

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN**

TIỂU LUẬN

**“TÌM HIỂU VỀ REACT JS VÀ XÂY DỰNG WEBSITE
THƯƠNG MẠI ĐIỆN TỬ”**

Học phần: 2021COMP104403 – Nhập Môn Công Nghệ Phần Mềm

Nhóm sinh viên thực hiện:

- | | |
|---------------------|------------------|
| 1. Nguyễn Văn Giàu | MSSV: 4501104061 |
| 2. Huỳnh Anh Dự | MSSV: 4501104041 |
| 3. Trương Quốc Hiệp | MSSV: 4501104081 |
| 4. Huỳnh Bảo Lộc | MSSV: 4501104134 |

Giảng viên hướng dẫn: Trần Thanh Nhã

Thành phố Hồ Chí Minh, ngày 18 tháng 09 năm 2021

MỤC LỤC

| | |
|---|----|
| MỞ ĐẦU | 5 |
| Chương 1: CƠ SỞ LÝ THUYẾT | 7 |
| 1.1. Giới thiệu về React JS | 8 |
| 1.2. Một số khái niệm, kiến thức khi nghiên cứu React | 9 |
| 1.2.1. Khái niệm | 9 |
| 1.2.2. Kiến thức | 12 |
| 1.3. Môi trường xây dựng web bằng React JS | 14 |
| 1.3.1. Giới thiệu về Meteor | 14 |
| 1.3.2. Khởi chạy ứng dụng đầu tiên “Hello World” | 15 |
| 1.3.3. Giới thiệu To-do-list app | 17 |
| 1.3.4. Render trong React | 21 |
| 1.4. Kết luận về React: | 21 |
| Chương 2: PHÂN TÍCH WEBSITE THƯƠNG MẠI ĐIỆN TỬ | 22 |
| 2.1. Xác định yêu cầu | 22 |
| 2.2. Phân tích yêu cầu | 23 |
| 2.3. Mô hình hóa yêu cầu | 24 |
| 2.3.1. Mô hình mua hàng | 24 |
| 2.4. Phân tích cơ sở dữ liệu | 25 |
| 2.4.1 Mô hình thực thể kết hợp | 25 |
| 2.4.2. Mô tả các thực thể | 25 |
| 2.4.3. Kiểu dữ liệu của các thực thể | 25 |
| Chương 3. THIẾT KẾ WEBSITE THƯƠNG MẠI ĐIỆN TỬ | 27 |
| 3.1. Sơ đồ giao diện | 27 |
| 3.1.1 Sơ đồ giao diện phía Client | 27 |
| 3.1.2. Sơ đồ giao diện phía Admin | 28 |
| 3.2. Thiết kế giao diện | 28 |
| 3.2.1. Giao diện phía người dùng | 28 |
| 3.2.2 Giao diện phía admin | 32 |
| Chương 5: TỔNG KẾT | 35 |
| 5.1. Kết quả đạt được | 35 |

LỜI CẢM ƠN

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành nhất đến thầy Trần Thanh Nhã. Trong quá trình học tập và tìm hiểu bộ môn Nhập môn công nghệ phần mềm, chúng em đã nhận được sự quan tâm giúp đỡ và hướng dẫn rất tận tình từ tâm huyết của thầy. Thầy đã giúp chúng em tích lũy thêm nhiều kiến thức để có cái nhìn sâu sắc và hoàn thiện hơn trong cuộc sống. Thông qua bài tiểu luận này nhóm chúng xin trình bày lại tất cả những gì mà mình đã đạt được trong thời gian qua.

Có lẽ kiến thức là vô hạn mà sự tiếp nhận kiến thức của bản thân mỗi người luôn tồn tại những hạn chế nhất định. Do đó, trong quá trình hình thành bài tiểu luận chắc chắn sẽ không tránh khỏi những thiếu sót. Nhóm em rất mong nhận được những góp ý từ thầy để bài tiểu luận của nhóm được hoàn thiện hơn.

Kính chúc thầy sức khỏe, hạnh phúc và thành công trên con đường sự nghiệp giảng dạy.

NHIỆM VỤ THÀNH VIÊN NHÓM

| MSSV | Họ Tên | Phân Công Công Việc |
|-------------|------------------|---|
| 4501104061 | Nguyễn Văn Giàu | Làm báo cáo word, xây dựng app To-do-list |
| 4501104041 | Huỳnh Anh Dự | Làm powerpoint, xây dựng sơ đồ tư duy, ứng dụng Hello World |
| 4501104081 | Trương Quốc Hiệp | Thiết kế các mô hình của website thương mại điện tử |
| 4501104134 | Huỳnh Bảo Lộc | Tìm tài liệu |

MỞ ĐẦU

1. Giới thiệu

Trên thế giới hiện nay, công nghệ thông tin là một ngành phát triển không ngừng. Thời kỳ công nghiệp đòi hỏi thông tin nhanh chóng chính xác. Có thể nói tin học đã chiếm một vị trí quan trọng trong cuộc sống hàng ngày, trong các ngành khoa học kỹ thuật. Chúng đang được phát triển và áp dụng rộng rãi đặc biệt là những lĩnh vực kinh doanh, buôn bán.

Việc đưa tin học vào công việc quản lý, nhằm giảm bớt sức lao động của con người, tiết kiệm thời gian, độ chính xác cao và tiện lợi rất nhiều so việc quản lý giấy tờ bằng thủ công như trước đây. Tin học hóa nhằm thu hẹp không gian lưu trữ, tránh được thất lạc dữ liệu, tự động hóa và cụ thể hóa các thông tin theo yêu cầu của con người.

Hiện nay, nhu cầu buôn bán, trao đổi hàng hóa ngày càng tăng và hình thức mua và bán cũng ngày càng được cải tiến. Ở trên một thị trường mà người mua và người bán không cần trực tiếp gặp nhau nhưng vẫn có thể thực hiện các giao dịch thì đòi hỏi cần phải có một môi trường ảo hỗ trợ những công việc đó và những website bán hàng chính là môi trường không thể thuận lợi hơn để thực hiện những giao dịch đó, khiến chúng trở nên đơn giản, nhanh chóng và thuận tiện. Chính vì vậy mà em đã nghiên cứu, tìm hiểu về một thư viện thiết kế trang web phổ biến nhất hiện nay là React và đồng thời xây dựng một website bán hàng.

2. Lý do chọn đề tài

Về phần lý thuyết: React bao gồm React JS và React Native, là một trong những thư viện phổ biến dùng để xây dựng một web application có tính tương tác trực tiếp cao, với phương châm **“Learn One Write Everywhere”** (Học một lần viết mọi nơi) ta cũng thấy được sự thông dụng và tích hợp của React.

Về phần thực hành: Hiện nay, với các công nghệ khác đã phổ biến trước đó như Angular, Developer phải ứng dụng vào rất nhiều các dự án khác nhau. Khi đó, họ phải sử dụng nhiều framework khác nhau, và luôn gặp thách thức code lại từ đầu nếu xuất hiện bất kỳ sự thay đổi nào từ component. Những gì họ muốn khi đó là một framework có thể phá vỡ những cấu trúc phức tạp, cũng như có thể tái sử dụng lại các đoạn code trước kia nhằm giúp project hoàn thành nhanh hơn.

Và ReactJS xuất hiện đúng lúc, cung cấp ngay giải pháp mà họ đang tìm kiếm. Công nghệ này sử dụng JSX (viết tắt của cụm từ JavaScript XML, giống như một dạng cú pháp mở rộng giúp code nhanh và dễ sử dụng hơn). Công nghệ này cực kỳ hữu dụng trong việc cấu trúc các đoạn code sử dụng trong lĩnh vực trí tuệ nhân tạo, hay lồng ghép các component vào một file chỉ cần xác thực một lần.

Ngày nay, ReactJS đang ngày một phổ biến hơn vì tính đơn giản và linh hoạt. Nhiều người còn cho rằng, đây sẽ là tương lai của ngành web development. Người ta ước tính có khoảng hơn 1.300 developer và hơn 94.000 website đang sử dụng ReactJS.

Sự phổ biến của ReactJS một phần nhờ vào các ông trùm về công nghệ như Facebook, PayPal, Uber, Instagram và Airbnb, sử dụng để giải quyết các vấn đề liên quan đến UI. Từ đó kéo theo làn sóng rất nhiều người và công ty tiếp tục tin tưởng sử dụng ReactJS.

3. Mục tiêu và nhiệm vụ nghiên cứu

- Tìm hiểu lý thuyết về React JS
- Xác định cấu trúc, các thành phần chính trong React
- Hệ thống hóa các kiến thức đã tìm hiểu được thành sơ đồ tư duy
- Xây dựng, thực thi và chạy được phần mềm React JS
- Áp dụng những kiến thức nghiên cứu được để ứng dụng vào thực tế
- Biết được cách thức để xây dựng 1 website thương mại điện tử

4. Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu:

- React JS
- Website thương mại điện tử
- React Bootstrap và các Component khác

Tìm hiểu khái quát về React, nắm được các định nghĩa, khái niệm về React, cách render, import trong react và các thuộc tính chủ chốt như prop, state,...

Đề tài thiết kế website thương mại điện tử được giới hạn ở việc khách hàng lựa chọn sản phẩm, thêm vào giỏ và tiến hành điền thông tin khách hàng để gửi hóa đơn cho bên admin. Về phía admin sẽ tiếp nhận đơn hàng, thực hiện giao hàng đồng thời cập nhật, chỉnh sửa giao diện website bán hàng của mình

5. Phương pháp nghiên cứu

Nghiên cứu thông qua các tài liệu như Ebook và các trang web đào tạo trực tuyến như Youtube, Blog:

- Youtube:

- + <https://www.youtube.com/watch?v=Ke90Tje7VS0>

- + <https://www.youtube.com/watch?v=w7ejDZ8SWv8>

- + <https://www.youtube.com/watch?v=4UZrsTqkcW4>

- + https://www.youtube.com/watch?v=gL5HBA_1bDQ&list=PLJ5qtRQovuEOoKffoCBzTfvzMTTORnoyp

- + https://www.youtube.com/watch?v=OH4msPNM2CI&list=PLQg6GaokU5CyvExiaMgXP_BS5WWNBfZJN

- Blog:

- + <https://reactjs.org>

- + <https://www.w3schools.com/REACT/DEFAULT.ASP>

- + <https://redux.js.org/tutorial/reactjs/>

- + <https://viblo.asia>

- + <https://hocwebchuan.com/tutorial/reactjs/>

- Ebook:

- + The Road to React: Your journey to master plain yet pragmatic React.js

- + ReactJS by Example – Building Modern Web Applications with React

- + Lập trình React thật đơn giản By VNTalking

- + React in Action by Mark Tielens Thomas

- + React.js Essentials by Artemij Fedosejev

Chương 1: CƠ SỞ LÝ THUYẾT

1.1. Giới thiệu về React JS

React JS là thư viện Javascript được xây dựng bởi các kỹ sư của Facebook, đang được rất nhiều công ty nổi tiếng sử dụng để phát triển các sản phẩm của họ như Yahoo, Airbnb và tất nhiên là trong nội tại Facebook, Instagram. Nó phù hợp với các dự án lớn có tính mở rộng hơn là các dự án nhỏ.

- Đặc tính của React JS

Tư tưởng React JS là xây dựng lên các **components** có tính tái sử dụng, dễ dàng cho việc chia nhỏ vấn đề, testing. Nó giúp chúng ta dễ dàng quản lý, mở rộng hệ thống, điều này nếu là Angular JS thì đòi hỏi cấu trúc, cách viết code phải thật tối ưu.

React JS luôn giữ các components ở trạng thái stateless (nhiều nhất có thể) khiến ta dễ dàng quản lý bởi nó chẳng khác gì một trang static HTML. Bản thân các components này không có trạng thái (state) nó nhận đầu vào từ bên ngoài và chỉ hiện thị ra dựa vào các đầu vào đó, điều này lý giải tại sao nó lại mang tính tái sử dụng (reuse) và dễ dàng cho việc test.

- Thế mạnh của React JS

React JS là một framework hiển thị view chú ý đến hiệu năng (performance-minded). Rất nhiều đối thủ nặng ký về framework MVVM (Model-View-ViewModel) mất một thời gian lớn để hiển thị những lượng data lớn, như trong trường hợp những danh sách (list) và tương tự. Nhưng React đó không còn là vấn đề, vì nó chỉ hiển thị những gì thay đổi.

Một trong những điểm mạnh nữa của React JS là virtual DOM - thứ nằm ẩn bên trong mỗi view và là lí do khiến cho React đạt được hiệu năng tốt. Khi một view yêu cầu gọi, tất cả mọi thứ sẽ được đưa vào trong một bản sao ảo của DOM. Sau khi việc gọi hoàn thành, React tiến hành một phép so sánh giữa DOM ảo và DOM thật, và thực hiện những thay đổi được chỉ ra trong phép so sánh trên.

