TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN MÔN HỌC

**PHÁT TRIỂN ỨNG DỤNG WEB VỚI MÃ NGUỒN MỞ**

**XÂY DỰNG WEBSITE**

**ĐẶT VÉ XE KHÁCH KIM HOÀNG**

|  |  |
| --- | --- |
| **Người hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Nhóm sinh viên thực hiện:**   1. 110121191 – Nguyễn Phan Bảo Duy 2. 110121020 – Âu Kim Hà |

TRÀ VINH, THÁNG 1 NĂM 2025

TRƯỜNG ĐẠI HỌC TRÀ VINH

KHOA KỸ THUẬT VÀ CÔNG NGHỆ

**BỘ MÔN CÔNG NGHỆ THÔNG TIN**



ĐỒ ÁN MÔN HỌC

**PHÁT TRIỂN ỨNG DỤNG WEB VỚI MÃ NGUỒN MỞ**

**XÂY DỰNG WEBSITE**

**ĐẶT VÉ XE KHÁCH KIM HOÀNG**

|  |  |
| --- | --- |
| **Người hướng dẫn:**  Nguyễn Ngọc Đan Thanh | **Nhóm sinh viên thực hiện:**   1. 110121191 – Nguyễn Phan Bảo Duy 2. 110121020 – Âu Kim Hà |

TRÀ VINH, THÁNG 1 NĂM 2025

# LỜI CẢM ƠN

Lời nói đầu tiên, nhóm chúng em xin trân trọng gửi lời cảm ơn chân thành tới các Thầy và Cô ngành Công nghệ thông tin Trường Đại học Trà Vinh, và đặc biệt là giảng viên hướng dẫn cô Phan Thị Phương Nam đã tận tình hướng dẫn và chỉ bảo em trong suốt quá trình thực hiện đồ án chuyên ngành.

Do chưa có nhiều kinh nghiệm cũng như hạn chế về kiến thức nên không tránh khỏi những thiếu sót trong bài báo cáo. Em rất mong nhận được sự góp ý của quý thầy để bài báo cáo được hoàn thiện tốt hơn.

Em xin chân thành cảm ơn!

# MỤC LỤC

[CHƯƠNG 1. TỔNG QUAN NGHIÊN CỨU 4](#_Toc181622304)

[CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT 4](#_Toc181622305)

[2.1. Nội dung 1 4](#_Toc181622306)

[2.2. Nội dung 2 5](#_Toc181622307)

[2.2.1. Nội dung thứ nhất 5](#_Toc181622308)

[2.2.2. Nội dung thứ hai 5](#_Toc181622309)

[2.3. Nội dung 3 6](#_Toc181622310)

[CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU 6](#_Toc181622311)

[3.1. Mô tả bài toán 6](#_Toc181622312)

[3.2. Phân tích đặc tả hệ thống 6](#_Toc181622313)

[3.2.1. Đặc tả yêu cầu chức năng 6](#_Toc181622314)

[3.2.2. Đặc tả yêu cầu phi chức năng 6](#_Toc181622315)

[3.3. Thiết kế hệ thống 6](#_Toc181622316)

[3.3.1. Kiến trúc hệ thống 6](#_Toc181622317)

[3.3.2. Thiết kế dữ liệu 6](#_Toc181622318)

[3.3.2.1 Mô hình dữ liệu 6](#_Toc181622319)

[3.3.2.2 Danh sách các thực thể và mối kết hợp 6](#_Toc181622320)

[3.3.2.3 Chi tiết thực thể và mối kết hợp 6](#_Toc181622321)

[3.3.3. Thiết kế xử lý 7](#_Toc181622322)

[3.3.3.1 Mô hình DFD mức ngữ cảnh 7](#_Toc181622323)

[3.3.3.2 Mô hình DFD cấp 1 7](#_Toc181622324)

[3.3.3.3 Mô hình DFD cấp 2 7](#_Toc181622325)

[3.3.4. Thiết kế giao diện 7](#_Toc181622326)

[3.3.4.1 Sơ đồ website 7](#_Toc181622327)

[3.3.4.2 Giao diện trang … 7](#_Toc181622328)

[3.3.4.3 Giao diện … 7](#_Toc181622329)

[CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU 7](#_Toc181622330)

[4.1. Bộ dữ liệu thử nghiệm 7](#_Toc181622331)

[4.2. Kết quả thực nghiệm 7](#_Toc181622332)

[4.2.1. Chức năng … 7](#_Toc181622333)

[4.2.2. Chức năng … 7](#_Toc181622334)

[4.2.3. Chức năng … 7](#_Toc181622335)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 7](#_Toc181622336)

[5.1. Kết luận 7](#_Toc181622337)

[5.2. Hướng phát triển 7](#_Toc181622338)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1 Tiêu đề của hình 4](#_Toc181621202)

[Hình 2.2 Gõ tiêu đề cho cái hình 5](#_Toc181621203)

**DANH MỤC BẢNG BIỂU**

[Bảng 3.1 Danh sách các thực thể và mối kết hợp 6](#_Toc181621211)

[Bảng 3.2 Chi tiết thực thể sinhvien 6](#_Toc181621212)

# TỔNG QUAN NGHIÊN CỨU

## Bối cảnh

Trong thời đại hiện nay với công nghệ phát triển mạnh mẽ việc đặt vé xe khách trực tuyến đã trở thành một nhu cầu thiết yếu của người dân. Việc đặt vé trực tuyến giúp người dùng tiết kiệm thời gian, công sức, và có nhiều lựa chọn hơn về hãng xe, tuyến đường và giờ khởi hành. Tuy nhiên, tại tỉnh Trà Vinh, việc đặt vé xe khách trực tuyến vẫn còn hạn chế, phần lớn các doanh nghiệp vẫn sử dụng phương thức đặt vé truyền thống.

