****

**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT**

**KHOA CÔNG NGHỆ SỐ**

**----------🙢🕮🙠----------**

A blue and white sign with a cloud and gears

Description automatically generated

**BÀI TẬP MÔN HỌC**

**CƠ SỞ DỮ LIỆU PHI QUAN HỆ**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE QUẢN LÝ GHI CHÚ**

Người hướng dẫn: **Th.S Ngô Lê Quân**

Sinh viên thực hiện: **Đoàn Huỳnh Ngọc Sơn**

Mã sinh viên: **21115053120140**

Lớp**: 124CPQH01**

Nhóm: **20**

**Đà Nẵng, 12/2**

**LỜI NÓI ĐẦU**

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin và nhu cầu lưu trữ dữ liệu ngày càng lớn, các hệ quản trị cơ sở dữ liệu truyền thống (SQL) dần bộc lộ một số hạn chế khi đối mặt với khối lượng dữ liệu khổng lồ và yêu cầu linh hoạt cao. Trong bối cảnh đó, các hệ cơ sở dữ liệu phi quan hệ (NoSQL) đã ra đời và nhanh chóng trở thành một giải pháp tối ưu cho việc lưu trữ và xử lý dữ liệu trong các hệ thống hiện đại.

Báo cáo này được thực hiện nhằm nghiên cứu và ứng dụng cơ sở dữ liệu NoSQL, cụ thể là MongoDB, trong việc xây dựng một hệ thống website ghi chú đơn giản. Thông qua đề tài này, em mong muốn:

* Tìm hiểu lý thuyết cơ bản về cơ sở dữ liệu NoSQL và sự khác biệt so với SQL truyền thống.
* Phân tích và thiết kế một hệ thống lưu trữ linh hoạt, hiệu quả bằng MongoDB.
* Xây dựng và triển khai hệ thống website ghi chú với các chức năng cơ bản như tạo, xem, cập nhật và xóa ghi chú.

Báo cáo được chia thành các chương như sau:

* Phần 1: Cơ sở lý thuyết về cơ sở dữ liệu NoSQL và MongoDB.
* Phần 2: Phân tích và thiết kế hệ thống sử dụng MongoDB.
* Phần 3: Ứng dụng NoSQL trong hệ thống website ghi chú.
* Kết luận và hướng phát triển: Tổng kết kết quả đạt được và đề xuất các cải tiến trong tương lai.

Trong quá trình thực hiện báo cáo, tôi đã cố gắng tìm hiểu và áp dụng những kiến thức đã học vào thực tế. Mặc dù đã nỗ lực hoàn thành, nhưng do thời gian và kiến thức còn hạn chế, báo cáo không tránh khỏi thiếu sót, rất mong nhận được ý kiến đóng góp của thầy để hoàn thiện hơn.

Xin chân thành cảm ơn thầy Ngô Lê Quân đã hướng dẫn và tạo điều kiện thuận lợi để em hoàn thành báo cáo này!

