**TRƯỜNG ĐẠI HỌC TRÀ VINH**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

A blue and white logo

AI-generated content may be incorrect.

**ĐỒ ÁN KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG DIỄN ĐÀN TRỰC TUYẾN**

**CHO SINH VIÊN TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giảng viên hướng dẫn : **ThS LÊ MINH TỰ**

Sinh viên thực hiện: **PHẠM ĐĂNG KHOA**

Mã số sinh viên: **110121158**

Lớp **: DA21TTC**

Khoá  **: 2021**

**Trà Vinh, tháng … năm 2025**

**TRƯỜNG ĐẠI HỌC TRÀ VINH**

**KHOA KỸ THUẬT VÀ CÔNG NGHỆ**

A blue and white logo

AI-generated content may be incorrect.

**ĐỒ ÁN KHÓA LUẬN TỐT NGHIỆP**

**XÂY DỰNG DIỄN ĐÀN TRỰC TUYẾN**

**CHO SINH VIÊN TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giảng viên hướng dẫn : **ThS LÊ MINH TỰ**

Sinh viên thực hiện: **PHẠM ĐĂNG KHOA**

Mã số sinh viên: **110121158**

Lớp **: DA21TTC**

Khoá  **: 2021**

**Trà Vinh, tháng … năm 2025**

# LỜI MỞ ĐẦU

## 1. Lý do chọn đề tài

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

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

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

## 2. Mục đích nghiên cứu

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

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

## 3. Đối tượng và phạm vi nghiên cứu

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

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

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

## 4. Phương pháp nghiên cứu

**Phương pháp lý thuyết:**

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

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

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

**Phương pháp thực nghiệm:**

- Thiết kế và xây dựng giao diện người dùng dựa trên các yêu cầu đã phân tích.

- Xây dựng hệ thống backend và cơ sở dữ liệu để xử lý các chức năng như đăng ký, đăng nhập, đăng bài viết, bình luận, phân loại chuyên mục và quản lý người dùng.

- Tiến hành kiểm thử hệ thống trên môi trường giả lập và thực tế để đánh giá hiệu năng, tính ổn định và mức độ thân thiện với người dùng.

- Ghi nhận ý kiến phản hồi từ người dùng thử nghiệm (sinh viên và quản trị viên) để cải tiến hệ thống.

**LỜI CẢM ƠN**

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

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

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

Tôi xin chân thành cảm ơn.

**NHẬN XÉT**

(Của cơ quan thực tập*, nếu có*)

**……………………………………………………………………………..**

**………………………………………………………………………………..**

**………………………………………………………………………………..**

**………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**…………………………………………………………………………………..**

**NHẬN XÉT**

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

...................................................................................................................... **…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

**Giảng viên hướng dẫn**

(ký và ghi rõ họ tên)

UBND TỈNH TRÀ VINH **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**TRƯỜNG ĐẠI HỌC TRÀ VINH Độc lập – Tự do – Hạnh Phúc**

**BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP**

*(Của giảng viên hướng dẫn)*

Họ và tên sinh viên: MSSV:

Ngành: Khóa:

Tên đề tài:

Họ và tên Giáo viên hướng dẫn:

Chức danh: Học vị:

**NHẬN XÉT**

1. Nội dung đề tài:

1. Ưu điểm:

1. Khuyết điểm:

1. Điểm mới đề tài:

1. Giá trị thực trên đề tài:

7. Đề nghị sửa chữa bổ sung:

8. Đánh giá:

Trà Vinh*, ngày tháng năm 20…*

Giảng viên hướng dẫn

*(Ký & ghi rõ họ tên)*

**NHẬN XÉT**

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

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**…………………………………………………………………………….**

**Giảng viên chấm**

(ký và ghi rõ họ tên)

UBND TỈNH TRÀ VINH **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**TRƯỜNG ĐẠI HỌC TRÀ VINH Độc lập - Tự do - Hạnh phúc**

**BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP**

*(Của cán bộ chấm đồ án, khóa luận)*

Họ và tên người nhận xét:

Chức danh: Học vị:

Chuyên ngành:

Cơ quan công tác:

Tên sinh viên:

Tên đề tài đồ án, khóa luận tốt nghiệp:

**I. Ý KIẾN NHẬN XÉT**

1. Nội dung:

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

3. Ứng dụng thực tế:

