BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN CƠ SỞ

ĐỀTÀI: THIẾT KẾ VÀ TỐI ƯU HÓA GIAO DIỆN WEBSITE NGƯỜI DÙNG FRONT-END

Ngành: **CÔNG NGHỆ THÔNG TIN**
Chuyên ngành: **CÔNG NGHỆ PHẦN MỀM**

Giảng viên hướng dẫn : **ThS. NguyễnĐình Ánh**
Sinh viên thực hiện:**Đặng Thế Lợi, Lai Văn Phú**
Mã số sinh viên: **1411061273, 1411061353**
Lớp: **14DTH10**

TP. Hồ Chí Minh, 2017

MỤC LỤC

Danh mục hình ảnh:

LỜI MỞ ĐẦU

Em xin chân thành cảm ơn Ths. Nguyễn Đình Ánh đã trực tiếp tận tình hướng dẫn, tin tưởng, tạo cơ hội cho em tiếp xúc thực tế và biết được các bước quy trình thực hiện để đưa ra mẫu website của riêng nhóm em.

Với lần đầu tiên nghiên cứu và thực hiện đề tài, em không tránh khỏi những sai sót trong quá trình tìm hiểu và thực hiện, em mong thầy đọc qua báo cáo và góp ý để giúp em có thêm kinh nghiệm trong học tập.

Một lần nữa, em xin chân thành cảm ơn và mong nhận được sự góp ý, chỉ bảo của thầy.

Sinh viên thực hiện

Đặng Thế Lợi , Lai Văn Phú

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI VÀ HƯỚNG NGHIÊN CỨU

1.1.Giới thiệu về đề tài:

“Thiết kế và tối ưu hóa giao diện website người dùng Front-end”

Bí mật thành công của những website phục vụ kinh doanh là làm thế nào để lấy thiện cảm người dùng ngay từ lần tiếp xúc đầu tiên, sau đó là giữ chân và thiết phục họ mua hàng.Việc thiết kế và tối ưu hóa giao diện web để giúp người dùng nhanh chóng nắm bắt được thông tin mà người dùng muốn tìm kiếm.

Việc kéo dài thời gian người dùng lưu lại trên website tùy thuộc vào nhiều yếu tố khách quan và chủ quan, nhưng có một số yếu tố có thể tác động đến nhiều đối tượng, đó là những trải nghiệm mà người dùng nhận được khi họ đang ở trên website.

Phần front-end của một trang web là phần tương tác với người dùng. Tất cả mọi thứ người dùng nhìn thấy khi điều hướng trên Internet, từ các font chữ, màu sắc cho tới các menu xổ xuống và các thanh trượt, là một sự kết hợp của HTML, CSS, và JavaScript được điều khiển bởi trình duyệt máy tính.

Vai trò của front-end trong 1 dự án là khá quan trọng, vì giao diện là thứ đập vào mắt người dùng đầu tiên. Front-end developer không chỉ thiết kế giao diện đẹp, mà còn phải rõ ràng, dễ sử dụng. Người dùng có thể làm việc mình muốn một cách đơn giản, nhanh gọn.

1.2.Khảo sát giao diện web trắc nghiệm mẫu:

Web mẫu: tungtung.vn

Hướng đề tài của chúng em là thiết kế giao diện web thi trắc nghiệm. Làm giao diện đẹp mắt và thân thiện ,tạo thiện cảm cho người dùng

Các trang chính được thiết kế bao gồm:

- Trang chủ
- Trang đăng nhập
- Trang đăng kí
- Trang đề thi
- Trang tạo đề thi
- Trang trắc nghiệm
- Trang kết quả

CHƯƠNG 2:CƠ SỞ LÝ THUYẾT

2.1.HTML

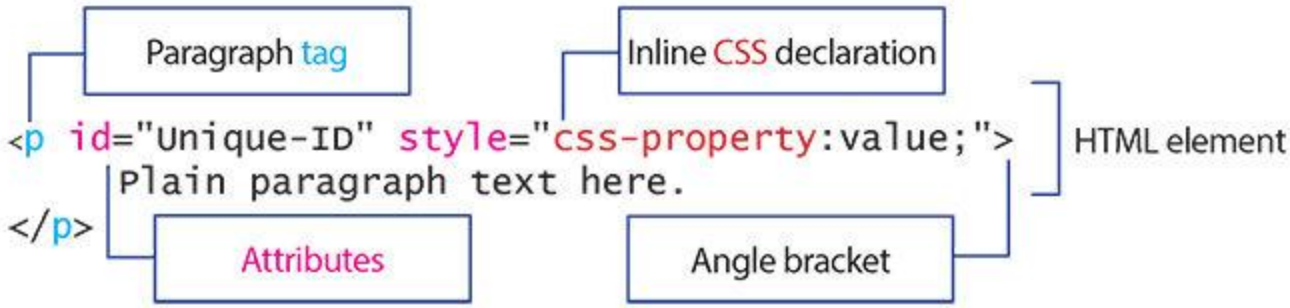
HTML là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web với các mẩu thông tin được trình bày trên World Wide Web. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho World Wide Web.

2.1.1.Tại sao bạn cần nó?

HTML là nơi mà ngữ nghĩa nội dung của bạn được đặt vào. Điều này quan trọng với những thiết bị đọc máy như spider search engine (robot của máy tìm kiếm) và screen reader (ứng dụng đọc màn hình). Qua thời gian, sự liên kết của sự phân biệt giữa đâu là ngữ nghĩa và đâu là cấu trúc càng được thắt chặt. Phiên bản mới nhất của HTML (5) giới thiệu những tag mới như <article>, <aside>, <nav>, <header> và <footer> với mục đích làm rõ ràng ngữ nghĩa và cấu trúc. Điều này mang lại lợi ích không chỉ cho con người mà còn cho các cỗ máy.

2.1.2Cấu trúc của một thành phần HTML

Những thành phần HTML, ở mức độ tối thiểu, là những cặp của những tag mở và đóng, mỗi tag được đóng trong một cặp dấu ngoặc nhọn "<" và ">", như thẻ "paragraph" chẳng hạn, ở hình dưới nó được đánh dấu bằng màu xanh lục lam. Những thành phần có thể được nhận những thuộc tính, ở hình dưới là màu hồng, như một "class", thứ có thể biến những thành phần thành thành viên của một nhóm mà HTML và JS có thể tác động. Thuộc tính style, nội dung của nó được biểu thị bằng màu đỏ ở dưới, là một cách để tạo một quy tắc CSS chỉ tác dộng lên thành phần đó.



