|  |
| --- |
| TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI  KHOA CÔNG NGHỆ THÔNG TIN  ---------------o0o---------------    **Báo cáo đồ án tốt nghiệp**  **Tên đề tài: Xây dựng website quản lý tài liệu sử dụng công nghệ ElasticSearch**  Giảng viên hướng dẫn: TS. Nguyễn Quốc Tuấn  Sinh viên thực hiện : Trương Tiến Đạt  Lớp : Công Nghệ Thông tin 2 \_ K62  Mã sinh viên : 211204032  Hà Nội – 2025 |

**Lời mở đầu**

Hiện nay, việc sử dụng và quản lý tài liệu khi làm việc trực tuyến ngày càng trở nên quan trọng hơn bao giờ hết. Vì vậy, dự án tôi xin giới thiệu tới các bạn đọc một ứng dụng quản lý tài liệu nhằm phục vụ cho hầu hết các công việc liên quan đến sử dụng tài liệu trong khi làm việc trực tuyến. Hệ thống với giao diện thân thiện với người dùng và dễ dàng sử dụng. Trong hệ thống, người dùng có thể tìm kiếm toàn văn với từ khóa mong muốn để có thể tìm tài liệu, hệ thống sẽ đưa ra danh sách các tài liệu liên quan một cách chính xác và nhanh nhất có thể.

Nhóm dự án xin cam đoan ứng dụng “Quản lý tài liệu” của nhóm là sản phẩm được phát triển bởi nhóm dưới sự hướng dẫn của giảng viên Hoàng Xuân Tùng. Tất cả những tài liệu tham khảo và hướng dẫn đều được nêu rõ ở mục “Tài liệu tham khảo”, những số liệu, thông tin trong báo cáo là hoàn toàn trung thực và minh bạch. Nếu có gì sai sót, nhóm dự án xin chịu trách nhiệm và kỷ luật của khoa.

Cuối cùng, nhóm dự án xin được gửi lời cảm ơn tới giảng viên Hoàng Xuân Tùng đã giúp đỡ nhóm nhiệt tình và tận tâm để nhóm có thể lên kế hoạch phát triển và hoàn thành sản phẩm. Bên cạnh đó, nhóm dự án cũng xin được gửi lời cảm ơn tới các thầy/cô của khoa Công Nghệ Thông Tin Trường Đại Học Công Nghệ đã tạo điều kiện để nhóm được làm việc cùng nhau một cách chuyên nghiệp và gắn kết với nhau như những đồng nghiệp. Một lần nữa, nhóm xin được gửi lời cảm ơn tới tất cả các thầy cô cũng như nhà Trường đã hỗ trợ và giúp đỡ cho nhóm dự án có thể bảo vệ và tốt nghiệp trong điều kiện dịch bệnh diễn biến phức tạp. Chính vì vậy, kết quả cũng như tính khả thi của dự án này trong thực tiễn là lời cảm ơn chân thành nhất của nhóm gửi tới mọi người và cũng là nguồn độc lực để nhóm có thể cùng nhau làm việc và phát triển trong tương lai.

**Mục lục**

[**Chương I: Tổng quan về lập trình web** 4](#_Toc190304981)

[**1.** **Website là gì?** 4](#_Toc190304982)

[**2.** **Lập trình web là gì?** 4](#_Toc190304983)

[**3.** **Lịch sử phát triển của lập trình web** 4](#_Toc190304984)

[**4.** **Tương lai của lập trình web** 4](#_Toc190304985)

[**Chương II: T** **ìm hiểu ngôn ngữ lập trình PHP** 5](#_Toc190304986)

[**1.** **Ngôn ngữ lập trình PHP là gì ?** 5](#_Toc190304987)

[**2.** **Ứng dụng của ngôn ngữ PHP trong lập trình** 5](#_Toc190304988)

[**3.** **Ưu nhược điểm của ngôn ngữ lập trình PHP:** 6](#_Toc190304989)

[**4.** **Giới thiệu Laravel Framework** 6](#_Toc190304996)

[**5. Sự khác nhau giữa các framework web khác với Laravel** 9](#_Toc190305000)

