BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA/VIỆN: CÔNG NGHỆ THÔNG TIN**

A blue and black logo

AI-generated content may be incorrect.

**ĐỒ ÁN TỐT NGHIỆP**

**TÊN ĐỒ ÁN: HỆ THỐNG GỢI Ý CHO DIỄN ĐÀN TRAO ĐỔI HỌC TẬP**

**Giảng viên hướng dẫn: Ths.Hà Thị Thanh Ngà**

**Sinh viên thực hiện: Ngô Hoàng Vũ**

**Mã số sinh viên: 63131717**

Khánh Hòa – 6/6/2025

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA/VIỆN: CÔNG NGHỆ THÔNG TIN**

A blue and black logo

AI-generated content may be incorrect.

**ĐỒ ÁN TỐT NGHIỆP**

**TÊN ĐỒ ÁN: HỆ THỐNG GỢI Ý CHO DIỄN ĐÀN TRAO ĐỔI HỌC TẬP**

**Giảng viên hướng dẫn: Ths.Hà Thị Thanh Ngà**

**Sinh viên thực hiện: Ngô Hoàng Vũ**

**Mã số sinh viên: 63131717**

Khánh Hòa – 6/6/2025

**A screenshot of a computer

AI-generated content may be incorrect.**

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa/Viện**: Công nghệ thông tin

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ ĐỒ ÁN / KHÓA LUẬN TỐT NGHIỆP  
*(Dùng cho CBHD và nộp cùng báo cáo ĐA/KLTN của sinh viên)***

Tên đề tài: Hệ thống gợi ý cho diễn đàn trao đổi học tập

Chuyên ngành: Kỹ thuật phần mềm

Họ và tên sinh viên: Ngô Hoàng Vũ Mã sinh viên: 63131717

Người hướng dẫn (học hàm, học vị, họ và tên): ThS.Hà Thị Thanh Ngà

Cơ quan công tác :

**Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số**  **(%)** | **Mô tả mức chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| Xây dựng đề cương nghiên cứu | 10 |  |  |  |  |  |
| Tinh thần và thái độ làm việc | 10 |  |  |  |  |  |
| Kiến thức và kỹ năng làm việc | 10 |  |  |  |  |  |
| Nội dung và kết quả đạt được | 40 |  |  |  |  |  |
| Kỹ năng viết và trình bày báo cáo | 30 |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KLTN):

…………………………………………………………………………………………….…………………………………………………………………………………………….…………………………………………………………………………………………………………….………………………………………………………………………………………Đồng ý cho sinh viên: Được bảo vệ:  Không được bảo vệ: 

*Khánh Hòa, ngày…….tháng…….năm………* **Cán bộ hướng dẫn** *(Ký và ghi rõ họ tên)*

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**Khoa/Viện: Công nghệ thông tin**

**PHIẾU CHẤM ĐIỂM ĐỒ ÁN/ KHÓA LUẬN TỐT NGHIỆP**

***(Dành cho cán bộ chấm phản biện)***

Tên đề tài: Hệ thống gợi ý cho diễn đàn trao đổi học tập

Chuyên ngành: Kỹ thuật phần mềm

Họ và tên sinh viên: Ngô Hoàng Vũ Mã sinh viên: 63131717

Người phản biện (học hàm, học vị, họ và tên):

Cơ quan công tác:

**I. Phần đánh giá và cho điểm của người phản biện (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số**  **(%)** | **Mô tả mức chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| Hình thức bản thuyết minh | 20 |  |  |  |  |  |
| Nội dung bản  thuyết minh | 30 |  |  |  |  |  |
| Mức độ trích dẫn và sao chép | 20 |  |  |  |  |  |
| Kết quả nghiên cứu | 30 |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Kết luận:**

Đồng ý cho sinh viên: Được bảo vệ:  Không được bảo vệ: 

*Khánh Hòa, ngày…….tháng………năm………..* **Cán bộ chấm phản biện** *(Ký và ghi rõ họ tên)*

**LỜI CAM ĐOAN**

Tôi xin cam đoan rằng toàn bộ kết quả nghiên cứu trong đề tài “Hệ thống gợi ý cho diễn đàn trao đổi học tập” là công trình nghiên cứu độc lập do cá nhân tôi thực hiện. Các nội dung trong đề tài chưa từng được công bố dưới bất kỳ hình thức nào trong các công trình khoa học hoặc báo cáo nghiên cứu khác trước đây. Tôi chịu hoàn toàn trách nhiệm trước pháp luật về tính chính xác và trung thực của các thông tin và số liệu được trình bày trong đề tài này.

# LỜI CẢM ƠN

Trong suốt thời gian thực hiện đề tài, tôi đã nhận được sự giúp đỡ của quý phòng ban trường Đại học Nha Trang, Khoa Công nghệ Thông tin nói chung và Bộ môn Kỹ thuật phần mềm nói riêng đã tạo điều kiện tốt nhất cho tôi được hoàn thành đề tài. Đặc biệt là sự hướng dẫn tận tình của ThS. Hà Thị Thanh Ngà đã giúp tôi hoàn thành tốt đề tài. Qua đây, tôi xin gửi lời cảm ơn sâu sắc đến sự giúp đỡ này.

Xin chân thành cảm ơn các quý thầy cô trong bộ môn Kỹ thuật phần mềm trong thời gian qua đã truyền đạt và trang bị cho em kiến thức giúp em hoàn thành tốt đề tài tốt nghiệp.

Cuối cùng tôi xin gửi lời cảm ơn chân thành đến gia đình và tất cả bạn bè đã giúp đỡ, động viên tôi trong suốt quá trình học tập và thực hiện đề tài.

Tôi xin chân thành cảm ơn!

Khánh Hòa, ngày tháng năm 2025

Sinh viên thực hiện

# TÓM TẮT KHÓA LUẬN

Đồ án này tập trung nghiên cứu và phát triển một hệ thống gợi ý sử dụng các kỹ thuật học máy (machine learning) để xử lý dữ liệu dạng bảng. Mục tiêu nhằm cá nhân hóa trải nghiệm người dùng trong các diễn đàn học tập thông qua việc đề xuất nội dung phù hợp với hành vi và đặc điểm cá nhân.  
Đồ án trình bày tổng quan các hệ thống gợi ý hiện nay, cơ sở lý thuyết về machine learning, các mô hình được áp dụng như Content-Based, Collaborative Filtering, và sử dụng thư viện scikit-learn, pandas, và ReactJS để xây dựng giao diện người dùng.  
Kết quả thử nghiệm trên tập dữ liệu giả lập cho thấy hệ thống hoạt động hiệu quả trong việc đề xuất nội dung phù hợp, nâng cao mức độ tương tác và cá nhân hóa cho người học.  
Khóa luận cũng chỉ ra một số hạn chế và đề xuất hướng phát triển tiếp theo nhằm nâng cao độ chính xác và khả năng mở rộng của hệ thống.

# MỤC LỤC

[LỜI CAM ĐOAN iii](#_Toc200193305)

[LỜI CẢM ƠN iv](#_Toc200193306)

[TÓM TẮT KHÓA LUẬN vi](#_Toc200193307)

[MỤC LỤC vii](#_Toc200193308)

[DANH MỤC HÌNH x](#_Toc200193309)

[DANH MỤC BẢNG xi](#_Toc200193310)

[LỜI MỞ ĐẦU 1](#_Toc200193311)

[CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ 2](#_Toc200193312)

[1.1. Tính cấp thiết của đề tài 2](#_Toc200193313)

[1.2. Mục tiêu đề tài 2](#_Toc200193314)

[1.3. Giải pháp thực hiện 2](#_Toc200193315)

[1.4. Đối tượng và phạm vi nghiên cứu 3](#_Toc200193316)

[1.5. Ý nghĩa lí luận và thực tiễn 3](#_Toc200193317)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc200193318)

[2.1. Ngôn Ngữ JavaScript 4](#_Toc200193319)

[2.1.1. Giới thiệu 4](#_Toc200193320)

[2.1.2. Vai trò 5](#_Toc200193321)

[2.1.3. Các yếu tố chính của JavaScript 5](#_Toc200193322)

[2.2 React JS 8](#_Toc200193323)

[2.3. Kiến trúc Component-Based 9](#_Toc200193324)

[2.4. Express 11](#_Toc200193325)

[2.5. Cơ sở dữ liệu 12](#_Toc200193326)

[2.6. Cơ sở dữ liệu NoSQL 13](#_Toc200193327)

[2.6.1. Khái niệm tổng quan 13](#_Toc200193328)

[2.6.2. Phân loại chính 13](#_Toc200193329)

[2.6.3. Ưu điểm 14](#_Toc200193330)

[2.6.4. Hạn chế 14](#_Toc200193331)

[2.6.5. Ứng dụng thực tế 14](#_Toc200193332)

[2.7. MongoDB 14](#_Toc200193333)

[2.8. Trí tuệ nhân tạo (Artificial Intelligence – AI) 15](#_Toc200193334)

[2.8.1. Vai trò của AI trong hệ thống gợi ý 15](#_Toc200193335)

[2.8.2. Kỹ thuật AI áp dụng trong đề tài 16](#_Toc200193336)

[2.8.3. Lợi ích khi tích hợp AI vào hệ thống 16](#_Toc200193337)

[2.9. Các loại AI và thuật toán thông dụng 16](#_Toc200193338)

[2.9.1. Các loại AI 16](#_Toc200193339)

[2.9.2. Các thuật toán AI thông dụng 17](#_Toc200193340)

[2.10. Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) trong hệ thống gợi ý 19](#_Toc200193341)

[2.10.1. Chuyển đổi văn bản thành dữ liệu số 19](#_Toc200193342)

[2.10.2. Cách đo lường mức độ tương đồng giữa hai bài viết 20](#_Toc200193343)

[2.10.3. Tích hợp NLP vào hệ thống gợi ý 20](#_Toc200193344)

[2.11. Thư viện xử lý ngôn ngữ tự nhiên - Natural 20](#_Toc200193345)

[2.11.1. Giới thiệu về thư viện Natural 20](#_Toc200193346)

[2.11.2. Các chức năng quan trọng của thư viện Natural 21](#_Toc200193347)

[2.12. Recommendation Service trong hệ thống gợi ý 22](#_Toc200193348)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 23](#_Toc200193349)

[3.1 Mô hình hóa yêu cầu 23](#_Toc200193350)

[3.1.1. Phân tích các tác nhân 23](#_Toc200193351)

[3.1.2 Biểu đồ UseCase 23](#_Toc200193352)

[3.2. Mô hình hóa hành vi 41](#_Toc200193355)

[3.2.1 Biều đồ hoạt động 41](#_Toc200193356)

[3.2.2 Biểu đồ tuần tự đăng nhập 46](#_Toc200193357)

[3.3 Mô hình hóa cấu trúc 50](#_Toc200193358)

[CHƯƠNG 4 : XÂY DỰNG CHƯƠNG TRÌNH 51](#_Toc200193359)

[4.1. Bảng phân tích cấu trúc các model dữ liệu 51](#_Toc200193360)

[4.2. Giao diện chương trình 53](#_Toc200193361)

[CHƯƠNG 5 : KẾT QUẢ ĐẠT ĐƯỢC 63](#_Toc200193362)

[5.1. Những thuận lợi và thách thức trong quá trình triển khai 63](#_Toc200193363)

[5.2. Các hạn chế của hệ thống 63](#_Toc200193364)

[KẾT LUẬN 64](#_Toc200193365)

[TÀI LIỆU THAM KHẢO 66](#_Toc200193366)

# DANH MỤC HÌNH

[Hình 2.2. Hoạt động của JSX (Nguồn ảnh: simplilearn.com) 9](#_Toc200033836)

[Hình 2.3. Virtual DOM của ReactJS (Nguồn ảnh: simplilearn.com) 10](#_Toc200033837)

[Hình 2.4. Mô Hình Cơ Sở Dữ Liệu 12](#_Toc200033838)

[Hình 2.5. MongODB 14](#_Toc200033839)

[Hình 3.1. Use case tổng quan 24](#_Toc200033840)

[Hình 3.2. Biểu đồ hoạt động đăng nhập 42](#_Toc200033873)

[Hình 3.3. Biểu đồ hoạt động Tạo và tương tác bài viết 43](#_Toc200033874)

[Hình 3.4. Biểu đồ hoạt động: Quản lý nhóm 44](#_Toc200033875)

[Hình 3.5. Biểu đồ hoạt động quản lý nội dung 45](#_Toc200033876)

[Hình 3.6. Biểu đồ tuần tự đăng nhập của khách hàng 46](#_Toc200033877)

[Hình 3.7. Biểu đồ tuần tự tương tác người dùng 46](#_Toc200033878)

[Hình 3.8. Biểu đồ tuần tự quản lý nhóm thảo luận 47](#_Toc200033879)

[Hình 3.9. Biểu đồ tuần tự quản lý nội dung 48](#_Toc200033880)

[Hình 3.10. Biểu đồ tuần tự quản lý người dùng 49](#_Toc200033881)

[Hình 3.11. Biểu đồ lớp phân tích 50](#_Toc200033882)

[Hình 4.1. Trang đăng kí 54](#_Toc200033883)

[Hình 4.2. Trang đăng nhập 54](#_Toc200033891)

[Hình 4.3. Trang reset mật khẩu 55](#_Toc200033899)

[Hình 4.4. Trang chủ 55](#_Toc200033907)

[Hình 4.5. Trang game 56](#_Toc200033915)

[Hình 4.6. Trang tạo bài viết 57](#_Toc200033923)

[Hình 4.7. Trang tin nhắn 57](#_Toc200033931)

[Hình 4.8. Trang nhóm 58](#_Toc200033932)

[Hình 4.9. Trang bạn bè 58](#_Toc200033940)

[Hình 4.10. Trang profile 59](#_Toc200033941)

[Hình 4.11. Trang cài đặt 59](#_Toc200033949)

[Hình 4.12. Dashboard 60](#_Toc200033950)

[Hình 4.13. Quản lí người dùng 61](#_Toc200033951)

[Hình 4.14. Quản lí bài viết 61](#_Toc200033952)

[Hình 4.15. Quản lí điểm 62](#_Toc200033953)

[Hình 4.16. Quản lí nhóm 62](#_Toc200033954)

# DANH MỤC BẢNG

[Bảng 3.1. Bảng mô tả quản lý người dùng-Admin 24](#_Toc200033683)

[Bảng 3.2. Bảng mô tả quản lý người dùng-User 25](#_Toc200033700)

[Bảng 3.3. Bảng mô tả đăng nhập / đăng kí 27](#_Toc200033702)

[Bảng 3.4. Bảng mô tả quản lý nội dung-Admin 29](#_Toc200033703)

[Bảng 3.5. Bảng mô tả quản lý nội dung-User 30](#_Toc200033720)

[Bảng 3.6. Bảng mô tả quản lý nhóm thảo luận-Admin 31](#_Toc200033721)

[Bảng 3.7. Bảng mô tả quản lý nhóm thảo luận-User 33](#_Toc200033738)

[Bảng 3.8. Bảng mô tả Đánh giá nội dung-Admin 34](#_Toc200033739)

[Bảng 3.9. Bảng mô tả Đánh giá nội dung-User 35](#_Toc200033756)

[Bảng 3.10. Bảng mô tả tìm kiếm nội dung 37](#_Toc200033757)

[Bảng 3.11 Bảng mô tả quản Lý Bài Viết / Câu Hỏi-Admin 37](#_Toc200033758)

[Bảng 3.12. Bảng mô tả quản Lý Bài Viết / Câu Hỏi-User 39](#_Toc200033775)

[Bảng 3.13. Bảng mô tả nhận thông báo 41](#_Toc200033776)

[Bảng 4.1. Bảng User 51](#_Toc200033777)

[Bảng 4.3. Bảng follow 51](#_Toc200033778)

[Bảng 4.4. Friendship 51](#_Toc200033779)

[Bảng 4.5. Bảng Group 52](#_Toc200033780)

[Bảng 4.6. Bảng Message 52](#_Toc200033781)

[Bảng 4.7. Bảng Notification 52](#_Toc200033782)

[Bảng 4.8. Bảng UserAcivity 52](#_Toc200033783)

[Bảng 4.9. Bảng Post 53](#_Toc200033817)

# LỜI MỞ ĐẦU

Trong bối cảnh toàn cầu hóa và công nghệ phát triển, việc chia sẻ và trao đổi kiến thức trực tuyến đang trở thành xu hướng quan trọng. Tuy nhiên, việc kết nối và giao lưu giữa người học và người dạy trên các nền tảng học trực tuyến hiện nay vẫn chưa thực sự tối ưu. Các nền tảng chủ yếu chỉ tập trung vào việc cung cấp khóa học, thiếu các tính năng để tạo môi trường thảo luận, chia sẻ kiến thức và kết nối người dùng trong cộng đồng.Một nền tảng trực tuyến cho phép người dùng chia sẻ kiến thức, giải đáp thắc mắc, tham gia thảo luận và học hỏi lẫn nhau sẽ giúp người học nâng cao kiến thức và thúc đẩy sự phát triển cộng đồng học thuật.

Nội dung của toàn văn báo cáo bao gồm:

Chương 1: Tổng quan vấn đề

Chương 2: Cơ sở lý thuyết

Chương 3: Phân tích, thiết kế hệ thống

Chương 4: Xây dựng chương trình

Chương 5: Kết quả đạt được

Kết luận

Tài liệu tham khảo

# CHƯƠNG 1: TỔNG QUAN VẤN ĐỀ

## 1.1. Tính cấp thiết của đề tài

Điện thoại thông minh là một thiết bị không thể thiếu trong mọi lĩnh vực cuộc sống của con người. Với điện thoại thông minh, chúng ta đã thực hiện được nhiều công việc với tốc độ nhanh hơn và chi phí thấp hơn nhiều so với cách thức truyền thống. Chính điều này đã thúc đầy sự khai sinh và phát triển của thương mại điện tử và chính phủ điện từ trên khắp thể giới, làm biến đối đáng kể bộ mặt văn hóa, nâng cao chẩt lượng cuộc sống con người.

