**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN MÔN HỌC**

**THIẾT KẾ VÀ XÂY DỰNG WEBSITE**

**ĐẶT VÉ XE TRỰC TUYẾN**

**Giảng viên hướng dẫn: ThS. ĐỖ HOÀNG NAM**

**Sinh viên thực hiện : HUỲNH NHẬT LONG**

**MSSV : 2200004430**

**Ngành : Kỹ thuật phần mềm**

**Chuyên ngành : Công nghệ Kỹ thuật phần mềm**

**Môn học : Chuyên đề chuyên sâu PTPM 1**

**Khóa : 2022**

**Tp.HCM, tháng 01 năm 2025**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH**

**KHOA CÔNG NGHỆ THÔNG TIN**



**TIỂU LUẬN MÔN HỌC**

**THIẾT KẾ VÀ XÂY DỰNG WEBSITE**

**ĐẶT VÉ XE TRỰC TUYẾN**

**Giảng viên hướng dẫn: ThS. ĐỖ HOÀNG NAM**

**Sinh viên thực hiện 1 : HUỲNH NHẬT LONG**

**MSSV : 2200004430**

**Sinh viên thực hiện 2 : ĐẶNG HỮU KHOA**

**MSSV : 2200003303**

**Ngành : Kỹ thuật phần mềm**

**Chuyên ngành : Công nghệ Kỹ thuật phần mềm**

**Môn học : Chuyên đề chuyên sâu PTPM 1**

**Khóa : 2022**

**Tp.HCM, tháng 01 năm 2025**

|  |  |
| --- | --- |
| Trường Đại học Nguyễn Tất Thành  **Khoa Công Nghệ Thông Tin**  🙜 🙜 🙝 | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập – Tự do – Hạnh phúc**  🙜 🙜 🙝 |

NHIỆM VỤ TIỂU LUẬN MÔN HỌC

Sinh viên thực hiện 1: **HUỲNH NHẬT LONG** MSSV: **2200004430**

Chuyên ngành: **KỸ THUẬT PHẦN MỀM** Lớp: **22DKTPM1B** Email: **2200004430@nttu.edu.vn** SĐT: **0777561284**

Sinh viên thực hiện 2: **ĐẶNG HỮU KHOA** MSSV: **2200003303**

Chuyên ngành: **KỸ THUẬT PHẦN MỀM** Lớp: **22DKTPM1A** Email: **2200003303@nttu.edu.vn** SĐT: **0859414317**

Tên đề tài: **THIẾT KẾ VÀ XÂY DỰNG WEBSITE ĐẶT VÉ XE TRỰC TUYẾN**

Giáo viên hướng dẫn: **ThS. Đỗ Hoàng Nam**

Thời gian thực hiện: **11/12/2024** **đến 17/01/2025**

Nhiệm vụ/nội dung (mô tả chi tiết nội dung, yêu cầu, phương pháp, … ) :

* Khảo sát hiện trạng và phân tích các phần mềm và các Website liên quan đến đề tài.
* Thiết kế cơ sở dữ liệu
* Phân tích và thiết kế hệ thống đã đăng kí
* Thiết kế các giao diện
* Viết chương trình
* Viết cuốn báo cáo tiểu luận môn học theo biểu mẫu
* In cuốn báo cáo và nộp cuốn báo cáo theo lịch thi cuối kỳ trên hệ thống

*Nội dung và yêu cầu đã được thông qua Bộ môn*.

*Tp.HCM, ngày tháng năm 2025*

|  |  |
| --- | --- |
| **TRƯỞNG BỘ MÔN**  *(Ký và ghi rõ họ tên)* | **GIÁO VIÊN HƯỚNG DẪN**  *(Ký và ghi rõ họ tên)*  **ThS. Đỗ Hoàng Nam** |

LỜI CẢM ƠN

Tri thức là ngọn đèn soi sáng con đường phát triển của mỗi con người. Trong hành trình học tập và nghiên cứu tại Trường Đại học Nguyễn Tất Thành, việc thực hiện và hoàn thành một đồ án không chỉ là trách nhiệm mà còn là cơ hội quý báu để chúng em áp dụng những kiến thức đã được học vào thực tế, phát huy tư duy sáng tạo và rèn luyện kỹ năng bản thân.

Để có cơ hội học tập và hoàn thành tiểu luận này trước hết, em xin gửi lời cảm ơn chân thành đến gia đình, những người luôn ở bên, đồng hành và là chỗ dựa tinh thần to lớn giúp em vượt qua mọi thử thách trong cuộc sống và học tập.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến quý Thầy/Cô Khoa Công nghệ Thông tin và Ban Giám hiệu Trường Đại học Nguyễn Tất Thành, đặc biệt là Thầy **Đỗ Hoàng Nam**. Với sự tận tâm, kiến thức sâu rộng và những kinh nghiệm quý giá, Thầy đã luôn định hướng, động viên và truyền cảm hứng cho em trong suốt quá trình thực hiện đồ án. Những gợi ý và lời chỉ bảo của Thầy không chỉ giúp em hoàn thiện đồ án này mà còn là hành trang quý báu cho em trong chặng đường sắp tới.

Bên cạnh đó, mình xin gửi lời cảm ơn chân thành đến bạn Đặng Hữu Khoa, người bạn đã luôn đồng hành, sát cánh, hỗ trợ và sẻ chia mọi khó khăn trong suốt quá trình thực hiện tiểu luận. Những đóng góp và sự cổ vũ từ bạn là động lực to lớn để mình vượt qua mọi thử thách, hướng tới mục tiêu hoàn thành tiểu luận một cách chuẩn chỉnh nhất.

Em hiểu rằng, với thời gian và năng lực còn hạn chế, tiểu luận này chắc chắn sẽ còn nhiều thiếu sót. Em rất mong nhận được sự góp ý, đánh giá từ quý Thầy/Cô để em có thể học hỏi, cải thiện và hoàn thiện bản thân trong tương lai. Hy vọng rằng, tiểu luận này sẽ đóng góp một phần nhỏ vào sự phát triển của xã hội, đồng thời góp phần vào công cuộc công nghiệp hóa, hiện đại hóa đất nước.

Cuối cùng, em xin kính chúc quý Thầy/Cô luôn dồi dào sức khỏe, thành công trong sự nghiệp và mãi giữ ngọn lửa đam mê với nghề giáo.

**Em xin chân thành cảm ơn!**

Tp.HCM, ngày 17 tháng 01 năm 2025

Huỳnh Nhật Long

LỜI MỞ ĐẦU

Trong xã hội hiện đại, khi nhu cầu đi lại, di chuyển của con người ngày càng tăng cao cả về tần suất lẫn chất lượng dịch vụ, việc quản lý và vận hành hiệu quả các hoạt động kinh doanh vận tải hành khách trở thành yếu tố quyết định sự thành công của doanh nghiệp. Đặc biệt, sự phát triển mạnh mẽ của công nghệ thông tin đã mở ra cơ hội lớn để áp dụng các giải pháp kỹ thuật số, giúp tối ưu hóa quy trình hoạt động, giảm thiểu chi phí và nâng cao chất lượng phục vụ khách hàng.

Ngành vận tải hành khách đường bộ, nhất là các doanh nghiệp kinh doanh dịch vụ vận tải, vẫn đang đối mặt với nhiều khó khăn trong việc quản lý đặt vé, sắp xếp chuyến xe, quản lý sơ đồ ghế và cải thiện trải nghiệm của hành khách. Nhiều doanh nghiệp vẫn dựa vào các phương pháp thủ công hoặc các hệ thống chưa tối ưu, dẫn đến việc vận hành kém hiệu quả, gia tăng chi phí và giảm tính cạnh tranh.

Xuất phát từ các yếu tố này, đề tài **"Thiết kế và xây dựng website đặt vé xe trực tuyến"** được triển khai nhằm tạo ra một hệ thống hiện đại, giúp giải quyết các vấn đề quản lý đặt vé và vận hành trong lĩnh vực này. Hệ thống được phát triển với công nghệ PHP và Laravel kết hợp cơ sở dữ liệu MySQL, đảm bảo tính ổn định, bảo mật, và khả năng mở rộng. Hệ thống cung cấp các tính năng toàn diện như quản lý chuyến xe, quản lý sơ đồ ghế, đặt vé trực tuyến, theo dõi doanh thu và tối ưu hóa quy trình vận hành.

Thông qua việc thiết kế và xây dựng hệ thống này, chúng em hy vọng mang đến một giải pháp công nghệ hiệu quả, góp phần giúp các doanh nghiệp vận tải tiết kiệm thời gian, giảm thiểu chi phí và nâng cao trải nghiệm khách hàng. Đây không chỉ là bước tiến trong việc áp dụng công nghệ vào ngành vận tải mà còn là nỗ lực đóng góp vào quá trình chuyển đổi số và nâng cao năng lực cạnh tranh của doanh nghiệp Việt Nam trên thị trường.

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH  **TRUNG TÂM KHẢO THÍ** | **KỲ THI KẾT THÚC HỌC PHẦN**  **HỌC KỲ I NĂM HỌC 2024 – 2025** |

PHIẾU CHẤM THI TIỂU LUẬN MÔN HỌC

BM-ChT-11

Môn thi: Chuyên đề chuyên sâu PTPM 1 Mã lớp học phần: 012307770001

Sinh viên thực hiện: Huỳnh Nhật Long

Ngày thi: 17/01/2025 Phòng thi: L.511

Đề tài tiểu luận: Thiết kế và xây dựng website đặt vé xe trực tuyến

Phần đánh giá của giảng viên (căn cứ trên thang rubrics của môn học):

|  |  |  |  |
| --- | --- | --- | --- |
| **Tiêu chí (theo CĐR HP)** | **Đánh giá của GV** | **Điểm tối đa** | **Điểm đạt được** |
| Cấu trúc của báo cáo |  | 2 |  |
| **Nội dung** |  |  |  |
| * Các nội dung thành phần |  | 2 |  |
| * Lập luận |  | 2 |  |
| * Kết luận |  | 1 |  |
| - Trình bày |  | 3 |  |
| **TỔNG ĐIỂM** |  | 10 |  |

|  |  |
| --- | --- |
|  | **Giảng viên chấm thi**  *(ký, ghi rõ họ tên)* |

**ThS. Đỗ Hoàng Nam**

MỤC LỤC

[NHIỆM VỤ TIỂU LUẬN MÔN HỌC i](#_Toc187670636)

[LỜI CẢM ƠN iii](#_Toc187670637)

[LỜI MỞ ĐẦU iv](#_Toc187670638)

[PHIẾU CHẤM THI TIỂU LUẬN MÔN HỌC v](#_Toc187670639)

[MỤC LỤC vi](#_Toc187670640)

[DANH MỤC CÁC HÌNH ix](#_Toc187670641)

[DANH MỤC CÁC BẢNG BIỂU xi](#_Toc187670642)

[DANH MỤC CHỮ VIẾT TẮT xii](#_Toc187670643)

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 1](#_Toc187670644)

[1.1. Khảo sát hiện trạng 1](#_Toc187670645)

[1.1.1. Giới thiệu về PHP và Laravel 1](#_Toc187670646)

[1.1.2. Tầm quan trọng của MySQL trong lưu trữ và quản lý dữ liệu 1](#_Toc187670647)

[1.2. Lý do chọn đề tài 1](#_Toc187670648)

[1.2.1. Tại sao lựa chọn PHP, Laravel và MySQL cho dự án này 1](#_Toc187670649)

[1.2.2. Những lợi ích và thách thức khi thực hiện đề tài 2](#_Toc187670650)

[1.3. Mục tiêu của đề tài 2](#_Toc187670651)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4](#_Toc187670652)

[2.1. Giới thiệu về PHP 4](#_Toc187670653)

[2.1.1. Tổng quan về PHP 4](#_Toc187670654)

