

**Trường Đại Học Mở Hà Nội**

**Khoa Công Nghệ Điện Tử Thông Tin**

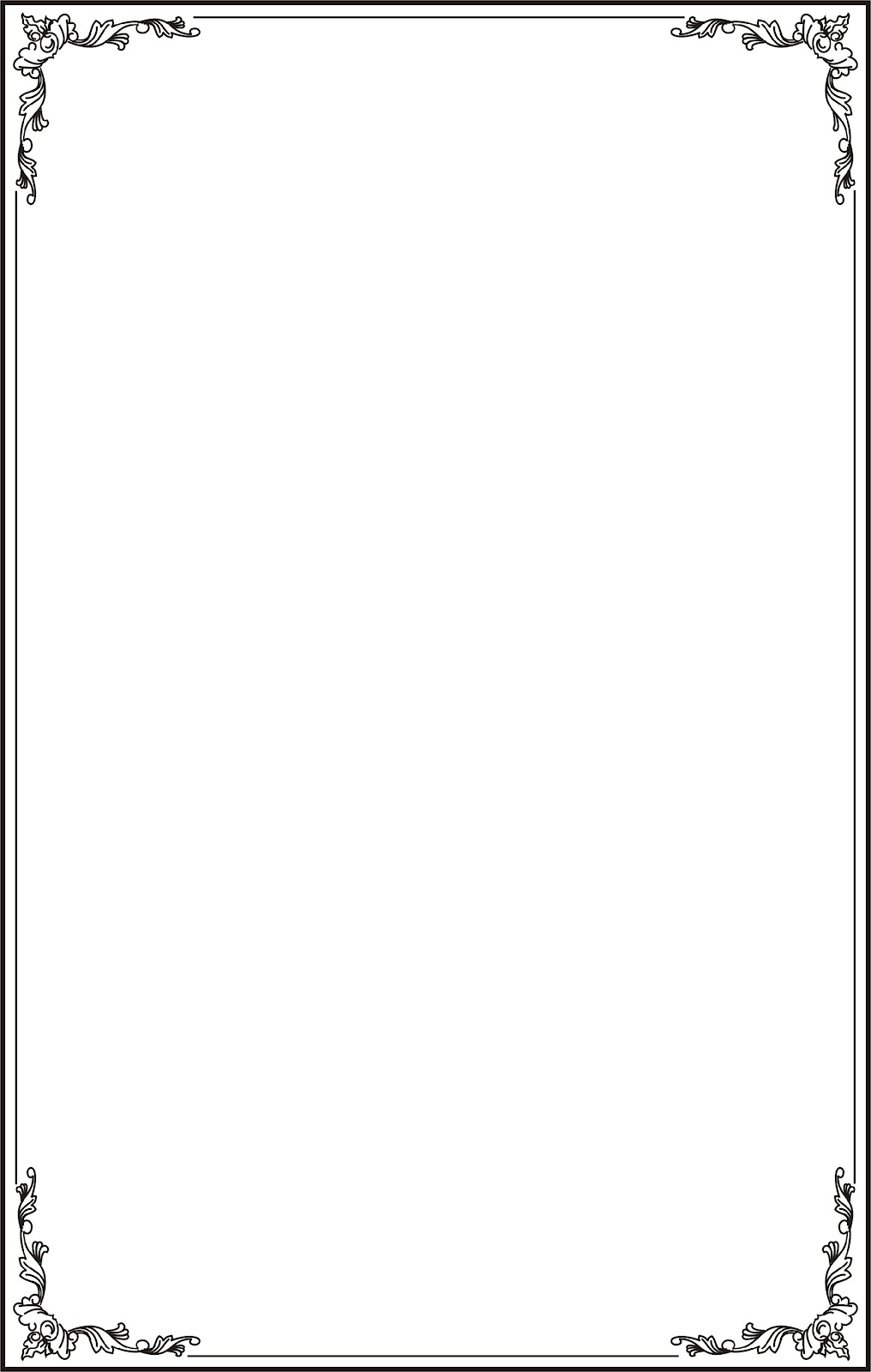


**ĐỒ ÁN TỐT NGHIỆP**

**Đề tài: Thiết kế và xây dựng ứng dụng web hỗ trợ sinh viên tìm kiếm nhà trọ**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn 1:** | **THS. ĐỖ ĐÌNH HƯNG** |
| **Giảng viên hướng dẫn 2:** | **THS. PHẠM TIẾN HUY** |
| **Sinh viên thực hiện :** | **NGÔ BÁ ĐÔNG** |
| **Lớp :** | **K21 – ĐT1** |
| **Khóa :** | **2018 – 2022** |
| **Hệ :** | **CHÍNH QUY** |

**Hà Nội, tháng 5/2022**



**Trường Đại Học Mở Hà Nội**

**Khoa Công Nghệ Điện Tử Thông Tin**



**ĐỒ ÁN TỐT NGHIỆP**

**Đề tài: Thiết kế và xây dựng ứng dụng web hỗ trợ sinh viên tìm kiếm nhà trọ**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn 1:** | **ThS. Đỗ Đình Hưng** |
| **Giảng viên hướng dẫn 2:** | **ThS. Phạm Tiến Huy** |
| **Sinh viên thực hiện :** | **Ngô Bá Đông** |
| **Lớp :** | **K21 – ĐT1** |
| **Khóa :** | **2018 – 2022** |
| **Hệ :** | **Chính Quy** |

**Hà Nội, tháng 5/2022**

|  |  |
| --- | --- |
| Trường Đại Học Mở Hà Nội  **KHOA CN ĐIỆN TỬ - THÔNG TIN** | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  ***Độc lập - Tự do – Hạnh phúc*** |

**ĐỀ TÀI ĐỒ ÁN TỐT NGHIỆP ĐẠI HỌC**

**Họ tên sinh viên:** Ngô Bá Đông

|  |  |  |
| --- | --- | --- |
| **Lớp:** K21 – ĐT1 | **Khóa :** K21 (2018 – 2022) | |
| **Ngành đào tạo:** Công nghệ KT Điện Tử Viễn Thông | | **Hệ đào tạo:** ĐHCQ |

**1. Tên đề tài ĐATN**

Thiết kế và xây dựng ứng dụng web hỗ trợ sinh viên tìm kiếm nhà trọ.

**2. Nội dung chính**

1. Cơ sở lý thuyết

2. Các kỹ thuật lập trình Web

3. Các công nghệ sử dụng

4. Phân tích thiết kế hệ thống ứng dụng

5. Kết luận

**3. Cơ sở dữ liệu ban đầu**

**4. Ngày giao: 03/2022**

**5. Ngày Nộp: 05/2022**

|  |  |
| --- | --- |
| **TRƯỞNG KHOA**  *(Ký, ghi rõ họ tên)* | **GIẢNG VIÊN HƯỚNG DẪN 1**  *(Ký, ghi rõ họ tên)* |
|  |  |
|  | **GIẢNG VIÊN HƯỚNG DẪN 2**  *(Ký, ghi rõ họ tên)* |
|  |  |

**MỞ ĐẦU**

Trong bối cảnh hiện nay, các ứng dụng phần mềm công nghệ thông tin đã được áp dụng trong mọi lĩnh vực, mọi ngành nghề nghiên cứu khoa học kĩ thuật cũng như trong đời sống, máy tính trở thành công cụ trợ giúp đắc lực cho con người trong lưu trữ, phân tích và xử lý thông tin. Ở nước ta, cùng với sự phát triển chung của nền kinh tế, tin học đang dần được ứng dụng rộng rãi trong công tác quản lý và từng bước khẳng định sức mạnh cũng như vị trí quan trọng của mình. Đặc biệt, trong viễn cảnh dịch Covid 19 đang hết sức căng thẳng, mọi công việc đều chuyển qua sang trực tuyến qua mạng internet thì việc áp dụng công nghệ thông tin vào đời sống là cần thiết và cấp bách, đảm bảo an toàn cho mọi người.

Trong ngành công nghệ thông tin nói chung và ngành phần mềm nói riêng thì phần mềm được thiết kế theo nhu cầu sử dụng của người dùng và khách hàng và để phát triển phần mềm thì cần dựa vào các mô hình tiến trình phần mềm như mô hình thác nước, mô hình xoẳn ốc, mô hình tăng trưởng, mô hình agile và mô hình scrum (scrum là một quy trình phát triển phần mềm thuộc họ agile),….

Trong đề tài này với mục đích xây dựng một hệ thống ứng dụng phần mềm chạy trên nền tảng web và nhằm triển khai phương án hỗ trợ sinh viên tìm kiếm nhà trọ trong thời điểm dịch dã căng thẳng và các trường đại học thông báo cho sinh viên quay trở lại trường học khiến nhiều sinh viên ngoại tỉnh đang hối hả, gấp gáp bắt xe khách để lên Hà Nội tìm phòng trọ trước ngày đi học sau gần 1 năm nghỉ dịch ở quê nhà. Chính vì thế đã có rất nhiều kẻ xấu đã lợi dụng việc này để trục lợi cá nhân, đăng bài thuê nhà gian dối để lấy tiền cọc nhà hoặc lừa đảo các tân sinh viên chân ướt chân ráo mới lần đầu lên Hà Nội để nhập học.

Vì vậy, em đã lên ý tưởng và bắt tay vào thực hiện đề tài dự án xây dựng một nền tảng ứng dụng web hỗ trợ sinh viên tìm kiếm nhà trọ, nơi mà mọi sinh viên có thể trao đổi giúp đỡ lẫn nhau về vấn đề tìm nhà trọ, cụ thể là những sinh viên đã có chỗ ở có thể giúp đỡ các sinh viên có hoàn cảnh khó khăn hoặc chưa có chỗ ở trên thành phố.

**LỜI CẢM ƠN**

Trước tiên, em xin gửi lời cảm ơn đến **thầy** **Đỗ Đình Hưng** người đã hướng dẫn em trong quá trình tìm hiểu nghiên cứu và hoàn thành đề tài này từ lý thuyết đến ứng dụng. Và em cũng xin cảm ơn sự hướng dẫn của **thầy** **Phạm Tiến Huy** đã giúp em có thêm những kiến thức về lập trình web, về các kiến thức liên quan đến phát triển phần mềm. Qua những chỉ dẫn ân cần của thầy đã giúp em hiểu sâu hơn về những kiến thức đã được học.

Đồng thời, em cũng xin cảm ơn các thầy cô khoa Công nghệ Điện tử thông tin, cũng như các thầy cô trong trường Đại Học Mở Hà Nội đã hỗ trợ chúng em từ những năm đầu đại học, và đã trang bị những kiến thức cơ bản cần thiết để em hoàn thành tốt đề tài đồ án tốt nghiệp này và tạo cho chúng em một môi trường học tập tốt trong suốt những năm học tập tại Trường.

Ngoài ra, em cũng xin cảm ơn đến tất cả những người bạn đã học, đồng hành cùng em trong suốt năm học tập tại trường, tại khoa đã luôn ở bên cạnh, tạo động lực, giúp đỡ em trong quá trình học tập và tìm hiểu nghiên cứu đề tài đồ án tốt nghiệp này.

Bài báo cáo và ứng dụng có thể còn nhiều hạn chế. Kính mong các thầy góp ý để đề tài của em được hoàn thiện hơn. Cho em những lời khuyên để có thêm kinh nghiệm bước vào tương lai.

**Em xin chân thành cảm ơn !**

**NHẬN XÉT CỦA GIẢNG VIÊN**

………………………………………………………………………………………….

…………………………………………………………………………………………..

…………………………………………………………………………………………...

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………

…………………………………………………………………………………………...

**MỤC LỤC**

[**MỞ ĐẦU** 3](#_Toc102858675)

[**LỜI CẢM ƠN** 4](#_Toc102858676)

[**DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH** 8](#_Toc102858677)

[**KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT** 9](#_Toc102858678)

[**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT** 10](#_Toc102858679)

[1. Tổng quan về đề tài 10](#_Toc102858680)

[2. Phương pháp nghiên cứu 11](#_Toc102858681)

[3. Giới thiệu về ứng dụng Web 12](#_Toc102858682)

[**KẾT LUẬN CHƯƠNG 1** 13](#_Toc102858683)

[**CHƯƠNG 2. KỸ THUẬT LẬP TRÌNH WEB** 14](#_Toc102858684)

[1. Các kỹ thuật lập trình ứng dụng web 14](#_Toc102858685)

[1.1 Kỹ thuật Frontend 14](#_Toc102858686)

[1.2 Kỹ thuật Backend 23](#_Toc102858687)

[1.3 Cơ sở dữ liệu (Database) 31](#_Toc102858688)

[1.4 Công nghệ và ngôn ngữ đã sử dụng 34](#_Toc102858689)

[**KẾT LUẬN CHƯƠNG 2** 40](#_Toc102858690)

[**CHƯƠNG 3: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG ỨNG DỤNG HỖ TRỢ SINH VIÊN TÌM KIẾM NHÀ TRỌ** 41](#_Toc102858691)

[1. Phân tích các yêu cầu của ứng dụng 41](#_Toc102858692)

[1.1 Yêu cầu chức năng 41](#_Toc102858693)

[1.2 Yêu cầu phi chức năng 41](#_Toc102858694)

[1.3 Yêu cầu chức năng khi đăng bài thuê nhà trọ 41](#_Toc102858695)

[1.4 Biểu đồ phân cấp chức năng 42](#_Toc102858696)

[2. Thiết kế hệ thống người dùng usecase 42](#_Toc102858697)

[2.1 Người dùng có tài khoản 42](#_Toc102858698)

[2.2 Quản lý (Người dùng hệ thống) 43](#_Toc102858699)

[3. Thiết kế hệ thống 43](#_Toc102858700)

[3.1 Biểu đồ hoạt động của hệ thống 43](#_Toc102858701)

[3.2 Biểu đồ tuần tự của hệ thống 47](#_Toc102858702)

[3.3 Phân tích hệ thống chức năng ứng dụng hỗ trợ sinh viên tìm kiếm nhà trọ 48](#_Toc102858703)

[3.4 Thiết kế cơ sở dữ liệu 50](#_Toc102858704)

[4. Thiết kế giao diện người dùng 52](#_Toc102858705)

[4.1 Giao diện đăng nhập 52](#_Toc102858706)

[4.2 Giao diện trang chủ 53](#_Toc102858707)

[4.3 Giao diện tìm kiếm phòng 54](#_Toc102858708)

[4.4 Giao diện chi tiết phòng 55](#_Toc102858709)

[4.5 Giao diện khi đăng bài 55](#_Toc102858710)

[4.6 Giao diện bài đăng của người dùng 56](#_Toc102858711)

[4.7 Giao diện danh sách đã lưu của người dùng 57](#_Toc102858712)

[4.8 Giao diện trang quản lý người dùng 57](#_Toc102858713)

[4.9 Giao diện trang quản lý bài đăng 58](#_Toc102858714)

[**KẾT LUẬN CHƯƠNG 3** 59](#_Toc102858715)

[**KẾT LUẬN** 60](#_Toc102858716)

[1. Kết quả đạt được 60](#_Toc102858717)

[2. Về kiến thức 60](#_Toc102858718)

[3. Hạn chế 60](#_Toc102858719)

[4. Hướng phát triển 61](#_Toc102858720)

[**DANH MỤC TÀI LIỆU THAM KHẢO** 62](#_Toc102858721)

**DANH MỤC CÁC BẢNG, SƠ ĐỒ, HÌNH**

[Hình 1: Sơ đồ DOM 21](#_Toc102858647)

[Hình 2 Quá trình hoạt động của web server 25](#_Toc102858648)

[Hình 3: Cấu trúc hoạt động của RESTful API 30](#_Toc102858649)

[Hình 4 Cấu trúc của Redux 35](#_Toc102858650)

[Hình 5: Biểu đồ phân cấp chức năng 42](#_Toc102858651)

[Hình 6: Sơ đồ use case hệ thống tổng quát 43](#_Toc102858652)

[Hình 7: Biểu đồ hoạt động use caser tìm kiếm phòng cùa hệ thống 44](#_Toc102858653)

[Hình 8: Biểu đồ hoạt động use caser đăng tin cho thuê - tìm người ở ghép cùa hệ thống 45](#_Toc102858654)

[Hình 9: Biểu đồ usecase xem danh sách bài đăng đã đăng 46](#_Toc102858655)

[Hình 10: Biểu đồ usecase lưu bài đăng 46](#_Toc102858656)

[Hình 11: Biểu đồ usecase duyệt bài đăng 47](#_Toc102858657)

[Hình 12: Biểu đồ tuần tự use case tìm kiếm phòng của hệ thống 47](#_Toc102858658)

