TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HÒ CHÍ MINH KHOA CÔNG NGHỆ THÔNG TIN



THỰC TẬP TỐT NGHIỆP

XÂY DỰNG GAME XÉP NHÀ TRÊN NỀN TẢNG WEB

Ngành: CÔNG NGHỆ THÔNG TIN

Chuyên ngành: CÔNG NGHỆ THÔNG TIN

Giảng viên hướng dẫn : Trần Đức Doanh

Sinh viên thực hiện :

1.Đinh Viết Sửu - 1551120046 - CN15A

2.Lê Đức Hoàng - 1551120012 - CN15A

3. Vũ Trường - 1551120060 - CN15A

TP. Hồ Chí Minh, 2019

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

TP. Hồ Chí Minh, ngày tháng năm
Giáo viên hướng dẫn

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Т	TP. Hồ Chí Minh, ngày tháng năm
	Giáo viên phản biện

LÒI CẨM ƠN

Lời nói đầu tiên cho phép nhóm em gửi lời cảm ơn chân thành đến ban giám hiệu nhà trường Đại học Giao thông vận tải thành phố Hồ Chí Minh, khoa Công nghệ thông tin, đặc biệt là thầy Trần Dức Doanh đã tạo điều kiện và hướng dẫn để nhóm có cơ hội được thực hiện đề tài thực tập tốt nghiệp này.

Nhóm xin được chân thành gửi lời cảm ơn sâu sắc tới các thầy cô giáo trong trường Đạt học Giao thông vận tải thành phố Hồ Chí Minh nói chung và các thầy cô trong Khoa công nghệ thông tin nói riêng, những người đã tận tình giảng dạy và truyền đạt những kiến thức, những kinh nghiệm quý báu trong suốt quá trình học tập và rèn luyện từ lúc vào học trường Đại học Giao thông vận tải thành phố Hồ Chí Minh.

Vì thời gian có hạn cũng như kinh nhiệm còn ít nên đề tài của nhóm không tránh khỏi những thiếu sót. Nhóm em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của thầy để nhóm em có điều kiện bổ sung và hoàn thiện đề tài.

Một lần nữa, nhóm em xin chân thành cảm ơn!

LÒI CAM ĐOAN

Tôi xin cam đoan đồ án Thực tập tốt nghiệp "Xây dựng game xếp nhà trên nền tảng web" là công trình nghiên cứu của nhóm. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

LÒI CAM ĐOAN

Tôi xin cam đoan đồ án Thực tập tốt nghiệp "Xây dựng game xếp nhà trên nền tảng web" là công trình nghiên cứu của nhóm. Những phần sử dụng tài liệu tham khảo trong đồ án đã được nêu rõ trong phần tài liệu tham khảo. Các số liệu, kết quả trình bày trong đồ án là hoàn toàn trung thực, nếu sai tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của bộ môn và nhà trường đề ra.

MỤC LỤC

MỤC LỤC	i
MỤC LỤC HÌNH ẢNH	iv
DANH MỤC TỪ VIẾT TẮT	v
CHƯƠNG 1: LỜI MỞ ĐẦU	1
CHƯƠNG 2: NHỮNG PHẦN SỄ THỰC HIỆN	2
CHƯƠNG 3: NGHIÊN CỨU CÁC CÔNG NGHỆ SỬ DỤNG	3
3.1. HTML (Hypertext Markup Language)	3
3.1.1. Giới thiệu về HTML	3
3.1.2. Quá trình hình thành và phát triển của HTML	3
3.1.2. Hoạt động của HTML	6
3.1.3 Tổng quan các thể thông dụng của HTML	7
3.1.3.1. Block Tags	7
3.1.3.2. Inline Tag	18
3.1.4. HTML 5 và sự khác biệt	23
3.1.5. Ưu điểm và nhược điểm của HTML	24
3.1.5.2. Khuyết điểm	24
3.2. CSS (Cascading Style Sheets)	25
3.2.1. Quá trình hình thành và phát triển	25
3.2.2. Hoạt động của CSS	26
3.2.3. Tổng quan các thuộc tính hay dùng	27

3.2.3.2.	Basic box Display	31
3.2.3.3.	Flex-box Flex	35
3.2.3.4.	Các thuộc tính text Text-align	37
3.2.3.5.	Các thuộc tính về font @font-face	39
3.2.3.6.	Các thuộc tính Animation @#keyframe	41
3.2.3.7.	Các thuộc tính Transform Perspective	43
3.3. Jav	ascript	45
3.3.1.	Tổng quan về Javascript	45
3.3.2.	Quá trình phát triển	45
3.3.3.	Ưu và nhược điểm	49
3.3.3.2.	Nhược điểm	49
3.3.3.	Hoạt động của Javascript	51
3.3.4.	Sự khác biệt giữa Javascript và các ngôn ngữ lập trình khác	53
3.3.5.	Các thư viện Javascript nổi tiếng hiện nay	54
3.4. Hu	ớng đối tượng trong lập trình (Object Oriented Programing)	54
3.4.1.	Tổng quan	54
3.4.2.	4 đặc tính của lập trình hướng đối tượng	55
3.4.2.2.	Tính kế thừa (Inheritance)	57
3.4.2.3.	Tính đa hình (Polymorphism)	59
3.4.2.4.	Tính trừu tượng (Abstraction)	60
HUONO	G 4: PHÂN TÍCH NỘI DUNG CỦA GAME	62

4.1. Tổng quan game	62
4.2. Giao diện của game	63
CHƯƠNG 5: KẾT LUẬN	68
CHƯƠNG 6: HƯỚNG PHÁT TRIỀN	69
CHƯƠNG 7: TÀI LIỆU THAM KHẢO	70

MỤC LỤC HÌNH ẢNH

Hình 3.1 Quá trình phát triển của HTML	8
Hình 3.2 Ba thành phần không thể thiếu của một trang web	37
Hình 3.3 Sơ đồ hoạt động của Javascript	40
Hình 3.4 4 đặc điểm của lập trình hướng đối tượng	47
Hình 4.1 Giao diện mở đầu	49
Hình 4.2 Giao diện khi chơi game và hướng dẫn chơi game	50
Hình 4.3 Giao diện khi chơi game	50
Hình 4.4 Các thành phần trong game	51
Hình 4.5 Giao diên game over	52

DANH MỤC TỪ VIẾT TẮT

STT	Tên viết tắt	Tên đầy đủ
1	CERN	European Organisation for Nuclear Research
2	CSS	Cascading Style Sheets
3	OOP	Object oriented program
4	HTML	Hyper Text Markup Language
5	PHP	Hypertext Preprocessor
6	W3C	World Wide Web Consortium

CHƯƠNG 1: LỜI MỞ ĐẦU

Trong những năm gần đây, Internet trở nên là một phần không thể thiếu đối với cuộc sống thường ngày, nó cho phép chúng ta tiếp cận với lượng thông tin khổng lồ và từ nhiều nguồn khác nhau. Ngoài ra, việc giải trí trên môi trường Internet cũng phát triển một cách mạnh mẽ, đặc biệt là game online. Cho phép hàng triệu người có thể tiếp cận và tương tác với nhau.

Ngoài những tựa game online phải được cài đặt thông qua máy tính thì người dùng mới có thể sử dụng được thì nền tảng game trên web lại cho phép người sử dụng có thể sử dụng trực tiếp mà không cần thông qua bước cài đặt phức tạp như trên. Nghĩ đến điều đó, nhóm quyết định phát triển một game nhỏ mà có thể hoạt động trên hầu hết các trình duyệt web hiện nay. Game có tên là Game xếp nhà.

