

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HCM  
KHOA CÔNG NGHỆ THÔNG TIN

MÔN HỌC: CƠ SỞ DỮ LIỆU NÂNG CAO



**BÁO CÁO ĐỒ ÁN**

ĐỀ TÀI

**TÌM HIỂU CSDL NOSQL, CÀI ĐẶT MONGODB  
THIẾT KẾ CSDL MONGODB VÀ THAO TÁC**

GVHD: ThS. Lê Thị Minh Châu

Mã lớp: ADDB331784\_23\_2\_02

Nhóm sinh viên thực hiện: Nhóm 1

1. Nguyễn Thị Lan Anh – 21110368
2. Huỳnh Nhật Nam – 21110903
3. Trịnh Thị Thanh Huyền – 21110477
4. Phan Minh Thuận – 21110667

TP. Hồ Chí Minh, tháng 04 năm 2024

## LỜI CẢM ƠN

Lời nói đầu tiên, chúng em xin gửi lời biết ơn sâu sắc và lòng tri ân đến Thạc sĩ Lê Thị Minh Châu - người đã dành thời gian và công sức để hỗ trợ, đồng hành cùng chúng em trong suốt quá trình nghiên cứu và học tập đề tài này. Từ khi chúng em bắt đầu bước vào lĩnh vực học tập này đến hiện tại, sự quan tâm và sự hỗ trợ tận tình từ Cô đã giúp chúng em phát triển và hoàn thành đề tài của mình một cách thành công.

Chúng em xin bày tỏ lòng biết ơn chân thành nhất đến Cô Lê Thị Minh Châu - người đã chia sẻ những kiến thức quý báu và cung cấp sự hướng dẫn cần thiết cho chúng em trong suốt quá trình học tập. Nhờ vào sự chỉ dạy và hướng dẫn tận tình của Cô mà chúng em đã có thể hoàn thiện đề tài nghiên cứu của mình.

Bài báo cáo về đề tài “TÌM HIỂU CSDL NOSQL, CÀI ĐẶT MONGODB, THIẾT KẾ CSDL MONGODB VÀ THAO TÁC” của môn "Cơ Sở Dữ Liệu Nâng Cao" đã được chúng em thực hiện trong vòng một tháng. Tuy chúng em còn hạn chế trong lượng kiến thức, nhưng chúng em hy vọng nhận được sự đóng góp quý báu từ Cô để bổ sung và hoàn thiện kiến thức của mình, từ đó học hỏi thêm kinh nghiệm và nâng cao trình độ, chuẩn bị cho tương lai công việc.

Chúng em xin gửi lời cảm ơn chân thành nhất đến Cô, và chúc Cô luôn vui vẻ, giữ lửa nhiệt huyết quyết tâm dành cho nghề, và luôn tràn đầy niềm tin trong cuộc sống. Chúng em hy vọng mọi điều tốt lành sẽ đến với Cô, và Cô sẽ tiếp tục đóng góp cho sự nghiệp trồng người với sự cao cả và vĩ đại như Cô đã từng làm.

## BẢNG PHÂN CÔNG

<b>STT</b>	<b>MSSV</b>	<b>Họ Và Tên</b>	<b>Nhiệm Vụ</b>	<b>Hoàn Thành</b>
1	21110903	Huỳnh Nhật Nam (Nhóm Trưởng)	<p>Soạn nội dung phần III: Làm việc với MongoDB(cài đặt và cấu hình MongoDB, tạo lập cơ sở dữ liệu)</p> <p>Các thao tác CRUD cơ bản</p> <p>Code front end và back end các chức năng:</p> <ul style="list-style-type: none"> <li>- Tài khoản (đăng nhập đăng ký , đăng xuất cập nhật thông tin tài khoản)</li> <li>- Đơn hàng trong quản lý hệ thống (xem đơn hàng, xem báo cáo đơn hàng)</li> <li>- Đơn hàng của người dùng(đặt hàng, hủy đơn hàng)</li> </ul>	100%
2	21110368	Nguyễn Thị Lan Anh	<p>Soạn nội dung phần III: Làm việc với MongoDB(Thao tác với cơ sở dữ liệu )</p> <p>Các thao tác CRUD cơ bản</p> <p>Code front end và back end các chức năng: Trang chủ (xem sản phẩm, tìm kiếm, xem chi tiết sản phẩm, xem sản phẩm, xem thêm, xem sản phẩm theo loại)</p>	100%
3	21110477	Trịnh Thị Thanh Huyền	Soạn nội dung phần I: Giới thiệu NoSQL.	100%

			Các thao tác CRUD cơ bản Code front end và back end các chức năng: Sản phẩm trong quản lý hệ thống( thêm sản phẩm, tìm kiếm sản phẩm, chỉnh sửa sản phẩm, xóa sản phẩm, xem list thông tin sản phẩm)	
4	21110667	Phan Minh Thuận	Soạn nội dung phần II: Cơ sở dữ liệu MongoDB.  Các thao tác CRUD cơ bản Code front end và back end các chức năng: Người dùng trong quản lý hệ thống (xóa thông tin, chỉnh sửa thông tin, tìm kiếm người dùng, xem list thông tin người dùng)	100%

## MỤC LỤC

<b>CHƯƠNG 1: GIỚI THIỆU NOSQL .....</b>	1
1.1. Giới thiệu .....	1
1.2. Đặc điểm của NoSQL .....	1
1.3. Phân loại cơ sở dữ liệu NoSQL .....	2
1.3.1. Key-value .....	2
1.3.2. Document .....	2
1.3.3. Graph .....	3
1.3.4. Column .....	4
1.4. So sánh cơ sở dữ liệu NoSQL với cơ sở dữ liệu quan hệ .....	4
<b>CHƯƠNG 2: CƠ SỞ DỮ LIỆU MONGODB .....</b>	6
2.1. Giới thiệu .....	6
2.2. Mô hình dữ liệu trong MongoDB .....	6
2.3. Kiến trúc của MongoDB .....	8
2.4. Các chế độ thực thi MongoDB .....	9
<b>CHƯƠNG 3: LÀM VIỆC VỚI MONGODB.....</b>	10
3.1. Cài đặt và cấu hình MongoDB .....	10
3.1.1. Cài đặt .....	10
3.1.2. Cấu hình .....	10
3.2. Tạo lập cơ sở dữ liệu .....	19
3.3. Thao tác với dữ liệu .....	19
3.3.1. Truy vấn dữ liệu .....	19
3.3.2. Thêm dữ liệu .....	21
3.3.3. Cập nhật dữ liệu .....	22
3.3.4. Xóa dữ liệu .....	25
<b>CHƯƠNG 4: LÀM VIỆC VỚI MONGODB.....</b>	27
4.1. Mô tả bài toán .....	27
4.2. Mô tả cơ sở dữ liệu, tập dữ liệu (nếu có) .....	27
4.3. Các chức năng chính của ứng dụng .....	28
4.4. Hướng dẫn cài đặt .....	29
4.5. Kết quả .....	29



## CHƯƠNG 1: GIỚI THIỆU NOSQL

### 1.1. Giới thiệu

NoSQL là một loại hệ thống quản lý cơ sở dữ liệu khác với cơ sở dữ liệu quan hệ truyền thống ở chỗ nó không sử dụng ngôn ngữ truy vấn có cấu trúc (SQL) để thao tác dữ liệu. Thay vào đó, cơ sở dữ liệu NoSQL sử dụng các cơ sở dữ liệu phi quan hệ có thể xử lý lượng lớn dữ liệu phi cấu trúc và bán cấu trúc, chẳng hạn như documents, graphs, và key-value pairs...

Cơ sở dữ liệu NoSQL ban đầu được phát triển để giải quyết các hạn chế về khả năng mở rộng và hiệu suất của cơ sở dữ liệu quan hệ truyền thống, vốn gặp khó khăn trong việc xử lý lượng dữ liệu khổng lồ được tạo ra bởi các ứng dụng web hiện đại và các hệ thống sử dụng nhiều dữ liệu khác. Cơ sở dữ liệu NoSQL được thiết kế để có khả năng mở rộng cao, linh hoạt và có khả năng chịu lỗi cao, khiến chúng trở nên lý tưởng để sử dụng trong các hệ thống phân tán quy mô lớn và môi trường điện toán đám mây.

### 1.2. Đặc điểm của NoSQL

**Tính linh hoạt của lược đồ:** Cơ sở dữ liệu NoSQL được thiết kế để xử lý dữ liệu phi cấu trúc và bán cấu trúc, chúng không yêu cầu lược đồ cố định như cơ sở dữ liệu quan hệ. Điều này giúp việc thêm dữ liệu mới vào cơ sở dữ liệu dễ dàng hơn mà không cần phải sửa đổi lược đồ.

**Khả năng mở rộng:** Cơ sở dữ liệu NoSQL có khả năng mở rộng cao, chúng có thể xử lý khối lượng dữ liệu lớn và lưu lượng truy cập cao. Nhiều cơ sở dữ liệu NoSQL được thiết kế để chạy trên nhiều nút hoặc máy chủ, giúp chúng mở rộng quy mô theo chiều ngang.

**Hiệu suất cao:** Cơ sở dữ liệu NoSQL được tối ưu hóa về hiệu suất, thường sử dụng kiến trúc phân tán và lưu trữ trong bộ nhớ để đạt được tốc độ đọc và ghi nhanh.

**Tính sẵn sàng:** Cơ sở dữ liệu NoSQL được thiết kế để có tính sẵn sàng cao, người dùng luôn có thể truy cập chúng ngay cả trong trường hợp xảy ra lỗi phần cứng hoặc mạng.

**Sao chép:** Nhiều cơ sở dữ liệu NoSQL hỗ trợ sao chép dữ liệu, cho phép chúng phân phối dữ liệu trên nhiều nút hoặc máy chủ để tăng tính khả dụng và khả năng chịu lỗi.

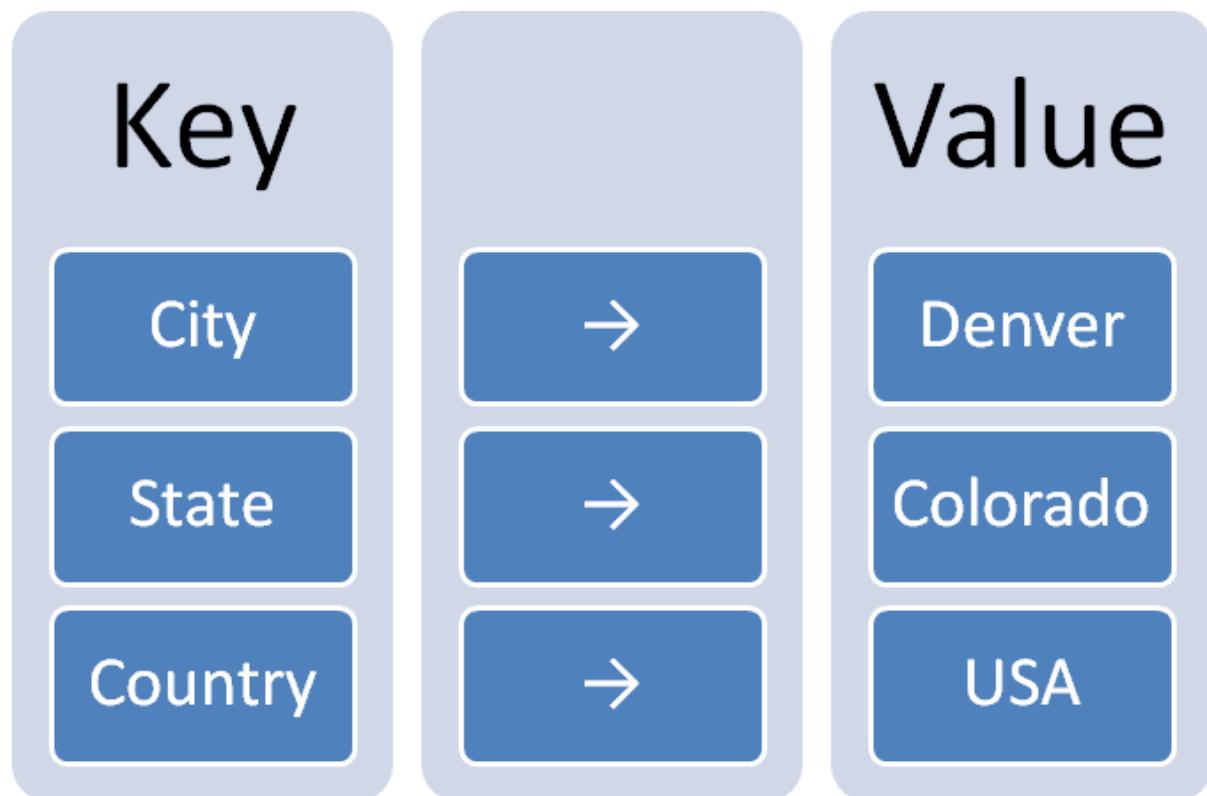
**Phân mảnh:** Cơ sở dữ liệu NoSQL thường hỗ trợ phân mảnh, giúp chúng phân vùng dữ liệu trên nhiều máy chủ hoặc nút để cải thiện khả năng mở rộng và hiệu suất.

**Mô hình dữ liệu NoSQL:** Cơ sở dữ liệu NoSQL sử dụng nhiều mô hình dữ liệu phi quan hệ, chẳng hạn như key-value, document, column-family, và graph.. Mỗi mô hình này đều có điểm mạnh và điểm yếu riêng, giúp cơ sở dữ liệu NoSQL có tính linh hoạt cao và thích ứng với nhiều trường hợp sử dụng.

### 1.3. Phân loại cơ sở dữ liệu NoSQL

#### 1.3.1. Key-value

Là một loại cơ sở dữ liệu lưu trữ dữ liệu dưới dạng tập hợp các cặp key-value. Trong kiểu lưu trữ dữ liệu này, mỗi mục dữ liệu được xác định bằng một khóa duy nhất và giá trị được liên kết với khóa đó có thể là bất kỳ thứ gì, chẳng hạn như chuỗi, số, đối tượng hoặc thậm chí là cấu trúc dữ liệu khác.



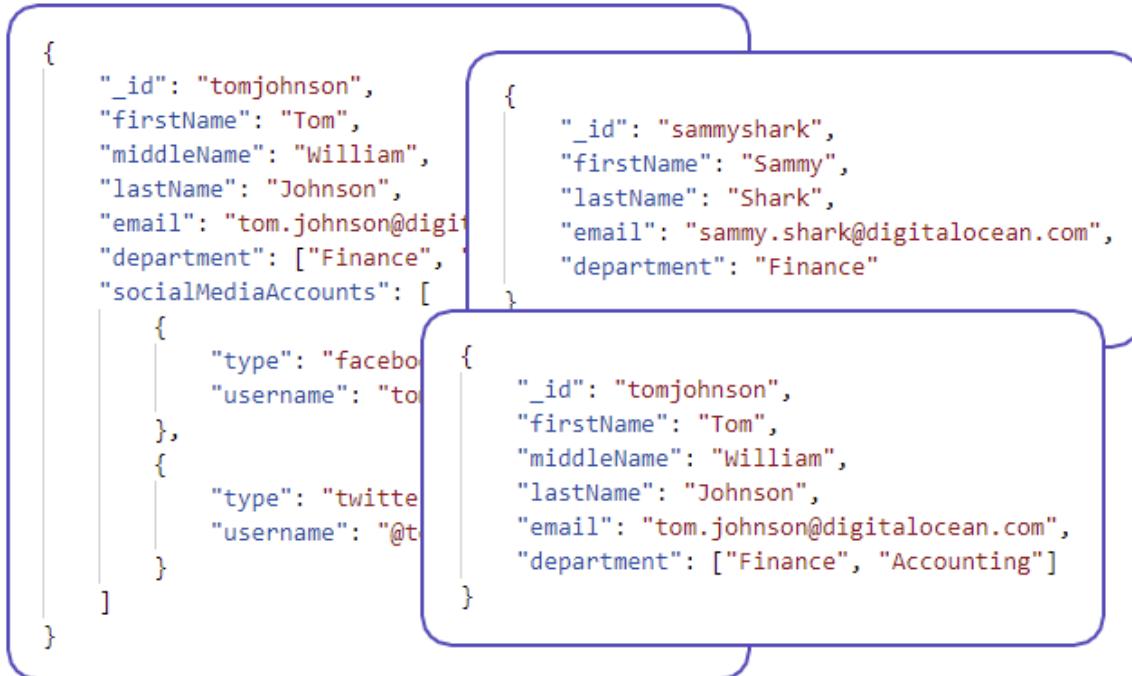
Các database phổ biến: Redis, Memcached và DynamoDB

#### 1.3.2. Document

Một document là một bản ghi trong cơ sở dữ liệu document. Một document thường lưu trữ thông tin về một đối tượng và bất kỳ siêu dữ liệu nào liên quan đến nó.

Document lưu trữ dữ liệu theo cặp trường giá trị. Các giá trị có thể có nhiều loại và cấu trúc khác nhau, bao gồm chuỗi, số, ngày tháng, mảng hoặc đối tượng. Tài

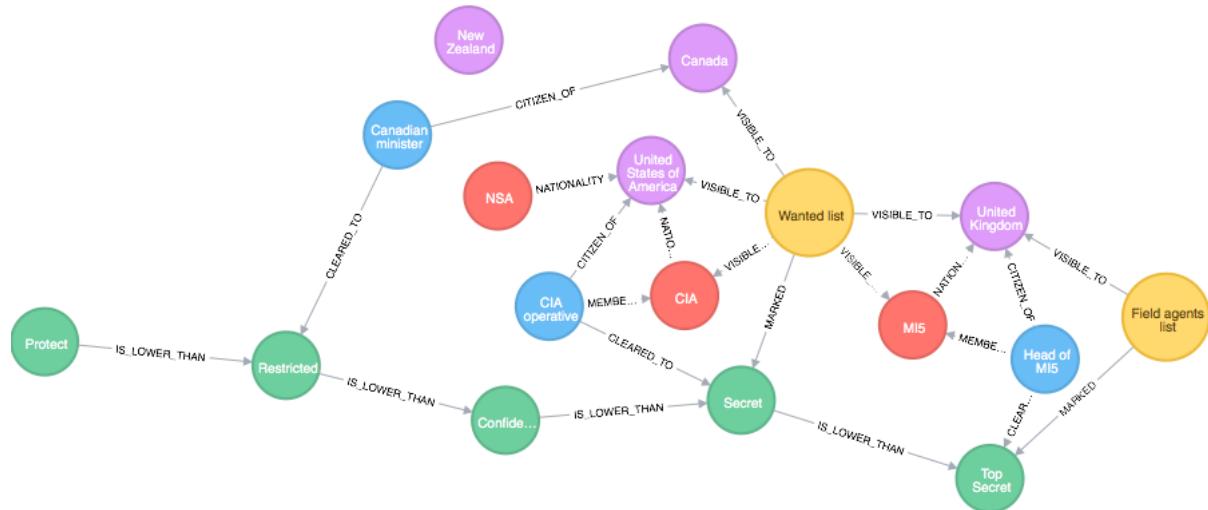
liệu có thể được lưu trữ ở các định dạng như JSON, BSON và XML.



Các database phổ biến: MongoDB, DocumentDB, CouchDB và Cloudant

### 1.3.3. Graph

Cơ sở dữ liệu loại graph lưu trữ các thực thể cũng như mối quan hệ giữa các thực thể đó. Thực thể được lưu trữ dưới dạng một nút với mối quan hệ là các cạnh. Một cạnh đưa ra mối quan hệ giữa các nút. Mỗi nút và cạnh có một mã định danh duy nhất.



Các database phổ biến: Neo4J và Cosmos DB

#### 1.3.4. Column

Sử dụng một tập hợp các cột để lưu trữ dữ liệu. Các tập hợp cột này được gọi là họ cột và người dùng có thể truy vấn trực tiếp các họ cột này mà không cần xem qua tất cả các bản ghi dữ liệu.