Người thực hiện

**Đoàn Huỳnh Ngọc Sơn**

**MỤC LỤC**

Phần 1: Cơ sở lý thuyết 1

1.1. Giới thiệu cơ sở dữ liệu phi quan hệ(NoSQL) 1

1.1.1. Khái niệm 1

1.1.2. Lý do ra đời 1

1.1.3. Đặc điểm chính của NoSQL 1

1.2. Phân loại 1

1.2.1. Document-based (Dạng tài liệu) 1

1.2.2. Key-Value (Dạng khóa-giá trị) 1

1.2.3. Column-Family (Dạng cột) 1

1.2.4. Graph (Dạng đồ thị) 2

1.3. So sánh NoSQL và RDBMS 2

1.4. Các hệ quản trị cơ sở dữ liệu NoSQL phổ biến 2

1.4.1. MongoDB (Document-based) 2

1.4.2. Redis (Key-Value) 2

1.4.3. Cassandra (Column-Family) 2

1.4.4. Neo4j (Graph) 2

1.5. Ứng dụng NoSQL trong thực tế 3

Phần 2: Phân tích thiết kế hệ thống 4

2.1. Giới thiệu tổng quan hệ thống 4

2.1.1. Các chức năng cơ bản 4

2.1.2. Yêu cầu hệ thống 4

2.1.3. Công nghệ sử dụng 4

2.2. Lựa chọn cơ sở dữ liệu 4

2.2.1. Lý do chọn NoSQL (MongoDB) 4

2.2.2. Cách dữ liệu được lưu trữ 4

2.3. Thiết kế cơ sở dữ liệu 5

2.3.1. Collection “Users” 5

2.3.2. Collection “Notes” 5

2.4. Kiến trúc hệ thống 5

2.4.1. Client 5

2.4.2. Backend (Server) 5

2.4.3. Database (MongoDB) 6

2.4.4. Kết quả thiết kế 6

Phần 3: Ứng dụng NoSQL trên hệ thống website quản lý ghi chú 7

3.1. Mô tả chức năng chính của hệ thống 7

3.1.1. Đăng ký tài khoản 7

3.1.2. Đăng nhập 7

3.1.3. Xem danh sách ghi chú 7

3.1.4. Tạo ghi chú mới 7

3.1.5. Chỉnh sửa ghi chú 7

3.1.6. Xóa ghi chú đã tạo 7

3.1.7. Ghim ghi chú cần chú ý 7

3.1.8. Tìm kiếm ghi chú 7

3.2. Triển khai hệ thống 7

3.2.1. Cài đặt môi trường 7

3.2.2. Cài đặt hệ thống 8

3.2.3. Kiểm thử hệ thống 8

3.3. Kết quả đạt được 8

3.3.1. Nội dung 8

3.3.2. Hình ảnh 9

Phần 4: Kết luận và hướng phát triển 10

4.1. Kết luận 10

4.1.1. Các chức năng đã đạt được 10

4.1.2. Lợi ích đạt được 10

4.1.3. Đánh giá tổng thể 10

4.2. Hướng phát triển 10

4.2.1. Bổ sung tính năng nâng cao 10

4.2.2. Cải thiện trải nghiệm người dùng 10

4.2.3. Tích hợp tính năng đồng bộ hóa 10

4.2.4. Tối ưu hệ thống 10

4.2.5. Triển khai trên môi trường thực tế 11

1. Cơ sở lý thuyết
   1. Giới thiệu cơ sở dữ liệu phi quan hệ(NoSQL)
      1. Khái niệm

NoSQL (Not Only SQL) là một dạng cơ sở dữ liệu không sử dụng mô hình quan hệ (tables, rows, columns) như RDBMS mà thay vào đó sử dụng các mô hình linh hoạt hơn như document, key-value, column-family hoặc graph. NoSQL phù hợp với các ứng dụng yêu cầu lưu trữ dữ liệu lớn, tốc độ truy cập nhanh và khả năng mở rộng dễ dàng.

* + 1. Lý do ra đời

NoSQL ra đời nhằm giải quyết các vấn đề hạn chế của cơ sở dữ liệu quan hệ (RDBMS) như:

* Xử lý dữ liệu lớn (Big Data).
* Khả năng mở rộng ngang (Horizontal Scaling).
* Đáp ứng nhanh các yêu cầu truy vấn cho hệ thống thời gian thực.
  + 1. Đặc điểm chính của NoSQL
* Linh hoạt: Không cần schema cố định, dễ dàng thay đổi cấu trúc dữ liệu.
* Mở rộng: Dễ dàng mở rộng theo chiều ngang bằng cách thêm máy chủ mới.
* Hiệu suất cao: Tối ưu hóa cho tốc độ truy cập và xử lý dữ liệu lớn.
* Hỗ trợ nhiều dạng dữ liệu: Lưu trữ các kiểu dữ liệu phi cấu trúc hoặc bán cấu trúc như JSON, XML.
  1. Phân loại