Game sẽ được thiết kế bố cục sử dụng HTML (Hyper Text Markup Language) và CSS (Cascading Style Sheets) và phát triển ở phía người dùng sử dụng Javascript và các thư viện mã nguồn mở Javascript khác.

CHƯƠNG 2: NHỮNG PHẦN SẼ THỰC HIỆN

Tìm hiểu HTML, là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẫu thông tin được trình bày trên World Wide Web. Nó dung để định dạng bố cục, các thuộc tính liên quan đến hiể thị và được hiển thị trên một chương trình đặc biệt gọi là Browser.

Tìm hiểu Javascript, là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc định nghĩa ra, Javascript được sử dụng rộng rãi trong các ứng dụng Website và hỗ trợ hầu hết tất cả các trình duyệt hiện nay như: Google Chrome, Microsoft EDGE, Mozilla Firefox, Opera, ...

Tìm hiểu CSS, là viết tắt của Cascading Style Sheet, được dùng để tìm và định dạng lại các phần tử được tạo ra bởi HTML. Giúp cho các phần tử này trở nên sinh động hơn về màu sắc, kiểu chữ, viền, hiệu ứng, ...

Tìm hiểu về lập trình hướng đối tượng, là một trong những kỹ thuật lập trình rất quan trọng hiện nay. Áp dụng cho hầu hết các ngôn ngữ hiện đại.

Cụ thể về ứng dụng minh họa, nhóm em sẽ xây dựng một webgame nhỏ cho phép người sử dụng tương tác và điều khiển với tốc độ nhanh và trải nghiệm tốt nhất.

CHƯƠNG 3: NGHIÊN CỨU CÁC CÔNG NGHỆ SỬ DỤNG

3.1. HTML (Hypertext Markup Language)

3.1.1. Giới thiệu về HTML

Đối với mỗi người phát triển web, chắc hẳn không có ai là không biết đến HTML một ngôn ngữ đánh dấu văn bản nó giúp cho người dùng tạo và cấu trúc các thành phần trên một trang web, phân chia các đoạn văn bản, phần Header, Body, Footer, Link, Blockquote, ...

HTML không phải là một ngôn ngữ lập trình, đồng nghĩa với việc nó không thể tạo ra các chức năng động được. Nó cũng giống như phần mềm Microsoft Word, dùng để bố cục và định dạng trang web.

3.1.2. Quá trình hình thành và phát triển của HTML

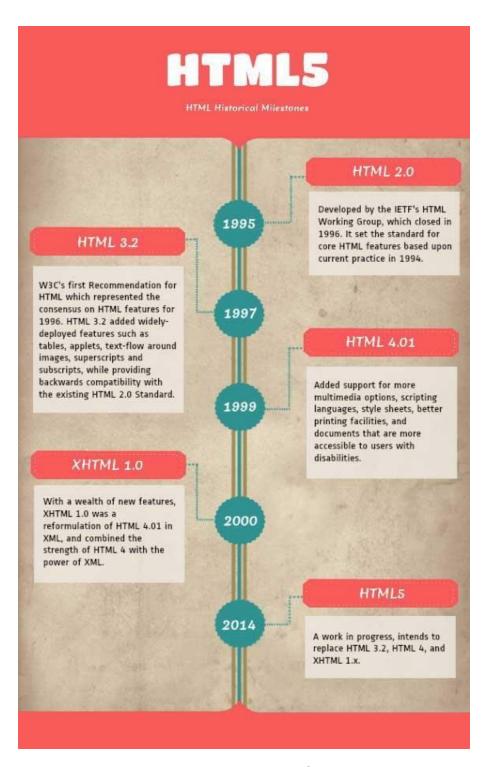
HTML được sang tạo ra bởi Tim Berners-Lee, là mọt nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Anh ta đã nghĩ ra được ý tưởng cho hệ thống hypertext trên nền Internet. Ý trưởng này xuất hiện vào năm 1989 khi Tim làm việc tại văn phòng dịch vụ điện toán ở CERN. Các nghiên cứu vật lý thường yêu cầu sự hợp tác của các nhà khoa học khắp mọi nơi trên thế giới, và để tiện cho việc tra cứu tài liệu, ý tưởng của Tim là tạo ra một nơi để mọi người kết nối vào và từ đó xem các tài liệu các thông qua các liên kết. Ý tưởng này là đột phá, thông thường

mọi người phải truy cập rồi tải tất cả tài lệu về đọc nhưng Tim muốn rằng các tài liệu phải được liên kết với nhau.

Hypertext có nghĩa là văn bản chứa liên kết, nơi người dùng có thể truy cập liên kết ngay lập tức. Anh cho ra phiên bản đầu tiên của HTML trong năm 1991 bao gồm 18 thẻ HTML. Từ đó, mỗi phiên bản mới của HTML đều có thêm các thẻ mới và thuộc tính mới.

Theo Mozilla Developer Network: HTML hiện tại có hơn 140 thẻ, mặc dù một vài thẻ trong số chúng đã bị ngưng sử dụng (do không hỗ trợ các trình duyệt hiện đại).

Nhanh chóng phổ biến ở mức độ chóng mặt, HTML được xem là chuẩn mực của một Website. Các thiết lập và cấu trúc HTML được vận hành và phát triển bởi World Wide Web Consortium (W3C). Chúng ta có thể kiểm tra tình trạng mới nhất của ngôn ngữ này bất kỳ lúc nào trên trang https://www.w3.org/



Hình 3.1 Quá trình phát triển của HTML

3.1.2. Hoạt động của HTML

HTML Document là các file kết thúc với đuôi **.html hay .htm**. Chúng ta có thể xem chúng bằng cách sử dụng bất kì trình duyệt web nào (như Google Chrome, Safari, Mozilla Firefox, ...). Trình duyệt đọc các file HTML này và xuất nột dung lên sao cho người dùng có thể xem được nó.

Thông thường trung bình một Website chứa nhiều trang web HTML khác nhau, ví dụ như: trang chủ, trang liên hệ, trang tìm kiếm, ... Tất cả đều là các trang HTML riêng biệt.

Mỗi trang chứa một bộ các thẻ (cũng được gọi là Elements), có thể xem như là việc xây dựng từng khối của một trang web. Nó tạo thành cấu trúc như một cây thư mục bao gồm các thẻ chính để chia bố cục như: header, footer, section, paragraph, và những khối nội dung khác.

Hầu hết các HTML elements đều có thẻ mở và thẻ đóng với cấu trúc tag>. Bên dưới là một ví du nhỏ về cấ trúc của một đoạn code HTML:

Element ngoài cùng là thẻ **div** (division) bao bọc lấy các thẻ khác ở trong. Dùng để markup cho toàn bộ nội dung lớn. Bên trong bao gồm hai thẻ tiêu đề **h1, h2**, một thẻ văn bản **p** và một thẻ ảnh **img**.

3.1.3 Tổng quan các thể thông dụng của HTML

HTML có hai loại thẻ chính là: block tag và inline tag.

- 1. Phần tử khối (block tag) là thuật ngữ chỉ chung các thẻ HTML có chức năng tạo ra một khu vực hay nói cách khác là một khối. Khối này có nghĩa là một thẻ mà khi ta khai báo nó thì nó sẽ được hiển thị ở mỗi dòng riêng biệt bao gồm các nộ dung khác nằm bên trong. Một số block tags cơ bản như: , , , <div>, <h1>, <h2> ...
- 2. Phần tử dòng (inline tags) là thuật ngữ chỉ chung các thẻ HTML khi mà khai báo vào nội dung thì nó vẫn sẽ nằm chung một dòng với các văn bản khác. Một sô thẻ inline rất hay dùng đó là **, **, **<i>, <u>**,

, ...