[Hình 13: Biểu đồ tuần tự use case đăng tin phòng cho thuê, tìm người ở ghép của hệ thống 48](#_Toc102858659)

[Hình 14: Sơ đồ thiết kế cơ sở dữ liệu 52](#_Toc102858660)

[Hình 15: Giao diện đăng nhập người dùng 53](#_Toc102858661)

[Hình 16: Giao diện trang chủ 54](#_Toc102858662)

[Hình 17: Giao diện tìm kiếm bài đăng thuê phòng 54](#_Toc102858663)

[Hình 18: Giao diện chi tiết bài đăng thuê phòng 55](#_Toc102858664)

[Hình 19: Giao diện đăng bài thuê phòng trọ 56](#_Toc102858665)

[Hình 20: Giao diện bài đăng của tôi 57](#_Toc102858666)

[Hình 21: Giao diện Bài đăng đã lưu 57](#_Toc102858667)

[Hình 22: Giao diện trang quản lý người dùng 58](#_Toc102858668)

[Hình 23: Giao diện quản lý bài đăng 58](#_Toc102858669)

[Bảng 1: Sự khác nhau giữa website và webapp 13](#_Toc102858670)

[Bảng 2: Bảng hệ thống chức năng ứng dụng 49](#_Toc102858671)

[Bảng 3: Bảng CSDL người dùng 50](#_Toc102858672)

[Bảng 4: Bảng CSDL phòng trọ 51](#_Toc102858673)

[Bảng 5: Bảng CSDL danh sách lưu các bài đăng 51](#_Toc102858674)

**KÝ HIỆU CÁC CỤM TỪ VIẾT TẮT**

UI/UX: User Interface/User Experient (Giao diện người dùng / Trải nghiệm người dùng)

API: Application Programming Interface (Giao diện lập trình ứng dụng)

CSDL: Cơ sở dữ liệu

HTML: HyperText Markup Language

CSS: Cascading Style Sheets

IP: Internet Protocol

HTTP: HyperText Transfer Protocol

HTTPS: HyperText Transfer Protocol Security

TCP: Transmission Control Protocol

URL: Uniform Resource Locator

SQL: Structured Query Language

DOM: Document Object Model

SSL: Secure Sockets Layer

DNS: Domain Name System

**CHƯƠNG 1: CƠ SỞ LÝ THUYẾT**

1. Tổng quan về đề tài
   1. Mục đích của đề tài

Vấn đề tìm phòng trọ không còn quá mới mẻ với những bạn sinh viên nữa vì bạn nào cũng ít nhất một lần trong đời sinh viên đều phải trải qua rơi vào tình huống này và hiểu rõ những gian nan và vất vả trong chuyện tìm và chuyển phòng trọ cho phù hợp với chuyện học hành, làm thêm và đặc biệt là những bạn tân sinh viên càng khó khăn hơn rất nhiều. Để tìm được một nơi an ninh tốt, vệ sinh sạch sẽ và giá cả hợp lý và địa điểm thuận tiện cho việc đi học và làm thêm là một điều rất khó đối với những sinh viên, tân sinh viên mới chập chững bước chân lên Hà Nội.

Ngoài việc đăng kí ở kí túc xá thì tìm kiếm một phòng trọ của sinh viên cũng là một nhu cầu cấp thiết khi bước vào môi trường học xa nhà khi lên các thành phố lớn. Tuy nhiên việc tìm một phòng trọ theo cách thủ công như tìm thông tin trên mạng xã hội, xem các tin được dán giấy ở nhiều cây cột điện còn nhiều hạn chế và bất cập. Hoặc đăng tin cho thuê trọ của các chủ trọ theo cách thủ công cần phải phát tờ rơi hoặc dán giấy tờ rơi khắp nơi rất phản cảm và mất mĩ quang đô thị.

Việc tạo ra một nền tảng hỗ trợ sinh viên tìm kiếm nhà trọ trên nền tảng ứng dụng web là hợp lý và tiện lợi trong thời kỳ công nghệ 4.0. Các sinh viên chỉ cần ngồi ở một nơi có đầy đủ mạng internet và có thể truy cập vào trình duyệt web là có thể tìm kiếm phòng trọ thật dễ dàng và nhanh chóng, trao đổi trực tiếp online với chủ nhà trọ trên các nền tảng chat miễn phí hiện nay hay thậm chí có thể trò chuyện trực tiếp với người đăng tin và chốt hạ ngày xem phòng ngay chính trên trang web.

* 1. Lí do chọn đề tài

Từ mục đích đề tài được nêu ở phía trên và hiểu được nỗi vất vả tìm phòng trọ nên em đã quyết định làm ứng dụng web để giúp đỡ những bạn sinh viên chưa tìm được phòng trọ phù hợp với chuyện học hành và làm thêm, muốn ổn định chỗ ở nhanh để phục vụ cho việc học tập, sinh hoạt. Và đề tài được đặt ra mục tiêu hướng đến mục đích cộng đồng là các bạn sinh viên của các trường đại học, cao đẳng trên Hà Nội.

Vì vậy dựa trên khảo sát thực tế em đã bắt đầu nghiên cứu thực hiện đề tài **Thiết kế và xây dựng ứng dụng web hỗ trợ sinh viên tìm kiếm nhà trọ**.

Lý do em thực hiện đồ án trên giao diện web vì trang web có thể hoạt động ở mọi nền tảng một cách dễ dàng, ở bất kì đâu, bất kì thời điểm chỉ cần bạn có máy tính hoặc smartphone hoặc bất cứ thiết bị nào có thể kết nối internet là có thể truy cập vào trang web và tìm kiếm phòng trọ.

* 1. Khảo sát và tìm hiểu nhu cầu

Theo khảo sát thực tế khi tốc độ đô thị hóa, dân số tăng cao, lượng sinh viên dồn về các thành phố lớn để học tập và làm việc ngày càng đông đảo, làm cho nhu cầu tìm kiếm chỗ ở phòng trọ luôn luôn trong tình trạng cấp thiết và nóng hổi.

Trước hết sẽ khảo sát tình hình chung của các trường đại học tại các thành phố lớn thì hầu như 80% - 90% sinh viên đang theo học tại các trường Đại học, Cao đẳng đều ở các phòng trọ cho thuê hoặc sinh viên nào gia đình khá giả hơn thì ở chung cư. Còn lại sinh viên sẽ ở các khu kí túc xá, đa số kí túc xá được xây dựng công lập nên số lượng chỗ ở cung cấp cho sinh viên còn hạn chế. Và cơ sở vật chất còn hạn hẹp, chưa đảm bảo chất lượng cho sinh viên sinh hoạt và học tập.

Sau khi đã nghiên cứu thị trường và phỏng vấn vài sinh viên ở một số trường đại học thì việc tìm hiểu nhu cầu ở phòng trọ của các bạn sinh viên, hầu hết đa số các bạn đều cho rằng việc cần tìm kiếm một phòng trọ là điều quan trong nhất khi nhập học tại các trường. Tuy nhiên việc tìm phòng trọ gặp rất nhiều khó khăn do thông tin và tra cứu còn hạn chế, không có công cụ chuyên nghiệp để hỗ trợ vì vậy, việc cần một ứng dụng cung cấp thông tin và hỗ trợ tìm kiếm và đăng tin tìm người ở ghép là một điều thực sự cần thiết cho các bạn sinh viên ngay lúc này.

1. Phương pháp nghiên cứu

Dựa trên khảo sát thực tế và tìm hiểu nhu cầu của sinh viên mỗi khi tìm nhà trọ, em đã đề xuất và thực hiện phương pháp nghiên cứu thực hiện đề tài.

Sau khi nghiên cứu về nhu cầu sinh viên tìm kiếm thông tin các phòng trọ quan tâm đến các tiêu chí như thế nào trước khi quyết định.

Các tiêu chí chọn phòng trọ của sinh viên thường sẽ ưu tiên theo thứ tự vị trí, giá cả, diện tích, và cùng các tiện nghi cơ bản tối thiểu như nhà xe, máy giặt, bếp nấu ăn. Hoặc sinh viên đăng tin tìm người cần ở ghép để có thể chia sẻ tiền chi phí phòng ở tạo dựng mối quan hệ. Đối với người đăng tin thuê phòng cũng cần chú ý các thông tin cơ bản chính xác và đầy đủ nhất cho người cần thuê.

1. Giới thiệu về ứng dụng Web

Trong [kỹ thuật phần mềm](https://vi.wikipedia.org/wiki/K%E1%BB%B9_thu%E1%BA%ADt_ph%E1%BA%A7n_m%E1%BB%81m" \o "Kỹ thuật phần mềm), một ứng dụng web hay webapp là một [ứng dụng](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_%E1%BB%A9ng_d%E1%BB%A5ng" \o "Phần mềm ứng dụng) mà có thể tiếp cận qua web thông qua mạng như [Internet](https://vi.wikipedia.org/wiki/Internet) hay [intranet](https://vi.wikipedia.org/wiki/Intranet). Ứng dụng web được phổ biến nhờ vào sự có mặt ở bất cứ nơi đâu của một chương trình và truy cập qua trình duyệt web. Khả năng cập nhật và bảo trì ứng dụng web mà không phải phân phối và cài đặt phần mềm trên hàng ngàn máy tính đấy là lý do chính cho sự phổ biến của nó.

Đơn giản hơn thì ứng dụng web là những ứng dụng đa nền tảng chạy trên web, thông qua ứng dụng web thì người dùng có thể tương tác với hệ thống và yêu cầu thực hiện một công việc gì đó. Tuy nhiên có nhiều người không rành về công nghệ, tất cả những thứ online đều vào được bằng trình duyệt đều là website cả. Do đó họ thường yêu cầu là: website quản lý siêu thị, website bán hàng, … thực chất chúng đều là ứng dụng web hết. Vậy ứng dụng web (web app) và website khác nhau ở điểm gì mà có thể nhầm lẫn như vậy. Đầu tiên em sẽ nói qua khái niệm về website.

Khi Internet còn thô sơ, web được viết bằng html đơn lẻ. Mỗi trang web đơn lẻ được viết bằng html gọi là **web page**. Tập hợp nhiều trang web đơn lẻ, thành một trang web lớn, có chung tên miền, được gọi là **website**. Trên thực tế, ranh giới giữa web app và website khá mong manh. Một trang báo mạng – vnexpress chẳng hạn, trong mắt người đọc nó là website. Nhưng trong mắt biên tập viên hoặc admin, nó lại là web app. Một số trang web cho phép người dùng tìm kiếm, bình luận nhưng nó vẫn chỉ là website, chưa phải là webapp. Dưới đây là bảng so sánh giữa ứng dụng web và website.

|  |  |
| --- | --- |
| WEBSITE | Ứng dụng Web (WebApp) |
| Tính tương tác thấp, ít chức năng (Xem, đọc, click qua lại giữa các link…) | Tính tương tác cao, nhiều chức năng (Đăng thông tin, upload file, xuất báo cáo…) |
| Được tạo thành từ các trang html tĩnh và một số tài nguyên (hình ảnh, âm thanh, video) | Được tạo bởi html và code ở back end (PHP, [C#](https://toidicodedao.com/tag/c-hay-ho/), Java, …) |
| Được dùng để lưu trữ, hiển thị thông tin | Được dùng để “thực hiện một công việc”, thực hiện các chức năng của một ứng dụng |

Bảng 1: Sự khác nhau giữa website và webapp

Như vậy, qua bảng so sánh ở trên đã hiểu qua về sự khác nhau giữa website và web app và cách xây dựng một ứng dụng thì cần những kiến thức gì và các kỹ thuật lập trình gì mà lập trình viên web cần phải thực hiện chúng trước khi bắt tay vào làm thì chúng ta hãy tiếp tục phần dưới để hiểu rõ hơn về cách xây dựng một ứng dụng web.

**KẾT LUẬN CHƯƠNG 1**

Chương này đã giới thiệu tổng quan về đề tài. Giúp hiểu rõ hơn lý  
do chọn đề tài dựa trên nhu cầu thực tế của xã hội cũng như nhu cầu sử dụng hệ thống  
tìm phòng trọ của các bạn sinh viên. Tìm hiểu rõ và xác định được các yêu cầu của hệ thống, đưa ra được các thông tin như:

* Đối tượng sử dụng ứng dụng.
* Các thông tin cụ thể được đưa ra trong ứng dụng.
* Các chức năng có thể có của hệ thống và nghiệp vụ của hệ thống.

Các chương tiếp theo sẽ đi sâu vào quy trình phát triển phần mềm để có thể hiểu  
rõ hơn về ứng dụng web hỗ trợ sinh viên tìm kiếm nhà trọ.

**CHƯƠNG 2. KỸ THUẬT LẬP TRÌNH WEB**

1. Các kỹ thuật lập trình ứng dụng web
   1. Kỹ thuật Frontend

Đây là kỹ thuật chủ yếu xử lý về giao diện người dùng và trải nghiệm người dùng (hay còn gọi là UI/UX), về giao diện web trên các thiết bị (máy tính, điện thoại, máy tính bảng,….). Front-end đơn giản là việc tính toán cài đặt font chữ, màu sắc, các menu xổ xuống và các thanh trượt bằng cách sử dụng kết hợp HTML, CSS, và JavaScript. Nhưng Front-end hiện đại đòi hỏi cao hơn, cần tính đến yếu tố tương tác với người dùng.

Để có thể nắm vững và thành thạo kỹ thuật Frontend thì chúng ta cần phải biết cơ bản và nâng cao về HTML, CSS và Javascript.

Trước tiên, em sẽ nói về HTML, **HTML** là viết tắt của cụm từ **Hypertext Markup Language**(tạm dịch là Ngôn ngữ đánh dấu [siêu văn bản](https://vi.wikipedia.org/wiki/Si%C3%AAu_v%C4%83n_b%E1%BA%A3n" \t "_blank" \o "Siêu văn bản - Wikipedia)). **HTML** được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes… và **HTML** không phải là ngôn ngữ lập trình.

Một tài liệu HTML được hình thành bởi các phần tử HTML được quy định bằng các cặp thẻ. Các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn. Ví dụ <html> và thường là sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng. Ví dụ:

<html></html> Cha đẻ của HTML là **Tim Berners-Lee**, cũng là người khai sinh ra World Wide Web và chủ tịch của **World Wide Web Consortium** (W3C – tổ chức thiết lập ra các chuẩn trên môi trường Internet).

Để hiểu HTML một cách rõ ràng và đơn giản, có thể xem qua một đoạn code HTML mẫu dưới đây.

<!DOCTYPE html>

<html>

<head>

<title>HTML Example</title>

</head>

<body>

<h1>Code HTML Example</h1>

<p>

<a href="https://www.freecodecamp.org/contribute/">Hello World </a>

shows you how and where you can contribute to freeCodeCamp's community and growth.

</p>

<a href="#">This is a dead link</a>

<a href="https://www.freecodecamp.org">

This is a live link to freeCodeCamp

</a>

<a href="https://html.com/attributes/a-href/">

more with a href attribute

</a>

</body>

</html>

Một trang web chỉ có mỗi HTML thôi thì chưa đủ hấp dẫn và lôi cuốn người dùng. Cứ tưởng tượng chúng ta đang xây một ngôi nhà vậy, HTML được ví như bộ khung, các viên gạch để xây nền móng và khung xương cho ngôi nhà thì CSS nó sẽ giúp trang trí, sơn màu cho ngôi nhà của bạn đẹp hơn, hấp dẫn hơn và sẽ có nhiều người vây quanh. Trong lập trình web nó cũng tương tự như vậy, HTML là bộ khung của trang web thì CSS làm cho trang web đẹp hơn, lunh linh, hấp dẫn hơn.

Vậy CSS là gì ?

**CSS** là chữ viết tắt của Cascading Style Sheets, nó là một ngôn ngữ được sử dụng để **tìm và định dạng** lại các phần tử được tạo ra bởi [HTML](https://topdev.vn/blog/html-la-gi/). Nói ngắn gọn hơn là ngôn ngữ tạo kiểu phong cách cho trang web. Bạn có thể hiểu đơn giản rằng, nếu HTML đóng vai trò định dạng các phần tử trên website như việc tạo ra các đoạn văn bản, các tiêu đề, bảng,… thì CSS sẽ giúp chúng ta có thể thêm style vào các phần tử HTML đó như đổi bố cục, màu sắc trang, đổi màu chữ, font chữ, thay đổi cấu trúc…

body {

background-color: blue;

background-image: url("barn.jpg");

}

h1 {

background–color: purple;

}

p {

font-size: 12px;

font-weight: 500;

}

Cuối cùng, Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa. Javascript được sử dụng rộng rãi trong các ứng dụng web, mobile. Javascript được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome, ... trên máy tính lẫn điện thoại.