Để đáp ứng nhu của khách hàng và nâng cao hiệu quả hoạt động của các doanh nghiệp, việc xây dựng một hệ thống đặt vé xe khách trực tuyến là vô cùng cần thiết. Hệ thống này sẽ giúp đơn giản hóa quá trình đặt vé, quản lý thông tin khách hàng, và tạo ra trải nghiệm tốt hơn cho người dùng.

## Mục tiêu

* Xây dựng hệ thống cơ sở dữ liệu NoSQL hỗ trợ công tác quản lý dịch vụ xe khách trực tuyến..
* Phát triển các chức năng tìm kiếm chuyến xe, tuyến xe, lịch trình, vé xe.
* Tạo tiền đề cho việc phát triển hệ thống đặt vé xe khách trực tuyến hoàn chỉnh trong tương lai.

## Phạm vi nghiên cứu

* Các tài liệu và hệ thống liên quan đến dịch vụ đặt vé xe khách trực tuyến.
* Áp dụng lý thuyết về thiết kế dữ liệu vào việc thu thập, phân tích và xây dựng cơ sở dữ liệu NoSQL.
* Xây dựng cơ sở dữ liệu và tạo dữ liệu mẫu thử để kiểm tra các chức năng cơ bản.

## Phương pháp nghiên cứu

Nghiên cứu cơ sở lý thuyết: Nghiên cứu về sự hài lòng của khách hàng trong dịch vụ đặt vé xe khách.

Nghiên cứu thực nghiệm: Nghiên cứu thực hiện nhằm thu thập dữ liệu từ khách hàng đã sử dụng dịch vụ đặt vé xe, mức độ hài lòng, các yếu tố ảnh hưởng như giá cả, chất lượng dịch vụ, giao diện website và hỗ trợ khách hàng. Nghiên cứu này không chỉ giúp cung cấp cái nhìn rõ ràng về trải nghiệm của khách hàng mà còn đưa ra các giải pháp cụ thể để cải thiện dịch vụ, từ đó nâng cao mức độ hài lòng của khách hàng với Kim Hoàng.

## Đối tượng nghiên cứu

* Hoạt động đặt vé xe khách tại các doanh nghiệp vận tải hành khách ở tỉnh Trà Vinh.
* Hệ quản trị cơ sở dữ liệu: MongoDB.

# NGHIÊN CỨU LÝ THUYẾT

## Giới thiệu hệ quản trị cơ sở dữ liệu NoSQL

Hệ quản trị cơ sở dữ liệu NoSQL (NoSQL Database Management System) là một hệ thống phần mềm được thiết kế để lưu trữ, quản lý và truy xuất dữ liệu phi quan hệ. Các hệ quản trị cơ sở dữ liệu NoSQL chủ yếu được xây dựng để phục vụ nhu cầu của các ứng dụng yêu cầu tốc độ xử lý cao, khả năng mở rộng linh hoạt và khả năng xử lý dữ liệu lớn.

NoSQL mang lại khả năng lưu trữ dữ liệu linh hoạt, hỗ trợ các mô hình dữ liệu phi quan hệ, có thể xử lý dữ liệu đa dạng và với khối lượng lớn mà không cần các quy tắc chặt chẽ như trong các cơ sở dữ liệu quan hệ (SQL).

* + 1. **Đặc điểm**

**Lưu trữ dữ liệu dạng cặp key-value:**

Một số hệ cơ sở dữ liệu NoSQL lưu trữ dữ liệu dưới dạng cặp "key – value", trong đó mỗi giá trị được gán một key duy nhất. Dữ liệu có thể được truy xuất thông qua key mà không cần phải thực hiện các phép toán phức tạp.

**Chấp nhận dữ liệu bị trùng lặp:**

Các hệ cơ sở dữ liệu NoSQL có thể cho phép dữ liệu trùng lặp do một số máy chủ có thể lưu trữ cùng một thông tin giống nhau, giúp gia tăng hiệu suất và khả năng sẵn sàng của hệ thống.

**Phi quan hệ:**

Hệ thống NoSQL không yêu cầu ràng buộc nhất quán dữ liệu chặt chẽ như trong các cơ sở dữ liệu quan hệ. Điều này giúp tăng tốc độ xử lý và hỗ trợ các mô hình dữ liệu linh hoạt.

**Hiệu suất và tính sẵn sàng cao:**

Với việc lưu trữ phân tán và khả năng mở rộng dễ dàng, hệ thống NoSQL có khả năng chịu tải cao và sẵn sàng trong môi trường có yêu cầu xử lý dữ liệu khối lượng lớn, thời gian phản hồi nhanh

**Khả năng mở rộng dễ dàng:**

Hệ quản trị cơ sở dữ liệu NoSQL hỗ trợ khả năng mở rộng, nghĩa là có thể mở rộng hệ thống bằng cách thêm nhiều server để đáp ứng nhu cầu lưu trữ và xử lý dữ liệu.