Chúng ta có thể đổi kiểu hiển thị của một phần tử bất kì từ inline sang block và ngược lại, hay chuyển sang kiểu hiển thị khác nữa là *table* với thuộc tính *display* trong CSS sẽ được trình bày ở phần sau.

3.1.3.1. Block Tags

<html>

Thẻ <html> cho trình duyệt biết đây là tài liệu HTML, <html> thể hiện gốc của tài liệu HTML.

<head>

Thẻ <head> là một vùng chứa cho tất cả các phần tử head. Thẻ <head> có thể bao gồm thẻ <title>, , <meta> và hơn thế nữa.

k>

Thẻ liên kết giữa tài liệu với các nguồn dữ liệu bên ngoài. Thẻ cũng giúp liên kết với các thành phần CSS và JS mà đề cập ở phần sau của bản báo cáo.

```
<link rel="stylesheet" href="./css/owlcarousel.css">
<link rel="stylesheet" type="text/css"
href="./css/owl.carousel.css">
<link rel="stylesheet" type="text/css"
href="./css/owl.theme.default.css">
```

<style>

Thẻ <style> chứa các định dạng cho các thành phần html. Thẻ <style> chỉ định các thành phần này được hiển thị như thế nào trên trình duyệt.

```
    table{
        border-collapse: collapse;
}
img{
        color: #141414;
        border: 1px solid currentColor;
        width: 500px;
        height: auto;
        clip-path: polygon(0 0,100px 0,100px 100px,0 100px);
}
</style>
```

<meta>

Thẻ <meta> cung cấp siêu dữ liệu về tài liệu HTML. Siêu dữ liệu sẽ không được hiển thị trên trang, nhưng sẽ được phân tích cú pháp bằng máy. Các phần tử meta thường được sử dụng để chỉ định mô tả trang, từ khóa, tác giả của tài liệu, sửa đổi lần cuối và siêu dữ liệu khác.

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width , initial-
scale=1.0">
<meta name="description" content="Free Web tutorials">
<meta name="keywords" content="HTML,CSS,XML,JavaScript">
<meta name="author" content="Duy">
```

<body>

Thẻ <body> xác định phần thân của tài liệu. Phần tử <body> chứa tất cả nội dung của tài liệu HTML, chẳng hạn như văn bản, siêu liên kết, hình ảnh, bảng, danh sách, v.v.

<div>

Thẻ <div> được sử dụng để định nghĩa một khu vực trong tài liệu của bạn. Với thẻ div ta có thể nhóm các khu vực lớn của các phần tử HTML với nhau và định dạng chúng với CSS.

<header>

Phần tử <header> đại diện cho một thùng chứa các nội dung giới thiệu hoặc một tập hợp các liên kết điều hướng trang.

<nav>

Thẻ <nav> là một tập hợp các liên kết điều hướng trang.

<main>

Thẻ <main> chỉ định nội dung chính của tài liệu. Nội dung bên trong phần tử <main> phải là duy nhất cho tài liệu. Không được chứa bất kỳ nội dung nào được lặp lại trên các tài liệu như thanh sidebar , liên kết điều hướng, thông tin bản quyền, logo trang web và biểu mẫu tìm kiếm.

```
<main>
   <h1>Web Browsers</h1>
   Google Chrome, Firefox, and Internet Explorer are the
most used browsers today.
    <article>
       <h1>Google Chrome</h1>
       Google Chrome is a free, open-source web browser
developed by Google, released in 2008.
    </article>
    <article>
       <h1>Internet Explorer</h1>
       Internet Explorer is a free web browser from
Microsoft, released in 1995.
    </article>
    <article>
       <h1>Mozilla Firefox</h1>
       Firefox is a free, open-source web browser from
Mozilla, released in 2004.
    </article>
</main>
```

<section>

Thẻ <section> xác định các phần trong tài liệu, chẳng hạn như chương, tiêu đề, cuối trang hoặc bất kỳ phần nào khác của tài liệu.

```
<section>
     <h1>WWF</h1>
     The World Wide Fund for Nature (WWF) is....
</section>
```

<footer>

Thẻ <footer> xác định phần xuất hiện dưới cùng của trang web. Thẻ <footer> thường chứa thông tin về trang web như:

- Thông tin tác giả
- Thông tin bản quyền
- Thông tin liên lạc
- Sơ đồ trang web
- Quay lại liên kết hàng đầu
- Tài liệu liên quan

<form>

Thẻ <form> được sử dụng để tạo biểu mẫu HTML cho đầu vào của người dùng. Phần tử <form> có thể chứa một hoặc nhiều phần tử biểu mẫu sau:

- <input>
- <textarea>
- <button>
- <select>
- <option>
- <optgroup>

- <fieldset>
- <label>

ul>

Thẻ xác định danh sách không có thứ tự (gạch đầu dòng). Sử dụng thẻ cùng với thẻ để tạo danh sách không theo thứ tự.

<hr>

Thẻ <hr> thể hiện 1 đường kẻ ngắt giữa các thành phần.

<hr>

<

Thẻ xác định một mục danh sách. Thẻ được sử dụng trong danh sách theo thứ tự (), danh sách không có thứ tự () và trong

danh sách menu (<menu>).

```
<a href="" data-text="Home">Home</a>
<a href="" data-text="About">About</a>
```

>

```
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Excepturi, facere!
```


Thẻ là một kiểu bảng trong HTML. Bảng này bao gồm phần tử và một hoặc nhiều phần tử >, và .

< h1 > - < h6 >

Các thẻ <h1> đến <h6> được sử dụng để xác định các tiêu đề HTML. Thẻ <h1> xác định tiêu đề quan trọng nhất. Thẻ <h6> xác định tiêu đề ít quan trọng nhất.

```
<h1 class="display-1">This is h1</h1>
<h2 class="display-2">This is h2</h2>
<h3 class="display-3">This is h3</h3>
<h4 class="display-4">This is h4</h4>
<h5 class="display-5">This is h5</h5>
<h6 class="display-6">This is h6</h6>
```

3.1.3.2. Inline Tag

<a>

Thẻ <a> xác định một siêu liên kết, được sử dụng để liên kết từ trang này sang trang khác. Thuộc tính quan trọng nhất của phần tử <a> là thuộc tính href, cho biết đích đến của liên kết.

```
<a class="nav-link " href="#">TRANG CHŮ</a>
<a class="nav-link" href="#">GIÓI THIỆU</a>
<a class="nav-link" href="#">NGHÀNH NGHỀ CHÍNH</a>
```

Thẻ xác định một đoạn văn bản.

How would you like tobe contact

<button>

Thẻ <button> xác định nút có thể nhấn. Trong phần tử <button> bạn có thể đặt nội dung, như văn bản hoặc hình ảnh. Đây là sự khác biệt giữa phần tử này và các nút được tạo bằng phần tử <input>.

```
<button type="button">Click Me!</button>
<button type="reset">Click Me!</button>
```

<textarea>

Thẻ <textarea> xác định điều khiển nhập văn bản nhiều dòng. Một vùng văn bản có thể chứa số lượng ký tự không giới hạn và văn bản hiển thị ở phông chữ có chiều rộng cố định.

