**BÀI TẬP THỰC HÀNH**

**LƯU TRỮ VÀ XỬ LÍ DỮ LIỆU TRONG MÔI TRƯỜNG ĐIỆN TOÁN ĐÁM MÂY**

# 1. Mục tiêu

* Trình bày đươc Hệ thống lưu trữ phân tán và đồng nhất bộ nhớ NFS, AFS
* Trình bày được Hệ thống lưu trữ HDFS, GFS
* Nhận biết được các loại và sử dụng được cơ sở NoSQL trong để truy vấn dữ liệu
* Đối sánh giữa NoSQL và RDBMS về cấu trúc, thành phần và cách truy vấn dữ liệu.
* Sử dụng được một trong những loại CSDL NoSQL.
* Tư duy logic, tư duy hệ thống trong phân tích thiết kế cơ sở dữ liệu.

# 2. Nội dung

## Câu 1: So sánh kỹ thuật lưu trữ và xử lí dữ liệu của cơ sỏ dữ liệu NoSQL và hệ quản trị cơ sở dữ liệu truyền thống.

1. **Cơ sở dữ liệu NoSQL:**

**Lưu trữ dữ liệu:** Dữ liệu không yêu cầu cấu trúc cố định và có thể lưu trữ ở dạng tài liệu (document), cột (column), key-value, đồ (graph), hoặc chuỗi thời gian.

Phù hợp cho dữ liệu không cố định, phức tạp, và linh hoạt.

**Xử lí dữ liệu:** Khả năng mở rộng dễ dàng, cho phép thêm các nút máy chủ một cách linh hoạt để tăng khả năng đối phó với lưu lượng dữ liệu lớn.

Thường sử dụng các phương pháp lưu trữ dữ liệu phi cấu trúc và cơ chế tìm kiếm, truy vấn dựa trên các tùy chỉnh riêng biệt.

**Hiệu năng:** NoSQL thường được thiết kế để đạt được hiệu suất cao ở các trường hợp sử dụng cụ thể, như lưu trữ và truy vấn dữ liệu phân tán, dữ liệu chuỗi thời gian, và dữ liệu đồ.

**Quyền truy cập và bảo mật:** NoSQL thường cung cấp quản lý quyền truy cập và bảo mật mạnh mẽ, nhưng cách thức thực hiện có thể khác nhau tùy theo hệ thống cụ thể.

1. **Hệ quản trị cơ sở dữ liệu truyền thống (RDBMS):**

**Lưu trữ dữ liệu:** Dữ liệu phải tuân theo mô hình quan hệ với các bảng, khóa ngoại, và ràng buộc cơ sở dữ liệu.

Phù hợp cho dữ liệu có cấu trúc và mối quan hệ phức tạp.

**Xử lí dữ liệu:** Cơ sở dữ liệu truyền thống sử dụng SQL (Structured Query Language) để truy vấn dữ liệu và cung cấp tính năng giao dịch (transaction) để đảm bảo tính toàn vẹn dữ liệu.

Khó mở rộng, thường cần phải thực hiện phân chia dữ liệu và dự án kết hợp (join) dữ liệu.

**Hiệu năng:** RDBMS thường thiết kế cho hiệu năng ổn định và đảm bảo tính toàn vẹn dữ liệu thông qua giao dịch.

Hiệu suất có thể trở nên kém khi cần xử lý lưu lượng dữ liệu lớn hoặc khi cần truy vấn đa dạng và phức tạp.

**Quyền truy cập và bảo mật:** RDBMS cung cấp cơ chế quản lý quyền truy cập và bảo mật mạnh mẽ bao gồm bảo mật cơ sở dữ liệu, bảo mật hàng ngang và hàng dọc.

## Câu 2: Tìm hiểu cách Tạo và quản trị database trên vertabelo?

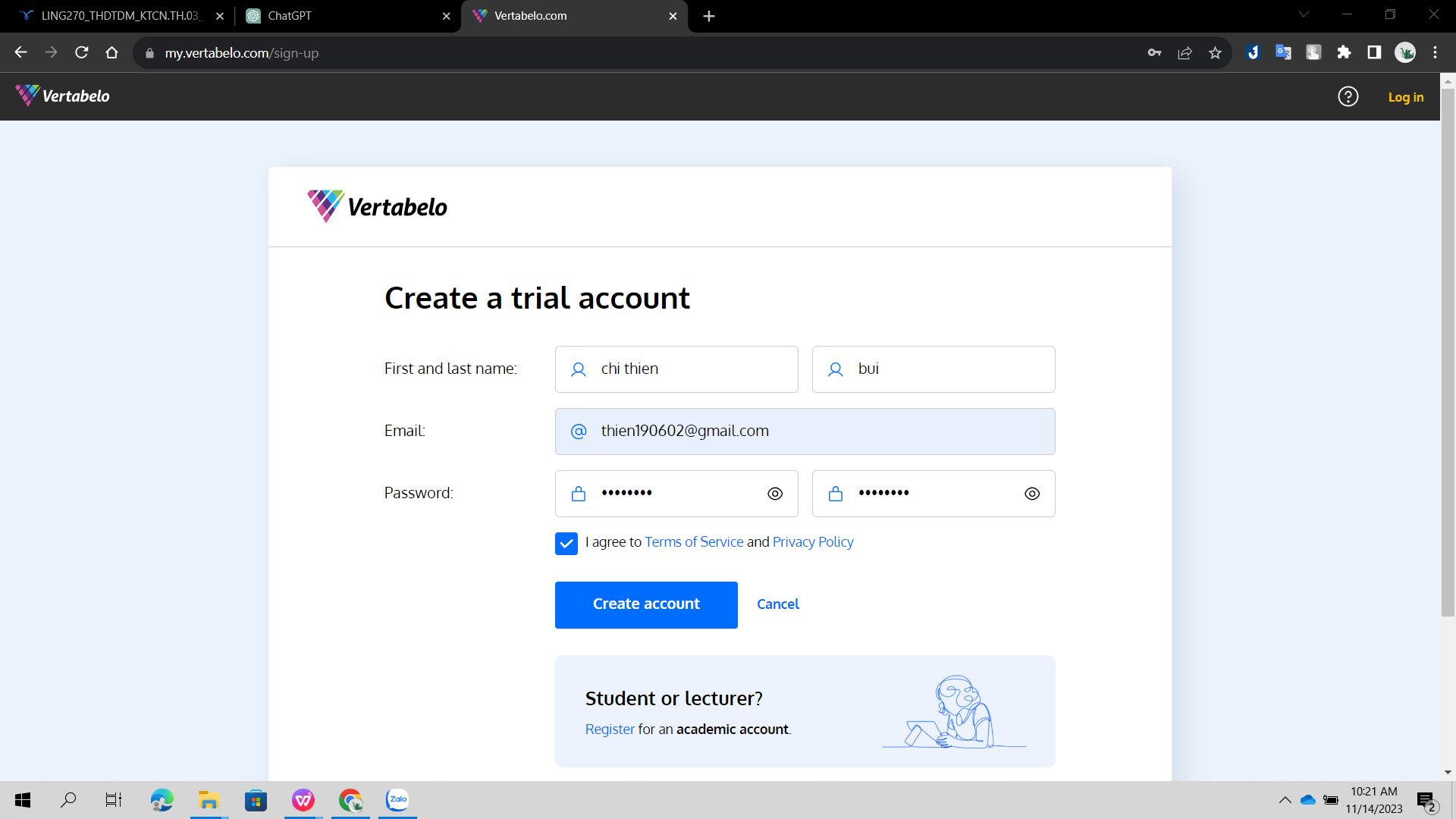
**Vertabelo** là một công cụ thiết kế cơ sở dữ liệu trực tuyến cho phép em tạo và quản lý cơ sở dữ liệu một cách dễ dàng. Vertabelo cung cấp một giao diện trực quan và dễ sử dụng để thiết kế và quản lý cơ sở dữ liệu. Nó giúp tối ưu hóa quy trình phát triển cơ sở dữ liệu và cải thiện hiệu suất làm việc của em.

