**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH FONTEND**

**ĐỀ TÀI: PHÁT TRIỂN ỨNG DỤNG WEB BAO GỒM GIAO DIỆN NGƯỜI DÙNG VÀ GIAO DIỆN QUẢN TRỊ CHO PHÉP ĐẶT PHÒNG TRỰC TUYẾN CỦA MỘT KHÁCH SẠN**

**Giáo viên hướng dẫn: ThS : Đặng Khánh Trung**

**Sinh viên thực hiện:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Stt** | **Mã sv** | **Họ và tên** | **Lớp** |
| **1** | **1671020094** | **Nguyễn Văn Giang** | **CNTT 16-04** |

**Hà Nội, năm 2022**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC ĐẠI NAM**

Logo, company name

Description automatically generated

**BÀI TẬP LỚN**

**TÊN HỌC PHẦN: LẬP TRÌNH FONTEND**

**ĐỀ TÀI: PHÁT TRIỂN ỨNG DỤNG WEB BAO GỒM GIAO DIỆN NGƯỜI DÙNG VÀ GIAO DIỆN QUẢN TRỊ CHO PHÉP ĐẶT PHÒNG TRỰC TUYẾN CỦA MỘT KHÁCH SẠN**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| STT | Mã Sinh Viên | Họ và Tên | Ngày Sinh | Điểm | |
| Bằng Số | Bằng Chữ |
| 1 | 1671020094 | Nguyễn Văn Giang | 27/11/2004 |  |  |

CÁN BỘ CHẤM THI

**Hà Nội, năm 2022**

**LỜI NÓI ĐẦU**

Front End (còn được biết đến như client-side) là tất cả những gì liên quan đến điều mà người dùng nhìn thấy mỗi khi truy cập vào một trang web, bao gồm phạm trù thiết kế và các ngôn ngữ như HTML hay CSS.

Người dùng tương tác trực tiếp với nhiều khía cạnh thuộc Front End như: nhận biết vị trí đặt để của logo, màu sắc chủ đạo, tìm kiếm và đọc thông tin, sử dụng các button và tính năng trên web… Mục đích cuối cùng của Front End là nhằm mang lại một giao diện bắt mắt, giúp người dùng dễ dàng thao tác và sử dụng.

Để xây “mặt tiền” này, lập trình viên Front End (Front End developer) phải thành thạo 3 ngôn ngữ: **HTML, CSS** và ngôn ngữ lập trình [JavaScript](https://itviec.com/viec-lam-it/javascript?utm_source=blogpost&utm_medium=center_post&utm_campaign=linktosite). Ngoài ra, lập trình viên Front End cần biết sử dụng các thư viện, framework như jQuery, Bootstrap, AngularJS, ReactJS cũng như có kiến thức về Photoshop, UI/UX.

Lựa chọn trở thành lập trình Front End Developer không chỉ mang đến cho bạn nhiều cơ hội học hỏi và cọ sát, mà còn giúp bạn nhận được mức [lương IT](https://itviec.com/blog/bao-cao-luong-it/?utm_source=blog&utm_medium=internal-link&utm_campaign=salary-report-2022&utm_content=dec2022) thuộc top đầu, cũng như con đường thăng tiến nhanh chóng

**MỤC LỤC**

**MỤC LỤC HÌNH ẢNH**

**(Đánh tự động nếu có)**

**BẢNG CÁC TỪ VIẾT TẮT**

|  |  |  |
| --- | --- | --- |
| **STT** | **TỪ VIẾT TẮT** | **VIẾT ĐẦY ĐỦ** |
| **1** | **CSDL** | **Cơ sở dữ liệu** |
| **2** | **LTFD** | **Lập trình fontend** |
| **3** | **JVT** | **Javacipt** |
| **4** |  |  |
|  |  |  |

**CHƯƠNG 1. MỞ ĐẦU**

## **1.1 : Lý do chọn đề tài**

Việc chọn đề tài "Phát triển ứng dụng web bao gồm giao diện người dùng và giao diện quản trị cho phép đặt phòng trực tuyến của một khách sạn" có thể được đánh giá cao với nhiều lý do. Dưới đây là một số lợi ích và lý do mà bạn có thể đã chọn đề tài này:

Xu hướng đặt phòng trực tuyến ngày càng phổ biến trong ngành du lịch và khách sạn.

Do nhu cầu sử dụng dịch vụ đặt phòng trực tuyến tăng cao, việc phát triển một ứng dụng web có thể giúp đáp ứng nhanh chóng và thuận tiện cho người dùng.

Cung cấp một giao diện người dùng thân thiện, dễ sử dụng giúp khách hàng dễ dàng đặt phòng một cách thuận tiện và nhanh chóng.

Tính năng đặt phòng trực tuyến giúp tiết kiệm thời gian và công sức cho khách hàng.

Giao diện quản trị sẽ hỗ trợ nhân viên khách sạn quản lý thông tin đặt phòng, theo dõi tình trạng phòng trống, và quản lý thông tin khách hàng một cách hiệu quả.

Có thể tích hợp các công cụ báo cáo để theo dõi hiệu suất kinh doanh và xu hướng đặt phòng.

Tích hợp các tính năng tương tác như xem trước phòng, đánh giá và đề xuất phòng giúp tăng trải nghiệm người dùng.

Phát triển một ứng dụng web yêu cầu kiến thức và kỹ năng về phần mềm, thiết kế giao diện, và quản trị cơ sở dữ liệu.

Cung cấp cơ hội để thách thức và nâng cao kỹ năng lập trình và phát triển ứng dụng web.

Cho phép áp dụng những kiến thức đã học được trong quá trình học, từ việc thiết kế giao diện đến quản lý cơ sở dữ liệu và triển khai ứng dụng web.

Có thể xem xét ứng dụng những công nghệ và khái niệm mới để tạo ra một sản phẩm hiện đại và đáp ứng yêu cầu ngày càng cao của thị trường.

Những lợi ích này có thể giúp định hình và làm cho dự án của bạn trở thành một ứng dụng web hữu ích và linh hoạt trong việc đáp ứng nhu cầu người dùng và quản lý khách sạn.

## **1.2 : Phạm vi nghiên cứu**

Phạm vi nghiên cứu của đề tài "Phát triển ứng dụng web bao gồm giao diện người dùng và giao diện quản trị cho phép đặt phòng trực tuyến của một khách sạn" có thể được xác định thông qua các yếu tố sau đây:

1. Giao diện Người dùng (Frontend)

- Thiết kế giao diện người dùng thân thiện, dễ sử dụng và thích ứng với nhiều thiết bị khác nhau (responsive design).

- Xây dựng các chức năng đặt phòng, kiểm tra tình trạng phòng, xem thông tin chi tiết về phòng và các tính năng khác cho khách hàng.

- Tích hợp các tính năng tương tác như xem trước phòng, đánh giá và đề xuất phòng

2. Giao diện Quản trị (Backend):

- Phát triển giao diện quản trị để quản lý đặt phòng, thông tin khách hàng, tình trạng phòng, và các yếu tố quản lý khác.

- Xây dựng chức năng quản lý người dùng, quản lý thông tin phòng, và cập nhật dữ liệu một cách hiệu quả.

3.Đặt phòng trực tuyến:

- Phát triển tính năng đặt phòng trực tuyến, bao gồm lựa chọn ngày, kiểm tra tình trạng phòng, xác nhận đặt phòng và gửi xác nhận cho khách hàng.

- Xử lý thanh toán trực tuyến (nếu có) và tích hợp các cổng thanh toán an toàn.

4. Quản lý dữ liệu:

- Xây dựng cơ sở dữ liệu để lưu trữ thông tin về phòng, khách hàng, đặt phòng và các thông tin liên quan khác.

- Thực hiện các thao tác CRUD (Create, Read, Update, Delete) để quản lý dữ liệu một cách linh hoạt.

5.Bảo mật

- Nghiên cứu và triển khai các biện pháp bảo mật để đảm bảo an toàn cho dữ liệu người dùng và giao dịch thanh toán.

- Kiểm thử bảo mật để đối mặt với các tình huống xâm nhập có thể xảy ra

6. Hiệu suất và Tối ưu hóa

- Tối ưu hóa hiệu suất ứng dụng để đảm bảo thời gian đáp ứng nhanh chóng và trải nghiệm người dùng mượt mà.

- Xử lý các tình huống đặc biệt như đồng thời nhiều đặt phòng, cập nhật thông tin đồng thời, và nâng cao khả năng mở rộng của hệ thống.

7. Thử nghiệm và Đánh giá:

- Thực hiện các bước kiểm thử đầy đủ để đảm bảo tính ổn định và độ tin cậy của ứng dụng.

- Thu thập ý kiến phản hồi từ người sử dụng và đánh giá hiệu suất của hệ thống trong môi trường thực tế.

8.Mở rộng và Tích hợp

- Xem xét khả năng mở rộng ứng dụng để hỗ trợ nhiều khách sạn và hệ thống đặt phòng lớn hơn.

- Tích hợp các dịch vụ bên ngoài như đồng bộ hóa lịch, tích hợp đánh giá từ bên thứ ba, hoặc tích hợp với các hệ thống thanh toán.

Bằng cách xác định rõ ràng phạm vi nghiên cứu, bạn có thể tập trung vào việc phát triển các tính năng và chức năng cụ thể, giúp đảm bảo dự án được triển khai một cách hiệu quả và đáp ứng đúng nhu cầu của khách hàng và quản trị khách sạn.

## **1.3 : Kết cấu của bài báo cáo**

Chương 1: Giới thiệu đề tài

Chương 2: Cơ sở lý thuyết

Chương 3: Cài đặt chương trình

Tài liệu tham khảo

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

## **2.1 Tổng quan về ngôn ngữ Javacript**

JavaScript là một ngôn ngữ lập trình thông dịch, đa nền tảng và dựa trên sự kiện. Ban đầu, JavaScript được thiết kế để thực hiện trên trình duyệt web để cải thiện trải nghiệm người dùng khi tương tác với trang web. Ngày nay, JavaScript không chỉ được sử dụng trên trình duyệt mà còn mở rộng ra nhiều lĩnh vực khác như phát triển ứng dụng web toàn diện, phát triển ứng dụng di động, máy chủ và nhiều hơn nữa.