Ví dụ code Javascript:

function getPower(decimalPlaces) {

return 10 \* decimalPlaces;

}

function capitalize(word) {

return word[0].toUpperCase() + word.slice(1);

}

function roundToDecimalPlace(number, decimalPlaces = 2) {

const round = getPower(decimalPlaces);

return Math.round(number \* round) / round;

}

Nhiệm vụ của Javascript là xử lý những đối tượng HTML trên trình duyệt. Nó có thể can thiệp với các hành động như thêm / xóa / sửa các thuộc tính CSS và các thẻ HTML một cách dễ dàng và xử lý phần tương tác người dùng với hệ thống ứng dụng. Hay nói cách khác, Javascript là một ngôn ngữ lập trình trên trình duyệt ở phía máy khách (client). Tuy nhiên, với sự xuất hiện của NodeJS đã giúp cho Javascript có thể làm việc ở phía máy chủ server.

Từ năm 2015 thì javascript ES6 được ra đời với nhiều tính năng mới học hỏi các ngôn ngữ cấp cao khác, hy vọng dần theo thời gian Javascript trở thành một ngôn ngữ lập trình hướng đối tượng.

Các chức năng mới của ES6

* Arrow function: Có thể viết tạo hàm bằng cách sử dụng dấu mũi tên =>.

const funcSample = () => { console.log(‘hello’) }

funcSample();

* Block Scoped: Định nghĩa biến với từ khóa let, cách định nghĩa này thì biến chỉ tồn tại trong phạm vi khối của nó (Block Scope), là phạm vi trong một khối, nghĩa là chỉ hoạt động trong phạm vi được khai báo bởi cặp {}.

for (let i = 0; i < 5; i++) {

setTimeout(function(){

console.log('hello ', i);

}, 1000);

}

* Destructuring Assignments: Có thể khởi tạo các biến từ một mảng hoặc một object bằng một dòng code đơn giản.

const x = [1, 2, 3, 4, 5];

const [y, z] = x;

console.log(y); // 1

console.log(z); // 2

* Default Parameters: Có thể gán giá trị mặc định cho các tham số.

function multiply(a, b = 1) {

return a \* b

}

multiply(5, 2) // 10

multiply(5) // 5

multiply(5, undefined) // 5

* Template Literals (template string)

let a = 5;

let b = 10;

console.log('Fifteen is ' + (a + b) + ' and\nnot ' + (2 \* a + b) + '.');

// "Fifteen is 15 and

// not 20."

* Weak, Set: các kiểu dữ liệu phức tạp mới
* Modules: Mọi thứ bên trong một ứng dụng JavaScript, ví dụ như các biến trên các tệp JavaScript khác nhau, đều có chung một phạm vi. ES6 giới thiệu tệp dựa trên module, trong đó mỗi module được biểu diễn bằng một tệp .js riêng biệt. Bây giờ, bạn có thể sử dụng câu lệnh export hoặc import trong một module để xuất hoặc nhập các biến, hàm, class hoặc bất kỳ thực thể nào khác đến / từ các module hoặc tệp khác.

// file main.js

let greet = "Xin chào ES6!";

const PI = 3.14;

function multiplyNumbers(a, b) {

    return a \* b;

}

// Xuất các biến và hàm

export { greet, PI, multiplyNumbers };

// file App.js

import { greet, PI, multiplyNumbers } from './main.js';

alert(greet); //Xin chào ES6!

alert(PI); // 3.14

alert(multiplyNumbers(2, 3)); // 6

Hiện nay, với sự phát triển của Javascript mà các lập trình viên đã phát triển ra những framework frontend được viết bởi chính ngôn ngữ javascript để cho việc code frontend dễ dàng hơn và tiết kiệm thời gian hơn. Em sẽ giới thiệu sơ qua các framework frontend phổ biến nhất hiện nay

* Angular: một framework rất nổi tiếng trong frontend dùng để phát triển những ứng dụng SPA và được xây dựng bởi Google và là phiên bản sau của Angularjs
* Vuejs là framework xây dựng giao diện người dùng và được xây dựng bởi nhiều lập trình viên trên toàn thế giới.
* ReactJs là thư viện javascript dùng để xây dựng giao diện người dùng, là 1 thư viện phổ biến nhất trong hiện nay và được phát triển bởi Meta (Facebook)

Ngoài một số framework frontend được nêu phía trên thì còn có rất nhiều framework hay thư viện frontend khác mà em chưa liệt kê ở đây. Ví dụ như: Jquery,….

Trong đề tài đồ án này, em sử dụng thư viện React – một thư viện Frontend phổ biến nhất trong giới lập trình web hiện nay với xu hướng ứng dụng SPA (Single Page Application). React là thư viện Javascript dùng để xây dựng giao diện người dùng.

Nói qua về thư viện **React**, đây là một thư viện được phát triển bởi đội ngũ kỹ sư phần mềm Meta (Facebook) và giúp xây dựng, hiển thị giao diện người dùng một cách tối ưu nhất và tư duy thiết kế giao diện theo component. React được chia làm các thư viện con: **React Core** và **ReactDOM** và còn thư viện con nữa dùng để phát triển môi trường ứng dụng di động android, ios bằng Javascript là **React Native**.

**React Core** cung cấp thư viện bao gồm các hàm và class để tạo ra component và quản lý state của từng component. Bất kỳ nền tảng nào chạy được JavaScript đều có thể sử dụng package “react”.

**ReactDOM** là thư viện bổ sung cho “react”, là cầu nối giữa React và DOM, thao tác với DOM để hiển thị (render) các component một cách tối ưu nhất có thể, dựa vào thuật toán DOM ảo (virtual DOM, thường gọi là vDOM). Khi các component xảy ra thay đổi, tất cả những thay đổi ấy sẽ được render lên vDOM, sau đó cây vDOM sẽ được so sánh với cây DOM thật của trình duyệt, đảm bảo chỉ những node nào thật sự khác biệt mới được cập nhật lên DOM thật. Thuật toán này giúp cho giao diện phức tạp vẫn giữ được sự mượt mà, tránh giật lag khi có nhiều component được cập nhật.

Nhắc về DOM, em sẽ giải thích thêm về khái niệm DOM là gì và tại sao một trang web cần có DOM? DOM là viết tắt của cụm từ Document Object Model là một giao diện lập trình ứng dụng. DOM được dùng để truy xuất các tài liệu dạng HTML và XML, có dạng một cây cấu trúc dữ liệu, và thông thường mô hình DOM độc lập với hệ điều hành và dựa theo kỹ thuật lập trình hướng đối tượng để mô tả tài liệu.



Hình 1: Sơ đồ DOM

Như hình trên, Có thể thấy tất cả các thẻ HTML sẽ được quản lý trong đối tượng document (DOM) và mỗi một ô vuông như Root element, element, Attribute, Text thì nó gọi là node, node cao nhất là root node chứa thẻ html, tiếp đến là phân nhánh thành 2 node element chứa head và body. Bên trong head thì có những node element chứa title và trong title thì có node text chứa một đoạn chữ,... và bên trong body chứa bất kì một node nào đó là thành phần của HTML.

Như vậy, có thể hiểu trong Javascript để thao tác được với các thẻ HTML thì phải thông qua đối tượng document là DOM. Và sơ đồ cây DOM như hình trên là tiêu chuẩn được World Wide Web Consortium (W3C) thống nhất mô hình này.

Tiếp theo, cần tìm hiểu xem DOM ảo là gì mà React lại làm việc qua DOM ảo mà không phải làm trực tiếp với DOM thật.

**DOM** ảo không được tạo ra bởi React tuy nhiên nó được React sử dụng và cung cấp miễn phí. Một cách tổng quát thì nó là một định dạng dữ liệu JavaScript nhẹ được dùng để thể hiện nội dung của DOM tại một thời điểm nhất định nào đó. Nó có tất cả các thuộc tính giống như **DOM thật** nhưng không có khả năng tương tác lên màn hình như **DOM**. Ở **DOM** có thẻ div và các thẻ p ở trong, ReactJs sử dụng **DOM ảo** bằng cách tạo ra các object React.div và React.p và khi tương tác, sẽ tương tác qua các object đó một cách nhanh chóng mà không phải đụng tới **DOM** hay **DOM API** của nó.

**Bên cạnh đó, thì React còn có những ưu điểm và nhược điểm.**

**Ưu điểm:**

* Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật len DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
* Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dung cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể them vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
* Có nhiều công cụ phát triển
* Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
* Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

Nhược điểm:

* Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax.
* Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
* React khá nặng nếu so với các framework khác React có kích thước tương tương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh.
* Khó tiếp cận cho người mới học Web.
  1. Kỹ thuật Backend

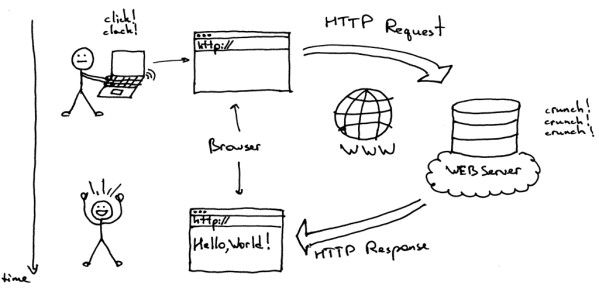
Kỹ thuật Backend xây dựng và duy trì công nghệ hay xử lý logic phía đằng sau của một ứng dụng, trang web có thể hoạt động và triển khai đến người dùng được. Để khiến cho máy chủ, ứng dụng, và cơ sở dữ liệu có thể giao tiếp được với nhau, các lập trình viên back-end sử dụng các ngôn ngữ lập trình server-side như PHP, NodeJs, Python, Java, và .Net để xây dựng một ứng dụng, và các công cụ như MySQL, SQL Server và MongoDB để tìm kiếm, lưu trữ, hoặc thay đổi dữ liệu của ứng dụng và chuyển dữ liệu đó cho phần Frontend để hiển thị ra giao diện cho người dùng.

Kỹ thuật Backend là sẽ phải làm việc với máy chủ và cơ sở dữ liệu nhiều hơn. Trước tiên sẽ tìm hiểu về web server để hiểu rõ hơn cách hoạt động như thế nào.

Web server hay còn gọi là máy chủ web, trong đó được kết nối và liên kết mạng máy tính mở rộng. Máy chủ web được cài đặt các chương trình để phục vụ ứng dụng web, chứa toàn bộ dữ liệu và nắm quyền quản lý. Web server có thể lấy thông tin yêu cầu từ phía trình duyệt web và gửi phản hồi về máy khách thông qua giao thức HTTP hoặc HTTPS hoặc giao thức khác. Về phần cứng, Web server là một máy tính được lưu trữ các thư mục thành phần để tạo nên một website (ví dụ: file HTML, các hình ảnh, file CSS, và file javacript...) và nhiệm vụ trả những file đó về trình duyệt web qua giao thức mạng và trình duyệt web biên dịch mã HTML và tạo ra cây DOM thì từ cây DOM trình duyệt web có thể dựa vào những các node trong DOM để hiển thị giao diện ra màn hình cho người dùng. Web server được kết nối đến internet và truy cập thông qua một tên miền.

Về phần mềm thì máy chủ web bao gồm một số phần kiểm soát người dùng web truy cập đến file host tại tối thiểu một HTTP server. Một HTTP server là một phần của phần mềm nó hiểu là URL và HTTP (là phương thức để trình duyệt của bạn hiển thị trang web). Ở mức cơ bản nhất, bất cứ một trình duyệt nào cần một file host trên một web server, trình duyệt đó sẽ gửi một yêu cầu thông qua HTTP hoặc HTTPS. Khi một yêu cầu được gửi đến đúng địa chỉ web server và web server nhận được yêu cầu đó, và xử lý trả về những gì từ yêu cầu đó thì HTTP server gửi trả lại một phản hồi thông qua HTTP hoặc HTTPS.

Vậy web server hoạt động như thế nào ?



Hình 2 Quá trình hoạt động của web server

Hình trên là quá trình hoạt động tổng quát của web server. Sau đây em sẽ trình bày chi tiết về hoạt động của web server

Khi người dùng truy cập vào một trang web thì sẽ phải nhập tên miền trang web đó. Lúc này trình duyệt có nhiệm vụ sẽ phân giải tên miền đó thành địa chỉ IP và tên miền đó trỏ về. Trình duyệt sẽ yêu cầu thông tin từ một hoặc nhiều máy chủ DNS (thông qua internet). Máy chủ DNS sẽ cho trình duyệt biết địa chỉ IP nào tên miền sẽ trỏ đến cũng là nơi đặt trang web. Sau khi web server nhận được yêu cầu từ trình duyệt gửi lên, nó sẽ phản hồi những thông tin mà bên trình duyệt (client) yêu cầu. Nếu như trang không tổn tại hoặc những lỗi khác xảy ra thì sẽ phản hổi một thông báo lỗi thích hợp. Trình duyệt web của bạn nhận lại được các tập tin html css (nhiều file khác)… từ server trả về và render hiển thị trang theo yêu cầu.

Một client hay trình duyệt muốn giao tiếp với web server thì phải thông qua giao thức HTTP hoặc HTTPS. Vậy giao thức HTTP là cái gì mà phải cần qua chúng để giao tiếp.

Để hiểu cấu trúc và hoạt động của Web Server thì việc nắm các kiến thức nền tảng liên quan như **giao thức liên mạng TCP/IP, TCP, UDP, Giao thức HTTP/HTTPS , Web Socket…**

Bộ giao thức TCP/IP (Internet Protocol Suite) là bộ giao thức mà Internet và các mạng máy tính đang sử dụng và chạy trên đó. Nó gồm 2 giao thức chính là TCP (Transmission Control Protocol - Giao thức điều khiển giao vận) và IP (Internet Protocol - Giao thức liên mạng). Bộ giao thức TCP/IP được coi là một tập hợp các tầng. Mỗi tầng giải quyết một tập các vấn đề liên quan đến việc truyền tải dữ liệu, và cung cấp cho các giao thức tầng cấp trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng các dịch vụ của các tầng thấp hơn.

Hay nói cách khác, các tầng trên gần với người dùng hơn và làm việc với dữ liệu trừu tượng hơn, chúng dựa vào các tầng cấp dưới để biến đổi dữ liệu thành các dạng mà cuối cùng có thể truyền đi một cách vật lý. Có 4 tầng của TCP/IP

* Tầng ứng dụng - Application Layer

Tầng cao nhất trong cấu trúc phân lớp của TCP/IP. Tầng này bao gồm tất cả các chương trình ứng dụng sử dụng các dịch vụ sẵn có thông qua một chồng giao thức TCP/IP như DHCP, DNS, HTTP, FTP, Telnet, SMTP, … Chúng sẽ tương tác với tầng vận chuyển để truyền hoặc nhận dữ liệu.

* Tầng giao vận - Transport Layer

Gồm 2 đại biểu chính là giao thức TCP và giao thức UDP (User Datagram Protocol) mà mình giới thiệu tiếp đây.

* Tầng mạng - Internet Layer

IP (IPv4, IPv6)

* Tầng liên kết - Link Layer

Ethernet, Wifi, Token Ring, PPP, Frame Relay…

HTTP - HyperText Transfer Protocol (Giao thức truyền siêu văn bản). Giao thức này nằm trong tầng Application Layer, được sử dụng để truyền nội dung trang web từ web server đến trình duyệt web ở client. Là giao thức Client/Server dùng cho Internet - World Wide Web, HTTP thuộc tầng ứng dụng của bộ giao thức TCP/IP (Các giao thức nền tảng cho Internet). Cơ chế hoạt động chính của HTTP là Request-Response: Web Client sẽ gửi Request đến Web Server, Web Server xử lý và trả về Response cho Web Client.

Phiên bản hoàn chỉnh đầu tiên của HTTP là **HTTP 0.9** (Ra đời năm 1991), Tiếp theo là **HTTP 1.0** (Giới thiệu chính thức năm 1996), **HTTP 1.1** (1997) và mới đây nhất là **HTTP 2.0**. Các phiên bản sau ra đời nhằm thay thế phiên bản trước, kế thừa những chức năng cốt lõi của phiên bản trước nhưng có nhiều cải tiến và bổ sung. Hiện nay thì HTTP 2.0 chưa được dùng phổ biến do còn khá mới và do các doanh nghiệp cũng phần nào ngại chuyển đổi. Do vậy, HTTP 1.1 vẫn là giao thức HTTP phổ biến nhất. HTTP 1.0 vẫn còn được sử dụng nhiều trong hệ thống Proxy và một số ứng dụng cũ.

