

TRƯỜNG ĐẠI HỌC BÀ RỊA - VŨNG TÀU
KHOA KỸ THUẬT - CÔNG NGHỆ



BARIA VUNGTAU
UNIVERSITY
CAP SAINT JACQUES

ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI:

PHÁT TRIỂN WEBSITE CHO THUÊ XE VIỆT KEY

Trình độ đào tạo	: Đại học chính quy
Ngành	: Công nghệ thông tin
Chuyên ngành	: Công nghệ thông tin
Khóa học	: 2019-2023
Giảng viên hướng dẫn	: TS. Phan Ngọc Hoàng
Sinh viên thực hiện	: Nguyễn Tuấn Kiệt
Mã số sinh viên	: 19034192
Lớp	: DH19CT

Vũng tàu, ngày 15 tháng 05 năm 2023

LỜI CẢM ƠN

Để hoàn thành đề tài báo cáo đồ án này trước hết, tôi xin gửi đến quý thầy, cô giáo trong khoa Công nghệ kỹ thuật – Công Nghệ trường Đại học Bà Rịa – Vũng Tàu lời cảm ơn chân thành.

Đặc biệt, tôi xin gửi đến thầy Phan Ngọc Hoàng, người đã tận tình hướng dẫn, giúp đỡ tôi hoàn thành đề tài đồ án tốt nghiệp lời cảm ơn sâu sắc nhất.

Tôi xin chân thành cảm ơn Ban Lãnh Đạo, các phòng ban của công ty TNHH Niềm Tin, đã tạo điều kiện thuận lợi cho tôi được tìm hiểu thực tiễn trong suốt quá trình thực tập tại công ty.

Đồng thời xin chân thành cảm ơn Ban lãnh đạo trường Đại học Bà Rịa – Vũng Tàu đã cho môi trường học tập lành mạnh và năng động. Kính chúc Trường Đại học Bà Rịa – Vũng Tàu tiếp tục gặt hái nhiều thành công trên con đường giáo dục thế hệ trẻ.

Vì kiến thức bản thân còn hạn chế và trình độ lý luận cũng như kinh nghiệm thực tiễn còn hạn chế nên bài báo cáo không thể tránh khỏi những thiếu sót, tôi rất mong nhận được ý kiến đóng góp thầy, cô để tôi học thêm được nhiều kinh nghiệm và sẽ hoàn thiện bản thân đáp ứng với nhu cầu công việc thực tế trong tương lai.

Tôi xin chân thành cảm ơn!

LỜI CAM ĐOAN

Tôi xin cam đoan kết quả đạt được trong đồ án là sản phẩm của riêng cá nhân, không sao chép lại của người khác. Trong toàn bộ nội dung của đồ án, những điều được trình bày hoặc là của cá nhân hoặc là được tổng hợp từ nhiều nguồn tài liệu. Tất cả các tài liệu tham khảo đều có xuất xứ rõ ràng và được trích dẫn hợp pháp.

Tôi xin hoàn toàn chịu trách nhiệm và chịu mọi hình thức kỷ luật theo quy định cho lời cam đoan của mình.

Vũng Tàu, ngày 01 tháng 05 năm 2023

Sinh viên thực hiện

Nguyễn Tuấn Kiệt

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Vũng Tàu, ngày ... tháng ... năm 20...

Người hướng dẫn

TS. Phan Ngọc Hoàng

MỤC LỤC

DANH MỤC BẢNG	I
DANH MỤC SƠ ĐỒ	II
DANH MỤC HÌNH	III
LỜI NÓI ĐẦU	IV
CHƯƠNG 1 TỔNG QUAN	17
1.1 Lý do chọn đề tài	17
1.2 Mục tiêu và chức năng của hệ thống	18
1.2.1 Mục tiêu	18
1.2.2 Chức năng	18
1.3 Mô tả bài toán	19
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	21
2.1 Giới thiệu ngôn ngữ Python	21
2.1.1 Tổng quát	21
2.1.2 Lịch sử phát triển của Python	22
2.1.3 Đặc điểm của ngôn ngữ Python	23
2.1.4 Tại sao nên chọn Python	25
2.1.5 Ứng dụng ngôn ngữ Python	27
2.2 Giới thiệu Django-Python	27
2.2.1 Tổng quát	27
2.3 Giới thiệu Jinja2	28
2.3.1 Tổng quát	28
2.3.2 Sự ra đời	29
2.3.3 Tính ưu việt của Jinja2	29

2.4 Giới thiệu về phpMyAdmin và MySQL	30
2.4.1 Giới thiệu về phpMyAdmin	30
2.4.2 Giới thiệu về MySQL	30
2.5 Một số thư viện khác	31
CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG	32
3.1 Mô tả hệ thống	32
3.1.1 Website giới thiệu	32
3.1.2 Quản trị website	32
3.1.3 Quản lý xe máy	32
3.1.4 Quản lý chỗ đỗ	33
3.1.5 Quản lý nhân viên	34
3.1.6 Quản lý hóa đơn	35
3.1.7 Quản lý phiếu thuê	35
3.1.8 Quản lý đánh giá	36
3.1.9 Quản lý bài viết	37
3.1.10 Quản lý chi nhánh	37
3.1.11 Quản lý tài khoản người dùng	37
3.1.12 Mô Hình ER của toàn bộ hệ thống	38
3.2 Xây dựng các chức năng của hệ thống	38
3.2.1 Mô tả các chức năng chi tiết	38
3.2.2 Xác định use case	43
3.2.3 Đặc tả các use case	46
3.2.4 Quy trình cho thuê xe	52

3.3 Thiết kế hệ thống	53
3.3.1 Sơ đồ cơ sở dữ liệu	53
3.3.2 Chi tiết cơ sở dữ liệu	54
CHƯƠNG 4 XÂY DỰNG WEBSITE	60
4.1 Giao diện và chức năng khách	60
4.2 Giao diện và chức năng Admin	66
4.3 Chức năng của user	85
CHƯƠNG 5 KẾT LUẬN	99
5.1 Kết quả đạt được	99
5.2 Hướng phát triển	99
TÀI LIỆU THAM KHẢO	101
PHỤ LỤC	102
1. Sơ đồ cấu trúc	102
2. Code xử lý	103
2.1. Account	103
2.2. Branch	111
2.3. Category	112
2.4. Employees	113
2.5. Parking	116
2.6. Posts	122
2.7. Store	125

DANH MỤC BẢNG

Bảng 3-1 Bảng mô tả chức năng quản lý loại xe máy	38
Bảng 3-2 Bảng mô tả chức năng quản lý xe máy.	39
Bảng 3-3 Bảng mô tả chức năng quản lý chi nhánh của xe	39
Bảng 3-4 Bảng mô tả chức năng quản lý chỗ đỗ xe	39
Bảng 3-5 Bảng mô tả chức năng quản lý phiếu đỗ.	40
Bảng 3-6 Bảng mô tả chức năng quản lý nhân viên.	40
Bảng 3-7 Bảng mô tả chức năng điểm danh	40
Bảng 3-8 Bảng mô tả chức năng quản lý bảng lương	40
Bảng 3-9 Bảng mô tả chức năng quản lý phiếu thuê xe	41
Bảng 3-10 Bảng mô tả chức năng quản lý hóa đơn	41
Bảng 3-11 Bảng mô tả chức năng quản lý chi tiết hóa đơn	42
Bảng 3-12 Bảng mô tả chức năng quản lý đánh giá	42
Bảng 3-13 Bảng mô tả chức năng quản lý bài viết	42
Bảng 3-14 Bảng mô tả chức năng quản lý tài khoản người dùng	43
Bảng 3-15 Cơ sở dữ liệu bảng tài khoản	54
Bảng 3-16 Cơ sở dữ liệu bảng chi nhánh	54
Bảng 3-17 Cơ sở dữ liệu bảng loại xe	55
Bảng 3-18 Cơ sở dữ liệu bảng điểm danh nhân viên	55
Bảng 3-19 Cơ sở dữ liệu bảng nhân viên	55
Bảng 3-20 Cơ sở dữ liệu bảng tính lương nhân viên	56
Bảng 3-21 Cơ sở dữ liệu bảng phiếu đỗ	56
Bảng 3-22 Cơ sở dữ liệu bảng chỗ đỗ xe	56
Bảng 3-23 Cơ sở dữ liệu bảng bài viết	57
Bảng 3-24 Cơ sở dữ liệu bảng hóa đơn	57
Bảng 3-25 Cơ sở dữ liệu bảng chi tiết hóa đơn	58
Bảng 3-26 Cơ sở dữ liệu bảng phiếu mượn	58
Bảng 3-27 Cơ sở dữ liệu bảng xe	59

Bảng 3-28 Cơ sở dữ liệu bảng xe _____ 59

DANH MỤC SƠ ĐỒ

Sơ đồ 3-1 Sơ đồ Use case của ADMIN	44
Sơ đồ 3-2 Sơ đồ Use case của Guest	45
Sơ đồ 3-3 Sơ đồ Use case của User	46
Sơ đồ 3-4 Sơ đồ tuần tự đăng nhập thành công	46
Sơ đồ 3-5 Sơ đồ hợp tác đăng nhập thành công	47
Sơ đồ 3-6 Sơ đồ tuần tự đăng nhập thất bại	47
Sơ đồ 3-7 Sơ đồ hợp tác đăng nhập thất bại	47
Sơ đồ 3-8 Sơ đồ tuần tự quá trình quản lý	48
Sơ đồ 3-9 Sơ đồ hợp tác quá trình quản lý	48
Sơ đồ 3-10 Sơ đồ tuần tự quá trình điểm danh	49
Sơ đồ 3-11 Sơ đồ hợp tác quá trình điểm danh	49
Sơ đồ 3-12 Sơ đồ tuần tự quá trình thuê	50
Sơ đồ 3-13 Sơ đồ hợp tác quá trình thuê	50
Sơ đồ 3-14 Sơ đồ tuần tự quá trình đánh giá	51
Sơ đồ 3-15 Sơ đồ hợp tác quá trình đánh giá	51
Sơ đồ 3-16 Quy trình thuê	52
Sơ đồ 3-17 Sơ đồ cơ sở dữ liệu	53

DANH MỤC HÌNH

Hình 2.1 Biểu tượng của Python.....	21
Hình 2.2 Đặc điểm của ngôn ngữ Python.....	23
Hình 2.3 Tại sao nên chọn Python.....	25
Hình 2.4 Biểu tượng của Django framework.....	27
Hình 2.2.5 Biểu tượng của Jinja2	28
Hình 2.2.6 Biểu tượng của phpMyAdmin	30
Hình 3.1 Mô hình ER quản lý xe	33
Hình 3.2 Mô hình ER quản lý chỗ đỗ	34
Hình 3.3 Mô hình ER quản lý nhân viên	34
Hình 3.4 Mô hình ER quản lý Hóa đơn.....	35
Hình 3.5 Mô hình ER quản lý phiếu thuê	36
Hình 3.6 Mô hình ER quản lý đánh giá	37
Hình 3.7 Mô hình ER toàn hệ thống.....	38
Hình 4.1 Phần header, navbar và silder của trang chủ.....	60
Hình 4.2 Phần tính năng nổi bật và hướng dẫn thuê xe.....	61
Hình 4.3 giới thiệu về tính năng cho thuê chỗ đỗ xe	62
Hình 4.4 Phần các bài viết liên quan để chức năng của website	62
Hình 4.5 Phần footer của trang chủ	63
Hình 4.6 Giao diện trang giới thiệu	63
Hình 4.7 Giao diện hiển thị tất cả các xe	64
Hình 4.8 Giao diện xe theo loại “Xe số”.	64
Hình 4.9 Giao diện khi sử dụng chức năng tìm kiếm.....	65
Hình 4.10 Giao diện danh sách chỗ đỗ xe	65
Hình 4.11 Giao diện ảnh khi lọc các vị trí châu đức	66
Hình 4.12 Giao diện ảnh lọc các vị trí châu đức.....	66
Hình 4.13 Giao diện trang đăng nhập.....	67
Hình 4.14 Giao diện trang đăng nhận không thành công	67

Hình 4.15 Trang thông tin tài khoản.....	68
Hình 4.16 Trang quản lý dashboard	69
Hình 4.17 Chức năng trong tài khoản.....	69
Hình 4.18 Chức năng chỉnh sửa tài khoản và phân quyền cho tài khoản.....	69
Hình 4.19 Chức năng trong chi nhánh	70
Hình 4.20 Chức năng chỉnh sửa chi nhánh	70
Hình 4.21 Chức năng trong loại xe.....	71
Hình 4.22 Chức năng chỉnh sửa loại xe.....	71
Hình 4.23 Chức năng quản lý nhân viên	72
Hình 4.24 Chức năng quản lý chi chấm công nhân viên	73
Hình 4.25 Chức năng xuất bảng lương nhân viên	73
Hình 4.26 Bảng lương nhân viên.....	74
Hình 4.27 quản lý chỗ đỗ xe.....	74
Hình 4.28 quản lý chỉnh sửa chỗ đỗ xe.....	75
Hình 4.29 quản lý phiếu đỗ.....	75
Hình 4.30 quản lý chỉ sửa phiếu đỗ và thanh toán.....	76
Hình 4.31 Quản lý bài viết.....	77
Hình 4.32 Quản lý hóa đơn.....	77
Hình 4.33 Quản lý chỉnh sửa hóa đơn	78
Hình 4.34 Quản lý chỉnh sửa chi tiết hóa đơn	78
Hình 4.35 Xem doanh thu theo năm.....	79
Hình 4.36 Xem doanh thu theo năm 2023 – loại xe xe tay côn.....	80
Hình 4.37 Quản lý phiếu mượn	80
Hình 4.38 Quản lý chi tiết phiếu mượn	81
Hình 4.39 Chức năng in biên bản mượn xe	81
Hình 4.40 Biên bản mượn xe.....	82
Hình 4.41 Quản lý sản phẩm	82
Hình 4.42 Quản lý chi tiết sản phẩm	83

Hình 4.43 Quản lý đánh giá.....	84
Hình 4.44 Quản lý chỉnh sửa đánh giá.....	84
Hình 4.45 Trang hướng dẫn thuê xe	85
Hình 4.46 header khi đăng nhập	86
Hình 4.47 Trang đăng nhập	86
Hình 4.48 Đăng ký thất bại cần nhập trường gmail.....	87
Hình 4.49 Trang đăng ký thành công vào gmail để kích hoạt tài khoản	88
Hình 4.50 Đăng nhập thất bại do chưa kích hoạt tài khoản.....	89
Hình 4.51 chưa kích hoạt tài khoản	89
Hình 4.52 link kích hoạt tài khoản.....	90
Hình 4.53 Tài khoản đã được kích hoạt.....	90
Hình 4.54 Đăng nhập vào hệ thống	91
Hình 4.55 Thực hiện thao tác thuê xe.....	91
Hình 4.56 Thông tin khi thuê xe	92
Hình 4.57 Mail chủ cửa hàng nhận được.....	92
Hình 4.58 Hủy thuê xe.....	93
Hình 4.59 Mail hủy từ chủ cửa hàng nhận được	93
Hình 4.60 Dùng tài khoản admin để trả xe về kho	94
Hình 4.61 Hóa đơn khi trả xe.....	94
Hình 4.62 Đánh giá xe	95
<i>Hình 4.63 Lịch sử thuê xe</i>	<i>95</i>
Hình 4.64 Danh sách đặt chỗ đỗ xe	96
Hình 4.65 Form đặt chỗ đỗ xe	96
Hình 4.66 Trạng thái đổi khi thuê chỗ đỗ thành công	97
Hình 4.67 Mail chủ cửa hàng nhận được khi có người thuê chỗ đỗ xe.....	97
Hình 4.68 Trả chỗ cho thuê chỗ đỗ xe.....	98
Hình 4.69 Hóa đơn chỗ đỗ xe	98

LỜI NÓI ĐẦU

Trong thời đại hiện nay, Công nghệ Thông tin (CNTT) đã trở thành một công cụ vô cùng quan trọng hỗ trợ cho các doanh nghiệp trong quá trình chuyển đổi. Tối ưu hóa lợi ích từ các khoản đầu tư vào CNTT để duy trì khả năng cạnh tranh là một trong những mục tiêu chiến lược hàng đầu của các nhà lãnh đạo doanh nghiệp ngày nay. Những quyết định và hành động của họ đóng vai trò vô cùng quan trọng trong việc đảm bảo sự phát triển bền vững và thành công của doanh nghiệp trong thế giới ngày càng cạnh tranh này.

Trên thực tế, việc không sử dụng công nghệ thông tin trong doanh nghiệp trở nên vô cùng khó khăn. Vì vậy, một trong những ưu tiên hàng đầu của các nhà lãnh đạo doanh nghiệp ngày nay là tận dụng tối đa các khoản đầu tư vào CNTT để duy trì khả năng cạnh tranh. Để đạt được mục tiêu kinh doanh và phù hợp với năng lực khai thác công nghệ của doanh nghiệp, mỗi doanh nghiệp cần chọn cho mình mô hình đầu tư CNTT phù hợp. Thậm chí, ngay cả trong lĩnh vực nông nghiệp, máy tính cũng đã trở thành một phần không thể thiếu trong sản xuất, tài chính, nghiên cứu và mua sắm. Cho nên để thành công trong kinh doanh, thúc đẩy đổi mới CNTT là một yếu tố không thể bỏ qua.

Trong đó kinh doanh trao đổi và buôn bán, thuê mướn là ngành có nhiệm vụ cung cấp và phát triển toàn diện trên bề mặt kinh tế. Mọi ngành, nghề, doanh nghiệp đều phải có trao đổi để phát triển để hoàn thiện. Thuê mướn xe máy tự lái và chỗ đỗ xe cũng chính là một ngành vô cùng cần độ chính xác cao, quản lý chặt chẽ cùng với mang lại độ phủ sóng lớn mạnh cho nên công tác thiết lập cho từng công đoạn yêu cầu phải thật chính xác. Để quản lý các sản phẩm và sản xuất ra những người làm điều hành doanh nghiệp, cửa hàng đang gặp phải nhiều thách thức và ứng dụng phần mềm quản lý thuê xe máy tự lái là yêu cầu bắt buộc phải được triển khai. Cụ thể về Công ty TNHH Niềm Tin nơi tôi vừa công tác thực tập tại đây là một doanh nghiệp trong ngành sáng tạo và phát triển phần mềm chuyển đổi số đã tạo điều kiện giúp tôi tiếp cận với cửa hàng thuê xe Việt Key đang gặp những bất cập khi không áp dụng công nghệ trong quá trình quản lý. Chính vì lý do đó đề tài “phát triển website cho thuê xe Việt Key” được ra đời. Hướng tới những doanh nghiệp cửa hàng cho thuê xe hoặc sản phẩm có quy mô nhỏ - vừa muốn áp dụng công nghệ với chi phí thấp

đơn giản dễ sử dụng. Hy vọng sẽ đem giúp cho quá trình quản lý được chặt chẽ, nâng cao hiệu suất.

CHƯƠNG 1 TỔNG QUAN

1.1 Lý do chọn đề tài

Trong thời đại công nghệ hiện nay, việc sử dụng các dịch vụ trực tuyến trở nên phổ biến hơn bao giờ hết. Điều này đặc biệt đúng trong lĩnh vực vận tải và du lịch. Trong khi việc thuê xe máy và đỗ xe ngày càng trở nên phổ biến, nhiều doanh nghiệp trong ngành vẫn chưa tận dụng hết tiềm năng của mạng internet để quảng bá và cung cấp dịch vụ của mình trực tuyến. Do đó, tôi quyết định chọn đề tài thực hiện website quản lý cho thuê xe máy và chỗ đỗ xe nhằm tận dụng cơ hội này.

Thực hiện website quản lý cho thuê xe máy và chỗ đỗ xe có nhiều ưu điểm. Trước hết, việc thực hiện website giúp cho các doanh nghiệp có thể quảng bá dịch vụ của mình đến một lượng khách hàng tiềm năng rộng lớn, không chỉ giới hạn ở phạm vi địa lý. Với sự phổ biến của internet, mọi người có thể truy cập website của các doanh nghiệp từ bất kỳ đâu và bất kỳ lúc nào, giúp cho doanh nghiệp tiếp cận được với nhiều khách hàng hơn.

Thứ hai, việc thực hiện website giúp cho việc quản lý dịch vụ thuê xe máy và đỗ xe trở nên dễ dàng hơn. Các thông tin về xe, giá cả, thời gian thuê và đặt chỗ có thể được hiển thị rõ ràng và cập nhật liên tục trên website, giúp cho khách hàng có thể dễ dàng lựa chọn dịch vụ mà mình mong muốn. Đồng thời, quản lý của doanh nghiệp cũng có thể tiện lợi hơn trong việc quản lý lịch trống, xác nhận đơn đặt hàng và theo dõi các khoản thanh toán.

Thứ ba, việc thực hiện website còn giúp cho doanh nghiệp tiết kiệm chi phí quảng bá và tiếp thị. Truyền thông truyền thống như quảng cáo trên truyền hình, tạp chí, báo... sẽ tốn kém hơn và không hiệu quả bằng các chiến lược tiếp thị trực tuyến. Ngoài ra, việc sử dụng các dịch vụ thuê xe và đỗ xe trực tuyến cũng đang trở nên phổ biến hơn trong thời đại số hóa này, khi mà nhu cầu di chuyển và sử dụng xe cộ ngày càng tăng. Những trang web cung cấp dịch vụ cho thuê xe máy và đỗ xe trực tuyến đã trở thành một phần quan trọng của ngành công nghiệp xe cộ, giúp cho việc sử dụng xe trở nên dễ dàng và thuận tiện hơn đối với khách hàng.

Bên cạnh đó, việc xây dựng một website quản lý cho thuê xe máy và đỗ xe sẽ giúp các doanh nghiệp cung cấp dịch vụ này tiếp cận với khách hàng một cách dễ dàng và thuận tiện hơn. Trang web sẽ cho phép khách hàng có thể dễ dàng tìm kiếm và đặt thuê xe máy hoặc đỗ xe tại các điểm có sẵn trên trang web, giúp tiết kiệm thời gian và nỗ lực cho khách hàng.

Từ những lý do trên, tôi đã chọn đề tài thực hiện một website quản lý cho thuê xe máy và đỗ xe để nghiên cứu và phát triển, nhằm cung cấp một giải pháp hiệu quả cho doanh nghiệp trong ngành công nghiệp xe cộ và đáp ứng nhu cầu ngày càng tăng của khách hàng trong thời đại số hóa.

1.2 Mục tiêu và chức năng của hệ thống

1.2.1 Mục tiêu

Giới thiệu về thông tin doanh nghiệp, các sản phẩm và dịch vụ mà doanh nghiệp có. Quản lý quá trình thuê xe máy và chỗ đỗ xe, các mục tiêu bao gồm:

- Đưa thông tin của doanh nghiệp và sản phẩm đến gần hơn với khách hàng.
- Số hóa việc lưu trữ thông tin về sản phẩm và nhân lực.
- Tìm kiếm, trích xuất dữ liệu nhanh chóng và chính xác.
- Giảm thiểu việc lưu trữ giấy tờ.
- Theo dõi tình hình kinh doanh.
- ...

Trong môi trường doanh nghiệp cửa hàng, công nghệ thông tin không chỉ thay đổi cách thức con người thực hiện công việc quản lý mà còn thay đổi cả cách thức họ thông tin cho nhau, xử lý thông tin và ra quyết định.

1.2.2 Chức năng

- ✓ Giới thiệu về cửa hàng cho thuê xe máy Việt Key
- ✓ Quản trị website
- ✓ Quản lý xe máy

- ✓ Quản lý chỗ đỗ
- ✓ Quản lý nhân viên
- ✓ Quản lý hóa đơn
- ✓ Quản lý phiếu thuê
- ✓ Quản lý đánh giá
- ✓ Quản lý chi nhánh
- ✓ Quản lý bài viết
- ✓ Quản lý tài khoản người dùng

1.3 Mô tả bài toán

- Mục đích của website cho thuê xe máy và chỗ đỗ xe là đưa thông tin xe của cửa hàng đến gần hơn với khách hàng.
 - ✓ Quản lý xe máy:
 - Quản lý danh mục xe máy: thêm, sửa, xóa, tìm kiếm
 - Quản lý thông tin xe máy: mã, tên, mô tả, giá, hình ảnh, ...
 - ✓ Quản lý chỗ đỗ xe:
 - Quản lý danh mục chỗ đỗ: thêm, sửa, xóa, chọn vị trí
 - Quản lý thông tin địa điểm đỗ: mã, vị trí, giá, ...
- Mục đích của website quản lý cho thuê là bao gồm quản lý thông tin phiếu thuê, thông tin nhân viên, quản lý bình luận, quản lý hóa đơn, quản lý bài viết và lưu trữ dữ liệu từ người dùng. Chi tiết các chức năng chính của hệ thống như sau:
 - ✓ Quản lý tài khoản người dùng:
 - Quản lý danh mục người dùng: thêm, sửa, xóa, phân quyền
 - Quản lý thông tin thiết bị: mã, tên, số địa thoại, email, quyền truy cập
 - ✓ Quản lý nhân viên:
 - Quản lý thông tin nhân viên: mã, tên, địa chỉ, số điện thoại, lương, ngày công, chi nhánh.
 - Quản lý danh mục nhân viên: thêm, sửa, xóa
 - Quản lý bảng lương.

- ✓ Quản lý phiếu thuê:
 - Quản lý phiếu thuê: mã phiếu, tên xe, chi nhánh, loại xe, phương thức thuê, địa chỉ thuê, người thuê, ngày thuê, ngày trả, ...
 - Quản lý danh mục phiếu thuê: thêm, sửa, xóa
- ✓ Quản lý hóa đơn:
 - Quản lý hóa đơn: thêm, sửa, xóa
 - Quản lý thông tin hóa đơn: mã, ngày trả xe thực tế, tổng tiền, tiền phạt,....
- ✓ Quản lý bình luận:
 - Quản lý bình luận: thêm, sửa, xóa bình luận
- ✓ Quản lý bài viết:
 - Quản lý bài viết: thêm, sửa, xóa bài viết

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu ngôn ngữ Python

2.1.1 Tổng quát



Hình 2.1 Biểu tượng của Python

Python là một ngôn ngữ lập trình bậc cao được phát triển cho các mục đích lập trình đa năng. Ngôn ngữ này được tạo ra bởi Guido van Rossum và được ra mắt lần đầu vào năm 1991. Trải qua hơn 30 năm phát triển và cải tiến, Python đã trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới với đóng góp lớn từ cộng đồng lập trình viên và các nhà khoa học máy tính.

Một trong những ưu điểm mạnh của Python là tính đơn giản, dễ đọc, dễ học và dễ nhớ. Python có cú pháp đơn giản, cấu trúc rõ ràng và dễ hiểu, thuận tiện cho người mới học lập trình. Với cách viết mã ngắn gọn, cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, tiết kiệm thời gian và công sức cho người lập trình.

Python cũng rất linh hoạt và dễ dàng tích hợp với các ngôn ngữ lập trình khác, giúp cho việc phát triển và tương tác với các ứng dụng khác trở nên dễ dàng hơn. Python được sử dụng rộng rãi trong nhiều lĩnh vực, từ khoa học dữ liệu, trí tuệ nhân tạo, đến web development và các ứng dụng mobile.

Trong quá trình sử dụng, người dùng Python còn được hưởng lợi từ tính động, tự động quản lý bộ nhớ và các thư viện phong phú được cung cấp sẵn. Tính động của Python có nghĩa là người dùng có thể khai báo biến mà không cần phải khai báo kiểu dữ liệu trước, giúp cho việc lập trình trở nên linh hoạt và tiện lợi hơn. Đồng thời, việc quản lý bộ nhớ được tự động hóa giúp người lập trình không cần phải lo lắng về việc giải phóng bộ nhớ,

tăng hiệu suất và độ ổn định cho ứng dụng. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

2.1.2 Lịch sử phát triển của Python

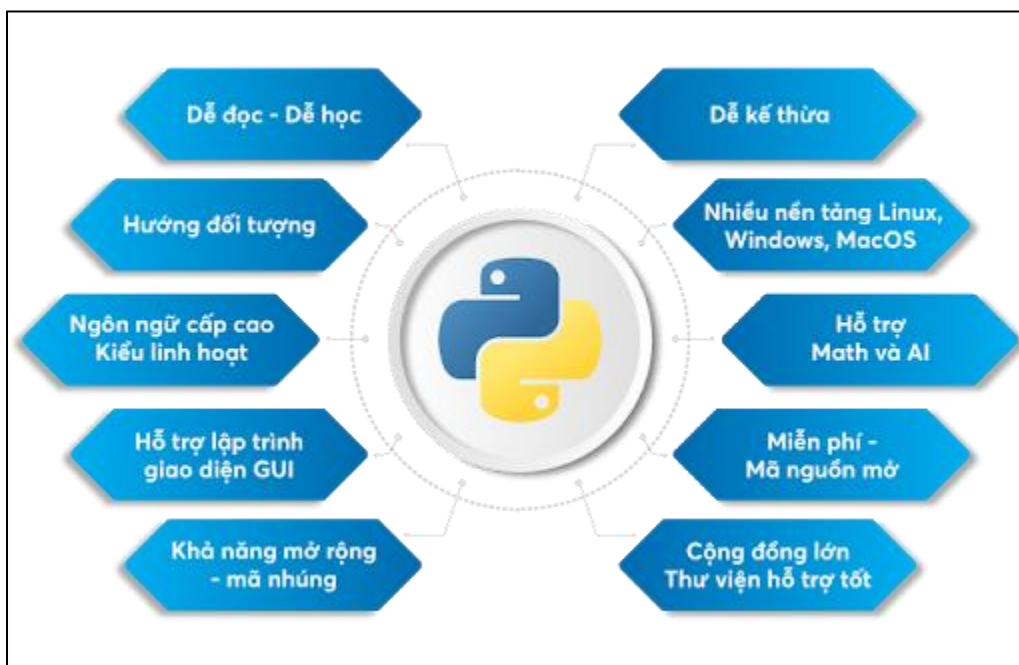
Sự phát triển Python đến nay có thể chia làm các giai đoạn:

- Python 1: bao gồm các bản phát hành 1.x. Giai đoạn này, kéo dài từ đầu đến cuối thập niên 1990. Từ năm 1990 đến 1995, Guido làm việc tại CWI (Centrum voor Wiskunde en Informatica – Trung tâm Toán-Tin học) tại Amsterdam, Hà Lan. Vì nguyên nhân này, các phiên bản Python đầu tiên đều do CWI phát hành. Phiên bản cuối cùng phát hành tại CWI là 1.2.
 - Vào năm 1995, Guido chuyển sang Corporation for National Research Initiatives (CNRI) ở Reston, Virginia. Tại đây, ông phát hành một số phiên bản khác. Python 1.6 là phiên bản cuối cùng phát hành tại CNRI.
 - Sau bản phát hành 1.6, Guido rời bỏ CNRI để làm việc với các lập trình viên chuyên viết phần mềm thương mại. Tại đây, ông có ý tưởng sử dụng Python với các phần mềm tuân theo chuẩn GPL. Sau đó, CNRI và Free Software Foundation (FSF) đã cùng nhau hợp tác để làm bản quyền Python phù hợp với GPL. Cùng năm đó, Guido được nhận giải thưởng FSF Award for the Advancement of Free Software.
 - Phiên bản 1.6.1 ra đời sau đó là phiên bản đầu tiên tuân theo bản quyền GPL. Tuy nhiên, bản này hoàn toàn giống bản 1.6, trừ một số sửa lỗi cần thiết.
- Python 2: vào năm 2000, Guido và nhóm phát triển Python dời đến BeOpen.com và thành lập BeOpen PythonLabs team. Phiên bản Python 2.0 được phát hành tại đây. Sau khi phát hành Python 2.0, Guido và các thành viên PythonLabs gia nhập Digital Creations.
 - Python 2.1 ra đời kế thừa từ Python 1.6.1 và Python 2.0. Bản quyền của phiên bản này được đổi thành Python Software Foundation License. Từ thời điểm này

trở đi, Python thuộc sở hữu của Python Software Foundation (PSF), một tổ chức phi lợi nhuận được thành lập theo mẫu Apache Software Foundation.

- Python 3: Về sự phát triển của Python trong tương lai, các nhà phát triển vẫn đang thảo luận về phiên bản mới: Python 3.0 (dự án gọi là Python 3000 hoặc Py3K). Dự kiến, dòng 3.x sẽ không hoàn toàn tương thích với dòng 2.x. Nhóm phát triển cho biết sẽ lấp những chỗ hờ của ngôn ngữ. Nguyên tắc chủ đạo để phát triển Python 3.x là “bỏ cách làm việc cũ nhằm hạn chế trùng lặp về mặt chức năng của Python”. Tuy nhiên, cho đến nay vẫn chưa có kế hoạch cụ thể cho phát triển Python 3.

2.1.3 Đặc điểm của ngôn ngữ Python



Hình 2.2 Đặc điểm của ngôn ngữ Python

- **Ngôn ngữ lập trình đơn giản, dễ học – dễ học**

Python có cú pháp rất đơn giản, rõ ràng. Nó dễ đọc và viết hơn rất nhiều khi so sánh với những ngôn ngữ lập trình khác như C++, Java, C#. Python làm cho việc lập trình trở nên thú vị, cho phép bạn tập trung vào những giải pháp chứ không phải cú pháp.

- **Miễn phí, mã nguồn mở**

Người dùng có thể tự do sử dụng và phân phối Python, thậm chí là dùng cho mục đích thương mại. Vì là mã nguồn mở, bạn không những có thể sử dụng các phần mềm, chương trình được viết trong Python mà còn có thể thay đổi mã nguồn của nó. Python có một cộng đồng rộng lớn, không ngừng cải thiện nó mỗi lần cập nhật.

➤ **Khả năng di động linh hoạt**

Nếu đã viết mã Python cho máy Windows trước đó. Bây giờ, nếu người dùng muốn chạy nó trên máy Mac, thì không cần phải thay đổi nó như cũ. Nói cách khác, vẫn có thể lấy một mã và chạy nó trên bất kỳ máy nào, không cần phải viết mã khác nhau cho các máy khác nhau. Điều này làm cho Python trở thành một ngôn ngữ di động. Tuy nhiên, người dùng phải tránh mọi tính năng phụ thuộc hệ thống trong trường hợp này. Nó chạy liền mạch trên hầu hết tất cả các nền tảng như Windows, macOS, Linux.

➤ **Khả năng mở rộng và có thể nhúng**

Một ứng dụng đòi hỏi sự phức tạp rất lớn, người dùng có thể dễ dàng kết hợp các phần code bằng C, C++ và những ngôn ngữ khác (có thể gọi được từ C) vào code Python. Điều này sẽ cung cấp cho ứng dụng của người dùng những tính năng tốt hơn cũng như khả năng scripting mà những ngôn ngữ lập trình khác khó có thể làm được.

➤ **Ngôn ngữ thông dịch cấp cao**

Không giống như C/C++, với Python những nhiệm vụ khó khăn như quản lý bộ nhớ, dọn dẹp những dữ liệu vô nghĩa,... Khi chạy code Python sẽ tự động chuyển đổi code sang ngôn ngữ máy tính có thể hiểu. Trong nội bộ, mã nguồn của nó được chuyển đổi thành một hình thức ngay lập tức được gọi là bytecode. Vì vậy, tất cả những gì người dùng cần làm là chạy đoạn code Python mà không phải lo lắng về việc liên kết với các thư viện và những thứ khác.

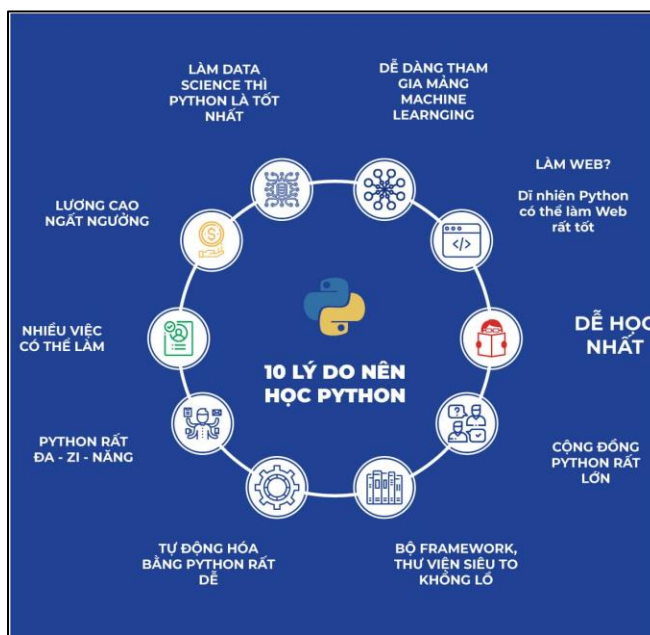
➤ **Thư viện tiêu chuẩn lớn để giải quyết những tác vụ phổ biến**

Python có một số lượng lớn thư viện tiêu chuẩn giúp cho công việc lập trình của người dùng trở nên dễ thở hơn rất nhiều, đơn giản vì không phải tự viết tất cả code. Ví dụ: Người dùng cần kết nối cơ sở dữ liệu MySQL trên Web server? Người dùng có thể nhập thư viện MySQLdb và sử dụng nó. Có các thư viện cho các biểu thức thông thường, tạo tài liệu, kiểm tra đơn vị, trình duyệt web, phân luồng, cơ sở dữ liệu, CGI, email, thao tác hình ảnh và rất nhiều chức năng khác. Vì vậy, người dùng có thể chắc chắn rằng nó sẽ không làm hỏng code hay ứng dụng của mình.

➤ Hướng đối tượng

Mọi thứ trong Python đều là hướng đối tượng. Lập trình hướng đối tượng (OOP) giúp giải quyết những vấn đề phức tạp một cách trực quan. Với OOP, người dùng có thể phân chia những vấn đề phức tạp thành những tập nhỏ hơn bằng cách tạo ra các đối tượng. Python hỗ trợ cả lập trình hướng đối tượng, một trong những tính năng chính của nó. Nó cũng hỗ trợ nhiều kế thừa, không giống như Java.

2.1.4 Tại sao nên chọn Python



Hình 2.3 Tại sao nên chọn Python

Tại sao chọn Python thứ nhất Python có thể làm Web rất tốt. Python là một ngôn ngữ lập trình mạnh mẽ và linh Python là một ngôn ngữ lập trình rất phổ biến và đang được ưa chuộng trong cộng đồng lập trình hiện nay với nhiều lợi ích quan trọng.

Đầu tiên, Python là một ngôn ngữ lập trình rất dễ học, đơn giản và rất dễ hiểu. Bạn có thể học Python chỉ trong vài tuần và bắt đầu làm việc với nó ngay lập tức. Điều này làm cho nó trở thành lựa chọn lý tưởng cho những người mới bắt đầu học lập trình.

Thứ hai, Python là một ngôn ngữ đa năng và có thể được sử dụng cho nhiều mục đích khác nhau, bao gồm làm việc với dữ liệu và thống kê, machine learning, web development, tự động hóa, và nhiều lĩnh vực khác. Điều này cho phép các lập trình viên có thể tìm được công việc với Python ở nhiều lĩnh vực khác nhau và mở ra nhiều cơ hội việc làm.

Thứ ba, Python đang trở thành ngôn ngữ lập trình phổ biến nhất trong lĩnh vực data science, machine learning và AI. Với các thư viện và framework như NumPy, Pandas, Scikit-learn, TensorFlow và Keras, Python cung cấp cho các nhà khoa học dữ liệu và kỹ sư machine learning các công cụ và thư viện cần thiết để xử lý và phân tích dữ liệu.

Thứ tư, Python là một ngôn ngữ rất thân thiện với các nhà phát triển web. Nó cung cấp cho người dùng nhiều framework và thư viện như Django, Flask, Pyramid, và nhiều hơn nữa, giúp cho việc phát triển web trở nên dễ dàng hơn bao giờ hết.

Thứ năm, Python có một cộng đồng lập trình viên rất lớn và đầy đủ hỗ trợ. Người dùng Python có thể tìm thấy nhiều tài liệu, hướng dẫn và các giải pháp được chia sẻ từ các thành viên khác trong cộng đồng, giúp đỡ khi gặp vấn đề. Bạn có thể tham gia các diễn đàn, nhóm người dùng và trang web chia sẻ kiến thức để trao đổi, hỏi đáp và học hỏi từ những người có kinh nghiệm trong lĩnh vực này.

Thứ sáu, Python có một bộ framework và thư viện siêu khổng lồ. Các framework như Django, Flask, và Pyramid cung cấp cơ sở hạ tầng cho việc phát triển ứng dụng web, trong khi các thư viện như NumPy, SciPy, Matplotlib, và Pandas giúp đơn giản hóa và tăng tốc quá trình xử lý dữ liệu và tính toán.

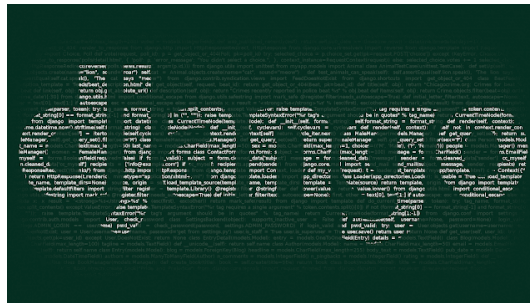
Với các lý do trên đã cho ta thấy được sự mạnh mẽ cũng như tính linh hoạt của Python vì vậy lựa chọn Python để thực hiện là một quyết định tốt.

2.1.5 Ứng dụng ngôn ngữ Python

- Phát triển ứng dụng website
- Lập trình ứng dụng
- Lập trình ứng dụng khoa học, vật lý, vi sinh
- Khoa học máy tính
- Phát triển game với PyGame
- Lĩnh vực IoT- Internet Vạn Vật
- Machine Learning
- Dự án Blockchain

2.2 Giới thiệu Django-Python

2.2.1 Tổng quát



Hình 2.4 Biểu tượng của Django framework

Django là một web framework phổ biến được viết bằng ngôn ngữ lập trình Python. Django được thiết kế để giúp lập trình viên xây dựng các ứng dụng web phức tạp một cách nhanh chóng và dễ dàng, đồng thời cung cấp các tính năng bảo mật, khả năng mở rộng và tương thích với nhiều cơ sở dữ liệu khác nhau.

Django có thể sử dụng các template engine khác nhau để hiển thị dữ liệu, trong đó Jinja là một trong những template engine được sử dụng phổ biến. Jinja là một engine mẫu

phát triển bởi Armin Ronacher, được viết bằng Python và cung cấp các tính năng mạnh mẽ như các biểu thức điều kiện, vòng lặp và kế thừa.

Ngoài ra, Django cũng cung cấp khả năng tương tác với nhiều loại cơ sở dữ liệu khác nhau, bao gồm PostgreSQL, MySQL, SQLite và Oracle. Điều này giúp cho Django có thể phát triển các ứng dụng web phức tạp và xử lý lượng dữ liệu lớn một cách hiệu quả.

Django cũng cung cấp các tính năng bảo mật để bảo vệ các ứng dụng web khỏi các cuộc tấn công bảo mật. Các tính năng bảo mật này bao gồm xác thực người dùng, quản lý phiên và hệ thống phân quyền.