[**Chương III: Chương trình demo** 10](#_Toc190305001)

[**1.** **Nội dung chương trình** 10](#_Toc190305002)

[**1.1.** **Tổng quan** 10](#_Toc190305003)

[**1.1.** **Trang danh sách danh mục** 10](#_Toc190305003)

[**1.1.** **Trang Thêm/Sửa Danh mục** 10](#_Toc190305003)

I. Introduction

1.Giới thiệu chung

* Elasticsearch là một công cụ tìm kiếm dựa trên nền tảng Apache Lucene. Nó cung cấp một bộ máy tìm kiếm dạng phân tán, có đầy đủ công cụ với một giao diện web HTTP có hỗ trợ dữ liệu JSON. Elasticsearch được phát triển bằng Java và được phát hành dạng nguồn mở theo giấy phép Apache.
* Elasticsearch thực chất hoạt động như 1 web server, có khả năng tìm kiếm nhanh chóng (near real time) thông qua giao thức RESTful.
* Elasticsearch là 1 hệ thống phân tán và có khả năng mở rộng tuyệt vời (horizontal scalability). Lắp thêm node cho nó là nó tự động auto mở rộng cho bạn.

2. Data model

* Elasticsearch là một hệ NoSQL database.
* Elasticsearch là một hệ document store database.

3. Mục đích

* Công cụ hỗ trợ cho những app có chức năng search hoặc yêu cầu phức tạp.
* Elasticsearch có khả năng phân tích và thống kê dữ liệu.
* Elasticsearch chạy trên server riêng và đồng thời giao tiếp thông qua RESTful do vậy nên nó không phụ thuộc vào client viết bằng gì hay hệ thống hiện tại của bạn viết bằng gì. Nên việc tích hợp nó vào hệ thống bạn là dễ dàng, bạn chỉ cần gửi request http lên là nó trả về kết quả.

4. Khả năng sử dụng

* Khi bạn chạy một web bán hàng online, bạn cho phép khách hàng có thể tìm kiếm tìm kiếm sản phẩm của bạn. Trong trường hợp này bạn có thể sử dụng ElasticSearch để lưu trữ toàn bộ danh mục sản phẩm, cung cấp các đề xuất tìm kiếm. Lưu trữ các khoảng không quảng cáo và làm đầy chúng khi cần thiết.
* Bạn muốn có một tập log hoặc có tập dữ liệu trao đổi và bạn muốn phân tích chúng thành các data dưới dạng xu hướng, thống kê, tóm tắt, hoặc không phải các loại trên. Trong trường hợp này bạn có thể sử dụng Logstash (một phần của ElasticSearch) để thu thập, tổng hợp và phân tích cú pháp dữ liệu của bạn, sau đó chuyển dữ liệu từ Logstash vào ElasticSearch. Lúc này, bạn có thể search hoặc tổng hợp thông tin theo cách mình muốn.
* Bạn có nhu cầu phân tích, hoặc kinh doanh mà muốn điều tra, phân tích, hoặc cái nhìn trực quan hoặc đặt câu hỏi quảng cáo cho một dữ liệu rất lớn. Trường hợp này , bạn cần sử dụng ElasticSearch để lưu trữ data của bạn và sau đó sử dụng Kibana (một phần của Elasticsearch/ Logstash/Kibana stack) để xây dựng bảng điều tra , điều đó giúp bạn có thể trực quan hóa data của mình, đây cũng là điều rất quan trọng. Thêm vào đó, bạn có thể sử dụng chức năng tổng hợp của Elasticsearch để xây dựng nhũng câu truy vấn thông minh phù hợp với yêu cầu bài toán.

5. Ưu điểm

* Tìm kiếm dữ liệu rất nhanh chóng, mạnh mẽ dựa trên Apache Lucene ( near - real time searching).
* Có khả năng phân tích dữ liệu (Analysis data).
* Khả năng mở rộng theo chiều ngang tuyệt vời.
* Hỗ trợ tìm kiếm mờ (fuzzy), tức là từ khóa tìm kiếm có thể bị sai lỗi chính tả hay không đúng cú pháp thì vẫn có khả năng elasticsearch trả về kết quả tốt.
* Hỗ trợ Structured Query DSL (Domain-Specific Language ), cung cấp việc đặc tả những câu truy vấn phức tạp một cách cụ thể và rõ ràng bằng JSON.
* Hỗ trợ nhiều Elasticsearch client như Java, PhP, Javascript, Ruby, .NET, Python.

