

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**

---



**BÁO CÁO ĐỒ ÁN 1**  
**TÌM HIỂU REACTJS-SPRINGBOOT**  
**FRAMEWORK**

**Giảng viên hướng dẫn: ThS. Nguyễn Công Hoan**

**Sinh viên thực hiện:**

Nguyễn Hoàng Thái Dương                      19521409

Nguyễn Âu Duy                                      19521423

**TP.Hồ Chí Minh, 2022**

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**  
**KHOA CÔNG NGHỆ PHẦN MỀM**

---



**BÁO CÁO ĐỒ ÁN 1**  
**TÌM HIỂU REACTJS-SPRINGBOOT**  
**FRAMEWORK**

**Giảng viên hướng dẫn: ThS. Nguyễn Công Hoan**

**Sinh viên thực hiện:**

Nguyễn Hoàng Thái Dương 19521409

Nguyễn Âu Duy 19521423

**TP.Hồ Chí Minh, 2022**

## **LỜI CẢM ƠN**

Nhóm xin gửi lời cảm ơn chân thành đến thầy Nguyễn Công Hoan đã tận tình giảng dạy, hướng dẫn chúng em trong suốt thời gian học vừa qua và các bạn học đã góp ý và giúp đỡ nhóm trong quá trình thực hiện đồ án này.

Do kiến thức và thời gian thực hiện hạn chế, đồ án của nhóm vẫn còn nhiều thiếu sót. Nhóm rất mong nhận được góp ý của Thầy Cô và các bạn để đồ án của nhóm được hoàn thiện.

**Nhóm sinh viên thực hiện**

**Nguyễn Hoàng Thái Dương-Nguyễn Âu Duy**

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

HCM, ngày ... tháng 6 năm 2022

**GVHD**

**ThS. Nguyễn Công Hoan**

# Mục Lục

<b>I. GIỚI THIỆU .....</b>	<b>1</b>
1. Thông tin nhóm .....	1
2. Tổng quan đề tài .....	1
2.1 Giới thiệu đề tài .....	1
2.2 Nội dung đề tài .....	2
2.3 Mục tiêu hướng tới .....	3
3. Công nghệ sử dụng .....	3
3.1 IntelliJ .....	3
3.2 Visual Studio Code.....	4
3.3 Postman .....	4
3.4 Microsoft SQL Server .....	5
3.5 Github .....	5
3.6 Ngôn ngữ Java.....	6
3.7 Ngôn ngữ thiết kế Web HTML-CSS-Javascript .....	6
3.8 Bootstrap.....	8
<b>II. REACTJS .....</b>	<b>9</b>
1. Giới thiệu .....	9
2. Các khái niệm chính trong React (ReactJs).....	10
2.1 Virtual DOM .....	10
2.2 JSX .....	11
2.3 Component .....	12
2.4 Props và State .....	13
2.5 Component Lifecycle .....	14
2.6 React Hook .....	16
3. Cách hoạt động .....	16
4. Nhược điểm .....	17

5. Ưu điểm và Tại sao nên dùng ReactJs.....	17
6. Các framework-package liên quan đến ReactJs .....	18
6.1 React Hook .....	18
6.2 React Router .....	20
6.3 Redux Toolkit.....	24
<b>III. SPRING BOOT.....</b>	<b>28</b>
1. Khái niệm.....	28
2. Nhược điểm của Spring Boot .....	28
3. Ưu điểm của Spring Boot .....	28
4. Các thư viện được sử dụng trong Spring Boot.....	29
4.1 Hibernate .....	29
4.2 Spring Data.....	30
4.3 Spring Security .....	31
4.4 Swagger .....	31
5. Xây dựng back-end cho đồ án bằng Spring Boot.....	32
<b>IV. TRANG WEB MINH HỌA.....</b>	<b>34</b>
1. Giới thiệu ứng dụng web .....	34
1.1 Bài toán.....	34
1.2 Hướng giải quyết.....	34
1.3 Giới thiệu ứng dụng phát triển .....	34
1.4 Source code .....	35
2. Product Requirement Document .....	35
2.1 Objective.....	35
2.2 Release.....	36
2.3 Features.....	37
2.4 Analytics – Phân tích .....	43
2.5 Future Work.....	46

3. Màn hình minh họa.....	47
3.1 Trang chủ.....	47
3.2 Trang sản phẩm quần áo.....	48
3.3 Trang sản phẩm phụ kiện .....	48
3.4 Trang chi tiết sản phẩm .....	49
3.5 Trang giỏ hàng.....	49
3.6 Trang đăng nhập.....	51
3.7 Trang đăng ký.....	52
<b>V.    KIẾN TRÚC.....</b>	<b>53</b>
1. ReactJs .....	53
2. Spring Boot.....	54
<b>VI.    KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN .....</b>	<b>55</b>
1. Kết quả đạt được.....	55
2. Hướng phát triển.....	55
<b>VII.  TÀI LIỆU THAM KHẢO.....</b>	<b>56</b>

## Danh Mục Hình Ảnh

Hình I-1 IntelliJ Logo	3
Hình I-2 VS code Logo	4
Hình I-3 Postman Logo	4
Hình I-4 SQL Server Logo	5
Hình I-5 Github Logo	5
Hình I-6 Java logo	6
Hình I-7 Logo HTML-CSS-Javascript	7
Hình I-8 Logo Bootstrap	8
Hình II-1 Mô hình DOM trong ứng dụng ReactJs	10
Hình II-2 Ví dụ về JSX	11
Hình II-3 Ví dụ về component	12
Hình II-4 Ví dụ props	13
Hình II-5 Component Lifecycle	14
Hình II-6 Top các framework phát triển Web theo năm 2021	18
Hình II-7 useState	19
Hình II-8 useEffect	19
Hình II-9 React-Router	21
Hình II-10 useNavigate	22
Hình II-11 useLocation	22
Hình II-12 useParams	23
Hình II-13 Link	24
Hình II-14 Redux Model	25
Hình II-15 ReduxToolkit Slice, AsyncThunk	26
Hình III-1 Hibernate: các hàm tương ứng sẽ có các anotation query và câu lệnh query tương ứng	29
Hình III-2 Ví dụ về việc sử dụng anotation	30



Hình III-3 Cấu trúc file đồ án	32
Hình IV-1 Trang	47
Hình IV-2 Trang quần áo	48
Hình IV-3 Trang chưa sản phẩm phụ kiện	48
Hình IV-4	48
Hình IV-5 Trang chi tiết sản phẩm	49
Hình IV-6 Trang đặt hàng tổng	50
Hình IV-7 Trang giỏ hàng đặt hàng	51
Hình IV-8 Trang đăng nhập	51
Hình IV-9 Trang đăng ký	52
Hình V-1 Kiến trúc áp dụng	53

# I. GIỚI THIỆU

## 1. Thông tin nhóm

<b>BẢNG PHÂN CÔNG THỰC HIỆN ĐỒ ÁN MÔN HỌC</b>	
Nguyễn Hoàng Thái Dương MSSV: 19521409	Nguyễn Âu Duy MSSV: 19521423
Xử lý front-end	Xử lý back-end
Tìm hiểu ReactJs	Xử lý Database
	Tìm hiểu SpringBoot
<b>SV thực hiện 1</b>  (Ký tên) <b>Nguyễn Hoàng Thái Dương</b>	<b>SV thực hiện 2</b>  (Ký tên) <b>Nguyễn Âu Duy</b>

## 2. Tổng quan đề tài

### 2.1 Giới thiệu đề tài

Với sự phát triển nhảy vọt của công nghệ thông tin hiện nay, Internet ngày càng giữ vai trò quan trọng trong các lĩnh vực khoa học kỹ thuật và đời sống. Internet là tập hợp kết nối giữa các máy tính, là một mạng máy tính toàn cầu mà bất kì ai

cũng có thể truy cập bằng laptop, PC, các thiết bị di động, thiết bị gia dụng thông minh,... Vì thế cùng với sự lớn mạnh của Internet thì hàng loạt các sản phẩm mang tính thương mại ra đời với quy mô từ nhỏ đến lớn ở các ngành nghề, lĩnh vực liên quan và phổ biến, thông dụng ở mọi lĩnh vực khác. Internet đã tạo ra một cuộc cách mạng trao đổi thông tin trong mọi lĩnh vực văn hóa, y tế, xã hội, giáo dục,... nhờ đó góp phần thúc đẩy sự phát triển của thế giới.

Trong đó nổi bật là các sản phẩm, thiết bị thương mại điện tử mà mọi người đều có thể sử dụng dễ dàng và hữu ích. Việc mua bán bây giờ không chỉ thông qua xem hàng – mua hàng trực tiếp mà còn có thể giao dịch qua các trang Web thương mại điện tử một cách tiện lợi nhưng vẫn đảm bảo được sự uy tín, chất lượng từ sản phẩm mà các doanh nghiệp, cửa hàng, siêu thị, cá nhân... cung cấp cho khách hàng cũng như sự hỗ trợ và chính sách bán hàng.

Việc để tạo ra một trang Web bán hàng bắt mắt, tiện lợi cho người mua là một trong những vấn đề đang được quan tâm hàng đầu. Có rất nhiều cách để tạo ra một trang Web như thông qua các phần mềm thứ ba để tạo với những người thích thiết kế giao diện nhưng không có hiểu biết sâu rộng về lập trình hay lập trình với các ngôn ngữ, framework hỗ trợ phát triển ứng dụng Web : .NET, HTML-CSS, Javascript, Bootstrap, Vue.js, Angular... Trong đó nhóm đã chọn **ReactJs** cho phần thiết kế giao diện người dùng (front-end) và framework **Java SpringBoot** cho phần xử lý tác vụ logic, kết nối Database để xây dựng một ứng dụng Web bán hàng online.

## ***2.2 Nội dung đề tài***

Thông qua việc tìm hiểu về Reactjs và SpringBoot nhóm sẽ áp dụng để thiết kế ứng dụng web để có thể hiểu sâu và thành thạo các kiến thức đã tìm hiểu.

Việc sử dụng các package – tính năng – thư viện liên quan đến ReactJs và Spring Boot sẽ được trình bày chi tiết trong quá trình tìm hiểu cũng như thiết kế ứng dụng Web.

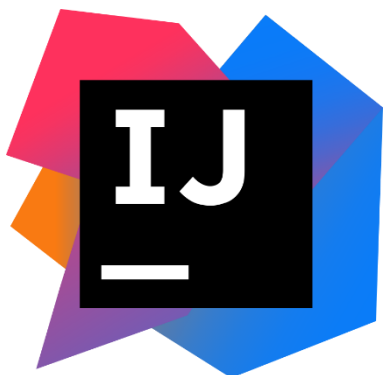
### **2.3 Mục tiêu hướng tới**

- Việc tìm hiểu và áp dụng sẽ giúp các thành viên trong nhóm có cơ hội học hỏi, tiếp thu, mở rộng những kiến thức mới về các framework cũng như cách để xây dựng các ứng dụng từ framework với quá trình thế nào. Từ đó không chỉ giúp nhóm có thêm kinh nghiệm để thiết kế các ứng dụng khác liên quan đến Reactjs và SpringBoot mà còn có thể hỗ trợ cho công việc thực tập, làm việc trong tương lai.
- Ngoài ra nhóm còn muốn dùng những gì mình đã thu hoạch được để có thể giúp những lập trình viên đang muốn tìm hiểu về cả hai framework này để có thể làm tài liệu tham khảo.

## **3. Công nghệ sử dụng**

Những công nghệ, tool,framework liên quan mà nhóm đã sử dụng trong quá trình thực hiện:

### **3.1 IntelliJ**



**Hình I-1 IntelliJ  
Logo**

Được phát hành lần đầu tiên vào năm 2001, IntelliJ IDEA là một môi trường phát triển tích hợp cho Windows, macOS và Linux. Nó chủ yếu được sử dụng để phát triển phần mềm bằng ngôn ngữ lập trình Java, nhưng cũng hỗ trợ các ngôn ngữ khác hoặc tự nhiên hoặc sử dụng một plugin. Nó cũng là cơ sở cho Android Studio của Google, môi trường phát triển

chính thức để tạo các ứng dụng Android. Nhóm sử dụng cho việc code Back-end

### 3.2 Visual Studio Code



**Hình I-2 VS code  
Logo**

Visual Studio Code là một trình soạn thảo mã nguồn được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có chức năng nổi bật cú pháp ( syntax highlighting ), smart code , snippets, và cải tiến mã nguồn. Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác. Nó miễn phí và là phần mềm mã nguồn mở theo giấy phép MIT. Nhóm sử dụng cho việc code Front-end.