Để tóm lại, Django là một web framework mạnh mẽ và linh hoạt, được viết bằng Python và cung cấp nhiều tính năng hữu ích để giúp lập trình viên xây dựng các ứng dụng web phức tạp một cách dễ dàng và hiệu quả. Jinja là một trong những template engine được sử dụng phổ biến trong Django, còn Django có khả năng tương tác với nhiều loại cơ sở dữ liệu khác nhau và cung cấp các tính năng bảo mật để bảo vệ các ứng dụng web khỏi các cuộc tấn công bảo mật.

2.3 Giới thiệu Jinja2

2.3.1 Tổng quát



Hình 2.2.5 Biểu tượng của Jinja2

Jinja2 là một thư viện mẫu (template) Python được sử dụng để tạo các trang web động. Nó cho phép lập trình viên tách biệt code logic và giao diện người dùng, giúp tăng tính linh hoạt và bảo trì trong quá trình phát triển.

Jinja2 cung cấp cú pháp đơn giản và rõ ràng, dễ dàng cho người mới học lập trình và cũng cho phép các lập trình viên có kinh nghiệm tùy chỉnh và mở rộng. Jinja2 cung cấp

một loạt các tính năng bao gồm các biểu thức, câu lệnh điều kiện, vòng lặp và biến, cho phép lập trình viên tạo các mẫu tùy chỉnh với sự linh hoạt cao.

2.3.2 Sự ra đời

Jinja2 là một hệ thống mẫu (template system) được viết bằng Python và được phát triển bởi Armin Ronacher. Jinja2 được phát hành lần đầu tiên vào năm 2008, và được sử dụng rộng rãi trong các ứng dụng web hiện đại.

Sự ra đời của Jinja2 đến từ nhu cầu tạo ra các trang web động dựa trên nội dung động. Trước khi Jinja2 được phát triển, các nhà phát triển thường sử dụng các thư viện mẫu khác nhau hoặc viết mã HTML bằng tay. Tuy nhiên, các phương pháp này thường gặp phải các vấn đề về khả năng tái sử dụng mã, bảo trì và khả năng mở rộng.

2.3.3 Tính ưu việt của Jinja2

Jinja2 đã giải quyết các vấn đề này bằng cách cung cấp cho người lập trình một cách tiếp cận dễ dàng hơn để tạo ra các mẫu HTML động. Jinja2 cho phép các nhà phát triển viết các mẫu HTML với các biểu thức lồng nhau và các rẽ nhánh điều kiện. Các biểu thức này có thể được kết hợp với các dữ liệu động để tạo ra các trang web động theo cách dễ dàng và hiệu quả.

Jinja2 cũng được thiết kế để tương thích với các ứng dụng web khác nhau, bao gồm các ứng dụng web dựa trên các framework phổ biến như Flask và Django. Do đó, Jinja2 đã trở thành một công cụ được sử dụng rộng rãi trong cộng đồng lập trình viên Python để xây dựng các ứng dụng web động.

2.4 Giới thiệu về phpMyAdmin và MySQL

2.4.1 Giới thiệu về phpMyAdmin



Hình 2.2.6 Biểu tượng của phpMyAdmin

phpMyAdmin là một công cụ quản lý cơ sở dữ liệu MySQL được viết bằng PHP. Nó cho phép người dùng quản lý cơ sở dữ liệu MySQL thông qua giao diện đồ họa trên web, thay vì sử dụng các câu lệnh SQL thông thường. Với phpMyAdmin, người dùng có thể thực hiện các tác vụ như tạo cơ sở dữ liệu, bảng, truy vấn, chèn dữ liệu và chỉnh sửa các bản ghi trong cơ sở dữ liệu.

phpMyAdmin được phát triển như một phần của dự án mã nguồn mở và được cung cấp miễn phí cho người dùng. Nó được thiết kế để dễ sử dụng và cài đặt, đặc biệt là cho các người dùng không có kinh nghiệm về MySQL hoặc lập trình web. Ngoài ra, phpMyAdmin cũng hỗ trợ nhiều ngôn ngữ và cung cấp một loạt các tính năng hữu ích cho quản trị cơ sở dữ liệu, như xuất và nhập dữ liệu, sao lưu và phục hồi cơ sở dữ liệu.

2.4.2 Giới thiệu về MySQL

MySQL là hệ thống quản trị cơ sở dữ liệu quan hệ mã nguồn mở được phát triển bởi một công ty đặt tại Thụy Điển là MySQL AB. MySQL được phát hành lần đầu vào năm 1995 và đã nhanh chóng trở thành một trong những hệ thống quản trị cơ sở dữ liệu phổ biến nhất trên thế giới.

Một trong những lý do chính để MySQL trở thành một trong những hệ thống quản trị cơ sở dữ liệu phổ biến nhất là vì nó được phát triển theo hướng mã nguồn mở. Do đó, người dùng có thể tải về và sử dụng MySQL miễn phí và có thể tùy chỉnh mã nguồn theo nhu cầu của mình.

Một ưu điểm khác của MySQL là tính ổn định và hiệu suất của nó. MySQL có thể xử lý hàng trăm ngàn truy vấn cùng lúc và hỗ trợ nhiều loại kết nối đồng thời, bao gồm kết nối TCP/IP, Unix socket và named pipes. Nó cũng được tối ưu hóa để hoạt động tốt trên các hệ thống phân tán và có khả năng mở rộng tốt để xử lý dữ liệu lớn.

Ngoài ra, MySQL còn cung cấp nhiều tính năng mạnh mẽ cho người dùng, bao gồm quản lý cơ sở dữ liệu, tạo, truy vấn, cập nhật và xóa dữ liệu, bảo mật cơ sở dữ liệu và quản lý người dùng, hỗ trợ các ngôn ngữ lập trình khác nhau và nhiều tính năng khác.

2.5 Một số thư viện khác

- ✓ Bootstrap-4,5: Bootstrap-4,5 là một thư viện CSS và JavaScript phổ biến được sử dụng để tạo giao diện động và phản hồi cho trang web. Nó cung cấp một bộ công cụ UI đa dạng để tạo các thành phần trên trang web, bao gồm nút, biểu mẫu, bảng, thanh trượt, menu thả xuống và nhiều hơn nữa
- ✓ Font-awesome: Font-awesome là một thư viện biểu tượng vector phổ biến, cung cấp cho người dùng các biểu tượng đẹp và dễ sử dụng để trang trí cho giao diện của trang web.
- ✓ HTML 5: HTML 5 là phiên bản mới nhất của ngôn ngữ đánh dấu HTML, được sử dụng để tạo các trang web động và phản hồi. Nó hỗ trợ nhiều tính năng mới, bao gồm hình ảnh, âm thanh, video, giải pháp lưu trữ cục bộ và nhiều hơn nữa.
- ✓ Chart.js: Chart.js là một thư viện JavaScript phổ biến được sử dụng để tạo các biểu đồ và đồ thị động trên trang web. Nó cung cấp một cách dễ dàng để tạo các biểu đồ phức tạp như biểu đồ vòng, biểu đồ cột, biểu đồ đường và nhiều hơn nữa.
- ✓ Thư viện khác: Canvas, Pisa, Pillow,.....

CHƯƠNG 3 PHÂN TÍCH HỆ THỐNG

3.1 Mô tả hệ thống

3.1.1 Website giới thiệu

- Thông tin của hàng thuê xe Việt Key. Giới thiệu địa chỉ, số điện thoại liên hệ.
- Giới thiệu tất cả các sản phẩm, sản phẩm theo danh mục loại.
- Tìm kiếm sản phẩm theo tên sản phẩm.

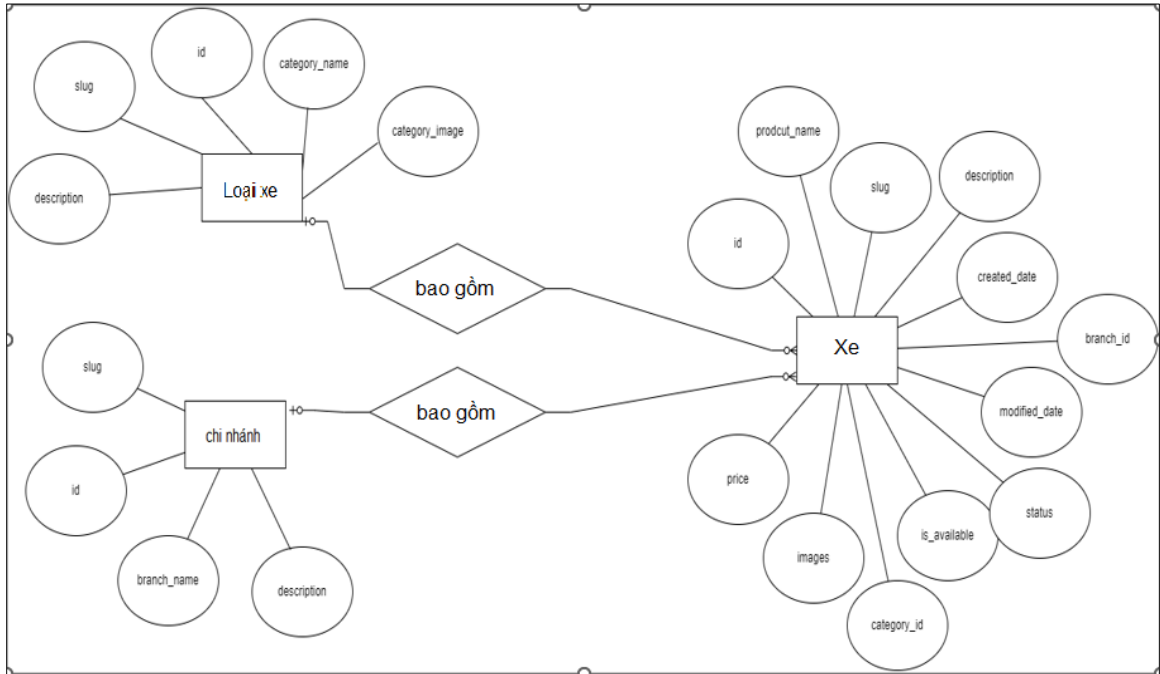
3.1.2 Quản trị website

- Quản lý thông tin khách hàng: Bao gồm thông tin cá nhân, lịch sử thuê xe và đặt chỗ, thông tin liên hệ, thanh toán, v.v.
- Quản lý thông tin xe và bãi đỗ xe: Bao gồm thông tin về giá thuê, vị trí đỗ xe, tình trạng sẵn có, v.v.
- Quản lý phiếu mượn và thanh toán: Đảm bảo việc thanh toán tiện lợi và an toàn cho khách hàng, giám sát sản phẩm, phiếu mượn và xử lý các vấn đề liên quan đến thanh toán.
- Tối ưu hóa trải nghiệm người dùng: Cải thiện trải nghiệm người dùng của khách hàng khi sử dụng website, đảm bảo giao diện thân thiện và dễ sử dụng.
- Đảm bảo an toàn và bảo mật thông tin: Đảm bảo thông tin của khách hàng được bảo mật, tránh tình trạng lộ thông tin cá nhân hay thanh toán của khách hàng.
- Định kỳ cập nhật và bảo trì website: Cập nhật website thường xuyên để đảm bảo hoạt động mượt mà và ổn định, đồng thời thực hiện các công việc bảo trì để giảm thiểu tình trạng sự cố và đảm bảo sự liên tục của dịch vụ.

3.1.3 Quản lý xe máy

- Quản lý danh mục xe máy. Quản lý xe máy bao gồm tên xe, đường dẫn xe, mô tả xe, giá thuê xe theo ngày, hình ảnh của xe, trạng thái xe có được sẵn sàng để mượn hay không, thể loại của xe, chi nhánh xe, ngày thêm xe vào hệ thống, ngày sửa đổi xe

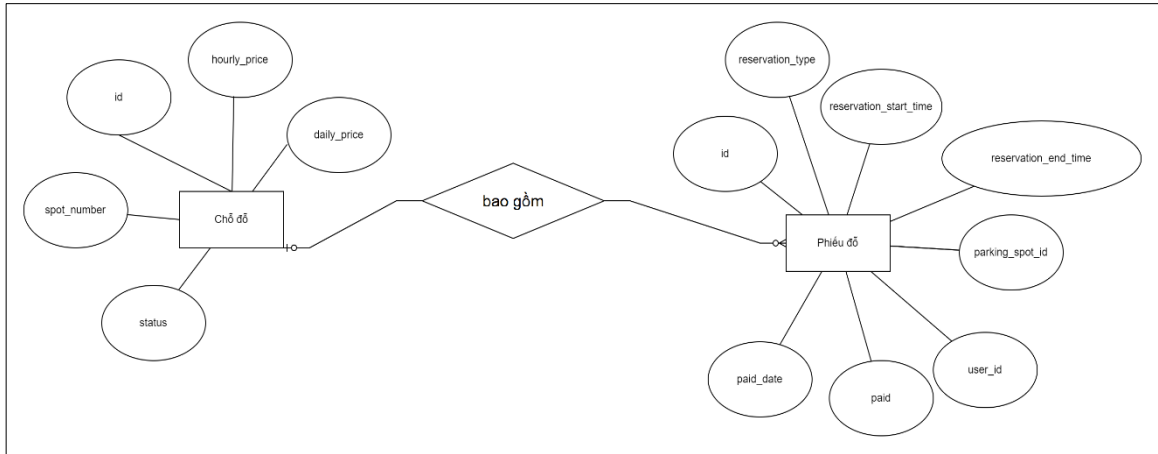
- Các xe máy của cửa hàng được quản lý theo danh mục loại loại xe và danh mục chi nhánh. Mỗi xe máy chỉ được thuộc một loại xe và một chi nhánh. Sẽ có loại xe trong đó chứa nhiều xe cùng loại, sẽ có chi nhánh chứa nhiều xe.



Hình 3.1 Mô hình ER quản lý xe

3.1.4 Quản lý chỗ đỗ

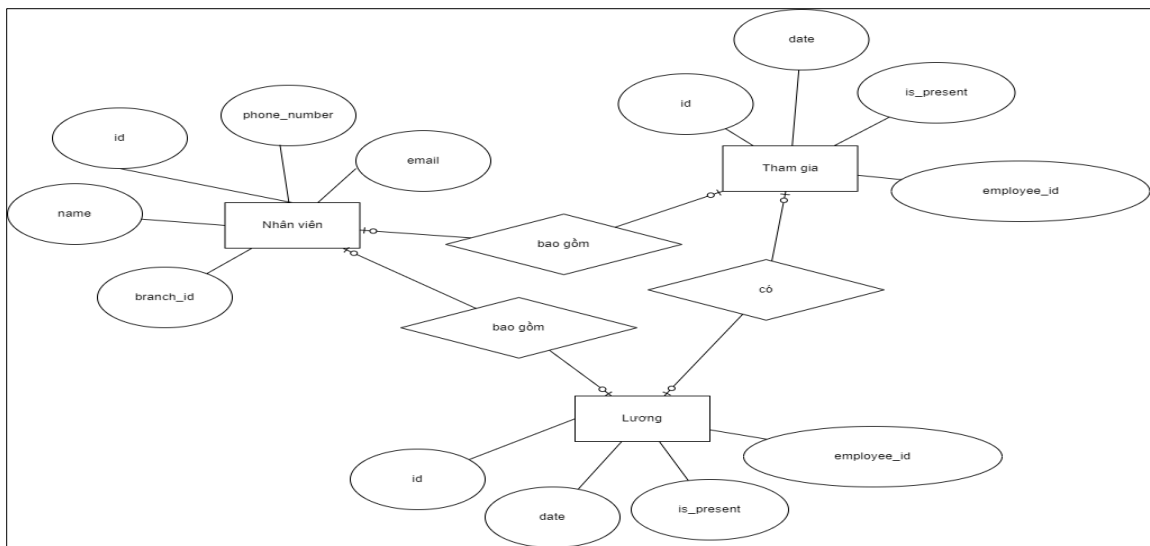
- Quản lý thông tin chỗ đỗ xe: mã, mô tả chỗ đỗ, trạng thái, giá thuê chỗ đỗ theo giờ, giá thuê chỗ đỗ theo ngày, vị trí của chỗ đỗ.
- Mỗi chỗ đỗ sẽ được hiển thị theo danh mục chỗ đỗ và được thuê chỗ đỗ dựa trên tài khoản đăng nhập và vị trí với mỗi chỗ đỗ xe chỉ đi chung với một phiếu đỗ cùng một thời điểm.



Hình 3.2 Mô hình ER quản lý chỗ đỗ

3.1.5 Quản lý nhân viên

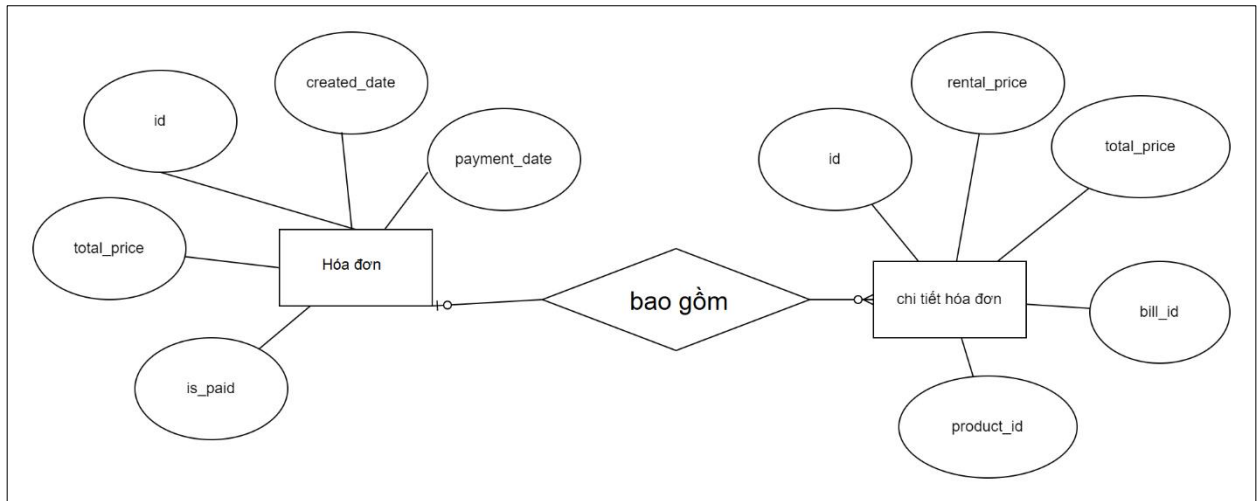
- Quản lý nhân viên theo mã, tên, số điện thoại, địa chỉ, email, chi nhánh.
- Mọi nhân viên đi làm thì đều được tính lương theo ngày công và có xuất bảng lương và trả tiền lương dựa trên số ngày công đi làm. Vì vậy mỗi nhân viên sẽ có được chấm công theo ngày làm và bảng lương.



Hình 3.3 Mô hình ER quản lý nhân viên

3.1.6 Quản lý hóa đơn

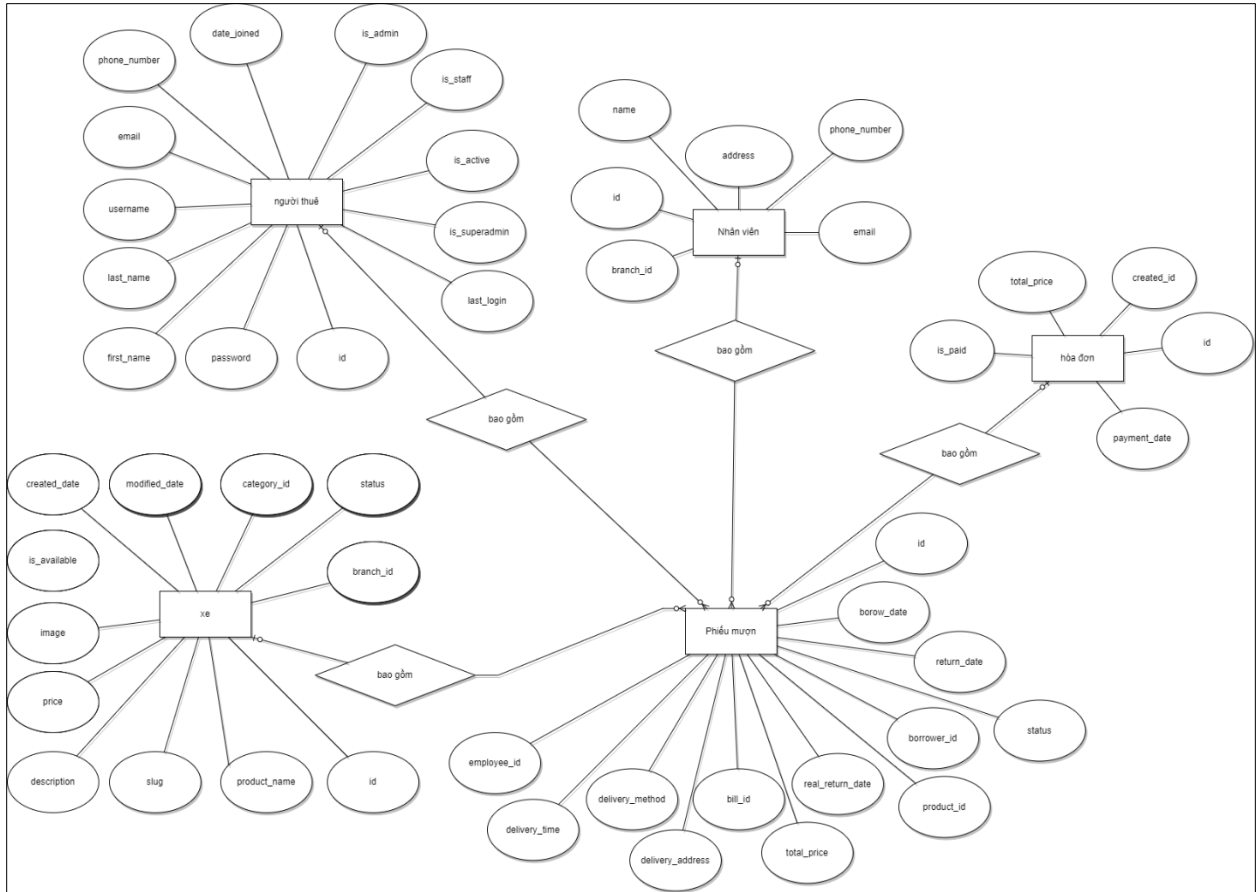
- Quản lý phiếu báo giá thông tin hóa đơn và sản phẩm trên hóa đơn cũng như tổng tiền phải trả và cho biết trạng thái hóa đơn đã được thanh toán hay chưa.
- Quản lý hóa đơn thông tin hóa đơn.



Hình 3.4 Mô hình ER quản lý Hóa đơn

3.1.7 Quản lý phiếu thuê

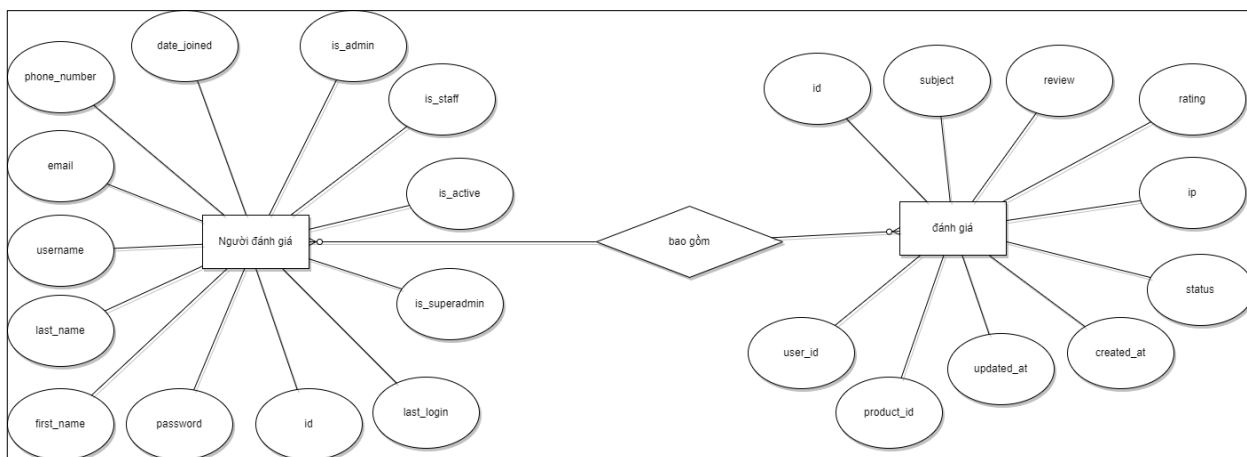
- Quản lý phiếu thuê sẽ bao gồm quản lý tất cả các phiếu thuê mà khách hàng đã sinh ra trong quá trình thuê xe. Cần quản lý mã phiếu thuê, ngày thuê, ngày trả, trạng thái của phiếu thuê, mã của người thuê, mã sản phẩm thuê, ngày trả thực tế, tổng tiền của phiếu thuê, hóa đơn cho phiếu thuê đó, địa chỉ cần giao xe thuê tới, phương thức thuê, thời gian giao xe, mã người giao xe.
- Mỗi một phiếu thuê được hình thành khi khách hàng muốn thuê xe và sinh ra phiếu thuê trong phiếu thuê buộc có sản phẩm và phải có hóa đơn kèm theo nếu trạng thái phiếu thuê không bị hủy và phải có nhân viên tham gia giao xe nếu trạng thái thuê là giao xe tận nơi.



Hình 3.5 Mô hình ER quản lý phiếu thuê

3.1.8 Quản lý đánh giá

- Quản lý đánh giá là quản lý: mã đánh giá, tiêu đề đánh giá, nội dung đánh giá, sao đánh giá, trạng thái có đánh giá đó có khả dụng hay không, ngày tạo ngày, ngày thay đổi, mã sản phẩm đánh giá, người đánh giá



Hình 3.6 Mô hình ER quản lý đánh giá

3.1.9 Quản lý bài viết

- Quản lý bài viết cần quản lý bao gồm các nội dung, hình ảnh, mã bài viết, tiêu đề bài viết ngày tạo ngày chỉnh sửa.

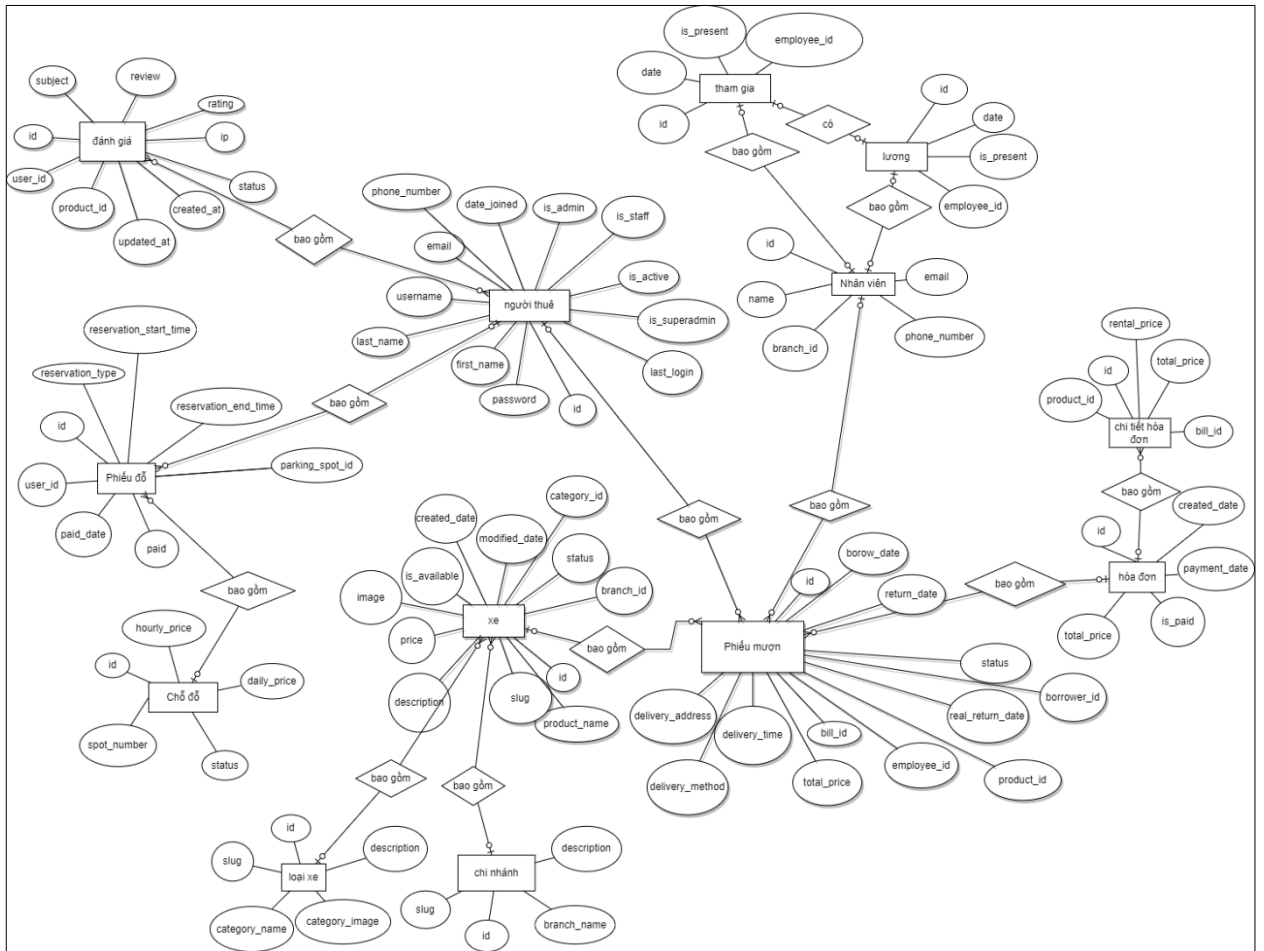
3.1.10 Quản lý chi nhánh

- Quản lý chi nhánh cần quản lý mã chi nhánh, tên chi nhánh, mô tả, đường dẫn chi nhánh

3.1.11 Quản lý tài khoản người dùng

- Quản lý tài khoản người dùng cần quản lý: mã tài khoản, mật khẩu tài khoản, họ và tên tài khoản, email của tài khoản, số điện thoại của tài khoản, ngày tạo tài khoản, lần cuối đăng nhập của tài khoản, phân quyền tài khoản.

3.1.12 Mô Hình ER của toàn bộ hệ thống



Hình 3.7 Mô hình ER toàn hệ thống

3.2 Xây dựng các chức năng của hệ thống

3.2.1 Mô tả các chức năng chi tiết

3.2.1.1 Quản lý xe máy

➤ Quản lý loại xe máy: Cho phép quản lý loại xe máy. Các xe sẽ được quản lý theo danh mục loại xe máy.

Bảng 3-1 Bảng mô tả chức năng quản lý loại xe máy

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới loại xe máy
2	Cập nhật thông tin	Cho phép cập nhật thông tin loại xe máy

3	Xóa	Cho phép xóa loại xe máy trong danh sách
4	Tìm kiếm	Cho phép tìm kiếm loại xe máy trong danh sách

➤ Quản lý sản phẩm: Cho phép quản lý xe máy hiển thị trên website của cửa hàng

Bảng 3-2 Bảng mô tả chức năng quản lý xe máy.

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới xe máy
2	Cập nhật thông tin	Cho phép cập nhật thông tin xe máy
3	Xóa	Cho phép xóa xe máy trong danh sách
4	Tìm kiếm	Cho phép tìm kiếm xe máy trong danh sách

➤ Quản lý ảnh sản phẩm: Cho phép quản lý nhiều ảnh cho một sản phẩm.

Bảng 3-3 Bảng mô tả chức năng quản lý chi nhánh của xe

STT	Chức năng	Diễn giải
1	Thêm mới	Cho phép tạo mới chi nhánh xe máy
2	Cập nhật thông tin	Cho phép cập nhật thông tin chi nhánh xe máy
3	Xóa	Cho phép xóa chi nhánh xe máy trong danh sách
4	Tìm kiếm	Cho phép tìm kiếm chi nhánh xe máy trong danh sách

3.2.1.2 Quản lý chỗ đỗ

➤ Quản lý chỗ đỗ: Cho phép quản lý chỗ đỗ xe. Các vị trí chỗ đỗ sẽ được quản lý theo danh mục chỗ đỗ.

Bảng 3-4 Bảng mô tả chức năng quản lý chỗ đỗ xe

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới chỗ đỗ xe
2	Cập nhật thông tin	Cho phép cập nhật thông tin chỗ đỗ xe
3	Xóa	Cho phép xóa chỗ đỗ xe trong danh sách
4	Tìm kiếm	Cho phép tìm kiếm chỗ đỗ xe theo chi nhánh trong danh sách

➤ Quản lý phiếu đỗ xe: Cho phép quản lý phiếu đỗ xe được lập khi người dùng sử dụng chức năng thuê chỗ đỗ,...

Bảng 3-5 Bảng mô tả chức năng quản lý phiếu đỗ.

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới phiếu đỗ
2	Cập nhật thông tin	Cho phép cập nhật thông tin phiếu đỗ
3	Xóa	Cho phép xóa phiếu đỗ trong danh sách
4	Tìm kiếm	Cho phép tìm kiếm các phiếu đỗ theo tên người đỗ

3.2.1.3 Quản lý nhân viên

- Quản lý nhân viên: Cho phép quản lý thông tin nhân viên lao động trong cửa hàng.

Bảng 3-6 Bảng mô tả chức năng quản lý nhân viên.

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới nhân viên
2	Cập nhật thông tin	Cho phép cập nhật thông tin nhân viên
3	Xóa	Cho phép xóa nhân viên trong danh sách
4	Tìm kiếm	Cho phép tìm kiếm trong danh sách

- Quản lý điểm danh nhân công: Mỗi ngày nhân viên sẽ được điểm danh và số ngày làm việc sẽ được hệ thống ghi nhận phụ vụ cho công tác tính lương.

Bảng 3-7 Bảng mô tả chức năng điểm danh

STT	Chức năng	Diễn giải
1	Điểm danh	Cho phép ghi nhận số ngày làm việc của nhân viên

- Quản lý bảng lương nhân công: Cho phép quản lý bảng lương được tính tự động thông qua lương căn bản, số ngày làm việc.

Bảng 3-8 Bảng mô tả chức năng quản lý bảng lương

STT	Chức năng	Diễn giải
1	Xóa	Cho phép xóa bảng lương trong danh sách
2	Xuất file xlsx	Cho phép xuất bảng lương ra file xlsx
3	Tìm kiếm	Cho phép tìm kiếm bảng lương trong danh sách

3.2.1.4 Quản lý phiếu thuê xe

➤ Quản lý phiếu thuê: sau khi đăng nhập tìm kiếm các xe ưng ý khách hàng sẽ thực hiện thao tác thuê xe. Khi khách hàng thực hiện chức năng thuê xe thì phiếu thuê sẽ được tạo ra và đang ở trạng thái chờ xác nhận, sau khi được chủ cửa hàng xác nhận xuất ra biên bản giao xe và ký thủ tục hoàn tất phiếu thuê sẽ được chuyển sang hoàn thành ngoài ra trong quá trình xác nhận khách hàng được phép hủy phiếu nhưng vẫn phải thông qua chủ cửa hàng.

Bảng 3-9 Bảng mô tả chức năng quản lý phiếu thuê xe

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới phiếu thuê xe
2	Cập nhật thông tin	Cho phép cập nhật thông tin phiếu thuê xe
3	Xóa	Cho phép xóa phiếu thuê xe trong danh sách
4	Xuất biên bản thuê xe	Cho phép xuất biên bản thuê xe
5	Tìm kiếm	Cho phép tìm kiếm phiếu thuê xe trong danh sách
6	Hủy phiếu	Cho phép chuyển trạng thái phiếu thuê sang trạng thái hủy
7	Xem lại lịch sử thuê	Cho phép khách hàng xem lại chính lịch sử thuê xe của mình

3.2.1.5 Quản lý Hóa đơn

➤ Quản lý hóa đơn: Cho phép quản lý hóa đơn khi có phiếu mượn xe. Cho phép xem trước và xuất file Pdf.

Bảng 3-10 Bảng mô tả chức năng quản lý hóa đơn

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới hóa đơn
2	Cập nhật thông tin	Cho phép cập nhật thông tin hóa đơn
3	Xóa	Cho phép xóa hóa đơn trong danh sách
4	Xuất file Pdf	Cho phép xuất hóa đơn ra Pdf
5	Lọc	Cho phép lọc hóa đơn trong danh sách

➤ Quản lý hóa đơn: Cho phép quản lý chi tiết hóa đơn

Bảng 3-11 Bảng mô tả chức năng quản lý chi tiết hóa đơn

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới chi tiết hóa đơn
2	Cập nhật thông tin	Cho phép cập nhật thông tin chi tiết hóa đơn
3	Xóa	Cho phép xóa chi tiết hóa đơn trong danh sách
4	Export Pdf	Cho phép xuất bảng lương ra Pdf
5	Lọc	Cho phép lọc chi tiết hóa đơn trong danh sách

3.2.1.6 Quản lý đánh giá

➤ Quản lý đánh giá: Cho phép quản lý đánh giá xe mà khách hàng đã mượn để có thể ghi lại trải nghiệm đồng thời giúp các khách hàng khác tham khảo chất lượng xe của cửa hàng.

Bảng 3-12 Bảng mô tả chức năng quản lý đánh giá

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới đánh giá
2	Cập nhật thông tin	Cho phép cập nhật thông tin đánh giá
3	Xóa	Cho phép xóa đánh giá trong danh sách

3.2.1.7 Quản lý bài viết

➤ Quản lý bài viết: Cho phép quản lý bài viết về các địa điểm du lịch gần cửa hàng thuê xe mà khách hàng có thể tới tham quan trong quá trình trải nghiệm đồng thời giúp các khách hàng khác tham khảo các địa điểm du lịch nổi tiếng gần cửa hàng.

Bảng 3-13 Bảng mô tả chức năng quản lý bài viết

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới bài viết
2	Cập nhật thông tin	Cho phép cập nhật thông tin bài viết
3	Xóa	Cho phép xóa bài viết trong danh sách

3.2.1.8 Quản lý tài khoản người dùng

➤ Quản lý tài khoản người dùng: Cho phép quản lý tài khoản của người dùng cho phép người dùng đăng nhập vào để thực hiện các chức năng khác của cửa hàng như thuê xe hoặc thuê chỗ đỗ xe, ngoài ra cần phải phân quyền cho tài khoản nếu muốn sử dụng các chức năng cần quyền quản trị.

