

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO TIỂU LUẬN MÔN LẬP TRÌNH WEB

TÌM HIỂU VÀ XÂY DỰNG
TRANG WEB SỬ DỤNG VUEJS

Người hướng dẫn: **GV NGUYỄN THÁI DUY**

Người thực hiện: **NGUYỄN MINH HẢI – 51703074**

NGUYỄN NGỌC HOÀNG GIA – 51703070

ĐINH TRUNG HIẾU – 51703081

HỒNG CƠ NGHIỆP – 51703141

Lớp : 17050303

Khoá : 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO TIỂU LUẬN MÔN LẬP TRÌNH WEB

TÌM HIỂU VÀ XÂY DỰNG
TRANG WEB SỬ DỤNG VUEJS

Người hướng dẫn: **GV NGUYỄN THÁI DUY**

Người thực hiện: **NGUYỄN MINH HẢI – 51703074**

NGUYỄN NGỌC HOÀNG GIA – 51703070

ĐINH TRUNG HIẾU – 51703081

HỒNG CƠ NGHIỆP – 51703141

Lớp : 17050303

Khoá : 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

LỜI CẢM ƠN

Trong suốt thời gian làm bài luận, chúng em đã nhận được rất nhiều ý kiến để có thể hoàn thành tốt nhất. Nhờ vào các kiến thức, sự tận tâm giảng dạy chi tiết từng chương của thầy Nguyễn Thái Duy đã giúp nhóm em biết thêm nhiều kiến thức mới và áp dụng vào bài tiểu luận này.

Nhóm em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Thái Duy và bạn bè đã hỗ trợ, giúp đỡ nhóm em.

ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Chúng tôi xin cam đoan đây là sản phẩm báo cáo của riêng chúng tôi và được sự hướng dẫn của thầy Nguyễn Thái Duy. Các nội dung nghiên cứu, kết quả trong bài báo cáo này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài báo cáo còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào chúng tôi xin hoàn toàn chịu trách nhiệm về nội dung bài báo cáo của mình. Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do chúng tôi gây ra trong quá trình thực hiện (nếu có).

TP. Hồ Chí Minh, ngày 20 tháng 11 năm 2020

Tác giả

(ký tên và ghi rõ họ tên)

Nguyễn Minh Hải

Nguyễn Ngọc Hoàng Gia

Đinh Trung Hiếu

Hồng Cơ Nghiệp

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

Phần đánh giá của GV chấm bài

Tp. Hồ Chí Minh, ngày tháng năm
(kí và ghi họ tên)

TÓM TẮT

Bài báo cáo giới thiệu về Vuejs cơ bản: tổng quan về Vuejs, lợi ích, ... Áp dụng những kiến thức đã tìm hiểu trên mạng, nhóm em đã demo thiết kế website bán hàng điện tử sử dụng Vuejs.

MỤC LỤC

CHƯƠNG 1 – GIỚI THIỆU VỀ ĐỀ TÀI.....	4
1.1 Đặt vấn đề	4
1.2 Giới thiệu đề tài.....	4
1.3 Lí do chọn đề tài.....	4
CHƯƠNG 2 – TÌM HIỂU VỀ VUEJS	5
2.1 VueJS là gì ?	5
2.2 Những kiến thức cần biết về VueJS	5
2.2.1 Cách cài đặt và bắt đầu sử dụng <i>VueJS</i>	5
2.2.2 Khai báo thành phần <i>VueJS</i>	6
2.2.3 <i>Vue Component</i>	8
2.2.3.1 Component là gì?	8
2.2.3.2 Khai báo component	8
2.2.3.3 Data Component	9
2.2.3.4 Quan hệ Component cha-con.....	10
2.2.4 Các khái niệm liên quan	11
2.2.4.1 Computed & Methods.....	11
2.2.4.2 Directives	11
2.3 Thư viện Vuex.....	16
2.3.1 <i>Vuex</i> là gì?	16
2.3.2 Cấu trúc của <i>Vuex</i>	16
2.3.3 Ứng dụng của <i>Vuex</i>	17
2.3.4 Cài đặt <i>Vuex</i>	18
2.4 Lí do nên chọn <i>Vuejs</i>	18
CHƯƠNG 3 – XÂY DỰNG ỨNG DỤNG	19
3.1 Chức năng của website:	19

3.2 Thiết kế database.....	19
3.3 Kết quả demo	20
3.4 Những điểm hạn chế trong project.....	23
3.5 Hướng phát triển trong tương lai	23
TÀI LIỆU THAM KHẢO.....	24

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

DANH MỤC HÌNH

Hình 1 : Logo Vuejs.....	5
Hình 2 : Hình ảnh tượng trưng quan hệ component cha-con.....	10
Hình 3: Sơ đồ database.	19
Hình 4: Giao diện trang chủ.....	20
Hình 5: Giao diện hiển thị sản phẩm.....	20
Hình 6: Giao diện phân trang sản phẩm.....	21
Hình 7: Giao diện giỏ hàng	21
Hình 8: Giao diện đăng nhập	22
Hình 9: Giao diện quản lý sản phẩm trang admin.....	22

