

**ĐẠI HỌC ĐÀ NẴNG - TRƯỜNG ĐẠI HỌC SƯ PHẠM**

**KHOA TOÁN – TIN**

**--🙞🙜🕮🙞🙜---**



**BÁO CÁO ĐỒ ÁN CUỐI KÌ**

**Môn: LẬP TRÌNH JAVASRCIPT**

**ĐỀ TÀI : XÂY DỰNG WEBSITE CÂU LẠC BỘ TIN HỌC**

Sinh Viên Thực Hiện: Trần Thị Tuyết Sang – 22CNTT2

Phạm Văn Hiệu – 22CNTT2

Giảng viên hướng dẫn: ThS.Mai Hà Thi

**MỤC LỤC**

*Trang*

[**CHƯƠNG 1 CƠ SỞ LÝ THUYẾT** 7](#_Toc199716152)

[**1.1 Giới thiệu về các công nghệ sử dụng trong đồ án** 7](#_Toc199716153)

[**1.1.1 Visual Studio Code** 7](#_Toc199716154)

[**1.1.2 JavaScipt** 8](#_Toc199716155)

[**1.1.3 Node.js** 9](#_Toc199716156)

[**1.1.4. Mongodb** 14](#_Toc199716157)

[**CHƯƠNG 2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG** 18](#_Toc199716158)

[**2.1 Mô tả bài toán** 18](#_Toc199716159)

[**2.2 Yêu cầu của hệ thống** 18](#_Toc199716160)

[**2.2.1 Yêu cầu chức năng** 18](#_Toc199716161)

[**2.2.2 Yêu cầu phi chức năng** 19](#_Toc199716162)

[**2.3 Phân tích hệ thống** 20](#_Toc199716163)

[**2.3.1 Sơ đồ Use Case** 20](#_Toc199716164)

[**2.4 Thiết Kế Hệ Thống** 23](#_Toc199716165)

[**2.4.1 Kiến trúc hệ thống** 23](#_Toc199716166)

[**2.4.2 Sơ đồ cơ sở dữ liệu** 23](#_Toc199716167)

[**CHƯƠNG 3: THỰC HIỆN XÂY DỰNG WEBSITE, TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ.** 35](#_Toc199716168)

[**3.1 Giao diện và chức năng đăng nhập, đăng ký** 35](#_Toc199716169)

[**3.2 Giao Diện Trang Chủ** 36](#_Toc199716170)

[**3.3 Giao Diện Khóa học** 36](#_Toc199716171)

[**3.4 Thêm, sửa và xóa khóa học** 38](#_Toc199716172)

[**3.5 Thêm điểm Sinh Viên** 40](#_Toc199716173)

[**3.6 Giao diện của danh sách sinh viên hộ trợ của câu lạc bộ** 41](#_Toc199716174)

[**3.7 Giao diện thêm người hỗ trợ** 42](#_Toc199716175)

[**3.8 Giao diện xóa người Hỗ Trợ** 42](#_Toc199716176)

[**3.9 Giao diện kết quả điểm của Sinh viên** 43](#_Toc199716177)

[**3.10 Giao Diện danh sách sinh viên** 43](#_Toc199716178)

[**3.11 Giao Diện Thêm sửa và xóa sinh viên** 44](#_Toc199716179)

[**3.12 Giao diện chi tiết khóa học** 48](#_Toc199716180)

[**3.13 Thêm, xóa học viên và support trong khóa học** 49](#_Toc199716181)

[**KẾT LUẬN** 51](#_Toc199716182)

[**TÀI LIỆU THAM KHẢO** 52](#_Toc199716183)

**DANH MỤC HÌNH ẢNH**

[Hình 1.1: Cách Hoạt Động Của Node J.S 12](#_Toc199715871)

[Hình 1.2: Cơ Chế Hoạt Động Của Event Loop 13](#_Toc199715872)

[Hình 2.1: Quản trị admin 20](#_Toc199715873)

[Hình 2.2: Quản trị user 22](#_Toc199715874)

[Hình 2.3: Mô hình dữ liệu 24](#_Toc199715875)

[Hình 2.4: Quản trị cơ sở dữ liệu 29](#_Toc199715876)

[Hình 2.5: Cơ sở dữ liệu courses 30](#_Toc199715877)

[Hình 2.6: Cơ sở dữ liệu results 31](#_Toc199715878)

[Hình 2.7: Cơ sở dữ liệu students 32](#_Toc199715879)

[Hình 2.8: Cơ sở dữ liệu supports 33](#_Toc199715880)

[Hình 2.9: Cơ sở dữ liệu users 34](#_Toc199715881)

[Hình 3.1: Giao diện đăng ký 35](#_Toc199715882)

[Hình 3.2: Giao diện đăng nhập 36](#_Toc199715883)

[Hình 3.3: Giao diện Home 36](#_Toc199715884)

[Hình 3.4: Giao diện khóa học admin 37](#_Toc199715885)

[Hình 3.5: Giao diện khóa học user 38](#_Toc199715886)

[Hình 3.6: Giao diện thêm khóa học mới 39](#_Toc199715887)

[Hình 3.7: Giao diện cập nhật khóa học 39](#_Toc199715888)

[Hình 3.8: Giao diện xóa khóa học 40](#_Toc199715889)

[Hình 3.9: Giao diện thêm điểm 41](#_Toc199715890)

[Hình 3.10: Giao diện support 42](#_Toc199715891)

[Hình 3.11: Giao diện thêm support 42](#_Toc199715892)

[Hình 3.12: Giao diện xóa support 43](#_Toc199715893)

[Hình 3.14: Giao diện sinh viên 44](#_Toc199715894)

[Hình 3.15: Giao diện thêm sinh viên 45](#_Toc199715895)

[Hình 3.16: Giao diện sửa sinh viên 46](#_Toc199715896)

[Hình 3.17: Giao diện xóa sinh viên 47](#_Toc199715897)

[Hình 3.18: Giao diện thêm sinh viên trong khóa học 48](#_Toc199715898)

[Hình 3.19: Giao diện thêm sinh viên 49](#_Toc199715899)

[Hình 3.20: Giao diện thêm sinh viên 50](#_Toc199715900)

**LỜI NÓI ĐẦU**

Trong thời kỳ chuyển đổi số, nhu cầu xây dựng các nền tảng trực tuyến phục vụ kết nối, học tập và quản lý thông tin ngày càng trở nên cấp thiết. Việc phát triển một website dành riêng cho câu lạc bộ tin học không chỉ tạo điều kiện thuận lợi trong việc tổ chức các hoạt động mà còn góp phần xây dựng môi trường giao lưu, học hỏi, và phát triển kỹ năng về công nghệ.

Báo cáo này tập trung vào quá trình thiết kế và phát triển website cho câu lạc bộ tin học với mục tiêu hỗ trợ quản lý hoạt động hiệu quả, nâng cao trải nghiệm của thành viên và tạo không gian học tập trực tuyến. Nội dung báo cáo được trình bày theo các chương như sau:

Chương 1: Giới thiệu tổng quan về các công nghệ web được sử dụng trong quá trình xây dựng, bao gồm các ngôn ngữ lập trình, cơ sở dữ liệu và các framework phổ biến.

Chương 2: Trình bày quá trình phân tích yêu cầu, thiết kế giao diện người dùng và thiết kế cơ sở dữ liệu cho website.

Chương 3: Thực hiện xây dựng website, triển khai và đánh giá kết quả.

Với báo cáo này, em mong muốn cung cấp một cái nhìn tổng quan và chi tiết về quy trình xây dựng một hệ thống website phục vụ hiệu quả cho hoạt động của câu lạc bộ tin học. Em xin gửi lời cảm ơn chân thành đến cô Mai Hà Thi vì đã nhiệt tình hướng dẫn và hỗ trợ trong suốt quá trình thực hiện đồ án. Dù đã cố gắng hoàn thiện, nhóm em hiểu rằng đồ án vẫn có những thiếu sót. Rất mong nhận được những ý kiến đóng góp từ cô để đồ án có thể được hoàn thiện hơn trong tương lai.

# **CHƯƠNG 1 CƠ SỞ LÝ THUYẾT**

**1.1 Giới thiệu về các công nghệ sử dụng trong đồ án** 

**1.1.1 Visual Studio Code**

**1.1.1.1 Giới Thiệu**

- Visual Studio Code (Viết tắt là VS Code) là một trình soạn thảo mã nguồn mã nguồn mở được phát triển bởi Microsoft, hỗ trợ nhiều ngôn ngữ lập trình và chạy mượt mà trên Windows, macOS và Linux.

- Điểm đặc biệt của VS Code là khả năng tương thích với các thiết bị có cấu hình tầm trung. Nó cung cấp các chức năng như Debug, tích hợp Git, Syntax Highlighting, và tự hoàn thành mã thông minh.

