

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CHUYÊN NGÀNH
HỌC KỲ 1, NĂM HỌC 2025 - 2026

**XÂY DỰNG WEBSITE QUẢN LÝ
PHÒNG KÝ TÚC XÁ CHO SINH VIÊN
TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giáo viên hướng dẫn:

ThS. Lê Minh Tự

Sinh viên thực hiện:

Họ tên: Đỗ Gia Hào

MSSV: 110122070

Lớp: DA22TTD

Vĩnh Long, tháng 12 năm 2025

TRƯỜNG KỸ THUẬT VÀ CÔNG NGHỆ
KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP ĐỒ ÁN CƠ SỞ NGÀNH
HỌC KỲ 1, NĂM HỌC 2024 - 2025

**XÂY DỰNG WEBSITE QUẢN LÝ
PHÒNG KÝ TÚC XÁ CHO SINH VIÊN
TRƯỜNG ĐẠI HỌC TRÀ VINH**

Giáo viên hướng dẫn:

ThS. Lê Minh Tự

Sinh viên thực hiện:

Họ tên: Đỗ Gia Hào

MSSV: 110122070

Lớp: DA22TTD

Vĩnh Long, tháng 12 năm 2025

[illegible]

Giáo viên hướng dẫn
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Trước khi đi sâu vào dự án, em xin gửi lời cảm ơn chân thành đến trường Đại học Trà Vinh đã tạo điều kiện cho em thực hiện dự án này, những cá nhân đã hỗ trợ và giúp đỡ em một cách tận tình. Cũng như là sự hỗ trợ và giúp đỡ của thầy Lê Minh Tự, người đã đóng vai trò quan trọng trong việc phát triển và hoàn thành dự án này.

Em rất cảm kích vì sự giúp đỡ của những người quanh em, những người đã dành thời gian, công sức và kiến thức của họ để giúp đỡ em. Các ý kiến đóng góp và sự hợp tác của mọi người là nguồn động lực giúp em phát triển bản thân.

Em rất quý trọng những người đã hỗ trợ và giúp đỡ em trong suốt thời gian qua, đặc biệt là sự giúp đỡ của thầy Lê Minh Tự, nhờ có sự giúp đỡ của cô mà em mới có thể thực hiện và hoàn thành dự án.

Một lần nữa, em xin cảm ơn trường Đại học Trà Vinh và thầy Lê Minh Tự đã giúp đỡ và em mong rằng sẽ nhận được sự ủng hộ của mọi người trong những dự án sắp tới.

Trân trọng

Sinh viên thực hiện

Đỗ Gia Hào

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN	10
1.1. Giới thiệu đề tài.....	10
1.2. Mục đích nghiên cứu.....	10
1.3. Đối tượng nghiên cứu.....	12
1.4. Phạm vi nghiên cứu.....	12
1.5. Phương pháp nghiên cứu.....	13
CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT	14
2.1. Tổng quan thiết kế web	14
2.1.1. Kiến trúc ứng dụng web.....	14
2.1.2. Lớp giao diện người dùng (Presentation Layer).....	15
2.1.3. Lớp xử lý nghiệp vụ (Business Logic Layer)	16
2.1.4. Lớp dữ liệu (Data Layer)	16
2.2. ReactJS	17
2.2.1. Đặc điểm	17
2.2.2. Kiến thức cơ bản về ReactJS	18
2.2.3. Chú thích trong ReactJS	18
2.2.4. Các thành phần trong ReactJS	19
2.2.4.1. Thành phần (Component).....	19
2.2.4.2. Props và State	19
2.2.4.3. Xử lý biểu mẫu (Form) trong ReactJS	20
2.2.5. Ứng dụng của ReactJS trong xây dựng website	21
2.2.6. Kết luận.....	22
2.3. NodeJS	22

2.3.1. Đặc điểm	23
2.3.2. Chú thích trong NodeJS	23
2.3.3. Cấu trúc cơ bản	23
2.3.4. Khai báo biến	24
2.3.5. Câu lệnh điều kiện	24
2.3.6. Kết nối với MySQL	24
2.3.7. Truy vấn dữ liệu	25
2.3.8. Xử lý form	26
2.3.9. Quản lý lỗi	26
2.3.10. Kết luận	27
2.4. Ngôn ngữ MySQL	27
2.4.1. Đặc điểm	28
2.4.2. Tạo cơ sở dữ liệu	28
2.4.3. Tạo cấu trúc bảng	29
2.4.4. Các câu lệnh truy vấn	29
2.4.5. Kết luận	31
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	32
3.1. Mô tả vấn đề	32
3.2. Xác định yêu cầu người dùng	32
3.3. Phân loại người dùng	33
3.4. Phân tích thiết kế hệ thống	33
3.4.1. Đặc tả yêu cầu hệ thống	33
3.4.1.1. Yêu cầu chức năng	33
3.4.1.2. Yêu cầu phi chức năng	34

3.4.2. Kiến trúc hệ thống.....	34
3.4.2.1. Lớp giao diện người dùng (Presentation Layer):	34
3.4.2.2. Lớp xử lý nghiệp vụ (Business Logic Layer):.....	34
3.4.2.3. Lớp dữ liệu (Data Layer):.....	35
3.4.3. Thiết kế dữ liệu	36
3.4.3.1. Mô hình ERD	36
3.4.3.2. Danh sách các thực thể và mối kết hợp	37
3.4.3.3. Chi tiết các thực thể	37
3.4.4. Thiết kế giao diện	42
3.4.4.1. Sơ đồ <i>Use Case</i>	42
3.4.4.2. Sơ đồ website.....	45
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	46
4.1. Dữ liệu thử nghiệm.....	46
4.2. Kết quả thực nghiệm	48
4.2.1. Chức năng trang người dùng sinh viên.....	48
4.2.1.1. Chức năng trang chủ.....	48
4.2.1.2. Chức năng xem giới thiệu ký túc xá.....	49
4.2.1.3. Chức năng hỗ trợ	50
4.2.1.4. Chức năng thanh toán hóa đơn	50
4.2.2. Chức năng trang quản lý ký túc xá	51
4.2.2.1. Chức năng hỗ trợ sinh viên.....	51
4.2.3. Trang quản trị viên.....	52
4.2.3.1. Chức năng tổng quan	52
4.2.3.2. Chức năng quản lý đơn đăng ký	52

4.2.3.3. Chức năng quản lý người dùng	53
4.2.3.4. Chức năng quản lý phòng ở.....	53
4.2.3.5. Chức năng quản lý thanh toán	54
4.2.3.6. Chức năng yêu cầu hỗ trợ.....	54
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	54
5.1. Kết luận	54
5.2. Hạn chế.....	55
5.3. Hướng phát triển.....	55
DANH MỤC TÀI LIỆU THAM KHẢO	56

DANH MỤC HÌNH ẢNH

Hình 3.1 Mô hình ERD.....	36
Hình 3.2 Sơ đồ Use Case về trang người dùng	42
Hình 3.3 Sơ đồ Use Case về trang nhân viên quản lý ký túc xá.....	43
Hình 3.4 Sơ đồ website.....	45
Hình 4.1 Dữ liệu của bảng dondangky	46
Hình 4.2 Dữ liệu của bảng hoadon	46
Hình 4.3 Dữ liệu của bảng hopdong.....	46
Hình 4.4 Dữ liệu của bảng phong	46
Hình 4.5 Dữ liệu của bảng sinhvien	46
Hình 4.6 Dữ liệu của bảng taikhoan	47
Hình 4.7 Dữ liệu của bảng yeucauhotro	47
Hình 4.8 Giao diện trang chủ.....	48
Hình 4.9 Giao diện giới thiệu ký túc xá.....	49
Hình 4.10 Giao diện hỗ trợ	50
Hình 4.11 Giao diện thanh toán.....	51
Hình 4.12 Giao diện chức năng hỗ trợ sinh viên.....	51
Hình 4.13 Giao diện chức năng tổng quát	52
Hình 4.14 Giao diện chức năng quản lý đơn đăng ký	52
Hình 4.15 Giao diện chức năng quản lý người dùng.....	53
Hình 4.16 Giao diện chức năng quản lý phòng ở.....	53
Hình 4.17 Giao diện chức năng quản lý thanh toán	54
Hình 4.18 Giao diện chức năng yêu cầu hỗ trợ	54

DANH MỤC BẢNG BIỂU

Bảng 3.1 Bảng thực thể	37
Bảng 3.2 Bảng thuộc tính của thực thể dondangky	37
Bảng 3.3 Bảng thuộc tính của thực thể hoadon	38
Bảng 3.4 Bảng thuộc tính của thực thể hopdong	39
Bảng 3.5 Bảng thuộc tính của thực thể phong	39
Bảng 3.6 Bảng thuộc tính của thực thể sinhvien	40
Bảng 3.7 Bảng thuộc tính của thực thể taikhoan	41
Bảng 3.8 Bảng thuộc tính của thực thể yeucauhotro	41

CHƯƠNG 1. TỔNG QUAN

1.1. Giới thiệu đề tài

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 đang diễn ra mạnh mẽ, việc ứng dụng công nghệ thông tin vào quản lý hành chính và giáo dục là xu hướng tất yếu. Tại Việt Nam, chuyển đổi số trong trường đại học không chỉ dừng lại ở việc giảng dạy trực tuyến mà còn bao gồm cả công tác quản lý cơ sở vật chất và đời sống sinh viên. Trường Đại học Trà Vinh (TVU) là một trong những cơ sở giáo dục đại học lớn với quy mô đào tạo đa ngành, đa hệ. Số lượng sinh viên theo học ngày càng tăng, kéo theo nhu cầu về chỗ ở nội trú tại ký túc xá (KTX) cũng tăng cao. Ký túc xá không chỉ là nơi lưu trú mà còn là môi trường rèn luyện, sinh hoạt của sinh viên. Tuy nhiên, công tác quản lý ký túc xá theo phương pháp truyền thống (sử dụng sổ sách, Excel rời rạc) hiện nay đang bộc lộ nhiều hạn chế như: tốn nhiều thời gian tra cứu, khó khăn trong việc thống kê báo cáo, quy trình đăng ký phòng phức tạp, và việc cập nhật thông tin điện nước hay sự cố cơ sở vật chất thường bị chậm trễ. Xuất phát từ thực tế đó, đề tài "Xây dựng website quản lý phòng ký túc xá cho sinh viên trường Đại học Trà Vinh" được thực hiện nhằm cung cấp một giải pháp công nghệ hiện đại, giúp số hóa quy trình quản lý, nâng cao hiệu quả làm việc của ban quản lý và mang lại sự thuận tiện cho sinh viên.

1.2. Mục đích nghiên cứu

Đề tài này hướng đến việc thiết kế và xây dựng một hệ thống website quản lý phòng ký túc xá nhằm hỗ trợ nhà trường trong việc quản lý sinh viên nội trú, thông tin phòng ở, hợp đồng thuê, hóa đơn điện nước và các hoạt động liên quan. Hệ thống giúp sinh viên dễ dàng đăng ký chỗ ở, tra cứu thông tin phòng, thanh toán và gửi yêu cầu hỗ trợ trực tuyến, đồng thời hỗ trợ ban quản lý ký túc xá và quản trị viên trong việc theo dõi, kiểm soát và xử lý dữ liệu một cách tự động, chính xác và minh bạch.

Cụ thể:

- **Đối với người dùng (Sinh viên):**

- Đăng ký tài khoản và đăng nhập vào hệ thống.
- Xem và cập nhật thông tin cá nhân.
- Tra cứu danh sách phòng còn trống, đăng ký thuê phòng.
- Xem thông tin hợp đồng thuê, hóa đơn tiền phòng, điện, nước.
- Gửi yêu cầu hỗ trợ hoặc phản hồi cho ban quản lý.
- **Đối với nhân viên quản lý ký túc:**
 - Quản lý thông tin sinh viên (thêm, sửa, xóa, tìm kiếm).
 - Quản lý danh sách phòng, tòa nhà và tình trạng phòng (còn trống, đã thuê, đang bảo trì).
 - Quản lý hợp đồng thuê và hóa đơn thanh toán.
 - Tiếp nhận và xử lý các yêu cầu hỗ trợ từ sinh viên.
 - Lập và xuất báo cáo thống kê định kỳ về tình trạng phòng và sinh viên nội trú.
- **Đối với quản trị viên (Admin):**
 - Quản lý toàn bộ tài khoản người dùng (sinh viên, nhân viên).
 - Phân quyền và kiểm soát hoạt động của từng nhóm người dùng.
 - Theo dõi, giám sát dữ liệu hệ thống và sao lưu cơ sở dữ liệu.
 - Xem, tổng hợp và xuất báo cáo thống kê toàn hệ thống.
 - Cấu hình các thông tin chung của ký túc xá như mức giá phòng, chính sách, thông báo.

Nghiên cứu này sử dụng ReactJS để xây dựng frontend, NodeJS (ExpressJS) để xây dựng backend, MySQL để thiết kế và quản lý cơ sở dữ liệu. Sự kết hợp này góp phần tạo nên một ứng dụng web trực quan, linh hoạt và đáp ứng tốt nhu cầu học tập của sinh viên.

1.3. Đối tượng nghiên cứu

Dự án hướng đến việc xây dựng một hệ thống web hỗ trợ cung cấp các chức năng giúp sinh viên dễ dàng đăng ký chỗ ở, tra cứu thông tin phòng, thanh toán và gửi yêu cầu hỗ trợ trực tuyến, đồng thời hỗ trợ ban quản lý ký túc xá và quản trị viên trong việc theo dõi, kiểm soát và xử lý dữ liệu một cách tự động, chính xác và minh bạch.

1.4. Phạm vi nghiên cứu

Các ngôn ngữ được dùng:

- Sử dụng ReactJS (kết hợp HTML, CSS và JavaScript) để xây dựng giao diện người dùng hiện đại, thân thiện và hỗ trợ tương tác động, cập nhật nội dung theo thời gian thực.
- Sử dụng NodeJS để xây dựng backend, xử lý logic nghiệp vụ, kiểm tra dữ liệu và quản lý yêu cầu từ phía người dùng.
- Sử dụng MySQL để thiết kế và quản trị cơ sở dữ liệu lưu trữ thông tin người dùng.

Chức năng ứng dụng:

- Chức năng đăng ký ở cho phép sinh viên đăng ký ký túc xá.
- Chức năng quản lý đơn đăng ký quản trị viên quản lý đơn đăng ký ký túc xá của sinh viên.
- Quản lý tài khoản người dùng bao gồm đăng ký, đăng nhập và chỉnh sửa thông tin cá nhân.
- Chức năng yêu cầu hỗ trợ quản trị viên và quản lý ký túc xá có thể nhận yêu cầu hỗ trợ của sinh viên,

Quy mô dự án:

- Ứng dụng tập trung vào việc hỗ trợ sinh viên

- Hệ thống được thiết kế để phục vụ sinh viên, giảng viên hoặc người dùng.

