

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỀ TÀI MÔN HỌC

Cơ sở dữ liệu NoSQL

**Đề tài 7: Tìm hiểu chức năng của GUI Tool NoSQL Booster.
Thực hiện quản trị cơ sở dữ liệu “Quản lý dịch vụ xe khách”**

TP. HỒ CHÍ MINH, tháng 09 năm 2024

BỘ CÔNG THƯƠNG

TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG TP. HCM

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỀ TÀI MÔN HỌC

Cơ sở dữ liệu NoSQL

**Đề tài 7: Tìm hiểu chức năng của GUI Tool NoSQL Booster.
Thực hiện quản trị cơ sở dữ liệu “Quản lý dịch vụ xe khách”**

GVHD: Nguyễn Thị Định

Nhóm 5 gồm:

2001216203 – Hà Văn Thy

2001210864 – Bùi Hoàng Khôi

2001215870 – Nguyễn Quốc Khánh

TP. HỒ CHÍ MINH, tháng 09 năm 2024

LỜI CAM ĐOAN

Nhóm 5 xin cam đoan rằng báo cáo này là kết quả của quá trình nghiên cứu và thực hiện của nhóm 5. Tất cả các số liệu, kết quả nêu trong báo cáo đều trung thực và chưa từng được công bố trong bất kỳ tài liệu nào khác. Nhóm 5 cũng xin cam đoan rằng mọi sự giúp đỡ trong quá trình thực hiện đề tài đã được ghi nhận và các nguồn tài liệu tham khảo đều được trích dẫn đầy đủ, rõ ràng theo đúng quy định.

Nếu có bất kỳ sai sót nào, nhóm 5 xin hoàn toàn chịu trách nhiệm.

Nhóm sinh viên thực hiện

(Ký và ghi rõ họ tên)

LỜI CẢM ƠN

Trong quá trình thực hiện bài báo cáo này, nhóm em đã nhận được sự hỗ trợ và hướng dẫn quý báu từ nhiều cá nhân. Nhóm em xin chân thành bày tỏ lòng biết ơn sâu sắc đến tất cả những người đã đóng góp vào sự thành công của nghiên cứu này.

Trước hết, nhóm em xin gửi lời cảm ơn chân thành và sâu sắc đến cô Nguyễn Thị Định, giảng viên hướng dẫn của nhóm em. Sự tận tâm, kiến thức uyên bác và những lời khuyên quý giá của cô đã giúp nhóm em định hướng nghiên cứu, vượt qua những khó khăn và hoàn thiện bài báo cáo này. Không có sự hướng dẫn tận tình của cô, bài báo cáo này sẽ không thể đạt được kết quả như mong đợi.

Nhóm em cũng xin cảm ơn các thầy cô trong khoa đã trang bị cho nhóm em nền tảng kiến thức vững chắc, tạo điều kiện thuận lợi cho nhóm em trong quá trình học tập.

Cuối cùng, nhóm em xin gửi lời cảm ơn đến cô và bạn bè đã luôn bên cạnh, động viên và tạo điều kiện tốt nhất để nhóm em có thể tập trung vào việc học tập và nghiên cứu.

Mặc dù đã có nhiều cố gắng, bài báo cáo này không thể tránh khỏi những thiếu sót. Nhóm em rất mong nhận được những ý kiến đóng góp để có thể hoàn thiện hơn trong tương lai.

Xin chân thành cảm ơn!

MỤC LỤC

LỜI CAM ĐOAN	iii
LỜI CẢM ƠN	iv
MỤC LỤC	v
DANH MỤC CÁC TỪ VIẾT TẮT	viii
DANH MỤC CÁC BẢNG	ix
DANH MỤC CÁC BIỂU ĐỒ, ĐỒ THỊ, SƠ ĐỒ, HÌNH ẢNH	x
MỞ ĐẦU	1
• Giới thiệu đề tài	1
• Lý do chọn đề tài	1
• Mục tiêu và phạm vi nghiên cứu	2
• Ý nghĩa của nghiên cứu	2
Chương 1 TỔNG QUAN VỀ NOSQLBOOSTER VÀ CƠ SỞ DỮ LIỆU MONGODB 4	
1.1. Giới thiệu về NoSQLBooster	4
1.1.1. Định nghĩa	4
1.1.2. Các tính năng chính	4
1.1.3. So sánh với các công cụ khác	4
1.2. Tổng quan về MongoDB	5

1.2.1.	Khái niệm cơ bản về MongoDB	5
1.2.2.	Ưu điểm và nhược điểm của MongoDB.....	7
1.2.3.	Các phương pháp trong quản trị MongoDB:.....	7
1.2.4.	So sánh MongoDB với các CSDL quan hệ	9
Chương 2	THIẾT KẾ HỆ THỐNG	9
2.1	Phân tích yêu cầu hệ thống.....	9
2.2	Thiết kế sơ đồ ERD	15
2.3	Xác định cấu trúc các collection	15
2.4	Thiết kế cơ sở dữ liệu "Quản lý dịch vụ xe khách"	18
Chương 3	TRIỂN KHAI HỆ THỐNG	19
3.1	Cài đặt và cấu hình NoSQLBooster	19
3.1.1	Cài đặt.....	19
3.1.2	Cấu hình.....	19
3.2	Tạo cơ sở dữ liệu và các collection.....	21
3.3	Thực hiện các thao tác CRUD.....	24
3.4	Viết truy vấn phức tạp bằng aggregation	26
3.5	Thực hiện import/export dữ liệu	30
3.5.1	Import dữ liệu	30
3.5.2	Export dữ liệu	32
Chương 4	QUẢN TRỊ VÀ TỐI ƯU HỆ THỐNG.....	34

4.1	Thiết lập user và phân quyền	34
4.2	Tạo lịch backup tự động.....	36
KẾT LUẬN VÀ KIẾN NGHỊ		47
•	Đánh giá kết quả đạt được	47
•	Hướng phát triển	47
TÀI LIỆU THAM KHẢO		48
PHỤ LỤC.....		49

DANH MỤC CÁC TỪ VIẾT TẮT

1. CSDL: Cơ sở dữ liệu
2. CRUD: Create, Read, Update, Delete (Tạo, Đọc, Cập nhật, Xóa)
3. ERD: Entity Relationship Diagram (Sơ đồ mối quan hệ thực thể)
4. JSON: JavaScript Object Notation
5. CSV: Comma-Separated Values
6. ID: Identifier (Mã định danh)
7. RDBMS: Relational Database Management System (Hệ quản trị cơ sở dữ liệu quan hệ)
8. GUI: Graphical User Interface (Giao diện đồ họa người dùng)
9. NoSQL: Not Only SQL

DANH MỤC CÁC BẢNG

1. Bảng so sánh NoSQLBooster với các công cụ khác

DANH MỤC CÁC BIỂU ĐỒ, ĐỒ THỊ, SƠ ĐỒ, HÌNH ẢNH

1. Hình ảnh giao diện NoSQLBooster (Chương 3, mục 3.1.1)
2. Sơ đồ ERD của hệ thống (Chương 2, mục 2.2)
3. Hình ảnh giao diện đăng nhập phần mềm (Chương 5, mục 5.1.a)
4. Hình ảnh giao diện trang chủ (Chương 5, mục 5.1.b)
5. Hình ảnh giao diện chức năng quản lý chuyến đi (Chương 5, mục 5.1.c)
6. Hình ảnh giao diện chức năng quản lý xe (Chương 5, mục 5.1.d)
7. Hình ảnh giao diện chức năng quản lý vé (Chương 5, mục 5.1.e)
8. Hình ảnh giao diện chức năng quản lý tài khoản (Chương 5, mục 5.1.f)

BẢNG PHÂN CHIA CÔNG VIỆC

Họ Tên	Công Việc	Tuần
Thy	<ul style="list-style-type: none"> - Tìm hiểu và demo NoSQL Booster. Kết nối CSDL - Import/export dữ liệu. - Viết truy vấn CRUD cơ bản, demo CRUD - Nhập dữ liệu mẫu (5+ bản ghi/collection) - Tạo CSDL "Quản lý dịch vụ xe khách" - Slide, Word 	
Khánh	<ul style="list-style-type: none"> - Phân tích yêu cầu hệ thống xe khách - Thiết kế sơ đồ ERD - Xác định cấu trúc các collection - Tạo CSDL "Quản lý dịch vụ xe khách" - Viết truy vấn phức tạp (aggregation) - Slide, Word 	
Khôi	<ul style="list-style-type: none"> - Khảo sát đề tài - Tìm hiểu quản trị MongoDB - Tạo lịch backup tự động - Thiết lập user và phân quyền - Viết quy trình vận hành CSDL - Slide, Word 	

MỞ ĐẦU

• Giới thiệu đề tài

Ở thời đại ngày nay với sự phát triển vượt trội của nền kinh tế - xã hội thế giới, nhu cầu đi lại của con người không ngừng tăng cao. Vì thế, đã có một số phương pháp được đưa ra nhằm cải thiện nhu cầu của người dân. Đặc biệt nổi trội trong các phương pháp được đưa ra đó chính là các dịch vụ vận tải hành khách và xe khách đóng vai trò vô cùng quan trọng và đó cũng chính là đáp án mà con người đang tìm. Tuy nhiên, việc quản lý thủ công có vẻ là một vấn đề vô cùng khó khăn bởi vì khi nhiều người sử dụng dịch vụ thì với các phương pháp thủ công khó có thể quản lý lượng thông tin khổng lồ như thế và với quá nhiều thông tin sẽ dễ dẫn tới sai sót. Điều này giảm mạnh hiệu quả về kinh doanh và ảnh hưởng trực tiếp đến trải nghiệm của hành khách. Để giải quyết vấn đề này, nhóm chúng em chọn đề tài “Quản lý dịch vụ xe khách” nhằm xây dựng một hệ thống hỗ trợ cho việc quản lý các hoạt động liên quan đến xe khách. Với đề tài này nhóm chúng em mong muốn mang lại một giải pháp thiết thực và hiệu quả, không chỉ để giải quyết các khó khăn trước mắt của người dân mà còn mở ra nhiều cơ hội phát triển cho chúng ta trong tương lai.

• Lý do chọn đề tài

- Nâng cao trải nghiệm, cải thiện nhu cầu của người dùng
- Tối ưu hóa được việc quản lý các chuyến đi và từng loại xe đang sử dụng
- Giảm thiểu rủi ro trong quá trình vận hành
- Đáp ứng được nhu cầu của thị trường về vận tải

- **Mục tiêu và phạm vi nghiên cứu**

- Mục tiêu chính của đề tài nhằm khám phá và phân tích các chức năng của GUI Tool NoSQL Bootser đồng thời áp dụng nó vào việc quản trị cơ sở dữ liệu “Quản lý dịch vụ xe khách” bao gồm các chức năng chính và cách thức hoạt động của nó trong môi trường MongoDB.
- Phạm vi nghiên cứu bao gồm việc đánh giá hiệu quả của NoSQL Booster trong các tác vụ quản lý, truy vấn và tối ưu hóa cơ sở dữ liệu NoSQL cũng như áp dụng tool để thiết kế và xây dựng một cơ sở dữ liệu cụ thể cho hệ thống “Quản lý dịch vụ xe khách” và thực hiện các thao tác quản trị từ cơ bản đến nâng cao.
- Thông qua việc kết hợp giữa nghiên cứu lý thuyết và áp dụng thực tế, bài tiểu luận này sẽ cung cấp cái nhìn toàn diện về tiềm năng của NoSQL Bootster trong việc nâng cao hiệu quả quản trị cơ sở dữ liệu MongoDB đặc biệt trong lĩnh vực quản lý giao thông vận tải.

