

[illegible]

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

[illegible]

Thành viên hội đồng
(Ký tên và ghi rõ họ tên)

LỜI CẢM ƠN

Tôi xin gửi lời cảm ơn đến quý thầy cô Trường Đại học Trà Vinh, đặc biệt là các thầy cô thuộc Bộ môn trong khoa Kỹ thuật và Công nghệ, đã tạo điều kiện thuận lợi để tôi có cơ hội thực hành và tiếp cận với thực tế, giúp tôi có thêm kinh nghiệm trong quá trình học tập và làm việc, giảm bớt những khó khăn và bỡ ngỡ trong quá trình chuẩn bị cho môi trường làm việc sau này.

Tôi cũng trân trọng cảm ơn thầy Phạm Minh Dương, người đã tận tình hướng dẫn và hỗ trợ tôi trong suốt quá trình thực hiện đồ án. Sự đồng hành và chỉ dẫn của Thầy đã giúp tôi hoàn thành công việc đúng thời hạn, tích lũy thêm nhiều kiến thức cần thiết cho bản thân.

Do hạn chế về thời gian cũng như kiến thức chuyên môn, nội dung đề tài của tôi vẫn còn những điểm cần cải thiện. Tôi rất mong nhận được sự đóng góp ý kiến từ các thầy cô để hoàn thiện và tốt hơn trong tương lai.

Tôi xin chân thành cảm ơn.

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN	12
CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT.....	13
2.1 Javascript	13
2.1.1 Sơ lược về ngôn ngữ Javascript	13
2.1.2 Đặc điểm	13
2.1.3 Cách hoạt động	13
2.2 React framework.....	14
2.2.1 Giới thiệu	14
2.2.2 Đặc điểm nổi bật	14
2.2.3 Các thành phần quan trọng	15
2.3 NodeJS	16
2.3.1 Giới thiệu	16
2.3.2 Đặc điểm vượt trội	17
2.3.3 Cơ chế hoạt động	17
2.4 Express.js framework.....	19
2.4.1 Giới thiệu.....	19
2.4.2 Các tính năng chính.....	19
2.5 Mô hình MVC.....	21
2.5.1 Tổng quan về mô hình MVC.....	21
2.5.2 Các thành phần	22
2.5.3 Ưu điểm và Nhược điểm	23
2.6 Tailwind CSS	24
2.6.1 Tổng quan về Tailwind CSS	24
2.6.2 Tổng quan về Tailwind CSS	24
2.7 Hệ quản trị cơ sở dữ liệu MongoDB	26
2.7.1 Giới thiệu.....	26
2.7.2 Cách hoạt động	26

2.7.3	Ưu điểm và Nhược điểm	27
2.8	Thanh toán điện tử	28
2.8.1	Tổng quan.....	28
2.8.2	Lợi ích của thanh toán điện tử.....	29
2.8.3	Các hình thức thanh toán điện tử phổ biến hiện nay	29
CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU.....		30
3.1	Mô tả bài toán.....	30
3.2	Xây dựng website.....	32
3.2.1	Mô hình dữ liệu mức quan niệm	32
3.2.2	Mô hình dữ liệu mức logic	32
3.2.3	Các thực thể.....	33
3.3	Thiết kế xử lý	35
3.3.1	Biểu đồ Use Case tổng quát	35
3.3.2	Biểu đồ Use Case tác nhân người bán (người cho thuê)	35
3.3.3	Biểu đồ Use Case tác nhân người mua (người thuê).....	36
3.3.4	Biểu đồ Use Case tác nhân quản trị.....	37
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU.....		38
4.1	Giao diện người dùng.....	38
4.1.1	Giao diện trang chủ	38
4.1.2	Giao diện đăng nhập.....	39
4.1.3	Giao diện đăng ký.....	39
4.1.4	Giao diện trang tìm kiếm bất động sản theo loại.....	39
4.1.5	Giao diện chi tiết bất động sản	39
4.1.6	Giao diện trang quản lý tin đăng (của người bán/ người cho thuê).....	39
4.1.7	Giao diện trang điền thông tin thanh toán	39
4.1.8	Giao diện trang thanh toán đăng tin bất động sản	39
4.1.9	Giao diện trang thanh toán chi phí liên hệ giữa người mua và người bán	39
4.1.10	Giao diện thanh toán qua VNPAY	39

4.2	Giao diện quản trị.....	39
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN		40
5.1	Kết luận	40
5.2	Hướng phát triển	40
DANH MỤC TÀI LIỆU THAM KHẢO.....		41
PHỤ LỤC		42

DANH MỤC HÌNH ẢNH – BẢNG BIỂU

Hình 1. Javascript	14
Hình 2. React	15
Hình 3. Folder frontend react.....	16
Hình 4. Cơ chế hoạt động của NodeJS	18
Hình 5. Folder backend NodeJS	18
Hình 6. ExpressJS framework	20
Hình 7. Cơ chế hoạt động MVC	22
Hình 8. Tailwind CSS	24
Hình 9. MongoDB	26
Hình 10. Cách hoạt động MongoDB	27
Hình 11. Biểu đồ Use Case tác nhân người bán (người cho thuê).....	35
Hình 12. Biểu đồ Use Case tác nhân người bán (người cho thuê) đăng tin.....	36
Hình 13. Biểu đồ Use Case tác nhân người mua (người thuê)	36
Hình 14. Biểu đồ Use Case liên hệ giữa người mua và người bán.....	37
Hình 15. Biểu đồ Use Case tác nhân quản trị	37
Hình 16. Giao diện trang chủ phần trên.....	38
Hình 17. Giao diện trang chủ phần dưới.....	38
Bảng 1. Bảng User (bảng người dùng)	33
Bảng 2. Bảng Bất động sản (bds)	33
Bảng 3. Bảng Images (ảnh bất động sản)	33
Bảng 4. Bảng Seller (người bán/ người cho thuê)	33
Bảng 5. Bảng Buyer (người mua/ người thuê)	34
Bảng 6. Bảng Transaction (bảng giao dịch)	34
Bảng 7. Bảng Payment (thanh toán)	34

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Diễn giải
API	Application programming interface
BDS (bds)	Bất động sản
MVC	Model-view-controller
JS	Javascript

TÓM TẮT ĐỒ ÁN CHUYÊN NGÀNH

Đề tài "Xây dựng website rao vặt bất động sản Hai Lúa tại Trà Vinh" tập trung phát triển một nền tảng trực tuyến để hỗ trợ việc mua bán, cho thuê bất động sản tại khu vực Trà Vinh. Website được xây dựng bằng React và NodeJS, hướng tới việc cung cấp giao diện thân thiện, dễ sử dụng cùng các tính năng như đăng tin rao vặt, tìm kiếm, lọc theo nhiều tiêu chí, và quản lý tin đăng. Ngoài ra, hệ thống còn tích hợp cổng thanh toán nhằm hỗ trợ thu phí đăng tin nếu cần, cùng chức năng liên hệ để kết nối người mua và người bán một cách nhanh chóng và hiệu quả.

Mục tiêu chính

- Phân tích nhu cầu của người dùng để xây dựng hệ thống quản lý tin rao vặt bất động sản tại khu vực Trà Vinh, với các chức năng đăng tin, chỉnh sửa, xóa tin và tìm kiếm hiệu quả.
- Phát triển công cụ tìm kiếm và lọc hiện đại, hỗ trợ người dùng tìm bất động sản theo tiêu chí loại hình, vị trí, giá cả, diện tích một cách nhanh chóng.
- Tích hợp cổng thanh toán trực tuyến và hệ thống liên hệ trực tiếp, đảm bảo giao dịch an toàn và tạo sự kết nối thuận tiện giữa người mua và người bán.

Hướng tiếp cận

- Sử dụng React và NodeJS để xây dựng hệ thống với giao diện thân thiện, hiệu quả, cùng khả năng quản lý dữ liệu nhanh chóng và mở rộng linh hoạt.
- Tích hợp công cụ tìm kiếm, lọc hiện đại và chức năng quản lý tin rao vặt, tạo điều kiện cho người dùng dễ dàng tìm kiếm và quản lý thông tin bất động sản.

Kết quả đạt được

- Qua quá trình phát triển website rao vặt bất động sản, tôi đã học được cách áp dụng các kiến thức về thiết kế giao diện, quản lý dữ liệu và tích hợp các tính năng tìm kiếm, lọc hiệu quả.
- Xây dựng thành công một nền tảng giao dịch bất động sản tại Trà Vinh, với các chức năng đăng tin, quản lý tin đăng, tìm kiếm và thanh toán trực tuyến, giúp người dùng dễ dàng tương tác và thực hiện giao dịch.

MỞ ĐẦU

1. Lý do chọn đề tài

Với sự phát triển mạnh mẽ của công nghệ thông tin, việc ứng dụng Internet vào các lĩnh vực kinh doanh và giao dịch trở nên ngày càng phổ biến. Internet giúp truyền tải thông tin nhanh chóng và hiệu quả, mang lại sự kết nối giữa người dùng ở bất kỳ đâu và vào bất kỳ thời điểm nào. Đặc biệt trong lĩnh vực bất động sản, sở hữu một website rao vặt là giải pháp tuyệt vời để kết nối người mua và người bán, tiết kiệm thời gian và công sức. Với xu hướng này, tôi đã chọn đề tài xây dựng website rao vặt bất động sản, sử dụng các công nghệ hiện đại như React cho phần frontend và NodeJS cho phần backend. React mang đến sự linh hoạt trong việc phát triển giao diện người dùng, giúp tối ưu hóa trải nghiệm và hiệu suất khi tương tác với hệ thống. NodeJS, với khả năng xử lý nhanh và mở rộng, hỗ trợ xây dựng hệ thống backend mạnh mẽ, dễ dàng xử lý lượng truy cập lớn và tích hợp các tính năng bảo mật. Việc kết hợp React và NodeJS không chỉ giúp tối ưu tốc độ mà còn tạo ra một nền tảng vững chắc cho website, sẵn sàng phát triển và mở rộng trong tương lai.

2. Mục tiêu

Tìm hiểu và sử dụng React và NodeJS, đồng thời sử dụng để xây dựng Website rao vặt bất động sản tại Trà Vinh, qua đó hiểu được cách thức hoạt động cũng như cách sử dụng các công cụ trong React và NodeJS.