**II. CÁC VẤN ĐỀ CẦN LÀM RÕ**

(Các câu hỏi của giáo viên phản biện)

**III. KẾT LUẬN**

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

*……………, ngày …… tháng …… năm 20…*

Người nhận xét

*(Ký & ghi rõ họ tên)*

**MỤC LỤC**

[LỜI MỞ ĐẦU 3](#_Toc198675701)

[1. Lý do chọn đề tài 3](#_Toc198675702)

[2. Mục đích nghiên cứu 3](#_Toc198675703)

[3. Đối tượng và phạm vi nghiên cứu 3](#_Toc198675704)

[4. Phương pháp nghiên cứu 4](#_Toc198675705)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 1](#_Toc198675706)

[CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT VÀ CÔNG NGHỆ SỬ DỤNG 3](#_Toc198675707)

[2.1 Javascript 3](#_Toc198675708)

[2.1.1 Sơ lược về ngôn ngữ Javascript 3](#_Toc198675709)

[2.1.2 Đặc điểm 3](#_Toc198675710)

[2.1.3 Cách hoạt động 3](#_Toc198675711)

[2.2 React framework 4](#_Toc198675712)

[2.2.1 Giới thiệu 4](#_Toc198675713)

[2.2.2 Đặc điểm nổi bật 4](#_Toc198675714)

[2.2.3 Các thành phần quan trọng 4](#_Toc198675715)

[2.3 NodeJs 6](#_Toc198675716)

[2.3.1 Giới thiệu 6](#_Toc198675717)

[2.3.2 Đặc điểm vượt trội 6](#_Toc198675718)

[2.3.3 Cơ chế hoạt động 6](#_Toc198675719)

[2.4 Express.js framework 8](#_Toc198675720)

[2.4.1 Giới thiệu 8](#_Toc198675721)

[2.4.2 Các tính năng chính 8](#_Toc198675722)

[2.5 Mô hình MVC 10](#_Toc198675723)

[2.5.1 Tổng quan về mô hình MVC 10](#_Toc198675724)

[2.5.2 Các thành phần 11](#_Toc198675725)

[2.5.3 Ưu điểm và Nhược điểm 12](#_Toc198675726)

[2.6 Material UI (MUI) 13](#_Toc198675727)

[2.6.1 Khái quát về Material UI (MUI) 13](#_Toc198675728)

[2.6.2 Các ưu điểm 13](#_Toc198675729)

[2.7 Hệ quản trị cơ sở dữ liệu MongoDB 13](#_Toc198675730)

[2.7.1 Giới thiệu 13](#_Toc198675731)

[2.7.2 Cách hoạt động 14](#_Toc198675732)

[2.7.3 Ưu điểm và Nhược điểm 15](#_Toc198675733)

[2.8 Chatbot AI 15](#_Toc198675734)

[2.8.1 Tổng quan 15](#_Toc198675735)

[2.8.2 Lợi ích 16](#_Toc198675736)

[2.8.3 Các chatbot AI hiện nay 16](#_Toc198675737)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 17](#_Toc198675738)

[3.1 Mô tả bài toán 17](#_Toc198675739)

[3.2 Xây dựng website 19](#_Toc198675740)

[3.2.1 Mô hình dữ liệu mức quan niệm 19](#_Toc198675741)

[3.2.2 Mô hình dữ liệu mức logic 19](#_Toc198675742)

[3.2.3 Các thực thể 19](#_Toc198675743)

[3.3 Thiết kế xử lý 19](#_Toc198675744)

[3.3.1 Biểu đồ Use Case tổng quát 19](#_Toc198675745)

[3.3.2 19](#_Toc198675746)

[3.3.3 19](#_Toc198675747)

[CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU 20](#_Toc198675748)

[4.1 20](#_Toc198675749)

[4.2 20](#_Toc198675750)

[CHƯƠNG 5: HƯỚNG PHÁT TRIỂN 21](#_Toc198675751)

[5.1 Kết luận 21](#_Toc198675752)

[5.2 Hướng phát triển 21](#_Toc198675753)

***Ghi chú:***

*- In đậm và in hoa tiêu đề của các chương, mục lớn*

*- Chữ số thứ nhất chỉ thứ tự chương*

*- Chữ số thứ 2 chỉ thứ tự mục trong chương*

*- Chữ số thứ 3,…,chỉ thứ tự các tiểu mục*

*- Các tiểu mục trình bày trong mục lục không quá 04 cấp*

**DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

BÁNG 1.1 (size 13)................................................

