

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Trà Vinh, ngày tháng năm
Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

ThS. Phạm Minh Dương

NHẬN XÉT CỦA THÀNH VIÊN HỘI ĐỒNG

This image shows a full page of a notebook or worksheet template. It consists of approximately 30 horizontal rows. Each row is defined by two parallel dotted lines, creating a series of uniform spaces for handwriting practice. The lines are evenly spaced and extend across the entire width of the page. There are no margins, text, or other markings present.

Trà Vinh, ngày tháng năm
Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Đầu tiên, tôi xin gửi lời cảm ơn chân thành đến Khoa Kỹ thuật và Công nghệ, Bộ môn Công nghệ Thông tin đã tạo điều kiện thuận lợi cho tôi trong suốt quá trình học tập và hoàn thành đồ án thực tập chuyên ngành. Tiếp theo, tôi xin bày tỏ lòng biết ơn sâu sắc đến thầy ThS. Phạm Minh Dương đã truyền đạt kiến thức và hướng dẫn tôi trong quá trình làm đồ án.

Tôi đã cố gắng vận dụng những kiến thức đã học được trong các học kỳ qua để hoàn thành đồ án. Kết quả đạt được là cả quá trình nỗ lực của tôi, tuy nhiên vẫn không tránh khỏi những thiếu sót trong quá trình nghiên cứu và trình bày. Rất kính mong sự góp ý của quý thầy (cô) để đồ án thực tập chuyên ngành của tôi được hoàn thiện hơn.

Xin trân trọng cảm ơn!

Sinh viên thực hiện

Dương Văn Hiệp

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN.....	1
1.1 Tổng quan về chủ đề.....	1
1.2 Các ứng dụng tương tự hiện có	2
1.3 Các công nghệ sử dụng.....	2
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT	4
2.1 Giới thiệu về NextJS.....	4
2.1.1 Tổng quan về NextJS.....	4
2.1.2 Ưu nhược điểm của NextJS	4
2.1.3 Cấu trúc của NextJS.....	5
2.1.4 Page Router và App Router trong NextJS	6
2.2 Giới thiệu về ASP.NET Core	7
2.2.1 Tổng quan về ASP.NET Core	7
2.2.2 Các đặc tính quan trọng của ASP.NET Core.....	8
2.2.3 Các nhánh của ASP.NET.....	8
2.2.4 ASP.NET và ASP.NET Core.....	8
2.2.5 Ưu nhược điểm của ASP.NET Core.....	9
2.3 Giới thiệu về SQL Server	10
2.3.1 Tổng quan về SQL Server	10
2.3.2 Ưu nhược điểm của SQL Server.....	10
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	11
3.1 Mô tả bài toán.....	11
3.2 Đặc tả các yêu cầu chức năng.....	11
3.2.1 Yêu cầu chức năng.....	11
3.2.2 Yêu cầu phi chức năng.....	12
3.3 Thiết kế dữ liệu.....	13
3.3.1 Lược đồ cơ sở dữ liệu	13
3.3.2 Danh sách các thực thể	13
3.3.3 Chi tiết các thực thể	13
3.4 Thiết kế giao diện	13
3.4.1 Sơ đồ website.....	13
3.4.2 Giao diện website	13
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	14
4.1 Dữ liệu thử nghiệm.....	14
4.2 Kết quả thử nghiệm	14
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	15
5.1 Kết luận	15
5.2 Hướng phát triển.....	15
DANH MỤC TÀI LIỆU THAM KHẢO	16
PHỤ LỤC.....	17

DANH MỤC HÌNH ẢNH - BẢNG BIỂU

No table of figures entries found.

