|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI



họ và tên: Nguyễn VĂN KHÁNH

TÊN ĐỀ TÀI: PHÁT TRIỂN HỆ THỐNG eHOTEL

ĐỂ ĐẶT PHÒNG CHO KHÁCH SẠN

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2021

TRƯỜNG ĐẠI HỌC THỦY LỢI

|  |  |
| --- | --- |
| **bỘ GIÁO DỤC VÀ ĐÀO TẠO** | **BỘ NÔNG NGHIỆP VÀ PTNT** |

TRƯỜNG ĐẠI HỌC THỦY LỢI

HỌ VÀ TÊN: NGUYỄN VĂN KHÁNH

PHÁT TRIỂN HỆ THỐNG eHOTEL ĐỂ ĐẶT PHÒNG CHO KHÁCH SẠN

|  |  |
| --- | --- |
| Ngành : | Công nghệ thông tin |
| Mã số: | 7480201 |

|  |  |
| --- | --- |
| NGƯỜI HƯỚNG DẪN | Tiến sĩ Lê Nguyễn Tuấn Thành |

HÀ NỘI, NĂM 2021

|  |  |
| --- | --- |
|  | CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM  **Độc lập - Tự do - Hạnh phúc**  --------------------  **NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP** |

Họ tên sinh viên: Nguyễn Văn Khánh Hệ đào tạo : Đại học chính quy

Lớp: 58TH3 Ngành: Công nghệ thông tin

Khoa: Công nghệ thông tin

1. **TÊN ĐỀ TÀI:**

PHÁT TRIỂN HỆ THỐNG EHOTEL ĐỂ ĐẶT PHÒNG CHO KHÁCH SẠN

1. **CÁC TÀI LIỆU CƠ BẢN:**

[1] https://spring.io/projects/spring-boot

[2] https://angular.io/

[3] https://material.angular.io/components/categories

[4] https://www.postgresql.org/about/

[5] https://www.tutorialspoint.com/angular\_material/angular\_material\_overview.htm

[6] https://topdev.vn/blog/restful-api-la-gi/

1. **- NỘI DUNG CÁC PHẦN THUYẾT MINH VÀ TÍNH TOÁN:**

|  |  |
| --- | --- |
| **Nội dung cần thuyết minh** | **Tỷ lệ %** |
| Chương 1: Cơ sở lý thuyết | 15% |
| Chương 2: Khảo sát và phân tích bài toán | 30% |
| Chương 3: Thiết kế cơ sở dữ liệu | 30% |
| Chương 4: Tổng kết đánh giá | 25% |

**4 - GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN**

|  |  |
| --- | --- |
| **Phần** | **Họ tên giáo viên hướng dẫn** |
| Chương 1: Cơ sở lý thuyết | TS. Lê Nguyễn Tuấn Thành |
| Chương 2: Khảo sát và phân tích bài toán |
| Chương 3: Thiết kế cở sở dữ liệu |
| Chương 4: Tổng kết đánh giá |

**5 - NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Ngày 20 tháng 03 năm 2021

|  |  |
| --- | --- |
| **Trưởng Bộ môn**  *(Ký và ghi rõ Họ tên)* | **Giáo viên hướng dẫn chính**  *(Ký và ghi rõ Họ tên)* |

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua.

Ngày. . . . tháng. . . . năm 202. . **Chủ tịch Hội đồng**

*(Ký và ghi rõ Họ tên)*

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi ngày 20 tháng 06 năm 2021

**Sinh viên làm Đồ án tốt nghiệp**

*(Ký và ghi rõ Họ tên)*

Nguyễn Văn Khánh

|  |  |
| --- | --- |
| logo | TRƯỜNG ĐẠI HỌC THUỶ LỢI  **KHOA CÔNG NGHỆ THÔNG TIN**  BẢN TÓM TẮT ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP |

TÊN ĐỀ TÀI: **PHÁT TRIỂN HỆ THỐNG EHOTEL ĐỂ ĐẶT PHÒNG CHO KHÁCH SẠN**

*Sinh viên thực hiện*: Nguyễn Văn Khánh

*Lớp*: 58TH3

*Giáo viên hướng dẫn*: TS. Lê Nguyễn Tuấn Thành

**TÓM TẮT ĐỀ TÀI**

Trong bối cảnh nền kinh tế của đất nước ngày càng phát triển, đời sống nhân dân ngày càng được cải thiện, nhu cầu vui chơi, giải trí, du lịch vì vậy ngày càng nhiều. Các dịch vụ du lịch thu hút hàng triệu lượt khách trong và ngoài nước mỗi năm. Điều này tạo điều kiện cho nghành du lịch, đặc biệt là ngành kinh doanh khách sạn phát triển.

Một trong những vấn đề quan tâm của khách du lịch là việc tìm kiếm và đặt chỗ được một phòng phù hợp trong thời gian đi du lịch. Việc đến tận nơi đặt phòng trực tiếp có rất nhiều hạn chế về thời gian và thủ tục, trong nhiều trường hợp là bất khả thi. Do đó, việc cho phép người dùng tìm kiếm thông tin trực tuyến về các khách sạn ở Việt Nam, đồng thời hỗ trợ đặt phòng nhanh chóng, thuận tiện sẽ giúp cải thiện chất lượng dịch vụ và đáp ứng nhu cầu người dùng tốt hơn.

Đồ án hướng tới phát triển hệ thống đặt phòng khách sạn online để hỗ trợ người dùng tốt hơn.

**CÁC MỤC TIÊU CHÍNH**

Kiến thức:

* Frontend: Tìm hiểu về Angular, HTML/CSS, JS, BootStrap
* Backend: Tìm hiểu về Java, Spring Boot, PostgreSQL

Kĩ năng:

* Sử dụng Angular và Spring Boot để xây dựng website đặt phòng khách sạn

**KẾT QUẢ DỰ KIẾN**

* Nắm được kiến thức và công nghệ về frontend và backend
* Ứng dụng kiến thức và công nghệ đã học để xây dựng một website đặt phòng khách sạn.

**TÀI LIỆU THAM KHẢO**

[1] Spring boot: https://spring.io/

[2] Angular: https://angular.io/

LỜI CAM ĐOAN

Tác giả xin cam đoan đây nội dung và kết quả của đồ án tốt nghiệp này là do bản thân tác giả nghiên cứu dưới sự hướng dẫn của TS. Lê Nguyễn Tuấn Thành.Các nội dung trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả xin cam đoan những lời nói trên là sự thật. Nếu nói sai tác giả sẽ hoàn toàn chịu trách nhiệm.

|  |  |
| --- | --- |
|  | **Tác giả ĐATN**  *(Chữ ký)*  **Nguyễn Văn Khánh** |

LỜI CÁM ƠN

Trong thời gian làm đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè. Đặc biệt được sự hướng dẫn của TS. Lê Nguyễn Tuấn Thành và sự đồng ý của Khoa Công nghệ thông tin trường Đại Học Thủy Lợi, em đã được nhận đề tài ***“*PHÁT TRIỂN HỆ THỐNG EHOTEL ĐỂ ĐẶT PHÒNG CHO KHÁCH SẠN”** cho đồ án tốt nghiệp của mình.

Trước hết, cho em gửi lời cảm ơn chân thành đến toàn thể các thầy cô trong trường Đại học Thủy Lợi nói chung và thầy cô trong khoa Công nghệ thông tin nói riêng, những người đã tận tâm chỉ dạy, truyền đạt kiến thức, kĩ năng học tập trong suốt thời gian học tập tại trường. Tạo điều kiện cho em hoàn thành các môn học tại trường và nghiên cứu đồ án tốt nghiệp của mình. Đặc biệt hơn, em xin gửi lời cảm ơn chân thành nhất tới thầy giáo – TS. Lê Nguyễn Tuấn Thành giảng viên khoa Công nghệ thông tin, người đã tận tình hướng dẫn, đóng góp ý kiến và chỉ bảo em suốt quá trình thực hiện đề tài này. Mặc dù đã rất cố gắng hoàn hiện để đạt được kết quả tốt nhất cho đề tài, nhưng không thể tránh khỏi những sai sót trong quá trình thực hiện. Rất mong nhận được những đóng góp của thầy cô, bạn bè để đồ án này được hoàn thiện hơn.

Em xin chân thành cảm ơn!

*Hà Nội, Ngày 20 tháng 06 năm 2021.*

Sinh viên thực hiện

Nguyễn Văn Khánh

MỤC LỤC

[MỤC LỤC iii](#_Toc74317155)

[DANH MỤC HÌNH ẢNH v](#_Toc74317156)

[DANH SÁCH CÁC BẢNG vii](#_Toc74317157)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 1](#_Toc74317158)

[1.1. Thông tin sơ lược. 1](#_Toc74317159)

[1.2. Hệ thống RESTful API 1](#_Toc74317160)

[1.3. Các công nghệ và ứng dụng. 3](#_Toc74317161)

[1.3.1. Ngôn ngữ HTML, CSS và JavaScript. 4](#_Toc74317162)

[1.3.2. Spring boot. 6](#_Toc74317163)

[1.3.3. Angular, Angular Material. 8](#_Toc74317164)

[1.3.4. PostgreSQL. 13](#_Toc74317165)

[CHƯƠNG 2: KHẢO SÁT VÀ PHÂN TÍCH BÀI TOÁN 15](#_Toc74317166)

[2.1. Khảo sát thực tế. 15](#_Toc74317167)

[2.2. Phân tích yêu cầu. 15](#_Toc74317168)

[2.2.1. Sơ đồ Use Case tổng quát. 15](#_Toc74317169)

[2.2.2. Xác định chức năng và vai trò người dùng. 18](#_Toc74317170)

[2.2.3. Phân tích chức năng người dùng. 18](#_Toc74317171)

[CHƯƠNG 3: THIẾT KẾ CƠ SỞ DỮ LIỆU. 33](#_Toc74317172)

[3.1. Biểu đồ lớp. 33](#_Toc74317173)

[3.1.1. Biểu đồ lớp tổng quát của hệ thống: 33](#_Toc74317174)

[3.1.2. Danh sách các lớp đối tượng của hệ thống: 33](#_Toc74317175)

[3.2. Thiết kế các bảng dữ liệu. 34](#_Toc74317176)

[3.2.1. Bảng User. 34](#_Toc74317177)

[3.2.2. Bảng City. 35](#_Toc74317178)

[3.2.3. Bảng Hotel. 35](#_Toc74317179)

[3.2.4. Bảng Room. 36](#_Toc74317180)

[3.2.5. Bảng RoomType. 36](#_Toc74317181)

[3.2.6. Bảng BookingRoom. 37](#_Toc74317182)

[3.2.7. Bảng Role. 37](#_Toc74317183)

[3.2.8. Bảng Permission. 38](#_Toc74317184)

[3.2.9. Bảng Post. 38](#_Toc74317185)

[3.2.10. Bảng Register\_Log. 39](#_Toc74317186)

[CHƯƠNG 4: TỔNG KẾT ĐÁNH GIÁ 40](#_Toc74317187)

[4.1. Kết quả xây dựng website. 40](#_Toc74317188)

[4.1.1. Giao diện đăng nhập. 40](#_Toc74317189)

[4.1.2. Giao diện đăng ký. 40](#_Toc74317190)

[4.1.3. Giao diện trang chủ. 43](#_Toc74317191)

[4.1.4. Giao diện danh sách khách sạn. 44](#_Toc74317192)

[4.1.5. Giao diện chi tiết khách sạn và đặt phòng. 45](#_Toc74317193)

[4.1.6. Giao diện thông tin người dùng. 46](#_Toc74317194)

[4.1.7. Giao diện danh sách đơn đặt phòng của khách hàng. 47](#_Toc74317195)

[4.1.8. Giao diện chi tiết đơn đặt phòng 48](#_Toc74317196)

[4.1.9. Giao diện trang quản trị 49](#_Toc74317197)

[4.2. Đánh giá kết quả 49](#_Toc74317198)

[TÀI LIỆU THAM KHẢO 51](#_Toc74317199)

DANH MỤC HÌNH ẢNH

[Hình 1-1: REST API Design 2](#_Toc74317048)

[Hình 1-2: Cách hoạt động của RESTful API 3](#_Toc74317049)

[Hình 1-3: Văn bản HTML cơ bản 4](#_Toc74317050)

[Hình 1-4: Giao diện trang web Spring Initializr 7](#_Toc74317051)

[Hình 1-5: Giao diện chọn dependencies 8](#_Toc74317052)

[Hình 1-6: Kiểm tra phiên bản Angular CLI 9](#_Toc74317053)

[Hình 1-7: Giao diện mặc định Angular 10](#_Toc74317054)

[Hình 1-8: Lựa chọn chủ đề cho ứng dụng 11](#_Toc74317055)

[Hình 1-9: Lựa chọn cài đặt HammerJS 11](#_Toc74317056)

[Hình 1-10: Lựa chọn cài đặt Browser Animations 12](#_Toc74317057)

[Hình 1-11: Công cụ quản lý cơ sở dữ liệu PgAdmin4 13](#_Toc74317058)

[Hình 2-1: Sơ đồ Use Case hệ thống 16](#_Toc74317059)

[Hình 2-2: Biểu đồ tuần tự chức năng đăng nhập 19](#_Toc74317060)

[Hình 2-3: Biểu đồ hoạt động chức năng đăng nhập 20](#_Toc74317061)

[Hình 2-4: Biểu đồ tuần tự chức năng đăng ký 21](#_Toc74317062)

[Hình 2-5: Biểu đồ hoạt động chức năng đăng ký 22](#_Toc74317063)

[Hình 2-6: Biểu đồ tuần tự chức năng đặt phòng 23](#_Toc74317064)

[Hình 2-7: Biểu đồ hoạt động chức năng đặt phòng. 24](#_Toc74317065)

[Hình 2-8: Biểu đồ tuần tự chức năng hủy đặt phòng. 25](#_Toc74317066)

[Hình 2-9: Biểu đồ chức năng hủy đặt phòng 26](#_Toc74317067)

[Hình 2-10: Biểu đồ tuần tự chức năng thay đổi thông tin. 27](#_Toc74317068)

[Hình 2-11: Biểu đồ hoạt động thay đổi thông tin. 28](#_Toc74317069)

[Hình 2-12: Biểu đồ tuần tự chức năng đổi mật khẩu. 29](#_Toc74317070)

[Hình 2-13: Biểu đồ hoạt động chức năng đổi mật khẩu. 30](#_Toc74317071)

[Hình 2-14: Biểu đồ tuần tự chức năng xác thực email 31](#_Toc74317072)

[Hình 2-15: Biểu đồ hoạt động xác thực email 32](#_Toc74317073)

[Hình 3-1: Biểu đồ lớp tổng quát 33](#_Toc74317074)

[Hình 4-1: Giao diện đăng nhập 40](#_Toc74317075)

[Hình 4-2: Giao diện đăng ký 41](#_Toc74317076)

[Hình 4-3: Giao diện xác thực email sau khi đăng ký 42](#_Toc74317077)

[Hình 4-4: Giao diện trang chủ 43](#_Toc74317078)

[Hình 4-5: Giao diện danh sách khách sạn 44](#_Toc74317079)

[Hình 4-6: Giao diện chi tiết và đặt phòng 45](#_Toc74317080)

[Hình 4-7: Giao diện thông tin người dùng 46](#_Toc74317081)

[Hình 4-8: Giao diện danh sách đơn đặt phòng 47](#_Toc74317082)

[Hình 4-9: Giao diện chi tiết đơn đặt phòng 48](#_Toc74317083)

[Hình 4-10: Giao diện trang quản trị 49](#_Toc74317084)

DANH SÁCH CÁC BẢNG

[Bảng 2-1: Danh sách Actor hệ thống 16](#_Toc74317085)

[Bảng 2-2: Danh sách Use Case hệ thống 17](#_Toc74317086)

[Bảng 3- 1: Bảng các lớp đối tượng 34](#_Toc74317087)

[Bảng 3- 2: Bảng dữ liệu User 34](#_Toc74317088)

[Bảng 3- 3: Bảng dữ liệu City 35](#_Toc74317089)

[Bảng 3- 4: Bảng dữ liệu Hotel 35](#_Toc74317090)

[Bảng 3- 5: Bảng dữ liệu Room 36](#_Toc74317091)

[Bảng 3- 6: Bảng dữ liệu RoomType 37](#_Toc74317092)

[Bảng 3- 7: Bảng dữ liệu BookingRoom 37](#_Toc74317093)

[Bảng 3- 8: Bảng dữ liệu Role 37](#_Toc74317094)

[Bảng 3- 9: Bảng dữ liệu Permission 38](#_Toc74317095)

[Bảng 3- 10: Bảng dữ liệu Post 38](#_Toc74317096)

[Bảng 3- 11: Bảng dữ liệu Register\_Log 39](#_Toc74317097)

1. CƠ SỞ LÝ THUYẾT
   1. Thông tin sơ lược.

Hệ thống được xây dựng trên môi trường web bằng các công nghệ sau:

* Java (back-end).
* Angular (front-end).
* Và một số công nghệ hỗ trợ khác như: javascript, bootstrap.v.v.