- **Ý nghĩa của nghiên cứu**

- Nghiên cứu mang lại nhiều ý nghĩa quan trọng trong lĩnh vực quản trị cơ sở dữ liệu NoSQL, đặc biệt là MongoDB. Giúp hiểu rõ hơn về công cụ NoSQL Bootster và giúp các nhà phát triển quản trị viên hiểu rõ hơn về tiềm năng và khả năng của công cụ trong việc tối ưu hóa quá trình quản lý dữ liệu.
- Bằng việc áp dụng NoSQL Booster vào phần mềm “Quản lý dịch vụ xe khách” nghiên cứu sẽ minh họa cách thức mà doanh nghiệp có thể tận dụng công cụ này để nâng cao hiệu quả quản lý và khai thác dữ liệu của họ giúp cải thiện quy trình làm việc, tăng năng suất và giảm chi phí vận hành.
- Từ góc độ học thuật, nghiên cứu này đóng góp vào kiến thức chung về quản trị cơ sở dữ liệu NoSQL, cung cấp một trường hợp nghiên cứu chi tiết có thể được sử dụng

- trong giảng dạy và nghiên cứu tiếp theo. Nó cũng mở ra hướng nghiên cứu mới về cách các công cụ GUI có thể cải thiện hiệu quả quản lý cơ sở dữ liệu phi quan hệ.
- Cuối cùng, bằng cách đề xuất các phương pháp tối ưu để sử dụng NoSQL Booster, nghiên cứu này có tiềm năng ảnh hưởng đến cách thức phát triển và cải tiến các công cụ quản trị cơ sở dữ liệu trong tương lai, góp phần vào sự phát triển chung của ngành công nghệ thông tin.

CHƯƠNG 1 TỔNG QUAN VỀ NOSQLBOOSTER VÀ CƠ SỞ DỮ LIỆU MONGODB

1.1. Giới thiệu về NoSQLBooster

1.1.1. Định nghĩa

NoSQLBooster là một công cụ quản lý cơ sở dữ liệu NoSQL, đặc biệt là MongoDB. Nó cung cấp một giao diện đồ họa trực quan, giúp người dùng tương tác với cơ sở dữ liệu một cách dễ dàng và hiệu quả hơn so với việc sử dụng các công cụ dòng lệnh.

1.1.2. Các tính năng chính

- **Truy vấn:** Hỗ trợ các truy vấn phức tạp, bao gồm cả việc sử dụng các aggregation pipeline.
- **Quản lý dữ liệu:** Cho phép người dùng tạo, sửa, xóa các collection, document, index một cách trực quan.
- **Import/export:** Dễ dàng import/export dữ liệu từ/vào các định dạng khác nhau như JSON, CSV.
- **Backup/restore:** Tạo các bản sao lưu và phục hồi cơ sở dữ liệu một cách nhanh chóng.
- **Visualize:** Hiển thị dữ liệu dưới dạng biểu đồ, giúp người dùng dễ dàng phân tích và hiểu dữ liệu.

1.1.3. So sánh với các công cụ khác

- **Ưu điểm:** Giao diện thân thiện, dễ sử dụng, hỗ trợ nhiều tính năng nâng cao.
- **Nhược điểm:** Có thể không linh hoạt bằng các công cụ dòng lệnh, cần phải cài đặt thêm phần mềm.

Bảng so sánh NoSQLBooster với các công cụ khác

Tính năng	NoSQLBooster	MongoDB Compass	Robomongo
Giao diện	Đồ họa, trực quan	Đồ họa, trực quan	Đồ họa, trực quan
Hỗ trợ	MongoDB	MongoDB	MongoDB
Tính năng	Truy vấn, quản lý dữ liệu, import/export, backup/restore, visualize	Truy vấn, quản lý dữ liệu, visualize	Truy vấn, quản lý dữ liệu
Giá cả	Thương mại	Miễn phí	Miễn phí

1.2. Tổng quan về MongoDB

1.2.1. Khái niệm cơ bản về MongoDB

1.2.1.1. Khái niệm

- MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu được hàng triệu người sử dụng. MongoDB được viết bằng ngôn ngữ C++.
- MongoDB có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế đến mục đích hướng đối tượng. Nó hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

1.2.1.2. Các thuật ngữ trong MongoDB

- Các thuật ngữ thường được sử dụng trong MongoDB:
 - + `_id`: là trường bắt buộc có trong mỗi document. Trường `_id` đại diện cho một giá trị duy nhất trong document. Trường `_id` cũng có thể hiểu là khóa chính trong document. Khi thêm mới một document thì MongoDB sẽ được tự động sinh ra một `_id` đại diện cho document đó và là duy nhất trong cơ sở dữ liệu MongoDB.
 - + Collection: là nhóm của nhiều document trong MongoDB. Collection có thể được hiểu là một bản tương ứng trong cơ sở dữ liệu RDBMS. Các collection không phải định nghĩa các cột, các hàng hay các kiểu dữ liệu trước.
 - + Cursor: là một con trỏ đến tập kết quả của một truy vấn. Máy khách có thể lặp qua một con trỏ để lấy kết quả.
 - + Database: là nơi chứa các Collection, giống với CSDL RDBMS chúng chứa các bảng. Mỗi database có một tập tin riêng lưu trữ trên bộ nhớ vật lý. Một máy chủ MongoDB có thể chứa nhiều database.
 - + Document: một bản ghi thuộc Collection thì được gọi là một Document. Các Document lần lượt bao gồm các trường và giá trị.
 - + Field: là một cặp name – value trong một document và có thể có hoặc không hoặc nhiều trường. Các trường giống các cột ở CSDL quan hệ.
 - + JSON (JavaScript Object Notation): có thể đọc được ở định dạng văn bản đơn giản thể hiện cho các dữ liệu có cấu trúc. Hiện tại JSON đang được hỗ trợ rất nhiều ngôn ngữ lập trình.
 - + Index: là cấu trúc dữ liệu đặc biệt, dùng để chứa phần nhỏ của các tập dữ liệu một cách dễ dàng để quét. Index hỗ trợ phân tích một cách hiệu quả các truy vấn.

1.2.2. Ưu điểm và nhược điểm của MongoDB

- Ưu điểm:
 - + Linh hoạt và dễ sử dụng: MongoDB cho phép lưu trữ các tài liệu không cấu trúc và linh hoạt, cho phép mô hình dữ liệu thay đổi dễ dàng mà không cần thay đổi cấu trúc CSDL.
 - + Khả năng mở rộng: hỗ trợ khả năng mở rộng ngang, cho phép người dùng tăng cường khả năng xử lý lưu lượng dữ liệu bằng cách thêm các nút mới vào hệ thống.
 - + Hiệu suất cao: MongoDB được thiết kế để cung cấp hiệu suất cao với khả năng xử lý truy vấn nhanh và thời gian đáp ứng ngắn.
 - + Tính sẵn sàng và độ tin cậy: MongoDB cung cấp tính năng sao chép dữ liệu và cân bằng tải giúp đảm bảo tính sẵn sàng và độ tin cậy của hệ thống.
- Nhược điểm:
 - + Khả năng mở rộng đọc hạn chế: một bảng trong MongoDB có thể chỉ chứa được một số lượng tài liệu hạn chế, điều này có thể gây hạn chế về khả năng mở rộng đọc.
 - + Rủi ro mất dữ liệu: do MongoDB không đảm bảo tính toàn vẹn dữ liệu mặc định nên có quy cơ bị mất dữ liệu trong trường hợp xảy ra sự cố chẳng hạn như mất điện hoặc lỗi phần cứng.
 - + Khó khăn truy vấn phức tạp: so với SQL, việc truy vấn phức tạp trong MongoDB có thể phức tạp hơn và đòi hỏi biết sâu về cấu trúc dữ liệu và cú pháp truy vấn.

1.2.3. Các phương pháp trong quản trị MongoDB:

- Thiết kế Schema hợp lý:
 - + Hiểu được sự khác biệt giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu tài liệu: thiết kế schema trong MongoDB khác với SQL nên phải tận dụng mô hình tài liệu để lưu trữ dữ liệu liên quan cùng nhau thay vì sử dụng các bảng và khóa ngoại.

- + Sử dụng các tài liệu nhúng: đối với dữ liệu thường xuyên được truy cập cùng nhau, hãy nhúng tài liệu để giảm thiểu số lượng truy vấn và cải thiện hiệu suất.
- Tối ưu hóa chỉ mục (Indexes):
 - + Tạo chỉ mục cho các trường thường xuyên truy vấn: sử dụng chỉ mục để tăng tốc độ truy vấn và đảm bảo rằng các chỉ mục được tối ưu hóa cho các truy vấn phổ biến.
 - + Tránh chỉ mục dư thừa: quá nhiều chỉ mục có thể làm chậm quá trình ghi dữ liệu. Hãy sử dụng Performance Advosor để xác định và loại bỏ các chỉ mục không cần thiết.
- Quản lý tài nguyên hệ thống:
 - + Kích thước máy chủ phù hợp: đảm bảo rằng máy chủ có đủ tài nguyên để xử lý công việc hiệu quả.
 - + Sử dụng replication và sharding: để đảm bảo tính sẵn sàng và khả năng mở rộng.
- Sao lưu và khôi phục dữ liệu:
 - + Thực hiện sao lưu thường xuyên: đảm bảo các kết hoạch sao lưu dữ liệu định kỳ để tránh mất mát dữ liệu trong trường hợp xảy ra sự cố.
 - + Kiểm tra khôi phục dữ liệu: thường xuyên kiểm tra khả năng khôi phục dữ liệu từ các bản sao lưu để đảm bảo quá trình này hoạt động hiệu quả.
- Giám sát và bảo mật:
 - + Giám sát hiệu suất: sử dụng công cụ giám sát để theo dõi hiệu suất của hệ thống và phát hiện sớm các vấn đề tiềm ẩn.
 - + Bảo mật dữ liệu: thiết lập các biện pháp như xác thực, phân quyền truy cập và mã hóa dữ liệu để bảo vệ thông tin nhạy cảm.