Hình 2.1.2.Cấu trúc html

2.2. CSS

CSS cho phép nội dung ngữ nghĩa và những thuộc tính trình chiếu bên ngoài của văn bản được chia ra làm 2 phần riêng biệt, làm những đặc điểm định kiểu như bố cục, màu sắc, và kiểu chữ linh động và dễ áp dụng vào những văn bản khác

nhau. Khi nội dung và thiết kế bên ngoài được chia ra riêng biệt, nhà phát triển có nhiều sự linh động và nhất quán hơn trong thiết kế trực quan.

2.2.1.Tại sao cần đến nó?

Những website không được thiết kế sẽ trông tồi tệ và thiếu thu hút. Trong khi có thể chúng có thể đọc được, CSS là đặt nền móng cho việc phân tầng thông tin trực quan nhờ vào bố cục được nó tạo ra. Lấy ví dụ, dưới đây là hình ảnh của thanh "navigation" trên một trang web không được áp dụng CSS.

Hình 2.2.1.Thanh navigation

Hãy chú ý đến kiểu chữ và màu sắc, menu chưa được định dạng có phương thẳng đứng vì đó là kiểu mặc định của trình duyệt .Thêm vào đó, với sự xuất hiện của những sản phẩm có kích cỡ khác nhau và được kết nối với nhau như iPhone, tablet,v.v, một trong những kỹ năng quan trọng nhất đã trở thành "Responsive Design - thiết kế có tính phản hồi", hoặc website có thể tùy chỉnh tùy vào kích cỡ màn hình. Tất cả chúng đều có thể thực hiện thông qua CSS.

2.2.2.Cấu trúc của một quy tắc CSS

Những quy tắc CSS được viết trong 1 trong 3 nơi sau:

a) Trong cùng dòng với một thành phần,

b) Bằng cách tạo ra một section <style> bên trong thẻ <head> của một văn bản HTML,

c) Trong một file style sheet riêng, ví dụ như : style.css.

2.3.Javascript

JavaScript là một ngôn ngữ lập trình của HTML và WEB. Nó là nhẹ và được sử dụng phổ biến nhất như là một phần của các trang web, mà sự thi hành của chúng cho phép Client-Side script tương tác với người sử dụng và tạo các trang web động. Nó là một ngôn ngữ chương trình thông dịch với các khả năng hướng đối tượng.

Những ngôn ngữ lập trình thường được phân loại bằng độ trừu tượng trong ngữ nghĩa, những ngôn ngữ tiền bối, mô hình lập trình, những quy tắc của lệnh. JavaScript được phân loại ở mức đơn giản vì nó đã được mở rộng sang quá nhiều framework-thư viện các lớp đã được xây dựng hoàn chỉnh, để phù hợp với quá nhiều mục đích. Nó là một ngôn ngữ linh động, có cấu trúc tương tự như C, đa mô hình, sặc sỡ màu sắc như một chú tắc kè với những dòng lệnh. Nó là một ví dụ tuyệt vời của những ngôn ngữ được dùng cho nhiều mục đích, hoặc là một ví dụ nghèo nàn của nhiều loại ngôn ngữ khác nhau.

2.3.1.Tại sao cần đến nó?

Ngôn ngữ này có những đối thủ xếp trên hoặc xếp dưới, cụ thể là về độ thích hợp đối với những người mới bắt đầu. JavaScript có thể là ngôn ngữ lập trình phổ biến nhất hiện tại. Trong khi nó không mang lại nguồn tri thức để hiểu hết về thế giới của "code .JS đơn thuần không thể tiến quá xa – nó đạt đến ngày hôm này là nhờ những framework.

2.4.Bootstrap

Bootstrap là một framework CSS được Twitter phát triển. Nó là một tập hợp các bộ chọn, thuộc tính và giá trị có sẵn để giúp web designer tránh việc lặp đi lặp lại trong quá trình tạo ra các class CSS và những đoạn mã HTML giống nhau trong dự án web của mình.

2.4.1.Cấu trúc và tính thực tiễn

Bootstrap chắc chắn là các khuôn khổ phổ biến nhất và sử dụng rộng rãi, ngày nay. Đó là một bộ thiết kế web đẹp, trực quan và mạnh mẽ để tạo qua trình duyệt, giao diện tìm kiếm phù hợp và tốt. Nó cung cấp rất nhiều các thành phần giao diện người dùng phổ biến với một phong cách đồng bằng, một hệ thống lưới điện và plugins JavaScript cho các kịch bản phổ biến.

Nó được xây dựng với LESS và bao gồm bốn phần chính:

- Scaffolding : Phong cách toàn cầu, đáp ứng lưới 12 cột và bố cục. Ghi nhớ rằng Bootstrap không bao gồm các tính năng đáp ứng theo mặc định. Nếu thiết kế của bạn cần phải được đáp ứng, bạn phải kích hoạt chức năng này bằng tay.

- Base CSS : Điều này bao gồm các phần tử HTML cơ bản như bảng, biểu mẫu, các nút, và hình ảnh, phong cách và tăng cường với các lớp học mở rộng.
- Components : Bộ sưu tập của các thành phần tái sử dụng như dropdowns, button groups, navigation controls (tabs, pills, lists, breadcrumbs, pagination),thumbnails, progress bars, media objects, và nhiều hơn nữa.

- JavaScript : JQuery plugin mà đưa những thành phần trên vào cuộc sống, plus transitions, modals, tool tips, popovers, scrollspy (cho tự động cập nhật các mục tiêu nav dựa trên vị trí di chuyển), carousel, typeahead (một thư viện tự động hoàn chỉnh nhanh chóng và đầy đủ tính năng), affix navigation, và nhiều hơn nữa.