Hệ thống cần đáp ứng được:

* Về người dùng: Giao diện thân thiện với người dùng, dễ dàng thao tác, đáp ứng được nhu cầu của khách hàng.
* Về hiệu năng: Tốc độ xử lý nhanh và chính xác, hạn chế lỗi.
* Về bảo mật: Tài khoản của người dùng là duy nhất, thông tin người dùng, đơn đặt phòng được bảo mật.
* Về hệ thống: Website được xây dựng trên các module, framework được dựng sẵn, dễ dàng nâng cấp bảo trì, tối ưu hóa hệ thống.
  1. Hệ thống RESTful API

**RESTful API** là một tiêu chuẩn dùng trong việc thiết kế API cho các ứng dụng web (thiết kế Web services) để tiện cho việc quản lý các resource. Nó chú trọng vào tài nguyên hệ thống (tệp văn bản, ảnh, âm thanh, video, hoặc dữ liệu động…), bao gồm các trạng thái tài nguyên được định dạng và được truyền tải qua HTTP.



Hình 1-1: REST API Design

* Diễn giải các thành phần:
* **API** (**A**pplication **P**rogramming **I**nterface) là một tập các quy tắc và cơ chế mà theo đó, một ứng dụng hay một thành phần sẽ tương tác với một ứng dụng hay thành phần khác. API có thể trả về dữ liệu mà bạn cần cho ứng dụng của mình ở những kiểu dữ liệu phổ biến như [JSON](https://topdev.vn/blog/json-la-gi/) hay XML.
* **REST** (**RE**presentational **S**tate **T**ransfer) 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, PUT, 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**.
* Cách RESTful hoạt động:



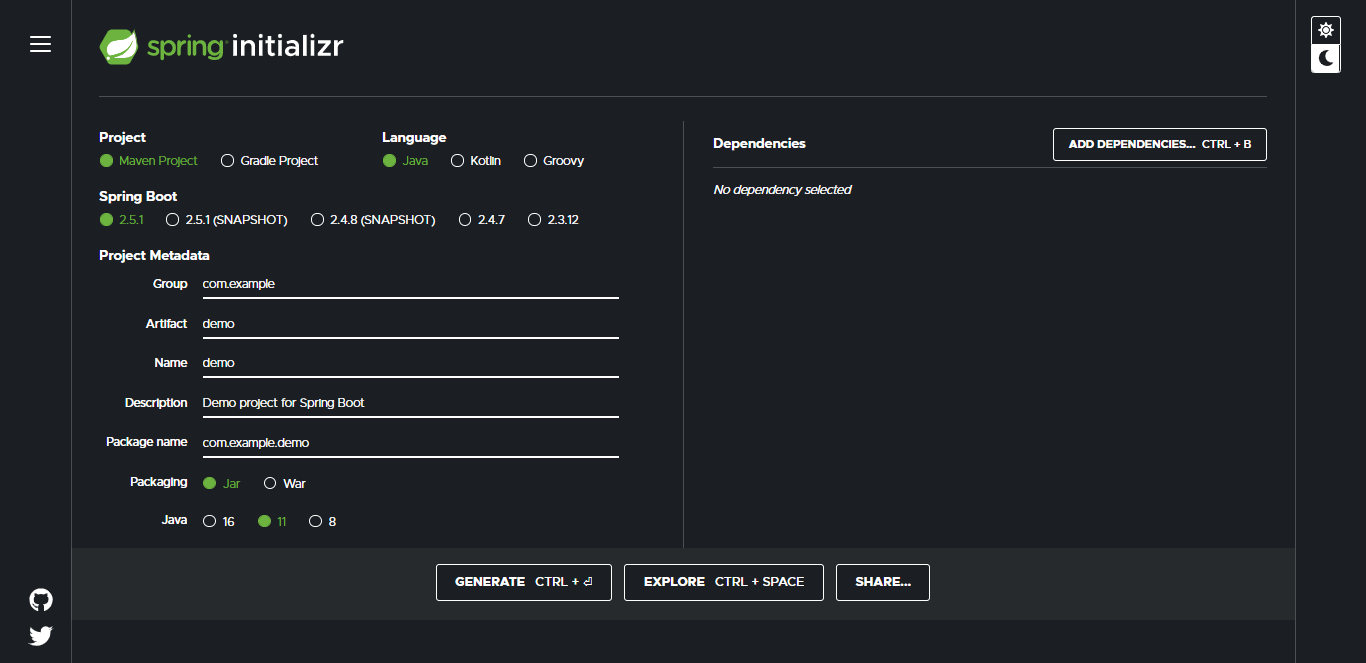
Hình 1-2: Cách hoạt động của RESTful API

* 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. Các công nghệ và ứng dụng.
* HTML, CSS và JavaScript
* Spring boot
* Angular, Angular Material
* PostgreSQL
  + 1. Ngôn ngữ HTML, CSS và JavaScript.
       1. HTML.
* HTML là viết tắt của Hypertext Markup Language là ngôn ngữ đánh dấu siêu văn bản dùng để xây dựng các Website.
* **HTML** được tạo ra bởi Tim Berners-Lee, một nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Hiện nay, HTML đã trở thành một chuẩn Internet được tổ chức W3C (World Wide Web Consortium) vận hành và phát triển.
* HTML bao gồm một loạt các [phần tử](https://developer.mozilla.org/en-US/docs/Glossary/Element) , mà bạn sử dụng để bao bọc hoặc bọc các phần khác nhau của nội dung để làm cho nội dung xuất hiện theo một cách nhất định hoặc hoạt động theo một cách nhất định. Các [thẻ](https://developer.mozilla.org/en-US/docs/Glossary/Tag) bao quanh có thể tạo siêu liên kết từ hoặc hình ảnh đến một nơi khác, có thể in nghiêng các từ, có thể làm cho phông chữ lớn hơn hoặc nhỏ hơn, v.v.
* Một số thẻ cơ bản của HTML: *<html>,<body>,<head>,<title>,<a>,<img>..*
* Ví dụ về một văn bản HTML đơn giản:



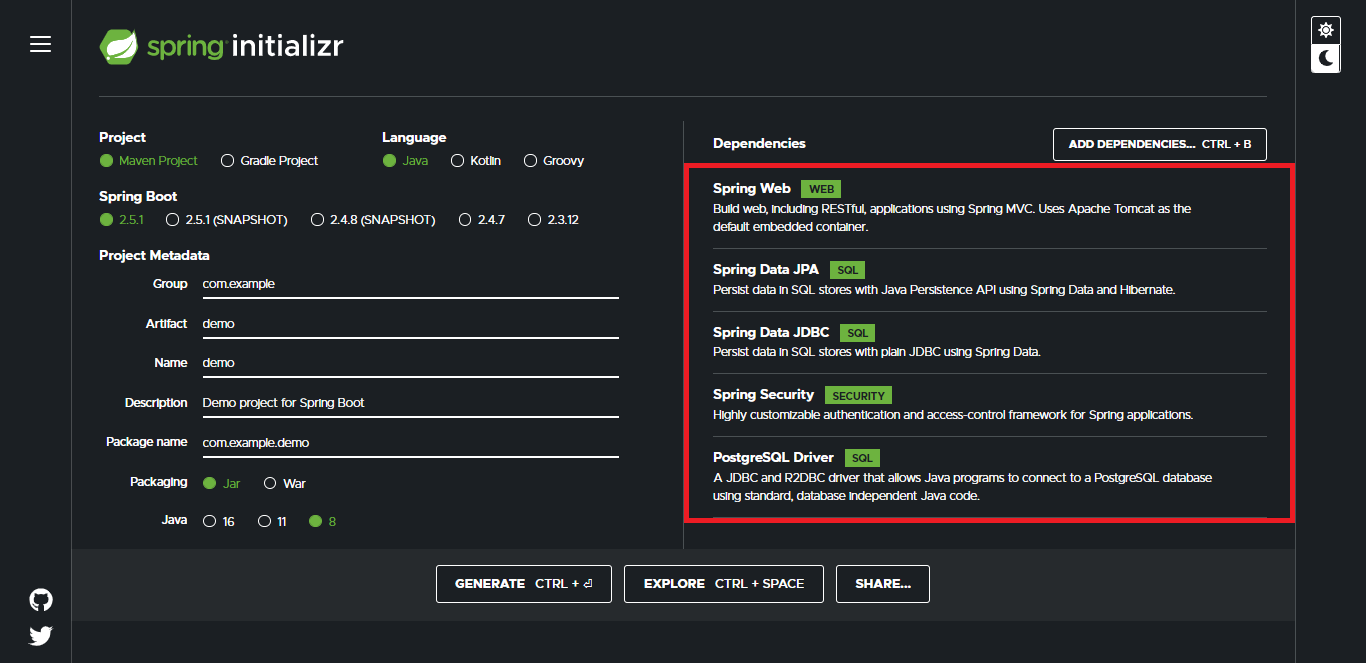
Hình 1-3: Văn bản HTML cơ bản

* + - 1. CSS
* CSS là ngôn ngữ tạo phong cách cho Website – Cascading Style Sheet Lenguage. Nó dùng để tạo phong cách và định kiểu cho các yếu tố được viết dưới dạng ngôn ngữ đánh dấu như là HTML.
* Do HTML không được thiết kế để gắn tag để giúp định dạng cho web, để giải quyết vấn đề này CSS được ra đời phát triển bởi **W3C** ([World Wide Web Consortium](https://www.w3.org/" \t "_blank)) vào năm 1996, giúp loại bỏ việc định dạng kiểu cách khỏi trang HTML và được thêm và trang HTML bằng một số cách sau:
* Định nghĩa kiểu cách trong tệp tin .css
* Sử dụng phẩn tử *<style>* trong các thẻ của HTML
* Mối tương quan giữa HTML và CSS rất mật thiết. HTML là ngôn ngữ đánh dấu và CSS định hình phong cách (tất cả những gì tạo nên giao diện website), chúng là không thể tách rời. CSS về lý thuyết không có cũng được, nhưng khi đó website sẽ không chỉ là một trang chứa văn bản mà không có gì khác.
  + - 1. JavaScript
* JavaScript theo phiên bản hiện hành, là một ngôn ngữ lập trình thông dịch được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web cũng như phía máy chủ.
* JavaScript được tạo bởi Brandan Eich, một nhân viên của Netscape, vào tháng 9 năm 1995. Được đặt tên đầu tiên là Mocha, tên của nó được đổi thành Mona rồi LiveScript trước khi thật sự trở thành JavaScript nổi tiếng như bây giờ.
* Nó còn là một ngôn ngữ lập trình phổ biến sử dụng trên 92% nền tảng website hiện nay, **JavaScript** đã thể hiện vai trò quan trọng với lĩnh vực này và xuất sắc trở thành công cụ đắc lực hàng đầu của các lập trình viên
* JavaScript vượt trội hơn hẳn những nền tảng lập trình khác ở khả năng tối ưu website. Thứ nhất, JavaScript tăng khả năng tương tác, giúp trang web chuyên nghiệp, sinh động hơn, dĩ nhiên, từ đó những người truy cập cũng dễ dàng tiếp cận hơn. Thứ hai, đây được xem là công cụ “dễ dãi” nhất đối với phần lớn các trình duyệt web và các thiết bị điện tử nhờ sở hữu ngôn ngữ dễ đọc, cú pháp linh hoạt. Cuối cùng, mặc dù không mang tính chất là một ngôn ngữ đa năng, tuy nhiên, JavaScript khai thác tối đa những các tính năng của nó và không ngừng cập nhật mới.
  + 1. Spring boot.
* **Spring Boot** là một dự án phát triển bởi **JAV**(ngôn ngữ java) trong hệ sinh thái Spring framework. Mục đích ra đời của chương trình này nhằm giúp cho các lập trình viên dễ dàng làm cho quá trình lập trình ứng dụng thông qua spring được đơn giản hóa, tập trung chủ yếu vào việc phát triển ứng dụng ở mảng Business.
* Spring Boot cung cấp khả năng bảo trì và tạo REST API dễ dàng, chỉ cần chú thích @RestController và @RequestMapping trên bộ lớp điều khiển hoạt động. Dễ dàng trong việc tích hợp với các hệ sinh thái của Spring như: Spring JDBC, Spring ORM, Spring Data, Spring Securit..v.v
* Spring Boot cung cấp công cụ CLI (command Line Interface) cho việc phát triển và test ứng dụng nhanh chóng và dễ dàng từ command line và rất nhiều các plugins để phát triển và test các ứng dụng Spring Boot nhanh chóng sử dụng các công cụ Build như Maven và Gradle.
* Tạo một dự án Spring boot:
* Spring Boot có một công cụ giúp chúng ta nhanh chóng khởi tạo project gọi là Spring Initializr. Spring Initializr có thể truy cập trên web tại [http://start.spring.io](http://start.spring.io/).