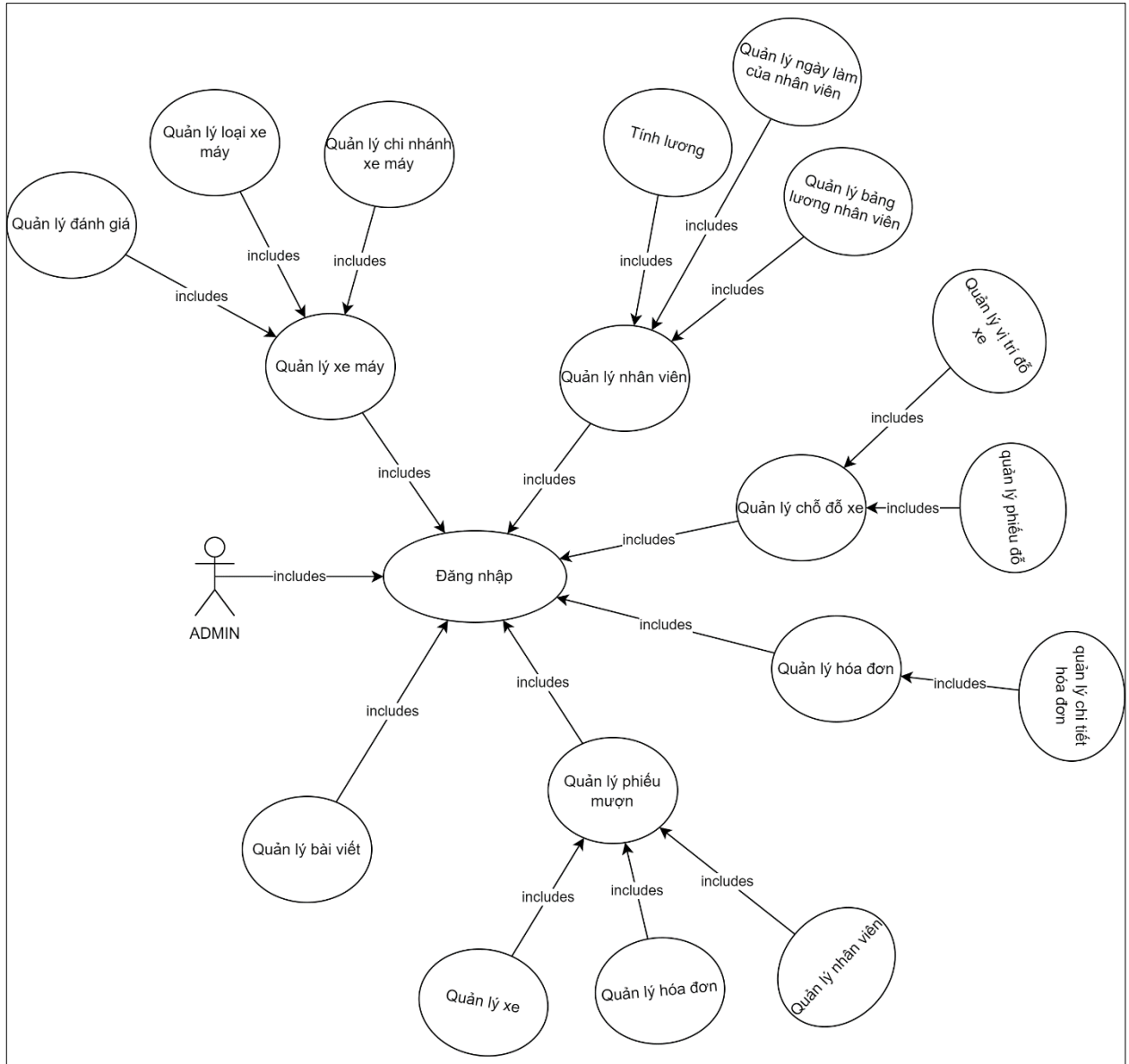
Bảng 3-14 Bảng mô tả chức năng quản lý tài khoản người dùng

STT	Chức năng	Diễn giải
1	Tạo mới	Cho phép tạo mới tài khoản
2	Cập nhật thông tin	Cho phép cập nhật thông tin tài khoản
3	Xóa	Cho phép xóa tài khoản trong danh sách
4	Phân quyền	Cho phép cấp quyền cho tài khoản trong danh sách

3.2.2 Xác định use case

3.2.2.1 ADMIN

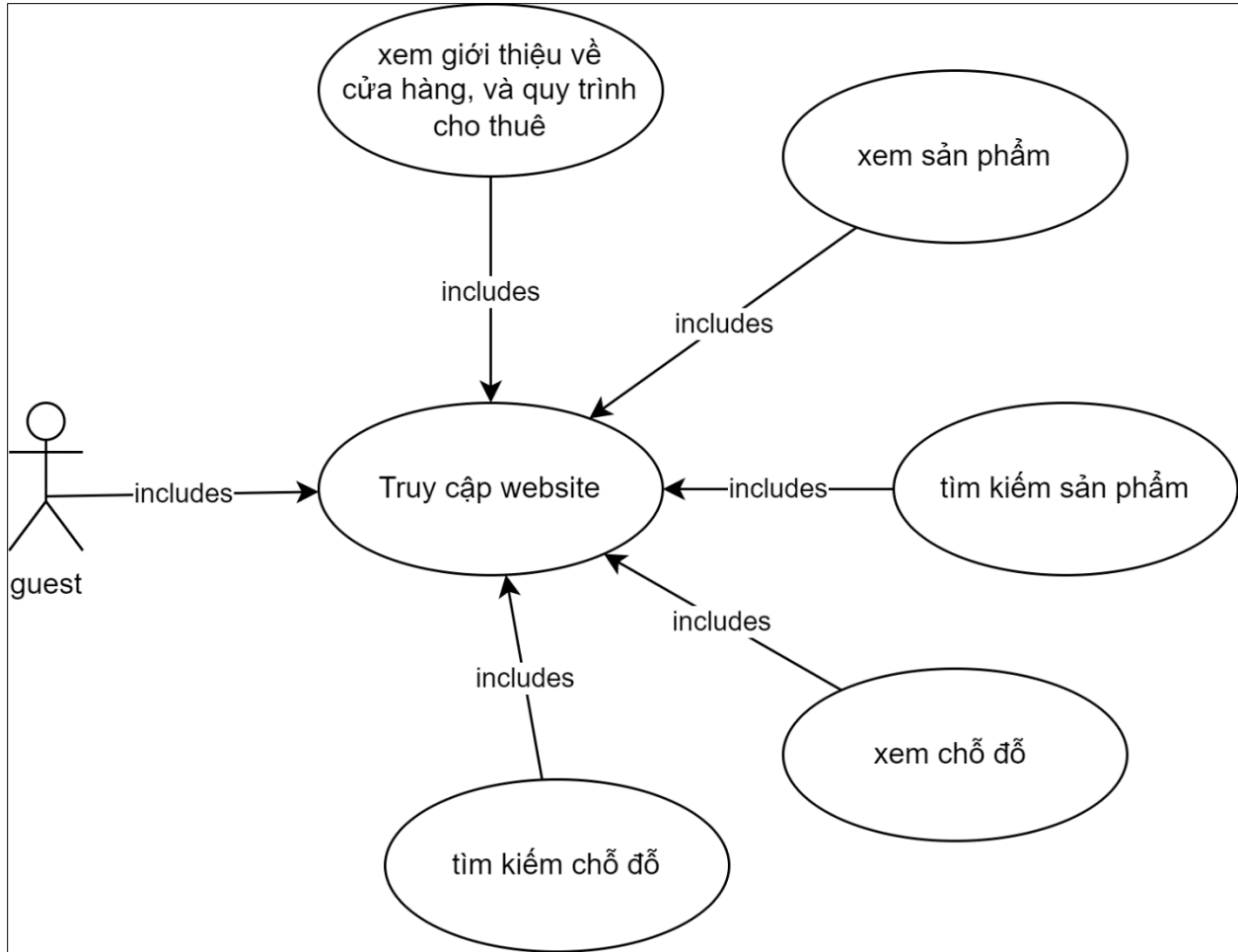
- Là người quản trị toàn bộ hệ thống
- Đăng nhập hệ thống
- Quản trị website
- Quản lý xe máy
- Quản lý nhân viên
- Điểm danh nhân viên
- Quản lý phiếu thuê
- Quản lý bài viết
- Quản lý bình luận
- Quản lý chi nhánh
- Quản lý chỗ đỗ
- Quản lý phiếu đỗ
- Quản lý hóa đơn



Sơ đồ 3-1 Sơ đồ Use case của ADMIN

3.2.2.2 Guest

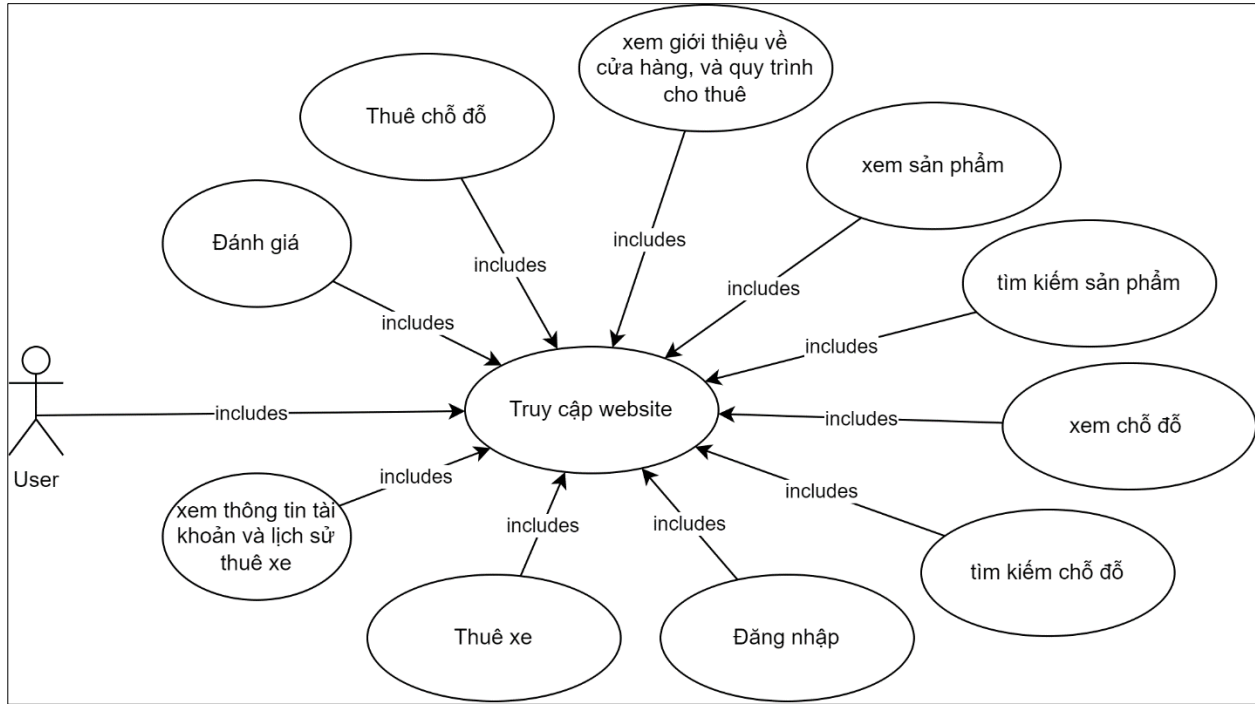
Guest truy cập vào website để xem thông tin cửa hàng, địa chỉ liên hệ, danh sách xe máy, tìm kiếm xe máy, danh sách chỗ đỗ, tìm kiếm chỗ đỗ



Sơ đồ 3-2 Sơ đồ Use case của Guest

3.2.2.3 User

User truy cập vào website để xem thông tin cửa hàng, địa chỉ liên hệ, danh sách xe máy, tìm kiếm xe máy, danh sách chỗ đỗ, tìm kiếm chỗ đỗ, thuê xe, thuê chỗ đỗ, đánh giá, xem thông tin tài khoản và lịch sử thuê xe.

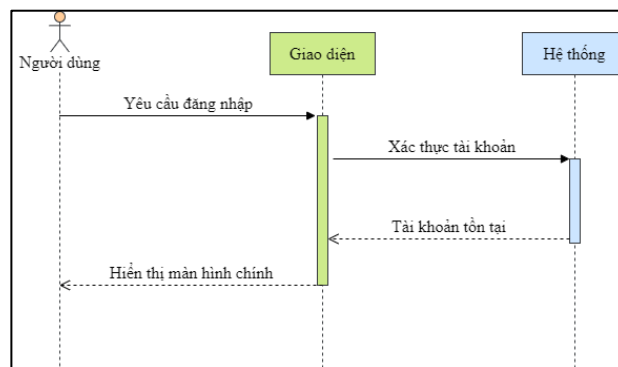


Sơ đồ 3-3 Sơ đồ Use case của User

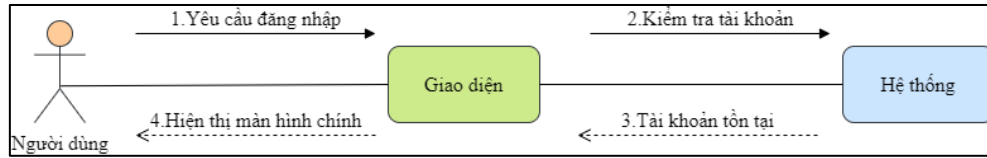
3.2.3 Đặc tả các use case

3.2.3.1 Mô tả quá trình đăng nhập

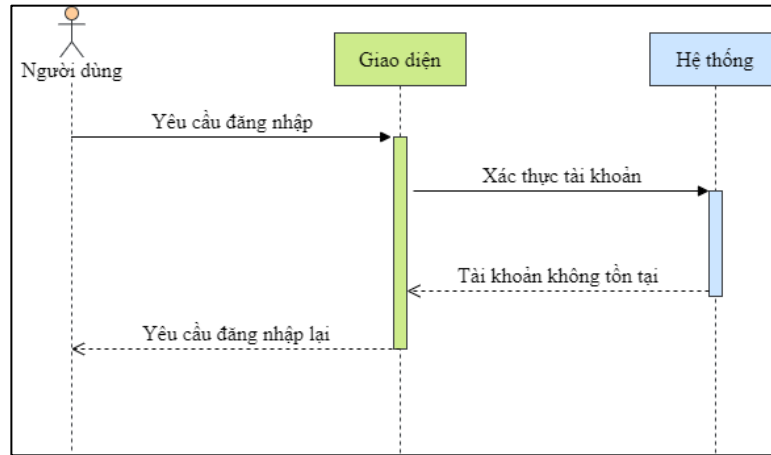
Đặc tả: Use case mô tả quá trình đăng nhập vào hệ thống. Người dùng đăng nhập user name/password. Nếu tài khoản được xác minh hiển thị giao diện trang màn hình chính. Nếu tài khoản không chính xác yêu cầu đăng nhập lại.



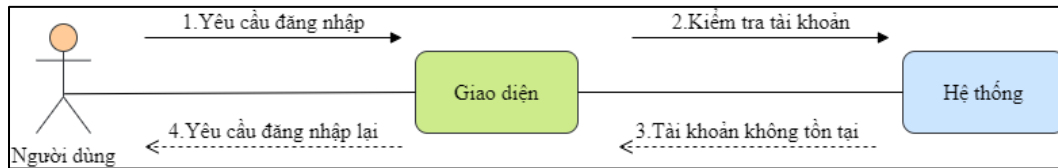
Sơ đồ 3-4 Sơ đồ tuần tự đăng nhập thành công



Sơ đồ 3-5 Sơ đồ hợp tác đăng nhập thành công



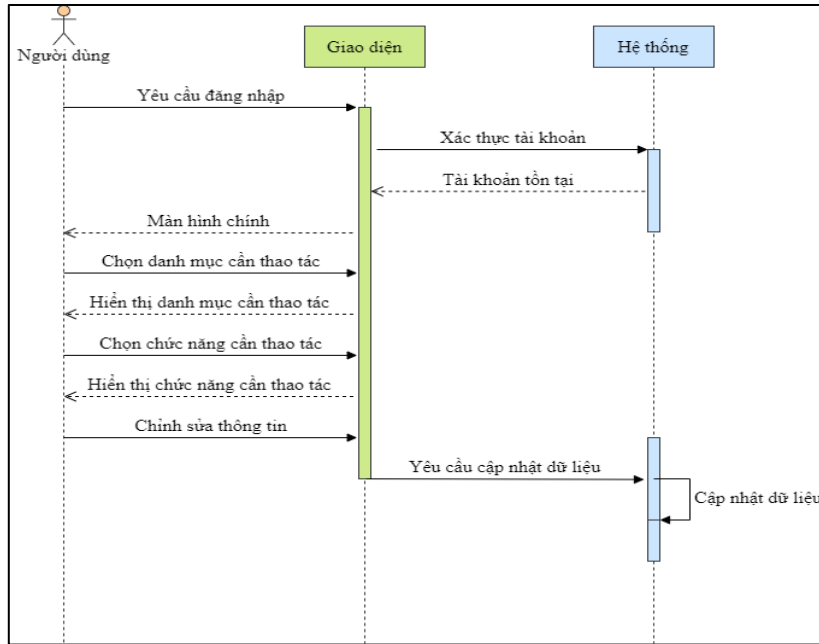
Sơ đồ 3-6 Sơ đồ tuần tự đăng nhập thất bại



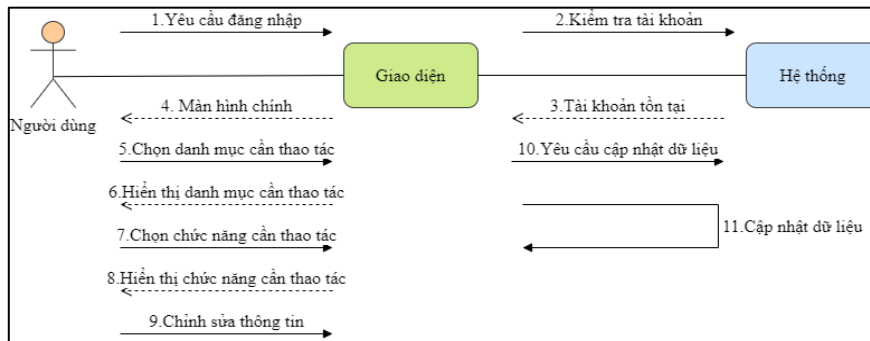
Sơ đồ 3-7 Sơ đồ hợp tác đăng nhập thất bại

3.2.3.2 Mô tả quá trình quản lý

Đặc tả: Khi người dùng đăng nhập và tài khoản đã được xác minh hiển thị trang màn hình chính. Người dùng lựa chọn danh mục cần thao tác, chọn chức năng cần thực hiện chẳng hạn tìm kiếm sản phẩm, xem thông tin tài khoản, lịch sử thuê xe,....



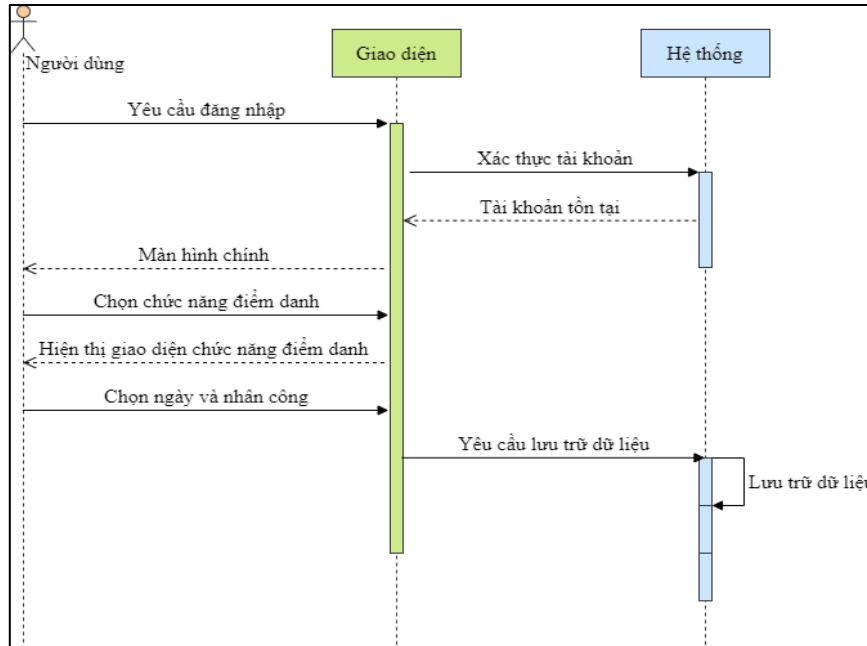
Sơ đồ 3-8 Sơ đồ tuần tự quá trình quản lý



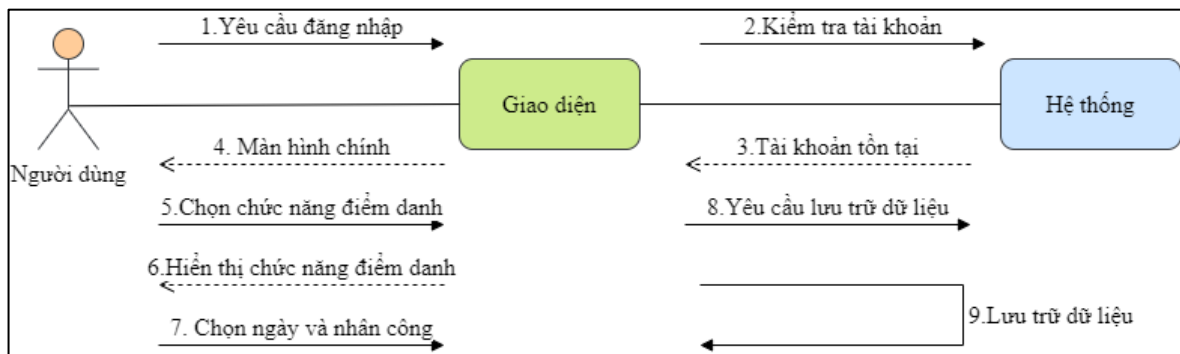
Sơ đồ 3-9 Sơ đồ hợp tác quá trình quản lý

3.2.3.3 Mô tả quá trình điểm danh

Đặc tả: Khi admin đăng nhập và tài khoản đã được xác minh hiển thị trang màn hình chính. Admin lựa chọn chức năng điểm danh cho nhân viên đi làm, chọn thời gian và nhân viên đi làm. Hệ thống sẽ lưu trữ thông tin phục vụ cho quy trình tính lương.



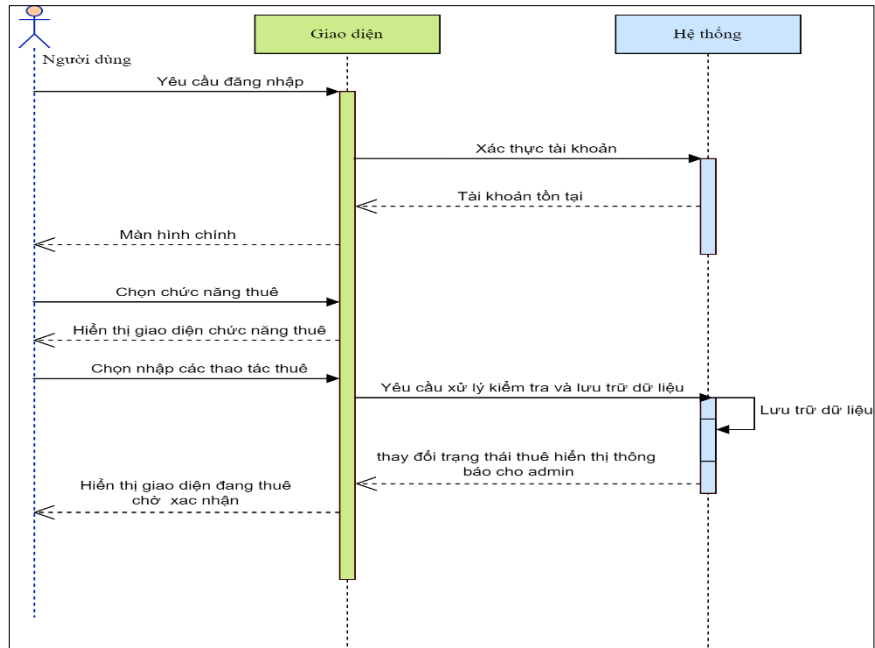
Sơ đồ 3-10 Sơ đồ tuần tự quá trình điểm danh



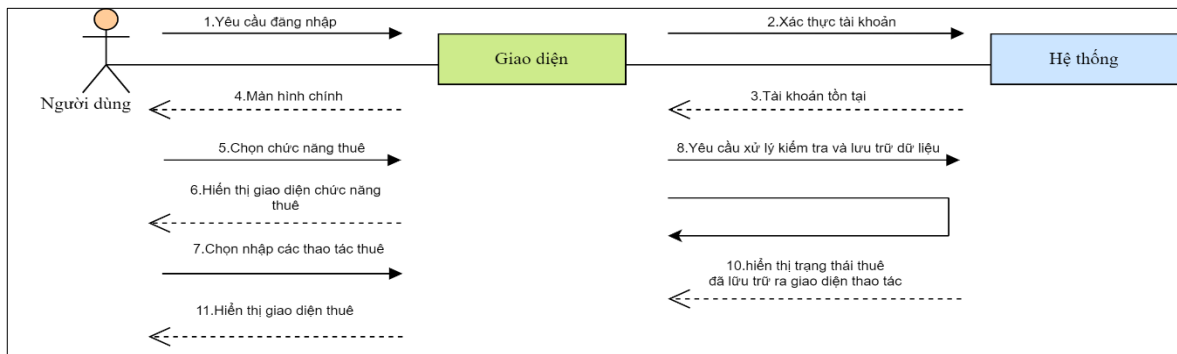
Sơ đồ 3-11 Sơ đồ hợp tác quá trình điểm danh

3.2.3.4 Mô tả quá trình thuê

Đặc tả: Khi người dùng đăng nhập và tài khoản đã được xác minh hiển thị trang màn hình chính. Người dùng lựa chọn chức năng xem xe máy hoặc xem chỗ đỗ xe sau đó thao tác chọn thuê và cung cấp các trường yêu cầu để thuê xong thì chọn thuê để gửi yêu cầu xác nhận cho chủ cửa hàng. Sau khi được chủ cửa hàng xác nhận thì trạng thái thuê sẽ thay đổi



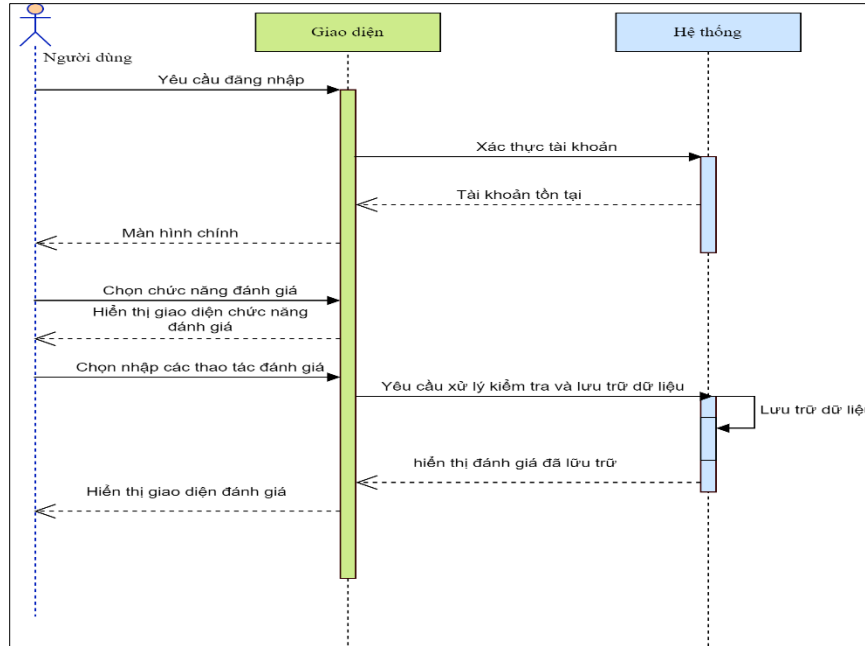
Sơ đồ 3-12 Sơ đồ tuần tự quá trình thuê



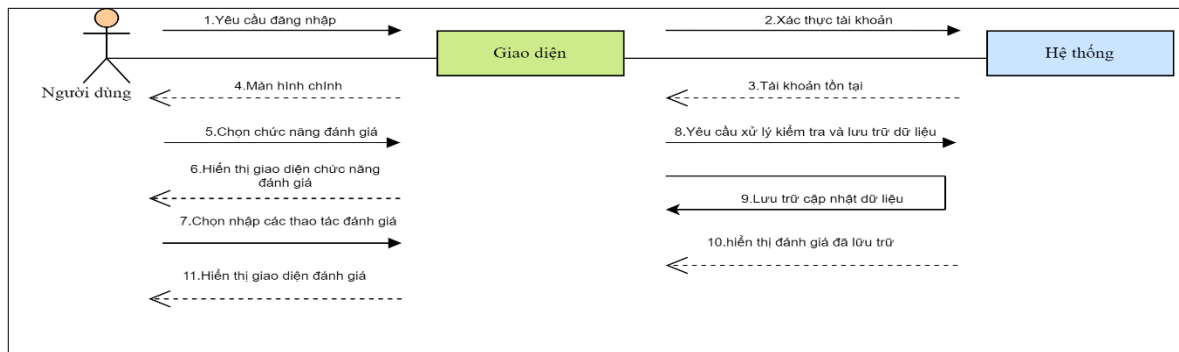
Sơ đồ 3-13 Sơ đồ hợp tác quá trình thuê

3.2.3.5 Mô tả quá trình đánh giá

Đặc tả: Khi người dùng đăng nhập và tài khoản đã được xác minh hiển thị trang màn hình chính. Người dùng lựa chọn chức năng xem xe máy sau đó thao tác chọn thuê và cung cấp các trường yêu cầu để đánh giá xe.



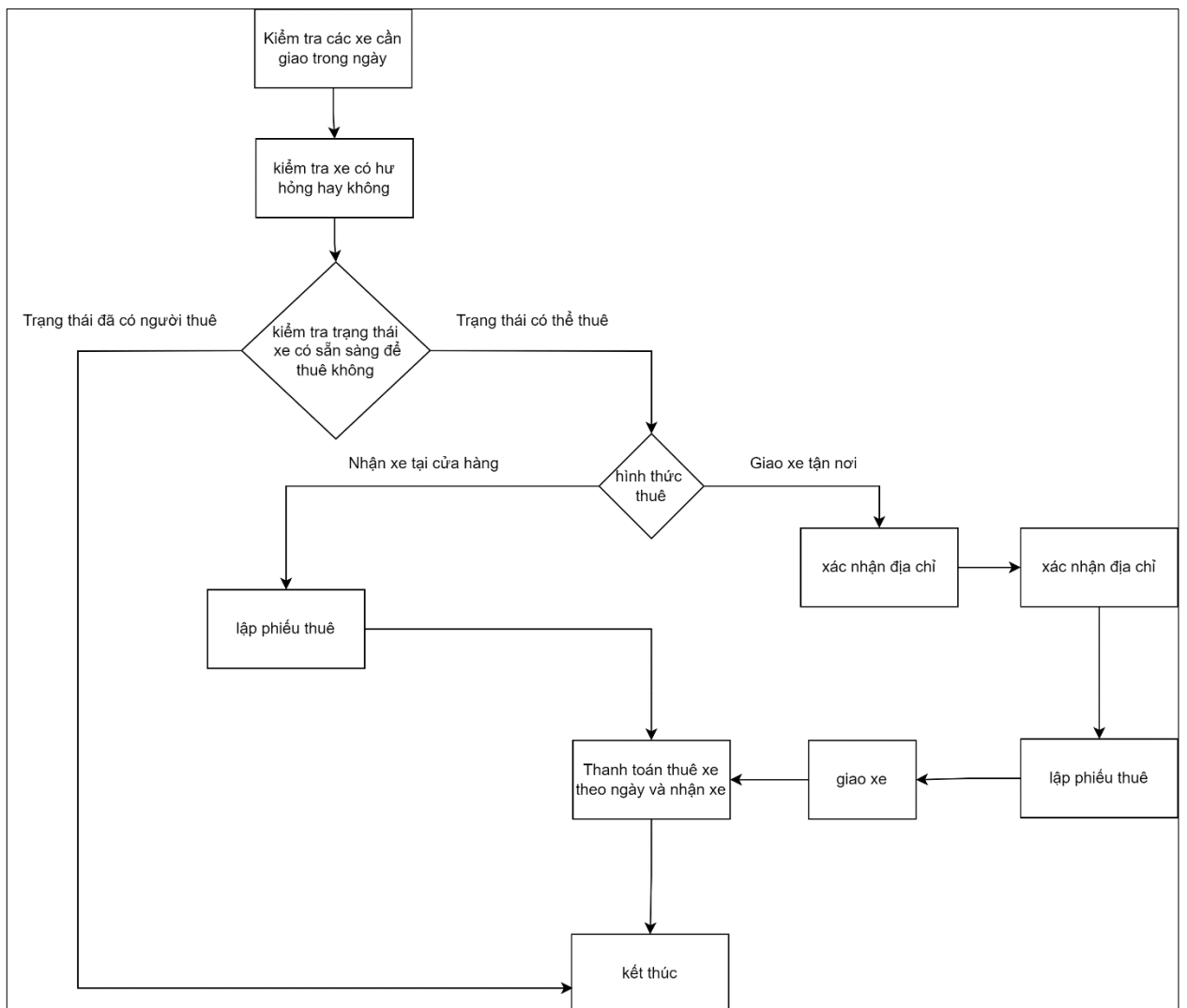
Sơ đồ 3-14 Sơ đồ tuần tự quá trình đánh giá



Sơ đồ 3-15 Sơ đồ hợp tác quá trình đánh giá

3.2.4 Quy trình cho thuê xe

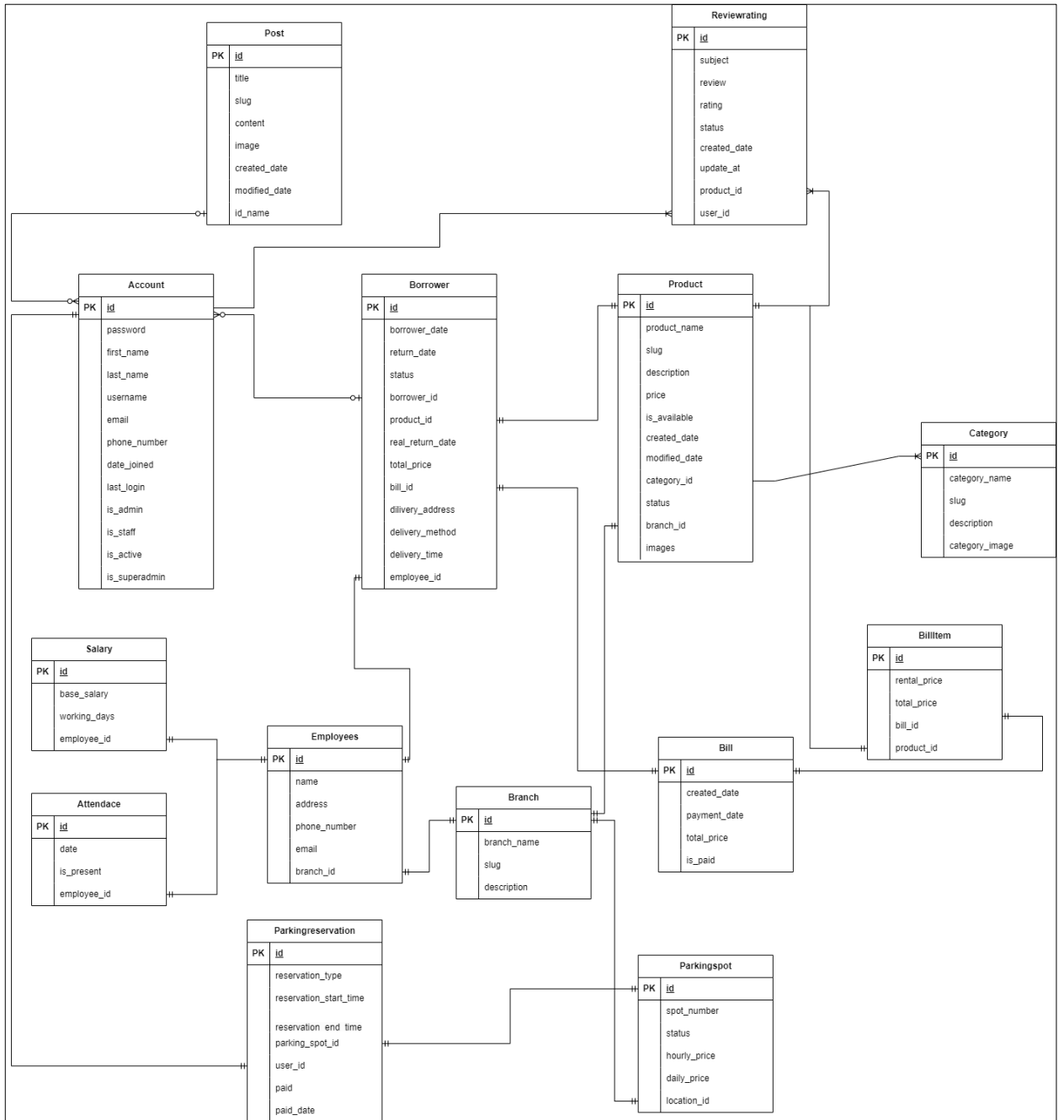
3.2.4.1 Kiểm tra xe có bị hư hao trước khi cho thuê



Sơ đồ 3-16 Quy trình thuê

3.3 Thiết kế hệ thống

3.3.1 Sơ đồ cơ sở dữ liệu



Sơ đồ 3-17 Sơ đồ cơ sở dữ liệu

3.3.2 Chi tiết cơ sở dữ liệu

➤ Quản lý tài khoản

Bảng 3-15 Cơ sở dữ liệu bảng tài khoản

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Account	id	Integer	Mã tài khoản
2		password	String(128)	Mật khẩu tài khoản
3		First_name	String(50)	Họ của tài khoản
4		Last_name	String(50)	Tên của tài khoản
5		Username	String(50)	Tên đăng nhập
6		Email	String(50)	Địa chỉ thư điện tử
7		Phone_number	String(12)	Số điện thoại
8		Date_joined	Datetime(6)	Ngày tạo tài khoản
9		Last_login	Datetime(6)	Ngày cuối đăng nhập vào website
10		Is_admin	Integer(1)	Quyền admin
11		Is_staff	Integer(1)	Quyền nhân viên
12		Is_active	Integer(1)	Quyền truy cập vào website
13		Is_superuser	Integer(1)	Quyền toàn quyền trên hệ thống

➤ Quản lý chi nhánh

Bảng 3-16 Cơ sở dữ liệu bảng chi nhánh

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Branch	id	Integer	Mã chi nhánh
2		Branch_ame	String(50)	Tên chi nhánh
3		Slug	String(100)	Chỉ hướng chi nhánh
4		description	String(4000)	Mô tả chi nhánh

➤ Quản lý loại xe máy

Bảng 3-17 Cơ sở dữ liệu bảng loại xe

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Category	<u>Id</u>	Integer	Mã loại xe
2		Category_name	String(50)	Tên loại xe
3		Slug	String(100)	Hướng dẫn tới loại xe
4		Description	String(4000)	Mô tả loại xe
5		Category_image	String(100)	Hình của loại xe

➤ Quản lý điểm danh nhân viên

Bảng 3-18 Cơ sở dữ liệu bảng điểm danh nhân viên

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Attendance	<u>Id</u>	Integer	Mã điểm danh nhân viên
2		Date	String(100)	Tên loại thiết bị
3		Is_present	Int(1)	Chấm công
4		Employee_id	Integer	Mã nhân viên

➤ Quản lý nhân viên

Bảng 3-19 Cơ sở dữ liệu bảng nhân viên

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Employee	<u>Id</u>	Integer	Mã nhân viên
2		name	String(255)	Tên nhân viên
3		Address	String(255)	Địa chỉ nhân viên
4		Phone_number	String(20)	Số điện thoại nhân viên
5		Email	String(255)	Địa chỉ thư điện tử
6		Branch_id	Integer	Chi nhánh nhân viên làm

➤ Bảng tính lương nhân viên

Bảng 3-20 Cơ sở dữ liệu bảng tính lương nhân viên

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Salary	<u>Id</u>	Integer	Mã tính lương
2		Base_salary	Decimal(10)	Lương cơ bản
3		Working_days	Integer	Số ngày làm được chấm công
4		Employee_id	Integer	Mã nhân viên

➤ Quản lý phiếu đỗ

Bảng 3-21 Cơ sở dữ liệu bảng phiếu đỗ

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Parkingreservation	<u>Id</u>	Integer	Mã phiếu đỗ
2		Reservation_type	String(200)	Loại đỗ xe
3		Reservation_start_time	Datetime(6)	Ngày bắt đầu đỗ
4		Reservation_end_time	Datetime(6)	Ngày lấy xe
5		Parking_spot_id	Integer	Mã chỗ đỗ
6		User_id	Integer	Mã người đỗ
7		Paid_date	Datetime(6)	Ngày thanh toán
8		Paid	Integer	Kiểm tra thanh toán

➤ Bảng quản lý chỗ đỗ xe

Bảng 3-22 Cơ sở dữ liệu bảng chỗ đỗ xe

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Parkingspot	<u>Id</u>	Integer	Mã chỗ đỗ
2		Spot_number	String(200)	Mô tả chỗ đỗ
3		Status	String(255)	Trạng thái chỗ đỗ
4		Hourly_price	Decimal(10,3)	Giá tiền đỗ theo giờ
5		Daily_price	Decimal(10,3)	Giá tiền đỗ theo ngày

6		<i>Location_id</i>	Integer	Mã chi nhánh
----------	--	--------------------	---------	--------------

➤ Quản lý bài viết

Bảng 3-23 Cơ sở dữ liệu bảng bài viết

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Post	<u>Id</u>	Integer	Mã bài viết
2		Title	String(200)	Tiêu đề bài viết
3		Slug	String(200)	Hướng dẫn đến bài viết
4		<i>Id_name</i>	Integer	Mã người viết
5		Content	String(4000)	Nội dung bài viết
6		Created_date	Datetime(6)	Ngày tạo bài viết
7		Modified_date	Datetime(6)	Ngày chỉnh sửa
8		Image	String(100)	Hình ảnh dùng trong bài viết

➤ Hóa đơn

Bảng 3-24 Cơ sở dữ liệu bảng hóa đơn

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Bill	<u>Id</u>	Integer	Mã hóa đơn
2		Created_date	Datetime(6)	Ngày lập hóa đơn
3		Payment_date	Datetime(6)	Ngày thanh toán hóa đơn
4		Total_price	Integer	Tổng tiền
5		Is_paid	Integer	Trạng thái thanh toán hay chưa

➤ Chi tiết hóa đơn

Bảng 3-25 Cơ sở dữ liệu bảng chi tiết hóa đơn

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	BillItem	<u>Id</u>	Integer	Mã chi tiết hóa đơn
2		Rental_price	Integer	Đơn giá xe trên hóa đơn
3		Total_price	Integer	Tổng tiền trong hóa đơn
4		Bill_id	Integer	Mã hóa đơn
5		Product_id	Integer	Mã xe

➤ Phiếu mượn

Bảng 3-26 Cơ sở dữ liệu bảng phiếu mượn

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Borrower	<u>Id</u>	Integer	Mã phiếu mượn
2		Borrow_date	Datetime(6)	Ngày mượn
3		Return_date	Datetime(6)	Ngày trả
4		Status	String(20)	Trạng thái
5		Borrower_id	Integer	Mã người mượn
6		Product_id	Integer	Mã xe mượn
7		Real_return_date	Datetime(6)	Ngày trả thực tế
8		Total_price	Integer	Tổng tiền
9		Bill_id	Integer	Mã hóa đơn
10		Delivery_address	String(4000)	Địa chỉ giao xe
11		Delivery_method	String(50)	Phương thức thuê xe
12		Delivery_time	Datetime(6)	Giờ giao xe
13		Employee_id	Integer	Mã nhân viên giao xe

➤ Bảng xe

Bảng 3-27 Cơ sở dữ liệu bảng xe

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Product	<u>Id</u>	Integer	Mã xe
2		Product_name	String(200)	Tên xe
3		Slug	String(200)	Hướng dẫn tới xe
4		Description	String(4000)	Mô tả xe
5		Price	Integer	Giá xe
6		Images	String(100)	Hình ảnh xe
7		Is_available	Integer	Tính khả dụng của xe
8		Created_date	Datetime(6)	Ngày thêm xe
9		Modified_date	Datetime(6)	Ngày chỉnh sửa
10		Category_id	Integer	Mã loại xe
11		Status	String(100)	Trạng thái xe
12		Branch_id	Integer	Chi nhánh xe

➤ Bảng đánh giá

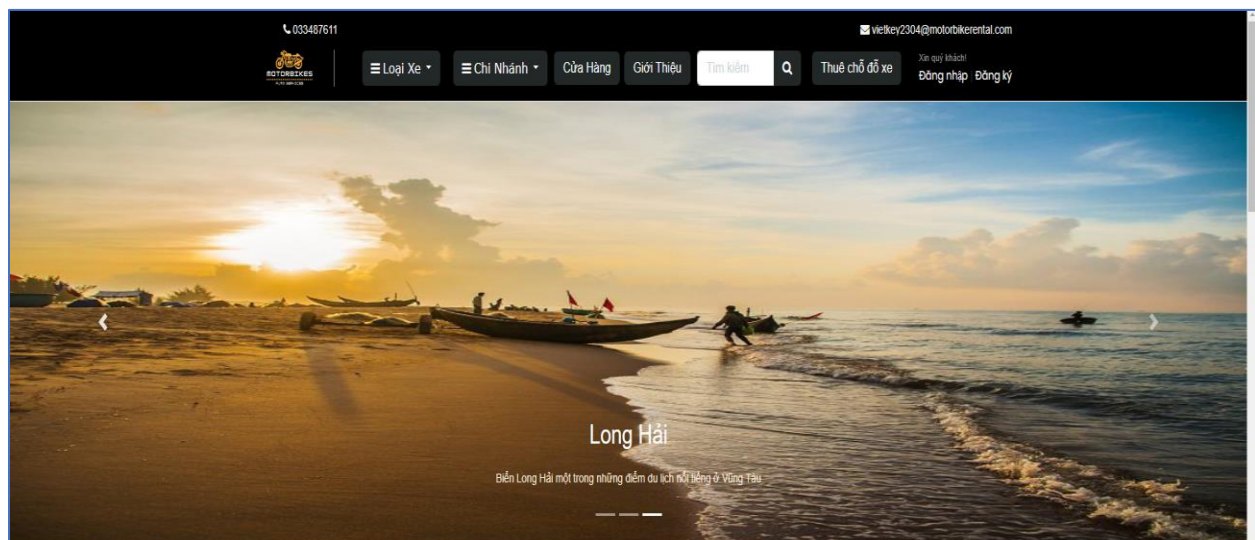
Bảng 3-28 Cơ sở dữ liệu bảng xe

STT	Tên bảng	Tên cột	Kiểu	Mục đích
1	Reviewrating	<u>Id</u>	Integer	Mã đánh giá
2		Subject	String(100)	Tiêu đề
3		Review	String(4000)	Đánh giá
4		Rating	Double	Tính sao đánh giá
5		Status	Integer	Trạng thái đánh giá
6		Created_at	Datetime(6)	Ngày đánh giá
7		update_at	Datetime(6)	Ngày chỉnh sửa
8		Product_id	Integer	Mã xe
9		User_id	Integer	Mã người đánh giá

CHƯƠNG 4 XÂY DỰNG WEBSITE

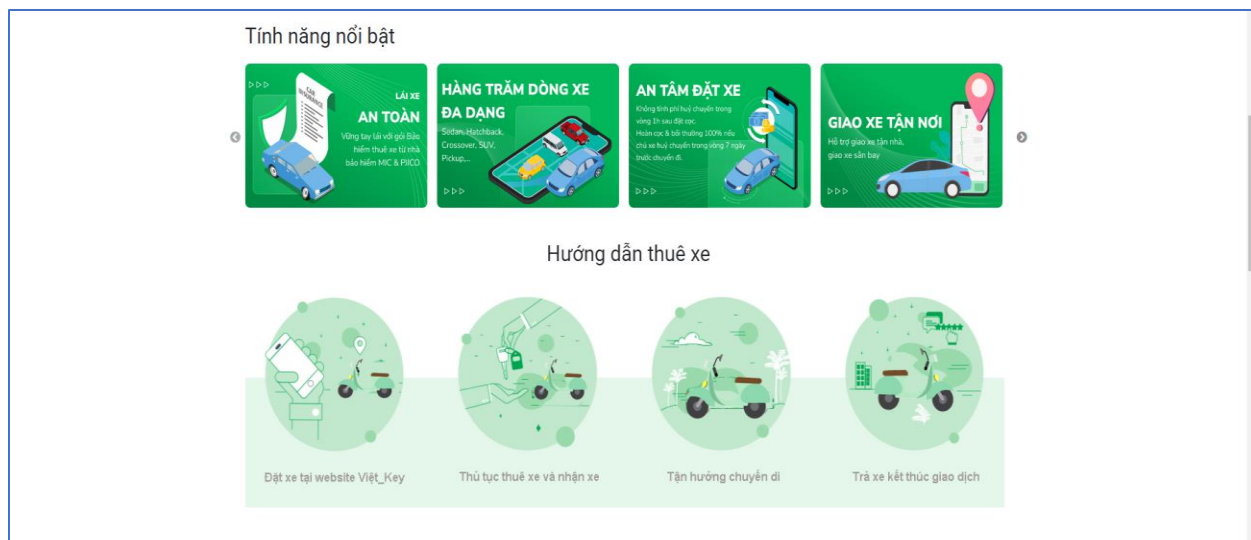
4.1 Giao diện và chức năng khách

Khi khách hàng truy cập trang web, hiển thị đầu tiên là trang chủ. Giao diện trang chủ thân thiện dễ nhìn giúp khách hàng sẽ dễ dàng biết được chức năng của website.