**Dưới đây là hướng dẫn cách tạo:**

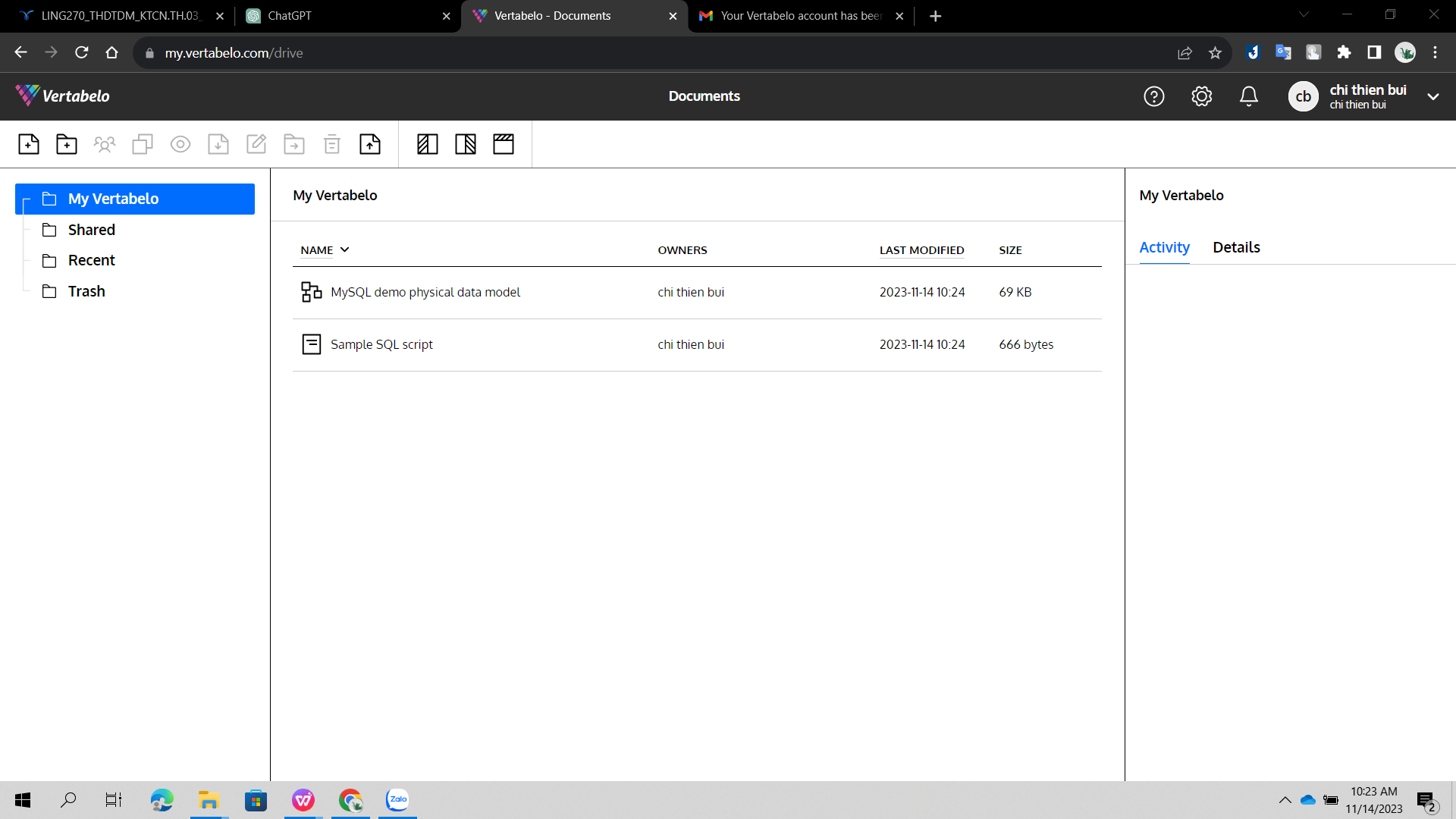
**Bước 1:** Đăng ký và đăng nhập:

Truy cập trang web của Vertabelo (https://www.vertabelo.com/).

Đăng ký tài khoản mới hoặc đăng nhập vào tài khoản đã có.



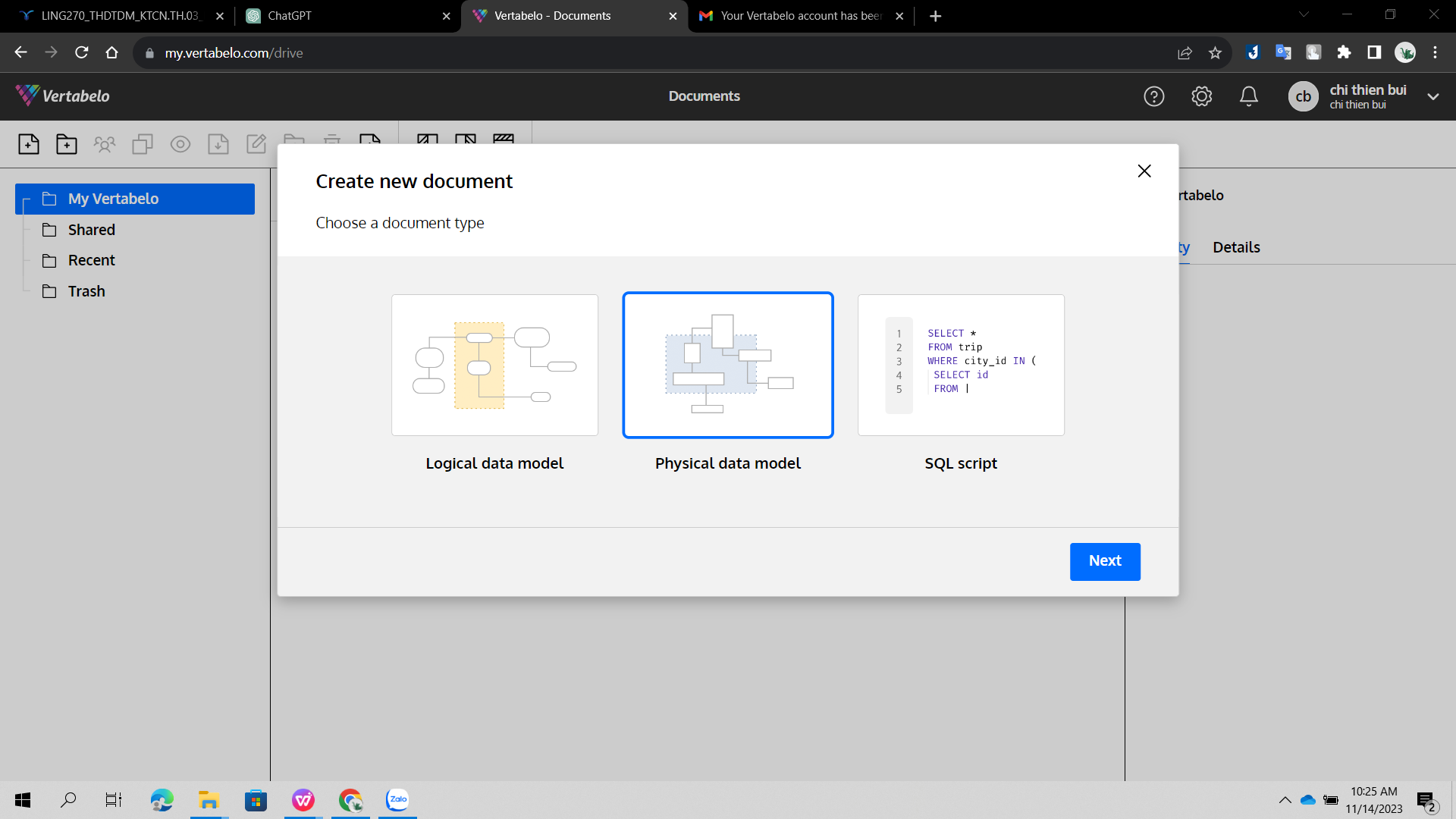
#### Tạo tài khoản Vertable



#### Giao diện của Vertabelo

**Bước 2:** Tạo project (dự án):

Sau khi đăng nhập, em sẽ thấy giao diện chính của Vertabelo. em có thể bắt đầu bằng cách tạo một dự án (project) mới. Nhấn vào nút "Create new project" hoặc tương tự.