Hình 1-4: Giao diện trang web Spring Initializr

* Như hình trên, chúng ta sẽ khai báo thông tin dự án như:
* Loại project: chọn loại package manager Maven hoặc Gradle.
* Ngôn ngữ: Chọn ngôn ngữ code (java)
* Phiên bản Spring boot: Các version có SNAPSHOT là bản chưa ổn định, không nên chọn.
* Thông tin dự án: Điền theo thông tin dự án.
* Loại file được build: nên chọn JAR để đỡ cấu hình Tomcat server.
* Phiên bản Java: nên chọn java 8 để ổn định.
* Lựa chọn các dependencies cho dự án.

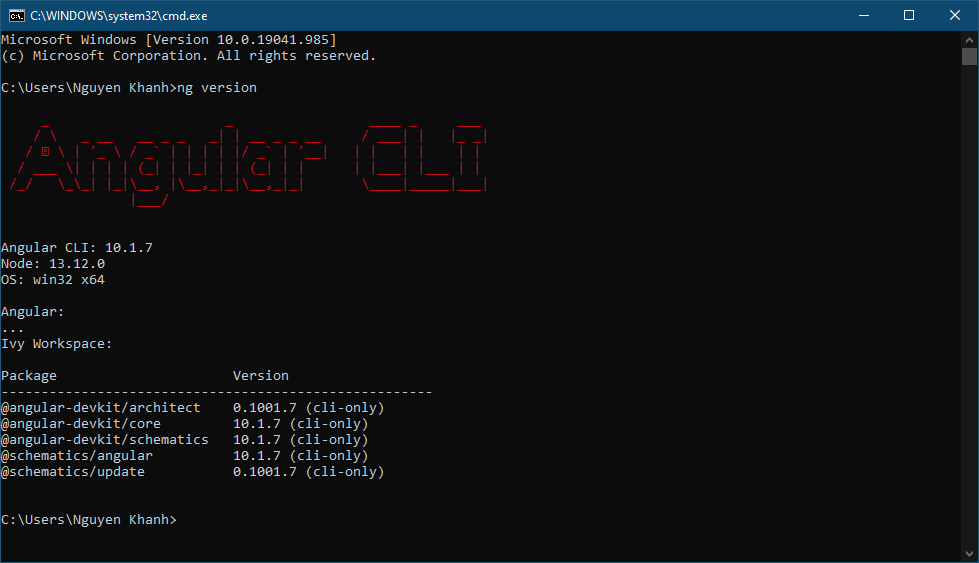


Hình 1-5: Giao diện chọn dependencies

* Sau khi chọn dependencies xong chọn Generate một file zip chứa source code sẽ được tải về.
  + 1. Angular, Angular Material.
       1. Angular.
* Angular là một javascript framework do google phát triển để xây dựng các Single Page Application (SPA) bằng JavaScript , HTML và TypeScript . Angular cung cấp các tính năng tích hợp cho animation , http service và có các tính năng như auto-complete , navigation , toolbar , menus ,… Code được viết bằng TypeScript , biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.
* Có 2 phiên bản của Angular là:
* AngularJs (Angular 1): Phiên bản đầu tiền của angular ra đời năm 2009 và đc ra mắt vào 20/10/2010,được viết bằng JavaScript do lập trình viên Misko Hevery tại Google viết.
* Angular (Angular 2 đến Angular 12): Ra đời vào năm 2016 được viết hoàn toàn bằng TypeScript phiên bản nâng cấp của JavaScript khác hoàn toàn so với AngularJs.
* Angular là 1 Framework rất mạnh mẽ mã nguồn mở hoàn toàn miễn phí, được các lập trình viên trên thế giới ưa chuộng và sử dụng.
* Nó cung cấp Angular CLI một công cụ được phát triển đề chạy trên giao diện cửa sổ ứng dụng (command line interface hay CLI) nhằm giúp đỡ việc tạo dự án, quản lý tập tin trong dự án và thực hiện nhiều tác vụ khác nhau như test (kiểm thử), bundle và deploy dự án một cách nhanh chóng và hiệu quả.
* Cài đặt Angular 10:
* Trước tiên cần cài đặt Angular CLI, mở cử sổ console và chạy lệnh sau:

npm install -g @angular/cli

* Kiểm tra cài đặt thành công và số phiên bản bằng cách chạy lệnh sau: ng version



Hình 1-6: Kiểm tra phiên bản Angular CLI

* Tạo project với Angular CLI bằng cách chạy lệnh sau:

ng new frontend

* Điều hướng đến thư mục làm việc bằng lệnh:

cd frontend

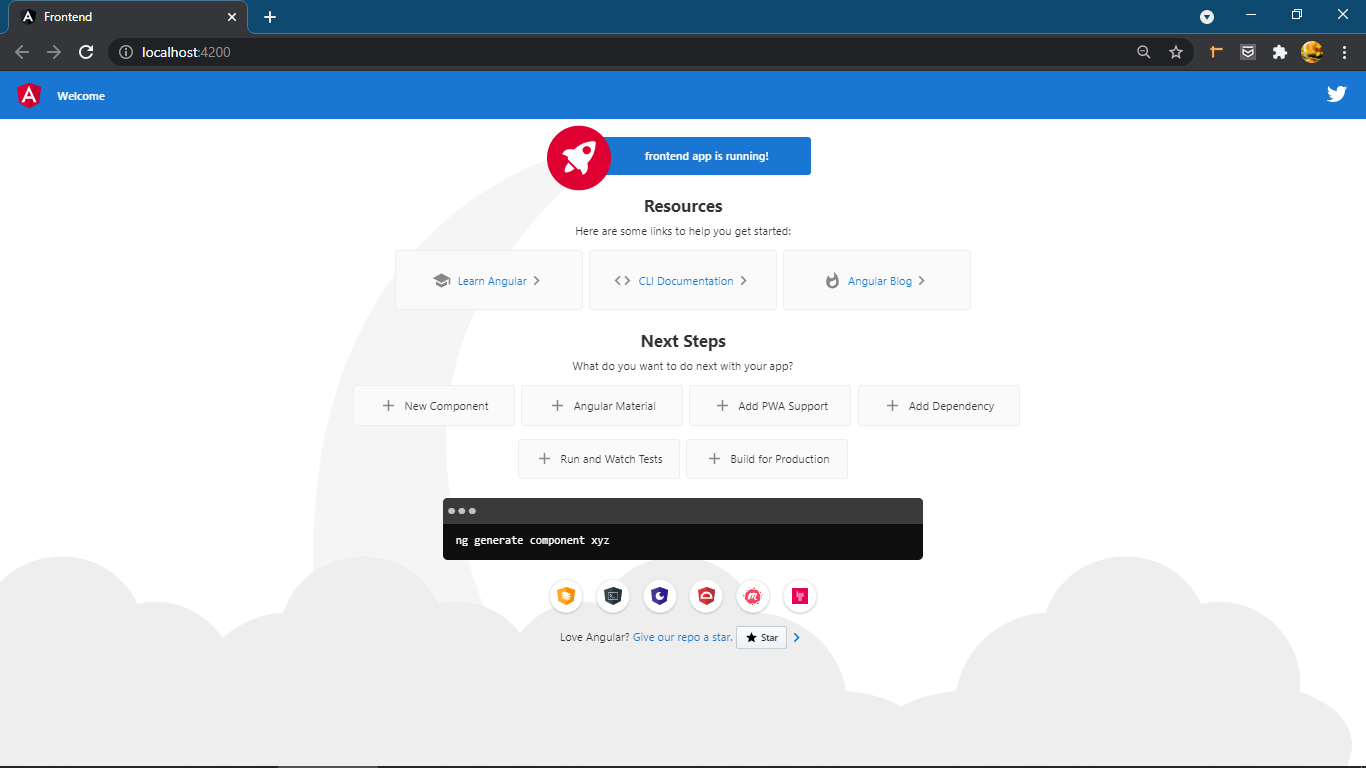
* Chạy project bằng lệnh:

ng serve

* Mặc định project sẽ được chạy trên địa chỉ http://localhost:4200. Có thể thay đổi port mặc định nhờ tham số --port. Ví dụ muốn đổi về port 4201 chạy lệnh:

ng serve --port 4201

* Chạy thành công sẽ hiển thị giao diện như sau:

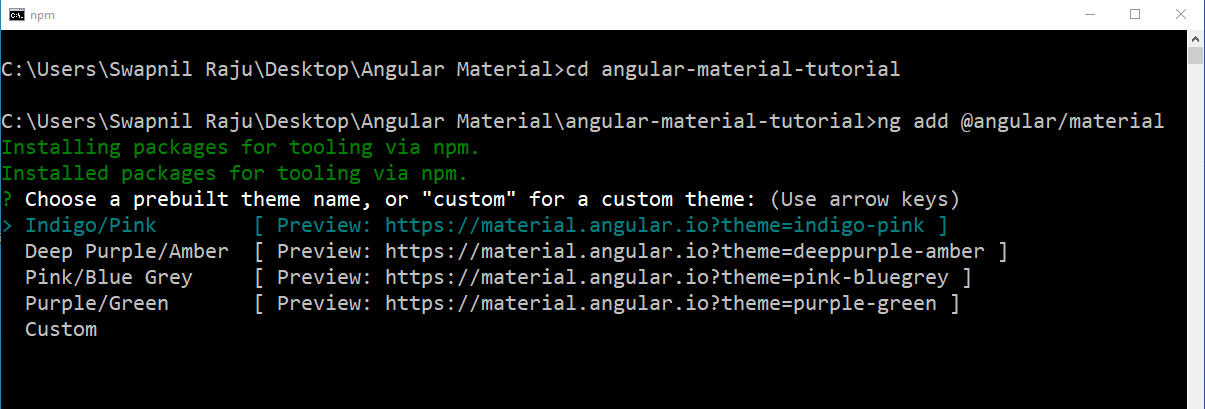


Hình 1-7: Giao diện mặc định Angular

* + - 1. Angular Material.
* Angular Material là một thư viện thành phần UI dành cho các nhà phát triển Angular. Các thành phần giao diện người dùng có thể tái sử dụng của Angular Material giúp xây dựng các trang web và ứng dụng web hấp dẫn, nhất quán nhanh chóng, linh hoạt và thậm chí là đáp ứng thiết kế cho trang web.
* Cài đặt Angular Material:
* Sử dụng Angular CLI để chạy lệnh sau:

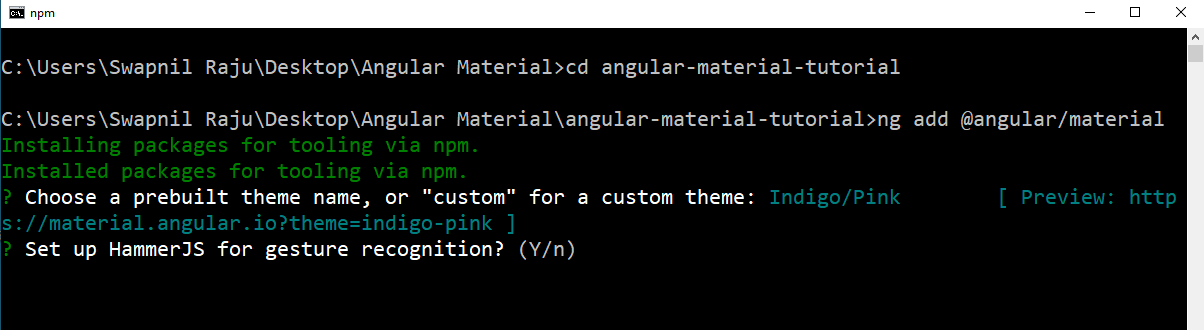
ng add @angular/material

* Đầu tiên, nó sẽ yêu cầu bạn chọn một tên chủ đề dựng sẵn hoặc một chủ đề tùy chỉnh:



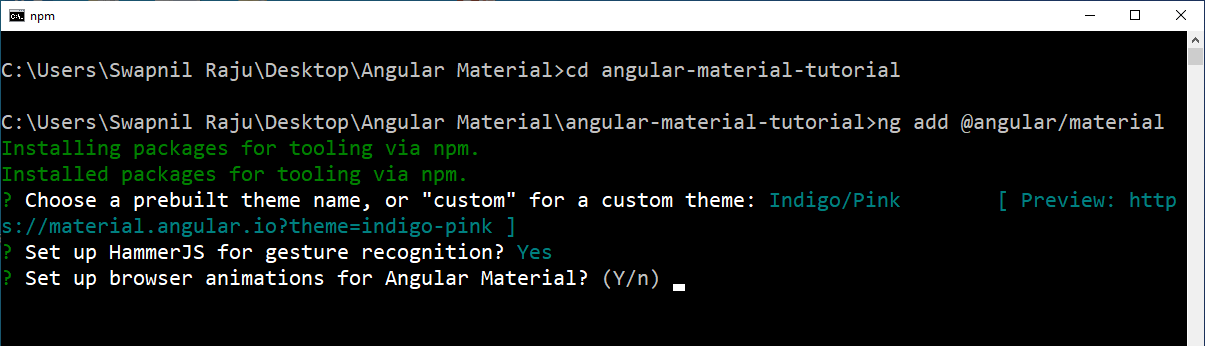
Hình 1-8: Lựa chọn chủ đề cho ứng dụng

* Tiếp theo, nó sẽ yêu cầu bạn thiết lập HammerJS:



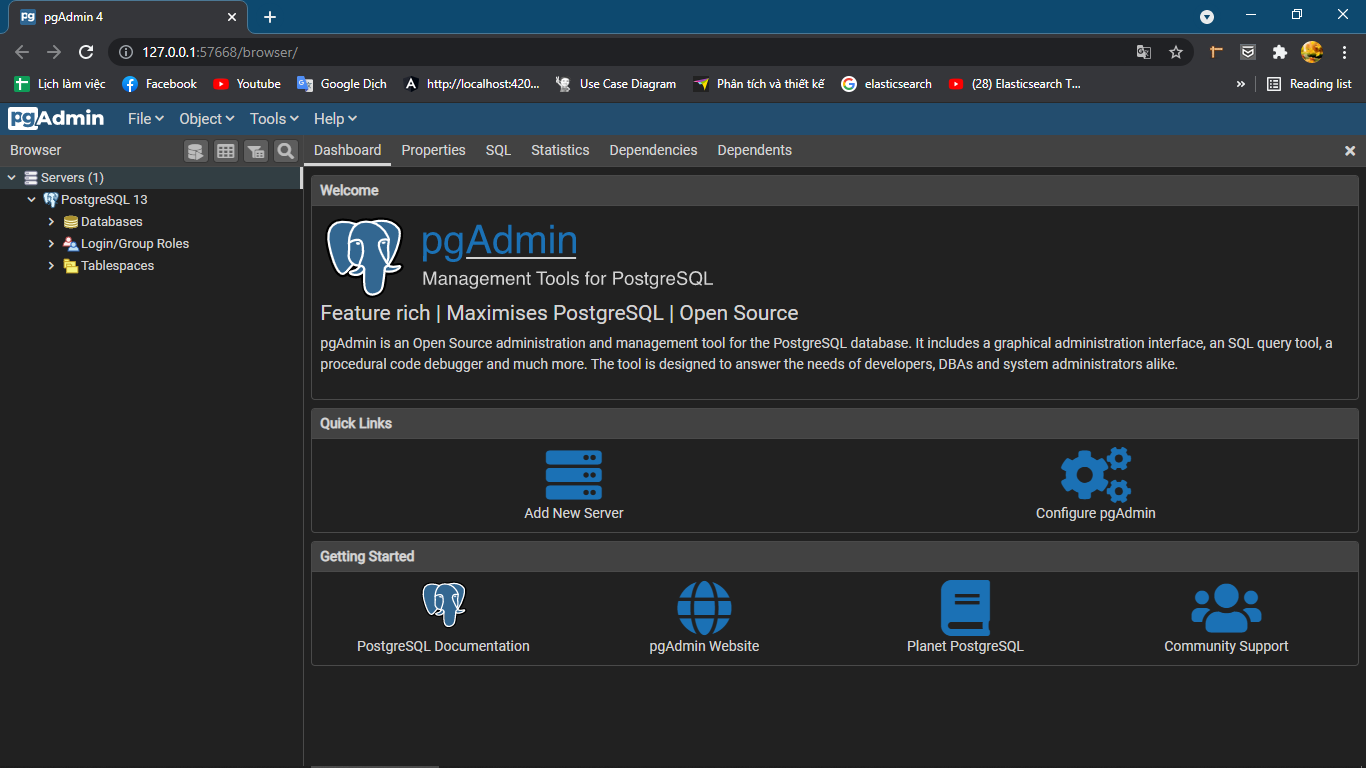
Hình 1-9: Lựa chọn cài đặt HammerJS

* Tiếp theo nó sẽ yêu cầu bạn thiết lập Browser Animations:



Hình 1-10: Lựa chọn cài đặt Browser Animations

* Sau đó, Angular Materials sẽ được cài đặt trong ứng dụng của bạn.
  + 1. PostgreSQL.
* PostgreSQL là một hệ thống cơ sở dữ liệu quan hệ đối tượng (object-relational database management system) hệ thống cơ sở dữ liệu mã nguồn mở tiên tiến nhất hiện nay, sử dụng và mở rộng ngôn ngữ SQL kết hợp với nhiều tính năng giúp lưu trữ và chia tỷ lệ một cách an toàn các khối lượng công việc dữ liệu phức tạp nhất.
* Nguồn gốc của PostgreSQL có từ năm 1986 như một phần của dự án [POSTGRES](https://www.postgresql.org/docs/current/history.html) tại Đại học California ở Berkeley và đã có hơn 30 năm phát triển tích cực trên nền tảng cốt lõi.



Hình 1-11: Công cụ quản lý cơ sở dữ liệu PgAdmin4

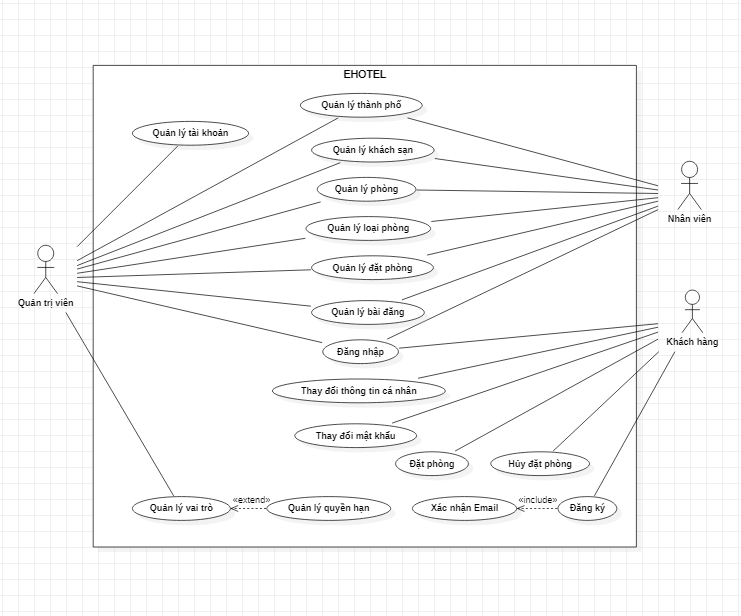
* PostgreSQL đã tạo được danh tiếng mạnh mẽ nhờ kiến ​​trúc đã được chứng minh, độ tin cậy, tính toàn vẹn của dữ liệu, bộ tính năng mạnh mẽ, khả năng mở rộng và sự cống hiến của cộng đồng nguồn mở đằng sau phần mềm để liên tục cung cấp các giải pháp hiệu quả và sáng tạo. PostgreSQL chạy trên [tất cả các hệ điều hành chính](https://www.postgresql.org/download/) , đã tuân thủ [ACID](https://en.wikipedia.org/wiki/ACID) từ năm 2001 và có các tiện ích bổ sung mạnh mẽ như bộ mở rộng cơ sở dữ liệu không gian địa lý [PostGIS](https://postgis.net/) phổ biến . Không có gì ngạc nhiên khi PostgreSQL đã trở thành cơ sở dữ liệu quan hệ nguồn mở được nhiều người và tổ chức lựa chọn.
* PostgreSQL đi kèm với [nhiều tính năng](https://www.postgresql.org/about/featurematrix/) nhằm giúp các nhà phát triển xây dựng ứng dụng, quản trị viên để bảo vệ tính toàn vẹn của dữ liệu và xây dựng môi trường chịu lỗi, đồng thời giúp bạn quản lý dữ liệu của mình bất kể tập dữ liệu lớn hay nhỏ. Ngoài việc [miễn phí và mã nguồn mở](https://www.postgresql.org/about/license/) , PostgreSQL có khả năng mở rộng cao.
* PostgreSQL cố gắng tuân thủ [tiêu chuẩn SQL](https://www.postgresql.org/docs/current/features.html) trong đó sự phù hợp đó không mâu thuẫn với các tính năng truyền thống hoặc có thể dẫn đến các quyết định kiến ​​trúc kém. Nhiều tính năng theo yêu cầu của tiêu chuẩn SQL được hỗ trợ, mặc dù đôi khi có cú pháp hoặc chức năng hơi khác nhau. Có thể mong đợi những động thái tiếp theo hướng tới sự tuân thủ theo thời gian. Kể từ bản phát hành phiên bản 13 vào tháng 9 năm 2020, PostgreSQL tuân thủ ít nhất 170 trong số 179 tính năng bắt buộc đối với sự tuân thủ của SQL: 2016 Core. Kể từ khi viết bài này, không có cơ sở dữ liệu quan hệ nào đáp ứng đầy đủ sự phù hợp với tiêu chuẩn này.

1. KHẢO SÁT VÀ PHÂN TÍCH BÀI TOÁN
   1. Khảo sát thực tế.

Hiện nay nhu cầu du lịch của con người tăng cao, đi kèm theo đó là các dịch vụ khách sạn, nhà nghỉ là nhu cầu tất yếu. Nhưng với những thủ tục đặt phòng nhận phòng tại trực tiếp quầy lễ tân còn nhiều hạn chế nên chúng tôi đã phát triển hệ thống Ehotel để giải đáp cho vấn đề này.

Hệ thống Ehotel cho phép người dùng tìm kiếm tra cứu thông tin phòng ở tại các khách sạn , địa điểm du lịch trên cả nước, đặt phòng trực tuyến với các thao tác đơn giản, giảm thời gian làm việc trực tiếp tại các khách sạn, nhà nghỉ nhanh chóng tiện lợi. Giúp người dùng tối ưu được thời gian lựa chọn địa điểm, chỗ nghỉ phù hợp.

* 1. Phân tích yêu cầu.
     1. Sơ đồ Use Case tổng quát.
* Sơ đồ Use Case tổng quát:



Hình 2-1: Sơ đồ Use Case hệ thống

* Danh sách các Actor:

Bảng 2-1: Danh sách Actor hệ thống

|  |  |  |
| --- | --- | --- |
| STT | Actor | Mô tả |
| 1 | Quản trị viên | Actor này có quyền quản lý tất cả các chức năng và kiểm soát hệ thống |
| 2 | Nhân viên | Actor này có quyền quản lý các chức năng trong phạm vi quyền của nhân viên |
| 3 | Khách hàng | Actor này có quyền sử dụng các chức năng của người dùng trong hệ thống |

* Dánh sách và đặc tả Use Case

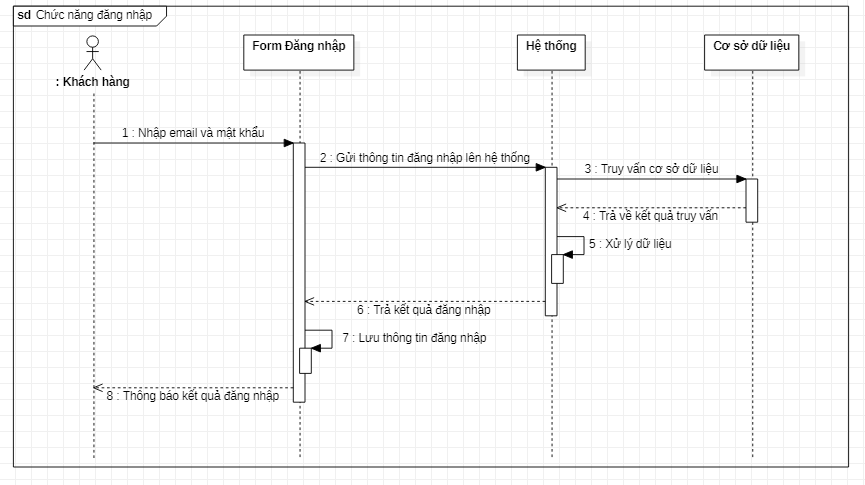
Bảng 2-2: Danh sách Use Case hệ thống

|  |  |  |
| --- | --- | --- |
| STT | Tên Use Case | Mô tả |
| 1 | Quản lý Tài khoản | Giám đốc quản lý tài khoản trên hệ thống |
| 2 | Quản lý Vai trò | Quản trị viên quản lý vai trò trên hệ thống |
| 3 | Quản lý Quyền hạn | Quản trị viên quản lý quyền hạn trên hệ thống |
| 4 | Quản lý Thành phố | Quản trị viên quản lý thành phố trên hệ thống |
| 5 | Quản lý Khách sạn | Quản trị viên và nhân viên quản lý hệ thống khách sạn trên hệ thống |
| 6 | Quản lý Loại phòng | Quản trị viên và nhân viên quản lý hệ thống loại phòng khách sạn trên hệ thống |
| 7 | Quản lý Phòng | Quản trị viên và nhân viên quản lý hệ thống phòng khách sạn trên hệ thống |
| 8 | Quản lý Bài đăng | Quản trị viên và nhân viên quản lý bài đăng trên hệ thống |
| 9 | Quản lý Đặt phòng | Quản trị viên và nhân viên quản lý đơn đặt phòng của người dùng trên hệ thống |
| 10 | Đăng nhập | Cho phép tất cả các tài khoản đăng nhập vào hệ thống |
| 11 | Đăng ký | Cho phép người dùng đăng ký tài khoản vào hệ thống |
| 12 | Đổi mật khẩu | Cho phép người dùng thay đổi mật khẩu tài khoản của mình |
| 13 | Đặt phòng | Người dùng sử dụng chức năng này để đặt phòng trên hệ thống |
| 14 | Hủy đặt phòng | Người dùng có thể hủy đơn đặt phòng nếu thấy chưa phù hợp |
| 15 | Xác nhận Email | Sau khi người dùng đăng ký tài khoản thì bắt buộc phải xác thực email mới kích hoạt được tài khoản |

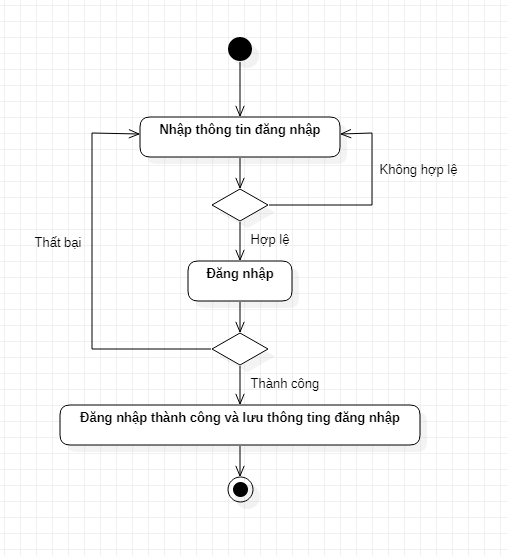
* + 1. Xác định chức năng và vai trò người dùng.
* Khách hàng là đối tượng người dùng duy nhất trong hệ thống.
* Người dùng có thể sử dụng các chức năng để đặt phòng, tra cứu địa điểm, khách sạn tại những địa danh du lịch,.
* Người dùng sẽ có một số chức năng sau trong hệ thống:
* Tìm kiếm thông tin địa điểm, khách sạn.
* Đăng nhập, đăng ký tài khoản.
* Đặt phòng, hủy đặt phòng.
* Thay đổi thông tin cá nhân, mật khẩu.
  + 1. Phân tích chức năng người dùng.
       1. Chức năng đăng nhập.

Mô tả: Chức năng này cho phép người dùng đăng nhập vào hệ thống.

Quy trình thực thi: Khi người dùng nhập thông tin và thực hiện đăng nhập, hệ thống sẽ kiểm tra thông tin người dùng nhập. Nếu thông tin hợp lệ, hệ thống sẽ tiến hành đăng nhập và lấy thông tin người dùng từ server, xử lý và lưu thông tin đăng nhập và thông báo đăng nhập thành công cho người dùng.



Hình 2-2: Biểu đồ tuần tự chức năng đăng nhập

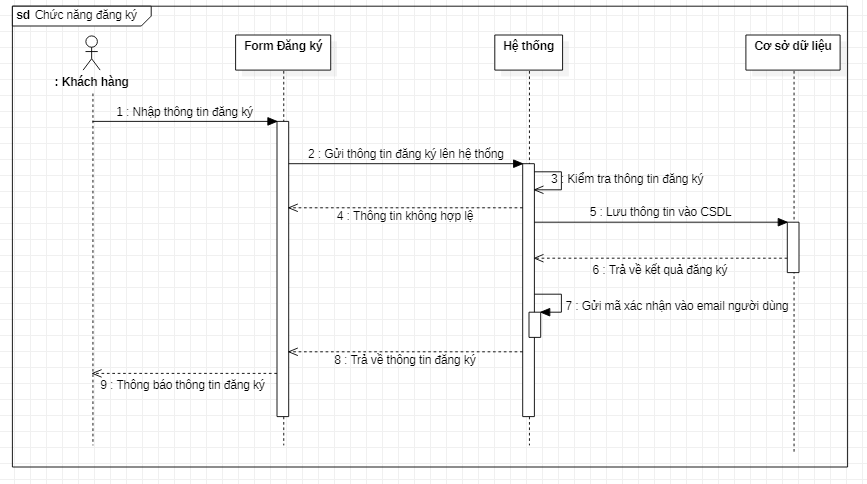


Hình 2-3: Biểu đồ hoạt động chức năng đăng nhập

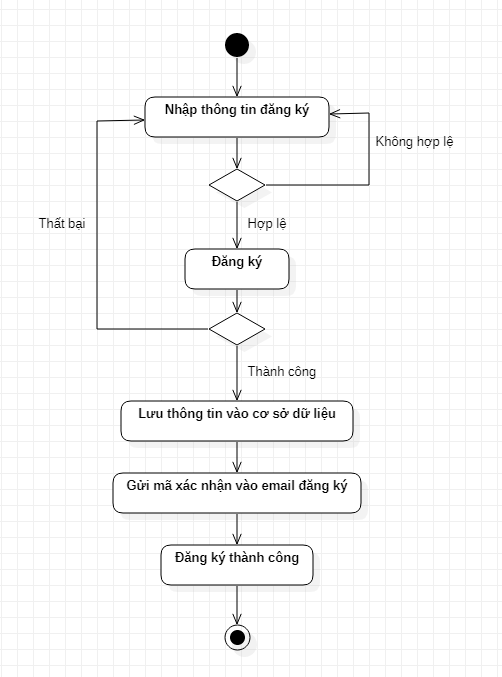
* + - 1. Chức năng đăng ký.