- VS Code cũng có tính năng tùy chỉnh cao, cho phép lập trình viên thay đổi giao diện, phím tắt, và nhiều tùy chọn khác. Mặc dù nhẹ nhàng, VS Code vẫn mạnh mẽ với đầy đủ tính năng, là lựa chọn phổ biến cho lập trình viên.

**1.1.1.2 Ưu Nhược điểm Visual Studio Code**

**Ưu Điểm**

- Đa dạng ngôn ngữ lập trình giúp người dùng thỏa sức sáng tạo và sử dụng như HTML, CSS, JavaScript, C++, …

- Ngôn ngữ, giao diện tối giản, thân thiện, giúp các lập trình viên dễ dàng định hình nội dung.

- Các tiện ích mở rộng rất đa dạng và phong phú.

- Tích hợp các tính năng quan trọng như tính năng bảo mật (Git), khả năng tăng tốc xử lý vòng lặp (Debug), …

- Đơn giản hóa việc tìm quản lý hết tất cả các Code có trên hệ thống.

**Nhược điểm**

- Không phải là một IDE đầy đủ cho các dự án lớn và phức tạp: Mặc dù mạnh mẽ, phần mềm lập trình web VS Code không cung cấp một số tính năng đầy đủ mà một số IDE chuyên biệt có thể cung cấp, như tích hợp dự án hoàn chỉnh cho một số framework.

- Cần cài đặt tiện ích mở rộng cho một số tính năng cụ thể: Để có được một số tính năng cụ thể hoặc hỗ trợ cho ngôn ngữ lập trình và framework riêng, bạn có thể cần cài đặt các tiện ích mở rộng từ cộng đồng

**1.1.2 JavaScipt**

**1.1.2.1 Giới thiệu**

JavaScript (viết tắt là "js") là một ngôn ngữ lập trình mang đầy đủ tính năng của một ngôn ngữ lập trình động mà khi nó được áp dụng vào một tài liệu HTML, nó có thể đem lại khả năng tương tác động trên các trang web. Cha đẻ của ngôn ngữ này là Brendan Eich, đồng sáng lập dự án Mozilla, quỹ Mozilla, và tập đoàn Mozilla.

javaScript thật sự rất linh hoạt.  Các nhà phát triển đã viết ra một số lượng lớn các công cụ thuộc top của core Javascript, mở ra một lượng lớn tính năng bổ sung với ít nỗ lực nhất. Nó bao gồm:

* Giao diện lập trình ứng dụng trên trình duyệt (API) — Các API được xây dựng bên trong các trình duyệt web, cung cấp tính năng như tạo HTML động, cài đặt CSS, thu tập và điều khiển video trực tiếp từ webcam của người dùng hoặc sinh ra đồ dọa 3D và các mẫu audio.
* Các API bên thứ ba cho phép nhà phát triển kết hợp tính năng trong website của họ từ người cung cấp nội dung khác chẳng hạn như Twitter hay Facebook.
* Từ các framework và thư viện bên thứ ba bạn có thể áp dụng tới tài liệu HTML của bạn, cho phép bạn nhanh chóng xây dựng được các trang web và các ứng dụng.

**1.1.2.2 Ưu điểm của JavaScript**

.

Tăng khả năng phản hồi và tốc độ xử lýVới các thao tác được thực hiện trực tiếp trên phía client, người dùng không cần chờ tải lại toàn bộ trang, từ đó cải thiện trải nghiệm và tốc độ truy cập.

Tạo giao diện tương tác cao : Khi kết hợp với HTML và CSS, JavaScript giúp xây dựng các chức năng động, tạo ra những hiệu ứng và phản hồi phong phú, mang lại trải nghiệm hấp dẫn cho người dùng.

Hệ sinh thái thư viện và công cụ phong phú: JavaScript sở hữu hệ thống thư viện mã nguồn mở rất mạnh mẽ, giúp lập trình viên dễ dàng xây dựng và mở rộng chức năng, tiêu biểu như:

Algolia Places: Hỗ trợ điền form tự động cho các trường thông tin như tên, địa chỉ,...

AOS (Animate On Scroll): Tạo hiệu ứng chuyển động mượt mà khi cuộn trang.

Chart.js: Hỗ trợ tạo biểu đồ dữ liệu trực quan.

ReactJS: Thư viện nổi bật trong phát triển giao diện người dùng theo mô hình UI thành phần.

#### **1.2.2.3 Nhược điểm của JavaScript**

Mặc dù JavaScript sở hữu nhiều ưu điểm nổi bật trong phát triển web, ngôn ngữ này vẫn tồn tại một số hạn chế cần được lưu ý:

Rủi ro bảo mật cao: Do JavaScript được thực thi trên phía client (trình duyệt người dùng), nên dễ bị tin tặc khai thác để chèn mã độc hoặc thực hiện các hành vi tấn công như Cross-site Scripting (XSS). Việc thiếu kiểm soát truy cập tệp từ phía máy người dùng khiến JavaScript trở nên kém an toàn hơn so với các ngôn ngữ phía server.

Không tương thích hoàn toàn giữa các trình duyệt: Một số trình duyệt cũ hoặc trình duyệt có cấu hình đặc biệt có thể không hỗ trợ đầy đủ các tính năng JavaScript hiện đại. Điều này có thể gây ra lỗi hiển thị hoặc làm gián đoạn chức năng trên website nếu không được kiểm tra và tối ưu hóa kỹ lưỡng.

Khả năng hiển thị không đồng nhất: Cùng một đoạn mã JavaScript có thể cho ra các kết quả hiển thị khác nhau trên các thiết bị và trình duyệt khác nhau. Do đó, các nhà phát triển phải tốn thêm công sức để kiểm thử và điều chỉnh giao diện nhằm đảm bảo tính nhất quán cho người dùng

**1.1.3 Node.js**

**1.1.3.1 Giới Thiệu**

**- Node.js** là một môi trường chạy (runtime) JavaScript/TypeScript được xây dựng trên công cụ JavaScript V8 của Google Chrome. Mặc dù Node.js có thể được sử dụng để xây dựng các ứng dụng độc lập, nhưng nó thường được dùng chủ yếu để phát triển các dịch vụ phụ trợ (backend services).

- Ứng dụng Node.js chạy trong một tiến trình duy nhất, không tạo luồng mới cho mọi yêu cầu. Node.js cung cấp một tập hợp các nguyên hàm I/O không đồng bộ trong thư viện chuẩn của nó để ngăn chặn mã JavaScript khỏi việc chặn và nói chung, các thư viện trong Node.js được viết bằng các mô hình không chặn, khiến hành vi chặn trở thành ngoại lệ chứ không phải là chuẩn mực.

**1.1.3.2 Ưu Nhược Điểm Node.Js**

**Ưu Điểm**

Node.js là khả năng xây dựng các dịch vụ siêu nhanh và có khả năng mở rộng cao. Chính vì vậy, nó đã được rất nhiều công ty lớn trên thế giới như Netflix, LinkedIn, PayPal, Trello, Uber, eBay và NASA tin dùng trong các hệ thống của họ.

Node.js cho phép các nhà phát triển giao diện người dùng (frontend developers) tận dụng những kỹ năng JavaScript hiện có để mở rộng ra phát triển ứng dụng full-stack, giúp họ dễ dàng chuyển sang xây dựng cả phần frontend và backend của hệ thống.

Ngoài ra, Node.js sở hữu một trong những hệ sinh thái thư viện mã nguồn mở lớn nhất hiện nay. Điều này có nghĩa là nếu bạn muốn thêm bất kỳ tính năng nào vào ứng dụng của mình, rất có thể bạn sẽ tìm thấy thư viện miễn phí, mã nguồn mở phù hợp. Điều này giúp bạn tiết kiệm thời gian, không phải bắt đầu từ con số 0, mà thay vào đó có thể tập trung vào việc phát triển các tính năng cốt lõi của ứng dụng.

Có khả năng xử lý hàng ngàn Process cho hiệu suất đạt mức tối ưu nhất.

Phù hợp để xây dựng những ứng dụng thời gian thực như các ứng dụng chat, mạng xã hội …

**Nhược Điểm**

- Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.

- Khó thao tác với cơ sử dữ liệu quan hệ.

- Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác

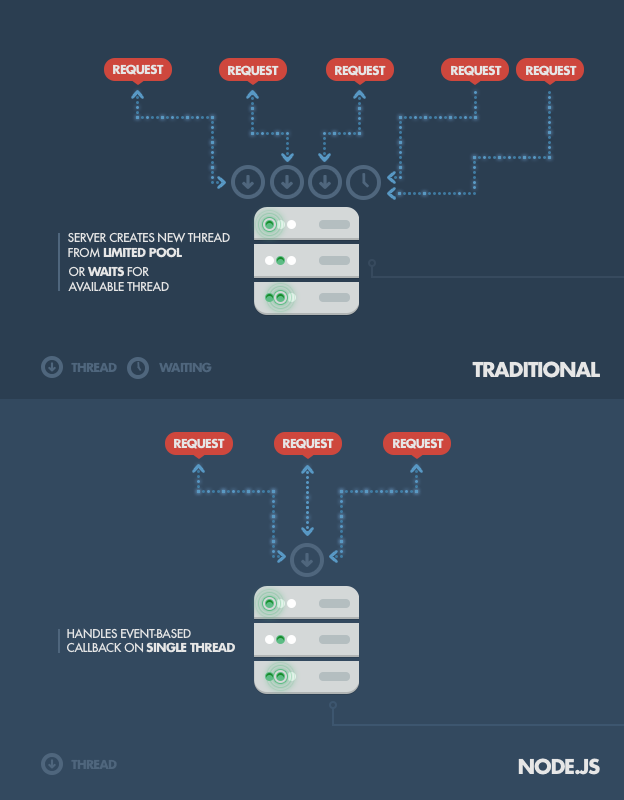
- Cần có kiến thức tốt về JavaScript.

- Không phù hợp với các tác vụ đòi hỏi nhiều CPU.

**Node.Js Hoạt Động Như Thế Nào?**

Ý tưởng chính của Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao.

Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.



Hình 1.1: Cách Hoạt Động Của Node J.S

**Giới thiệu về Event Loop**

**Event Loop** là một trong những khái niệm quan trọng nhất trong Node.js, vì nó giải thích cách Node.js xử lý các tác vụ đồng thời mặc dù nó chỉ chạy trên một luồng duy nhất (single-threaded). Node.js sử dụng event loop để quản lý các tác vụ bất đồng bộ như đọc/ghi file, yêu cầu HTTP, hoặc truy vấn cơ sở dữ liệu mà không chặn (blocking) luồng chính.

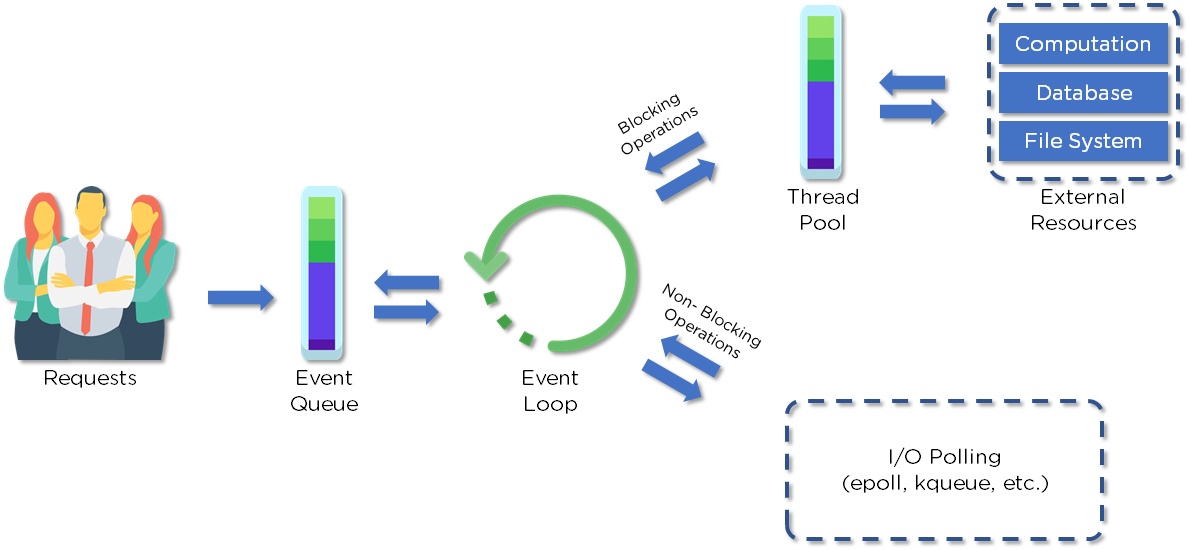
Trong Node.js, mã JavaScript chạy trong một luồng duy nhất, còn được gọi là luồng chính (main thread). Tuy nhiên, để xử lý các yêu cầu I/O không đồng bộ, như đọc và ghi vào tệp, gọi API mạng hoặc truy vấn cơ sở dữ liệu, Node.js sử dụng mô hình sự kiện và non-blocking I/O.

Client gửi các REQUEST đến SERVER để tương tác với ứng dụng web. Các REQUESTs này có thể là Blocking hoặc Non-Blocking

* Truy vấn dữ liệu
* Xóa dữ liệu
* Cập nhật dữ liệu

Node.JS tiếp nhận các Request gửi đến và thêm chúng vào hàng đợi Event Queue

Sau đó các yêu cầu Request này được xử lý lần lượt thông qua Event Loop.



Hình 1.2: Cơ Chế Hoạt Động Của Event Loop

Cơ chế hoạt động của **Event Loop**:

1. **Requests (Yêu cầu)**: Các yêu cầu từ người dùng được đưa vào **Hàng đợi sự kiện (Event Queue)**. Đây là nơi lưu trữ các sự kiện cần được xử lý.
2. **Event Loop (Vòng lặp sự kiện)**: Event Loop liên tục kiểm tra hàng đợi sự kiện để xem có sự kiện nào cần được xử lý hay không. Nếu có, nó sẽ lấy sự kiện ra khỏi hàng đợi và chuyển nó tới **Thread Pool** để xử lý. Các hoạt động không chặn (non-blocking operations) sẽ được xử lý trực tiếp, trong khi các hoạt động chặn (blocking operations) sẽ được gửi tới các nguồn tài nguyên bên ngoài như cơ sở dữ liệu, hệ thống tệp, v.v.
3. **Thread Pool (Nhóm luồng)**: Thread Pool xử lý các hoạt động chặn. Nó sẽ tạo ra các luồng riêng biệt để xử lý các tác vụ này.
4. **I/O Polling (Kiểm tra I/O)**: Có một hộp "I/O Polling" kết nối với Thread Pool, đại diện cho các cơ chế kiểm tra I/O như epoll, kqueue, v.v.
5. **External Resources (Nguồn tài nguyên bên ngoài)**: Khi công việc đã được hoàn thành, kết quả sau cùng sẽ được trả lại thông qua Event Loop và gửi lại cho người dùng.

Cơ chế này giúp Node.js xử lý nhiều yêu cầu một cách hiệu quả, đồng thời duy trì tính không chặn của ứng dụng

**Cách hoạt động của Event Loop**

Node.js hoạt động dựa trên một vòng lặp sự kiện (event loop) để xử lý tất cả các tác vụ bất đồng bộ. Khi một tác vụ (như đọc file) được yêu cầu, nó được đưa vào hàng đợi của event loop. Node.js tiếp tục xử lý các yêu cầu khác trong khi chờ tác vụ đó hoàn thành. Khi tác vụ hoàn thành, callback tương ứng sẽ được đưa vào hàng đợi và thực thi trong lần lặp tiếp theo của event loop.

Event loop có 6 giai đoạn chính:

* **Timers**: Xử lý các callback từ setTimeout() và setInterval().
* **I/O Callbacks**: Xử lý các callback từ các hoạt động I/O như đọc file hoặc gửi HTTP request.
* **Idle, prepare**: Sử dụng nội bộ, ít khi được quan tâm.
* **Poll**: Chờ và xử lý các sự kiện I/O mới.
* **Check**: Xử lý các callback từ setImmediate().
* **Close Callbacks**: Xử lý các callback từ sự kiện close() của các kết nối I/O.

**1.1.4. Mongodb**

**1.1.4.1 Giới Thiệu Về Mongodb**

- MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSql và được hàng triệu người sử dụng.

- MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JSON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh.

- Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MySQL hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng

- So với RDBMS thì trong MongoDB collection ứng với table, còn document sẽ ứng với row, MongoDB sẽ dùng các document thay cho row trong RDBMS.

- Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

- Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

**1.1.4.2 Ưu Nhược Điểm Của Mongodb**

**Ưu điểm của MongoDB**

- Dữ liệu lưu trữ phi cấu trúc, không có tính ràng buộc, toàn vẹn nên tính sẵn sàng cao, hiệu suất lớn và dễ dàng mở rộng lưu trữ.

- Dữ liệu được caching (ghi đệm) lên RAM, hạn chế truy cập vào ổ cứng nên tốc độ đọc và ghi cao

**Nhược điểm của MongoDB**

- Không ứng dụng được cho các mô hình giao dịch nào có yêu cầu độ chính xác cao do không có ràng buộc.

- Không có cơ chế transaction (giao dịch) để phục vụ các ứng dụng ngân hàng.

- Dữ liệu lấy RAM làm trọng tâm hoạt động vì vậy khi hoạt động yêu cầu một bộ nhớ RAM lớn.

- Mọi thay đổi về dữ liệu mặc định đều chưa được ghi xuống ổ cứng ngay lập tức vì vậy khả năng bị mất dữ liệu từ nguyên nhân mất điện đột xuất là rất cao.