* + 1. **Các dạng NoSQL cơ bản**

**Key-Value Data Stores:**

Đặc điểm: Dữ liệu được lưu trữ dưới dạng các cặp "key - value". Mỗi key là duy nhất và được gán với một giá trị (value) có thể là bất kỳ loại dữ liệu nào (chuỗi, số, JSON, v.v.).

Ưu điểm: Rất nhanh và hiệu quả trong việc truy xuất dữ liệu theo key. Thích hợp cho các ứng dụng có tính năng lưu trữ đơn giản.

Ví dụ: Redis, DynamoDB.

**Document-Based Stores:**

Đặc điểm: Dữ liệu được tổ chức và lưu trữ dưới dạng các document. Một document có thể chứa các cấu trúc dữ liệu như mảng hoặc đối tượng JSON, và mỗi document có thể có các trường khác nhau.

Ưu điểm: Linh hoạt, vì mỗi document có thể có cấu trúc khác nhau. Tốt cho các ứng dụng với dữ liệu bán cấu trúc hoặc dữ liệu không đồng nhất.

Ví dụ: MongoDB, CouchDB.

**Column-Family Stores:**

Đặc điểm: Dữ liệu được tổ chức dưới dạng các cột (column) thay vì các hàng (row) như trong cơ sở dữ liệu quan hệ. Mỗi cột có thể chứa nhiều giá trị cho từng khóa (key) cụ thể.

Ưu điểm: Thích hợp cho các ứng dụng cần xử lý dữ liệu theo cột, giúp truy xuất dữ liệu nhanh chóng và tối ưu cho việc lưu trữ các loại dữ liệu lớn và phân tán.

Ví dụ: Apache Cassandra, HBase.

* + 1. **Ưu điểm của NoSQL**

Xử lý dữ liệu lớn: NoSQL là sự lựa chọn hoàn hảo cho các ứng dụng cần xử lý khối lượng dữ liệu lớn và không có cấu trúc cố định, ví dụ như các dịch vụ web xã hội, phân tích dữ liệu lớn.

Tính linh hoạt: Các hệ cơ sở dữ liệu NoSQL linh hoạt trong việc lưu trữ và tổ chức dữ liệu, cho phép lưu trữ các loại dữ liệu khác nhau mà không cần tuân thủ các quy tắc nghiêm ngặt như trong cơ sở dữ liệu quan hệ.

Khả năng mở rộng dễ dàng: Hệ thống NoSQL hỗ trợ khả năng mở rộng ngang một cách dễ dàng, giúp doanh nghiệp có thể mở rộng cơ sở hạ tầng một cách linh hoạt khi cần thiết.

## Cơ sở dữ liệu trong MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, thuộc loại NoSQL, với khả năng lưu trữ và quản lý dữ liệu phi quan hệ (non-relational). MongoDB được thiết kế để đáp ứng các yêu cầu của các ứng dụng hiện đại, đặc biệt là những ứng dụng có khối lượng dữ liệu lớn, cần khả năng mở rộng linh hoạt và xử lý dữ liệu nhanh chóng.

MongoDB sử dụng mô hình lưu trữ dữ liệu dưới dạng document (tài liệu) thay vì bảng (tables) như trong cơ sở dữ liệu quan hệ (RDBMS). Dữ liệu trong MongoDB được lưu trữ dưới dạng JSON-like documents, tức là các document kiểu BSON (Binary JSON). Với cách tiếp cận này, MongoDB có thể xử lý dữ liệu một cách linh hoạt và dễ dàng mở rộng.

**Đặc điểm của MongoDB**

Hệ quản trị cơ sở dữ liệu NoSQL: MongoDB không yêu cầu cấu trúc dữ liệu cố định và không sử dụng các ràng buộc như các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Điều này giúp MongoDB dễ dàng thích nghi với dữ liệu phi cấu trúc và bán cấu trúc.

Document-based storage: Dữ liệu trong MongoDB được lưu trữ dưới dạng các document BSON, mang lại khả năng mở rộng và linh hoạt cao. Mỗi document có thể chứa các cặp key-value và các mảng dữ liệu phức tạp.

**Ưu điểm:**

**Linh hoạt trong việc lưu trữ dữ liệu**: MongoDB cho phép bạn lưu trữ dữ liệu theo bất kỳ cấu trúc nào mà bạn muốn. Mỗi collection có thể chứa các document với cấu trúc khác nhau. Điều này mang lại sự linh hoạt rất cao trong việc làm việc với dữ liệu không đồng nhất.

**Tốc độ truy vấn cao**: Dữ liệu trong MongoDB được lưu trữ dưới dạng các document JSON, giúp việc truy vấn dữ liệu nhanh chóng và hiệu quả. Ngoài ra, trường dữ liệu "\_id" luôn được đánh chỉ mục tự động, giúp truy xuất dữ liệu nhanh hơn.