Mô tả: Chức năng này cho phép người dùng đăng ký tài khoản vào hệ thống.

Quy trình thực thi: Khi người dùng nhập thông tin đăng ký và tiến hành đăng ký. Nếu thông tin hợp lệ, hệ thống sẽ thực hiện đăng ký tài khoản mới và gửi link xác nhận vào email người dùng vừa đăng ký và thông báo đăng ký thành công. Người dùng phải xác nhận email để kích hoạt tài khoản mới có thể đăng nhập vào thế thống.



Hình 2-4: Biểu đồ tuần tự chức năng đăng ký

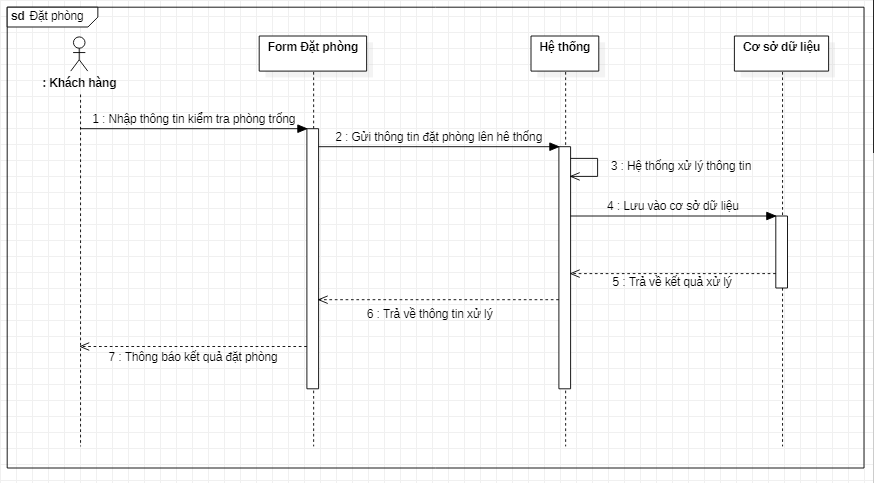


Hình 2-5: Biểu đồ hoạt động chức năng đăng ký

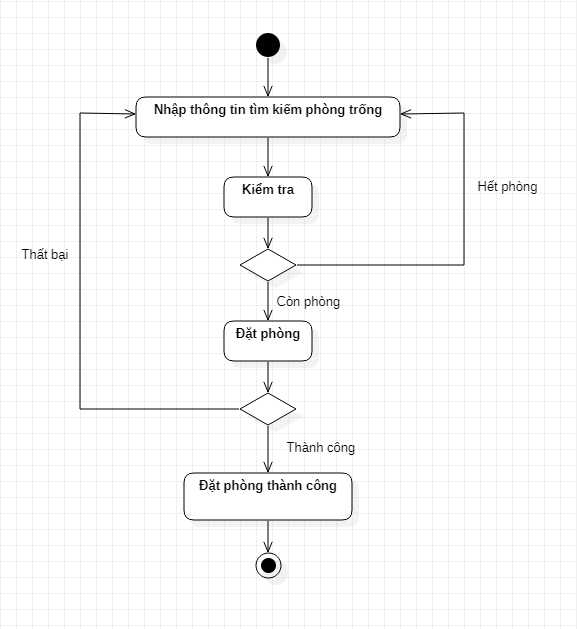
* + - 1. Chức năng đặt phòng.

Mô tả: Chức năng cho phép người dùng đặt phòng trên hệ thống.

Quy trình thực thi: Người dùng sẽ tìm kiếm phòng trên hệ thống và lọc theo thời gian, khách sạn. Sau khi tìm được phòng trống có thể tiến hành đặt phòng.



Hình 2-6: Biểu đồ tuần tự chức năng đặt phòng

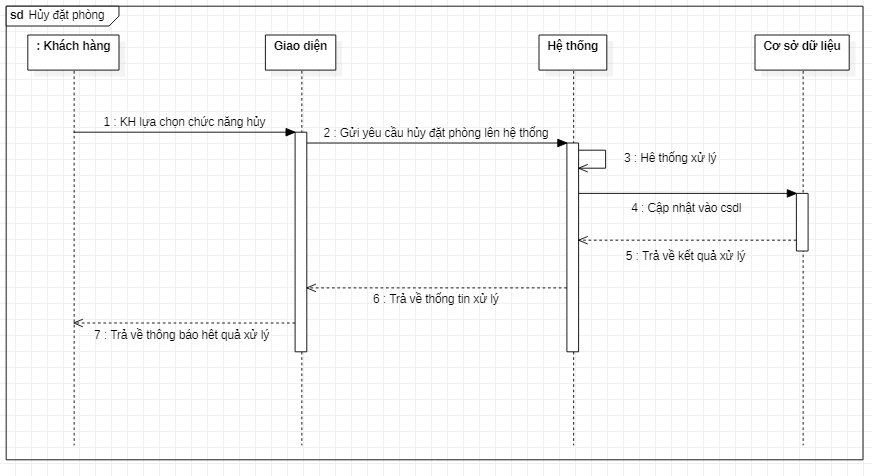


Hình 2-7: Biểu đồ hoạt động chức năng đặt phòng.

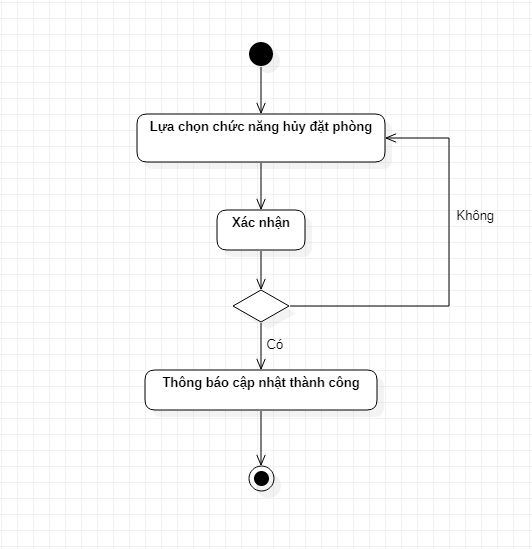
* + - 1. Chức năng hủy phòng.

Mô tả: Chức năng cho phép người dùng hủy đặt phòng trên hệ thống.

Quy trình thực thi: Sau khi đặt phòng người dùng cảm thấy chưa phù hợp hay muốn đặt lại thì có thể sử dụng chức năng này để hủy đơn đã đặt trước và tiến hành đặt theo chức năng đặt phòng.



Hình 2-8: Biểu đồ tuần tự chức năng hủy đặt phòng.

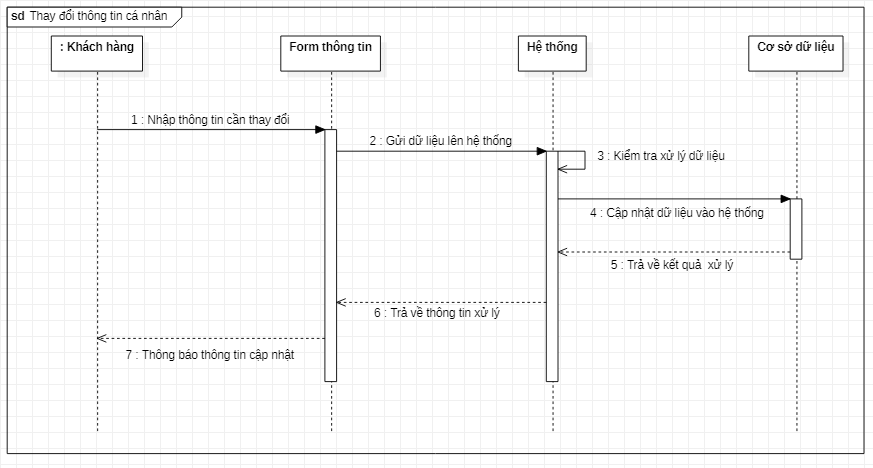


Hình 2-9: Biểu đồ chức năng hủy đặt phòng

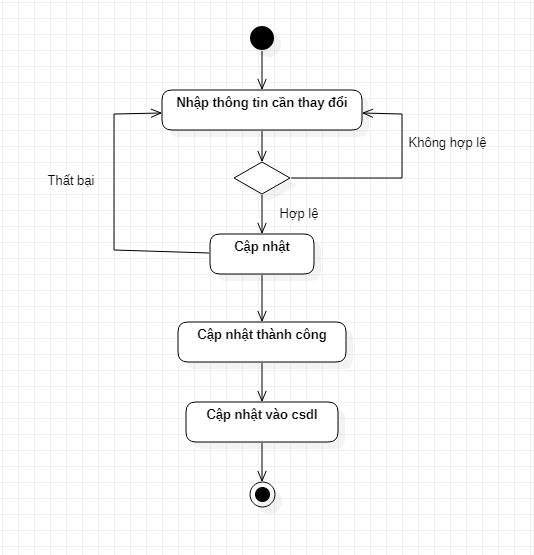
* + - 1. Chức năng thay đổi thông tin cá nhân.

Mô tả: Chức năng cho phép người dùng thay đổi thông tin cá nhân.

Quy trình thực thi: Sau khi đăng ký tài khoản, người dùng có thể cập nhật thông tin cá nhân còn thiếu trong phần hồ sơ cá nhân.



Hình 2-10: Biểu đồ tuần tự chức năng thay đổi thông tin.

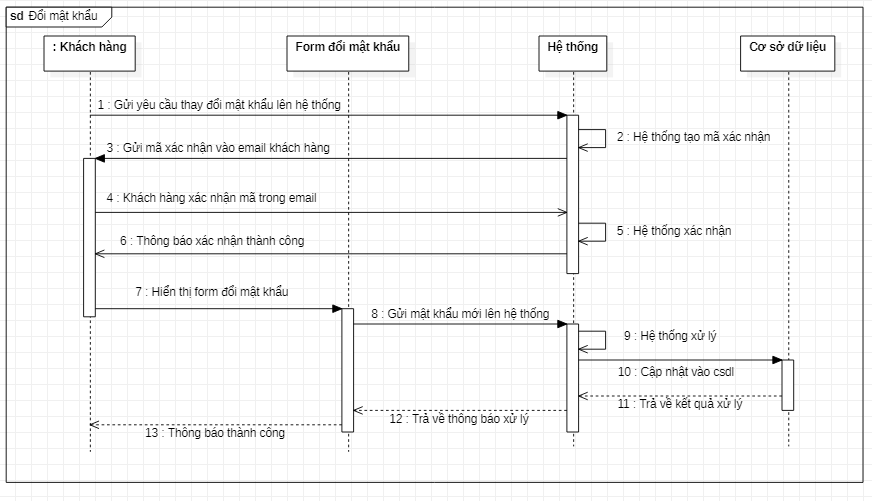


Hình 2-11: Biểu đồ hoạt động thay đổi thông tin.

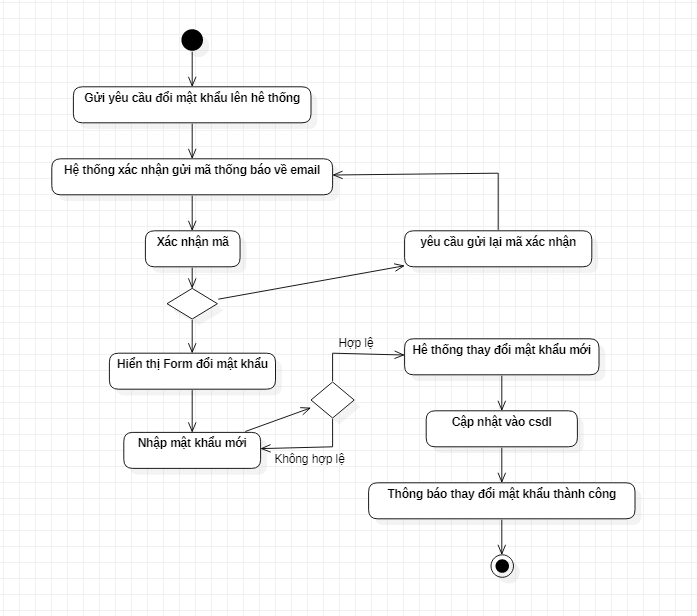
* + - 1. Chức năng đổi mật khẩu.

Mô tả: Chức năng cho phép người dùng thay đổi mật khẩu.

Quy trình thực thi: Người dùng chọn nút đổi mật khẩu sau đó chọn gửi email để nhận mã thay đổi mật khẩu. Sau khi nhận được mã này người dùng xác nhận và được chuyển đến trang thay đổi mật khẩu và tiến hành thay đổi mật khẩu mới.



Hình 2-12: Biểu đồ tuần tự chức năng đổi mật khẩu.

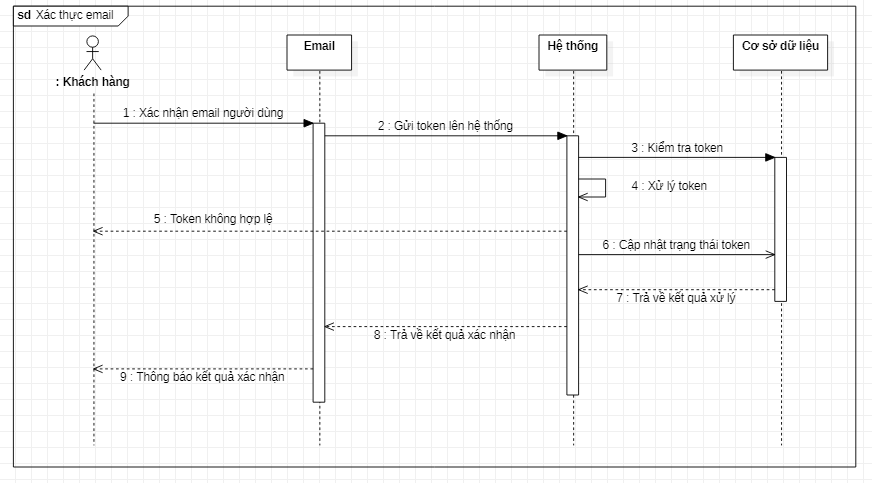


Hình 2-13: Biểu đồ hoạt động chức năng đổi mật khẩu.