NoSQL được chia làm bốn loại chính:

* + 1. Document-based (Dạng tài liệu)
* Dữ liệu được lưu dưới dạng tài liệu (ví dụ: JSON, BSON).
* Các hệ quản trị tiêu biểu: MongoDB, CouchDB.
  + 1. Key-Value (Dạng khóa-giá trị)
* Dữ liệu được lưu dưới dạng cặp khóa-giá trị.
* Các hệ quản trị tiêu biểu: Redis, DynamoDB.
  + 1. Column-Family (Dạng cột)
* Dữ liệu được lưu trữ trong các bảng có cột linh hoạt.
* Các hệ quản trị tiêu biểu: Cassandra, HBase.
  + 1. Graph (Dạng đồ thị)
* Dữ liệu được lưu dưới dạng các nút và cạnh để biểu diễn mối quan hệ.
* Các hệ quản trị tiêu biểu: Neo4j, ArangoDB.
  1. So sánh NoSQL và RDBMS

|  |  |  |
| --- | --- | --- |
| Tiêu chí | RDBMS | NoSQL |
| Mô hình dữ liệu | Quan hệ (Tables, Rows) | Linh hoạt (Document, Key-Value) |
| Schema | Cố định, chặt chẽ | Linh hoạt, không cố định |
| Mở rộng | Chiều dọc (Vertical) | Chiều ngang (Horizontal) |
| Truy vấn | SQL (có JOIN) | Không có JOIN, đơn giản |
| Tính nhất quán | ACID (Consistency cao) | BASE (Consistency linh hoạt) |
| Hiệu suất | Chậm khi dữ liệu lớn | Tối ưu cho dữ liệu lớn |
| Ví dụ | MySQL, PostgreSQL | MongoDB, Redis, Cassandra |

* 1. Các hệ quản trị cơ sở dữ liệu NoSQL phổ biến
     1. MongoDB (Document-based)
* Đặc điểm: Dữ liệu lưu trữ dưới dạng JSON hoặc BSON.
* Ưu điểm: Truy vấn linh hoạt, hỗ trợ tốt cho ứng dụng web và mobile.
* Sử dụng: Website, hệ thống ghi chú, blog.
  + 1. Redis (Key-Value)
* Đặc điểm: Lưu trữ dữ liệu trong bộ nhớ (In-memory).
* Ưu điểm: Tốc độ đọc/ghi cực nhanh.
* Sử dụng: Cache, quản lý phiên người dùng.
  + 1. Cassandra (Column-Family)
* Đặc điểm: Lưu trữ dữ liệu dạng bảng cột linh hoạt.
* Ưu điểm: Mở rộng tốt cho hệ thống lớn.
* Sử dụng: Hệ thống phân tán, Big Data.
  + 1. Neo4j (Graph)
* Đặc điểm: Biểu diễn dữ liệu dưới dạng đồ thị (nodes và relationships).
* Ưu điểm: Tốt cho dữ liệu có quan hệ phức tạp.
* Sử dụng: Mạng xã hội, hệ thống gợi ý.
  1. Ứng dụng NoSQL trong thực tế
* Web và Mobile: Dùng NoSQL để lưu trữ dữ liệu người dùng, nội dung (ví dụ: MongoDB cho blog hoặc website ghi chú).
* IoT (Internet of Things): Thu thập và lưu trữ dữ liệu từ các thiết bị IoT theo thời gian thực.
* Big Data: Xử lý và phân tích lượng lớn dữ liệu.
* Hệ thống gợi ý: Dùng cơ sở dữ liệu đồ thị (Graph) để gợi ý sản phẩm hoặc mối quan hệ.