Trong môi trường học tập ngày càng cạnh tranh, sinh viên cần liên tục nâng cao kiến thức và kỹ năng. Việc có một hệ thống gợi ý thông minh giúp họ dễ dàng truy cập vào các tài liệu, khóa học và thông tin liên quan sẽ đáp ứng nhu cầu học tập của họ một cách hiệu quả. Sinh viên thường cần một không gian để trao đổi ý tưởng và thảo luận về các vấn đề học thuật. Hệ thống gợi ý sẽ khuyến khích họ tham gia vào các cuộc thảo luận, từ đó tạo ra một môi trường học tập tích cực và sáng tạo.

Hệ thống gợi ý giúp sinh viên nhanh chóng tìm kiếm thông tin và nội dung cần thiết mà không phải mất nhiều thời gian để lọc qua hàng loạt thông tin không liên quan. Điều này rất quan trọng trong bối cảnh sinh viên thường phải cân bằng giữa học tập, làm việc và các hoạt động khác. Hệ thống sẽ tạo điều kiện cho sinh viên kết nối với nhau, hỗ trợ lẫn nhau trong việc học tập. Điều này không chỉ giúp họ xây dựng mạng lưới bạn bè mà còn phát triển kỹ năng làm việc nhóm, rất cần thiết trong môi trường làm việc tương lai.

Hệ thống gợi ý sẽ giúp sinh viên phát triển kỹ năng tự học, khuyến khích họ khám phá và tìm hiểu các chủ đề mà họ quan tâm. Điều này không chỉ giúp họ trong quá trình học tập tại trường mà còn trong sự nghiệp sau này.

Với những lý do nêu trên, việc xây dựng **“Hệ Thống Gợi Ý Cho Diễn Đàn Trao Đổi Học Tập”** với các chức năng đơn giản, dễ sử dụng, đặc biệt chỉ hoạt động trong khuôn viên Trường Đại học Nha Trang, là nhu cầu thiết yếu cho mỗi sinh viên. Hệ thống này không chỉ giúp sinh viên dễ dàng tìm kiếm và chia sẻ tài liệu học tập mà còn tạo ra một môi trường thảo luận sôi nổi, khuyến khích sự tương tác và hợp tác giữa các bạn học.

## 1.2. Mục tiêu đề tài

Đề tài **“Hệ Thống Gợi Ý Cho Diễn Đàn Trao Đổi Học Tập”** là xây dựng một nền tảng trực tuyến giao lưu và chia sẻ kiến thức, nơi người dùng có thể đặt câu hỏi, tham gia thảo luận và chia sẻ nội dung học tập, đồng thời tích hợp các tính năng gợi ý, đánh giá, và gamification để khuyến khích sự tương tác và phát triển cộng đồng học thuật, từ đó nâng cao chất lượng giáo dục và trải nghiệm học tập cho sinh viên và người học.

## 1.3. Giải pháp thực hiện

Để xây dựng ứng dụng hoàn chỉnh cần:

* **Thu thập thông tin**:
* Tập hợp dữ liệu về các chủ đề học tập, câu hỏi thường gặp và nội dung chia sẻ từ người dùng.
* **Nghiên cứu công nghệ**:
* Tìm hiểu cách sử dụng React cho frontend và Node.js cho backend.
* Nắm vững cách tích hợp MongoDB để lưu trữ và quản lý dữ liệu.
* **Phát triển chức năng chính**:
* Xây dựng các tính năng như:
  + Đăng ký và đăng nhập người dùng.
  + Đặt câu hỏi và trả lời.
  + Đánh giá và phản hồi nội dung.
  + Tìm kiếm thông tin và gợi ý nội dung.
* **Xây dựng hệ thống gợi ý**:
* Phát triển thuật toán gợi ý dựa trên hành vi người dùng và sử dụng machine learning để cải thiện kết quả gợi ý.
* **Thiết kế giao diện người dùng (UI/UX)**:
* Tạo giao diện dễ sử dụng, thân thiện với người dùng, hỗ trợ tìm kiếm và tương tác.
* **Tích hợp gamification**:
* Thiết kế các yếu tố như điểm thưởng, huy hiệu và bảng xếp hạng để khuyến khích sự tham gia.
* **Kiểm tra và triển khai**:
* Thực hiện kiểm tra hệ thống trên môi trường phát triển và trên thiết bị thực tế để đảm bảo hiệu suất và tính ổn định.
* **Thu thập phản hồi và cải tiến**:
* Liên tục thu thập phản hồi từ người dùng để cải tiến và nâng cao chất lượng dịch vụ của hệ thống.

## 1.4. Đối tượng và phạm vi nghiên cứu

**Đối tượng nghiên cứu**

Đối tượng nghiên cứu: Sinh viên Trường Đại học Nha Trang

**Phạm vi nghiên cứu**

Phạm vi không gian: Nghiên cứu và tìm hiểu tại Trường Đại học Nha Trang

Phạm vi thời gian: Nghiên cứu được thực hiện trong khoảng thời gian từ 15/01/2025 - 26/05/2025.

Phạm vi nội dung: Thu thập thông tin về những nhu cầu cần thiết về việc học tập của sinh viên sau đó cập nhật thông tin đưa vào cơ sở dữ liệu để thiết kế ứng dụng.

## 1.5. Ý nghĩa lí luận và thực tiễn

Thông qua việc thực hiện đồ án này, giúp em hệ thống và củng cố được các vấn đề lí luận cơ bản về ứng dụng của công nghệ thông tin trong các hoạt động đăng bài chia sẻ thông tin trên ứng dụng di động, góp phần cho việc phát triển ứng dụng ngày càng logic, hiện đại và chuyên nghiệp hơn.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## 2.1. Ngôn Ngữ JavaScript

### 2.1.1. Giới thiệu

Javascript là ngôn ngữ lập trình bậc cao, cực kỳ linh hoạt được sử dụng chủ yếu để tao ra ứng dụng chạy trên trình duyệt web. Được tạo ra bởi Brendan Eich vào năm 1995. Nó thể viết code ở text editor và chạy nó trực tiếp trên trình duyệt mà không cần phải trải qua quá trình biên dịch như C++ hoặc Java.



**Hình 2.1. Biểu tượng JavaScript**

Javascript tập trung nâng cao sự tương tác của người dùng với trang web. Bạn có thể làm cho trang web trở nên sống động và tăng tính tương tác hơn. Trong các ứng dụng web, người ta hay dùng JS để làm các hiệu ứng đặc biệt như sliders, pop-ups, hoặc xác thực dữ liệu các form (form validations) trước khi gửi dữ liệu lên server .v.v...

Vì nó cực kỳ linh hoạt, nên gần như rất nhiều lĩnh vực mà Js có thể tạo nên:

* Web App: [ReactJS](https://react.dev/), [VueJS](https://vuejs.org/), [Svelte](https://svelte.dev/), [Astro](https://astro.build/), [Solid](https://www.solidjs.com/)...
* Mobile App: Có thể tạo ra được các ứng dụng di động chạy được đa nền tảng, như IOS và Android, điển hình có thể kể đến là [React Native](https://reactnative.dev/), [Ionic](https://ionicframework.com/), và [NativeScript](https://nativescript.org/)
* Desktop App: Js cho phép lập trình viên tạo ra được các ứng dụng trên Desktop dựa trên website một cách dễ dàng, điển hình là [Tauri](https://tauri.app/), [Electron](https://electronjs.org/) và [NeutralinoJS](https://neutralino.js.org/)
* Server-side app: Không chỉ ở mỗi website, JS đã được nâng tầm để đáp ứng nhu cầu của cộng đồng, xây các ứng dụng ở phía server như [Nodejs](https://nodejs.org/en), [Bun](https://bun.sh/), và [Deno](https://deno.com/)
* Graphic/Game: Nhắc đến game, JS ThreeJs, [BabylonJs](https://www.babylonjs.com/), [Phaser](https://phaser.io/), [PlayCanvas](https://playcanvas.com/), [GDevelop](https://gdevelop.io/), [Pixijs](https://pixijs.com/), [MelonJs](https://melonjs.org/)...
* Internet of Things(IoT): Vâng, JS có thể giúp bạn lập trình và điều khiển các thiết bị IoT, như nhà thông minh(smart home) và tự động hoá các quy trình làm việc của máy móc trong nhà máy sản xuất.
* Machine Learning: JS có thể dùng để train và deploy Machine Learning models. Gần đây nổi nhất là [Hugging Face](https://huggingface.co/) và [TensorFlow](https://www.tensorflow.org/).
* Virtual Reality(VR) & Augmented Reality(AR): Để tạo ra trải nghiệm vượt bậc trên browser, những công cụ như [AFrame](https://aframe.io/) và [BabylonJs](https://www.babylonjs.com/) sẽ giúp chúng ta xây dựng được những thứ không tưởng!
* Web Scraping: Nếu muốn đi lấy dữ liệu từ bên ngoài về(website, video, hình ảnh, thông tin...), hãy sử dụng các thư viện như [Cheerio](https://cheerio.js.org/), [Puppeteer](https://pptr.dev/) hoặc [Playwight](https://playwright.dev/).
* Automation: Với một số thao tác lặp đi lặp lại, như việc release sản phẩm theo định kỳ, hoặc chạy cron job

### 2.1.2. Vai trò

- Tăng cường tính năng động của trang web

- Tạo điều kiện cho tương tác của người dùng để tạo ra trải nghiệm người dùng phong phú và liền mạch

- Áp dụng cho cả giao diện người dùng và giao diện người dùng (sử dụng Node.js)

Các thuộc tính chính của JavaScript:

- Khả năng tương thích đa nền tảng: JavaScript tương thích với tất cả các trình duyệt web hiện tại.

- Ngôn ngữ lập trình hướng đối tượng: hỗ trợ các ý tưởng như kế thừa, lớp và đối tượng.

- Xử lý sự kiện và không đồng bộ: xử lý các truy vấn bằng phương pháp Vòng lặp sự kiện để luồng chính không bị gián đoạn. `

- Tính linh hoạt: JavaScript có thể được áp dụng cho cả giao diện người dùng (Node.js) và giao diện người dùng (trình duyệt).

- Tích hợp: Tạo giao diện dễ dàng bằng HTML và CSS

Các ứng dụng JavaScript của dự án:

- Khả năng tương thích đa nền tảng: JavaScript hoạt động với mọi trình duyệt web hiện có.

- Ngôn ngữ lập trình hướng đối tượng: hỗ trợ các khái niệm như lớp, kế thừa và đối tượng.

- Không đồng bộ và theo sự kiện: các yêu cầu được xử lý thông qua kỹ thuật Vòng lặp sự kiện để ngăn chặn sự gián đoạn đối với luồng chính.

- Tính linh hoạt: JavaScript hoạt động với cả frontend (trình duyệt) và backend (Node.js).

- Tích hợp: Sử dụng HTML và CSS để dễ dàng tạo giao diện

### 2.1.3. Các yếu tố chính của JavaScript

**Variables:** có thể được khai báo bằng let, const hoặc var. Ví dụ:



**Figure 1: Biến trong JavaScript**

**Functions:** Xử lý logic riêng biệt. Ví dụ:



**Figure 2: Function in JavaScrip**

**Events:** Lắng nghe và xử lý hành động của người dùng ví dụ:



**Figure 3: Events in JavaScript**

**Using the DOM:** Mô hình đối tượng tài liệu, hay DOM, đóng vai trò là liên kết giữa HTML và JavaScript.

e.g:



**Figure 4: DOM in JavaScript**

**Thư viện và công cụ hỗ trợ**

Công cụ: Công cụ phát triển bảng điều khiển dựa trên trình duyệt: hỗ trợ kiểm tra và gỡ lỗi JavaScript.

Khung & Thư viện: jQuery: Tạo điều kiện thuận lợi cho thao tác DOM; Chart.js: Tạo biểu đồ dữ liệu đồ họa; React.js, Angular và Vue.js: Xây dựng giao diện người dùng phức tạp.

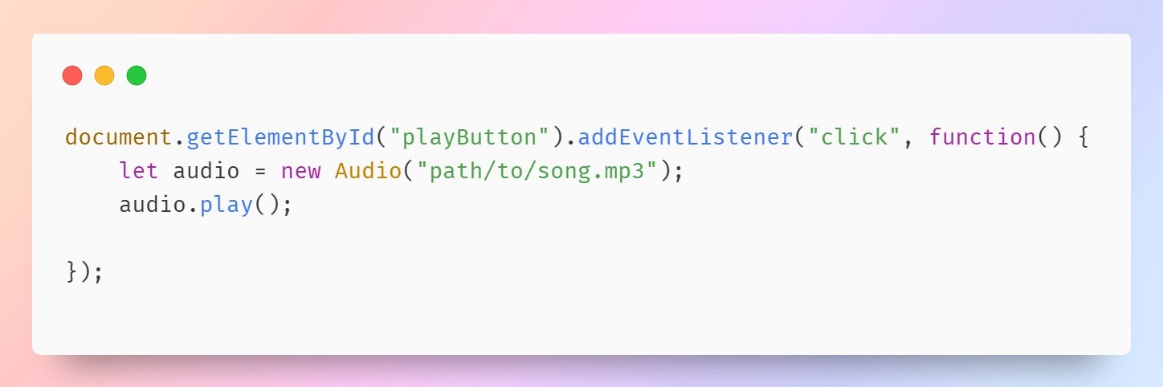
Một ví dụ về việc sử dụng JavaScript trong một dự án:

Tạo danh sách phát nhạc sôi động:



**Figure 5: Create a playlist of lively music**

Phát nhạc khi nhấn nút.



**Figure 6: Play music when a button is pressed.**

Ưu và nhược điểm:

- Ưu điểm: Không cần cài đặt bổ sung và dễ vận hành trên trình duyệt; hiệu quả và dễ hiểu; mạng lưới hỗ trợ rộng khắp.

- Nhược điểm: Nếu không được bảo trì đúng cách, dễ bị tấn công bảo mật; Xử lý lượng dữ liệu lớn có thể làm giảm hiệu suất.

Lý do đưa JavaScript vào dự án:

- JavaScript là ngôn ngữ thiết yếu để phát triển các tương tác trực tuyến.

- Tạo điều kiện giao tiếp tức thời giữa người dùng và phần phụ trợ.

- Dễ tích hợp và cung cấp nhiều thư viện mạnh mẽ để hỗ trợ phát triển dự án.

## 2.2 React JS

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được. React được sử dụng tại Facebook trong production, và [www.instagram.com](http://www.instagram.com/) được viết hoàn toàn trên React

- Ưu Điểm

* Xây dựng DOM ảo tùy chỉnh, bởi vì DOM ảo JavaScript hoạt động nhanh hơn do  với DOM thông thường, điều này sẽ giúp nâng cao hiệu suất của ứng dụng.
* ReactJS tạo ra giao diện người dùng UI cho website hấp dẫn và ấn tượng.
* Thân thiện với công cụ tìm kiếm.
* Module và dữ liệu giúp quản lý ứng dụng lớn dễ dàng hơn bằng cách cải thiện khả năng đọc.
* Tích hợp nhiều kiến trúc khác nhau.
* Tối ưu và đơn giản hóa quy trình môi trường tập lệnh.
* Bảo trì dễ dàng và tăng cường đầu ra.
* Đảm bảo render nhanh hơn.
* Có thể sử dụng nhiều hệ thống khác nhau, trên cả máy khách (client-side) và máy chủ (server-side).

## 2.3. Kiến trúc Component-Based

ReactJS cung cấp tính năng chia nhỏ UI thành các component nhỏ hơn và có tính độc lập. Mỗi components có trạng thái và thuộc tính (props) riêng biệt.

**JSX (JavaScript Syntax Extension)**

JSX là phần mở rộng cú pháp cho JavaScript cho phép lập trình viên viết mã giống HTML trong các tệp JavaScript của nó. Đồng thời, JSX làm cho components ReactJS dễ đọc và thu hút hơn.

const name="GeekforGeeks";

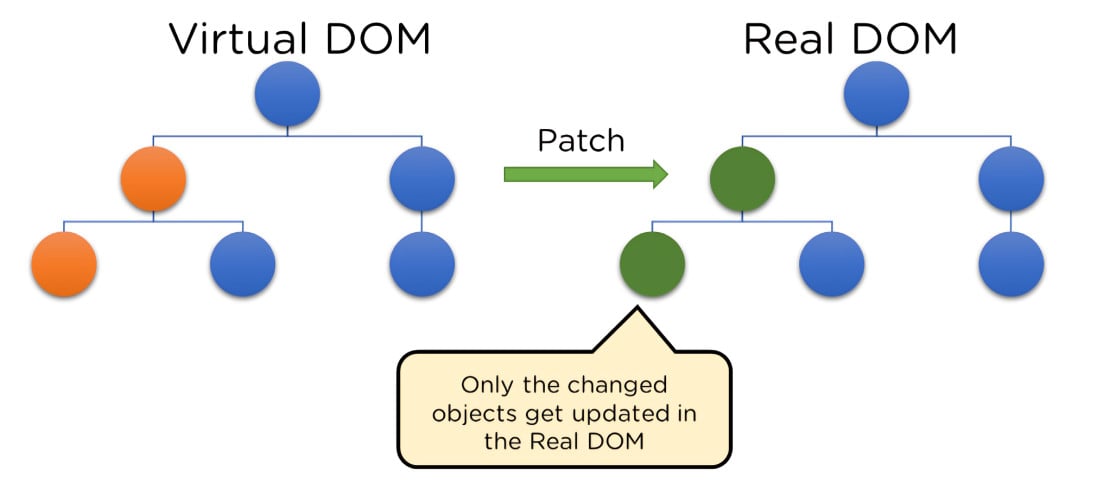
const ele = <h1>Welcome to {name}</h1>;



**Hình 2.2. Hoạt động của JSX (Nguồn ảnh: simplilearn.com)**

**Virtual DOM**

ReactJS duy trì một biểu diễn nhẹ (lightweight representation) của DOM thực tế trong bộ nhớ. Khi có thay đổi, ReactJS chỉ cập nhật hiệu quả của các phần cần thiết trong DOM.