1.2.4. So sánh MongoDB với các CSDL quan hệ

	MongoDB	RDBMS
Cấu trúc dữ liệu	Tài liệu JSON/BSON	Bảng với hàng và cột
Ngôn ngữ truy vấn	Cú pháp JSON	SQL
Khả năng mở rộng	Mở rộng ngang	Mở rộng dọc
Hiệu suất và tính nhất quán	Cao, phù hợp cho dữ liệu lớn	Tốt cho giao dịch phức tạp, nhất quán mạnh
Tính linh hoạt	Linh hoạt, không cần thay đổi schema	Cần thay đổi schema khi thay đổi cấu trúc
Ứng dụng	Web, ứng dụng di động, dữ liệu phi cấu trúc	Ứng dụng doanh nghiệp, tài chính

CHƯƠNG 2 THIẾT KẾ HỆ THỐNG

2.1 Phân tích yêu cầu hệ thống

a. Yêu cầu chức năng

- **Chức năng quản lý khách hàng (Customers Management)**

Yêu cầu: Quản lý thông tin khách hàng, theo dõi việc đặt vé của từng khách hàng.

Dữ liệu cần quản lý:

- Mã khách hàng (CustomerID)
- Tên (Name)
- Số điện thoại(Phone)
- Địa chỉ(Address)
- Email

Quy trình:

- Nhân viên nhập thông tin khách hàng khi khách hàng đăng ký hoặc đặt vé lần đầu tiên.
- Khách hàng có thể cập nhật thông tin cá nhân khi cần.

- **Chức năng quản lý xe khách (Bus Management)**

Yêu cầu: Quản lý các thông tin liên quan đến xe khách (loại xe, số ghế, biển số, nhà xe).

Dữ liệu cần quản lý:

- Mã xe (BusID)
- Biển số xe (LicensePlate)
- Loại xe (Type)
- Số ghế (Seats)
- Nhà xe (BusCompany)

Quy trình:

- Nhân viên quản lý cập nhật thông tin các xe hiện có, thêm xe mới, và loại bỏ xe không còn sử dụng được.

- **Chức năng quản lý chuyến xe (Trip Management)**

Yêu cầu: Quản lý các chuyến xe bao gồm thời gian khởi hành, điểm đi, điểm đến, và xe khách sử dụng.

Dữ liệu cần quản lý:

- Mã chuyến xe (TripID)
- Thời gian khởi hành (DepartureTime)
- Điểm đi, điểm đến (Departure, Destination)
- Mã xe (BusID - liên kết với thông tin xe khách)

Quy trình:

- Nhân viên lập kế hoạch và đăng ký chuyến xe (ngày, giờ khởi hành, tuyến đường, và xe khách sử dụng).
- Cập nhật tình trạng chuyến xe (hoàn thành, hủy, hoặc thay đổi).

- **Chức năng quản lý vé xe (Ticket Management)**

Yêu cầu: Quản lý thông tin vé bao gồm số ghế, trạng thái thanh toán, liên kết với chuyến xe và khách hàng.

Dữ liệu cần quản lý:

- Mã vé (TicketID)
- Số ghế (SeatNumber)
- Trạng thái thanh toán (PaymentStatus)
- Mã chuyến xe (TripID - liên kết với chuyến xe)
- Mã khách hàng (CustomerID - liên kết với khách hàng)

Quy trình:

- Khách hàng đặt vé cho chuyến xe mong muốn.
- Nhân viên kiểm tra trạng thái thanh toán và cập nhật sau khi khách hàng thanh toán.

- **Chức năng quản lý tài xế (Driver Management)**

Yêu cầu: Quản lý thông tin về tài xế, theo dõi giấy phép và liên kết các chuyến xe mà tài xế được phân công.

Dữ liệu cần quản lý:

- Mã tài xế (DriverID)
- Tên, số điện thoại
- Giấy phép lái xe (License)

Quy trình:

- Nhân viên quản lý nhập thông tin tài xế và theo dõi việc phân công tài xế cho từng chuyến xe.

- **Chức năng quản lý phân công tài xế (Assignment Management)**

Yêu cầu: Quản lý việc phân công tài xế cho chuyến xe.

Dữ liệu cần quản lý:

- Mã phân công (AssignmentID)
- Mã chuyến xe (TripID - liên kết với chuyến xe)
- Mã tài xế (DriverID - liên kết với tài xế)

Quy trình:

- Nhân viên quản lý việc phân công tài xế cho các chuyến xe dựa trên lịch trình.

- **Chức năng quản lý tài xế (Driver Management)**

Yêu cầu: Quản lý thông tin về tài xế, theo dõi giấy phép và liên kết các chuyến xe mà tài xế được phân công.

Dữ liệu cần quản lý:

- Mã tài xế (DriverID)
- Tên, số điện thoại
- Giấy phép lái xe (License)

Quy trình:

- Nhân viên quản lý nhập thông tin tài xế và theo dõi việc phân công tài xế cho từng chuyến xe.

- **Chức năng quản lý người dùng (User Management)**

Yêu cầu: Quản lý thông tin người dùng hệ thống, phân quyền sử dụng (khách hàng, tài xế, nhân viên quản lý).

Dữ liệu cần quản lý:

- Tên người dùng (username)
- Mật khẩu (password)
- Vai trò (role: customer, driver, admin)

Quy trình:

- Người dùng đăng ký và đăng nhập vào hệ thống.
- Phân quyền cho người dùng dựa trên vai trò để truy cập các chức năng tương ứng.

- **Báo cáo và thống kê (Reporting and Analytics)**

Yêu cầu: Hệ thống cần cung cấp các báo cáo về tình hình hoạt động, bao gồm số vé bán ra, doanh thu, danh sách xe không còn sử dụng được, và các chuyến xe phổ biến.

Dữ liệu cần quản lý:

- Thống kê doanh thu theo ngày, tuần, tháng
- Thống kê các chuyến xe được ưa chuộng nhất
- Báo cáo về tình trạng các xe (hỏng hóc, bảo trì, không còn sử dụng)

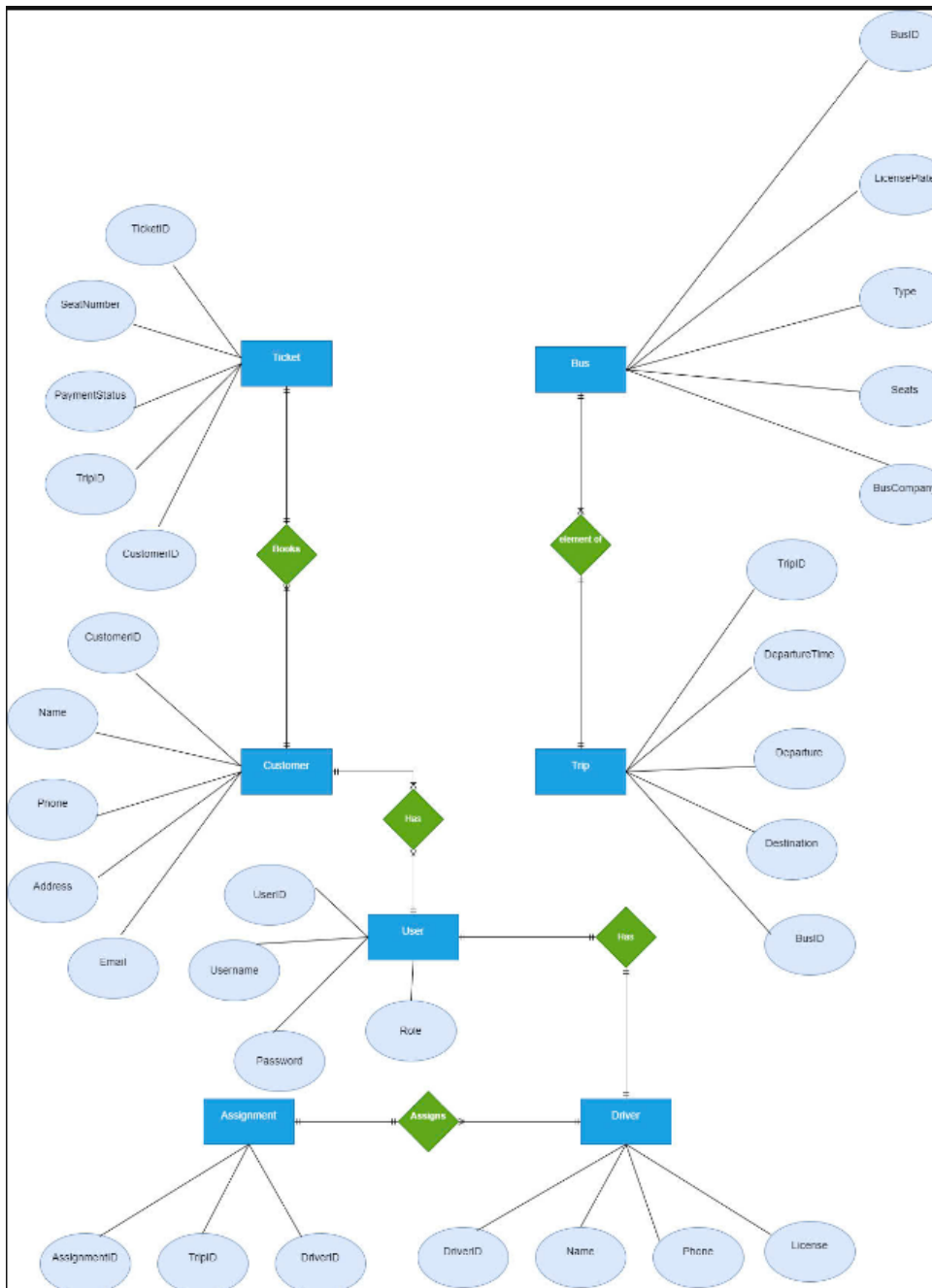
Quy trình:

- Nhân viên hoặc quản lý có thể tạo và xem báo cáo bất cứ lúc nào để theo dõi tình hình hoạt động.

b. Yêu cầu phi chức năng

- **Hiệu suất:** Hệ thống cần đảm bảo hiệu suất cao, đặc biệt là trong việc tìm kiếm vé, chuyến xe, và cập nhật trạng thái thanh toán.
- **Bảo mật:** Đảm bảo dữ liệu khách hàng, tài xế và các giao dịch được bảo mật.
- **Khả năng mở rộng:** Hệ thống cần có khả năng mở rộng khi số lượng khách hàng, vé, và chuyến xe tăng lên.
- **Truy cập đa nền tảng:** Hệ thống có thể cần truy cập qua các ứng dụng web, mobile hoặc desktop.