Dưới đây là một số điểm tổng quan về JavaScript:

1. Loại ngôn ngữ:

- JavaScript là một ngôn ngữ kịch bản (scripting language), chủ yếu được sử dụng để tương tác với các trang web và thay đổi nội dung của chúng mà không cần tải lại trang.

2.Đa nền tảng:

- JavaScript có thể chạy trên nhiều nền tảng khác nhau, bao gồm trình duyệt web phổ biến như Chrome, Firefox, Safari, và Edge

3. Thực thi dựa trên sự kiện (Event-driven):

- JavaScript thường được sử dụng để phản ứng đối với sự kiện người dùng trên trang web, chẳng hạn như nhấp chuột, nhập liệu từ bàn phím, và các sự kiện khác.

4. Ngôn ngữ không đồng bộ (Asynchronous)

- JavaScript hỗ trợ lập trình không đồng bộ thông qua các callback, Promises, và từ phiên bản ES6 trở đi, Async/Await. Điều này làm cho nó thích hợp cho việc xử lý các tác vụ như tải dữ liệu từ máy chủ mà không làm đóng băng giao diện người dùng.

5.Dùng cho phía người dùng (Client-side) và phía máy chủ (Server-side):

- Trong phát triển web, JavaScript thường được sử dụng phía người dùng để tạo ra các trang web tương tác. Ngoài ra, với sự xuất hiện của Node.js, JavaScript cũng có thể được sử dụng để phát triển phía máy chủ.

6. ECMAScript:

- JavaScript tuân thủ các chuẩn được đặt ra bởi ECMA International. ECMAScript là một phần quan trọng của ngôn ngữ và định nghĩa cách JavaScript nên hoạt động.

7. Thư viện và Framework:

- JavaScript có nhiều thư viện và framework mạnh mẽ như React, Angular, và Vue.js giúp phát triển ứng dụng web một cách hiệu quả và dễ bảo trì.

8. Hỗ trợ đa môi trường

- JavaScript không chỉ chạy trong môi trường trình duyệt, mà còn có thể thực thi trên nhiều môi trường khác nhau như máy tính cá nhân, điện thoại di động, máy chủ, thiết bị IoT, v.v

JavaScript đã trở thành một ngôn ngữ lập trình rất quan trọng và có ảnh hưởng lớn trong lĩnh vực phát triển web và nhiều lĩnh vực công nghiệp khác. Sự mở rộng và linh hoạt của nó giúp các nhà phát triển xây dựng nhanh chóng và duy trì dễ dàng các ứng dụng và dịch vụ trực tuyến**.**



Hình 1 : Tổng quan về JVT

### 2.1.1 : Jiquery

jQuery là một thư viện JavaScript nhẹ, linh hoạt và được thiết kế để tăng cường khả năng tương tác và xử lý sự kiện trong trình duyệt web. Dưới đây là một tổng quan về jQuery

1. Nguyên tắc hoạt động:

- jQuery giúp đơn giản hóa việc thao tác với HTML Document Object Model (DOM) và xử lý sự kiện trong trình duyệt.

- Sử dụng cú pháp ngắn gọn để chọn các phần tử HTML và thực hiện các thao tác trên chúng.

2. Chọn lọc phần tử (Element Selection):

- jQuery cung cấp các phương thức để dễ dàng chọn các phần tử HTML bằng cách sử dụng các biểu thức chọn CSS, ID, class, hoặc các thuộc tính khác

3.Thao tác với DOM:

- jQuery cung cấp nhiều phương thức để thêm, xoá, thay đổi nội dung, và thậm chí là thay đổi cấu trúc của DOM một cách thuận tiện.

4. Xử lý sự kiện (Event Handling):

- Đơn giản hóa việc xử lý sự kiện như click, hover, change bằng cách sử dụng các phương thức dễ đọc và triển khai.

5. Hiệu ứng và hoạt hình (Effects and Animations):

- jQuery có thư viện hiệu ứng và hoạt hình giúp tạo ra các hiệu ứng đẹp mắt như slide, fade, và animate mà không cần phải viết mã JavaScript phức tạp.

6. Ajax (Asynchronous JavaScript and XML):

- Hỗ trợ Ajax để thực hiện các truy vấn HTTP mà không cần tải lại trang web, giúp tạo ra các trang web động và tương tác.

7.Plugins:

- Có sẵn nhiều plugin mở rộng chức năng của jQuery, giúp mở rộng khả năng của thư viện và giảm bớt công việc lập trình.

8. Hỗ trợ đa trình duyệt:

- jQuery giúp đảm bảo tính tương thích đa trình duyệt, giúp các lập trình viên tránh được các vấn đề liên quan đến sự khác biệt giữa các trình duyệt web.

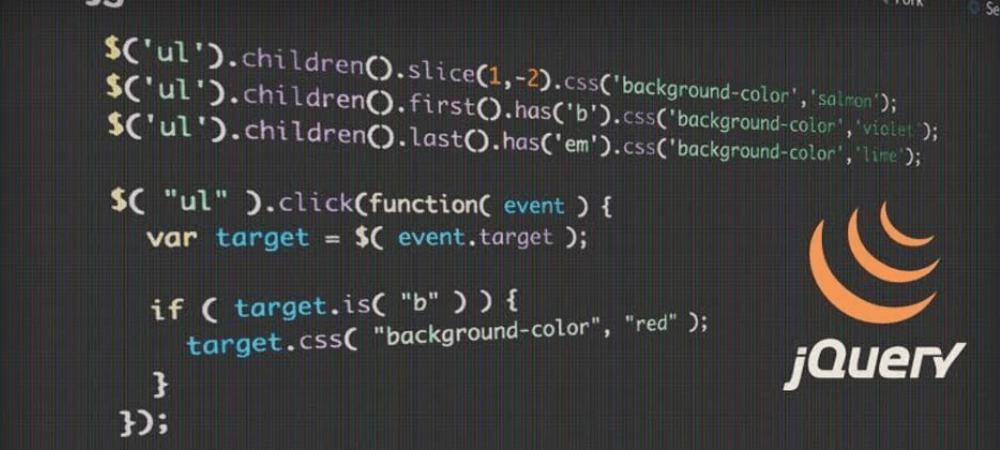
9. Nhỏ gọn và tốc độ:

- jQuery được thiết kế để nhẹ và tối ưu hóa tốc độ thực thi, giúp giảm bớt gánh nặng cho trình duyệt.

10. Dễ học và triển khai:

- Cú pháp đơn giản của jQuery giúp người mới học lập trình web có thể nhanh chóng làm quen và triển khai.

Mặc dù jQuery đã trở thành một công cụ quan trọng trong lịch sử phát triển web, nhưng do sự xuất hiện của các framework JavaScript hiện đại như React, Angular, và Vue.js, sự phổ biến của jQuery đã giảm đi một chút. Tuy nhiên, jQuery vẫn được sử dụng trong nhiều dự án và vẫn là một lựa chọn tốt cho những dự án đơn giản hoặc muốn tích hợp nhanh chóng các chức năng tương tác trên trang web.



Hình 2 : Hình ảnh Jiquery

### 2.1.2 : Bootstrap

Bootstrap là một framework front-end mã nguồn mở phổ biến được sử dụng để phát triển trang web và ứng dụng di động. Nó được tạo ra bởi Twitter và đã trở thành một công cụ quan trọng trong cộng đồng phát triển web. Dưới đây là một tổng quan về Bootstrap:

1.Responsive Design:

- Bootstrap cung cấp lưới (grid system) responsive, giúp dễ dàng tạo ra giao diện phản ứng với các kích thước màn hình khác nhau.

2. CSS và JavaScript:

- Bootstrap bao gồm một bộ CSS và JavaScript đã được thiết kế sẵn để giúp trang web có giao diện hiện đại và tương tác.

3. Components:

- Bootstrap cung cấp nhiều thành phần giao diện sẵn có như nút, biểu đồ, thanh định hình, hộp thoại, thanh điều hướng, biểu mẫu và nhiều hơn nữa. Các thành phần này giúp giảm thiểu công việc thiết kế và phát triển.

4. Typographic Elements:

- Bootstrap đi kèm với một loạt các phần tử chữ (typography) đã được thiết kế trước, giúp tạo ra trang web có định dạng chữ đẹp mắt và dễ đọc.

5. Theming và Tùy chỉnh:

- Bootstrap hỗ trợ tùy chỉnh thông qua việc sử dụng biến CSS và một hệ thống grid linh hoạt. Bạn cũng có thể tạo ra chủ đề riêng của mình để phù hợp với thiết kế của trang web.

6. Hỗ trợ đa trình duyệt:

- Bootstrap được kiểm thử và tối ưu hóa để hoạt động trên nhiều trình duyệt khác nhau, giúp đảm bảo tính tương thích.

7. Community và Documentation:

- Bootstrap có một cộng đồng lớn và tích cực, cung cấp nhiều tài liệu và ví dụ để giúp những người mới sử dụng framework.

8. Kích thước nhẹ:

- Mặc dù có nhiều tính năng, Bootstrap vẫn giữ kích thước nhẹ, giúp tối ưu hóa tốc độ tải trang.