**Hình 2.3. Virtual DOM của ReactJS (Nguồn ảnh: simplilearn.com)**

Liên kết dữ liệu 1 chiều (One-way Data Binding)

Dữ liệu trong ReactJS chỉ chạy theo một hướng, tức là dữ liệu được truyền từ trên xuống dưới, từ thành phần lớn đến thành phần nhỏ. Các thuộc tính (props) trong components con không thể trả về dữ liệu cho components lớn hơn nó, nhưng nó có thể giao tiếp với components cha để sửa đổi các trạng thái theo đầu vào được cung cấp.

Performance

ReactJS sử dụng DOM ảo và chỉ cập nhật các phần đã sửa đổi. Do đó, điều này làm cho DOM chạy nhanh hơn. DOM thực thi trong bộ nhớ để bạn có thể tạo components giúp DOM chạy nhanh hơn.

**Components**

ReactJS chia trang web thành nhiều component khác nhau, bởi vì chúng hoạt động dựa trên các components. Mỗi component là một phần của UI có logic và thiết kế riêng. Do đó, logic component được viết bằng JavaScript sẽ giúp khởi chạy dễ dàng, nhanh chóng và có thể tái sử dụng.

Dữ liệu một trang (Single-Page Applications – SPA)

ReactJS được khuyến khích sử dụng trong việc tạo SPA, cho phép cập nhật nội dung mượt mà mà không cần tải lại trang. Tập trung vào các thành phần có thể tái sử dụng khiến nó trở nên thích hợp cho các ứng dụng thời gian thực.

**Props**

Props là từ viết tắt của Properties trong ReactJS Props, cho phép người dùng truyền tham số hoặc dữ liệu cho component. Props giúp component trở nên năng động hơn và một thành phần chỉ đọc, không thể thay đổi.

## 2.4. Express

Express.js là một framework ứng dụng web cho Node.js, được phát triển để xây dựng các ứng dụng web và API một cách nhanh chóng và dễ dàng. Nó cung cấp một bộ công cụ mạnh mẽ để xử lý các yêu cầu HTTP và quản lý routing. Một số điểm nổi bật của Express.js :

* Đơn giản và nhẹ: Express.js có một API đơn giản và nhẹ, giúp lập trình viên dễ dàng thiết lập và cấu hình ứng dụng của mình.
* Routing mạnh mẽ: Express cho phép định nghĩa các route một cách linh hoạt, giúp xử lý các yêu cầu HTTP đến các URL cụ thể.
* Middleware: Express hỗ trợ middleware, cho phép bạn chèn các chức năng vào chuỗi xử lý yêu cầu và phản hồi, giúp quản lý các yêu cầu dễ dàng hơn (ví dụ: xác thực, phân tích dữ liệu).
* Hỗ trợ RESTful API: Express rất thích hợp để xây dựng các API RESTful, giúp kết nối frontend và backend một cách hiệu quả.
* Ví Dụ :

const express = require(‘express’);

const app = express();

app.get('/', (req, res) => {

   res.send('Hello World!'); }

  );

app.listen(3000, () => {

     console.log('Server is running on port 3000'); });

-Các thành phần của Express JS

+ Middleware là các hàm được gọi trong quá trình xử lý yêu cầu HTTP. Chúng có thể thực hiện các thao tác trước khi yêu cầu được chuyển đến route tiếp theo..

Ví dụ: xác thực người dùng, phân tích thân yêu cầu, xử lý lỗi.

+ Routing:

Express cho phép bạn định nghĩa các route để xử lý các yêu cầu đến một URL cụ thể. Điều này giúp tổ chức mã nguồn và quản lý các yêu cầu một cách dễ dàng hơn.

Ví Dụ :

const authcontroller = require(‘./controller/authcontrollers’);

const express = require(‘express’);

const router = express.router ;

router.post(‘/signup’ , authcontrollers.signup);

module.exports = router

* Response và Request Objects
* Express cung cấp các đối tượng req và res để xử lý yêu cầu và phản hồi.
* req: Chứa thông tin về yêu cầu (URL, phương thức, tham số, thân yêu cầu).
* res: Dùng để gửi phản hồi về cho người dùng (trả về HTML, JSON, tệp, v.v.).

## 2.5. Cơ sở dữ liệu

Database là từ được sử dụng phổ biến trong các lĩnh vực thuộc công nghệ thông tin, dữ liệu, lập trình và phần mềm… .Database là cơ sở dữ liệu, là một bộ sưu tập dữ liệu được tổ chức bày bản và thường được truy cập từ hệ thống máy tính hoặc tồn tại dưới dạng tập tin trong hệ quản trị cơ sở dữ liệu. Database còn có thể được lưu trữ trên thiết bị có chức năng ghi nhớ như: thẻ nhớ, đĩa cứng, CD…

Các mô hình Database phổ biến hiện nay



**Hình 2.4. Mô Hình Cơ Sở Dữ Liệu**

* **Cơ sở dữ liệu Database dạng File:**Database dạng File là dạng thường gặp nhất và cũng là mô hình phổ biến nhất hiện nay. Có thể dễ dàng thấy những tập tin, những thông tin được lưu trữ, chia sẻ và truy cập dưới dạng File. Cơ sở dữ liệu dạng File thường gặp nhất chính là \*.mdb Foxpro. Bên cạnh đó, cơ sở dữ liệu này còn bao gồm những định dạng khác như text, dạng file ascii, dạng file \*.dbf…
* **Cơ sở dữ liệu Database quan hệ:**Đây cũng là một trong những mô hình dữ liệu được sử dụng rộng rãi nhất hiện nay. Ở mô hình này, các thực thể khác nhau, tức các nguồn dữ liệu khác nhau sẽ cùng được quy hợp, lưu trữ cùng một bảng dữ liệu và các dữ liệu này phải có quan hệ với nhau. Một vài cơ sở dữ liệu nổi tiếng có hỗ trợ Database quan hệ có thể kể đến như: Oracle, MS SQL Server, MySQL…
* **Cơ sở dữ liệu hướng đối tượng:**Hệ cơ sở dữ liệu này cũng là một bảng dữ liệu thuần. Tuy nhiên, Database bổ sung thêm các trường hợp hướng đối tượng khác như: hành vi đối tượng nhằm thể hiện hành vi của đối tượng. Các đối tượng này cũng được phân cấp rất rõ ràng, mỗi cấp được gọi là một lớp dữ liệu. Hiểu một cách đơn giản hơn, chúng là tập hợp các nhóm đối tượng trong cùng một bảng và được thể hiện bằng dòng dữ liệu. Những hệ quản trị cơ sở dữ liệu hỗ trợ Database này bao gồm: MS SQL Server, Postgres, Oracle
* **Database bán cấu trúc:** Cơ sở dữ liệu Database bán cấu trúc có thể lưu trữ nhiều dạng dữ liệu khác nhau. Và các thông tin, dữ liệu này thường được lưu trữ dưới định dạng XML. Ở mô hình này, các thông tin mô tả dữ liệu và đối tượng đều được trình bày trong các thẻ tag. Nhờ vào ưu điểm nổi bật này, hệ cơ sở dữ liệu vừa được mở rộng, vừa tiện lợi giúp người dùng có thể dễ dàng truy xuất những thông tin cần thiết. Theo các chuyên gia nghiên cứu, Database bán cấu trúc có thể sẽ là hướng đi mới đầy tiềm năng trong các lĩnh vực ứng dụng công nghệ dữ liệu.

## 2.6. Cơ sở dữ liệu NoSQL

Trong những năm gần đây, sự bùng nổ của các hệ thống xử lý dữ liệu lớn (Big Data), ứng dụng thời gian thực và dịch vụ web quy mô lớn đã làm nổi bật những giới hạn của cơ sở dữ liệu quan hệ (RDBMS). Chính trong bối cảnh đó, các hệ quản trị cơ sở dữ liệu NoSQL (Not Only SQL) ra đời nhằm giải quyết các bài toán mà mô hình truyền thống khó đáp ứng hiệu quả.

### 2.6.1. Khái niệm tổng quan

Cơ sở dữ liệu NoSQL là một tập hợp các mô hình lưu trữ dữ liệu phi quan hệ, không sử dụng ngôn ngữ SQL truyền thống để truy vấn. Thay vào đó, NoSQL sử dụng các kỹ thuật linh hoạt hơn để lưu trữ dữ liệu phi cấu trúc hoặc bán cấu trúc như JSON, XML, hay dạng key-value đơn giản.

### 2.6.2. Phân loại chính

NoSQL bao gồm nhiều mô hình dữ liệu khác nhau, mỗi mô hình có mục tiêu và đặc điểm riêng:

1. Document-oriented (CSDL hướng tài liệu)
   * Lưu trữ dữ liệu dưới dạng tài liệu (thường là JSON hoặc BSON).
   * Phù hợp với ứng dụng cần cấu trúc linh hoạt như hệ thống blog, CMS, REST API.
   * Ví dụ: MongoDB, CouchDB.
2. Key-Value store (Lưu trữ cặp khóa-giá trị)
   * Dữ liệu được truy cập thông qua khóa duy nhất, truy xuất nhanh.
   * Dùng nhiều trong caching hoặc session management.
   * Ví dụ: Redis, DynamoDB.
3. Wide-column store (CSDL cột mở rộng)
   * Dữ liệu được lưu trữ theo cột thay vì dòng.
   * Phù hợp với hệ thống phân tích dữ liệu lớn (Big Data).
   * Ví dụ: Apache Cassandra, HBase.
4. Graph database (CSDL đồ thị)
   * Lưu trữ mối quan hệ giữa các nút, tối ưu hóa cho truy vấn quan hệ phức tạp.
   * Ứng dụng trong mạng xã hội, hệ thống gợi ý.
   * Ví dụ: Neo4j, ArangoDB.

### 2.6.3. Ưu điểm

* Khả năng mở rộng ngang (horizontal scaling) tốt, hỗ trợ phân tán dữ liệu.
* Hiệu năng cao trong các tác vụ đọc/ghi lớn, thời gian thực.
* Linh hoạt trong thiết kế lược đồ, dễ thay đổi cấu trúc dữ liệu.
* Phù hợp với hệ thống đa tenant, microservices, cloud-native.

### 2.6.4. Hạn chế

* Thiếu tính nhất quán mạnh (strong consistency) – nhiều hệ thống áp dụng mô hình eventually consistent.
* Không hỗ trợ đầy đủ các phép toán JOIN như SQL.
* Khó khăn trong việc đảm bảo toàn vẹn dữ liệu nếu không thiết kế cẩn thận.
* Quản trị phức tạp hơn đối với hệ thống phân tán lớn.

### 2.6.5. Ứng dụng thực tế

* Hệ thống phân tích log theo thời gian thực (ELK stack).
* Ứng dụng thương mại điện tử với giỏ hàng động.
* Mạng xã hội cần lưu trữ mối quan hệ người dùng.
* Các hệ thống gợi ý (recommendation system), livestream, game online.

## 2.7. MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu **NoSQL** phổ biến, sử dụng mô hình **document-oriented** để lưu trữ dữ liệu dưới dạng BSON (Binary JSON). Điều này giúp MongoDB linh hoạt trong việc lưu trữ dữ liệu phi cấu trúc hoặc bán cấu trúc. Trong dự án này, MongoDB được lựa chọn vì:



**Hình 2.5. MongODB**

* **Tính linh hoạt cao** trong việc thiết kế cấu trúc dữ liệu.
* **Hiệu suất truy xuất nhanh**, đặc biệt với dữ liệu không yêu cầu ràng buộc chặt chẽ.
* **Dễ mở rộng**, thích hợp với hệ thống cần xử lý dữ liệu lớn theo thời gian thực.

|  |
| --- |
| * **Bảng (Table)** |

|  |
| --- |
|  |
| Dữ liệu lưu trong collection, mỗi bản ghi là một document | |
| * **Các ad hoc query:** hỗ trợ search bằng field, các phép search thông thường, regular expression searches, và range queries. * **Indexing:** bất kì field nào trong BSON document cũng có thể được index. * **Replication:** có ý nghĩa là “nhân bản”, là có một phiên bản giống hệt phiên bản đang tồn tại, đang sử dụng. Với cơ sở dữ liệu, nhu cầu lưu trữ lớn, đòi hỏi cơ sở dữ liệu toàn vẹn, không bị mất mát trước những sự cố ngoài dự đoán là rất cao. Vì vậy, người ta nghĩ ra khái niệm “nhân bản”, tạo một phiên bản cơ sở dữ liệu giống hệt cơ sở dữ liệu đang tồn tại, và lưu trữ ở một nơi khác, đề phòng có sự cố. * **Aggregation:** Các Aggregation operation xử lý các bản ghi dữ liệu và trả về kết quả đã được tính toán. Các phép toán tập hợp nhóm các giá trị từ nhiều Document lại với nhau, và có thể thực hiện nhiều phép toán đa dạng trên dữ liệu đã được nhóm đó để trả về một kết quả duy nhất. Trong SQL, count(\*) và GROUP BY là tương đương với Aggregation trong MongoDB. * **Lưu trữ file:** MongoDB được dùng như một hệ thống file tận dụng những function trên và hoạt động như một cách phân phối qua sharding. | |

## 2.8. Trí tuệ nhân tạo (Artificial Intelligence – AI)

Trí tuệ nhân tạo (AI) là lĩnh vực trong khoa học máy tính nghiên cứu và phát triển các hệ thống có khả năng mô phỏng các chức năng nhận thức của con người như học tập, suy luận, tự động ra quyết định, xử lý ngôn ngữ và nhận dạng mẫu. Trong bối cảnh công nghệ số hiện nay, AI đang được ứng dụng rộng rãi trong nhiều lĩnh vực như y tế, tài chính, giáo dục, thương mại điện tử và đặc biệt là trong **hệ thống gợi ý thông minh**.

### 2.8.1. Vai trò của AI trong hệ thống gợi ý

Trong hệ thống gợi ý, AI đóng vai trò như một "bộ não thông minh", cho phép hệ thống:

* **Phân tích hành vi người dùng**: Ghi nhận và xử lý các tương tác như lượt xem, lượt thích, bình luận, truy vấn tìm kiếm và hoạt động theo dõi.
* **Học được mối quan tâm của từng người dùng** thông qua lịch sử hoạt động, từ đó xây dựng hồ sơ người dùng (user profile).
* **So sánh mức độ tương đồng ngữ nghĩa** giữa các bài viết và hồ sơ người dùng dựa trên biểu diễn văn bản dưới dạng vector.
* **Đưa ra các đề xuất mang tính cá nhân hóa**, phù hợp với từng người dùng khác nhau, thay vì đề xuất nội dung chung cho tất cả.

### 2.8.2. Kỹ thuật AI áp dụng trong đề tài

Đề tài sử dụng AI thông qua các kỹ thuật sau:

* **Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP)**: Các bài viết và nội dung người dùng tạo ra được phân tích, xử lý và mã hóa thành các biểu diễn số thông qua mô hình **vector nhúng ngữ nghĩa** (semantic embedding). Đây là bước quan trọng để giúp máy tính hiểu được nội dung của văn bản.
* **Biểu diễn văn bản bằng vector (Embedding)**: Văn bản của bài viết hoặc từ khóa được chuyển đổi thành vector nhiều chiều thông qua hàm createEmbedding. Kỹ thuật này có thể sử dụng các mô hình học sâu như Word2Vec, FastText, hoặc hiện đại hơn như BERT, nhằm đảm bảo giữ được ngữ nghĩa của văn bản.
* **Tính toán độ tương đồng (Cosine Similarity)**: Sau khi văn bản được chuyển thành vector, hệ thống sử dụng công thức cosine similarity để tính toán mức độ tương đồng giữa vector sở thích người dùng và các bài viết có trong hệ thống. Kết quả là một danh sách các bài viết có nội dung gần giống với sở thích của người dùng.
* **Gợi ý theo nội dung (Content-Based Recommendation)**: Thay vì dựa trên hành vi của người dùng khác (collaborative filtering), hệ thống sử dụng thông tin nội dung bài viết để đưa ra gợi ý, kết hợp với phân tích hành vi cá nhân để tăng độ chính xác.

### 2.8.3. Lợi ích khi tích hợp AI vào hệ thống

* **Cá nhân hóa trải nghiệm học tập**: Người dùng nhận được gợi ý phù hợp với mối quan tâm học thuật của họ.
* **Tiết kiệm thời gian**: Giúp người học nhanh chóng tìm được nội dung có giá trị mà không cần duyệt qua toàn bộ hệ thống.
* **Nâng cao tương tác**: Khi người dùng nhận được những đề xuất chính xác, họ sẽ dành nhiều thời gian hơn cho việc khám phá, từ đó tăng mức độ tương tác và xây dựng cộng đồng học thuật năng động hơn.