```
<textarea rows="4" cols="50">
At w3schools.com you will learn how to make a website. We
offer free tutorials in all web development technologies.
</textarea>
```


Thẻ được sử dụng để nhóm các thành phần nội tuyến trong tài liệu. Thẻ không cung cấp thay đổi trực quan. Thẻ cung cấp một cách để thêm một mắt xích vào một phần của văn bản hoặc một phần của tài liệu.

```
My mother has <span style="color:blue">blue</span> eyes.
```

<i>>

Thẻ <i> được sử dụng để hiển thị nội dung in nghiêng.

```
<i class="fas fa-gift"></i><i class="fas fa-glass-martini-alt"></i><i class="fas fa-globe-asia"></i><i class="fas fa-graduation-cap"></i></i>
```


Thẻ xác định hình ảnh trong trang HTML. Thẻ có hai thuộc tính bắt buộc: src và alt.

<label>

Thẻ <nhãn> xác định nhãn cho phần tử <button>, <input>, <meter>, <output>, <progress>, <select> hoặc <textarea>. Phần tử <label > không hiển thị bất kỳ điều gì đặc biệt cho người dùng. Tuy nhiên, nó cung cấp một cải tiến về khả năng sử dụng cho người dùng chuột, bởi vì nếu người dùng nhấp vào văn bản trong phần tử <label>, nó sẽ chuyển đổi điều khiển. Thuộc tính for của thẻ

<label> phải bằng thuộc tính id của thành phần liên quan để liên kết chúng lại với nhau.

<label for="select"></label>

<input>

Thẻ <input> chỉ định vùng nhập đầu vào nơi người dùng có thể nhập dữ liệu. Các phần tử <input> được sử dụng trong phần tử <form> để khai báo các điều khiển đầu vào cho phép người dùng nhập dữ liệu. Một trường đầu vào có thể thay đổi theo nhiều cách, tùy thuộc vào loại thuộc tính.

```
<input type="email" placeholder="Email ">
<input type="password" placeholder="Password">
```

<select>

Phần tử <select> được sử dụng để tạo danh sách thả xuống. Các thẻ <option> bên trong phần tử <select> xác định các tùy chọn khả dụng trong danh sách.

<time>

Thẻ <time> xác định ngày / giờ có thể đọc được của con người. Phần tử này cũng có thể được sử dụng để mã hóa ngày và giờ theo cách dễ đọc bằng máy để các tác nhân người dùng có thể cung cấp để thêm lời nhắc sinh nhật hoặc sự kiện theo lịch vào lịch của người dùng và công cụ tìm kiếm có thể tạo ra kết quả tìm kiếm thông minh hơn.

<time datetime="2008-02-14 20:00">Valentinesday</time>

<object>

Thẻ <object> xác định một đối tượng được nhúng trong tài liệu HTML. Sử dụng yếu tố này để nhúng đa phương tiện (như âm thanh, video, tiểu dụng Java, ActiveX, PDF và Flash) trong các trang web của bạn.

<object width="400" height="400"
data="helloworld.swf"></object>

<script>

Thẻ <script> được sử dụng để xác định tập lệnh phía máy khách (JavaScript). Phần tử <script> chứa các câu lệnh script hoặc nó trỏ đến tệp script bên ngoài thông qua thuộc tính src. Các hành động phổ biến với JavaScript là thao tác hình ảnh, xác thực mẫu và thay đổi động của nội dung.

```
<script type="text/javascript " src="./js/jquery-
3.3.1.js"></script>
<script type="text/javascript"
src="./js/owl.carousel.min.js"></script>
```

3.1.4. HTML 5 và sự khác biệt

Từ những ngày đầu tiên, HTML đã là một phát minh vượt thời đại. W3C liên tục tung ra các cập nhật và nâng cấp.

HTML 4 (cũng thường được gọi chung là HTML) được phát hành vào năm 1999, trong khi phiên bản mới nhất, lớn nhất là HTML 5 đã được phát hành vào năm 2014. Phiên bản này đã ra mắt rất nhiều tính năng quan trọng mới.

Một trong số các tính năng hấp dẫn nhất của HTML 5 là khả năng hỗ trợ audio và video. Thay vì sử dụng Flash player, chúng ta có thể chỉ cần nhúng video và audio vào trong trang web bằng cách sử dụng thẻ **<audio>** và **<video>**. Nó cũng hỗ trợ tính năng như scalable vector graphics (SVG), MathML cho các công thức toán học và phương trình khoa học.

HTML 5 có rất nhiều cải thiện về mặt ngôn ngữ, giúp cho trình duyệt biết mục đích chính của nội dung trong Website là gì, việc này có lợi cho cả người xem lẫn các bộ máy tìm kiếm.

3.1.5. Ưu điểm và nhược điểm của HTML

3.1.5.1. Ưu điểm

- Ngôn ngữ được sử dụng rộng rãi ngày nay và có rất nhiều nguồn tài nguyên hỗ trợ và cộng đồng sử dụng cực lớn.
- Sử dụng mượt mà trên hầu hết mọi trình duyệt.
- Có quá trình học đơn giản và trực tiếp.
- Mã nguồn mở và hoàn toàn miễn phí.
- Markup gọn gàng và đồng nhất.
- Chuẩn chính của web được vận hành bởi World Wide Web Consortium (W3C).
- Dễ dàng tích hợp với ngôn ngữ backend như PHP.

3.1.5.2. Khuyết điểm

- Được dùng chủ yếu cho web tĩnh. Đối với các tính ăng động, phải sử dụng thêm Javascript hoặc ngôn ngữ backend như PHP.
- Có thể thực thi một số logic nhất định cho người dùng. Vì vậy, hầu hết các trang đều cần được tạo riêng biệt kể cả header và footer.
- Một số trình duyệt chậm hỗ trợ tính năng mới.
- Khó kiểm soát cách thực thi của trình duyệt (ví dụ những trình duyệt cũ không hiển thị được thẻ mới).

3.2. CSS (Cascading Style Sheets)

3.2.1. Quá trình hình thành và phát triển

CSS là một ngôn ngữ tạo phong cách và định kiểu cho những thành phần được viết dưới dạng ngôn ngữ đánh dấu như HTML. Nó có thể điều khiển định dạng của nhiều trang web cùng lúc để tiết kiệm công sức cho người phát triển web. Nó phân biệt cách hiển thị của trang web với nội dung chính của trang web bằng cách điều khiển bố cục, màu sắc và font chữ.

CSS được phát triển bởi W3C (World Wide Web Consortium) vào năm 1996, vì một lý do đơn giản: HTML không được thiết kết để gắn tag giúp định dạng trang web. Ta chỉ có thể dùng nó để "dánh dấu" văn bản.

Những thẻ như **** được ra mắt trong HTML phiên bản 3.2, nó gây ra nhiều rắc rối cho lập trình viên. Vì Website có thể sử dụng nhiều font khác nhau, màu sắc và phong cách cũng khác nhau. Để viết lại code cho cả trang web là cả một quá trình dài, cực nhọc. Vì vậy, CSS được tạo ra bởi W3C để giải quyết vấn đề này.

Mối tương quan giữa HTML và CSS là rất mật thiết. HTML là ngôn ngữ đánh dấu (nền tảng cảu một Website) và CSS định hình phong cách (tất cả những gì tạo nên giao diện cho trang web), do đó chúng không thể tách rời nhau. CSS về lý thuyết thì sự có mặt của nó là không quan trọng, nhưng khi đó, trang web sẽ chỉ là một trang chứa văn bản mà không có gì khác.

3.2.2. Hoạt động của CSS

CSS sử dụng cấu trúc tiếng Anh đơn giản để tạo ra một bộ các quy tắc có thể tận dụng. Như đã nhắc đến ở trên, HTML không được dùng để tạo phong cách, nó chỉ dùng để tạo cơ sở, và chia bố cục từng phần cho trang web. CSS sẽ đảm nhiệm phần đó. CSS rất đơn giản, cấu trúc gồm 2 phần: phần chọn thẻ và phần khai báo style cho thẻ đó. Phần chọn thẻ sẽ trỏ về thẻ HTML muốn tạo style, phần khai báo style sẽ gồm một hay nhiều dòng khai báo cách nhau bởi dấu chấm phẩy. Mỗi khai báo phải được để trong cập dấu ngoặc nhọn.