### 3.3 Postman



**Hình I-3 Postman Logo**

Postman là một công cụ cho phép chúng ta thao tác với API, phổ biến nhất là REST.

Postman hiện là một trong những công cụ phổ biến nhất được sử dụng trong thử nghiệm các API. Với Postman, ta có thể gọi Rest API mà không cần viết dòng code nào.

Postman hỗ trợ tất cả các phương thức HTTP (GET, POST, PUT, PATCH, DELETE, ...). Bên cạnh đó, nó còn cho phép lưu lại lịch sử các lần request, rất tiện cho việc sử dụng lại khi cần.

Nhóm sử dụng Postman cho việc test API.

### 3.4 Microsoft SQL Server



**Hình I-4 SQL Server  
Logo**

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, database engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

Các thành cơ bản trong SQL Server gồm có: Reporting Services, Database Engine, Integration Services, Notification Services, Full Text Search Service, ... Tất cả kết hợp với nhau tạo thành một giải pháp hoàn chỉnh giúp cho việc phân tích và lưu trữ dữ liệu trở nên dễ dàng hơn.

Nhóm sử dụng CSDL SQL Server làm Database cho ứng dụng.

### 3.5 Github



**Hình I-5 Github Logo**

GitHub là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để

tạo ra các kho chứa của riêng mình để có thể làm việc.

GitHub được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:

Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user.

Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phần mềm thế nào, project phát triển ra sao

Follow user: theo dõi hoạt động của người khác.

Github giúp ta quản lý source code dễ dàng, tracking sự thay đổi version.

### ***3.6 Ngôn ngữ Java***



Java là ngôn ngữ lập trình hướng đối tượng được sử dụng các cú pháp C và C++, thường được dùng trong phát triển phần mềm, trang web, game hay ứng dụng

Java được phát triển bởi James Gosling và đồng nghiệp ở Sun MicroSystem năm 1991 và được phát hành 1994 đến năm 2010 được Oracle mua lại.

**Hình I-6 Java logo**      Các phiên bản của Java:

- Java SE: Là nền tảng cơ bản phát triển giao diện ứng dụng Winform.
- Java EE: Dựa trên SE nhưng dùng để phát triển web.
- Java ME: Phát triển dành cho mobile.

### ***3.7 Ngôn ngữ thiết kế Web HTML-CSS-Javascript***

Đây là bộ ba ngôn ngữ nền tảng cơ bản cho việc thiết kế Web từ lâu đời, ReactJs chỉ là thư viện được mở rộng và đóng gói các dòng lệnh của các ngôn ngữ trên.

Trong đó:



**Hình I-7 Logo HTML-CSS-Javascript**

- **HTML:** (*HyperText Markup Language* – *Ngôn ngữ đánh dấu siêu văn bản*) là một ngôn ngữ đánh dấu được thiết kế để tạo nên các trang web với các mẫu thông tin được trình bày trên WWW. Cùng với CSS, Javascript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các Website.

HTML là cốt lõi của mọi trang web. Bất kể sự phức tạp của một trang web hoặc số lượng công nghệ liên quan. Đó là một kỹ năng thiết yếu cho bất kỳ chuyên gia web. Đó là điểm khởi đầu cho bất cứ ai học cách tạo nội dung cho web. Và thật may mắn cho những bạn mới bắt đầu là HTML rất dễ học.

- **CSS:** (*Cascading Style Sheets*). Ngôn ngữ lập trình này chỉ ra cách các yếu tố HTML của trang web thực sự sẽ xuất hiện trên giao diện của trang. Nếu HTML cung cấp các công cụ thô cần thiết để cấu trúc nội dung trên một trang web thì CSS sẽ giúp định hình kiểu nội dung này để trang web xuất hiện trước người dùng theo một cách đẹp hơn.
- **Javascript:** JavaScript là ngôn ngữ lập trình dựa trên logic. Nó có thể được sử dụng để sửa đổi nội dung trang web. JavaScript là ngôn ngữ lập trình cho phép các nhà phát triển web thiết kế các trang web tương tác.



Nhóm sử dụng để code cho Front-end bởi vì chủ yếu ReactJs sẽ dùng các cú pháp ES6 (thuộc Javascript) và các dòng HTML trong render component. Và SCSS (là tập CSS đã được biên dịch).

### 3.8 *Bootstrap*



Hình I-8 Logo Bootstrap

Bootstrap là một framework bao gồm các HTML, CSS và JavaScript template dùng để phát triển website chuẩn **Responsive-tương thích với mọi phiên bản nền tảng (mobile desktop hay tablet, ...)**.

Nhóm sử dụng Bootstrap 5 là một phiên bản mới của Bootstrap giúp thiết kế ứng dụng trông đẹp mắt hơn.

## II. REACTJS

### 1. Giới thiệu

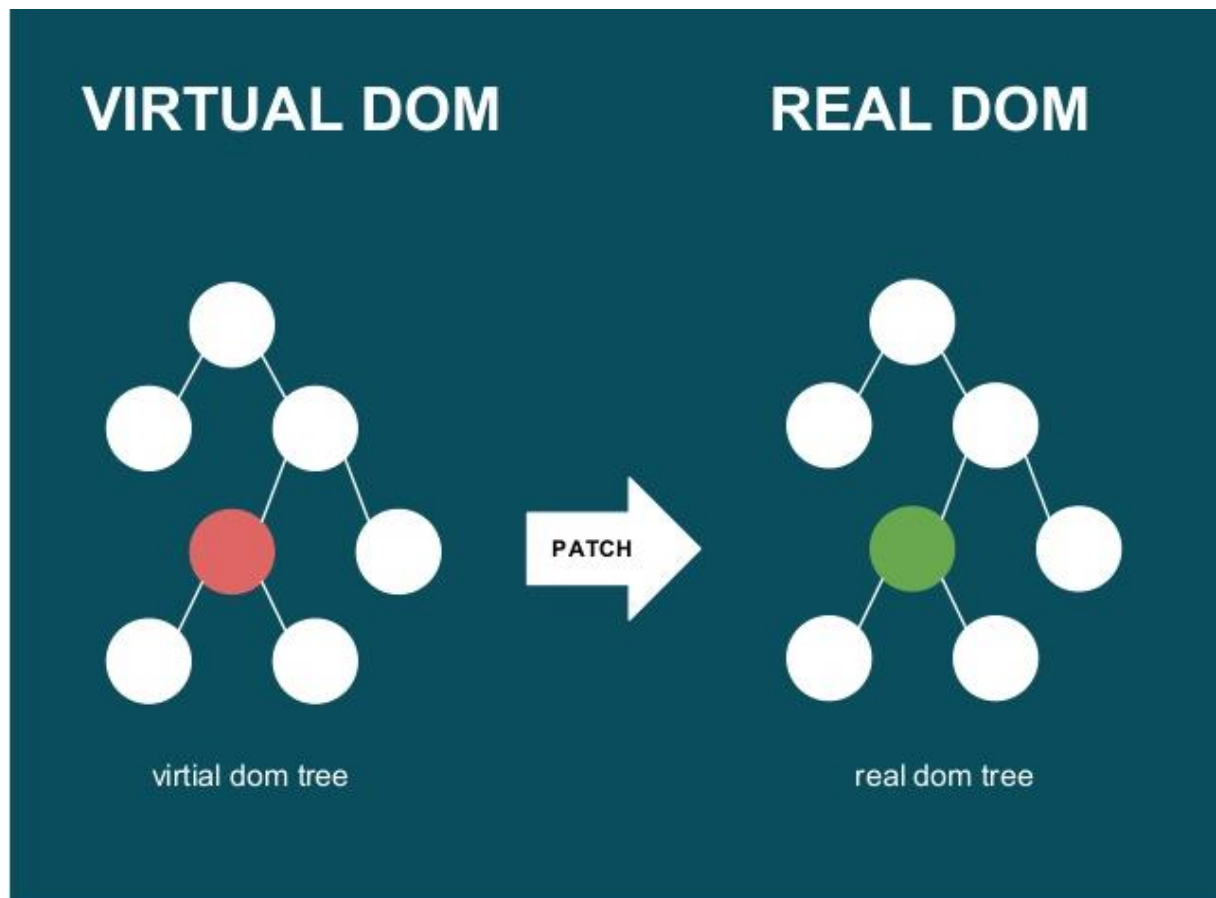
**React** (hay **ReactJs**, **React.js**) là một thư viện Javascript mã nguồn mở để xây dựng các thành phần giao diện có thể tái sử dụng. Nó được tạo ra bởi **Jordan Walke**, một kỹ sư phần mềm tại **Facebook**. Người bị ảnh hưởng bởi **XHP** ( Một nền tảng thành phần HTML cho PHP ). ReactJs lần đầu tiên được triển khai cho ứng dụng **Newsfeed** của **Facebook** năm 2011, sau đó được triển khai cho **Instagram.com** năm 2012. Nó được mở mã nguồn ( open-sourced) tại **JSConf US** tháng 5 2013

ReactJs đang nổi lên trong những năm gần đây với xu hướng Single Page Application (**SPA**). Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code Javascript trong code HTML thông qua các attribute như ng-model, ng-repeat...thì với React là một library cho phép nhúng code HTML trong code Javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa Javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn.

## 2. Các khái niệm chính trong React (ReactJs)

### 2.1 Virtual DOM

Virtual DOM chỉ là một đại diện ảo của DOM, Virtual DOM ra đời đã tăng hiệu suất các ứng dụng ReactJS một cách đáng kể.



**Hình II-1 Mô hình DOM trong ứng dụng ReactJs**

Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.

⇒ Cách hoạt động cũng tựa như commit branch trên Github, Virtual DOM chỉ ghi nhận những thay đổi trên các component xảy ra từ đó chỉ thực việc render lại các component đó trên DOM thật.

## 2.2 JSX

JSX là một dạng ngôn ngữ cho phép viết các mã HTML trong Javascript.

Đặc điểm:

- Nhanh hơn: JSX thực hiện tối ưu hóa trong khi biên dịch sang mã Javascript. Các mã này cho thời gian thực hiện nhanh hơn nhiều so với một mã tương đương viết trực tiếp bằng Javascript.
- An toàn hơn. Ngược với Javascript, JSX là kiểu statically typed, nghĩa là nó được biên dịch trước khi chạy, giống như Java, C++. Vì thế các lỗi sẽ được phát hiện ngay trong quá trình biên dịch. Ngoài ra, nó cũng cung cấp tính năng gỡ lỗi khi biên dịch rất tốt.
- Dễ dàng hơn. JSX kế thừa dựa trên Javascript, vì vậy rất dễ dàng để cho các lập trình viên Javascripts có thể sử dụng.

```
const CatalogNotFound = () => {  
  return (  
    <div>Không tìm thấy kết quả lọc </div>  
  )  
}
```

Hình II-2 Ví dụ về JSX

Như hình trên ta có thể thấy việc viết trực tiếp HTML vào function Javascript mà không có lỗi hay cần các dấu “” như xưa.

## 2.3 Component

Component có cú pháp như một **function**.

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Trong React, chúng ta xây dựng trang web sử dụng những thành phần (component) nhỏ. Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác. Mỗi component trong React có một trạng thái riêng, có thể thay đổi, và React sẽ thực hiện cập nhật component dựa trên những thay đổi của trạng thái. Mọi thứ React đều là component.

```
const Layout = () => {  
  return (  
    <BrowserRouter>  
      <Header />  
      <div className='container'>  
        <React.Fragment>  
          <div className="main">  
            <ARoutes />  
          </div>  
        </React.Fragment>  
      </div>  
      <Footer />  
      <ProductViewModal />  
      <LoginModal />  
      <RegisterModal />  
      <AlertMessage />  
    </BrowserRouter>  
  )  
}  
  
export default Layout
```

Hình II-3 Ví dụ về component

Ta có thể trong hình thì component cha Layout chính là nơi chứa các component con khác. Và component Layout cũng chính là giao diện chính của

Web. Việc thay đổi các các component con chỉ ảnh hưởng đến các component con cần render.

Ví dụ như component LoginModal sẽ là Modal đăng nhập của trang Web và Modal này sẽ được hiện khi người dùng click đăng nhập. Điều này có nghĩa các component khác khi mà LoginModal xuất hiện hay ẩn đi đều không thay đổi vì chỉ có Login Modal là được thay đổi thuộc tính để render lại cho người dùng thấy.