Row Key	Customer		Sales	
Customer Id	Name	City	Product	Amount
101	John White	Los Angeles, CA	Chairs	\$400.00
102	Jane Brown	Atlanta, GA	Lamps	\$200.00
103	Bill Green	Pittsburgh, PA	Desk	\$500.00
104	Jack Black	St. Louis, MO	Bed	\$1600.00

Column Families

Các database phổ biến: Cassandra và HBase

#### 1.4. So sánh cơ sở dữ liệu NoSQL với cơ sở dữ liệu quan hệ

	NoSQL	RDBMS
Mô hình dữ liệu	Đa dạng kiểu mô hình dữ liệu như documents, graphs, và key-value pairs. Không yêu cầu schema cố định và có thể lưu trữ dữ liệu có cấu trúc, bán cấu trúc hoặc phi cấu trúc.	Sử dụng mô hình dữ liệu quan hệ, trong đó dữ liệu được tổ chức thành các bảng có các hàng và cột. Các bảng liên kết thông qua khóa ngoại.
Khả năng mở rộng	Thường mở rộng theo chiều ngang (horizontal scaling) bằng cách thêm các node hoặc server	Thường mở rộng theo chiều dọc (vertical scaling) bằng cách nâng cấp phần cứng (tăng CPU, bộ nhớ,

	mới vào hệ thống.	...).
Tính chất	Linh hoạt về tính nhất quán, có thể đánh đổi tính nhất quán để đạt được hiệu suất cao hơn. Tính linh hoạt, khả năng mở rộng và tính sẵn sàng cao được ưu tiên.	Sử dụng ACID (nguyên tử - atomicity, nhất quán - consistency, cô lập - isolation, bền vững - durability) để đảm bảo tính toàn vẹn của giao dịch.
Linh hoạt trong cấu trúc dữ liệu	Linh hoạt hơn trong việc thay đổi cấu trúc dữ liệu, có thể sử dụng để lưu trữ dữ liệu phân tán.	Phức tạp trong việc thay đổi cấu trúc dữ liệu, không phù hợp cho việc lưu trữ phân tán.
Ứng dụng phù hợp	Thường được dùng trong các dự án yêu cầu khả năng mở rộng và kiểu lưu trữ dữ liệu đa dạng (có cấu trúc, bán hoặc phi cấu trúc); xử lý có tính mối quan hệ phức tạp như mạng xã hội, phân tích dữ liệu lớn, IoT, ...	Thích hợp cho các dự án yêu cầu tính toàn vẹn dữ liệu cao như hệ thống quản lý giao dịch, quản lý ngân hàng, ...

## CHƯƠNG 2: CƠ SỞ DỮ LIỆU MONGODB

### 2.1. Giới thiệu

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, thuộc loại NoSQL là được sử dụng rỗng rãi hiện nay.

MongoDB là một dạng database hướng tài liệu hay còn gọi là document, các dữ liệu được lưu trữ dưới dạng document kiểu JSON thay vì dưới dạng bảng như là cơ sở dữ liệu quan hệ cho nên là việc truy vấn sẽ rất là nhanh.

Đối với cơ sở dữ liệu quan hệ thì chúng ta có khái niệm như là bảng, các cơ sở dữ liệu quan hệ như MySQL hay là SQL Server sẽ sử dụng các bảng để lưu trữ dữ liệu thì đối với MongoDB, chúng ta sẽ có khái niệm là collection thay vì là bảng.

So với cơ sở dữ liệu quan hệ thì trong MongoDB, collection tương ứng với table, còn các document thì sẽ ứng với các row, MongoDB sẽ dùng các document này thay cho các row trong cơ sở dữ liệu quan hệ.

Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu được lưu trữ mà không cần tuân theo một cấu trúc nhất định nào.

Thông tin liên quan được lưu trữ cùng nhau để truy cập, truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB.

### 2.2. Mô hình dữ liệu trong MongoDB

Có 2 cách mô hình hóa dữ liệu trong MongoDB:

**Embedded documents:** Mô hình tài liệu nhúng được biết đến là mô hình hoặc lược đồ dữ liệu “không chuẩn hóa”.

- Một ví dụ cụ thể là một bài đăng trên blog và các bình luận tương ứng của chúng là nhúng tài liệu con vào tài liệu cha. Trong ví dụ sau là bài đăng trên blog là tài liệu cha và bình luận là tài liệu con.
- Ưu điểm của mô hình này là có hiệu suất tốt hơn vì chúng ta có thể đọc và cập nhật dữ liệu trong một hoạt động cơ sở dữ liệu duy nhất.
- Nhược điểm của mô hình này là có thể trùng lặp dữ liệu.

```

{
  _id: <ObjectId123>,
  title: "Data Modelling in MongoDB",
  body: "some long text...",
  comments: [
    {
      _id: <ObjectId111>,
      comment: "some text...",
      author: "mike@email.com"
    },
    {
      _id: <ObjectId222>,
      comment: "some text...",
      author: "jake@email.com"
    }
  ]
}

```

**References:** Sử dụng mô hình này, chúng ta có thể mô tả mối quan hệ giữa các tài liệu bằng cách sử dụng tham chiếu. Điều này cũng được biết đến là mô hình hoặc lược đồ dữ liệu “Chuẩn hóa”.

- Ví dụ về mô hình này: cũng là bài đăng trên blog và các bình luận. Ngoài dữ liệu của riêng nó, mỗi tài liệu bình luận cũng chứa một tham chiếu đến bài đăng trên blog cha bằng cách sử dụng id của tài liệu bài đăng trên blog cha.
- **Ưu điểm:** Không có sự trùng lặp dữ liệu, có thể biểu diễn mối quan hệ phức tạp nhiều - nhiều và có thể biểu diễn tập dữ liệu phân cấp.
- **Nhược điểm:** Khi cần truy xuất dữ liệu thì cần nhiều thao tác với cơ sở dữ liệu hơn hoặc cần tham gia đến nhiều collections.

```

// blog post
{
  _id: <ObjectId123>,
  title: "Data Modelling in MongoDB",
  body: "some long text..."
}

// comments
{
  _id: <ObjectId111>,
  comment: "some text...",
  author: "mike@email.com",
  postId: <ObjectId123> // reference to the blog post
},
{
  _id: <ObjectId222>,
  comment: "some text...",
  author: "jake@email.com",
  postId: <ObjectId123> // reference to the blog post
}

```

### 2.3. Kiến trúc của MongoDB

MongoDB sử dụng mô hình dữ liệu NoSQL (không có cấu trúc cố định như trong SQL), cho phép lưu trữ và truy vấn dữ liệu dưới dạng JSON. Mô hình này linh hoạt và có thể mở rộng, cho phép thay đổi cấu trúc dữ liệu mà không cần phải thay đổi cấu trúc của cơ sở dữ liệu.

Một số thuộc tính kiến trúc chính của MongoDB bao gồm:

- Tài liệu (Documents): Dữ liệu được lưu trữ trong các tài liệu JSON, mỗi tài liệu được lưu trữ trong một bản ghi trong một bộ sưu tập.
- Bộ sưu tập (Collections): Một bộ sưu tập tương tự như một bảng trong cơ sở dữ liệu quan hệ, nhưng không có cấu trúc cố định. Nó chứa một tập hợp các tài liệu có thể có cấu trúc khác nhau.
- Cơ sở dữ liệu (Database): MongoDB hỗ trợ nhiều cơ sở dữ liệu, mỗi cơ sở dữ liệu có thể chứa nhiều bộ sưu tập.
- Chỉ mục (Indexes): MongoDB hỗ trợ việc tạo các chỉ mục để tăng tốc độ truy vấn dữ liệu.

- Sao chép (Replication): MongoDB hỗ trợ sao chép dữ liệu sang nhiều nút, giúp tăng tính sẵn sàng và độ tin cậy của hệ thống.
- Phân vùng (Sharding): MongoDB cung cấp tính năng phân vùng dữ liệu, cho phép phân tán dữ liệu trên nhiều nút, giúp tăng khả năng mở rộng của hệ thống.  
Kiến trúc của MongoDB được thiết kế để tối ưu hóa cho các ứng dụng có yêu cầu về tính linh hoạt, mở rộng và hiệu suất cao.

## 2.4. Các chế độ thực thi MongoDB

### 2.4.1. Mongo Shell

Mongo Shell là một giao diện dòng lệnh cung cấp các công cụ để tương tác trực tiếp với cơ sở dữ liệu MongoDB.

Có thể sử dụng Mongo Shell để thực hiện các thao tác như truy vấn dữ liệu, thêm, sửa đổi hoặc xóa các tài liệu, tạo và quản lý các chỉ mục, xem và quản lý dữ liệu, và nhiều hơn nữa. Shell được viết bằng JavaScript và cung cấp một loạt các lệnh và hàm để thực hiện các thao tác trên cơ sở dữ liệu.

### 2.4.2. Admin API

Admin API là một tập hợp các API RESTful cho phép quản lý các cụm MongoDB, cơ sở dữ liệu, bộ sưu tập và người dùng từ xa thông qua HTTP.

Với Admin API có thể thực hiện các thao tác như tạo và quản lý cụm, xem thông tin về các cụm và nút, tạo và quản lý người dùng và vai trò, cũng như quản lý bộ sưu tập và chỉ mục.

Admin API cung cấp một phương tiện linh hoạt và tự động hóa để quản lý cơ sở dữ liệu MongoDB từ xa, điều này rất hữu ích cho việc tự động hóa công việc quản trị hệ thống.

## CHƯƠNG 3: LÀM VIỆC VỚI MONGODB

### 3.1. Cài đặt và cấu hình MongoDB

#### 3.1.1. Cài đặt

Trước tiên, tiến hành cài đặt MongoDB Community Server và MongoDB Shell bằng cách truy cập vào các liên kết sau:

- MongoDB Community Server:

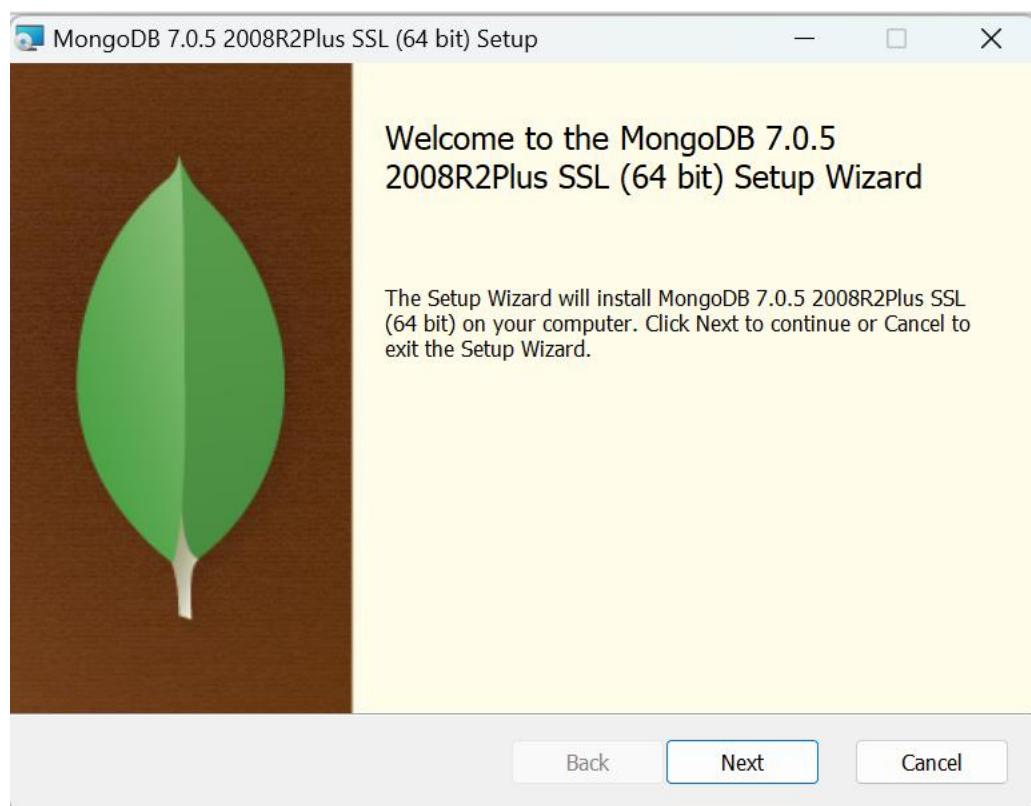
<https://www.mongodb.com/try/download/community>

- MongoDB Shell: <https://www.mongodb.com/try/download/shell>

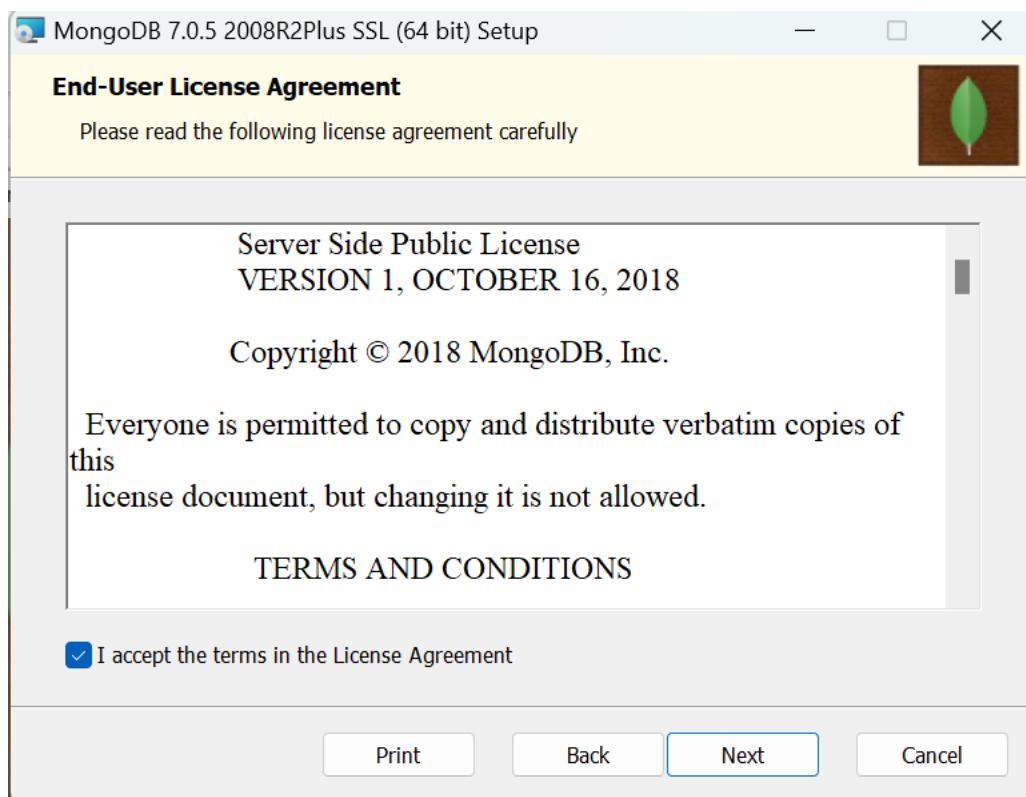
#### 3.1.2. Cấu hình

##### 3.1.2.1. Cấu Hình MongoDB Compass

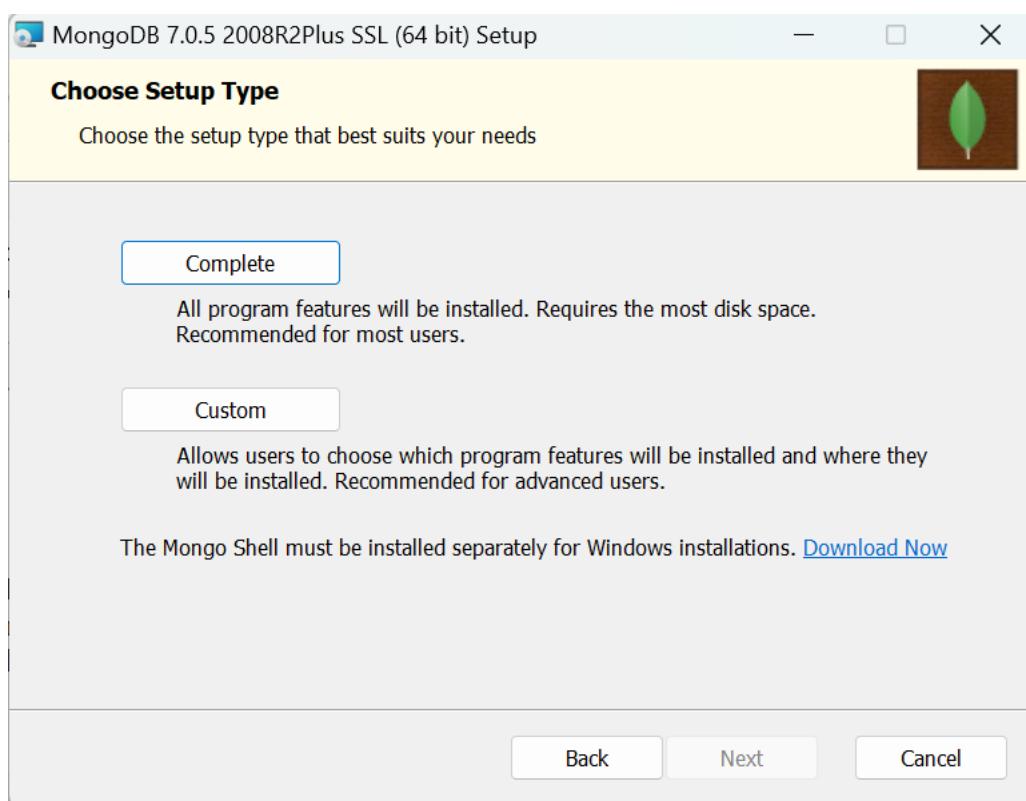
Nhấn “Next”.



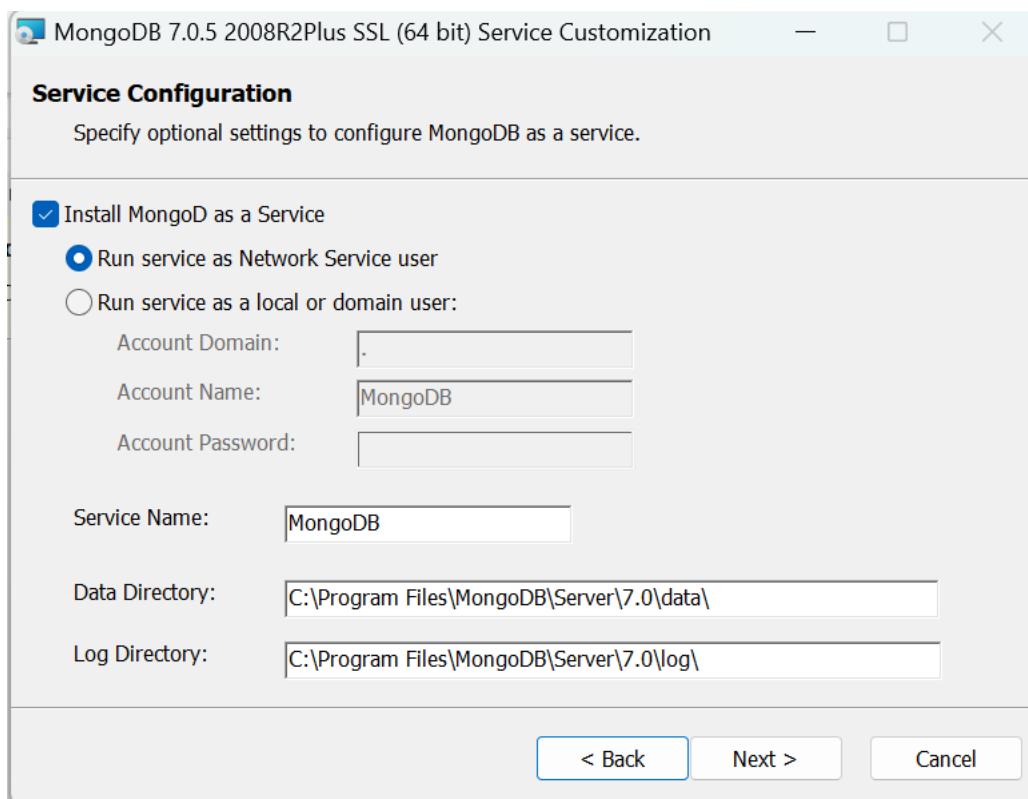
Tích “I accept the term in the License Agreement” và chọn “Next”.