- Các chức năng được xây dựng ở mức phù hợp cho một hệ thống web đơn giản nhưng đủ hoàn chỉnh để sử dụng thực tế.

Phạm vi nghiên cứu tập trung vào giao diện người dùng, xử lý logic phía máy chủ, thiết kế cơ sở dữ liệu và sự tương tác giữa các thành phần. Việc giới hạn phạm vi giúp tối ưu thời gian và nguồn lực, đảm bảo hệ thống hoạt động hiệu quả và đáp ứng nhu cầu chia sẻ tài liệu và hỏi đáp của sinh viên.

1.5. Phương pháp nghiên cứu

- Phương pháp nghiên cứu: Tìm hiểu tài liệu, giáo trình và các nguồn trực tuyến về Node.js, MySQL và các công nghệ giao diện như HTML, CSS, JavaScript, ReactJS để hiểu rõ và nắm vững các công nghệ sử dụng trong hệ thống.

- Phương pháp phân tích và thiết kế: Sử dụng sơ đồ Use Case, mô hình phân tích hướng chức năng và mô hình ERD để xác định các chức năng, luồng xử lý và mối quan hệ giữa các bảng dữ liệu.

- Phương pháp xây dựng hệ thống: Phát triển backend bằng Node.js để xây dựng logic nghiệp vụ theo kiến trúc mô-đun, frontend bằng HTML, CSS, JavaScript, React. Cơ sở dữ liệu được quản lý bằng MySQL.

- Phương pháp kiểm thử: Tiến hành kiểm thử chức năng, tích hợp và giao diện nhằm đảm bảo hệ thống hoạt động đúng yêu cầu và ổn định.

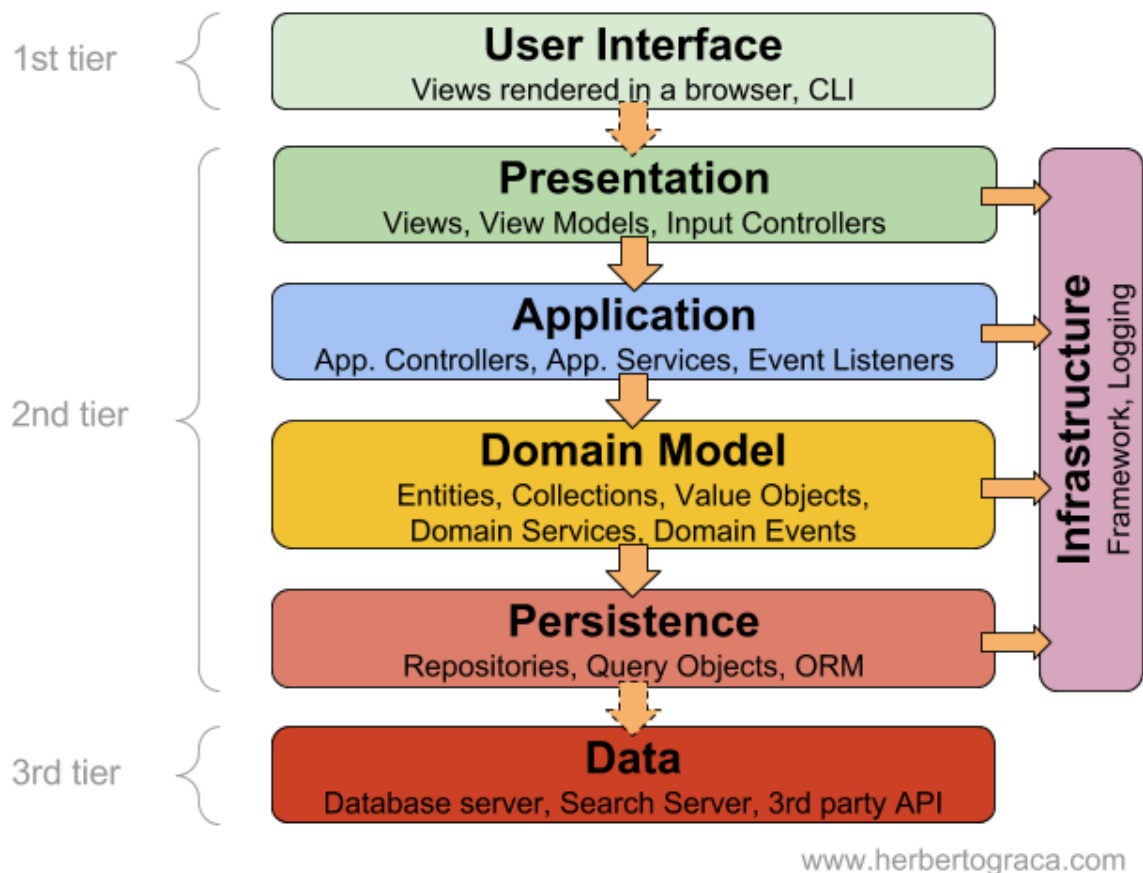
- Phương pháp triển khai: Cài đặt và chạy thử hệ thống trên môi trường Xampp.

CHƯƠNG 2. NGHIÊN CỨU LÝ THUYẾT

2.1. Tổng quan thiết kế web

2.1.1. Kiến trúc ứng dụng web

Ứng dụng web trong đề tài được thiết kế dựa trên kiến trúc phân lớp (Layered Architecture). Đây là mô hình kiến trúc phổ biến trong phát triển ứng dụng web hiện đại, cho phép tách biệt rõ ràng các thành phần theo chức năng, giúp hệ thống dễ phát triển, mở rộng và bảo trì.



Hình 2.1 Kiến trúc phân lớp (Layered Architecture)

Kiến trúc phân lớp chia hệ thống thành ba lớp chính:

Lớp giao diện người dùng (Presentation Layer) chịu trách nhiệm hiển thị dữ liệu và xử lý các tương tác trực tiếp với người dùng; mọi thao tác như nhập

thông tin, tìm kiếm hay xem phòng trọ đều được lớp này tiếp nhận và gửi về backend thông qua API.

Lớp xử lý nghiệp vụ (Business Logic Layer) đóng vai trò trung tâm, nơi thực hiện các quy tắc nghiệp vụ, xử lý yêu cầu từ frontend, kiểm tra dữ liệu, xác thực người dùng và điều phối giao tiếp với cơ sở dữ liệu.

Lớp dữ liệu (Data Layer) đảm nhận việc lưu trữ, truy vấn và quản lý các bảng dữ liệu trong hệ quản trị MySQL, bảo đảm tính toàn vẹn và nhất quán của thông tin. Ba lớp kết hợp với nhau giúp hệ thống hoạt động ổn định, dễ mở rộng và dễ bảo trì.

2.1.2. Lớp giao diện người dùng (Presentation Layer)

Lớp giao diện người dùng (Presentation Layer) là tầng chịu trách nhiệm hiển thị dữ liệu và xử lý toàn bộ các tương tác trực tiếp giữa người dùng và hệ thống. Trong đề tài này, lớp giao diện được phát triển bằng ReactJS, một thư viện JavaScript hiện đại hỗ trợ xây dựng giao diện theo hướng thành phần (component-based), giúp tăng khả năng tái sử dụng, cải thiện hiệu suất và tối ưu trải nghiệm người dùng.

Tầng này đảm nhiệm các chức năng chính như: hiển thị thông tin phòng trọ, xử lý thao tác tìm kiếm, tiếp nhận dữ liệu từ biểu mẫu (đăng nhập, đăng ký...) và gửi yêu cầu đến backend thông qua API. Các công nghệ được sử dụng bao gồm:

- **ReactJS:** Dùng để xây dựng cấu trúc giao diện và quản lý trạng thái. ReactJS hỗ trợ cập nhật dữ liệu theo thời gian thực, xử lý điều hướng và mang lại trải nghiệm tương tác mượt mà.
- **HTML & CSS:** Kết hợp trong các thành phần React để tạo bố cục, định dạng nội dung và đảm bảo tính thẩm mỹ của giao diện. CSS giúp giao diện thích ứng tốt trên nhiều thiết bị như máy tính, điện thoại và máy tính bảng.
- **JavaScript:** Được sử dụng để triển khai các logic phía client, xử lý sự kiện, gọi API, cập nhật trạng thái và tạo ra những tương tác động trong ứng dụng.

Với sự kết hợp của các công nghệ web hiện đại, lớp giao diện người dùng bảo đảm khả năng phản hồi nhanh, hiển thị trực quan và mang lại trải nghiệm liền mạch khi người dùng thực hiện tra cứu, xem thông tin phòng trọ hoặc tương tác với các chức năng khác của hệ thống.

2.1.3. Lớp xử lý nghiệp vụ (Business Logic Layer)

Lớp xử lý nghiệp vụ (Business Logic Layer) là tầng trung tâm của hệ thống, nơi thực thi các quy tắc nghiệp vụ, xử lý yêu cầu từ giao diện người dùng và điều phối quá trình truy xuất dữ liệu. Trong đề tài này, lớp nghiệp vụ được phát triển bằng NodeJS kết hợp với ExpressJS, cho phép xây dựng các API nhẹ, linh hoạt và dễ mở rộng.

Backend đảm nhiệm việc tiếp nhận yêu cầu từ frontend, kiểm tra và xử lý dữ liệu, thực hiện xác thực người dùng, áp dụng các quy tắc nghiệp vụ và tương tác với cơ sở dữ liệu. Các công nghệ chính được sử dụng gồm:

- **NodeJS**: Nền tảng chạy JavaScript phía server, hỗ trợ xử lý bất đồng bộ, phù hợp với các ứng dụng web cần tốc độ phản hồi nhanh và khả năng mở rộng. NodeJS thực hiện các tác vụ như xử lý logic nghiệp vụ, quản lý phiên, kiểm tra dữ liệu và điều hướng yêu cầu.

- **ExpressJS**: Framework nhẹ cho NodeJS, cung cấp cấu trúc rõ ràng để xây dựng API theo mô hình RESTful. ExpressJS hỗ trợ định tuyến (routing), xử lý middleware, quản lý request/response và kết nối với cơ sở dữ liệu.

Thông qua sự kết hợp của NodeJS và ExpressJS, lớp nghiệp vụ đảm bảo xử lý yêu cầu hiệu quả, duy trì tính nhất quán dữ liệu và tạo cầu nối giữa frontend và cơ sở dữ liệu một cách nhanh chóng, ổn định và dễ bảo trì.

2.1.4. Lớp dữ liệu (Data Layer)

Lớp dữ liệu (Data Layer) là tầng chịu trách nhiệm lưu trữ, quản lý và truy vấn toàn bộ thông tin của hệ thống. Trong đề tài này, tầng dữ liệu được xây dựng bằng MySQL, một hệ quản trị cơ sở dữ liệu quan hệ mạnh mẽ, hỗ trợ tốt cho các hệ thống web cần quản lý dữ liệu có cấu trúc và đảm bảo tính toàn vẹn.

Tầng dữ liệu thực hiện các chức năng chính như: lưu trữ thông tin người dùng, tài liệu học tập, câu hỏi, câu trả lời, đánh giá, phân loại theo môn học và ngành cũng như quản lý các tương tác của người dùng đối với tài liệu và nội dung hỏi đáp. Backend (NodeJS – ExpressJS) kết nối với MySQL thông qua các truy vấn SQL để xử lý dữ liệu theo yêu cầu của người dùng.

Một số đặc điểm nổi bật của lớp dữ liệu gồm:

- MySQL hỗ trợ thiết kế dữ liệu theo mô hình quan hệ, cho phép xây dựng các bảng có khóa chính, khóa ngoại để đảm bảo tính liên kết giữa người dùng.
- Cơ sở dữ liệu được thiết kế dựa trên mô hình ERD của hệ thống, giúp tối ưu hóa việc truy vấn và tránh trùng lặp dữ liệu.
- Tầng backend sử dụng các thư viện như mysql2 hoặc ORM như Sequelize để giao tiếp với MySQL, đảm bảo quá trình thêm, sửa, xóa và truy vấn dữ liệu được thực hiện nhanh chóng và an toàn.

Nhờ sử dụng MySQL làm nền tảng lưu trữ, lớp dữ liệu đảm bảo hệ thống hoạt động ổn định, dữ liệu nhất quán và đáp ứng tốt nhu cầu truy cập, chia sẻ thông tin học tập và hỏi đáp giữa sinh viên.

2.2. ReactJS

ReactJS là một thư viện JavaScript mã nguồn mở do Facebook phát triển, được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web hiện đại. ReactJS cho phép phát triển UI theo hướng thành phần (component-based), giúp tối ưu khả năng tái sử dụng mã nguồn, tăng tính linh hoạt và cải thiện hiệu suất hiển thị. Trong hệ thống chia sẻ tài liệu học tập và hỏi đáp giữa sinh viên, ReactJS đóng vai trò quan trọng trong việc tạo ra giao diện trực quan, phản hồi nhanh và dễ mở rộng.

2.2.1. Đặc điểm

ReactJS được đánh giá là một trong những thư viện frontend phổ biến nhất nhờ sở hữu nhiều ưu điểm nổi bật. Trước hết, ReactJS phát triển giao diện theo hướng thành phần (component-based), cho phép chia nhỏ giao diện thành các đơn vị độc lập, dễ tái sử dụng và đơn giản hóa quá trình bảo trì. Bên cạnh đó, ReactJS sử dụng cơ chế Virtual DOM giúp tối ưu hóa việc cập nhật giao diện: chỉ những thành phần thực sự thay đổi mới được render lại, từ đó cải thiện đáng kể hiệu suất so với cách thao tác trực tiếp trên DOM truyền thống. Thư viện này cũng áp dụng mô hình

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

truyền dữ liệu một chiều (one-way data binding), giúp luồng dữ liệu trở nên rõ ràng, dễ kiểm soát và hạn chế xung đột trạng thái. Ngoài ra, ReactJS hỗ trợ JSX – một phần mở rộng cú pháp cho phép viết giao diện bằng mã giống HTML ngay trong JavaScript, giúp mã nguồn trực quan, dễ đọc và dễ quản lý hơn. Nhờ sự kết hợp của các đặc điểm trên, ReactJS mang lại tốc độ phản hồi nhanh và hiệu quả cao, đặc biệt phù hợp với các ứng dụng web có yêu cầu cập nhật dữ liệu liên tục.

2.2.2. Kiến thức cơ bản về ReactJS

Trong ReactJS, ứng dụng được xây dựng dựa trên các thành phần (components). Một component đơn giản có thể được định nghĩa như sau:

```
function Header() {  
  return (  
    <header className="header">  
      <h1>Hệ thống chia sẻ tài liệu học tập</h1>  
    </header>  
  );  
}
```

Cấu trúc cơ bản của một ứng dụng React gồm:

- **index.html:** Tập HTML gốc chứa phần tử <div id="root"></div> là nơi React gắn giao diện.
- **index.js:** Điểm khởi chạy ứng dụng, dùng để render App vào giao diện HTML.
- **App.js:** Thành phần cha quản lý toàn bộ giao diện và điều hướng.
- **Các component con:** Header, Footer, danh sách tài liệu, giao diện câu hỏi, biểu mẫu đăng nhập,...

