

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

KHOA ĐA PHƯƠNG TIỆN



**ĐỒ ÁN
TỐT NGHIỆP ĐẠI HỌC**

**ĐỀ TÀI: PHÁT TRIỂN WEBSITE ĐẶT PHÒNG VÀ QUẢN
LÝ KHÁCH SẠN SỬ DỤNG NODEJS VÀ REACTJS**

Giảng viên hướng dẫn: TS. Trần Quý Nam

Sinh viên thực hiện: Bùi Thị Mai

Mã sinh viên: B19DCPT154

Lớp : D19PTDPT

Hệ đào tạo: Đại học chính quy

Khoá: 2019-2024

Hà Nội - 2023

ĐỀ TÀI ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC

Họ và tên sinh viên: Bùi Thị Mai

MSV: B19DCPT154

Lớp: D19PTDPT

Khoá: 2019-2023

Ngành đào tạo: Công Nghệ Đa Phương Tiện

Hệ đào tạo: Đại học chính quy

1. Tên đề án/khoá luận tốt nghiệp: Xây dựng Website đặt phòng, quản lý khách sạn sử dụng ReactJS và NodeJs

2. Lý do chọn đề tài:

Sau thời kỳ khó khăn do ảnh hưởng của dịch bệnh, ngành du lịch và khách sạn cũng đã bắt đầu phục hồi. Sự hồi phục này tạo ra nhu cầu lớn cho việc đặt phòng khách sạn, và với sự phát triển của internet và công nghệ thông tin việc đặt phòng trực tuyến so với cách thủ công tạo nên sự tiện lợi đối với người dùng.

Nhắm vào các đối tượng là các bạn trẻ bận rộn với cuộc sống hàng ngày có nhu cầu chọn địa điểm khách sạn nghỉ ngơi thư giãn, trang web đặt phòng khách sạn phát triển nhằm giúp cho các bạn dễ dàng tìm kiếm các khách sạn tốt nhất, tiết kiệm thời gian và chi phí đặt phòng.

Đối với các khách sạn và nhà nghỉ, việc quản lý danh sách phòng, việc đặt phòng, thanh toán và thông tin liên quan đòi hỏi nhiều công sức. Một hệ thống quản lý khách sạn trực tuyến sẽ giúp họ dễ dàng quản lý tình trạng phòng, tối ưu hóa doanh nghiệp của họ và nâng cao hiệu quả làm việc.

Trang web có thể cung cấp nhiều thông tin hữu ích về các khách sạn, bao gồm hình ảnh, tiện nghi, vị trí, đánh giá của khách hàng, và các khuyến mãi đặc biệt. Điều này giúp người dùng lựa chọn phòng khách sạn phù hợp với nhu cầu của họ.

ReactJS là một framework phát triển front-end phổ biến, được thiết kế để xây dựng giao diện người dùng dễ sử dụng, tương tác và hiệu quả. NodeJS, là một môi trường chạy mã JavaScript phía máy chủ, được sử dụng phổ biến trong việc phát triển back-

end của các ứng dụng web. Chọn sử dụng chúng trong dự án này mang lại tính linh hoạt, hiệu suất cao, và có cơ hội để áp dụng kiến thức mới và sáng tạo, đồng thời giúp tạo ra một sản phẩm chất lượng.

Dự án này không chỉ đáp ứng nhu cầu của ngành du lịch và khách hàng mà còn mang lại lợi ích cho nhà cung cấp dịch vụ khách sạn, đóng góp vào sự phát triển của ngành du lịch và sử dụng công nghệ tiên tiến để xây dựng một sản phẩm chất lượng.

3. Nội dung chính của đồ án:

Đồ án được chia làm các chương như sau:

Chương 1: Tổng quan về đề tài.

Chương 2: Cơ sở lý thuyết.

Chương 3: Phân tích thiết kế hệ thống.

Chương 4: Xây dựng ứng dụng web.

Chương 5: Đưa ra kết quả và kết luận của việc xây dựng hệ thống.

4. Ngày giao đề tài://20...

5. Ngày nộp quyển:/...../20.....

GIẢNG VIÊN HƯỚNG DẪN

(Ký, ghi rõ họ tên)

SINH VIÊN THỰC HIỆN

(Ký, ghi rõ họ tên)

TRƯỞNG KHOA

(Ký, ghi rõ họ tên)

LỜI CẢM ƠN

Được học tập và nghiên cứu tại khoa Đa phương tiện – Học viện Công nghệ Bưu chính Viễn thông, em đã được trải nghiệm trong môi trường đào tạo tốt và nhận được sự chỉ dạy nhiệt tình của các thầy, các cô trong khoa.

Trước hết, em xin được bày tỏ lòng biết ơn và gửi lời cảm ơn chân thành đến **thầy Trần Quý Nam** đã dạy cho em có được những kiến thức vững chắc, tận tình chỉ bảo, nhắc nhở và hướng dẫn em trong suốt quá trình làm đồ án tốt nghiệp.

Em cũng xin chân thành cảm ơn các thầy cô giáo trong khoa Đa phương tiện nói riêng và Học viện Công nghệ Bưu chính Viễn thông nói chung đã trang bị cho em những kiến thức quý báu làm hành trang trong những năm học vừa qua.

Em cũng xin bày tỏ lòng biết ơn sâu sắc đến: Cha mẹ và những người thân trong gia đình đã chăm sóc, nuôi dạy, hỗ trợ, động viên và tạo mọi điều kiện thuận lợi nhất cho em trong suốt thời gian qua và đặc biệt trong thời gian em làm đồ án tốt nghiệp.

Trong quá trình nghiên cứu của mình, mặc dù được sự hướng dẫn rất nhiệt tình, tận tâm của **thầy Trần Quý Nam** cùng với sự nỗ lực của cá nhân nhưng cũng không thể tránh được những thiếu sót. Em rất mong nhận được sự cảm thông và những góp ý từ quý Thầy, Cô và các bạn để đề tài được hoàn thiện hơn.

Em xin chân thành cảm ơn.

Hà nội, ngày 15 tháng 12 năm 2023

Sinh viên thực hiện

Mai

Bùi Thị Mai

MỤC LỤC

LỜI CẢM ƠN.....	1
MỤC LỤC	2
DANH MỤC BẢNG	7
CHƯƠNG I: GIỚI THIỆU CHUNG	8
1.1 Bối cảnh và mục tiêu	8
1.1.1 Tên đề tài	8
1.1.2 Lý do chọn đề tài	8
1.2 Đối tượng và phạm vi nghiên cứu	9
1.2.1 Đối tượng nghiên cứu	9
1.2.2 Môi trường sử dụng	11
1.2.3 Phạm vi nghiên cứu	11
1.3 Cấu trúc đồ án	11
1.4 Kết luận chương	12
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	13
2.1 Tổng quan về một hệ thống website.....	13
2.1.1 Khái niệm	13
2.1.2 Cách thức hoạt động của 1 website	13
2.1.3 Các thành phần của 1 website.....	14
2.2 Tìm hiểu về HTML, CSS	14
2.2.1 Tìm hiểu về HTML.....	14
2.2.2 Tìm hiểu về CSS	15
2.2.3 Mối quan hệ giữa HTML và CSS	16
2.3 Tìm hiểu về Javascript.....	16
2.3.2 Vai trò.....	17
2.3.3 Ưu, nhược điểm của Javascript	17
2.4 Tìm hiểu về ReactJS trong thiết kế website	18
2.4.1 Giới thiệu về ReactJS	18
2.4.2 Cách ReactJS hoạt động	20
2.4.3 Khái niệm cơ bản trong ReactJS.....	21

2.4.4 Vòng đời trong ReactJS	22
2.4.5 React Hooks	24
2.4.6. Tìm hiểu về Redux	24
2.4.7 Tìm hiểu về Material UI.....	25
2.4.8 Tìm hiểu về axios	26
2.5 Tìm hiểu về NodeJS trong lập trình website	27
2.5.1 Khái niệm	27
2.5.2 Các đặc điểm của NodeJS.....	28
2.5.3 Ưu điểm	28
2.5.4 Tìm hiểu về npm	29
2.5.5 Tìm hiểu về JWT	29
2.6 Tìm hiểu về hệ quản trị CSDL MongoDB	29
2.6.1 Khái niệm	29
2.6.2 Ưu điểm	30
2.7 Tìm hiểu về Postman.....	31
2.7.1 Khái niệm	31
2.7.2 Đặc điểm của Postman	31
2.8 Kết luận chương.....	32
CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	33
3.1 Khảo sát và phân tích người dùng.....	33
3.1.1 Mục tiêu của website	33
3.1.2 Đối tượng chính của website đặt phòng khách sạn.....	33
3.1.3 Nghiên cứu đối thủ cạnh tranh.....	33
3.2 Kiến trúc hệ thống.....	34
3.1.2 Kiến trúc của hệ thống Client – Server.....	34
3.2.2 Phương thức giao tiếp giữa Client – Server.....	34
3.3 Công nghệ triển khai	34
3.3.1 Phía backend.....	34
3.3.2 Phía client	35
3.4 Phân tích thiết kế hệ thống	36
3.4.1 Mô tả các chức năng trong hệ thống.....	36

3.4.2 Sơ đồ use case hệ thống	36
3.4.3 Scenario	40
3.4.4 Biểu đồ tuần tự.....	50
3.5 Kết luận chương	53
CHƯƠNG IV: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG	55
4.1. Chuẩn bị môi trường	55
4.1.1 NodeJs.....	55
4.1.2 Express.....	56
4.1.3 Cài đặt Visual Studio Code.....	56
4.2 Xây dựng Frontend.....	57
4.2.1 Cấu trúc frontend	57
4.2.2 Cấu trúc backend	59
4.3 Giao diện website.....	60
4.3.1 Các thành phần của website	60
4.3.2 Các trang của website.....	63
CHƯƠNG V: TỔNG KẾT	75
5.2 Hạn chế.....	75
5.3 Định hướng phát triển	75

DANH MỤC HÌNH VẼ

Hình ảnh 2.1. 1: Cách thức hoạt động của 1 website	13
Hình ảnh 2.2.1 1: Tìm hiểu về HTML	15
Hình ảnh 2.2.2. 1: Tìm hiểu về CSS	15
Hình ảnh 2.2.3. 1: Mối quan hệ giữa HTML và CSS	16
Hình ảnh 2.4.1. 1: Tìm hiểu về Reactjs.....	18
Hình ảnh 2.4.2. 1: Virtual DOM và Brower DOM.....	20
Hình ảnh 2.4.3. 1: Components trong ReactJS	21
Hình ảnh 2.4.4. 1: Vòng đời của Components trong ReactJS	23
Hình ảnh 2.5.1. 1 NodeJS.....	27
Hình ảnh 2.6.1 1 MongoDB.....	30
Hình ảnh 2.7.1. 1 Postman	31
Hình ảnh 3.4.2. 1: Sơ đồ usecase hệ thống website đặt phòng khách sạn	37
Hình ảnh 3.4.2. 2: Usecase chức năng đặt phòng phía khách hàng	37
Hình ảnh 3.4.2. 3: Usecase chức năng quản lý các phòng đã đặt phía khách hàng	38
Hình ảnh 3.4.2. 4: Usecase quản lý tài khoản phía user	38
Hình ảnh 3.4.2. 5: Usecase quản lý tài khoản người dùng phía admin.....	39
Hình ảnh 3.4.2. 6: Usecase quản lý phòng phía admin	39
Hình ảnh 3.4.2. 7: Usecase quản lý đặt phòng phía admin	40
Hình ảnh 3.4.4. 1: Biểu đồ tuần tự chức năng đăng nhập	50
Hình ảnh 3.4.4. 2: Biểu đồ tuần tự chức năng đăng ký.....	51
Hình ảnh 3.4.4. 3: Biểu đồ tuần tự luồng tìm kiếm phòng.....	51
Hình ảnh 3.4.4. 4: Biểu đồ tuần tự luồng đặt phòng	51
Hình ảnh 3.4.4. 5: Biểu đồ tuần tự luồng xem chi tiết đặt phòng	52
Hình ảnh 3.4.4. 6: Biểu đồ tuần tự luồng huỷ đặt phòng phía người dùng.....	52
Hình ảnh 3.4.4. 7: Biểu đồ tuần tự luồng cập nhật thông tin người dùng.....	52
Hình ảnh 3.4.4. 8: Biểu đồ tuần tự luồng thêm tài khoản người dùng phía admin	53
Hình ảnh 3.4.4. 9: Biểu đồ tuần tự quản lý đặt phòng phía admin.....	53
Hình ảnh 4.1.1 1: Website cài đặt NodeJS	55
Hình ảnh 4.1.1 2: Hình ảnh kiểm tra version Node.....	56
Hình ảnh 4.1.3 1: Trang website cài đặt Visual Studio Code	57

Hình ảnh 4.2.1 1: Câu lệnh chạy dự án ReactJS	57
Hình ảnh 4.2.1 2: Cấu trúc mã nguồn dự án	58
Hình ảnh 4.2.2 1: Cấu trúc thư mục phía backend.....	59
Hình ảnh 4.2.2 2: Các API phía backend	60
Hình ảnh 4.3.1. 1: Thành phần header của website	60
Hình ảnh 4.3.1. 2: Mã code thành phần Top Header	61
Hình ảnh 4.3.1. 3: Mã code thành phần main header của website.....	61
Hình ảnh 4.3.1. 4: mã code thành phần list room trong website.....	62
Hình ảnh 4.3.1. 5: Mã code thành phần header của website.....	62
Hình ảnh 4.3.2. 1: Giao diện website trang đăng ký tài khoản	63
Hình ảnh 4.3.2. 2: Giao diện website trang đăng nhập tài khoản	64
Hình ảnh 4.3.2. 3: Giao diện trang chủ của website.....	65
Hình ảnh 4.3.2. 4: Giao diện trang giới thiệu khách sạn.....	67
Hình ảnh 4.3.2. 5: Giao diện trang liên hệ	67
Hình ảnh 4.3.2. 6: Giao diện trang danh sách phòng	68
Hình ảnh 4.3.2. 7:Giao diện trang danh sách phòng đơn của website	69
Hình ảnh 4.3.2. 8: Giao diện trang chi tiết phòng	70
Hình ảnh 4.3.2. 9: Giao diện trang quản lý đặt phòng	71
Hình ảnh 4.3.2. 10: Giao diện trang chi tiết thông tin đặt phòng.....	72
Hình ảnh 4.3.2. 11: Giao diện trang quản lý phòng phía admin của website	72
Hình ảnh 4.3.2. 12: Giao diện trang thêm phòng.....	73
Hình ảnh 4.3.2. 13: Giao diện trang quản lý tài khoản phía admin	74
Hình ảnh 4.3.2. 14: Giao diện trang quản lý đặt phòng	74

DANH MỤC BẢNG

Bảng 3.4.3 1: Bảng Scenario chức năng đăng nhập	41
Bảng 3.4.3 2: Bảng scenario chức năng đăng ký	41
Bảng 3.4.3 3: Bảng scenario chức năng đăng xuất	42
Bảng 3.4.3 4: Bảng scenario xem danh sách phòng.....	42
Bảng 3.4.3 5: Bảng scenario tìm kiếm phòng theo tên	43
Bảng 3.4.3 6: Bảng Scenario chức năng đặt phòng	43
Bảng 3.4.3 7: Bảng scenario huỷ phòng	44
Bảng 3.4.3 8: Bảng Scenario xem chi tiết phòng đã đặt	45
Bảng 3.4.3 9: Bảng scenario chỉnh sửa thông tin cá nhân phía khách hàng	46
Bảng 3.4.3 10: Bảng scenario thêm mới phòng phía quản trị viên.....	46
Bảng 3.4.3 11: Bảng scenario Chỉnh sửa phòng phía quản trị viên.....	47
Bảng 3.4.3 12: Bảng scenario Xem thông tin phòng phía quản trị viên	48
Bảng 3.4.3 13: Bảng scenario xoá phòng phía quản trị viên	48
Bảng 3.4.3 14: Bảng scenario xem danh sách đặt phòng phía admin	49
Bảng 3.4.3 15: Bảng scenario quản lý trạng thái đặt phòng	49
Bảng 3.4.3 16: Bảng scenario xác nhận trả phòng phía admin.....	50

CHƯƠNG I: GIỚI THIỆU CHUNG

1.1 Bối cảnh và mục tiêu

1.1.1 Tên đề tài

Tên đề tài đồ án: Phát triển website đặt phòng, quản lý khách sạn sử dụng framework Reactjs và Nodejs

1.1.2 Lý do chọn đề tài

Sau thời kỳ khó khăn do ảnh hưởng của dịch bệnh, ngành du lịch và khách sạn cũng đã bắt đầu phục hồi. Sự hồi phục này tạo ra nhu cầu lớn cho việc đặt phòng khách sạn, và với sự phát triển của internet và công nghệ thông tin việc đặt phòng trực tuyến so với cách thủ công tạo nên sự tiện lợi đối với người dùng. Phát triển một website đặt phòng, quản lý khách sạn sẽ đem lại những lợi ích:

- Tăng cường tiện ích cho người dùng: Việc phát triển một website đặt phòng và quản lý khách sạn sẽ cung cấp một nền tảng trực tuyến tiện lợi cho người dùng. Họ có thể truy cập và tìm kiếm thông tin về các khách sạn, các loại phòng khác nhau, giá cả, và các dịch vụ đi kèm một cách dễ dàng từ bất kỳ nơi đâu có kết nối Internet.
- Tối ưu hóa trải nghiệm người dùng: Phát triển một giao diện người dùng thân thiện, dễ sử dụng và tương tác trực tiếp với khách hàng sẽ cải thiện trải nghiệm người dùng. Tính năng đặt phòng trực tuyến, xem các đánh giá, hình ảnh thực tế của khách sạn, và thông tin chi tiết về các tiện ích sẽ giúp người dùng có quyết định chính xác hơn khi đặt phòng.
- Tối ưu hóa quản lý khách sạn: Hệ thống quản lý sẽ giúp chủ khách sạn quản lý thông tin về phòng trống, đặt phòng, thanh toán, dịch vụ và các hoạt động khác một cách hiệu quả. Việc tự động hóa các quy trình này giúp tiết kiệm thời gian và nguồn lực, từ đó tập trung hơn vào việc cải thiện chất lượng dịch vụ.
- Tăng tính cạnh tranh: Môi trường kinh doanh ngày càng cạnh tranh. Một website đặt phòng và quản lý khách sạn tốt sẽ giúp khách sạn thu hút khách hàng mới và giữ chân khách hàng cũ bằng cách cung cấp dịch vụ tiện ích và trải nghiệm tốt nhất.

- Cải thiện hiệu suất và tiết kiệm chi phí: Sử dụng công nghệ để quản lý khách sạn có thể giúp giảm thiểu sai sót do con người gây ra và tối ưu hóa hoạt động. Điều này có thể dẫn đến tiết kiệm chi phí và cải thiện hiệu suất toàn diện của doanh nghiệp.
- Thích ứng với xu hướng công nghệ: Trong thời đại số hóa, việc có một nền tảng trực tuyến phản ánh sự thích ứng với xu hướng công nghệ mới nhất. Điều này giúp doanh nghiệp duy trì sự hấp dẫn và tiếp cận khách hàng theo cách hiện đại nhất.

Chính vì vậy, việc phát triển một website đặt phòng và quản lý khách sạn không chỉ là về việc cung cấp tiện ích cho khách hàng mà còn về việc tối ưu hóa hoạt động nội bộ và nâng cao cạnh tranh của doanh nghiệp trong ngành du lịch và khách sạn.

1.2 Đối tượng và phạm vi nghiên cứu

1.2.1 Đối tượng nghiên cứu

a. Người dùng

Đối tượng nghiên cứu của hệ thống bao gồm hai nhóm chính là người dùng cuối (khách hàng sử dụng website để đặt phòng) và nhà quản lý khách sạn (những người sử dụng nền tảng để quản lý thông tin về khách sạn và các giao dịch).

- Người dùng cuối (khách hàng):

Khách du lịch và khách nghỉ dưỡng: Bao gồm những người đang tìm kiếm thông tin về khách sạn, cần đặt phòng cho chuyến đi du lịch hoặc kỳ nghỉ của họ. Họ mong muốn có trải nghiệm đặt phòng thuận tiện, nhanh chóng và thông tin trực tuyến chính xác về các dịch vụ, tiện ích và phòng trống.

Người sử dụng thường xuyên dịch vụ khách sạn: Đây là những người có nhu cầu sử dụng dịch vụ khách sạn thường xuyên, ví dụ như doanh nhân, người đi công tác, hay những gia đình thường xuyên di chuyển. Họ cần một giao diện dễ sử dụng và linh hoạt để quản lý các đặt phòng, theo dõi lịch sử giao dịch và nhận thông tin cập nhật về khách sạn.

- Admin quản lý hệ thống:

Quản lý khách sạn: Bao gồm những người đứng đầu quản lý khách sạn hoặc nhân viên quản lý trực tiếp. Họ cần một hệ thống quản lý thông tin toàn diện để theo dõi lịch trình đặt phòng, thông tin về khách hàng, và tối ưu hóa hoạt động của khách sạn.

Nghiên cứu đối tượng sử dụng sẽ giúp hiểu rõ hơn về nhu cầu, mong đợi và thách thức mà mỗi nhóm đối tượng phải đối mặt khi sử dụng và quản lý thông tin trên nền tảng website đặt phòng và quản lý khách sạn. Điều này sẽ giúp cải thiện và tối ưu hóa trải nghiệm của họ trên nền tảng này.

b. Công nghệ sử dụng

Công nghệ chính được sử dụng để nghiên cứu đề tài là thư viện ReactJS cho phía giao diện người dùng và môi trường thực thi mã Node.js cho phần máy chủ của ứng dụng

ReactJS là một framework phát triển front-end phổ biến, được thiết kế để xây dựng giao diện người dùng dễ sử dụng, tương tác và hiệu quả.

NodeJS, là một môi trường chạy mã JavaScript phía máy chủ, được sử dụng phổ biến trong việc phát triển back-end của các ứng dụng web.

So với các ngôn ngữ khác:

- Spring Boot (Java): Thường có cú pháp phức tạp hơn và cần nhiều đoạn mã hơn so với JavaScript. Tốc độ phát triển có thể chậm hơn so với Node.js và ReactJS.
- Laravel (PHP): Mặc dù Laravel cung cấp tốc độ phát triển nhanh và cú pháp dễ đọc, nhưng PHP không có tính linh hoạt và hiệu suất cao như Node.js và ReactJS.
- Django (Python): Django có sự linh hoạt trong việc xây dựng ứng dụng, nhưng Python thường không hiệu quả trong việc xử lý đồng thời và thời gian thực như Node.js. Django cũng có thể không linh hoạt như ReactJS trong việc xây dựng giao diện người dùng phức tạp và tương tác mượt mà.

So với Spring Boot, Laravel và Django, ReactJS và Node.js được lựa chọn cho các dự án có yêu cầu tốc độ, tính linh hoạt, và hiệu suất cao. Trong bài toán quản lý khách sạn, sự linh hoạt và hiệu suất của ReactJS và Node.js

có thể cung cấp trải nghiệm người dùng tốt và quản lý hiệu quả hệ thống khách sạn.

Sự kết hợp giữa ReactJS và Node.js trong đề tài nghiên cứu cung cấp một cơ sở linh hoạt và mạnh mẽ cho việc xây dựng ứng dụng web đa nền tảng, tối ưu hóa hiệu suất và cải thiện trải nghiệm người dùng.

1.2.2 Môi trường sử dụng

Website đặt phòng và quản lý khách sạn được sử dụng trên tất cả trình duyệt website trên máy tính như Chrome, Cốc cốc, Microsoft Edge, Safari, Firefox,...