2.2 Thiết kế sơ đồ ERD



2.3 Xác định cấu trúc các collection

- **Collection customers**

{

```
"_id": ObjectId("..."),
"customerID": "CUST001",
"name": "Nguyễn Văn A",
"phone": "0901234567",
"address": "123 Đường ABC, Quận X, TP.HCM",
"email": "nguyenvana@example.com",
"userID": ObjectId("...")
}
```

- **Collection buses**

```
{
  "_id": ObjectId("..."),
  "busID": "BUS001",
  "licensePlate": "79A-12345",
  "type": "Limousine",
  "seats": 20,
  "busCompany": "Công ty ABC"
}
```

- **Collection trips**

```
{
  "_id": ObjectId("..."),
  "tripID": "TRIP001",
  "departureTime": ISODate("2024-09-20T08:00:00Z"),
  "departure": "Hà Nội",
  "destination": "TP.HCM",
  "busID": ObjectId("...")
}
```

- **Collection drivers**

```
{  
  "_id": ObjectId("..."),  
  "driverID": "DRIVER001",  
  "name": "Trần Văn B",  
  "phone": "0912345678",  
  "license": "123456789"  
}
```

- **Collection assignment**

```
{  
  "_id": ObjectId("..."),  
  "assignmentID": "ASSIGN001",  
  "tripID": ObjectId("..."),  
  "driverID": ObjectId("...")  
}
```

- **Collection tickets**

```
{  
  "_id": ObjectId("..."),  
  "ticketID": "TICKET001",  
  "seatNumber": "12A",  
  "paymentStatus": "paid",  
  "tripID": ObjectId("..."),  
  "customerID": ObjectId("...")  
}
```

- **Collection user**

```
{
  "_id": ObjectId("..."),
  "username": "user123",
  "password": "hashed_password",
  "role": "customer",
  "customerId": ObjectId("...")
}
```

2.4 Thiết kế cơ sở dữ liệu "Quản lý dịch vụ xe khách"

Collection Customers (Khách hàng): Lưu trữ thông tin khách hàng, gồm các thuộc tính như : tên, số điện thoại, email, địa chỉ.

Collection Buses (Xe khách): Lưu trữ thông tin về xe khách gồm các thuộc tính như biển số, loại xe, số ghế, nhà xe.

Collection trips (Chuyến xe): Lưu thông tin về các chuyến xe như thời gian khởi hành, điểm đi, điểm đến, xe được sử dụng (busId).

Collection Tickets (Vé): Lưu thông tin về vé đặt chỗ, gồm các thuộc tính như số ghế, trạng thái thanh toán, liên kết với chuyến xe và khách hàng (tripId và customerId)

Collection Drivers (Tài xế): Lưu thông tin tài xế gồm các thuộc tính như tên, số điện thoại, giấy phép lái xe

Collection Assignments (Phân công): Lưu thông tin về phân công tài xế cho từng chuyến xe

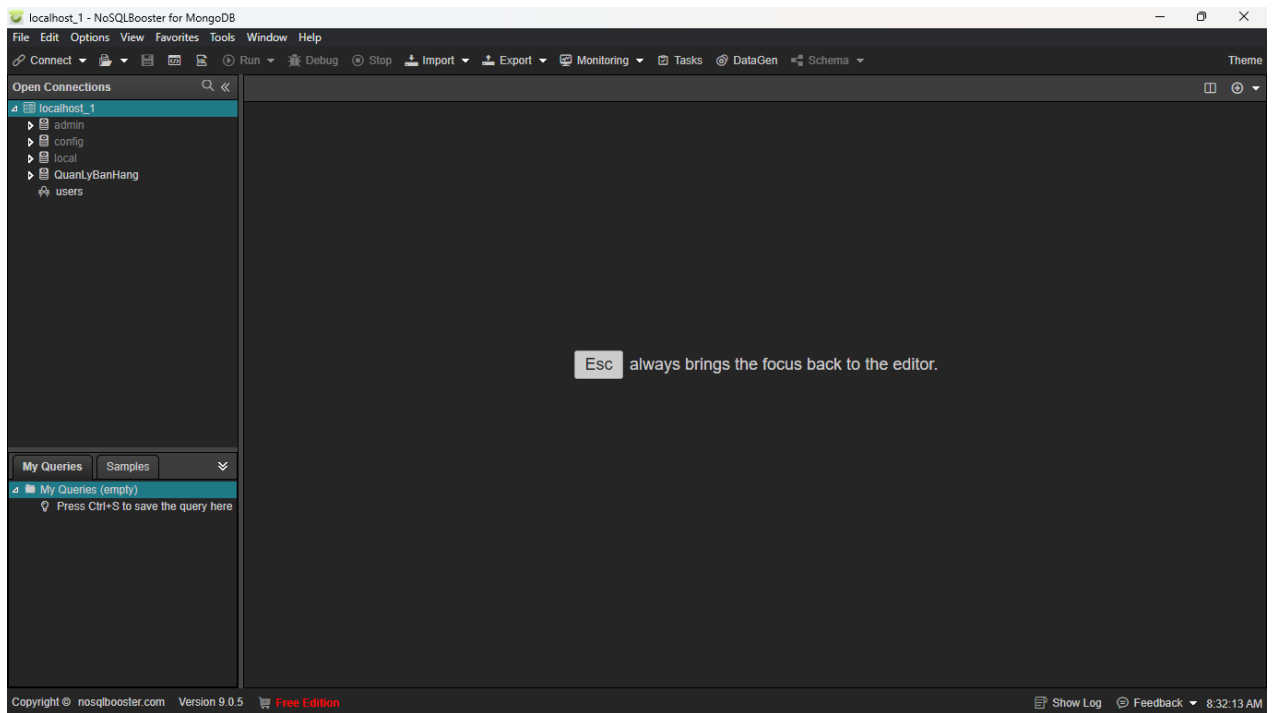
Collection Users (Người dùng): Lưu thông tin về người dùng bao gồm tài khoản, mật khẩu, mã khách hàng, mã tài xế hoặc quản trị viên (tùy vào roles đã chọn)

CHƯƠNG 3 TRIỂN KHAI HỆ THỐNG

3.1 Cài đặt và cấu hình NoSQLBooster

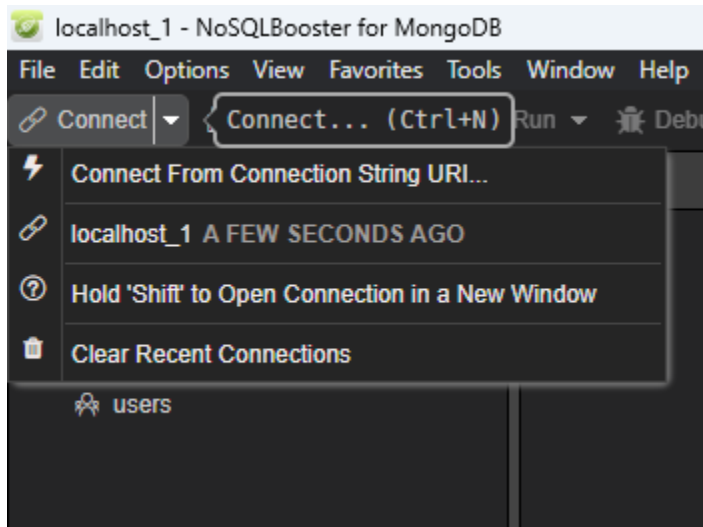
3.1.1 Cài đặt

- Tải file .exe phù hợp với hệ điều hành từ trang chủ NoSQLBooster
<https://www.nosqlbooster.com/downloads>
- Chạy file .exe và sau đó giao diện NoSQLBooster sẽ hiện lên như sau

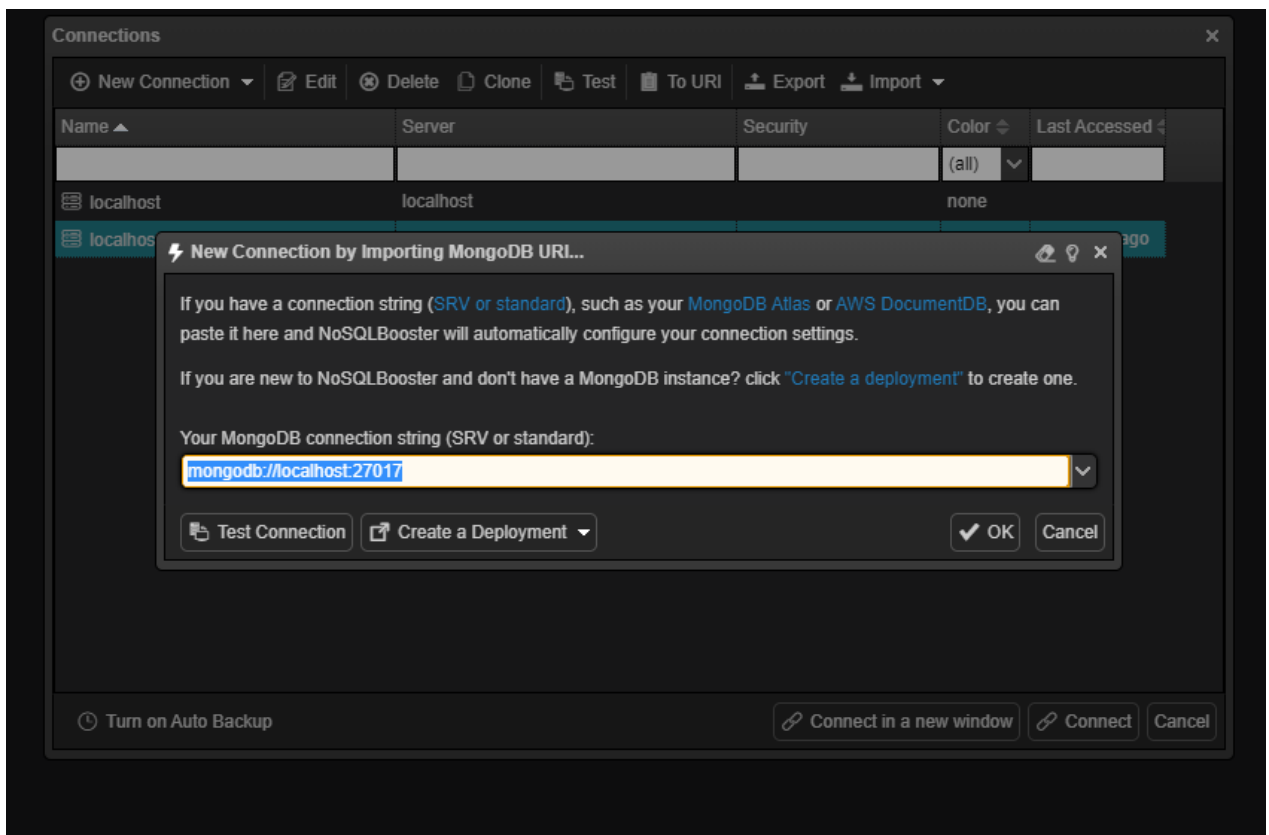


3.1.2 Cấu hình

- Mở NoSQLBooster và chọn Connect.