2.2.3. Chú thích trong ReactJS

Trong ReactJS, chú thích được viết trong JavaScript bằng các cú pháp:

- Chú thích một dòng: // nội dung
- Chú thích nhiều dòng: /* nội dung */

- Chú thích trong JSX: `{/* Đây là chú thích trong JSX */}`

2.2.4. Các thành phần trong ReactJS

2.2.4.1. Thành phần (Component)

Component trong ReactJS dùng để chia giao diện thành các phần độc lập, mỗi phần tự quản lý hiển thị và logic của mình. Chúng giúp ứng dụng dễ bảo trì và tái sử dụng. Hiện nay, ReactJS hỗ trợ hai kiểu component chính:

- **Functional Component (phổ biến nhất):** Là dạng component được xây dựng dưới dạng hàm, có cú pháp ngắn gọn và hiệu suất tốt. Nhờ sự hỗ trợ của React Hooks, Functional Component có thể quản lý trạng thái và xử lý vòng đời, trở thành lựa chọn tiêu chuẩn trong các ứng dụng React hiện đại. Ví dụ Functional Component:

```
function Button(props) {  
  return <button>{props.label}</button>;  
}
```

- **Class Component (ít dùng hơn):** Được xây dựng dựa trên cú pháp lớp của JavaScript, hỗ trợ state và các phương thức vòng đời truyền thống. Mặc dù ít được sử dụng trong các dự án mới do cú pháp phức tạp hơn, Class Component vẫn quan trọng trong việc duy trì và hiểu các ứng dụng React phiên bản cũ. Ví dụ Class Component:

```
class Welcome extends React.Component {  
  render() {  
    return <h2>Chào mừng, {this.props.name}</h2>;  
  }  
}
```

2.2.4.2. Props và State

Trong ReactJS, *props* và *state* là hai khái niệm quan trọng dùng để quản lý dữ liệu và điều khiển cách component hoạt động.

- **Props** (viết tắt của “properties”) là dữ liệu được truyền từ component cha sang component con, giúp component con có thể hiển thị nội dung hoặc hành vi

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

theo giá trị được cung cấp. Props có tính chất *bất biến* (immutable), nghĩa là component con không được phép thay đổi giá trị props nhận vào. Ví dụ một component nhận props và hiển thị lời chào như sau:

```
function Hello(props) {  
    return <h3>Xin chào, {props.name}!</h3>;  
}
```

• **State** là dữ liệu nội tại của component, có thể thay đổi theo thời gian thông qua các thao tác của người dùng hoặc các sự kiện trong hệ thống. State giúp giao diện có khả năng phản hồi và cập nhật động. Khi state thay đổi, component sẽ được render lại để hiển thị dữ liệu mới. Dưới đây là ví dụ quản lý state trong Functional Component bằng React Hook useState:

```
import { useState } from "react";  
function Counter() {  
    const [count, setCount] = useState(0);  
    return (  
        <div>  
            <p>Số lần bấm: {count}</p>  
            <button onClick={() => setCount(count +  
1)}>Tăng</button>  
        </div>  
    );  
}
```

Props và state đóng vai trò trọng tâm trong cơ chế hoạt động của ReactJS, giúp component vừa có khả năng nhận dữ liệu từ bên ngoài, vừa tự quản lý và cập nhật thông tin theo các tương tác của người dùng.

2.2.4.3. Xử lý biểu mẫu (Form) trong ReactJS

Trong React, dữ liệu của form được quản lý bằng state. Code ví dụ:

```
import { useState } from "react";  
function SearchForm() {  
    const [keyword, setKeyword] = useState("");
```

```
const handleSubmit = (e) => {  
    e.preventDefault();  
    if (!keyword.trim()) {  
        alert("Thanh tìm kiếm rỗng.");  
        return;  
    }  
    console.log("Từ khóa:", keyword);  
};  
return (  
    <form onSubmit={handleSubmit}>  
        <input  
            type="text"  
            value={keyword}  
            onChange={(e) => setKeyword(e.target.value)}  
            placeholder="Nhập từ khóa..."  
        />  
        <button type="submit">Tìm kiếm</button>  
    </form>  
);  
}
```

2.2.5. Ứng dụng của ReactJS trong xây dựng website

ReactJS được sử dụng để xây dựng giao diện người dùng một cách linh hoạt và hiệu quả trong các ứng dụng web hiện đại. Một số ứng dụng quan trọng của ReactJS bao gồm:

- **Xây dựng giao diện người dùng linh hoạt:** ReactJS cho phép chia giao diện thành các component độc lập, dễ tái sử dụng và dễ bảo trì.

- **Cập nhật giao diện hiệu quả:** Nhờ cơ chế Virtual DOM, ReactJS giúp website phản hồi nhanh và cập nhật dữ liệu động mà không cần tải lại toàn bộ trang.

- **Hỗ trợ điều hướng trong ứng dụng:** React Router cho phép xây dựng hệ thống điều hướng mượt mà trong ứng dụng đơn trang (SPA).

- **Kết hợp dễ dàng với các công nghệ khác:** ReactJS hoạt động tốt cùng CSS, thư viện UI, API backend và các công cụ JavaScript, giúp mở rộng tính năng và tối ưu thiết kế giao diện.

- **Tăng hiệu suất và trải nghiệm người dùng:** Nhờ cơ chế render tối ưu và mô hình quản lý trạng thái rõ ràng, ReactJS giúp website vận hành ổn định và mang lại trải nghiệm liền mạch cho người dùng.

2.2.6. Kết luận

ReactJS là một công nghệ mạnh mẽ và hiện đại, đóng vai trò quan trọng trong việc xây dựng giao diện ứng dụng web. Với mô hình phát triển hướng thành phần, cơ chế Virtual DOM và khả năng mở rộng linh hoạt, ReactJS giúp tạo ra giao diện trực quan, dễ sử dụng và hiệu suất cao. Việc ứng dụng ReactJS trong hệ thống chia sẻ tài liệu học tập và hỏi đáp giữa sinh viên giúp nâng cao trải nghiệm người dùng và đảm bảo tính ổn định của giao diện trong quá trình vận hành.

2.3. NodeJS

NodeJS là một môi trường chạy JavaScript phía server, cho phép lập trình viên phát triển các ứng dụng web động bằng ngôn ngữ JavaScript cả ở phía client và server. Nhờ sử dụng cơ chế xử lý bất đồng bộ (asynchronous) và non-blocking I/O, NodeJS đặc biệt phù hợp với các ứng dụng yêu cầu tốc độ cao và khả năng xử lý nhiều yêu cầu đồng thời. NodeJS thường được sử dụng để xây dựng API backend, xử lý dữ liệu, giao tiếp với cơ sở dữ liệu và triển khai nhiều tác vụ phía server một cách linh hoạt và hiệu quả. [4]

2.3.1. Đặc điểm

NodeJS là nền tảng lập trình phổ biến nhờ vào hiệu năng cao, khả năng mở rộng tốt và sử dụng chung một ngôn ngữ JavaScript cho cả frontend và backend. Được xây dựng trên V8 Engine của Google Chrome, NodeJS có tốc độ xử lý nhanh và tối ưu cho các tác vụ I/O. Các thư viện phong phú thông qua npm (Node Package Manager) cho phép mở rộng tính năng dễ dàng, đồng thời cộng đồng phát triển rộng lớn giúp NodeJS trở thành một lựa chọn mạnh mẽ cho các ứng dụng web hiện đại.

Các đặc điểm nổi bật của NodeJS gồm:

- **Bất đồng bộ và Non-blocking I/O:** Giúp xử lý nhiều yêu cầu cùng lúc mà không làm nghẽn hệ thống.
- **Một ngôn ngữ thống nhất:** Sử dụng JavaScript ở cả frontend và backend.
- **Hiệu năng cao:** Dựa trên V8 Engine tối ưu hóa tốc độ xử lý.
- **Hệ sinh thái phong phú:** npm cung cấp hàng trăm nghìn thư viện hỗ trợ phát triển nhanh chóng.

2.3.2. Chú thích trong NodeJS

NodeJS sử dụng cú pháp chú thích của JavaScript:

- Chú thích một dòng: `// nội dung`
- Chú thích nhiều dòng: `/* nội dung */`

2.3.3. Cấu trúc cơ bản

Trong NodeJS, mã nguồn được lưu dưới dạng tệp .js và được thực thi bằng lệnh node. Một ứng dụng backend thường sử dụng ExpressJS để xây dựng API.

Ví dụ:

```
const express = require('express');  
  
const app = express();  
  
app.get('/redirect', (req, res) => {
```

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

```
res.redirect('/login');

});

app.listen(3000, () => {

    console.log('Server đang chạy trên cổng 3000');

});
```

Trong đó:

- `require('express')`: Tải module Express vào chương trình.
- `app.get(...)`: Xử lý yêu cầu GET tại đường dẫn chỉ định.
- `res.redirect()`: Chuyển hướng người dùng sang URL khác.
- `app.listen()`: Khởi động server.

2.3.4. Khai báo biến

NodeJS hỗ trợ khai báo biến bằng ba từ khóa: `var`, `let`, và `const`. Trong thực tế, `let` và `const` được ưu tiên do an toàn và rõ ràng hơn.

Ví dụ:

```
const dbName = "shared_docs";

let totalDocuments = 100;
```

2.3.5. Câu lệnh điều kiện

NodeJS sử dụng các câu lệnh điều kiện giống JavaScript như: `if`, `else if`, `else` và `switch`.

Ví dụ:

```
if (!req.session.user) {

    return res.redirect('/login');

}
```

2.3.6. Kết nối với MySQL

NodeJS kết nối MySQL thông qua các thư viện như `mysql2` hoặc ORM như `Sequelize`.

Ví dụ kết nối bằng mysql2:

```
const mysql = require('mysql2');

const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'shared_docs'
});

connection.connect((err) => {
  if (err) {
    console.error('Kết nối thất bại:', err);
    return;
  }
  console.log('Kết nối MySQL thành công');
});
```

2.3.7. Truy vấn dữ liệu

NodeJS cho phép thực hiện truy vấn SQL linh hoạt thông qua hàm `query()` của `mysql2`.

Ví dụ:

```
const mysql = require('mysql2');

const connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'shared_docs'
});
```

```
connection.connect((err) => {

    if (err) {

        console.error('Kết nối thất bại:', err);

        return;

    }

    console.log('Kết nối MySQL thành công');

});
```

2.3.8. Xử lý form

Trong ExpressJS, dữ liệu gửi từ form được xử lý thông qua req.body sau khi cấu hình middleware express.urlencoded() hoặc express.json().

Ví dụ:

```
app.post('/login', (req, res) => {

    const { username, password } = req.body;

    const sql = "SELECT * FROM users WHERE username = ? AND password = ?";

    connection.query(sql, [username, password], (err, result) => {

        if (result.length > 0) {

            res.send("Đăng nhập thành công");

        } else {

            res.send("Sai thông tin đăng nhập");

        }

    });

});
```

2.3.9. Quản lý lỗi

NodeJS hỗ trợ quản lý lỗi bằng khối try-catch, đặc biệt khi xử lý promise hoặc async/await.

Ví dụ:

```
async function deleteFile(fileId) {
```

```
try {
    const result = await fetch('/delete-file', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify({ id: fileId })
    });

    const response = await result.json();

    if (response.success) {
        console.log("Xóa thành công");
    } else {
        console.log("Lỗi khi xóa");
    }
} catch (error) {
    console.error("Đã xảy ra lỗi:", error);
}
```

2.3.10. Kết luận

NodeJS là một nền tảng mạnh mẽ cho phát triển backend hiện đại nhờ khả năng xử lý bất đồng bộ, hiệu năng cao và sử dụng chung ngôn ngữ JavaScript trên toàn bộ hệ thống. Việc ứng dụng NodeJS giúp đơn giản hóa quá trình xây dựng API, tối ưu hóa tốc độ phản hồi và tăng tính linh hoạt trong việc mở rộng ứng dụng web. Đây là lựa chọn phù hợp cho các hệ thống yêu cầu khả năng xử lý đồng thời lớn và tốc độ xử lý nhanh.

2.4. Ngôn ngữ MySQL

MySQL là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) hữu hiệu, được thiết kế để quản lý và truy xuất dữ liệu có cấu trúc một cách nhanh chóng và tin cậy. MySQL thường được tích hợp trong các ứng dụng web lớn như WordPress, Drupal, và Magento. [5]

2.4.1. Đặc điểm

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (“RDBMS”) được sử dụng rộng rãi trong nhiều lĩnh vực như quản lý doanh nghiệp, phát triển web, và ứng dụng di động. Tính phổ biến của nó bắt nguồn từ hiệu quả vận hành, độ tin cậy và tính linh hoạt cao trong việc tích hợp với nhiều ngôn ngữ lập trình.

- Hiệu suất cao: MySQL đã chứng minh khả năng xử lý khối lượng dữ liệu cực lớn với độ trễ thấp, đáp ứng tính toàn vẹn cho các hệ thống giao dịch trực tuyến và ứng dụng doanh nghiệp có khối lượng truy cập lớn.

- Tính bền vững: MySQL đảm bảo bảo toàn dữ liệu thông qua các cơ chế sao lưu (backup), khôi phục (recovery), và hỗ trợ giao dịch (transaction support). Khả năng này rất hữu ích cho các doanh nghiệp cần bảo đảm sự liên tục của hệ thống.

- Khả năng tích hợp: MySQL hỗ trợ nhiều giao thức lập trình ứng dụng (API) và khả năng tích hợp dễ dàng với các ngôn ngữ như PHP, Python, Java, Ruby và Node.js. Tính linh hoạt này giúp nó phù hợp cho nhiều loại hệ thống và ứng dụng từ web đơn giản đến các hệ thống quy mô lớn.

Nhờ vào những đặc điểm nêu trên, MySQL không chỉ là sự lựa chọn hàng đầu cho những nhà phát triển mà còn là một nền tảng quan trọng trong hầu hết các hệ sinh thái phần mềm hiện đại.

2.4.2. Tạo cơ sở dữ liệu

MySQL sử dụng ngôn ngữ SQL (Structured Query Language) để thực hiện các thao tác trên cơ sở dữ liệu. Dưới đây là một số khái niệm và thao tác cơ bản:

Một tập hợp có tổ chức của dữ liệu, thường được lưu trữ trong các bảng.

Cú pháp tạo cơ sở dữ liệu:

```
create database ktx_db
```

Cú pháp chọn cơ sở dữ liệu:

```
use ktx_db
```

2.4.3. Tạo cấu trúc bảng

Cấu trúc dữ liệu chính trong MySQL bao gồm các bản ghi (rows) và các trường (columns).

