

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN MÔN HỌC
THUẬT TOÁN VÀ ỨNG DỤNG
ĐỀ TÀI:
XÂY DỰNG CHƯƠNG TRÌNH TRA CỨU MÔN
CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

Giảng viên hướng dẫn: TS. Nguyễn Đình Hiền

Sinh viên thực hiện:

MSSV:

- Bùi Văn Tân	6051071104
- Nguyễn Công Chí	6051071009
- Võ Văn Trí	6051071126
- Hồ Ngọc Thống	6051071118
- Huỳnh Xuân Lãm	6051071062

TP. Hồ Chí Minh, năm 2021

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH**



**BÁO CÁO ĐỒ ÁN MÔN HỌC
THUẬT TOÁN VÀ ỨNG DỤNG
ĐỀ TÀI:
XÂY DỰNG CHƯƠNG TRÌNH TRA CỨU MÔN
CẤU TRÚC DỮ LIỆU VÀ GIẢI THUẬT**

Giảng viên hướng dẫn: TS. Nguyễn Đình Hiền

Sinh viên thực hiện:

MSSV:

- Bùi Văn Tân	6051071104
- Nguyễn Công Chí	6051071009
- Võ Văn Trí	6051071126
- Hồ Ngọc Thống	6051071118
- Huỳnh Xuân Lãm	6051071062

TP. Hồ Chí Minh, năm 2021

LỜI MỞ ĐẦU

Lý do chọn đề tài.

Ngày nay, nền khoa học kỹ thuật ngày càng phát triển, đặc biệt là ngành công nghệ thông tin. Ứng dụng của công nghệ đã tràn ngập mọi mặt trong đời sống. Từ vật chất như điện thoại, tivi, tủ lạnh, điều hòa, điều khiển máy ... tới các mạng xã hội như facebook, zalo, googletất cả đều là ứng dụng của công nghệ thông tin. Thế giới ngày càng hiện đại là do ngành công nghệ thông tin ngày càng phát triển.

Chính vì tầm quan trọng và đa dạng của ngành công nghệ thông tin mà đòi hỏi các sinh viên công nghệ thông tin cần phải nắm vững các nguyên lý và ứng dụng của các môn học công nghệ thông tin nói chung và môn Thuật toán & ứng dụng nói riêng. Môn học trang bị cho sinh viên những kiến thức cơ bản về các thuật toán dựa trên cấu trúc dữ liệu để thiết kế và cài đặt các phần mềm. Trong phạm vi môn học, nhằm tìm hiểu về các thuật toán, cấu trúc dữ liệu, nhóm em xin trình bày đề tài: “Xây dựng chương trình tra cứu kiến thức môn Cấu trúc dữ liệu và Giải thuật” nhằm cài đặt chương trình tạo kết nối danh sách liên kết, thực hiện các thao tác, thủ tục trên cấu trúc dữ liệu, phân dạng nội dung theo từng chương của bài học.

Hướng tiếp cận của đề tài.

- Nghiên cứu và hiểu được cách thức hoạt động của database.
- Thiết kế cơ sở dữ liệu sao cho phù hợp với ứng dụng đang phát triển.
- Nắm vững kiến thức nền tảng của ngôn ngữ lập Javascript, tiến đến phát triển ứng dụng.
- Tiến hành kiểm tra và chạy thử ứng dụng.

Mục tiêu nghiên cứu.

- Phát triển thành công ứng dụng nội dung môn học cấu trúc dữ liệu và giải thuật.
- Hiểu được cách thức hoạt động của cơ sở dữ liệu.
- Khiến ứng dụng trở nên có ích hơn cho xã hội.

Đối tượng và phạm vi nghiên cứu.

- Đối tượng: Các học sinh, sinh viên, giảng viên và phụ huynh.
- Phạm vi ứng dụng: Mong muốn ứng dụng hoàn thiện và được tất cả mọi người quan tâm biết đến để tra cứu thông tin cần tìm hiểu.

This image shows a full page of a document template designed for writing. It features a series of evenly spaced, horizontal grey lines that run across the entire width of the page. The lines are thin and light, providing a guide for text alignment without being distracting. There are no margins, headers, footers, or other markings present on the page.

Giảng viên hướng dẫn
Nguyễn Đình Hiền

MỤC LỤC

LỜI MỞ ĐẦU	
NHẬN XÉT CỦA GIÁO VIÊN	
CHƯƠNG 1: CƠ SỞ LÝ THUYẾT.....	1
1.1 Giới thiệu:.....	1
1.1.1 Khái niệm ReactJS:	1
1.1.2 Khái niệm JavaScript:	3
1.1.3 Khái niệm Node.JS:	4
1.1.4 Khái niệm Express.js:.....	6
1.1.5 Khái niệm MongoDB:.....	8
CHƯƠNG 2: PHÂN TÍCH HỆ THỐNG	11
2.1 Sơ đồ Use Case:.....	11
2.2 Sơ đồ phân cấp chức năng của quản lý:	12
2.3 Sơ đồ phân cấp chức năng của người dùng:.....	12
CHƯƠNG 3: THIẾT KẾ HỆ THỐNG	13
3.1 Giao diện người quản lý:	13
3.1.1 Giao diện chính của người quản lý:	13
3.1.2 Giao diện chức năng tạo bài viết:.....	13
3.1.3 Giao diện chức năng quản lý danh mục:.....	14
3.1.4 Giao diện chức năng quản lý chủ đề:	14
3.1.5 Giao diện chức năng tìm kiếm:	15
3.2 Giao diện của người dùng:	15
3.2.1 Giao diện chính của người dùng:	15
3.2.2 Giao diện chức năng xem danh mục bài học:	16
3.2.3 Giao diện liệt kê bài viết được tra cứu nhiều nhất:	16
3.2.4 Giao diện tra cứu chính:	17
3.2.5 Giao diện tra cứu bằng chatbot:	17
KẾT LUẬN VÀ KHUYẾN NGHỊ.....	18
TÀI LIỆU THAM KHẢO	19
DANH SÁCH PHÂN CÔNG KHỐI LƯỢNG THỰC HIỆN.....	19

DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ Use Case.....	11
Hình 2: Sơ đồ phân cấp chức năng của người quản lý	12
Hình 3: Sơ đồ phân cấp chức năng của người dùng	12
Hình 4: Giao diện chính của người quản lý.....	13
Hình 5: Giao diện chức năng tạo bài viết	14
Hình 6: Giao diện chức năng quản lý danh mục	14
Hình 7: Giao diện chức năng quản lý chủ đề	14
Hình 8: Giao diện chức năng tìm kiếm bài viết	15
Hình 9: Giao diện chính của người dùng	15
Hình 10: Giao diện chức năng xem danh mục bài học	16
Hình 11: Giao diện liệt kê bài viết được tra cứu nhiều nhất	16
Hình 12: Giao diện tra cứu thông tin bài học	17
Hình 13: Giao diện tra cứu bằng chatbot.....	17

DANH MỤC HÌNH ẢNH

Bảng 1: So sánh giữa Props và State	2
Bảng 2: Phân biệt Expressjs và Nodejs	7
Bảng 3: Công việc của mỗi thành viên và mức độ hoàn thành	19

CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

1.1 Giới thiệu:

1.1.1 Khái niệm ReactJS:

React là thư viện JavaScript phổ biến nhất để xây dựng giao diện người dùng (UI). Cho tốc độ phản hồi tuyệt vời khi user nhập liệu bằng cách sử dụng phương pháp mới để render trang web. Components của công cụ này được phát triển bởi Facebook. Và được ra mắt như một công cụ JavaScript mã nguồn mở vào năm 2013. Hiện tại, nó đã đi trước các đối thủ chính như Angular và Bootstrap, hai thư viện JavaScript bán chạy nhất hiện tại.