**Dễ dàng mở rộng**: MongoDB hỗ trợ mở rộng theo chiều ngang , nghĩa là bạn có thể thêm các máy chủ vào hệ thống khi cần thiết mà không ảnh hưởng đến hiệu suất. Điều này giúp MongoDB là một lựa chọn lý tưởng cho các ứng dụng có yêu cầu về khả năng mở rộng và xử lý dữ liệu lớn.

**Không cần cấu trúc dữ liệu cố định**: Mỗi document có thể có một cấu trúc khác nhau, giúp việc thay đổi lược đồ dễ dàng mà không phải làm lại toàn bộ hệ thống cơ sở dữ liệu. Điều này làm giảm bớt sự phức tạp trong việc quản lý và thay đổi cấu trúc của dữ liệu.

Khả năng làm việc với dữ liệu không cấu trúc: MongoDB có thể lưu trữ và truy vấn dữ liệu không cấu trúc, bán cấu trúc hoặc cấu trúc có sự thay đổi theo thời gian. Điều này rất hữu ích trong các ứng dụng cần xử lý dữ liệu như tài liệu văn bản, hình ảnh, video,, v.v.

**Nhược điểm:**

**Thiếu tính nhất quán**: Một trong những nhược điểm lớn nhất của MongoDB là thiếu tính nhất quán dữ liệu như trong các hệ quản trị cơ sở dữ liệu quan hệ

**Tốn bộ nhớ:** Do MongoDB lưu trữ dữ liệu dưới dạng key-value, các collection có thể có dữ liệu trùng lặp và việc lặp lại key sẽ khiến bộ nhớ bị tiêu tốn nhiều hơn. Điều này có thể dẫn đến việc sử dụng tài nguyên bộ nhớ cao nếu không quản lý và tối ưu dữ liệu một cách hợp lý.

**Có thể gây dữ thừa dữ liệu**: Khi không sử dụng, MongoDB yêu cầu bạn lưu trữ dữ liệu lặp lại trong các document khác nhau. Điều này có thể dẫn đến việc thừa dữ liệu và khó khăn trong việc duy trì tính nhất quán khi cập nhật dữ liệu.

### Tạo collection

Collections trong MongoDB tương đương với các bảng trong hệ quản trị cơ sở dữ liệu quan hệ. Tạo một collection đơn giản bằng lệnh:

Cú pháp để tạo một collection trong MongoDB như sau:

use myDatabase

db.createCollection(“myCollection”)

Ví dụ: Tạo collection NhaXe trong testdb

use testdb

db.createCollection(“NhaXe”)

### 2.2.2. Lệnh Insert

Thêm dữ liệu cào collection

Cú pháp để thêm dữ liệu trong MongoDB như sau:

db.myCollection.insert({"name": "Kim", "age": 20})

Ví dụ: Thêm dữ liệu vào nha xe

db.NhaXe.insert({

"tennhaxe": "Nhà Xe Kim Hoàng",

"diachi": "Số 25, Trà Vinh",

"sodienthoai": "02923745678"

})

### 2.2.3. Lệnh Find

Truy vấn dữ liệu bằng lệnh find()

Cú pháp để truy vấn dữ liệu trong MongoDB như sau:

db.myCollection.find({"age": 20})

Ví dụ: Tìm dữ liệu của nhà xe

db.NhaXe.find({

"tennhaxe": "Nhà Xe Kim Hoàng",

"sodienthoai": "02923745678"

})

### 2.2.4. Lệnh Update

Cập nhật dữ liệu bằng lệnh Update()

Cú pháp để cập nhật dữ liệu trong MongoDB như sau:

db.myCollection.updateOne({"name": "Kim"}, {$set: {"age": 21}})

Ví dụ: Cập nhật dữ liệu nhà xe

db.NhaXe.updateOne(

{"tennhaxe": "Nhà Xe Kim Hoàng "},

{$set: {"diachi": "Số 30, Nguyễn Trãi, Cần Thơ"}}

)

### 2.2.5. Lệnh Delete

Xóa dữ liệu bằng lệnh deleteOne()

Cú pháp để xóa dữ liệu trong MongoDB như sau:

db.myCollection.deleteOne({"name": "Kim"})

Ví dụ: Xóa dữ liệu nhà xe

db.NhaXe.deleteOne({"tennhaxe": "Nhà Xe Minh Tâm"})

Xóa toàn bộ collection nhà xe

db.NhaXe.deleteMany({})

## Các cơ chế bảo mật trong MongoDB

MongoDB cung cấp nhiều cơ chế bảo mật để bảo vệ dữ liệu và hệ thống khỏi các mối đe dọa an ninh:

**Xác thực người dùng:** MongoDB hỗ trợ nhiều phương thức xác thực như xác thực nội bộ. Người dùng phải cung cấp thông tin đăng nhập để truy cập cơ sở dữ liệu.

**Phân quyền truy cập:** Hệ thống cho phép chỉ định quyền truy cập chi tiết cho từng người dùng và nhóm người dùng thông qua các vai trò có sẵn hoặc tùy chỉnh.

**Mã hóa dữ liệu:** MongoDB hỗ trợ mã hóa dữ liệu ở cả cấp độ ổ đĩa và truyền tải thông tin, đảm bảo dữ liệu không bị truy cập trái phép.

