# PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**1. Thông tin chung**

|  |  |
| --- | --- |
| Tên đề tài | Xây dựng website bán đồ nội thất bằng ReactJs, NodeJs và Firebase |
| Họ và tên sinh viên: | Hà Văn Huy |
| Điện thoại liên lạc: | 0865569042 |
| Email: | Huyn11677@gmail.com |
| Lớp: | Công Nghệ Phần Mềm K64 |
| Hệ đào tạo: | Đại học chính quy |
| Đồ án tốt nghiệp được thực hiện tại: | Hà Nội |
| Thời gian làm ĐATN: | 2023 |

**2. Mục tiêu của ĐATN**

M

|  |  |  |
| --- | --- | --- |
| Chức năng | Quản trị | Khách hàng |
| Đăng nhập | Đăng nhập |
| Đăng kí, đổi mật khẩu, quên mật khẩu | Đăng xuất, đổi mật khẩu, quên mật khẩu |
| Tìm kiếm | Quản lý sản phẩm |
| Xem sản phẩm | Quản lý đơn hàng |
| Thêm giỏ hàng |  |
| Quản lý giỏ hàng |  |
| Đặt hàng |  |
| Thanh toán |  |

Bảng 0-1: Bảng mục tiêu đồ án

**3. Các nhiệm vụ cụ thể của ĐATN**

Để đạt được mục tiêu đề ra trong đề tài sẽ có những nội dung nghiên cứu sau:

* Khảo sát hệ thống:
* Khảo sát tình trạng cửa hàng hiện nayddang kinh doanh, quy trình bán hàng như thế nào, từ đó xây dựng website phù hợp với cửa hàng
* Phân tích thiết kế hệ thống
  + Từ quá trình khảo sát, phân tích và đưa ra bảng chức năng chính của hệ thống và phân tích chi tiết từng chức năng

**4. Lời cam đoan của sinh viên:**

Tôi – Hà Văn Huy – cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của ThS. Nguyễn Thị Hữu Phương

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Tác giả ĐATN |

**5. Xác nhận của giáo viên hướng dẫn về mức độ hoàn thành của ĐATN và cho phép bảo vệ:**

|  |  |
| --- | --- |
|  | *Hà Nội, ngày tháng năm*  Cán bộ hướng dẫn |

# TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

Sau một thời gian dày công tập trung vào đề tài của mình, em tự hào thông báo rằng dự án xây dựng trang web bán đồ nội thất của em đã hoàn thành thành công. Trang web này không chỉ có giao diện đẹp mắt, mà còn mang lại trải nghiệm thân thiện và dễ dàng sử dụng cho người dùng.

Quá trình này đã giúp em học được rất nhiều, không chỉ trong việc xây dựng trang web mà còn trong việc phát triển ứng dụng web nói chung. Em đã nắm vững được kiến thức cơ bản về Node.js và Firebase, và áp dụng chúng vào việc xây dựng trang web. Sự thành thạo trong việc sử dụng ReactJS cũng đã giúp em tạo ra trải nghiệm web tốt hơn cho người dùng.

Không chỉ dừng lại ở đó, em đã tự mình khám phá và nắm bắt các kiến thức lập trình mới để áp dụng vào dự án này. Khả năng tự học và áp dụng kiến thức mới là một điểm mạnh quan trọng mà em đã phát triển trong quá trình này.

Ngoài ra, em đã tích luỹ nhiều kinh nghiệm quý giá từ việc xây dựng trang web và phát triển phần mềm. Em đã học cách quản lý thời gian, giải quyết các thách thức kỹ thuật, và làm việc trong môi trường đòi hỏi sự cộng tác và sáng tạo.

Tất cả những điều này đã tạo nên một nền tảng vững chắc cho em để phát triển sự nghiệp trong tương lai. Em tự tin hơn về khả năng của mình và sẵn sàng đối mặt với những thách thức mới trong lĩnh vực công nghệ thông tin. Điều này cũng mở ra cơ hội cho em để làm việc trong các doanh nghiệp và thực hiện các dự án thú vị trong tương lai.

# LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành và sâu sắc đến tất cả những người và tổ chức đã hỗ trợ và đóng góp cho đề tài này. Sự giúp đỡ của các bạn đã là một phần không thể thiếu và vô cùng quý báu trong suốt quá trình làm đề tài.

Trong suốt thời gian này, các bạn đã luôn bên cạnh, động viên, và chia sẻ kiến thức. Sự hiểu biết và sự tận tâm của các bạn đã giúp em vượt qua những khó khăn và thách thức trong đề tài. Tôi biết ơn tất cả mọi người vì sự hỗ trợ không điều kiện và lòng nhiệt thành của các bạn.

Không thể không đề cập đến sự hướng dẫn và sự đồng hành quý báu của giảng viên hướng dẫn của em cô Nguyễn Thị Lan Phương. Cô đã tận dụng thời gian và kiến thức để hỗ trợ em trong từng bước của đề tài này. Sự kiên nhẫn và sự khích lệ của cô đã giúp em phát triển không chỉ về mặt chuyên môn mà còn về khả năng tự tin và khả năng giải quyết vấn đề.

Em cũng xin gửi lời cảm ơn đến gia đình và bạn bè vì đã luôn ủng hộ và động viên em trong suốt thời gian đề tài. Sự hỗ trợ tinh thần của họ đã đóng vai trò quan trọng trong việc duy trì động lực và tinh thần làm việc của em.

Cuối cùng, em muốn biết ơn tất cả những người bạn, đồng nghiệp và những người đã đóng góp ý kiến và phản hồi trong quá trình nghiên cứu. Cuộc hội thoại và sự chia sẻ kiến thức của các bạn đã làm cho dự án trở nên phong phú và sáng sủa hơn.