Một trong những điểm hấp dẫn của React là thư viện này không chỉ hoạt động trên phía client, mà còn được render trên server và có thể kết nối với nhau. React so sánh sự thay đổi giữa các giá trị của lần render này với lần render trước và cập nhật ít thay đổi nhất trên DOM.

▪ Giới thiệu về Virtual DOM:

+ Công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.

▪ Giới thiệu về JSX:

+ Là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript. Đặc điểm: Faster: Nhanh hơn. JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript. Safer: an toàn hơn.

+ Ngược với Javascript, JSX là kiểu statically-typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt. Easier: Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

▪ Giới thiệu về Components:

+ React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác.

+ Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component. Chúng giúp bảo trì mã code khi làm việc với các dự án lớn. Một react component đơn giản chỉ cần một method render. Có rất nhiều methods khả dụng khác, nhưng render là method chủ đạo.

▪ **Giới thiệu Props và State:**

- ⇒ **Props:** giúp các component tương tác với nhau, component nhận input gọi là props, và trả thuộc tính mô tả những gì component con sẽ render. Prop là bất biến.
- ⇒ **State:** thể hiện trạng thái của ứng dụng, khi state thay đổi thì component đồng thời render lại để cập nhật UI.

So Sánh	Props	State
Có thể nhận giá trị ban đầu từ component cha	✓	✓
Có thể thay đổi bởi component cha	✓	X
Có thể set giá trị mặc định bên trong component	✓	✓
Có thể thay đổi bên trong component	X	✓
Có thể set giá trị ban đầu cho các component con	✓	✓
Có thể thay đổi trong các component con	✓	X

Bảng 1: So sánh giữa Props và State

❖ **Ưu điểm của việc sử dụng JavaScript:**

- **Reactjs cực kì hiệu quả:** Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
- **Reactjs giúp việc viết các đoạn code JS dễ dàng hơn:** Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
- **Nó có nhiều công cụ phát triển:** Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.

- **Render tầng server:** Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm.
- **Làm việc với vấn đề test giao diện:** Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
- **Hiệu năng** cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

❖ **Hạn chế của JavaScript**

- Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax
- Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
- React khá nặng nếu so với các framework khác React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh
- Khó tiếp cận cho người mới học Web.

1.1.2 Khái niệm JavaScript:

+ JavaScript là ngôn ngữ thông dụng của 1 chương trình của nguồn được nhúng hoặc tích hợp vào 1 tập tin như HTML. khi mà trang web được tải trong trình duyệt hỗ trợ JavaScript. JavaScript là ngôn ngữ lập trình phổ biến bậc nhất thế giới, nó là ngôn ngữ cho các thẻ HTML, web, server, PC, laptop, tablet.

+ Hiện nay, JS không chỉ có thể viết client side mà còn mở rộng ra nhiều lĩnh vực khác, có thể kể tới như viết app với Electron hoặc web service với Node.js. Các thư viện, framework cho JS phát triển rất mạnh, như jquery, React, Angular, Vue,... Lưu ý là JS không phải Java, là ngôn ngữ có phân biệt hoa thường.

❖ **Javascript với HTML:**

- Đây là một trong số các ngôn ngữ lập trình phổ biến nhất hiện nay, nó được dùng để xây dựng phần khung chính của một website.
- Bên cạnh đó, khác với Javascript, HTML không thực sự được coi là một ngôn ngữ lập trình, nó không thể dùng để tạo ra các chức năng động mà chỉ được dùng để bố cục và định dạng website.

❖ **Javascript với CSS:**

- CSS là viết tắt của Cascading Style Sheets – hiểu một cách đơn giản thì nhiệm vụ của nó xác định nội dung sẽ xuất hiện cho người dùng thấy như thế nào. Ngôn ngữ này ảnh hưởng tới toàn bộ bố cục và giao diện của website. Trong khi đó, ngôn ngữ lập trình JavaScript đảm nhiệm phần lập trình confirmation boxes, calls-to-action, và thêm thông tin mới vào thông tin hiện có...

❖ **Javascript với PHP:**

- Và cuối cùng là PHP. Ngôn ngữ PHP được viết đầy đủ là Hypertext Preprocessor, nó đảm nhiệm vai trò xử lý các biểu mẫu, lưu file dữ liệu, trả lại data cho người dùng, thu thập data từ các file và nhiều hơn thế nữa trong khi JavaScript được thiết kế để tạo các ứng dụng tập trung vào mạng. Ngôn ngữ này được tích hợp nhiều cơ sở dữ liệu khác nhau như Postgre SQL, Oracle, Sybase, SQL và MySQL.

❖ **Ưu điểm của việc sử dụng JavaScript:**

- **Ít tương tác với máy chủ** – Bạn có thể xác thực đầu vào của người dùng trước khi gửi trang đến máy chủ. Điều này giúp tiết kiệm lưu lượng máy chủ, có nghĩa là tải ít hơn trên máy chủ của bạn.
- **Phản hồi ngay lập tức cho khách truy cập** – Họ không phải đợi tải lại trang để xem họ có quên nhập nội dung nào không.
- **Tăng tính tương tác** – Bạn có thể tạo các giao diện phản ứng khi người dùng di chuột qua chúng hoặc kích hoạt chúng thông qua bàn phím.
- **Giao diện phong phú hơn** – Bạn có thể sử dụng JavaScript để bao gồm các mục như các thành phần kéo và thả để trượt Giao diện phong phú cho khách truy cập trang web của bạn.

❖ **Hạn chế của JavaScript**

- **JavaScript:** phía máy khách không cho phép đọc hoặc ghi tệp. Điều này đã được giữ vì lý do an ninh.
- **JavaScript:** không thể được sử dụng cho các ứng dụng mạng vì không có hỗ trợ như vậy.
- **JavaScript:** không có bất kỳ khả năng đa luồng hoặc đa xử lý nào.
- Một lần nữa, Js là ngôn ngữ lập trình nhẹ, được giải thích cho phép bạn xây dựng tính tương tác vào các trang HTML tĩnh.

1.1.3 Khái niệm Node.JS:

+ Node.js là một nền tảng chạy trên môi trường V8 JavaScript runtime – một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Bình thường cũng có thể tải bộ V8 và nhúng nó vào bất cứ thứ gì, Node.js làm điều đó đối với các web server. JavaScript suy cho cùng cũng chỉ là một ngôn ngữ – vậy thì không có lý do gì để nói nó

không thể sử dụng trên môi trường server tốt như là trong trình duyệt của người dùng được.

❖ Các đặc tính của NodeJS

- **Không đồng bộ:** Tất cả các API của NodeJS đều không đồng bộ (non-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
- **Chạy rất nhanh:** NodeJS được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
- **Đơn luồng nhưng khả năng mở rộng cao:** Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo ra hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
- **Không đệm:** NodeJS không đệm bất kỳ một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
- **Có giấy phép:** NodeJS đã được cấp giấy phép bởi MIT License.