Bootstrap là đã đủ mạnh để trao quyền cho bất kỳ giao diện web. Nhưng để sử dụng nhiều hơn của nó và làm cho quá trình phát triển dễ dàng hơn, bạn có thể tìm thấy rất nhiều công cụ và nguồn lực bổ sung cho nó. Một số trong số họ được liệt kê dưới đây:

- jQuery UI Bootstrap - một nguồn tài nguyên tuyệt vời cho jQuery và người hâm mộ Bootstrap kết hợp sức mạnh của cả hai. Nó mang đến cho độc đáo các slickness của Bootstrap để jQuery UI widget.
- jQuery Mobile Bootstrap Theme - tương tự với chủ đề jQuery UI trên, đây là một chủ đề được xây dựng cho jQuery Mobile. Nó là một nguồn tài nguyên hữu ích nếu bạn có một trang web front-end được xây dựng với Bootstrap và muốn cung cấp một cái nhìn tương tự cho điện thoại di động.
- Fuel UX - điều này kéo dài Bootstrap có thêm điều khiển JavaScript nhẹ. Thật dễ dàng để cài đặt, tùy chỉnh, cập nhật và tối ưu hóa.
- StyleBootstrap.info - Bootstrap có Customizer riêng của mình nhưng StyleBootstrap là một trong những chi tiết hơn với các bộ chọn màu sắc và khả năng để tạo kiểu cho mỗi thành phần khác nhau
- BootSwatchr - một chủ đề con lăn Bootstrap cho thấy kết quả ngay lập tức thay đổi của bạn. Đối với mọi phong cách được tạo ra, các ứng dụng tạo ra một URL duy nhất trong trường hợp bạn muốn chia sẻ nó với những người khác hoặc trả lại và chỉnh sửa bất cứ lúc nào sau đó.
- Bootswatch - một bộ tốt đẹp của chủ đề miễn phí cho Bootstrap.
- Bootsnipp - một bộ sưu tập tốt của các yếu tố thiết kế và các đoạn mã HTML cho
- Bootstrap. Nó cũng cung cấp hình thành và xây dựng nút.
- LayoutIt - kéo và thả giao diện người xây dựng dựa trên các yếu tố và các thành phần của Bootstrap. Nó giúp bạn soạn thiết kế trực quan của bạn bằng cách đặt và sắp xếp các yếu tố khác nhau vào bố cục của bạn bằng

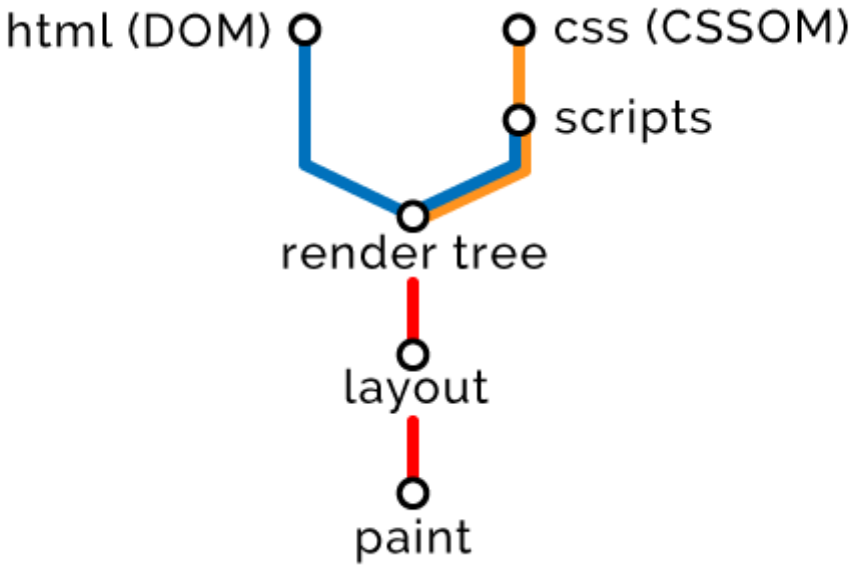
cách kéo và thả và sau đó cho phép bạn chỉnh sửa các thuộc tính của họ. Bạn nhận được các mã cơ sở và sau đó mở rộng nó. Đơn giản và dễ dàng.

2.5.Tối ưu hóa:

Tối ưu hóa tức là tập trung vào cải thiện tốc độ và mức độ hài lòng.Việc tối ưu này sẽ khiến cả user, developer happy hơn và đồng thời cải thiện SEO ranking của của trang web.

Khi browser nhận được file HTML, nó sẽ chạy qua một chuỗi các bước để xây dựng và hiển thị trang web.

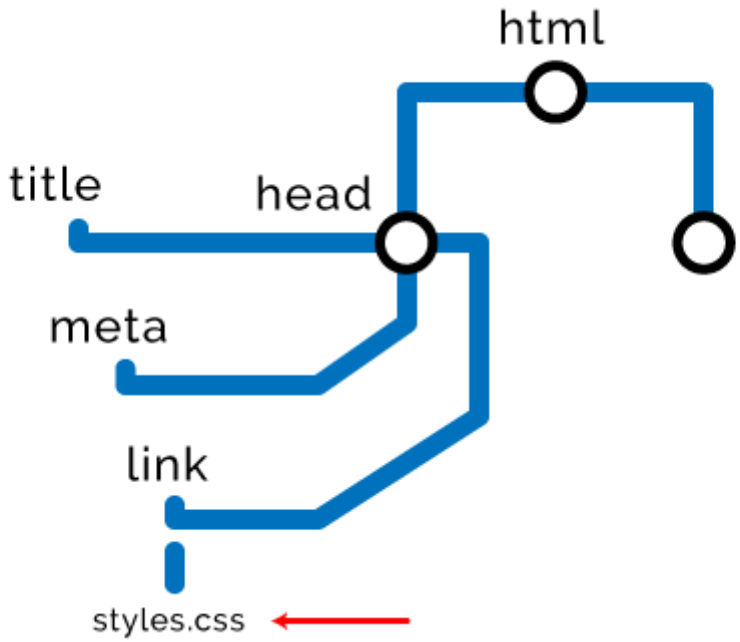
1. Sử dụng file HTML để tạo DOM (Document Object Model)
2. Sử dụng CSS để tạo CSSOM (CSS Object Model)
3. Chạy script xử lý DOM và CSSOM đã có
4. Kết hợp DOM và CSSOM để tạo thành Render Tree
5. Sử dụng Render Tree để Layout (xác định size và position của toàn bộ phần tử trên trang web)



Hình 2.5.Chuỗi bước xây dựng web

2.5.1.Html

Trình duyệt đọc file HTML từ trên xuống dưới và sử dụng nó để tạo ra các node từ file HTML, từ đó tạo thành cây DOM.



