Tên: Trần Vũ Thanh Lâm

Mssv: 27211202238

**Báo cáo BTVN: Xây dựng ứng dụng web Node.js và React**

**Link github:** https://github.com/TranVuThanhLam/PerfomanceHomework.git

**Mục tiêu**

* Tạo tập dữ liệu giả lớn (20.000 dòng) với cấu trúc cụ thể.
* Lưu trữ dữ liệu vào cơ sở dữ liệu SQL Server.
* Xây dựng ứng dụng web sử dụng Node.js làm backend và React làm frontend.
* Kết nối ứng dụng với cơ sở dữ liệu và hiển thị dữ liệu lên giao diện.
* Sử dụng tính năng non-clustered indexing để tối ưu hóa hiệu suất truy vấn.
* Sử dụng kỹ thuật caching để tối ưu hóa hiệu suất truy vấn. (chưa làm được)
* Sử dụng kỹ thuật compressing để tối ưu hóa hiệu suất truy vấn. (chưa làm được)

**Thực hiện**

1. **Tạo dữ liệu:**
   * Sử dụng thư viện faker để tạo 20.000 dòng dữ liệu giả với các trường Title, Description, Content và ID.
2. **Lưu trữ dữ liệu:**
   * Kết nối với cơ sở dữ liệu SQL Server bằng Node.js và sử dụng thư viện mssql.
   * Tạo bảng để lưu trữ dữ liệu và thêm index cho trường Title và Description để tối ưu tốc độ truy vấn (trường ID là primary key nên có clustered index là mặc định)
3. **Xây dựng ứng dụng:**
   * Tạo hai thư mục riêng biệt cho backend (Node.js) và frontend (React).
   * **Backend:**
     + Sử dụng Express.js để tạo server.
     + Viết API để truy xuất dữ liệu từ cơ sở dữ liệu và trả về kết quả cho frontend.
     + Sử dụng Connection Pool: Việc sử dụng connection pool giúp tối ưu hóa việc kết nối đến cơ sở dữ liệu, tránh tình trạng quá tải khi có nhiều yêu cầu đồng thời.
     + Middleware: Middleware connectToDatabase giúp đảm bảo rằng mỗi yêu cầu đều có một kết nối riêng đến database, tăng tính độc lập và bảo mật.
     + Xử lý lỗi: Code đã có cơ log lỗi giúp cho ứng dụng chạy ổn định hơn.
   * **Frontend:**
     + Sử dụng React để xây dựng giao diện người dùng.
     + Gọi API từ backend để lấy dữ liệu và hiển thị trên trang web.
     + **NewsList Component (Danh Sách Tin Tức):**
       - **Lấy dữ liệu tin tức (Lấy dữ liệu từ API):**
       - Sử dụng useState để quản lý dữ liệu tin tức (news), trạng thái tải (loading), và trạng thái lỗi (error).
       - Sử dụng useEffect với mảng phụ thuộc trống ([]) để lấy dữ liệu tin tức khi component được khởi tạo.
       - Thực hiện API call đến /api/news bằng axios.get.
       - Cập nhật state news với dữ liệu lấy được trong khối try.
       - Đặt loading thành false và xử lý các lỗi tiềm ẩn trong khối finally.
       - Sử dụng console.time và console.timeEnd để đo thời gian tải danh sách tin tức.
     + **Tìm kiếm (Chức năng tìm kiếm):**
       - Nhận prop onSearch từ component SearchBar (Thanh tìm kiếm).
       - Định nghĩa hàm handleSearch để xử lý cập nhật thuật ngữ tìm kiếm.
       - Nếu thuật ngữ tìm kiếm rỗng, thiết lập lại state news về dữ liệu gốc bằng setNews(originalNews).
       - Ngược lại, lọc dữ liệu originalNews dựa trên thuật ngữ tìm kiếm trong các trường Title và Description (không phân biệt chữ hoa chữ thường).
       - Cập nhật state news với các kết quả được lọc.
4. **Kết nối và hiển thị dữ liệu:**
   * Sử dụng thư viện fetch hoặc axios để gọi API từ frontend.
   * Hiển thị dữ liệu nhận được trong các component React.

**Kết quả**

* Đã xây dựng thành công một ứng dụng web đơn giản nhưng đầy đủ chức năng.
* Dữ liệu được lưu trữ và truy xuất hiệu quả từ cơ sở dữ liệu SQL Server.
* Tính năng indexing đã được sử dụng để cải thiện hiệu suất truy vấn.