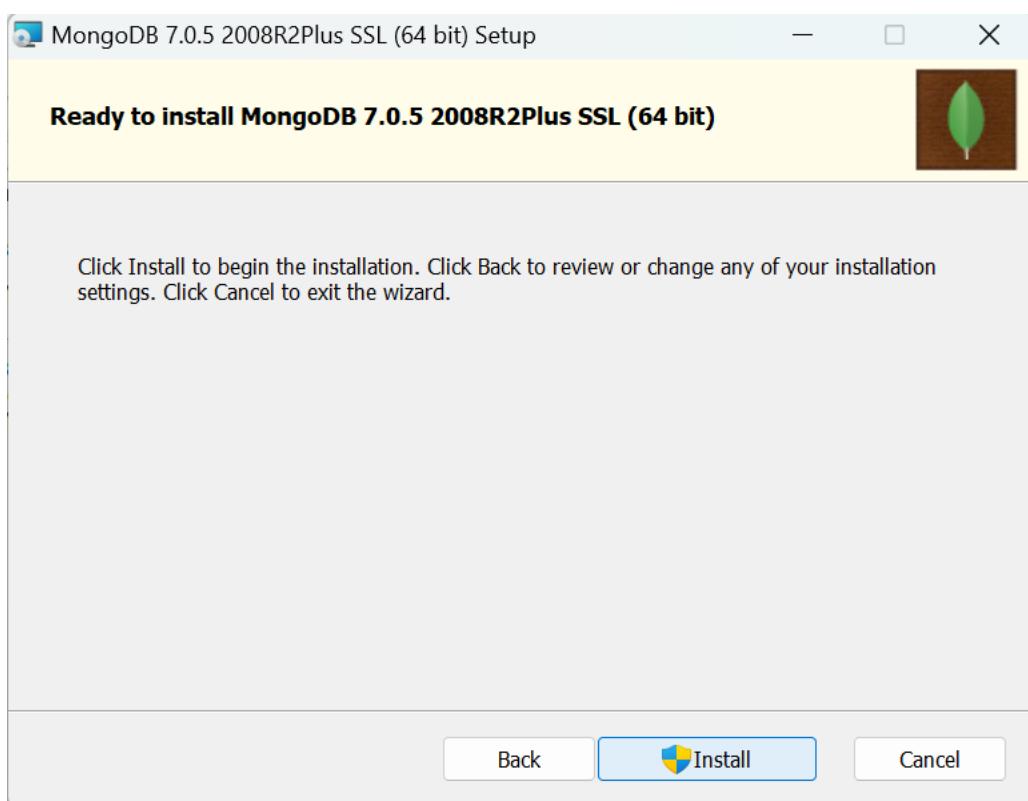
Nhấn “Complete”.

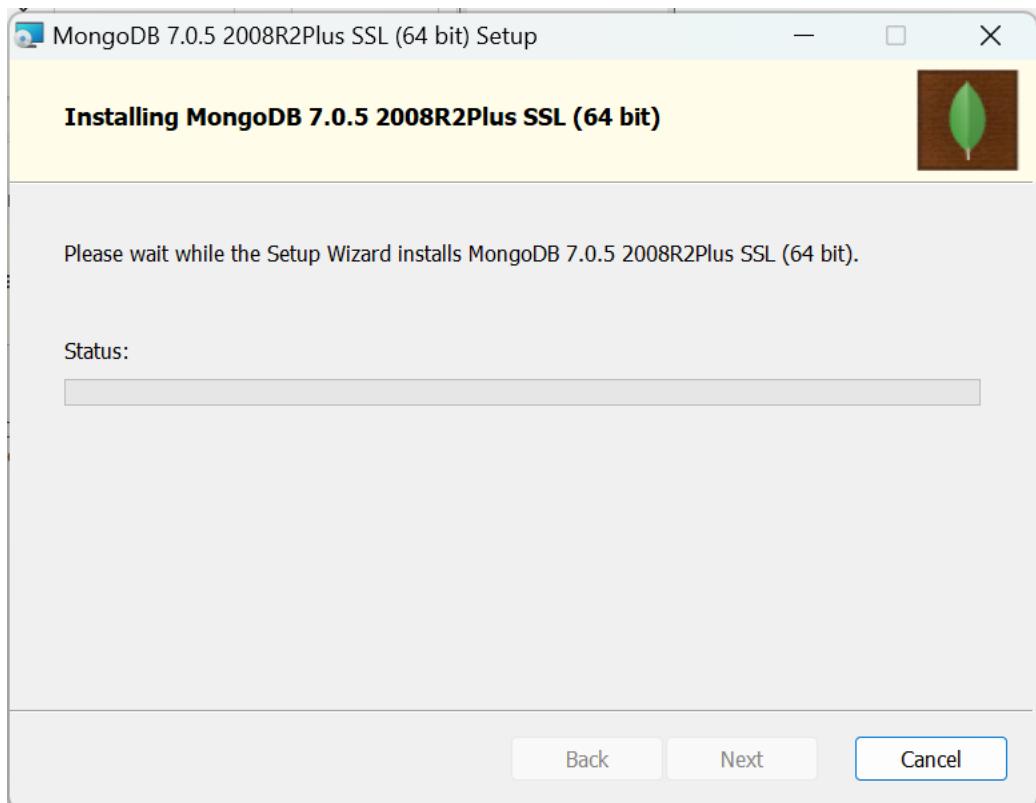


Nhập các thông tin và chọn “Next”.

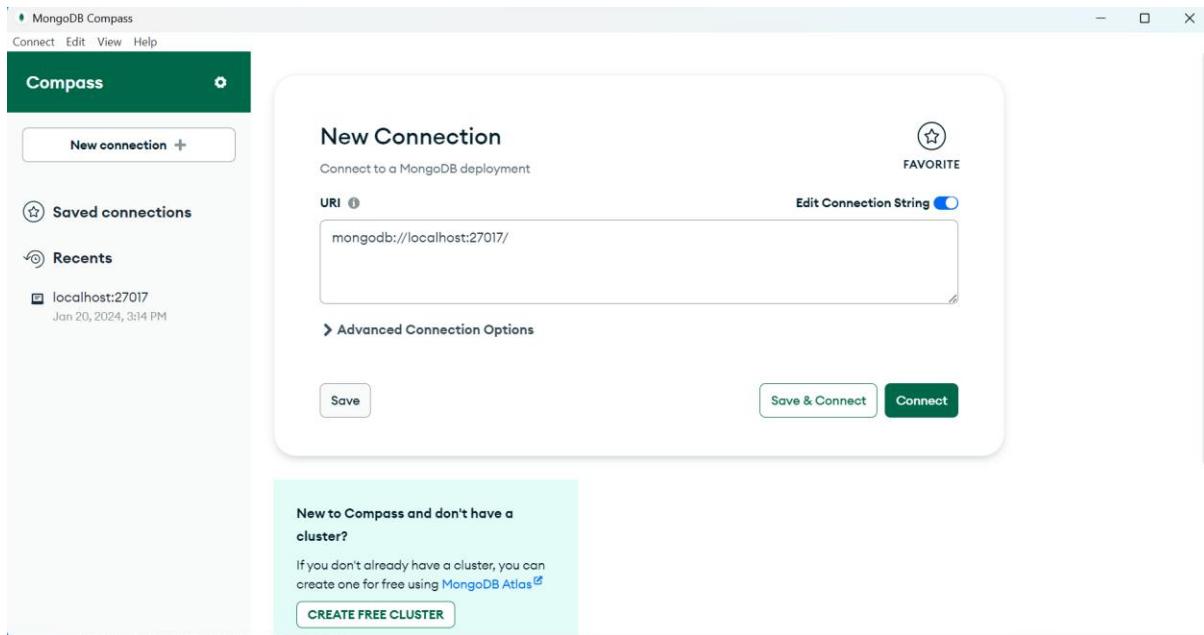


Chọn “Install”.

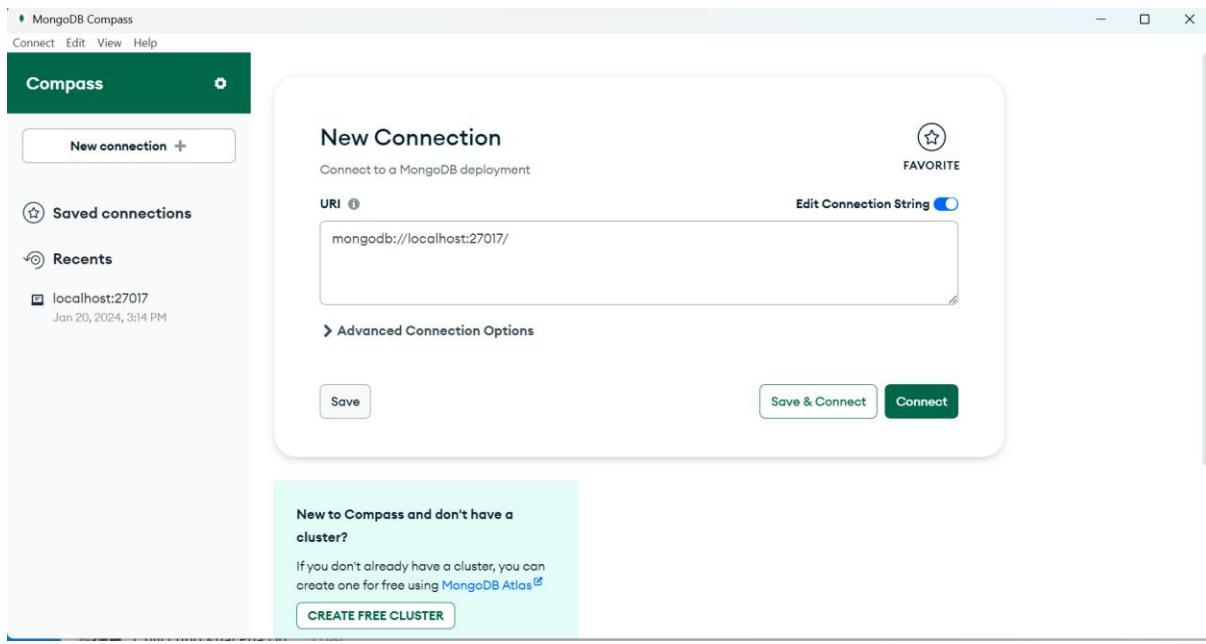




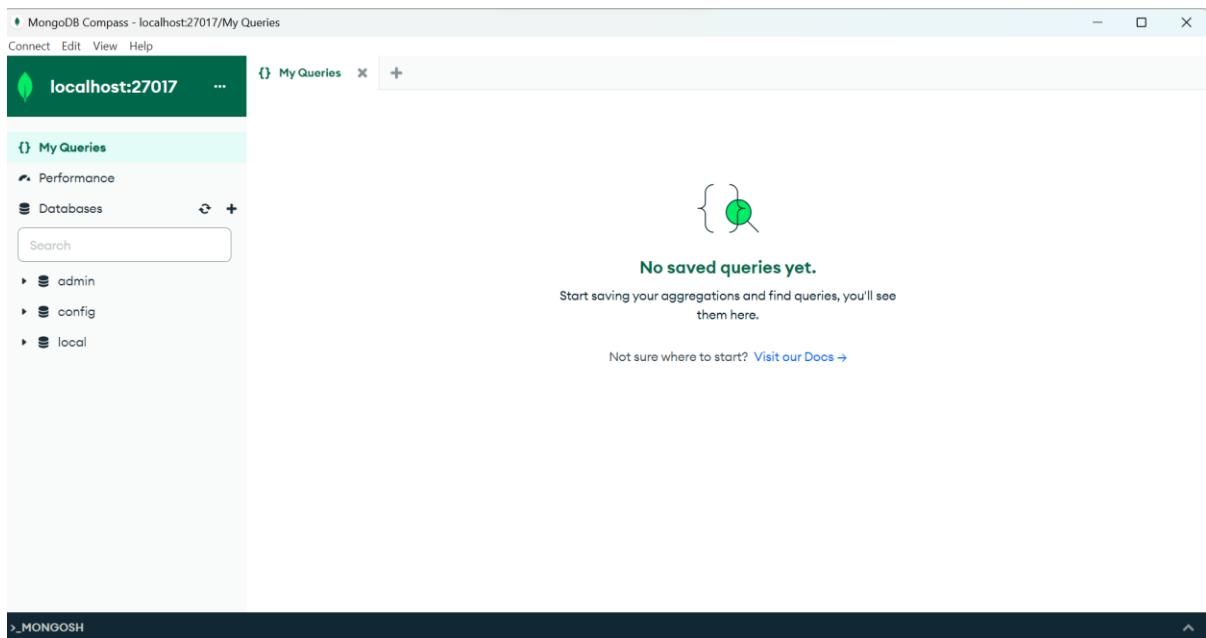
## Cài đặt thành công



Kết nối với mongoDB server: Nhấn connect để kết nối mongodb://localhost:27017

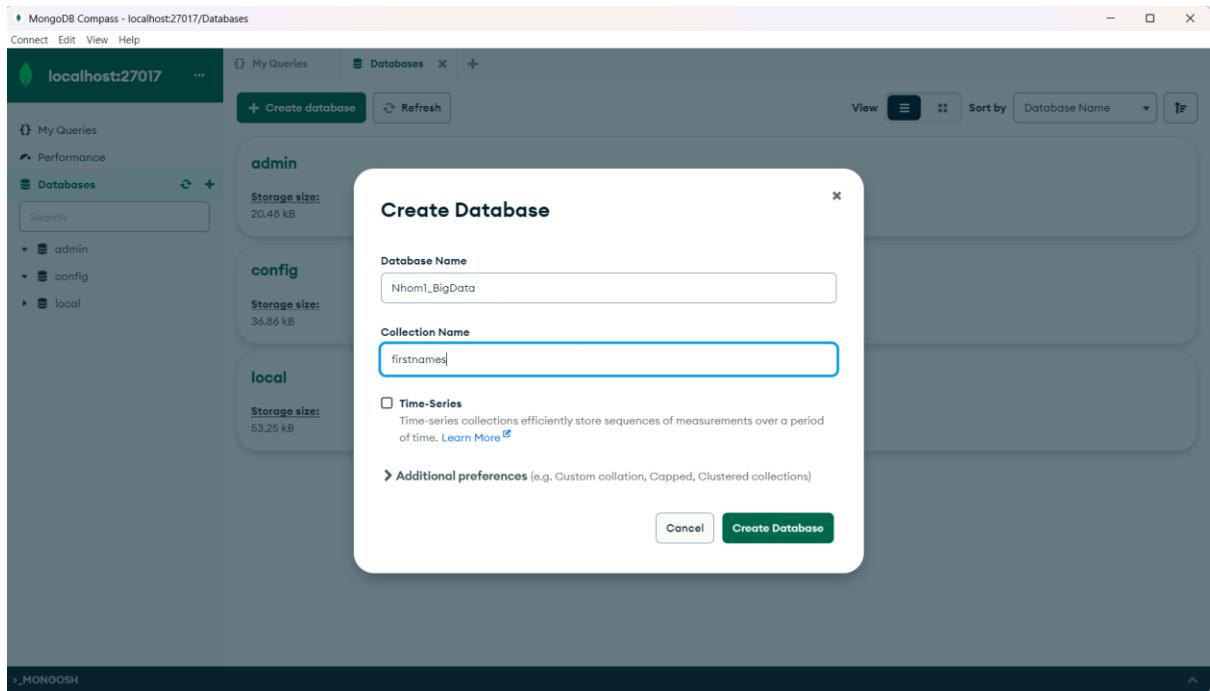


Kết quả: Giao diện chính mongoDB Compass với localhost:27017

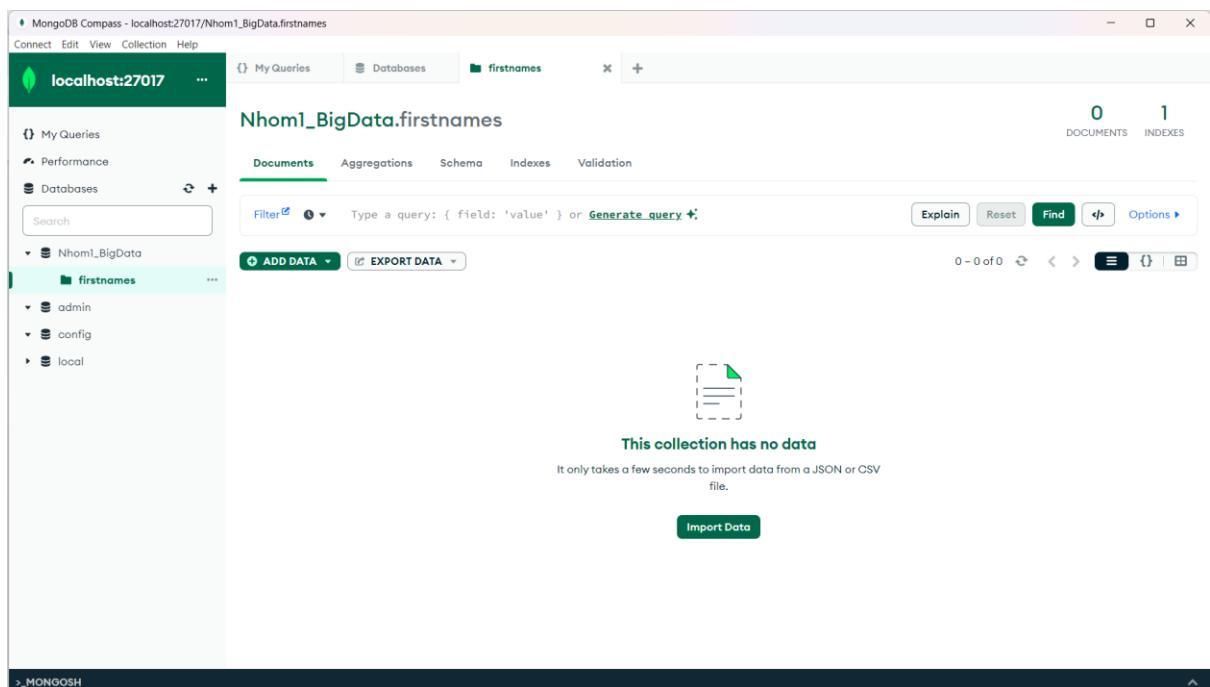


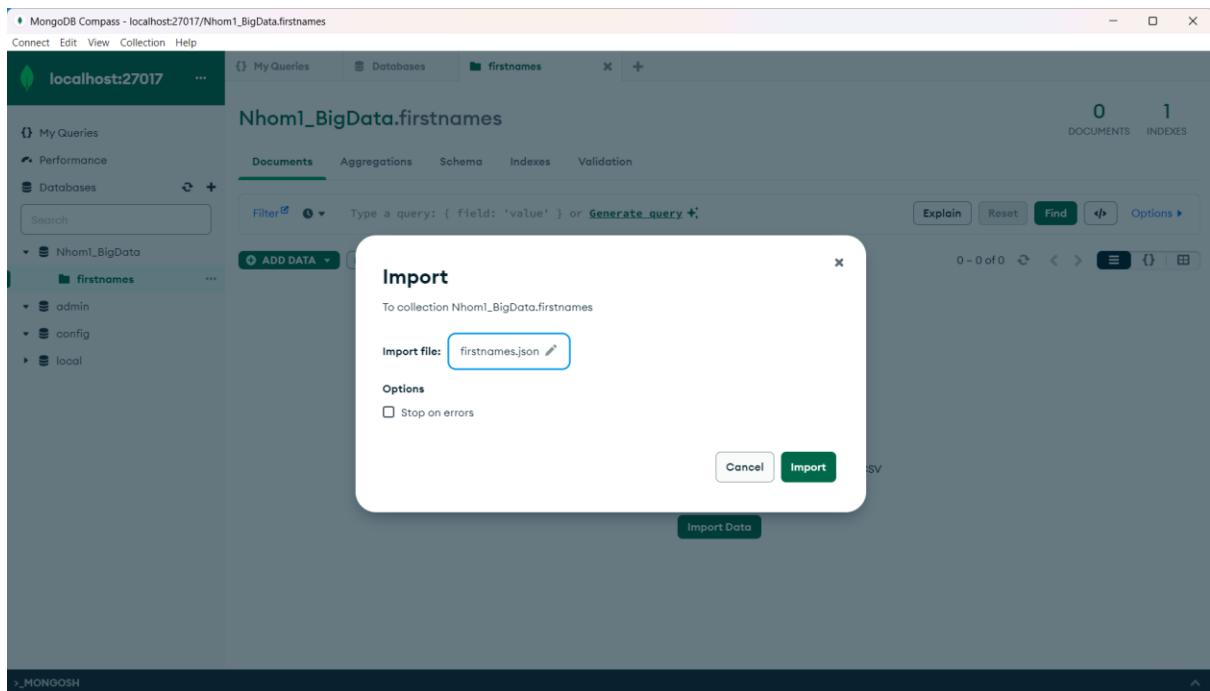
Tạo database trên mongoDB compass: Nhấn nút icon dấu “+” bên phải chữ “Databases”, hộp thoại “Create Database” được mở ra