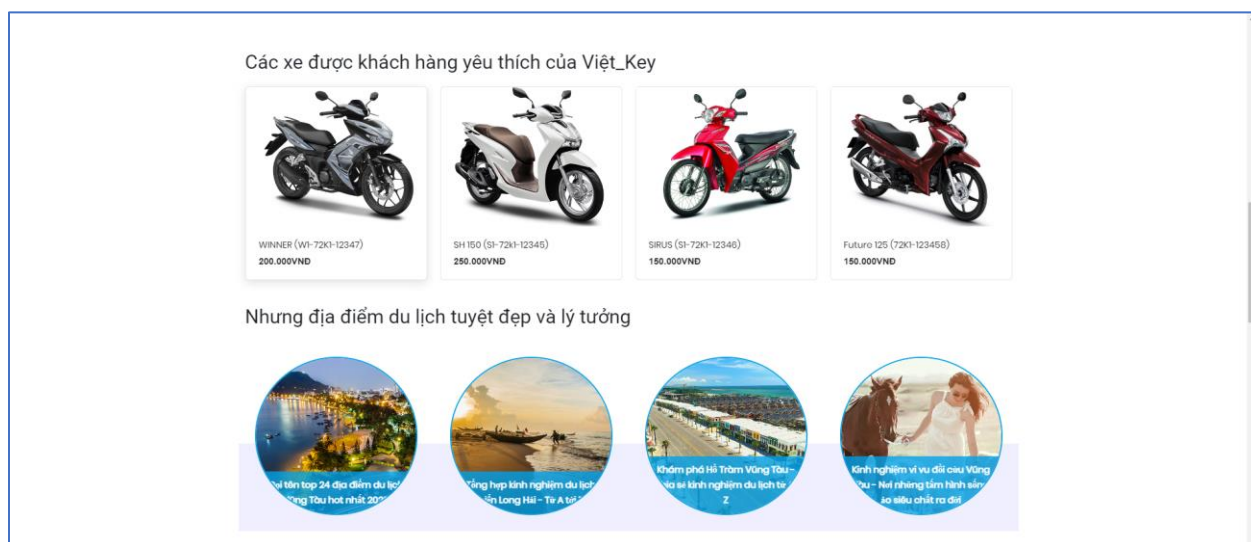
Hình 4.1 Phần header, navbar và silder của trang chủ

Khối đầu tiên hình 4.1 bao gồm phần header chứa logo của cửa hàng, số điện thoại và gmail liên hệ. Phần navbar bao gồm các trang như giới thiệu, cửa hàng, loại xe, tìm kiếm xe, chi nhánh thuê chỗ đỗ xe, còn có một dòng chữ xin chào nếu là khách hàng chưa đăng nhập sẽ là quý khách còn khi đăng nhập sẽ là tên của khách hàng đăng ký, ngoài ra còn có hai nút đăng nhập và đăng ký để ở navbar để khách hàng dễ dàng sử dụng hai chức năng này và silder thể hiện các địa điểm du lịch lý tưởng ở gần cửa hàng.



Hình 4.2 Phần tính năng nổi bật và hướng dẫn thuê xe

Khởi tiếp theo hình 4.2 giới thiệu phần tính năng nổi bật và hướng dẫn thuê xe, để có thể cho khách hàng thấy được sự tiện ích, dễ nhìn, dễ sử dụng của website nên hướng dẫn thuê xe là điều cần thiết khách hàng cần phải quan tâm đến.



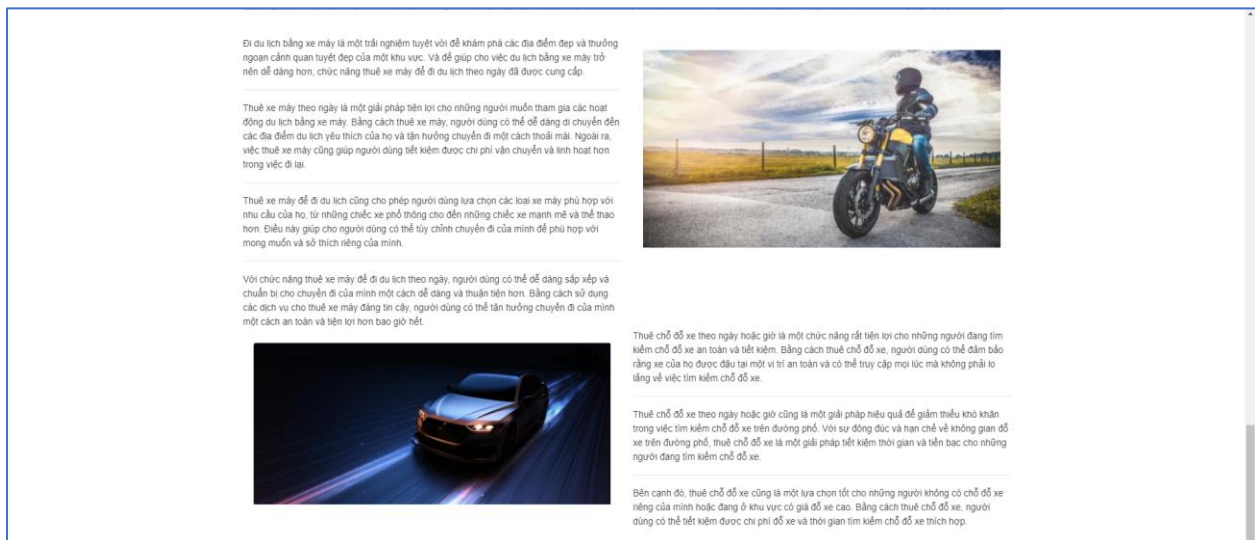
Hình 4.3 phần các xe được khách hàng yêu thích và các bài viết về địa điểm du lịch

Khởi tiếp theo hình 4.3 giới thiệu về các xe được khách hàng yêu thích và các bài viết về địa điểm du lịch. Khách hàng có thể dễ dàng nhìn thấy được các xe được nhiều người ưu tiên lựa chọn ở cửa hàng ngoài ra có thể tham khảo thêm các bài viết địa điểm du lịch ở gần các chi nhánh.



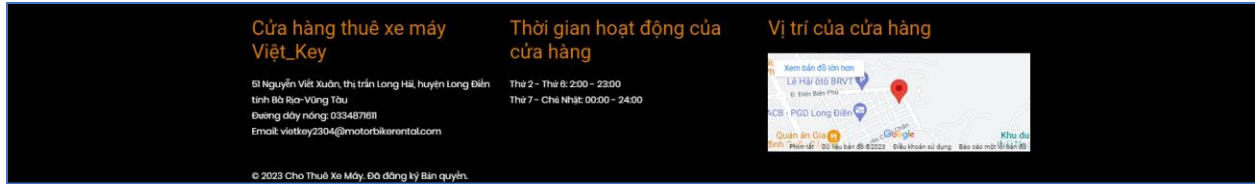
Hình 4.3 giới thiệu về tính năng cho thuê chỗ đỗ xe

Khối tiếp theo hình 4.4 khi tới khối này thì khách hàng có thể thấy được phần giới thiệu về chức năng thuê chỗ đỗ và hiểu rõ hơn về chức năng này trên website cũng như dễ dàng thao tác chức năng này trên website.



Hình 4.4 Phần các bài viết liên quan để chức năng của website

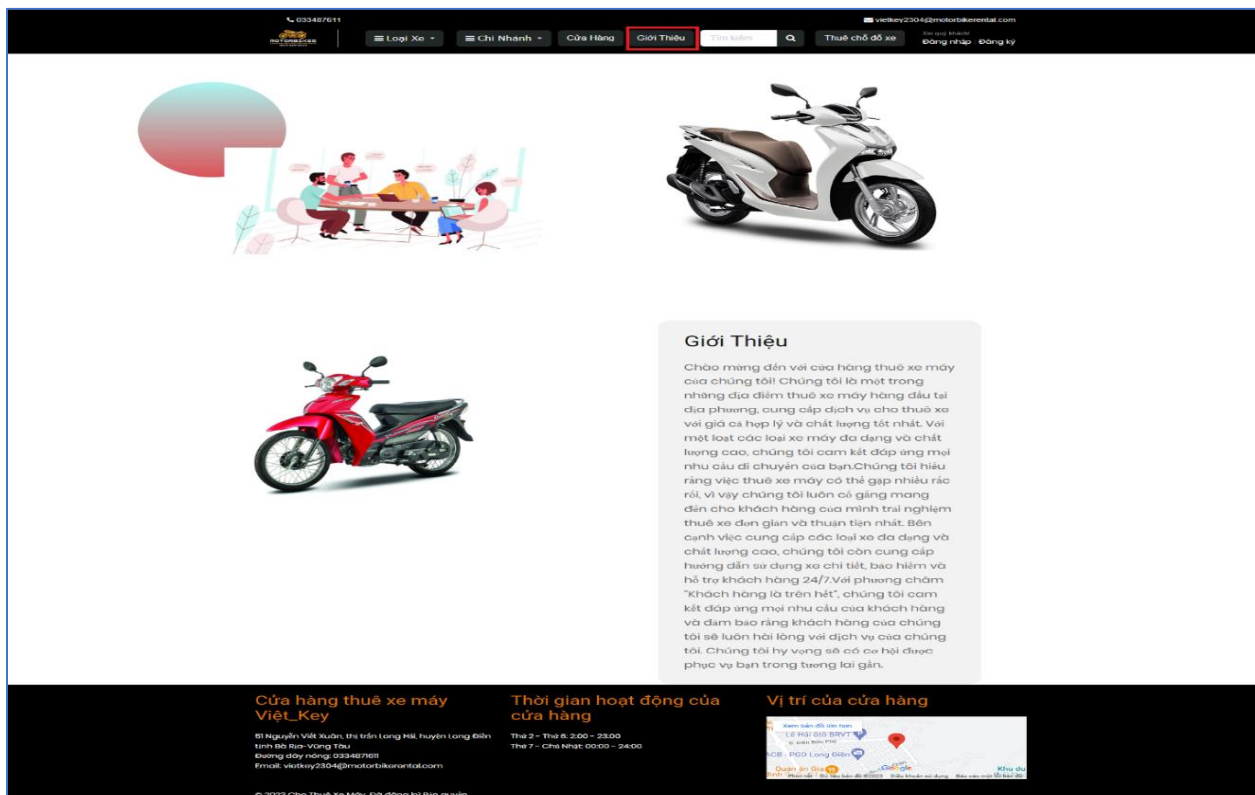
Khối tiếp theo hình 4.5 những lời giới thiệu về tiện ích của website mang lại để khách hàng nắm rõ mục tiêu của website mang lại cho khách hàng sự tiện lợi và nhanh chóng khi sử dụng dịch vụ của cửa hàng Việt Key.



Hình 4.5 Phần footer của trang chủ

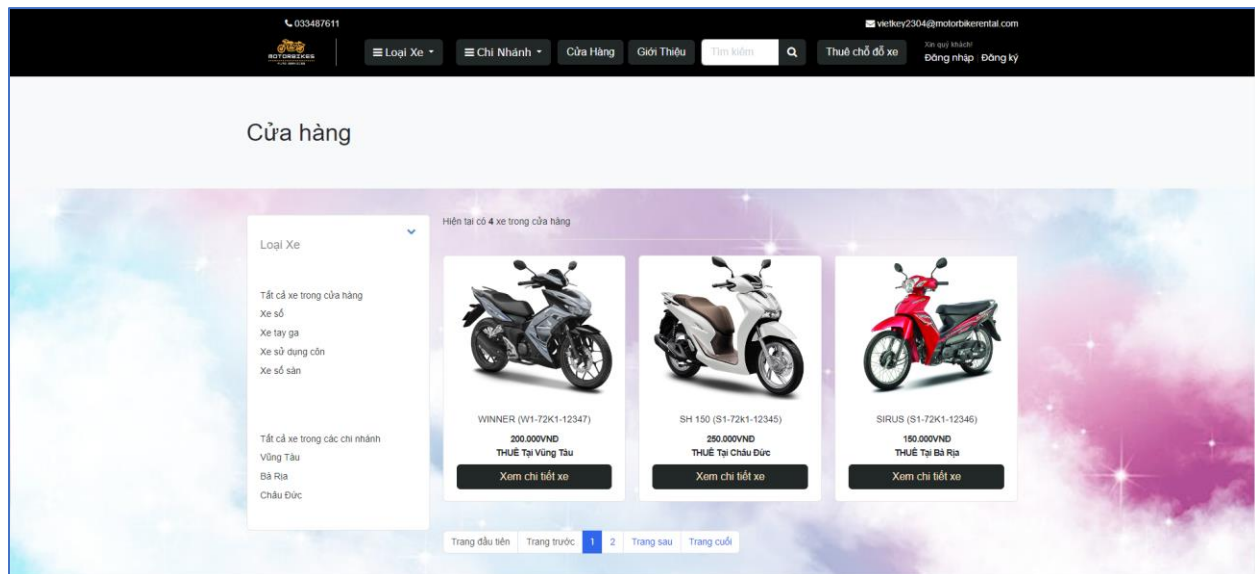
Khối cuối cùng hình 4.6 phần footer của trang chủ hiện thị tóm gọn thông tin của cửa hàng Việt Key, định vị trên bản đồ và các thông tin hoạt động của cửa hàng.

Tiếp theo khi khách hàng chọn trang giới thiệu, giao diện như hình 4.7 giới thiệu về tiêu chí đem công nghệ chuyên đổi số vào cuộc sống để giúp cho ước mơ thế giới công nghệ ngày càng tiến gần hơn với cuộc sống của chúng ta, để khách hàng thấy được sự tiện lợi của công nghệ của website thuê xe của cửa hàng mang lại là vô cùng hữu ích và có triển vọng phát triển lớn mạnh hơn.



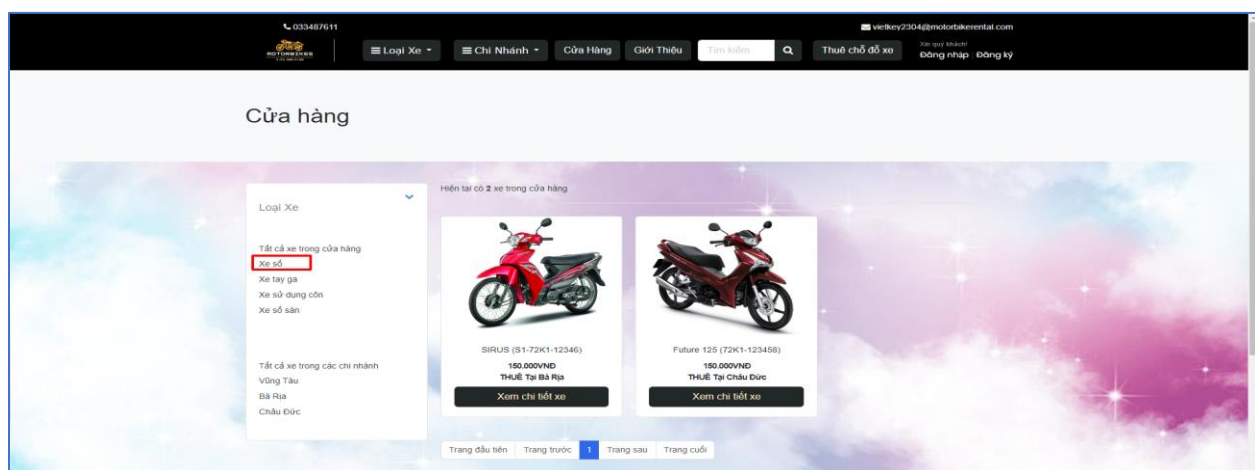
Hình 4.6 Giao diện trang giới thiệu

Tiếp theo khi khách hàng chọn menu cửa hàng thì sẽ được chuyển sang trang danh sách xe có trong cửa hàng ngoài ra còn hiển thị cả chi nhánh và loại xe trong trang này để khách hàng dễ dàng xem được thông tin xe, dễ dàng chọn chức năng lọc loại xe thích hợp với bản thân. Chọn vào cửa hàng thì tất cả xe sẽ được hiện ra kèm theo trạng thái của xe để khách hàng dễ dàng phân biệt và lựa chọn thuê xe. Xe thì tất cả các xe đang sử dụng có trong cơ sở dữ liệu sẽ được hiển thị. Mỗi trang sẽ hiển thị 3 sản phẩm.



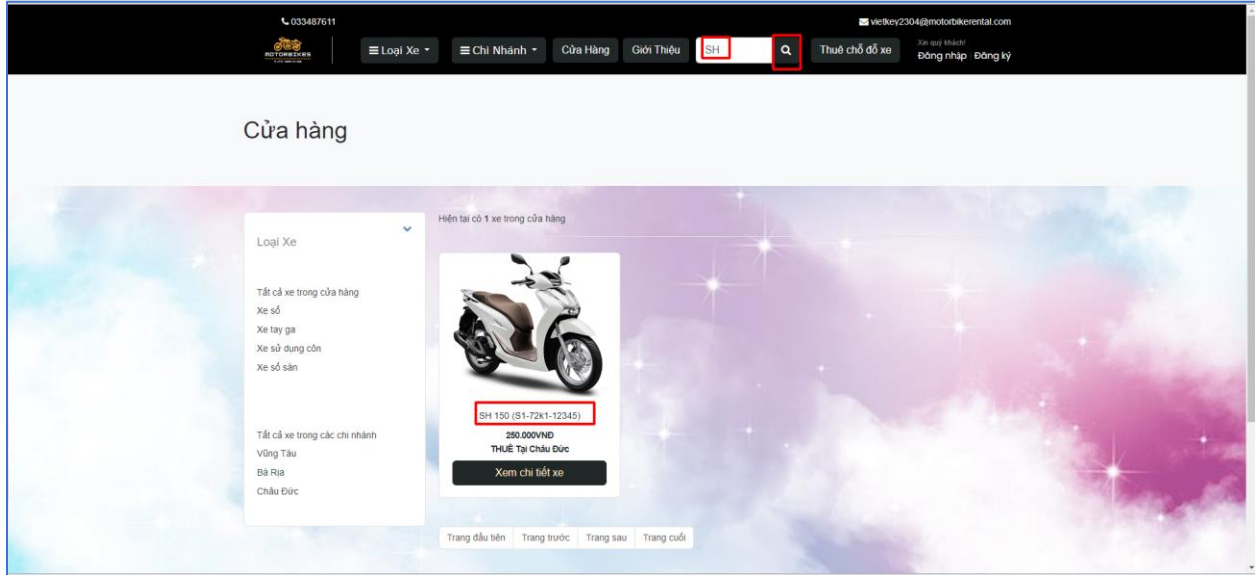
Hình 4.7 Giao diện hiển thị tất cả các xe

Khi chọn vào loại xe thì các xe trong loại đó sẽ được hiển thị.



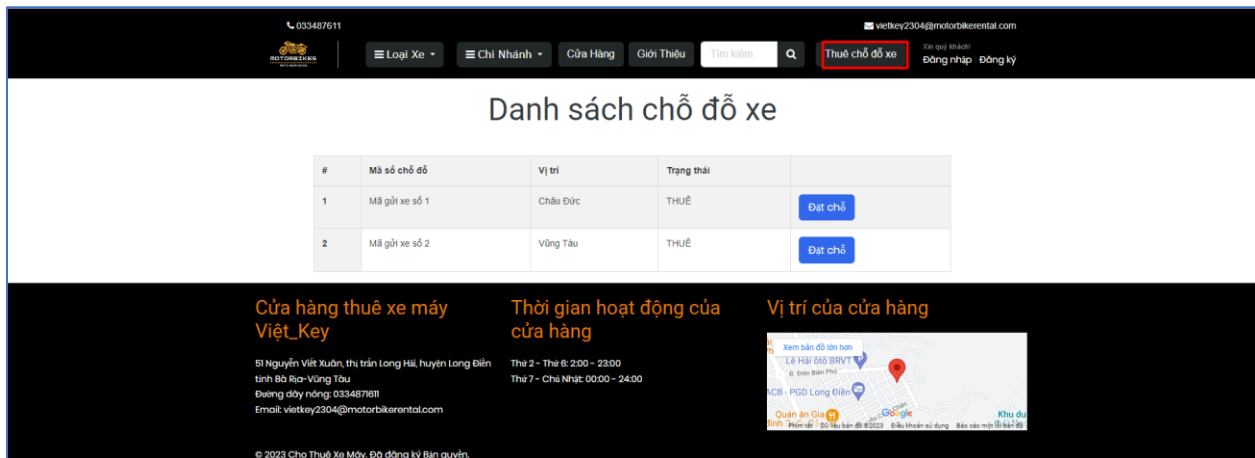
Hình 4.8 Giao diện xe theo loại “Xe số”.

Tiếp theo khách hàng có thể ở bất kỳ trang nào có navbar cũng đều có thể sử dụng tính năng tìm kiếm xe hình 4.9 sẽ cho thấy được chức năng hoạt động một cách dễ dàng và tiện lợi cho khách hàng.



Hình 4.9 Giao diện khi sử dụng chức năng tìm kiếm

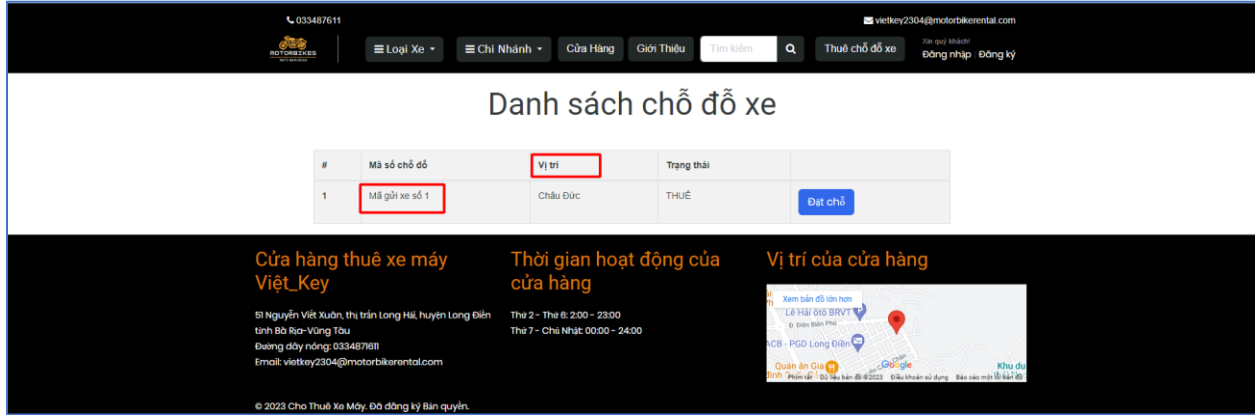
Tiếp theo khi khách chọn vào trang thuê chỗ đỗ xe mặc định tất cả các chỗ đỗ sẽ được hiển thị như hình 4.10



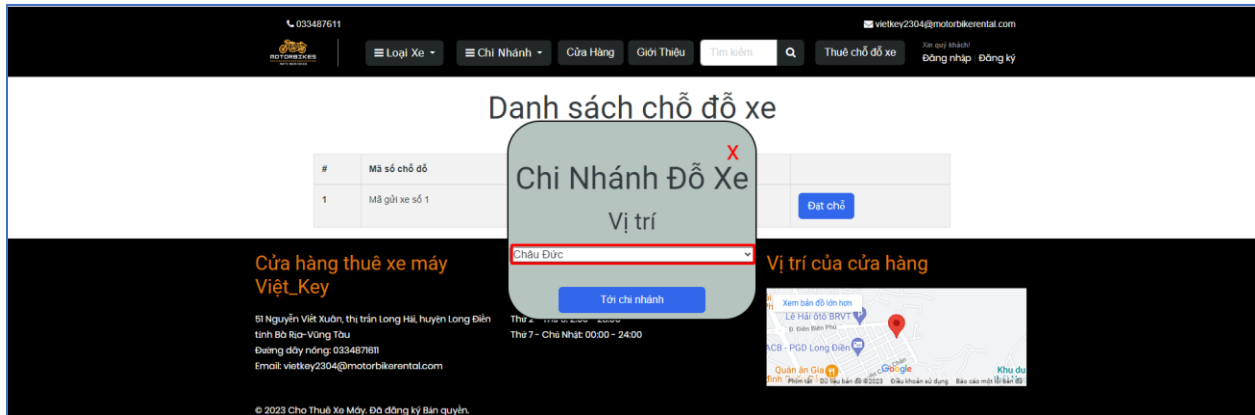
Hình 4.10 Giao diện danh sách chỗ đỗ xe

Đối với thuê chỗ đỗ xe thì khách hàng có thể dễ dàng quan sát được trạng thái chỗ đỗ có khả dụng để thuê hay không, ngoài ra khách hàng dễ dàng thấy được vị trí muốn đỗ và

mã số đồ xe và nút đặt chỗ vô cùng đơn giản và dễ dùng để khách hàng có thể sử dụng một cách không e ngại. Ngoài ra, khách hàng còn có thể sử dụng tính năng lọc để xem vị trí gần mình có những chỗ nào bằng cửa sổ nhỏ vị trí.



Hình 4.11 Giao diện ảnh khi lọc các vị trí châu đức

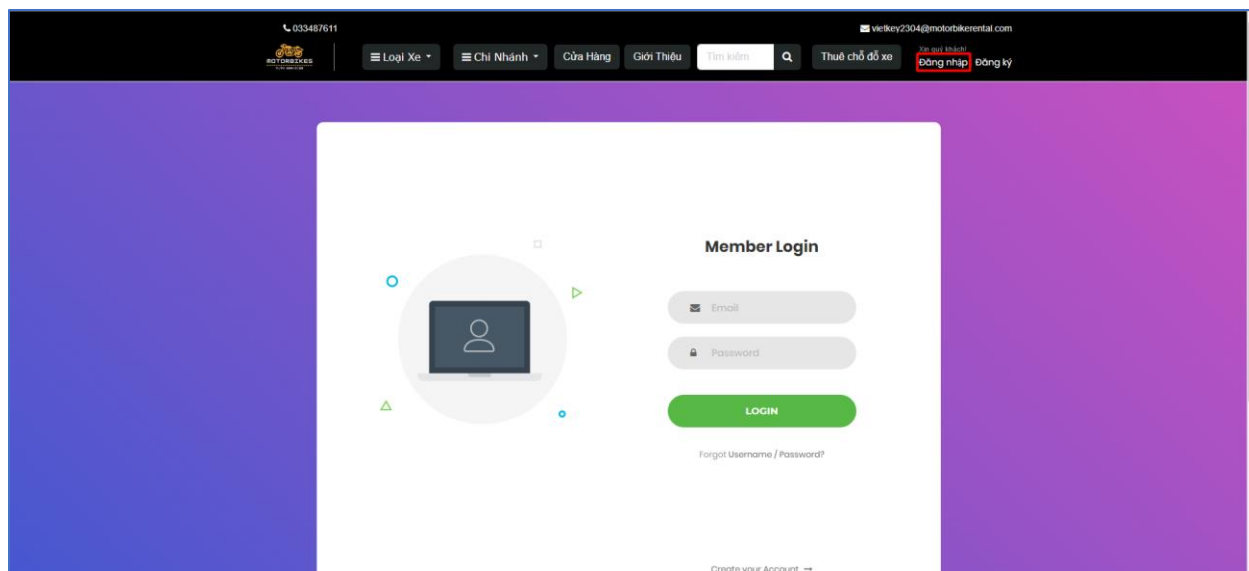


Hình 4.12 Giao diện ảnh lọc các vị trí châu đức

Như vậy phần giao diện và chức năng người dùng bao gồm giới thiệu cửa hàng, giới thiệu xe, thông tin chỗ đồ và tìm kiếm thông tin. Đem lại những hình ảnh chân thật và tạo thêm nhiều sự lựa chọn cho khách hàng có nhu cầu.

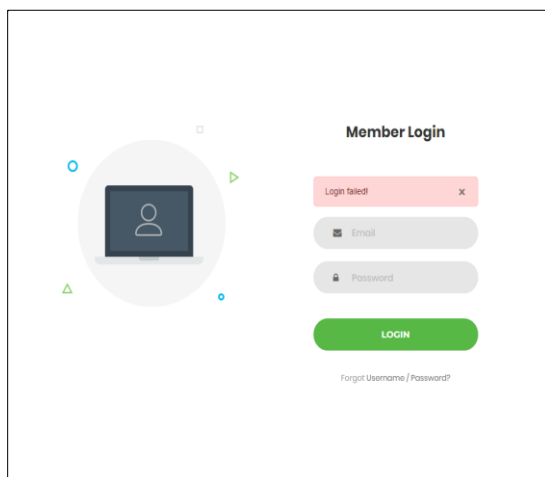
4.2 Giao diện và chức năng Admin

Khi người dùng muốn truy cập vào trang admin thì phải thông qua bước đăng nhập vào website. Giao diện đăng nhập như hình 4.13



Hình 4.13 Giao diện trang đăng nhập

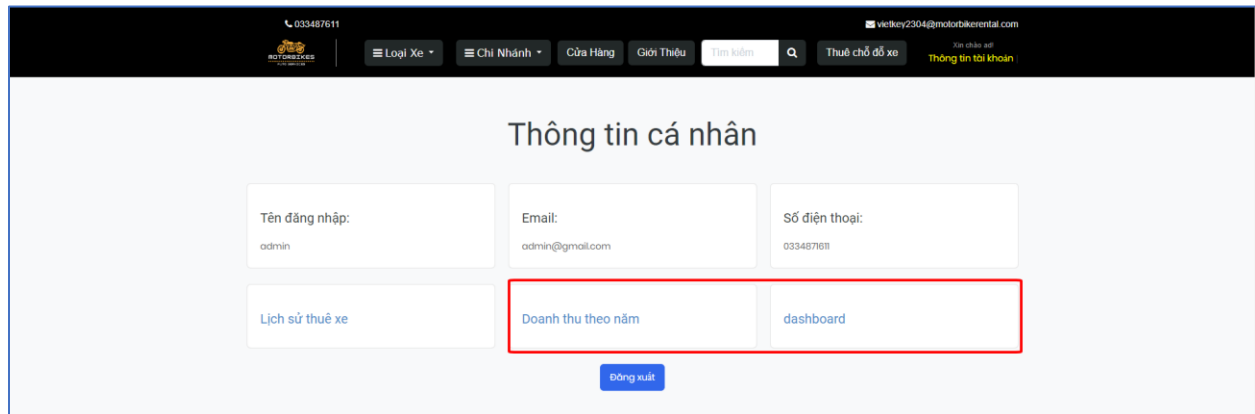
Người dùng phải điền đầy đủ thông tin tài khoản và mật khẩu, nếu không thành công giao diện sẽ hiển thị như hình 4.14



Hình 4.14 Giao diện trang đăng nhập không thành công

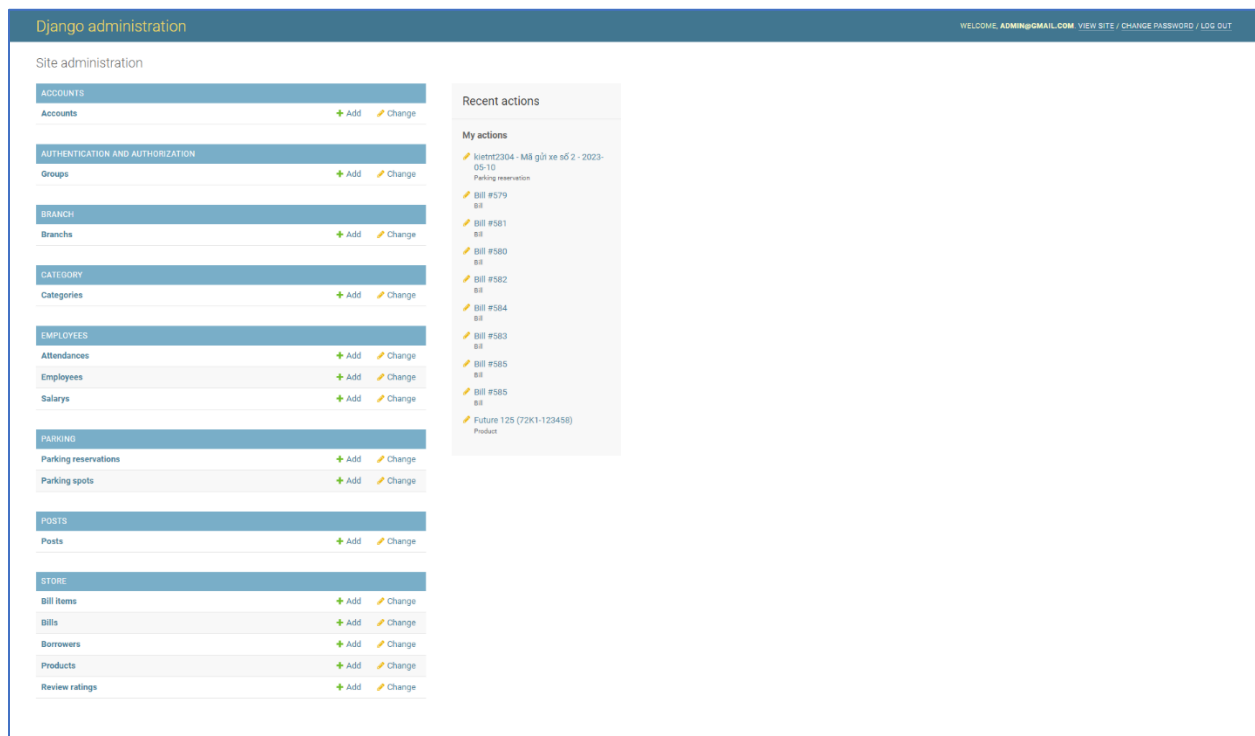
Khi tài khoản được đăng ký nhưng không được kích hoạt qua gmail thì khi đăng nhập cũng sẽ xuất hiện không đăng nhập thành công. Các trường hợp đăng nhập không thành công là: Không nhập các trường bắt buộc tài khoản và mật khẩu, sai mật khẩu hoặc tài khoản, chưa kích hoạt tài khoản.

Sau khi đăng nhập thành công sẽ hiển thị giao diện trang quản lý thông tin tài khoản ở đây có thể nhìn thấy tài khoản của bản thân nếu là tài khoản của admin thì sẽ có thêm hai đường dẫn để đến dashboard và thông kê còn nếu tài khoản của khách hàng thì sẽ không thấy hai đường dẫn này.



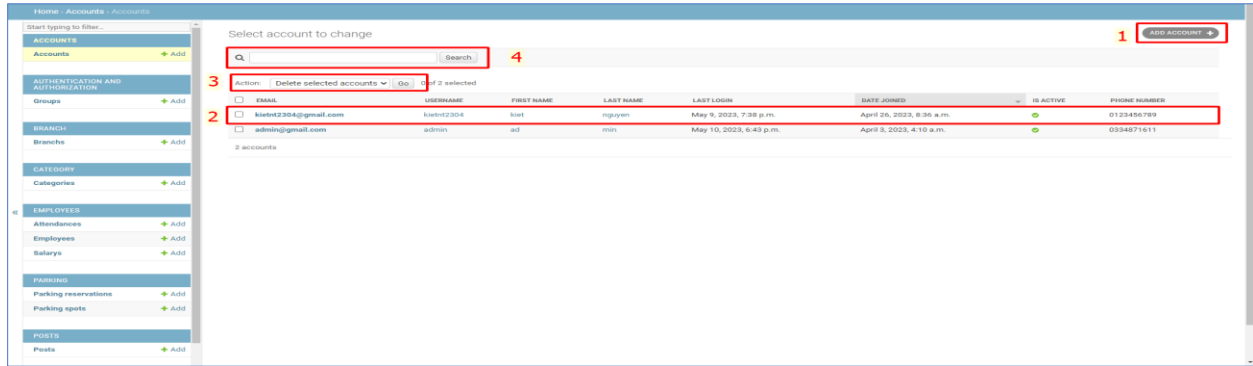
Hình 4.15 Trang thông tin tài khoản

Có thể vào trang dashboard của admin bằng cách chọn vào đường dẫn dashboard để đến trung tâm quản lý website.



Hình 4.16 Trang quản lý dashboard

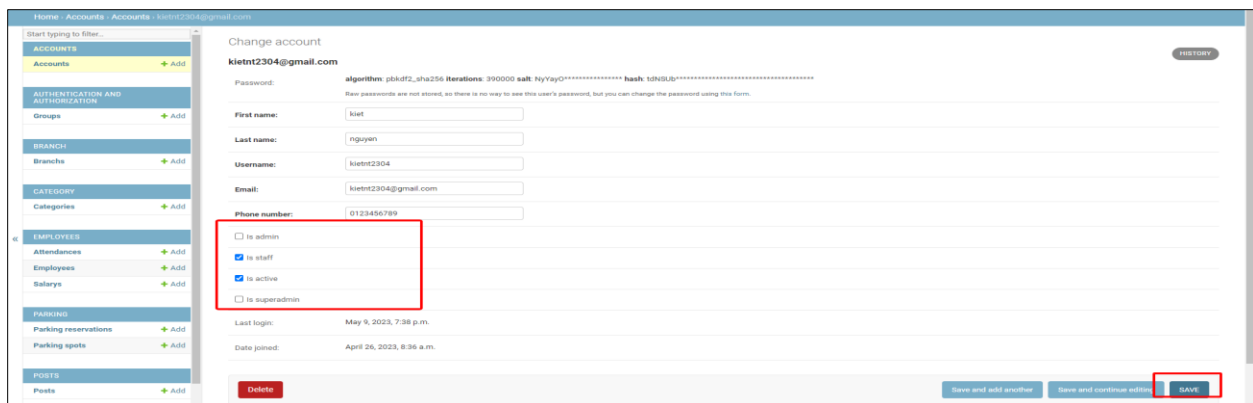
ở trang dashboard này có thể quản lý được tất cả các thành phần của website thực hiện được các chức năng thêm sửa xóa từng thành phần một của website đầu tiên chính là tài khoản.



Hình 4.17 Chức năng trong tài khoản

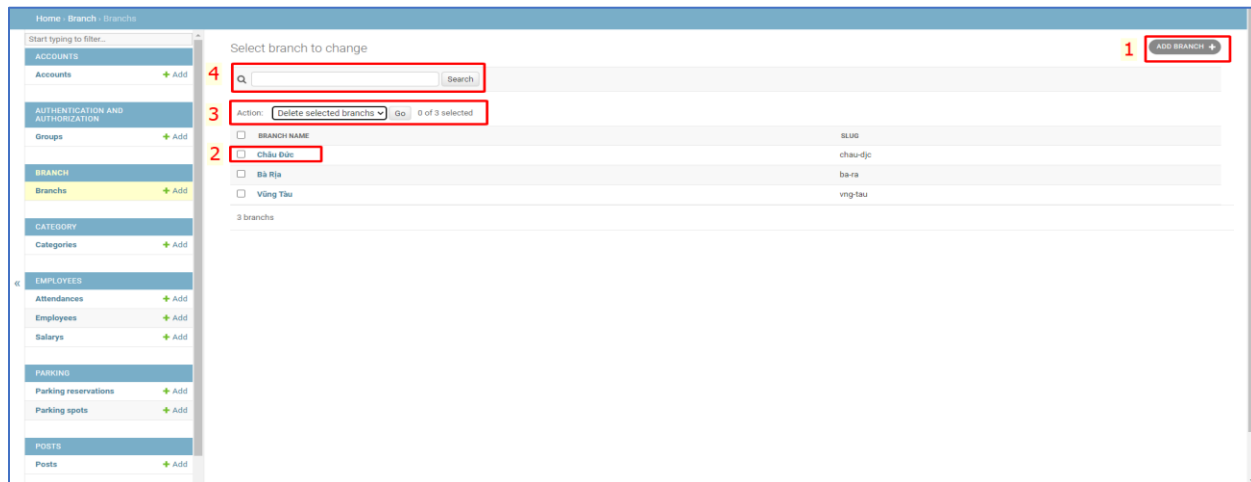
Trong trang tài khoản có thể thực hiện các chức năng như thêm, sửa, xóa, tìm kiếm tài khoản đang có trên hệ thống:

1. Thêm tài khoản ở nút này
2. Chỉnh sửa tài khoản bằng cách chọn vào dòng tài khoản muốn sửa
3. Chọn vào ô vuông trước tài khoản muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa
4. Để tìm kiếm tài khoản nhập username vào ô và nhấn search



Hình 4.18 Chức năng chỉnh sửa tài khoản và phân quyền cho tài khoản

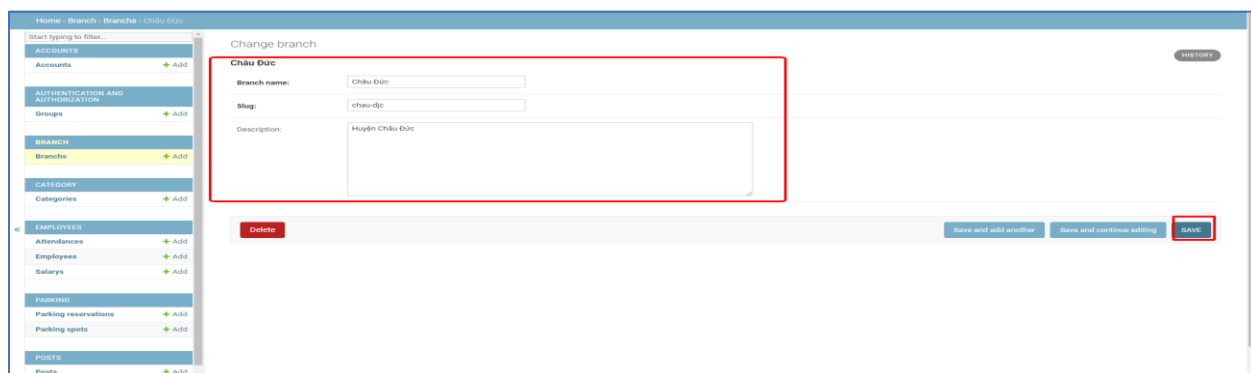
ở hình 4.18 là chức năng sửa tài khoản khi chọn vào dòng tài khoản để sửa thì sẽ chuyển qua trang như hình. còn có thể thiết lập quyền cho tài khoản sau đó khi chỉnh sửa xong thì nhấn save để lưu chỉnh sửa.