[2.1.2. Các khái niệm cơ bản về PHP 4](#_Toc187670655)

[2.2. Laravel Framework 4](#_Toc187670656)

[2.2.1. Tổng quan về Laravel 4](#_Toc187670657)

[2.2.2. Cấu trúc của một dự án Laravel 5](#_Toc187670658)

[2.2.3. Cơ chế Route, Controller, View và Middleware 6](#_Toc187670659)

[2.3. MySQL 6](#_Toc187670660)

[2.3.1. Tổng quan về MySQL 6](#_Toc187670661)

[2.3.2. Các khái niệm cơ bản 6](#_Toc187670662)

[2.3.3. Cách kết nối Laravel với MySQL 7](#_Toc187670663)

[CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 8](#_Toc187670664)

[3.1. Phân tích yêu cầu 8](#_Toc187670665)

[3.1.1. Yêu cầu chức năng 8](#_Toc187670666)

[3.1.2. Yêu cầu phi chức năng 9](#_Toc187670667)

[3.2. Thiết kế hệ thống 10](#_Toc187670668)

[3.2.1. Mô hình MVC trong Laravel 10](#_Toc187670669)

[3.2.2. Mô tả tổng quan hệ thống 10](#_Toc187670670)

[3.2.3. Phân tích sơ đồ Use-case 14](#_Toc187670671)

[3.2.3.1. Sơ đồ Use-case 14](#_Toc187670672)

[3.2.3.2. Đặc tả các sơ đồ Use-case 16](#_Toc187670673)

[3.2.4. Thiết kế cơ sở dữ liệu 20](#_Toc187670674)

[3.2.4.1. Xác định các class 20](#_Toc187670675)

[3.2.4.2. Bổ sung thuộc tính cho các class 21](#_Toc187670676)

[3.2.4.3. Giải thích mỗi quan hệ giữa các class 23](#_Toc187670677)

[3.2.4.4. Chuyển đổi class sang lược đồ cơ sở dữ liệu quan hệ 24](#_Toc187670678)

[3.2.4.5. Mô tả chi tiết các bảng dữ liệu 24](#_Toc187670679)

[3.3. Thiết kế giao diện người dùng 26](#_Toc187670680)

[3.3.1. Xây dựng Wireframe và Mockup các trang chính 26](#_Toc187670681)

[3.3.2. Thiết kế UI/UX với Blade Template Engine của Laravel 27](#_Toc187670682)

[CHƯƠNG 4: THỰC NGHIỆM VÀ TRIỂN KHAI HỆ THỐNG 29](#_Toc187670683)

[4.1. Mô tả cấu trúc thư mục code 29](#_Toc187670684)

[4.1.1. Cấu trúc thư mục chính 29](#_Toc187670685)

[4.1.2. Cấu trúc các tệp tin chính 30](#_Toc187670686)

[4.2. Phát triển Backend (Laravel, PHP) 30](#_Toc187670687)

[4.2.1. Xây dựng các route, controller, và model trong Laravel 30](#_Toc187670688)

[4.2.1.1. Route 30](#_Toc187670689)

[4.2.1.2. Controller 31](#_Toc187670690)

[4.2.1.3. Model 31](#_Toc187670691)

[4.2.2. Quản lý cơ sở dữ liệu với Eloquent ORM 32](#_Toc187670692)

[4.2.3. Triển khai xác thực người dùng và phân quyền 33](#_Toc187670693)

[4.3. Phát triển Frontend 34](#_Toc187670694)

[4.3.1. Tích hợp giao diện người dùng với Blade Template 34](#_Toc187670695)

[4.3.2. Sử dụng Bootstrap responsive design 34](#_Toc187670696)

[4.3.3. Giao diện Frontend 35](#_Toc187670697)

[4.3.3.1. Giao diện đăng nhập 35](#_Toc187670698)

[4.3.3.2. Giao diện chức năng đặt vé 36](#_Toc187670699)

[KẾT LUẬN 39](#_Toc187670700)

[TÀI LIỆU THAM KHẢO 40](#_Toc187670701)

DANH MỤC CÁC HÌNH

[Hình 2.1: Logo biểu trưng của PHP 4](#_Toc187670603)

[Hình 2.2: Logo nhận diện của Laravel 4](#_Toc187670604)

[Hình 2.3: Minh họa cách Eloquent làm việc với DB 5](#_Toc187670605)

[Hình 2.4: Minh họa cấu trúc thư mục của một project Laravel 5](#_Toc187670606)

[Hình 2.5: Minh họa cách middleware xử lý các request 6](#_Toc187670607)

[Hình 2.6: Logo nhận diện của MySQL 6](#_Toc187670608)

[Hình 2.7: Minh họa cách kết nối MySQL 7](#_Toc187670609)

[Hình 2.8: Minh họa câu lệnh tạo bảng CSDL 7](#_Toc187670610)

[Hình 3.1: Sơ đồ phân cấp chức năng của nhóm chức năng chính phía người dùng 13](#_Toc187670611)

[Hình 3.2: Sơ đồ phân cấp chức năng nhóm chức năng chính trang quản trị 13](#_Toc187670612)

[Hình 3.3: Sơ đồ Use-case tổng quát của hệ thống đặt vé xe trực tuyến 15](#_Toc187670613)

[Hình 3.4: Sơ đồ Use-case chi tiết của chức năng đăng nhập 15](#_Toc187670614)

[Hình 3.5: Sơ đồ Use-case chi tiết của chức năng đăng ký 16](#_Toc187670615)

[Hình 3.6: Sơ đồ Use-case chi tiết chức năng quản lý chuyến xe 16](#_Toc187670616)

[Hình 3.7: Sơ đồ class 22](#_Toc187670617)

[Hình 3.8: Sơ đồ mô hình cơ sở dữ liệu quan hệ 25](#_Toc187670618)

[Hình 3.9: Wireframe phác họa tổng thể trang chủ của hệ thống 26](#_Toc187670619)

[Hình 3.10: Layout tổng thể sau khi thiết kế của trang chủ 28](#_Toc187670620)

[Hình 4.1: Cấu trúc thư mục của dự án 29](#_Toc187670621)

[Hình 4.2: Cấu trúc các tệp tin quan trọng của dự án 30](#_Toc187670622)

[Hình 4.3: Ví dụ minh họa cho Routes(admin) trong web.php 31](#_Toc187670623)

[Hình 4.4: Minh họa Controller (HomeController) 31](#_Toc187670624)

[Hình 4.5: Minh họa Model (Role) 31](#_Toc187670625)

[Hình 4.6: Minh họa cho Migration 32](#_Toc187670626)

[Hình 4.7: Minh họa về Seeder 33](#_Toc187670627)

[Hình 4.8: Minh họa triển khai xác thực phân quyền 33](#_Toc187670628)

[Hình 4.9: Minh họa sử dụng phân quyền trong routes 33](#_Toc187670629)

[Hình 4.10: Minh họa layout trong blade template 34](#_Toc187670630)

[Hình 4.11: Minh họa cho việc kế thừa layout từ admin 34](#_Toc187670631)

[Hình 4.12: Cài đặt Boostrap 35](#_Toc187670632)

[Hình 4.13: Thành phần giao diện form 35](#_Toc187670633)

[Hình 4.14: Giao diện trang đăng nhập của hệ thống 36](#_Toc187670634)

[Hình 4.15: Giao diện chức năng đặt vé xe 37](#_Toc187670635)

DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1: Bảng mô tả Use-case của chức năng đăng nhập 17](#_Toc187670517)

[Bảng 3‑2: Bảng mô tả Use-case của chức năng đăng ký 18](#_Toc187670518)

[Bảng 3‑3: Bảng mô tả Use-case của chức năng quản lý chuyến xe 19](#_Toc187670519)

[Bảng 3‑4: Bảng mô tả tổng quát các bảng dữ liệu 24](#_Toc187670520)

[Bảng 3‑5: Bảng mô tả chi tiết bảng dữ liệu Users 25](#_Toc187670521)

[Bảng 3‑6: Bảng mô tả chi tiết bảng dữ liệu Role 25](#_Toc187670522)

[Bảng 4‑1: Bảng mô tả giao diện trang đăng nhập 36](#_Toc187670523)

[Bảng 4‑2: Bảng mô tả chức năng đặt vé 37](#_Toc187670524)

DANH MỤC CHỮ VIẾT TẮT

|  |  |
| --- | --- |
| **Các từ viết tắt** | **Giải thích** |
| MVC | Model – View – Controller |
| PHP | PHP: Hypertext Preprocessor |
| ORM | Object-Relational Mapping |
| CRUD | Create – Read – Update – Delete |
| CLI | Command Line Interface |
| UI/UX | User Interface/User Experience |
| URL | Uniform Resource Locator |
| RDBMS | Relational Database Management System |

## TỔNG QUAN ĐỀ TÀI

### Khảo sát hiện trạng

Trong thời đại công nghệ phát triển vượt bậc như hiện nay, các ứng dụng và hệ thống thông minh đóng vai trò vô cùng quan trọng trong việc hỗ trợ quản lý và vận hành doanh nghiệp một cách hiệu quả. Trong đó, lĩnh vực vận tải hành khách, đặc biệt là việc đặt vé xe trực tuyến, đang ngày càng trở nên phổ biến nhờ tính tiện lợi và hiệu quả mà nó mang lại.

#### Giới thiệu về PHP và Laravel

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình phổ biến, được sử dụng rộng rãi để phát triển các ứng dụng web động nhờ khả năng tương tác mạnh mẽ với cơ sở dữ liệu. Laravel, một framework hiện đại dựa trên PHP, ra đời nhằm đơn giản hóa và tối ưu hóa quy trình phát triển web. Với cấu trúc dựa trên mô hình MVC (Model-View-Controller), Laravel giúp tách biệt rõ ràng các phần xử lý logic, giao diện và dữ liệu, từ đó tăng tính dễ bảo trì và mở rộng cho hệ thống. Các tính năng nổi bật của Laravel bao gồm ORM Eloquent hỗ trợ tương tác cơ sở dữ liệu, hệ thống định tuyến thông minh, và các công cụ bảo mật tích hợp sẵn, giúp lập trình viên xây dựng hệ thống hiệu quả và an toàn [1].

#### Tầm quan trọng của MySQL trong lưu trữ và quản lý dữ liệu

MySQL, một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở, đóng vai trò quan trọng trong việc lưu trữ và quản lý dữ liệu. Với khả năng xử lý nhanh, ổn định và dễ mở rộng, MySQL là lựa chọn lý tưởng cho các hệ thống lớn như đặt vé xe trực tuyến. Khi tích hợp với Laravel, ORM Eloquent của framework này tối ưu hóa các truy vấn và thao tác cơ sở dữ liệu, cải thiện đáng kể hiệu suất và trải nghiệm người dùng [2].

### Lý do chọn đề tài

#### Tại sao lựa chọn PHP, Laravel và MySQL cho dự án này

PHP là một ngôn ngữ phổ biến, dễ học và có cộng đồng lớn mạnh, cung cấp nhiều tài nguyên hỗ trợ lập trình viên. Laravel, với các tính năng nổi bật như mô hình MVC, hệ thống route linh hoạt, và công cụ bảo mật mạnh mẽ, giúp tăng tốc độ phát triển và giảm chi phí duy trì hệ thống. Bên cạnh đó, MySQL đáp ứng tốt nhu cầu lưu trữ và quản lý dữ liệu lớn với hiệu suất cao và tính ổn định vượt trội.

Sự kết hợp giữa PHP, Laravel và MySQL mang lại nhiều lợi ích:

* **Đồng bộ công nghệ**: PHP và Laravel hỗ trợ phát triển toàn diện từ giao diện người dùng đến backend.
* **Hiệu suất cao**: Laravel cung cấp giao diện tương tác tốt, trong khi MySQL đảm bảo xử lý dữ liệu nhanh chóng và hiệu quả.
* **Tính linh hoạt**: Hệ thống có thể dễ dàng mở rộng hoặc điều chỉnh để đáp ứng các nhu cầu kinh doanh thay đổi.