**Bảo mật mạng** Quản trị viên có thể cấu hình danh sách IP cho phép và sử dụng tường lửa để ngăn chặn các kết nối trái phép.

**Giám sát và Audit:** MongoDB cung cấp công cụ giám sát

## 2.4. Giới thiệu về Docker

### 2.4.1. Docker là gì?

Docker là một nền tảng phần mềm mã nguồn mở, được sử dụng để phát triển, triển khai và chạy các ứng dụng trong các môi trường được đóng gói gọi là container.

Docker giúp các lập trình viên chỉ cần chạy các container lên mà không cần phải lo về việc cài các thư viện hoặc môi trường, điều này giúp các lập trình viên phát triển ứng dụng 1 cách nhanh chóng, đảm bảo tính nhất quán giữa môi trường phát triển và môi trường triển khai.

**Container** là một đơn vị chạy độc lập, bao gồm tất cả các thư viện, tệp hệ thống và các thành phần cần thiết để ứng dụng có thể chạy. Điều này đảm bảo tính nhất quán và dễ dàng quản lý trong môi trường phát triển, kiểm thử và sản xuất.

**Docker Image** là một bản sao của môi trường container, chứa tất cả các cài đặt cần thiết để chạy ứng dụng. Docker Images có thể được chia sẻ và tái sử dụng giữa các dự án và người dùng.

**Docker Engine** là công cụ cho phép tạo và chạy các container, bao gồm các API, và các công cụ cần thiết để tương tác với Docker.

### 2.4.2. Docker Compoes là gì?

Docker Compose là một công cụ dùng để định nghĩa và chạy các chương trình Docker sử dụng nhiều container. Với Docker Compose, chúng ta sử dụng một file YAML để thiết lập các service cần thiết cho chương trình. Cuối cùng, với một câu lệnh, chúng ta sẽ tạo và chạy tất cả dịch vụ từ các thiết lập đó.

**Cách sử dụng Compose**

**Khai báo các môi trường trong Dockerfile:** Mỗi container có thể có một Dockerfile riêng, trong đó khai báo cách xây dựng image cho container đó. Dockerfile sẽ chứa các bước để cài đặt và cấu hình các phần mềm cần thiết cho container.

**Khai báo các service trong docker-compose.yml**: Docker Compose sử dụng file docker-compose.yml để mô tả các service. Mỗi service có thể liên kết với một container, được cấu hình chi tiết với các tham số như image, environment variables, network, volume,…

**Chạy câu lệnh docker-compose up**: Khi mọi thứ đã được cấu hình xong, bạn chỉ cần chạy câu lệnh docker-compose up để tạo và khởi động tất cả các container theo cấu hình trong file docker-compose.yml. Docker Compose sẽ tự động tạo các mạng riêng cho các container, quản lý các dependencies giữa các service và khởi động các container đúng thứ tự.

**Những lợi ích khi sử dụng Compose**

* Tạo ra nhiều môi trường độc lập
* Chỉ tạo lại các container đã thay đổi
* Điều chỉnh các biến sử dụng cho các môi trường
* Tự động hóa quá trình khởi tạo
* Tiết kiệm tài nguyên
* Dễ dàng xóa hoặc dừng các container

## 2.5. Ngôn ngữ PHP

PHP là ngôn ngữ lập trình rất thông dụng, được dùng để phát triển website và các ứng dung mã nguồn mở, hoạt động ở trên máy chủ. Ngôn ngũ lập trình PHP được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn. PHP dễ học nên nhanh chóng trở thành ngôn ngữ lập trình phổ biến nhất thế giới.

**Ưu điểm ngôn ngữ PHP**

* Dễ học và sử dụng
* Phát triển web nhanh chóng
* Tính tương thích cao
* Chi phí thấp
* Cộng đồng hỗ trợ lớn
* Quản lý dễ dàng trên máy chủ

**Nhược điểm ngôn ngữ PHP**

* Không có tính hỗ trợ đối tượng mạnh mẽ như các ngôn ngữ khác
* Tốc độ chậm so với các ngôn ngữ khác
* Bảo mật không cao nếu không cẩn thận
* Quá nhiều phiên bản và sự không tương thích

**Lý do nên sử dụng ngôn ngữ PHP**

Dễ học: PHP có cú pháp đơn giản và dễ học, đặc biệt là đối với những người đã quen thuộc với các ngôn ngữ lập trình phổ biến khác như C hoặc JavaScript. Điều này giúp các lập trình viên mới nhanh chóng có thể bắt tay vào phát triển ứng dụng.

Sử dụng rộng rãi: PHP là một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng bởi hàng triệu trang web và ứng dụng. Các nền tảng nổi tiếng như WordPress, Facebook, và Wikipedia đều sử dụng PHP. Điều này chứng tỏ tính ổn định và khả năng mở rộng của ngôn ngữ.

Chi phí thấp: PHP giúp giảm chi phí phát triển và duy trì ứng dụng web. Là ngôn ngữ mã nguồn mở, PHP có sẵn miễn phí và có thể chạy trên bất kỳ máy chủ nào mà không cần phần mềm đắt tiền hoặc giấy phép bản quyền.