CHƯƠNG 1 – GIỚI THIỆU VỀ ĐỀ TÀI

1.1 Đặt vấn đề

Với công nghệ ngày càng hiện đại hơn, nhu cầu của khách hàng cũng ngày một tăng. Việc thiết kế ra một website phải đảm bảo đầy đủ các yếu tố : giao diện, chức năng, có thể thay đổi phát triển trong tương lai, và yêu cầu quan trọng nhất là phải thiết kế ra một website nhanh và hoàn thành đúng hạn cho khách hàng. Vì những yêu cầu này nên những framework hỗ trợ cho việc lập trình web đã ra đời. Ở riêng phần Frontend ta có Reactjs, AngularJs, Vuejs,... Ở phần Backend thì ta có Laravel, Django, Ruby on Rails,...

Ở trong bài tiểu luận này, nhóm em sẽ tập trung vào Vuejs để tìm hiểu kiến thức, chức năng, công dụng và cách sử dụng cơ bản của loại framework này.

1.2 Giới thiệu đề tài

Thông qua framework VueJS, tổ chức xây dựng trang web theo cấu trúc chia thành các component rẽ nhánh, tạo ra trang web bán hàng (gear máy tính) linh động, thông minh, nhẹ.

1.3 Lí do chọn đề tài

Dự kiến trong tương lai có thể nhóm sẽ đi theo hướng fullstack, về phần backend thì sẽ chọn framework Laravel. Hiện tại thì có rất nhiều framework có thể hỗ trợ tốt cho phần Front-end như ReacJs, AngularJs,... Theo như thông tin tìm hiểu thì Vuejs đã được Taylor Well (người sáng lập ra Larvavel) tích hợp sẵn vào Laravel. Đường như Laravel và Vuejs là một sự kết hoàn hảo nên nhóm đã quyết định chọn đề tài tìm hiểu về Vuejs và xây dựng trang web bán hàng sử dụng Vuejs để hiểu rõ hơn về công dụng của framework này.

CHƯƠNG 2 – TÌM HIỂU VỀ VUEJS

2.1 VueJS là gì ?

Để phát triển ứng dụng, trang web, các developer đã sử dụng rất nhiều các công nghệ phần mềm. Hiện nay, Vuejs là một trong những công nghệ phát triển web đang được sử dụng rộng rãi trên toàn thế giới. Vuejs là javascript framework dùng để xây dựng giao diện người dùng. Đây là một framework linh động (progressive framework) vì nó được chia thành các phần nhỏ khác nhau, người dùng có thể lựa chọn các thành phần phù hợp với project của mình. Phần core của Vuejs chỉ tập trung vào lớp view, kiến thức cũng không quá khó để người dùng có thể tiếp cận nhanh chóng.



Hình 1 : Logo Vuejs.

2.2 Những kiến thức cần biết về VueJS

2.2.1 Cách cài đặt và bắt đầu sử dụng VueJS

Để có thể học và sử dụng Vuejs thì người dùng cần có kiến thức cơ bản về Javascript vì Vuejs được viết ở Javascript và vì Vuejs làm việc ở frontend nên người dùng cần biết về HTML, CSS.

Để bắt đầu với Vuejs, ta có thể tải về máy để chạy offline hoặc nhúng đường link CDN theo 2 cách sau:

Cách 1: cài đặt trực tiếp file về máy với đường dẫn phía dưới

Installation

Cách 2: Nhúng đường dẫn CDN để chạy trực tiếp. Trang chủ VueJS cung cấp 2 link CDN cho người dùng:

```
<!-- development version, includes helpful console warnings -->
<script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
```

Link đầu tiên dành cho developer trong trường hợp đang phát triển web với các thông báo lỗi ở phần console hỗ trợ cho việc debug , Hoặc:

```
<!-- production version, optimized for size and speed -->
<script src="https://cdn.jsdelivr.net/npm/vue@2"></script>
```

Link thứ 2 dành cho sản phẩm chính thức, với kích thước file được thu gọn và file cũng được tối ưu hóa hơn.

Vì là mã nguồn mở cho nên có thể truy cập file để xem code của VueJS để hiểu rõ hơn về framework này.