❖ Ưu điểm của việc sử dụng NodeJS:

- **JavaScript trên máy chủ:** Node.js đã mở ra cánh cửa cho sự phát triển full-stack của Javascript, kế thừa những giá trị của lập trình JavaScript, cũng như cho phép các kỹ sư sử dụng các thư viện và tính năng của nó. JavaScript nhẹ đạt được hiệu suất cao với ít dòng mã hơn khi so sánh với Java hoặc C. Ngoài ra, front-end và back-end dễ đồng bộ hơn vì sử dụng một ngôn ngữ duy nhất trên cả hai mặt của ứng dụng.
- **Tốt cho microservices:** Node.js có khả năng mở rộng cao và nhẹ. Đó là lý do tại sao nó rất được yêu thích đối với các kiến trúc dịch vụ vi mô. Kiến trúc microservice có nghĩa là chia nhỏ ứng dụng thành các dịch vụ riêng biệt và độc lập, giúp cập nhật và duy trì kiến trúc dễ dàng hơn khi các dịch vụ của bạn được tách rời.
- **Hỗ trợ và cộng đồng:** Vào năm 2015, một số công ty bao gồm IBM, Microsoft, PayPal, Fidelity đã tổ chức thành lập nền tảng Node.js. Đó là một cộng đồng độc lập nhằm tạo điều kiện phát triển các công cụ cốt lõi của Node.js. Bất kể sự tham gia tích cực của công ty, Node.js là một dự án mã nguồn mở đã tập hợp một cộng đồng lớn các nhà phát triển cung cấp các giải pháp và hướng dẫn làm sẵn cho từng vấn đề trên GitHub.

- **Gói:** một số ngôn ngữ lập trình có thể tự hào như hệ sinh thái đóng gói phong phú như node.js hàng ngàn thư viện và công cụ để phát triển Javascript được tập hợp trên NPM, một trình quản lý gói node.js mặc định và thị trường với sự hỗ trợ liên tục của cộng đồng node.js NPM tập trung vào việc khuyến khích người dùng để thêm các gói mới.
- **Dễ học:** Vì Node.js kế thừa nhiều tính năng JavaScript bao gồm các thư viện của nó, các nhà phát triển front-end có thể sẽ học nó khá nhanh.

❖ **Hạn chế của NodeJS:**

- **Nút cổ chai của CPU:** Như chúng tôi đã đề cập, Node.js sử dụng mô hình đầu vào-đầu ra không chặn để xử lý các yêu cầu. Vì vậy, nó được coi là đơn luồng. Về bản chất, Node.js không có khả năng xử lý một số tác vụ đơn giản, giống như đọc các truy vấn cơ sở dữ liệu mà không chặn luồng chính.
- **Các công cụ chưa trưởng thành:** Đăng ký NPM dường như là số lượng chứ không phải theo định hướng chất lượng. Trong khi các sản phẩm cốt lõi được chứng minh là ổn định và được kiểm tra tốt, phần còn lại của cơ quan đăng ký NPM có cấu trúc kém và được ghi chép không tốt, đó là các dự án mã nguồn mở.
- **Ít nhà phát triển có kinh nghiệm:** Điểm đáng nói cuối cùng là thiếu các nhà phát triển JavaScript có kinh nghiệm làm việc với back-end.

1.1.4 Khái niệm Express.js:

+ Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

❖ **Các đặc tính của Express.JS:**

- **Phát triển máy chủ nhanh chóng:** Express.js cung cấp nhiều tính năng thường được sử dụng của Node.js dưới dạng các hàm có thể dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này loại bỏ nhu cầu viết mã trong vài giờ và do đó tiết kiệm thời gian.
- **Phần mềm trung gian – Middleware:** Là một phần của chương trình có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và các phần mềm trung gian khác. Nó chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng khác nhau của Express.js.
- **Định tuyến – Routing:** Express.js cung cấp cơ chế định tuyến nâng cao giúp duy trì trạng thái của trang web với sự trợ giúp của URL của chúng.

- **Tạo mẫu – Templating:** Expressjs cung cấp các công cụ tạo khuôn mẫu cho phép các nhà phát triển xây dựng nội dung động trên các trang web bằng cách xây dựng các mẫu HTML ở phía máy chủ.
- **Gỡ lỗi – Debugging:** Gỡ lỗi là rất quan trọng để phát triển thành công các ứng dụng web. ExpressJS giúp việc gỡ lỗi trở nên dễ dàng hơn bằng cách cung cấp cơ chế gỡ lỗi có khả năng xác định chính xác phần ứng dụng web có lỗi.

❖ **Những tính năng của Express.JS:**

- **Phát triển máy chủ nhanh chóng:** Expressjs cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian.
- **Phần mềm trung gian Middleware:** Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.js.
- **Định tuyến - Routing:** Express js cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL.
- **Tạo mẫu - Templating:** Các công cụ tạo khuôn mẫu được Express.js cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ.
- **Gỡ lỗi - Debugging:** Để phát triển thành công các ứng dụng web không thể thiếu việc gỡ lỗi. Giờ đây với Expressjs việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi.

❖ **Phân biệt Expressjs và Nodejs:**

Bảng 2: Phân biệt Expressjs và Nodejs

Express.js	Node.js
- Là framework của Node.js, sử dụng để xây dựng phần phụ trợ của ứng dụng web.	- Được sử dụng để xây dựng cả frontend và backend của ứng dụng web.
- Được viết bằng một ngôn ngữ lập trình duy nhất là Javascript,...	- Được viết bằng nhiều ngôn ngữ lập trình khác nhau như C/C++,
- Là một framework	- Không phải là framework
- Để sử dụng Expressjs các lập trình viên cần cài đặt Expressjs cùng Node.js.	- Các lập trình viên chỉ cần cài đặt Node.js trên thiết bị của mình để có thể sử dụng
- Được sử dụng để xây dựng nên các ứng	- Được sử dụng để phát triển lên các ứng

dụng phía máy chủ trên Node.js.	dụng mạng và phía máy chủ.
- Express js phù hợp với các dự án có quy mô nhỏ.	- Nodejs được sử dụng cho các dự án có quy mô lớn.
- Express js chỉ được sử dụng ở phía máy chủ.	- Node.js có thể sử dụng được cả phía máy chủ và máy khách.
- Express.js tương thích với tất cả các hệ điều hành tương thích với Node.js.	- Node.js tương thích với tất cả các hệ điều hành chính.
- Cung cấp các thành phần định tuyến và có phần mềm trung gian để hỗ trợ giúp phát triển ứng dụng web một cách dễ dàng hơn.	- Cung cấp nhiều tính năng cho các nhà phát triển trong xây dựng một ứng dụng web.
- Express.js chỉ hỗ trợ ngôn ngữ JavaScript.	- Node.js hỗ trợ nhiều ngôn ngữ khác nhau như: TypeScript, CoffeeScript và Ruby.
- Nó được sử dụng bởi IBM, PayPal, Fox Sports,...	- Nó được sử dụng bởi LinkedIn, PayPal, Walmart, Uber,...

1.1.5 Khái niệm MongoDB:

+ MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

+ Các Collection trong MongoDB có cấu trúc cực kỳ linh hoạt. Điều này cho phép dữ liệu không cần thiết phải tuân theo bất kỳ một dạng cấu trúc nào. Vì thế, MongoDB có thể lưu trữ những dữ liệu có cấu trúc đa dạng và phức tạp. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.

❖ Các đặc tính của MongoDB:

- **Truy vấn:** Hỗ trợ truy vấn đặc biệt và truy vấn dựa trên tài liệu.
- **Hỗ trợ chỉ mục:** Bất kỳ trường nào trong tài liệu đều có thể được lập chỉ mục
- **Nhân rộng:** Nó hỗ trợ nhân rộng Master. MongoDB sử dụng ứng dụng gốc để duy trì nhiều bản sao dữ liệu. Ngăn chặn thời gian chết của cơ sở dữ liệu là một trong những tính năng của bản sao vì nó có khả năng tự phục hồi.
- **Nhiều máy chủ:** Cơ sở dữ liệu có thể chạy trên nhiều máy chủ. Dữ liệu được sao chép để bảo vệ hệ thống trong trường hợp lỗi phần cứng.
- **Tự động hủy:** Quá trình này phân phối dữ liệu trên nhiều phân vùng vật lý được gọi là phân đoạn. Do MongoDB có tính năng cân bằng tải tự động.
- **MapReduce:** Hỗ trợ MapReduce và các công cụ tổng hợp linh hoạt.