Cộng đồng lớn: PHP có cộng đồng lập trình viên rất lớn và năng động. Bạn có thể dễ dàng nhận được sự hỗ trợ từ cộng đồng qua các diễn đàn, tài liệu, video hướng dẫn hoặc các hội nhóm chuyên sâu. Điều này giúp bạn nhanh chóng giải quyết các vấn đề gặp phải trong quá trình phát triển.

## 2.6. Ngôn ngữ HTML

### 2.6.1 Giới thiệu

HTML là viết tắt của Hypertext Markup Language. HTML dùng để đánh dấu dữ liệu trong các thẻ, định nghĩa dữ liệu và mô tả mục đích của nó với trang web. HTML có chức năng giúp người dùng xây dựng và cấu trúc các phần trong trang web hoặc ứng dụng, thường được sửu dụng trong phân chia các đoạn văn, hình ảnh, …

### 2.6.2. Cấu trúc

HTML có nhiều dạng thẻ khác nhau, mỗi thẻ sẽ có những nhiệm vụ khác nhau và ý nghĩa nhất định, ảnh hưởng và tác động đến nhau.

Về cơ bản, cấu trúc của HTML thường có ba phần:

*Phần khai báo*: có cấu trúc là <!Doctype>. Phần này cho người dùng biết được trình duyện đang sử dụng hiện đang dùng phiên bản HTML nào.

*Phần tiêu đề:* Phần khai báo ban đầu, khai báo về về meta, little, javascript, css... Phần này có cấu trúc bắt đầu bằng thẻ <head> và kết thúc bởi thẻ </head>. Đây là phần chứa tiêu đề và tiêu đề sẽ được hiển thị trên thanh điều hướng của trang web.

*Phần thân:* Chứa nội dung của trang web, là nơi hiển thị nội dung của trang web. Phần thân bắt đầu bằng thẻ <body> và kết thúc bằng thẻ </body>.

## 2.7. Giới thiệu về CSS

### 2.7.1. Giới thiệu

CSS là viết tắt của Cascading Style Sheets, được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi HTML. Nói ngắn gọn hơn CSS là ngôn ngữ tạo phong cách cho trang web.

Phương thức hoạt động của CSS là tìm dựa vào các vùng chọn, vùng chọn có thể là tên một thẻ HTML, tên một ID, class hay nhiều kiểu khác. Sau đó là nó sẽ áp dụng các thuộc tính cần thay đổi lên vùng chọn đó.

### 2.7.2. Phân loại

Để CSS có thể thực thi trên website hoặc HTML Documents thì phải tiến hành nhúng CSS vào Website. Có 3 cách nhúng CSS vào website:

*Inline*: Nhúng trực tiếp vào HTML thông qua thuộc tính style của phần tử.

*Internal*: Dùng thẻ <style> bên trong thẻ <head> của HTML để tạo ra nơi viết mã CSS.

*External*: Tạo một tập tin .css riêng và nhúng vào tài liệu HTML thông qua cặp thẻ <link>

**Ưu điểm:**

Dễ dàng quản lý và chỉnh sửa cho một tài liệu HTML duy nhất.

Phù hợp với các dự án nhỏ hoặc các trang web đơn giản.

**Nhược điểm:**

Không thể chia sẻ mã CSS giữa các trang khác nhau. Mã CSS sẽ bị nhúng trực tiếp vào mỗi tài liệu HTML, khiến cho tài liệu HTML trở nên nặng nề hơn.

## 2.8. Giới thiệu về JavaScript

JavaScript, viết tắt là JS, là một ngôn ngữ lập trình phổ biến được sử dụng chủ yếu để tạo ra các trang web tương tác. Với sự phát triển mạnh mẽ, JavaScript đã trở thành một thành phần thiết yếu trong việc phát triển ứng dụng web, giúp người dùng có trải nghiệm tốt hơn khi tương tác với trang web.

JavaScript được tích hợp và nhúng vào HTML, cho phép các nhà phát triển tạo ra những trang web sống động và đầy tính năng. Nó không chỉ cho phép người dùng lưu trữ các giá trị bên trong các biến mà còn thao tác trên các đoạn văn bản và thực thi các chương trình để phản hồi lại các sự kiện người dùng, như nhấp chuột, di chuyển chuột, hoặc nhập liệu.

JavaScript có thể được thêm vào trang web theo ba cách:

**Inline**: JavaScript được nhúng trực tiếp vào trong thẻ HTML thông qua thuộc tính onclick, onchange, hoặc các sự kiện khác của các phần tử HTML.

**Internal**: JavaScript được nhúng trực tiếp vào trong phần tử <script> bên trong thẻ <head> hoặc <body> của HTML.

**External**: JavaScript được lưu trong một tệp .js riêng biệt và được liên kết với tài liệu HTML thông qua thẻ <script src="file.js"></script>.