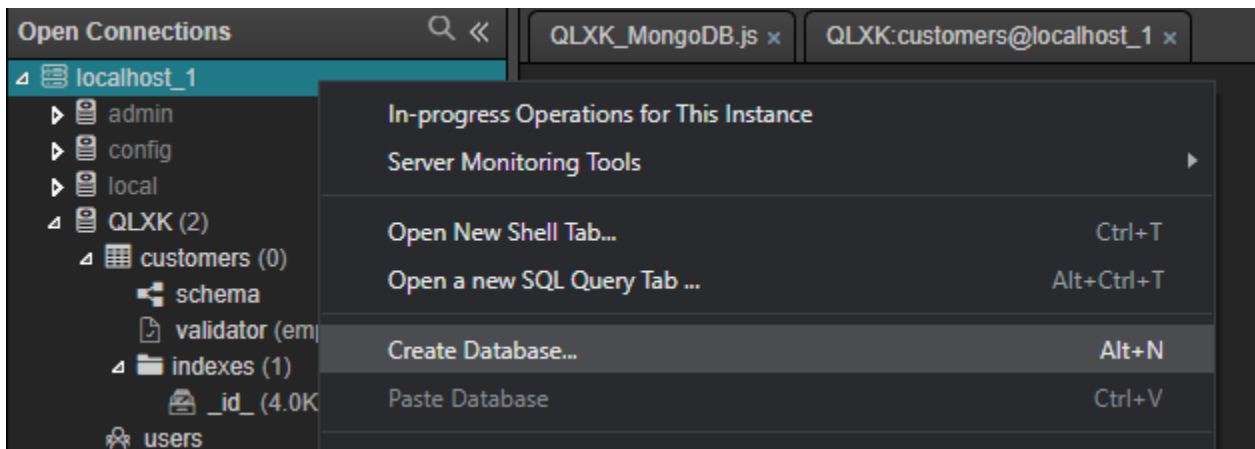


- Điền thông tin kết nối cơ sở dữ liệu MongoDB, bao gồm: Hostname, Port, và thông tin xác thực (username, password) nếu có.

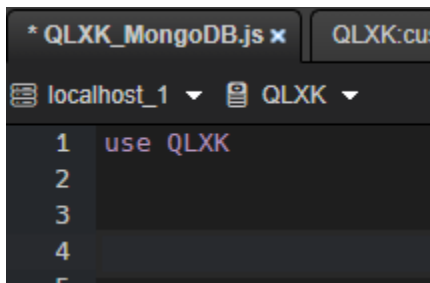


3.2 Tạo cơ sở dữ liệu và các collection

- Ấn chuột phải vào tên localhost -> Create Database.

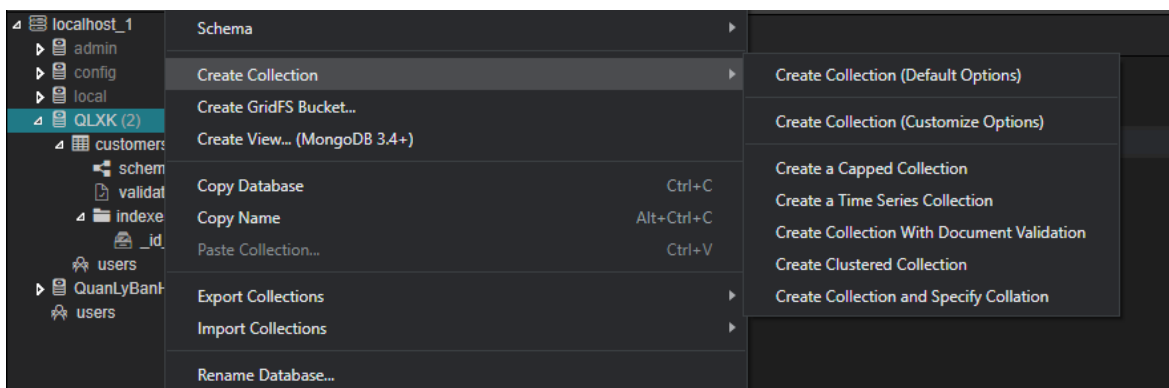


- Hoặc mở Shell Tab và nhập lệnh như sau: Use + tên CSDL cần tạo.



Use QLXK

- Tạo collection bằng thao tác: Chuột phải vào tên CSDL > Create Collection



- Hoặc tiếp tục trên Shell Tab và nhập các lệnh tạo collection như sau:

```
db.createCollection("customers");
```

```
db.createCollection("buses");
```

```
db.createCollection("trips");
```

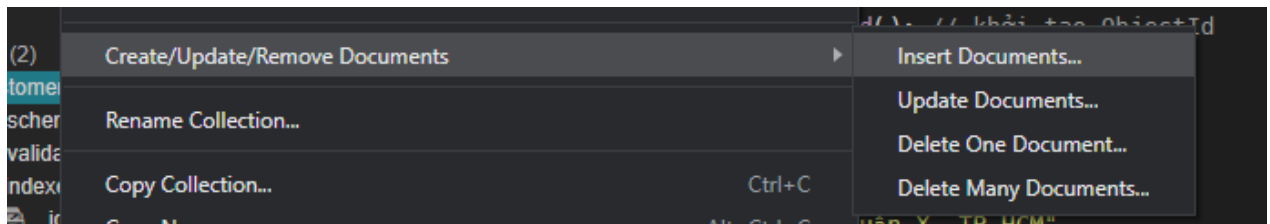
```
db.createCollection("drivers");
```

```
db.createCollection("assignment");
```

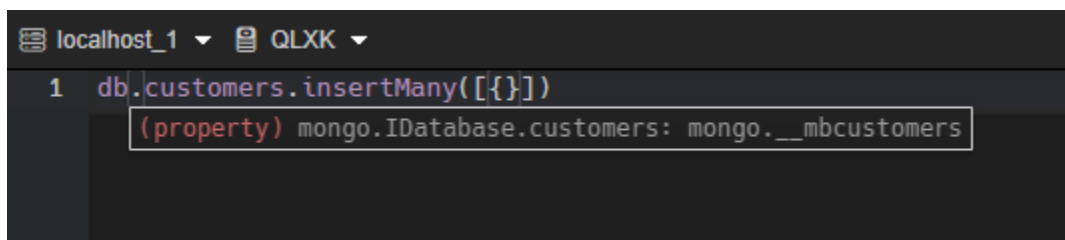
```
db.createCollection("tickets");
```

```
db.createCollection("user");
```

- Thao tác nhập dữ liệu vào collection: Chuột phải vào tên collection muốn nhập dữ liệu > Create/Update... > Insert Documents..



- Tiếp đó TabShell được hiện ra với câu lệnh insert có sẵn.



- Nhập dữ liệu như sau:

```
db.customers.insertMany([
```

```
{  
  
  customerID: "CUST001",  
  
  name: "Nguyễn Văn A",  
  
  phone: "0901234567",  
  
  address: "123 Đường ABC, Quận X, TP.HCM",  
  
  email: "nguyenvana@example.com",  
  
  userID: ObjectId()  
  
},  
  
{  
  
  customerID: "CUST002",  
  
  name: "Trần Thị B",  
  
  phone: "0909876543",  
  
  address: "456 Đường XYZ, Quận Y, Hà Nội",  
  
  email: "tranthib@example.com",  
  
  userID: ObjectId()  
  
}  
  
D)
```

3.3 Thực hiện các thao tác CRUD

// CREATE (Tạo mới)

// Thêm một khách hàng mới

```
db.customers.insertOne({  
  
  customerID: "CUST003",  
  
  name: "Phạm Thị D",  
  
  phone: "0934567890",  
  
  address: "789 Đường LMN, Quận Z, Đà Nẵng",  
  
  email: "phamthid@example.com",  
  
  userID: ObjectId()  
  
})
```

// READ (Đọc)

// Tìm kiếm khách hàng theo tên

```
db.customers.find({ name: "Nguyễn Văn A" })
```

// Tìm tất cả các chuyến đi từ Hà Nội đến TP.HCM

```
db.trips.find({ departure: "Hà Nội", destination: "TP.HCM" })
```

// UPDATE (Cập nhật)

// Cập nhật số điện thoại của khách hàng

```
db.customers.updateOne(  
  { customerID: "CUST001" },  
  { $set: { phone: "0901111222" } }  
)
```

// Tăng số ghế của tất cả xe buýt loại "Limousine" lên 2

```
db.buses.updateMany(  
  { type: "Limousine" },  
  { $inc: { seats: 2 } }  
)
```

// DELETE (Xóa)

// Xóa một vé cụ thể

```
db.tickets.deleteOne({ ticketID: "TICKET001" })
```

// Xóa tất cả các chuyến đi đã hoàn thành (giả sử có trường status)

```
db.trips.deleteMany({ status: "completed" })
```

3.4 Viết truy vấn phức tạp bằng aggregation

//Tìm số lượng vé đã bán theo từng chuyến đi

```
db.tickets.aggregate([  
  
  {  
  
    $group: {  
  
      _id: "$TripID",  
  
      totalTicketsSold: { $sum: 1 }  
  
    }  
  
  },  
  
  {  
  
    $lookup: {  
  
      from: "trips",  
  
      localField: "_id",  
  
      foreignField: "TripID",  
  
      as: "tripDetails"  
  
    }  
  
  },  
  
])
```

```

{
  $unwind: "$tripDetails"
},
{
  $project: {
    _id: 0,
    trip: "$tripDetails.Departure"
    destination: "$tripDetails.Destination",
    totalTicketsSold: 1
  }
}
])

```

// Tìm số lượng bán vé theo từng khách hàng

```

db.tickets.aggregate([
{
  $group: {
    _id: "$CustomerID",
    totalTicketsSold: { $sum: 1 }
  }
}
])

```

```
}

},

{

  $lookup: {

    from: "customers",

    localField: "_id",

    foreignField: "CustomerID",

    as: "customerDetails"

  }

},

{

  $unwind: "$customerDetails"

},

{

  $project: {

    _id: 0,

    customerName: "$customerDetails.Name",

    totalTicketsSold: 1

  }

}
```

```
}
```

```
])
```

```
// Thống kê số lượng tài xế được phân công theo từng chuyến
```

```
db.assignments.aggregate([
```

```
{
```

```
  $group: {
```

```
    _id: "$TripID",
```

```
    totalDriversAssigned: { $sum: 1 } 
```

```
  }
```

```
},
```

```
{
```

```
  $lookup: {
```

```
    from: "trips",
```

```
    localField: "_id",
```

```
    foreignField: "TripID",
```

```
    as: "tripDetails"
```

```
  }
```

```
},
```



```

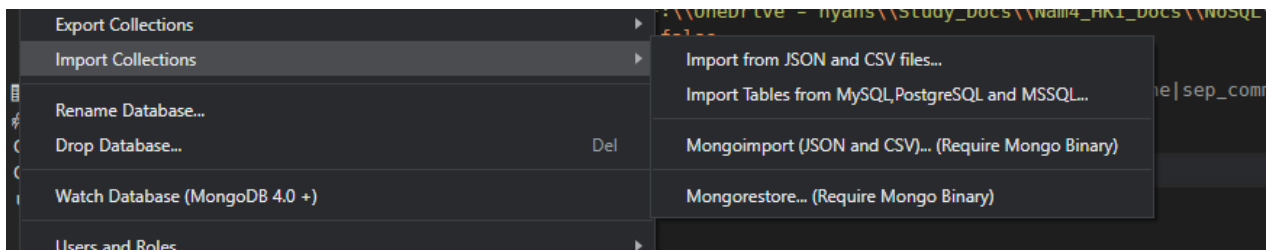
{
  $unwind: "$tripDetails"
},
{
  $project: {
    _id: 0,
    trip: "$tripDetails.Departure",
    destination: "$tripDetails.Destination",
    totalDriversAssigned: 1
  }
}
)