2.2.2 Khai báo thành phần VueJS

Đầu tiên là cách khai báo một đối tượng Vue (tại JS):

```
var ex = new Vue({ })
```

Framework VueJS liên kết giữa hệ thống và cấu trúc DOM để có thể render dữ liệu một cách ĐƠN GIẢN, NHANH CHÓNG, LINH ĐỘNG. Sau khi liên kết có thể dễ dàng quản lý, thành đổi hay làm việc bất kỳ trên vùng đã khai báo trước đó, ví dụ:

Với một div có id = “example”, dùng option *el* bên trong đối tượng Vue đã khai báo trước đó để tạo liên kết giữa đối tượng Vue và thành phần trên DOM.

```
var ex = new Vue({
  el: '#example'
})
```

Cùng với option el, thì data cũng là 1 option k thể thiếu khi tạo một đối tượng Vue, với mục đích chứa các biến dùng trong việc render dữ liệu linh động ra bên ngoài trang web:

```
var ex = new Vue({
  el: '#example',
  data :{
    *tên biến* : *giá trị của biến*
  }
})
```

Một ví dụ đơn giản về cách thức vận hành của VueJS khi render dữ liệu ra website, với cú pháp (ở HTML)

```
<div id="example">
  {{ *tên biến* }}
</div>
```

Và khi kiểm tra phần tử trực tiếp trên website sẽ là:

```
<div id="example">
  *giá trị của biến*
</div>
```

Không kèm theo bất kỳ thành phần nào của VueJS.

2.2.3 Vue Component

2.2.3.1 Component là gì?

Component được xem như là một trong các tính năng mạnh mẽ nhất của Vuejs. Chức năng của component là nhóm các code như HTML, CSS, Javascript lại với nhau như một khối để tái sử dụng cho các module tương tự. Khi muốn thay đổi gì thì chỉ cần thay đổi trong component, tất cả những chỗ ta gọi ra cũng sẽ thay đổi theo, tránh được tình trạng thiếu sót khi sửa code, việc này giúp giảm được đáng kể thời gian code và chỉnh sửa cho người dùng. *Sau đó chỉ việc gọi component tại trang index (xem component như tag HTML đã khai báo trước đó) để tạo thành khối web hoàn chỉnh*

2.2.3.2 Khai báo component

Khai báo component có 2 kiểu là **global** và **local**.

Global: khai báo component toàn cục thì component này sẽ được truy xuất ở bất cứ đâu. Cú pháp:

```
var app = new Vue({
  el: '#example',
});

Vue.component('component-name',{
  template: *Cấu trúc HTML, bao gồm các tag*
});
```

component-name là tên muốn gán cho component. Sau khi khai báo xong component thì tiến hành gọi như một thẻ html:

```
<div id="example">
  <component-name> </component-name>
</div>
```

Local: khai báo component cục bộ sẽ chỉ sử dụng được ở trong phạm vi được khai báo. Cú pháp để khai báo kiểu local sẽ giống như khi định nghĩa một object:

```
var ex = new Vue({
  el: '#example',
  components: {
    'component-name': {
      template : '<h1>Hello World!</h1>'
    }
  }
});
```

Kết hợp với ví dụ component global, khi khai báo kiểu local thì component-name sẽ chỉ được sử dụng trong #example, còn lại sẽ không hiển thị ra kết quả.

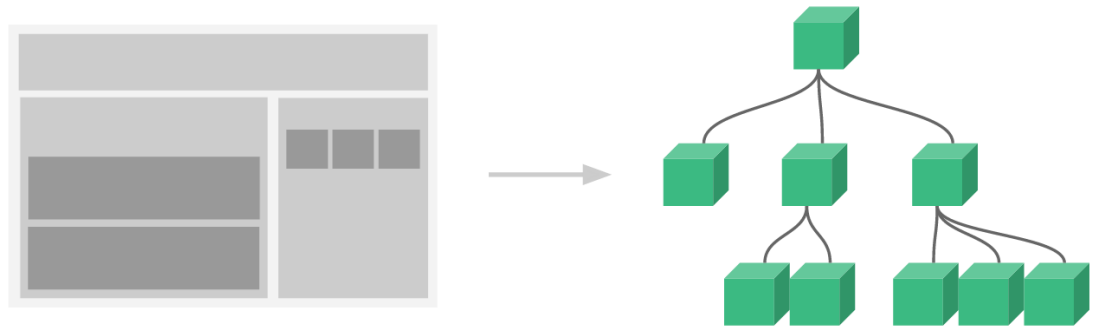
2.2.3.3 Data Component

Khác với Vue instance khai báo data ở dạng object thì **component** phải khai báo **data** ở dạng **function**. Vì trường hợp gọi component nhiều lần, nếu không khai báo ở dạng object thì những instance sẽ nhận chung 1 giá trị data, khi thay đổi data ở một instance nào đó thì tất cả các instance khác cũng sẽ thay đổi theo.

Đây cũng là đặc điểm tạo nên thế mạnh của Vue component ở tính tái sử dụng mà không bị trùng lặp dữ liệu, dễ thay đổi.