### 2.8.1. Ưu điểm của JavaScript

* Chương trình rất dễ học.
* Những lỗi Javascript rất dễ để phát hiện, từ đó giúp bạn sửa lỗi một cách nhanh chóng hơn.
* Giao diện của ứng dụng phong phú
* Tốc độ nhanh
* Hỗ trợ công nghệ web tiên tiến

### 2.8.2.Nhược điểm của JavaScript

* Những thiết bị khác nhau có thể sẽ thực hiện JS khác nhau, từ đó dẫn đến sự không đồng nhất
* Không thể truy cập tất cả tài nguyên máy tính
* Dễ bị khai thác, chèn mã độc
* Không được hỗ trợ khi thiết bị chưa được kết nối mạng

# HIỆN THỰC HÓA NGHIÊN CỨU

## Phân tích yêu cầu hệ thống

### Các chức năng của người sử dụng

* Chức năng quản lý
* Đăng nhập hệ thống
* Xem lịch trình
* Chức năng khách hàng
* Đăng nhập
* Xem các thông tin cơ bản về nhà xe
* Tìm chuyến xe, tuyến xe
* Đặt vé xe

### Tác nhân

* Khách hàng

Có quyền tìm kiếm, xem thông tin về lịch trình xe chạy. Quyền đặt vé, hủy vé, chirh sửa thông tin vé của mình.

* Nhóm người qunar lý

Có quyền cập nhật thông tin về lochj trình xe chạy, tạo mới, hủy lịch trình. Cập nhật thông tin về các loại xe. Đặt hủy sửa thông tin.

### Các yêu cầu phi chức năng

- Hệ thống chưa phục vụ tốt 24/24

- Chức năng bảo mật tốt

## Mô hình hóa dữ liệu

**Tên nhà xe:** "tennhaxe"

**Địa chỉ:** "diachi"

**Số điện thoại:** "sodienthoai"

**Danh sách xe:** "Xe", trong đó mỗi xe có các thuộc tính:

* **Mã số hiệu xe:** "sohieuxe"
* **Biển số xe:** "biensoxe"
* **Loại xe:** "loaixe"
* **Danh sách chuyến xe:** "ChuyenXe", mỗi chuyến xe có các thuộc tính:
  + **Mã chuyến xe:** "machuyen"
  + **Điểm xuất phát:** "diemxuatphat"
  + **Điểm đến:** "diemden"
  + **Thời gian khởi hành:** "thoigiankhoihanh"
  + **Trạng thái chuyến xe:** "trangthai"
  + **Danh sách vé xe:** "VeXe", trong đó mỗi vé có các thuộc tính:
    - **Tên hành khách:** "tenhanhkhach"
    - **Số điện thoại hành khách:** "sodienthoai"
    - **Số lượng vé:** "soluongve"
  + **Ngày khởi hành:** "ngaykhoihanh"

### Dữ liệu mẫu thử

* Thông tin nhà xe:
* Nhà xe Kim Hoàng
* Địa chỉ: Số 91, Phạm Hồng Thái, Trà Vinh
* Số điện thoại: 02839242424
* Dữ liệu bao gồm thông tin 3 xe:

Xe 1:

* Số hiệu xe: XE001
* Biển số xe: 84B-12345
* Loại xe: Ghế ngồi

*Bảng 1: Danh sách chuyến xe XE001*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã Chuyến | Điểm xuất phát | Điển đến | Thời gai khởi hành | Trạng Thái |
| CH001 | Trà Vinh | TP.HCM | 01:00 | Còn vé |
| CH002 | Trà Vinh | TP.HCM | 09:00 | Còn vé |
| CH003 | Trà Vinh | TP.HCM | 17:00 | Còn vé |

Xe 2:

* Số hiệu xe: XE002
* Biển số xe: 84B-23456
* Loại xe: Giường nằm

*Bảng 2: Danh sách chuyến xe XE002*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã Chuyến | Điểm xuất phát | Điển đến | Thời gai khởi hành | Trạng Thái |
| CH004 | Trà Vinh | TP.HCM | 01:00 | Còn vé |
| CH005 | Trà Vinh | TP.HCM | 09:00 | Còn vé |
| CH006 | Trà Vinh | TP.HCM | 17:00 | Còn vé |

Xe 3:

* Số hiệu xe: XE003
* Biển số xe: 29B-34567
* Loại xe: Ghế ngồi

Danh sách chuyến xe XE001

*Bảng 3: Danh sách chuyến xe XE003*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mã Chuyến | Điểm xuất phát | Điển đến | Thời gai khởi hành | Trạng Thái |
| CH007 | Trà Vinh | TP.HCM | 03:00 | Còn vé |
| CH008 | Trà Vinh | TP.HCM | 11:00 | Còn vé |
| CH009 | Trà Vinh | TP.HCM | 19:00 | Còn vé |

*Bảng 4: Bảng thông tin chung của Nhà Xe*

|  |  |
| --- | --- |
| Trường | Mô tả |
| \_id | Mã định danh duy nhất của nhà xe (ObjectId). |
| tennhaxe | Tên nhà xe (Ví dụ: "Nhà Xe Kim Hoàng"). |
| diachi | Địa chỉ của nhà xe (Ví dụ: "Số 91, Phạm Hồng Thái, Trà Vinh"). |
| sodienthoai | Số điện thoại liên hệ của nhà xe. |
| Xe | Mảng chứa danh sách thông tin các xe thuộc nhà xe. |

*Bảng 5: Bảng thông tin của Xe*

|  |  |
| --- | --- |
| Trường | Mô tả |
| sohieuxe | Số hiệu xe (Ví dụ: "XE001", "XE002"). |
| biensoxe | Biển số xe (Ví dụ: "84B-12345"). |
| loaixe | Loại xe (Ví dụ: "Ghế ngồi", "Giường nằm"). |
| ChuyenXe | Mảng chứa danh sách thông tin các chuyến xe thuộc từng xe. |

