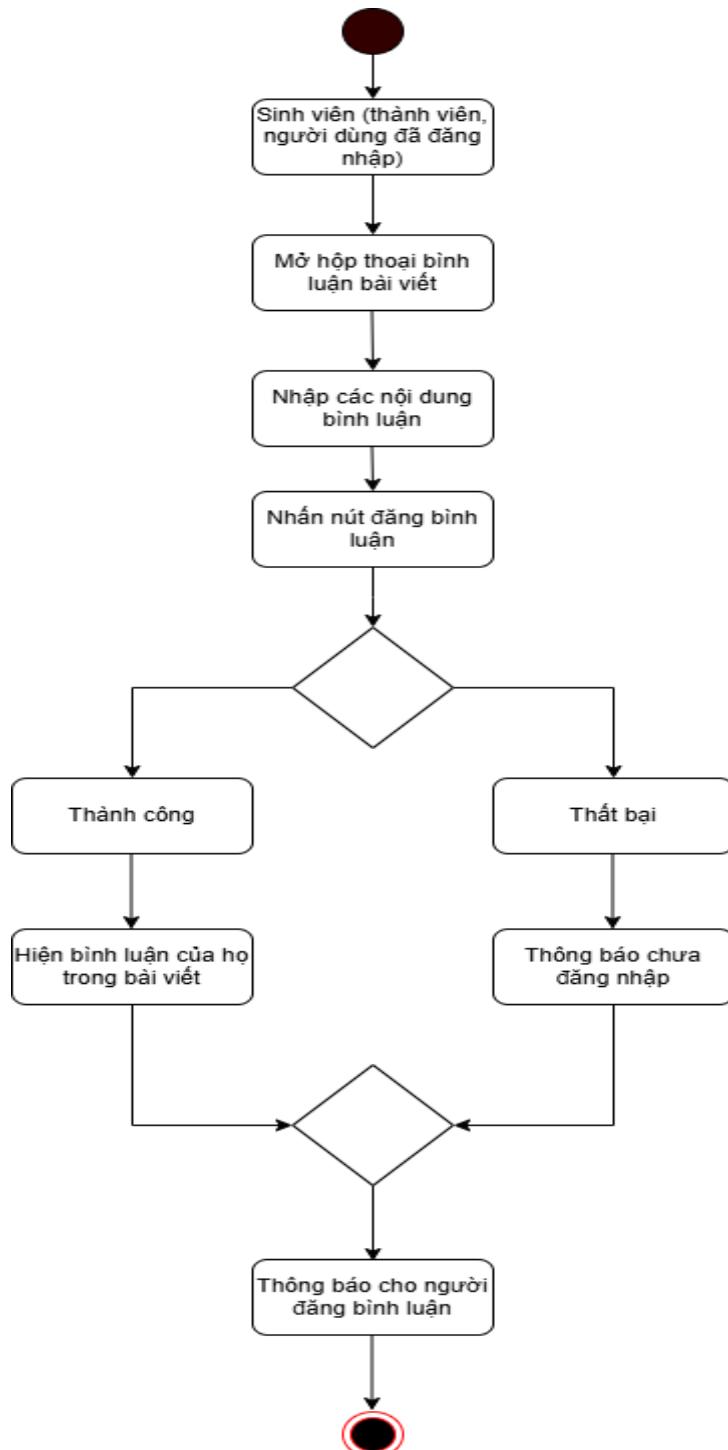
**Sơ đồ hoạt động đăng bài viết lên diễn đàn:**



**Hình 3.10 Sơ đồ hoạt động đăng bài viết**

Mô tả luồng hoạt động của sinh viên (thành viên, người dùng đã đăng nhập) khi đăng bài một cách chi tiết nhất: mở hộp thoại đăng bài, nhập các thông tin bài viết, nhấn nút đăng bài. Có hai trường hợp: “thành công” đưa người dùng để bài viết họ đã đăng. “Thất bại” gửi thông báo người đăng đã nhập thiếu thông tin cho bài viết.

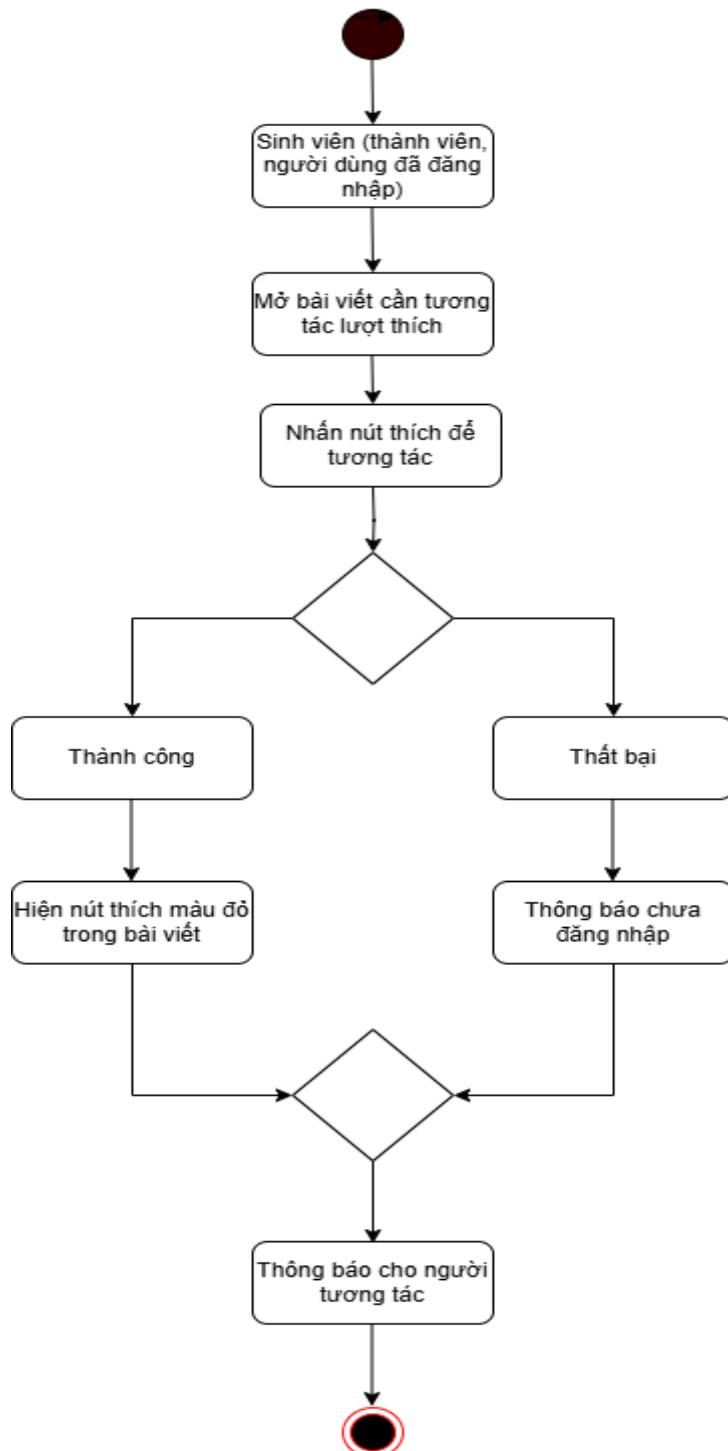
### Sơ đồ hoạt động bình luận bài viết:



**Hình 3.11 Sơ đồ hoạt động bình luận bài viết**

Mô tả luồng hoạt động của sinh viên (thành viên, người dùng đã đăng nhập) khi bình luận bài viết một cách chi tiết nhất: mở hộp thoại bình luận, nhập nội dung bình luận, nhấn nút đăng bình luận. Có hai trường hợp: “thành công” hiện bình luận của họ trong bài viết. “Thất bại” gửi thông báo người bình luận chưa đăng nhập.

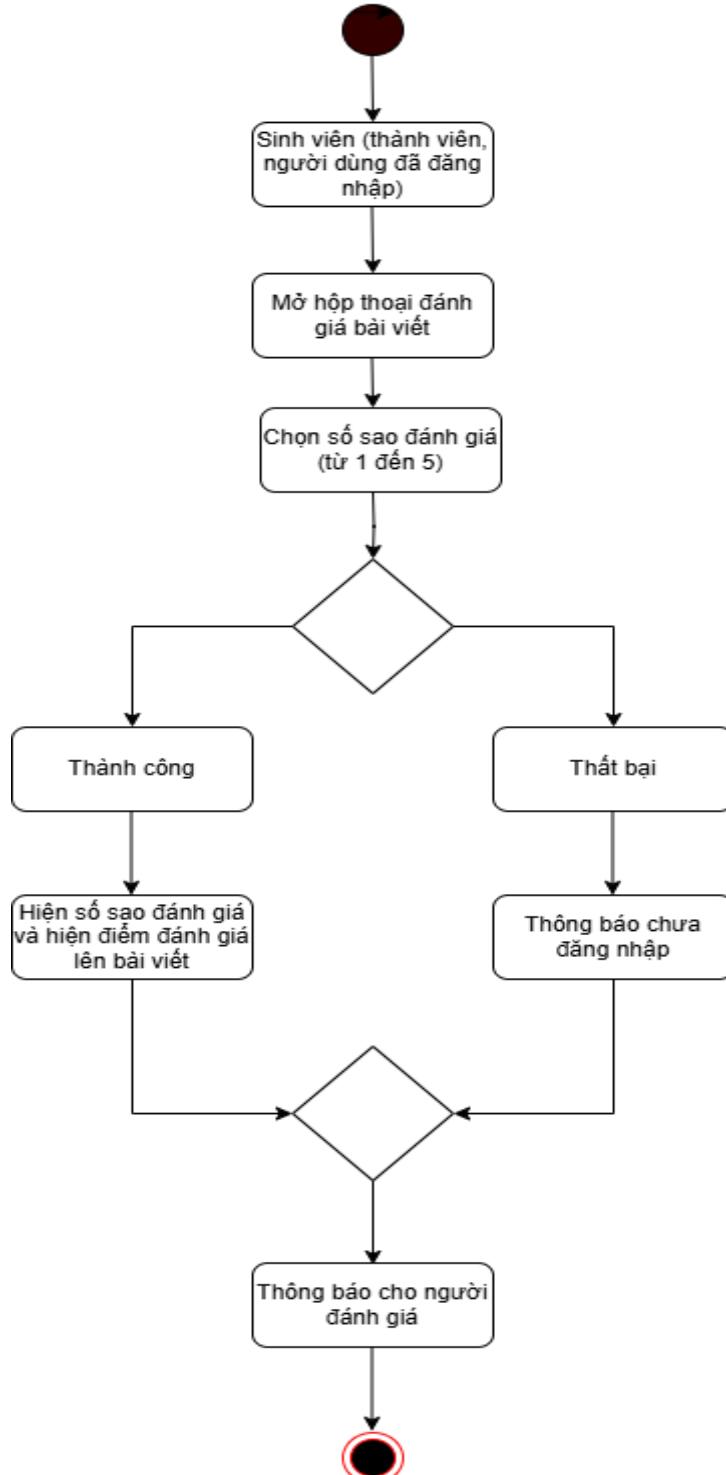
### Sơ đồ hoạt động tương tác bài viết bằng lượt thích:



Hình 3.12 Sơ đồ hoạt động tương tác bài viết bằng lượt thích

Mô tả luồng hoạt động của sinh viên (thành viên, người dùng đã đăng nhập) khi tương tác bài viết bằng lượt thích một cách chi tiết nhất: nhấn nút thích trong bài viết. Có hai trường hợp: “thành công” hiện nút thích màu đỏ, hiện thông tin người đã thích. “Thất bại” gửi thông báo người tương tác chưa đăng nhập.