```
html, body {
    background-color: #fff;
    color: #636b6f;
    font-family: 'Nunito', sans-serif;
    font-weight: 200;
    height: 100vh;
    margin: 0;
}

.full-height {
    height: 100vh;
}

.flex-center {
    align-items: center;
    display: flex;
    justify-content: center;
}

.position-ref {
    position: relative;
}
```

3.2.3. Tổng quan các thuộc tính hay dùng

3.2.3.1. Màu

sắc Color

Đặt màu sắc cho văn bản

color: #212529;

Background-color

Đặt màu sắc nền cho phần tử

background-color: #fff;

Opacity

Đặt độ trong suốt cho thành phần đó

opacity: 1;

3.2.3.1 Background và Border Background-image

Đặt hình nền cho phần tử

```
background-image: url("data:image/svg+xml,%3csvg
xmlns='http://www.w3.org/2000/svg' viewBox='0 0 8
8'%3e%3cpath fill='%23fff' d='M6.564.75l-3.59 3.612-1.538-
1.55L0 4.26 2.974 7.25 8 2.193z'/%3e%3c/svg%3e");
```

Background-position

Chỉ định vị trí của hình nền

```
background-position: 1rem 0;
```

Background-repeat

Xác định thành phần có được lặp lại không

```
background-repeat: no-repeat;
```

Border

Thiết lập đường viền cho văn bản

```
border: solid transparent;
```

Border-color

Xác định màu cho đường viền

border-color: #28a745;

Boder-width

Thiết lập độ rộng cho cả bốn đường viền

border-width: 1px 0;

Border-radius

Thiết lập đường bo các góc cho thành phần

border-radius: 0.3rem;

Box-shadow

Gắn một hoặc nhiều đổ bóng vào phần tử.

box-shadow: 0 0 0 0.2rem rgba(40, 167, 69, 0.25);

3.2.3.2. Basic

box Display

Xác định loại hiển thị của thành phần\

display: block;

Position

Thuộc tính vị trí xác định loại phương thức định vị được sử dụng cho một phần tử (tĩnh, tương đối, tuyệt đối, cố định hoặc dính).

position: relative;

Margin

Các thuộc tính căn lề CSS được sử dụng để tạo không gian xung quanh các phần tử, bên ngoài bất kỳ đường viền xác định nào.

Với CSS, bạn có toàn quyền kiểm soát việc căn lề. Có các thuộc tính để căn lề cho mỗi bên của một phần tử (trên cùng, bên phải, dưới cùng và bên trái).

```
margin-top: 0;
margin-bottom: 1rem;
```

Padding

Thiết lập vùng đệm cho thành phần

```
padding: 0.2rem 0.4rem;

padding-top: 0.75rem;

padding-bottom: 0.75rem;
```

Visibility

Xác định phần tử có được thấy hay không

visibility: visible!important;

z-index

Xác định vị trí xếp chồng của các thành phần có thuộc tính position.

z-index: 5;

3.2.3.3. Flex-

box Flex

Xác định lại độ dài của các mục hoặc tính tương đối của nó đối với các thành phần khác trong flexbox.

flex: 0 0 auto;

Flex-direction

Xác định hướng sắp đặt của các thành phần.

flex-direction: column;

Justify-content

Xác định sự căn chỉnh của các thành phần bên trong khối với nhau khi các thành phần này không chiếm hết khoảng cách có sẵn.

justify-content: center;

Order

Thiết lập thứ tự hiển thị của thành phần so với các thành phần còn lại.

order: 2;

3.2.3.4. Các thuộc tính

text Text-align

Xác định căn chỉnh nội dung theo chiều ngang

text-align: center;

Line-height

Xác định độ cao của dòng.

line-height: inherit;

Text-indent

Xác định khoảng cách thụt đầu dòng của mỗi đoạn văn bản.

text-indent: -999px;

Text-transform

Xác định các định dạng chữ (in hoa, thường, viết hoa chữ cái đầu).

text-transform: uppercase!important;

Word-spacing

Tăng giảm khoảng cách các từ trong nội dung

word-spacing: normal;

3.2.3.5. Các thuộc tính về

font @font-face

Khai báo tên cùng đường lẫn tới thư mục lưu font chữ

```
@font-face {
  font-family: myFirstFont;
  src: url(sansation_light.woff);
}
```

Font

Đặt tất cả các thuộc tình font trong lời khai báo.

font: 0/0 a;

Font-family

Xác định font chữ cho văn bản

```
font-family: -apple-system, BlinkMacSystemFont, "Segoe UI",
Roboto, "Helvetica Neue", Arial, "Noto Sans", sans-serif,
"Apple Color Emoji", "Segoe UI Emoji", "Segoe UI Symbol",
"Noto Color Emoji";
```

Font size

Xác định kích thước font chữ cho đoạn văn bản

font-size: 1rem;

Font-weight

Xác định độ đậm cho font chữ.

font-weight: 400;

3.2.3.6. Các thuộc tính

Animation @#keyframe

Xác định mã animation và hành động của animation