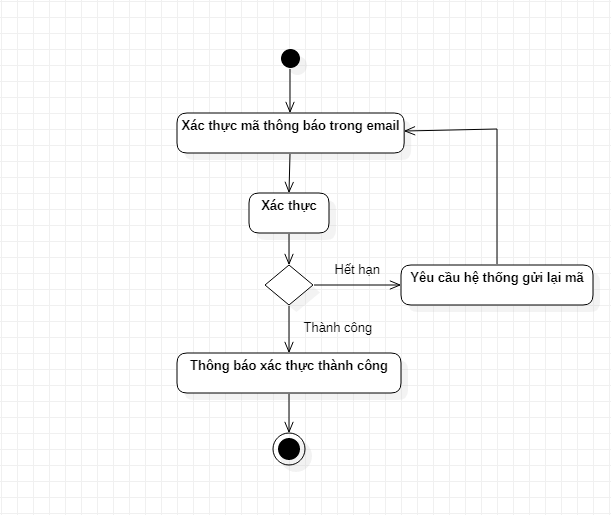
* + - 1. Chức năng xác thực email

Mô tả: Chức năng cho phép người dùng xác thực email sau khi đăng ký tài khoản.

Quy trình thực thi: Sau khi người dùng đăng ký thành công, hệ thống gửi mã xác nhận vào email. Người dùng sẽ đăng nhập vào email của họ và bấm vào nút xác nhận, sau khi xác nhận tài khoản sẽ được kích hoạt và hoạt động bình thường.



Hình 2-14: Biểu đồ tuần tự chức năng xác thực email



Hình 2-15: Biểu đồ hoạt động xác thực email

1. THIẾT KẾ CƠ SỞ DỮ LIỆU.
   1. Biểu đồ lớp.
      1. Biểu đồ lớp tổng quát của hệ thống:



Hình 3-1: Biểu đồ lớp tổng quát

* + 1. Danh sách các lớp đối tượng của hệ thống:

Bảng 3- 1: Bảng các lớp đối tượng

|  |  |  |
| --- | --- | --- |
| STT | Tên lớp | Mô tả |
| 1 | User | Lưu trữ thông tin người dùng |
| 2 | Role | Lưu trữ các vai trò của hê thống |
| 3 | Permission | Lưu trữ các vai trò cụ thể trong hệ thống |
| 4 | City | Lưu trữ thông tin các thành phố |
| 5 | Hotel | Lưu trữ thông tin khách sạn |
| 6 | Room | Lưu trữ thông tin phòng |
| 7 | Room Type | Lưu trữ thông tin loại phòng |
| 8 | Booking Room | Lưu trữ thông tin đặt phòng của khách hàng |
| 9 | Post | Lưu trữ thông tin bài đăng |
| 10 | Register\_Log | Lưu trữ thông tin xác thực tài khoản |

* 1. Thiết kế các bảng dữ liệu.
     1. Bảng User.

Bảng 3- 2: Bảng dữ liệu User

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| UserID | Id người dùng | Int(11) |
| Email | Địa chỉ email | Varchar(50) |
| UserType | Loại người dùng | Varchar (10) |
| FullName | Họ và tên | Varchar (50) |
| Password | Mật khẩu | Varchar (50) |
| PhoneNumber | Số điện thoại | Varchar (20) |
| DateOfBirth | Ngày sinh | DateTime |
| Avatar | Ảnh | Varchar (255) |
| Gender | Giới tính | Varchar (10) |
| Status | Trạng thái | Varchar (10) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng City.

Bảng 3- 3: Bảng dữ liệu City

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| CityID | Id Thành phố | Int(11) |
| CityName | Tên thành phố | Varchar(50) |
| Description | Mô tả | Varchar(255) |
| Image | Ảnh | Varchar(255) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng Hotel.

Bảng 3- 4: Bảng dữ liệu Hotel

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| HotelID | Id khách sạn | Int(11) |
| HotelName | Tên khách sạn | Varchar(50) |
| Addrees | Địa chỉ | Varchar(255) |
| Image | Ảnh | Varchar(255) |
| Email | Địa chỉ email | Varchar(50) |
| PhoneNumber | Số điện thoại | Varchar(20) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng Room.

Bảng 3- 5: Bảng dữ liệu Room

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| RoomID | Id phòng | Int(11) |
| RoomNumber | Số phòng | Varchar(20) |
| Status | Trạng thái | Varchar(10) |
| Image | Ảnh | Varchar(255) |
| Contents | Nội dung | Text |
| NumberOfPeople | Số người | Int(11) |
| RoomCost | Giá phòng | Int(11) |
| Discount | Giá giảm giá | Int(11) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng RoomType.

Bảng 3- 6: Bảng dữ liệu RoomType

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| RoomTypeID | Id loại phòng | Int(11) |
| RoomTypeName | Tên loại phòng | Varchar(50) |
| Description | Mô tả | Varchar(255) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng BookingRoom.

Bảng 3- 7: Bảng dữ liệu BookingRoom

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| BookingRoomID | Id đơn đặt phòng | Int(11) |
| CheckInDate | Ngày đặt phòng | DateTime |
| CheckOutDate | Ngày trả phòng | DateTime |
| TotalNumberOfPeople | Tổng số người | Int(11) |
| Status | Trạng thái | Varchar(10) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng Role.

Bảng 3- 8: Bảng dữ liệu Role

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| RoleID | Id vài trò | Int(11) |
| RoleName | Tên vài trò | Varchar(50) |
| Status | Trạng thái | Varchar(10) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng Permission.

Bảng 3- 9: Bảng dữ liệu Permission

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| PermissionID | Id quyền hạn | Int(11) |
| PermissionName | Tên quyền hạn | Varchar(50) |
| PermissionKey | Khóa quyền hạn | Varchar(10) |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng Post.

Bảng 3- 10: Bảng dữ liệu Post

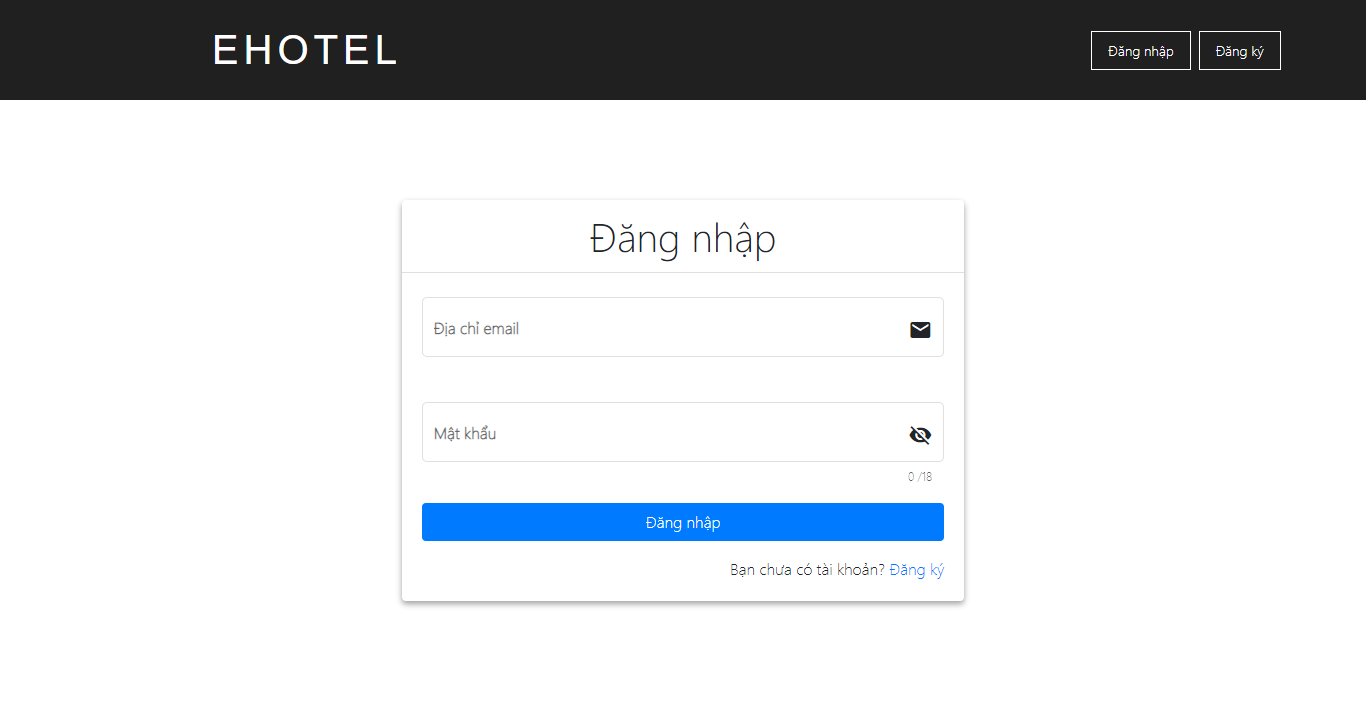
|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| PostID | Id bài đăng | Int(11) |
| Title | Tiêu đề | Varchar(255) |
| Contents | Nội dung | Text |
| Image | Ảnh | Varchar |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày chỉnh sửa | DateTime |

* + 1. Bảng Register\_Log.

Bảng 3- 11: Bảng dữ liệu Register\_Log

|  |  |  |
| --- | --- | --- |
| Tên cột | Mô tả | Kiểu dữ liệu |
| ID | Id | Int(11) |
| Token | Mã thông báo | Varchar(255) |
| Status | Trạng thái | Varchar(10) |
| DateActive | Thời gian sống | DateTime |
| DateCreated | Ngày tạo | DateTime |
| UpdatedDate | Ngày kích hoạt | DateTime |

1. TỔNG KẾT ĐÁNH GIÁ
   1. Kết quả xây dựng website.
      1. Giao diện đăng nhập.

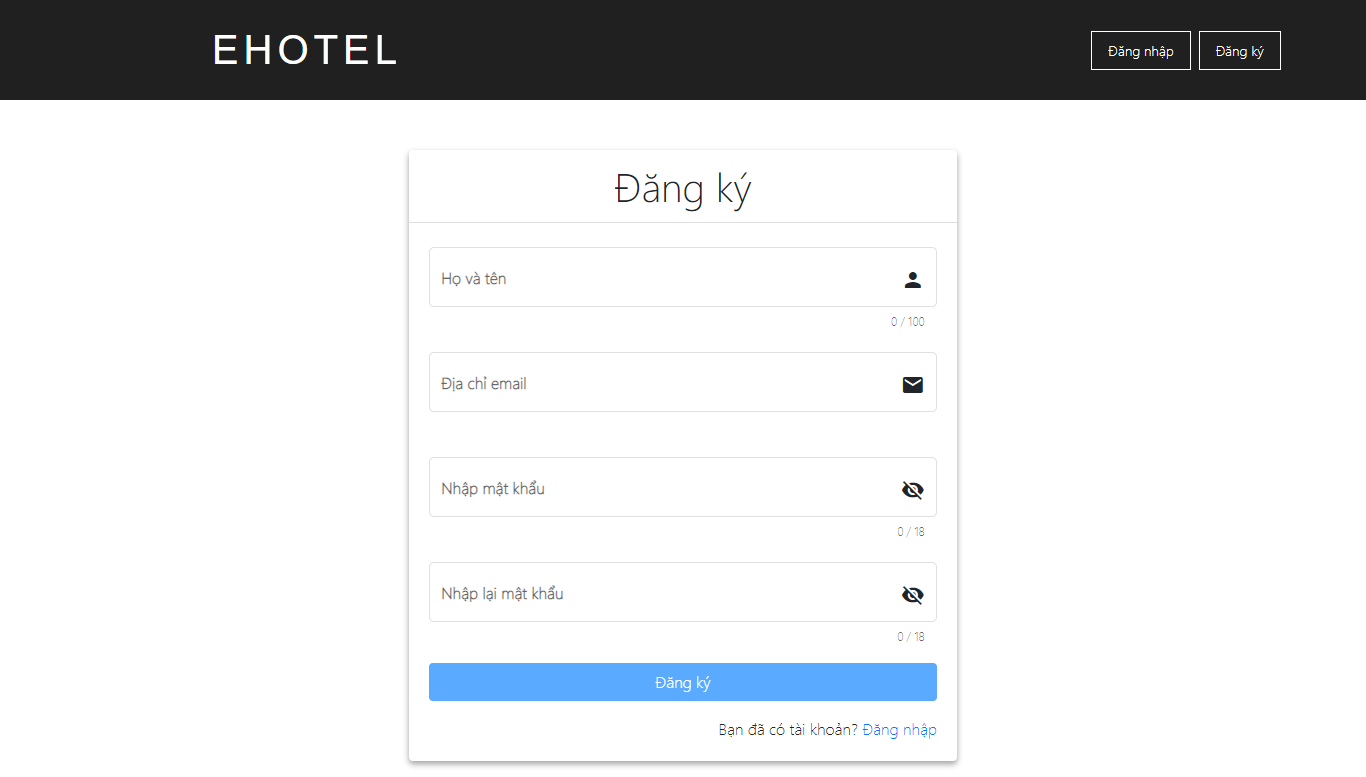


Hình 4-1: Giao diện đăng nhập

Giao diện đăng nhập gồm các ô nhập địa chỉ email, mật khẩu (có thể hiển thị mật khẩu bằng icon), nút đăng nhập và đường link đến trang đăng ký nếu chưa có tài khoản.

Cho phép người dùng đăng nhập vào hệ thống nếu đã có tài khoản trong hệ thống.

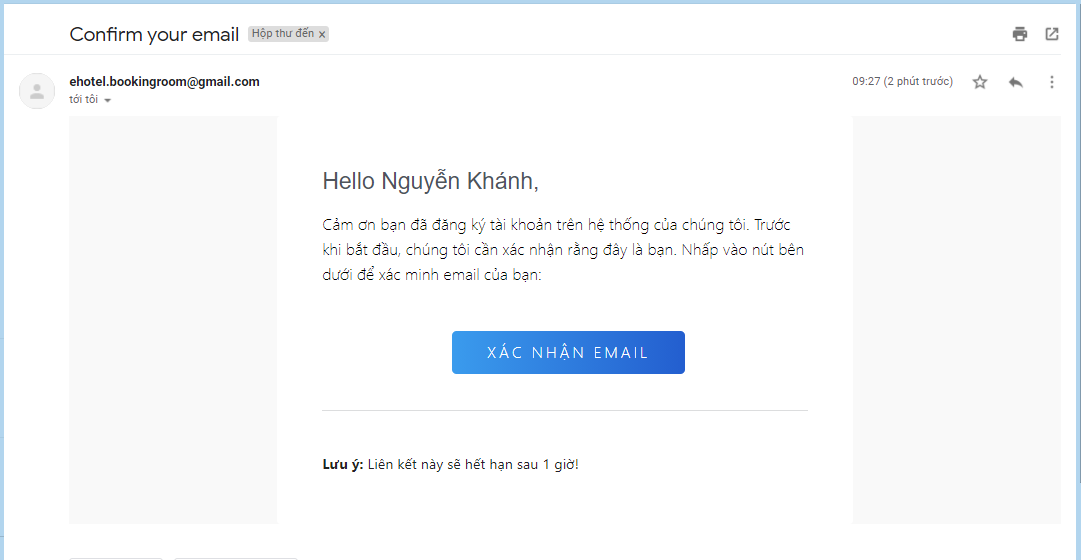
* + 1. Giao diện đăng ký.