#### Những lợi ích và thách thức khi thực hiện đề tài

**Lợi ích:**

* Cung cấp giải pháp quản lý tích hợp toàn diện, giúp tiết kiệm thời gian, nhân lực và chi phí.
* Tạo giao diện trực quan, dễ sử dụng cho mọi đối tượng người dùng, từ quản lý đến người dùng.
* Đảm bảo xử lý nhanh chóng các thao tác đặt vé, quản lý lịch trình, và theo dõi doanh thu.

**Thách thức:**

* Yêu cầu kiến thức chuyên sâu về cả Laravel và MySQL để phát triển hệ thống hiệu quả.
* Cần áp dụng các biện pháp bảo mật mạnh mẽ để bảo vệ dữ liệu người dùng.

### Mục tiêu của đề tài

Đề tài mà chúng em thực hiện trong tiểu luận này hướng đến xây dựng một hệ thống website đặt vé xe trực tuyến giúp quản lý toàn diện các hoạt động vận tải hành khách, từ đặt vé, quản lý lịch trình xe đến thống kê doanh thu. Cụ thể:

1. **Phát triển hệ thống quản lý dữ liệu:**

* Tích hợp tính năng quản lý vé bao gồm đặt vé, hủy vé, và theo dõi trạng thái vé.
* Quản lý lịch trình xe với các tuyến đường, điểm đón và điểm trả cụ thể.

1. **Thiết kế giao diện người dùng hiện đại:**

* Sử dụng Laravel Blade Template để xây dựng giao diện trực quan, dễ thao tác.
* Tối ưu hóa trải nghiệm người dùng trên cả thiết bị di động và máy tính.

1. **Xử lý dữ liệu hiệu quả:**

* Áp dụng ORM Eloquent kết hợp với MySQL để tối ưu hóa truy vấn và hiệu suất.
* Sử dụng caching và tối ưu hóa cấu trúc cơ sở dữ liệu để tăng tốc độ phản hồi.

1. **Đảm bảo an toàn và tính mở rộng:**

* Sử dụng các biện pháp bảo mật như mã hóa dữ liệu, xác thực người dùng và phân quyền truy cập.
* Thiết kế hệ thống với kiến trúc mở, sẵn sàng tích hợp thêm các tính năng trong tương lai.

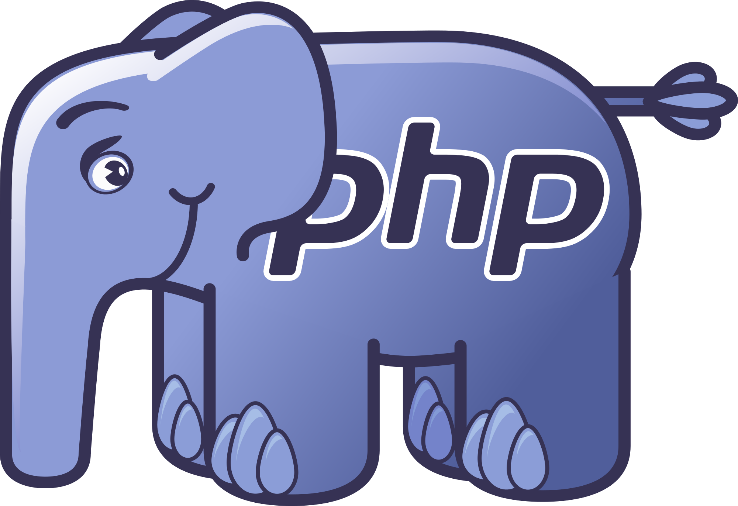
Hệ thống sau khi hoàn thành không chỉ giải quyết được các vấn đề quản lý cơ bản mà còn có khả năng mở rộng để áp dụng cho các doanh nghiệp vận tải lớn hơn. Hệ thống sẽ hỗ trợ tối ưu hóa quy trình làm việc, nâng cao năng suất lao động và mang lại lợi thế cạnh tranh trên thị trường.

## CƠ SỞ LÝ THUYẾT

### Giới thiệu về PHP

#### Tổng quan về PHP

PHP (Hypertext Preprocessor) là một ngôn ngữ lập trình phía server được thiết kế để phát triển các ứng dụng web động. Ngôn ngữ này được Rasmus Lerdorf giới thiệu lần đầu vào năm 1994 và đã nhanh chóng trở thành một trong những ngôn ngữ phổ biến nhất trong lĩnh vực phát triển web [3]



Hình 2.1: Logo biểu trưng của PHP

#### Các khái niệm cơ bản về PHP

PHP cung cấp các khái niệm cơ bản như biến, hàm, điều kiện và vòng lặp để xây dựng các ứng dụng web động. Biến trong PHP được khai báo với dấu $, giúp lưu trữ thông tin cần thiết. Hàm được dùng để tái sử dụng mã nguồn bằng cách tập hợp các đoạn mã thực hiện một chức năng nhất định. Điều kiện được sử dụng để kiểm soát luồng xử lý với các câu lệnh như if, else, elseif, switch, cho phép chương trình hoạt động linh hoạt hơn. Các vòng lặp như for, while, do-while, foreach hỗ trợ việc thực hiện các tác vụ lặp đi lặp lại trong một số lần hoặc khi một điều kiện nào đó còn đúng.

### Laravel Framework

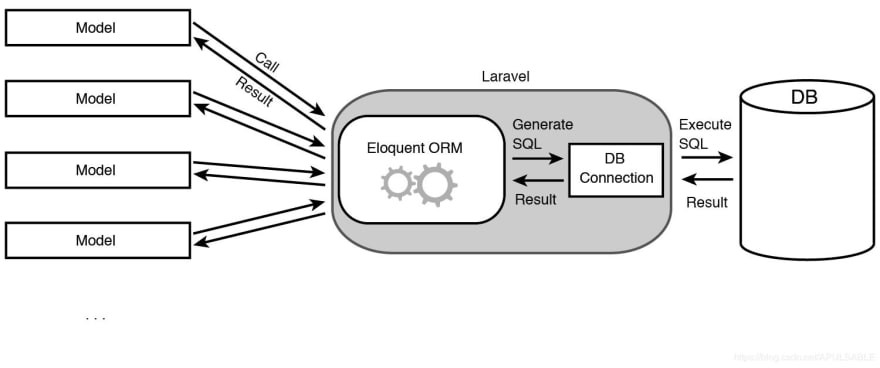
#### Tổng quan về Laravel



Hình 2.2: Logo nhận diện của Laravel

Laravel là một framework PHP mã nguồn mở được phát triển bởi Taylor Otwell, lần đầu ra mắt vào năm 2011. Laravel được đánh giá cao nhờ vào việc mang lại các công cụ hữu ích giúp tối ưu hoá vòng đời phát triển [4].

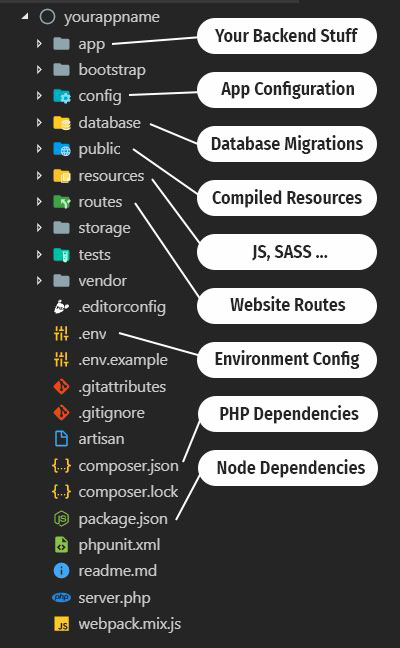
Artisan, Blade và Eloquent ORM: Artisan là công cụ dòng lệnh CLI hỗ trợ việc tạo migration, controller và model một cách nhanh chóng. Blade là công cụ templating engine giúp tối ưu hóa giao diện với cú pháp đơn giản, hỗ trợ kế thừa và chia nhỏ các thành phần giao diện. Eloquent ORM là hệ thống ORM cho phép quản lý cơ sở dữ liệu dưới dạng các model, giúp tương tác với dữ liệu dễ dàng mà không cần viết truy vấn SQL thủ công.



Hình 2.3: Minh họa cách Eloquent làm việc với DB

#### Cấu trúc của một dự án Laravel

Thư mục app/, routes/ và resources/: Thư mục app/ chứa các model và controller, thư mục routes/ chứa các file route như web.php, api.php, và thư mục resources/ chứa các file giao diện Blade được sử dụng để xây dựng các view.



Hình 2.4: Minh họa cấu trúc thư mục của một project Laravel

#### Cơ chế Route, Controller, View và Middleware

Route, Controller, View và Middleware: Cơ chế định tuyến (Route) định nghĩa các URL và gửi request đến các controller tương ứng để xử lý logic. Các controller chứa mã xử lý chính, trong khi các view là nơi chứa giao diện người dùng và hiển thị kết quả trả về. Middleware là lớp trung gian xử lý request trước khi chuyển đến controller, đảm bảo các điều kiện như bảo mật và kiểm tra phiên làm việc được thực hiện.

**A diagram of a diagram

Description automatically generated**

Hình 2.5: Minh họa cách middleware xử lý các request

### MySQL

#### Tổng quan về MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) phổ biến được Oracle Corporation bảo trì và phát triển. MySQL đóng vai trò quan trọng trong việc quản lý dữ liệu và là backend cho nhiều ứng dụng web [5].



Hình 2.6: Logo nhận diện của MySQL

#### Các khái niệm cơ bản

Bảng, Cột, Truy vấn, JOIN và Index: Bảng là cơ sở chứa dữ liệu dưới dạng hàng và cột, trong đó cột đại diện cho các thuộc tính của bảng. Các truy vấn SQL như SELECT, INSERT, UPDATE, DELETE được sử dụng để tương tác với dữ liệu. JOIN cho phép kết hợp dữ liệu từ nhiều bảng để tạo kết quả tổng hợp, trong khi Index giúp tối ưu hóa việc tìm kiếm dữ liệu trong cơ sở dữ liệu.

#### Cách kết nối Laravel với MySQL

Để kết nối Laravel với MySQL, trước tiên, cần cập nhật tệp .env của dự án với các thông số kết nối cơ sở dữ liệu, bao gồm loại kết nối (DB\_CONNECTION), địa chỉ máy chủ (DB\_HOST), cổng (DB\_PORT), tên cơ sở dữ liệu (DB\_DATABASE), tên người dùng (DB\_USERNAME) và mật khẩu (DB\_PASSWORD)

**A computer screen with white text

Description automatically generated**

Hình 2.7: Minh họa cách kết nối MySQL

Sau khi cấu hình xong, bạn có thể sử dụng lệnh php artisan migrate để tạo các bảng trong cơ sở dữ liệu dựa trên các tệp migration đã được định nghĩa trong dự án Laravel.



Hình 2.8: Minh họa câu lệnh tạo bảng CSDL

## PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

### Phân tích yêu cầu

#### Yêu cầu chức năng

Hệ thống đặt vé xe trực tuyến cần tích hợp các chức năng chính nhằm đảm bảo đáp ứng hiệu quả nhu cầu quản lý của doanh nghiệp vận tải và mang lại trải nghiệm tốt nhất cho người dùng. Các chức năng cụ thể bao gồm:

**Quản lý xe:**

* Hệ thống cần hỗ trợ thêm mới, sửa đổi và xóa thông tin về các phương tiện trong doanh nghiệp, bao gồm biển số xe, loại xe, số lượng ghế và sơ đồ ghế.
* Mỗi xe được gắn với một hoặc nhiều tuyến đường cụ thể để thuận tiện trong việc quản lý.
* Quản lý trạng thái hoạt động của xe như: đang hoạt động, bảo trì, hoặc ngừng hoạt động.