## 2.9. Các loại AI và thuật toán thông dụng

### 2.9.1. Các loại AI

Trí tuệ nhân tạo (AI) có thể được phân loại theo nhiều tiêu chí khác nhau, tùy vào khả năng xử lý và phạm vi ứng dụng. Dưới đây là các phân loại chính:

**A. Phân loại theo khả năng xử lý**

1. **AI hẹp (Narrow AI)**
   * Là loại AI phổ biến nhất hiện nay, chỉ có thể thực hiện một nhiệm vụ cụ thể.
   * Ví dụ: Trợ lý ảo (Siri, Google Assistant), hệ thống gợi ý nội dung (Netflix, YouTube), chatbot hỗ trợ khách hàng.
2. **AI tổng quát (General AI)**
   * Có khả năng học hỏi và thực hiện nhiều nhiệm vụ khác nhau giống con người.
   * AI tổng quát có thể suy luận, thích nghi với môi trường mới mà không cần lập trình lại.
   * Hiện nay, AI tổng quát vẫn đang trong quá trình nghiên cứu và phát triển.
3. **AI siêu việt (Super AI)**
   * Một dạng AI giả thuyết có thể vượt qua trí tuệ con người.
   * AI siêu việt có thể tự nhận thức, sáng tạo và đưa ra quyết định độc lập.
   * Đây là một chủ đề gây tranh cãi trong cộng đồng khoa học và chưa có ứng dụng thực tế.

**B. Phân loại theo phương thức học tập**

1. **Machine Learning (ML) – Học máy**
   * Học từ dữ liệu để đưa ra dự đoán hoặc quyết định mà không cần lập trình trực tiếp.
   * Ứng dụng: Dự đoán giá cổ phiếu, phân loại email spam, nhận diện khuôn mặt.
2. **Deep Learning (DL) – Học sâu**
   * Sử dụng mạng nơ-ron nhiều lớp để xử lý dữ liệu phức tạp.
   * Ứng dụng: Nhận diện hình ảnh, giọng nói, dịch thuật tự động.
3. **Natural Language Processing (NLP) – Xử lý ngôn ngữ tự nhiên**
   * Giúp máy tính hiểu và tạo ra ngôn ngữ con người.
   * Ứng dụng: Chatbot, dịch thuật tự động, phân tích cảm xúc từ văn bản.
4. **Computer Vision – Thị giác máy tính**
   * Nhận diện và phân tích hình ảnh, video.
   * Ứng dụng: Xe tự lái, giám sát an ninh, phân tích ảnh y tế.
5. **Reinforcement Learning (RL) – Học tăng cường**
   * Học từ thử và sai để tối ưu hóa hành động.
   * Ứng dụng: AI chơi game, robot tự động hóa, tối ưu hóa chiến lược kinh doanh.

### 2.9.2. Các thuật toán AI thông dụng

Các thuật toán AI đóng vai trò quan trọng trong việc xử lý dữ liệu và đưa ra quyết định. Dưới đây là một số thuật toán phổ biến:

**A. Thuật toán học có giám sát (Supervised Learning)**

1. **Hồi quy tuyến tính (Linear Regression)**
   * Dự đoán giá trị liên tục dựa trên dữ liệu đầu vào.
   * Ứng dụng: Dự đoán giá nhà, dự báo doanh thu.
2. **Hồi quy logistic (Logistic Regression)**
   * Phân loại nhị phân, ví dụ như xác định email có phải là spam hay không.
   * Ứng dụng: Phát hiện gian lận, chẩn đoán bệnh.
3. **Máy vectơ hỗ trợ (Support Vector Machine - SVM)**
   * Tìm siêu phẳng tối ưu để phân loại dữ liệu.
   * Ứng dụng: Nhận diện chữ viết tay, phân loại hình ảnh.
4. **Cây quyết định (Decision Tree)**
   * Mô hình phân loại hoặc hồi quy theo dạng cây.
   * Ứng dụng: Dự đoán hành vi khách hàng, phân tích rủi ro tài chính.
5. **Rừng ngẫu nhiên (Random Forest)**
   * Tập hợp nhiều cây quyết định để cải thiện độ chính xác.
   * Ứng dụng: Phân loại bệnh, dự đoán xu hướng thị trường.

**B. Thuật toán học không giám sát (Unsupervised Learning)**

1. **Phân cụm K-Means (K-Means Clustering)**
   * Phân nhóm dữ liệu thành các cụm tương đồng.
   * Ứng dụng: Phân loại khách hàng, phân tích hành vi người dùng.
2. **Phân tích thành phần chính (Principal Component Analysis - PCA)**
   * Giảm chiều dữ liệu, giúp trực quan hóa và tăng hiệu quả mô hình.
   * Ứng dụng: Nhận diện khuôn mặt, xử lý dữ liệu lớn.

**C. Thuật toán học tăng cường (Reinforcement Learning)**

1. **Q-Learning**
   * Học cách tối ưu hóa hành động dựa trên phần thưởng.
   * Ứng dụng: AI chơi game, robot tự động hóa.
2. **Deep Q-Networks (DQN)**
   * Kết hợp học sâu với Q-Learning để xử lý các bài toán phức tạp.
   * Ứng dụng: Xe tự lái, tối ưu hóa chiến lược kinh doanh.

**D. Mạng nơ-ron nhân tạo (Artificial Neural Networks - ANN)**

1. **Perceptron**
   * Mô hình nơ-ron đơn giản, nền tảng của học sâu.
   * Ứng dụng: Phân loại dữ liệu đơn giản.
2. **Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN)**
   * Chuyên xử lý hình ảnh và video.
   * Ứng dụng: Nhận diện khuôn mặt, phân tích ảnh y tế.
3. **Mạng nơ-ron hồi quy (Recurrent Neural Networks - RNN)**
   * Xử lý dữ liệu chuỗi như văn bản, âm thanh.
   * Ứng dụng: Dịch thuật tự động, nhận diện giọng nói.
4. **Transformer**
   * Mô hình hiện đại nhất trong NLP, nền tảng của ChatGPT.
   * Ứng dụng: Dịch thuật, chatbot, phân tích văn bản.

**2.9.3. Ứng dụng thực tế của AI**

* **Y tế**: AI hỗ trợ chẩn đoán bệnh, phân tích hình ảnh y tế.
* **Tài chính**: Dự đoán xu hướng thị trường, phát hiện gian lận.
* **Thương mại điện tử**: Hệ thống gợi ý sản phẩm, chatbot hỗ trợ khách hàng.
* **Giáo dục**: Hệ thống gợi ý nội dung học tập, trợ lý ảo giảng dạy.
* **Giao thông**: Xe tự lái, tối ưu hóa tuyến đường.

## 2.10. Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) trong hệ thống gợi ý

### 2.10.1. Chuyển đổi văn bản thành dữ liệu số

Máy tính không thể hiểu trực tiếp ngôn ngữ con người như chúng ta. Vì vậy, hệ thống cần **chuyển đổi văn bản thành dạng số** để có thể phân tích, so sánh và đưa ra gợi ý phù hợp. Quá trình này bao gồm các bước sau:

**A. Tokenization – Chia nhỏ văn bản thành từ**

Tokenization là bước đầu tiên trong xử lý ngôn ngữ tự nhiên, giúp chia văn bản thành các từ hoặc cụm từ nhỏ hơn. Ví dụ:

> Văn bản gốc: *"Hệ thống gợi ý giúp sinh viên tìm tài liệu học tập dễ dàng hơn."* > Sau khi token hóa: ["Hệ", "thống", "gợi", "ý", "giúp", "sinh", "viên", "tìm", "tài", "liệu", "học", "tập", "dễ", "dàng", "hơn"]

**B. Stemming – Đưa từ về dạng gốc**

Một từ có thể có nhiều biến thể khác nhau, nhưng chúng có cùng ý nghĩa. Stemming giúp chuẩn hóa từ về dạng gốc để giảm số lượng biến thể. Ví dụ:

> Các từ *học*, *học tập*, *học hỏi* → đều được đưa về dạng gốc là **"học"**.

**C. TF-IDF – Xác định từ quan trọng**

TF-IDF (Term Frequency - Inverse Document Frequency) giúp xác định **độ quan trọng của một từ** trong một bài viết so với toàn bộ hệ thống.

Công thức:

TF-IDF(t, d) = TF(t, d) × log (N / DF(t))

Trong đó:

* **TF(t, d)**: Tần suất xuất hiện của từ *t* trong tài liệu *d*.
* **DF(t)**: Số tài liệu chứa từ *t*.
* **N**: Tổng số tài liệu trong tập dữ liệu.

Ví dụ:

* Nếu từ *"AI"* xuất hiện nhiều trong một bài viết nhưng ít xuất hiện ở các bài khác, nó sẽ có trọng số cao.
* Ngược lại, từ *"và"* xuất hiện ở hầu hết các bài viết, nên trọng số của nó sẽ thấp.

**D. Hashing – Gán vị trí cho từ trong vector**

Sau khi tính toán trọng số TF-IDF, hệ thống cần chuyển đổi từ thành vector số để có thể so sánh. Một cách tiếp cận phổ biến là sử dụng hàm băm (MD5) để ánh xạ từ vào một vị trí trong vector 512 chiều.

Quy trình:

1. Token hóa văn bản thành các từ.
2. Tính trọng số TF-IDF cho từng từ.
3. Sử dụng hàm băm để ánh xạ từ vào một vị trí trong vector.
4. Chuẩn hóa vector bằng L2 normalization để đảm bảo độ dài không ảnh hưởng đến việc so sánh.

### 2.10.2. Cách đo lường mức độ tương đồng giữa hai bài viết

Sau khi biến văn bản thành dữ liệu số, hệ thống cần **so sánh để xem hai bài viết có liên quan đến nhau hay không**. Một phương pháp phổ biến được dùng là:

**A. Cosine Similarity – Công thức tính độ giống nhau**

Cosine Similarity giúp đo mức độ tương đồng giữa hai bài viết bằng cách **so sánh hướng của chúng** trong không gian vector.

Công thức:

Trong đó:

* **A, B** là hai bài viết đã được chuyển đổi thành vector số.
* Nếu kết quả gần **1**, hai bài viết có nội dung giống nhau.
* Nếu kết quả gần **0**, chúng không liên quan.

Ví dụ:

* Nếu hai bài viết cùng nói về "AI", chúng sẽ có **góc gần nhau** → tương đồng cao.
* Nếu một bài viết nói về "AI", bài kia nói về "Ẩm thực", góc sẽ **xa nhau** → không liên quan.

**B. Ứng dụng của Cosine Similarity trong hệ thống**

Hệ thống sẽ sử dụng **Cosine Similarity** để:

* **Gợi ý nội dung phù hợp** với sở thích của người dùng.
* **Tìm kiếm thông tin liên quan** đến từ khóa người dùng nhập vào.
* **Phân loại bài viết** để nhóm những nội dung tương đồng.

### 2.10.3. Tích hợp NLP vào hệ thống gợi ý

Hệ thống gợi ý sử dụng NLP để **phân tích nội dung bài viết**, **tạo vector biểu diễn**, và **so sánh mức độ tương đồng** nhằm đưa ra gợi ý chính xác. Các bước chính:

1. **Tiền xử lý văn bản**: Loại bỏ ký tự đặc biệt, chuyển văn bản về chữ thường.
2. **Phân tích từ vựng**: Tokenization, stemming.
3. **Tính toán trọng số từ**: TF-IDF.
4. **Chuyển đổi văn bản thành vector**: Hashing-based embedding.
5. **So sánh nội dung**: Cosine Similarity.
6. **Đưa ra gợi ý**: Dựa trên mức độ tương đồng giữa bài viết và sở thích người dùng.

## 2.11. Thư viện xử lý ngôn ngữ tự nhiên - Natural

### 2.11.1. Giới thiệu về thư viện Natural

**Natural** là một thư viện xử lý ngôn ngữ tự nhiên (NLP) mạnh mẽ dành cho JavaScript. Nó được sử dụng phổ biến để **phân tích văn bản, tìm kiếm thông tin và xây dựng hệ thống gợi ý**.

Natural cung cấp các công cụ quan trọng để **làm việc với ngôn ngữ tự nhiên mà không cần các dịch vụ bên ngoài**, giúp hệ thống hoạt động **nhanh chóng và hiệu quả**.

**A. Tài liệu tham khảo**

- Natural trên npm: https://www.npmjs.com/package/natural

- Natural Documentation: https://github.com/NaturalNode/natural#documentation

- Natural trên GitHub: <https://github.com/NaturalNode/natural>

### 2.11.2. Các chức năng quan trọng của thư viện Natural

Dưới đây là **những chức năng chính của Natural**, giúp hệ thống phân tích văn bản và đưa ra gợi ý:

**A. Tokenization – Chia nhỏ văn bản thành từ**

Natural hỗ trợ nhiều phương pháp tokenization khác nhau:

* WordTokenizer – Tách văn bản thành từng từ.
* RegexpTokenizer – Chia tách văn bản dựa trên biểu thức chính quy.
* SentenceTokenizer – Phân tách văn bản thành từng câu.

**Ứng dụng trong hệ thống:**

* Giúp phân tách câu hỏi hoặc bài viết để phân tích nội dung.
* Cải thiện tìm kiếm và gợi ý nội dung liên quan.

**B. Stemming – Đưa từ về dạng gốc**

Natural hỗ trợ nhiều thuật toán stemming khác nhau, trong đó phổ biến nhất là:

* PorterStemmer – Đưa từ về dạng gốc bằng thuật toán Porter.
* LancasterStemmer – Phiên bản tối ưu hóa của Porter Stemmer.

**Ứng dụng trong hệ thống:**

* Chuẩn hóa từ trước khi tính toán độ tương đồng.
* Giảm số lượng biến thể của từ, giúp tìm kiếm chính xác hơn.

**C. Tính toán TF-IDF – Đánh giá từ quan trọng**

Natural cung cấp TfIdf, giúp tính toán độ quan trọng của từng từ trong văn bản dựa trên:

* Tần suất xuất hiện của từ trong văn bản (*TF*).
* Mức độ phổ biến của từ trong tập dữ liệu lớn (*IDF*).

**Ứng dụng trong hệ thống:**

* Xác định từ quan trọng trong bài viết để tạo gợi ý phù hợp.
* Loại bỏ những từ phổ biến nhưng ít giá trị (ví dụ: "và", "của").

**D. Đo lường mức độ tương đồng giữa văn bản**

Natural không trực tiếp hỗ trợ **Cosine Similarity**, nhưng nó cho phép:

* Chuyển đổi văn bản thành **vector TF-IDF**, giúp tính toán độ tương đồng.
* Kết hợp với **các thư viện toán học** để đo lường mức độ giống nhau.

**Ứng dụng trong hệ thống:**

* So sánh bài viết để đưa ra gợi ý nội dung liên quan.
* Phân loại văn bản theo chủ đề.

**2.11.3. Ứng dụng của Natural trong hệ thống gợi ý**

- Phân tích nội dung bài viết bằng tokenization và stemming.

- Xác định từ quan trọng bằng TF-IDF.

- Chuyển đổi văn bản thành vector số để so sánh nội dung.

- Tính toán mức độ tương đồng giữa các bài viết.

- Đưa ra gợi ý chính xác dựa trên nội dung liên quan.

## 2.12. Recommendation Service trong hệ thống gợi ý

Trong đề tài này, hệ thống gợi ý được xây dựng dựa trên một Recommendation Service hoạt động như một bộ não phân tích dữ liệu người dùng và đưa ra các bài viết phù hợp nhất. Thành phần này đóng vai trò quan trọng trong việc cá nhân hóa trải nghiệm học tập và nâng cao hiệu quả sử dụng diễn đàn.

Recommendation Service hoạt động theo quy trình:

**1. Thu thập và xử lý dữ liệu người dùng**

* Lịch sử bài viết đã xem, đã tương tác (thích, bình luận, chia sẻ).
* Các truy vấn tìm kiếm gần đây.
* Các bài viết đã đăng hoặc được đánh giá cao.

Dữ liệu được lưu trữ trong MongoDB, phục vụ cho việc phân tích nội dung và hành vi.

**2. Phân tích nội dung bài viết bằng NLP**

* Sử dụng thư viện **Natural (JavaScript)** để xử lý ngôn ngữ tự nhiên.
* Gồm các bước:
  + **Tokenization:** Tách bài viết thành các từ đơn lẻ.
  + **Stemming:** Đưa từ về gốc (ví dụ: học, học tập, học hỏi → học).
  + **TF-IDF:** Tính trọng số từ khóa để xác định từ quan trọng nhất trong bài viết.

**3. Biểu diễn bài viết và hành vi người dùng bằng vector**

* Mỗi bài viết được chuyển thành một vector số trong không gian nhiều chiều.
* Vector đại diện cho “ngữ nghĩa” của nội dung.
* Hồ sơ người dùng cũng được tổng hợp thành một vector trung bình từ các bài viết đã tương tác.

**4. Tính độ tương đồng bằng Cosine Similarity**

* Sử dụng công thức Cosine Similarity để đo khoảng cách giữa hai vector (người dùng ↔ bài viết).
* Các bài viết có vector gần với sở thích người dùng sẽ được ưu tiên gợi ý.

**5. Hiển thị gợi ý ở giao diện người dùng**

* Các bài viết gợi ý sẽ xuất hiện ở phần “For you” hoặc “Bài viết của tôi”.
* Nội dung cập nhật tùy thuộc vào hành vi người dùng.

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

**3.1 Mô hình hóa yêu cầu**

**3.1.1. Phân tích các tác nhân**

a. Người dùng (User)

* Người sử dụng hệ thống với mục đích học tập, chia sẻ kiến thức và tương tác cộng đồng.