Hình 4.19 Chức năng trong chi nhánh

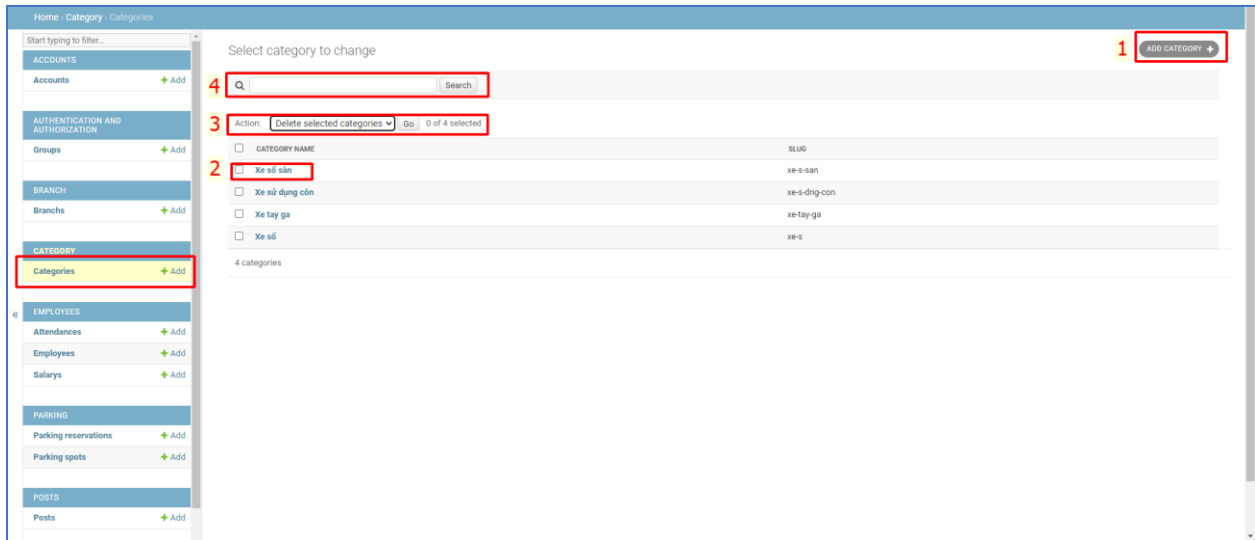
Trong trang dashboard của chi nhánh có thể thực hiện các chức năng như thêm, sửa, xóa, tìm kiếm chi nhánh đang có trên hệ thống:

1. Thêm chi nhánh ở nút này
2. Chỉnh sửa chi nhánh bằng cách chọn vào dòng chi nhánh muốn sửa
3. Chọn vào ô vuông trước chi nhánh muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa
4. Để tìm kiếm chi nhánh nhập branchname vào ô và nhấn search



Hình 4.20 Chức năng chỉnh sửa chi nhánh

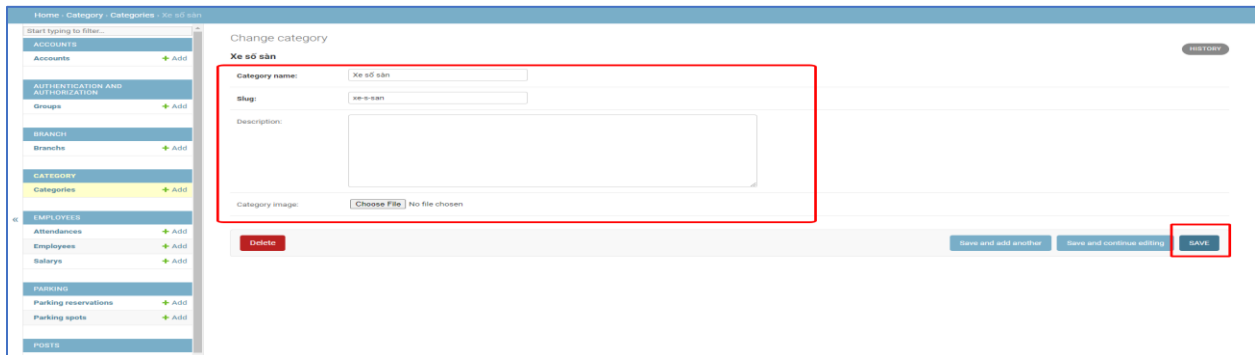
ở hình 4.20 là chức năng sửa chi nhánh khi chọn vào dòng chi nhánh để sửa thì sẽ chuyển qua trang như hình, sau đó khi chỉnh sửa xong thì nhấn save để lưu chỉnh sửa.



Hình 4.21 Chức năng trong loại xe

Trong trang dashboard của loại xe có thể thực hiện các chức năng như thêm, sửa, xóa, tìm kiếm loại xe đang có trên hệ thống:

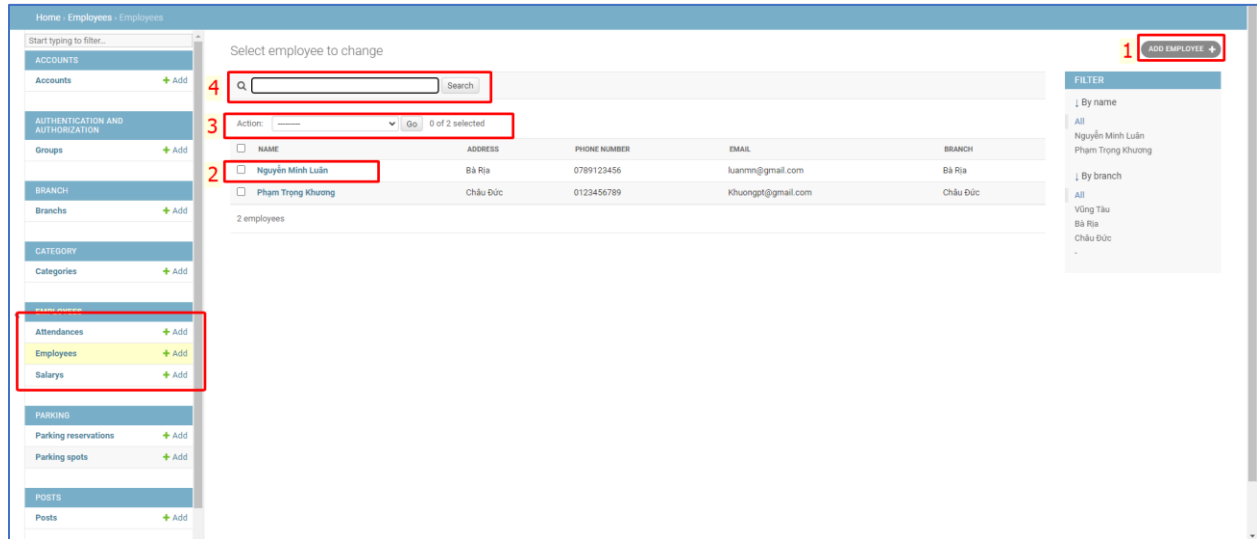
1. Thêm loại xe ở nút này
2. Chỉnh sửa loại xe bằng cách chọn vào dòng loại xe muốn sửa
3. Chọn vào ô vuông trước loại xe muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa
4. Để tìm kiếm loại xe nhận category_name vào ô và nhấn search



Hình 4.22 Chức năng chỉnh sửa loại xe

ở hình 4.22 là chức năng sửa loại xe khi chọn vào dòng loại xe để sửa thì sẽ chuyển qua trang như hình, sau đó khi chỉnh sửa xong thì nhấn save để lưu chỉnh sửa.

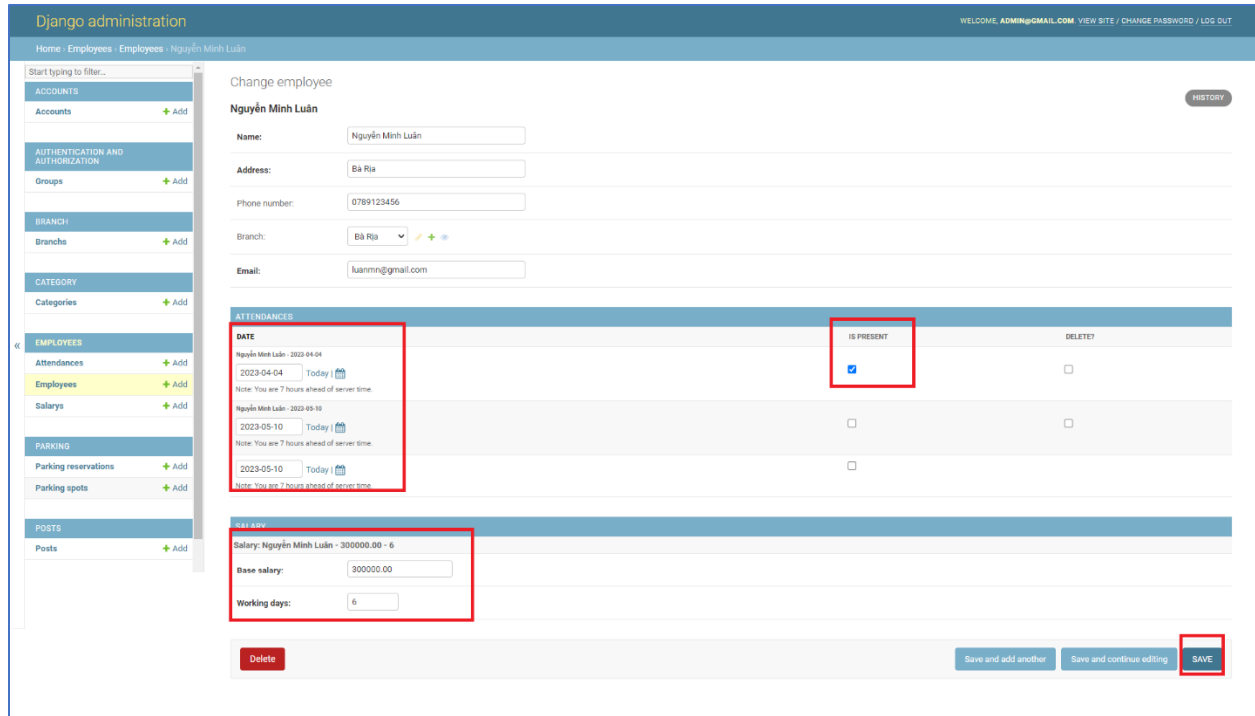
Với Dashboard có thể thêm nhân viên và có thể chấm công tại dashboard cũng như xuất bảng lương tại đây.



Hình 4.23 Chức năng quản lý nhân viên

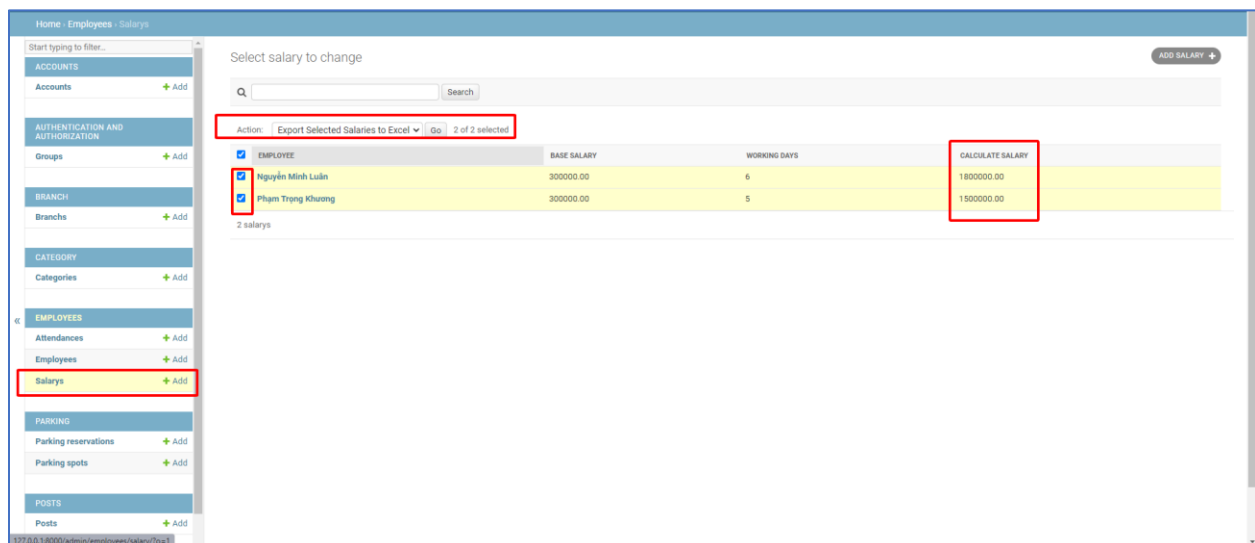
1. Thêm nhân viên mới ở nút này
2. Chỉnh sửa thông tin nhân viên bằng cách chọn vào dòng nhân viên muốn sửa thông tin
3. Chọn vào ô vuông trước nhân viên muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa
4. Để tìm kiếm nhân viên nhập name vào ô và nhấn search

Ngoài ra còn có thể chỉnh sửa cũng như chấm công cho nhân viên ngay tại màn hình chi tiết chỉnh sửa nhân viên, còn có thể nhìn thấy lương cơ bản cùng ngày đi làm của nhân viên trong tháng.



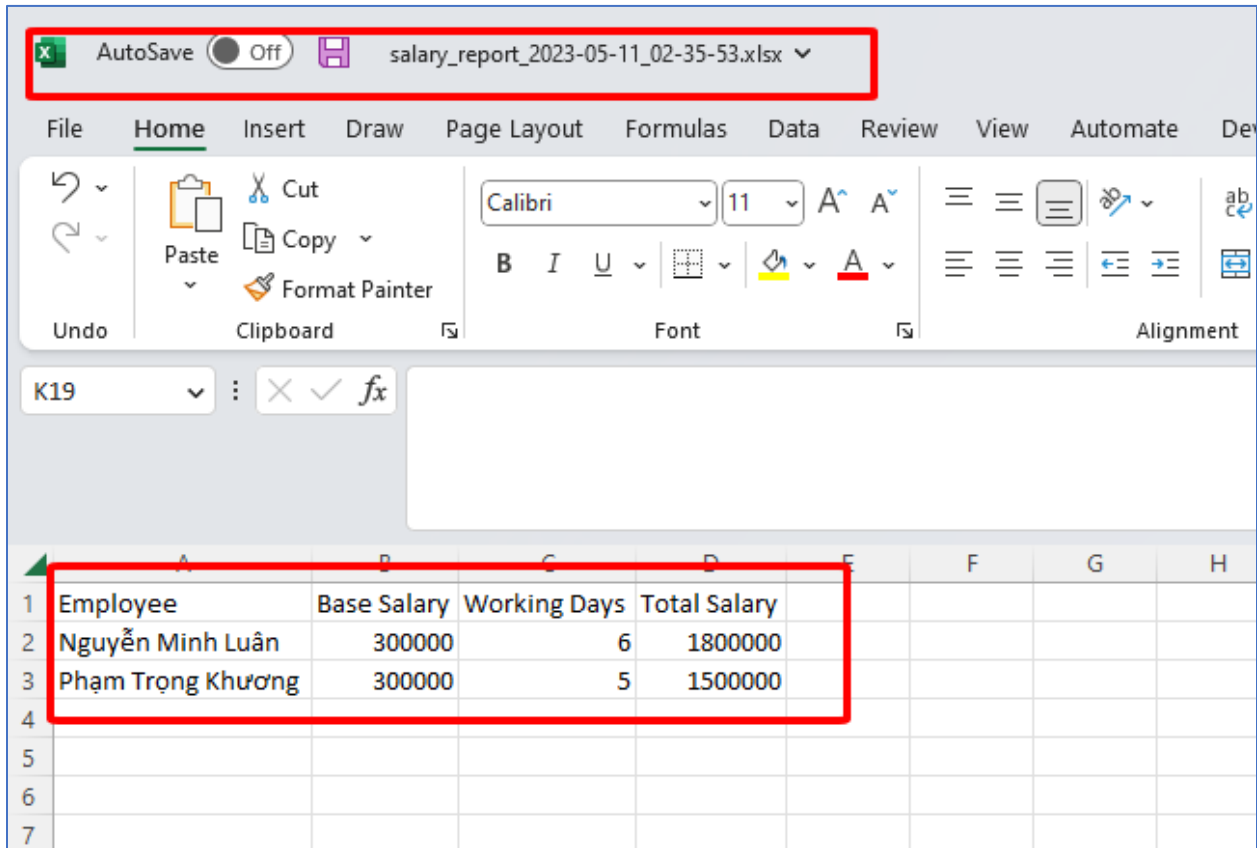
Hình 4.24 Chức năng quản lý chi chấm công nhân viên

Tích vào ô is_present nếu ngày hôm đó nhân viên đó đi làm để điểm danh sau đó tổng số ngày đi làm sẽ được tự động cộng lên và sau dựa dựa vào hai dữ kiện trên xuất ra bảng lương vào mỗi cuối tháng



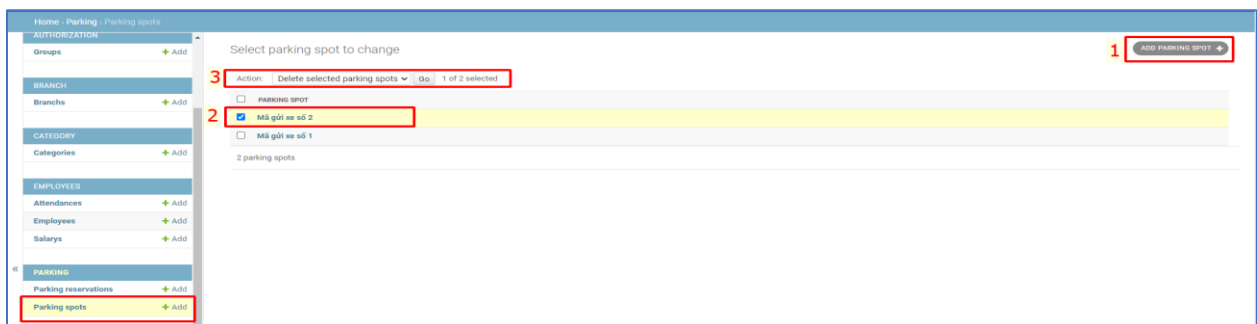
Hình 4.25 Chức năng xuất bảng lương nhân viên

Sau khi chọn nhân viên cần xuất bảng lương thì ngay action chọn xem bảng lương và nhấn “go” để xuất ra file excel thể hiện bảng lương của nhân viên.



Hình 4.26 Bảng lương nhân viên

Tiếp theo sẽ là quản lý chỗ đỗ xe và phiếu đỗ cũng tương tự cách quản lý các chức năng khác của trang thì chức năng chính trong dashboard là thêm sửa xóa các chỗ đỗ và các phiếu đỗ.



Hình 4.27 quản lý chỗ đỗ xe

1. Thêm chỗ đỗ mới ở nút này
2. Chỉnh sửa chỗ đỗ bằng cách chọn vào dòng chỗ đỗ muốn sửa thông tin
3. Chọn vào ô vuông trước chỗ đỗ muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa

Hình 4.28 quản lý chỉnh sửa chỗ đỗ xe

ở hình 4.28 là chức năng sửa chỗ đỗ xe khi chọn vào dòng chỗ đỗ để sửa thì sẽ chuyển qua trang như hình, sau đó khi chỉnh sửa xong thì nhấn save để lưu chỉnh sửa.

ID	USER	PARKING SPOT	RESERVATION TYPE	RESERVATION START TIME	RESERVATION END TIME	PAID	PAID DATE
21	ketn2304@gmail.com	Mã gửi xe số 2	Hourly	May 10, 2023, 2:38 a.m.	May 10, 2023, 3:38 a.m.	🟢	May 9, 2023, 10 p.m.
20	ketn2304@gmail.com	Mã gửi xe số 2	Hourly	May 10, 2023, 2:38 a.m.	May 10, 2023, 3:38 a.m.	🟢	May 9, 2023, 9:59 p.m.
19	admin@gmail.com	Mã gửi xe số 1	Daily	May 10, 2023, 2:29 a.m.	May 11, 2023, 2:29 a.m.	🟢	May 9, 2023, 7:30 p.m.
18	admin@gmail.com	Mã gửi xe số 2	Daily	May 8, 2023, 11:24 p.m.	May 9, 2023, 11:24 p.m.	🟢	May 8, 2023, 4:25 p.m.
17	admin@gmail.com	Mã gửi xe số 1	Daily	May 8, 2023, 11:24 p.m.	May 10, 2023, 11:24 p.m.	🟢	May 8, 2023, 4:24 p.m.
16	admin@gmail.com	Mã gửi xe số 1	Daily	May 8, 2023, 11:09 p.m.	May 9, 2023, 11:09 p.m.	🟢	May 8, 2023, 4:10 p.m.
15	admin@gmail.com	Mã gửi xe số 1	Daily	May 8, 2023, 11:09 p.m.	May 10, 2023, 11:09 p.m.	🟢	May 8, 2023, 4:09 p.m.
14	admin@gmail.com	Mã gửi xe số 1	Daily	May 8, 2023, 11:08 p.m.	May 9, 2023, 11:08 p.m.	🟢	May 8, 2023, 4:08 p.m.
13	admin@gmail.com	Mã gửi xe số 1	Hourly	May 7, 2023, 4:15 p.m.	May 7, 2023, 7:15 p.m.	🟢	May 8, 2023, 4:04 p.m.
12	admin@gmail.com	Mã gửi xe số 1	Hourly	May 7, 2023, 4:14 p.m.	May 7, 2023, 8:14 p.m.	🟢	May 7, 2023, 9:14 a.m.
11	admin@gmail.com	Mã gửi xe số 1	Hourly	May 7, 2023, 4:13 p.m.	May 7, 2023, 6:13 p.m.	🟢	May 7, 2023, 9:14 a.m.
10	admin@gmail.com	Mã gửi xe số 1	Hourly	May 7, 2023, 4:12 p.m.	May 7, 2023, 6:12 p.m.	🟢	May 7, 2023, 9:12 a.m.
9	admin@gmail.com	Mã gửi xe số 1	Hourly	May 7, 2023, 5:11 p.m.	May 7, 2023, 7:11 p.m.	🟢	May 7, 2023, 9:11 a.m.
8	admin@gmail.com	Mã gửi xe số 2	Daily	May 7, 2023, 4:08 p.m.	May 8, 2023, 4:08 p.m.	🟢	May 7, 2023, 9:08 a.m.
7	ketn2304@gmail.com	Mã gửi xe số 2	Daily	April 26, 2023, 5:58 p.m.	April 27, 2023, 5:58 p.m.	🟢	May 5, 2023, 1 a.m.
6	admin@gmail.com	Mã gửi xe số 1	Daily	April 26, 2023, 5:13 p.m.	April 27, 2023, 5:13 p.m.	🟢	May 5, 2023, 1 a.m.
5	admin@gmail.com	Mã gửi xe số 1	Hourly	April 26, 2023, 5:04 p.m.	April 26, 2023, 6:04 p.m.	🟢	April 26, 2023, 10:13 a.m.

Hình 4.29 quản lý phiếu đỗ

Có thể thêm, sửa, xóa tìm kiếm phiếu đỗ khi trong dashboard một cách đơn giản, với thao tác tương tự thêm, sửa, xóa các thành phần khác.

1. Thêm phiếu đỗ mới ở nút này.
2. Chỉnh sửa phiếu đỗ bằng cách chọn vào dòng phiếu đỗ muốn sửa thông tin.
3. Chọn vào ô vuông trước phiếu đỗ muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa.
4. Tìm kiếm phiếu đỗ.

Còn có thể biết được phiếu đỗ thanh toán hay chưa qua trạng thái và qua trường thanh toán và ngày thanh toán ở trang chi tiết phiếu đỗ.

The screenshot shows a web interface for managing parking reservations. The main content area is titled 'Change parking reservation' and displays the following information:

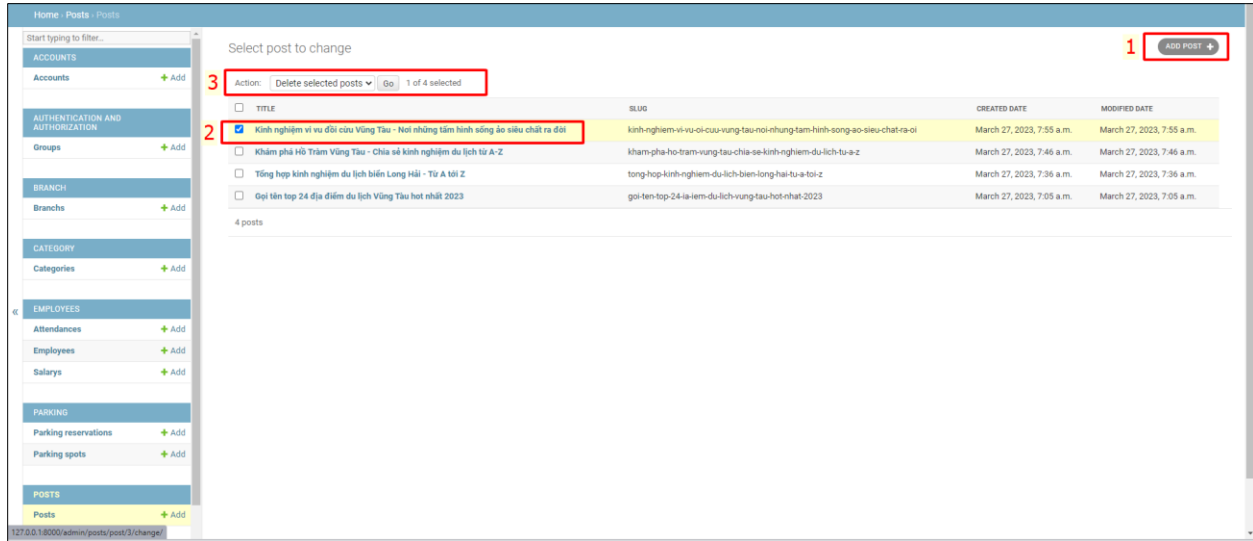
- User: kietnt2304@gmail.com
- Parking spot: Mã gửi xe số 2
- Reservation type: Hourly
- Reservation start time: Date: 2023-05-10, Time: 02:38:00
- Reservation end time: Date: 2023-05-10, Time: 03:38:00
- Paid: Paid
- Paid date: Date: 2023-05-09, Time: 22:00:10

At the bottom of the form, there are three buttons: 'Delete', 'Save and add another', and 'Save and continue editing'. The 'Save and continue editing' button is highlighted with a red box.

Hình 4.30 quản lý chi sửa phiếu đỗ và thanh toán

Khi thanh toán thì sẽ chọn vào nút paid và chọn ngày thanh toán sau đó save lại để lưu thông tin, hoặc có thể chỉnh sửa phiếu đỗ thẳng trên trang này và sau đó lưu lại.

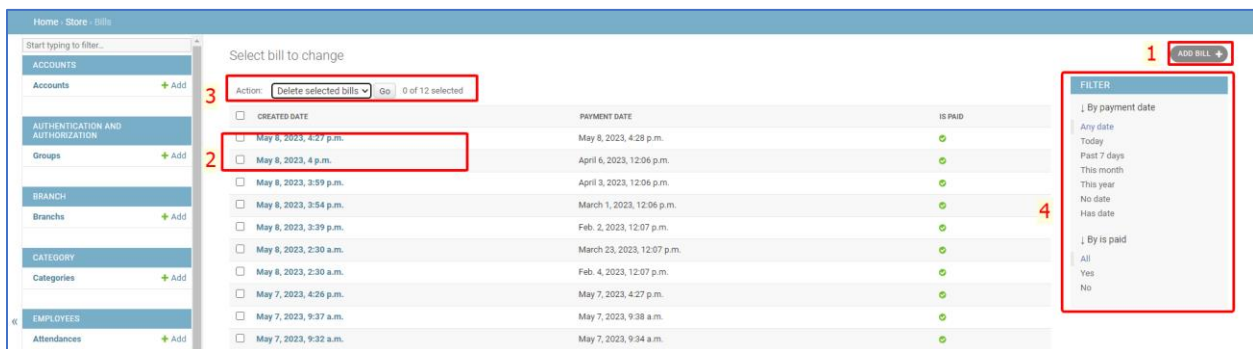
Tiếp theo là quản lý bài viết cửa website, với chức năng này chỉ có admin mới tạo ra nhưng bài viết tĩnh và sẽ được admin cập nhật liên tục, tạo mới liên tục để tạo độ phong phú cho trang của cửa hàng, cùng với các chức năng như trên không sai biệt lắm.



Hình 4.31 Quản lý bài viết

1. Thêm bài viết mới ở nút này.
2. Chỉnh sửa bài viết bằng cách chọn vào dòng bài viết muốn sửa thông tin.
3. Chọn vào ô vuông trước bài viết muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa.

Quản lý hóa đơn và chi tiết hóa đơn là một chức năng vô cùng quan trọng cũng như dựa vào hóa đơn có thể tính được thống kê doanh số theo năm và theo loại xe từ đó xây dựng chiến lược marketing cũng như nên phổ biến các loại xe gì.

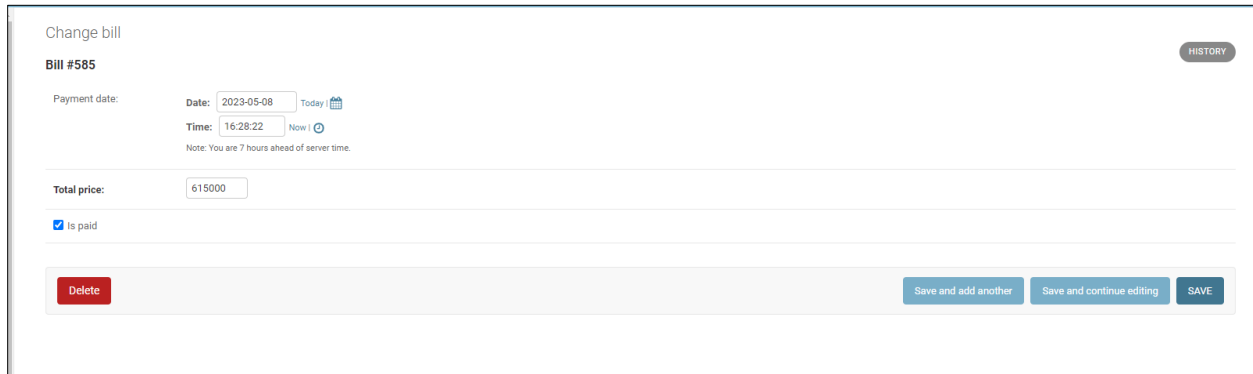


Hình 4.32 Quản lý hóa đơn

1. Thêm hóa đơn mới ở nút này.
2. Chỉnh sửa hóa đơn bằng cách chọn vào dòng hóa đơn muốn sửa thông tin.

3. Chọn vào ô vuông trước hóa đơn muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa.
4. Chọn bộ lọc theo ý muốn

Dựa vào hình 4.33 có thể thấy được ngày thanh toán của hóa đơn cũng như tổng tiền trên hóa đơn và hóa đơn đã được thanh toán hay chưa.



Hình 4.33 Quản lý chỉnh sửa hóa đơn

Tiếp theo là chi tiết hóa đơn để có thể thấy được hóa đơn này là của xe nào và được giá đơn là bao nhiêu tổng giá là bao nhiêu.

The screenshot shows a table titled 'Select bill item to change'. The table has columns for ID, BILL, PRODUCT, TOTAL PRICE, and RENTAL PRICE. Item 584 is highlighted in yellow. The table also includes an 'Action' dropdown menu set to 'Delete selected bill items' and a 'Go' button. A red box highlights the 'ADD BILL ITEM +' button in the top right corner.

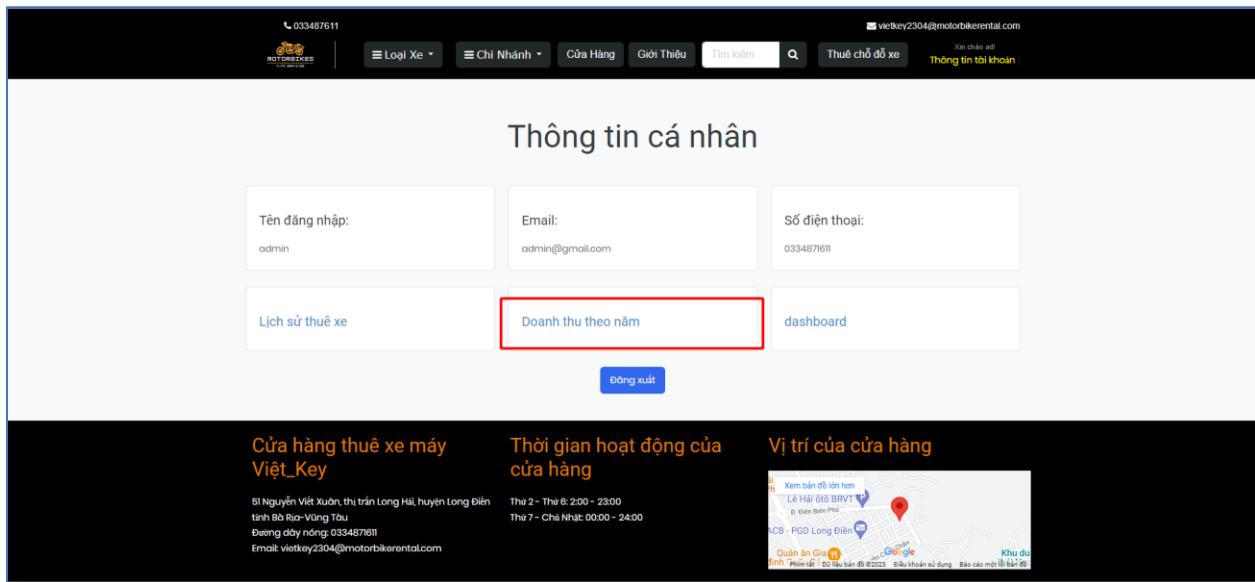
ID	BILL	PRODUCT	TOTAL PRICE	RENTAL PRICE	
<input checked="" type="checkbox"/>	584	Bill #585	Future 125 (72K1-123458)	615000	150000
<input type="checkbox"/>	583	Bill #584	SH 150 (S1-72K1-12345)	525000	250000
<input type="checkbox"/>	582	Bill #583	SH 150 (S1-72K1-12345)	500000	250000
<input type="checkbox"/>	581	Bill #582	WINNER (W1-72K1-12347)	420000	200000
<input type="checkbox"/>	580	Bill #581	SIRUS (S1-72K1-12346)	300000	150000
<input type="checkbox"/>	579	Bill #580	SH 150 (S1-72K1-12345)	250000	250000
<input type="checkbox"/>	578	Bill #579	WINNER (W1-72K1-12347)	420000	200000
<input type="checkbox"/>	577	Bill #578	SH 150 (S1-72K1-12345)	500000	250000
<input type="checkbox"/>	576	Bill #577	SH 150 (S1-72K1-12345)	250000	250000
<input type="checkbox"/>	575	Bill #576	SH 150 (S1-72K1-12345)	3550000	250000
<input type="checkbox"/>	574	Bill #575	WINNER (W1-72K1-12347)	200000	200000
<input type="checkbox"/>	573	Bill #574	WINNER (W1-72K1-12347)	200000	200000

Hình 4.34 Quản lý chỉnh sửa chi tiết hóa đơn

Cũng gồm có các chức năng tương tự là thêm, sửa, xóa chi tiết hóa đơn.

1. Thêm chi tiết hóa đơn mới ở nút này.
2. Chỉnh sửa chi tiết hóa đơn bằng cách chọn vào dòng chi tiết hóa đơn muốn sửa thông tin.
3. Chọn vào ô vuông trước chi tiết hóa đơn muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa.

Ngoài ra còn có thể xem được doanh số thống kê dựa trên hóa đơn được thống kê dựa theo tháng, năm, và loại sản phẩm.



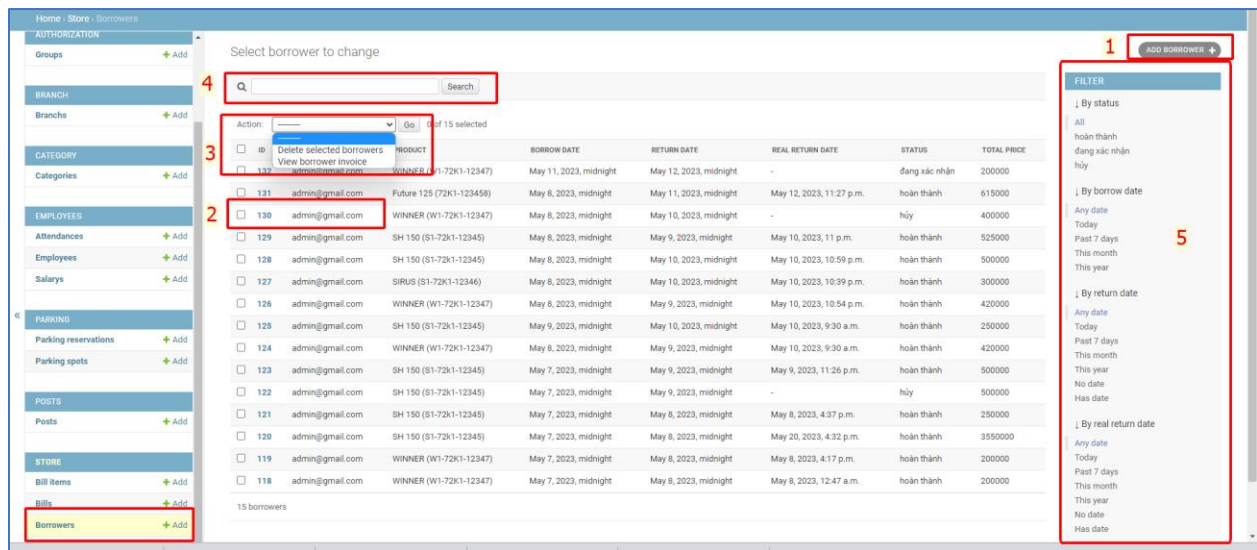
Hình 4.35 Xem doanh thu theo năm



Hình 4.36 Xem doanh thu theo năm 2023 – loại xe xe tay côn

Có thể thấy được doanh số trong năm 2023 xe tay côn rất cao vào tháng 5 từ đó có thể xây dựng chiến lược phù hợp để giúp phát triển doanh thu của cửa hàng.

Tiếp theo đây chính là chức năng chính cũng là chức năng mang lại doanh thu cho cửa hàng chính là thuê xe máy và để quản lý thuê xe máy thì cần phải quản lý phiếu mượn của cửa hàng.



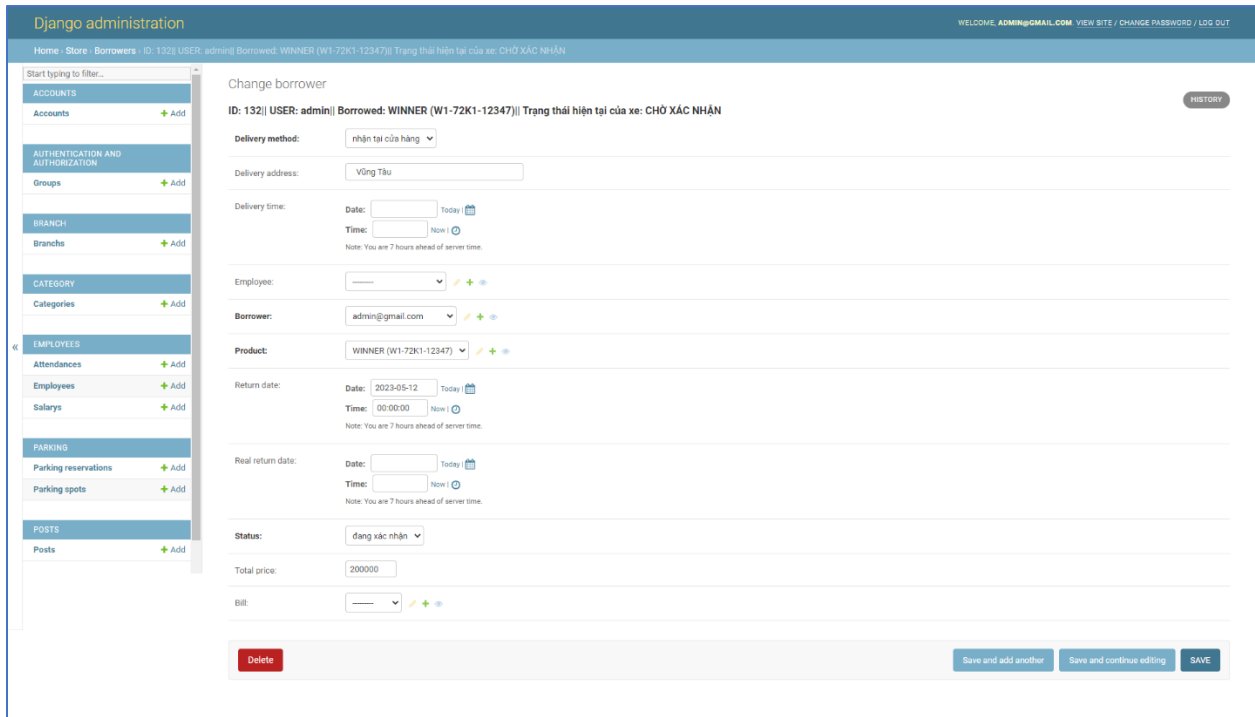
Hình 4.37 Quản lý phiếu mượn

Quản lý phiếu mượn cần quản lý tất cả các thành phần nên việc có thêm chức năng lọc và tìm kiếm là vô cùng cần thiết dựa vào phiếu mượn sẽ kiểm soát được xe đang được mượn đang ở trạng thái gì, phương thức giao, người giao, biên bản giao, hóa đơn của phiếu....,

1. Thêm phiếu mượn mới ở nút này.
2. Chỉnh sửa phiếu mượn bằng cách chọn vào dòng phiếu mượn muốn sửa thông tin.
3. Chọn vào ô vuông trước phiếu mượn muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa.
4. Tìm kiếm phiếu mượn mà muốn tìm theo id, xe,...