Cú pháp tạo bảng:

```
CREATE TABLE `quantrivien` (  
  `maQuanTriVien` int(11) NOT NULL,  
  `hoTenQTV` varchar(255) NOT NULL,  
  `emailQTV` varchar(255) NOT NULL,  
  `matKhuQTV` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_vietnamese_ci;
```

2.4.4. Các câu lệnh truy vấn

Câu lệnh SELECT: Truy xuất dữ liệu từ bảng.

```
SELECT maQuanTriVien, hoTenQTV, emailQTV FROM quantrivien  
ORDER BY maQuanTriVien;
```

Câu lệnh INSERT: Thêm dữ liệu mới vào bảng.

```
INSERT INTO `nganh` (`maNganh`, `tenNganh`) VALUES  
(1, 'Công nghệ thông tin');
```

Câu lệnh UPDATE: Cập nhật dữ liệu trong bảng.

```
UPDATE quantrivien SET hoTenQTV = adminTest WHERE  
maQuanTriVien = 3;
```

Câu lệnh DELETE: Xóa dữ liệu khỏi bảng.

```
DELETE FROM quantrivien WHERE maQuanTriVien = 3;
```

Khóa chính (Primary Key): Một cột hoặc nhóm cột xác định duy nhất mỗi hàng trong bảng. Có 2 cách tạo khóa chính:

Tạo khóa chính ngay khi tạo bảng:

```
CREATE TABLE `quantrivien` (  
  `maQuanTriVien` int(11) NOT NULL PRIMARY KEY,  
  `hoTenQTV` varchar(255) NOT NULL,  
  `emailQTV` varchar(255) NOT NULL,
```

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

```
`matKhuQTV` varchar(255) NOT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_vietnamese_ci;
```

Tạo khóa chính sau khi tạo bảng:

```
ALTER TABLE `quantrivien`  
ADD PRIMARY KEY (`maQuanTriVien`);
```

Khóa ngoại (Foreign Key): Một cột hoặc nhóm cột liên kết với khóa chính của một bảng khác, dùng để duy trì tính toàn vẹn dữ liệu.

Tạo khóa ngoại ngay khi tạo bảng:

```
CREATE TABLE `phananh` (  
  `maPhanAnh` int(11) NOT NULL AUTO_INCREMENT,  
  `maSinhVien` int(11) NOT NULL,  
  `maPhong` int(11) NOT NULL  
  `tieuDeCH` varchar(255) NOT NULL,  
  `noiDungCH` text DEFAULT NULL,  
  `ngayDatCH` datetime NOT NULL DEFAULT  
current_timestamp(),  
  `trangThai` enum('cho_tiep_nhan', 'dang_xu_ly',  
'da_hoan_thanh', 'huy_bo') NOT NULL DEFAULT 'cho_tiep_nhan',  
  `luotTraLoi` int(11) NOT NULL  
  FOREIGN KEY (`maSinhVien`) REFERENCES `sinhvien`  
(`maSinhVien`),  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4  
COLLATE=utf8mb4_vietnamese_ci;
```

Tạo khóa ngoại sau khi tạo bảng:

```
ALTER TABLE `phananh`  
ADD CONSTRAINT `fk_phananh_phong`  
FOREIGN KEY (`maPhong`) REFERENCES `phong` (`maPhong`)  
ON UPDATE CASCADE ON DELETE RESTRICT;
```

Nhờ vào những đặc điểm và khả năng trên, MySQL trở thành công cụ không thể thiếu trong việc quản lý và xử lý dữ liệu trong các hệ thống phần mềm hiện đại.

2.4.5. Kết luận

MySQL là hệ quản trị cơ sở dữ liệu quan hệ ổn định và đáng tin cậy, hỗ trợ lưu trữ và truy vấn dữ liệu hiệu quả cho các ứng dụng web. Nhờ cấu trúc dữ liệu chặt chẽ, khả năng xử lý truy vấn mạnh mẽ và tính toàn vẹn cao, MySQL giúp đảm bảo dữ liệu được quản lý an toàn và nhất quán. Đây là nền tảng quan trọng giúp hệ thống vận hành bền vững và đáp ứng tốt các nhu cầu quản lý dữ liệu quy mô vừa và lớn.

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả vấn đề

Trong bối cảnh quy mô đào tạo của Trường Đại học Trà Vinh ngày càng mở rộng, số lượng sinh viên có nhu cầu nội trú tại ký túc xá tăng cao qua các năm. Tuy nhiên, công tác quản lý ký túc xá theo phương thức truyền thống (dựa trên sổ sách giấy tờ hoặc các file Excel rời rạc) đang bộc lộ nhiều hạn chế và bất cập. Sinh viên thường gặp khó khăn trong việc cập nhật thông tin về tình trạng phòng trống, giá phòng và phải tốn nhiều thời gian, công sức để trực tiếp đến văn phòng thực hiện các thủ tục đăng ký, xem hóa đơn điện nước hay báo cáo sự cố hư hỏng.

Về phía nhà trường, việc quản lý hồ sơ lưu trú, tính toán chi phí sinh hoạt và theo dõi cơ sở vật chất một cách thủ công dễ dẫn đến sai sót, thất lạc dữ liệu và khó khăn trong việc thống kê, báo cáo kịp thời.

Vì vậy, đề tài “Xây dựng Website quản lý phòng Ký túc xá cho sinh viên Trường Đại học Trà Vinh” được thực hiện nhằm cung cấp một giải pháp quản lý tập trung, hiện đại hóa quy trình đăng ký và quản lý nội trú. Hệ thống giúp sinh viên chủ động trong việc tìm kiếm chỗ ở, theo dõi chi phí sinh hoạt, đồng thời hỗ trợ Ban quản lý tối ưu hóa quy trình làm việc, đảm bảo tính minh bạch và nâng cao chất lượng phục vụ sinh viên.

3.2. Xác định yêu cầu người dùng

Trước khi xác định yêu cầu chi tiết, cần làm rõ các chức năng cốt lõi mà hệ thống quản lý ký túc xá phải đáp ứng. Mục tiêu của hệ thống là tạo ra một nền tảng tập trung giúp sinh viên dễ dàng thực hiện các thủ tục đăng ký nội trú, theo dõi sinh hoạt phí và tương tác với Ban quản lý một cách nhanh chóng, minh bạch. Các yêu cầu cơ bản của hệ thống bao gồm:

- Giao diện đơn giản, thân thiện: Hỗ trợ sinh viên truy cập dễ dàng trên cả máy tính và thiết bị di động để tra cứu thông tin phòng ở và các thông báo từ nhà trường.

- **Quản lý thông tin và đăng ký phòng:** Cho phép sinh viên xem sơ đồ tòa nhà, tình trạng phòng (trống/đầy), giá phòng và thực hiện đăng ký giữ chỗ trực tuyến thay vì làm thủ tục giấy.

- **Quản lý sự cố và phản ánh:** Hỗ trợ sinh viên gửi các yêu cầu sửa chữa cơ sở vật chất (điện, nước, giường tủ...) hoặc đóng góp ý kiến trực tiếp đến Ban quản lý.

- **Hệ thống thông báo:** Đảm bảo sinh viên nhận được các thông báo khẩn cấp hoặc định kỳ về quy định, lịch cúp điện/nước hoặc lịch nộp phí kịp thời.

3.3. Phân loại người dùng

Những người sử dụng hệ thống này gồm ba nhóm chính:

- **Sinh viên:** Người dùng đăng ký ở trọ ký túc xá ,thanh toán tiền ký túc xá, yêu cầu hỗ trợ từ quản lý ký túc xá.

- **Quản lý ký túc xá:** xét duyệt đơn yêu cầu hỗ trợ của người dùng.

- **Quản trị viên:** quản trị viên có thể Thống kê đăng ký theo tháng, quản lý đơn đăng ký, quản lý người dùng, quản lý phòng ở, quản lý thanh toán, quản lý yêu cầu hỗ trợ.

3.4. Phân tích thiết kế hệ thống

3.4.1. Đặc tả yêu cầu hệ thống

3.4.1.1. Yêu cầu chức năng

Để dự án này có thể hoạt động đúng với mục đích ban đầu thì cần phải có các chức năng:

- Đăng nhập.
- Đăng ký phòng ở.
- Chức năng thanh toán.
- Chức năng yêu cầu hỗ trợ
- Chức năng quản lý yêu cầu hỗ trợ

- Chức năng quản trị dành cho quản trị viên.

3.4.1.2. Yêu cầu phi chức năng

Để dự án này có thể hoạt động tốt và hiệu quả thì cần phải đảm bảo các yêu cầu sau:

- Ứng dụng hoạt động mượt mà, hiệu quả cao.
- Phải bảo mật dữ liệu của người dùng.
- Giao diện thân thiện và dễ sử dụng.
- Đảm bảo tính mở rộng và bảo trì.

3.4.2. Kiến trúc hệ thống

Để hệ thống web chia sẻ tài liệu và hỏi đáp hoạt động hiệu quả, dễ mở rộng và dễ bảo trì, đề tài áp dụng kiến trúc phân tầng (Layered Architecture) — mô hình đã được trình bày ở chương 2. Kiến trúc này giúp tách biệt rõ ràng giữa giao diện, xử lý nghiệp vụ và dữ liệu, đảm bảo sự linh hoạt khi phát triển và triển khai hệ thống. Hệ thống được chia thành ba tầng chính:

3.4.2.1. Lớp giao diện người dùng (Presentation Layer):

Lớp giao diện người dùng chịu trách nhiệm hiển thị dữ liệu và tương tác trực tiếp với người dùng. Ở đề tài này, giao diện được xây dựng bằng ReactJS, kết hợp HTML, CSS và JavaScript. Lớp này đảm nhiệm các chức năng:

- Nhận yêu cầu từ người dùng như đăng nhập, đăng ký, tra cứu, đánh giá hoặc tải tài liệu.
- Gửi yêu cầu đến backend thông qua API.
- Hiển thị dữ liệu nhận được từ backend theo bố cục dễ hiểu.
- Đảm bảo trải nghiệm người dùng mượt mà, trực quan.

3.4.2.2. Lớp xử lý nghiệp vụ (Business Logic Layer):

Lớp xử lý nghiệp vụ được xây dựng bằng NodeJS (ExpressJS) theo mô hình mô-đun. Đây là thành phần trung tâm thực hiện:

- Xử lý các yêu cầu từ frontend và điều phối tới các module tương ứng.
- Triển khai các chức năng chính như xác thực người dùng, xử lý đánh giá, tìm kiếm dữ liệu, quản lý tài liệu và câu hỏi.
- Kiểm tra dữ liệu đầu vào, đảm bảo tính hợp lệ và bảo mật.
- Tương tác với cơ sở dữ liệu thông qua các truy vấn MySQL.

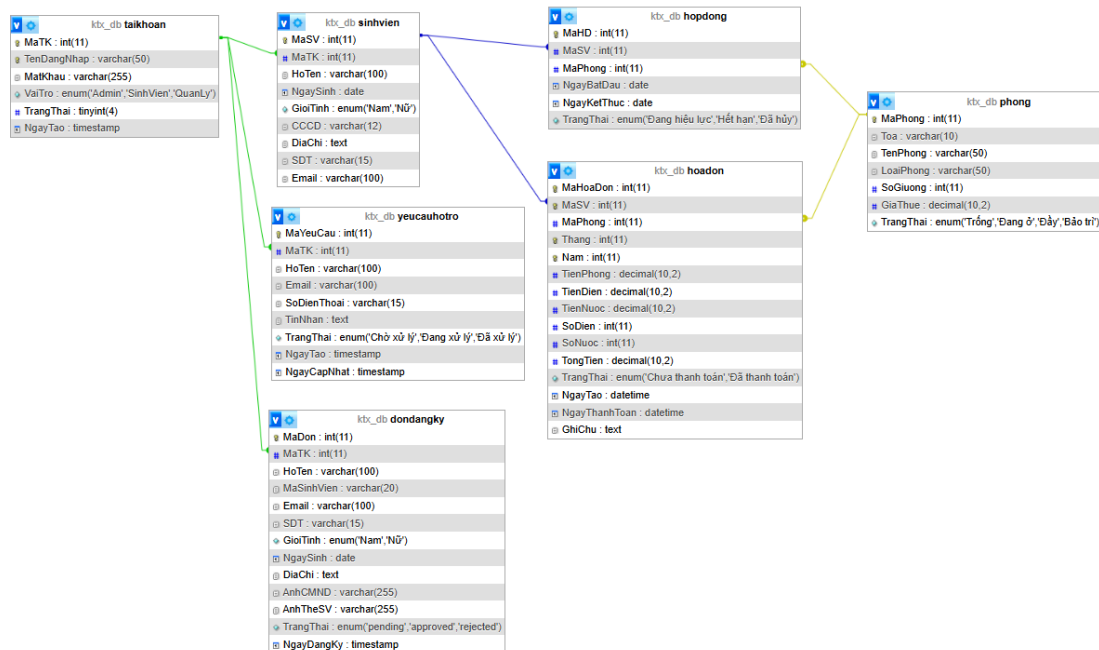
3.4.2.3. Lớp dữ liệu (Data Layer):

Lớp dữ liệu sử dụng MySQL để lưu trữ và quản lý toàn bộ thông tin của hệ thống. Tầng này cung cấp:

- Lưu trữ dữ liệu bền vững và có cấu trúc.
- Các thao tác CRUD (Create, Read, Update, Delete).
- Đảm bảo tính toàn vẹn dữ liệu.

3.4.3. Thiết kế dữ liệu

3.4.3.1. Mô hình ERD



Hình 3.1 Mô hình ERD

3.4.3.2. Danh sách các thực thể và mối kết hợp

Bảng 3.1 Bảng thực thể

STT	Tên thực thể	Diễn giải
1	dondangky	Lưu trữ thông tin đơn đăng ký ở trọ của người dùng
2	hoadon	Lưu trữ hóa đơn tiền điện, nước, ký túc xá của người dùng
3	hopdong	Lưu trữ các thông tin hợp đồng, ngày bắt đầu hợp đồng, ngày kết thúc
4	yeucauhotro	Lưu trữ các yêu cầu hỗ trợ của sinh viên
5	phong	Lưu trữ thông tin phòng
6	sinhvien	Lưu trữ thông tin sinh viên
7	taikhoan	Lưu trữ tất cả tài khoản

3.4.3.3. Chi tiết các thực thể

Tên thực thể: **dondangky**

Mô tả: Lưu trữ thông tin đơn đăng ký.