### Sơ đồ hoạt động đánh giá bài viết:



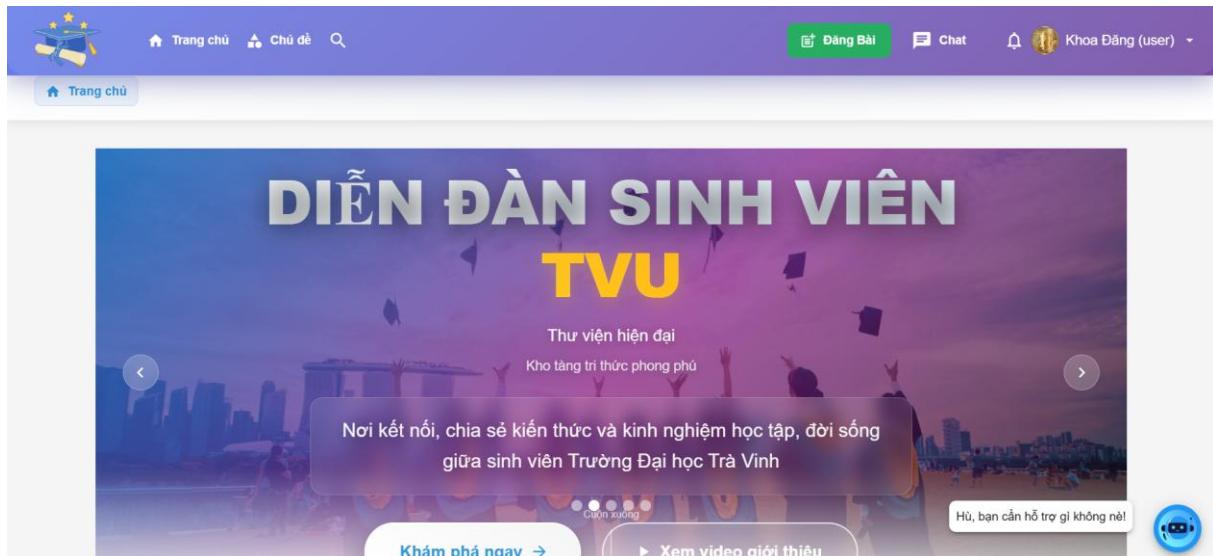
**Hình 3.13 Sơ đồ hoạt động đánh giá bài viết**

Mô tả luồng hoạt động của sinh viên (thành viên, người dùng đã đăng nhập) khi đánh giá bài viết một cách chi tiết nhất: mở hộp thoại đánh giá, chọn số sao đánh giá từ 1 đến 5, nhấn nút đánh giá. Có hai trường hợp: “thành công” hiện số sao đánh giá và điểm đánh giá của họ trong bài viết. “Thất bại” gửi thông báo người đánh giá chưa đăng nhập.

## CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

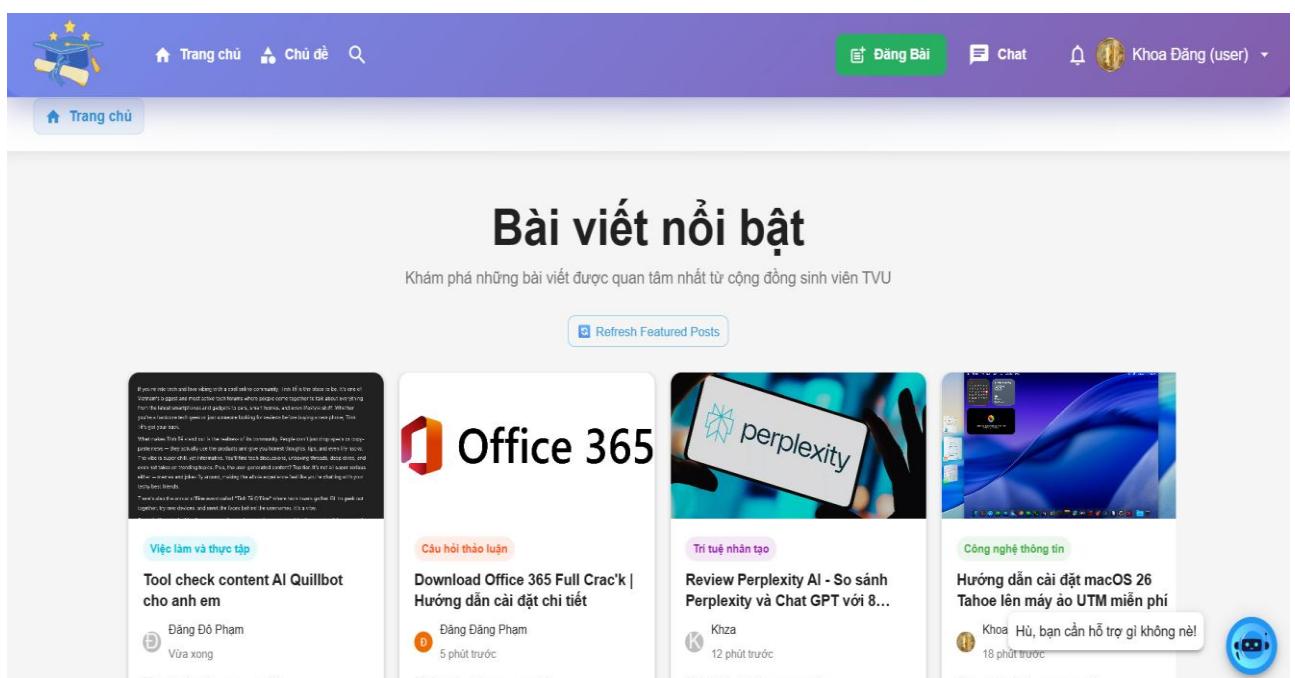
### 4.1 Giao diện người dùng

#### 4.1.1 Giao diện trang chủ



Hình 4.1 Giao diện Trang chủ (a)

Trang chủ của diễn đàn được thiết kế hiện đại, phù hợp với sinh viên trường Đại học.



Hình 4.2 Giao diện trang chủ (b)

Hiện các bài viết nổi bật có nhiều lượt tương tác: lượt thích, bình luận, đánh giá. Người xem có thể xem và tương tác với các bài viết này nếu đã đăng nhập.

The screenshot shows the homepage of a website with a purple header bar. The header includes a logo, navigation links for 'Trang chủ', 'Chủ đề', and a search icon, along with user icons for 'Đăng Bài', 'Chat', and 'Khoa Đặng (user)'. Below the header, a blue button labeled 'Trang chủ' is visible. The main content area features a section titled 'Chủ đề thịnh hành' (Trending topics) with a subtitle 'Những chủ đề được quan tâm và thảo luận nhiều nhất trong tuần' (The most popular topics and discussions in the week). There are eight topic cards arranged in two rows of four. Each card has a colored circular icon labeled 'topic', the topic name, the number of posts, and a small badge indicating new posts ('Thịnh hành') and their age ('+2 tuần này').

Topic	Bài viết	Thịnh hành	+2 tuần này
TRÍ TUỆ NHÂN TẠO	2 bài viết		
Trí tuệ nhân tạo	3 bài viết		+1 tuần này
Lập trình web	3 bài viết		+1 tuần này
Đời sống sinh viên	2 bài viết		
Khoa học máy tính	9 bài viết		
Công nghệ thông tin	3 bài viết		
Việc làm và thực tập	3 bài viết		
Câu hỏi thảo luận	3 bài viết		Hỗ trợ, bạn cần hỗ trợ gì không nè?

Hình 4.3 Giao diện trang chủ (c)

Các chủ đề thịnh hành được trình bày đơn giản. Nhấn vào để xem các chi tiết các chủ đề này.

The screenshot shows the homepage of a website with a purple header bar. The header includes a logo, navigation links for 'Trang chủ', 'Chủ đề', and a search icon, along with user icons for 'Đăng Bài', 'Chat', and 'Khoa Đặng (user)'. Below the header, a blue button labeled 'Trang chủ' is visible. The main content area features a section titled 'Tìm kiếm chủ đề' (Search for topics) with a subtitle 'Khám phá và tham gia thảo luận về các chủ đề bạn quan tâm' (Discover and participate in discussions about topics you care about). A search input field contains the placeholder 'Tim kiếm chủ đề...'. Below this, there is a table with columns: 'Chủ đề', 'Mô tả', 'Bài viết', and 'Bài viết gần nhất'. It lists two topics: 'TRÍ TUỆ NHÂN TẠO' and 'Khoa học máy tính'. Each topic card includes a small 'topic' icon, the topic name, a brief description, the number of posts, and a post by 'Ngù ngon' from 'bởi Khoa' posted 23 giờ trước.

Chủ đề	Mô tả	Bài viết	Bài viết gần nhất
TRÍ TUỆ NHÂN TẠO Thịnh hành	Đây là chủ đề đang khá thịnh hành ngày nay cho các bạn	2 bài viết	Ngù ngon bởi Khoa 23 giờ trước
Khoa học máy tính	Thảo luận về các chủ đề liên quan đến khoa học máy tính.	9 bài viết	Ngù ngon nè bởi Khoa 22 giờ trước

Hình 4.4 Giao diện trang chủ (d)

Người dùng có thể tìm kiếm các chủ đề theo tên. Các chủ đề trùng với từ khóa tìm kiếm sẽ hiện ngay bên dưới.