Ngày nay, một số trình duyệt web yêu cầu sử dụng giao thức HTTPS cho tất cả trang web nhằm giúp bảo mật thông tin hơn.

**HTTPS là phiên bản an toàn của HTTP**(viết tắt của Hyper Text Transfer Protocol Secure - giao thức truyền tải siêu văn bản bảo mật), giao thức mà qua đó dữ liệu được gửi giữa trình duyệt và trang web bạn đang kết nối. Để chuyển từ giao thức HTTP sang HTTPS thì sẽ cần phải cài thêm hoặc mua thêm SSL để có thể sử dụng được HTTPS.

Các phương thức yêu cầu phổ biến nhất của HTTP:

* GET
* POST
* PUT
* DELETE
* TRACE
* OPTIONS
* CONNECT
* PATCH

Đặc điểm của HTTP là Stateless Protocol - Phi trạng thái là một giao thức giao tiếp mà mỗi yêu cầu được xem là một phiên giao dịch độc lập (nghĩa là nó không lưu giữ bất kỳ thông tin nào liên quan đến các yêu cầu trước đó cũng như các phiên làm việc trước)

Đối với kỹ thuật Backend này, em sử dụng NodeJs và framework ExpressJs để xử lý logic giao tiếp với cơ sở dữ liệu và trả dữ liệu về cho Frontend để hiển thị ra cho người dùng thông qua API.

**NodeJs** là một nền tảng chạy trên môi trường V8 JavaScript runtime - một trình thông dịch JavaScript cực nhanh chạy trên trình duyệt Chrome. Node JS nổi trội về tốc độ thực thi và khả năng mở rộng. Đó là một yêu cầu khá quan trọng khi bạn tạo ra một sản phẩm mới và muốn đảm bảo có thể mở rộng nhanh, đáp ứng được một lượng lớn người dùng truy cập website.

Node JS có thể xử lý hàng ngàn kết nối, hỗ trợ hàng chục nghìn người dùng truy cập cùng lúc mà không bao giờ bị khóa - điều khó thực hiện với PHP. Đi kèm với Nodejs là ExpressJs - là một Framework nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile.

Nodejs đóng vai trò như một web server. Khi tạo một dự án liên quan đến server làm việc với CSDL hoặc khi chạy một dự án bằng những framework Frontend thì Nodejs là một web server ở phía client với tên gọi là localhost. Như đã nói ở trên thì Nodejs là một trình biên dịch javascript nên những file javscript sẽ được thực thi và chạy lệnh ngay trên chính nodejs bằng cách mở Terminal và gõ lệnh node + [tên file js] mà không cần đến trình duyệt.

Tiếp theo là framework **ExpressJs** là một framework nhỏ, được xây dựng dựa trên nền tảng Nodejs và nó cung cấp nhiều tính năng mạnh mẽ trong việc phát triển các ứng dụng web hay ứng dụng di động. Expressjs còn hỗ trợ nhiều phương thức http và middleware tạo ra một API linh hoạt, mạnh mẽ và dễ sử dụng. Ngoài ra còn có một vài chức năng chính như:

* Thiết lập các middleware để xử lý các HTTP request và trả về.
* Định nghĩa các route để cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
* Trả về các trang html dựa vào các tham số.

Để giao tiếp giữa backend và frontend và bên Frontend có thể nhận được dữ liệu từ phía backend để hiển thị giao diện thì phải thông qua API. Sẽ tìm hiểu một chút về API là gì?

API là cơ chế cho phép 2 thành phần phần mềm giao tiếp với nhau bằng một tập hợp các định nghĩa và giao thức. API là cụm viết tắt của Application Program Interface (Giao diện chương trình ứng dụng). Trong ngữ cảnh API, từ “Ứng dụng” đề cập đến mọi phần mềm có chức năng riêng biệt. Giao diện có thể được xem là một hợp đồng dịch vụ giữa 2 ứng dụng. Hợp đồng này xác định cách thức hai ứng dụng giao tiếp với nhau thông qua các yêu cầu và phản hồi. Tài liệu API của các ứng dụng này chứa thông tin về cách nhà phát triển xây dựng cấu trúc cho những yêu cầu và phản hồi đó.

Ví dụ: hệ thống phần mềm của cơ quan thời tiết chứa dữ liệu về thời tiết hàng ngày. Ứng dụng thời tiết trên điện thoại của bạn sẽ “trò chuyện” với hệ thống này qua API và hiển thị thông tin cập nhật về thời tiết hàng ngày trên điện

Kiến trúc API thường được giải thích dưới dạng máy chủ và máy khách. Ứng dụng gửi yêu cầu được gọi là máy khách, còn ứng dụng gửi phản hồi được gọi là máy chủ. Như vậy, trong ví dụ về thời tiết, cơ sở dữ liệu của cơ quan thời tiết là máy chủ còn ứng dụng di động là máy khách.

API hoạt động theo 4 cách khác nhau, tùy vào thời điểm và lý do chúng được tạo ra.

Trong đó API được sử dụng phổ biến trong các ứng dụng, trang web và em cũng sử dụng loại API này trong đề tài đó là RESTful API.

REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE, vv đến một URL để xử lý dữ liệu.

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế các API cho các ứng dụng web để quản lý các resource. RESTful là một trong những kiểu thiết kế API được sử dụng phổ biến ngày nay để cho các ứng dụng (web, mobile…) khác nhau giao tiếp với nhau.

Chức năng quan trọng nhất của REST là quy định cách sử dụng các HTTP method (như GET, POST, PUT, DELETE…) và cách định dạng các URL cho ứng dụng web để quản các resource. RESTful không quy định logic code ứng dụng và không giới hạn bởi ngôn ngữ lập trình ứng dụng, bất kỳ ngôn ngữ hoặc framework nào cũng có thể sử dụng để thiết kế một RESTful API.

RESTful API hoạt động như thế nào ?



Hình 3: Cấu trúc hoạt động của RESTful API

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng.

GET (SELECT): Trả về một Resource hoặc một danh sách Resource.

POST (CREATE): Tạo mới một Resource.

PUT (UPDATE): Cập nhật thông tin cho Resource.

DELETE (DELETE): Xoá một Resource.

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

* 1. Cơ sở dữ liệu (Database)

Cơ sở dữ liệu là một bộ sưu tập dữ liệu được tổ chức bày bản và thường được truy cập từ hệ thống máy tính hoặc tồn tại dưới dạng tập tin trong hệ quản trị cơ sở dữ liệu.

Database có vai trò vô cùng quan trọng khi làm việc với hệ thống dữ liệu. Chúng giúp người dùng thành công trong việc kết nối các dữ liệu. Người dùng có thể truy cập hệ cơ sở dữ liệu nhanh chóng và dễ dàng hơn. Database chính là cơ sở nguồn để người dùng có thể truy xuất ra những thông tin cần thiết.

Đặc điểm chính của Database chính là truy xuất ra những thông tin, dữ liệu bằng nhiều phương thức khác nhau. Các nội dung truy xuất được đảm bảo toàn vẹn dữ liệu ở mức độ cao. Đồng thời, nguồn thông tin khi xuất ra hoàn toàn không bị trùng lặp, nếu có thì xác suất cũng rất thấp. Một cơ sở dữ liệu Database cho phép nhiều người dùng đồng thời truy cập trong cùng một thời gian. Có 2 loại hệ quản trị CSDL đó là SQL và NoSQL.

**SQL** chính là viết tắt của **S**tructured **Q**uery **L**anguage là ngôn ngữ chuẩn để xử lý. Cơ sở dữ liệu quan hệ. Một cơ sở dữ liệu quan hệ xác định các mối quan hệ dưới dạng các bảng.

Các cơ sở dữ liệu quan hệ như Cơ sở dữ liệu MySQL, Oracle, SQL Server, Sybase*,* v.v. sử dụng **SQL**.

**NoSQL** là một thuật ngữ chung cho các hệ CSDL không sử dụng mô hình dữ liệu quan hệ. **NoSQL** đặc biệt nhấn mạnh đến mô hình lưu trữ cặp giá trị - khóa và hệ thống lưu trữ phân tán.

**NoSQL** ban đầu là một cơ sở dữ liệu cung cấp cơ chế lưu trữ và truy xuất dữ liệu. Dữ liệu này được mô hình hóa bằng các phương tiện khác với các quan hệ dạng bảng được sử dụng trong cơ sở dữ liệu quan hệ. Những cơ sở dữ liệu như vậy đã ra đời vào cuối những năm 1960, nhưng không có được biệt danh **NoSQL** cho đến khi sự gia tăng phổ biến vào đầu thế kỷ XXI.

Thuật ngữ **NoSQL** được giới thiệu lần đầu vào năm 1998 sử dụng làm tên gọi chung cho các hệ CSDL quan hệ nguồn mở nhỏ không sử dụng **SQL** để truy vấn. Thuật ngữ **NoSQL** đánh dấu bước phát triển của thế hệ CSDL mới: **phân tán** + **không ràng buộc**. Hệ thống **NoSQL** đôi khi cũng được gọi là **Not Only SQL** để nhấn mạnh thực tế là chúng có thể hỗ trợ các ngôn ngữ truy vấn giống **SQL**.

Đặc điểm của NoSQL:

* **Phi quan hệ:** không có ràng buộc nào cho việc nhất quán dữ liệu.
* **Mô hình lưu trữ phân tán** các tập tin hoặc dữ liệu ra nhiều máy khác nhau trong mạng LAN hoặc Internet dưới sự kiểm soát của phần mềm.
* **NoSQL** lưu trữ dữ liệu của mình theo dạng cặp giá trị “key – value”. Sử dụng số lượng lớn các node để lưu trữ thông tin.
* **Tính nhất quán không theo thời gian thực:** Sau mỗi thay đổi CSDL, không cần tác động ngay đến tất cả các CSDL liên quan mà được lan truyền theo thời gian.
* Mô hình dữ liệu và truy vấn linh hoạt.
* Triển khai đơn giản, dễ nâng cấp và mở rộng.

Để lưu trữ dữ liệu ứng dụng thì em sử dụng cơ sở dữ liệu là MongoDB – Mongodb là hệ quản trị cơ sở dữ liệu mã nguồn mở, và thuộc dạng NoSQL. Là một cơ sở dữ liệu hướng tài liệu, các dữ liệu được lưu trữ trong tài liệu kiểu JSON thay vì dạng bảng như cơ sở dữ liệu quan hệ nên truy vấn sẽ rất nhanh.

**MongoDB** là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng ngôn ngữ lập trình C++.

Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.

Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

MongoDB cũng có những ưu điểm và nhược điểm khi dùng để phát triển ứng dụng.

Ưu điểm:

* Schema linh hoạt: Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ các các kích cỡ và các document khác nhau.
* Cấu trúc đối tượng rõ ràng: Tuy rằng cấu trúc của dữ liệu là linh hoạt nhưng đối tượng của nó được xác định rất rõ ràng.
* Sử dụng bộ nhớ nội tại, nên truy vấn sẽ rất nhanh.
* MongoDB rất dễ mở rộng.
* Không có các join: Điều này cũng góp phần tạo nên tốc độ truy vấn cực nhanh trên mongoDB.
* MongoDB phù hợp cho các ứng dụng realtime.

Nhược điểm:

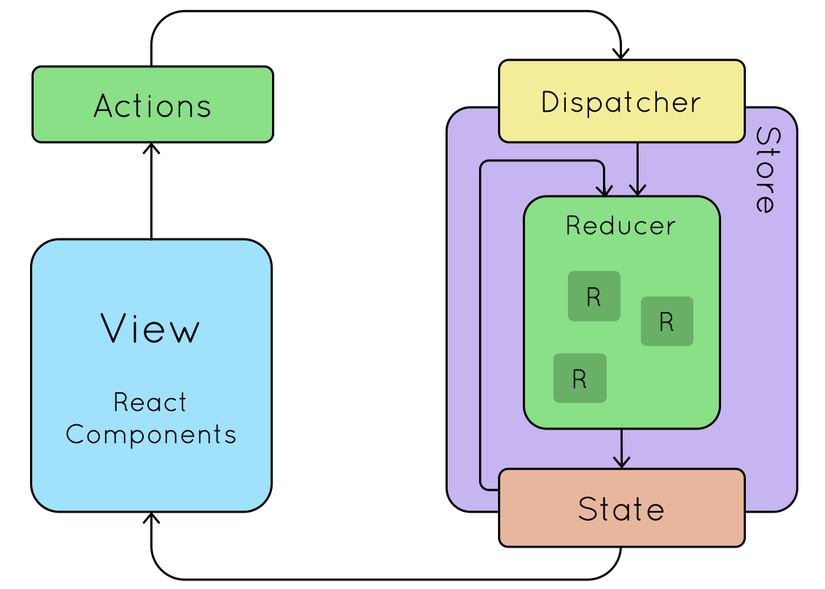
* + Điều đầu tiên phải kể đến ở đây là MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận.
  + Một nhược điểm của NoSQL là hầu hết các giải pháp đều không tuân thủ ACID mạnh mẽ (Atomic, Consistency, Isolation, Durability) như các hệ thống RDBMS được thiết lập tốt hơn.
  + Giao dịch phức tạp
  + Không có chức năng hoặc thủ tục lưu trữ tồn tại nơi bạn có thể liên kết logic
  1. Công nghệ và ngôn ngữ đã sử dụng

Để tiếp thu và vận dụng những kiến thức đã học trên trường, trên lớp và tự học thì em quyết định chọn đề tài xây dựng một ứng dụng web tìm kiếm nhà trọ hỗ trợ cho sinh viên. Trong đề tài đồ án này, công nghệ và ngôn ngữ mà em sử dụng để phát triển ứng dụng bao gồm:

* + Về Frontend sử dụng các ngôn ngữ và thư viện như ReactJs, Redux, CSS module, SASS/SCSS.
  + Về Backend sử dụng ngôn ngữ NodeJs, framework ExpressJs, Firebase và cơ sở dữ liệu là MongoDB.

**Redux** là 1 thư viện Javascript giải quyết tạo ra một kho lưu trữ trạng thái dữ liệu của ứng dụng. Kho chung lưu trữ trạng thái của ứng dụng chỉ được phép đọc nếu muốn thay đổi trạng thái của ứng dụng thì cần phải dispatch một action (là một object mô tả hành động muốn thay đổi trạng thái) và để thay đổi các trạng thái đó thì cần một Reducer (là nơi xử lý, cập nhật trạng thái mới theo action và trả về cho kho chung trạng thái mới). Kho chung nhận được trạng thái mới được cập nhật từ phía reducer và cập nhật cho view để hiển thị cho người dùng.

Cấu trúc của Redux:



Hình 4 Cấu trúc của Redux

**CSS Module** là một file css strong đó tất cả các tên class hay tên các animation được xác định trong phạm vi cục bộ theo mặc định. Bản chất một dự án React sẽ cần phải chia ra nhiều component để dễ quản lý và bảo trì nên việc viết style css cho tất cả các component vào 1 file thì sẽ gây ra vấn đề trùng lặp tên class và ảnh hưởng đến các component khác nên CSS Module sẽ giải quyết việc đó, nó giúp style css từng component một cách độc lập và riêng lẻ các file và không xảy ra vấn đề trùng lặp class.

Và CSS cũng giống như các ngôn ngữ css mở rộng như sass hay scss thì không thể thực thi trực tiếp trên trình duyệt được mà phải thông qua trình biên dịch (webpack,….).

CSS Module đảm bảo tất cả các style chỉ cho một component và được tồn tại ở một nơi và chỉ được sử dụng cho riêng component đó mà không được sử dụng ở bất kỳ chỗ khác, điều này khắc phục được vấn đề phạm vi toàn cầu (global) trong ứng dụng.

**SASS/SCSS** là ngôn ngữ mở rộng của css. Nó giúp có thể viết css theo phong cách ngôn ngữ lập trình, có cấu trúc rõ ràng, lành mạnh và bảo trì code hơn.

SASS (Syntactically Awesome StyleSheets) là một tiền xử lí CSS giúp viết CSS nhanh hơn và có cấu trúc rõ ràng hơn. Với SASS, có thể viết CSS theo thứ tự rõ ràng, quản lý các biến đã được định nghĩa sẵn, các class dùng chung hay có thể tự động nén tập tin CSS lại để bạn tiết kiệm dung lượng. SASS được thiết kế và viết bởi các lập trình viên Ruby. Bởi vậy, Sass sử dụng cú pháp giống như Ruby với việc không có các dấu ngoặc nhọn {}, dấu chấm phẩy, việc viết CSS như vậy không giống CSS thuần nên việc hiểu và đọc code SASS rất khó hiểu.