Đề tài này đã được hoàn thành thành công nhờ vào sự kết hợp của tất cả những đóng góp và hỗ trợ từ các bạn và những người thân yêu. Em biết ơn mọi người vì đã đồng hành và đóng góp vào đề tài này.

Xin chân thành cảm ơn và hy vọng rằng sự thành công và hạnh phúc sẽ đến với mọi người!

# MỤC LỤC

[PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP I](#_Toc511117910)

[TÓM TẮT NỘI DUNG ĐỒ ÁN TỐT NGHIỆP III](#_Toc511117911)

[LỜI CẢM ƠN IV](#_Toc511117912)

[MỤC LỤC V](#_Toc511117913)

[DANH MỤC CÁC HÌNH VẼ VII](#_Toc511117914)

[DANH MỤC CÁC BẢNG BIỂU VIII](#_Toc511117915)

[MỞ ĐẦU 1](#_Toc511117916)

[CHƯƠNG 1 TỔNG QUAN 2](#_Toc511117917)

[Tóm tắt chương 2](#_Toc511117918)

[1.1 Hướng dẫn sử dụng mẫu đồ án tốt nghiệp (Tiêu đề mục – Style Heading 2) 2](#_Toc511117919)

[1.2 Kết chương 4](#_Toc511117920)

[CHƯƠNG 2 CƠ SỞ LÝ THUYẾT 5](#_Toc511117921)

[2.1 Tóm tắt chương 5](#_Toc511117922)

[2.2 Trình bày Headings 5](#_Toc511117923)

[2.3 Cách chèn nhãn cho bảng biểu và hình vẽ 5](#_Toc511117924)

[2.4 Hướng dẫn chèn công thức và nhãn 7](#_Toc511117925)

[2.5 Hướng dẫn chèn tài liệu tham khảo: 7](#_Toc511117926)

[2.6 Chèn danh mục tài liệu tham khảo 9](#_Toc511117927)

[2.7 Kết chương 11](#_Toc511117928)

[CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG 12](#_Toc511117929)

[3.1 Tóm tắt chương 12](#_Toc511117930)

[3.2 Tiêu đề mục 12](#_Toc511117931)

[3.2.1 Tiêu đề tiểu mục 12](#_Toc511117932)

[3.2.2 Tiêu đề tiểu mục 12](#_Toc511117933)

[3.3 Tiêu đề mục 13](#_Toc511117934)

[3.3.1 Tiêu đề tiểu mục 13](#_Toc511117935)

[3.3.2 Tiêu đề tiểu mục 13](#_Toc511117936)

[3.4 Kết chương 13](#_Toc511117937)

[CHƯƠNG 4 KẾT QUẢ CÀI ĐẶT, THỬ NGHIỆM HOẶC KẾT QUẢ NGHIÊN CỨU 14](#_Toc511117938)

[4.1 Tóm tắt chương 14](#_Toc511117939)

[4.2 Tiêu đề mục 14](#_Toc511117940)

[4.2.1 Tiêu đề tiểu mục 14](#_Toc511117941)

[4.2.2 Tiêu đề tiểu mục 14](#_Toc511117942)

[4.3 Tiêu đề mục 15](#_Toc511117943)

[4.3.1 Tiêu đề tiểu mục 15](#_Toc511117944)

[4.3.2 Tiêu đề tiểu mục 15](#_Toc511117945)

[4.4 Kết chương 15](#_Toc511117946)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 16](#_Toc511117947)

[TÀI LIỆU THAM KHẢO 17](#_Toc511117948)

[PHỤ LỤC 18](#_Toc511117949)

# DANH MỤC CÁC HÌNH VẼ

[Hình 1‑1 Thao tác cập nhật mục lục 3](#_Toc511117950)

[Hình 2‑1 Cách chèn nhãn cho hình 5](#_Toc511117951)

[Hình 2‑2 Cách tạo một nhãn mới 6](#_Toc511117952)

[Hình 2‑3 Cách tham chiếu đến một nhãn 6](#_Toc511117953)

[Hình 2‑4 Quản lý tài liệu tham khảo 7](#_Toc511117954)

[Hình 2‑5 Tạo mới một tài liệu tham khảo 7](#_Toc511117955)

[Hình 2‑6 Thêm chi tiết tài liệu tham khảo 8](#_Toc511117956)

[Hình 2‑7 Thêm tài liệu tham khảo vào danh sách 8](#_Toc511117957)

[Hình 2‑8 Chọn kiểu hiển thị tài liệu tham khảo 9](#_Toc511117958)

[Hình 2‑9 Thêm một tài liệu tham khảo vào văn bản. 9](#_Toc511117959)

[Hình 2‑10 Thêm danh sách các tài liệu tham khảo 10](#_Toc511117960)

[Hình 2‑11 Cập nhật danh sách tài liệu tham khảo 10](#_Toc511117961)

[Hình 2‑12 Chỉnh sửa định dạng hiển thị danh sách các tài liệu tham khảo 10](#_Toc511117962)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 2‑1 Tên bảng 6](#_Toc511117963)

# Danh mục các từ viết tắt

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Từ viết tắt | Từ tiếng Anh | Từ và nghĩa tiếng Việt |
| 1 | CSDL |  | Cơ sở dữ liệu |
| 2 | DB | Database | Cơ sở dữ liệu |
| 3 | TMDT | Ecommerce | Thương mại điện tử |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# MỞ ĐẦU

1. Tổng quan tình hình nghiên cứu thuộc lĩnh vực của đề tài

Hiện nay, cuộc sống ngày càng nhộn nhịp và hối hả. Chúng ta đều luôn muốn có thể làm nhiều việc cùng một lúc. Bạn nghĩ sao nếu chúng ta không cần đến siêu thị mà vẫn có thể mua đồ dùng cho cả gia đình? Trước kia với việc đó là không thể, nhưng bây giờ với sự phát triển của khoa học công nghệ điều đó sẽ trở nên dễ dàng hơn rất nhiều nếu chúng ta có trên tay một chiếc smartphone.