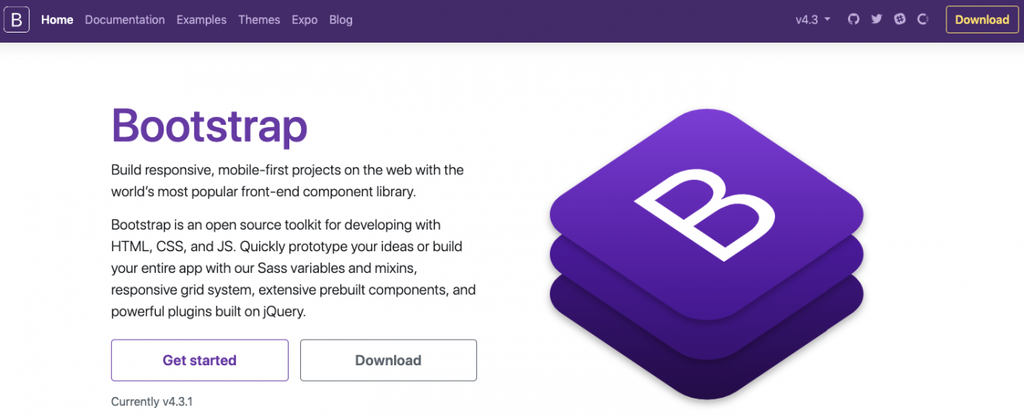
9. Tương thích với các dự án lớn và nhỏ:

- Bootstrap có thể được sử dụng trong nhiều loại dự án, từ trang web cá nhân đến ứng dụng doanh nghiệp lớn.

10. Phiên bản và Cập nhật:

- Bootstrap thường xuyên được cập nhật để cung cấp các tính năng mới và vá lỗi, đồng thời duy trì sự ổn định và bảo mật cho ứng dụng.

Bootstrap giúp giảm thiểu thời gian và công sức cần thiết để xây dựng giao diện người dùng cho trang web và ứng dụng, đặc biệt là trong các dự án cần sự linh hoạt và tốc độ phát triển cao.



Hình 3 : Bootstrap

### 2.1.3 : CSS

CSS, hay viết tắt của Cascading Style Sheets, là một ngôn ngữ lập trình sử dụng để mô tả cách mà các phần tử HTML sẽ được hiển thị trên màn hình, giấy in, hoặc các phương tiện trình bày khác. Dưới đây là một tổng quan về CSS:

1. Mục đích của CSS

- CSS được sử dụng để kiểm soát định dạng và trình bày của trang web hoặc ứng dụng web, giúp tách biệt lớp nội dung HTML và kiểu dáng.

2. Cú pháp:

- CSS sử dụng cú pháp đơn giản với các quy tắc gọi là "rules" (quy tắc), mỗi quy tắc bao gồm một "selector" (bộ chọn) và một "declaration block" (khối khai báo)

```css

selector {

property: value;

}

```

3.Selectors (Bộ chọn):

- Bộ chọn là các phần tử HTML mà bạn muốn áp dụng các quy tắc CSS. Có nhiều loại bộ chọn, bao gồm tên thẻ, class, ID, và nhiều bộ chọn khác.

```css

/\* Selector by tag name \*/

p {

color: blue;

}

/\* Selector by class \*/

.my-class {

font-size: 16px;

}

/\* Selector by ID \*/

#my-id {

background-color: #f0f0f0;

}

``

4. Properties (Thuộc tính):

- Thuộc tính là các đặc tính mà bạn muốn áp dụng cho bộ chọn. Ví dụ bao gồm màu chữ, kích thước phông chữ, margin, padding, và nhiều thuộc tính khác.

```css

selector {

property: value;

}

```

5. Values (Giá trị):

- Giá trị là giá trị cụ thể bạn muốn đặt cho một thuộc tính. Ví dụ, nếu bạn đặt màu chữ, giá trị có thể là tên màu (ví dụ: "red") hoặc mã màu hex (ví dụ: "#ff0000").

```css

p {

color: red;