Cho đến khi phiên bản 3.0 được phát hành vào tháng 5/2010, nó giới thiệu một cú pháp mới được gọi là SCSS (Sassy CSS). Cú pháp này nhằm thu hẹp khoảng cách giữa Sass và CSS bằng cách một cú pháp thân thiện với CSS.

Những tính năng mới và hay ho mà SASS đem lại để giúp CSS được hiệu quả hơn.

* Sử dụng biến.

Biến là một kí hiệu để lưu giá trị. Và có **tác dụng** để thay thế hàng loạt các thuộc tính của website.

Ví dụ như:

$mx: #c69;

$bg\_color: #c3c3c3;

body {

background-color: $bg\_color;

}

h1.display-3 {

color: $mx;

}

p {

color: $mx;

font-size: 20px

}

* Tính năng xếp chồng trong SASS.

$mx: #FFCC00;

$mx2: #fff;

$fs1: 13px;

$fs2: 15px;

$fs3: 25px;

$fs4: 48px;

$bg\_color: #699;

$font\_family: seoge;

$t-center: center;

.cart {

width: 900px;

margin: auto;

background: $bcolor;

h1 {

text-align: $t\_center;

color: $mx2;

}

p {

text-align: $t\_center;

color: $mx;

font : {

family: $font\_family;

size: $fs2;

weight: bold;

style: italic;

}

}

}

* Tính năng sử dụng lại code trong SASS.

Trong SASS có khái niệm là @mixin, [@mixin](https://viblo.asia/u/mixin) có thể hiểu nó tương tự như 1 hàm trong lập trình vậy.

Đơn giản chỉ cần đặt tên cho mixin và ở dưới image em gán @include tên\_mixin là được, như ví dụ phía dưới.

@mixin left\_image() {

float: left;

width: 100px;

margin-right: 20px;

}

.cart {

width: 900px;

margin: auto;

background: $bg\_color;

img {

@include left-image;

}

}

.shop {

width: 600px;

margin: auto;

img {

@include left\_image;

}

}

* Tính năng viết pseudo class nhanh trong SASS.

h1 {

text-align: $t\_center;

color: $mx2;

&:hover {

color: #FF00F0;

}

}

& ở đây có thể hiểu là nó lấy địa chỉ của phần tử chứa nó, ở đây là h1.

* Tính năng tính toán trong SASS.

Tóm lại, SCSS như là một bản nâng cấp của SASS, giúp chúng ta viết CSS dễ dàng hơn, dễ hiểu hơn.

**Firebase** là một nền tảng đa năng, nó cung cấp rất nhiều dịch vụ khác nhau cho người dùng.

Năm 2011, tiền thân của Firebase chính là Envolve. Envolve được biết đến dưới mô hình khởi nghiệp, startup, Envolve cung cấp cho người dùng các API để dễ dàng tích hợp tính năng trò chuyện vào trong trang web. Tuy nhiên, người dùng lại sử dụng chúng để truyền dữ liệu ứng dụng chứ không đơn thuần là nhắn tin trò chuyện. Chính điều này đã khiến các nhà phát triển Envolve quyết định tách riêng hệ thống nhắn tin và kiến trúc thời gian thực.

Đến năm 2014, Google mua lại thì Firebase nhanh chóng phát triển thành nền tảng phát triển ứng dụng di động và trang web. Nền tảng này là sự kết hợp giữa hệ thống đám mây với hệ thống máy chủ của Google để tập trung chính cho 2 đối tượng là:

* Develop & test your app: phát triển và thử nghiệm các ứng dụng được thiết kế.
* Grow & engage your audience: phân tích dữ liệu và tối ưu hóa trải nghiệm với người dùng.

Firebase là một nền tảng đa năng, nó cung cấp rất nhiều dịch vụ khác nhau cho người dùng. Trong đề tài này, em sử dụng dịch vụ authentication của firebase để xác thực người dùng cho việc đăng nhập, đăng ký tài khoản trên ứng dụng.

**KẾT LUẬN CHƯƠNG 2**

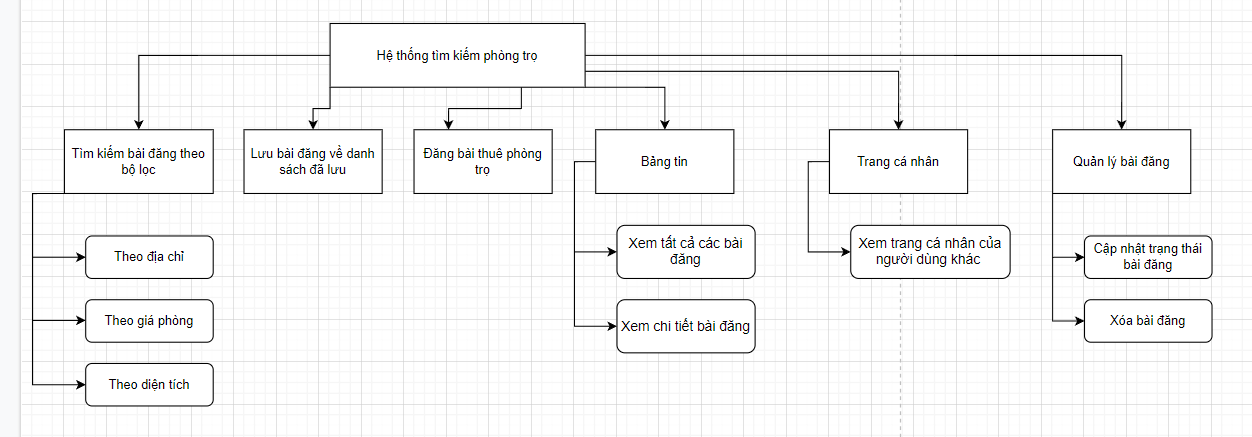
Qua chương này em đã hiểu được rõ khái niệm và những kỹ thuật trong lập trình web, cách áp dụng những công nghệ mới, ngôn ngữ lập trình mới giúp em có thể áp dụng những kỹ thuật nói trên để phát triển một ứng dụng phần mềm theo một quy trình rõ ràng và hoàn thiện.

Ở chương sau sẽ tập trung chủ yếu nói sâu hơn về hệ thống ứng dụng web mà em đã áp dụng những kiến thức đã được học để phát triển và cũng như phân tích nghiệp vụ, lấy yêu cầu của người dùng, khách hàng, thiết kế cơ sở dữ liệu cho ứng dụng web tìm nhà trọ hỗ trợ sinh viên.

**CHƯƠNG 3: PHÂN TÍCH, THIẾT KẾ HỆ THỐNG ỨNG DỤNG HỖ TRỢ SINH VIÊN TÌM KIẾM NHÀ TRỌ**

1. Phân tích các yêu cầu của ứng dụng
   1. Yêu cầu chức năng

* Tìm kiếm bài đăng theo bộ lọc
* Lọc theo giá tiền
* Lọc theo địa chỉ
* Lọc theo diện tích
* Đăng bài cho thuê phòng trọ
* Lưu bài đăng của người khác về danh sách đã lưu
* Xem tất cả các bài đăng trên trang chủ
* Xem chi tiết bài đăng thuê phòng trọ
* Xem trang cá nhân của người dùng khác
* Quản lý người dùng trên ứng dụng (Chức năng bên trang quản trị)
  + Khóa tài khoản người dùng nếu vi phạm chính sách nội quy
* Quản lý bài đăng của tất cả người dùng (Chức năng bên trang quản trị)
  + Cập nhật trạng thái bài đăng
  + Xóa bài đăng
  1. Yêu cầu phi chức năng
* Đăng nhập, đăng ký bằng tài khoản facebook và google.
* Đăng xuất tài khoản người dùng.
* Ghi nhớ đăng nhập.
* Giao diện UI/UX đơn giản, dễ nhìn.
* Đăng bài xong sẽ có thông báo cho người dùng biết đã đăng bài thành công hay thất bại.
  1. Yêu cầu chức năng khi đăng bài thuê nhà trọ
* Tiêu đề bài đăng thuê phòng trọ
* Địa chỉ phòng trọ
* Địa chỉ chi tiết
* Loại phòng (phòng trọ, chung cư mini, tìm người ở ghép,…)
* Giá phòng
* Giá điện, nước
* Diện tích
* Số điện thoại chủ phòng
* Mô tả
* Hình ảnh thực tế
* Các tiện ích khác (wifi, máy giặt, nóng lạnh, điều hòa, chỗ để xe, nhà vệ sinh,….)
  1. Biểu đồ phân cấp chức năng

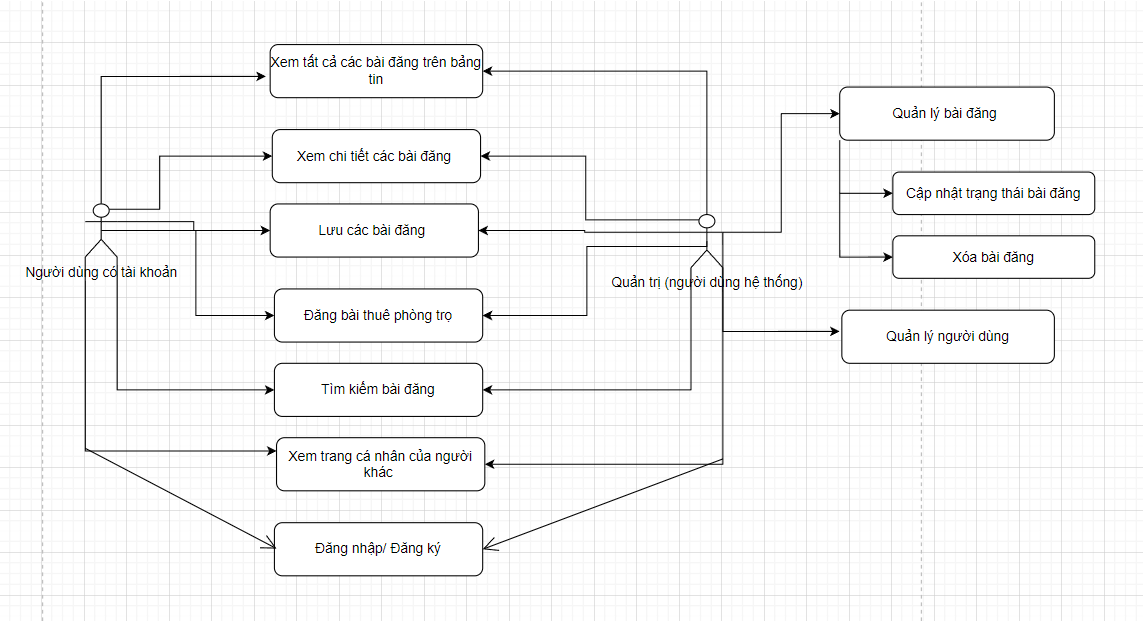


Hình 5: Biểu đồ phân cấp chức năng

1. Thiết kế hệ thống người dùng usecase
   1. Người dùng có tài khoản

* Tìm kiếm bài đăng thuê phòng
* Xem tất cả các bài đăng trên bảng tin
* Xem chi tiết tất cả các bài đăng
* Lưu bài đăng thuê phòng
* Xem trang cá nhân của người dùng khác
* Đăng nhập / Đăng xuất
* Đăng bài thuê phòng trọ
  1. Quản lý (Người dùng hệ thống)
* Đầy đủ tính năng của người dùng có tài khoản
* Quản lý người dùng
  + Khóa tài khoản người dùng vi phạm chính sách, nội quy
* Quản lý các bài đăng của người dùng
  + Cập nhật trạng thái bài đăng
  + Xóa bài đăng không hợp lệ

Sau khi phân tích hệ thống người dùng, em đã thiết kế biểu đồ tổng quát usecase hệ thống ứng dụng như hình bên dưới.

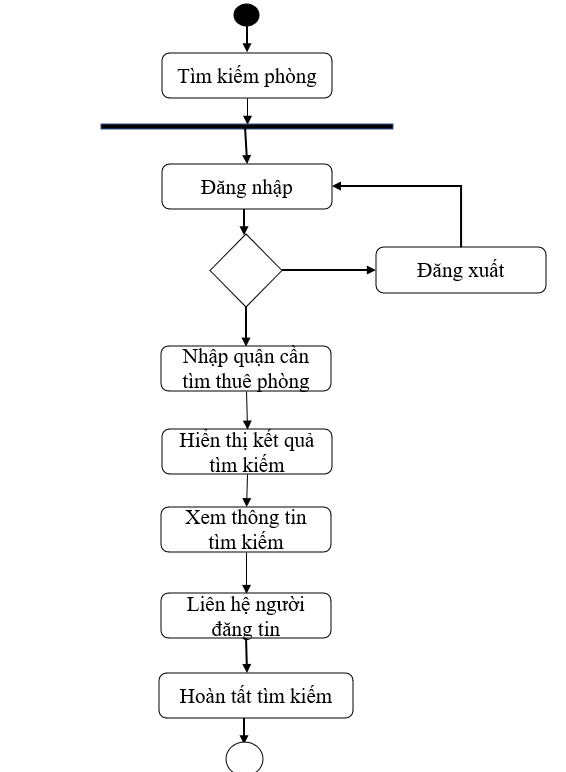


Hình 6: Sơ đồ use case hệ thống tổng quát

Nhìn vào biểu đồ hệ thống người dùng ta có thể thấy được những chức năng cơ bản cần có cũng như các đối tượng tham gia vào hệ thống bao gồm: Người dùng có tài khoản và Quản lý.

1. Thiết kế hệ thống
   1. Biểu đồ hoạt động của hệ thống

3.1.1 Biểu đồ use case tìm kiếm phòng

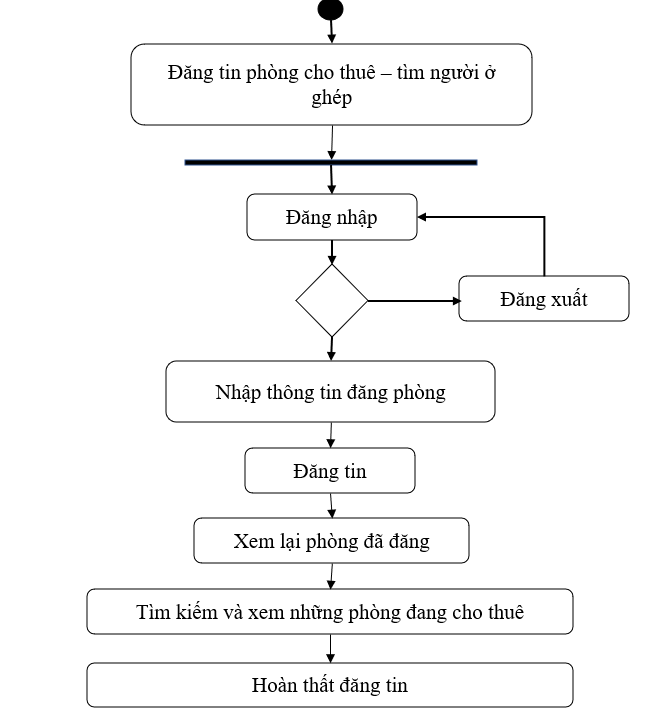


Hình 7: Biểu đồ hoạt động use caser tìm kiếm phòng cùa hệ thống

**Mô tả hoạt động:**

* Người tìm phòng đăng nhập tài khoản vào ứng dụng
* Giao diện đăng nhập hiển thị cho phép người tìm kiếm đăng nhập tài khoản
* Sau khi đăng nhập thành công, người tìm kiếm có thể xem các phòng đang cho thuê ở giao diện trang chủ.
* Người tìm kiếm có thể xem chi tiết thông tin của phòng muốn thuê
* Người tìm kiếm có thể liên hệ người cho thuê phòng gọi qua số điện thoại hoặc nhắn tin
* Ngoài ra người tìm kiếm còn có thể nhập thông tin địa chỉ thành phố quận tìm kiếm vào ô tìm kiếm để tìm phòng trọ ở quận thành phố cần thuê

3.1.2 Biểu đồ use case người đăng tin cho thuê, tìm người ở ghép

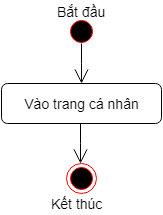


Hình 8: Biểu đồ hoạt động use caser đăng tin cho thuê - tìm người ở ghép cùa hệ thống