2.2.3.4 Quan hệ Component cha-con

Vì một trang web được cấu thành từ nhiều component khác nhau nên trang web sẽ có cấu trúc dạng rẽ nhánh (hay còn gọi là quan hệ cha-con) :



Hình 2 : Hình ảnh tượng trưng quan hệ component cha-con.

Trong Vuejs, thì các component được tạo ra sẽ có sự tương tác qua lại với nhau, sử dụng component này bên trong component khác, tạm gọi là parent-child components. Quan hệ này được gọi là **props down, events up**. Trong VueJS nó có thể được tóm gọn lại như props down hay events up - parent truyền props cho child và child phản hồi lại cho parent bằng event.

Props down: truyền dữ liệu trực tiếp từ component cha vào con, component con chỉ được xài thôi, không được thay đổi trực tiếp và cũng không truyền dữ liệu từ con sang cha. Data có thể gửi xuống từ component cha thông qua một custom attribute là props.

Event up: truyền sự kiện thông báo cho component cha biết là component con muốn thay đổi dữ liệu. Nhiệm vụ của component cha là nhận thông báo và tiến hành thay đổi data.

2.2.4 Các khái niệm liên quan

2.2.4.1 Computed & Methods

Nói dễ hiểu thì cả 2 thuộc tính trên đều dùng để tính toán các dữ liệu và được gọi ra biểu diễn. Điểm khác nhau giữa 2 thuộc tính trên là:

- Computed thì được xem như một hàm không có tham số, còn Methods thì có.
- Cách gọi Computed: `<p> {{ Function }} </p>`.

Cách gọi Methods: `<p>{{ Function() }} </p>`.

Thậm chí với những phép tính đơn giản Computed có thể tính toán ngay tại vị trí cần sử dụng: `<p> {{ *tên biến*.hàm javascript* }} </p>`

Ngoài ra, sự khác biệt lớn nhất giữa 2 thuộc tính trên là việc giá trị của Computed được lưu vào bộ nhớ đệm để tránh việc tính toán lại nếu dữ liệu không có gì thay đổi, vì thế Computed thường được sử dụng như một công cụ để tính toán trực tiếp các dữ liệu có sẵn chứa trong data để phục vụ những mục đích cơ bản. Methods sẽ trở nên như một hàm của javascript, nhận các tham số đầu vào và tính toán khi được gọi.

2.2.4.2 Directives

Ngoài ra, Vuejs cũng có *các thuộc tính* được gọi là *directive*, tương tự như attribute của một thẻ HTML. Một *directive* sẽ được sử dụng bắt đầu bằng **v-** để xác định đây là thuộc tính do Vue cung cấp.

v-if, v-else, v-else-if: để xét điều kiện. Ví dụ:

```
<div id="example2">
  <p v-if="seen">Thanks you</p>
  <p v-else>I'm sorry.</p>
```

```
</div>
```

```
var ex2= new Vue({
  el: '#example2',
  data: {
    seen: true
  }
});
```

Ở ví dụ if else này thì data lưu dữ liệu seen với giá trị là true, nên giao diện sẽ in ra dòng Thanks you, ngược lại nếu giá trị là false thì sẽ in dòng I'm sorry.

Để hiển thị dữ liệu lên giao diện trang web ta sẽ viết theo dạng khai báo với cú pháp đơn giản `{{ x }}` với x được xem như là một biến được khai báo ở javascript. Ví dụ đơn giản:

```
<div id="project"> {{ message }} </div>
var myProj = new Vue({
  el: '#project',
  data: {
    message: 'Hello world!'
  }
});
```

Ngoài ra trong `{{ }}` ta còn có thể sử dụng các hàm của javascript. Ví dụ để chuyển đổi tất cả kí tự trong đoạn chuỗi trên thành in hoa:

```
<div id="project"> {{ message.toUpperCase() }} </div>
```

v-html: để hiển thị dữ liệu dưới dạng code HTML thì ta khai báo theo cú pháp:

`<tag v-html="data">` với **tag** là các thẻ tag trong HTML và data là dữ liệu đã được khai báo ở data trong vuejs. Ví dụ in 1 đoạn chuỗi Hello World! màu đỏ ra html thông qua vuejs:

```
<div id=" project">
  <h1 v-html="message"></h1>
</div>
<script type="text/javascript">
  var proj = new Vue({
    el: '# project',
    data: {
      message: '<font color="red"> Hello World! </font>'
    },
  });
</script>
```

v-bind: là cú pháp để thêm thuộc tính vào tag HTML bằng dữ liệu được tạo ra ở vuejs với cú pháp:

```
<tag v-bind:attributeName= "data" ></tag>
```

hoặc viết tắt:

```
<tag :attributeName="data" ></tag>.
```

attributeName là tên loại thuộc tính mà ta muốn sử dụng, **data** là dữ liệu được khai báo ở data vuejs. Ví dụ ta muốn thêm 1 class và 1 style vào thẻ p.

```
<style type="text/css">
  . colorRed {
    color: red;
  }
</style>
```

```

<div id=" project">
  <p v-bind:class=" className" v-bind:style="styleData" >{{ message }} </p>
</div>
<script type="text/javascript">
  var proj = new Vue({
    el: '# project',
    data: {
      message: 'Hello World!',
      className: 'colorRed',
      styleData: 'text-transform: uppercase',
    },
  });
</script>