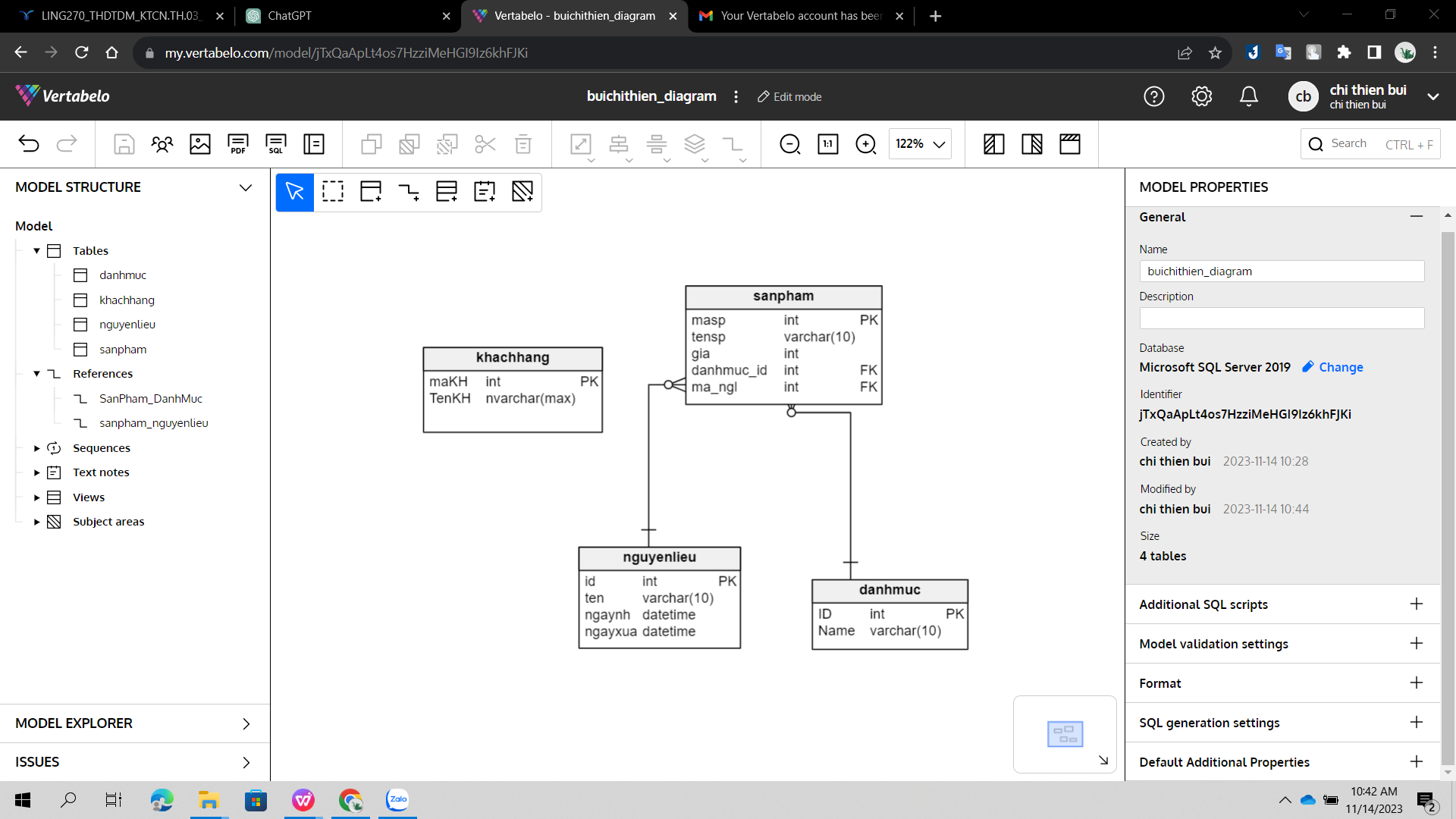
#### Tạo project dự án

**Bước 3:** Tạo sơ đồ ERD (Entity-Relationship Diagram):

Trong dự án của em, em có thể bắt đầu tạo sơ đồ ERD bằng cách thêm các thực thể (entities) và quan hệ (relationships) vào sơ đồ. Sử dụng các công cụ vẽ để tạo bảng và thiết lập quan hệ giữa chúng.

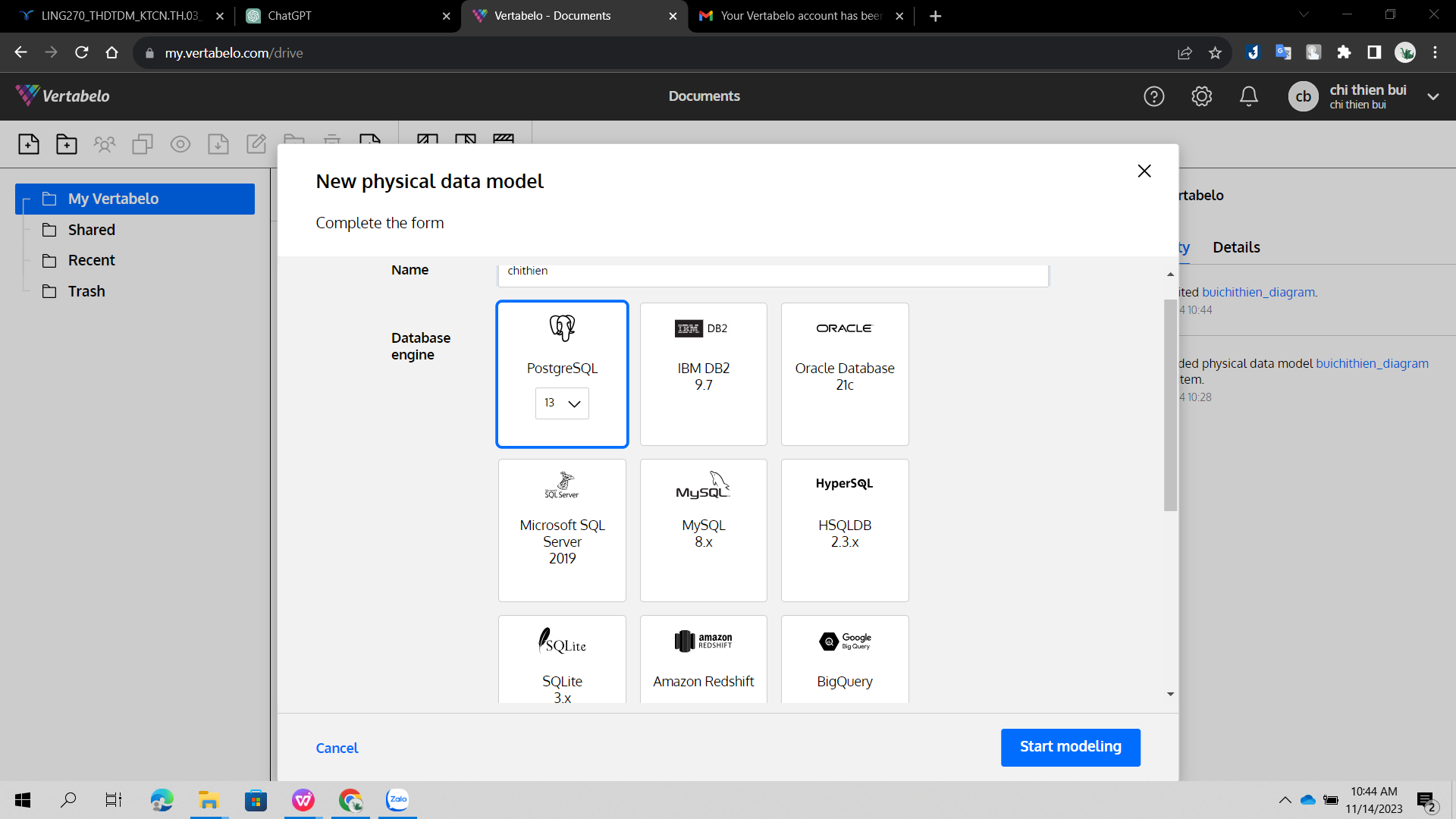
**Bước 4:** Thiết kế cơ sở dữ liệu:

Sau khi có sơ đồ ERD hoàn chỉnh, em có thể bắt đầu thiết kế cơ sở dữ liệu bằng cách ánh xạ các thực thể và quan hệ thành bảng và khóa ngoại trong cơ sở dữ liệu.



**Bước 5:** Synchronize cơ sở dữ liệu với hệ quản trị cơ sở dữ liệu:

Vertabelo hỗ trợ nhiều hệ quản trị cơ sở dữ liệu, bao gồm PostgreSQL, MySQL, Oracle, SQL Server và nhiều loại khác. em có thể chọn hệ quản trị cơ sở dữ liệu mục tiêu trong menu và thực hiện việc đồng bộ hóa sơ đồ ERD với cơ sở dữ liệu thực tế.