Bảng 1 So sánh Page Router và App Router	6
Bảng 2 So sánh giữa APS.NET và ASP.NET Core.....	8

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Vấn đề nghiên cứu

Đề tài tập trung xây dựng một hệ thống quản lý và đặt vé xe cho hãng xe khách, bao gồm các chức năng quản lý cho nội bộ và chức năng dành cho khách hàng. Sử dụng công nghệ .NET Core và NextJS để phát triển. Vấn đề trọng tâm là tích hợp được các công nghệ hiện đại nhằm cung cấp trải nghiệm người dùng tốt, bảo đảm hiệu suất cao và khả năng mở rộng cho hệ thống.

Hướng tiếp cận

Frontend: Xây dựng giao diện người dùng bằng NextJS bao gồm cả trang chủ và trang quản trị. Giao diện được thiết kế thân thiện với người dùng, đơn giản dễ dàng sử dụng, hỗ trợ responsive tối ưu giao diện cho nhiều màn hình thiết bị.

Backend: Phát triển bằng .Net Core, cụ thể là ASP.NET Core Web API, xây dựng các API cho hệ thống, kết hợp với hệ quản trị cơ sở dữ liệu Sql Server cho hiệu suất cao.

Cách giải quyết vấn đề

Tiến hành phân tích yêu cầu nhằm xác định các tính năng cần thiết gồm cả các tính năng cho khách hàng như đặt vé xe, tra cứu thông tin và các tính năng cho quản trị quản lý các xe, chuyến đi, quản lý khách hàng,... Dựa vào đó, thiết kế cơ sở dữ liệu, xây dựng API, thiết kế và xây dựng giao diện với các chức năng đã đặt ra sử dụng các công nghệ đã chọn.

Một số kết quả đạt được

Dự án hoàn thiện với các tính năng cần thiết cho một hệ thống đặt vé xe cho hãng xe khách bao gồm trang người dùng và trang quản trị, tối ưu hóa trải nghiệm người dùng. Backend sử dụng ASP.Net Core Web API đảm bảo tính an toàn về bảo mật cùng với tốc độ xử lý mạnh mẽ. Triển khai hệ thống lên nền tảng Điện toán đám mây, giúp người dùng có thể truy cập nhanh chóng và thuận tiện.

MỞ ĐẦU

Lý do chọn đề tài

Trong giai đoạn chuyển đổi số hiện nay, ứng dụng công nghệ thông tin ngày càng phát triển trên mọi lĩnh vực, trong đó nhu cầu tra cứu và đặt trực tuyến các loại vé nói chung và đặt vé xe khách nói riêng là một tất yếu. Với hệ thống giao thông vận tải của quốc gia đang trên đà phát triển, các tuyến xe khách cũng đang ngày mở rộng, nhu cầu di chuyển giữa các tỉnh thành của người dân ngày càng tăng thì việc phát triển một ứng dụng công nghệ thông tin để đáp ứng các nhu cầu đó là rất cần thiết.

Mục đích nghiên cứu

Mục đích xây dựng một hệ thống đặt vé xe khách dành cho một hãng xe cụ thể, cung cấp các tính năng cho người dùng đặt vé cũng như tra cứu, thanh toán trực tuyến và quản lý dễ dàng cho phía quản trị. Tích hợp các công nghệ hiện đại và mạnh mẽ như .NET Core và NextJS, xây dựng trải nghiệm người dùng tốt, dễ dàng truy cập và sử dụng.

Đối tượng nghiên cứu

Đối tượng nghiên cứu là các công nghệ chính được sử dụng để xây dựng hệ thống bao gồm .NET Core cụ thể là ASP.NET Core Web API, hệ quản trị cơ sở dữ liệu Sql Server, NextJS cùng với một số thư viện hỗ trợ liên quan khác. Đề tài tập trung vào cách kết hợp các công nghệ này lại với nhau để tạo nên một hệ thống hiệu quả, đáp ứng yêu cầu của người dùng, dễ dàng triển khai, nâng cấp và bảo trì.