## 2.4 Props và State

- **Props:**

Viết tắt của **properties**, nhưng trong React lại là một khái niệm. Props là một đối tượng. Nó sẽ lưu trữ những thuộc tính, giá trị được truyền vào từ component cha hay trong hàm hay bất cứ khi nào Tag của component này được gọi.

```
function Welcome(props) {  
  return <h1>Hello, {props.name}</h1>;  
}  
  
const element = <Welcome name="ReactJS" />;  
ReactDOM.render(  
  element,  
  document.getElementById('root')  
);
```

**Hình II-4 Ví dụ props**

Ta có thể thấy element là một biến được gán cho giá trị là một component Welcome và được truyền trong Tag với giá trị name="ReactJs".

Và trong component Welcome thì ta có thể thấy việc truy cập names thông qua props. Ở đây props được là một đối tượng nhận vào giá trị name như sau:

```
props = {  
  name: "ReactJs"  
}.
```

⇒ Lưu ý là chúng ta không nên thay đổi được props trong component con do ReactJs tuân theo quy tắc **pure function**: không làm thay đổi giá trị đầu vào

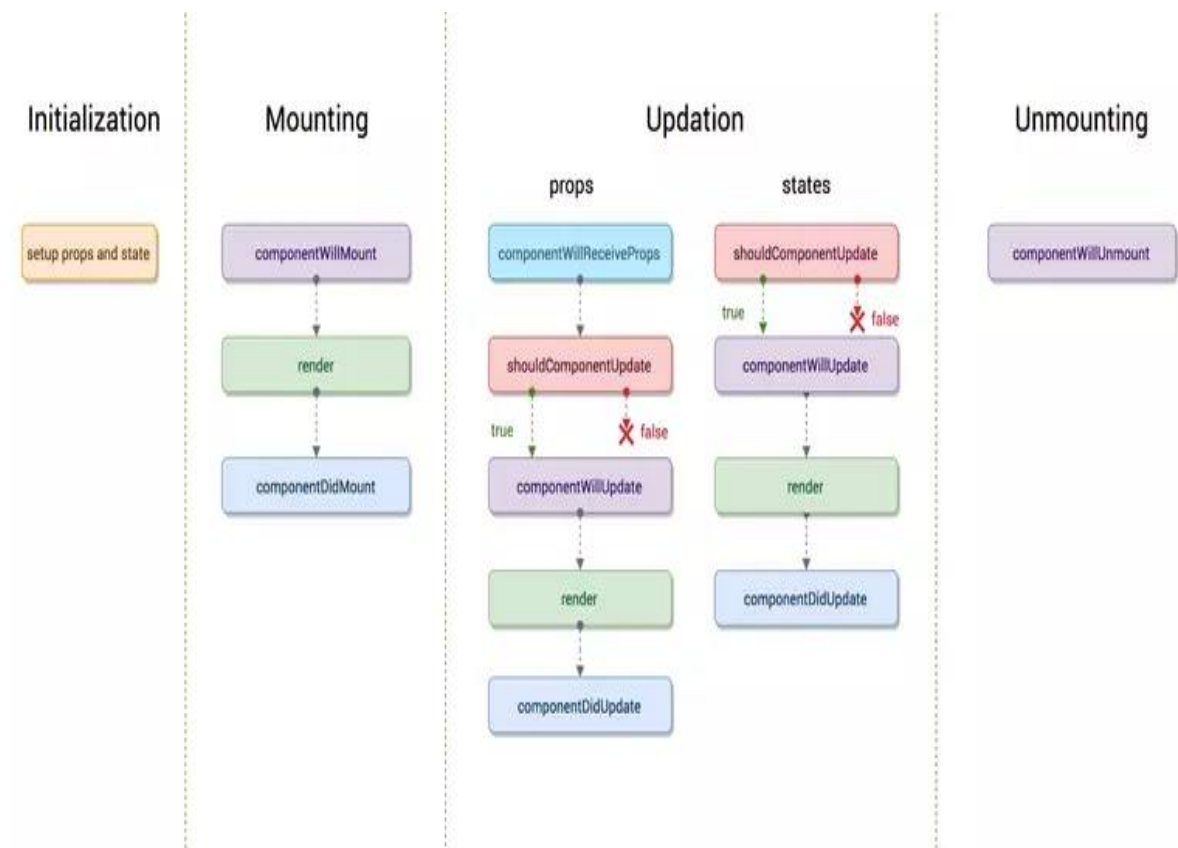
- **State:**

State cũng có vai trò như Props, nhưng là lưu trữ dữ liệu động của một component.

Do đó State thường được dùng để theo dõi sự thay đổi bên trong component và render lại.

## 2.5 Component Lifecycle

Trong React Component, components được khởi tạo (hiển thị ra DOM), update, và kết thúc (unmount), đó được gọi là một component life cycle.



Hình II-5 Component Lifecycle

React cho phép chúng ta tham gia vào các giai đoạn của mỗi component bằng cách sử dụng các phương thức được xây dựng sẵn trong mỗi giai đoạn đó. Khi một component được khởi chạy nó sẽ phải trải qua 4 giai đoạn chính:

- **Initialization:**

Đây là giai đoạn mà component sẽ bắt đầu khởi tạo state và props.

- **Mounting:**

Giai đoạn này thực hiện sau khi quá trình initialization (khởi tạo) được hoàn thành. Nó sẽ chuyển virtual DOM thành DOM và hiển thị lên người dùng. Component sẽ được render lần đầu tiên.

- **componentWillMount ()**

Được khởi chạy khi một component chuẩn bị được mount (tức là trước khi thực hiện render)

- **()**

Được gọi khi component đã được mount (render thành công).

- **Updating:**

Khi state hoặc props của component thay đổi, các hàm trong nhóm này sẽ được thực thi và quyết định xem có cần phải render lại UI hay không. Trường hợp UI không được re-render khi hàm `shouldComponentUpdate ()` trả về giá trị `false`.

- **Unmounting:**

Đây là bước cuối cùng trong mỗi component, khi tất cả các tác vụ hoàn thành và bạn tiến hành unmount DOM.



## 2.6 React Hook

Trước khi react Hook ra đời thì chúng ta thường khai báo một component là class component và sử dụng các phương thức trong lifecycle một cách nhập nhằng. Trong khi đó functional component thì không thể dùng state và lifecycle dù cách trình bày dễ hiểu hơn class component.

Hooks là một bổ sung mới trong React 16.8.

Hooks là những hàm cho phép bạn “kết nối” React state và lifecycle vào các functional component.

Nhóm hầu như sử dụng React Hook cho đồ án vì nó dễ dùng và dễ hiểu hơn class component xong cũng không phủ nhận class component vẫn sẽ là đối tượng được nhóm nghiên cứu tổng tương lai vì trong một số trường hợp thì class component vẫn có thể mang lại hiệu quả.

## 3. Cách hoạt động

ReactJs là một thư viện hỗ trợ cho việc phát triển các ứng dụng Web SPA – single page application. Điều này có nghĩa là ứng dụng chỉ có một trang ( thông thường là index.html ) cho cả ứng dụng và được chia thành các thành phần – component. Mỗi component sẽ chứa các props- được truyền từ component cha và state- lưu trữ các giá trị có trong component để render lại khi có thay đổi logic hay sự kiện từ phía người dùng.

Chính vì vậy ứng dụng sẽ không cần render lại mà chỉ component thay đổi render lại, từ đó giúp tăng đáng kể hiệu suất người dùng và giúp ích trong việc phát triển do có thể tái sử dụng ở nhiều nơi trong ứng dụng.

## 4. 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.

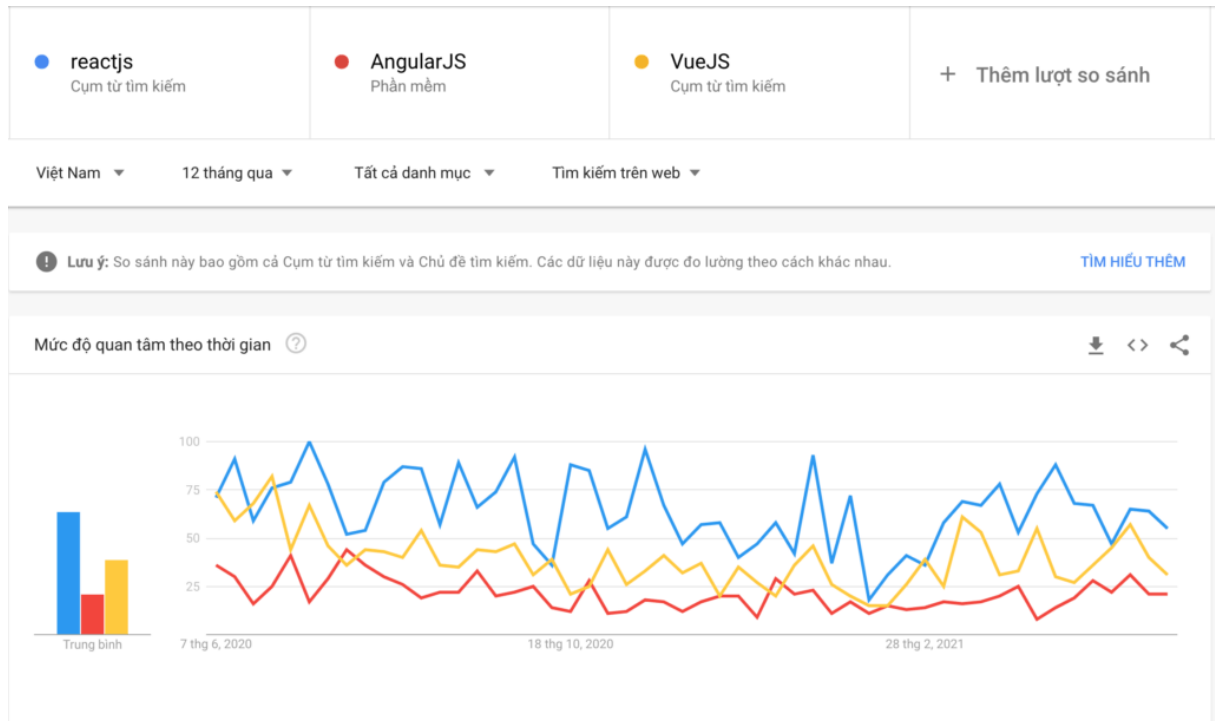
Khó tiếp cận cho người mới học Web.

## 5. Ưu điểm và Tại sao nên dùng ReactJs

Ngoài việc hỗ trợ xây dựng giao diện nhanh, hạn chế lỗi trong quá trình code, cải thiện hiệu suất website thì những tính năng đặc biệt dưới đây có thể là lý do khiến bạn bắt đầu tìm hiểu nó từ bây giờ:

- Phù hợp với đa dạng thể loại website: ReactJS khiến cho việc khởi tạo website dễ dàng hơn bởi vì bạn không cần phải code nhiều như khi tạo trang web thuần chỉ dùng JavaScript, HTML.
- Tái sử dụng các Component
- Có thể sử dụng cho cả Mobile application: React Native – một framework khác được phát triển cũng chính Facebook tương tự như ReactJs

- Debug dễ dàng: Facebook đã phát hành một Chrome extension dùng trong việc debug trong quá trình phát triển ứng dụng.



**Hình II-6 Top các framework phát triển Web theo năm 2021**

- Hot khi tính tới 2021: Nếu bạn nhìn vào số liệu thống kê từ Google Trend ở Việt Nam ở hình bên dưới, dạo lướt qua các trang tuyển dụng hàng đầu ở Việt Nam như Topdev, Itviec,...bạn sẽ thấy số lượng tuyển dụng cho vị trí React Developer là cực kỳ lớn cùng với mức lương vô cùng hấp dẫn và độ phổ biến hiện tại của ReactJS trên thị trường Việt Nam là như thế nào.

## 6. Các framework-package liên quan đến ReactJs

### 6.1 React Hook

Khiến các component trở nên gọn nhẹ hơn.

Giảm đáng kể số lượng code, dễ tiếp cận.

Cho phép chúng ta sử dụng state ngay trong function component

Các Hooks:

### 6.1.1 *useState*

Hàm này nhận đầu vào là giá trị khởi tạo của 1 state và trả ra 1 mảng gồm có 2 phần tử, phần tử đầu tiên là state hiện tại, phần tử thứ 2 là 1 function dùng để cập nhật state.

```
const [isLoading, setLoading] = useState(false);

onClick() {
  setLoading(true)
}
```

Hình II-7 *useState*

### 6.1.2 *useEffect*

Nó giúp chúng ta xử lý các side effects, *useEffect* sẽ tương đương với các hàm **componentDidMount**, **componentDidUpdate** và **componentWillUnmount** trong *LifeCycle*.

```
useEffect(
  () => {
    const subscription = props.source.subscribe();
    return () => {
      subscription.unsubscribe();
    };
  },
  [props.source], // giá trị được subscribe
);
```

Hình II-8 *useEffect*

*useEffect* có thể không cần code cleanup như hàm *return*.