}

```

6. Box Model:

- Box model mô tả cách mỗi phần tử HTML được hiển thị dưới dạng hộp, bao gồm margin, border, padding, và content.

7. Responsive Design:

- CSS được sử dụng để tạo ra giao diện phản ứng (responsive) bằng cách sử dụng các kỹ thuật như media queries để điều chỉnh kiểu dáng dựa trên kích thước màn hình.

8. CSS Preprocessors:

- Có nhiều CSS preprocessors như Sass và Less, giúp tăng cường khả năng quản lý mã nguồn CSS bằng cách cung cấp các tính năng như biến, nesting, và mixins.

9. CSS Frameworks

- Có các framework CSS như Bootstrap, Foundation, giúp tạo ra giao diện đẹp mắt mà không cần viết mã CSS từ đầu.

10. Cross-browser Compatibility:

- CSS cung cấp các thuộc tính và phương thức giúp đảm bảo tính tương thích với nhiều trình duyệt khác nhau.

CSS là một phần quan trọng của phát triển web và chịu trách nhiệm cho giao diện người dùng và trải nghiệm người dùng trên trang web. Nó cho phép nhà phát triển tạo ra giao diện linh hoạt và đẹp mắt thông qua các quy tắc kiểu dáng.



Hình 4 : CSS

## **2.2 Tổng quan về lập trình fontend**

Lập trình frontend là quá trình phát triển phần giao diện người dùng của một ứng dụng web hoặc trang web. Nhiệm vụ chính của lập trình frontend là xây dựng, tối ưu hóa và tương tác với giao diện người dùng, nhằm mang lại trải nghiệm tốt nhất cho người dùng cuối. Dưới đây là một tổng quan về lập trình frontend:

1. Ngôn ngữ lập trình

- HTML (HyperText Markup Language): Sử dụng để xây dựng cấu trúc của trang web bằng cách định rõ các phần tử và cấu trúc của chúng.

- CSS (Cascading Style Sheets): Định rõ kiểu dáng, màu sắc, font chữ và layout cho các phần tử HTML, giúp trang web trở nên hấp dẫn hơn.

- JavaScript: Thêm tính năng tương tác động, xử lý sự kiện, và thay đổi nội dung của trang web mà không cần phải tải lại trang.

2. Framework và Thư viện

- Framework: React, Angular, Vue.js là các framework phổ biến giúp tổ chức code, quản lý trạng thái ứng dụng, và xây dựng ứng dụng frontend mạnh mẽ.

- Thư viện: jQuery, Bootstrap, và lodash là những thư viện giúp đơn giản hóa việc phát triển và tối ưu hóa mã nguồn

3. Responsive Design:

- Tính chất responsive là khả năng của trang web tự điều chỉnh để phù hợp với kích thước màn hình khác nhau. CSS và media queries thường được sử dụng để đạt được tính chất này.

4. Single Page Applications (SPAs):

- Các ứng dụng frontend ngày nay thường được xây dựng dưới dạng Single Page Applications, nơi mà mọi tương tác xảy ra trên một trang duy nhất mà không cần phải tải lại trang.

5.Performance Optimization:

- Tối ưu hóa hiệu suất của trang web là một phần quan trọng của lập trình frontend. Điều này bao gồm tải các tài nguyên cần thiết, tối ưu hóa hình ảnh, sử dụng caching và minification để giảm dung lượng trang.

6. Cross-Browser Compatibility:

- Đảm bảo rằng trang web hoạt động một cách nhất quán trên nhiều trình duyệt khác nhau là một thách thức quan trọng trong lập trình frontend. CSS prefixing và kiểm thử đa trình duyệt là những cách để đối mặt với vấn đề này.

7. Version Control:

- Sử dụng hệ thống quản lý phiên bản như Git để theo dõi và quản lý mã nguồn, giúp phục hồi dễ dàng và làm việc hiệu quả trong nhóm.

8. Build Tools và Task Runners:

- Sử dụng các công cụ như Webpack, Babel, và Gulp để tự động hóa quá trình build, minify, và transpile mã nguồn.

9.Testing:

- Việc thực hiện kiểm thử đơn vị (unit testing) và kiểm thử tích hợp (integration testing) là quan trọng để đảm bảo tính ổn định và chất lượng của ứng dụng.

10. Security:

- Bảo vệ ứng dụng khỏi các lỗ hổng bảo mật là một yếu tố quan trọng. An toàn chứng chỉ HTTPS, xác thực người dùng, và quản lý cẩn thận dữ liệu đầu vào là các phần quan trọng của lập trình frontend.

Lập trình frontend đóng vai trò quan trọng trong việc tạo ra trải nghiệm người dùng cuối cùng và đóng góp lớn vào thành công của một ứng dụng web hoặc trang web.

### 2.2.1 : fontend

Nếu bạn đang nói về "frontend" trong ngữ cảnh phát triển web, frontend thường được hiểu là phần giao diện người dùng của một trang web hoặc ứng dụng, nơi mà người dùng tương tác trực tiếp. Frontend bao gồm HTML, CSS và JavaScript, cùng với các framework và thư viện để giúp việc phát triển trở nên dễ dàng và hiệu quả hơn.

Dưới đây là một cái nhìn tổng quan về các yếu tố quan trọng trong lập trình frontend:

1. HTML (HyperText Markup Language):

- Định nghĩa cấu trúc của trang web, xác định các phần tử và chúng được sắp xếp như thế nào trên trang

2. CSS (Cascading Style Sheets):

- Định dạng và kiểu dáng cho các phần tử HTML, quyết định màu sắc, font chữ, độ rộng, và vị trí của chúng trên trang.