- **Xử lý lỗi:** Trong MongoDB, dễ dàng quản trị trong trường hợp xảy ra lỗi. Số lượng lớn các bản sao cung cấp khả năng bảo vệ và dữ liệu sẵn có được đưa lên trước thời gian ngừng hoạt động của cơ sở dữ liệu như lỗi giá, nhiều lỗi máy, lỗi trung tâm dữ liệu hoặc lỗi phân vùng mạng.
- **GridFS:** Bất kỳ kích thước tệp nào cũng được lưu trữ, tính năng GridFS chia các tệp thành các phần nhỏ hơn và lưu trữ chúng dưới dạng tài liệu riêng biệt.

❖ **Ưu điểm của việc sử dụng JavaScript:**

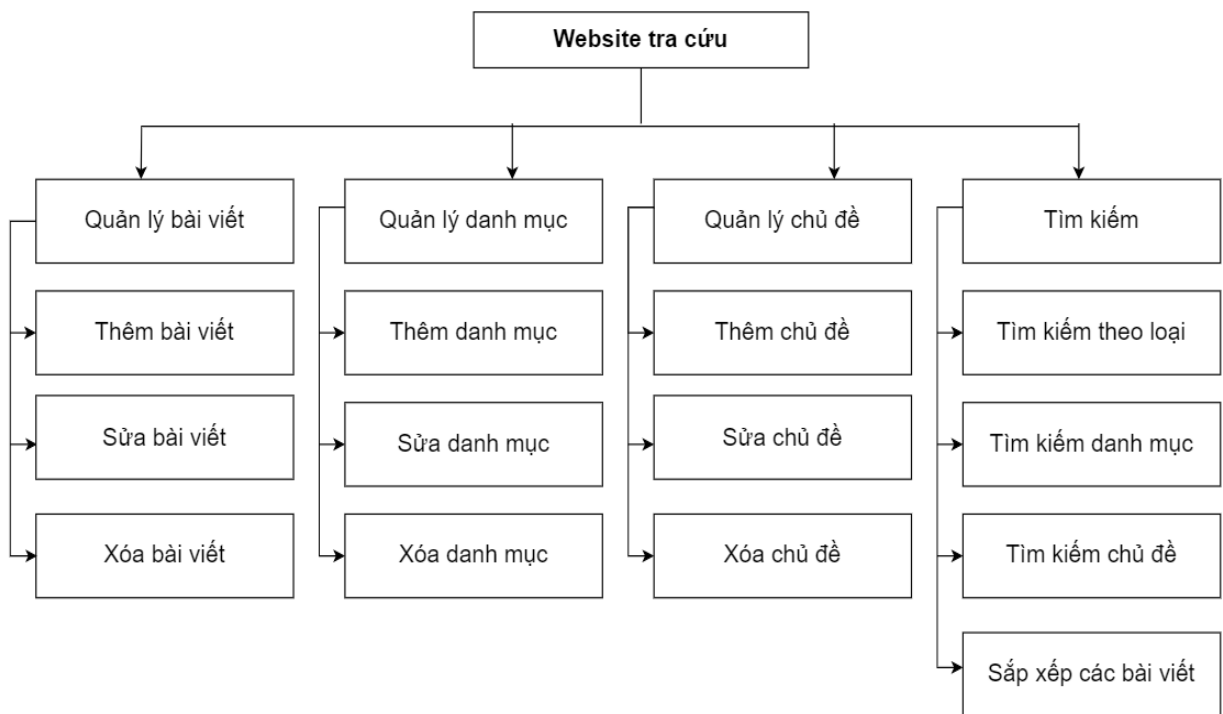
- **Open Source:** MongoDB là phần mềm mã nguồn mở miễn phí, có cộng đồng phát triển rất lớn
- **Hiệu năng cao:**
 - Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS).
 - Thử nghiệm cho thấy tốc độ insert, tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.
- **Các lý do sau:**
 - MongoDB lưu dữ liệu dạng JSON, khi bạn insert nhiều đối tượng thì nó sẽ insert một mảng JSON gần như với trường hợp insert 1 đối tượng.
 - Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau như trong RDBMS, khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các bảng liên quan như trong RDBMS.
 - Dữ liệu trong MongoDB được đánh chỉ mục (đánh index) nên khi truy vấn sẽ tìm rất nhanh.
 - Khi thực hiện insert, find... MongoDB sẽ khóa các thao tác khác lại, ví dụ khi nó thực hiện find(), trong quá trình find mà có thêm thao tác insert, update thì nó sẽ dừng hết lại để chờ find() xong đã.
- **Dữ liệu linh hoạt:** MongoDB là document database, dữ liệu lưu dưới dạng JSON, không bị bó buộc về số lượng field, kiểu dữ liệu... bạn có thể insert thoải mái dữ liệu mà mình muốn.
- **Là Rich Query Language:** MongoDB là một rich query language tức là nó có sẵn các method để thực hiện create, read, update, delete dữ liệu (CRUD).
- **Tính sẵn có:** MongoDB hỗ trợ replica set nhằm đảm bảo việc sao lưu và khôi phục dữ liệu.

- **Khả năng mở rộng Horizontal Scalability:** Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node vào cluster.

❖ **Hạn chế của JavaScript**

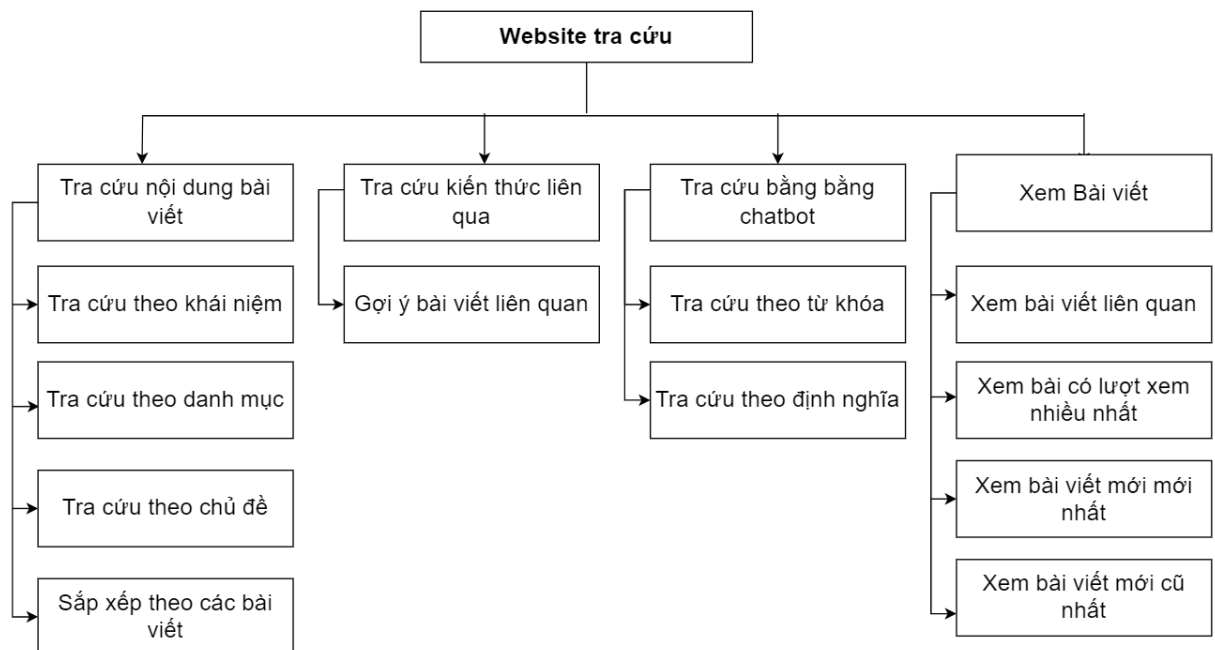
- MongoDB không có các tính chất ràng buộc như trong RDBMS → dễ bị làm sai dữ liệu.
- Không hỗ trợ join giống như RDBMS nên khi viết function join trong code ta phải làm bằng tay khiến cho tốc độ truy vấn bị giảm.
- Sử dụng nhiều bộ nhớ: do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên sẽ bị dư thừa dữ liệu
- Bị giới hạn kích thước bản ghi: mỗi document không được có kích thước > 16Mb và không mức độ các document con trong 1 document không được > 100