Chức năng chính:

* Quản lý tài khoản: Đăng ký/đăng nhập (email hoặc mạng xã hội), khôi phục mật khẩu, quản lý thông tin cá nhân
* Quản lý nội dung: Tạo, chỉnh sửa, xóa bài viết và câu hỏi, gắn thẻ và hình ảnh
* Tương tác: Đánh giá, bình luận, tìm kiếm nội dung theo nhiều tiêu chí
* Tương tác cộng đồng: Theo dõi, kết bạn, nhắn tin, tham gia nhóm thảo luận
* Trò chơi: Tích lũy điểm, đạt huy hiệu, tham gia thử thách học tập
* Trải nghiệm cá nhân hóa: Nhận gợi ý nội dung phù hợp, tùy chỉnh giao diện

b. Quản trị viên (Admin)

* Người có quyền quản lý và giám sát toàn bộ hệ thống.

Chức năng chính:

* Quản lý người dùng: Xem danh sách, khóa/mở tài khoản, phân quyền, xử lý báo cáo
* Quản lý nội dung: Duyệt, chỉnh sửa, xóa bài viết, kiểm duyệt nội dung vi phạm
* Quản lý nhóm: Tạo/xóa nhóm, quản lý thành viên, phân quyền nhóm
* Quản lý gamification: Cấu hình điểm thưởng, huy hiệu, cộng/trừ điểm người dùng

c. Hệ thống Gợi ý (Recommendation System)

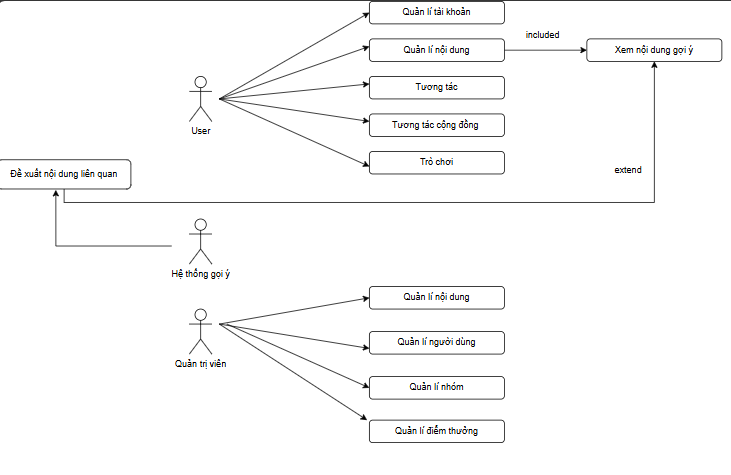
* Hệ thống hỗ trợ thông minh dựa trên AI và phân tích nội dung.

Chức năng chính:

* Phân tích hành vi: Theo dõi lịch sử xem, tìm kiếm, tương tác của người dùng
* Gợi ý cá nhân hóa: Đề xuất bài viết, câu hỏi phù hợp với sở thích
* Đề xuất kết nối: Gợi ý người dùng, nhóm có cùng sở thích/lĩnh vực
* Phân loại nội dung: Tự động phân loại, gắn thẻ nội dung theo chủ đề
* Hỗ trợ tìm kiếm: Nâng cao kết quả tìm kiếm dựa trên ngữ cảnh và nhu cầu người dùng

**3.1.2 Biểu đồ UseCase**

**a. Usecase tổng quát**



**Hình 3.1. Use case tổng quan**

**b. Các usecase chính**

**- Use case Quản Lý Người Dùng**

**Bảng 3.1. Bảng mô tả quản lý người dùng-Admin**

|  |  |
| --- | --- |
| **Use-case** | Quản lý người dùng |
| **Actor** | Admin |
| **Mô tả** | Quản trị viên có thể xem danh sách, chỉnh sửa thông tin, quản lý quyền, thay đổi trạng thái và xóa người dùng trong hệ thống. |
| **Điều kiện kích hoạt** | Truy cập chức năng "Quản lý người dùng" từ giao diện quản trị. |
| **Tiền điều kiện** | Đã đăng nhập với quyền quản trị. |
| **Hậu điều kiện** | Thông tin người dùng được cập nhật, quyền được phân công, hoặc tài khoản bị xóa/khóa. |
| **Luồng sự kiện chính** | 1. Hệ thống hiển thị danh sách người dùng. 2. Admin có thể thực hiện các hành động:  - Xem chi tiết thông tin người dùng  - Chỉnh sửa thông tin người dùng  - Cấp/quản lý quyền người dùng  - Thay đổi trạng thái người dùng (active, inactive, banned)  - Xóa người dùng 3. Admin chọn một hành động. 4. Hệ thống thực hiện hành động và hiển thị kết quả. |
| **Luồng sự kiện nhánh** | Chỉnh sửa thông tin người dùng: - Admin chọn người dùng cần chỉnh sửa thông tin. - Hệ thống hiển thị form với thông tin hiện tại của người dùng. - Admin cập nhật thông tin và lưu lại. - Hệ thống kiểm tra và lưu thông tin, hiển thị thông báo thành công.  Cấp/quản lý quyền người dùng: - Admin chọn người dùng cần cấp/thay đổi quyền. - Hệ thống hiển thị form với các quyền hiện tại của người dùng. - Admin thay đổi quyền và lưu lại. - Hệ thống cập nhật quyền và hiển thị thông báo.  Xóa người dùng: - Admin chọn người dùng cần xóa. - Hệ thống hiển thị hộp thoại xác nhận xóa. - Admin xác nhận xóa. - Hệ thống xóa người dùng và hiển thị thông báo thành công.  Thay đổi trạng thái người dùng: - Admin chọn người dùng và trạng thái mới (active, inactive, banned). - Hệ thống cập nhật trạng thái và hiển thị thông báo. |

**Bảng 3.2. Bảng mô tả quản lý người dùng-User**

|  |  |
| --- | --- |
| **Use-case** | Quản lý người dùng |
| **Actor** | User |
| **Mô tả** | Người dùng có thể đăng ký tài khoản, đăng nhập, đăng xuất, xem và chỉnh sửa thông tin cá nhân, thay đổi mật khẩu. |
| **Điều kiện kích hoạt** | Truy cập vào hệ thống hoặc vào trang thông tin cá nhân. |
| **Tiền điều kiện** | Có kết nối internet và truy cập được vào hệ thống. |
| **Hậu điều kiện** | Tài khoản người dùng được tạo mới, thông tin được cập nhật, hoặc trạng thái đăng nhập được thay đổi. |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào hệ thống. 2. Người dùng có thể thực hiện các hành động:  - Đăng ký tài khoản mới (nếu chưa có)  - Đăng nhập vào hệ thống  - Xem thông tin cá nhân  - Chỉnh sửa thông tin cá nhân  - Thay đổi mật khẩu  - Đăng xuất 3. Người dùng chọn một hành động. 4. Hệ thống thực hiện hành động và hiển thị kết quả. |
| **Luồng sự kiện nhánh** | Đăng ký tài khoản: - Người dùng chọn "Đăng ký". - Hệ thống hiển thị form đăng ký. - Người dùng nhập thông tin cá nhân, tên đăng nhập, email, mật khẩu. - Hệ thống kiểm tra tính hợp lệ của thông tin và tạo tài khoản mới.  Đăng nhập: - Người dùng nhập thông tin đăng nhập (tên đăng nhập/email và mật khẩu). - Hệ thống xác thực thông tin và cho phép truy cập nếu hợp lệ.  Chỉnh sửa thông tin cá nhân: - Người dùng truy cập trang thông tin cá nhân. - Người dùng chọn chỉnh sửa thông tin. - Hệ thống hiển thị form với thông tin hiện tại. - Người dùng cập nhật thông tin và lưu lại.  Thay đổi mật khẩu: - Người dùng chọn chức năng thay đổi mật khẩu. - Hệ thống yêu cầu nhập mật khẩu cũ và mật khẩu mới. - Hệ thống kiểm tra tính hợp lệ và cập nhật mật khẩu mới.  Đăng xuất: - Người dùng chọn "Đăng xuất". - Hệ thống xóa phiên đăng nhập và chuyển hướng về trang đăng nhập. |

**- Use case: Đăng ký & Đăng nhập**

**Bảng 3.3. Bảng mô tả đăng nhập / đăng kí**

|  |  |
| --- | --- |
| **Use-case** | Đăng nhập và đăng kí |
| **Actor** | User ,Admin |
| **Mô tả** | Người dùng có thể đăng ký và đăng nhập vào hệ thống qua email hoặc tài khoản mạng xã hội (Google/Facebook). Hệ thống cũng hỗ trợ tính năng quên mật khẩu. |
| **Điều kiện kích hoạt** | Người dùng truy cập vào trang đăng ký hoặc đăng nhập của hệ thống. |
| **Tiền điều kiện** | Người dùng chưa đăng nhập vào hệ thống. |
| **Hậu điều kiện** | Người dùng đăng nhập thành công và truy cập vào giao diện người dùng. |
| **Luồng sự kiện chính** | Người dùng truy cập vào trang đăng nhập hoặc đăng ký.  Người dùng nhập thông tin đăng nhập (email và mật khẩu) hoặc chọn đăng nhập bằng tài khoản mạng xã hội.  Hệ thống xác thực thông tin đăng nhập và cho phép người dùng truy cập vào hệ thống.  Nếu người dùng quên mật khẩu, họ có thể yêu cầu hệ thống gửi email khôi phục mật khẩu. |
| **Luồng sự kiện rẽ nhánh** | Đăng nhập thành công: Người dùng được chuyển đến trang chủ hoặc giao diện cá nhân.Đăng nhập thất bại: Hệ thống yêu cầu người dùng nhập lại thông tin chính xác. |

**- Usecase quản lý nội dung**

**Bảng 3.4. Bảng mô tả quản lý nội dung-Admin**

|  |  |
| --- | --- |
| **Use-case** | Quản lý nội dung |
| **Actor** | Admin |
| **Mô tả** | Quản trị viên có thể kiểm duyệt, phê duyệt, ẩn hoặc xóa nội dung do người dùng đăng phát hiện tiềm ẩn khả năng xấu, bình hận. |
| **Điều kiện kích hoạt** | Truy cập chức năng "Quản lý nội dung". |
| **Tiền điều kiện** | Đã đăng nhập với quyền quản trị. |
| **Hậu điều kiện** | Nội dung được cập nhật trạng thái: phê duyệt, ẩn, xóa hoặc hiển thị lại. |
| **Luồng sự kiện chính** | 1. Hệ thống hiển thị danh sách nội dung (bao gồm tiêu đề, nội dung, tác giả, thời gian đăng). 2. Admin chọn nội dung cần kiểm duyệt. 3. Hệ thống hiển thị chi tiết nội dung (bài viết, bình luận, hình ảnh). 4. Admin kiểm tra nội dung và chọn 1 trong các hành động sau:  - Phê duyệt nội dung  - Ẩn/khóa nội dung vi phạm  - Xóa nội dung  - Khôi phục nội dung đã ẩn trước đó 5. Hệ thống thực hiện hành động và cập nhật trạng thái nội dung. |
| **Luồng sự kiện nhánh** | Phê duyệt nội dung: - Admin chọn nội dung cần duyệt. - Hệ thống hiển thị chi tiết nội dung. - Admin xác nhận phê duyệt. - Hệ thống cập nhật trạng thái và hiển thị thông báo thành công.  Xử lý nội dung vi phạm: - Admin chọn nội dung vi phạm. - Admin chọn lý do vi phạm từ danh sách. - Hệ thống cập nhật trạng thái và gửi thông báo tới người đăng.  Tìm kiếm nội dung: - Admin nhập từ khóa tìm kiếm. - Hệ thống hiển thị kết quả phù hợp. |

**Bảng 3.5. Bảng mô tả quản lý nội dung-User**

|  |  |
| --- | --- |
| **Use-case** | Quản lý nội dung |
| **Actor** | User |
| **Mô tả** | Người dùng có thể đăng, chỉnh sửa, xóa và quản lý nội dung của mình trên hệ thống. |
| **Điều kiện kích hoạt** | Truy cập chức năng "Quản lý nội dung" từ trang cá nhân. |
| **Tiền điều kiện** | Đã đăng nhập vào tài khoản. |
| **Hậu điều kiện** | Nội dung được tạo mới, cập nhật hoặc xóa khỏi hệ thống. |
| **Luồng sự kiện chính** | 1. Hệ thống hiển thị danh sách nội dung đã đăng của người dùng. 2. Người dùng có thể thực hiện các hành động:  - Tạo nội dung mới  - Xem chi tiết nội dung đã đăng  - Chỉnh sửa nội dung  - Xóa nội dung 3. Hệ thống xử lý yêu cầu và cập nhật trạng thái. 4. Hệ thống hiển thị thông báo kết quả. |
| **Luồng sự kiện nhánh** | Tạo nội dung mới: - Người dùng nhấn "Tạo bài viết mới". - Hệ thống hiển thị form nhập liệu. - Người dùng điền thông tin và đăng nội dung. - Hệ thống kiểm tra và lưu nội dung với trạng thái chờ duyệt.  Chỉnh sửa nội dung: - Người dùng chọn nội dung cần chỉnh sửa. - Hệ thống hiển thị form với thông tin hiện tại. - Người dùng cập nhật và lưu thông tin. - Hệ thống cập nhật dữ liệu và hiển thị thông báo.  Xóa nội dung: - Người dùng chọn nội dung cần xóa. - Hệ thống hiển thị hộp thoại xác nhận. - Người dùng xác nhận xóa. - Hệ thống xóa nội dung và hiển thị thông báo. |

**- Use Case : Quản lý nhóm thảo luận**

**Bảng 3.6. Bảng mô tả quản lý nhóm thảo luận-Admin**

|  |  |
| --- | --- |
| **Use-case** | Quản lý nhóm thảo luận |
| **Actor** | Admin |
| **Mô tả** | Quản trị viên có thể tìm kiếm, xem danh sách, chỉnh sửa, xóa, quản lý thành viên nhóm thảo luận và phê duyệt nội dung trong nhóm. |
| **Điều kiện kích hoạt** | Truy cập chức năng "Quản lý nhóm thảo luận" từ giao diện quản trị. |
| **Tiền điều kiện** | Đã đăng nhập với quyền quản trị. |
| **Hậu điều kiện** | Thông tin nhóm được cập nhật, trạng thái nhóm thay đổi, thành viên được quản lý, hoặc nhóm bị xóa. |
| **Luồng sự kiện chính** | 1. Hệ thống hiển thị danh sách nhóm thảo luận. 2. Admin có thể thực hiện các hành động:  - Tìm kiếm nhóm  - Xem danh sách nhóm  - Chỉnh sửa thông tin nhóm  - Cập nhật trạng thái nhóm  - Quản lý thành viên nhóm  - Xem thống kê nhóm  - Phê duyệt nội dung  - Xóa nhóm 3. Admin chọn một hành động. 4. Hệ thống thực hiện hành động và hiển thị kết quả. |
| **Luồng sự kiện nhánh** | Tìm kiếm nhóm: - Admin nhập từ khóa tìm kiếm. - Hệ thống hiển thị các nhóm phù hợp với từ khóa.  Chỉnh sửa thông tin nhóm: - Admin chọn nhóm cần chỉnh sửa. - Hệ thống hiển thị form với thông tin hiện tại của nhóm. - Admin cập nhật thông tin và lưu lại. - Hệ thống kiểm tra và lưu thông tin, hiển thị thông báo thành công.  Cập nhật trạng thái nhóm: - Admin chọn nhóm và trạng thái mới (active, inactive, featured, pending, blocked). - Hệ thống cập nhật trạng thái và hiển thị thông báo. Quản lý thành viên nhóm: - Admin chọn nhóm để quản lý thành viên. - Hệ thống hiển thị danh sách thành viên. - Admin có thể thêm/xóa thành viên hoặc thay đổi vai trò của họ. - Hệ thống cập nhật thông tin thành viên.  Xem thống kê nhóm: - Admin chọn nhóm để xem thống kê. - Hệ thống hiển thị thông tin thống kê (số thành viên, số bài đăng, hoạt động...).  Phê duyệt nội dung: - Admin xem danh sách nội dung chờ phê duyệt trong nhóm. - Admin chọn phê duyệt hoặc từ chối nội dung. - Hệ thống cập nhật trạng thái nội dung.  Xóa nhóm: - Admin chọn nhóm cần xóa. - Hệ thống hiển thị hộp thoại xác nhận xóa. - Admin xác nhận xóa. - Hệ thống xóa nhóm và hiển thị thông báo thành công. |