- Nhập tên database
  - Nhập tên collection
- ➔ Create Database

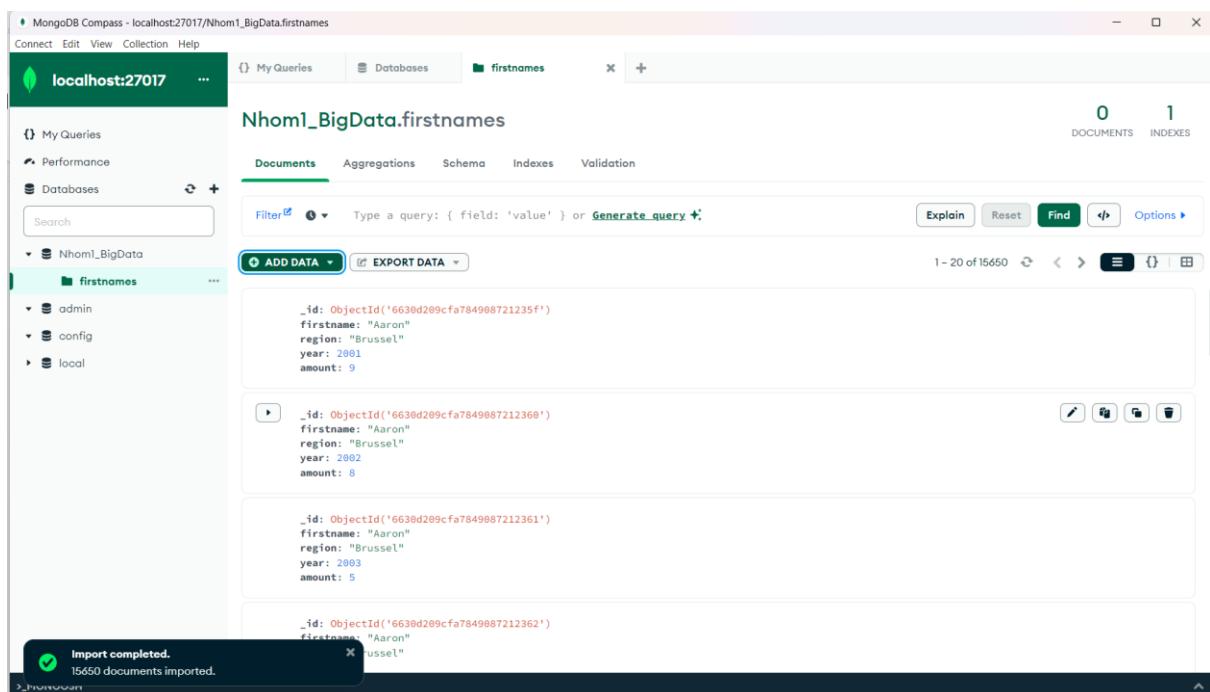


Import data: Sau khi kết nối với MongoDB Server xong, chọn tùy chọn "Import Data" và sau đó chọn tập tin dữ liệu muốn import.



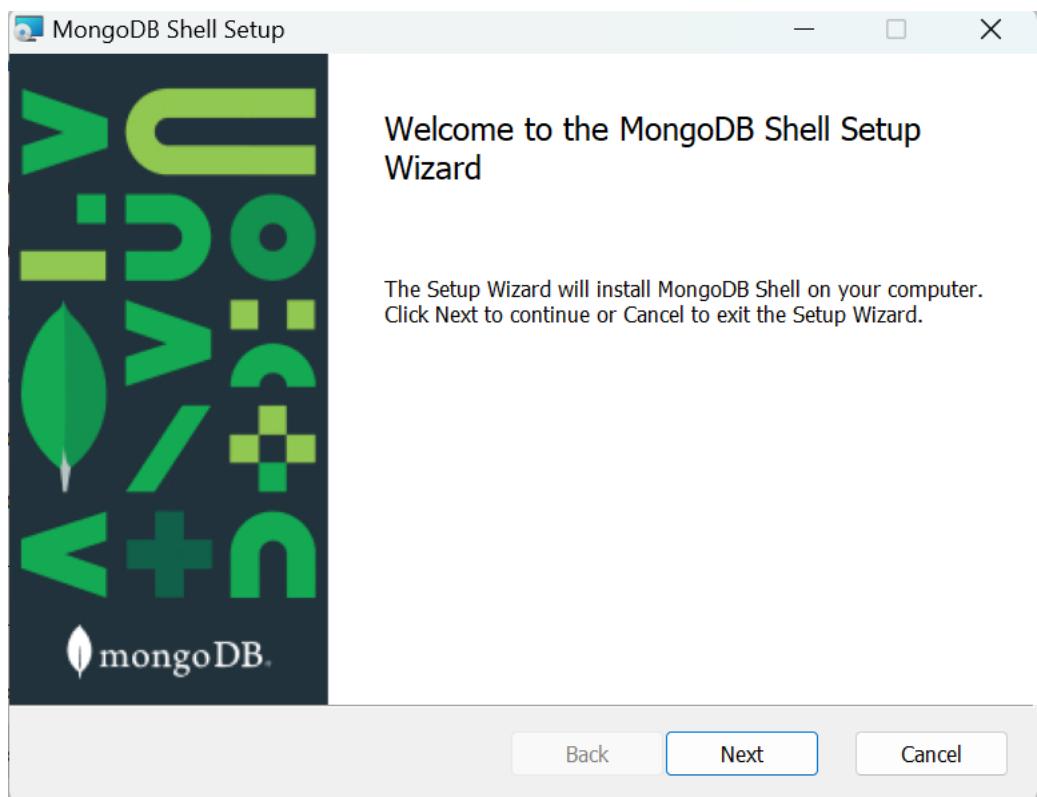


Import dữ liệu thành công

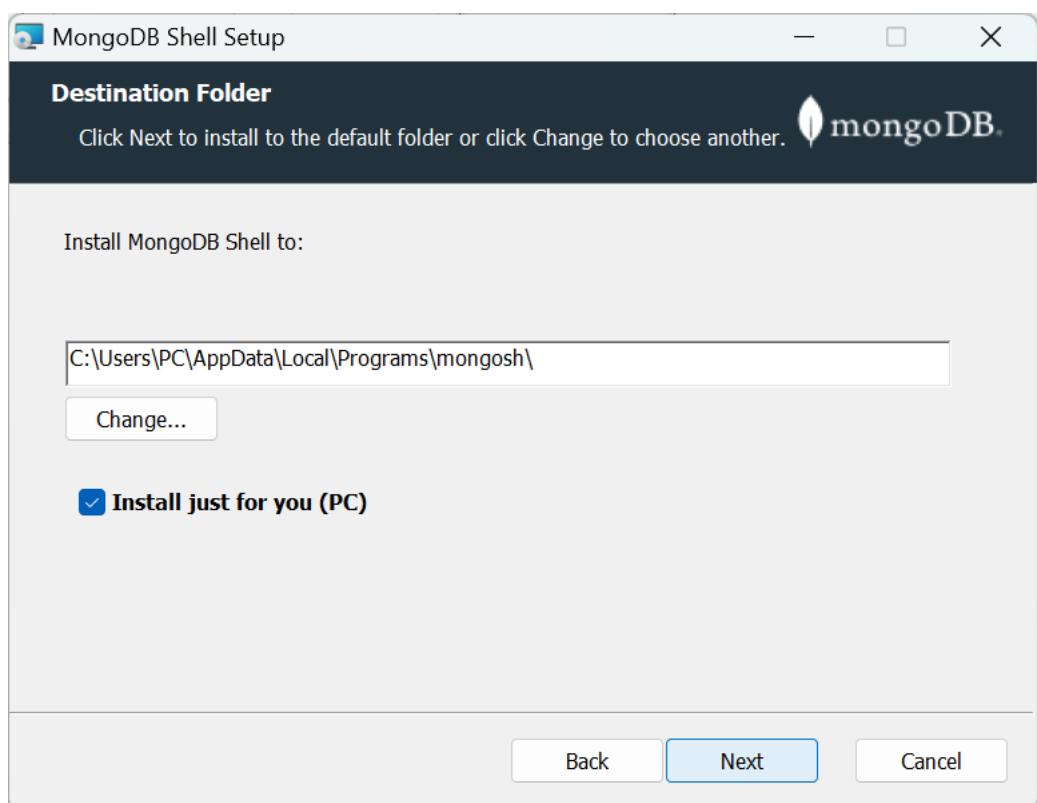


### 3.1.2.2. Cài đặt Mongo Shell

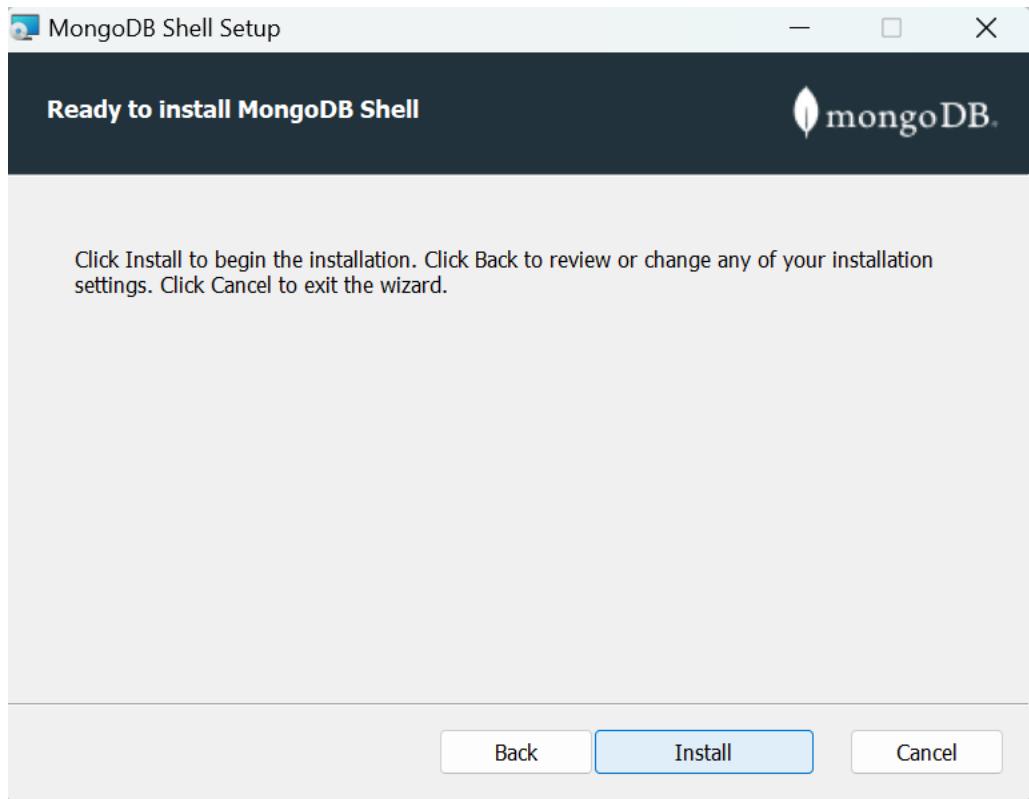
Chọn “Next”.



Điền thông tin và chọn “Next”.



Chọn “Install”.



Cài đặt thành công và kiểm tra phiên bản của mongosh

```
C:\Users\Admin>mongosh
Current Mongosh Log ID: 65a56a1cd4a52cd01d835342
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.1
Using MongoDB:     7.0.5
Using Mongosh:    2.1.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

-----
The server generated these startup warnings when booting
2024-01-15T23:05:47.652+07:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test> |
```

Xem các cơ sở dữ liệu có trên máy:

```
test> show dbs
admin          40.00 KiB
config         72.00 KiB
local          88.00 KiB
mydatabase     8.00 KiB
school         72.00 KiB
test> use school
```

Sử dụng thành công cơ sở dữ liệu

```
test> use school
switched to db school
school> |
```

### 3.2. Tạo lập cơ sở dữ liệu

**Chọn hoặc tạo cơ sở dữ liệu (Database):** Sử dụng lệnh ‘use’ để chọn một cơ sở dữ liệu đã tồn tại hoặc để tạo một cơ sở dữ liệu mới.

Ví dụ: use mydatabase

```
test> use mydatabase
switched to db mydatabase
mydatabase> |
```

Trong ví dụ này, nếu mydatabase đã tồn tại, cơ sở dữ liệu này sẽ được chọn. Nếu không, cơ sở dữ liệu mydatabase sẽ được tạo mới.

**Tạo hoặc cập nhật các bảng (Collections):** Có thể tạo mới hoặc cập nhật các bảng trong cơ sở dữ liệu bằng cách sử dụng các lệnh CRUD (Create, Read, Update, Delete) trong MongoDB Shell.

Ví dụ: Để tạo một bảng mới tên “users”, sử dụng lệnh: db.createCollection(“users”)

```
mydatabase> db.createCollection("users")
{ ok: 1 }
mydatabase> |
```

Sau đó kiểm tra xem tất cả các bảng hiện có trong cơ sở dữ liệu bằng lệnh: show collections

```
mydatabase> show collections
users
mydatabase> |
```

### 3.3. Thao tác với dữ liệu

#### 3.3.1. Truy vấn dữ liệu

Để truy vấn dữ liệu trong MongoDB, sử dụng lệnh find() trên collection.

Ví dụ: Ta có collection “students” trong database “school”.

Truy vấn tất cả các documents trong collection "students" thì sử dụng lệnh:

```
db.students.find()
```

```
'school> db. students.find()
[
  {
    _id: ObjectId('6628b95788d868f3ff9f990a'),
    name: 'Alice',
    age: 20,
    grade: 'A'
  },
  {
    _id: ObjectId('6628b95788d868f3ff9f990b'),
    name: 'Bob',
    age: 21,
    grade: 'B'
  },
  {
    _id: ObjectId('6628b95788d868f3ff9f990c'),
    name: 'Charlie',
    age: 22,
    grade: 'C'
  },
  {
    _id: ObjectId('6628b95788d868f3ff9f990d'),
    name: 'David',
    age: 19,
    grade: 'A'
  },
  {
    _id: ObjectId('6628b95788d868f3ff9f990e'),
    name: 'Emma',
    age: 20,
    grade: 'B'
  }
]
```

school> |

Truy vấn documents với điều kiện cụ thể (ví dụ: age lớn hơn 20) thì sử dụng lệnh:

```
db.students.find({age: {$gt: 20}})
```

```
school> db.students.find({ age: { $gt: 20 } })
[
  {
    _id: ObjectId('6628b95788d868f3ff9f990b'),
    name: 'Bob',
    age: 21,
    grade: 'B'
  },
  {
    _id: ObjectId('6628b95788d868f3ff9f990c'),
    name: 'Charlie',
    age: 22,
    grade: 'C'
  }
]
school> |
```

### 3.3.2. Thêm dữ liệu

Để thêm dữ liệu vào một collection, sử dụng lệnh insertOne() hoặc insertMany().

Ví dụ: Chèn một document vào collection "students" sử dụng lệnh:

```
db.students.insertOne({
  name: "John",
  age: 23,
  grade: "B"
})
```

```

school> db.students.insertOne({
...     name: "John",
...     age: 23,
...     grade: "B"
... })
{
    acknowledged: true,
    insertedId: ObjectId('6628bab688d868f3ff9f990f')
}
school> |

```

Ví dụ: Chèn nhiều documents vào collection "students" ta sử dụng lệnh:

```

db.students.insertMany([
    { name: "Sarah", age: 22, grade: "A" },
    { name: "Michael", age: 24, grade: "B" },
    { name: "Emily", age: 21, grade: "A" }
])

school> db.students.insertMany([
...     { name: "Sarah", age: 22, grade: "A" },
...     { name: "Michael", age: 24, grade: "B" },
...     { name: "Emily", age: 21, grade: "A" }
... ])
{
    acknowledged: true,
    insertedIds: {
        '0': ObjectId('6628bb0588d868f3ff9f9910'),
        '1': ObjectId('6628bb0588d868f3ff9f9911'),
        '2': ObjectId('6628bb0588d868f3ff9f9912')
    }
}
school> |

```

### 3.3.3. Cập nhật dữ liệu

Để cập nhật dữ liệu trong MongoDB, sử dụng lệnh updateOne() hoặc updateMany().

Ví dụ: Cập nhật một document trong collection "students" có tên là "John".

```

db.students.updateOne(
    { name: "John" }, // Điều kiện để tìm document cần cập nhật
    { $set: { grade: "A" } } // Cập nhật trường "grade" thành "A"
)

```

```
school> db.students.updateOne(  
...     { name: "John" },  
...     { $set: { grade: "A" } }  
... )  
{  
    acknowledged: true,  
    insertedId: null,  
    matchedCount: 1,  
    modifiedCount: 1,  
    upsertedCount: 0  
}  
school> |
```

→ Dữ liệu được cập nhật thành công sử dụng lệnh db.students.find() để truy vấn lại thông tin

```
{  
    _id: ObjectId('6628bab688d868f3ff9f990f'),  
    name: 'John',  
    age: '23',  
    grade: 'A'  
},  
{
```

Ví dụ: Cập nhật nhiều documents trong collection "students" có điều kiện tuổi lớn hơn 20.

```
db.students.updateMany(  
    { age: { $gt: 20 } }, // Điều kiện để tìm các documents cần cập nhật  
    { $set: { grade: "B" } } // Cập nhật trường "grade" thành "B"  
)
```

```
school> db.students.updateMany(  
...     { age: { $gt: 20 } },  
...     { $set: { grade: "B" } }  
... )  
{  
    acknowledged: true,  
    insertedId: null,  
    matchedCount: 6,  
    modifiedCount: 4,  
    upsertedCount: 0  
}  
school> |
```

→ Dữ liệu được cập nhật thành công

```
school> db.students.find()
[  
  {  
    _id: ObjectId('6628b95788d868f3ff9f990a'),  
    name: 'Alice',  
    age: 20,  
    grade: 'A'  
  },  
  {  
    _id: ObjectId('6628b95788d868f3ff9f990b'),  
    name: 'Bob',  
    age: 21,  
    grade: 'B'  
  },  
  {  
    _id: ObjectId('6628b95788d868f3ff9f990c'),  
    name: 'Charlie',  
    age: 22,  
    grade: 'B'  
  },  
  {  
    _id: ObjectId('6628b95788d868f3ff9f990d'),  
    name: 'David',  
    age: 19,  
    grade: 'A'  
  },  
  {  
    _id: ObjectId('6628b95788d868f3ff9f990e'),  
    name: 'Emma',  
    age: 20,  
    grade: 'B'  
  },  
]
```

### 3.3.4. Xóa dữ liệu

Để xóa dữ liệu trong MongoDB, sử dụng lệnh deleteOne() hoặc deleteMany().