Có 2 loại *useEffect*:

- `useEffect` with no dependencies

`useEffect (function ())`

`useEffect` này sẽ luôn thực thi khi hàm render được gọi trong quá trình mà component tồn tại, một số trường hợp sẽ dẫn đến việc loop vô tận.

- `useEffect` with dependencies

Ở hình trên ta có thể `useEffect` nhận vào một dependencies là `props.source` điều này có nghĩa là `useEffect` sẽ thực thi mỗi khi `props.source` thay đổi giá trị bao gồm lúc component mount (`props.source` được khởi tạo). Nếu như ta truyền vào là một state thì `useEffect` sẽ thực thi mỗi khi `setState`.

Nếu ta không truyền bất cứ giá trị nào hay state nào thì `useEffect` sẽ chỉ thực thi mỗi khi component được gọi (mount). Vd:

`useEffect(function(),[])` đây là `useEffect` with dependencies trường hợp không truyền giá trị hay state.

`useEffect(function(),[state,props,A,B])` đây là `useEffect` with dependencies. Trường hợp truyền nhiều giá trị hay state và sẽ căn cứ xem giá trị.state nào thay đổi mà thực thi `function()` đã được định nghĩa.

### 6.1.3 *useRef*

Dùng để lưu trữ giá trị của một biến qua các lần render. Sau mỗi lần render, giá trị của một số biến trong component sẽ quay lại giá trị ban đầu, sử dụng `useRef` để lưu trữ các giá trị này.

Điều này giúp ta có thể quản lý một số tag mà không cần phải gọi `document.getElementsByTagName` như trong Javascript thuần.

## 6.2 *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 bạn định tuyến "luồng dữ liệu" (data flow) trong ứng dụng của bạn một cách rõ ràng. Nó tương đương với sự khẳng định, nếu bạn có URL này, nó sẽ tương đương với Route này, và giao diện tương ứng.

Phiên bản hiện tại nhóm đang dùng là react-router v6.

### 6.2.1 *BrowserRouter, Routes và Route*

Một bộ cơ bản của react-router trong việc điều hướng trong ReactJs.

Mỗi một project Reactjs chỉ nên có một Browser Router bao bọc toàn bộ chương trình.

Có thể có nhiều Routes để phân chi việc điều hướng trong mỗi Component.

Route chỉ có thể được bọc bởi Routes và Routes chỉ chứa các children là Route. Mặc định khi vào chương trình thì ta sẽ được đưa tới Route có path là “/”. Thông thường sẽ page Home/ Trang chủ

Để điều hướng ta sẽ dùng thẻ <Link/> trong thư viện react-router-dom

Để điều hướng ta sẽ useNavigate trong thư viện react-router-dom

Truy cập pathname bằng useLocation hay useParams để truy cập URL

```
const ARoutes = () => {  
  return (  
    <BrowserRouter>  
      <Routes>  
        <Route path="/" element={<Home />}></Route>  
        <Route path="/catalog/:slug" element={<Product />}></Route>  
        <Route path="/catalog" element={<Catalog />}></Route>  
        <Route path="/cart" element={<Cart />}></Route>  
        <Route path="/accessories" element={<Accessories />}></Route>  
        <Route path="/contact" element={<Contact />}></Route>  
        <Route path="/order" element={<Order />}></Route>  
        <Route path="/customer" element={<CustomerInfo />}></Route>  
      </Routes>  
    </BrowserRouter>  
  )  
}  
  
export default ARoutes
```

Hình II-9 React-Router

### 6.2.2 *useNavigate*

```
// v6
import { useNavigate } from 'react-router-dom';

function MyButton() {
  let navigate = useNavigate();
  function handleClick() {
    navigate('/home');
  };
  return <button onClick={handleClick}>Submit</button>;
};
```

**Hình II-10 useNavigate**

Khai báo biến khởi tạo bằng `useNavigate` trong component để có thể điều trang thông qua các pathname đã khai báo trước đó trong route.

Ở các phiên bản react-router trước, `useHistory` sẽ đảm nhận vai trò này.

### 6.2.3 *useLocation*

```
import React from 'react';
import { useLocation } from 'react-router-dom';

const Example = props => {
  const location = useLocation();
  console.log(location);

  // ...
};
```

**Hình II-11 useLocation**

Cách khai báo tương tự như `useNavigate`.

`useLocation` sẽ chứa thông tin về path mà bạn đang ở. Nếu bạn đang ở trang admin và trước đó trong route bạn khai báo path của route chứa trang admin là `path="/admin"` thì `location.pathname = "/admin"`.

### 6.2.4 useParam

```
1  import * as React from 'react';
2  import { Routes, Route, useParams } from 'react-router-dom';
3
4  function ProfilePage() {
5    // Get the userId param from the URL.
6    let { userId } = useParams();
7    // ...
8  }
9
10 function App() {
11   return (
12     <Routes>
13       <Route path="users">
14         <Route path=":userId" element={<ProfilePage />} />
15         <Route path="me" element={...} />
16       </Route>
17     </Routes>
18   );
19 }
```

**Hình II-12 useParam**

useParam cho phép bạn lấy các param trên url website.

Các param này được truyền vào bằng cách thêm vào path trong route theo sau các dấu :,?,....

### 6.2.5 Link

Thẻ <Link/> tương tự như thẻ <a/> trong HTML và chức năng như useNavigate nhưng <link/> sẽ được render ra như một component, element UI trên website để người dùng tương tác như các click, ...



```

1  import * as React from "react";
2  import { Link } from "react-router-dom";
3
4  function UsersIndexPage({ users }) {
5    return (
6      <div>
7        <h1>Users</h1>
8        <ul>
9          {users.map((user) => (
10            <li key={user.id}>
11              <Link to={user.id}>{user.name}</Link>
12            </li>
13          ))}
14        </ul>
15      </div>
16    );
17  }

```

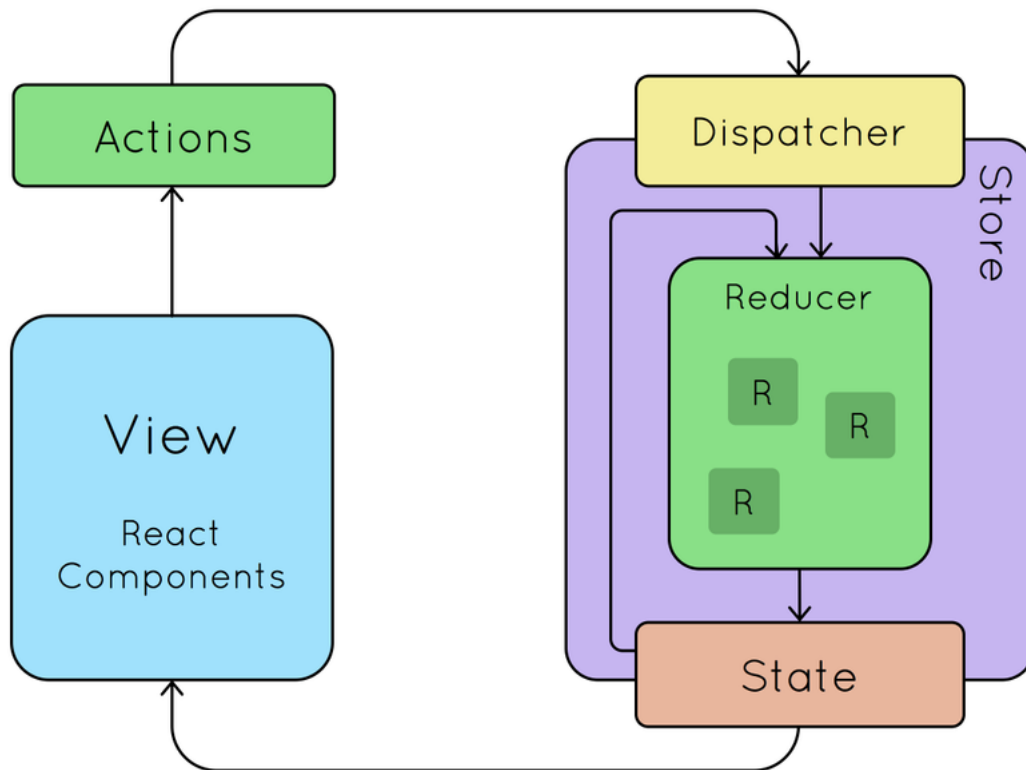
Hình II-13 Link

## 6.3 Redux Toolkit

### 6.3.1 React Redux - Redux toolkit

- **Redux** là một thư viện Javascript giúp tạo ra thành một lớp quản lý trạng thái của ứng dụng. Tránh trường hợp phải truyền các props từ component cha sang component con và sâu hơn nữa. Vì có những props mà component con này không dùng tới nhưng vẫn phải nhận và truyền lại cho component con sau. Vì thế Redux ra đời để giải quyết vấn đề này.
- **Redux toolkit:** là một package được sinh ra để tiết kiệm thời gian trong việc cấu hình store, các file action hay reducer trong redux khá tách biệt và code lặp lại khá nhiều. Tuy nhiên, đến với redux toolkit, việc config store, action, reducer trở nên dễ dàng, nhanh chóng và ngắn gọn.

Các khái niệm: store, action, reducer và nguyên lý hoạt động



**Hình II-14 Redux Model**

- **Store** là nơi lưu trạng thái của ứng dụng và là duy nhất, có thể lưu trữ, truy xuất hoặc cập nhật giá trị state trong store thông qua các action.
- **Action**: là các event, các event này bao gồm type (để reducer biết đây là loại action gì, từ đó thực hiện các hành động cập nhật state thích hợp) và payload (chứa thông tin state mới), các payload này thông thường sẽ được cấu hình từ View hay UI tại nơi người dùng tạo ra các sự kiện như click, ...
- **Reducer**: sau khi action được gọi thông qua dispatch reducer là các pure function lấy state hiện tại, kết hợp với type, payload từ action để cập nhật giá trị state mới.

Các tính năng - tiện ích của redux được thay thế bởi redux toolkit:

```

import { createSlice,createAsyncThunk } from "@reduxjs/toolkit";

import axios from "axios";
import { apiUrl } from "../../utils/constant";

export const getAllProduct = createAsyncThunk(
  'getAllProduct',
  async (data, { rejectWithValue }) => {
    const rs = await axios.get(`${apiUrl}/product`)
    if (rs.status < 200 || rs.status >= 300) {
      return rejectWithValue(rs.data)
    }
    return rs.data
  }
)

export const productSlice = createSlice({
  name: 'productSlice',
  initialState: {
    value: []
  },
  reducers: {

  },
  extraReducers: (builder) => {
    builder.addCase(getAllProduct.fulfilled, (state, action) => {
      state.value = action.payload
    })
  }
})

export default productSlice.reducer

```

Hình II-15 ReduxToolkit Slice, AsyncThunk

- **createSlice:** Với redux, khi định nghĩa action và reducer phải tách ra thành các file khác nhau, trong khi đó đối với slice, reducer và action được kết hợp lại trong cùng một file.
- **createAsyncThunk:** Thông thường các action, reducer không cho phép truyền vào , hay xử lý các hàm bất đồng bộ, vì createAsyncThunk giúp

cho việc này có thể diễn ra trong Redux. Và các xử lý này sẽ được gửi tới phần **extraReducers** trong slice để có thể cập nhật state.

⇒ Vì vậy nhóm đã sử dụng Redux Toolkit thay vì Redux và Redux Thunk mà vẫn đảm bảo với tính thích hợp với thiết đồ án.

## III. SPRING BOOT

### 1. Khái niệm

Spring Boot là một module của Spring Framework, cung cấp tính năng RAD (Rapid Application Development - Phát triển ứng dụng nhanh). Spring Boot ra đời nhằm rút ngắn thời gian cài đặt và cấu hình Spring MVC Project. Giúp các lập trình viên tập trung hơn vào việc phát triển nghiệp vụ thay vì tốn nhiều thời gian cho việc cấu hình dự án. Spring Boot dễ dàng trong việc tích hợp với các hệ sinh thái của Spring như: Spring JDBC, Spring ORM, Spring Security,

### 2. Nhược điểm của Spring Boot

**Thiếu kiểm soát.** Do style cố định, Spring Boot tạo ra nhiều phụ thuộc không được sử dụng dẫn đến kích thước tệp triển khai lớn.