1. Phân tích thiết kế hệ thống
   1. Giới thiệu tổng quan hệ thống
      1. Các chức năng cơ bản

* Đăng ký tài khoản.
* Đăng nhập.
* Tạo ghi chú mới.
* Xem danh sách các ghi chú.
* Chỉnh sửa ghi chú đã có.
* Xóa ghi chú không cần thiết.
* Ghim ghi chú.
* Tìm kiếm ghi chú.
  + 1. Yêu cầu hệ thống
* Dữ liệu cần được lưu trữ linh hoạt và không yêu cầu schema cố định.
* Hệ thống cần đảm bảo tốc độ xử lý nhanh, khả năng mở rộng khi số lượng ghi chú lớn.
  + 1. Công nghệ sử dụng
* Backend: Node.js và Express.js.
* Cơ sở dữ liệu: MongoDB (NoSQL).
* Frontend: ReactJS.
  1. Lựa chọn cơ sở dữ liệu
     1. Lý do chọn NoSQL (MongoDB)
* MongoDB là cơ sở dữ liệu NoSQL dạng document-based, rất phù hợp cho việc lưu trữ các ghi chú dưới dạng tài liệu JSON.
* Linh hoạt: Không yêu cầu schema cố định, dễ dàng mở rộng hoặc thay đổi cấu trúc dữ liệu.
* Truy vấn nhanh và đơn giản, đặc biệt cho các thao tác CRUD (Create, Read, Update, Delete).
* Khả năng mở rộng: MongoDB hỗ trợ mở rộng theo chiều ngang, phù hợp khi dữ liệu tăng trưởng lớn.
  + 1. Cách dữ liệu được lưu trữ
* Dữ liệu ghi chú sẽ được lưu trong một collection (tập hợp các tài liệu).
  1. Thiết kế cơ sở dữ liệu
     1. Collection “Users”

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | ID duy nhất của user (tự động). |
| fullName | String | Họ và tên của user. |
| email | String | Email của user. |
| password | String | Password của user. |
| createOn | DateTime | Ngày giờ tạo tài khoản của user. |

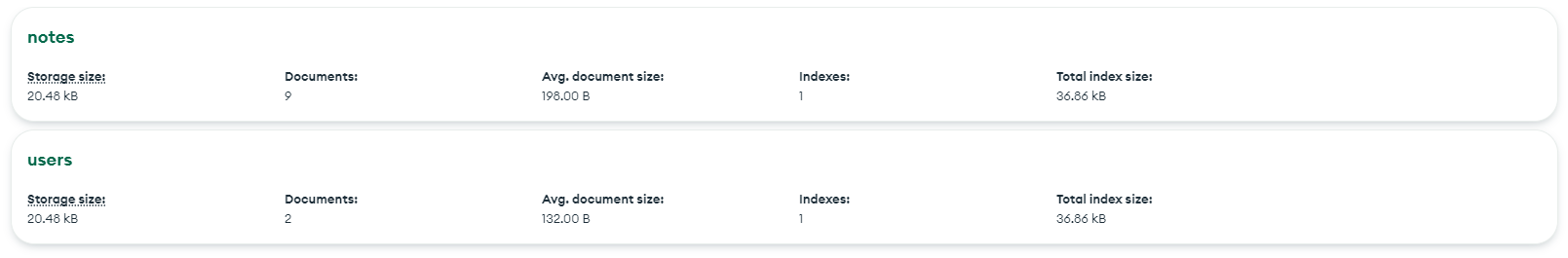
* + 1. Collection “Notes”

|  |  |  |
| --- | --- | --- |
| Trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | ID duy nhất của user (tự động). |
| title | String | Tiêu đề của ghi chú. |
| content | String | Nội dung của ghi chú. |
| tags | String | Các tags của ghi chú. |
| isPinned | Boolean | Cho biết ghi chú có được ghim hay không. |
| userId | ObjectId | ID của người dùng dã tạo ghi chú. |
| createOn | DateTime | Ngày giờ tạo ghi chú. |