**Mô tả hoạt động:**

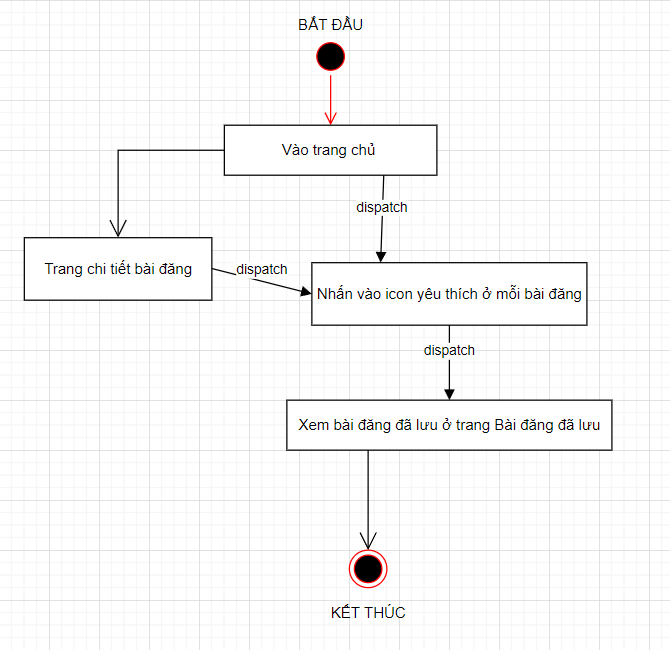
* Người đăng tin đăng nhập tài khoản vào ứng dụng.
* Trong giao diện trang chủ sẽ cho phép người đăng tin thực hiện việc đăng tin phòng trọ cần cho thuê hoặc phòng đang tìm người ở ghép.
* Giao diện đăng tin cho phép người đăng tin nhập đầy đủ thông tin của phòng trọ gồm: thành phố, quận, địa chỉ chi tiết, loại phòng là cho thuê phòng hoặc tìm người ở ghép, giá một tháng, diện tích, các tiện ích khác, mô tả và hình ảnh.
* Sau khi đăng tin thành công thì người đăng tin có thể xem lại những bài đăng mà mình đã đăng tin trước đó trong trang bài đăng của tôi.
* Ngoài ra người đăng tin cũng có thể xem các phòng đang cho thuê hoặc đang tìm người ở ghép trên trang chủ.
* Người đăng tin có thể tìm kiếm những phòng đang cho thuê theo địa chỉ quận thành phố

3.1.3 Biểu đồ usecase xem danh sách bài đăng đã đăng



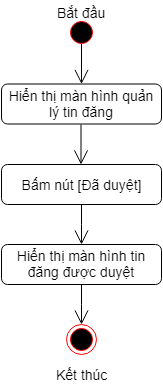
Hình 9: Biểu đồ usecase xem danh sách bài đăng đã đăng

3.1.4 Biểu đồ usecase lưu bài đăng



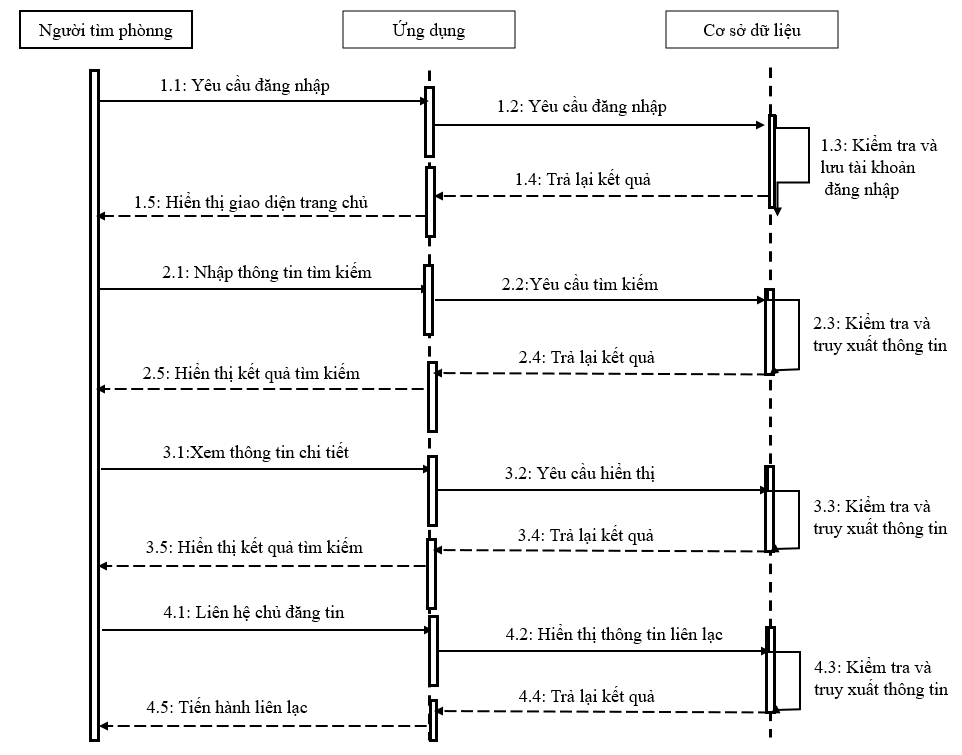
Hình 10: Biểu đồ usecase lưu bài đăng

3.1.5 Biểu đồ usecase quản lý bài đăng (Duyệt bài đăng)



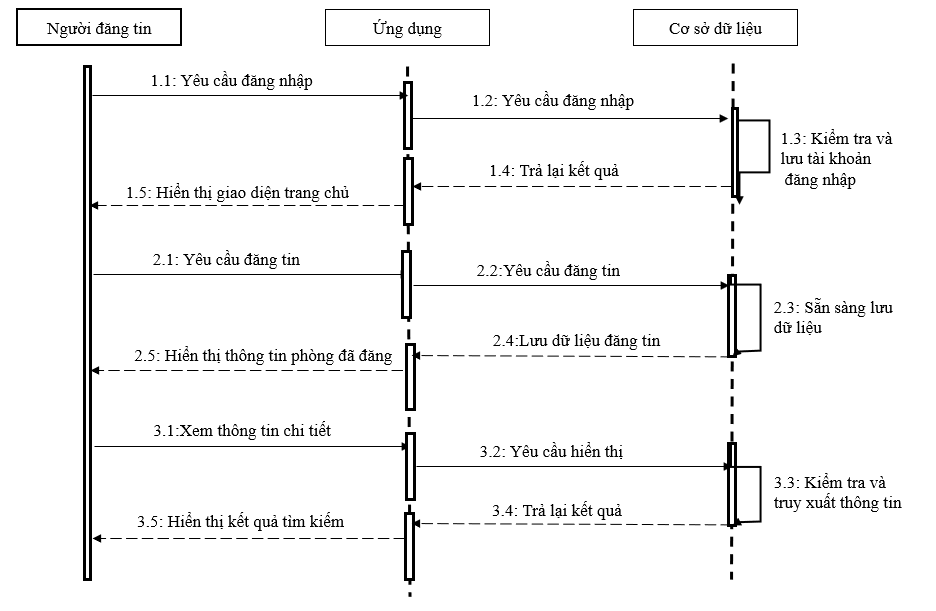
Hình 11: Biểu đồ usecase duyệt bài đăng

* 1. Biểu đồ tuần tự của hệ thống
     1. Biểu đồ tuần tự use case tìm kiếm phòng của hệ thống



Hình 12: Biểu đồ tuần tự use case tìm kiếm phòng của hệ thống

* + 1. Biểu đồ tuần tự use case đăng tin phòng cho thuê, tìm người ở ghép của hệ thống



Hình 13: Biểu đồ tuần tự use case đăng tin phòng cho thuê, tìm người ở ghép của hệ thống

* 1. Phân tích hệ thống chức năng ứng dụng hỗ trợ sinh viên tìm kiếm nhà trọ
     1. Chức năng của ứng dụng

|  |  |
| --- | --- |
| **Chức năng** | **Mô tả** |
| Xem tất cả các bài đăng trên trang chủ | Xem thông tin xem trước bài đăng các phòng đang cho thuê |
| Xem chi tiết các bài đăng thông tin phòng trọ cho thuê | Người dùng có thể xem chi tiết thông tin phòng đang cho thuê với giá cả, diện tích và các tiện ích |
| Tìm kiếm bài đăng | Cho phép người dùng tìm kiếm sau khi nhập thông tin quận thành phố cần tìm dể có thể hiện ra kết quả cần tìm kiếm |
| Đăng bài cho thuê phòng trọ | Người đăng tin phòng cho thuê sẽ dăng đầy đủ thông tin của phòng cho thuê như giá cả, diện tích, địa chỉ để người tìm kiếm dễ nắm bắt thông tin |
| Lưu bài đăng | Cho phép người dùng có thể lưu lại bài đăng thuê phòng trọ về danh sách đã lưu của |
| Xem lại bài đã đăng | Người đăng tin cho thuê – người tìm người ở ghép có thể xem lại bài đăng của mình đã từng đăng |
| Đăng nhập với tài khoản Facebook, Google cá nhân | Ứng dụng cho phép người dùng tài khoản google có sẵn để đăng nhập vào ứng dụng, rất tiện lợi |

Bảng 2: Bảng hệ thống chức năng ứng dụng

* + 1. Phân tích hệ thống về dữ liệu

A, Xác định các thực thể

Dựa vào các chức năng và các biểu đồ phân tích hệ thống, ta có thể xác định được các thực thể và các trường thuộc tính của các thực thể trong ứng dụng hỗ trợ tìm kiếm phòng trọ sinh viên gồm:

* Thực thể
  + Người dùng
  + Bài đăng thuê phòng trọ

B, Xác định các thuộc tính của thực thể

* Thực thể ‘Người dùng’
* ID
* Tên người dùng
* Email
* accessToken
* providerID
* Điện thoại
* Thực thể ‘Bài đăng thuê phòng trọ’
  + ID
  + Tiêu đề bài đăng
  + Loại phòng
  + Địa chỉ
  + Địa chỉ chi tiết
  + Giá phòng
  + Giá điện, nước
  + Diện tích phòng
  + Điện thoại
  + Các tiện ích khác
  + Hình ảnh nổi bật
  + Hình ảnh chi tiết
  + Mô tả
  + Trạng thái bài đăng
  + UserID
  + Trạng thái đã lưu của bài đăng
  1. Thiết kế cơ sở dữ liệu
     1. Bảng CSDL Người dùng

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường | Kiểu Dữ liệu | Mô Tả |
| 1 | ID | ObjectID | ID người dùng |
| 2 | name | String | Họ tên |
| 3 | email | String | email |
| 4 | phone | Number | Số Điện thoại |
| 5 | accessToken | String |  |
| 6 | providerID | String | Tên nhà cung cấp tài khoản |

Bảng 3: Bảng CSDL người dùng

* + 1. Bảng CSDL phòng trọ

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường | Kiểu dữ liệu | Mô tả |
| 1 | ID | ObjectID | ID bài đăng |
| 2 | created\_by | String | Id của người dùng (người đăng) |
| 3 | title | String | Tiêu đề bài đăng |
| 4 | type\_room | String | Loại phòng |
| 5 | price\_room | String | Giá phòng |
| 6 | address | String | Địa chỉ phòng |
| 7 | detail\_address | String | Địa chỉ chi tiết |
| 8 | area\_room | String | Diện tích phòng |
| 9 | Price\_electron | String | Giá điện |
| 10 | Price\_water | String | Giá nước |
| 11 | phone | String | Số điện thoại chủ phòng |
| 12 | utils | Array | Các Tiện ích |
| 13 | description | String | Mô tả |
| 14 | Thumbnail\_img | String | Hình ảnh nổi bật |
| 15 | Detail\_img | Array | Hình ảnh chi tiết |
| 16 | status | Number | Trạng thái bài đăng |
| 17 | isSaved | Boolean | Trạng thái đã lưu của bài đăng |

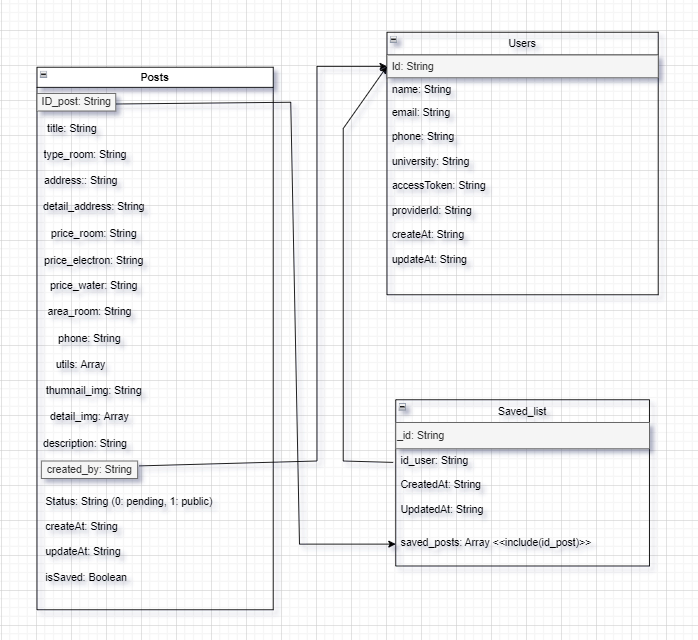
Bảng 4: Bảng CSDL phòng trọ

* + 1. Bảng CSDL danh sách đã lưu

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Trường | Kiểu dữ liệu | Mô tả |
| 1 | ID | ObjectID | ID danh sách đã lưu |
| 2 | id\_user | String | ID chủ sở hữu danh sách |
| 3 | saved\_posts | Array | Chứa các ID bài đăng đã lưu của người dùng |

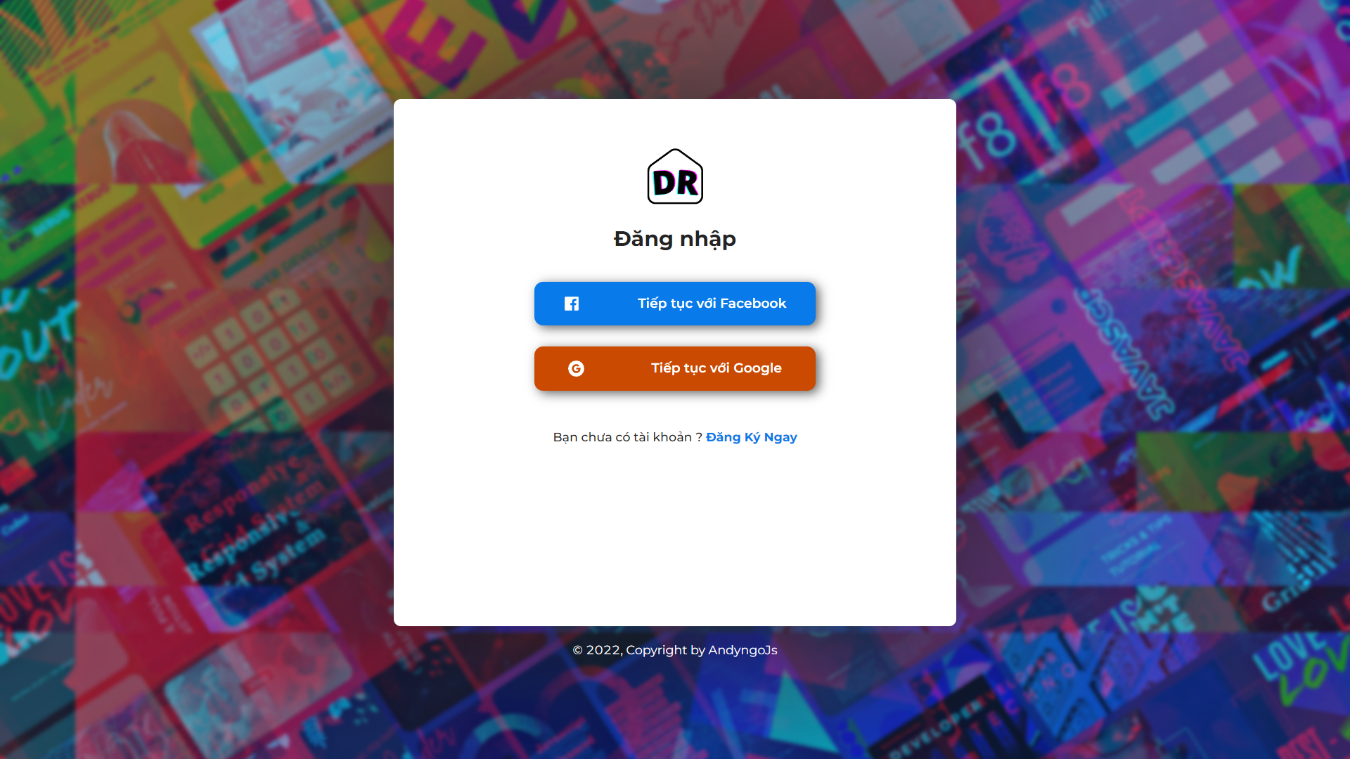
Bảng 5: Bảng CSDL danh sách lưu các bài đăng