Hình 2.5.1. DOM

Chiến lược tối ưu hóa HTML:

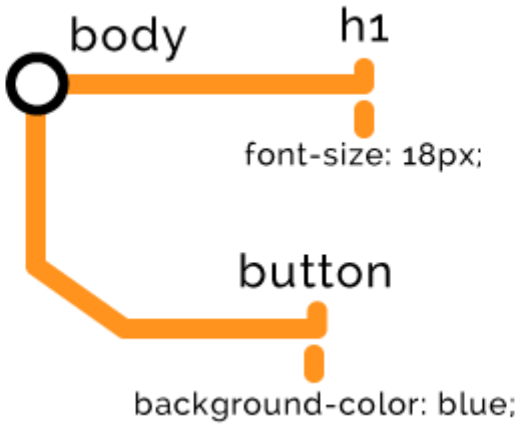
- Style đặt ở trên và script đặt ở cuối file HTML
  Chúng ta nên load style càng sớm càng tốt và load script càng muộn càng tốt. Lý do là bởi script có thể cần HTML và CSS kết thúc việc parse trước khi chúng chạy.
- Rút gọn và nén
  Việc làm này được áp dụng với mọi loại tài nguyên mà trình duyệt cần tải về như HTML, CSS, JavaScript, ảnh và những những loại file khác.

Rút gọn sẽ xóa bất cứ ký tự thừa nào bao gồm khoảng trắng, comments, dấu chấm phẩy thừa,... Nén (chẳng hạn GZip), thay thế những dữ liệu trong code hoặc nội dung được lặp đi lặp lại bằng con trỏ đến đối tượng ban đầu. Cố gắng nén kích thước file cần download và dựa vào phía client để giải nén chúng. Thông qua việc thực hiện những công việc trên, kích thước của file cần tải về có thể giảm 80 đến 90%.

- Khả năng truy cập
  Việc làm này không làm tăng tốc độ tải trang nhưng sẽ giúp người dùng hài lòng hơn. Hãy sử dụng "aria" cho các phần tử của trang, và "alt" text cho ảnh.

2.5.2. Css:

Những node CSSOM được tạo ra tương tự những node DOM.



Hình 2.5.2.CSSOM

Trình duyệt chỉ tạo Render Tree khi DOM và CSSOM đã được tạo hoàn chỉnh. Vì việc tạo CSSOM chặn quá trình render trang web do vậy những node liên quan đến style dùng để tạo CSSOM được gọi là "Render Blocking". Chúng ta nên làm giảm kích thước style, và load chúng càng sớm càng tốt hoặc trì hoãn việc tải chúng nếu được.

Chiến lược tối ưu hóa CSS:

- Sử dụng media attribute
  Media attribute quy định điều kiện cần thỏa mãn để xem có load các file style hay không.

  Ví dụ, mobile thường không có khả năng xử lý mạnh mẽ như desktop cho nên để tối giản hóa những style cần xử lý, chúng ta có thể load file CSS cho mobile trước sau đó thêm điều kiện cần thỏa mãn để load style cho desktop. Các file CSS cho desktop sẽ vẫn được tải xuống nhưng chúng sẽ không chặn việc render trang web trên mobile.

- Trì hoãn tải CSS
  Nếu ta có những style có thể được trì hoãn việc tải về và sử dụng để những phần quan trọng của của trang hiển thị trước ví dụ những style cho phần

below the fold, chúng ta có thể dùng script để đợi trang load xong, sau đó mới sử dụng những style này.

- Hạn chế dùng selector quá cụ thể
  Nhược điểm có thể thấy rõ, dữ liệu tải về nhiều hơn, nhiều selector chuỗi với nhau hơn và làm cho file CSS lớn hơn ngoài ra cũng tăng độ phức tạp cho việc tính toán style ở client.

- Chỉ tải về những gì bạn cần

2.5.3.Javascript:

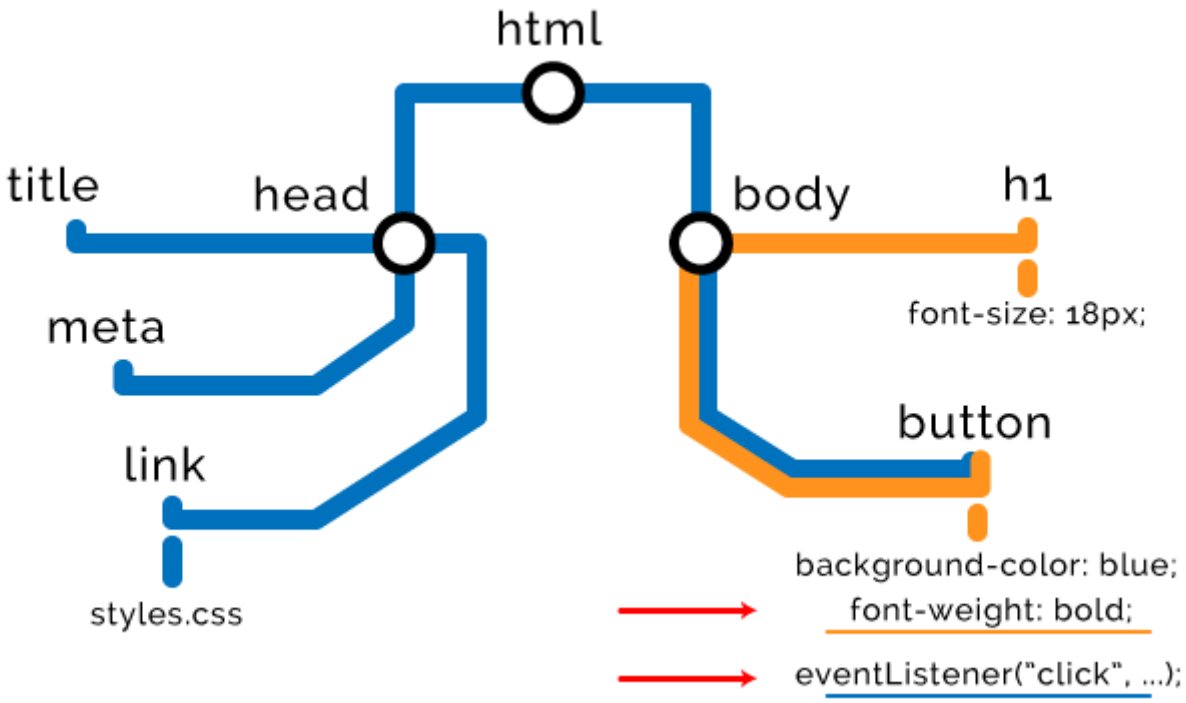
Việc parse file HTML và tạo DOM/CSSOM node sẽ bị dừng lại khi trình duyệt tìm thấy những javascript node, external hoặc internal, cho nên JavaScript được gọi là "Parser blocking". Khi ta có nhiều external script, trình duyệt tải về và chạy từng script một, sau đó tiếp tục tải về và chạy script khác theo thứ tự trước sau.

Vì script có thể cần đến HTML hoặc style để xử lý do vậy chúng ta nên chạy script khi những phần HTML, style đó đã sẵn sàng.