**Khi nào sử dụng MongoDB?**

- Quản lý và truyền tải content – Quản lý đa dạng nhiều product của content chỉ trong một kho lưu trữ data cho phép thay đổi và phản hồi nhanh chóng mà không chịu thêm phức tạp thêm từ hệ thống content.

- Cấu trúc Mobile và Social – MongoDB cung cấp một platform có sẵn, phản xạ nhanh, và dễ mở rộng cho phép rất nhiều khả năng đột phá, phân tích real-time, và hỗ trợ toàn cầu.

- Quản lý data khách hàng – Tận dụng khả năng query nhanh chóng cho phân tích real-time trên cơ sở dữ liệu người dùng cực lớn vớ các mô hình data phức tạp bằng các schema linh hoạt và tự động sharding cho mở rộng chiều ngang.

**Tích hợp MongoDB vào NodeJs**

Sử dụng MongoDB qua thư viện Mongoose (Database ORM) giúp thao tác dễ hơn về mặt cú pháp:

**npm install mongoose --save**

**yarn add mongoose --save**

**Các câu lệnh cơ bản của MongoDB**

* **Model.deleteMany():** Xóa nhiều
* **Model.deleteOne():** Xóa một
* **Model.find():** Tìm kiếm nhiều documents
* **Model.findById():** Tìm kiếm bởi ID
* **Model.findByIdAndDelete():** Tìm bởi ID nếu thấy thì Xóa
* **Model.findByIdAndRemove():** Tìm bởi ID nếu thấy thì Xóa
* **Model.findByIdAndUpdate():** Tìm bởi ID nếu thấy thì Update
* **Model.findOne():** Tìm một documents
* **Model.findOneAndDelete():** Tìm một documents và xóa
* **Model.findOneAndRemove():** Tìm một documents và xóa
* **Model.findOneAndReplace():** Tìm một documents và thay thế
* **Model.findOneAndUpdate():** Tìm một documents và update
* **Model.replaceOne():** Thay thế một document
* **Model.updateMany():** Update nhiều documents
* **Model.updateOne():** Update một document

# **CHƯƠNG 2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG**

**2.1 Mô tả bài toán**

Trong một hệ thống quản lý người dùng, chức năng đăng nhập, đăng ký và phân quyền là những yếu tố cốt lõi giúp đảm bảo tính bảo mật và khả năng kiểm soát truy cập. Mô-duyn này có mục tiêu xây dựng một cơ chế quản lý người dùng hiệu quả với các chức năng sau:

+ Đăng ký tài khoản: Hệ thống sẽ cho phép người dùng tạo tài khoản thông qua một biểu mẫu đăng ký. Người dùng cần cung cấp thông tin cá nhân cơ bản như: Tên đầy đủ, email, mật khẩu, các thông tin bổ sung khác.

+ Xác thực thông tin người dùng: Hệ thống sẽ thực hiện chức năng đăng nhập, trong đó người dùng cần nhập địa chỉ email và mật khẩu đã đăng ký. Hệ thống sẽ xác thực thông tin để cho phép người dùng truy cập vào hệ thống. Nếu thông tin không chính xác, người dùng sẽ nhận được thông báo lỗi phù hợp.

Hệ thống cần đảm bảo:

* Tính bảo mật cao: Hệ thống phải đảm bảo rằng tất cả thông tin nhạy cảm (như mật khẩu và token xác thực) đều được mã hóa trước khi lưu trữ. Việc sử dụng các thuật toán mã hóa mạnh và phương thức bảo mật hiện đại là cần thiết để ngăn chặn các cuộc tấn công và lộ thông tin.
* Giao diện thân thiện: Giao diện của hệ thống cần thân thiện, dễ sử dụng và trực quan với người dùng cuối. Việc tổ chức bố cục hợp lý và chỉ dẫn rõ ràng sẽ giúp người dùng dễ dàng thực hiện các thao tác đăng ký và đăng nhập.
* Kiến trúc linh hoạt, hỗ trợ mở rộng khi cần thêm tính năng hoặc tích hợp hệ thống.

**2.2 Yêu cầu của hệ thống**

**2.2.1 Yêu cầu chức năng**

Hệ thống cần đáp ứng các chức năng chính sau:

* Quản lý khóa học:
* Thêm, xóa, sửa thông tin: Cung cấp giao diện để người dùng có thể thêm mới, chỉnh sửa thông tin hoặc xóa các khóa học không còn hoạt động.
* Hiển thị danh sách khóa học: Hiển thị danh sách tất cả các khóa học hiện có theo các tiêu chí như thời gian, tên khóa học và tình trạng.
* Quản lý học viên:
* Thêm mới, chỉnh sửa và xóa thông tin học viên: Cho phép nhập thông tin cho học viên mới, cập nhật thông tin cá nhân và xóa học viên nếu cần.
* Tìm kiếm học viên: Hệ thống cho phép tìm kiếm học viên theo tên hoặc mã số sinh viên để dễ dàng theo dõi và quản lý.
* Quản lý giáo viên:
* Quản lý thông tin giáo viên: Cung cấp chức năng để thêm mới, chỉnh sửa, và xóa thông tin của giáo viên, bao gồm tên, bộ môn, và kinh nghiệm.
* Gán giáo viên cho khóa học: Cho phép phân công giáo viên cho các khóa học tương ứng để quản lý thông tin giảng dạy.
* Quản lý kết quả khóa học:
* Lưu trữ điểm và xếp loại học viên: Hệ thống sẽ ghi nhận điểm số cuối kỳ và xếp loại học tập cho mỗi học viên.
* Xuất báo cáo kết quả: Cung cấp chức năng để xuất và in báo cáo kết quả các khóa học nhằm theo dõi sự tiến bộ của học viên.

**2.2.2 Yêu cầu phi chức năng**

+ Bảo mật: Hệ thống sẽ sử dụng thuật toán BCrypt để mã hóa mật khẩu trước khi lưu trữ. BCrypt không chỉ cung cấp khả năng mã hóa một chiều mà còn tự động tạo ra các giá trị salt, giúp tăng cường bảo mật cho thông tin nhạy cảm.

+ Hiệu suất: Hệ thống phải đảm bảo rằng các thao tác đăng nhập và đăng ký được xử lý nhanh chóng, ngay cả khi có nhiều người dùng truy cập đồng thời. Hệ thống cần có khả năng xử lý ít nhất 100 yêu cầu đăng nhập hoặc đăng ký mỗi giây mà không làm giảm hiệu suất, sử dụng các kỹ thuật tối ưu hóa như caching và load balancing nếu cần.

+ Tương thích: Hệ thống phải hỗ trợ truy cập trên các thiết bị khác nhau, bao gồm máy tính để bàn, laptop, tablet và smartphone. Giao diện người dùng cần phải phản hồi nhanh và tối ưu hóa cho các kích thước màn hình khác nhau.

**2.3 Phân tích hệ thống**

**2.3.1 Sơ đồ Use Case**

**Admin:**

Sơ đồ Use Case (hình ảnh hoặc mô tả) là một công cụ quan trọng giúp hình dung các chức năng của hệ thống và mối quan hệ giữa người dùng (actor) và hệ thống. Dưới đây là cách trình bày sơ đồ Use Case cho hệ thống quản lý người dùng, bao gồm các chức năng chính như đăng ký, đăng nhập, phân quyền và các chức năng liên quan.

A diagram of a diagram

AI-generated content may be incorrect.

Hình 2.1: Quản trị admin

Admin: Người quản lý hệ thống với quyền hạn cao nhất, có khả năng thực hiện nhiều chức năng quản lý khác nhau.

Use Cases

* Đăng nhập: Quản trị viên được cung cấp tài khoản admin mới và đăng nhập vào hệ thống.
* Đăng xuất: Quản trị viên có thể đăng xuất khỏi hệ thống để bảo mật thông tin cá nhân.
* Thêm (support, student, khóa học, kết quả): Admin có thể thêm thông tin mới cho các đối tượng trong hệ thống như hỗ trợ, sinh viên, khóa học, và kết quả học tập.
* Sửa (support, student, khóa học, kết quả): Admin có thể chỉnh sửa thông tin cho các đối tượng đã có trong hệ thống.
* Xóa (support, student, khóa học, kết quả): Admin có khả năng xóa thông tin không còn cần thiết hoặc không chính xác.
* Xử lý lỗi đăng nhập: Hệ thống có cơ chế để quản trị viên xử lý các lỗi xảy ra trong quá trình đăng nhập, chẳng hạn như quên mật khẩu hoặc tài khoản bị khóa.
* Thay đổi giao diện: Admin có thể điều chỉnh hoặc tùy chỉnh giao diện của hệ thống để nâng cao trải nghiệm người dùng.

**User:**

A diagram of a student

AI-generated content may be incorrect.