Chi tiết thực thể:

Bảng 3.2 Bảng thuộc tính của thực thể dondangky

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	MaDon	Mã đơn	int	PK AUTO_INCREMENT
2	MaTK	Mã tài khoản	int	FK
3	HoTen	Họ tên	varchar(100)	NOT NULL
4	MaSinhVien	Mã sinh viên	varchar(20)	NOT NULL
5	Email	Email	varchar(100)	NOT NULL
6	SDT	Sdt	varchar(15)	NOT NULL
7	GioiTinh	Giới tính	enum('Nam', 'Nữ')	NULL
8	NgaySinh	Ngày sinh	date	NULL
9	DiaChi	Địa chỉ	text	NULL

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

10	AnhCMND	Ảnh CMND	varchar(255)	NULL
11	AnhTheSV	Ảnh thẻ sinh viên	varchar(255)	NULL
12	TrangThai	Trạng thái	enum('pending', 'approved', 'rejected')	NULL
13	NgayDangKy		timestamp	NOT NULL

Tên thực thể: **hoadon**

Mô tả: Lưu trữ thông tin hóa đơn.

Chi tiết thực thể:

Bảng 3.3 Bảng thuộc tính của thực thể hoadon

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	MaHoaDon	Mã hóa đơn	int	PK AUTO_INCREMENT
2	MaSV	Mã sinh viên	int	FK
3	MaPhong	Mã phòng	int	FK
4	Thang	Tháng	int	FK
5	Nam	Năm	int	FK
6	TienPhong	Tiền phòng	decimal(10,2)	NULL
7	TienDien	Tiền điện	decimal(10,2)	NULL
8	TienNuoc	Tiền nước	decimal(10,2)	NULL
9	SoDien	Số điện	int	NULL
10	SoNuoc	Số nước	int	NULL
11	TongTien	Tổng tiền	decimal(10,2)	NOT NULL
12	TrangThai	Trạng thái	enum('Chưa thanh toán', 'Đã thanh toán')	NULL
13	NgayTao	Ngày tạo	datetime	NULL
14	NgayThanhToan	Ngày thanh toán	datetime	NULL
15	GhiChu	Ghi chú	text	NULL

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

Tên thực thể: **hopdong**

Mô tả: Lưu trữ hợp đồng.

Chi tiết thực thể:

Bảng 3.4 Bảng thuộc tính của thực thể hopdong

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	MaHD	Mã hợp đồng	int	PK AUTO_INCREMENT
2	MaSV	Mã sinh viên	int	FK
3	MaPhong	Mã phòng	int	FK
4	NgayBatDau	Ngày bắt đầu	date	NULL
5	NgayKetThuc	Ngày kết thúc	date	NULL
6	TrangThai	Trạng thái	enum('Đang hiệu lực', 'Hết hạn', 'Đã hủy')	NULL

Tên thực thể: **phong**

Mô tả: Lưu trữ danh sách các môn học thuộc từng ngành.

Chi tiết thực thể:

Bảng 3.5 Bảng thuộc tính của thực thể phong

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	MaPhong	Mã phòng	int	PK AUTO_INCREMENT
2	Toa	Tòa	varchar(10)	NULL
3	TenPhong	Tên phòng	varchar(50)	NOT NULL
4	LoaiPhong	Loại phòng	varchar(50)	NULL
5	SoGiuong	Số giường	int	NULL
6	GiaThue	Giá thuê	decimal(10,2)	NULL
7	TrangThai	Trạng thái	enum('Trống', 'Đang ở', 'Đầy', 'Bảo trì')	NULL

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

Tên thực thể: **sinhvien**

Mô tả: Lưu trữ danh sách loại tài liệu học tập.

Chi tiết thực thể:

Bảng 3.6 Bảng thuộc tính của thực thể sinhvien

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	MaSV	Mã sinh viên	int	PK AUTO_INCREMENT
2	MaTK	Mã tài khoản	int	FK
3	HoTen	Họ tên	varchar(100)	NOT NULL
4	NgaySinh	Ngày sinh	date	NULL
5	GioiTinh	Giới tính	enum('Nam', 'Nữ')	NULL
6	CCCD	Căn cước công dân	varchar(12)	NULL
7	DiaChi	Địa chỉ	text	NULL
8	SĐT	Số điện thoại	varchar(15)	NULL
9	Email	Email	varchar(100)	NULL

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

Tên thực thể: **taikhoan**

Mô tả: Lưu trữ thông tin chi tiết của tài liệu học tập do sinh viên chia sẻ.

Chi tiết thực thể:

Bảng 3.7 Bảng thuộc tính của thực thể taikhoan

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	MaTK	Mã tài khoản	int	PK AUTO_INCREMENT
2	TenDangNhap	Tên đăng nhập	varchar(50)	FK
3	MatKhau	Mật khẩu	varchar(255)	NOT NULL
4	VaiTro	Vai trò	enum('Admin', 'SinhVien', 'QuanLy')	NULL
5	TrangThai	Trạng thái	tinyint(4)	NULL
6	NgayTao	Ngày tạo	timestamp	NOT NULL

Tên thực thể: **yeucauhotro**

Mô tả: Lưu trữ danh sách các định dạng tệp tài liệu.

Chi tiết thực thể:

Bảng 3.8 Bảng thuộc tính của thực thể yeucauhotro

STT	Thuộc tính	Diễn giải	Kiểu dữ liệu	Ràng buộc
1	MaYeuCau	Mã yêu cầu	int	PK AUTO_INCREMENT
2	MaTK	Mã tài khoản	int	FK
3	HoTen	Họ tên	varchar(100)	NOT NULL
4	Email	Email	varchar(100)	NOT NULL
5	SoDienThoai	Số điện thoại	varchar(15)	NULL
6	TinNhan	Tin nhắn	text	NOT NULL
7	TrangThai	Trạng thái	enum('Chờ xử lý', 'Đang xử lý', 'Đã xử lý')	NULL

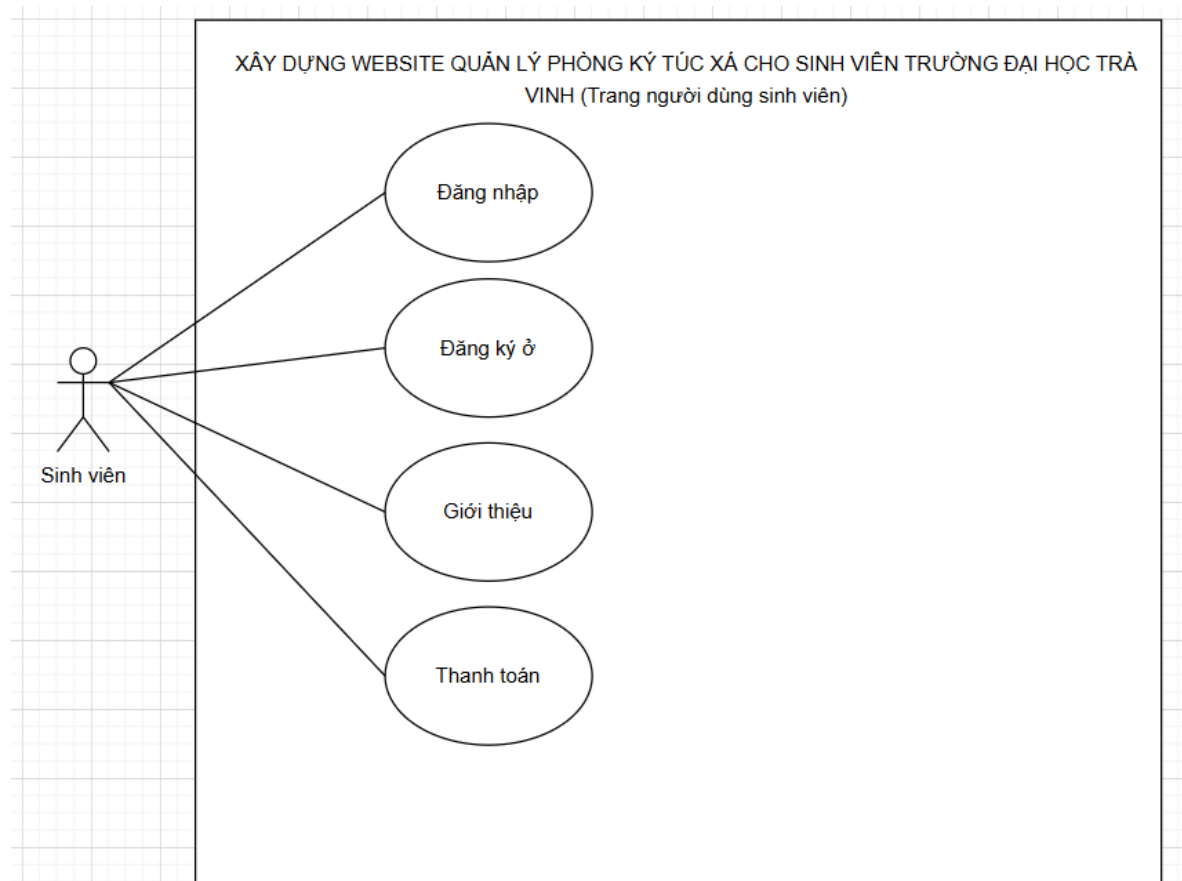
Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

8	NgàyTao	Ngày tạo	timestamp	NOT NULL
9	NgàyCapNhat	Ngày cập nhật	timestamp	NOT NULL

3.4.4. Thiết kế giao diện

3.4.4.1. Sơ đồ Use Case

3.4.4.1.1. Trang người dùng

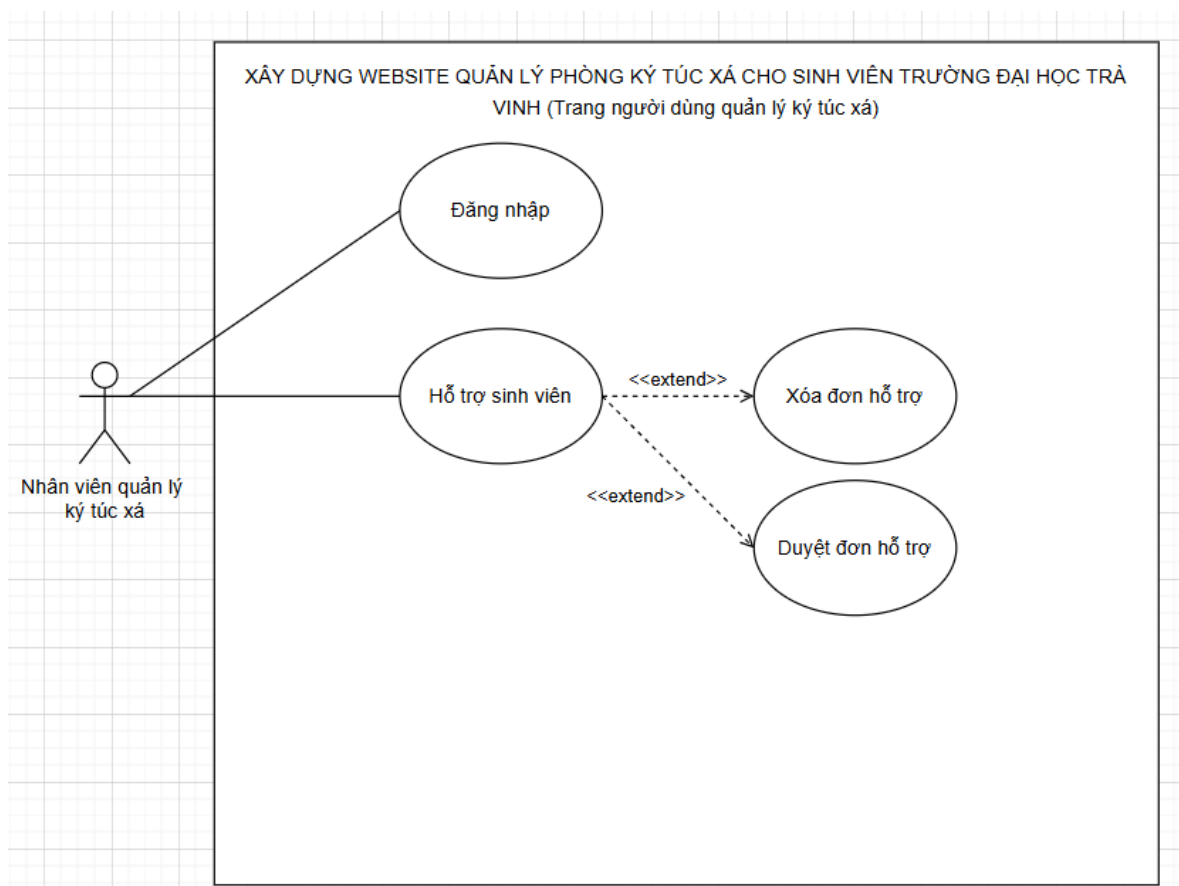


Hình 3.2 Sơ đồ Use Case về trang người dùng

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

Tên Use Case	Tác nhân	Mô tả
Đăng nhập	Sinh viên	Sinh viên đăng nhập để sử dụng để đăng ký ở
Đăng ký ở		Sinh viên sau khi đăng nhập có thể đăng ký ở
Giới thiệu		Sinh viên có thể xem trang giới thiệu về ký túc xá trường
Thanh toán		Sinh viên thanh toán tiền ký túc xá