Ví dụ: nếu chúng ta đang xem một danh sách có 20 products được hiển thị bởi React, và chúng ta thay đổi product thứ 2, thì chỉ product đó được hiển thị lại, và 19 products còn lại vẫn giữ nguyên (không cần hiển thị lại hay reload lại trang). React đã dùng cái gọi là “DOM ảo” (“virtual DOM”) để tăng hiệu năng bằng cách xuất ra một

hiển thị ảo, sau đó kiểm tra sự khác biệt giữa hiển thị ảo và những gì có trên DOM và tạo một bản vá (a patch).

1.2. Một số khái niệm, kiến thức khi nghiên cứu React

1.2.1. Khái niệm

- Component

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Ta có thể tạo ra một component bằng cách extend Component như sau

```
class Welcome extends React.Component {  
  render() {  
    return <h1>Hello, {this.props.name}</h1>;  
  }  
}
```

Phương thức quan trọng nhất là render, phương thức này được trigger khi component đã sẵn sàng để được render lên trên page.

Trong hàm đó, bạn sẽ trả về một mô tả cho việc bạn muốn React render cái gì lên trên page. Như trong ví dụ ở trên, đơn giản mình muốn render 1 thẻ h1 chứa lời chào và tên.

Hàm render chính là mô tả cụ thể của UI tại bất cứ thời điểm nào. Vì thế nếu dữ liệu thay đổi, React sẽ take care việc update UI với dữ liệu tương ứng. Các bạn có thể hiểu đơn giản là, khi dữ liệu thay đổi, React sẽ tự động gọi hàm render để update lại UI.

- Multiple Components

Nếu muốn lồng nhiều component vào nhau, ta sẽ làm điều này trong lệnh return của phương thức render.

```

class Welcome extends React.Component {
  render() {
    return <div>Hello: {this.props.name}<Date date='2010-10-10' /></div>;
  }
}
class Date extends React.Component {
  render() {
    return <div>Date: {this.props.date}</div>;
  }
}
const element = <Welcome name="Sara" />;
ReactDOM.render(
  element,
  document.getElementById('root')
);

```

- Prop

Props ở đây chính là properties của một component, chúng ta có thể thay đổi props của component bằng cách truyền dữ liệu từ bên ngoài vào. Props có thể là 1 object, function, string, number...

```

import PropTypes from 'prop-types';

class Greeting extends React.Component {
  render() {
    return (
      <h1>Hello, {this.props.name}</h1>
    );
  }
}

Greeting.propTypes = {
  name: PropTypes.string
};

```

Chú ý: Khi một props được truyền vào component thì nó là bất biến tức là dữ liệu của nó không được thay đổi kiểu như một "pure" function.

Ví dụ về “pure” function và “not pure” function

Ta xét function như dưới:

```
function sum(a, b) {  
  return a + b;  
}
```

Function được gọi là "pure" vì nó không làm thay đổi giá trị đầu vào của nó và luôn trả về một kết quả tương tự cho các đầu vào như nhau.

Ta xét function thứ 2 như hình dưới

```
function sum(sum, a) {  
  sum += a;  
}
```

Ở đây function trên đã thay đổi chính giá trị sum đầu vào của nó và điều này khiến nó không là "pure" function

⇒ **Tất cả các component của react phải hoạt động như "pure" function**

- State

Một component trong React có 2 cách để lấy thông tin props và state. Không như props, state của 1 component không được truyền từ bên ngoài vào. Một component sẽ tự quyết định state của chính nó. Để tạo ra state cho component, ta sẽ phải khai báo nó trong hàm constructor

```

class Clock extends React.Component {
  constructor(props) {
    super(props);
    this.state = {date: new Date()};
  }

  render() {
    return (
      <div>
        <h1>Hello, world!</h1>
        <h2>It is {this.state.date.toLocaleTimeString()}.</h2>
      </div>
    );
  }
}

```

- Sự giống và khác nhau giữa Prop và State

Giống:

- Props và state đều là plain JS objects
- Props và state đều trigger render update khi thay đổi

Khác:

| | Prop | State |
|--|------|-------|
| Can get initial value from parent Component? | Yes | Yes |
| Can be changed by parent Component? | Yes | No |
| Can set default values inside Component?* | Yes | Yes |
| Can change inside Component? | No | Yes |
| Can change in child Components? | Yes | No |

1.2.2. Kiến thức

React JS là một framework đơn nhất, nhưng để xây dựng được ứng dụng hoàn chỉnh, chúng ta cần nhiều thứ hơn nữa. Đây là một số kiến thức khi muốn nghiên cứu React:

- Javascript:

Ngôn ngữ xây dựng nên React, yêu cầu mức độ trung bình: các khái niệm “object”, “prototype”, “callback” là bắt buộc, nếu bạn chưa biết gì, tôi nghĩ bạn nên đọc tài liệu tại đây và thực hành trong ít nhất 3 tuần trước khi quay lại để bắt đầu.

- Npm:

Công cụ quản lý package của Node.js và là phương pháp phổ biến nhất để các lập trình viên front-end và các nhà thiết kế chia sẻ mã nguồn Javascript. Nó bao gồm một hệ thống quản lý module gọi là CommonJS và cho phép bạn cài đặt bất cứ công cụ command-line nào được viết bằng Javascript.

Hầu hết các components, thư viện và công cụ tái sử dụng trong hệ sinh thái của React đều được triển khai như là thành các module CommonJS và có thể cài đặt thông qua npm.

- JavaScript “bundlers”

Hay còn gọi là các công cụ đóng gói JavaScript. Vì một vài lý do kỹ thuật mà module CommonJS (ví dụ, các thư viện trên npm) không thể sử dụng native trên trình duyệt. Bạn cần có một “công cụ đóng gói” Javascript để “đóng gói” các modules này thành các files .js mà có thể include được trên trang web qua thẻ <script>.

- ES6:

Là phiên bản mới nhất của Javascript, vì còn quá mới nên chưa được hỗ trợ trên các trình duyệt, nhưng các công cụ đóng gói sẽ thực hiện phiên dịch với cấu hình đúng để có thể sử dụng được trên trình duyệt.

- Routing

“Các ứng dụng single-page” trở nên khá rầm rộ ngày nay. Đây là những trang web mà chỉ thực hiện khởi tạo một lần, và khi người dùng ấn vào một đường dẫn hay nút ấn thì Javascript trên trang web sẽ thực thi mà trang web không cần phải reload lại. Việc quản lý địa chỉ trên thanh địa chỉ được thực hiện bởi **router**.

Router được sử dụng phổ biến nhất trong hệ sinh thái React là react-router. Nếu như bạn đang xây dựng một ứng dụng single-page, hãy sử dụng nó trừ khi bạn có lý do mà không thể sử dụng.

- **Server Rendering**

Server rendering thường được gọi là “toàn bộ” hay “đồng bộ” JS. Điều đó có nghĩa là bạn có thể lấy bất cứ React components nào và render chúng thành HTML tĩnh trên server. Điều này cải thiện được tốc độ ban đầu vì người dùng không cần phải chờ JS thực hiện download để có thể thấy được UI ban đầu, và React có thể tái sử dụng HTML đã được render phía server, vì vậy mà không cần xử lý ở phía client.

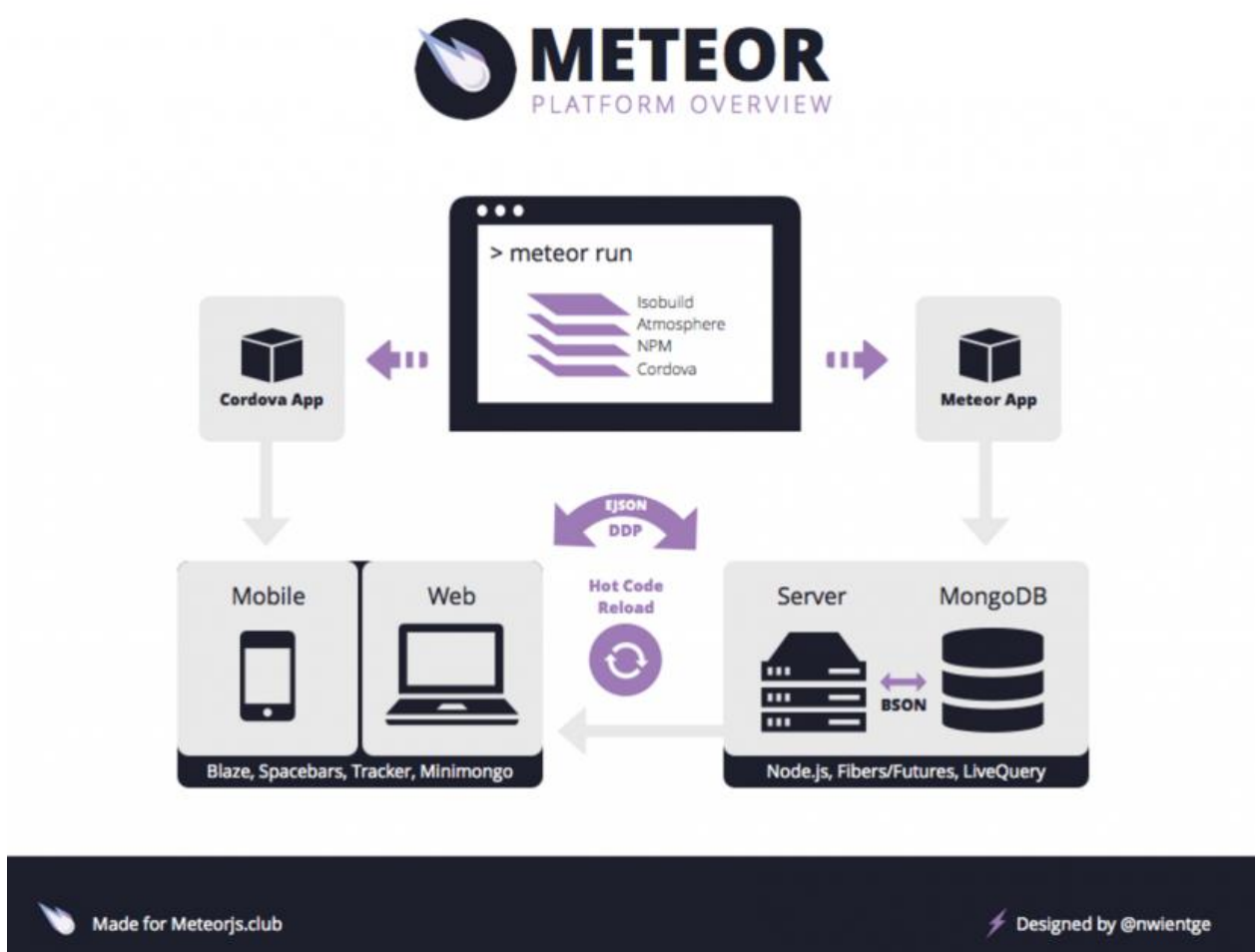
Chỉ cần sử dụng server rendering khi ta thấy việc render ban đầu trở nên quá chậm hoặc muốn cải thiện thứ hạng tìm kiếm. Mặc dù Google bây giờ đã thực hiện index các nội dung render phía client, nhưng tại thời điểm tháng một 2016, mỗi khi được lấy ra đo lường thì thấy rõ thứ hạng bị ảnh hưởng đi xuống, có khả năng là do vi phạm về tốc độ render phía client.

Việc render phía server vẫn cần nhiều công cụ để làm cho nó trở nên chính xác. Vì việc này mặc định được hỗ trợ trong các React components mà không cần quan tâm tới việc render phía server, chúng ta nên xây dựng ứng dụng trước và quan tâm tới việc render phía server sau. Ta sẽ không cần thiết phải viết lại tất cả các components để xử lý việc đó.

1.3. Môi trường xây dựng web bằng React JS

1.3.1. Giới thiệu về Meteor

Meteor là một nền tảng được xây dựng dựa trên môi trường node.js cho phép tạo ra các ứng dụng web theo thời gian thực. Nó đảm bảo việc đồng bộ thông tin giữa cơ sở dữ liệu của ứng dụng và giao diện người dùng. Vì được xây dựng trên nền Node.js nên Meteor sử dụng JavaScript trên cả máy khách và máy chủ.