```
@keyframes progress-bar-stripes {
  from {
    background-position: 1rem 0;
  }
  to {
    background-position: 0 0;
  }
}
```

Animation

Là thuộc tính viết tắt cho các thuộc tính animation

animation: progress-bar-stripes 1s linear infinite;

Animation-name

Xác định tên của @keyframes animation

animation-name: some-animation-name;

Animation-duration

Xác định thời gian để hoàn thành một chu kỳ của animation

animation-duration: 10s, 35s, 230ms;

Animation-delay

Xác định độ trễ của animation

animation-delay: 2s;

3.2.3.7. Các thuộc tính

Transform Perspective

Xác định chiều sâu

perspective: 100px;

Transform

Xác đinh chuyển đổi 2D hoặc 3D lên một thành phần.

transform: translateX(0.75rem);

Transform-duration

Xác định thời gian để chuyển đổi thực hiện xong

transition-duration: 5s

Transfom-delay

Xác định độ trễ của chuyển đổi

transition-delay: 2s;

3.3. Javascript

3.3.1. Tổng quan về Javascript

Javascript là một ngôn ngữ lập trình kịch bản dựa vào đối tượng phát triển có sẵn hoặc tự định nghĩa ra. Javascript được sử dụng rộng rãi trong các ứng dụng web. Javascript được hỗ trợ ở hầu hết các trình duyệt web phổ biến hiện nay như: Google Chome, Mozilla Filrefox, Safari, Opera, ... thậm chí ngay cả trên các thiết bị di động như điện thoại.

Javascript là ngôn ngữ mang lại sự sinh động của Website. Nó khác với HTML (chuyên cho nội dung, bố cục) và CSS (chuyên cho phong cách các thẻ HTML) và khác hẳn với PHP (PHP chạy trên server chứ không chạy dưới máy client).

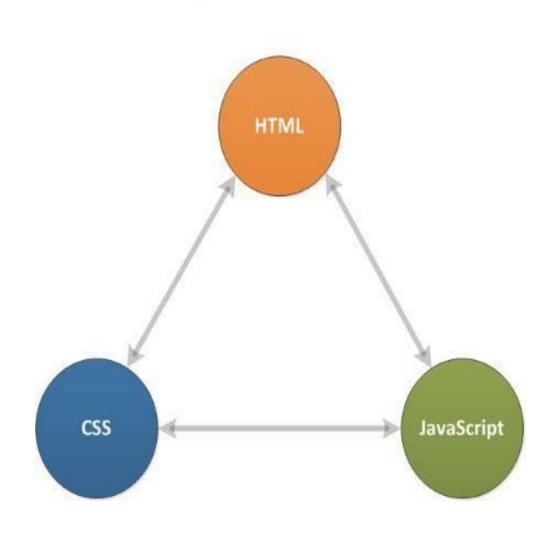
3.3.2. Quá trình phát triển

Javascript được tạo ra trong vòng ba mươi ngày bởi Brandan Eich, một nhân viên của Netscape vào tháng 9 năm 1995. Được đặt tên là Mocha, tên của nó được đổi thành Mona rồi LiveScript trước khi thất sự trở thành Javascript nổi tiếng như bây giờ. Phiên bản đầu tiên của ngôn ngữ này bị giới hạn độc

quyền bởi Netscape và chỉ có các tính năng hạn chế. Nhưng nó tiếp tục phát triển theo thời gian, nhờ một phần vào cộng đồng các lập trình viên đã liên tục làm việc với nó.

Trong năm 1996, Javascript được chính thức đặt tên là ECMAScript. ECMAScript 2 được phát hành vào năm 1998 và ECMAScript 3 tiếp tục ra mắt vào năm 1999. Nó liên tục phát triển thành Javascript ngày nay, giờ đã hoạt động trên khắp mọi trình duyệt và trên khắp các thiết bị từ di động đến máy tính.

Javascript tiếp tục phát triển, chỉ trong 20 năm nó từ một ngôn ngữ lập trình đã trở thành một phần không thể thiếu trong bọ công cụ lập trình web.



Hình 3.2 Ba thành phần không thể thiếu của một trang web

3.3.3. Ưu và nhược điểm

3.3.3.1 Ưu điểm

Javascript có rất nhiều ưu điểm vượt trội so với các đối thủ, đặc biệt trong các trường hợp thực tế:

- Chúng ta không cần một compiler vì trình duyệt web có thể biên dịch nó.
- Nó dễ học hơn các ngôn ngữ khác.
- Lỗi dễ phát hiện và vì thế dễ sửa hơn.
- Nó có thể được gắn trên một số Element của treang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới.
- Javascript hoạt động trên nhiều trình duyệt và nền tảng.
- Giúp Website tương tác tốt hơn với người sử dụng, nâng cao tính thân thiện với người dùng.
- Nhanh và nhẹ hơn các ngôn ngữ khác.

3.3.3.2. Nhược điểm

Mọi ngôn ngữ lập trình đều có các khuyết điểm. Một phần vì ngôn ngữ đó khi phát triển đến một mức độ như Javascript thì sẽ thu hút lượng lớn hacker, scammer, những người có tư tưởng tìm kiếm các lỗ hồng bảo mật để lơi dụng nó.

- Dễ bị khai thác.
- Có thể được dùng để thực thi mã độc trên máy tính của nạn nhân.
- Các phiên bản mới không hỗ trợ các trình duyệt cũ.

- Javascript code snippet lớn.
 - Có thể bị triển khai khác nhau tùy từng thiết bị dẫn tới việc không đồng nhất.

3.3.3. Hoạt động của Javascript

JavaScript thường được nhúng trực tiếp vào một trang web hoặc được tham chiếu qua file **.js** riêng. Nó là ngôn ngữ phía client, tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập.

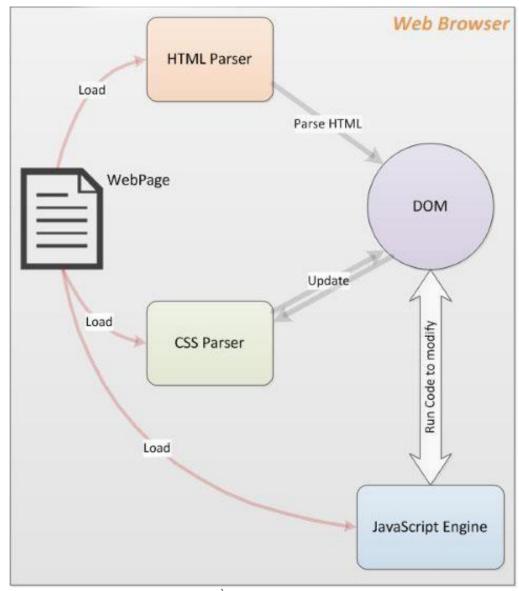
Tải Javascript vào trang web: Cách phổ biến nhất để tải JavaScript trong web page là sử dụng thẻ **<script>**. Tùy thuộc vào nhu cầu, chúng ta có thể sử dụng một trong các phương pháp sau:

- Tải một tệp javascript bên ngoài vào một trang web như sau:
 - <script type="text/javascript" src="/path/to/javascript"></script>
- Nhúng trực tiếp vào HTML:

<script type="text/javascript"> alert ("Page is loaded"); </script>

Khi web browser tải một web page, HTML parser bắt đầu phân tích cú pháp HTML code và tạo DOM. Bất cứ khi nào trình phân tích cú pháp gặp một chỉ thị CSS hoặc JavaScript (inline hoặc externally loaded), nó sẽ được chuyển giao cho CSS parser hoặc JavaScript engine theo yêu cầu. JavaScript engine tải các tệp JavaScript bên ngoài và inline code, nhưng không chạy mã ngay lập tức, mà đợi HTML và CSS phân tích hoàn tất. Khi điều này được thực hiện, JavaScript được thực thi theo thứ tự chúng được tìm thấy trên trang web: các biến và hàm được xác định, các lời gọi hàm được thực thi, trình xử lý sự kiện được kích hoạt, ... Các hoạt

động này dẫn đến việc DOM được cập nhật bởi JavaScript và được hiển thị ngay lập tức bởi trình duyệt.



Hình 3.3 Sơ đồ hoạt động của Javascript

3.3.4. Sự khác biệt giữa Javascript và các ngôn ngữ lập trình khác

Lý do vì sao JavaScript là một trong các ngôn ngữ lập trình phổ biến nhất là nó rất linh hoạt. Trên thực tế, có nhiều lập trình viên chọn nó làm ngôn ngữ chính và chỉ sử dụng các ngôn ngữ khác trong danh sách bên dưới:

JavaScript	JavaScript hoặc JS sẽ giúp tăng tính tương tác trên website. Script này chạy trên các trình duyệt của người dùng thay vì trên server và thường sử dụng thư vuiên của bên thứ 3 nên có thể tăng thêm chức năng cho website mà không phải code từ đầu.