Hình 4-2: Giao diện đăng ký

Giao diện gồm có các ô nhập thông tin đăng ký như: họ và tên, địa chỉ email, mật khẩu (có thể hiển thị mật khẩu bằng icon), nút đăng ký và một đường link đến trang đăng nhập.

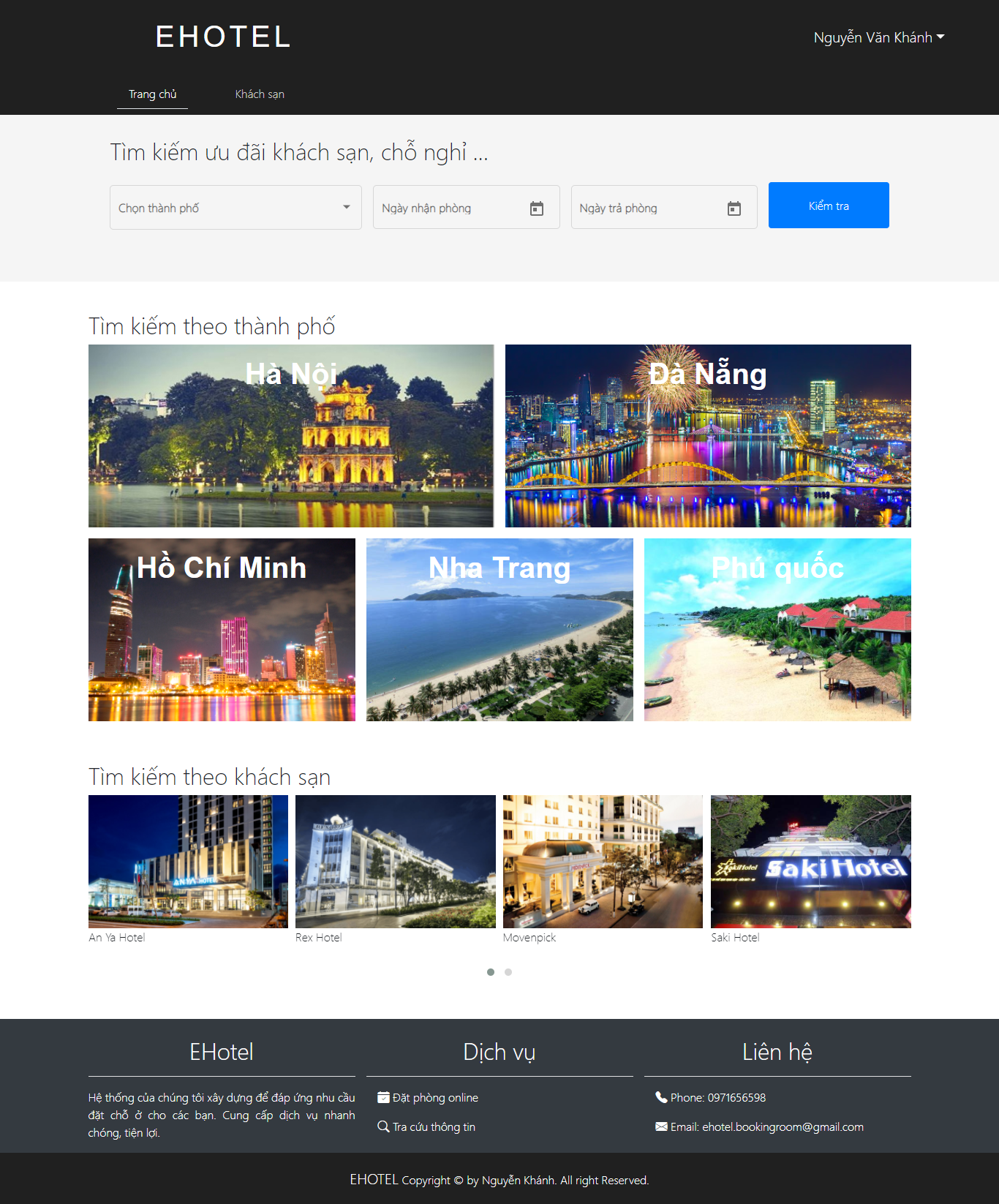
Cho phép người dùng tạo tài khoản mới vào hệ thống bằng cách nhập thông tin theo form đăng ký. Sau khi đăng ký người dùng sẽ nhận được mã xác thực gửi vào email vừa đăng ký. Người dùng cần xác thực email để kích hoạt tài khoản mới có thế đăng nhập vào hệ thống.



Hình 4-3: Giao diện xác thực email sau khi đăng ký

Cho phép người dùng bấm xác nhận, nếu link hợp lệ thì hệ thống thông báo xác nhận thành công, nếu mã xác nhận hết hạn hoặc không hợp lệ thì hệ thống sẽ cho phép người dùng gửi lại mã xác nhận.

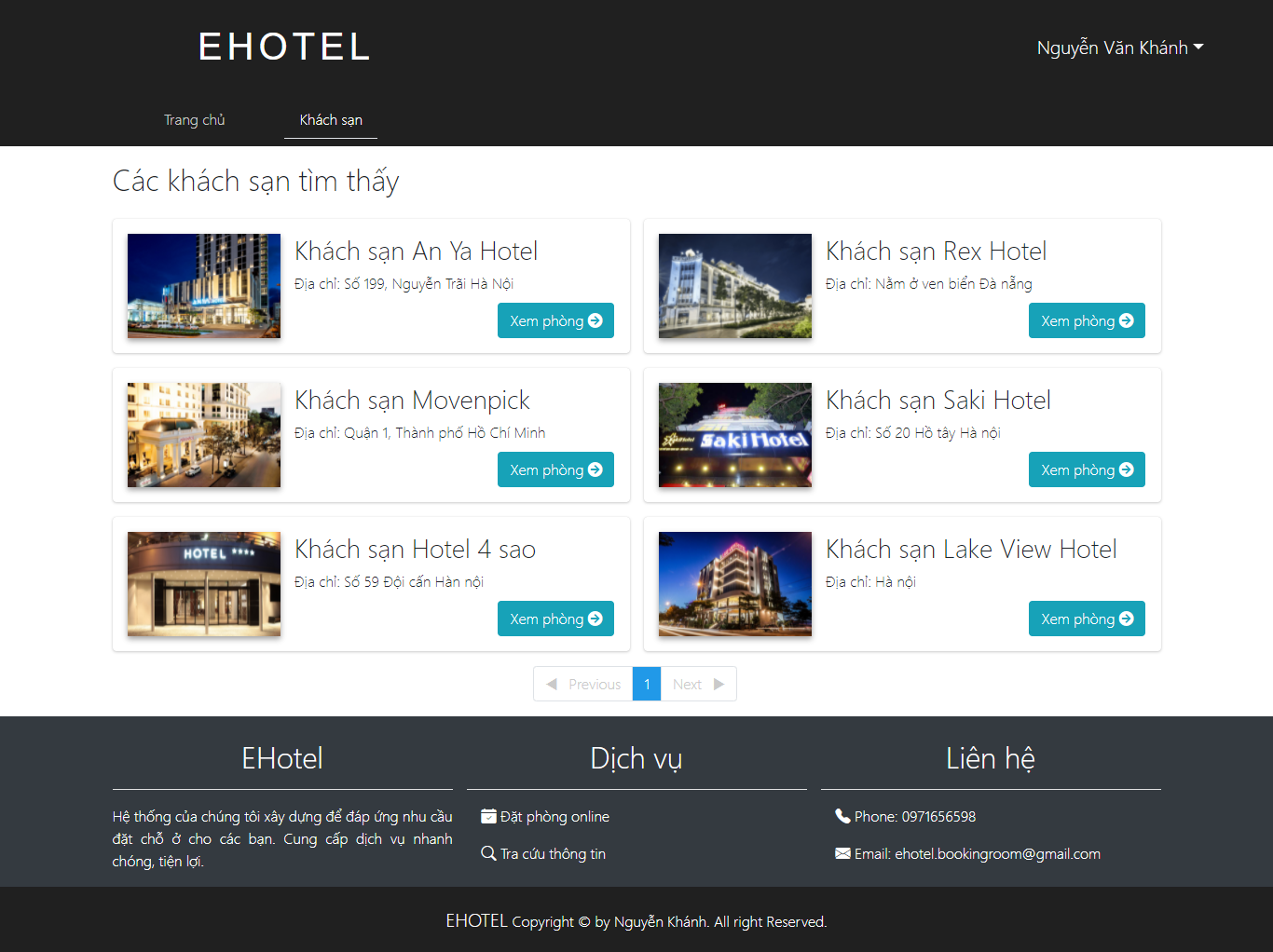
* + 1. Giao diện trang chủ.



Hình 4-4: Giao diện trang chủ

Giao diện trang chủ gồm có:

* Một thanh menu để điều hướng đến các trang.
* Thanh tìm kiếm theo cho phép tìm kiếm theo thành phố và thời gian.
* Một danh sách tìm kiếm theo thành phố, danh sách tìm kiếm theo khách sạn.
* Cuối trang hiển thị thông tin cơ bản của trang web: thông tin, dịch vụ, liên hệ.
  + 1. Giao diện danh sách khách sạn.

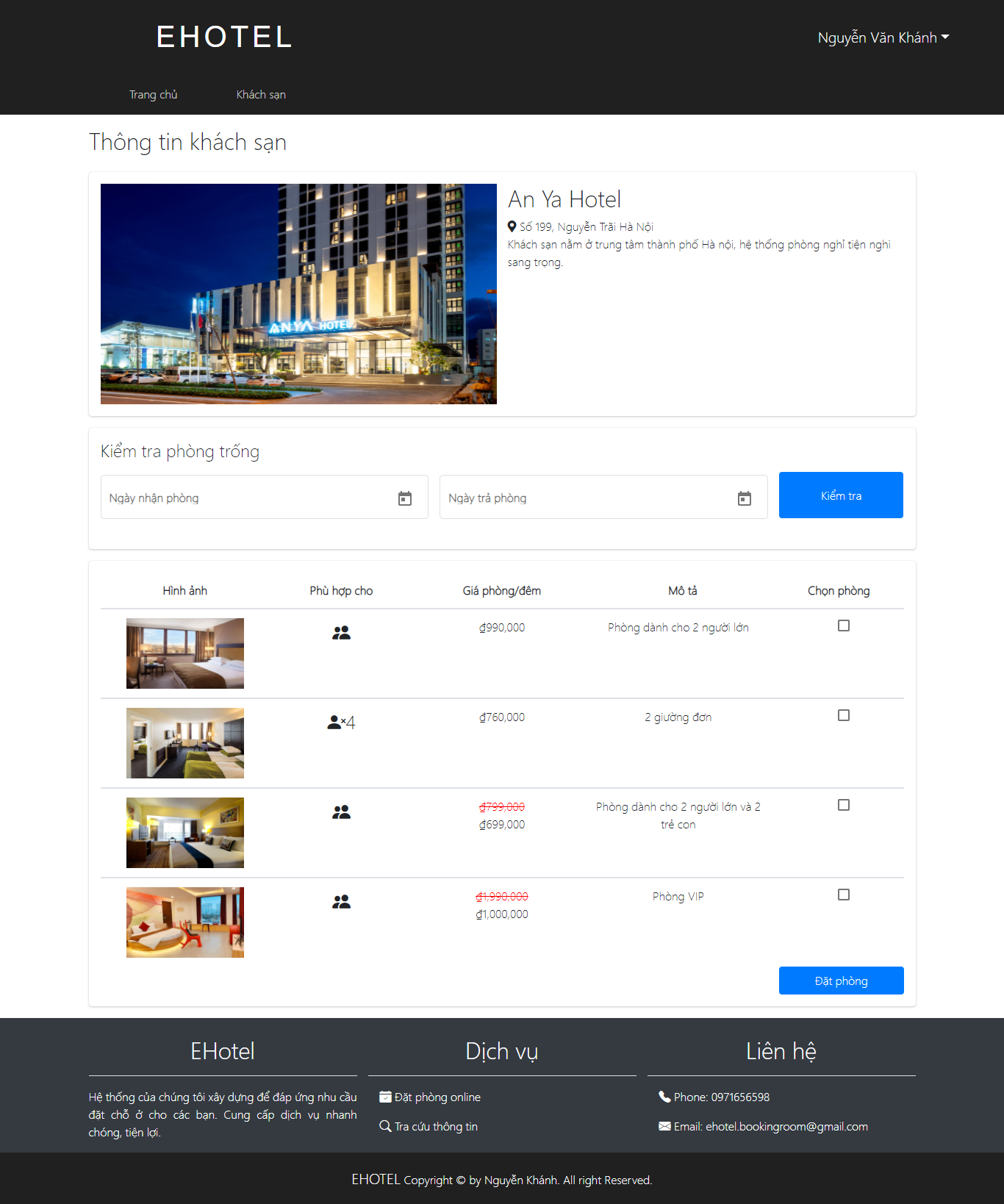


Hình 4-5: Giao diện danh sách khách sạn

Giao diện này hiển thị tất cả các khách sạn trong hệ thống.

Cho phép người dùng tra cứu, tìm phòng trong hệ thống.

* + 1. Giao diện chi tiết khách sạn và đặt phòng.