3. Nội dung

Tìm hiểu và ứng dụng React và NodeJS để xây dựng website rao vặt bất động sản.

Hiểu rõ quy trình phát triển website với sự ứng dụng thực tế của React và NodeJS.

4. Phương pháp nghiên cứu

Phương pháp nghiên cứu lý thuyết: Tìm hiểu về React, NodeJS và các công nghệ liên quan qua tài liệu, sách, khóa học trực tuyến để nắm vững kiến thức về xây dựng ứng dụng web động.

Phương pháp khảo sát: Phỏng vấn và khảo sát người dùng về nhu cầu sử dụng website rao vặt bất động sản, các tính năng họ mong muốn và các vấn đề họ gặp phải khi giao dịch bất động sản trực tuyến.

5. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu

Framework là một bộ mã nguồn được xây dựng, phát triển và đóng gói, phân phối bởi các chuyên gia lập trình hoặc bởi các công ty lập trình. React giúp phát triển giao diện, NodeJS với Express cung cấp môi trường backend mạnh mẽ cho việc lập trình.

Mô hình MVC: Mô hình MVC (Model-View-Controller) được sử dụng để phân chia cấu trúc ứng dụng web thành ba thành phần chính, giúp quản lý mã nguồn dễ dàng hơn và tăng khả năng mở rộng.

React và NodeJS (Express): React sử dụng để xây dựng giao diện người dùng động, trong khi NodeJS (Express) được sử dụng cho việc phát triển API và xử lý các thao tác backend như lưu trữ dữ liệu, thanh toán và quản lý tin đăng.

Phạm vi nghiên cứu

Nghiên cứu tập trung vào việc ứng dụng React và NodeJS (Express) để xây dựng website rao vặt bất động sản, bao gồm việc áp dụng mô hình MVC cho việc quản lý dữ liệu và phát triển giao diện người dùng, tối ưu hóa hiệu suất và bảo mật của hệ thống.

CHƯƠNG 1: TỔNG QUAN

Trong bối cảnh công nghệ thông tin ngày càng phát triển, website đã trở thành công cụ không thể thiếu trong nhiều lĩnh vực, từ giải trí đến kinh doanh và giao dịch trực tuyến. Đặc biệt trong lĩnh vực bất động sản, việc xây dựng một website rao vặt không chỉ giúp kết nối người mua và người bán mà còn nâng cao hiệu quả giao dịch và tiết kiệm thời gian cho người sử dụng. Một website rao vặt bất động sản sẽ giúp người dùng dễ dàng đăng tin về các loại bất động sản như nhà đất bán, cho thuê, căn hộ, và đất nền. Đồng thời, website này cũng cung cấp các công cụ tìm kiếm và lọc thông tin, giúp người dùng dễ dàng tìm kiếm bất động sản theo nhiều tiêu chí như giá tiền, diện tích, vị trí, và loại bất động sản.

Website rao vặt bất động sản này hướng đến các đối tượng người mua, bán bất động sản trong và ngoài khu vực Trà Vinh, mang lại những lợi ích lớn cho việc kết nối giữa người sử dụng và giao dịch bất động sản một cách nhanh chóng, thuận tiện. Các tính năng quan trọng của website bao gồm: đăng ký/đăng nhập tài khoản cá nhân hoặc doanh nghiệp, quản lý tin đăng, tìm kiếm và lọc theo các tiêu chí cụ thể, và tích hợp cổng thanh toán nếu cần thu phí đăng tin. Hệ thống cũng phải đảm bảo bảo mật thông tin người dùng, đồng thời hỗ trợ liên hệ trực tiếp giữa người mua và người bán.

Về phương thức phát triển, website sẽ sử dụng mô hình MVC (Model - View - Controller) để tách biệt và quản lý các thành phần của ứng dụng một cách khoa học và hiệu quả. Mô hình MVC giúp chia ứng dụng thành ba phần riêng biệt, với Model xử lý dữ liệu và logic nghiệp vụ, View hiển thị giao diện người dùng, và Controller làm cầu nối giữa Model và View. Cách tiếp cận này giúp tăng cường tính linh hoạt, dễ dàng mở rộng và bảo trì ứng dụng. Đặc biệt, việc sử dụng React cho frontend sẽ giúp tối ưu hóa trải nghiệm người dùng với giao diện nhanh chóng và dễ dàng tương tác. Node.js (Express) sẽ đảm nhận backend, giúp xử lý các yêu cầu của người dùng một cách nhanh chóng, mở rộng khả năng của website và đảm bảo bảo mật.

Đề tài ứng dụng React, Node.js (Express), và mô hình MVC để xây dựng website rao vặt bất động sản tại Trà Vinh. Website đảm bảo bảo mật, dễ mở rộng, và tối ưu tốc độ xử lý.

CHƯƠNG 2: NGHIÊN CỨU LÝ THUYẾT

2.1 Javascript

2.1.1 Sơ lược về ngôn ngữ Javascript

JavaScript, còn được viết tắt là JS, là một ngôn ngữ lập trình, chạy dưới dạng ngôn ngữ kịch bản phía máy khách trong môi trường hosting. Môi trường hosting phổ biến nhất là trình duyệt web, dùng ngôn ngữ này để tạo tính tương tác cho trang web.

JavaScript cho phép thực thi mã lệnh ở phía client, tức là ngay trên trình duyệt thay vì ở máy chủ từ xa (client-side).

Client-side JavaScript là mã JavaScript được thực thi trực tiếp trên trình duyệt của người dùng (Google Chrome, Microsoft Edge, Fire Fox), thường được sử dụng để tạo ra các tính năng tương tác và động trên trang web [1].

2.1.2 Đặc điểm

- Là ngôn ngữ kịch bản (scripting language) hướng đối tượng.
- Là ngôn ngữ nhập động (dynamic typing) dựa trên prototype.
- Là ngôn ngữ lightweight (dễ học và viết code nhanh).
- Là ngôn ngữ đa mô hình (multi-paradigm) và đơn luồng (single threaded).

2.1.3 Cách hoạt động

Đối với JS phía máy khách (client-side):

- Tích hợp sẵn bên trong hầu hết các trình duyệt web phổ biến.
- Trình duyệt chuyển đổi trang web đó và tất cả các phần tử của nó thành dạng cấu trúc dữ liệu gọi là DOM
- Công cụ JavaScript của trình duyệt thực hiện chuyển mã JavaScript thành mã byte.
- Khi có sự kiện xảy ra, ví dụ như nhấp vào button, việc thực thi code block JavaScript liên kết sẽ được kích hoạt.

Đối với JS phía máy chủ (server-side):

- Truy cập cơ sở dữ liệu

- Thực hiện các tính toán logic khác nhau
- Phản hồi các sự kiện khác nhau do hệ điều hành của máy chủ kích hoạt [2].



Hình 1. Javascript

2.2 React framework

2.2.1 Giới thiệu

React (ReactJS) là một thư viện JavaScript mã nguồn mở, được dùng để xây dựng giao diện người dùng (frontend) cho web. React chỉ tập trung vào phần hiển thị giao diện (view), chứ không can thiệp vào cách sắp xếp logic nghiệp vụ hoặc cấu trúc ứng dụng.

React tập trung vào việc hiển thị giao diện người dùng và cho phép lập trình viên tự do quyết định cách sắp xếp logic nghiệp vụ.

Khác với các framework có kiến trúc cố định như Angular, React không ép buộc người dùng vào một mô hình cụ thể, khiến nó linh hoạt cho nhiều dự án khác nhau.

2.2.2 Đặc điểm nổi bật

- Linh hoạt trong thiết kế kiến trúc
- Kiến trúc Component đơn giản và nhẹ: mỗi thành phần có thể được tái sử dụng, giúp ứng dụng dễ mở rộng và duy trì. Các component trong React rất nhẹ và có thể chỉ là các hàm đơn giản trả về JSX.
- Cộng đồng hỗ trợ lớn: React được duy trì và phát triển bởi **Facebook** (Meta), một trong những công ty công nghệ lớn nhất thế giới. Điều này đảm bảo rằng React

sẽ nhận được sự hỗ trợ liên tục và các bản cập nhật mới, giúp lập trình viên yên tâm rằng React sẽ tiếp tục phát triển và phù hợp với các dự án dài hạn.

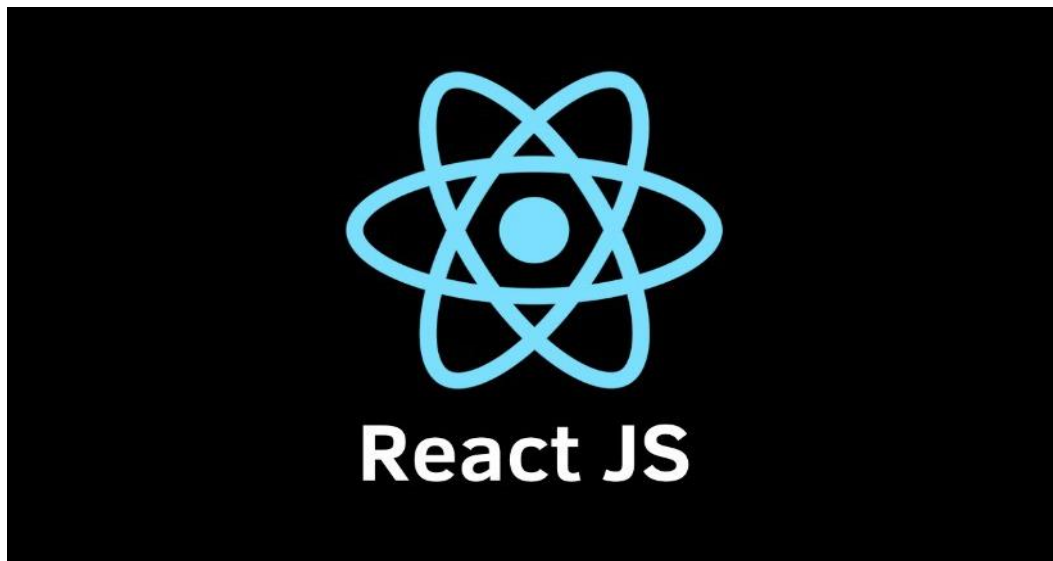
2.2.3 Các thành phần quan trọng

- JSX (JavaScript XML) là một cú pháp mở rộng cho phép bạn viết mã giống như HTML trong JavaScript. Trong các ngôn ngữ khác, bạn thường phải viết code HTML và JavaScript riêng rẽ. Tuy nhiên, với JSX, React cho phép bạn kết hợp cả hai trong cùng một mã nguồn, giúp quản lý dễ dàng hơn, đặc biệt là trong các ứng dụng phức tạp.