*Bảng 6: Bảng thông tin Chuyến Xe*

|  |  |
| --- | --- |
| Trường | Mô tả |
| machuyen | Mã chuyến xe (Ví dụ: "CH001", "CH002"). |
| diemxuatphat | Điểm xuất phát của chuyến xe (Ví dụ: "Trà Vinh"). |
| diemden | Điểm đến của chuyến xe (Ví dụ: "TP.HCM"). |
| thoigiankhoihanh | Thời gian khởi hành của chuyến xe (Ví dụ: "01:00", "09:00"). |

*Bảng 7: Bảng thông tin Vé Xe (trong từng Chuyến Xe)*

|  |  |
| --- | --- |
| Trường | Mô tả |
| tenhanhkhach | Tên hành khách đã đặt vé (Ví dụ: "Nguyễn Văn A"). |
| sodienthoai | Số điện thoại liên hệ của hành khách (Ví dụ: "0123456789"). |

### Đặc tả yêu cầu phi chức năng

Hệ thống chỉ hoạt động trong khung giờ nhất định, chưa phục vụ, đáp ứng các thắc mắc của khách hàng 24/24

Thời gian bảo trì dự kiến cần được thông báo trước.

Chức năng bảo mật tốt

Dữ liệu người dùng chưa được mã hóa tốt

Phân quyền người dùng chưa rõ ràngThiết kế hệ thống

### Thiết kế dữ liệu

### Thiết kế xử lý

#### Mô hình DFD mức ngữ cảnh

#### Mô hình DFD cấp 1

#### Mô hình DFD cấp 2

### Thiết kế giao diện

#### Sơ đồ website

#### Giao diện trang …

#### Giao diện …

# KẾT QUẢ NGHIÊN CỨU

## Bộ dữ liệu thử nghiệm

{

"\_id": {

"$oid": "67628cd1f5f8380569cbdde3"

},

"tennhaxe": "Nhà Xe Kim Hoàng",

"diachi": "Số 91, Phạm Hồng Thái, Trà Vinh",

"sodienthoai": "02839242424",

"Xe": [

{

"sohieuxe": "XE001",

"biensoxe": "84B-12345",

"loaixe": "Ghế ngồi",

"ChuyenXe": [

{

"machuyen": "CH002",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "09:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "02-01-2025"

},

{

"machuyen": "CH003",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "17:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "03-01-2025"

},

{

"machuyen": "CH001",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "01:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "01-01-2025"

},

{

"machuyen": "CH010",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "02:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "01-01-2025"

},

{

"machuyen": "CH011",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "03:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "01-01-2025"

}

]

},

{

"sohieuxe": "XE002",

"biensoxe": "84B-23456",

"loaixe": "Giường nằm",

"ChuyenXe": [

{

"machuyen": "CH004",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "01:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "04-01-2025"

},

{

"machuyen": "CH005",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "09:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "05-01-2025"

},

{

"machuyen": "CH006",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "17:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "06-01-2025"

}

]

},

{

"sohieuxe": "XE003",

"biensoxe": "29B-34567",

"loaixe": "Ghế ngồi",

"ChuyenXe": [

{

"machuyen": "CH007",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "03:00",

"trangthai": "Còn vé",

"VeXe": [

{

"tenhanhkhach": "Nguyễn Văn A",

"sodienthoai": "0123456789",

"soluongve": 2

},

{

"tenhanhkhach": "Trần Thị A",

"sodienthoai": "0987654321",

"soluongve": 1

}

],

"ngaykhoihanh": "07-01-2025"

},

{

"machuyen": "CH008",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "11:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "08-01-2025"

},

{

"machuyen": "CH009",

"diemxuatphat": "Trà Vinh",

"diemden": "TP.HCM",

"thoigiankhoihanh": "19:00",

"trangthai": "Còn vé",

"VeXe": [],

"ngaykhoihanh": "09-01-2025"

}

]

}

]

}

## Kết quả thực nghiệm

### Chức năng …

### Chức năng …

### Chức năng …

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Hệ thống đặt vé xe khách trực tuyến cho một số doanh nghiệp dịch vụ xe khách tại tỉnh Trà Vinh đã được xây dựng thành công với cơ sở dữ liệu NoSQL. Dựa trên nền tảng này, có thể phát triển thêm các ứng dụng web và di động để người dùng dễ dàng truy cập, tìm kiếm và đặt vé. Hệ thống cũng có thể tích hợp thêm các chức năng như đánh giá dịch vụ, hỗ trợ khách hàng và đa dạng hóa các phương thức thanh toán, nhằm nâng cao trải nghiệm người dùng và tối ưu hóa quy trình quản lý cho các doanh nghiệp.

## Hướng phát triển

**TÀI LIỆU THAM KHẢO**

|  |  |
| --- | --- |
| [1] | N. N. Đ. Thanh, Kỹ thuật lập trình, Trà Vinh: Đại học Cần Thơ, 2014. |
| [2] | "Lập trình web cơ bản với PHP (P2)," 4 1 2024. [Online]. Available: https://topdev.vn/. |