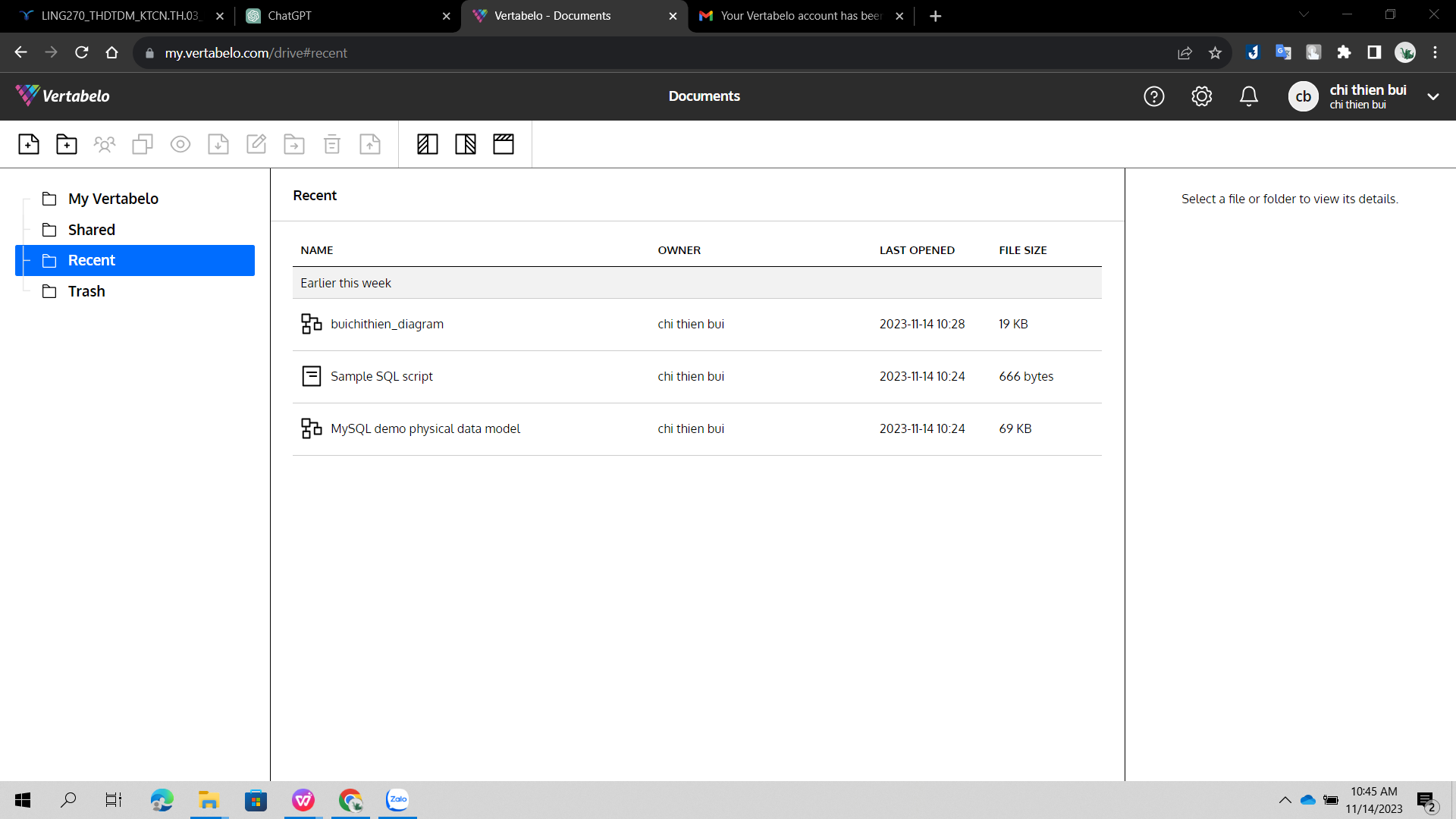
**Bước 6:** Quản lý cơ sở dữ liệu:

Trong dự án, em có thể quản lý cơ sở dữ liệu của mình, thực hiện thay đổi cấu trúc, tạo bảng mới, và thực hiện các hoạt động quản lý khác.

**Bước 7:** Xuất bản sơ đồ ERD và tài liệu:

em có thể xuất bản sơ đồ ERD và tạo tài liệu cho cơ sở dữ liệu của mình từ Vertabelo để chia sẻ với đội làm việc hoặc các bên liên quan.

**Quản lý database trên Vertabelo:**



**Bước 1:** Đăng nhập vào Vertabelo:

Truy cập trang web của Vertabelo (https://www.vertabelo.com/).

Đăng nhập vào tài khoản Vertabelo của em.

**Bước 2:** Truy cập dự án cần quản lý:

Sau khi đăng nhập, em sẽ thấy danh sách các dự án (projects) của em. Nhấn vào dự án chứa sơ đồ cơ sở dữ liệu em muốn quản lý.

**Bước 3:** Sơ đồ ERD và cơ sở dữ liệu:

Trong dự án của em, em sẽ thấy sơ đồ ERD (Entity-Relationship Diagram) đã được thiết kế. Sơ đồ ERD là biểu đồ thể hiện cấu trúc cơ sở dữ liệu của em.

**Bước 4:** Sửa đổi cơ sở dữ liệu:

Để sửa đổi cơ sở dữ liệu, em có thể thêm, chỉnh sửa hoặc xóa các bảng, cột, khóa ngoại và quan hệ trong sơ đồ ERD.

Để thêm mới, nhấn vào biểu tượng "Thêm bảng" hoặc "Thêm quan hệ" trên thanh công cụ và thực hiện thay đổi cần thiết.

Để chỉnh sửa, nhấn vào phần cần sửa đổi và thay đổi thông tin liên quan.

Để xóa, nhấn vào phần cần xóa và chọn tùy chọn "Xóa."

**Bước 5:** Đồng bộ hóa với hệ quản trị cơ sở dữ liệu:

Vertabelo hỗ trợ nhiều hệ quản trị cơ sở dữ liệu, bao gồm PostgreSQL, MySQL, Oracle, SQL Server và nhiều loại khác. em có thể chọn hệ quản trị cơ sở dữ liệu mục tiêu từ menu.

Sau đó, em có thể thực hiện việc đồng bộ hóa sơ đồ ERD với cơ sở dữ liệu thực tế. Các thay đổi trong sơ đồ sẽ được ánh xạ và áp dụng vào cơ sở dữ liệu của em.

**Bước 6:** Quản lý cơ sở dữ liệu:

em có thể quản lý cơ sở dữ liệu của mình bằng cách thêm bảng, chỉnh sửa cấu trúc cơ sở dữ liệu, thực hiện truy vấn, và quản lý dữ liệu thực tế trên hệ quản trị cơ sở dữ liệu mục tiêu.

**Bước 7:** Xuất bản sơ đồ ERD và tài liệu:

em có thể xuất bản sơ đồ ERD và tạo tài liệu cho cơ sở dữ liệu của mình từ Vertabelo để chia sẻ với đội làm việc hoặc các bên liên quan.

## **Câu 3**: Tìm hiểu các đặc điểm lưu trữ của hệ thống HDFS, những ưu nhược điểm của mô hình MapReduce, so sánh MapReduce với mô hình thực thi MPI. Những xu thế tính toán và cải tiến của MapReduce.

1. **Tìm hiểu các đặc điểm lưu trữ của hệ thống HDFS**

* HDFS là hệ thống tệp tin phân tán được sử dụng trong hệ thống Hadoop, phù hợp cho việc lưu trữ và xử lý dữ liệu lớn. Dưới đây là các đặc điểm lưu trữ của HDFS:
* Phân tán: HDFS lưu trữ dữ liệu trên nhiều nút máy chủ phân tán, giúp tối ưu hóa hiệu suất và đảm bảo độ tin cậy.
* Sao lưu dữ liệu: Dữ liệu trong HDFS được sao lưu tự động thành nhiều bản để đảm bảo tính toàn vẹn và sẵn sàng khi có lỗi.
* Khả năng mở rộng: HDFS có khả năng mở rộng dễ dàng bằng cách thêm nút máy chủ vào hệ thống.
* Lý thuyết về dữ liệu lớn: HDFS thiết kế để xử lý lượng dữ liệu lớn và phục vụ cho mô hình xử lý dữ liệu lớn như MapReduce.

1. **Những ưu nhược điểm của mô hình MapReduce**

**Ưu điểm của mô hình MapReduce:**

* **Tính dễ sử dụng:** MapReduce cung cấp một mô hình lập trình cấp cao, giúp người phát triển tập trung vào xử lý dữ liệu thay vì quản lý các khía cạnh phức tạp của tính toán phân tán.
* **Khả năng mở rộng:** MapReduce có khả năng mở rộng tốt. Bạn có thể thêm nhiều nút máy chủ vào hệ thống để xử lý lượng dữ liệu lớn một cách hiệu quả.
* **Tự động phân chia công việc:** MapReduce tự động phân chia công việc xử lý thành các nhiệm vụ nhỏ và phân phối chúng trên các nút máy chủ. Điều này giúp tối ưu hóa việc quản lý tài nguyên.
* **Tích hợp với Hadoop và hệ thống cơ sở dữ liệu phân tán:** MapReduce dễ dàng tích hợp với các công cụ và hệ thống phân tán như Hadoop, Hive và HBase.
* **Tính toàn vẹn dữ liệu:** MapReduce hỗ trợ tính toàn vẹn dữ liệu thông qua việc sao lưu dữ liệu và thực hiện nhiều lần các pha Map và Reduce.

**Nhược điểm của mô hình MapReduce:**

* **Không phù hợp cho các tác vụ không phải là tính toán phân tán:** MapReduce chủ yếu phù hợp cho các tác vụ mà có thể phân chia thành các pha Map và Reduce. Điều này làm cho nó không phù hợp cho các tác vụ tương tác và truy vấn phức tạp.
* **Khả năng mở rộng giới hạn:** Mặc dù MapReduce có khả năng mở rộng tốt, nhưng không phải tất cả các ứng dụng đều dễ dàng có thể tận dụng được tính toàn vẹn và khả năng mở rộng của nó.
* **Hiệu suất yếu cho các truy vấn tương tác:** MapReduce không phù hợp cho các tác vụ yêu cầu truy vấn tương tác và thời gian thực. Thời gian phản hồi có thể bị kéo dài trong các tình huống như vậy.
* **Phải viết mã phức tạp cho một số tác vụ:** Một số tác vụ cần viết mã phức tạp và đòi hỏi hiểu biết sâu rộng về MapReduce, điều này có thể làm tăng độ phức tạp trong việc phát triển ứng dụng.
* **Cần nắm rõ cấu trúc dữ liệu:** Để tận dụng tối đa MapReduce, bạn cần phải nắm rõ cấu trúc dữ liệu của dữ liệu đầu vào và quy trình xử lý.