- Virtual DOM (Document Object Model ảo) là một bản sao nhẹ hơn của DOM thật. DOM thật là cấu trúc cây chứa tất cả các thành phần HTML trong trang web. Khi người dùng tương tác với ứng dụng (ví dụ: nhập văn bản, nhấn nút), ứng dụng sẽ thay đổi nội dung và DOM thật phải được cập nhật.

- Component là các đơn vị cơ bản trong React, cung cấp cấu trúc cho giao diện người dùng. Mỗi component được khuyến khích càng nhỏ gọn và độc lập càng tốt để có thể tái sử dụng trong nhiều phần của ứng dụng hoặc thậm chí trong các ứng dụng khác.

- Liên kết dữ liệu một chiều: trong React, dữ liệu được quản lý và luân chuyển theo hướng từ component cha xuống component con. Các component cha có thể truyền dữ liệu cho component con thông qua props, nhưng các component con không thể trực tiếp thay đổi dữ liệu của component cha [3].



Hình 2. React

- Các bước cài đặt:

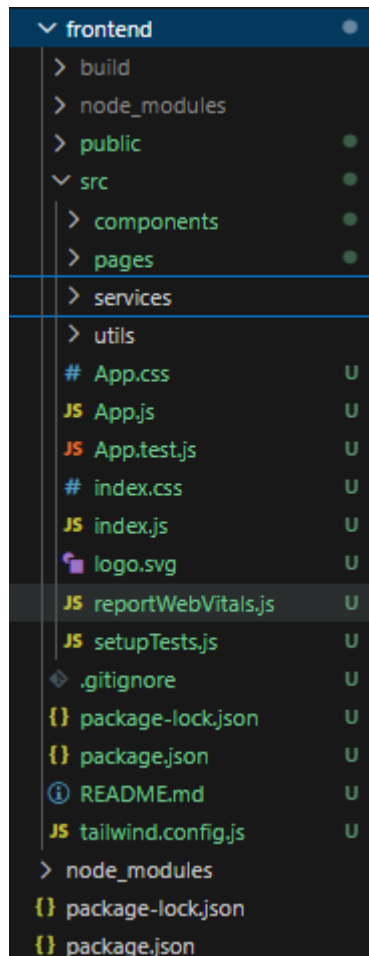
+ Cần phải có môi trường cài đặt, NodeJS: <https://nodejs.org/en>

. Mở cmd (Command Prompt) và gõ:

+ Tạo một thư mục frontend chứa mã nguồn React: **npx create-react-app frontend**

+ Di chuyển vào thư mục frontend: **cd frontend**

+ Chạy ứng dụng React: **npm start**



Hình 3. Folder frontend react

+ Tạo các folder để dễ dàng quản lý mã nguồn theo chức năng nhất định.

2.3 NodeJS

2.3.1 Giới thiệu

Nodejs là một môi trường runtime chạy JavaScript đa nền tảng, open-source (mã nguồn mở), được sử dụng rộng rãi bởi lập trình viên cho nhiều dự án. Nodejs cung cấp nhiều thư viện khác nhau, giúp đơn giản hóa việc lập trình.

Nodejs là một môi trường JavaScript runtime, đa nền tảng: Windows, Linux, Unix, Mac OS X,...

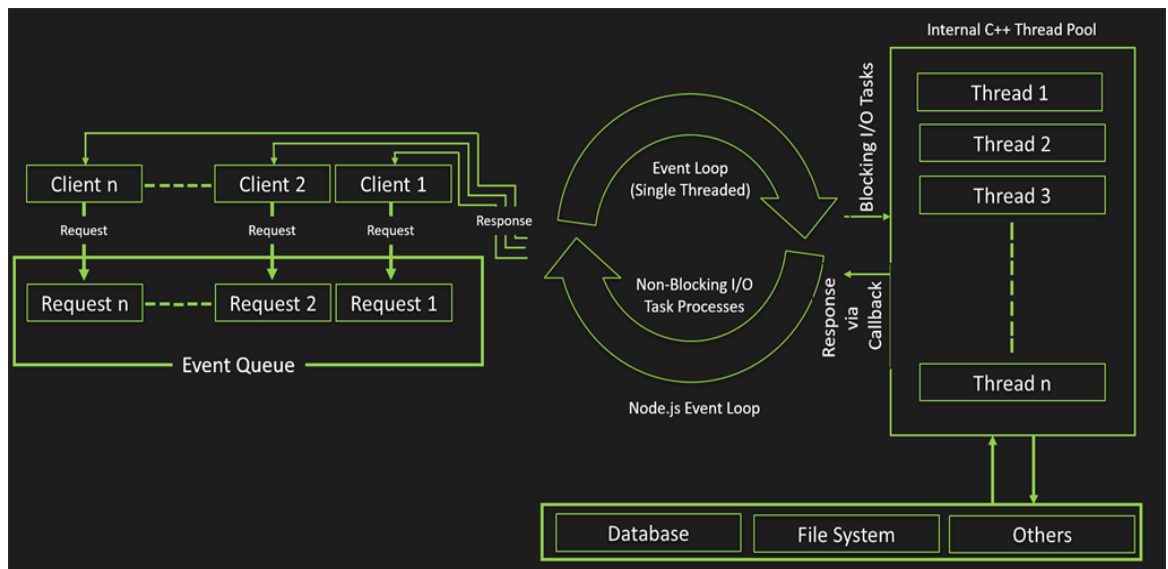
Một trong những tính năng chính của Nodejs là, cho phép xử lý đồng thời nhiều yêu cầu, mà không chặn các yêu cầu khác.

2.3.2 Đặc điểm vượt trội

- Lập trình bất đồng bộ: Server không cần chờ dữ liệu mà API trả về, vẫn có thể xử lý các yêu cầu khác.
- Tốc độ: Được xây dựng dựa trên engine JavaScript V8, Nodejs thực thi mã nhanh.
- NPM: Với hơn 50,000 package khác nhau, lập trình viên dễ dàng tìm kiếm, lựa chọn và tích hợp tính năng cần thiết vào dự án của mình.
- Không có buffering: Nodejs xử lý các tệp âm thanh và video hiệu quả bằng cách xuất dữ liệu theo từng phần (chunk) thay vì buffer dữ liệu.
- Đơn luồng: Nodejs sử dụng mô hình đơn luồng với vòng lặp sự kiện (Event loop), giúp xử lý số lượng request nhiều hơn so với các server truyền thống như Apache HTTP Server.

2.3.3 Cơ chế hoạt động

- Nhiều client sẽ gửi request đến server, sẽ được đưa vào Event Queue.
- Event Loop sẽ lấy lần lượt các request từ hàng đợi và xử lý chúng.
- Nếu request cần thực hiện các tác vụ non-blocking I/O, Event Loop sẽ tự động xử lý rồi trả về response.
- Đối với những tác vụ blocking I/O, Event Loop sẽ đưa tasks qua Thread Pool (thông qua thư viện Libuv).
- Khi thread đã thực hiện xong, kết quả sẽ được Nodejs đẩy callback của tasks đó về Event Queue để xử lý.
- Cuối cùng, response sẽ được phản hồi lại cho các client [4].



Hình 4. Cơ chế hoạt động của NodeJS

- Các bước cài đặt:

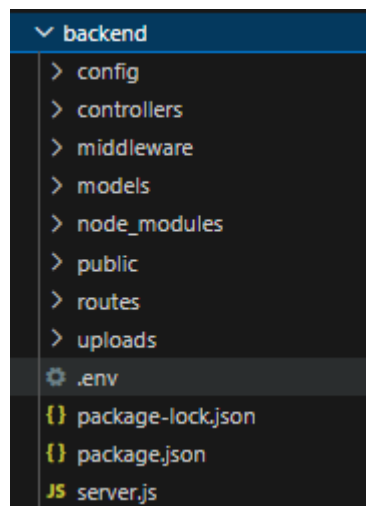
+ Cài đặt Node.js tại đường link: <https://nodejs.org/en>

. Mở cmd (Command Prompt) và gõ:

+ Tạo thư mục cho backend: **mkdir backend**

+ Di chuyển vào thư mục: **cd backend**

+ Khởi tạo một dự án Node.js mới trong thư mục backend: **npm init -y**



Hình 5. Folder backend NodeJS

2.4 Express.js framework

2.4.1 Giới thiệu

Express.js là một framework đơn giản được xây dựng trên nền tảng Node.js, ra đời với mục đích làm cho việc phát triển các ứng dụng web và API trở nên đơn giản, hiệu quả và dễ bảo trì hơn. Bằng cách cung cấp các công cụ tiện lợi để xử lý yêu cầu HTTP, quản lý định tuyến (routing), và thêm các chức năng khác vào ứng dụng thông qua middleware.

Với API RESTful, lập trình viên có thể dễ dàng hơn và quản lý dữ liệu hiệu quả mà không cần viết quá nhiều code phức tạp. Express.js giúp bạn tập trung vào logic ứng dụng thay vì xử lý chi tiết các yêu cầu HTTP.

2.4.2 Các tính năng chính

- Templating engines cho phép bạn tạo các mẫu (templates) chứa các placeholder (chỗ trống) để sau đó có thể thay thế bằng dữ liệu thực tế khi trang được hiển thị, giúp tạo ra các trang HTML linh hoạt và dễ dàng hơn, đặc biệt là khi dữ liệu thay đổi thường xuyên.

- Middleware `express.static()` giúp việc phục vụ các tệp tĩnh trong dự án trở nên đơn giản. Bạn chỉ cần chỉ định thư mục chứa các tệp tĩnh (như hình ảnh, CSS, JavaScript), và Express.js sẽ tự động xử lý việc cung cấp chúng cho người dùng khi họ yêu cầu.