#### 4.1.2 Giao diện trang chủ với phần các Chủ đề

The screenshot shows a dashboard with a purple header. In the top right corner, there are buttons for 'Đăng Bài' (Post), 'Chat', and a user profile 'Khoa Đăng (user)'. Below the header, a navigation bar has 'Trang chủ' and 'Chủ đề' selected. The main content area displays four topics:

Chủ đề	Mô tả	Bài viết	Bài viết gần nhất
topic TRÍ TUỆ NHÂN TẠO	Đây là chủ đề đang khá thịnh hành ngày nay cho các bạn	2 bài viết	K Ngữ ngon bởi Khoa Đăng 23 giờ trước
topic Khoa học máy tính	Thảo luận về các chủ đề liên quan đến khoa học máy tính.	9 bài viết	K Ngữ ngon nè bởi Khoa Đăng 22 giờ trước
topic Lập trình web	Chia sẻ kiến thức về lập trình web.	3 bài viết	K I LOVE WEB bởi Khoa Đăng 9 giờ trước
topic Trí tuệ nhân tạo	Các bài viết, thảo luận về trí tuệ nhân tạo.	3 bài viết	K Review Perplexity AI - S^--^-- Hù, bạn cần hỏi gì không nè!

Hình 4.5 Giao diện Trang chủ với các Chủ đề

Bảng các chủ đề có trong diễn đàn gồm: số bài viết, bài viết gần nhất và tên tác giả của bài viết đó.

#### 4.1.3 Giao diện trang quy định diễn đàn

The screenshot shows a modal window titled 'Quy định diễn đàn' (Forum Rules). At the top, it says 'Phiên bản: 1 | Cập nhật: 15/6/2025'. The main content is 'Quy định diễn đàn sinh viên TVU' (Forum Rules for TVU Students) with three sections:

- 1. Quy định chung**
  - Tôn trọng các thành viên khác trong diễn đàn
  - Không sử dụng ngôn từ thô tục, xúc phạm
  - Không spam hoặc đăng nội dung không liên quan
- 2. Quy định về nội dung**
  - Nội dung phải phù hợp với chủ đề diễn đàn
  - Không đăng nội dung vi phạm pháp luật
  - Không chia sẻ thông tin cá nhân của người khác
- 3. Quy định về hình ảnh**
  - Không đăng hình ảnh không phù hợp

Below the sections, there is a checkbox:  Tôi đã đọc và đồng ý với tất cả các quy định trên. At the bottom right is a 'Đồng ý' (Agree) button.

Hình 4.6 Giao diện trang quy định diễn đàn

Quy định của diễn đàn là phần bắt buộc chấp nhận nếu người dùng tiến hành đăng bài hoặc tham gia các hoạt động khác trên diễn đàn.

#### 4.1.4 Giao diện trang chi tiết các Chủ đề

The screenshot shows a forum interface. At the top, there is a navigation bar with icons for Trang chủ, Chủ đề, Chat, Đăng Bài, and user information (dzng (user)). Below the navigation bar, the breadcrumb path shows 'Trang chủ > Công nghệ thông tin'. The main content area displays a topic titled 'Xin chào, dzng! 🎉' with a message: 'Chia sẻ kiến thức, kinh nghiệm hoặc đặt câu hỏi với cộng đồng'. Below this, there is a button to 'Viết bài mới hoặc đặt câu hỏi thú vị...' and another to 'Chia sẻ kinh nghiệm' or 'Đặt câu hỏi'. To the left, a sidebar lists categories: Chủ đề nổi bật, Khoa học máy tính, Lập trình web, Trí tuệ nhân tạo, Công nghệ thông tin, and Đời sống sinh viên. On the right, a sidebar titled 'Thành viên' lists users: dzng (Member), Administrator (Admin), hí hí (Member), and Đăng Đăng Phạm (Member). A message bubble at the bottom right says 'Hãy, bạn cần hỗ trợ gì không nè!'.

Hình 4.7 Giao diện trang chi tiết Chủ đề

Giao diện trang chi tiết bài viết hướng đến việc hiện các bài viết thuộc một chủ đề trong diễn đàn, các thông tin về các thành viên đang hoạt động cũng được cập nhật.

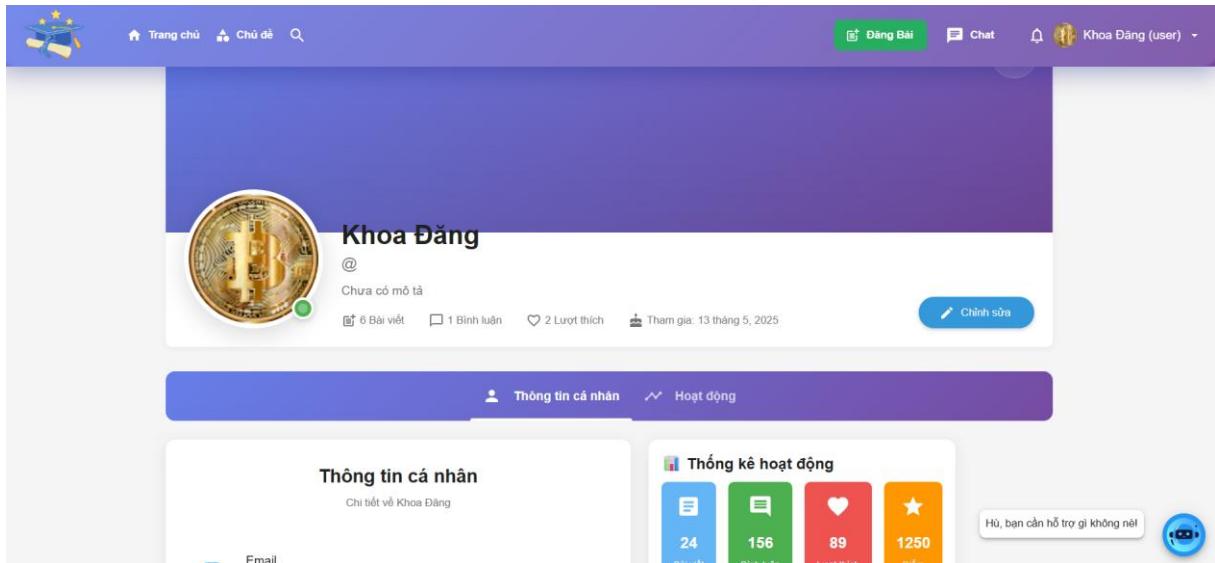
#### 4.1.5 Giao diện trang chi tiết bài viết

The screenshot shows a forum post detail page. The top navigation bar includes icons for Trang chủ, Chủ đề, Chat, Đăng Bài, and user information (Khoa Đăng (user)). The breadcrumb path shows 'Trang chủ > Câu hỏi thảo luận > Download Office 365 Full Crac'k | Hướng dẫn cài đặt chi tiết'. The main content area features a large title 'Download Office 365 Full Crac'k | Hướng dẫn cài đặt chi tiết'. Below the title, the author is 'Đăng Đăng Phạm' (with a profile picture 'D'), posted on '15/6/2025' and has '3 phút đọc'. There are '0 lượt xem' and a link 'Office 365'. To the right, there is a sidebar titled 'Bài viết liên quan' showing two related posts: 'Thảo luận: Ngôn ngữ lập trình nào nên học đầu tiên?' and 'Học đại học có còn cần thiết trong ngành IT?'. A message bubble at the bottom right says 'Hãy, bạn cần hỗ trợ gì không nè!'.

Hình 4.8 Giao diện trang chi tiết bài viết

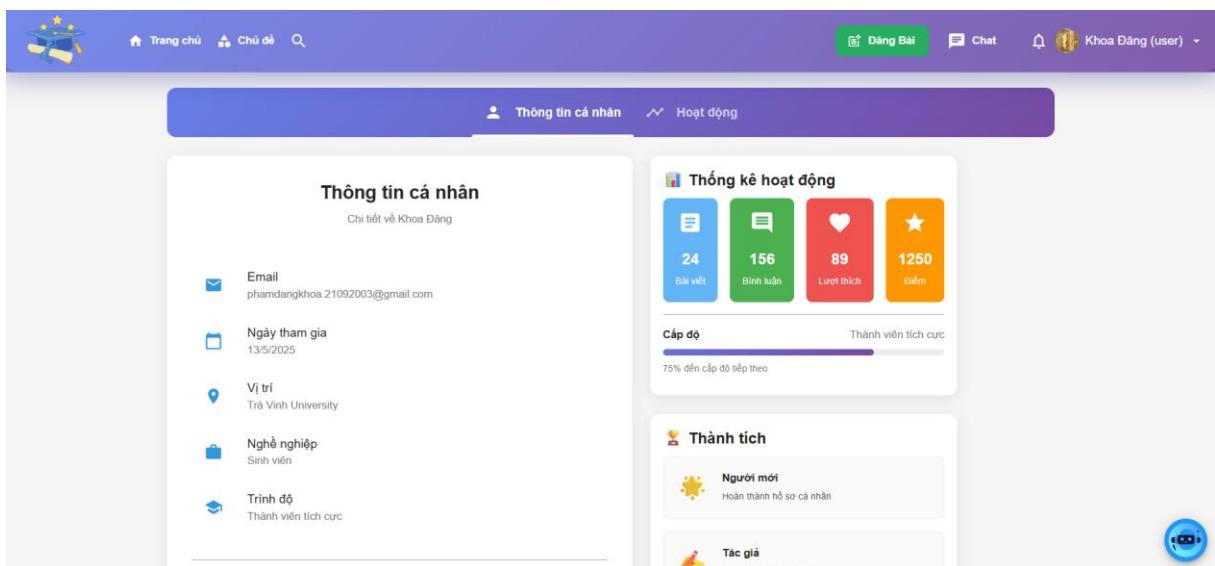
Trang chi tiết bài viết hiện các thông tin chi tiết như nội dung, tác giả, các lượt thích, bình luận, đánh giá một cách cụ thể. Bên cạnh là thanh bên phải với các bài viết tương tự có trong chủ đề đó.