3. JavaScript:

- Là ngôn ngữ lập trình dùng để thêm tính năng tương tác và động cho trang web. Nó có thể được sử dụng để thực hiện các tác vụ như xử lý sự kiện, thay đổi nội dung trang mà không cần tải lại, và giao tiếp với máy chủ.

### 2.2.2 : Backend

Backend là phần của một hệ thống thông tin chịu trách nhiệm xử lý dữ liệu và thực hiện các chức năng logic của ứng dụng. Nó là nơi xử lý các yêu cầu từ phía người dùng (frontend), tương tác với cơ sở dữ liệu, và thực hiện các nhiệm vụ khác để hỗ trợ hoạt động của toàn bộ hệ thống. Dưới đây là một tổng quan về backend:

1. Ngôn ngữ lập trình:

- Backend có thể được phát triển bằng nhiều ngôn ngữ lập trình khác nhau. Một số ngôn ngữ phổ biến bao gồm Java, Python, Ruby, PHP, Node.js (JavaScript), và C#.

2. Framework:

- Sử dụng framework là phổ biến trong lập trình backend để giúp tổ chức mã nguồn, tăng hiệu suất, và giảm thời gian phát triển. Ví dụ, Spring (Java), Django (Python), Ruby on Rails (Ruby), Laravel (PHP), và Express.js (Node.js).

3. Server:

- Backend thường chạy trên các máy chủ (servers) chuyên biệt hoặc các dịch vụ đám mây như AWS, Azure, hoặc Google Cloud Platform.

4. \*\*Databases:\*\*

- Làm việc với cơ sở dữ liệu là một phần quan trọng của backend. Các hệ quản lý cơ sở dữ liệu phổ biến bao gồm MySQL, PostgreSQL, MongoDB (NoSQL), và Microsoft SQL Server.

5. APIs (Application Programming Interfaces):

- Backend thường cung cấp các API để cho phép frontend và các ứng dụng khác tương tác với dữ liệu và chức năng của nó. RESTful API và GraphQL là hai chuẩn phổ biến cho việc xây dựng các API.

6.Authentication và Authorization:

- Xác thực (Authentication) là quá trình xác định xem người dùng có quyền truy cập hay không. Ở phía sau, ủy quyền (Authorization) quyết định những điều gì mà người dùng được phép thực hiện.

7. Security:

- Backend phải đảm bảo an toàn cho dữ liệu và hệ thống. Điều này bao gồm bảo vệ khỏi các cuộc tấn công như SQL injection, cross-site scripting (XSS), và cross-site request forgery (CSRF).

8. Logging và Monitoring:

- Ghi log và giám sát là quan trọng để theo dõi hiệu suất của hệ thống, xác định lỗi và đảm bảo rằng mọi thứ đang hoạt động đúng cách.

9. Performance Optimization:

- Backend cần được tối ưu hóa để đảm bảo hiệu suất cao. Điều này bao gồm việc sử dụng caching, tối ưu hóa truy vấn cơ sở dữ liệu, và sử dụng các kỹ thuật khác để giảm thời gian phản hồi.

10. Scalability:

- Backend cần có khả năng mở rộng để xử lý một lượng người dùng ngày càng tăng. Các phương tiện như load balancing, horizontal scaling, và sử dụng các dịch vụ đám mây có thể giúp đạt được điều này.

Backend và frontend cùng hợp tác để tạo ra một hệ thống web hoạt động mượt mà và đáp ứng nhu cầu của người dùng. Trong kiến trúc phân tách này, frontend chịu trách nhiệm về giao diện người dùng và trải nghiệm người dùng, trong khi backend chịu trách nhiệm về xử lý logic, dữ liệu, và tương tác với cơ sở dữ liệu.

### 2.2.3 : Sự khác nhau của Backend và Fontend

Backend và Frontend là hai khía cạnh quan trọng của phát triển web, mỗi khía cạnh đảm nhận các nhiệm vụ và trách nhiệm khác nhau. Dưới đây là sự khác nhau giữa Backend và Frontend:

Backend:

1. Ngôn ngữ Lập trình:

- Backend thường được phát triển bằng các ngôn ngữ lập trình như Java, Python, Ruby, PHP, Node.js, C#.

2. Xử lý Logic:

- Backend chịu trách nhiệm xử lý logic kinh doanh, tính toán, và quyết định dữ liệu nên được lưu trữ, xử lý, và truyền đến Frontend.

3. Cơ sở Dữ liệu:

- Backend tương tác với cơ sở dữ liệu để lưu trữ và truy xuất dữ liệu. Các hệ quản lý cơ sở dữ liệu phổ biến bao gồm MySQL, PostgreSQL, MongoDB, và Redis.

4. APIs (Application Programming Interfaces):

- Backend thường cung cấp các API để cho phép Frontend và các ứng dụng khác tương tác với dữ liệu và chức năng của nó.

5. Authentication và Authorization:

- Backend xác định và quản lý quyền truy cập của người dùng thông qua quá trình xác thực và ủy quyền.

6. An toàn và Bảo mật:

- Backend chịu trách nhiệm về bảo mật, đảm bảo rằng dữ liệu và hệ thống được bảo vệ khỏi các mối đe dọa.

7. Performance và Scalability:

- Backend cần được tối ưu hóa để đảm bảo hiệu suất và khả năng mở rộng để xử lý lượng người dùng ngày càng tăng

Frontend

1. Ngôn ngữ Lập trình:

- Frontend sử dụng HTML, CSS, và JavaScript. Các framework như React, Angular, và Vue.js thường được sử dụng để phát triển Frontend một cách hiệu quả.

2. Giao Diện Người Dùng:

- Frontend chịu trách nhiệm về giao diện người dùng và trải nghiệm người dùng. Nó xác định cách mà thông tin và chức năng được hiển thị trên trình duyệt.