Hàng ngày chúng ta bắt đầu với hàng tá công việc như đi làm, dọn dẹp, chăm sóc gia đình,...và những kế hoạch khác. Mỗi lần siêu thị mua đồ chiếm khá nhiều thời gian vì phải đi lại chọn đồ và hãy thử nghĩ xem nếu chúng ta không ở gần siêu thị đó chúng ta sẽ vất vả như thế nào cho mỗi lần đi siêu thị. Hiện nay hầu hết các siêu thị đã xây dựng những trang web riêng cho mình để khách hàng có thể tiện lợi trong việc mua hàng hóa qua mạng mà không cần đến tận nơi. Nhưng có thể thấy những năm gần đây dường như là kỷ nguyên của thiết bị di động thông minh, nó phát triển một cách nhanh chóng và bất cứ nơi đầu ta cũng có thể thấy sự xuất hiện của nó, nhất là những chiếc điện thoại thông minh. Không ai muốn bỏ cả công việc, cuộc hẹn,...để đến tận nơi mua đồ hay dành thời gian với chiếc máy tính để đặt hàng trong khi có thể làm mọi việc cùng một lúc. Cái họ cần là một thiết bị cầm tay tiện lợi, giúp họ có thể trải nghiệm tìm kiếm thông tin, mua bán mọi lúc mọi nơi.

2. Tính cấp thiết, ý nghĩa khoa học và thực tiễn của đề tài

Từ những phân tích trên, ý tưởng cho một ứng dụng mua bán trên di động hình thành nhằm mục đích giúp cho người sử dụng cũng như các doanh nghiệp thuận tiện hơn trong việc trao đổi mua bán và giới thiệu sản phẩm tới người sử dụng.

Trong báo cáo đồ án này sẽ tập trung trình bày những nghiên cứu kiến thức cơ bản về lập trình di động, đi sâu vào nền tảng Android. Triển khai các bước thiết kế, xây dựng ứng dụng giúp người dùng tra cứu thông tin, mua hàng, tìm kiếm, áp dụng công nghệ bản đồ hỗ trợ hiển thị các địa điểm, giúp người dùng tra cứu tiện lợi.

Đây là ứng dụng thực tế khá hay trên điện thoại, tuy không phải là mới nhưng do kinh nghiệm của em chưa nhiều và còn nhiều hạn chế cũng như sai sót mong thầy cô và các bạn đóng góp ý kiến để đề tài được hoàn thiện hơn.

# TỔNG QUAN

## Lý do chọn đề tài

Thương mại điện tử, hay còn được gọi tắt là TMĐT, đã trở thành một phần không thể thiếu trong cuộc sống hiện đại. Đây là một lĩnh vực mà con người mua bán hàng hóa và dịch vụ thông qua mạng internet và các nền tảng kỹ thuật số khác. TMĐT không chỉ là một xu hướng, mà còn là một cuộc cách mạng đã thay đổi cách con người mua sắm và kinh doanh. Trong bối cảnh này, việc áp dụng thương mại điện tử vào lĩnh vực bán đồ nội thất trở thành một quyết định có ý nghĩa và tiềm năng đầy hứa hẹn.

TMĐT mang đến nhiều lợi ích đối với cả người tiêu dùng và doanh nghiệp. Đối với người tiêu dùng, nó mang lại sự tiện lợi không thể tưởng tượng. Bạn có thể mua sắm từ bất kỳ đâu và bất kỳ lúc nào, không còn cần phải đi đến cửa hàng vật lý. Điều này giúp tiết kiệm thời gian và công sức đáng kể. Thêm vào đó, TMĐT giúp người tiêu dùng tiếp cận sản phẩm đa dạng từ nhiều nguồn khác nhau, từ đó tạo điều kiện cho họ để tìm được sản phẩm và giá cả phù hợp nhất cho nhu cầu cá nhân.

Đối với doanh nghiệp, TMĐT cung cấp một cơ hội lớn để mở rộng thị trường. Bằng cách có một trang web TMĐT chuyên về bán đồ nội thất, các doanh nghiệp trong ngành có thể tiếp cận nhiều khách hàng hơn, không bị giới hạn bởi địa lý. Điều này mở ra cơ hội thị trường toàn cầu, cho phép họ kinh doanh với khách hàng trên khắp cả nước và thậm chí quốc tế. Đồng thời, trang web TMĐT cung cấp một nền tảng để trưng bày các sản phẩm đa dạng, giúp khách hàng dễ dàng xem xét và so sánh chúng, tạo ra sự thuận tiện trong việc mua sắm và đồng thời giúp họ đưa ra quyết định mua sắm thông thái hơn.

Với những lợi ích này, việc áp dụng thương mại điện tử vào lĩnh vực bán đồ nội thất là một quyết định có lý do và tiềm năng lớn. Đồ nội thất đang trở thành một phần không thể thiếu trong cuộc sống hàng ngày, và việc có một trang web TMĐT chuyên về đồ nội thất giúp đáp ứng nhu cầu ngày càng tăng của người tiêu dùng. Nó không chỉ mở rộng thị trường cho doanh nghiệp mà còn giúp tạo ra cơ hội kinh doanh đáng giá. Thế giới đang chuyển đổi nhanh chóng, và sứ mệnh của TMĐT trong lĩnh vực bán đồ nội thất là mở cửa cho sự thay đổi và phát triển, đồng thời mang lại sự tiện lợi và lựa chọn đa dạng cho người tiêu dùng.