Hình 2.2: Quản trị user

Sơ đồ Use Case là một biểu đồ minh họa các tương tác giữa User và một hệ thống quản lý “Câu lạc bộ Tin Học”, cung cấp cái nhìn tổng quan về các chức năng chính và mối quan hệ giữa chúng. Trong đó, User đóng vai trò là tác nhân chính là sinh viên thực hiện các hoạt động như đăng ký sinh viên, đăng ký khóa học, truy cập thông tin và thoát hệ thống. Hệ thống được thiết kế với các chức năng cho user bao gồm Thêm student, Đăng ký, đăng nhập, Đăng ký khóa học, Xem thông tin khóa học, support, kết quả, và Đăng xuất, tất cả đều được liên kết chặt chẽ thông qua mối quan hệ <<include>>, cho thấy tính bắt buộc của việc đăng nhập trước khi thực hiện bất kỳ chức năng nào khác.

Về mặt chức năng, Thêm student cho phép người dùng thêm thông tin sinh viên mới, trong khi Đăng ký, đăng nhập đóng vai trò như bước xác thực danh tính, là nền tảng bắt buộc để truy cập các dịch vụ khác. Đăng ký khóa học hỗ trợ người dùng tham gia các khóa học phù hợp, và Xem thông tin khóa học, support, kết quả cung cấp thông tin chi tiết về lịch học, hỗ trợ học tập, cũng như kết quả học tập như điểm số và tiến độ. Cuối cùng, Đăng xuất đảm bảo an toàn bằng cách cho phép người dùng kết thúc phiên làm việc một cách an toàn. Mối quan hệ <<include>> giữa các chức năng này với Đăng ký, đăng nhập nhấn mạnh tính bảo mật của hệ thống, đảm bảo rằng chỉ người dùng đã xác thực mới có thể truy cập dữ liệu nhạy cảm.

**2.4 Thiết Kế Hệ Thống**

**2.4.1 Kiến trúc hệ thống**

Kiến trúc hệ thống được thiết kế với sự tích hợp chặt chẽ giữa các thành phần Frontend, Backend, và Cơ sở dữ liệu, nhằm tạo ra một hệ thống quản lý học tập linh hoạt và hiệu quả.

**Frontend:** được xây dựng bằng HTML, CSS, và JavaScript, đóng vai trò cung cấp giao diện người dùng trực quan và tương tác. HTML định hình cấu trúc các trang như đăng ký, đăng nhập, và quản lý thông tin, trong khi CSS đảm bảo thiết kế thẩm mỹ và đáp ứng. JavaScript, với các công cụ như fetch API hoặc Axios, mang lại tính năng động, cho phép gửi yêu cầu HTTP (GET, POST) đến backend một cách hiệu quả, tạo cầu nối giữa người dùng và hệ thống.

**Backend:** được phát triển bằng Node.js thuần , xử lý toàn bộ logic kinh doanh và các yêu cầu từ frontend. Đây là nơi quản lý phiên làm việc, thực thi các chức năng như đăng ký, đăng nhập, và đảm bảo luồng dữ liệu thông suốt. Về bảo mật, backend tích hợp các giải pháp như JSON Web Tokens (JWT) cho xác thực người dùng và các thư viện như Passport.js hoặc bcrypt để mã hóa mật khẩu, đảm bảo chỉ người dùng được ủy quyền mới truy cập các chức năng nhạy cảm.

**Cơ sở dữ liệu:** sử dụng MongoDB Compass với mô hình NoSQL, chịu trách nhiệm lưu trữ và quản lý dữ liệu hệ thống. Các collection như "users" (thông tin người dùng), "roles" (vai trò), và "permissions" (quyền truy cập) được thiết kế để hỗ trợ truy xuất và quản lý thông tin một cách linh hoạt, phù hợp với đặc thù của hệ thống quản lý học tập.

**2.4.2 Sơ đồ cơ sở dữ liệu**

**2.4.2.1 Mô hình ERD**

**A screenshot of a computer

AI-generated content may be incorrect.**

Hình 2.3: Mô hình dữ liệu

**1. Phân tích các bảng trong CSDL**

* Bảng supports:
  + Lưu trữ thông tin hỗ trợ hoặc quản trị viên liên quan đến các khóa học.
  + Cột:
    - MaSupport (VARCHAR(20)): Khóa chính, mã định danh hỗ trợ.
    - HoTen (VARCHAR(100)): Họ và tên.
    - Email (VARCHAR(100)): Địa chỉ email.
    - LopSinhHoat (VARCHAR(5...)): Lớp sinh hoạt.
    - SoDienThoai (VARCHAR(11...)): Số điện thoại.
    - HinhAnh (VARCHAR(255)): Hình ảnh (có thể là đường dẫn hoặc dữ liệu hình ảnh).
  + Bảng này chứa thông tin của những người hỗ trợ (có thể là giáo viên hoặc nhân viên quản lý) liên quan đến các khóa học.
* Bảng courses:
  + Lưu trữ thông tin về các khóa học.
  + Cột:
    - MaKhoaHoc (INT): Khóa chính, mã định danh khóa học.
    - TenKhoaHoc (VARCHAR(10...)): Tên khóa học.
    - MoTa (TEXT): Mô tả khóa học.
    - NgayBatDau (DATE): Ngày bắt đầu.
    - NgayKetThuc (DATE): Ngày kết thúc.
    - image (VARCHAR(255)): Hình ảnh (có thể là ảnh minh họa khóa học).
  + Bảng này quản lý danh mục các khóa học trong hệ thống, bao gồm thời gian và mô tả.
* Bảng results:
  + Lưu trữ kết quả học tập của sinh viên.
  + Cột:
    - MaKetQua (INT): Khóa chính, mã định danh kết quả.
    - MaSinhVien (VARCHAR(2...)): Khóa ngoại, liên kết với MaSinhVien trong bảng sinhvien.
    - MaKhoaHoc (INT): Khóa ngoại, liên kết với MaKhoaHoc trong bảng khoahoc.
    - Score (FLOAT): Điểm cuối kỳ.
    - NgayCapNhat (DATE): Ngày cập nhật kết quả.
  + Bảng này ghi nhận điểm số và thông tin cập nhật kết quả học tập của sinh viên theo từng khóa học.
* Bảng students:
  + Lưu trữ thông tin cá nhân của sinh viên.
  + Cột:
    - MaSinhVien (VARCHAR(20)): Khóa chính, mã định danh sinh viên.
    - HoTen (VARCHAR(50)): Họ và tên.
    - LopSinhHoat (VARCHAR(5...)): Lớp sinh hoạt.
    - Email (VARCHAR(50)): Địa chỉ email.
    - SoDienThoai (VARCHAR(11...)): Số điện thoại.
    - NgayThamGia (DATE): Ngày tham gia.
  + Bảng này quản lý thông tin cơ bản của sinh viên, bao gồm ngày tham gia hệ thống hoặc khóa học.
* Bảng support\_ course:
  + Lưu trữ mối quan hệ giữa người hỗ trợ và khóa học.
  + Cột:
    - MaSupport (VARCHAR(20)): Khóa ngoại, liên kết với MaSupport trong bảng support.
    - MaKhoaHoc (INT): Khóa ngoại, liên kết với MaKhoaHoc trong bảng khoahoc.
  + Bảng trung gian này thể hiện mối quan hệ nhiều-nhiều giữa người hỗ trợ và các khóa học họ phụ trách.
* Bảng student\_ course:
  + Lưu trữ mối quan hệ giữa sinh viên và khóa học.
  + Cột:
    - MaSinhVien (VARCHAR(20)): Khóa ngoại, liên kết với MaSinhVien trong bảng sinhvien.
    - MaKhoaHoc (INT): Khóa ngoại, liên kết với MaKhoaHoc trong bảng khoahoc.
  + Bảng trung gian này thể hiện mối quan hệ nhiều-nhiều giữa sinh viên và các khóa học họ tham gia.
* Bảng user\_login:
  + Lưu trữ thông tin đăng nhập của người dùng (có thể là admin, sinh viên, hoặc hỗ trợ).
  + Cột:
    - id (INT): Khóa chính, mã định danh người dùng.
    - studentid (VARCHAR(20)): Mã sinh viên hoặc định danh người dùng.
    - username (VARCHAR(50)): Tên đăng nhập.
    - email (VARCHAR(100)): Địa chỉ email.
    - SoDienThoai (VARCHAR(11...)): Số điện thoại.
    - NgaySinh (DATE): Ngày sinh.
    - pass (VARCHAR(100)): Mật khẩu.
    - Role (VARCHAR(100)): Vai trò (quyền hạn) của người dùng.
  + Bảng này quản lý thông tin đăng nhập và vai trò của người dùng trong hệ thống.