Ví dụ: Xóa một document từ collection "students" có tên là "John" sử dụng lệnh:

```
db.students.deleteOne({ name: "John" })
```

```
] school> db.students.deleteOne({ name: "John" })  
{ acknowledged: true, deletedCount: 1 }  
school> |
```

Ví dụ: Xóa nhiều documents từ collection "students" có điều kiện tuổi nhỏ hơn 22 sử dụng lệnh:

```
db.students.deleteMany({ age: { $lt: 22 } })
```

```
school> db.students.deleteMany({ age: { $lt: 22 } })
{ acknowledged: true, deletedCount: 5 }
school> |
```

## CHƯƠNG 4: LÀM VIỆC VỚI MONGODB

### 4.1. Mô tả bài toán

Bài toán yêu cầu xây dựng một trang web thương mại điện tử tương tự như tiki.vn. Trang web này sẽ cung cấp các sản phẩm từ nhiều danh mục khác nhau, cho phép người dùng tìm kiếm, xem thông tin chi tiết sản phẩm, thêm sản phẩm vào giỏ hàng và thực hiện thanh toán.

### 4.2. Mô tả cơ sở dữ liệu, tập dữ liệu (nếu có)

Cơ sở dữ liệu Ecommerce bao gồm 3 collection: users, products, orders với mô tả các thuộc tính cho các đối tượng khác nhau trong hệ thống:

#### - **users:**

- **email:** Địa chỉ email của người dùng.
- **password:** Mật khẩu của người dùng.
- **isAdmin:** Đánh dấu xem người dùng có quyền quản trị (admin) hay không.
- **createdAt:** Thời điểm tạo ra tài khoản người dùng.
- **updatedAt:** Thời điểm cập nhật gần nhất của tài khoản người dùng.
- **address:** Địa chỉ của người dùng.
- **avatar:** Hình đại diện của người dùng.
- **name:** Tên của người dùng.
- **phone:** Số điện thoại của người dùng.
- **city:** Thành phố của người dùng.

#### - **products:**

- **name:** Tên sản phẩm.
- **image:** Hình ảnh sản phẩm.
- **type:** Loại sản phẩm.
- **price:** Giá của sản phẩm.
- **countInStock:** Số lượng sản phẩm có sẵn trong kho.
- **rating:** Điểm đánh giá của sản phẩm.
- **description:** Mô tả về sản phẩm.
- **discount:** Mức giảm giá của sản phẩm (nếu có).
- **createdAt:** Thời điểm tạo ra sản phẩm.
- **updatedAt:** Thời điểm cập nhật gần nhất của sản phẩm.
- **selled:** Số lượng sản phẩm đã bán được.

#### - **orders:**

- **orderItems:** Các mặt hàng trong đơn hàng.

- **Object:** Một đối tượng không được xác định cụ thể.
- **paymentMethod:** Phương thức thanh toán của đơn hàng.
- **shippingPrice:** Phí vận chuyển của đơn hàng.
- **totalPrice:** Tổng giá của đơn hàng.
- **user:** Người dùng đặt hàng.
- **isPaid:** Đánh dấu xem đơn hàng đã thanh toán hay chưa.
- **isDelivered:** Đánh dấu xem đơn hàng đã được giao hay chưa.
- **createdAt:** Thời điểm đơn hàng được tạo ra.
- **updatedAt:** Thời điểm cập nhật gần nhất của đơn hàng.
- **\_v:** Phiên bản của đối tượng.

#### **4.3. Các chức năng chính của ứng dụng**

##### **Trang chủ:**

- Xem sản phẩm
- Xem chi tiết sản phẩm
- Xem sản phẩm , Xem thêm
- Xem sản phẩm theo loại.

##### **Tài khoản :**

- Đăng nhập
- Đăng ký
- Đăng xuất
- Cập nhật thông tin tài khoản

##### **Đơn Hàng:**

- Đặt hàng
- Hủy đơn hàng

##### **Quản lý hệ thống:**

- **Người dùng:**
  - Xóa thông tin
  - Chính sửa thông tin
  - Tìm kiếm người dùng
  - Xem danh sách thông tin người dùng
- **Sản phẩm:**
  - Thêm Sản phẩm
  - Tìm kiếm sản phẩm
  - Chính sửa sản phẩm
  - Xóa sản phẩm
  - xem danh sách thông tin sản phẩm

##### **- Đơn hàng:**

- Xem đơn hàng
- Xem báo cáo đơn hàng

#### 4.4. Hướng dẫn cài đặt

Bước 1: Đứng tại thư mục C:\ECommerce\server, chạy cmd và gõ lệnh npm start

```
C:\ECommerce\server>npm start
> ecommerce-backend@1.0.0 start
> nodemon src/index.js

[nodemon] 2.0.22
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): ***!
[nodemon] watching extensions: js,mjs,json
[nodemon] starting 'node src/index.js'
(node:1144) [MONGOOSE] DeprecationWarning: Mongoose: the 'strictQuery' option will be switched back to 'false' by default in Mongoose 7. Use `mongoose.set('strictQuery', false);` if you want to prepare for this change. Or use `mongoose.set('strictQuery', true);` to suppress this warning.
(Use `node --trace-deprecation ...` to show where the warning was created)
Server is running in port: 3001
Connect is successfully!!!
|
```

Bước 2: Đứng tại thư mục C:\ECommerce\server, chạy cmd và gõ lệnh npm start

```
Compiled successfully!

You can now view ecommerce in the browser.

http://localhost:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
|
```

Kết quả: Hệ thống chạy trên <http://localhost:3000>

#### 4.5. Kết quả

Sau khi hoàn thành, một trang web thương mại điện tử hoàn chỉnh, cho phép người dùng thực hiện các chức năng như tìm kiếm, mua hàng và quản lý tài khoản.

##### Trang chủ

###### - Xem sản phẩm

Code frontend sẽ tiến hành gọi API lấy tất cả sản phẩm với giới hạn là 12.

```
const searchProduct = useSelector((state) => state?.product?.search)
const [products, setProducts] = useState([])
const [bounce, bounceSearchProduct] = useBounce(searchProduct, 500)
const [limit, setLimit] = useState(12)
const [typeProducts, setTypeProducts] = useState([])
```

```
}

const { isLoading, data: products, isPreviousData } = useQuery(['products', limit, searchDebounce], fetchProductAll, { retry: 3, retryDelay: 1000, keepPreviousData: true })

💡

const fetchProductAll = async (context) => {
  const limit = context?.queryKey && context?.queryKey[1]
  const search = context?.queryKey && context?.queryKey[2]
  const res = await ProductService.getAllProduct(search, limit)

  return res
}

export const getAllTypeProduct = async () => {
  const res = await axios.get(
    `${process.env.REACT_APP_API_URL}/product/get-all-type`
  );
  return res.data;
}
```

Code backend sẽ có API lấy tất cả sản phẩm:

```
7  const getAllProduct = (limit, page, sort, filter) => {
8    return new Promise(async (resolve, reject) => {
9      try {
10        let totalProduct = await Product.count();
11        let allProduct = [];
12        if (filter) {
13          const label = filter[0];
14          totalProduct = (
15            await Product.find({ [label]: { $regex: filter[1], $options: "i" } })
16          ).length;
17          const allObjectFilter = await Product.find({
18            [label]: { $regex: filter[1], $options: "i" },
19          })
20            .limit(limit)
21            .skip(page * limit)
22            .sort({ createdAt: -1, updatedAt: -1 });
23          resolve({
24            status: "OK",
25            message: "Success",
26            data: allObjectFilter,
27            total: totalProduct,
28            pageCurrent: Number(page + 1),
29            totalPages: Math.ceil(totalProduct / limit),
30          });
31        }
32        if (sort) {
33          const objectSort = {};
34          objectSort[sort[1]] = sort[0];
35          const allProductSort = await Product.find()
36            .limit(limit)
37            .skip(page * limit)
38            .sort(objectSort)
39            .sort({ createdAt: -1, updatedAt: -1 });
40        }
41      }
42    }
43  }
```

```

commerce > server > src > services > JS ProductService.js > [x] getAllType > [x] <function>
37  const getAllProduct = (limit, page, sort, filter) => {
38    return new Promise(async (resolve, reject) => {
39      ...
40      .sort({ createdAt: -1, updatedAt: -1 });
41      resolve({
42        status: "OK",
43        message: "Success",
44        data: allObjectFilter,
45        total: totalProduct,
46        pageCurrent: Number(page + 1),
47        totalPages: Math.ceil(totalProduct / limit),
48      });
49    }
50    if (sort) {
51      const objectSort = {};
52      objectSort[sort[1]] = sort[0];
53      const allProductSort = await Product.find()
54        .limit(limit)
55        .skip(page * limit)
56        .sort(objectSort)
57        .sort({ createdAt: -1, updatedAt: -1 });
58      resolve({
59        status: "OK",
60        message: "Success",
61        data: allProductSort,
62        total: totalProduct,
63        pageCurrent: Number(page + 1),
64        totalPages: Math.ceil(totalProduct / limit),
65      });
66    }
67    if (!limit) {
68      allProduct = await Product.find().sort({
69        createdAt: -1,
70        updatedAt: -1,
71      });
72    } else {
73      allProduct = await Product.find()
74        ...
75        .sort({ createdAt: -1, updatedAt: -1 });
76        resolve({
77          status: "OK",
78          message: "Success",
79          data: allProductSort,
80          total: totalProduct,
81          pageCurrent: Number(page + 1),
82          totalPages: Math.ceil(totalProduct / limit),
83        });
84    }
85  });

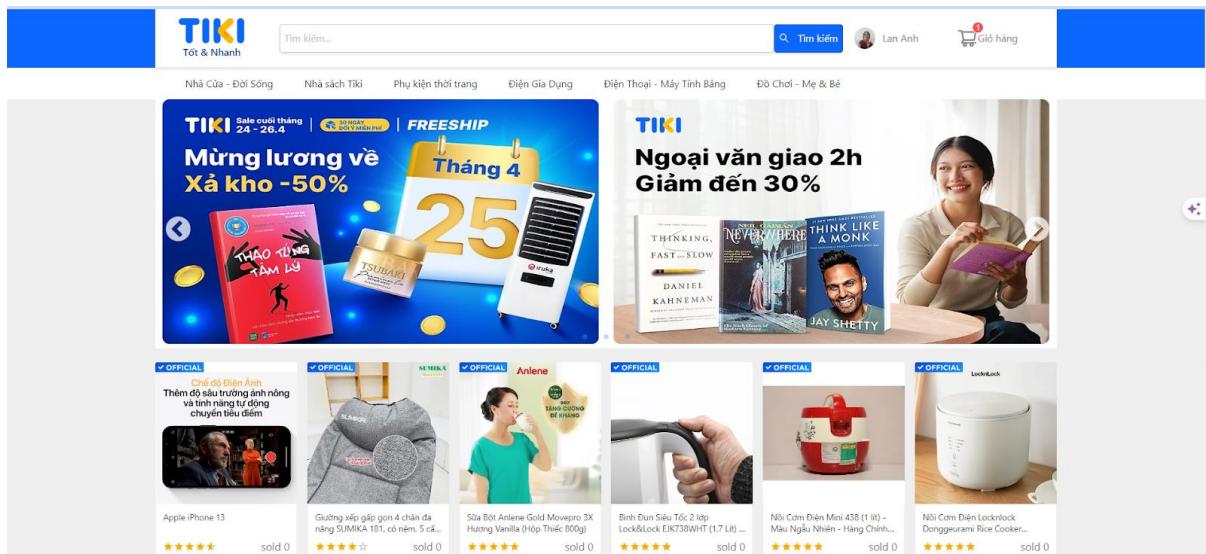
```

```

commerce > server > src > services > JS ProductService.js > [x] getAllType > [x] <function>
37  const getAllProduct = (limit, page, sort, filter) => {
38    return new Promise(async (resolve, reject) => {
39      ...
40      .sort({ createdAt: -1, updatedAt: -1 });
41      resolve({
42        status: "OK",
43        message: "Success",
44        data: allObjectFilter,
45        total: totalProduct,
46        pageCurrent: Number(page + 1),
47        totalPages: Math.ceil(totalProduct / limit),
48      });
49    }
50    if (sort) {
51      const objectSort = {};
52      objectSort[sort[1]] = sort[0];
53      const allProductSort = await Product.find()
54        .limit(limit)
55        .skip(page * limit)
56        .sort(objectSort)
57        .sort({ createdAt: -1, updatedAt: -1 });
58      resolve({
59        status: "OK",
60        message: "Success",
61        data: allProductSort,
62        total: totalProduct,
63        pageCurrent: Number(page + 1),
64        totalPages: Math.ceil(totalProduct / limit),
65      });
66    }
67    if (!limit) {
68      allProduct = await Product.find().sort({
69        createdAt: -1,
70        updatedAt: -1,
71      });
72    } else {
73      allProduct = await Product.find()
74        ...
75        .sort({ createdAt: -1, updatedAt: -1 });
76        resolve({
77          status: "OK",
78          message: "Success",
79          data: allProductSort,
80          total: totalProduct,
81          pageCurrent: Number(page + 1),
82          totalPages: Math.ceil(totalProduct / limit),
83        });
84    }
85  });

```

Kết quả:



## - Xem chi tiết sản phẩm

Chọn sản phẩm muốn xem thông tin chi tiết.

Code frontend sẽ thực hiện gọi API lấy chi tiết sản phẩm:

```

const fetchGetDetailsProduct = async (context) => {
  const id = context?.queryKey && context?.queryKey[1]
  if(id) {
    const res = await ProductService.getDetailsProduct(id)
    return res.data
  }
};

export const getDetailsProduct = async (id) => {
  const res = await axios.get(
    `${process.env.REACT_APP_API_URL}/product/get-details/${id}`
  );
  return res.data;
};

```

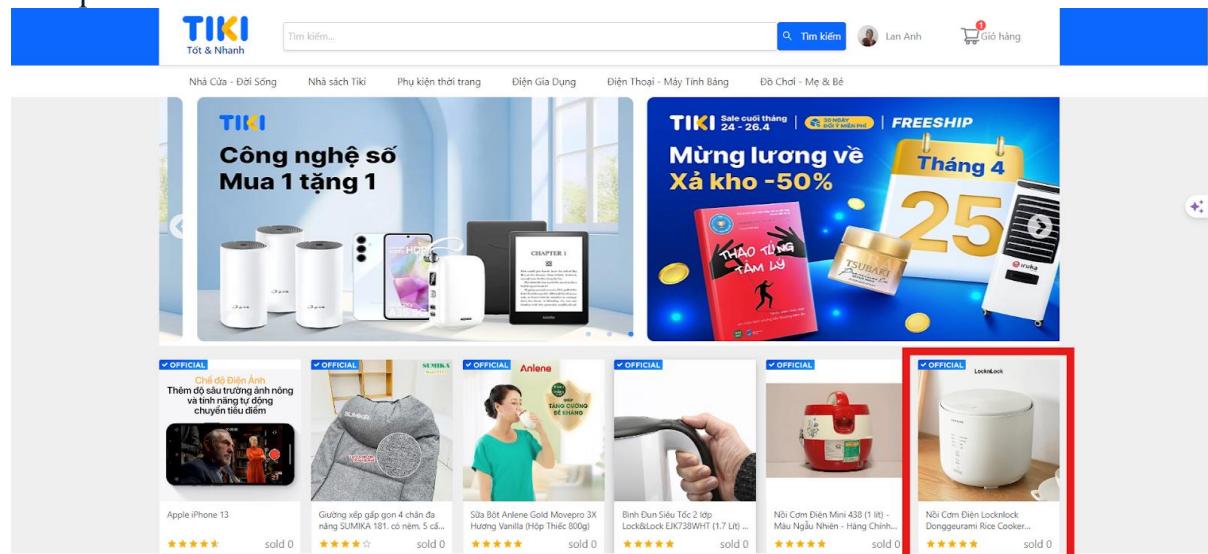
Code backend sẽ có API lấy chi tiết sản phẩm.

```

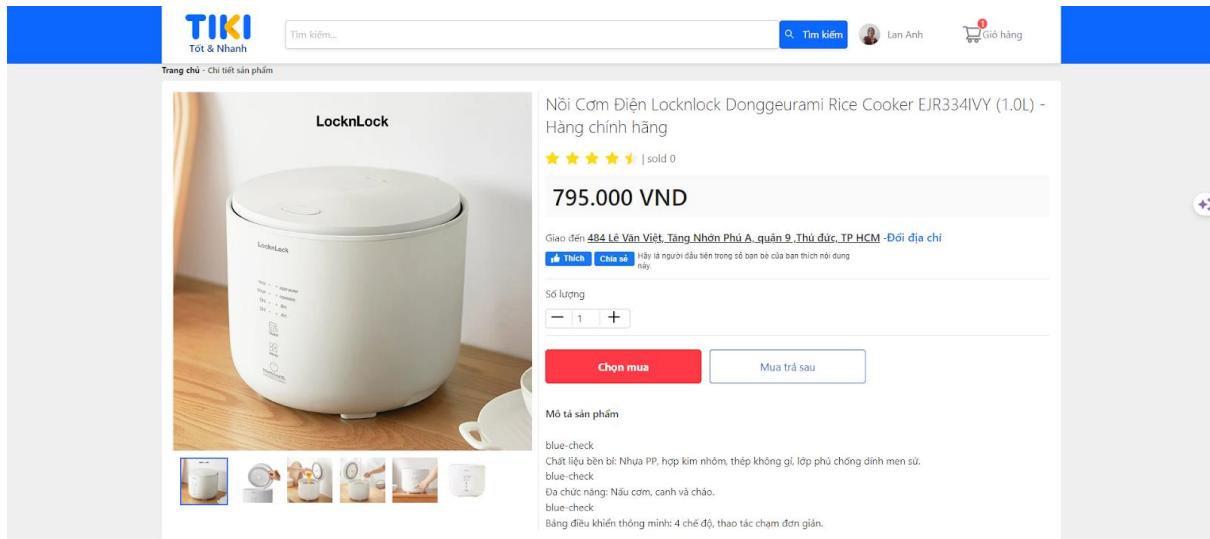
1   };
2
3   const getDetailsProduct = (id) => {
4     return new Promise(async (resolve, reject) => {
5       try {
6         const product = await Product.findOne({
7           _id: id,
8         });
9         if (product === null) {
10           resolve({
11             status: "ERR",
12             message: "The product is not defined",
13           });
14         }
15
16         resolve({
17           status: "OK",
18           message: "SUCESS",
19           data: product,
20         });
21       } catch (e) {
22         reject(e);
23       }
24     });
25   };

```

Kết quả:



Click chuột vào sản phẩm đã chọn -> Giao diện chi tiết sản phẩm được hiển thị.



### - Xem thêm

Nhấn vào nút “Xem thêm” để hiển thị thêm các sản phẩm trên trang chủ.