```

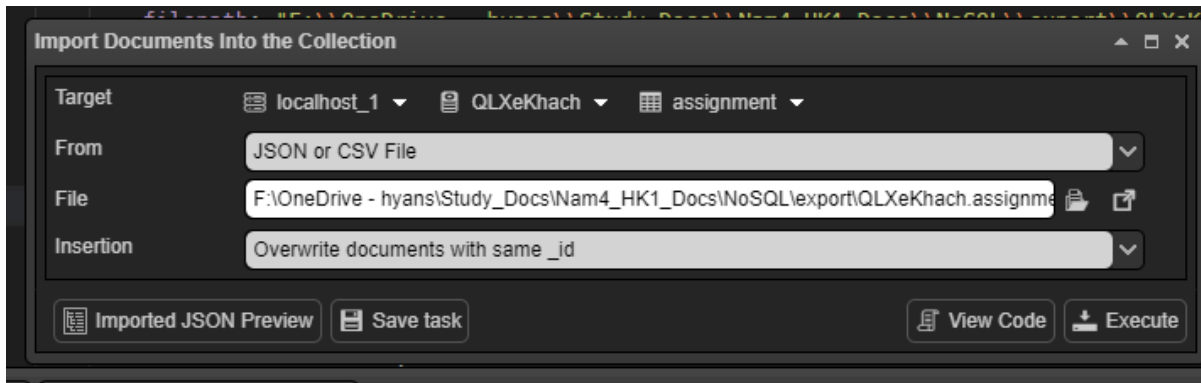
3.5 Thực hiện import/export dữ liệu

3.5.1 Import dữ liệu

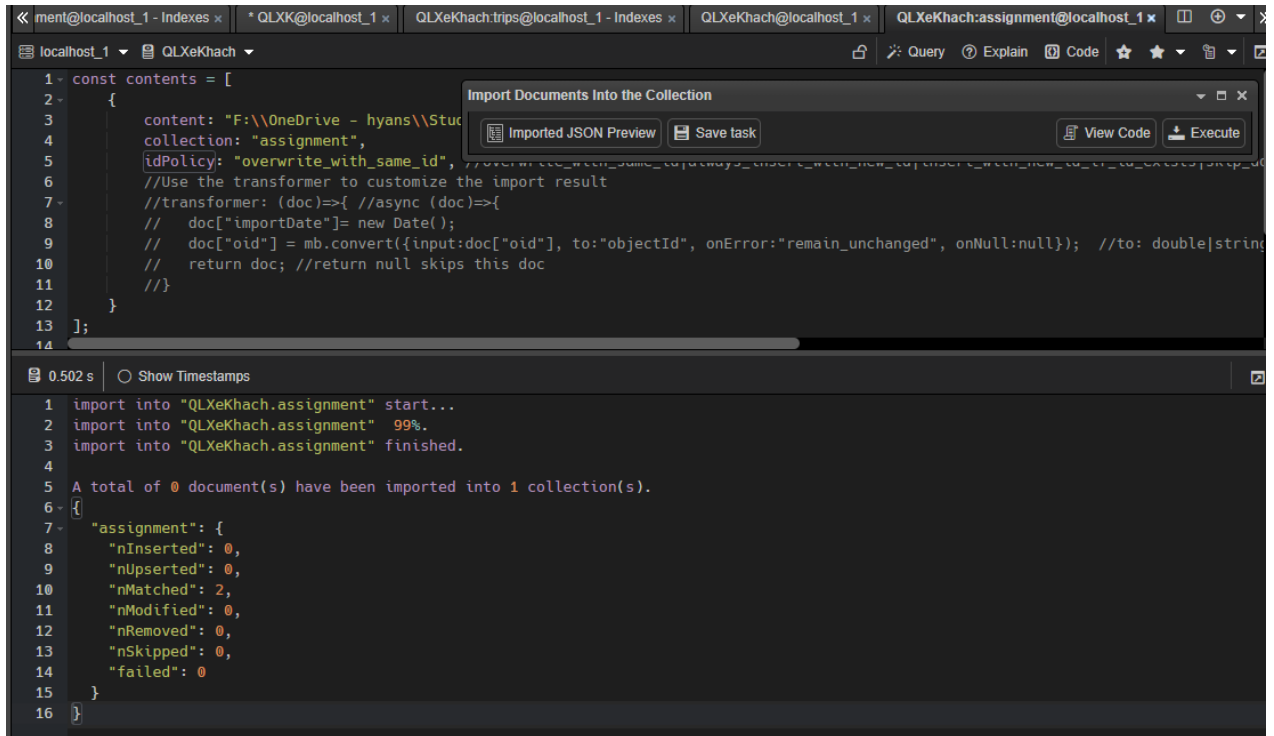
- Ấn chuột phải vào CSDL muốn import collection > Chọn loại file muốn import



- Hộp thoại tùy chỉnh import được hiện lên sau khi chọn file collection. Chọn execute nếu muốn import ngay hoặc có thể tùy chọn một vài cài đặt khác:
 - ☐ **Chọn file:** Duyệt và chọn file dữ liệu cần nhập.
 - ☐ **Chỉ định collection:** Chọn collection đích để lưu trữ dữ liệu.
 - ☐ **Xử lý dữ liệu trùng lặp:** Quyết định có ghi đè lên các tài liệu có cùng ID hay không.
 - ☐ **Xem trước dữ liệu:** Kiểm tra dữ liệu trước khi nhập để đảm bảo chính xác.

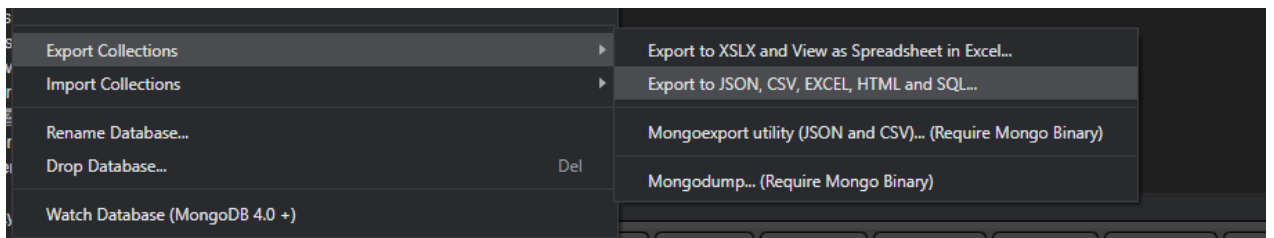


- Sau khi export thành công sẽ hiện như sau:

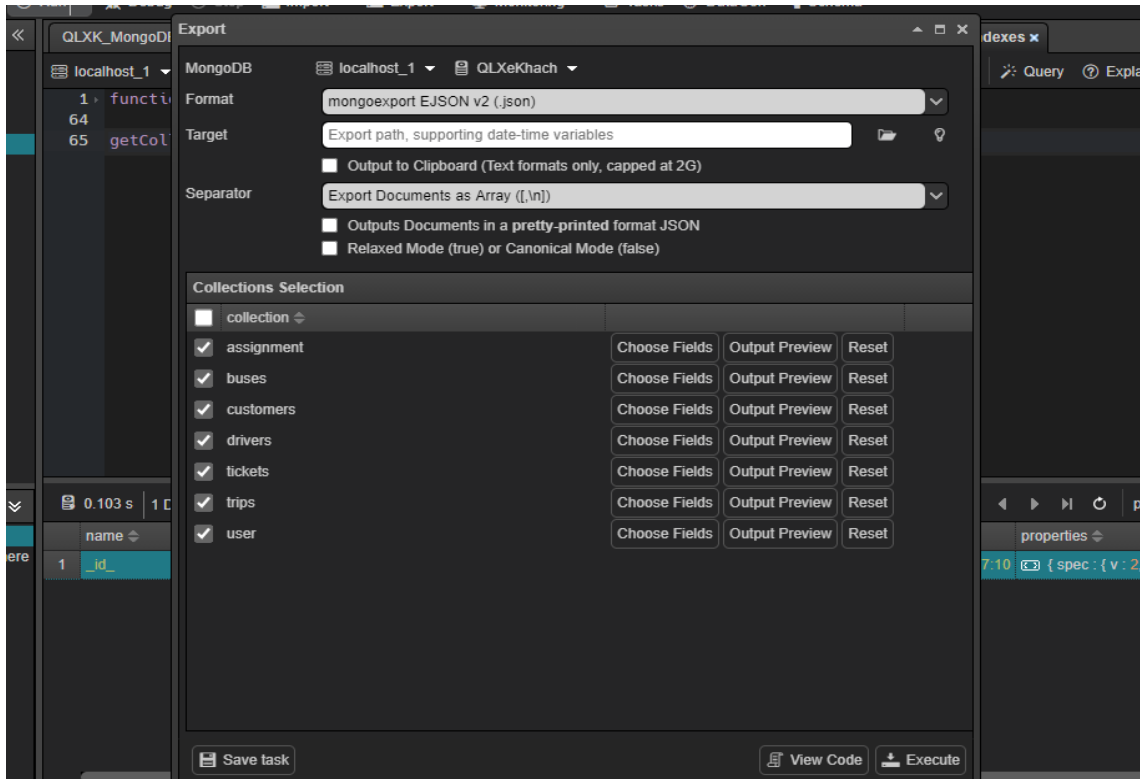


3.5.2 Export dữ liệu

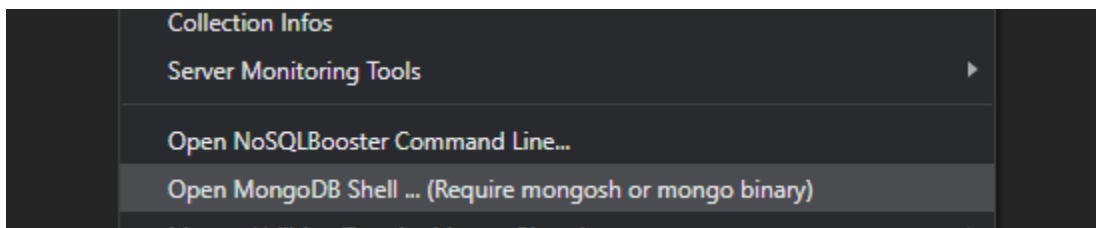
- Chuột phải vào CSDL cần export > Chọn định dạng file muốn export



- Trên cửa sổ export hiện ra các tùy chọn để xuất dữ liệu sang các định dạng khác nhau và tùy chỉnh theo yêu cầu, đặc biệt là định dạng JSON. Người dùng có thể tùy chỉnh quá trình xuất bằng cách chọn các trường, định dạng xuất và xem trước dữ liệu trước khi xuất.