Giả sử ta có đoạn script

```
var button = document.querySelector("button");
button.style.fontWeight = "bold";
button.addEventListener("click", function () {
  alert("Well done.");
});
```

Nó sẽ ảnh hưởng đến DOM và CSSOM như sau:



Hình 2.5.3: Sơ đồ ảnh hưởng

Trình duyệt ngày nay đều hỗ trợ "Preload Scanner". Khi tìm thấy node script, nó sẽ scan toàn bộ những tài nguyên cần tải còn lại trên trang web (ảnh, file js khác,...) ở background trong thời gian parser bị block bởi script đầu tiên đó. Nhờ đó khi parse đến những tài nguyên này thì chúng đã được tải về sẵn và không tốn thời gian chờ đợi việc đó nữa qua đó đẩy nhanh tốc độ.

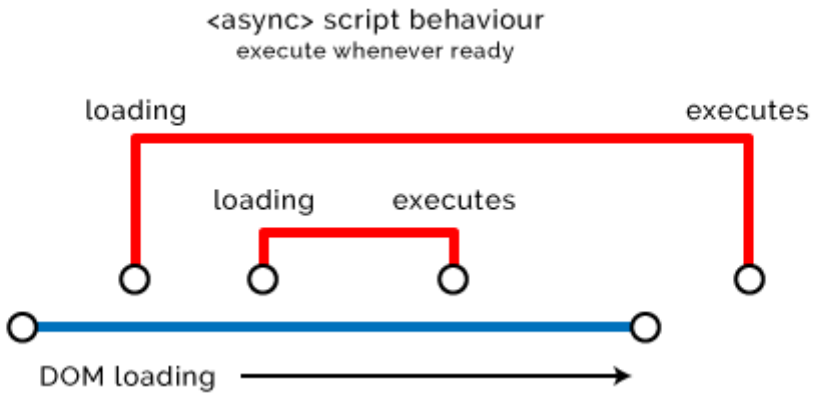
Chiến lược tối ưu hóa Javascript

- Tải script về không đồng bộ
  Bằng việc thêm attribute "async" vào thẻ script, chúng ta yêu cầu trình duyệt tải file JS đó trên thread khác và tiếp tục parse trang.

```
<script src="async-script.js" async></script>
```

Khi việc tải về file .js đó hoàn tất, nó sẽ được chạy. Điều này cũng có nghĩa nó sẽ chạy ở bất kỳ thời điểm nào và có 2 vấn đề có thể xảy ra. Thứ nhất, nó có bắt đầu chạy sau khi trang đã load xong từ rất lâu vì vậy nếu phần UX nào phụ thuộc vào nó, có thể đem lại trải nghiệm không như mong muốn cho người dùng. Thứ hai, nếu nó được chạy trước khi trang web load xong, ta không thể biết được phần

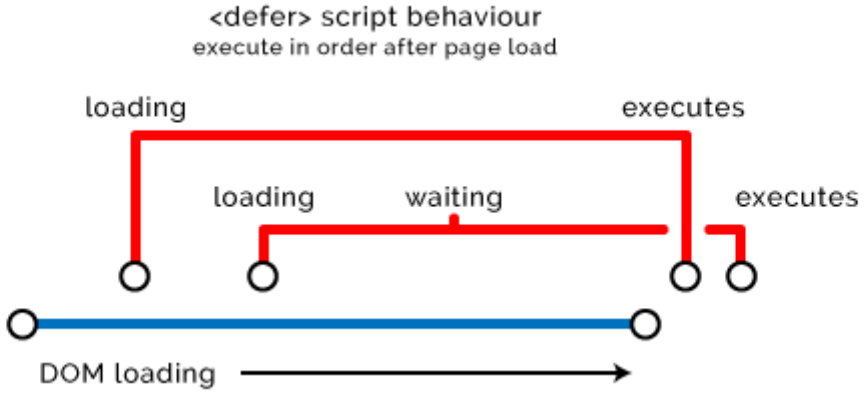
DOM/CSSOM nó cần duyệt đã có sẵn hay chưa.



Từ đây ta có thể thấy rằng async sẽ trở nên hữu ích với những script không tác động gì đến DOM/CSSOM đặc biệt là những script bên ngoài không liên quan đến code của ta và không ảnh hưởng gì đến UX, ví dụ analytics hoặc tracking.

- Trì hoãn tải về script
  "defer" cũng tương tự như "async" khi script không chặn việc load trang, tuy nhiên nó sẽ chỉ chạy khi HTML parse xong và chạy theo thứ tự.

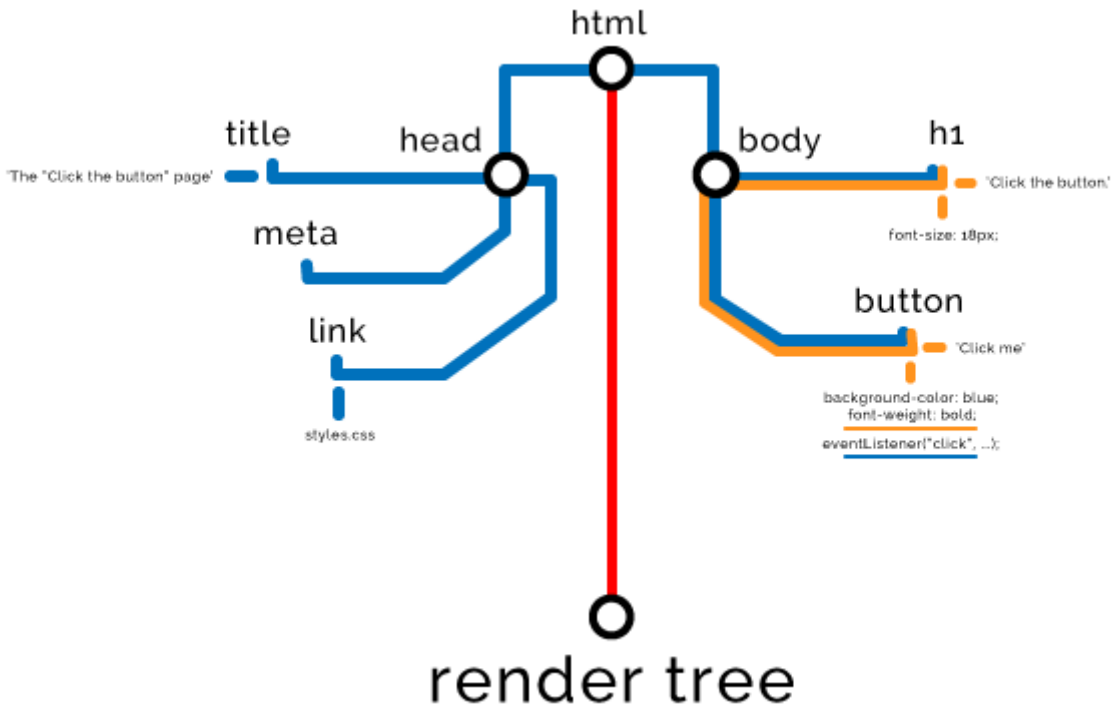
```
<script src="defer-script.js" defer></script>
```



Nó sẽ hữu ích với những script có tác động đến Render Tree, nhưng không liên quan gì đến việc hiển thị phần above the fold của trang web hoặc cần những script khác chạy trước nó.