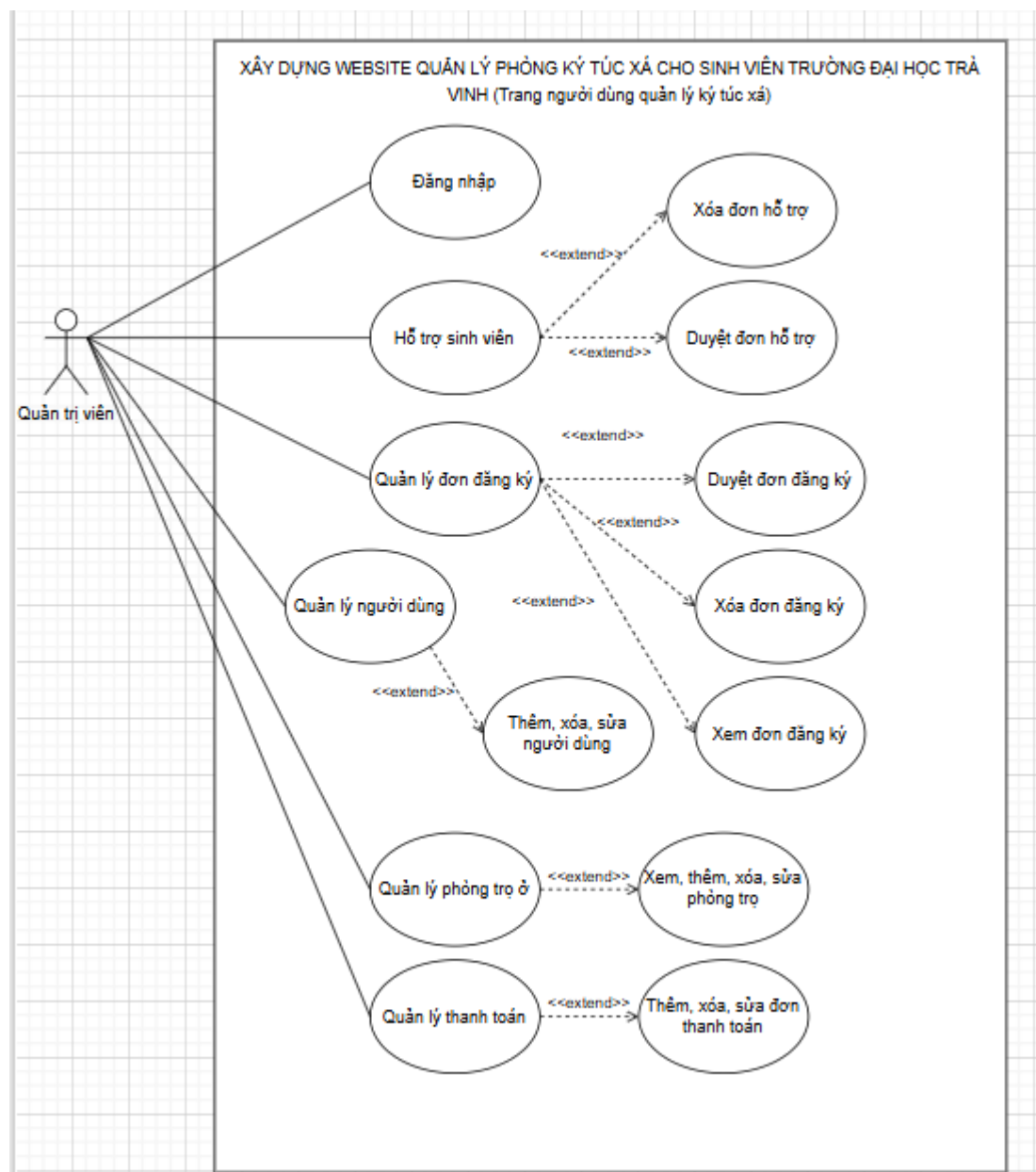
3.4.4.1.2. Trang quản trị



Hình 3.3 Sơ đồ Use Case về trang nhân viên quản lý ký túc xá

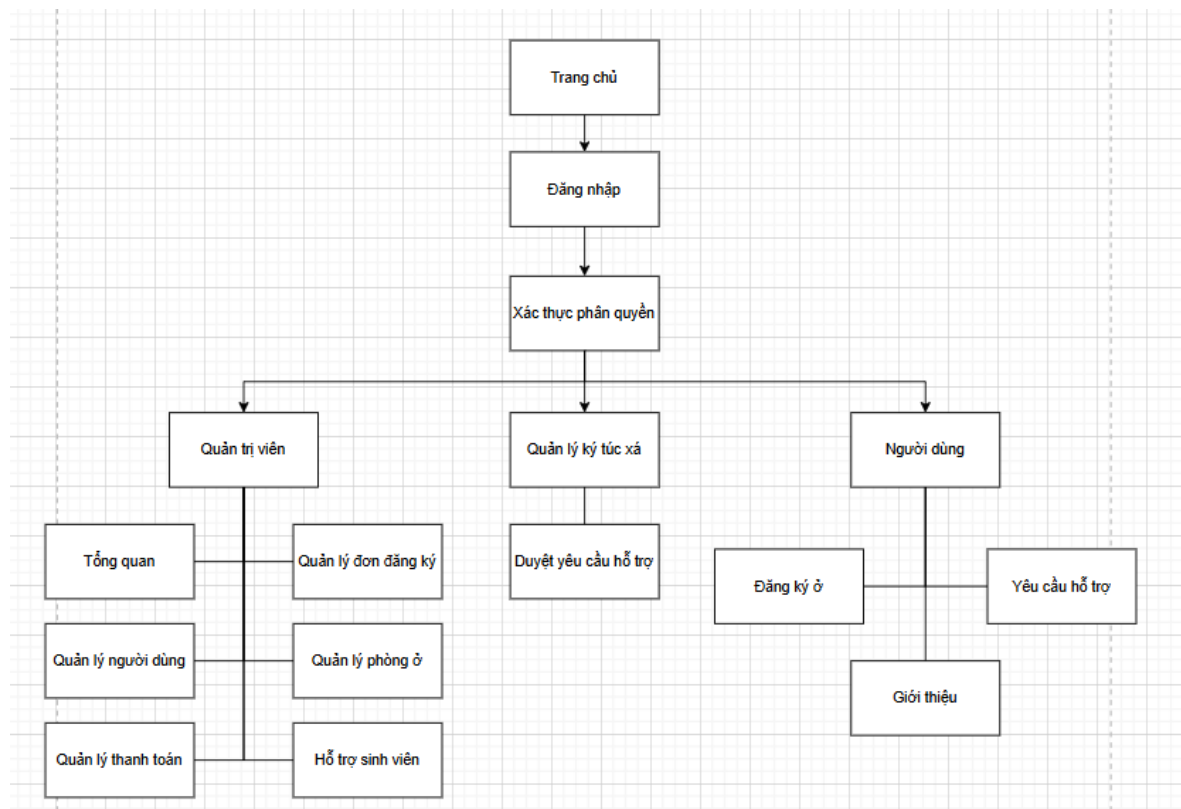
Tên Use Case	Tác nhân	Mô tả
Đăng nhập	Nhân viên quản lý ký túc xá	Nhân viên quản lý ký túc xá
Hỗ trợ sinh viên		Nhân viên quản lý ký túc xá duyệt đơn yêu cầu hỗ trợ của sinh viên

3.4.4.1.3. Trang quản trị



Tên Use Case	Tác nhân	Mô tả
Đăng nhập	Quản trị viên	Quản trị viên đăng nhập để sử dụng các chức năng
Hỗ trợ sinh viên		Quản trị viên duyệt đơn yêu cầu hỗ trợ của sinh viên
Quản lý đơn đăng ký		Quản trị viên xem, xóa, duyệt, từ chối đơn đăng ký
Quản lý người dùng		Quản trị viên có thể thêm, xóa, sửa người dùng
Quản lý phòng trọ ở		Quản trị viên có thể thêm, xóa, sửa phòng trọ
Quản lý thanh toán		Quản trị viên có thể thêm đơn thành toán tiền ký túc xá cho sinh viên



















3.4.4.2. Sơ đồ website



Hình 3.4 Sơ đồ website

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Dữ liệu thử nghiệm

		▼ MaDon	MaTK	HoTen	MaSinhVien	Email	SDT	GioiTinh	NgaySinh	DiaChi	AnhCMND	AnhTheSV	TrangThai	NgayDangKy
<input type="checkbox"/>	  	4	NULL	User1	01	user1@gmail.com	1234567890	Nam	0000-00-00	Trà Vinh	NULL	NULL	approved	2025-11-25 19:01:30
<input type="checkbox"/>	  	6	NULL	User3	03	user3@gmail.com	1234567890	Nam	0000-00-00	Trà Vinh	/uploads/ldCardFront-1764073091972-179430279.PNG	/uploads/studentCard-1764073091973-555334385.PNG	approved	2025-11-25 19:18:11
<input type="checkbox"/>	  	7	NULL	User2	02	user2@gmail.com	1234567890	Nữ	0000-00-00	Trà Vinh	/uploads/ldCardFront-1764084384636-571321620.PNG	/uploads/studentCard-1764084384639-923449784.jpg	approved	2025-11-25 22:26:24
<input type="checkbox"/>	  	8	NULL	User4	04	user4@gmail.com	1234567890	Nam	0000-00-00	Trà Vinh	/uploads/ldCardFront-1764084546636-448761350.PNG	/uploads/studentCard-1764084546638-775285166.PNG	approved	2025-11-25 22:29:06
<input type="checkbox"/>	  	11	NULL	User5	05	user5@gmail.com	1234567890	Nam	0000-00-00	Trà Vinh	/uploads/ldCardFront-1764204367752-804736015.PNG	/uploads/studentCard-1764204367752-306152010.PNG	approved	2025-11-27 07:46:07
<input type="checkbox"/>	  	12	NULL	User5	123	user5@gmail.com	1234567890	Nam	0000-00-00	Trà Vinh	/uploads/ldCardFront-1764224117645-546497082.PNG	/uploads/studentCard-1764224117646-865552779.png	pending	2025-11-27 13:15:17

Hình 4.1 Dữ liệu của bảng dondangky

		▼	MaHoaDon	MaSV	MaPhong	Thang	Nam	TienPhong	TienDien	TienNuoc	SoDien	SoNuoc	TongTien	TrangThai	NgayTao	NgayThanhToan	GhiChu	
<input type="checkbox"/>				5	1	9	12	2025	1500000.00	0.00	0.00	0	0	1500000.00	Đã thanh toán	2025-12-11 18:38:14	2025-12-11 18:53:58	
<input type="checkbox"/>				6	1	9	1	2025	1500000.00	63000.00	75000.00	21	5	1638000.00	Chưa thanh toán	2025-12-11 18:46:46	NULL	test

Hình 4.2 Dữ liệu của bảng hoadon

				MaHD	MaSV	MaPhong	NgayBatDau	NgayKetThuc	TrangThai
<input type="checkbox"/>				20	3	9	2025-11-25	2026-11-25	Đang hiệu lực
<input type="checkbox"/>				25	2	9	2025-11-25	2026-11-25	Đang hiệu lực
<input type="checkbox"/>				27	4	9	2025-11-25	2026-11-25	Đang hiệu lực
<input type="checkbox"/>				35	1	9	2025-11-27	2026-11-27	Đang hiệu lực

Hình 4.3 Dữ liệu của bảng hopdong

				MaPhong	Toa	TenPhong	LoaiPhong	SoGiuong	GiaThue	TrangThai
<input type="checkbox"/>				9	A	692	4 người	4	1500000.00	Đầy
<input type="checkbox"/>				10	B	201	4 người	4	1500000.00	Trống
<input type="checkbox"/>				12	C	111	4 người	4	1500000.00	Trống

Hình 4.4 Dữ liệu của bảng phong

				MaSV	MaTK	HoTen	NgaySinh	GioiTinh	CCCD	DiaChi	SDT	Email
<input type="checkbox"/>				1	2	user1	NULL	NULL	NULL	NULL	NULL	user1@gmail.com
<input type="checkbox"/>				2	3	user2	NULL	NULL	NULL	NULL	NULL	user2@gmail.com
<input type="checkbox"/>				3	4	user3	NULL	NULL	NULL	NULL	NULL	user3@gmail.com
<input type="checkbox"/>				4	5	user4	NULL	NULL	NULL	NULL	NULL	user4@gmail.com
<input type="checkbox"/>				6	8	user5	NULL	NULL	NULL	NULL	NULL	user5@gmail.com

Hình 4.5 Dữ liệu của bảng sinhvien

Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh

		MaTK	TenDangNhap	MatKhau	VaiTro	TrangThai	NgayTao
<input type="checkbox"/>	Edit Copy Delete	1	admin	\$2b\$10\$5lEgMp1oznOQlmdHdtM6RuS/T/G4Ougt.kl4Wjam7O2...	Admin	1	2025-11-25 15:07:33
<input type="checkbox"/>	Edit Copy Delete	2	user1	\$2b\$10\$jtVjJ.Tn0n9q7QzBV4hTe6ceDnoIA5.LKnEq0PVhtp...	SinhVien	1	2025-11-25 15:49:44
<input type="checkbox"/>	Edit Copy Delete	3	user2	\$2b\$10\$enqLcySRaqHyvM6cEZRBOfld93ahmpQw20mJC5jYlq...	SinhVien	1	2025-11-25 16:07:36
<input type="checkbox"/>	Edit Copy Delete	4	user3	\$2b\$10\$IAHGyQvRpoing6aUUI8hj.jnmk0eBx3vftoHx7XCQDL...	SinhVien	1	2025-11-25 18:41:13
<input type="checkbox"/>	Edit Copy Delete	5	user4	\$2b\$10\$fduxRC/38wW9rl.GF3UjuZNM.2xb25aLtMEwIDYmZM...	SinhVien	1	2025-11-25 22:27:57
<input type="checkbox"/>	Edit Copy Delete	7	quanly1	\$2b\$10\$VJlomT3ucUDm1S2SvNL7G.S3J0Ed5d4BV3dkyNUusdW...	QuanLy	1	2025-11-26 21:46:44
<input type="checkbox"/>	Edit Copy Delete	8	user5	\$2b\$10\$j12zANrIZY8NDevj.g5eMOAFITzDvJ8TexzNvHmM9qw...	SinhVien	1	2025-11-26 23:08:06
<input type="checkbox"/>	Edit Copy Delete	9	nhanvien	\$2b\$10\$NbtFssMe6nx1nf6Tordai.qWNZz14PeUdw7X/ybCfWT...	QuanLy	1	2025-11-27 12:18:43
<input type="checkbox"/>	Edit Copy Delete	10	nhanvien2	\$2b\$10\$VsJ1ZXAbRLsC2r4H5Ej9mub4fG5ZkbQ3L34/87pS8E/...	QuanLy	1	2025-11-27 12:27:38
<input type="checkbox"/>	Edit Copy Delete	11	nhanvien3	\$2b\$10\$0AZbz1VOSPk1w8yWfV5TIO1EGxD/xM4l1nDvp0spO3K...	QuanLy	1	2025-11-27 12:30:35