HTML	Viết tắt của "Hypertext Markup Language", HTML là một trong số các ngôn ngữ lập trình phổ biến nhất trên web và xây dựng nên các khối chính của một trang web. Ví dụ về HTML tags là cho đoạn văn và cho hình ảnh.
PHP	PHP là ngôn ngữ phía server, khác với JavaScript chạy trên máy client. Nó thường được sử dụng trong các hệ quản trị nội dung nền PHP như WordPress, nhưng cũng thường được dùng với lập trình back-end và có thể tạo ra kênh truyền thông tin hiệu quả nhất tới và từ database.
CSS	CSS viết tắt của "Cascading Style Sheets", nó giúp webmaster xác định styles và định nghĩa nhiều loại nội dung. Bạn có thể làm vậy thủ công với mọi yếu tố trong HTML, nhưng nếu vậy bạn sẽ cứ lặp đi lặp lại thành phần đó mà bạn dùng ở nhiều nơi khác nhau.

3.3.5. Các thư viện Javascript nổi tiếng hiện nay

Những ứng dụng to lớn của Javascript khiến người ta không thể quên nó được.

Hiện nay có rất nhiều libraries và framework được viết từ Javascript như:

- AngularJS: Một thư viện để xây đựng ứng dụng Web Single Page.
- **NodeJS:** Một thư viện được pát triển phía server dùng để xây dựng ứng dụng thời gian thực.
- **ReactJS:** Thư viện để xây dựng ứng dung Web Single Page.
- React Native: Xây dựng ứng dụng mobie.
- Jquery: Thư viện giúp xây dựng các chức năng bằng Javascript dễ dàng hơn.

3.4. Hướng đối tượng trong lập trình (Object Oriented Programing)

3.4.1. Tổng quan

Lập trình hướng đối tượng (OOP) là một trong những kỹ thuật lập trình rất quan trọng hiện nay. Nó được áp dụng ở hầu hết các ứng dụng thực tế xây dựng tại các doanh nghiệp. Hầu hết các ngôn ngữ lập trình và Framework lập trình phổ biến hiện nay như Java, Javascript, PHP, .NET, Ruby, ... Đều hỗ trợ lập trình hướng đối tượng. Các lập trình viên đa phần đã được học về lập trình hướng đối tượng ở trường đại học.

Lập trình hướng đối tượng là một kỹ thuật lập trình cho phép lập trình viên tạo ra các đối tượng trong code của mình để trừu tượng hóa các đối tượng thực tế

trong cuộc sống. Hướng tiếp cận này hiện đang rất thành công và trở thành một khuôn mẫu trong phát triển phần mềm.

Khi phát triển ứng dụng sử dụng OOP, chúng ta sẽ định nghĩa các lớp (class) để mô hình các đối tượng thực tế. Trong ứng dụng, các lớp này được khởi tạo thành các đối tượng và trong suốt quá trình ứng dụng chạy, các phương thức (method) của các đối tượng này sẽ được gọi.

Lớp định nghĩa đối tượng sẽ bao gồm những thuộc tính và phương thức đặc trưng cho đối tượng đó. Một đối tượng chỉ là một thể hiện của lớp. Các lớp tương tác với nhau qua các public API là các phương thức và thuộc tính public của nó.

3.4.2. 4 đặc tính của lập trình hướng đối tượng

3.4.2.1. Tính đóng gói (Encapsulation)

Tính đóng gói tức là quy tắc yêu cầu trạng thái bên trong của một đối tượng được bảo vệ và tránh truy cập được từ code bên ngoài (tức là code bên ngoài không thể trực tiếp nhìn thấy và thay đổi trạng thái của đối tượng đó). Bất cứ truy cập nào tới trạng thái bên trong này bắt buộc phải thông qua một public API để đảm bảo trạng thái của đối tượng luôn hợp lệ bởi vì các public API chịu trách nhiệm thực hiện kiểm tra tính hợp lệ cũng như trình tự cập nhật trạng thái của đối tượng đó.

Nói chung trạng thái đối tượng không hợp lệ thường do: chưa được kiểm tra tính hợp lệ, các bước thực hiện không đúng trình tự hoặc bị bỏ qua nên trong OOP có một quy tắc quan trọng cần nhớ đó là phải luôn khai báo các trạng thái bên trong của đối tượng là private và chỉ cho truy cập qua các

public/protected

method/property. Khi sử dụng các đối tượng ta không cần biết bên trong nó làm việc như thế nào, ta chỉ cần biết các public API là gì và điều này đảm bảo những gì thay đổi đối tượng sẽ được kiểm tra bởi các quy tắc logic bên trong, tránh đối tượng bị sử dụng không chính xác.

Nguyên lý đóng gói như thế này ở đâu ta cũng có thể bắt gặp ví dụ như thiết kế viên thuốc, chúng ta chỉ biết nó chữa bệnh này, bệnh kia và một số thành phần chính còn cụ thể bên trong nó có những gì thì hoàn toàn không biết.

3.4.2.2. Tính kế thừa (Inheritance)

Khi bắt đầu xây dựng ứng dụng chúng ta sẽ bắt đầu việc thiết kế các lớp, thông thường chúng ta sẽ thấy có trường hợp một số lớp dường như có quan hệ với những lớp khác, chúng có những đặc tính khá giống nhau. VD: 3 lớp AndroidPhone, IPhone, WindowsPhone

Mỗi lớp đều đại diện cho một loại smartphone khác nhau nhưng lại có những thuộc tính giống nhau. Thay vì sao chép những thuộc tính này, sẽ hay hơn nếu ta đặt chúng ở một nơi có thể dùng bởi những lớp khác. Điều này được thực hiện bởi tính kế thừa trong OOP: chúng ta có thể định nghĩa lớp cha – base class (trong trường hợp này là Smartphone) và có những lớp con kế thừa từ nó (derived class), tạo ra một mối quan hệ cha/con.

Bây giờ, các lớp con có thể kế thừa 3 thuộc tính từ lớp cha. Nếu các chức năng của lớp cha đã được định nghĩa đầy đủ thì lập trình viên sẽ không phải làm bất cứ việc gì ở lớp con. Còn nếu một lớp con muốn chức năng khác so với định nghĩa ở lớp cha thì nó có thể ghi đè (override) chức năng đã được định nghĩa trên lớp cha này.

3.4.2.3. Tính đa hình (Polymorphism)

Với đa số lập trình viên thì tính Kế thừa và Đóng gói trong OOP khá dễ hiểu còn tính Đa hình khi mới tiếp cận sẽ thấy khó hiểu hơn một chút. Tuy nhiên đây lại là một tính chất có thể nói là chứa đựng hầu hết sức mạnh của lập trình hướng đối tượng. Hiểu một cách đơn giản: Đa hình là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

Ví dụ như ở phần trên, mỗi một smartphone kế thừa từ lớp Smartphone nhưng có thể lưu trữ dữ liệu trên cloud theo những cách khác nhau:

AndroidPhone lưu trữ bằng Google Drive Iphone lưu trên iCloud WindowsPhone sử dụng SkyDrive.