**2. Mối quan hệ giữa các bảng**

* **Mối quan hệ 1-n**:
  + Bảng support liên kết với support\_khoahoc qua MaSupport, nghĩa là một người hỗ trợ có thể quản lý nhiều khóa học.
  + Bảng khoahoc liên kết với support\_khoahoc qua MaKhoaHoc, một khóa học có thể có nhiều người hỗ trợ.
  + Bảng sinhvien liên kết với student\_khoahoc qua MaSinhVien, một sinh viên có thể tham gia nhiều khóa học.
  + Bảng khoahoc liên kết với student\_khoahoc qua MaKhoaHoc, một khóa học có thể có nhiều sinh viên.
  + Bảng sinhvien liên kết với ketqua qua MaSinhVien, một sinh viên có thể có nhiều kết quả học tập.
  + Bảng khoahoc liên kết với ketqua qua MaKhoaHoc, một khóa học có thể có nhiều kết quả học tập.
* **Mối quan hệ n-n**:
  + Mối quan hệ nhiều-nhiều giữa support và khoahoc được thể hiện qua bảng trung gian support\_khoahoc.
  + Mối quan hệ nhiều-nhiều giữa sinhvien và khoahoc được thể hiện qua bảng trung gian student\_khoahoc.

**2.4.2.2 Các Collection trong NoSql**

**Mối quan hệ giữa các collection**

Cơ sở dữ liệu NoSQL của MongoDB không sử dụng các khóa ngoại như cơ sở dữ liệu quan hệ, nhưng các collection trong clb-tinhoc được liên kết thông qua các trường tham chiếu (reference fields) và dữ liệu liên quan. Dưới đây là phân tích mối quan hệ:

* Courses và students
* Mối quan hệ: Một khóa học (courses) có thể có nhiều sinh viên tham gia, được thể hiện qua trường students trong courses. Ví dụ:
  + Trong courses, một tài liệu có students: ["SV001", "SV002", "SV003"], nghĩa là các sinh viên có mã SV001, SV002, SV003 (từ collection students) đang tham gia khóa học này.
  + Ngược lại, thông tin sinh viên trong students (như HoTen, LopSinhHoat) không trực tiếp tham chiếu đến courses, nhưng có thể được truy vấn thông qua mã sinh viên (SV001).
* Mối quan hệ này hỗ trợ chức năng Đăng ký khóa học cho phép hệ thống quản lý danh sách sinh viên tham gia từng khóa học.
* Courses và results
* Mối quan hệ: Collection results liên kết sinh viên và khóa học thông qua các trường student và course. Ví dụ:
  + Một tài liệu trong results có student: "SV001" và course: "640600000000000000000001", trong đó:
    - SV001 là mã sinh viên từ students.
    - 640600000000000000000001 là \_id của một khóa học trong courses.
* Mối quan hệ này hỗ trợ chức năng Xem thông tin khóa học, kết quả cho phép sinh viên hoặc quản trị viên xem điểm số và tiến độ học tập của từng sinh viên trong một khóa học cụ thể.
* Courses và supports
* Mối quan hệ: Một khóa học trong courses có thể liên quan đến nhiều yêu cầu hỗ trợ, được thể hiện qua trường supports. Ví dụ:
  + Trong courses, một tài liệu có supports: ["SUP002", "SUP003"], nghĩa là các yêu cầu hỗ trợ có mã SUP002, SUP003 (từ collection supports) liên quan đến khóa học này.
  + Ngược lại, supports không trực tiếp tham chiếu lại courses, nhưng có thể truy vấn thông qua mã hỗ trợ (SUP002).
* Ý nghĩa: Mối quan hệ này hỗ trợ chức năng Support cho phép hệ thống quản lý các yêu cầu hỗ trợ liên quan đến từng khóa học.

**Phân tích dữ liệu Collection**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.4: Quản trị cơ sở dữ liệu

Cơ sở dữ liệu của hệ thống quản lý học tập được quản lý thông qua MongoDB Compass, một công cụ trực quan hóa cho MongoDB, cho phép phân tích chi tiết cấu trúc, dữ liệu và hiệu suất. Database chính clb-tinhoc bao gồm các collection quan trọng như courses, results, students, supports, và users, mỗi collection đóng vai trò cụ thể trong việc hỗ trợ các chức năng của hệ thống. Dưới đây là phân tích chi tiết về cơ sở dữ liệu.

* **Collection courses:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.5: Cơ sở dữ liệu courses

* **Cấu trúc dữ liệu**: Mỗi tài liệu bao gồm các trường như:
* \_id: Định danh duy nhất (ví dụ: ObjectId("164600000000000000000001")).
* TenKhoaHoc: Tên khóa học (ví dụ: "Khóa học lập trình Java với bản đề nâng cao").
* MoTa: Mô tả khóa học (ví dụ: "Khóa học lập trình Java cd bản đề nâng cao").
* NgayBatDau: Ngày bắt đầu (ví dụ: "2024-01-01T00:00:00.002Z").
* NgayKetThuc: Ngày kết thúc (ví dụ: "2024-06-01T00:00:00.002Z").
* students: Danh sách mã sinh viên tham gia (ví dụ: ["SV001", "SV002", "SV003"]).
* supports: Danh sách mã hỗ trợ liên quan (ví dụ: ["SUP002", "SUP003"]).
* createdAt và updatedAt: Thời gian tạo và cập nhật (ví dụ: "2025-05-23T15:25:00.627Z").

Collection này đóng vai trò trung tâm trong việc quản lý các khóa học, lưu trữ thông tin cần thiết để hỗ trợ chức năng đăng ký khóa học và xem thông tin khóa học.

* **Collection results:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.6: Cơ sở dữ liệu results

* **Cấu trúc dữ liệu**:
* \_id: Định danh duy nhất (ví dụ: ObjectId("6837323843d7348f4d6cfca7")).
* student: Mã sinh viên (ví dụ: "SV001").
* course: Mã khóa học (ví dụ: "640600000000000000000001").
* score: Điểm số (ví dụ: 8.5).
* createdAt và updatedAt: Thời gian tạo và cập nhật (ví dụ: "2025-05-23T16:21:08Z").

Collection này hỗ trợ chức năng xem kết quả học tập, cho phép sinh viên hoặc quản trị viên theo dõi tiến độ học tập. Dữ liệu thời gian giúp phân tích hiệu suất theo thời gian.

* **Collection students:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.7: Cơ sở dữ liệu students

* **Cấu trúc dữ liệu**:
* \_id: Định danh duy nhất (ví dụ: ObjectId("6837323843d7348f4d6cfca7")).
* HoTen: Họ tên sinh viên (ví dụ: "Nguyen Van A").
* LopSinhHoat: Lớp sinh hoạt (ví dụ: "23CNNTT1").
* Email: Địa chỉ email (ví dụ: "nguyenvana@example.com").
* SoDienThoai: Số điện thoại (ví dụ: "0912345678").
* NgayThamGia: Ngày tham gia (ví dụ: "2023-09-01T00:00:00.002Z").

Collection này là nền tảng cho chức năng thêm sinh viên và quản lý danh sách sinh viên, hỗ trợ các tác vụ như đăng ký khóa học và theo dõi kết quả.

* **Collection supports:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.8: Cơ sở dữ liệu supports

* **Cấu trúc dữ liệu**:
* \_id: Định danh duy nhất (ví dụ: ObjectId("6837368a5b9c2e9a8e18857")).
* HoTen: Họ tên người yêu cầu (ví dụ: "Do Thi F").
* Email: Địa chỉ email (ví dụ: "doth@example.com").
* LopSinhHoat: Lớp sinh hoạt (ví dụ: "21CNNT1").
* SoDienThoai: Số điện thoại (ví dụ: "0935254789").
* hinhAnh: Hình ảnh minh họa (ví dụ: "image3.jpg").

Collection này phục vụ chức năng hỗ trợ học tập, cho phép sinh viên gửi yêu cầu và nhận hỗ trợ từ hệ thống, với hình ảnh minh họa để làm rõ vấn đề.

* **Collection users:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 2.9: Cơ sở dữ liệu users

* **Cấu trúc dữ liệu**:
* \_id: Định danh duy nhất (ví dụ: ObjectId("6837334243d7348f4d6cfcb1")).
* name: Tên người dùng (ví dụ: "sang").
* email: Địa chỉ email (ví dụ: "sang0@ami.com").
* phone: Số điện thoại (ví dụ: "0935254789").
* dateOfBirth: Ngày sinh (ví dụ: "5281-05-23T15:52:17.799Z").
* password: Mật khẩu mã hóa (ví dụ: "$2b$10$sbvJ140YbWj5vBWGvC828.fecA9xB8f3ONBXkG/O").
* role: Vai trò (ví dụ: "admin").

Collection này quản lý thông tin người dùng, bao gồm quản trị viên và người dùng thông thường, hỗ trợ chức năng đăng ký, đăng nhập và phân quyền.