1. **So sánh MapReduce với MPI (Message Passing Interface):**

**MapReduce:**

* **Mô hình lập trình cấp cao:** MapReduce là một mô hình lập trình cấp cao giúp đơn giản hóa quá trình xử lý dữ liệu song song trên nhiều nút máy chủ. Nó tập trung vào việc phân chia dữ liệu và xử lý kết quả.
* **Xử lý dữ liệu lớn:** MapReduce thích hợp cho xử lý dữ liệu lớn và phân tán. Nó được sử dụng rộng rãi trong các ứng dụng Big Data và phân tích dữ liệu lớn.
* **Hiệu suất ổn định:** MapReduce được thiết kế để đảm bảo hiệu suất ổn định và khả năng mở rộng khi xử lý dữ liệu lớn.
* **Tự động quản lý tài nguyên:** MapReduce quản lý tài nguyên tự động, bao gồm việc phân chia nhiệm vụ và giao tiếp giữa các nút máy chủ.

**MPI (Message Passing Interface):**

* **Giao diện lập trình mức thấp:** MPI là một giao diện lập trình mức thấp hơn, cho phép kiểm soát tường tận việc quản lý dữ liệu và giao tiếp giữa các quy trình (processes). Điều này làm cho nó mạnh mẽ và linh hoạt, nhưng cũng đòi hỏi nhiều công sức trong việc lập trình và quản lý tài nguyên.
* **Tính toán khoa học và mức thấp:** MPI thường được sử dụng trong tính toán khoa học và tính toán mức thấp, nơi việc kiểm soát giao tiếp và tài nguyên là quan trọng.
* **Tương tác trực tiếp:** MPI cho phép tương tác trực tiếp giữa các quy trình, chẳng hạn như trao đổi dữ liệu và thông điệp. Điều này phù hợp cho các tác vụ có sự giao tiếp phức tạp.

**Những xu thế tính toán và cải tiến của MapReduce.**

MapReduce là một mô hình xử lý dữ liệu lớn đã tạo ra sự cách mạng trong ngành công nghiệp dữ liệu lớn và tính toán phân tán. Mặc dù đã có nhiều sự cải tiến và phát triển, nhưng MapReduce vẫn đóng vai trò quan trọng trong cơ sở dữ liệu lớn và tính toán phân tán. Dưới đây là một số xu hướng và cải tiến của MapReduce:

* **Hệ sinh thái Hadoop và Apache Spark:** Hadoop là một hệ sinh thái mở rộng xung quanh MapReduce, bao gồm HDFS (Hadoop Distributed File System) và nhiều dự án phụ khác. Apache Spark, một dự án đáng chú ý trong hệ sinh thái này, đã nhanh chóng trở thành một giải pháp thay thế cho MapReduce. Spark cung cấp hiệu suất cao hơn và hỗ trợ nhiều kiểu dữ liệu và ứng dụng hơn.
* **Cải tiến hiệu năng:** Các cải tiến trong tối ưu hóa hiệu năng đã được thực hiện trong MapReduce và các dự án liên quan. Các cải tiến này bao gồm tối ưu hóa thời gian phân phối và đồng bộ hóa, tối ưu hóa lưu trữ tạm thời, và sử dụng bộ nhớ đệm hiệu quả hơn.
* **Tích hợp với trí tuệ nhân tạo và học máy:** MapReduce và các dự án liên quan đã bắt đầu tích hợp với trí tuệ nhân tạo và học máy để xử lý và phân tích dữ liệu lớn. Điều này cho phép việc huấn luyện mô hình học máy và phát triển ứng dụng trí tuệ nhân tạo trên nền tảng MapReduce.
* **Tích hợp tính toán dưới dạng dịch vụ đám mây:** Các dịch vụ đám mây như Amazon Web Services (AWS) và Microsoft Azure đã cung cấp dịch vụ tính toán dựa trên MapReduce, giúp tối ưu hóa việc quản lý tài nguyên và cung cấp tài nguyên tính toán linh hoạt theo nhu cầu.
* **Hệ thống quản lý tài nguyên:** Các hệ thống quản lý tài nguyên như Apache YARN (Yet Another Resource Negotiator) đã được phát triển để tối ưu hóa việc quản lý tài nguyên và thực hiện tính toán phân tán hiệu quả.
* **Sự tích hợp với hệ thống cơ sở dữ liệu truyền thống:** Sự kết hợp giữa MapReduce và các hệ thống cơ sở dữ liệu truyền thống đã tạo ra giải pháp hoàn chỉnh cho việc lưu trữ và xử lý dữ liệu lớn.
* **Sự linh hoạt trong cách tiếp cận dữ liệu:** MapReduce đã mở ra nhiều cách tiếp cận dữ liệu khác nhau, bao gồm xử lý dữ liệu cấu trúc và không cấu trúc, dữ liệu văn bản, hình ảnh, âm thanh và video.

## **Câu 4**: Tìm hiểu các mô hình lưu trữ và xử lí dữ liệu của Amazon Web Services, OpenStack swift.

**Amazon Web Services (AWS):**

* **Amazon S3 (Simple Storage Service):** Amazon S3 là một dịch vụ lưu trữ đối tượng phân tán của AWS. Nó cho phép bạn lưu trữ và quản lý các đối tượng dữ liệu, bao gồm hình ảnh, tệp tin, video và nhiều loại dữ liệu khác. S3 sử dụng giao thức HTTP/HTTPS để tải lên và tải xuống dữ liệu.
* **Amazon EBS (Elastic Block Store):** Amazon EBS cung cấp lưu trữ đĩa dựa trên mạng cho máy ảo EC2 (Elastic Compute Cloud) của AWS. Nó cho phép bạn tạo và quản lý các ổ đĩa ảo để lưu trữ dữ liệu, và bạn có thể gắn/détach các ổ đĩa này từ các máy ảo EC2.
* **Amazon RDS (Relational Database Service):** Amazon RDS là dịch vụ quản lý cơ sở dữ liệu quan hệ phân tán. Nó hỗ trợ các hệ quản trị cơ sở dữ liệu quan hệ phổ biến như MySQL, PostgreSQL, Oracle và SQL Server.
* **Amazon Redshift:** Amazon Redshift là dịch vụ lưu trữ và xử lý dữ liệu phân tán cho việc phân tích dữ liệu lớn. Nó được thiết kế để thực hiện các truy vấn phức tạp trên dữ liệu lớn.