* + 1. Sơ đồ cơ sở dữ liệu



Hình 14: Sơ đồ thiết kế cơ sở dữ liệu

1. Thiết kế giao diện người dùng
   1. Giao diện đăng nhập

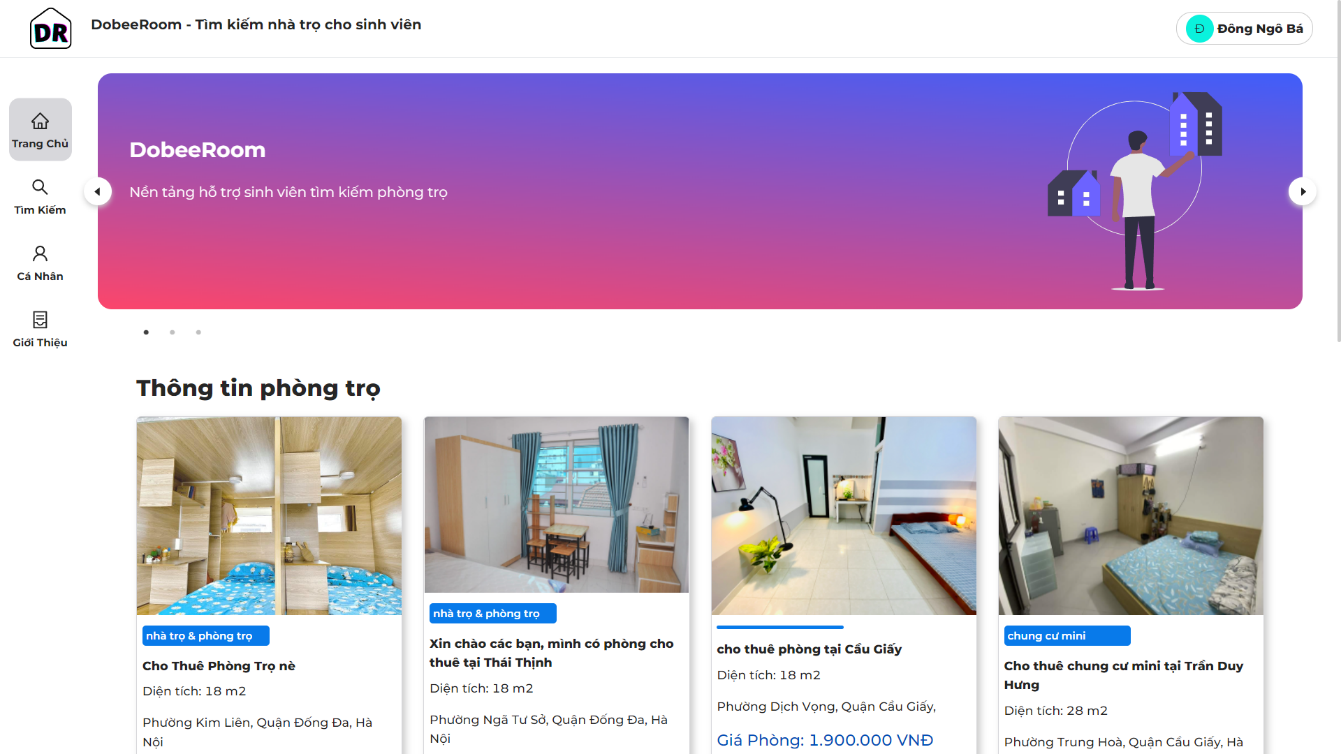


Hình 15: Giao diện đăng nhập người dùng

Giao diện đăng nhập được hiển thị khi người dùng lần đầu truy cập vào ứng dụng web hoặc đăng xuất khỏi ứng dụng.

Hệ thống cho phép người dùng đăng nhập/ đăng ký tài khoản qua các tài khoản Facebook, Google cá nhân. Giúp người dùng tiết kiệm thời gian và đăng nhập nhanh chóng, thuận tiện.

* 1. Giao diện trang chủ

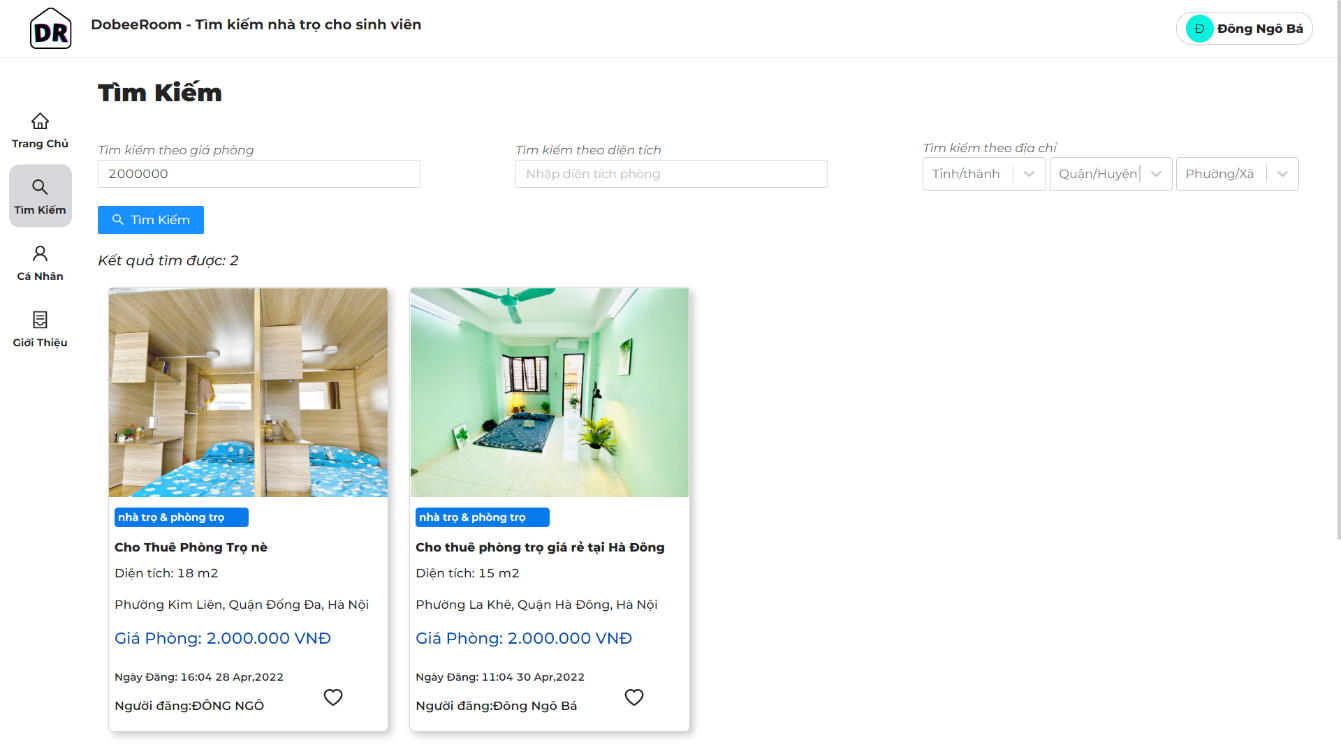


Hình 16: Giao diện trang chủ

Đây là giao diện trang chủ được hiển thị tất cả các bài đăng phòng trọ của người dùng khác. Hiển thị đầy đủ những thông tin chi tiết ở chế độ xem nhanh.

Cập nhật bài đăng nhanh chóng và thuận tiện cho người dùng.

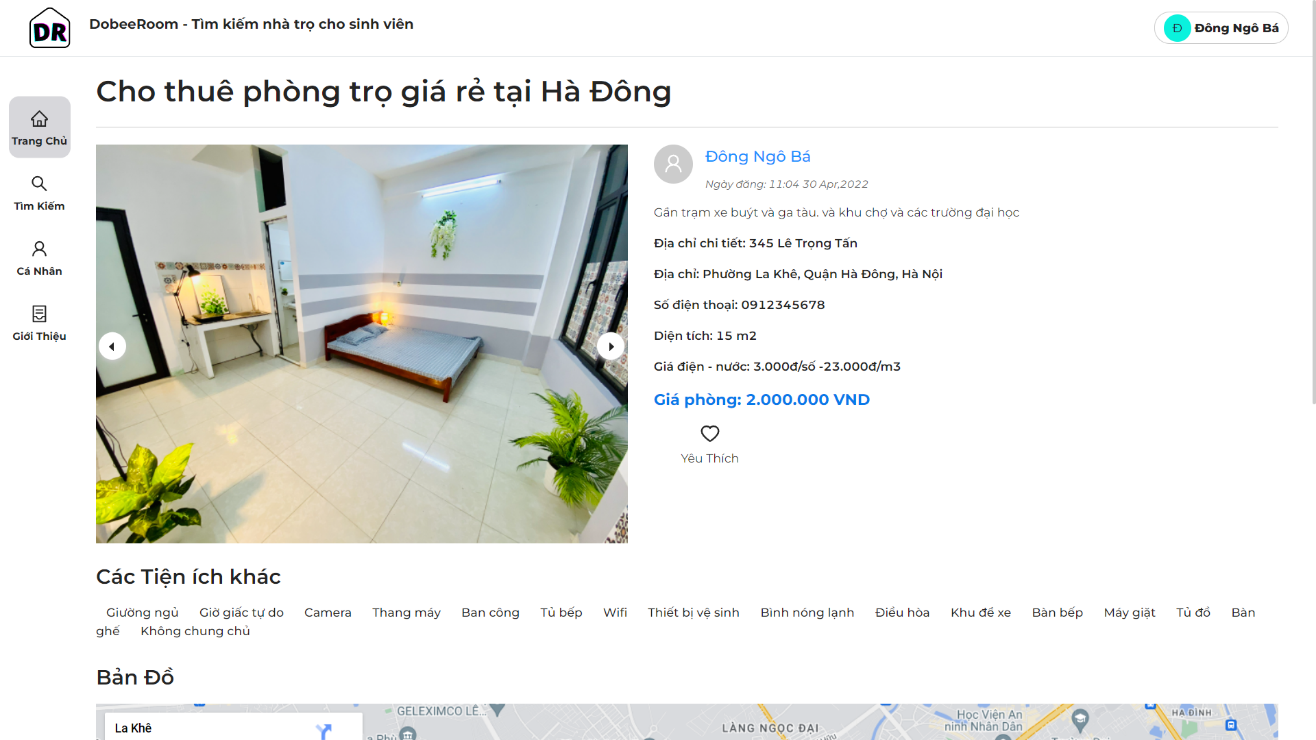
* 1. Giao diện tìm kiếm phòng



Hình 17: Giao diện tìm kiếm bài đăng thuê phòng

Đây là giao diện trang tìm kiếm bài đăng thuê phòng với đầy đủ chức năng lọc tìm kiếm theo yêu cầu của người dùng và khách hàng bao gồm lọc theo giá phòng, lọc theo diện tích, lọc theo địa chỉ và hiển thị các bài đăng sau khi tìm kiếm đơn giản, dễ nhìn để người dùng có thể tìm kiếm được những phòng ưng ý của mình.

* 1. Giao diện chi tiết phòng

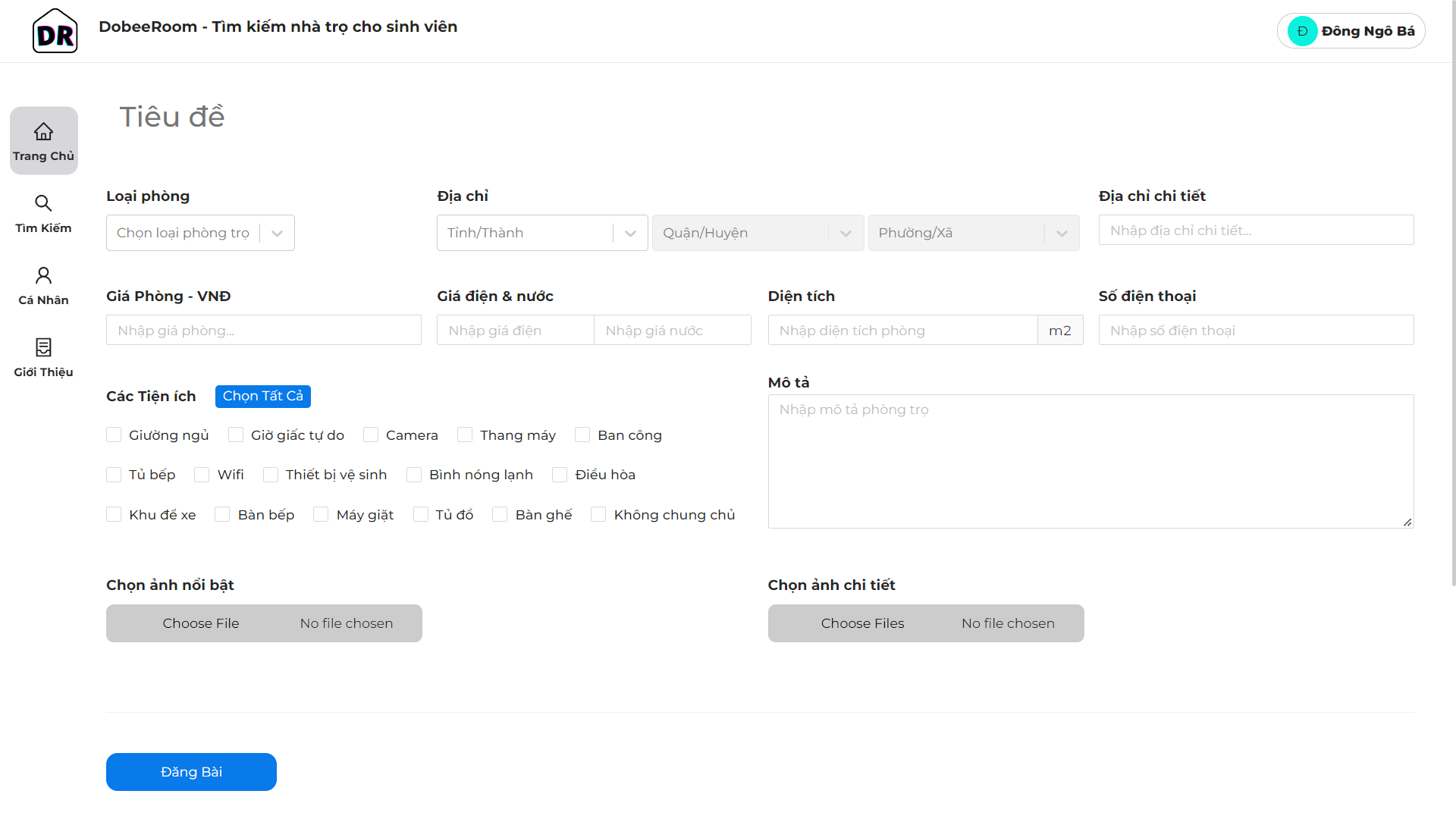


Hình 18: Giao diện chi tiết bài đăng thuê phòng

Đây là giao diện trang chi tiết bài đăng phòng trọ hiển thị đầy đủ thông tin chi tiết bao gồm hình ảnh, địa chỉ, diện tích, giá phòng, mô tả, người đăng, ngày đăng và các tiện ích khác như wifi, điện, nước, ….

Đặc biệt còn tích hợp thêm bản đồ để cho người dùng có thể hình dung nơi phòng trọ cho thuê và đường đi.