1.2.3 Phạm vi nghiên cứu

Phạm vi nghiên cứu của đồ án xoay quanh những phần mềm, công nghệ sử dụng chính để nghiên cứu và phát triển đề tài:

- Công nghệ sử dụng website: Phần này sẽ tìm hiểu về những ngôn ngữ được sử dụng để lập trình hệ thống, những thư viện có thể sử dụng và cơ sở lý thuyết của những công nghệ đó
- Công cụ hỗ trợ phân tích thiết kế hệ thống: quá trình phân tích thiết kế hệ thống là quá trình quan trọng trước giai đoạn triển khai đối với mỗi một hệ thống, ứng dụng, bao gồm giai đoạn nghiên cứu chức năng cho tới phân tích chức năng, xác định các sơ đồ cần thiết của chức năng.
- Công nghệ triển khai và cài đặt hệ thống: thực thi và triển khai hệ thống, những công nghệ, thư viện sử dụng và lập trình sẽ được nghiên cứu cụ thể về ưu nhược điểm cũng như lý do sử dụng
- Kiểm thử hệ thống: kiểm tra về giao diện, chức năng của hệ thống để đảm bảo thân thiện đối với người dùng

1.3 Cấu trúc đồ án

Như vậy sau khi phân tích được phạm vi nghiên cứu của đồ án thì đồ án sẽ được chia thành các chương như sau:

Chương 1: Giới thiệu chung. Chương này miêu tả, đặt vấn đề bài toán và mục đích, đối tượng nghiên cứu của đề tài

Chương 2: Cơ sở lý thuyết. Chương này tập trung vào các khái niệm, lý thuyết cơ bản xoay quanh đề tài bao gồm ngôn ngữ, công cụ chính để thực hiện đề tài

Chương 3: Phân tích và thiết kế hệ thống. Chương này mô tả quá trình xây dựng

ứng dụng, từ quá trình phân tích giao diện cho tới phân tích hệ thống

Chương 4: Cài đặt và triển khai. Chương này bao gồm cách cài đặt môi trường, công cụ xây dựng và kết quả giao diện hoàn thành

Chương 5: Kết luận và định hướng nghiên cứu trong tương lai

1.4 Kết luận chương

Chương đầu tiên của đồ án đã đưa ra mục tiêu cụ thể của đề tài cũng như từng bước trong quá trình thực hiện đề tài, đây là phần được coi như kim chỉ nam trong quá trình phát triển một hệ thống website, xác định hướng đi của đề tài này.

Tóm lại, “Website đặt lịch và quản lý khách sạn sử dụng ReactJS và NodeJS” nhằm tạo ra một trải nghiệm đặt phòng thuận tiện, an toàn và linh hoạt cho khách hàng, đồng thời cung cấp cho khách sạn một cách quản lý hiệu quả và cải thiện trải nghiệm dịch vụ để tăng khả năng cạnh tranh và thúc đẩy doanh số bán hàng. Tuy nhiên do thời gian thực hiện đồ án và kiến thức của bản thân còn nhiều hạn chế nên quá trình và cách thức xây dựng website còn nhiều thiếu sót

Chương 2 sẽ nghiên cứu về cơ sở lý thuyết, các khái niệm, công nghệ xây dựng đề tài.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về một hệ thống website

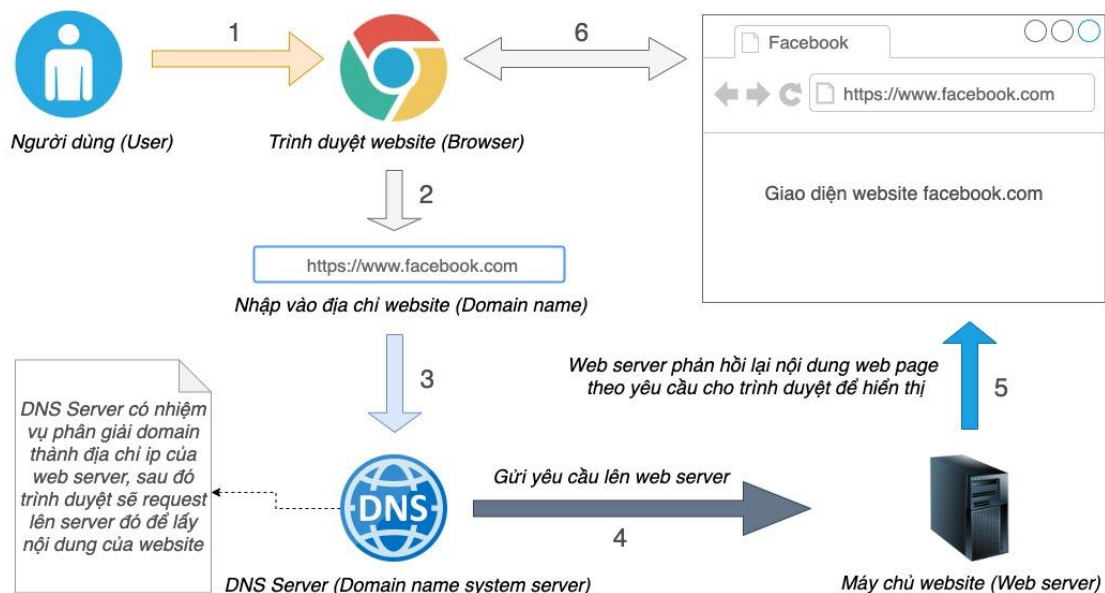
2.1.1 Khái niệm

Website còn gọi là trang web hoặc trang mạng, là một tập hợp trang thông tin có chứa các dạng hình ảnh, văn bản, video, audio... được lưu trữ trên web server (máy chủ) và người dùng có thể truy cập từ xa thông qua internet

Các ứng dụng phần mềm được sử dụng trên những thiết bị này dùng để truy cập vào trang web được gọi chung là trình duyệt web. Ví dụ như Google Chrome, Safari, Firefox, Internet Explorer, ... Một trang web được truy cập trực tiếp bằng cách nhập địa chỉ website của nó hoặc bằng cách thông qua các công cụ tìm kiếm như Google hoặc Bing...

2.1.2 Cách thức hoạt động của 1 website

Cách thức hoạt động của một website:



Hình ảnh 2.1. 1: Cách thức hoạt động của 1 website

- Người dùng gửi yêu cầu: Người dùng mở trình duyệt web và nhập URL hoặc tên miền của website mà họ muốn truy cập.
- Yêu cầu đến máy chủ web: Trình duyệt gửi yêu cầu đến máy chủ web chứa trang web tương ứng. Yêu cầu này thường sử dụng giao thức HTTP hoặc HTTPS.

- Xử lý yêu cầu: Máy chủ web nhận yêu cầu và xử lý nó. Nó tìm kiếm các tập tin và thông tin cần thiết để tạo nên trang web được yêu cầu.
- Tạo trang web: Máy chủ sử dụng thông tin từ cơ sở dữ liệu (nếu cần) cùng với mã nguồn, dữ liệu và tài nguyên được lưu trữ để tạo ra trang web cuối cùng. Đây bao gồm HTML, CSS, JavaScript, hình ảnh, video và nội dung tương tác khác.
- Trả về phản hồi: Máy chủ gửi trang web được tạo ra trở lại cho trình duyệt của người dùng dưới dạng các tệp tin HTML, CSS, JavaScript và các tài nguyên khác.
- Hiển thị trang web: Trình duyệt web nhận các tệp tin và tài nguyên từ máy chủ và hiển thị trang web cho người dùng. Trình duyệt tải và hiển thị HTML để hiển thị nội dung, CSS để định dạng và JavaScript để tạo ra các tương tác hoặc chức năng động trên trang.

2.1.3 Các thành phần của 1 website

Website bao gồm 4 thành phần cơ bản:

- + Hosting: Đây là nơi lưu trữ mã nguồn và nội dung của website. Nếu không có hosting website sẽ không thể xuất hiện trên internet và cũng không thể tiếp cận được với người dùng.
 - + Domain (tên miền): Là địa chỉ mà người dùng truy cập sử dụng để tìm ra website trên mạng lưới Internet. Website muốn hoạt động được bắt buộc phải có domain.
 - + Dữ liệu website: Dữ liệu website gọi chung cho tất cả các tập tin đa phương tiện như văn bản, âm thanh, hình ảnh, video,... được lưu trữ trên máy chủ web.
 - + Source code (mã nguồn): Là tập hợp của rất nhiều dòng lệnh khác nhau giúp tạo ra những tác vụ mà người dùng có thể thực hiện ngay trên website.
- Ngoài 4 thành phần cấu tạo chính này còn có thể kể đến một số các thành phần khác như : Bảng thông, quản trị website, giao diện, sidebar, banner,...

2.2 Tìm hiểu về HTML, CSS

2.2.1 Tìm hiểu về HTML



Hình ảnh 2.2.1 1: Tìm hiểu về HTML

HTML (HyperText Markup Language) là ngôn ngữ đánh dấu được sử dụng để tạo cấu trúc và hiển thị nội dung trên trình duyệt web. HTML cung cấp các phần tử (elements) được sắp xếp theo cấu trúc cây để mô tả cách mà nội dung của trang web được tổ chức và hiển thị.

Một tài liệu HTML được tạo thành từ nhiều thẻ HTML và mỗi thẻ HTML chứa nội dung khác nhau.

HTML là một ngôn ngữ cơ bản và quan trọng trong việc xây dựng trang web. Nó thường được kết hợp với CSS (Cascading Style Sheets) để tạo ra giao diện trực quan và JavaScript để thêm tính năng tương tác cho trang web

2.2.2 Tìm hiểu về CSS



Hình ảnh 2.2.2. 1: Tìm hiểu về CSS

CSS, viết tắt của Cascading Style Sheets, là ngôn ngữ trình bày được dùng để tạo kiểu cho sự xuất hiện của nội dung trên trang web, chẳng hạn như phông chữ hoặc màu sắc.

Đây là ngôn ngữ giúp trình duyệt hiểu được các thiết lập định dạng và bố cục cho một website. CSS cho phép điều khiển thiết kế của nhiều thành phần HTML chỉ với duy nhất 1 vùng chọn. CSS sử dụng các quy tắc, các thuộc tính và các giá trị để thay đổi diện mạo của các phần tử trong trang

2.2.3 Mối quan hệ giữa HTML và CSS



Hình ảnh 2.2.3. 1: Mối quan hệ giữa HTML và CSS

Nhìn chung:

HTML: Sử dụng để xác định cấu trúc và nội dung của trang web. Nó sử dụng thẻ và phần tử để tạo các phần trên trang như tiêu đề, đoạn văn bản, hình ảnh, liên kết, ...

CSS: Sử dụng để tạo kiểu và định dạng cho trang web. Nó sử dụng các quy tắc và thuộc tính để điều chỉnh màu sắc, kích thước, vị trí, và các yếu tố khác của các phần tử trong trang

Hai ngôn ngữ HTML và CSS hoạt động độc lập với nhau nhưng lại hỗ trợ cho nhau. Chúng đóng một vai trò rất lớn giúp website hiển thị một cách hoàn hảo nhất.

2.3 Tìm hiểu về Javascript

2.3.1 Tổng quan

JavaScript là một ngôn ngữ lập trình phía client được sử dụng để tạo ra các trang web tương tác. Nó là một trong những ngôn ngữ lập trình phía client phổ

biến nhất trên thế giới, cho phép các nhà phát triển tạo ra các ứng dụng web động, các trò chơi trực tuyến, các hiệu ứng hình ảnh và các tính năng tương tác khác trên các trang web.

JavaScript thường được sử dụng kết hợp với HTML và CSS để tạo ra các trang web đáp ứng và hiệu quả hơn. Nó có thể được sử dụng để thêm các tính năng động vào các trang web, bao gồm các hộp thoại thông báo, các hiệu ứng chuyển động, các phương thức gửi và nhận dữ liệu từ máy chủ, và rất nhiều tính năng khác.

JavaScript có thể chạy trên nhiều nền tảng khác nhau, bao gồm các trình duyệt web như Chrome, Firefox, Safari và Edge, cũng như trên máy chủ thông qua các nền tảng như Node.js.

2.3.2 Vai trò

Vai trò của JavaScript trong lập trình web:

- Tương tác trực tiếp với HTML/CSS: JavaScript cho phép thay đổi nội dung và giao diện của trang web mà không cần phải tải lại toàn bộ trang. Điều này tạo ra trải nghiệm người dùng mượt mà và nhanh chóng.
- Tương tác người dùng: JavaScript được sử dụng để xử lý sự kiện từ người dùng như click, hover, và input. Điều này giúp tạo ra trang web đáp ứng và tương tác, cung cấp trải nghiệm người dùng tốt hơn.
- Giao tiếp máy chủ: JavaScript cho phép truyền dữ liệu giữa trình duyệt và máy chủ mà không làm tải lại trang. Sử dụng AJAX (Asynchronous JavaScript and XML) hoặc các phương thức như Fetch API, JavaScript giúp tạo ra ứng dụng web linh hoạt và nhanh chóng.

2.3.3 Ưu, nhược điểm của Javascript

• Ưu điểm

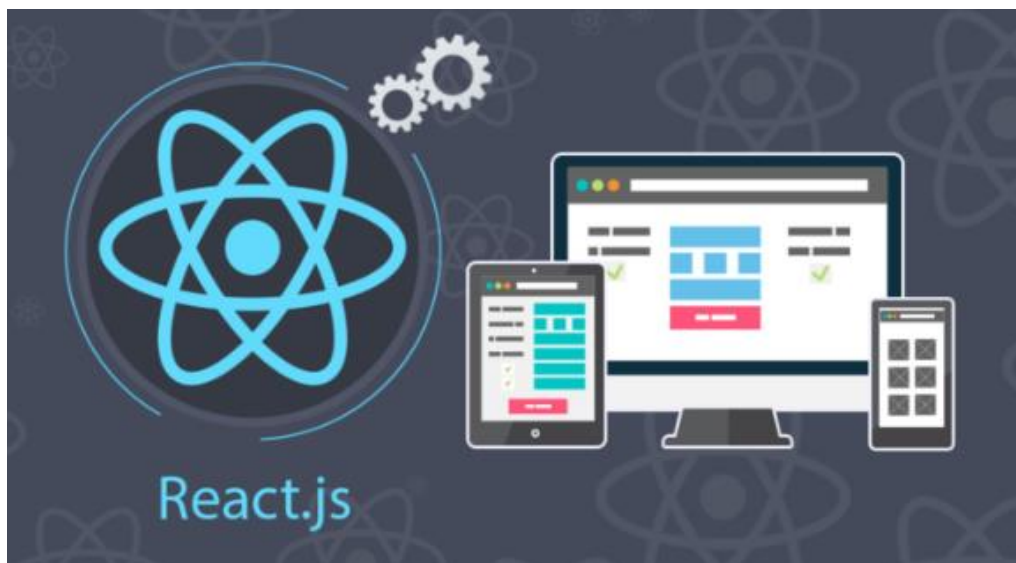
- Chương trình rất dễ học.
- Những lỗi Javascript rất dễ để phát hiện, từ đó giúp sửa lỗi một cách nhanh chóng hơn.
- Những trình duyệt web có thể dịch thông qua HTML mà không cần sử dụng đến một compiler.
- JS có thể hoạt động ở trên nhiều nền tảng và các trình duyệt web khác nhau.
- Những website có sử dụng JS thì chúng sẽ giúp cho trang web đó có sự

tương tác cũng như tăng thêm nhiều trải nghiệm mới cho người dùng.

- Người dùng cũng có thể tận dụng JS với mục đích là để kiểm tra những input thay vì cách kiểm tra thủ công thông qua hoạt động truy xuất database.
- Giao diện của ứng dụng phong phú với nhiều thành phần như Drag and Drop, Slider để cung cấp đến cho người dùng một Rich Interface (giao diện giàu tính năng).
- Giúp thao tác với người dùng phía Client và tách biệt giữa các Client với nhau.
- **Nhược điểm**
 - JS dễ bị các hacker và scammer khai thác hơn.
 - JS cũng không có khả năng đa luồng hoặc đa dạng xử lý.
 - Có thể được dùng để thực thi những mã độc ở trên máy tính của người sử dụng.
 - Những thiết bị khác nhau có thể sẽ thực hiện JS khác nhau, từ đó dẫn đến sự không đồng nhất.
 - Vì tính bảo mật và an toàn nên các Client-Side Javascript sẽ không cho phép đọc hoặc ghi các file.
 - JS không được hỗ trợ khi sử dụng ở trong tình trạng thiết bị được kết nối mạng.

2.4 Tìm hiểu về ReactJS trong thiết kế website

2.4.1 Giới thiệu về ReactJS



Hình ảnh 2.4.1. 1: Tìm hiểu về Reactjs

ReactJS là một thư viện mã nguồn mở JavaScript được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web, thư viện này chỉ tương tác với tầng giao diện của ứng dụng, đây được xem như một công cụ mạnh mẽ trong việc xử lý những thành phần giao diện.

ReactJs được phát triển bởi Jordan Walke – người từng là kỹ sư phần mềm ở Facebook. Những ngày đầu tiên, thư viện này được phát triển và duy trì bởi Facebook nhưng sau này được sử dụng lên các sản phẩm như WhatsApp, Instagram. Facebook phát triển ReactJs vào năm 2011 và chính thức được chia sẻ công khai từ đầu năm 2013

Điểm mạnh của ReactJS:

- Dễ học và dễ sử dụng: lập trình viên đã có kiến thức nền tảng về Javascript có thể hoàn toàn hiểu và sử dụng ReactJs
- Giúp việc xây dựng website động dễ dàng hơn: Việc xây dựng trang web động với HTML rất khó khăn, đòi hỏi quy trình cài đặt phức tạp, nhưng ReactJs đã giải quyết vấn đề đó.
- Có thể tái sử dụng các thành phần trong ReactJs
- Cải thiện chất lượng: cải thiện chất lượng, trải nghiệm web
- Nhiều công cụ hỗ trợ tiện ích

Lợi ích của ReactJS:

- Hiệu suất cao: ReactJS sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Virtual DOM cho phép ReactJS cập nhật các thay đổi trên trang web một cách nhanh chóng và hiệu quả hơn so với cách truyền thống, giúp tăng tốc độ và hiệu suất của ứng dụng.
- Tái sử dụng: ReactJS cho phép tái sử dụng các thành phần UI, giúp giảm thiểu thời gian và chi phí phát triển. Các thành phần UI có thể được sử dụng lại trong nhiều phần khác nhau của ứng dụng, giúp tăng tính linh hoạt và khả năng mở rộng của ứng dụng.
- Dễ dàng quản lý trạng thái: ReactJS giúp quản lý trạng thái của ứng dụng một cách dễ dàng. Sử dụng State và Props, ReactJS cho phép các nhà phát triển quản lý trạng thái của các thành phần UI một cách chính xác và dễ dàng.
- Hỗ trợ tốt cho SEO: ReactJS cho phép các nhà phát triển xây dựng ứng

dụng web với khả năng tương thích tốt với SEO. Với sự hỗ trợ của các thư viện như React Helmet, ReactJS cho phép các nhà phát triển tùy chỉnh và quản lý các phần tử meta và title cho từng trang web.

- Hỗ trợ đa nền tảng: ReactJS không chỉ được sử dụng để phát triển các ứng dụng web, mà còn được sử dụng để phát triển các ứng dụng di động với React Native. Sử dụng React Native, các nhà phát triển có thể xây dựng ứng dụng di động cho cả iOS và Android sử dụng cùng một mã nguồn.

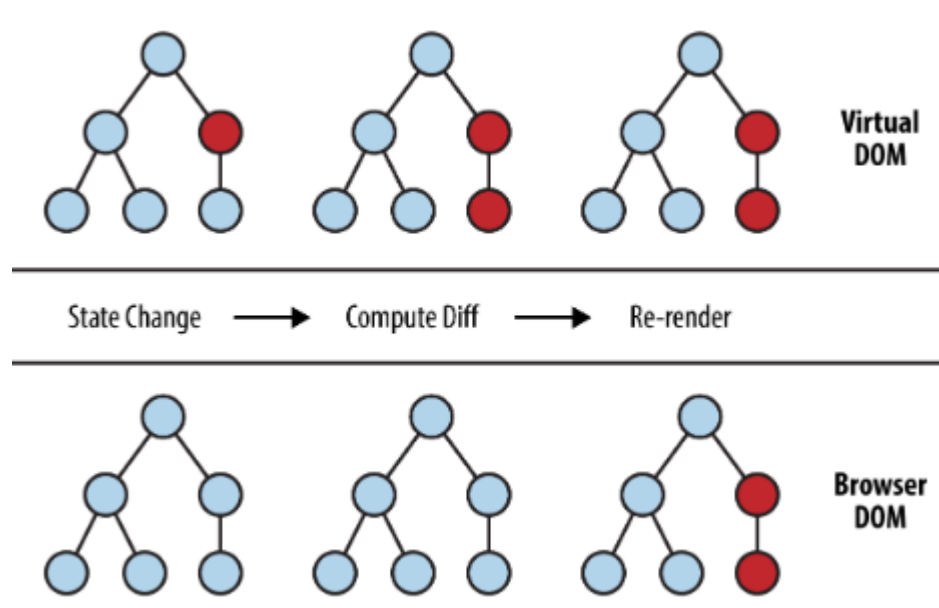
2.4.2 Cách ReactJS hoạt động

a. Khái niệm DOM và React Virtual DOM

DOM – Document Object Model có thể hiểu là những thành phần đại diện cho giao diện ứng dụng, mỗi khi có sự thay đổi về trạng thái của giao diện, DOM sẽ được cập nhật theo sự thay đổi đó

DOM được biểu hiện dưới dạng cấu trúc dữ liệu cây. Vì vậy, khi các thay đổi từ người dùng lên giao diện được gọi tới, trình duyệt hoặc ứng dụng sẽ tải lại toàn bộ

Chính vì vậy trong ReactJS xuất hiện khái niệm Virtual DOM (DOM ảo) giúp đẩy nhanh tốc độ bằng thuật toán Diff- Tạo một cấu trúc cây tương tự để kiểm tra xem thành phần nào thay đổi so với cây ban đầu, từ đó ra quyết định tải lại thay vì tải lại toàn bộ mỗi khi có thay đổi



Hình ảnh 2.4.2. 1: Virtual DOM và Browser DOM

b. Tìm hiểu về JSX

JSX là viết tắt là Javascript XML, một template languages nhưng nó lại mang hầu hết tính năng của Javascript. Nó cho phép viết các đoạn mã HTML trong React một cách dễ dàng và có cấu trúc hơn. React sử dụng JSX cho việc xây dựng bố cục thay vì javascript thông thường.

Việc sử dụng nó trong ReactJS rất hữu ích bởi:

- JSX giúp cho việc xây dựng các ứng dụng React một cách nhanh hơn, dễ tối ưu trong việc compile code sang javascript.
- JSX rất dễ xem các lỗi trong quá trình triển khai bởi hầu hết các lỗi sẽ được hiển thị trong quá trình compile, không như các đoạn mã HTML có thể thừa thiếu các thẻ div khiến giao diện bị hiển thị sai. JSX lại hoàn toàn ngược lại, nó sẽ hiển thị thông báo ngay khi gặp lỗi
- Cú pháp khá giống với HTML nên dễ dàng cho việc viết chuyển đổi.

2.4.3 Khái niệm cơ bản trong ReactJS

a. Components

Components (các thành phần) là các khối xây dựng giao diện người dùng. Chúng là các phần tử độc lập có thể tái sử dụng được tạo thành từ HTML, JavaScript, và CSS, cho phép bạn phân chia giao diện thành các phần nhỏ hơn, dễ quản lý hơn và tái sử dụng trong ứng dụng của mình.



Hình ảnh 2.4.3. 1: Components trong ReactJS

Các thành phần trong ReactJS có thể là các phần tử như button, input, form, header, footer, và cũng có thể là các phần tử phức tạp hơn như menu, trình chiếu slide, hoặc các trang web hoàn chỉnh.

Components được chia thành hai loại chính tùy theo mục đích sử dụng:

- Functional components: là một loại component được viết dưới dạng hàm JavaScript thông thường
- Class Components: là một cách để tạo các thành phần của giao diện người dùng bằng cách sử dụng cú pháp của lớp trong JavaScript. Trước khi React Hooks được giới thiệu, class components là một trong những cách chính để tạo các thành phần có trạng thái (stateful components) và thực hiện các phương thức lifecycle

b. Props

Props (viết tắt của properties - thuộc tính) là một cách để truyền dữ liệu từ một component cha (parent component) đến một component con (child component). Props là các giá trị không thể thay đổi được truyền xuống từ component cha, và các component con có thể sử dụng props này để hiển thị dữ liệu hoặc điều chỉnh hành vi của mình.

Props được truyền dưới dạng các đối tượng (object) có các thuộc tính (properties), và chúng là read-only (chỉ đọc) trong component con - nghĩa là không thể thay đổi giá trị của props từ bên trong component con.