2.2 Sơ đồ phân cấp chức năng của quản lý:



Hình 2: Sơ đồ phân cấp chức năng của người quản lý

2.3 Sơ đồ phân cấp chức năng của người dùng:



Hình 3: Sơ đồ phân cấp chức năng của người dùng

CHƯƠNG 3: THIẾT KẾ HỆ THỐNG

3.1 Giao diện người quản lý:

3.1.1 Giao diện chính của người quản lý:

The screenshot displays the user management interface. On the left, there is a sidebar with navigation links: 'Quản lý bài viết', 'Tạo bài viết', 'Quản lý danh mục', and 'Quản lý chủ đề'. The main content area shows two article cards. The first card, titled 'Cây nhị phân - Binary Tree', includes a diagram of a binary tree, a description, and metadata such as ID, title, type, and view count. The second card, titled 'Danh sách liên kết đơn - Single linked list', includes a description and a 'DANH SÁCH LIÊN KẾT ĐƠN' diagram. Both cards have 'Chỉnh sửa' (Edit) and 'Xóa' (Delete) buttons.

Hình 4: Giao diện chính của người quản lý

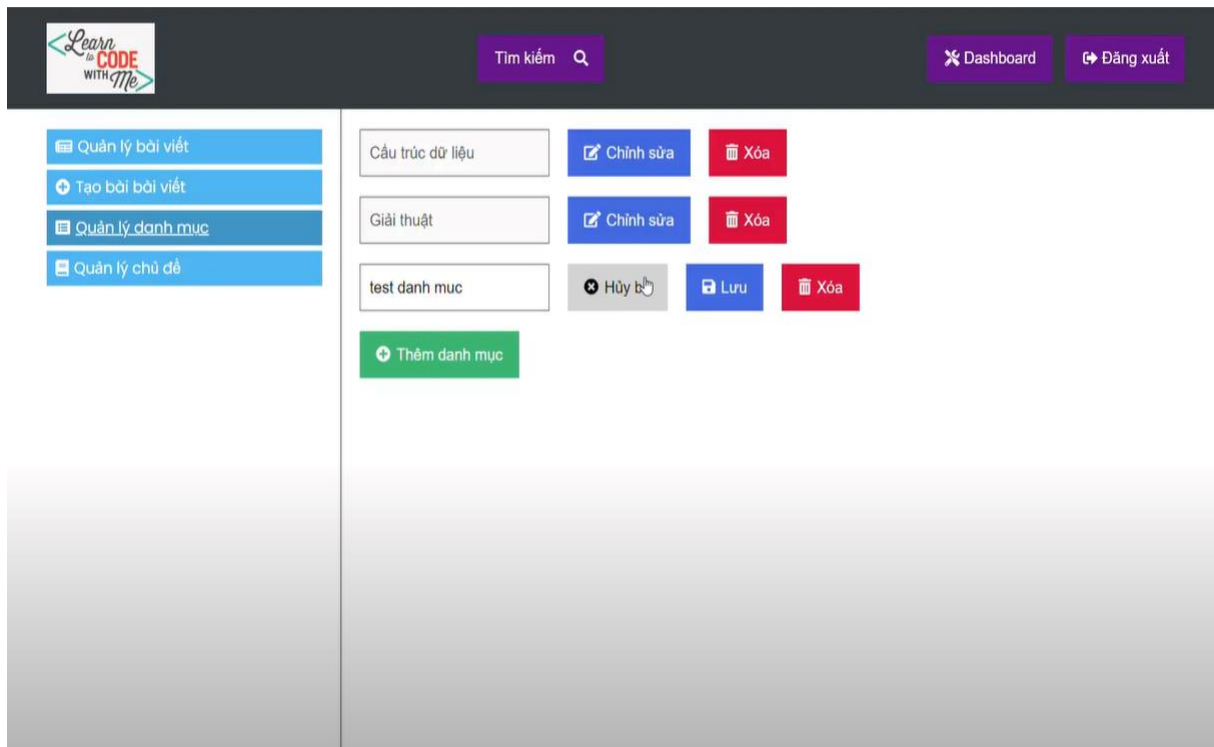
3.1.2 Giao diện chức năng tạo bài viết:

The screenshot displays the article creation form. It includes a sidebar with navigation links: 'Quản lý bài viết', 'Tạo bài viết', 'Quản lý danh mục', and 'Quản lý chủ đề'. The main content area contains the following fields:

- Nhập tên bài viết ***: A text input field.
- Nhập mô tả ***: A text area for the description.
- Upload ảnh ***: A file upload button labeled 'Choose File' and 'No file chosen'.
- Chọn loại ***: A dropdown menu with the option '-- Nhập loại --'.

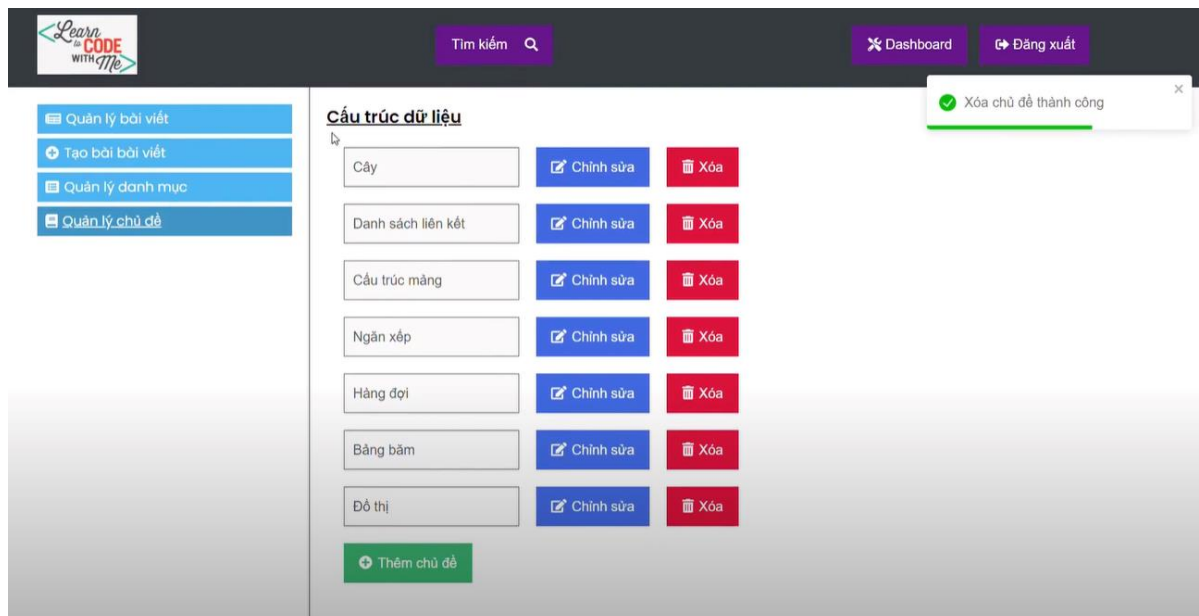
Hình 5: Giao diện chức năng tạo bài viết