Quá trình chuyển đổi dự án Spring cũ hoặc hiện có thành các ứng dụng Spring Boot nhiều khó khăn và tốn thời gian.

**Không thích hợp cho các dự án quy mô lớn.** Hoạt động liên tục với các microservices, theo nhiều nhà phát triển, Spring Boot không phù hợp để xây dựng các ứng dụng nguyên khối.

### 3. Ưu điểm của Spring Boot

Spring Boot hỗ trợ nhúng trực tiếp các file Server như Tomcat, Jetty hoặc Undertow (Do đó, khi sử dụng Spring Boot không cần phải deploy ra file war). Điều này hỗ trợ cho lập trình viên rất nhiều trong việc phát triển ứng dụng nhanh và đặc biệt là tiết kiệm được rất nhiều thời gian trong việc deploy ứng dụng Spring.

Khác với Spring Framework đơn thuần, Spring Boot hỗ trợ việc tự động cấu hình Spring khi cần thiết. Và đặc biệt Spring Boot không sinh ra code cấu hình và cũng không cần cấu hình bằng XML. Những việc cấu hình thường rất tốn thời

gian trong việc phát triển một ứng dụng. Với tiêu chí phát triển một ứng dụng nhanh,

Spring Boot đã giúp cho người dùng giảm thiểu lượng thời gian đáng kể cho việc cấu hình. Spring Boot còn được biết đến là chuẩn cho Microservices (Cloud support, giảm việc setup, config các thư viện hỗ trợ). Do đó, việc triển khai Microservices trên các ứng dụng Spring Boot là rất dễ dàng.

## 4. Các thư viện được sử dụng trong Spring Boot

### 4.1 Hibernate

Khái niệm: Hibernate là một ORM (object-relationship-mapping: công nghệ cho phép chuyển đổi từ các object trong ngôn ngữ hướng đối tượng sang database quan hệ và ngược lại) framework mã nguồn mở giúp viết các ứng dụng Java có thể tham chiếu các đối tượng (entity) với bảng (table) trong hệ quản trị cơ sở dữ liệu.

Hibernate giúp thực hiện việc giao tiếp giữa tầng ứng dụng và tầng dữ liệu (kết nối, truy suất, lưu trữ...), cải thiện các vấn đề khi sử dụng JDBC như việc không cần phải lặp đi lặp lại các câu lệnh để truy xuất vào database, thay vào đó ta sẽ viết sẵn các kiểu query cần thiết và chỉ cần gọi đến khi cần truy xuất vào database.

```
@Query("SELECT new com.example.Backend.dto.ProductDto(p.id, p.title, p.id_cate, p.category,
    \"p.size, p.descriptions) FROM Product p")
List<ProductDto> getAll();

1 usage  👤 Duy
@Query("SELECT new com.example.Backend.dto.ProductDto(p.id, p.title, p.id_cate, p.category,
    \"p.descriptions) FROM Product p WHERE p.id = ?1")
ProductDto getDetail_byID(String id);
```

**Hình III-1 Hibernate: các hàm tương ứng sẽ có các anotation query và câu lệnh query tương ứng**

Sử dụng Hibernate là độc lập với hệ quản trị cơ sở dữ liệu, nghĩa là ta không cần thay đổi câu lệnh HQL khi ta chuyển từ hệ quản trị CSDL MySQL sang Oracle, hay các hệ quản trị CSDL khác... Do đó rất dễ để ta thay đổi CSDL quan hệ, đơn giản bằng cách thay đổi thông tin cấu hình hệ quản trị CSDL trong file cấu hình.

Hibernate còn giúp cho việc mapping giữa Object Java và Table tương ứng trong database tương đối dễ dàng thông qua việc sử dụng các anotation

A screenshot of a code editor showing Java code. The code is as follows:

```
8  
9  
10  
11  
12  
13  
14  
15  
16  
4 usages  Duy +1  
@Entity  
@Table(name = "Product")  
public class Product {  
    3 usages  
    @Id  
    private String id;  
    3 usages  
    private String title;  
    3 usages  
    private String id_cate;  
    3 usages  
    private String categorySlug;  
    3 usages
```

Hình III-2 Ví dụ về việc sử dụng anotation

## 4.2 Spring Data

Spring Data JPA là giảm thiểu việc thực hiện quá nhiều bước để có thể implement được JPA. Spring Data JPA là một phần của Spring Data và nó hỗ trợ Hibernate 5, OpenJPA 2.4 và EclipseLink 2.6.1. Trong đồ án này, chúng em sử dụng JPA như một interface và Hibernate để thực hiện interface này.

-Cách hoạt động: Trong lập trình java, khi phải truy suất dữ liệu từ database, ta cần phải khởi tạo một class DAO (Data access object) ứng với object mà ta cần

truy suất để lấy, lưu hoặc xóa dữ liệu. Vì vậy class DAO tốn rất nhiều thời gian để hoàn thiện, Spring data đơn giản hóa việc này bằng cách loại bỏ hoàn toàn việc triển khai class DAO mà chỉ sử dụng nó như một interface. Một interface DAO sẽ cần extends JPA specific Repository interface, JpaRepository. Điều này cho phép Spring Data tìm thấy interface này và tự động tạo 1 implementation cho nó.

-Bằng cách extends interface, chúng ta nhận được các method CRUD (Create, Read, Update, Delete) phù hợp nhất để truy cập dữ liệu tiêu chuẩn có sẵn trong một DAO tiêu chuẩn.

### *4.3 Spring Security*

-Khái niệm: Spring security là 1 framework thuộc hệ thống Spring, dành riêng cho việc thiết lập bảo mật của ứng dụng bao gồm authentication và authorization.

- Authentication: quá trình xác minh user, dựa vào thông tin đăng nhập mà user cung cấp. Ví dụ khi login, bạn nhập username và password, nó giúp hệ thống nhận ra bạn là ai.
- Authorization: Quá trình xác định xem user có quyền thực hiện những chức năng nào của hệ thống (đọc/sửa/xóa data), sau khi user đã authenticated thành công.

Hiểu nôm na thì authentication là cái công thứ nhất, xem user có thuộc hệ thống hay không, authorization là cái công thứ hai, xem user được phép làm những gì trong hệ thống đó.

### *4.4 Swagger*

Open API Specification: là một định dạng mô tả API dành cho REST APIs. Một file OpenAPI cho phép bạn mô tả toàn bộ API bao gồm cả:

- Cho phép những endpoints (/users) và cách thức hoạt động của mỗi endpoint (GET /users, POST /users)



- Các tham số đầu vào & đầu ra của từng hoạt động
- Phương thức xác thực
- Thông tin liên lạc, chứng chỉ, điều khoản sử dụng và những thông tin khác

API specifications có thể được viết bằng YAML hoặc JSON. Định dạng này dễ đọc, dễ hiểu cho cả người dùng lẫn ngôn ngữ máy tính

Swagger là một bộ công cụ mã nguồn mở để xây dựng **OpenAPI specifications** giúp bạn có thể thiết kế, xây dựng tài liệu và sử dụng REST APIs.

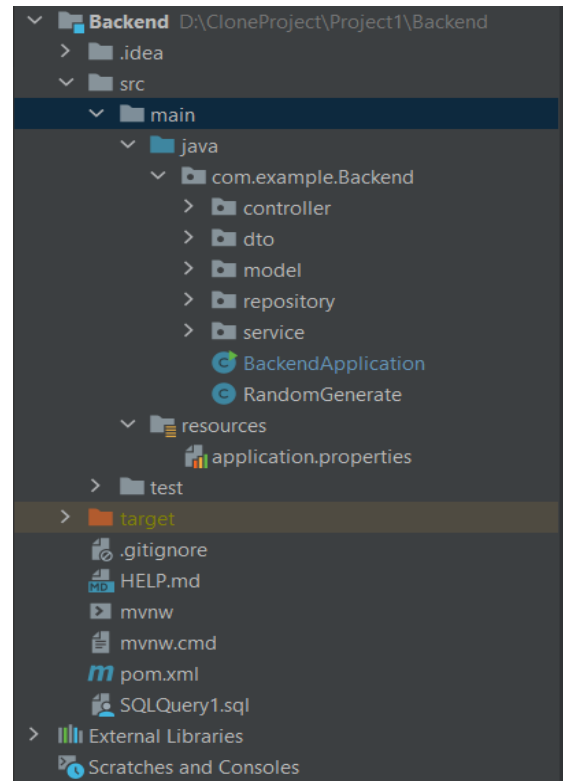
## 5. Xây dựng back-end cho đồ án bằng Spring Boot

Ở đồ án này, chúng em đã tìm hiểu và xây dựng một codebase để triển khai ứng dụng Spring Boot làm phần backend cho website của đồ án, codebase này tương đối hoàn chỉnh và đáp ứng được các nhu cầu như dễ đọc, dễ triển khai, bảo trì, nâng cấp, sửa chữa và hoàn toàn có thể được tái sử dụng cho các đồ án khác có liên quan.

Các file và tệp quan trọng của đồ án:

**Tệp model** chứa các model là các class object cần quản lí. Các object này đóng vai trò là entity có khả năng mapping đến các table bên dưới database

**Tệp dto** chứa các class dto (data transfer object) được sử dụng làm chỗ chứa tạm



**Hình III-3 Cấu trúc file đồ án**

thời cho dữ liệu trong quá trình đọc, viết, ghi, xóa dữ liệu từ database lên trên hoặc ngược lại.

**Tập repository** chứa các repository của đồ án, các repository này kế thừa từ JPA repository

Tập service chứa 2 phần:

- Các interface của service tương ứng với từng model mà service đó hướng đến.
- Các implement của các service interface.

**Tập controller** chứa các class controller (các RestController trong Spring Boot) các controller này giúp điều hướng và mapping các request được gửi đến từ server

**File pom.xml** chứa các thông tin cần thiết để định nghĩa project và các plugin cũng như các dependency được khai báo và sử dụng trong project Spring Boot

**File BackendApplication** là file java chính để chạy project Spring Boot, file này còn có các hàm cấu hình security của project.

## IV. TRANG WEB MINH HỌA

### 1. Giới thiệu ứng dụng web

#### 1.1 Bài toán

Hiện nay càng nhiều cửa hàng, siêu thị quần áo xuất hiện ở khắp mọi nơi để đáp ứng về nhu cầu của người tiêu dùng. Nhiều mẫu mã với nhiều giá cả cạnh tranh trên thị trường. Thế nhưng một số cửa hàng lại vẫn không thu hút được khách hàng dù sản phẩm có chất lượng tốt, giá cả hợp lý. Nguyên do có thể do vị trí cửa hàng, độ quảng bá chưa tới được người tiêu dùng. Cùng với độ lớn mạnh của Internet thì các cửa hàng chỉ bán tại chỗ thì không đủ sức cạnh tranh với các cửa hàng lớn hơn.

Vì thế thị trường cần những trang web thương mại về quần áo,... đảm bảo về chất lượng giống với quảng cáo, chính sách chăm sóc khách hàng sau khi mua sản phẩm.

#### 1.2 Hướng giải quyết

Nhận thấy điều đó nhóm đã đưa ra một hướng phát triển web mua hàng online Sunshine CLOTHES.

Sunshine cung cấp các sản phẩm liên quan đến quần áo, phụ kiện, cho khách hàng đảm bảo về chất lượng, giá thành và những trải nghiệm tốt hơn cho người tiêu dùng.

#### 1.3 Giới thiệu ứng dụng phát triển

Về vắn front-end nhóm đã chọn ReactJs vì những ưu điểm mà framework này mang lại:

- Quản lý code và mở rộng sản phẩm dễ dàng.
- Hiệu suất mà ReactJs mang lại giúp tăng trải nghiệm người dùng.

- Ngày càng phổ biến nên được rất nhiều thư viện mở rộng khác hỗ trợ, cộng đồng lớn mạnh, dễ dàng trong việc hỏi đáp.

Về phần back-end Spring Boot là một lựa chọn tuyệt vời:

- Spring Boot cũng được nhiều thư viện khác hỗ trợ như ReactJs, từ đó có thể giúp nhóm giải quyết được yêu cầu thực tế ngày nay.
- Dễ dàng bảo trì cũng như nâng cấp lên Microservice trong tương lai khi Spring Boot là chuẩn hướng phát triển này.
- Ngoài ra cũng có thể nói Java là ngôn ngữ lập trình hướng đối tượng mạnh, phổ biến nên việc sử dụng Spring Boot sẽ giúp các lập trình viên tiếp xúc nhiều với loại ngôn ngữ lập trình rất thông dụng này.