6. Nhược điểm

* Elasticsearch được thiết kế cho mục đích search, do vậy với những nhiệm vụ khác ngoài search như CRUD thì Elasticsearch kém thế hơn so với những database khác như Mongodb, Mysql …. Do vậy người ta ít khi dùng Elasticsearch làm database chính, mà thường kết hợp nó với 1 database khác.
* Trong Elasticsearch không có khái niệm database transaction , tức là nó sẽ không đảm bảo được toàn vẹn dữ liệu trong các hoạt động Insert, Update, Delete. Tức khi chúng ta thực hiện thay đổi nhiều bản ghi nếu xảy ra lỗi thì sẽ làm cho logic của mình bị sai hay dẫn tới mất mát dữ liệu. Đây cũng là 1 phần khiến Elasticsearch không nên là database chính.
* Không thích hợp với những hệ thống thường xuyên cập nhật dữ liệu. Sẽ rất tốn kém cho việc đánh index dữ liệu.

7. Bảo mật

* Các tính năng bảo mật của Elasticsearch cung cấp cho người dùng quyền truy cập phù hợp. Các nhóm công nghệ thông tin, vận hành và ứng dụng có thể tận dụng các khả năng phù hợp này để quản lý người dung có thiện chí và đẩy lùi các tác nhân độc hại.

8. Ứng dụng nổi tiếng đã sử dụng HQT CSDL Elasticsearch

* Mozilla, Quora.
* Sound Cloud, GitHub.
* Stack Exchange.
* Center for Open Science.
* Reverb, Netflix.
* Người dùng chỉ truy cập trên dữ liệu của Elasticsearch.
* Người quản trị thì truy cập trực tiếp trên Database SQL (MySQL) để cập nhật thêm xóa sửa và nó tự động cập nhật sang ElasticSearch.

9. Elasticsearch có thể tích hợp được với tất cả các ứng dụng có sử dụng ngôn ngữ

* Java
* JavaScript
* Groovy
* .NET
* PHP
* Perl
* Python
* Ruby

II. Data model / Type

1. Mô hình dữ liệu

* Cơ sở dữ liệu NoSQL là Cơ sở dữ liệu được xây dựng dành riêng cho mô hình dữ liệu và có sơ đồ linh hoạt để xây dựng các ứng dụng hiện đại. Cơ sở dữ liệu NoSQL được công nhận rộng rãi vì khả năng dễ phát triển, chức năng cũng như hiệu năng ở quy mô lớn. Các Cơ sở dữ liệu này sử dụng nhiều mô hình dữ liệu đa dạng, trong đó có văn bản, đồ thị, khóa – giá trị, trong bộ nhớ và tìm kiếm.
* Document stores, hay còn gọi là hệ thống database hướng document (document-oriented) là một kiểu database mà ở đó data được tổ chức một cách tự do không theo một lược đồ nào cả (schema-free).

Điều này có nghĩa là:

* Mỗi bản ghi không cần phải có cấu trúc cố định, các bản ghi khác nhau có thể có nhiều cột khác nhau.
* Loại dữ liệu trong mỗi cột giữa các bản ghi cũng có thể khác nhau.
* Một cột có thể có nhiều hơn một giá trị (mảng – array).
* Các bản ghi có thể có cấu trúc lồng nhau (trong một bản ghi chứa một hoặc nhiều bản ghi khác – nested structure).