* 1. Kiến trúc hệ thống

Hệ thống được thiết kế theo kiến trúc Client-Server với các thành phần chính:

* + 1. Client
* Gửi yêu cầu HTTP đến server (ví dụ: yêu cầu tạo ghi chú mới, đọc ghi chú).
  + 1. Backend (Server)
* Xử lý các yêu cầu từ client và tương tác với cơ sở dữ liệu MongoDB.
* Các API chính bao gồm:
  + POST /create-account: Tạo tài khoản mới.
  + POST /login: Đăng nhập.
  + GET /get-user: Lấy thông tin người dùng.
  + POST /add-notes: Tạo ghi chú mới.
  + PUT /edit-note/:noteId : Cập nhật ghi chú.
  + GET /get-all-notes: Lấy danh sách ghi chú.
  + DELETE /delete-note/:noteId: Xóa ghi chú.
  + PUT /update-note-pinned/:noteId: Cập nhật ghim ghi chú.
  + GET /search-notes: Tìm kiếm ghi chú.
    1. Database (MongoDB)
* Lưu trữ các ghi chú dưới dạng document JSON.



* + 1. Kết quả thiết kế
* Cơ sở dữ liệu MongoDB đã được thiết kế đơn giản và hiệu quả, phù hợp cho hệ thống ghi chú.
* API CRUD đã được triển khai đầy đủ và hoạt động tốt.
* Hệ thống đảm bảo tốc độ truy vấn nhanh và có khả năng mở rộng trong tương lai.

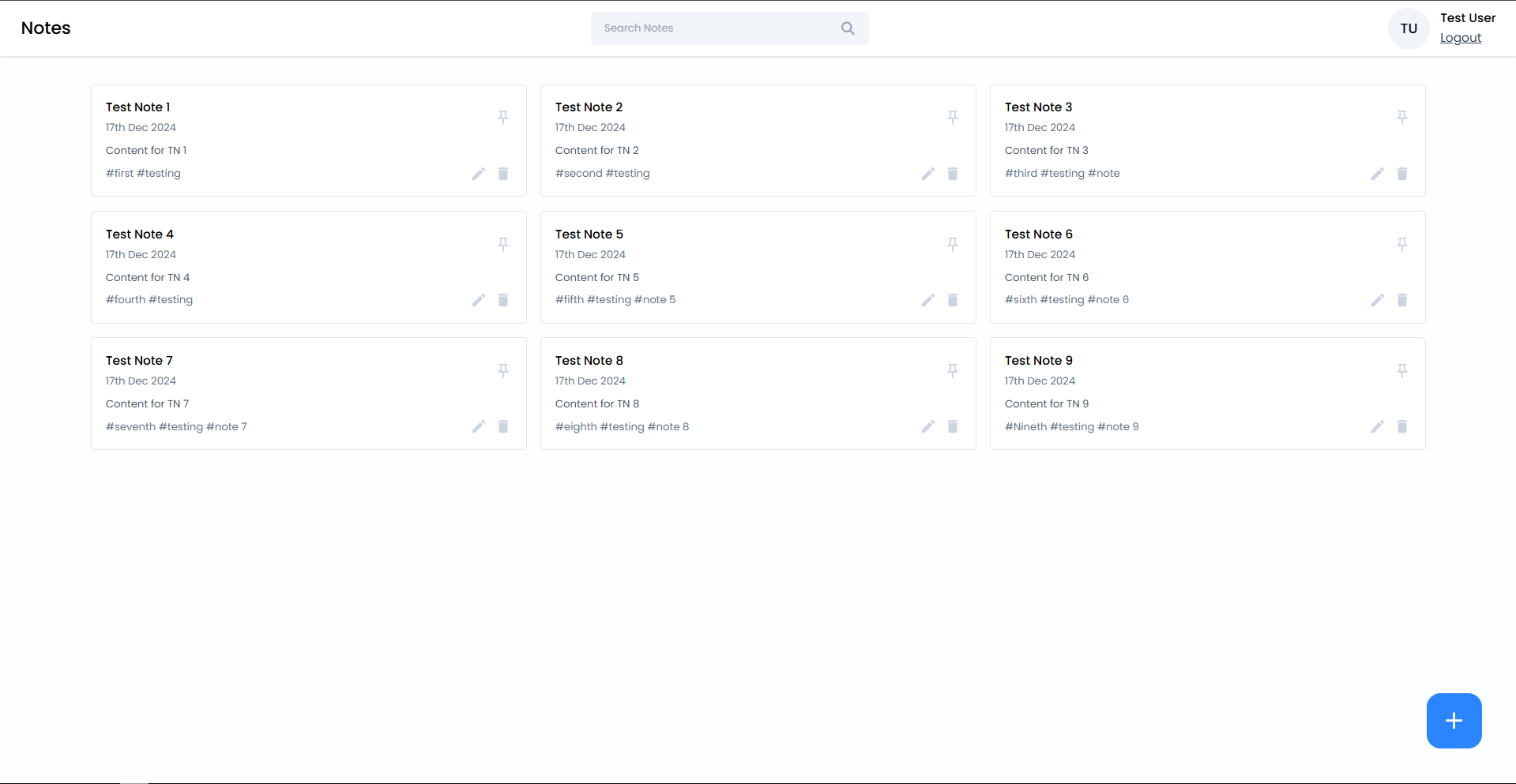
1. Ứng dụng NoSQL trên hệ thống website quản lý ghi chú
   1. Mô tả chức năng chính của hệ thống