Meteor không chỉ là 1 [Javascript](#) framework mà là cả **1 hệ sinh thái** (ecosystem). Meteor cũng cấp cho ta “gần như” toàn bộ những thứ để làm một ứng dụng web/mobile:

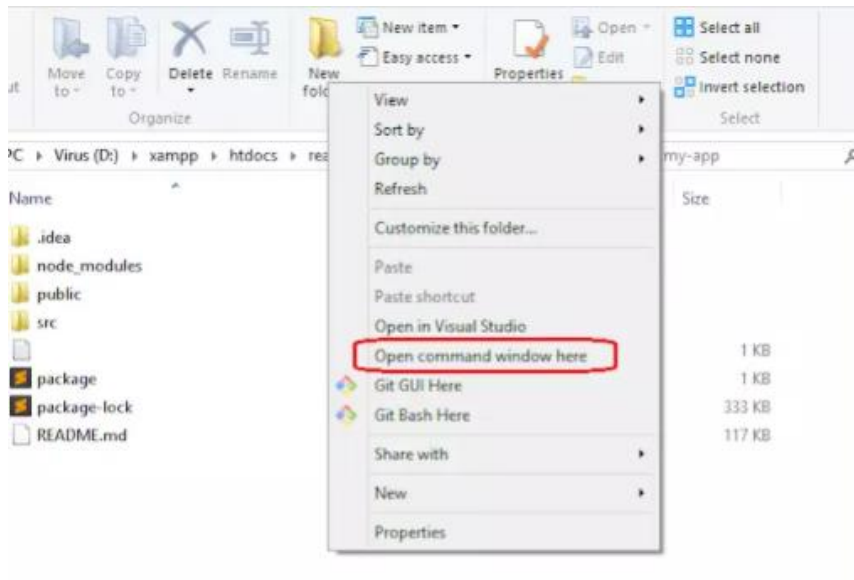
- Phần front-end sử dụng framework Blaze để bind dữ liệu.
- Phần back-end sử dụng NodeJS và Express để làm server, MongoDB là database chính. Bạn không cần biết gì về NodeJS hay Express, chỉ cần code Meteor bằng javascript thôi.
- Một hệ thống thư viện/module tương tự npm, Maven, Nuget.

Vì sử dụng trên nền JavaScript nên có thể nói việc kết hợp Meteor và React JS sẽ giúp người dùng có thể tự viết một trang web real-time lẫn back-end và front-end. Và trong quyển khóa luận này, em xin được sử dụng Meteor tạo một web application kết hợp với React để tìm hiểu rõ hơn về cách hoạt động của React.

1.3.2. Khởi chạy ứng dụng đầu tiên “Hello World”

Đầu tiên các ta cần cài đặt Node.js để tạo môi trường chạy React, việc cài đặt rất đơn giản và nhanh chóng.

Tạo 1 folder để chứa project React, vào trong folder đề shift nhấn chuột phải và chọn như hình.



Tiếp theo gõ các dòng lệnh sau để tạo project

```
npm install -g create-react-app
```

```
create-react-app my-app
```

NPM sẽ tự động tạo cho ta 1 project tên là my-app và install các module và lib cần thiết cho chúng ta.

Vào thư mục src xóa hết các file trong đây, lưu ý không xóa cả thư src mà chỉ xóa các file ở trong đó.

Sau đó tạo mới 2 file Index.js và App.jsx.

Trong App.jsx ta thực hiện code như sau: import React from 'react'

```
class App extends React.Component {  
  render() {  
    return (  
      <div>  
        Hello World!!!  
      </div>  
    );  
  }  
}  
export default App;
```

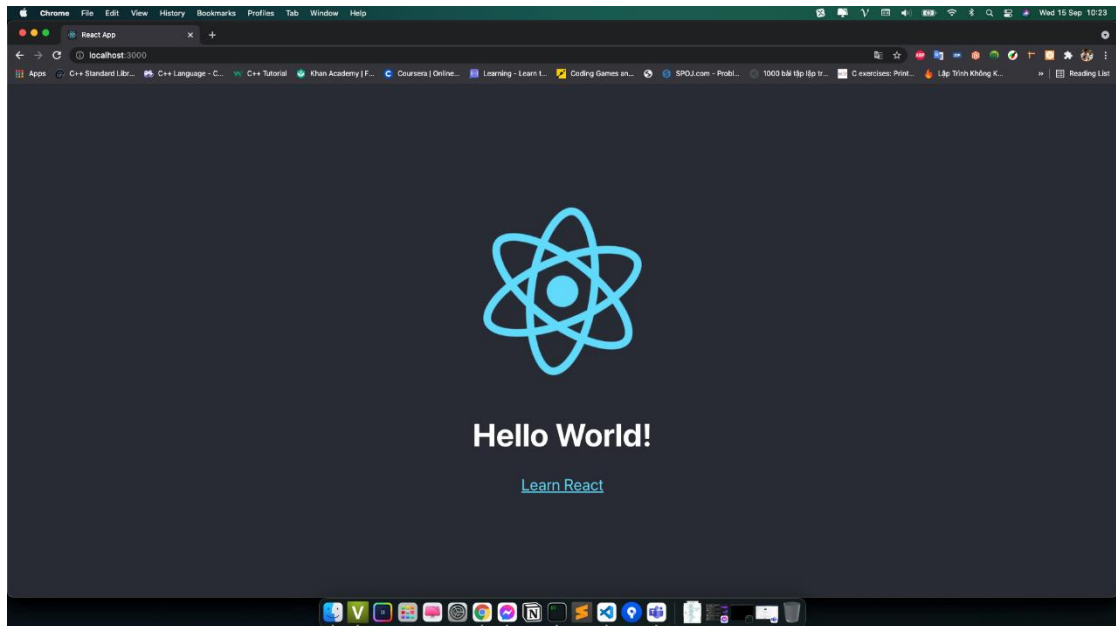
Đây là React component đầu tiên của chúng ta, chúng ta sẽ tìm hiểu component là gì ở các bài sau, component này là App sẽ render Hello World ra màn hình. Tiếp theo trong index.js ta code:

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App.jsx';

ReactDOM.render(<App />, document.getElementById('root'));
```

Running the Server: `npm start`

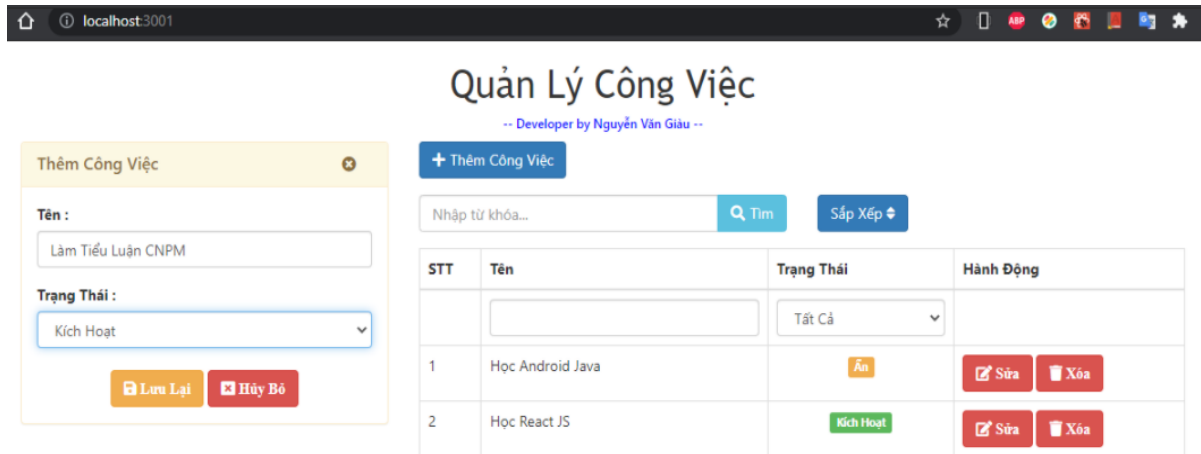
Và đây là kết quả



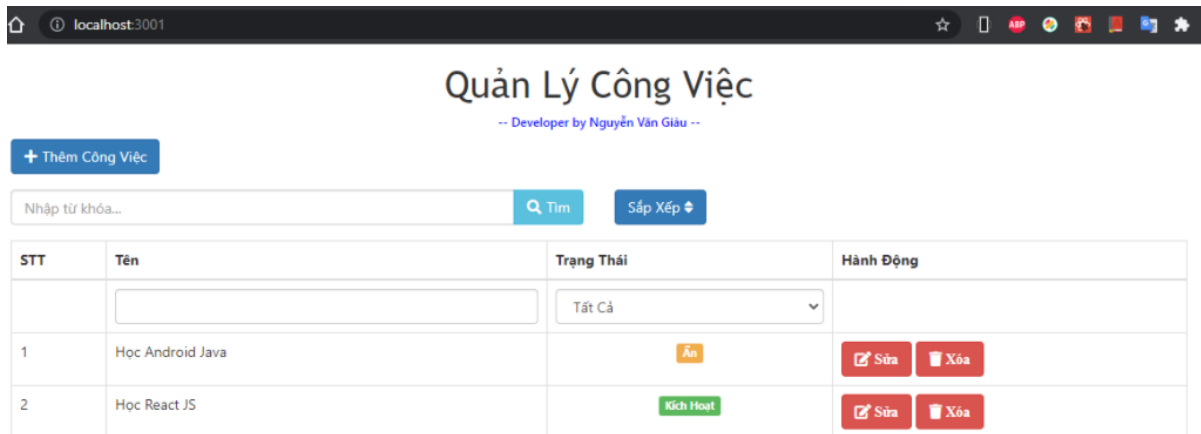
1.3.3. Giới thiệu To-do-list app

To-do-list app là một ví dụ điển hình về cách hoạt động của Meteor và React, nó cho ta thấy được cách thức làm việc, cách cấu tạo thư mục trong meteor và cách sử dụng các hàm trong thư viện React để tạo một trang web real-time ổn định nhất

Giao diện



Cửa sổ “thêm công việc” có thể đóng mở linh hoạt như hình dưới:



Phân tích code:

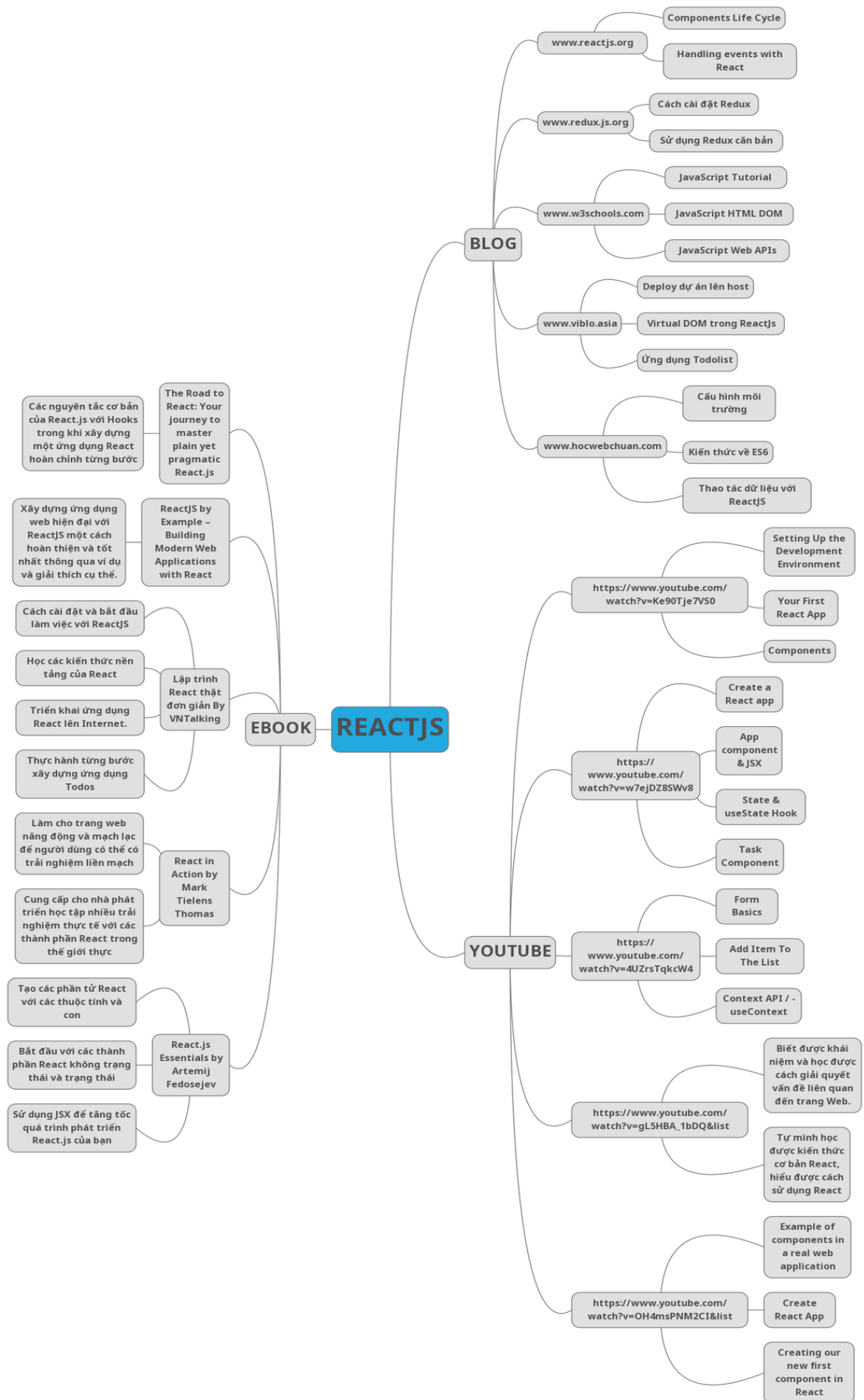
Các component được phân chia nhỏ để thuận lợi cho việc quản lý cũng như tối ưu được tối độ chạy của web mỗi khi render