**Quản lý tuyến đường:**

* Hỗ trợ định nghĩa các tuyến đường chính bao gồm điểm đầu và điểm cuối.
* Cho phép điều chỉnh khoảng cách và thời gian dự kiến giữa các điểm để tối ưu hóa lịch trình.
* Quản lý thông tin chi tiết của từng tuyến, đảm bảo tính chính xác cho việc lập kế hoạch và cung cấp thông tin cho hành khách.

**Quản lý chuyến đi:**

* Tạo và quản lý các chuyến đi cụ thể dựa trên các tuyến đường đã định nghĩa, bao gồm thời gian khởi hành và phương tiện được sử dụng.
* Hỗ trợ cập nhật trạng thái chuyến đi như: chưa khởi hành, đang khởi hành, hoặc đã hoàn thành.
* Cho phép điều chỉnh lịch trình chuyến đi khi cần thiết để đáp ứng các yếu tố khách quan như giao thông hoặc yêu cầu từ hành khách.

**Quản lý vé:**

* Cung cấp công cụ để hành khách tìm kiếm chuyến đi theo điểm đi, điểm đến và ngày giờ khởi hành.
* Hỗ trợ đặt vé, giữ vé tạm thời và hủy vé theo yêu cầu của khách hàng.
* Quản lý chi tiết từng vé, bao gồm số ghế, thông tin hành khách và trạng thái vé (đã thanh toán, chưa thanh toán, hoặc hủy bỏ).

**Báo cáo và thống kê:**

* Cung cấp các báo cáo chi tiết về doanh thu, tỷ lệ lấp đầy ghế, hiệu suất hoạt động của xe, và các tuyến đường phổ biến.
* Xuất báo cáo định kỳ theo ngày, tuần, hoặc tháng, đồng thời cung cấp khả năng tuỳ chỉnh báo cáo theo nhu cầu.

Các chức năng trên được thiết kế với mục tiêu tối ưu hóa quy trình vận hành, giảm thiểu công việc thủ công và nâng cao hiệu quả kinh doanh, đồng thời mang đến cho khách hàng trải nghiệm dịch vụ chuyên nghiệp, tiện lợi và an toàn.

#### Yêu cầu phi chức năng

Hệ thống đặt vé xe trực tuyến không chỉ cần đáp ứng các yêu cầu chức năng cơ bản mà còn phải đạt các tiêu chuẩn phi chức năng để mang lại trải nghiệm tốt nhất cho người dùng:

* **Hiệu năng:** Hệ thống phải đảm bảo tốc độ phản hồi nhanh, đặc biệt khi người dùng tìm kiếm chuyến xe hoặc thực hiện thanh toán. Ngoài ra, hệ thống cần có khả năng chịu tải tốt để phục vụ lượng lớn người dùng cùng lúc.
* **Bảo mật:** Tính bảo mật phải được ưu tiên hàng đầu. Hệ thống cần sử dụng các giao thức an toàn như HTTPS để mã hóa dữ liệu. Việc xác thực người dùng, phân quyền truy cập và bảo vệ khỏi các lỗ hổng bảo mật như SQL Injection hay Cross-Site Scripting (XSS) là bắt buộc.
* **Khả năng sử dụng:** Giao diện cần được thiết kế đơn giản, trực quan và dễ sử dụng ngay cả với người dùng không quen thuộc với công nghệ. Hệ thống cũng cần hỗ trợ tốt trên các thiết bị di động và đa ngôn ngữ để mở rộng đối tượng người dùng.
* **Tính linh hoạt và mở rộng:** Hệ thống cần có khả năng dễ dàng mở rộng hoặc tích hợp thêm các tính năng mới trong tương lai, đồng thời dễ dàng bảo trì và nâng cấp.

### Thiết kế hệ thống

#### Mô hình MVC trong Laravel

Laravel là một framework PHP phổ biến và mạnh mẽ, áp dụng mô hình MVC (Model-View-Controller) để tổ chức và quản lý mã nguồn một cách khoa học, giúp ứng dụng dễ dàng bảo trì và mở rộng. Mô hình này bao gồm ba thành phần chính: Model, View và Controller.

Model (M) đảm nhận nhiệm vụ quản lý dữ liệu và các logic nghiệp vụ liên quan đến dữ liệu. Đây là nơi tương tác trực tiếp với cơ sở dữ liệu, thực hiện các thao tác như thêm, sửa, xóa và truy vấn dữ liệu. Model cũng định nghĩa các mối quan hệ giữa các bảng trong cơ sở dữ liệu để đảm bảo dữ liệu được tổ chức chặt chẽ. Các model được đặt trong thư mục app/Models của ứng dụng. Ví dụ, model Trip đại diện cho bảng trips trong cơ sở dữ liệu và chứa các thuộc tính như route\_id, bus\_id, departure\_time, duration, status, đồng thời định nghĩa quan hệ giữa các chuyến đi với bảng buses và routes.

View (V) là thành phần chịu trách nhiệm hiển thị dữ liệu cho người dùng. Laravel sử dụng công cụ Blade Template Engine để xây dựng giao diện, cho phép sử dụng cú pháp đơn giản để hiển thị dữ liệu động và tái sử dụng các thành phần giao diện như header và footer. Các file view được lưu trữ trong thư mục resources/views. Ví dụ, file booking\_history.blade.php hiển thị lịch sử đặt vé của người dùng bằng cách hiển thị thông tin từ Model theo định dạng bảng.

Controller (C) là thành phần xử lý các yêu cầu từ người dùng, thực hiện các tác vụ logic và trả kết quả về cho View để hiển thị. Controller đóng vai trò trung gian, kết nối giữa Model và View, đảm bảo luồng xử lý logic của ứng dụng diễn ra một cách mạch lạc. Các controller được đặt trong thư mục app/Http/Controllers. Ví dụ, controller BookingHistoryController nhận yêu cầu hiển thị lịch sử đặt vé từ người dùng, lấy dữ liệu từ bảng bookings thông qua Model Booking, và trả về view booking\_history với dữ liệu lịch sử đặt vé tương ứng.

#### Mô tả tổng quan hệ thống

Hệ thống đặt vé xe trực tuyến là một website cung cấp các chức năng hỗ trợ cho việc tìm kiếm chuyến xe, đặt vé và quản lý các chuyến xe. Hệ thống cung cấp dịch vụ cho khách hàng tìm kiếm và đặt vé các chuyến xe từ nhiều nhà xe khác nhau. Khách hàng có thể đặt vé qua website từ bất kỳ đâu và vào bất kỳ thời gian nào, miễn là có kết nối internet.

Các chuyến xe được phân loại theo tuyến đường, giờ khởi hành, loại xe và các dịch vụ khác. Trong mỗi tuyến đường, khách hàng có thể xem các chuyến xe với thông tin chi tiết về giá vé, điểm đi, điểm đến, giờ xuất bến, loại xe và các dịch vụ kèm theo.

Ngoài chức năng đặt vé, hệ thống còn cung cấp các tính năng quản lý vé, thống kê doanh thu và báo cáo cho quản trị viên, giúp quản lý công việc và vận hành hệ thống một cách hiệu quả.

Chức năng người dùng:

1. **Tìm kiếm chuyến xe**: Khách hàng có thể tìm kiếm chuyến xe dựa trên các tiêu chí như nơi đi, nơi đến, ngày giờ và loại xe.
2. **Xem chi tiết chuyến xe**: Sau khi tìm kiếm, khách hàng có thể xem thông tin chi tiết về chuyến xe như giá vé, thời gian xuất bến, thời gian dự kiến đến, và các điểm dừng trên tuyến đường.
3. **Chọn chuyến**: Sau khi xem chi tiết, khách hàng có thể chọn chuyến xe phù hợp với nhu cầu của mình.
4. **Đặt vé**: Khách hàng có thể chọn vé, điền thông tin cá nhân và tiến hành đặt vé.
5. **Thanh toán**: Sau khi xác nhận thông tin chuyến đi và vé, khách hàng sẽ thanh toán cho chuyến xe đã chọn.
6. **Đăng ký tài khoản**: Khách hàng mới sẽ cần đăng ký tài khoản với thông tin cơ bản như tên, email, số điện thoại và mật khẩu để tạo tài khoản trên hệ thống.
7. **Đăng nhập**: Khách hàng đã có tài khoản sẽ đăng nhập vào hệ thống để thực hiện các thao tác tìm kiếm và đặt vé.
8. **Nhận vé**: Sau khi thanh toán, khách hàng nhận vé điện tử qua email hoặc qua hệ thống, có thể lưu trữ trên điện thoại hoặc máy tính để sử dụng khi đi xe.
9. **Quản lý thông tin cá nhân**: Khách hàng có thể quản lý thông tin cá nhân như thông tin liên lạc, địa chỉ nhận vé, và phương thức thanh toán đã lưu.
10. **Quản lý lịch sử đặt vé**: Khách hàng có thể xem lại lịch sử các chuyến xe đã đặt và theo dõi trạng thái vé.

Chức năng phía trang quản trị:

1. **Quản lý xe**: Quản trị viên có thể thêm, sửa, xóa thông tin các loại xe mà hệ thống đang quản lý. Mỗi xe sẽ có thông tin như tên xe, số lượng ghế, loại xe (giường nằm, ghế ngồi) và các đặc điểm khác.
2. **Quản lý tuyến đường**: Quản trị viên có thể thêm, sửa, xóa các tuyến đường mà hệ thống hỗ trợ, bao gồm các điểm đón, điểm trả khách, giờ xuất bến, và các thông tin liên quan.
3. **Quản lý chuyến đi**: Quản trị viên có thể tạo, chỉnh sửa hoặc xóa các chuyến xe, bao gồm thông tin về chuyến đi, số lượng vé còn lại, giờ khởi hành và các yêu cầu đặc biệt khác.
4. **Quản lý vé**: Quản trị viên có thể theo dõi tình trạng vé, xác nhận các vé đã đặt, thay đổi trạng thái vé (đã bán, đã hủy, đã sử dụng) và quản lý các đơn đặt vé.
5. **Báo cáo và thống kê**: Quản trị viên có thể xem báo cáo doanh thu theo từng mốc thời gian, thống kê số lượng vé bán được, các chuyến xe bán chạy, và các dữ liệu khác để phục vụ công tác quản lý.

Tóm lại, hệ thống quản lý đặt vé xe trực tuyến được thiết kế để cung cấp một nền tảng tiện lợi, hiệu quả cho cả người dùng và quản trị viên. Người dùng có thể dễ dàng tìm kiếm, chọn lựa chuyến xe, đặt vé và thanh toán, trong khi quản trị viên có thể quản lý các chuyến xe, tuyến đường, vé và báo cáo một cách thuận tiện. Với sự phân quyền rõ ràng và các chức năng quản lý đầy đủ, hệ thống không chỉ hỗ trợ quy trình đặt vé hiệu quả mà còn nâng cao trải nghiệm người dùng và sự quản lý chặt chẽ cho doanh nghiệp.

**Xác định các chức năng nghiệp vụ cần có của hệ thống:**

In đậm các động từ và bổ ngữ thể hiện chức năng tổng thể của hệ thống cần thiết kế từ mô tả bên trên ta được:

* Tìm kiếm chuyến xe
* Xem chi tiết chuyến xe
* Chọn chuyến
* Đặt vé
* Thanh toán
* Đăng ký tài khoản
* Đăng nhập
* Nhận vé
* Quản lý thông tin cá nhân
* Quản lý lịch sử đặt vé
* Quản lý vé
* Quản lý tuyến đường
* Quản lý chuyến đi
* Quản lý vé
* Báo cáo và thống kê

**Sơ đồ phân cấp chức năng:**

* Nhóm chức năng phía người dùng:

A diagram of a company

Description automatically generated

Hình 3.1: Sơ đồ phân cấp chức năng của nhóm chức năng chính phía người dùng

* Nhóm chức năng trang quản trị:

A diagram of a company

Description automatically generated

Hình 3.2: Sơ đồ phân cấp chức năng nhóm chức năng chính trang quản trị

#### Phân tích sơ đồ Use-case

Dựa vào những ý đã phân tích ở phần mô tả tổng quan về hệ thống, ta xác định được những actor tác động lên trên hệ thống đặt vé xe trực tuyến và các use-case tương ứng.