#### 4.1.6 Giao diện trang cá nhân



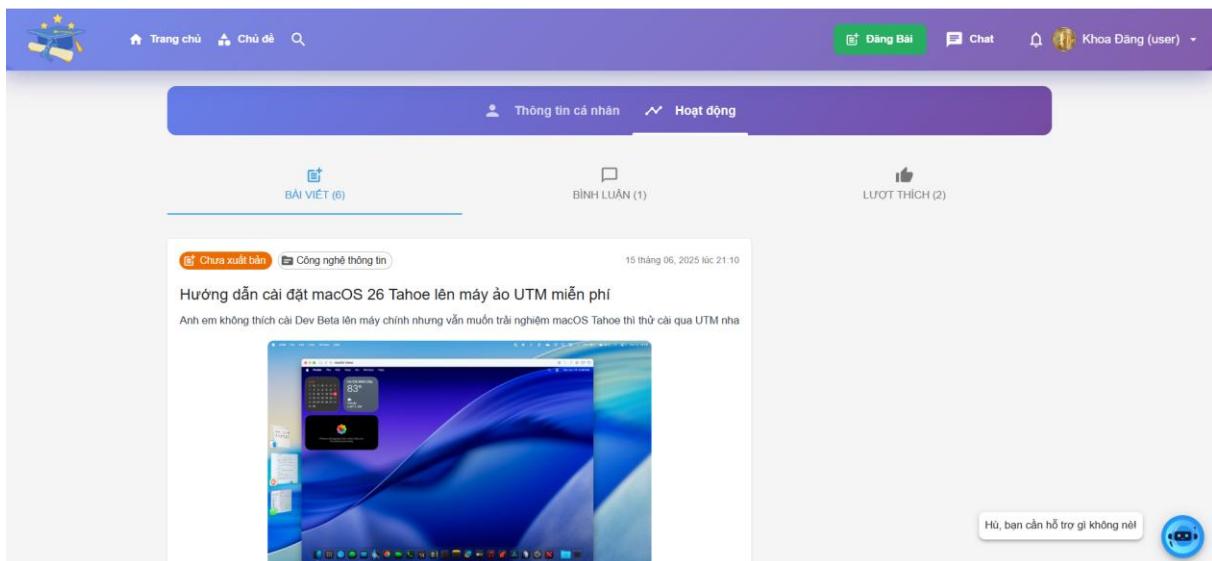
Hình 4.9 Giao diện trang cá nhân người dùng (a)

Giao diện trang cá nhân của người dùng hiển thị với ảnh đại diện, nút chỉnh sửa và các thông tin cá nhân.



Hình 4.10 Giao diện trang cá nhân người dùng (b)

Các thông tin như Email, Ngày tham gia, Vị trí, Nghề nghiệp cũng được hiển thị. Thông kê các hoạt động như số bài viết, các lượt tương tác, bình luận và đánh giá.



**Hình 4.11 Giao diện trang cá nhân người dùng (c)**

Nội dung các bài viết kèm theo nút điều hướng đến trang chi tiết bài viết.

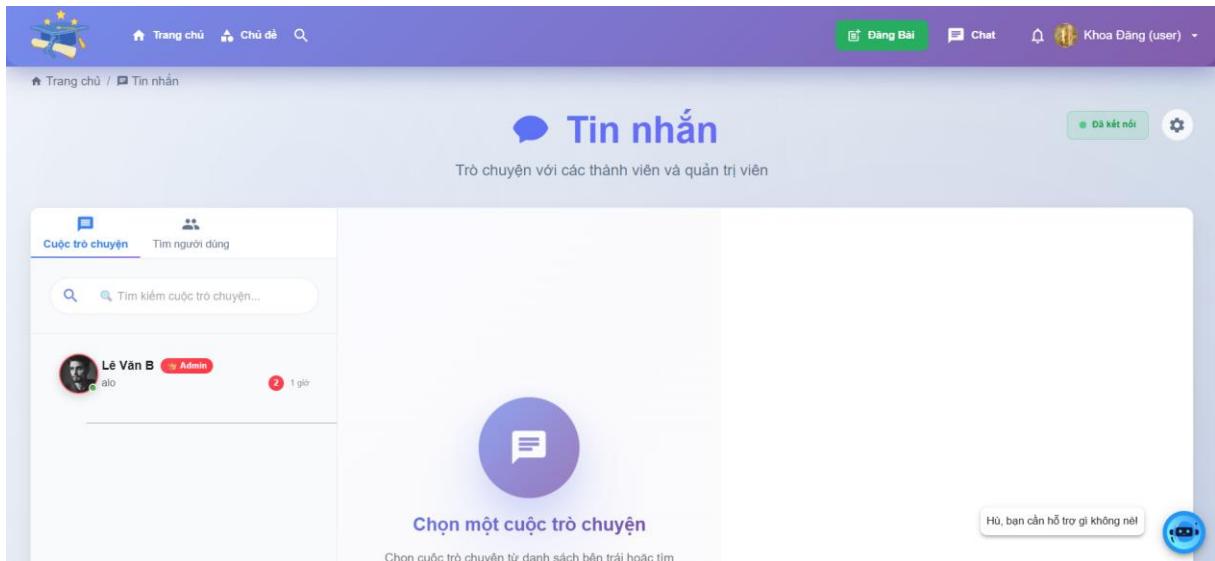
#### 4.1.7 Giao diện trang các thành viên diễn đàn

The screenshot shows a list of forum members. At the top, there are navigation links: Trang chủ, Chủ đề, and a search bar. On the right, there are buttons for Đăng Bài, Chat, and Khoa Đặng (user). The main title is 'Thành viên diễn đàn'. Below it, a sub-header says 'Khám phá và kết nối với cộng đồng thành viên tại diễn đàn của chúng tôi'. A search bar allows users to 'Tim kiếm thành viên theo tên, username hoặc email...'. Below the search bar, a message bubble says 'Tim thấy 9 thành viên'. The member list consists of nine cards, each showing a profile picture, name, username, and a summary of their posts and activity. The members are: Đăng Đô Phạm (@dazngdo121212), dzng (@dznghai121212), Administrator (@admin), hí hí (@khoa111), Đăng Đăng Phạm (@dzng121212), B (@a), and Khoa Đặng (@phamdangkhoa\_21092003).

**Hình 4.12 Giao diện trang các thành viên diễn đàn**

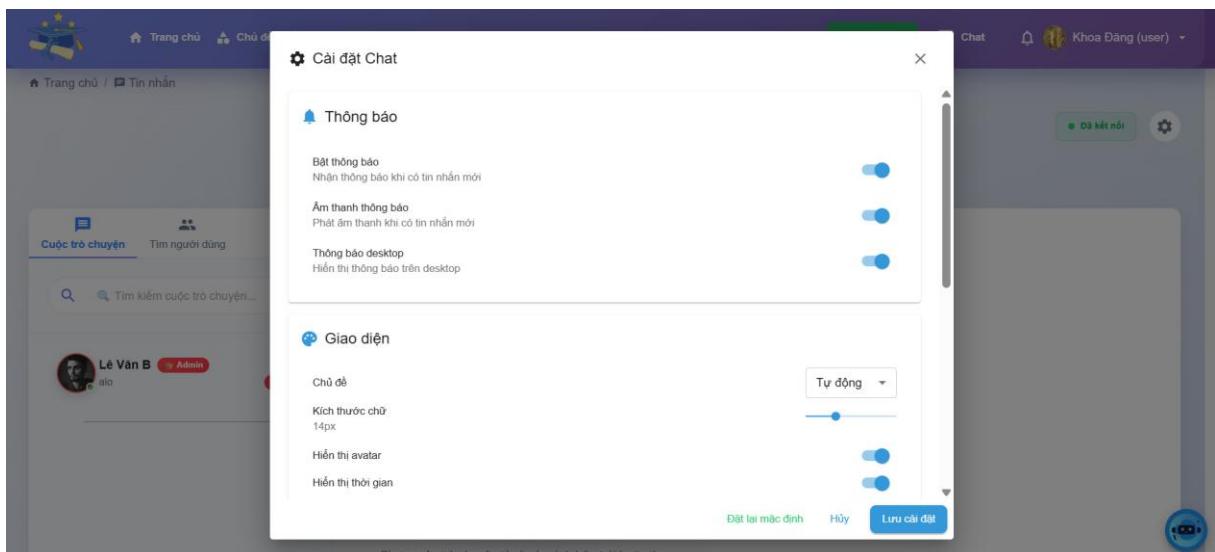
Thành viên có thể xem các danh sách các thành viên khác trong diễn đàn một cách dễ dàng.

#### 4.1.8 Giao diện trang trò chuyện



Hình 4.13 Giao diện trang trò chuyện (a)

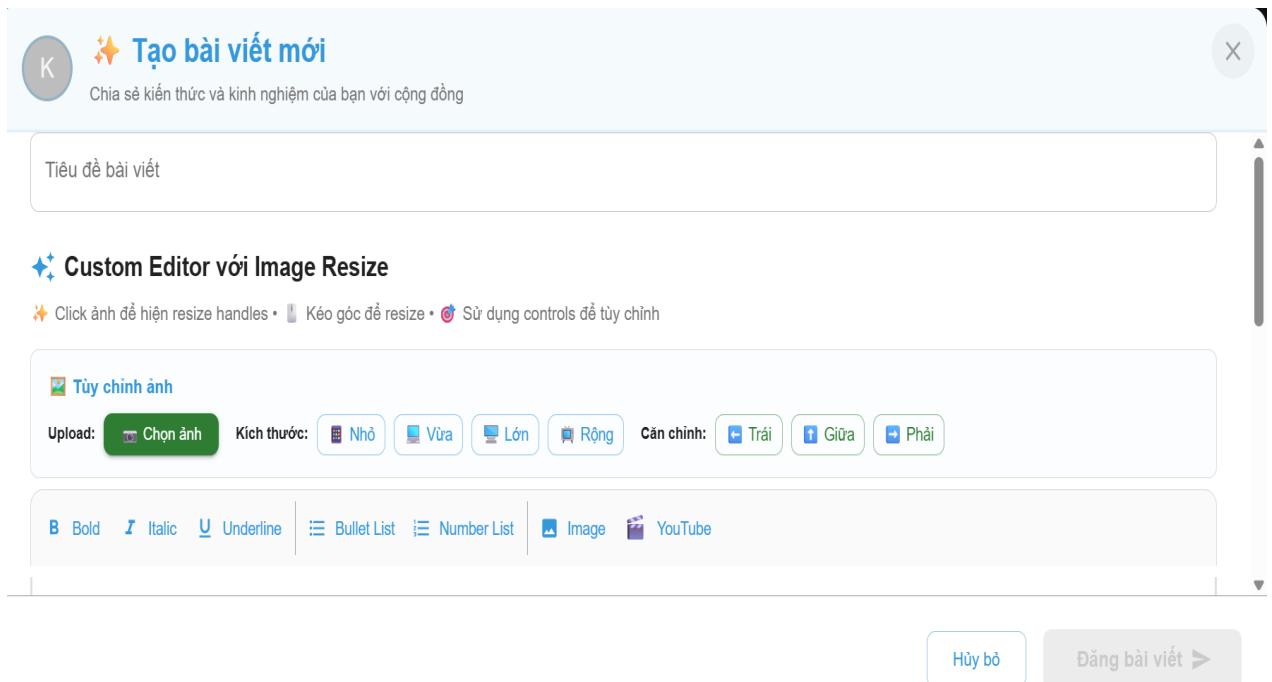
Giao diện trang trò chuyện cho phép các thành viên trò chuyện với nhau. Đảm bảo quyền riêng tư. Được phép nhắn tin trực tiếp với quản trị viên.