- Hoặc mở terminal như sau và nhập câu lệnh để export dữ liệu:



- Sau khi export hoàn tất, các file collection sẽ xuất hiện như sau

Name	Status	Date modified	Type	Size
QLXeKhach.assignment.json		9/26/2024 8:12 PM	JSON File	1 KB
QLXeKhach.buses.json		9/26/2024 8:12 PM	JSON File	1 KB
QLXeKhach.customers.json		9/26/2024 8:12 PM	JSON File	1 KB
QLXeKhach.drivers.json		9/26/2024 8:12 PM	JSON File	1 KB
QLXeKhach.tickets.json		9/26/2024 8:12 PM	JSON File	1 KB
QLXeKhach.trips.json		9/26/2024 8:12 PM	JSON File	1 KB
QLXeKhach.user.json		9/26/2024 8:12 PM	JSON File	1 KB

CHƯƠNG 4 QUẢN TRỊ VÀ TỐI ƯU HỆ THỐNG

4.1 Thiết lập user và phân quyền

// Tạo quyền customer (Chỉ có quyền đọc dữ liệu)

```
db.createRole({  
  
  role: "customer",  
  
  privileges: [  
  
    { resource: { db: "QLXeKhach", collection: "" }, actions: ["find"] }  
  
  ],  
  
  roles: []  
  
});
```

// Tạo quyền admin (Có toàn quyền trên database)

```
db.createRole({  
  
  role: "admin",  
  
  privileges: [  
  
    { resource: { db: "QLXeKhach", collection: "" }, actions: ["anyAction"] }  
  
  ],  
  
  roles: []  
  
});
```

```
});
```

```
// Tạo user với quyền customer (Chỉ có thể xem thông tin)
```

```
db.createUser({
```

```
  user: "KhachHang",
```

```
  pwd: passwordPrompt(), // Nhập mật khẩu bảo mật
```

```
  roles: [{ role: "customer", db: "QLXeKhach" }]
```

```
});
```

```
// Tạo user với quyền admin (Có toàn quyền)
```

```
db.createUser({
```

```
  user: "admin",
```

```
  pwd: passwordPrompt(),
```

```
  roles: [{ role: "admin", db: "QLXeKhach" }]
```

```
});
```

```
// Kích hoạt xác thực để người dùng phải đăng nhập
```

```
db.adminCommand({ setParameter: 1, authenticationMechanisms: ["SCRAM-SHA-256"]
```

```
});
```

```
// Kiểm tra các user đã tạo
```

```
db.getUsers();
```

```
// Kiểm tra các roles đã tạo
```

```
db.getRoles({ showBuiltinRoles: false });
```

- Tạo roles:
 - Customer: chỉ có quyền đọc dữ liệu và quản lý xe khách, chuyển đi, vé
 - Admin: có toàn quyền trên database.
- Kích hoạt xác thực: điều này đảm bảo rằng người dùng phải đăng nhập để truy cập database sau đó kiểm tra users và roles đã tạo để xác nhận mọi thứ được thiết lập đúng.

4.2 Tạo lịch backup tự động

- Đầu tiên, chúng ta sẽ tạo một script PowerShell để thực hiện backup. Lưu script sau với tên MongoDBBackup.ps1:

```
# Thông tin kết nối
```

```
$DB_NAME = "QLXeKhach"
```

```
$BACKUP_DIR = "C:\MongoBackups"
```

```

$DATE = Get-Date -Format "yyyy-MM-dd_HH-mm-ss"

$MONGODUMP_PATH="C:\Program Files\MongoDB\Server\5.0\bin\mongodump.exe"

# Tạo thư mục backup nếu chưa tồn tại

if (-not (Test-Path $BACKUP_DIR)) {

    New-Item -ItemType Directory -Path $BACKUP_DIR | Out-Null

}

# Thực hiện backup

& $MONGODUMP_PATH --db $DB_NAME --out "$BACKUP_DIR\$DATE"

# Nén file backup

Compress-Archive      -Path      "$BACKUP_DIR\$DATE"      -DestinationPath
"$BACKUP_DIR\$DATE.zip"

# Xóa thư mục gốc sau khi nén

Remove-Item -Recurse -Force "$BACKUP_DIR\$DATE"

# Giữ lại backup trong 30 ngày

Get-ChildItem $BACKUP_DIR -Filter "*.zip" | Where-Object { $_.LastWriteTime -lt (Get-
Date).AddDays(-30) } | Remove-Item

# Ghi log

Add-Content -Path "$BACKUP_DIR\backup_log.txt" -Value "Backup completed on
$DATE"

```

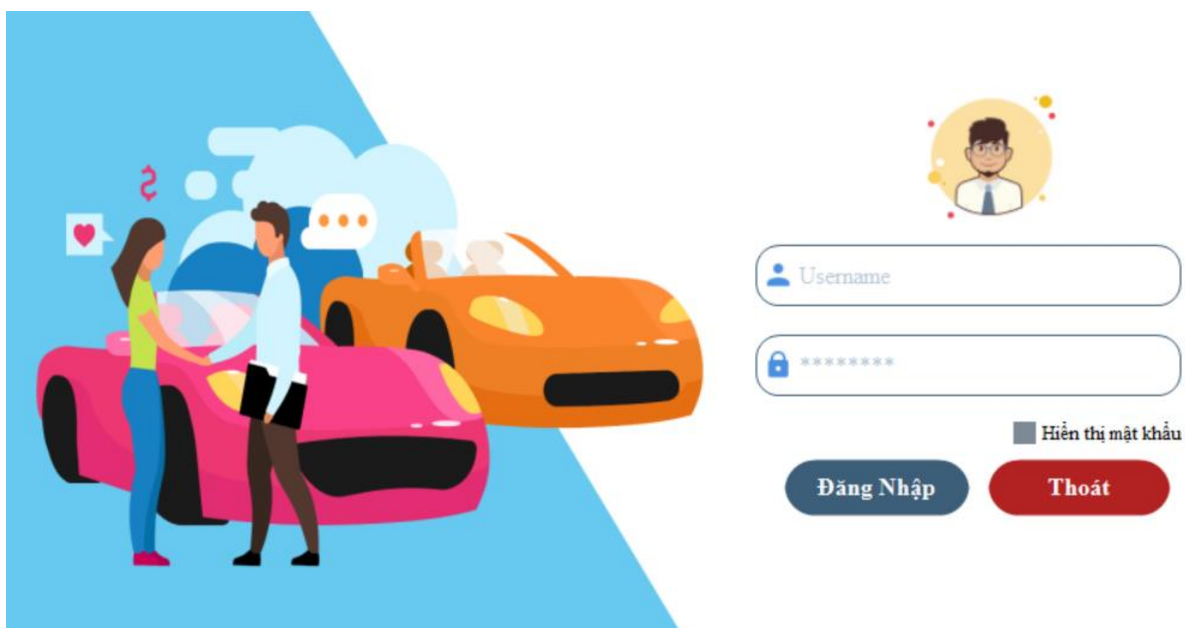
– Điều chỉnh Script:

- Thay đổi \$DB_NAME thành tên database của bạn.
- Điều chỉnh \$BACKUP_DIR đến thư mục bạn muốn lưu các bản backup.
- Cập nhật \$MONGODUMP_PATH để trỏ đến vị trí chính xác của mongodump.exe trên máy của bạn.
- Tạo Task tự động trong Task Scheduler:
 - Mở "Task Scheduler" trên Windows.
 - Chọn "Create Basic Task".
 - Đặt tên cho task, ví dụ: "MongoDB Daily Backup".
 - Chọn tần suất chạy (ví dụ: Daily).
 - Chọn thời gian chạy (ví dụ: 2:00 AM).
 - Trong phần "Action", chọn "Start a program".
 - Trong "Program/script", nhập: powershell.exe
 - Trong "Add arguments", nhập: -ExecutionPolicy Bypass -File "C:\Path\To\MongoDBBackup.ps1"
 - Điều chỉnh đường dẫn tới vị trí bạn lưu script PowerShell.