3. Tương Tác Người Dùng:

- Frontend xử lý tương tác người dùng, như các sự kiện click, hover, và thực hiện các thao tác theo yêu cầu của người dùng.

4. Hiệu Ứng và Đồ Họa:

- Frontend đảm nhiệm việc thực hiện các hiệu ứng và đồ họa để tạo ra trải nghiệm người dung hấp dẫn.

5. Responsive Design:

- Frontend cần được thiết kế để tự động điều chỉnh và phản hồi với các kích thước màn hình khác nhau, từ máy tính đến điện thoại di động.

6. Cross-Browser Compatibility:

- Frontend cần hoạt động một cách nhất quán trên nhiều trình duyệt khác nhau, đảm bảo rằng trải nghiệm người dùng không thay đổi.

7. Thử Nghiệm và Kiểm Thử:

- Frontend thường được kiểm thử để đảm bảo tính ổn định và chất lượng. Thử nghiệm đơn vị và kiểm thử tích hợp thường được thực hiện.

8. Version Control:

- Sử dụng hệ thống quản lý phiên bản để theo dõi và quản lý mã nguồn

Tổng Quan:

- Backend và Frontend hoạt động cùng nhau: Backend và Frontend là hai phần của cùng một hệ thống và cần phải làm việc cùng nhau để đảm bảo một ứng dụng web hoạt động mượt mà và hiệu quả.

- Tách Biệt Tính Chức Năng: Backend chủ yếu xử lý logic và dữ liệu, trong khi Frontend chủ yếu tập trung vào giao diện và trải nghiệm người dùng.

- Quản lý Dữ liệu: Backend chịu trách nhiệm lưu trữ và quản lý dữ liệu, trong khi Frontend sử dụng dữ liệu đó để hiển thị thông tin cho người dùng.

Hiểu rõ sự khác nhau giữa Backend và Frontend là quan trọng để có cái nhìn tổng thể về cách một ứng dụng web hoạt động.

### 2.2.4 : Nguyên lý cơ bản của fontend

Nguyên lý cơ bản của frontend tập trung vào việc phát triển và tối ưu hóa giao diện người dùng để đảm bảo trải nghiệm người dùng tốt nhất. Dưới đây là một số nguyên lý cơ bản của frontend:

1. User-Centric (Người Dùng Trung Tâm):

- Đặt người dùng làm trung tâm của quá trình phát triển. Thiết kế và xây dựng giao diện sao cho người dùng dễ sử dụng và hiểu.

2. Responsive Design (Thiết Kế Tương ứng):

- Đảm bảo rằng giao diện có thể hiển thị một cách đẹp mắt và hoạt động mượt mà trên nhiều loại thiết bị và kích thước màn hình khác nhau.

3. Cross-Browser Compatibility (Tương Thích Trình Duyệt)

- Đảm bảo rằng ứng dụng hoạt động đúng trên nhiều trình duyệt khác nhau mà không gặp vấn đề tương thích

4. Performance Optimization (Tối Ưu Hiệu Suất):

- Tối ưu hóa tải trang và thời gian phản hồi để đảm bảo ứng dụng chạy nhanh chóng và hiệu quả.

5. Simplicity (Sự Đơn Giản):

- Giữ cho giao diện đơn giản và dễ hiểu. Người dùng nên có thể tìm thấy thông tin và thực hiện các hành động một cách dễ dàng.

6. Consistency (Nhất Quán):

- Bảo đảm rằng giao diện người dùng giữ một mức độ nhất quán trong cả thiết kế và trải nghiệm sử dụng.

7. Hierarchy (Sự Ưu Tiên)

- Sắp xếp các phần tử trên trang sao cho người dùng có thể hiểu rõ về sự ưu tiên và mức độ quan trọng của chúng.

8. Accessibility (Truy Cập Dễ Dàng):

- Thiết kế sao cho mọi người, kể cả những người có khuyết tật, có thể truy cập và sử dụng một cách thuận tiện.

9. Feedback (Phản Hồi)

- Cung cấp phản hồi người dùng để họ biết rõ họ đã thực hiện hành động nào và ứng dụng đang phản ứng như thế nào

10. Testing (Kiểm Thử):

- Thực hiện kiểm thử đối với nhiều loại thiết bị và trình duyệt để đảm bảo tính nhất quán và hiệu suất.

11. Scalability (Khả Năng Mở Rộng):

- Thiết kế giao diện sao cho nó có khả năng mở rộng, có thể xử lý số lượng người dùng ngày càng tăng.

12.Version Control (Quản Lý Phiên Bản):

- Sử dụng hệ thống quản lý phiên bản để theo dõi và quản lý mã nguồn.

13. Security (Bảo Mật):

- Bảo vệ giao diện và dữ liệu khỏi các mối đe dọa bảo mật như SQL injection, cross-site scripting, và các cuộc tấn công khác.

Những nguyên lý này giúp định hình cách mà frontend được phát triển để đảm bảo một trải nghiệm người dùng tích cực và hiệu quả.

**CHƯƠNG 3 XÂY DỰNG CHƯƠNG TRÌNH**

## **3.1 Phân tích đề bài**

1. Mục Tiêu của Ứng Dụng:

- Xác định mục tiêu chính của ứng dụng: Cung cấp dịch vụ đặt phòng trực tuyến cho khách sạn. Mục tiêu này có thể bao gồm tăng cường trải nghiệm người dùng, giảm thời gian đặt phòng, và nâng cao quản lý đặt phòng của khách sạn.

2. Chức Năng Cơ Bản:

- Đặt ra các chức năng cơ bản mà ứng dụng cần phải hỗ trợ. Ví dụ: đặt phòng theo ngày, xem thông tin phòng và giá cả, xác nhận đặt phòng, quản lý hủy đặt phòng, và quản lý tài khoản người dùng.

3. Giao Diện Người Dùng:

- Mô tả yêu cầu về giao diện người dùng (UI). Điều này bao gồm cách hiển thị thông tin phòng, quy trình đặt phòng, các biểu tượng và nút điều hướng, cũng như thiết kế tổng thể để tạo ra trải nghiệm người dùng dễ sử dụng và thuận lợi.