**Bảng 3.7. Bảng mô tả quản lý nhóm thảo luận-User**

|  |  |
| --- | --- |
| **Use-case** | Quản lý nhóm thảo luận |
| **Actor** | User |
| **Mô tả** | Người dùng có thể tạo nhóm mới, tham gia vào nhóm hiện có, quản lý nhóm (nếu là chủ nhóm) hoặc rời khỏi nhóm. |
| **Điều kiện kích hoạt** | Truy cập vào trang nhóm thảo luận. |
| **Tiền điều kiện** | Đã đăng nhập vào hệ thống. |
| **Hậu điều kiện** | Nhóm mới được tạo, người dùng tham gia/rời khỏi nhóm, hoặc thông tin nhóm được cập nhật. |
| **Luồng sự kiện chính** | 1. Người dùng truy cập trang nhóm thảo luận. 2. Người dùng có thể thực hiện các hành động:  - Tạo nhóm mới  - Tham gia nhóm  - Quản lý nhóm (nếu là chủ nhóm)  - Rời khỏi nhóm 3. Người dùng chọn một hành động. 4. Hệ thống thực hiện hành động và hiển thị kết quả. |
| **Luồng sự kiện nhánh** | Tạo nhóm mới: - Người dùng chọn "Tạo nhóm". - Hệ thống hiển thị form tạo nhóm. - Người dùng nhập thông tin nhóm (tên, mô tả, hình ảnh, quyền riêng tư...). - Hệ thống kiểm tra và tạo nhóm mới, đặt người dùng làm chủ nhóm.  Tham gia nhóm: - Người dùng tìm kiếm và chọn nhóm muốn tham gia. - Người dùng nhấn "Tham gia nhóm". - Nếu là nhóm công khai, hệ thống thêm người dùng vào nhóm ngay. - Nếu là nhóm riêng tư, hệ thống gửi yêu cầu tham gia đến chủ nhóm.  Quản lý nhóm (chủ nhóm): - Chủ nhóm có thể thay đổi thông tin nhóm, mời thêm thành viên, quản lý thành viên. - Chủ nhóm có thể duyệt các yêu cầu tham gia và nội dung đăng trong nhóm. - Chủ nhóm có thể chỉ định vai trò cho thành viên khác.  Rời khỏi nhóm: - Người dùng chọn nhóm muốn rời. - Người dùng nhấn "Rời khỏi nhóm". - Hệ thống hiển thị hộp thoại xác nhận. - Người dùng xác nhận rời nhóm. - Hệ thống loại bỏ người dùng khỏi nhóm. |

**- Usecase Đánh giá câu trả lời và bài viết**

**Bảng 3.8. Bảng mô tả Đánh giá nội dung-Admin**

|  |  |
| --- | --- |
| **Use-case** | Hệ thống Đánh giá Nội dung |
| **Actor** | Quản trị viên |
| **Mô tả** | Quản trị viên có thể tương tác với nội dung như người dùng thông thường và có thêm quyền cập nhật số lượng thích của nội dung. |
| **Điều kiện kích hoạt** | Quản trị viên xem một bài viết hoặc bình luận trong hệ thống hoặc truy cập trang quản trị. |
| **Tiền điều kiện** | Đã đăng nhập với quyền quản trị viên. |
| **Hậu điều kiện** | Phản hồi hoặc cập nhật của quản trị viên được lưu lại và hiển thị trên bài viết hoặc bình luận. |
| **Luồng sự kiện chính** | 1. Quản trị viên truy cập bài viết, xem bình luận, hoặc vào trang quản trị. 2. Quản trị viên có thể thực hiện các hành động:  - Thích bài viết  - Thích bình luận  - Cập nhật số lượng thích  - Xem trạng thái đánh giá 3. Quản trị viên chọn một hành động. 4. Hệ thống thực hiện hành động và cập nhật hiển thị. |
| **Luồng sự kiện nhánh** | Thích bài viết hoặc bình luận: - Tương tự như người dùng thông thường.  Cập nhật số lượng thích: - Quản trị viên truy cập trang quản trị cho bài viết hoặc bình luận cần cập nhật. - Quản trị viên thay đổi số lượng lượt thích hoặc xóa lượt thích của một người dùng cụ thể. - Hệ thống kiểm tra và cập nhật số lượng, hiển thị thông báo thành công.  Xem trạng thái đánh giá: - Quản trị viên có thể xem thống kê chi tiết hơn so với người dùng thường. - Quản trị viên có thể xem danh sách đầy đủ người đã thích, thời gian thích, và các chỉ số khác. - Quản trị viên có thể xuất báo cáo về thống kê tương tác cho bài viết hoặc bình luận. |

**Bảng 3.9. Bảng mô tả Đánh giá nội dung-User**

|  |  |
| --- | --- |
| **Use-case** | Hệ thống Đánh giá Nội dung |
| **Actor** | Người dùng |
| **Mô tả** | Người dùng có thể tương tác với nội dung thông qua việc thích bài viết, bình luận bài viết, thích bình luận và xem trạng thái đánh giá. |
| **Điều kiện kích hoạt** | Người dùng xem một bài viết hoặc bình luận trong hệ thống. |
| **Tiền điều kiện** | Đã đăng nhập vào hệ thống. |
| **Hậu điều kiện** | Phản hồi của người dùng được lưu lại và hiển thị trên bài viết hoặc bình luận. |
| **Luồng sự kiện chính** | 1. Người dùng truy cập bài viết hoặc xem bình luận. 2. Người dùng có thể thực hiện các hành động:  - Thích bài viết  - Bình luận bài viết  - Thích bình luận  - Xem trạng thái đánh giá (số lượt thích, bình luận) 3. Người dùng chọn một hành động. 4. Hệ thống thực hiện hành động và cập nhật hiển thị. |
| **Luồng sự kiện nhánh** | Thích bài viết: - Người dùng nhấn vào biểu tượng "thích" trên bài viết. - Nếu người dùng chưa thích bài viết trước đó, hệ thống ghi nhận lượt thích và cập nhật số lượng. - Nếu người dùng đã thích bài viết trước đó, hệ thống hủy lượt thích và giảm số lượng.  Bình luận bài viết: - Người dùng nhập nội dung bình luận vào ô bình luận. - Người dùng nhấn nút gửi bình luận. - Hệ thống kiểm tra nội dung bình luận và lưu lại. - Hệ thống hiển thị bình luận mới trong danh sách bình luận.  Thích bình luận: - Người dùng nhấn vào biểu tượng "thích" trên một bình luận. - Tương tự như thích bài viết, hệ thống ghi nhận hoặc hủy lượt thích và cập nhật số lượng.  Hiển thị trạng thái đánh giá: - Hệ thống tự động hiển thị số lượt thích, số bình luận của bài viết. - Hệ thống hiển thị danh sách người dùng đã thích (nếu có chức năng này). |

**Usecase Tìm kiếm nội dung**

**Bảng 3.10. Bảng mô tả tìm kiếm nội dung**

|  |  |
| --- | --- |
| **Use-case** | Tìm Kiếm Nội Dung |
| **Actor** | Người Dùng,Admin |
| **Mô tả** | Người dùng có thể tìm kiếm câu hỏi, bài viết theo từ khóa, chủ đề, hoặc người đăng. |
| **Điều kiện kích hoạt** | Người dùng muốn tìm kiếm nội dung. |
| **Tiền điều kiện** | Người dùng đã đăng nhập vào hệ thống. |
| **Hậu điều kiện** | Hệ thống hiển thị kết quả tìm kiếm. |
| **Luồng sự kiện chính** | Người dùng nhập từ khóa, chủ đề hoặc tên người đăng vào ô tìm kiếm.  Hệ thống tìm kiếm và hiển thị kết quả liên quan. |

**- Usecase Quản lý bài viết/câu hỏi**

**Bảng 3.11 Bảng mô tả quản Lý Bài Viết / Câu Hỏi-Admin**

|  |  |
| --- | --- |
| **Use-case** | Quản lý bài viết/câu hỏi |
| **Actor** | Admin |
| **Mô tả** | Admin có thể xem, tìm kiếm, quản lý nội dung vi phạm, phê duyệt và khôi phục bài viết/câu hỏi trong hệ thống. |
| **Điều kiện kích hoạt** | Admin truy cập vào trang quản trị nội dung. |
| **Tiền điều kiện** | Đã đăng nhập với quyền quản trị viên. |
| **Hậu điều kiện** | Bài viết/câu hỏi được duyệt, khôi phục hoặc xử lý vi phạm theo quy định. |
| **Luồng sự kiện chính** | 1. Admin truy cập vào trang quản trị nội dung. 2. Admin có thể thực hiện các hành động:  - Xem bài viết/câu hỏi  - Tìm kiếm nội dung  - Quản lý nội dung vi phạm  - Phê duyệt nội dung  - Khôi phục bài viết đã xóa 3. Admin chọn một hành động. 4. Hệ thống thực hiện hành động và hiển thị kết quả. |
| **Luồng sự kiện nhánh** | Xem bài viết/câu hỏi: - Admin xem danh sách bài viết và các thông tin chi tiết bao gồm cả thông tin người đăng. - Admin có thể áp dụng nhiều bộ lọc để xem các bài viết theo trạng thái, người đăng, thời gian...  Tìm kiếm nội dung: - Admin sử dụng công cụ tìm kiếm nâng cao. - Hệ thống hiển thị kết quả với các thông tin chi tiết và các tùy chọn quản lý.  Quản lý nội dung vi phạm: - Admin xem danh sách nội dung bị báo cáo vi phạm. - Admin có thể đánh dấu, cảnh báo, ẩn, xóa hoặc khôi phục nội dung. - Hệ thống cập nhật trạng thái của nội dung vi phạm và có thể gửi thông báo cho người đăng.  Phê duyệt nội dung: - Admin xem danh sách nội dung chờ phê duyệt. - Admin kiểm tra nội dung và chọn phê duyệt hoặc từ chối. - Nếu từ chối, admin nhập lý do để gửi thông báo cho người đăng. - Hệ thống cập nhật trạng thái nội dung và gửi thông báo.  Khôi phục bài viết: - Admin xem danh sách bài viết đã bị xóa. - Admin chọn bài viết cần khôi phục. - Admin xác nhận khôi phục. - Hệ thống khôi phục bài viết và hiển thị thông báo thành công. |

**Bảng 3.12. Bảng mô tả quản Lý Bài Viết / Câu Hỏi-User**

|  |  |
| --- | --- |
| **Use-case** | Quản lý bài viết/câu hỏi |
| **Actor** | Người dùng |
| **Mô tả** | Người dùng có thể xem, tìm kiếm, tạo mới, chỉnh sửa và xóa bài viết/câu hỏi của họ trên hệ thống. |
| **Điều kiện kích hoạt** | Người dùng truy cập vào trang bài viết hoặc khu vực câu hỏi. |
| **Tiền điều kiện** | Đã đăng nhập vào hệ thống (ngoại trừ chức năng xem và tìm kiếm có thể không cần đăng nhập). |
| **Hậu điều kiện** | Nội dung được hiển thị, thêm mới, cập nhật hoặc xóa khỏi hệ thống. |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào khu vực bài viết/câu hỏi. 2. Người dùng có thể thực hiện các hành động:  - Xem bài viết/câu hỏi  - Tìm kiếm nội dung  - Tạo bài viết mới  - Chỉnh sửa bài viết đã đăng  - Xóa bài viết 3. Người dùng chọn một hành động. 4. Hệ thống thực hiện hành động và hiển thị kết quả. |
| **Luồng sự kiện nhánh** | Xem bài viết/câu hỏi: - Người dùng chọn bài viết cần xem. - Hệ thống hiển thị chi tiết bài viết và các bình luận liên quan.  Tìm kiếm nội dung: - Người dùng nhập từ khóa tìm kiếm. - Hệ thống hiển thị các bài viết/câu hỏi phù hợp với từ khóa. - Người dùng có thể lọc kết quả theo tiêu chí (thời gian, chủ đề...).  Tạo bài viết: - Người dùng chọn "Tạo bài viết mới". - Hệ thống hiển thị form nhập nội dung. - Người dùng điền thông tin, nội dung và gửi đi. - Hệ thống kiểm tra và lưu bài viết, hiển thị thông báo thành công.  Chỉnh sửa bài viết: - Người dùng chọn bài viết cần chỉnh sửa. - Hệ thống kiểm tra quyền và hiển thị form chỉnh sửa với nội dung hiện tại. - Người dùng cập nhật nội dung và gửi đi. - Hệ thống lưu thông tin mới và hiển thị thông báo.  Xóa bài viết: - Người dùng chọn bài viết cần xóa. - Hệ thống hiển thị hộp thoại xác nhận. - Người dùng xác nhận xóa. - Hệ thống xóa bài viết và hiển thị thông báo. |

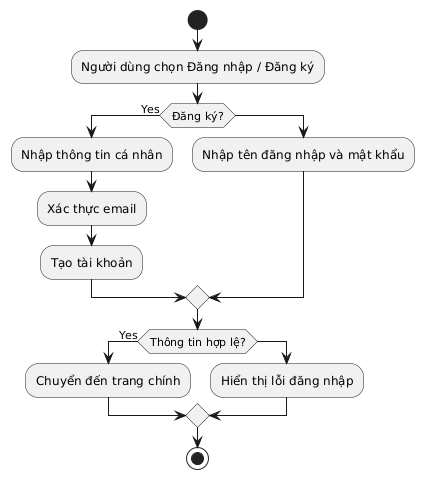
**- UseCase Nhận Thông Báo**

|  |  |
| --- | --- |
| **Use-case** | Nhận Thông Báo |
| **Actor** | Người Dùng |
| **Mô tả** | |  | | --- | | Người dùng nhận thông báo khi có hoạt động liên quan như bình luận, mời nhóm, v.v. |  |  | | --- | |  | |
| **Điều kiện kích hoạt** | |  | | --- | | Có sự kiện mới từ hệ thống (ví dụ: bài viết mới, bình luận, tag...) |  |  | | --- | |  | |
| **Tiền điều kiện** | |  | | --- | | Người dùng đã đăng nhập, và bật tính năng thông báo |  |  | | --- | |  | |
| **Hậu điều kiện** | |  | | --- | | Người dùng nhận được thông báo |  |  | | --- | |  | |
| **Luồng sự kiện chính** | 1. Có sự kiện trong hệ thống.  2. Hệ thống tạo và gửi thông báo.  3. Người dùng thấy thông báo. |

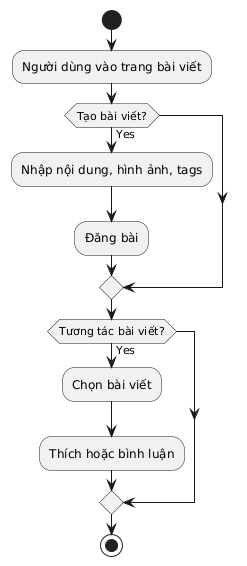
**Bảng 3.13. Bảng mô tả nhận thông báo**

## 3.2. Mô hình hóa hành vi

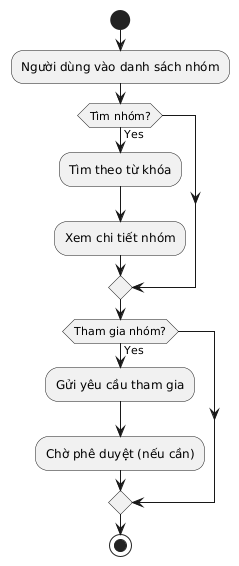
### 3.2.1 Biều đồ hoạt động



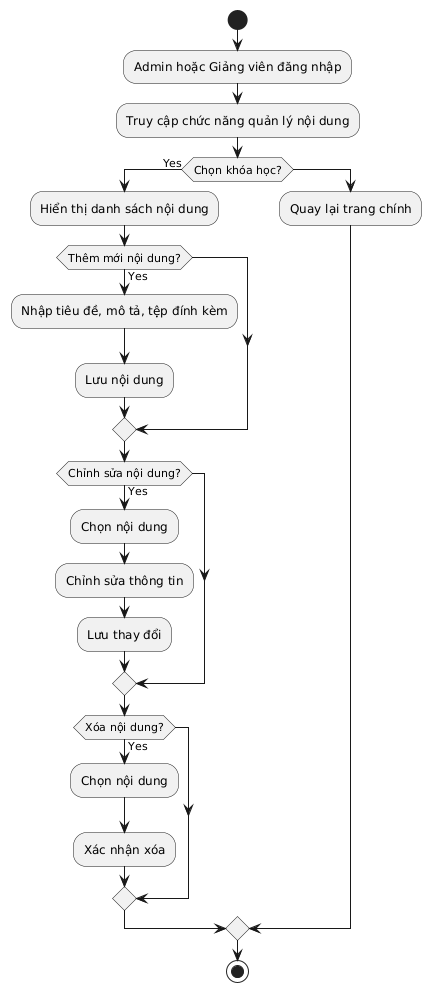
**Hình 3.2. Biểu đồ hoạt động đăng nhập**