```
JS App.js X JS Control.js JS Sort.js JS Search.js JS TaskList.js JS TaskForm.js JS TaskItem.js
src > JS App.js > App > render
193 return (
194   <div>
195     <div className="container">
196       <div className="text-center">
197         <h1>Quản Lý Công Việc</h1>
198         <h6>-- Developer by Nguyễn Văn Giàu --</h6>
199       </div>
200       <div className="row">
201         <div className={isDisplayForm ? 'col-xs-4 col-sm-4 col-md-4 col-lg-4' : ''}>
202           {elementTaskForm}
203         </div>
204         <div className={isDisplayForm ? 'col-xs-8 col-sm-8 col-md-8 col-lg-8' : 'col-xs-12 col-sm-12 col-md-12'}>
205           <button type="button" className="btn btn-primary" onClick={this.onToggleForm}>
206             <i className="fa fa-plus mr-5"></i>Thêm Công Việc
207           </button>
208           <Control onSearch={this.onSearch}
209             onSort={this.onSort}
210           />
211           <div className="row mt-15">
212             <div className="col-xs-12 col-sm-12 col-md-12 col-lg-12">
213               <TaskList
214                 tasks = {tasks}
215                 onDelete={this.onDelete}
216                 onUpdate={this.onUpdate}
217                 onUpdateStatus={this.onUpdateStatus}
218                 onFilter={this.onFilter}/>
219             </div>
220           </div>
221         </div>
222       </div>
223     </div>
224   </div>

```

Sourcecode To-do-list app: <https://github.com/vangiaurecca/ToDoList.git>

➔ Sơ đồ tư duy hệ thống kiến thức từ tài liệu nghiên cứu



1.3.4. Render trong React

Render trong React có thể hiểu là phần giao diện mà component đó chạy trên trang web đó. Một trang web react có thể import nhiều component, mỗi component sẽ hiển thị 1 phần nhỏ trong giao diện được hiện lên ấy.

Trở lại ví dụ trên, ở trang chính của To-do-list app ta đã “import” các file như Item.jsx, ItemProgress.jsx... và bản thân từng file này sẽ render cho người dùng đúng chức năng của nó, không render cả trang web.

Ví dụ như file ItemProgress.jsx sẽ render thành tiến trình hình dưới



Tương tự như vậy, ta có thể viết nhiều component nhỏ khác nhau, với từng component sẽ render từng thành phần nhỏ trong trang web, tập hợp những component đó ta sẽ tạo ra được 1 trang hoàn chỉnh.

1.4. Kết luận về React:

Ưu và nhược điểm của React

Ưu điểm:

- Reactjs cực kì hiệu quả: Reactjs tạo ra cho chính nó DOM ảo – nơi mà các component thực sự tồn tại trên đó. Điều này sẽ giúp cải thiện hiệu suất rất nhiều. Reactjs cũng tính toán những thay đổi nào cần cập nhật lên DOM và chỉ thực hiện chúng. Điều này giúp Reactjs tránh những thao tác cần trên DOM mà nhiều chi phí.