Hình 4.14 Giao diện trang trò chuyện (b)

Cho phép cài đặt các chức năng khác nhau trong khung trò chuyện. Tạo nên sự tự do thoải mái.

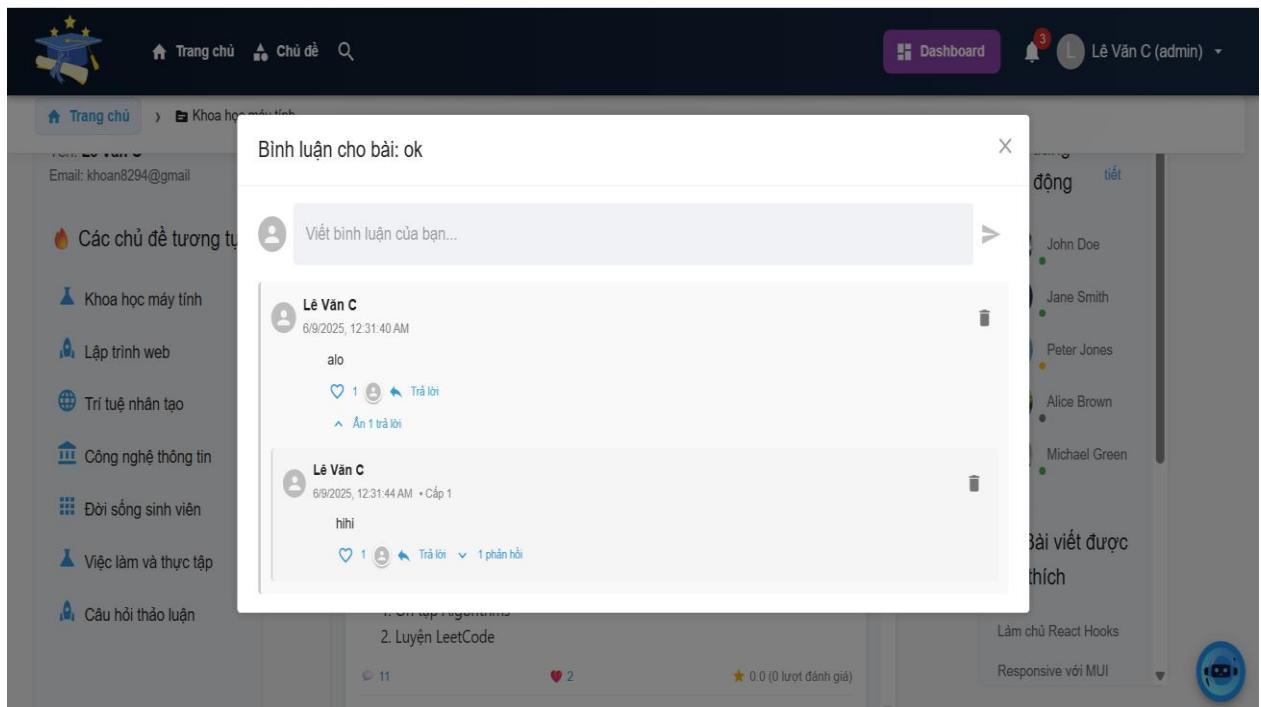
#### 4.1.9 Giao diện trang đăng bài, câu hỏi thảo luận



Hình 4.15 Giao diện trang đăng bài, câu hỏi thảo luận

Trang đăng bài cho phép người dùng đăng các nội dung: các chữ, hình ảnh từ thiết bị, ảnh copy và các link video từ Youtube với các hình thức đa dạng.

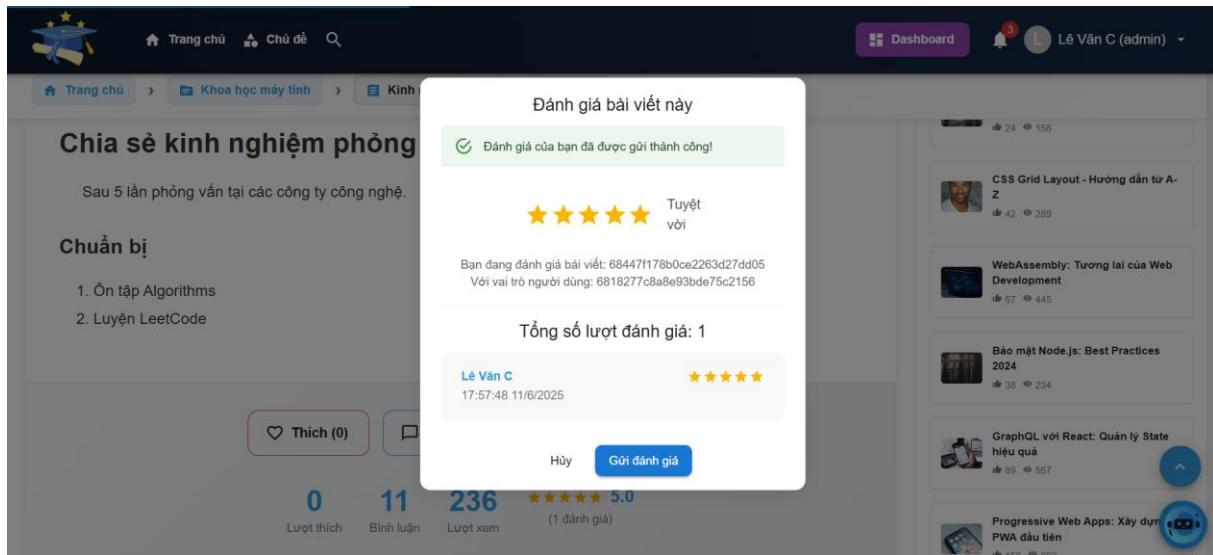
#### 4.1.10 Giao diện trang bình luận



Hình 4.16 Giao diện trang Bình luận

Người dùng để lại các bình luận, phản hồi một cách dễ dàng. Có thêm tương tác lượt thích để thảo luận về các bài viết, câu hỏi thêm sôi nổi và sống động.

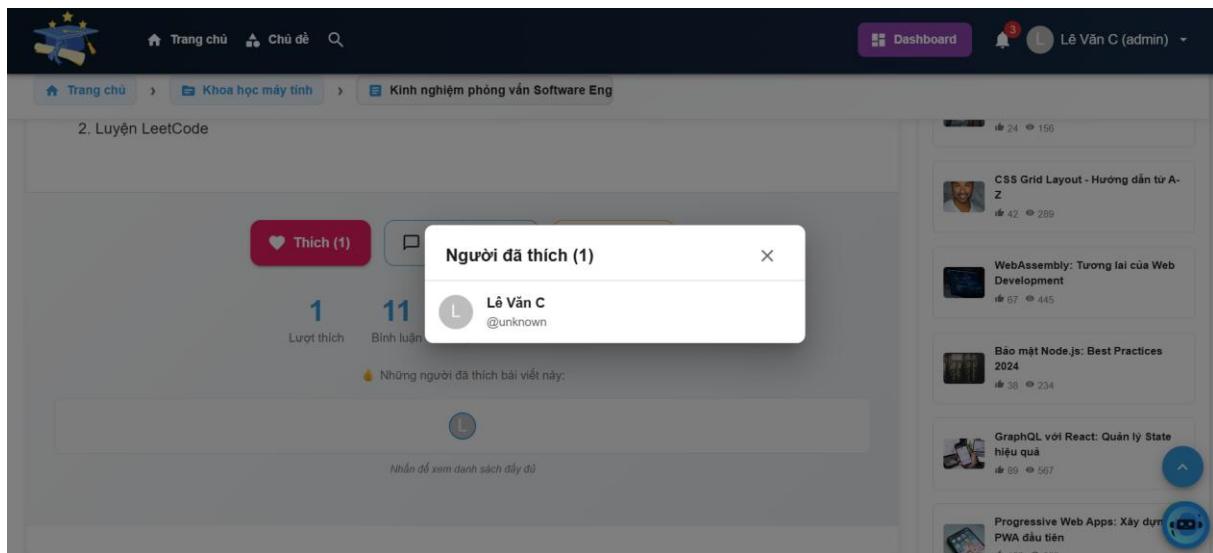
#### 4.1.11 Giao diện trang đánh giá



Hình 4.17 Giao diện trang Đánh giá

Việc đánh giá được tiến hành khi người dùng nhấn vào nút đánh giá bên cạnh bài viết. Đánh giá từ 1 đến 5 sao tương ứng với các cấp độ hài lòng của người đọc.

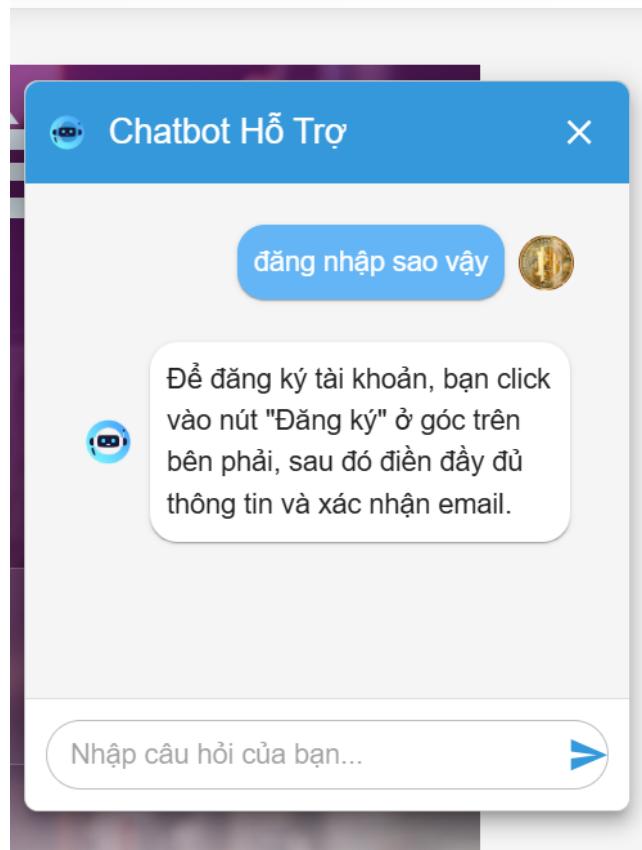
#### 4.1.12 Giao diện trang tương tác lượt thích



Hình 4.18 Giao diện trang tương tác lượt thích

Người đọc có thể tương tác với bài viết bằng lượt thích, qua đó thể hiện được sự quan tâm đến nội dung và tác giả bài viết.