Lập trình viên có thể tự tạo props cho các thành phần của mình hoặc sử dụng props sẵn có của những thành phần cơ bản trong ReactJS, hầu hết các component được xây dựng đều có thể tùy chỉnh props

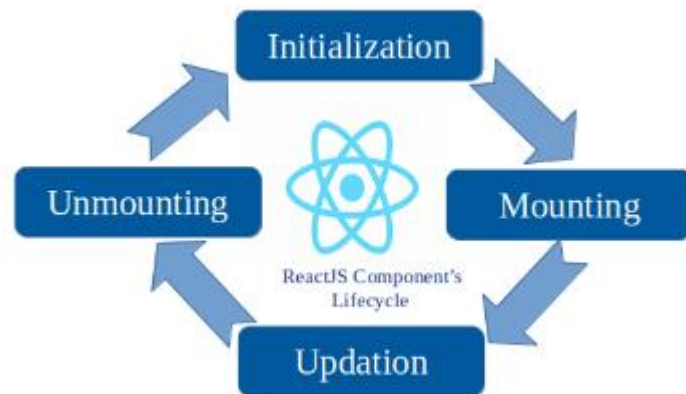
c. State

State là 1 đối tượng được sử dụng để chứa dữ liệu hoặc thông tin về components, có thể được thay đổi bất cứ khi nào mong muốn

Khác với Props có thể truyền sang các components khác nhau vì state chỉ tồn tại trong phạm vi components chứa nó, mỗi khi state thay đổi, components sẽ tải lại, lập trình viên quản lý state như một cách trả lời tương tác từ người dùng

2.4.4 Vòng đời trong ReactJS

Vòng đời (lifecycle) của một component trong ReactJS bao gồm các giai đoạn khác nhau trong quá trình mà component được tạo, cập nhật, và xóa khỏi DOM. Các phương thức lifecycle này cho phép thực hiện các hành động nhất định tại các điểm cụ thể trong vòng đời của component.



Hình ảnh 2.4.4. 1: Vòng đời của Components trong ReactJS

Trong ReatJS, vòng đời của một thành phần có thể chia thành 4 giai đoạn:

1. Initialization (Khởi tạo)

- Đây là giai đoạn mà thành phần trong Reactjs sẽ bắt đầu cuộc hành trình của nó. Nhà phát triển phải xác định các đạo cụ và trạng thái ban đầu của thành phần. Điều này thường được thực hiện trong phương thức khởi tạo

2. Mounting (Mount - Gắn kết)

- `constructor()`: Phương thức khởi tạo, được gọi đầu tiên khi một component được tạo.
- `componentDidMount()`: Được gọi sau khi component được render lần đầu tiên trong DOM. Thường được sử dụng để khởi tạo dữ liệu từ API hoặc thực hiện các tác vụ khởi tạo.

3. Updating (Cập nhật)

- `componentDidUpdate(prevProps, prevState)`: Được gọi sau khi một component đã được cập nhật và re-render. Cho phép thực hiện các tác vụ sau khi component cập nhật, như xử lý dữ liệu mới từ props hoặc state.

4. Unmounting (Unmount - Hủy bỏ)

- `componentWillUnmount()`: Được gọi trước khi component bị xóa khỏi DOM. Thường được sử dụng để "dọn dẹp", hủy các subscription hoặc xóa các tác vụ không còn cần thiết.

2.4.5 React Hooks

React Hooks là một tính năng mới được giới thiệu trong React từ phiên bản 16.8 trở lên. Hooks cho phép bạn sử dụng state và các tính năng của lifecycle trong functional components mà trước đây chỉ có thể sử dụng trong class components. Điều này giúp việc viết code trở nên ngắn gọn, dễ hiểu hơn và giúp quản lý state và lifecycle trong functional components một cách linh hoạt.

Có một số hooks phổ biến trong React:

- useState()

Hooks useState cho phép sử dụng state trong functional components. Nó trả về một cặp giá trị: giá trị hiện tại của state và một hàm để cập nhật state đó.

- useEffect()

Hooks useEffect cho phép bạn thực hiện các tác vụ sau khi component được render hoặc cập nhật. Đây tương đương với các phương thức lifecycle như componentDidMount, componentDidUpdate, và componentWillUnmount trong class components.

- useContext()

Hooks useContext cho phép truy cập giá trị của một Context trong functional components. Nó giúp truy cập các giá trị được chia sẻ mà không cần thông qua các thành phần trung gian.

- useReducer()

Hook useReducer được sử dụng để xử lý các state phức tạp và việc chia sẻ state giữa các component

2.4.6. Tìm hiểu về Redux

Redux là một thư viện quản lý trạng thái (state management) trong ứng dụng React và các ứng dụng JavaScript khác. Nó giúp quản lý trạng thái của ứng dụng một cách hiệu quả và dễ dàng, đặc biệt là khi ứng dụng có trạng thái phức tạp hoặc có nhiều thành phần cần chia sẻ dữ liệu.

Các thành phần chính trong Redux:

- Store

Store là nơi lưu trữ trạng thái của ứng dụng. Nó duy trì một trạng thái duy nhất và là nơi duy nhất để thay đổi trạng thái đó. Ta có thể truy cập state

của store bằng cách sử dụng `getState()`, cập nhật state thông qua các actions, và đăng ký để nhận thông báo khi state thay đổi.

- Actions

Actions là các đối tượng chứa thông tin về sự kiện (event) đã xảy ra trong ứng dụng. Mỗi action cần phải có một thuộc tính type để chỉ định loại hành động và có thể có các dữ liệu khác nhau để truyền đi. Actions được gửi từ ứng dụng đến store thông qua `store.dispatch()`.

- Reducers

Reducers là các pure functions chịu trách nhiệm thay đổi trạng thái của ứng dụng dựa trên các actions gửi đến. Mỗi reducer xử lý một phần của state và trả về một state mới sau khi thực hiện hành động. Reducer nhận state hiện tại và action, và không thay đổi state trực tiếp mà tạo ra một bản sao mới của state.

- Middleware (trung gian)

Middleware trong Redux là các chức năng có thể gọi trước hoặc sau khi action được gửi đến reducer. Middleware có thể thực hiện các tác vụ như ghi log, xử lý các action đặc biệt, gọi API, và nhiều công việc xử lý khác.

2.4.7 Tìm hiểu về Material UI

Material-UI là một thư viện UI (User Interface) cho ReactJS, được xây dựng dựa trên nguyên tắc thiết kế của Google Material Design. Đây là một trong những thư viện phổ biến nhất để xây dựng giao diện người dùng trong ứng dụng web React.

Đặc Điểm Chính của Material-UI:

- Thiết Kế Theo Nguyên Tắc Material Design: Material-UI cung cấp các thành phần UI sử dụng nguyên tắc của Google Material Design, giúp tạo ra giao diện hiện đại, sạch sẽ và dễ sử dụng.
- Thành Phần UI Sẵn Có (Pre-built UI Components): Bao gồm nhiều thành phần giao diện như Buttons, Dialogs, Forms, Tables, Icons, v.v. giúp việc xây dựng giao diện trở nên nhanh chóng và dễ dàng.
- Tùy Biến Linh Hoạt (Customization): Material-UI cho phép tùy chỉnh các

thành phần giao diện theo ý muốn của người dùng, từ màu sắc, kích thước đến hành vi hoạt động.

- Hỗ Trợ React Hooks: Cung cấp hỗ trợ cho React Hooks, giúp quản lý trạng thái và logic trong các thành phần giao diện.
- Đang Được Phát Triển và Cập Nhật Liên Tục: Material-UI được cộng đồng React phát triển và duy trì, có nhiều bản cập nhật mới giúp cải thiện và bổ sung tính năng, sửa lỗi và tối ưu hóa hiệu suất

Để sử dụng Material-UI trong ứng dụng React cần cài đặt thư viện `@mui/material` và `@emotion/react`, cùng với một số dependencies khác

Cài đặt: `npm install @mui/material @emotion/react @emotion/styled`

2.4.8 Tìm hiểu về axios

Axios React được sử dụng trong React để giao tiếp với các API bên ngoài và gửi các yêu cầu HTTP. Đây là một thư viện HTTP client dựa trên Promise và cung cấp một API đơn giản để thực hiện các yêu cầu AJAX và xử lý phản hồi từ máy chủ

Dưới đây là lý do tại sao Axios quan trọng trong việc xây dựng ứng dụng React:

- Dễ sử dụng: Axios cung cấp một API đơn giản và dễ hiểu cho việc gửi và nhận các yêu cầu HTTP. Điều này giúp cho việc làm việc với các yêu cầu HTTP trở nên dễ dàng và thuận tiện.
- Hỗ trợ cho Promise: Axios dựa trên Promise, cho phép xử lý các yêu cầu và phản hồi không đồng bộ. Điều này giúp tránh sự rối loạn và giữ cho mã của bạn có cấu trúc rõ ràng và dễ đọc.
- Giao tiếp với máy chủ: Trong việc xây dựng ứng dụng React, ta thường phải gửi các yêu cầu HTTP đến máy chủ để lấy dữ liệu. Axios giúp xử lý quá trình này một cách dễ dàng và cho phép bạn tương tác với API của máy chủ.
- Xử lý lỗi dễ dàng: Axios cung cấp cơ chế xử lý lỗi linh hoạt. Bạn có thể dễ dàng xử lý lỗi phản hồi từ máy chủ và thực hiện các hành động tương ứng, chẳng hạn như hiển thị thông báo lỗi hoặc chuyển hướng

người dùng đến trang khác.

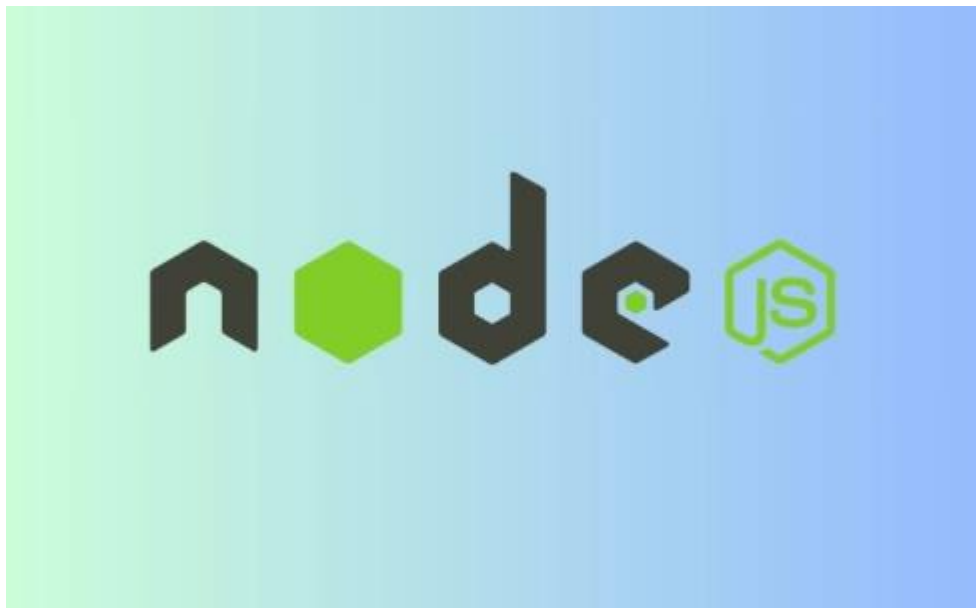
- Thư viện phổ biến: Axios là một trong những thư viện HTTP client phổ biến nhất trong cộng đồng React. Điều này có nghĩa là có nhiều tài liệu, hướng dẫn và sự hỗ trợ từ cộng đồng để bạn có thể tự tin xây dựng ứng dụng của mình.

2.5 Tìm hiểu về NodeJS trong lập trình website

2.5.1 Khái niệm

Nodejs là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng

Node.js sử dụng Google V8 JavaScript engine để thực thi mã, và một tỷ lệ lớn các mô-đun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằng JavaScript.



Hình ảnh 2.5.1. 1 NodeJS

Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.

Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao

Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing)

2.5.2 Các đặc điểm của NodeJS

Các đặc tính của NodeJS:

- Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (non-blocking), nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
- Chạy rất nhanh: NodeJS được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
- Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lập. Cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo ra hạn chế để xử lý yêu cầu.
- Không đệm: NodeJS không đệm bất kỳ một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.

2.5.3 Ưu điểm

NodeJS hiện đang được ứng dụng rất rộng rãi và trở thành lựa chọn hàng đầu trong việc xây dựng web. Điều này đã được chứng minh bởi NodeJS sở hữu những ưu điểm như sau:

- NodeJS được viết bằng JavaScript khá thông dụng và dễ dàng sử dụng trong lập trình web. Ngôn ngữ này không có khái niệm đa luồng và tất cả đều đơn luồng, hướng sự kiện.
- NodeJS chạy đa nền tảng phía Server và sử dụng kiến trúc sự kiện Event-driven với cơ chế non-blocking I/O hiệu quả hơn so với lựa chọn khác có chức năng tương tự.
- Có thể khởi chạy và sử dụng NodeJS ở bất cứ đâu dù là máy Mac, Windows hay Linux. Hơn nữa, cộng đồng người dùng NodeJS rất lớn với tài nguyên miễn phí cho người dùng thỏa sức trải nghiệm.
- Các ứng dụng của NodeJS có thể đáp ứng tốt thời gian thực và dễ dàng chạy

đa nền tảng, đa thiết bị mà thời gian hiển thị phù hợp và chuẩn xác, cho người dùng có thể khai thác một cách tiện lợi và hiệu quả.

2.5.4 Tìm hiểu về npm

“npm” là viết tắt của "Node Package Manager", là một trình quản lý gói (package manager) cho Node.js. Nó được sử dụng để quản lý các gói, thư viện, module, và các công cụ được sử dụng trong phát triển ứng dụng Node.js.

Chức năng và tính năng chính của npm:

- Quản lý Gói (Packages): npm cho phép người dùng cài đặt, cập nhật và xóa các gói Node.js thông qua dòng lệnh.
- Công cụ Tạo Ứng Dụng (CLI): npm cung cấp một command-line interface (CLI) để thực hiện các thao tác như cài đặt, cập nhật, gỡ bỏ các gói, quản lý scripts, v.v.
- Cơ sở Dữ Liệu Tự Động (Registry): npm Registry lưu trữ các gói Node.js có sẵn để tải về. Gói mặc định của npm là npmjs.com.
- Công Cụ Quản Lý Phiên Bản (Version Management): npm hỗ trợ quản lý và xác định các phiên bản của các gói Node.js thông qua Semantic Versioning

2.5.5 Tìm hiểu về JWT

JWT (JSON Web Token) là một tiêu chuẩn mở để tạo ra một cách an toàn để truyền thông tin giữa hai bên dưới dạng JSON. Trong ngữ cảnh của xác thực người dùng, JWT thường được sử dụng để tạo token khi người dùng đăng nhập thành công. Token này sau đó được gửi với mỗi yêu cầu từ client đến server để chứng thực người dùng.

Đặc điểm

JWT có đặc điểm chính là kích thước nhỏ, dễ parse, và có thể chứa các thông tin tùy chỉnh. Token này có thể được ký và xác thực để đảm bảo tính toàn vẹn của nó

Cài đặt: `npm i jsonwebtoken`

2.6 Tìm hiểu về hệ quản trị CSDL MongoDB

2.6.1 Khái niệm

MongoDB là phần mềm cơ sở dữ liệu mã nguồn mở NoSQL, được thiết kế hướng theo đối tượng và hỗ trợ trên đa nền tảng. Các bảng MongoDB có cấu trúc

linh hoạt, cho phép dữ liệu không cần tuân theo bất kỳ dạng cấu trúc nào.



Hình ảnh 2.6.1 1 MongoDB

MongoDB hoạt động trên collection, hướng tài liệu kiểu JSON thay cho bảng để tăng tốc độ truy vấn. MongoDB có chức năng định hướng tài liệu cung cấp, hiệu suất cao, tính sẵn sàng cao và khả năng mở rộng dễ dàng.

Collection trong MongoDB về bản chất thì có thể hiểu là nhóm các document, một collection sẽ chứa các tập document.

2.6.2 Ưu điểm

- MongoDB là một cơ sở dữ liệu hướng tài liệu, sử dụng bộ nhớ nội tại, truy cập dễ dàng nhờ vào việc lập ra các chỉ mục, tăng tốc độ phản hồi truy vấn nhanh. Theo đánh giá thì tốc độ MongoDB có thể nhanh hơn 100 lần so với cơ sở dữ liệu quan hệ.
- Sự linh hoạt của cơ sở dữ liệu: MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi collection sẽ có kích cỡ khác nhau và các document cũng khác nhau. Do sử dụng cơ sở dữ liệu không có lược đồ nên điều này mang lại sự linh hoạt và tự do lưu trữ dữ liệu thuộc nhiều loại khác nhau.
- Khả năng mở rộng: Lợi thế về cơ sở dữ liệu theo chiều ngang, vì vậy, khi xử lý một dữ liệu lớn thì chúng ta có thể phân phối cho nhiều máy.
- Đội ngũ hỗ trợ chuyên nghiệp: Khi gặp phải bất kỳ sự cố nào chúng ta có thể liên hệ trực tiếp đến hệ thống hỗ trợ để xử lý kịp thời.
- Tính khả dụng cao: MongoDB không chỉ có các tính năng về sao chép mà có thể

sử dụng gridFS (có thể lưu trữ và truy xuất các tệp vượt quá kích thước 16MB).
Các tính năng giúp tăng tính khả dụng và đạt hiệu suất cao.

2.7 Tìm hiểu về Postman

2.7.1 Khái niệm

Postman là một công cụ (tool) phổ biến được sử dụng trong quá trình phát triển và kiểm thử các API (Application Programming Interface). Đây là một ứng dụng desktop cho phép người dùng tạo, gửi và kiểm thử các yêu cầu HTTP (như GET, POST, PUT, DELETE) đến các endpoint khác nhau và kiểm tra các phản hồi từ server.



Hình ảnh 2.7.1. 1 Postman

2.7.2 Đặc điểm của Postman

- Tạo và Gửi Yêu Cầu HTTP: Postman cho phép người dùng tạo và gửi các yêu cầu HTTP đến các endpoint khác nhau của API.
- Kiểm Tra và Quản Lý Yêu Cầu: Nó cho phép người dùng kiểm tra các thông số của yêu cầu HTTP, như headers, body, parameters, v.v.
- Kiểm Tra Phản Hồi từ Server: Postman hiển thị phản hồi từ server sau khi gửi yêu cầu, cho phép người dùng kiểm tra và phân tích dữ liệu trả về.
- Tạo và Quản Lý Môi Trường (Environment): Cung cấp khả năng tạo và quản lý các môi trường để cấu hình thông tin môi trường (environment variables) như URLs, tokens, v.v.
- Kiểm Thử Tự Động (Automated Testing): Postman hỗ trợ việc viết các kịch

bản kiểm thử (test scripts) để kiểm tra và đánh giá tính đúng đắn của các API endpoints.

- Chia Sẻ và Hợp Tác (Sharing and Collaboration): Cho phép người dùng chia sẻ các bộ sưu tập (collection) yêu cầu API và làm việc cùng nhau trên các dự án.

2.8 Kết luận chương

Kết thúc chương 2 về cơ sở lý thuyết giới thiệu về tổng quan của một hệ thống website, các ngôn ngữ lập trình website và việc chuẩn bị cho chương tiếp theo về phân tích thiết kế hệ thống website đặt phòng khách sạn đã cung cấp cái nhìn tổng quan vững chắc về nền tảng của một trang web hiện đại.

Qua chương này, đã rõ sự quan trọng của việc thiết kế và xây dựng một hệ thống website. Cơ sở lý thuyết đã giới thiệu về các thành phần cần thiết của một trang web, từ cấu trúc tổng quan đến các ngôn ngữ lập trình quan trọng như HTML, CSS và JavaScript. Mỗi ngôn ngữ này đóng vai trò quan trọng trong việc tạo ra trải nghiệm người dùng tốt hơn và cách chúng tương tác với nhau để xây dựng nền tảng một trang web linh hoạt và đa dạng.

Chương tiếp theo sẽ đi sâu vào phân tích thiết kế hệ thống website đặt phòng khách sạn. Tập trung vào việc phân tích yêu cầu, thiết kế cấu trúc hệ thống và xác định các tính năng chính cần có để xây dựng một hệ thống đặt phòng khách sạn hiệu quả và dễ sử dụng.

CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1 Khảo sát và phân tích người dùng

3.1.1 Mục tiêu của website

- Tăng doanh số số lượng đặt phòng và doanh thu từ các đặt phòng trực tuyến.
- Tăng sự tiện lợi cho khách hàng: Cung cấp một giao diện dễ sử dụng và hiệu quả để khách hàng có thể dễ dàng đặt phòng, thay đổi thông tin đặt phòng, và thanh toán trực tuyến.
- Xây dựng uy tín và tin cậy: Cung cấp thông tin chi tiết và chính xác về các phòng, dịch vụ và tiện nghi để xây dựng lòng tin từ phía khách hàng.

3.1.2 Đối tượng chính của website đặt phòng khách sạn

Các đối tượng người dùng sử dụng website đặt phòng khách sạn

- Khách hàng cuối cùng (Người sử dụng chính): Đây là nhóm người dùng chính sử dụng website để tìm kiếm thông tin, đặt phòng và thanh toán. Họ có thể là du khách cá nhân, du lịch gia đình, người đi công tác hoặc nhóm du lịch. Nhu cầu của họ bao gồm việc tìm kiếm chỗ ở phù hợp với ngân sách, tiện ích, vị trí của khách sạn.
- Quản trị viên khách sạn: Đối với nhóm này, họ cần truy cập vào hệ thống để quản lý phòng, xác nhận đơn đặt phòng, cập nhật thông tin về phòng trống và giá cả.

3.1.3 Nghiên cứu đối thủ cạnh tranh

- Các trang web đặt phòng khác: Xác định và phân tích các trang web đặt phòng khác trong cùng lĩnh vực. Điều này bao gồm cả các đối thủ lớn như Booking.com, Expedia, và cả các đối thủ địa phương.
- Các khách sạn cùng kích thước và loại hình: Nghiên cứu các khách sạn cạnh tranh trực tiếp hoặc giống nhau về kích thước và loại hình để hiểu cách họ tiếp cận và giữ chân khách hàng.
- Phân tích chiến lược và ưu điểm cạnh tranh: Xem xét chiến lược tiếp thị, ưu đãi, chính sách giá và các yếu tố khác để định rõ những gì đối thủ đang làm tốt và những điểm yếu cần tận dụng.

Quá trình nghiên cứu này giúp xác định được những điểm mạnh và yếu của trang web đặt phòng, cũng như cung cấp cơ sở để phát triển chiến lược marketing và phục vụ khách hàng hiệu quả hơn.

3.2 Kiến trúc hệ thống

3.1.2 Kiến trúc của hệ thống Client – Server

Mô hình client – server bao gồm một máy chủ và nhiều máy khách kết nối với máy chủ đó, đặc điểm của mô hình này là máy khách gửi yêu cầu tới và đợi phản hồi từ máy chủ

Ưu điểm của kiến trúc này là các truy cập được bảo mật, giúp cho việc chia sẻ dữ liệu dễ dàng hơn khi ở xa, không phụ thuộc và công nghệ, ngôn ngữ lập trình, bảo trì dễ dàng

Nhược điểm lớn nhất là sức chịu tải của server khi có quá nhiều yêu cầu từ nhiều máy khách

- Server: Bao gồm cơ sở dữ liệu và hệ thống cung cấp api được cung cấp
- Client: Giao diện xây dựng bởi ReactJs + NodeJS và CSDL MongoDB

3.2.2 Phương thức giao tiếp giữa Client – Server

API (Application Programming Interface) là phương thức giao tiếp giữa phần mềm này với phần mềm khác, hay còn gọi là giao diện lập trình ứng dụng

Web API là 1 phương thức được dùng để cho phép các ứng dụng khác nhau dễ dàng giao tiếp, trao đổi thông tin, dữ liệu được web api trả lại ở dạng JSON (Javascript Object Notation) thông qua giao thức HTTPS hoặc HTTP

3.3 Công nghệ triển khai

3.3.1 Phía backend

- Node.js là một nền tảng được xây dựng dựa trên JavaScript runtime, cho phép chạy mã JavaScript không chỉ trên trình duyệt mà còn trên máy chủ. Điều này mang lại ưu điểm lớn về tính đồng nhất trong cả frontend và backend của trang web.
- Express.js: Express.js là một framework Node.js mạnh mẽ và linh hoạt, được sử dụng rộng rãi để xây dựng các ứng dụng web và API. Nó cung cấp các tính năng đơn giản nhưng mạnh mẽ để xử lý yêu cầu HTTP, định tuyến (routing), middleware, và quản lý session.