```

v-if: sử dụng để xét điều kiện true, false tương tự như các ngôn ngữ lập trình khác. Một ví dụ đơn giản để hiện thị 1 đoạn chuỗi với giá trị **true**.

```

<div id="example">
  <span v-if=" appearance ">Xuất hiện! </span>
</div>
var ex = new Vue({
  el: '#example',
  data: { appearance: true
  }
})

```

Ngoài ra ta còn có thể sử dụng trực tiếp câu điều kiện bên trong `{{}}` bằng cú pháp sau:

```

<div id="example">
  <p>{{ num }} là số : {{ num % 2 == 0 ? 'Chẵn' : 'Lẻ' }}</p>

```

```

</div>
<script type="text/javascript">
    var ex = new Vue({
        el: '# example',
        data: { num : 15
        }, });
</script>

```

v-for: tạo ra một vòng lặp, ví dụ để hiển thị ra một danh sách các loại điện thoại với dữ liệu được lưu trong 1 mảng:

```

<div id="exampleFor">
    <ol> <li v-for="item in items"> {{ item.product }} </li> </ol>
</div>

```

Ta sẽ tạo ra một biến **item** và chạy trong mảng **items** để lấy ra từng dữ liệu và hiển thị qua cú pháp **item.product**.

```

var exFor = new Vue({
    el: '#exampleFor',
    data: {
        items: [ { product: 'Iphone' },
                  { product: 'SamSung' },
                  { product: 'Nokia' } ]
    }
})

```

v-on:click: dùng để thêm sự kiện click vào button. Cú pháp:

<button v-on:click="myFunction">Click me!</button> với myFunction là hàm xử lý được viết trong methods của vuejs. Vuejs còn cung cấp 1 ký hiệu viết tắt cho **v-on:** là

@ để giúp người dùng viết code nhanh hơn(**v-on:click** - **@click**). Ví dụ ta muốn hiển thị thông báo dạng alert khi click vào button thì:

```
<script type="text/javascript">
    var app = new Vue({
        el: '#demo',
        methods: {
            myFunction: function () {
                alert('Bạn vừa click vào button');
            }
        }
    });
</script>
```

2.3 Thư viện Vuex

2.3.1 Vuex là gì?

Vuex là thư viện giúp quản lý trạng thái các component trong VueJS, nó là nơi lưu trữ tập trung cho tất cả các component trong một ứng dụng, với nguyên tắc trạng thái chỉ có thể được thay đổi theo kiểu có thể dự đoán được.

2.3.2 Cấu trúc của Vuex

Store trong vuex chính là nơi quản lý tập trung state của các ứng dụng bao gồm 5 phần chính:

State (trạng thái): Vuex sử dụng một cây trạng thái duy nhất, đối tượng này sẽ chứa tất cả các trạng thái của ứng dụng, như vậy chỉ có duy nhất một kho lưu trữ cho mỗi ứng dụng, điều này làm cho việc xác định các trạng thái là dễ dàng.

Mutation (thay đổi trạng thái): Điểm mạnh của store chính là component chỉ có thể đọc dữ liệu mà không được phép thay đổi trạng thái một cách trực tiếp. Nếu

muốn thay đổi trạng thái thì Mutations là nơi duy nhất đảm nhiệm chức năng này bằng cách thay đổi thông qua commit và mỗi commit này là một hành động được thực hiện tại action. Vuex mutation tương tự như sự kiện, mỗi mutation có một kiểu string và một handel.

Action (hành động): Action cũng tương tự như mutation, tuy nhiên có một vài điểm khác biệt, là nơi chứa các hành động như lấy dữ liệu api từ server hay một hành động làm thay đổi dữ liệu trong CSDL(thêm, sửa, xóa) và hơn thế nữa action commit các thay đổi state ở mutation. Action có thể chứa các hoạt động không đồng bộ.

Getters (lọc trạng thái): lấy các trạng thái dựa vào việc tính toán, lọc bỏ các trạng thái được cung cấp bởi kho lưu trữ.

Module: Vuex sử dụng cây trạng thái duy nhất, tất cả các trạng thái của ứng dụng được đưa vào một đối tượng, điều này cũng có nghĩa là khi ứng dụng của bạn ngày càng phát triển thì store sẽ to lên rất nhiều. Vuex cho phép chia nhỏ store thành các module nhỏ hơn, mỗi module cũng có state, mutation, action, getter và thậm chí còn cho phép các module lồng nhau.