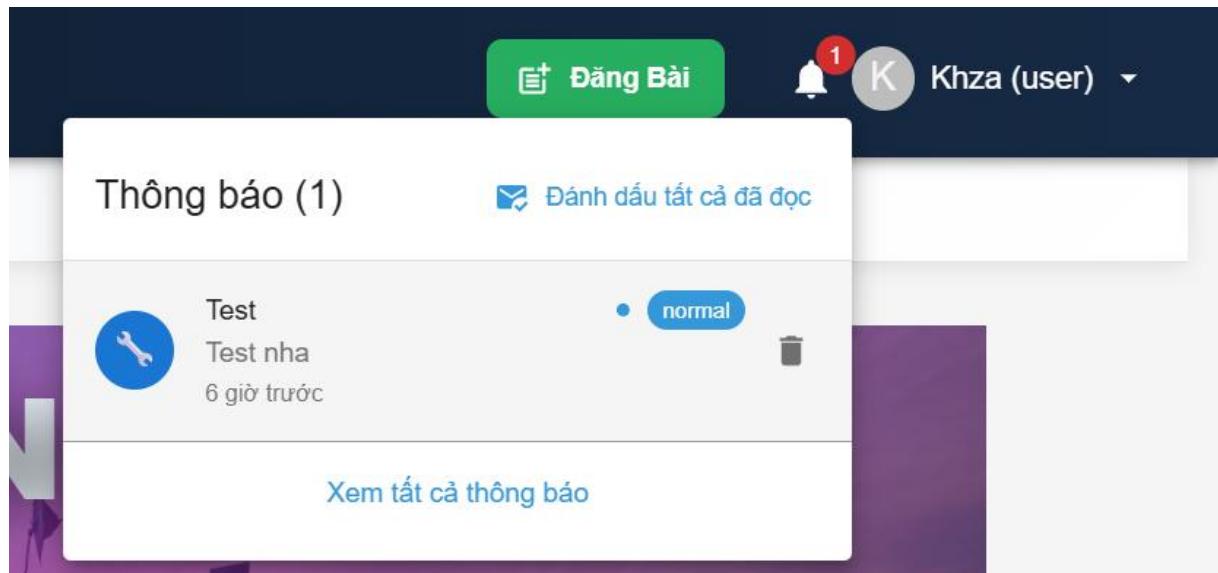
#### 4.1.13 Giao diện trang Chat với ChatBot



Hình 4.19 Giao diện trò chuyện với ChatBot

Người dùng chat với ChatBot để được hướng dẫn, giới thiệu một cách dễ dàng về các quy định, cách đăng bài,...

#### 4.1.14 Giao diện hộp thoại thông báo của người dùng:



Hình 4.20 Giao diện hộp thoại thông báo của người dùng

Hộp thoại sẽ nhận thông báo từ hệ thống như: thông báo về các bình luận, lượt thích, đánh giá... mà người dùng nhận được và cả các thông báo quan trọng đến từ quản trị viên.

#### 4.1.15 Giao diện trang Đăng nhập

The screenshot shows the login interface of a website. At the top, there is a navigation bar with icons for Home, Main content, Search, and user account management. On the right side of the bar are buttons for 'Đăng Nhập' (Login) and 'Đăng Ký' (Register). The main area has a large circular arrow icon above the title 'Đăng nhập'. Below the title, a welcome message reads: 'Chào mừng trở lại! Vui lòng đăng nhập vào tài khoản của bạn'. There are two input fields: 'Email \*' containing 'dznghai121212@gmail.com' and 'Mật khẩu \*' containing '\*\*\*\*\*'. Below these fields are links for 'Quên mật khẩu?' (Forgot password?) and 'Đăng nhập' (Login). A horizontal line with the word 'hoặc' (or) separates this from a 'Đăng nhập bằng Google' (Google sign-in) button. To the right of the input fields, there is a message bubble saying 'Hãy, bạn cần hỗ trợ gì không nè!' and a small video camera icon.

**Hình 4.21 Giao diện trang Đăng nhập**

Trang đăng nhập cho phép người dùng đăng nhập vào hệ thống diễn đàn thông qua việc đã đăng ký trước đó. Người dùng có thể đăng nhập bằng tài khoản Google để thuận tiện hơn trong việc xác thực tài khoản.

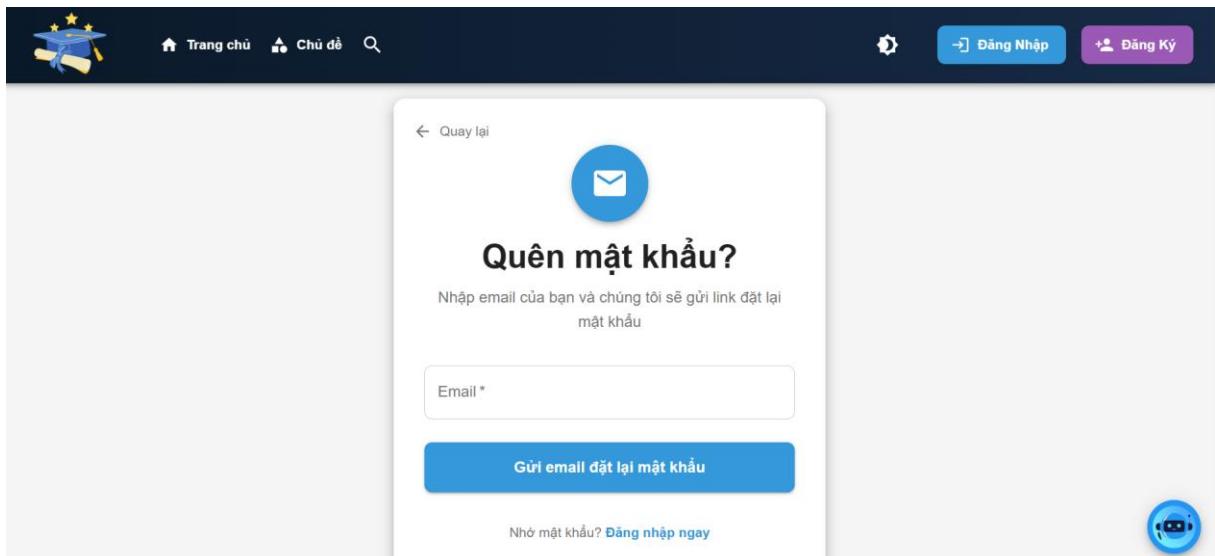
#### 4.1.16 Giao diện trang Đăng ký

The screenshot shows the registration interface of a website. At the top, there is a navigation bar with icons for Home, Main content, Search, and user account management. On the right side of the bar are buttons for 'Đăng Nhập' (Login) and 'Đăng Ký' (Register). The main area features a green circular icon with a plus sign and a person icon. The title 'Đăng ký' is displayed below it, with a subtitle 'Tạo tài khoản mới để tham gia cộng đồng TVU'. There are three input fields: 'Họ tên \*' (Name), 'Email \*' containing 'dznghai121212@gmail.com', and 'Mật khẩu \*' containing '\*\*\*\*\*'. Below these fields is a 'Đăng ký tài khoản' (Register account) button. At the bottom of the form, there is a link 'Đã có tài khoản? Đăng nhập ngay' (Already have an account? Log in now).

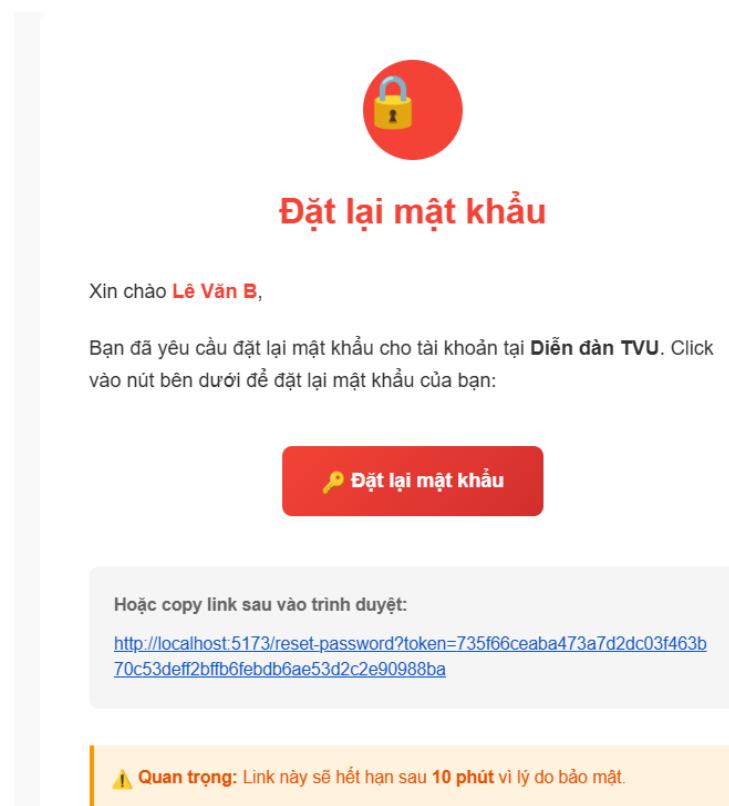
**Hình 4.22 Giao diện trang Đăng ký**

Đăng ký một tài khoản là điều bắt buộc nếu muốn trở thành thành viên của diễn đàn, việc đăng ký cho phép người dùng bắt đầu một hành trình trở thành các sinh viên năng động, sẵn sàng thảo luận về các chủ đề, câu hỏi từ mọi người.