3.1.3 Giao diện chức năng quản lý danh mục:



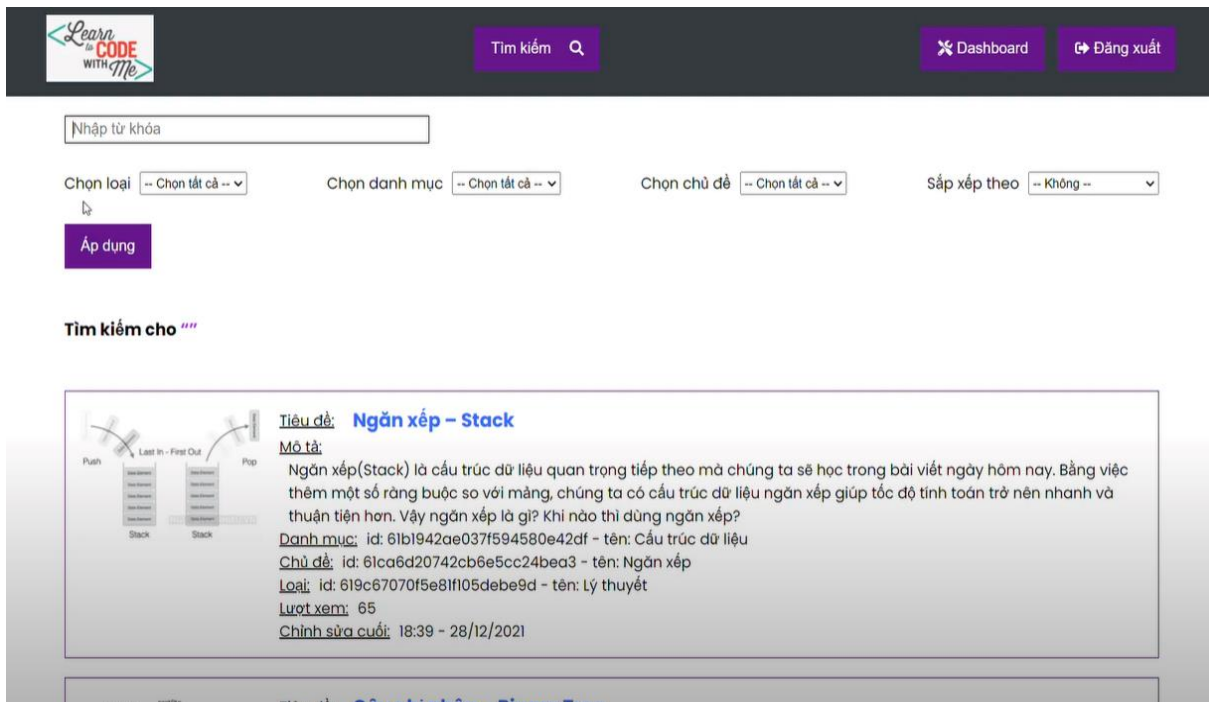
Hình 6: Giao diện chức năng quản lý chủ đề

3.1.4 Giao diện chức năng quản lý chủ đề:



Hình 7: Giao diện chức năng quản lý chủ đề

3.1.5 Giao diện chức năng tìm kiếm:



Learn CODE WITH Me

Tìm kiếm

Dashboard Đăng xuất

Nhập từ khóa

Chọn loại -- Chọn tất cả -- Chọn danh mục -- Chọn tất cả -- Chọn chủ đề -- Chọn tất cả -- Sắp xếp theo -- Không --

Áp dụng

Tìm kiếm cho ""

Tiêu đề: Ngăn xếp - Stack

Mô tả: Ngăn xếp(Stack) là cấu trúc dữ liệu quan trọng tiếp theo mà chúng ta sẽ học trong bài viết ngày hôm nay. Bằng việc thêm một số ràng buộc so với mảng, chúng ta có cấu trúc dữ liệu ngăn xếp giúp tốc độ tính toán trở nên nhanh và thuận tiện hơn. Vậy ngăn xếp là gì? Khi nào thì dùng ngăn xếp?

Danh mục: id: 61b1942ae037f594580e42df - tên: Cấu trúc dữ liệu

Chủ đề: id: 61ca6d20742cb6e5cc24bea3 - tên: Ngăn xếp

Loại: id: 619c670f5e81f105debe9d - tên: Lý thuyết

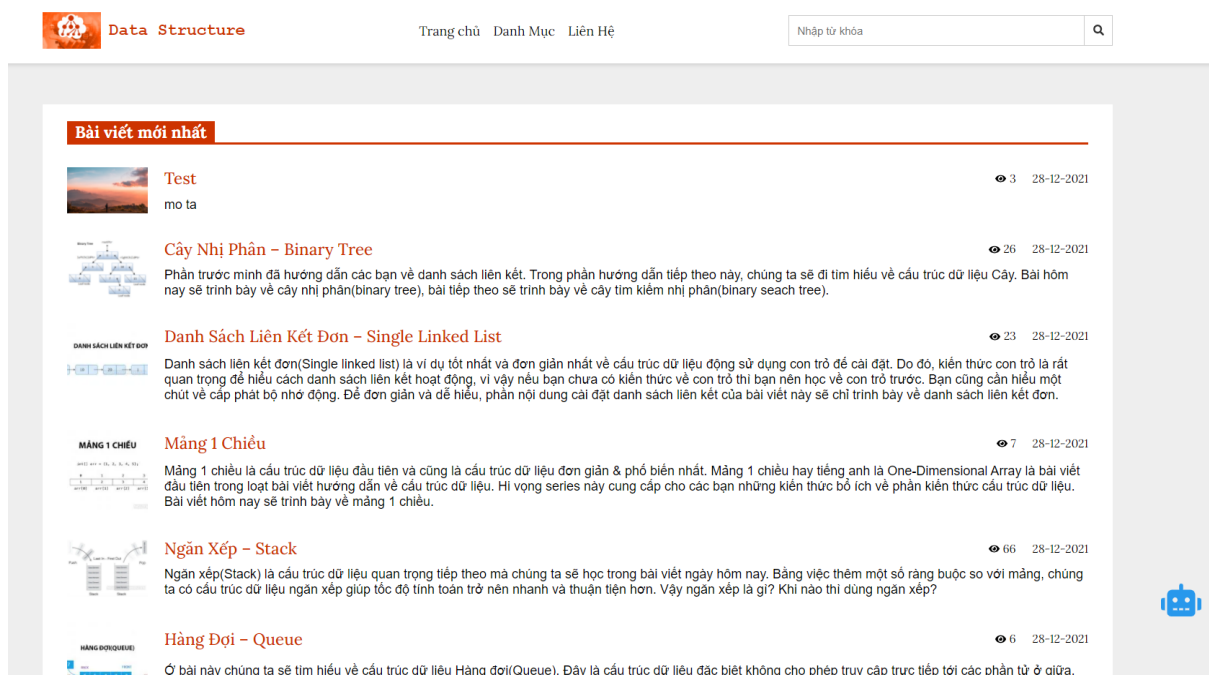
Lượt xem: 65

Chỉnh sửa cuối: 18:39 - 28/12/2021

Hình 8: Giao diện chức năng tìm kiếm bài viết

3.2 Giao diện của người dùng:

3.2.1 Giao diện chính của người dùng:



Data Structure

Trang chủ Danh Mục Liên Hệ

Nhập từ khóa

Bài viết mới nhất

Test 3 28-12-2021

Cây Nhị Phân - Binary Tree 26 28-12-2021

Danh Sách Liên Kết Đơn - Single Linked List 23 28-12-2021


Mảng 1 Chiều 7 28-12-2021

Ngăn Xếp - Stack 66 28-12-2021


Hàng Đợi - Queue 6 28-12-2021

Hình 9: Giao diện chính của người dùng

3.2.2 Giao diện chức năng xem danh mục bài học:

 **Data Structure**

Trang chủ Danh Mục Liên Hệ




LEARN DATA STRUCTURE
organizing data

Cấu Trúc Dữ Liệu

- Bảng băm
- Cây
- Cấu trúc mảng
- Danh sách liên kết
- Hàng đợi
- Ngăn xếp
- Đồ thị