## Mục tiêu của đề tài

Để tạo ra một trang website thương mại điện tử chuyên về đồ nội thất với giao diện thân thiện, dễ sử dụng và tích hợp các tính năng quan trọng, website đó cần đáp ứng được các yêu cầu của người dùng như:

* Về chức năng của quản trị viên:

+ Chức năng đăng nhập/đăng xuất

+ Quản lý thuộc loại sản phẩm

+ Thêm bài viết, xóa các bài viết trong Blog

+ Quản lý sản phẩm: thêm, sửa, xóa các sản phẩm, thông tin của sản phẩm

+ Quản lý vouche

+ Quản lý nhà cung cấp

+ Quản lý thương hiệu

* Về chức năng của khách hàng:

+ Tìm kiếm sản phẩm theo tên sản phẩm

+ Xem chi tiết sản phẩm, thêm giỏ hàng, thanh toán (tạo hóa đơn)

+ Xem, kiểm tra tình trạng của hóa đơn

+ Đăng nhập, đăng ký bằng tải khoản đã đăng ký

+ Xem hoặc thay đổi thông tin cá nhân

+ Xem hoặc hủy đơn hàng đã đặt

## Phạm vi nghiên cứu

Website được xây dựng bằng các phần mềm và công nghệ:

* Ngôn ngữ: ReactJs, NodeJs
* Cơ sở dữ liệu Firebase
* Phần mềm Visual Studio Code

Trang web tham khảo:

* woodmart.xtemos.com
* colorlib.com
* 10015.io

## Bố cục đồ án

* Chương 1: Tổng quan

Giới thiệu tổng quan về đề tài website bán đồ nội thất gồm lý do chọn đề tài , mục tiêu và phạm vi nghiên cứu của đề tài.

* Chương 2: Cơ sở lý thuyết

Chú trọng đến lý thuyết trọng tâm của đề tài: Giới thiệu cơ sở lý thuyết về ngôn ngữ được sử dụng trong đồ án gồm ReactJs và NodeJs,… Cơ sở dữ liệu Firebase

* Chương 3: Phân tích và thiết kế hệ thống
* Chương 4: Xây dựng website

# CHƯƠNG 2 Cơ SỞ LÝ THUYẾT

#### 2.1 Giới thiệu chung về ReactJs

ReactJS, được tạo ra bởi Facebook, là một thư viện JavaScript mạnh mẽ và phổ biến trong việc phát triển giao diện người dùng trên web. Với thiết kế "component-based," React cho phép chia giao diện thành các thành phần nhỏ, dễ quản lý và tái sử dụng. Điều này giúp tạo ra mã nguồn dễ bảo trì và phát triển nhanh chóng.

Một trong những ưu điểm lớn của React là sử dụng Virtual DOM để tối ưu hóa hiệu suất ứng dụng. Thay vì làm mới toàn bộ DOM mỗi khi có thay đổi, React chỉ cập nhật các phần cần thiết, làm cho ứng dụng nhanh hơn và mượt mà hơn. Hơn nữa, React có sự hỗ trợ mạnh mẽ từ cộng đồng và nhiều thư viện bổ sung, giúp việc phát triển ứng dụng trở nên dễ dàng hơn.



Hình 2.1: Biểu tượng ReactJs

Với React, bạn có khả năng xây dựng các ứng dụng web hiện đại, từ trang web tĩnh đến ứng dụng động phức tạp. React không chỉ giúp tạo ra giao diện đẹp mắt mà còn giúp quản lý trạng thái ứng dụng một cách hiệu quả với các thư viện quản lý trạng thái như Redux hoặc MobX. Điều này biến React thành một công nghệ không thể thiếu cho những người phát triển ứng dụng web.

### 2.1.1 React là gì?

React (ReactJs hay React.js) là 1 thư viện JavaScript mã nguồn mở được phát triển bởi đội ngũ kỹ sư đến từ Facebook; nó được giới thiệu vào năm 2011, cho đến nay là đã được hơn 10 năm. Nguyên lý xây dựng của React dựa trên components (component-based approach), có thể tái sử dụng và phù hợp với ứng dụng 1 trang (Single Page Application – SPA). React giúp lập trình viên xây dựng giao diện người dùng dựa trên JSX (môt cú pháp mở rộng của JavaScript), tạo ra các DOM ảo (virtual DOM) để tối ưu việc render 1 trang web.

ReactJs sau khi ra đời đã cho thấy sự phù hợp của nó trong việc phát triển các ứng dụng Web với nhiều chức năng được tích hợp. Nó đã tạo thành 1 xu thế, 1 hình mẫu phát triển website với nhiều chức năng, khả năng tương tác đa dạng với người dùng. Hiện tại sau hơn 10 năm phát triển thì React vẫn đang chiếm vị trí số 1 trong các thư viện Front-end hiện tại.

### 2.1.2 Virtual Dom

Giới thiệu: Virtual DOM là một biểu diễn ảo của Document Object Model (DOM). Nó là một cây DOM nhẹ được duyệt bởi React và được sử dụng để thể hiện trạng thái hiện tại của giao diện người dùng của ứng dụng.