* 1. Giao diện khi đăng bài

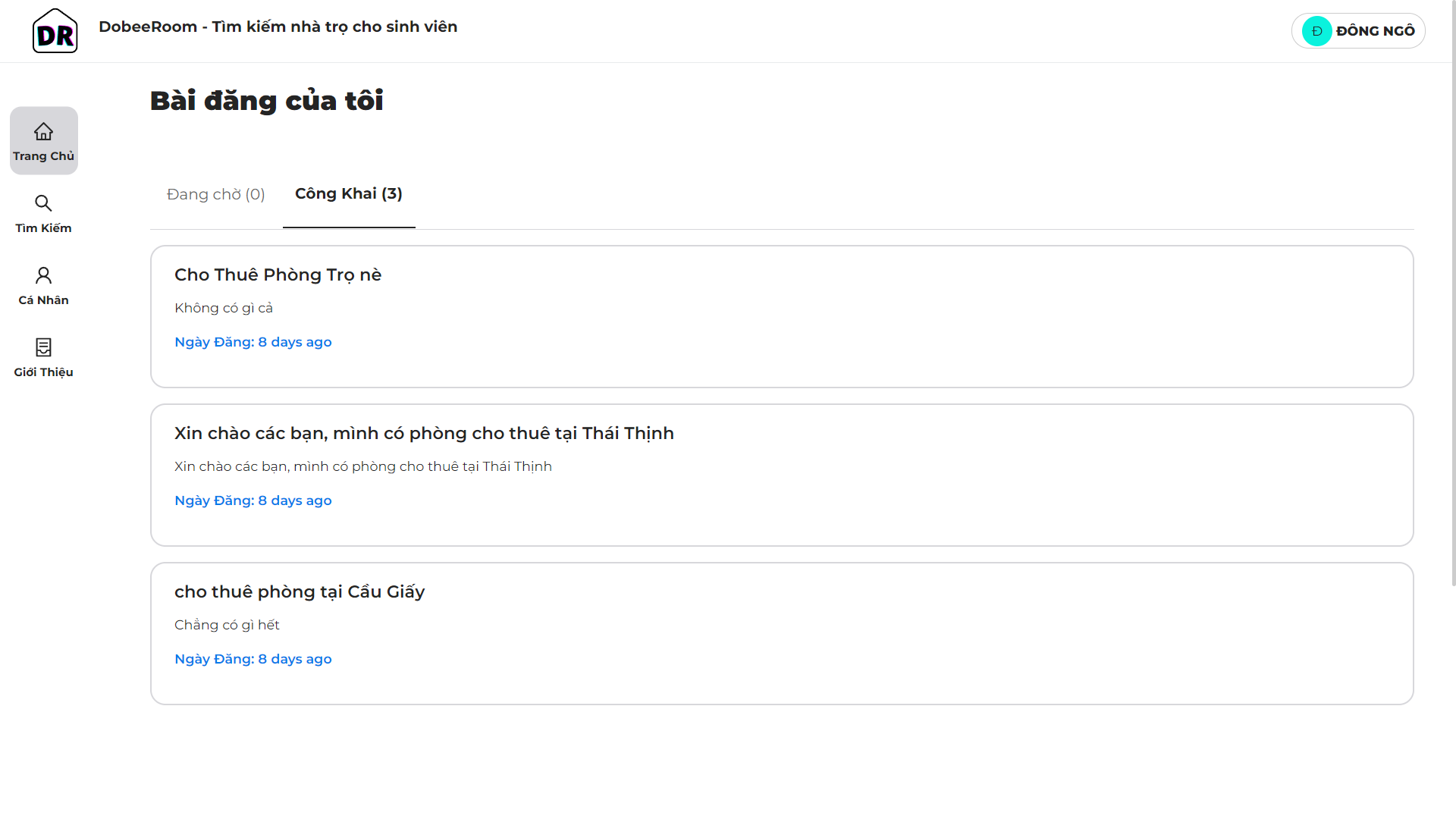


Hình 19: Giao diện đăng bài thuê phòng trọ

Đây là giao diện trang đăng bài thuê phòng trọ và trang này chỉ hiển thị cho người dùng có tài khoản. Và mọi thông tin được ghi trên phải là thông tin chính xác, có tính xác thực cao.

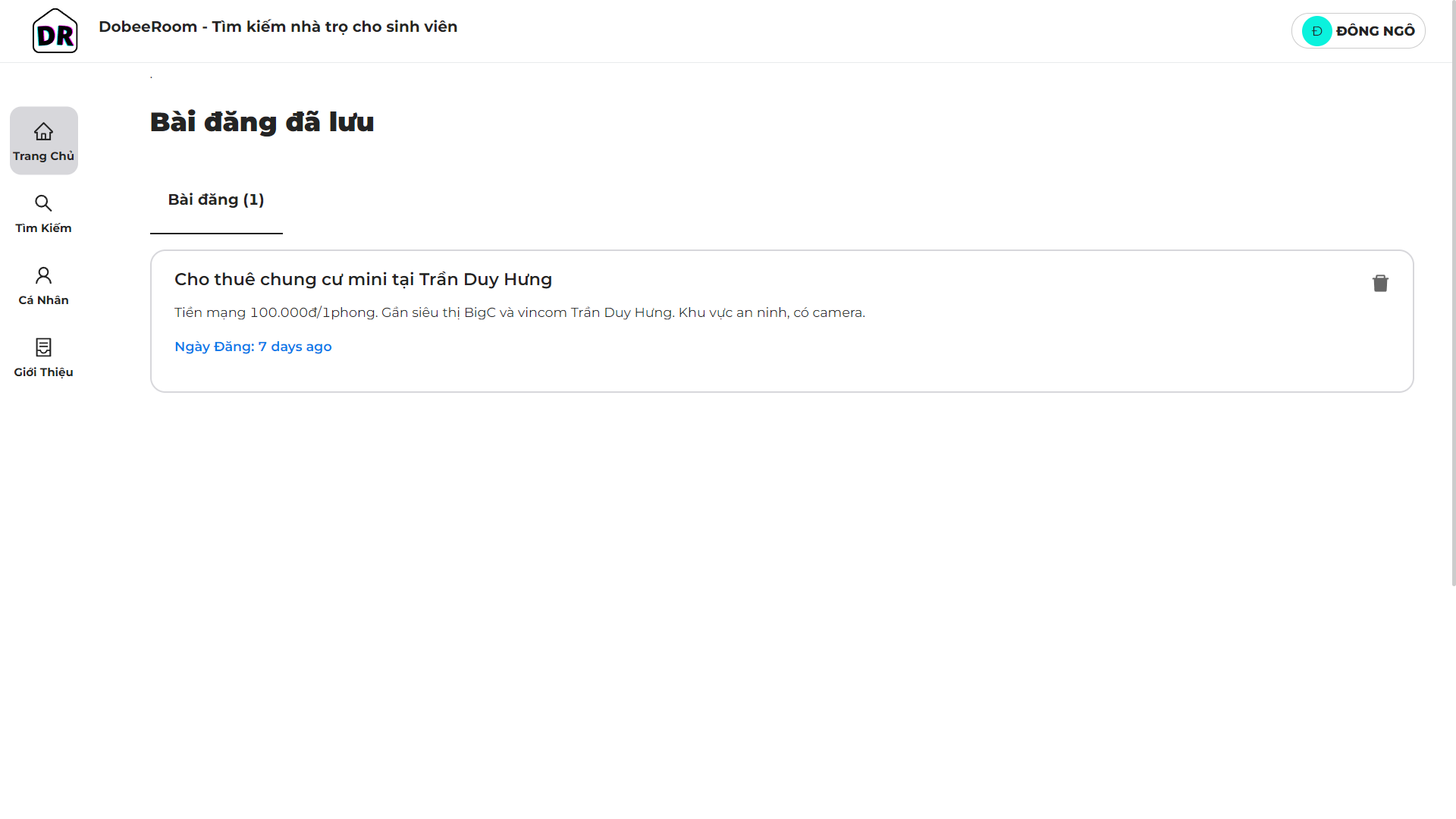
Khi người dùng nhấn nút đăng bài thì sẽ có một thông báo hiện ra là thông báo đã đăng bài thành công hoặc nếu đăng bài thất bại thì sẽ thông báo lỗi đăng bài thất bại.

4.6 Giao diện bài đăng của người dùng



Hình 20: Giao diện bài đăng của tôi

* 1. Giao diện danh sách đã lưu của người dùng



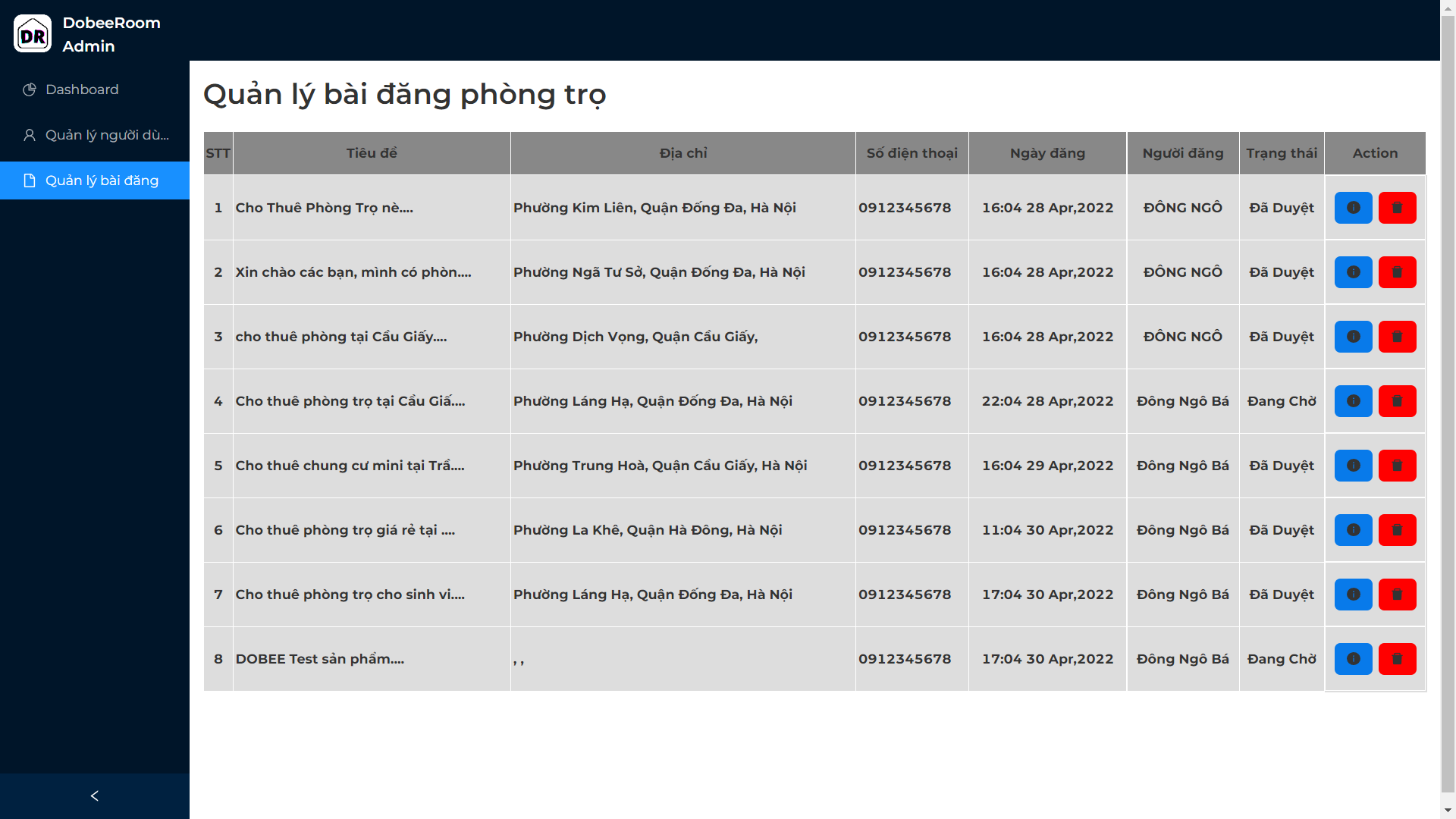
Hình 21: Giao diện Bài đăng đã lưu

* 1. Giao diện trang quản lý người dùng



Hình 22: Giao diện trang quản lý người dùng

* 1. Giao diện trang quản lý bài đăng



Hình 23: Giao diện quản lý bài đăng

**KẾT LUẬN CHƯƠNG 3**

Chương này tập trung chủ yếu vào phân tích nghiệp vụ và đặc tả yêu cầu của người dùng, khách hàng và xác định được các chức năng cần có trong ứng dụng và những chức năng được bổ sung từ người dùng.

Ở chương này giúp em có thêm kiến thức về phân tích nghiệp vụ, phân tích hệ thống và thiết kế cơ sở dữ liệu để cho việc code, lập trình phát triển dễ dàng , đơn giản hơn. Đưa vào một quy trình phát triển phần mềm rõ ràng và theo nguyên tắc nhất định. Nắm được những kiến thức về thiết kế giao diện UI/UX giúp người dùng có được trải nghiệm người dùng tốt nhất, thỏa mản nhất trong ứng dụng web hỗ trợ sinh viên tìm kiếm nhà trọ.

**KẾT LUẬN**

1. Kết quả đạt được

Ứng dụng web sau khi hoàn thành về cơ bản đã đáp ứng đầy đủ nhu cầu tìm kiếm phòng trọ của sinh viên như việc đăng bài thuê phòng và tìm người ở ghép để tiết kiệm chi phí và thời gian.

Giao diện tương đối dễ sử dụng, thao tác cơ bản, nhanh gọn thuận tiện cho người dùng. Song ứng dụng còn nhiều hạn chế cần phải khắc phục và phát triển thêm để được hoàn thiện trong tương lai. Ứng dụng web đang được sử dụng backend là Nodejs mang lại nhiều ưu điểm cho ứng dụng web.

1. Về kiến thức

Sau thời gian thực hiện, nghiên cứu đề tài đồ án tốt nghiệp, em đã đạt được nhiều tiến bộ về mặt tìm hiểu, nghiên cứu lý thuyết công nghệ mới và kỹ năng phân tích thiết kế phần mềm. Có thể nói thông qua đề tài này em đã đạt được:

* Hiểu biết nhiều hơn về quy trình phát triển phần mềm và phân tích, thiết kế các biểu đồ của phần mềm.
* Kỹ năng lập trình và giải quyết vấn đề.
* Nâng cao tinh thần tự tìm hiểu, tự nghiên cứu những công nghệ mới.

1. Hạn chế

Do thời gian thực hiện đề tài còn tương đối hạn chế nên không tránh khỏi những sai lầm, thiếu sót. Bên cạnh đó, việc tạo ra sản phẩm ứng dụng hoàn chỉnh cần có một thời gian dài nhất định nên ứng dụng còn một số chức năng mới chưa được hoàn thiện.

Do kinh nghiệm và kiến thức của em chưa nhiều nên trong quá trình tìm hiểu, nghiên cứu còn nhiều khúc mắc, vấn đề mà chưa giải quyết được và sản phẩm còn nhiều lỗi nên mong các thầy cô giáo có thể bỏ qua và đóng góp ý kiến để em được học hỏi, phát triển thêm để có thêm kinh nghiệm bước vào tương lai.

1. Hướng phát triển

Để tiếp tục phát triển đề tài này và có thể triển khai ứng dụng đến người dùng cuối trên môi trường thực tế, em nhận thấy cần phải thực hiện tiếp các công việc như sau:

* Cập nhật những tính năng mới để phục vụ người dùng tốt hơn.
* Thu thập ý kiến phản hồi từ người dùng, tối ưu lại trang web được hiệu quả nhất.
* Cài thiện hiệu năng và sửa lỗi sau quá trình kiểm thử phần mềm.

Đây là một chương trình ứng dụng thực tế lớn và phức tạp, cần có một nhóm các chuyên gia cùng thực hiện. Vì vậy trong phạm vi đồ án này chỉ có thể đưa ra những chức năng cơ bản, mô phỏng một phần nhỏ yêu cầu của thực tế.

Rất mong nhận được sự đóng góp ý kiến của các thầy cô giáo và bạn bè để chương trình được tiếp tục hoàn thiện, đáp ứng một cách tốt nhất cho công tác quản lý hỗ trợ sinh viên tìm kiếm nhà trọ.

***Em xin chân thành cảm ơn!***

**DANH MỤC TÀI LIỆU THAM KHẢO**

Danh mục Website tham khảo

1. Khái niệm ứng dụng web: [https://vi.wikipedia.org/wiki/Ứng\_dụng\_web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web)
2. Sự khác nhau giữa website và webapp: https://toidicodedao.com/2015/08/13/su-khac-biet-giua-web-site-va-web-application
3. Kỹ thuật lập trình ứng dụng web: <https://topdev.vn/blog/frontend-la-gi-backend-la-gi-va-fullstack-la-gi>
4. Web server: https://bizflycloud.vn/tin-tuc/tat-tat-kien-thuc-co-ban-ve-web-server-ban-phai-biet-20180515115521302.htm
5. TCP/IP: <http://notes.viphat.work/tong-quan-ve-tcp-ip-va-http>
6. ReactJS: <https://reactjs.org/>
7. Redux: <https://redux.js.org/>