Phạm vi nghiên cứu

Giới hạn trong việc xây dựng một hệ thống website đặt vé xe khách và chức năng quản lý cho phía hãng xe. Bao gồm các nội dung như phân tích yêu cầu, thiết kế và triển khai hệ thống với các chức năng thiết yếu như đặt vé xe, tra cứu hành trình, thanh toán trực tuyến, quản lý xe, quản lý chuyến đi, quản lý khách hàng.

CHƯƠNG 1. TỔNG QUAN

1.1 Tổng quan về chủ đề

Nhằm tạo ra một nền tảng dành riêng cho một hãng xe cụ thể, cung cấp các tính năng tiện ích như tra cứu tuyến xe, đặt vé, thanh toán trực tuyến, đồng thời hỗ trợ quản lý hiệu quả cho phía quản trị viên. Hệ thống này sẽ đóng vai trò quan trọng trong việc nâng cao chất lượng dịch vụ vận tải, đáp ứng tốt hơn nhu cầu của khách hàng cũng như cải thiện hiệu quả vận hành cho doanh nghiệp.

Tập trung vào việc nghiên cứu với trọng tâm chính gồm các khía cạnh sau:

Phân tích nhu cầu và nghiệp vụ đặt vé xe khách: Tìm hiểu quy trình đặt vé truyền thống và các vấn đề mà cả hành khách lẫn nhà xe thường gặp phải. Từ đó, xác định các yêu cầu nghiệp vụ cần thiết như tra cứu tuyến xe, chọn ghế ngồi, đặt vé, thanh toán trực tuyến, và quản lý vé cho nhà xe.

Thiết kế và triển khai hệ thống:

Phía Client: Xây dựng giao diện thân thiện, dễ sử dụng cho khách hàng đặt vé và quản trị viên quản lý.

Phía Backend: Tập trung vào việc thiết kế cơ sở dữ liệu và triển khai các API để xử lý dữ liệu, đảm bảo hệ thống hoạt động ổn định, bảo mật và hiệu quả.

Tích hợp công nghệ hiện đại: Đồ án sẽ tập trung nghiên cứu việc áp dụng các công nghệ tiên tiến như .NET Core để xây dựng backend mạnh mẽ và NextJS để phát triển frontend. Sự kết hợp này nhằm tối ưu hóa hiệu suất, tính bảo mật, và khả năng mở rộng của hệ thống.

Giải pháp thanh toán trực tuyến: Nghiên cứu và tích hợp các cổng thanh toán phổ biến, đảm bảo quy trình thanh toán diễn ra nhanh chóng, an toàn và phù hợp với người dùng Việt Nam.

Trải nghiệm người dùng: Nghiên cứu các yếu tố để tối ưu hóa trải nghiệm người dùng, bao gồm hiệu suất tải trang, thiết kế giao diện trực quan, thân thiện và khả năng truy cập dễ dàng trên nhiều thiết bị.

Quản lý và vận hành hệ thống: Tìm hiểu các chức năng quản lý dành cho nhà xe, bao gồm quản lý lịch trình, ghế trống, báo cáo doanh thu, và thống kê dữ liệu khách hàng để cải thiện hiệu quả hoạt động.

1.2 Các ứng dụng tương tự hiện có

...

1.3 Các công nghệ sử dụng

.Net Core: Một nền tảng phát triển ứng dụng mã nguồn mở, đa nền tảng do Microsoft phát triển, được sử dụng rộng rãi trong việc xây dựng các ứng dụng web và API mạnh mẽ. Với hiệu suất cao, khả năng mở rộng tốt và tích hợp nhiều thư viện hỗ trợ, .NET Core cung cấp một nền tảng vững chắc để phát triển backend cho hệ thống. Đặc biệt, tính năng hỗ trợ đa nền tảng giúp hệ thống có thể triển khai trên Windows, Linux hoặc macOS, mang lại tính linh hoạt và tiết kiệm chi phí. Ngoài ra, các công cụ tích hợp của .NET Core giúp tối ưu hóa quá trình phát triển và bảo trì ứng dụng, đảm bảo hiệu quả vận hành trong dài hạn.