- Reactjs giúp việc viết các đoạn code JS dễ dàng hơn: Nó dùng cú pháp đặc biệt là JSX (Javascript mở rộng) cho phép ta trộn giữa code HTML và Javascript. Ta có thể thêm vào các đoạn HTML vào trong hàm render mà không cần phải nối chuỗi. Đây là đặc tính thú vị của Reactjs. Nó sẽ chuyển đổi các đoạn HTML thành các hàm khởi tạo đối tượng HTML bằng bộ biến đổi JSX.
- Nó có nhiều công cụ phát triển: Khi bạn bắt đầu Reactjs, đừng quên cài đặt ứng dụng mở rộng của Chrome dành cho Reactjs. Nó giúp bạn debug code dễ

dàng hơn. Sau khi bạn cài đặt ứng dụng này, bạn sẽ có cái nhìn trực tiếp vào virtual DOM như thể bạn đang xem cây DOM thông thường.

- Render tầng server: Một trong những vấn đề với các ứng dụng đơn trang là tối ưu SEO và thời gian tải trang. Nếu tất cả việc xây dựng và hiển thị trang đều thực hiện ở client, thì người dùng sẽ phải chờ cho trang được khởi tạo và hiển thị lên. Điều này thực tế là chậm. Hoặc nếu giả sử người dùng vô hiệu hóa Javascript thì sao? Reactjs là một thư viện component, nó có thể vừa render ở ngoài trình duyệt sử dụng DOM và cũng có thể render bằng các chuỗi HTML mà server trả về.
- Làm việc với vấn đề test giao diện: Nó cực kì dễ để viết các test case giao diện vì virtual DOM được cài đặt hoàn toàn bằng JS.
- Hiệu năng cao đối với các ứng dụng có dữ liệu thay đổi liên tục, dễ dàng cho bảo trì và sửa lỗi.

Nhược điểm:

- Reactjs chỉ phục vụ cho tầng View. React chỉ là View Library nó không phải là một MVC framework như những framework khác. Đây chỉ là thư viện của Facebook giúp render ra phần view. Vì thế React sẽ không có phần Model và Controller, mà phải kết hợp với các thư viện khác. React cũng sẽ không có 2-way binding hay là Ajax
- Tích hợp Reactjs vào các framework MVC truyền thống yêu cầu cần phải cấu hình lại.
- React khá nặng nếu so với các framework khác, React có kích thước tương đương với Angular (Khoảng 35kb so với 39kb của Angular). Trong khi đó Angular là một framework hoàn chỉnh
- Khó tiếp cận cho người mới học Web

Chương 2: PHÂN TÍCH WEBSITE THƯƠNG MẠI ĐIỆN TỬ

2.1. Xác định yêu cầu

Khách hàng:

- Chọn lựa sản phẩm

- Xem giá
- Tìm kiếm sản phẩm
- Mua hàng
- Cung cấp thông tin
- Thanh toán sản phẩm

Chủ cửa hàng (admin)

- Tùy chỉnh website
- Thêm xóa sửa sản phẩm
- Cập nhật sản phẩm nổi bật, sản phẩm mới
- Thay đổi banner, slideshow hình ảnh trang web
- Đăng thông tin về cửa hàng, thông báo...
- Quản lý hóa đơn khách hàng
- Xem thống kê các hóa đơn đã lập

2.2. Phân tích yêu cầu

Chủ cửa hàng:

Cung cấp sản phẩm: Cung cấp các sản phẩm đồ chơi theo chủng loại. Các mặt hàng được chia theo từng chủng loại khác nhau. Mỗi mặt hàng tương ứng với 1 chủng loại, nếu mặt hàng nằm ngoài những chủng loại đã có thì chủ cửa hàng có thể thêm chủng loại mới. Tạo ra sự đa dạng trong mặt hàng