Nhóm người dùng tổng thể của hệ thống đặt vé xe trực tuyến bao gồm: Khách vãng lai, khách hàng và quản lý hệ thống

* Khách vãng lai:
  + Đăng ký tài khoản
  + Tra cứu vé
  + Tìm kiếm chuyến xe
  + Xem chi tiết chuyến xe
* Khách hàng: Kế thừa các chức năng của khách vãng lai, ngoài ra còn:
* Đăng nhập vào trang người dùng
* Chọn chuyến
* Đặt vé
* Thanh toán
* Quản lý thông tin
* Lịch sử đặt vé
* Quản lý hệ thống:
* Đăng nhập vào trang quản trị
* Quản lý xe
* Quản lý tuyến đường
* Quản lý chuyến xe
* Quản lý vé
* Báo cáo và thống kê doanh thu

##### Sơ đồ Use-case

* Sơ đồ Use-case tổng quát của hệ thống đặt vé xe trực tuyến:

A diagram of a diagram

Description automatically generated

Hình 3.3: Sơ đồ Use-case tổng quát của hệ thống đặt vé xe trực tuyến

* Sơ đồ Use-case chi tiết của hệ thống:

A diagram of a diagram

Description automatically generated

Hình 3.4: Sơ đồ Use-case chi tiết của chức năng đăng nhập

A diagram of a diagram

Description automatically generated

Hình 3.5: Sơ đồ Use-case chi tiết của chức năng đăng ký

A diagram of a diagram

Description automatically generated

Hình 3.6: Sơ đồ Use-case chi tiết chức năng quản lý chuyến xe

##### Đặc tả các sơ đồ Use-case

* Chức năng đăng nhập và đăng ký:

Bảng 3‑1: Bảng mô tả Use-case của chức năng đăng nhập

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên Use-case | Đăng nhập |
| Mô tả | Use-case cho phép khách hàng và quản lý hệ thống đăng nhập vào hệ thống website để thực hiện những chức năng sẵn có trên hệ thống. |
| Actor | Khách hàng, quản lý hệ thống |
| Sự kiện kích hoạt | Khi người dùng bắt đầu sử dụng hệ thống. |
| Tiền điều kiện | Người dùng phải có tài khoản trên hệ thống. |
| Hậu điều kiện | Người dùng đăng nhập thành công và sử dụng các chức năng trên hệ thống. |
| Luồng sự kiện chính | 1. Người dùng truy cập vào hệ thống. 2. Hệ thống hiển thị màn hình đăng nhập để người dùng nhập thông tin đăng nhập. 3. Người dùng nhập thông tin đăng nhập (tên tài khoản và mật khẩu). 4. Hệ thống kiểm tra tính hợp lệ của thông tin đăng nhập. 5. Nếu thông tin hợp lệ, người dùng đăng nhập thành công. 6. Hệ thống sẽ thông báo đăng nhập thành công và cho người dùng sử dụng các chức năng trên hệ thống |
| Luồng sự kiện phụ | A1 – Lỗi đăng nhập: Khi người dùng nhập sai thông tin đăng nhập   1. Người dùng thực hiện chức năng đăng nhập như mô tả trên nhưng nhập sai các thông tin như tên tài khoản và mật khẩu. 2. Hệ thống sẽ thống báo lỗi đăng nhập và yêu cầu người dùng nhập lại sau đó quay về bước 2 của luồng sự kiện chính. 3. Use-case kết thúc. |

Bảng 3‑2: Bảng mô tả Use-case của chức năng đăng ký

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên Use-case | Đăng ký |
| Mô tả | Use-case cho phép khách vãng lai tạo tài khoản trên hệ thống bằng cách nhập thông tin đăng ký hợp lệ, từ đó trở thành khách hàng và có thể sử dụng các chức năng của hệ thống. |
| Actor | Khách vãng lai (người chưa có tài khoản). |
| Sự kiện kích hoạt | Khi người dùng truy cập hệ thống và chọn chức năng đăng ký tài khoản. |
| Tiền điều kiện | Người dùng chưa có tài khoản trên hệ thống. |
| Hậu điều kiện | Người dùng đăng ký thành công và có thể sử dụng chức năng của hệ thống với vai trò khách hàng. |
| Luồng sự kiện chính | 1. Người dùng chọn chức năng "Đăng ký" trên hệ thống. 2. Hệ thống hiển thị màn hình nhập thông tin đăng ký tài khoản. 3. Người dùng nhập thông tin đăng ký (ví dụ: họ tên, email, số điện thoại, mật khẩu). 4. Hệ thống kiểm tra tính hợp lệ của thông tin đăng ký (bao gồm kiểm tra định dạng email, số điện thoại, và mật khẩu). 5. Nếu thông tin hợp lệ, hệ thống lưu thông tin đăng ký vào cơ sở dữ liệu. 6. Hệ thống thông báo đăng ký thành công. Người dùng được chuyển sang màn hình đăng nhập để đăng nhập vào hệ thống. |
| Luồng sự kiện phụ | A1 – Lỗi thông tin đăng ký:   1. Khi người dùng nhập thông tin không hợp lệ (ví dụ: email sai định dạng, số điện thoại không đúng hoặc mật khẩu yếu). 2. Hệ thống hiển thị thông báo lỗi chi tiết (ví dụ: "Email không hợp lệ", "Mật khẩu phải có ít nhất 8 ký tự"). 3. Người dùng sửa lại thông tin và quay lại bước 3 của luồng chính. 4. Use-case kết thúc khi người dùng quyết định không tiếp tục đăng ký. |

* Chức năng quản lý chuyến xe:

Bảng 3‑3: Bảng mô tả Use-case của chức năng quản lý chuyến xe

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| Tên Use-case | Quản lý chuyến xe |
| Mô tả | Use-case cho phép quản lý hệ thống thực hiện các thao tác quản lý thông tin liên quan đến chuyến xe, bao gồm việc thêm mới, sửa đổi, hủy chuyến, cập nhật thông tin vé, và hỗ trợ đặt vé hoặc in vé cho khách hàng. |
| Actor | Quản lý hệ thống |
| Sự kiện kích hoạt | Khi quản lý hệ thống muốn thực hiện thao tác quản lý chuyến xe hoặc vé của các chuyến xe. |
| Tiền điều kiện | * Quản lý hệ thống đã đăng nhập vào hệ thống. * Thông tin chuyến xe đã tồn tại (ngoại trừ khi thêm mới chuyến xe). |
| Hậu điều kiện | * Thông tin chuyến xe hoặc vé được cập nhật thành công trên hệ thống. * Hành động tương ứng (như thêm, sửa, hủy, in vé) được thực hiện chính xác. |
| Luồng sự kiện chính | 1. Quản lý hệ thống đăng nhập vào hệ thống. 2. Quản lý chọn chức năng "Quản lý chuyến xe". 3. Hệ thống hiển thị danh sách chuyến xe hiện tại. 4. Quản lý thực hiện các thao tác như:  * **Thêm chuyến xe mới**: Nhập thông tin chuyến xe và lưu vào hệ thống. * **Sửa thông tin chuyến xe**: Cập nhật thông tin chuyến xe khi có thay đổi. * **Hủy chuyến xe**: Xóa chuyến xe nếu không còn hoạt động. * **Quản lý vé theo từng chuyến xe**:   + 1. Xem chi tiết số lượng vé đã bán, ghế còn trống, giá vé.     2. Cập nhật thông tin vé khi cần.     3. Đặt vé cho khách hàng hoặc hủy vé.     4. In vé cho khách hàng. |
| Luồng sự kiện phụ | A1 – Lỗi khi thêm chuyến xe mới:   1. Nếu thông tin nhập thiếu hoặc không hợp lệ (ví dụ: thiếu giờ khởi hành, tuyến đường không hợp lệ). 2. Hệ thống hiển thị thông báo lỗi và yêu cầu người dùng sửa lại thông tin. 3. Quản lý hệ thống quay lại bước 4 của luồng sự kiện chính để sửa đổi.   A2 – Hủy vé thất bại:   1. Nếu vé đã được sử dụng (chuyến xe đã khởi hành), hệ thống không cho phép hủy vé. 2. Hệ thống hiển thị thông báo lỗi. 3. Quản lý không thể thực hiện thao tác hủy vé này. |

#### Thiết kế cơ sở dữ liệu

##### Xác định các class

Sau khi xem qua các yêu cầu của hệ thống, sơ đồ Use-case đã phân tích bên trên và nghiên cứu kỹ các hệ thống tương tự, ta xác định được các class như sau:

1. **Phân tích Use-case “Quản lý tài khoản, Đăng Nhập và Đăng Ký”**: Ta có các class **Users** đại diện cho tài khoản người dùng, bao gồm thông tin đăng nhập, email, mật khẩu và trạng thái. Class này chịu trách nhiệm quản lý thông tin người dùng và thực hiện các chức năng như đăng nhập, đăng ký, và đăng xuất. Bên cạnh đó, class **Role** đại diện cho quyền hạn của từng tài khoản. Class này có nhiệm vụ phân quyền truy cập và xác định chức năng mà từng loại tài khoản được phép thực hiện trong hệ thống.
2. **Phân tích Use-case “Quản lý chuyến xe và quản lý tuyến đường”**: Ta xác định được class **Routes** đại diện cho thông tin tuyến đường, bao gồm các điểm xuất phát và điểm đến. Class này liên kết với class **Trips**, đại diện cho các chuyến xe cụ thể, bao gồm thông tin về thời gian khởi hành, trạng thái, và xe liên quan. Các class này giúp quản lý chi tiết các hành trình, tuyến đường và chuyến xe chính của hệ thống đặt vé xe trực tuyến.
3. **Phân tích Use-case “Quản lý xe”**: Ta xác định được class **Bus** đại diện cho thông tin xe, bao gồm loại xe, sức chứa, và giá vé mỗi ghế. Class **SeatLayout** mô tả bố trí ghế trên các xe, hỗ trợ quản lý cấu hình chỗ ngồi. Liên quan đến bố trí ghế, class **Seat** đại diện cho từng ghế cụ thể trên xe và class **BusSeatStatus** giúp theo dõi trạng thái ghế trong từng chuyến đi, như ghế đã đặt hay còn trống.
4. **Phân tích Use-case “Quản lý vé”**: Ta xác định class **Ticket** đại diện cho thông tin vé xe, bao gồm thông tin về chuyến đi, ghế ngồi, và người dùng. Class này chịu trách nhiệm quản lý việc đặt vé, hủy vé và lưu trữ trạng thái của từng vé.

Với các class đã xác định, hệ thống đảm bảo quản lý toàn diện các thông tin về tài khoản, quyền hạn, chuyến đi, xe, ghế ngồi và vé xe. Các class này không chỉ đáp ứng các yêu cầu hiện tại mà còn hỗ trợ dễ dàng mở rộng và tích hợp thêm các chức năng trong tương lai.