Giải Thuật

- Sắp xếp
- Tìm kiếm

> Cấu trúc dữ liệu > Danh sách liên kết
Danh Sách Liên Kết Đơn – Single Linked List
Chỉnh sửa lần cuối: 28-12-2021  25


1. Lý thuyết về danh sách liên kết

Về bản chất, danh sách liên kết có chức năng như một mảng, có thể thêm và xóa các phần tử ở bất kỳ vị trí nào khi cần thiết. Một sự khác nhau giữa danh sách liên kết và mảng:


Nội dung	Mảng	Danh sách liên kết
Kích thước	<ul style="list-style-type: none">Kích thước cố địnhCần chỉ rõ kích thước trong khi khai báo	<ul style="list-style-type: none">Kích thước thay đổi trong quá trình thêm/ xóa phần tửKích thước tối đa phụ thuộc vào bộ nhớ
Cấp phát bộ nhớ	<ul style="list-style-type: none">Tĩnh: Bộ nhớ được cấp phát trong quá trình biên dịch	<ul style="list-style-type: none">Động: Bộ nhớ được cấp phát trong quá trình chạy
Thứ tự & sắp xếp	<ul style="list-style-type: none">Được lưu trữ trên một dãy ô nhớ liên tục	<ul style="list-style-type: none">Được lưu trữ trên các ô nhớ ngẫu nhiên
Truy cập	<ul style="list-style-type: none">Truy cập tới phần tử ngẫu nhiên trực tiếp bằng cách sử dụng chỉ số mảng: $O(1)$	<ul style="list-style-type: none">Truy cập tới phần tử ngẫu nhiên cần phải duyệt từ đầu/cuối đến phần tử đó: $O(n)$
Tìm kiếm	<ul style="list-style-type: none">Tìm kiếm tuyến tính hoặc tìm kiếm nhị phân	<ul style="list-style-type: none">Chỉ có thể tìm kiếm tuyến tính

Hình 10: Giao diện chức năng xem danh mục bài học


3.2.3 Giao diện liệt kê bài viết được tra cứu nhiều nhất:


 **Data Structure**


Trang chủ Danh Mục Liên Hệ





Bài viết mới nhất


 **Test**
mo ta


 **Cây Nhị Phân – Binary Tree**
Phản trước mình đã hướng dẫn các bạn về danh sách liên kết. Trong phần hướng dẫn tiếp theo này, chúng ta sẽ đi tìm hiểu về cấu trúc dữ liệu Cây. Bài hôm nay sẽ trình bày về cây nhị phân(binary tree), bài tiếp theo sẽ trình bày về cây tìm kiếm nhị phân(binary search tree).


 **Danh Sách Liên Kết Đơn – Single Linked List**
Danh sách liên kết đơn(Single linked list) là ví dụ tốt nhất và đơn giản nhất về cấu trúc dữ liệu động sử dụng con trỏ để cài đặt. Do đó, kiến thức con trỏ là rất quan trọng để hiểu cách danh sách liên kết hoạt động, vì vậy nếu bạn chưa có kiến thức về con trỏ thì bạn nên học về con trỏ trước. Bạn cũng cần hiểu một chút về cấp phát bộ nhớ động. Để đơn giản và dễ hiểu, phần nội dung cài đặt danh sách liên kết của bài viết này sẽ chỉ trình bày về danh sách liên kết đơn.


 **Mảng 1 Chiều**
Mảng 1 chiều là cấu trúc dữ liệu đầu tiên và cũng là cấu trúc dữ liệu đơn giản & phổ biến nhất. Mảng 1 chiều hay tiếng anh là One-Dimensional Array là bài viết đầu tiên trong loạt bài viết hướng dẫn về cấu trúc dữ liệu. Hi vọng series này cung cấp cho các bạn những kiến thức bổ ích về phần kiến thức cấu trúc dữ liệu. Bài viết hôm nay sẽ trình bày về mảng 1 chiều.


 **Ngăn Xếp – Stack**
Ngăn xếp(Stack) là cấu trúc dữ liệu quan trọng tiếp theo mà chúng ta sẽ học trong bài viết ngày hôm nay. Bằng việc thêm một số ràng buộc so với mảng, chúng ta có cấu trúc dữ liệu ngăn xếp giúp tốc độ tính toán trở nên nhanh và thuận tiện hơn. Vậy ngăn xếp là gì? Khi nào thì dùng ngăn xếp?


 **Hàng Đợi – Queue**
Ở bài này chúng ta sẽ tìm hiểu về cấu trúc dữ liệu Hàng đợi(Queue). Đây là cấu trúc dữ liệu đặc biệt không cho phép truy cập trực tiếp tới các phần tử ở giữa. Bài này sẽ trình bày cho các bạn lý thuyết về hàng đợi, cách cài đặt hàng đợi và một số biến thể của hàng đợi trong C/C++.

 **Cây Nhị Phân – Binary Tree**

 **Test**


 **Cấu Trúc Dữ Liệu**

 **Cây Nhị Phân**



Hình 11: Giao diện liệt kê bài viết được tra cứu nhiều nhất

3.2.4 Giao diện tra cứu chính:

**Data Structure**

Trang chủ Danh Mục Liên Hệ

Q

Cấu trúc dữ liệu


Cấu trúc mảng

Chọn loại

Sắp xếp theo


Áp dụng

Kết quả tìm kiếm của “cây nhị phân”

**Cây Nhị Phân – Binary Tree**


👁 27 28-12-2021

Phản trước mình đã hướng dẫn các bạn về danh sách liên kết. Trong phần hướng dẫn tiếp theo này, chúng ta sẽ đi tìm hiểu về cấu trúc dữ liệu Cây. Bài hôm nay sẽ trình bày về cây nhị phân(binary tree), bài tiếp theo sẽ trình bày về cây tìm kiếm nhị phân(binary search tree).

**Danh Sách Liên Kết Đơn – Single Linked List**


👁 24 28-12-2021

Danh sách liên kết đơn(Single linked list) là ví dụ tốt nhất và đơn giản nhất về cấu trúc dữ liệu động sử dụng con trỏ để cài đặt. Do đó, kiến thức con trỏ là rất quan trọng để hiểu cách danh sách liên kết hoạt động, vì vậy nếu bạn chưa có kiến thức về con trỏ thì bạn nên học về con trỏ trước. Bạn cũng cần hiểu một chút về cấp phát bộ nhớ động. Để đơn giản và dễ hiểu, phần nội dung cài đặt danh sách liên kết của bài viết này sẽ chỉ trình bày về danh sách liên kết đơn.

**Bảng Băm – Hash Tables**


👁 8 28-12-2021

Trong khoa học máy tính, bảng băm(Hash Tables) là một cấu trúc dữ liệu sử dụng hàm băm để ánh xạ từ giá trị xác định, được gọi là khóa (ví dụ như tên của một người), đến giá trị tương ứng (ví dụ như số điện thoại của họ). Do đó, bảng băm là một mảng kết hợp. Hàm băm được sử dụng để chuyển đổi từ khóa thành chỉ số (giá trị băm) trong mảng lưu trữ các giá trị tìm kiếm.