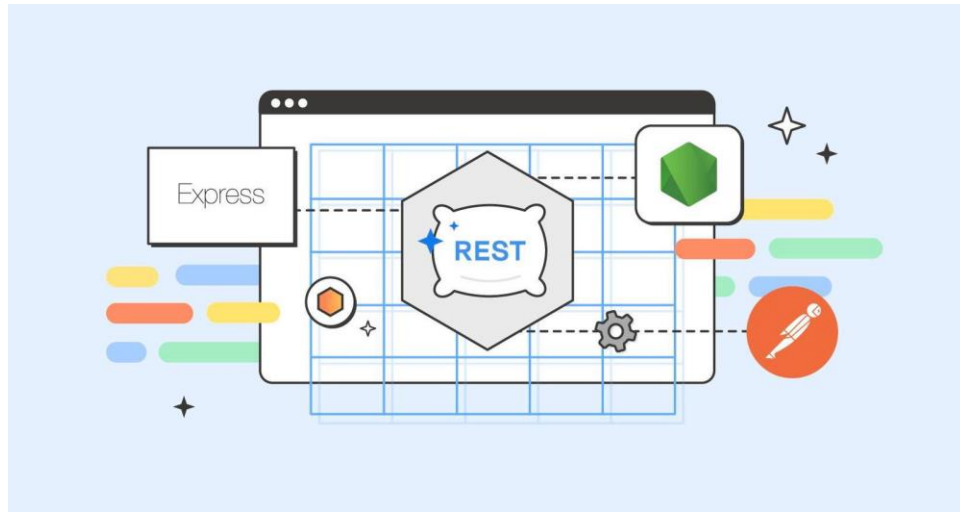
- Tệp tĩnh (static files) là những tệp không cần xử lý phức tạp từ máy chủ, mà chỉ được gửi trực tiếp đến người dùng, như hình ảnh, tệp CSS để tạo kiểu, hoặc tệp JavaScript để chạy trên trình duyệt.

- Routing (Định tuyến) trong Express.js là cách bạn thiết lập các quy tắc để xử lý các yêu cầu HTTP từ người dùng đến đúng nơi trong ứng dụng của bạn. Mỗi URL sẽ được gán với một route (định tuyến), và mỗi route có thể xử lý các loại yêu cầu khác nhau như GET, POST, PUT, DELETE, v.v.

- Middleware là những hàm đặc biệt trong Express.js, được thực thi khi ứng dụng của bạn nhận yêu cầu từ người dùng và trước khi gửi phản hồi về cho họ. Middleware có thể thực hiện nhiều nhiệm vụ như:

- + Ghi lại thông tin về yêu cầu.

- + Xác thực người dùng.
- + Xử lý dữ liệu.
- + Trả về lỗi khi có vấn đề xảy ra [5].



Hình 6. ExpressJS framework

Các bước cài đặt:

- + Sau khi khởi tạo xong thư mục backend với NodeJS ➔ tải Express (framework): **npm install express cors**

Kết nối frontend với backend:

- + Mở tệp frontend/package.json và thêm mục proxy như sau:

```
{  
  "name": "frontend",  
  "version": "0.1.0",  
  "private": true,  
  "dependencies": {  
    // Các phụ thuộc khác  
  },  
  "proxy": http://localhost:5000 //Dòng thêm vào  
}
```

- ➔ Thực hiện các yêu cầu từ React (frontend) đến một URL tương đối (ví dụ: /api/), React sẽ tự động chuyển hướng yêu cầu đến server Node.js đang chạy trên <http://localhost:5000>

2.5 Mô hình MVC

2.5.1 Tổng quan về mô hình MVC

Mô hình MVC, viết tắt của Model-View-Controller, là một kiến trúc phần mềm phổ biến được sử dụng rộng rãi trong phát triển ứng dụng. Được thiết kế để tách biệt logic xử lý dữ liệu (Model), hiển thị giao diện người dùng (View), và điều phối cả hai (Controller), mô hình này mang lại nhiều lợi ích trong việc quản lý mã nguồn và cải thiện khả năng bảo trì của hệ thống.

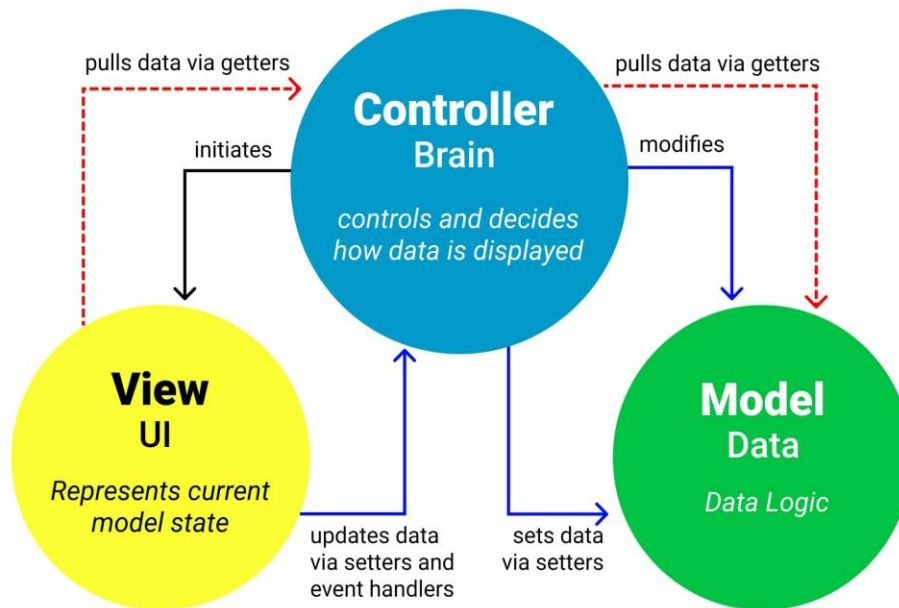
Model (M - Model): Đây là thành phần chịu trách nhiệm cho xử lý dữ liệu và logic kinh doanh của ứng dụng. Model là nơi lưu trữ thông tin, thực hiện các thao tác cập nhật và truy vấn dữ liệu, mà không quan tâm đến cách dữ liệu được hiển thị hoặc tương tác với người dùng.

View (V - View): View là thành phần hiển thị giao diện người dùng và đảm nhận trách nhiệm hiển thị thông tin từ Model. View không có logic kinh doanh và chỉ chịu trách nhiệm về việc hiển thị dữ liệu một cách đẹp mắt và dễ hiểu cho người dùng.

Controller (C - Controller): Controller là thành phần điều phối và xử lý sự kiện từ người dùng. Nó nhận lệnh từ người dùng thông qua View, sau đó cập nhật Model dựa trên những thay đổi này và điều hướng hiển thị trở lại View. Controller giữ vai trò quan trọng trong việc duy trì sự đồng bộ giữa Model và View.

Mô hình MVC giúp tách biệt logic ứng dụng thành các phần nhỏ, dễ quản lý và bảo trì. Nó cũng thúc đẩy tái sử dụng mã nguồn, khiến cho quá trình phát triển và bảo trì ứng dụng trở nên hiệu quả hơn.

MVC Architecture Pattern



Hình 7. Cơ chế hoạt động MVC

2.5.2 Các thành phần

*** **Model:** Đại diện cho dữ liệu và logic kinh doanh của ứng dụng. Nó là nơi lưu trữ thông tin, thực hiện các thao tác cập nhật và truy vấn dữ liệu. Model không quan tâm đến cách dữ liệu được hiển thị hoặc tương tác với người dùng.

Trách nhiệm:

- Lưu trữ và quản lý dữ liệu.
- Thực hiện logic kinh doanh và xử lý sự kiện.
- Cung cấp API cho các thành phần khác để truy xuất và cập nhật dữ liệu.

*** **View:** View là thành phần chịu trách nhiệm hiển thị thông tin từ Model cho người dùng. Nó không có logic kinh doanh và chỉ tập trung vào việc hiển thị dữ liệu một cách đẹp mắt và dễ hiểu.

Trách nhiệm:

- Hiển thị dữ liệu cho người dùng.
- Tương tác với người dùng thông qua giao diện người dùng.

- Thông báo về các sự kiện tương tác đến Controller.

*** **Controller:** Controller là thành phần điều phối và xử lý sự kiện từ người dùng. Nó nhận lệnh từ người dùng thông qua View, sau đó cập nhật Model dựa trên những thay đổi này và điều hướng hiển thị trở lại View.

Trách nhiệm:

- Nhận các sự kiện từ View.
- Gửi các yêu cầu cập nhật hoặc truy vấn dữ liệu đến Model.
- Cập nhật View dựa trên sự thay đổi trong Model.
- Giữ sự đồng bộ giữa Model và View.

2.5.3 Ưu điểm và Nhược điểm

Ưu điểm:

- + Tách biệt logic: MVC giúp tách biệt logic ứng dụng thành các phần riêng biệt (Model, View, và Controller), làm cho mã nguồn dễ hiểu và dễ bảo trì.
- + Tái sử dụng mã nguồn: Các thành phần trong MVC có thể được tái sử dụng một cách linh hoạt. Model và View có thể được sử dụng lại trong nhiều phần của ứng dụng hoặc trong các ứng dụng khác nhau.
- + Phân chia công việc: Mô hình này giúp phân chia công việc giữa các thành viên trong nhóm phát triển, với người làm Model tập trung vào logic xử lý dữ liệu, người làm View chịu trách nhiệm về giao diện người dùng, và người làm Controller quản lý sự tương tác.
- + Dễ mở rộng: Do sự tách biệt giữa các thành phần, việc mở rộng và thay đổi ứng dụng trở nên dễ dàng hơn. Một thành phần có thể được thay thế hoặc mở rộng mà không làm ảnh hưởng đến các thành phần khác.

Nhược điểm:

- + Phức tạp cho ứng dụng nhỏ: Đối với các ứng dụng nhỏ hoặc đơn giản, việc triển khai mô hình MVC có thể quá phức tạp và làm tăng độ phức tạp của mã nguồn.

+ Tăng độ phức tạp của mã nguồn: Trong một số trường hợp, việc chia nhỏ ứng dụng thành nhiều thành phần có thể làm tăng độ phức tạp của mã nguồn và làm cho mã trở nên khó hiểu [6].