- Cơ sở dữ liệu: Trong việc phân tích công nghệ backend, việc lựa chọn cơ sở dữ liệu cũng rất quan trọng. Node.js và Express hỗ trợ nhiều loại cơ sở dữ liệu như MongoDB (NoSQL), MySQL, PostgreSQL, và các hệ thống khác. Sự lựa chọn phụ thuộc vào yêu cầu cụ thể của dự án.
- RESTful APIs: Khi xây dựng backend, việc tạo ra các RESTful APIs (Application Programming Interfaces) để truy cập và tương tác với dữ liệu từ frontend là một phần quan trọng. Express cung cấp cơ chế để xây dựng các API này một cách dễ dàng và hiệu quả.
- Authentication và Authorization: Để bảo vệ và quản lý quyền truy cập, sử dụng các công nghệ như JSON Web Tokens (JWT) hoặc OAuth để xác thực người dùng và quản lý quyền truy cập.
- Async programming: Vì Node.js là môi trường lập trình không đồng bộ (asynchronous), việc hiểu và sử dụng các kỹ thuật lập trình không đồng bộ là rất quan trọng để tối ưu hóa hiệu suất của ứng dụng.

3.3.2 Phía client

- Thành phần giao diện người dùng: ReactJS là một thư viện JavaScript cho việc xây dựng giao diện người dùng. Nó tập trung vào việc tạo ra các thành phần (components) tái sử dụng, giúp quản lý và hiển thị dữ liệu theo cách linh hoạt và dễ dàng.
- Virtual DOM: ReactJS sử dụng Virtual DOM để cải thiện hiệu suất. Thay vì cập nhật toàn bộ DOM mỗi khi dữ liệu thay đổi, React sử dụng Virtual DOM để tìm ra sự thay đổi thực sự và chỉ cập nhật những phần cần thiết trong DOM, giúp tăng tốc độ hiển thị trang web.
- Redux: Quản lý trạng thái ứng dụng: Redux là một thư viện quản lý trạng thái cho ứng dụng JavaScript. Nó giúp quản lý trạng thái của toàn bộ ứng dụng trong một store duy nhất, làm cho việc theo dõi và cập nhật trạng thái trở nên dễ dàng hơn.

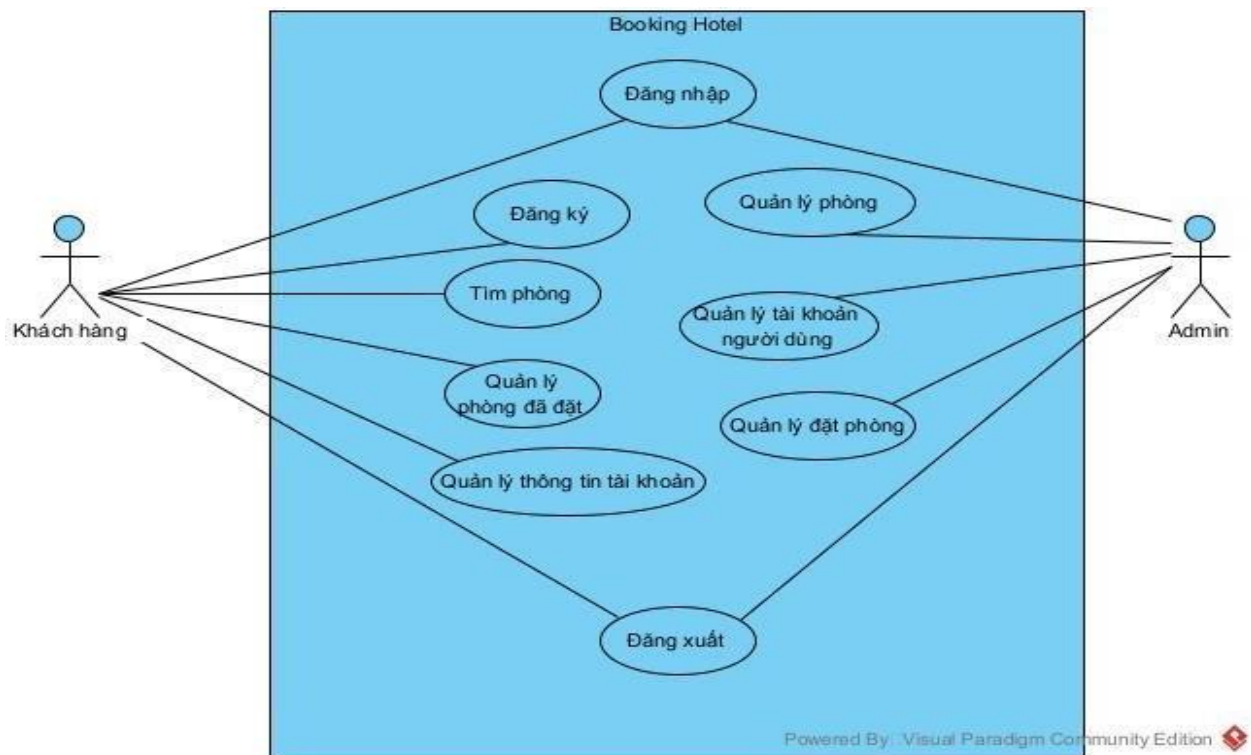
3.4 Phân tích thiết kế hệ thống

3.4.1 Mô tả các chức năng trong hệ thống

- Đăng nhập: Usecase cho phép thành viên trong hệ thống đăng nhập vào tài khoản của mình bằng email và password
- Đăng ký: Usecase cho phép người dùng tạo tài khoản để truy cập và hệ thống
- Tìm kiếm phòng: tìm kiếm thông tin phòng muốn đặt
- Quản lý đặt phòng: usecase cho phép người quản lý các phòng đã đặt bao gồm: xem, thêm, huỷ phòng
- Quản lý thông tin tài khoản: usecase cho phép người dùng xem, cập nhật thông tin, thay đổi mật khẩu
- Đăng xuất: usecase cho phép người dùng đăng xuất khỏi hệ thống để kết thúc phiên làm việc
- Quản lý phòng: Usecase cho phép admin quản lý phong khách sạn: xem, thêm, sửa, xoá phòng
- Quản lý tài khoản người dùng: Usecase cho phép admin tạo và quản lý, các tài khoản trong hệ thống
- Quản lý đặt phòng: Usecase cho phép admin quản lý các phòng mà khách hàng đã đặt bao gồm: giao phòng, trả phòng, huỷ đặt phòng

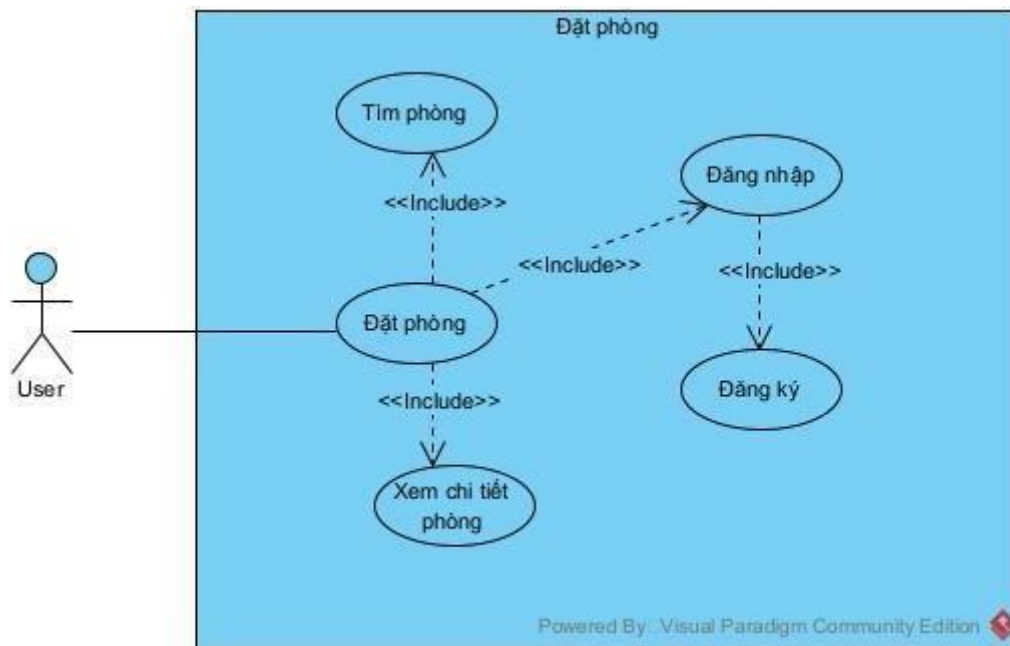
3.4.2 Sơ đồ use case hệ thống

- Usecase toàn bộ hệ thống



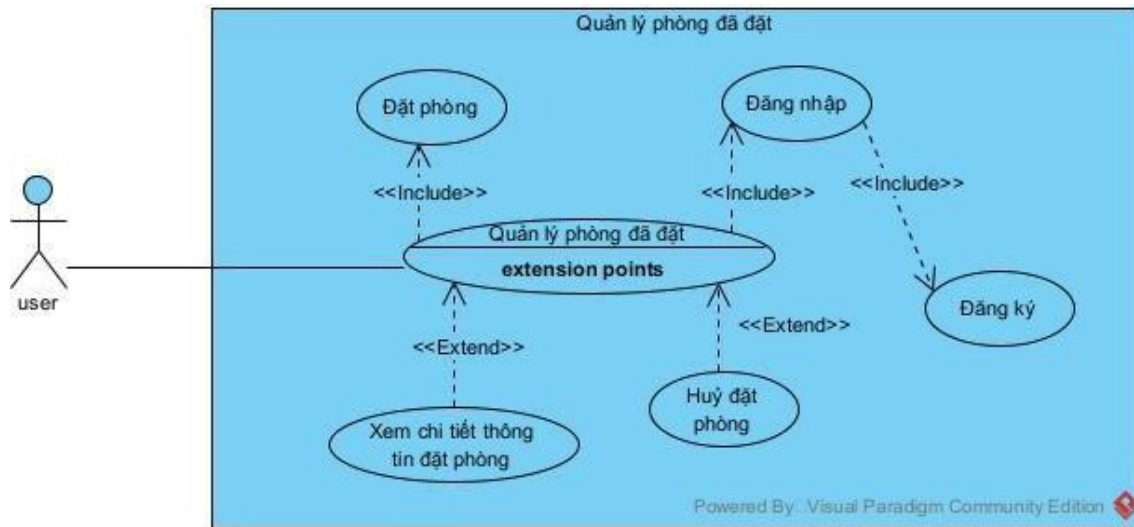
Hình ảnh 3.4.2. 1: Sơ đồ usecase hệ thống website đặt phòng khách sạn

- Usecase chức năng đặt phòng khách sạn phía khách hàng



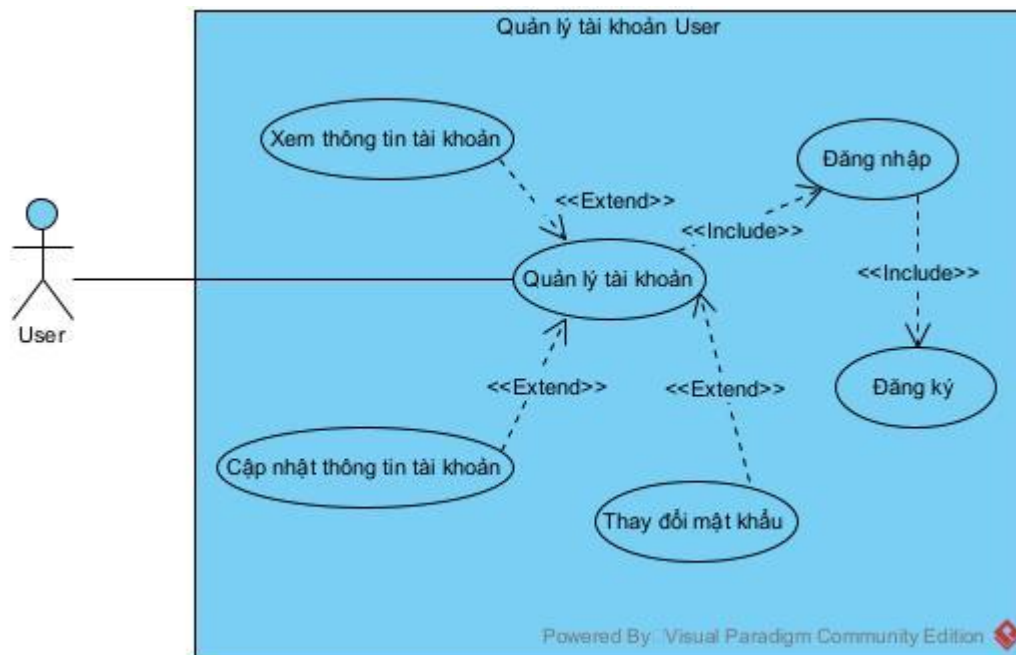
Hình ảnh 3.4.2. 2: Usecase chức năng đặt phòng phía khách hàng

- Usecase chức năng quản lý các phòng đã đặt phía khách hàng



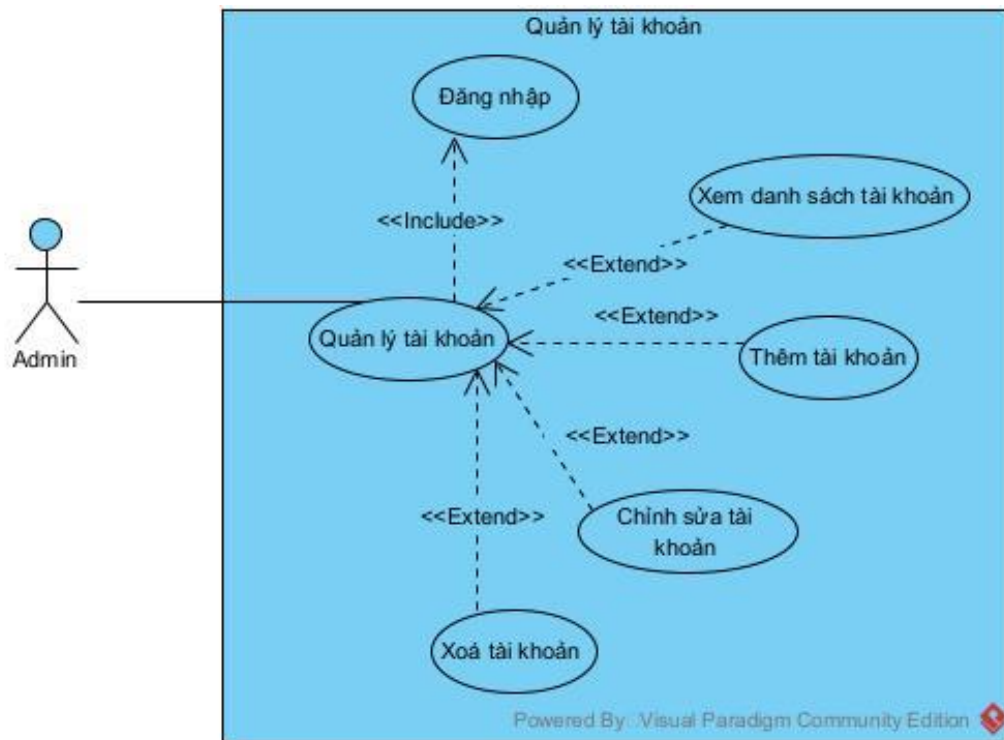
Hình ảnh 3.4.2. 3: Usecase chức năng quản lý các phòng đã đặt phía khách hàng

- Usecase Quản lý thông tin tài khoản của người dùng phía user



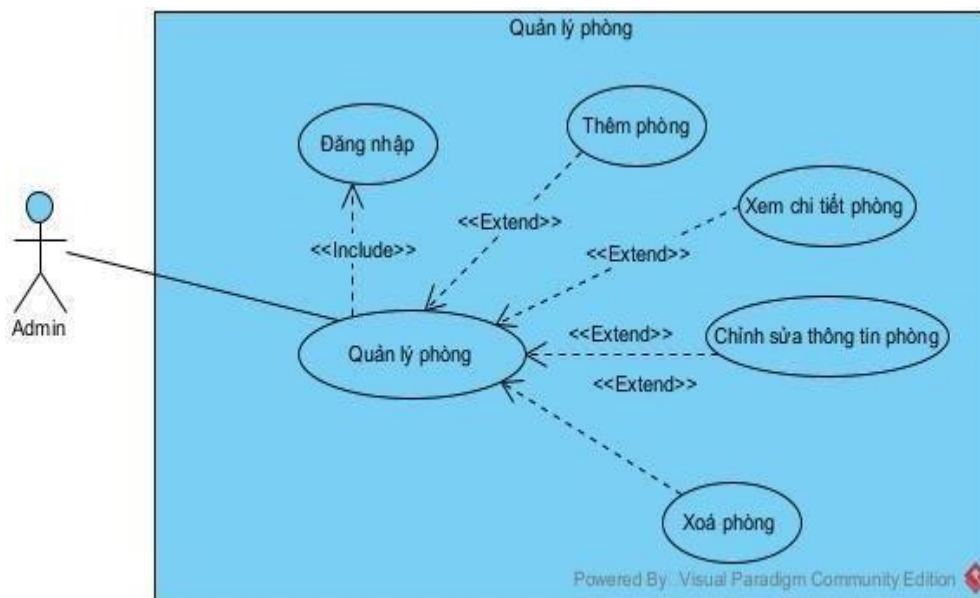
Hình ảnh 3.4.2. 4: Usecase quản lý tài khoản phía user

- Usecase quản lý tài khoản của người dùng phía admin



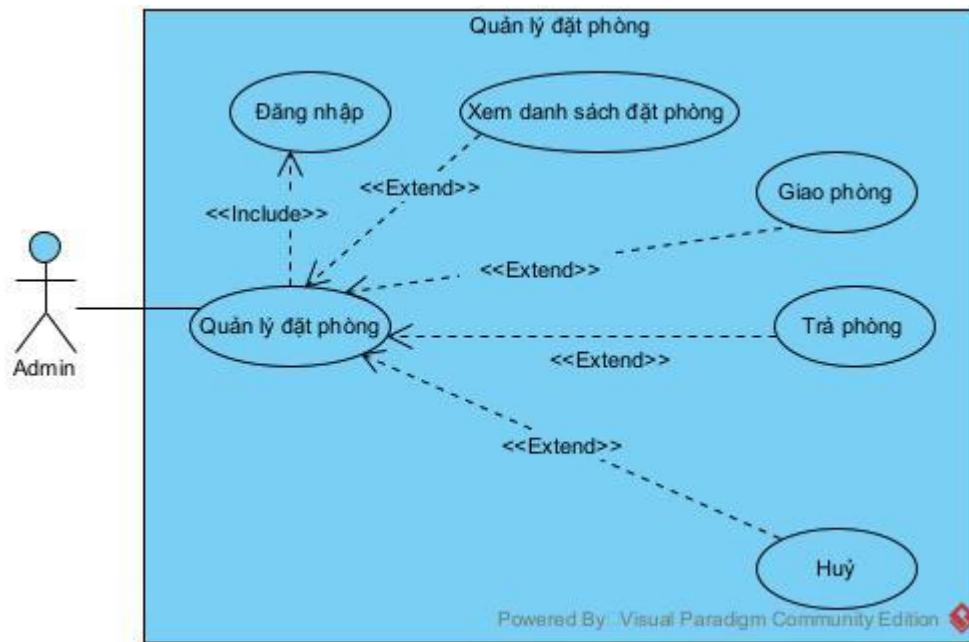
Hình ảnh 3.4.2. 5: Usecase quản lý tài khoản người dùng phía admin

- Usecase quản lý phòng của admin



Hình ảnh 3.4.2. 6: Usecase quản lý phòng phía admin

- Usecase quản lý đặt phòng phía admin



Hình ảnh 3.4.2. 7: Usecase quản lý đặt phòng phía admin

3.4.3 Scenario

- Scenario chức năng Đăng nhập

Tên	Đăng nhập
Tác nhân	Khách hàng
Mô tả mục tiêu	Người dùng đăng nhập vào hệ thống thành công
Yêu cầu liên quan	Usecase Đăng ký
Tiền điều kiện	Người dùng đã có tài khoản Người dùng phải chọn chức năng đăng nhập
Mô tả	1. Người dùng nhập Email, password 2. Người dùng chọn button Log In Hệ thống kiểm tra thông tin đầu vào
Hậu điều kiện	Người dùng đăng nhập thành công, hệ thống hiển thị giao diện trang chủ
Biến thể	Người dùng muốn tạo tài khoản mới -> Usecase Đăng ký
Ngoại lệ	TH1: Người dùng nhập thiếu email hoặc password 1. Hệ thống báo nhập thiếu email hoặc password 2. Quay lại bước 1

	<p>TH2: Người dùng nhập sai email hoặc password</p> <p>1. Hệ thống báo nhập sai email hoặc password</p> <p>Quay lại bước 1</p>
--	--

Bảng 3.4.3 1: Bảng Scenario chức năng đăng nhập

- Scenario Đăng ký

Tên	Đăng ký
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Người dùng đăng ký tài khoản thành công
Yêu cầu liên quan	Không
Tiền điều kiện	Người dùng phải chọn chức năng đăng ký
Mô tả	<p>1. Người dùng nhập username,email, password</p> <p>2. Người dùng chọn đăng ký</p> <p>3. Hệ thống kiểm tra thông tin đầu vào</p> <p>4. Hệ thống kiểm tra sự tồn tại của tài khoản</p> <p>5. Hiện thị thông báo đăng ký thành công</p>
Hậu điều kiện	Đăng ký tài khoản thành công
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 2: Bảng scenario chức năng đăng ký

- Scenario Đăng xuất

Tên	Đăng xuất
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Người dùng có thể đăng xuất
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Người dùng đã ở trong hệ thống
Mô tả	<p>1. Người dùng chọn Log out</p> <p>2. Hệ thống kiểm tra điều kiện đăng xuất</p> <p>3. Đăng xuất khỏi hệ thống</p>

Hậu điều kiện	Đăng xuất tài khoản thành công
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 3: Bảng scenario chức năng đăng xuất

- Scenario Xem phòng

Tên	Xem xem phòng
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Người dùng có thể xem các phòng được đăng trên hệ thống
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Người dùng đăng nhập vào hệ thống
Mô tả	1. Người dùng truy cập vào hệ thống 2. Hệ thống lấy dữ liệu từ DB Hệ thống hiển thị danh sách phòng ra màn hình
Hậu điều kiện	Người dùng xem được thông tin hệ thống hiển thị
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 4: Bảng scenario xem danh sách phòng

- Scenario Tìm kiếm phòng

Tên	Tìm kiếm phòng theo tên
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Người dùng tìm kiếm phòng theo tên thành công
Yêu cầu liên quan	Không
Tiền điều kiện	Không
Mô tả	1. Người dùng nhập thông tin tìm kiếm vào thanh tìm kiếm 2. Người dùng nhấn Enter 3. Hệ thống hiển thị danh sách các phòng tương ứng

Hậu điều kiện	Người dùng tìm kiếm phòng thành công
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 5: Bảng scenario tìm kiếm phòng theo tên

- Scenario Đặt phòng

Tên	Đặt phòng
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Người dùng đặt phòng thành công
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Người dùng đăng nhập vào hệ thống
Mô tả	<ol style="list-style-type: none"> 1. Tại giao diện danh sách phòng, người dùng nhấn vào xem chi tiết phòng 2. Người dùng nhập thông tin cá nhân, ngày đặt phòng, ngày trả phòng, chọn phòng, nhập số lượng phòng, nhấn vào nút Đặt phòng <p>Hệ thống hiển thị thông báo đặt phòng thành công và chuyển sang giao diện danh sách phòng đã đặt</p>
Hậu điều kiện	Người dùng đặt phòng thành công
Biến thể	Không
Ngoại lệ	Phòng đã có người khác đặt

Bảng 3.4.3 6: Bảng Scenario chức năng đặt phòng

- Scenario Huỷ phòng

Tên	Huỷ đặt phòng
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Người dùng huỷ phòng thành công
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đăng nhập vào hệ thống - Người dùng đặt phòng thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại giao diện trang chủ người dùng nhấn vào Các phòng đã đặt ở trên thanh header 2. Hệ thống hiển thị danh sách các phòng mà người dùng đã đặt 3. Người dùng nhấn vào button Huỷ ở một phòng đã đặt 4. Hệ thống hiển thị thông báo xác nhận huỷ phòng 5. Người dùng nhấn nút Huỷ 6. Hệ thống hiển thị thông báo Huỷ phòng thành công
Hậu điều kiện	Admin chưa xác nhận thông tin đặt phòng
Biến thể	Không
Ngoại lệ	Phòng đã có người khác đặt

Bảng 3.4.3 7: Bảng scenario huỷ phòng

- Scenario Xem chi tiết phòng đã đặt

Tên	Xem chi tiết phòng đã đặt
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Người dùng xem thông tin chi tiết về phòng đã đặt
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	<ul style="list-style-type: none"> - Người dùng đăng nhập vào hệ thống - Người dùng đặt phòng thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại giao diện trang chủ người dùng nhấn vào Các phòng đã đặt ở trên thanh header 2. Hệ thống hiển thị danh sách các phòng mà người dùng đã đặt 3. Người dùng nhấn vào button Chi tiết ở một phòng đã đặt 4. Hệ thống hiển thị thông tin chi tiết về phòng mà người dùng đặt
Hậu điều kiện	Người dùng đặt phòng thành công
Biến thể	Không
Ngoại lệ	Phòng đã có người khác đặt

Bảng 3.4.3 8: Bảng Scenario xem chi tiết phòng đã đặt

- Scenario Chỉnh sửa thông tin cá nhân

Tên	Chỉnh sửa thông tin cá nhân
Tác nhân	Cá nhân người dùng
Mô tả mục tiêu	Chỉnh sửa thành công thông tin cá nhân
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Người dùng đăng nhập thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại thanh header người dùng nhấn vào tài khoản 2. Hệ thống hiển thị giao diện thông tin cá nhân 3. Người dùng nhấn vào Cập nhật thông tin ở sidebar phía bên trái màn hình 4. Hệ thống hiển thị giao diện thay đổi thông tin cá nhân 5. Người dùng nhập các thông tin cần chỉnh sửa: họ, tên đệm và tên, số điện thoại

	6. Người dùng nhấn nút Cập nhật 7. Hệ thống lưu thông tin chỉnh sửa thành công
Hậu điều kiện	Người dùng chỉnh sửa thành công thông tin cá nhân
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 9: Bảng scenario chỉnh sửa thông tin cá nhân phía khách hàng

- Scenario thêm phòng ở quản trị viên

Tên	Thêm mới phòng
Tác nhân	Quản trị viên
Mô tả mục tiêu	Thêm mới danh mục sản phẩm
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Quản trị viên đăng nhập thành công bằng tài khoản admin
Mô tả	1. Tại thanh sidebar Quản trị viên chọn Quản lý phòng 2. Hệ thống hiển thị danh sách phòng có trong hệ thống 3. Quản trị viên chọn button Thêm phòng 4. Hệ thống hiển thị giao diện Thêm phòng 5. Quản trị viên nhập các thông tin về phòng và nhấn button Thêm phòng 6. Hệ thống hiển thị thông tin về phòng được thêm ở giao diện danh sách phòng
Hậu điều kiện	Người dùng thêm mới thành công phòng
Biến thể	Không
Ngoại lệ	Người dùng không điền đầy đủ thông tin vào các trường bắt buộc =>Thông báo lỗi.