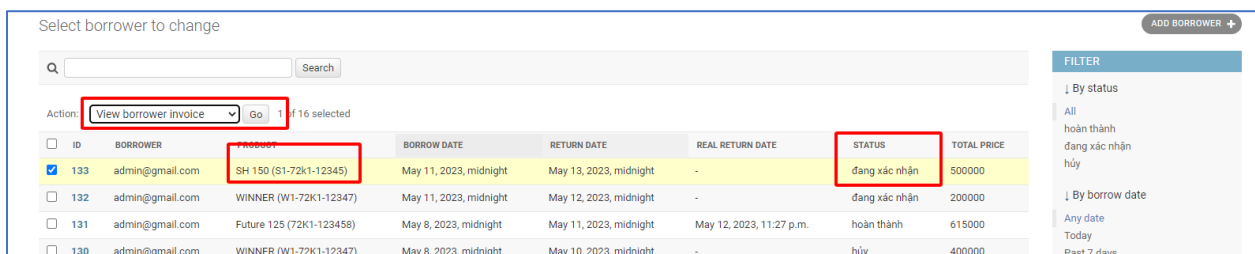
5. Lọc phiếu mượn theo nhu cầu theo trạng thái,..

Đối với chi tiết phiếu mượn thì bao gồm đầy đủ thông tin để chủ cửa hàng, admin nắm rõ và quản lý xe cho mượn một cách dễ dàng hợp lý và chặt chẽ.



Hình 4.38 Quản lý chi tiết phiếu mượn

Với chức năng xuất biên bản cho mượn xe để ký hợp đồng phù hợp với pháp luật cũng như bảo vệ tài sản thì chức năng này vô cùng cần thiết và được thiết lập thẳng trong dashboard, chỉ những phiếu đang xác nhận chưa giao xe cho khách hàng mới in được biên bản.



Hình 4.39 Chức năng in biên bản mượn xe

BIÊN BẢN GIAO NHẬN XE MÁY

Thông tin xe máy	
Tên xe:	SH 150 (S1-72k1-12345)
Loại xe:	Xe tay ga
Chi Nhánh:	Châu Đức
Giá xe một ngày:	250000
Thông tin người giao nhận	
Họ và tên:	Admin
Số điện thoại:	0334871611
Thông tin người nhận	
Họ và tên:	admin@gmail.com
Số điện thoại:	0334871611
Ngày thuê xe:	2023-05-11
Ngày trả xe:	2023-05-13
Thông tin thời gian và địa điểm	
Thời gian giao nhận:	None
Địa điểm giao nhận:	54 Nguyễn Văn Trỗi Khu phố 1 Châu Đức

Biên bản nhận xe này được thống nhất từ cả hai phía giữa bên cho thuê xe (Cửa hàng thuê xe Việt_Key) và bên thuê xe admin@gmail.com. Bên thuê phải đồng ý chấp thuận các điều khoản thuê xe như sau :

- + Có giấy phép lái xe (A1)
- + Cung cấp Căn Cước Công Dân cho bên cho thuê xe
- + Thanh toán trước khi thuê xe
- + Nếu có vấn đề liên quan đến giao thông trong quá trình trải nghiệm xe thì báo ngay cho bên cho thuê xe qua số điện thoại 0334871611 hoặc gửi mail: vietkey2304@motorbikerental.com

Xe được giao bởi nhân viên: Admin, số điện thoại: 0334871611.

Người đại diện bên cho thuê

(Ký và ghi rõ họ tên)

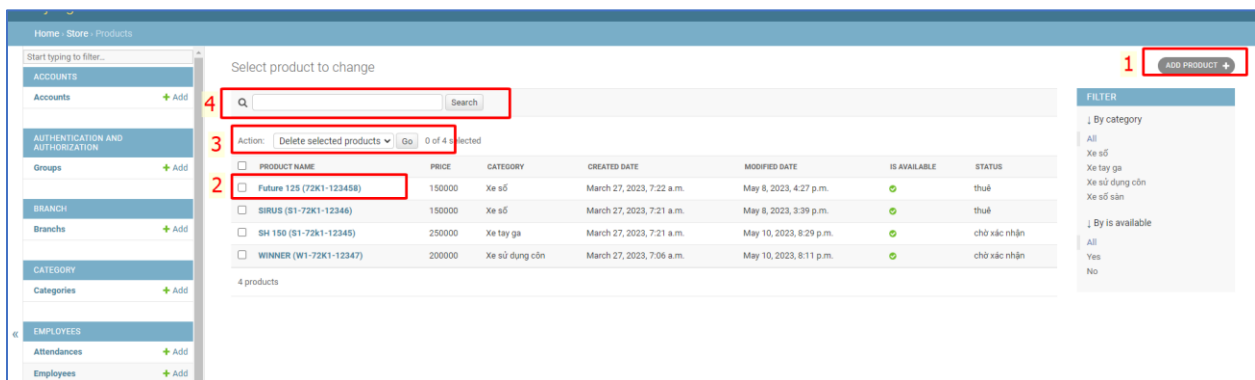
Người đại diện bên thuê

(Ký và ghi rõ họ tên)

1:800D/store/review_borrow/133/

Hình 4.40 Biên bản mượn xe

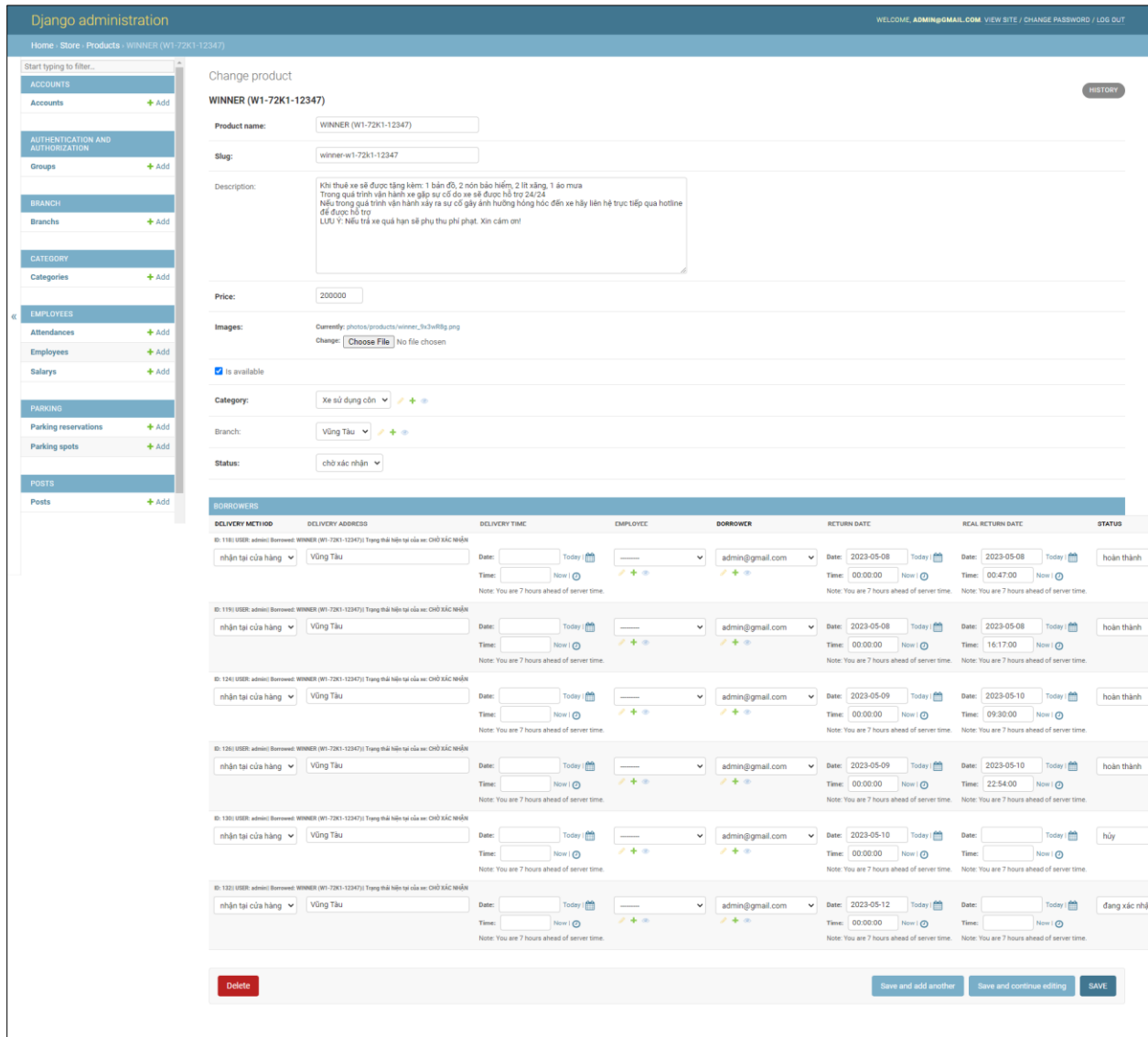
Quản lý xe cũng tương tự như cách quản lý các thành phần trên xe là sản phẩm được chú trọng nhất cửa hàng nên thông tin luôn đầy đủ và trực quan nhất.



Hình 4.41 Quản lý sản phẩm

1. Thêm xe ở nút này
2. Chỉnh sửa xe bằng cách chọn vào dòng xe muốn sửa
3. Chọn vào ô vuông trước xe muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa
4. Để tìm kiếm xe nhận category_name vào ô và nhấn search

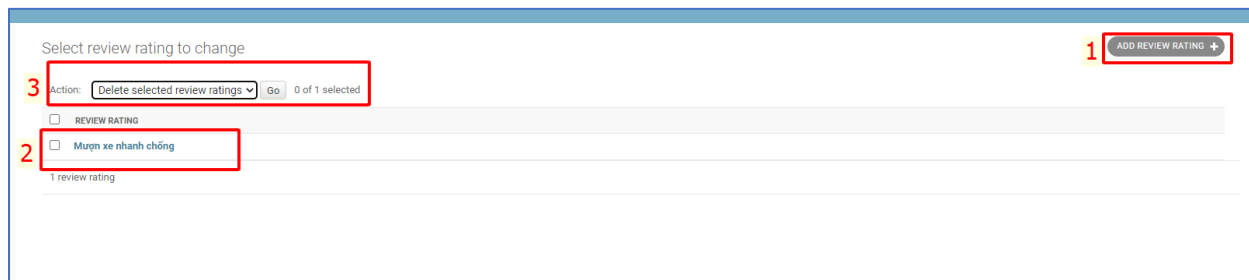
Không chỉ hiển thị xe mà còn hiển thị tất cả các phiếu mượn liên quan đến xe đó giúp admin, chủ cửa hàng dễ dàng quản lý và sử dụng tiện lợi.



Hình 4.42 Quản lý chi tiết sản phẩm

Có thể chỉnh sửa thông tin sản phẩm hoặc phiếu mượn ngày trên màn hình ở hình 4.42 sau khi chỉnh sửa nhấn save để lưu lại.

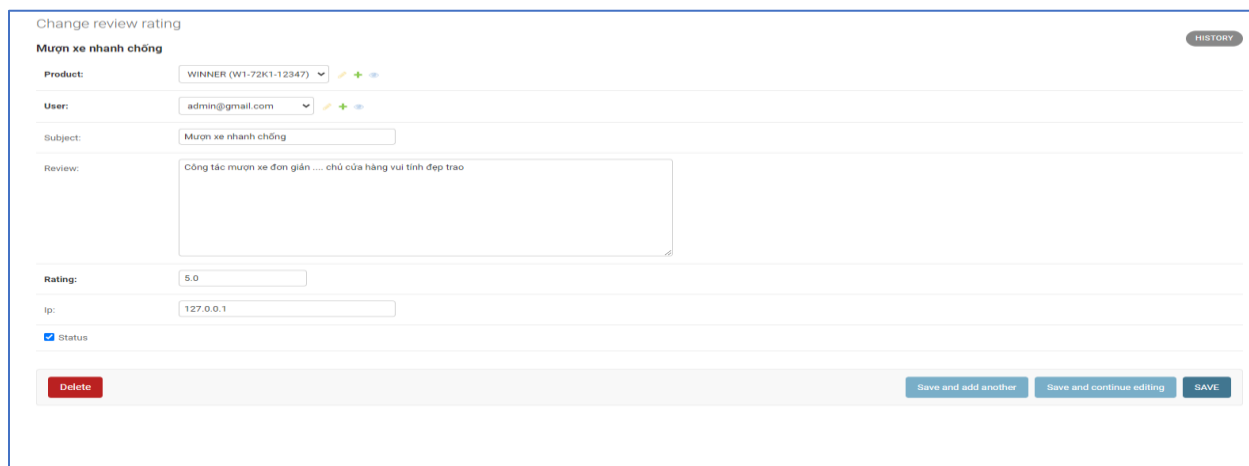
Cuối cùng là thành phần cuối cùng mà dashboard quản lý đó chính là đánh giá sản phẩm. Cũng gồm ba chức năng chính là thêm, sửa, xóa đánh giá.



Hình 4.43 Quản lý đánh giá

1. Thêm đánh giá ở nút này
2. Chỉnh sửa đánh giá bằng cách chọn vào dòng đánh giá muốn sửa
3. Chọn vào ô vuông trước đánh giá muốn xóa chọn chức năng xóa ở ô action sau đó nhấn go để xóa
4. Để tìm kiếm đánh giá nhận category_name vào ô và nhấn search

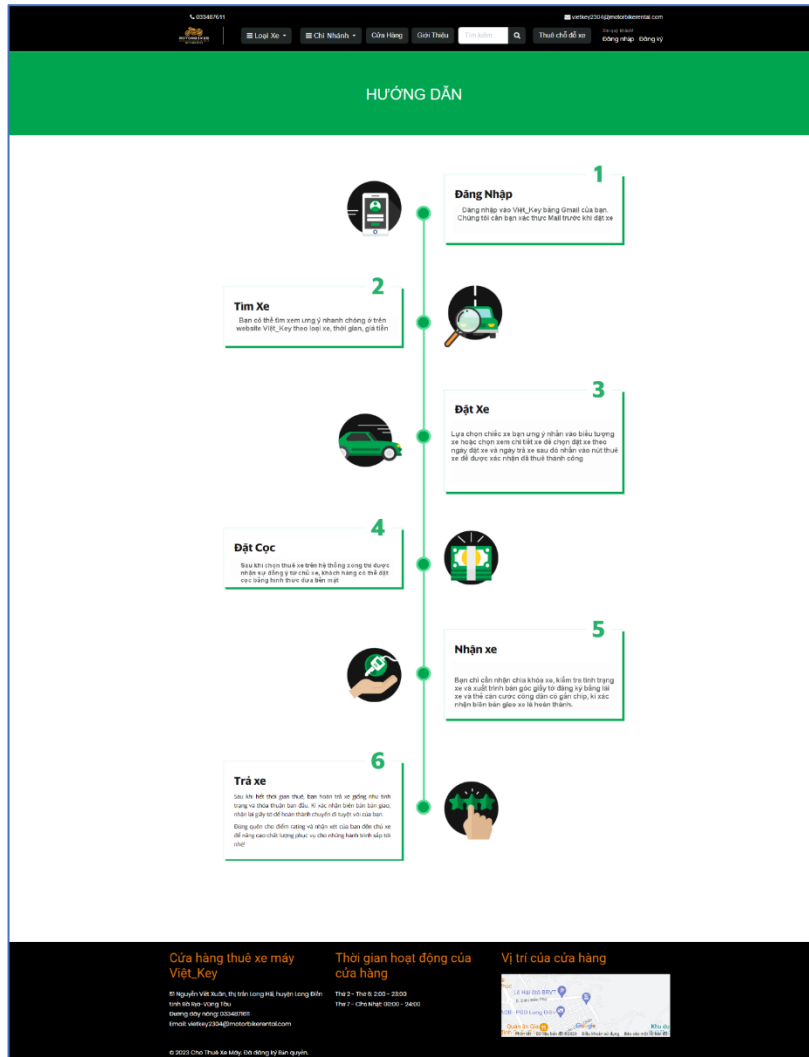
Chỉnh sửa đánh giá cũng như các thành phần trước chọn đánh giá muốn chỉnh sửa và sau khi chỉnh sửa xong sau đó nhấn nút save để lưu lại



Hình 4.44 Quản lý chỉnh sửa đánh giá

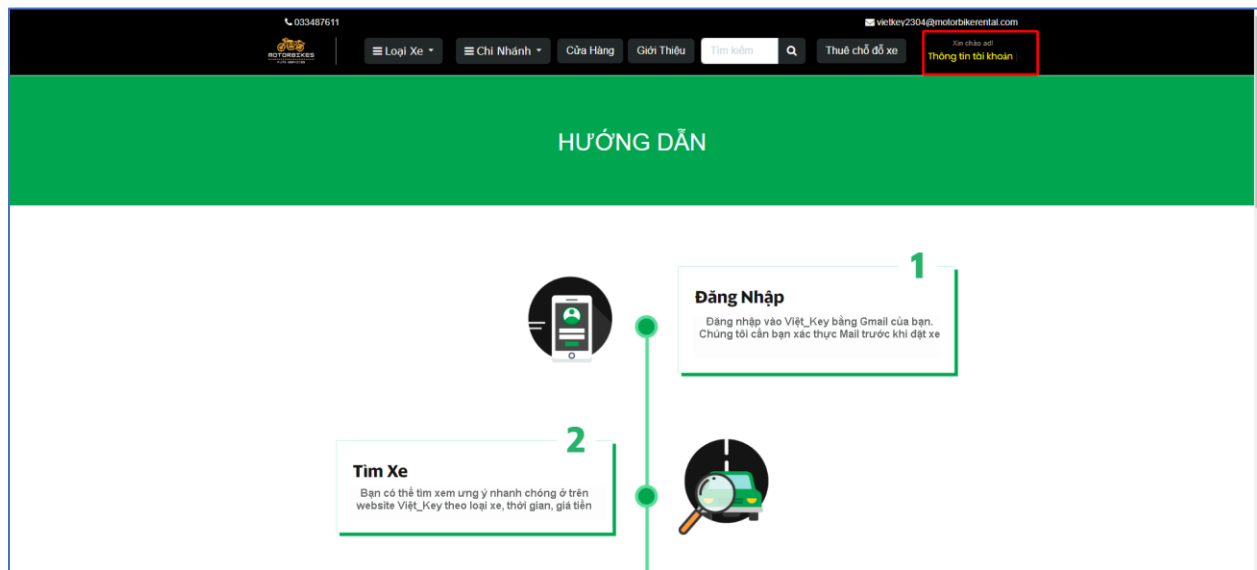
4.3 Chức năng của user

Chức năng của khách hàng có tài khoản sẽ bao gồm tất cả các chức năng của khách ở trên như xem website, quy trình cho mượn xe.



Hình 4.45 Trang hướng dẫn thuê xe

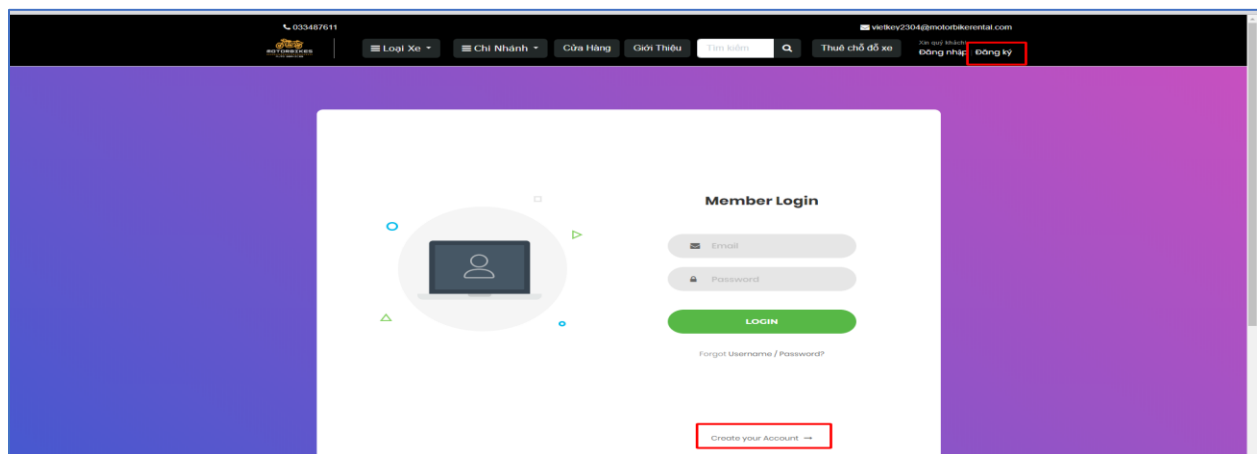
Giao diện sẽ khác đôi chút ở phần header.



Hình 4.46 header khi đăng nhập

Để thực hiện các chức năng thuê xe, thuê chỗ đỗ, đánh giá, xem lịch sử thuê xe thì cần phải đăng nhập. Để đăng nhập vào hệ thống thì cần phải có tài khoản, vì vậy cần thực hiện chức năng đăng ký trước tiên khi muốn thuê xe.

Để vào trang đăng ký có thể chọn thẳng trên thanh menu navbar hoặc trong trang đăng nhập có phần tạo tài khoản.



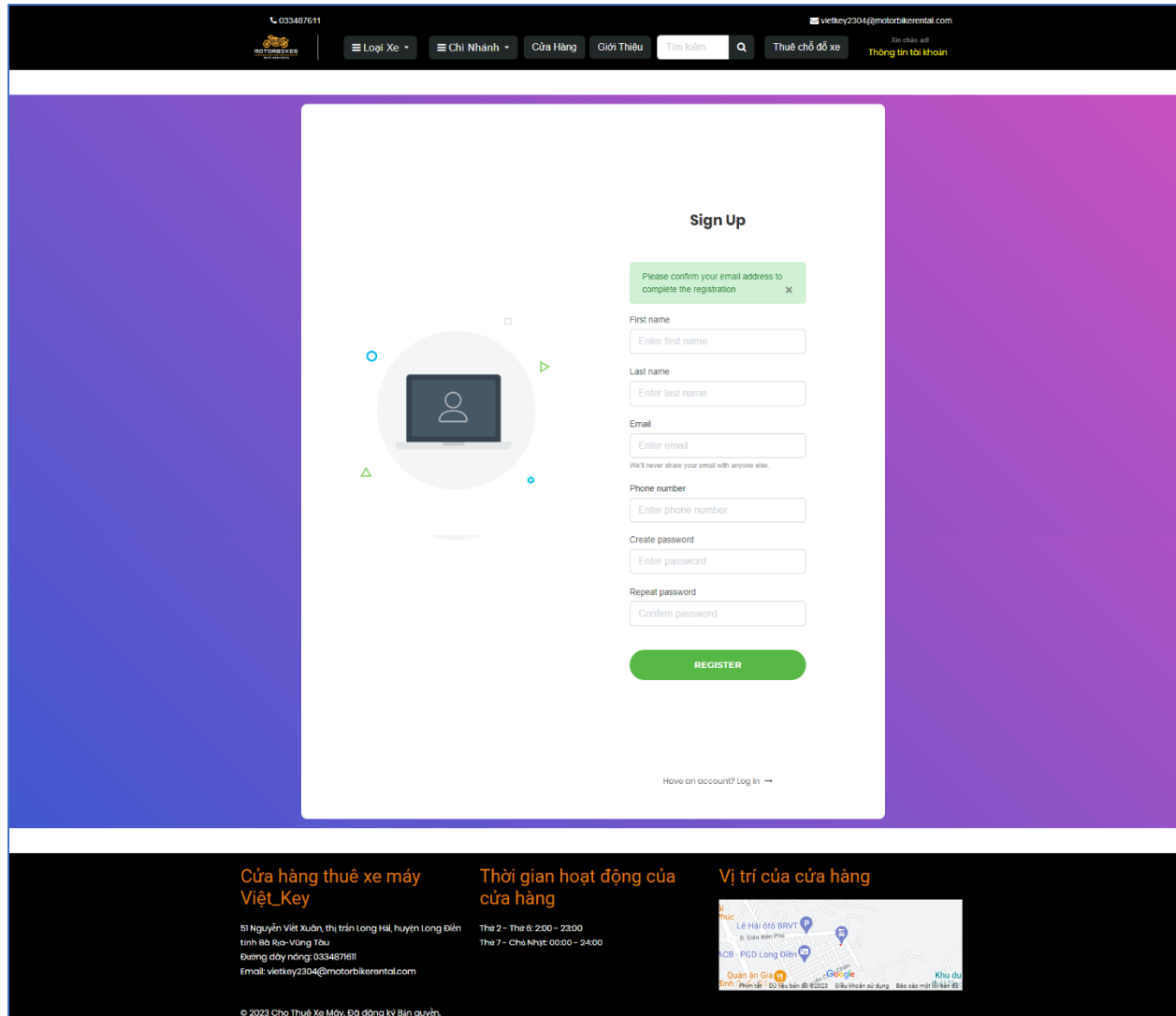
Hình 4.47 Trang đăng nhập

Để đăng ký cần phải nhập đầy đủ các trường bắt buộc, ngoài ra email đăng ký phải là email khả dụng để có thể kích hoạt khi đăng ký nếu không đúng yêu cầu sẽ xuất hiện thông báo.

The screenshot shows a web browser window with the URL `vietkey2304@motorbikerental.com`. The page features a navigation bar with links for 'Loại Xe', 'Chi Nhánh', 'Cửa Hàng', 'Giới Thiệu', 'Thuê chỗ để xe', 'Đăng nhập', and 'Đăng ký'. The main content area is a 'Sign Up' form with the following fields: 'First name' (placeholder: 'Enter first name'), 'Last name' (placeholder: 'Enter last name', with a red error message 'Please fill out this field.'), 'Email' (placeholder: 'Enter email'), 'Phone number' (placeholder: 'Enter phone number'), 'Create password' (placeholder: 'Enter password'), and 'Repeat password' (placeholder: 'Confirm password'). A green 'REGISTER' button is at the bottom of the form. Below the form, there is a link: 'Have an account? Log in'. The footer contains contact information for 'Cửa hàng thuê xe máy Việt_Key' at 'Số Nguyễn Văn Xuân, thị trấn Long Hải, huyện Long Điền, tỉnh Bà Rịa- Vũng Tàu', phone '033487811', and email 'vietkey2304@motorbikerental.com'. It also lists 'Thời gian hoạt động của cửa hàng' (Thu 2 - Thu 6: 200 - 2300, Thu 7 - Chủ Nhật: 00:00 - 24:00) and 'Vị trí của cửa hàng' with a map showing the location in Long Hải, Bà Rịa-Vũng Tàu.

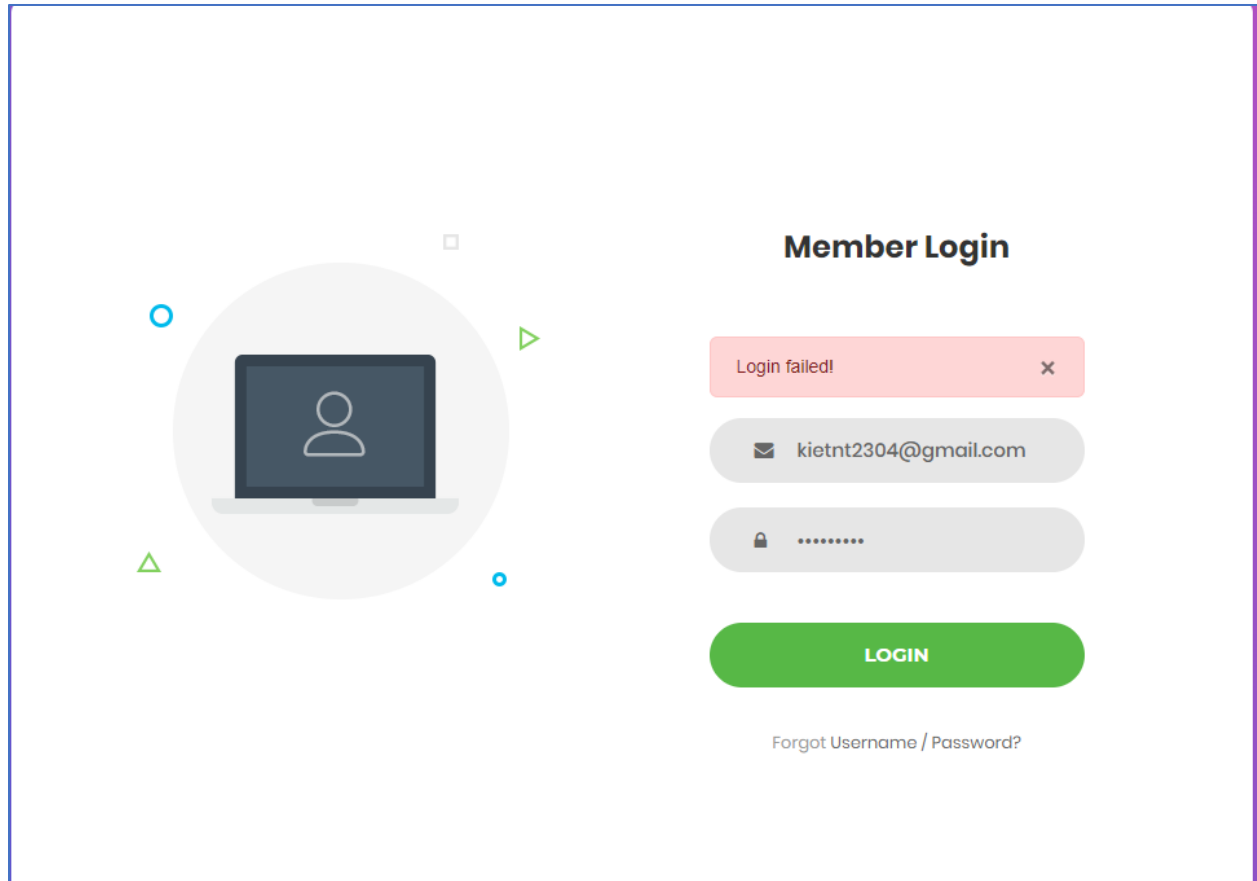
Hình 4.48 Đăng ký thất bại cần nhập trường gmail

Nếu đăng ký thành công thì sẽ xuất hiện hình như 4.49 sau.



Hình 4.49 Trang đăng ký thành công vào gmail để kích hoạt tài khoản

Cần phải vào mail vừa đăng ký để kích hoạt tài khoản nếu không kích hoạt dù đăng nhập vẫn không thể vào hệ thống.



Hình 4.50 Đăng nhập thất bại do chưa kích hoạt tài khoản

Lúc này trong dashboard vẫn sẽ nhìn thấy tài khoản trường kích hoạt chưa được chọn.

Select account to change ADD ACCOUNT +

Q [] Search

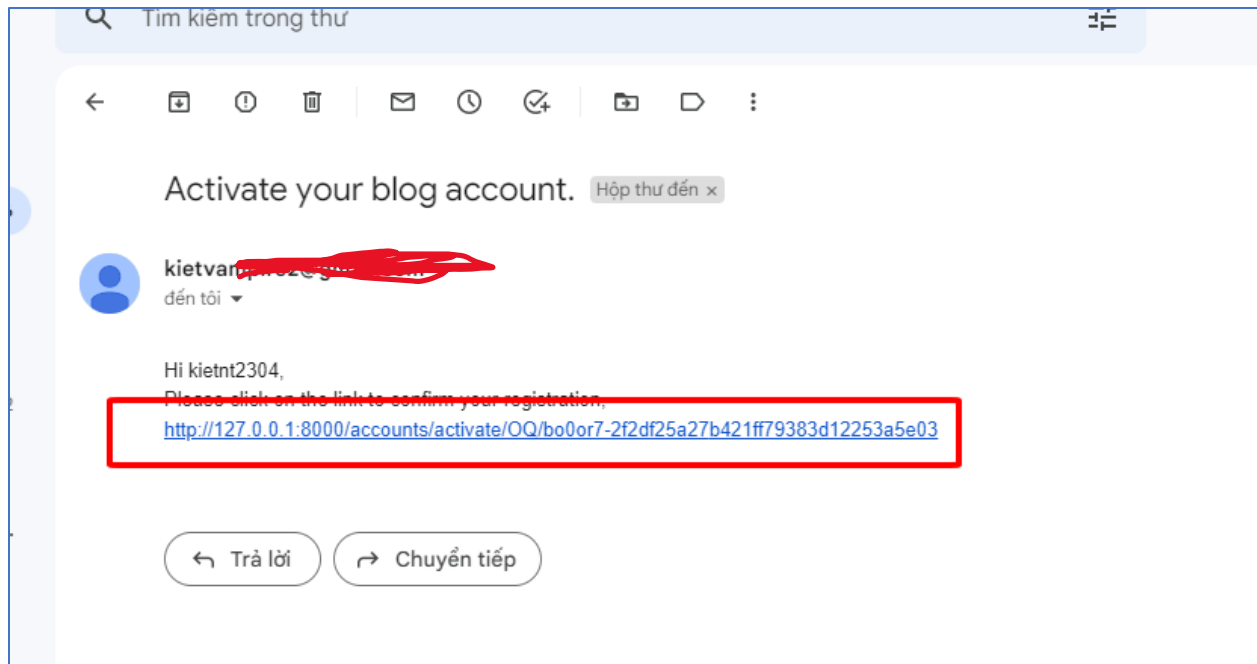
Action: [] Go 0 of 2 selected

<input type="checkbox"/>	EMAIL	USERNAME	FIRST NAME	LAST NAME	LAST LOGIN	DATE JOINED	IS ACTIVE	PHONE NUMBER
<input type="checkbox"/>	kietnt2304@gmail.com	kietnt2304	Kiệt	Nguyễn	May 10, 2023, 9:06 p.m.	May 10, 2023, 9:06 p.m.	<input type="radio"/>	+10334871611
<input type="checkbox"/>	admin@gmail.com	admin	ad	min	May 10, 2023, 9:10 p.m.	April 3, 2023, 4:10 a.m.	<input checked="" type="radio"/>	0334871611

2 accounts

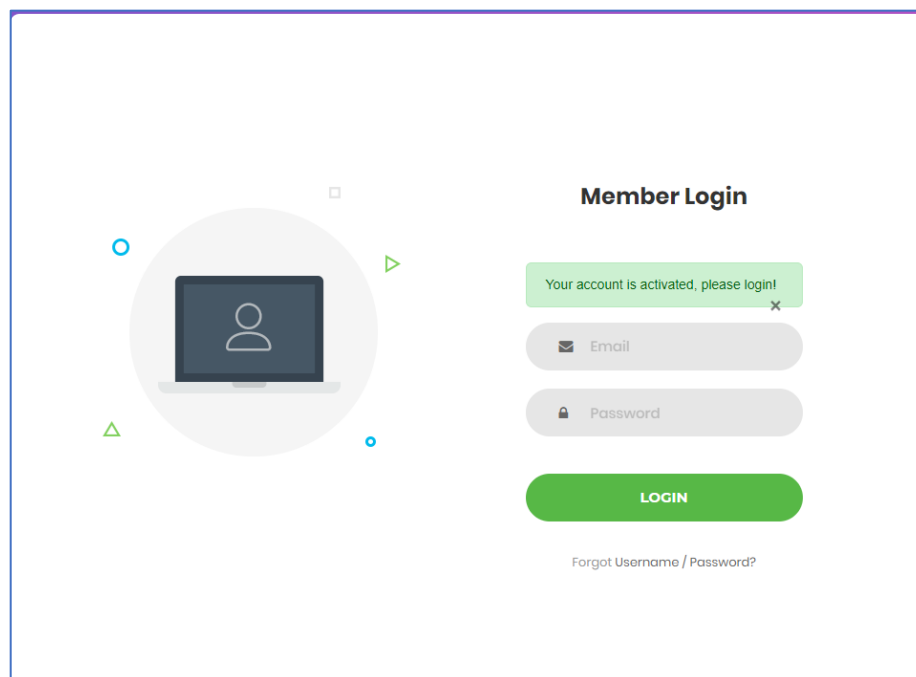
Hình 4.51 chưa kích hoạt tài khoản

Cần phải vào mail để kích hoạt tài khoản nhấn vào đường link trong mail là có thể kích hoạt.



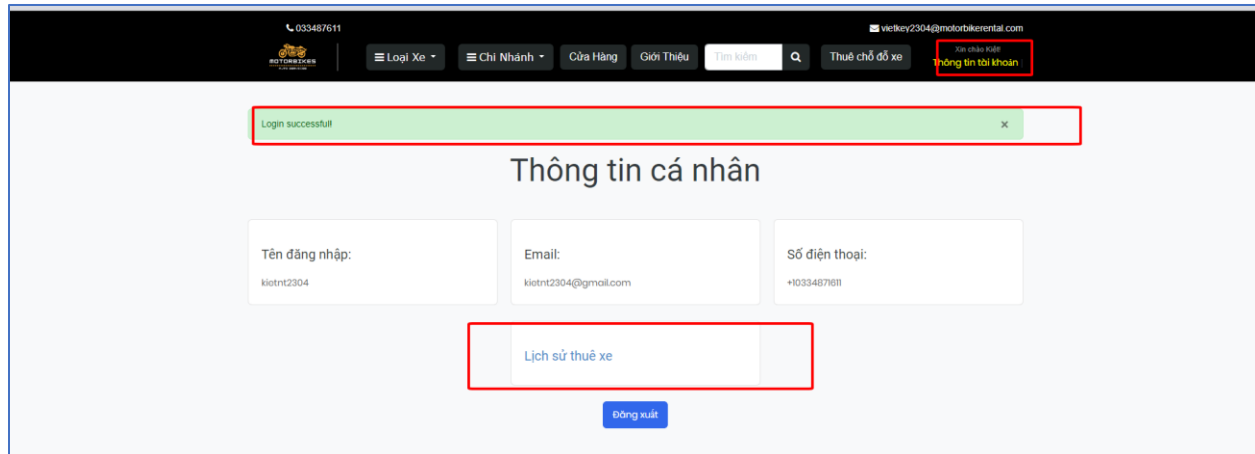
Hình 4.52 link kích hoạt tài khoản

Khi kích hoạt thành công sẽ dẫn đến trang đăng nhập như sau:



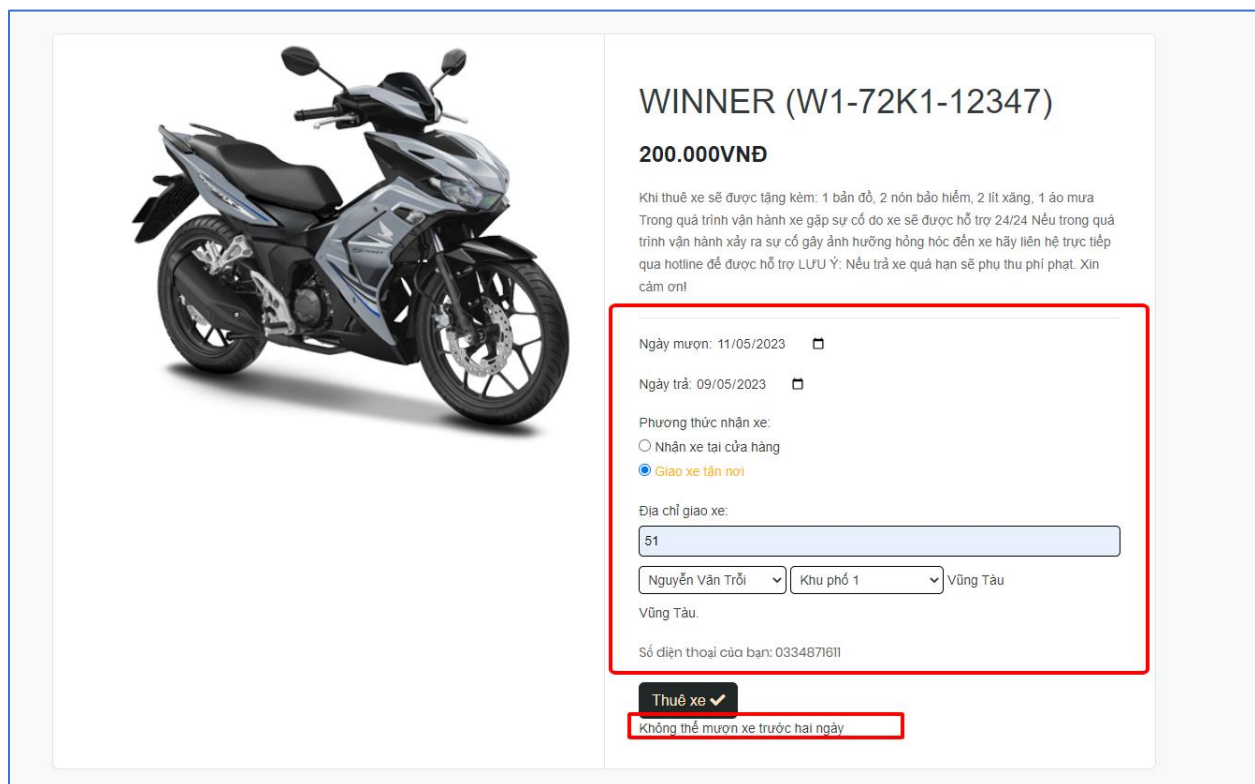
Hình 4.53 Tài khoản đã được kích hoạt

Tiếp theo sau khi đăng nhập thành công sẽ chuyển hướng tới thông tin tài khoản vừa đăng nhập.



Hình 4.54 Đăng nhập vào hệ thống

Sau khi đăng nhập thành công thì bây giờ có thể thực hiện chức năng thuê xe và thuê chỗ.



Hình 4.55 Thực hiện thao tác thuê xe

Khi thuê xe cần nhập các trường yêu cầu cần thiết để thuê xe như: ngày mượn, ngày trả, phương thức thuê xe, nếu là giao xe tận nơi cần phải chọn địa chỉ, lưu ý thuê xe không

thể đặt trước quá hai ngày, hay ngày trả không thể nhỏ hơn ngày mượn nếu không sẽ xuất hiện thông báo như trên hình 4.55.