2.6 Tailwind CSS

2.6.1 Tổng quan về Tailwind CSS

Tailwind CSS là một framework CSS utility-first giúp tạo kiểu nhanh chóng và hiệu quả cho website mà không cần viết CSS thủ công. Thay vì cung cấp các thành phần (component) hoặc kiểu thiết kế mặc định, Tailwind cung cấp một loạt các lớp tiện ích (utility classes) mà bạn có thể kết hợp linh hoạt để xây dựng giao diện tùy chỉnh.



Hình 8. Tailwind CSS

2.6.2 Tổng quan về Tailwind CSS

Ưu điểm:

+ Tùy biến cao, hiệu suất cao, định nghĩa những phần phù hợp với dự án của bạn mà không bị gò bó.

+ Cho phép xây dựng responsive layout phức tạp.

+ Responsive và phát triển dễ dàng.

+ Tạo thành phần dễ dàng.

+ Hỗ trợ cài đặt với nhiều framework front-end khác như react, vuejs,...

Nhược điểm:

+ Thiếu tiêu đề và thành phần điều hướng (navigation).

+ Cần có thời gian để làm quen, tìm hiểu và nhớ tên các class [7].

+ Có kiến thức về CSS thì mới sử dụng tốt được.

Hướng dẫn cài đặt:

Trong dự án có sử dụng npm như React NodeJS (ExpressJS):

Mở cmd (Command Prompt) và gõ:

+ **npm install tailwindcss postcss autoprefixer**

+ **npx tailwindcss init** (tạo cấu hình)

+ Tạo một postcss.config.js file trong thư mục gốc của thư mục dự án và thêm nội dung sau:

```
module.exports = {  
  plugins: [  
    require('tailwindcss'),  
    require('autoprefixer'),  
  ],  
}
```

Sau đó, tạo một styles.css file trong thư mục dự án của bạn src (hoặc bất kỳ thư mục nào khác mà bạn thích) và thêm nội dung sau:

@tailwind base;

@tailwind components;

@tailwind utilities;

Cuối cùng, thêm tập lệnh vào package.json file của bạn để tạo CSS:

```
"scripts": {  
  "build": "postcss src/styles.css -o dist/styles.css"  
}
```

2.7 Hệ quản trị cơ sở dữ liệu MongoDB

2.7.1 Giới thiệu

MongoDB là một trong những cơ sở dữ liệu phi quan hệ (NoSQL) phổ biến nhất hiện nay, được sử dụng rộng rãi trong nhiều lĩnh vực. MongoDB cung cấp các tính năng hữu ích giúp cho việc phân tích, truy vấn và cập nhật dữ liệu trở nên dễ dàng và nhanh chóng.

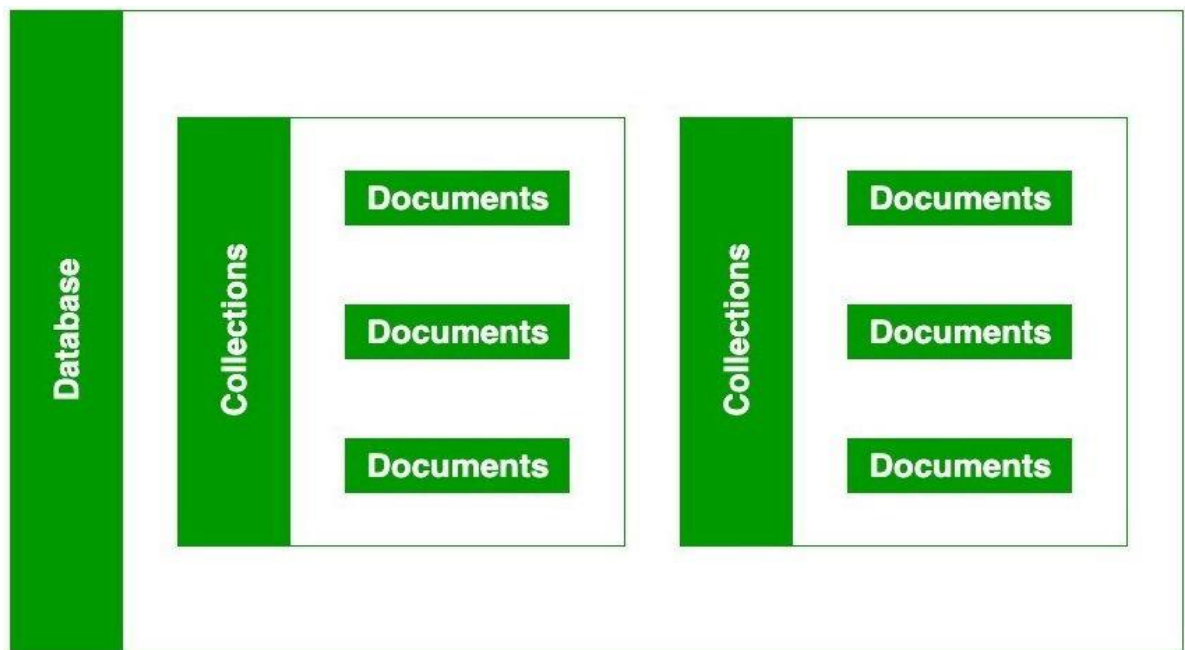
MongoDB hỗ trợ tất cả các ngôn ngữ phổ biến như C, C++, C# và .Net, Go, Java, Node.js, Perl, PHP, Python, Motor, Ruby, Scala, Swift, Mongoid.



Hình 9. MongoDB

2.7.2 Cách hoạt động

- **MongoDB** hoạt động dưới dạng một hệ thống cơ sở dữ liệu phi quan hệ, lưu trữ dữ liệu dưới dạng tài liệu (document) JSON. Dữ liệu được lưu trữ trong collections và documents.



Hình 10. Cách hoạt động MongoDB

- Cơ sở dữ liệu MongoDB lưu trữ tài liệu trong các collections, tương tự như bảng trong cơ sở dữ liệu quan hệ. Mỗi collection có thể chứa nhiều tài liệu (documents) có cấu trúc dữ liệu tùy ý, có thể chứa nhiều fields dữ liệu, mỗi field được định danh bằng tên và có giá trị tương ứng.

- Các tài liệu (documents) được tạo bằng cách sử dụng các field. Các field là các key-value pair trong tài liệu, nó giống như các cột trong cơ sở dữ liệu quan hệ.

- Giá trị của fields có thể thuộc bất kỳ loại dữ liệu BSON nào như double, string, boolean,...

- MongoDB hỗ trợ việc tạo index cho các field dữ liệu trong collection, giúp tăng tốc độ truy vấn. Chúng còn hỗ trợ sao chép dữ liệu giữa các node trong một cluster giúp đảm bảo tính khả dụng và độ tin cậy của hệ thống.

- MongoDB phân tán dữ liệu trên nhiều node, giúp tăng khả năng mở rộng của hệ thống, đồng thời chúng còn hỗ trợ tính toán phân tán bằng cách sử dụng MapReduce giúp xử lý dữ liệu lớn một cách hiệu quả.

2.7.3 Ưu điểm và Nhược điểm

Ưu điểm:

- **Tính linh hoạt:** MongoDB là một hệ thống cơ sở dữ liệu phi quan hệ, nó cung cấp khả năng lưu trữ dữ liệu bất cứ khi nào, bất cứ nơi đâu, không cần phải tuân thủ một mô hình quan hệ cụ thể.

- **Khả năng mở rộng, truy xuất nhanh:** MongoDB có khả năng mở rộng dễ dàng, nhờ tính năng sharding cho phép phân chia dữ liệu thành nhiều phần và lưu trữ trên nhiều máy chủ, truy vấn dữ liệu trong thời gian ngắn.

- **Tính khả dụng cao:** MongoDB cung cấp tính năng sao lưu và phục hồi dữ liệu, giúp người dùng bảo vệ dữ liệu của mình khỏi những rủi ro.

- **Dễ sử dụng:** MongoDB cung cấp các công cụ quản lý dữ liệu trực quan và dễ sử dụng, giúp người dùng tối ưu hóa hiệu suất và quản lý cơ sở dữ liệu một cách dễ dàng.

Nhược điểm:

- Cần sử dụng bộ nhớ cao để lưu trữ dữ liệu (data storage).
- Không được phép lưu trữ hơn 16MB data trong tài liệu.
- Data nesting trong BSON cũng bị hạn chế, bạn không được phép nest data quá 100 cấp độ [8].

➔ Cài đặt tại đường link: <https://www.mongodb.com/try/download/community>

2.8 Thanh toán điện tử

2.8.1 Tổng quan

Thanh toán trực tuyến (còn gọi là thanh toán điện tử) là hình thức thực hiện các giao dịch ngay trên điện thoại thông minh hoặc máy tính có kết nối mạng. Với số tiền có sẵn trong tài khoản ngân hàng, người dùng có thể chuyển tiền, chi trả hóa đơn mua sắm hay hóa đơn điện, nước, truyền hình cáp, hoặc nạp tiền điện thoại... nhanh chóng mà không cần sử dụng tiền mặt. Thanh toán điện tử bao gồm đa dạng hình thức, có thể thanh toán trực tiếp qua tài khoản ngân hàng của người dùng hoặc qua các đơn vị trung gian liên kết với ngân hàng như cổng thanh toán, ví điện tử,...

2.8.2 Lợi ích của thanh toán điện tử

- Tiện lợi và tiết kiệm thời gian: Với dịch vụ thanh toán trực tuyến, bạn chỉ cần sử dụng thiết bị di động kết nối mạng là đã có thể chuyển, nhận tiền, thanh toán hóa đơn... Quá trình giao dịch nhanh chóng, tiện lợi, có thể linh hoạt thực hiện mọi nơi mọi lúc.

- Quản lý tài chính chặt chẽ hơn: Khi thanh toán điện tử, những khoản chi tiêu đều được lưu lại rõ ràng trong lịch sử giao dịch. Điều này giúp bạn dễ dàng kiểm tra và quản lý số dư, từ đó cân chỉnh kế hoạch tài chính cá nhân hiệu quả.

- Hạn chế rủi ro so với thanh toán tiền mặt: Thanh toán online không cần mang theo hay sử dụng tiền mặt, nên bạn có thể hạn chế những rủi ro như quên ví, mất ví, cũng như tránh nguy cơ bị trộm cắp tiền.