BẢNG 1.2.................................................................

……...........................................................................

……...........................................................................

……...........................................................................

SƠ ĐỒ 1.1

……........................................................................

……........................................................................

……........................................................................

HÌNH 1.1

…….....................................................................

**……......................................................................**

***Ghi chú:***

***-*** *Xếp sau trang Mục lục*

* *Chữ số thứ nhất chỉ tên chương*
* *Chữ số thứ hai chỉ thứ tự bảng biểu, sơ đồ, hình,…trong mỗi chương*
* *Ở cuối mỗi bảng biểu, sơ đồ, hình,…trong mỗi chương phải có ghi chú, giải thích, nêu rõ nguồn trích hoặc sao chụp,…*

**KÍ HIỆU CÁC CỤM TỪ VIẾT TẮT**

SXKD: Sản xuất kinh doanh

CTM : Chế tạo máy

QTKD………. Quản trị kinh doanh

……….

*Ghi chú:*

*Cụm từ viết viết tắt là các chữ cái và các ký hiệu thay chữ được viết liền nhau, để thay cho một cụm từ có nghĩa thường được lặp nhiều lần trong văn bản hoặc được mọi người mặc nhiên chấp nhận.*

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

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

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

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

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

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

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

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

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

## 2.1 Javascript

### 2.1.1 Sơ lược về ngôn ngữ Javascript

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

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

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

### 2.1.2 Đặc điểm

- Là ngôn ngữ kịch bản (scripting language) hướng đối tượng.

- Là ngôn ngữ nhập động (dynamic typing) dựa trên prototype.

- Là ngôn ngữ lightweight (dễ học và viết code nhanh).

- Là ngôn ngữ đa mô hình (multi-paradigm) và đơn luồng (single threaded).

### 2.1.3 Cách hoạt động

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

- Tích hợp sẵn bên trong hầu hết các trình duyệt web phổ biến.

- Trình duyệt chuyển đổi trang web đó và tất cả các phần tử của nó thành dạng cấu trúc dữ liệu gọi là DOM

- Công cụ JavaScript của trình duyệt thực hiện chuyển mã JavaScript thành mã byte.

- Khi có sự kiện xảy ra, ví dụ như nhấp vào button, việc thực thi code block JavaScript liên kết sẽ được kích hoạt.

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

- Truy cập cơ sở dữ liệu

- Thực hiện các tính toán logic khác nhau

- Phản hồi các sự kiện khác nhau do hệ điều hành của máy chủ kích hoạt [].

## 2.2 React framework

### 2.2.1 Giới thiệu

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

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

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

### 2.2.2 Đặc điểm nổi bật

- Linh hoạt trong thiết kế kiến trúc

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

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

### 2.2.3 Các thành phần quan trọng

- JSX (JavaScript XML) là một cú pháp mở rộng cho phép lập trình viên viết mã giống như [HTML](https://200lab.io/blog/html-la-gi/) trong JavaScript. Trong các ngôn ngữ khác, lập trình viên thường phải viết code HTML và JavaScript riêng rẽ. Tuy nhiên, với JSX, React cho phép kết hợp cả hai trong cùng một mã nguồn, giúp quản lý dễ dàng hơn, đặc biệt là trong các ứng dụng phức tạp.

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

- Liên kết dữ liệu một chiều: trong React, dữ liệu được quản lý và luân chuyển theo hướng từ component cha xuống component con. Các component cha có thể truyền dữ liệu cho component con thông qua props, nhưng các component con không thể trực tiếp thay đổi dữ liệu của component cha [6].

- Các bước cài đặt:

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

+ Mở cmd (Command Prompt) và gõ:

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

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

**npm start** (Chạy ứng dụng React)

A screenshot of a computer

AI-generated content may be incorrect.

## 2.3 NodeJs

### 2.3.1 Giới thiệu

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

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

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

### 2.3.2 Đặc điểm vượt trội

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

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

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

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

Đơn luồng: Nodejs sử dụng mô hình đơn luồng với vòng lặp sự kiện ([Event loop](https://200lab.io/blog/event-loop-la-gi/)), giúp xử lý số lượng request nhiều hơn so với các server truyền thống như Apache HTTP Server.

### 2.3.3 Cơ chế hoạt động

- Nhiều client sẽ gửi request đến server, sẽ được đưa vào Event Queue.