Bảng 3.4.3 10: Bảng scenario thêm mới phòng phía quản trị viên

- Scenario Chỉnh sửa phòng ở Quản trị viên

Tên	Chỉnh sửa phòng
Tác nhân	Quản trị viên

Mô tả mục tiêu	Chỉnh sửa phòng
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Người dùng đăng nhập thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại thanh sidebar Quản trị viên chọn Quản lý phòng 2. Hệ thống hiển thị danh sách phòng có trong hệ thống 3. Quản trị viên chọn button Chỉnh sửa 4. Hệ thống hiển thị giao diện Chỉnh sửa thông tin phòng 5. Quản trị viên chỉnh sửa các thông tin về phòng và nhấn button Cập nhật 6. Hệ thống hiển thị thông tin về phòng được chỉnh ở giao diện danh sách phòng
Hậu điều kiện	Người dùng chỉnh sửa thành công phòng
Biến thể	Không
Ngoại lệ	Người dùng không điền đầy đủ thông tin vào các trường bắt buộc => Thông báo lỗi.

Bảng 3.4.3 11: Bảng scenario Chỉnh sửa phòng phía quản trị viên

- Scenario xem phòng ở Quản trị viên

Tên	Xem phòng
Tác nhân	Quản trị viên
Mô tả mục tiêu	Xem phòng
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Người dùng đăng nhập thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại thanh sidebar Quản trị viên chọn Quản lý phòng 2. Hệ thống hiển thị danh sách phòng có trong hệ thống 3. Quản trị viên chọn button Xem 4. Hệ thống hiển thị giao diện thông tin chi tiết về phòng

Hậu điều kiện	Thêm phòng thành công
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 12: Bảng scenario Xem thông tin phòng phía quản trị viên

- Scenario Xóa phòng

Tên	Xóa phòng
Tác nhân	Quản trị viên
Mô tả mục tiêu	Xóa phòng
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Quản trị viên đăng nhập thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại thanh sidebar Quản trị viên chọn Quản lý phòng 2. Hệ thống hiển thị danh sách phòng có trong hệ thống 3. Quản trị viên chọn button Xóa 4. Hệ thống hiển thị thông báo xác nhận xóa phòng 5. Quản trị viên nhấn vào button Xóa 6. Xóa thành công sản phẩm khỏi hệ thống
Hậu điều kiện	Thêm phòng thành công
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 13: Bảng scenario xóa phòng phía quản trị viên

- Scenario Xem danh sách đặt phòng

Tên	Xem danh sách đặt phòng
Tác nhân	Quản trị viên
Mô tả mục tiêu	Xem danh sách đặt phòng của người dùng
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Quản trị viên đăng nhập thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại thanh sidebar Quản trị viên chọn Quản lý đặt phòng

	2. Hệ thống hiển thị danh sách phòng đã đặt
Hậu điều kiện	Không
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 14: Bảng scenario xem danh sách đặt phòng phía admin

- Scenario chức năng quản lý trạng thái đặt phòng

Tên	Quản lý trạng thái đặt phòng
Tác nhân	Quản trị viên
Mô tả mục tiêu	Quản lý trạng thái đặt phòng
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	Quản trị viên đăng nhập thành công
Mô tả	<ol style="list-style-type: none"> 1. Tại thanh sidebar Quản trị viên chọn Quản lý đặt phòng 2. Hệ thống hiển thị danh sách phòng đã đặt 3. Quản trị viên nhấn button Giao phòng ở cột Hành động 4. Hệ thống enable nút Giao phòng và kích hoạt button Trả phòng
Hậu điều kiện	Không
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 15: Bảng scenario quản lý trạng thái đặt phòng

- Scenario Xác nhận trả phòng

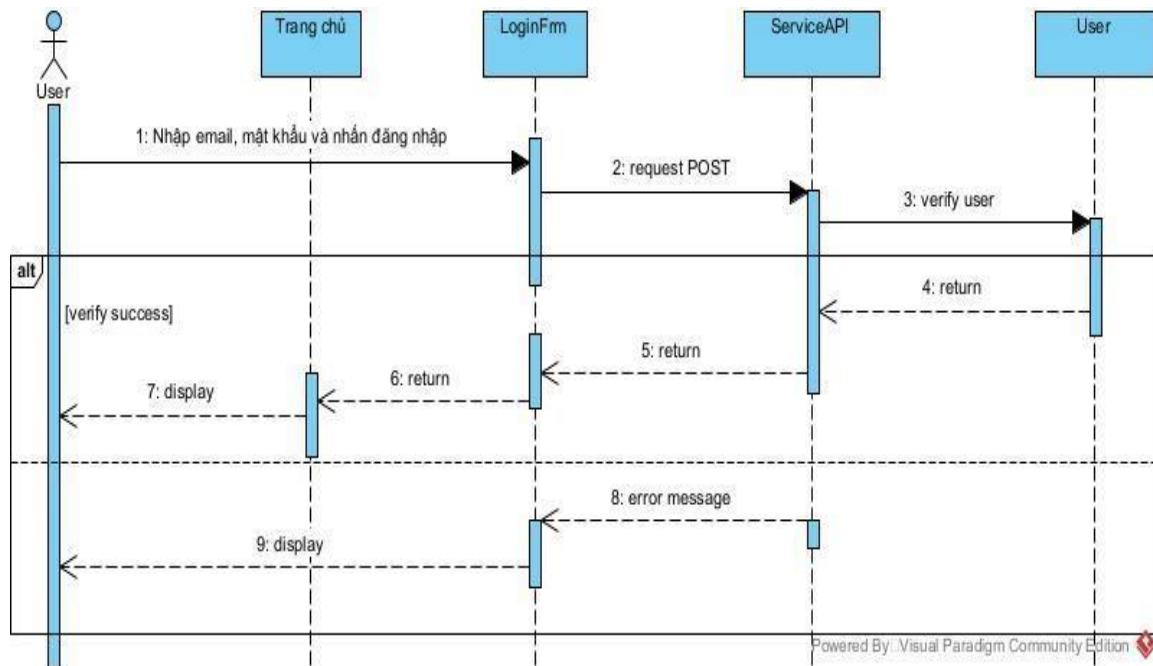
Tên	Quản lý trạng thái trả phòng
Tác nhân	Quản trị viên
Mô tả mục tiêu	Quản lý trạng thái trả phòng
Yêu cầu liên quan	Usecase đăng nhập
Tiền điều kiện	<ul style="list-style-type: none"> - Quản trị viên đăng nhập thành công - Quản trị viên xác nhận giao phòng

Mô tả	<ol style="list-style-type: none"> 1. Tại thanh sidebar Quản trị viên chọn Quản lý đặt phòng 2. Hệ thống hiển thị danh sách phòng đã đặt 3. Quản trị viên nhấn button Trả phòng ở cột Hành động 4. Hệ thống enable nút Trả phòng
Hậu điều kiện	Không
Biến thể	Không
Ngoại lệ	Không

Bảng 3.4.3 16: Bảng scenario xác nhận trả phòng phía admin

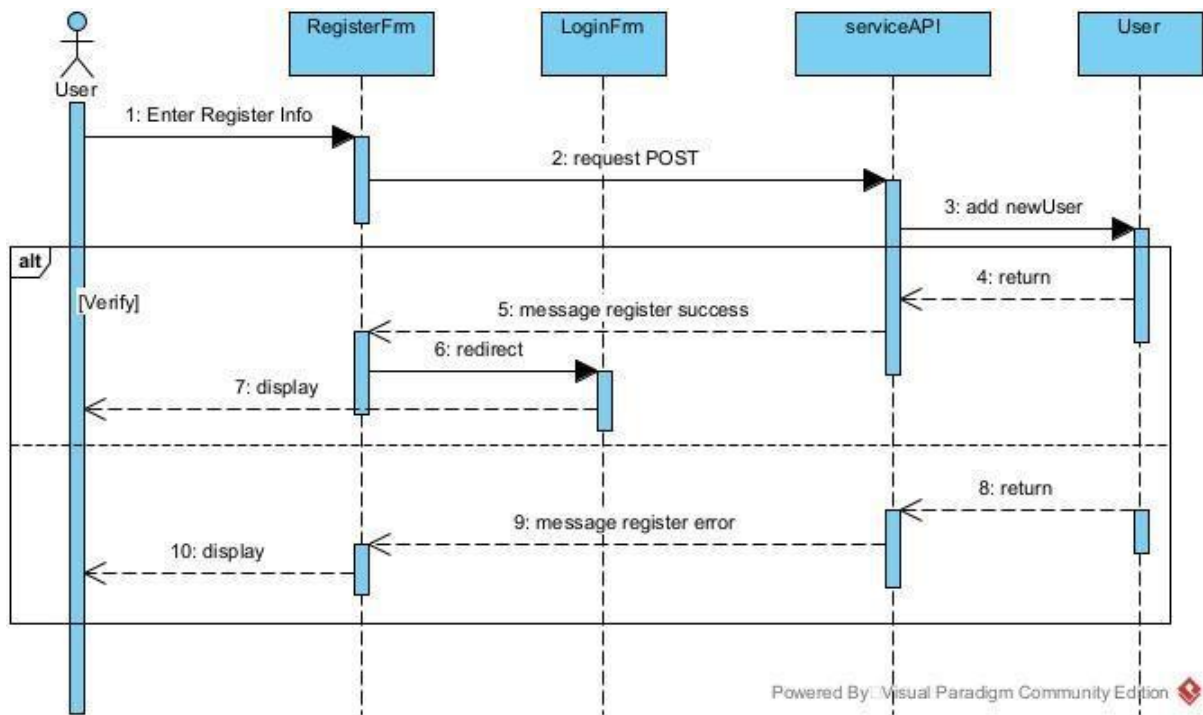
3.4.4 Biểu đồ tuần tự

- Biểu đồ tuần tự luồng đăng nhập



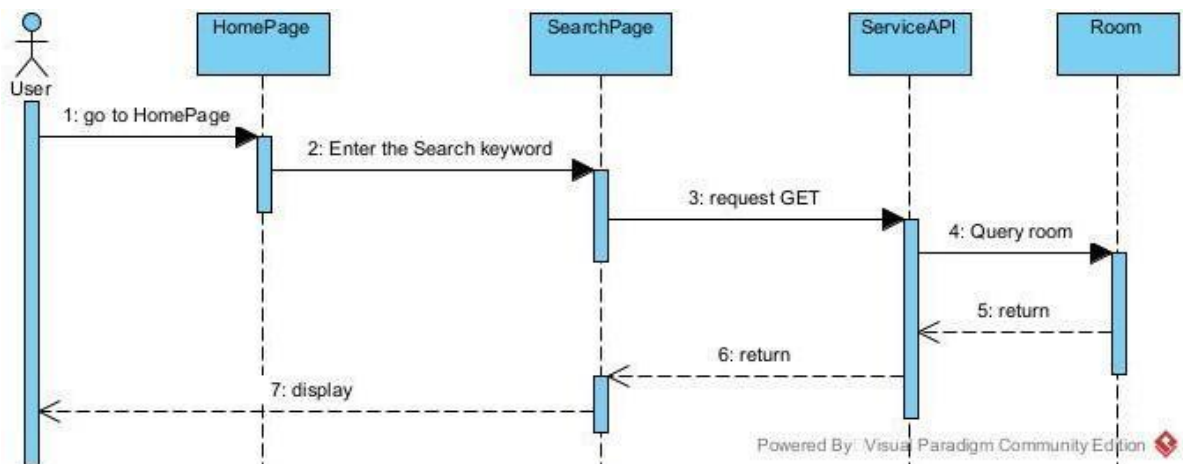
Hình ảnh 3.4.4. 1: Biểu đồ tuần tự chức năng đăng nhập

- Biểu đồ tuần tự luồng đăng ký



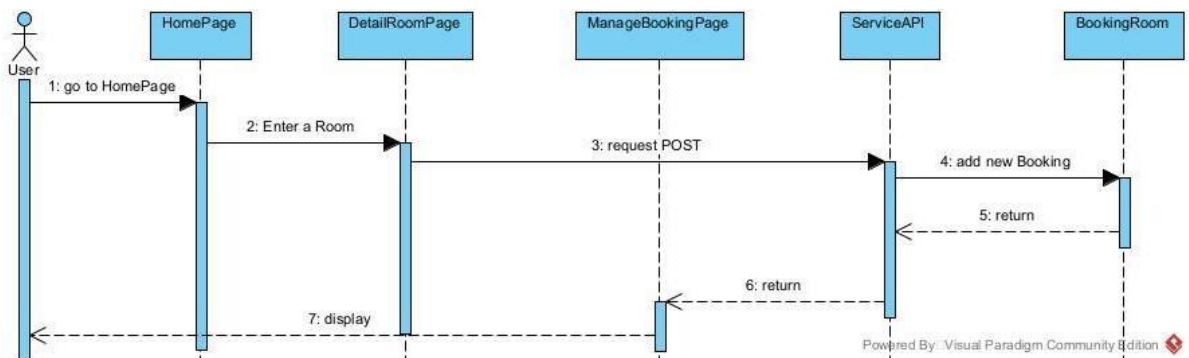
Hình ảnh 3.4.4. 2: Biểu đồ tuần tự chức năng đăng ký

- Biểu đồ tuần tự luồng tìm kiếm phòng



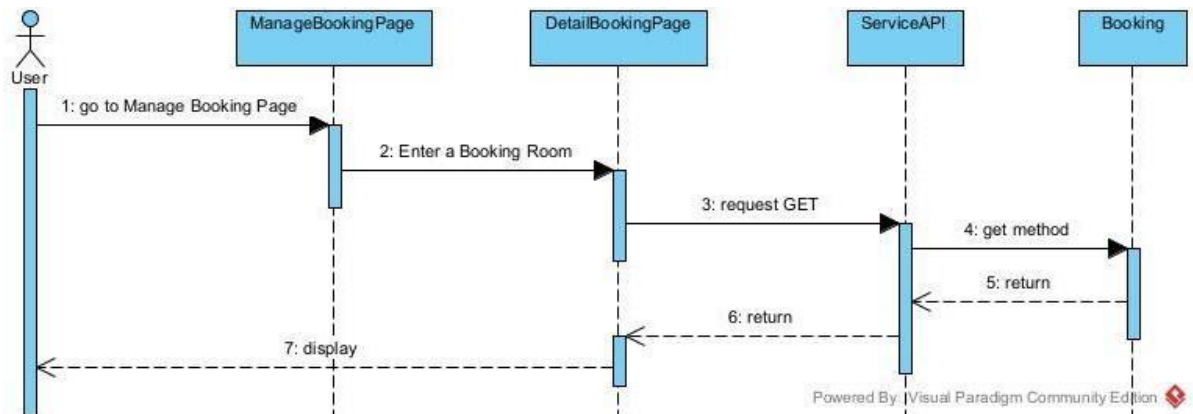
Hình ảnh 3.4.4. 3: Biểu đồ tuần tự luồng tìm kiếm phòng

- Biểu đồ tuần tự luồng đặt phòng



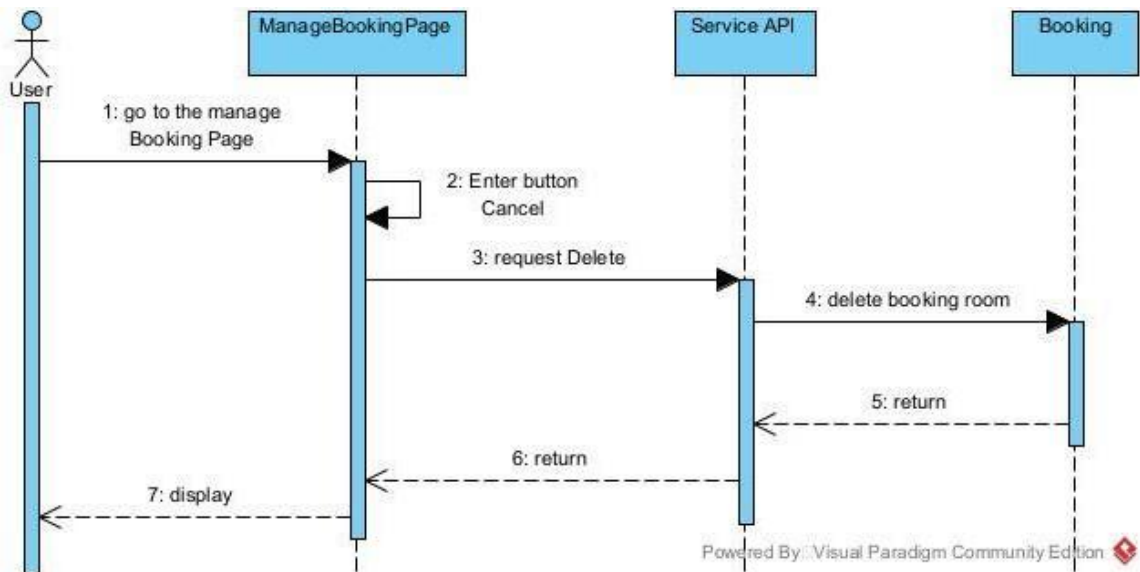
Hình ảnh 3.4.4. 4: Biểu đồ tuần tự luồng đặt phòng

- Biểu đồ tuần tự luồng xem chi tiết đặt phòng



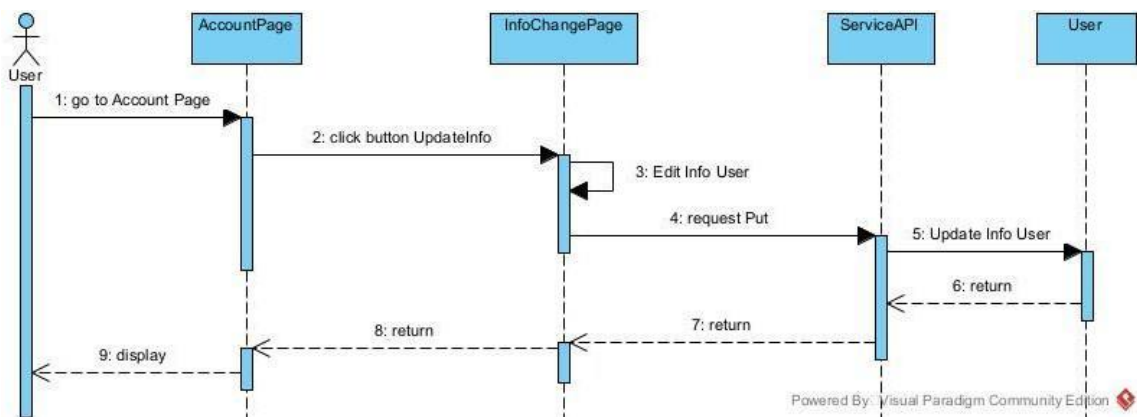
Hình ảnh 3.4.4. 5: Biểu đồ tuần tự luồng xem chi tiết đặt phòng

- Biểu đồ tuần tự luồng hủy đặt phòng phía người dùng



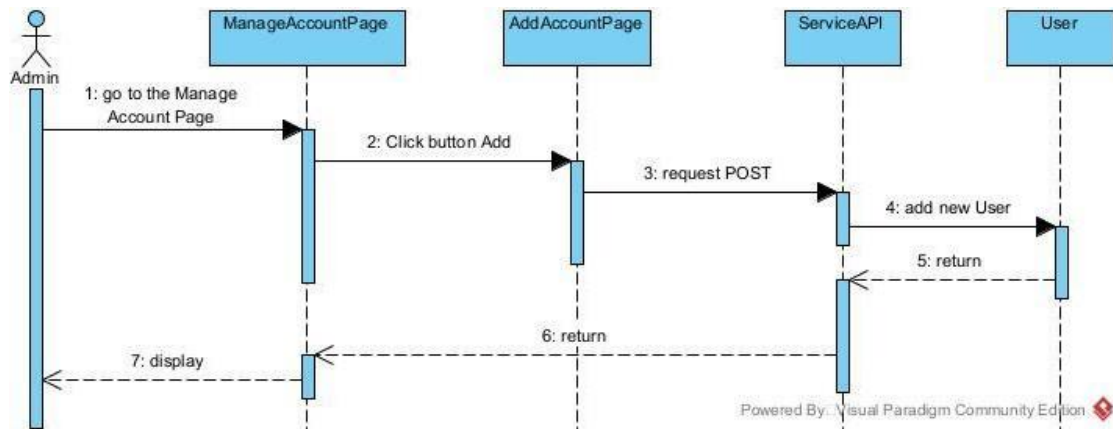
Hình ảnh 3.4.4. 6: Biểu đồ tuần tự luồng hủy đặt phòng phía người dùng

- Biểu đồ tuần tự luồng cập nhật thông tin người dùng



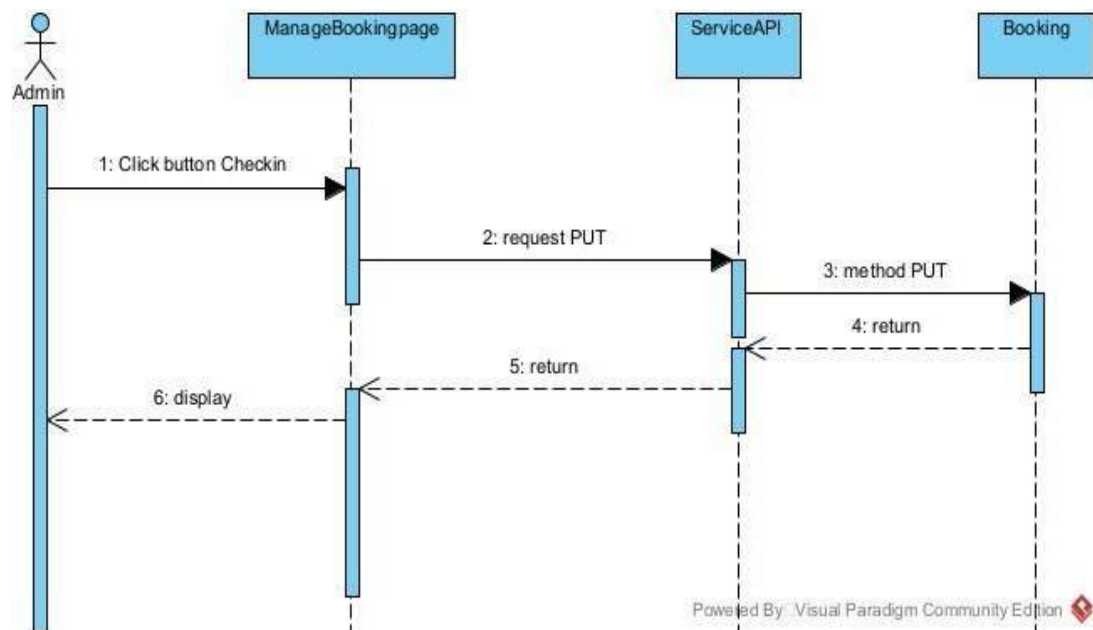
Hình ảnh 3.4.4. 7: Biểu đồ tuần tự luồng cập nhật thông tin người dùng

- Biểu đồ tuần tự luồng thêm tài khoản người dùng phía admin



Hình ảnh 3.4.4. 8: Biểu đồ tuần tự luồng thêm tài khoản người dùng phía admin

- Biểu đồ tuần tự luồng quản lý đặt phòng phía admin



Hình ảnh 3.4.4. 9: Biểu đồ tuần tự quản lý đặt phòng phía admin

3.5 Kết luận chương

Kết thúc chương 3 với việc phân tích và thiết kế hệ thống website đặt phòng khách sạn đã cung cấp một cái nhìn chi tiết về cách hệ thống sẽ hoạt động, từ việc tìm hiểu yêu cầu đến việc thiết kế các chức năng cụ thể để đáp ứng nhu cầu của người dùng.