- Bảo mật thông tin cao: Các hình thức thanh toán điện tử tại ngân hàng, ví điện tử... đều được tích hợp những công nghệ mã hóa như nhận diện khuôn mặt, quét vân tay nên có tính an toàn, bảo mật cao. Bên cạnh đó, khi thanh toán online cho bất kỳ giao dịch nào, bạn sẽ được yêu cầu nhập mã PIN, xác nhận mã OTP... để đảm bảo an toàn.

2.8.3 Các hình thức thanh toán điện tử phổ biến hiện nay

- Thanh toán trực tuyến bằng thẻ: Đây là hình thức thanh toán điện tử cho các hóa đơn mua sắm, đóng tiền điện, nước,... qua số thẻ tín dụng hoặc số thẻ ghi nợ nội địa - quốc tế (VISA, Mastercard, JCB).

- Thanh toán qua ví điện tử: Người đăng ký ví điện tử có thể dùng để nhận, chuyển tiền, mua thẻ điện thoại, vé xem phim, thanh toán trực tuyến các loại phí trên internet như tiền điện nước, cước viễn thông, và mua sắm trực tuyến trên các trang thương mại điện tử

- Thanh toán qua cổng thanh toán điện tử: Đây là hình thức thanh toán thông qua một hệ thống trung gian, kết nối giữa người mua, người bán và ngân hàng để thực hiện các giao dịch. Người dùng cần có tài khoản tín dụng tại ngân hàng. Thông qua cổng thanh toán điện tử, quý khách có thể thực hiện được các giao dịch chuyển nhận tiền giữa bên mua, bán một cách dễ dàng [9].

CHƯƠNG 3: HIỆN THỰC HÓA NGHIÊN CỨU

3.1 Mô tả bài toán

Công ty Hai Lúa Digital, một doanh nghiệp trẻ hoạt động trong lĩnh vực công nghệ thông tin, nhận thấy nhu cầu trao đổi bất động sản tại Trà Vinh ngày càng tăng cao. Tuy nhiên, hầu hết các giao dịch vẫn diễn ra trên các nền tảng truyền thống hoặc thông qua các kênh nhỏ lẻ, dẫn đến khó khăn trong việc tìm kiếm thông tin, xác thực giao dịch, và quản lý các tin đăng. Để đáp ứng nhu cầu thực tiễn và thúc đẩy giao dịch bất động sản tại khu vực, công ty quyết định phát triển một website rao vặt bất động sản chuyên biệt, phục vụ cho cả người mua và người bán.

Mục tiêu chính của dự án là tạo ra một nền tảng trực tuyến hiệu quả, hỗ trợ người dùng dễ dàng đăng tin rao vặt bất động sản, tìm kiếm thông tin theo nhu cầu cụ thể, và kết nối trực tiếp giữa người mua và người bán. Trang web sẽ cung cấp đầy đủ các tính năng cần thiết như quản lý tin đăng, tìm kiếm bất động sản chi tiết, và tích hợp cổng thanh toán cho các dịch vụ thu phí, đảm bảo tính minh bạch và tiện lợi.

Các yêu cầu chính của hệ thống

Quản lý người dùng

Hỗ trợ người dùng cá nhân hoặc doanh nghiệp tạo tài khoản thông qua chức năng Đăng nhập/Đăng ký.

Cho phép người dùng quản lý thông tin cá nhân, đăng tin rao vặt, và theo dõi trạng thái tin đăng.

Đăng tin rao vặt

Người dùng có thể đăng tin bán hoặc cho thuê bất động sản, kèm theo thông tin chi tiết như: loại bất động sản, vị trí, giá tiền, diện tích, và hình ảnh thực tế.

Hệ thống hỗ trợ chỉnh sửa hoặc xóa tin đăng khi cần thiết.

Tìm kiếm và lọc

Cung cấp chức năng tìm kiếm với các tiêu chí: loại bất động sản (nhà, đất nền, căn hộ,...), khu vực, giá tiền, và diện tích.

Cho phép người dùng sắp xếp kết quả theo mức giá, diện tích hoặc thời gian đăng tin.

Thanh toán và quản lý phí dịch vụ

Tích hợp cổng thanh toán để hỗ trợ thu phí đăng tin (nếu có), với các phương thức thanh toán trực tuyến thông dụng.

Quản lý thông tin thanh toán và cung cấp hóa đơn cho người dùng khi cần.

Liên hệ giữa người mua và người bán

Hỗ trợ chức năng liên hệ trực tiếp qua số điện thoại, email, hoặc hộp tin nhắn nội bộ giữa người mua và người bán.

Trang quản trị

Hệ thống quản trị viên có thể giám sát và phê duyệt các tin đăng, quản lý tài khoản người dùng, xử lý các khiếu nại hoặc vi phạm, và thống kê dữ liệu giao dịch trên website.

Yêu cầu phi chức năng

Giao diện website cần hiện đại, thân thiện, và dễ sử dụng, phù hợp với đối tượng người dùng không chuyên về công nghệ.

Đảm bảo hiệu suất hệ thống ổn định khi có nhiều người dùng đồng thời truy cập.

Cung cấp bảo mật dữ liệu, đặc biệt là thông tin tài khoản và các giao dịch thanh toán trực tuyến.

Quy trình làm việc cơ bản

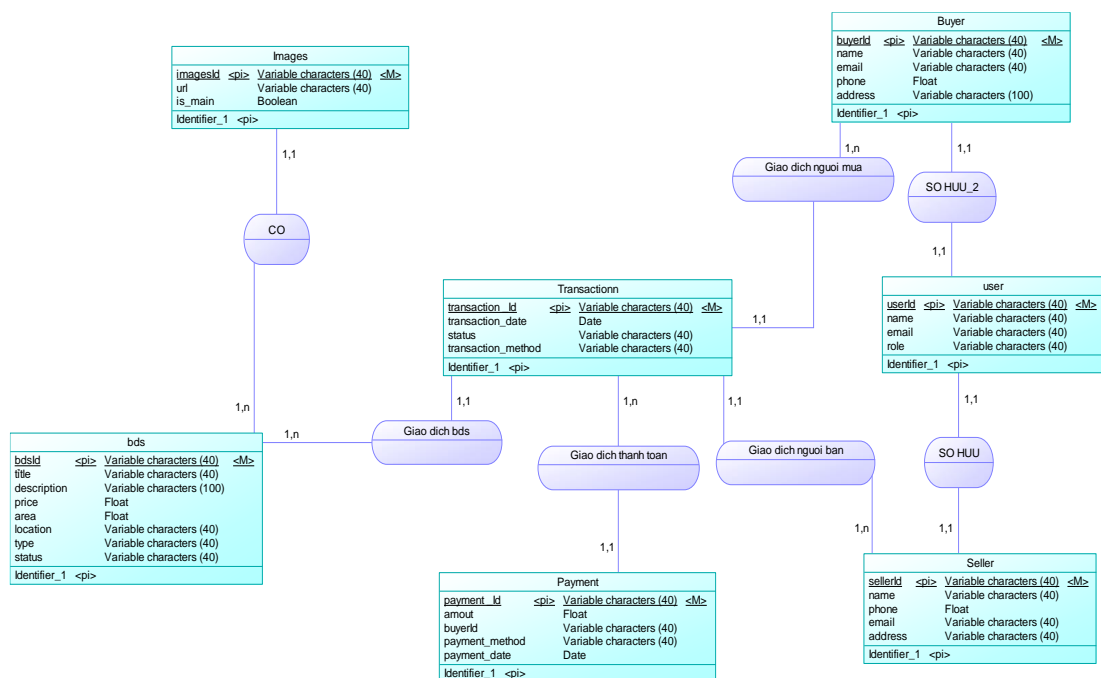
1. Người dùng truy cập website, đăng ký tài khoản hoặc đăng nhập.
2. Người bán đăng tin rao vặt, cập nhật thông tin chi tiết và hình ảnh bất động sản.
3. Người mua sử dụng các tính năng tìm kiếm và lọc để tìm bất động sản phù hợp.
4. Sau khi tìm được tin đăng phù hợp, người mua liên hệ với người bán hoặc thực hiện giao dịch qua nền tảng.

5. Quản trị viên kiểm tra và phê duyệt các tin đăng, giám sát hoạt động trên hệ thống, và xử lý các yêu cầu từ người dùng.

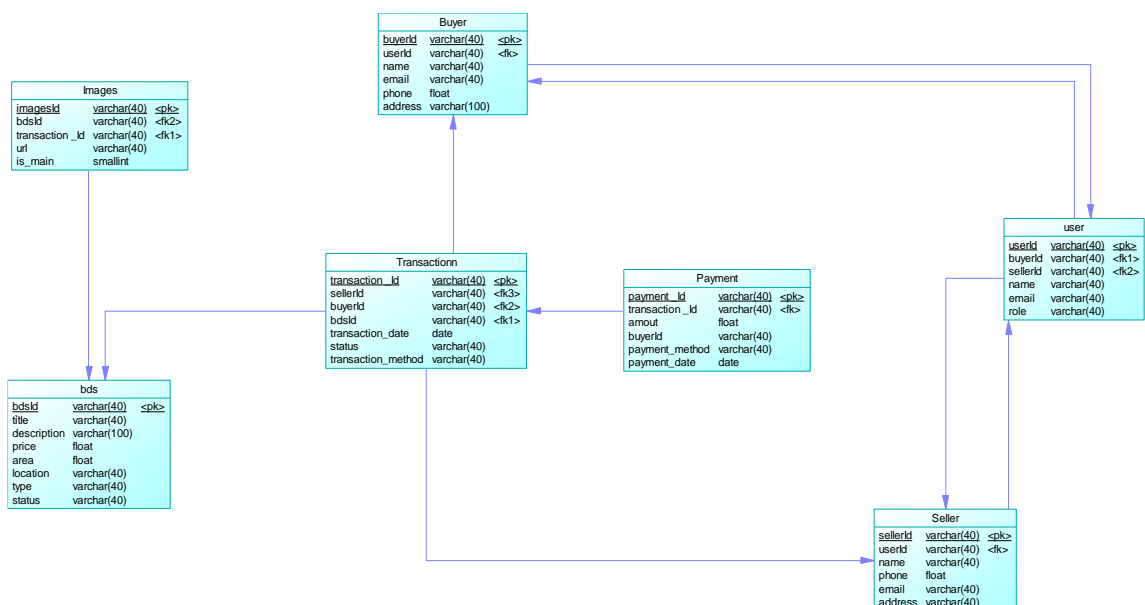
Website rao vặt bất động sản này hứa hẹn sẽ trở thành cầu nối hiệu quả cho các giao dịch nhà đất tại Trà Vinh, góp phần thúc đẩy sự phát triển của thị trường bất động sản trong khu vực.