## 1.4 Source code

Project: [yang020501/Project1: Simple E-commerce web \(github.com\)](https://github.com/yang020501/Project1: Simple E-commerce web)

## 2. Product Requirement Document

### 2.1 Objective

Vision	<p>Sunshine muốn trở thành một trang web có thể cung cấp các sản phẩm quần áo năng động, mới mẻ, trẻ trung</p> <p>Sunshine luôn đảm bảo chất lượng và dịch vụ chăm sóc khách hàng tốt nhất.</p>
Goals	Là trang web sẽ được mọi người nghĩ tới khi mua sắm quần áo.

Initiatives	<p>Luôn cập nhật những sản phẩm mới nhất, được quan tâm nhiều nhất trong thị trường.</p> <p>Áp dụng các chính sách giao hàng, chăm sóc khách hàng nhằm phục vụ khách hàng.</p> <p>Tìm hiểu thêm về các chiến lược marketing, ưu đãi nhằm tiếp cận đến các khách hàng</p>
Persona(s)	<p>Khách hàng muốn mua sắm những bộ quần áo nhưng không có thời gian để ghé trực tiếp tại tiệm.</p> <p>Khách hàng có nhu cầu mua nhưng khoảng cách địa lý không cho phép.</p> <p>Khách hàng thích mua sắm online.</p>

## 2.2 Release

Release	Sunshine
Date	Dự kiến : 25/06/2022
Initiative	<p>Luôn luôn cập nhật và tìm hiểu tình hình để có thể đưa ra thời gian ra mắt hợp lý nhất.</p> <p>Thu thập đánh giá cộng đồng, thị trường để tăng cường trải nghiệm người dùng cũng như khách hàng.</p>
Milestones	None

Features	Thanh điều hướng Banner sản phẩm Tạo giỏ hàng và đặt hàng Thanh tìm kiếm sản phẩm Lọc theo loại sản phẩm Cập nhật thông tin người dùng Xem thông tin chi tiết sản phẩm Chính sách khách hàng
Dependencies	Release dependencies

## 2.3 Features

### 2.3.1 Thanh điều hướng

Feature	Thanh điều hướng
Description	Điều hướng đến các trang khác nhau của trang web
Purpose	Giúp người dùng có thể nhanh chóng tìm được sản phẩm mình muốn
User problem	Nếu không có thanh điều hướng sẽ khó khăn trong việc tìm kiếm sản phẩm theo ý muốn hay sản phẩm cần tìm trong các trang đã được phân loại

User value	Tiết kiệm thời gian
Assumptions	Đễ nhìn, nằm ở vị trí trực quan người dùng.
Not doing	None
Acceptance criteria	None

### 2.3.2 Banner

Feature	Banner sản phẩm
Description	Đề xuất các sản phẩm mới hoặc đang được quan tâm nhiều
Purpose	Giúp người dùng quảng cáo sản phẩm mới, hot đến khách hàng theo thị trường tiêu dùng
User problem	Khách hàng có thể không quan tâm hoặc khiến trang web không đủ sức hấp dẫn
User value	Quảng bá sản phẩm , sinh động trang web
Assumptions	Thu hút sự chú ý của người dùng, tăng khả năng bán sản phẩm của shop
Not doing	None

Acceptance criteria	Đẹp, màu sắc bắt mắt.
---------------------	-----------------------

### 2.3.3 Tạo giỏ hàng và đặt hàng

Feature	Tạo giỏ hàng và đặt hàng
Description	Lưu trữ các sản phẩm khách hàng muốn mua
Purpose	Tăng trải nghiệm mua của khách hàng
User problem	Nếu không có giỏ hàng và đặt hàng , khách hàng không thể nào mua cũng như ghi nhớ các sản phẩm đã chọn trong quá trình trải nghiệm mua hàng
User value	Tiết kiệm thời gian, giúp khách hàng dễ quản lý sản phẩm và giá tiền
Assumptions	Sẽ là nơi thực hiện việc giao dịch, thanh toán, mua bán sản phẩm giữa người dùng và shop
Not doing	None
Acceptance criteria	Có các form để điền thông tin liên lạc, địa chỉ, thông tin đơn hàng và tổng giá đơn hàng

### 2.3.4 Thanh tìm kiếm sản phẩm

Feature	Thanh tìm kiếm sản phẩm
---------	-------------------------



Description	Tìm kiếm nhanh sản phẩm mà khách hàng muốn khi đã biết được sản phẩm
Purpose	Tìm được sản phẩm muốn mua mà không cần phải tìm kiếm trong các trang một cách thủ công.
User problem	Khiến khách hàng tốn thời gian trong việc tìm kiếm sản phẩm
User value	Tiết kiệm thời gian qua việc tìm được sản phẩm theo mong muốn một cách nhanh chóng
Assumptions	None
Not doing	None
Acceptance criteria	Dễ nhìn thấy, kết quả tìm kiếm rõ ràng

### 2.3.5 Lọc theo loại sản phẩm

Feature	Lọc theo loại sản phẩm
Description	Tìm kiếm sản phẩm thông qua hệ thống phân loại như màu sắc , kích cỡ, loại sản phẩm.
Purpose	Tìm được các sản phẩm muốn mua mà không cần phải tìm kiếm trong các trang như kéo kéo chuột. Phân loại sản phẩm cũng sẽ giúp hàng chọn được các sản phẩm theo ý thích cá nhân

User problem	Khiến khách hàng tốn thời gian trong việc tìm kiếm sản phẩm nếu có quá nhiều loại, kiểu sản phẩm
User value	Tiết kiệm thời gian qua việc tìm được sản phẩm theo mong muốn một cách nhanh chóng
Assumptions	None
Not doing	None
Acceptance criteria	Có tick box, phân loại rõ ràng

### 2.3.6 Cập nhật thông tin người dùng

Feature	Cập nhật thông tin người dùng
Description	Chỉnh sửa các thông tin cá nhân như tên, sđt, địa chỉ giao hàng,..
Purpose	Giúp cho việc mua hàng tại Sunshine thuận tiện hơn khi có thay đổi
User problem	Khiến khách hàng không biết phải đặt hàng cũng như mua hàng thế nào nếu muốn thay đổi thông tin cá nhân. Sẽ khiến việc mua sắm trở nên khó khăn và rườm rà hơn.

User value	Thuận tiện cho khách hàng dễ dàng cập nhật lại thông tin để tiện giao hàng cũng như mua hàng.
Assumptions	None
Not doing	None
Acceptance criteria	None

### 2.3.7 Chính sách khách hàng

Feature	Chính sách khách hàng
Description	Liệt kê các chính sách giao hàng, chăm sóc khách hàng, bảo hành sản phẩm,...
Purpose	Giúp khách hàng hiểu rõ quy định mua và bán.  Giúp khách hàng yên tâm hơn để có thể đặt niềm tin vào mua và sử dụng sản phẩm
User problem	Không an tâm liệu sản phẩm mua có được bảo hành hay hoàn trả hay không.  Không tin dùng nếu như không có chính sách rõ ràng.
User value	Hài lòng và có thể an tâm mua hàng khi đã nắm rõ các chính sách mà cửa hàng đưa ra.

Assumptions	None
Not doing	None
Acceptance criteria	Chính sách rõ ràng, hợp lý

## 2.4 Analytics – Phân tích

### 2.4.1 Tính năng thanh điều hướng

Key performance indicator	Baseline	Target	Timeframe
Phần trăm người dùng tương tác	90%	95%	Sau khi dùng ứng dụng 1 tuần

### 2.4.2 Tính năng banner

Key performance indicator	Baseline	Target	Timeframe
Phần trăm người dùng tương tác	70%	80%	Sau khi dùng ứng dụng 1 tháng

### 2.4.3 Tính năng tạo giỏ hàng và đặt hàng

Key performance indicator	Baseline	Target	Timeframe
---------------------------	----------	--------	-----------

Phần trăm người dùng tương tác	70%	60%	Sau khi dùng ứng dụng 1 tháng
Thời gian người dùng hoàn thành	15p	< 7p	Sau 1 tháng khi thanh toán

#### 2.4.4 Tính năng thanh tìm kiếm sản phẩm

Key performance indicator	Baseline	Target	Timeframe
Phần trăm người dùng tương tác	80%	80%	Sau khi dùng ứng dụng 1 tháng
Thời gian người dùng hoàn thành	2p	< 1p	Sau khi dùng ứng dụng 1 tuần

#### 2.4.5 Tính năng lọc sản phẩm theo loại

Key performance indicator	Baseline	Target	Timeframe
Phần trăm người dùng tương tác	70%	90%	Sau khi dùng ứng dụng 1 tháng

Thời gian người dùng hoàn thành	2p	<1p	Sau khi dùng ứng dụng 1 tuần
---------------------------------	----	-----	------------------------------

#### 2.4.6 Tính năng cập nhật thông tin người dùng

Key performance indicator	Baseline	Target	Timeframe
Phần trăm người dùng tương tác	50%	60%	Sau khi dùng ứng dụng 2 tháng
Thời gian người dùng hoàn thành	10p	<7p	Sau khi dùng ứng dụng 2 tháng

#### 2.4.7 Tính năng chính sách khách hàng

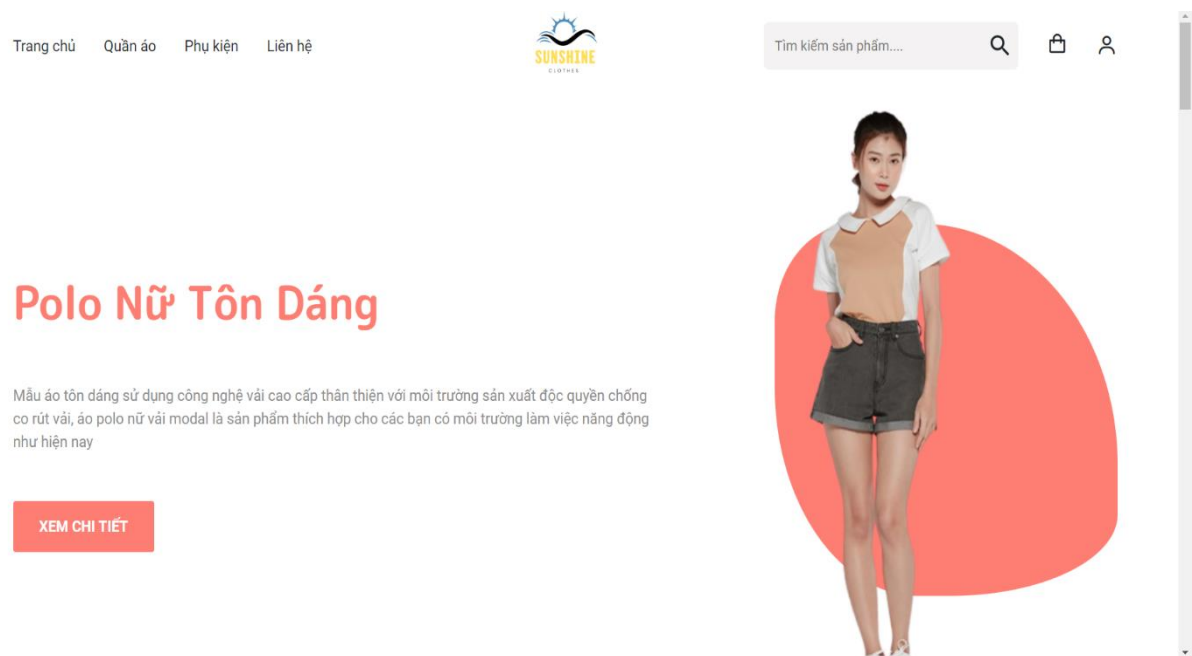
Key performance indicator	Baseline	Target	Timeframe
Phần trăm người dùng tương tác	80%	50%	Sau khi dùng ứng dụng 1 tháng
Thời gian người dùng hoàn thành	10p	<7p	Sau khi dùng ứng dụng 1 tháng

## 2.5 Future Work

Future features	Purpose	Priority	Timeframe
Nhắn tin trực tiếp	Tư vấn khách hàng	1	Sau khi số người dùng đạt 2000 lượt
Đăng nhập qua các mạng xã hội	Thuận tiện cho việc tạo tài khoản cũng như đồng bộ thông tin	4	Sau khi số người dùng đạt 5000 lượt
Thêm khuyến nghị sản phẩm	Giúp khách hàng thấy được sản phẩm nổi trội, hot trên thị trường	3	Sau khi số người dùng đạt 2500 lượt