2.5.4.Render Tree

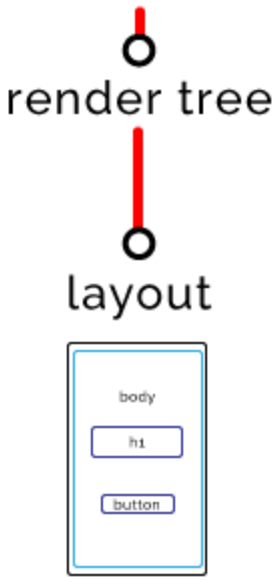
Khi tất cả các node đã được đọc, DOM và CSSOM đã sẵn sàng, trình duyệt sẽ ghép chúng lại để dựng Render Tree.



Hình 2.5.4.Render Tree

2.5.5.Layout

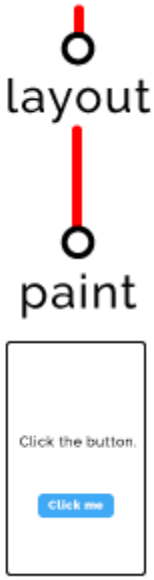
Ở bước Layout, trình duyệt xác định size và position của tất cả các phần tử trên trang web.



Hình 2.5.5.Layout

2.5.6.Paint

Giai đoạn cuối cùng là Paint, hiển thị trang trên màn hình.



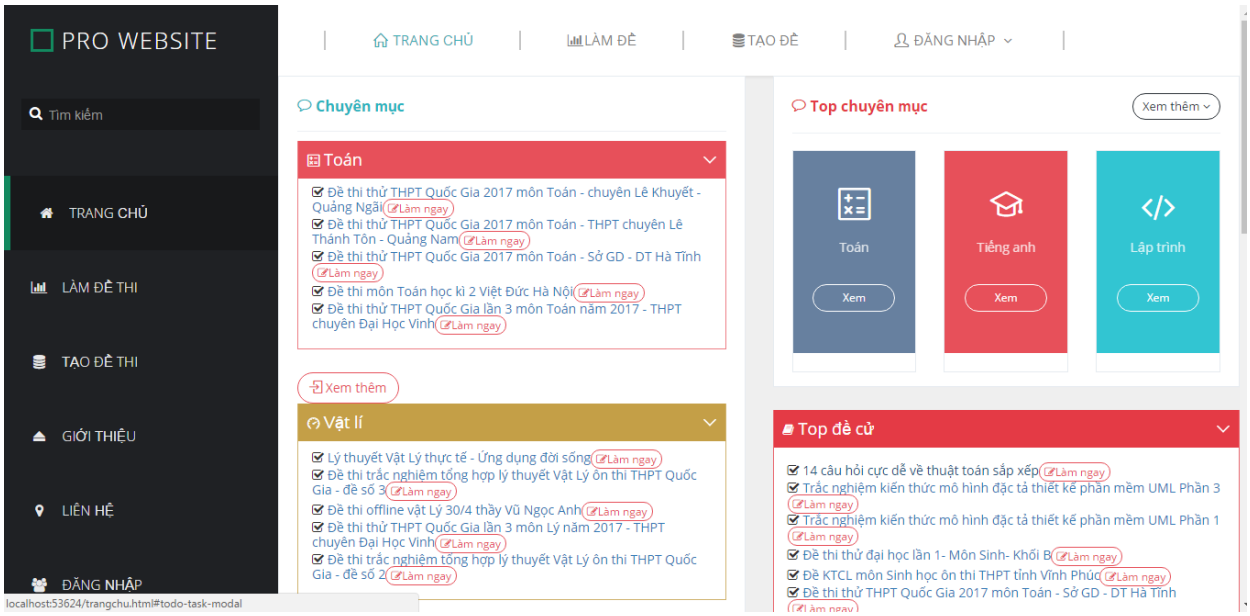
Hình 2.5.6.Paint

Toàn bộ quá trình này thường diễn ra trong vài giây hoặc hàng chục giây. Nhiệm vụ của chúng ta là làm cho nó nhanh hơn. Nếu JavaScript thay đổi một phần của trang, nó sẽ khiến Render Tree được tạo lại, và các bước Layout và Paint cũng được lặp lại. Tuy nhiên, trình duyệt hiện đại đã đủ thông minh để chỉ lặp lại những việc đó đối với phần được thay đổi.