Hình 4.6 Dữ liệu của bảng taikhoan

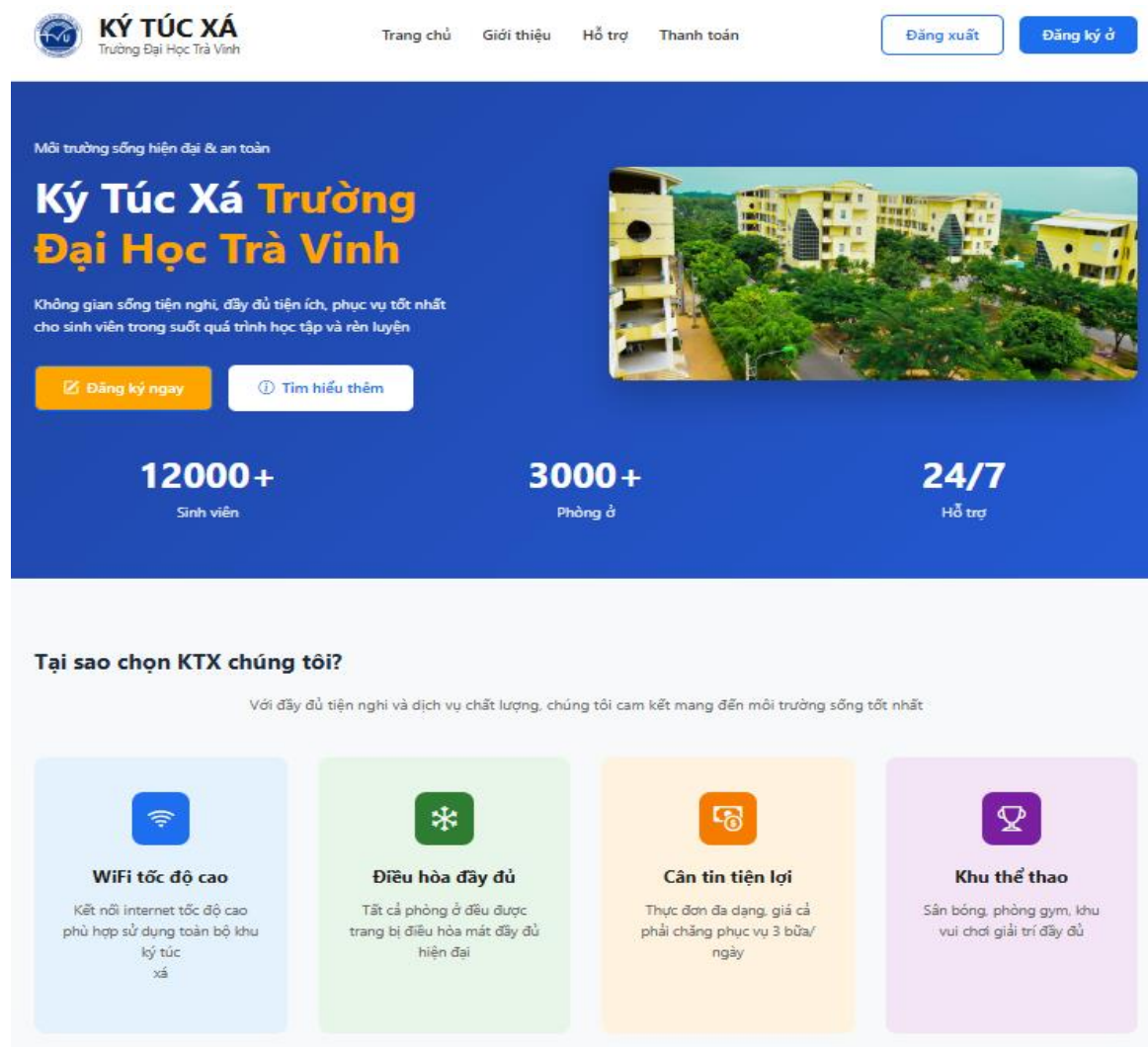
		MaYeuCau	MaTK	HoTen	Email	SoDienThoai	TinNhan	TrangThai	NgayTao	NgayCapNhat
<input type="checkbox"/>	Edit Copy Delete	6	NULL	User1	user1@gmail.com	1234567890	cửa bị hư	Chờ xử lý	2025-11-26 22:19:47	2025-11-26 22:22:47
<input type="checkbox"/>	Edit Copy Delete	7	NULL	User5	user5@gmail.com	1234567890	Cửa sổ phòng B21.201 bị hư	Chờ xử lý	2025-11-27 12:36:42	2025-11-27 12:36:42
<input type="checkbox"/>	Edit Copy Delete	8	NULL	User5	user5@gmail.com	1234567890	Vòi nước bị hư	Đã xử lý	2025-11-27 12:40:14	2025-12-02 09:48:03
<input type="checkbox"/>	Edit Copy Delete	12	NULL	User5	user5@gmail.com	1234567890	test	Đang xử lý	2025-11-27 13:13:19	2025-12-02 09:52:08

Hình 4.7 Dữ liệu của bảng yeucauhotro

4.2. Kết quả thực nghiệm

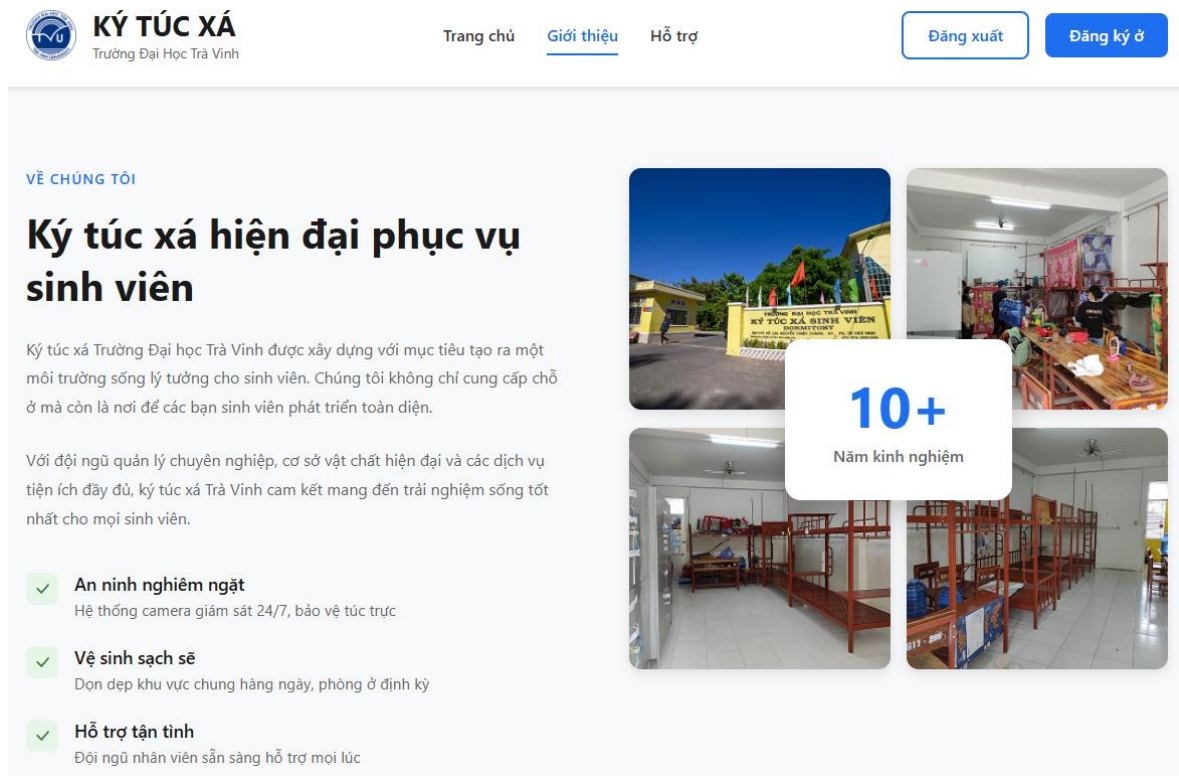
4.2.1. Chức năng trang người dùng sinh viên

4.2.1.1. Chức năng trang chủ




Hình 4.8 Giao diện trang chủ

4.2.1.2. Chức năng xem giới thiệu ký túc xá




Hình 4.9 Giao diện giới thiệu ký túc xá


4.2.1.3. Chức năng hỗ trợ


**KÝ TÚC XÁ**
Trường Đại Học Trà Vinh

Trang chủGiới thiệuHỗ trợ

Đăng xuấtĐăng ký ở

**Địa chỉ**
126 Nguyễn Thiện Thành, Khóm 4, Phường 5, TP. Trà Vinh

**Điện thoại**
(0294) 3855 246

**Email**
ktx@tvu.edu.vn

Liên hệ với chúng tôi

Có thắc mắc hay cần hỗ trợ? Đội ngũ của chúng tôi luôn sẵn sàng giúp đỡ bạn.

Liên hệ hỗ trợ

Họ và tên

Nhập họ và tên

Email

Email

Số điện thoại

Số điện thoại


Tin nhắn

Tin nhắn

Gửi tin nhắn

Hình 4.10 Giao diện hỗ trợ

4.2.1.4. Chức năng thanh toán hóa đơn

**KÝ TÚC XÁ**
Sinh viên

Trang chủThanh toán

Đăng xuất

Thanh toán hóa đơn

1

Hóa đơn chưa thanh toán

1.638.000 đ

1

Hóa đơn đã thanh toán

1.500.000 đ

2

Tổng hóa đơn

3.138.000 đ

Hóa đơn chưa thanh toán

Tháng 1/2025

Chưa thanh toán

Phòng A692

Tiền phòng: 1.500.000 đ

Tiền điện (21 kWh): 63.000 đ

Tiền nước (5 m³): 75.000 đ

Tổng cộng: 1.638.000 đ

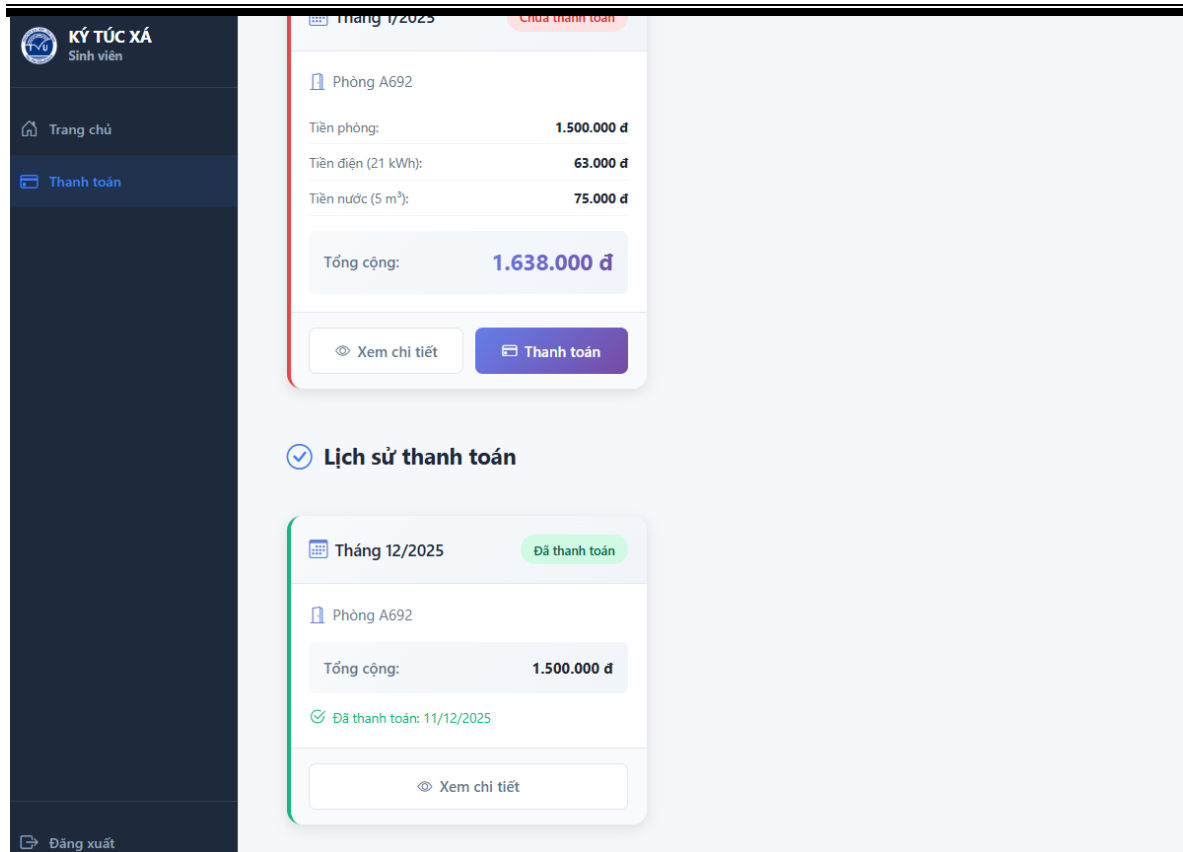
Xem chi tiếtThanh toán

Lịch sử thanh toán

Đỗ Gia Hào

50

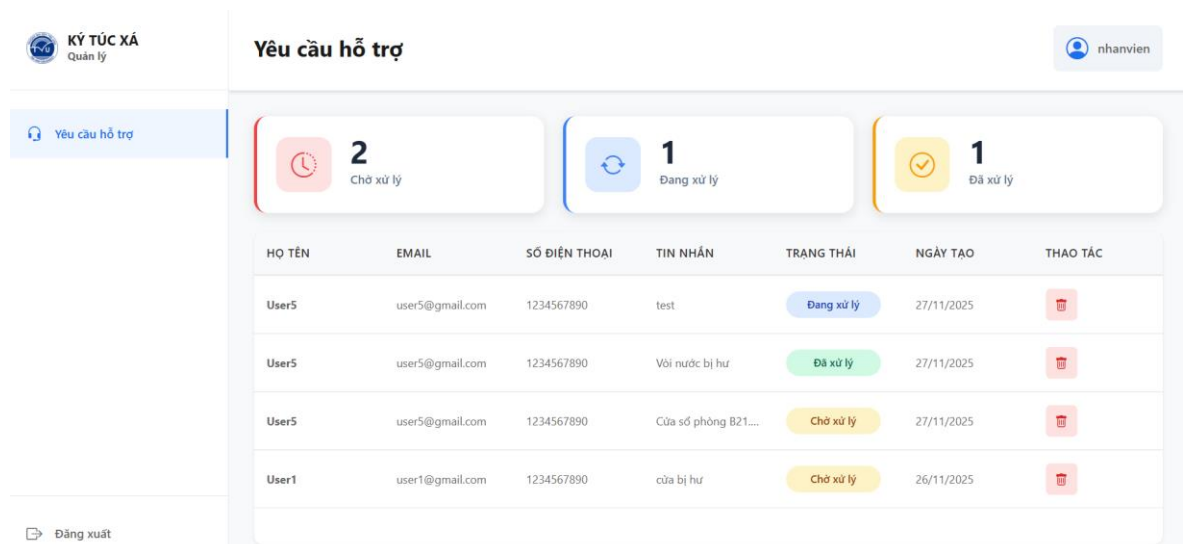
Xây dựng website quản lý phòng ký túc xá cho sinh viên trường đại học Trà Vinh