Code frontend xử lý “Load more”. Khi click vào nút “Load more” thì limit sẽ tăng lên

```

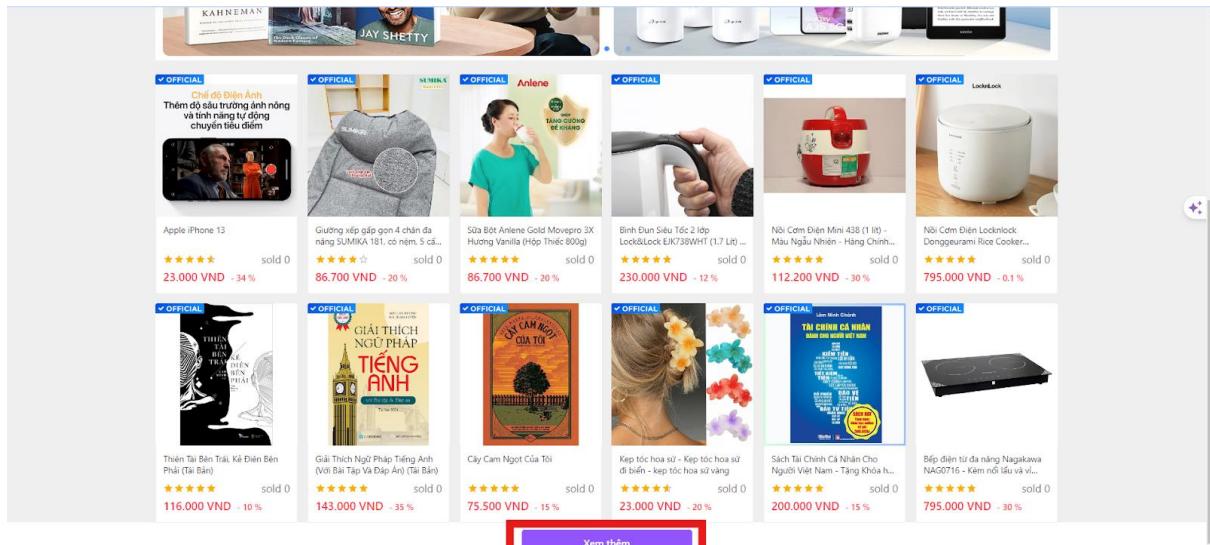
        ...
    </WrapperProducts>
<div style={{ width: '100%', display: 'flex', justifyContent: 'center', marginTop: '10px' }}>
    <WrapperButtonMore
        textbutton={isPreviousData ? 'Load more' : "Xem thêm"} type="outline" styleButton={({
            border: `1px solid ${products?.total === products?.data?.length ? '#f5f5f5' : '#9255FD'}`,
            color: `${products?.total === products?.data?.length ? '#f5f5f5' : '#9255FD'}`,
            width: '240px',
            height: '38px',
            borderRadius: '4px'
        })}
        disabled={products?.total === products?.data?.length || products?.totalPage === 1}
        styleTextButton={{ fontWeight: 500, color: products?.total === products?.data?.length && '#fff' }}
        onClick={() => setLimit((prev) => prev + 6)}
    />
</div>

const fetchProductAll = async (context) => {
    const limit = context?.queryKey && context?.queryKey[1]
    const search = context?.queryKey && context?.queryKey[2]
    const res = await ProductService.getAllProduct(search, limit)

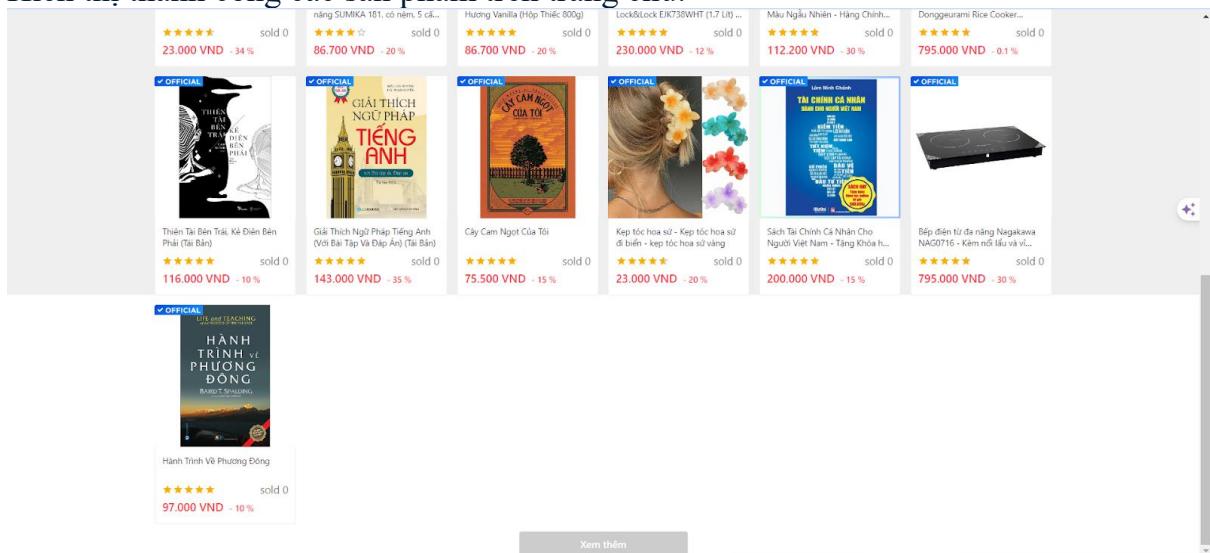
    return res
}

```

Sau đó xử lý tương tự như Xem sản phẩm.



## Hiển thị thành công các sản phẩm trên trang chủ.



### - Xem sản phẩm theo loại.

Chọn các loại sản phẩm được hiển thị trên giao diện.

Code frontend xử lý lấy sản phẩm theo loại, API là API lấy tất cả sản phẩm nhưng truyền thêm tham số type là loại sản phẩm và page để tiến hành phân trang.

```

})}

const fetchProductType = async (type, page, limit) => {
  setLoading(true)
  const res = await ProductService.getProductType(type, page, limit)
  if(res?.status === 'OK') {
    setLoading(false)
    setProducts(res?.data)
    setPanigate({...panigate, total: res?.total})
  }else {
    setLoading(false)
  }
}

```

```


export const getProductType = async (type, page, limit) => {
  if (type) {
    const res = await axios.get(
      `${process.env.REACT_APP_API_URL}/product/get-all?filter=type&filter=${type}&limit=${limit}&page=${page}`
    );
    return res.data;
  }
};

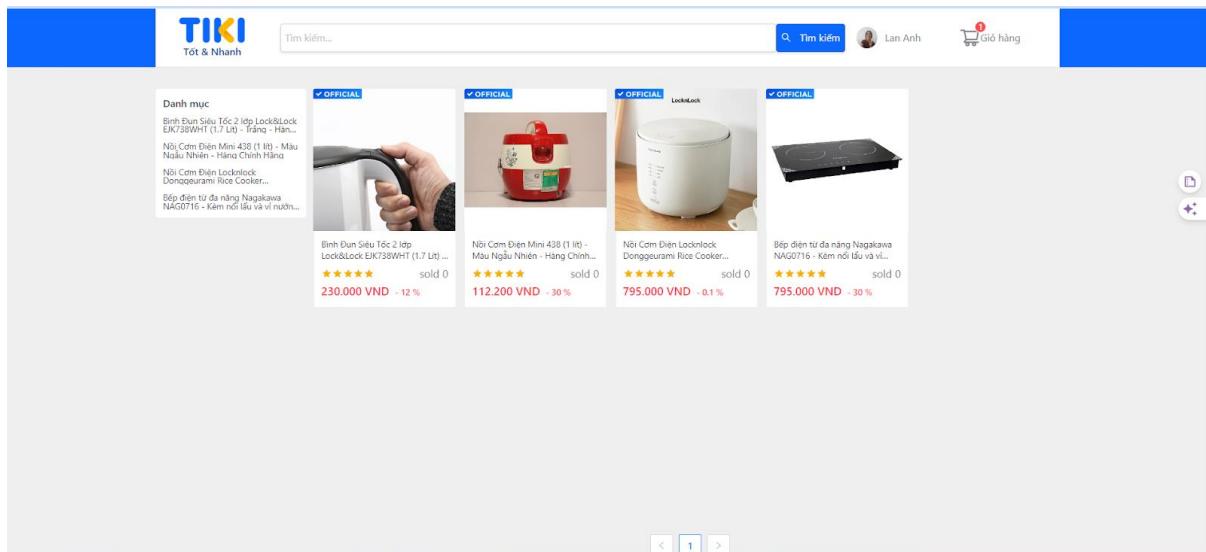

```

Code backend tương tự như “Xem sản phẩm”.  
Kết quả:

The screenshot shows the TIKI website homepage. At the top, there's a navigation bar with links like "Nhà Cửa - Đời Sống", "Nhà sách Tiki", "Phụ kiện thời trang", "Điện Gia Dụng", "Điện Thoại - Máy Tính Bảng", and "Đồ Chơi - Mẹ & Bé". A red box highlights the "Điện Gia Dụng" link. Below the navigation, there are two main promotional banners: one for "Mừng lương về Xả kho -50%" featuring a book and a skincare product, and another for "Ngoại văn giao 2h Giảm đến 30%" featuring books. Underneath these are several product cards for items like an iPhone 13, a SUMIKA ironing board, Anlene milk, a hand mixer, a rice cooker, and a rice cooker from LocknLock.

Click chuột vào loại sản phẩm “Điện Gia Dụng” để xem danh sách sản phẩm theo loại sản phẩm.

This screenshot shows the "Điện Gia Dụng" category page. The layout is identical to the homepage, with the "Điện Gia Dụng" link now highlighted in the navigation bar. The main content area displays the same promotional banners and product cards as the homepage, but the products shown are specifically from the "Điện Gia Dụng" category, such as the Anlene milk and the rice cookers.



## Tài khoản

### - Đăng nhập

Người dùng tiến hành bấm vào link Tạo tài khoản:

```
</Loading>
<p><WrapperTextLight>Quên mật khẩu?</WrapperTextLight></p>
<p>Chưa có tài khoản? <WrapperTextLight onClick={handleNavigateSignUp}> Tạo tài khoản</WrapperTextLight></p>
</WrapperContainerLeft>
```

Xử lý chuyển hướng tới trang đăng ký.

```
const handleNavigateSignUp = () => {
  navigate('/sign-up')
}
```

Người dùng sau khi điền thông tin tài khoản và bấm đăng ký:

```
const mutation = useMutationHooks(
  data => UserService.signupUser(data)
)

}

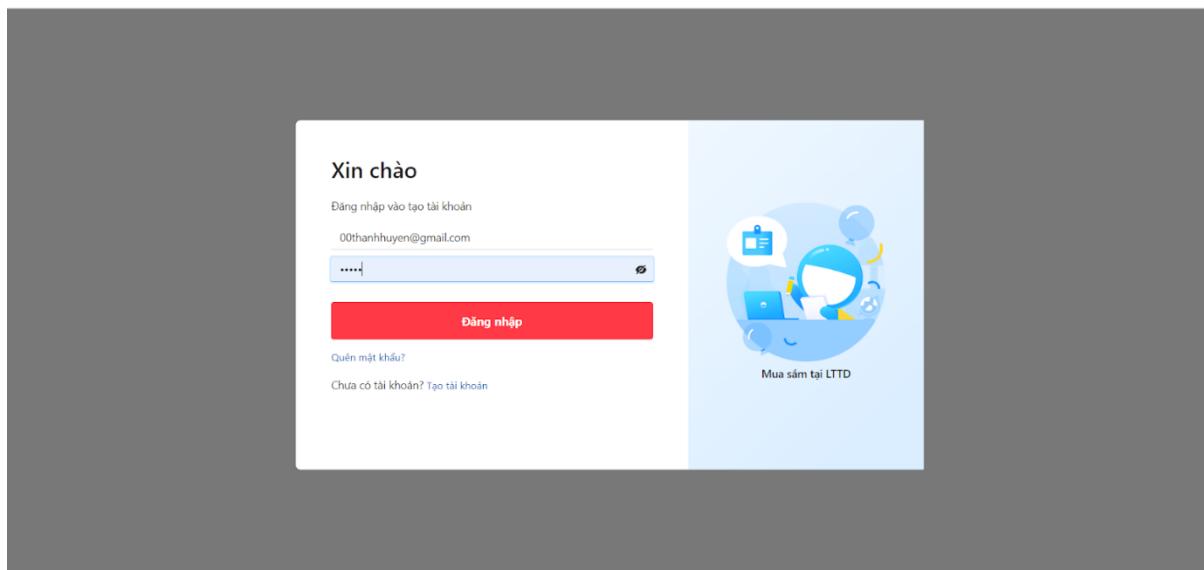
export const signupUser = async (data) => {
  const res = await axios.post(`${process.env.REACT_APP_API_URL}/user/sign-up`, data)
  return res.data
}
```

## code BE tiến hành tạo tài khoản:

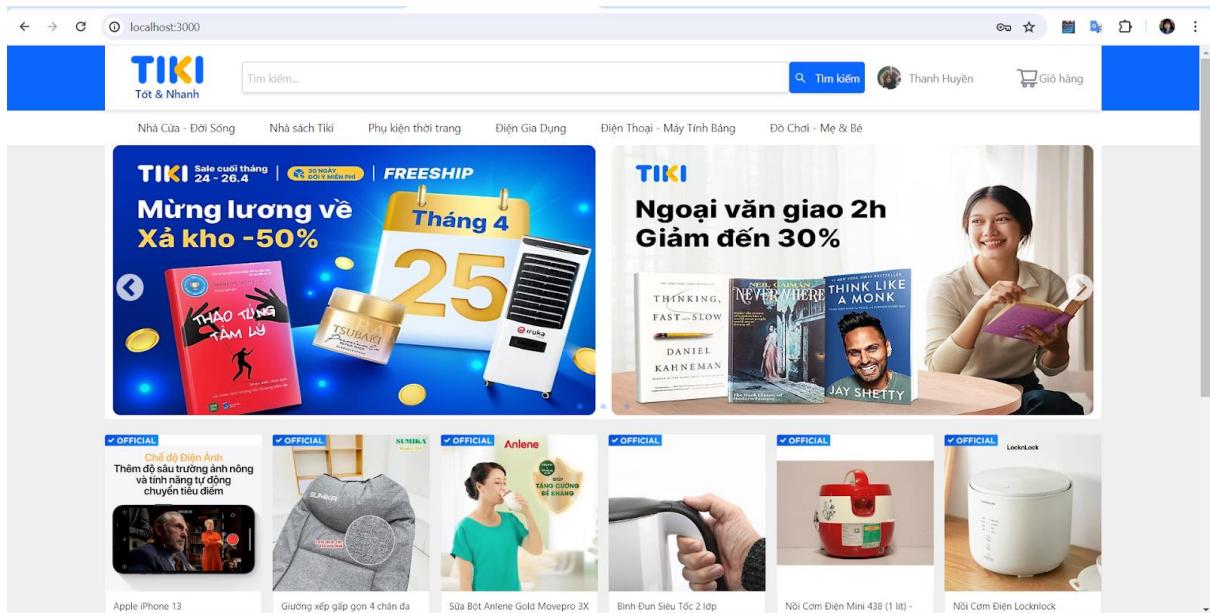
```
const createUser = (newUser) => {
  return new Promise(async (resolve, reject) => {
    const { name, email, password, confirmPassword, phone } = newUser;
    try {
      const checkUser = await User.findOne({
        email: email,
      });
      if (checkUser !== null) {
        resolve({
          status: "ERR",
          message: "The email is already",
        });
      }
      const hash = bcrypt.hashSync(password, 10);

      const createdUser = await User.create({
        name,
        email,
        password: hash,
        phone,
      });
      console.log(createdUser);
      if (createdUser) {
        resolve({
          status: "OK",
          message: "SUCCESS",
          data: createdUser,
        });
      }
    } catch (e) {
      reject(e);
    }
  });
}
```

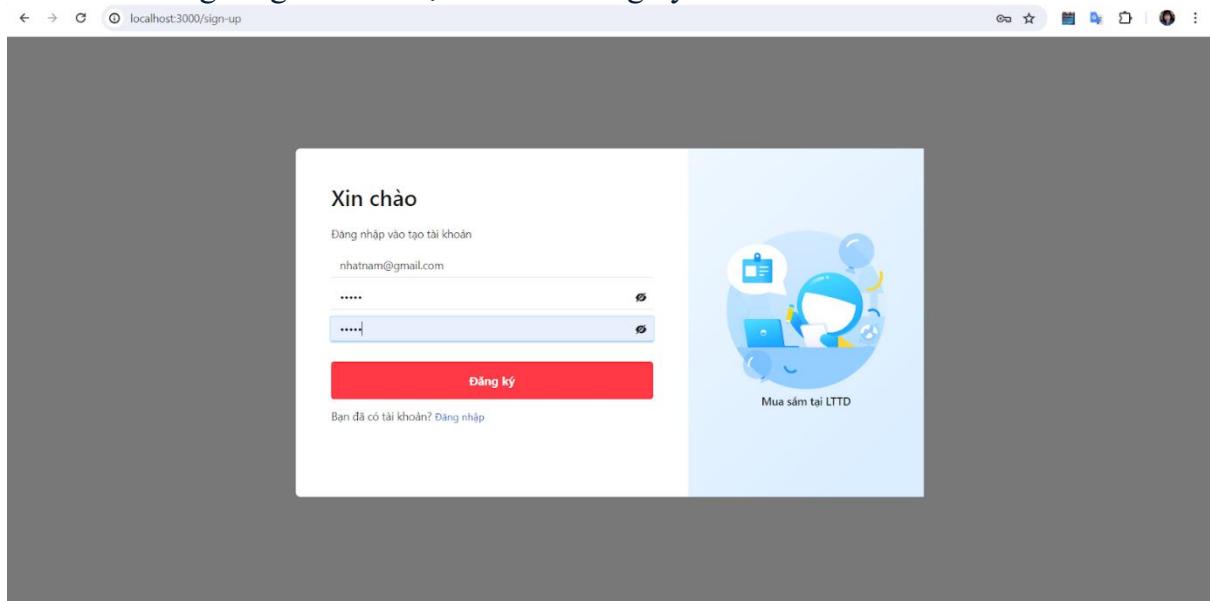
Điền gmail và mật khẩu để đăng nhập.



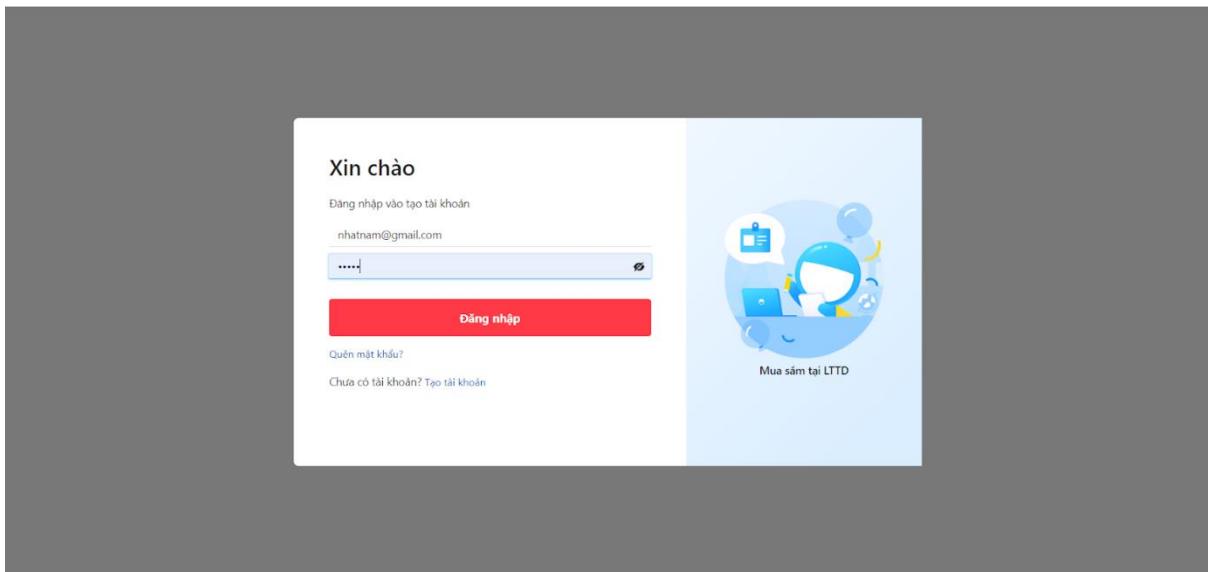
Sau khi đăng nhập thì sẽ chuyển đến trang chủ.