2.3.3 Ứng dụng của Vuex

Điều quan trọng nhất khi sử dụng Vuex đó chính là xác định cái nào là state sẽ lưu trữ trong store, cái nào chỉ là local state nằm trong component. Hoàn toàn có thể đưa tất cả vào trong store, tuy nhiên đến khi quy mô của ứng dụng càng lớn kéo theo store cũng sẽ to ra dẫn đến việc khó khăn trong việc quản lý state. Vuex chính là một giải pháp tối ưu để giải quyết những vấn đề trong việc đồng bộ, hiển thị và update dữ liệu, data được sử dụng bởi nhiều component, khi sử dụng vuejs. Đa số các sản phẩm vuejs lớn đều cần sử dụng vuex cho việc quản lý và xử lý thay đổi trạng thái hiệu quả nhất.

2.3.4 Cài đặt Vuex

Cài đặt Vuex khá đơn giản, không quá phức tạp. Ta cần dùng npm để cài đặt Vuex:

```
npm install vuex --save
```

Import thư viện Vuex vào dự án và thông báo việc sử dụng thư viện cho VueJS bằng hàm `Vue.use()`.

```
import Vue from 'vue'
```

```
import Vuex from 'vuex'
```

```
Vue.use(Vuex)
```

2.4 Lí do nên chọn Vuejs

Sở dĩ Vuejs đang rất nhiều người sử dụng vì nó có **kích thước nhỏ gọn**, kích thước của Vuejs framework chỉ tầm 18-21KB, người dùng có thể tải xuống và sử dụng ngay, hoặc chúng ta có thể thêm thư viện trực tiếp vào đồ án qua thẻ script. Lợi ích kế tiếp là nó rất **dễ dàng tìm hiểu và sử dụng**, qua ví dụ cơ bản ở mục 2.2.1, ta có thể thấy được cấu trúc không quá phức tạp của Vuejs, người dùng có thể dễ dàng thêm phần mềm vào dự án website của mình. Nhờ vào cấu trúc đơn giản, người dùng cũng sẽ dễ dàng tìm ra lỗi và sửa chữa nhanh chóng. Vuejs có **tài liệu chi tiết**, đây là một điểm cộng rất lớn vì các developer thường muốn sử dụng các framework hướng dẫn chi tiết để có thể dễ dàng viết các ứng dụng đầu tiên. Hiện tại thì vuejs có trang web tài liệu tiếng Anh <https://vuejs.org/> và cả tiếng Việt <https://vi.vuejs.org/> để người Việt tiếp cận dễ dàng hơn.

CHƯƠNG 3 – XÂY DỰNG ỨNG DỤNG

3.1 Chức năng của website:

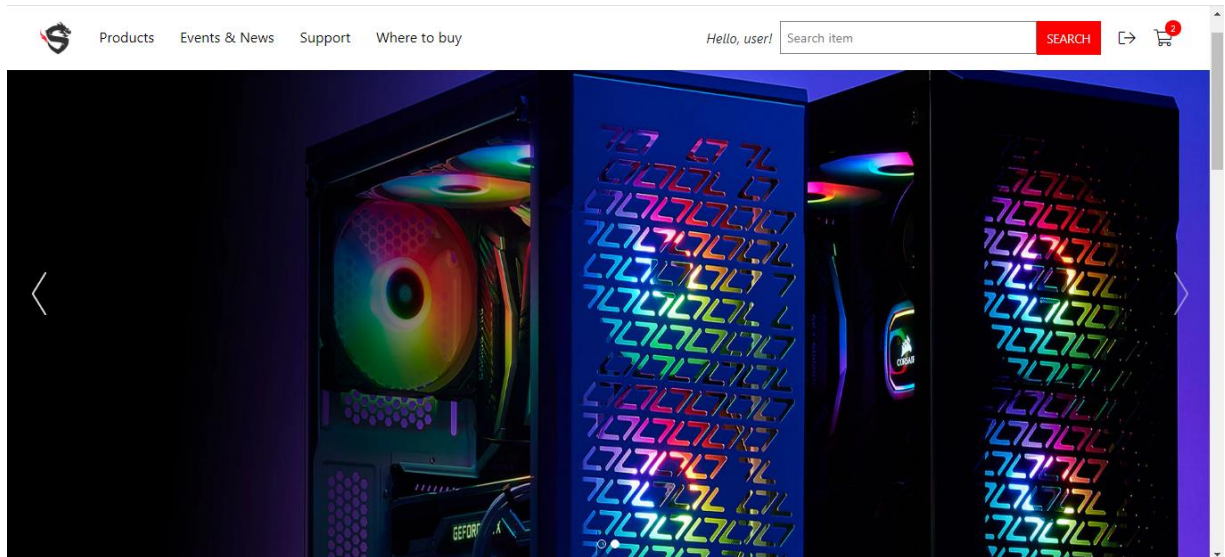
- Phân quyền admin, khách hàng.
- Chức năng admin: thêm, sửa, xoá sản phẩm.
- Thêm sản phẩm vào giỏ hàng.
- Chức năng giỏ hàng: tự động cập nhật giá khi tăng số lượng.
- Hiển thị sản phẩm trong database.
- Phân trang