##### Bổ sung thuộc tính cho các class

Dựa vào phân tích ở phần trên, ta tiến hành bổ sung thuộc tính cụ thể cho các class đã phân tích như sau:

1. **Users** (id, name, email, password, phone, role\_id, image\_path, created\_at, updated\_at): Lưu trữ thông tin tài khoản người dùng bao gồm tên, email, mật khẩu, số điện thoại, quyền hạn, và trạng thái tài khoản.
2. **Role** (id, name, description, created\_at, updated\_at): Xác định quyền hạn của từng loại tài khoản với các thuộc tính mô tả chi tiết quyền và vai trò.
3. **Routes** (id, name, departure, destination, created\_at, updated\_at): Đại diện cho tuyến đường với các thuộc tính xác định điểm xuất phát, điểm đến, và thông tin quản lý tuyến.
4. **Trips** (id, route\_id, bus\_id, departure\_time, duration, status, created\_at, updated\_at): Quản lý thông tin chuyến đi bao gồm tuyến đường, xe bus sử dụng, thời gian khởi hành và trạng thái.
5. **Bus** (id, license\_plate, type, capacity, price\_per\_seat, created\_at, updated\_at): Đại diện cho các xe bus với thông tin chi tiết như biển số, loại xe, sức chứa và giá vé.
6. **SeatLayout** (id, name, description, created\_at, updated\_at): Xác định cách bố trí ghế của các loại xe bus.
7. **Seat** (id, bus\_id, name, floor, row\_number, column\_number, created\_at, updated\_at): Lưu trữ thông tin từng ghế cụ thể trên xe bus, bao gồm vị trí và tầng.
8. **BusSeatStatus** (id, trip\_id, seat\_id, created\_at, updated\_at): Theo dõi trạng thái ghế trên mỗi chuyến đi, giúp kiểm soát việc đặt và sử dụng ghế.
9. **Ticket** (id, trip\_id, user\_id, seat\_id, total\_price, status, created\_at, updated\_at): Đại diện cho vé xe, chứa các thông tin về chuyến đi, người đặt, và ghế ngồi đã đặt.

A diagram of a computer

Description automatically generated with medium confidence

Hình 3.7: Sơ đồ class

##### Giải thích mỗi quan hệ giữa các class

* Một vai trò (Role) được gán cho một hoặc nhiều tài khoản người dùng (Users), trong khi mỗi tài khoản chỉ thuộc về một vai trò cụ thể (1-N). Điều này đảm bảo quản lý quyền hạn người dùng theo vai trò được phân định rõ ràng.
* Một tài khoản (Users) được cấp cho mỗi người dùng, mỗi người chỉ có một tài khoản duy nhất để truy cập và sử dụng hệ thống quản lý (1-1).
* Một tuyến đường (Routes) có thể liên kết với nhiều chuyến đi (Trips), nhưng mỗi chuyến đi chỉ thuộc về một tuyến đường cụ thể (1-N). Điều này hỗ trợ quản lý lộ trình và lịch trình đi lại của các chuyến xe.
* Một chuyến đi (Trips) liên kết với một xe bus (Bus), nhưng một xe bus có thể phục vụ nhiều chuyến đi khác nhau (N-1). Mối quan hệ này đảm bảo mỗi chuyến đi được gắn với thông tin xe bus sử dụng.
* Một xe bus (Bus) có thể chứa nhiều ghế (Seat), nhưng mỗi ghế thuộc về duy nhất một xe bus (1-N). Mối quan hệ này hỗ trợ quản lý vị trí ghế ngồi và sức chứa của từng xe.
* Một bố trí ghế (SeatLayout) có thể áp dụng cho nhiều xe bus, nhưng mỗi xe bus chỉ được cấu hình theo một bố trí ghế duy nhất (1-N). Điều này đảm bảo tính linh hoạt trong thiết kế xe.
* Một người dùng (Users) có thể đặt nhiều vé (Ticket), nhưng mỗi vé chỉ liên quan đến một người dùng (1-N). Điều này giúp quản lý giao dịch và lịch sử đặt vé của người dùng.
* Một vé (Ticket) liên kết với một chuyến đi (Trips) và một ghế (Seat) cụ thể, đảm bảo rằng mỗi vé đại diện cho một ghế đã đặt trên một chuyến đi (1-1).
* Một chuyến đi (Trips) có thể chứa nhiều vé được đặt trước (Ticket), nhưng mỗi vé chỉ thuộc về một chuyến đi cụ thể (1-N). Mối quan hệ này hỗ trợ theo dõi trạng thái đặt vé và sức chứa.
* Một trạng thái ghế (BusSeatStatus) được quản lý cho từng ghế (Seat) và từng chuyến đi (Trips), hỗ trợ theo dõi trạng thái ghế trống hoặc đã được đặt (1-1).

##### Chuyển đổi class sang lược đồ cơ sở dữ liệu quan hệ

Thông qua các class đã phân tích như trên, ta xác định được các bảng dữ liệu tượng trưng cho các class và các thuộc tính nằm trong class là các cột dữ liệu của bảng:

**Users** (**id**, name, email, password, phone, *role\_id*, image\_path, created\_at, updated\_at)

**Role** (**id**, name, description, created\_at, updated\_at)

**Routes** (**id**, name, departure, destination, created\_at, updated\_at)

**Trips** (**id**, *route\_id*, *bus\_id*, departure\_time, duration, status, created\_at, updated\_at)

**Bus** (**id**, license\_plate, type, capacity, price\_per\_seat, created\_at, updated\_at)

**SeatLayout** (**id**, name, description, created\_at, updated\_at)

**Seat** (**id**, *bus\_id*, name, floor, row\_number, column\_number, created\_at, updated\_at)

**BusSeatStatus** (**id**, *trip\_id*, *seat\_id*, created\_at, updated\_at)

**Ticket** (**id**, *trip\_id*, *user\_id*, seat\_id, total\_price, status, created\_at, updated\_at)

*Chú thích:*

* Field **chữ đậm** và gạch chân là khóa chính (Primary Key).
* Field *chữ nghiên*g và *gạch đứt* là khóa ngoại (Foreign Key).

##### Mô tả chi tiết các bảng dữ liệu

Bảng 3‑4: Bảng mô tả tổng quát các bảng dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Diễn giải** |
| 1 | Role | Bảng quyền trên hệ thống |
| 2 | Users | Bảng lưu thông tin tài khoản |
| 3 | Routes | Bảng lưu thông tin tuyến đường |
| 4 | Trips | Bảng lưu thông tin các chuyến xe |
| 5 | Bus | Bảng lưu thông tin các xe trên hệ thống |
| 6 | SeatLayout | Bảng lưu sơ đồ ghế của xe |
| 7 | Seat | Bảng lưu thông tin ghế của xe |
| 8 | BusSeatStatus | Bảng lưu thông tin trạng thái ghế của mỗi chuyến xe |
| 9 | Ticket | Bảng lưu thông tin vé |

Bảng 3‑5: Bảng mô tả chi tiết bảng dữ liệu Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | id | Mã tài khoản | INT | Khóa chính |
| 2 | name | Tên đăng nhập | NVARCHAR(100) |  |
| 3 | password | Mật khẩu | NVARCHAR(150) |  |
| 4 | email | Họ và tên | NVARCHAR(250) |  |
| 5 | role\_id | Mã quyền tài khoản | INT | Khóa ngoại |
| 6 | phone | Số điện thoại | VARCHAR(12) |  |
| 7 | image\_path | Đường dẫn ảnh | TEXT |  |
| 8 | created\_at | Ngày tạo | DATETIME |  |
| 9 | updated\_at | Ngày cập nhật | DATETIME |  |

Bảng 3‑6: Bảng mô tả chi tiết bảng dữ liệu Role

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Mô tả** | **Kiểu dữ liệu** | **Ràng buộc** |
| 1 | id | Mã quyền | INT | Khóa chính |
| 2 | name | Tên quyền | NVARCHAR(100) |  |
| 3 | description | Mô tả | NVARCHAR(150) |  |
| 4 | created\_at | Ngày tạo | DATETIME |  |
| 5 | updated\_at | Ngày cập nhật | DATETIME |  |

A computer screen shot of a computer

Description automatically generated

Hình 3.8: Sơ đồ mô hình cơ sở dữ liệu quan hệ

### Thiết kế giao diện người dùng

Hệ thống đặt vé xe trực tuyến được xây dựng nhằm mang lại trải nghiệm thuận tiện cho cả khách hàng và nhà quản lý. Để đạt được mục tiêu này, giao diện người dùng (UI) và trải nghiệm người dùng (UX) là các yếu tố quan trọng cần được thiết kế một cách trực quan, thân thiện và dễ sử dụng. Dưới đây là các phân tích và thiết kế giao diện của từng thành phần chính trong hệ thống.

#### Xây dựng Wireframe và Mockup các trang chính

Giao diện của hệ thống bao gồm các thành phần chính như: trang chủ, trang chi tiết chuyến xe, và trang quản lý (Admin Dashboard). Trong đó, **trang chủ** là nơi khách hàng thực hiện tìm kiếm các chuyến xe phù hợp với nhu cầu thông qua form nhập thông tin như nơi đi, nơi đến và ngày khởi hành. Sau khi tìm kiếm, kết quả sẽ hiển thị danh sách các chuyến xe với thông tin cơ bản như tuyến đường, thời gian khởi hành, giá vé và liên kết đến trang chi tiết từng chuyến xe. Đối với **trang chi tiết chuyến xe**, giao diện sẽ cung cấp đầy đủ thông tin về chuyến đi, bao gồm sơ đồ ghế theo thời gian thực, các trạng thái ghế (trống, đã đặt, đã chọn), và form đặt vé. Form này yêu cầu khách hàng nhập thông tin cá nhân như họ tên và số điện thoại để xác nhận đặt vé. Ngoài ra, hệ thống cũng tích hợp **Admin Dashboard** dành riêng cho quản trị viên. Tại đây, các chức năng như quản lý chuyến xe, quản lý vé, và xem báo cáo thống kê được sắp xếp khoa học thông qua sidebar (thanh điều hướng bên trái) và các bảng dữ liệu trực quan.

A screenshot of a web page

Description automatically generated

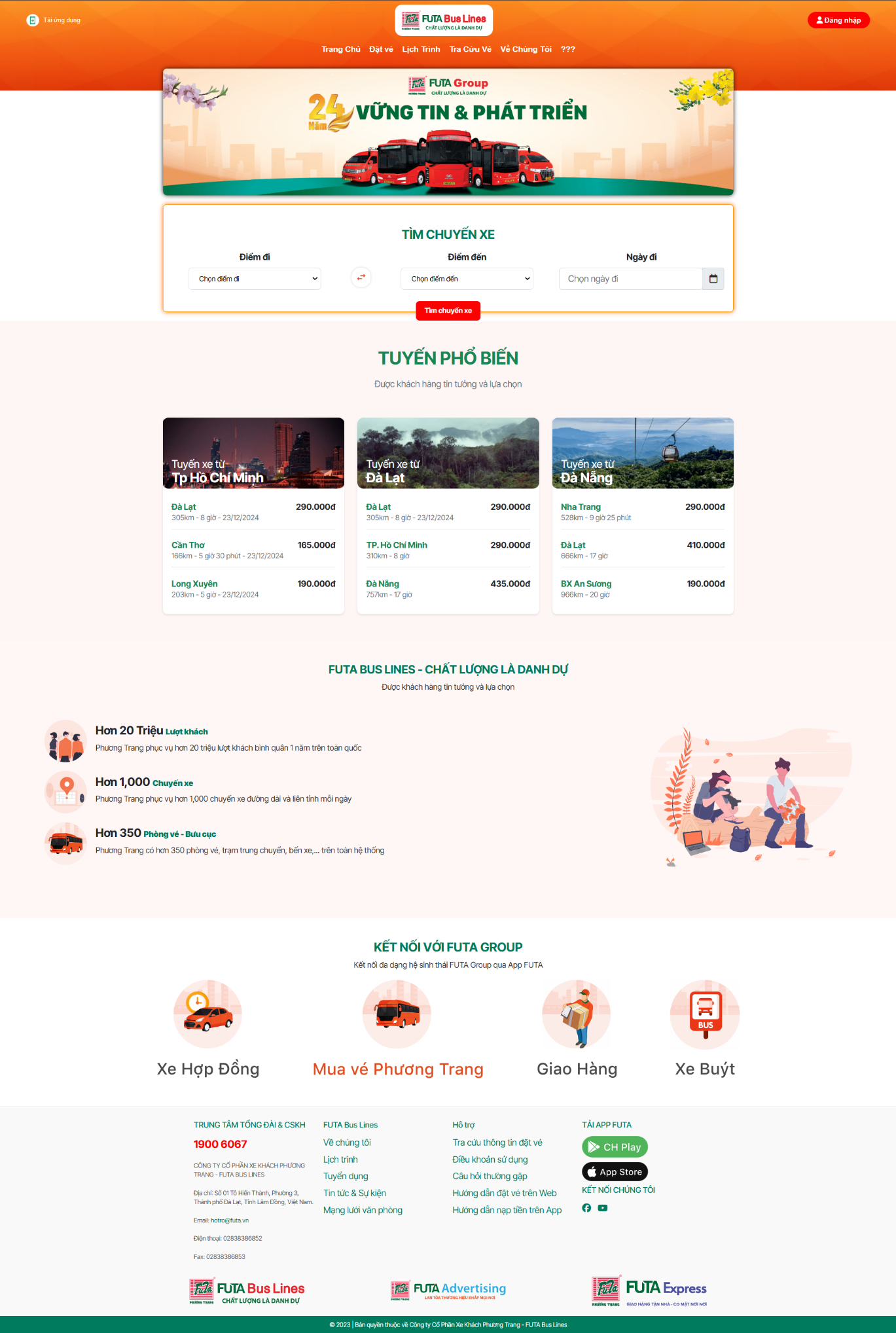
Hình 3.9: Wireframe phác họa tổng thể trang chủ của hệ thống