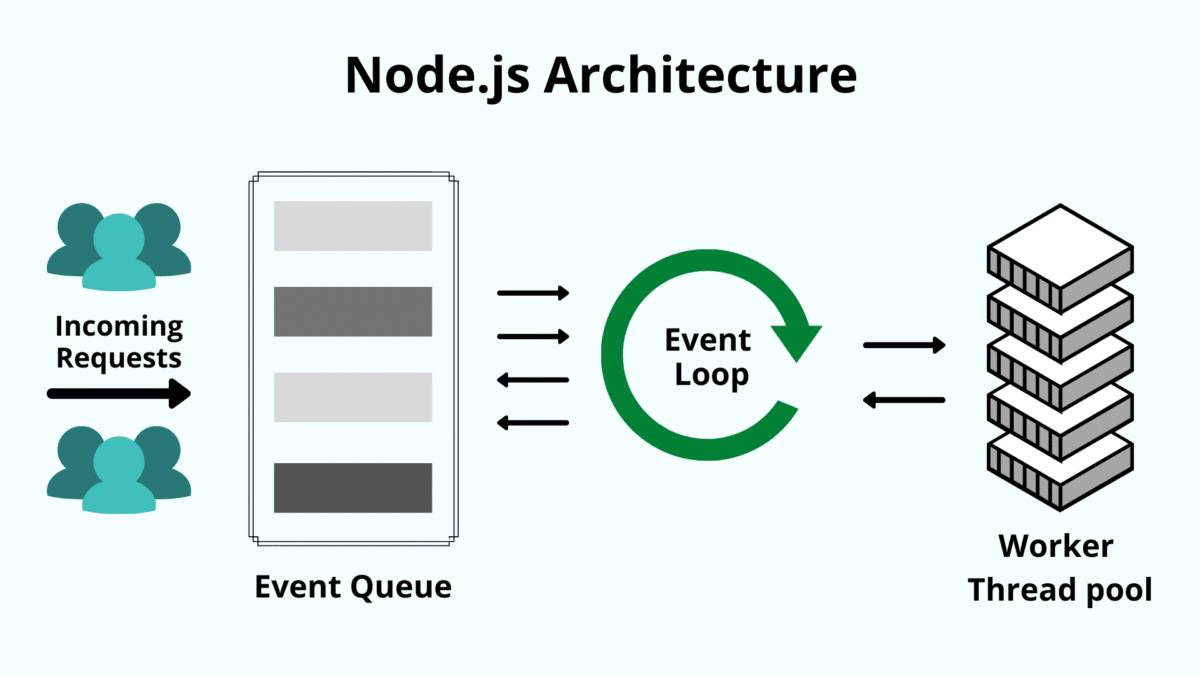
- Event Loop sẽ lấy lần lượt các request từ hàng đợi và xử lý chúng.

- Nếu request cần thực hiện các tác vụ non-blocking I/O, Event Loop sẽ tự động xử lý rồi trả về response.

- Đối với những tác vụ blocking I/O, Event Loop sẽ đưa tasks qua Thread Pool (thông qua thư viện Libuv).

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

- Cuối cùng, response sẽ được phản hồi lại cho các client [7].



- Các bước cài đặt:

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

+ Mở cmd (Command Prompt) và gõ:

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

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

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

A screenshot of a computer

AI-generated content may be incorrect.

## 2.4 Express.js framework

### 2.4.1 Giới thiệu

Express.js là một framework đơn giản được xây dựng trên nền tảng [Node.js](https://200lab.io/blog/nodejs-la-gi/), ra đời với mục đích làm cho việc phát triển các ứng dụng web và [API](https://200lab.io/blog/api-la-gi/) trở nên đơn giản, hiệu quả và dễ bảo trì hơn. Bằng cách cung cấp các công cụ tiện lợi để xử lý yêu cầu HTTP, quản lý định tuyến (routing), và thêm các chức năng khác vào ứng dụng thông qua [middleware](https://200lab.io/blog/middleware-la-gi/).

Với [API RESTful](https://200lab.io/blog/rest-api-la-gi/), lập trình viên có thể dễ dàng hơn và quản lý dữ liệu hiệu quả mà không cần viết quá nhiều code phức tạp. Express.js giúp lập tập trung vào logic ứng dụng thay vì xử lý chi tiết các yêu cầu HTTP.