Chỉnh sửa giao diện: Chủ cửa hàng có thể tùy chỉnh giao diện trang web, thay đổi banner, slide show hình ảnh. Thêm sản phẩm nổi bật cũng như post những bài thông báo, hướng dẫn mua hàng.

Quản lý hóa đơn: Xem những hóa đơn mà khách hàng đã gửi. Chỉnh sửa nếu có nhu cầu thay đổi từ khách hàng. Xóa nếu khách hàng hủy đơn hàng.

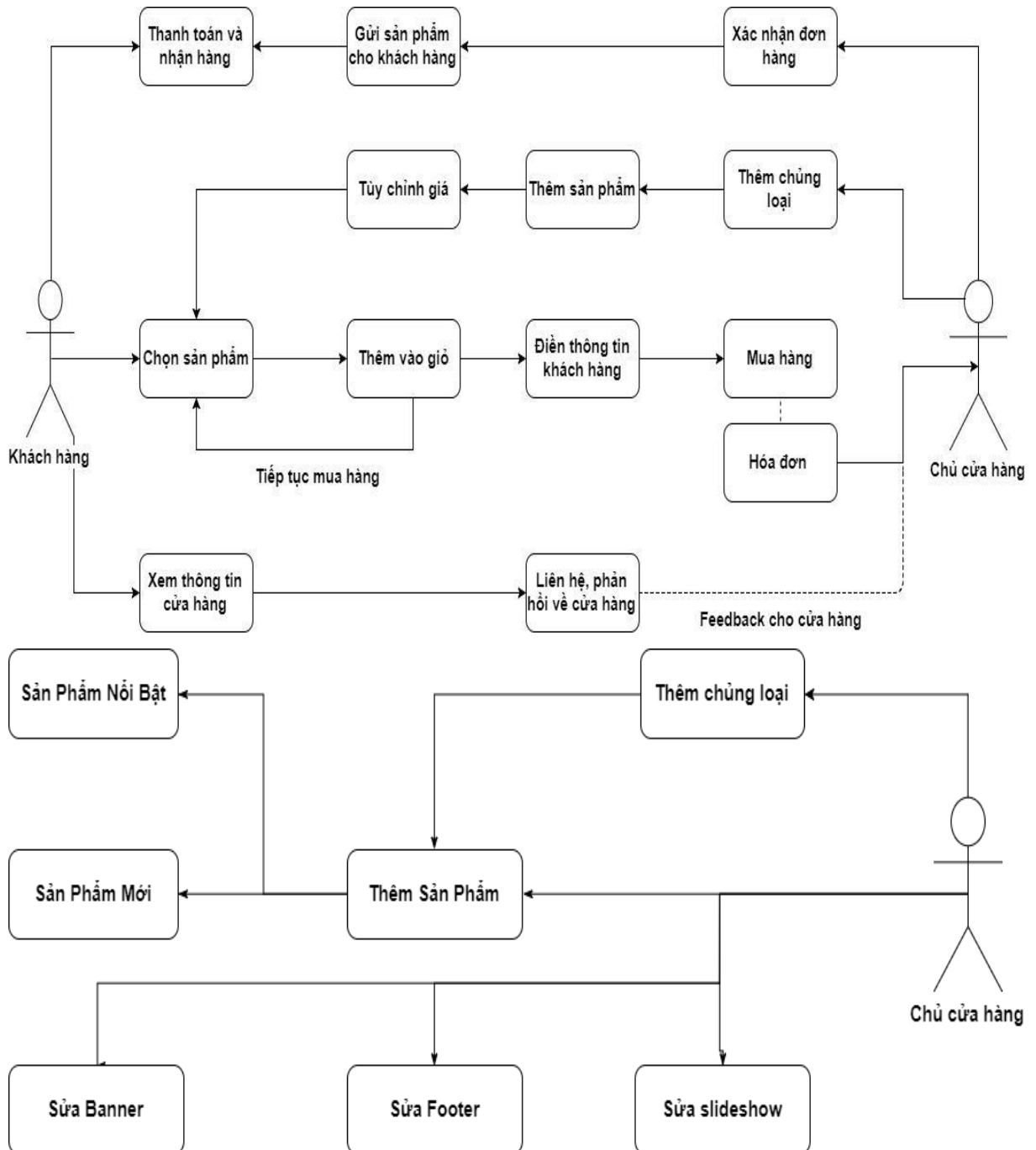
Khách hàng:

Tìm kiếm, lựa chọn sản phẩm từ cửa hàng cung cấp. Để tiến hành mua hàng, khách hàng cho sản phẩm vào giỏ, chỉnh số lượng mặt hàng. Cung cấp thông tin cho cửa hàng. Sau khi gửi đơn hàng, cửa hàng sẽ gọi điện xác nhận và giao tận nhà cho khách qua hình thức COD.

Ngoài ra khách hàng có thể xem thông tin về cửa hàng, liên hệ đóng góp ý kiến cũng như phản hồi về các sản phẩm đã mua.

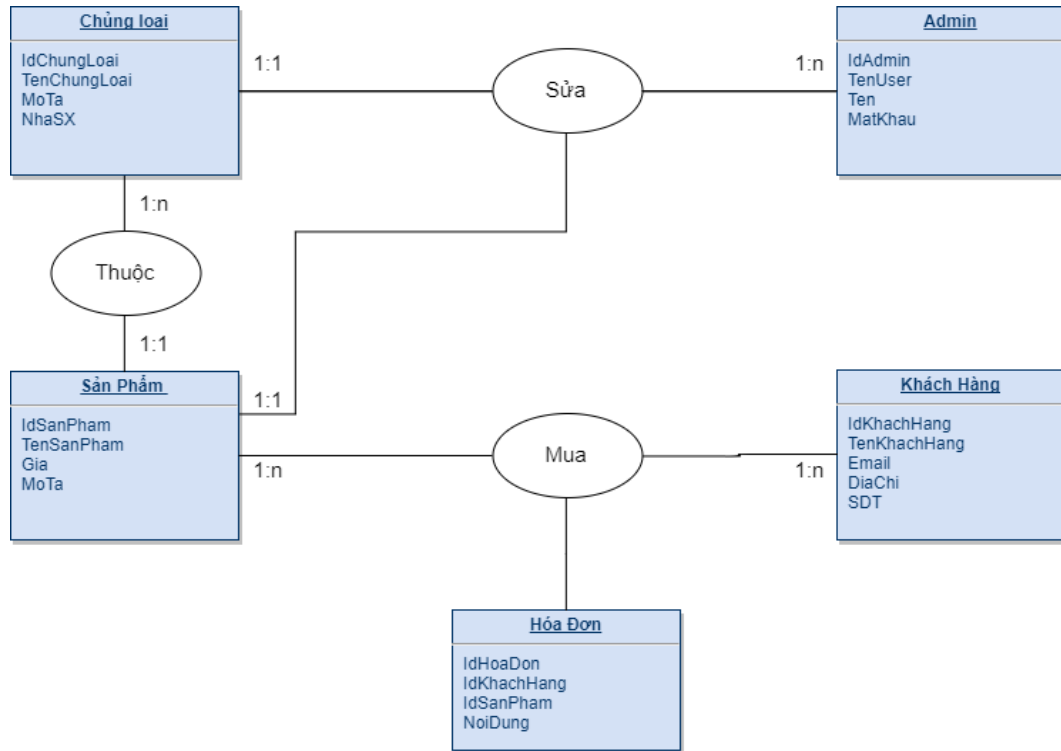
2.3. Mô hình hóa yêu cầu

2.3.1. Mô hình mua hàng



2.4. Phân tích cơ sở dữ liệu

2.4.1 Mô hình thực thể kết hợp



2.4.2. Mô tả các thực thể

ADMIN (**IDADMIN**, TENUSER, TEN, MATKHAU);

KHACHHANG (**IDKHACHHANG**, TENKHACHHANG, EMAIL, DIACHI, SDT)

CHUNGLOAI (**IDCHUNGLOAI**, TENCHUNGLOAI, MOTA, NHASX)

SANPHAM(**IDSANPHAM**, TENSANPHAM, GIA, MOTA, **IDCHUNGLOAI**)

HOADON(**IDKHACHHANG**, **IDSANPHAM**, NOIDUNG)

2.4.3. Kiểu dữ liệu của các thực thể

Admin

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|------------|--------------|--|
| IDADMIN | INT | Khóa chính của bảng admin, tự động tăng khi thêm mới |
| TENUSER | VARCHAR | Tên dùng để đăng nhập, không dấu không khoảng cách |
| TEN | VARCHAR | Tên hiển thị trong phần quản lí |
| MATKHAU | VARCHAR | Mật khẩu đăng nhập vào trang quản trị |

Khách Hàng

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|--------------|--------------|---|
| IDKHACHHANG | INT | Khóa chính của bảng khách hàng, tự động tăng khi thêm mới |
| TENKHACHHANG | VARCHAR | Tên khách hàng |
| EMAIL | VARCHAR | Email khách hàng dùng để xác nhận đơn hàng |
| DIACHI | VARCHAR | Địa chỉ khách hàng dùng khi gửi hàng |
| SDT | VARCHAR | Số điện thoại liên lạc của khách hàng |

Chủng Loại

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|--------------|--------------|---|
| IDCHUNGLOAI | INT | Khóa chính của bảng chủng loại, tự động tăng khi thêm mới |
| TENCHUNGLOAI | VARCHAR | Tên chủng loại |
| MOTA | VARCHAR | Miêu tả chủng loại sản phẩm |
| NHASX | VARCHAR | Tên nhà sản xuất |

Sản Phẩm

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|-------------|--------------|---|
| IDSANPHAM | INT | Khóa chính của bảng sản phẩm, tự động tăng khi thêm mới |
| TENSANPHAM | VARCHAR | Tên sản phẩm |