Cách hoạt động: Khi trạng thái của ứng dụng React thay đổi (ví dụ: do dữ liệu thay đổi hoặc sự kiện người dùng), React tạo ra một cây DOM ảo mới, được so sánh với cây DOM ảo cũ và xác định các thay đổi cần thiết để cập nhật giao diện người dùng thật sự. Thay vì cập nhật trực tiếp DOM thật, React chỉ cập nhật các phần của DOM cần thay đổi, giúp tối ưu hóa hiệu suất ứng dụng.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, Đồ họa

Mô tả được tạo tự động

Hình 2.2: Ảnh cập nhật Virtual DOM sang Real DOM

Tối ưu hóa hiệu suất: Sử dụng Virtual DOM giúp giảm thiểu việc tương tác trực tiếp với DOM, điều này làm giảm tải cho trình duyệt và làm tăng hiệu suất của ứng dụng web. Thay vì thực hiện nhiều thao tác DOM tốn kém, React chỉ cập nhật những phần DOM cần thiết, giúp tránh việc làm mới toàn bộ trang.

Đảm bảo tính nhất quán: Virtual DOM giúp đảm bảo tính nhất quán của giao diện người dùng, bởi vì nó xác định cách cập nhật DOM dựa trên trạng thái hiện tại của ứng dụng, không phụ thuộc vào các thay đổi DOM bên ngoài.

Hỗ trợ bởi React: Virtual DOM là một phần quan trọng của cách React hoạt động. React sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện và đảm bảo rằng các thay đổi trạng thái của ứng dụng được phản ánh một cách hiệu quả trên giao diện người dùng.

### 2.1.3 Giới thiệu về JSX

* Cú pháp rõ ràng: JSX sử dụng cú pháp rất gần với HTML, với thẻ mở và thẻ đóng, thuộc tính và sự kiện. Điều này giúp làm cho mã nguồn dễ đọc và dễ hiểu hơn, đặc biệt là đối với những người đã quen thuộc với HTML.
* Tích hợp JavaScript mạnh mẽ: JSX cho phép bạn sử dụng biểu thức JavaScript bên trong mã HTML bằng cách đặt chúng trong cặp dấu ngoặc nhọn {}. Điều này giúp tích hợp dữ liệu động và logic JavaScript vào giao diện người dùng một cách dễ dàng.
* Component-Based Development: JSX thúc đẩy phát triển dựa trên thành phần (component-based development). Bạn có thể tạo các thành phần UI độc lập, sau đó kết hợp chúng lại với nhau để xây dựng giao diện phức tạp. Điều này giúp tái sử dụng mã và quản lý mã nguồn một cách hiệu quả.
* Kiểm tra tĩnh: JSX hỗ trợ kiểm tra tĩnh (static analysis) bằng cách sử dụng các trình kiểm tra mã như ESLint hoặc TypeScript. Điều này giúp phát hiện các lỗi trong mã trước khi chạy ứng dụng, giúp tăng tính đáng tin cậy của ứng dụng.
* Tích hợp dễ dàng: JSX không phải là một ngôn ngữ riêng biệt, mà là JavaScript thông thường. Điều này có nghĩa rằng bạn có thể sử dụng JSX bên trong các tệp JavaScript và tích hợp nó vào các dự án JavaScript hiện có một cách dễ dàng.

### 2.1.4 Giới thiệu về Component

Thành phần (Component) là một khái niệm quan trọng trong phát triển ứng dụng sử dụng React, một thư viện JavaScript phổ biến cho việc xây dựng giao diện người dùng động trên web và ứng dụng di động. Thành phần là khối xây dựng cơ bản của mọi ứng dụng React, và chúng giúp tạo ra một cách tổ chức, tái sử dụng và quản lý giao diện người dùng một cách hiệu quả. Dưới đây là một sự giới thiệu về thành phần:

* Sự Tái Sử Dụng: Thành phần cho phép xây dựng một lần và sử dụng nhiều lần. Có thể tạo các thành phần như nút, biểu đồ, menu, hoặc bất kỳ phần giao diện nào có thể được sử dụng lại trong ứng dụng web.
* Tách Biệt Trạng Thái: Mỗi thành phần có thể quản lý trạng thái của riêng nó. Điều này có nghĩa rằng có thể tạo ra các thành phần độc lập, mỗi thành phần quản lý dữ liệu của mình, giúp duy trì tính nhất quán và dễ dàng kiểm tra.
* Tổ Chức Rõ Ràng: Thành phần giúp tổ chức mã nguồn một cách rõ ràng hơn. Có thể chia thành các phần nhỏ dựa trên chức năng hoặc mục đích, mỗi phần chứa một hoặc nhiều thành phần liên quan.
* Dễ Quản Lý: Với thành phần có thể dễ dàng thay đổi và bảo trì các phần cụ thể của giao diện mà không cần can thiệp vào các phần khác của ứng dụng.
* Phát Triển Song Song: Điều này cho phép nhiều nhà phát triển làm việc đồng thời trên các thành phần khác nhau mà không gây xung đột.
* Tích Hợp Dễ Dàng: React cho phép tổng hợp các thành phần lại với nhau để xây dựng giao diện người dùng hoàn chỉnh, có thể kết hợp các thành phần con thành các thành phần cha, tạo thành một cây thành phần phức tạp.

### 2.1.5 Props và state

* Props (Properties)
  + Props là viết tắt của "properties" và là một cách để truyền dữ liệu từ một thành phần cha (parent component) đến một thành phần con (child component).
  + Props được sử dụng để truyền dữ liệu tĩnh, không thay đổi, từ ngoài vào bên trong một thành phần. Thường thì props được khai báo trong thành phần cha và sau đó được truyền cho các thành phần con thông qua các thuộc tính HTML giống như {}.
  + Props là không thể thay đổi (immutable) từ bên trong thành phần con. Nghĩa là một thành phần con không thể thay đổi giá trị của props mà nó nhận được từ thành phần cha.
* State
  + State là một cách để quản lý và theo dõi trạng thái dữ liệu của một thành phần trong thời gian chạy. State là dữ liệu có thể thay đổi và ảnh hưởng đến giao diện người dùng.
  + Thường thì state được khai báo và quản lý trong thành phần bằng cách sử dụng phương thức setState. Khi state thay đổi, React sẽ tự động cập nhật lại giao diện người dùng để phản ánh trạng thái mới.
  + State thường được sử dụng cho các dữ liệu có thể thay đổi như thông tin đăng nhập, danh sách các mục trong một ứng dụng, và các tương tác người dùng khác.
* Tóm lại, props được sử dụng để truyền dữ liệu từ cha sang con và là không thể thay đổi, trong khi state là cách React quản lý và theo dõi trạng thái dữ liệu có thể thay đổi bên trong một thành phần. Sử dụng cả hai khái niệm này cùng nhau giúp bạn xây dựng và quản lý giao diện người dùng động trong ReactJS.