**Hình 3.3. Biểu đồ hoạt động Tạo và tương tác bài viết**

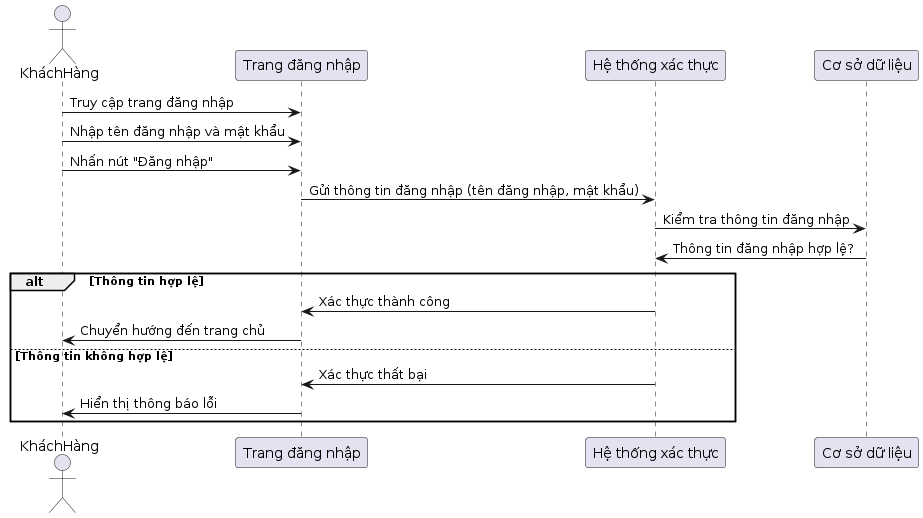


**Hình 3.4. Biểu đồ hoạt động: Quản lý nhóm**

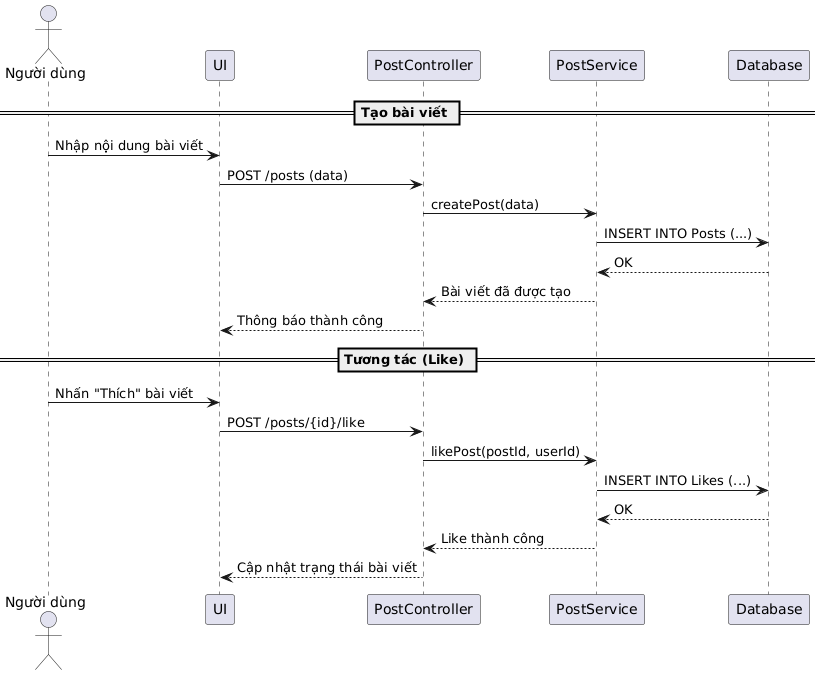


**Hình 3.5. Biểu đồ hoạt động quản lý nội dung**

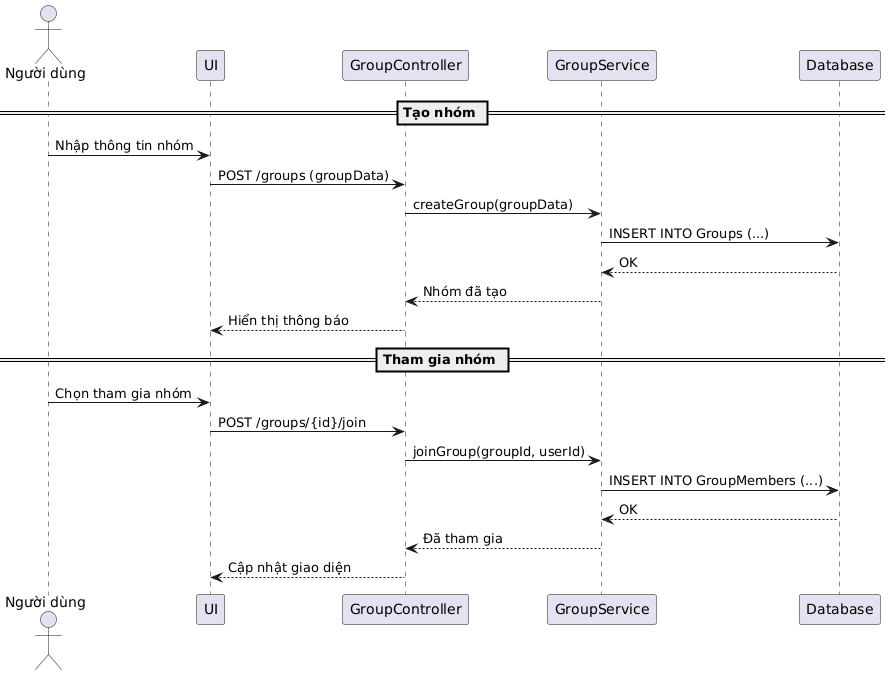
3.2.2 Biểu đồ tuần tự đăng nhập



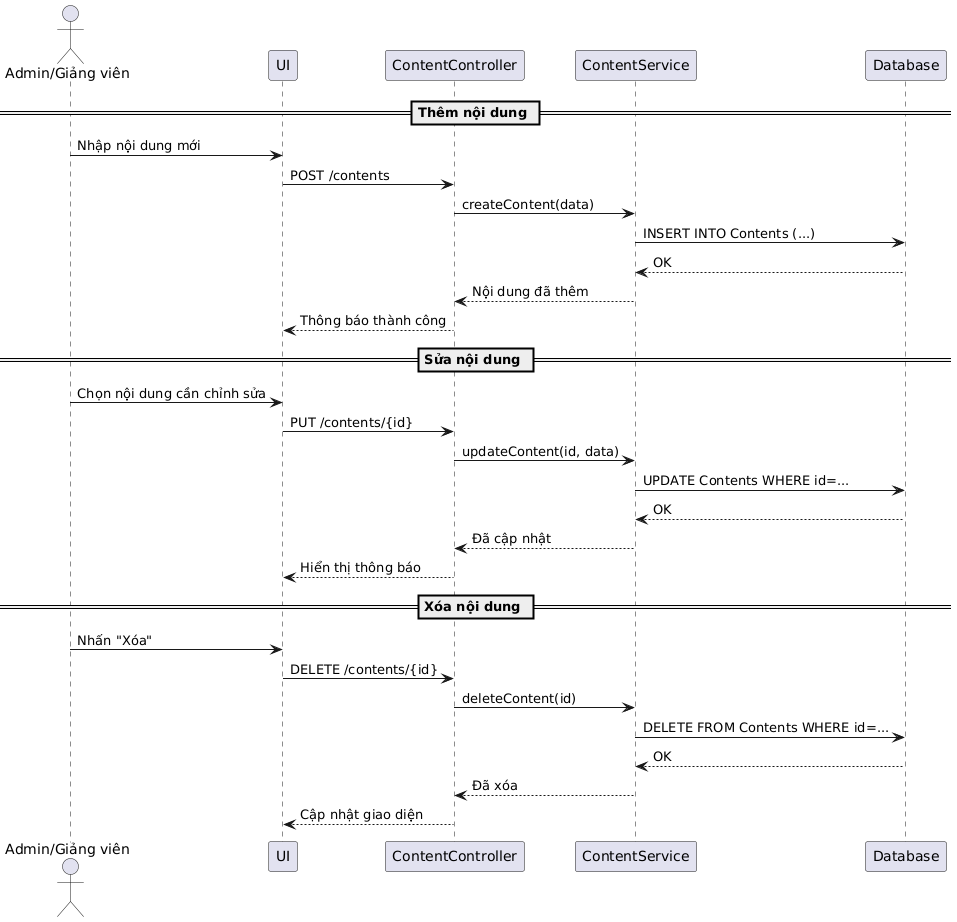
**Hình 3.6. Biểu đồ tuần tự đăng nhập của khách hàng**



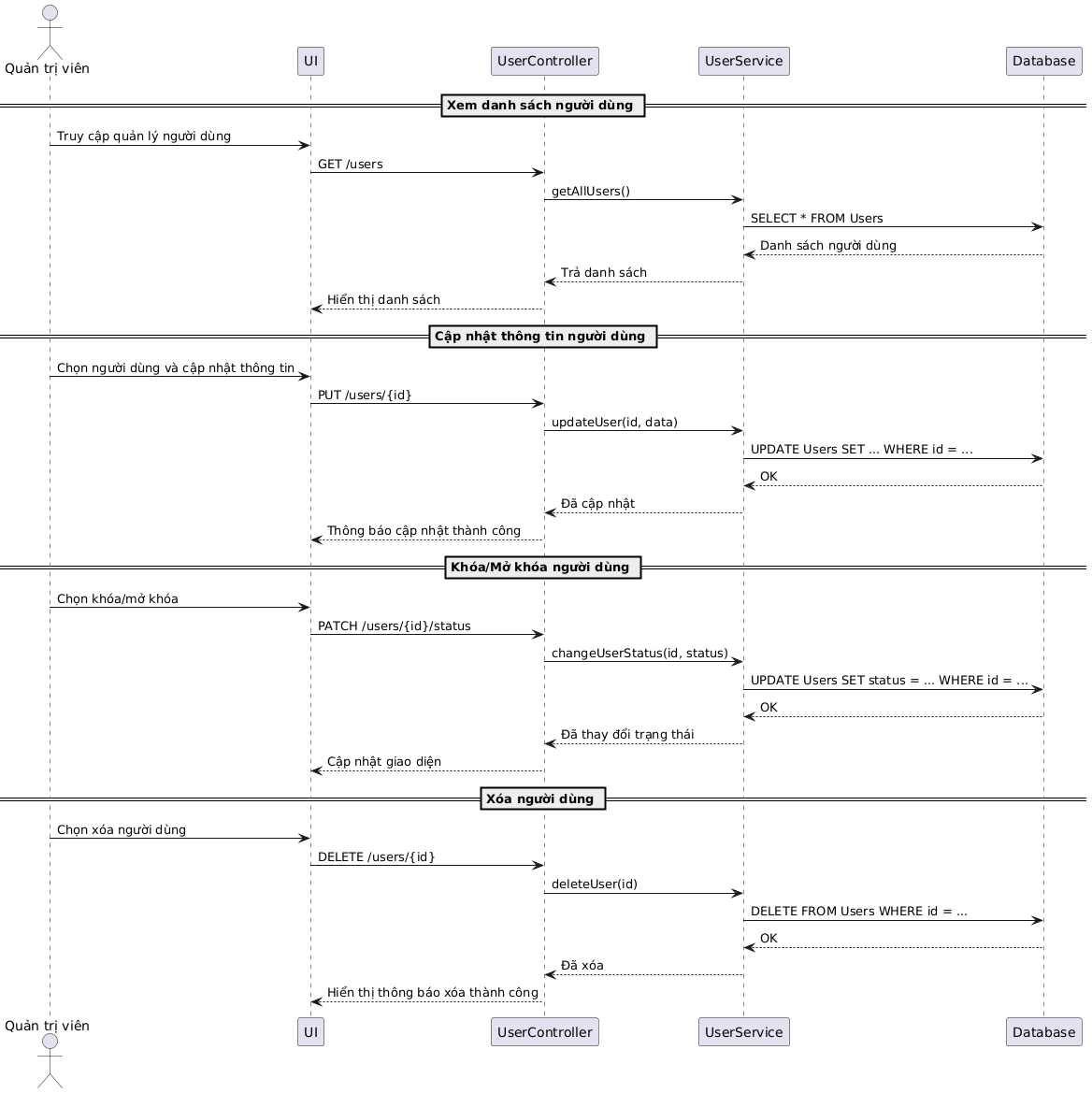
**Hình 3.7. Biểu đồ tuần tự tương tác người dùng**



**Hình 3.8. Biểu đồ tuần tự quản lý nhóm thảo luận**



**Hình 3.9. Biểu đồ tuần tự quản lý nội dung**



**Hình 3.10. Biểu đồ tuần tự quản lý người dùng**

## 3.3 Mô hình hóa cấu trúc

A diagram of a group

AI-generated content may be incorrect.

**Hình 3.11. Biểu đồ lớp phân tích**

# CHƯƠNG 4 : XÂY DỰNG CHƯƠNG TRÌNH

## 4.1. Bảng phân tích cấu trúc các model dữ liệu

**Bảng 4.1. Bảng User**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| username | String | Tên đăng nhập, bắt buộc, duy nhất |
| email | String | Email, bắt buộc, duy nhất |
| password | String | Mật khẩu (tối thiểu 6 ký tự) |
| fullname | String | Họ tên đầy đủ |
| phone | String | Số điện thoại (10 chữ số) |
| address | String | Địa chỉ |
| bio | String | Tiểu sử người dùng |
| points | Number | Điểm tích lũy |
| rank | Enum | Hạng người dùng |
| badge | Object | Huy hiệu đạt được |
| role | Enum | Phân quyền (user/admin) |
| avatar | String | Ảnh đại diện |
| status | Enum | active, inactive, banned |
| lastLogin | Date | Lần đăng nhập gần nhất |

**Bảng 4.2. Bảng Feedback**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| userId | ObjectId | Người gửi phản hồi |
| postId | ObjectId | Bài viết liên quan |
| type | Enum | Loại phản hồi (like, comment...) |
| content | String | Nội dung bình luận |
| image | String | Ảnh đính kèm |
| parentId | ObjectId | Phản hồi cha |
| likes | Array | Danh sách người thích phản hồi |

**Bảng 4.3. Bảng follow**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| follower | ObjectId | Người theo dõi |
| following | ObjectId | Người được theo dõi |

**Bảng 4.4. Friendship**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| userId | ObjectId | Người gửi lời mời |
| friendId | ObjectId | Người được mời kết bạn |
| status | Enum | Trạng thái (pending, accepted...) |
| initiatedBy | ObjectId | Người khởi tạo |

**Bảng 4.5. Bảng Group**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| name | String | Tên nhóm |
| description | String | Mô tả nhóm |
| coverImage | String | Ảnh bìa |
| createdBy | ObjectId | Người tạo |
| members | Array | Danh sách thành viên |
| isPrivate | Boolean | Riêng tư hay không |
| settings | Object | Cài đặt nhóm |
| status | Enum | Trạng thái hoạt động |
| tags | Array | Thẻ liên quan |

**Bảng 4.6. Bảng Message**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| receiverId | ObjectId | Người nhận |
| senderId | ObjectId | Người gửi |
| message | String | Nội dung tin nhắn |
| type | Enum | Loại (text, image...) |
| read | Boolean | Đã đọc hay chưa |
| readAt | Date | Thời điểm đọc |
| status | Enum | Trạng thái gửi |
| clientTempId | String | ID tạm thời từ client |
| deletedBy | Array | Danh sách người đã xóa |

**Bảng 4.7. Bảng Notification**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| userId | ObjectId | Người nhận thông báo |
| message | String | Nội dung thông báo |
| read | Boolean | Đã đọc |
| type | Enum | Loại thông báo |
| relatedId | ObjectId | Đối tượng liên quan |
| sender | Object | Người gửi |

**Bảng 4.8. Bảng UserAcivity**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| userId | ObjectId | Người thực hiện |
| postId | ObjectId | Bài viết liên quan |
| type | Enum | Loại hoạt động |
| searchQuery | String | Truy vấn tìm kiếm |
| targetUserId | ObjectId | Người bị tác động |

**Bảng 4.9. Bảng Post**

|  |  |  |
| --- | --- | --- |
| **Tên thuộc tính** | **Kiểu dữ liệu** | **Mô tả** |
| title | String | Tiêu đề bài viết |
| content | String | Nội dung bài viết |
| author | ObjectId | Tác giả |
| groupId | ObjectId | Nhóm chứa bài viết |
| status | Enum | Trạng thái bài viết |
| tags | Array | Thẻ tag |
| images | Array | Danh sách ảnh |
| deleted | Boolean | Đánh dấu đã xóa |
| views | Number | Lượt xem |
| offensiveContent | Boolean | Nội dung nhạy cảm |

## 4.2. Giao diện chương trình

**A.Giao diện User**

Chức năng: Cho phép người dùng tạo tài khoản mới trong hệ thống.

Quá trình thực hiện:

1. Người dùng nhập thông tin đăng ký (email, họ tên đầy đủ, tên người dùng, mật khẩu)

2. Hệ thống kiểm tra tính hợp lệ của thông tin (định dạng email, độ mạnh của mật khẩu)

3. Hệ thống gửi yêu cầu đăng ký đến server qua API

4. Nếu đăng ký thành công, hệ thống chuyển hướng người dùng đến trang xác thực email

5. Nếu thông tin không hợp lệ, hiển thị thông báo lỗi tương ứng

A screenshot of a login form

AI-generated content may be incorrect.

**Hình 4.1. Trang đăng kí**

Chức năng: Cho phép người dùng đăng nhập vào hệ thống bằng email và mật khẩu.

Quá trình thực hiện:

1. Người dùng nhập email và mật khẩu vào form đăng nhập

2. Hệ thống gửi thông tin đăng nhập đến API để xác thực

3. Nếu thông tin hợp lệ, hệ thống lưu token xác thực và chuyển người dùng đến trang chủ

4. Nếu tài khoản chưa xác thực, chuyển hướng đến trang xác thực email

5. Nếu thông tin không chính xác, hiển thị thông báo lỗi

A screenshot of a login form

AI-generated content may be incorrect.

**Hình 4.2. Trang đăng nhập**

Chức năng: Cho phép người dùng khôi phục mật khẩu khi quên.

Quá trình thực hiện:

1. Người dùng nhập email đã đăng ký vào form

2. Hệ thống gửi yêu cầu đặt lại mật khẩu đến server qua API

3. Server gửi email khôi phục mật khẩu đến địa chỉ email

4. Hệ thống chuyển hướng người dùng đến trang nhập mã xác thực

5. Sau khi xác thực thành công, người dùng có thể đặt mật khẩu mới

*A screenshot of a login box

AI-generated content may be incorrect.*

**Hình 4.3. Trang reset mật khẩu**

Chức năng: Hiển thị các bài viết mới nhất, bài viết được đề xuất, phổ biến,theo dõi và các hoạt động của người dùng và bạn bè.