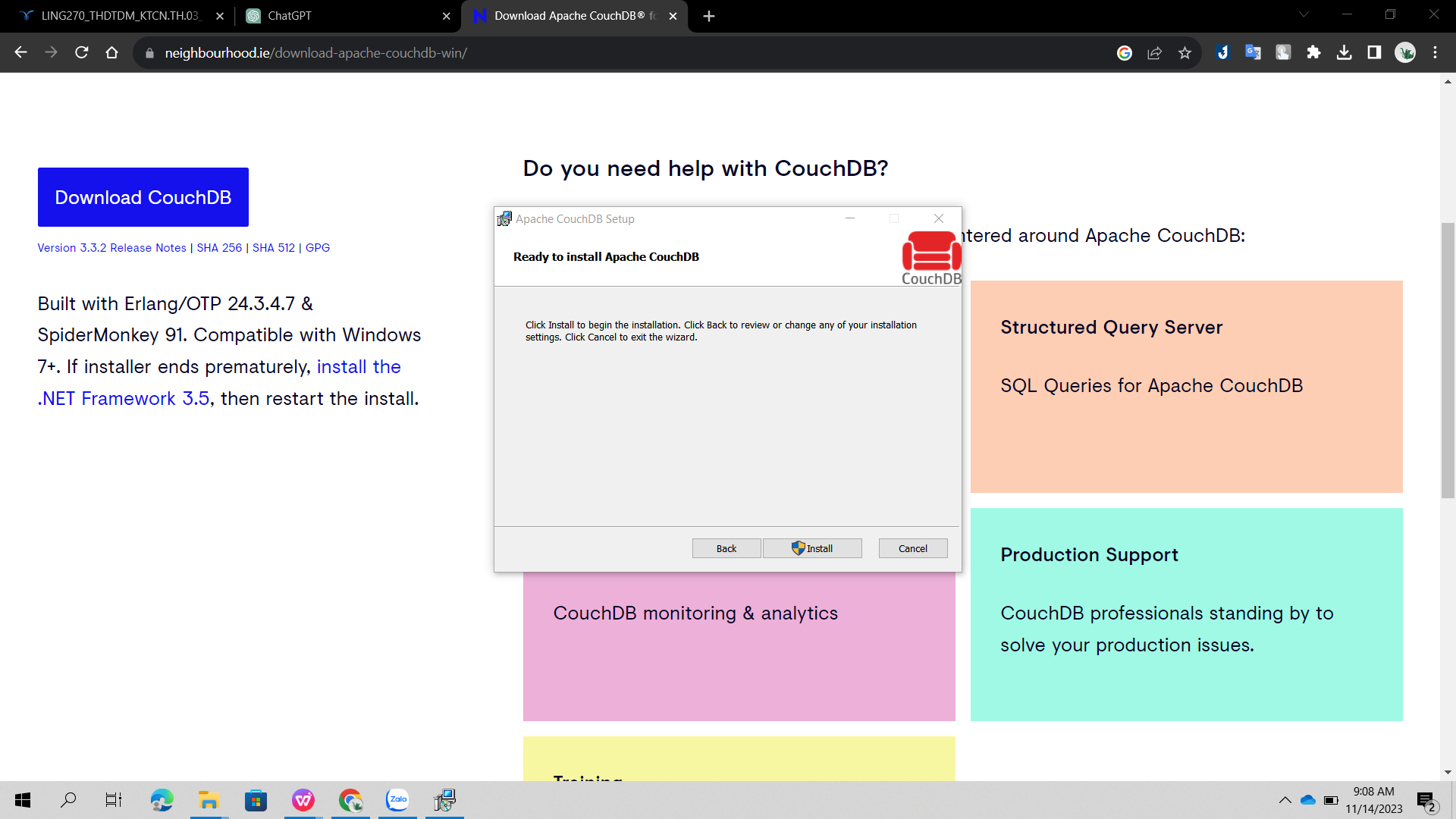
**OpenStack Swift:**

* **OpenStack Swift:** OpenStack Swift là một dự án mã nguồn mở của OpenStack dùng để xây dựng dịch vụ lưu trữ đối tượng phân tán. Nó cho phép bạn lưu trữ và quản lý các đối tượng dữ liệu trong một môi trường đám mây riêng của bạn. Swift hỗ trợ việc mở rộng và phân tán dữ liệu.
* **OpenStack Cinder:** OpenStack Cinder là dự án dùng để quản lý và cung cấp lưu trữ đĩa dựa trên mạng cho máy ảo trong môi trường đám mây OpenStack. Nó tương tự như Amazon EBS và cho phép bạn gắn/détach các ổ đĩa từ các máy ảo.
* **OpenStack Trove:** OpenStack Trove là dự án quản lý cơ sở dữ liệu phân tán trong môi trường đám mây OpenStack. Nó cung cấp khả năng triển khai và quản lý các cơ sở dữ liệu quan hệ và không quan hệ.
* **OpenStack Sahara:** OpenStack Sahara là dự án để triển khai và quản lý các hệ thống xử lý dữ liệu lớn như Apache Hadoop và Apache Spark trên môi trường đám mây OpenStack.