### 2.1.6 React Router

React-Router là một thư viện định tuyến (routing) tiêu chuẩn trong React. Nó giữ cho giao diện của ứng dụng đồng bộ với URL trên trình duyệt. React-Router cho phép định tuyến "luồng dữ liệu" (data flow) trong ứng dụng một cách rõ ràng.

Một số thành phần quan trọng trong React Router bao gồm:

+ BrowserRouter: là thành phần chính để bọc toàn bộ ứng dụng và quản lý việc định tuyến.

+ Route: được sử dụng để xác định đường dẫn URL và nội dung tương ứng hiển thị trên trang.

+ Link: là một thành phần tạo liên kết trong ứng dụng. Khi người dùng nhấp vào liên kết này, trình duyệt sẽ chuyển đến đường dẫn URL tương ứng.

Để thực hiện việc định tuyến trong React Router, chúng ta cần xác định các Route và nội dung tương ứng trong một component chính. Các Route sẽ được xác định dựa trên đường dẫn URL và liên kết với các component tương ứng. Khi người dùng truy cập vào một đường dẫn URL cụ thể, Route sẽ hiển thị nội dung của component tương ứng.

### 2.1.6.1 Router

Route: Định nghĩa một ánh xạ (mapping) giữa một URL và một Component. Điều đó có nghĩa là khi người dùng truy cập theo một URL trên trình duyệt, một Component tương ứng sẽ được render trên giao diện.

Trong đó:

* path: Là đường dẫn trên URL.
* exact: Liúp cho route này này chỉ hoạt động nếu URL trên trình duyệt phù hợp tuyệt đối với giá trị của thuộc tính path của nó.
* component: Là component sẽ đươc load ra tương ứng với Route đó.

### 2.1.6.2 Link

Trong HTML thì cặp thẻ để chuyển hướng đó là thẻ <a></a> thì trong react chúng ta sẽ dử dụng cặp thẻ <Link></Link> được import từ React-Router.

trong đó:

* to: Giống như thuộc tính href trong thẻ a.

### 2.1.6.3 NavLink

NavLink thì rất giống với Link về cách sử dụng, nhưng NavLink tốt hơn vì nó hỗ trợ thêm một số thuộc tính như là activeClassName và activeStyle 2 thuộc tính này giúp cho khi mà nó trùng khớp thì nó sẽ được active lên và chúng ta có thể style cho nó.

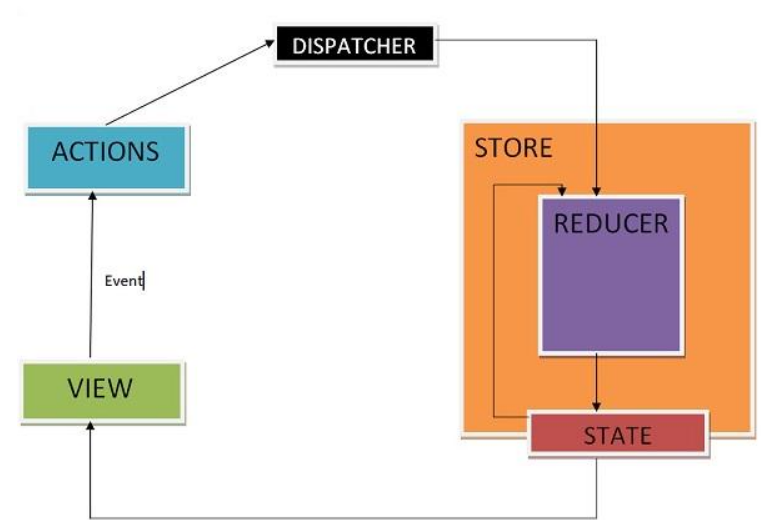
### 2.1.6.4 Redirect

* Chức năng dùng để chuyển trang.
* Có thể truy xuất thông tin trang trước đó thông qua đối tượng location. Để sử dụng Redirect ta chỉ cần import nó từ react-router.

Khi bạn muốn sử dụng location thì tại cấu hình Router ta chỉ cần truyền thêm đối tượng location vào component mà cần sử dụng đối tượng location.

## 2.1.7 React Redux

Redux là một thư viện quản lý State (trạng thái) mã nguồn mở và chạy trên phía client, tạo ra một store toàn cục trong đó State có thể được chia sẻ giữa các component mà không cần thiết phải có kết nối cha-con (Truyền từ cha xuống con).