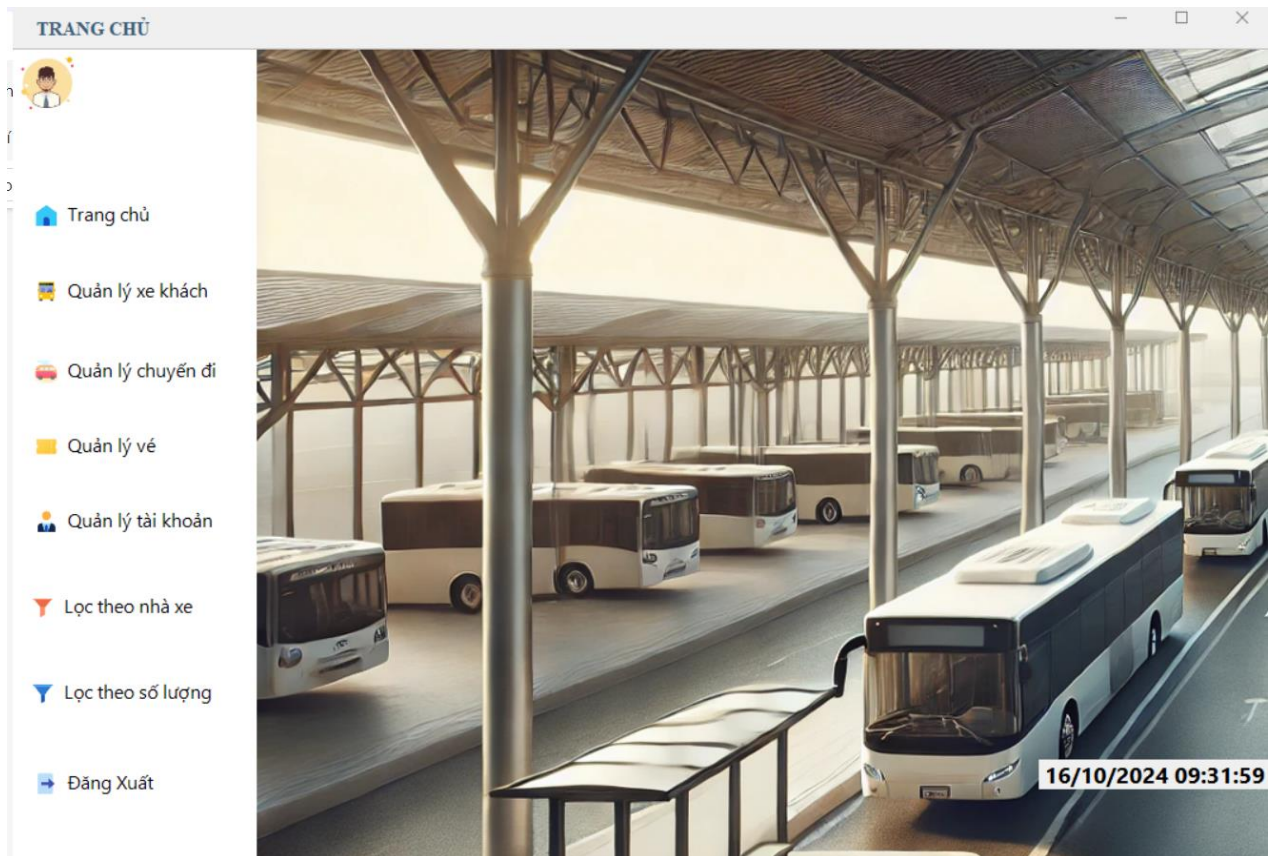
CHƯƠNG 5: DEMO

5.1. Giao diện phần mềm

a. Giao diện đăng nhập



b. Giao diện trang chủ



c. Giao diện chức năng quản lý chuyến đi

QUẢN LÝ CHUYẾN ĐI

Trang chủ

Quản lý xe khách

Quản lý chuyến đi

Quản lý vé

Quản lý tài khoản

Lọc theo nhà xe

Lọc theo số lượng

Đăng Xuất

Mã Chuyến Đi

TRIP001

Thời Gian Khởi Hành

9/20/2024

Điểm Xuất Phát

TP.HCM

Xe Khách Thực Hiện

BUS001

Điểm Đến

Đà Lạt

Thêm

Sửa

Xóa

tripID	departure	destination	departureTime	busID
TRIP001	TP.HCM	Đà Lạt	20/09/2024 13:00	BUS001
TRIP002	TP.HCM	Đà Lạt	20/09/2024 15:00	BUS002
TRIP003	TP.HCM	Đà Lạt	20/09/2024 17:00	BUS003
TRIP004	TP.HCM	Đà Lạt	20/09/2024 21:00	BUS004
TRIP005	TP.HCM	Đà Lạt	20/09/2024 23:00	BUS005
TRIP006	TP.HCM	Đà Lạt	21/09/2024 03:00	Unknown
TRIP007	TP.HCM	Đà Lạt	21/09/2024 13:00	BUS004
TRIP008	TP.HCM	Đà Lạt	30/09/2024 13:00	BUS008
TRIP009	TP.HCM	Đà Lạt	24/09/2024 13:00	BUS008

d. Giao diện chức năng quản lý xe

QUẢN LÝ XE KHÁCH

Trang chủ

Quản lý xe khách

Quản lý chuyến đi

Quản lý vé

Quản lý tài khoản

Lọc theo nhà xe

Lọc theo số lượng

Đăng Xuất

Mã xe khách

BUS001

Số ghế trên xe

30

Biển số xe

79A-12345

Tên công ty quản lý xe

Phương Trang

Loại xe

Ghế ngồi

Thêm

Xóa

Sửa

Reset

busID	licensePlate	type	seats	busCompany
BUS001	79A-12345	Ghế ngồi	30	Phương Trang
BUS002	51B-67890	Giường nằm	22	Phương Trang
BUS003	30A-11223	Limousine	18	Phương Trang
BUS004	29B-44556	Ghế ngồi	30	Thành Bưởi
BUS005	77C-77889	Giường nằm	22	Thành Bưởi
BUS008	73C-37482	Giường nằm	30	Thành Bưởi

e. Giao diện chức năng quản lý vé

QUẢN LÝ VÉ

Trang chủ

Quản lý xe khách

Quản lý chuyến đi

Quản lý vé

Quản lý tài khoản

Đăng Xuất

Mã vé

Ngày bắt đầu

05/07/2021

Số ghế

Thêm

Sửa

Xóa

ticketID	seatNumber	startdate
TICKET002	15B	01/01/2024
TICKET003	18C	01/01/2024
TICKET004	21D	01/01/2024
TICKET005	24E	01/01/2024

f. Giao diện chức năng quản lý tài khoản

43

h

í

o

Trang chủ

Quản lý xe khách

Quản lý chuyến đi

Quản lý vé

Quản lý tài khoản

Lọc theo nhà xe

Lọc theo số lượng

Đăng Xuất

Tên Tài Khoản

user123

Quyền

customer

Mật Khẩu

Hiện thị mật khẩu

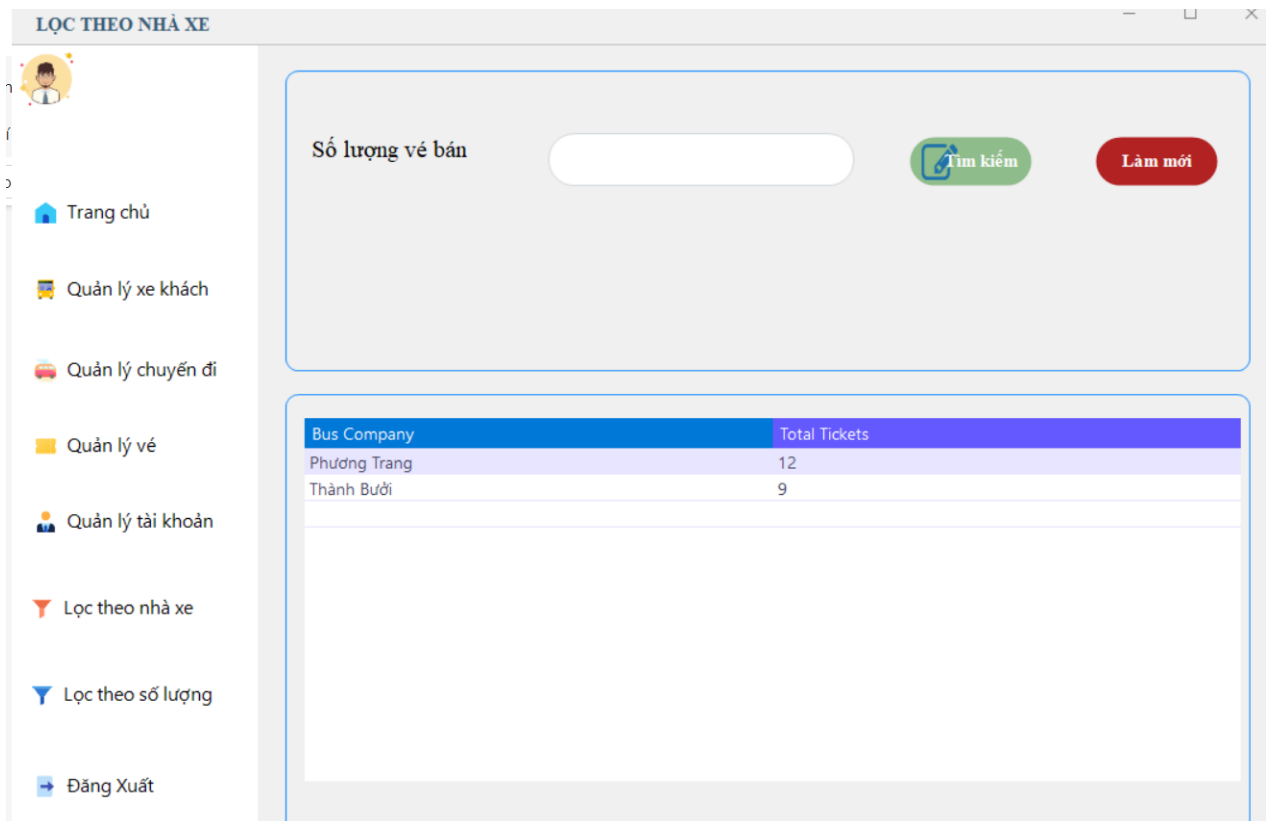
Thêm

Sửa

Xóa

username	role	_id
user123	customer	670e77e746a5037e138d08cb
admin456	ad customer	670e77e746a5037e138d08cc
user789	customer	670e77e746a5037e138d08cd
user101	customer	670e77e746a5037e138d08ce
user112	customer	670e77e746a5037e138d08cf
newuser	customer	670e77e746a5037e138d08d4
khoi	admin	670e91548f0c4c41f13e0a77
khoi	admin	670e9c42ab4d31f4acce72c
khoi123	admin	670ea4c6e90015b4ec2d5eb1

g. Giao diện chức năng lọc theo nhà xe



h. Giao diện chức năng lọc theo số lượng

Lọc theo số lượng khách hàng

Trang chủ

Quản lý xe khách

Quản lý chuyến đi

Quản lý vé

Quản lý tài khoản

Lọc theo nhà xe

Lọc theo số lượng

Đăng Xuất

Tổng số lượng vé

Tìm kiếm

Làm mới

Customer Name	Total Tickets Sold
Trần Thị B	5
Phạm Thị D	5
Hoàng Thị F	6
Nguyễn Văn A	3
Lê Văn E	5

46

KẾT LUẬN VÀ KIẾN NGHỊ

• **Đánh giá kết quả đạt được**

Sau quá trình nghiên cứu và triển khai hệ thống "Quản lý dịch vụ xe khách" sử dụng NoSQL Booster và MongoDB, nhóm chúng em đã đạt được một số kết quả tích cực:

- Hệ thống giúp quản lý dữ liệu hiệu quả, linh hoạt trong việc lưu trữ và truy xuất dữ liệu lớn.
- NoSQL Booster với giao diện trực quan giúp thao tác trên cơ sở dữ liệu dễ dàng, cải thiện hiệu suất làm việc.
- Hệ thống có khả năng mở rộng tốt và đảm bảo bảo mật thông tin nhờ vào việc phân quyền người dùng hợp lý.

• **Hướng phát triển**

Trong tương lai, hệ thống có thể phát triển thêm theo các hướng sau:

- Tối ưu hóa hơn nữa hiệu suất truy vấn và xử lý dữ liệu.
- Cải thiện trải nghiệm người dùng với các tính năng tự động hoàn thiện truy vấn và chỉnh sửa trực tiếp.
- Mở rộng thêm các chức năng mới như tích hợp thanh toán trực tuyến, theo dõi chuyến xe theo thời gian thực.
- Tăng cường bảo mật với các cơ chế mã hóa và quản lý quyền truy cập chặt chẽ hơn.

TÀI LIỆU THAM KHẢO

1. Sách:

- **Chen, M., Han, J., Yu, P. S.** (2014). *Big Data: Related Technologies, Challenges and Future Prospects*. Springer, New York.

2. Tài liệu từ Internet:

- **MongoDB Inc.** (2023). *MongoDB Manual* [online], viewed 3 October 2024, from: <https://www.mongodb.com/docs/manual/>.
- **Nosqlbooster.com** (2023). *NoSQLBooster Documentation* [online], viewed 1 October 2024, from: <https://nosqlbooster.com/blog/>.

PHỤ LỤC