| MOTA | VARCHAR | Miêu tả sản phẩm |
| GIA | VARCHAR | Giá bán |
| IDCHUNGLOAI | INT | Khóa phụ, tham chiếu đến bảng Chủng Loại |

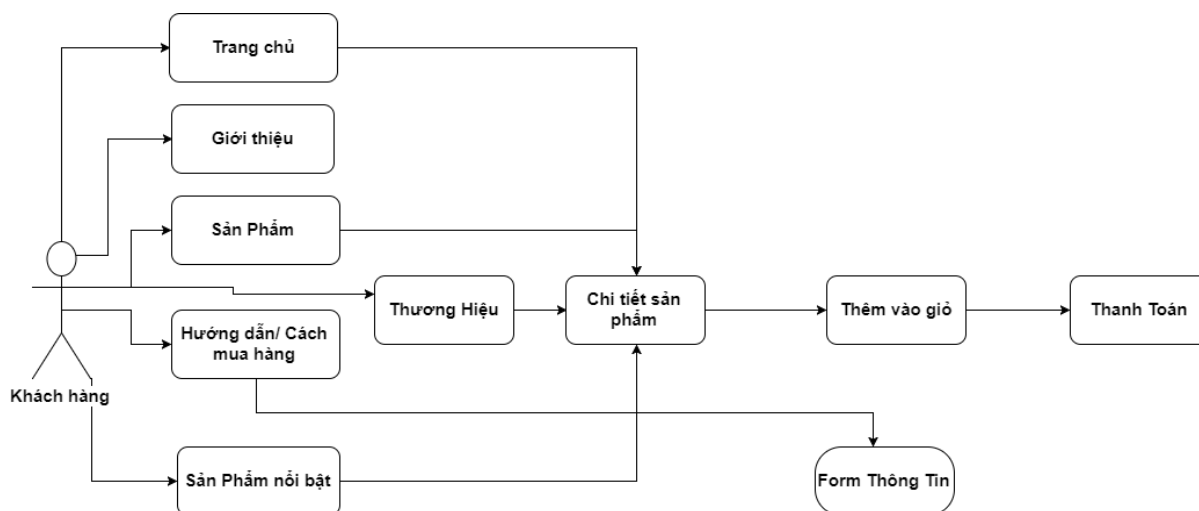
Hóa Đơn

| Thuộc tính | Kiểu dữ liệu | Mô tả |
|-------------|--------------|-------------------------------------|
| IDKHACHHANG | INT | Cùng là khóa chính của bảng hóa đơn |
| IDSANPHAM | INT | |
| NOIDUNG | VARCHAR | Tin nhắn mà người mua gửi |

Chương 3. THIẾT KẾ WEBSITE THƯƠNG MẠI ĐIỆN TỬ

3.1. Sơ đồ giao diện

3.1.1 Sơ đồ giao diện phía Client



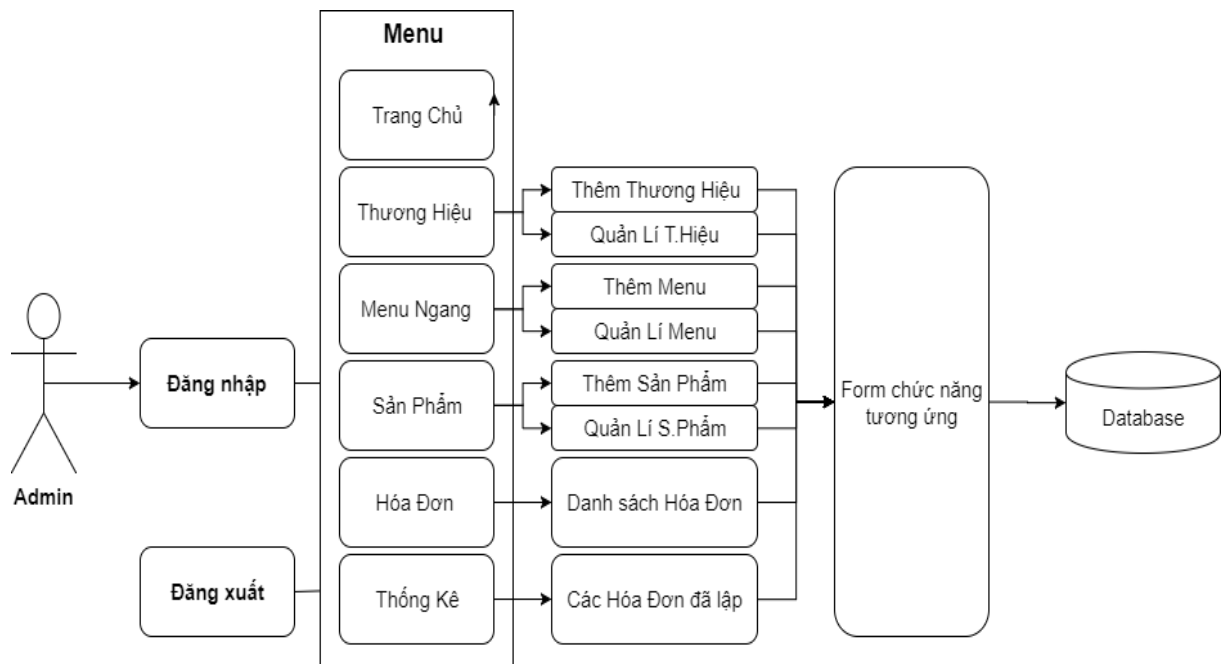
Mô tả:

Khách hàng khi đến với trang web có thể lựa chọn sản phẩm ở ngay “Trang Chủ” hoặc vào danh mục “Sản Phẩm” để tìm món đồ chơi mình muốn mua. Ngoài ra, khu vực những “Sản Phẩm Nổi Bật” cho phép khách hàng truy cập nhanh vào sản phẩm đó.

Website còn có danh mục sản phẩm được lọc theo chủng loại (Thương hiệu). Bằng cách chọn vào “Thương Hiệu” những sản phẩm của dòng thương hiệu đã chọn sẽ xuất hiện

Phần “Hướng Dẫn” và “Cách Mua Hàng” là 2 trang thông tin hướng dẫn khách hàng cách thức chọn mua, gửi đơn hàng và cách thanh toán đơn hàng.

3.1.2. Sơ đồ giao diện phía Admin



Mô tả:

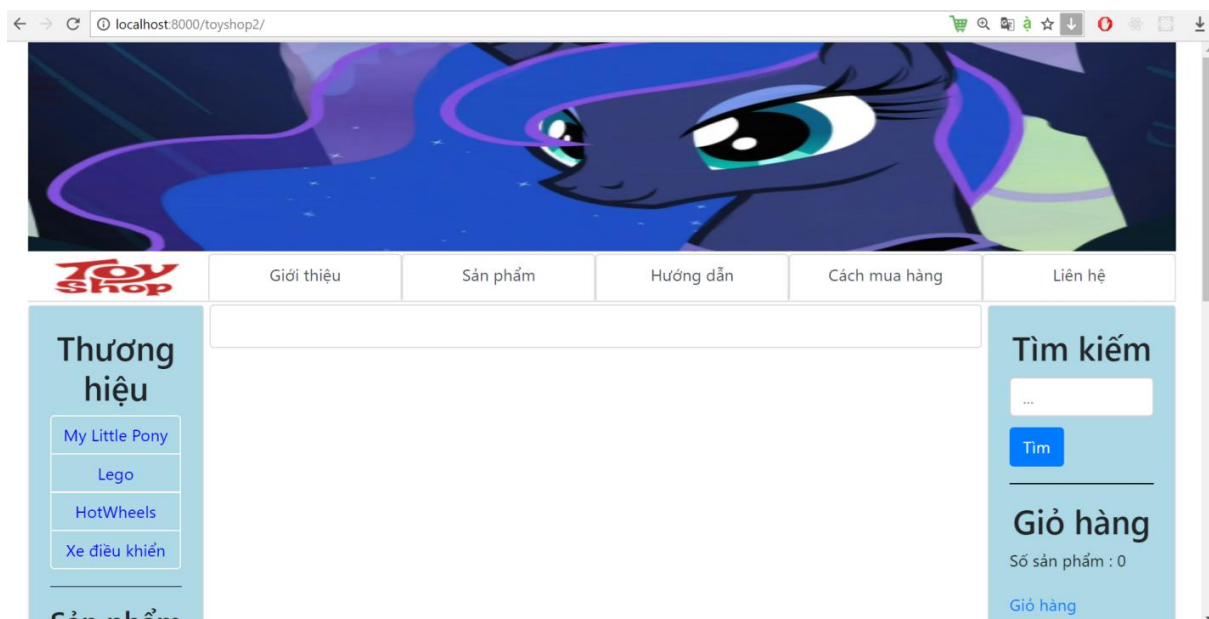
Để vào được trang quản trị cần có một tài khoản của admin, đã có 1 tài khoản mặc định với User: Admin, Pass: admin.

Tại trang Menu của quản trị, admin có thể thay đổi, chỉnh sửa trang web của mình, cũng như có thể thêm xóa sửa sản phẩm đang có. Tùy chỉnh banner hình ảnh slideshow. Tất cả các thao tác sau khi hoàn tất xong đều được thay đổi trong database.

3.2. Thiết kế giao diện

3.2.1. Giao diện phía người dùng

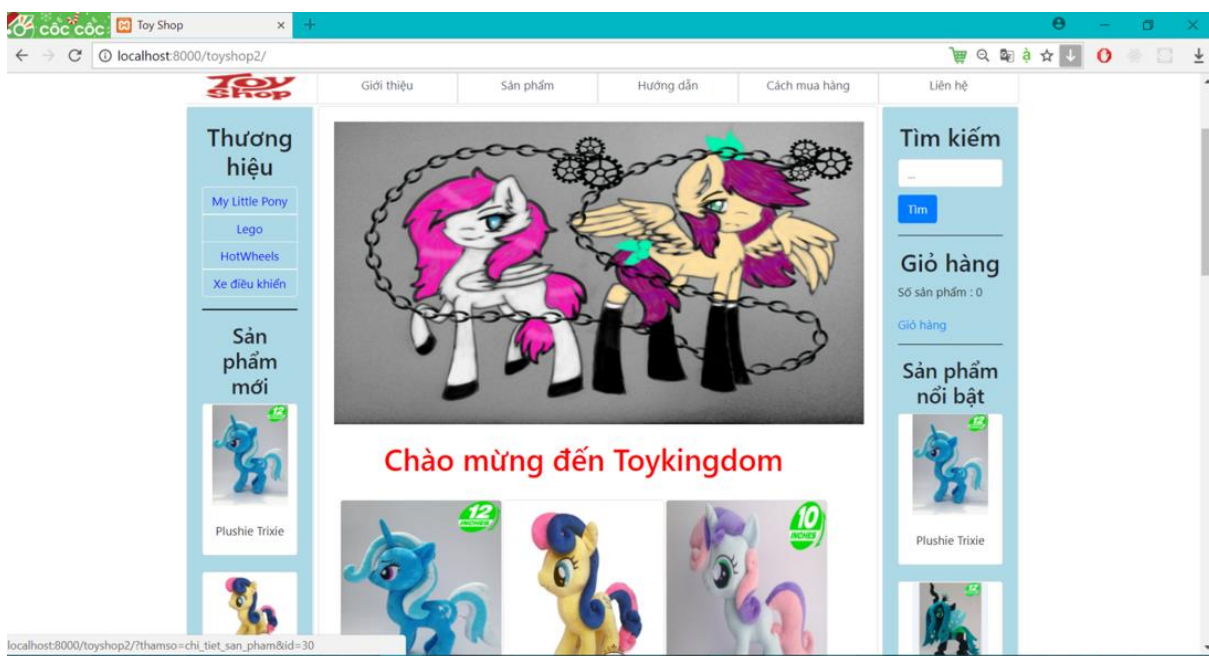
☐ Phần Main Layout



Phần Mainlayout là phần luôn hiển thị cho dù người dùng có nhấn vào bất kì menu nào. Ở đây phần Mainlayout bao gồm phần Banner, thanh Menu ngang và 2 cột dọc hai bên chứa các chức năng phục vụ cho trang web. Phần cuối cùng là footer của trang web.

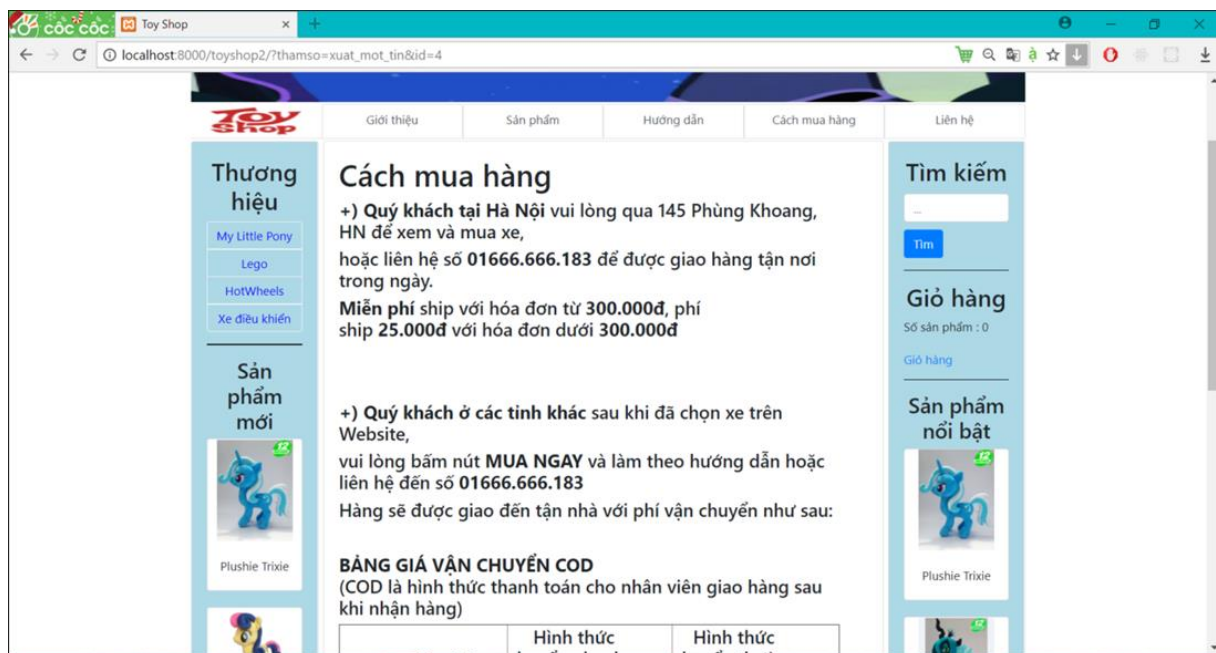
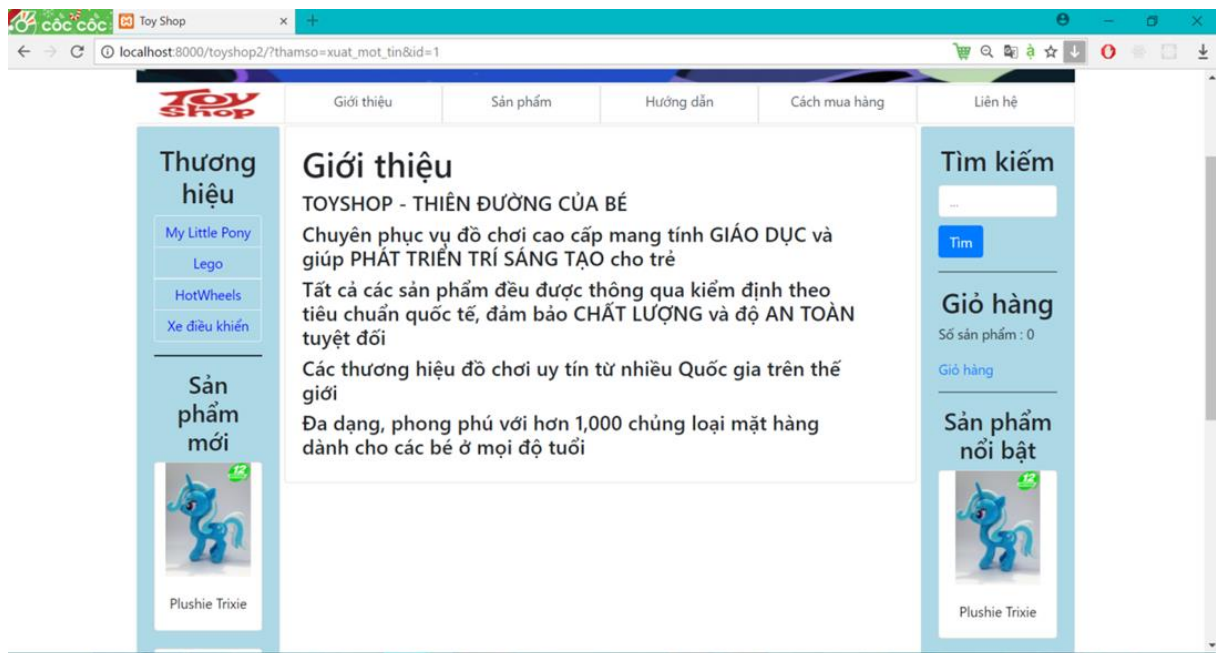
❑ Phần trang chủ

Phần trang chủ bao gồm phần Slide Show, lời chào mừng và một số sản phẩm nổi bật. Để truy cập vào trang chủ cần thông qua địa chỉ web <http://localhost:8000/toyshop2/>



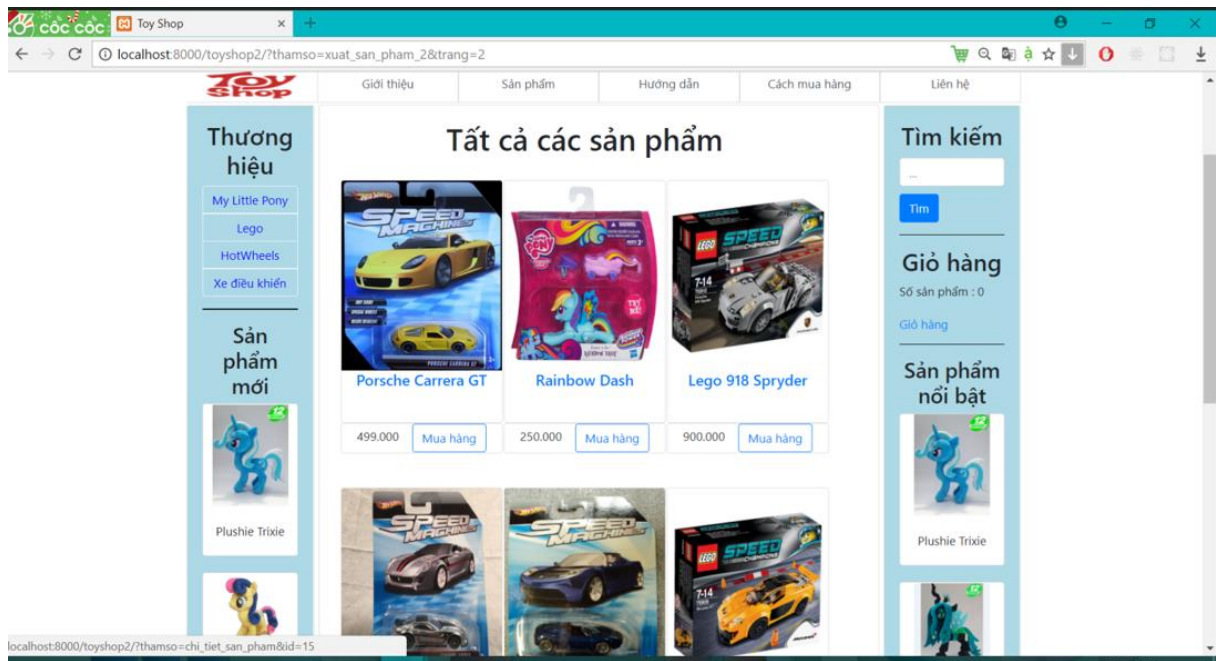
❑ Phần giới thiệu, hướng dẫn, cách mua hàng, thanh toán

Các phần này là xuất 1 bảng tin, 1 thông báo do phía admin ghi lại, nhằm cung cấp thông tin cũng như hướng dẫn khách hàng cách thức mua hàng, thanh toán,...



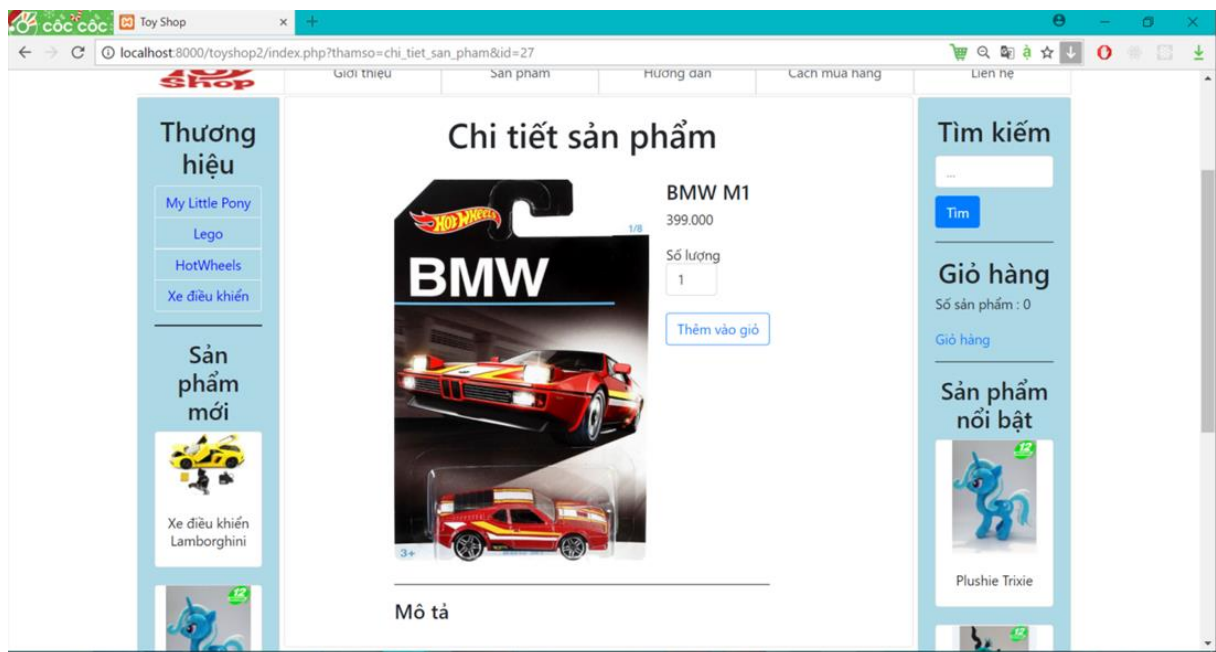
☐ Trang sản phẩm

Trang sản phẩm sẽ xuất toàn bộ sản phẩm trong kho, được phân thành nhiều trang, mỗi trang hiển thị 9 sản phẩm