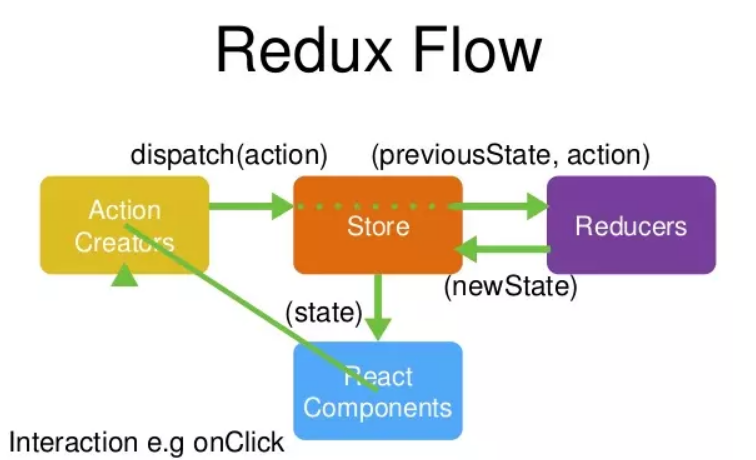
Hình 2.3: Cấu trúc của Redux

Redux gồm có 3 thành phần chính là:

* **Action:** Action là một hàm đơn giản trả về một đối tượng và có trách nhiệm gửi dữ liệu đến kho lưu trữ (store) thông qua hàm dispatch và reducer. Nó có hai thuộc tính - payload và type. Hành động được gọi là “type” và giá trị được áp dụng để thay đổi trạng thái của ứng dụng được gọi là “payload”.
* **Reducer:** “Reducer” là một hàm được sử dụng để thay đổi trạng thái của ứng dụng theo cách phụ thuộc vào các loại action cụ thể. Hàm nhận hai đối số: trạng thái hiện tại và action, và trả về một trạng thái mới. Chức năng chính của reducer là tạo ra một trạng thái mới cho store.
* **Store:** Store là một đối tượng JavaScript và là một kho chứa trung tâm quản lý trạng thái, có thể truy cập từ toàn bộ ứng dụng. Phương thức duy nhất để truy cập vào store là thông qua việc gửi action.

Nguyên lý hoạt động của Redux:

* Bước 1: Khi có 1 sự kiện (event) như là GET, POST, UPDATE, DELETE... thì thằng action creators sẽ sinh ra 1 action mô tả những gì đang xảy ra.
* Bước 2: Action sẽ thực hiện điều phối Reducer xử lý event thông qua hàm dispatch(action).



Hình 2.4: Nguyên lý đoạt động của Redux

* Bước 3: Reducer dựa vào những mô tả của Action để biết cần thực hiện thay đổi gì trên State và thực hiện update.
* Bước 4: Khi State được update thì các trigger đang theo dõi state đó sẽ nhận được thông tin update và tiến hành render lại phần view để hiển thị ra cho người dùng

Quá trình xây dựng Redux được các chuyên gia ứng dụng 3 nguyên lý vận hành cơ bản sau:

+ Nguyên lý đầu tiên (Sử dụng duy nhất một nguồn dữ liệu đáng tin cậy): Các state của tất cả các ứng dụng đều được nằm trong cùng một object tree thuộc duy nhất một Store.

+ Nguyên lý thứ 2 (Chỉ được phép đọc trạng thái): Điều này có nghĩa rằng, cách duy nhất để người dùng có thể thay đổi state của ứng dụng là phát một action.

+ Nguyên lý thứ 3 (Thay đổi bằng hàm thuần túy): Người dùng cần sử dụng các pure function gọi là Reducer để chỉ ra cách state được biến đổi từ action.



## 2.2 Tổng quan về NodeJs

### 2.2.1 Khái niệm

Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ. Đây là một nền tảng (platform) phát triển độc lập dựa trên V8 JavaScript engine. Đây là một trình thông dịch thực thi mã JavaScript cho phép tạo các ứng dụng web như video clip và diễn đàn, đặc biệt có thể mở rộng nhanh chóng và dễ dàng cũng như thu hẹp phạm vi hoạt động của các trang mạng mạng xã hội.

* Phần core sử dụng Javascript và C++ cho phép xử lý với hiệu năng cao.
* Phù hợp với các ứng dụng xử lý nhanh, real time hoặc những ứng dụng cần thay đổi công nghệ nhanh.
* Chạy sigle thread nhưng có cơ chế non blocking giúp chạy bất đồng bộ, tăng khả năng xử lý.
* Thường phát triển theo hướng module hóa thành các phần nhỏ của ứng dụng và có thể thêm module có sẵn bên ngoài thông qua NPM, yarn....

Các ứng dụng viết bằng NodeJS:

* **Ứng dụng cho việc trò chuyện trong thời gian thực:** Nhờ vào cấu trúc mang tính không đồng bộ đơn luồng, NodeJS là ứng cử viên thích hợp với mục đích xử lý giao tiếp tại thời gian thực. Đến với nền tảng này, bạn dễ dàng có thể mở rộng quy mô và nó cũng thường dùng để tạo các chatbot. Hơn thế nữa, những tính năng liên quan đến ứng dụng trò chuyện này như: chat với nhiều người, thông báo đẩy cũng dễ dàng được bổ sung nhờ vào NodeJS.
* **Internet of Things (IoT):** Các ứng dụng IoT này thường bao gồm các bộ cảm biến phức tạp nhằm gửi các phần dữ liệu nhỏ. NodeJS cũng chính là một lựa chọn hợp lý để xử lý các yêu cầu đồng thời này với một tốc độ cực kỳ nhanh.\
* **Tính năng truyền dữ liệu:** Netflix chính là một trong số các công ty lớn điển hình trên thế giới chuyên sử dụng NodeJS với mục đích truyền các dữ liệu. Sở dĩ đây chính là nền tảng nhẹ và cực kỳ nhanh. Đồng thời nó còn cung cấp API một chuyên dùng cho stream.
* **Các Single-page application (SPA) phức tạp:** Trong SPA, toàn bộ các ứng dụng được load vào ở một trang duy nhất, nhờ đó mà sẽ có một số yêu cầu (request) được thực hiện ở nền. Vòng lặp sự kiện – event loop – của NodeJS cho phép xử lý được những request theo hướng không đồng bộ (non-blocking).
* **Những ứng dụng REST dựa vào API:** JavaScript được sử dụng vào cả frontend lẫn backend của trang đó. Do đó mà một server dễ dàng có thể giao tiếp với frontend qua REST API bằng NodeJS. Thậm chí, NodeJS cũng cung cấp đa package bao gồm Express.js hay Koa để có thể hỗ trợ việc xây dựng ứng dụng web dễ dàng hơn bao giờ hết.