- Đăng ký  
Điền các thông tin gmail và mật khẩu để đăng ký.



Sau khi đăng ký thành công sẽ chuyển sang trang đăng nhập để tiến hành đăng nhập.

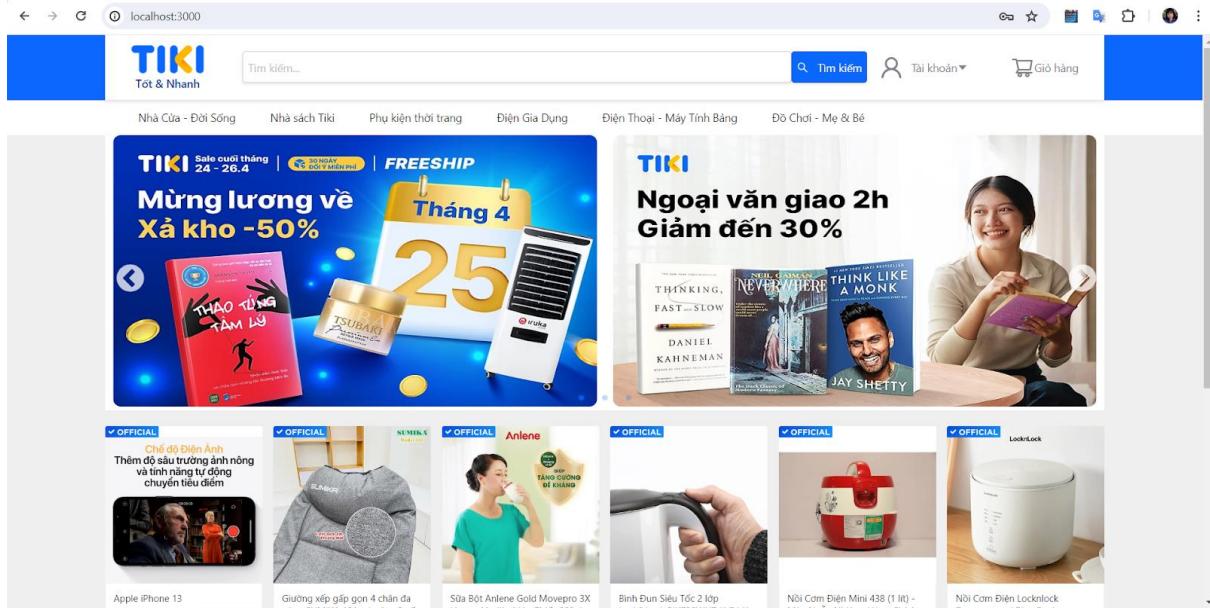


Đăng nhập thành công sẽ chuyển qua trang chủ.

## - Đăng xuất

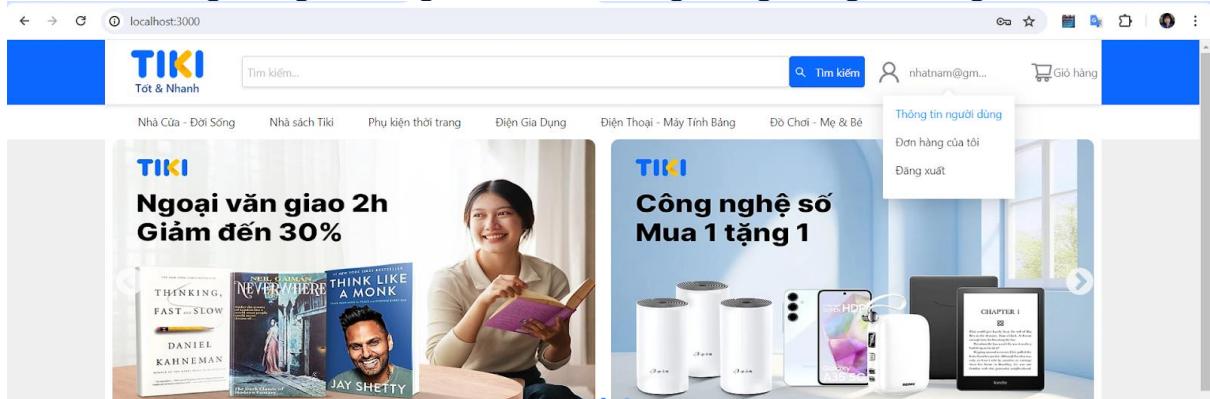
Nhấn nút “Đăng xuất” để đăng xuất.

Sau khi đăng xuất thành công sẽ chuyển qua trang đăng nhập.



### - Cập nhật thông tin tài khoản

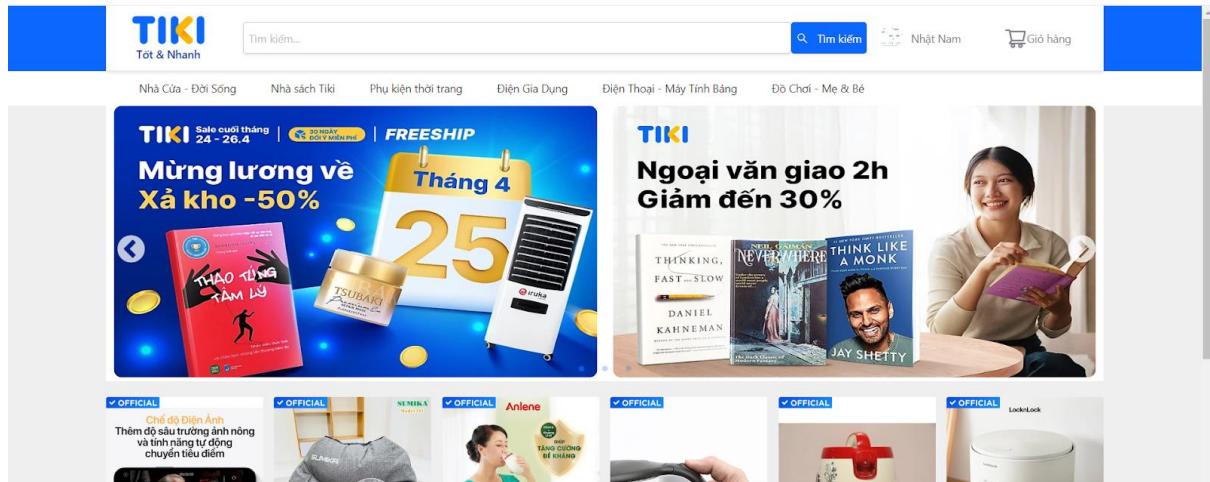
Chọn “Thông tin người dùng” để hiển thị trang thông tin người dùng.



Điền các thông tin cần cập nhật.

Name	Nhật Nam	<b>Cập nhật</b>
Email	nhatnam@gmail.com	<b>Cập nhật</b>
Phone	3454657867	<b>Cập nhật</b>
Avatar	<input type="button" value="Select File"/> 	<b>Cập nhật</b>
Address	Thành phố Hồ Chí Minh	<b>Cập nhật</b>

Đây là kết quả sau khi cập nhật thông tin tài khoản, tên tài khoản đã trở thành “Nhật Nam”



### - Đơn Hàng

### - Đặt Hàng

Chọn mua đơn hàng bằng cách đặt số lượng và nhấn nút “Chọn mua”, sản phẩm sẽ được chuyển qua giỏ hàng.

localhost:3000/product-details/663a03662bac3b643806d4cb

Cây Cam Ngot Của Tôi

75.500 VND

Giao đến Thành phố Hồ Chí Minh - Đổi địa chỉ

Thích  Chia sẻ

Số lượng  
- 2 +

**Chọn mua** Mua trả sau

Mô tả sản phẩm

"Vị chua chát của cải nghèo hòa trộn với vị ngọt ngào khi khám phá ra những điều khiến cuộc đời này đáng sống một cách kinh điển của Brazil."

- Booklist

"Một cách nhìn cuộc sống gần như hoàn chỉnh từ con mắt trẻ thơ... có sức mạnh sưởi ấm và làm tan nát cõi lòng. dù người đọc ở lứa tuổi nào."

Nhấn nút “Mua hàng”

localhost:3000/order

Địa chỉ: Thành phố Hồ Chí Minh [Thay đổi](#)

Tạm tính	151.000 VND
Giảm giá	45.300 VND
Phí giao hàng	10.000 VND
<b>Tổng tiền</b>	<b>115.700 VND</b>
(Đã bao gồm VAT nếu có)	

**Mua hàng**

Điện thông tin nhận hàng.

Cập nhật thông tin giao hàng

\* Name: Nhật Nam

\* City: Hồ Chí Minh

\* Phone: 3454657867

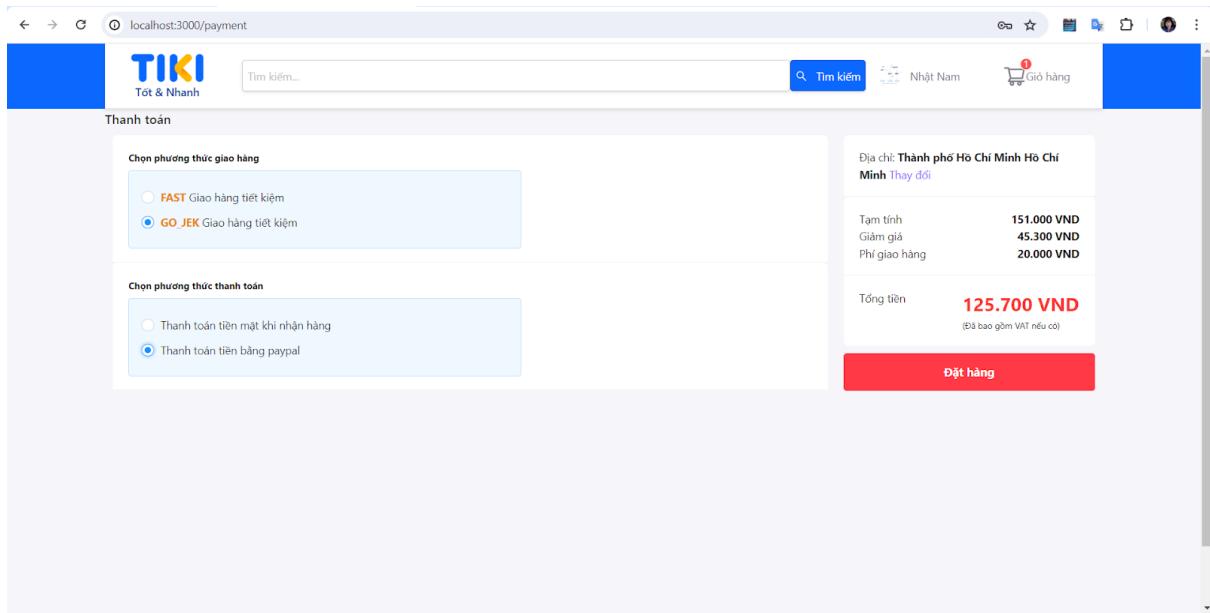
\* Address: Thành phố Hồ Chí Minh

Nhấn nút “Mua hàng”.

The screenshot shows the TIKI website at localhost:3000/order. The top navigation bar includes the TIKI logo, search bar, and language selection (Nhật Nam). The main content area displays a shopping cart with one item: "Cây Cam Ngọt Của Tôi" (Sweet Orange Tree) priced at 75,500 VND. The payment summary on the right shows a total of 115,700 VND, including shipping fees. The "Mua hàng" button is highlighted in red at the bottom right.

Tạm tính	151.000 VND
Giảm giá	45.300 VND
Phi giao hàng	10.000 VND
<b>115.700 VND</b>	
(Đã bao gồm VAT nếu có)	

Chọn phương thức giao hàng và phương thức thanh toán và nhấn nút “Đặt hàng”.



### - Hủy Đơn Hàng

Đây là danh sách đơn hàng ban đầu.  
Nhấn nút “Hủy đơn hàng” vào đơn hàng cần hủy.

A screenshot of a web browser displaying the TIKI my-order page at localhost:3000/my-order. The page lists two pending orders. The first order, for Cây Cam Ngọt Cùa Tỏi, has a total of 75,500 VND and includes a "Hủy đơn hàng" (Cancel Order) button. The second order, for Sữa Bột Anlene Gold Movepro 3X Hương... (Anlene Gold Movepro 3X Milk Powder), has a total of 86,700 VND. Both orders show a status of "Chưa giao hàng" (Not delivered) and "Chưa thanh toán" (Not paid).

Kết quả: Đơn hàng được hủy thành công.

## - Quản lý hệ thống: Người dùng

### - Xóa thông tin

Danh sách người dùng lúc đầu:

	Name	Email	Address	Admin	Phone	Action
<input type="checkbox"/>	Lan Emm	21110368@student.hcmute.edu.vn	Đák Mil, Mil Nông	TRUE	915226536	
<input type="checkbox"/>	Thanh Huyền	00thanhthanhuyen@gmail.com	Linh Xuân, Thủ Đức, Hồ Chí Minh	TRUE	376391194	
<input type="checkbox"/>	Minh Thuận	21110667@student.hcmute.edu.vn	Phường 15, Quận Tân Bình, Thành phố Hồ Chí Minh	TRUE	974817038	
<input type="checkbox"/>	Lan Anh	nguyenthilananh113388@gmail.com	484 Lê Văn Việt, Tăng Nhơn Phú A, Quận 9, TP. Thủ Đức	TRUE	384940597	
<input type="checkbox"/>	Nhật Nam	nhatnamit214@gmail.com	Số 1 VNN Phường Linh Triều	TRUE	348846016	

Chọn tài khoản và nhấn icon xóa:

The screenshot shows the TIKI management interface for user management. A modal dialog box titled "Xóa người dùng" (Delete user) is displayed, asking "Bạn có chắc chắn xóa tài khoản này không?" (Are you sure you want to delete this account?). The background table lists users with columns: Name, Email, Address, Admin status, Phone number, and Action (Delete/Edit). One row for "Lan Emm" is selected.

Name	Email	Address	Admin	Phone	Action
Lan Emm	21110368@student.hcmute.edu.vn	Đăk Mil, Mil Nông	TRUE	915226536	
Thanh Huyền	00thanhthanhuyen@gmail.com	Linh Xuân, Thủ Đức, Hồ Chí Minh	TRUE	376391194	
Minh Thuận	21110667@student.hcmute.edu.vn	Phường 15, Quận Tân Bình, Thành phố Hồ Chí Minh	TRUE	974817038	
Lan Anh	nguyenthilananh113388@gmail.com	484 Lê Văn Việt, Tăng Nhơn Phú A, Quận 9, TP. Thủ Đức	TRUE	384940597	
Nhật Nam	nhatnamit214@gmail.com	Số 1 VNN Phường Linh Triều	TRUE	348846016	

Sau khi xóa:

The screenshot shows the TIKI management interface for user management after one user has been deleted. The user "Lan Emm" is no longer present in the list. The other four users remain listed with their respective details.

Name	Email	Address	Admin	Phone	Action
Thanh Huyền	00thanhthanhuyen@gmail.com	Linh Xuân, Thủ Đức, Hồ Chí Minh	TRUE	376391194	
Minh Thuận	21110667@student.hcmute.edu.vn	Phường 15, Quận Tân Bình, Thành phố Hồ Chí Minh	TRUE	974817038	
Lan Anh	nguyenthilananh113388@gmail.com	484 Lê Văn Việt, Tăng Nhơn Phú A, Quận 9, TP. Thủ Đức	TRUE	384940597	
Nhật Nam	nhatnamit214@gmail.com	Số 1 VNN Phường Linh Triều	TRUE	348846016	

### - Chỉnh sửa thông tin:

Thông tin lúc đầu:

The screenshot shows the TIKI management interface for editing user information. The form fields are as follows:

- \* Name: Minh Thuận
- \* Email: 21110667@student.hcmute.edu.vn
- \* Admin: Yes (radio button selected)
- \* Phone: 974817038
- \* Address: Tân Bình
- \* Avatar: Select File (with a preview image of a person)

An "Apply" button is located at the bottom right of the form.

Thông tin cần chỉnh sửa:

**Chi tiết người dùng**

* Name:	Minh Thuận
* Email:	21110667@student.hcmute.edu.vn
* Admin:	<input checked="" type="radio"/> Yes <input type="radio"/> No
* Phone:	974817038
* Address:	Phường 15, Quận Tân Bình, Thành phố Hồ Chí Minh
* Avatar:	Select File 

**Apply**

Thông tin đã cập nhật sau khi chỉnh sửa:



**Quản lý người dùng**

<b>Quản lý người dùng</b>					
<input type="button" value="Export Excel"/> Name <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Email <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Address <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Admin <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Phone <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Action					
<input type="checkbox"/>	Minh Thuận	21110667@student.hcmute.edu.vn	Phường 15, Quận Tân Bình, Thành phố Hồ Chí Minh	TRUE	974817038 

### - Tìm kiếm người dùng

Nhập địa chỉ Email cần tìm kiếm:



**Quản lý người dùng**

<b>Quản lý người dùng</b>					
<input type="button" value="Export Excel"/> Name <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Email <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Address <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Admin <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Phone <input type="text"/> <input type="button" value="Search"/> <input type="button" value="Reset"/> Action					
<input type="checkbox"/>	Lan Emm	21110368@...	Dak Mil, Mil Nông	TRUE	915226536 
<input type="checkbox"/>	Thanh Huyền	00thanhthanhuyen@gmail.com	Linh Xuân, Thủ Đức, Hồ Chí Minh	TRUE	376391194 
<input type="checkbox"/>	Minh Thuận	21110667@student.hcmute.edu.vn	Phường 15, Quận Tân Bình, Thành phố Hồ Chí Minh	TRUE	974817038 
<input type="checkbox"/>	Lan Anh	nguyenthilanhanh113388@gmail.com	484 Lê Văn Việt, Tăng Nhơn Phú A, Quận 9, TP. Thủ Đức	TRUE	384940597 
<input type="checkbox"/>	Nhật Nam	nhatnamit214@gmail.com	Số 1 VNN Phường Linh Triệu	TRUE	348846016 

Kết quả tìm kiếm:

Name	Email	Address	Admin	Phone	Action
Minh Thuận	21110667@student.hcmute.edu.vn	Tân Bình	TRUE	974817038	

- Xem danh sách thông tin người dùng:

Bấm vào Người dùng để xem danh sách người dùng:

Name	Email	Address	Admin	Phone	Action
Lan Emm	21110368@student.hcmute.edu.vn	Đák Mil, Mil Nông	TRUE	915226536	
Thanh Huyền	00thanhthanhhuyen@gmail.com	Linh Xuân, Thủ Đức, Hồ Chí Minh	TRUE	376391194	
Minh Thuận	21110667@student.hcmute.edu.vn	Tân Bình	TRUE	974817038	
Lan Anh	nguyenthanhlananh11338@gmail.com	484 Lê Văn Việt, Tăng Nhơn Phú A, Quận 9, TP. Thủ Đức	TRUE	384940597	
Nhật Nam	nhatnamit214@gmail.com	Số 1 VNN Phường Linh Triều	TRUE	348846016	

- Quản lý hệ thống: Sản phẩm

- Xem danh sách thông tin sản phẩm

Đây là danh sách sản phẩm có trong hệ thống.