3.2 Thiết kế database

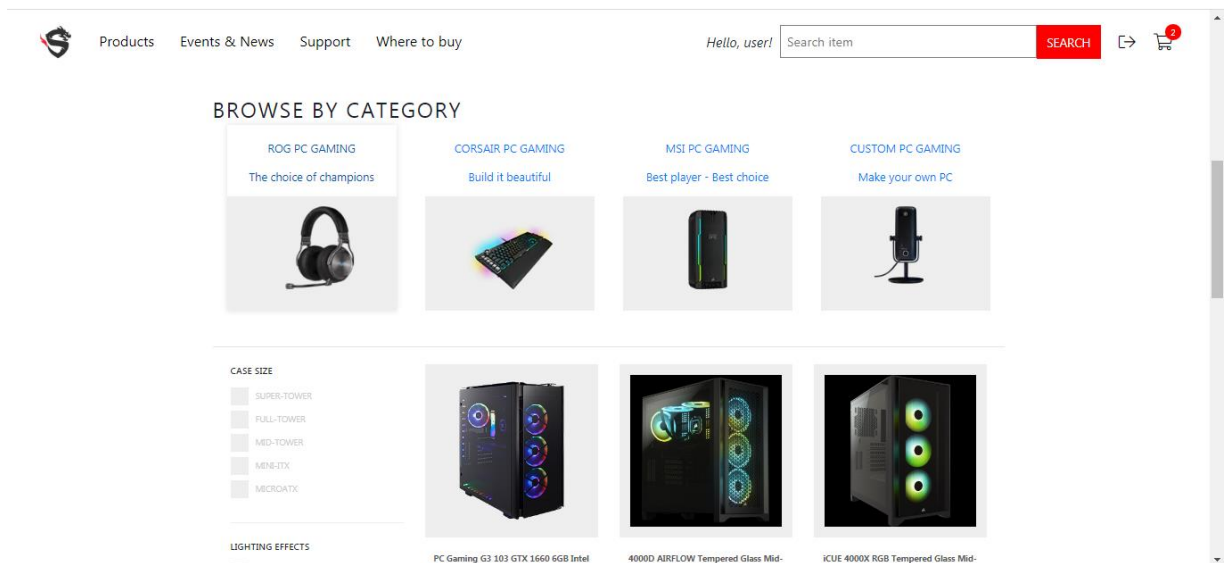
<i>Products</i>
Id
Userid
imgSrc
ProductName
Quantity
Price
discount
description
number
Show_products

Hình 3: Sơ đồ database.