SQL Server: hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, được biết đến với khả năng xử lý khối lượng lớn dữ liệu và bảo mật cao. Đây là một lựa chọn lý tưởng để lưu trữ thông tin liên quan đến tuyến xe, thông tin người dùng, vé đặt và các giao dịch thanh toán. SQL Server cung cấp các tính năng như tối ưu hóa truy vấn, quản lý giao dịch hiệu quả và bảo mật dữ liệu ở mức độ cao. Tích hợp với .NET Core, SQL Server giúp đảm bảo tính ổn định và hiệu năng trong việc quản lý và truy xuất dữ liệu của hệ thống.

NextJS: Một framework phát triển giao diện người dùng mạnh mẽ dựa trên React, cung cấp khả năng kết xuất phía máy chủ (Server-side Rendering) và tạo trang tĩnh (Static Generation). NextJS giúp tối ưu hóa tốc độ tải trang, cải thiện SEO và mang lại trải nghiệm người dùng mượt mà. Với khả năng tích hợp linh hoạt, framework này giúp xây dựng giao diện trực quan, tương tác cao và dễ sử dụng trên nhiều loại thiết bị. Ngoài ra, NextJS hỗ trợ quản lý state hiệu quả, định tuyến động và tích hợp API, giúp frontend hoạt động trơn tru và dễ dàng kết nối với backend xây dựng bằng .NET Core.

Sự kết hợp giữa .NET Core, SQL Server và NextJS tạo nên một hệ thống toàn diện, vừa mạnh mẽ, ổn định về mặt xử lý dữ liệu, vừa thân thiện và tối ưu về trải nghiệm người dùng.

CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

2.1 Giới thiệu về NextJS

2.1.1 Tổng quan về NextJS

NextJS là một framework mã nguồn mở được phát triển dựa trên React, cho phép xây dựng các ứng dụng web với nhiều tính năng tối ưu hóa, bao gồm Server-Side Rendering (SSR) và Static Site Generation (SSG). Được phát triển bởi công ty Vercel, NextJS giúp tăng cường hiệu suất và khả năng SEO cho các ứng dụng web bằng cách cho phép máy chủ xử lý dữ liệu và tạo HTML trước khi gửi đến trình duyệt.

2.1.2 Ưu nhược điểm của NextJS

Ưu điểm:

Tối ưu hóa SEO: Hỗ trợ Server-Side Rendering (SSR) và Static Site Generation (SSG), giúp các trang web được lập chỉ mục dễ dàng hơn, cải thiện thứ hạng trên các công cụ tìm kiếm.

Hiệu suất cao: NextJS cho phép tạo ra các trang web tải nhanh và mượt mà nhờ các tính năng như SSG và SSR, mang lại trải nghiệm người dùng tốt hơn.

Hệ thống Route tự động: Tự động tạo các route dựa trên cấu trúc thư mục, giúp việc quản lý và phát triển ứng dụng trở nên dễ dàng hơn.

Hỗ trợ CSS và JavaScript hiện đại: Cho phép nhập tệp CSS trực tiếp trong tệp JavaScript và hỗ trợ các tính năng JavaScript hiện đại, giúp phát triển ứng dụng nhanh chóng và hiệu quả.

Tích hợp API routes: NextJS cho phép tạo các API routes để xử lý các yêu cầu HTTP, giúp xây dựng các API RESTful một cách dễ dàng.

Nhược điểm:

Giới hạn trong hệ thống định tuyến: NextJS bị giới hạn việc chỉ sử dụng bộ định tuyến dựa trên cấu trúc tệp, không thể tùy chỉnh cách xử lý các route.

Hệ sinh thái và thư viện còn hạn chế: ít thư viện hơn so với các framework khác, có thể gây khó khăn khi tìm kiếm giải pháp cho các yêu cầu cụ thể.