Trong quá trình này, các chức năng quan trọng như đăng nhập, đăng ký, và đặt phòng khách sạn đã được xác định rõ ràng, cùng với việc mô tả chi tiết về cách người dùng sẽ tương tác với hệ thống. Đối tượng sử dụng hệ thống cũng được đề cập kỹ lưỡng để đảm bảo rằng mọi người dùng có thể được xem xét dựa trên nhu cầu cụ thể của họ.

Tiếp theo, chương 4 sẽ tập trung vào việc triển khai và cài đặt hệ thống, chuyển đổi từ thiết kế đã hoàn chỉnh sang hiện thực. Sẽ áp dụng các công nghệ và ngôn ngữ lập trình đã được xác định trước đó để xây dựng hệ thống. Mục tiêu chính là đưa thiết kế thành mã nguồn, triển khai tính năng và kiểm tra hệ thống để đảm bảo hoạt động ổn định và đáp ứng đúng yêu cầu đã được xác định.

CHƯƠNG IV: CÀI ĐẶT VÀ TRIỂN KHAI HỆ THỐNG

4.1. Chuẩn bị môi trường

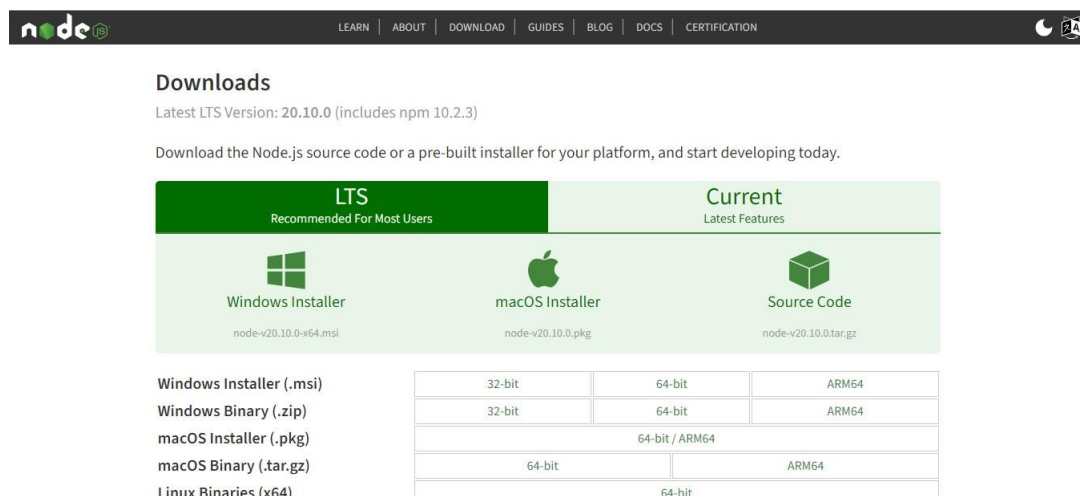
4.1.1 NodeJs

NodeJs được xây dựng bằng Javascript Engine V8 của Google Chrome, đây là nền tảng mã nguồn mở được rất nhiều lập trình viên sử dụng trong việc xây dựng và phát triển các ứng dụng website.

Việc cung cấp khả năng mở rộng ứng dụng và tập trung vào việc phát triển ứng dụng thiên về dữ liệu của NodeJs khiến cho nhiều ứng dụng ngày nay sử dụng thư viện này. Ý tưởng chính của thư viện là hướng sự ra vào của dữ liệu thông qua các tác vụ thời gian thực theo một cách nhanh nhất có thể, vì vậy điều này giảm thiểu tối đa thời gian truy cập, tăng khả năng xử lý các tác vụ dữ liệu.

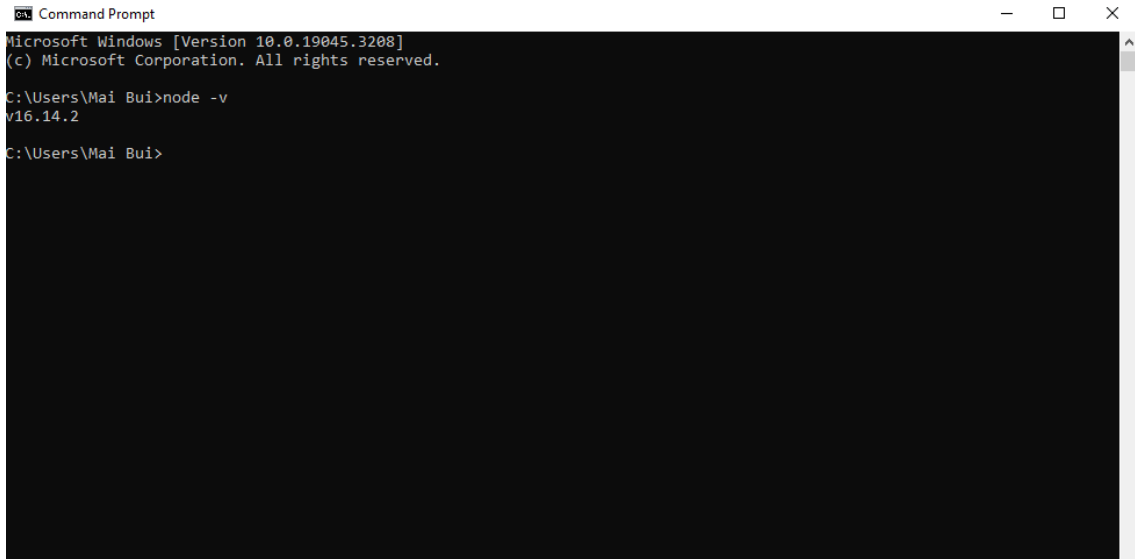
Cách cài đặt NodeJs:

- Tải về và cài đặt NodeJs tại trang chủ: <https://nodejs.org/en/download/>



Hình ảnh 4.1.1 1: Website cài đặt NodeJS

- Kiểm tra phiên bản Node đã cài tại Command Prompt với lệnh “node -v”



Hình ảnh 4.1.1 2: Hình ảnh kiểm tra version Node

4.1.2 Express

Cài đặt Express Framework:

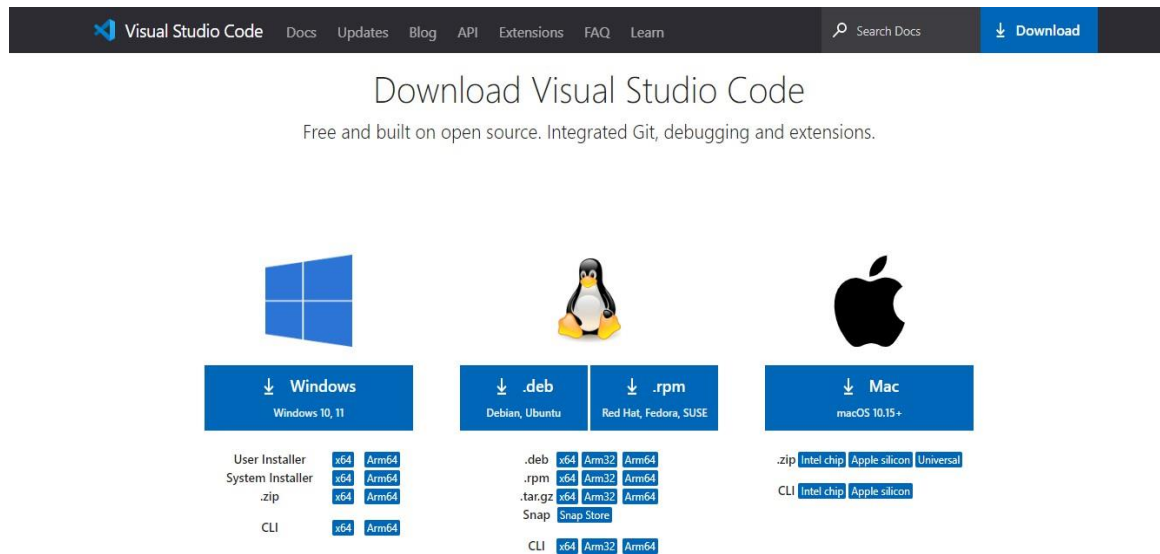
- Tạo một thư mục để lưu trữ
 - Sử dụng lệnh `npm init` để tạo một file `package.json`
- Cài đặt framework express trong thư mục vừa tạo trên bằng cách sử dụng lệnh: `npm install express --save`. Lệnh này lưu phần cài đặt trong thư mục `node_modules` và tạo thư mục `express` bên trong thư mục đó

4.1.3 Cài đặt Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở gọn nhẹ nhưng có khả năng vận hành mạnh mẽ trên 3 nền tảng là Windows, Linux và macOS

được phát triển bởi Microsoft. Nó hỗ trợ cho JavaScript, Node.js và TypeScript, cũng như cung cấp một hệ sinh thái mở rộng vô cùng phong phú cho nhiều ngôn ngữ lập trình khác.

Cài đặt Visual Studio Code: tải về và cài đặt Visual Studio Code tại trang web: <https://code.visualstudio.com/download>.



Hình ảnh 4.1.3 1: Trang website cài đặt Visual Studio Code

- Lựa chọn phiên bản Visual Studio Code phù hợp với máy tính và cài đặt

4.2 Xây dựng Frontend

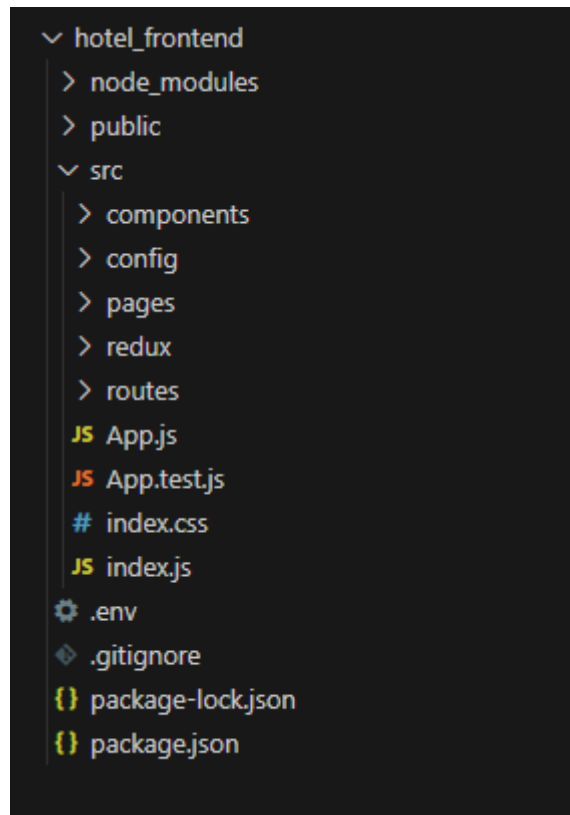
4.2.1 Cấu trúc frontend

- Khởi tạo hệ thống: Mở Command Prompt (Windows) hoặc Terminal trong Visual Studio Code và chạy lệnh sau: `npx create -react-app tên project`
- Sau khi chạy lệnh trên, Create React App sẽ tạo ra một dự án React mới với cấu trúc thư mục và tệp tin chuẩn. Có thể mở dự án bằng cách sử dụng Terminal hoặc Command Prompt và chạy các lệnh sau:

```
C:\Users\Mai Bui\Documents\Hotel\hotel_frontend>npm start
```

Hình ảnh 4.2.1 1: Câu lệnh chạy dự án ReactJS

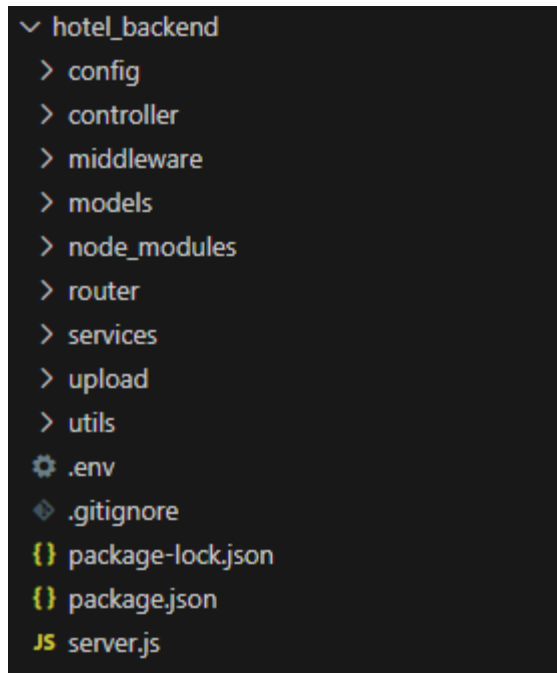
- Cấu trúc project:



Hình ảnh 4.2.1 2: Cấu trúc mã nguồn dự án

- Thư mục `node_modules`: chứa tất cả các module Node.js cần thiết để chạy ứng dụng React
- Thư mục `public`: chứa các tệp tin tĩnh như hình ảnh, file `index.html`
- Thư mục `src`: chứa mã nguồn chính của hệ thống, Tại đây được chia thành các thư mục nhỏ hơn để quản lý từng thành phần khác nhau:
 - + Component: chứa các khối UI có thể được tái sử dụng trong hệ thống
 - + Config: Dùng để cấu hình thư viện axios
 - + Page: chứa các trang giao diện của website
 - + Redux: chứa các thành phần quản lý trạng thái của hệ thống
 - + Routes: quản lý đường dẫn của trang web

4.2.2 Cấu trúc backend



Hình ảnh 4.2.2 1: Cấu trúc thư mục phía backend

- Thư mục config: sử dụng để chứa các tệp tin cấu hình liên quan đến cài đặt và cấu hình của ứng dụng
- Thư mục controller: dùng để tổ chức và quản lý logic xử lý yêu cầu từ client một cách có tổ chức, giúp cho việc phát triển và bảo trì ứng dụng trở nên dễ dàng hơn
- Thư mục middleware: Chứa các hàm middleware được sử dụng để xử lý các yêu cầu trước khi chúng đến tay các route hoặc controller.
- Thư mục model: Chứa các định nghĩa của các đối tượng (models) đại diện cho dữ liệu trong cơ sở dữ liệu
- Thư mục node_modules: chứa các module và dependencies (phụ thuộc) mà dự án cần để chạy.
- Thư mục router: chứa các file hoặc module để xác định và quản lý các routes (đường dẫn) của ứng dụng.
- Thư mục service: lưu trữ tệp tin dùng để kết nối với database
- Thư mục upload: dùng để lưu các file tĩnh về ảnh
- File server.js: chứa mã để khởi tạo và cấu hình server, kết nối các middleware, và định nghĩa các endpoint hoặc route chính cho ứng dụng.

- Các API phía backend:

```
require("dotenv").config();
connectDb();
app.use(cors());
app.use("/upload", express.static("upload"));
app.use("/auth/admin", userRoute);
app.use("/api/auth", authRoute);
app.use("/api/room", roomRoute);
app.use("/search", searchRoute);
app.use("/api/booking", bookingRoute);
app.use("/api/order", orderRoute);
app.use("/api/contact", contactRoute);

httpServer.listen(process.env.PORT, () => {
  console.log(`Server đã khởi động trên cổng ${process.env.PORT}`);
});
```

Hình ảnh 4.2.2 2: Các API phía backend

4.3 Giao diện website

4.3.1 Các thành phần của website

a. Header:

- Đây là phần trên cùng của trang web, nơi chứa các thông tin quan trọng và liên kết điều hướng



Hình ảnh 4.3.1. 1: Thành phần header của website

- Thành phần header của website gồm 2 phần:

+ TopHeader:

Gồm các mục: Tiêu đề chào mừng, nút đăng nhập, đăng ký nếu người dùng chưa đăng nhập, tên tài khoản nếu người dùng đã đăng nhập, nút tìm kiếm

Code:

```

return (
  <div position="static">
    <div>
      <div
        style={{ minHeight: "3.4rem", justifyContent: "flex-end" }}>
        <div style={{ display: "block" }}>
          Chào mừng bạn đến với Meritel Hotel!
        </div>
        <div>
          <Link to={"/register"}>
            <div>Đăng ký</div>
          </Link>
          <div />
          <Link to={"/login"}>
            <div>Đăng nhập</div>
          </Link>
          <div />
        </div>
      </div>
    </div>
    <div>
      <div>
        <div>
          sx={{
            cursor: "pointer",
          }}
          onClick={handleLogout}>
            Đăng xuất
          </div>
          <Separate />
          <Link to={"/account"}>
            <div>Tài khoản</div>
          </Link>
        </div>
      </div>
    </div>
  </div>
)

```

Hình ảnh 4.3.1. 2: Mã code thành phần Top Header

+ MainHeader: Gồm các mục: Logo, Trang chủ, Về chúng tôi, Phòng, Liên hệ, nút Đặt phòng

```

return (
  <div position="static">
    <div>
      <Toolbar
        style={{
          paddingLeft: { xs: "10px", md: "10px", lg: "24px" },
          minHeight: { md: "60px", md2: "67px", lg: "90px" },
          justifyContent: { xs: "space-between", sm: "space-between", md: "unset" },
          paddingRight: { xs: "0", sm: "14px", md: "14px", md2: "14px" },
        }}>
        <Link to={"/"}>
          <Logo src="/images/logo-meritel.jpg" />
        </Link>
        <Link to={"/"}>
          <div>
            <div>
              style={{ color: getColor("/"), display: { xs: "none", sm: "none", md: "block" },
                padding: { md: "7px 8px", lg: "7px 15px" }, }}
              disableRipple>Trang chủ
            </div>
          </div>
        </Link>
        <Link to={"/about"}>
          <div>
            <div>
              style={{ color: getColor("/about") }}
              sx={{
                display: { xs: "none", sm: "none", md: "block" },
                padding: { md: "7px 8px", lg: "7px 15px" },
              }}
              disableRipple>Về chúng tôi
            </div>
          </div>
        </Link>
      </Toolbar>
    </div>
  </div>
)

```

Hình ảnh 4.3.1. 3: Mã code thành phần main header của website

b. Thành phần card phòng

Hiển thị danh sách các phòng ra ngoài giao diện


```
return (
  <>
    {rooms &&
      rooms.map((item, index) => {
        return (
          <Grid key={index} item lg={3} md={6} sm={12} xs={12}>
            <Card>
              <CardMedia
                onClick={() => {
                  navigate(`/room/${item._id}`);
                  window.scrollTo(0, 0);
                }}
                sx={{ height: "165px", cursor: "pointer" }}
                image={` ${process.env.REACT_APP_HOST_URL}${item.cover}`}
              />
              <CardContent sx={{ textAlign: "center" }}>
                <Typography
                  fontSize="1.4rem"
                  style={{
                    width: "100%",
                    textTransform: "uppercase",
                    textAlign: "center",
                    fontWeight: "600",
                    cursor: "pointer",
                    "&:hover": {
                      color: "var(--primary-color)",
                    },
                  }}
                />
                <Link href={`/room/${item._id}`}>
                  {item.name}
                </Link>
              </CardContent>
            </Card>
          </Grid>
        )
      })
    }
  </>
)
```

Hình ảnh 4.3.1. 4: mã code thành phần list room trong website

c. Thành phần footer

Footer của trang web là phần nằm ở cuối trang, chứa các thông tin hữu ích và liên kết hỗ trợ người dùng sau khi khách hàng đã xem xét nội dung chính của trang

Code:

```
return (
  <div>
    <div>
      <div style={{margin: "30px"}}>
        <div>
          <Link to="/">
            <Logo src="/images/logo-footer.jpg" />
          </Link>
          <div>CÔNG TY TNHH KHÁCH SẠN MERITEL HOTEL</div>
          <div>
            style={{fontSize: "1.8rem",marginTop: "16px",padding: "0 24px",textAlign: "justify",color: "#000000"}}
          </div>
          <div>
            Một điểm đến không thể nào quên!
          </div>
          <div style={{
            fontSize: "1.5rem",marginTop: "16px",padding: "0 24px",textAlign: "justify",color: "#000000"
          }}>
            Nằm ở trung tâm thành phố và nhiều thành phố khác. Chúng tôi cung cấp chỗ nghỉ thanh lịch đây phong cách với truy cập Wifi miễn phí trong các khu vực chung. Khách sạn có lễ tân 24 giờ, hồ bơi trong nhà, trung tâm thể dục và bãi đỗ xe miễn phí trong khuôn viên.
          </div>
        </div>
        <div>
          <div style={{
            textTransform: "uppercase",fontSize: "1.8rem",marginTop: "16px",padding: "0 24px",color: "#000000"
          }}>
            Liên hệ với chúng tôi
          </div>
        </div>
      </div>
    </div>
  </div>
)
```

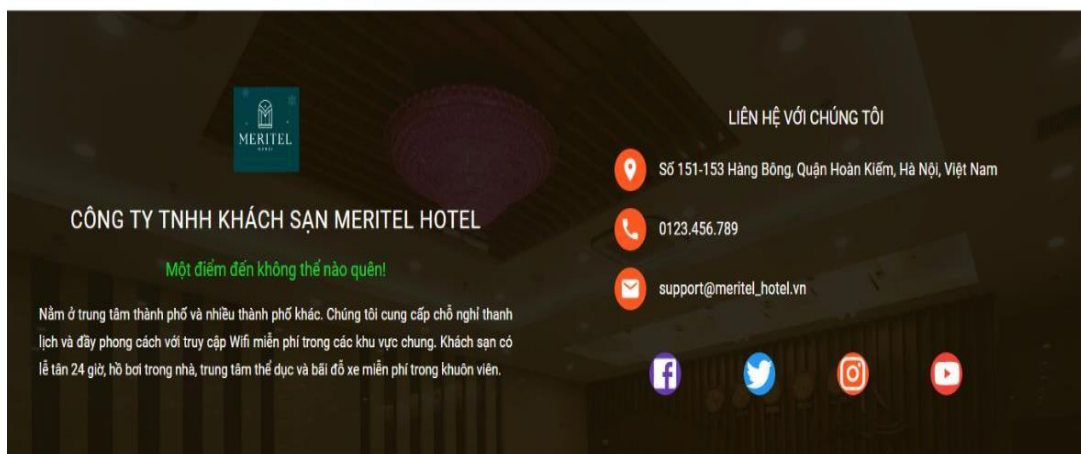
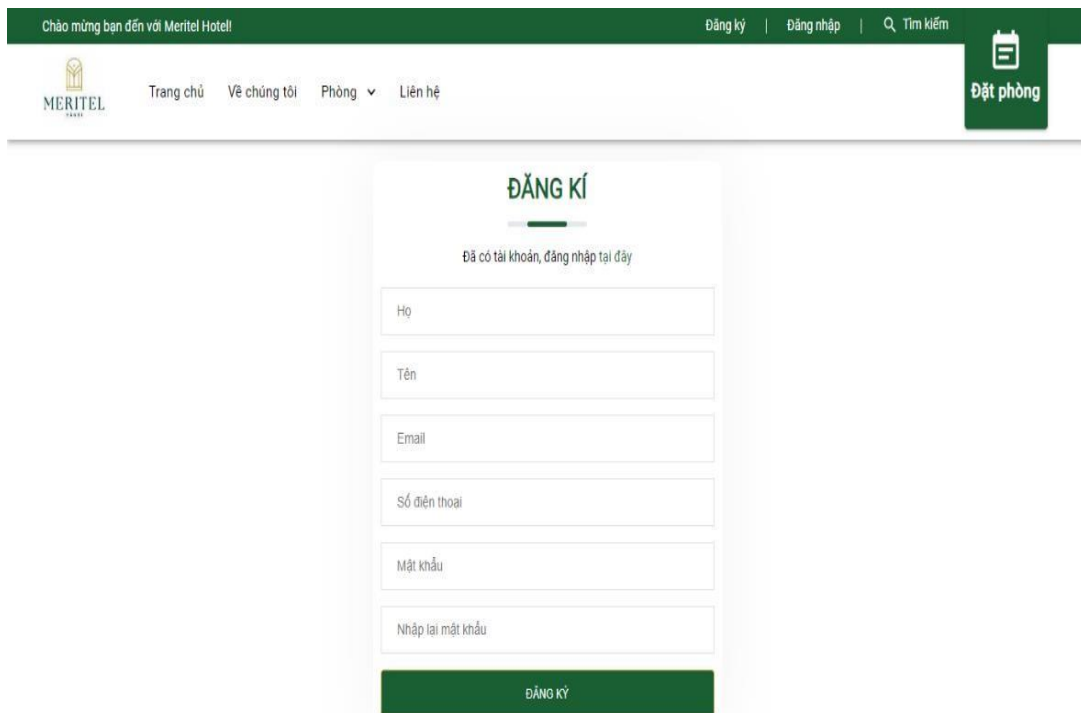
Hình ảnh 4.3.1. 5: Mã code thành phần header của website