Bởi vì tất cả đều là Smartphone nên nếu ta viết một hàm dùng kiểu Smartphone làm tham số thì khi gọi hàm ta có thể truyền vào một đối tượng kiểu AndroidPhone, Iphone hoặc WindowsPhone bởi vì chúng đều kế thừa từ lớp Smartphone nên được chấp nhận (hiểu nôm na một AndroidPhone, Iphone, WindowsPhone cũng là một Smartphone). Bên cạnh đó hàm này thậm chí không cần quan tâm smartphone nào được truyền vào do nó chỉ cần biết đối tượng đang

xử lý ở đây là Smartphone với những public method/property đã được định nghĩa. Nếu các lớp con không định nghĩa lại (override) phương thức CloudStore() thì phương thức CloudStore() trên lớp cha (Smartphone) sẽ được gọi. Còn nếu lớp con override lại phương thức CloudStore() của lớp cha như ở hình trên thì phương thức CloudStore() trên lớp con sẽ được gọi mặc dù code trong hàm đang thao tác với đối tượng kiểu Smartphone.

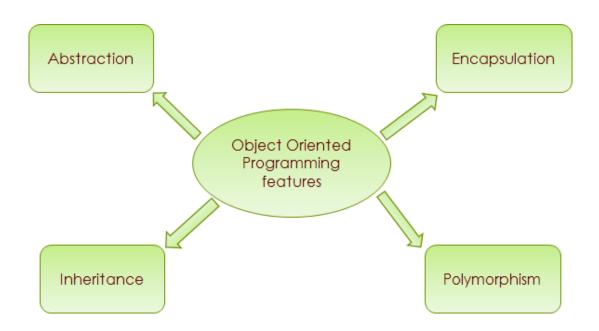
Tính Đa hình như trên là một tính chất rất mạnh mẽ bởi vì nó mang lại cho code khả năng tổng quát hóa cao. Chúng ta không cần tạo ra phương thức cho mỗi kiểu kế thừa từ lớp cha Smartphone mà chỉ cần nhận một biến kiểu Smartphone và có thể làm việc với bất cứ lớp nào kế thừa từ nó. Điều duy nhất không làm được ở đây là sử dụng những phương thức mà chỉ được khai báo trên các lớp con. VD: nếu ta có một phương thức trên lớp IPhone gọi là OpenSiri() nhưng không được khai báo trên lớp Smartphone, khi đó muốn gọi nó sẽ bắt buộc phải ép kiểu từ Smartphone sang IPhone trước khi gọi.

3.4.2.4. Tính trừu tượng (Abstraction)

Tính trừu tượng là một tiến trình ẩn các chi tiết trình triển khai và chỉ hiển thị tính năng tới người dùng. Tính trừu tượng cho phép chúng ta loại bỏ tính chất phức tạp của đối tượng bằng cách chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng trong lập trình.

Tính trừu tượng giúp chúng ta tập trung vào những cốt lõi cần thiết của đối tượng thay vì quan tâm đến cách nó thực hiện.

Chúng là sử dụng các **abstract class** và **abstract interface** để có tính trừu tượng.



Hình 3.4 4 đặc điểm của lập trình hướng đối tượng

CHƯƠNG 4: PHÂN TÍCH NỘI DUNG CỦA GAME

4.1. Tổng quan game

• Giới thiệu game:

Game xếp nhà là một game được thiết kế chạy trên nền tảng trang web.

Game hiển thị tốt nhất trên màn hình có độ phân giải từ 1024 x 768 trở lên (laptop, PC).

• Mô tả game:

Game được thiết kết với giao diện đơn giản. dễ nhìn và dễ thao tác. Người chơi sẽ sử dụng kết hợp chuột để có thể điều khiển và sắp xếp các tầng nhà. Để có được những tòa nhà cao tầng vững chắc bạn hãy cố gắng xếp chồng những khối nhà lên nhau sao cho thật ăn khớp nhé!

Ngoài ra, trên mà hình game cũng sẽ hiển thị số lượng tầng nhà mà người chơi xây dựng được và số điểm tương ứng tầng tòa nhà. Hình mặt người biểu diễn cho sinh mạng duy trì trong 1 round (vòng). Nếu bạn cho để rớt quá với sinh mạng bạn trò chơi sẽ dừng lại và đồng thời hiện ra số điểm cao nhất bạn đạt được và nút chơi lại. Trong khi chơi các tầng nhà mới sẽ xuất hiện liên tục và nhanh dần mỗi khi bạn xếp 1 tầng nhà thành công. Nếu bạn xếp đúng vị trí tầng nhà ở chính giữa, tầng nhà đó sẽ đổi màu bạn sẽ được gấp đôi số điểm tầng nhà đó và ngôi nhà của bạn sẽ không bị lung lay. Ngược lại, bạn xếp không đúng vị trí chính giữa ngôi sẽ bị lung lay và khiếu bạn dễ chết hơn.

4.2. Giao diện của game

Giao diện mở đầu



Hình 4.1 Giao diện mở đầu.

Giao diện khi chơi game và hướng dẫn cách chơi



Hình 4.2 Giao diện khi chơi game và hướng dẫn cách chơi



Hình 4.3 Giao diện khi chơi game

Các thành phần trong game



Hình 4.4 Các thành phần trong game

Giao diện game over và đưa ra thong tin người chơi



Hình 4.5 Giao diện game over

CHƯƠNG 5: KẾT LUẬN

Việc ứng dụng công nghệ vào đời sống và giải trí đang trở thành ngày một thiết thực và đem lại nhiều giá trị to lớn. Bởi vậy, việc phát triển các ứng dụng giải trí như game hiện nay là một công việc có lợi cho sự giải trí của con người và nó đi cùng với thời đại. Tin học là ngành công nghệ mũi nhọn lớn thuận tiện cho mỗi quốc gia. Việc ứng dụng của những sản phẩm tin học đã và đang góp phần phát triển các công nghệ trở nên hiện đại hơn.

Việc xây dựng trò chơi xếp nhà giúp cho người dùng có thể giải trí sau những giờ học tập và làm việc căng thẳng, đơn giản chỉ cần bật trình duyệt web lên và chiến!

Trong quá trình thực hiện đề tài, nhóm em đã thực hiện những việc sau:

- Nghiên cứu các công nghệ để phát triển và xây dựng giao diện người dùng là HTML, CSS, Javascript.
- Thiết kế bố cục, hình ảnh và âm thanh cho game.
- Phân tích và áp dụng hướng đối tượng vào game giúp cấu trúc và code trở nên chuyên nghiệp và dễ bảo trì hơn.
- Sau khi hoàn thành đồ án thực tập tốt nghiệp này nhóm em đã có thêm nhiều kinh nghiệm trong việc phân tích thiết kế và phát triển một ứng dụng web cụ thể ở đây là game xếp nhà và kinh nghiệm tối ưu hóa code để nâng cao tốc độ và hiệu suất cho ứng dụng.

CHƯƠNG 6: HƯỚNG PHÁT TRIỂN

Qua quá trình nghiên cứu đề tài, nhóm em thấy được tầm quan trọng của việc phân tích ứng dụng và áp dụng được các kỹ thuật, công nghệ đang là xu hướng hiện nay vào dự án. Tuy nhiên, đây mới chỉ là một vài chức năng cơ bản của ứng dụng.

Trong tương lai, nhóm em có nhiều thời gian hơn và sẽ củng cố kiến thức, nắm rõ hơn các kỹ thuật, và nhóm em sẽ:

- Tối ưu hóa ứng dụng giúp tải ứng dụng nhanh hơn
- Xây dựng thêm các màn chơi, lối chơi và giao diện đẹp mắt hơn mang lại trải nghiệm tốt cho người dùng.

CHƯƠNG 7: TÀI LIỆU THAM KHẢO

1. Website: https://developer.mozilla.org/en-US/docs/Web

2. Website: https://www.w3schools.com/html/

3. Website: https://www.w3schools.com/css/

4. Website: https://www.w3schools.com/js/

5. Website: https://getbootstrap.com/

 The Fundamental Concepts of Object - Oriented Programming, Kindle Edition