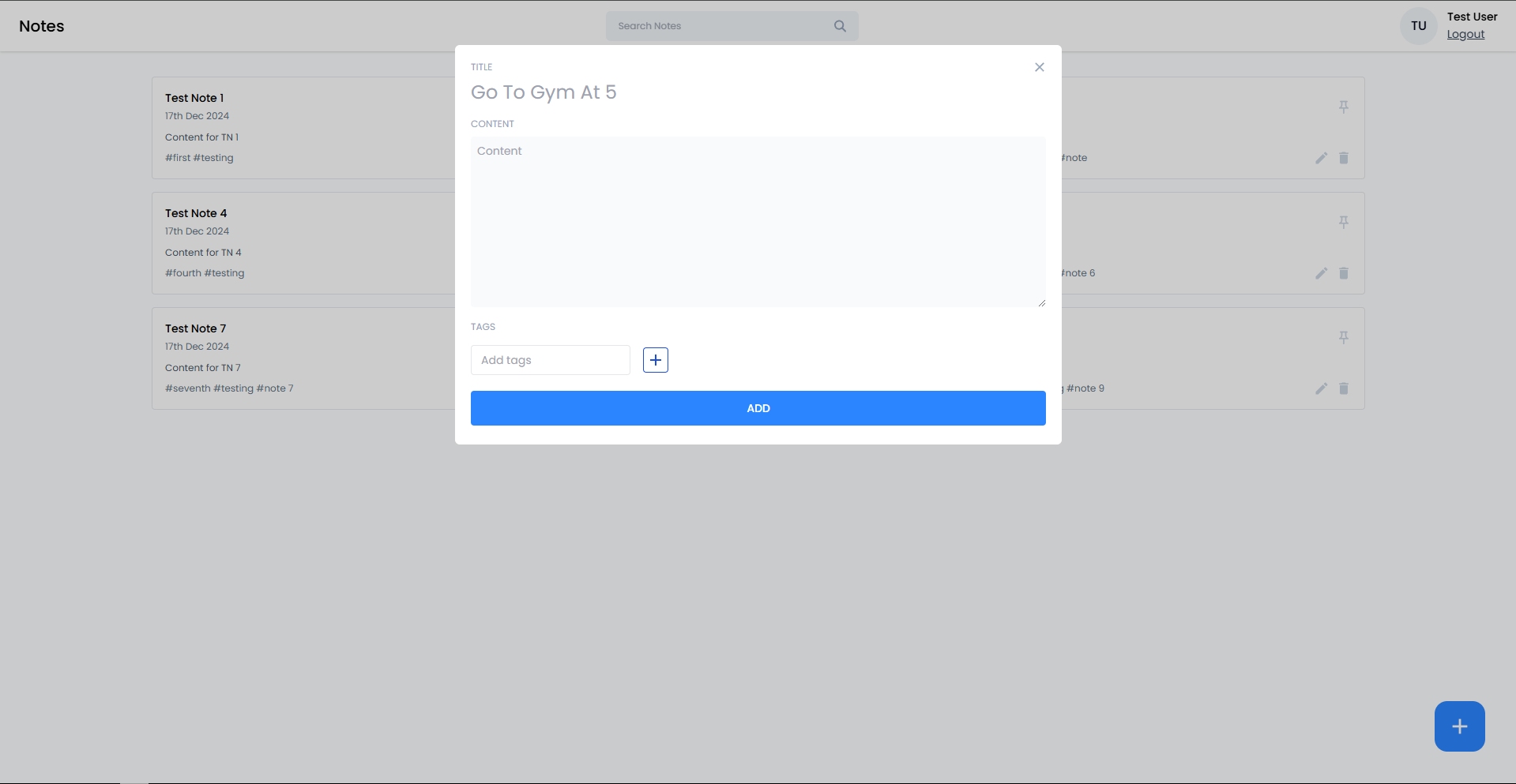
Hệ thống website ghi chú đơn giản cung cấp các chức năng cơ bản sau:

* + 1. Đăng ký tài khoản
* Người dùng có thể dễ dàng tạo tài khoản để truy cập vào website.
  + 1. Đăng nhập
* Người dùng đã tạo tài khoản có thể trực tiếp đăng nhập vào website.
  + 1. Xem danh sách ghi chú
* Hiển thị tất cả các ghi chú đã tạo.
  + 1. Tạo ghi chú mới
* Người dùng có thể tạo ghi chú với tiêu đề, nội dung và các tag.
  + 1. Chỉnh sửa ghi chú
* Người dùng có thể cập nhật nội dung, tiêu đề và các tag của ghi chú.
  + 1. Xóa ghi chú đã tạo
* Xóa ghi chú không cần thiết.
  + 1. Ghim ghi chú cần chú ý
* Ghim các ghi chú quan trọng.
  + 1. Tìm kiếm ghi chú
* Dễ dàng tìm kiếm các ghi chú.
  1. Triển khai hệ thống
     1. Cài đặt môi trường

Hệ thống được triển khai trên môi trường local với các công nghệ sau:

* Backend: Node.js và Express.js.
* Frontend: ReactJS.
* Database: MongoDB (cài đặt MongoDB local hoặc sử dụng MongoDB Atlas trên cloud).
* Công cụ hỗ trợ: Postman (kiểm thử API).
  + 1. Cài đặt hệ thống
* Cài đặt Node.js, ReactJS và MongoDB.
* Khởi tạo project:
  + Frontend:
    - npm create vite@latest notes-app
    - npm install -D tailwindcss postcss autoprefixer
    - npx tailwindcss init
    - npm install react-router-dom
  + Backend:
    - npm init
    - npm i express mongoose jsonwebtoken dotenv cors nodemon axios moment
* Kết nối MongoDB cho backend: sử dụng thư viện Mongose.
* Triển khai các API CRUD:
  + POST /create-account: Tạo tài khoản mới.
  + POST /login: Đăng nhập.
  + GET /get-user: Lấy thông tin người dùng.
  + POST /add-notes: Tạo ghi chú mới.
  + PUT /edit-note/:noteId : Cập nhật ghi chú.
  + GET /get-all-notes: Lấy danh sách ghi chú.
  + DELETE /delete-note/:noteId: Xóa ghi chú.
  + PUT /update-note-pinned/:noteId: Cập nhật ghim ghi chú.
  + GET /search-notes: Tìm kiếm ghi chú.
    1. Kiểm thử hệ thống
* Kiểm thử hệ thống bằng Postman hoặc các công cụ tương tự.
  1. Kết quả đạt được
     1. Nội dung
* Lưu trữ dữ liệu linh hoạt: MongoDB cho phép lưu trữ các ghi chú dưới dạng JSON, không cần schema cố định.
* Hiệu suất tốt: Truy vấn dữ liệu nhanh và đơn giản nhờ MongoDB.
* Hệ thống hoạt động ổn định: Các API CRUD hoạt động chính xác và hiệu quả.
  + 1. Hình ảnh