4.3.2 Các trang của website

- Giao diện trang đăng ký:

Người dùng truy cập vào website nhấn vào nút đăng ký ở trên thanh header sẽ hiển thị giao diện trang đăng ký

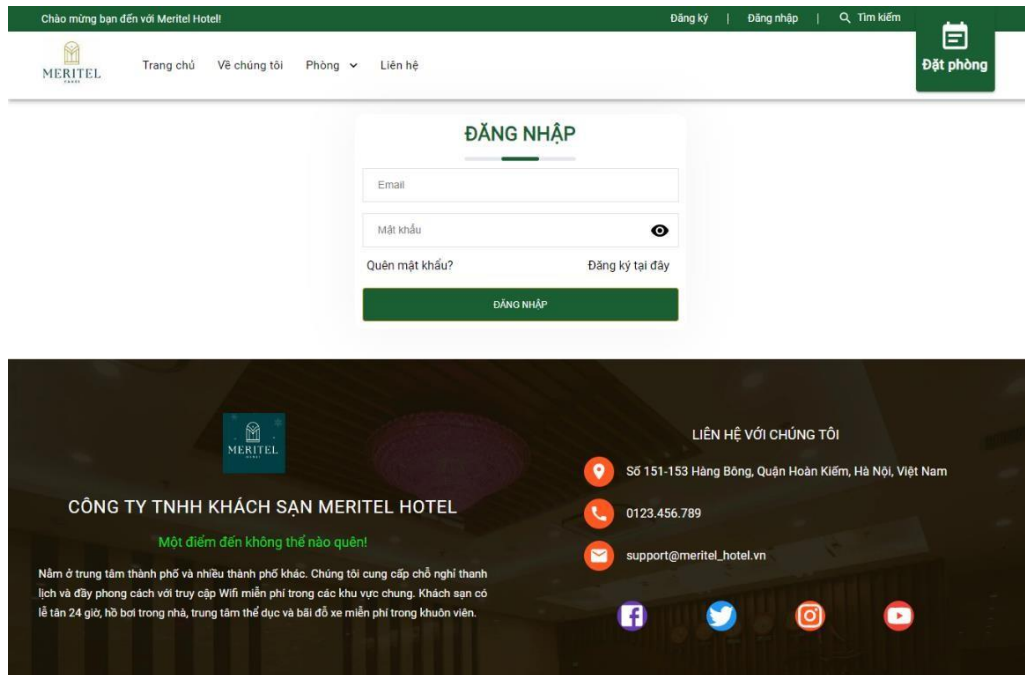
Ở giao diện này gồm những thông tin để người dùng nhập vào: họ, tên, email, số điện thoại, mật khẩu. Người dùng nhập đầy đủ thông tin trên mới đăng ký thành công tài khoản



Hình ảnh 4.3.2. 1: Giao diện website trang đăng ký tài khoản

- Giao diện trang đăng nhập

Sau khi đã tạo được tài khoản, người dùng chỉ cần nhập thông tin email, mật khẩu vào giao diện đăng nhập sẽ thành công truy nhập vào hệ thống



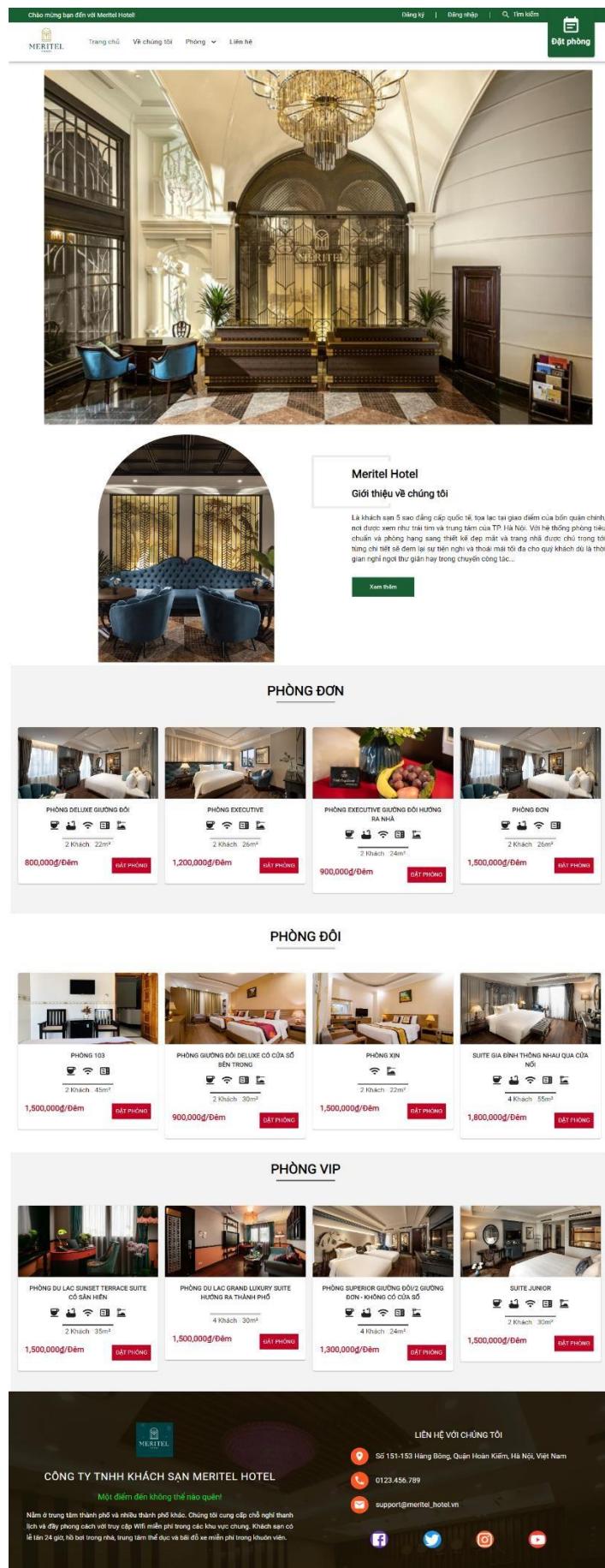
Hình ảnh 4.3.2. 2: Giao diện website trang đăng nhập tài khoản

- Giao diện trang chủ

Người dùng có thể truy cập vào trang chủ của website khi chưa đăng nhập và đăng nhập thành công vào hệ thống

Giao diện trang chủ gồm các thành phần:

- Header: logo, trang chủ, về chúng tôi, phòng, liên hệ
- Bên dưới header là banner của khách sạn
- Phần mô tả ngắn về khách sạn và hình ảnh
- Danh sách các phòng được hiển thị theo loại phòng
- Cuối cùng là footer bao gồm thông tin về khách sạn, thông tin liên hệ của khách sạn, các đường dẫn liên kết mạng xã hội của khách sạn





Hình ảnh 4.3.2. 3: Giao diện trang chủ của website

- **Giao diện trang giới thiệu**
Trang giới thiệu bao gồm những thông tin mô tả, hình ảnh về khách sạn

Chào mừng bạn đến với Meritel Hotel!


Đăng ký | Đăng nhập | Q. Tìm kiếm

 Trang chủ Về chúng tôi Phòng Liên hệ

 Đặt phòng



VỀ CHÚNG TÔI

Meritel Hanoi có hồ bơi ngoài trời, trung tâm thể dục, sân hiên và quầy bar ở Hà Nội. Các điểm tham quan nổi tiếng gần đó có Hồ Hoàn Kiếm, Hoàng thành Huế và Văn Miếu - Quốc Tử Giám. Chỗ nghỉ cung cấp lễ tân 24/24, dịch vụ đưa đón sân bay, dịch vụ tiền sảnh và Wi-Fi miễn phí ở toàn bộ chỗ nghỉ.



Các phòng được trang bị điều hòa, TV có truyền hình cáp, tủ lạnh, ấm đun nước, vòi sen, đồ vệ sinh cá nhân miễn phí và bàn làm việc. Phòng khách có phòng tắm riêng, máy sấy tóc và ga trải giường.


Chỗ nghỉ có phục vụ bữa sáng mỗi sáng, gồm các lựa chọn thực đơn buffet, kiểu Mỹ và kiểu Á. Tại chỗ nghỉ, khách sẽ tìm thấy nhà hàng phục vụ ẩm thực Anh, Việt và địa phương. Bên cạnh đó, họ có thể yêu cầu món chay, không chứa sữa và món thuần chay.



Cùng với đội ngũ nhân viên được tuyển chọn và đào tạo chuyên nghiệp, chu đáo và thân thiện, **Meritel Hotel** hứa hẹn sẽ mang đến cho Quý khách sự thoải mái và hài lòng nhất.

Đến với **Meritel Hotel** là đến với sự tinh tế nhất về chất lượng, dịch vụ và sự thân thiện như chính ngôi nhà của bạn.

HÃY ĐẾN MERITEL HOTEL ĐỂ TRẢI NGHIỆM SỰ KHÁC BIỆT!





CÔNG TY TNHH KHÁCH SẠN MERITEL HOTEL


Một điểm đến không thể nào quên!





Nằm ở trung tâm thành phố và nhiều thành phố khác. Chúng tôi cung cấp chỗ nghỉ thanh lịch và đầy phong cách với truy cập Wifi miễn phí trong các khu vực chung. Khách sạn có lễ tân 24 giờ, hồ bơi trong nhà, trung tâm thể dục và bãi đỗ xe miễn phí trong khuôn viên.

LIÊN HỆ VỚI CHÚNG TÔI

 Số 151-153 Hàng Bông, Quận Hoàn Kiếm, Hà Nội, Việt Nam

 0123.456.789

 support@meritel_hotel.vn

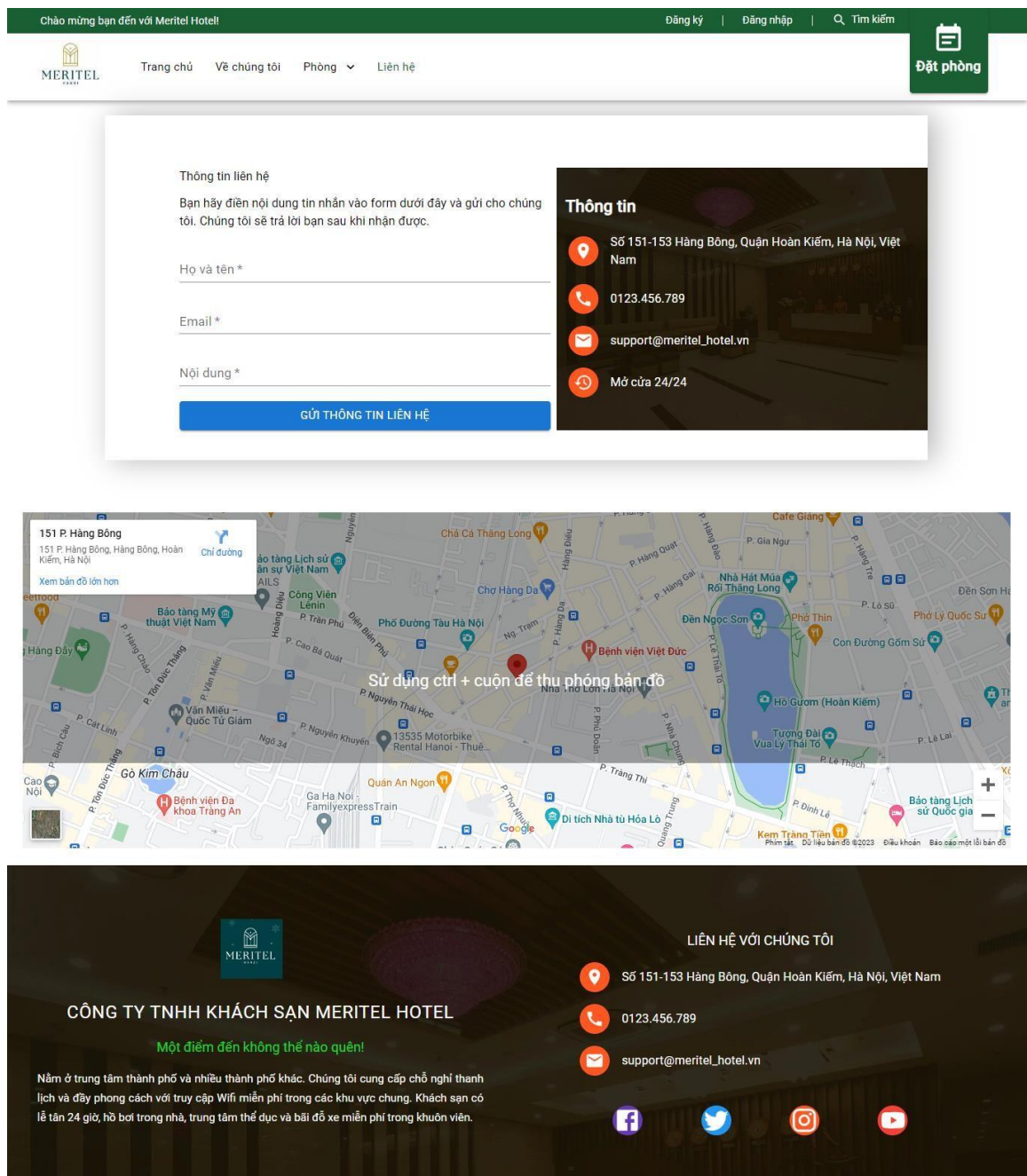


Hình ảnh 4.3.2. 4: Giao diện trang giới thiệu khách sạn

- Giao diện trang liên hệ

Trang liên hệ giúp khách hàng có thể gửi thông tin đến quản trị viên của hệ thống, hoặc người dùng có thể xem chi tiết thông tin liên hệ về khách sạn bao gồm: địa chỉ, số điện thoại, email hỗ trợ

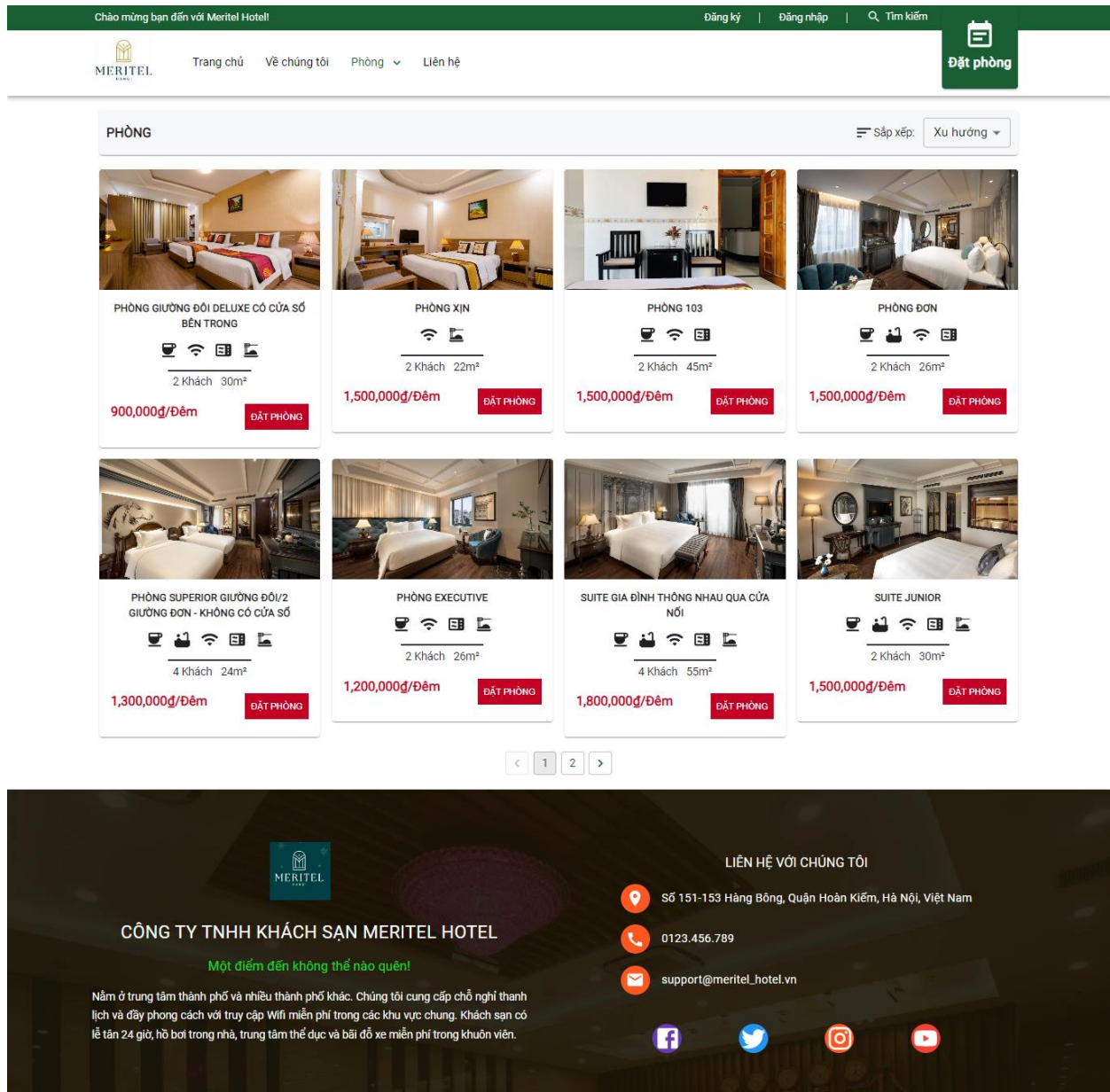
Bên dưới là phần bản đồ giúp khách hàng có thể dễ dàng tìm kiếm địa chỉ của khách sạn



Hình ảnh 4.3.2. 5: Giao diện trang liên hệ

- Giao diện trang danh sách phòng

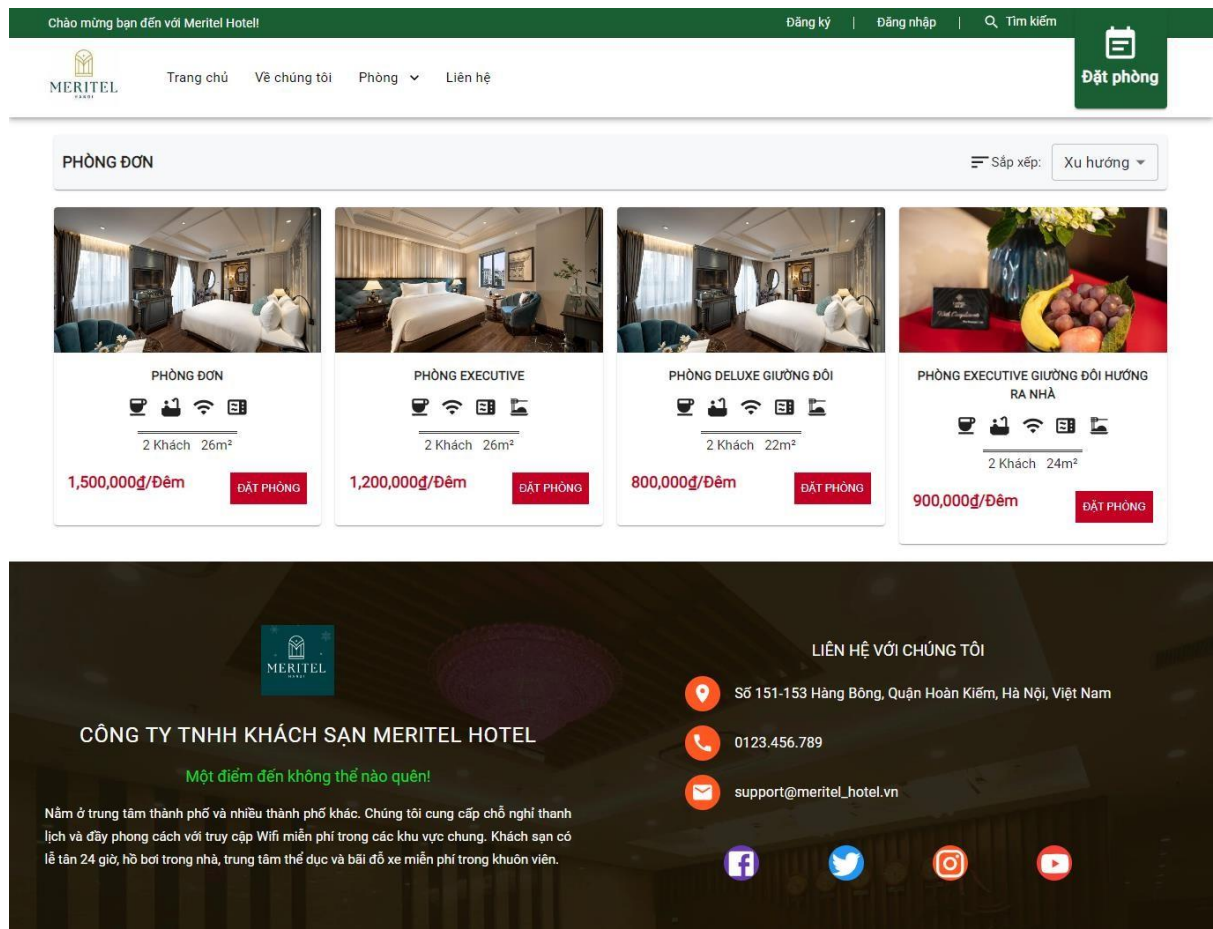
Giao diện hiển thị toàn bộ danh sách phòng có trong hệ thống, trong giao diện sẽ hiển thị danh sách các phòng gồm 2 hàng và 4 cột, khi số lượng phòng nhiều hơn sẽ hiển thị số trang để khách hàng chọn



Hình ảnh 4.3.2. 6: Giao diện trang danh sách phòng

- Giao diện trang danh sách phòng theo loại phòng

Giao diện hiển thị danh sách phòng theo loại phòng, tại đây người dùng có thể xem thông tin về phòng: hình ảnh, các dịch vụ, số lượng người ở, diện tích phòng, giá tiền phòng trong 1 ngày, nút đặt phòng ở ngay bên ngoài để người dùng có thể dễ dàng đặt phòng



Hình ảnh 4.3.2. 7: Giao diện trang danh sách phòng đơn của website

- **Giao diện trang chi tiết phòng**

Giao diện được hiển thị khi nhấn vào 1 phòng ở trang chủ hoặc trang danh sách phòng. Tại đây người dùng có thể xem thông tin chi tiết hơn về phòng: hiển thị nhiều ảnh hơn về phòng, mô tả thông tin về phòng, phân nhập thông tin để đặt phòng, phân gợi ý các phòng tương tự.