### 2.4.2 Các tính năng chính

- Templating engines cho phép lập trình viên tạo các mẫu (templates) chứa các placeholder (chỗ trống) để sau đó có thể thay thế bằng dữ liệu thực tế khi trang được hiển thị, giúp tạo ra các trang HTML linh hoạt và dễ dàng hơn, đặc biệt là khi dữ liệu thay đổi thường xuyên.

- Middleware express.static() giúp việc phục vụ các tệp tĩnh trong dự án trở nên đơn giản. Lập trình viên chỉ cần chỉ định thư mục chứa các tệp tĩnh (như hình ảnh, CSS, JavaScript), và Express.js sẽ tự động xử lý việc cung cấp chúng cho người dùng khi họ yêu cầu.

- Tệp tĩnh (static files) là những tệp không cần xử lý phức tạp từ máy chủ, mà chỉ được gửi trực tiếp đến người dùng, như hình ảnh, tệp CSS để tạo kiểu, hoặc tệp JavaScript để chạy trên trình duyệt.

- Routing(Định tuyến) trong Express.js là cách lập trình viên thiết lập các quy tắc để xử lý các yêu cầu HTTP từ người dùng đến đúng nơi trong ứng dụng của lập trình viên. Mỗi URL sẽ được gán với một route (định tuyến), và mỗi route có thể xử lý các loại yêu cầu khác nhau như GET, POST, PUT, DELETE,...

- Middleware là những hàm đặc biệt trong Express.js, được thực thi khi ứng dụng của lập trình viên nhận yêu cầu từ người dùng và trước khi gửi phản hồi về cho họ. Middleware có thể thực hiện nhiều nhiệm vụ như:

+ Ghi lại thông tin về yêu cầu.

+ Xác thực người dùng.

+ Xử lý dữ liệu.

+ Trả về lỗi khi có vấn đề xảy ra [8].

Các bước cài đặt:

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

Kết nối frontend với backend:

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

**{**

**"name": "frontend",**

**"version": "0.1.0",**

**"private": true,**

**"dependencies": {**

**// Các phụ thuộc khác**

**},**

**"proxy":** [**http://localhost:5000**](http://localhost:5000)//Dòng thêm vào

**}**

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

## 2.5 Mô hình MVC

### 2.5.1 Tổng quan về mô hình MVC

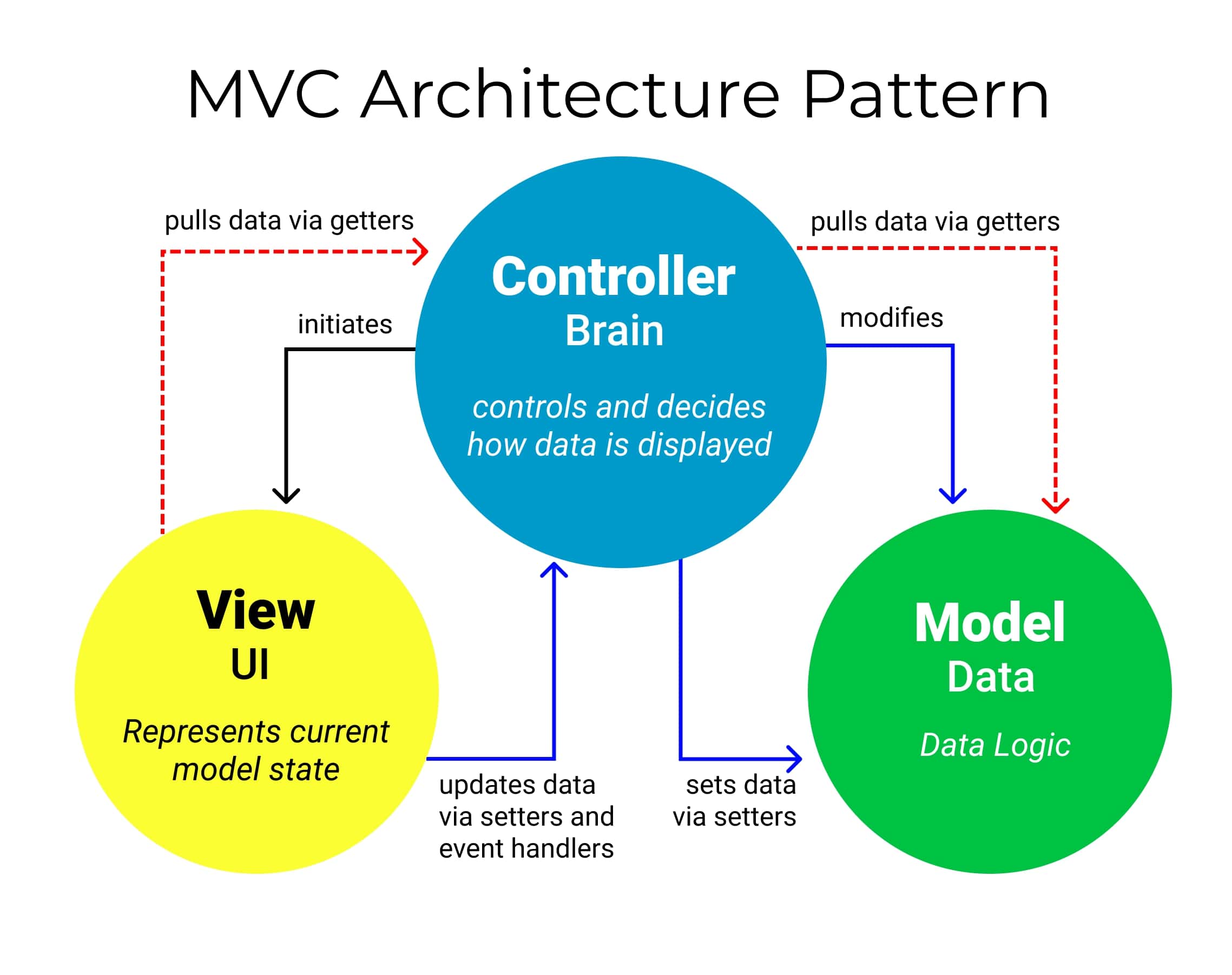
Mô hình MVC, viết tắt của Model-View-Controller, là một kiến trúc phần mềm phổ biến được sử dụng rộng rãi trong phát triển ứng dụng. Được thiết kế để tách biệt logic xử lý dữ liệu (Model), hiển thị giao diện người dùng (View), và điều phối cả hai (Controller), mô hình này mang lại nhiều lợi ích trong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

**Model (M - Model):** Đây là thành phần chịu trách nhiệm cho xử lý dữ liệu và logic kinh doanh của ứng dụng. Model là nơi lưu trữ thông tin, thực hiện các thao tác cập nhật và truy vấn dữ liệu, mà không quan tâm đến cách dữ liệu được hiển thị hoặc tương tác với người dùng.

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

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

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



### 2.5.2 Các thành phần

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

**Trách nhiệm:**

- Lưu trữ và quản lý dữ liệu.

- Thực hiện logic kinh doanh và xử lý sự kiện.

- Cung cấp API cho các thành phần khác để truy xuất và cập nhật dữ liệu.

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

**Trách nhiệm:**

- Hiển thị dữ liệu cho người dùng.

- Tương tác với người dùng thông qua giao diện người dùng.

- Thông báo về các sự kiện tương tác đến Controller.

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

**Trách nhiệm:**

- Nhận các sự kiện từ View.

- Gửi các yêu cầu cập nhật hoặc truy vấn dữ liệu đến Model.

- Cập nhật View dựa trên sự thay đổi trong Model.

- Giữ sự đồng bộ giữa Model và View.

### 2.5.3 Ưu điểm và Nhược điểm

**Ưu điểm:**

+ Tách biệt logic: MVC giúp tách biệt logic ứng dụng thành các phần riêng biệt (Model, View, và Controller), làm cho mã nguồn dễ hiểu và dễ bảo trì.

+ Tái sử dụng mã nguồn:Các thành phần trong MVC có thể được tái sử dụng một cách linh hoạt. Model và View có thể được sử dụng lại trong nhiều phần của ứng dụng hoặc trong các ứng dụng khác nhau.

+ Phân chia công việc: Mô hình này giúp phân chia công việc giữa các thành viên trong nhóm phát triển, với người làm Model tập trung vào logic xử lý dữ liệu, người làm View chịu trách nhiệm về giao diện người dùng, và người làm Controller quản lý sự tương tác.

+ Dễ mở rộng: Do sự tách biệt giữa các thành phần, việc mở rộng và thay đổi ứng dụng trở nên dễ dàng hơn. Một thành phần có thể được thay thế hoặc mở rộng mà không làm ảnh hưởng đến các thành phần khác [9].