2.1.3 Cấu trúc của NextJS

Cấu trúc thư mục hiện tại của NextJS như sau:

```
my-nextjs-app/  
|- .next/  
|- node_modules/  
|- public/  
|- src/  
|   |- app/  
|   |- components/  
|   |- styles/  
|   |- utils/  
|- package.json  
|- next.config.js  
|- README.md
```

Trong đó:

.next/: Thư mục này được tạo ra sau khi build dự án, chứa các tệp liên quan đến quá trình biên dịch và tối ưu hóa.

node_modules/: Chứa các module và thư viện mà dự án phụ thuộc, được quản lý bởi npm hoặc yarn.

public/: Lưu trữ các tệp tĩnh như hình ảnh, favicon, và các tài nguyên khác. Các tệp trong thư mục này có thể được truy cập trực tiếp thông qua đường dẫn URL.

src/: Thư mục gốc cho mã nguồn ứng dụng, giúp tổ chức code một cách rõ ràng và dễ quản lý.

app/: Chứa các thành phần ứng dụng và định tuyến. Trong NextJS, có thể tạo các trang bằng cách thêm các tệp vào thư mục này, NextJS sẽ tự động thiết lập các route dựa trên cấu trúc thư mục.

components/: Lưu trữ các component React dùng chung trong ứng dụng, giúp tái sử dụng và quản lý code hiệu quả.

styles/: Chứa các tệp CSS hoặc các giải pháp styling khác cho ứng dụng, giúp tách biệt phần giao diện và logic.

utils/: Lưu trữ các hàm tiện ích, helper functions, và logic chung được sử dụng trong nhiều phần của ứng dụng.

package.json: Tập này chứa thông tin về dự án, bao gồm các gói phụ thuộc, script, và các cấu hình khác.

next.config.js: Tập cấu hình cho NextJS, cho phép tùy chỉnh các thiết lập mặc định của framework theo nhu cầu dự án.

README.md: Tập mô tả dự án, thường được sử dụng để cung cấp thông tin tổng quan, hướng dẫn cài đặt và sử dụng.

Ngoài ra, còn có thể tạo thêm cái thư mục khác tùy theo nhu cầu thực tế của từng dự án.

2.1.4 Page Router và App Router trong NextJS

Trong Next.js, có hai hệ thống định tuyến chính: Pages Router và App Router.

Bảng 1 So sánh Page Router và App Router

	<i>Page Router</i>	<i>App Router</i>
Cấu trúc thư mục	Sử dụng thư mục pages/ , mỗi tệp tương ứng với một route.	Sử dụng thư mục app/ , tạo các route lồng nhau thông qua cấu trúc thư mục.
Thành phần	Mặc định sử dụng thành phần phía client.	Mặc định sử dụng các thành phần phía server, giúp tối ưu hiệu suất.
Gọi dữ liệu	Sử dụng các hàm getStaticProps , getServerSideProps , getInitialProps để lấy dữ liệu.	Sử dụng hàm fetch trực tiếp trong các thành phần để lấy dữ liệu.

Layouts	Layouts tĩnh, không hỗ trợ lồng nhau một cách linh hoạt.	Hỗ trợ layouts động và có thể lồng nhau, cho phép cấu trúc giao diện phức tạp hơn.
Server Components	Không hỗ trợ.	Có hỗ trợ, tận dụng lợi thế của React Server Components.

Lựa chọn giữa Pages Router và App Router:

Dự án mới: Nên sử dụng App Router để tận dụng các tính năng hiện đại và hiệu suất cao.

Yêu cầu đơn giản: Pages Router phù hợp với các dự án nhỏ, yêu cầu định tuyến đơn giản.

Yêu cầu phức tạp: App Router thích hợp cho các ứng dụng lớn với cấu trúc định tuyến và dữ liệu phức tạp.