# **CHƯƠNG 3: THỰC HIỆN XÂY DỰNG WEBSITE, TRIỂN KHAI VÀ ĐÁNH GIÁ KẾT QUẢ.**

**3.1 Giao diện và chức năng đăng nhập, đăng ký**

Chức năng đăng ký tài khoản:

Chức các trường nhập thông tin bao gồm, mã sinh viên, tên sinh viên, email, số điện thoại, ngày sinh, password. Tự động đăng nhập người dùng sau khi họ hoàn tất quá trình đăng ký.

**A screenshot of a login page

AI-generated content may be incorrect.**

Hình 3.1: Giao diện đăng ký

Chức năng đăng nhập

Các trường nhập bao gồm mã sinh viên, password. Kiểm tra tính hợp lệ của thông tin đăng nhập (Student or admin và mật khẩu). Hiển thị thông tin người dùng sau khi đăng nhập, bao gồm tên và các tính năng cá nhân khác.

Giao diện đăng nhập

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.2: Giao diện đăng nhập

## **3.2 Giao Diện Trang Chủ**

Trang chủ của CLB Tin học có các header với logo phần giới thiệu ngắn về CLB, thông tin sự kiện và hoạt động nổi bật, bài viết chia sẻ kiến thức, giới thiệu thành viên tiêu biểu, form đăng ký tham gia và footer với thông tin liên hệ, mạng xã hội. Giao diện thiết kế hiện đại, thân thiện và nhấn mạnh đam mê công nghệ.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.3: Giao diện Home

## **3.3 Giao Diện Khóa học**

Giao diện "Danh sách khóa học" được thiết kế dành cho admin với mục tiêu quản lý toàn diện các khóa. Phần chính hiển thị danh sách các khóa học ( Java, Python, C++, JavaScript,... ) trong các khung riêng biệt, mỗi khung chứa logo đặc trưng, mô tả ngắn (ví dụ: "Khóa học lập trình Java cơ bản dành cho người mới"), ngày bắt đầu, và ngày kết thúc (ví dụ: Java bắt đầu 1/1/2024 và kết thúc 1/6/2024). Admin có thể sử dụng ô "Tìm kiếm khóa học..." để tìm kiếm nhanh, và nút "Thêm khóa học" để mở bảng thêm khóa học mới, liên kết với bảng khoahoc trong cơ sở dữ liệu. Mỗi khung khóa học có các nút "Chi tiết", "Sửa", và "Xóa", cho phép admin xem thông tin chi tiết, chỉnh sửa thông tin, hoặc xóa khóa học.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.4: Giao diện khóa học admin

Đối với user, giao diện "Danh sách khóa học" sẽ có sự khác biệt về chức năng và quyền truy cập. User chỉ xem danh sách khóa học mà không có quyền thêm, chỉnh sửa hoặc xóa. Các khung khóa học (Java, Python, C++, JavaScript,...) vẫn hiển thị với logo, mô tả ngắn, ngày bắt đầu, và ngày kết thúc, nhưng các nút "Thêm khóa học" , "Sửa" và "Xóa" có thể bị ẩn hoặc vô hiệu hóa để hạn chế quyền truy cập. User có thể sử dụng ô "Tìm kiếm khóa học..." để tìm kiếm khóa học phù hợp. Nút "Chi tiết" vẫn hoạt động, cho phép user xem thông tin chi tiết về khóa học chẳng hạn như lịch học hoặc yêu cầu tham gia.

A screenshot of a computer program

AI-generated content may be incorrect.

Hình 3.5: Giao diện khóa học user

**3.4 Thêm, sửa và xóa khóa học**

**Thêm khóa học:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.6: Giao diện thêm khóa học mới

Giao diện "Thêm khóa học mới" được thiết kế để hỗ trợ admin bổ sung thông tin khóa học vào hệ thống một cách nhanh chóng và hiệu quả. Giao diện này hiển thị các trường thông tin cần thiết bao gồm "Tên khóa học", "Mô tả", "Ngày bắt đầu" (định dạng dd/mm/yyyy), và "Ngày kết thúc" (định dạng dd/mm/yyyy), cùng với một trường "Hình ảnh" cho phép admin chọn tệp hình ảnh từ thiết bị.

Khi admin nhấn nút "Lưu", hệ thống sẽ kiểm tra và lưu dữ liệu vào bảng khoahoc trong cơ sở dữ liệu, bao gồm các cột như TenKhoaHoc, MoTa, NgayBatDau, NgayKetThuc, và image. Chức năng này đảm bảo admin có thể dễ dàng thêm khóa học mới mà không cần rời khỏi giao diện chính, đồng thời hỗ trợ quản lý thông tin khóa học một cách hiệu quả và trực quan.

**Sửa khóa học:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.7: Giao diện cập nhật khóa học

Giao diện "Sửa khóa học" thiết kế dành cho admin để chỉnh sửa thông tin khóa học, trong trường hợp này là khóa học "Java" với mô tả "Khóa học lập trình Java cơ bản dành cho người mới". Hiển thị các trường thông tin như "Tên khóa học" (đã điền là "Java"), "Mô tả" (đã điền nội dung trên), "Ngày bắt đầu" (01/01/2024), "Ngày kết thúc" (01/06/2024), và "Hình ảnh" (chưa có tệp được chọn, với nút "Chọn tệp" để upload). Các trường được bố trí rõ ràng với placeholder định dạng ngày (dd/mm/yyyy), giúp admin cập nhật thông tin chính xác. Hai nút chức năng "Hủy" để đóng mà không lưu thay đổi và "Cập nhật" để lưu các chỉnh sửa vào bảng khoahoc trong cơ sở dữ liệu, liên kết với các cột như TenKhoaHoc, MoTa, NgayBatDau, NgayKetThuc, và image, đảm bảo admin có thể quản lý và cập nhật thông tin khóa học một cách hiệu quả.

**Xóa khóa học:**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.8: Giao diện xóa khóa học

Chức năng xóa khóa học trong giao diện "Danh sách khóa học" của hệ thống được thiết kế để admin loại bỏ khóa học không còn cần thiết, với cơ chế bảo vệ dữ liệu để tránh xóa nhầm. Mỗi khung khóa học (Java, Python, C++, JavaScript) hiển thị thông tin cơ bản như logo, mô tả (ví dụ: "Khóa học lập trình Java cơ bản dành cho người mới"), ngày bắt đầu, và ngày kết thúc (ví dụ: Java từ 1/1/2024 đến 1/6/2024), cùng các nút "Chi tiết", "Sửa", và "Xóa". Khi admin nhấn nút "Xóa" trên khung khóa học (ví dụ: Java), hệ thống hiển thị thông báo xác nhận "Bạn có chắc chắn muốn xóa khóa học không?" với hai tùy chọn "OK" (xác nhận xóa) và "Hủy" (hủy thao tác), đảm bảo admin không xóa nhầm dữ liệu liên kết với bảng khoahoc trong cơ sở dữ liệu, nơi lưu trữ các cột như MaKhoaHoc, TenKhoaHoc, NgayBatDau, và NgayKetThuc.

**3.5 Thêm điểm Sinh Viên**

thêm điểm sinh viên là cập nhật kết quả học tập của sinh viên cho học kỳ vào hệ thống quản lý điểm. Góp phần tạo điều kiện thuận lợi cho việc tra cứu thông tin và đánh giá năng lực học tập của sinh viên

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.9: Giao diện thêm điểm

**3.6 Giao diện của danh sách sinh viên hộ trợ của câu lạc bộ**

Giao diện danh sách support được thiết kế dành riêng cho trưởng câu lạc bộ, với mục tiêu hiển thị chi tiết và quản lý hiệu quả thông tin support trong trong câu lạc bộ. Giao diện tập trung vào sự trực quan và dễ thao tác, cho phép quản lý theo dõi danh sách sinh viên một cách khoa học và quản lý dữ liệu thông qua các nút chức năng như thêm, sửa, xóa. Tiêu đề được thiết kế nổi bật, kết hợp với bảng danh sách hiển thị rõ ràng, giúp dễ dàng quản lý câu lạc bộ truy cập và xử lý thông tin chỉ trong vài thao tác đơn giản.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.10: Giao diện support

## **3.7 Giao diện thêm người hỗ trợ**

Đây là giao diện thêm người giảng viên(hỗ trợ) nhằm đáp ứng đủ các khóa học và cũng như mở rộng thêm qui mô về giảng dạy của CLB\_tin học

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.11: Giao diện thêm support

**3.8 Giao diện xóa người Hỗ Trợ**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.12: Giao diện xóa support

**3.9 Giao diện kết quả điểm của Sinh viên**

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.13: Giao diện kết quả

**3.10 Giao Diện danh sách sinh viên**

Giao diện danh sách sinh viên được thiết kế dành riêng cho trưởng câu lạc bộ, với mục tiêu hiển thị chi tiết và quản lý hiệu quả thông tin sinh viên trong trong câu lạc bộ. Giao diện tập trung vào sự trực quan và dễ thao tác, cho phép quản lý theo dõi danh sách sinh viên một cách khoa học và quản lý dữ liệu thông qua các nút chức năng như thêm, sửa, xóa. Tiêu đề được thiết kế nổi bật, kết hợp với bảng danh sách hiển thị rõ ràng, giúp dễ quản lý câu lạc bộ truy cập và xử lý thông tin chỉ trong vài thao tác đơn giản.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.14: Giao diện sinh viên