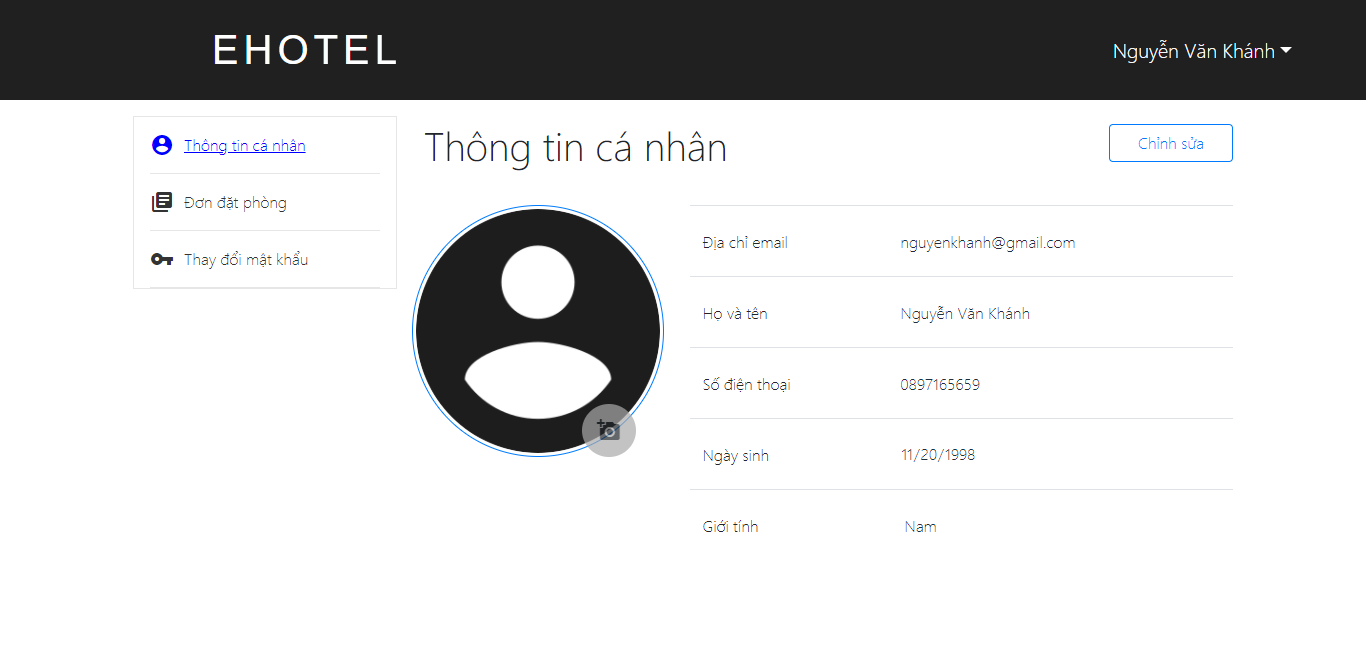
Hình 4-6: Giao diện chi tiết và đặt phòng

Giao diện gồm có:

* Thông tin chi tiết của khách sạn đó.
* Một thanh kiểm tra phòng trống trong khách sạn theo thời gian.
* Danh sách tất cả phòng trong khách sạn đó.

Cho phép người dùng đặt phòng sau khi đăng nhập, kiểm tra phòng trống.

* + 1. Giao diện thông tin người dùng.



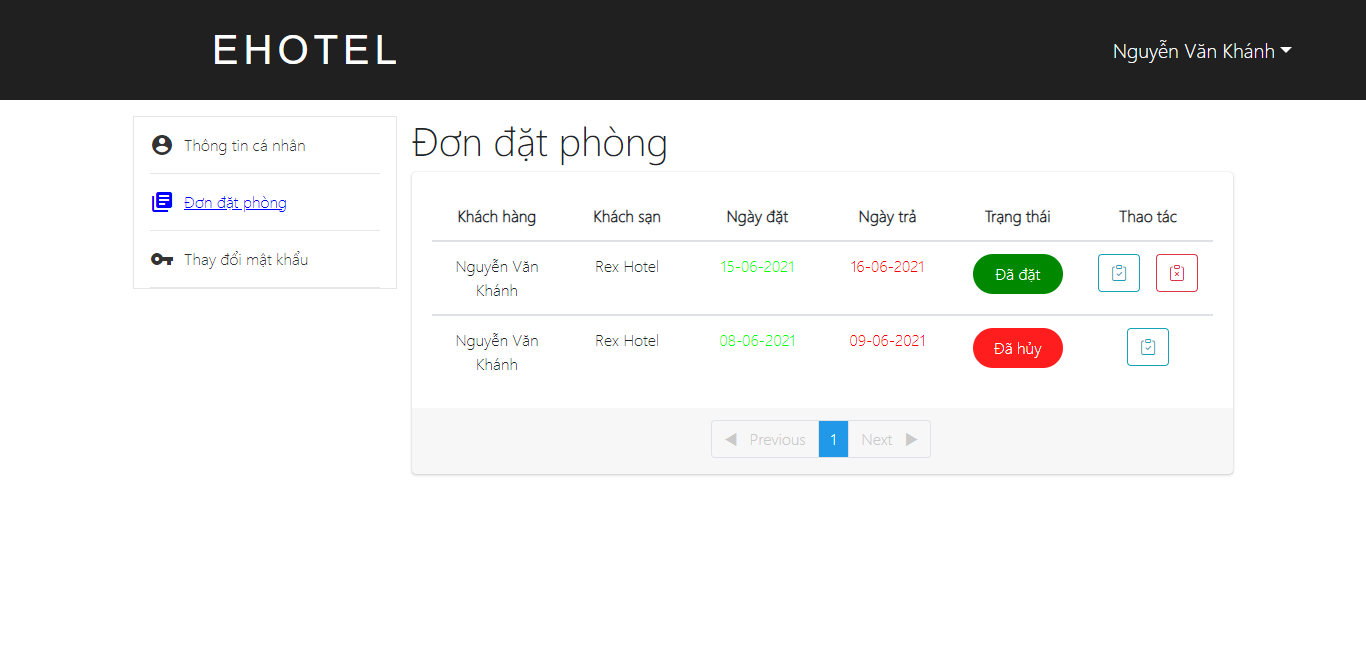
Hình 4-7: Giao diện thông tin người dùng

Giao diện gồm có:

* Một thanh menu điều hướng đến: thông tin cá nhân, đơn đặt phòng, thay đổi mật khẩu.
* Hình ảnh, thông tin cá nhân của người dùng

Cho phép người dùng:

* Thay đổi hình ảnh, thông tin cá nhân của người dùng.
  + 1. Giao diện danh sách đơn đặt phòng của khách hàng.



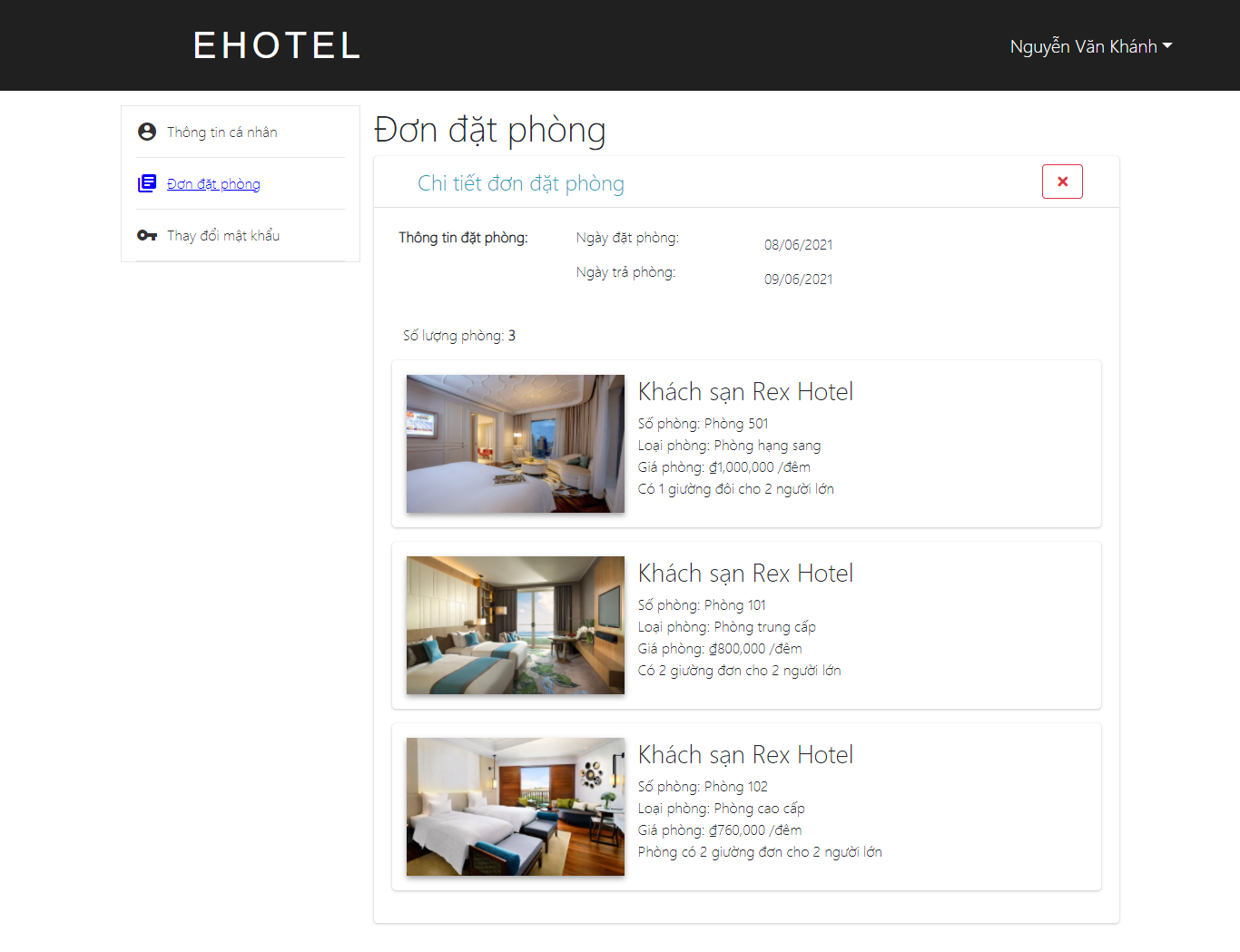
Hình 4-8: Giao diện danh sách đơn đặt phòng

Giao diện gồm có:

* Một thanh menu điều hướng đến: thông tin cá nhân, đơn đặt phòng, thay đổi mật khẩu.
* Danh sách tất cả các đơn đặt phòng của khách hàng.

Cho phép người dùng:

* Xem danh sách đặt phòng, hủy đặt phòng.
  + 1. Giao diện chi tiết đơn đặt phòng



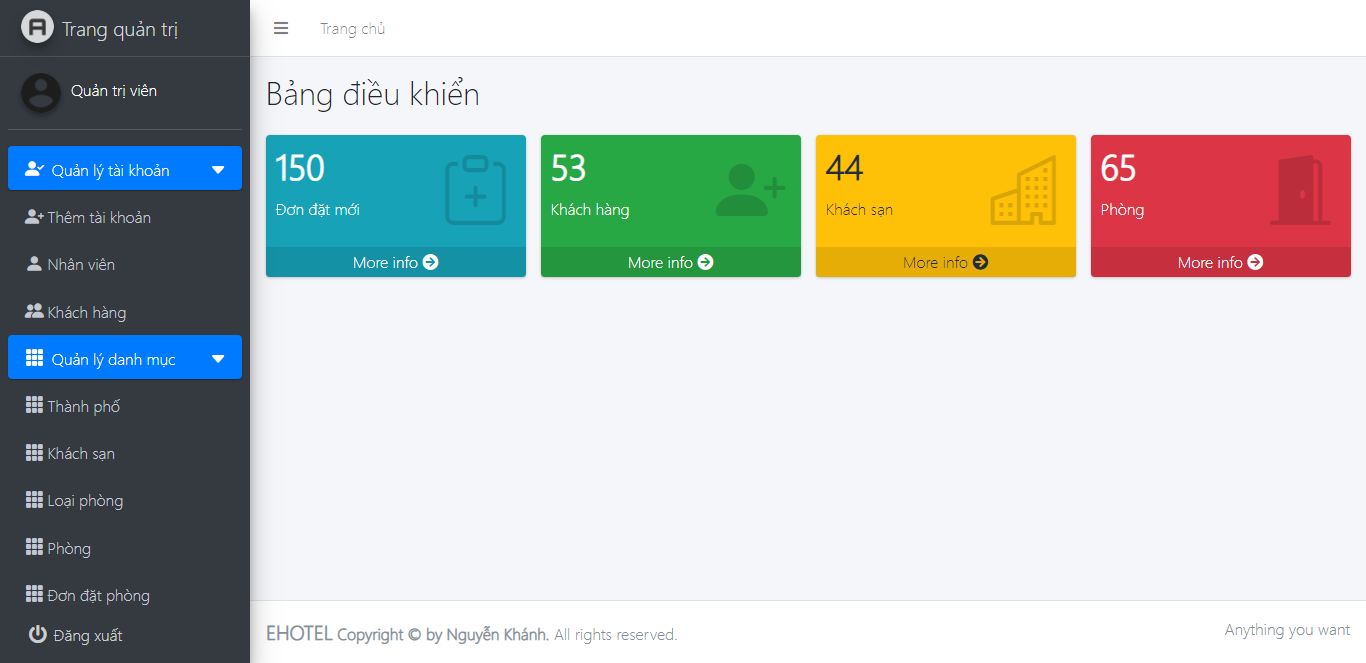
Hình 4-9: Giao diện chi tiết đơn đặt phòng

Giao diện gồm có:

* Một thanh menu điều hướng đến: thông tin cá nhân, đơn đặt phòng, thay đổi mật khẩu.
* Danh sách chi tiết các phòng trong đơn đặt phòng.

Cho phép người dùng:

* Xem thông tin chi tiết đơn đặt phòng.
  + 1. Giao diện trang quản trị



Hình 4-10: Giao diện trang quản trị

Giao diện gồm có:

* Một thanh menu điều hướng đến các chức năng.
* Các mục thống kê số lượng đơn đặt mới, khách hàng, khách sạn, phòng.

Cho phép quản trị viên:

* Quản lý nhân viên, khách hàng.
* Quản lý thành phố, khách sạn, phòng.
* Quản lý đơn đặt phòng của khách hàng.
  1. Đánh giá kết quả

Trong quá trình nhận và xây dựng đề tài này, em đã đạt được một số mục tiêu đề ra như sau:

* Nắm bắt được kiến thức về Spring Boot, Angular và một số kiến thức khác.
* Phân tích và xây dựng mô hình cơ sở dữ liệu với hệ quản trị cơ sở dữ liệu PostgreSql.
* Hiểu cơ bản về phân tích thiết kế hệ thống thông tin.
* Áp dụng kiến thức tìm hiểu được vào xây dựng và phát triển ứng dụng đặt phòng trên môi trường web.

Kết quả thực nghiệm thực hiện đã hoàn thành được 95% phần Front-end và các API phía Back-end thực chức năng thêm, sửa, xóa, đọc dữ liệu… do một số lý do khách quan.

Trong tương lai, em hi vọng sẽ cố gắng hoàn thiện tốt đề tài này và cố gắng đáp ứng được yêu cầu hệ thống thực đòi hỏi và sử dụng trong thực tế. Vì vậy kính mong các thầy giáo, cô giáo và các bạn xem xét, chỉ bảo và giúp đỡ em để em có thể hoàn thành đề tài này tốt hơn.

TÀI LIỆU THAM KHẢO

[1] <https://spring.io/projects/spring-boot>

[2] <https://angular.io/docs>

[3] <https://material.angular.io/components/categories>

[4] <https://www.postgresql.org/about/>

[5] <https://www.tutorialspoint.com/angular_material/angular_material_overview.html>

[6] https://topdev.vn/blog/restful-api-la-gi/