Ảnh có chứa Đồ họa, ảnh chụp màn hình, màu xanh lá cây, thiết kế đồ họa

Mô tả được tạo tự động

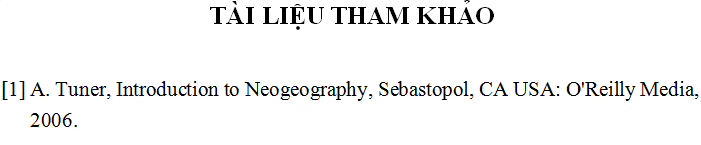
Hình 2.5: Biểu tượng NodeJs

Những sai lầm về Node.JS:

* Node JS là web framework: Node.JS không phải là web framework, nó là một nền tảng cho phép chạy Javascript
* Node JS là một ngôn ngữ lập trình: Node.JS không phải ngôn ngữ lập trình mà chỉ là một môi trường runtime.
* Node JS chỉ sử dụng cho frontend hoặc backend: Không hoàn toàn chính xác, Node.JS có thể sử dụng cho cả hai phía vô cùng tiện lợi.

2.5.2 Ưu nhược điểm của NodeJs

* Ưu điểm
  + Có mô hình I/O hướng sự kiện không đồng bộ (non-blocking), nó cho phép xử lý các yêu cầu đồng thời. Mọi API của Node.JS đều có cho mình tính không đồng bộ, nhờ vậy mà một server dựa trên NodeJS sẽ không cần phải đợi trả về dữ liệu từ API.
  + Sử dụng ngôn ngữ lập trình dễ học là JavaScript. Bên cạnh đó, nó cũng sở hữu tốc độ cực nhanh do xây dựng dựa trên engine JavaScript V8 từ Google Chrome. Do đó mà những thư viện của nó có được khả năng thực thi code chỉ một cách nhanh chóng.
  + Chia sẻ cùng code ở cả phía của client và server.
  + Node Package Manager (NPM) và module Node đang ngày càng phát triển một cách mạnh mẽ. Nó sở hữu hơn 50,000 package khác, những developer có thể dễ dàng trong việc lựa chọn bất kỳ các tính năng nào nhằm xây dựng cho ứng dụng của mình.
  + Sở hữu cộng đồng hỗ trợ tích cực.
  + Cho phép stream những file có kích thước lớn do nó không có buffering. Nhờ vậy mà NodeJS giúp tiết kiệm thời gian cho việc xử lý file khi cần upload các file âm thanh hoặc video. Sở dĩ có tính năng này là vì các ứng dụng không bao giờ buffer dữ liệu mà nó chỉ xuất dữ liệu theo từng phần.
  + Cuối cùng chính là tính đơn luồng: NodeJS sử dụng mô hình mang tính đơn luồng với vòng lặp sự kiện. Nhờ vậy những ứng dụng có thể xử lý được số lượng yêu cầu lớn hơn rất nhiều so với những server truyền thống chẳng hạn như Apache HTTP Server.
* Nhược điểm
  + Đầu tiên phải kể đến chính là nó không có khả năng mở rộng. Do đó mà nó không thể tận dụng được lợi thế mô hình đa lõi ở các phần cứng cấp server trên thị trường hiện nay.
  + Khó thao tác được với cơ sử dữ liệu quan hệ.
  + Mỗi callback của nó sẽ đi kèm với nhiều callback lồng nhau khác.
  + Cần trang bị kiến thức tốt về JavaScript.
  + Không phù hợp với những tác vụ đòi hỏi nhiều CPU.



Hình 2‑13 Chỉnh sửa định dạng hiển thị danh sách các tài liệu tham khảo

## Kết chương

Đoạn này nêu tóm tắt về nội dung của chương vừa viết và viết một vài câu ngắn về nội dung sẽ có trong chương tiếp theo.

# PHÂN TÍCH THIẾT KẾ HỆ THỐNG

# KẾT QUẢ CÀI ĐẶT, THỬ NGHIỆM HOẶC KẾT QUẢ NGHIÊN CỨU

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Tác giả đã làm đồ án này như thế nào? kết quả ra sao? có giống như kết quả đã dự kiến không?

Phần này sinh viên cần bám vào mục tiêu và nội dung đã nêu ở PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP và phần MỞ ĐẦU, đối chiếu với kết quả đã làm để có kết luận phù hợp.

Còn vướng mắc ở chỗ nào? lý do vướng mắc, hướng khắc phục ... (dựa vào đây để các sinh viên khóa sau tham khảo và phát triển đồ án hoặc tránh lặp lại để đỡ mất thời gian)

Kết luận chung

TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
| [1] | A. Tuner, Introduction to Neogeography, Sebastopol, CA USA: O'Reilly Media, 2006. |
| [2] | Hoàng Anh Đức, Lê Văn Hưng, Thiết kế website, Hanoi: NXB Giáo dục Việt Nam, 2016. |

# PHỤ LỤC

Các mã nguồn, các bảng biểu lớn, các phụ lục cho vào mục này. Trong phần đồ án, phần PHỤ LỤC không được tính vào tổng số trang của đồ án tốt nghiệp.

Các mã nguồn sử dụng Style “Code”, ví dụ như sau:

**<?php**

**echo “Hello world!”;**

**?>**