#### ****Thiết kế UI/UX với Blade Template Engine của Laravel****

Hệ thống giao diện được xây dựng dựa trên **Blade Template Engine**, một công cụ mạnh mẽ trong Laravel, cho phép tổ chức và xây dựng giao diện một cách linh hoạt và dễ quản lý. Blade cho phép sử dụng các cấu trúc điều kiện, vòng lặp, và phân chia giao diện thành các thành phần tái sử dụng, giúp tối ưu hóa hiệu quả phát triển giao diện. Kiến trúc giao diện của hệ thống được chia thành các layout tổng thể và các thành phần trang riêng biệt. Layout chính (thường được đặt tên là app.blade.php) chịu trách nhiệm quản lý cấu trúc khung cơ bản, bao gồm phần header, footer, và nội dung động (thông qua sử dụng lệnh @yield để chứa các nội dung của từng trang). Toàn bộ giao diện được thiết kế linh hoạt, phù hợp với người dùng. Header hiển thị các nút "Đăng nhập" và "Đăng ký" cho khách chưa đăng nhập, hoặc "Tài khoản cá nhân" và "Trang quản lý" cho người đã đăng nhập. Bên dưới header là navbar với các mục như "Trang chủ", "Tìm chuyến xe", "Liên hệ", và "Hỗ trợ". Footer được thiết kế tối giản, hiển thị thông tin bản quyền và các liên kết hữu ích.

Trang chủ tập trung vào tính rõ ràng và thân thiện. Trung tâm là form tìm kiếm chuyến xe, giúp người dùng dễ dàng nhập thông tin điểm đi, điểm đến và thời gian. Bên dưới là các section tuyến phổ biến, hiển thị các chuyến xe được nhiều người chọn, dưới dạng danh sách hoặc thẻ với thông tin như tuyến đường, thời gian, giá vé, và trạng thái còn chỗ. Người dùng có thể nhấp vào "Xem chi tiết" để xem thêm thông tin chuyến đi. Trang chi tiết chuyến xe cung cấp thông tin cụ thể như thời gian khởi hành, điểm đón/trả khách, cùng sơ đồ ghế trực quan. Các trạng thái ghế (còn trống, đã đặt) được phân biệt bằng màu sắc. Form đặt vé hỗ trợ khách hàng hoàn tất giao dịch dễ dàng và nhanh chóng, đi kèm thông tin tóm tắt chuyến đi và thanh toán.

Tại trang quản trị tức Admin Dashboard là trung tâm quản lý của hệ thống, tập trung vào các chức năng quản trị quan trọng như quản lý chuyến xe, vé và báo cáo doanh thu. Giao diện được thiết kế trực quan và dễ sử dụng, với một sidebar rõ ràng giúp người quản trị dễ dàng thực hiện các tác vụ như thêm, sửa, xóa thông tin chuyến xe, cũng như theo dõi và phân tích báo cáo doanh thu. Mỗi mục trong sidebar được phân chia hợp lý, giúp admin nhanh chóng tiếp cận các phần cần thiết để điều hành và quản lý hiệu quả hệ thống, từ việc giám sát lịch trình xe cho đến việc tối ưu hóa doanh thu.



Hình 3.10: Layout tổng thể sau khi thiết kế của trang chủ

## THỰC NGHIỆM VÀ TRIỂN KHAI HỆ THỐNG

### Mô tả cấu trúc thư mục code

Việc mô tả cấu trúc thư mục của dự án là bước quan trọng để người đọc hiểu được cách tổ chức mã nguồn, giúp dễ dàng bảo trì và phát triển hệ thống trong tương lai.

#### Cấu trúc thư mục chính

* **app/**: Chứa mã nguồn chính của ứng dụng, bao gồm các Controllers, Models, Middleware, và Services.
* **bootstrap/**: Chứa các tệp khởi động cho ứng dụng, bao gồm cả tệp app.php.
* **config/**: Lưu trữ các tệp cấu hình cho ứng dụng.
* **database/**: Bao gồm các tệp liên quan đến cơ sở dữ liệu như migrations, seeders, và factories.
* **public/**: Thư mục công khai chứa các tài nguyên như CSS, JavaScript, và hình ảnh. Tệp index.php của Laravel cũng nằm ở đây.
* **resources/**: Chứa các tài nguyên như Blade templates, ngôn ngữ, và các tệp asset.
* **routes/**: Bao gồm các tệp định nghĩa các route cho ứng dụng.
* **storage/**: Dùng để lưu trữ các tệp được tạo ra bởi ứng dụng như logs, cache, và các tệp tải lên.
* **tests/**: Chứa các tệp kiểm thử cho ứng dụng.
* **vendor/**: Thư mục do Composer quản lý, chứa các thư viện bên thứ ba.

A screenshot of a computer screen

Description automatically generated

Hình 4.1: Cấu trúc thư mục của dự án

#### Cấu trúc các tệp tin chính

* **Tệp .env** chứa các biến môi trường quan trọng, giúp cấu hình ứng dụng mà không cần thay đổi mã nguồn. Điều này cho phép bạn dễ dàng điều chỉnh cấu hình cho các môi trường khác nhau (phát triển, thử nghiệm, sản xuất).
* **artisan** là giao diện dòng lệnh (CLI) của Laravel, giúp thực hiện nhiều tác vụ quan trọng như tạo controller, chạy migration, seed dữ liệu, và nhiều hơn nữa.
* **composer.json** là tệp tin cấu hình cho Composer, trình quản lý phụ thuộc (dependencies) của PHP. Nó liệt kê các thư viện mà dự án phụ thuộc vào và các thông tin liên quan đến dự án.
* **package.json** là tệp tin cấu hình cho npm (Node Package Manager), quản lý các gói JavaScript và các công cụ frontend như Webpack, Laravel Mix, Tailwind CSS, v.v.

A screenshot of a computer program

Description automatically generated

Hình 4.2: Cấu trúc các tệp tin quan trọng của dự án

### Phát triển Backend (Laravel, PHP)

#### Xây dựng các route, controller, và model trong Laravel

##### Route

Trong Laravel, việc xây dựng cấu trúc ứng dụng bắt đầu từ việc định nghĩa các tuyến đường (**routes**) để xử lý các yêu cầu từ người dùng. Các route được định nghĩa rõ ràng trong các file như web, giúp phân loại và quản lý các endpoint một cách hiệu quả. Mỗi route thường được liên kết với một phương thức trong controller, đảm nhận việc xử lý logic nghiệp vụ.

A screen shot of a computer code

Description automatically generated

Hình 4.3: Ví dụ minh họa cho Routes(admin) trong web.php

##### Controller

Controllers đóng vai trò là lớp trung gian giữa các route và các mô hình dữ liệu (models). Chúng nhận dữ liệu từ yêu cầu, xử lý thông tin cần thiết và gọi các phương thức của mô hình để tương tác với cơ sở dữ liệu. Nhờ vào tính tổ chức cao, controllers giúp mã nguồn trở nên dễ bảo trì và mở rộng.

A screen shot of a computer program

Description automatically generated

Hình 4.4: Minh họa Controller (HomeController)

##### Model

Models trong Laravel sử dụng Eloquent ORM để đại diện cho các bảng trong cơ sở dữ liệu. Chúng cung cấp các phương thức tiện ích để thực hiện các thao tác CRUD (Create, Read, Update, Delete) một cách dễ dàng và hiệu quả. Việc sử dụng models giúp giảm thiểu việc viết các truy vấn SQL phức tạp, đồng thời đảm bảo tính toàn vẹn của dữ liệu thông qua các quy tắc và ràng buộc được định nghĩa.

A screen shot of a computer program

Description automatically generated

Hình 4.5: Minh họa Model (Role)

#### Quản lý cơ sở dữ liệu với Eloquent ORM

Eloquent ORM là công cụ mạnh mẽ của Laravel cho phép lập trình viên tương tác với cơ sở dữ liệu thông qua các mô hình đối tượng (object models) thay vì viết các câu truy vấn SQL trực tiếp. Điều này không chỉ làm giảm nhầm lẫn mà còn tăng tốc độ phát triển ứng dụng.

Thông qua Eloquent, các mối quan hệ giữa các bảng có thể được định nghĩa một cách trực quan, chẳng hạn như quan hệ một-một (one-to-one), một-nhiều (one-to-many), và nhiều-nhiều (many-to-many). Điều này giúp dễ dàng truy xuất và quản lý dữ liệu liên quan mà không cần thực hiện các join phức tạp.

A screen shot of a computer screen

Description automatically generated

Hình 4.6: Minh họa cho Migration

Trong quá trình phát triển ứng dụng, việc quản lý và khởi tạo dữ liệu mẫu (**seed data**) là một yếu tố quan trọng giúp đảm bảo rằng hệ thống luôn có dữ liệu cần thiết để hoạt động đúng chức năng, đặc biệt trong môi trường phát triển và thử nghiệm. Laravel cung cấp một công cụ mạnh mẽ gọi là Seeder để hỗ trợ việc này một cách tự động và hiệu quả.

**Ví dụ**, trong dự án FutaBuslines-CD1, tệp [RolesSeeder.php](vscode-file://vscode-app/d:/Program%20Files/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) được sử dụng để khởi tạo các vai trò người dùng như "admin" và "user". Khi chạy Seeder này, các vai trò này sẽ được thêm vào bảng roles trong cơ sở dữ liệu mà không cần phải nhập liệu thủ công mỗi lần làm mới dữ liệu. Điều này đảm bảo rằng hệ thống luôn có sẵn các vai trò cơ bản cần thiết để phân quyền truy cập, từ đó hỗ trợ cho các chức năng xác thực và phân quyền người dùng trong ứng dụng.