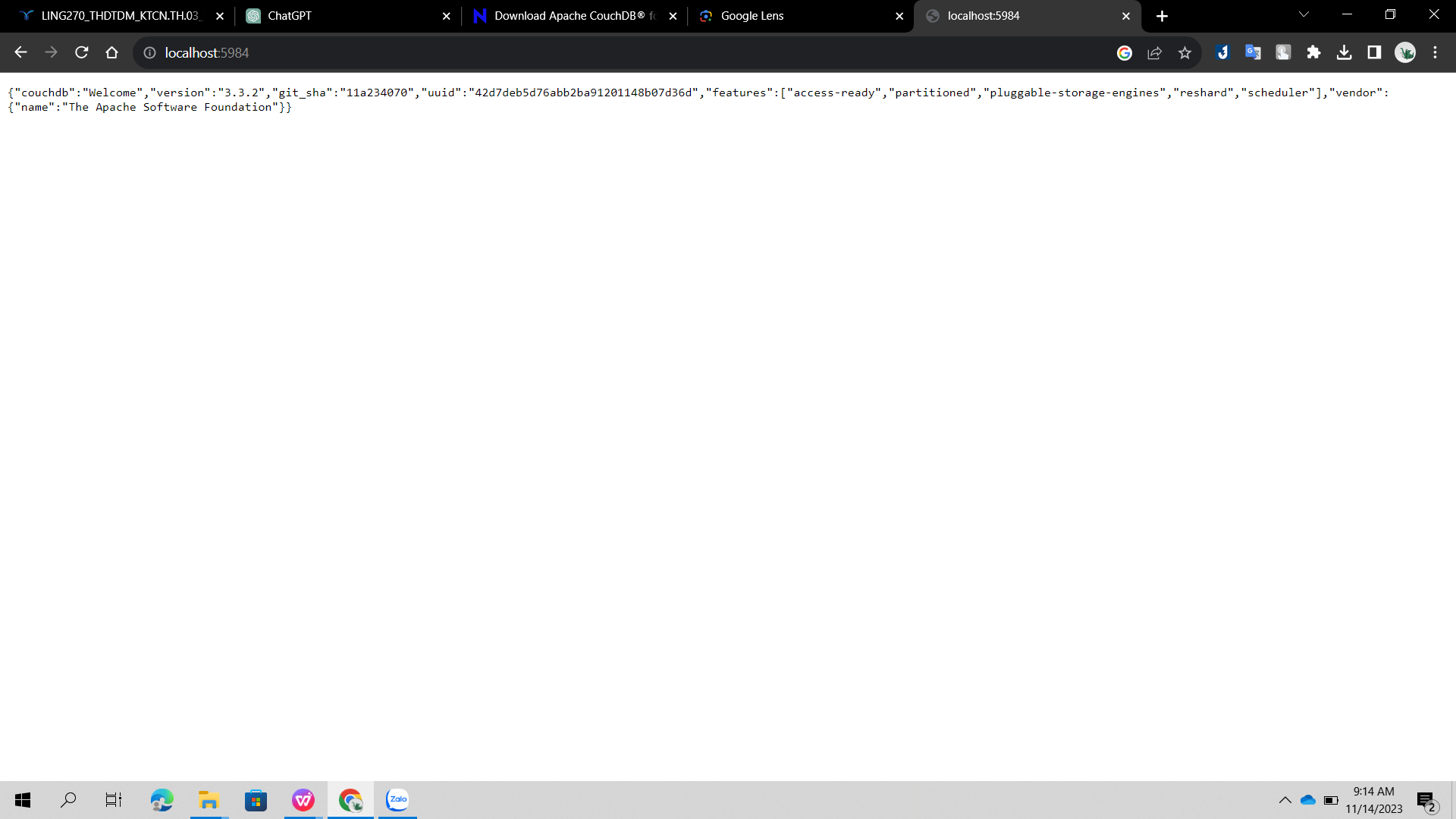
**THƯC HÀNH**

**TẠO BẢNG VÀ TRUY VẤN DỮ LIỆU TRÊN COUCHDB (N0SQL)**

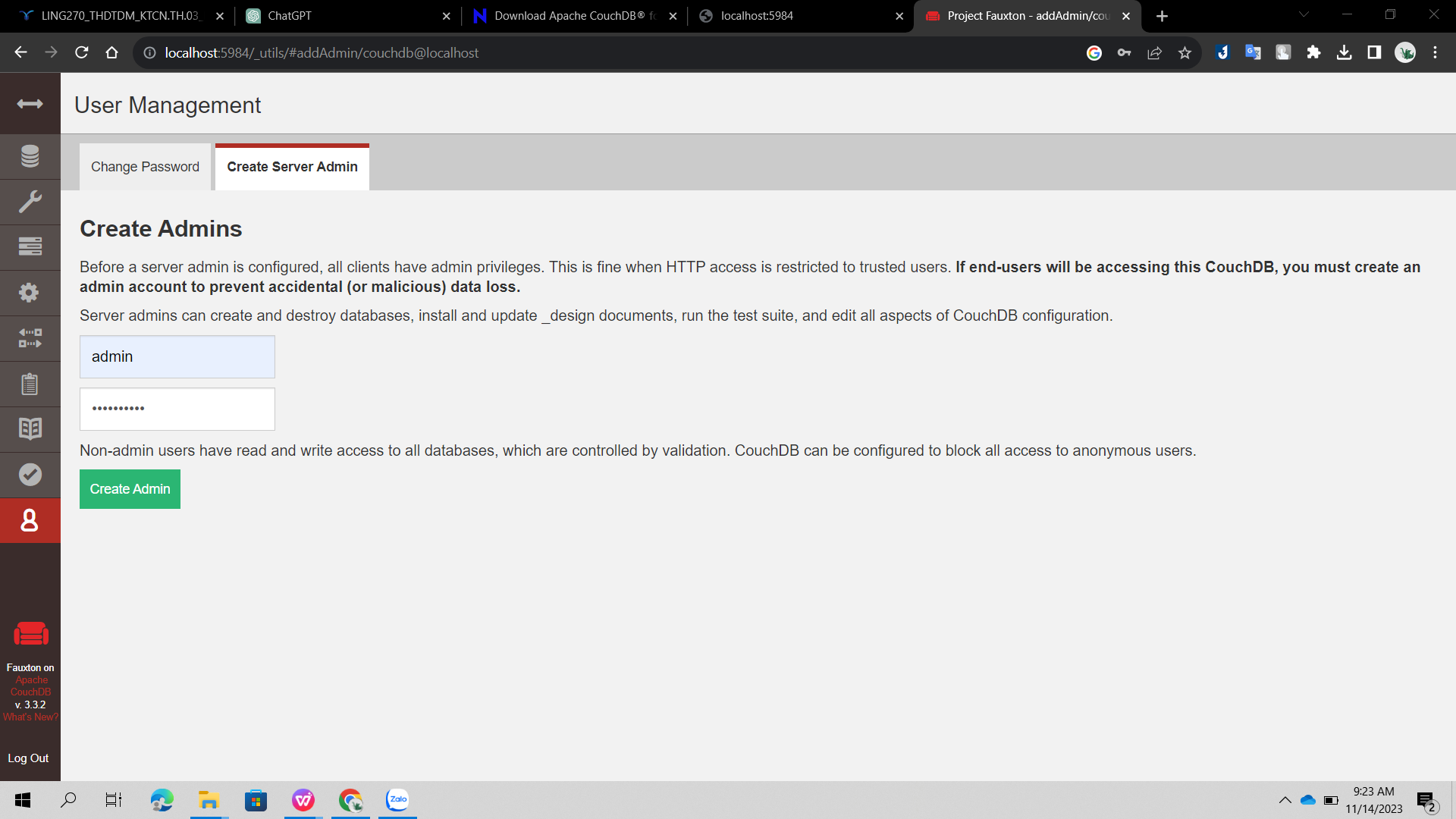
**Bước 1. Cài đặt CouchDB**



#### Cài đặt CouchDB

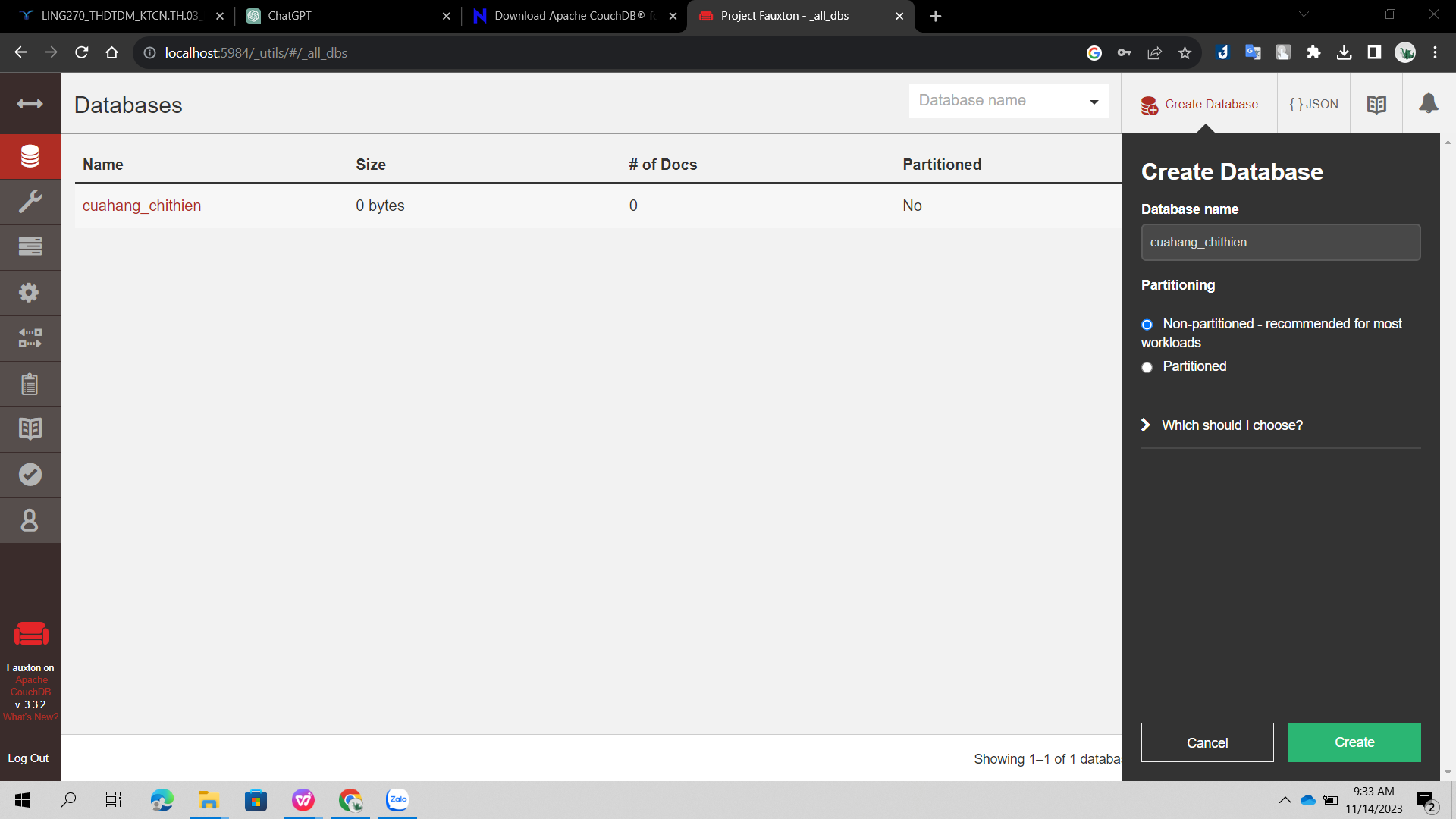


#### truy cập vào địa chỉ http://localhost:5984/ để kiểm tra xem CouchDB đã chạy thành công hay chưa

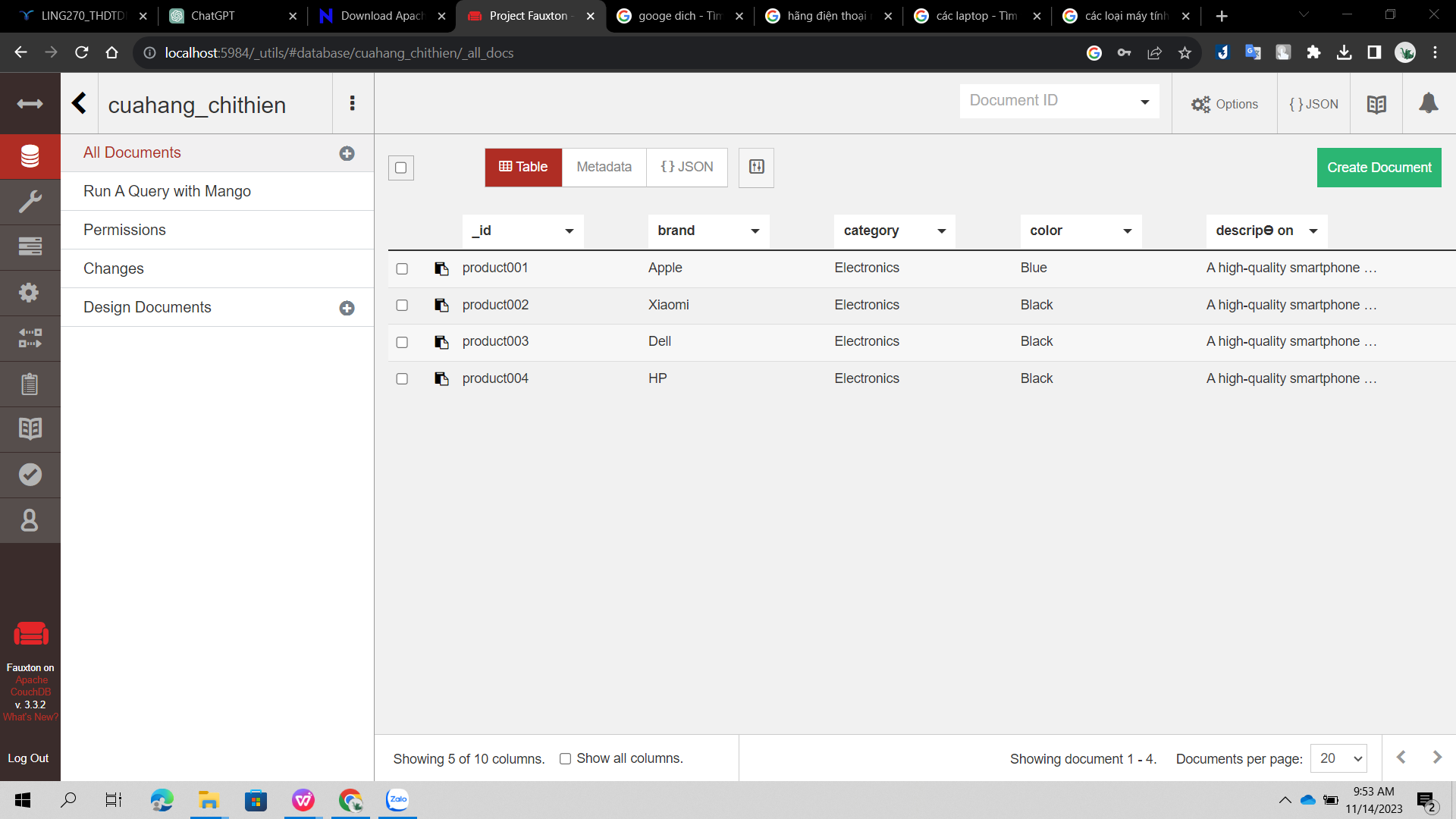


#### Đặt mật khẩu quản trị:admin/mk:admin123

**Bước 2: Tạo một bảng và truy vấn dữ liệu trên CouchDB:**



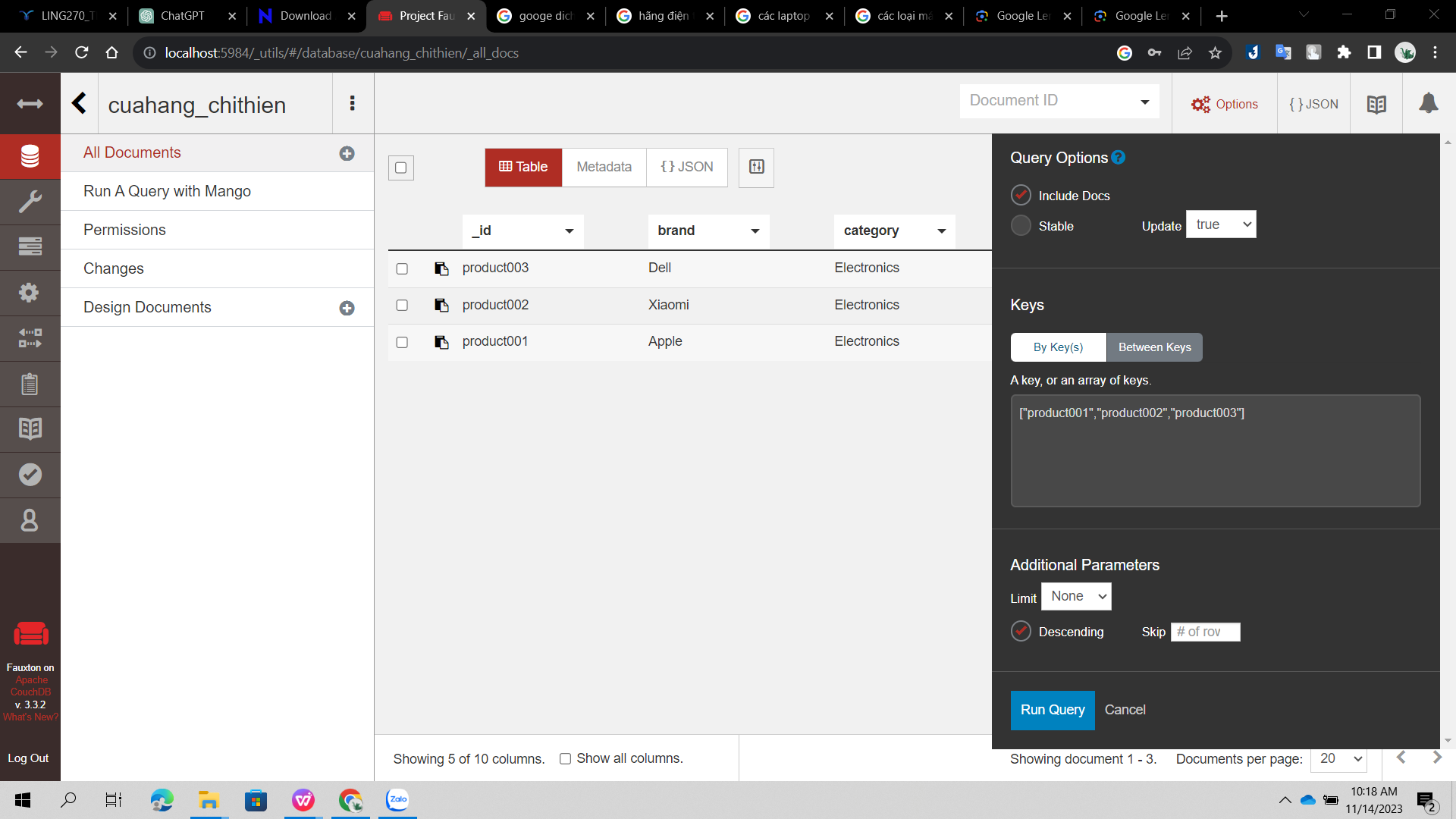