3.2 Xây dựng website

3.2.1 Mô hình dữ liệu mức quan niệm



3.2.2 Mô hình dữ liệu mức logic



3.2.3 Các thực thể

Các thực thể được xây dựng trong MongoDB:

Bảng 1. Bảng User (bảng người dùng)

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	userId	Id người dùng	String
2	name	Họ và tên người dùng	String
3	sellerId	Id người bán (người cho thuê)	String
4	buyerId	Id người mua (người thuê)	String
5	email	Email của người dùng	String
6	role	Phân quyền của người dùng	String (user, admin)

Bảng 2. Bảng Bất động sản (bds)

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	bdsId	Id bds	String
2	title	Tiêu đề bds	String
3	description	Mô tả bds	String
4	price	Giá bds	Number
5	area	Diện tích bds	Number
6	location	Vị trí bds	String
7	type	Loại bds	String
8	status	Trạng thái bds (đang bán, đang cho thuê, đang thỏa thuận,...)	String

Bảng 3. Bảng Images (ảnh bất động sản)

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	imagesId	Id ảnh	String
2	bdsId	Id bds	String
3	url	Đường dẫn ảnh	String
4	is_main	ảnh chính/ phụ	Boolean

Bảng 4. Bảng Seller (người bán/ người cho thuê)

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	sellerId	Id người bán/cho thuê	String
2	userId	Id người dùng	String
3	name	Tên người bán/ cho thuê	String
4	phone	Số điện thoại người bán/ cho thuê	Number
5	email	Email người bán/ cho thuê	String
6	address	Địa chỉ người bán/ cho thuê	String

Bảng 5. Bảng Buyer (người mua/ người thuê)

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	buyerId	Id người mua/ thuê	String
2	userId	Id người dùng	String
3	name	Tên người mua/ thuê	String
4	phone	Số điện thoại mua/ thuê	Number
5	email	Email người mua/ thuê	String
6	address	Địa chỉ người mua/ thuê	String

Bảng 6. Bảng Transaction (bảng giao dịch)

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	transaction _Id	Id giao dịch	String
2	bdsId	Id bds	String
3	sellerId	Id người bán/ cho thuê	String
4	buyerId	Id người mua/ thuê	String
5	transaction_date	ngày giao dịch	Date
6	status	Trạng thái (hoàn thành/ chưa hoàn thành)	String
7	transaction_method	Loại giao dịch (bán/ cho thuê)	String

Bảng 7. Bảng Payment (thanh toán)

STT	Tên thuộc tính	Mô tả	Kiểu dữ liệu
1	payment _Id	Id thanh toán	String
2	transaction _Id	Id giao dịch	String
3	amout	Số tiền thanh toán	Number

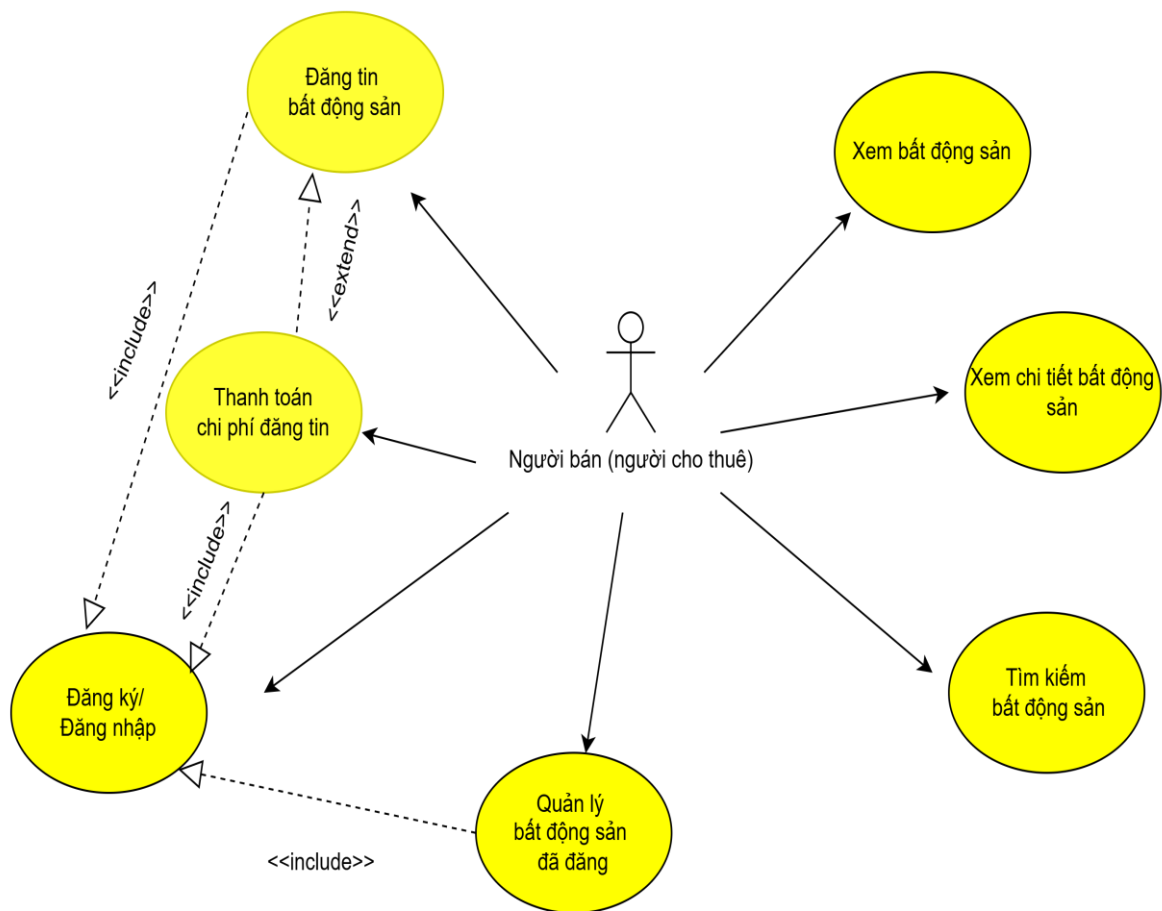
4	buyerId	Id người mua/ thuê	String
5	payment_method	Phương thức thanh toán	String
6	payment_date	Ngày thanh toán	Date

3.3 Thiết kế xử lý

3.3.1 Biểu đồ Use Case tổng quát

...

3.3.2 Biểu đồ Use Case tác nhân người bán (người cho thuê)



Hình 11. Biểu đồ Use Case tác nhân người bán (người cho thuê)

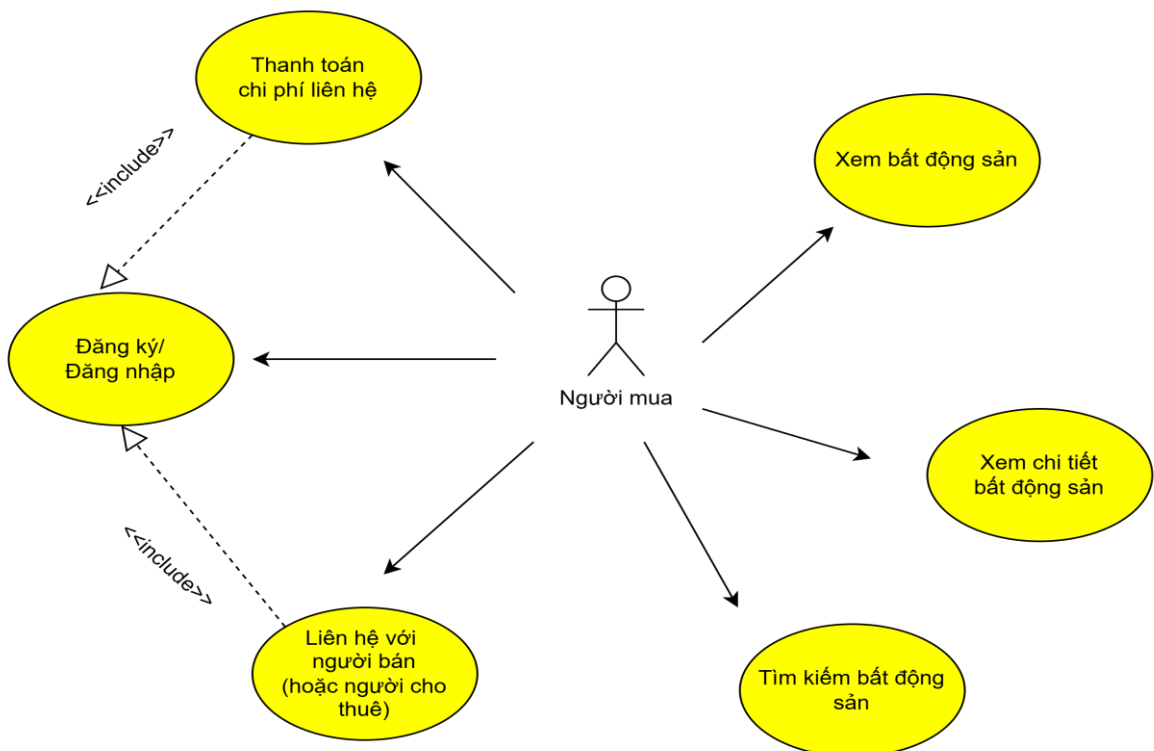
- Khi tác nhân người bán (người cho thuê) truy cập vào hệ thống có thể thực hiện các thao tác như: Xem bất động sản, Xem chi tiết bất động sản, Tìm kiếm. Sau khi đăng nhập (đăng ký nếu chưa có tài khoản), người bán có thể đăng bất động sản miễn phí hoặc muốn được xếp vào loại bất động sản nổi bật thì phải qua thu phí, được phép xem trang quản lý bất động sản mà người bán đã đăng.