### 3. Màn hình minh họa

#### 3.1 Trang chủ



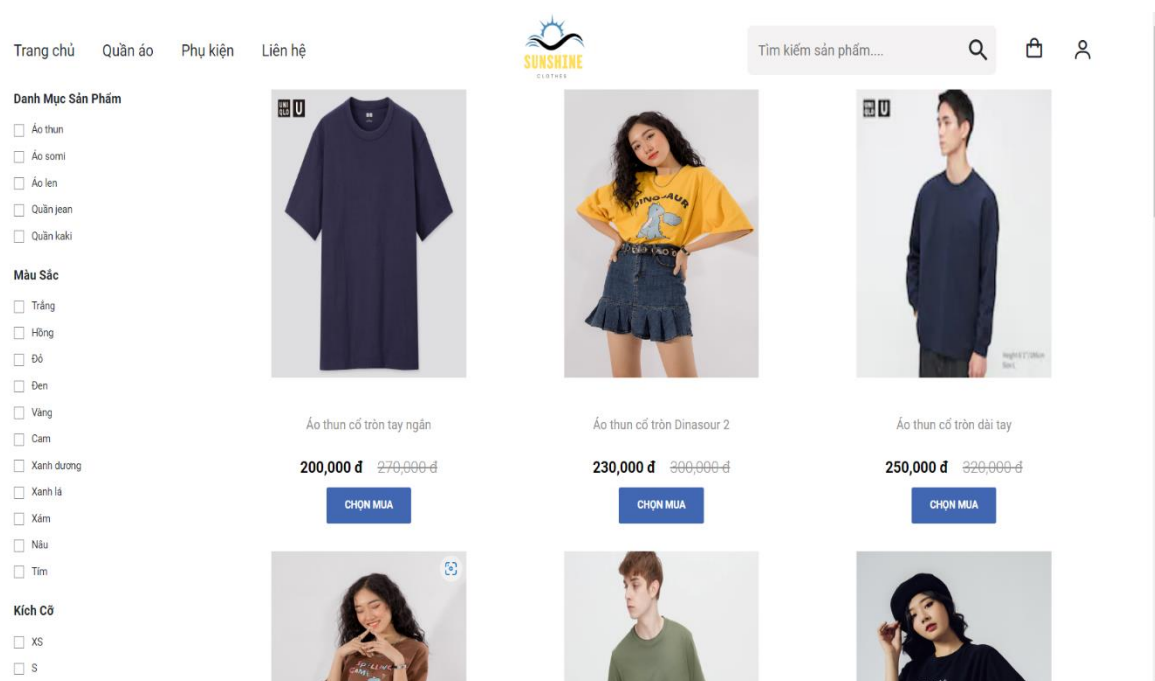
**Hình IV-1 Trang**

Trang chủ sẽ bao gồm các thanh điều hướng, thanh tìm kiếm.

Banner giới thiệu sản phẩm để thu hút khách hàng.



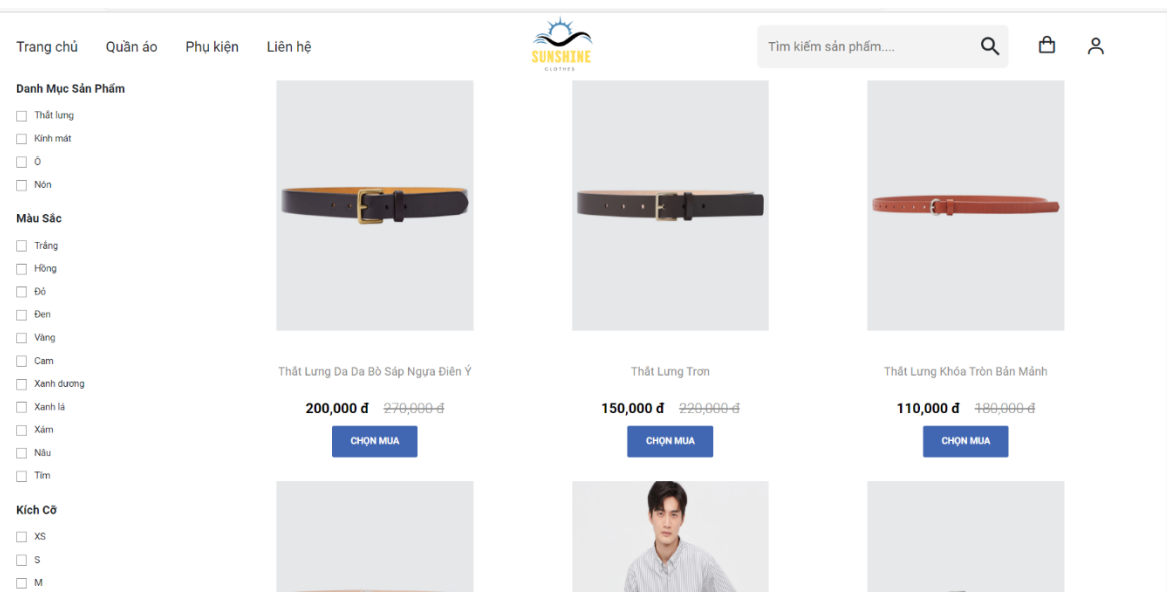
## 3.2 Trang sản phẩm quần áo



Hình IV-2 Trang quần áo

Chứa thanh lọc và các tính năng như trang chủ

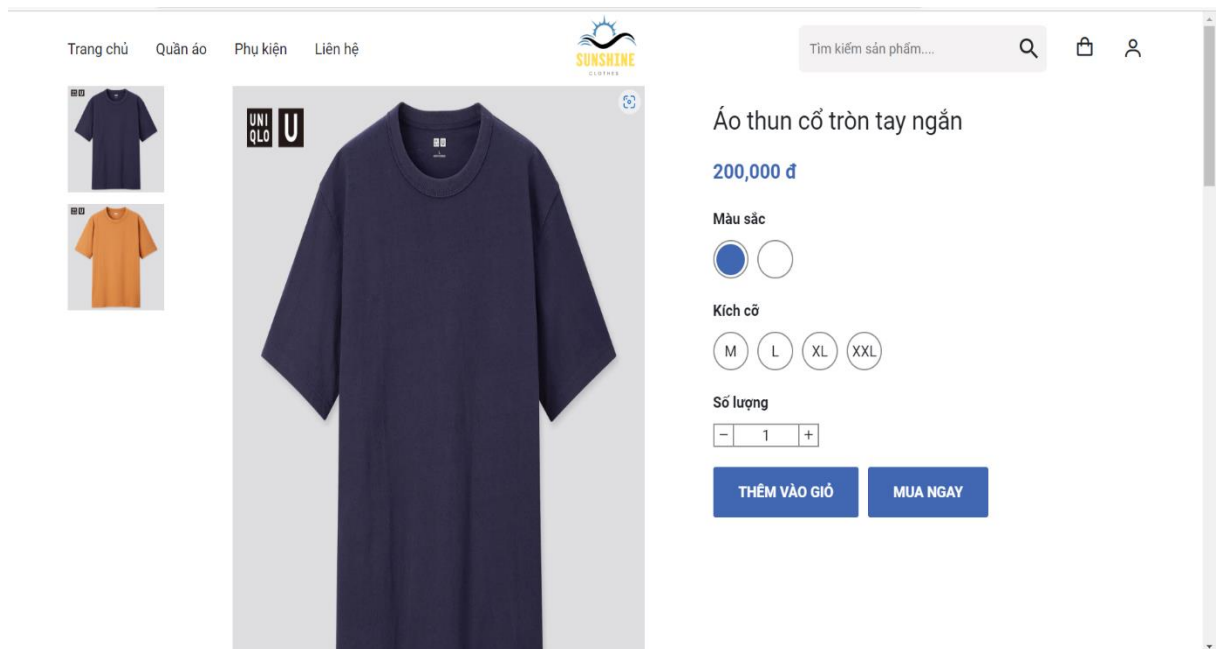
## 3.3 Trang sản phẩm phụ kiện



Hình IV-3 Trang chưa sản phẩm phụ kiện

Tính năng tương tự trang sản phẩm quần áo

### 3.4 Trang chi tiết sản phẩm



Hình IV-5 Trang chi tiết sản phẩm

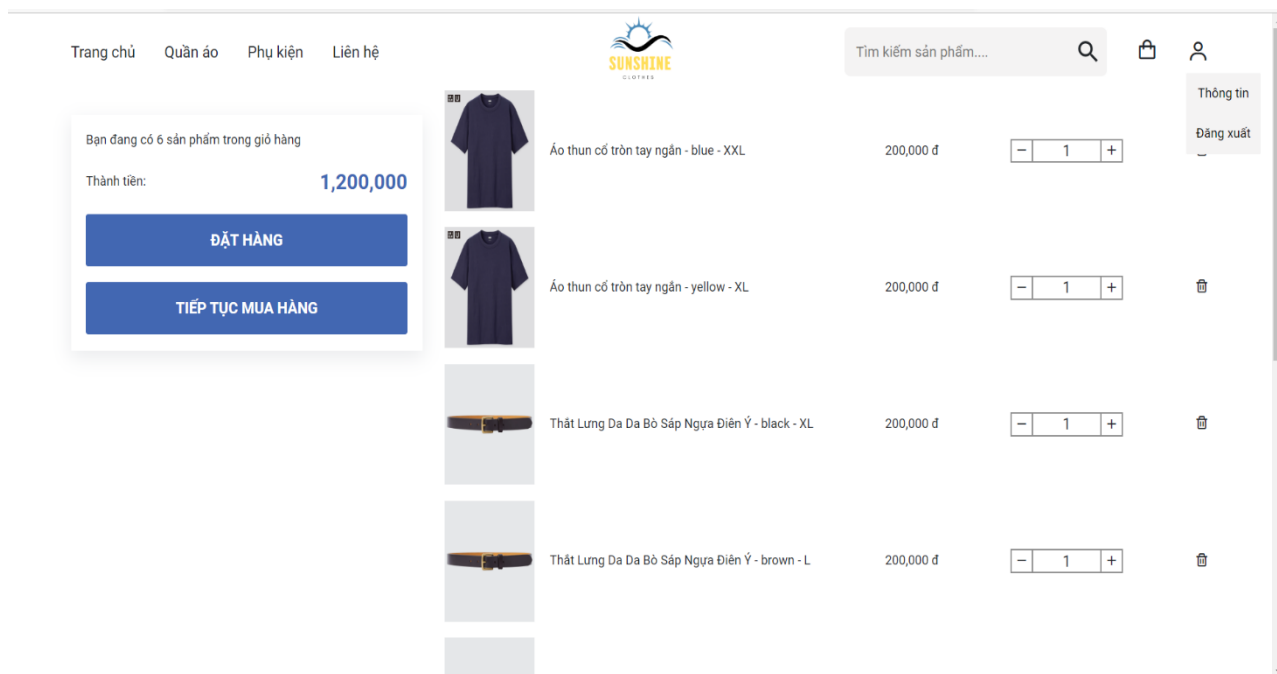
Chứa các lựa chọn cho người dùng.

Gồm button để click mua ngay và lưu lựa chọn sản phẩm và tiến vào giỏ hàng.

Và button click thêm vào giỏ để lưu và tiếp tục mua sản phẩm khác.