2. Khái niệm cơ bản

* RESTful API là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.
* Node là trung tâm hoạt động của Elasticsearch. Là nơi lưu trữ dữ liệu, tham gia thực hiện đánh index của cluster cũng như thực hiện các thao tác tìm kiếm. Mỗi node được định danh bằng 1 unique name.
* Cluster là tập hợp các nodes hoạt động cùng với nhau, chia sẻ cùng thuộc tính cluster.name. Chính vì thế Cluster sẽ được xác định bằng 1 'unique name'. Việc định danh các cluster trùng tên sẽ gây nên lỗi cho các node vì vậy khi setup cần hết sức chú ý điểm này. Mỗi cluster có một node chính (master), được lựa chọn một cách tự động và có thể thay thế nếu sự cố xảy ra. Một cluster có thể gồm 1 hoặc nhiều nodes. Các nodes có thể hoạt động trên cùng 1 server. Tuy nhiên trong thực tế, một cluster sẽ gồm nhiều nodes hoạt động trên các server khác nhau để đảm bảo nếu 1 server gặp sự cố thì server khác (node khác) có thể hoạt động đầy đủ chức năng so với khi có 2 servers. Các node có thể tìm thấy nhau để hoạt động trên cùng 1 cluster qua giao thức unicast. Chức năng chính của Cluster đó chính là quyết định xem shards nào được phân bổ cho node nào và khi nào thì di chuyển các Cluster để cân bằng lại Cluster.
* Index trong Elasticsearch là cấu trúc inverted index, được thiết kế để cho phép tìm kiếm full-text. Cách thức của nó khá đơn giản, các văn bản được phân tách ra thành từng từ có nghĩa sau đó sẽ được map xem thuộc văn bản nào.
* Document là một JSON object với một số dữ liệu. Đây là basic information unit trong Elasticsearch. Hiểu 1 cách cơ bản thì đây là đơn vị nhỏ nhất để lưu trữ dữ liệu trong Elasticsearch.
* Shard là đối tượng của Lucene , là tập con các documents của 1 Index. Một Index có thể được chia thành nhiều Shard. Mỗi node bao gồm nhiều Shard. Chính vì thế Shard mà là đối tượng nhỏ nhất, hoạt động ở mức thấp nhất, đóng vai trò lưu trữ dữ liệu. Chúng ta gần như không bao giờ làm việc trực tiếp với các Shard vì Elasticsearch đã support toàn bộ việc giao tiếp cũng như tự động thay đổi các Shard khi cần thiết. Có 2 loại Shard là : primary shard và replica shard.
* Primary Shard: Primary Shard là sẽ lưu trữ dữ liệu và đánh index. Sau khi đánh xong dữ liệu sẽ được vận chuyển tới các Replica Shard. Mặc định của Elasticsearch là mỗi index sẽ có 5 Primary shard và với mỗi Primary shard thì sẽ đi kèm với 1 Replica Shard.
* Replica Shard: đúng như cái tên của nó, nó là nơi lưu trữ dữ liệu nhân bản của Primary Shard. Replica Shard có vai trò đảm bảo tính toàn vẹn của dữ liệu khi Primary Shard xảy ra vấn đề. Ngoài ra Replica Shard có thể giúp tăng cường tốc độ tìm kiếm vì chúng ta có thể setup lượng Replica Shard nhiều hơn mặc định của Elasticsearch.

3. Lưu trữ

* Lưu trữ dữ liệu theo dạng JSON – JSON document storage.
* Sử dụng RESTful API để lưu trữ document.
* Mỗi document gửi tới Elasticsearch được lưu trữ qua một thuật toán và sau đó được gửi đến shard. Elasticsearch cố gắng để phân tán document thông qua các shard.
* Khi lưu trữ document, Elasticsearch tạo ra inverted index, map các thuật ngữ / từ khóa xuất hiện trong document này tới chính document đó.
* Một điều quan trọng khi lưu trữ document đó là được quyết định cách tốt nhất để lưu trữ chúng, giúp nâng cao tốc độ truy vấn. Khi thiết kế các giải pháp sử dụng Elasticsearch, điều đáng lưu tâm nhất khi lưu trữ document chính là: tôi sẽ truy vấn document này như thế nào? “First query” này tiếp cận với việc sử dụng cho tất cả các khả năng Elasticsearch indexing để thực hiện truy vấn cực kỳ nhanh chóng.