## **3.11 Giao Diện Thêm sửa và xóa sinh viên**

**Thêm sinh viên:**

Đây là giao diện thêm các sinh viên tham gia câu lạc bộ để quản lý và đáp ứng các nhu cầu của sinh viên trong tương lại như đăng kí tham gia các khóa học thêm từ câu lạc bộ hoặc theo dõi các thông báo thông tin từ khoa,vv

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.15: Giao diện thêm sinh viên

**Sửa sinh viên:**

Giao diện này là một phần của một hệ thống quản lý thông tin sinh viên, được thiết kế để hỗ trợ việc nhập, chỉnh sửa và lưu trữ dữ liệu cá nhân một cách có tổ chức. Hệ thống bao gồm các trường thông tin cơ bản như mã sinh viên, họ và tên, email, số điện thoại, và lớp học, được sắp xếp theo trình tự logic để tối ưu hóa trải nghiệm người dùng. Chức năng "Cập nhật" cho phép lưu thông tin sau khi chỉnh sửa, trong khi nút "Hủy" hỗ trợ hủy bỏ thao tác khi cần thiết.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.16: Giao diện sửa sinh viên

**Xóa Sinh Viên**

Hệ thống quản lý danh sách học viên được thiết kế để hỗ trợ admin quản lý thông tin sinh viên một cách hiệu quả. Giao diện hiển thị danh sách học viên trong các khung riêng biệt, bao gồm thông tin chi tiết như tên, mã sinh viên, email, số điện thoại, lớp học và ngày tham gia. Admin có thể thực hiện các thao tác như thêm, sửa, xóa học viên, tìm kiếm và lọc danh sách theo lớp học, với mục tiêu đảm bảo quản lý thông tin rõ ràng và thuận tiện. Khi admin nhấn nút "Xóa" màu đỏ nằm dưới khung thông tin của học viên, hệ thống sẽ kiểm tra trạng thái của học viên đó. Nếu học viên không thuộc diện bảo vệ, nghĩa là không trúng tuyển các khóa học cụ thể như Java, Python, C++, hoặc thiết, hệ thống sẽ tiến hành xóa ngay lập tức, giúp admin nhanh chóng cập nhật danh sách mà không cần bước xác nhận bổ sung.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.17: Giao diện xóa sinh viên

Trong trường hợp học viên đã trúng tuyển các khóa học được bảo vệ như Java, Python, C++, hoặc thiết, việc nhấn nút "Xóa" sẽ kích hoạt một thông báo cảnh báo từ hệ thống. Thông báo này hiển thị dòng chữ "Sinh viên đã trúng tuyển các khóa học sau không thể xóa: Java, Python, C++, thiết", kèm theo hai tùy chọn "OK" để xác nhận và "Hủy" để hủy hành động. Cơ chế này được thiết kế để bảo vệ dữ liệu quan trọng, ngăn chặn việc xóa nhầm các học viên đã tham gia khóa học, đồng thời cung cấp thông tin rõ ràng để admin hiểu lý do không thể thực hiện thao tác xóa. Hệ thống đảm bảo tính minh bạch và an toàn dữ liệu, giúp admin quản lý danh sách học viên một cách hiệu quả và đáng tin cậy.

A screenshot of a computer

AI-generated content may be incorrect.

Hình 3.18: Giao diện thêm sinh viên trong khóa học

**3.12 Giao diện chi tiết khóa học**

Chức năng này cho phép hiển thị danh sách học viên đã đăng ký, với nút "Thêm học viên" hỗ trợ bổ sung sinh viên mới một cách dễ dàng. Đồng thời, khu vực "Danh sách học viên" cung cấp khả năng theo dõi thông tin chi tiết của từng học viên tham gia khóa học. Nút "Kết quả" tích hợp trong giao diện giúp người quản lý xem điểm số hoặc tiến độ học tập của học viên, đảm bảo việc đánh giá hiệu suất được thực hiện thuận tiện. Đồng thời tùy chọn xóa (biểu tượng "X") giúp loại bỏ thành viên khỏi danh sách khi cần thiết.

Trong một giao diện khác, tôi nhận thấy chức năng "Danh sách support" hỗ trợ quản lý đội ngũ hỗ trợ của khóa học. Chức năng này hiển thị danh sách các thành viên hỗ trợ, ví dụ như "Do Thi F" và "Bùi Văn G", với mã định danh tương ứng (SUP002, SUP003). Nút "Thêm support" cho phép bổ sung nhân sự hỗ trợ mới, đồng thời tùy chọn xóa (biểu tượng "X") giúp loại bỏ thành viên khỏi danh sách khi cần thiết.

A screenshot of a phone

AI-generated content may be incorrect.A white rectangular object with a black border

AI-generated content may be incorrect.

Hình 3.19: Giao diện thêm sinh viên

**3.13 Thêm, xóa học viên và support trong khóa học**

**Thêm sinh viên vào khóa học:**

Giao diện chức năng "Thêm học viên vào khóa học" và nhận thấy đây là một công cụ hỗ trợ quản lý danh sách học viên hiệu quả. Giao diện cung cấp ô tìm kiếm "Tìm kiếm học viên..." để người dùng có thể nhanh chóng lọc danh sách dựa trên tên hoặc mã sinh viên. Danh sách hiển thị bao gồm các học viên như Tran Van F (SV006), Le Thi G (SV007), Pham Van H (SV008), Bui Van J (SV010), và Nguyen Thi K (SV011),... với mã sinh viên tương ứng được liệt kê rõ ràng bên cạnh tên. Thiết kế này giúp dễ dàng theo dõi và chọn học viên để thêm vào khóa học. Nút "X" ở góc trên cùng bên phải cho phép hủy thao tác, đảm bảo tính linh hoạt trong quá trình sử dụng

A screenshot of a chat

AI-generated content may be incorrect.

Hình 3.20: Giao diện thêm sinh viên

**Xóa sinh viên khỏi khóa học:**

Chức năng này được tích hợp dưới dạng nút "X" bên cạnh mỗi học viên, như Nguyen Van Aa (SV001), Tran Thi B (SV002),.... cho phép người dùng loại bỏ học viên khỏi khóa học một cách dễ dàng. Khi nhấp vào nút "X", hệ thống hiển thị một thông báo xác nhận "Bạn có chắc chắn muốn xóa học viên khỏi | khóa học?" với hai tùy chọn "OK" và "Hủy", đảm bảo tránh thao tác nhầm lẫn và tăng tính an toàn. Sau khi xác nhận bằng "OK", học viên được xóa khỏi danh sách, trong khi "Hủy" cho phép hủy thao tác.

* Chức năng thêm và xóa support trong chi tiết khóa học cũng tương tự như các phần trên của sinh viên.

**KẾT LUẬN**

Ứng dụng quản lý câu lạc bộ tin học của Trường Đại Học Sư Phạm Đà Nẵng được phát triển nhằm tối ưu hóa quy trình quản lý thông tin khóa học, học viên và giáo viên. Với các chức năng chính như quản lý khóa học, lưu trữ thông tin học viên, giáo viên và kết quả học tập, ứng dụng không chỉ giúp cải thiện tính hiệu quả trong công tác quản lý mà còn nâng cao trải nghiệm của người dùng.

Việc tích hợp các tính năng cho phép theo dõi tiến độ học tập và đánh giá kết quả khóa học sẽ hỗ trợ các giáo viên trong việc điều chỉnh chương trình giảng dạy và tạo ra môi trường học tập tốt nhất cho học viên. Đồng thời, ứng dụng cũng giúp học viên dễ dàng truy cập thông tin cần thiết về khóa học của mình.

Nhìn chung, hệ thống này không chỉ đáp ứng nhu cầu hiện tại của câu lạc bộ mà còn có khả năng mở rộng trong tương lai, góp phần thúc đẩy sự phát triển bền vững của các hoạt động liên quan đến tin học tại trường.

**TÀI LIỆU THAM KHẢO**

* 1. **Wikipedia – JavaScript**: <https://vi.wikipedia.org/wiki/JavaScript>
  2. **W3Schools - JavaScript Tutorial**: <https://www.w3schools.com/js>
  3. **Mozilla Developer Network (MDN) – JavaScript**: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>

1. **Freetuts.net - Tài liệu học JavaScript từ cơ bản đến nâng cao**: <https://freetuts.net/hoc-javascript>
2. MongoDB Official Documentation: <https://www.mongodb.com/docs/>
3. W3Schools - MongoDB Tutorial: https://www.w3schools.com/mongodb/
4. GeeksforGeeks - Introduction to MongoD: <https://www.geeksforgeeks.org/mongodb>