Sau khi thêm đầy đủ các trường và nhấn thuê xe thì trạng thái xe sẽ chuyển thành đang chờ xác nhận và gửi mail về cho chủ cửa hàng để xác nhận lại.

The screenshot shows a web interface for a motorcycle rental service. The main heading is "Thông tin giao hàng" (Delivery Information). Below it is a table with the following data:

Sản phẩm	WINNER (W1-72K1-12347)
Tên người mượn	admin
Số điện thoại người mượn	0334871611
Ngày mượn	May 11, 2023, midnight
Ngày trả	May 12, 2023, midnight
Phương thức nhận hàng	GIAO XE TẬN NƠI
Địa chỉ giao	51 Nguyễn Tri Phương Khu phố 1 Vũng Tàu
Giá thuê	200000.0
Tổng giá	200000.0

Below the table is a section titled "Thông tin nhân viên giao hàng" (Delivery Staff Information) with two input fields: "Người giao xe" and "Điện thoại người giao xe". At the bottom right, there is a blue button labeled "Quay lại cửa hàng" (Return to store).

Hình 4.56 Thông tin khi thuê xe

The screenshot shows an email notification with the subject "[Thông báo thuê sản phẩm] WINNER (W1-72K1-12347)". The email content is as follows:

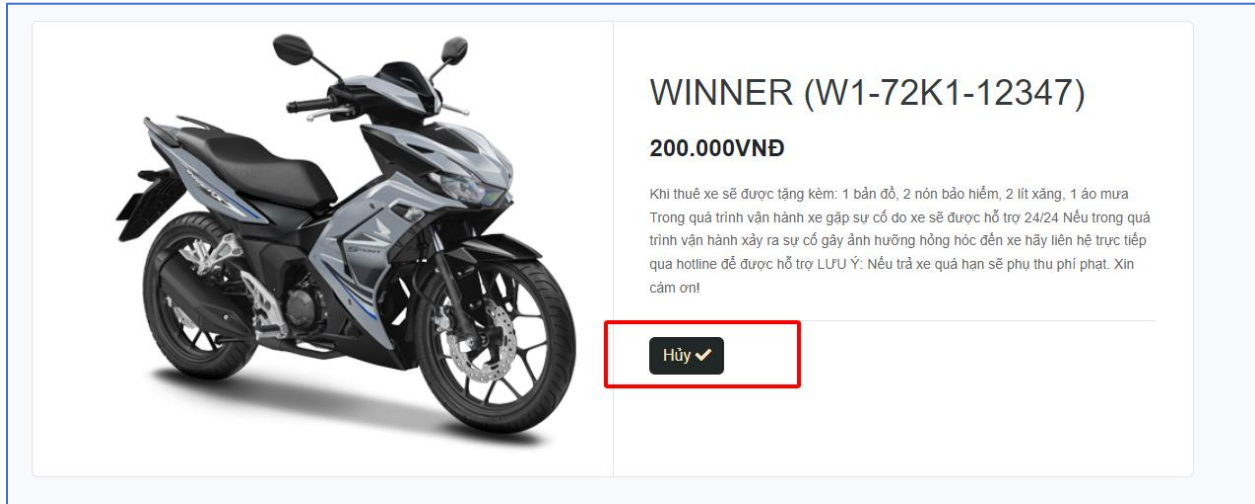
Khách hàng admin admin@gmail.com , đã thuê sản phẩm WINNER (W1-72K1-12347). Thông tin chi tiết như sau:

Số điện thoại: 0334871611
 Ngày mượn: 2023-05-11 00:00:00
 Ngày trả: 2023-05-12 00:00:00
 Phương thức nhận hàng: GIAO XE TẬN NƠI
 Địa chỉ giao xe: 51 Nguyễn Tri Phương Khu phố 1 Vũng Tàu
 Nhân viên giao xe: Admin\Số điện thoại Nhân viên giao xe: 033487611
 Giá thuê: 200000.0

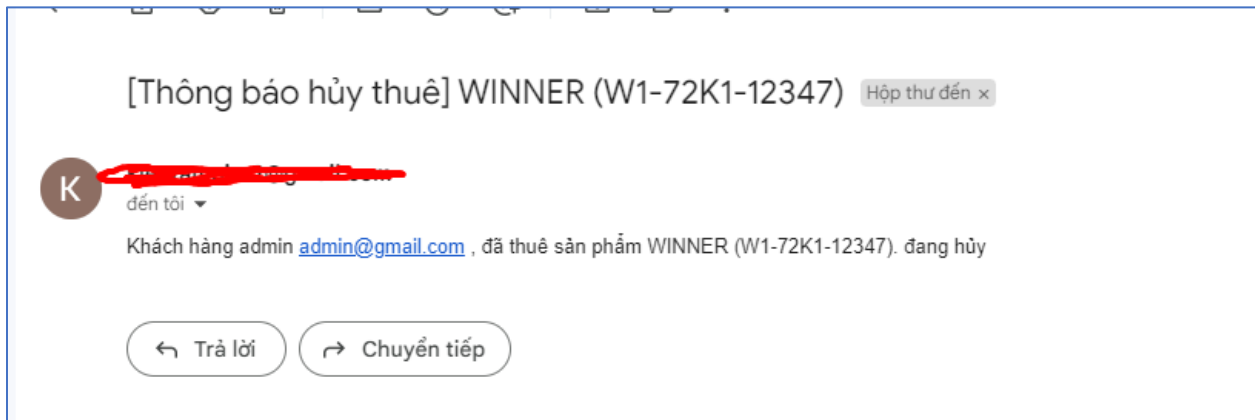
Hình 4.57 Mail chủ cửa hàng nhận được

Bây giờ chủ cửa hàng cần thực hiện là dựa vào mail được nhận vào xét lại trạng thái và lập biên bản giao xe và giao xe cho khách hàng.

Trước khi nhận xe thì khách hàng được phép hủy nếu xe chưa được giao tới bằng cách nhấn nút hủy sau khi nhấn nút hủy thì sẽ gửi mail đến chủ cửa hàng để xác nhận thông tin hủy thuê.



Hình 4.58 Hủy thuê xe



Hình 4.59 Mail hủy từ chủ cửa hàng nhận được

Còn nếu nhận được xe và trả nghiệm thì khi trả xe khách hàng sẽ tới trả xe hoặc liên hệ đem xe về khi đem xe về thì chủ cửa hàng sẽ trả xe về kho để đảm bảo xe trả về phải ở trong kho, còn khách hàng không thể tùy ý trả xe về kho.

Chủ cửa hàng sẽ trả về kho và in ra hóa đơn:

Trả xe - WINNER (W1-72K1-12347)

Tổng số tiền bạn phải trả khi bàn giao lại xe là: 200000.0 VND

Ngày trả thực tế: 13/05/2023 04:33 This field is required.

[In ra hóa đơn - Trả xe](#)

[Trang chủ](#)

Hình 4.60 Dừng tài khoản admin để trả xe về kho

Hóa đơn

Tên xe	WINNER (W1-72K1-12347)
Tên người thuê	admin@gmail.com
Ngày thuê	2023-05-11
Ngày Trả	2023-05-12
Ngày trả thực tế	2023-05-13
Số tiền trễ hạn	220000.0

Tổng tiền trả trễ hạn: 220000.0

Cảm ơn bạn đã sử dụng dịch vụ!

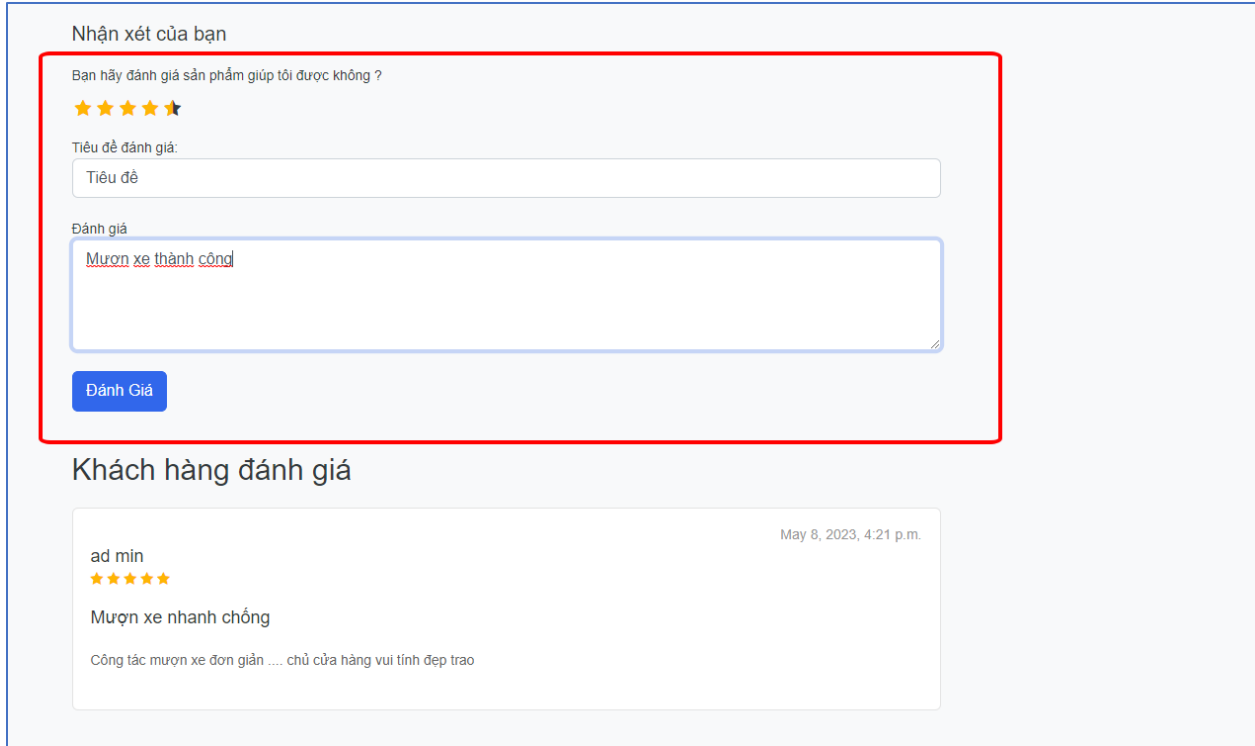
Trang chủ

Đây là hóa đơn được tạo tự động. Xin đừng trả lời thư này.

Hình 4.61 Hóa đơn khi trả xe

Chủ cửa hàng cần vào trong dashboard quản lý bill và phiếu mượn để chọn vào chế độ đã thanh toán nếu đã nhận được tiền để tính doanh thu.

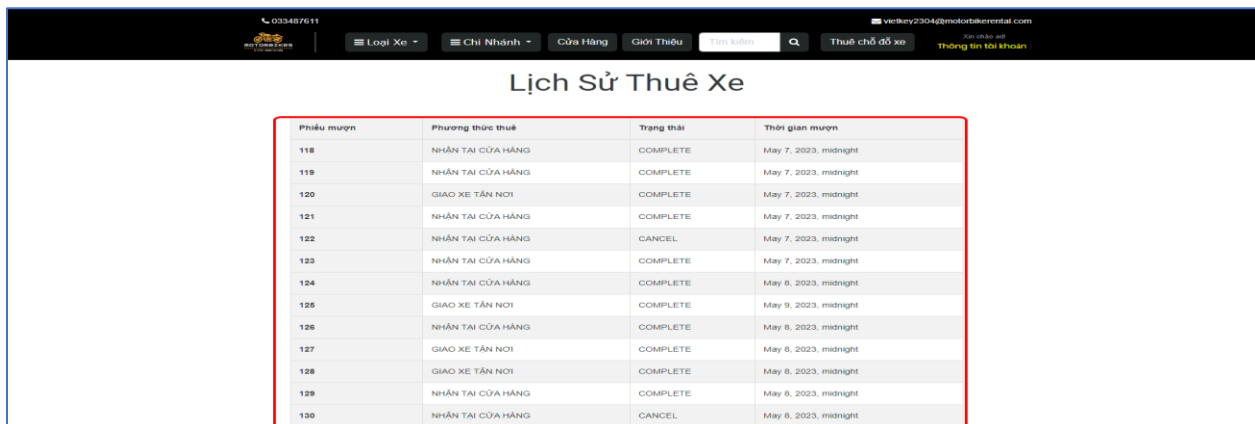
Sau khi thuê xe thì khách hàng được phép đánh giá xe ở bên dưới form thuê xe như sau: nhập tiêu đề và chọn sao đánh giá và cuối cùng là đánh giá.



Hình 4.62 Đánh giá xe

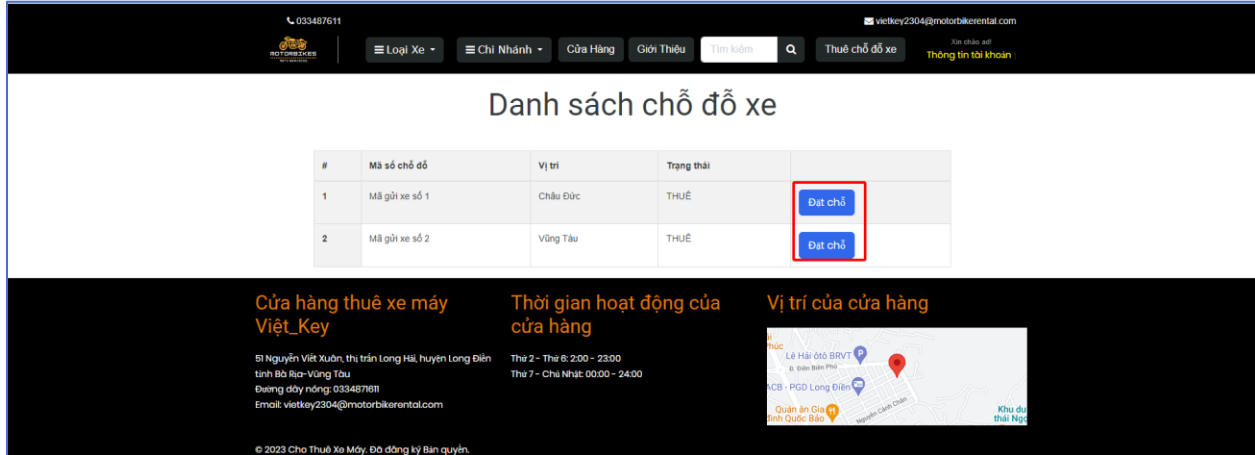
Ngoài có thể đánh giá theo sao và viết đánh giá thì khách hàng còn có thể thấy đánh giá trước đó của khách hàng khác để tham khảo về chất lượng dịch vụ và chất lượng của xe.

Khách hàng còn có thể vào thông tin tài khoản để xem lịch sử dùng dịch vụ thuê xe như sau: vào thông tin tài khoản chọn lịch sử thuê xe.

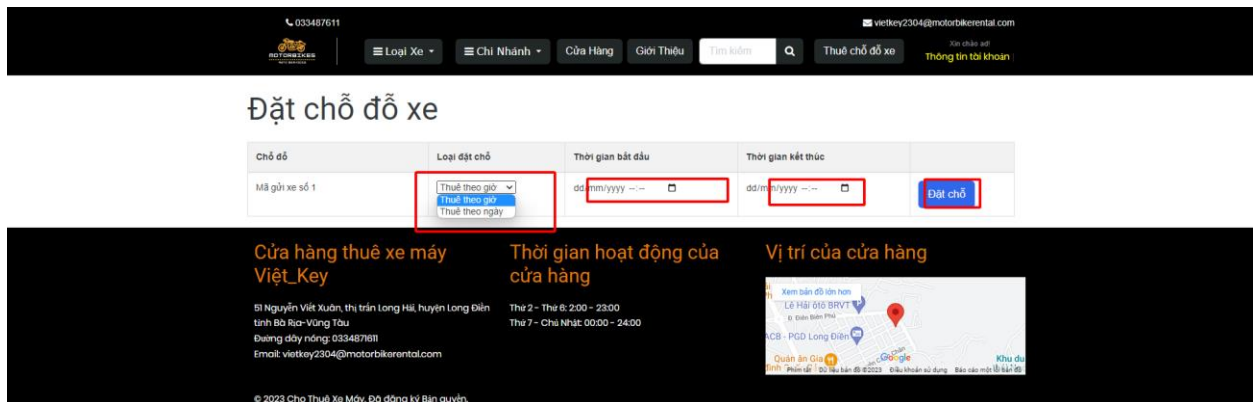


Hình 4.63 Lịch sử thuê xe

Chức năng thứ hai mà khách hàng đăng nhập được sử dụng đó chính là thuê chỗ đỗ chọn vào thuê chỗ đỗ trên thanh navbar sau đó xem danh sách chỗ đỗ và chọn đặt chỗ muốn đỗ xe.

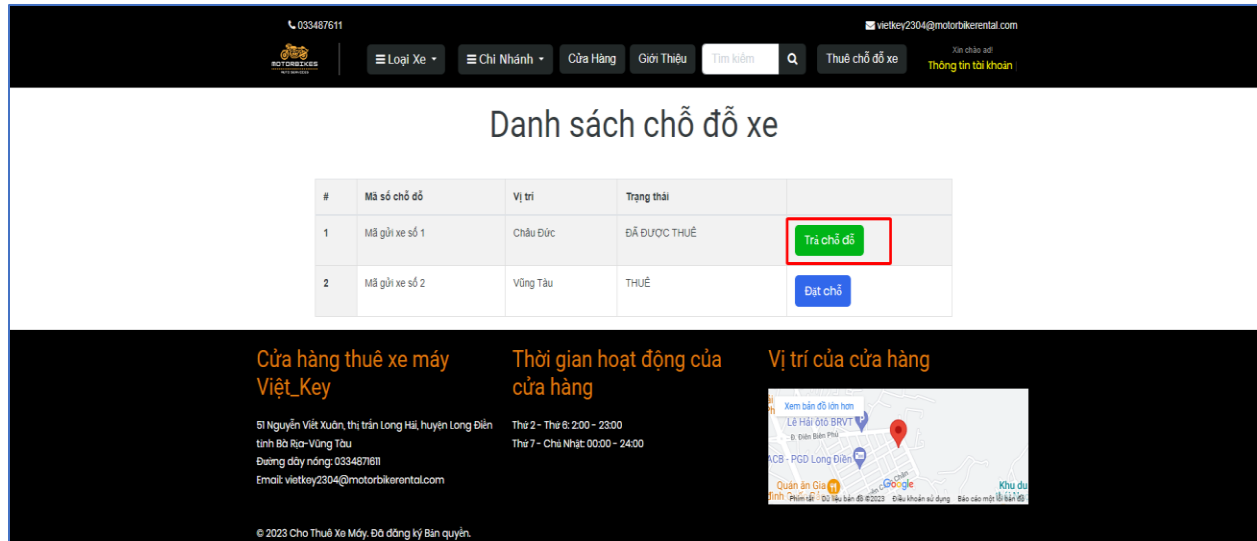


Hình 4.64 Danh sách đặt chỗ đỗ xe

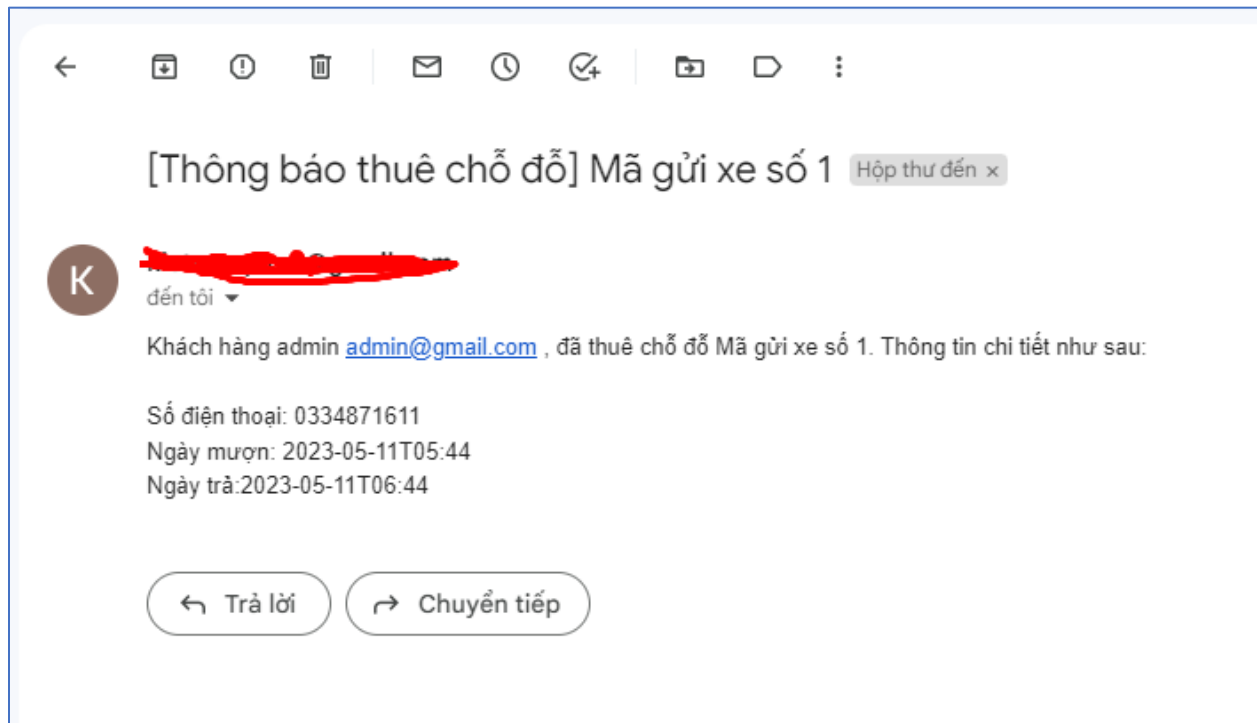


Hình 4.65 Form đặt chỗ đỗ xe

Chọn đủ các trường yêu cầu của form sau đó nhấn nút đặt chỗ mail sẽ được gửi về cho chủ cửa hàng và đổi trạng thái cho chỗ đỗ xe đó và đặt thành công chỗ.

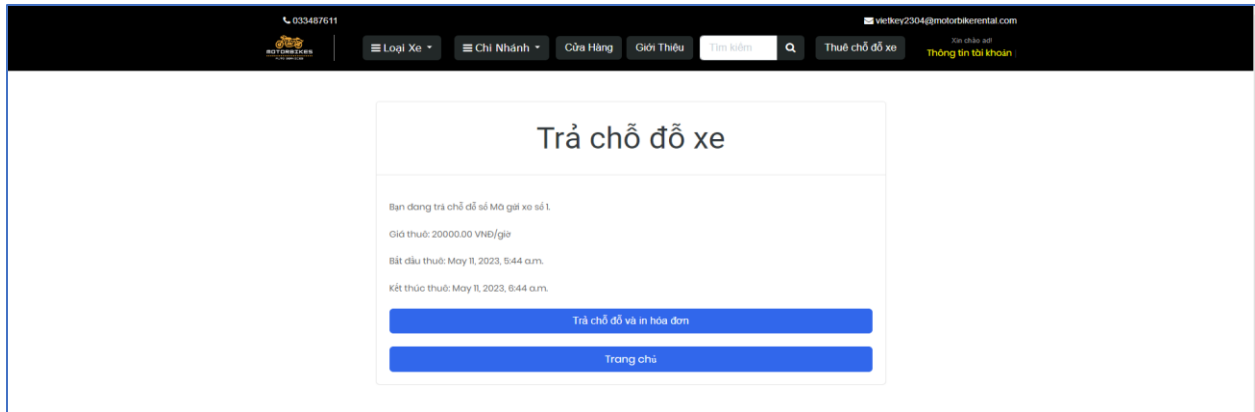


Hình 4.66 Trạng thái đổi khi thuê chỗ đỗ thành công



Hình 4.67 Mail chủ cửa hàng nhận được khi có người thuê chỗ đỗ xe

Khi lấy xe đi cần phải thanh toán tiền thuê chỗ đỗ và xuất hóa đơn, tương tự như thuê xe máy việc trả xe và xuất hóa đơn sẽ do admin, chủ cửa hàng xuất.



Hình 4.68 Trả chỗ cho thuê chỗ đỗ xe

Hóa đơn đỗ xe	
Mã số đặt chỗ	22
Mã số chỗ đỗ	Mã gửi xe số 1
Giá thuê	20000.00 VND/giờ
Bắt đầu thuê	May 11, 2023, 5:44 a.m.
Kết thúc thuê	May 11, 2023, 6:44 a.m.
Ngày thành toán	May 10, 2023, 9:48 p.m.
Tổng số tiền phải trả	20000.00 VND

[Trang chủ](#)

Hình 4.69 Hóa đơn chỗ đỗ xe

CHƯƠNG 5 KẾT LUẬN

5.1 Kết quả đạt được

- ✓ Vận dụng kiến thức về ngôn ngữ Python vào dự án thực tế
- ✓ Thực hành và nghiên cứu sâu vào framework Django
- ✓ Kỹ năng làm việc trao đổi, thu thập thông tin
- ✓ Nắm được quy trình quản lý thuê xe trong cửa hàng thuê xe Việt Key
- ✓ Học tập được trao đổi kiến thức thiết kế website và chuyển đổi số từ công ty TNHH Niềm Tin
- ✓ Phân tích, thiết kế hệ thống hoàn thiện chi tiết áp dụng công nghệ trong quá trình quản lý thuê sản phẩm
- ✓ Xây dựng website giới thiệu tiện lợi, đầy đủ thông tin, dễ thao tác truy cập
- ✓ Xây dựng và chuẩn hóa dữ liệu đảm bảo tính đúng đắn
- ✓ Giao diện quản lý dễ nhìn, rõ ràng dễ sử dụng

5.2 Hướng phát triển

- Xây dựng bảo dưỡng và đăng kiểm cho xe
- Nâng cấp giao diện thân thiện, dễ sử dụng cho người dùng
- Mở rộng cơ sở dữ liệu, quản lý chi tiết hơn nhằm đáp ứng nhu cầu của doanh nghiệp
- Quản lý xe để bảo vệ quy định số km phạm vi xe có thể hoạt động,
- Hỗ trợ xuất file PDF
- Nhân viên có thể tham gia vào hệ thống
- Tối ưu hóa quy trình thuê xe và thuê chỗ đỗ có thể đặt lịch thời gian dài
- Người dùng cũng sẽ tự tạo bài viết tự trao đổi real time để có trải nghiệm tốt khi cùng thời điểm thuê xe tại cửa hàng

Do thời gian thực hiện còn hạn hẹp, kiến thức và kinh nghiệm còn nhiều hạn chế nên bài báo cáo này không tránh khỏi những sai sót. Rất mong nhận được những đóng góp ý kiến quý báu của thầy cô trường Đại học Bà Rịa – Vũng Tàu.

Một lần nữa tôi xin chân thành cảm ơn quý thầy cô trong khoa Công nghệ kỹ thuật-công nghệ và đặc biệt tôi xin gửi đến thầy Phan Ngọc Hoàng, người đã tận tình hướng dẫn, giúp đỡ tôi hoàn thành đề tài đồ án tốt nghiệp lời cảm ơn sâu sắc nhất.

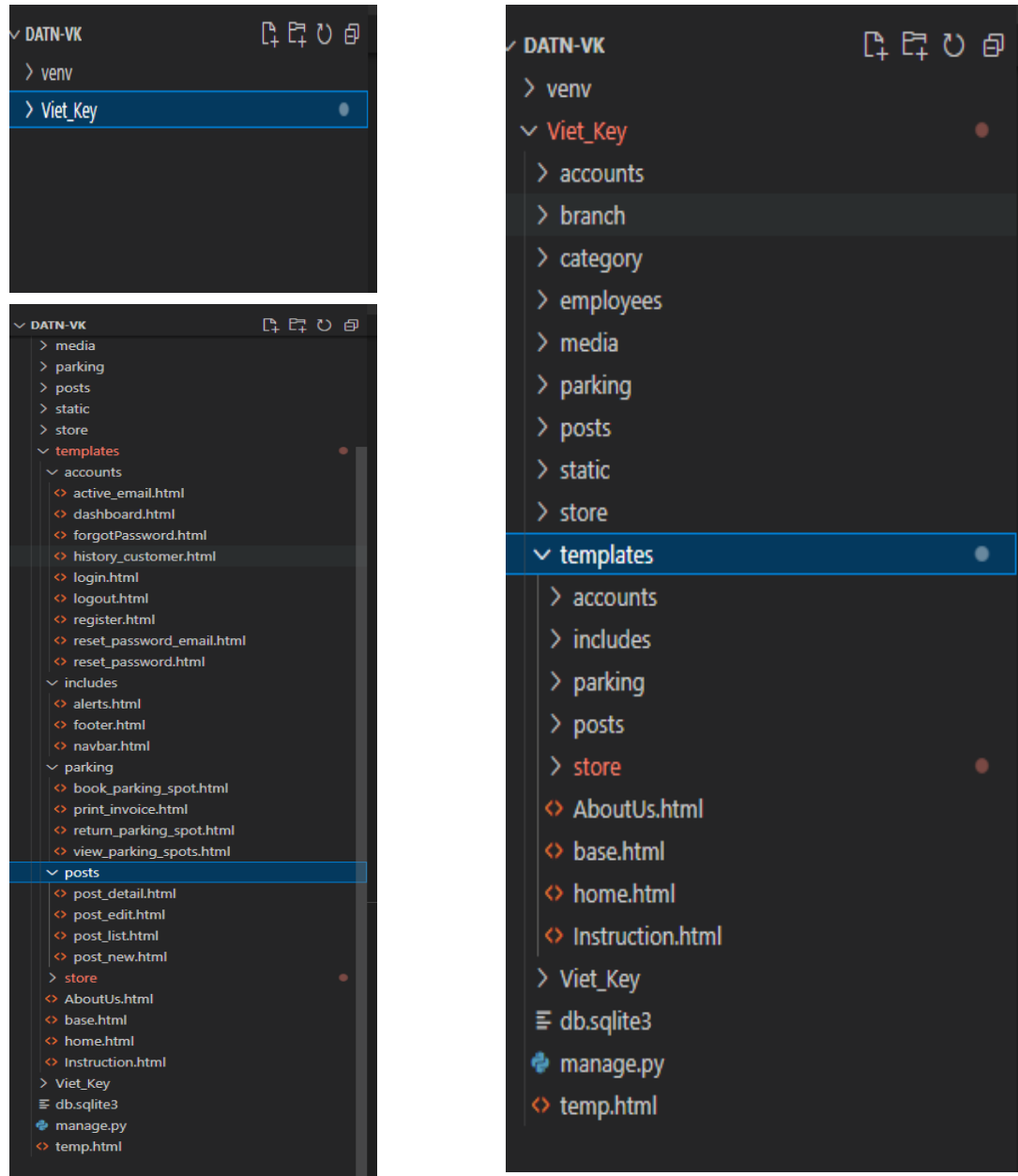
Tôi xin chân thành cảm ơn Ban Lãnh Đạo, các phòng ban của công ty TNHH Niềm Tin, đã tạo điều kiện thuận lợi cho t được tìm hiểu thực tiễn trong suốt quá trình thực tập tại công ty.

TÀI LIỆU THAM KHẢO

- [1] Poman (2020), ‘Học Python Để Làm Gì Cho Đời?’, 15/12/2020 Lấy từ URL: <https://codelearn.io/sharing/hoc-python-de-lam-gi-cho-doi>
- [2] Joefrey (2020). Templates Engineers.[Internet]. 19/10/2020 Lấy từ: URL: <https://colorlib.com/wp/template/engineers/>
- [3] Denise Chen (2019). Dashboard Tutorial (I): Flask and Chart.js. 19/10/2020 Lấy từ: URL: <https://towardsdatascience.com/flask-and-chart-js-tutorial-i-d33e05fba845>
- [4] Dương Tuấn Anh – Nguyễn Trung Trực (2006), Hệ cơ sở dữ liệu, nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh, Thành phố Hồ Chí Minh.
- [5] Tìm hiểu về django lấy từ: <https://viblo.asia/p/tim-hieu-ve-django-framework-ho-tro-python-trong-lap-trinh-web-QpmlxbkZrd>

PHỤ LỤC

1. Sơ đồ cấu trúc



Hình phụ lục 1a,b,c,d,e,f Cấu trúc thư mục

Bảng phụ lục 1 Bảng mô tả cấu trúc thư mục

STT	Tên thư mục	Ý nghĩa
1	static	Thư mục chứa hình ảnh, file CSS, Bootstrap, Js
2	template	Thư mục chứa các thiết kế giao diện (*.html)
3	venv	Thư mục chứa môi trường, các thư viện
4	migration	Thư mục chứa các file chuyển đổi dữ liệu

2. Code xử lý

2.1.Account

2.1.1 Models.py

```

from django.core.checks.messages import Error
from django.db import models
from django.contrib.auth.models import AbstractBaseUser, BaseUserManager

class MyAccountManager(BaseUserManager):
    def create_user(self, first_name, last_name, username, email,
password=None):
        if not email:
            raise ValueError('Email address is required')

        if not username:
            raise ValueError('User name is required')

        # Tạo đối tượng user mới
        user = self.model(
            email=self.normalize_email(email=email), # Chuyển email về
dạng bình thường
            username=username,
            first_name=first_name,
            last_name=last_name,
            is_staff=True,
        )

        user.set_password(password)
        user.save(using=self._db)
        return user

```

```
def create_superuser(self, first_name, last_name, email, username,
password):
    user = self.create_user(
        email=self.normalize_email(email=email),
        username=username,
        password=password,
        first_name=first_name,
        last_name=last_name,
    )
    user.is_admin = True
    user.is_active = True
    user.is_staff = True
    user.is_superuser = True
    user.save(using=self._db)
    return user

class Account(AbstractBaseUser):
    first_name = models.CharField(max_length=50)
    last_name = models.CharField(max_length=50)
    username = models.CharField(max_length=50, unique=True)
    email = models.EmailField(max_length=100, unique=True)
    phone_number = models.CharField(max_length=50)

    # required
    date_joined = models.DateTimeField(auto_now_add=True)
    last_login = models.DateTimeField(auto_now_add=True)
    is_admin = models.BooleanField(default=False)
    is_staff = models.BooleanField(default=False)
    is_active = models.BooleanField(default=False)
    is_superuser = models.BooleanField(default=False)

    USERNAME_FIELD = 'email' # Trường quyết định khi login
    REQUIRED_FIELDS = ['username', 'first_name', 'last_name'] # Các
trường yêu cầu khi đk tài khoản (mặc định đã có email), mặc định có
password

    objects = MyAccountManager()

    def __str__(self):
        return self.email
```



```
def has_perm(self, perm, obj=None):
    return self.is_admin # Admin có tất cả quyền trong hệ thống

def has_module_perms(self, add_label):
    return True

def full_name(self):
    return self.first_name + " " + self.last_name
```

2.1.2 Admin.py

```
from django.contrib import admin
from django.contrib.auth.admin import UserAdmin
from .models import Account

class AccountAdmin(UserAdmin):
    list_display = ('email', 'username', 'first_name', 'last_name',
'last_login', 'date_joined', 'is_active', 'phone_number')
    list_display_links = ('email', 'username', 'first_name',
'last_name') # Các trường có gắn link dẫn đến trang detail
    readonly_fields = ('last_login', 'date_joined') # Chỉ cho phép
đọc
    ordering = ('-date_joined',) # Sắp xếp theo chiều ngược

    # Bắt buộc phải khai báo
    filter_horizontal = ()
    list_filter = ()
    fieldsets = ()

admin.site.register(Account, AccountAdmin)
```

2.1.3 Forms.py

```
from django import forms
from django.forms import CharField
from django.forms import widgets
from .models import Account

class RegistrationForm(forms.ModelForm):
```

```
first_name = forms.CharField(max_length=100)

last_name = forms.CharField(max_length=100)

phone_number = forms.CharField(max_length=50)

email = forms.EmailField(max_length=50)

password = forms.CharField(widget=forms.PasswordInput(attrs={
    'placeholder': 'Enter password'
}))

confirm_password = forms.CharField(widget=forms.PasswordInput(attrs={
    'placeholder': 'Confirm password'
}))

class Meta:
    model = Account
    fields = ['first_name', 'last_name', 'phone_number', 'email',
'password']

def __init__(self, *args, **kwargs):
    super(RegistrationForm, self).__init__(*args, **kwargs)
    self.fields['first_name'].widget.attrs['placeholder'] = 'Enter
first name'
    self.fields['last_name'].widget.attrs['placeholder'] = 'Enter
last name'
    self.fields['phone_number'].widget.attrs['placeholder'] = 'Enter
phone number'
    self.fields['email'].widget.attrs['placeholder'] = 'Enter email'
    for field in self.fields:
        self.fields[field].widget.attrs['class'] = 'form-control'

def clean(self):
    cleaned_data = super(RegistrationForm, self).clean()
    password = cleaned_data.get('password')
    confirm_password = cleaned_data.get('confirm_password')

    if password != confirm_password:
        raise forms.ValidationError(
            'Password does not match!'
        )
```

2.1.4 Views.py

```
from re import split
from django.shortcuts import redirect, render
from django.contrib import messages, auth
from django.contrib.sites.shortcuts import get_current_site
from django.template.loader import render_to_string
from django.utils.http import urlsafe_base64_encode,
urlsafe_base64_decode
from django.utils.encoding import force_bytes
from django.core.mail import EmailMessage
from django.contrib.auth.decorators import login_required
from django.contrib.auth.tokens import default_token_generator
from store.models import Borrower

from .forms import RegistrationForm
from accounts.models import Account

import requests

def register(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            first_name = form.cleaned_data['first_name']
            last_name = form.cleaned_data['last_name']
            email = form.cleaned_data['email']
            phone_number = form.cleaned_data['phone_number']
            password = form.cleaned_data['password']
            username = email.split('@')[0]

            user = Account.objects.create_user(
                first_name=first_name, last_name=last_name, email=email,
                username=username, password=password)
            user.phone_number = phone_number
            user.save()

            current_site = get_current_site(request)
            mail_subject = 'Activate your blog account.'
            message = render_to_string('accounts/active_email.html', {
                'user': user,
                'domain': current_site.domain,
                'uid': urlsafe_base64_encode(force_bytes(user.pk)),
                'token': default_token_generator.make_token(user)
            })
```

```
        send_email = EmailMessage(mail_subject, message, to=[email])
        send_email.send()
        messages.success(
            request=request,
            message="Please confirm your email address to complete
the registration"
        )
        return redirect('register')
    else:
        messages.error(request=request, message="Register failed!")
    else:
        form = RegistrationForm()
    context = {
        'form': form,
    }
    return render(request, 'accounts/register.html', context)

def login(request):
    if request.method == "POST":
        email = request.POST.get('email')
        password = request.POST.get('password')
        user = auth.authenticate(email=email, password=password)
        if user is not None:
            auth.login(request=request, user=user)
            messages.success(request=request, message="Login
successful!")

            url = request.META.get('HTTP_REFERER')
            try:
                query = requests.utils.urlparse(url).query
                params = dict(x.split("=") for x in query.split("&"))
                if "next" in params:
                    next_page = params["next"]
                    return redirect(next_page)
            except Exception:
                return redirect('dashboard')
        else:
            messages.error(request=request, message="Login failed!")
    context = {
        'email': email if 'email' in locals() else '',
        'password': password if 'password' in locals() else '',
    }

    return render(request, 'accounts/login.html', context=context)
```

```
@login_required(login_url="login")
def logout(request):
    auth.logout(request)
    messages.success(request=request, message="You are logged out!")
    return redirect('login')

def activate(request, uidb64, token):
    try:
        uid = urlsafe_base64_decode(uidb64).decode()
        user = Account.objects.get(pk=uid)
    except Exception:
        user = None

    if user is not None and default_token_generator.check_token(user,
token):
        user.is_active = True
        user.save()
        messages.success(
            request=request, message="Your account is activated, please
login!")
        return render(request, 'accounts/login.html')
    else:
        messages.error(request=request, message="Activation link is
invalid!")
        return redirect('home')

@login_required(login_url="login")
def dashboard(request):
    return render(request, "accounts/dashboard.html")

def forgotPassword(request):
    try:
        if request.method == 'POST':
            email = request.POST.get('email')
            user = Account.objects.get(email__exact=email)

            current_site = get_current_site(request=request)
            mail_subject = 'Reset your password'
            message =
render_to_string('accounts/reset_password_email.html', {
```

```
        'user': user,
        'domain': current_site.domain,
        'uid': urlsafe_base64_encode(force_bytes(user.pk)),
        'token': default_token_generator.make_token(user)
    })
    send_email = EmailMessage(mail_subject, message, to=[email])
    send_email.send()

    messages.success(
        request=request, message="Password reset email has been
sent to your email address")
    except Exception:
        messages.error(request=request, message="Account does not
exist!")
    finally:
        context = {
            'email': email if 'email' in locals() else '',
        }
        return render(request, "accounts/forgotPassword.html",
context=context)

def reset_password_validate(request, uidb64, token):
    try:
        uid = urlsafe_base64_decode(uidb64).decode()
        user = Account.objects.get(pk=uid)
    except Exception:
        user = None

    if user is not None and default_token_generator.check_token(user,
token):
        request.session['uid'] = uid
        messages.info(request=request, message='Please reset your
password')
        return redirect('reset_password')
    else:
        messages.error(request=request, message="This link has been
expired!")
        return redirect('home')

def reset_password(request):
    if request.method == 'POST':
        password = request.POST.get('password')
        confirm_password = request.POST.get('confirm_password')
```

```
        if password == confirm_password:
            uid = request.session.get('uid')
            user = Account.objects.get(pk=uid)
            user.set_password(password)
            user.save()
            messages.success(request, message="Password reset
successful!")
            return redirect('login')
        else:
            messages.error(request, message="Password do not match!")
            return render(request, 'accounts/reset_password.html')

@login_required
def history_customer(request):
    user= request.user
    borrower = list(Borrower.objects.filter(borrower=user))

    context ={
        'borrower':borrower,
    }
    return render(request, 'accounts/history_customer.html', context)
```

2.2.Branch

2.2.1 Models.py

```
from django.db import models
from django.urls import reverse

class Branch(models.Model):
    branch_name = models.CharField(max_length=50, unique=True)
    slug = models.SlugField(max_length=100, unique=True)
    description = models.TextField(max_length=255, blank=True)

    class Meta:
        verbose_name = 'branch'
        verbose_name_plural = 'branches'

    def get_branch_url(self):
        return reverse('products_by_branch', args=[self.slug])
```

```
def __str__(self):  
    return self.branch_name
```

2.2.2 Admin.py

```
from django.contrib import admin  
from .models import Branch  
  
class BranchAdmin(admin.ModelAdmin):  
    prepopulated_fields = {'slug': ('branch_name',)} # Gợi ý  
    trường slug theo branch_name  
    list_display = ('branch_name', 'slug')  
    search_fields = ['branch_name']  
  
admin.site.register(Branch, BranchAdmin)
```

2.2.3 Views.py

```
from .models import Branch  
  
def menu_links(request):  
    linkbranch = Branch.objects.all()  
    return dict(linkbranch=linkbranch)
```