4. Giao Diện Quản Trị:

- Xác định các chức năng cần thiết cho giao diện quản trị. Nó có thể bao gồm quản lý phòng, quản lý đặt phòng, thống kê, và các chức năng khác để giúp quản trị viên của khách sạn quản lý hệ thống dễ dàng.

5. Tính Năng Bảo Mật:

- Đặt ra các biện pháp bảo mật như xác thực người dùng, kết nối an toàn qua HTTPS, và quản lý an toàn thông tin cá nhân để bảo vệ dữ liệu người dùng và thông tin đặt phòng.

6. Dữ Liệu và Cơ Sở Dữ Liệu:

- Xác định cách thông tin đặt phòng và thông tin về phòng được lưu trữ. Sử dụng cơ sở dữ liệu phù hợp như MySQL, PostgreSQL hoặc MongoDB để quản lý dữ liệu một cách hiệu quả.

7. Tương Tác Hệ Thống:

- Xác định cách mà ứng dụng sẽ tương tác với hệ thống bên ngoài, chẳng hạn như gửi email xác nhận đặt phòng, tích hợp thanh toán trực tuyến, hoặc tương tác với các hệ thống quản lý khác của khách sạn.

8. Tích Hợp Thanh Toán:

- Nếu có tính năng thanh toán trực tuyến, xác định cách tích hợp các cổng thanh toán an toàn và đáng tin cậy để xử lý thanh toán từ khách hàng.

9. Kiểm Thử và Debugging:

- Xác định quy trình kiểm thử để đảm bảo tính ổn định, hiệu suất, và bảo mật của ứng dụng. Chuẩn bị kế hoạch debugging để xử lý các vấn đề một cách hiệu quả.

10. Responsive Design và Tương Thích Trình Duyệt:

- Đảm bảo rằng giao diện người dùng và giao diện quản trị đều hỗ trợ responsive design và tương thích trên nhiều trình duyệt khác nhau.

11. Thống Kê và Phản Hồi:

- Xem xét việc tích hợp các công cụ thống kê để theo dõi hoạt động người dùng và thu thập phản hồi để cải thiện trải nghiệm người dùng và chức năng của ứng dụng.

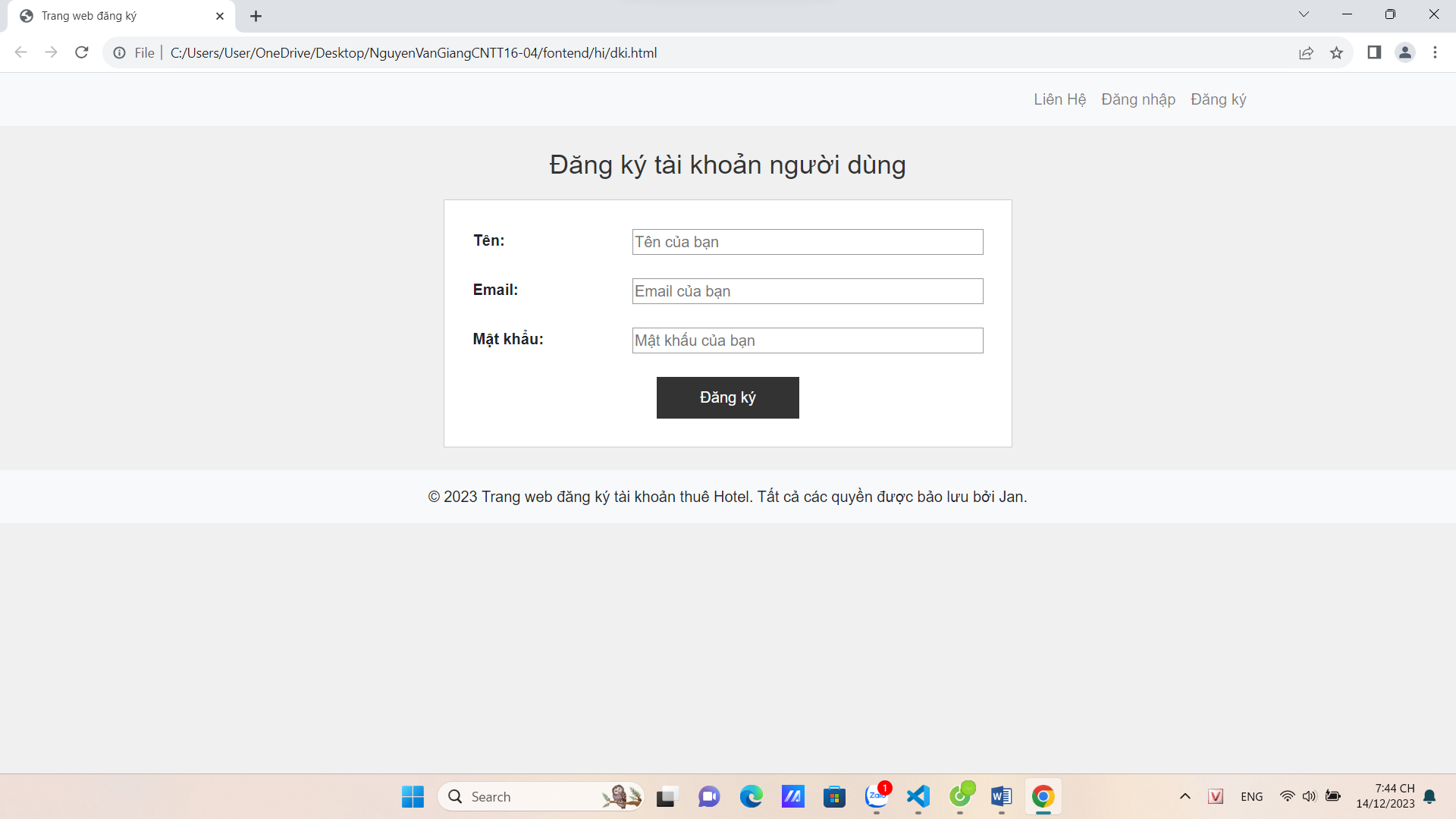
12. Phân Quyền Truy Cập:

- Đặt ra cơ chế phân quyền truy cập để quản trị viên có thể quản lý thông tin và chức năng theo cách hiệu quả.

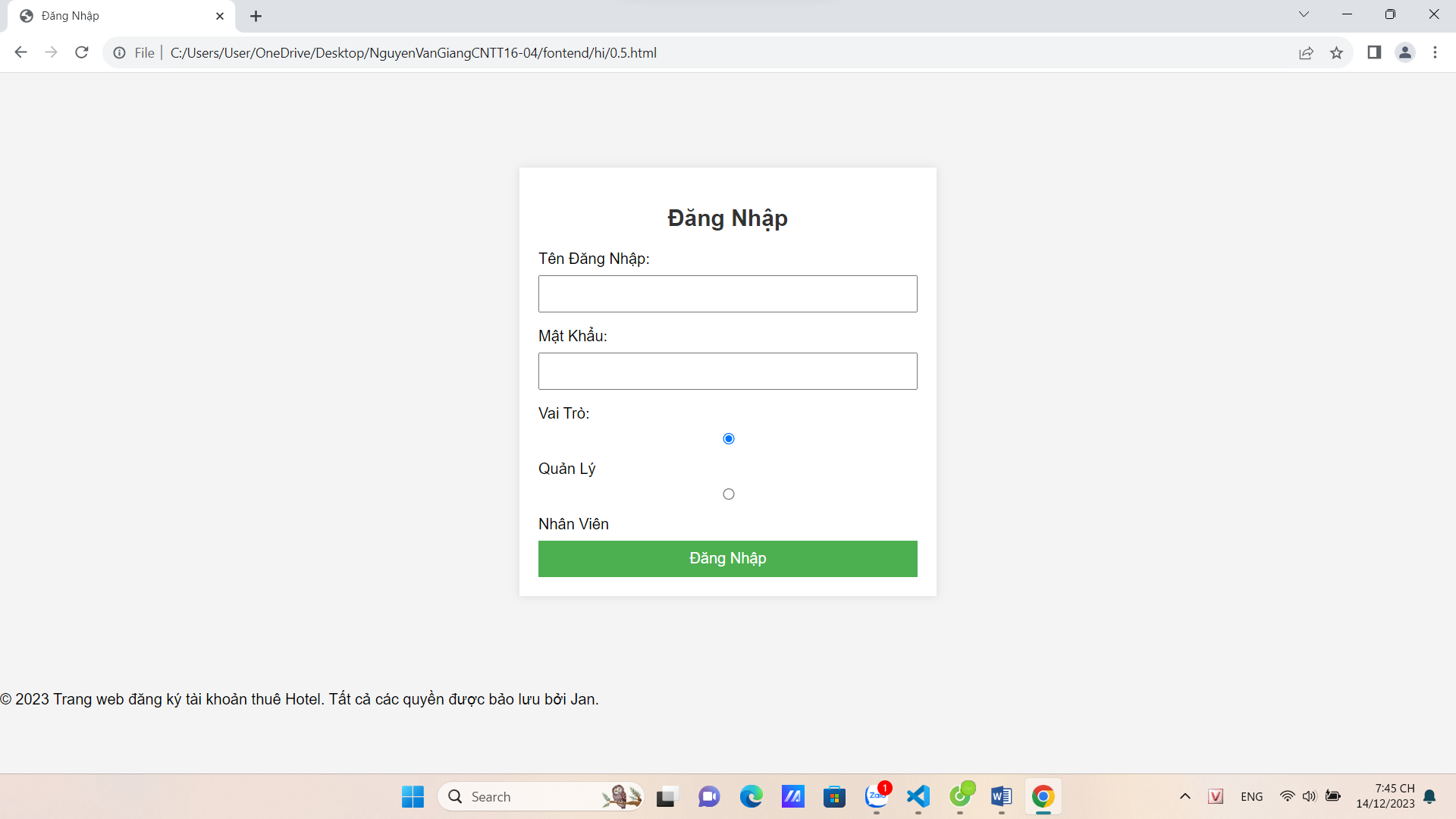
Bằng cách phân tích chi tiết các yêu cầu và chức năng, bạn có thể xây dựng một kế hoạch phát triển và triển khai ứng dụng web đặt phòng trực tuyến cho khách sạn.

## **3.2 : Trình bầy bài làm**

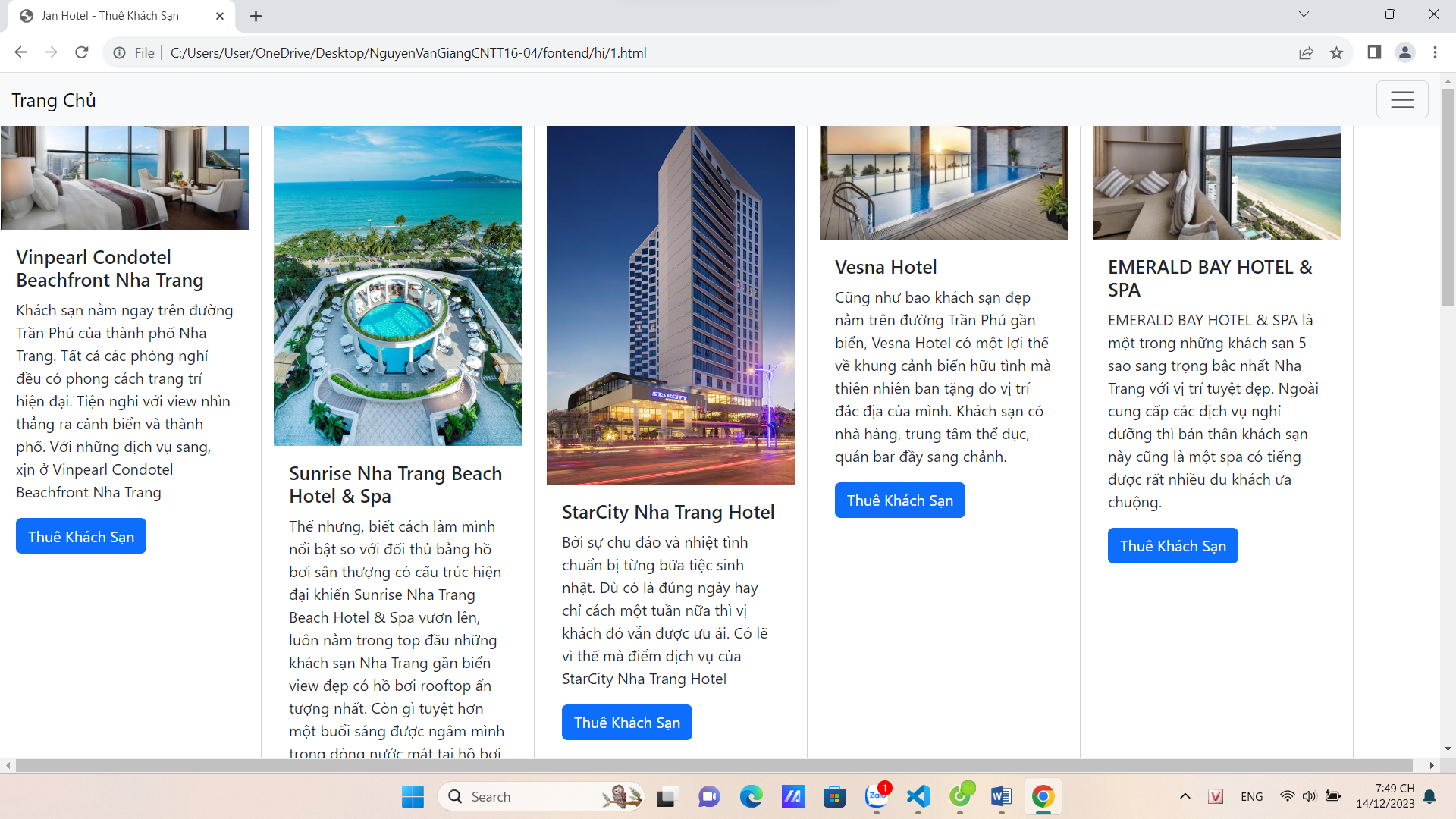
3.2.1 : Giao diện đăng nhập và đăng ký



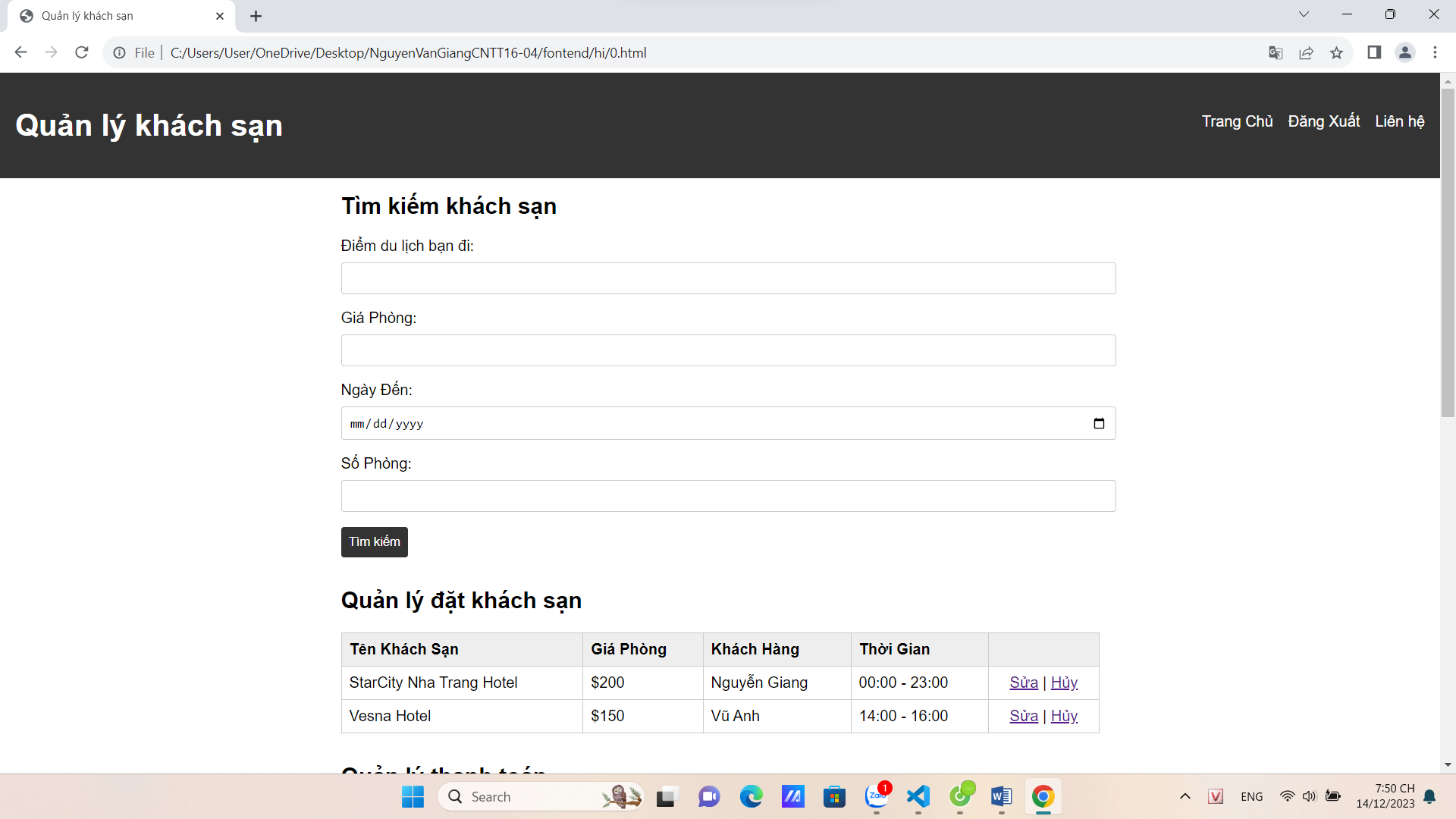
Hình 5 : Trang đăng ký tài khoản



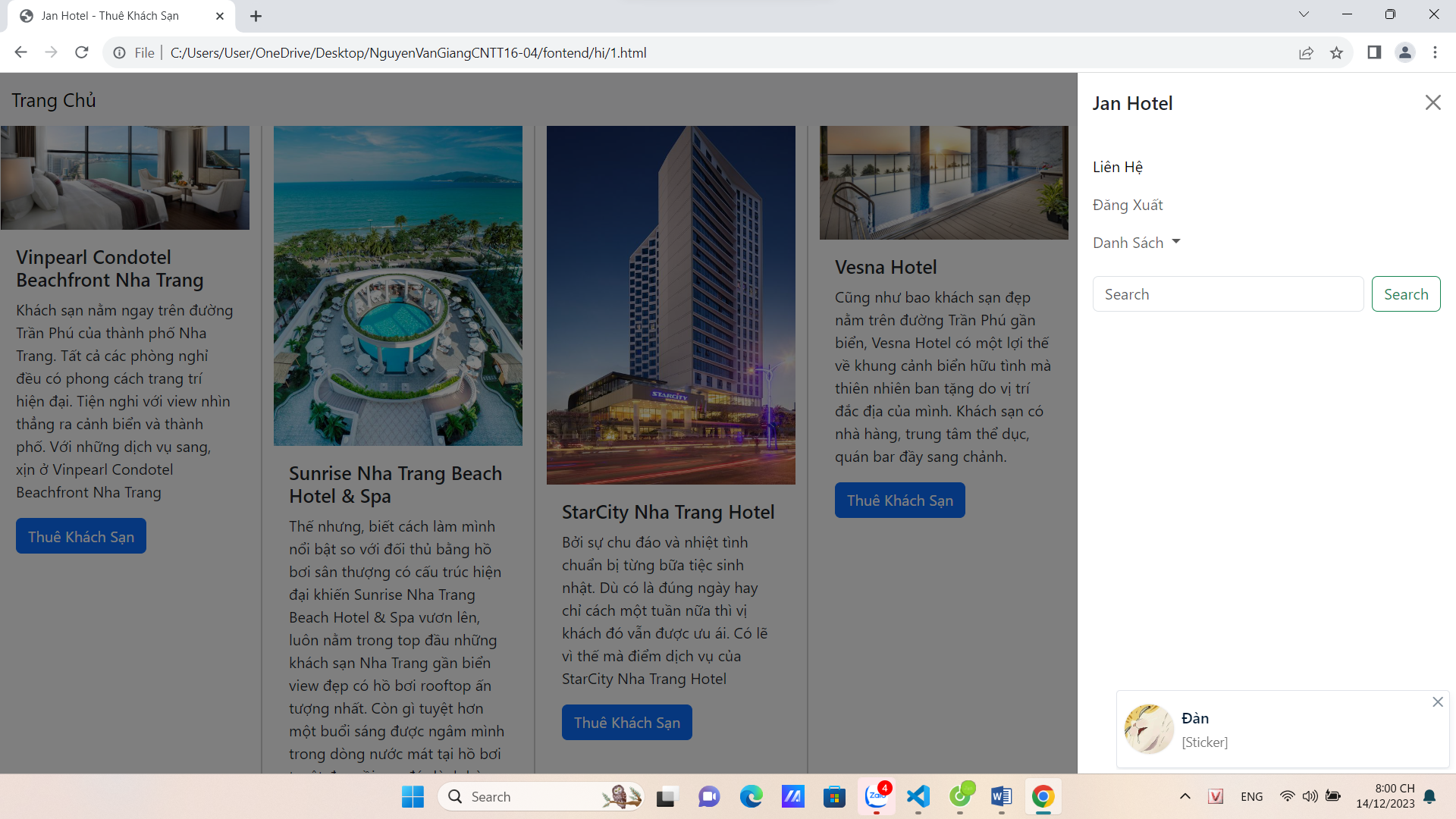
Hình 6 : Trang đăng nhập có 2 lựa chọn là đăng nhập vào trang quản lý Khách Sạn hoặc Khách Hàng vào book phòng



Hình 7 : Giao diện của Khách Hàng chọn phòng để book đi du lịch



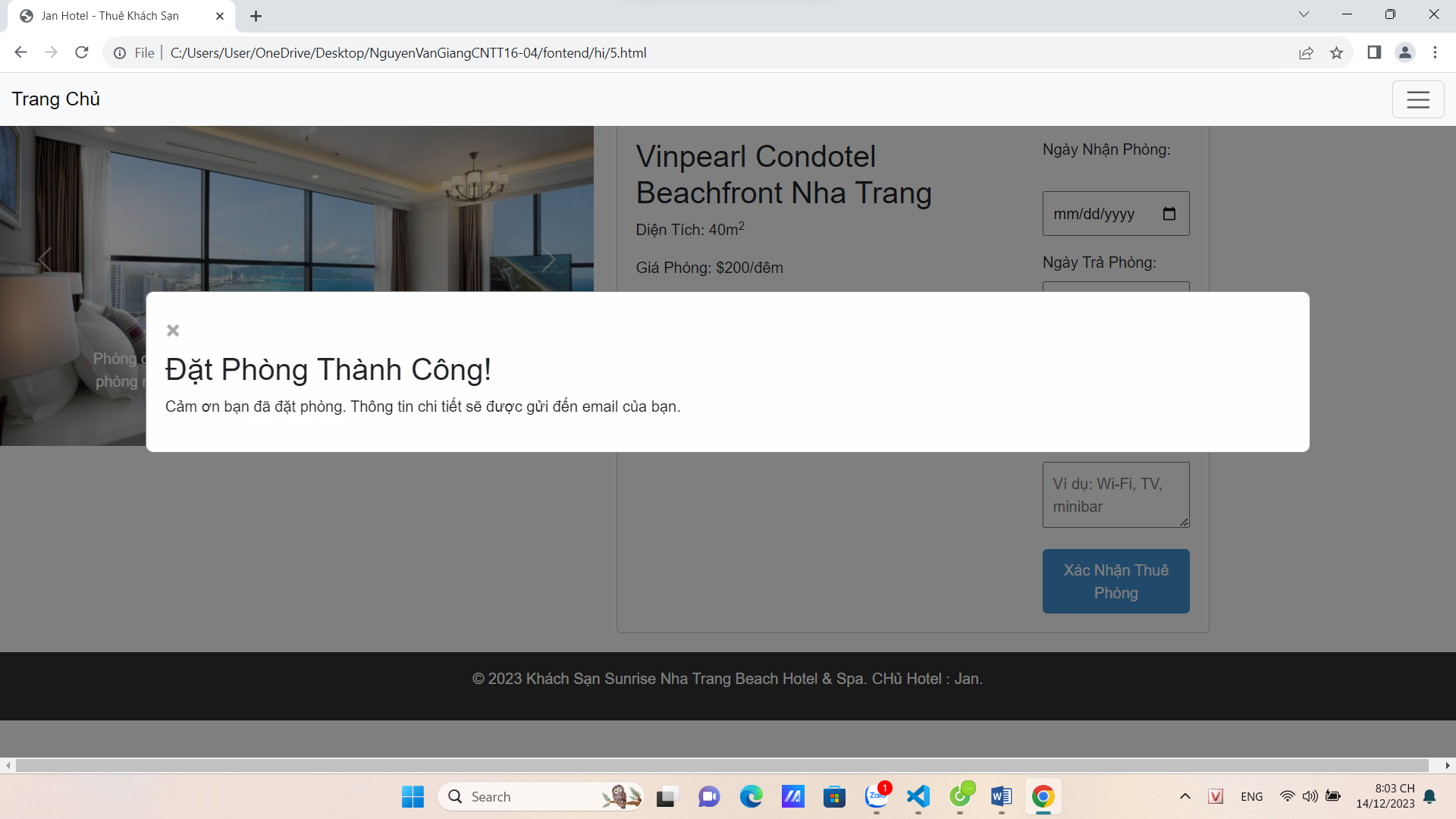
Hình 8 : Giao diện nhân viên quản lý web mà khách hang vào book phòng



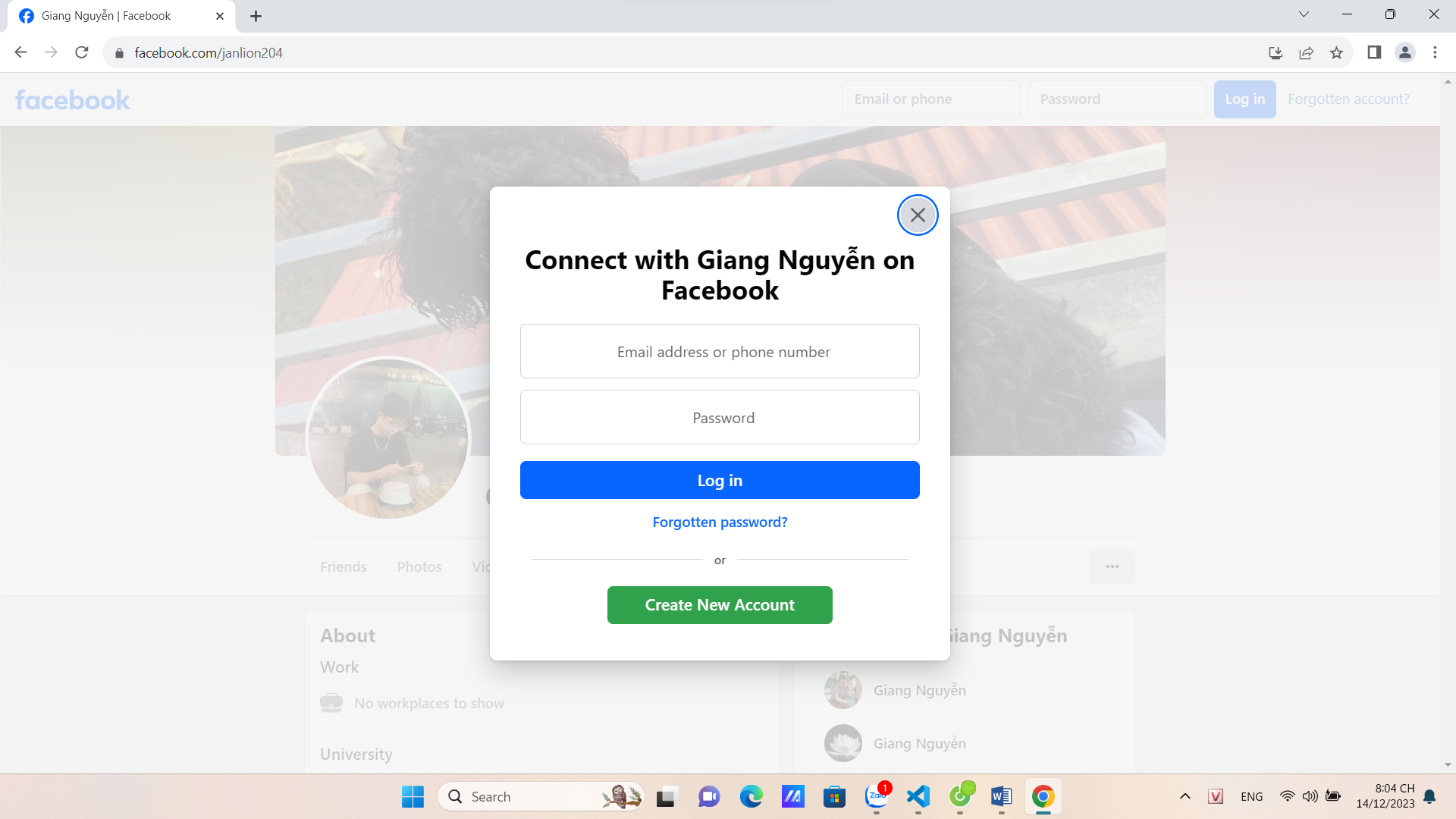
Hình 9 : Chọn các lựa trọn bên phải liên hệ đăng xuất và tìm các Hotel nơi các bạn đi du lịch



Hình 10 : Giao diện khách hang khi chọn Hotel



Hình 11 : Khi bạn đặt phòng thành công



Hình 12 : Liên hệ ở đây

**KẾT LUẬN**

1. Ưu Điểm:

Phát triển ứng dụng web quản lý đặt phòng trực tuyến cho khách sạn mang lại nhiều ưu điểm quan trọng:

- Tiện Ích Cho Người Sử Dụng:

- Dễ Sử Dụng: Giao diện người dùng thân thiện và dễ sử dụng giúp khách hàng dễ dàng đặt phòng một cách nhanh chóng và thuận tiện.

- Tiện Lợi: Khả năng đặt phòng trực tuyến giúp khách hàng tiết kiệm thời gian và công sức so với việc đến trực tiếp khách sạn hoặc gọi điện để đặt phòng.

- Hiệu Suất Cao:

- Tự Động Hóa Quy Trình:Giao diện quản trị giúp quản lý dễ dàng theo dõi và kiểm soát tình trạng phòng, đồng thời giảm bớt công việc thủ công.

- Quản Lý Hiệu Quả:

- Tổ Chức Thông Tin: Giao diện quản trị cung cấp cái nhìn toàn diện về tình trạng phòng, đặt phòng và các dịch vụ khác, giúp quản lý dễ dàng ra quyết định và tổ chức công việc.

2. Nhược Điểm:

Mặc dù có nhiều ưu điểm, nhưng cũng có những thách thức cần đối mặt khi phát triển ứng dụng web đặt phòng cho khách sạn:

- Bảo Mật và Riêng Tư:

- Rủi Ro An Ninh: Với thông tin cá nhân và tài khoản thanh toán, đảm bảo an ninh thông tin là một thách thức lớn.

- Quản Lý Thông Tin: Việc quản lý thông tin khách hàng đòi hỏi sự chú ý đặc biệt để tránh rủi ro về việc mất mát dữ liệu.

- Tương Tác và Hỗ Trợ:

- Tương Tác Trực Tiếp: Thiếu tương tác trực tiếp với nhân viên có thể làm giảm trải nghiệm của khách hàng, đặc biệt là khi có vấn đề cần giải quyết.

- Hỗ Trợ Trực Tuyến: Cần có giải pháp hỗ trợ trực tuyến hiệu quả để giúp khách hàng khi gặp vấn đề trong quá trình đặt phòng.

3. Hướng Phát Triển:

Để phát triển ứng dụng web đặt phòng cho khách sạn một cách hiệu quả, có một số hướng phát triển quan trọng:

- ích Hợp Trải Nghiệm Người Dùng:

- Giao Diện Người Dùng: Cải thiện trải nghiệm người dùng để làm cho quy trình đặt phòng trở nên linh hoạt và thú vị.