A screenshot of a computer program

Description automatically generated

Hình 4.7: Minh họa về Seeder

#### Triển khai xác thực người dùng và phân quyền

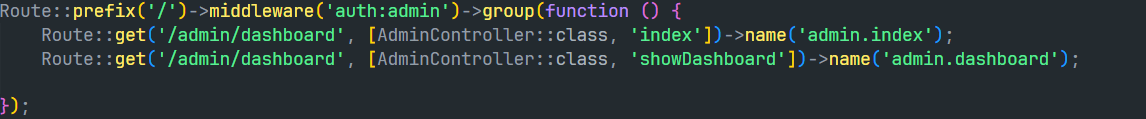
Thông qua Laravel's Authentication scaffolding, lập trình viên có thể nhanh chóng triển khai các tính năng cơ bản như đăng nhập, đăng xuất và đăng ký người dùng. Các biện pháp bảo mật như hashing mật khẩu, xác thực hai yếu tố (2FA), và khóa tài khoản sau nhiều lần đăng nhập thất bại cũng dễ dàng được tích hợp.

Phân quyền người dùng được quản lý thông qua các middleware và hệ thống roles-permissions. Bằng cách định nghĩa các quyền truy cập cụ thể cho từng vai trò, Laravel đảm bảo rằng người dùng chỉ có thể truy cập vào những phần của ứng dụng mà họ được phép. Điều này không chỉ tăng cường bảo mật mà còn giúp quản lý người dùng một cách linh hoạt và hiệu quả.

A screen shot of a computer program

Description automatically generated

Hình 4.8: Minh họa triển khai xác thực phân quyền



Hình 4.9: Minh họa sử dụng phân quyền trong routes

### Phát triển Frontend

#### Tích hợp giao diện người dùng với Blade Template

Việc sử dụng Blade Template cũng mang lại lợi ích về mặt tối ưu hóa hiệu suất. Các tệp Blade được biên dịch thành mã PHP gốc, giúp giảm tối đa thời gian xử lý và tải trang, đảm bảo ứng dụng hoạt động mượt mà ngay cả khi có lượng truy cập lớn. Đồng thời, Blade cũng hỗ trợ việc cache các phần giao diện tĩnh, further nâng cao tốc độ phản hồi của ứng dụng.

Cấu Trúc Template Đơn Giản: Layouts (Bố cục): Tạo các layout cơ bản chứa các phần chung như header, footer, sidebar. Điều này giúp giảm thiểu việc lặp lại mã nguồn và dễ dàng quản lý

A computer screen with text

Description automatically generated

Hình 4.10: Minh họa layout trong blade template

Kế Thừa và Mở Rộng giao diện: Kế thừa layouts, các trang cụ thể sẽ kế thừa từ layout chính và chỉ cần định nghĩa những phần thay đổi.

A screenshot of a computer

Description automatically generated

Hình 4.11: Minh họa cho việc kế thừa layout từ admin

#### Sử dụng Bootstrap responsive design

Bootstrap là một trong những framework CSS phổ biến nhất, cung cấp các công cụ mạnh mẽ để xây dựng giao diện phản hồi (responsive) và hiện đại.

* Cài Đặt Bootstrap:

A screen shot of a computer code

Description automatically generated

Hình 4.12: Cài đặt Boostrap

* Sử Dụng Các Thành Phần Sẵn Có: Navbar, Modals, Forms,…: Bootstrap cung cấp các thành phần giao diện sẵn có giúp tiết kiệm thời gian phát triển và đảm bảo tính nhất quán

A screen shot of a computer code

Description automatically generated

Hình 4.13: Thành phần giao diện form

#### Giao diện Frontend

##### Giao diện đăng nhập

Giao diện đăng nhập là một phần quan trọng của hệ thống, đảm bảo rằng chỉ những người dùng có quyền hợp lệ mới có thể truy cập vào các chức năng bảo mật của ứng dụng. Trong dự án FutaBuslines-CD1, giao diện đăng nhập được thiết kế tối ưu để cung cấp trải nghiệm người dùng mượt mà, thân thiện và bảo mật cao thông qua việc sử dụng Blade Template của Laravel kết hợp với các công nghệ frontend hiện đại như Bootstrap và Font Awesome

A screenshot of a web page

Description automatically generated

Hình 4.14: Giao diện trang đăng nhập của hệ thống

Bảng 4‑1: Bảng mô tả giao diện trang đăng nhập

|  |  |  |
| --- | --- | --- |
| **Ký hiệu** | **Chức năng** | **Mô tả** |
|  | Trường nhập số điện thoại | Cho phép người dùng nhập số điện thoại đã đăng ký làm tài khoản đăng nhập. |
|  | Trường nhập mật khẩu | Cho phép người dùng nhập mật khẩu để đăng nhập |
|  | Xác thực tài khoản | Khi nhấn nút "Đăng nhập", hệ thống sẽ kiểm tra thông tin đăng nhập, nếu hợp lệ sẽ chuyển người dùng đến trang chính. |
| **A black and orange text  Description automatically generated** | Đăng ký tài khoản | Cho phép người dùng chuyển sang Form đăng ký |

##### Giao diện chức năng đặt vé

Giao diện chọn ghế hiển thị sơ đồ chỗ ngồi hai tầng (tầng dưới và tầng trên), giúp người dùng dễ dàng chọn vị trí phù hợp theo sở thích. Các ghế có màu sắc khác nhau nhằm biểu thị trạng thái cụ thể như ghế trống (màu xanh dương), ghế đã bán (màu đỏ) và ghế đang chọn (màu xanh lá cây). Người dùng có thể click vào các ghế còn trống để lựa chọn, khi đó trạng thái ghế sẽ chuyển thành "đang chọn" để xác nhận rằng ghế đó đã được thêm vào yêu cầu đặt vé.

Thông tin lộ trình chuyến xe được trình bày chi tiết ở phần bên phải màn hình, bao gồm các thông tin như điểm khởi hành, điểm đến, thời gian xuất phát và thời gian dự kiến kết thúc hành trình. Chức năng này giúp người dùng kiểm tra và xác nhận lại thông tin chuyến đi trước khi tiến hành thanh toán. Bên dưới phần thông tin chuyến đi là mục hiển thị giá vé và tổng tiền thanh toán. Hệ thống tự động cập nhật tổng chi phí dựa trên công thức: Tổng tiền = Số lượng ghế \* Giá vé mỗi ghế, giúp người dùng dễ dàng theo dõi tổng chi phí thay đổi khi thêm hoặc bỏ ghế khỏi lựa chọn.

Cuối cùng, phần xử lý xác nhận. Khi nhấn nút "Xác nhận", hệ thống sẽ lưu thông tin vé vào cơ sở dữ liệu và hiển thị thông báo thành công.

A screenshot of a phone

Description automatically generated

Hình 4.15: Giao diện chức năng đặt vé xe

Bảng 4‑2: Bảng mô tả chức năng đặt vé

|  |  |  |
| --- | --- | --- |
| **Ký hiệu** | **Chức năng** | **Mô tả** |
|  | Chọn ghế | Người dùng có thể chọn ghế còn trống để đặt chỗ. Sau khi chọn, ghế sẽ chuyển sang trạng thái ‘đang chọn’. |
|  | Ghế không thể chọn | Ghế có màu đỏ là ghế đã được người khác đặt mua thành công, người dùng không thể chọn ghế này nữa. |
|  | Xác nhận ghế đang chọn | Khi người dùng chọn vào ghế, hệ thống tạm thời đánh dấu ghế đó để chuẩn bị thêm vào đơn đặt vé. |
|  | Hoàn tất đặt vé | Khi người dùng nhấn nút ‘Xác nhận’, hệ thống lưu thông tin vé và hoàn tất quá trình đặt vé. |
|  | Trở về chọn chuyến | Khi bấm vào nút “trở về” người dùng sẽ được trả về trang ‘chọn chuyến’ |
|  | Hiển thị tuyến đi | Thông tin chi tiết về hành trình bao gồm điểm xuất phát và điểm đến, ví dụ: "BX Nam Tuy Hòa - BX An Sương". |
|  | Hiển thị giá vé | Tổng chi phí cho vé xe được tính dựa trên số lượng ghế được chọn nhân với giá vé từng ghế. |
|  | Thời gian khởi hành | Hiển thị thời gian chính xác mà xe sẽ xuất bến. |
|  | Đếm số lượng ghế chọn | Hiển thị tổng số lượng ghế mà người dùng đã chọn trong phiên làm việc hiện tại. |

KẾT LUẬN

Qua quá trình nghiên cứu và triển khai đồ án, chúng em đã xây dựng thành công một hệ thống đặt vé xe khách trực tuyến sử dụng **PHP** và **Laravel Framework**, cung cấp giải pháp tiện lợi và hiệu quả cho người dùng trong việc tìm kiếm, lựa chọn và đặt vé, đồng thời hỗ trợ quản trị viên quản lý chuyến đi, ghế ngồi và vé một cách dễ dàng. Hệ thống đã hoàn thiện các chức năng chính như tìm kiếm chuyến xe theo tuyến đường, thời gian và loại xe, lựa chọn ghế ngồi với cập nhật thời gian thực, quản lý vé và lịch sử đặt vé, cùng với giao diện người dùng thân thiện và responsive trên nhiều thiết bị.

Tuy nhiên, dự án hiện tại vẫn còn một số hạn chế cần được khắc phục và cải tiến trong tương lai. Hiện tại, hệ thống chưa tích hợp các phương thức thanh toán trực tuyến như thẻ tín dụng hay ví điện tử, điều này hạn chế sự tiện lợi cho người dùng. Ngoài ra, các biện pháp bảo mật cần được nâng cao hơn với việc áp dụng xác thực hai yếu tố (2FA) và mã hóa dữ liệu nhạy cảm. Hiệu suất của hệ thống cũng cần được tối ưu hóa thêm để hỗ trợ số lượng người dùng tăng cao, cùng với việc triển khai kiểm thử tự động để đảm bảo chất lượng mã nguồn. Cuối cùng, cải thiện giao diện người dùng với các tính năng tương tác nâng cao sẽ nâng cao trải nghiệm người dùng một cách đáng kể.

Dù còn một số thách thức, nhưng dự án đã giúp chúng em nâng cao kỹ năng lập trình, quản lý dự án và hiểu rõ hơn về quy trình phát triển ứng dụng thực tế. Những kinh nghiệm thu được sẽ là nền tảng vững chắc để tiếp tục phát triển các dự án công nghệ phức tạp hơn trong tương lai. Chúng em hy vọng hệ thống sẽ đóng góp tích cực vào việc ứng dụng công nghệ trong lĩnh vực vận tải, mang lại lợi ích thiết thực cho doanh nghiệp và hành khách. Cuối cùng, chúng em xin chân thành cảm ơn sự hướng dẫn tận tình của Thầy Đỗ Hoàng Nam đã giúp chúng em tìm ra những sai xót trong quá trình thực hiện tiểu luận này, chúng mình đồng thời cũng gửi lời cảm ơn đến những người bạn đã đồng hành trong suốt thời gian học tập và thực hiện tiểu luận vừa qua những lần chia sẻ kiến thức, động viên lẫn nhau là một phần không thể thiếu để chúng mình có được những ý tưởng hay và những góp ý đã phần nào đó giúp chúng mình thấy được những thiếu xót và hoàn thiện nó tốt hơn.

TÀI LIỆU THAM KHẢO

[1] Laravel Documentation. (n.d.). Truy cập tại: <https://laravel.com/docs/10.x> . Vào ngày 11/01/2025.

[2] MySQL Documentation. (n.d.). Truy cập tại: <https://dev.mysql.com/doc/> . Vào ngày 11/01/2025

[3] PHP là gì? Những kiến thức cần biết về ngôn ngữ lập trình PHP. Truy cập tại: <https://dncloud.net/blog/php-la-gi/> . Vào ngày 12/01/2025

[4] Laravel là gì? Lợi ích của Laravel, hướng dẫn cài đặt Laravel. Truy cập tại: <https://wiki.matbao.net/laravel-la-gi-huong-dan-cai-dat-laravel-nhanh-chong/> . Vào ngày 12/01/2025

[5] MySQL là gì? Hướng dẫn Cài đặt và Sử dụng MySQL. Truy cập tại: <https://200lab.io/blog/mysql-la-gi> . Vào ngày 12/01./2025