### 3.5 Trang giỏ hàng

**Trang giỏ hàng tổng quan:**

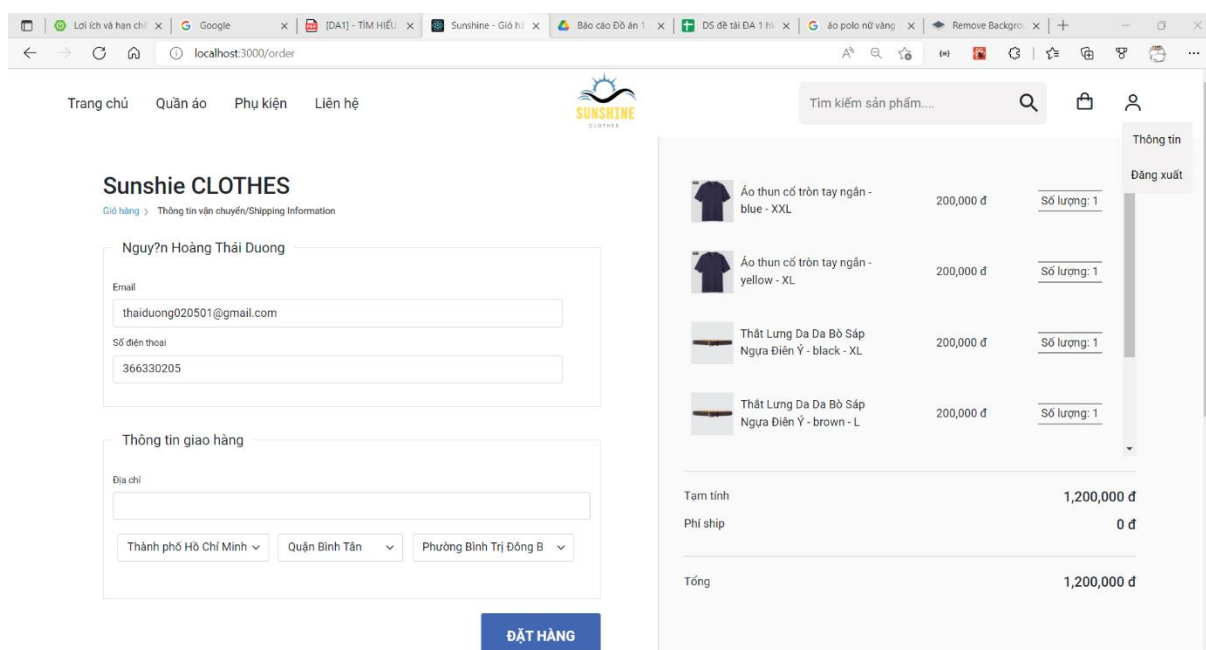


**Hình IV-6 Trang đặt hàng tổng**

Thẻ hiện toàn bộ sản phẩm đã chọn và giá tiền tổng.

Cho phép người dùng chỉnh sửa số lượng cũng như xóa nếu không muốn mua hàng.

**Trang giỏ hàng đặt hàng:**

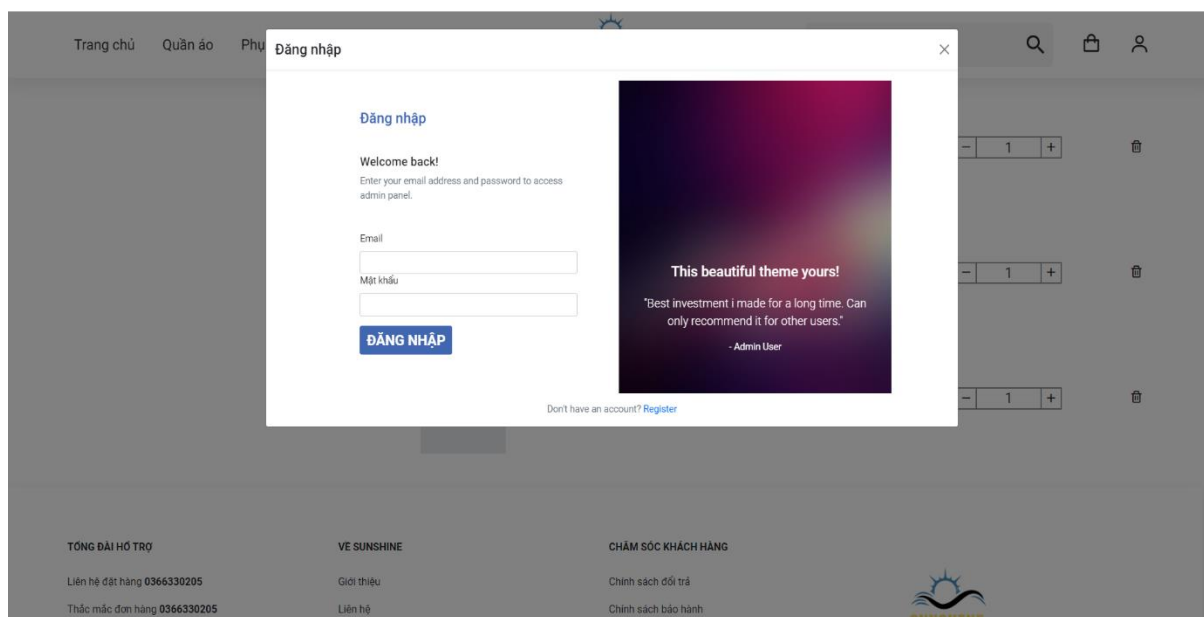


**Hình IV-7 Trang giỏ hàng đặt hàng**

Điều chỉnh thông tin đặt hàng thông tin liên lạc trước khi đặt hàng.

Thẻ hiện các sản phẩm đã chọn và tính giá tiền bao gồm tiền phí vận chuyển.

### 3.6 Trang đăng nhập



**Hình IV-8 Trang đăng nhập**

Là trang modal để đăng nhập

### 3.7 Trang đăng ký

Trang chủ Quần áo Phụ kiện Register Form

**Register**  
**Just Do Register.**  
If You Really Want To Know, Look In The Register.

Họ và tên

Email

Mật khẩu

Lặp lại Mật khẩu

**REGISTER**


Already have an account? [Login](#)

**This beautiful theme yours!**  
"Best investment i made for a long time. Can only recommend it for other users."  
- Admin User

TỔNG ĐÀI HỖ TRỢ VE SUNSHINE CHĂM SÓC KHÁCH HÀNG

Liên hệ đặt hàng 0366330205 Giới thiệu Chính sách đổi trả

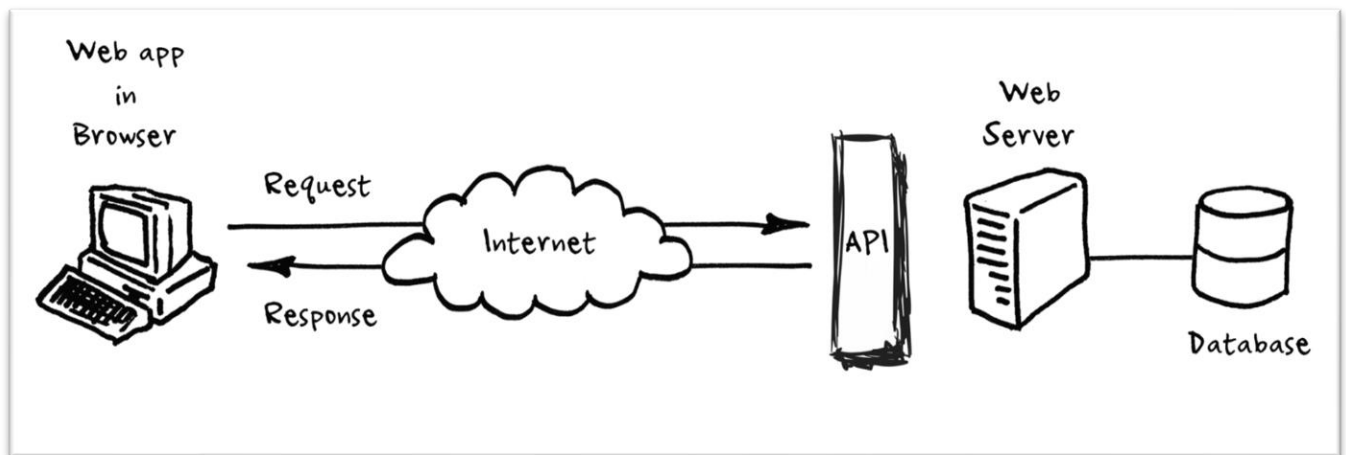
Thắc mắc đơn hàng 0366330205 Liên hệ Chính sách bảo hành



**Hình IV-9 Trang đăng ký**

Là trang modal để người dùng đăng ký tài khoản

## V. KIẾN TRÚC



Hình IV-1 Kiến trúc áp dụng

Ứng dụng được xây theo kiểu client server với hai tầng frontend ReactJs và backend viết bằng Java Spring Boot và hai tầng giao tiếp với nhau thông qua Restful API.

### 1. ReactJs

Khi người dùng tương tác với component ở phía UI ứng dụng, nguồn dữ liệu được lấy từ các nguồn:

- Truyền dữ liệu thông qua các component từ cha sang con.
- Lấy từ các state được lưu trữ thông qua store mà nhóm đã sử dụng ReduxToolkit để quản lý. Trong đó các một số state sẽ cần gọi api để lấy từ server. Vì vậy nhóm đã sử dụng createAsyncThunk trong ReduxToolkit để fetch data mà không cần phải tạo thêm các file phức tạp khác.

Do hiện tại nhóm vẫn chưa áp dụng được JWT nên hầu như các request sẽ không cần header hay token đi kèm.

## 2. Spring Boot

Request được gửi từ phía client là frontend sẽ được Controller điều hướng đến các service tương ứng để xử lý các business logic. Sau khi áp dụng các business logic, dữ liệu muốn gửi lên database hoặc các yêu cầu lấy dữ liệu từ database sẽ được xử lý bởi repository. Ở repository, chúng em sử dụng g Spring data JPA để có thể áp dụng nhiều database mà không cần phải viết lại query. Model có Entity mapping đến từng bảng trong database, khi muốn gửi một data nào đó lên database thì sẽ dùng Entity trong Model còn khi nhận data từ database thì sẽ hứng bằng DTO sau đó dựa liệu sẽ được trả về frontend hiển thị lên cho người dùng.

## **VI. KẾT QUẢ ĐẠT ĐƯỢC VÀ HƯỚNG PHÁT TRIỂN**

### **1. Kết quả đạt được**

Qua bài báo cáo trên, nhóm đã thực hiện được việc áp dụng các kiến thức mà mình tìm hiểu được về ReactJs – SpringBoot để thiết một website bán hàng cơ bản mà vẫn đáp ứng được các chức năng thiết yếu.

Quá trình thiết kế cũng như tìm hiểu đã được ghi nhận lại cho các lập trình viên hay các bạn sinh viên muốn sử dụng tài liệu tham khảo cho các chủ đề liên quan.

### **2. Hướng phát triển**

Hiện tại nhóm vẫn chưa ứng dụng phần JWT nên trong tương lai sẽ áp dụng các phần này vào sản phẩm. Ngoài ra còn một số chức năng như khuyến nghị sản phẩm, đồng bộ với tài khoản mạng xã hội và các kiến trúc thiết kế đang phổ biến trong việc thể Website cũng được nhóm đưa vào các dự định trong tương lai cho sản phẩm.

Thay đổi giao diện bắt mắt theo yêu cầu cũng như các đánh giá khách quan về UI trên các trang diễn đàn, học tập về lập trình,... để có thể phát triển ứng dụng. Sử dụng thêm các framework thiết kế khác như Tailwind,...



## VII. TÀI LIỆU THAM KHẢO

Ctrl+click để xuất hiện link

[1]: Nguyễn Hồng Thủy (2022), *Bài giảng môn ngôn ngữ lập trình Java*

[2]: Tổng quan về HTML-CSS-Javascript: [Lập trình HTML, CSS và JavaScript trong việc xây dựng website \(codegym.vn\)](#).

[3]: Redux: [https://www.youtube.com/watch?v=g\\_K1w8e0lLo&list=WL&index=3&t=2197s](https://www.youtube.com/watch?v=g_K1w8e0lLo&list=WL&index=3&t=2197s)

[4]: Lập trình ReactJs, HTML-CSS-Javascript: <https://fullstack.edu.vn/>

[5]: Tài liệu ReactJs:

**Giới thiệu ReactJs:** <https://viblo.asia/p/gioi-thieu-ve-reactjs-phan-i-cac-khai-niem-co-ban-V3m5WzjblO7>

**Props và State:** <https://viblo.asia/p/su-khac-nhau-giua-props-va-state-trong-reactjs-OeVKBvrJKkW>

**ReactJs Docs:** <https://reactjs.org/docs/getting-started.html>

**Component Lifecycle:** <https://viblo.asia/p/lifecycle-component-trong-reactjs-gGJ59jzxKX2>

**ReactHook:** <https://viblo.asia/p/cung-tim-hieu-ve-cac-hook-trong-react-hooks-Ljy5VYgjlra>

[6]: React-Router: <https://reactrouter.com/docs/en/v6>

[7] Tài liệu Spring Boot:

**Trang tài liệu Spring :** <https://spring.io/projects/spring-boot>

**Videos về Spring:** [https://www.youtube.com/watch?v=O\\_XL9oQ1\\_To&t=947s](https://www.youtube.com/watch?v=O_XL9oQ1_To&t=947s)

**Hibernate:** <https://topdev.vn/blog/hibernate-la-gi-sao-phai-dung-no-thay-jdbc/>

**Spring Data:** [Giới thiệu về Spring Data JPA - Tại sao cần sử dụng nó? - Học Spring Boot \(hocspringboot.net\)](#)