- Responsive Design: Đảm bảo giao diện người dùng tương thích với các thiết bị di động để mở rộng đối tượng sử dụng.

- Bảo Mật và Tuân Thủ Pháp Luật:

- SSL và Bảo Mật Thanh Toán: Đảm bảo rằng tất cả thông tin thanh toán được bảo mật với SSL và tuân thủ các chuẩn an ninh dữ liệu.

- Tuân Thủ Pháp Luật: Tuân thủ các quy định pháp luật về bảo mật dữ liệu và quyền riêng tư khách hàng.

- Hệ Thống Quản Lý Phòng:

- Thông Tin Phòng Chi Tiết: Mở rộng tính năng hiển thị thông tin chi tiết về từng loại phòng và tiện ích kèm theo.

- Quản Lý Giá và Ưu Đãi: Tích hợp hệ thống quản lý giá và ưu đãi để thu hút khách hàng và tối ưu hóa doanh thu.

- Hỗ Trợ và Tương Tác Khách Hàng:

- Hỗ Trợ Trực Tuyến: Tích hợp các tùy chọn hỗ trợ trực tuyến như chat trực tiếp để giúp khách hàng nhanh chóng giải quyết vấn đề.

- Tương Tác Thêm: Xem xét việc thêm các tùy chọn tương tác, chẳng hạn như đánh giá và đánh giá khách hàng.

Phát triển một ứng dụng web đặt phòng trực tuyến cho khách sạn đòi hỏi sự linh hoạt, chú ý đến chi tiết và luôn cập nhật để đáp ứng nhu cầu đa dạng của khách hàng và giúp doanh nghiệp phát triển mạnh mẽ.

**DANH MỤC TÀI LIỆU THAM KHẢO**

Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.

Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.

James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.

**QUY ĐỊNH TRÌNH BÀY TRONG BÁO CÁO CÀI TẬP LỚN**

- Bài tập lớn được in trên một mặt giấy trắng khổ A4 (210 x 297mm), dày lớn hơn 30 trang, nhỏ hơn 100 trang, không kể hình vẽ, bảng biểu, đồ thị và danh mục tài liệu tham khảo.

- Phần nội dung trình bày trong bài tập lớn sử dụng Font chữ **Times New Roman** cỡ **13**, hệ soạn thảo Microsoft Word; mật độ chữ bình thường, không được nén hoặc kéo dãn khoảng cách giữa các chữ; dãn dòng đặt ở chế độ **1,5 lines**; lề trên **2,0 cm**; lề dưới **2,0 cm**; lề trái **2,5 cm**, lề phải **2,0 cm**. Số trang được đánh ở giữa, phía dưới trang giấy.

- Cách ghi trích dẫn tài liệu tham khảo: Cuối đoạn trích dẫn đánh số thứ tự tài liệu tham khảo (ví dụ: [1]: tham khảo tài liệu số 1; [3,4,8]: tham khảo 3 tài liệu số 3, 4, 8).

- Tuyệt đối không được tẩy, xoá, sửa chữa trong bài tập lớn.

- Quy cách trình bày nội dung

|  |  |  |  |
| --- | --- | --- | --- |
| **Đề mục** | **Cỡ chữ** | **Định dạng** | **Canh lề trang** |
| Tên chương | 14 | In hoa, đậm | Giữa |
| Tên tiểu mục mức 1 | 13 | Chữ thường, đậm | Trái |
| Tên tiểu mục mức 2 | 13 | Chữ thường, đậm, nghiêng | Trái |
| Tên tiểu mục mức 3 | 13 | Đánh chỉ mục bằng chữ cái thường  a), ...  b), ..... | Trái |
| Nội dung | 13 | Chữ thường (Normal) | Đều hai bên |
| Nội dung bảng (table) | 12 | Normal | Giữa ô |
| Tên bảng | 12 | Chữ thường, nghiêng | Giữa, trên bảng |
| Tên hình | 12 | Chữ thường, nghiêng | Giữa, dưới hình |
| Tài liệu tham khảo | 12 | APA style | Chú thích bên dưới |

***Cách đánh dấu câu****:*

Các dấu: : , . ; ) } ] ! ? ” được gõ ngay sau ký tự cuối cùng (không khoảng cách), và gõ 1 phím cách (space) sau chúng. Sau các dấu: “{ ( [ không gõ dấu cách.

***\* Cách đánh số các tiểu đề mục nhiều nhất là 3 mức và không lùi sang phải***

Hướng dẫn xếp tài liệu tham khảo

1. Tài liệu tham khảo xếp theo thứ tự trích dẫn trong bài tập lớn.
2. Tài liệu tham khảo là sách, luận án, báo cáo phải ghi đầy đủ các thông tin sau:

* Tên các tác giả hoặc cơ quan ban hành (không có dấu ngăn cách)
* (năm xuất bản), (đặt trong ngoặc đơn, dấu phẩy sau ngoặc đơn)
* Tên sách, luận án hoặc báo cáo, (in nghiêng, dấu phẩy cuối tên)
* Nhà xuất bản, (dấu phẩy cuối tên nhà xuất bản)
* Nơi xuất bản, (dấu chấm kết thúc tài liệu tham khảo).

Tài liệu tham khảo là bài báo trong tạp chí, bài trong một cuốn sách… ghi đầy đủ các thông tin sau:

* Tên các tác giả (không có dấu ngăn cách);
* (Năm công bố), (đặt trong ngoặc đơn, dấu phẩy sau ngoặc đơn)
* “Tên bài báo”, (đặt trong ngoặc kép, không in nghiên, dấu phẩy cuối tên)
* Tên tạp chí hoặc tên sách, (in nghiên, dấu phẩy cuối tên)
* Tập (không có dấu ngăn cách)
* (Sổ), (đặt trong ngoặc đơn, dấu phẩy sau ngoặc đơn)
* Các số trang, (gạch ngang giữa hai chữ số, dấu chấm kết thúc)

Cần chú ý những chi tiết về trình bày nêu trên. Nếu tài liệu dài hơn một dòng thì nên trình bày sau cho từ dòng thứ hai lùi vào so với dòng thứ nhất 1 cm để phần tài liệu tham khảo được rõ ràng và dễ theo dõi.

Ví dụ:

1. Nguyễn Hồng Sơn (2007), *Giáo trình hệ thống Mạng máy tính CCNA* (Semester 1), NXB Lao động xã hội.
2. Phạm Quốc Hùng (2017), *Đề cương bài giảng Mạng máy tính*, Đại học SPKT Hưng Yên.
3. James F. Kurose and Keith W. Ross (2013), *Computer Networking: A top-down approach sixth Edition*, Pearson Education.