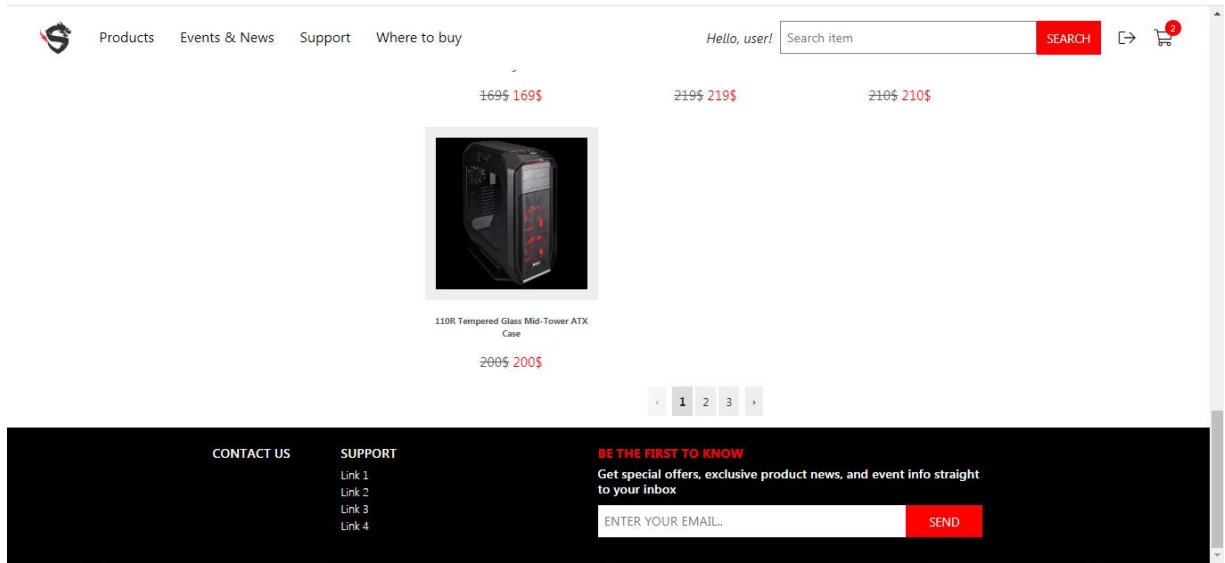
3.3 Kết quả demo



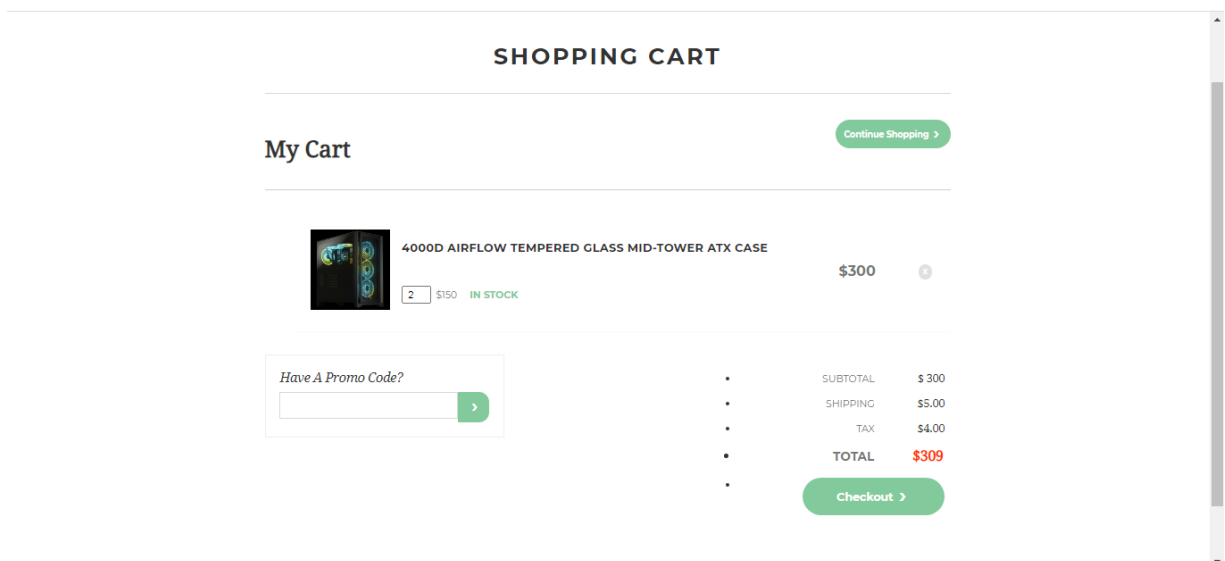
Hình 4: Giao diện trang chủ



Hình 5: Giao diện hiển thị sản phẩm



Hình 6: Giao diện phân trang sản phẩm.



Hình 7: Giao diện giỏ hàng

User Login

Username

Password

☐ Remember login







Login

Don't have an account yet? [Register now.](#)

Forgot your password? [Reset your password.](#)

Đăng nhập bằng tài khoản: **admin - 123456**

Hình 8: Giao diện đăng nhập

Thêm sản phẩm				Đăng xuất
Image	Name	Price	Description	Action
	PC Gaming G3 103 GTX 1660 6GB Intel Core i3-9100F	200 \$	quia et suscipit suscipit recusandae consequuntur expedita et cum reprehenderit molestiae ut ut quas	Edit Delete
	4000D AIRFLOW Tempered Glass Mid-Tower ATX Case	150 \$	est rerum tempore vitae sequi sint nihil reprehenderit dolor beatae ea dolores neque fugiat blanditi	Edit Delete
	iCUE 4000X RGB Tempered Glass Mid-Tower ATX Case	150 \$	et iusto sed quo iure voluptatem occaecati omnis eligendi aut ad voluptatem doloribus vel accusantium	Edit Delete
	4000D Tempered Glass Mid-Tower ATX Case Black	179 \$	ullam et saepe reiciendis voluptatem adipisci sit amet autem assumenda provident rerum culpa quis hi	Edit Delete
	Carbide Series SPEC-01 Red LED Mid-Tower Gaming C	200 \$	repudiandae veniam quaerat sunt sed alias aut fugiat sit autem sed est voluptatem omnis possimus ess	Edit Delete
				

Hình 9: Giao diện quản lý sản phẩm trang admin

3.4 Những điểm hạn chế trong project

- Chưa có chức năng thanh toán.
- Xoá sản phẩm trong giỏ hàng.

3.5 Hướng phát triển trong tương lai

- Hoàn thiện chức năng thanh toán.
- Hiện thị sản phẩm theo danh mục.
- Xem chi tiết từng sản phẩm.

TÀI LIỆU THAM KHẢO

<https://vuejs.org/>