2.2 Giới thiệu về ASP.NET Core

2.2.1 Tổng quan về ASP.NET Core

ASP.NET Core là một tập hợp các thư viện chuẩn như một framework để xây dựng ứng dụng web. ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET mà là một cái tên mới được xây dựng từ đầu. Có một sự thay đổi lớn về kiến trúc và kết quả là nó gọn hơn, phân chia module tốt hơn. ASP.NET Core có thể chạy trên cả .NET Core hoặc full .NET Framework.

.NET Core là môi trường thực thi. Nó được thiết kế lại hoàn toàn của .NET Framework. Mục tiêu chính của .NET Core là hỗ trợ phát triển ứng dụng đa nền tảng cho ứng dụng .NET. Nó được hỗ trợ trên Windows, MacOS và Linux. .NET Core là một framework mã nguồn mở được xây dựng và phát triển bởi Microsoft và cộng đồng .NET trên Github

Nó cũng triển khai đặc điểm của .NET Standard. .NET Standard là một đặc tả chuẩn của .NET API hướng tới hỗ trợ trên tất cả các triển khai của nền tảng .NET.

Nó định nghĩa một tập các quy tắc thống nhất cần thiết để hỗ trợ tất cả các ứng dụng trên nền .NET.

2.2.2 Các đặc tính quan trọng của ASP.NET Core

Có thể xây dựng và chạy ứng dụng ASP.NET đa nền tảng trên Windows, Mac và Linux.

ASP.NET Core hợp nhất ASP.NET MVC và ASP.NET Web API.

Có thể host trên IIS hoặc tự host.

Có sẵn Dependency Injection.

Dễ dàng tích hợp với các framework frontend như Angular, Knockout...

Hỗ trợ cấu hình cho nhiều môi trường.

Cơ chế HTTP Request pipeline mới.

Hỗ trợ quản lý phiên bản

Dùng chung toàn bộ Nuget Package.

2.2.3 Các nhánh của ASP.NET

Có hai nhánh của ASP.NET cho đến hiện tại là ASP.NET và ASP.NET Core. ASP.NET là phiên bản hiện tại của ASP.NET và nó cần .NET Framework để chạy.

ASP.NET Core là cách mới để xây dựng ứng dụng web. Nó có thể chạy cả trên .NET Framework và .NET Core.

2.2.4 ASP.NET và ASP.NET Core

Bảng 2 So sánh giữa ASP.NET và ASP.NET Core

ASP.NET	ASP.NET CORE
Nền tảng đã có từ lâu	Hoàn toàn được thiết kế mới
Chạy trên .NET Framework	Chạy trên cả .NET Core và .NET Framework
Chỉ trên Windows	Chạy trên tất cả các OS sử dụng .NET Core

Nền tảng ổn định với tính năng phong phú	Chưa hoàn chỉnh nhưng mong đợi sẽ hoàn chỉnh trong tương lai
WebForms được hỗ trợ	Không hỗ trợ WebForms
System.web.dll công kênh	Nhỏ, nhẹ và module hóa
Bản quyền của Microsoft	ASP.NET Core là mã nguồn mở

2.2.5 Ưu nhược điểm của ASP.NET Core

Ưu điểm:

Hiệu năng cao: ASP.NET Core được tối ưu hóa để cung cấp hiệu suất tốt hơn so với các phiên bản trước, đồng thời tối ưu hóa việc sử dụng hệ thống tài nguyên.

Đa nền tảng: Nhờ vào .NET Core, có thể xây dựng phần mềm trên Windows, Linux, macOS một cách dễ dàng.

Mã nguồn mở: ASP.NET Core là một mã nguồn mở, một xu thế mà các ngôn ngữ lập trình hiện nay hướng đến.

Tích hợp tốt với các framework phía client: ASP.NET Core được thiết kế tích hợp với nhiều client side frameworks một cách liên tục bao gồm AngularJS, ReactJS,...

Hỗ trợ triển khai trên đám mây: ASP.NET Core được thiết kế để tối ưu development framework cho những ứng dụng cái mà được chạy on-promise hay được triển khai trên đám mây.