Quá trình thực hiện:

1. Hệ thống tải dữ liệu bài viết từ server dựa trên bộ lọc đang chọn (đề xuất, mới nhất, phổ biến, theo dõi)

2. Người dùng có thể chuyển đổi giữa các bộ lọc để xem các loại bài viết khác nhau

3. Người dùng có thể tương tác với bài viết (thích, bình luận, chia sẻ)

4. Khi cuộn xuống cuối trang, hệ thống tải thêm bài viết (infinite scrolling)

5. Người dùng có thể tạo bài viết mới bằng cách nhấp vào nút "Tạo bài viết"

*A screen shot of a cell phone

AI-generated content may be incorrect.*

**Hình 4.4. Trang chủ**

Chức năng: Hiển thị các bài viết được đề xuất dựa vào hệ thống gợi ý

Quá trình thực hiện:

2. Người dùng chuyển vào For you hay Của bạn, hệ thống sẽ tự động tạo bài viết dựa vào phân tích hành vi của người dùng ( hệ thống gợi ý trong hệ thống )

3. Người dùng có thể tương tác với bài viết (thích, bình luận, chia sẻ)

4. Khi cuộn xuống cuối trang, hệ thống tải thêm bài viết (infinite scrolling)

*A screen shot of a cell phone

AI-generated content may be incorrect.*

**Hình 4.5. Gợi ý bài viết**

Chức năng: Cung cấp các trò chơi giáo dục và giải trí cho người dùng.

Quá trình thực hiện:

1. Hệ thống hiển thị danh sách các trò chơi có sẵn (Code Challenge, Math Puzzles, Tech Quiz)

2. Mỗi trò chơi được hiển thị với hình ảnh, tiêu đề và mô tả ngắn

3. Người dùng chọn trò chơi muốn tham gia bằng cách nhấp vào trò chơi tương ứng

4. Hệ thống tải giao diện và dữ liệu của trò chơi đã chọn

5. Người dùng tham gia trò chơi và nhận điểm thưởng dựa trên kết quả

*A screenshot of a game

AI-generated content may be incorrect.*

**Hình 4.6. Trang game**

Chức năng: Cho phép người dùng tạo bài viết mới với nội dung, ảnh và các thẻ gắn kèm.

Quá trình thực hiện:

1. Người dùng nhập tiêu đề và nội dung bài viết vào form

2. Người dùng có thể thêm thẻ (tags) bằng cách nhập và nhấn Enter

3. Người dùng có thể đính kèm ảnh bằng cách kéo thả hoặc chọn từ thiết bị (tối đa 10 ảnh)

4. Hệ thống hiển thị xem trước ảnh đã tải lên và cho phép xóa ảnh

5. Khi hoàn tất, người dùng nhấn nút "Đăng" để đăng bài viết

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.7. Trang tạo bài viết**

Chức năng: Cho phép người dùng gửi và nhận tin nhắn với bạn bè.

Quá trình thực hiện:

1. Hệ thống hiển thị danh sách cuộc trò chuyện ở bên trái và nội dung tin nhắn ở bên phải

2. Người dùng chọn một cuộc trò chuyện từ danh sách để xem và gửi tin nhắn

3. Hệ thống tải lịch sử tin nhắn của cuộc trò chuyện đã chọn

4. Người dùng nhập nội dung tin nhắn và gửi

5. Tin nhắn mới được gửi đến người nhận thông qua WebSocket để cập nhật theo thời gian thực

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.8. Trang tin nhắn**

Chức năng: Cho phép người dùng tham gia và tương tác trong các nhóm thảo luận.

Quá trình thực hiện:

1. Hệ thống hiển thị danh sách nhóm mà người dùng tham gia

2. Người dùng có thể tìm kiếm và tham gia các nhóm mới hoặc tạo nhóm mới

3. Khi chọn một nhóm, hệ thống hiển thị các bài viết và thảo luận trong nhóm đó

4. Người dùng có thể đăng bài viết, bình luận và tương tác với nội dung trong nhóm

5. Quản trị viên nhóm có quyền quản lý thành viên và nội dung của nhóm

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.9. Trang nhóm**

Chức năng: Quản lý danh sách bạn bè, lời mời kết bạn và đề xuất kết bạn.

Quá trình thực hiện:

1. Hệ thống hiển thị danh sách bạn bè hiện tại, lời mời kết bạn đang chờ, và đề xuất kết bạn

2. Người dùng có thể tìm kiếm bạn bè theo tên hoặc email

3. Người dùng có thể gửi lời mời kết bạn đến người dùng khác

4. Người dùng có thể chấp nhận hoặc từ chối lời mời kết bạn đã nhận

5. Khi nhấp vào một người bạn, hệ thống hiển thị trang cá nhân của người đó

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.10. Trang bạn bè**

Chức năng: Hiển thị thông tin cá nhân, bài viết, và hoạt động của người dùng.

Quá trình thực hiện:

1. Hệ thống tải thông tin cá nhân và các bài viết của người dùng từ server

2. Trang hiển thị ảnh đại diện, ảnh bìa, thông tin cá nhân, và hoạt động gần đây

3. Người dùng có thể xem các bài viết đã đăng theo thứ tự thời gian

4. Nếu là trang cá nhân của chính mình, người dùng có thể chỉnh sửa thông tin và thay đổi ảnh

5. Người khác có thể xem trang cá nhân, gửi lời mời kết bạn, và tương tác với bài viết

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.11. Trang profile**

Chức năng: Cho phép người dùng cập nhật và quản lý thông tin tài khoản.

Quá trình thực hiện:

1. Hệ thống hiển thị các mục cài đặt theo danh mục (hồ sơ, bảo mật, thông báo, ngôn ngữ)

2. Người dùng có thể cập nhật thông tin cá nhân (họ tên, ngày sinh, giới tính, địa chỉ)

3. Người dùng có thể thay đổi các cài đặt quyền riêng tư và thông báo

4. Người dùng có thể chọn ngôn ngữ hiển thị (Tiếng Việt, Tiếng Anh)

5. Các thay đổi được lưu tự động hoặc khi người dùng nhấn nút "Lưu"

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.12. Trang cài đặt**

**B.Giao diện Admin**

Chức năng: Cung cấp giao diện tổng quan cho quản trị viên về hoạt động của hệ thống.

Quá trình thực hiện:

1. Quản trị viên đăng nhập và truy cập vào trang Dashboard

2. Hệ thống hiển thị sidebar với các chức năng quản trị và khu vực nội dung chính

3. Dashboard hiển thị tổng quan về số lượng người dùng, bài viết, nhóm và hoạt động

4. Các biểu đồ và thống kê hiển thị dữ liệu về lưu lượng truy cập và tương tác

5. Quản trị viên có thể chuyển đổi giữa các tab để quản lý các khía cạnh khác nhau của hệ thống

*A screenshot of a graph

AI-generated content may be incorrect.*

**Hình 4.13. Dashboard**

Chức năng: Cho phép quản trị viên quản lý tài khoản người dùng trong hệ thống.

Quá trình thực hiện:

1. Quản trị viên chọn tab "Quản lý người dùng" từ sidebar

2. Hệ thống hiển thị danh sách tất cả người dùng trong hệ thống dưới dạng bảng

3. Quản trị viên có thể tìm kiếm, lọc và sắp xếp danh sách người dùng

4. Quản trị viên có thể xem chi tiết, khóa/mở khóa hoặc xóa tài khoản người dùng

5. Quản trị viên có thể phân quyền cho người dùng (thành viên, mod, admin)

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.14. Quản lí người dùng**

Chức năng: Cho phép quản trị viên quản lý nội dung bài viết trên hệ thống.

Quá trình thực hiện:

1. Quản trị viên chọn tab "Quản lý bài viết" từ sidebar

2. Hệ thống hiển thị danh sách tất cả bài viết với thông tin tác giả, thời gian đăng

3. Quản trị viên có thể tìm kiếm, lọc bài viết theo tiêu đề, tác giả hoặc nội dung

4. Quản trị viên có thể xem chi tiết, chỉnh sửa hoặc xóa bài viết

5. Quản trị viên có thể xử lý các báo cáo vi phạm từ người dùng về bài viết

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.15. Quản lí bài viết**

Chức năng: Cho phép quản trị viên quản lý hệ thống điểm thưởng của người dùng.

Quá trình thực hiện:

1. Quản trị viên chọn tab "Quản lý điểm" từ sidebar

2. Hệ thống hiển thị danh sách người dùng với số điểm hiện tại và lịch sử giao dịch

3. Quản trị viên có thể tìm kiếm người dùng để quản lý điểm

4. Quản trị viên có thể cộng/trừ điểm cho người dùng với lý do cụ thể

5. Hệ thống ghi lại lịch sử thay đổi điểm để theo dõi và kiểm tra

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.16. Quản lí điểm**

Chức năng: Cho phép quản trị viên quản lý các nhóm thảo luận trong hệ thống.

Quá trình thực hiện:

1. Quản trị viên chọn tab "Quản lý nhóm" từ sidebar

2. Hệ thống hiển thị danh sách tất cả các nhóm với thông tin số lượng thành viên, hoạt động

3. Quản trị viên có thể tìm kiếm nhóm theo tên hoặc chủ đề

4. Quản trị viên có thể xem chi tiết, chỉnh sửa thông tin hoặc xóa nhóm

5. Quản trị viên có thể thay đổi người quản trị nhóm hoặc xử lý các báo cáo vi phạm trong nhóm

*A screenshot of a computer

AI-generated content may be incorrect.*

**Hình 4.17. Quản lí nhóm**

CHƯƠNG 5 : KẾT QUẢ ĐẠT ĐƯỢC

## 5.1. Những thuận lợi và thách thức trong quá trình triển khai

**Thuận lợi**

Trong suốt quá trình nghiên cứu và phát triển hệ thống, một trong những yếu tố then chốt giúp đề tài tiến triển thuận lợi là sự hỗ trợ nhiệt tình, định hướng sát sao từ giảng viên hướng dẫn. Bên cạnh đó, tài nguyên học liệu về các công nghệ sử dụng trong dự án như ReactJS, NodeJS và MongoDB rất phong phú trên các nền tảng học tập mở (Open Source), giúp sinh viên dễ dàng tiếp cận và áp dụng thực tế.

Môi trường học tập tại Trường Đại học Nha Trang cũng là một điều kiện thuận lợi, khi có đầy đủ hệ thống máy tính, tài liệu tham khảo và đặc biệt là sự hợp tác, đóng góp ý kiến tích cực từ bạn bè cùng lớp trong quá trình thử nghiệm hệ thống. Đây là nền tảng quý báu giúp sinh viên vừa học vừa thực hành, rèn luyện được các kỹ năng nghiên cứu, phân tích và lập trình.

**Khó khăn**

Tuy nhiên, đề tài cũng đối mặt với không ít khó khăn. Khó khăn lớn nhất là việc làm chủ đồng thời nhiều công nghệ mới trong thời gian ngắn. Việc tích hợp giữa ReactJS cho frontend, NodeJS cho backend và MongoDB cho lưu trữ đòi hỏi sinh viên không chỉ nắm rõ từng công nghệ riêng lẻ mà còn phải hiểu sâu cách kết nối, đồng bộ dữ liệu và xử lý logic tổng thể trong hệ thống.

Bên cạnh đó, hệ thống gợi ý – phần trọng tâm của đề tài – vẫn còn phụ thuộc nhiều vào các thuật toán cơ bản và chưa đủ dữ liệu hành vi người dùng để đạt hiệu quả tối ưu. Đây là thách thức chung của các hệ thống AI khi chưa có lượng người dùng đủ lớn để mô hình học hỏi và cải thiện dần theo thời gian.

Ngoài ra, việc xây dựng giao diện thân thiện, dễ sử dụng cho nhiều loại thiết bị (desktop, mobile) cũng là một khâu đòi hỏi nhiều thời gian tinh chỉnh và kiểm thử.

## 5.2. Các hạn chế của hệ thống

Mặc dù đề tài đã đạt được nhiều thành tựu nhất định, tuy nhiên vẫn còn tồn tại những điểm hạn chế cần được cải thiện trong các giai đoạn phát triển tiếp theo:

* **Hệ thống gợi ý** hiện tại còn mang tính chất tĩnh, chủ yếu dựa trên các từ khóa hoặc hành vi đơn giản như lượt xem và tìm kiếm gần nhất. Chưa áp dụng các mô hình học sâu (deep learning) hoặc collaborative filtering để đưa ra khuyến nghị cá nhân hóa hiệu quả hơn.
* **Cơ sở dữ liệu người dùng** vẫn còn hạn chế về số lượng và độ phong phú, dẫn đến kết quả gợi ý chưa phản ánh sát nhu cầu học tập đa dạng của sinh viên.
* **Tính năng gamification** – điểm nhấn để tăng sự tương tác – hiện chỉ mới triển khai ở mức cơ bản như điểm thưởng và huy hiệu. Chưa có cơ chế nâng cao như nhiệm vụ, thử thách, hoặc thi đua nhóm.
* Giao diện hệ thống tuy đã hoàn chỉnh nhưng cần được cải thiện thêm về trải nghiệm người dùng (UX), như tốc độ phản hồi, điều hướng mượt mà và cá nhân hóa giao diện theo sở thích người dùng.

**5.3. Định Hướng Phát Triển Trong Tương Lai**

Trong tương lai, để nâng cao chất lượng và hiệu quả của hệ thống, một số định hướng phát triển quan trọng có thể được đề xuất như sau:

* **Tăng cường trí tuệ nhân tạo** cho hệ thống gợi ý bằng cách áp dụng các kỹ thuật như học máy giám sát (supervised learning), học không giám sát (unsupervised learning), và mạng nơ-ron nhân tạo (neural networks) để xây dựng mô hình gợi ý cá nhân hóa hơn dựa trên hành vi và sở thích người dùng.
* **Mở rộng tích hợp đa nền tảng**, bao gồm phiên bản ứng dụng di động (Android/iOS) bên cạnh web để tăng tính linh hoạt và tiện ích cho sinh viên trong việc học mọi lúc, mọi nơi.
* **Xây dựng hệ thống phân tích học tập (Learning Analytics)**, giúp người dùng và quản trị viên có thể theo dõi tiến độ học tập, mức độ tham gia, từ đó đưa ra kế hoạch cải thiện cá nhân hóa việc học.
* **Phát triển thêm các tính năng tương tác xã hội**, như chia sẻ bài viết, bình luận theo chuỗi, hashtag chủ đề, hay tạo sự kiện học tập theo nhóm để tăng tính kết nối cộng đồng.
* **Mở rộng phạm vi triển khai**, không chỉ giới hạn trong Trường Đại học Nha Trang mà hướng đến các trường đại học khác thông qua hệ thống tài khoản liên trường và API mở.

# KẾT LUẬN

Sau quá trình nghiên cứu, xây dựng và thử nghiệm, đề tài "Hệ thống gợi ý cho diễn đàn trao đổi học tập" đã hoàn thiện được một hệ thống nền tảng phục vụ mục tiêu nâng cao hiệu quả học tập và tương tác trong môi trường sinh viên. Với việc tích hợp các công nghệ hiện đại như ReactJS, NodeJS và MongoDB, hệ thống đáp ứng được các yêu cầu cơ bản về quản lý người dùng, tạo nội dung, tìm kiếm và nhận gợi ý học tập.

Đề tài không chỉ dừng lại ở khía cạnh kỹ thuật mà còn có ý nghĩa thực tiễn sâu sắc trong việc ứng dụng công nghệ thông tin vào đổi mới phương pháp học tập, thúc đẩy học tập cộng tác và phát triển năng lực tự học – các yếu tố then chốt trong giáo dục đại học hiện đại.

Tuy vẫn còn những hạn chế nhất định về độ thông minh của hệ thống gợi ý và phạm vi sử dụng, nhưng kết quả đạt được đã mở ra hướng đi tiềm năng cho việc nghiên cứu sâu hơn trong tương lai. Việc mở rộng, cải tiến thuật toán, tích hợp công nghệ mới, và đặc biệt là tăng cường kết nối với các nền tảng học trực tuyến khác sẽ là bước tiến tiếp theo giúp hệ thống trở thành một công cụ học tập hiệu quả, hiện đại và linh hoạt hơn.

Qua quá trình thực hiện đề tài, sinh viên không chỉ nâng cao kiến thức chuyên môn mà còn rèn luyện được kỹ năng giải quyết vấn đề, kỹ năng lập trình, tư duy logic và tinh thần làm việc nghiêm túc – những yếu tố cốt lõi để phát triển nghề nghiệp bền vững trong tương lai.

# TÀi LIỆU THAM KHẢO

[1] Đồng Thị Bích Thủy, Nguyễn Trần Minh Thư, Phạm Thị Bạch Huệ (2022), *Giáo trình Cơ sở dữ liệu*, Nhà xuất bản Khoa học và Kỹ thuật.

[2] Nguyễn Quốc Tuấn (2020), *Lập trình ReactJS từ cơ bản đến nâng cao*, Nhà xuất bản Đại học Quốc gia TP.HCM.

[3] Nguyễn Văn Hiệp (2021), *Giáo trình Hệ điều hành*, Đại học Bách Khoa TP.HCM.

[4] Nguyễn Văn Hiếu (2023), *Hệ thống gợi ý — Machine Learning cho dữ liệu dạng bảng*, [Tài liệu tham khảo nội bộ].

[5] Nguyễn Văn Hùng (2021), *Cơ sở dữ liệu NoSQL và ứng dụng thực tế*, Nhà xuất bản Khoa học và Công nghệ.

[6] Stuart Russel & Peter Norvig (2022), *Trí tuệ nhân tạo: Hướng tiếp cận mới nhất*, Nhà xuất bản Khoa học và Kỹ thuật.

[7] Aggarwal, C. C. (2016), *Recommender Systems: The Textbook*, Springer.

[8] Natural Node (n.d.), *Natural GitHub Repository*, <https://github.com/NaturalNode/natural>

[9] Ricci, F., Rokach, L., & Shapira, B. (2015), *Recommender Systems Handbook*, Springer.

[10] Towards Data Science (2020), *Content-Based Recommendation Systems*, [Online article].