Chào mừng bạn đến với Meritel Hotel!
Đăng ký | Đăng nhập | Tìm kiếm

MERITEL
Trang chủ | Về chúng tôi | Phòng | Liên hệ

Đặt phòng

Phòng Du LAC Sunset Terrace Suite Có Sân Hiên

2 Người lớn

1 Trẻ em

Phòng 35m²

Các phòng trang nhã và đầy phòng trang nghiêm của chúng tôi gợi nhớ về một thời đại đã qua. Mỗi tình năng như đường cong, thềm sang trọng, trần nhà cao, phòng tắm lát đá cẩm thạch, thiết bị làm sạch và nhiều không gian đều được bố trí một cách chu đáo để gọi cho riêng bạn. Tổng màu nâu phong phú và gỗ sồi tự nhiên tạo nên những khu bảo tồn yên tĩnh và yên tĩnh, được tôn lên một cách tuyệt vời bởi đồ nội thất trang nhã.

Dịch vụ phòng

Cafe Buổi Sáng

Bồn Tắm Hoa Sen

Internet Không Dây

Gọi Đồ Ăn Tại Phòng

Bếp Nấu Tại Phòng

Mô tả

Giường thoải mái 9.5 – Dựa trên 344 đánh giá
Suite này được bố trí 1 phòng khách, 1 phòng ngủ riêng biệt và 1 phòng tắm với buồng tắm đứng cùng đồ vệ sinh cá nhân miễn phí. Suite có máy điều hòa, khu vực ghế ngồi với TV màn hình phẳng, tủ quần áo, minibar, máy pha trà/cà phê cũng như tủ để quần áo. Căn này được trang bị 2 giường.

Danh sách phòng

202

Phòng trống

Phòng đã được thuê

Phòng bạn đã chọn

Phòng tương tự

SUITE JUNIOR

2 Khách

30m²

1,500,000đ/Đêm

ĐẶT PHÒNG

PHÒNG DU LAC GRAND LUXURY SUITE HƯỚNG RA THÀNH PHỐ

4 Khách

30m²

1,500,000đ/Đêm

ĐẶT PHÒNG

PHÒNG SUPERIOR GIƯỜNG ĐÔI/2 GIƯỜNG ĐƠN - KHÔNG CÓ CỬA SỔ

4 Khách

24m²

1,300,000đ/Đêm

ĐẶT PHÒNG

Phòng tốt nhất

Suite Junior

1,500,000đ

Phòng Du Lac Grand Luxury Suite Hướng Ra Thành Phố

1,500,000đ

Phòng Du Lac Sunset Terrace Suite Có Sân Hiên

1,500,000đ

Phòng Superior Giường Đôi/2 Giường Đơn - Không có Cửa sổ

1,300,000đ

CÔNG TY TNHH KHÁCH SẠN MERITEL HOTEL

Một điểm đến không thể nào quên!

Nằm ở trung tâm thành phố và nhiều thành phố khác. Chúng tôi cung cấp chỗ nghỉ thành lịch và đầy phong cách với truy cập Wifi miễn phí trong các khu vực chung. Khách sạn có lễ tân 24 giờ, hồ bơi trong nhà, trung tâm thể dục và bãi đỗ xe miễn phí trong khuôn viên.

LIÊN HỆ VỚI CHÚNG TÔI

Số 151-153 Hàng Bông, Quận Hoàn Kiếm, Hà Nội, Việt Nam

0123.456.789

support@meritel_hotel.vn

Facebook

Twitter

Instagram

YouTube

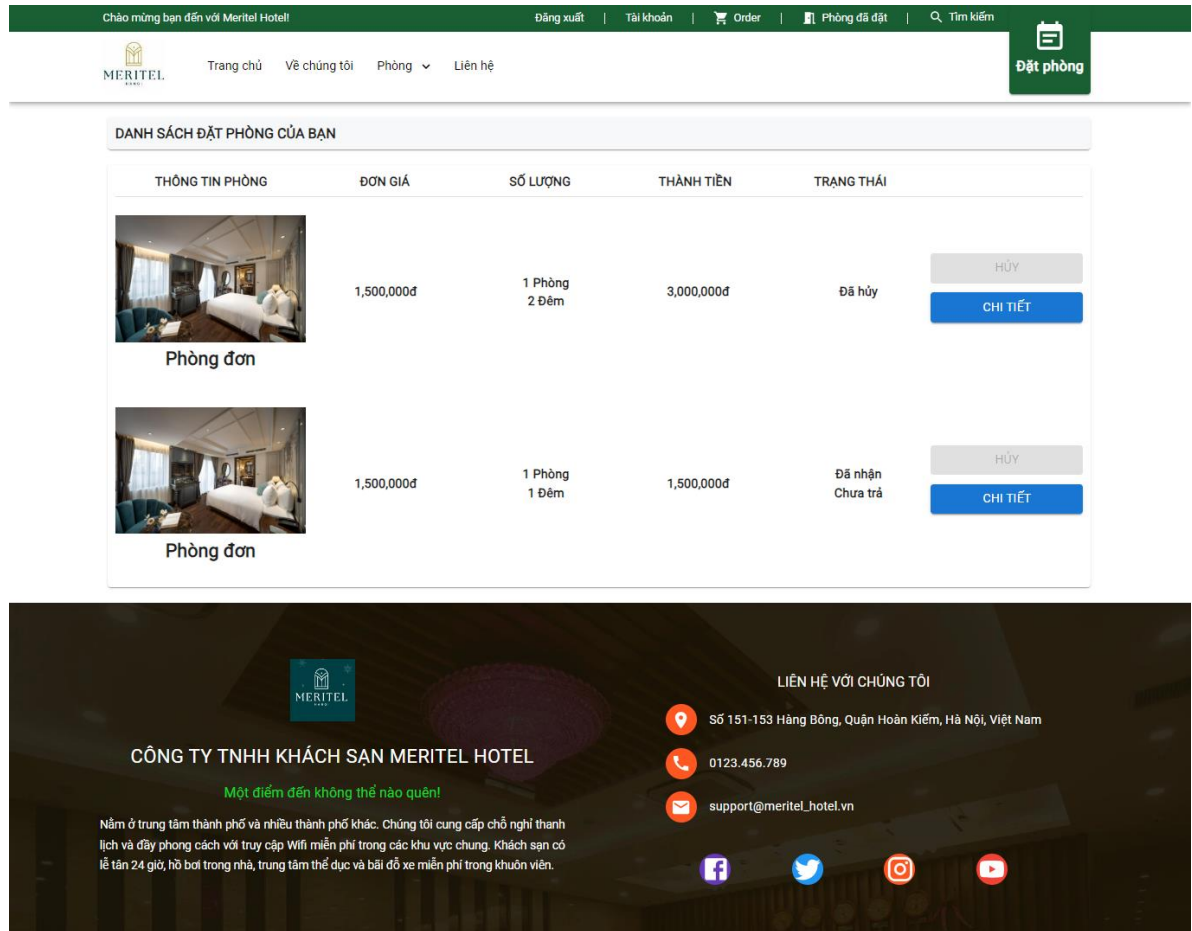
Hình ảnh 4.3.2. 8: Giao diện trang chi tiết phòng

Bùi Thị Mai – D19PTDPT

70

- Giao diện trang quản lý đặt phòng

Giao diện hiển thị danh sách các phòng đã đặt của khách hàng gồm các thông tin về phòng, đơn giá, số lượng phòng và số ngày đặt phòng, tổng số tiền thanh toán, trạng thái đặt phòng, nút xem chi tiết và nút huỷ phòng

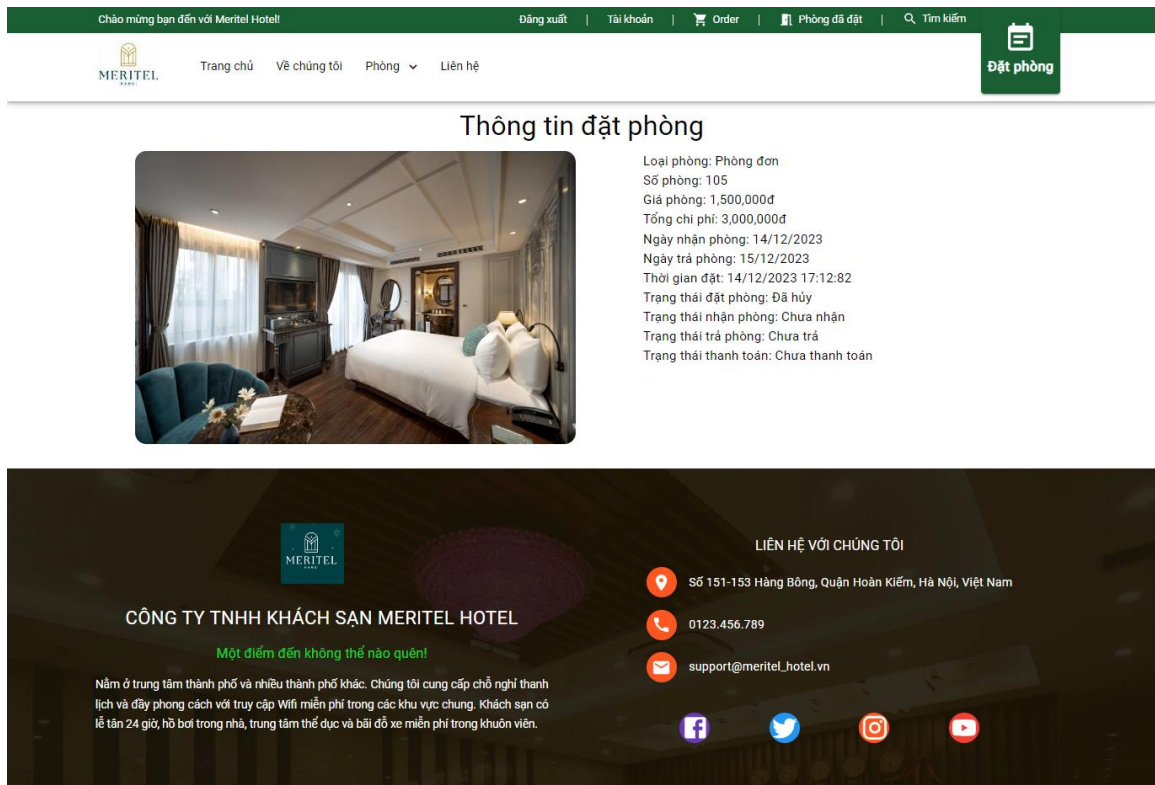


Hình ảnh 4.3.2. 9: Giao diện trang quản lý đặt phòng

- Giao diện trang thông tin đặt phòng

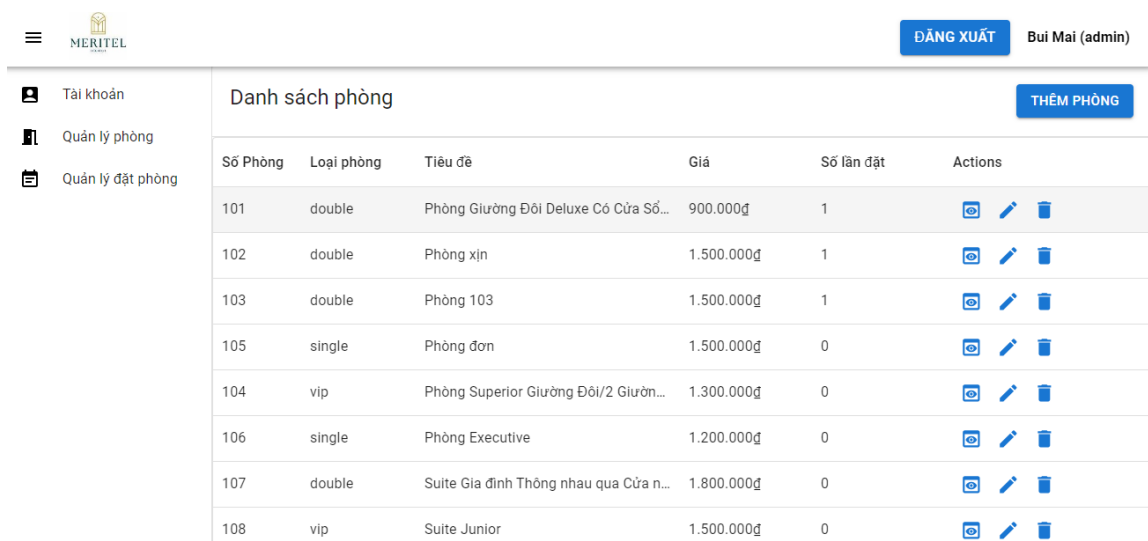
Hiển thị sau khi khách hàng đặt phòng thành công và nhấn vào nút phòng đã đặt -> Nhấn vào xem chi tiết

Tại đây hiển thị thông tin chi tiết về phòng mà khách hàng đã đặt



Hình ảnh 4.3.2. 10: Giao diện trang chi tiết thông tin đặt phòng

- Giao diện trang quản lý phòng
Để vào được trang giao diện đặt phòng người dùng phải đăng nhập bằng tài khoản admin
Tại đây người dùng có thể xem danh sách các phòng đã được đặt từ phía người dùng. Tại đây admin có thể xem, sửa, xóa phòng



Hình ảnh 4.3.2. 11: Giao diện trang quản lý phòng phía admin của website

- Giao diện trang thêm phòng
Ở giao diện Quản lý phòng quản trị viên nhấn vào nút Thêm phòng, website hiển thị giao diện trang Thêm phòng.
Tại đây quản trị viên nhập các thông tin về phòng như số phòng tiêu đề, loại phòng, lựa chọn các dịch vụ của phòng, nhập số lượng người lớn và

trẻ em, diện tích, giá phòng.

Quản trị viên có thể thêm nhiều hình ảnh về phòng bằng cách nhấn nút choose file và chọn file ảnh về phòng có trong máy tính

Cuối cùng là phần quản trị viên nhập mô tả về phòng và nhấn nút Thêm phòng để lưu phòng thành công

- Tài khoản
- Quản lý phòng
- Quản lý đặt phòng

ĐĂNG XUẤT

Bùi Mai (admin)

Thêm phòng

Số phòng (101,102,...)

Tiêu đề

Loại phòng

210

Phòng đơn

Phòng đơn

Dịch vụ phòng

☐ Cà phê buổi sáng
 ☐ Gọi đồ ăn tại phòng
 ☐ Bếp nấu tại phòng
 ☐ Bồn tắm hoa sen
 ☐ Internet không dây

Người lớn

Trẻ em

Diện tích

Giá phòng

2

1

20

5000000

Ảnh

Choose Files

2 files

1

2

Mô tả

Được thiết kế để cung cấp không gian thoải mái và tiện nghi cho du khách nghỉ ngơi và tận hưởng kỳ nghỉ của họ. Dưới đây là mô tả chi tiết về các yếu tố thường có trong một phòng khách sạn:

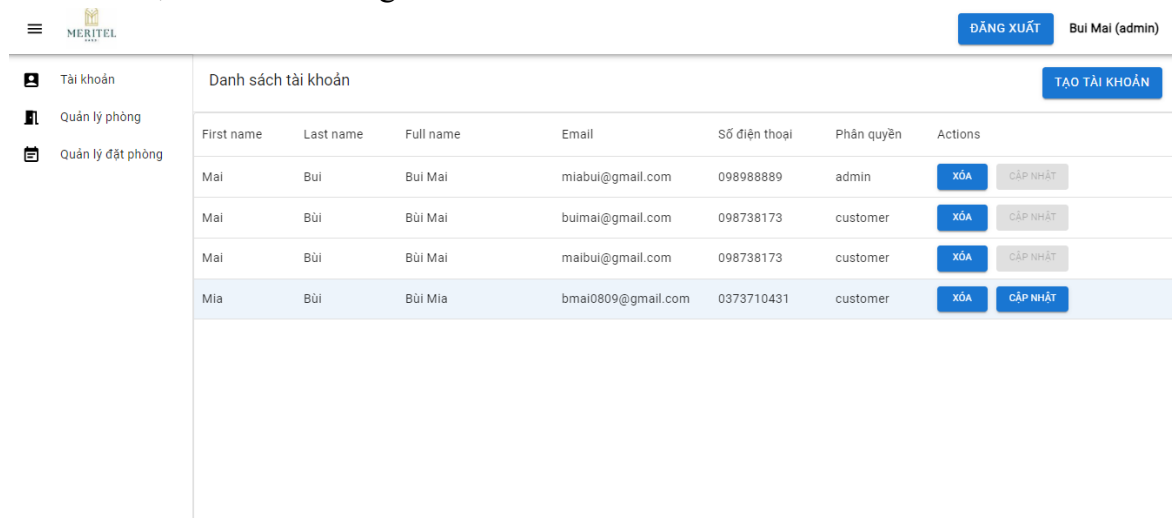
- Kích thước và bố trí: Phòng khách sạn thường có diện tích từ nhỏ đến rộng lớn tùy thuộc vào loại phòng và tiêu chuẩn của khách sạn. Bố trí thông thường bao gồm giường ngủ, bàn làm việc hoặc khu vực làm việc, phòng tắm riêng và không gian nghỉ ngơi.
- Giường ngủ: Trang bị với giường thoải mái, bộ ga trải giường chất lượng cao, gối ai phòng, chăn và đồ sạch sẽ. Một số phòng có thể có giường đôi (king-size hoặc queen-size).

THÊM PHÒNG

Hình ảnh 4.3.2. 12: Giao diện trang thêm phòng

- Giao diện trang quản lý tài khoản

Màn hình hiển thị danh sách các tài khoản có trong hệ thống, admin có thể tạo, chỉnh sửa thông tin hoặc xóa tài khoản



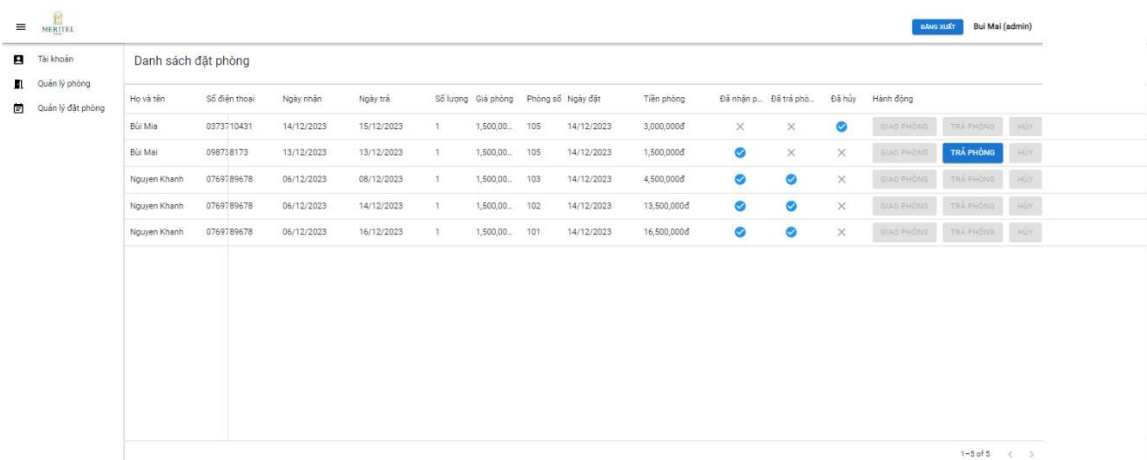
First name	Last name	Full name	Email	Số điện thoại	Phân quyền	Actions
Mai	Bui	Bui Mai	miabui@gmail.com	098988889	admin	XÓA CẬP NHẬT
Mai	Bui	Bui Mai	buimai@gmail.com	098738173	customer	XÓA CẬP NHẬT
Mai	Bui	Bui Mai	maibui@gmail.com	098738173	customer	XÓA CẬP NHẬT
Mia	Bui	Bui Mia	bmai0809@gmail.com	0373710431	customer	XÓA CẬP NHẬT

Hình ảnh 4.3.2. 13: Giao diện trang quản lý tài khoản phía admin

- Giao diện trang quản lý đặt phòng

Giao diện hiển thị danh sách các phòng được đặt từ phía khách hàng, admin có thể thay đổi trạng thái phòng

- Khi khách hàng đặt phòng từ phía giao diện người dùng, phòng được đặt sẽ hiển thị trong đây
- Khi khách hàng đến khách sạn nhận phòng, admin sẽ xác nhận thông tin của khách hàng và nhấn vào nút Giao phòng cho khách hàng
- Khi hết thời gian thuê phòng và khách hàng xuống trả phòng, admin nhấn nút trả phòng.
- Sau khi nhấn trả phòng, những khách hàng khác sẽ có thể đặt phòng
- Khi khách hàng đặt phòng mà không đến nhận phòng, admin có thể nhấn huỷ phòng, phòng đó sẽ tự động trống và những khách hàng khác có thể đặt



Họ và tên	Số điện thoại	Ngày nhận	Ngày trả	Số lượng	Giá phòng	Phòng số	Ngày đặt	Tiền phòng	Đã nhận p...	Đã trả pho...	Đã hủy	Hành động
Bui Mia	0373710431	14/12/2023	15/12/2023	1	1,500,000	105	14/12/2023	3,000,000đ	X	X	✓	ĐIỀU CHỈNH TRẢ PHÒNG Hủy
Bui Mai	098738173	13/12/2023	13/12/2023	1	1,500,000	105	14/12/2023	1,500,000đ	✓	X	X	ĐIỀU CHỈNH TRẢ PHÒNG Hủy
Nguyen Khanh	0769789678	06/12/2023	08/12/2023	1	1,500,000	103	14/12/2023	4,500,000đ	✓	✓	X	ĐIỀU CHỈNH TRẢ PHÒNG Hủy
Nguyen Khanh	0769789678	06/12/2023	14/12/2023	1	1,500,000	102	14/12/2023	13,500,000đ	✓	✓	X	ĐIỀU CHỈNH TRẢ PHÒNG Hủy
Nguyen Khanh	0769789678	06/12/2023	16/12/2023	1	1,500,000	101	14/12/2023	16,500,000đ	✓	✓	X	ĐIỀU CHỈNH TRẢ PHÒNG Hủy

Hình ảnh 4.3.2. 14: Giao diện trang quản lý đặt phòng

CHƯƠNG V: TỔNG KẾT

5.1 Kết quả đạt được

- Xây dựng một ứng dụng website đặt phòng khách sạn linh hoạt và dễ sử dụng.
 - Sử dụng ReactJS cho phần frontend giúp tạo ra giao diện người dùng mượt mà, linh hoạt và tương tác tốt.
 - Sử dụng Node.js cho phần backend đã cung cấp một hệ thống server linh hoạt, có khả năng mở rộng và xử lý yêu cầu từ người dùng.
- Tính năng đặt phòng và quản lý khách sạn:
 - Thực hiện thành công chức năng đặt phòng khách sạn với giao diện thân thiện và dễ sử dụng.
- Cung cấp tính năng quản lý đặt phòng, xem thông tin chi tiết của khách sạn, chọn phòng
- Tính năng quản lý khách sạn
 - Giúp quản trị viên dễ dàng quản lý danh sách người dùng và quản lý các đơn đặt phòng từ khách hàng
- Tương tác và trải nghiệm người dùng:
 - Tạo ra trải nghiệm người dùng tốt, từ quá trình tìm kiếm và chọn phòng đến việc đặt phòng
 - Tối ưu hóa giao diện người dùng để tương tác trực quan và dễ dàng hiểu.

5.2 Hạn chế

- Chưa thực hiện được hết các chức năng ban đầu đặt ra như quản lý đánh giá từ người dùng, quản lý các bài viết blog, quản lý doanh thu từ quản trị
- Bản thân em còn thiếu nhiều kiến thức và kinh nghiệm khi làm việc với nodejs và reactjs nên chưa phát triển được hết các chức năng đã đề ra
- Do bản thân em chưa dành nhiều thời gian để nghiên cứu và phân tích hệ thống nên hệ thống còn một số vấn đề về giao diện

5.3 Định hướng phát triển

- Phát triển thêm các tính năng mới liên quan đến thanh toán và thống kê để website có thể hoàn thiện và có thể đưa vào sử dụng được trong thực tế
- Tối ưu hoá hiệu năng và hiệu suất của hệ thống
- Bản thân em cần học hỏi và nghiên cứu thêm nhiều kiến thức về lập trình để hệ thống có thể hoàn thiện và tối ưu nhất cho người dùng

TÀI LIỆU THAM KHẢO

- [1] <https://legacy.reactjs.org/docs/getting-started.html>
- [2] https://www.w3schools.com/nodejs/nodejs_intro.asp.
- [3] <https://developer.mozilla.org/en-US/docs/Web/javascript>.
- [4] <https://www.mongodb.com/docs/>