2.3.Category

2.3.1 Models.py

```
from django.db import models  
from django.urls import reverse  
  
class Category(models.Model):  
    category_name = models.CharField(max_length=50, unique=True)  
    slug = models.SlugField(max_length=100, unique=True)  
    description = models.TextField(max_length=255, blank=True)  
    category_image =  
    models.ImageField(upload_to='photos/categories/', blank=True)  
  
    class Meta:
```



```
verbose_name = 'category'
verbose_name_plural = 'categories'

def get_url(self):
    return reverse('products_by_category', args=[self.slug])

def __str__(self):
    return self.category_name
```

2.3.2 Admin.py

```
from django.contrib import admin
from .models import Category

class CategoryAdmin(admin.ModelAdmin):
    prepopulated_fields = {'slug': ('category_name',)} # Gợi ý
    trường slug theo category_name
    list_display = ('category_name', 'slug')
    search_fields = ['category_name']

admin.site.register(Category, CategoryAdmin)
```

2.3.3 Context_processors.py

```
from .models import Category

def menu_links(request):
    links = Category.objects.all()
    return dict(links=links)
```

2.4. Employees

2.4.1 Models.py

```
from django.db import models
from django.utils import timezone
from branch.models import Branch

class Employee(models.Model):
    name = models.CharField(max_length=255)
```

```
address = models.CharField(max_length=255)
phone_number = models.CharField(max_length=20, null=
True, blank=True)
branch = models.ForeignKey(Branch, on_delete=models.CASCADE, null=
True, blank=True)
email = models.EmailField(max_length=255)

def __str__(self):
    return self.name

class Attendance(models.Model):
    employee = models.ForeignKey(Employee, on_delete=models.CASCADE,
related_name='attendance')
    date = models.DateField(default=timezone.now)
    is_present = models.BooleanField(default=False)

def __str__(self):
    return f"{self.employee} - {self.date}"

def save(self, *args, **kwargs):
    super(Attendance, self).save(*args, **kwargs)
    if self.is_present:
        self.employee.salary.update_working_days()

class Salary(models.Model):
    employee = models.OneToOneField(Employee,
on_delete=models.CASCADE, related_name='salary')
    base_salary = models.DecimalField(max_digits=10,
decimal_places=2, default=0)
    working_days = models.IntegerField(default=0)

def __str__(self):
    return f"{self.employee} - {self.base_salary} -
{self.working_days}"

def update_working_days(self):
    self.working_days += 1
    self.save()

def calculate_salary(self):
    return self.base_salary * self.working_days
```

2.4.2 Admin.py

```
import datetime
from django.contrib import admin
from django.http import HttpResponse
from openpyxl import Workbook
from .models import Employee, Attendance, Salary

class AttendanceInline(admin.TabularInline):
    model = Attendance
    extra = 1

class SalaryInline(admin.StackedInline):
    model = Salary
    can_delete = False

@admin.register(Employee)
class EmployeeAdmin(admin.ModelAdmin):
    inlines = [AttendanceInline, SalaryInline]
    list_display = ['name', 'address', 'phone_number', 'email',
'branch']
    list_filter = ['name', 'branch']
    search_fields = ['name']

    def branch(self, obj):
        return obj.branch.branch_name
    branch.short_description = 'Branch'

@admin.register(Attendance)
class AttendanceAdmin(admin.ModelAdmin):
    list_display = ['employee', 'date', 'is_present']
    list_filter = ['date']
    search_fields = ['employee__name']

@admin.register(Salary)
class SalaryAdmin(admin.ModelAdmin):
    list_display = ['employee', 'base_salary', 'working_days',
'calculate_salary']
    search_fields = ['employee__name']
    readonly_fields = ['working_days', 'calculate_salary']
```

```
def export_salary_to_excel(self, request, queryset):
    # Define the response object with a specific header.
    response =
HttpResponse(content_type='application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet')
        response['Content-Disposition'] = f'attachment;
filename=salary_report_{datetime.datetime.now().strftime("%Y-%m-
%d_%H-%M-%S")}.xlsx'

    # Create the workbook and worksheet objects.
    wb = Workbook()
    ws = wb.active

    # Define the header row.
    header = ['Employee', 'Base Salary', 'Working Days', 'Total
Salary']

    # Write the header row to the worksheet.
    ws.append(header)

    # Write each Salary object to the worksheet.
    for salary in queryset:
        row = [salary.employee.name, salary.base_salary,
salary.working_days, salary.calculate_salary()]
        ws.append(row)

    # Save the workbook to the response.
    wb.save(response)

    return response

    # Define the custom action for the export_salary_to_excel
function.
    export_salary_to_excel.short_description = 'Export Selected
Salaries to Excel'

    # Add the custom action to the admin.
    actions = [export_salary_to_excel]
```

2.5.Parking

2.5.1 Models.py

```
from django.db import models
from accounts.models import Account
from branch.models import Branch

class ParkingSpot(models.Model):
    spot_number = models.CharField(max_length=200)
    STATUS_CHOICES = (
        ('THUÊ', 'thuê'),
        ('ĐÃ ĐƯỢC THUÊ', 'đã được thuê'),
    )
    status = models.CharField(max_length=250, choices=STATUS_CHOICES,
default='THUÊ')
    hourly_price = models.DecimalField(max_digits=10,
decimal_places=2)
    daily_price = models.DecimalField(max_digits=10,
decimal_places=2)
    location =
models.ForeignKey(Branch,on_delete=models.CASCADE,null=True,blank=Tru
e)

    def __str__(self):
        return self.spot_number

class ParkingReservation(models.Model):
    reservation_types = (
        ('HOURLY', 'Hourly'),
        ('DAILY', 'Daily'),
    )

    user = models.ForeignKey(Account, on_delete=models.CASCADE)
    parking_spot = models.ForeignKey(ParkingSpot,
on_delete=models.CASCADE)
    reservation_type = models.CharField(max_length=200,
choices=reservation_types)
    reservation_start_time = models.DateTimeField()
    reservation_end_time = models.DateTimeField()
    paid = models.BooleanField(default=False)
    paid_date = models.DateTimeField(null=True, blank=True)

    def __str__(self):
        return f"{self.user.username} -
{self.parking_spot.spot_number} -
{self.reservation_start_time.date()}"
```

```
def amount_due(self):
    if self.reservation_type == 'HOUR':
        hours = (self.reservation_end_time -
self.reservation_start_time).seconds // 3600
        return self.parking_spot.hourly_price * hours
    elif self.reservation_type == 'DAY':
        days = (self.reservation_end_time -
self.reservation_start_time).days + 1
        return self.parking_spot.daily_price * days
    else:
        return 0.0
```

2.5.2 Admin.py

```
from django.contrib import admin
from .models import ParkingSpot, ParkingReservation

class ParkingReservationAdmin(admin.ModelAdmin):
    list_display = ['id', 'user',
'parking_spot', 'reservation_type', 'reservation_start_time', 'reservati
on_end_time', 'paid', 'paid_date']
    search_fields = ['user']

admin.site.register(ParkingSpot)
admin.site.register(ParkingReservation, ParkingReservationAdmin)
```

2.5.3 Views.py

```
import io
from reportlab.pdfgen import canvas
from django.shortcuts import get_object_or_404, render, redirect
from django.http import Http404, HttpResponse
from .models import ParkingSpot, ParkingReservation
from django.contrib import messages
from django.utils import timezone
from django.contrib.auth.decorators import login_required
from django.contrib.admin.views.decorators import import
staff_member_required
from io import BytesIO
```

```
from django.template.loader import render_to_string
from xhtml2pdf import pisa
from django.template.loader import get_template
import pprint
from .models import Branch
from email.mime.text import MIMEText
from django.conf import settings
import smtplib

def parking(request):
    parking_spots = ParkingSpot.objects.all()
    context = {
        'parking_spots': parking_spots,
        'parking_spots1': parking_spots
    }
    return render(request, 'parking/view_parking_spots.html', context)

@login_required
def book_parking_spot(request, spot_number):
    parking_spot =
list(ParkingSpot.objects.filter(spot_number=spot_number))[0]
    pprint.pprint(parking_spot)
    if request.method == 'POST':

        user = request.user
        reservation_type = request.POST["reservation_type"]
        reservation_start_time =
request.POST["reservation_start_time"]
        reservation_end_time = request.POST["reservation_end_time"]

        parkingReservation=ParkingReservation.objects.create(
            user = user,
            parking_spot = parking_spot,
            reservation_type = reservation_type,
            reservation_start_time = reservation_start_time,
            reservation_end_time = reservation_end_time,
            paid = False,
            paid_date = None
        )
        parkingReservation.save()
        parking_spot.status="ĐÃ ĐƯỢC THUÊ"
```

```
        parking_spot.save()
        email_body = f"Khách hàng {user.username} {user.email} , đã
thuê chỗ đỗ {parking_spot}. Thông tin chi tiết như sau:\n\nSố điện
thoại: {user.phone_number}\nNgày mượn: {reservation_start_time}\nNgày
trả:{reservation_end_time}"

        # Tạo message object
        msg = MIMEText(email_body)
        msg['Subject'] = f"[Thông báo thuê chỗ đỗ] {parking_spot}"
        msg['From'] = settings.EMAIL_HOST_USER
        msg['To'] = settings.NOTIFICATION_EMAIL

        # Gửi email thông báo
        with smtplib.SMTP(settings.EMAIL_HOST, settings.EMAIL_PORT) as
server:
            server.starttls()
            server.login(settings.EMAIL_HOST_USER,
settings.EMAIL_HOST_PASSWORD)
            server.sendmail(settings.EMAIL_HOST_USER,
[settings.NOTIFICATION_EMAIL], msg.as_string())
            return redirect('parking')

    context = {

        'parking_spot': parking_spot
    }

    return render(request, 'parking/book_parking_spot.html', context)

@staff_member_required
def return_parking_spot(request, spot_number):
    try:
        reservations =
ParkingReservation.objects.filter(parking_spot__spot_number=spot_numbe
r, paid=False)
        reservation = reservations.first()
    except ParkingReservation.DoesNotExist:
        raise Http404("Reservation does not exist")

    if reservation.parking_spot.status == 'THUÊ':
        messages.warning(request, 'Chỗ đỗ xe hiện không có người
thuê.')
    return redirect('parking')
```



```
if request.method == 'POST':
    reservation.paid = True
    reservation.paid_date = timezone.now()
    reservation.save()

    spot = reservation.parking_spot
    spot.status = 'THUÊ'
    spot.save()

    amount_due = reservation.amount_due()
    context = {'reservation': reservation, 'amount_due':
amount_due}

    # Create invoice PDF
    # template_path = 'parking/print_invoice.html'
    # template = get_template(template_path)
    # html = template.render(context)
    # file = BytesIO()
    # pisa.CreatePDF(html, dest=file)
    # file.seek(0)
    # pdf = file.read()
    # file.close()

    # # Return PDF as response
    # response = HttpResponse(pdf, content_type='application/pdf')
    # response['Content-Disposition'] = f'attachment;
filename=invoice_{reservation.id}.pdf'
    # return response
    return render(request, 'parking/print_invoice.html', context)

context = {'reservation': reservation}
return render(request, 'parking/return_parking_spot.html',
context)

def parking_location(request, location):
    id_location = Branch.objects.filter(branch_name=location)[0]

    parking_spots = ParkingSpot.objects.filter(location=id_location)
    parking_spots1 = ParkingSpot.objects.all()
    context = {
        'parking_spots': parking_spots,
        'parking_spots1': parking_spots1
    }
```

```
return render(request, 'parking/view_parking_spots.html', context)
```

2.6.Posts

2.6.1 Models.py

```
from django.db import models
from django.utils.text import slugify
from django.urls import reverse
from accounts.models import Account

class Post(models.Model):
    id_name=
    models.ForeignKey(Account,on_delete=models.CASCADE,blank=True,null
    =True)
    title = models.CharField(max_length=200)
    title1 = models.CharField(max_length=200, blank=True,
    null=True)
    title2 = models.CharField(max_length=200, blank=True,
    null=True)
    title3 = models.CharField(max_length=200, blank=True,
    null=True)
    title4 = models.CharField(max_length=200, blank=True,
    null=True)
    title5 = models.CharField(max_length=200, blank=True,
    null=True)
    title6 = models.CharField(max_length=200, blank=True,
    null=True)

    content = models.TextField()
    content1 = models.TextField(blank=True, null=True)
    content2 = models.TextField(blank=True, null=True)
    content3 = models.TextField(blank=True, null=True)
    content4 = models.TextField(blank=True, null=True)
    content5 = models.TextField(blank=True, null=True)
    content6 = models.TextField(blank=True, null=True)

    image1= models.ImageField(upload_to='photos', blank=True,
    null=True)
    image2 = models.ImageField(upload_to='photos', blank=True,
    null=True)
```

```
        image = models.ImageField(upload_to='photos')
        image3 = models.ImageField(upload_to='photos', blank=True,
null=True)
        image4 = models.ImageField(upload_to='photos', blank=True,
null=True)
        image5 = models.ImageField(upload_to='photos', blank=True,
null=True)
        image6 = models.ImageField(upload_to='photos', blank=True,
null=True)

        slug = models.SlugField(max_length=200, unique=True)
        created_date = models.DateTimeField(auto_now_add=True)
        modified_date = models.DateTimeField(auto_now=True)

    def save(self, *args, **kwargs):
        if not self.id:
            self.slug = slugify(self.title)
            super().save(*args, **kwargs)

    def get_absolute_url(self):
        return reverse('post_detail', args=[str(self.id)])
```

2.6.2 Admin.py

```
from django.contrib import admin
from .models import Post

class PostAdmin(admin.ModelAdmin):
    list_display = ('title', 'slug', 'created_date', 'modified_date')
    prepopulated_fields = {'slug': ('title',)}

admin.site.register(Post, PostAdmin)
```

2.6.3 Forms.py

```
from django import forms
from .models import Post

class PostForm(forms.ModelForm):
    class Meta:
        model = Post
```

```
fields = ['title', 'content', 'image']
```

2.6.4 Views.py

```
from django.shortcuts import render, get_object_or_404, redirect
from .models import Post
from .forms import PostForm

def post_list(request):
    posts = Post.objects.all()
    return render(request, 'posts/post_list.html', {'posts': posts})

def post_detail(request, slug):
    post = get_object_or_404(Post, slug=slug)
    return render(request, 'posts/post_detail.html', {'post': post})

def post_new(request):
    if request.method == "POST":
        form = PostForm(request.POST)
        if form.is_valid():
            post = form.save(commit=False)
            post.author = request.user
            post.save()
            return redirect('posts/post_detail', slug=post.slug)
    else:
        form = PostForm()
    return render(request, 'posts/post_edit.html', {'form': form})

def post_edit(request, slug):
    post = get_object_or_404(Post, slug=slug)
    if request.method == "POST":
        form = PostForm(request.POST, instance=post)
        if form.is_valid():
            post = form.save(commit=False)
            post.author = request.user
            post.save()
            return redirect('posts/post_detail', slug=post.slug)
    else:
        form = PostForm(instance=post)
    return render(request, 'posts/post_edit.html', {'form': form})
```

2.7.Store

2.7.1 Models.py

```
from django.urls import reverse
from category.models import Category
from branch.models import Branch
from employees.models import Employee
from accounts.models import Account
from django.db import models
from django.core.exceptions import ValidationError

class Product(models.Model):
    product_name = models.CharField(max_length=200, unique=True)
    slug = models.SlugField(max_length=200, unique=True)
    description = models.TextField(max_length=4000, blank=True)
    price = models.IntegerField()
    images = models.ImageField(upload_to='photos/products')
    is_available = models.BooleanField(default=True)
    category = models.ForeignKey(Category,
on_delete=models.CASCADE) # Khi xóa category thì Product bị xóa
    branch = models.ForeignKey(Branch,
on_delete=models.CASCADE,null=True,blank=True) # Khi xóa branch
thì Product bị xóa
    created_date = models.DateTimeField(auto_now_add=True)
    modified_date = models.DateTimeField(auto_now=True)
    STATUS_CHOICES = (
        ('THUÊ', 'thuê'),
        ('CHỜ XÁC NHẬN', 'chờ xác nhận'),
        ('CHỜ HỦY', 'chờ hủy'),
        ('ĐANG GIAO XE', 'đang giao xe'),
        ('ĐÃ ĐƯỢC THUÊ', 'đã được thuê'),
    )
    status = models.CharField(max_length=250, choices=STATUS_CHOICES,
default='THUÊ')

    def get_url(self):
        if self.branch:
```

```
        return reverse('product_detail',
args=[self.category.slug, self.branch.slug, self.slug])
    else:
        return reverse('product_detail',
args=[self.category.slug, self.slug])

    def __str__(self):
        return self.product_name

class ReviewRating(models.Model):
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    user = models.ForeignKey(Account, on_delete=models.CASCADE)
    subject = models.CharField(max_length=100, blank=True)
    review = models.TextField(max_length=500, blank=True)
    rating = models.FloatField()
    ip = models.CharField(max_length=20, blank=True)
    status = models.BooleanField(default=True)
    created_at = models.DateTimeField(auto_now_add=True)
    updated_at = models.DateTimeField(auto_now_add=True)

    def __str__(self):
        return self.subject

class Bill(models.Model):
    created_date = models.DateTimeField(auto_now_add=True)
    payment_date = models.DateTimeField(null=True, blank=True)
    total_price = models.IntegerField()
    is_paid = models.BooleanField(default=False)

    def __str__(self):
        return f"Bill #{self.pk}"

class BillItem(models.Model):
    bill = models.ForeignKey(Bill, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    rental_price = models.IntegerField()
    total_price = models.IntegerField()

    def __str__(self):
        return f"{self.product.product_name} ({self.total_price})"

class Borrower(models.Model):
    BORROW_STATUS_CHOICES = (
        ('COMPLETE', 'hoàn thành'),
```

```

        ('CONFIRM', 'đang xác nhận'),
        ('CANCEL', 'hủy'),
    )
    delivery_method_CHOICES = (
        ('NHẬN TẠI CỬA HÀNG', 'nhận tại cửa hàng'),
        ('GIAO XE TẬN NƠI', 'giao xe tận nơi'),
    )
    delivery_method = models.CharField(max_length=50,
choices=delivery_method_CHOICES, default='NHẬN TẠI CỬA HÀNG')
    delivery_address =
models.CharField(max_length=4000, null=True, blank=True)
    delivery_time = models.DateTimeField(null=True, blank=True)
    employee = models.ForeignKey(Employee,
on_delete=models.CASCADE, null=True, blank=True)
    borrower = models.ForeignKey(Account, on_delete=models.CASCADE)
    product = models.ForeignKey(Product, on_delete=models.CASCADE)
    borrow_date = models.DateTimeField(auto_now_add=True)
    return_date = models.DateTimeField(null=True, blank=True)
    real_return_date = models.DateTimeField(null=True, blank=True)
    status = models.CharField(max_length=20,
choices=BORROW_STATUS_CHOICES, default='COMPLETE')
    total_price = models.IntegerField(null=True, blank=True)
    bill = models.ForeignKey(Bill, on_delete=models.CASCADE,
null=True, blank=True)

    def __str__(self):
        return f"ID: { str(self.id)} || USER:
{self.borrower.username} || Borrowed: {self.product.product_name} ||
Trạng thái hiện tại của xe: {self.product.status} "

    def get_rental_price(self):
        if self.return_date and self.real_return_date:
            if self.return_date < self.borrow_date or
self.real_return_date < self.borrow_date:
                raise ValidationError('Ngày không được trước ngày
mượn')
            else:
                days_late = (self.real_return_date.date() -
self.return_date.date()).days
            else:
                days_late = 0

        total_days = (self.return_date.date() -
self.borrow_date.date()).days if self.return_date else 0

```

```
        total_price = (self.product.price * total_days) +
        (self.product.price * days_late)+(self.product.price * days_late*0.1)
        return total_price

    def save(self, *args, **kwargs):
        self.total_price = self.get_rental_price()
        super(Borrower, self).save(*args, **kwargs)

    if self.status == 'COMPLETE' and not self.bill:
        bill = Bill.objects.create(total_price=self.total_price)
        BillItem.objects.create(
            bill=bill,
            product=self.product,
            rental_price=self.product.price,
            total_price=self.total_price
        )
        self.bill = bill
        self.save()
    elif self.status != 'COMPLETE' and self.bill:
        self.bill.delete()
        self.bill = None
        self.save()
```

2.7.2 Admin.py

```
from django.contrib import admin
from django.http import HttpResponseRedirect
from django.urls import reverse
from .models import Product, Borrower, ReviewRating, Bill, BillItem
from .views import revenue_report, review_borrowr
class BillItemInline(admin.TabularInline):
    model = BillItem
    extra = 1

class BorrowerInline(admin.TabularInline):
    model = Borrower
    extra = 0

class BillAdmin(admin.ModelAdmin):
    list_display = ('created_date', 'payment_date', 'is_paid')
    list_filter = ('payment_date', 'is_paid')
```



```
class BillItemAdmin(admin.ModelAdmin):
    list_display=('id','bill','product','total_price','rental_price')

class ProductAdmin(admin.ModelAdmin):
    list_display = ('product_name', 'price','category',
'created_date', 'modified_date', 'is_available','status')
    prepopulated_fields = {'slug': ('product_name',)}
    list_filter = ('category', 'is_available')
    search_fields = ('product_name', 'description')
    inlines = [BorrowerInline]

class BorrowerAdmin(admin.ModelAdmin):

    list_display = ('id','borrower', 'product', 'borrow_date',
'return_date', 'real_return_date', 'status', 'total_price')
    list_filter = ('status', 'borrow_date', 'return_date',
'real_return_date')
    search_fields = ('borrower__username', 'product__product_name')
    actions = ['review_borrowr']

    def review_borrowr(self, request, queryset):
        # Chuyển hướng tới trang báo cáo
        for obj in queryset:
            borrower_id = obj.id
            return HttpResponseRedirect(reverse('review_borrowr',
args=[borrower_id]))
        review_borrowr.short_description = "View borrower invoice"

admin.site.register(Product, ProductAdmin)
admin.site.register(ReviewRating)
admin.site.register(Borrower,BorrowerAdmin)
admin.site.register(Bill, BillAdmin)
admin.site.register(BillItem,BillItemAdmin)
```

2.7.3 Forms.py

```
from django import forms
import requests
import re
from .models import ReviewRating,Borrower
import datetime
```

```
from django.utils import timezone
from .models import Borrower, Product
from django.core.exceptions import ValidationError

class ReviewForm(forms.ModelForm):
    class Meta:
        model = ReviewRating
        fields = ['subject', 'review', 'rating']

class ReturnProductForm(forms.Form):
    real_return_date = forms.DateTimeField(label="Ngày trả thực tế",
required=True, widget=forms.DateTimeInput(attrs={'type': 'datetime-
local'}))

    def __init__(self, *args, **kwargs):
        self.borrower = kwargs.pop('borrower')
        super().__init__(*args, **kwargs)

    def clean_real_return_date(self):
        real_return_date = self.cleaned_data['real_return_date']

        if real_return_date < self.borrower.borrow_date:
            raise ValidationError('Ngày trả thực tế không được trước
ngày mượn.')
        return real_return_date

    def save(self):
        borrower = self.borrower
        borrower.real_return_date =
self.cleaned_data['real_return_date']
        borrower.status = 'COMPLETE'
        borrower.save()
        product = borrower.product
        product.status = 'THUÊ'
        product.save()

class BorrowProductForm(forms.Form):
    delivery_method_CHOICES = (
        ('NHẬN TẠI CỬA HÀNG', 'nhận tại cửa hàng'),
        ('GIAO XE TẬN NƠI', 'giao xe tận nơi'),
    )
```

```
borrow_date = forms.DateTimeField(
    label='Ngày mượn',
    widget=forms.DateTimeInput(attrs={'type': 'datetime-local'})
)
return_date = forms.DateTimeField(
    label='Ngày trả',
    widget=forms.DateTimeInput(attrs={'type': 'datetime-local'})
)
delivery_method = forms.ChoiceField(
    choices=delivery_method_CHOICES,
    widget=forms.RadioSelect(),
    required=True
)
delivery_address = forms.CharField(
    label='Địa chỉ giao hàng',
    widget=forms.Textarea(attrs={'rows': 3, 'cols': 30}),
    required=False
)

# Set minimum borrow date to current date and time
def __init__(self, *args, **kwargs):
    user = kwargs.pop('user', None)
    super().__init__(*args, **kwargs)
    if user is not None:
        self.fields['borrow_date'].widget.attrs['min'] =
datetime.datetime.now().strftime('%Y-%m-%dT%H:%M')

# Validate borrow and return dates
def clean(self):
    cleaned_data = super().clean()
    borrow_date = cleaned_data.get('borrow_date')
    return_date = cleaned_data.get('return_date')
    if borrow_date and return_date:
        if return_date <= borrow_date:
            raise ValidationError('Ngày trả phải lớn hơn ngày
mượn.')
    return cleaned_data

# Validate delivery address if borrow method is delivery
def clean_delivery_address(self):
    delivery_method = self.cleaned_data.get('delivery_method')
    delivery_address = self.cleaned_data.get('delivery_address')
```

```
        if delivery_method == 'GIAO XE TẬN NƠI' and not
delivery_address:
            raise forms.ValidationError('Bạn phải nhập địa chỉ giao
hàng.')

        return delivery_address
```

2.7.4 Views.py

```
from audioop import reverse
import datetime
import io
from django.shortcuts import render
from django.urls import reverse
from django.http import HttpResponse
from django.utils.html import format_html
from .models import Product, Bill
import random
from django.shortcuts import render, redirect
from django.utils import timezone
from django.contrib import messages
from django.http import Http404, HttpResponse
from category.models import Category
from branch.models import Branch
from django.shortcuts import get_object_or_404, redirect, render
from store.forms import ReviewForm, ReturnProductForm
from .models import Employee
from employees.models import Attendance
from .models import Borrower
from django.core.paginator import Paginator
from django.db.models import Q
from store.models import Product, ReviewRating
from django.contrib.auth.decorators import login_required
from category.models import Category
from django.contrib.admin.views.decorators import login_required
from django.contrib.admin.views.decorators import staff_member_required
from django.template.loader import get_template
from xhtml2pdf import pisa
import smtplib
from email.mime.text import MIMEText
from django.conf import settings
from io import BytesIO, StringIO
```

```
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import letter
from reportlab.pdfbase import pdfmetrics
from reportlab.pdfbase.ttfonts import TTFont

def store(request, category_slug=None, branch_slug=None):
    if category_slug is not None:
        categories = get_object_or_404(Category, slug=category_slug)
        products = Product.objects.all().filter(category=categories,
is_available=True)
    elif branch_slug is not None:
        branches = get_object_or_404(Branch, slug=branch_slug)
        products = Product.objects.all().filter(branch=branches,
is_available=True)
    else:
        products =
Product.objects.all().filter(is_available=True).order_by('id')

    page = request.GET.get('page')
    page = page or 1
    paginator = Paginator(products, 3)
    paged_products = paginator.get_page(page)
    product_count = products.count()

    context = {
        'products': paged_products,
        'product_count': product_count,
    }
    return render(request, 'store/store.html', context=context)

def product_detail(request, category_slug, product_slug,
branch_slug=None):
    single_product =
Product.objects.get(category__slug=category_slug, slug=product_slug,
branch__slug=branch_slug)
    user=request.user
    reviews =
ReviewRating.objects.filter(product_id=single_product.id,
status=True)

    cancel = False

    if single_product.status == "CHỜ XÁC NHẬN" :
```

```
        borrower_obj =
list(Borrower.objects.filter(product=single_product,status =
"CONFIRM"))
        print(borrower_obj)
        print(user)
        print(borrower_obj[0].borrower)
        if borrower_obj[0].borrower==user:
            print(borrower_obj[0])
            print(user)
            cancel =True
        print(cancel)
    context = {
        'single_product': single_product,
        'reviews': reviews,
        'trangthai':cancel,
    }
    return render(request, 'store/product_detail.html',
context=context)

def search(request):
    if 'q' in request.GET:
        q = request.GET.get('q')
        products = Product.objects.order_by('-
created_date').filter(Q(product_name__icontains=q) |
Q(description__icontains=q))
        product_count = products.count()
        context = {
            'products': products,
            'q': q,
            'product_count': product_count
        }
        return render(request, 'store/store.html', context=context)

def submit_review(request, product_id):
    url = request.META.get('HTTP_REFERER')
    if request.method == "POST":
        try:
            review =
ReviewRating.objects.get(user__id=request.user.id,
product__id=product_id)
            form = ReviewForm(request.POST, instance=review)
            form.save()
            messages.success(request, "Cảm ơn! Đánh giá của bạn đã
được cập nhật.")
```

```
        return redirect(url)
    except Exception:
        form = ReviewForm(request.POST)
        if form.is_valid():
            data = ReviewRating()
            data.subject = form.cleaned_data['subject']
            data.rating = form.cleaned_data['rating']
            data.review = form.cleaned_data['review']
            data.ip = request.META.get('REMOTE_ADDR')
            data.product_id = product_id
            data.user_id = request.user.id
            data.save()
            messages.success(request, "Cảm ơn! Đánh giá của bạn
đã được gửi.")
            return redirect(url)

@login_required
def borrow_product(request, product_slug):
    product = get_object_or_404(Product, slug=product_slug)

    todays_attendance =
Attendance.objects.filter(date=datetime.date.today(),
employee__branch__id=product.branch_id)
    present = Attendance.objects.filter(date=datetime.date.today(),
is_present=True, employee__branch__id=product.branch_id)
    if todays_attendance and present:
        attended_employee_ids = [attendance.employee_id for
attendance in present]
        random_employee_id = random.choice(attended_employee_ids)
        employee = get_object_or_404(Employee, id=random_employee_id)

    else:
        employee = None
        print(todays_attendance)
        print(present)

    borrower = request.user
    phone_number = borrower.phone_number
    Error1 = None
    Error2 = None
    if request.method == 'POST':
        print("post")
        user=request.user
```

```
borrower_obj, created = Borrower.objects.get_or_create(
    borrower=borrower,
    product=product,
    status='CONFIRM',
    employee=employee,
)

borrower_obj.borrow_date =
datetime.datetime.strptime(request.POST['borrow_date'], '%Y-%m-%d')
borrower_obj.return_date =
datetime.datetime.strptime(request.POST['return_date'], '%Y-%m-%d')

borrower_obj.delivery_method=request.POST['delivery_method']
delivery_address_nb_st
=request.POST['delivery_address_nb_st']
delivery_address_n_st =request.POST['delivery_address_n_st']
delivery_address_n_lc =request.POST['delivery_address_n_lc']
delivery_address_n_bran
=request.POST['delivery_address_n_bran']

if borrower_obj.borrow_date >
datetime.datetime.now()+datetime.timedelta(days=1):
    Error1 = "Không thể mượn xe trước hai ngày"

if Error1 :
    reviews =
ReviewRating.objects.filter(product_id=product.id, status=True)
cancel = False
if product.status == "CHỜ XÁC NHẬN" :
    borrower_obj1 =
list(Borrower.objects.filter(product=product,status = "CONFIRM"))
    if borrower_obj1[0].borrower==user:
        cancel =True
    context = {
        'single_product': product,
        'reviews': reviews,
        'trangthai':cancel,
        'error1': Error1,
    }
    return render(request, 'store/product_detail.html',
context=context)

if borrower_obj.return_date <= borrower_obj.borrow_date:
```



```
        Error2= "Ngày trả phải lớn hơn ngày mượn."  
    if Error2 :  
        reviews =  
ReviewRating.objects.filter(product_id=product.id, status=True)  
        cancel = False  
        if product.status == "CHỜ XÁC NHẬN" :  
            borrower_obj1 =  
list(Borrower.objects.filter(product=product,status = "CONFIRM"))  
            if borrower_obj1[0].borrower==user:  
                cancel =True  
            context = {  
                'single_product': product,  
                'reviews': reviews,  
                'trangthai':cancel,  
                'error2': Error2,  
            }  
            return render(request, 'store/product_detail.html',  
context=context)  
        else:  
            borrower_obj.delivery_address=delivery_address_nb_st +  
"+ delivery_address_n_st + " "+ delivery_address_n_lc + " "+  
delivery_address_n_bran  
            borrower_obj.save()  
  
            product.status = 'CHỜ XÁC NHẬN'  
            product.save()  
  
            context = {  
                'product': product,  
                'borrower': borrower,  
                'phone_number': phone_number,  
                'borrow_date': borrower_obj.borrow_date,  
                'return_date': borrower_obj.return_date,  
                'rental_price': borrower_obj.get_rental_price(),  
                'total_price': borrower_obj.get_rental_price(),  
                'delivery_method': borrower_obj.delivery_method,  
                'delivery_address':delivery_address_nb_st + " "+  
delivery_address_n_st + " "+ delivery_address_n_lc + " "+  
delivery_address_n_bran,  
                'employee': employee,  
            }  
  
            if borrower_obj.delivery_method == 'NHẬN TẠI CỬA HÀNG':  
                # Tạo nội dung email
```

```

        email_body = f"Khách hàng {borrower.username}
{borrower.email} , đã thuê sản phẩm {product.product_name}. Thông tin
chi tiết như sau:\n\nSố điện thoại: {phone_number}\nNgày mượn:
{borrower_obj.borrow_date}\nNgày trả:
{borrower_obj.return_date}\nPhương thức nhận hàng:
{borrower_obj.delivery_method}\nGiá thuê:
{borrower_obj.get_rental_price()}"

        # Tạo message object
        msg = MIMEText(email_body)
        msg['Subject'] = f"[Thông báo thuê sản phẩm]
{product.product_name}"
        msg['From'] = settings.EMAIL_HOST_USER
        msg['To'] = settings.NOTIFICATION_EMAIL

        # Gửi email thông báo
        with smtplib.SMTP(settings.EMAIL_HOST,
settings.EMAIL_PORT) as server:
            server.starttls()
            server.login(settings.EMAIL_HOST_USER,
settings.EMAIL_HOST_PASSWORD)
            server.sendmail(settings.EMAIL_HOST_USER,
[settings.NOTIFICATION_EMAIL], msg.as_string())

        return render(request, 'store/borrow_success.html',
context)

    elif borrower_obj.delivery_method == 'GIAO XE TẬN NƠI':
        if employee:

            employee_name = employee.name
            employee_phone= employee.phone_number
        else:
            messages.error(request, 'Không có nhân viên nào
đi làm hôm nay.')
            employee_name = "Admin"
            employee_phone="033487611"
            email_body = f"Khách hàng {borrower.username}
{borrower.email} , đã thuê sản phẩm {product.product_name}. Thông tin
chi tiết như sau:\n\nSố điện thoại: {phone_number}\nNgày mượn:
{borrower_obj.borrow_date}\nNgày trả:
{borrower_obj.return_date}\nPhương thức nhận hàng:
{borrower_obj.delivery_method}\nĐịa chỉ giao xe:
{borrower_obj.delivery_address}\nNhân viên giao xe:

```

```
{employee_name}\Số điện thoại Nhân viên giao xe:
{employee_phone}\nGiá thuê: {borrower_obj.get_rental_price()}"

    # Tạo message object
    msg = MIMEText(email_body)
    msg['Subject'] = f"[Thông báo thuê sản phẩm]
{product.product_name}"
    msg['From'] = settings.EMAIL_HOST_USER
    msg['To'] = settings.NOTIFICATION_EMAIL

    # Gửi email thông báo
    with smtplib.SMTP(settings.EMAIL_HOST,
settings.EMAIL_PORT) as server:
        server.starttls()
        server.login(settings.EMAIL_HOST_USER,
settings.EMAIL_HOST_PASSWORD)
        server.sendmail(settings.EMAIL_HOST_USER,
[settings.NOTIFICATION_EMAIL], msg.as_string())

    return render(request, 'store/borrow_delivery.html',
context)

    return render(request, 'store/borrow_product.html', {'product':
product, 'phone_number': phone_number})

@login_required
def borrow_success(request):
    return render(request, 'store/borrow_success.html')

@login_required
def cancel_product(request, product_slug):
    if request.method == 'POST':
        product = get_object_or_404(Product, slug=product_slug)

        borrower =
list(Borrower.objects.filter(product=product, status="CONFIRM"))
        product.status="CHỜ HỦY"
        # Tạo nội dung email
        email_body = f"Khách hàng {borrower[0].borrower.username}
{borrower[0].borrower.email} , đã thuê sản phẩm
{product.product_name}. đang hủy"
```

```
# Tạo message object
msg = MIMEText(email_body)
msg['Subject'] = f"[Thông báo hủy thuê]
{product.product_name}"
msg['From'] = settings.EMAIL_HOST_USER
msg['To'] = settings.NOTIFICATION_EMAIL

# Gửi email thông báo
with smtplib.SMTP(settings.EMAIL_HOST, settings.EMAIL_PORT)
as server:
    server.starttls()
    server.login(settings.EMAIL_HOST_USER,
settings.EMAIL_HOST_PASSWORD)
    server.sendmail(settings.EMAIL_HOST_USER,
[settings.NOTIFICATION_EMAIL], msg.as_string())

    product.save()
    borrower[0].status="CANCEL"
    borrower[0].save()

    return redirect("/")

@staff_member_required
def return_product(request, product_slug):
    product = get_object_or_404(Product, slug=product_slug)
    borrower = Borrower.objects.filter(product=product,
status='CONFIRM').first()
    if borrower is not None:
        if request.method == 'POST':
            form = ReturnProductForm(request.POST, borrower=borrower)
            if form.is_valid():
                form.save()
                messages.success(request, 'Sản phẩm đã được trả lại
thành công.')
                return redirect('export_invoice',
borrower_id=borrower.id)

            else:
                form = ReturnProductForm(initial={'real_return_date':
timezone.now()}, borrower=borrower)
        else:
            raise Http404("Người thuê không tồn tại.")
```

```
context = {
    'product': product,
    'form': form,
    'rental_price': borrower.get_rental_price(),
    'total_price': borrower.get_rental_price(),
}
return render(request, 'store/return_product.html', context)

@staff_member_required
def return_success(request):
    return render(request, 'store/return_success.html',
{'invoice_link': reverse('export_invoice', kwargs={'borrower_id':
request.GET.get('borrower_id')}})})

@staff_member_required
def export_invoice(request, borrower_id):
    borrower_obj = get_object_or_404(Borrower, id=borrower_id,
status='COMPLETE')
    borrow_date = borrower_obj.borrow_date
    return_date = borrower_obj.return_date
    real_return_date = borrower_obj.real_return_date
    product_name = borrower_obj.product.product_name
    borrower_name = borrower_obj.borrower

    invoice_data = {
        'product_name': product_name,
        'borrower_name': borrower_name,
        'borrow_date': borrow_date.strftime('%Y-%m-%d'),
        'return_date': return_date.strftime('%Y-%m-%d'),
        'real_return_date': real_return_date.strftime('%Y-%m-%d'),
    }

    late_fee = 0
    if real_return_date > return_date:
        delta = real_return_date - return_date
        days_late = delta.days
        late_fee = days_late *
borrower_obj.product.price+(days_late*borrower_obj.product.price*0.1)
    invoice_data['late_fee'] = late_fee

    # response = HttpResponse(content_type='application/pdf')
    # response['Content-Disposition'] = f'attachment;
filename="invoice_{borrower_obj.id}.pdf"
```

```
# template = get_template('store/invoice.html')
# html = template.render(invoice_data)
# pisa_status = pisa.CreatePDF(html, dest=response)
# if pisa_status.err:
#     return HttpResponse('There was an error generating the PDF
file.')
# return response
return render(request, 'store/invoice.html',invoice_data)

@staff_member_required
def revenue_report(request):

    category=None
    year=None
    yearinput=None
    yearstart = 2000
    yearnow = int(datetime.datetime.now().year)
    listyear=[]

    Allproduct = Product.objects.all()
    listbor=[]
    for all in Allproduct:
        if all.category not in listbor:

            listbor.append(all.category)

    if request.method == 'POST':
        category =request.POST['category']

        yearinput =request.POST['yearinput']

    if yearinput :
        try:
            year=int(yearinput)
        except:
            year = yearnow
    else:
        year = yearnow

    for l in range(yearstart,yearnow+1):
        listyear.append(l)

    month = int(datetime.datetime.now().month)
```

```
bill_completed = Bill.objects.filter(is_paid=True)
data_key = []
data_content = []
for k in range(1, month + 1):
    total_rice = 0

    if category and category != "all":
        for i in bill_completed:
            if int(i.payment_date.strftime("%m")) == k and
int(i.payment_date.strftime("%Y")) == year:
                p = Borrower.objects.filter(bill=i)[0]
                if str(p.product.category).strip() ==
str(category).strip():
                    total_rice += i.total_price
            else:
                for i in bill_completed:
                    if int(i.payment_date.strftime("%m")) == k and
int(i.payment_date.strftime("%Y")) == year:
                        total_rice += i.total_price
                data_key.append(k)
                data_content.append(total_rice)
    context = {
        'data_key': data_key,
        'data_value': data_content,
        'listyear': listyear,
        'categorys': listbor,
    }

    return render(request, 'store/revenue_report.html', context)

@staff_member_required
def review_borrowr(request, borrower_id):
    borrower_obj = get_object_or_404(Borrower, id=borrower_id,
status='CONFIRM')
    print(borrower_obj)
    borrow_date = borrower_obj.borrow_date
    return_date = borrower_obj.return_date
    delivery_method = borrower_obj.delivery_method
```

```
product_name = borrower_obj.product.product_name
borrower_name = borrower_obj.borrower
employee=borrower_obj.employee
if borrower_obj.employee == None:
    employee="Admin"
    phonemloyee="0334871611"
else:
    phonemloyee=borrower_obj.employee.phone_number
total_price=borrower_obj.total_price
borrower_product_category = borrower_obj.product.category
borrower_product_branch = borrower_obj.product.branch
borrower_product_price =borrower_obj.product.price
borrower_numberphone = borrower_obj.borrower.phone_number
time_delivery = borrower_obj.delivery_time
address_delivery = borrower_obj.delivery_address

invoice_data = {
    'product_name': product_name,
    'borrower_name': borrower_name,
    'borrow_date': borrow_date.strftime('%Y-%m-%d'),
    'return_date': return_date.strftime('%Y-%m-%d'),
    'delivery_method': delivery_method,
    'employee': employee,
    'phonemloyee': phonemloyee,
    'borrower_product_category': borrower_product_category,
    'borrower_product_branch': borrower_product_branch,
    'borrower_product_price': borrower_product_price,
    'borrower_numberphone': borrower_numberphone,
    'time_delivery': time_delivery,
    'address_delivery': address_delivery,

}

# response = HttpResponse(content_type='application/pdf')
# response['Content-Disposition'] = f'attachment;
filename="invoice_{borrower_obj.id}.pdf"

# template = get_template('store/review_borrowr.html')
# html = template.render(invoice_data)
# font_path = 'C:/Users/kietv/OneDrive/Desktop/DATN-
VK/Viet_Key/Viet_Key/static/fonts/arial-unicode-ms.ttf'
```



```
# pisa_status = pisa.CreatePDF(io.StringIO(html), dest=response,
encoding='utf-8', path=font_path)
# if pisa_status.err:
#     return HttpResponse('There was an error generating the PDF
file.')
# return response

return render(request, 'store/review_borrowr.html', invoice_data)
```