Nhược điểm:

Số lượng thư viện hỗ trợ hạn chế: Có ít thư viện hơn so với .NET Framework có hệ sinh thái và nhiều thư viện hỗ trợ.

Yêu cầu công cụ phát triển: Nếu không có Visual Studio thì việc phát triển khó khăn hơn.

Không hỗ trợ Web Forms: ASP.NET Core không hỗ trợ cho Web Forms.

2.3 Giới thiệu về SQL Server

2.3.1 Tổng quan về SQL Server

SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) do Microsoft phát triển, được thiết kế để lưu trữ và quản lý dữ liệu một cách hiệu quả. Nền tảng này hỗ trợ ngôn ngữ truy vấn Transact-SQL (T-SQL), cho phép người dùng thực hiện các thao tác như tạo, đọc, cập nhật và xóa dữ liệu trong cơ sở dữ liệu. SQL Server được tối ưu để hoạt động trên các môi trường cơ sở dữ liệu lớn, có thể phục vụ hàng ngàn người dùng đồng thời và tích hợp tốt với các sản phẩm khác của Microsoft như Internet Information Services (IIS) và Visual Studio. Hệ thống này cung cấp các tính năng mạnh mẽ như bảo mật nâng cao, khả năng mở rộng và hiệu suất cao, đáp ứng nhu cầu của các ứng dụng doanh nghiệp và hệ thống xử lý dữ liệu phức tạp.

2.3.2 Ưu nhược điểm của SQL Server

Ưu điểm

Hiệu suất mạnh mẽ: SQL Server cung cấp khả năng tối ưu hóa truy vấn và tích hợp với các tính năng mở rộng như tích hợp dữ liệu với SSIS, phân tích dữ liệu với SSAS, và hỗ trợ tạo báo cáo với SSRS.

Quản lý dự án doanh nghiệp: SQL Server là sự lựa chọn hàng đầu cho các doanh nghiệp lớn với các yêu cầu quản lý dự án phức tạp.

Tài liệu học tập phong phú: Với sự phổ biến rộng rãi, SQL Server có nhiều tài liệu học tập, sách và tài liệu trực tuyến, giúp việc học và phát triển dự án trở nên dễ dàng.

Nhược điểm

Phụ thuộc vào hệ điều hành Windows: SQL Server được thiết kế tối ưu với hệ điều hành Windows, điều này có thể giới hạn sự linh hoạt trong việc triển khai trên các hệ điều hành khác.

Quản lý phức tạp: SQL Server có nhiều tính năng, nhưng điều này cũng có nghĩa rằng việc quản lý và cấu hình có thể trở nên phức tạp đối với người mới bắt đầu.

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả bài toán

Trong bối cảnh nhu cầu di chuyển ngày càng tăng cao, việc đặt vé xe khách theo phương thức truyền thống thường gặp nhiều bất cập như mất thời gian, khó khăn trong việc tra cứu thông tin tuyến xe, tình trạng quá tải hoặc thiếu minh bạch trong việc quản lý vé. Những hạn chế này không chỉ gây phiền toái cho hành khách mà còn làm giảm hiệu quả vận hành của các doanh nghiệp vận tải. Vì vậy, bài toán đặt ra là cần phát triển một hệ thống đặt vé xe khách trực tuyến, cho phép người dùng dễ dàng tra cứu thông tin tuyến xe, đặt vé, chọn ghế và thanh toán trực tuyến nhanh chóng, an toàn. Đồng thời, hệ thống cũng phải hỗ trợ nhà xe quản lý lịch trình, số lượng vé đã bán, ghế trống và các báo cáo doanh thu một cách hiệu quả, góp phần cải thiện chất lượng dịch vụ và tối ưu hóa hoạt động kinh doanh.