#### 4.1.17 Giao diện trang quên mật khẩu

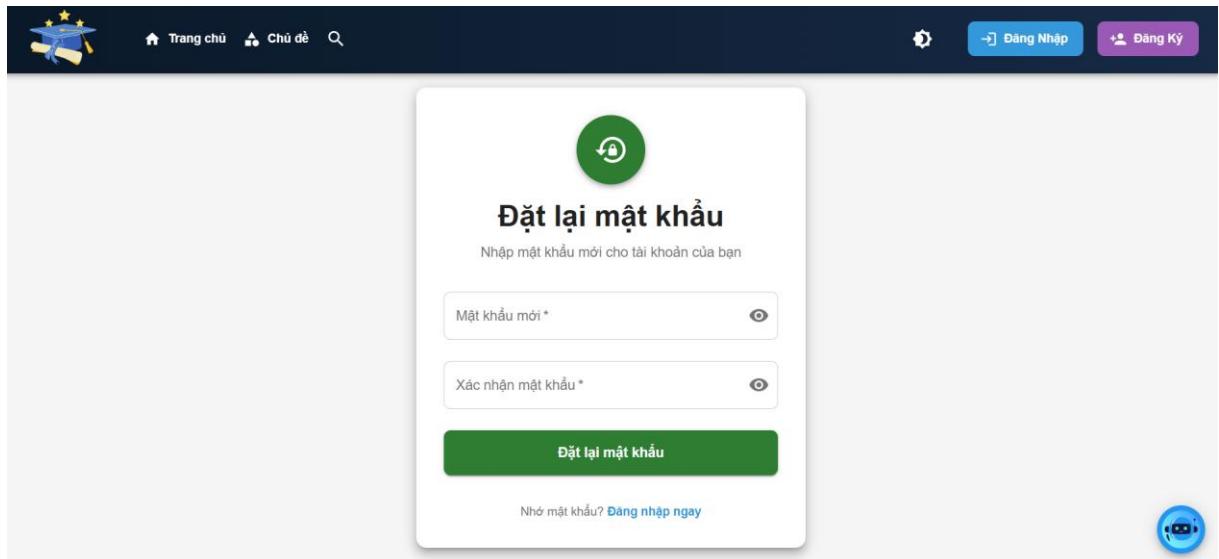


Hình 4.23 Giao diện trang quên mật khẩu



Hình 4.24 Nội dung Email nhận được khi đặt lại mật khẩu

Thành viên diễn đàn khi đã quên mật khẩu có thể điền Email đã đăng ký trước đó trên diễn đàn để hệ thống gửi link chuyển đến trang đăng ký mật khẩu mới.

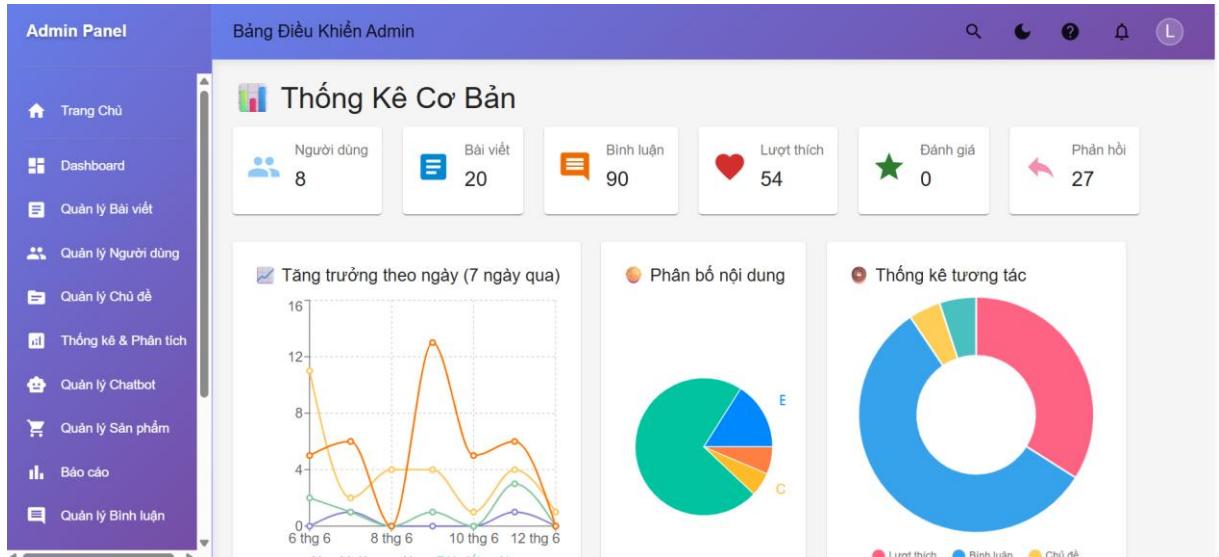


Hình 4.25 Giao diện đặt lại mật khẩu

Sau khi đã nhận được Email xác nhận từ hệ thống diễn đàn, người dùng copy link lên trình duyệt hoặc nhấp vào nút “Đặt lại mật khẩu” để chuyển đến trang trên. Tiến hành đặt lại mật khẩu mới.

## 4.2 Giao diện quản trị

### 4.2.1 Giao diện trang chủ quản trị



Hình 4.26 Giao diện trang admin

Giao diện khi đăng nhập với quyền quản trị là các trang: Thống kê cơ bản các hoạt động như số bài viết, người dùng, bình luận, lượt thích, đánh giá, phản hồi và các trang quản lý chi tiết cho các nội dung nói trên.

#### 4.2.2 Giao diện trang quản lý bài viết

Tiêu đề	Tác giả	Chủ đề	Trạng thái	Ngày tạo	Ngày cập nhật	Hành động
Chào mọi người	dzng	Khoa học máy tính	PENDING	12/06/2025 22:47	13/06/2025 16:55	
alo	Khiza	Khoa học máy tính	PENDING	12/06/2025 17:57	12/06/2025 22:01	
Test Post with Image Uplo...	N/A	N/A	PENDING	12/06/2025 17:34	12/06/2025 17:34	
alo	Lê Văn B	Khoa học máy tính	PENDING	10/06/2025 11:17	12/06/2025 16:11	

Hình 4.27 Giao diện trang Quản lý bài viết

Quản trị viên có thể tạo bài viết mới vào một chủ đề, cập nhật và xóa nếu bài viết vi phạm quy định. Có quyền chuyển bài viết “lạc đề” vào một chủ đề phù hợp.

#### 4.2.3 Giao diện trang quản lý chủ đề

Thứ tự	Tên chủ đề	Danh mục	Trạng thái	Bài viết	Lượt xem	Ngày tạo	Hành động
1	Khoa học máy tính Thảo luận về các chủ đề liên quan đến khoa học máy...	Học tập	Hoạt động	7 bài viết	0	2/5/2025 18:38	

Hình 4.28 Giao diện trang Quản lý chủ đề

Các chủ đề được thống kê chi tiết. Các quyền thêm, cập nhật thông tin vẫn được phép đổi với quản trị viên. Xóa chủ đề nếu thấy có quá ít các hoạt động bài viết bên trong.

#### 4.2.4 Giao diện trang quản lý, huấn luyện ChatBot

Tên Intent	Danh mục	Trạng thái	Sync Status	Training Phrases	Responses	Trigger Count	Success Rate	Thao tác
Tạm biệt goodbye	(greeting)	active	✓	5	3	12	undefined%	
Chủ đề phổ biến popular_topics	(information)	active	✓	5	1	21	undefined%	

Hình 4.29 Giao diện trang Quản lý, huấn luyện ChatBot

Trang này cho phép quản trị viên quản lý các câu hỏi, trả lời của ChatBot qua đó huấn luyện để giao tiếp với người dùng. Ghi nhận các câu hỏi phổ biến của người dùng khi họ nói chuyện với ChatBot qua đó huấn luyện để có giải đáp đa dạng hơn các thắc mắc của người dùng khi họ tham gia diễn đàn.

#### 4.2.5 Giao diện trang Thống kê, phân tích

Người dùng: 8  
+5 mới
Bài viết: 20  
+19 mới
Chủ đề: 7
Bình luận: 90  
+90 mới
Lượt thích: 54

XU HƯỚNG TĂNG TRƯỞNG

Xu hướng tăng trưởng theo thời gian

TỶ LỆ TĂNG TRƯỞNG

Người dùng: +66.67%  
Bài viết: +1800%

HOẠT ĐỘNG TRONG 30 NGÀY QUA

Người dùng hoạt động: 0  
Bài viết mới: 19  
Bình luận mới: 90

Hình 4.30 Giao diện trang Thống kê, phân tích

Trang Thống kê, phân tích sẽ tổng hợp các thông tin chi tiết hơn về tỉ lệ tăng trưởng của số người dùng, bài viết, đánh giá, bình luận và các lượt thích qua đó có cái nhìn chi tiết hơn cho quản trị viên trong việc bảo trì và nâng cấp hệ thống.

#### 4.2.6 Giao diện trang quản lý bình luận

TÁT CẢ (90)	CHỜ DUYỆT (86)	ĐÃ DUYỆT (4)
<input type="checkbox"/> Người dùng	<input type="checkbox"/> Nội dung	<input type="checkbox"/> Bài viết
	dzng 13/6/2025	máy tính mua ở đâu vây bạn
		Chào mọi người
		<span>0</span> <span>0</span> <span>Chờ duyệt</span> <span>✓</span> <span>edit</span> <span>eye</span> <span>trash</span>
	Lê Văn B 0345476413 zalo nha em	

Hình 4.31 Giao diện trang Quản lý Bình luận

Các bình luận và các phản hồi luôn được cập nhật để quản trị viên khảo sát các bài viết, xem sự quan tâm của người dùng về bài viết đó.

#### 4.2.7 Giao diện trang cá nhân của quản trị viên

Lê Văn B

Quản trị viên Đã xác thực

Quản trị viên hệ thống diễn đàn sinh viên Trà Vinh University. Chuyên quản lý nội dung, người dùng và phân tích dữ liệu.

edit CHỈNH SỬA

**Thống kê nhanh**

Thời gian online  
8 giờ 30 phút hôm nay

Hình 4.32 Giao diện trang cá nhân của quản trị viên (a)

Trang cá nhân của quản trị viên cho phép chỉnh sửa các thông tin như: ảnh đại diện, họ tên, địa chỉ, số điện thoại,... để từ đó có thể cung cấp cho trang người dùng nếu họ muốn liên hệ trực tiếp với quản trị viên.

The screenshot shows the Admin Panel interface with a sidebar on the left containing navigation links such as Trang Chủ, Dashboard, Quản lý Bài viết, Quản lý Người dùng, Quản lý Chủ đề, Thông kê & Phân tích, Quản lý Chatbot, Quản lý Sản phẩm, and Báo cáo. The main content area is titled 'Bảng Điều Khiển Admin' and displays two sections: 'Thống kê nhanh' and 'Thông tin chi tiết'. The 'Thống kê nhanh' section shows online time (8 giờ 30 phút hôm nay) and activity (25 tác vụ hoàn thành). The 'Thông tin chi tiết' section provides detailed information about the user: Họ và tên (Lê Văn B), Email (khoan8294@gmail.com), Địa chỉ (Trà Vinh University, Trà Vinh), and Giới thiệu (Quản trị viên hệ thống diễn đàn sinh viên Tra Vinh University).