Hình 4.11 Giao diện thanh toán

4.2.2. Chức năng trang quản lý ký túc xá

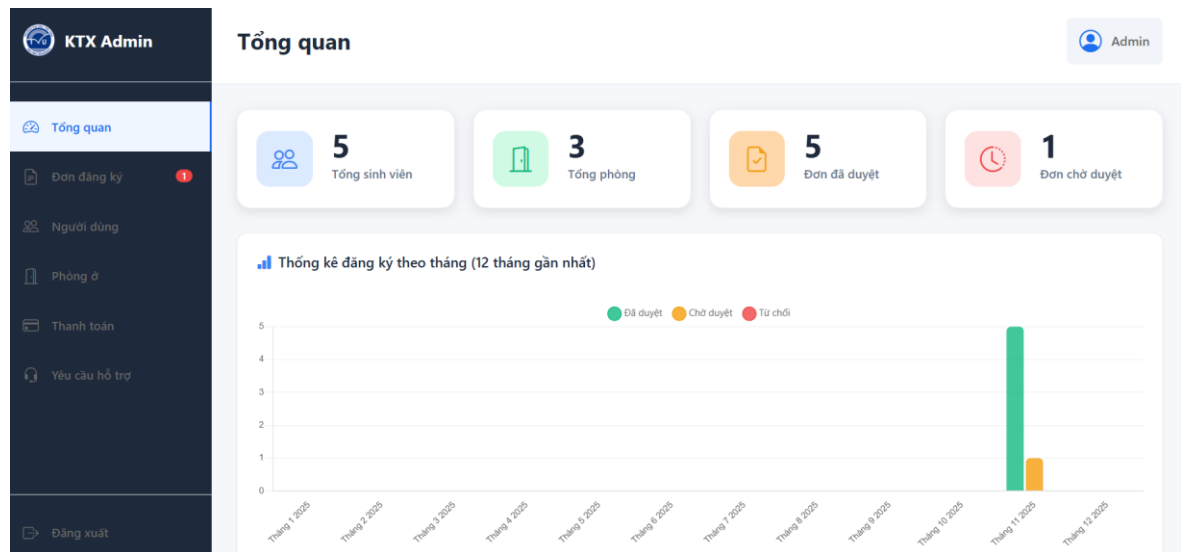
4.2.2.1. Chức năng hỗ trợ sinh viên



Hình 4.12 Giao diện chức năng hỗ trợ sinh viên

4.2.3. Trang quản trị viên

4.2.3.1. Chức năng tổng quan



Hình 4.13 Giao diện chức năng tổng quát

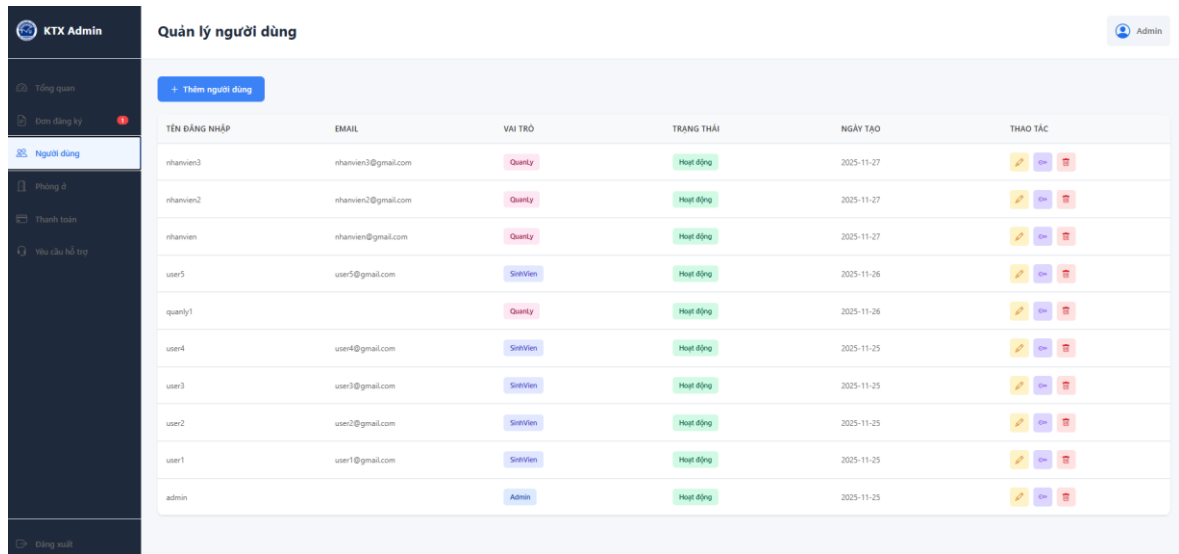
4.2.3.2. Chức năng quản lý đơn đăng ký

The screenshot shows the 'Quản lý đơn đăng ký' (Manage Registration) page of the KTX Admin system. The left sidebar contains navigation links: Tổng quan, Đơn đăng ký, Người dùng, Phòng ở, Thanh toán, Yêu cầu hỗ trợ, and Đăng xuất. The main content area displays three summary cards: Đơn đã duyệt (5), Đơn chờ duyệt (1), and Đơn từ chối (0). Below these is a table with columns: MÃ SV, HỌ VÀ TÊN, EMAIL, SỐ ĐIỆN THOẠI, NGÀY ĐĂNG KÝ, TRẠNG THÁI, and THAO TÁC. The table lists registration details for six users, including their status (Chờ duyệt, Đã duyệt) and action buttons (view, edit, delete).

MÃ SV	HỌ VÀ TÊN	EMAIL	SỐ ĐIỆN THOẠI	NGÀY ĐĂNG KÝ	TRẠNG THÁI	THAO TÁC
123	User5	user5@gmail.com	1234567890	2025-11-27	Chờ duyệt	✓ ✗ 👁 🗑
05	User5	user5@gmail.com	1234567890	2025-11-27	Đã duyệt	👁 🗑
04	User4	user4@gmail.com	1234567890	2025-11-25	Đã duyệt	👁 🗑
02	User2	user2@gmail.com	1234567890	2025-11-25	Đã duyệt	👁 🗑
03	User3	user3@gmail.com	1234567890	2025-11-25	Đã duyệt	👁 🗑
01	User1	user1@gmail.com	1234567890	2025-11-25	Đã duyệt	👁 🗑

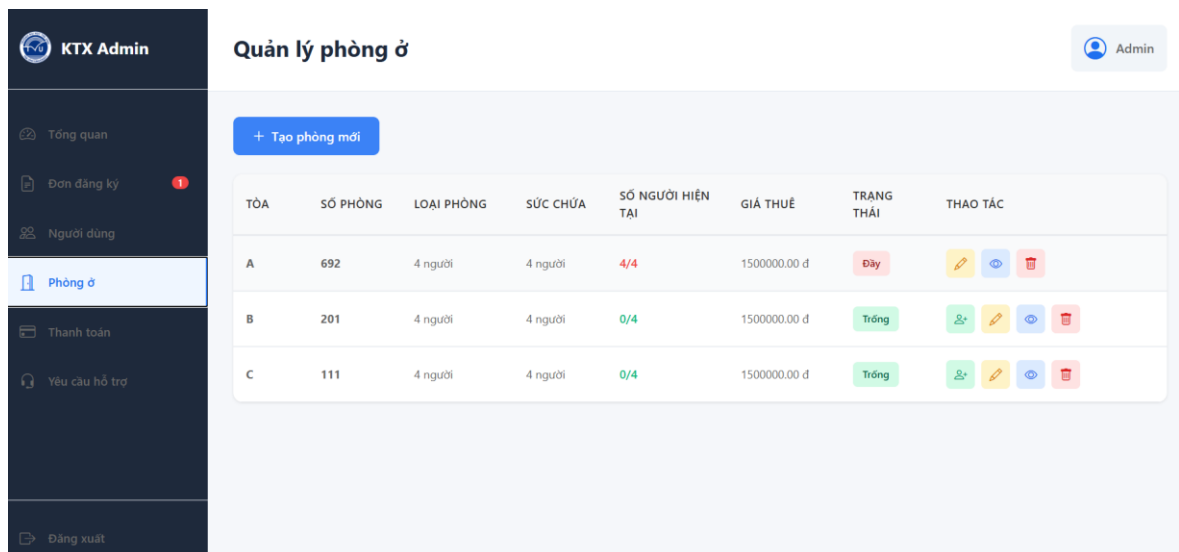
Hình 4.14 Giao diện chức năng quản lý đơn đăng ký

4.2.3.3. Chức năng quản lý người dùng



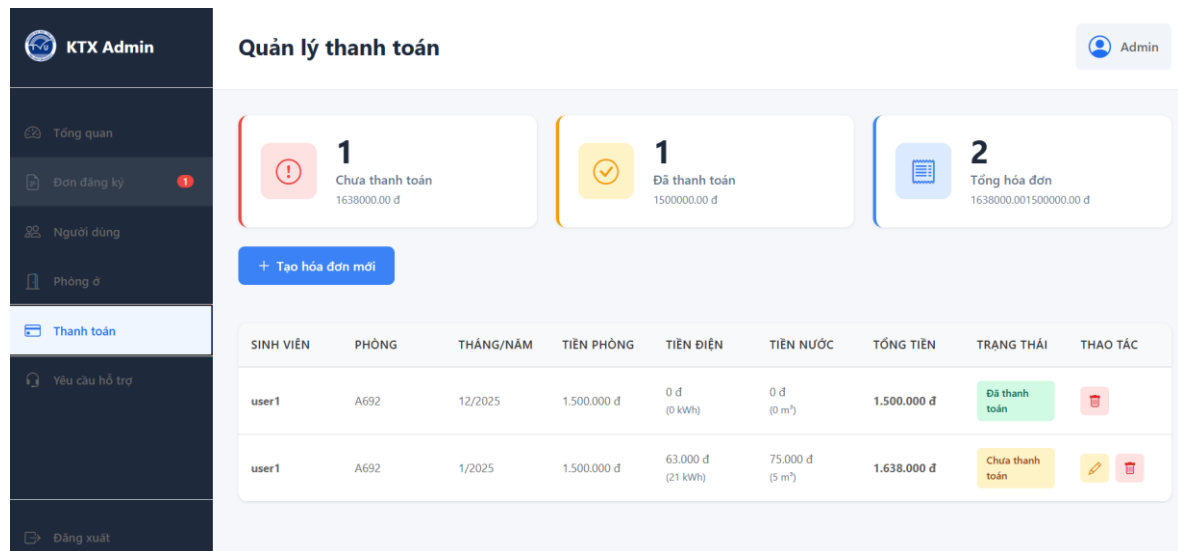
Hình 4.15 Giao diện chức năng quản lý người dùng

4.2.3.4. Chức năng quản lý phòng ở



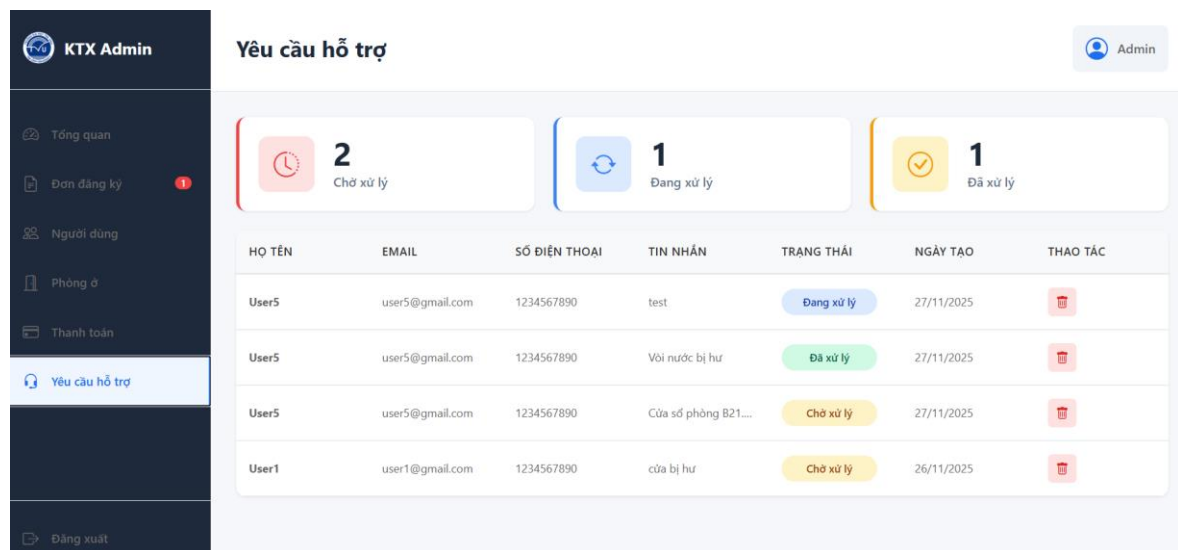
Hình 4.16 Giao diện chức năng quản lý phòng ở

4.2.3.5. Chức năng quản lý thanh toán



Hình 4.17 Giao diện chức năng quản lý thanh toán

4.2.3.6. Chức năng yêu cầu hỗ trợ



Hình 4.18 Giao diện chức năng yêu cầu hỗ trợ

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Sau quá trình nghiên cứu, phân tích yêu cầu, thiết kế và thi công hệ thống, đề tài "Xây dựng Website quản lý phòng Ký túc xá cho sinh viên Trường Đại học Trà Vinh" đã hoàn thành và đạt được những kết quả cụ thể như sau:

Hệ thống hóa được quy trình quản lý ký túc xá tại Trường Đại học Trà Vinh, từ khâu đăng ký nội trú, xét duyệt, quản lý sinh hoạt đến khi sinh viên làm thủ tục trả phòng.

Nắm vững và vận dụng được các công nghệ lập trình web hiện đại.

5.2. Hạn chế

Mặc dù đã nỗ lực hoàn thiện, tuy nhiên do giới hạn về mặt thời gian và kiến thức, đề tài vẫn còn tồn tại một số hạn chế nhất định:

Về tính năng thanh toán: Hệ thống hiện tại mới chỉ dừng lại ở việc thông báo hóa đơn và xác nhận thanh toán thủ công chưa tích hợp được cổng thanh toán trực tuyến tự động

Về bảo mật: Các cơ chế bảo mật mới chỉ ở mức cơ bản, chưa áp dụng các công nghệ bảo mật nâng cao như xác thực 2 yếu tố (2FA) hay mã hóa dữ liệu mức cao.

5.3. Hướng phát triển

Dựa trên những hạn chế đã nêu và nhu cầu thực tế của mô hình "Đại học thông minh" (Smart University) tại Đại học Trà Vinh, hướng phát triển tiếp theo của đề tài bao gồm:

Tích hợp cổng thanh toán trực tuyến: Kết nối API với các ví điện tử hoặc cổng thanh toán ngân hàng để sinh viên có thể quét mã QR thanh toán tiền phòng, tiền điện nước tự động, hệ thống sẽ tự động cập nhật trạng thái "Đã đóng tiền" mà không cần admin xác nhận.

Bảo mật: nhận thức được tầm quan trọng của việc bảo vệ dữ liệu cá nhân sinh viên và dữ liệu tài chính (hóa đơn điện nước, phí nội trú), trong tương lai hệ thống cần được nâng cấp các cơ chế bảo mật.

DANH MỤC TÀI LIỆU THAM KHẢO

Wikipedia, “HTML,” 15 11 2024. [Trực tuyến]. Available:
1] <https://vi.wikipedia.org/wiki/HTML>. [Đã truy cập 1 1 2025].

Wikipedia, “CSS,” 31 12 2024. [Trực tuyến]. Available:
2] <https://vi.wikipedia.org/wiki/CSS>. [Đã truy cập 1 1 2025].

Wikipedia, “JavaScript,” 30 11 2024. [Trực tuyến]. Available:
3] <https://vi.wikipedia.org/wiki/JavaScript>. [Đã truy cập 1 1 2025].

Wikipedia, “PHP,” 30 11 2024. [Trực tuyến]. Available:
4] <https://vi.wikipedia.org/wiki/PHP>. [Đã truy cập 1 1 2025].

Wikipedia, “MySQL,” 12 12 2023. [Trực tuyến]. Available:
5] <https://vi.wikipedia.org/wiki/MySQL>. [Đã truy cập 1 1 2025].