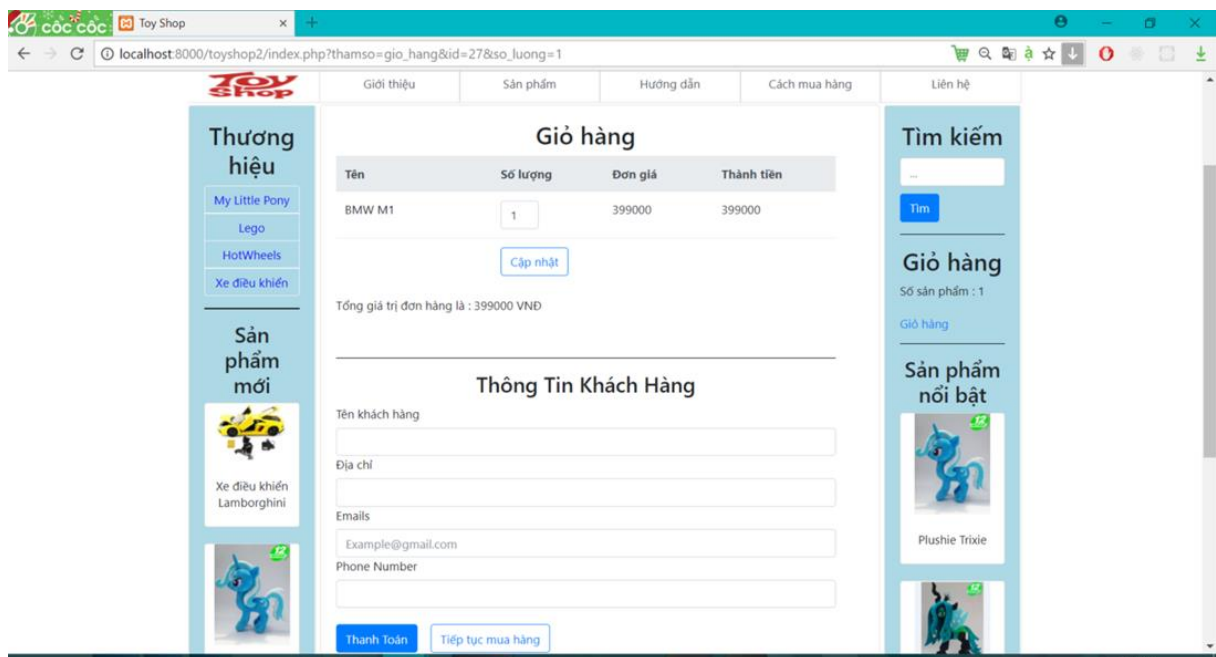


☐ Trang chi tiết sản phẩm

Trang chi tiết sản phẩm chỉ xuất hiện khi khách hàng nhấp chọn 1 sản phẩm. Ở đây sẽ đưa ra thông tin sản phẩm. Khách hàng cũng có thể chọn lựa số lượng sản phẩm và nhấn “Thêm vào giỏ” để mua hàng

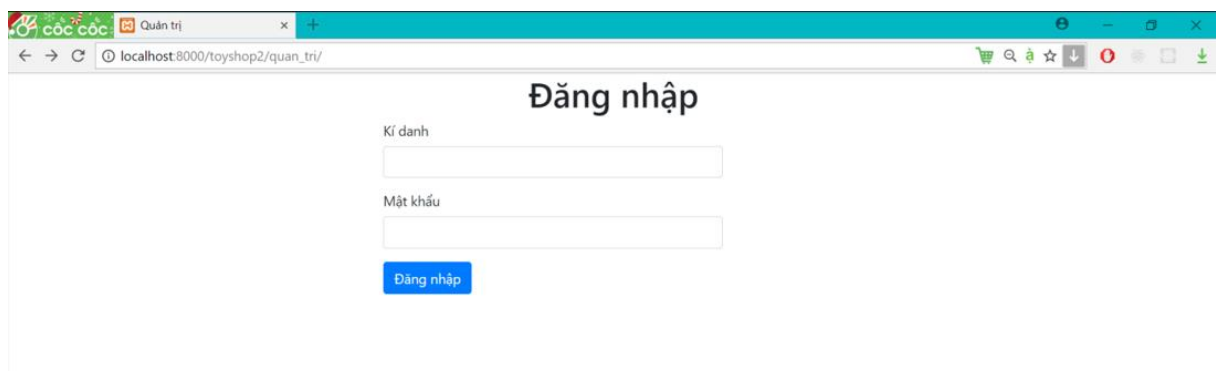


☐ Phần giỏ hàng

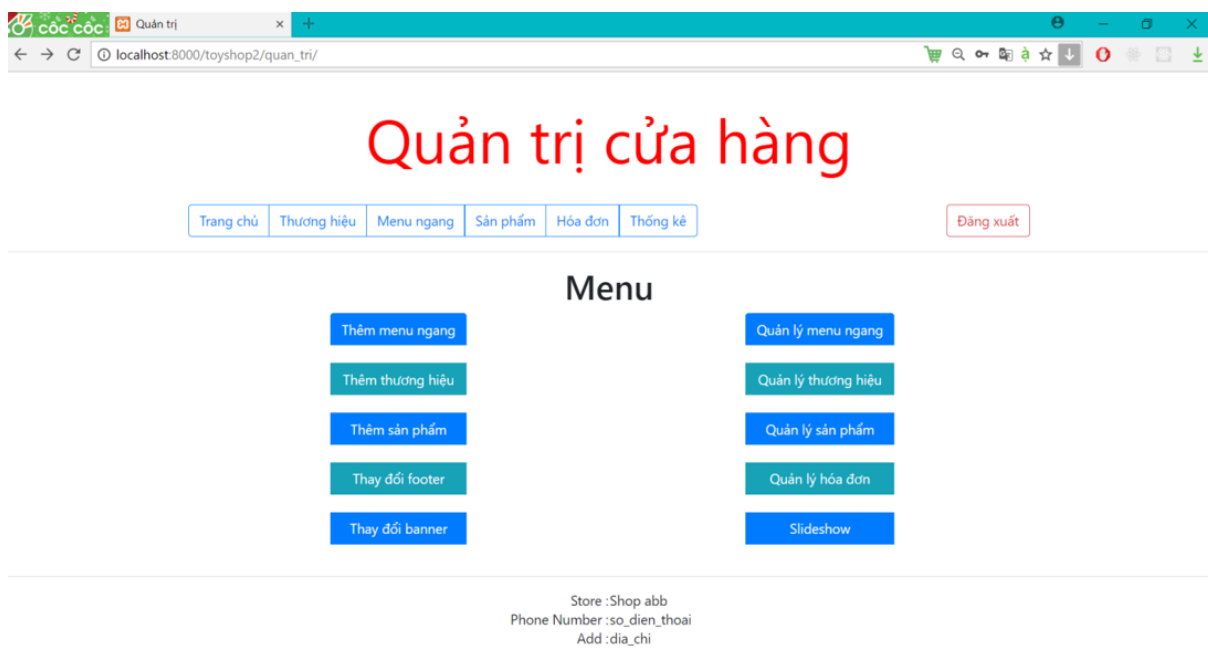


3.2.2 Giao diện phía admin

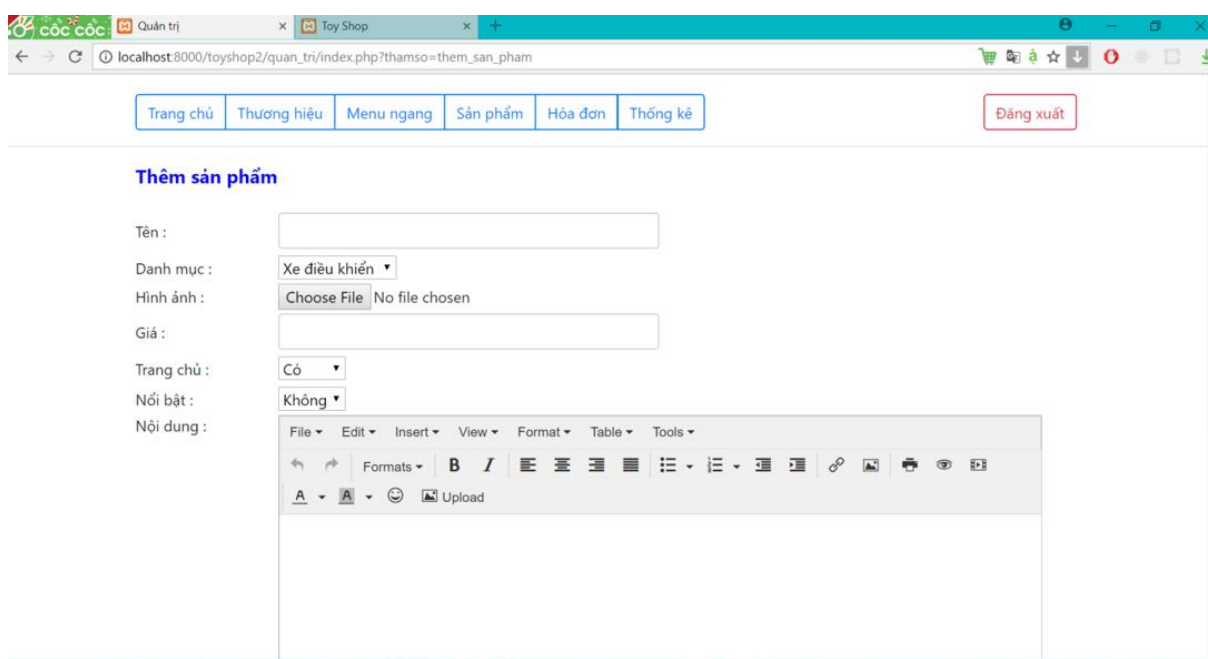
❑ Form đăng nhập

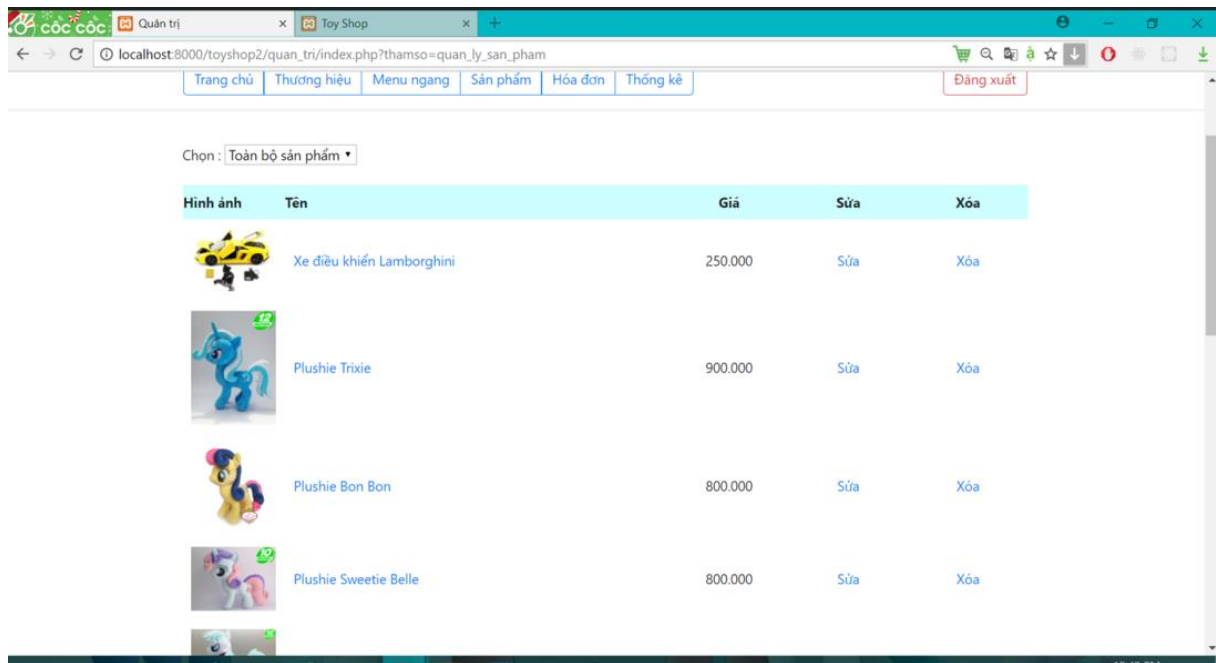


❑ Trang menu chức năng



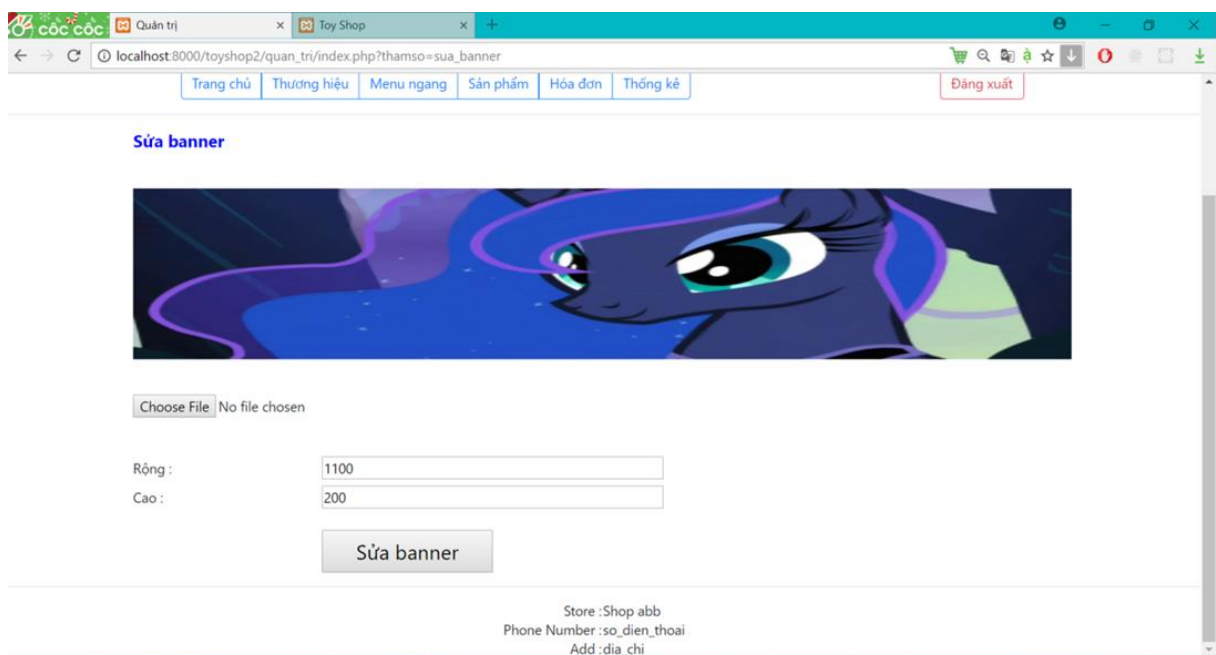
Ở trang này có đầy đủ các chức năng để admin tùy chỉnh website của mình
Bên cạnh đó admin có thể quản lý thêm sửa xóa các sản phẩm của mình

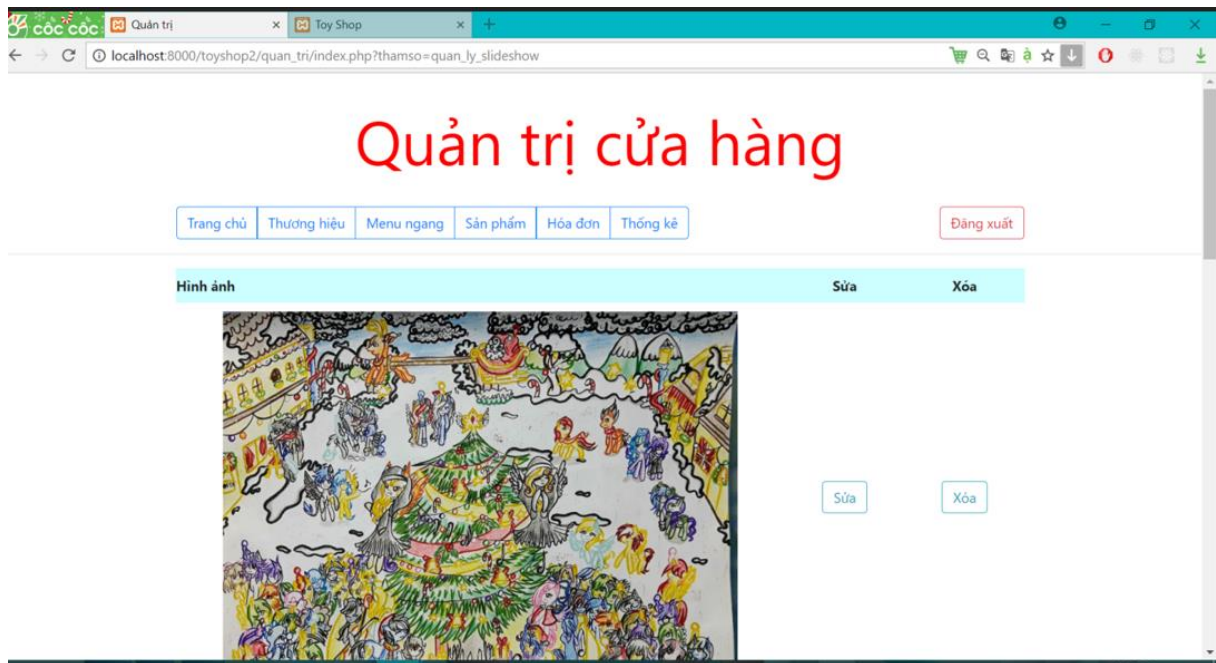




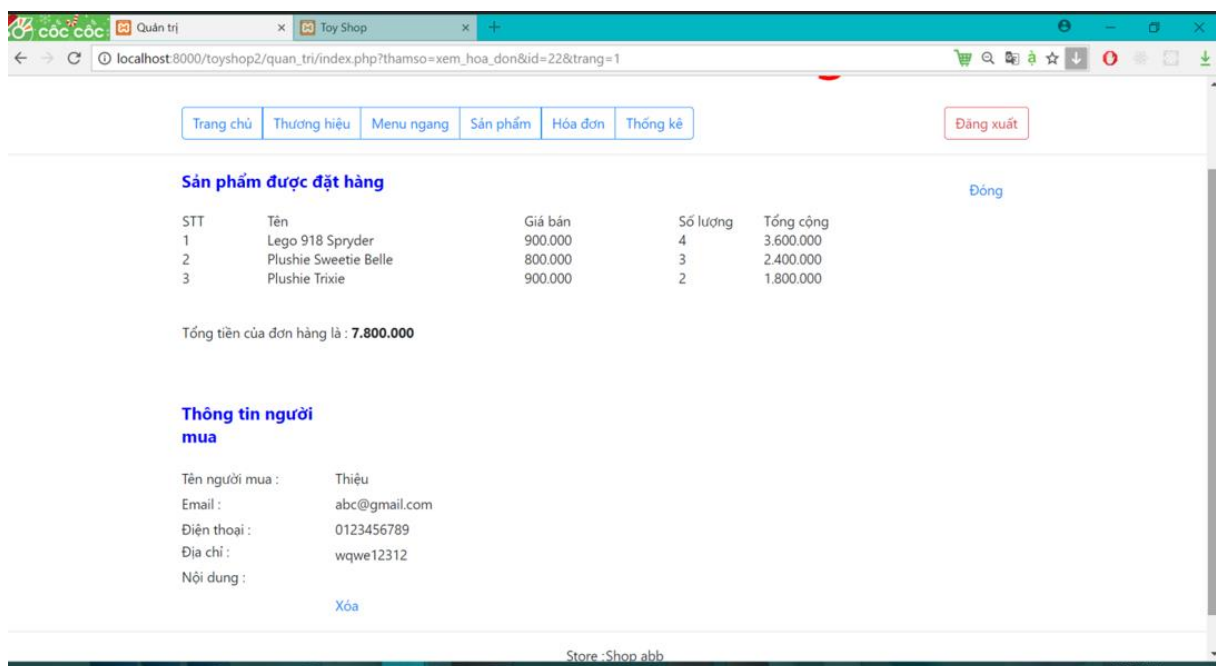
☐ Tùy chỉnh giao diện website

Quản lý Slideshow và Banner và Footer





❑ Xem chi tiết hóa đơn



Chương 5: TỔNG KẾT

5.1. Kết quả đạt được

Trong quá trình thực hiện tiểu luận, nhóm em đã có thêm cơ hội củng cố, tích lũy thêm kiến thức chuyên môn về lập trình web cũng như kinh nghiệm thiết kế website, lên kế hoạch cho đề tài, dự án, viết báo cáo.... Nắm rõ hơn về quy trình phát triển phần

mềm cũng như cấu trúc cú pháp của ngôn ngữ Javascript thông qua những tìm hiểu về React. Vận dụng những kiến thức đã học về mẫu thiết kế về hệ thống cũng như xây dựng cơ sở dữ liệu.

Sau khoảng thời gian tìm tòi, vận dụng những gì đã học cũng như với sự giúp đỡ học tập từ thầy, sản phẩm của nhóm em đã làm được những chức năng sau:

- Tạo ra giao diện thân thiện với người dùng
- Tạo được phần front-end lẫn back-end
- Quản lý được sản phẩm, chủng loại sản phẩm, hóa đơn
- Xử lý được yêu cầu đặt mua hàng từ người dùng
- Xử lý các form thông tin từ người dùng gửi lên hệ thống

Ngoài ra, chúng em còn củng cố thêm kiến thức về

- React JS, thư viện hỗ trợ tạo giao diện người dùng
- Meteor, framework tạo real-time web application

5.2. Hướng phát triển

Dù đã tạo được trang web tương đối đầy đủ các chức năng nhưng với thời gian có hạn, ứng dụng vẫn chưa đáp ứng được hết mọi yêu cầu người dùng

Trong tương lai em sẽ cố gắng khắc phục, chỉnh sửa một vài sai sót của trang web:

- Tạo Đăng Ký, Đăng Nhập cho người dùng
- Lọc sản phẩm một cách tốt hơn
- Áp dụng thêm các kỹ thuật về React JS để trang web được tối ưu hơn
- Thống kê sản phẩm đa dạng hơn, xuất hóa đơn

TÀI LIỆU THAM KHẢO

[1]. Thạc sĩ (DEA) Phạm Nguyễn Cương, Giáo trình Phân tích Thiết kế Hệ Thống Thông Tin, Nhà xuất bản Giáo Dục, 2014.

[2]. Ken Rogers, Meteor + React, Ft. Collins, Colorado, March 2016

[3] Tham khảo thông tin ở một số trang web:

+

<https://facebook.github.io/reactjs>

+ <https://viblo.asia>