3.2 Đặc tả các yêu cầu chức năng

3.2.1 Yêu cầu chức năng

Dành cho khách hàng

Đăng ký và đăng nhập: Khách hàng có thể tạo tài khoản, đăng nhập để sử dụng các dịch vụ của hệ thống.

Tra cứu thông tin tuyến xe: Cho phép khách hàng tìm kiếm các tuyến xe dựa trên điểm đi, điểm đến, thời gian khởi hành, loại xe, và hãng xe.

Đặt vé: Cung cấp chức năng chọn tuyến xe, số ghế, và xác nhận đặt vé. Hệ thống phải hiển thị số ghế còn trống theo thời gian thực.

Thanh toán trực tuyến: Tích hợp phương thức thanh toán chuyển khoản và hệ thống tự động xác nhận để người dùng thanh toán nhanh chóng và an toàn.

Quản lý thông tin cá nhân và lịch sử đặt vé: Khách hàng có thể cập nhật thông tin cá nhân, xem và quản lý các vé đã đặt (bao gồm hủy vé hoặc thay đổi lịch trình, nếu cần).

Nhận thông báo: Hệ thống gửi thông báo qua email hoặc tin nhắn về tình trạng vé (đặt thành công, hủy, hoặc thay đổi).

Dành cho quản trị viên

Quản lý tài khoản: Cho phép thêm, chỉnh sửa, hoặc xóa tài khoản quản trị viên hoặc nhân viên hỗ trợ.

Quản lý tuyến xe: Cung cấp chức năng tạo mới, cập nhật, và xóa tuyến xe, bao gồm thông tin như điểm đi, điểm đến, lịch trình, giá vé, và số lượng ghế.

Quản lý vé và ghế ngồi: Hiện thị danh sách vé đã đặt, số ghế còn trống theo từng tuyến xe, cho phép kiểm tra và xác nhận vé.

Báo cáo và thống kê: Hỗ trợ tạo báo cáo doanh thu, số lượng vé bán ra, và tình trạng hoạt động của các tuyến xe theo ngày, tuần, tháng hoặc năm.

Quản lý cổng thanh toán: Cấu hình và theo dõi các giao dịch thanh toán trực tuyến, đảm bảo tính minh bạch và đồng bộ với hệ thống.

Hỗ trợ khách hàng: Cung cấp công cụ trả lời các thắc mắc của người dùng, xử lý các yêu cầu hủy vé hoặc thay đổi lịch trình.

3.2.2 Yêu cầu phi chức năng

Tính bảo mật: Hệ thống phải đảm bảo an toàn cho thông tin cá nhân và giao dịch của người dùng.

Tính khả dụng: Hệ thống cần hoạt động ổn định, đáp ứng lưu lượng truy cập lớn mà không bị gián đoạn.

Bảo đảm hiệu suất: Có khả năng xử lý lượng lớn truy cập cùng lúc, thời gian hệ thống phản hồi nhanh, không bị gián đoạn trong các giờ cao điểm.

Tính dễ sử dụng: Giao diện cần thân thiện, trực quan, dễ dàng sử dụng cho cả người dùng lần đầu.

Khả năng mở rộng: Hệ thống phải hỗ trợ mở rộng để tích hợp thêm các tính năng mới trong tương lai, như kết nối với nhiều hãng xe khác nhau hoặc bổ sung phương thức thanh toán mới.

3.3 Thiết kế dữ liệu

3.3.1 Lược đồ cơ sở dữ liệu

3.3.1.1 Mô hình thực thể kết hợp

3.3.1.2 Mô hình vật lý

3.3.2 Danh sách các thực thể

3.3.3 Chi tiết các thực thể

3.4 Thiết kế giao diện

3.4.1 Sơ đồ website

3.4.2 Giao diện website

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1 Dữ liệu thử nghiệm

4.2 Kết quả thử nghiệm

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.2 Hướng phát triển

DANH MỤC TÀI LIỆU THAM KHẢO

PHỤ LỤC