**Nhược điểm:**

+ Phức tạp cho ứng dụng nhỏ: Đối với các ứng dụng nhỏ hoặc đơn giản, việc triển khai mô hình MVC có thể quá phức tạp và làm tăng độ phức tạp của mã nguồn.

**+** Tăng độ phức tạp của mã nguồn: Trong một số trường hợp, việc chia nhỏ ứng dụng thành nhiều thành phần có thể làm tăng độ phức tạp của mã nguồn và làm cho mã trở nên khó hiểu [9].

## 2.6 Material UI (MUI)

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

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

### 2.6.2 Các ưu điểm

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

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

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

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

## 2.7 Hệ quản trị cơ sở dữ liệu MongoDB

### 2.7.1 Giới thiệu

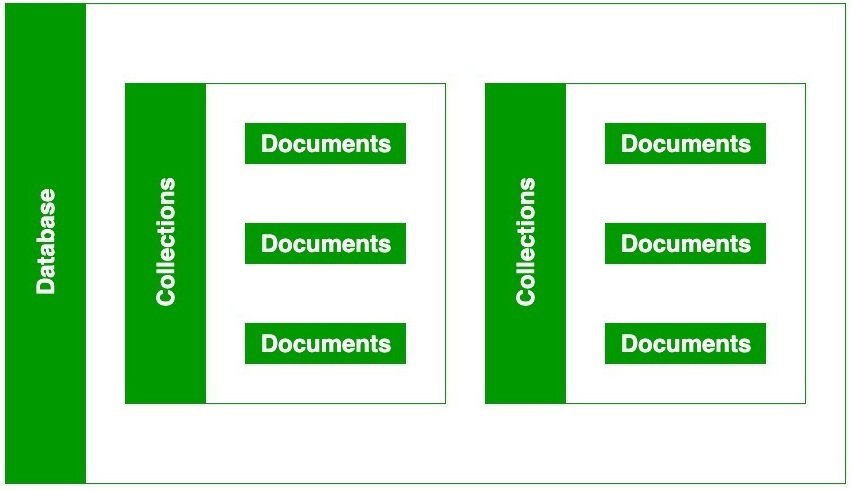
MongoDB là một trong những cơ sở dữ liệu phi quan hệ (NoSQL) phổ biến nhất hiện nay, được sử dụng rộng rãi trong nhiều lĩnh vực. MongoDB cung cấp các tính năng hữu ích giúp cho việc phân tích, truy vấn và cập nhật dữ liệu trở nên dễ dàng và nhanh chóng.

MongoDB hỗ trợ tất cả các ngôn ngữ phổ biến như C, C++, C# và .Net, Go,[Java](https://200lab.io/blog/typescript-la-gi/)**,** Node.js, Perl,[PHP](https://200lab.io/blog/html-la-gi/)**,**[Python](https://200lab.io/blog/python-la-gi/)**,** Motor, Ruby,[Scala](https://200lab.io/blog/scala-la-gi/)**,** Swift, Mongoid [11].

### 2.7.2 Cách hoạt động

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

***GVHD: ThS Lê Minh Tự SVTH: Phạm Đăng Khoa***



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

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

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

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

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

### 2.7.3 Ưu điểm và Nhược điểm

**Ưu điểm:**

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

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

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

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

**Nhược điểm:**

- Cần sử dụng bộ nhớ cao để lưu trữ dữ liệu (data storage).

- Không được phép lưu trữ hơn 16MB data trong tài liệu.

- Data nesting trong BSON cũng bị hạn chế, lập trình viên không được phép nest data quá 100 cấp độ [11].

🡺 Cài đặt tại đường link: [**https://www.mongodb.com/try/download/community**](https://www.mongodb.com/try/download/community)

## 2.8 Chatbot AI

### 2.8.1 Tổng quan

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

### 2.8.2 Lợi ích

- Giảm chi phí

- Trả lời khách hàng nhanh chóng

- Tăng tính chuyên nghiệp

### 2.8.3 Các chatbot AI hiện nay

- Chatbot AI bán hàng

- Chatbot AI chăm sóc khách hàng

- Chatbot trò chuyện

# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## 3.1 Mô tả bài toán

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

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

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

**Yêu cầu về chức năng:**

Đối với sinh viên (người dùng):

- Đăng ký và đăng nhập:

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

- Tạo bài viết chia sẻ:

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

- Bình luận và phản hồi bài viết:

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

- Đánh giá bài viết:

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

- Phân loại bài viết:

* + Khi đăng bài, người dùng có thể chọn chuyên mục (ví dụ: Học tập, Chia sẻ kinh nghiệm, Học bổng…).