Xóa tất cả

[Export Excel](#)

<input type="checkbox"/> Name	Price	Rating	Type	Action
<input type="checkbox"/> Thiên Tài Bên Trái, Kè Điện Bên Phải (Tái Bản)	116000	5	Nhà sách Tiki	
<input type="checkbox"/> Giải Thích Ngữ Pháp Tiếng Anh (Với Bài Tập Và Đáp Án) (Tái Bản)	143000	5	Nhà sách Tiki	
<input type="checkbox"/> Cây Cam Ngọt Của Tôi	75500	4.9	Nhà sách Tiki	
<input type="checkbox"/> Kẹp tóc hoa sú - Kẹp tóc hoa sú đi biển - kẹp tóc hoa sú vàng	23000	4.5	Phụ kiện thời trang	
<input type="checkbox"/> Sách Tài Chính Cá Nhân Cho Người Việt Nam - Tặng Khóa học Online về Tài chính	200000	4.9	Nhà sách Tiki	
<input type="checkbox"/> Bếp điện từ đa năng Nagakawa NAG0716 - Kèm nồi lẩu và vỉ nướng chuyên dụng - Hàng chính hãng	795000	4.7	Điện Gia Dụng	
<input type="checkbox"/> Hành Trình Về Phương Đông	97000	5	Nhà sách Tiki	

[<>](#) [\*\*1\*\*](#) [<<](#)

## - Thêm sản phẩm

Nhấn dấu “+” để thêm sản phẩm.

localhost:3000/system/admin

**TIKI**  
Tốt & Nhanh

Thanh Huyền

[Người dùng](#)

[Sản phẩm](#) **(Current)**

[Đơn hàng](#)

Quản lý sản phẩm

[Export Excel](#)

<input type="checkbox"/> Name	Price	Rating	Type	Action
<input type="checkbox"/> Hành Trình Về Phương Đông	97000	5	Nhà sách Tiki	

[<>](#) [\*\*1\*\*](#) [<<](#)

Điền các thông tin sản phẩm và nhấn nút “Submit”

The screenshot shows the Tiki Admin interface at [localhost:3000/system/admin](http://localhost:3000/system/admin). On the left, there's a sidebar with 'Người dùng', 'Sản phẩm' (selected), and 'Đơn hàng'. The main area has a title 'Quản lý sản phẩm' with a large '+' button. A modal window titled 'Tạo sản phẩm' is open, containing fields for Name (Sách Tài Chính Cá Nhân Cho Người Việt Nam - Tặng), Type (Nhà sách Tiki), Count inStock (100), Price (200000), Description (describing a financial course), Rating (4.9), Discount (15), and Image (two small thumbnail images). Below the modal is a table listing products with columns for Name, Price, Rating, Type, and Action (edit/delete).

	Price	Rating	Type	Action
795000	4.7	Điện Gia Dụng		
97000	5	Nhà sách Tiki		

Kết quả: Sản phẩm được thêm thành công và hiện trong danh sách sản phẩm.

The screenshot shows the Tiki Admin interface at [localhost:3000/system/admin](http://localhost:3000/system/admin). The sidebar and main area are similar to the previous screenshot, but the table now shows three products listed: 'Sách Tài Chính Cá Nhân Cho Người Việt Nam - Tặng' (Price 200000, Rating 4.9, Type Nhà sách Tiki), 'Bếp điện từ đa năng Nagakawa NAG0716 - Kèm nồi lẩu và vỉ nướng chuyên dụng - Hàng chính hãng' (Price 795000, Rating 4.7, Type Điện Gia Dụng), and 'Hành Trình Về Phương Đông' (Price 97000, Rating 5, Type Nhà sách Tiki). The table includes columns for Name, Price, Rating, Type, and Action (edit/delete).

### - Tìm kiếm sản phẩm

Nhấn icon tìm kiếm và nhập nội dung cần tìm, nhấn nút “Search”

The screenshot shows the TIKI Admin product management interface at localhost:3000/system/admin. The left sidebar has tabs for Người dùng, Sản phẩm (selected), and Đơn hàng. The main area is titled 'Quản lý sản phẩm' with a large '+' button. A search bar contains 'sữa'. The results table includes columns for Name, Price, Rating, Type, and Action (with icons for delete and edit). One item is selected: 'Sữa Bột Anlene Gold Movepro 3X Hương Vanilla (Hộp Thiếc 800g)'.

Name	Price	Rating	Type	Action
Sữa Bột Anlene Gold Movepro 3X Hương Vanilla (Hộp Thiếc 800g)	195000	4.8	Đồ Chơi - Mẹ & Bé	
Thiên Tài Bên Trái, Ké Điện Bên Phải (Tái Bản)	116000	5	Nhà sách Tiki	
Giải Thích Ngữ Pháp Tiếng Anh (Với Bài Tập Và Đáp Án) (Tái Bản)	143000	5	Nhà sách Tiki	
Cây Cam Ngọt Của Tôi	75500	4.9	Nhà sách Tiki	
Kẹp tóc hoa sú - Kẹp tóc hoa sú di biến - kẹp tóc hoa sú vàng	23000	4.5	Phụ kiện thời trang	

Kết quả: Tìm kiếm được sản phẩm cần tìm.

The screenshot shows the TIKI Admin product management interface at localhost:3000/system/admin. The left sidebar has tabs for Người dùng, Sản phẩm (selected), and Đơn hàng. The main area is titled 'Quản lý sản phẩm' with a large '+' button. The results table shows one item: 'Sữa Bột Anlene Gold Movepro 3X Hương Vanilla (Hộp Thiếc 800g)'.

Name	Price	Rating	Type	Action
Sữa Bột Anlene Gold Movepro 3X Hương Vanilla (Hộp Thiếc 800g)	195000	4.8	Đồ Chơi - Mẹ & Bé	

### - Chính sửa sản phẩm

Đây là sản phẩm trước khi chỉnh sửa có giá 53000.  
Nhấn icon chỉnh sửa để chỉnh sửa.

The screenshot shows the TIKI Admin product management interface at localhost:3000/system/admin. The left sidebar has tabs for Người dùng, Sản phẩm (selected), and Đơn hàng. The main area is titled 'Quản lý sản phẩm' with a large '+' button. The results table shows one item: 'Đắc Nhân Tâm (Bìa Mềm)(Tái Bản)'. The 'Edit' icon for this item is highlighted.

Name	Price	Rating	Type	Action
Đắc Nhân Tâm (Bìa Mềm)(Tái Bản)	53000	4.8	Nhà sách Tiki	

Chỉnh sửa giá sản phẩm thành 105000 và nhấn nút “Apply”.

Chi tiết sản phẩm

* Name:	Đắc Nhân Tâm (Bìa Mềm)(Tái Bản)
* Type:	Nhà sách Tiki
* Count inStock:	30
* Price:	105000đ
* Description:	Đắc nhân tâm của Dale Carnegie là quyển sách nổi tiếng nhất, bán chạy nhất và có tầm ảnh hưởng nhất của mọi thời đại. Tác phẩm đã được chuyển ngữ sang hầu hết các thứ tiếng trên t
* Rating:	4.8
* Discount:	38
* Image:	<input type="button" value="Select File"/> <input type="button" value="Reset"/> 
<input type="button" value="Apply"/>	

Kết quả: Giá của sản phẩm đã được sửa thành 105000.

<input type="checkbox"/> Đắc Nhân Tâm (Bìa Mềm)(Tái Bản)	1050000	4.8	Nhà sách Tiki	
--	---------	-----	---------------	---

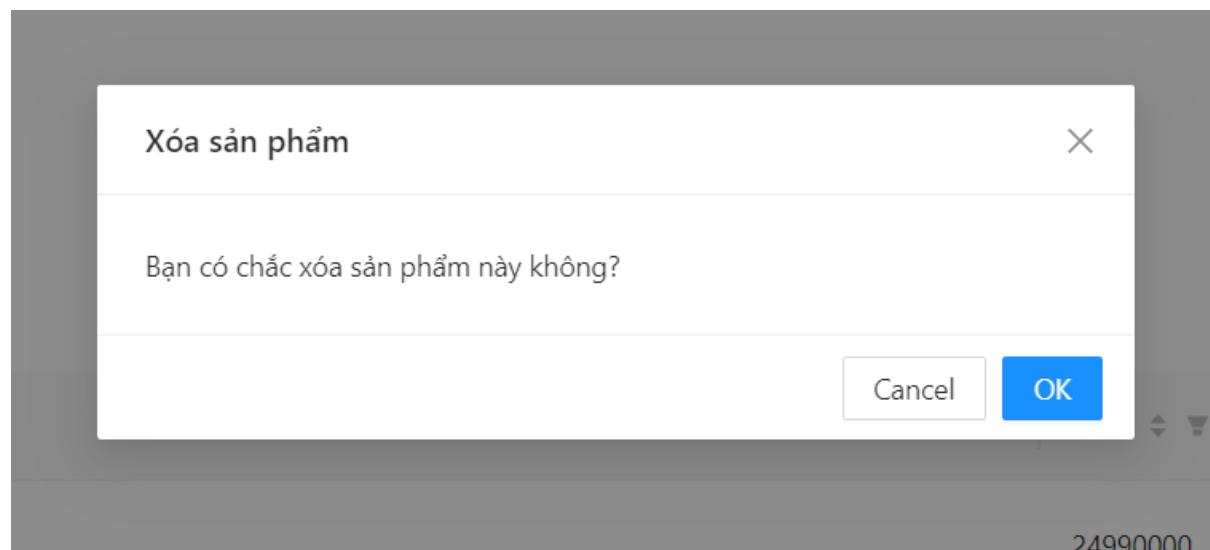
### - Xóa một sản phẩm

Đây là sản phẩm trước khi xóa

Chọn một sản phẩm để xóa là “MacBook Air M2 2022” và nhấn icon xóa.

<input type="checkbox"/> Name	Price	Rating	Type	Action
<input type="checkbox"/> MacBook Air M2 2022	24990000	4.7	Laptop - Máy Vi Tính - Linh kiện	
<input type="checkbox"/> Đắc Nhân Tâm (Bìa Mềm)(Tái Bản)	1050000	4.8	Nhà sách Tiki	
<input type="checkbox"/> Sữa Bột Anlene Gold Movepro 3X Hương Vanilla (Hộp Thiếc 800g)	195000	4.8	Đồ Chơi - Mẹ & Bé	
<input type="checkbox"/> Thiên Tài Bên Trái, Kẻ Điện Bên Phải (Tái Bản)	116000	5	Nhà sách Tiki	

Xác nhận xóa bằng cách nhấn nút “OK”



Kết quả: Sản phẩm bị xóa thành công và không còn hiện trong danh sách sản phẩm.

<input type="checkbox"/>	Hiếu Về Trump	186000	5	Nhà sách Tiki	
<input type="checkbox"/>	Đắc Nhân Tâm (Bìa Mềm)(Tái Bản)	1050000	4.8	Nhà sách Tiki	
<input type="checkbox"/>	Sữa Bột Anlene Gold Movepro 3X Hương Vanilla (Hộp Thiếc 800g)	195000	4.8	Đồ Chơi - Mẹ & Bé	
<input type="checkbox"/>	Thiên Tài Bên Trái, Kè Điện Bên Phải (Tái Bản)	116000	5	Nhà sách Tiki	

### - Xóa nhiều sản phẩm

Đây là danh sách sản phẩm trước khi xóa.

Nhấn tích chọn các sản phẩm cần xóa và chọn “Xóa tất cả”.

Xóa tất cả					
<input type="button" value="Export Excel"/> <input checked="" type="checkbox"/> Name <span style="float: right;">Price Rating Type Action</span>					
<input checked="" type="checkbox"/>	Hiếu Về Trump	186000	5	Nhà sách Tiki	
<input checked="" type="checkbox"/>	Đắc Nhân Tâm (Bìa Mềm)(Tái Bản)	1050000	4.8	Nhà sách Tiki	
<input checked="" type="checkbox"/>	Sữa Bột Anlene Gold Movepro 3X Hương Vanilla (Hộp Thiếc 800g)	195000	4.8	Đồ Chơi - Mẹ & Bé	
<input type="checkbox"/>	Thiên Tài Bên Trái, Kè Điện Bên Phải (Tái Bản)	116000	5	Nhà sách Tiki	
<input type="checkbox"/>	Giải Thích Ngữ Pháp Tiếng Anh (Với Bài Tập Và Đáp Án) (Tái Bản)	143000	5	Nhà sách Tiki	
<input type="checkbox"/>	Cây Cam Ngọt Của Tôi	75500	4.9	Nhà sách Tiki	
<input type="checkbox"/>	Kẹp tóc hoa sứ - Kẹp tóc hoa sứ đi biển - kẹp tóc hoa sứ vàng	23000	4.5	Phụ kiện thời trang	

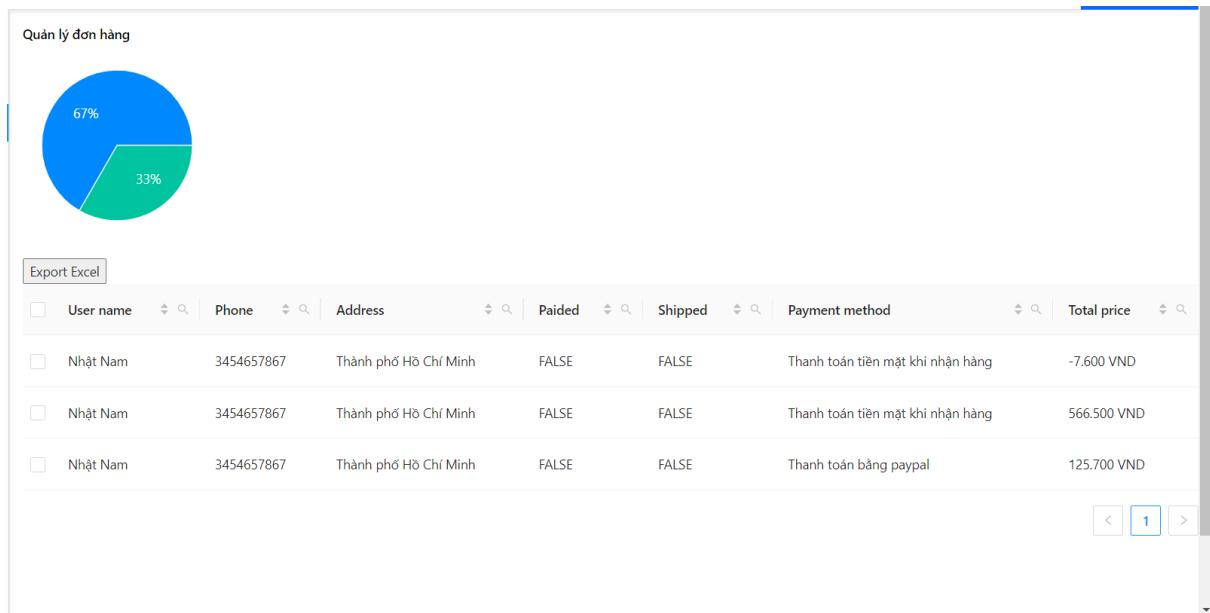
Kết quả: Xóa thành công các sản phẩm đã tích chọn và không còn hiện trong danh sách sản phẩm.

Xóa tất cả					
<input type="button" value="Export Excel"/> <input type="checkbox"/> Name <span style="float: right;">Price Rating Type Action</span>					
<input type="checkbox"/>	Thiên Tài Bên Trái, Kè Điện Bên Phải (Tái Bản)	116000	5	Nhà sách Tiki	
<input type="checkbox"/>	Giải Thích Ngữ Pháp Tiếng Anh (Với Bài Tập Và Đáp Án) (Tái Bản)	143000	5	Nhà sách Tiki	
<input type="checkbox"/>	Cây Cam Ngọt Của Tôi	75500	4.9	Nhà sách Tiki	
<input type="checkbox"/>	Kẹp tóc hoa sứ - Kẹp tóc hoa sứ đi biển - kẹp tóc hoa sứ vàng	23000	4.5	Phụ kiện thời trang	
<input type="checkbox"/>	Sách Tài Chính Cá Nhân Cho Người Việt Nam - Tặng Khóa học Online về Tài chính	200000	4.9	Nhà sách Tiki	

### - Quản lý hệ thống: Đơn hàng

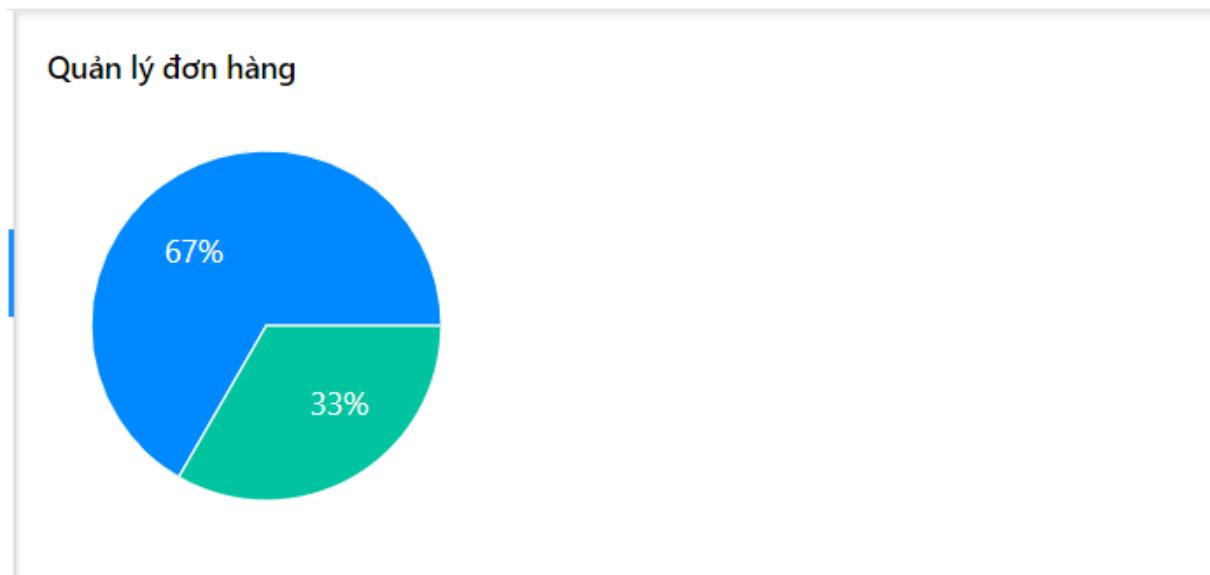
#### - Xem đơn hàng

Giao diện hiển thị danh sách các đơn hàng mà người dùng đã đặt.



### - Xem báo cáo đơn hàng

Giao diện hiển thị sơ đồ đơn hàng: Xanh dương là phần trăm thanh toán bằng tiền mặt, màu xanh lá cây là phần trăm thanh toán bằng paypal.



## TÀI LIỆU THAM KHẢO

[1] Tutorialspoint, truy cập ngày 19/02/2024,

[https://www.tutorialspoint.com/mongodb/mongodb\\_overview.html](https://www.tutorialspoint.com/mongodb/mongodb_overview.html)

[2] Stackjava , “Hướng dẫn cấu hình MongoDB trên Win 10”, truy cập ngày

19/02/2024, <https://stackjava.com/mongodb/huong-dan-cai-dat-cau-hinh-mongodb-tren-windows-10.html>

[3] Pratibha Kumari J, “What is a NoSQL database?”, truy cập ngày 20/02/2024,

<https://www.linkedin.com/pulse/what-nosql-database-pratibha-kumari-jha>

[4] David Taylor, “NoSQL Tutorial: What is, Types of NoSQL Databases & Example”, truy cập ngày 20/02/2024, <https://www.guru99.com/nosql-tutorial.html>

[5] Yousaf Khan, “What is data modeling in MongoDB?”, educative, truy cập ngày 26/4/2024, <https://www.educative.io/answers/what-is-data-modeling-in-mongodb>

[6] Swetha Doddamani, “What is MongoDB?”, intellipaat, truy cập ngày 23/02/2024,

<https://intellipaat.com/blog/what-is-mongodb/>