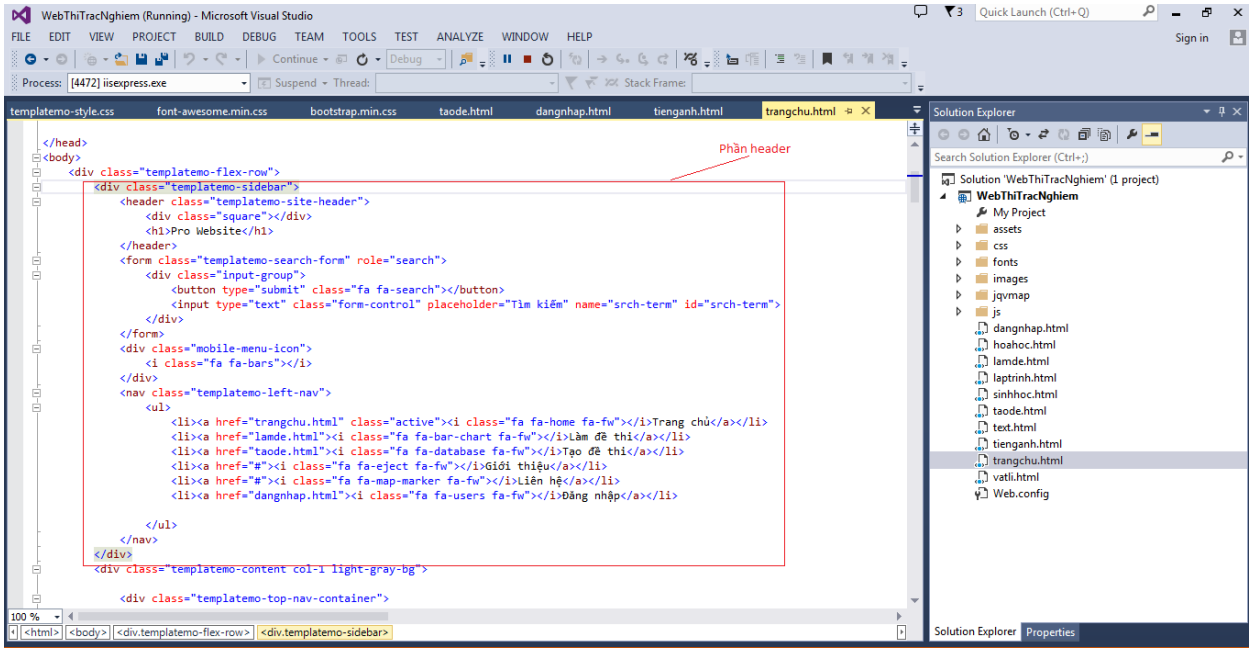
CHƯƠNG 3:KẾT QUẢ ĐẠT ĐƯỢC

3.1.Các giao diện đã thiết kế:

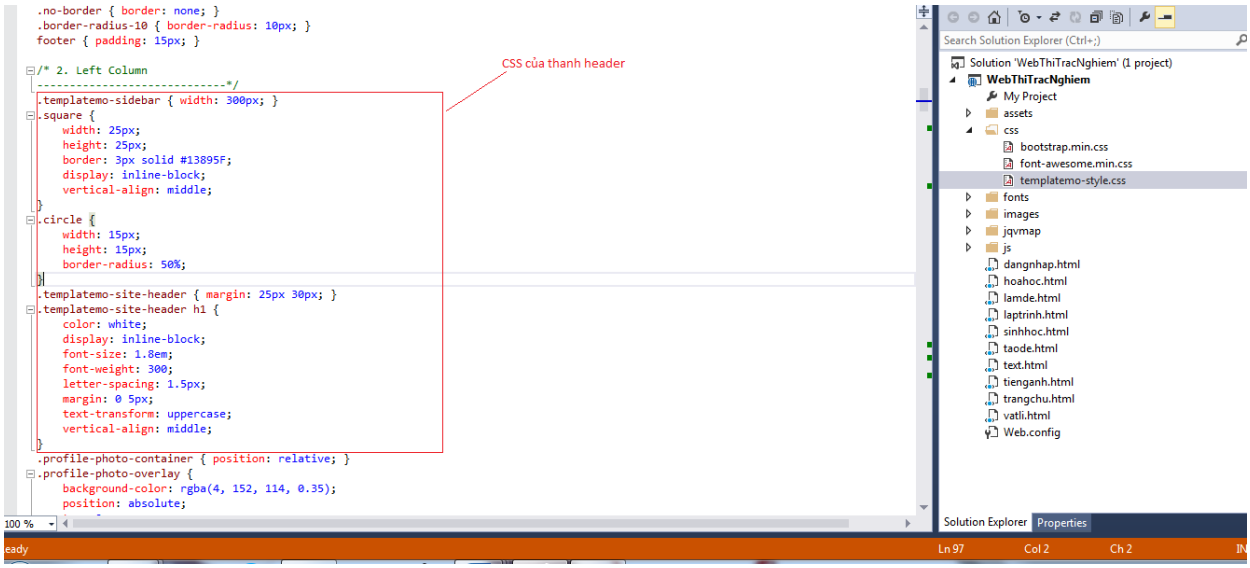
-Thiết kế giao diện web trắc nghiệm

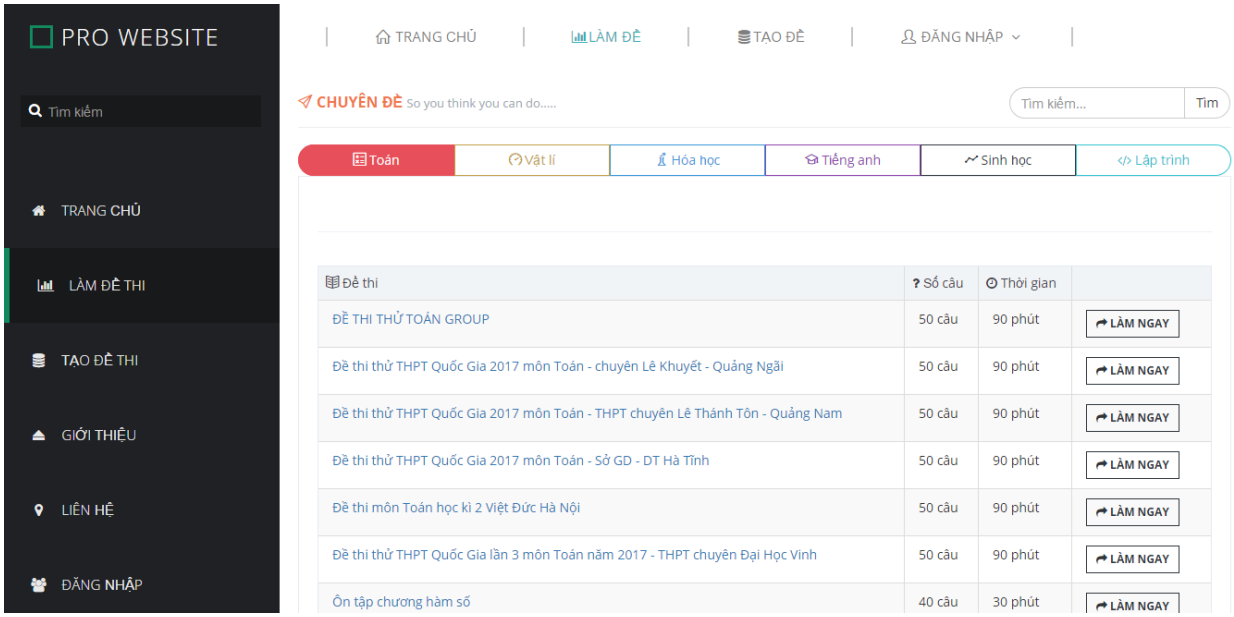


Hình 3.1.1.Giao diện trang chủ

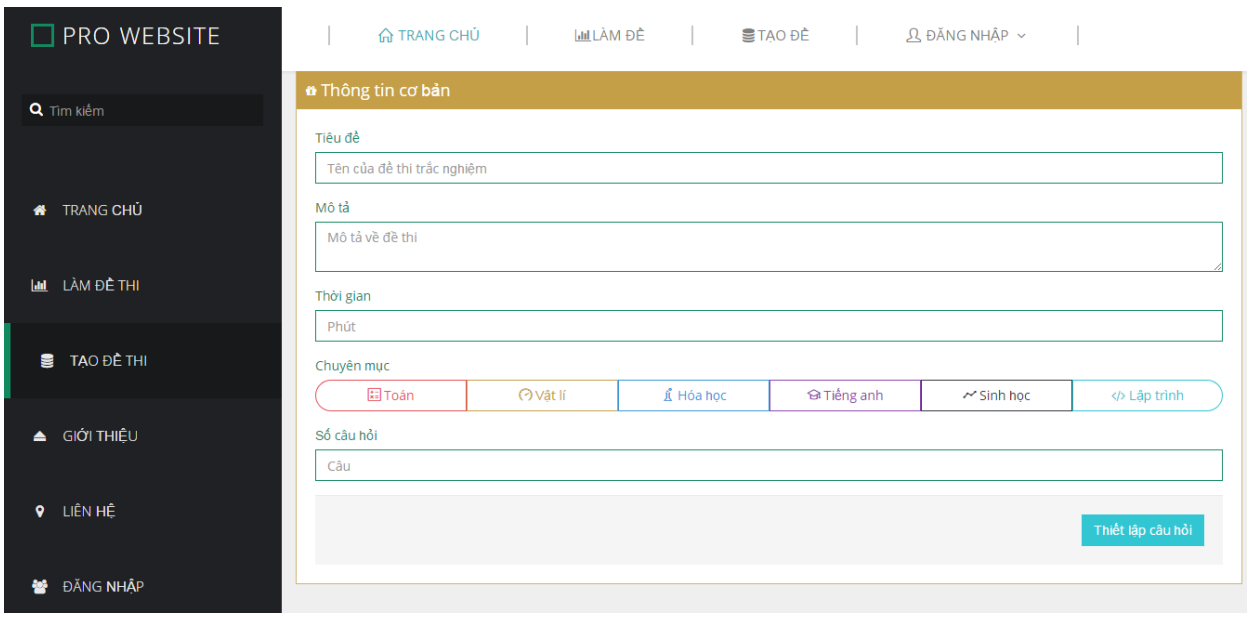


Hình 3.1.2.Header trang chủ

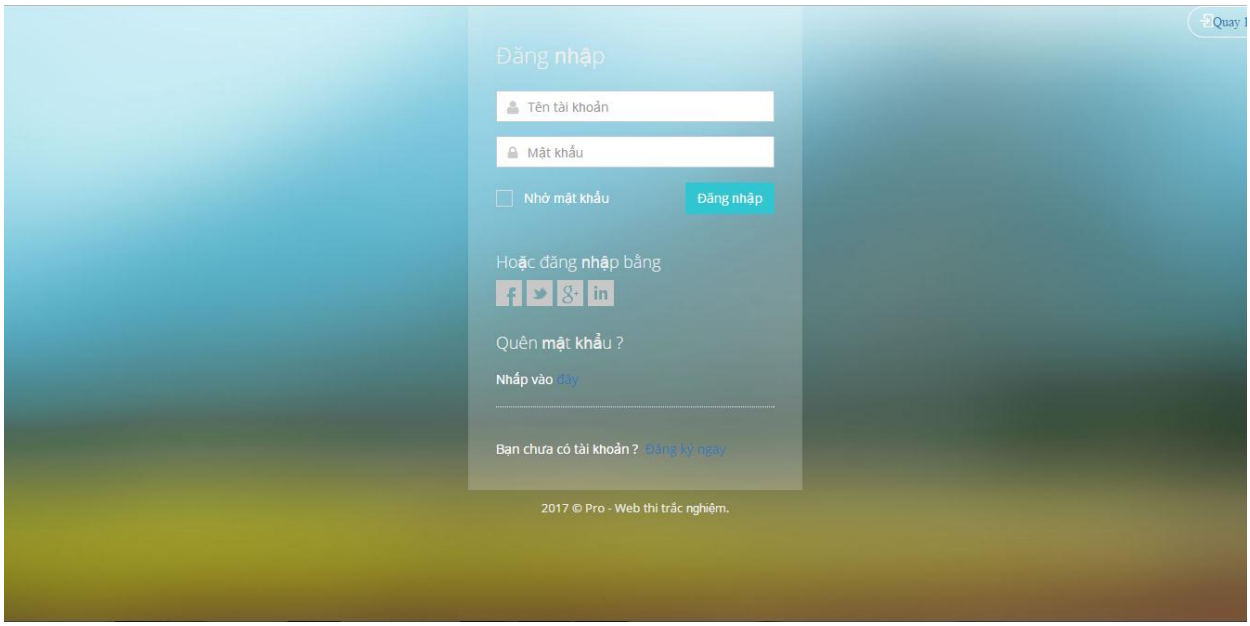




Hình 3.1.3.Giao diện trang làm đề



Hình 3.1.4.Giao diện trang tạo đề



Hình 3.1.5.Giao diện trang đăng nhập

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1.Kết quả làm được:

-Thiết kế giao diện web trắc nghiệm ,áp dụng một số template như core ui, metronic để giao diện trở nên đẹp mắt hơn ,thân thiện hơn với người dùng.

-Tối ưu hóa giao diện như hiển thị rõ ra các thông tin và thanh tìm kiếm để người dùng có truy cập nhanh vào nội dung cần thiết ,cấu trúc nội dung ,các bảng đơn giản dễ hiểu.

-Đầy đủ thông tin hợp lệ của người dùng khi đăng nhập hệ thống

-Bố cục nội dung rõ ràng ,chi tiết, các bảng đầy đủ những thông tin cần thiết.

4.2.Hướng phát triển:

-Thiết kế web theo chuẩn SEO

-Làm các nút thu gọn thanh header ,chạy trên mọi hệ điều hành

Tài liệu kham khảo

1. http://vnsolution.com.vn/kien-thuc-thiet-ke-web

2. http://www.brandsvietnam.com/7132-Infographic-8-bi-quyet-de-toi-uu-hoa-giao-dien-nguoi-dung-User-Experience

3. https://techmaster.vn/khoa-hoc/25487/web-co-ban-html5-css3-va-javascript