**Hình 4.33 Giao diện trang cá nhân của quản trị viên (b)**

Các thông tin về họ tên, Email, số điện thoại, địa chỉ,... được hiện một cách chi tiết cho quản trị viên quản lý.

The screenshot shows the Admin Panel interface with the same sidebar as before. The main content area is titled 'Bảng Điều Khiển Admin' and displays two sections: 'Thông tin hệ thống' and 'Cài đặt bảo mật'. The 'Thông tin hệ thống' section shows Ngày tham gia (1/1/2024) and Lần đăng nhập cuối (09:44:39 12/6/2025). The 'Cài đặt bảo mật' section shows two items: Xác thực hai yếu tố (Đã bật - Bảo vệ tài khoản với mã OTP) and Email đã xác thực (Địa chỉ email đã được xác minh - Đã xác thực). There is also a 'ĐỔI MẬT KHẨU' button at the bottom.

**Hình 4.34 Giao diện trang cá nhân của quản trị viên (c)**

Thông báo rõ ràng về việc xác thực chật chẽ tài khoản. Đảm bảo tính bảo mật cho tài khoản quản trị.

#### 4.2.8 Giao diện trang quản lý các chủ đề thịnh hành và các bài viết nổi bật

Hình 4.35 Giao diện trang quản lý chủ đề thịnh hành, bài viết nổi bật

Các chủ đề thịnh hành và các chủ đề nổi bật được quản trị viên quyền lựa chọn, theo các tiêu chí như: chủ đề có nhiều bài viết và các bài viết có nhiều tương tác.

#### 4.2.9 Giao diện trang quản lý dữ liệu

Hình 4.36 Giao diện trang quản lý dữ liệu

Quản trị viên quản lý các dữ liệu dưới dạng file Excel, có quyền nhập (import) hoặc xuất dữ liệu (export) để dễ dàng quản lý.

#### 4.2.10 Giao diện trang quản lý thông báo

The screenshot shows the Admin Panel interface with a sidebar on the left containing various management links. The main area is titled 'Bảng Điều Khiển Admin' and displays a summary of notifications: 63 Total notifications, 22 Unread, 41 Read, and 36 Today. Below this are two buttons: 'GỬI THÔNG BÁO CHUNG' (Send General Notification) and 'GỬI THÔNG BÁO CÁ NHÂN' (Send Personal Notification). A table lists notifications with columns for Type, Subject, Recipient, Priority, Status, Creation Date, and Action. Two notifications are listed:

Loại	Tiêu đề	Người nhận	Độ ưu tiên	Trạng thái	Ngày tạo	Thao tác
new_comment_added	Bình luận mới dzng đã bình luận trong bài viết "Chào mọi người!"...	Administrator	high	Chưa đọc	13/6/2025	
feature_update	Test Test nha...	dzng	normal	Chưa đọc	13/6/2025	

Hình 4.37 Giao diện trang quản lý thông báo

Quản trị viên có quyền quản trị các thông báo trong hệ thống. Có thể gửi thông báo cho tất cả thành viên hoặc riêng thành viên nào đó.

#### 4.2.11 Giao diện hộp thoại (nhận thông báo) của quản trị viên

The screenshot shows a mobile notification inbox titled 'Thông báo (0)'. It displays a single notification: 'Bình luận mới' from 'dzng' with the subject 'Chào mọi người!' and priority 'high'. The message was sent 4 hours ago. Below the message is a blue button labeled 'XEM TẤT CẢ THÔNG BÁO' (View all notifications).

Hình 4.38 Giao diện hộp thoại thông báo

Các hoạt động như người dùng vừa tạo bài viết, các hoạt động như người dùng vừa đăng ký thành công,... sẽ được thông báo cho quản trị viên. Nắm bắt các hoạt động nhằm bảo đảm các hoạt động trong diễn đàn luôn theo quy định đã đưa ra.

#### 4.2.12 Giao diện trang quản lý quy định diễn đàn

Hình 4.39 Giao diện trang quản lý quy định diễn đàn

Quản trị viên có thể điều chỉnh lại quy định của diễn đàn một cách dễ dàng. Đảm bảo các hoạt động luôn diễn ra một cách có kỷ luật và đảm bảo sự thảo luận, hỗ trợ lẫn nhau giữa các thành viên một cách dễ dàng.

## CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

### 5.1 Kết luận

#### Về kiến thức:

Trong quá trình thực hiện đề tài "Diễn đàn trực tuyến cho sinh viên trường Đại học Trà Vinh", tôi đã tích lũy được nhiều kiến thức bổ ích về cả lý thuyết lẫn thực hành phát triển ứng dụng web hiện đại.

Tôi đã hiểu rõ hơn về kiến trúc web theo mô hình client-server, biết cách xây dựng API tuân theo chuẩn RESTful để giao tiếp giữa frontend (React) và backend (Express). Việc sử dụng Express giúp tôi nắm được quy trình xử lý request-response, middleware, routing và cơ chế bảo mật cơ bản như xác thực bằng JWT.

Ở phía frontend, React giúp tôi học cách tổ chức giao diện theo component, sử dụng state, props, và quản lý luồng dữ liệu hiệu quả. Tôi cũng đã biết cách kết nối frontend với backend bằng cách sử dụng các thư viện như Axios để gọi API, hiển thị dữ liệu theo thời gian thực và phản hồi tương tác của người dùng.

#### Về thực hành:

Trong quá trình triển khai, tôi đã xây dựng thành công một hệ thống diễn đàn có thể đáp ứng các chức năng cơ bản sau:

- Đăng ký, đăng nhập, xác thực và quản lý người dùng.
- Đăng bài viết, phản hồi bình luận và phân loại bài viết theo chủ đề.
- Chức năng tìm kiếm bài viết theo từ khóa và chủ đề.
- Quản trị viên có thể duyệt bài, xóa bài, quản lý người dùng và chủ đề.
- Giao diện dễ sử dụng, có khả năng phản hồi tốt.

Ngoài ra, tôi cũng đã kiểm thử các chức năng quan trọng, đảm bảo tính ổn định của hệ thống và đáp ứng được các yêu cầu ban đầu. Quá trình thực hành này giúp tôi làm quen với quy trình phát triển phần mềm thực tế, từ phân tích yêu cầu, thiết kế hệ thống, lập trình, kiểm thử cho đến triển khai.

### 5.2 Hướng phát triển

Mặc dù hệ thống đã hoàn thiện về mặt chức năng cơ bản, nhưng vẫn còn nhiều tiềm năng để mở rộng trong tương lai. Một số hướng phát triển có thể kể đến như:

- Phát triển phiên bản mobile: Xây dựng ứng dụng diễm đàm dành cho điện thoại, hỗ trợ sinh viên sử dụng thuận tiện hơn qua Android/iOS.
- Cải thiện trải nghiệm người dùng (UI/UX): Nâng cấp giao diện hiện đại, trực quan hơn, phù hợp với thói quen sử dụng của sinh viên, đặc biệt là trên thiết bị di động.
- Kết nối với hệ thống sinh viên của nhà trường: Hỗ trợ đăng nhập bằng tài khoản sinh viên, đồng bộ dữ liệu và phân quyền theo khoa/lớp.

Với những định hướng trên, diễm đàm trực tuyến có thể trở thành một công cụ hữu ích hỗ trợ sinh viên trong việc trao đổi học tập, chia sẻ kiến thức và xây dựng cộng đồng sinh viên tích cực, năng động trong môi trường đại học.

## **PHỤ LỤC**

## **DANH MỤC TÀI LIỆU THAM KHẢO**

### **Sách/ Tài liệu**

- [1] Đoàn Phước Miền, Phạm Thị Trúc Mai (2014), “Thiết kế và lập trình Web”, Trường Đại học Trà Vinh.
- [2] Hà Thị Thúy Vi (2013), “Cơ sở dữ liệu”, Trường Đại học Trà Vinh.
- [3] Phạm Minh Đương (2014), “Phân tích và thiết kế hệ thống thông tin”, Trường Đại học Trà Vinh.

### **Website**

- [4] 200Lab (n.d.), “Giới thiệu về React: Tổng quan cơ bản về React”, [Online]. Available: <https://200lab.io/blog/reactjs-la-gi/>. [Accessed: 12/05/2025].
- [5] 200Lab (n.d.), “Giới thiệu cơ bản về ExpressJS: Khái quát điểm về ExpressJS”, [Online]. Available: <https://200lab.io/blog/expressjs-la-gi/>. [Accessed: 16/05/2025].
- [6] 200Lab (n.d.), “NodeJS là gì? Tổng quan cơ bản về NodeJS”, [Online]. Available: <https://200lab.io/blog/nodejs-la-gi/>. [Accessed: 08/12/2024].
- [7] 200Lab (n.d.), “Tìm hiểu về JavaScript: Giới thiệu tổng quan về JavaScript”, [Online]. Available: <https://200lab.io/blog/javascript-la-gi/>. [Accessed: 21/05/2024].
- [8] 200Lab (n.d.), “Material UI là gì? Tổng quan về thư viện Material UI trong React”, [Online]. Available: <https://200lab.io/blog/material-ui-la-gi/>. [Accessed: 03/05/2025].
- [9] FPT Cloud (n.d.), “Mô hình MVC là gì? Tìm hiểu về mô hình MVC”, [Online]. Available: <https://cloud.z.com/vn/news/mvc/>. [Accessed: 12/05/2025].
- [10] ITviec (n.d.), “Tìm hiểu về JavaScript: Tổng quan cơ bản về JavaScript”, [Online]. Available: <https://itviec.com/blog/javascript-la-gi/>. [Accessed: 18/05/2025].
- [11] 200Lab (n.d.), “Tổng quan về MongoDB: MongoDB có ưu, nhược điểm gì?”, [Online]. Available: <https://200lab.io/blog/mongodb-la-gi/>. [Accessed: 12/05/2025].

[12] Điện Thoại Vui (n.d.), “Chatbot AI là gì? Những điều bạn cần biết về chatbot AI”, [Online]. Available: <https://dienthoaivui.com.vn/chatbot-ai>. [Accessed: 18/05/2025].

[13] Dead Simple Chat (2024), “WebSockets and NodeJs: Real time chat app,” [Online]. Available: <https://deadsimplechat.com/blog/websockets-and-nodejs-real-time-chat-app/>. [Accessed: 18/05/2025].