1. Kết luận và hướng phát triển
   1. Kết luận
      1. Các chức năng đã đạt được

Hệ thống website ghi chú đơn giản đã được xây dựng thành công và đáp ứng đầy đủ các chức năng cơ bản:

* Tạo, xem, cập nhật, xóa, ghim và tìm kiếm ghi chú một cách nhanh chóng và dễ dàng.
* Dữ liệu được lưu trữ linh hoạt bằng MongoDB – một hệ quản trị cơ sở dữ liệu NoSQL mạnh mẽ.
* Hệ thống hoạt động ổn định, đảm bảo hiệu suất tốt và dễ dàng mở rộng trong tương lai.
  + 1. Lợi ích đạt được
* Lưu trữ linh hoạt: Sử dụng MongoDB giúp lưu trữ dữ liệu dưới dạng document JSON mà không cần schema cố định.
* Quản lý ghi chú hiệu quả: Thao tác CRUD được triển khai đầy đủ và chính xác.
* Dễ dàng mở rộng: Kiến trúc hệ thống cho phép nâng cấp và bổ sung tính năng trong tương lai mà không ảnh hưởng đến dữ liệu cũ.
  + 1. Đánh giá tổng thể

Hệ thống đã minh chứng được tính hiệu quả và tính phù hợp của NoSQL (MongoDB) trong việc xây dựng các ứng dụng đơn giản, yêu cầu tính linh hoạt cao và tốc độ truy xuất nhanh.

* 1. Hướng phát triển

Để mở rộng và nâng cao hệ thống trong tương lai, có thể thực hiện các cải tiến sau:

* + 1. Bổ sung tính năng nâng cao
* Ghi chú có nhắc nhở (reminder): Cho phép người dùng đặt thời gian nhắc nhở cho ghi chú.
  + 1. Cải thiện trải nghiệm người dùng
* Tối ưu hiển thị danh sách ghi chú khi số lượng ghi chú lớn.
  + 1. Tích hợp tính năng đồng bộ hóa
* Đồng bộ ghi chú giữa nhiều thiết bị bằng cách tích hợp các dịch vụ lưu trữ đám mây (như Google Drive, Dropbox).
  + 1. Tối ưu hệ thống
* Tối ưu cơ sở dữ liệu MongoDB bằng cách sử dụng indexing và phân vùng dữ liệu (sharding) để cải thiện hiệu suất.
  + 1. Triển khai trên môi trường thực tế
* Triển khai hệ thống lên các nền tảng cloud như Heroku, AWS, hoặc DigitalOcean để hệ thống hoạt động ổn định và phục vụ nhiều người dùng.
* Sử dụng MongoDB Atlas để quản lý cơ sở dữ liệu dễ dàng và an toàn hơn.

**BẢNG PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| STT | Thành viên | Công việc |
| 1 | Đoàn Huỳnh Ngọc Sơn | Xây dựng chức năng và giao diện: đăng ký, đăng nhập, thêm xóa sửa ghi chú, ghim ghi chú và tìm kiếm ghi chú. |