- Trò chuyện với chatbot hỗ trợ:

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

Đối với quản trị viên:

- Kiểm duyệt bài viết và bình luận:

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

- Quản lý người dùng:

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

- Quản lý chuyên mục:

* + Thêm, sửa, xóa chuyên mục phù hợp với nhu cầu học tập và xu hướng sinh viên.

- Thống kê và phân tích hoạt động:

* + Thống kê số lượng người dùng, bài viết, chủ đề phổ biến.
  + Phân tích hành vi người dùng như thời gian truy cập, từ khóa tìm kiếm.

- Quản lý chatbot:

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

- Bảo trì và cập nhật hệ thống:

* + Theo dõi lỗi, cập nhật tính năng mới, vá lỗi bảo mật khi cần thiết.

**Yêu cầu phi chức năng:**

 Tính dễ sử dụng:

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

 Tính bảo mật:

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

 Hiệu năng hệ thống:

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

 Khả năng mở rộng:

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

 Khả năng tương thích:

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

 Khả năng bảo trì:

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

## 3.2 Xây dựng website

### 3.2.1 Mô hình dữ liệu mức quan niệm

### 3.2.2 Mô hình dữ liệu mức logic

### 3.2.3 Các thực thể

## 3.3 Thiết kế xử lý

### 3.3.1 Biểu đồ Use Case tổng quát

### 3.3.2

### 3.3.3

# CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

## 4.1

## 4.2

# CHƯƠNG 5: HƯỚNG PHÁT TRIỂN

## 5.1 Kết luận

## 5.2 Hướng phát triển

**PHỤ LỤC**

(Trang Phụ lục kèm theo)

**DANH MỤC TÀI LIỆU THAM KHẢO**

**Tiếng Việt**

1. Quách Ngọc Ân (1992), “Nhìn lại hai năm phát triển lúa lai”, *Di truyền học ứng dụng*, 98(1), tr. 10 – 16.
2. Bộ Nông nghiệp và PTNT (1996), *Báo cáo tổng kết 5 năm (1992 – 1996) Phát triển lúa lai*, Hà Nội.
3. Nguyễn Hữu Đống, Đào Thanh Bằng (1997), Đột biến – Cơ sở lý luận và ứng dụng, Nxb Nông nghiệp, Hà Nội.

**Tiếng Anh**

1. Anderson J. E. (1985), The Relative Inefficiency of Quota, The Cheese Case, *American Economic Review*, 75(1), PP. 178 – 90.
2. Boulding K. E. (1955), *Economic Analysis*, Hamish Hamilton, London.
3. Central Statistical Oraganisation (1995), *Statistical Year Book*, Beijing.
4. FAO (1971), *Agricultural Commodity Projections (1970 – 1980)*, Vol. II. Rome

***Ghi chú:***

* Danh mục tài liệu tham khảo *xếp* ***thứ tự abc theo họ tên tác giả****, sau đó đến tên tài liệu, nơi và năm phát hành:*
* Tác giả là người nước ngoài: xếp thứ tự ABC theo họ tác giả.
* Tác giả là người Việt Nam: xếp thứ tự ABC theo tên tác giả.

Tài liệu không có tên tác giả thì xếp theo thứ tự ABC theo từ đầu của tên cơ quan ban hành báo cáo hay ấn phẩm.

Tài liệu tham khảo là sách, đồ án tốt nghiệp đại học, luận án, báo cáo phải ghi đầy đủ thông tin sau:

* Tên tác giả hoặc cơ quan ban hành (năm xuất bản), *tên sách, đồ án tốt nghiệp đại học, luận án, báo cáo* (in nghiêng, dấu phẩy cuối tên), nhà xuất bản, nơi xuất bản.

Tài liệu tham khảo là bài báo trong tạp chí, bài trong một cuốn sách… phải ghi đầy đủ các thông tin sau:

* Tên tác giả (năm công bố), “Tên bài báo”, *Tạp chí hoặc tên sách,* tập, (số), các số trang.
* ***Danh mục tài liệu tham khảo xếp cuối cùng, sau các trang phụ lục***