Hình 12. Biểu đồ Use Case tác nhân người bán (người cho thuê) đăng tin

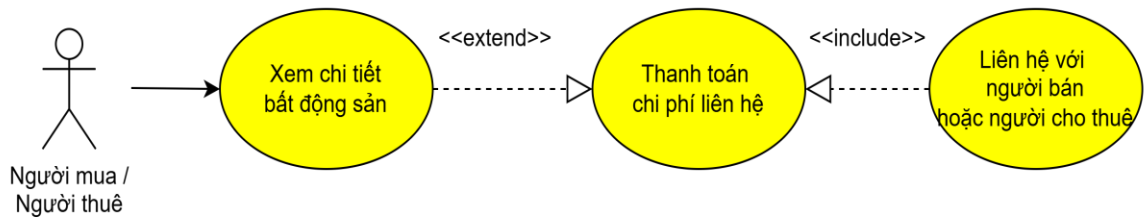
- Người bán (người cho thuê) được phép đăng tin bất động sản miễn phí, hoặc qthu phí nếu muốn bất động sản xếp vào loại nổi bật

3.3.3 Biểu đồ Use Case tác nhân người mua (người thuê)



Hình 13. Biểu đồ Use Case tác nhân người mua (người thuê)

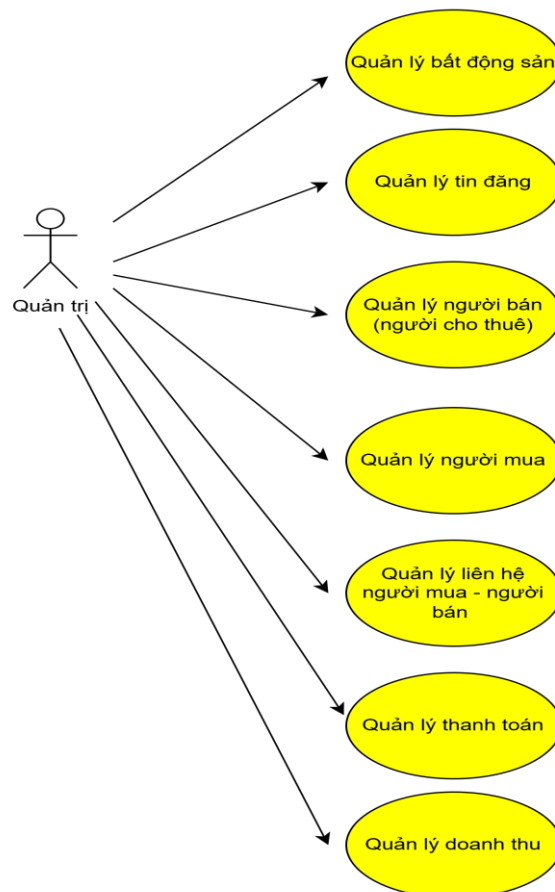
Khi tác nhân người mua (người thuê) truy cập vào hệ thống có thể thực hiện các thao tác như: Xem bất động sản, Xem chi tiết bất động sản, Tìm kiếm. Sau khi đăng nhập (đăng ký nếu chưa có tài khoản), người mua có thể liên hệ với người bán trong trang chi tiết bất động sản, để có được các thông tin liên hệ với người bán (người cho thuê), người mua (người thuê) phải thanh toán phí.



Hình 14. Biểu đồ Use Case liên hệ giữa người mua và người bán

Liên hệ với người bán (người cho thuê) bằng các thông tin như số điện thoại, email, ... Người mua (người thuê) phải thanh toán phí cho doanh nghiệp thông qua chức năng thanh toán trực tuyến.

3.3.4 Biểu đồ Use Case tác nhân quản trị



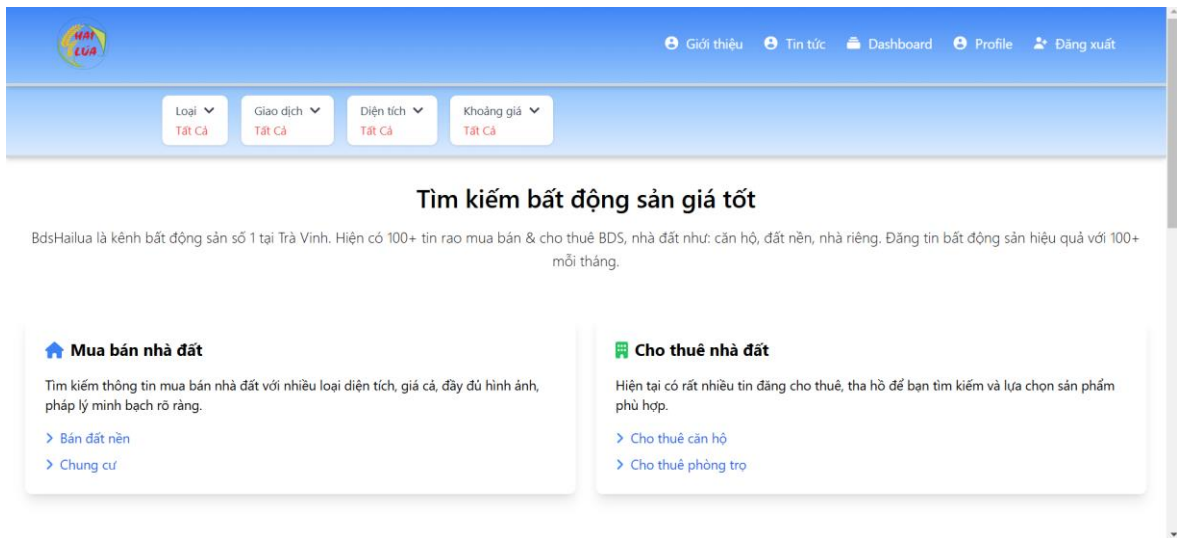
Hình 15. Biểu đồ Use Case tác nhân quản trị

.... Còn chi tiết các Use Case thành phần của tác nhân quản trị:

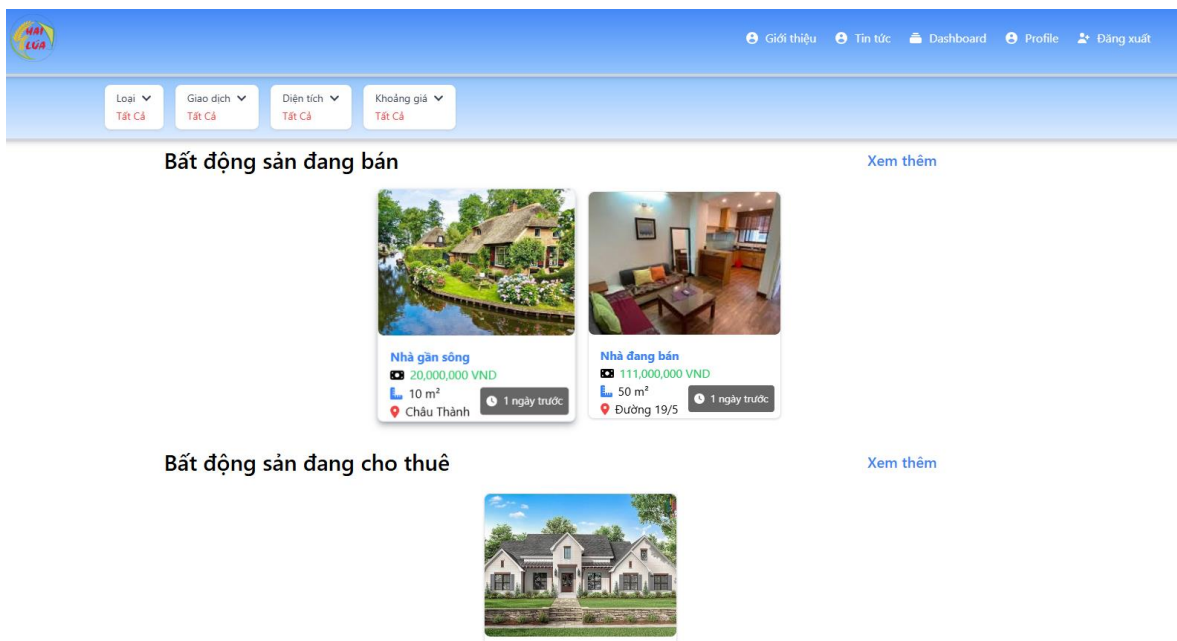
CHƯƠNG 4: KẾT QUẢ NGHIÊN CỨU

4.1 Giao diện người dùng

4.1.1 Giao diện trang chủ



Hình 16. Giao diện trang chủ phần trên



Hình 17. Giao diện trang chủ phần dưới

4.1.2 Giao diện đăng nhập

4.1.3 Giao diện đăng ký

4.1.4 Giao diện trang tìm kiếm bất động sản theo loại

4.1.5 Giao diện chi tiết bất động sản

4.1.6 Giao diện trang quản lý tin đăng (của người bán/ người cho thuê)

4.1.7 Giao diện trang điền thông tin thanh toán

4.1.8 Giao diện trang thanh toán đăng tin bất động sản

4.1.9 Giao diện trang thanh toán chi phí liên hệ giữa người mua và người bán

4.1.10 Giao diện thanh toán qua VNPAY

4.2 Giao diện quản trị

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.2 Hướng phát triển

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] <https://itviec.com/blog/javascript-la-gi/>
- [2] <https://200lab.io/blog/javascript-la-gi/>
- [3] <https://200lab.io/blog/reactjs-la-gi/>
- [4] <https://200lab.io/blog/nodejs-la-gi/>
- [5] <https://200lab.io/blog/expressjs-la-gi/>
- [6] <https://cloud.z.com/vn/news/mvc/>
- [7] <https://200lab.io/blog/huong-dan-cai-dat-tailwind-css-co-ban/>
- [8] <https://200lab.io/blog/mongodb-la-gi>
- [9] <https://www.hlbank.com.vn/vi/personal-banking/blog/huong-dan-dang-ky-thanh-toan-truc-tuyen-tai-ngan-hang-Hong-Leong.html>

PHỤ LỤC