#### Tạo Database name: cuahang\_chithien



#### Tạo dữ liệu trên CouchDB

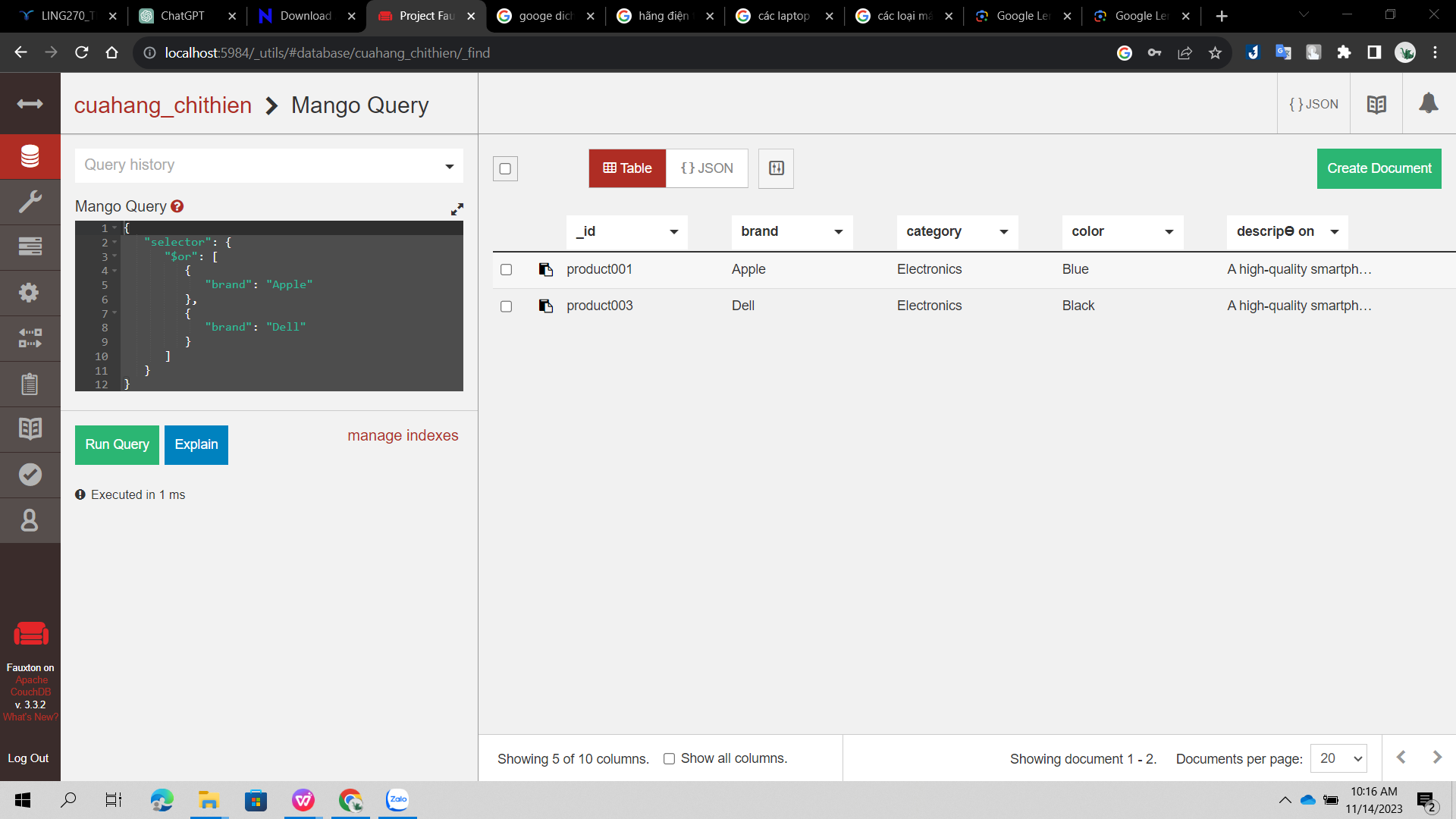
**Bước 4: Truy vấn dữ liệu trên CouchDB**

Truy vấn với Query Options



#### Chỉ lấy ra sản phẩm có key là product001, product002, product003

**Chạy truy vấn CouchDB với Mango Query.**



#### Lấy ra dữ liệu có hãng là Apple hoặc Dell

------------------------HẾT-----------------------

*Lưu ý : Sinh viên làm trực tiếp trên file này, đổi tên file thành MaSV\_HoTen và gửi lênE-Learning?*