**Cấu Trúc Dữ Liệu Đồ Thị – Graph**

👁 9 28-12-2021

Đồ thị là một đối tượng tổ hợp (combinatorial object) được nghiên cứu và ứng dụng rất nhiều trong thực tế.


**Tìm Kiếm Nhị Phân – Binary Search**

👁 5 28-12-2021

Thuật toán tìm kiếm nhị phân là một trong các thuật toán sắp xếp được sử dụng rất nhiều trong thực tế. Hãy cùng mình đi tìm hiểu thuật toán tìm kiếm này nhé. Tìm kiếm là một phần không thể thiếu của mọi ứng dụng, website hay phần mềm. Tính năng tìm kiếm cho phép người sử dụng nhanh chóng truy vấn và tìm kiếm các bản ghi theo mong muốn. Và một công cụ tìm kiếm nổi tiếng nhất hàng ngày chúng ta vẫn thường sử dụng đó là Google search.

Hình 12: Giao diện tra cứu thông tin bài học


3.2.5 Giao diện tra cứu bằng chatbot:

**Data Structure**

Trang chủ Danh Mục Liên Hệ


Q

Bài viết mới nhất


**Test**

👁 3 28-12-2021


mo ta

**Cây Nhị Phân – Binary Tree**


Phản trước mình đã hướng dẫn các bạn về danh sách liên kết. Trong phần hướng dẫn tiếp theo này, chúng ta sẽ đi tìm hiểu về cấu trúc dữ liệu Cây. Bài hôm nay sẽ trình bày về cây nhị phân(binary tree), bài tiếp theo sẽ trình bày về cây tìm kiếm nhị phân(binary search tree).

**Danh Sách Liên Kết Đơn – Single Linked List**


Danh sách liên kết đơn(Single linked list) là ví dụ tốt nhất và đơn giản nhất về cấu trúc dữ liệu động sử dụng con trỏ để cài đặt. Do đó, kiến thức con trỏ là rất quan trọng để hiểu cách danh sách liên kết hoạt động, vì vậy nếu bạn chưa có kiến thức về con trỏ thì bạn nên học về con trỏ trước. Bạn cũng cần hiểu một chút về cấp phát bộ nhớ động. Để đơn giản và dễ hiểu, phần nội dung cài đặt danh sách liên kết của bài viết này sẽ chỉ trình bày về danh sách liên kết đơn.

**Mảng 1 Chiều**

Mảng 1 chiều là cấu trúc dữ liệu đầu tiên và cũng là cấu trúc dữ liệu đơn giản & phổ biến nhất. Mảng 1 chiều hay tiếng anh là One-Dimension array. Bài viết hôm nay sẽ trình bày về mảng 1 chiều.

**Ngăn Xếp – Stack**

Ngăn xếp(Stack) là cấu trúc dữ liệu quan trọng tiếp theo mà chúng ta sẽ học trong bài viết ngày hôm nay. Bằng việc thêm một số ràng buộc cho cấu trúc dữ liệu ngăn xếp giúp tốc độ tính toán trở nên nhanh và thuận tiện hơn. Vậy ngăn xếp là gì? Khi nào thì dùng ngăn xếp?

**Hàng Đợi – Queue**

Ở bài này chúng ta sẽ tìm hiểu về cấu trúc dữ liệu Hàng đợi(Queue). Đây là cấu trúc dữ liệu đặc biệt không cho phép truy cập trực tiếp tới các phần tử ở giữa. Bài viết sẽ trình bày về cách cài đặt hàng đợi và một số kiến thức cần biết về hàng đợi.

Conversation with Bot

B: Xin chào! Tên tôi là Daisy. Tôi có thể giúp bạn trả lời một số định nghĩa!

B: Cây nhị phân là một cấu trúc dữ liệu đặc biệt được sử dụng cho mục đích lưu trữ dữ liệu. Một cây nhị phân có một điều kiện đặc biệt là mỗi nút có thể có tối đa hai nút con. Một cây nhị phân tận dụng lợi thế của hai kiểu

👁 6 28-12-2021

Hình 13: Giao diện tra cứu bằng chatbot

KẾT LUẬN VÀ KHUYẾN NGHỊ

❖ Kết quả đạt được

- Thực hiện thành công ứng dụng. Có thêm kiến thức về cơ sở dữ liệu mongodb , framework react và đặc biệt là ngôn ngữ javascript.
- Tra cứu theo nội dung bài học.
- Tra cứu theo kiến thức liên quan.
- Tra cứu bài viết theo danh mục, chủ đề, loại bài viết, sắp xếp các bài viết.
- Tra cứu bằng chatbot
- Xem bài viết mới nhất.
- Gợi ý từ khóa được tìm kiếm nhiều nhất.
- Gợi ý bài viết theo từ khóa đang tìm bằng tiêu đề.
- Gợi ý bài viết liên quan khi xem 1 bài viết
- Phần mềm thân thiện, hài hòa, với người dùng.

❖ Hạn chế của đề tài

- Nhiều trường hợp thực tế áp dụng vào ứng dụng quản lý vẫn chưa giải quyết được.
- Các chức năng còn chưa thực sự tối ưu.

❖ Hướng dẫn phát triển của đề tài

- Tiếp tục phát triển hoàn thiện ứng dụng giúp cho người dùng, có thể giải quyết các vấn đề gặp phải trong thực tế.
- Ứng dụng hướng tới việc phát triển mở rộng toàn phần như người dùng có thể đăng ký, đăng nhập, bình luận bài viết lưu các bài viết đã xem.
- Bổ sung các hạn chế của đề tài, mở rộng các chức năng của hệ thống phần mềm, đảm bảo độ tin cậy, và chính xác của hệ thống.
- Linh hoạt, thân thiện hơn cho người sử dụng.

TÀI LIỆU THAM KHẢO

- [1]. **Đỗ Xuân Lôi**, *Cấu trúc dữ liệu và giải thuật*, NXB ĐHQGHN, 2007.
- [2]. **Phạm Thế Bảo**, *Cấu trúc dữ liệu và giải thuật*, NXB ĐHQG TP.HCM, 2013.
- [3]. **Đinh Mạnh Tường**, *Cấu trúc dữ liệu và thuật toán*, NXB ĐHQG HN, 2008.
- [4]. **Dương Anh Đức -Trần Hạnh Nhi**, *Nhập môn Cấu trúc dữ liệu và giải thuật*, NXB ĐHQG TP.HCM, 2003
- [5]. **Steve Suehring**, *JavaScript Hướng dẫn học qua ví dụ*, NXB Khoa Học & Kỹ Thuật, 2013.

DANH SÁCH PHÂN CÔNG KHỐI LƯỢNG THỰC HIỆN ĐỀ TÀI

Bảng 3: Công việc của mỗi thành viên và mức độ hoàn thành

Tên Thành viên	Chi tiết công việc	Số điểm đánh giá của cả nhóm
Bùi Văn Tân	<ul style="list-style-type: none">- Bàn luận- Vẽ biểu đồ phân rã- Code thiết kế giao diện	10 đ
Nguyễn Công Chí	<ul style="list-style-type: none">- Bàn luận- Vẽ biểu đồ Use Case	10 đ
Võ Văn Trí	<ul style="list-style-type: none">- Bàn luận- Thiết kế cơ sở dữ liệu- Code xử lý chức năng	10đ
Huỳnh Xuân Lãm	<ul style="list-style-type: none">- Bàn luận- Soạn word	10 đ
Hồ Ngọc Thống	<ul style="list-style-type: none">- Bàn luận- Soạn powerpoint	10 đ